

intLib

Generated by Doxygen 1.8.6

Mon Mar 31 2014 16:47:50



# Contents

<b>1</b>	<b>uIntPLib</b>	<b>1</b>
<b>2</b>	<b>Data Structure Index</b>	<b>3</b>
2.1	Data Structures . . . . .	3
<b>3</b>	<b>File Index</b>	<b>5</b>
3.1	File List . . . . .	5
<b>4</b>	<b>Data Structure Documentation</b>	<b>7</b>
4.1	CommandInstance Struct Reference . . . . .	7
4.1.1	Detailed Description . . . . .	7
4.1.2	Field Documentation . . . . .	7
4.1.2.1	charIn . . . . .	7
4.1.2.2	charOut . . . . .	7
4.1.2.3	charOutPtr . . . . .	7
4.1.2.4	cmdBuffer . . . . .	7
4.2	IRInstance Struct Reference . . . . .	8
4.2.1	Detailed Description . . . . .	8
4.2.2	Field Documentation . . . . .	8
4.2.2.1	CarrierFrequency . . . . .	8
4.2.2.2	CarrierPeriod . . . . .	8
4.2.2.3	LastData . . . . .	8
4.2.2.4	Mode . . . . .	8
4.2.2.5	Pulses . . . . .	8
4.2.2.6	ReceiveAddress . . . . .	8
4.2.2.7	ReceiveBuffer . . . . .	9
4.2.2.8	RxPin . . . . .	9
4.2.2.9	RxPort . . . . .	9
4.2.2.10	TxPin . . . . .	9
4.2.2.11	TxPort . . . . .	9
4.3	LCDStatus Struct Reference . . . . .	9
4.3.1	Detailed Description . . . . .	9

4.3.2	Field Documentation	9
4.3.2.1	cgramAdress	9
4.3.2.2	col	9
4.3.2.3	display	10
4.3.2.4	row	10
4.3.2.5	shift	10
4.3.2.6	specialChar	10
4.4	UARTInstance Struct Reference	10
4.4.1	Detailed Description	10
4.4.2	Field Documentation	10
4.4.2.1	Mode	10
4.4.2.2	RxBuffer	10
4.4.2.3	RxBufferPtr	10
4.4.2.4	TxBuffer	11
4.4.2.5	TxBufferPtr	11
4.4.2.6	TxLastSent	11
4.4.2.7	TxLastSentPtr	11
<b>5</b>	<b>File Documentation</b>	<b>13</b>
5.1	depl_spc/chip_specific.c File Reference	13
5.2	chip_specific.c	13
5.3	depl_spc/chip_specific.h File Reference	13
5.4	chip_specific.h	13
5.5	depl_spc/cmd_list.h File Reference	14
5.6	cmd_list.h	14
5.7	depl_spc/device_init/hardwareInit.c File Reference	14
5.7.1	Function Documentation	14
5.7.1.1	HardwareInit	14
5.8	hardwareInit.c	14
5.9	depl_spc/device_init/hardwareInit.h File Reference	14
5.9.1	Function Documentation	15
5.9.1.1	HardwareInit	15
5.10	hardwareInit.h	15
5.11	depl_spc/device_init/softwareInit.c File Reference	15
5.11.1	Function Documentation	15
5.11.1.1	SoftwareInit	15
5.12	softwareInit.c	15
5.13	depl_spc/device_init/softwareInit.h File Reference	16
5.13.1	Function Documentation	16
5.13.1.1	SoftwareInit	16

5.14	softwareInit.h	16
5.15	depl_spc/globalParam.h File Reference	16
5.15.1	Macro Definition Documentation	17
5.15.1.1	BUS_CLOCK	17
5.15.1.2	BUSHZ_CLOCK	17
5.15.1.3	CPU_CLOCK	17
5.15.1.4	CPUHZ_CLOCK	17
5.15.1.5	LCD_SPLASHSCREEN	17
5.15.1.6	LCD_SPLASHSCREEN1	17
5.15.1.7	PROJECT_NAME	17
5.16	globalParam.h	17
5.17	depl_spc/includeAll_hw.h File Reference	18
5.18	includeAll_hw.h	18
5.19	depl_spc/includeAll_sw.h File Reference	18
5.20	includeAll_sw.h	18
5.21	depl_spc/lib_comp/external_cons.h File Reference	19
5.21.1	Macro Definition Documentation	19
5.21.1.1	LCD_char_heigh	19
5.21.1.2	LCD_char_width	19
5.21.1.3	LCD_CLK_Pin	19
5.21.1.4	LCD_CLK_Port	19
5.21.1.5	LCD_col_num	19
5.21.1.6	LCD_DTA_Pin	19
5.21.1.7	LCD_DTA_Port	20
5.21.1.8	LCD_EN_Pin	20
5.21.1.9	LCD_EN_Port	20
5.21.1.10	LCD_row_num	20
5.21.1.11	LCD_RS_Pin	20
5.21.1.12	LCD_RS_Port	20
5.22	external_cons.h	20
5.23	depl_spc/lib_comp/libraryCompatible.h File Reference	21
5.23.1	Macro Definition Documentation	21
5.23.1.1	PinAddrClear	21
5.23.1.2	PinAddrSet	21
5.23.1.3	PinClear	21
5.23.1.4	PinSet	21
5.23.1.5	PinToggle	21
5.23.1.6	SysDelay	21
5.23.1.7	SysDelayMs	21
5.23.1.8	SysDelayUs	21

5.24	libraryCompatible.h	22
5.25	depl_spc/variables.h File Reference	22
5.25.1	Macro Definition Documentation	22
5.25.1.1	variable_h	22
5.26	variables.h	22
5.27	my_lib/ascii.h File Reference	22
5.27.1	Macro Definition Documentation	23
5.27.1.1	ASCII_ACK	23
5.27.1.2	ASCII_BEL	23
5.27.1.3	ASCII_BS	23
5.27.1.4	ASCII_CAN	23
5.27.1.5	ASCII_CR	23
5.27.1.6	ASCII_DC1	24
5.27.1.7	ASCII_DC2	24
5.27.1.8	ASCII_DC3	24
5.27.1.9	ASCII_DC4	24
5.27.1.10	ASCII_DLE	24
5.27.1.11	ASCII_EM	24
5.27.1.12	ASCII_ENQ	24
5.27.1.13	ASCII_EOT	24
5.27.1.14	ASCII_ESC	24
5.27.1.15	ASCII_ETB	24
5.27.1.16	ASCII_ETX	24
5.27.1.17	ASCII_FF	24
5.27.1.18	ASCII_FS	25
5.27.1.19	ASCII_GS	25
5.27.1.20	ASCII_HT	25
5.27.1.21	ASCII_LF	25
5.27.1.22	ASCII_NAK	25
5.27.1.23	ASCII_NULL	25
5.27.1.24	ASCII_RS	25
5.27.1.25	ASCII_SI	25
5.27.1.26	ASCII_SO	25
5.27.1.27	ASCII_SOH	25
5.27.1.28	ASCII_STX	25
5.27.1.29	ASCII_SUB	25
5.27.1.30	ASCII_SYN	26
5.27.1.31	ASCII_US	26
5.27.1.32	ASCII_VT	26
5.28	ascii.h	26

5.29 my_lib/cmd_sort.c File Reference . . . . .	27
5.29.1 Function Documentation . . . . .	27
5.29.1.1 CommandSort . . . . .	27
5.30 cmd_sort.c . . . . .	27
5.31 my_lib/cmd_sort.h File Reference . . . . .	27
5.31.1 Macro Definition Documentation . . . . .	28
5.31.1.1 MAX_BUFFER_SIZE . . . . .	28
5.31.2 Function Documentation . . . . .	28
5.31.2.1 CommandSort . . . . .	28
5.32 cmd_sort.h . . . . .	28
5.33 my_lib/gpioPin_masks.h File Reference . . . . .	28
5.33.1 Macro Definition Documentation . . . . .	29
5.33.1.1 IOPin_0 . . . . .	29
5.33.1.2 IOPin_1 . . . . .	29
5.33.1.3 IOPin_10 . . . . .	29
5.33.1.4 IOPin_11 . . . . .	29
5.33.1.5 IOPin_12 . . . . .	29
5.33.1.6 IOPin_13 . . . . .	29
5.33.1.7 IOPin_14 . . . . .	29
5.33.1.8 IOPin_15 . . . . .	30
5.33.1.9 IOPin_16 . . . . .	30
5.33.1.10 IOPin_17 . . . . .	30
5.33.1.11 IOPin_18 . . . . .	30
5.33.1.12 IOPin_19 . . . . .	30
5.33.1.13 IOPin_2 . . . . .	30
5.33.1.14 IOPin_20 . . . . .	30
5.33.1.15 IOPin_21 . . . . .	30
5.33.1.16 IOPin_22 . . . . .	30
5.33.1.17 IOPin_23 . . . . .	30
5.33.1.18 IOPin_24 . . . . .	30
5.33.1.19 IOPin_25 . . . . .	30
5.33.1.20 IOPin_26 . . . . .	31
5.33.1.21 IOPin_27 . . . . .	31
5.33.1.22 IOPin_28 . . . . .	31
5.33.1.23 IOPin_29 . . . . .	31
5.33.1.24 IOPin_3 . . . . .	31
5.33.1.25 IOPin_30 . . . . .	31
5.33.1.26 IOPin_31 . . . . .	31
5.33.1.27 IOPin_4 . . . . .	31
5.33.1.28 IOPin_5 . . . . .	31

5.33.1.29 IOPin_6	31
5.33.1.30 IOPin_7	31
5.33.1.31 IOPin_8	31
5.33.1.32 IOPin_9	32
5.34 gpioPin_masks.h	32
5.35 my_lib/ir.c File Reference	32
5.35.1 Function Documentation	32
5.35.1.1 IRByteBySoftware	32
5.35.1.2 IRInit	33
5.35.1.3 IRRepeat	33
5.35.1.4 IRSend	33
5.36 ir.c	33
5.37 my_lib/ir.h File Reference	36
5.37.1 Macro Definition Documentation	37
5.37.1.1 IR_BY_EXTERNAL_TIEMR	37
5.37.1.2 IR_BY_SOFTWARE	37
5.37.1.3 IR_BY_TIMER	37
5.37.1.4 IR_BY_UART	37
5.37.1.5 IR_MAX_INSTANCES	37
5.37.1.6 IR_MY_PROTOCOL	37
5.37.1.7 IR_NEC_EXTENDED	37
5.37.1.8 IR_NEC_PROTOCOL	37
5.37.1.9 IR_RC5_PROTOCOL	37
5.37.1.10 IR_REPEAT_COMMAND_DISABLE	37
5.37.1.11 IR_REPEAT_COMMAND_ENABLE	38
5.37.1.12 IRDelay	38
5.37.1.13 IRDelayMs	38
5.37.1.14 IRDelayUs	38
5.37.1.15 IRPinClear	38
5.37.1.16 IRPinSet	38
5.37.1.17 NEC_PULSE_TIME	38
5.37.1.18 RC5_PULSE_TIME	38
5.37.2 Function Documentation	38
5.37.2.1 IRByteBySoftware	38
5.37.2.2 IRInit	38
5.37.2.3 IRRepeat	38
5.37.2.4 IRSend	38
5.38 ir.h	39
5.39 my_lib/lcd.c File Reference	40
5.39.1 Macro Definition Documentation	41



5.39.1.1	<a href="#">false</a>	41
5.39.1.2	<a href="#">lcd_vector_index</a>	41
5.39.1.3	<a href="#">true</a>	41
5.39.1.4	<a href="#">trueDefinedLCD</a>	41
5.39.2	<a href="#">Function Documentation</a>	41
5.39.2.1	<a href="#">arrayToNum</a>	41
5.39.2.2	<a href="#">LCDClear</a>	41
5.39.2.3	<a href="#">LCDDisplayOn</a>	41
5.39.2.4	<a href="#">LCDHome</a>	41
5.39.2.5	<a href="#">LCDInit</a>	42
5.39.2.6	<a href="#">LCDPosition</a>	42
5.39.2.7	<a href="#">LCDPositionNoDelay</a>	42
5.39.2.8	<a href="#">LCDRegisterSpecial</a>	42
5.39.2.9	<a href="#">LCDSend</a>	42
5.39.2.10	<a href="#">LCDSendChar</a>	42
5.39.2.11	<a href="#">LCDSendCmd</a>	42
5.39.2.12	<a href="#">LCDSendHex</a>	42
5.39.2.13	<a href="#">LCDSendNum</a>	42
5.39.2.14	<a href="#">LCDSendNumArray</a>	42
5.39.2.15	<a href="#">LCDSendNumStrict</a>	43
5.39.2.16	<a href="#">LCDSendString</a>	43
5.39.2.17	<a href="#">LCDSendVU</a>	43
5.39.2.18	<a href="#">LCDShift</a>	43
5.39.2.19	<a href="#">numToArray</a>	43
5.39.3	<a href="#">Variable Documentation</a>	43
5.39.3.1	<a href="#">LCD_CmdInit_Vector</a>	43
5.39.3.2	<a href="#">LCD_InitDelay_Vector</a>	43
5.40	<a href="#">lcd.c</a>	43
5.41	<a href="#">my_lib/lcd.h File Reference</a>	49
5.41.1	<a href="#">Macro Definition Documentation</a>	50
5.41.1.1	<a href="#">LCD_BLINK_OFF</a>	50
5.41.1.2	<a href="#">LCD_BLINK_ON</a>	50
5.41.1.3	<a href="#">LCD_CLK_High</a>	51
5.41.1.4	<a href="#">LCD_CLK_LoW</a>	51
5.41.1.5	<a href="#">LCD_CURSOR_OFF</a>	51
5.41.1.6	<a href="#">LCD_CURSOR_ON</a>	51
5.41.1.7	<a href="#">LCD_DISPLAY_CONFIG</a>	51
5.41.1.8	<a href="#">LCD_DISPLAY_INCREMENT</a>	51
5.41.1.9	<a href="#">LCD_DISPLAY_OFF</a>	51
5.41.1.10	<a href="#">LCD_DISPLAY_ON</a>	51

5.41.1.11 LCD_DTA_Send . . . . .	51
5.41.1.12 LCD_EN_High . . . . .	51
5.41.1.13 LCD_EN_Low . . . . .	51
5.41.1.14 LCD_INCREMENT . . . . .	52
5.41.1.15 LCD_INCREMENT_NEGATIVE . . . . .	52
5.41.1.16 LCD_INCREMENT_NO_SHIFT . . . . .	52
5.41.1.17 LCD_INCREMENT_POSITIVE . . . . .	52
5.41.1.18 LCD_INCREMENT_SHIFT . . . . .	52
5.41.1.19 LCD_RS_High . . . . .	52
5.41.1.20 LCD_RS_Low . . . . .	52
5.41.1.21 LCD_SET_CGRAM . . . . .	52
5.41.1.22 LCD_SHIFT . . . . .	52
5.41.1.23 LCD_SHIFT_CURSOR . . . . .	52
5.41.1.24 LCD_SHIFT_DISPLAY . . . . .	52
5.41.1.25 LCD_SHIFT_LEFT . . . . .	52
5.41.1.26 LCD_SHIFT_RIGHT . . . . .	53
5.41.1.27 LCD_splashscreen2_row1 . . . . .	53
5.41.1.28 LCD_splashscreen2_row2 . . . . .	53
5.41.1.29 LCD_splashscreen_row1 . . . . .	53
5.41.1.30 LCD_splashscreen_row2 . . . . .	53
5.41.1.31 LCDDelay . . . . .	53
5.41.1.32 LCDPinClear . . . . .	53
5.41.1.33 LCDPinSet . . . . .	53
5.41.1.34 maxLengthOut . . . . .	53
5.41.2 Function Documentation . . . . .	53
5.41.2.1 arrayToNum . . . . .	53
5.41.2.2 LCDClear . . . . .	53
5.41.2.3 LCDHome . . . . .	54
5.41.2.4 LCDInit . . . . .	54
5.41.2.5 LCDPosition . . . . .	54
5.41.2.6 LCDPositionNoDelay . . . . .	54
5.41.2.7 LCDRegisterSpecial . . . . .	54
5.41.2.8 LCDSend . . . . .	54
5.41.2.9 LCDSendChar . . . . .	54
5.41.2.10 LCDSendCmd . . . . .	54
5.41.2.11 LCDSendHex . . . . .	54
5.41.2.12 LCDSendNum . . . . .	54
5.41.2.13 LCDSendNumArray . . . . .	55
5.41.2.14 LCDSendNumStrict . . . . .	55
5.41.2.15 LCDSendString . . . . .	55

5.41.2.16 LCDSendVU . . . . .	55
5.41.2.17 LCDShift . . . . .	55
5.41.2.18 numToArray . . . . .	55
5.41.3 Variable Documentation . . . . .	55
5.41.3.1 LCD0Status . . . . .	55
5.42 lcd.h . . . . .	55
5.43 my_lib/my_use.c File Reference . . . . .	57
5.43.1 Function Documentation . . . . .	57
5.43.1.1 ShiftSerialSend . . . . .	57
5.44 my_use.c . . . . .	57
5.45 my_lib/my_use.h File Reference . . . . .	58
5.45.1 Macro Definition Documentation . . . . .	59
5.45.1.1 bTrue0 . . . . .	59
5.45.1.2 bTrue1 . . . . .	59
5.45.1.3 bTrue2 . . . . .	59
5.45.1.4 bTrue3 . . . . .	59
5.45.1.5 bTrue4 . . . . .	59
5.45.1.6 bTrue5 . . . . .	59
5.45.1.7 bTrue6 . . . . .	59
5.45.1.8 bTrue7 . . . . .	59
5.45.1.9 charBinaryLength . . . . .	59
5.45.1.10 charDecadeLength . . . . .	59
5.45.1.11 intBinaryLength . . . . .	59
5.45.1.12 intDecadeLength . . . . .	59
5.45.1.13 limitCeilValue . . . . .	60
5.45.1.14 limitCycleValueUpOff . . . . .	60
5.45.1.15 limitCycleValueUpZero . . . . .	60
5.45.1.16 limitFloorValue . . . . .	60
5.45.1.17 shortBinaryLength . . . . .	60
5.45.1.18 shortDecadeLength . . . . .	60
5.45.2 Function Documentation . . . . .	60
5.45.2.1 ShiftSerialSend . . . . .	60
5.46 my_use.h . . . . .	61
5.47 my_lib/myUart.c File Reference . . . . .	61
5.47.1 Function Documentation . . . . .	61
5.47.1.1 myUARTSendString . . . . .	61
5.48 myUart.c . . . . .	62
5.49 my_lib/myUart.h File Reference . . . . .	62
5.49.1 Macro Definition Documentation . . . . .	62
5.49.1.1 myUARTDelay . . . . .	62

5.49.1.2	myUARTPC	63
5.49.1.3	myUARTPCSend	63
5.49.1.4	myUARTSend	63
5.49.1.5	UART_BUFFER_SIZE	63
5.49.1.6	UART_DIRECT_TRANSFER_MODE	63
5.49.1.7	UART_NORMAL_OP_MODE	63
5.49.2	Function Documentation	63
5.49.2.1	myUARTSendString	63
5.50	myUart.h	63
5.51	my_lib/uk_mapping.h File Reference	64
5.51.1	Macro Definition Documentation	64
5.51.1.1	UKM_ASCII_FF	64
5.51.1.2	UKM_ASCII_LF	64
5.51.1.3	UKM_ASCII_TAB	64
5.51.1.4	UKM_ASCII_VT	65
5.51.1.5	UKM_BS	65
5.51.1.6	UKM_BSPACE	65
5.51.1.7	UKM_CLS	65
5.51.1.8	UKM_CR	65
5.51.1.9	UKM_CTRL_E	65
5.51.1.10	UKM_ENTER	65
5.51.1.11	UKM_ESCAPE	65
5.51.1.12	UKM_LF	65
5.51.1.13	UKM_LINEFEED	65
5.51.1.14	UKM_SPACE	65
5.51.1.15	UKM_TAB	65
5.51.1.16	UKM_TILDA	66
5.51.1.17	UKM_VT	66
5.52	uk_mapping.h	66
5.53	README.md File Reference	66
5.54	README.md	66
<b>Index</b>		<b>67</b>

# Chapter 1

## uIntPLib

Universal Integrated Peripheral Library

This is a library made with functions masks to medium level programming. Intended to make code more portable, while maintaining its performance.



## Chapter 2

# Data Structure Index

### 2.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">CommandInstance</a>	7
<a href="#">IRInstance</a>	8
<a href="#">LCDStatus</a>	9
<a href="#">UARTInstance</a>	10





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

depl_spc/chip_specific.c	13
depl_spc/chip_specific.h	13
depl_spc/cmd_list.h	14
depl_spc/globalParam.h	17
depl_spc/includeAll_hw.h	18
depl_spc/includeAll_sw.h	18
depl_spc/variables.h	22
depl_spc/device_init/hardwareInit.c	14
depl_spc/device_init/hardwareInit.h	15
depl_spc/device_init/softwareInit.c	15
depl_spc/device_init/softwareInit.h	16
depl_spc/lib_comp/external_cons.h	20
depl_spc/lib_comp/libraryCompatible.h	22
my_lib/ascii.h	26
my_lib/cmd_sort.c	27
my_lib/cmd_sort.h	28
my_lib/gpioPin_masks.h	32
my_lib/ir.c	33
my_lib/ir.h	39
my_lib/lcd.c	43
my_lib/lcd.h	55
my_lib/my_use.c	57
my_lib/my_use.h	61
my_lib/myUart.c	62
my_lib/myUart.h	63
my_lib/uk_mapping.h	66



## Chapter 4

# Data Structure Documentation

### 4.1 CommandInstance Struct Reference

```
#include <cmd_sort.h>
```

#### Data Fields

- [uint8\\_t charIn](#)
- [uint8\\_t cmdBuffer \[MAX\\_BUFFER\\_SIZE\]](#)
- [uint16\\_t charOut \[MAX\\_BUFFER\\_SIZE\]](#)
- [uint8\\_t charOutPtr](#)

#### 4.1.1 Detailed Description

Definition at line 16 of file [cmd\\_sort.h](#).

#### 4.1.2 Field Documentation

##### 4.1.2.1 [uint8\\_t CommandInstance::charIn](#)

Definition at line 17 of file [cmd\\_sort.h](#).

##### 4.1.2.2 [uint16\\_t CommandInstance::charOut\[MAX\\_BUFFER\\_SIZE\]](#)

Definition at line 19 of file [cmd\\_sort.h](#).

##### 4.1.2.3 [uint8\\_t CommandInstance::charOutPtr](#)

Definition at line 20 of file [cmd\\_sort.h](#).

##### 4.1.2.4 [uint8\\_t CommandInstance::cmdBuffer\[MAX\\_BUFFER\\_SIZE\]](#)

Definition at line 18 of file [cmd\\_sort.h](#).

The documentation for this struct was generated from the following file:

- [my\\_lib/cmd\\_sort.h](#)

## 4.2 IRInstance Struct Reference

```
#include <ir.h>
```

### Data Fields

- uint16\_t [Mode](#)
- uint8\_t [CarrierFrequency](#)
- uint16\_t [CarrierPeriod](#)
- uint32\_t [TxPin](#)
- uint32\_t [TxPort](#)
- uint32\_t [RxPin](#)
- uint32\_t [RxPort](#)
- uint16\_t [ReceiveAddress](#)
- uint16\_t [ReceiveBuffer](#)
- uint16\_t [Pulses](#)
- uint8\_t [LastData](#)

### 4.2.1 Detailed Description

Definition at line [87](#) of file [ir.h](#).

### 4.2.2 Field Documentation

#### 4.2.2.1 uint8\_t IRInstance::CarrierFrequency

Definition at line [89](#) of file [ir.h](#).

#### 4.2.2.2 uint16\_t IRInstance::CarrierPeriod

Definition at line [90](#) of file [ir.h](#).

#### 4.2.2.3 uint8\_t IRInstance::LastData

Definition at line [98](#) of file [ir.h](#).

#### 4.2.2.4 uint16\_t IRInstance::Mode

Definition at line [88](#) of file [ir.h](#).

#### 4.2.2.5 uint16\_t IRInstance::Pulses

Definition at line [97](#) of file [ir.h](#).

#### 4.2.2.6 uint16\_t IRInstance::ReceiveAddress

Definition at line [95](#) of file [ir.h](#).

#### 4.2.2.7 uint16\_t IRInstance::ReceiveBuffer

Definition at line 96 of file [ir.h](#).

#### 4.2.2.8 uint32\_t IRInstance::RxPin

Definition at line 93 of file [ir.h](#).

#### 4.2.2.9 uint32\_t IRInstance::RxPort

Definition at line 94 of file [ir.h](#).

#### 4.2.2.10 uint32\_t IRInstance::TxPin

Definition at line 91 of file [ir.h](#).

#### 4.2.2.11 uint32\_t IRInstance::TxPort

Definition at line 92 of file [ir.h](#).

The documentation for this struct was generated from the following file:

- [my\\_lib/ir.h](#)

## 4.3 LCDStatus Struct Reference

```
#include <lcd.h>
```

### Data Fields

- [uint8\\_t row](#)
- [uint8\\_t col](#)
- [uint8\\_t display](#)
- [uint8\\_t shift](#)
- [uint8\\_t cgramAdress](#)
- [uint8\\_t specialChar](#) [8]

#### 4.3.1 Detailed Description

Definition at line 120 of file [lcd.h](#).

#### 4.3.2 Field Documentation

##### 4.3.2.1 uint8\_t LCDStatus::cgramAdress

Definition at line 126 of file [lcd.h](#).

##### 4.3.2.2 uint8\_t LCDStatus::col

Definition at line 123 of file [lcd.h](#).

#### 4.3.2.3 uint8\_t LCDStatus::display

Definition at line 124 of file [lcd.h](#).

#### 4.3.2.4 uint8\_t LCDStatus::row

Definition at line 122 of file [lcd.h](#).

#### 4.3.2.5 uint8\_t LCDStatus::shift

Definition at line 125 of file [lcd.h](#).

#### 4.3.2.6 uint8\_t LCDStatus::specialChar[8]

Definition at line 127 of file [lcd.h](#).

The documentation for this struct was generated from the following file:

- [my\\_lib/lcd.h](#)

## 4.4 UARTInstance Struct Reference

```
#include <myUart.h>
```

### Data Fields

- uint8\_t [RxBuffer](#) [[UART\\_BUFFER\\_SIZE](#)]
- uint8\_t [RxBufferPtr](#)
- uint8\_t [TxBuffer](#) [[UART\\_BUFFER\\_SIZE](#)]
- uint8\_t [TxBufferPtr](#)
- uint16\_t [Mode](#)
- uint8\_t [TxLastSent](#) [[UART\\_BUFFER\\_SIZE](#)]
- uint8\_t [TxLastSentPtr](#)

#### 4.4.1 Detailed Description

Definition at line 25 of file [myUart.h](#).

#### 4.4.2 Field Documentation

##### 4.4.2.1 uint16\_t UARTInstance::Mode

Definition at line 30 of file [myUart.h](#).

##### 4.4.2.2 uint8\_t UARTInstance::RxBuffer[UART\_BUFFER\_SIZE]

Definition at line 26 of file [myUart.h](#).

##### 4.4.2.3 uint8\_t UARTInstance::RxBufferPtr

Definition at line 27 of file [myUart.h](#).

#### 4.4.2.4 uint8\_t UARTInstance::TxBuffer[UART\_BUFFER\_SIZE]

Definition at line 28 of file [myUart.h](#).

#### 4.4.2.5 uint8\_t UARTInstance::TxBufferPtr

Definition at line 29 of file [myUart.h](#).

#### 4.4.2.6 uint8\_t UARTInstance::TxLastSent[UART\_BUFFER\_SIZE]

Definition at line 31 of file [myUart.h](#).

#### 4.4.2.7 uint8\_t UARTInstance::TxLastSentPtr

Definition at line 32 of file [myUart.h](#).

The documentation for this struct was generated from the following file:

- [my\\_lib/myUart.h](#)





## Chapter 5

# File Documentation

### 5.1 depl\_spc/chip\_specific.c File Reference

```
#include "chip_specific.h"
```

### 5.2 chip\_specific.c

```
00001 /*
00002  * chip_specific.c
00003  *
00004  * Created on: Mar 25, 2014
00005  * Author: rikardo
00006  */
00007
00008
00009 #include "chip_specific.h"
00010
00011
00012
00013
00014
```

### 5.3 depl\_spc/chip\_specific.h File Reference

```
#include "includeAll_sw.h"
#include "includeAll_hw.h"
```

### 5.4 chip\_specific.h

```
00001 /*
00002  * chip_specific.h
00003  *
00004  * Created on: Mar 25, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef CHIP_SPECIFIC_H_
00009 #define CHIP_SPECIFIC_H_
00010
00011 #include "includeAll_sw.h"
00012 #include "includeAll_hw.h"
00013
00014
00015
00016 #endif /* CHIP_SPECIFIC_H_ */
```

## 5.5 depl\_spc/cmd\_list.h File Reference

## 5.6 cmd\_list.h

```
00001 /*
00002  * cmd_list.h
00003  *
00004  * Created on: Mar 25, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef CMD_LIST_H_
00009 #define CMD_LIST_H_
00010
00011
00012
00013 #endif /* CMD_LIST_H_ */
```

## 5.7 depl\_spc/device\_init/hardwareInit.c File Reference

```
#include "hardwareInit.h"
```

### Functions

- void [HardwareInit](#) (void)

### 5.7.1 Function Documentation

#### 5.7.1.1 void HardwareInit ( void )

Definition at line 13 of file [hardwareInit.c](#).

## 5.8 hardwareInit.c

```
00001 /*
00002  * hardwareInit.c
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008
00009
00010 #include "hardwareInit.h"
00011
00012
00013 void HardwareInit(void)
00014 {
00015
00016 }
00017
00018
00019
```

## 5.9 depl\_spc/device\_init/hardwareInit.h File Reference

```
#include "depl_spc/includeAll_hw.h"
```

## Functions

- void [HardwareInit](#) (void)

### 5.9.1 Function Documentation

#### 5.9.1.1 void HardwareInit ( void )

Definition at line 13 of file [hardwareInit.c](#).

## 5.10 hardwareInit.h

```
00001 /*
00002  * hardwareInit.h
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef HARDWAREINIT_H_
00009 #define HARDWAREINIT_H_
00010
00011 #include "depl_spc/includeAll_hw.h"
00012
00013
00014
00015
00016
00017
00018
00019 void HardwareInit(void);
00020
00021 #endif /* HARDWAREINIT_H_ */
```

## 5.11 depl\_spc/device\_init/softwareInit.c File Reference

```
#include "softwareInit.h"
```

## Functions

- void [SoftwareInit](#) (void)

### 5.11.1 Function Documentation

#### 5.11.1.1 void SoftwareInit ( void )

Definition at line 12 of file [softwareInit.c](#).

## 5.12 softwareInit.c

```
00001 /*
00002  * softwareInit.c
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008 #include "softwareInit.h"
00009
00010
```

```

00011
00012 void SoftwareInit(void)
00013 {
00014
00015 }
00016

```

## 5.13 depl\_spc/device\_init/softwareInit.h File Reference

```
#include "depl_spc/includeAll_sw.h"
```

### Functions

- void [SoftwareInit](#) (void)

### 5.13.1 Function Documentation

#### 5.13.1.1 void SoftwareInit ( void )

Definition at line 12 of file [softwareInit.c](#).

## 5.14 softwareInit.h

```

00001 /*
00002  * softwareInit.h
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef SOFTWAREINIT_H_
00009 #define SOFTWAREINIT_H_
00010
00011 #include "depl_spc/includeAll_sw.h"
00012
00013
00014
00015
00016
00017
00018 void SoftwareInit(void);
00019
00020 #endif /* SOFTWAREINIT_H_ */

```

## 5.15 depl\_spc/globalParam.h File Reference

### Macros

- #define [PROJECT\\_NAME](#) ("your projects name here")
- #define [LCD\\_SPLASHSCREEN1](#) 1
- #define [LCD\\_SPLASHSCREEN](#) 1
- #define [CPU\\_CLOCK](#) 48
- #define [BUS\\_CLOCK](#) [CPU\\_CLOCK](#)/2
- #define [CPUHZ\\_CLOCK](#) 48000000
- #define [BUSHZ\\_CLOCK](#) [CPUHZ\\_CLOCK](#)/2

### 5.15.1 Macro Definition Documentation

#### 5.15.1.1 #define BUS\_CLOCK CPU\_CLOCK/2

Definition at line 19 of file [globalParam.h](#).

#### 5.15.1.2 #define BUSHZ\_CLOCK CPUHZ\_CLOCK/2

Definition at line 21 of file [globalParam.h](#).

#### 5.15.1.3 #define CPU\_CLOCK 48

Definition at line 18 of file [globalParam.h](#).

#### 5.15.1.4 #define CPUHZ\_CLOCK 48000000

Definition at line 20 of file [globalParam.h](#).

#### 5.15.1.5 #define LCD\_SPLASHSCREEN 1

Definition at line 15 of file [globalParam.h](#).

#### 5.15.1.6 #define LCD\_SPLASHSCREEN1 1

Definition at line 14 of file [globalParam.h](#).

#### 5.15.1.7 #define PROJECT\_NAME ("your projects name here")

Definition at line 13 of file [globalParam.h](#).

## 5.16 globalParam.h

```
00001 /*
00002  * globalParam.h
00003  *
00004  * Created on: Mar 26, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef GLOBALPARAM_H_
00009 #define GLOBALPARAM_H_
00010
00011
00012
00013 #define PROJECT_NAME ("your projects name here")
00014 #define LCD_SPLASHSCREEN1 1 //enables proejct name in 2 secs splash
00015 #define LCD_SPLASHSCREEN 1 //enables date and time of compilation
00016
00017
00018 #define CPU_CLOCK 48
00019 #define BUS_CLOCK CPU_CLOCK/2
00020 #define CPUHZ_CLOCK 48000000
00021 #define BUSHZ_CLOCK CPUHZ_CLOCK/2
00022
00023
00024
00025
00026
00027 #endif /* GLOBALPARAM_H_ */
```

## 5.17 depl\_spc/includeAll\_hw.h File Reference

```
#include "globalParam.h"
#include "depl_spc/device_init/hardwareInit.h"
#include "depl_spc/lib_comp/external_cons.h"
#include "chip_specific.h"
```

## 5.18 includeAll\_hw.h

```
00001 /*
00002  * includeAll_hw.h
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef INCLUDEALL_HW_H_
00009 #define INCLUDEALL_HW_H_
00010
00011 //program definitions
00012 #include "globalParam.h"
00013
00014
00015 //masks for chip
00016
00017 //functions for peripherals
00018
00019
00020
00021 #include "depl_spc/device_init/hardwareInit.h"
00022 #include "depl_spc/lib_comp/external_cons.h"
00023 #include "chip_specific.h"
00024
00025
00026 #endif /* INCLUDEALL_HW_H_ */
```

## 5.19 depl\_spc/includeAll\_sw.h File Reference

```
#include "stdint.h"
#include "stdbool.h"
#include "depl_spc/lib_comp/libraryCompatible.h"
#include "my_use.h"
#include "lcd.h"
#include "depl_spc/device_init/softwareInit.h"
#include "depl_spc/device_init/hardwareInit.h"
```

## 5.20 includeAll\_sw.h

```
00001 /*
00002  * includeAll_sw.h
00003  *
00004  * Created on: Mar 25, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef INCLUDEALL_SW_H_
00009 #define INCLUDEALL_SW_H_
00010
00011 #include "stdint.h"
00012 #include "stdbool.h"
00013
00014 //basic low level functions
00015 #include "depl_spc/lib_comp/libraryCompatible.h"
00016 #include "my_use.h"
00017
```

```
00018 //external peripherals
00019 #include "lcd.h"
00020
00021
00022
00023 #include "depl_spc/device_init/softwareInit.h"
00024 #include "depl_spc/device_init/hardwareInit.h"
00025
00026 #endif /* INCLUDEALL_SW_H_ */
```

## 5.21 depl\_spc/lib\_comp/external\_cons.h File Reference

### Macros

- #define [LCD\\_RS\\_Port](#) PTE\_BASE\_PTR
- #define [LCD\\_RS\\_Pin](#) IOPin\_30
- #define [LCD\\_EN\\_Port](#) PTE\_BASE\_PTR
- #define [LCD\\_EN\\_Pin](#) IOPin\_29
- #define [LCD\\_DTA\\_Port](#) PTE\_BASE\_PTR
- #define [LCD\\_DTA\\_Pin](#) IOPin\_22
- #define [LCD\\_CLK\\_Port](#) PTE\_BASE\_PTR
- #define [LCD\\_CLK\\_Pin](#) IOPin\_23
- #define [LCD\\_row\\_num](#) 2
- #define [LCD\\_col\\_num](#) 16
- #define [LCD\\_char\\_heigh](#) 8
- #define [LCD\\_char\\_width](#) 5

### 5.21.1 Macro Definition Documentation

#### 5.21.1.1 #define [LCD\\_char\\_heigh](#) 8

Definition at line 29 of file [external\\_cons.h](#).

#### 5.21.1.2 #define [LCD\\_char\\_width](#) 5

Definition at line 30 of file [external\\_cons.h](#).

#### 5.21.1.3 #define [LCD\\_CLK\\_Pin](#) IOPin\_23

Definition at line 25 of file [external\\_cons.h](#).

#### 5.21.1.4 #define [LCD\\_CLK\\_Port](#) PTE\_BASE\_PTR

Definition at line 24 of file [external\\_cons.h](#).

#### 5.21.1.5 #define [LCD\\_col\\_num](#) 16

Definition at line 28 of file [external\\_cons.h](#).

#### 5.21.1.6 #define [LCD\\_DTA\\_Pin](#) IOPin\_22

Definition at line 22 of file [external\\_cons.h](#).

#### 5.21.1.7 `#define LCD_DTA_Port PTE_BASE_PTR`

Definition at line 21 of file [external\\_cons.h](#).

#### 5.21.1.8 `#define LCD_EN_Pin IOPin_29`

Definition at line 19 of file [external\\_cons.h](#).

#### 5.21.1.9 `#define LCD_EN_Port PTE_BASE_PTR`

Definition at line 18 of file [external\\_cons.h](#).

#### 5.21.1.10 `#define LCD_row_num 2`

Definition at line 27 of file [external\\_cons.h](#).

#### 5.21.1.11 `#define LCD_RS_Pin IOPin_30`

Definition at line 16 of file [external\\_cons.h](#).

#### 5.21.1.12 `#define LCD_RS_Port PTE_BASE_PTR`

Definition at line 15 of file [external\\_cons.h](#).

## 5.22 `external_cons.h`

```

00001 #ifndef external_cons_h
00002 #define external_cons_h
00003
00004
00005 /*
00006  * file used to declare masks to external peripherals
00007  *
00008  */
00009
00010
00011 /*
00012  * Definitions for LCD peripheral
00013  */
00014 //LCD
00015 #define LCD_RS_Port      PTE_BASE_PTR
00016 #define LCD_RS_Pin       IOPin_30
00017
00018 #define LCD_EN_Port      PTE_BASE_PTR
00019 #define LCD_EN_Pin       IOPin_29
00020
00021 #define LCD_DTA_Port      PTE_BASE_PTR
00022 #define LCD_DTA_Pin       IOPin_22
00023
00024 #define LCD_CLK_Port      PTE_BASE_PTR
00025 #define LCD_CLK_Pin       IOPin_23
00026
00027 #define LCD_row_num       2
00028 #define LCD_col_num       16
00029 #define LCD_char_heigh    8
00030 #define LCD_char_width    5
00031
00032
00033
00034 #endif//external_cons_h

```



## 5.23 depl\_spc/lib\_comp/libraryCompatible.h File Reference

```
#include "depl_spc/includeAll_hw.h"
#include "gpioPin_masks.h"
```

### Macros

- `#define PinSet(port, pin) (port##_PSOR = pin)`
- `#define PinClear(port, pin) (port##_PCOR = pin)`
- `#define PinToggle(port, pin) (port##_PTOR = pin)`
- `#define PinAddrSet(port, pin) (GPIO_PSOR_REG((GPIO_MemMapPtr)port) = pin)`
- `#define PinAddrClear(port, pin) (GPIO_PCOR_REG((GPIO_MemMapPtr)port) = pin)`
- `#define SysDelay(time) SysDelayFRDM(time)`
- `#define SysDelayUs(time) SysDelay((time*BUS_CLOCK)/6)`
- `#define SysDelayMs(time) SysDelayUs(time*1000)`

### 5.23.1 Macro Definition Documentation

5.23.1.1 `#define PinAddrClear( port, pin ) (GPIO_PCOR_REG((GPIO_MemMapPtr)port) = pin)`

Definition at line 22 of file [libraryCompatible.h](#).

5.23.1.2 `#define PinAddrSet( port, pin ) (GPIO_PSOR_REG((GPIO_MemMapPtr)port) = pin)`

Definition at line 21 of file [libraryCompatible.h](#).

5.23.1.3 `#define PinClear( port, pin ) (port##_PCOR = pin)`

Definition at line 17 of file [libraryCompatible.h](#).

5.23.1.4 `#define PinSet( port, pin ) (port##_PSOR = pin)`

Definition at line 16 of file [libraryCompatible.h](#).

5.23.1.5 `#define PinToggle( port, pin ) (port##_PTOR = pin)`

Definition at line 18 of file [libraryCompatible.h](#).

5.23.1.6 `#define SysDelay( time ) SysDelayFRDM(time)`

Definition at line 26 of file [libraryCompatible.h](#).

5.23.1.7 `#define SysDelayMs( time ) SysDelayUs(time*1000)`

Definition at line 28 of file [libraryCompatible.h](#).

5.23.1.8 `#define SysDelayUs( time ) SysDelay((time*BUS_CLOCK)/6)`

Definition at line 27 of file [libraryCompatible.h](#).

## 5.24 libraryCompatible.h

```

00001 /*
00002  * libraryCompatible.h
00003  *
00004  * Created on: Feb 5, 2014
00005  * Author: rikardo
00006  */
00007
00008 #ifndef LIBRARYCOMPATIBLE_H_
00009 #define LIBRARYCOMPATIBLE_H_
00010
00011 #include "depl_spc/includeAll_hw.h"
00012 #include "gpioPin_masks.h"
00013
00014
00015 //direct setting, uses a pre-casted object
00016 #define PinSet(port, pin) (port##_PSOR = pin)
00017 #define PinClear(port, pin) (port##_PCOR = pin)
00018 #define PinToggle(port, pin) (port##_PTOR = pin)
00019
00020 //casts the address to the structure referenced in the memory mapping file
00021 #define PinAddrSet(port, pin) (GPIO_PSOR_REG((GPIO_MemMapPtr)port) = pin)
00022 #define PinAddrClear(port, pin) (GPIO_PCOR_REG((GPIO_MemMapPtr)port) = pin)
00023
00024
00025
00026 #define SysDelay(time) SysDelayFRDM(time) //chip specific
00027 #define SysDelayUs(time) SysDelay((time*BUS_CLOCK)/6)
00028 #define SysDelayMs(time) SysDelayUs(time*1000)
00029
00030
00031
00032
00033
00034 #endif /* LIBRARYCOMPATIBLE_H_ */

```

## 5.25 depl\_spc/variables.h File Reference

### Macros

- #define [variable\\_h](#)

### 5.25.1 Macro Definition Documentation

#### 5.25.1.1 #define variable\_h

Definition at line 2 of file [variables.h](#).

## 5.26 variables.h

```

00001 #ifndef variables_h
00002 #define variable_h
00003
00004
00005
00006 #endif

```

## 5.27 my\_lib/ascii.h File Reference

### Macros

- #define [ASCII\\_NULL](#) 0
- #define [ASCII\\_SOH](#) 1
- #define [ASCII\\_STX](#) 2

- #define [ASCII\\_ETX](#) 3
- #define [ASCII\\_EOT](#) 4
- #define [ASCII\\_ENQ](#) 5
- #define [ASCII\\_ACK](#) 6
- #define [ASCII\\_BEL](#) 7
- #define [ASCII\\_BS](#) 8
- #define [ASCII\\_HT](#) 9
- #define [ASCII\\_LF](#) 10
- #define [ASCII\\_VT](#) 11
- #define [ASCII\\_FF](#) 12
- #define [ASCII\\_CR](#) 13
- #define [ASCII\\_SO](#) 14
- #define [ASCII\\_SI](#) 15
- #define [ASCII\\_DLE](#) 16
- #define [ASCII\\_DC1](#) 17
- #define [ASCII\\_DC2](#) 18
- #define [ASCII\\_DC3](#) 19
- #define [ASCII\\_DC4](#) 20
- #define [ASCII\\_NAK](#) 21
- #define [ASCII\\_SYN](#) 22
- #define [ASCII\\_ETB](#) 23
- #define [ASCII\\_CAN](#) 24
- #define [ASCII\\_EM](#) 25
- #define [ASCII\\_SUB](#) 26
- #define [ASCII\\_ESC](#) 27
- #define [ASCII\\_FS](#) 28
- #define [ASCII\\_GS](#) 29
- #define [ASCII\\_RS](#) 30
- #define [ASCII\\_US](#) 31

## 5.27.1 Macro Definition Documentation

### 5.27.1.1 #define ASCII\_ACK 6

Definition at line 22 of file [ascii.h](#).

### 5.27.1.2 #define ASCII\_BEL 7

Definition at line 23 of file [ascii.h](#).

### 5.27.1.3 #define ASCII\_BS 8

Definition at line 24 of file [ascii.h](#).

### 5.27.1.4 #define ASCII\_CAN 24

Definition at line 40 of file [ascii.h](#).

### 5.27.1.5 #define ASCII\_CR 13

Definition at line 29 of file [ascii.h](#).

5.27.1.6 `#define ASCII_DC1 17`

Definition at line 33 of file [ascii.h](#).

5.27.1.7 `#define ASCII_DC2 18`

Definition at line 34 of file [ascii.h](#).

5.27.1.8 `#define ASCII_DC3 19`

Definition at line 35 of file [ascii.h](#).

5.27.1.9 `#define ASCII_DC4 20`

Definition at line 36 of file [ascii.h](#).

5.27.1.10 `#define ASCII_DLE 16`

Definition at line 32 of file [ascii.h](#).

5.27.1.11 `#define ASCII_EM 25`

Definition at line 41 of file [ascii.h](#).

5.27.1.12 `#define ASCII_ENQ 5`

Definition at line 21 of file [ascii.h](#).

5.27.1.13 `#define ASCII_EOT 4`

Definition at line 20 of file [ascii.h](#).

5.27.1.14 `#define ASCII_ESC 27`

Definition at line 43 of file [ascii.h](#).

5.27.1.15 `#define ASCII_ETB 23`

Definition at line 39 of file [ascii.h](#).

5.27.1.16 `#define ASCII_ETX 3`

Definition at line 19 of file [ascii.h](#).

5.27.1.17 `#define ASCII_FF 12`

Definition at line 28 of file [ascii.h](#).

5.27.1.18 `#define ASCII_FS 28`

Definition at line 44 of file [ascii.h](#).

5.27.1.19 `#define ASCII_GS 29`

Definition at line 45 of file [ascii.h](#).

5.27.1.20 `#define ASCII_HT 9`

Definition at line 25 of file [ascii.h](#).

5.27.1.21 `#define ASCII_LF 10`

Definition at line 26 of file [ascii.h](#).

5.27.1.22 `#define ASCII_NAK 21`

Definition at line 37 of file [ascii.h](#).

5.27.1.23 `#define ASCII_NULL 0`

Definition at line 16 of file [ascii.h](#).

5.27.1.24 `#define ASCII_RS 30`

Definition at line 46 of file [ascii.h](#).

5.27.1.25 `#define ASCII_SI 15`

Definition at line 31 of file [ascii.h](#).

5.27.1.26 `#define ASCII_SO 14`

Definition at line 30 of file [ascii.h](#).

5.27.1.27 `#define ASCII_SOH 1`

Definition at line 17 of file [ascii.h](#).

5.27.1.28 `#define ASCII_STX 2`

Definition at line 18 of file [ascii.h](#).

5.27.1.29 `#define ASCII_SUB 26`

Definition at line 42 of file [ascii.h](#).

### 5.27.1.30 #define ASCII\_SYN 22

Definition at line 38 of file [ascii.h](#).

### 5.27.1.31 #define ASCII\_US 31

Definition at line 47 of file [ascii.h](#).

### 5.27.1.32 #define ASCII\_VT 11

Definition at line 27 of file [ascii.h](#).

## 5.28 ascii.h

```

00001 /*
00002  * ascii.h
00003  *
00004  * Created on: Nov 25, 2013
00005  * Author: rikardo
00006  */
00007
00008 #ifndef ASCII_H_
00009 #define ASCII_H_
00010
00011 /*
00012  * File containing ASCII Masks
00013  */
00014
00015
00016 #define ASCII_NULL 0 //Null Char
00017 #define ASCII_SOH 1 //Start of Header
00018 #define ASCII_STX 2 //Start of Text
00019 #define ASCII_ETX 3 //End of Text
00020 #define ASCII_EOT 4 //End of Transmission
00021 #define ASCII_ENQ 5 //Enquiry
00022 #define ASCII_ACK 6 //Ack
00023 #define ASCII_BEL 7 //Bell
00024 #define ASCII_BS 8 //BackSpace
00025 #define ASCII_HT 9 //Horizontal Tab
00026 #define ASCII_LF 10 //Line Feed
00027 #define ASCII_VT 11 //Vertical Tab
00028 #define ASCII_FF 12 //Form Feed
00029 #define ASCII_CR 13 //Carriage Return
00030 #define ASCII_SO 14 //Shift Out
00031 #define ASCII_SI 15 //Shift In
00032 #define ASCII_DLE 16 //Data Link Escape
00033 #define ASCII_DC1 17 //Device Control 1
00034 #define ASCII_DC2 18
00035 #define ASCII_DC3 19
00036 #define ASCII_DC4 20
00037 #define ASCII_NAK 21 //Negative Ack
00038 #define ASCII_SYN 22 //Synchronous idle
00039 #define ASCII_ETB 23 //End of Transmission Block
00040 #define ASCII_CAN 24 //Cancel
00041 #define ASCII_EM 25 //End of Medium
00042 #define ASCII_SUB 26 //Substitute
00043 #define ASCII_ESC 27 //Escape
00044 #define ASCII_FS 28 //File Separator
00045 #define ASCII_GS 29 //Group Separator
00046 #define ASCII_RS 30 //Record Separator
00047 #define ASCII_US 31 //Unit Separator
00048
00049
00050
00051
00052
00053
00054
00055 #endif /* ASCII_H_ */

```

## 5.29 my\_lib/cmd\_sort.c File Reference

```
#include "cmd_sort.h"
```

### Functions

- void [CommandSort](#) (uint8\_t \*cmdString)

#### 5.29.1 Function Documentation

##### 5.29.1.1 void CommandSort ( uint8\_t \* cmdString )

Definition at line 16 of file [cmd\\_sort.c](#).

## 5.30 cmd\_sort.c

```
00001 /*
00002  * cmd_sort.c
00003  *
00004  * Created on: Nov 28, 2013
00005  * Author: rikardo
00006  */
00007
00008 #include "cmd_sort.h"
00009
00010
00011 /*
00012  * Processes a string as a command
00013  * todo: make software interrupt for routines, call from here
00014  * todo: return function pointer
00015  */
00016 void CommandSort(uint8_t *cmdString)
00017 {
00018
00019 }
00020
00021
```

## 5.31 my\_lib/cmd\_sort.h File Reference

```
#include "depl_spc/includeAll_sw.h"
#include "depl_spc/cmd_list.h"
```

### Data Structures

- struct [CommandInstance](#)

### Macros

- #define [MAX\\_BUFFER\\_SIZE](#) 30

### Functions

- void [CommandSort](#) (uint8\_t \*cmdString)

### 5.31.1 Macro Definition Documentation

#### 5.31.1.1 #define MAX\_BUFFER\_SIZE 30

Definition at line 14 of file [cmd\\_sort.h](#).

### 5.31.2 Function Documentation

#### 5.31.2.1 void CommandSort ( uint8\_t \* cmdString )

Definition at line 16 of file [cmd\\_sort.c](#).

## 5.32 cmd\_sort.h

```

00001 /*
00002  * cmd_sort.h
00003  *
00004  * Created on: Nov 28, 2013
00005  * Author: rikardo
00006  */
00007
00008 #ifndef CMD_SORT_H_
00009 #define CMD_SORT_H_
00010
00011 #include "depl_spc/includeAll_sw.h"
00012 #include "depl_spc/cmd_list.h"
00013
00014 #define MAX_BUFFER_SIZE 30
00015
00016 typedef struct{
00017     uint8_t charIn;
00018     uint8_t cmdBuffer[MAX_BUFFER_SIZE];
00019     uint16_t charOut[MAX_BUFFER_SIZE];
00020     uint8_t charOutPtr;
00021 } CommandInstance;
00022
00023
00024
00025 void CommandSort(uint8_t *cmdString);
00026
00027
00028
00029
00030 #endif /* CMD_SORT_H_ */

```

## 5.33 my\_lib/gpioPin\_masks.h File Reference

### Macros

- #define [IOPin\\_0](#) 0x00000001
- #define [IOPin\\_1](#) 0x00000002
- #define [IOPin\\_2](#) 0x00000004
- #define [IOPin\\_3](#) 0x00000008
- #define [IOPin\\_4](#) 0x00000010
- #define [IOPin\\_5](#) 0x00000020
- #define [IOPin\\_6](#) 0x00000040
- #define [IOPin\\_7](#) 0x00000080
- #define [IOPin\\_8](#) 0x00000100
- #define [IOPin\\_9](#) 0x00000200
- #define [IOPin\\_10](#) 0x00000400
- #define [IOPin\\_11](#) 0x00000800
- #define [IOPin\\_12](#) 0x00001000
- #define [IOPin\\_13](#) 0x00002000



- `#define IOPin_14 0x00004000`
- `#define IOPin_15 0x00008000`
- `#define IOPin_16 0x00010000`
- `#define IOPin_17 0x00020000`
- `#define IOPin_18 0x00040000`
- `#define IOPin_19 0x00080000`
- `#define IOPin_20 0x00100000`
- `#define IOPin_21 0x00200000`
- `#define IOPin_22 0x00400000`
- `#define IOPin_23 0x00800000`
- `#define IOPin_24 0x01000000`
- `#define IOPin_25 0x02000000`
- `#define IOPin_26 0x04000000`
- `#define IOPin_27 0x08000000`
- `#define IOPin_28 0x10000000`
- `#define IOPin_29 0x20000000`
- `#define IOPin_30 0x40000000`
- `#define IOPin_31 0x08000000`

### 5.33.1 Macro Definition Documentation

#### 5.33.1.1 `#define IOPin_0 0x00000001`

Definition at line 5 of file [gpioPin\\_masks.h](#).

#### 5.33.1.2 `#define IOPin_1 0x00000002`

Definition at line 6 of file [gpioPin\\_masks.h](#).

#### 5.33.1.3 `#define IOPin_10 0x00000400`

Definition at line 15 of file [gpioPin\\_masks.h](#).

#### 5.33.1.4 `#define IOPin_11 0x00000800`

Definition at line 16 of file [gpioPin\\_masks.h](#).

#### 5.33.1.5 `#define IOPin_12 0x00001000`

Definition at line 17 of file [gpioPin\\_masks.h](#).

#### 5.33.1.6 `#define IOPin_13 0x00002000`

Definition at line 18 of file [gpioPin\\_masks.h](#).

#### 5.33.1.7 `#define IOPin_14 0x00004000`

Definition at line 19 of file [gpioPin\\_masks.h](#).

5.33.1.8 `#define IOPin_15 0x00008000`

Definition at line 20 of file [gpioPin\\_masks.h](#).

5.33.1.9 `#define IOPin_16 0x00010000`

Definition at line 21 of file [gpioPin\\_masks.h](#).

5.33.1.10 `#define IOPin_17 0x00020000`

Definition at line 22 of file [gpioPin\\_masks.h](#).

5.33.1.11 `#define IOPin_18 0x00040000`

Definition at line 23 of file [gpioPin\\_masks.h](#).

5.33.1.12 `#define IOPin_19 0x00080000`

Definition at line 24 of file [gpioPin\\_masks.h](#).

5.33.1.13 `#define IOPin_2 0x00000004`

Definition at line 7 of file [gpioPin\\_masks.h](#).

5.33.1.14 `#define IOPin_20 0x00100000`

Definition at line 25 of file [gpioPin\\_masks.h](#).

5.33.1.15 `#define IOPin_21 0x00200000`

Definition at line 26 of file [gpioPin\\_masks.h](#).

5.33.1.16 `#define IOPin_22 0x00400000`

Definition at line 27 of file [gpioPin\\_masks.h](#).

5.33.1.17 `#define IOPin_23 0x00800000`

Definition at line 28 of file [gpioPin\\_masks.h](#).

5.33.1.18 `#define IOPin_24 0x01000000`

Definition at line 29 of file [gpioPin\\_masks.h](#).

5.33.1.19 `#define IOPin_25 0x02000000`

Definition at line 30 of file [gpioPin\\_masks.h](#).

5.33.1.20 `#define IOPin_26 0x04000000`

Definition at line 31 of file [gpioPin\\_masks.h](#).

5.33.1.21 `#define IOPin_27 0x08000000`

Definition at line 32 of file [gpioPin\\_masks.h](#).

5.33.1.22 `#define IOPin_28 0x10000000`

Definition at line 33 of file [gpioPin\\_masks.h](#).

5.33.1.23 `#define IOPin_29 0x20000000`

Definition at line 34 of file [gpioPin\\_masks.h](#).

5.33.1.24 `#define IOPin_3 0x00000008`

Definition at line 8 of file [gpioPin\\_masks.h](#).

5.33.1.25 `#define IOPin_30 0x40000000`

Definition at line 35 of file [gpioPin\\_masks.h](#).

5.33.1.26 `#define IOPin_31 0x08000000`

Definition at line 36 of file [gpioPin\\_masks.h](#).

5.33.1.27 `#define IOPin_4 0x00000010`

Definition at line 9 of file [gpioPin\\_masks.h](#).

5.33.1.28 `#define IOPin_5 0x00000020`

Definition at line 10 of file [gpioPin\\_masks.h](#).

5.33.1.29 `#define IOPin_6 0x00000040`

Definition at line 11 of file [gpioPin\\_masks.h](#).

5.33.1.30 `#define IOPin_7 0x00000080`

Definition at line 12 of file [gpioPin\\_masks.h](#).

5.33.1.31 `#define IOPin_8 0x00000100`

Definition at line 13 of file [gpioPin\\_masks.h](#).

### 5.33.1.32 `#define IOPin_9 0x00000200`

Definition at line 14 of file [gpioPin\\_masks.h](#).

## 5.34 `gpioPin_masks.h`

```

00001 #ifndef GPIOPIN_MASKS
00002 #define GPIOPIN_MASKS
00003
00004
00005 #define IOPin_0      0x00000001
00006 #define IOPin_1      0x00000002
00007 #define IOPin_2      0x00000004
00008 #define IOPin_3      0x00000008
00009 #define IOPin_4      0x00000010
00010 #define IOPin_5      0x00000020
00011 #define IOPin_6      0x00000040
00012 #define IOPin_7      0x00000080
00013 #define IOPin_8      0x00000100
00014 #define IOPin_9      0x00000200
00015 #define IOPin_10     0x00000400
00016 #define IOPin_11     0x00000800
00017 #define IOPin_12     0x00001000
00018 #define IOPin_13     0x00002000
00019 #define IOPin_14     0x00004000
00020 #define IOPin_15     0x00008000
00021 #define IOPin_16     0x00010000
00022 #define IOPin_17     0x00020000
00023 #define IOPin_18     0x00040000
00024 #define IOPin_19     0x00080000
00025 #define IOPin_20     0x00100000
00026 #define IOPin_21     0x00200000
00027 #define IOPin_22     0x00400000
00028 #define IOPin_23     0x00800000
00029 #define IOPin_24     0x01000000
00030 #define IOPin_25     0x02000000
00031 #define IOPin_26     0x04000000
00032 #define IOPin_27     0x08000000
00033 #define IOPin_28     0x10000000
00034 #define IOPin_29     0x20000000
00035 #define IOPin_30     0x40000000
00036 #define IOPin_31     0x80000000
00037
00038
00039
00040 #endif //gpiopin_masks

```

## 5.35 `my_lib/ir.c` File Reference

```
#include "ir.h"
```

### Functions

- void [IRInit](#) ([IRInstance](#) \*instPtr)
- void [IRSend](#) ([IRInstance](#) \*instPtr, uint16\_t address, uint16\_t byte)
- void \_\_inline [IRByteBySoftware](#) ([IRInstance](#) \*instPtr, uint16\_t address, uint16\_t byte)
- void \_\_inline [IRRepeat](#) (uint32\_t port, uint32\_t pin, uint8\_t pulses, uint16\_t delay)

### 5.35.1 Function Documentation

5.35.1.1 void \_\_inline [IRByteBySoftware](#) ( [IRInstance](#) \* *instPtr*, uint16\_t *address*, uint16\_t *byte* )

Definition at line 146 of file [ir.c](#).

## 5.35.1.2 void IRInit ( IRInstance \* instPtr )

Definition at line 16 of file [ir.c](#).

## 5.35.1.3 void \_\_inline IRRepeat ( uint32\_t port, uint32\_t pin, uint8\_t pulses, uint16\_t delay )

Definition at line 212 of file [ir.c](#).

## 5.35.1.4 void IRSend ( IRInstance \* instPtr, uint16\_t address, uint16\_t byte )

Definition at line 57 of file [ir.c](#).

## 5.36 ir.c

```

00001 #include "ir.h"
00002
00003 #include "ir.h"
00004
00005
00006
00007
00008
00009
00010 /*
00011  * Calls IR init
00012  * Modes: IR_BY_SOFTWARE
00013  *         IR_BY_UART
00014  *         IR_BY_TIMER
00015  */
00016 void IRInit(IRInstance *instPtr)
00017 {
00018
00019     if((instPtr->Mode&IR_BY_SOFTWARE)!=0)
00020     {
00021         instPtr->CarrierPeriod = (uint16_t) 1000/instPtr->
CarrierFrequency;
00022         #ifdef IR_BY_SOFTWARE_EN
00023         /*
00024          * for software modulation, configure delay timing
00025          */
00026         if((instPtr->Mode & (IR_NEC_PROTOCOL|IR_NEC_EXTENDED))!=0)
00027             instPtr->Pulses = (uint16_t) (NEC_PULSE_TIME*((uint16_t) instPtr->
CarrierFrequency))/2000;
00028         if((instPtr->Mode&IR_RC5_PROTOCOL)!=0)
00029             instPtr->Pulses = (uint16_t) (RC5_PULSE_TIME*((uint16_t) instPtr->
CarrierFrequency))/1000;
00030         #endif
00031     }
00032
00033
00034     if(instPtr->Mode == IR_BY_UART)
00035     {
00036         #ifdef IR_BY_UART_EN
00037
00038         #endif
00039     }
00040
00041
00042     if(instPtr->Mode == IR_BY_TIMER)
00043     {
00044         #ifdef IR_BY_TIMER_EN
00045
00046         #endif
00047     }
00048
00049
00050 }
00051
00052
00053
00054 /*
00055  * Sends IR data according to instance
00056  */
00057 void IRSend(IRInstance *instPtr, uint16_t address, uint16_t byte)
00058 {
00059     uint32_t data;

```

```

00060     uint8_t tempAddress=0;
00061     uint8_t tempByte=0;
00062
00063 #ifdef IR_BY_SOFTWARE_EN
00064     uint16_t pulses;
00065     uint8_t roller;
00066     uint16_t delay = instPtr->CarrierPeriod/2;
00067 #endif
00068
00069     if ((instPtr->Mode&IR_NEC_PROTOCOL) !=0)                //inversdo enderee
    dados
    {
00070         tempAddress = ~address;
00071         address = ((address&0xFF) | ((tempAddress&0xFF)<<8);
00072         tempByte = ~byte;
00073         byte = ((byte&0xFF) | ((tempByte&0xFF)<<8);
00074     }
00075     if ((instPtr->Mode&(IR_NEC_PROTOCOL|IR_NEC_EXTENDED)) !=0)
00076         data = address|byte<<16;
00077
00078 #ifdef IR_BY_SOFTWARE_EN
00079     if ((instPtr->Mode&IR_BY_SOFTWARE) !=0)
00080     {
00081         if ((instPtr->Mode&(IR_NEC_EXTENDED|IR_NEC_PROTOCOL)) !=0)    //
    padrde envio
    {
00082         {
00083             pulses = instPtr->Pulses*32;
00084             roller = 32;
00085             while(pulses>0)                //start signal send 9ms
00086             {
00087                 IRPinSet(instPtr->TxPort, instPtr->TxPin);
00088                 IRDelayUs(delay);
00089                 IRPinClear(instPtr->TxPort, instPtr->TxPin);
00090                 IRDelayUs(delay);
00091                 pulses--;
00092             }
00093             IRDelayMs(4);                //protocol wait time
00094             IRDelayUs(500);
00095             while(roller>0)
00096             {
00097                 pulses = instPtr->Pulses;
00098                 while(pulses>0)                //carrier send
00099                 {
00100                     IRPinSet(instPtr->TxPort, instPtr->TxPin);
00101                     IRDelayUs(delay);
00102                     IRPinClear(instPtr->TxPort, instPtr->TxPin);
00103                     IRDelayUs(delay);
00104                     pulses--;
00105                 }
00106                 if ((data&0x1) !=0)
00107                     IRDelayUs(NEC_PULSE_TIME*2);
00108                 IRDelayUs(NEC_PULSE_TIME);
00109                 data >>= 1;
00110                 roller --;
00111             }
00112             pulses = instPtr->Pulses;
00113             while(pulses>0)                //end signal send 562.5 us
00114             {
00115                 IRPinSet(instPtr->TxPort, instPtr->TxPin);
00116                 IRDelayUs(delay);
00117                 IRPinClear(instPtr->TxPort, instPtr->TxPin);
00118                 IRDelayUs(delay);
00119                 pulses--;
00120             }
00121         }
00122     }
00123     if ((instPtr->Mode&IR_RC5_PROTOCOL) !=0)
00124     {
00125         //todo: to be implemented. sem sacco anymore.
00126     }
00127 }
00128 #endif
00129 #ifdef IR_BY_UART_EN
00130     if ((instPtr->Mode&IR_BY_UART) !=0)
00131     {
00132         //todo: make uart send buffer/command
00133     }
00134 }
00135 #endif
00136 #ifdef IR_BY_TIMER_EN
00137
00138 #endif
00139 }
00140
00141
00142 #ifdef IR_BY_SOFTWARE
00143 /*
00144 * sends a modulated bit

```

```

00145 */
00146 void __inline IRByteBySoftware(IRInstance *instPtr, uint16_t address, uint16_t
byte)
00147 {
00148     uint8_t tempAddress=0;
00149     uint8_t tempByte=0;
00150     uint16_t pulses;
00151     uint32_t data;
00152     uint8_t roller;
00153     uint16_t delay = instPtr->CarrierPeriod/2;
00154
00155     if((instPtr->Mode&IR_NEC_PROTOCOL)!=0) //inversdo enderee
dados
00156     {
00157         tempAddress = ~address;
00158         address = ((address&0xFF)|((tempAddress&0xFF)<<8);
00159         tempByte = ~byte;
00160         byte = ((byte&0xFF)|((tempByte&0xFF)<<8);
00161     }
00162     if((instPtr->Mode&(IR_NEC_EXTENDED|IR_NEC_PROTOCOL))!=0) //
padrde envio
00163     {
00164         data = address|byte<<16;
00165         pulses = instPtr->Pulses*32;
00166         roller = 32;
00167         while(pulses>0) //start signal send 9ms
00168         {
00169             IRPinSet(instPtr->TxPort, instPtr->TxPin);
00170             IRDelayUs(delay);
00171             IRPinClear(instPtr->TxPort, instPtr->TxPin);
00172             IRDelayUs(delay);
00173             pulses--;
00174         }
00175         IRDelayMs(4); //protocol wait time
00176         IRDelayUs(500);
00177         while(roller>0)
00178         {
00179             pulses = instPtr->Pulses;
00180             while(pulses>0) //carrier send
00181             {
00182                 IRPinSet(instPtr->TxPort, instPtr->TxPin);
00183                 IRDelayUs(delay);
00184                 IRPinClear(instPtr->TxPort, instPtr->TxPin);
00185                 IRDelayUs(delay);
00186                 pulses--;
00187             }
00188             if((data&0x1)!=0)
00189                 IRDelayUs(NEC_PULSE_TIME*2);
00190             IRDelayUs(NEC_PULSE_TIME);
00191             data >>= 1;
00192             roller --;
00193         }
00194         pulses = instPtr->Pulses;
00195         while(pulses>0) //end signal send 562.5 us
00196         {
00197             IRPinSet(instPtr->TxPort, instPtr->TxPin);
00198             IRDelayUs(delay);
00199             IRPinClear(instPtr->TxPort, instPtr->TxPin);
00200             IRDelayUs(delay);
00201             pulses--;
00202         }
00203     }
00204     if((instPtr->Mode&IR_RC5_PROTOCOL)!=0)
00205     {
00206         //todo: to be implemented. sem saco anymore.
00207     }
00208 }
00209
00210
00211
00212 void __inline IRRepeat(uint32_t port, uint32_t pin, uint8_t pulses, uint16_t delay)
00213 {
00214     uint8_t tempPulses;
00215     //fixme: repeat codes should be sent at 108ms intervals
00216     tempPulses = pulses;
00217     pulses *= 16;
00218     delay /= 2;
00219     while(pulses>0) //start signal send 9ms
00220     {
00221         IRPinSet(port, pin);
00222         IRDelayUs(delay);
00223         IRPinClear(port, pin);
00224         IRDelayUs(delay);
00225         pulses--;
00226     }
00227     IRDelayMs(2);
00228     IRDelayUs(250);

```

```

00229     pulses = tempPulses;
00230     while(pulses>0)                //end signal send 562.5 us
00231     {
00232         IRPinSet(port, pin);
00233         IRDelayUs(delay);
00234         IRPinClear(port, pin);
00235         IRDelayUs(delay);
00236         pulses--;
00237     }
00238 }
00239
00240
00241
00242
00243 #endif //ir_by_software
00244
00245
00246
00247
00248
00249
00250
00251
00252
00253
00254
00255

```

## 5.37 my\_lib/ir.h File Reference

```

#include "depl_spc/includeAll_sw.h"
#include "depl_spc/includeAll_hw.h"
#include "stdint.h"
#include "stdbool.h"

```

### Data Structures

- struct [IRInstance](#)

### Macros

- #define [IR\\_MAX\\_INSTANCES](#) 4
- #define [IR\\_BY\\_SOFTWARE](#) 0x0001
- #define [IR\\_BY\\_UART](#) 0x0002
- #define [IR\\_BY\\_TIMER](#) 0x0004
- #define [IR\\_BY\\_EXTERNAL\\_TIEMR](#) 0x0008
- #define [IR\\_NEC\\_PROTOCOL](#) 0x0010
- #define [IR\\_NEC\\_EXTENDED](#) 0x0020
- #define [IR\\_MY\\_PROTOCOL](#) 0x0040
- #define [IR\\_RC5\\_PROTOCOL](#) 0x0080
- #define [IR\\_REPEAT\\_COMMAND\\_ENABLE](#) 0x0100
- #define [IR\\_REPEAT\\_COMMAND\\_DISABLE](#) 0x0000
- #define [NEC\\_PULSE\\_TIME](#) 562
- #define [RC5\\_PULSE\\_TIME](#) 889
- #define [IRPinSet](#)(port, pin) GPIOPinWrite(port, pin, pin)
- #define [IRPinClear](#)(port, pin) GPIOPinWrite(port, pin, 0)
- #define [IRDelayMs](#)(delay) [SysDelayMs](#)(delay)
- #define [IRDelayUs](#)(delay) [SysDelayUs](#)(delay)
- #define [IRDelay](#)(delay) [SysDelay](#)(delay)



## Functions

- void [IRInit](#) ([IRInstance](#) \*instPtr)
- void [IRSend](#) ([IRInstance](#) \*instPtr, uint16\_t address, uint16\_t byte)
- void [IRByteBySoftware](#) ([IRInstance](#) \*instPtr, uint16\_t address, uint16\_t byte)
- void [IRRepeat](#) (uint32\_t port, uint32\_t pin, uint8\_t pulses, uint16\_t delay)

### 5.37.1 Macro Definition Documentation

#### 5.37.1.1 #define IR\_BY\_EXTERNAL\_TIEMR 0x0008

Definition at line 67 of file [ir.h](#).

#### 5.37.1.2 #define IR\_BY\_SOFTWARE 0x0001

Definition at line 64 of file [ir.h](#).

#### 5.37.1.3 #define IR\_BY\_TIMER 0x0004

Definition at line 66 of file [ir.h](#).

#### 5.37.1.4 #define IR\_BY\_UART 0x0002

Definition at line 65 of file [ir.h](#).

#### 5.37.1.5 #define IR\_MAX\_INSTANCES 4

Definition at line 59 of file [ir.h](#).

#### 5.37.1.6 #define IR\_MY\_PROTOCOL 0x0040

Definition at line 71 of file [ir.h](#).

#### 5.37.1.7 #define IR\_NEC\_EXTENDED 0x0020

Definition at line 70 of file [ir.h](#).

#### 5.37.1.8 #define IR\_NEC\_PROTOCOL 0x0010

Definition at line 69 of file [ir.h](#).

#### 5.37.1.9 #define IR\_RC5\_PROTOCOL 0x0080

Definition at line 72 of file [ir.h](#).

#### 5.37.1.10 #define IR\_REPEAT\_COMMAND\_DISABLE 0x0000

Definition at line 75 of file [ir.h](#).

5.37.1.11 `#define IR_REPEAT_COMMAND_ENABLE 0x0100`

Definition at line 74 of file [ir.h](#).

5.37.1.12 `#define IRDelay( delay ) SysDelay(delay)`

Definition at line 106 of file [ir.h](#).

5.37.1.13 `#define IRDelayMs( delay ) SysDelayMs(delay)`

Definition at line 104 of file [ir.h](#).

5.37.1.14 `#define IRDelayUs( delay ) SysDelayUs(delay)`

Definition at line 105 of file [ir.h](#).

5.37.1.15 `#define IRPinClear( port, pin ) GPIOPinWrite(port, pin, 0)`

Definition at line 103 of file [ir.h](#).

5.37.1.16 `#define IRPinSet( port, pin ) GPIOPinWrite(port, pin, pin)`

Definition at line 102 of file [ir.h](#).

5.37.1.17 `#define NEC_PULSE_TIME 562`

Definition at line 81 of file [ir.h](#).

5.37.1.18 `#define RC5_PULSE_TIME 889`

Definition at line 83 of file [ir.h](#).

## 5.37.2 Function Documentation

5.37.2.1 `void IRByteBySoftware ( IRInstance * instPtr, uint16_t address, uint16_t byte )`

Definition at line 146 of file [ir.c](#).

5.37.2.2 `void IRInit ( IRInstance * instPtr )`

Definition at line 16 of file [ir.c](#).

5.37.2.3 `void IRRepeat ( uint32_t port, uint32_t pin, uint8_t pulses, uint16_t delay )`

Definition at line 212 of file [ir.c](#).

5.37.2.4 `void IRSend ( IRInstance * instPtr, uint16_t address, uint16_t byte )`

Definition at line 57 of file [ir.c](#).

## 5.38 ir.h

```

00001 #ifndef ir_h
00002 #define ir_h
00003
00004
00005
00006 //main file header
00007 #include "depl_spc/includeAll_sw.h"
00008 #include "depl_spc/includeAll_hw.h"
00009
00010 #include "stdint.h"
00011 #include "stdbool.h"
00012
00013
00014 /*
00015  * need to declare:
00016  *
00017
00018 #define IRTX_FREQ          38000
00019 #define IRTX_PORT         GPIO_PORTB_BASE
00020 #define IRTX_PIN          GPIO_PIN_1
00021 #define IRRX_PORT         GPIO_PORTB_BASE
00022 #define IRRX_PIN          GPIO_PIN_0
00023 //larger compiled file
00024 #define IR_BY_SOFTWARE_EN
00025 #define IR_BY_UART_EN
00026 #define IR_UART_BASE      UART1_BASE
00027 #define IR_BY_TIMER_EN
00028 #define IR_TIMER_BASE     TIMER3_BASE
00029
00030
00031
00032 */
00033
00034
00035
00036 /*
00037  * InfraRed transceiver library
00038  * todo:    uart
00039  *          timer
00040  *          software
00041  *
00042  * -> uart peripheral support
00043  *      initiated for stellaris/tivaC uCs, uart IR coding support
00044  * -> timer peripheral support
00045  *      common to all uCs
00046  *      carrier frequency generation
00047  * -> software support
00048  *      support for full software control and emulation
00049  *      delay by cpu time use
00050  *
00051  *
00052  * devBy: rnm (17/11/13)
00053  */
00054
00055
00056 /*
00057  * Op. Param.
00058  */
00059 #define IR_MAX_INSTANCES          4
00060
00061 /*
00062  * Op. Masks
00063  */
00064 #define IR_BY_SOFTWARE            0x0001
00065 #define IR_BY_UART                0x0002
00066 #define IR_BY_TIMER              0x0004
00067 #define IR_BY_EXTERNAL_TIEMR     0x0008
00068
00069 #define IR_NEC_PROTOCOL           0x0010
00070 #define IR_NEC_EXTENDED          0x0020
00071 #define IR_MY_PROTOCOL           0x0040
00072 #define IR_RC5_PROTOCOL          0x0080
00073
00074 #define IR_REPEAT_COMMAND_ENABLE 0x0100
00075 #define IR_REPEAT_COMMAND_DISABLE 0x0000
00076
00077 /*
00078  * Protocol Definitions
00079  */
00080
00081 #define NEC_PULSE_TIME            562
00082
00083 #define RC5_PULSE_TIME            889
00084

```

```

00085
00086
00087 typedef struct{
00088     uint16_t Mode;                //IR_BY_XX | IR_XX_PROTOCOL
00089     uint8_t CarrierFrequency;    //in kHz
00090     uint16_t CarrierPeriod;      // in uS
00091     uint32_t TxPin;
00092     uint32_t TxPort;
00093     uint32_t RxPin;
00094     uint32_t RxPort;
00095     uint16_t ReceiveAddress;
00096     uint16_t ReceiveBuffer;
00097     uint16_t Pulses;
00098     uint8_t LastData;
00099 }IRInstance;
00100
00101
00102 #define IRPinSet(port, pin)      GPIOWrite(port, pin, pin)
00103 #define IRPinClear(port, pin)   GPIOWrite(port, pin, 0)
00104 #define IRDelayMs(delay)        SysDelayMs(delay)
00105 #define IRDelayUs(delay)        SysDelayUs(delay)
00106 #define IRDelay(delay)          SysDelay(delay)
00107
00108
00109
00110 void IRInit(IRInstance *instPtr);
00111 void IREnd(IRInstance *instPtr, uint16_t address, uint16_t byte);
00112 void IRByteBySoftware(IRInstance *instPtr, uint16_t address, uint16_t byte);
00113 void IRRepeat(uint32_t port, uint32_t pin, uint8_t pulses, uint16_t delay);
00114
00115
00116
00117 #endif// if_h

```

## 5.39 my\_lib/lcd.c File Reference

```
#include "lcd.h"
```

### Macros

- #define [true](#) 1
- #define [false](#) 0
- #define [trueDefinedLCD](#)
- #define [lcd\\_vector\\_index](#) 9

### Functions

- void [LCDInit](#) (void)  
*Initializes the LCD Module.*
- void [LCDSendCmd](#) (uint8\_t cmd)
- void [LCDSendChar](#) (uint8\_t txt)  
*send single character to LCD.*
- \_\_inline void [LCDSend](#) (uint8\_t send)
- void [LCDPosition](#) (uint8\_t row, uint8\_t col)
- void [LCDPositionNoDelay](#) (uint8\_t row, uint8\_t col)
- void [LCDSendString](#) (uint8\_t \*string, uint8\_t breakLine)
- void [LCDSendNumStrict](#) (int64\_t num, uint8\_t length, uint8\_t isSigned, uint8\_t showZeros)
- void [LCDSendNum](#) (int64\_t num, uint8\_t length, uint8\_t isSigned, uint8\_t showZeros)
- void [LCDSendNumArray](#) (uint8\_t \*index)
- void [LCDClear](#) (void)
- void [LCDDisplayOn](#) (uint8\_t onOff)
- void [LCDSendHex](#) (uint8\_t \*array)
- void [numToArray](#) (int32\_t num, uint8\_t \*array, uint8\_t length, uint16\_t base)

- void [LCDRegisterSpecial](#) (uint8\_t number, uint8\_t \*character)
- void [LCDShift](#) (uint8\_t shift)
- void [LCDHome](#) (void)
- void [arrayToNum](#) (uint8\_t \*array, uint32\_t \*num, uint8\_t base)
- void [LCDSendVU](#) (uint32\_t num, uint32\_t base)

## Variables

- const char [LCD\\_CmdInit\\_Vector](#) [[lcd\\_vector\\_index](#)]
- const unsigned int [LCD\\_InitDelay\\_Vector](#) [[lcd\\_vector\\_index](#)]

## 5.39.1 Macro Definition Documentation

### 5.39.1.1 #define false 0

Definition at line 12 of file [lcd.c](#).

### 5.39.1.2 #define lcd\_vector\_index 9

Definition at line 21 of file [lcd.c](#).

### 5.39.1.3 #define true 1

Checks and defines boolean values.

Definition at line 11 of file [lcd.c](#).

### 5.39.1.4 #define trueDefinedLCD

Definition at line 13 of file [lcd.c](#).

## 5.39.2 Function Documentation

### 5.39.2.1 void arrayToNum ( uint8\_t \* array, uint32\_t \* num, uint8\_t base )

Definition at line 422 of file [lcd.c](#).

### 5.39.2.2 void LCDClear ( void )

Definition at line 286 of file [lcd.c](#).

### 5.39.2.3 void LCDDisplayOn ( uint8\_t onOff )

Definition at line 300 of file [lcd.c](#).

### 5.39.2.4 void LCDHome ( void )

Definition at line 414 of file [lcd.c](#).

#### 5.39.2.5 void LCDInit ( void )

Initializes the LCD Module.

Called once at startup. Takes no parameters.

##### Returns

None.

Definition at line 47 of file [lcd.c](#).

#### 5.39.2.6 void LCDPosition ( uint8\_t row, uint8\_t col )

Definition at line 117 of file [lcd.c](#).

#### 5.39.2.7 void LCDPositionNoDelay ( uint8\_t row, uint8\_t col )

Definition at line 135 of file [lcd.c](#).

#### 5.39.2.8 void LCDRegisterSpecial ( uint8\_t number, uint8\_t \* character )

Definition at line 387 of file [lcd.c](#).

#### 5.39.2.9 \_\_inline void LCDSend ( uint8\_t send )

Definition at line 102 of file [lcd.c](#).

#### 5.39.2.10 void LCDSendChar ( uint8\_t txt )

send single character to LCD.

##### Parameters

<i>txt</i>	char type data to be sent, 8 bits.
------------	------------------------------------

Definition at line 91 of file [lcd.c](#).

#### 5.39.2.11 void LCDSendCmd ( uint8\_t cmd )

Definition at line 77 of file [lcd.c](#).

#### 5.39.2.12 void LCDSendHex ( uint8\_t \* array )

Definition at line 314 of file [lcd.c](#).

#### 5.39.2.13 void LCDSendNum ( int64\_t num, uint8\_t length, uint8\_t isSigned, uint8\_t showZeros )

Definition at line 235 of file [lcd.c](#).

#### 5.39.2.14 void LCDSendNumArray ( uint8\_t \* index )

Definition at line 273 of file [lcd.c](#).

5.39.2.15 void LCDSendNumStrict ( int64\_t *num*, uint8\_t *length*, uint8\_t *isSigned*, uint8\_t *showZeros* )

Definition at line 188 of file [lcd.c](#).

5.39.2.16 void LCDSendString ( uint8\_t \* *string*, uint8\_t *breakLine* )

Definition at line 153 of file [lcd.c](#).

5.39.2.17 void LCDSendVU ( uint32\_t *num*, uint32\_t *base* )

Definition at line 434 of file [lcd.c](#).

5.39.2.18 void LCDShift ( uint8\_t *shift* )

Definition at line 405 of file [lcd.c](#).

5.39.2.19 void numToArray ( int32\_t *num*, uint8\_t \* *array*, uint8\_t *length*, uint16\_t *base* )

Definition at line 347 of file [lcd.c](#).

### 5.39.3 Variable Documentation

5.39.3.1 const char LCD\_CmdInit\_Vector[lcd\_vector\_index]

**Initial value:**

```
=
{
    0x03, 0x38, 0x38, 0x38, 0x01, LCD_DISPLAY_CONFIG,
    LCD_DISPLAY_INCREMENT, 0x01, 0x02,
}
```

Definition at line 22 of file [lcd.c](#).

5.39.3.2 const unsigned int LCD\_InitDelay\_Vector[lcd\_vector\_index]

**Initial value:**

```
=
{
    8000, 800, 800, 800, 600, 600, 200, 200, 200
}
```

LCD Init command delay vector, in uS

Definition at line 32 of file [lcd.c](#).

## 5.40 lcd.c

```
00001
00002 #include "lcd.h"
00003
00004
00005
00006
00007 /**
00008  * Checks and defines boolean values.
```

```

00009  */
00010 #ifndef true
00011 #define true 1
00012 #define false 0
00013 #define trueDefinedLCD
00014 #endif
00015
00016
00017 /*
00018  * Initialization Sequence:
00019  * TODO: create masks for LCD commands
00020  */
00021 #define lcd_vector_index 9
00022 const char LCD_CmdInit_Vector [lcd_vector_index] = \
00023 {
00024     0x03, 0x38, 0x38, 0x38, 0x01, LCD_DISPLAY_CONFIG,
00025     LCD_DISPLAY_INCREMENT, 0x01, 0x02,
00026 };
00027 /*
00028  * Delay time in uSs
00029  */
00030 /**
00031  * LCD Init command delay vector, in uS
00032  */
00033 const unsigned int LCD_InitDelay_Vector[lcd_vector_index] = \
00034 {
00035     8000, 800, 800, 800, 600, 600, 200, 200, 200
00036 };
00037
00038
00039
00040 /**
00041  * \brief Initializes the LCD Module
00042  *
00043  * Called once at startup. Takes no parameters.
00044  *
00045  * \return None.
00046  */
00047 void LCDInit(void)
00048 {
00049     uint8_t Vector_Scan = 0;
00050     LCDDelay(15000);
00051     for(Vector_Scan=0; Vector_Scan < lcd_vector_index; Vector_Scan++)
00052     {
00053         LCDSendCmd(LCD_CmdInit_Vector[Vector_Scan]);
00054         LCDDelay(LCD_InitDelay_Vector[Vector_Scan]);
00055     }
00056     //splash screen
00057 #if LCD_SPLASHSCREEN1 == 1
00058     LCDPosition(1,1);
00059     LCDSendString(LCD_splashscreen_row1, false);
00060     LCDPosition(2,1);
00061     LCDSendString(LCD_splashscreen_row2, false);
00062     LCDDelay(2*1000*1000);
00063 #endif
00064 #if LCD_SPLASHSCREEN2 == 1
00065     LCDPosition(1,1);
00066     LCDSendString(LCD_splashscreen2_row1, false);
00067     LCDPosition(2,1);
00068     LCDSendString(LCD_splashscreen2_row2, false);
00069     LCDDelay(2*1000*1000);
00070 #endif
00071     LCDClear();
00072 }
00073
00074 /*
00075  * Send a Command to the LCD
00076  */
00077 void LCDSendCmd(uint8_t cmd)
00078 {
00079     LCD_RS_Low;
00080     LCDSend(cmd);
00081 }
00082
00083 /*
00084  * Send a Text to the LCD
00085  */
00086 /**
00087  * \brief send single character to LCD.
00088  *
00089  * \param txt char type data to be sent, 8 bits.
00090  */
00091 void LCDSendChar(uint8_t txt)
00092 {
00093     LCD_RS_High;
00094     LCDSend(txt);

```



```

00095     LCD0Status.col ++;
00096     LCD_RS_Low;
00097 }
00098
00099 /*
00100  * Write on serial shifter, pulse LCD EN
00101  */
00102 __inline void LCDSend(uint8_t send)
00103 {
00104     LCD_EN_Low;
00105     LCD_DTA_Send(send);
00106     LCDDelay(4);
00107     LCD_EN_High;
00108     LCDDelay(4);
00109     LCD_EN_Low;
00110 }
00111
00112 /*
00113  * Changes the LCD Cursor Position
00114  * Updates LCDStatus
00115  * PARAM: row, col
00116  */
00117 void LCDPosition(uint8_t row, uint8_t col)
00118 {
00119     LCD0Status.row = row;
00120     LCD0Status.col = col;
00121     col--;
00122     if(row==1)
00123         row = 0x80;
00124     if(row==2)
00125         row = 0xC0;
00126     LCDSendCmd(row+col);
00127     LCDDelay(20);
00128 }
00129
00130 /*
00131  * Changes the LCD Cursor Position
00132  * Updates LCDStatus
00133  * PARAM: row, col
00134  */
00135 void LCDPositionNoDelay(uint8_t row, uint8_t col)
00136 {
00137     LCD0Status.row = row;
00138     LCD0Status.col = col;
00139     col--;
00140     if(row==1)
00141         row = 0x80;
00142     if(row==2)
00143         row = 0xC0;
00144     LCDSendCmd(row+col);
00145 }
00146
00147
00148 /*
00149  * Writes a string of characteres on display
00150  * Processes according to the ASCII code
00151  * 0 - NULL
00152  */
00153 void LCDSendString(uint8_t *string, uint8_t breakLine)
00154 {
00155     while(*string)
00156     {
00157         LCDSendChar(*string);
00158         string++;
00159         if(LCD0Status.col==LCD_col_num && breakLine==true)
00160             if(LCD0Status.row<=LCD_row_num)
00161                 LCDPosition(LCD0Status.row+1, 1);
00162             else
00163                 LCDPosition(0, 1);
00164     }
00165 }
00166
00167
00168
00169
00170 /*
00171  * TODO: make function to print string and remaining spaces in LCD
00172  */
00173
00174
00175 /*
00176  * Prints a number, from a variable, to the LCD
00177  * PARAM: num, length, isSigned, showZeros
00178  * IF signed
00179  *     Limits to a max of 10 digits to a positive number
00180  *     Limits to a max of 9 digits to a negative number
00181  * ELSE

```

```

00182 *      limits to 10 chars
00183 * IF showzeros
00184 *      shows all leading zeros
00185 * else
00186 *      supresses zeros; places space instead
00187 */
00188 void LCDSendNumStrict(int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)
00189 {
00190     uint8_t index =0;
00191     uint8_t out;
00192     uint64_t multiple = 1;
00193     limitCeilValue(length,10);
00194     if(num<0 && isSigned==true)
00195     {
00196         LCDSendChar('-');
00197         num *= -1;
00198         length--;
00199     }
00200     index = length;
00201     while(length>1)
00202     {
00203         multiple *= 10;
00204         length--;
00205     }
00206     while(index >= 1)
00207     {
00208         out = (uint32_t) (num/multiple);
00209         num -= out*(multiple);
00210         if(out!=0)
00211             showZeros = true;
00212         if(out==0 && showZeros==false)
00213             out -= 16;
00214         LCDSendChar(out+48);
00215         multiple /= 10;
00216         index--;
00217     }
00218 }
00219
00220
00221
00222 /*
00223 * Prints a number, from a variable, to the LCD
00224 * PARAM: num, length, isSigned, showZeros
00225 * IF signed
00226 *     Limits to a max of 10 digits to a positive number
00227 *     Limits to a max of 9 digits to a negative number
00228 * ELSE
00229 *     limits to 10 chars
00230 * IF showzeros
00231 *     shows all leading zeros
00232 * else
00233 *     supresses zeros; places space instead
00234 */
00235 void LCDSendNum(int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)
00236 {
00237     uint8_t index =0;
00238     uint8_t out = ' ';
00239     uint64_t multiple = 1;
00240     limitCeilValue(length,10);
00241     if(num<0 && isSigned==true)
00242     {
00243         out = '-';
00244         num *= -1;
00245     }
00246     LCDSendChar(out);
00247     index = length;
00248     while(length>1)
00249     {
00250         multiple *= 10;
00251         length--;
00252     }
00253     while(index >= 1)
00254     {
00255         out = (uint32_t) (num/multiple);
00256         num -= out*(multiple);
00257         if(out!=0)
00258             showZeros = true;
00259         if(out==0 && showZeros==false)
00260             out -= 16;
00261         LCDSendChar(out+48);
00262         multiple /= 10;
00263         index--;
00264     }
00265 }
00266
00267
00268 /*

```

```

00269  * Sends numerical values to LCD
00270  * Values between 0 and base;
00271  * max base value is defined as 32 (32bit wide buses)
00272  */
00273 void LCDSendNumArray(uint8_t *index)
00274 {
00275     while(*index<33)
00276     {
00277         LCDSendChar(*index+'0');
00278         index++;
00279     }
00280 }
00281
00282 /*
00283  * Clears display
00284  * Updates LCDStatus
00285  */
00286 void LCDClear(void)
00287 {
00288     LCDSendCmd(0x01);
00289     LCD0Status.row=1;
00290     LCD0Status.col=1;
00291     LCDDelay(800);
00292 }
00293
00294 /*
00295  * Turns
00296  * LCD_DISPLAY_ON/OFF
00297  * LCD_CURSOR_ON/OFF
00298  * LCD_BLINK_ON/OFF
00299  */
00300 void LCDDisplayOn(uint8_t onOff)
00301 {
00302     LCD0Status.display = onOff;
00303     LCDSendCmd(onOff);
00304 }
00305
00306
00307
00308
00309 /*
00310  * Prints the value of the array in hex format
00311  * As HEX, it'll print in base 16
00312  * Takes out 2 leading digits
00313  */
00314 void LCDSendHex(uint8_t *array)
00315 {
00316     uint8_t offset, temp;
00317     LCDSendChar('0');
00318     LCDSendChar('x');
00319     array += 2;
00320     while(*array<=32)
00321     {
00322         temp = *array;
00323         if(temp>9)
00324         {
00325             temp -= 10;
00326             offset = 'A';
00327         }
00328         else
00329             offset = '0';
00330         LCDSendChar(temp+offset);
00331         array++;
00332     }
00333 }
00334
00335
00336
00337 //void LCDSendNum(long num, char length, uint8_t isSigned, uint8_t showZeros)
00338
00339 /*
00340  * Passes a number to a vector
00341  * num -> number
00342  * vector -> pointer to vector
00343  * base -> base of output (max: 32)
00344  *
00345  * Last number in vector output is 33
00346  */
00347 void numToArray(int32_t num, uint8_t *array, uint8_t length, uint16_t base)
00348 {
00349     uint16_t index =1;
00350     uint8_t out;
00351     uint64_t multiple = 1;
00352
00353     limitCeilValue(length, (unsigned char) 1<<(64/base));
00354     limitCeilValue(length, maxLengthOut);
00355

```

```

00356
00357 //create multiple number
00358 while(index<length)
00359 {
00360     multiple *= base;
00361     index++;
00362 }
00363 //sort multiples
00364 while(index >= 1)
00365 {
00366     //determines the multiple
00367     out = (uint8_t) (num/multiple);
00368     //takes out multiple
00369     num -= out*(multiple);
00370
00371     //escreve no vetor, desloca indice
00372     *array = out;
00373     array++;
00374     multiple /= base;
00375     //change multiple position
00376     index--;
00377 }
00378 *array = 33;
00379 }
00380
00381 /*
00382 * registers special characteres
00383 * number -> from 0 to 7
00384 * *character -> first index to 8 bytes long vector
00385 *             scans char downward
00386 */
00387 void LCDRegisterSpecial(uint8_t number, uint8_t *character)
00388 {
00389     uint8_t scan=0, data=0;
00390     LCDSendCmd(0x40+(number<<3));
00391     do
00392     {
00393         data = *(character+scan);
00394         LCDDelay(640);
00395         LCDSendChar(data&0x1F);
00396         scan++;
00397     }
00398     while(scan<8);
00399     LCDDelay(320);
00400 }
00401
00402 /*
00403 * Shifts data on LCD Display
00404 */
00405 void LCDShift(uint8_t shift)
00406 {
00407     LCDSendCmd(shift|LCD_SHIFT);
00408 }
00409
00410 /*
00411 * Sends LCD cursor to home position
00412 * PARAM: none
00413 */
00414 void LCDHome(void)
00415 {
00416     LCDSendCmd(0x02);
00417     LCDDelay(1500);
00418 }
00419
00420
00421
00422 void arrayToNum(uint8_t *array, uint32_t *num, uint8_t base)
00423 {
00424     while(*array<33)
00425     {
00426         *num += *array * base;
00427         array++;
00428     }
00429 }
00430
00431
00432
00433
00434 void LCDSendVU(uint32_t num, uint32_t base)
00435 {
00436     uint8_t index, pass=1;
00437     num = (unsigned int) num*(LCD_col_num*LCD_char_width)/base;
00438     while(num>0)
00439     {
00440         index = LCD_char_width;
00441         while(num<LCD_char_width)
00442         {

```

```

00443         index--;
00444         num++;
00445     }
00446     LCDSendChar(index);
00447     num -= LCD_char_width;
00448     pass++;
00449 }
00450 while(pass<=LCD_col_num)
00451 {
00452     pass++;
00453     LCDSendChar(0);
00454 }
00455 }
00456
00457
00458
00459
00460 #ifdef trueDefinedLCD
00461 #undef true
00462 #undef false
00463 #endif
00464
00465
00466

```

## 5.41 my\_lib/lcd.h File Reference

```

#include "depl_spc/includeAll_hw.h"
#include "depl_spc/includeAll_sw.h"

```

### Data Structures

- struct [LCDStatus](#)

### Macros

- #define [LCD\\_splashscreen\\_row1](#) PROJECT\_NAME  
*geneartion of project name in LCD*
- #define [LCD\\_splashscreen\\_row2](#) ("rnm sys undvpd")  
*creator's watermark*
- #define [LCD\\_splashscreen2\\_row1](#) \_\_DATE\_\_  
*compile date, used as program version*
- #define [LCD\\_splashscreen2\\_row2](#) \_\_TIME\_\_  
*compile time, used as program version*
- #define [LCDDelay\(x\)](#) SysDelayUs(x)
- #define [LCDPinSet\(x, y\)](#) PinAddrSet(x, y)
- #define [LCDPinClear\(x, y\)](#) PinAddrClear(x,y)
- #define [LCD\\_RS\\_High](#) LCDPinSet(LCD\_RS\_Port, LCD\_RS\_Pin)
- #define [LCD\\_RS\\_Low](#) LCDPinClear(LCD\_RS\_Port, LCD\_RS\_Pin)
- #define [LCD\\_EN\\_High](#) LCDPinSet(LCD\_EN\_Port, LCD\_EN\_Pin)
- #define [LCD\\_EN\\_Low](#) LCDPinClear(LCD\_EN\_Port, LCD\_EN\_Pin)
- #define [LCD\\_CLK\\_High](#) LCDPinSet(LCD\_CLK\_Port, LCD\_CLK\_Pin)
- #define [LCD\\_CLK\\_LoW](#) LCDPinClear(LCD\_CLK\_Port, LCD\_CLK\_Pin)
- #define [LCD\\_DTA\\_Send](#)(text)
- #define [LCD\\_DISPLAY\\_ON](#) 0x0C
- #define [LCD\\_DISPLAY\\_OFF](#) 0x08
- #define [LCD\\_CURSOR\\_ON](#) 0x0A
- #define [LCD\\_CURSOR\\_OFF](#) 0x08
- #define [LCD\\_BLINK\\_ON](#) 0x09

- `#define LCD_BLINK_OFF 0x08`
- `#define LCD_SHIFT 0x10`
- `#define LCD_SHIFT_DISPLAY 0x08`
- `#define LCD_SHIFT_CURSOR 0x02`
- `#define LCD_SHIFT_RIGHT 0x04`
- `#define LCD_SHIFT_LEFT 0x00`
- `#define LCD_SET_CGRAM 0x40`
- `#define LCD_INCREMENT 0x04`
- `#define LCD_INCREMENT_NO_SHIFT 0x00`
- `#define LCD_INCREMENT_SHIFT 0x01`
- `#define LCD_INCREMENT_POSITIVE 0x02`
- `#define LCD_INCREMENT_NEGATIVE 0x00`
- `#define LCD_DISPLAY_CONFIG (LCD_DISPLAY_ON|LCD_CURSOR_OFF|LCD_BLINK_OFF)`
- `#define LCD_DISPLAY_INCREMENT (LCD_INCREMENT|LCD_INCREMENT_NO_SHIFT)`
- `#define maxLengthOut 16`

## Functions

- `void LCDInit (void)`  
*Initializes the LCD Module.*
- `void LCDSendCmd (uint8_t cmd)`
- `void LCDSendChar (uint8_t txt)`  
*send single character to LCD.*
- `void LCDSend (uint8_t send)`
- `void LCDPosition (uint8_t row, uint8_t col)`
- `void LCDPositionNoDelay (uint8_t row, uint8_t col)`
- `void LCDSendString (uint8_t *string, uint8_t breakLine)`
- `void LCDSendNumStrict (int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)`
- `void LCDSendNum (int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)`
- `void LCDSendNumArray (uint8_t *vector)`
- `void LCDClear (void)`
- `void LCDSendHex (uint8_t *array)`
- `void numToArray (int32_t num, uint8_t *array, uint8_t length, uint16_t base)`
- `void LCDRegisterSpecial (uint8_t number, uint8_t *character)`
- `void LCDShift (uint8_t shift)`
- `void LCDHome (void)`
- `void arrayToNum (uint8_t *array, uint32_t *num, uint8_t base)`
- `void LCDSendVU (uint32_t num, uint32_t base)`

## Variables

- `LCDStatus LCD0Status`

### 5.41.1 Macro Definition Documentation

#### 5.41.1.1 `#define LCD_BLINK_OFF 0x08`

Definition at line 101 of file [lcd.h](#).

#### 5.41.1.2 `#define LCD_BLINK_ON 0x09`

Definition at line 100 of file [lcd.h](#).

5.41.1.3 `#define LCD_CLK_High LCDPinSet(LCD_CLK_Port, LCD_CLK_Pin)`

Definition at line 87 of file [lcd.h](#).

5.41.1.4 `#define LCD_CLK_LoW LCDPinClear(LCD_CLK_Port, LCD_CLK_Pin)`

Definition at line 88 of file [lcd.h](#).

5.41.1.5 `#define LCD_CURSOR_OFF 0x08`

Definition at line 99 of file [lcd.h](#).

5.41.1.6 `#define LCD_CURSOR_ON 0x0A`

Definition at line 98 of file [lcd.h](#).

5.41.1.7 `#define LCD_DISPLAY_CONFIG (LCD_DISPLAY_ON|LCD_CURSOR_OFF|LCD_BLINK_OFF)`

Definition at line 116 of file [lcd.h](#).

5.41.1.8 `#define LCD_DISPLAY_INCREMENT (LCD_INCREMENT|LCD_INCREMENT_NO_SHIFT)`

Definition at line 117 of file [lcd.h](#).

5.41.1.9 `#define LCD_DISPLAY_OFF 0x08`

Definition at line 97 of file [lcd.h](#).

5.41.1.10 `#define LCD_DISPLAY_ON 0x0C`

Definition at line 96 of file [lcd.h](#).

5.41.1.11 `#define LCD_DTA_Send( text )`

**Value:**

```
ShiftSerialSend(LCD_DTA_Port, \
                  LCD_DTA_Pin, \
                  LCD_CLK_Port, \
                  LCD_CLK_Pin, text)
```

Definition at line 89 of file [lcd.h](#).

5.41.1.12 `#define LCD_EN_High LCDPinSet(LCD_EN_Port, LCD_EN_Pin)`

Definition at line 85 of file [lcd.h](#).

5.41.1.13 `#define LCD_EN_Low LCDPinClear(LCD_EN_Port, LCD_EN_Pin)`

Definition at line 86 of file [lcd.h](#).

5.41.1.14 `#define LCD_INCREMENT 0X04`

Definition at line 108 of file [lcd.h](#).

5.41.1.15 `#define LCD_INCREMENT_NEGATIVE 0x00`

Definition at line 112 of file [lcd.h](#).

5.41.1.16 `#define LCD_INCREMENT_NO_SHIFT 0x00`

Definition at line 109 of file [lcd.h](#).

5.41.1.17 `#define LCD_INCREMENT_POSITIVE 0x02`

Definition at line 111 of file [lcd.h](#).

5.41.1.18 `#define LCD_INCREMENT_SHIFT 0x01`

Definition at line 110 of file [lcd.h](#).

5.41.1.19 `#define LCD_RS_High LCDPinSet(LCD_RS_Port, LCD_RS_Pin)`

Definition at line 83 of file [lcd.h](#).

5.41.1.20 `#define LCD_RS_Low LCDPinClear(LCD_RS_Port, LCD_RS_Pin)`

Definition at line 84 of file [lcd.h](#).

5.41.1.21 `#define LCD_SET_CGRAM 0x40`

Definition at line 107 of file [lcd.h](#).

5.41.1.22 `#define LCD_SHIFT 0x10`

Definition at line 102 of file [lcd.h](#).

5.41.1.23 `#define LCD_SHIFT_CURSOR 0x02`

Definition at line 104 of file [lcd.h](#).

5.41.1.24 `#define LCD_SHIFT_DISPLAY 0x08`

Definition at line 103 of file [lcd.h](#).

5.41.1.25 `#define LCD_SHIFT_LEFT 0x00`

Definition at line 106 of file [lcd.h](#).



5.41.1.26 `#define LCD_SHIFT_RIGHT 0x04`

Definition at line 105 of file [lcd.h](#).

5.41.1.27 `#define LCD_splashscreen2_row1 __DATE__`

compile date, used as program version

Definition at line 13 of file [lcd.h](#).

5.41.1.28 `#define LCD_splashscreen2_row2 __TIME__`

compile time, used as program version

Definition at line 14 of file [lcd.h](#).

5.41.1.29 `#define LCD_splashscreen_row1 PROJECT_NAME`

generation of project name in LCD

Definition at line 10 of file [lcd.h](#).

5.41.1.30 `#define LCD_splashscreen_row2 ("rnm sys undvpd")`

creator's watermark

Definition at line 11 of file [lcd.h](#).

5.41.1.31 `#define LCDDelay( x ) SysDelayUs(x)`

Definition at line 76 of file [lcd.h](#).

5.41.1.32 `#define LCDPinClear( x, y ) PinAddrClear(x,y)`

Definition at line 78 of file [lcd.h](#).

5.41.1.33 `#define LCDPinSet( x, y ) PinAddrSet(x, y)`

Definition at line 77 of file [lcd.h](#).

5.41.1.34 `#define maxLengthOut 16`

Definition at line 155 of file [lcd.h](#).

## 5.41.2 Function Documentation

5.41.2.1 `void arrayToNum ( uint8_t * array, uint32_t * num, uint8_t base )`

Definition at line 422 of file [lcd.c](#).

5.41.2.2 `void LCDClear ( void )`

Definition at line 286 of file [lcd.c](#).

#### 5.41.2.3 void LCDHome ( void )

Definition at line 414 of file [lcd.c](#).

#### 5.41.2.4 void LCDInit ( void )

Initializes the LCD Module.

Called once at startup. Takes no parameters.

##### Returns

None.

Definition at line 47 of file [lcd.c](#).

#### 5.41.2.5 void LCDPosition ( uint8\_t row, uint8\_t col )

Definition at line 117 of file [lcd.c](#).

#### 5.41.2.6 void LCDPositionNoDelay ( uint8\_t row, uint8\_t col )

Definition at line 135 of file [lcd.c](#).

#### 5.41.2.7 void LCDRegisterSpecial ( uint8\_t number, uint8\_t \* character )

Definition at line 387 of file [lcd.c](#).

#### 5.41.2.8 void LCDSend ( uint8\_t send )

Definition at line 102 of file [lcd.c](#).

#### 5.41.2.9 void LCDSendChar ( uint8\_t txt )

send single character to LCD.

##### Parameters

<i>txt</i>	char type data to be sent, 8 bits.
------------	------------------------------------

Definition at line 91 of file [lcd.c](#).

#### 5.41.2.10 void LCDSendCmd ( uint8\_t cmd )

Definition at line 77 of file [lcd.c](#).

#### 5.41.2.11 void LCDSendHex ( uint8\_t \* array )

Definition at line 314 of file [lcd.c](#).

#### 5.41.2.12 void LCDSendNum ( int64\_t num, uint8\_t length, uint8\_t isSigned, uint8\_t showZeros )

Definition at line 235 of file [lcd.c](#).

5.41.2.13 void LCDSendNumArray ( uint8\_t \* *vector* )

Definition at line 273 of file [lcd.c](#).

5.41.2.14 void LCDSendNumStrict ( int64\_t *num*, uint8\_t *length*, uint8\_t *isSigned*, uint8\_t *showZeros* )

Definition at line 188 of file [lcd.c](#).

5.41.2.15 void LCDSendString ( uint8\_t \* *string*, uint8\_t *breakLine* )

Definition at line 153 of file [lcd.c](#).

5.41.2.16 void LCDSendVU ( uint32\_t *num*, uint32\_t *base* )

Definition at line 434 of file [lcd.c](#).

5.41.2.17 void LCDShift ( uint8\_t *shift* )

Definition at line 405 of file [lcd.c](#).

5.41.2.18 void numToArray ( int32\_t *num*, uint8\_t \* *array*, uint8\_t *length*, uint16\_t *base* )

Definition at line 347 of file [lcd.c](#).

## 5.41.3 Variable Documentation

5.41.3.1 LCDStatus LCD0Status

## 5.42 lcd.h

```

00001 #ifndef lcd_h
00002 #define lcd_h
00003
00004
00005 #include "depl_spc/includeAll_hw.h"
00006 #include "depl_spc/includeAll_sw.h"
00007
00008
00009
00010 #define LCD_splashscreen_row1 PROJECT_NAME           //!< generation of project name in LCD
00011 #define LCD_splashscreen_row2 ("rnm sys undvdp")      //!< creator's watermark
00012
00013 #define LCD_splashscreen2_row1 __DATE__               //!< compile date, used as program version
00014 #define LCD_splashscreen2_row2 __TIME__              //!< compile time, used as program version
00015
00016
00017
00018 /*
00019 NEED TO DECLARE
00020
00021 //LCD
00022 #define LCD_RS J1_05 //E5
00023 #define LCD_RS_Port GPIO_PORTA_BASE
00024 #define LCD_RS_Pin GPIO_PIN_5
00025
00026 #define LCD_EN J1_06 //E4
00027 #define LCD_EN_Port GPIO_PORTA_BASE
00028 #define LCD_EN_Pin GPIO_PIN_4
00029
00030 #define LCD_DTA J2_09 //A2
00031 #define LCD_DTA_Port GPIO_PORTA_BASE
00032 #define LCD_DTA_Pin GPIO_PIN_3
00033
00034 #define LCD_CLK J2_10 //A3

```

```

00035 #define LCD_CLK_Port      GPIO_PORTA_BASE
00036 #define LCD_CLK_Pin        GPIO_PIN_2
00037
00038 #define LCD_row_num        2
00039 #define LCD_col_num        16
00040
00041 LCDStatus LCD0Status;
00042
00043
00044 #define LCD_splashscreen_row1  ("odqd")
00045 #define LCD_splashscreen_row2  ("rnm sys undvdp")
00046
00047
00048
00049 uint8_t specialChar[8][8] = { //ultima coluna, linha de baixo, reservada para cursor
00050     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x07, 0x00, 0x00,\
00051     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x06, 0x01, 0x00,\
00052     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x05, 0x02, 0x00,\
00053     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x04, 0x03, 0x00,\
00054     0x1F, 0x0F, 0x07, 0x03, 0x00, 0x03, 0x04, 0x00,\
00055     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x02, 0x05, 0x00,\
00056     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x01, 0x06, 0x00,\
00057     0x1F, 0x0F, 0x07, 0x03, 0x01, 0x00, 0x07, 0x00
00058 };
00059
00060
00061
00062 */
00063
00064
00065
00066 /*
00067  * LIBRARY FOR LCD USE
00068  * SERIAL COMM
00069  * USE OF SHIFT REGISTERS FOR DATA
00070  */
00071
00072
00073
00074 //sub-function masks
00075 //External Function Masks
00076 #define LCDDelay(x)                SysDelayUs(x)
00077 #define LCDPinSet(x, y)            PinAddrSet(x, y)
00078 #define LCDPinClear(x, y)          PinAddrClear(x,y)
00079
00080
00081
00082 //sub-function masks
00083 #define LCD_RS_High                LCDPinSet(LCD_RS_Port, LCD_RS_Pin)
00084 #define LCD_RS_Low                 LCDPinClear(LCD_RS_Port, LCD_RS_Pin)
00085 #define LCD_EN_High                LCDPinSet(LCD_EN_Port, LCD_EN_Pin)
00086 #define LCD_EN_Low                 LCDPinClear(LCD_EN_Port, LCD_EN_Pin)
00087 #define LCD_CLK_High               LCDPinSet(LCD_CLK_Port, LCD_CLK_Pin)
00088 #define LCD_CLK_LoW                LCDPinClear(LCD_CLK_Port, LCD_CLK_Pin)
00089 #define LCD_DTA_Send(text)         ShiftSerialSend(LCD_DTA_Port,\
00090     LCD_DTA_Pin,\
00091     LCD_CLK_Port,\
00092     LCD_CLK_Pin, text)
00093
00094
00095 //LCD Command Masks
00096 #define LCD_DISPLAY_ON              0x0C
00097 #define LCD_DISPLAY_OFF             0x08
00098 #define LCD_CURSOR_ON              0x0A
00099 #define LCD_CURSOR_OFF             0x08
00100 #define LCD_BLINK_ON               0x09
00101 #define LCD_BLINK_OFF              0x08
00102 #define LCD_SHIFT                  0x10
00103 #define LCD_SHIFT_DISPLAY           0x08
00104 #define LCD_SHIFT_CURSOR           0x02
00105 #define LCD_SHIFT_RIGHT             0x04
00106 #define LCD_SHIFT_LEFT             0x00
00107 #define LCD_SET_CGRAM               0x40
00108 #define LCD_INCREMENT              0x04
00109 #define LCD_INCREMENT_NO_SHIFT      0x00
00110 #define LCD_INCREMENT_SHIFT        0x01
00111 #define LCD_INCREMENT_POSITIVE      0x02
00112 #define LCD_INCREMENT_NEGATIVE      0x00
00113
00114
00115 //LCD Command Initial State - Config
00116 #define LCD_DISPLAY_CONFIG          (LCD_DISPLAY_ON|LCD_CURSOR_OFF|LCD_BLINK_OFF)
00117 #define LCD_DISPLAY_INCREMENT       (LCD_INCREMENT|LCD_INCREMENT_NO_SHIFT)
00118
00119
00120 typedef struct
00121 {

```

```

00122     uint8_t row;
00123     uint8_t col;
00124     uint8_t display;
00125     uint8_t shift;
00126     uint8_t cgramAddress;
00127     uint8_t specialChar[8];    //defined by fonts
00128 }LCDStatus;
00129
00130
00131
00132 // \TODO: fix this shit. decide if is to be used with structs or no use at all.
00133 extern LCDStatus LCD0Status;
00134
00135
00136
00137
00138 //functions declarations
00139 void LCDInit(void);
00140 void LCDSendCmd(uint8_t cmd);
00141 void LCDSendChar(uint8_t txt);
00142 void LCDSend(uint8_t send);
00143 void LCDPosition(uint8_t row, uint8_t col);
00144 void LCDPositionNoDelay(uint8_t row, uint8_t col);
00145 void LCDSendString(uint8_t *string, uint8_t breakLine);
00146 void LCDSendNumStrict(int64_t num, uint8_t length,\
00147     uint8_t isSigned, uint8_t showZeros);
00148 void LCDSendNum(int64_t num, uint8_t length,\
00149     uint8_t isSigned, uint8_t showZeros);
00150 void LCDSendNumArray(uint8_t *vector);
00151 void LCDClear(void);
00152 void LCDSendHex(uint8_t *array);
00153
00154 //limited by uint64 max counting
00155 #define maxLengthOut 16
00156 void numToArray(int32_t num, uint8_t *array,\
00157     uint8_t length, uint16_t base);
00158
00159 void LCDRegisterSpecial(uint8_t number,\
00160     uint8_t *character);
00161 void LCDShift(uint8_t shift);
00162 void LCDHome(void);
00163
00164
00165 void arrayToNum(uint8_t *array, uint32_t *num, uint8_t base);
00166
00167
00168 void LCDSendVU(uint32_t num, uint32_t base);
00169
00170 #endif

```

## 5.43 my\_lib/my\_use.c File Reference

```
#include "my_use.h"
```

### Functions

- void \_\_inline [ShiftSerialSend](#) (uint32\_t data\_port, uint32\_t data\_pin, uint32\_t clk\_port, uint32\_t clk\_pin, uint8\_t text)

#### 5.43.1 Function Documentation

5.43.1.1 void \_\_inline [ShiftSerialSend](#) ( uint32\_t data\_port, uint32\_t data\_pin, uint32\_t clk\_port, uint32\_t clk\_pin, uint8\_t text )

Definition at line 9 of file [my\\_use.c](#).

## 5.44 my\_use.c

```

00001 #include "my_use.h"
00002

```

```

00003
00004
00005
00006 /*
00007  * Shift Serial Send function
00008  */
00009 void __inline ShiftSerialSend(uint32_t data_port, uint32_t data_pin, \
00010                             uint32_t clk_port, uint32_t clk_pin, uint8_t text)
00011 {
00012
00013     PinAddrClear(clk_port, clk_pin);
00014     char i=8;
00015     while(i>0)
00016     {
00017         if((text&0x80)==0)
00018             PinAddrClear(data_port, data_pin);
00019         else
00020             PinAddrSet(data_port, data_pin);
00021         text <=<= 1;
00022         i--;
00023         PinAddrSet(clk_port, clk_pin);
00024         SysDelay(2);
00025         PinAddrClear(clk_port, clk_pin);
00026         SysDelay(2);
00027     }
00028
00029 }
00030
00031
00032
00033
00034

```

## 5.45 my\_lib/my\_use.h File Reference

```

#include "depl_spc/includeAll_sw.h"
#include "depl_spc/includeAll_hw.h"

```

### Macros

- #define [bTrue0](#) 0x01
- #define [bTrue1](#) 0x02
- #define [bTrue2](#) 0x04
- #define [bTrue3](#) 0x08
- #define [bTrue4](#) 0x10
- #define [bTrue5](#) 0x20
- #define [bTrue6](#) 0x40
- #define [bTrue7](#) 0x80
- #define [charDecadeLength](#) 3
- #define [charBinaryLength](#) 8
- #define [shortDecadeLength](#) 5
- #define [shortBinaryLength](#) 16
- #define [intDecadeLength](#) 10
- #define [intBinaryLength](#) 32
- #define [limitCeilValue](#)(value, lim)
- #define [limitCycleValueUpZero](#)(value, lim)
- #define [limitCycleValueUpOff](#)(value, lim, reset)
- #define [limitFloorValue](#)(value, lim)

### Functions

- void [ShiftSerialSend](#) (uint32\_t data\_port, uint32\_t data\_pin, uint32\_t clk\_port, uint32\_t clk\_pin, uint8\_t text)

## 5.45.1 Macro Definition Documentation

### 5.45.1.1 `#define bTrue0 0x01`

Definition at line 7 of file [my\\_use.h](#).

### 5.45.1.2 `#define bTrue1 0x02`

Definition at line 8 of file [my\\_use.h](#).

### 5.45.1.3 `#define bTrue2 0x04`

Definition at line 9 of file [my\\_use.h](#).

### 5.45.1.4 `#define bTrue3 0x08`

Definition at line 10 of file [my\\_use.h](#).

### 5.45.1.5 `#define bTrue4 0x10`

Definition at line 11 of file [my\\_use.h](#).

### 5.45.1.6 `#define bTrue5 0x20`

Definition at line 12 of file [my\\_use.h](#).

### 5.45.1.7 `#define bTrue6 0x40`

Definition at line 13 of file [my\\_use.h](#).

### 5.45.1.8 `#define bTrue7 0x80`

Definition at line 14 of file [my\\_use.h](#).

### 5.45.1.9 `#define charBinaryLength 8`

Definition at line 17 of file [my\\_use.h](#).

### 5.45.1.10 `#define charDecadeLength 3`

Definition at line 16 of file [my\\_use.h](#).

### 5.45.1.11 `#define intBinaryLength 32`

Definition at line 23 of file [my\\_use.h](#).

### 5.45.1.12 `#define intDecadeLength 10`

Definition at line 22 of file [my\\_use.h](#).

#### 5.45.1.13 `#define limitCeilValue( value, lim )`

##### Value:

```
if (value>=lim) \
                                value=lim;
```

Definition at line 28 of file [my\\_use.h](#).

#### 5.45.1.14 `#define limitCycleValueUpOff( value, lim, reset )`

##### Value:

```
if (value>=lim) \
                                value=reset;
```

Definition at line 32 of file [my\\_use.h](#).

#### 5.45.1.15 `#define limitCycleValueUpZero( value, lim )`

##### Value:

```
if (value>=lim) \
                                value=0;
```

Definition at line 30 of file [my\\_use.h](#).

#### 5.45.1.16 `#define limitFloorValue( value, lim )`

##### Value:

```
if (value<=lim) \
                                value=lim;
```

Definition at line 34 of file [my\\_use.h](#).

#### 5.45.1.17 `#define shortBinaryLength 16`

Definition at line 20 of file [my\\_use.h](#).

#### 5.45.1.18 `#define shortDecadeLength 5`

Definition at line 19 of file [my\\_use.h](#).

### 5.45.2 Function Documentation

#### 5.45.2.1 `void ShiftSerialSend ( uint32_t data_port, uint32_t data_pin, uint32_t clk_port, uint32_t clk_pin, uint8_t text )`

Definition at line 9 of file [my\\_use.c](#).



## 5.46 my\_use.h

```

00001 #ifndef my_use_h
00002 #define my_use_h
00003
00004 #include "depl_spc/includeAll_sw.h"
00005 #include "depl_spc/includeAll_hw.h"
00006
00007 #define bTrue0          0x01
00008 #define bTrue1          0x02
00009 #define bTrue2          0x04
00010 #define bTrue3          0x08
00011 #define bTrue4          0x10
00012 #define bTrue5          0x20
00013 #define bTrue6          0x40
00014 #define bTrue7          0x80
00015
00016 #define charDecadeLength 3
00017 #define charBinaryLength 8
00018
00019 #define shortDecadeLength 5
00020 #define shortBinaryLength 16
00021
00022 #define intDecadeLength 10
00023 #define intBinaryLength 32
00024
00025
00026
00027 //function masks
00028 #define limitCeilValue(value, lim)          if (value>=lim)\
00029                                             value=lim;
00030 #define limitCycleValueUpZero(value,lim)    if (value>=lim)\
00031                                             value=0;
00032 #define limitCycleValueUpOff(value, lim, reset) if (value>=lim)\
00033                                             value=reset;
00034 #define limitFloorValue(value, lim)        if (value<=lim)\
00035                                             value=lim;
00036
00037
00038
00039
00040 /*
00041  * Function Declarations
00042  */
00043 void ShiftSerialSend(uint32_t data_port, uint32_t data_pin, \
00044                     uint32_t clk_port, uint32_t clk_pin, uint8_t text);
00045
00046
00047
00048
00049
00050
00051
00052
00053
00054 #endif// my_use_h

```

## 5.47 my\_lib/myUart.c File Reference

```
#include "myUart.h"
```

### Functions

- void [myUARTSendString](#) (uint32\_t instance, uint8\_t \*string)

### 5.47.1 Function Documentation

#### 5.47.1.1 void myUARTSendString ( uint32\_t instance, uint8\_t \* string )

Definition at line 17 of file [myUart.c](#).

## 5.48 myUart.c

```

00001 /*
00002  * myUart.c
00003  *
00004  * Created on: Nov 25, 2013
00005  * Author: rikardo
00006  */
00007
00008
00009 #include "myUart.h"
00010
00011
00012
00013 /*
00014  * Sends a string, NULL terminator
00015  * returns none
00016  */
00017 void myUARTSendString(uint32_t instance, uint8_t *string)
00018 {
00019     while(*string)
00020     {
00021         myUARTSend(instance, *string);
00022         string++;
00023         myUARTDelay(1);
00024     }
00025 }
00026

```

## 5.49 my\_lib/myUart.h File Reference

```

#include "depl_spc/includeAll_sw.h"
#include "depl_spc/includeAll_hw.h"

```

### Data Structures

- struct [UARTInstance](#)

### Macros

- #define [UART\\_NORMAL\\_OP\\_MODE](#) 0x0001
- #define [UART\\_DIRECT\\_TRANSFER\\_MODE](#) 0x0002
- #define [UART\\_BUFFER\\_SIZE](#) 30
- #define [myUARTPC](#) UART0\_BASE
- #define [myUARTSend](#)(instance, charToGo) MAP\_UARTCharPutNonBlocking(instance, charToGo)
- #define [myUARTDelay](#)(delay) [SysDelay](#)(delay)
- #define [myUARTPCSend](#)(charToGo) [myUARTSend](#)(myUARTPC, charToGo)

### Functions

- void [myUARTSendString](#) (uint32\_t instance, uint8\_t \*string)

### 5.49.1 Macro Definition Documentation

#### 5.49.1.1 #define myUARTDelay( delay ) SysDelay(delay)

Definition at line 39 of file [myUart.h](#).

5.49.1.2 `#define myUARTPC UART0_BASE`

Definition at line 36 of file [myUart.h](#).

5.49.1.3 `#define myUARTPCSend( charToGo ) myUARTSend(myUARTPC, charToGo)`

Definition at line 40 of file [myUart.h](#).

5.49.1.4 `#define myUARTSend( instance, charToGo ) MAP_UARTCharPutNonBlocking(instance, charToGo)`

Definition at line 38 of file [myUart.h](#).

5.49.1.5 `#define UART_BUFFER_SIZE 30`

Definition at line 23 of file [myUart.h](#).

5.49.1.6 `#define UART_DIRECT_TRANSFER_MODE 0x0002`

Definition at line 19 of file [myUart.h](#).

5.49.1.7 `#define UART_NORMAL_OP_MODE 0x0001`

Definition at line 18 of file [myUart.h](#).

## 5.49.2 Function Documentation

5.49.2.1 `void myUARTSendString ( uint32_t instance, uint8_t * string )`

Definition at line 17 of file [myUart.c](#).

## 5.50 myUart.h

```

00001 /*
00002  * myUart.h
00003  *
00004  * Created on: Nov 25, 2013
00005  * Author: rikardo
00006  */
00007
00008 #ifndef MYUART_H_
00009 #define MYUART_H_
00010
00011 #include "depl_spc/includeAll_sw.h"
00012 #include "depl_spc/includeAll_hw.h"
00013
00014
00015 /*
00016  * UART operation mode masks
00017  */
00018 #define UART_NORMAL_OP_MODE 0x0001
00019 #define UART_DIRECT_TRANSFER_MODE 0x0002
00020
00021
00022
00023 #define UART_BUFFER_SIZE 30
00024
00025 typedef struct{
00026     uint8_t RxBuffer[UART_BUFFER_SIZE];
00027     uint8_t RxBufferPtr;
00028     uint8_t TxBuffer[UART_BUFFER_SIZE];
00029     uint8_t TxBufferPtr;

```

```

00030     uint16_t Mode;
00031     uint8_t TxLastSent[UART_BUFFER_SIZE];
00032     uint8_t TxLastSentPtr;
00033 }UARTInstance;
00034
00035
00036 #define myUARTPC                                UART0_BASE
00037
00038 #define myUARTSend(instance, charToGo)          MAP_UARTCharPutNonBlocking(instance, charToGo)
00039 #define myUARTDelay(delay)                      SysDelay(delay)
00040 #define myUARTPCSend(charToGo)                  myUARTSend(myUARTPC, charToGo)
00041
00042
00043
00044 void myUARTSendString(uint32_t instance, uint8_t *string);
00045
00046
00047
00048
00049 #endif /* MYUART_H_ */

```

## 5.51 my\_lib/uk\_mapping.h File Reference

### Macros

- #define UKM\_SPACE 32
- #define UKM\_BSPACE 127
- #define UKM\_BS 8
- #define UKM\_ENTER 13
- #define UKM\_TILDA 126
- #define UKM\_ESCAPE 27
- #define UKM\_TAB 9
- #define UKM\_CTRL\_E 5
- #define UKM\_ASCII\_TAB 9
- #define UKM\_ASCII\_LF 10
- #define UKM\_LF 10
- #define UKM\_LINEFEED 10
- #define UKM\_CR 13
- #define UKM\_ASCII\_VT 11
- #define UKM\_VT 11
- #define UKM\_ASCII\_FF 12
- #define UKM\_CLS 12

### 5.51.1 Macro Definition Documentation

#### 5.51.1.1 #define UKM\_ASCII\_FF 12

Definition at line 28 of file [uk\\_mapping.h](#).

#### 5.51.1.2 #define UKM\_ASCII\_LF 10

Definition at line 22 of file [uk\\_mapping.h](#).

#### 5.51.1.3 #define UKM\_ASCII\_TAB 9

Definition at line 21 of file [uk\\_mapping.h](#).

#### 5.51.1.4 `#define UKM_ASCII_VT 11`

Definition at line 26 of file [uk\\_mapping.h](#).

#### 5.51.1.5 `#define UKM_BS 8`

Definition at line 15 of file [uk\\_mapping.h](#).

#### 5.51.1.6 `#define UKM_BSPACE 127`

Definition at line 14 of file [uk\\_mapping.h](#).

#### 5.51.1.7 `#define UKM_CLS 12`

Definition at line 29 of file [uk\\_mapping.h](#).

#### 5.51.1.8 `#define UKM_CR 13`

Definition at line 25 of file [uk\\_mapping.h](#).

#### 5.51.1.9 `#define UKM_CTRL_E 5`

Definition at line 20 of file [uk\\_mapping.h](#).

#### 5.51.1.10 `#define UKM_ENTER 13`

Definition at line 16 of file [uk\\_mapping.h](#).

#### 5.51.1.11 `#define UKM_ESCAPE 27`

Definition at line 18 of file [uk\\_mapping.h](#).

#### 5.51.1.12 `#define UKM_LF 10`

Definition at line 23 of file [uk\\_mapping.h](#).

#### 5.51.1.13 `#define UKM_LINEFEED 10`

Definition at line 24 of file [uk\\_mapping.h](#).

#### 5.51.1.14 `#define UKM_SPACE 32`

Definition at line 13 of file [uk\\_mapping.h](#).

#### 5.51.1.15 `#define UKM_TAB 9`

Definition at line 19 of file [uk\\_mapping.h](#).

#### 5.51.1.16 #define UKM\_TILDA 126

Definition at line 17 of file [uk\\_mapping.h](#).

#### 5.51.1.17 #define UKM\_VT 11

Definition at line 27 of file [uk\\_mapping.h](#).

### 5.52 uk\_mapping.h

```

00001 /*
00002  * uart_keyboard_mapping.h
00003  *
00004  * Created on: Nov 27, 2013
00005  * Author: rikardo
00006  */
00007
00008 #ifndef UART_KEYBOARD_MAPPING_H_
00009 #define UART_KEYBOARD_MAPPING_H_
00010
00011
00012
00013 #define UKM_SPACE 32
00014 #define UKM_BSPACE 127
00015 #define UKM_BS 8
00016 #define UKM_ENTER 13
00017 #define UKM_TILDA 126
00018 #define UKM_ESCAPE 27
00019 #define UKM_TAB 9
00020 #define UKM_CTRL_E 5
00021 #define UKM_ASCII_TAB 9
00022 #define UKM_ASCII_LF 10
00023 #define UKM_LF 10
00024 #define UKM_LINEFEED 10
00025 #define UKM_CR 13
00026 #define UKM_ASCII_VT 11
00027 #define UKM_VT 11
00028 #define UKM_ASCII_FF 12
00029 #define UKM_CLS 12
00030
00031
00032
00033 #endif /* UART_KEYBOARD_MAPPING_H_ */

```

### 5.53 README.md File Reference

#### 5.54 README.md

```

00001 uIntPLib
00002 =====
00003
00004 Universal Integrated Peripheral Library
00005
00006 This is a library made with functions masks to medium level programming.
00007 Intended to make code more portable, while maintaining its performance.

```

# Index

ASCII\_ACK  
    [ascii.h](#), 23  
ASCII\_BEL  
    [ascii.h](#), 23  
ASCII\_BS  
    [ascii.h](#), 23  
ASCII\_CAN  
    [ascii.h](#), 23  
ASCII\_CR  
    [ascii.h](#), 23  
ASCII\_DC1  
    [ascii.h](#), 23  
ASCII\_DC2  
    [ascii.h](#), 24  
ASCII\_DC3  
    [ascii.h](#), 24  
ASCII\_DC4  
    [ascii.h](#), 24  
ASCII\_DLE  
    [ascii.h](#), 24  
ASCII\_EM  
    [ascii.h](#), 24  
ASCII\_ENQ  
    [ascii.h](#), 24  
ASCII\_EOT  
    [ascii.h](#), 24  
ASCII\_ESC  
    [ascii.h](#), 24  
ASCII\_ETB  
    [ascii.h](#), 24  
ASCII\_ETX  
    [ascii.h](#), 24  
ASCII\_FF  
    [ascii.h](#), 24  
ASCII\_FS  
    [ascii.h](#), 24  
ASCII\_GS  
    [ascii.h](#), 25  
ASCII\_HT  
    [ascii.h](#), 25  
ASCII\_LF  
    [ascii.h](#), 25  
ASCII\_NAK  
    [ascii.h](#), 25  
ASCII\_NULL  
    [ascii.h](#), 25  
ASCII\_RS  
    [ascii.h](#), 25  
ASCII\_SI  
    [ascii.h](#), 25  
ASCII\_SO  
    [ascii.h](#), 25  
ASCII\_SOH  
    [ascii.h](#), 25  
ASCII\_STX  
    [ascii.h](#), 25  
ASCII\_SUB  
    [ascii.h](#), 25  
ASCII\_SYN  
    [ascii.h](#), 25  
ASCII\_US  
    [ascii.h](#), 26  
ASCII\_VT  
    [ascii.h](#), 26  
arrayToNum  
    [lcd.c](#), 41  
    [lcd.h](#), 53  
[ascii.h](#)  
    ASCII\_ACK, 23  
    ASCII\_BEL, 23  
    ASCII\_BS, 23  
    ASCII\_CAN, 23  
    ASCII\_CR, 23  
    ASCII\_DC1, 23  
    ASCII\_DC2, 24  
    ASCII\_DC3, 24  
    ASCII\_DC4, 24  
    ASCII\_DLE, 24  
    ASCII\_EM, 24  
    ASCII\_ENQ, 24  
    ASCII\_EOT, 24  
    ASCII\_ESC, 24  
    ASCII\_ETB, 24  
    ASCII\_ETX, 24  
    ASCII\_FF, 24  
    ASCII\_FS, 24  
    ASCII\_GS, 25  
    ASCII\_HT, 25  
    ASCII\_LF, 25  
    ASCII\_NAK, 25  
    ASCII\_NULL, 25  
    ASCII\_RS, 25  
    ASCII\_SI, 25  
    ASCII\_SO, 25  
    ASCII\_SOH, 25  
    ASCII\_STX, 25  
    ASCII\_SUB, 25  
    ASCII\_SYN, 25

- ASCII\_US, [26](#)
- ASCII\_VT, [26](#)
- bTrue0
  - my\_use.h, [59](#)
- bTrue1
  - my\_use.h, [59](#)
- bTrue2
  - my\_use.h, [59](#)
- bTrue3
  - my\_use.h, [59](#)
- bTrue4
  - my\_use.h, [59](#)
- bTrue5
  - my\_use.h, [59](#)
- bTrue6
  - my\_use.h, [59](#)
- bTrue7
  - my\_use.h, [59](#)
- BUS\_CLOCK
  - globalParam.h, [17](#)
- BUSHZ\_CLOCK
  - globalParam.h, [17](#)
- CPU\_CLOCK
  - globalParam.h, [17](#)
- CPUHZ\_CLOCK
  - globalParam.h, [17](#)
- CarrierFrequency
  - IRInstance, [8](#)
- CarrierPeriod
  - IRInstance, [8](#)
- cgramAdress
  - LCDStatus, [9](#)
- charBinaryLength
  - my\_use.h, [59](#)
- charDecadeLength
  - my\_use.h, [59](#)
- charIn
  - CommandInstance, [7](#)
- charOut
  - CommandInstance, [7](#)
- charOutPtr
  - CommandInstance, [7](#)
- cmd\_sort.c
  - CommandSort, [27](#)
- cmd\_sort.h
  - CommandSort, [28](#)
  - MAX\_BUFFER\_SIZE, [28](#)
- cmdBuffer
  - CommandInstance, [7](#)
- col
  - LCDStatus, [9](#)
- CommandInstance, [7](#)
  - charIn, [7](#)
  - charOut, [7](#)
  - charOutPtr, [7](#)
  - cmdBuffer, [7](#)
- CommandSort
  - cmd\_sort.c, [27](#)
  - cmd\_sort.h, [28](#)
- depl\_spc/chip\_specific.c, [13](#)
- depl\_spc/chip\_specific.h, [13](#)
- depl\_spc/cmd\_list.h, [14](#)
- depl\_spc/device\_init/hardwareInit.c, [14](#)
- depl\_spc/device\_init/hardwareInit.h, [14](#), [15](#)
- depl\_spc/device\_init/softwareInit.c, [15](#)
- depl\_spc/device\_init/softwareInit.h, [16](#)
- depl\_spc/globalParam.h, [16](#), [17](#)
- depl\_spc/includeAll\_hw.h, [18](#)
- depl\_spc/includeAll\_sw.h, [18](#)
- depl\_spc/lib\_comp/external\_cons.h, [19](#), [20](#)
- depl\_spc/lib\_comp/libraryCompatible.h, [21](#), [22](#)
- depl\_spc/variables.h, [22](#)
- display
  - LCDStatus, [9](#)
- external\_cons.h
  - LCD\_CLK\_Pin, [19](#)
  - LCD\_CLK\_Port, [19](#)
  - LCD\_DTA\_Pin, [19](#)
  - LCD\_DTA\_Port, [19](#)
  - LCD\_EN\_Pin, [20](#)
  - LCD\_EN\_Port, [20](#)
  - LCD\_RS\_Pin, [20](#)
  - LCD\_RS\_Port, [20](#)
  - LCD\_char\_heigh, [19](#)
  - LCD\_char\_width, [19](#)
  - LCD\_col\_num, [19](#)
  - LCD\_row\_num, [20](#)
- false
  - lcd.c, [41](#)
- globalParam.h
  - BUS\_CLOCK, [17](#)
  - BUSHZ\_CLOCK, [17](#)
  - CPU\_CLOCK, [17](#)
  - CPUHZ\_CLOCK, [17](#)
  - LCD\_SPLASHSCREEN, [17](#)
  - LCD\_SPLASHSCREEN1, [17](#)
  - PROJECT\_NAME, [17](#)
- gpioPin\_masks.h
  - IOPin\_0, [29](#)
  - IOPin\_1, [29](#)
  - IOPin\_10, [29](#)
  - IOPin\_11, [29](#)
  - IOPin\_12, [29](#)
  - IOPin\_13, [29](#)
  - IOPin\_14, [29](#)
  - IOPin\_15, [29](#)
  - IOPin\_16, [30](#)
  - IOPin\_17, [30](#)
  - IOPin\_18, [30](#)
  - IOPin\_19, [30](#)
  - IOPin\_2, [30](#)
  - IOPin\_20, [30](#)



- IOPin\_21, [30](#)
- IOPin\_22, [30](#)
- IOPin\_23, [30](#)
- IOPin\_24, [30](#)
- IOPin\_25, [30](#)
- IOPin\_26, [30](#)
- IOPin\_27, [31](#)
- IOPin\_28, [31](#)
- IOPin\_29, [31](#)
- IOPin\_3, [31](#)
- IOPin\_30, [31](#)
- IOPin\_31, [31](#)
- IOPin\_4, [31](#)
- IOPin\_5, [31](#)
- IOPin\_6, [31](#)
- IOPin\_7, [31](#)
- IOPin\_8, [31](#)
- IOPin\_9, [31](#)
- HardwareInit
  - hardwareInit.c, [14](#)
  - hardwareInit.h, [15](#)
- hardwareInit.c
  - HardwareInit, [14](#)
- hardwareInit.h
  - HardwareInit, [15](#)
- IOPin\_0
  - gpioPin\_masks.h, [29](#)
- IOPin\_1
  - gpioPin\_masks.h, [29](#)
- IOPin\_10
  - gpioPin\_masks.h, [29](#)
- IOPin\_11
  - gpioPin\_masks.h, [29](#)
- IOPin\_12
  - gpioPin\_masks.h, [29](#)
- IOPin\_13
  - gpioPin\_masks.h, [29](#)
- IOPin\_14
  - gpioPin\_masks.h, [29](#)
- IOPin\_15
  - gpioPin\_masks.h, [29](#)
- IOPin\_16
  - gpioPin\_masks.h, [30](#)
- IOPin\_17
  - gpioPin\_masks.h, [30](#)
- IOPin\_18
  - gpioPin\_masks.h, [30](#)
- IOPin\_19
  - gpioPin\_masks.h, [30](#)
- IOPin\_2
  - gpioPin\_masks.h, [30](#)
- IOPin\_20
  - gpioPin\_masks.h, [30](#)
- IOPin\_21
  - gpioPin\_masks.h, [30](#)
- IOPin\_22
  - gpioPin\_masks.h, [30](#)
- IOPin\_23
  - gpioPin\_masks.h, [30](#)
- IOPin\_24
  - gpioPin\_masks.h, [30](#)
- IOPin\_25
  - gpioPin\_masks.h, [30](#)
- IOPin\_26
  - gpioPin\_masks.h, [30](#)
- IOPin\_27
  - gpioPin\_masks.h, [31](#)
- IOPin\_28
  - gpioPin\_masks.h, [31](#)
- IOPin\_29
  - gpioPin\_masks.h, [31](#)
- IOPin\_3
  - gpioPin\_masks.h, [31](#)
- IOPin\_30
  - gpioPin\_masks.h, [31](#)
- IOPin\_31
  - gpioPin\_masks.h, [31](#)
- IOPin\_4
  - gpioPin\_masks.h, [31](#)
- IOPin\_5
  - gpioPin\_masks.h, [31](#)
- IOPin\_6
  - gpioPin\_masks.h, [31](#)
- IOPin\_7
  - gpioPin\_masks.h, [31](#)
- IOPin\_8
  - gpioPin\_masks.h, [31](#)
- IOPin\_9
  - gpioPin\_masks.h, [31](#)
- IR\_BY\_SOFTWARE
  - ir.h, [37](#)
- IR\_BY\_TIMER
  - ir.h, [37](#)
- IR\_BY\_UART
  - ir.h, [37](#)
- IR\_MAX\_INSTANCES
  - ir.h, [37](#)
- IR\_MY\_PROTOCOL
  - ir.h, [37](#)
- IR\_NEC\_EXTENDED
  - ir.h, [37](#)
- IR\_NEC\_PROTOCOL
  - ir.h, [37](#)
- IR\_RC5\_PROTOCOL
  - ir.h, [37](#)
- IRByteBySoftware
  - ir.c, [32](#)
  - ir.h, [38](#)
- IRDelay
  - ir.h, [38](#)
- IRDelayMs
  - ir.h, [38](#)
- IRDelayUs
  - ir.h, [38](#)
- IRInit

- ir.c, [32](#)
  - ir.h, [38](#)
- IRInstance, [8](#)
  - CarrierFrequency, [8](#)
  - CarrierPeriod, [8](#)
  - LastData, [8](#)
  - Mode, [8](#)
  - Pulses, [8](#)
  - ReceiveAddress, [8](#)
  - ReceiveBuffer, [8](#)
  - RxPin, [9](#)
  - RxPort, [9](#)
  - TxPin, [9](#)
  - TxPort, [9](#)
- IRPinClear
  - ir.h, [38](#)
- IRPinSet
  - ir.h, [38](#)
- IRRepeat
  - ir.c, [33](#)
  - ir.h, [38](#)
- IRSend
  - ir.c, [33](#)
  - ir.h, [38](#)
- intBinaryLength
  - my\_use.h, [59](#)
- intDecadeLength
  - my\_use.h, [59](#)
- ir.c
  - IRByteBySoftware, [32](#)
  - IRInit, [32](#)
  - IRRepeat, [33](#)
  - IRSend, [33](#)
- ir.h
  - IR\_BY\_SOFTWARE, [37](#)
  - IR\_BY\_TIMER, [37](#)
  - IR\_BY\_UART, [37](#)
  - IR\_MAX\_INSTANCES, [37](#)
  - IR\_MY\_PROTOCOL, [37](#)
  - IR\_NEC\_EXTENDED, [37](#)
  - IR\_NEC\_PROTOCOL, [37](#)
  - IR\_RC5\_PROTOCOL, [37](#)
  - IRByteBySoftware, [38](#)
  - IRDelay, [38](#)
  - IRDelayMs, [38](#)
  - IRDelayUs, [38](#)
  - IRInit, [38](#)
  - IRPinClear, [38](#)
  - IRPinSet, [38](#)
  - IRRepeat, [38](#)
  - IRSend, [38](#)
  - NEC\_PULSE\_TIME, [38](#)
  - RC5\_PULSE\_TIME, [38](#)
- LCD0Status
  - lcd.h, [55](#)
- LCD\_BLINK\_OFF
  - lcd.h, [50](#)
- LCD\_BLINK\_ON
  - lcd.h, [50](#)
- LCD\_CLK\_High
  - lcd.h, [50](#)
- LCD\_CLK\_LoW
  - lcd.h, [51](#)
- LCD\_CLK\_Pin
  - external\_cons.h, [19](#)
- LCD\_CLK\_Port
  - external\_cons.h, [19](#)
- LCD\_CURSOR\_OFF
  - lcd.h, [51](#)
- LCD\_CURSOR\_ON
  - lcd.h, [51](#)
- LCD\_CmdInit\_Vector
  - lcd.c, [43](#)
- LCD\_DISPLAY\_CONFIG
  - lcd.h, [51](#)
- LCD\_DISPLAY\_OFF
  - lcd.h, [51](#)
- LCD\_DISPLAY\_ON
  - lcd.h, [51](#)
- LCD\_DTA\_Pin
  - external\_cons.h, [19](#)
- LCD\_DTA\_Port
  - external\_cons.h, [19](#)
- LCD\_DTA\_Send
  - lcd.h, [51](#)
- LCD\_EN\_High
  - lcd.h, [51](#)
- LCD\_EN\_Low
  - lcd.h, [51](#)
- LCD\_EN\_Pin
  - external\_cons.h, [20](#)
- LCD\_EN\_Port
  - external\_cons.h, [20](#)
- LCD\_INCREMENT
  - lcd.h, [51](#)
- LCD\_INCREMENT\_SHIFT
  - lcd.h, [52](#)
- LCD\_InitDelay\_Vector
  - lcd.c, [43](#)
- LCD\_RS\_High
  - lcd.h, [52](#)
- LCD\_RS\_Low
  - lcd.h, [52](#)
- LCD\_RS\_Pin
  - external\_cons.h, [20](#)
- LCD\_RS\_Port
  - external\_cons.h, [20](#)
- LCD\_SET\_CGRAM
  - lcd.h, [52](#)
- LCD\_SHIFT
  - lcd.h, [52](#)
- LCD\_SHIFT\_CURSOR
  - lcd.h, [52](#)
- LCD\_SHIFT\_DISPLAY
  - lcd.h, [52](#)
- LCD\_SHIFT\_LEFT
  - lcd.h, [52](#)

- lcd.h, [52](#)
- LCD\_SHIFT\_RIGHT
  - lcd.h, [52](#)
- LCD\_SPLASHSCREEN
  - globalParam.h, [17](#)
- LCD\_SPLASHSCREEN1
  - globalParam.h, [17](#)
- LCD\_char\_heigh
  - external\_cons.h, [19](#)
- LCD\_char\_width
  - external\_cons.h, [19](#)
- LCD\_col\_num
  - external\_cons.h, [19](#)
- LCD\_row\_num
  - external\_cons.h, [20](#)
- LCD\_splashscreen2\_row1
  - lcd.h, [53](#)
- LCD\_splashscreen2\_row2
  - lcd.h, [53](#)
- LCD\_splashscreen\_row1
  - lcd.h, [53](#)
- LCD\_splashscreen\_row2
  - lcd.h, [53](#)
- LCDClear
  - lcd.c, [41](#)
  - lcd.h, [53](#)
- LCDDelay
  - lcd.h, [53](#)
- LCDDisplayOn
  - lcd.c, [41](#)
- LCDHome
  - lcd.c, [41](#)
  - lcd.h, [53](#)
- LCDInit
  - lcd.c, [41](#)
  - lcd.h, [54](#)
- LCDPinClear
  - lcd.h, [53](#)
- LCDPinSet
  - lcd.h, [53](#)
- LCDPosition
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDPositionNoDelay
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDRegisterSpecial
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSend
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendChar
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendCmd
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendHex
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendNum
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendNumArray
  - lcd.c, [42](#)
  - lcd.h, [54](#)
- LCDSendNumStrict
  - lcd.c, [42](#)
  - lcd.h, [55](#)
- LCDSendString
  - lcd.c, [43](#)
  - lcd.h, [55](#)
- LCDSendVU
  - lcd.c, [43](#)
  - lcd.h, [55](#)
- LCDShift
  - lcd.c, [43](#)
  - lcd.h, [55](#)
- LCDStatus, [9](#)
  - cgramAddress, [9](#)
  - col, [9](#)
  - display, [9](#)
  - row, [10](#)
  - shift, [10](#)
  - specialChar, [10](#)
- LastData
  - IRInstance, [8](#)
- lcd.c
  - arrayToNum, [41](#)
  - false, [41](#)
  - LCD\_CmdInit\_Vector, [43](#)
  - LCD\_InitDelay\_Vector, [43](#)
  - LCDClear, [41](#)
  - LCDDisplayOn, [41](#)
  - LCDHome, [41](#)
  - LCDInit, [41](#)
  - LCDPosition, [42](#)
  - LCDPositionNoDelay, [42](#)
  - LCDRegisterSpecial, [42](#)
  - LCDSend, [42](#)
  - LCDSendChar, [42](#)
  - LCDSendCmd, [42](#)
  - LCDSendHex, [42](#)
  - LCDSendNum, [42](#)
  - LCDSendNumArray, [42](#)
  - LCDSendNumStrict, [42](#)
  - LCDSendString, [43](#)
  - LCDSendVU, [43](#)
  - LCDShift, [43](#)
  - lcd\_vector\_index, [41](#)
  - numToArray, [43](#)
  - true, [41](#)
  - trueDefinedLCD, [41](#)
- lcd.h
  - arrayToNum, [53](#)

- LCD0Status, 55
- LCD\_BLINK\_OFF, 50
- LCD\_BLINK\_ON, 50
- LCD\_CLK\_High, 50
- LCD\_CLK\_LoW, 51
- LCD\_CURSOR\_OFF, 51
- LCD\_CURSOR\_ON, 51
- LCD\_DISPLAY\_CONFIG, 51
- LCD\_DISPLAY\_OFF, 51
- LCD\_DISPLAY\_ON, 51
- LCD\_DTA\_Send, 51
- LCD\_EN\_High, 51
- LCD\_EN\_Low, 51
- LCD\_INCREMENT, 51
- LCD\_INCREMENT\_SHIFT, 52
- LCD\_RS\_High, 52
- LCD\_RS\_Low, 52
- LCD\_SET\_CGRAM, 52
- LCD\_SHIFT, 52
- LCD\_SHIFT\_CURSOR, 52
- LCD\_SHIFT\_DISPLAY, 52
- LCD\_SHIFT\_LEFT, 52
- LCD\_SHIFT\_RIGHT, 52
- LCD\_splashscreen2\_row1, 53
- LCD\_splashscreen2\_row2, 53
- LCD\_splashscreen\_row1, 53
- LCD\_splashscreen\_row2, 53
- LCDClear, 53
- LCDDelay, 53
- LCDHome, 53
- LCDInit, 54
- LCDPinClear, 53
- LCDPinSet, 53
- LCDDPosition, 54
- LCDDPositionNoDelay, 54
- LCDRegisterSpecial, 54
- LCDSend, 54
- LCDSendChar, 54
- LCDSendCmd, 54
- LCDSendHex, 54
- LCDSendNum, 54
- LCDSendNumArray, 54
- LCDSendNumStrict, 55
- LCDSendString, 55
- LCDSendVU, 55
- LCDSHift, 55
- maxLengthOut, 53
- numToArray, 55
- lcd\_vector\_index
  - lcd.c, 41
- libraryCompatible.h
  - PinAddrClear, 21
  - PinAddrSet, 21
  - PinClear, 21
  - PinSet, 21
  - PinToggle, 21
  - SysDelay, 21
  - SysDelayMs, 21
  - SysDelayUs, 21
- limitCeilValue
  - my\_use.h, 59
- limitCycleValueUpOff
  - my\_use.h, 60
- limitCycleValueUpZero
  - my\_use.h, 60
- limitFloorValue
  - my\_use.h, 60
- MAX\_BUFFER\_SIZE
  - cmd\_sort.h, 28
- maxLengthOut
  - lcd.h, 53
- Mode
  - IRInstance, 8
  - UARTInstance, 10
- my\_lib/ascii.h, 22, 26
- my\_lib/cmd\_sort.c, 27
- my\_lib/cmd\_sort.h, 27, 28
- my\_lib/gpioPin\_masks.h, 28, 32
- my\_lib/ir.c, 32, 33
- my\_lib/ir.h, 36, 39
- my\_lib/lcd.c, 40, 43
- my\_lib/lcd.h, 49, 55
- my\_lib/my\_use.c, 57
- my\_lib/my\_use.h, 58, 61
- my\_lib/myUart.c, 61, 62
- my\_lib/myUart.h, 62, 63
- my\_lib/uk\_mapping.h, 64, 66
- my\_use.c
  - ShiftSerialSend, 57
- my\_use.h
  - bTrue0, 59
  - bTrue1, 59
  - bTrue2, 59
  - bTrue3, 59
  - bTrue4, 59
  - bTrue5, 59
  - bTrue6, 59
  - bTrue7, 59
  - charBinaryLength, 59
  - charDecadeLength, 59
  - intBinaryLength, 59
  - intDecadeLength, 59
  - limitCeilValue, 59
  - limitCycleValueUpOff, 60
  - limitCycleValueUpZero, 60
  - limitFloorValue, 60
  - ShiftSerialSend, 60
  - shortBinaryLength, 60
  - shortDecadeLength, 60
- myUARTDelay
  - myUart.h, 62
- myUARTPC
  - myUart.h, 62
- myUARTPCSend
  - myUart.h, 63
- myUARTSend

- myUart.h, 63
- myUARTSendString
  - myUart.c, 61
  - myUart.h, 63
- myUart.c
  - myUARTSendString, 61
- myUart.h
  - myUARTDelay, 62
  - myUARTPC, 62
  - myUARTPCSend, 63
  - myUARTSend, 63
  - myUARTSendString, 63
  - UART\_BUFFER\_SIZE, 63
- NEC\_PULSE\_TIME
  - ir.h, 38
- numToArray
  - lcd.c, 43
  - lcd.h, 55
- PROJECT\_NAME
  - globalParam.h, 17
- PinAddrClear
  - libraryCompatible.h, 21
- PinAddrSet
  - libraryCompatible.h, 21
- PinClear
  - libraryCompatible.h, 21
- PinSet
  - libraryCompatible.h, 21
- PinToggle
  - libraryCompatible.h, 21
- Pulses
  - IRInstance, 8
- RC5\_PULSE\_TIME
  - ir.h, 38
- README.md, 66
- ReceiveAddress
  - IRInstance, 8
- ReceiveBuffer
  - IRInstance, 8
- row
  - LCDStatus, 10
- RxBuffer
  - UARTInstance, 10
- RxBufferPtr
  - UARTInstance, 10
- RxPin
  - IRInstance, 9
- RxPort
  - IRInstance, 9
- shift
  - LCDStatus, 10
- ShiftSerialSend
  - my\_use.c, 57
  - my\_use.h, 60
- shortBinaryLength
  - my\_use.h, 60
- shortDecadeLength
  - my\_use.h, 60
- SoftwareInit
  - softwareInit.c, 15
  - softwareInit.h, 16
- softwareInit.c
  - SoftwareInit, 15
- softwareInit.h
  - SoftwareInit, 16
- specialChar
  - LCDStatus, 10
- SysDelay
  - libraryCompatible.h, 21
- SysDelayMs
  - libraryCompatible.h, 21
- SysDelayUs
  - libraryCompatible.h, 21
- true
  - lcd.c, 41
- trueDefinedLCD
  - lcd.c, 41
- TxBuffer
  - UARTInstance, 10
- TxBufferPtr
  - UARTInstance, 11
- TxLastSent
  - UARTInstance, 11
- TxLastSentPtr
  - UARTInstance, 11
- TxPin
  - IRInstance, 9
- TxPort
  - IRInstance, 9
- UART\_BUFFER\_SIZE
  - myUart.h, 63
- UARTInstance, 10
  - Mode, 10
  - RxBuffer, 10
  - RxBufferPtr, 10
  - TxBuffer, 10
  - TxBufferPtr, 11
  - TxLastSent, 11
  - TxLastSentPtr, 11
- UKM\_ASCII\_FF
  - uk\_mapping.h, 64
- UKM\_ASCII\_LF
  - uk\_mapping.h, 64
- UKM\_ASCII\_TAB
  - uk\_mapping.h, 64
- UKM\_ASCII\_VT
  - uk\_mapping.h, 64
- UKM\_BS
  - uk\_mapping.h, 65
- UKM\_BSPACE
  - uk\_mapping.h, 65
- UKM\_CLS

- [uk\\_mapping.h](#), [65](#)
- UKM\_CR
  - [uk\\_mapping.h](#), [65](#)
- UKM\_CTRL\_E
  - [uk\\_mapping.h](#), [65](#)
- UKM\_ENTER
  - [uk\\_mapping.h](#), [65](#)
- UKM\_ESCAPE
  - [uk\\_mapping.h](#), [65](#)
- UKM\_LF
  - [uk\\_mapping.h](#), [65](#)
- UKM\_LINEFEED
  - [uk\\_mapping.h](#), [65](#)
- UKM\_SPACE
  - [uk\\_mapping.h](#), [65](#)
- UKM\_TAB
  - [uk\\_mapping.h](#), [65](#)
- UKM\_TILDA
  - [uk\\_mapping.h](#), [65](#)
- UKM\_VT
  - [uk\\_mapping.h](#), [66](#)
- [uk\\_mapping.h](#)
  - [UKM\\_ASCII\\_FF](#), [64](#)
  - [UKM\\_ASCII\\_LF](#), [64](#)
  - [UKM\\_ASCII\\_TAB](#), [64](#)
  - [UKM\\_ASCII\\_VT](#), [64](#)
  - [UKM\\_BS](#), [65](#)
  - [UKM\\_BSPACE](#), [65](#)
  - [UKM\\_CLS](#), [65](#)
  - [UKM\\_CR](#), [65](#)
  - [UKM\\_CTRL\\_E](#), [65](#)
  - [UKM\\_ENTER](#), [65](#)
  - [UKM\\_ESCAPE](#), [65](#)
  - [UKM\\_LF](#), [65](#)
  - [UKM\\_LINEFEED](#), [65](#)
  - [UKM\\_SPACE](#), [65](#)
  - [UKM\\_TAB](#), [65](#)
  - [UKM\\_TILDA](#), [65](#)
  - [UKM\\_VT](#), [66](#)
- [variable\\_h](#)
  - [variables.h](#), [22](#)
- [variables.h](#)
  - [variable\\_h](#), [22](#)