Student Lab Manual

Internet of Things

2180709

Index

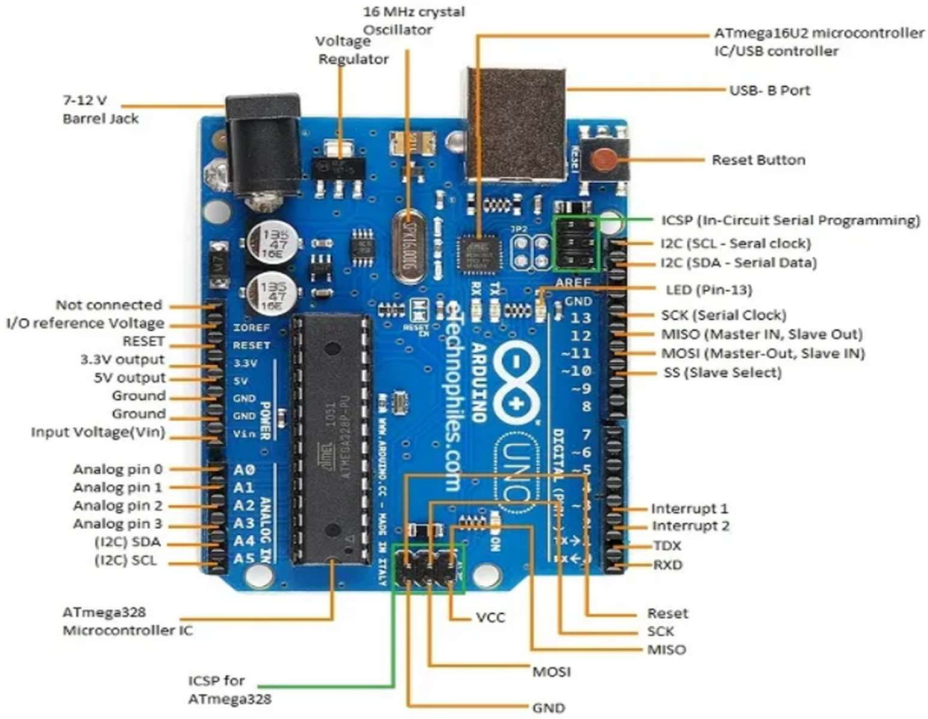
Experiment 1	3
Experiment 2	13
Experiment 3	27
Experiment 4	32
Experiment 5	37
Experiment 15	43

Experiment 1

Introduction to Open Source Hardware & its Application: (a) Arduino and (b) Raspberry Pi

A. Arduino:

➢ Pin Diagram & Description:



Pin Category	Pin Name	Details
Power	Vin, 3.3V, 5V, GND	Vin: Input voltage to Arduino when using an external power source. 5V: Regulated power supply used to power microcontroller and other components on the board. 3.3V: 3.3V supply generated by on-board voltage regulator. Maximum current draw is 50mA. GND: ground pins.
Reset	Reset	Resets the microcontroller.
Analog Pins	A0 – A5	Used to provide analog input in the range of 0-5V
Input/Output	Digital Pins 0 - 13	Can be used as input or output pins.

Pins		
Serial	0(Rx), 1(Tx)	Used to receive and transmit TTL serial data.
External Interrupts	2, 3	To trigger an interrupt.
PWM	3, 5, 6, 9, 11	Provides 8-bit PWM output.
SPI	10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK)	Used for SPI communication.
Inbuilt LED	13	To turn on the inbuilt LED.
TWI	A4 (SDA), A5 (SCA)	Used for TWI communication.
AREF	AREF	To provide reference voltage for input voltage.

➢ Specification

- ✓ **Microcontroller:** ATmega328p
- ✓ **Operating Voltage:** 5V
- ✓ **Input Voltage (recommended):** 7-12V
- ✓ **Input Voltage (limits):** 6-20V
- ✓ **Digital I/O Pins:** 14 pins (of which 6 are PWM output pins)
- ✓ **Analog Input Pins:** 6
- ✓ **DC Current per I/O Pin:** 40 mA
- ✓ **DC Current for 3.3V Pin:** 50 mA
- ✓ **Flash Memory:** 32 KB (of which 0.5 KB is taken by bootloader)
- ✓ **SRAM:** 2 KB (ATmega328)
- ✓ **EEPROM:** 1 KB (ATmega328)
- ✓ **Clock Speed:** 16 MHz
- ✓ **Length:**6 mm
- ✓ **Width:**4 mm
- ✓ **Weight:** 25 g

➢ Applications:

- ✓ **Arduino Based Home Automation System:**
  - The project is designed by using Arduino uno board for the development of home automation system with Bluetooth which is remotely controlled and operated by an Android OS smart phone. Houses are becoming smarter and well developed by using such kind

of advanced technologies. Modern houses are gradually increasing the way of design by shifting to centralized control system with remote controlled switches instead of conventional switches.

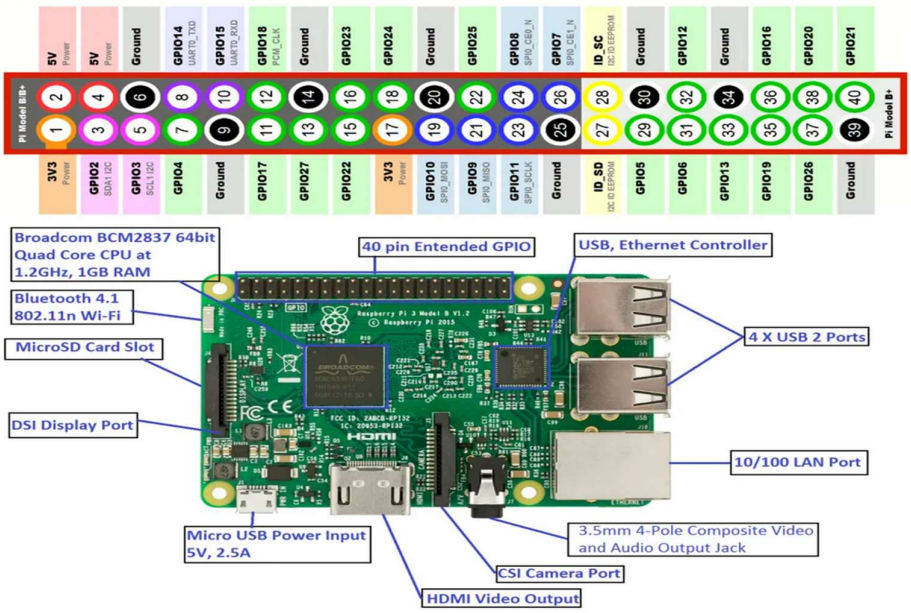
- In order to achieve this, a Bluetooth module is interfaced to the Arduino Uno board at the receiver end while on the transmitter end, a Graphical User Interface application on the cell phone sends ON/OFF commands to the receiver where loads are connected. By touching the identified location on the Graphical User Interface, lamps are used as loads in this project can be turned ON/OFF remotely by using this technology. The loads are operated by using Arduino Uno board through thyristors using triacs and OPTO-Isolators.

✓ **Arduino based Auto Intensity Control of Street Lights**

- As the intensity is cannot be controlled by using High Intensity Discharge (HID) lamps power saving is not possible in street lights with these lamps as the density on roads is decreasing from peak hours of nights to early morning.
- Thus, this system overcomes this problem by controlling the intensity of LED lights on street by gradually reducing intensity by controlling the voltage applied to these lamps. This system uses Arduino board to produce PWM pulses and it is programmed in such a way that it decreases the voltage applied to these lamps gradually till late nights and completely shutdowns at morning.
- Thus, Arduino development board can sense the environment by receiving input from different sensors and affects its surroundings by controlling motors, lights and other actuators. The microcontroller on the board is programmed using the Arduino programming language.

B. Raspberry PI:

➢ Pin Diagram & Description:



PIN GROUP	PIN NAME	DESCRIPTION
POWER SOURCE	+5V, +3.3V, GND and Vin	+5V -power output +3.3V -power output GND – GROUND pin
COMMUNICATION INTERFACE	UART Interface(RXD, TXD) [(GPIO15,GPIO14)]	UART (Universal Asynchronous Receiver Transmitter) used for interfacing sensors and other devices.
SPI Interface(MOSI, MISO, CLK,CE) x 2 [SPI0-(GPIO10 ,GPIO9,	SPI (Serial Peripheral Interface) used for communicating with other boards or peripherals.	

GPIO11 ,GPIO8)] [SPI1--(GPIO20 ,GPIO19, GPIO21 ,GPIO7)]		
TWI Interface(SDA, SCL) x 2 [(GPIO2, GPIO3)] [(ID_SD,ID_SC)]	TWI (Two Wire Interface) Interface can be used to connect peripherals.	
INPUT OUTPUT PINS	26 I/O	Although these some pins have multiple functionsthey can be considered as I/O pins.
PWM	Hardware PWM available on GPIO12, GPIO13, GPIO18, GPIO19	These 4 channels can provide PWM (Pulse Width Modulation) outputs. *Software PWM available on all pins
EXTERNAL INTERRUPTS	All I/O	In the board all I/O pins can be used as Interrupts.

➢ Specification:

- ✓ Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
- ✓ 2GB, 4GB or 8GB LPDDR4-3200 SDRAM (depending on model)
- ✓ 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless, Bluetooth 5.0, BLE
- ✓ Gigabit Ethernet
- ✓ 2 USB 3.0 ports; 2 USB 2.0 ports.
- ✓ Raspberry Pi standard 40 pin GPIO header (fully backwards compatible with previous boards)
- ✓ 2 × micro-HDMI ports (up to 4kp60 supported)

- ✓ 2-lane MIPI DSI display port
- ✓ 2-lane MIPI CSI camera port
- ✓ 4-pole stereo audio and composite video port
- ✓ H.265 (4kp60 decode), H264 (1080p60 decode, 1080p30 encode)
- ✓ OpenGL ES 3.0 graphics
- ✓ Micro-SD card slot for loading operating system and data storage
- ✓ 5V DC via USB-C connector (minimum 3A*)
- ✓ 5V DC via GPIO header (minimum 3A*)
- ✓ Power over Ethernet (PoE) enabled (requires separate PoE HAT)
- ✓ Operating temperature: 0 – 50 degrees C ambient

* A good quality 2.5A power supply can be used if downstream USB peripherals consume less than 500mA in total

➢ Applications:

✓ Desktop PC:

- Using Raspberry Pi, the microSD card, and a power supply a simple desktop can be made. We would also need an HDMI cable and a suitable display, maybe an old monitor. A USB keyboard and mouse are also needed.
- The new version which is Raspberry Pi 3 has built-in Wi-Fi and Bluetooth too. If a different model is used, compatible USB dongles would be required.
- Once everything is set up, and preferred operating system installed (the latest version of Raspbian), your desktop computer is ready to be used.

✓ Wireless print server:

- This requires installing Samba file-sharing software and CUPS (Common Unix Printing System). CUPS provide drivers for the printer and administration console.
- After this, Pi configuration is needed to ensure a Windows or Mac computers can access the printer via a network. The printer must have a USB cable.

✓ Media Usage:

- Many estimates suggest one of the main uses of Raspberry Pi is a Kodi media center. Several Kodi builds have been released as disk images. OSMC and OpenElec are among the most popular.
- Installing Kodi comes with some caveats. It is recommended that install only safe and legal add-ons from the official Kodi repositories.

Also, a Raspberry Pi running Kodi is vulnerable to a few security issues. Hence, setting up a VPN to encrypt data is recommended.

✓ **Game Servers:**

- Raspbian, default OS of pi comes with a special version of Minecraft game pre-installed. But, the applications of Raspberry Pi can be used as a game server as well. It is an excellent game server for Minecraft. If multiple Raspberry Pis are used, making one as a dedicated server, a great gaming experience can be achieved.
- Other multiplayer network games can be set up on the Raspberry Pi.

✓ **Retro Gaming Machine:**

- Raspberry Pi is ideal as a retro gaming machine. it fits as one of the lightest components of a machine. Particularly, it's a version, The Raspberry Pi Zero can fit into small spaces for gaming projects. There are two main options, Recalbox and RetroPie. Other platforms can be emulated too. Classic MS-DOS PC gaming and Commodore 64 can also be set-up and also many other popular 16-bit games consoles.

✓ **Robot Controller:**

- There are many robot-controller Raspberry Pi projects. There is a dedicated robotics package for Pi, duly powered with the device battery and used to communicate and control robots.
- for robots, Pi Zero W can only be used. Zero, which is a slimline version of the Raspberry Pi, has features of onboard wireless connectivity suitable for lightweight robots.
- It's quite lighter than the Model B+ boards of version 2 and 3 of pi, and the low profile ensures it can be placed in an efficient position, without having a concern of USB ports.

✓ **Stop Motion Camera:**

- Using the Python and a suitable mount (standard tripod for clay- or toy-based) and availability of well-lit area Stop motion camera can be built. But, this is a time-consuming process. One needs a good amount of practice to get good results.

✓ **Time-lapse Camera:**

- Combine the Raspberry Pi camera module along with a different script creates another use which is capturing time-lapse movies. This can be achieved by taking single frames with a time delay. Also needed is, perhaps a portable battery solution, and a tripod can be used. A smartphone tripod is most preferred to ensure the device remains sturdy.

✓ **FM Radio Station:**

- Raspberry Pi can also be used to broadcast on the FM radio. Pi can broadcast only over a short-range. A portable battery and soldering skills may be required here. Any audio which needs to broadcast will need to be loaded beforehand to the microSD card.

✓ **Web Servers:**

- Another great application of Raspberry Pi is to create a web server out of it. What this means is that it can be configured to host a website much like any other server. It can host blogs too. First of all, the right software needs to be installed and that is Apache and its dependent libraries. A full LAMP stack can also be installed with PHP, MySQL, and Apache, too. Setting up FTP is also helpful.
- Once all these steps as mentioned are completed, HTML files can be saved into the /www/ directory, and the webserver is ready to be used. Specific web software like WordPress can also be used once server setup is complete.

C. Difference Between Arduino & Raspberry PI:

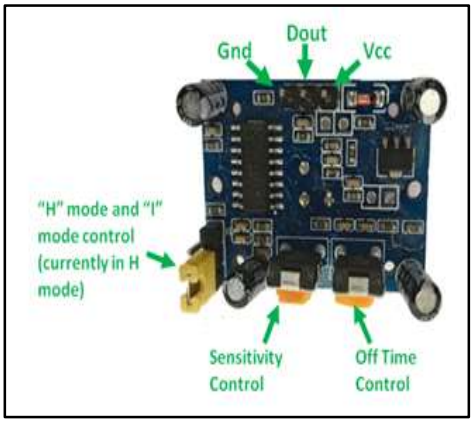
SL	Raspberry Pi	Arduino
1	It is a mini computer with Raspbian OS.It can run multiple programs at a time.	Arduino is a microcontroller, which is a part of the computer. It runs only one program again and again.
2	It is difficult to power using a battery pack.	Arduino can be powered using a battery pack.
3	It requires complex tasks like installing libraries and software for interfacing sensors and other components	It is very simple to interface sensors and other electronic components to Arduino.
4	It is expensive	It is available for low cost.
5	Raspberry Pi can be easily connected to the internet using Ethernet port and USB Wi-Fi dongles.	Arduino requires external hardware to connect to the internet and this hardware is addressed properly using code.
6	Raspberry Pi did not have storage on board. It provides an SD card port.	Arduino can provide onboard storage.
7	Raspberry Pi has 4 USB ports to connect different devices.	Arduino has only one USB port to connect to the computer.
8	The processor used is from ARM family.	Processor used in Arduino is from AVR family Atmega328P
9	This should be properly shutdown otherwise there is a risk of files corruption and software problems.	This is a just plug and play device. If power is connected it starts running the program and if disconnected it simply stops.
10	The Recommended programming language is python but C, C++, Python, ruby are pre-installed.	Arduino uses Arduino, C/C++

Experiment 2

Introduction to Following Hardware / Sensors & its Application

a. PIR Motion Sensor
b. Ultrasonic Sensor
c. Rain Drop Sensor
d. Moisture Sensor
e. Temperature Sensor
f. Touch Sensor
g. Infrared Sensor
h. Servo Motor
i. RFID Sensor
j. Bluetooth Module
k. Wi-Fi Module

A. PIR Motion Sensor:



➢ Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	Input voltage is +5V for typical applications. Can range from 4.5V- 12V
2	High/Low Ouput (Dout)	Digital pulse high (3.3V) when triggered (motion detected) digital low(0V) when idle (no motion detected
3	Ground	Connected to ground of circuit

➢ Specifications:

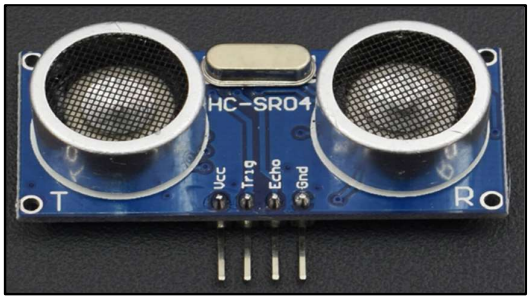
- ✓ Wide range on input voltage varying from 4.V to 12V (+5V recommended).
- ✓ Output voltage is High/Low (3.3V TTL).
- ✓ Can distinguish between object movement and human movement.
- ✓ Has to operating modes - Repeatable(H) and Non- Repeatable(H).
- ✓ Cover distance of about 120° and 7 meters.
- ✓ Low power consumption of 65mA.
- ✓ Operating temperature from -20° to +80° Celsius.

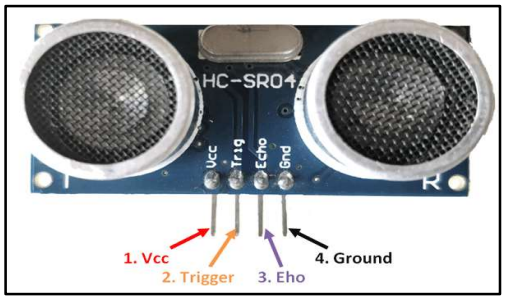
➢ Applications:

- ✓ All outdoor Lights
- ✓ Lift Lobby
- ✓ Multi Apartment Complexes
- ✓ Common staircases

- ✓ For Basement or Covered Parking Area
- ✓ Shopping Malls
- ✓ For garden light

B. Ultrasonic Sensor:





➢ Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	The Vcc pin powers the sensor, typically with +5V
2	Trigger	Trigger pin is an Input pin. This pin has to be kept high for 10us to initialize measurement by sending US wave.
3	Echo	Echo pin is an Output pin. This pin goes high for a period of time which will be equal to the time taken for the US wave to return back to the sensor.
4	Ground	This pin is connected to the Ground of the system.

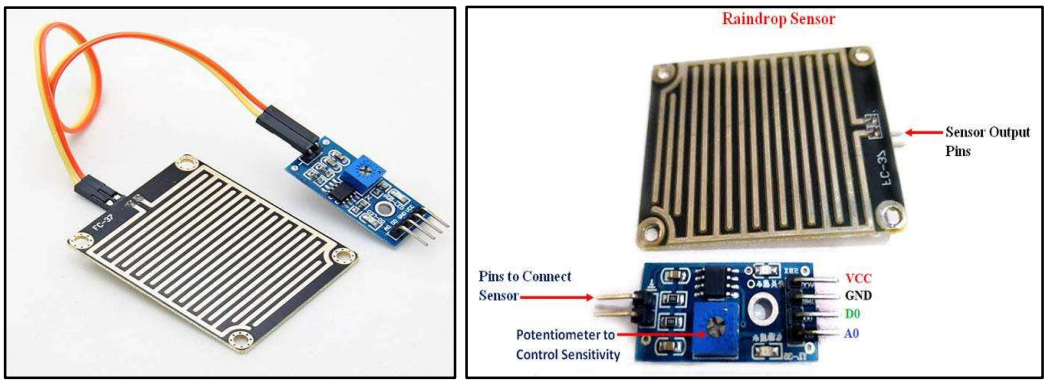
➢ Specifications:

- ✓ **Operating voltage:** +5V.
- ✓ **Theoretical Measuring Distance:** 2cm to 450cm.
- ✓ **Practical Measuring Distance:** 2cm to 80cm.
- ✓ **Accuracy:** 3mm.
- ✓ **Measuring angle covered:** <15°.
- ✓ **Operating Current:** <15mA.
- ✓ **Operating Frequency:** 40Hz.

➢ Applications:

- ✓ Used to avoid and detect obstacles with robots like biped robot, obstacle avoider robot, path finding robot etc.
- ✓ Used to measure the distance within a wide range of 2cm to 400cm
- ✓ Can be used to map the objects surrounding the sensor by rotating it
- ✓ Depth of certain places like wells, pits etc can be measured since the waves can penetrate through water

C. Rain Drop Sensor:



➢ Pin Configuration:

S.No:	Name	Function
1	VCC	Connects supply voltage- 5V
2	GND	Connected to ground
3	D0	Digital pin to get digital output
4	A0	Analog pin to get analog output

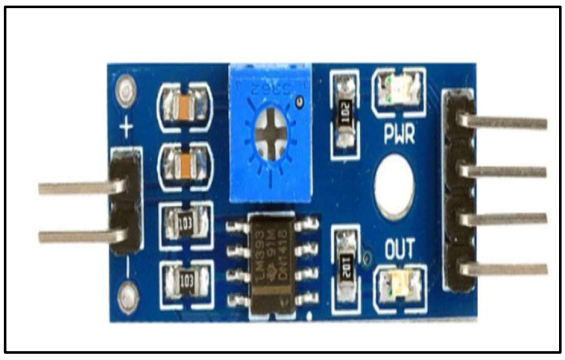
➢ Specifications:

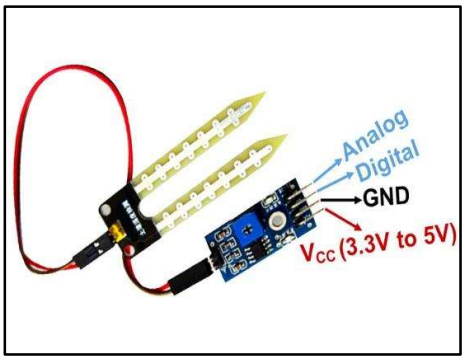
- ✓ Working voltage 5V
- ✓ Output format: Digital switching output (0 and 1), and analog voltage output AO
- ✓ Potentiometer adjust the sensitivity
- ✓ Uses a wide voltage LM393 comparator
- ✓ Comparator output signal clean waveform is good, driving ability, over 15mA
- ✓ Anti-oxidation, anti-conductivity, with long use time
- ✓ With bolt holes for easy installation
- ✓ Small board PCB size: 3.2cm x 1.4cm

➢ Applications:

- ✓ Automatic windshield wipers
- ✓ Smart Agriculture
- ✓ Home-Automation

D. Moisture Sensor:





➢ Pinout Configuration:

Pin Name	Description
VCC	The Vcc pin powers the module, typically with +5V
GND	Power Supply Ground
DO	Digital Out Pin for Digital Output.
AO	Analog Out Pin for Analog Output

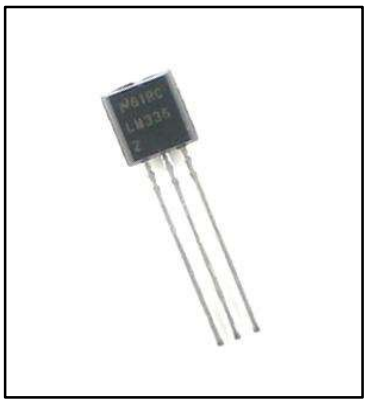
➢ Specifications:

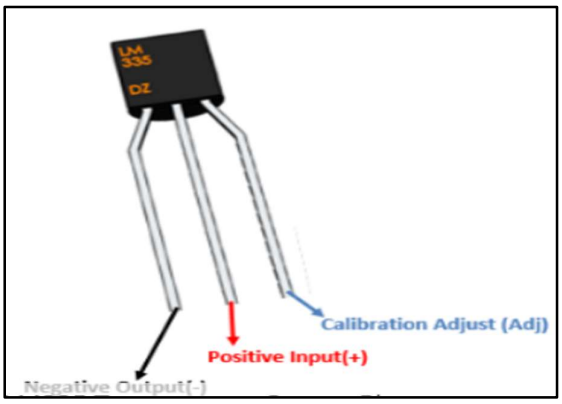
- ✓ **Operating Voltage:** 3.3V to 5V DC
- ✓ **Operating Current:** 15mA
- ✓ Output Digital - 0V to 5V, Adjustable trigger level from preset
- ✓ Output Analog - 0V to 5V based on infrared radiation from fire flame falling on the sensor
- ✓ LEDs indicating output and power
- ✓ **PCB Size:** 3.2cm x 1.4cm
- ✓ LM393 based design
- ✓ Easy to use with Microcontrollers or even with normal Digital/Analog IC
- ✓ Small, cheap and easily available.

➢ Applications:

- ✓ Gardening
- ✓ Irrigation Systems
- ✓ Used in Controlled Environments

E. Temperature Sensor:





➢ Pin Configuration:

Pin No.	Pin Name	Description
1	Adj	Calibration adjust pin
2	+	Positive Input/ Output pin
3	-	Negative output/ Ground

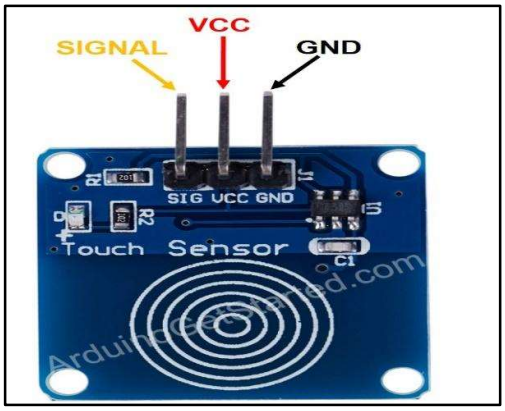
➢ Specification:

- ✓ **Reverse Current:** 15mA
- ✓ **Forward Current:** 10mA
- ✓ **Operating Output Voltage:** 2.95V to 3.01V
- ✓ **Temperature Error (at 25°C):** 2°C (max)
- ✓ **Thermal Resistance:** 202°C/W
- ✓ **Specified Temperature Range:** -40 to 100°C
- ✓ **Storage Temperature Range:** -60 to 150°C

➢ Application:

- ✓ Power Supplies
- ✓ Battery management
- ✓ HVAC
- ✓ Appliances

F. Touch Sensor:



➢ Pinout Configuration:

Pin	Description
GND	Pin needs to be connected to GND (0V) of Controller.
VCC	Pin needs to be connected to VCC (5V or 3.3v) of Controller for Power Source.
Signal	Pin is an output pin: LOW when it is NOT touched, HIGH when it is touched. This pin needs to be connected to Arduino's input pin.

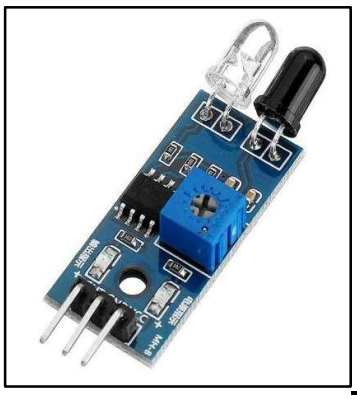
➢ Specifications

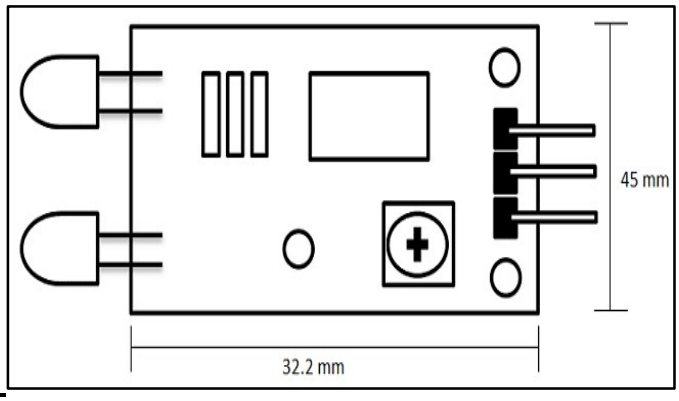
- ✓ Operating Voltage: 2.0 - 5.5V
- ✓ Operating Current(Vcc=3V):1.5 - 3.0µA
- ✓ Operating Current(VDD=3V):3.5 - 7.0µA
- ✓ Output Response Time: 60 - 220mS
- ✓ Used Chipset: TTP223-BA6

➢ Applications:

- ✓ The capacitive touch sensors can be easily manufactured with an attractive design and in less investment. Hence these are widely preferred in mobile phones, iPods, various industrial and automotive applications. In the measurement of distance, pressure, etc. these sensors are used.
- ✓ The other type of touch sensors includes resistive. It doesn't get affected due to small touching sense or contact. It needs a certain amount of force to start operating. So, this type of sensor is utilized in the keypads of musical instruments, touchpads that are resistive, etc.
- ✓ The most common application of touch sensors can be seen in water taps such that with one touch, the flow of running water can be controlled. These sensors are also utilized in home automation using Arduino or Raspberry pi projects.

G. Infrared Sensor:





➢ Pin Configuration:

Pin Name	Description
VCC	Power Supply Input
GND	Power Supply Ground
OUT	Active High Output

➢ Specifications:

- ✓ 5VDC Operating voltage
- ✓ I/O pins are 5V and 3.3V compliant
- ✓ Range: Up to 20cm
- ✓ Adjustable Sensing range

- ✓ Built-in Ambient Light Sensor
- ✓ 20mA supply current
- ✓ Mounting hole

➢ Applications:

- ✓ Obstacle Detection
- ✓ Industrial safety devices
- ✓ Wheel encoder

H. Servo Motor:

➢ Wire Configuration:

Wire Number	Wire Colour	Description
1	Brown	Ground wire connected to the ground of system
2	Red	Powers the motor typically +5V is used
3	Orange	PWM signal is given in through this wire to drive the motor

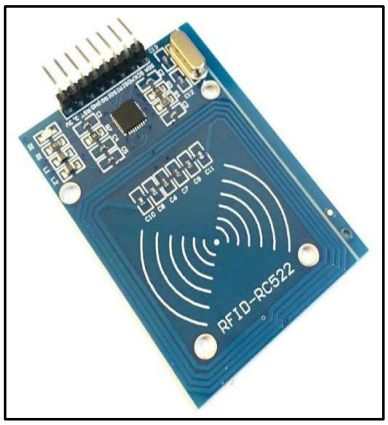
➢ Specifications:

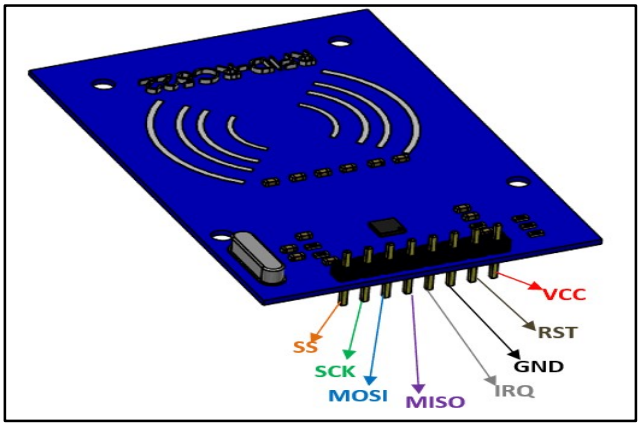
- ✓ Operating Voltage is +5V typically
- ✓ **Torque:** 2.5kg/cm
- ✓ Operating speed is 0.1s/60°
- ✓ **Gear Type:** Plastic
- ✓ **Rotation:** 0°-180°
- ✓ **Weight of motor:** 9gm
- ✓ Package includes gear horns and screws

➢ Applications

- ✓ Used as actuators in many robots like Biped Robot, Hexapod, robotic arm etc..
- ✓ Commonly used for steering system in RC toys
- ✓ Robots where position control is required without feedback
- ✓ Less weight hence used in multi DOF robots like humanoid robots

I. RFID Sensor:





➢ Pin Configuration:

Pin Number	Pin Name	Description
1	Vcc	Used to Power the module, typically 3.3V is used
2	RST	Reset pin – used to reset or power down the module
3	Ground	Connected to Ground of system
4	IRQ	Interrupt pin – used to wake up the module when a device comes into range
5	MISO/SCL/Tx	MISO pin when used for SPI communication, acts as SCL for I2c and Tx for UART.
6	MOSI	Master out slave in pin for SPI communication
7	SCK	Serial Clock pin – used to provide clock source
8	SS/SDA/Rx	Acts as Serial input (SS) for SPI communication, SDA for IIC and Rx during UART

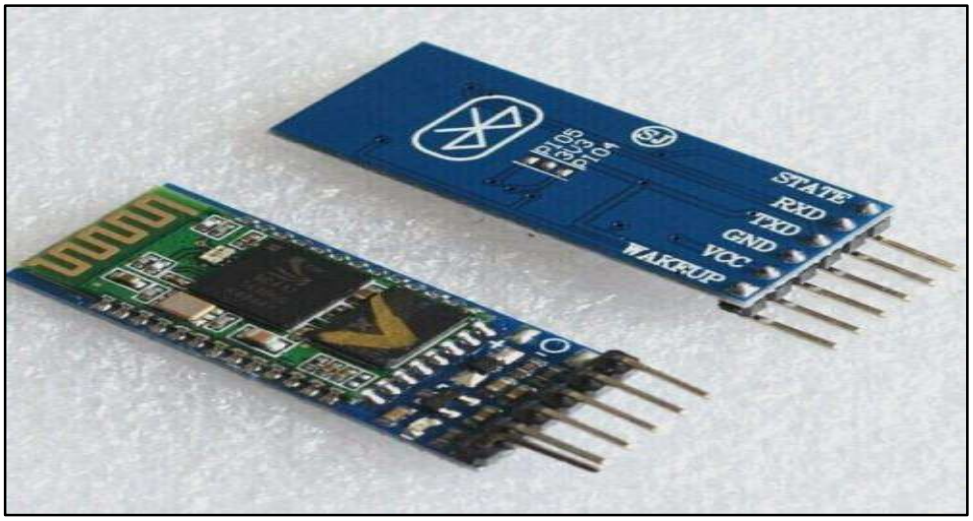
➢ Specification:

- ✓ 13.56MHz RFID module
- ✓ **Operating voltage:** 2.5V to 3.3V
- ✓ **Communication:** SPI, I2C protocol, UART
- ✓ **Maximum Data Rate:** 10Mbps
- ✓ **Read Range:** 5cm
- ✓ **Current Consumption:** 13-26mA
- ✓ **Power down mode consumption:** 10uA (min)

➢ Applications:

- ✓ Automatic billing systems
- ✓ Attendance systems
- ✓ Verification/Identification system
- ✓ Access control system.

J. Bluetooth Module:



➢ Pin Configuration:

Pin Number	Pin Name	Description
1	Enable / Key	This pin is used to toggle between Data Mode (set low) and AT command mode (set high). By default it is in Data mode

2	Vcc	Powers the module. Connect to +5V Supply voltage
3	Ground	Ground pin of module, connect to system ground.
4	TX – Transmitter	Transmits Serial Data. Everything received via Bluetooth will be given out by this pin as serial data.
5	RX – Receiver	Receive Serial Data. Every serial data given to this pin will be broadcasted via Bluetooth
6	State	The state pin is connected to on board LED, it can be used as a feedback to check if Bluetooth is working properly.
7	LED	Indicates the status of Module • Blink once in 2 sec: Module has entered Command Mode • Repeated Blinking: Waiting for connection in Data Mode • Blink twice in 1 sec: Connection successful in Data Mode
8	Button	Used to control the Key/Enable pin to toggle between Data and command Mode

➢ Specifications:

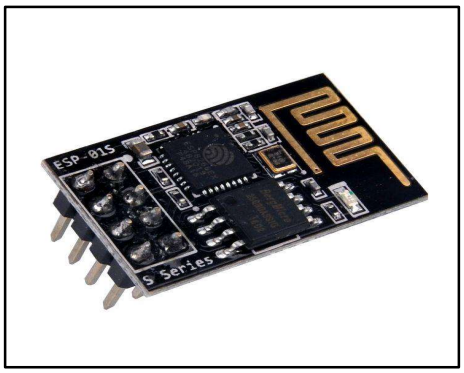
- ✓ Serial Bluetooth module for Arduino and other microcontrollers
- ✓ Operating Voltage: 4V to 6V (Typically +5V)
- ✓ Operating Current: 30mA
- ✓ Range: <100m
- ✓ Works with Serial communication (USART) and TTL compatible
- ✓ Follows IEEE 802.15.1 standardized protocol
- ✓ Uses Frequency-Hopping Spread spectrum (FHSS)
- ✓ Can operate in Master, Slave or Master/Slave mode
- ✓ Can be easily interfaced with Laptop or Mobile phones with Bluetooth
- ✓ Supported baud rate: 9600,19200,38400,57600,115200,230400,460800.

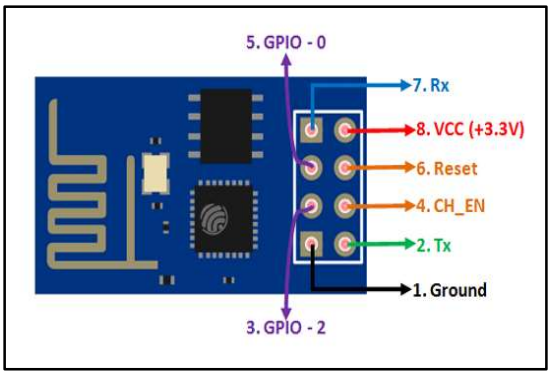
➢ Applications:

- ✓ Wireless communication between two microcontrollers
- ✓ Communicate with Laptop, Desktops and mobile phones
- ✓ Data Logging application
- ✓ Consumer applications

- ✓ Wireless Robots
- ✓ Home Automation

K. Wi-Fi Module:





➢ Pin Configuration:

Pin Number	Pin Name	Alternate Name	Normally used for	Alternate purpose
1	**Ground**	-	Connected to the ground of the circuit	-
2	**TX**	GPIO – 1	Connected to Rx pin of programmer/uC to upload program	Can act as a General purpose Input/output pin when not used as TX
3	**GPIO-2**	-	General purpose Input/output pin	-
4	**CH_EN**	-	Chip Enable – Active high	-
5	**GPIO - 0**	Flash	General purpose Input/output pin	Takes module into serial programming when held low during start up
6	**Reset**	-	Resets the module	-
7	**RX**	GPIO - 3	General purpose Input/output pin	Can act as a General purpose Input/output pin when not used as RX
8	**Vcc**	-	Connect to +3.3V only	

➢ Specifications:

- ✓ Low cost, compact and powerful Wi-Fi Module
- ✓ **Power Supply:** +3.3V only
- ✓ **Current Consumption:** 100mA
- ✓ **I/O Voltage:** 3.6V (max)
- ✓ **I/O source current:** 12mA (max)
- ✓ Built-in low power 32-bit MCU @ 80MHz
- ✓ 512kB Flash Memory
- ✓ Can be used as Station or Access Point or both combined
- ✓ Supports Deep sleep (<10uA)
- ✓ Supports serial communication hence compatible with many development platform like Arduino
- ✓ Can be programmed using Arduino IDE or AT-commands or Lua Script

➢ Applications:

- ✓ IOT Projects
- ✓ Access Point Portals
- ✓ Wireless Data logging
- ✓ Smart Home Automation
- ✓ Learn basics of networking
- ✓ Portable Electronics
- ✓ Smart bulbs and Sockets

Experiment 3

Perform Experiment using Arduino Uno to keep Buzzer/LED ON/OFF

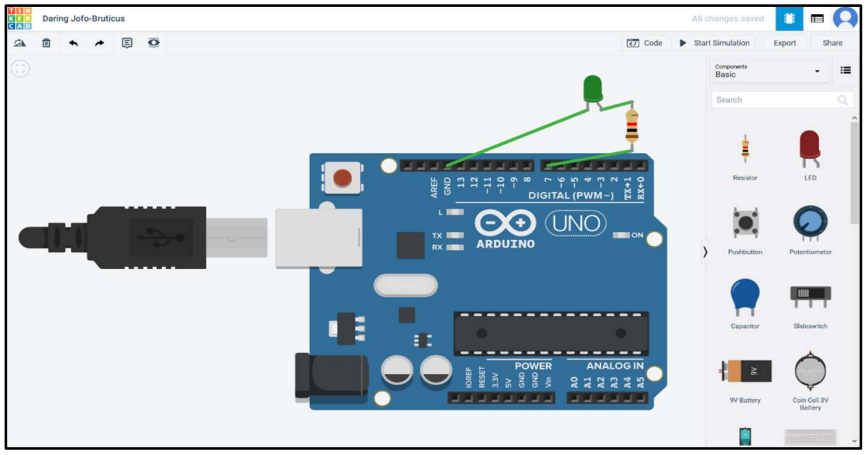
A. Required Components:

- ✓ Arduino UNO R3 Board
- ✓ LED
- ✓ Register (1 kΩ)
- ✓ Jumper Wires (Connectors)
- ✓ Breadboard (Optional)
- ✓ Simulator (https://www.tinkercad.com / Circuits)

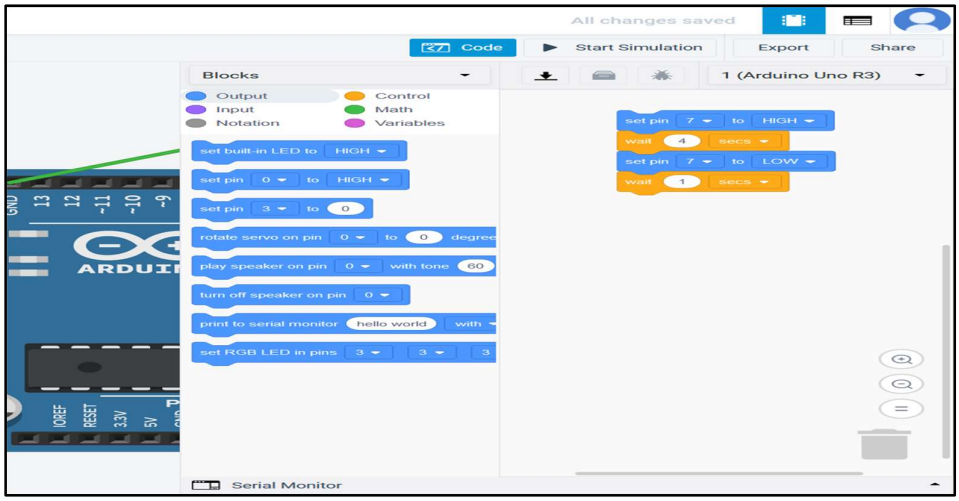
B. Procedure:

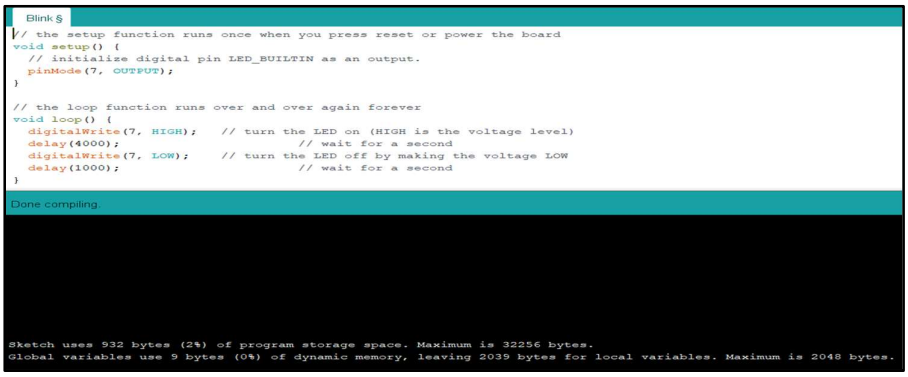
- ➢ **Step-1:**
  - ✓ Take LED and Connect Cathode pin with GND pin and Anode pin with any pin of Arduino board via register and using Jumper Wires (Here We connecting with pin 7).
  - ✓ One Can Use Breadboard too in connection instead of direct connecting with Arduino Board.

- ➢ **Step-2:**

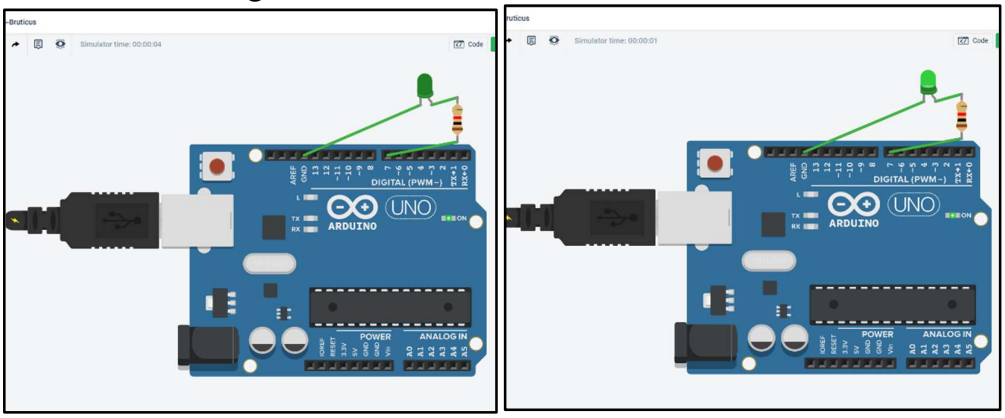


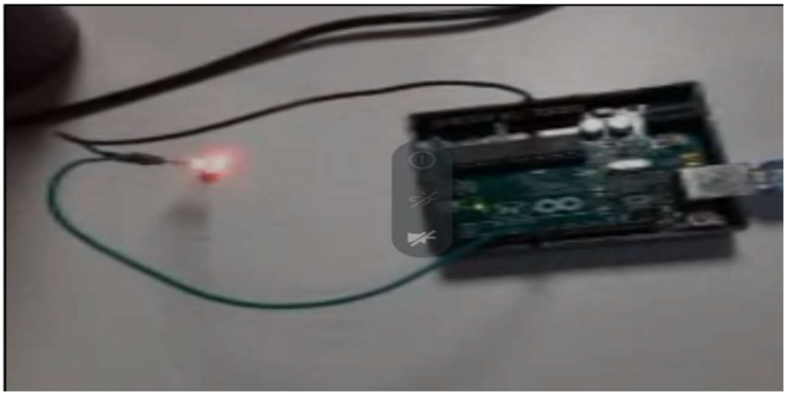
  - ✓ Prepare Code and Compile in Arduino IDE or Code Section of Simulator.
  - ✓ After Successful Compilation and No error Upload the Code in Arduino Board by Connect The board with PC.
  - ✓ Select Arduino UNO and Port in IDE if using Components not Simulator.





- **Step-3:**
  - ✓ To Run the Project, Connect Power Source with the Arduino board.
  - ✓ If Using Simulator Then Just Start Simulation.





C. Code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(7, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(7, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(4000);                       // wait for a second
  digitalWrite(7, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

D. Analysis:

- ✓ As it is very simple to implement so from this one can easily able to understand how to use Arduino as controller and implement electronic circuit using it.
- ✓ Also, from that one can understand how to implement Arduino based project using simulator & also using physical components.
- ✓ Basic idea of how to code for Arduino in Arduino IDE and upload and run code in Arduino Uno board.

E. Applications:

- **Automatic Lighting System for Residence and Public Places:**
  - ✓ By Connecting various sensors like Light Detection Sensor, we can implement system such like when level of light reach at some amount automatically Turn ON / OFF the Light.
  - ✓ Also, By Attaching Clock Module we can Turn ON / OFF lights on certain Time.
  - ✓ **Ex:** Street Lights, Lights for Home Decoration.

- **Simple Buzzer / Switch System:**
  - ✓ Also, Can Implement Simple Systems Like Buzzer System which we are using in Competition which have buzzer rounds.
  - ✓ Small System of Toys & Decoration to include functionality that when user press the button then light get turn on and off to display that operation is performed and also to create decorating lights using simple turn ON & OFF functionality of LED by creating pattern.
  - ✓ **Ex:** Buzzer System of Quiz, Series of Light for Decoration.

Experiment 4

Perform Experiment using Arduino Uno to prepare Traffic management system using LED and Buzzer

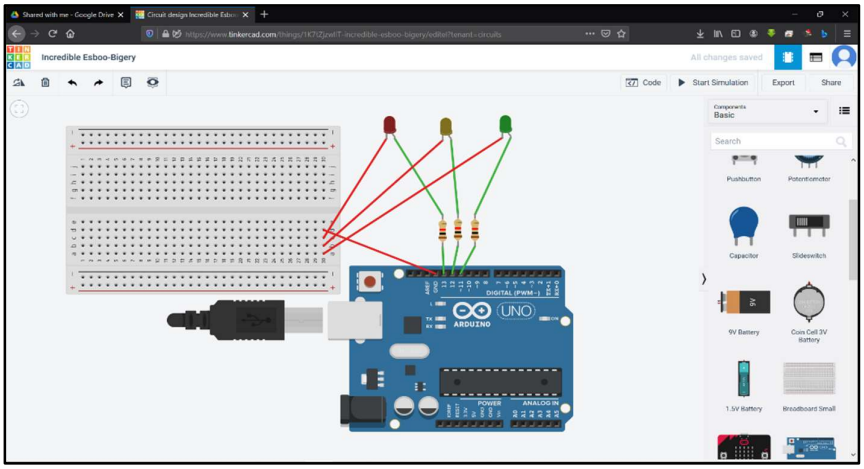
A. Required Components:

- ✓ Arduino UNO R3 Board
- ✓ LEDs (Green, Red, Yellow)
- ✓ Registers (1 kΩ each)
- ✓ Jumper Wires (Connectors)
- ✓ Breadboard (Optional)
- ✓ Simulator (https://www.tinkercad.com / Circuits)

B. Procedure:

➢ **Step-1:**

✓ Take LEDs and Connect Cathode pins of every LED with GND pin and Anode pin of LEDs (Red, Yellow, Green) with different pins of Arduino board via registers and using Jumper Wires and breadboard. (Here We connecting with Red, Yellow and Green With 13,12 and 11 Accordingly).



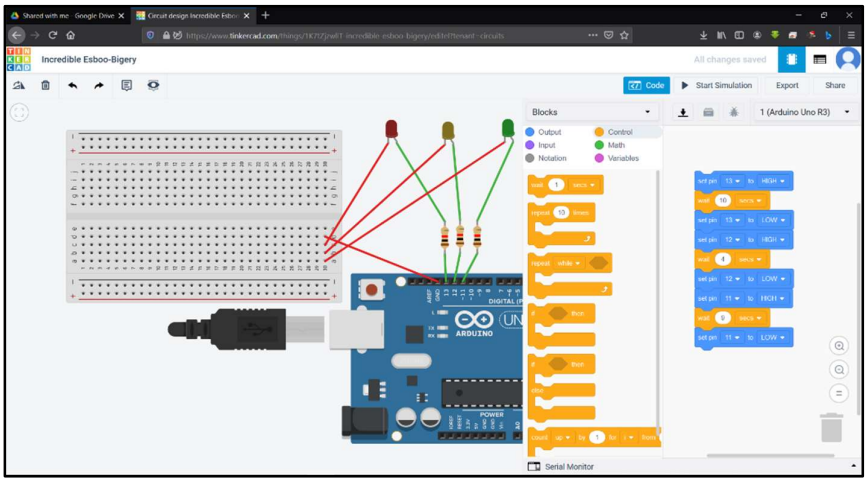
➢ **Step-2:**

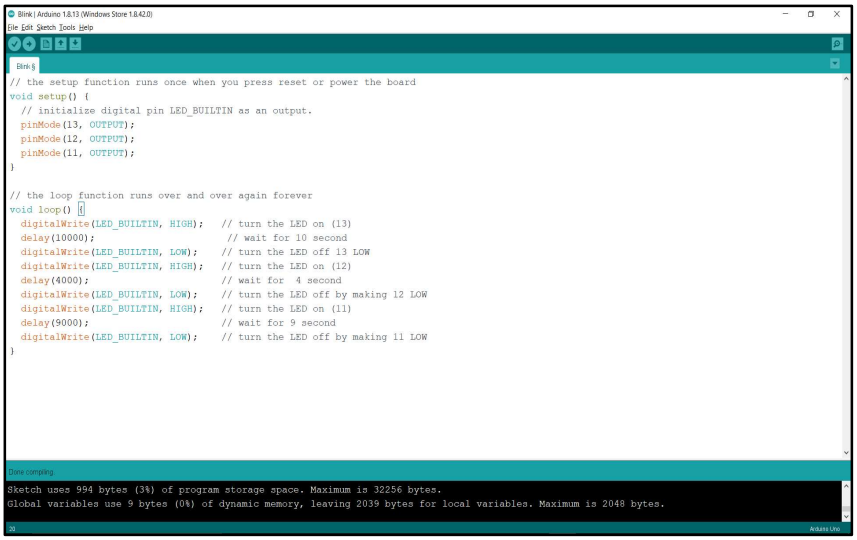
✓ Prepare Code and Compile in Arduino IDE or Code Section of Simulator.

✓ Set Three Various Delay for Red, Yellow and Green LEDs to turn ON / OFF one by one.

✓ After Successful Compilation and No error Upload the Code in Arduino Board by Connect The board with PC.

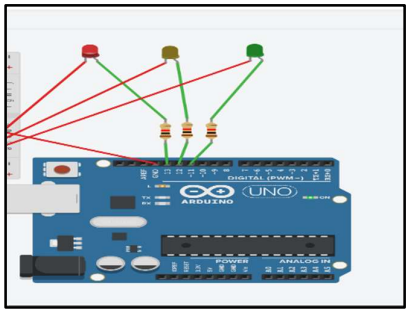
✓ Select Arduino UNO and Port in IDE if using Components not Simulator.

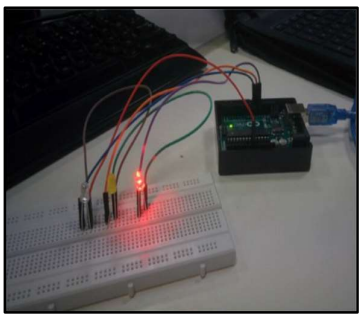




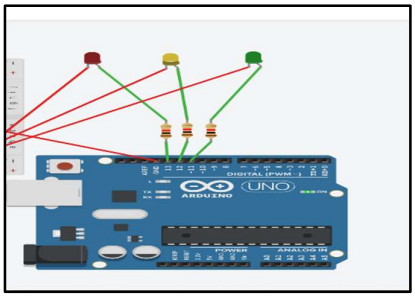
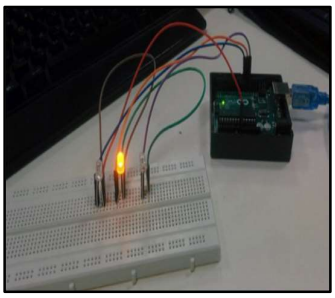
- **Step-3:**
  - ✓ To Run the Project, Connect Power Source with the Arduino board.
  - ✓ If Using Simulator Then Just Start Simulation.

Red LED Blink:

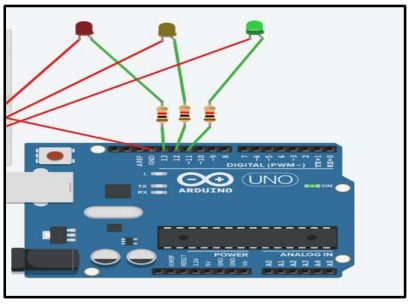
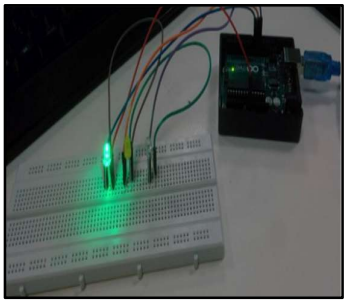




Yellow LED Blink:

Green LED Blink:

C. Code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (13)
  delay(10000);                       // wait for 10 second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off 13 LOW
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (12)
  delay(4000);                       // wait for  4 second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making 12 LOW
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (11)
  delay(9000);                       // wait for 9 second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making 11 LOW
}
```

D. Analysis:

- ✓ Here Using more than one output pin one by one by setting proper time interval, also here we want output on only one specified pin at a time so for that before set one output pin to high setting current output pin to low.
- ✓ One Can Easily learn parallel process of output as in we are parallelly controlling two pins.

E. Applications:

- ✓ Traffic Lights
- ✓ Decoration Purpose to create light series
- ✓ By Connecting various sensors based on sensors input can control multiple pins for example by connecting motion sensor can indicate speed by different colours based on speed interval. **Ex:** Speedometer of Vehicles. Can Implement such system for whether too.

Experiment 5

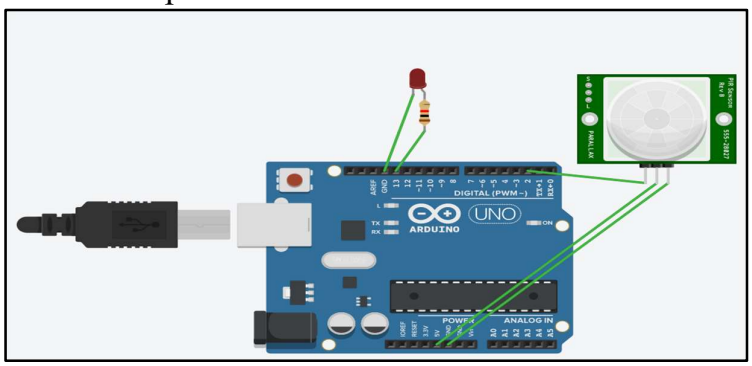
Perform Experiment using Arduino Uno to keep Buzzer/LED ON/OFF based on the moment of the object in the range of PIR Motion Sensor.

A. Required Components:

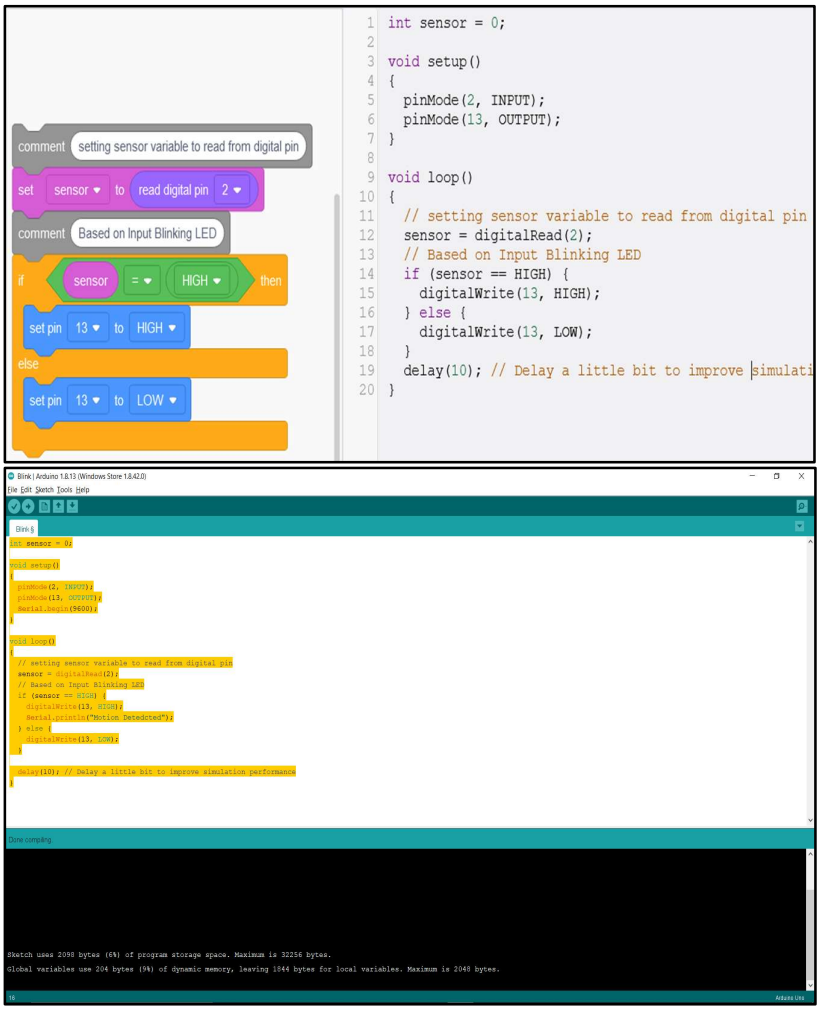
- ✓ Arduino UNO R3 Board
- ✓ LEDs (Green, Red, Yellow)
- ✓ Registers (1 kΩ each)
- ✓ Jumper Wires (Connectors)
- ✓ Breadboard (Optional)
- ✓ Simulator (https://www.tinkercad.com / Circuits).
- ✓ PIR Motion Sensor

B. Procedure:

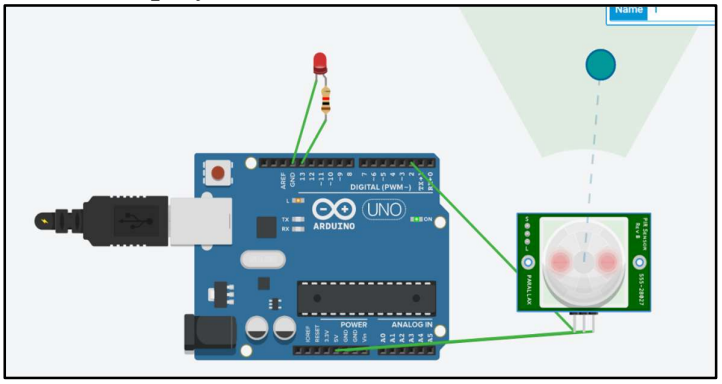
- **Step-1:**
  - ✓ Take the All Components and then one by one connect all the components.
  - ✓ Connect LED to Any Output pin and Ground using Register, Jumper wires (Directly to Board or using Breadboard.
  - ✓ Now To Connect PIR Motion Sensor Connect Signal Pin to Any Digital Input, GND to ground and Power to Power Source pin 5V.

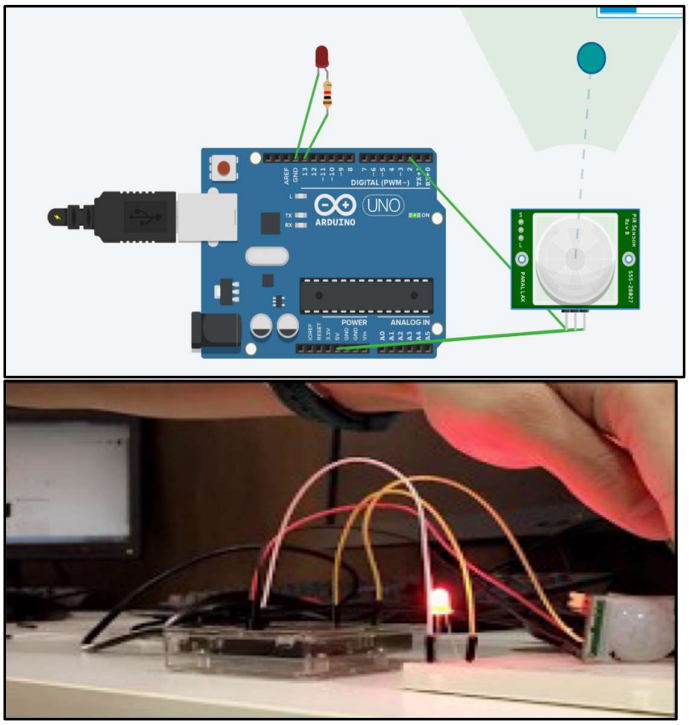


- **Step-2:**
  - ✓ Now Code and Upload that in to Arduino Board.
  - ✓ For Simulator First Create Variable and then Set Which digital will read the input.
  - ✓ Then as We Have two Options so take If else block and set led high and low condition.



- **Step-3:**
  - ✓ Now Let's Run the Code if motion detects by sensor LED Will Blink else it set to off.
  - ✓ Note: For Simulator First Click the PIR Sensor show Dot will display and Then Move That for Motion





C. Code:

✓ For Simulator:

i. Create Variable Named Sensor

ii. Take Set One from Sensor It self and in then in field where **1** is there default, take Read Digital PIN and set pin number where we connected Sensor.

iii. Take If Else and In that, In Hexagon Condition Field, Take Mathematical expression of logical Compression (<) and then in first filed take variable from variable and in second one High/Low from Math Expression.

iv. Then in Block if & else output logic of pin setting

✓ Code for Arduino IDE:

```
int sensor = 0;

void setup()

{
```

```
  pinMode(2, INPUT);
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  // setting sensor variable to read from digital pin
  sensor = digitalRead(2);
  // Based on Input Blinking LED
  if (sensor == HIGH) {
    digitalWrite(13, HIGH);
    Serial.println("Motion Detedcted");
  } else {
    digitalWrite(13, LOW);
  }
  delay(10); // Delay a little bit
}
```

D. Analysis:

- ✓ As just by use LED we can't create too much useful application but by use sensors and by taking input from that we can make useful applications like here by adding motion sensor we can detect motion and if motion is there, we can take action according there.
- ✓ We Can say it is very similar to automatic door which sense the user's entrance and based that open or close the door.

E. Applications:

- ✓ So we can use this same concept to create automatic items by detecting Motion like Automatic Door based on user's Entrance, Security System which if detect any motion then Turn on alarm or Light to warn.

Experiment 15

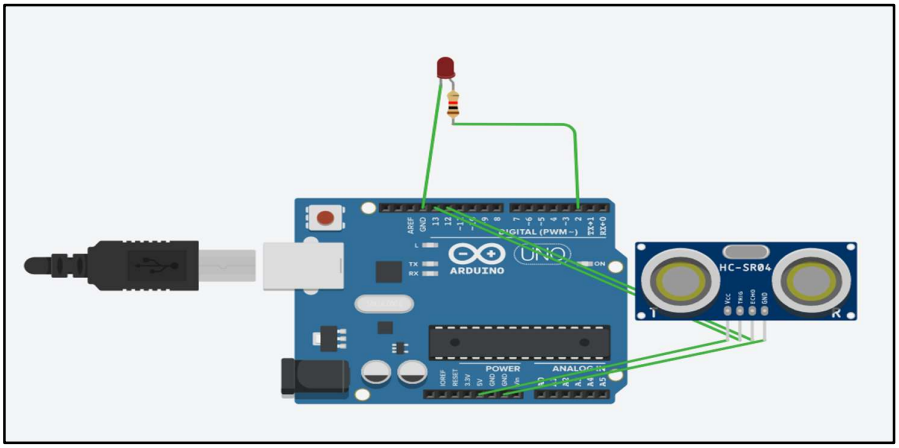
Perform Laboratory distance measuring system using Arduino Uno to keep Buzzer/LED ON/OFF based on the distance value. [using ultrasonic sensor]

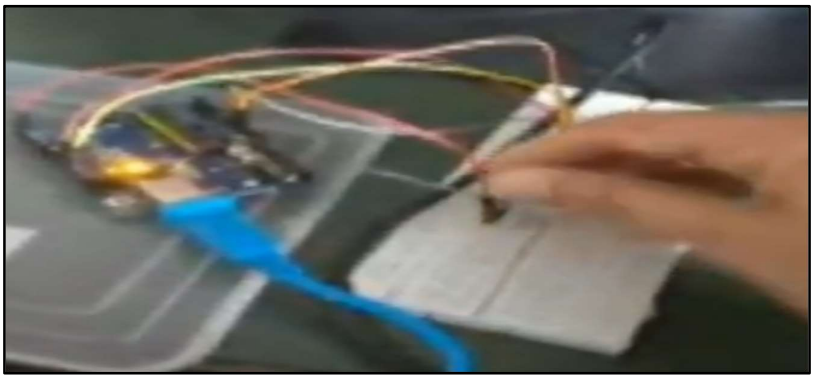
A. Required Components:

- ✓ Arduino UNO R3 Board
- ✓ LEDs (Green, Red, Yellow)
- ✓ Registers (1 kΩ each)
- ✓ Jumper Wires (Connectors)
- ✓ Breadboard (Optional)
- ✓ Simulator (https://www.tinkercad.com / Circuits).
- ✓ Ultrasonic Sensor

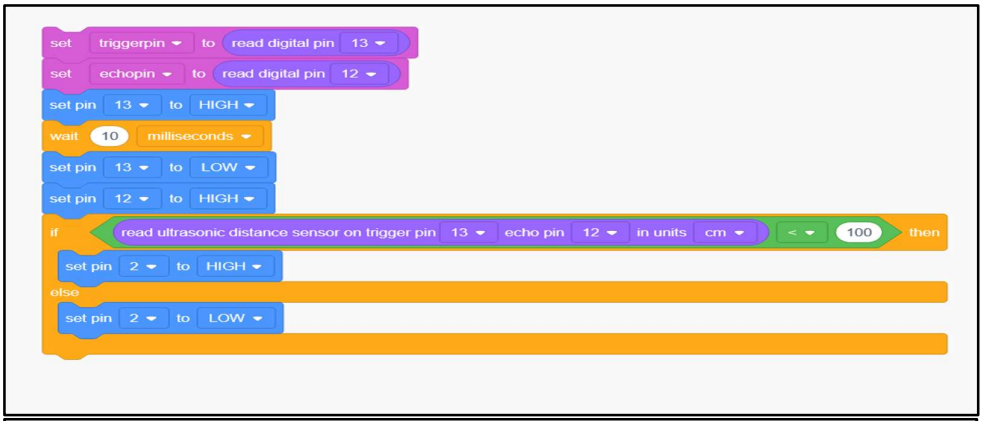
B. Procedure:

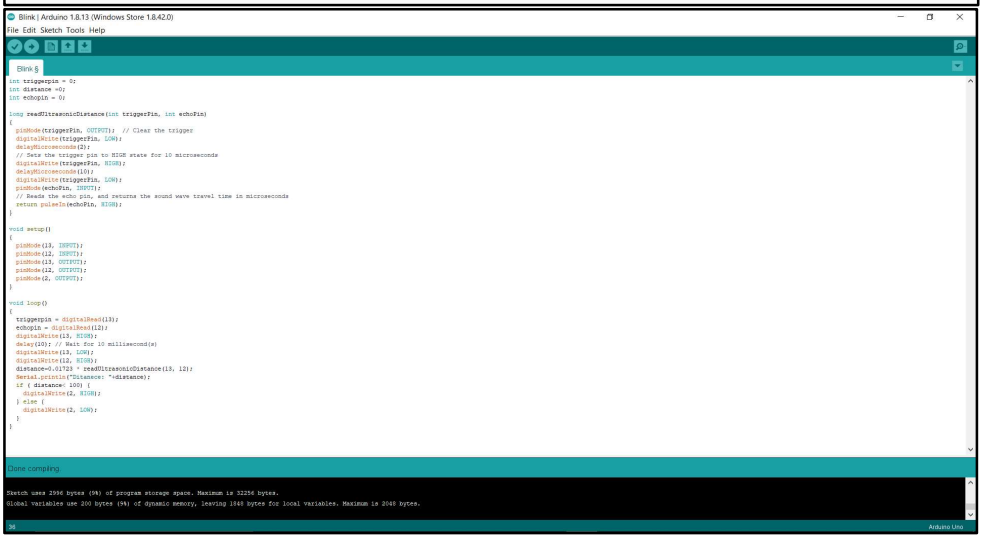
- ➢ **Step-1:**
  - ✓ Take the All Components and then one by one connect all the components.
  - ✓ Connect LED to Any Output pin and Ground using Register, Jumper wires (Directly to Board or using Breadboard.
  - ✓ Now to Connect Ultrasonic Sensor Connect Trigger and Echo Pins to Any Digital Input pins, GND to ground and VCC to Power Source pin 5V.



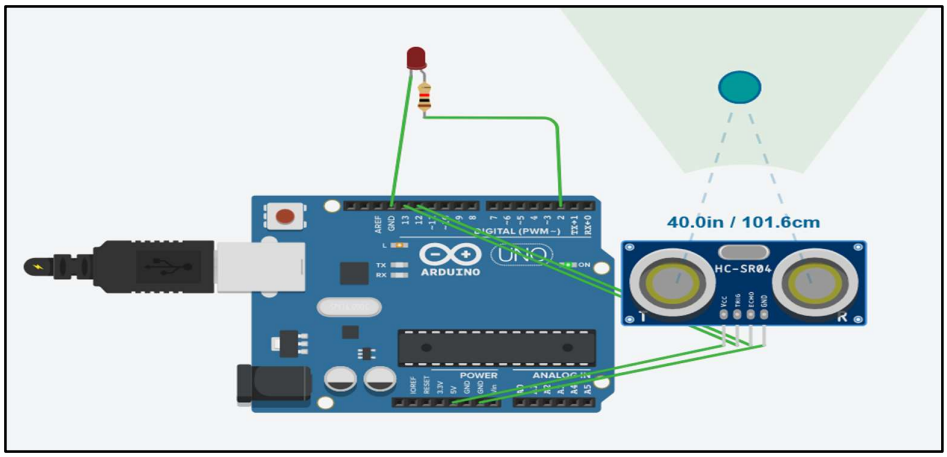


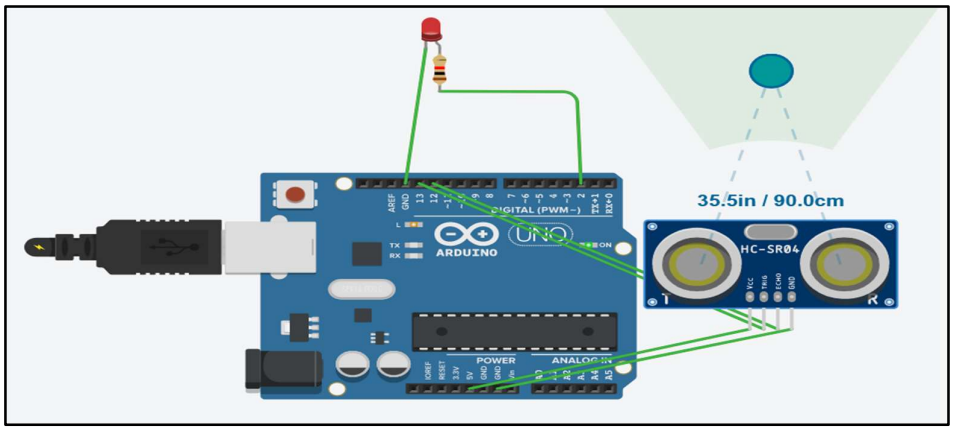
- **Step-2:**
  - ✓ Now Code and Upload that in to Arduino Board.
  - ✓ For Simulator First Create Variable and then Set Which digital will read the input.
  - ✓ Then as We Have two Options so take If else block and set led high and low condition.





- **Step-3:**
  - ✓ Now Let's Run the Code it detects distance and if distance is fulfilling condition set in code to glow LED to indicate then LED glows and as distance not fulfilling condition then led set to off.
  - ✓ Note: For Simulator First Click the Sensor show Dot will display and Then Move That for variant output.





C. Code:

✓ For Simulator:

i. Create Variables Named triggerpin and echo pin
ii. Take Set block of both the variables and set pin number in both.
iii. Set Pin (trigger connected digital pin) High
iv. Set Wait mode till how much time want to trigger signal.
v. Then set (trigger connected digital pin) Low.
vi. Set Pin (echo connected digital pin) High
vii. Then Set If Else Condition to perform task on base of distance.
viii. Take Math Block of condition checking in that set read input block (read ultrasonic ….) and in that set input pin number and unit of distance.
ix. Set condition and value.
x. If Condition True Then Set Connected LED's Pin High else false.

✓ Code for Arduino IDE:

```
int triggerpin = 0;
int distance =0;
int echopin = 0;

long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);  // Clear the trigger
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  // Sets the trigger pin to HIGH state for 10 microseconds
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
  pinMode(echoPin, INPUT);
  // Reads the echo pin, and returns the sound wave travel
time in microseconds
  return pulseIn(echoPin, HIGH);
}

void setup()
{
  pinMode(13, INPUT);
  pinMode(12, INPUT);
```

```
  pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(2, OUTPUT);
}

void loop()
{
  triggerpin = digitalRead(13);
  echopin = digitalRead(12);
  digitalWrite(13, HIGH);
  delay(10); // Wait for 10 millisecond(s)
  digitalWrite(13, LOW);
  digitalWrite(12, HIGH);
  distance=0.01723 * readUltrasonicDistance(13, 12);
  Serial.println("Ditanece: "+distance);
  if ( distance< 100) {
    digitalWrite(2, HIGH);
  } else {
    digitalWrite(2, LOW);
  }
}
```

D. Analysis:

- ✓ Motion Sensor is helpful to detect object but we cannot get distance from that so Motion sensor it helpful when we want to just detect object but when we want to perform some task based on distance of object then Ultrasonic sensor is more Helpful.

E. Applications:

- ✓ It is also helpful to create security system like if in exhibition we can use this sensor, we can set safe distance so that we are allowing users to get more details but till safe distance only of cross safe distance we can set warning system.
- ✓ Also We Can create many system where we set object at some default distance if user move object to increment distance then do something like Decrement the brightness of Light / Speed of Fan if move object and decrement the distance then increment brightness of Light / Speed of Fan