Lab 04 - I2C Communication Protocol

Name of student (Batch No / Roll No)

Divyang Bagla (D2 /PD 33)

Performance of Experiment	Journal Submission	Total Marks	Remarks	Instructor Sign

Aim: To study I2C communication and simulation on tinker cad using ardunios such that 1 master arduino and 2 slaves arduino.

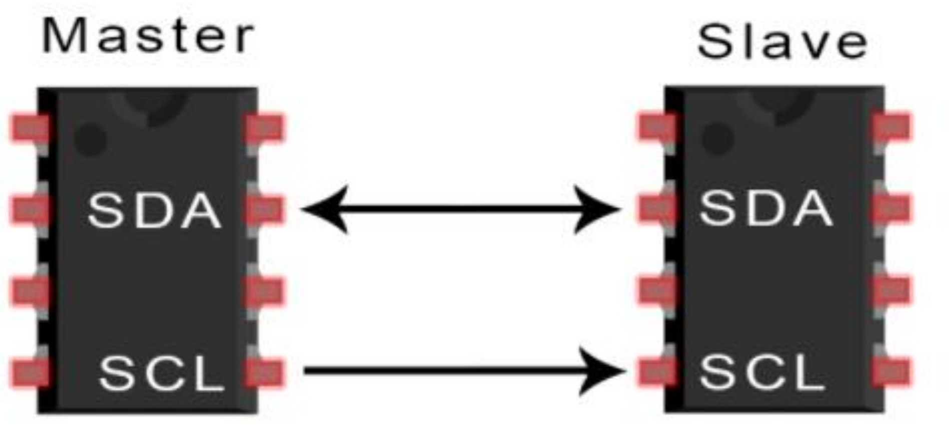
Journal content- Theory and frequently asked questions and experiment

- Introduction to I2C Communication
- How does I2C Works ?
- Steps of I2C Data Transmission
- Single Master with Multiple Slaves
- Advantages & Disadvantages
- Conclusions

1. Introduction to I2C Communication

I2C combines the best features of SPI and UARTs. With I2C, you can connect multiple slaves to a single master (like SPI) and you can have multiple masters controlling single, or multiple slaves. This is really useful when you want to have more than one microcontroller logging data to a single memory card or displaying text to a single LCD.

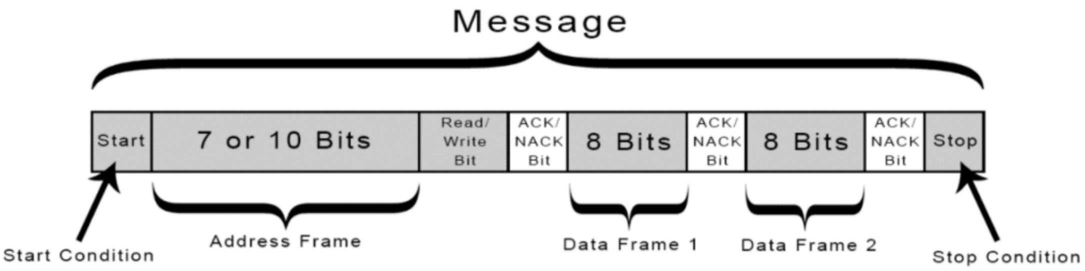
Like UART communication, I2C only uses two wires to transmit data between devices:



SDA (Serial Data) – The line for the master and slave to send and receive data.
SCL (Serial Clock) – The line that carries the clock signal.

2. How does I2C Works ?

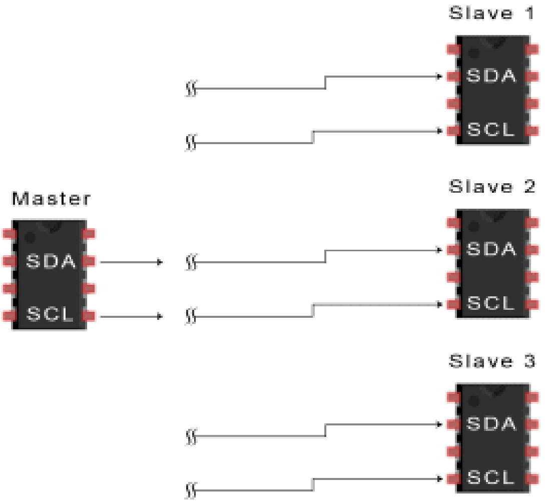
With I2C, data is transferred in *messages*. Messages are broken up into *frames* of data. Each message has an address frame that contains the binary address of the slave, and one or more data frames that contain the data being transmitted. The message also includes start and stop conditions, read/write bits, and ACK/NACK bits between each data frame:



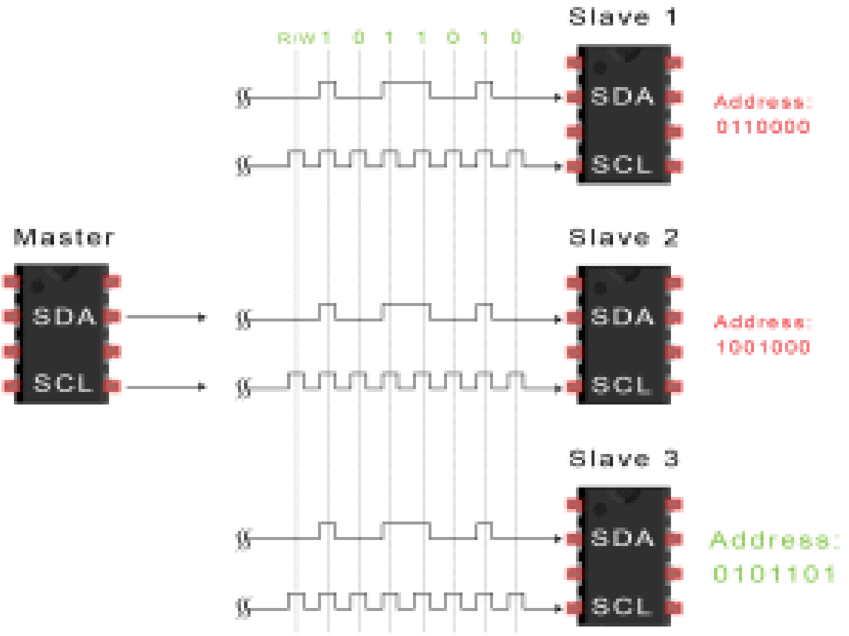
Start Condition: The SDA line switches from a high voltage level to a low voltage level *before* the SCL line switches from high to low.
Stop Condition: The SDA line switches from a low voltage level to a high voltage level *after* the SCL line switches from low to high.
Address Frame: A 7 or 10 bit sequence unique to each slave that identifies the slave when the master wants to talk to it.
Read/Write Bit: A single bit specifying whether the master is sending data to the slave (low voltage level) or requesting data from it (high voltage level).
ACK/NACK Bit: Each frame in a message is followed by an acknowledge/no-acknowledge bit. If an address frame or data frame was successfully received, an ACK bit is returned to the sender from the receiving device.

3. Steps of I2C Data Transmission

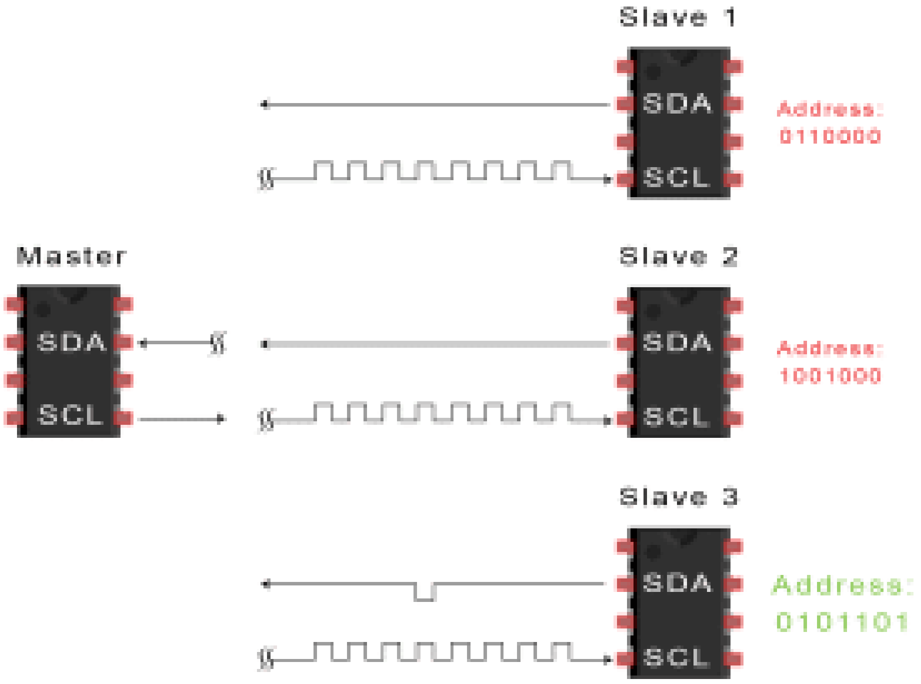
1. The master sends the start condition to every connected slave by switching the SDA line from a high voltage level to a low voltage level *before* switching the SCL line from high to low:



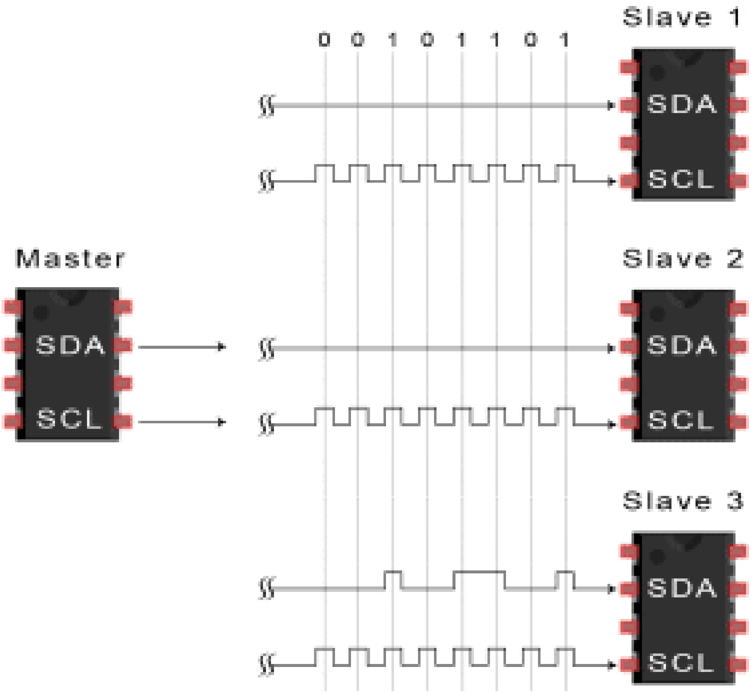
2. The master sends each slave the 7 or 10 bit address of the slave it wants to communicate with, along with the read/write bit:



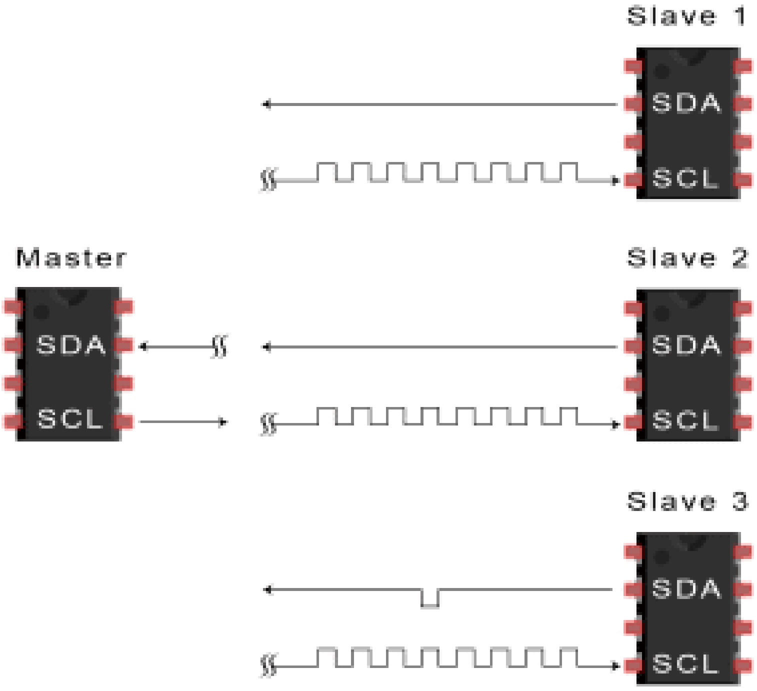
3. Each slave compares the address sent from the master to its own address. If the address matches, the slave returns an ACK bit by pulling the SDA line low for one bit. If the address from the master does not match the slave's own address, the slave leaves the SDA line high.



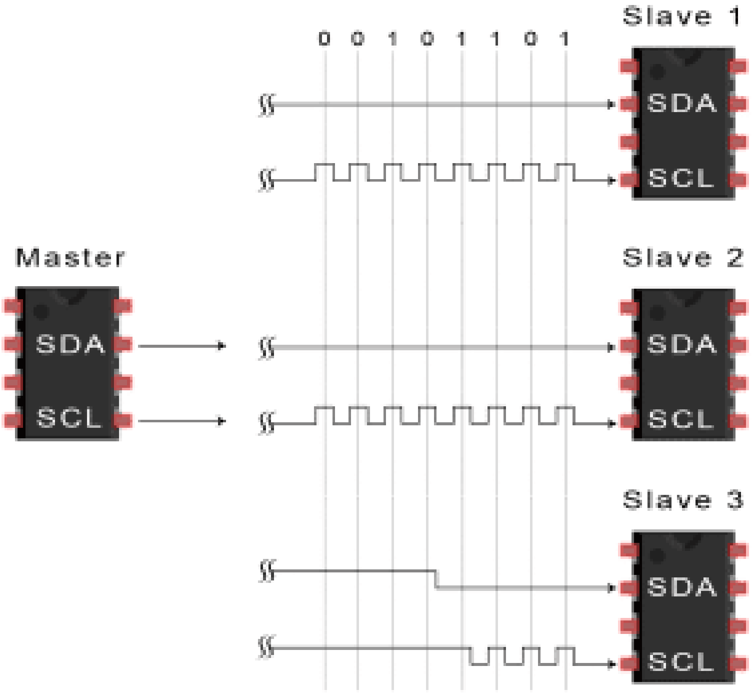
4. The master sends or receives the data frame:



5. After each data frame has been transferred, the receiving device returns another ACK bit to the sender to acknowledge successful receipt of the frame:

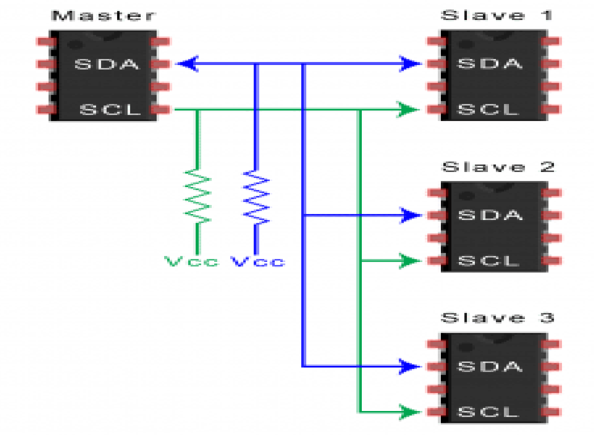


6. To stop the data transmission, the master sends a stop condition to the slave by switching SCL high before switching SDA high:



4. Single Master with Multiple Slaves

Because I2C uses addressing, multiple slaves can be controlled from a single master. With a 7 bit address, 128 (2^7) unique address are available. Using 10 bit addresses is uncommon, but provides 1,024 (2^{10}) unique addresses. To connect multiple slaves to a single master, wire them like this, with 4.7K Ohm pull-up resistors connecting the SDA and SCL lines to Vcc:



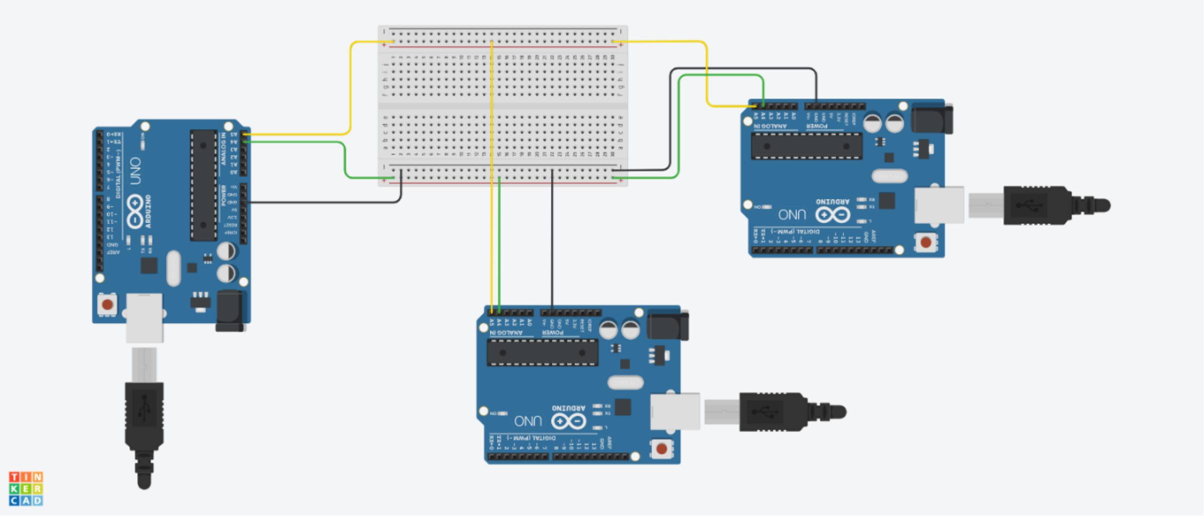
ADVANTAGES

- Only uses two wires
- Supports multiple masters and multiple slaves
- ACK/NACK bit gives confirmation that each frame is transferred successfully
- Hardware is less complicated than with UARTs
- Well known and widely used protocol

DISADVANTAGES

- Slower data transfer rate than SPI
- The size of the data frame is limited to 8 bits
- More complicated hardware needed to implement than SPI

Circuit Digarm :-



MASTER SLAVE CODE BY **DIVYANG BAGLA (PD33)** :-

MASTER CODE

```
#include<Wire.h>

void setup()

{

  Wire.begin();

}

byte x = 0;

void loop()

{

  Wire.beginTransmission(7);

  Wire.write("X is ");

  Wire.write(x);

  Wire.endTransmission();



  x++;

  delay(1000);

}
```

SLAVE 7 CODE

```
#include<Wire.h>

void setup()

{

  Wire.begin(7);

  Wire.onReceive(receiveEvent);

  Serial.begin(9600);

}

void loop()

{

  delay(100);

}

void receiveEvent(int howMany){

  while(1 < Wire.available()){

    char c =  Wire.read();

    Serial.print(c);

  }

  int x = Wire.read();

  Serial.println(x);

}
```

SLAVE 8 CODE

```
#include<Wire.h>

void setup()

{

  Wire.begin(8);

  Wire.onReceive(receiveEvent);

  Serial.begin(9600);

}

void loop()

{

  delay(100);

}

void receiveEvent(int howMany){

  while(1 < Wire.available()){

    char c =  Wire.read();

    Serial.print(c);

  }

  int x = Wire.read();

  Serial.println(x);

}
```

CONCLUSION :-

Thus learned about I2C communication and completed the simulation on tinker cad.