SHIELD-LCD16×2 custom firmware note

1-07-16 OLIMEX LTD

SHIELD-LCD16×2 is a two-row display compatible with the shield layout of Arduino and Arduino-like boards.

It is equipped with a PIC16 microcontroller that holds custom firmware. The firmware makes sending and receiving commands to the display easier – it implements I2C and UART communication to the display and the rest of the board's features (buttons, LEDs, etc). There are also several commands implemented and they are discussed in this document.

The sources of the firmware are also published on the product's page. You can freely use the sources as a template or a foundation to building an even more sophisticated firmware.

A good way to further comprehend the firmware and the command usage would be inspecting either the firmware source code or the examples available. These are also available at the product's web page.

1. I2C commands

All available i2c commands are using standard protocol format. You should use only 100kHz bus. The communcation with the shield might not be possible at higher clock speeds!

1.1 SET_TRIS (0×01)

This command sets port direction of all available pins. These are GPIO1 to GPIO9. To make GPIO output you should write 0 as value. For input – 1.

S	ADDR	W	ACK	0×01	ACK	GPIO#	ACK	VALUE	ACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read
GPIO#:
1 → GPIO1
2 → GPIO2
3 → GPIO3
4 → GPIO4
5 → GPIO5
6 → GPIO6
7 → GPIO7
8 → GPIO8
9 → GPIO9
VALUE:
1 → INPUT
0 → OUTPUT

1.2 SET_LAT (0×02)

If a pin is configured as output you can set its level using this command. Valid values as 0 for low level and 1 – for high. They correspond to 0V and 3.3V.

S	ADDR	W	ACK	0×02	ACK	GPIO#	ACK	VALUE	ACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read
GPIO#:
1 → GPIO1
2 → GPIO2
3 → GPIO3
4 → GPIO4
5 → GPIO5
6 → GPIO6
7 → GPIO7
8 → GPIO8
9 → GPIO9

VALUE:
1 → HIGH (3.3V)
0 → LOW (0V)

1.3 GET_PORT (0×03)

If GPIO is configured as input you can periodically read its state with this command. Like in the previous command 1 corresponds to high level (3.3V) and 0 – low (0V).

S	ADDR	W	ACK	0×03	ACK	GPIO#	ACK	P	S	ADDR	R	ACK	VAL	NACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read
R:
Write/read flag:
0 → write
1 → read
GPIO#:
1 → GPIO1
2 → GPIO2
3 → GPIO3
4 → GPIO4
5 → GPIO5
6 → GPIO6
7 → GPIO7
8 → GPIO8
9 → GPIO9

VALUE:
1 → HIGH (3.3V)
0 → LOW (0V)

1.4 GET_BUT (0×05)

The state of the four buttons can be read all the same time using one command. The four LSB of the returned value represent the state of the buttons. For example when the comm returns 0×05 this means:

0×05 → 0b00000101

This means:
button1 → 1 → OFF
button2 → 0 → ON
button3 → 1 → OFF
button4 → 0 → ON

Notice that buttons default state is 1, so when the value is 0 the button is pressed.

S	ADDR	W	ACK	0×05	ACK	P	S	ADDR	R	ACK	VAL	NACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read
R:
Write/read flag:
0 → write
1 → read

1.5 GET_ID (0×20)

You can verify that device is present using GET_ID command. The returned value should be compared to the default one (0×65).

S	ADDR	W	ACK	0×20	ACK	P	S	ADDR	R	ACK	VAL	NACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read
R:
Write/read flag:
0 → write
1 → read

1.6 GET_FRM (0×21)

Same as above to check firmware version.

S	ADDR	W	ACK	0×21	ACK	P	S	ADDR	R	ACK	VAL	NACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write

1 → read

R:

Write/read flag:
0 → write
1 → read

1.7 LCD_CLR (0×60)

Clears everything that is displayed on the lcd.

S	ADDR	W	ACK	0×60	ACK	P

ADDR:

I2C slave address → 0×30 (default)

W:

Write/read flag:
0 → write
1 → read

1.8 LCD_WR (0×61)

Writes one character at time at position Y, X.

X must be between 0 and 15. Y must be between 0 and 1.

S	ADDR	W	ACK	0×61	ACK	Y	ACK	X	ACK	CHAR	ACK	P

ADDR:

I2C slave address → 0×30 (default)

W:

Write/read flag:
0 → write
1 → read

Y:

Y coordinate → 0 – 1

X:

X coordinate → 0 – 15

CHAR:

Hex code of displayed character

1.9 SET_BL (0×62)

This is used to turn on and off the backlight of LCD. It can vary from 0 to 255.

S	ADDR	W	ACK	0×62	ACK	VALUE	ACK	P

ADDR:

I2C slave address → 0×30 (default)

W:

Write/read flag:
0 → write
1 → read

VALUE:

Blacklisting level → from 0 to 255

1.10 UART_EN (0×10)

Enables UART communication to the module.

S	ADDR	W	ACK	0×10	ACK	VALUE	ACK	P

ADDR:
I2C slave address → 0×30 (default)
W:
Write/read flag:
0 → write
1 → read

VALUE:
0 → UART disable
1 → UART enable

2. UART commands

UART needs to be enabled with UART_EN command first!

When using UART, baud-rate should be set at 9600. All commands should end with '/r' and '/n'.

1. TRIS:g:v

Same as SET_TRIS:
g → GPIO number – 1 to 9
v → value

2. LAT:g:v

Same as SET_LAT:
g → GPIO number – 1 to 9
v → value

3. PORT:g

Same as GET_PORT:
g → GPIO number – 1 to 9

4. BUT

Same as GET_BUT.

5. GOTOXY:x:y

Set cursor to given X and Y coordinate.

6. BLKL:v

Same as SET_BL
v → value – 0 to 255

7. WR:v

Write characters to lcd.

8. CLEAR

Clear screen.