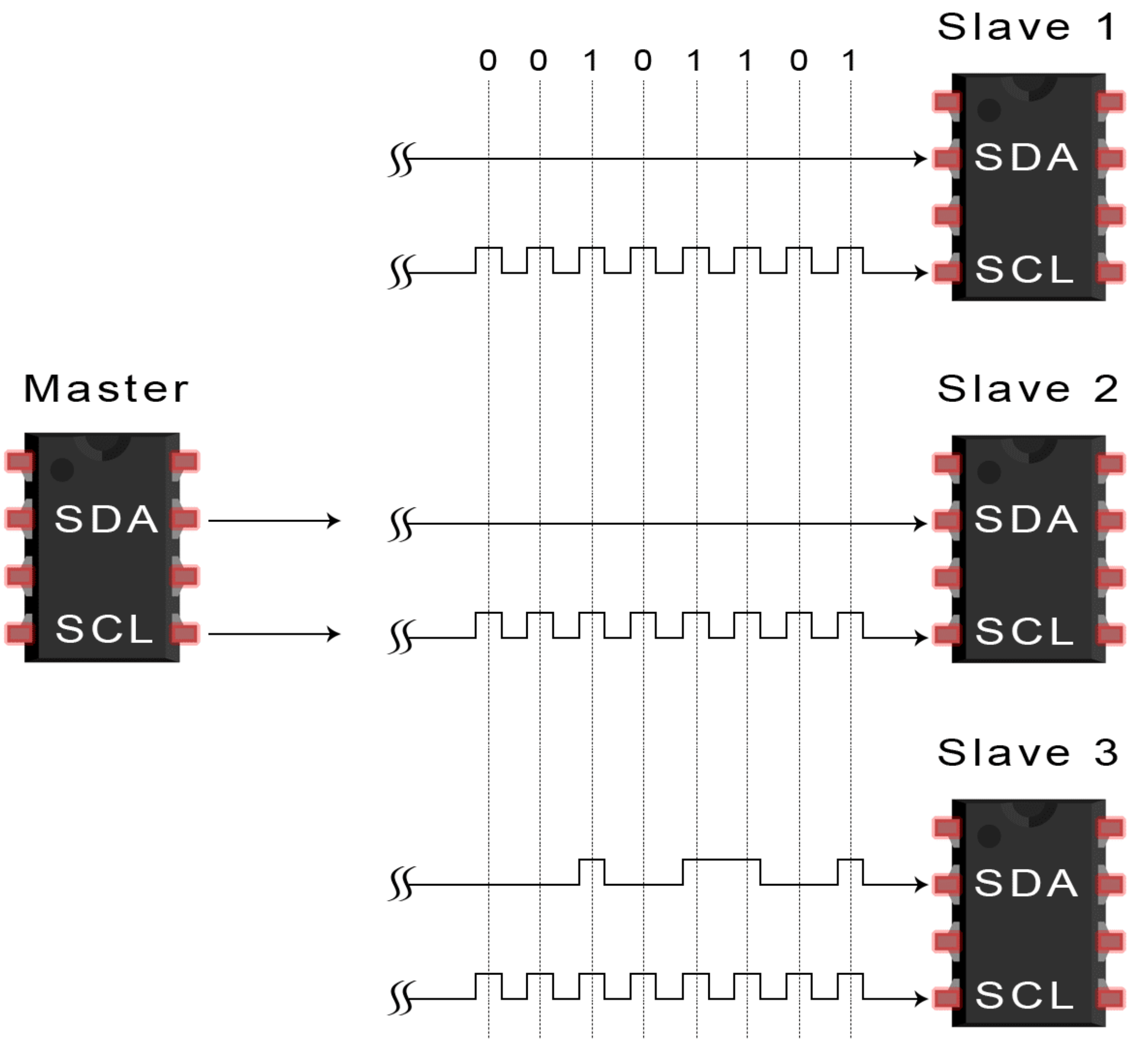
I2C Communication



Team

Tanuj

SreeTeja

AIM

The aim of the project is to establish a communication between arduinos to send and receive data among different arduinos and do a specific task.

What is i2c communication..?

Transmitting and receiving the information between two or more than two devices require a communication path called as a bus system. A I2C bus is a bidirectional two-wired serial bus which is used to transport the data between integrated circuits. The I2C stands for “Inter Integrated Circuit”. It was first introduced by the Philips semiconductors in 1982. The I2C bus consists of three data transfer speeds such as standard, fast-mode and high-speed-mode. The I2C bus supports 7-bit and 10-bit address space device and its operation differ with low voltages.

Constituents of I2C

The I2C is a serial bus protocol consisting of two signal lines such as SCL and SDL lines which are used to communicate with the devices. The SCL stands for a ‘serial clock line’ and this signal is always driven by the ‘master device’. The SDL stands for the ‘serial data line’, and this signal is driven by either the master or the I2C peripherals.

I2C SPEEDS

The original I2C bus had a maximum speed of 100 KHz. Most common applications still use this speed, as it is quite sufficient for transferring data from sensors and to simple displays.

I2C and has some higher speed modes. Not all I2C devices support these modes:

Fast Mode : This has a maximum clock speed of 400 KHz.

Hi-Speed Mode : A maximum clock frequency fo 3.4 MHz

Ultra Fast Mode : Maximum clock frequency of 5 MHz

On an I2C bus it is the master that determines the clock speed.

Procedure

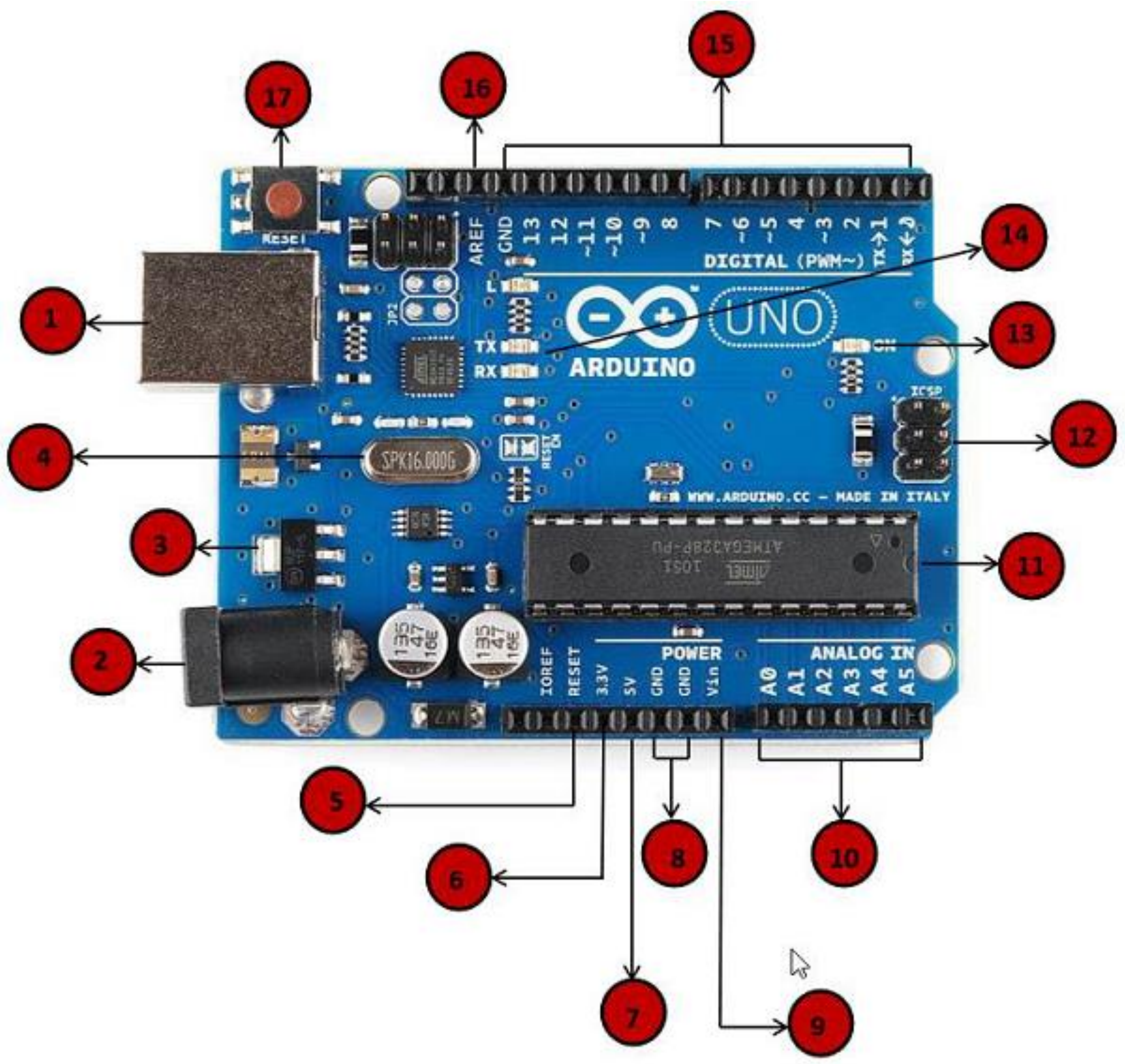
Step1: First, the master device issues a start condition to inform all the slave devices so that they listen on the serial data line.
Step2: The master device sends the address of the target slave device which is compared with all the slave devices' addresses as connected to the SCL and SDL lines. If anyone address matches, that device is selected, and the remaining all devices are disconnected from the SCL and SDL lines.
Step3: The slave device with a matched address received from the master, responds with an acknowledgement to the master thereafter communication is established between both the master and slave devices on the data bus.
Step4: Both the master and slave receive and transmit the data depending on whether the communication is read or write.
Step5: Then, the master can transmit 8-bit of data to the receiver which replies with a 1-bit acknowledgement.

Why I2C

- maintains low pin/signal count even with numerous devices on the bus
- adapts to the needs of different slave devices
- readily supports multiple masters
- incorporates ACK/NACK functionality for improved error handling

Arduino

Arduino is an open-source platform used for building electronics projects. Arduino consists of both a physical programmable circuit board (often referred to as a microcontroller) and a piece of software, or IDE (Integrated Development Environment) that runs on your computer, used to write and upload computer code to the physical board.



1	**Power USB** Arduino board can be powered by using the USB cable from your computer. All you need to do is connect the USB cable to the USB connection (1).
2	**Power (Barrel Jack)** Arduino boards can be powered directly from the AC mains power supply by connecting it to the Barrel Jack (2).
3	**Voltage Regulator** The function of the voltage regulator is to control the voltage given to the Arduino board and stabilize the DC voltages used by the processor and other

	elements.
4	**Crystal Oscillator** The crystal oscillator helps Arduino in dealing with time issues. How does Arduino calculate time? The answer is, by using the crystal oscillator. The number printed on top of the Arduino crystal is 16.000H9H. It tells us that the frequency is 16,000,000 Hertz or 16 MHz.
5,17	**Arduino Reset** You can reset your Arduino board, i.e., start your program from the beginning. You can reset the UNO board in two ways. First, by using the reset button (17) on the board. Second, you can connect an external reset button to the Arduino pin labelled RESET (5).
6,7 8,9	**Pins (3.3, 5, GND, Vin)** • 3.3V (6) − Supply 3.3 output volt • 5V (7) − Supply 5 output volt • Most of the components used with Arduino board works fine with 3.3 volt and 5 volt. • GND (8)(Ground) − There are several GND pins on the Arduino, any of which can be used to ground your circuit. • Vin (9) − This pin also can be used to power the Arduino board from an external power source, like AC mains power supply.
10	**Analog pins** The Arduino UNO board has six analog input pins A0 through A5. These pins can read the signal from an analog sensor like the humidity sensor or temperature sensor and convert it into a digital value that can be read by the microprocessor.
11	**Main microcontroller** Each Arduino board has its own microcontroller (11). You can assume it as the brain of your board. The main IC (integrated circuit) on the Arduino is slightly different from board to board. The microcontrollers are usually of the ATMEL Company. You must know what IC your board has before loading up a new program from the Arduino IDE. This information is available on the top of the IC. For more details about the IC construction and functions, you can refer to the data sheet.

12	**ICSP pin** Mostly, ICSP (12) is an AVR, a tiny programming header for the Arduino consisting of MOSI, MISO, SCK, RESET, VCC, and GND. It is often referred to as an SPI (Serial Peripheral Interface), which could be considered as an "expansion" of the output. Actually, you are slaving the output device to the master of the SPI bus.
13	**Power LED indicator** This LED should light up when you plug your Arduino into a power source to indicate that your board is powered up correctly. If this light does not turn on, then there is something wrong with the connection.
14	**TX and RX LEDs** On your board, you will find two labels: TX (transmit) and RX (receive). They appear in two places on the Arduino UNO board. First, at the digital pins 0 and 1, to indicate the pins responsible for serial communication. Second, the TX and RX led (13). The TX led flashes with different speed while sending the serial data. The speed of flashing depends on the baud rate used by the board. RX flashes during the receiving process.
15	**Digital I/O** The Arduino UNO board has 14 digital I/O pins (15) (of which 6 provide PWM (Pulse Width Modulation) output. These pins can be configured to work as input digital pins to read logic values (0 or 1) or as digital output pins to drive different modules like LEDs, relays, etc. The pins labeled “~” can be used to generate PWM.
16	**AREF**

WIRE LIBRARY

I2C functions for Arduino

Wire.write (data)

It is used to write (transmit) data to the master or slave device.

Parameter

Data can be single byte value, string, array of data.

Wire.available()

This function is used by a master or slave to check the requested data is available or not. It returns the no. of bytes available.

Wire.read()

It is used to read the requested data by master from slave or read the data transmitted from a master to a slave.

Functions for Arduino I2C Master

Wire.begin ()

It initiates the Wire library and joins the bus as a master.

Wire.beginTransmission (slave address)

This function begins a transmission with the I2C slave device having specified slave address.

slave address 7-bit address of device with which we want to communicate.

e.g. Wire.beginTransmission (50)   //begin transmission with slave having address 50

Wire. requestFrom(address, no of byte) OR

Wire. requestFrom(address, no of byte, stop)

This function is used by master to request or receive data from slave device. The requested data can be read by using Wire.read().

Parameters

address of device with which we want to communicate

no. of byte needs to request

Wire.endTransmission()

It ends a transmission to a slave device that was begun by beginTransmission() and transmits the bytes that were queued by write().

Functions for Arduino I2C Slave

Wire.begin (address)

It initiates the Wire library and joins the I2C bus as a slave with specified address.

Parameter

7-bit slave address, if not specified then join the bus as master

Wire.onReceive(handler)

The handler function to be called when a slave device receives a transmitted data from a master.

OBJECTIVE

The objective of the project is to establish a serial communication between 3 arduinos. It consists of 3 devices.

1.Master

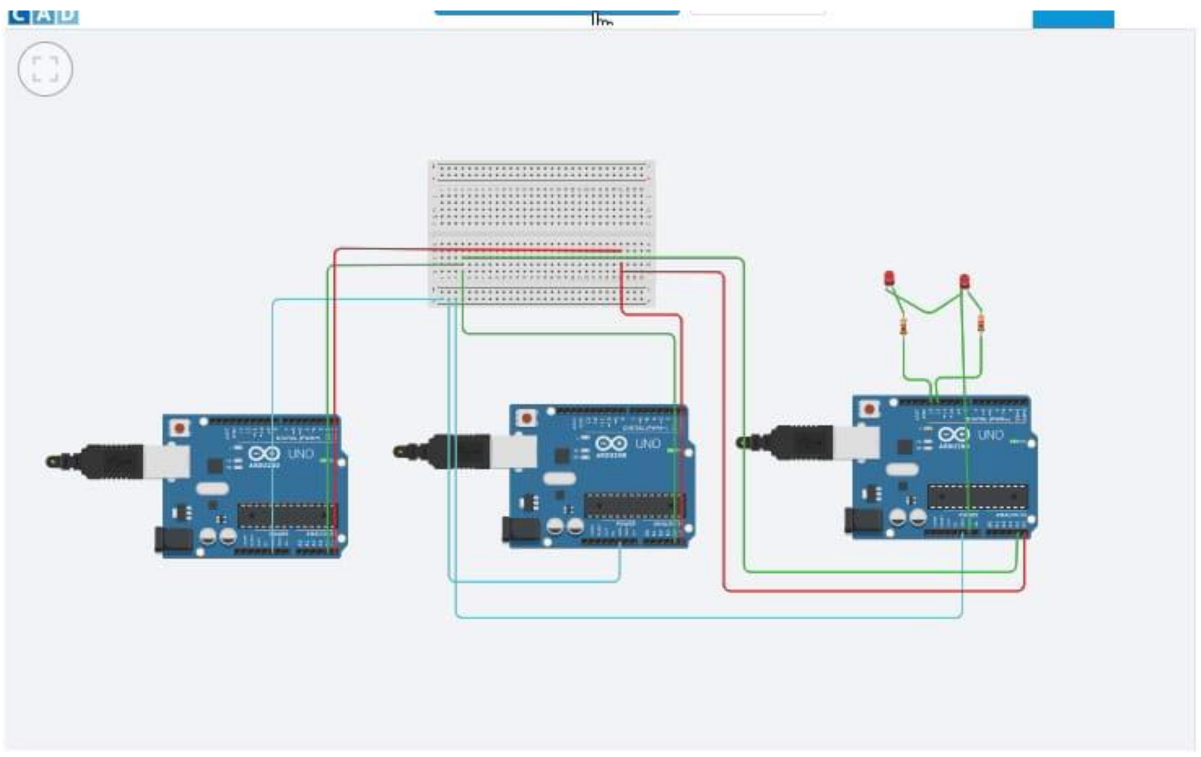
2.Slave 1

3.Slave 2

In the project the slave1 runs a for loop to generate the numbers 0,1,2 continuously. It sends that number to master. The master sends the number to slave2 which glows the led's connected to it, based on the number it receives.

"Wire Library" is used to write the code for the arduino's in the Arduino IDE.

The connection for the arduinos is as follows:-



Conclusion

An I2c is the easy and cheap communication protocol, It can be multi-master or multi-slave.