

MOD-TC-MK2

1. Description

Built with MPLAB X 1.30 with HI-TECH PICC compiler version 9.83 for the slave devices.

This demo shows how to control MOD-TC-MK2 boards via I2C communication protocol. The baud-rate of the communication is standard 100kHz. There are several built-in commands:

Command	Value	Parameters	Comment																																															
SET_TRIS	0x01	0bXDDDDDDD	Set direction of the GPIOs.																																															
			<table><tr><td>X</td><td>GPIO_6</td><td>GPIO_5</td><td>GPIO_4</td><td>GPIO_3</td><td>GPIO_2</td><td>GPIO_1</td><td>GPIO_0</td></tr><tr><td>-</td><td>0/1</td><td>0/1</td><td>0/1</td><td>1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table>	X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0	-	0/1	0/1	0/1	1	0/1	0/1	0/1																															
			X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0																																								
			-	0/1	0/1	0/1	1	0/1	0/1	0/1																																								
Where: 0 - output 1 - input																																																		
SET_LAT	0x02	0bXDDDDDDD	Set output state of the GPIOs.																																															
			<table><tr><td>X</td><td>GPIO_6</td><td>GPIO_5</td><td>GPIO_4</td><td>GPIO_3</td><td>GPIO_2</td><td>GPIO_1</td><td>GPIO_0</td></tr><tr><td>-</td><td>0/1</td><td>0/1</td><td>0/1</td><td>-</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table>	X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0	-	0/1	0/1	0/1	-	0/1	0/1	0/1																															
			X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0																																								
			-	0/1	0/1	0/1	-	0/1	0/1	0/1																																								
Where: 0 - low(0V) 1 - high(3.3V)																																																		
GET_PORT	0x03	-	Read input state of the GPIOs.																																															
			<table><tr><td>X</td><td>GPIO_6</td><td>GPIO_5</td><td>GPIO_4</td><td>GPIO_3</td><td>GPIO_2</td><td>GPIO_1</td><td>GPIO_0</td></tr><tr><td>-</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table>	X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0	-	0/1	0/1	0/1	0/1	0/1	0/1	0/1																															
			X	GPIO_6	GPIO_5	GPIO_4	GPIO_3	GPIO_2	GPIO_1	GPIO_0																																								
			-	0/1	0/1	0/1	0/1	0/1	0/1	0/1																																								
Where: 0 - low(0V) 1 - high(3.3V)																																																		
SET_PU	0x04	0bXXXDDDDD	Set pull-up ressiostors of the GPIOs. NOTE: Only PORTA can use pullups.																																															
			<table><tr><td>X</td><td>X</td><td>X</td><td>GPIO_5</td><td>GPIO_3</td><td>GPIO_2</td><td>GPIO_1</td><td>GPIO_0</td></tr><tr><td>-</td><td>-</td><td>-</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table>	X	X	X	GPIO_5	GPIO_3	GPIO_2	GPIO_1	GPIO_0	-	-	-	0/1	0/1	0/1	0/1	0/1																															
			X	X	X	GPIO_5	GPIO_3	GPIO_2	GPIO_1	GPIO_0																																								
			-	-	-	0/1	0/1	0/1	0/1	0/1																																								
Where: 0 - Pullup is disabled 1 - Pullup is enabled																																																		
GET_AN0	0x10	-	Get the analog value of AN0.																																															
			<table><tr><td>D7</td><td>D6</td><td>D5</td><td>D4</td><td>D3</td><td>D2</td><td>D1</td><td>D0</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>D9</td><td>D8</td></tr><tr><td colspan="8">BYTE 0</td><td colspan="8">BYTE 1</td></tr><tr><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td><td>0/1</td></tr></table>	D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X	X	D9	D8	BYTE 0								BYTE 1								0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			D7	D6	D5	D4	D3	D2	D1	D0	X	X	X	X	X	X	D9	D8																																
			BYTE 0								BYTE 1																																							
0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1																																				
To read data: ((BYTE1)*256 + BYTE0)*(3.3/1023))V																																																		
GET_AN1	0x11	-	Get the analog value of AN1.																																															
GET_AN2	0x12	-	Get the analog value of AN2.																																															
GET_AN3	0x16	-	GGet the analog value of AN6.																																															

GET_AN7	0x17	-	Get the analog value of AN7.	
GET_TEMP	0x21	-	Read temperature from termocouple.	
			D31 D30 D29 D28 D27 D26 D25 D24 D23 D22 D21 D20 D19 D18 D17 D16	
			BYTE 3	BYTE 2
			0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1	
			D15 D14 D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0	
			BYTE 1	BYTE 0
			0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1 0/1	
			To convert this temperature read MAX31855 datasheet.	
SET_ADDR	0xB0	0bDDDDDDDD	Set new address of the board.	

IMPORTANT NOTE: By default the address of the board is 0xA0. If this is the only device - you could set new address without the use of the PROG jumper. If there are multiple devices on same address put the jumper and reset the board. This will set the address to 0xF0. After that set the new address and remove the jumper.

The I2C communication is:

START	ADDRESS 0			ADDRESS 1		ADDRESS 2		COMMAND		DATA		STOP
1bit	7bit	R/W	ACK	8bit	ACK	8bit	ACK	8bit	ACK	8bit	ACK	1bit
WRITING DATA: for example with SET_TRIS, SET_LAT, SET_PU, SET_ADDR												

S	ADDRESS 0			ADDRESS 1		ADDRESS 2		COMMAND		P	S	ADDRESS 0			DATA		P
1bit	7bit	R/W	ACK	8bit	ACK	8bit	ACK	8bit	ACK	1bit	1bit	7bit	R/W	ACK	8bit	ACK	1bit
READING DATA: for example with GET_PORT, GET_ANx and GET_TEMP																	

Where

- **Address 0** – the address of all Olimex I2C devices - **0x48**
- **Address 1** – the address of all MOD-TC-MK2 devices – **0x01**
- **Address 2** – the individual address of specific MOD-TC-MK2 device.

For example you could control the device with OLinuXino MAXI and **i2c-tool**.

First you must compile the i2c-tool:

```
# gcc i2c-tool.c -o i2c-tool
```

If you want to read temperature use the following commands:

```
# ./i2c-tool -w 0 0x48 3 0x02 0xA0 0x21
```

```
# ./i2c-tool -r 4
```

or

```
# ./i2c-tool -w 0 0x48 3 0x02 0xA0 0x10  
# ./i2c-tool -r 2
```

to read AN0

NOTE: In all examples the device address is 0xA0, which is the default value.

2. Support - <https://www.olimex.com/>

3. Release Notes - 12 July 2012 – Initial release
 29 September 2012 – Updated for the new firmware version
 27 May 2013 – Updated command information