NXP software library (LPC) package code documentation version 1.0

Content

NXP software library (LPC) package code documentation version 1.0

1 Symbol Reference

1.1 Structs, Records, Enums

1.1.1 API_S

```
struct API_S {
  PFI open;
  PFI close;
  PFI read;
  PFI write;
  PFI ioctl;
};
```

File

lpc_api.h (see page 155)

Members

Members	Description
`PFI open;`	Open the device
`PFI close;`	Close the device
`PFI read;`	Read data from the device
`PFI write;`	Wrote data to the device
`PFI ioctl;`	Device control and configuration

Description

System API data structure

1.1.2 API_TABLE_S

```
struct API_TABLE_S {
  API_T driver;
  INT_32 id;
  INT_32 devid;
  INT_32 fd;
```

```
  INT_32 opened;
};
```

File

lpc_api.h (see page 155)

Members

Members	Description
API_T driver;	Device driver callbacks
INT_32 id;	Device Id
INT_32 devid;	Driver device id
INT_32 fd;	File descriptor
INT_32 opened;	Driver state

Description

Api system device lookup table

1.2 Functions

1.2.1 api_add_device

```
STATIC STATUS api_add_device(INT_32 id, void* open, void* close, void* read, void* write,
void* ioctl);
```

File

lpc_api.c (see page 154)

Parameters

Parameters	Description
INT_32 id	Device id
void* open	Driver open method
void* close	Driver close method
void* read	Driver read method
void* write	Driver write method
void* ioctl	Driver ioctl method
Outputs	None

Returns

_NO_ERROR (see page 100) if the device is added to the io system. _ERROR (see page 100) if the table is full or the name is not valid.

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: api_add_device

Purpose: To add a device to the api (see page 74) table

Processing: This function checks for a device id collision in the api (see page 74) system. If the id is valid it looks for a vacant entry. If the table is not full it binds itself to the api (see page 74) system.

1.2.2 api_find_device

```
STATIC INT_32 api_find_device(INT_32 id);
```

File

lpc_api.c (see page 154)

Parameters

Parameters	Description
INT_32 id	device id.
Outputs	None

Returns

index of the device bound to the id -1 if the device does not exist

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: api_find_device

Purpose: To find a device using a numerical representation

Processing: Search the device table for an id and return return the index of the device in the table.

1.2.3 api_find_empty

```
STATIC INT_32 api_find_empty(void);
```

File

lpc_api.c (see page 154)

Parameters

Parameters	Description
Outputs	None

Returns

index of the device bound to the name -1 if the device does not exist

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: api_find_empty

Purpose: To find a vacant table entry

Processing: Search the device table for a vacant space and return the index in the table.

1.2.4 api_remove_device

```
STATIC STATUS api_remove_device(INT_32 id);
```

File

lpc_api.c (see page 154)

Parameters

Parameters	Description
INT_32 id	Device id
Outputs	None

Returns

_NO_ERROR (see page 100) on success _ERROR (see page 100) on error

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Private methods

Function: api_remove_device

Purpose: To remove a device from the api (see page 74) table

Processing: This function finds the table entry that is associated with the devid. Once the entry is found it is cleared which will set it to the idle state. When a table entry is in the idle state a new device my use this entry to bind itself to the system.

1.2.5 bmp_allocate_structure

```
BMP_T * bmp_allocate_structure(INT_32 xsize, INT_32 ysize, BMP_STORAGE_T bits_per_pixel);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
INT_32 xsize	Horizontal size of the image storage space
INT_32 ysize	Vertical size of the image storage space
BMP_STORAGE_T bits_per_pixel	number of bits per pixel, used to set the size of the buffer and color table (enumerator)
Outputs	Nothing

Returns

A pointer to a new allocated BMP structure, or NULL (see page 143) if an error occurred.

Notes: The bits_per_pixel parameter is important for optimal memory usage. Setting this value will 'adjust' the sizing of the allocated BMP structure, modifying the sizes of the color table and data area. If unsure of the bits per pixel, use BPP24, extra memory will be allocated for BPP24, but no memory allocation problems will occur.

Description

Allocates storage for a new BMP file

Function: bmp_convert_image (see page 5)

Purpose: Allocates storage for a new BMP file structure.

Processing: This function computes the required size needed for the BMP header, color table, and image data, based on the color depth. Memory for an image (with header and color table) is allocated and the pointer returned to the caller.

1.2.6 bmp_convert_color

```
COLOR_T bmp_convert_color(BMP_COLOR_TABLE_T * color_entry);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
`BMP_COLOR_TABLE_T * color_entry`	Color table entry pointer
`Outputs`	None

Returns

A converted color_type entry from the color data

Notes: Not valid for 16-bit or 32-bit color formats.

Description

Converts a BMP color table entry to a color_type color

Function: bmp_convert_color

Purpose: Converts a BMP color table entry to a COLOR_T (see page 61) color

Processing: A color table entry (or raw 24-bit entry) is converted into the native (compiled) color type by masking and shifting the red, green, and blue components of color and computing the closest color in the native format (either 233, 555, or 565).

1.2.7 bmp_convert_image

```
BMP_STORAGE_T bmp_convert_image(BMP_T * bmp_data, INT_16 * xsize, INT_16 * ysize, COLOR_T *
bufout);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
`BMP_T * bmp_data`	pointer to a BMP data structure
`INT_16 * xsize`	Pointer to place the horizontal size of the image
`INT_16 * ysize`	Pointer to place the vertical size of the image
`COLOR_T * bufout`	Pointer to where to place the converted image
`Outputs`	None

Returns

Nothing

Notes: Only uncompressed 1, 4, 8, and 24 bit per pixel formats are supported. Before converting, be sure that the target buffer, bufout, is large enough for the converted image.

Description

Convert a BMP image to a color_type image

Function: bmp_convert_image

Purpose: Convert a BMP image to a COLOR_T (see page 61) image

Processing: See function.

1.2.8 bmp_get_color_table

```
BMP_COLOR_TABLE_T * bmp_get_color_table(BMP_T * bmp_data);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
`BMP_T * bmp_data`	Pointer to a BMP data structure.
`Outputs`	Nothing

Returns

A pointer to the color table, or 0x0 if one does not exist

Notes: 1, 4, and 8 bit per pixel BMP images have color tables.

Description

Returns a pointer to the color table

Function: bmp_get_color_table

Purpose: Returns a pointer to the color table

Processing: A call to bmp_is_header_valid (see page 7) is performed to determine the BMP file type. If the BMP file type is BPP1, BPP4, or BPP8, then the color table is assigned a pointer after the BMP header information.

1.2.9 bmp_get_image_data

```
void * bmp_get_image_data(BMP_T * bmp_data);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
`BMP_T * bmp_data`	Pointer to a BMP data structure.
`Outputs`	Nothing

Returns

A pointer to the BMP image data.

Notes: None

Description

Returns a pointer to the BMP image data

Function: bmp_get_image_data

Purpose: Returns a pointer to the BMP image data.

Processing: A call to bmp_is_header_valid (see page 7) is performed to determine the BMP file type. Based on the BMP file type, the number of entries in the color table is computed. The pointer to the image data is computed at the end of the header plus an offset for the color table.

1.2.10 bmp_is_header_valid

```
BMP_STORAGE_T bmp_is_header_valid(BMP_T * bmp_data);
```

File

lpc_bmp.h (see page 161)

Parameters

Parameters	Description
BMP_T * bmp_data	Pointer to a BMP data structure.
Outputs	None

Returns

Enumeration that defines the BMP color depth, or INVALID_BMP if the BMP type is unsupported.

Notes: None

Description

Determine if the structure is a BMP structure

Function: bmp_is_header_valid

Purpose: Determine if the structure is a BMP structure

Processing: The header type (bftype) is examined to match 'BM'. If it matches and the file type is uncompressed, then the color depth is examined and the return value set to the appropriate color depth enumeration. If an unsupported type is found, type INVALID_BMP will be returned.

1.2.11 cp15_dcache_flush

```
void cp15_dcache_flush(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

Nothing

Notes: None

Description

Force an data cache flush

Function: cp15_dcache_flush

Purpose: Force an data cache flush

Processing: Flush each data cache entry using the segment/index method.

1.2.12 cp15_force_cache_coherence

```
void cp15_force_cache_coherence(UNS_32 * start_adr, UNS_32 * end_adr);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 * start_adr	The first address in the code block
UNS_32 * end_adr	The last address in the code block
Outputs	None

Returns

Nothing

Notes: None

Description

Force cache coherence between memory and cache for the selected address range

Function: cp15_force_cache_coherence

Purpose: Force the CPU to recognize the block of code that was just written to memory between start_adr and end_adr even if caching and write buffering is on.

Processing: Cache lines are 32-bytes (8 words); clean and invalidate each line of D-cache and invalidate each line of I-cache within the address range.

Invalidate the I-TLB within the the address range. The I-TLB has 256 word granularity.

1.2.13 cp15_get_mmu_control_reg

```
UNS_32 cp15_get_mmu_control_reg(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

The current value of the MMU Control register (cp15) as an UNS_32 (see page 73)

Notes: None

Description

Return the current value of MMU Coprocessor(CP15) Control register

Function: cp15_get_mmu_control_reg

Purpose: To return the current value of the MMU Coprocessor (CP15) Control register.

Processing: Fetch the MMU control register to a variable and return it

1.2.14 cp15_get_ttb

```
UNS_32 * cp15_get_ttb(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None.

Returns

The base address of the MMU translation table

Notes: None

Description

Return the physical address of the MMU translation table

Function: cp15_get_ttb

Purpose: Return the physical address of the MMU translation table

Processing: Read the TTB register from coprocessor 15 and return it to the caller.

1.2.15 cp15_init_mmu_trans_table

```
BOOL_32 cp15_init_mmu_trans_table(TRANSTABLE_T * tt, TT_SECTION_BLOCK_T * ttsbp);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
TRANSTABLE_T * tt	address of Translation Table in RAM.
TT_SECTION_BLOCK_T * ttsbp	address of the beginning of the initialization array
Outputs	None.

Returns

This function returns _ERROR (see page 100) when the MMU is enabled, or the target address is not 16K aligned. Otherwise, it returns _NO_ERROR (see page 100).

Notes: This function is not intended to be used when the MMU is enabled.

Description

Setup MMU page tables

Function: cp15_init_mmu_trans_table

Purpose: Initializes the MMU page table

Processing: Return error if MMU is enabled. Return error if target Translation Table address is not 16K aligned. Clear the Translation Table area. Build the Translation Table from the initialization data in the Section Block array. Return no error.

1.2.16 cp15_invalidate_cache

```
void cp15_invalidate_cache(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

Nothing

Notes: This function invalidates all cache data including dirty data (data that has been modified in cache but not yet written to main memory). Use with caution. See ARM922T TRM.

Description

Invalidates the Instruction and Data caches

Function: cp15_invalidate_cache

Purpose: Invalidates the Instruction and Data caches

Processing: Use the ARM instruction to unconditionally invalidate the entire cache.

1.2.17 cp15_invalidate_tlb

```
void cp15_invalidate_tlb(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

Nothing

Notes: See the ARM922T TRM.

Description

Invalidates the Translation Lookaside Buffers

Function: cp15_invalidate_tlb

Purpose: Invalidates the Translation Lookaside Buffers

Processing: Use the ARM instruction to unconditionally invalidate the I- and D- TLBs.

1.2.18 cp15_map_physical_to_virtual

```
void * cp15_map_physical_to_virtual(UNS_32 addr);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 addr	The physical address to be converted
Outputs	None

Returns

The virtual address or 0 if the address does not translate.

Notes: None

Description

Get a virtual address from a passed physical address

Function: cp15_map_physical_to_virtual

Purpose: Return a virtual address for a passed physical address

Processing: Test if MMU is on, return if not. Search for the virtual address of the provided physical address. If found, return a void pointer to virtual address.

1.2.19 cp15_map_virtual_to_physical

```
UNS_32 cp15_map_virtual_to_physical(void * addr);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
void * addr	The virtual address to be converted
Outputs	None

Returns

The physical address or 0 if the address does not translate.

Notes: None

Description

Return a physical address for a passed virtual address

Function: cp15_map_virtual_to_physical

Purpose: Return a physical address for a passed virtual address

Processing: Return (UNS_32 (see page 73))addr if MMU is turned off. Otherwise, read the address of the translation table from the translation table base address register. Use the upper 12 bits of the addr to index the translation table and read out the descriptor. If the descriptor is invalid, return 0. If the descriptor is for a 1 Meg section, read back the upper 12 bits of the physical address. The lower 20 bits of the physical address is the lower 20 bits of the virtual address. If the descriptor is for a coarse page table, read the coarse page table descriptor and use the most significant 22 bits as the base address of the page table. If the descriptor is for a fine page table, read the fine page table descriptor and use the most significant 20 bits as the base address of the page table.

If not a section base, read the level 2 page descriptor from the page table. If bits 1..0 of the level2 descriptor are 01, then it is a large page table descriptor. The most significant 16 bits of the descriptor are the most significant 16 bits of the physical address; the least significant 16-bits of the virtual address are the least significant 16-bits of the address. If bits 1..0 of the level2 descriptor are 10, then it is a small page table descriptor. The most significant 20 bits of the level2 descriptor are the

most significant 20 bits of the physical address; the least significant 12 bits are the least significant 12 bits of the physical address. If bits 1..0 of the level2 descriptor are 11, then it is a tiny page table descriptor. The most significant 22 bits of the level2 descriptor are the most significant 22 bits of the physical address; the least significant 10 bits are the least significant 10 bits of the physical address. If bits 1..0 of the level2 descriptor are 0, return 0 (invalid).

1.2.20 cp15_mmu_enabled

```
BOOL_32 cp15_mmu_enabled(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

TRUE if the MMU is enabled FALSE if the MMU is disabled

Notes: None

Description

Checks to see if the MMU is enabled

Function: cp15_mmu_enabled

Purpose: Checks to see if the MMU is enabled

Processing: Read the MMU control register and check if the MMU enable bit (bit 0) is set.

1.2.21 cp15_set_dcache

```
void cp15_set_dcache(BOOL_32 enable);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
BOOL_32 enable	TRUE if the D-cache must be enabled FALSE if the D-cache must be disabled
Outputs	None

Returns

Nothing

Notes: None

Description

Enables or disables the data cache

Function: cp15_set_dcache

Purpose: Enables or disables the data cache

Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the D-cache enable bit, otherwise, clear it. Write the resultant value back to the control register.

1.2.22 cp15_set_domain_access

```
void cp15_set_domain_access(UNS_32 dac);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 dac	32-bit value encoded as follows: 31 29 27 25 23 21 19 17 15 13 11 9 8 7 6 5 4 3 2 1 0
Outputs	None

Returns

Nothing

Notes: See the ARM922T TRM.

Description

Define the access permissions for the 16 MMU domains.

Function: cp15_set_domain_access

Purpose: Define the access permissions for the 16 MMU domains.

Processing: Use the ARM instruction to write the value passed as argument to the domain access control regsiter.

1.2.23 cp15_set_icache

```
void cp15_set_icache(BOOL_32 enable);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
BOOL_32 enable	TRUE if the I-cache must be enabled FALSE if the I-cache must be disabled
Outputs	None

Returns

Nothing

Notes: None

Description

Enables or disables the instruction cache

Function: cp15_set_icache

Purpose: Enables or disables the instruction cache

Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the I-cache enable bit, otherwise, clear it. Write the resultant value back to the control register.

1.2.24 cp15_set_mmu

```
void cp15_set_mmu(BOOL_32 enable);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
BOOL_32 enable	TRUE if the MMU must be enabled FALSE if the MMU must be disabled
Outputs	None

Returns

Nothing

Notes: None

Description

Enable/Disable MMU

Function: cp15_set_mmu

Purpose: To enable or disable the MMU as specified.

Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the MMU enable bit, otherwise, clear it. Write the resultant value back to the control register.

1.2.25 cp15_set_mmu_control_reg

```
void cp15_set_mmu_control_reg(UNS_32 mmu_reg);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 mmu_reg	The value to be set in the MMU Control register (cp15).
Outputs	None

Returns

None

Notes: None

Description

Set MMU Coprocessor(CP15) Control register

Function: cp15_set_mmu_control_reg

Purpose: To set MMU Coprocessor (CP15) Control register.

Processing: Set the MMU control register to a value passed as parameter.

1.2.26 cp15_set_transtable_base

```
void cp15_set_transtable_base(UNS_32 addr);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 addr	Translation table base address
Outputs	None

Returns

Nothing

Notes: The address must be aligned on a 16K boundary. See ARM922T TRM.

Description

Sets the first-level translation table base address

Function: cp15_set_transtable_base

Purpose: Sets the first-level translation table base address

Processing: Masks out the lower 12 bits of the address passed. Writes register 2 of CP15 with the base address passed as parameter.

1.2.27 cp15_set_vmmu_addr

```
void cp15_set_vmmu_addr(UNS_32 * addr);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
UNS_32 * addr	Virtual address of start of MMU table
Outputs	None.

Returns

Nothing

Notes: This function must be called if the driver MMU functions are being used. This should be set after the call to the cp15_init_mmu_trans_table (see page 9)() function.

Description

Set the virtual address of the MMU table

Function: cp15_set_vmmu_addr

Purpose: Set the virtual address of the MMU table

Processing: Set the saved virtual MMU table address to the passed value.

1.2.28 cp15_write_buffer_flush

```
void cp15_write_buffer_flush(void);
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Parameters

Parameters	Description
Outputs	None

Returns

Nothing

Notes: None

Description

Force an write buffer flush

Function: cp15_write_buffer_flush

Purpose: Force an write buffer flush

Processing: Flush the write buffer and wait for completion of the flush.

1.2.29 fat16_cd

```
INT_32 fat16_cd(CHAR * path, FILE_TYPE * file_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
CHAR * path	Path of new directory
FILE_TYPE * file_data	Pointer to a FILE data structure to populate
Outputs	Data in file_data will be updated.

Returns

'1' if the operation was successful, '0' otherwise.

Notes: None

Description

- Directory management functions

****

Set the active directory

Function: fat16_cd

Purpose: Set the active directory.

Processing: Prior to any operations, the current directory index data is saved. If the first character is a '/', the directory pointer is set to the root directory. If the dir_commit flag is set, the cached directory will be written back to the device before the change.

The next name in the path will then be parsed. The active directory will be searched for the name. If the name is found, the cluster number to the new directory will be fetched and the new directory cached in. This process continues for all parsed names. If no errors occurred, the active directory index is updated to the new index. If an error occurred, the original directory and index are restored.

1.2.30 fat16_close_file

```
void fat16_close_file(FILE_TYPE * file_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
Outputs	None

Returns

Nothing

Notes: None

Description

Close a file that was open for reading or writing, or anything else (will destroy the file descriptor)

Function: fat16_close_file

Purpose: Close a file that was open for reading or writing.

Processing: See function.

1.2.31 fat16_compare

```
INT_32 fat16_compare(CHAR * source, CHAR * dest, INT_32 size);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
CHAR * source	Source address
CHAR * dest	Destination address
INT_32 size	Number of characters to compare
Outputs	Nothing

Returns

'1' if the strings are the same, '0' otherwise

Notes: None

Description

Compares two strings for similarity

Function: fat16_compare

Purpose: Simple data comparison routine.

Processing: Two strings are compared in lowercase up to the number of characters set by 'size'.

1.2.32 fat16_create_new_file_descriptor

```
FILE_TYPE * fat16_create_new_file_descriptor(FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT device structure
Outputs	None

Returns

A pointer to a new file descriptor or NULL (see page 143) if there was not enough memory available.

Notes: None

Description

File descriptor creation/destroy functions

****

Fills a file structure with the device and FAT data - multiple file structures can be created to access and control multiple files

Function: fat16_create_new_file_descriptor

Purpose: Creates a file structure with the device and FAT data.

Processing: Allocates memory for a new file descriptor. Sets the initial file mode to FINVALID. Links the FAT device structure to the file descriptor. Sets up and caches the default directory used with the file descriptor as the root directory with an initial directory index at the start of the directory table. Allocates space for data storage during file operations (read/write).

1.2.33 fat16_delete

```
INT_32 fat16_delete(FILE_TYPE * file_data, CHAR * name);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
CHAR * name	Name of file to delete
Outputs	None

Returns

'1' if the operation was successful, '0' otherwise.

Notes: None

Description

- Basic FAT16 filesystem functions

****

Deletes a file in the active directory

Function: fat16_delete

Purpose: Deletes a file in the active directory.

Processing: See function.

1.2.34 fat16_destroy_file_descriptor

```
void fat16_destroy_file_descriptor(FILE_TYPE * file_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file descriptor to free.
Outputs	None

Returns

Nothing

Notes: None

Description

Destroys a created file descriptor

Function: fat16_destroy_file_descriptor

Purpose: Destroys a created file descriptor.

Processing: Prior to destroying the file descriptor, a call to fat16_close is performed to write any data in the write buffer out to the device. If the directory has been changed in any way, the cached directory is written back to the device. The structures used in the file descriptor and the file descriptor itself are then de-allocated.

1.2.35 fat16_find_file

```
INT_32 fat16_find_file(CHAR * name, FILE_TYPE * file_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
CHAR * name	Unpadded 8.3 name to search for in the directory
FILE_TYPE * file_data	Pointer to a file data structure
Outputs	If the file was found, the structure pointed to by newdir will be populated with the file/directory information.

Returns

Index to matching directory structure in active dir, or (-1) if a match was not found.

Notes: None

Description

Finds and returns the directory structure of the passed name in the active directory

Function: fat16_find_file

Purpose: Finds and returns the directory structure of the passed name in the active directory.

Processing: See function.

1.2.36 fat16_find_free_cluster

```
UNS_16 fat16_find_free_cluster(FAT_DEVICE_TYPE * fat_data, UNS_16 cluster_start);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT device structure
UNS_16 cluster_start	Starting cluster in list where to search
Outputs	None

Returns

Next free cluster, or '0' if a free cluster was not found

Notes: None

Description

Find the next free cluster in the cluster list. Searches down from the passed cluster

Function: fat16_find_free_cluster

Purpose: Find the next free cluster in the cluster list. Searches down from the passed cluster.

Processing: See function.

1.2.37 fat16_get_active_mbr

```
INT_32 fat16_get_active_mbr(FAT_DEVICE_TYPE * fat_data, INT_32 use_active_only, INT_32
support_no_mbr);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
INT_32 use_active_only	Flag that indicates that active partions are used
INT_32 support_no_mbr	Flag that allows MBR-less device support
Outputs	None
file_data	Pointer to a file data structure

Returns

Nothing

Notes: None

Description

- Extended/extra functions

****

Returns an index to the first FAT partition

Function: fat16_get_active_mbr

Purpose: Returns an index to the first FAT partition.

Processing: See function.

1.2.38 fat16_get_dirname

```
INT_32 fat16_get_dirname(FILE_TYPE * file_data, CHAR * name, UNS_8 * etype, INT_32 * empty,
INT_32 * last);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
`FILE_TYPE * file_data`	Pointer to a file data structure
`CHAR * name`	Pointer of where to return name
`UNS_8 * etype`	Pointer of where to return dir entry type
`INT_32 * empty`	Pointer of where to return dir entry use flag
`INT_32 * last`	If set, this was the last entry in the directory
`Outputs`	None

Returns

The index to the active directory entry (only valid if empty and last are not set).

Notes: The type and empty flags should be checked after a call to this function. If empty is set(1), the dir entry is not used.

Description

Returns the name and type of the (next) entry in the active directory

Function: fat16_get_dirname

Purpose: Returns the name and type of the entry in the active directory (in unpadded 8.3 format).

Processing: See function.

1.2.39 fat16_get_free_dir_entry

```
INT_32 fat16_get_free_dir_entry(FILE_TYPE * file_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
`FILE_TYPE * file_data`	Pointer to a file data structure
`Outputs`	None

Returns

The index of the added dir entry, or (-1) if unsuccessful.

Notes: None

Description

Allocates a new directory entry for the passed name

Function: fat16_get_free_dir_entry

Purpose: Allocates a new directory entry for the passed name.

Processing: See function.

1.2.40 fat16_get_next_cluster

```
UNS_32 fat16_get_next_cluster(FAT_DEVICE_TYPE * fat_data, UNS_16 cluster_num);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
`FAT_DEVICE_TYPE * fat_data`	Pointer to a FAT device structure
`UNS_16 cluster_num`	Cluster number to use for next cluster search
`Outputs`	None

Returns

The next cluster in the list.

Notes: None

Description

Returns the next cluster in a cluster link chain

Function: fat16_get_next_cluster

Purpose: Returns the next cluster in a cluster link chain.

Processing: See function.

1.2.41 fat16_get_status

```
void fat16_get_status(FAT_DEVICE_TYPE * fat_data, UNS_8 * status, UNS_8 * ptype, INT_32
pnum);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
`FAT_DEVICE_TYPE * fat_data`	Pointer to a FAT data structure
`UNS_8 * status`	Pointer to status flag to populate
`UNS_8 * ptype`	Pointer to partition type flag to populate
`INT_32 pnum`	Partition number to return status on (1 - 4)

Outputs	The partition type and the status will be updated in memory pointed to by status and ptype. The only valid ptype and status values are (FAT16_LT32M (see page 132), FAT16_EXDOS (see page 132), FAT16_GT32M (see page 132)).

Returns

Nothing

Notes: Only partition numbers 1 through 4 are valid.

Description

- MBR functions

****

Get the status of the partition from the MBR

Function: fat16_get_status

Purpose: Get the status of the partition from the MBR.

Processing: Return the status and partition type values from the partition table in the FAT device structure.

1.2.42 fat16_init_device

```
FAT_DEVICE_TYPE * fat16_init_device(CHAR * device, ivfunc init_func, vvfunc shutdown_func,
ivfunc insert_ck_func, ivfunc ready_ck_func, ivfunc busy_ck_func, void (*set_sector_func)
(UNS_32), vvfunc start_read_func, vvfunc start_write_func, ivifunc read_func, ivifunc
write_func);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
CHAR * device	Device name
ivfunc init_func	Pointer to initialization function
vvfunc shutdown_func	Pointer to shutdown function
ivfunc insert_ck_func	Pointer to insertion check function
ivfunc ready_ck_func	Pointer to ready check function
ivfunc busy_ck_func	Pointer to bust check function
vvfunc start_read_func	Pointer to read start function
vvfunc start_write_func	Pointer to write start function
ivifunc read_func	Pointer to read buffer function
ivifunc write_func	Pointer to write buffer function
Outputs	Data in fat_data will be updated.
set_sector_func	Pointer to sector set function

Returns

The pointer to a binded device structure, or NULL (see page 143) if the device was not detected.

Notes: The calling function should check to make sure that NULL (see page 143) was not returned. If NULL (see page 143) was returned, the device does not exist or memory could not be allocated.

Description

Pointer for write of data

Function: fat16_init_device

Purpose: Initializes the FAT16 interface for the selected device.

Processing: Copy the device name and function pointers into the FAT device structure. Clear the commit flag to indicate the FAT cluster table does not need to be written back to the device. Call the device initialization function. If the device was

initialized, read the MBR into the FAT device structure.

1.2.43 fat16_moveto

```
void fat16_moveto(void * source, void * dest, INT_32 size);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
`void * source`	Source address
`void * dest`	Destination address
`INT_32 size`	Number of bytes to move
`Outputs`	Data pointed to by source will be updated.

Returns

Nothing

Notes: None

Description

- Support functions for the FAT16 driver

****

Moves a number of bytes from a source to destination

Function: fat16_moveto

Purpose: Simple data movement routine.

Processing: Move a number of bytes from the source to destination.

1.2.44 fat16_name_break

```
void fat16_name_break(CHAR * full_name, CHAR * name);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
`CHAR * full_name`	a name string in unpadded 8.3 format
`CHAR * name`	a name string in padded 8.3 format
`Outputs`	None

Returns

Nothing

Notes: None

Description

Converts a filename in unpadded 8.3 format to a format that is compatible with a directory format

Function: fat16_name_break

Purpose: Converts a filename in unpadded 8.3 format to a format that is compatible with a directory format.

Processing: See function.

1.2.45 fat16_name_check

```
INT_32 fat16_name_check(CHAR * name, ROOT_ENTRY_TYPE * dir_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
`CHAR * name`	Padded 8.3 name
`ROOT_ENTRY_TYPE * dir_data`	Pointer to a directory structure
`Outputs`	None

Returns

'1' if the name matches the directory entry name

Notes: None

Description

Compares a passed name in padded 8.3 format with a name in a directory entry structure

Function: fat16_name_check

Purpose: Compares a passed name in padded 8.3 format with a name in a directory entry structure.

Processing: Compare the first 11 characters of the passed name with the 11 characters in the passed directory structure.

1.2.46 fat16_open_file

```
INT_32 fat16_open_file(CHAR * name, FILE_TYPE * file_data, INT_32 mode);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
`CHAR * name`	Name of file
`FILE_TYPE * file_data`	Pointer to a FILE data structure to use
`INT_32 mode`	File mode (FREAD or FWRITE)
`Outputs`	None

Returns

'1' if the operation was successful, '0' otherwise.

Notes: None

Description

Open a file for reading or writing

Function: fat16_open_file

Purpose: Open a file for reading or writing.

Processing: See function.

1.2.47 fat16_parse_path

```
INT_32 fat16_parse_path(CHAR * path);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
CHAR * path	a path name string
Outputs	None

Returns

Size of data parsed if the operation was successful, otherwise -1.

Notes: None

Description

Finds the next directory name in a path

Function: fat16_parse_path

Purpose: Finds the next directory name in a path.

Processing: See function.

1.2.48 fat16_read

```
INT_32 fat16_read(FILE_TYPE * file_data, INT_32 bytes_to_copy, void * buffer_ptr, INT_32 *
bytes_copied, INT_32 * eof);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
INT_32 bytes_to_copy	Number of bytes to copy
void * buffer_ptr	Pointer to buffer to copy
INT_32 * bytes_copied	Pointer to where to return number of bytes copied
INT_32 * eof	Pointer to end of file flag, set on eof
Outputs	None

Returns

'1' if the operation was successful, '0' otherwise.

Notes: None

Description

Read data from a file

Function: fat16_read

Purpose: Read data from a file.

Processing: See function.

1.2.49 fat16_read_mbr

```
void fat16_read_mbr(FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a device data structure.
Outputs	Data in fat_data will be updated.

Returns

Nothing

Notes: None

Description

Reads the FAT MBR and puts the partition tables in the passed structure

Function: fat16_read_mbr

Purpose: Reads the FAT MBR and puts the partition tables in the passed structure.

Processing: Read CHS (0, 0, 1) from the device (this is always the MBR in a storage device). Copy the partition data from the device data into the partition data table. Set the selected active partition to (-1), indicating that a partition has not been selected.

1.2.50 fat16_read_sectors

```
void fat16_read_sectors(FAT_DEVICE_TYPE * fat_data, void * data, UNS_32 first_sector,
UNS_32 num_sectors);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a device data structure
void * data	Pointer to data buffer to fill
UNS_32 first_sector	Starting absolute sector to read
UNS_32 num_sectors	Number of sectors to read
Outputs	None

Returns

Nothing

Notes: None

Description

Reads a number of sectors from a device into a buffer

Function: fat16_read_sectors

Purpose: Reads a number of sectors from a device into a buffer.

Processing: See function.

1.2.51 fat16_save_all

```
void fat16_save_all(FILE_TYPE * file_data, FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT data structure
Outputs	None

Returns

Nothing

Notes: None

Description

Function: fat16_save_all

Purpose: Shutdown the FAT16 interface for the selected device.

Processing: If the commit flag is set, write the cached FAT cluster table back to the device. Free the allocated memory for the cluster table and device structure.

1.2.52 fat16_seek

```
INT_32 fat16_seek(FILE_TYPE * file_data, INT_32 seek_bytes);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
eof	Pointer to end of file flag, set on eof
Outputs	None
bytes_copied	Pointer to where to return number of bytes copied
bytes_to_copy	Number of bytes to copy
buffer_ptr	Pointer to buffer to copy

Returns

'1' if the operation was successful, '0' otherwise.

Notes: None

Description

Function: fat16_seek

Purpose: Seek data pointer.

Processing: See function.

1.2.53 fat16_set_dir_index

```
void fat16_set_dir_index(FILE_TYPE * file_data, INT_32 index);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
INT_32 index	DIR entry index to set the active dir entry to
Outputs	None

Returns

Nothing

Notes: None

Description

Resets the directory index to a location of the directory (used with get_dirname)

Function: fat16_set_dir_index

Purpose: Resets the directory index to a location of the directory (used with get_dirname)

Processing: See function.

1.2.54 fat16_set_no_mbr

```
void fat16_set_no_mbr(FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT device structure
Outputs	None

Returns

1 if the partition was mounted as FAT16, '-1' otherwise.

Notes: This function can be used to setup the cached partition table to use the fat16 functions without an MBR. (Some smaller storage devices may not have an MBR).

Description

Support function to set up the first partition in the driver to point to sector 1 for the boot record

Function: fat16_set_no_mbr

Purpose: Sets up the first partition in the cached parition table to point to sector 1 as a FAT16 boot record.

Processing: See function.

1.2.55 fat16_set_partition

```
INT_32 fat16_set_partition(INT_32 partnum, FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
INT_32 partnum	Partition number of set (1 - 4) on this device
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT data structure
Outputs	Data in fat_data will be updated.

Returns

'1' if the partition was set, '0' otherwise.

Notes: Only partition numbers 1 through 4 are valid.

Description

Set the active (FAT16) partition and cache cluster table

Function: fat16_set_partition

Purpose: Set the active partition.

Processing: If the partition is a valid type (FAT16), the starting sector value for the partition will be determined and the appropriate sector containing the boot record will be read from the device. Once the boot record has been read in, the partition dimensions are computed. Appropriate space for the FAT cluster table is allocated and the cluster table is cached in memory.

1.2.56 fat16_shutdown

```
void fat16_shutdown(FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a FAT data structure
Outputs	None

Returns

Nothing

Notes: None

Description

Shutdowns the FAT16 interface for the selected device (will destroy the FAT device structure)

Function: fat16_shutdown

Purpose: Shutdown the FAT16 interface for the selected device.

Processing: If the commit flag is set, write the cached FAT cluster table back to the device. Free the allocated memory for the cluster table and device structure.

1.2.57 fat16_translate_cluster_to_sector

```
UNS_32 fat16_translate_cluster_to_sector(FAT_DEVICE_TYPE * fat_data, UNS_16 cluster);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a device data structure
UNS_16 cluster	Cluster number
Outputs	None

Returns

An absolute sector number.

Notes: None

Description

Translate a cluster number to a (absolute) sector number

Function: fat16_translate_cluster_to_sector

Purpose: Translate a cluster number to a (absolute) sector number.

Processing: See function.

1.2.58 fat16_wait_busy

```
void fat16_wait_busy(FAT_DEVICE_TYPE * fat_data);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a device data structure.
Outputs	None

Returns

Nothing

Notes: None

Description

Wait for the device to go 'unbusy

Function: fat16_wait_busy

Purpose: Wait for the device to go 'unbusy'.

Processing: Check the status of the device busy function. If the device is busy, perform a small loop and check again until the device is no longer busy.

1.2.59 fat16_write

```
INT_32 fat16_write(FILE_TYPE * file_data, void * buffer_ptr, INT_32 bytes_to_copy);
```

File

lpc_fat16.h (see page 166)

Parameters

Parameters	Description
FILE_TYPE * file_data	Pointer to a file data structure
void * buffer_ptr	Pointer to buffer to copy
INT_32 bytes_to_copy	Number of bytes to write
Outputs	None

Returns

'1' if the operation was successful, '0' if the device is out of storage space.

Notes: None

Description

Write data to a file

Function: fat16_write

Purpose: Write data to a file.

Processing: See function.

1.2.60 fat16_write_sectors

```
void fat16_write_sectors(FAT_DEVICE_TYPE * fat_data, void * data, UNS_32 first_sector,
UNS_32 num_sectors);
```

File

lpc_fat16_private.h (see page 169)

Parameters

Parameters	Description
FAT_DEVICE_TYPE * fat_data	Pointer to a device data structure
void * data	Pointer to data buffer to copy from
UNS_32 first_sector	Starting absolute sector to write
UNS_32 num_sectors	Number of sectors to write
Outputs	None

Returns

Nothing

Notes: None

Description

Writes a number of sectors from a buffer to a device

Function: fat16_write_sectors

Purpose: Writes a number of sectors from a buffer to a device.

Processing: See function.

1.2.61 lpc_api_init

```
EXTERN void lpc_api_init(void* cfg);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
Outputs	None
config	Not used

Returns

None

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Public APIs used to access device drivers that are registered with the API sub system.

Function: lpc_api_init

Purpose: To initialize the api (see page 74) system

Processing: This function clears the api (see page 74) system table and marks it as initialized. Once the table has been initialized the devices can be bound to the io system and make use of the common API.

1.2.62 lpc_api_register

```
EXTERN INT_32 lpc_api_register(INT_32 devid, void* open, void* close, void* read, void*
write, void* ioctl);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
void* open	driver open method
void* close	driver close method
void* read	driver read method
void* write	driver write method
void* ioctl	driver io control method
Outputs	None
id	device id.

Returns

None

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: lpc_api_register

Purpose: To register a device with the system

Processing: This funtion is used to bind a device to the system. Once bound the device can make use of the common API

layer.

1.2.63 lpc_close

```
EXTERN INT_32 lpc_close(INT_32 fd);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
INT_32 fd	file descriptor of the device to be closed
Outputs	None

Returns

_NO_ERROR (see page 100) if the device has been closed _ERROR (see page 100) if the device could not be closed

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: lpc_close

Purpose: closes a session with an device driver

Processing: This routine marks the device as closed and then calls the associated close method at the device driver layer to disable the hardware.

1.2.64 lpc_colors_set_palette

```
void lpc_colors_set_palette(UNS_16 * palette_table);
```

File

lpc_colors.h (see page 164)

Parameters

Parameters	Description
UNS_16 * palette_table	Pointer of where to put the 256 8-bit to 16-bit palette conversion entries.
Outputs	None

Returns

Nothing

Notes: If compiled in 16-bit color mode, this will be a NULL (see page 143) function. Select the appropriate define in this function for 555 or 565 color mode displays when using an 256 color frame buffer.

Description

Generate a palette table (only in 8-bit mode). If compiled in 16-bit color mode, this will be a NULL (see page 143) function.

Function: lpc_colors_set_palette

Purpose: Generate a palette table (only in 8-bit mode).

Processing: Depending on the target LCD color mapping (either 555 or 565), a palette table will be generated to convert colors stored in 233 format to either 555 or 565 format through a lookup table.

1.2.65 lpc_free

```
INT_32 lpc_free(void * free_addr);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
void * free_addr	Address of allocated entry to return to heap
Outputs	None

Returns

'1' if the entry was deleted, otherwise '0'.

Notes: None

Description

Return an allocated area to the heap

Function: lpc_free

Purpose: Returns an allocated entry of memory to the heap.

Processing: See function.

1.2.66 lpc_get_allocated_count

```
UNS_32 lpc_get_allocated_count(void);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
Outputs	None

Returns

The number of allocated heap entries.

Notes: None

Description

Return the number of allocated items in the heap

Function: lpc_get_allocated_count

Purpose: Return the number of allocated items in the heap.

Processing: This function traverses through the heap list. If an entry has an available size of 0 bytes, then the entry is assumed as allocated and the allocated count is incremented.

1.2.67 lpc_get_heap_base

```
void * lpc_get_heap_base(void);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
Outputs	None

Returns

The base address of where heap memory starts.

Notes: None

Description

Return the heap base address

Function: lpc_get_heap_base

Purpose: Return the heap base address.

Processing: See function.

1.2.68 lpc_get_heapsize

```
UNS_32 lpc_get_heapsize(void);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
Outputs	None

Returns

The size of the heap area in bytes.

Notes: None

Description

Return the size of the heap area

Function: lpc_get_heapsize

Purpose: Returns the size of the heap.

Processing: See function.

1.2.69 lpc_get_largest_chunk

```
UNS_32 lpc_get_largest_chunk(void);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
Outputs	None

Returns

The size of the largest chunk available in the heap area in bytes.

Notes: None

Description

Return the size of the largest unallocated heap chunk

Function: lpc_get_largest_chunk

Purpose: Returns the largest available chunk in the heap.

Processing: This function traverses through the heap list. If an entry has an available size of greater than 0 bytes, then the entry is assumed as free and the size of the chunk is compared to the running size count. If the size is larger, the running size count is updated with the new size.

1.2.70 lpc_heap_init

```
void lpc_heap_init(void * base_addr, UNS_32 heap_size);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
void * base_addr	Base address of where heap starts
UNS_32 heap_size	Size of heap area in bytes
Outputs	None

Returns

Nothing

Notes: None

Description

Setup the heap area

Function: lpc_heap_init

Purpose: Setup the heap area.

Processing: The heap base address and size counters are set with the passed parameter values. The first entry of the heap is set up with an unallocated heap list entry.

1.2.71 lpc_ioctl

```
EXTERN INT_32 lpc_ioctl(INT_32 fd, INT_32 cmd, INT_32 arg);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
INT_32 fd	device file descriptor.
INT_32 cmd	command to execute.
INT_32 arg	generic arg.
Outputs	None

Returns

_ERROR (see page 100) if the operation failed return code of the ioctl associated with the io system.

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: lpc_ioctl

Purpose: device io control routine

Processing: This routine controls the associated device driver via the callback method that has been bound to a driver. If the device is not registered -1 is returned else return code by the driver ioctl is returned.

1.2.72 lpc_new

```
void * lpc_new(UNS_32 size_in_bytes);
```

File

lpc_heap.h (see page 173)

Parameters

Parameters	Description
UNS_32 size_in_bytes	Byte size of the requested allocation chunk
Outputs	None

Returns

A pointer to the allocated chunk, or '0' if no room is available.

Notes: None

Description

Get an allocated area from the heap

Function: lpc_new

Purpose: Get an allocated area from the heap.

Processing: See function.

1.2.73 lpc_open

```
EXTERN INT_32 lpc_open(INT_32 devid, INT_32 arg);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
INT_32 arg	Options used to open the device
Outputs	None
id	Device id to open

Returns

device file decriptor -1 if the device does not exist

Notes: See sma_iosys.h for structure definitions

Description

Function: lpc_open

Purpose: Connects to a system device

Processing: This routine calls the associated open method in the io subsystem array. If the device asscoiated with the name is not registered an error -1 is returned. If the device is registered and not already opened a file descriptor that uniquely identifies this device is returned.

1.2.74 lpc_read

```
EXTERN INT_32 lpc_read(INT_32 fd, CHAR* buffer, INT_32 max_bytes);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
INT_32 fd	device file descriptor.
CHAR* buffer	data buffer.
INT_32 max_bytes	max number of bytes to read.
Outputs	None

Returns

-1 if the device is not registered. actual number of bytes read.

Notes: See lpc_api.h (see page 155) for structure definitions

Description

Function: lpc_read

Purpose: reads data from a registered api (see page 74) system device.

Processing: This routine reads data from a registered api (see page 74) device by using the callback method that has been bound to a driver. If the device is not registered -1 is returned. If the device is registered the user can pass in a buffer and a max number of bytes for the driver to use.

1.2.75 lpc_write

```
EXTERN INT_32 lpc_write(INT_32 fd, CHAR* buffer, INT_32 n_bytes);
```

File

lpc_api.h (see page 155)

Parameters

Parameters	Description
INT_32 fd	device file descriptor.
CHAR* buffer	generic arg.
INT_32 n_bytes	number of bytes contained in the arg.
Outputs	None

Returns

-1 if the write operation failed number of bytes written.

Notes: See sma_iosys.h for structure definitions

Description

Function: lpc_write

Purpose: write data to a registered device

Processing: This routine writes data to a registered api (see page 74) device by using the callback method that has been bound to a driver. If the device is not registered -1 is returned. If the device is registered a generic pointer and the number of bytes represented by the pointer are being passed to the

1.2.76 swim_clear_screen

```
void swim_clear_screen(SWIM_WINDOW_T * win, COLOR_T colr);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
COLOR_T colr	Color to place in the window
Outputs	None

Returns

Nothing

Notes: None

Description

Fills the draw area of the display with the selected color

Function: swim_clear_screen

Purpose: Fills the draw area of the display with the selected color

Processing: Loop through all virtual window (draw area) locations and updates them with the passed color value.

1.2.77 swim_get_font_height

```
INT_16 swim_get_font_height(SWIM_WINDOW_T * win);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None

Returns

The height of the active font in pixels.

Notes: None

Description

Returns the active font's height in pixels

Function: swim_get_font_height

Purpose: Returns the active font's height in pixels

Processing: See function.

1.2.78 swim_get_horizontal_size

```
INT_32 swim_get_horizontal_size(SWIM_WINDOW_T * win);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None

Returns

The virtual window horizontal size

Notes: None

Description

Get the virtual window horizontal size

Function: swim_get_horizontal_size

Purpose: Get the virtual window horizontal size

Processing: For the passed window ID, return the x size of the window.

1.2.79 swim_get_vertical_size

```
INT_32 swim_get_vertical_size(SWIM_WINDOW_T * win);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None

Returns

The virtual window horizontal size

Notes: None

Description

Get the virtual window vertical size

Function: swim_get_vertical_size

Purpose: Get the virtual window vertical size

Processing: For the passed window ID, return the x size of the window.

1.2.80 swim_get_xy

```
void swim_get_xy(SWIM_WINDOW_T * win, INT_32 * x, INT_32 * y);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 * x	Address of where to return virtual X value
INT_32 * y	Address of where to return virtual X value
Outputs	None

Returns

Nothing

Notes: X, Y coords are in virtual pixels!

Description

Returns the X, Y pixel coordinates for the next text operation

Function: swim_get_xy

Purpose: Returns the X, Y pixel coordinates for the next text operation

Processing: The logical X and Y positions are computed by subtracting the physical text position values by the physical minimum window limits.

1.2.81 swim_put_box

```
void swim_put_box(SWIM_WINDOW_T * win, INT_32 x1, INT_32 y1, INT_32 x2, INT_32 y2);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 x1	Virtual left position of box
INT_32 y1	Virtual upper position of box
INT_32 x2	Virtual right position of box

INT_32 y2	Virtual lower position of box
Outputs	None

Returns

Nothing

Notes: None

Description

Place a box with corners (X1, Y1) and (X2, Y2). Use pen color for edges and fill color for center

Function: swim_put_box

Purpose: Place a box with corners (X1, Y1) and (X2, Y2)

Processing: See function.

1.2.82 swim_put_char

```
void swim_put_char(SWIM_WINDOW_T * win, const CHAR textchar);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const CHAR textchar	Text string to output in window
Outputs	None

Returns

Nothing

Notes: None

Description

Puts a single character to the window

Function: swim_put_char

Purpose: Puts a character in the window.

Processing: See function.

1.2.83 swim_put_diamond

```
void swim_put_diamond(SWIM_WINDOW_T * win, INT_32 x, INT_32 y, INT_32 rx, INT_32 ry);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 x	Virtual X position of the diamond
INT_32 y	Virtual Y position of the diamond
INT_32 rx	Radius for horizontal

INT_32 ry	Radius for vertical
Outputs	None

Returns

Nothing

Notes: This function supports clipping.

Description

Draw a diamond in the virtual window

Function: swim_put_diamond

Purpose: Purpose: Draw a diamond in the virtual window

Processing: See function.

1.2.84 swim_put_image

```
void swim_put_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32 ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	Pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts a raw image into a window

Function: swim_put_image

Purpose: Puts an raw image in a window unscaled, clips off edges

Processing: See function.

1.2.85 swim_put_invert_image

```
void swim_put_invert_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32
ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier

const COLOR_T * image	Pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts a raw image into a window inverted

Function: swim_put_invert_image

Purpose: Puts an raw image in a window unscaled, inverted, with clipped edges.

Processing: See function.

1.2.86 swim_put_left_image

```
void swim_put_left_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32
ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	Pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts a raw image into a window rotated left

Function: swim_put_left_image

Purpose: Puts an raw image in a window unscaled, rotated left, with clipped edges.

Processing: See function.

1.2.87 swim_put_line

```
void swim_put_line(SWIM_WINDOW_T * win, INT_32 x1, INT_32 y1, INT_32 x2, INT_32 y2);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 x1	Virtual X position of X line start
INT_32 y1	Virtual Y position of Y line start
INT_32 x2	Virtual X position of X line end
INT_32 y2	Virtual Y position of Y line end
Outputs	None

Returns

Nothing

Notes: This function supports clipping.

Description

Draw a line in the virtual window

Function: swim_put_line

Purpose: Draw a line in the virtual window with clipping.

Processing: See function.

1.2.88 swim_put_ltext

```
void swim_put_ltext(SWIM_WINDOW_T * win, const CHAR * text);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const CHAR * text	Text string to output in window
Outputs	None

Returns

Nothing

Notes: None

Description

Puts a null-terminated string of text in a window, but will move an entire word to the next line if it will not fit on the present line

Function: swim_put_ltext

Purpose: Puts a string of text in a window, but will adjust the position of a word if the word length exceeds the edge of the display.

Processing: While the string has data in it, check for the newline character. If it exists, output a newline. If the string data is inside the font character table, output the first word in the string (with support for generating a newline if the word will exceed the window edge). Continue until all words/characters are output.

1.2.89 swim_put_newline

```
void swim_put_newline(SWIM_WINDOW_T * win);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None

Returns

Nothing

Notes: None

Description

Puts a newline in the window

Function: swim_put_newline

Purpose: Performs a newline in a window

Processing: Set the text pointer for the next text character operation to the beginning of the following line. If the following lin exceeds the window size, perform a line scroll.

1.2.90 swim_put_pixel

```
void swim_put_pixel(SWIM_WINDOW_T * win, INT_32 x1, INT_32 y1);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 x1	Virtual X position of pixel
INT_32 y1	Virtual Y position of pixel
Outputs	None

Returns

Nothing

Notes: The pixel will not be displayed if the pixel exceeds the window virtual size. Pixel positions below 0 should not be used with this function.

Description

Puts a pixel at (X, Y) in the pen color

Function: swim_put_pixel

Purpose: Puts a pixel at the virtual X, Y coordinate in the window

Processing: Convert the virtual pixel position to a physical position. If the pixel is inside the window draw area, update the pixel on the display.

1.2.91 swim_put_right_image

```
void swim_put_right_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32
```

```
ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	Pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts a raw image into a window rotated right

Function: swim_put_right_image

Purpose: Puts an raw image in a window unscaled, rotated right, with clipped edges.

Processing: See function.

1.2.92 swim_put_scale_image

```
void swim_put_scale_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32
ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts and scales a raw image into a window

Function: swim_put_scale_image

Purpose: Puts an raw image in a window scaled.

Processing: See function.

1.2.93 swim_put_scale_invert_image

```
void swim_put_scale_invert_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize,
INT_32 ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts and scales a raw image into a window inverted

Function: swim_put_scale_invert_image

Purpose: Puts an raw image in a window scaled and inverted.

Processing: See function.

1.2.94 swim_put_scale_left_image

```
void swim_put_scale_left_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize,
INT_32 ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts and scales a raw image into a window rotated left

Function: swim_put_scale_left_image

Purpose: Puts an raw image in a window scaled and rotated left.

Processing: See function.

1.2.95 swim_put_scale_right_image

```
void swim_put_scale_right_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize,
INT_32 ysize);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
Outputs	None

Returns

Nothing

Notes: Pixels should be organized in the image from left to right, top to bottom. (BMP images are not stored like this.)

Description

Puts and scales a raw image into a window rotated right

Function: swim_put_scale_right_image

Purpose: Puts an raw image in a window scaled and rotated right.

Processing: See function.

1.2.96 swim_put_text

```
void swim_put_text(SWIM_WINDOW_T * win, const CHAR * text);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const CHAR * text	Text string to output in window
Outputs	None

Returns

Nothing

Notes: None

Description

Puts a null-terminated string of text in a window

Function: swim_put_text

Purpose: Puts a string of text in a window

Processing: Each character will be routed to the swim_put_char (see page 43) function until a string terminator is reached. For newline characters, a newline will occur instead of a character output.

1.2.97 swim_put_text_xy

```
void swim_put_text_xy(SWIM_WINDOW_T * win, const CHAR * text, INT_32 x, INT_32 y);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const CHAR * text	Text string to output in window
INT_32 x	Virtual X position of start of text
INT_32 y	Virtual Y position of start of text
Outputs	None

Returns

Nothing

Notes: X, Y coords are in virtual pixels!

Description

Put a text message at an X, Y pixel coordinate in the window

Function: swim_put_text_xy

Purpose: Put text at x, y (char) position on screen

Processing: Set the virtual (upper left) text position in the window and render the text string at this position.

1.2.98 swim_put_win_image

```
void swim_put_win_image(SWIM_WINDOW_T * win, const COLOR_T * image, INT_32 xsize, INT_32
ysize, INT_32 scale, SWIM_ROTATION_T rtype);
```

File

lpc_swim_image.h (see page 182)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const COLOR_T * image	pointer to image data, must be in display color format
INT_32 xsize	Size of the image in horizontal pixels
INT_32 ysize	Size of the image in vertical pixels
INT_32 scale	If set, the picture will be scaled to the window size If not set, the picture will be clipped
SWIM_ROTATION_T rtype	Rotation type flag, either Norotation, Left, Right, or Invert
Outputs	None

Returns

Nothing

Notes: None

Description

One API for all the functions

Function: swim_put_win_image

Purpose: This function simply provides a single API for all the image functions.

Processing: See function.

1.2.99 swim_set_bkg_color

```
void swim_set_bkg_color(SWIM_WINDOW_T * win, COLOR_T bkg_color);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None
tbkg_color	New background color

Returns

Nothing

Notes: None

Description

Set background color

Function: swim_set_bkg_color

Purpose: Sets the color used for backgrounds

Processing: For the passed window ID, update to the passed background color.

1.2.100 swim_set_fill_color

```
void swim_set_fill_color(SWIM_WINDOW_T * win, COLOR_T fill_color);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
COLOR_T fill_color	New fill color
Outputs	None

Returns

Nothing

Notes: None

Description

Set fill color (used for boxes and circles)

Function: swim_set_fill_color

Purpose: Sets the fill color

Processing: For the passed window ID, update to the passed fill color.

1.2.101 swim_set_font

```
void swim_set_font(SWIM_WINDOW_T * win, FONT_T * font);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
FONT_T * font	Pointer to the selected font data structure
Outputs	None

Returns

Nothing

Notes: None

Description

Select the active font

Function: swim_set_font

Purpose: Sets the active font

Processing: Switch to the selected font by setting the font structure pointer in the windows structure based on the passed enumeration. If the next character output in the new font will exceed the window limit, perform a window text scroll.

1.2.102 swim_set_font_trasparency

```
void swim_set_font_trasparency(SWIM_WINDOW_T * win, INT_32 trans);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 trans	When not 0, the font backgrounds will not be drawn
Outputs	None

Returns

Nothing

Notes: None

Description

Enables and disables font backgrounds

Function: swim_set_font_trasparency

Purpose: Enables and disables font backgrounds. When set, the font background will not be drawn in the background color

(useful for painting text over pictures).

Processing: See function.

1.2.103 swim_set_pen_color

```
void swim_set_pen_color(SWIM_WINDOW_T * win, COLOR_T pen_color);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
COLOR_T pen_color	New pen color
Outputs	None

Returns

Nothing

Notes: None

Description

Set the pen color

Function: swim_set_pen_color

Purpose: Sets the pen color

Processing: For the passed window ID, update to the passed pen color.

1.2.104 swim_set_title

```
void swim_set_title(SWIM_WINDOW_T * win, const CHAR * title, COLOR_T ttlbkcolor);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
const CHAR * title	title string to use for window
COLOR_T ttlbkcolor	Background color in title area
Outputs	None

Returns

Nothing

Notes: Do not call this function more than once for a window or problems may occur.

Description

Create a title bar

Function: swim_set_title

Purpose: Creates a title bar in the window and adjusts the client area to be outside the title bar area.

Processing: See function.

1.2.105 swim_set_xy

```
void swim_set_xy(SWIM_WINDOW_T * win, INT_32 x, INT_32 y);
```

File

lpc_swim_font.h (see page 180)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
INT_32 x	Virtual X position of start of text
INT_32 y	Virtual Y position of start of text
Outputs	None

Returns

Nothing

Notes: X, Y coords are in virtual pixels!

Description

Sets the X, Y pixel coordinates for the next text operation

Function: swim_set_xy

Purpose: Sets the X, Y pixel coordinates for the next text operation

Processing: Update the X, Y text position pointers, limiting the position to the window dimensions.

1.2.106 swim_window_close

```
void swim_window_close(SWIM_WINDOW_T * win);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Window identifier
Outputs	None

Returns

Nothing

Notes: This is a defunct function and is not needed.

Description

Destroy a window

Function: swim_window_close

Purpose: Reallocates a window for use

Processing: For the passed window ID, clear the window used flag.

1.2.107 swim_window_open

```
BOOL_32 swim_window_open(SWIM_WINDOW_T * win, INT_32 xsize, INT_32 ysize, COLOR_T * fbaddr,
INT_32 xwin_min, INT_32 ywin_min, INT_32 xwin_max, INT_32 ywin_max, INT_32 border_width,
COLOR_T pcolor, COLOR_T bkcolor, COLOR_T fcolor);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Preallocated windows structure to fill
INT_32 xsize	Physical horizontal dimension of the display
INT_32 ysize	Physical vertical dimension of the display
COLOR_T * fbaddr	Address of the display's frame buffer
INT_32 xwin_min	Physical window left coordinate
INT_32 ywin_min	Physical window top coordinate
INT_32 xwin_max	Physical window right coordinate
INT_32 ywin_max	Physical window bottom coordinate
INT_32 border_width	Width of the window border in pixels
COLOR_T pcolor	Pen color
COLOR_T bkcolor	Background color
COLOR_T fcolor	Fill color
Outputs	None

Returns

TRUE if the window was initialized correctly, otherwise FALSE

Notes: This function must be called prior to any other window function

Description

Initialize a window

Function: swim_window_open

Purpose: Initializes a window and the default values for the window

Processing: See function.

1.2.108 swim_window_open_noclear

```
BOOL_32 swim_window_open_noclear(SWIM_WINDOW_T * win, INT_32 xsize, INT_32 ysize, COLOR_T *
fbaddr, INT_32 xwin_min, INT_32 ywin_min, INT_32 xwin_max, INT_32 ywin_max, INT_32
border_width, COLOR_T pcolor, COLOR_T bkcolor, COLOR_T fcolor);
```

File

lpc_swim.h (see page 178)

Parameters

Parameters	Description
SWIM_WINDOW_T * win	Preallocated windows structure to fill
INT_32 xsize	Physical horizontal dimension of the display
INT_32 ysize	Physical vertical dimension of the display
COLOR_T * fbaddr	Address of the display's frame buffer
INT_32 xwin_min	Physical window left coordinate
INT_32 ywin_min	Physical window top coordinate

INT_32 xwin_max	Physical window right coordinate
INT_32 ywin_max	Physical window bottom coordinate
INT_32 border_width	Width of the window border in pixels
COLOR_T pcolor	Pen color
COLOR_T bkcolor	Background color
COLOR_T fcolor	Fill color
Outputs	None

Returns

TRUE if the window was initialized correctly, otherwise FALSE

Notes: This function must be called prior to any other window function

Description

Initialize a window without clearing it

Function: swim_window_open_noclear

Purpose: Initializes a window and the default values for the window

Processing: See function.

1.3 Types

1.3.1 API_T

```
typedef struct API_S {
  PFI open;
  PFI close;
  PFI read;
  PFI write;
  PFI ioctl;
} API_T, * PAPI_T;
```

File

lpc_api.h (see page 155)

Members

Members	Description
PFI open;	Open the device
PFI close;	Close the device
PFI read;	Read data from the device
PFI write;	Wrote data to the device
PFI ioctl;	Device control and configuration

Description

System API data structure

1.3.2 API_TABLE_T

```
typedef struct API_TABLE_S {
  API_T driver;
```

```
  INT_32 id;
  INT_32 devid;
  INT_32 fd;
  INT_32 opened;
} API_TABLE_T, * PAPI_TABLE_T;
```

File

lpc_api.h (see page 155)

Members

Members	Description
`API_T driver;`	Device driver callbacks
`INT_32 id;`	Device Id
`INT_32 devid;`	Driver device id
`INT_32 fd;`	File descriptor
`INT_32 opened;`	Driver state

Description

Api system device lookup table

1.3.3 BMP_COLOR_TABLE_T

```
typedef struct {
  UNS_8 blue;
  UNS_8 green;
  UNS_8 red;
  UNS_8 unused;
} BMP_COLOR_TABLE_T;
```

File

lpc_bmp.h (see page 161)

Description

Color table entry format (used with BPP1, BPP4, and BPP8)

1.3.4 BMP_STORAGE_T

```
typedef enum {
  INVALID_BMP = -1,
  BPP1 = 0,
  BPP4,
  BPP8,
  BPP24
} BMP_STORAGE_T;
```

File

lpc_bmp.h (see page 161)

Members

Members	Description
`BPP1 = 0`	1 bit per pixel with color table
`BPP4`	4 bits per pixel with color table
`BPP8`	8 bits per pixel with color table
`BPP24`	24 bits per pixel

Description

Supported BMP file formats (no compressed or masked color modes are supported)

1.3.5 BMP_T

```
typedef struct {
  CHAR bftype[2];
  UNS_32 bfsize;
  UNS_32 rsv1;
  UNS_32 dataoffset;
  UNS_32 bisize;
  UNS_32 biwidth;
  UNS_32 biheight;
  UNS_16 biplanes;
  UNS_16 bibitcount;
  UNS_32 bicompressn;
  UNS_32 bisizeimage;
  UNS_32 rsv3;
  UNS_32 rsv4;
  UNS_32 buclrused;
  UNS_32 biclrimp;
  INT_32 ct_data;
} BMP_T;
```

File

lpc_bmp.h (see page 161)

Members

Members	Description
`CHAR bftype[2];`	Always ("BM") for BMP files
`UNS_32 bfsize;`	Size of file in bytes
`UNS_32 rsv1;`	Reserved
`UNS_32 dataoffset;`	Offset from file to start to data
`UNS_32 bisize;`	Size of this structure
`UNS_32 biwidth;`	Pixel width image size
`UNS_32 biheight;`	Pixel height image size
`UNS_16 biplanes;`	color planes
`UNS_16 bibitcount;`	Bits per pixel
`UNS_32 bicompressn;`	Compression type, 0 = BMP
`UNS_32 bisizeimage;`	Size of image in bytes
`UNS_32 rsv3;`	Normally used for metrics
`UNS_32 rsv4;`	Normally used for metrics
`UNS_32 buclrused;`	Colors used in the bitmap
`UNS_32 biclrimp;`	Number of important colors
`INT_32 ct_data;`	Start of color table or data

Description

BMP header structure, not used with files

1.3.6 BMP24_COLOR_TABLE_T

```
typedef struct {
  UNS_8 blue;
  UNS_8 green;
  UNS_8 red;
} BMP24_COLOR_TABLE_T;
```

File

lpc_bmp.h (see page 161)

Description

Color table entry format used with BPP24

1.3.7 BOOL_16

```
typedef INT_16 BOOL_16;
```

File

lpc_types.h (see page 183)

Description

16 bit boolean type

1.3.8 BOOL_32

```
typedef INT_32 BOOL_32;
```

File

lpc_types.h (see page 183)

Description

32 bit boolean type

1.3.9 BOOL_8

```
typedef INT_8 BOOL_8;
```

File

lpc_types.h (see page 183)

Description

8 bit boolean type

1.3.10 CHAR

```
typedef char CHAR;
```

File

lpc_types.h (see page 183)

Description

SMA type for character type

1.3.11 COLOR_T

```
typedef UNS_8 COLOR_T;
```

File

lpc_colors.h (see page 164)

Description

Color type is a 8-bit value

1.3.12 CPAGETABLE_T

```
typedef struct {
  UNS_32 vidx[ARM922T_CPT_ENTRIES];
} CPAGETABLE_T;
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Description

ARM 922T MMU Coarse page table type

1.3.13 DEVICE_FUNCS_TYPE

```
typedef struct {
  ivfunc init_func;
  vvfunc shutdown_func;
  ivfunc insert_ck_func;
  ivfunc ready_ck_func;
  ivfunc busy_ck_func;
  vvfunc start_read_func;
  vvfunc start_write_func;
  ivifunc read_func;
  ivifunc write_func;
} DEVICE_FUNCS_TYPE;
```

File

lpc_fat16.h (see page 166)

Description

This is type DEVICE_FUNCS_TYPE.

1.3.14 FAT_DEVICE_TYPE

```
typedef struct {
  CHAR device[DSIZE];
  INT_8 act_part;
  INT_32 fat_commit;
  PARTITION_TYPE part[4];
  FATGEOM_TYPE pat_hdr;
  FATDATA_TYPE cfat;
```

```
  DEVICE_FUNCS_TYPE func;
  UNS_16 * clusters;
} FAT_DEVICE_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
`CHAR device[DSIZE];`	Name of device
`INT_8 act_part;`	Active partition number (0 - 3), or (-1)
`INT_32 fat_commit;`	FAT commit flag, if set, FAT has changed
`PARTITION_TYPE part[4];`	Information about the 4 partitions
`FATGEOM_TYPE pat_hdr;`	Partition header from selected part.
`FATDATA_TYPE cfat;`	Computed FAT architecture data
`DEVICE_FUNCS_TYPE func;`	Pointer to device driver functions
`UNS_16 * clusters;`	Cached cluster table

Description

FAT device structure, used to bind a device driver to the FAT driver

1.3.15 FATDATA_TYPE

```
typedef struct {
  UNS_32 first_boot_sector;
  UNS_32 boot_sectors;
  UNS_32 fat_sectors;
  UNS_32 first_fat1_sector;
  UNS_32 last_fat1_sector;
  UNS_32 first_fat2_sector;
  UNS_32 last_fat2_sector;
  UNS_32 first_root_sector;
  UNS_32 root_sectors;
  UNS_32 first_data_sector;
  UNS_32 total_sectors;
  UNS_32 data_sectors;
  UNS_32 clusters;
  UNS_32 total_size;
  UNS_16 cluster_size;
} FATDATA_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
`UNS_32 first_boot_sector;`	First boot sector
`UNS_32 boot_sectors;`	Total boot sectors
`UNS_32 fat_sectors;`	FAT sectors (single FAT)
`UNS_32 first_fat1_sector;`	First FAT1 sector
`UNS_32 last_fat1_sector;`	Last FAT1 sector
`UNS_32 first_fat2_sector;`	First FAT2 sector
`UNS_32 last_fat2_sector;`	Last FAT2 sector
`UNS_32 first_root_sector;`	First root sector
`UNS_32 root_sectors;`	Total root sectors
`UNS_32 first_data_sector;`	First data sector
`UNS_32 total_sectors;`	Total sectors on device
`UNS_32 data_sectors;`	Total data sectors
`UNS_32 clusters;`	Total number of clusters
`UNS_32 total_size;`	Total size of device in bytes

`UNS_16 cluster_size;`	Cluster size in bytes

Description

The following structure holds computed information about the device

1.3.16 FATGEOM_TYPE

```
typedef struct {
  UNS_8 jump[3];
  UNS_8 oem_id[8];
  UNS_16 bytes_sector;
  UNS_8 sectors_cluster;
  UNS_16 res_sectors;
  UNS_8 fat_copies;
  UNS_16 root_entries;
  UNS_16 small_sectors;
  UNS_8 media_desc;
  UNS_16 sectors_fat;
  UNS_16 sectors_track;
  UNS_16 number_heads;
  UNS_32 hidden_sectors;
  UNS_32 large_sectors;
  UNS_8 drive_number;
  UNS_8 reserved;
  UNS_8 ext_boot_sig;
  UNS_32 serial_number;
  CHAR label[11];
  CHAR fs_name[8];
} FATGEOM_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
`UNS_8 jump[3];`	Boot code jump point
`UNS_8 oem_id[8];`	Name of formatting OS
`UNS_16 bytes_sector;`	Bytes per sector
`UNS_8 sectors_cluster;`	Sectors per cluster
`UNS_16 res_sectors;`	Reserved sectors from start
`UNS_8 fat_copies;`	Number of FAT copies
`UNS_16 root_entries;`	Number of root entries
`UNS_16 small_sectors;`	Small number of sectors
`UNS_8 media_desc;`	Media descriptor
`UNS_16 sectors_fat;`	Sectors per FAT
`UNS_16 sectors_track;`	Sectors per track
`UNS_16 number_heads;`	Number of heads
`UNS_32 hidden_sectors;`	Number of hidden sectors
`UNS_32 large_sectors;`	Large number of sectors
`UNS_8 drive_number;`	Drive number
`UNS_8 ext_boot_sig;`	Extended boot signature
`UNS_32 serial_number;`	Volume serial number
`CHAR label[11];`	Volume label
`CHAR fs_name[8];`	File system name (FAT16)

Description

Drive geometry structure for partition, filled in by the driver. (Not everything in this sector is saved)

1.3.17 FILE_MODE_TYPE

```
typedef enum {
  FINVALID,
  FREAD,
  FWRITE
} FILE_MODE_TYPE;
```

File

lpc_fat16.h (see page 166)

Description

File modes

1.3.18 FILE_TYPE

```
typedef struct {
  FILE_MODE_TYPE fmode;
  INT_32 dir_commit;
  UNS_16 clusternum;
  UNS_32 filesize;
  INT_32 file_dir_entry;
  FAT_DEVICE_TYPE * fat_data;
  UNS_32 sector_dir;
  UNS_8 * data;
  UNS_32 buf_index;
  ROOT_ENTRY_TYPE * dir_data;
  INT_32 dir_index;
} FILE_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
`FILE_MODE_TYPE fmode;`	File operational mode
`INT_32 dir_commit;`	DIR commit flag, if set, DIR has changed
`UNS_16 clusternum;`	Present working cluster number
`UNS_32 filesize;`	File size in bytes
`INT_32 file_dir_entry;`	Active file working entry (read/write)
`FAT_DEVICE_TYPE * fat_data;`	Pointer to binded FAT structure
`UNS_32 sector_dir;`	Sector number of start of active dir
`UNS_8 * data;`	Pointer to allocated data buffer
`UNS_32 buf_index;`	Buffer read/write index
`ROOT_ENTRY_TYPE * dir_data;`	Cached active directory structure
`INT_32 dir_index;`	Directory entry lookup index

Description

File descriptor

1.3.19 FONT_T

```
typedef struct {
  INT_16 font_height;
```

```
  UNS_8 first_char;
  UNS_8 last_char;
  UNS_16 * font_table;
  UNS_8 * font_width_table;
} FONT_T;
```

File

lpc_fonts.h (see page 171)

Description

Font data structure

1.3.20 FPAGETABLE_T

```
typedef struct {
  UNS_32 vidx[ARM922T_FPT_ENTRIES];
} FPAGETABLE_T;
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Description

ARM 922T MMU Fine page table type

1.3.21 HEAP_DESCRIPTOR_T

```
typedef struct {
  UNS_32 entry_size;
  void * next_descriptor;
  void * prev_descriptor;
} HEAP_DESCRIPTOR_T;
```

File

lpc_heap.c (see page 172)

Members

Members	Description
`UNS_32 entry_size;`	Size of this heap entry including the descriptor (0 = used)
`void * next_descriptor;`	Pointer to next descriptor (0 = last)
`void * prev_descriptor;`	Pointer to previous descriptor (heap_base (see page 76) = no previous entry)

Description

Heap descriptor

1.3.22 INT_16

```
typedef signed short INT_16;
```

File

lpc_types.h (see page 183)

Description

SMA type for 16 bit signed value

1.3.23 INT_32

```
typedef signed int INT_32;
```

File

lpc_types.h (see page 183)

Description

SMA type for 32 bit signed value

1.3.24 INT_64

```
typedef long long INT_64;
```

File

lpc_types.h (see page 183)

Description

SMA type for 64 bit signed value

1.3.25 INT_8

```
typedef signed char INT_8;
```

File

lpc_types.h (see page 183)

Description

SMA type for 8 bit signed value

1.3.26 ivfunc

```
typedef INT_32 (* ivfunc)(void);
```

File

lpc_fat16.h (see page 166)

Description

This is type ivfunc.

1.3.27 ivifunc

```
typedef INT_32 (* ivifunc)(void *, INT_32);
```

File

lpc_fat16.h (see page 166)

Description

This is type ivifunc.

1.3.28 LCD_PANEL_T

```
typedef enum {
  TFT = 0,
  ADTFT,
  HRTFT,
  MONO_4BIT,
  MONO_8BIT,
  CSTN
} LCD_PANEL_T;
```

File

lpc_lcd_params.h (see page 175)

Members

Members	Description
TFT = 0	Panel type is standard TFT
ADTFT	Panel type is advanced TFT
HRTFT	Panel type is highly reflective TFT
MONO_4BIT	Panel type is 4-bit mono
MONO_8BIT	Panel type is 8-bit mono
CSTN	Panel type is color STN

Description

LCD display types

1.3.29 LCD_PARAM_T

```
typedef struct {
  UNS_8 h_back_porch;
  UNS_8 h_front_porch;
  UNS_8 h_sync_pulse_width;
  UNS_16 pixels_per_line;
  UNS_8 v_back_porch;
  UNS_8 v_front_porch;
  UNS_8 v_sync_pulse_width;
  UNS_16 lines_per_panel;
  UNS_8 invert_output_enable;
  UNS_8 invert_panel_clock;
  UNS_8 invert_hsync;
  UNS_8 invert_vsync;
  UNS_8 ac_bias_frequency;
  UNS_8 bits_per_pixel;
  UNS_32 optimal_clock;
  LCD_PANEL_T lcd_panel_type;
  UNS_8 dual_panel;
  UNS_8 hrtft_cls_enable;
  UNS_8 hrtft_sps_enable;
  UNS_8 hrtft_lp_to_ps_delay;
  UNS_8 hrtft_polarity_delay;
  UNS_8 hrtft_lp_delay;
  UNS_8 hrtft_spl_delay;
```

```
  UNS_16 hrtft_spl_to_cls_delay;
} LCD_PARAM_T;
```

File

lpc_lcd_params.h (see page 175)

Members

Members	Description
UNS_8 h_back_porch;	Horizontal back porch in clocks (minimum of 1)
UNS_8 h_front_porch;	Horizontal front porch in clocks (minimum of 1)
UNS_8 h_sync_pulse_width;	HSYNC pulse width in clocks (minimum of 1)
UNS_16 pixels_per_line;	Pixels per line (horizontal resolution)
UNS_8 v_back_porch;	Vertical back porch in clocks
UNS_8 v_front_porch;	Vertical front porch in clocks
UNS_8 v_sync_pulse_width;	VSYNC pulse width in clocks (minimum 1 clock)
UNS_16 lines_per_panel;	Lines per panel (vertical resolution)
UNS_8 invert_output_enable;	Invert output enable, 1 = invert
UNS_8 invert_panel_clock;	Invert panel clock, 1 = invert
UNS_8 invert_hsync;	Invert HSYNC, 1 = invert
UNS_8 invert_vsync;	Invert VSYNC, 1 = invert
UNS_8 ac_bias_frequency;	AC bias frequency in clocks (minimum 1)
UNS_8 bits_per_pixel;	Maximum bits per pixel the display supports
UNS_32 optimal_clock;	Optimal clock rate (Hz)
LCD_PANEL_T lcd_panel_type;	LCD panel type
UNS_8 dual_panel;	Dual panel, 1 = dual panel display
UNS_8 hrtft_cls_enable;	HRTFT CLS enable flag, 1 = enable
UNS_8 hrtft_sps_enable;	HRTFT SPS enable flag, 1 = enable
UNS_8 hrtft_lp_to_ps_delay;	HRTFT LP to PS delay in clocks
UNS_8 hrtft_polarity_delay;	HRTFT polarity delay in clocks
UNS_8 hrtft_lp_delay;	HRTFT LP delay in clocks
UNS_8 hrtft_spl_delay;	HRTFT SPL delay in clocks HRTFT SPL to CLKS delay

Description

Structure containing the parameters for the LCD panel

1.3.30 PAPI_T

```
typedef struct API_S {
  PFI open;
  PFI close;
  PFI read;
  PFI write;
  PFI ioctl;
} API_T, * PAPI_T;
```

File

lpc_api.h (see page 155)

Members

Members	Description
PFI open;	Open the device
PFI close;	Close the device
PFI read;	Read data from the device
PFI write;	Wrote data to the device
PFI ioctl;	Device control and configuration

Description

System API data structure

1.3.31 PAPI_TABLE_T

```
typedef struct API_TABLE_S {
  API_T driver;
  INT_32 id;
  INT_32 devid;
  INT_32 fd;
  INT_32 opened;
} API_TABLE_T, * PAPI_TABLE_T;
```

File

lpc_api.h (see page 155)

Members

Members	Description
`API_T driver;`	Device driver callbacks
`INT_32 id;`	Device Id
`INT_32 devid;`	Driver device id
`INT_32 fd;`	File descriptor
`INT_32 opened;`	Driver state

Description

Api system device lookup table

1.3.32 PARTITION_TYPE

```
typedef struct {
  UNS_8 state;
  UNS_8 head_start;
  UNS_16 cyl_sec_start;
  UNS_8 partype;
  UNS_8 head_end;
  UNS_16 cyl_sec_end;
  UNS_32 mbr_sec_offset;
  UNS_32 partsecs;
} PARTITION_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
`UNS_8 state;`	State of the partition
`UNS_8 head_start;`	Sector start head
`UNS_16 cyl_sec_start;`	Partition cylinder/sector start
`UNS_8 partype;`	Partition type
`UNS_8 head_end;`	Sector start end
`UNS_16 cyl_sec_end;`	Partition cylinder/sector end
`UNS_32 mbr_sec_offset;`	Offset from MBR to start of part.
`UNS_32 partsecs;`	Number of sectors in the partition

Description

Partition entries

1.3.33 PFI

```
typedef INT_32 (* PFI)();
```

File

lpc_types.h (see page 183)

Description

Pointer to Function returning INT_32 (see page 66) (any number of parameters)

1.3.34 PFV

```
typedef void (* PFV)();
```

File

lpc_types.h (see page 183)

Description

Pointer to Function returning Void (any number of parameters)

1.3.35 ROOT_ENTRY_TYPE

```
typedef struct {
  CHAR name[8];
  CHAR ext[3];
  UNS_8 attribute;
  UNS_8 reserved1;
  UNS_8 createtimems;
  UNS_16 createtime;
  UNS_16 createdate;
  UNS_16 accessdate;
  UNS_16 clusterhi;
  UNS_16 updatetime;
  UNS_16 updatedate;
  UNS_16 clusternum;
  UNS_32 filesize;
} ROOT_ENTRY_TYPE;
```

File

lpc_fat16.h (see page 166)

Members

Members	Description
CHAR name[8];	Left space padded name
CHAR ext[3];	Left space padded extension
UNS_8 attribute;	File attribute
UNS_8 createtimems;	timestamp in 10mS
UNS_16 createtime;	timestamp, time
UNS_16 createdate;	timestamp, date
UNS_16 accessdate;	Last date of access
UNS_16 clusterhi;	High cluster (FAT32 only)
UNS_16 updatetime;	Last time of change
UNS_16 updatedate;	Last date of change

UNS_16 clusternum;	Cluster link number
UNS_32 filesize;	Size of file in bytes

Description

Initialization functions

****

Directory structure root entry stored on device

1.3.36 STATUS

```
typedef INT_32 STATUS;
```

File

lpc_types.h (see page 183)

Description

Status type

1.3.37 SWIM_ROTATION_T

```
typedef enum {
  NOROTATION,
  RIGHT,
  INVERT,
  LEFT
} SWIM_ROTATION_T;
```

File

lpc_swim_image.h (see page 182)

Description

Image rotation tags

1.3.38 SWIM_WINDOW_T

```
typedef struct {
  INT_32 xpsize;
  INT_32 ypsize;
  INT_32 xpmin;
  INT_32 ypmin;
  INT_32 xpmax;
  INT_32 ypmax;
  INT_32 bdsize;
  INT_32 xvsize;
  INT_32 yvsize;
  INT_32 xpvmin;
  INT_32 ypvmin;
  INT_32 xpvmax;
  INT_32 ypvmax;
  INT_32 xvpos;
  INT_32 yvpos;
  COLOR_T pen;
  COLOR_T bkg;
```

```
  COLOR_T fill;
  FONT_T * font;
  INT_32 tfont;
  COLOR_T * fb;
  INT_32 winused;
  BOOL_32 tfonts;
} SWIM_WINDOW_T;
```

File

lpc_swim.h (see page 178)

Members

Members	Description
INT_32 xpsize;	Physical (absolute) horizontal screen size
INT_32 ypsize;	Physical (absolute) vertical screen size
INT_32 xpmin;	Physical left edge of window
INT_32 ypmin;	Physical top edge of window
INT_32 xpmax;	Physical right edge of window
INT_32 ypmax;	Physical bottom edge of window
INT_32 bdsize;	Size of window frame in pixels
INT_32 xvsize;	Virtual horizontal window size
INT_32 yvsize;	Virtual vertical window size
INT_32 xpvmin;	Physical left edge of draw window
INT_32 ypvmin;	Physical top edge of draw window
INT_32 xpvmax;	Physical right edge of draw window
INT_32 ypvmax;	Physical bottom edge of draw window
INT_32 xvpos;	Next virtual 'x' position of output
INT_32 yvpos;	Next virtual 'y' position of output
COLOR_T pen;	Pen/text color
COLOR_T bkg;	Window/text background color
COLOR_T fill;	Fill/border color
FONT_T * font;	Selected font structure
INT_32 tfont;	Transparent font background flag when true
COLOR_T * fb;	Frame buffer address for the physical display
INT_32 winused;	Window used flag
BOOL_32 tfonts;	Transparent font background flag

Description

Structure is used to store information about a specific window

1.3.39 TRANSTABLE_T

```
typedef struct {
  UNS_32 vidx[ARM922T_TT_ENTRIES];
} TRANSTABLE_T;
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Description

ARM 922T MMU Translation table structure

1.3.40 TT_SECTION_BLOCK_T

```
typedef struct {
  UNS_32 num_sections;
```

```
  UNS_32 virt_addr;
  UNS_32 phys_addr;
  UNS_32 entry;
} TT_SECTION_BLOCK_T;
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Members

Members	Description
`UNS_32 num_sections;`	Number of 1MByte sections
`UNS_32 virt_addr;`	Virtual address of section
`UNS_32 phys_addr;`	Physical address of section Section attributes - an 'OR'ed combination of ARM922T_L1D_AP_x, ARM922T_L1D_DOMAIN (see page 104), ARM922T_L1D_CACHEABLE (see page 103), ARM922T_L1D_BUFFERABLE (see page 103), and ARM922T_L1D_TYPE_x

Description

UNS_32 (see page 73) num_sections: number of 1MByte sections >=1 for all blocks except last; last = 0 UNS_32 (see page 73) virt_addr: as required, base Virtual address for block UNS_32 (see page 73) phys_addr: as required, PT address or Section address UNS_32 (see page 73) entry is composed of the following 'or'd' together: access_perm: ARM922T_L1D_AP_x (x = SVC_ONLY, USR_RO, ALL) domain: ARM922T_L1D_DOMAIN (see page 104)(n) as applicable cacheable: ARM922T_L1D_CACHEABLE (see page 103) if applicable write_buffered: ARM922T_L1D_BUFFERABLE (see page 103) if applicable descriptor_type: ARM922T_L1D_TYPE_x (x = FAULT, PAGE, SECTION)

1.3.41 UNS_16

```
typedef unsigned short UNS_16;
```

File

lpc_types.h (see page 183)

Description

SMA type for 16 bit unsigned value

1.3.42 UNS_32

```
typedef unsigned int UNS_32;
```

File

lpc_types.h (see page 183)

Description

SMA type for 32 bit unsigned value

1.3.43 UNS_64

```
typedef unsigned long long UNS_64;
```

File

lpc_types.h (see page 183)

Description

SMA type for 64 bit unsigned value

1.3.44 UNS_8

```
typedef unsigned char UNS_8;
```

File

lpc_types.h (see page 183)

Description

SMA type for 8 bit unsigned value

1.3.45 vvfunc

```
typedef void (* vvfunc)(void);
```

File

lpc_fat16.h (see page 166)

Description

Device function list

1.4 Variables

1.4.1 api

```
STATIC API_TABLE_T api[MAX_API_TABLE];
```

File

lpc_api.c (see page 154)

Description

Private io system table

1.4.2 api_is_init

```
STATIC INT_32 api_is_init = FALSE;
```

File

lpc_api.c (see page 154)

Description

State variable for init

1.4.3 font_helvr10

```
const FONT_T font_helvr10;
```

File

lpc_helvr10.c (see page 174)

Description

Externally available font information structure

1.4.4 font_rom8x16

```
const FONT_T font_rom8x16;
```

File

lpc_rom8x16.c (see page 176)

Description

Externally available font information structure

1.4.5 font_rom8x8

```
const FONT_T font_rom8x8;
```

File

lpc_rom8x8.c (see page 177)

Description

Externally available font information structure

1.4.6 font_winfreesys14x16

```
const FONT_T font_winfreesys14x16;
```

File

lpc_winfreesystem14x16.c (see page 185)

Description

Externally available font information structure

1.4.7 font_x5x7

```
const FONT_T font_x5x7;
```

File

lpc_x5x7.c (see page 186)

Description

Externally available font information structure

1.4.8 font_x6x13

```
const FONT_T font_x6x13;
```

File

lpc_x6x13.c (see page 187)

Description

Externally available font information structure

1.4.9 heap_base

```
HEAP_DESCRIPTOR_T * heap_base;
```

File

lpc_heap.c (see page 172)

Description

Heap base address

1.4.10 heap_size_saved

```
UNS_32 heap_size_saved;
```

File

lpc_heap.c (see page 172)

Description

Heap size

1.4.11 helvr10_bits

```
static UNS_16 helvr10_bits[] = { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0x4000, 0x4000,
0x4000, 0x4000, 0x0000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5000, 0x5000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2800,
```

```
0x2800, 0x7c00, 0x2800, 0xf800, 0x5000, 0x5000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000,
0x7000, 0xa800, 0xa000, 0x7000, 0x2800, 0xa800, 0x7000, 0x2000, 0x0000, 0x0000, 0x0000,
0x6400, 0x9400, 0x6800, 0x0800, 0x1000, 0x1600, 0x2900, 0x2600, 0x0000, 0x0000, 0x0000,
0x0000, 0x1000, 0x2800, 0x2800, 0x3000, 0x5200, 0x4c00, 0x4c00, 0x3200, 0x0000, 0x0000,
0x0000, 0x0000, 0x2000, 0x2000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2000, 0x4000, 0x4000, 0x8000, 0x8000, 0x8000, 0x8000, 0x4000,
0x4000, 0x2000, 0x0000, 0x0000, 0x4000, 0x2000, 0x2000, 0x1000, 0x1000, 0x1000, 0x1000,
0x2000, 0x2000, 0x4000, 0x0000, 0x0000, 0xa000, 0x4000, 0xa000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x2000, 0xf800,
0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000,
0x2000, 0x4000, 0x4000, 0x4000, 0x4000, 0x8000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7000, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2000, 0x6000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7000, 0x8800, 0x0800, 0x0800, 0x3000, 0x4000, 0x8000, 0xf800, 0x0000,
0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x0800, 0x3000, 0x0800, 0x0800, 0x8800, 0x7000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x3000, 0x5000, 0x5000, 0x9000, 0xf800, 0x1000,
0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x8000, 0x8000, 0xf000, 0x0800, 0x0800,
0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8000, 0xb000, 0xc800,
0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0800, 0x1000, 0x1000,
0x2000, 0x2000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8800,
0x7000, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800,
0x8800, 0x9800, 0x6800, 0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1000, 0x2000, 0x4000, 0x2000, 0x1000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf000, 0x0000, 0xf000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x2000, 0x1000, 0x2000, 0x4000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x4800, 0x0800, 0x1000, 0x2000, 0x2000, 0x0000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1f00, 0x2080, 0x4d40, 0x9240, 0xa240, 0xa480,
0xa480, 0x9b00, 0x4000, 0x3e00, 0x0000, 0x0000, 0x1000, 0x1000, 0x2800, 0x2800, 0x4400,
0x7c00, 0x8200, 0x8200, 0x0000, 0x0000, 0x0000, 0x0000, 0x7800, 0x4400, 0x4400, 0x7800,
0x4400, 0x4400, 0x4400, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x4200, 0x4000,
0x4000, 0x4000, 0x4000, 0x4200, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x7800, 0x4400,
0x4200, 0x4200, 0x4200, 0x4200, 0x4400, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00,
0x4000, 0x4000, 0x7c00, 0x4000, 0x4000, 0x4000, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x7c00, 0x4000, 0x4000, 0x7800, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3c00, 0x4200, 0x4000, 0x4000, 0x4600, 0x4200, 0x4600, 0x3a00, 0x0000, 0x0000,
0x0000, 0x0000, 0x4200, 0x4200, 0x4200, 0x7e00, 0x4200, 0x4200, 0x4200, 0x4200, 0x0000,
0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x1000, 0x1000, 0x1000, 0x1000, 0x1000, 0x9000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4400, 0x4800, 0x5000, 0x7000, 0x4800, 0x4800,
0x4400, 0x4400, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000,
0x4000, 0x4000, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x4100, 0x6300, 0x6300, 0x5500,
0x5500, 0x4900, 0x4900, 0x4900, 0x0000, 0x0000, 0x0000, 0x0000, 0x6200, 0x6200, 0x5200,
0x5200, 0x4a00, 0x4a00, 0x4600, 0x4600, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x4200,
0x4200, 0x4200, 0x4200, 0x4200, 0x4200, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x7800,
0x4400, 0x4400, 0x7800, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3c00, 0x4200, 0x4200, 0x4200, 0x4200, 0x4a00, 0x4600, 0x3e00, 0x0100, 0x0000, 0x0000,
0x0000, 0x7800, 0x4400, 0x4400, 0x7800, 0x4400, 0x4400, 0x4400, 0x4400, 0x0000, 0x0000,
0x0000, 0x0000, 0x3800, 0x4400, 0x4000, 0x3800, 0x0400, 0x4400, 0x4400, 0x3800, 0x0000,
0x0000, 0x0000, 0x0000, 0xf800, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x0000, 0x0000, 0x0000, 0x0000, 0x4200, 0x4200, 0x4200, 0x4200, 0x4200, 0x4200, 0x4200,
0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x8200, 0x8200, 0x4400, 0x4400, 0x4400, 0x2800,
0x2800, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8880, 0x8880, 0x4900, 0x4900, 0x5500,
0x2200, 0x2200, 0x2200, 0x0000, 0x0000, 0x0000, 0x0000, 0x4400, 0x4400, 0x2800, 0x1000,
0x2800, 0x2800, 0x4400, 0x4400, 0x0000, 0x0000, 0x0000, 0x0000, 0x8200, 0x4400, 0x4400,
0x2800, 0x2800, 0x1000, 0x1000, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0x0400,
0x0800, 0x1000, 0x1000, 0x2000, 0x4000, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x6000, 0x0000, 0x0000,
0x8000, 0x8000, 0x4000, 0x4000, 0x4000, 0x4000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000,
0x0000, 0xc000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0xc000,
0x0000, 0x0000, 0x2000, 0x2000, 0x5000, 0x5000, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xfc00, 0x0000, 0x0000, 0x4000, 0x8000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xe000, 0x1000, 0x7000, 0x9000,
0x9000, 0x6800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0xb000, 0xc800, 0x8800,
0x8800, 0xc800, 0xb000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x9000,
0x8000, 0x8000, 0x9000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0800, 0x0800, 0x6800,
0x9800, 0x8800, 0x8800, 0x9800, 0x6800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
```

```
0x6000, 0x9000, 0xf000, 0x8000, 0x9000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000,
0x4000, 0xe000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6800, 0x9800, 0x8800, 0x8800, 0x9800, 0x6800, 0x0800, 0x7000, 0x0000,
0x0000, 0x8000, 0x8000, 0xb000, 0xc800, 0x8800, 0x8800, 0x8800, 0x8800, 0x0000, 0x0000,
0x0000, 0x0000, 0x8000, 0x0000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x8000, 0x0000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000,
0x8000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0x9000, 0xa000, 0xc000, 0xa000, 0x9000,
0x9000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000,
0x8000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xec00, 0x9200, 0x9200,
0x9200, 0x9200, 0x9200, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xb000, 0xc800,
0x8800, 0x8800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000,
0x8800, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xb000, 0xc800, 0x8800, 0x8800, 0xc800, 0xb000, 0x8000, 0x8000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6800, 0x9800, 0x8800, 0x8800, 0x9800, 0x6800, 0x0800, 0x0800, 0x0000, 0x0000,
0x0000, 0x0000, 0xa000, 0xc000, 0x8000, 0x8000, 0x8000, 0x8000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x9000, 0x6000, 0x1000, 0x9000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x4000, 0x4000, 0xe000, 0x4000, 0x4000, 0x4000, 0x4000, 0x6000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x9000, 0x9000, 0x9000, 0x9000, 0x9000, 0x7000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x5000, 0x5000, 0x2000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x9200, 0x9200, 0x5400, 0x5400,
0x2800, 0x2800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x5000, 0x2000,
0x5000, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x9000, 0x9000,
0xa000, 0xa000, 0x6000, 0x4000, 0x4000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf000,
0x1000, 0x2000, 0x4000, 0x8000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x4000,
0x4000, 0x4000, 0x8000, 0x4000, 0x4000, 0x4000, 0x4000, 0x2000, 0x0000, 0x0000, 0x4000,
0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x0000,
0x8000, 0x4000, 0x4000, 0x4000, 0x2000, 0x4000, 0x4000, 0x4000, 0x4000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6400, 0x9800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, };
```

File

lpc_helvr10.c (see page 174)

Description

Font character bitmap data.

1.4.12 helvR10_width

```
static UNS_8 helvR10_width[] = { 3, 3, 4, 6, 6, 9, 8, 3, 4, 4, 4, 6, 3, 7, 3, 3, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 3, 3, 6, 5, 6, 6, 11, 7, 7, 8, 8, 7, 6, 8, 8, 3, 5, 7, 6, 9, 8, 8, 7,
8, 7, 7, 5, 8, 7, 9, 7, 7, 7, 3, 3, 3, 6, 6, 3, 5, 6, 5, 6, 5, 4, 6, 6, 2, 2, 5, 2, 8, 6,
6, 6, 6, 4, 5, 4, 5, 6, 8, 6, 5, 5, 3, 3, 3, 7, };
```

File

lpc_helvr10.c (see page 174)

Description

Character width data.

1.4.13 rom8x16_bits

```
static UNS_16 rom8x16_bits[] = { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7e00, 0x8100, 0xa500, 0x8100, 0x8100, 0xbd00, 0x9900, 0x8100, 0x8100, 0x7e00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xfe00, 0xfe00, 0xd600, 0xfe00, 0xfe00,
0xba00, 0xc600, 0xfe00, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6c00, 0xee00, 0xfe00, 0xfe00, 0xfe00, 0xfe00, 0x7c00, 0x3800, 0x1000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x3800, 0x7c00, 0xfe00, 0x7c00, 0x3800,
0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000,
0x3800, 0x3800, 0x1000, 0x6c00, 0xee00, 0x6c00, 0x1000, 0x3800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1000, 0x3800, 0x7c00, 0x7c00, 0xfe00, 0xfe00, 0xfe00, 0x6c00,
0x1000, 0x3800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
```

```
0x1800, 0x3c00, 0x3c00, 0x3c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xe700, 0xc300, 0xc300, 0xc300, 0xe700, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00,
0x6600, 0x6600, 0x6600, 0x3c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00,
0xff00, 0xff00, 0xff00, 0xe700, 0xc300, 0x9900, 0x9900, 0x9900, 0xc300, 0xe700, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0x0000, 0x0000, 0x1e00, 0x0e00, 0x1e00, 0x3600, 0x7800,
0xcc00, 0xcc00, 0xcc00, 0xcc00, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3c00, 0x6600, 0x6600, 0x6600, 0x3c00, 0x1800, 0x7e00, 0x1800, 0x1800, 0x1800, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1e00, 0x1a00, 0x1e00, 0x1800, 0x1800, 0x1800,
0x1800, 0x7800, 0xf800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3e00,
0x3600, 0x3e00, 0x3600, 0x3600, 0x7600, 0xf600, 0x6600, 0x0e00, 0x1e00, 0x0c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0xdb00, 0x7e00, 0x3c00, 0x6600, 0x6600, 0x3c00,
0x7e00, 0xdb00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000,
0xe000, 0xf000, 0xfc00, 0xfe00, 0xfc00, 0xf000, 0xe000, 0x8000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0200, 0x0e00, 0x3e00, 0x7e00, 0xfe00, 0x7e00, 0x3e00,
0x0e00, 0x0200, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x7e00,
0x1800, 0x1800, 0x1800, 0x1800, 0x7e00, 0x3c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x0000, 0x6600,
0x6600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7f00, 0xdb00, 0xdb00, 0xdb00,
0xdb00, 0x7b00, 0x1b00, 0x1b00, 0x1b00, 0x1b00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc600, 0x6000, 0x7c00, 0xf600, 0xde00, 0x7c00, 0x0c00, 0xc600,
0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xfe00, 0xfe00, 0xfe00, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1800, 0x3c00, 0x7e00, 0x1800, 0x1800, 0x1800, 0x7e00, 0x3c00, 0x1800, 0x7e00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x7e00, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x7e00, 0x3c00, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x0e00, 0xff00, 0x0e00,
0x0c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3000, 0x7000, 0xfe00, 0x7000, 0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc000, 0xc000, 0xc000, 0xfe00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x2400, 0x6600, 0xff00, 0x6600, 0x2400, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1000, 0x3800, 0x3800, 0x3800, 0x7c00, 0x7c00, 0xfe00, 0xfe00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0xfe00, 0x7c00,
0x7c00, 0x7c00, 0x3800, 0x3800, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x3c00, 0x3c00, 0x3c00,
0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3600,
0x3600, 0x3600, 0x3600, 0x1400, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0xfe00, 0x6c00, 0x6c00,
0xfe00, 0x6c00, 0x6c00, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x1800, 0x7c00, 0xc600, 0xc000, 0x7800, 0x3c00, 0x0600, 0xc600, 0x7c00, 0x1800, 0x1800,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6200, 0x6600, 0x0c00, 0x1800,
0x3000, 0x6600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00,
0x3800, 0x3000, 0x7600, 0x7e00, 0xcc00, 0xcc00, 0xcc00, 0x7600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0c00, 0x0c00, 0x0c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1800, 0x3000,
0x3000, 0x3000, 0x3000, 0x3000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3000, 0x1800, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x1800,
0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6c00,
0x3800, 0xfe00, 0x3800, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x7e00, 0x1800, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0c00, 0x0c00, 0x0c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0200, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000, 0xc000, 0x8000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xce00, 0xde00, 0xf600, 0xe600,
0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x7800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x7e00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000,
0xc600, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0x0600,
0x0600, 0x3c00, 0x0600, 0x0600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0c00, 0x1c00, 0x3c00, 0x6c00, 0xcc00, 0xcc00, 0xfe00, 0x0c00, 0x0c00,
0x1e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0xc000, 0xc000, 0xc000,
0xfc00, 0x0600, 0x0600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc000, 0xc000, 0xfc00, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0xc600, 0x0600, 0x0c00, 0x1800,
0x3000, 0x3000, 0x3000, 0x3000, 0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7c00, 0xc600, 0xc600, 0xc600, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000,
```

```
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0x7e00,
0x0600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0c00, 0x0c00, 0x0000, 0x0000, 0x0c00, 0x0c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x0c00, 0x0000, 0x0000,
0x0c00, 0x0c00, 0x0c00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00,
0x1800, 0x3000, 0x6000, 0xc000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0000, 0xfe00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x3000,
0x1800, 0x0c00, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0x0c00, 0x1800, 0x1800, 0x1800, 0x0000, 0x1800,
0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600,
0xde00, 0xde00, 0xde00, 0xdc00, 0xc000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3800, 0x6c00, 0xc600, 0xc600, 0xc600, 0xfe00, 0xc600, 0xc600, 0xc600, 0xc600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfc00, 0x6600, 0x6600, 0x6600, 0x7c00,
0x6600, 0x6600, 0x6600, 0x6600, 0xfc00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3c00, 0x6600, 0xc200, 0xc000, 0xc000, 0xc000, 0xc000, 0xc200, 0x6600, 0x3c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x6c00, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x6600, 0x6c00, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00,
0x6600, 0x6000, 0x6400, 0x7c00, 0x6400, 0x6000, 0x6000, 0x6600, 0xfe00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x6600, 0x6000, 0x6400, 0x7c00, 0x6400, 0x6000,
0x6000, 0x6000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600,
0xc600, 0xc000, 0xc000, 0xc000, 0xce00, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0xfe00, 0xc600, 0xc600, 0xc600,
0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3c00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xd800, 0xd800,
0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xcc00, 0xd800,
0xf000, 0xf000, 0xd800, 0xcc00, 0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6200, 0x6600, 0xfe00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xee00, 0xee00, 0xfe00,
0xd600, 0xd600, 0xd600, 0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xc600, 0xc600, 0xe600, 0xe600, 0xf600, 0xde00, 0xce00, 0xce00, 0xc600, 0xc600, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600,
0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfc00,
0x6600, 0x6600, 0x6600, 0x6600, 0x7c00, 0x6000, 0x6000, 0x6000, 0xf000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600,
0xd600, 0xd600, 0x7c00, 0x0600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfc00, 0x6600,
0x6600, 0x6600, 0x7c00, 0x7800, 0x6c00, 0x6600, 0x6600, 0xe600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc000, 0xc000, 0x7000, 0x1c00, 0x0600, 0x0600,
0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00, 0x5a00, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600,
0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600,
0xc600, 0xc600, 0xc600, 0x6c00, 0x3800, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xc600, 0xc600, 0xc600, 0xd600, 0xd600, 0xd600, 0xfe00, 0xee00, 0xc600, 0xc600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0x6c00, 0x3800,
0x3800, 0x6c00, 0xc600, 0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0xc600, 0x8600, 0x0c00, 0x1800, 0x3000,
0x6000, 0xc200, 0xc600, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x7c00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0xc000, 0x6000, 0x3000, 0x1800,
0x0c00, 0x0600, 0x0200, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0x0c00,
0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x7c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1000, 0x3800, 0x6c00, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x0000, 0x0000,
0x0000, 0x1800, 0x1800, 0x1800, 0x0c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7800,
0x0c00, 0x7c00, 0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xe000, 0x6000, 0x6000, 0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0xfc00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600,
0xc000, 0xc000, 0xc000, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1c00, 0x0c00, 0x0c00, 0x7c00, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0x7e00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600,
0xfe00, 0xc000, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1c00,
0x3600, 0x3000, 0x3000, 0xfc00, 0x3000, 0x3000, 0x3000, 0x3000, 0x7800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7600, 0xce00, 0xc600, 0xc600,
0xce00, 0x7600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0xe000, 0x6000,
0x6000, 0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0xe600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x3800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x0c00, 0x0000,
```

```
0x1c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0xcc00, 0xcc00, 0x7800, 0x0000, 0x0000,
0x0000, 0x0000, 0xe000, 0x6000, 0x6000, 0x6600, 0x6600, 0x6c00, 0x7800, 0x6c00, 0x6600,
0xe600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6c00, 0xfe00, 0xd600, 0xd600, 0xc600, 0xc600, 0xc600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xdc00, 0x6600,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xdc00, 0x6600, 0x6600,
0x6600, 0x6600, 0x7c00, 0x6000, 0x6000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7600, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0x7c00, 0x0c00, 0x0c00, 0x1e00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xdc00, 0x6600, 0x6000, 0x6000,
0x6000, 0x6000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc000, 0x7c00, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3000, 0x3000, 0x3000, 0xfc00, 0x3000, 0x3000, 0x3000, 0x3000,
0x3600, 0x1c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xcc00, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0x6c00, 0x3800,
0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600,
0xc600, 0xd600, 0xd600, 0xd600, 0xfe00, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0x6c00, 0x3800, 0x6c00, 0xc600, 0xc600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600,
0xc600, 0xc600, 0xce00, 0x7600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xfe00, 0x8600, 0x0c00, 0x1800, 0x3000, 0x6200, 0xfe00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0e00, 0x1800, 0x1800, 0x1800, 0x7000, 0x1800,
0x1800, 0x1800, 0x1800, 0x0e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x1800, 0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x1800, 0x1800, 0x1800, 0x0e00, 0x1800, 0x1800,
0x1800, 0x1800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7600, 0xdc00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x3800, 0x3800, 0x6c00, 0x6c00,
0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0xc000,
0xc000, 0xc000, 0xc600, 0x6600, 0x3c00, 0x1800, 0x0c00, 0xcc00, 0x3800, 0x0000, 0x0000,
0x0000, 0x0000, 0xc600, 0x0000, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xce00,
0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1800, 0x3000, 0x0000, 0x7c00,
0xc600, 0xc600, 0xfe00, 0xc000, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3000, 0x7800, 0xcc00, 0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0xcc00, 0xdc00, 0x7600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xcc00, 0x0000, 0x0000, 0x7800, 0x0c00,
0x7c00, 0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x3000, 0x1800, 0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00, 0x3800, 0x0000, 0x7800, 0x0c00, 0x7c00,
0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc000, 0xc000, 0xc600, 0x7c00, 0x1800, 0x0c00, 0x6c00, 0x3800,
0x0000, 0x0000, 0x0000, 0x3000, 0x7800, 0xcc00, 0x0000, 0x7c00, 0xc600, 0xc600, 0xfe00,
0xc000, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xcc00, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc600, 0xfe00, 0xc000, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x7c00, 0xc600, 0xc600, 0xfe00, 0xc000,
0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x0000, 0x0000,
0x3800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1800, 0x3c00, 0x6600, 0x0000, 0x3800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x3000, 0x1800, 0x0000, 0x3800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xc600, 0x0000, 0x3800, 0x6c00, 0xc600, 0xc600, 0xc600, 0xfe00, 0xc600, 0xc600, 0xc600,
0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00, 0x3800, 0x0000, 0x3800, 0x6c00, 0xc600,
0xc600, 0xfe00, 0xc600, 0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1800,
0x3000, 0x0000, 0xfe00, 0x6000, 0x6000, 0x7c00, 0x6000, 0x6000, 0x6000, 0xfe00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0xdb00, 0x1b00, 0x7f00,
0xd800, 0xd800, 0xdf00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00,
0xd800, 0xd800, 0xd800, 0xd800, 0xfe00, 0xd800, 0xd800, 0xd800, 0xde00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3000, 0x7800, 0xcc00, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600,
0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0x0000,
0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600,
0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x7800, 0xcc00, 0x0000,
0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xce00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x3000, 0x1800, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xce00,
0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0x0000, 0x0000, 0xc600, 0xc600,
0xc600, 0xc600, 0xce00, 0x7600, 0x0600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0xc600, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc600, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600,
0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1800, 0x1800, 0x7c00, 0xc600, 0xc000, 0xc000, 0xc600, 0x7c00, 0x1800, 0x1800, 0x0000,
```

```
0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00, 0x6000, 0x6000, 0xf000, 0x6000, 0x6000,
0x6000, 0x6600, 0xf600, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600,
0x6600, 0x6600, 0x3c00, 0x1800, 0x7e00, 0x1800, 0x3c00, 0x1800, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xfc00, 0xc600, 0xc600, 0xfc00, 0xc000, 0xcc00, 0xde00, 0xcc00,
0xcc00, 0xcc00, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0e00, 0x1b00, 0x1800,
0x1800, 0x1800, 0x7e00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xd800, 0x7000, 0x0000,
0x0000, 0x0000, 0x0c00, 0x1800, 0x3000, 0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0xcc00,
0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1800, 0x3000, 0x0000,
0x3800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0c00, 0x1800, 0x3000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600,
0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3000, 0x6000, 0x0000, 0xcc00,
0xcc00, 0xcc00, 0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7600, 0xdc00, 0x0000, 0xbc00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0xe600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7600, 0xdc00, 0x0000, 0xc600, 0xc600, 0xe600,
0xf600, 0xde00, 0xce00, 0xc600, 0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00,
0x6c00, 0x6c00, 0x3e00, 0x0000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00, 0x6c00, 0x3800, 0x0000, 0x7c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000,
0x3000, 0x0000, 0x3000, 0x3000, 0x3000, 0x6000, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7e00, 0x0600, 0x0600, 0x0600, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6200, 0x6600, 0x6c00, 0x1800, 0x3000, 0x6000, 0xdc00,
0x3600, 0x0c00, 0x1800, 0x3e00, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6200, 0x6600,
0x6c00, 0x1800, 0x3600, 0x6e00, 0xde00, 0x3600, 0x7e00, 0x0600, 0x0600, 0x0000, 0x0000,
0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x3c00, 0x3c00, 0x3c00, 0x3c00,
0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3600,
0x6c00, 0xd800, 0x6c00, 0x3600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xd800, 0x6c00, 0x3600, 0x6c00, 0xd800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1100, 0x4400, 0x1100, 0x4400, 0x1100, 0x4400, 0x1100,
0x4400, 0x1100, 0x4400, 0x1100, 0x4400, 0x1100, 0x4400, 0x1100, 0x4400, 0xaa00, 0x5500,
0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00,
0x5500, 0xaa00, 0x5500, 0xdd00, 0x7700, 0xdd00, 0x7700, 0xdd00, 0x7700, 0xdd00, 0x7700,
0xdd00, 0x7700, 0xdd00, 0x7700, 0xdd00, 0x7700, 0xdd00, 0x7700, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xf800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0xf800, 0x1800, 0xf800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xf600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xfe00, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x1800, 0xf800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xf600,
0x0600, 0xf600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0600,
0xf600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0xf600, 0x0600, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xfe00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0xf800, 0x1800, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xff00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1f00, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xff00, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1f00, 0x1800,
0x1f00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3700, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3700, 0x3000, 0x3f00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3f00, 0x3000, 0x3700, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xf700, 0x0000, 0xff00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xff00, 0x0000, 0xf700, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3700, 0x3000, 0x3700, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
```

```
0xff00, 0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xf700, 0x0000, 0xf700, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xff00,
0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x0000,
0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3f00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1f00, 0x1800, 0x1f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1f00, 0x1800, 0x1f00, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3f00, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xff00, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0xff00, 0x1800, 0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xf800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1f00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0xf000, 0xf000,
0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000,
0xf000, 0xf000, 0xf000, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00,
0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0xff00, 0xff00, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7600, 0xdc00, 0xd800, 0xd800, 0xd800,
0xd800, 0xdc00, 0x7600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7800, 0xcc00,
0xcc00, 0xd800, 0xfc00, 0xc600, 0xc600, 0xc600, 0xc600, 0xcc00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xfe00, 0x6600, 0x6200, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xfe00, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xfe00, 0xc600, 0x6200, 0x3000, 0x1800, 0x1800, 0x3000, 0x6200, 0xc600,
0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00,
0xd800, 0xcc00, 0xcc00, 0xcc00, 0xd800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x7c00, 0x6000, 0xc000,
0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7600, 0xdc00,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xfe00, 0x3800, 0x3800, 0x6c00, 0xc600, 0xc600, 0x6c00, 0x3800, 0x3800, 0xfe00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3800, 0x6c00, 0xc600, 0xc600, 0xfe00,
0xc600, 0xc600, 0x6c00, 0x3800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3800,
0x6c00, 0xc600, 0xc600, 0xc600, 0xc600, 0x6c00, 0x6c00, 0x6c00, 0xee00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3e00, 0x6000, 0x6000, 0x3c00, 0x6600, 0xc600, 0xc600,
0xc600, 0xcc00, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7e00, 0xdb00, 0xdb00, 0xdb00, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0200, 0x0600, 0x7c00, 0xce00, 0xde00, 0xf600, 0xf600, 0x7c00,
0x6000, 0xc000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1c00, 0x3000,
0x6000, 0x6000, 0x7c00, 0x6000, 0x6000, 0x3000, 0x1c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600,
0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0000,
0x0000, 0xfe00, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x7e00, 0x1800, 0x1800, 0x0000, 0x0000, 0x7e00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x1800, 0x0c00, 0x0600, 0x0c00,
0x1800, 0x3000, 0x0000, 0x0000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0c00, 0x1800, 0x3000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x0000, 0x7e00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1e00, 0x1a00, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x5800, 0x7800, 0x3000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x7e00, 0x0000,
0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7600, 0xdc00, 0x0000, 0x7600, 0xdc00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7800, 0xcc00, 0xcc00, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1f00, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0xd800, 0xd800, 0x7800, 0x3800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xd800, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0xd800, 0x1800, 0x3000, 0x6000,
0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7e00, 0x7e00, 0x7e00, 0x7e00, 0x7e00, 0x7e00, 0x0000, 0x0000,
```

```
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, };
```

File

lpc_rom8x16.c (see page 176)

Description

This is variable rom8x16_bits.

1.4.14 rom8x16_width

```
static UNS_8 rom8x16_width[] = { 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, };
```

File

lpc_rom8x16.c (see page 176)

Description

Character width data.

1.4.15 rom8x8_bits

```
static UNS_16 rom8x8_bits[] = { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7e00, 0x8100, 0xa500, 0x8100, 0xbd00, 0x9900, 0x8100, 0x7e00, 0x7c00, 0xfe00,
0xd600, 0xba00, 0xc600, 0xfe00, 0x7c00, 0x0000, 0xc600, 0xee00, 0xfe00, 0xfe00, 0x7c00,
0x3800, 0x1000, 0x0000, 0x1000, 0x3800, 0x7c00, 0xfe00, 0x7c00, 0x3800, 0x1000, 0x0000,
0x1000, 0x3800, 0x1000, 0xee00, 0xee00, 0x1000, 0x3800, 0x0000, 0x3800, 0x7c00, 0xfe00,
0xfe00, 0x6c00, 0x1000, 0x3800, 0x0000, 0x0000, 0x1800, 0x3c00, 0x7e00, 0x3c00, 0x1800,
0x0000, 0x0000, 0xff00, 0xe700, 0xc300, 0x8100, 0xc300, 0xe700, 0xff00, 0xff00, 0x0000,
0x1800, 0x3c00, 0x6600, 0x6600, 0x3c00, 0x1800, 0x0000, 0xff00, 0xe700, 0xc300, 0x9900,
0x9900, 0xc300, 0xe700, 0xff00, 0x1e00, 0x0e00, 0x1e00, 0x3600, 0x7800, 0xcc00, 0xcc00,
0x7800, 0x7e00, 0xc300, 0xc300, 0x7e00, 0x1800, 0x7e00, 0x1800, 0x1800, 0x1e00, 0x1a00,
0x1e00, 0x1800, 0x1800, 0x7000, 0xf000, 0x6000, 0x3e00, 0x3e00, 0x3600, 0x3600, 0xf600,
0x6600, 0x1e00, 0x0c00, 0xdb00, 0x3c00, 0x6600, 0xe700, 0x6600, 0x3c00, 0xdb00, 0x0000,
0x8000, 0xc000, 0xf000, 0xf800, 0xf000, 0xc000, 0x8000, 0x0000, 0x0200, 0x0600, 0x1e00,
0x3e00, 0x1e00, 0x0600, 0x0200, 0x0000, 0x1800, 0x3c00, 0x7e00, 0x1800, 0x7e00, 0x3c00,
0x1800, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x0000, 0x6600, 0x0000, 0x7f00,
0xdb00, 0x7b00, 0x3b00, 0x1b00, 0x1b00, 0x1b00, 0x0000, 0x3c00, 0x6600, 0x3800, 0x6c00,
0x6c00, 0x3800, 0xcc00, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0xfe00, 0xfe00,
0x0000, 0x1800, 0x3c00, 0x7e00, 0x1800, 0x7e00, 0x3c00, 0x1800, 0x7e00, 0x1800, 0x3c00,
0x7e00, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x1800, 0x1800, 0x7e00,
0x3c00, 0x1800, 0x0000, 0x0000, 0x1800, 0x1c00, 0xfe00, 0x1c00, 0x1800, 0x0000, 0x0000,
0x0000, 0x3000, 0x7000, 0xfe00, 0x7000, 0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc000,
0xc000, 0xc000, 0xfe00, 0x0000, 0x0000, 0x0000, 0x2400, 0x6600, 0xff00, 0x6600, 0x2400,
0x0000, 0x0000, 0x0000, 0x1000, 0x3800, 0x7c00, 0x7c00, 0xfe00, 0x0000, 0x0000, 0x0000,
```

```
0xfe00, 0x7c00, 0x7c00, 0x3800, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x3c00, 0x1800, 0x1800, 0x0000, 0x1800,
0x0000, 0x6c00, 0x6c00, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6c00, 0x6c00,
0xfe00, 0x6c00, 0xfe00, 0x6c00, 0x6c00, 0x0000, 0x1800, 0x7e00, 0xc000, 0x7c00, 0x0600,
0xfc00, 0x1800, 0x0000, 0x0000, 0xc600, 0x0c00, 0x1800, 0x3000, 0x6000, 0xc600, 0x0000,
0x3800, 0x6c00, 0x3800, 0x7600, 0xcc00, 0xcc00, 0x7600, 0x0000, 0x1800, 0x1800, 0x3000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3000, 0x6000, 0x6000, 0x6000, 0x3000,
0x1800, 0x0000, 0x6000, 0x3000, 0x1800, 0x1800, 0x1800, 0x3000, 0x6000, 0x0000, 0x0000,
0xee00, 0x7c00, 0xfe00, 0x7c00, 0xee00, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x7e00,
0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x3000,
0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3800, 0x3800, 0x0000, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000,
0xc000, 0x8000, 0x0000, 0x7c00, 0xc600, 0xce00, 0xde00, 0xf600, 0xe600, 0x7c00, 0x0000,
0x1800, 0x7800, 0x1800, 0x1800, 0x1800, 0x1800, 0x7e00, 0x0000, 0x7c00, 0xc600, 0x0c00,
0x1800, 0x3000, 0x6600, 0xfe00, 0x0000, 0x7c00, 0xc600, 0x0600, 0x3c00, 0x0600, 0xc600,
0x7c00, 0x0000, 0x0c00, 0x1c00, 0x3c00, 0x6c00, 0xfe00, 0x0c00, 0x0c00, 0x0000, 0xfe00,
0xc000, 0xfc00, 0x0600, 0x0600, 0xc600, 0x7c00, 0x0000, 0x7c00, 0xc600, 0xc000, 0xfc00,
0xc600, 0xc600, 0x7c00, 0x0000, 0xfe00, 0xc600, 0x0600, 0x0c00, 0x1800, 0x1800, 0x1800,
0x0000, 0x7c00, 0xc600, 0xc600, 0x7c00, 0xc600, 0xc600, 0x7c00, 0x0000, 0x7c00, 0xc600,
0xc600, 0x7e00, 0x0600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x1c00, 0x1c00, 0x0000, 0x0000,
0x1c00, 0x1c00, 0x0000, 0x0000, 0x1800, 0x1800, 0x0000, 0x0000, 0x1800, 0x1800, 0x3000,
0x0c00, 0x1800, 0x3000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x0000, 0x0000, 0x0000, 0xfe00,
0x0000, 0x0000, 0xfe00, 0x0000, 0x0000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x1800, 0x3000,
0x6000, 0x0000, 0x7c00, 0xc600, 0x0600, 0x0c00, 0x1800, 0x0000, 0x1800, 0x0000, 0x7c00,
0xc600, 0xc600, 0xde00, 0xdc00, 0xc000, 0x7e00, 0x0000, 0x3800, 0x6c00, 0xc600, 0xc600,
0xfe00, 0xc600, 0xc600, 0x0000, 0xfc00, 0x6600, 0x6600, 0x7c00, 0x6600, 0x6600, 0xfc00,
0x0000, 0x3c00, 0x6600, 0xc000, 0xc000, 0xc000, 0x6600, 0x3c00, 0x0000, 0xf800, 0x6c00,
0x6600, 0x6600, 0x6600, 0x6c00, 0xf800, 0x0000, 0xfe00, 0xc200, 0xc000, 0xf800, 0xc000,
0xc200, 0xfe00, 0x0000, 0xfe00, 0x6200, 0x6000, 0x7c00, 0x6000, 0x6000, 0xf000, 0x0000,
0x7c00, 0xc600, 0xc000, 0xc000, 0xde00, 0xc600, 0x7c00, 0x0000, 0xc600, 0xc600, 0xc600,
0xfe00, 0xc600, 0xc600, 0xc600, 0x0000, 0x3c00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x3c00, 0x0000, 0x3c00, 0x1800, 0x1800, 0x1800, 0xd800, 0xd800, 0x7000, 0x0000, 0xc600,
0xcc00, 0xd800, 0xf000, 0xd800, 0xcc00, 0xc600, 0x0000, 0xf000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6200, 0xfe00, 0x0000, 0xc600, 0xee00, 0xfe00, 0xd600, 0xd600, 0xc600, 0xc600,
0x0000, 0xc600, 0xe600, 0xe600, 0xf600, 0xde00, 0xce00, 0xc600, 0x0000, 0x7c00, 0xc600,
0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0xfc00, 0x6600, 0x6600, 0x7c00, 0x6000,
0x6000, 0xf000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xd600, 0x7c00, 0x0600,
0xfc00, 0xc600, 0xc600, 0xfc00, 0xd800, 0xcc00, 0xc600, 0x0000, 0x7c00, 0xc600, 0xc000,
0x7c00, 0x0600, 0xc600, 0x7c00, 0x0000, 0x7e00, 0x5a00, 0x1800, 0x1800, 0x1800, 0x1800,
0x3c00, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0xc600,
0xc600, 0xc600, 0xc600, 0x6c00, 0x3800, 0x1000, 0x0000, 0xc600, 0xc600, 0xd600, 0xd600,
0xfe00, 0xee00, 0xc600, 0x0000, 0xc600, 0x6c00, 0x3800, 0x3800, 0x3800, 0x6c00, 0xc600,
0x0000, 0x6600, 0x6600, 0x6600, 0x3c00, 0x1800, 0x1800, 0x3c00, 0x0000, 0xfe00, 0x8600,
0x0c00, 0x1800, 0x3000, 0x6200, 0xfe00, 0x0000, 0x7c00, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x7c00, 0x0000, 0xc000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x0600, 0x0200, 0x0000,
0x7c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x7c00, 0x0000, 0x1000, 0x3800, 0x6c00,
0xc600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xff00, 0x3000, 0x3000, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0x7e00, 0x0000, 0xe000, 0x6000, 0x7c00, 0x6600,
0x6600, 0x6600, 0xfc00, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc000, 0xc600, 0x7c00,
0x0000, 0x1c00, 0x0c00, 0x7c00, 0xcc00, 0xcc00, 0xcc00, 0x7e00, 0x0000, 0x0000, 0x0000,
0x7c00, 0xc600, 0xfe00, 0xc000, 0x7c00, 0x0000, 0x1c00, 0x3600, 0x3000, 0xfc00, 0x3000,
0x3000, 0x7800, 0x0000, 0x0000, 0x0000, 0x7600, 0xce00, 0xc600, 0x7e00, 0x0600, 0x7c00,
0xe000, 0x6000, 0x7c00, 0x6600, 0x6600, 0x6600, 0xe600, 0x0000, 0x1800, 0x0000, 0x3800,
0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0x0c00, 0x0000, 0x1c00, 0x0c00, 0x0c00, 0x0c00,
0xcc00, 0x7800, 0xe000, 0x6000, 0x6600, 0x6c00, 0x7800, 0x6c00, 0xe600, 0x0000, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1c00, 0x0000, 0x0000, 0x0000, 0x6c00, 0xfe00,
0xd600, 0xd600, 0xc600, 0x0000, 0x0000, 0x0000, 0xdc00, 0x6600, 0x6600, 0x6600, 0x6600,
0x0000, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0x0000,
0xdc00, 0x6600, 0x6600, 0x7c00, 0x6000, 0xf000, 0x0000, 0x0000, 0x7600, 0xcc00, 0xcc00,
0x7c00, 0x0c00, 0x1e00, 0x0000, 0x0000, 0xdc00, 0x6600, 0x6000, 0x6000, 0xf000, 0x0000,
0x0000, 0x0000, 0x7c00, 0xc000, 0x7c00, 0x0600, 0x7c00, 0x0000, 0x3000, 0x3000, 0xfc00,
0x3000, 0x3000, 0x3600, 0x1c00, 0x0000, 0x0000, 0x0000, 0xcc00, 0xcc00, 0xcc00, 0xcc00,
0x7600, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0x6c00, 0x3800, 0x1000, 0x0000, 0x0000,
0x0000, 0xc600, 0xc600, 0xd600, 0xfe00, 0x6c00, 0x0000, 0x0000, 0x0000, 0xc600, 0x6c00,
0x3800, 0x6c00, 0xc600, 0x0000, 0x0000, 0x0000, 0xc600, 0xc600, 0xce00, 0x7600, 0x0600,
0x7c00, 0x0000, 0x0000, 0xfc00, 0x9800, 0x3000, 0x6400, 0xfc00, 0x0000, 0x0e00, 0x1800,
0x1800, 0x7000, 0x1800, 0x1800, 0x0e00, 0x0000, 0x1800, 0x1800, 0x1800, 0x0000, 0x1800,
0x1800, 0x1800, 0x0000, 0x7000, 0x1800, 0x1800, 0x0e00, 0x1800, 0x1800, 0x7000, 0x0000,
0x7600, 0xdc00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x3800,
0x3800, 0x6c00, 0x6c00, 0xfe00, 0x0000, 0x3c00, 0x6600, 0xc000, 0x6600, 0x3c00, 0x1800,
0xcc00, 0x7800, 0x0000, 0xc600, 0x0000, 0xc600, 0xc600, 0xce00, 0x7600, 0x0000, 0x0e00,
```

```
0x0000, 0x7c00, 0xc600, 0xfe00, 0xc000, 0x7c00, 0x0000, 0x7c00, 0xc600, 0x7800, 0x0c00,
0x7c00, 0xcc00, 0x7e00, 0x0000, 0xc600, 0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0x7e00,
0x0000, 0xe000, 0x0000, 0x7800, 0x0c00, 0x7c00, 0xcc00, 0x7e00, 0x0000, 0x3800, 0x3800,
0x7800, 0x0c00, 0x7c00, 0xcc00, 0x7e00, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc000, 0x7c00,
0x1800, 0x6c00, 0x3800, 0x7c00, 0xc600, 0x7c00, 0xc600, 0xfe00, 0xc000, 0x7c00, 0x0000,
0xc600, 0x0000, 0x7c00, 0xc600, 0xfe00, 0xc000, 0x7c00, 0x0000, 0xe000, 0x0000, 0x7c00,
0xc600, 0xfe00, 0xc000, 0x7c00, 0x0000, 0x6600, 0x0000, 0x3800, 0x1800, 0x1800, 0x1800,
0x3c00, 0x0000, 0x7c00, 0xc600, 0x3800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0xe000,
0x0000, 0x3800, 0x1800, 0x1800, 0x1800, 0x3c00, 0x0000, 0xc600, 0x3800, 0x6c00, 0xc600,
0xfe00, 0xc600, 0xc600, 0x0000, 0x3800, 0x3800, 0x0000, 0x7c00, 0xc600, 0xfe00, 0xc600,
0x0000, 0x0e00, 0x0000, 0xfe00, 0xc000, 0xf800, 0xc000, 0xfe00, 0x0000, 0x0000, 0x0000,
0x6c00, 0x9a00, 0x7e00, 0xd800, 0x6e00, 0x0000, 0x7e00, 0xd800, 0xd800, 0xfe00, 0xd800,
0xd800, 0xde00, 0x0000, 0x7c00, 0xc600, 0x0000, 0x7c00, 0xc600, 0xc600, 0x7c00, 0x0000,
0x0000, 0xc600, 0x0000, 0x7c00, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000, 0xe000, 0x0000,
0x7c00, 0xc600, 0xc600, 0x7c00, 0x0000, 0x7c00, 0xc600, 0x0000, 0xc600, 0xc600, 0xce00,
0x7600, 0x0000, 0x0000, 0xe000, 0x0000, 0xc600, 0xc600, 0xce00, 0x7600, 0x0000, 0x0000,
0xc600, 0x0000, 0xc600, 0xce00, 0x7600, 0x0600, 0x7c00, 0xc600, 0x3800, 0x6c00, 0xc600,
0xc600, 0x6c00, 0x3800, 0x0000, 0xc600, 0x0000, 0xc600, 0xc600, 0xc600, 0xc600, 0x7c00,
0x0000, 0x0000, 0x1800, 0x7e00, 0xd800, 0xd800, 0x7e00, 0x1800, 0x0000, 0x3800, 0x6c00,
0x6000, 0xf000, 0x6600, 0xf600, 0x6c00, 0x0000, 0xc300, 0x6600, 0x3c00, 0x7e00, 0x1800,
0x3c00, 0x1800, 0x0000, 0xfc00, 0xc600, 0xfc00, 0xcc00, 0xde00, 0xcc00, 0xce00, 0x0000,
0x0c00, 0x1e00, 0x1800, 0x7e00, 0x1800, 0x1800, 0xd800, 0x7000, 0x0e00, 0x0000, 0x7800,
0x0c00, 0x7c00, 0xcc00, 0x7e00, 0x0000, 0x1c00, 0x0000, 0x3800, 0x1800, 0x1800, 0x1800,
0x3c00, 0x0000, 0x0000, 0x0e00, 0x0000, 0x7c00, 0xc600, 0xc600, 0x7c00, 0x0000, 0x0000,
0x0e00, 0x0000, 0xcc00, 0xcc00, 0xdc00, 0x7600, 0x0000, 0x0000, 0xfc00, 0x0000, 0xbc00,
0x6600, 0x6600, 0xe600, 0x0000, 0xfe00, 0x0000, 0xc600, 0xe600, 0xf600, 0xce00, 0xc600,
0x0000, 0x3800, 0x6c00, 0x3e00, 0x0000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x7c00, 0xc600,
0x7c00, 0x0000, 0x7c00, 0x0000, 0x0000, 0x0000, 0x1800, 0x0000, 0x1800, 0x3000, 0x6000,
0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0x0c00, 0x0c00, 0x0000, 0x0000, 0xc000, 0xcc00, 0xd800,
0x3000, 0x7c00, 0x3600, 0x0c00, 0x3e00, 0xc000, 0xcc00, 0xd800, 0x3000, 0x6c00, 0x3c00,
0x7e00, 0x0c00, 0x1800, 0x0000, 0x1800, 0x1800, 0x3c00, 0x3c00, 0x1800, 0x0000, 0x0000,
0x3600, 0x6c00, 0xd800, 0x6c00, 0x3600, 0x0000, 0x0000, 0x0000, 0xd800, 0x6c00, 0x3600,
0x6c00, 0xd800, 0x0000, 0x0000, 0x2200, 0x8800, 0x2200, 0x8800, 0x2200, 0x8800, 0x2200,
0x8800, 0x5500, 0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00, 0x5500, 0xaa00, 0xdd00, 0x7700,
0xdd00, 0x7700, 0xdd00, 0x7700, 0xdd00, 0x7700, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xf800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0xf800, 0x1800, 0xf800, 0x1800, 0x1800, 0x1800, 0x3600, 0x3600, 0x3600,
0x3600, 0xf600, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0x0000, 0x0000, 0xfe00, 0x3600,
0x3600, 0x3600, 0x0000, 0x0000, 0xf800, 0x1800, 0xf800, 0x1800, 0x1800, 0x1800, 0x3600,
0x3600, 0xf600, 0x0600, 0xf600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0xfe00, 0x0600, 0xf600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3600, 0xf600, 0x0600, 0xfe00, 0x0000, 0x0000, 0x0000, 0x3600, 0x3600,
0x3600, 0x3600, 0xfe00, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0xf800, 0x1800, 0xf800,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1f00, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1800,
0x1800, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1f00, 0x1800, 0x1800, 0x1800, 0x0000,
0x0000, 0x0000, 0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1800, 0x1800,
0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1f00, 0x1800, 0x1f00, 0x1800, 0x1800,
0x1800, 0x3600, 0x3600, 0x3600, 0x3600, 0x3700, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3700, 0x3000, 0x3f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3f00, 0x3000, 0x3700,
0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0xf700, 0x0000, 0xff00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xff00, 0x0000, 0xf700, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3700,
0x3000, 0x3700, 0x3600, 0x3600, 0x3600, 0x0000, 0x0000, 0xff00, 0x0000, 0xff00, 0x0000,
0x0000, 0x0000, 0x3600, 0x3600, 0xf700, 0x0000, 0xf700, 0x3600, 0x3600, 0x3600, 0x1800,
0x1800, 0xff00, 0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x3600, 0x3600, 0x3600, 0x3600,
0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x0000, 0xff00, 0x1800, 0x1800,
0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0x3600, 0x3f00, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x1f00, 0x1800, 0x1f00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1f00, 0x1800, 0x1f00, 0x1800, 0x1800, 0x1800,
0x0000, 0x0000, 0x0000, 0x0000, 0x3f00, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600, 0x3600,
0x3600, 0xff00, 0x3600, 0x3600, 0x3600, 0x1800, 0x1800, 0xff00, 0x1800, 0xff00, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x1f00, 0x1800, 0x1800, 0x1800, 0xff00, 0xff00, 0xff00, 0xff00,
0xff00, 0xff00, 0xff00, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0xff00, 0xff00,
0xff00, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0x0f00, 0x0f00,
0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0x0f00, 0xff00, 0xff00, 0xff00, 0xff00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0xdc00, 0xd800, 0xdc00, 0x6600, 0x0000,
0x0000, 0x7800, 0xcc00, 0xf800, 0xcc00, 0xc600, 0xcc00, 0x0000, 0x0000, 0xfe00, 0x6200,
0x6000, 0x6000, 0x6000, 0xe000, 0x0000, 0x0000, 0xfe00, 0x6c00, 0x6c00, 0x6c00, 0x6c00,
0x6c00, 0x0000, 0xfe00, 0xc600, 0x6000, 0x3000, 0x6000, 0xc600, 0xfe00, 0x0000, 0x0000,
```

```
0x7e00, 0xd800, 0xcc00, 0xcc00, 0xd800, 0x7000, 0x0000, 0x0000, 0x6600, 0x6600, 0x6600,
0x6600, 0x7c00, 0xc000, 0x0000, 0x0000, 0x7600, 0xdc00, 0x1800, 0x1800, 0x1800, 0x3800,
0x0000, 0xfe00, 0x3800, 0x6c00, 0xc600, 0x6c00, 0x3800, 0xfe00, 0x0000, 0x3800, 0x6c00,
0xc600, 0xfe00, 0xc600, 0x6c00, 0x3800, 0x0000, 0x3800, 0x6c00, 0xc600, 0xc600, 0x6c00,
0x6c00, 0xee00, 0x0000, 0x3e00, 0x6000, 0x3800, 0x6600, 0xc600, 0xcc00, 0x7800, 0x0000,
0x0000, 0x0000, 0x7e00, 0xdb00, 0xdb00, 0x7e00, 0x0000, 0x0000, 0x0600, 0x7c00, 0xde00,
0xf600, 0xe600, 0x7c00, 0xc000, 0x0000, 0x3800, 0x6000, 0xc000, 0xf800, 0xc000, 0x6000,
0x3800, 0x0000, 0x7c00, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0xc600, 0x0000, 0x0000,
0xfe00, 0x0000, 0xfe00, 0x0000, 0xfe00, 0x0000, 0x0000, 0x1800, 0x1800, 0x7e00, 0x1800,
0x1800, 0x0000, 0x7e00, 0x0000, 0x3000, 0x1800, 0x0c00, 0x1800, 0x3000, 0x0000, 0x7e00,
0x0000, 0x0c00, 0x1800, 0x3000, 0x1800, 0x0c00, 0x0000, 0x7e00, 0x0000, 0x0c00, 0x1e00,
0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x7800, 0x3000, 0x0000, 0x0000, 0x0000, 0x1800, 0x0000, 0x7e00, 0x0000, 0x1800, 0x0000,
0x0000, 0x7600, 0xdc00, 0x0000, 0x7600, 0xdc00, 0x0000, 0x0000, 0x7c00, 0xc600, 0xc600,
0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x0000, 0x0000, 0x0000, 0x1f00,
0x1800, 0x1800, 0x1800, 0xf800, 0x3800, 0x1800, 0x0000, 0xd800, 0x6c00, 0x6c00, 0x6c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0xd800, 0x3000, 0xf800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0x7c00, 0x7c00, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, };
```

File

lpc_rom8x8.c (see page 177)

Description

This is variable rom8x8_bits.

1.4.16 rom8x8_width

```
static UNS_8 rom8x8_width[] = { 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8,
8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, };
```

File

lpc_rom8x8.c (see page 177)

Description

Character width data.

1.4.17 sharp_lm057qb

```
const LCD_PARAM_T sharp_lm057qb;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LM057QB STN display

1.4.18 sharp_lm057qc

```
const LCD_PARAM_T sharp_lm057qc;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LM057QC STN display

1.4.19 sharp_lm10v

```
const LCD_PARAM_T sharp_lm10v;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LM10V DSTN display

1.4.20 sharp_lm64k11

```
const LCD_PARAM_T sharp_lm64k11;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LM64K11 STN display

1.4.21 sharp_lq035

```
const LCD_PARAM_T sharp_lq035;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ035 portrait mode ADTFT display

1.4.22 sharp_lq039

```
const LCD_PARAM_T sharp_lq039;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ039 HRTFT display

1.4.23 sharp_lq050

```
const LCD_PARAM_T sharp_lq050;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ050 TFT display - also works for the LQ036 and LQ038 LCDs

1.4.24 sharp_lq057

```
const LCD_PARAM_T sharp_lq057;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ057 TFT display

1.4.25 sharp_lq064

```
const LCD_PARAM_T sharp_lq064;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ064 TFT display

1.4.26 sharp_lq104

```
const LCD_PARAM_T sharp_lq104;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ104 TFT display

1.4.27 sharp_lq121

```
const LCD_PARAM_T sharp_lq121;
```

File

lpc_lcd_params.c (see page 175)

Description

Sharp LQ121 TFT display

1.4.28 virtual_tlb_addr

```
UNS_32 * virtual_tlb_addr;
```

File

lpc_arm922t_cp15_driver.c (see page 158)

Description

The address translation functions of this driver require a saved pointer to the virtual base address of the MMU table.

1.4.29 winfreesystem14x16_bits

```
static UNS_16 winfreesystem14x16_bits[] = { 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xcc00, 0xcc00, 0xcc00, 0xcc00, 0xcc00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3600, 0x3600, 0x7f00, 0x7f00, 0x3600, 0x3600, 0x6c00, 0x6c00, 0xfe00, 0xfe00, 0x6c00,
0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x7e00, 0xdb00, 0xdb00, 0xd800, 0xfc00,
0x3e00, 0x1b00, 0xdb00, 0xdb00, 0x7e00, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7060,
0xd8c0, 0xd980, 0xdb00, 0x7600, 0x0600, 0x0dc0, 0x1b60, 0x3360, 0x6360, 0xc1c0, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1c00, 0x3600, 0x2200, 0x2200, 0x3600, 0x1c00, 0x3900,
0x6d00, 0x4700, 0x6600, 0x3f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x3000, 0x0000, 0x0000, 0xc000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0xc000,
0x0000, 0x0000, 0x3000, 0x3000, 0xfc00, 0x3000, 0x7800, 0x4800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800,
0x1800, 0x1800, 0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0xc000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xf000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xe000, 0xe000, 0xe000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x3000, 0x3000, 0x3000, 0x7000, 0x6000,
0x6000, 0x6000, 0x6000, 0xe000, 0xc000, 0xc000, 0xc000, 0x0000, 0x0000, 0x0000, 0x3c00,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x7800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600,
```

```
0x6600, 0x6600, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000, 0x6000, 0x7e00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600, 0x0600, 0x0600, 0x1c00, 0x0600, 0x0600,
0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600,
0x6600, 0x6600, 0x6600, 0x7e00, 0x7e00, 0x0600, 0x0600, 0x0600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7e00, 0x6000, 0x6000, 0x6000, 0x7c00, 0x6600, 0x0600, 0x0600, 0x6600,
0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6000,
0x6000, 0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7e00, 0x0600, 0x0600, 0x0c00, 0x0c00, 0x7e00, 0x1800, 0x1800, 0x3000, 0x3000,
0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6600, 0x6600,
0x3c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0600, 0x0600, 0x0600, 0x6600, 0x3c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0xc000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0600, 0x0c00, 0x1800, 0x3000, 0x6000, 0x6000,
0x3000, 0x1800, 0x0c00, 0x0600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7e00, 0x7e00, 0x0000, 0x7e00, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x3000, 0x1800, 0x0c00, 0x0600, 0x0600, 0x0c00,
0x1800, 0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600,
0x6600, 0x0600, 0x0c00, 0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0780, 0x1ce0, 0x3870, 0x3330, 0x6798, 0x66d8, 0x6cd8, 0x6cd8, 0x6d98, 0x6798,
0x32f0, 0x3000, 0x1c70, 0x07c0, 0x0000, 0x0000, 0x0000, 0x1800, 0x1800, 0x3c00, 0x3c00,
0x2400, 0x6600, 0x7e00, 0x7e00, 0xe700, 0xc300, 0xc300, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x7f00, 0x6180, 0x6180, 0x6180, 0x6180, 0x7f00, 0x6180, 0x6180, 0x6180, 0x6180,
0x7f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1e00, 0x3300, 0x6100, 0x6100, 0x6000,
0x6000, 0x6000, 0x6100, 0x6100, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7e00, 0x6300, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6300, 0x7e00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7f00, 0x6000, 0x6000, 0x6000, 0x6000, 0x7f00,
0x6000, 0x6000, 0x6000, 0x6000, 0x7f00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7f00,
0x6000, 0x6000, 0x6000, 0x6000, 0x7f00, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1f00, 0x3180, 0x6080, 0x6080, 0x6000, 0x6000, 0x6780,
0x6180, 0x6180, 0x3180, 0x1e80, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x7f80, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00, 0x0c00, 0x0c00,
0x0c00, 0x0c00, 0x0c00, 0x0c00, 0xcc00, 0xcc00, 0xcc00, 0x7800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6180, 0x6300, 0x6600, 0x6c00, 0x7800, 0x7000, 0x7800, 0x6c00, 0x6600,
0x6300, 0x6180, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x7f00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6060, 0x6060, 0x70e0, 0x70e0, 0x79e0, 0x79e0, 0x6f60, 0x6f60, 0x6660, 0x6660,
0x6060, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6180, 0x7180, 0x7180, 0x7980, 0x7d80,
0x6d80, 0x6f80, 0x6780, 0x6380, 0x6380, 0x6180, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1e00, 0x3300, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7f00, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180,
0x7f00, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1e00,
0x3300, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6580, 0x6780, 0x3300, 0x1f80, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7f00, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x7f00,
0x6180, 0x6180, 0x6180, 0x60c0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3e00, 0x6300,
0x6300, 0x6300, 0x3800, 0x0e00, 0x0300, 0x6300, 0x6300, 0x6300, 0x3e00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xff00, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800, 0x1800,
0x1800, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6180, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000,
0x0000, 0x0000, 0xc300, 0xc300, 0xc300, 0x6600, 0x6600, 0x6600, 0x2400, 0x3c00, 0x3c00,
0x1800, 0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc00c, 0xc30c, 0xc30c, 0xc30c,
0x6798, 0x6798, 0x34b0, 0x3cf0, 0x1860, 0x1860, 0x1860, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xc180, 0xc180, 0x6300, 0x3600, 0x1c00, 0x1c00, 0x1c00, 0x3600, 0x6300, 0xc180,
0xc180, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc0c0, 0xc0c0, 0xc0c0, 0x6180, 0x3300,
0x1e00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xff80, 0x0180, 0x0300, 0x0600, 0x0c00, 0x0800, 0x1800, 0x3000, 0x6000, 0xc000, 0xff80,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x7000, 0x0000, 0x0000, 0xc000,
0xc000, 0xc000, 0xc000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x3000, 0x3000, 0x3000,
0x3000, 0x0000, 0x0000, 0x0000, 0xe000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0xe000, 0x0000, 0x0000, 0x2000, 0x7000,
0xf800, 0xd800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x0000, 0x6000, 0x7000, 0x3000, 0x1800,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x4600, 0x1e00, 0x3600, 0x6600,
0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6000, 0x6000, 0x6000, 0x6000, 0x6600,
```

```
0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0600, 0x0600, 0x0600, 0x0600, 0x3e00,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600, 0x7e00, 0x6000, 0x6000, 0x6600, 0x3c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x6000, 0x6000, 0x6000, 0xf000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3e00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0600,
0x6600, 0x3c00, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x7c00, 0x6600, 0x6600,
0x6600, 0x6600, 0x6600, 0x6600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0xc000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6600, 0x6c00, 0x7800, 0x7000, 0x7800, 0x6c00, 0x6600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7fc0,
0x6660, 0x6660, 0x6660, 0x6660, 0x6660, 0x6660, 0x6660, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x7c00,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3e00, 0x6600, 0x6600,
0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0600, 0x0600, 0x0600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7800, 0x7000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6200, 0x3800,
0x1c00, 0x4600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x6000, 0xf000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x3000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xc300, 0xc300, 0x6600, 0x6600, 0x6600, 0x3c00, 0x1800, 0x1800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc0c0, 0xcccc, 0xcccc, 0x6d80, 0x6d80, 0x7f80,
0x3300, 0x3300, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xc300,
0x6600, 0x3c00, 0x1800, 0x1800, 0x3c00, 0x6600, 0xc300, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xc300, 0xc300, 0xc300, 0x6600, 0x6600, 0x3c00, 0x3c00,
0x1800, 0x1800, 0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00, 0x0600,
0x0c00, 0x1800, 0x1800, 0x3000, 0x6000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1800, 0x3000, 0x3000, 0x3000, 0x3000, 0x3000, 0x6000, 0x6000, 0x3000, 0x3000, 0x3000,
0x3000, 0x3000, 0x1800, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0xc000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x3000, 0x3000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0xc000, 0x0000, 0x0000, 0xe800, 0xb800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000,
0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0xc000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000,
0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0xc000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000,
0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0xc000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000,
```

```
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0c00,
0x3c00, 0x6e00, 0x6800, 0x6800, 0x6800, 0x7000, 0x7600, 0x3c00, 0x3000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3800, 0x6c00, 0x6000, 0x6000, 0x6000, 0x3000, 0x7800, 0x3000, 0x3000,
0x6600, 0x7c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x3c00, 0x2400, 0x2400,
0x2400, 0x3c00, 0x6600, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xc300, 0xc300, 0xc300, 0x6600, 0x6600, 0xff00, 0x1800, 0xff00, 0x1800, 0x1800,
0x1800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x3c00, 0x6600, 0x2400, 0x3800, 0x3800, 0x6c00, 0x3600, 0x1c00, 0x1c00, 0x2400, 0x6600,
0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0xd800, 0xd800, 0xd800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3f00,
0x6180, 0xccc0, 0xd2c0, 0xd2c0, 0xd0c0, 0xd2c0, 0xd2c0, 0xccc0, 0x6180, 0x3f00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x1000, 0x3000, 0x5000, 0x3000, 0x0000,
0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3600, 0x6c00, 0xd800, 0x6c00, 0x3600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00, 0x0600, 0x0600,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3f00, 0x6180, 0xdcc0, 0xd2c0, 0xd2c0, 0xd2c0, 0xdcc0, 0xd4c0, 0xd2c0,
0x6180, 0x3f00, 0x0000, 0x0000, 0x0000, 0x0000, 0xff00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7000, 0x5000, 0x5000, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x1800, 0x7e00, 0x1800, 0x1800, 0x0000, 0x7e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0xb000, 0x3000, 0x6000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0xb000, 0x6000, 0xb000, 0x6000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3800,
0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x6600, 0x6600, 0x7f00, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x3c00,
0x7c00, 0x7c00, 0x7c00, 0x7c00, 0x3c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x1800, 0x7000,
0x0000, 0x0000, 0x0000, 0x6000, 0xe000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x5000, 0x5000,
0x5000, 0x7000, 0x0000, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xd800, 0x6c00, 0x3600, 0x6c00, 0xd800,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3060, 0x70c0, 0x3180, 0x3300,
0x3600, 0x0cc0, 0x19c0, 0x3340, 0x63c0, 0xc0c0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3060, 0x70c0, 0x3180, 0x3300, 0x3600, 0x0d80, 0x1ac0, 0x30c0, 0x6180, 0xc3c0,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3060, 0x58c0, 0x3180, 0x5b00, 0x3600,
0x0cc0, 0x19c0, 0x3340, 0x63c0, 0xc0c0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1800, 0x1800, 0x0000, 0x1800, 0x1800, 0x3000, 0x6000, 0x6600, 0x6600, 0x3c00, 0x0000,
0x0000, 0x0000, 0x3800, 0x1800, 0x0c00, 0x1800, 0x1800, 0x3c00, 0x3c00, 0x2400, 0x6600,
0x6600, 0x7e00, 0xc300, 0xc300, 0x0000, 0x0000, 0x0000, 0x1c00, 0x1800, 0x3000, 0x1800,
0x1800, 0x3c00, 0x3c00, 0x2400, 0x6600, 0x6600, 0x7e00, 0xc300, 0xc300, 0x0000, 0x0000,
0x0000, 0x1800, 0x3c00, 0x6600, 0x1800, 0x1800, 0x3c00, 0x3c00, 0x2400, 0x6600, 0x6600,
0x7e00, 0xc300, 0xc300, 0x0000, 0x0000, 0x0000, 0x7a00, 0x5e00, 0x0000, 0x1800, 0x1800,
0x3c00, 0x3c00, 0x2400, 0x6600, 0x6600, 0x7e00, 0xc300, 0xc300, 0x0000, 0x0000, 0x0000,
0x6600, 0x6600, 0x0000, 0x1800, 0x1800, 0x3c00, 0x3c00, 0x2400, 0x6600, 0x6600, 0x7e00,
0xc300, 0xc300, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x1800, 0x0000, 0x1800, 0x1800,
0x3c00, 0x2400, 0x6600, 0x6600, 0x7e00, 0xc300, 0xc300, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1ff0, 0x1e00, 0x3600, 0x3600, 0x37e0, 0x6600, 0x6600, 0x7e00, 0xc600,
```

```
0xc7f0, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1e00, 0x3300, 0x6100, 0x6000,
0x6000, 0x6000, 0x6000, 0x6100, 0x3300, 0x1e00, 0x0c00, 0x0600, 0x1c00, 0x1c00, 0x0c00,
0x0600, 0x7f00, 0x6000, 0x6000, 0x6000, 0x7e00, 0x6000, 0x6000, 0x6000, 0x6000, 0x7f00,
0x0000, 0x0000, 0x0000, 0x0e00, 0x0c00, 0x1800, 0x7f00, 0x6000, 0x6000, 0x6000, 0x7e00,
0x6000, 0x6000, 0x6000, 0x6000, 0x7f00, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1e00, 0x3300,
0x7f00, 0x6000, 0x6000, 0x6000, 0x7e00, 0x6000, 0x6000, 0x6000, 0x6000, 0x7f00, 0x0000,
0x0000, 0x0000, 0x3300, 0x3300, 0x0000, 0x7f00, 0x6000, 0x6000, 0x6000, 0x7e00, 0x6000,
0x6000, 0x6000, 0x6000, 0x7f00, 0x0000, 0x0000, 0x0000, 0xc000, 0x6000, 0x0000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x3000, 0x6000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x6000, 0x9000, 0x0000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x9000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7e00, 0x6300, 0x6180,
0x6180, 0xf980, 0x6180, 0x6180, 0x6180, 0x6300, 0x7e00, 0x0000, 0x0000, 0x0000, 0x3d00,
0x2f00, 0x0000, 0x6180, 0x7180, 0x7980, 0x7980, 0x6d80, 0x6d80, 0x6780, 0x6780, 0x6380,
0x6180, 0x0000, 0x0000, 0x0000, 0x1c00, 0x0c00, 0x0600, 0x1e00, 0x3300, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x0e00, 0x0c00,
0x1800, 0x1e00, 0x3300, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00,
0x0000, 0x0000, 0x0000, 0x0c00, 0x1e00, 0x3300, 0x1e00, 0x3300, 0x6180, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x3d00, 0x2f00, 0x0000,
0x1e00, 0x3300, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000,
0x0000, 0x0000, 0x3300, 0x3300, 0x0000, 0x1e00, 0x3300, 0x6180, 0x6180, 0x6180, 0x6180,
0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6600, 0x3c00, 0x1800, 0x3c00, 0x6600, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1f80, 0x3300, 0x6380, 0x6780, 0x6d80, 0x6d80, 0x7980,
0x7180, 0x3300, 0x7e00, 0x0000, 0x0000, 0x0000, 0x1c00, 0x0c00, 0x0600, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000,
0x0e00, 0x0c00, 0x1800, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180,
0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x0c00, 0x1e00, 0x3300, 0x0000, 0x6180, 0x6180,
0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300, 0x1e00, 0x0000, 0x0000, 0x0000, 0x3300,
0x3300, 0x0000, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x6180, 0x3300,
0x1e00, 0x0000, 0x0000, 0x0000, 0x0e00, 0x0c00, 0x1800, 0xc0c0, 0xc0c0, 0x6180, 0x3300,
0x1e00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x6000, 0x6000, 0x7e00, 0x6300, 0x6300, 0x6300, 0x6300, 0x7e00, 0x6000, 0x6000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6600, 0x6c00,
0x6600, 0x6600, 0x6600, 0x6600, 0x6c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3800,
0x1800, 0x0c00, 0x0000, 0x3c00, 0x6600, 0x1e00, 0x3600, 0x6600, 0x6600, 0x3e00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x1c00, 0x1800, 0x3000, 0x0000, 0x3c00, 0x6600, 0x1e00,
0x3600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00,
0x6600, 0x0000, 0x3c00, 0x6600, 0x1e00, 0x3600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7a00, 0x5e00, 0x0000, 0x3c00, 0x6600, 0x1e00, 0x3600,
0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600,
0x0000, 0x3c00, 0x6600, 0x1e00, 0x3600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1800, 0x3c00, 0x1800, 0x0000, 0x3c00, 0x6600, 0x1e00, 0x3600, 0x6600,
0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3fc0, 0x6660, 0x1fe0, 0x3600, 0x6600, 0x6660, 0x3fc0, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x6600, 0x6000, 0x6000, 0x6000, 0x6600,
0x3c00, 0x1800, 0x0c00, 0x3800, 0x0000, 0x0000, 0x3800, 0x1800, 0x0c00, 0x0000, 0x3c00,
0x6600, 0x7e00, 0x6000, 0x6000, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1c00, 0x1800, 0x3000, 0x0000, 0x3c00, 0x6600, 0x7e00, 0x6000, 0x6000, 0x6600, 0x3c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800, 0x3c00, 0x6600, 0x0000, 0x3c00, 0x6600,
0x7e00, 0x6000, 0x6000, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6600, 0x6600, 0x0000, 0x3c00, 0x6600, 0x7e00, 0x6000, 0x6000, 0x6600, 0x3c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xe000, 0x6000, 0x3000, 0x0000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x6000,
0xc000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x6000, 0xf000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x6000,
0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x9000, 0x9000,
0x0000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7600, 0x1800, 0x6c00, 0x0c00, 0x3e00, 0x6600, 0x6600, 0x6600,
0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7a00, 0x5e00, 0x0000,
0x7c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3800, 0x1800, 0x0c00, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600,
0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1c00, 0x1800, 0x3000, 0x0000, 0x3c00,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1800, 0x3c00, 0x6600, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7a00, 0x5e00, 0x0000, 0x3c00, 0x6600,
0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x6600, 0x6600, 0x0000, 0x3c00, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3c00, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x3000, 0x0000, 0x7800,
0x0000, 0x3000, 0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
```

```
0x0000, 0x0000, 0x3e00, 0x6e00, 0x6e00, 0x6600, 0x7600, 0x7600, 0x7c00, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x3800, 0x1800, 0x0c00, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1c00, 0x1800, 0x3000,
0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x1800, 0x3c00, 0x6600, 0x0000, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600,
0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6600, 0x6600, 0x0000,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x3e00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x1c00, 0x1800, 0x3000, 0x0000, 0xc300, 0xc300, 0x6600, 0x6600, 0x3c00, 0x3c00,
0x1800, 0x1800, 0x3000, 0x6000, 0x0000, 0x0000, 0x0000, 0x6000, 0x6000, 0x6000, 0x7c00,
0x6600, 0x6600, 0x6600, 0x6600, 0x6600, 0x7c00, 0x6000, 0x6000, 0x6000, 0x0000, 0x0000,
0x0000, 0x6600, 0x6600, 0x0000, 0xc300, 0xc300, 0x6600, 0x6600, 0x3c00, 0x3c00, 0x1800,
0x1800, 0x3000, 0x6000, };
```

File

lpc_winfreesystem14x16.c (see page 185)

Description

This is variable winfreesystem14x16_bits.

1.4.30 winfreesystem14x16_width

```
static UNS_8 winfreesystem14x16_width[] = { 4, 4, 6, 8, 8, 11, 9, 4, 4, 4, 6, 8, 4, 4, 4,
4, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 4, 4, 8, 8, 8, 8, 14, 8, 10, 9, 10, 9, 8, 10, 10, 4, 7, 9,
8, 12, 10, 10, 9, 10, 10, 9, 8, 10, 8, 14, 9, 10, 9, 4, 4, 4, 5, 8, 5, 8, 8, 7, 8, 8, 4, 8,
8, 4, 4, 7, 4, 12, 8, 8, 8, 8, 5, 8, 4, 8, 8, 10, 8, 8, 8, 5, 4, 5, 5, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 9, 4, 8, 8,
8, 8, 4, 8, 5, 10, 5, 7, 8, 4, 10, 8, 5, 8, 4, 4, 5, 8, 7, 4, 5, 4, 5, 7, 11, 11, 11, 8, 8,
8, 8, 8, 8, 8, 13, 9, 9, 9, 9, 9, 4, 4, 4, 4, 10, 10, 10, 10, 10, 10, 10, 8, 10, 10, 10,
10, 10, 10, 9, 8, 8, 8, 8, 8, 8, 8, 8, 12, 7, 8, 8, 8, 8, 4, 4, 4, 4, 8, 8, 8, 8, 8, 8, 8, 6,
8, 8, 8, 8, 8, 8, 8, 8, };
```

File

lpc_winfreesystem14x16.c (see page 185)

Description

Character width data.

1.4.31 x5x7_bits

```
static UNS_16 x5x7_bits[] = { 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0xf000, 0x0000,
0x0000, 0x2000, 0x7000, 0xf800, 0x7000, 0x2000, 0x0000, 0x5000, 0xa000, 0x5000, 0xa000,
0x5000, 0xa000, 0x0000, 0xa000, 0xe000, 0xa000, 0xa000, 0x7000, 0x2000, 0x2000, 0xc000,
0x8000, 0xc000, 0xb000, 0x2000, 0x3000, 0x2000, 0xc000, 0x8000, 0xc000, 0x6000, 0x5000,
0x6000, 0x5000, 0x8000, 0x8000, 0xc000, 0x3000, 0x2000, 0x3000, 0x2000, 0x2000, 0x5000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x7000, 0x2000, 0x0000, 0x7000, 0x0000,
0x0000, 0x9000, 0xd000, 0xb000, 0x9000, 0x2000, 0x2000, 0x3000, 0xa000, 0xa000, 0xa000,
0x4000, 0x7000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0xe000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xe000, 0x2000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0x3800,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x3800, 0x0000, 0x0000, 0x0000, 0x2000,
0x2000, 0x2000, 0xf800, 0x2000, 0x2000, 0x2000, 0x0000, 0xf800, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0000, 0x2000, 0x2000, 0x2000,
0x3800, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0xe000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0x2000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x1000,
0x2000, 0x4000, 0x2000, 0x1000, 0x7000, 0x0000, 0x4000, 0x2000, 0x1000, 0x2000, 0x4000,
0x7000, 0x0000, 0x0000, 0x0000, 0x7000, 0x5000, 0x5000, 0x5000, 0x0000, 0x0000, 0x1000,
0x7000, 0x2000, 0x7000, 0x4000, 0x0000, 0x0000, 0x3000, 0x4000, 0xe000, 0x4000, 0xb000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000, 0x2000, 0x0000,
```

```
  0x5000, 0x5000, 0x5000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5000, 0xf800, 0x5000,
  0xf800, 0x5000, 0x0000, 0x0000, 0x7000, 0xa000, 0x7000, 0x2800, 0x7000, 0x0000, 0x8000,
  0x9000, 0x2000, 0x4000, 0x9000, 0x1000, 0x0000, 0x0000, 0x4000, 0xa000, 0x4000, 0xa000,
  0x5000, 0x0000, 0x6000, 0x4000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x4000,
  0x4000, 0x4000, 0x4000, 0x2000, 0x0000, 0x4000, 0x2000, 0x2000, 0x2000, 0x2000, 0x4000,
  0x0000, 0x0000, 0xa000, 0x4000, 0xe000, 0x4000, 0xa000, 0x0000, 0x0000, 0x2000, 0x2000,
  0xf800, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x6000, 0x4000, 0x8000,
  0x0000, 0x0000, 0x0000, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
  0x6000, 0x6000, 0x0000, 0x0000, 0x1000, 0x2000, 0x4000, 0x8000, 0x0000, 0x0000, 0x4000,
  0xa000, 0xa000, 0xa000, 0xa000, 0x4000, 0x0000, 0x4000, 0xc000, 0x4000, 0x4000, 0x4000,
  0xe000, 0x0000, 0x6000, 0x9000, 0x1000, 0x2000, 0x4000, 0xf000, 0x0000, 0xf000, 0x1000,
  0x6000, 0x1000, 0x9000, 0x6000, 0x0000, 0x2000, 0x6000, 0xa000, 0xf000, 0x2000, 0x2000,
  0x0000, 0xf000, 0x8000, 0xe000, 0x1000, 0x9000, 0x6000, 0x0000, 0x6000, 0x8000, 0xe000,
  0x9000, 0x9000, 0x6000, 0x0000, 0xf000, 0x1000, 0x2000, 0x2000, 0x4000, 0x4000, 0x0000,
  0x6000, 0x9000, 0x6000, 0x9000, 0x9000, 0x6000, 0x0000, 0x6000, 0x9000, 0x9000, 0x7000,
  0x1000, 0x6000, 0x0000, 0x0000, 0x6000, 0x6000, 0x0000, 0x6000, 0x6000, 0x0000, 0x0000,
  0x6000, 0x6000, 0x0000, 0x6000, 0x4000, 0x8000, 0x0000, 0x2000, 0x4000, 0x8000, 0x4000,
  0x2000, 0x0000, 0x0000, 0x0000, 0xf000, 0x0000, 0xf000, 0x0000, 0x0000, 0x0000, 0x8000,
  0x4000, 0x2000, 0x4000, 0x8000, 0x0000, 0x4000, 0xa000, 0x2000, 0x4000, 0x0000, 0x4000,
  0x0000, 0x6000, 0x9000, 0xb000, 0xb000, 0x8000, 0x6000, 0x0000, 0x6000, 0x9000, 0x9000,
  0xf000, 0x9000, 0x9000, 0x0000, 0xe000, 0x9000, 0xe000, 0x9000, 0x9000, 0xe000, 0x0000,
  0x6000, 0x9000, 0x8000, 0x8000, 0x9000, 0x6000, 0x0000, 0xe000, 0x9000, 0x9000, 0x9000,
  0x9000, 0xe000, 0x0000, 0xf000, 0x8000, 0xe000, 0x8000, 0x8000, 0xf000, 0x0000, 0xf000,
  0x8000, 0xe000, 0x8000, 0x8000, 0x8000, 0x0000, 0x6000, 0x9000, 0x8000, 0xb000, 0x9000,
  0x7000, 0x0000, 0x9000, 0x9000, 0xf000, 0x9000, 0x9000, 0x9000, 0x0000, 0xe000, 0x4000,
  0x4000, 0x4000, 0x4000, 0xe000, 0x0000, 0x1000, 0x1000, 0x1000, 0x1000, 0x9000, 0x6000,
  0x0000, 0x9000, 0xa000, 0xc000, 0xc000, 0xa000, 0x9000, 0x0000, 0x8000, 0x8000, 0x8000,
  0x8000, 0x8000, 0xf000, 0x0000, 0x9000, 0xf000, 0xf000, 0x9000, 0x9000, 0x9000, 0x0000,
  0x9000, 0xd000, 0xd000, 0xb000, 0xb000, 0x9000, 0x0000, 0x6000, 0x9000, 0x9000, 0x9000,
  0x9000, 0x6000, 0x0000, 0xe000, 0x9000, 0x9000, 0xe000, 0x8000, 0x8000, 0x0000, 0x6000,
  0x9000, 0x9000, 0x9000, 0xd000, 0x6000, 0x1000, 0xe000, 0x9000, 0x9000, 0xe000, 0xa000,
  0x9000, 0x0000, 0x6000, 0x9000, 0x4000, 0x2000, 0x9000, 0x6000, 0x0000, 0xe000, 0x4000,
  0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x9000, 0x9000, 0x9000, 0x9000, 0x9000, 0x6000,
  0x0000, 0x9000, 0x9000, 0x9000, 0x9000, 0x6000, 0x6000, 0x0000, 0x9000, 0x9000, 0x9000,
  0xf000, 0xf000, 0x9000, 0x0000, 0x9000, 0x9000, 0x6000, 0x6000, 0x9000, 0x9000, 0x0000,
  0xa000, 0xa000, 0xa000, 0x4000, 0x4000, 0x4000, 0x0000, 0xf000, 0x1000, 0x2000, 0x4000,
  0x8000, 0xf000, 0x0000, 0xe000, 0x8000, 0x8000, 0x8000, 0x8000, 0xe000, 0x0000, 0x0000,
  0x8000, 0x4000, 0x2000, 0x1000, 0x0000, 0x0000, 0xe000, 0x2000, 0x2000, 0x2000, 0x2000,
  0xe000, 0x0000, 0x4000, 0xa000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
  0x0000, 0x0000, 0x0000, 0xf000, 0x0000, 0xc000, 0x4000, 0x2000, 0x0000, 0x0000, 0x0000,
  0x0000, 0x0000, 0x0000, 0x7000, 0x9000, 0xb000, 0x5000, 0x0000, 0x8000, 0x8000, 0xe000,
  0x9000, 0x9000, 0xe000, 0x0000, 0x0000, 0x0000, 0x6000, 0x8000, 0x8000, 0x6000, 0x0000,
  0x1000, 0x1000, 0x7000, 0x9000, 0x9000, 0x7000, 0x0000, 0x0000, 0x0000, 0x6000, 0xb000,
  0xc000, 0x6000, 0x0000, 0x2000, 0x5000, 0x4000, 0xe000, 0x4000, 0x4000, 0x0000, 0x0000,
  0x0000, 0x7000, 0x9000, 0x6000, 0x8000, 0x7000, 0x8000, 0x8000, 0xe000, 0x9000, 0x9000,
  0x9000, 0x0000, 0x4000, 0x0000, 0xc000, 0x4000, 0x4000, 0xe000, 0x0000, 0x2000, 0x0000,
  0x2000, 0x2000, 0x2000, 0xa000, 0x4000, 0x8000, 0x8000, 0xa000, 0xc000, 0xa000, 0x9000,
  0x0000, 0xc000, 0x4000, 0x4000, 0x4000, 0x4000, 0xe000, 0x0000, 0x0000, 0x0000, 0xa000,
  0xf000, 0x9000, 0x9000, 0x0000, 0x0000, 0x0000, 0xe000, 0x9000, 0x9000, 0x9000, 0x0000,
  0x0000, 0x0000, 0x6000, 0x9000, 0x9000, 0x6000, 0x0000, 0x0000, 0x0000, 0xe000, 0x9000,
  0x9000, 0xe000, 0x8000, 0x0000, 0x0000, 0x7000, 0x9000, 0x9000, 0x7000, 0x1000, 0x0000,
  0x0000, 0xe000, 0x9000, 0x8000, 0x8000, 0x0000, 0x0000, 0x0000, 0x7000, 0xc000, 0x3000,
  0xe000, 0x0000, 0x4000, 0x4000, 0xe000, 0x4000, 0x4000, 0x3000, 0x0000, 0x0000, 0x0000,
  0x9000, 0x9000, 0x9000, 0x7000, 0x0000, 0x0000, 0x0000, 0xa000, 0xa000, 0xa000, 0x4000,
  0x0000, 0x0000, 0x0000, 0x9000, 0x9000, 0xf000, 0xf000, 0x0000, 0x0000, 0x0000, 0x9000,
  0x6000, 0x6000, 0x9000, 0x0000, 0x0000, 0x0000, 0x9000, 0x9000, 0x5000, 0x2000, 0x4000,
  0x0000, 0x0000, 0xf000, 0x2000, 0x4000, 0xf000, 0x0000, 0x2000, 0x4000, 0xc000, 0x4000,
  0x4000, 0x2000, 0x0000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000, 0x8000,
  0x4000, 0x6000, 0x4000, 0x4000, 0x8000, 0x0000, 0x5000, 0xa000, 0x0000, 0x0000, 0x0000,
  0x0000, 0x0000, };
```

File

lpc_x5x7.c (see page 186)

Description

Font character bitmap data.

1.4.32 x5x7_width

```
static UNS_8 x5x7_width[] = { 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, };
```

File

lpc_x5x7.c (see page 186)

Description

Character width data.

1.4.33 x6x13_bits

```
static UNS_16 x6x13_bits[] = { 0x0000, 0x7800, 0x7800, 0x7800, 0x7800, 0x7800, 0x7800,
0x7800, 0x7800, 0x7800, 0x7800, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x2000, 0x7000, 0xf800, 0x7000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5400, 0xa800,
0x5400, 0xa800, 0x5400, 0xa800, 0x5400, 0xa800, 0x5400, 0xa800, 0x5400, 0xa800, 0x0000,
0x0000, 0x0000, 0x0000, 0xa000, 0xa000, 0xe000, 0xa000, 0xa000, 0x7000, 0x2000, 0x2000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0xe000, 0x8000, 0xc000, 0x8000, 0xf000, 0x4000,
0x6000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8000, 0x8000, 0x7000,
0x7000, 0x4800, 0x7000, 0x5000, 0x4800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000,
0x8000, 0xe000, 0x7000, 0x4000, 0x6000, 0x4000, 0x4000, 0x0000, 0x0000, 0x6000, 0x9000,
0x9000, 0x6000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x2000, 0x2000, 0xf800, 0x2000, 0x2000, 0x0000, 0xf800, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0xc800, 0xa800, 0x9800, 0x8800, 0x4000, 0x4000,
0x4000, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x5000, 0x2000, 0x0000,
0xf800, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0xe000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xe000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x3c00, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x3c00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0xfc00, 0x2000, 0x2000,
0x2000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0xfc00, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfc00,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0xfc00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xfc00, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xfc00, 0x0000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x3c00, 0x2000,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0xe000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0xfc00, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0xfc00, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0800, 0x1000, 0x2000, 0x4000, 0x2000, 0x1000, 0x0800,
0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x4000, 0x2000, 0x1000, 0x2000,
0x4000, 0x8000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800,
0x5000, 0x5000, 0x5000, 0x5000, 0x9000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0800, 0xf800, 0x2000, 0xf800, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3000, 0x4800, 0x4000, 0x4000, 0xe000, 0x4000, 0x4000, 0x4800, 0xb000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0x0000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x5000, 0x5000, 0x5000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x5000, 0x5000, 0xf800, 0x5000, 0xf800, 0x5000, 0x5000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x2000, 0x7800, 0xa000, 0xa000, 0x7000, 0x2800, 0x2800, 0xf000, 0x2000, 0x0000,
0x0000, 0x0000, 0x0000, 0x4800, 0xa800, 0x5000, 0x1000, 0x2000, 0x4000, 0x5000, 0xa800,
0x9000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0xa000, 0xa000, 0x4000, 0xa000, 0x9800,
```

```
0x9000, 0x6800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x3000, 0x2000, 0x4000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000, 0x2000,
0x2000, 0x4000, 0x4000, 0x4000, 0x2000, 0x2000, 0x1000, 0x0000, 0x0000, 0x0000, 0x0000,
0x4000, 0x2000, 0x2000, 0x1000, 0x1000, 0x1000, 0x2000, 0x2000, 0x4000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2000, 0xa800, 0xf800, 0x7000, 0xf800, 0xa800, 0x2000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x2000, 0xf800, 0x2000, 0x2000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x3000, 0x2000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x7000, 0x2000, 0x0000, 0x0000,
0x0000, 0x0800, 0x0800, 0x1000, 0x1000, 0x2000, 0x4000, 0x4000, 0x8000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2000, 0x5000, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x5000,
0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x6000, 0xa000, 0x2000, 0x2000, 0x2000,
0x2000, 0x2000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8800, 0x0800,
0x1000, 0x2000, 0x4000, 0x8000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0800,
0x1000, 0x2000, 0x7000, 0x0800, 0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000,
0x1000, 0x1000, 0x3000, 0x5000, 0x5000, 0x9000, 0xf800, 0x1000, 0x1000, 0x0000, 0x0000,
0x0000, 0x0000, 0xf800, 0x8000, 0x8000, 0xb000, 0xc800, 0x0800, 0x0800, 0x8800, 0x7000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8000, 0x8000, 0xf000, 0x8800, 0x8800,
0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0800, 0x1000, 0x2000,
0x2000, 0x4000, 0x4000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8800,
0x8800, 0x7000, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000,
0x8800, 0x8800, 0x8800, 0x7800, 0x0800, 0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x2000, 0x7000, 0x2000, 0x0000, 0x0000, 0x2000, 0x7000, 0x2000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x7000, 0x2000, 0x0000, 0x0000, 0x3000,
0x2000, 0x4000, 0x0000, 0x0000, 0x0000, 0x0800, 0x1000, 0x2000, 0x4000, 0x8000, 0x4000,
0x2000, 0x1000, 0x0800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800,
0x0000, 0x0000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x4000,
0x2000, 0x1000, 0x0800, 0x1000, 0x2000, 0x4000, 0x8000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7000, 0x8800, 0x8800, 0x0800, 0x1000, 0x2000, 0x2000, 0x0000, 0x2000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7000, 0x8800, 0x8800, 0x9800, 0xa800, 0xa800, 0xb000, 0x8000, 0x7800,
0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x5000, 0x8800, 0x8800, 0x8800, 0xf800, 0x8800,
0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0xf000, 0x4800, 0x4800, 0x4800, 0x7000,
0x4800, 0x4800, 0x4800, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8000,
0x8000, 0x8000, 0x8000, 0x8000, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf000,
0x4800, 0x4800, 0x4800, 0x4800, 0x4800, 0x4800, 0x4800, 0xf000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf800, 0x8000, 0x8000, 0x8000, 0xf000, 0x8000, 0x8000, 0x8000, 0xf800, 0x0000,
0x0000, 0x0000, 0x0000, 0xf800, 0x8000, 0x8000, 0x8000, 0xf000, 0x8000, 0x8000, 0x8000,
0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8000, 0x8000, 0x8000, 0x9800,
0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x8800, 0x8800,
0xf800, 0x8800, 0x8800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x2000,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000,
0x3800, 0x1000, 0x1000, 0x1000, 0x1000, 0x1000, 0x1000, 0x9000, 0x6000, 0x0000, 0x0000,
0x0000, 0x0000, 0x8800, 0x8800, 0x9000, 0xa000, 0xc000, 0xa000, 0x9000, 0x8800, 0x8800,
0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000, 0x8000,
0x8000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0xd800, 0xa800, 0xa800,
0x8800, 0x8800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0xc800, 0xc800,
0xa800, 0xa800, 0x9800, 0x9800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000,
0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000,
0x0000, 0xf000, 0x8800, 0x8800, 0x8800, 0xf000, 0x8000, 0x8000, 0x8000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0xa800,
0x7000, 0x0800, 0x0000, 0x0000, 0x0000, 0xf000, 0x8800, 0x8800, 0x8800, 0xf000, 0xa000,
0x9000, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8000, 0x8000,
0x7000, 0x0800, 0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x2000,
0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000,
0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000,
0x0000, 0x0000, 0x8800, 0x8800, 0x8800, 0x8800, 0x5000, 0x5000, 0x5000, 0x2000, 0x2000,
0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x8800, 0x8800, 0xa800, 0xa800, 0xa800,
0xd800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x5000, 0x5000, 0x2000,
0x5000, 0x5000, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x5000,
0x5000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800,
0x0800, 0x1000, 0x1000, 0x2000, 0x4000, 0x4000, 0x8000, 0xf800, 0x0000, 0x0000, 0x0000,
0x0000, 0x7000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x4000, 0x7000, 0x0000,
0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0x4000, 0x4000, 0x2000, 0x1000, 0x1000, 0x0800,
0x0800, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x1000, 0x1000, 0x1000, 0x1000, 0x1000,
0x1000, 0x1000, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x5000, 0x8800, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xf800, 0x0000, 0x0000, 0x0000,
0x3000, 0x1000, 0x0800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x0800, 0x7800, 0x8800, 0x8800, 0x7800,
0x0000, 0x0000, 0x0000, 0x0000, 0x8000, 0x8000, 0x8000, 0xf000, 0x8800, 0x8800, 0x8800,
0x8800, 0xf000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800,
```

```
0x8000, 0x8000, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0800, 0x0800, 0x0800,
0x7800, 0x8800, 0x8800, 0x8800, 0x8800, 0x7800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x7000, 0x8800, 0xf800, 0x8000, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000,
0x0000, 0x3000, 0x4800, 0x4000, 0x4000, 0xf000, 0x4000, 0x4000, 0x4000, 0x4000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x8800, 0x8800, 0x7800,
0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x8000, 0x8000, 0x8000, 0xb000, 0xc800, 0x8800,
0x8800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x2000, 0x0000, 0x6000,
0x2000, 0x2000, 0x2000, 0x2000, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x1000,
0x0000, 0x3000, 0x1000, 0x1000, 0x1000, 0x1000, 0x9000, 0x9000, 0x6000, 0x0000, 0x0000,
0x8000, 0x8000, 0x8000, 0x9000, 0xa000, 0xc000, 0xa000, 0x9000, 0x8800, 0x0000, 0x0000,
0x0000, 0x0000, 0x6000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x7000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xd000, 0xa800, 0xa800, 0xa800,
0xa800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xb000, 0xc800,
0x8800, 0x8800, 0x8800, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x7000, 0x8800, 0x8800, 0x8800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0xf000, 0x8800, 0x8800, 0x8800, 0xf000, 0x8000, 0x8000, 0x8000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x7800, 0x8800, 0x8800, 0x8800, 0x7800, 0x0800, 0x0800,
0x0800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0xb000, 0xc800, 0x8000, 0x8000, 0x8000,
0x8000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x7000, 0x8800, 0x6000,
0x1000, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4000, 0x4000, 0xf000,
0x4000, 0x4000, 0x4000, 0x4800, 0x3000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x8800, 0x8800, 0x8800, 0x8800, 0x9800, 0x6800, 0x0000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0x8800, 0x5000, 0x5000, 0x2000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800, 0xa800, 0xa800, 0xa800, 0x5000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x5000, 0x2000, 0x2000,
0x5000, 0x8800, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000, 0x8800, 0x8800,
0x8800, 0x9800, 0x6800, 0x0800, 0x8800, 0x7000, 0x0000, 0x0000, 0x0000, 0x0000, 0x0000,
0xf800, 0x1000, 0x2000, 0x4000, 0x8000, 0xf800, 0x0000, 0x0000, 0x0000, 0x0000, 0x1800,
0x2000, 0x2000, 0x2000, 0xc000, 0x2000, 0x2000, 0x2000, 0x1800, 0x0000, 0x0000, 0x0000,
0x0000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x2000, 0x0000,
0x0000, 0x0000, 0x0000, 0xc000, 0x2000, 0x2000, 0x2000, 0x1800, 0x2000, 0x2000, 0x2000,
0xc000, 0x0000, 0x0000, 0x0000, 0x0000, 0x4800, 0xa800, 0x9000, 0x0000, 0x0000, 0x0000,
0x0000, 0x0000, 0x0000, 0x0000, 0x0000, };
```

File

lpc_x6x13.c (see page 187)

Description

Font character bitmap data.

1.4.34 x6x13_width

```
static UNS_8 x6x13_width[] = { 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,
6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, };
```

File

lpc_x6x13.c (see page 187)

Description

Character width data.

1.5 Macros

1.5.1 _BIT

```
#define _BIT(n) (((UNS_32)(1)) << (n))
```

File

lpc_types.h (see page 183)

Description

Set bit macro

1.5.2 _BITMASK

```
#define _BITMASK(field_width) ( _BIT(field_width) - 1)
```

File

lpc_types.h (see page 183)

Description

Bitmask creation macro

1.5.3 _ERROR

```
#define _ERROR (INT_32)(-1)
```

File

lpc_types.h (see page 183)

Description

ERROR macro

1.5.4 _NO_ERROR

```
#define _NO_ERROR (INT_32)(0)
```

File

lpc_types.h (see page 183)

Description

NO_ERROR macro

1.5.5 _SBF

```
#define _SBF(f,v) (((UNS_32)(v)) << (f))
```

File

lpc_types.h (see page 183)

Description

Set bit field macro

1.5.6 ARM922T_CACHE_CP

```
#define ARM922T_CACHE_CP p15
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: cache

1.5.7 ARM922T_CPT_ENTRIES

```
#define ARM922T_CPT_ENTRIES 256
```

File

lpc_arm922t_arch.h (see page 156)

Description

Number of entries in ARM922T coarse page table

1.5.8 ARM922T_CPT_INDEX_MASK

```
#define ARM922T_CPT_INDEX_MASK (ARM922T_CPT_ENTRIES - 1)
```

File

lpc_arm922t_arch.h (see page 156)

Description

Mask to get the coarse page table index

1.5.9 ARM922T_CPT_SIZE

```
#define ARM922T_CPT_SIZE (ARM922T_CPT_ENTRIES * 4)
```

File

lpc_arm922t_arch.h (see page 156)

Description

Size of the ARM922T coarse page table

1.5.10 ARM922T_FPT_ENTRIES

```
#define ARM922T_FPT_ENTRIES 1024
```

File

lpc_arm922t_arch.h (see page 156)

Description

Number of entries in ARM922T fine page table

1.5.11 ARM922T_FPT_INDEX_MASK

```
#define ARM922T_FPT_INDEX_MASK (ARM922T_FPT_ENTRIES - 1)
```

File

lpc_arm922t_arch.h (see page 156)

Description

Mask to get the fine page table index

1.5.12 ARM922T_FPT_SIZE

```
#define ARM922T_FPT_SIZE (ARM922T_FPT_ENTRIES * 4)
```

File

lpc_arm922t_arch.h (see page 156)

Description

Size of the ARM922T fine page table

1.5.13 ARM922T_L1D_AP_ALL

```
#define ARM922T_L1D_AP_ALL 0x00000C00
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 section all access bit

1.5.14 ARM922T_L1D_AP_SVC_ONLY

```
#define ARM922T_L1D_AP_SVC_ONLY 0x00000400
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 section service access only bit

1.5.15 ARM922T_L1D_AP_USR_RO

```
#define ARM922T_L1D_AP_USR_RO 0x00000800
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 section client read-only access only bit

1.5.16 ARM922T_L1D_BUFFERABLE

```
#define ARM922T_L1D_BUFFERABLE 0x00000004
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 page or section identifier

1.5.17 ARM922T_L1D_CACHEABLE

```
#define ARM922T_L1D_CACHEABLE 0x00000008
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 page or section cacheable bit

1.5.18 ARM922T_L1D_COMP_BIT

```
#define ARM922T_L1D_COMP_BIT 0x00000010
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 page or section bufferable bit

1.5.19 ARM922T_L1D_CP_BASE_ADDR

```
#define ARM922T_L1D_CP_BASE_ADDR(n) _SBF(10, ((n) & 0x003FFFFF))
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 coarse page address load macro

1.5.20 ARM922T_L1D_DOMAIN

```
#define ARM922T_L1D_DOMAIN(n) _SBF(5, ((n) & 0x0F))
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 page or section domain load macro

1.5.21 ARM922T_L1D_FP_BASE_ADDR

```
#define ARM922T_L1D_FP_BASE_ADDR(n) _SBF(12, ((n) & 0x000FFFFF))
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 fine page address load macro

1.5.22 ARM922T_L1D_SN_BASE_ADDR

```
#define ARM922T_L1D_SN_BASE_ADDR(n) _SBF(20, ((n) & 0x00000FFF))
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 section address load macro

1.5.23 ARM922T_L1D_TYPE_CPAGE

```
#define ARM922T_L1D_TYPE_CPAGE 0x00000001
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 coarse page identifier

1.5.24 ARM922T_L1D_TYPE_FAULT

```
#define ARM922T_L1D_TYPE_FAULT 0x00000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

Level 1 Descriptor fields ARM922T MMU level 1 invalid page or section identifier

1.5.25 ARM922T_L1D_TYPE_FPAGE

```
#define ARM922T_L1D_TYPE_FPAGE 0x00000003
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 fine page identifier

1.5.26 ARM922T_L1D_TYPE_PG_SN_MASK

```
#define ARM922T_L1D_TYPE_PG_SN_MASK 0x00000003
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 compatibility bit

1.5.27 ARM922T_L1D_TYPE_SECTION

```
#define ARM922T_L1D_TYPE_SECTION 0x00000002
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 1 1MByte section identifier

1.5.28 ARM922T_L2D_AP0_ALL

```
#define ARM922T_L2D_AP0_ALL 0x00000030
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP0 all access bit

1.5.29 ARM922T_L2D_AP0_SVC_ONLY

```
#define ARM922T_L2D_AP0_SVC_ONLY 0x00000010
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP0 service access only bit

1.5.30 ARM922T_L2D_AP0_USR_RO

```
#define ARM922T_L2D_AP0_USR_RO 0x00000020
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP0 client read-only access only bit

1.5.31 ARM922T_L2D_AP1_ALL

```
#define ARM922T_L2D_AP1_ALL 0x000000C0
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP1 all access bit

1.5.32 ARM922T_L2D_AP1_SVC_ONLY

```
#define ARM922T_L2D_AP1_SVC_ONLY 0x00000040
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP1 service access only bit

1.5.33 ARM922T_L2D_AP1_USR_RO

```
#define ARM922T_L2D_AP1_USR_RO 0x00000080
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP1 client read-only access only bit

1.5.34 ARM922T_L2D_AP2_ALL

```
#define ARM922T_L2D_AP2_ALL _SBF(8,3)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP2 all access bit

1.5.35 ARM922T_L2D_AP2_SVC_ONLY

```
#define ARM922T_L2D_AP2_SVC_ONLY _SBF(8,1)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP2 service access only bit

1.5.36 ARM922T_L2D_AP2_USR_RO

```
#define ARM922T_L2D_AP2_USR_RO _SBF(8,2)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 section AP2 client read-only access only bit

1.5.37 ARM922T_L2D_AP3_ALL

```
#define ARM922T_L2D_AP3_ALL _SBF(10,3)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 3 section AP2 all access bit

1.5.38 ARM922T_L2D_AP3_SVC_ONLY

```
#define ARM922T_L2D_AP3_SVC_ONLY _SBF(10,1)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 3 section AP2 service access only bit

1.5.39 ARM922T_L2D_AP3_USR_RO

```
#define ARM922T_L2D_AP3_USR_RO _SBF(10,2)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 3 section AP2 client read-only access only bit

1.5.40 ARM922T_L2D_BUFFERABLE

```
#define ARM922T_L2D_BUFFERABLE 0x00000004
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 page buffer enable bit

1.5.41 ARM922T_L2D_CACHEABLE

```
#define ARM922T_L2D_CACHEABLE 0x00000008
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 page cache enable bit

1.5.42 ARM922T_L2D_CP_BASE_MASK

```
#define ARM922T_L2D_CP_BASE_MASK 0xFFFFFC00
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 large page address mask

1.5.43 ARM922T_L2D_FP_BASE_MASK

```
#define ARM922T_L2D_FP_BASE_MASK 0xFFFFF000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 fine page address mask

1.5.44 ARM922T_L2D_LPAGE_ADDR

```
#define ARM922T_L2D_LPAGE_ADDR(n) _SBF(16, ((n) & 0x0000FFFF)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 large page address load macro

1.5.45 ARM922T_L2D_LPAGE_MASK

```
#define ARM922T_L2D_LPAGE_MASK 0xFFFF0000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 large page address mask

1.5.46 ARM922T_L2D_SN_BASE_MASK

```
#define ARM922T_L2D_SN_BASE_MASK 0xFFF00000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 large page address mask

1.5.47 ARM922T_L2D_SPAGE_ADDR

```
#define ARM922T_L2D_SPAGE_ADDR(n) _SBF(12, ((n) & 0x000FFFFF)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 small page address load macro

1.5.48 ARM922T_L2D_SPAGE_MASK

```
#define ARM922T_L2D_SPAGE_MASK 0xFFFFF000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 small page address mask

1.5.49 ARM922T_L2D_TPAGE_ADDR

```
#define ARM922T_L2D_TPAGE_ADDR(n) _SBF(10, ((n) & 0x003FFFFF)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 tiny page address load macro

1.5.50 ARM922T_L2D_TPAGE_MASK

```
#define ARM922T_L2D_TPAGE_MASK 0xFFFFFC00
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 tiny page address mask

1.5.51 ARM922T_L2D_TYPE_FAULT

```
#define ARM922T_L2D_TYPE_FAULT 0x00000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

Level 2 Descriptor fields ARM922T MMU level 2 invalid page (fault) identifier

1.5.52 ARM922T_L2D_TYPE_LARGE_PAGE

```
#define ARM922T_L2D_TYPE_LARGE_PAGE 0x00000001
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 large page (fault) identifier

1.5.53 ARM922T_L2D_TYPE_PAGE_MASK

```
#define ARM922T_L2D_TYPE_PAGE_MASK 0x00000003
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 page mask

1.5.54 ARM922T_L2D_TYPE_SMALL_PAGE

```
#define ARM922T_L2D_TYPE_SMALL_PAGE 0x00000002
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 small page (fault) identifier

1.5.55 ARM922T_L2D_TYPE_TINY_PAGE

```
#define ARM922T_L2D_TYPE_TINY_PAGE 0x00000003
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU level 2 tiny page (fault) identifier

1.5.56 ARM922T_MMU_CONTROL_A

```
#define ARM922T_MMU_CONTROL_A 0x00000002
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register alignment fault bit

1.5.57 ARM922T_MMU_CONTROL_ASYNC

```
#define ARM922T_MMU_CONTROL_ASYNC 0xC0000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register 'Asynchronous bus' mode

1.5.58 ARM922T_MMU_CONTROL_BUSMASK

```
#define ARM922T_MMU_CONTROL_BUSMASK 0x3FFFFFFF
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register bus mode mask

1.5.59 ARM922T_MMU_CONTROL_C

```
#define ARM922T_MMU_CONTROL_C 0x00000004
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register data cache bit

1.5.60 ARM922T_MMU_CONTROL_FASTBUS

```
#define ARM922T_MMU_CONTROL_FASTBUS 0x00000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register 'FastBus' mode

1.5.61 ARM922T_MMU_CONTROL_I

```
#define ARM922T_MMU_CONTROL_I 0x00001000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register instruction cache bit

1.5.62 ARM922T_MMU_CONTROL_IA

```
#define ARM922T_MMU_CONTROL_IA 0x80000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register 'Asynchronous Clock Select' bit

1.5.63 ARM922T_MMU_CONTROL_M

```
#define ARM922T_MMU_CONTROL_M 0x00000001
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register MMU enable bit

1.5.64 ARM922T_MMU_CONTROL_NF

```
#define ARM922T_MMU_CONTROL_NF 0x40000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register 'Not FastBus' bit

1.5.65 ARM922T_MMU_CONTROL_R

```
#define ARM922T_MMU_CONTROL_R 0x00000200
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register ROM protection bit

1.5.66 ARM922T_MMU_CONTROL_RR

```
#define ARM922T_MMU_CONTROL_RR 0x00004000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register round robin replacement bit

1.5.67 ARM922T_MMU_CONTROL_S

```
#define ARM922T_MMU_CONTROL_S 0x00000100
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register system protection bit

1.5.68 ARM922T_MMU_CONTROL_SYNC

```
#define ARM922T_MMU_CONTROL_SYNC 0x40000000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register 'Synchronous bus' mode

1.5.69 ARM922T_MMU_CONTROL_V

```
#define ARM922T_MMU_CONTROL_V 0x00002000
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU control register vector relocation bit

1.5.70 ARM922T_MMU_CP

```
#define ARM922T_MMU_CP p15
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: MMU

1.5.71 ARM922T_MMU_DC_SIZE

```
#define ARM922T_MMU_DC_SIZE(n) (((n) >> 18) & 0x7)
```

File

lpc_arm922t_arch.h (see page 156)

Description

- ARM920T/ARM922T MMU Cache type register fields

****

DCache Size

1.5.72 ARM922T_MMU_DN_ACCESS

```
#define ARM922T_MMU_DN_ACCESS(n,m) ((m & 0x3) << ((n) * 2))
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU domain register load macro for domain and access

1.5.73 ARM922T_MMU_DN_CLIENT

```
#define ARM922T_MMU_DN_CLIENT 1
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU domain register 'client access' ID field

1.5.74 ARM922T_MMU_DN_MANAGER

```
#define ARM922T_MMU_DN_MANAGER 3
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU domain register 'all access' ID field

1.5.75 ARM922T_MMU_DN_NONE

```
#define ARM922T_MMU_DN_NONE 0
```

File

lpc_arm922t_arch.h (see page 156)

Description

- MMU Domain access control register fields

****

ARM922T MMU domain register 'no access' ID field

1.5.76 ARM922T_MMU_FSR_DOMAIN

```
#define ARM922T_MMU_FSR_DOMAIN(n) (((n) & 0xF0) >> 4)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU fault status register fault domain load macro

1.5.77 ARM922T_MMU_FSR_TYPE

```
#define ARM922T_MMU_FSR_TYPE(n) ((n) & 0x0F)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM922T MMU fault status register fault type load macro

1.5.78 ARM922T_MMU_IC_SIZE

```
#define ARM922T_MMU_IC_SIZE(n) (((n) >> 6) & 0x7)
```

File

lpc_arm922t_arch.h (see page 156)

Description

ICache Size

1.5.79 ARM922T_MMU_REG_CACHE_LOCKDOWN

```
#define ARM922T_MMU_REG_CACHE_LOCKDOWN c9
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: cache lockdown register

1.5.80 ARM922T_MMU_REG_CACHE_OPS

```
#define ARM922T_MMU_REG_CACHE_OPS c7
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: cache operations register

1.5.81 ARM922T_MMU_REG_CACHE_TYPE

```
#define ARM922T_MMU_REG_CACHE_TYPE c0
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: cache type register

1.5.82 ARM922T_MMU_REG_CONTROL

```
#define ARM922T_MMU_REG_CONTROL c1
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: control register

1.5.83 ARM922T_MMU_REG_DAC

```
#define ARM922T_MMU_REG_DAC c3
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: domain control register

1.5.84 ARM922T_MMU_REG_FAULT_ADDRESS

```
#define ARM922T_MMU_REG_FAULT_ADDRESS c6
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: fault address register

1.5.85 ARM922T_MMU_REG_FAULT_STATUS

```
#define ARM922T_MMU_REG_FAULT_STATUS c5
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: fault status registers

1.5.86 ARM922T_MMU_REG_FSCE_PID

```
#define ARM922T_MMU_REG_FSCE_PID c13
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: FCSE PID register

1.5.87 ARM922T_MMU_REG_ID

```
#define ARM922T_MMU_REG_ID c0
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: ID code register

1.5.88 ARM922T_MMU_REG_TLB_LOCKDOWN

```
#define ARM922T_MMU_REG_TLB_LOCKDOWN c10
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: TLB operations register

1.5.89 ARM922T_MMU_REG_TLB_OPS

```
#define ARM922T_MMU_REG_TLB_OPS c8
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: TLB operations register

1.5.90 ARM922T_MMU_REG_TTB

```
#define ARM922T_MMU_REG_TTB c2
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: translation table base reg

1.5.91 ARM922T_SYS_CONTROL_CP

```
#define ARM922T_SYS_CONTROL_CP p15
```

File

lpc_arm922t_arch.h (see page 156)

Description

ARM and GHS tool coprocessor define: system control

1.5.92 ARM922T_TT_ADDR_MASK

```
#define ARM922T_TT_ADDR_MASK 0xFFFFC000
```

File

lpc_arm922t_arch.h (see page 156)

Description

Level 1 translation table address mask

1.5.93 ARM922T_TT_ENTRIES

```
#define ARM922T_TT_ENTRIES 4096
```

File

lpc_arm922t_arch.h (see page 156)

Description

Number of entries in ARM922T Translation table

1.5.94 ARM922T_TT_SIZE

```
#define ARM922T_TT_SIZE (ARM922T_TT_ENTRIES * 4)
```

File

lpc_arm922t_arch.h (see page 156)

Description

Size of the ARM922T Translation table

1.5.95 ATTB_ARCHIVE

```
#define ATTB_ARCHIVE 0x20
```

File

lpc_fat16.h (see page 166)

Description

Archive bit

1.5.96 ATTB_DIR

```
#define ATTB_DIR 0x10
```

File

lpc_fat16.h (see page 166)

Description

Directory bit

1.5.97 ATTB_HIDDEN

```
#define ATTB_HIDDEN 0x04
```

File

lpc_fat16.h (see page 166)

Description

Hidden file bit

1.5.98 ATTB_LFN

```
#define ATTB_LFN 0x0F
```

File

lpc_fat16.h (see page 166)

Description

LFN entry flag

1.5.99 ATTB_NORMAL

```
#define ATTB_NORMAL 0x00
```

File

lpc_fat16.h (see page 166)

Description

Normal file type (no bits set)

1.5.100 ATTB_RO

```
#define ATTB_RO 0x01
```

File

lpc_fat16.h (see page 166)

Description

Read only bit

1.5.101 ATTB_SYS

```
#define ATTB_SYS 0x02
```

File

lpc_fat16.h (see page 166)

Description

System file bit

1.5.102 ATTB_VOLUME

```
#define ATTB_VOLUME 0x08
```

File

lpc_fat16.h (see page 166)

Description

Volume bit

1.5.103 BI_BITFIELDS

```
#define BI_BITFIELDS 0x00000003
```

File

lpc_bmp.h (see page 161)

Description

Uncomp RGB with sample packing

1.5.104 BI_RGB

```
#define BI_RGB 0x00000000
```

File

lpc_bmp.h (see page 161)

Description

Uncompressed image identifier

1.5.105 BI_RGBA

```
#define BI_RGBA 0x32424752
```

File

lpc_bmp.h (see page 161)

Description

Uncompressed image identifier alias for BI_RGB (see page 123)

1.5.106 BI_RLE4

```
#define BI_RLE4 0x00000002
```

File

lpc_bmp.h (see page 161)

Description

4-bit RLE compression

1.5.107 BI_RLE8

```
#define BI_RLE8 0x00000001
```

File

lpc_bmp.h (see page 161)

Description

8-bit RLE compression

1.5.108 BI_RLE8A

```
#define BI_RLE8A 0x38454C52
```

File

lpc_bmp.h (see page 161)

Description

8-bit RLE compression for BI_RLE8 (see page 123)

1.5.109 BLACK

```
#define BLACK 0x00
```

File

lpc_colors.h (see page 164)

Description

Black color, 323 mode

1.5.110 BLUE

```
#define BLUE 0x03
```

File

lpc_colors.h (see page 164)

Description

Blue color, 323 mode

1.5.111 BLUE_COLORS

```
#define BLUE_COLORS 0x08
```

File

lpc_colors.h (see page 164)

Description

Number of blue colors in 332 mode

1.5.112 BLUEMASK

```
#define BLUEMASK 0x3
```

File

lpc_colors.h (see page 164)

Description

Blue color mask, 323 mode

1.5.113 BLUESHIFT

```
#define BLUESHIFT 0
```

File

lpc_colors.h (see page 164)

Description

Blue shift value, 323 mode

1.5.114 BMP_ID0

```
#define BMP_ID0 'B'
```

File

lpc_bmp.h (see page 161)

Description

BMP file identifier character 1

1.5.115 BMP_ID1

```
#define BMP_ID1 'M'
```

File

lpc_bmp.h (see page 161)

Description

BMP file identifier character 2

1.5.116 BT_SIG_OFS

```
#define BT_SIG_OFS (RSV_OFS + RSV_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro BT_SIG_OFS.

1.5.117 BT_SIG_SZ

```
#define BT_SIG_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro BT_SIG_SZ.

1.5.118 BYTES_SEC_OFS

```
#define BYTES_SEC_OFS (OEMID_OFS + OEMID_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro BYTES_SEC_OFS.

1.5.119 BYTES_SEC_SZ

```
#define BYTES_SEC_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro BYTES_SEC_SZ.

1.5.120 CLUSTER_AV

```
#define CLUSTER_AV 0x0000
```

File

lpc_fat16.h (see page 166)

Description

Cluster available

1.5.121 CLUSTER_BAD

```
#define CLUSTER_BAD 0xFFF7
```

File

lpc_fat16.h (see page 166)

Description

Bad cluster flag

1.5.122 CLUSTER_LAST

```
#define CLUSTER_LAST 0xFFF8
```

File

lpc_fat16.h (see page 166)

Description

Minimum (16-bit) value for last cluster

1.5.123 CLUSTER_MAX

```
#define CLUSTER_MAX 0xFFFF
```

File

lpc_fat16.h (see page 166)

Description

Maximum amount of cluster entries

1.5.124 CLUSTERR_MAX

```
#define CLUSTERR_MAX 0xFFF6
```

File

lpc_fat16.h (see page 166)

Description

Maximum reserved cluster flag

1.5.125 CLUSTERR_MIN

```
#define CLUSTERR_MIN 0xFFF0
```

File

lpc_fat16.h (see page 166)

Description

Minimum reserved cluster flag

1.5.126 CLUSTERU_MAX

```
#define CLUSTERU_MAX 0xFFEF
```

File

lpc_fat16.h (see page 166)

Description

Maximum cluster chain range

1.5.127 CLUSTERU_MIN

```
#define CLUSTERU_MIN 0x0002
```

File

lpc_fat16.h (see page 166)

Description

Minimum cluster chain range

1.5.128 COLORS_DEF

```
#define COLORS_DEF 16
```

File

lpc_colors.h (see page 164)

Description

16-bit 565 color mode #define COLORS_DEF 15 /* 15-bit 555 color mode */ #define COLORS_DEF 12 /* 12-bit 444 color mode */

#define COLORS_DEF 8 /* 8-bit color mode

1.5.129 CYAN

```
#define CYAN (GREEN | BLUE)
```

File

lpc_colors.h (see page 164)

Description

Cyan color, 323 mode

1.5.130 DARKGRAY

```
#define DARKGRAY 0x25
```

File

lpc_colors.h (see page 164)

Description

Dark gray color, 323 mode

1.5.131 DEFAULT_CR_DATE

```
#define DEFAULT_CR_DATE 0x2C21
```

File

lpc_fat16.h (see page 166)

Description

January 1, 2002

1.5.132 DEFAULT_CR_TIME

```
#define DEFAULT_CR_TIME 0xC000
```

File

lpc_fat16.h (see page 166)

Description

12:00:00

1.5.133 DIR_ERASED

```
#define DIR_ERASED 0xE5
```

File

lpc_fat16.h (see page 166)

Description

Erased (free) directory entry

1.5.134 DIR_FREE

```
#define DIR_FREE 0x00
```

File

lpc_fat16.h (see page 166)

Description

Free directory entry

1.5.135 DSIZE

```
#define DSIZE 16
```

File

lpc_fat16.h (see page 166)

Description

Device name string size

1.5.136 DV_NUM_OFS

```
#define DV_NUM_OFS (LG_SECS_OFS + LG_SECS_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro DV_NUM_OFS.

1.5.137 DV_NUM_SZ

```
#define DV_NUM_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro DV_NUM_SZ.

1.5.138 EXTENDED_SIG

```
#define EXTENDED_SIG 0x29
```

File

lpc_fat16.h (see page 166)

Description

FAT16 extended signature

1.5.139 EXTENDED_SIG_IDX

```
#define EXTENDED_SIG_IDX 0x26
```

File

lpc_fat16.h (see page 166)

Description

Extended signature index in data

1.5.140 EXTERN

```
#define EXTERN extern
```

File

lpc_types.h (see page 183)

Description

This is macro EXTERN.

1.5.141 FALSE

```
#define FALSE (0==1)
```

File

lpc_types.h (see page 183)

Description

FALSE macro

1.5.142 FAT_COPY_OFS

```
#define FAT_COPY_OFS (RES_SECT_OFS + RES_SECT_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro FAT_COPY_OFS.

1.5.143 FAT_COPY_SZ

```
#define FAT_COPY_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro FAT_COPY_SZ.

1.5.144 FAT12

```
#define FAT12 0x01
```

File

lpc_fat16.h (see page 166)

Description

Partition type FAT12

1.5.145 FAT16_EXDOS

```
#define FAT16_EXDOS 0x05
```

File

lpc_fat16.h (see page 166)

Description

Partition type extended MSDOS

1.5.146 FAT16_GT32M

```
#define FAT16_GT32M 0x06
```

File

lpc_fat16.h (see page 166)

Description

Partition type FAT16 size more than 32M

1.5.147 FAT16_LT32M

```
#define FAT16_LT32M 0x04
```

File

lpc_fat16.h (see page 166)

Description

Partition type FAT16 size less than 32M

1.5.148 FSNAME_OFS

```
#define FSNAME_OFS (LABEL_OFS + LABEL_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro FSNAME_OFS.

1.5.149 FSNAME_SZ

```
#define FSNAME_SZ 8
```

File

lpc_fat16.c (see page 165)

Description

This is macro FSNAME_SZ.

1.5.150 GREEN

```
#define GREEN 0x1C
```

File

lpc_colors.h (see page 164)

Description

Green color, 323 mode

1.5.151 GREEN_COLORS

```
#define GREEN_COLORS 0x08
```

File

lpc_colors.h (see page 164)

Description

Number of green colors in 332 mode

1.5.152 GREENMASK

```
#define GREENMASK 0x1C
```

File

lpc_colors.h (see page 164)

Description

Green color mask, 323 mode

1.5.153 GREENSHIFT

```
#define GREENSHIFT 2
```

File

lpc_colors.h (see page 164)

Description

Green shift value, 323 mode

1.5.154 HDN_SECS_OFS

```
#define HDN_SECS_OFS (NUM_HDS_OFS + NUM_HDS_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro HDN_SECS_OFS.

1.5.155 HDN_SECS_SZ

```
#define HDN_SECS_SZ 4
```

File

lpc_fat16.c (see page 165)

Description

This is macro HDN_SECS_SZ.

1.5.156 HEAP_HEAD_SIZE

```
#define HEAP_HEAD_SIZE (sizeof (HEAP_DESCRIPTOR_T))
```

File

lpc_heap.c (see page 172)

Description

Heap descriptor size

1.5.157 HEAP_POINTER_NULL

```
#define HEAP_POINTER_NULL ((HEAP_DESCRIPTOR_T *) 0)
```

File

lpc_heap.c (see page 172)

Description

Pointer to NULL (see page 143) heap descriptor

1.5.158 JUMP_OFS

```
#define JUMP_OFS 0
```

File

lpc_fat16.c (see page 165)

Description

Local defines

****

Computed offsets from the unaligned partition header

1.5.159 JUMP_SZ

```
#define JUMP_SZ 3
```

File

lpc_fat16.c (see page 165)

Description

This is macro JUMP_SZ.

1.5.160 LABEL_OFS

```
#define LABEL_OFS (SERNUM_OFS + SERNUM_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro LABEL_OFS.

1.5.161 LABEL_SZ

```
#define LABEL_SZ 11
```

File

lpc_fat16.c (see page 165)

Description

This is macro LABEL_SZ.

1.5.162 LG_SECS_OFS

```
#define LG_SECS_OFS (HDN_SECS_OFS + HDN_SECS_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro LG_SECS_OFS.

1.5.163 LG_SECS_SZ

```
#define LG_SECS_SZ 4
```

File

lpc_fat16.c (see page 165)

Description

This is macro LG_SECS_SZ.

1.5.164 LIGHTBLUE

```
#define LIGHTBLUE 0x01
```

File

lpc_colors.h (see page 164)

Description

Light blue color, 323 mode

1.5.165 LIGHTCYAN

```
#define LIGHTCYAN (LIGHTGREEN | LIGHTBLUE)
```

File

lpc_colors.h (see page 164)

Description

Light cyan color, 323 mode

1.5.166 LIGHTGRAY

```
#define LIGHTGRAY 0x6E
```

File

lpc_colors.h (see page 164)

Description

Light gray color, 323 mode

1.5.167 LIGHTGREEN

```
#define LIGHTGREEN 0x0C
```

File

lpc_colors.h (see page 164)

Description

Light green color, 323 mode

1.5.168 LIGHTMAGENTA

```
#define LIGHTMAGENTA (LIGHTRED | LIGHTBLUE)
```

File

lpc_colors.h (see page 164)

Description

Light magenta color, 323 mode

1.5.169 LIGHTRED

```
#define LIGHTRED 0x60
```

File

lpc_colors.h (see page 164)

Description

Light red color, 323 mode

1.5.170 LIGHTYELLOW

```
#define LIGHTYELLOW (LIGHTRED | LIGHTGREEN)
```

File

lpc_colors.h (see page 164)

Description

Light yellow color, 323 mode

1.5.171 LPC_API_H

```
#define LPC_API_H
```

File

lpc_api.h (see page 155)

Description

This is macro LPC_API_H.

1.5.172 LPC_ARM922T_ARCH_H

```
#define LPC_ARM922T_ARCH_H
```

File

lpc_arm922t_arch.h (see page 156)

Description

This is macro LPC_ARM922T_ARCH_H.

1.5.173 LPC_ARM922T_CP15_DRIVER_H

```
#define LPC_ARM922T_CP15_DRIVER_H
```

File

lpc_arm922t_cp15_driver.h (see page 159)

Description

This is macro LPC_ARM922T_CP15_DRIVER_H.

1.5.174 LPC_BMP_H

```
#define LPC_BMP_H
```

File

lpc_bmp.h (see page 161)

Description

This is macro LPC_BMP_H.

1.5.175 LPC_COLOR_TYPES_H

```
#define LPC_COLOR_TYPES_H
```

File

lpc_colors.h (see page 164)

Description

This is macro LPC_COLOR_TYPES_H.

1.5.176 LPC_FAT16_H

```
#define LPC_FAT16_H
```

File

lpc_fat16.h (see page 166)

Description

This is macro LPC_FAT16_H.

1.5.177 LPC_FAT16_PRIVATE_H

```
#define LPC_FAT16_PRIVATE_H
```

File

lpc_fat16_private.h (see page 169)

Description

This is macro LPC_FAT16_PRIVATE_H.

1.5.178 LPC_FONTS_H

```
#define LPC_FONTS_H
```

File

lpc_fonts.h (see page 171)

Description

This is macro LPC_FONTS_H.

1.5.179 LPC_HEAP_H

```
#define LPC_HEAP_H
```

File

lpc_heap.h (see page 173)

Description

This is macro LPC_HEAP_H.

1.5.180 LPC_HEVR10_FONT_H

```
#define LPC_HEVR10_FONT_H
```

File

lpc_helvr10.h (see page 174)

Description

This is macro LPC_HEVR10_FONT_H.

1.5.181 LPC_ROM8X16_FONT_H

```
#define LPC_ROM8X16_FONT_H
```

File

lpc_rom8x16.h (see page 176)

Description

This is macro LPC_ROM8X16_FONT_H.

1.5.182 LPC_ROM8X8_FONT_H

```
#define LPC_ROM8X8_FONT_H
```

File

lpc_rom8x8.h (see page 177)

Description

This is macro LPC_ROM8X8_FONT_H.

1.5.183 LPC_SHARP_LCD_PARAM_H

```
#define LPC_SHARP_LCD_PARAM_H
```

File

lpc_lcd_params.h (see page 175)

Description

This is macro LPC_SHARP_LCD_PARAM_H.

1.5.184 LPC_SWIM_FONT_H

```
#define LPC_SWIM_FONT_H
```

File

lpc_swim_font.h (see page 180)

Description

This is macro LPC_SWIM_FONT_H.

1.5.185 LPC_SWIM_H

```
#define LPC_SWIM_H
```

File

lpc_swim.h (see page 178)

Description

This is macro LPC_SWIM_H.

1.5.186 LPC_SWIM_IMAGE_H

```
#define LPC_SWIM_IMAGE_H
```

File

lpc_swim_image.h (see page 182)

Description

This is macro LPC_SWIM_IMAGE_H.

1.5.187 LPC_TYPES_H

```
#define LPC_TYPES_H
```

File

lpc_types.h (see page 183)

Description

This is macro LPC_TYPES_H.

1.5.188 LPC_WINFREESYS_14X16_FONT_H

```
#define LPC_WINFREESYS_14X16_FONT_H
```

File

lpc_winfreesystem14x16.h (see page 185)

Description

This is macro LPC_WINFREESYS_14X16_FONT_H.

1.5.189 LPC_X5X7_FONT_H

```
#define LPC_X5X7_FONT_H
```

File

lpc_x5x7.h (see page 186)

Description

This is macro LPC_X5X7_FONT_H.

1.5.190 LPC_X6X13_FONT_H

```
#define LPC_X6X13_FONT_H
```

File

lpc_x6x13.h (see page 187)

Description

This is macro LPC_X6X13_FONT_H.

1.5.191 MAGENTA

```
#define MAGENTA (RED | BLUE)
```

File

lpc_colors.h (see page 164)

Description

Magenta color, 323 mode

1.5.192 MAX_API_DEVS

```
#define MAX_API_DEVS NELEMENTS(api)
```

File

lpc_api.c (see page 154)

Description

Max size of the device table

1.5.193 MAX_API_TABLE

```
#define MAX_API_TABLE (20)
```

File

lpc_api.h (see page 155)

Description

Max number of devices in the subsystem

1.5.194 MEDIA_DES_OFS

```
#define MEDIA_DES_OFS (SMALL_SEC_OFS + SMALL_SEC_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro MEDIA_DES_OFS.

1.5.195 MEDIA_DES_SZ

```
#define MEDIA_DES_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro MEDIA_DES_SZ.

1.5.196 NELEMENTS

```
#define NELEMENTS(array) (sizeof (array) / sizeof (array[0]))
```

File

lpc_types.h (see page 183)

Description

Number of elements in an array

1.5.197 NULL

```
#define NULL ((void*) 0)
```

File

lpc_types.h (see page 183)

Description

NULL pointer

1.5.198 NUM_COLORS

```
#define NUM_COLORS 256
```

File

lpc_colors.h (see page 164)

Description

Number of colors in 332 mode

1.5.199 NUM_HDS_OFS

```
#define NUM_HDS_OFS (SECS_TK_OFS + SECS_TK_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro NUM_HDS_OFS.

1.5.200 NUM_HDS_SZ

```
#define NUM_HDS_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro NUM_HDS_SZ.

1.5.201 OEMID_OFS

```
#define OEMID_OFS (JUMP_OFS + JUMP_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro OEMID_OFS.

1.5.202 OEMID_SZ

```
#define OEMID_SZ 8
```

File

lpc_fat16.c (see page 165)

Description

This is macro OEMID_SZ.

1.5.203 PART_ACTV

```
#define PART_ACTV 0x80
```

File

lpc_fat16.h (see page 166)

Description

Partition active flag bit

1.5.204 PTAB_SIZE

```
#define PTAB_SIZE 512
```

File

lpc_fat16_private.h (see page 169)

Description

Size of MBR and boot records

1.5.205 RED

```
#define RED 0xE0
```

File

lpc_colors.h (see page 164)

Description

Red color, 323 mode

1.5.206 RED_COLORS

```
#define RED_COLORS 0x08
```

File

lpc_colors.h (see page 164)

Description

Number of red colors in 332 mode

1.5.207 REDMASK

```
#define REDMASK 0xE0
```

File

lpc_colors.h (see page 164)

Description

Red color mask, 323 mode

1.5.208 REDSHIFT

```
#define REDSHIFT 5
```

File

lpc_colors.h (see page 164)

Description

Red shift value, 323 mode

1.5.209 RES_SECT_OFS

```
#define RES_SECT_OFS (SECS_CLUS_OFS + SECS_CLUS_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro RES_SECT_OFS.

1.5.210 RES_SECT_SZ

```
#define RES_SECT_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro RES_SECT_SZ.

1.5.211 RGBA

```
#define RGBA 0x41424752
```

File

lpc_bmp.h (see page 161)

Description

Raw RGB with alpha

1.5.212 RGBT

```
#define RGBT 0x54424752
```

File

lpc_bmp.h (see page 161)

Description

Raw RGB with a transparency field

1.5.213 ROOT_ENT_OFS

```
#define ROOT_ENT_OFS (FAT_COPY_OFS + FAT_COPY_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro ROOT_ENT_OFS.

1.5.214 ROOT_ENT_SZ

```
#define ROOT_ENT_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro ROOT_ENT_SZ.

1.5.215 RSV_OFS

```
#define RSV_OFS (DV_NUM_OFS + DV_NUM_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro RSV_OFS.

1.5.216 RSV_SZ

```
#define RSV_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro RSV_SZ.

1.5.217 SECS_CLUS_OFS

```
#define SECS_CLUS_OFS (BYTES_SEC_OFS + BYTES_SEC_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_CLUS_OFS.

1.5.218 SECS_CLUS_SZ

```
#define SECS_CLUS_SZ 1
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_CLUS_SZ.

1.5.219 SECS_FAT_OFS

```
#define SECS_FAT_OFS (MEDIA_DES_OFS + MEDIA_DES_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_FAT_OFS.

1.5.220 SECS_FAT_SZ

```
#define SECS_FAT_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_FAT_SZ.

1.5.221 SECS_TK_OFS

```
#define SECS_TK_OFS (SECS_FAT_OFS + SECS_FAT_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_TK_OFS.

1.5.222 SECS_TK_SZ

```
#define SECS_TK_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro SECS_TK_SZ.

1.5.223 SERNUM_OFS

```
#define SERNUM_OFS (BT_SIG_OFS + BT_SIG_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro SERNUM_OFS.

1.5.224 SERNUM_SZ

```
#define SERNUM_SZ 4
```

File

lpc_fat16.c (see page 165)

Description

This is macro SERNUM_SZ.

1.5.225 SMA_BAD_CLK

```
#define SMA_BAD_CLK (INT_32)(-9)
```

File

lpc_types.h (see page 183)

Description

Bad device clock macro

1.5.226 SMA_BAD_HANDLE

```
#define SMA_BAD_HANDLE (INT_32)(-8)
```

File

lpc_types.h (see page 183)

Description

Bad device handle macro

1.5.227 SMA_BAD_PARAMS

```
#define SMA_BAD_PARAMS (INT_32)(-7)
```

File

lpc_types.h (see page 183)

Description

Device bad paramaters macro

1.5.228 SMA_CANT_START

```
#define SMA_CANT_START (INT_32)(-10)
```

File

lpc_types.h (see page 183)

Description

Device can't start macro

1.5.229 SMA_CANT_STOP

```
#define SMA_CANT_STOP (INT_32)(-11)
```

File

lpc_types.h (see page 183)

Description

Device can't stop macro

1.5.230 SMA_DEV_UNKNOWN

```
#define SMA_DEV_UNKNOWN (INT_32)(-2)
```

File

lpc_types.h (see page 183)

Description

Device unknown macro

1.5.231 SMA_IN_USE

```
#define SMA_IN_USE (INT_32)(-5)
```

File

lpc_types.h (see page 183)

Description

Device in use macro

1.5.232 SMA_NOT_OPEN

```
#define SMA_NOT_OPEN (INT_32)(-4)
```

File

lpc_types.h (see page 183)

Description

Device not open macro

1.5.233 SMA_NOT_SUPPORTED

```
#define SMA_NOT_SUPPORTED (INT_32)(-3)
```

File

lpc_types.h (see page 183)

Description

Device not supported macro

1.5.234 SMA_PIN_CONFLICT

```
#define SMA_PIN_CONFLICT (INT_32)(-6)
```

File

lpc_types.h (see page 183)

Description

Device oin conflict macro

1.5.235 SMALL_SEC_OFS

```
#define SMALL_SEC_OFS (ROOT_ENT_OFS + ROOT_ENT_SZ)
```

File

lpc_fat16.c (see page 165)

Description

This is macro SMALL_SEC_OFS.

1.5.236 SMALL_SEC_SZ

```
#define SMALL_SEC_SZ 2
```

File

lpc_fat16.c (see page 165)

Description

This is macro SMALL_SEC_SZ.

1.5.237 SMALLEST_ENTRY_SIZE

```
#define SMALLEST_ENTRY_SIZE (HEAP_HEAD_SIZE + sizeof (UNS_32))
```

File

lpc_heap.c (see page 172)

Description

Smallest heap descriptor entry

1.5.238 STATIC

```
#define STATIC
```

File

lpc_types.h (see page 183)

Description

External data/function define

1.5.239 SUCCESS

```
#define SUCCESS 0
```

File

lpc_types.h (see page 183)

Description

SUCCESS macro

1.5.240 TRUE

```
#define TRUE (!(FALSE))
```

File

lpc_types.h (see page 183)

Description

TRUE macro

1.5.241 WHITE

```
#define WHITE 0xFF
```

File

lpc_colors.h (see page 164)

Description

White color, 323 mode

1.5.242 YELLOW

```
#define YELLOW (RED | GREEN)
```

File

lpc_colors.h (see page 164)

Description

Yellow color, 323 mode

1.6 Files

1.6.1 lpc_api.c

- $Id:: lpc_api.c 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Standard API

*

- Description:
- This file implements non hardware specific I/O system

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
api_add_device (see page 2)	Function: api_add_device Purpose: To add a device to the api (see page 74) table Processing: This function checks for a device id collision in the api (see page 74) system. If the id is valid it looks for a vacant entry. If the table is not full it binds itself to the api (see page 74) system.
api_find_device (see page 3)	Function: api_find_device Purpose: To find a device using a numerical representation Processing: Search the device table for an id and return return the index of the device in the table.
api_find_empty (see page 3)	Function: api_find_empty Purpose: To find a vacant table entry Processing: Search the device table for a vacant space and return the index in the table.
api_remove_device (see page 3)	Private methods Function: api_remove_device Purpose: To remove a device from the api (see page 74) table Processing: This function finds the table entry that is associated with the devid. Once the entry is found it is cleared which will set it to the idle state. When a table entry is in the idle state a new device my use this entry to bind itself to the system.

Macros

Macro	Description
MAX_API_DEVS (see page 142)	Max size of the device table

Variables

Variable	Description
api (see page 74)	Private io system table
api_is_init (see page 74)	State variable for init

1.6.2 lpc_api.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Standard API

*

- Description:
- This file implements non hardware specific IO system mechanism

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
lpc_api_init (see page 33)	Public APIs used to access device drivers that are registered with the API sub system. Function: lpc_api_init Purpose: To initialize the api (see page 74) system Processing: This function clears the api (see page 74) system table and marks it as initialized. Once the table has been initialized the devices can be bound to the io system and make use of the common API.
lpc_api_register (see page 33)	Function: lpc_api_register Purpose: To register a device with the system Processing: This funtion is used to bind a device to the system. Once bound the device can make use of the common API layer.
lpc_close (see page 34)	Function: lpc_close Purpose: closes a session with an device driver Processing: This routine marks the device as closed and then calls the associated close method at the device driver layer to disable the hardware.
lpc_ioctl (see page 37)	Function: lpc_ioctl Purpose: device io control routine Processing: This routine controls the associated device driver via the callback method that has been bound to a driver. If the device is not registered -1 is returned else return code by the driver ioctl is returned.
lpc_open (see page 38)	Function: lpc_open Purpose: Connects to a system device Processing: This routine calls the associated open method in the io subsystem array. If the device asscoiated with the name is not registered an error -1 is returned. If the device is registered and not already opened a file descriptor that uniquely identifies this device is returned.
lpc_read (see page 39)	Function: lpc_read Purpose: reads data from a registered api (see page 74) system device. Processing: This routine reads data from a registered api (see page 74) device by using the callback method that has been bound to a driver. If the device is not registered -1 is returned. If the device is registered the user can pass in a buffer and a max number of bytes for the driver to use.
lpc_write (see page 39)	Function: lpc_write Purpose: write data to a registered device Processing: This routine writes data to a registered api (see page 74) device by using the callback method that has been bound to a driver. If the device is not registered -1 is returned. If the device is registered a generic pointer and the number of bytes represented by the pointer are being passed to the

Macros

Macro	Description
LPC_API_H (see page 138)	This is macro LPC_API_H.
MAX_API_TABLE (see page 142)	Max number of devices in the subsystem

Structs

Struct	Description
API_S (see page 1)	System API data structure
API_TABLE_S (see page 1)	Api system device lookup table

Types

Type	Description
API_T (see page 57)	System API data structure
API_TABLE_T (see page 57)	Api system device lookup table
PAPI_T (see page 68)	System API data structure
PAPI_TABLE_T (see page 69)	Api system device lookup table

1.6.3 lpc_arm922t_arch.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: General Utilities

*

- Description:
- This file contains constant and macro definitions specific
- to the ARM922T architecture.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
ARM922T_CACHE_CP (see page 101)	ARM and GHS tool coprocessor define: cache
ARM922T_CPT_ENTRIES (see page 101)	Number of entries in ARM922T coarse page table
ARM922T_CPT_INDEX_MASK (see page 101)	Mask to get the coarse page table index
ARM922T_CPT_SIZE (see page 101)	Size of the ARM922T coarse page table
ARM922T_FPT_ENTRIES (see page 102)	Number of entries in ARM922T fine page table
ARM922T_FPT_INDEX_MASK (see page 102)	Mask to get the fine page table index
ARM922T_FPT_SIZE (see page 102)	Size of the ARM922T fine page table
ARM922T_L1D_AP_ALL (see page 102)	ARM922T MMU level 1 section all access bit
ARM922T_L1D_AP_SVC_ONLY (see page 102)	ARM922T MMU level 1 section service access only bit
ARM922T_L1D_AP_USR_RO (see page 103)	ARM922T MMU level 1 section client read-only access only bit

ARM922T_L1D_BUFFERABLE (see page 103)	ARM922T MMU level 1 page or section identifier
ARM922T_L1D_CACHEABLE (see page 103)	ARM922T MMU level 1 page or section cacheable bit
ARM922T_L1D_COMP_BIT (see page 103)	ARM922T MMU level 1 page or section bufferable bit
ARM922T_L1D_CP_BASE_ADDR (see page 104)	ARM922T MMU level 1 coarse page address load macro
ARM922T_L1D_DOMAIN (see page 104)	ARM922T MMU level 1 page or section domain load macro
ARM922T_L1D_FP_BASE_ADDR (see page 104)	ARM922T MMU level 1 fine page address load macro
ARM922T_L1D_SN_BASE_ADDR (see page 104)	ARM922T MMU level 1 section address load macro
ARM922T_L1D_TYPE_CPAGE (see page 104)	ARM922T MMU level 1 coarse page identifier
ARM922T_L1D_TYPE_FAULT (see page 105)	Level 1 Descriptor fields ARM922T MMU level 1 invalid page or section identifier
ARM922T_L1D_TYPE_FPAGE (see page 105)	ARM922T MMU level 1 fine page identifier
ARM922T_L1D_TYPE_PG_SN_MASK (see page 105)	ARM922T MMU level 1 compatibility bit
ARM922T_L1D_TYPE_SECTION (see page 105)	ARM922T MMU level 1 1MByte section identifier
ARM922T_L2D_AP0_ALL (see page 106)	ARM922T MMU level 2 section AP0 all access bit
ARM922T_L2D_AP0_SVC_ONLY (see page 106)	ARM922T MMU level 2 section AP0 service access only bit
ARM922T_L2D_AP0_USR_RO (see page 106)	ARM922T MMU level 2 section AP0 client read-only access only bit
ARM922T_L2D_AP1_ALL (see page 106)	ARM922T MMU level 2 section AP1 all access bit
ARM922T_L2D_AP1_SVC_ONLY (see page 106)	ARM922T MMU level 2 section AP1 service access only bit
ARM922T_L2D_AP1_USR_RO (see page 107)	ARM922T MMU level 2 section AP1 client read-only access only bit
ARM922T_L2D_AP2_ALL (see page 107)	ARM922T MMU level 2 section AP2 all access bit
ARM922T_L2D_AP2_SVC_ONLY (see page 107)	ARM922T MMU level 2 section AP2 service access only bit
ARM922T_L2D_AP2_USR_RO (see page 107)	ARM922T MMU level 2 section AP2 client read-only access only bit
ARM922T_L2D_AP3_ALL (see page 108)	ARM922T MMU level 3 section AP2 all access bit
ARM922T_L2D_AP3_SVC_ONLY (see page 108)	ARM922T MMU level 3 section AP2 service access only bit
ARM922T_L2D_AP3_USR_RO (see page 108)	ARM922T MMU level 3 section AP2 client read-only access only bit
ARM922T_L2D_BUFFERABLE (see page 108)	ARM922T MMU level 2 page buffer enable bit
ARM922T_L2D_CACHEABLE (see page 108)	ARM922T MMU level 2 page cache enable bit
ARM922T_L2D_CP_BASE_MASK (see page 109)	ARM922T MMU level 2 large page address mask
ARM922T_L2D_FP_BASE_MASK (see page 109)	ARM922T MMU level 2 fine page address mask
ARM922T_L2D_LPAGE_ADDR (see page 109)	ARM922T MMU level 2 large page address load macro
ARM922T_L2D_LPAGE_MASK (see page 109)	ARM922T MMU level 2 large page address mask
ARM922T_L2D_SN_BASE_MASK (see page 110)	ARM922T MMU level 2 large page address mask
ARM922T_L2D_SPAGE_ADDR (see page 110)	ARM922T MMU level 2 small page address load macro
ARM922T_L2D_SPAGE_MASK (see page 110)	ARM922T MMU level 2 small page address mask
ARM922T_L2D_TPAGE_ADDR (see page 110)	ARM922T MMU level 2 tiny page address load macro
ARM922T_L2D_TPAGE_MASK (see page 110)	ARM922T MMU level 2 tiny page address mask
ARM922T_L2D_TYPE_FAULT (see page 111)	Level 2 Descriptor fields ARM922T MMU level 2 invalid page (fault) identifier
ARM922T_L2D_TYPE_LARGE_PAGE (see page 111)	ARM922T MMU level 2 large page (fault) identifier
ARM922T_L2D_TYPE_PAGE_MASK (see page 111)	ARM922T MMU level 2 page mask
ARM922T_L2D_TYPE_SMALL_PAGE (see page 111)	ARM922T MMU level 2 small page (fault) identifier
ARM922T_L2D_TYPE_TINY_PAGE (see page 112)	ARM922T MMU level 2 tiny page (fault) identifier
ARM922T_MMU_CONTROL_A (see page 112)	ARM922T MMU control register alignment fault bit
ARM922T_MMU_CONTROL_ASYNC (see page 112)	ARM922T MMU control register 'Asynchronous bus' mode
ARM922T_MMU_CONTROL_BUSMASK (see page 112)	ARM922T MMU control register bus mode mask
ARM922T_MMU_CONTROL_C (see page 112)	ARM922T MMU control register data cache bit
ARM922T_MMU_CONTROL_FASTBUS (see page 113)	ARM922T MMU control register 'FastBus' mode
ARM922T_MMU_CONTROL_I (see page 113)	ARM922T MMU control register instruction cache bit
ARM922T_MMU_CONTROL_IA (see page 113)	ARM922T MMU control register 'Asynchronous Clock Select' bit
ARM922T_MMU_CONTROL_M (see page 113)	ARM922T MMU control register MMU enable bit
ARM922T_MMU_CONTROL_NF (see page 114)	ARM922T MMU control register 'Not FastBus' bit
ARM922T_MMU_CONTROL_R (see page 114)	ARM922T MMU control register ROM protection bit
ARM922T_MMU_CONTROL_RR (see page 114)	ARM922T MMU control register round robin replacement bit
ARM922T_MMU_CONTROL_S (see page 114)	ARM922T MMU control register system protection bit
ARM922T_MMU_CONTROL_SYNC (see page 114)	ARM922T MMU control register 'Synchronous bus' mode
ARM922T_MMU_CONTROL_V (see page 115)	ARM922T MMU control register vector relocation bit
ARM922T_MMU_CP (see page 115)	ARM and GHS tool coprocessor define: MMU
ARM922T_MMU_DC_SIZE (see page 115)	
ARM922T_MMU_DN_ACCESS (see page 115)	ARM922T MMU domain register load macro for domain and access
ARM922T_MMU_DN_CLIENT (see page 116)	ARM922T MMU domain register 'client access' ID field
ARM922T_MMU_DN_MANAGER (see page 116)	ARM922T MMU domain register 'all access' ID field

ARM922T_MMU_DN_NONE (see page 116)	
ARM922T_MMU_FSR_DOMAIN (see page 116)	ARM922T MMU fault status register fault domain load macro
ARM922T_MMU_FSR_TYPE (see page 117)	ARM922T MMU fault status register fault type load macro
ARM922T_MMU_IC_SIZE (see page 117)	ICache Size
ARM922T_MMU_REG_CACHE_LOCKDOWN (see page 117)	ARM and GHS tool coprocessor define: cache lockdown register
ARM922T_MMU_REG_CACHE_OPS (see page 117)	ARM and GHS tool coprocessor define: cache operations register
ARM922T_MMU_REG_CACHE_TYPE (see page 117)	ARM and GHS tool coprocessor define: cache type register
ARM922T_MMU_REG_CONTROL (see page 118)	ARM and GHS tool coprocessor define: control register
ARM922T_MMU_REG_DAC (see page 118)	ARM and GHS tool coprocessor define: domain control register
ARM922T_MMU_REG_FAULT_ADDRESS (see page 118)	ARM and GHS tool coprocessor define: fault address register
ARM922T_MMU_REG_FAULT_STATUS (see page 118)	ARM and GHS tool coprocessor define: fault status registers
ARM922T_MMU_REG_FSCE_PID (see page 119)	ARM and GHS tool coprocessor define: FCSE PID register
ARM922T_MMU_REG_ID (see page 119)	ARM and GHS tool coprocessor define: ID code register
ARM922T_MMU_REG_TLB_LOCKDOWN (see page 119)	ARM and GHS tool coprocessor define: TLB operations register
ARM922T_MMU_REG_TLB_OPS (see page 119)	ARM and GHS tool coprocessor define: TLB operations register
ARM922T_MMU_REG_TTB (see page 119)	ARM and GHS tool coprocessor define: translation table base reg
ARM922T_SYS_CONTROL_CP (see page 120)	ARM and GHS tool coprocessor define: system control
ARM922T_TT_ADDR_MASK (see page 120)	Level 1 translation table address mask
ARM922T_TT_ENTRIES (see page 120)	Number of entries in ARM922T Translation table
ARM922T_TT_SIZE (see page 120)	Size of the ARM922T Translation table
LPC_ARM922T_ARCH_H (see page 138)	This is macro LPC_ARM922T_ARCH_H.

1.6.4 lpc_arm922t_cp15_driver.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: ARM922T Coprocessor 15 driver

*

- Description:
- This file contains driver support for the MMU and cache
- coprocessor (15) of the ARM922T.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Variables

Variable	Description
virtual_tlb_addr (see page 90)	The address translation functions of this driver require a saved pointer to the virtual base address of the MMU table.

1.6.5 lpc_arm922t_cp15_driver.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: ARM922T Coprocessor 15 driver

*

- Description:
- This file contains driver support for the MMU and cache
- coprocessor (15) of the ARM922T.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
cp15_dcache_flush (see page 7)	Force an data cache flush Function: cp15_dcache_flush Purpose: Force an data cache flush Processing: Flush each data cache entry using the segment/index method.
cp15_force_cache_coherence (see page 8)	Force cache coherence between memory and cache for the selected address range Function: cp15_force_cache_coherence Purpose: Force the CPU to recognize the block of code that was just written to memory between start_adr and end_adr even if caching and write buffering is on. Processing: Cache lines are 32-bytes (8 words); clean and invalidate each line of D-cache and invalidate each line of I-cache within the address range. Invalidate the I-TLB within the the address range. The I-TLB has 256 word granularity.
cp15_get_mmu_control_reg (see page 8)	Return the current value of MMU Coprocessor(CP15) Control register Function: cp15_get_mmu_control_reg Purpose: To return the current value of the MMU Coprocessor (CP15) Control register. Processing: Fetch the MMU control register to a variable and return it
cp15_get_ttb (see page 9)	Return the physical address of the MMU translation table Function: cp15_get_ttb Purpose: Return the physical address of the MMU translation table Processing: Read the TTB register from coprocessor 15 and return it to the caller.
cp15_init_mmu_trans_table (see page 9)	Setup MMU page tables Function: cp15_init_mmu_trans_table Purpose: Initializes the MMU page table Processing: Return error if MMU is enabled. Return error if target Translation Table address is not 16K aligned. Clear the Translation Table area. Build the Translation Table from the initialization data in the Section Block array. Return no error.
cp15_invalidate_cache (see page 10)	Invalidates the Instruction and Data caches Function: cp15_invalidate_cache Purpose: Invalidates the Instruction and Data caches Processing: Use the ARM instruction to unconditionally invalidate the entire cache.

cp15_invalidate_tlb (see page 10)	Invalidates the Translation Lookaside Buffers Function: cp15_invalidate_tlb Purpose: Invalidates the Translation Lookaside Buffers Processing: Use the ARM instruction to unconditionally invalidate the I- and D- TLBs.
cp15_map_physical_to_virtual (see page 10)	Get a virtual address from a passed physical address Function: cp15_map_physical_to_virtual Purpose: Return a virtual address for a passed physical address Processing: Test if MMU is on, return if not. Search for the virtual address of the provided physical address. If found, return a void pointer to virtual address.
cp15_map_virtual_to_physical (see page 11)	Return a physical address for a passed virtual address Function: cp15_map_virtual_to_physical Purpose: Return a physical address for a passed virtual address Processing: Return (UNS_32 (see page 73))addr if MMU is turned off. Otherwise, read the address of the translation table from the translation table base address register. Use the upper 12 bits of the addr to index the translation table and read out the descriptor. If the descriptor is invalid, return 0. If the descriptor is for a 1 Meg section, read back the upper 12 bits of the physical address. The lower 20 bits of the physical address is the lower 20 bits of the virtual address. If the descriptor is for a coarse page table, read the coarse page table descriptor and use the most significant 22 bits as the base address of the page table. If the descriptor is for a fine page table, read the fine page table descriptor and use the most significant 20 bits as the base address of the page table. If not a section base, read the level 2 page descriptor from the page table. If bits 1..0 of the level2 descriptor are 01, then it is a large page table descriptor. The most significant 16 bits of the descriptor are the most significant 16 bits of the physical address; the least significant 16-bits of the virtual address are the least significant 16-bits of the address. If bits 1..0 of the level2 descriptor are 10, then it is a small page table descriptor. The most significant 20 bits of the level2 descriptor are the most significant 20 bits of the physical address; the least significant 12 bits are the least significant 12 bits of the physical address. If bits 1..0 of the level2 descriptor are 11, then it is a tiny page table descriptor. The most significant 22 bits of the level2 descriptor are the most significant 22 bits of the physical address; the least significant 10 bits are the least significant 10 bits of the physical address. If bits 1..0 of the level2 descriptor are 0, return 0 (invalid).
cp15_mmu_enabled (see page 12)	Checks to see if the MMU is enabled Function: cp15_mmu_enabled Purpose: Checks to see if the MMU is enabled Processing: Read the MMU control register and check if the MMU enable bit (bit 0) is set.
cp15_set_dcache (see page 12)	Enables or disables the data cache Function: cp15_set_dcache Purpose: Enables or disables the data cache Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the D-cache enable bit, otherwise, clear it. Write the resultant value back to the control register.
cp15_set_domain_access (see page 13)	Define the access permissions for the 16 MMU domains. Function: cp15_set_domain_access Purpose: Define the access permissions for the 16 MMU domains. Processing: Use the ARM instruction to write the value passed as argument to the domain access control regsiter.
cp15_set_icache (see page 13)	Enables or disables the instruction cache Function: cp15_set_icache Purpose: Enables or disables the instruction cache Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the I-cache enable bit, otherwise, clear it. Write the resultant value back to the control register.
cp15_set_mmu (see page 14)	Enable/Disable MMU Function: cp15_set_mmu Purpose: To enable or disable the MMU as specified. Processing: Fetch the MMU control register to a variable. If the argument passed is true, set the MMU enable bit, otherwise, clear it. Write the resultant value back to the control register.
cp15_set_mmu_control_reg (see page 14)	Set MMU Coprocessor(CP15) Control register Function: cp15_set_mmu_control_reg Purpose: To set MMU Coprocessor (CP15) Control register. Processing: Set the MMU control register to a value passed as parameter.
cp15_set_transtable_base (see page 15)	Sets the first-level translation table base address Function: cp15_set_transtable_base Purpose: Sets the first-level translation table base address Processing: Masks out the lower 12 bits of the address passed. Writes register 2 of CP15 with the base address passed as parameter.
cp15_set_vmmu_addr (see page 15)	Set the virtual address of the MMU table Function: cp15_set_vmmu_addr Purpose: Set the virtual address of the MMU table Processing: Set the saved virtual MMU table address to the passed value.
cp15_write_buffer_flush (see page 16)	Force an write buffer flush Function: cp15_write_buffer_flush Purpose: Force an write buffer flush Processing: Flush the write buffer and wait for completion of the flush.

Macros

Macro	Description
LPC_ARM922T_CP15_DRIVER_H (see page 138)	This is macro LPC_ARM922T_CP15_DRIVER_H.

Types

Type	Description
CPAGETABLE_T (see page 61)	ARM 922T MMU Coarse page table type
FPAGETABLE_T (see page 65)	ARM 922T MMU Fine page table type
TRANSTABLE_T (see page 72)	ARM 922T MMU Translation table structure
TT_SECTION_BLOCK_T (see page 72)	UNS_32 (see page 73) num_sections: number of 1MByte sections >=1 for all blocks except last; last = 0 UNS_32 (see page 73) virt_addr: as required, base Virtual address for block UNS_32 (see page 73) phys_addr: as required, PT address or Section address UNS_32 (see page 73) entry is composed of the following 'or'd' together: access_perm: ARM922T_L1D_AP_x (x = SVC_ONLY, USR_RO, ALL) domain: ARM922T_L1D_DOMAIN (see page 104)(n) as applicable cacheable: ARM922T_L1D_CACHEABLE (see page 103) if applicable write_buffered: ARM922T_L1D_BUFFERABLE (see page 103) if applicable descriptor_type: ARM922T_L1D_TYPE_x (x = FAULT, PAGE, SECTION)

1.6.6 lpc_bmp.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: BMP file structures

*

- Description:
- See the bmp.h header file for a description of this package.

*

- This package uses *malloc*. If you want to use this package, you
- should replace malloc with your own dynamic allocation call if
- malloc is an invalid function.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

1.6.7 lpc_bmp.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: BMP file structures

*

- Description:
- This package contains the structure of the BMP file format.

*

- Notes:
- Data in the BMP header (as read from a file) is not stored word
- aligned after the identifier. If the structure is read from a
- file, the header information may need to be realigned to the
- structure alignment.

*

- It is the intention of this package to support the most common
- BMP image formats in use. Not all BMP formats are supported.

*

- Unsupported BMP formats:
- RLE compression is not supported
- 16-bit and 32-bit color images are not supported
- Masks stored in the color table are not supported

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
bmp_allocate_structure (see page 4)	Allocates storage for a new BMP file Function: bmp_convert_image (see page 5) Purpose: Allocates storage for a new BMP file structure. Processing: This function computes the required size needed for the BMP header, color table, and image data, based on the color depth. Memory for an image (with header and color table) is allocated and the pointer returned to the caller.
bmp_convert_color (see page 4)	Converts a BMP color table entry to a color_type color Function: bmp_convert_color Purpose: Converts a BMP color table entry to a COLOR_T (see page 61) color Processing: A color table entry (or raw 24-bit entry) is converted into the native (compiled) color type by masking and shifting the red, green, and blue components of color and computing the closest color in the native format (either 233, 555, or 565).
bmp_convert_image (see page 5)	Convert a BMP image to a color_type image Function: bmp_convert_image Purpose: Convert a BMP image to a COLOR_T (see page 61) image Processing: See function.
bmp_get_color_table (see page 6)	Returns a pointer to the color table Function: bmp_get_color_table Purpose: Returns a pointer to the color table Processing: A call to bmp_is_header_valid (see page 7) is performed to determine the BMP file type. If the BMP file type is BPP1, BPP4, or BPP8, then the color table is assigned a pointer after the BMP header information.

bmp_get_image_data (see page 6)	Returns a pointer to the BMP image data Function: bmp_get_image_data Purpose: Returns a pointer to the BMP image data. Processing: A call to bmp_is_header_valid (see page 7) is performed to determine the BMP file type. Based on the BMP file type, the number of entries in the color table is computed. The pointer to the image data is computed at the end of the header plus an offset for the color table.
bmp_is_header_valid (see page 7)	Determine if the structure is a BMP structure Function: bmp_is_header_valid Purpose: Determine if the structure is a BMP structure Processing: The header type (bftype) is examined to match 'BM'. If it matches and the file type is uncompressed, then the color depth is examined and the return value set to the appropriate color depth enumeration. If an unsupported type is found, type INVALID_BMP will be returned.

Macros

Macro	Description
BI_BITFIELDS (see page 122)	Uncomp RGB with sample packing
BI_RGB (see page 123)	Uncompressed image identifier
BI_RGBA (see page 123)	Uncompressed image identifier alias for BI_RGB (see page 123)
BI_RLE4 (see page 123)	4-bit RLE compression
BI_RLE8 (see page 123)	8-bit RLE compression
BI_RLE8A (see page 123)	8-bit RLE compression for BI_RLE8 (see page 123)
BMP_ID0 (see page 125)	BMP file identifier character 1
BMP_ID1 (see page 125)	BMP file identifier character 2
LPC_BMP_H (see page 138)	This is macro LPC_BMP_H.
RGBA (see page 146)	Raw RGB with alpha
RGBT (see page 147)	Raw RGB with a transparency field

Types

Type	Description
BMP_COLOR_TABLE_T (see page 58)	Color table entry format (used with BPP1, BPP4, and BPP8)
BMP_STORAGE_T (see page 58)	Supported BMP file formats (no compressed or masked color modes are supported)
BMP_T (see page 59)	BMP header structure, not used with files
BMP24_COLOR_TABLE_T (see page 59)	Color table entry format used with BPP24

1.6.8 lpc_colors.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Color definitions

*

- Description:
- See the SMA_colors.h header file for a description of this
- package.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified

- use without further testing or modification.

1.6.9 lpc_colors.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Color definitions

*

- Description:
- This package contains functions for color mapping, color
- conversion, and common defines.

*

- The palette table function can be configured for 555 or 565
- color.

*

- Notes:
- Color entries are stored in BGR format, with blue mapped to the
- most significant bits of a color type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
lpc_colors_set_palette (see page 34)	Generate a palette table (only in 8-bit mode). If compiled in 16-bit color mode, this will be a NULL (see page 143) function. Function: lpc_colors_set_palette Purpose: Generate a palette table (only in 8-bit mode). Processing: Depending on the target LCD color mapping (either 555 or 565), a palette table will be generated to convert colors stored in 233 format to either 555 or 565 format through a lookup table.

Macros

Macro	Description
BLACK (see page 124)	Black color, 323 mode
BLUE (see page 124)	Blue color, 323 mode
BLUE_COLORS (see page 124)	Number of blue colors in 332 mode
BLUEMASK (see page 124)	Blue color mask, 323 mode
BLUESHIFT (see page 125)	Blue shift value, 323 mode
COLORS_DEF (see page 128)	16-bit 565 color mode #define COLORS_DEF 15 /* 15-bit 555 color mode */ #define COLORS_DEF 12 /* 12-bit 444 color mode */
CYAN (see page 128)	Cyan color, 323 mode

DARKGRAY (see page 128)	Dark gray color, 323 mode
GREEN (see page 133)	Green color, 323 mode
GREEN_COLORS (see page 133)	Number of green colors in 332 mode
GREENMASK (see page 133)	Green color mask, 323 mode
GREENSHIFT (see page 133)	Green shift value, 323 mode
LIGHTBLUE (see page 136)	Light blue color, 323 mode
LIGHTCYAN (see page 136)	Light cyan color, 323 mode
LIGHTGRAY (see page 136)	Light gray color, 323 mode
LIGHTGREEN (see page 137)	Light green color, 323 mode
LIGHTMAGENTA (see page 137)	Light magenta color, 323 mode
LIGHTRED (see page 137)	Light red color, 323 mode
LIGHTYELLOW (see page 137)	Light yellow color, 323 mode
LPC_COLOR_TYPES_H (see page 138)	This is macro LPC_COLOR_TYPES_H.
MAGENTA (see page 142)	Magenta color, 323 mode
NUM_COLORS (see page 144)	Number of colors in 332 mode
RED (see page 145)	Red color, 323 mode
RED_COLORS (see page 145)	Number of red colors in 332 mode
REDMASK (see page 146)	Red color mask, 323 mode
REDSHIFT (see page 146)	Red shift value, 323 mode
WHITE (see page 153)	White color, 323 mode
YELLOW (see page 153)	Yellow color, 323 mode

Types

Type	Description
COLOR_T (see page 61)	Color type is a 8-bit value

1.6.10 lpc_fat16.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

Project: FAT16 driver

This package uses heap functions in lpc_heap.c (see page 172) All filenames must be in uppercase letters and 8.3 format

Macros

Macro	Description
BT_SIG_OFS (see page 125)	This is macro BT_SIG_OFS.
BT_SIG_SZ (see page 125)	This is macro BT_SIG_SZ.
BYTES_SEC_OFS (see page 126)	This is macro BYTES_SEC_OFS.
BYTES_SEC_SZ (see page 126)	This is macro BYTES_SEC_SZ.
DV_NUM_OFS (see page 130)	This is macro DV_NUM_OFS.
DV_NUM_SZ (see page 130)	This is macro DV_NUM_SZ.
FAT_COPY_OFS (see page 131)	This is macro FAT_COPY_OFS.
FAT_COPY_SZ (see page 131)	This is macro FAT_COPY_SZ.
FSNAME_OFS (see page 132)	This is macro FSNAME_OFS.
FSNAME_SZ (see page 133)	This is macro FSNAME_SZ.
HDN_SECS_OFS (see page 134)	This is macro HDN_SECS_OFS.
HDN_SECS_SZ (see page 134)	This is macro HDN_SECS_SZ.
JUMP_OFS (see page 135)	Local defines
JUMP_SZ (see page 135)	This is macro JUMP_SZ.

LABEL_OFS (see page 135)	This is macro LABEL_OFS.
LABEL_SZ (see page 135)	This is macro LABEL_SZ.
LG_SECS_OFS (see page 136)	This is macro LG_SECS_OFS.
LG_SECS_SZ (see page 136)	This is macro LG_SECS_SZ.
MEDIA_DES_OFS (see page 143)	This is macro MEDIA_DES_OFS.
MEDIA_DES_SZ (see page 143)	This is macro MEDIA_DES_SZ.
NUM_HDS_OFS (see page 144)	This is macro NUM_HDS_OFS.
NUM_HDS_SZ (see page 144)	This is macro NUM_HDS_SZ.
OEMID_OFS (see page 144)	This is macro OEMID_OFS.
OEMID_SZ (see page 144)	This is macro OEMID_SZ.
RES_SECT_OFS (see page 146)	This is macro RES_SECT_OFS.
RES_SECT_SZ (see page 146)	This is macro RES_SECT_SZ.
ROOT_ENT_OFS (see page 147)	This is macro ROOT_ENT_OFS.
ROOT_ENT_SZ (see page 147)	This is macro ROOT_ENT_SZ.
RSV_OFS (see page 147)	This is macro RSV_OFS.
RSV_SZ (see page 148)	This is macro RSV_SZ.
SECS_CLUS_OFS (see page 148)	This is macro SECS_CLUS_OFS.
SECS_CLUS_SZ (see page 148)	This is macro SECS_CLUS_SZ.
SECS_FAT_OFS (see page 148)	This is macro SECS_FAT_OFS.
SECS_FAT_SZ (see page 148)	This is macro SECS_FAT_SZ.
SECS_TK_OFS (see page 149)	This is macro SECS_TK_OFS.
SECS_TK_SZ (see page 149)	This is macro SECS_TK_SZ.
SERNUM_OFS (see page 149)	This is macro SERNUM_OFS.
SERNUM_SZ (see page 149)	This is macro SERNUM_SZ.
SMALL_SEC_OFS (see page 152)	This is macro SMALL_SEC_OFS.
SMALL_SEC_SZ (see page 152)	This is macro SMALL_SEC_SZ.

1.6.11 lpc_fat16.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: FAT16 driver

*

- Description:
- This package contains a set of functions to provide simple
- management functions for FAT16 devices, such as a Compact Flash
- card or MMC card. The actual device type does not matter, and a
- set of standard routines are needed to bind the device driver
- to this FAT16 driver.

*

- This driver supports functions only related to FAT16 functionality
- and has very simple error checking. Some file related functions
- that are normally not included in the FAT16 layer are included in
- this driver to keep functionality simple. MBR functions are also
- included as part of the FAT16 driver for convienence only.

*

- The following functions are supported in this driver:
- FAT16 to device binding (initialization and shutdown)
- Get device partition data (active status, type)

- Mount partition/filesystem
- Set an active directory
- Reset a directory pointer to the head of directory table
- Get a directory entry (may be a file or other directory)
- File operations (operations occur in active directory)
- Open a file, read data from a file, close file
- Create a file, write data to a file, close file
- Delete a file.

*

- Use of this driver is explained in the fat16.txt document. There
- are limitations with this driver - read the fat16.txt file for
- important information on opening multiple files.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
fat16_cd (see page 16)	
fat16_close_file (see page 17)	Close a file that was open for reading or writing, or anything else (will destroy the file descriptor) Function: fat16_close_file Purpose: Close a file that was open for reading or writing. Processing: See function.
fat16_create_new_file_descriptor (see page 18)	File descriptor creation/destroy functions
fat16_delete (see page 18)	
fat16_destroy_file_descriptor (see page 19)	Destroys a created file descriptor Function: fat16_destroy_file_descriptor Purpose: Destroys a created file descriptor. Processing: Prior to destroying the file descriptor, a call to fat16_close is performed to write any data in the write buffer out to the device. If the directory has been changed in any way, the cached directory is written back to the device. The structures used in the file descriptor and the file descriptor itself are then de-allocated.
fat16_get_active_mbr (see page 20)	
fat16_get_dirname (see page 21)	Returns the name and type of the (next) entry in the active directory Function: fat16_get_dirname Purpose: Returns the name and type of the entry in the active directory (in unpadded 8.3 format). Processing: See function.
fat16_get_status (see page 22)	
fat16_init_device (see page 23)	Pointer for write of data Function: fat16_init_device Purpose: Initializes the FAT16 interface for the selected device. Processing: Copy the device name and function pointers into the FAT device structure. Clear the commit flag to indicate the FAT cluster table does not need to be written back to the device. Call the device initialization function. If the device was initialized, read the MBR into the FAT device structure.

fat16_open_file (see page 25)	Open a file for reading or writing Function: fat16_open_file Purpose: Open a file for reading or writing. Processing: See function.
fat16_read (see page 26)	Read data from a file Function: fat16_read Purpose: Read data from a file. Processing: See function.
fat16_save_all (see page 28)	Function: fat16_save_all Purpose: Shutdown the FAT16 interface for the selected device. Processing: If the commit flag is set, write the cached FAT cluster table back to the device. Free the allocated memory for the cluster table and device structure.
fat16_seek (see page 28)	Function: fat16_seek Purpose: Seek data pointer. Processing: See function.
fat16_set_dir_index (see page 29)	Resets the directory index to a location of the directory (used with get_dirname) Function: fat16_set_dir_index Purpose: Resets the directory index to a location of the directory (used with get_dirname) Processing: See function.
fat16_set_partition (see page 30)	Set the active (FAT16) partition and cache cluster table Function: fat16_set_partition Purpose: Set the active partition. Processing: If the partition is a valid type (FAT16), the starting sector value for the partition will be determined and the appropriate sector containing the boot record will be read from the device. Once the boot record has been read in, the partition dimensions are computed. Appropriate space for the FAT cluster table is allocated and the cluster table is cached in memory.
fat16_shutdown (see page 30)	Shutdowns the FAT16 interface for the selected device (will destroy the FAT device structure) Function: fat16_shutdown Purpose: Shutdown the FAT16 interface for the selected device. Processing: If the commit flag is set, write the cached FAT cluster table back to the device. Free the allocated memory for the cluster table and device structure.
fat16_write (see page 32)	Write data to a file Function: fat16_write Purpose: Write data to a file. Processing: See function.

Macros

Macro	Description
ATTB_ARCHIVE (see page 121)	Archive bit
ATTB_DIR (see page 121)	Directory bit
ATTB_HIDDEN (see page 121)	Hidden file bit
ATTB_LFN (see page 121)	LFN entry flag
ATTB_NORMAL (see page 121)	Normal file type (no bits set)
ATTB_RO (see page 122)	Read only bit
ATTB_SYS (see page 122)	System file bit
ATTB_VOLUME (see page 122)	Volume bit
CLUSTER_AV (see page 126)	Cluster available
CLUSTER_BAD (see page 126)	Bad cluster flag
CLUSTER_LAST (see page 127)	Minimum (16-bit) value for last cluster
CLUSTER_MAX (see page 127)	Maximum amount of cluster entries
CLUSTERR_MAX (see page 127)	Maximum reserved cluster flag
CLUSTERR_MIN (see page 127)	Minimum reserved cluster flag
CLUSTERU_MAX (see page 127)	Maximum cluster chain range
CLUSTERU_MIN (see page 128)	Minimum cluster chain range
DEFAULT_CR_DATE (see page 129)	January 1, 2002
DEFAULT_CR_TIME (see page 129)	12:00:00
DIR_ERASED (see page 129)	Erased (free) directory entry
DIR_FREE (see page 129)	Free directory entry
DSIZE (see page 129)	Device name string size
EXTENDED_SIG (see page 130)	FAT16 extended signature
EXTENDED_SIG_IDX (see page 130)	Extended signature index in data
FAT12 (see page 131)	Partition type FAT12
FAT16_EXDOS (see page 132)	Partition type extended MSDOS
FAT16_GT32M (see page 132)	Partition type FAT16 size more than 32M

FAT16_LT32M (see page 132)	Partition type FAT16 size less than 32M
LPC_FAT16_H (see page 139)	This is macro LPC_FAT16_H.
PART_ACTV (see page 145)	Partition active flag bit

Types

Type	Description
DEVICE_FUNCS_TYPE (see page 61)	This is type DEVICE_FUNCS_TYPE.
FAT_DEVICE_TYPE (see page 61)	FAT device structure, used to bind a device driver to the FAT driver
FATDATA_TYPE (see page 62)	The following structure holds computed information about the device
FATGEOM_TYPE (see page 63)	Drive geometry structure for partition, filled in by the driver. (Not everything in this sector is saved)
FILE_MODE_TYPE (see page 64)	File modes
FILE_TYPE (see page 64)	File descriptor
ivfunc (see page 66)	This is type ivfunc.
ivifunc (see page 66)	This is type ivifunc.
PARTITION_TYPE (see page 69)	Partition entries
ROOT_ENTRY_TYPE (see page 70)	Initialization functions
vvfunc (see page 74)	Device function list

1.6.12 lpc_fat16_private.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

Project: FAT16 support functions

Software that is described herein is for illustrative purposes only which provides customers with programming information regarding the products. This software is supplied "AS IS" without any warranties. NXP Semiconductors assumes no responsibility or liability for the use of the software, conveys no license or title under any patent, copyright, or mask work right to the product. NXP Semiconductors reserves the right to make changes in the software without notification. NXP Semiconductors also make no representation or warranty that such application will be suitable for the specified use without further testing or modification.

1.6.13 lpc_fat16_private.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: FAT16 support functions

*

- Description:
- This package contains support functions for the FAT16 driver.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or

- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
fat16_compare (see page 17)	Compares two strings for similarity Function: fat16_compare Purpose: Simple data comparison routine. Processing: Two strings are compared in lowercase up to the number of characters set by 'size'.
fat16_find_file (see page 19)	Finds and returns the directory structure of the passed name in the active directory Function: fat16_find_file Purpose: Finds and returns the directory structure of the passed name in the active directory. Processing: See function.
fat16_find_free_cluster (see page 20)	Find the next free cluster in the cluster list. Searches down from the passed cluster Function: fat16_find_free_cluster Purpose: Find the next free cluster in the cluster list. Searches down from the passed cluster. Processing: See function.
fat16_get_free_dir_entry (see page 21)	Allocates a new directory entry for the passed name Function: fat16_get_free_dir_entry Purpose: Allocates a new directory entry for the passed name. Processing: See function.
fat16_get_next_cluster (see page 22)	Returns the next cluster in a cluster link chain Function: fat16_get_next_cluster Purpose: Returns the next cluster in a cluster link chain. Processing: See function.
fat16_moveto (see page 24)	
fat16_name_break (see page 24)	Converts a filename in unpadded 8.3 format to a format that is compatible with a directory format Function: fat16_name_break Purpose: Converts a filename in unpadded 8.3 format to a format that is compatible with a directory format. Processing: See function.
fat16_name_check (see page 25)	Compares a passed name in padded 8.3 format with a name in a directory entry structure Function: fat16_name_check Purpose: Compares a passed name in padded 8.3 format with a name in a directory entry structure. Processing: Compare the first 11 characters of the passed name with the 11 characters in the passed directory structure.
fat16_parse_path (see page 26)	Finds the next directory name in a path Function: fat16_parse_path Purpose: Finds the next directory name in a path. Processing: See function.
fat16_read_mbr (see page 27)	Reads the FAT MBR and puts the partition tables in the passed structure Function: fat16_read_mbr Purpose: Reads the FAT MBR and puts the partition tables in the passed structure. Processing: Read CHS (0, 0, 1) from the device (this is always the MBR in a storage device). Copy the partition data from the device data into the partition data table. Set the selected active partition to (-1), indicating that a partition has not been selected.
fat16_read_sectors (see page 27)	Reads a number of sectors from a device into a buffer Function: fat16_read_sectors Purpose: Reads a number of sectors from a device into a buffer. Processing: See function.
fat16_set_no_mbr (see page 29)	Support function to set up the first partition in the driver to point to sector 1 for the boot record Function: fat16_set_no_mbr Purpose: Sets up the first partition in the cached parition table to point to sector 1 as a FAT16 boot record. Processing: See function.
fat16_translate_cluster_to_sector (see page 31)	Translate a cluster number to a (absolute) sector number Function: fat16_translate_cluster_to_sector Purpose: Translate a cluster number to a (absolute) sector number. Processing: See function.
fat16_wait_busy (see page 31)	Wait for the device to go 'unbusy Function: fat16_wait_busy Purpose: Wait for the device to go 'unbusy'. Processing: Check the status of the device busy function. If the device is busy, perform a small loop and check again until the device is no longer busy.
fat16_write_sectors (see page 32)	Writes a number of sectors from a buffer to a device Function: fat16_write_sectors Purpose: Writes a number of sectors from a buffer to a device. Processing: See function.

Macros

Macro	Description
LPC_FAT16_PRIVATE_H (see page 139)	This is macro LPC_FAT16_PRIVATE_H.
PTAB_SIZE (see page 145)	Size of MBR and boot records

1.6.14 lpc_fonts.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Fonts selection

*

- Description:
- This package provides a common font information structure.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

1.6.15 lpc_fonts.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Fonts selection

*

- Description:
- This package provides a common font information structure.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or

- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_FONTS_H (see page 139)	This is macro LPC_FONTS_H.

Types

Type	Description
FONT_T (see page 64)	Font data structure

1.6.16 lpc_heap.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Simple heap manager

*

- Description:
- See the header file for a description of this package.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
HEAP_HEAD_SIZE (see page 134)	Heap descriptor size
HEAP_POINTER_NULL (see page 134)	Pointer to NULL (see page 143) heap descriptor
SMALLEST_ENTRY_SIZE (see page 152)	Smallest heap descriptor entry

Types

Type	Description
HEAP_DESCRIPTOR_T (see page 65)	Heap descriptor

Variables

Variable	Description
heap_base (see page 76)	Heap base address
heap_size_saved (see page 76)	Heap size

1.6.17 lpc_heap.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Simple heap manager

*

- Description:
- This package provides a simple heap manager with the first-fit
- algorithm. Before the package can be used, a call to
- lpc_heap_init (see page 37) must be performed with the base heap address and
- the size of the heap in bytes.

*

- All returned allocation areas are 32-bit aligned.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
lpc_free (see page 35)	Return an allocated area to the heap Function: lpc_free Purpose: Returns an allocated entry of memory to the heap. Processing: See function.
lpc_get_allocated_count (see page 35)	Return the number of allocated items in the heap Function: lpc_get_allocated_count Purpose: Return the number of allocated items in the heap. Processing: This function traverses through the heap list. If an entry has an available size of 0 bytes, then the entry is assumed as allocated and the allocated count is incremented.
lpc_get_heap_base (see page 36)	Return the heap base address Function: lpc_get_heap_base Purpose: Return the heap base address. Processing: See function.
lpc_get_heapsize (see page 36)	Return the size of the heap area Function: lpc_get_heapsize Purpose: Returns the size of the heap. Processing: See function.
lpc_get_largest_chunk (see page 36)	Return the size of the largest unallocated heap chunk Function: lpc_get_largest_chunk Purpose: Returns the largest available chunk in the heap. Processing: This function traverses through the heap list. If an entry has an available size of greater than 0 bytes, then the entry is assumed as free and the size of the chunk is compared to the running size count. If the size is larger, the running size count is updated with the new size.

lpc_heap_init (see page 37)	Setup the heap area Function: lpc_heap_init Purpose: Setup the heap area. Processing: The heap base address and size counters are set with the passed parameter values. The first entry of the heap is set up with an unallocated heap list entry.
lpc_new (see page 38)	Get an allocated area from the heap Function: lpc_new Purpose: Get an allocated area from the heap. Processing: See function.

Macros

Macro	Description
LPC_HEAP_H (see page 139)	This is macro LPC_HEAP_H.

1.6.18 lpc_helvr10.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convbdf on Tue Oct 3 00:24:24 MDT 2000. Font information:

name: -Adobe-Helvetica-Medium-R-Normal--10-100-75-75-P-56-ISO8859-1 pixel size: 10 ascent: 10 descent: 2

Variables

Variable	Description
font_helvr10 (see page 75)	Externally available font information structure
helvr10_bits (see page 76)	Font character bitmap data.
helvR10_width (see page 78)	Character width data.

1.6.19 lpc_helvr10.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Helvetica 10-point proportional font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_HEVR10_FONT_H (see page 140)	This is macro LPC_HEVR10_FONT_H.

1.6.20 lpc_lcd_params.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Sharp LCD parameters

*

- Description:
- This file contains common LCD parameters used on all Sharp
- evaluation boards.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Variables

Variable	Description
sharp_lm057qb (see page 87)	Sharp LM057QB STN display
sharp_lm057qc (see page 88)	Sharp LM057QC STN display
sharp_lm10v (see page 88)	Sharp LM10V DSTN display
sharp_lm64k11 (see page 88)	Sharp LM64K11 STN display
sharp_lq035 (see page 88)	Sharp LQ035 portrait mode ADTFT display
sharp_lq039 (see page 89)	Sharp LQ039 HRTFT display
sharp_lq050 (see page 89)	Sharp LQ050 TFT display - also works for the LQ036 and LQ038 LCDs
sharp_lq057 (see page 89)	Sharp LQ057 TFT display
sharp_lq064 (see page 89)	Sharp LQ064 TFT display
sharp_lq104 (see page 89)	Sharp LQ104 TFT display
sharp_lq121 (see page 90)	Sharp LQ121 TFT display

1.6.21 lpc_lcd_params.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Sharp LCD parameters

*

- Description:
- This file contains common LCD parameters used on all Sharp
- evaluation boards.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_SHARP_LCD_PARAM_H (see page 140)	This is macro LPC_SHARP_LCD_PARAM_H.

Types

Type	Description
LCD_PANEL_T (see page 67)	LCD display types
LCD_PARAM_T (see page 67)	Structure containing the parameters for the LCD panel

1.6.22 lpc_rom8x16.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convrom.exe ROM 8x16 Font bios mode 12

Variables

Variable	Description
font_rom8x16 (see page 75)	Externally available font information structure
rom8x16_bits (see page 78)	This is variable rom8x16_bits.
rom8x16_width (see page 84)	Character width data.

1.6.23 lpc_rom8x16.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: 8x16 proportional font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors

- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_ROM8X16_FONT_H (see page 140)	This is macro LPC_ROM8X16_FONT_H.

1.6.24 lpc_rom8x8.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convrom.exe ROM 8x8 Font bios mode 10

Variables

Variable	Description
font_rom8x8 (see page 75)	Externally available font information structure
rom8x8_bits (see page 84)	This is variable rom8x8_bits.
rom8x8_width (see page 87)	Character width data.

1.6.25 lpc_rom8x8.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: 8x8 proportional font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_ROM8X8_FONT_H (see page 140)	This is macro LPC_ROM8X8_FONT_H.

1.6.26 lpc_swim.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $
- *
- Project: Simple Windowing Interface Manager (SWIM)
- *
- Description:
- See the swim.h header file for a description of this package.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

1.6.27 lpc_swim.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $
- *
- Project: Simple Windowing Interface Manager (SWIM)
- *
- Description:
- This package provides a simple windows manager that provides the
- following functions:
- Windows initialization and validity checks
- Must be in physical display space
- Color support for background, primary pen, and fill
- Simple graphics primatives (pixels, lines, boxes)
- Window deallocation

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors

- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
swim_clear_screen (see page 40)	Fills the draw area of the display with the selected color Function: swim_clear_screen Purpose: Fills the draw area of the display with the selected color Processing: Loop through all virtual window (draw area) locations and updates them with the passed color value.
swim_get_horizontal_size (see page 41)	Get the virtual window horizontal size Function: swim_get_horizontal_size Purpose: Get the virtual window horizontal size Processing: For the passed window ID, return the x size of the window.
swim_get_vertical_size (see page 41)	Get the virtual window vertical size Function: swim_get_vertical_size Purpose: Get the virtual window vertical size Processing: For the passed window ID, return the x size of the window.
swim_put_box (see page 42)	Place a box with corners (X1, Y1) and (X2, Y2). Use pen color for edges and fill color for center Function: swim_put_box Purpose: Place a box with corners (X1, Y1) and (X2, Y2) Processing: See function.
swim_put_diamond (see page 43)	Draw a diamond in the virtual window Function: swim_put_diamond Purpose: Purpose: Draw a diamond in the virtual window Processing: See function.
swim_put_line (see page 45)	Draw a line in the virtual window Function: swim_put_line Purpose: Draw a line in the virtual window with clipping. Processing: See function.
swim_put_pixel (see page 47)	Puts a pixel at (X, Y) in the pen color Function: swim_put_pixel Purpose: Puts a pixel at the virtual X, Y coordinate in the window Processing: Convert the virtual pixel position to a physical position. If the pixel is inside the window draw area, update the pixel on the display.
swim_set_bkg_color (see page 52)	Set background color Function: swim_set_bkg_color Purpose: Sets the color used for backgrounds Processing: For the passed window ID, update to the passed background color.
swim_set_fill_color (see page 52)	Set fill color (used for boxes and circles) Function: swim_set_fill_color Purpose: Sets the fill color Processing: For the passed window ID, update to the passed fill color.
swim_set_pen_color (see page 54)	Set the pen color Function: swim_set_pen_color Purpose: Sets the pen color Processing: For the passed window ID, update to the passed pen color.
swim_window_close (see page 55)	Destroy a window Function: swim_window_close Purpose: Reallocates a window for use Processing: For the passed window ID, clear the window used flag.
swim_window_open (see page 56)	Initialize a window Function: swim_window_open Purpose: Initializes a window and the default values for the window Processing: See function.
swim_window_open_noclear (see page 56)	Initialize a window without clearing it Function: swim_window_open_noclear Purpose: Initializes a window and the default values for the window Processing: See function.

Macros

Macro	Description
LPC_SWIM_H (see page 141)	This is macro LPC_SWIM_H.

Types

Type	Description
SWIM_WINDOW_T (see page 71)	Structure is used to store information about a specific window

1.6.28 lpc_swim_font.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Font management for SWIM

*

- Description:
- See the sma_swim_font.h header file for a description of this
- package.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

1.6.29 lpc_swim_font.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Font management for SWIM

*

- Description:
- This package provides the following font capabilities with SWIM:
- Font selection
- Text positioning
- newline and window scrolling
- Text display with multiple, selectable fonts

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.

- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
swim_get_font_height (see page 40)	Returns the active font's height in pixels Function: swim_get_font_height Purpose: Returns the active font's height in pixels Processing: See function.
swim_get_xy (see page 42)	Returns the X, Y pixel coordinates for the next text operation Function: swim_get_xy Purpose: Returns the X, Y pixel coordinates for the next text operation Processing: The logical X and Y positions are computed by subtracting the physical text position values by the physical minimum window limits.
swim_put_char (see page 43)	Puts a single character to the window Function: swim_put_char Purpose: Puts a character in the window. Processing: See function.
swim_put_ltext (see page 46)	Puts a null-terminated string of text in a window, but will move an entire word to the next line if it will not fit on the present line Function: swim_put_ltext Purpose: Puts a string of text in a window, but will adjust the position of a word if the word length exceeds the edge of the display. Processing: While the string has data in it, check for the newline character. If it exists, output a newline. If the string data is inside the font character table, output the first word in the string (with support for generating a newline if the word will exceed the window edge). Continue until all words/characters are output.
swim_put_newline (see page 46)	Puts a newline in the window Function: swim_put_newline Purpose: Performs a newline in a window Processing: Set the text pointer for the next text character operation to the beginning of the following line. If the following lin exceeds the window size, perform a line scroll.
swim_put_text (see page 50)	Puts a null-terminated string of text in a window Function: swim_put_text Purpose: Puts a string of text in a window Processing: Each character will be routed to the swim_put_char (see page 43) function until a string terminator is reached. For newline characters, a newline will occur instead of a character output.
swim_put_text_xy (see page 51)	Put a text message at an X, Y pixel coordinate in the window Function: swim_put_text_xy Purpose: Put text at x, y (char) position on screen Processing: Set the virtual (upper left) text position in the window and render the text string at this position.
swim_set_font (see page 53)	Select the active font Function: swim_set_font Purpose: Sets the active font Processing: Switch to the selected font by setting the font structure pointer in the windows structure based on the passed enumeration. If the next character output in the new font will exceed the window limit, perform a window text scroll.
swim_set_font_trasparency (see page 53)	Enables and disables font backgrounds Function: swim_set_font_trasparency Purpose: Enables and disables font backgrounds. When set, the font background will not be drawn in the background color (useful for painting text over pictures). Processing: See function.
swim_set_title (see page 54)	Create a title bar Function: swim_set_title Purpose: Creates a title bar in the window and adjusts the client area to be outside the title bar area. Processing: See function.
swim_set_xy (see page 55)	Sets the X, Y pixel coordinates for the next text operation Function: swim_set_xy Purpose: Sets the X, Y pixel coordinates for the next text operation Processing: Update the X, Y text position pointers, limiting the position to the window dimensions.

Macros

Macro	Description
LPC_SWIM_FONT_H (see page 140)	This is macro LPC_SWIM_FONT_H.

1.6.30 lpc_swim_image.c

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $
- *
- Project: Image management for SWIM
- *
- Description:
- See the swim.h header file for a description of this package.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

1.6.31 lpc_swim_image.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $
- *
- Project: Image management for SWIM
- *
- Description:
- This package provides the following image capabilities with SWIM:
- Display of raw image data (stored left to right, top to
- bottom)
- Stored raw images MUST be stored in the same color format as
- color_type
- Image scaling, rotation, and clipping

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without

- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Functions

Function	Description
swim_put_image (see page 44)	Puts a raw image into a window Function: swim_put_image Purpose: Puts an raw image in a window unscaled, clips off edges Processing: See function.
swim_put_invert_image (see page 44)	Puts a raw image into a window inverted Function: swim_put_invert_image Purpose: Puts an raw image in a window unscaled, inverted, with clipped edges. Processing: See function.
swim_put_left_image (see page 45)	Puts a raw image into a window rotated left Function: swim_put_left_image Purpose: Puts an raw image in a window unscaled, rotated left, with clipped edges. Processing: See function.
swim_put_right_image (see page 47)	Puts a raw image into a window rotated right Function: swim_put_right_image Purpose: Puts an raw image in a window unscaled, rotated right, with clipped edges. Processing: See function.
swim_put_scale_image (see page 48)	Puts and scales a raw image into a window Function: swim_put_scale_image Purpose: Puts an raw image in a window scaled. Processing: See function.
swim_put_scale_invert_image (see page 49)	Puts and scales a raw image into a window inverted Function: swim_put_scale_invert_image Purpose: Puts an raw image in a window scaled and inverted. Processing: See function.
swim_put_scale_left_image (see page 49)	Puts and scales a raw image into a window rotated left Function: swim_put_scale_left_image Purpose: Puts an raw image in a window scaled and rotated left. Processing: See function.
swim_put_scale_right_image (see page 50)	Puts and scales a raw image into a window rotated right Function: swim_put_scale_right_image Purpose: Puts an raw image in a window scaled and rotated right. Processing: See function.
swim_put_win_image (see page 51)	One API for all the functions Function: swim_put_win_image Purpose: This function simply provides a single API for all the image functions. Processing: See function.

Macros

Macro	Description
LPC_SWIM_IMAGE_H (see page 141)	This is macro LPC_SWIM_IMAGE_H.

Types

Type	Description
SWIM_ROTATION_T (see page 71)	Image rotation tags

1.6.32 lpc_types.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Common Include Files

*

- Description:
- lpc_types.h contains the NXP ABL typedefs for C standard types.
- It is intended to be used in ISO C conforming development
- environments and checks for this insofar as it is possible
- to do so.

*

- lpc_types.h ensures that the name used to define types correctly
- identifies a representation size, and by direct inference the
- storage size, in bits. E.g., UNS_32 (see page 73) identifies an unsigned
- integer type stored in 32 bits.

*

- It requires that the basic storage unit (char) be stored in
- 8 bits.

*

- No assumptions about Endianess are made or implied.

*

- lpc_types.h also contains NXP ABL Global Macros:
- _BIT (see page 100)
- _SBF (see page 100)
- _BITMAP
- These #defines are not strictly types, but rather Preprocessor
- Macros that have been found to be generally useful.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
_BIT (see page 100)	Set bit macro
_BITMASK (see page 100)	Bitmask creation macro
_ERROR (see page 100)	ERROR macro
_NO_ERROR (see page 100)	NO_ERROR macro
_SBF (see page 100)	Set bit field macro
EXTERN (see page 131)	This is macro EXTERN.
FALSE (see page 131)	FALSE macro
LPC_TYPES_H (see page 141)	This is macro LPC_TYPES_H.
NELEMENTS (see page 143)	Number of elements in an array
NULL (see page 143)	NULL pointer
SMA_BAD_CLK (see page 150)	Bad device clock macro
SMA_BAD_HANDLE (see page 150)	Bad device handle macro
SMA_BAD_PARAMS (see page 150)	Device bad paramaters macro

SMA_CANT_START (see page 150)	Device can't start macro
SMA_CANT_STOP (see page 150)	Device can't stop macro
SMA_DEV_UNKNOWN (see page 151)	Device unknown macro
SMA_IN_USE (see page 151)	Device in use macro
SMA_NOT_OPEN (see page 151)	Device not open macro
SMA_NOT_SUPPORTED (see page 151)	Device not supported macro
SMA_PIN_CONFLICT (see page 152)	Device oin conflict macro
STATIC (see page 152)	External data/function define
SUCCESS (see page 153)	SUCCESS macro
TRUE (see page 153)	TRUE macro

Types

Type	Description
BOOL_16 (see page 60)	16 bit boolean type
BOOL_32 (see page 60)	32 bit boolean type
BOOL_8 (see page 60)	8 bit boolean type
CHAR (see page 60)	SMA type for character type
INT_16 (see page 65)	SMA type for 16 bit signed value
INT_32 (see page 66)	SMA type for 32 bit signed value
INT_64 (see page 66)	SMA type for 64 bit signed value
INT_8 (see page 66)	SMA type for 8 bit signed value
PFI (see page 70)	Pointer to Function returning INT_32 (see page 66) (any number of parameters)
PFV (see page 70)	Pointer to Function returning Void (any number of parameters)
STATUS (see page 71)	Status type
UNS_16 (see page 73)	SMA type for 16 bit unsigned value
UNS_32 (see page 73)	SMA type for 32 bit unsigned value
UNS_64 (see page 73)	SMA type for 64 bit unsigned value
UNS_8 (see page 74)	SMA type for 8 bit unsigned value

1.6.33 lpc_winfreesystem14x16.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convfnt.exe Windows FreeSystem 14x16 Font

Variables

Variable	Description
font_winfreesys14x16 (see page 75)	Externally available font information structure
winfreesystem14x16_bits (see page 90)	This is variable winfreesystem14x16_bits.
winfreesystem14x16_width (see page 95)	Character width data.

1.6.34 lpc_winfreesystem14x16.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Windows FreeSystem 14x16 Font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_WINFREESYS_14X16_FONT_H (see page 141)	This is macro LPC_WINFREESYS_14X16_FONT_H.

1.6.35 lpc_x5x7.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convbdf on Tue Oct 3 00:24:24 MDT 2000. Font information:

name: "-Misc-Fixed-Medium-R-Normal--7-70-75-75-C-50-ISO8859-1" pixel size: 7 ascent: 6 descent: 1

Variables

Variable	Description
font_x5x7 (see page 76)	Externally available font information structure
x5x7_bits (see page 95)	Font character bitmap data.
x5x7_width (see page 97)	Character width data.

1.6.36 lpc_x5x7.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Fixed 5x7 proportional font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or

- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_X5X7_FONT_H (see page 142)	This is macro LPC_X5X7_FONT_H.

1.6.37 lpc_x6x13.c

$Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $ Generated by convbdf on Tue Oct 3 00:24:25 MDT 2000. Font information:

name: "-Misc-Fixed-Medium-R-SemiCondensed--13-120-75-75-C-60-ISO8859-1" pixel size: 13 ascent: 11 descent: 2

Variables

Variable	Description
font_x6x13 (see page 76)	Externally available font information structure
x6x13_bits (see page 97)	Font character bitmap data.
x6x13_width (see page 99)	Character width data.

1.6.38 lpc_x6x13.h

- $Id:: lpc_api.c (see page 154) 4 2007-08-23 00:08:42Z kevinw $

*

- Project: Fixed 6x13 proportional font

*

- Description:
- This package provides bit information for a font type.

****

- Software that is described herein is for illustrative purposes only
- which provides customers with programming information regarding the
- products. This software is supplied "AS IS" without any warranties.
- NXP Semiconductors assumes no responsibility or liability for the
- use of the software, conveys no license or title under any patent,
- copyright, or mask work right to the product. NXP Semiconductors
- reserves the right to make changes in the software without
- notification. NXP Semiconductors also make no representation or
- warranty that such application will be suitable for the specified
- use without further testing or modification.

Macros

Macro	Description
LPC_X6X13_FONT_H (see page 142)	This is macro LPC_X6X13_FONT_H.

Index

B

C

D

E

F

M

N

O

P

R

S

T

U

V

W

X

Y