intLib

Generated by Doxygen 1.8.6

Thu Apr 10 2014 20:58:11

Contents

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 maintaning its performance.

Doxyen generated documentation is located at latex/refman.pdf Complete documentation is under construction.

Chapter 2

uIntPLib

Universal Integrated Peripheral Library

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

Doxyen generated documentation is located at latex/refman.pdf Complete documentation is under construction.

Chapter 3

Todo List

Global LCD0Status

create masks for LCD commands.

Global LCDSend (uint8_t send)

make function to send parallel data.

Chapter 4

Module Documentation

4.1 Lcd_h

Data Structures

- struct LCDStatus

Macros

- #define **true** 1
- #define **false** 0
- #define **trueDefinedLCD**
- #define lcd_vector_index 7

  Initialization command sequence length.
- #define **maxLengthOut** 16

Functions

- void LCDInit (void)

  Initializes the LCD Module.
- void LCDSendCmd (uint8_t cmd)

  Send a Command to the LCD.
- void LCDSendChar (uint8_t txt)

  send single character to LCD.
- __inline void LCDSend (uint8_t send)

  Send data to LCD, no RS control.
- void LCDPosition (uint8_t row, uint8_t col)

  Set LCD write position.
- void LCDPositionNoDelay (uint8_t row, uint8_t col)

  Set LCD write position, no delay in function.
- void LCDSendString (uint8_t ∗string, uint8_t breakLine)

  Send string to LCD Writes a string of characteres on display Processes according to the ASCII code 0 - NULL.
- void LCDSendNumStrict (int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)

  Send decimal number to LCD with a strict length.
- void LCDSendNum (int64_t num, uint8_t length, uint8_t isSigned, uint8_t showZeros)

  Send decimal number to LCD with a variable length.
- void LCDSendNumArray (uint8_t ∗index)

Sends a number in a array to the LCD.
- void LCDClear (void)

  Clears the LCD.
- void LCDDisplayOn (uint8_t onOff)

  Configures LCD appearence option.
- void LCDSendHex (uint8_t ∗array)

  Prints a decimal number in hexadecimal format.
- void numToArray (int32_t num, uint8_t ∗array, uint8_t length, uint16_t base)

  Converts a variable to an array of numbers.
- void LCDRegisterSpecial (uint8_t number, uint8_t ∗character)

  Registers special characters in the LCD.
- void LCDShift (uint8_t shift)

  Enables the data shift option in LCD.
- void LCDHome (void)

  Sends the LCD cursor to home position.
- void arrayToNum (uint8_t ∗array, uint32_t ∗num, uint8_t base)

  Converts a number array to a variable.
- void LCDSendVU (uint32_t num, uint32_t base)

  Send LCD VU level.

Variables

- const char LCD_CmdInit_Vector [lcd_vector_index] ={0x38, 0x38, 0x38, 0x01, LCD_DISPLAY_CONFIG, L-CD_DISPLAY_INCREMENT, 0x01}
- const unsigned int LCD_InitDelay_Vector [lcd_vector_index] ={8000, 200, 200, 16000, 600, 200, 15000}
- LCDStatus LCD0Status

  LCD struct for current position.

LCD_Splahscreen

LCD splashscreen text configuration.

- #define LCD_splashscreen_row1 __DATE__

  compile date, used as program version
- #define LCD_splashscreen_row2 __TIME__

  compile time, used as program version
- #define LCD_splashscreen2_row1 PROJECT_NAME

  geneartion of project name in LCD
- #define LCD_splashscreen2_row2 ("rnm sys undvpd")

  creator's watermark

LCD_Function_Masks

Function masks used in the lib.

- #define LCDDelay(x) SysDelayUs(x)

  delay loop mask.
- #define LCDPinSet(x, y) PinAddrSet(x, y)

  pin set function mask.
- #define LCDPinClear(x, y) PinAddrClear(x,y)

pin clear function mask.

- #define LCD_DTA_Send(text) ShiftSerialSend(LCD_DTA_Port, LCD_DTA_Pin, LCD_CLK_Port, LCD_CLK-_Pin, text)

  lcd send function mask.
- #define LCD_RS_High LCDPinSet(LCD_RS_Port, LCD_RS_Pin)

  RS pin set mask.
- #define LCD_RS_Low LCDPinClear(LCD_RS_Port, LCD_RS_Pin)

  RS pin clear mask.
- #define LCD_EN_High LCDPinSet(LCD_EN_Port, LCD_EN_Pin)

  EN pin set mask.
- #define LCD_EN_Low LCDPinClear(LCD_EN_Port, LCD_EN_Pin)

  EN pin clear mask.

LCD_Option_Flags

LCD options flags.

- #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

LCD_Options_Select

Defitions of initial LCD options configuration.

- #define **LCD_DISPLAY_CONFIG** (LCD_DISPLAY_ON|LCD_CURSOR_OFF|LCD_BLINK_OFF)
- #define **LCD_DISPLAY_INCREMENT** (LCD_INCREMENT|LCD_INCREMENT_NO_SHIFT)

4.1.1 Detailed Description

This file contains the functions to properly (hopefully) drive the LCD peripheral. Current implementation uses a serial shift register to drive the LCD in 8 bit mode.

Further modifications to this library include:

- Parallel data transfer (make function to).

For the complete execution of the library, the following macros have to be created specifically for this library:

- LCD_SPLASHSCREEN1
  (1 or 0)
- LCD_SPLASHSCREEN2
  (1 or 0)
- LCD_SPLASHSCREEN_CLEAR
  (1 or 0)

The library uses theses global value macros:

- PROJECT_NAME
  ("string_of_name_here")

The library uses these global function macros:

- SysDelayUs(time)
  CPU delay loop, usually.
- PinAddrSet(port, pin)
  Direct address write.
- PinAddrClear(port, pin)
  Direct address write.
- ShiftSerialSend(DTA_PORT, DTA_PIN, CLK_PORT, CLK_PIN, text)
  Used by LCD_DTA_Send(text) function, determines how data is transfered.

The following external connection macros have to be set:

- LCD_RS_PORT
  (register address)
- LCD_RS_PIN
  (bit position)
- LCD_EN_PORT
  (register address)
- LCD_EN_PORT
  (bit position)
- LCD_DTA_PORT
  (register address)
- LCD_DTA_PIN
  (bit position)
- LCD_CLK_PIN
  (bit position)

4.1.2 Function Documentation

4.1.2.1 void arrayToNum (uint8_t ∗ *array,* uint32_t ∗ *num,* uint8_t *base*)

Parameters

∗*array*	uint8_t source array.
∗*num*	uint32_t target variable.
base	uint8_t base of digits in array.

Definition at line 477 of file lcd.c.

```
{
    while(*array<33)
    {
        *num += *array * base;
        array++;
    }
}
```

4.1.2.2 void LCDClear (void)

Sends a clear all command to the LCD.

Returns

None.

Definition at line 304 of file lcd.c.

```
{
    LCDSendCmd(0x01);
    LCD0Status.row=1;
    LCD0Status.col=1;
    LCDDelay(800);
}
```

4.1.2.3 void LCDDisplayOn (uint8_t *onOff*)

Sends the logical combination of the following flags:

- LCD_DISPLAY_ON/OFF
- LCD_CURSOR_ON/OFF
- LCD_BLINK_ON/OFF

Returns

None.

Definition at line 324 of file lcd.c.

```
{
    LCD0Status.display = onOff;
    LCDSendCmd(onOff);
}
```

4.1.2.4 void LCDHome (void)

Returns

None.

Definition at line 463 of file lcd.c.

```
{
    LCDSendCmd(0x02);
    LCDDelay(1500);
}
```

4.1.2.5 void LCDInit (void)

Called once at startup. Takes no parameters.

Returns

None.

Definition at line 35 of file lcd.c.

```
{
    uint8_t Vector_Scan = 0;
    LCDDelay(15000);
    for(Vector_Scan=0; Vector_Scan < lcd_vector_index; Vector_Scan++)
    {
        LCDSendCmd(LCD_CmdInit_Vector[Vector_Scan]);
        LCDDelay(LCD_InitDelay_Vector[Vector_Scan]);
    }
    //splash screen
#if LCD_SPLASHSCREEN1 == 1
    LCDPosition(1,1);
    LCDSendString(LCD_splashscreen_row1, 0);
    LCDPosition(2,1);
    LCDSendString(LCD_splashscreen_row2, 0);
    LCDDelay(2*1000*1000);
#endif
#if LCD_SPLASHSCREEN2 == 1
    LCDPosition(1,1);
    LCDSendString(LCD_splashscreen2_row1, false);
    LCDPosition(2,1);
    LCDSendString(LCD_splashscreen2_row2, false);
    LCDDelay(2*1000*1000);
#endif
#if LCD_SPLAHSCREEN_CLEAR == 1
    LCDClear();
#endif
}
```

4.1.2.6 void LCDPosition (uint8_t *row*, uint8_t *col*)

Parameters

row	uint8_t row.
col	uint8_t column.

Returns

None.

Definition at line 116 of file lcd.c.

```
{
    LCD0Status.row = row;
    LCD0Status.col = col;
    col--;
    if(row==1)
        row = 0x80;
    if(row==2)
        row = 0xC0;
    LCDSendCmd(row+col);
    LCDDelay(20);
}
```

4.1.2.7 void LCDPositionNoDelay (uint8_t *row*, uint8_t *col*)

Sends the LCD position command but does not implement a delay after.

Parameters

row	uint8_t row.
col	uint8_t column.

Returns

None.

Definition at line 140 of file lcd.c.

```
{
    LCD0Status.row = row;
    LCD0Status.col = col;
    col--;
    if(row==1)
        row = 0x80;
    if(row==2)
        row = 0xC0;
    LCDSendCmd(row+col);
}
```

4.1.2.8 void LCDRegisterSpecial (uint8_t *number,* uint8_t ∗ *character*)

Parameters

number	uint8_t number, from 1 to 8, of special character to be transferred.
∗*character*	uint8_t array containing the bits of the character to be set.

Returns

None.

Definition at line 421 of file lcd.c.

```
{
    uint8_t scan=0, data=0;
    LCDSendCmd(0x40+(number<<3));
    do
    {
        data = *(character+scan);
        LCDDelay(640);
        LCDSendChar(data&0x1F);
        scan++;
    }
    while(scan<8);
    LCDDelay(320);
}
```

4.1.2.9 void LCDSend (uint8_t *send*)

Parameters

send	uint8_t data to be sent.

Returns

None.

Todo make function to send parallel data.

Definition at line 97 of file lcd.c.

```
{
    LCD_EN_Low;
    LCD_DTA_Send(send);
    LCDDelay(4);
    LCD_EN_High;
    LCDDelay(4);
    LCD_EN_Low;
}
```

4.1.2.10 void LCDSendChar (uint8_t *txt*)

Parameters

txt	uint8_t type data to be sent, 8 bits.

Returns

None.

Definition at line 82 of file lcd.c.

```
{
    LCD_RS_High;
    LCDSend(txt);
    LCD0Status.col ++;
    LCD_RS_Low;
}
```

4.1.2.11 void LCDSendCmd (uint8_t *cmd*)

Returns

None.

Definition at line 68 of file lcd.c.

```
{
    LCD_RS_Low;
    LCDSend(cmd);
}
```

4.1.2.12 void LCDSendHex (uint8_t ∗ *array*)

Parameters

∗*array*	uint8_t number array to be written.

Returns

None.

Definition at line 338 of file lcd.c.

```
{
    uint8_t offset, temp;
    LCDSendChar('0');
    LCDSendChar('x');
    array += 2;
    while(*array<=32)
    {
        temp = *array;
        if(temp>9)
        {
            temp -= 10;
            offset = 'A';
        }
        else
            offset = '0';
        LCDSendChar(temp+offset);
        array++;
    }
}
```

4.1.2.13 void LCDSendNum (int64_t *num*, uint8_t *length*, uint8_t *isSigned*, uint8_t *showZeros*)

Writes a decimal number with a variable length in the LCD.

Parameters

num	int64_t number to be written.
length	uint8_t length, in decimal digits, of the number.
isSigned	uint8_t flag to determine if the number is to be treted as a negative number.
showZeros	uint8_t flag to determine if leading zeros will be shown.

Returns

None.

Definition at line 246 of file lcd.c.

```
{
    uint8_t index =0;
    uint8_t out = ' ';
    uint64_t multiple = 1;
    limitCeilValue(length,10);
    if(num<0 && isSigned==true)
    {
        out = '-';
        num *= -1;
    }
    LCDSendChar(out);
    index = length;
    while(length>1)
    {
        multiple *= 10;
        length--;
    }
    while(index >= 1)
    {
        out = (uint32_t) (num/multiple);
        num -= out*(multiple);
        if(out!=0)
            showZeros = true;
        if(out==0 && showZeros==false)
            out -= 16;
        LCDSendChar(out+48);
        multiple /= 10;
        index--;
    }
}
```

4.1.2.14 void LCDSendNumArray (uint8_t ∗ *index*)

Sends a number arranged in an array to the LCD. Each cell corresponds to a digit in the LCD.

Parameters

∗*index*	uint8_t base address of the array to be written.

Returns

None.

Definition at line 288 of file lcd.c.

```
{
    while(*index<33)
    {
        LCDSendChar(*index+'0');
        index++;
    }
}
```

4.1.2.15 void LCDSendNumStrict (int64_t *num,* uint8_t *length,* uint8_t *isSigned,* uint8_t *showZeros*)

Writes a decimal number with a strict length in the LCD.

Parameters

num	int64_t number to be written.
length	uint8_t length, in decimal digits, of the number.
isSigned	uint8_t flag to determine if the number is to be treted as a negative number.
showZeros	uint8_t flag to determine if leading zeros will be shown.

Returns

None.

Definition at line 200 of file lcd.c.

```
{
    uint8_t index =0;
    uint8_t out;
    uint64_t multiple = 1;
    limitCeilValue(length,10);
    if(num<0 && isSigned==true)
    {
        LCDSendChar('-');
        num *= -1;
        length--;
    }
    index = length;
    while(length>1)
    {
        multiple *= 10;
        length--;
    }
    while(index >= 1)
    {
        out = (uint32_t) (num/multiple);
        num -= out*(multiple);
        if(out!=0)
            showZeros = true;
        if(out==0 && showZeros==false)
            out -= 16;
        LCDSendChar(out+48);
        multiple /= 10;
        index--;
    }
}
```

4.1.2.16 void LCDSendString (uint8_t * *string,* uint8_t *breakLine*)

Parameters

∗string	uint8_t string to be sent.
breakLine	uint8_t break line at the end of LCD length.

Returns

None.

Definition at line 164 of file lcd.c.

```
{
    while(*string)
    {
        LCDSendChar(*string);
        string++;
        if(LCD0Status.col==LCD_col_num && breakLine==true)
        {
            if(LCD0Status.row<=LCD_row_num)
                LCDPosition(LCD0Status.row+1, 1);
            else
                LCDPosition(0, 1);
        }
    }
}
```

4.1.2.17 void LCDSendVU (uint32_t *num,* uint32_t *base*)

This function uses special characters filled in horizontal increasing steps to make a Visual Units Display from the LCD.

Parameters

num	uint32_t number to be written.
base	uint32_t max number value.

Returns

None.

Definition at line 498 of file lcd.c.

```
{
    uint8_t index, pass=1;
    num  = (unsigned int) num*(LCD_col_num*LCD_char_width)/base;
    while(num>0)
    {
        index = LCD_char_width;
        while(num<LCD_char_width)
        {
            index--;
            num++;
        }
        LCDSendChar(index);
        num -= LCD_char_width;
        pass++;
    }
    while(pass<=LCD_col_num)
    {
        pass++;
        LCDSendChar(0);
    }
}
```

4.1.2.18 void LCDShift (uint8_t *shift*)

Sets the configuration according to the following flags:

- LCD_SHIFT
- LCD_SHIFT_DISPLAY
- LCD_SHIFT_CURSOR
- LCD_SHIFT_RIGHT
- LCD_SHIFT_LEFT

Parameters

shift	uint8_t option flag to be set.

Returns

None.

Definition at line 452 of file lcd.c.

```
{
    LCDSendCmd(shift|LCD_SHIFT);
}
```

4.1.2.19 void numToArray (int32_t *num*, uint8_t ∗ *array*, uint8_t *length*, uint16_t *base*)

The termination of array is done by the number 33. The highest selectable number base is 32.

Parameters

num	int32_t number to be converted.
∗*array*	uint8_t destination array.
length	uint8_t number length, in decimal digits.
base	uint8_t base of output array.

Returns

None.

Definition at line 373 of file lcd.c.

```
{
    uint16_t index =1;
    uint8_t out;
    uint64_t multiple = 1;

    limitCeilValue(length, (unsigned char) 1<<64/base);
    limitCeilValue(length, maxLengthOut);


    //create multiple number
    while(index<length)
    {
        multiple *= base;
        index++;
    }
    //sort multiples
    while(index >= 1)
    {
        //determines the multiple
        out = (uint8_t) (num/multiple);
        //takes out multiple
        num -= out*(multiple);

        //escreve no vetor, desloca indice
        *array = out;
        array++;
        multiple /= base;
        //change multiple position
        index--;
    }
    *array = 33;
}
```

4.1.3 Variable Documentation

4.1.3.1 LCDStatus LCD0Status

Todo create masks for LCD commands.

4.1.3.2 const char LCD_CmdInit_Vector[lcd_vector_index] ={0x38, 0x38, 0x38, 0x01, LCD_DISPLAY_CONFIG, LCD_DISPLAY_INCREMENT, 0x01}

Initialization Commands Sequence:

Definition at line 20 of file lcd.c.

4.1.3.3 const unsigned int LCD_InitDelay_Vector[lcd_vector_index] ={8000, 200, 200, 16000, 600, 200, 15000}

LCD Init command delay vector, in uS.

Definition at line 25 of file lcd.c.

Chapter 5

Data Structure Documentation

5.1 CommandInstance Struct Reference

Data Fields

- uint8_t **charIn**
- uint8_t **cmdBuffer** [MAX_BUFFER_SIZE]
- uint16_t **charOut** [MAX_BUFFER_SIZE]
- uint8_t **charOutPtr**

5.1.1 Detailed Description

Definition at line 15 of file cmd_sort.h.

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

- my_lib/cmd_sort.h

5.2 IRInstance Struct Reference

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**

5.2.1 Detailed Description

Definition at line 83 of file ir.h.

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

- my_lib/ir.h

5.3 LCDStatus Struct Reference

Data Fields

- uint8_t **row**
- uint8_t **col**
- uint8_t **display**
- uint8_t **shift**
- uint8_t **cgramAdress**
- uint8_t **specialChar** [8]

5.3.1 Detailed Description

Definition at line 135 of file lcd.h.

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

- my_lib/lcd.h

5.4 UARTInstance Struct Reference

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**

5.4.1 Detailed Description

Definition at line 23 of file myUart.h.

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

- my_lib/myUart.h

Index