

# MOD-RS485-ISO

## 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 several MOD-RS485-ISO 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_CLC	0x30	0bXXXXXXDD	Set TX and RX state, TX=1 – the channel is open, if 0 – closed.
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 protocol is the standard one:

START		ADDRESS 0		ADDRESS 1		ADDRESS 2		COMMAND		DATA		STOP
1bit	7bit	R/W	ACK	8bit	ACK	8bit	ACK	8bit	ACK	8bit	ACK	1bit
<b>WRITING DATA:</b> for example with SET_CLC and SET_ADDR												

Where

- **Address 0** – the address of all Olimex I2C devices - **0x48**
- **Address 1** – the address of all MOD-RS485-ISO devices – **0x04**
- **Address 2** – the individual address of specific MOD-RS485-ISO 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 enable both TX and RX use:

```
# ./i2c-tool -w 0 0x48 4 0x04 0xA0 0x30 0x03
```

**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  
04 February 2013 – Modified command table