

INTERNET OF THINGS

ASSIGNMENT - 1

PREPARED BY

Garv Baheti

(2020BTechCSE031)

SUBMITTED TO

Mr. Divanshu Jain



NAAC 'A' Grade Accredited

**Department of Computer Science Engineering
Institute of Engineering and Technology
JK Lakshmipat University Jaipur**

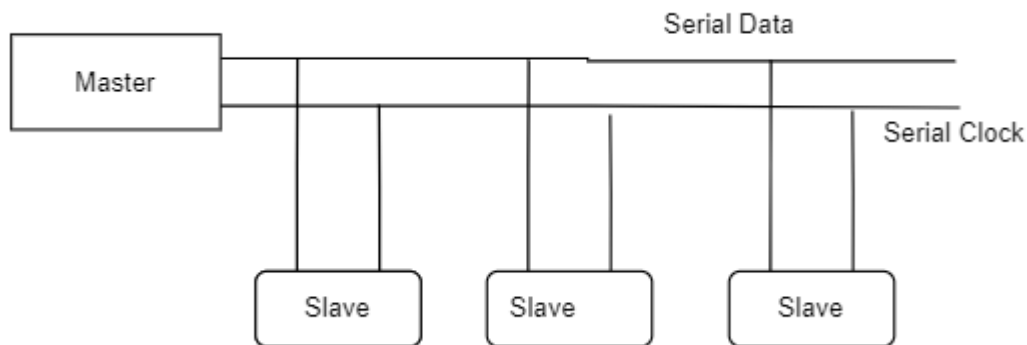
September 22, 2022

I²C PROTOCOL

I²C is termed as the abbreviated form of Inter-Integrated Circuit. This is a type of communication bus which is mainly designed and developed to establish inter-chip communication. This protocol is a bus interface connection that is embedded into multiple devices to set up serial connections. I2C is the most prominent chip-to-chip communication protocol as it holds the features of both UART and SPI.

Features of the I²C protocol

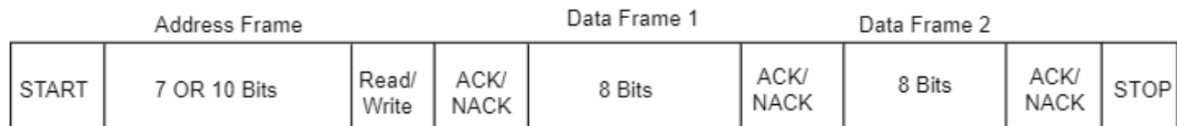
1. Information transfer happens in the form of blocks or frames, and it can be configured in a multi-master approach
2. Clock stretching takes place which means when the slave device is not in the condition of accepting more data, then the clock is stretched making the SCL signal to LOW therefore master gets disabled to raise the CLK signal. The master signal does not raise the CLK signal as the wires are connected in AND format and must wait till the slave releases the SCL line making sure that it is ready to send the subsequent bit.
3. It follows a simple approach for validation of data transfer
4. Using two shared bus lines, any device on the I2C network can be controlled
5. Adjustment of data transfer can happen at any point of time which makes sense for no prior agreement like in UART protocol



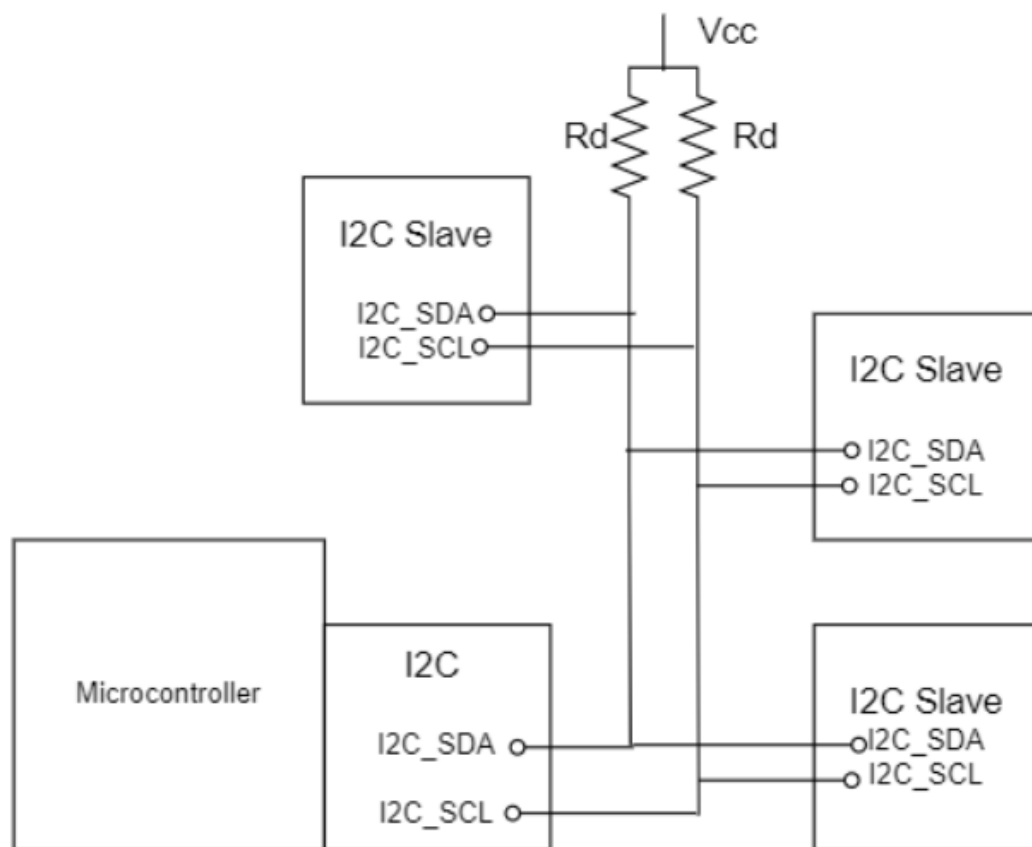
This is a multi-slave and multi-master communication protocol that gives the flexibility of connecting multiple IC at a single instance of time to a single bus. In general, in I2C devices, the master takes the initiation of establishing communication whereas in the case of multi-master, only one master holds the possession of setting up communication.

I2C Protocol Working Principle

The working of the I2C communication protocol happens through open drain lines which are Serial Data (SDA) and SCL (Serial Clock). Initially, both the SDA and SCL lines are pulled high and the bus mainly functions in two modes which are Master and Slave.

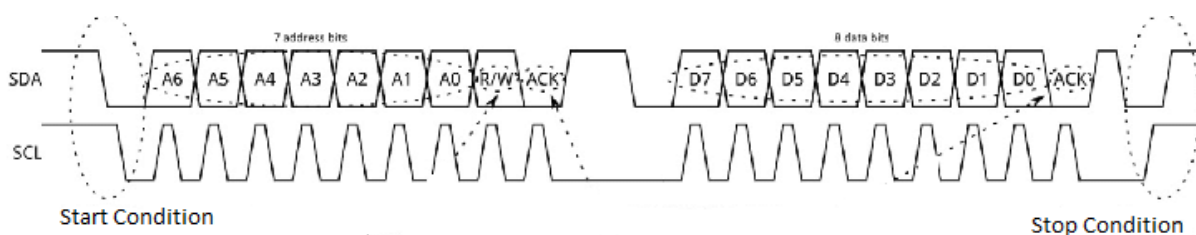


Every bit which is transferred on the serial data line gets synchronized by HIGH to LOW pulse for every clock signal that is received on the serial clock. The SDA line will not change when the SCL is in a HIGH state and it is changed only when SCL is in a LOW state. As discussed SDA and SCL are open-drain lines, they require of pull-up resistor to make them high as because the devices on the bus are in the active LOW state.



Start and Stop Bits

The switching of serial data line takes place from high to low voltage level and low to high voltage levels before and after the serial clock line switches from HIGH to LOW and LOW to HIGH. The below picture shows the switching of SDA and SCL lines.



Timing Diagram

Address

This is the immediate frame after the start bit. The master bit transfers the slave's address with which it has to communicate to each slave that was connected to it. Then the slave bit compares its own address with the address of the slave that was sent by the master. When both the addresses match, it transmits a low voltage ACK signal to the master signal. Whereas when both the addresses do not match, the slave remains idle and the serial data line stays at HIGH.

Read/Write

When the Read/Write indicates '1', then the master is transmitting data to the slave whereas when Read/Write indicates '0', then the master is receiving data from the slave signal.

ACK/NACK

The acknowledge/no-acknowledge bit is the subsequent bit of every frame in a message. When data/address was successfully transmitted, then the ACK signal gets back to the sender from the receiver device.

Data Frame

When the ACK bit is detected by master from slave it indicates that the first data frame can be transferred. This data frame is of 8-bit length. After the data frame, ACK/NACK bit detects the successful transmission of the frame. Once the transmission was successful, then the next data frame will be ready to transmit.

I2C PROTOCOL ARDUINO

Potentiometers are used at each Arduino board in order to know the transmission values from master to slave and vice versa. So, by altering the potentiometer, values can be determined. Few of the wiring library functionalities to be known here are explained below:

`Wire.begin(address)` – This initiates the wiring library and connects the bus either as master or slave.

`Wire.read()` – Used to read the information which was received either from slave or master device.

`Wire.write()` – Used to write the information to slave or master device.

`Wire.beginTransmission(address)` – With the provided slave's address, this library initiates the transmission to I2C components.

`Wire.endTransmission()` – It is used to end the transmission which was initiated by `beginTransmission()`.

`Wire.onRequest()` – This library is called when the master requests information from the slave device.

`Wire.onReceive()` – This library is called when the slave receives information from the master device.

`Wire.requestForm(address, quantity)` – This library helps when the master requests bytes from the slave.

At pin A0, analog input is provided which ranges between 0V – 5V, and convert the analog value to digital value (0 to 1023). The converted value is further converted to 7-bit values (0 to 127) because only 7-bit information is transmitted for I2C communication. And the next process takes place at A4 and A5 pins of Arduino.

Specifications

The specifications allow manufacturers to design I2C devices to be in compatible with the bus requirements. Below is the **I2C protocol specification** which is generally considered:

- The clock frequency for Serial Clock has a minimum of '0' and a maximum of 400-1000kHz
- The data setup time in the fast mode operation is 50 ns
- The data validation time ranges between a maximum of 0.9 – 0.45 μ sec
- The data validation acknowledgment time ranges between a maximum of 0.9 – 0.45 μ sec
- The noise margins at the HIGH and LOW levels are $0.2V_{DD}$ and $0.1 V_{DD}$ volts
- The rising time of serial data and serial clock signals has a minimum of 20 ns and a maximum of 300 ns

- The falling time of serial data and serial clock signals has a minimum of 20 ns and a maximum of 300 ns
- The capacitive load at every bus is 400pF
- The time where the bus is free between STOP and START conditions is a minimum of 0.5μsec

Difference between UART, SPI and I2C

The below tabular column compares the features of various kinds of interfaces by considering multiple aspects such as the interfaces data rate, type of communication, complexity, and many others.

Feature	I2C	UART	SPI
Termed as	Inter-Integrated Circuit	Universal Asynchronous Receiver/Transmitter	Serial Peripheral Interface
Type of Transmission	Synchronous	Asynchronous	Synchronous
The maximum level of Speed and Distance	1Mbps and 0.5 mts	115.2Kbps and 15 mts	4Mbps and 0.1 mts
Pins	SDA – Serial Data SCL – Serial Clock	TxD – Data Transmit RxD – Data Receive	SS – Select Slave MISO – Master Input and Slave Output MOSI – Master Output and Slave Input SCLK – Serial Clock
Clock	I2C has one clock signal between multiple slaves and masters	There is no common clock signal. There will be three independent clock signals	A common serial clock signal exists between slave and master devices
Software Addressing	In the I2C interface, 2^7 or 2^{10} slave devices can be addressed	No addressing is required	In order to address slave that is connected with master, slave select lines are used
Complexity of Hardware	More complex	Less	Less

Advantages

- Even though there are multiple devices on the bus, the signals required are less
- It has the ability to support many masters

- It has enhanced adaptability where the requirements of various slave devices can be achieved
- It has multiple-master and multiple-slave flexibility
- I2C protocol has no complications in design

Disadvantages

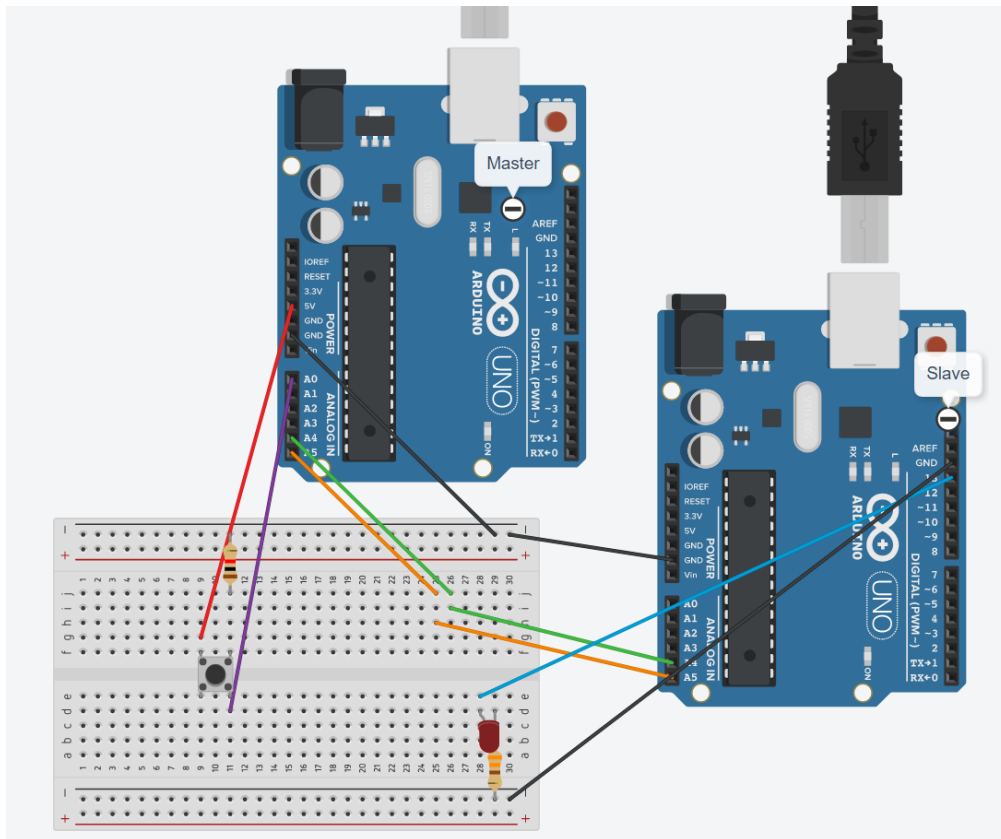
- I2C protocol is designed with an open-drain configuration where it limits the speed and also pull-up resistors are used which further lessens the communication speed
- It needs more space

Applications of I2C Protocol

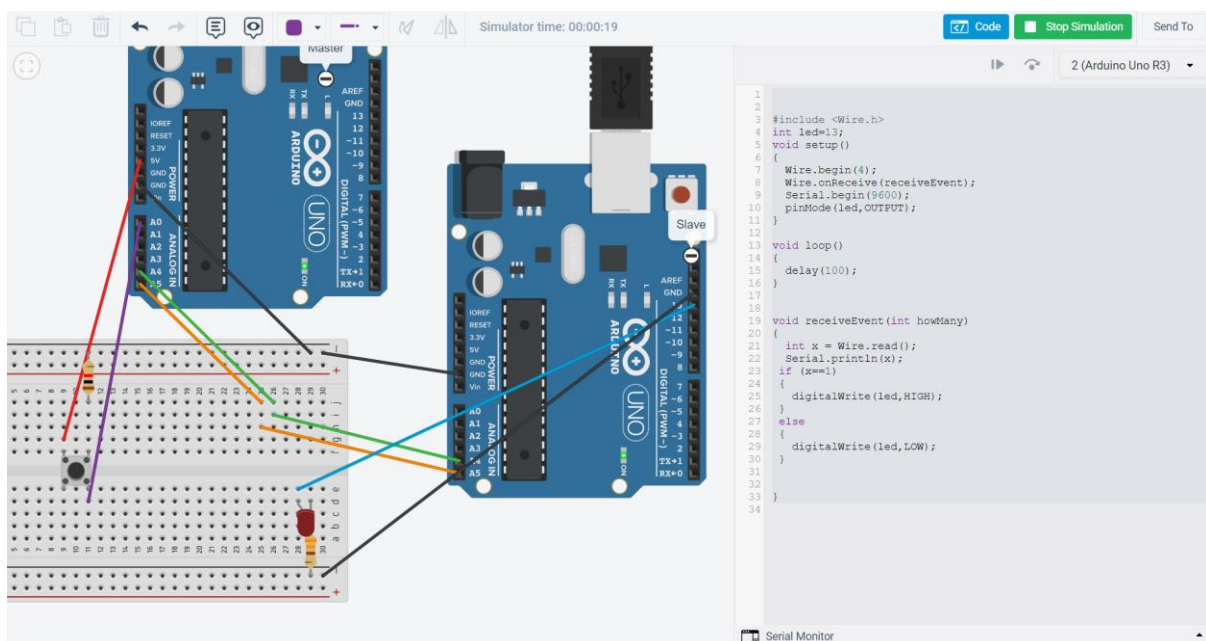
The applications of Inter-Integrated interface can be found in the devices that operate at less speed. A few of those are:

1. Hardware sensors
2. Used in communications having multiple micro-controller
3. Employed in the assessment of ADC and DAC circuits
4. Used in EEPROM devices

CONTROLLING THE LED CONNECTED TO ARDUINO 2 USING SWITCH 1 CONNECTED TO ARDUINO 1



Circuit Diagram



CODE

```
#include <Wire.h>

int led=13;

void setup(){
    Wire.begin(4);

    Wire.onReceive(receiveEvent);

    Serial.begin(9600);

    pinMode(led,OUTPUT);
}

void loop(){
    delay(100);
}

void receiveEvent(int howMany){
    int x = Wire.read();

    Serial.println(x);

    if (x==1){
        digitalWrite(led,HIGH);
    }
    else
    {
        digitalWrite(led,LOW);
    }
}
```

(i) When Switch is Pressed >> LED on Slave Arduino is turned on

