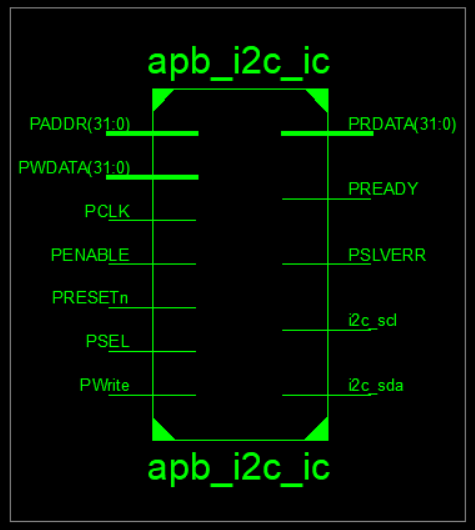
**APB TO I2C BRIDGE IC**

**Features**

1. Supports up to 32-bit wide data width
2. Half- duplex transmission
3. Operating frequencies – 100KHz, 400KHz, 1MHz, 3.3MHz
4. 7-bit slave address, up to 127 slaves
5. Repeated start available

**Pin description**



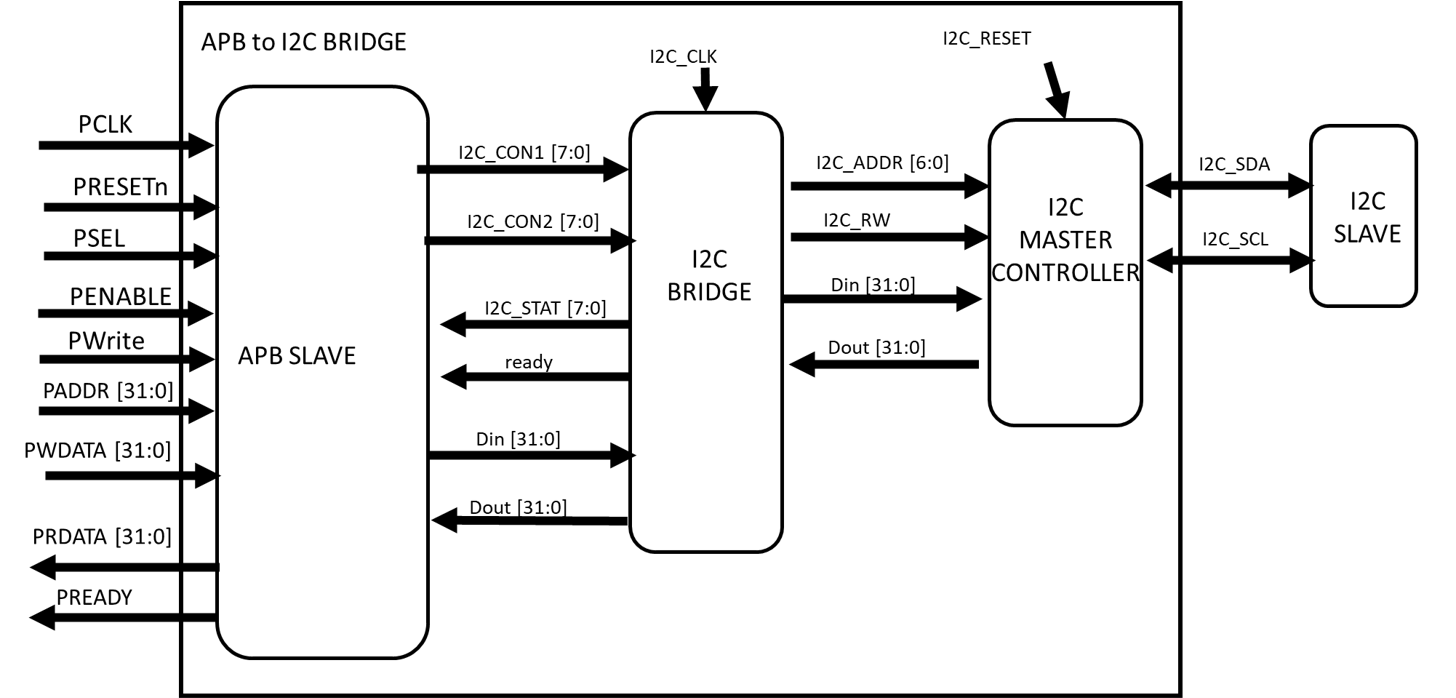
**Input pin’s**

* PRESETn- Active High, synchronous RESET
* PCLK-Peripheral Clock (100 MHz)
* PSEL- Peripheral select for APB\_I2C\_IC
* PENABLE- Data Transmission ENABLE
* PADDR-Configuration Bits (32-bit wide)
* PWrite-Enable Read/ Write Operation (1 -Write, 0-Read)
* PWDATA-Data Input (32-bit wide)

**Output pin’s**

* PREADY-APB Slave Output to indicate that APB-Slave is ready for next transaction (to SoC)
* PRDATA-APB Slave data Output (32-bit wide) (to SoC)
* PSLVERR- APB Slave Output to indicate error in transaction (to SoC)
* I2c\_scl – Serial Clock Line for I2C devices, bidirectional, pull-up required (to I2C bus)
* I2c\_sda- Serial Data Line for I2C devices bidirectional, pull-up required (to I2C bus)

**Note:** All pins are 1-bit wide, unless mentioned.



**Description**

The I2C is a protocol which helps to communicate the external devices (RTC, temperature, voltage regulator etc.), sensors, data storage devices (EPROMs).

The I2C protocol half- duplex, 2-wire simple protocol.

**Register Description**

**I2C Control Registers:**

**I2C con1**: This register controls the transaction parameters of the I2C master. It is 8-bit length [7:0].

I2C con1[0]- it sets or resets the master

I2C con1[1]- it enables the master controller for clock, for clock gating and power saving

I2C con1[2:3]- byte-count, indicates transaction length-8,16,24,32 bit

I2C con1[4]- it indicates that the data send was a data, or an address location

I2C con1[5]- it is set a transaction with repeated start (for page reading in EEPROMs)

I2C con1[6:7]- it sets the frequency of the Serial clock line, 2’b00-100KHz, 2’b01- 400KHz, 2’b10- 1MHz, 2’b11-3.3MHz.

**I2C con2:** This register has the slave address and the read/write bit. It is 8 bit length [7:0]

I2C con2[0]- read/write bit

I2C con2[1:6]- 7-bit slave address.

**I2C Status Register:**

**I2C stat:** This register indicates the status of the I2C master. It is 8-bit length [7:0].

I2C con1[0]- it indicates if the master is busy in a transaction

I2C con1[1]- it indicates whether the master is in a read cycle or write cycle.

I2C con1[2:3]- byte-count, indicates the transaction length-8,16,24,32 bit.

I2C con1[4]- it indicates that whether the master is in start state.

I2C con1[5]- it indicates whether the master is in data sending or receiving state.

I2C con1[6]- it indicates whether the master has received a the acknowledge or not.

I2C con1[7]- it indicates that whether the master is in stop state

**Protocol operation**

The pin configuration as follows to perform read/write operations:

* **To check status of I2C-master:**

**APB-read to read status of I2C master:** Useful for knowing the status of I2C-master, because I2C is very slow protocol compared to APB, hence knowing its status before issuing a transaction is essential, because it prevents over-writing. I2C status registers indicate whether the I2C master is in transaction, along with the information of transaction parameter, where is it in the transaction timeline.

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

PWrite = 0;

PADDR = 32'h0000\_0000; // ZERO in all 32-bit indicates the data is sent/received to/from I2C-control and status registers.

PRDATA <= 0000\_0000\_{I2C-STAT}\_{I2C-CON2}\_{I2C-CON2};

* **To write data to I2C-slave device:**

**APB-write to configure I2C master:** To configure the I2C control registers, the parameters are set using PWDATA, I2C-control are selected using PADDR.

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

PWrite = 1;

PADDR = 32'h0000\_0000; // ZERO in all 32-bit indicates the data is sent/received to/from I2C-control and status registers.

//con2 con1

|addr 0-6| |r/w| |ff R D/A cc e r|

PWDATA = 32'b0000\_0000 1100\_011 0 11\_0\_\_1\_\_00\_1\_1; // PWDATA[0:7] -> CON1

// PWDATA[8:15] -> CON2

**APB-write to send data to I2C master:** To send data using PWDATA, Din is selected using PADDR.

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

PWrite = 1;

PADDR = 32'hffff\_0000; // NON-ZERO value indicates the data is sent/received to/from Din and Dout registers.

PWDATA = 32'habcd\_ef12; // PWDATA[0:31] -> Din[0:31]

* **To read data form I2C-Slave devices:**

**APB-write to configure I2C master:** To configure the I2C control registers, the parameters are set using PWDATA, I2C-control are selected using PADDR.

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

PWrite = 1;

PADDR = 32'h0000\_0000; // ZERO in all 32-bit indicates the data is sent/received to/from I2C-control and status registers.

//con2 con1

|addr 0-6| |r/w| |ff R D/A cc e r|

PWDATA = 32'b0000\_0000 1100\_011 1 11\_0\_\_1\_\_00\_1\_1; // PWDATA[0:7] -> CON1

// PWDATA[8:15] -> CON2

**APB-read to read status of read transaction:** for knowing the status of read, because I2C is very slow protocol compared to APB, hence knowing its status before reading the data from APB-slave is essential. If I2C status registers indicate that the reading form I2C-Slave is completed then a data read cycle is issued (APB data read).

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

PWrite = 0;

PADDR = 32'h0000\_0000; // ZERO in all 32-bit indicates the data is sent/received to/from I2C-control and status registers.

PRDATA <= 0000\_0000\_{I2C-STAT}\_{I2C-CON2}\_{I2C-CON2};

**Note:** Time delay to issue the APB-read for data read is predicted by the status of I2C-master in transaction time-line, frequency of operation, transaction length.

**APB-read to read data from I2C master:** To read data using PRDATA, Dout is selected using PADDR. Done after a time delay.

PRESETn = 1;

PSEL = 1;

PENABLE = 1;

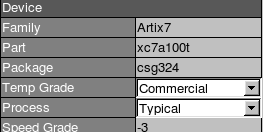
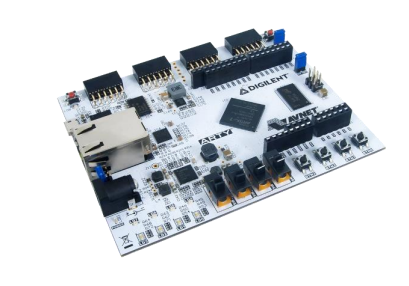
PWrite = 1;

PADDR = 32'hffff\_0000; // NON-ZERO value indicates the data is sent/received to/from Din and Dout registers.

PRDATA[0:31] <= Dout[0:31] ;

**Tested board:**

I2C protocol tested using artix-7 board with following chip device.

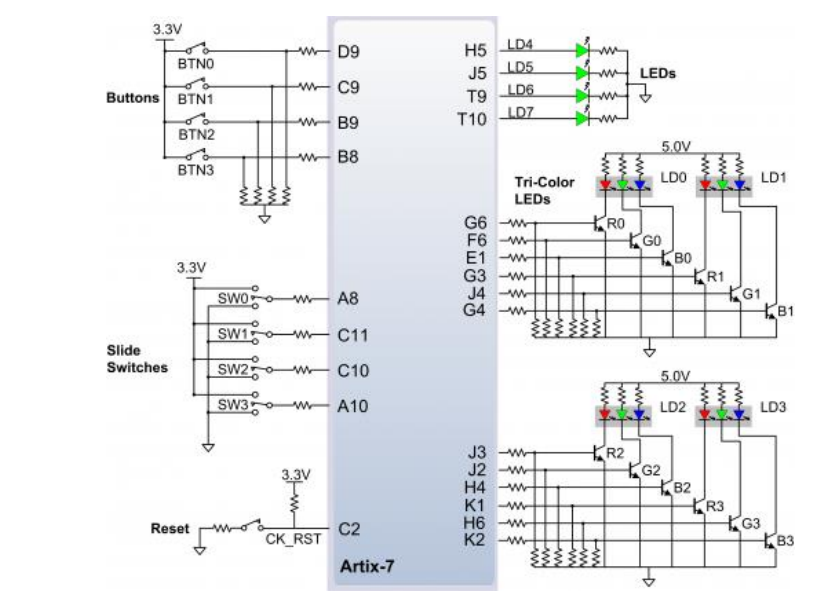
 

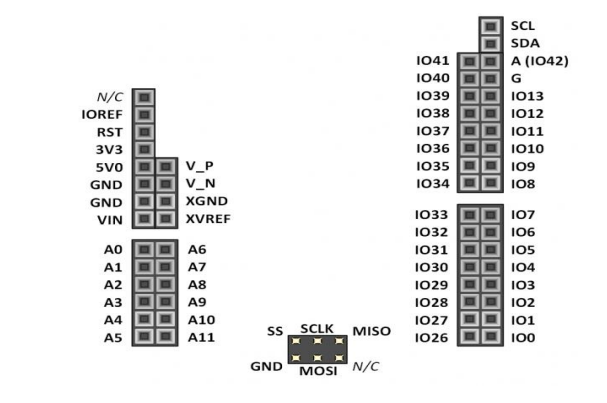
**Pin Description**

For FPGA (Arty-7) implementation on board toggle switches are used to simulate signals from APB-bus. Single color LEDs for address, PREADY. RGB LEDs for 8-bit DATA register. Buttons for indexing of 32-bit DATA.

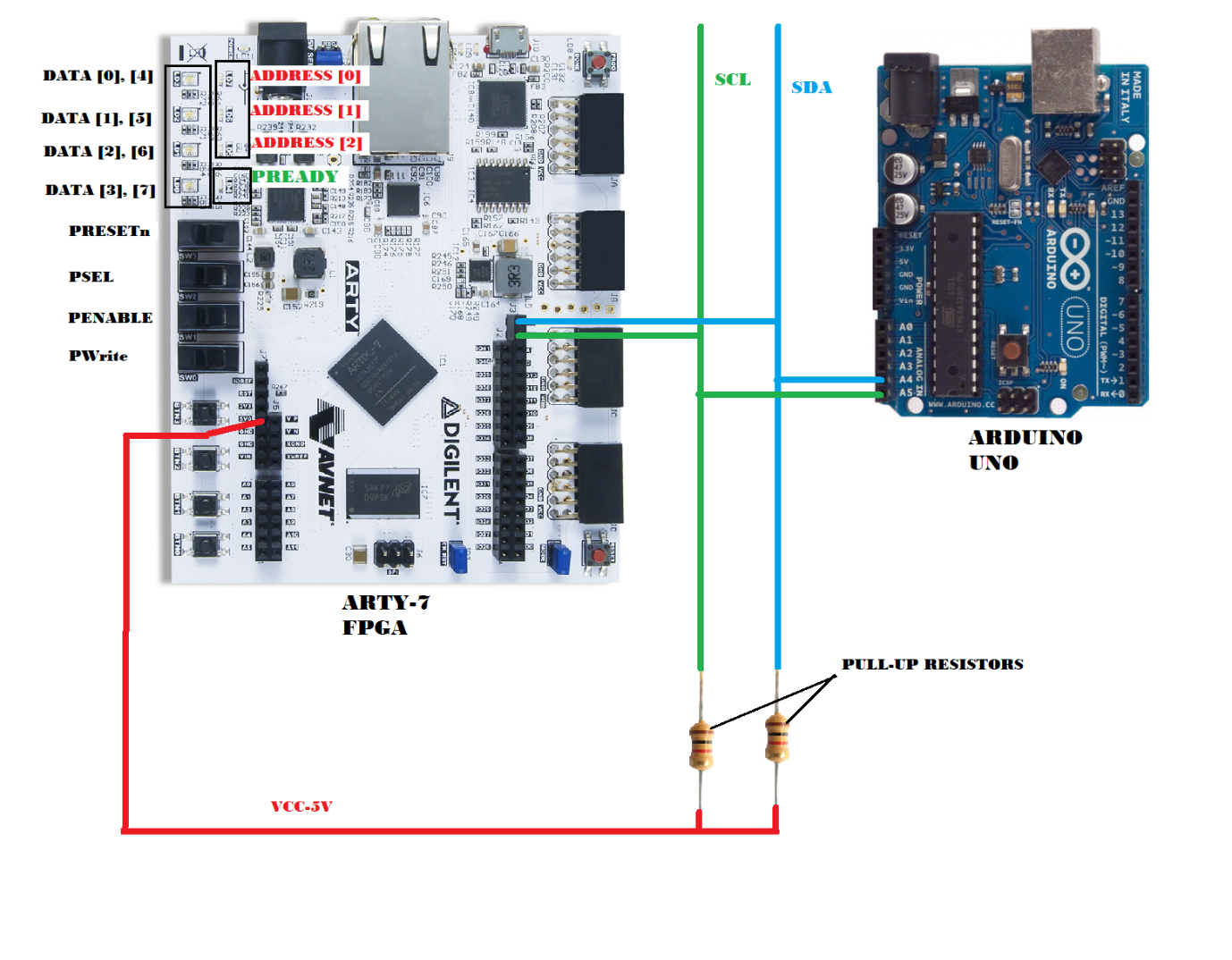
When, PWrite = 1, RGB LEDs -> PWDATA [Index] [0:7], BUTTONs -> Index

When, PWrite = 0, RGB LEDs -> PRDATA [Index] [0:7], BUTTONs -> Index

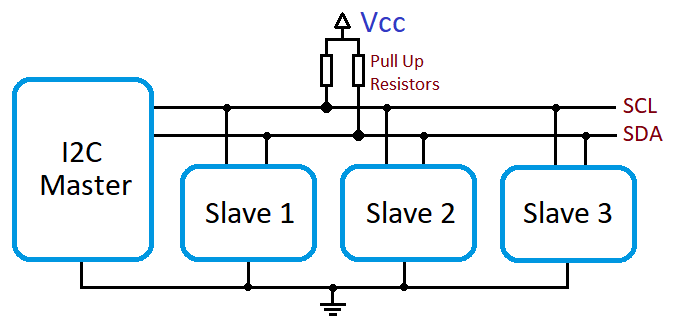




|  |  |  |
| --- | --- | --- |
| **NET** | **Type** | **Port Number** |
| PCLK | Input | E3 (CLK 100MHz, internal pin) |
| PRESETn | Input | A10 (SW3) |
| PSEL | Input | C10 (SW2) |
| PENABLE | Input | C11 (SW1) |
| PWrite | Input | A8 (SW0) |
|  |  |  |
| BUTTON [0] | Input | D9 (BTN0) |
| BUTTON [1] | Input | C9 (BTN1) |
| BUTTON [2] | Input | B9 (BTN2) |
| BUTTON [3] | Input | B8 (BTN3) |
|  |  |  |
| ADDR [0] | Output | J5 (LED5) |
| ADDR [1] | Output | T9 (LED6) |
| ADDR [2] | Output | T10 (LED7) |
|  |  |  |
| PREADY | Output | H5 (LED4) |
|  |  |  |
| DATA [7] | Output | F6 (led0\_g) |
| DATA [6] | Output | J4 (led1\_g) |
| DATA [5] | Output | J2 (led2\_g) |
| DATA [4] | Output | H6 (led3\_g) |
| DATA [3] | Output | E1 (led0\_b) |
| DATA [2] | Output | G4 (led1\_b) |
| DATA [1] | Output | H4 (led2\_b) |
| DATA [0] | Output | K2 (led3\_b) |
|  |  |  |
| I2C-SCL | Inout (bidirectional) | L18 (scl) |
| I2C-SDA | Inout (bidirectional) | M18 (sda) |

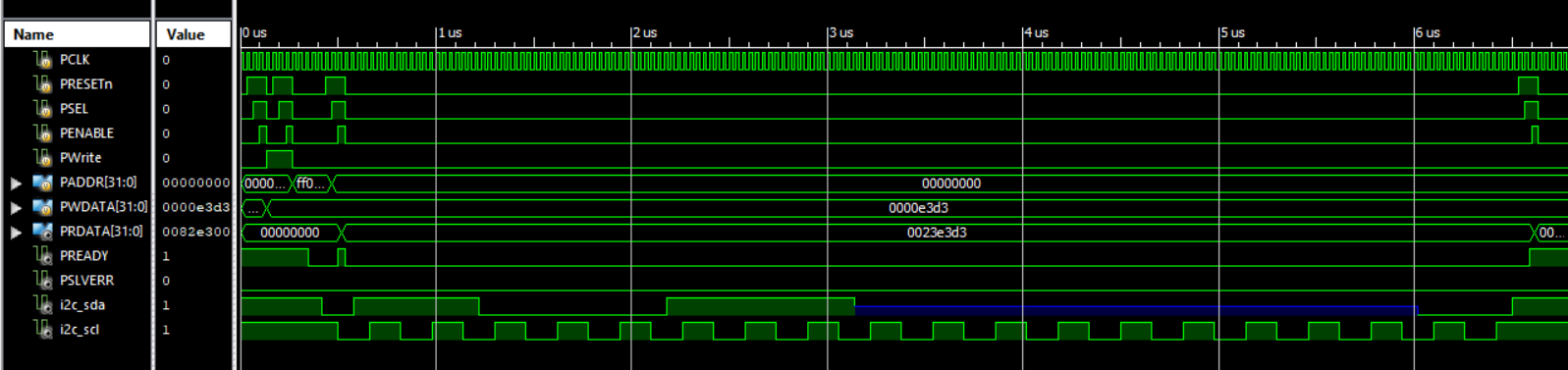
**Circuit Connections:**

**Bus connection:**

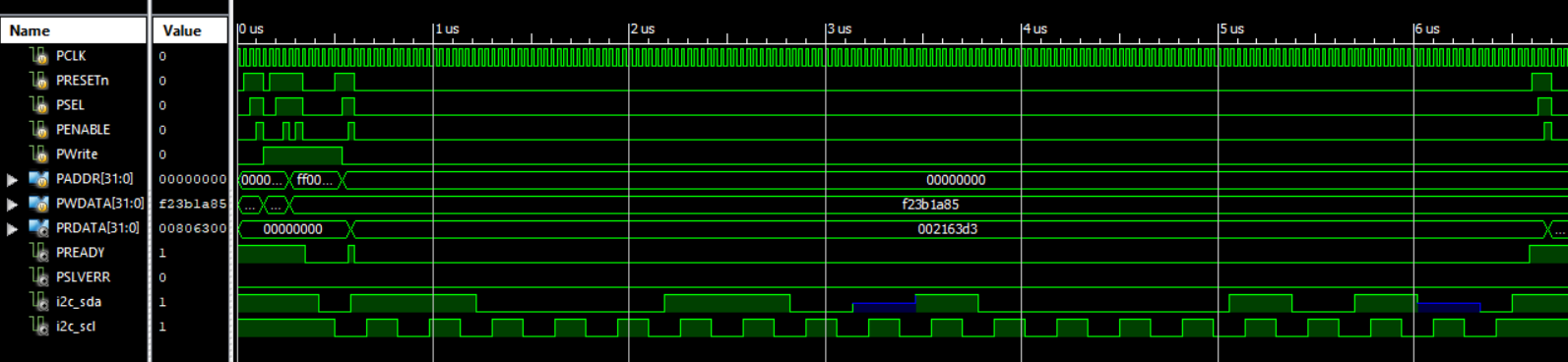
****

**Output:**

**Reading from I2C slave device:**

****

**Writing to I2C slave device:**

****