Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

MIT-WPU

Second Year B. Tech (EL & CE)

Sememester: IV **Subject:** Basic IoT Lab

Name: Shreerang Mhatre **Class:** SY

Roll No: 29 **Batch:** A2

Experiment No: 01

Name of the Experiment: To introduce various hardware platforms for IoT based design. (Example platforms are Arduino Uno /Node MCU/ Raspberry Pi/ ESP8266 / Beagle board/ Tiva / MSP430 /Jetson Nano/ Intel Galileo)

	Marks	Teacher's Signature with date
Performed on:		
Submitted on:		

Aim: To introduce various hardware platforms for IoT based design. (Example platforms are Arduino Uno /Node MCU/ Raspberry Pi/ ESP8266 / Beagle board/ CC3200 Tiva / MSP430 /Jetson Nano/ Intel Galileo)

Pre-requisite: Knowledge of Electronics and circuits, Knowledge of Assembly and C Programming language.

Objective:

1. To introduce various hardware platforms for IoT based design.
2. To understand pin configuration of various hardware platforms.
3. To compare various hardware platforms

Components and equipment required:

Arduino Uno Board, NodeMCU, Raspberry Pi Board

Theory:

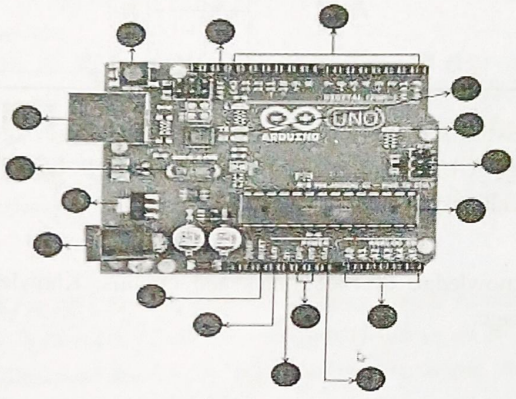
IoT hardware platforms are considered as the most significant component of the IoT environment. One IoT device can be connected to other IoT device to enable the information exchange using suitable Internet protocols. IoT platforms works as a bridge between the device sensors and data networks. It gives data connectivity to the sensor system and helps to understand using back-end applications and helps to analyse the huge data generated by sensors.

Arduino

Reference Link: https://www.tutorialspoint.com/arduino/arduino_overview.htm

Arduino is a single-board microcontroller intended to make the application more manageable which are interactive objects and its surroundings. The hardware features with an open-source hardware board designed around an 8-bit Atmel AVR microcontroller or a 32-bit Atmel ARM. Arduino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog input pins, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery.

Pin Diagram of Arduino Uno:



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 Pins

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 labelled "~" can be used to generate PWM.

16: AREF

AREF stands for Analog Reference. It is sometimes, used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

Raspberry Pi

Reference Link: https://www.electronicwings.com/raspberry-pi/raspberry-pi-introduction

Raspberry Pi is popularly used for real time Image/Video Processing, IoT based applications and Robotics applications. Raspberry Pi is slower than laptop or desktop but is still a computer which can provide all the expected features or abilities, at a low power consumption. Raspberry Pi Foundation officially provides Debian based Raspbian OS. Also, they provide NOOBS OS for Raspberry Pi. We can install several Third-Party versions of OS like Ubuntu, Arch linux, RISC OS, Windows 10 IOT Core, etc.

Raspbian OS is official Operating System available for free to use. This OS is efficiently optimized to use with Raspberry Pi. Raspbian have GUI which includes tools for Browsing, Python programming, office, games, etc. We should use SD card (minimum 8 GB recommended) to store the OS (operating System). Raspberry Pi is more than computer as it provides access to the on-chip hardware i.e. GPIOs for developing an application. By accessing GPIO, we can connect devices like LED, motors, sensors, etc and can control them too.

It has ARM based Broadcom Processor SoC along with on-chip GPU (Graphics Processing Unit). The CPU speed of Raspberry Pi varies from 700 MHz to 1.2 GHz. Also, it has on-board SDRAM that ranges from 256 MB to 1 GB. Raspberry Pi also provides on-chip SPI, I2C, I2S and UART modules.

Pin Diagram of Raspberry Pi 3 Model:

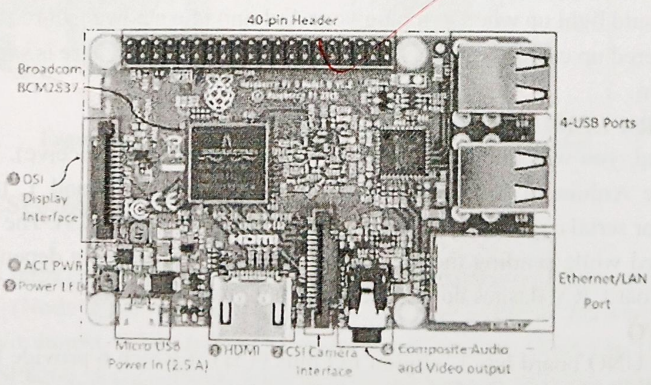


Figure 1.2: Raspberry Pi 3 Model B Hardware

1. **HDMI (High-Definition Multimedia Interface):** It is used for transmitting uncompressed video or digital audio data to the Computer Monitor, Digital TV, etc. Generally, this HDMI port helps to connect Raspberry Pi to the Digital television.
2. **CSI Camera Interface:** CSI (Camera Serial Interface) interface provides a connection in between Broadcom Processor and Pi camera. This interface provides electrical connections between two devices.
3. **DSI Display Interface:** DSI (Display Serial Interface) Display Interface is used for connecting LCD to the Raspberry Pi using 15-pin ribbon cable. DSI provides fast High-resolution display interface specifically used for sending video data directly from GPU to the LCD display.
4. **Composite Video and Audio Output:** The composite Video and Audio output port carries video along with audio signal to the Audio/Video systems.
5. **Power LED:** It is a RED coloured LED which is used for Power indication. This LED will turn ON when Power is connected to the Raspberry Pi. It is connected to 5V directly and will start blinking whenever the supply voltage drops below 4.63V.
6. **ACT PWR:** ACT PWR is Green LED which shows the SD card activity.

GPIO Pinout of Raspberry Pi 3 Model:

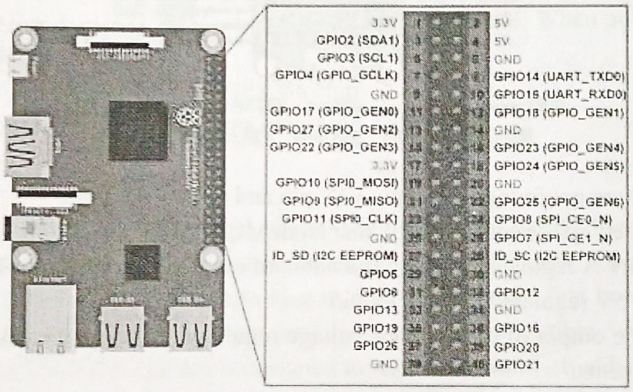


Figure 1.3: Raspberry Pi 3 Model B GPIO Pinout

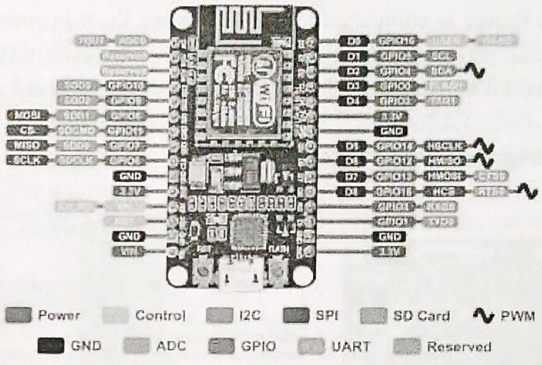
- **Node MCU ESP8266**

Reference Link: https://www.electronicwings.com/nodemcu/introduction-to-nodemcu

The NodeMCU (Node MicroController Unit) is an open-source software and hardware development environment built around an inexpensive System-on-a-Chip (SoC) called the ESP8266. The ESP8266, designed and manufactured by Espressif Systems, contains the crucial

elements of a computer: CPU, RAM, networking (WiFi), and even a modern operating system and SDK. That makes it an excellent choice for the Internet of Things (IoT) projects of all kinds. But, what about Arduino? The Arduino project created an open-source hardware design and software SDK for their versatile IoT controller. Similar to NodeMCU, the Arduino hardware is a microcontroller board with a USB connector, LED lights, and standard data pins. It also defines standard interfaces to interact with sensors or other boards. But unlike NodeMCU, the Arduino board can have different types of CPU chips (typically an ARM or Intel x86 chip) with memory chips, and a variety of programming environments. There is an Arduino reference design for the ESP8266 chip as well. However, the flexibility of Arduino also means significant variations across different vendors. For example, most Arduino boards do not have WiFi capabilities, and some even have a serial data port instead of a USB port.

Pin Diagram of NodeMCU:



Power Pins: There are four power pins. VIN pin and three 3.3V pins.

- VIN can be used to directly supply the NodeMCU/ESP8266 and its peripherals. Power delivered on VIN is regulated through the onboard regulator on the NodeMCU module – you can also supply 5V regulated to the VIN pin
- 3.3V pins are the output of the onboard voltage regulator and can be used to supply power to external components.

I2C Pins: are used to connect I2C sensors and peripherals. Both I2C Master and I2C Slave are supported. I2C interface functionality can be realized programmatically, and the clock frequency is 100 kHz at a maximum. It should be noted that I2C clock frequency should be higher than the slowest clock frequency of the slave device.

GPIO Pins: NodeMCU/ESP8266 has 17 GPIO pins which can be assigned to functions such as I2C, I2S, UART, PWM, IR Remote Control, LED Light and Button programmatically. Each digital enabled GPIO can be configured to internal pull-up or pull-down, or set to high impedance. When configured as an input, it can also be set to edge-trigger or level-trigger to generate CPU interrupts.

ADC Channel: The NodeMCU is embedded with a 10-bit precision SAR ADC. The two functions can be implemented using ADC. Testing power supply voltage of VDD3P3 pin and testing input voltage of TOUT pin. However, they cannot be implemented at the same time.

UART Pins: NodeMCU/ESP8266 has 2 UART interfaces (UART0 and UART1) which provide asynchronous communication (RS232 and RS485), and can communicate at up to 4.5 Mbps. UART0 (TXD0, RXD0, RST0 & CTS0 pins) can be used for communication. However, UART1 (TXD1 pin) features only data transmit signal so, it is usually used for printing log.

SPI Pins: NodeMCU/ESP8266 features two SPIs (SPI and HSPI) in slave and master modes. These SPIs also support the following general-purpose SPI features:

- 4 timing modes of the SPI format transfer
- Up to 80 MHz and the divided clocks of 80 MHz
- Up to 64-Byte FIFO

SDIO Pins: NodeMCU/ESP8266 features Secure Digital Input/Output Interface (SDIO) which is used to directly interface SD cards. 4-bit 25 MHz SDIO v1.1 and 4-bit 50 MHz SDIO v2.0 are supported.

PWM Pins: The board has 4 channels of Pulse Width Modulation (PWM). The PWM output can be implemented programmatically and used for driving digital motors and LEDs. PWM frequency range is adjustable from 1000 µs to 10000 µs (100 Hz and 1 kHz).

Control Pins: are used to control the NodeMCU/ESP8266. These pins include Chip Enable pin (EN), Reset pin (RST) and WAKE pin.

- **EN:** The ESP8266 chip is enabled when EN pin is pulled HIGH. When pulled LOW the chip works at minimum power.
- **RST:** RST pin is used to reset the ESP8266 chip.
- **WAKE:** Wake pin is used to wake the chip from deep-sleep.

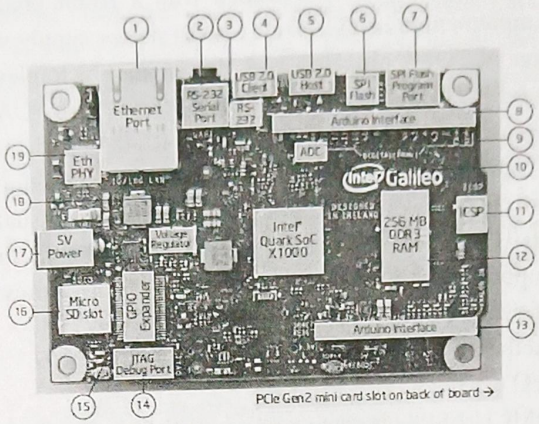
Intel Galileo

Reference Link: https://www.arduino.cc/en/ArduinoCertified/IntelGalileo

Intel Galileo is a Microcontroller board based on the Intel Quark SoC X1000 Application Processor, a 32-bit Intel Pentium class system on a chip. It's the first board based on an Intel architecture designed to be hardware and software pin-compatible with Arduino Shields designed for the UNO R3. Galileo is designed to support Arduino shields that operate at either 3.3V or 5V. The core operating voltage of Galileo is 3.3V. In spite of this, a jumper on the board can convert the voltage up to 5V at the I/O pins.

In addition to supporting the Arduino shield ecosystem, the Intel development board comes with many computing industry standard I/O interfaces, including ACPI Express, PCI Express, 10/100 Mbit Ethernet, Micro SD or SDHD, USB 2.0 device and EHCI/OHCI USB host ports, high-speed UART, RS-232 serial port, programmable 8 MB NOR flash and a JTAG port for easy debugging.

Pin Diagram of Arduino Intel Galileo:



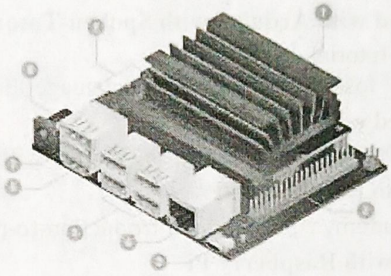
1. Ethernet Port: 10/100 Ethernet connector
2. RS-232 Serial Port: 3-pin 3.5mm jack (not audio)
3. RS-232: RS-232 transceiver
4. USB 2.0 Client: USB Client connector (Micro-USB Type B)
5. USB 2.0 Host: USB 2.0 Host connector (Micro-USB Type AB)
6. SPI Flash: 8 MByte Legacy SPI Flash to store the firmware (or bootloader) and the latest sketch.
7. SPI Flash Program Port: 7-pin header for Serial Peripheral Interface (SPI) programming
8. Shield Interface: Complies with Arduino Uno Revision 3 shield pinout
9. ADC: Analog to Digital converter
10. Intel® Quark SoC X1000: 400 MHz 32-bit Intel® Pentium processor
11. ICSP: 6-pin in-circuit serial programming (ICSP) header
12. 256 MB DDR3 RAM: 256 MByte DRAM, enabled by the firmware by default
13. Arduino Interface: Complies with Arduino Uno Revision 3 pinout
14. JTAG Debug Port: 10-pin standard JTAG header for debugging
15. GPIO Expander: GPIO pulse width modulation provided by a single I2C I/O expander
16. Micro SD slot: Supports micro SD card up to 32 GBytes (Optional)
17. 5V Power: The board is powered via an AC-to-DC adapter
18. Voltage Regulator: Generates 3.3 volt supply
19. Eth PHY: Ethernet Physical layer transceiver

- **Jetson Nano**

NVIDIA Jetson Nano is an embedded system-on-module (SoM) and developer kit from the NVIDIA Jetson family, including an integrated 128-core Maxwell GPU, quad-core ARM A57 64-bit CPU, 4GB LPDDR4 memory, along with support for MIPI CSI-2 and PCIe Gen2 high-speed I/O. There is also the Jetson Nano 2GB Developer Kit with 2GB memory and the same processing specs.

Useful for deploying computer vision and deep learning, Jetson Nano runs Linux and provides 472 GFLOPS of FP16 compute performance with 5-10W of power consumption. NVIDIA Jetson today is widely used in diverse fields such as robotics, retail, industrial, agriculture, and AIoT.

Pin Configuration of Jetson Nano:



20. microSD card slot for main storage
21. 40-pin expansion header
22. Micro-USB port for 5V power input, or for Device Mode
23. Gigabit Ethernet port
24. USB 3.0 ports (x4)
25. HDMI output port
26. DisplayPort connector
27. DC Barrel jack for 5V power input
28. MIPI CSI-2 camera connectors

Conclusion:

Thus, we have performed arduino based lighting using arduino software & coding in it.

Post Lab Questions:

1. List out different Arduino flavours with features
2. Mention different Arduino shields with their use
3. Compare various versions of Raspberry Pi with respect to different parameters
4. Different Models of the NodeMCU with Specifications
5. Brief use cases of any one IoT hardware platform

Additional links for more information:

1. **Getting started with Arduino with Spoken-Tutorial**
   https://spoken-tutorial.org/tutorial-search/?search_foss=Arduino&search_language=English
2. **Getting started with NodeMCU**
   https://www.electronicshub.org/getting-started-with-nodemcu/
3. **Introduction to Raspberry Pi**
   https://www.udemy.com/course/introduction-to-raspberry-pi-4/
4. **Building IoT with Raspberry Pi**
   https://www.blemobileapps.com/blog/building-internet-things-iot-raspberry-pi/
5. **An overview of IoT Hardware Development Platforms**
   Dhawan Singh, Amanpreet Sandhu, Aditi Thakur and Nikhil Priyanka, "An overview of IoT Hardware Development Platforms", International Journal on Emerging Technologies 11(5):155-163(2020)

* Post Lab Questions -

Q1) List out different Arduino flavours with features.

→ ① Arduino UNO:
- ATmega328P microcontroller
- 14 digital input/output pins
- 6 analog inputs
- 32 KB flash memory
- 16 MHz clock speed

② Arduino Mega:
- ATmega 2560 microcontroller
- 54 digital input/output pins
- 16 analog inputs
- 256 KB flash memory
- 16 MHz clock speed

③ Arduino Nano:
- ATmega 3828P micro controller
- 14 digital input/output pins
- 8 analog inputs
- 32 KB flash memory
- 16 MHz clock speed
- small form factor for compact projects.

Q2) Mention different Arduino sheilds with their use

→ ① Ethernet Shield:
- Allows the Arduino board to connect to the internet through an Ethernet cable.
- Used for network communication & remote data logging.

② Motor Shield:
- Controls multiple motors simoltaneously.
- Can drive DC motors, stepper motors, and servo motors.

③ LCD Shield:
- Displays text and graphics on a liquid crystal display (LCD)
- Used for displaying data from sensors, debugging, and user interfaces.

④ wifi Shield:
- Allows the Arduino board to connect to a wi-fi network
- Used for internet connectivity, cloud communication, and IoT projects.

Q3) Compare various versions of Raspberry Pi with respect to different parameters.

-> ① Processing Power:-
- Raspberry Pi 4 Model B: Broadcom BCM2711, Quad core Cortex-A72 (ARM v8) 64 bit SoC @ 1.5GHz
- Raspberry Pi 3 Model B+: Broadcom BCM2837B0, Cortex A53 (ARM v8) 64 bit SoC @ 1.4GHz

② Memory:-
- Raspberry Pi 4 Model B: 2GB, 4GB, or 8GB LPDDR4-3200 SDRAM
- Raspberry Pi 3 Model B+: 1GB LPDDR2 SDRAM

③ Connectivity:
- Raspberry Pi 4 Model B: Gigabit Ethernet, dual-band 802.11ac wireless, Bluetooth 5.0, BLE, 2 USB 3.0 ports, 2 USB 2.0 ports.
- Raspberry Pi 3 Model B+: Gigabit Ethernet, dual-band 82.11ac wireless, Bluetooth 4.2/BLE, and 4 USB 2.0 ports.

Q4) different models of the Node MCU with specifications

→ ① Node MCU v0.9:
- ESP 8266 Ex SoC
- 80 MHz clock speed
- 32MB flash memory
- GPIO, PWM, I2C, SPI, UART
- Micro USB port for power & programming.

② Node MCU v1.0:
- ESP 8266 EX SoC
- 80 MHz clock speed
- 32MB flash memory
- Micro USB port for power & pg

③ Node MCU ESP-12E:
- ESP 8266 EX SoC
- 80 MHz clock speed
- 4MB flash memory
- wifi 802.11 b/g/n
- GPIO, PWM, I2C, SPI, UART
- Micro USB port for power
- Compatible with Lua programming Language.

Q5) Brief use cases of any one IoT hardware plat form.

→ One popular IoT hardware platform is the Raspberry Pi. Here are some common use cases for the Raspberry Pi.

1. Media center: The Raspberry Pi can be used to build a low-cost, low power media center for streaming video, music, & photos.
2. Retro gaming: The Raspberry Pi can be used to play classic video games from old gamming consoles like the Nintendo Entertainment system, Super Nintendo & Sega Genesis.
3. Home automation: The Raspberry Pi can be used to control smart home devices, such as lights, thermostats, and security, systems, using programs and scripts written in Python
4. Robotics: The Raspberry Pi can be used to control robots and other physical devices, such as robots and other physical devices, hobby projects & industrial applications.

9/12/23.

Second Year B. Tech (EL&CE)

Semester : IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY**

Roll No: 29 **Batch: A2**

Experiment No: 02

Name of the Experiment: Understanding Arduino IDE and Interfacing Basic Sensors with hardware platforms.

Performed on:

Submitted on:

Marks	Teacher's Signature with date

Aim: Understanding Arduino IDE and Interfacing Basic Sensors with hardware platforms.

Prerequisite: Basic knowledge of sensors, Layout of Arduino Uno board

Objective:

1. To understand Arduino IDE
2. To understand sensor specification and working and applications
3. To sense the physical quantity using sensor and interface with Arduino Uno

Components and equipment required:

Arduino Uno Model, LED, Resistors, Sensors, LEDs, USB Cable, Breadboard etc.

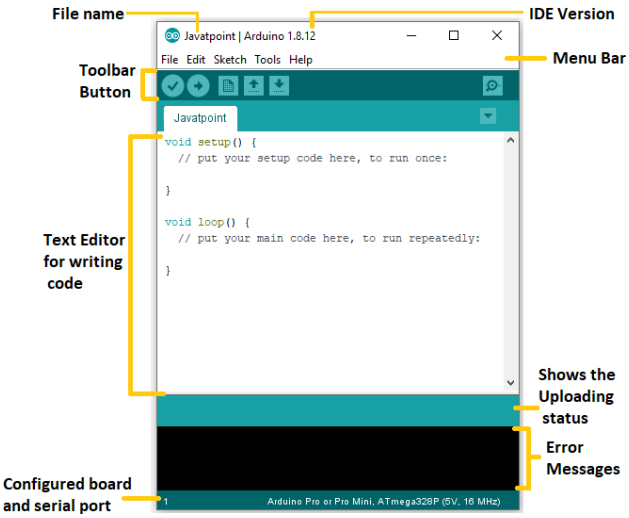
Theory:

The Arduino is an open source microcontroller development platform paired with an intuitive Programming language called as Arduino integrated development environment (IDE). As it is open source hardware, all the design files, schematics and source code are freely available to everybody.

Arduino IDE Basics:

Reference Link: https://www.javatpoint.com/arduino-ide

IDE stands for Integrated Development Environment - An official software introduced by Arduino.cc, that is mainly used for writing, compiling and uploading the code in the Arduino Device. Almost all Arduino modules are compatible with this software that is an open source and is readily available to install and start compiling the code on the go.

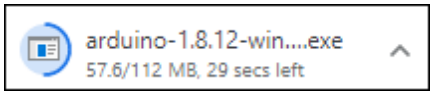


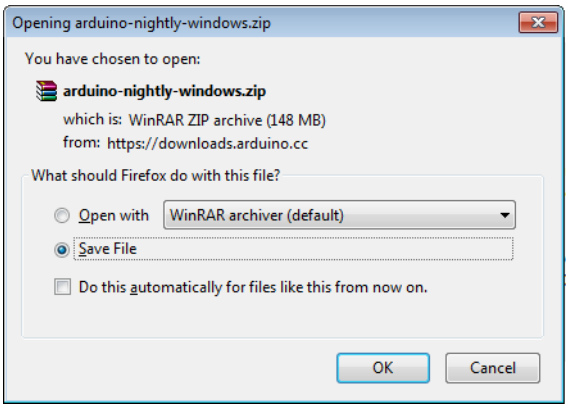
The main code, also known as a sketch, created on the IDE platform will ultimately generate a Hex File which is then transferred and uploaded in the controller on the board. The IDE environment mainly contains two basic parts: Editor and Compiler where former is used for writing the required code and later is used for compiling and uploading the code into the given Arduino Module. This environment supports both C and C++ languages.

Installation of Arduino IDE:

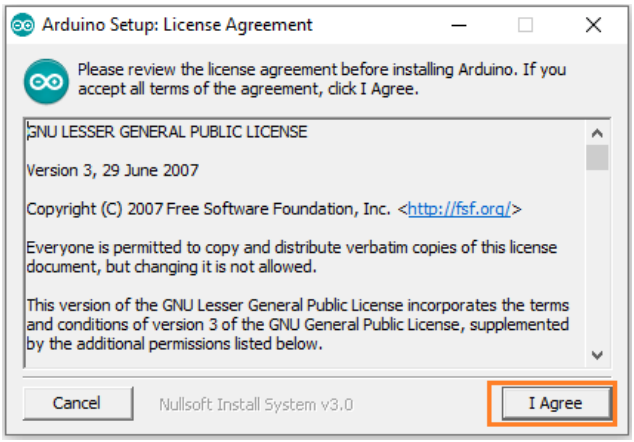
1. Go to the official website of Arduino (https://www.arduino.cc/) > Click on **SOFTWARE** < click on **DOWNLOADS**, as shown below:
2. Select option to download for Windows

3. Select Just Download option, the downloading process will start. The downloading file will look like the below image:





4. After completion of downloading process, open the downloaded files then Accept the license by clicking on '**I Agree**' button, as shown below

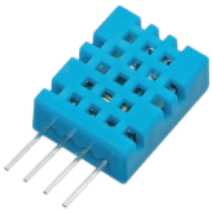


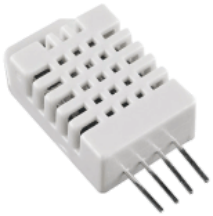
5. Complete the installation of process by selecting install options for various drivers.
6. The Arduino IDE software will appear on your desktop, as shown below:

7. After selecting this icon the Arduino IDE window will appear on which we can write the program

Temperature and Humidity sensor interfacing with Arduino

DHT11 sensor measures and provides humidity and temperature values serially over a single wire. It can measure relative humidity in percentage (20 to 90% RH) and temperature in degree Celsius in the range of 0 to 50°C. It has 4 pins; one of which is used for data communication in serial form. Pulses of different TON and TOFF are decoded as logic 1 or logic 0 or start pulse or end of a frame. DHT11 is a Digital Sensor consisting of two different sensors in a single package. The sensor contains an NTC (Negative Temperature Coefficient) Temperature Sensor, a Resistive-type Humidity Sensor and an 8-bit Microcontroller to convert the analog signals from these sensors and produce a Digital Output.

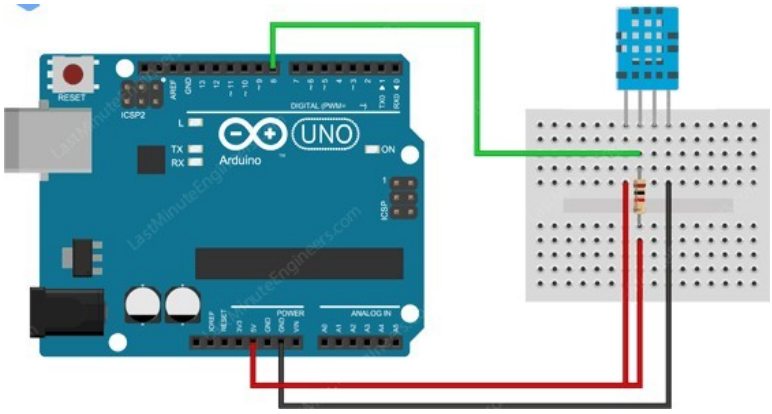




DHT11		DHT22
0 - 50°C / ± 2°C	*Temperature Range*	-40 - 125 °C / ± 0.5 °C
20 - 80% / ± 5%	*Humidity Range*	0 - 100 % / ± 2-5%
1Hz (one reading every second)	*Sampling Rate*	0.5 Hz (one reading every two seconds)
15.5mm x 12mm x 5.5mm	*Body Size*	15.1mm x 25mm x 7.7mm
3 - 5V	*Operating Voltage*	3 - 5V
2.5mA	*Max Current During Measuring*	2.5mA

The DHT22 is the more expensive version which obviously has better specifications. Its temperature measuring range is from -40 to +125 degrees Celsius with +-0.5 degrees accuracy, while the DHT11 temperature range is from 0 to 50 degrees Celsius with +-2 degrees accuracy. Also the DHT22 sensor has better humidity measuring range, from 0 to 100% with 2-5% accuracy, while the DHT11 humidity range is from 20 to 80% with 5% accuracy.
Sampling rate for the DHT11 is 1Hz or one reading every second, while the DHT22 sampling rate is 0.5Hz or one reading every two seconds and also the DHT11 has smaller body size. The operating voltage of both sensors is from 3 to 5 volts, while the max current used when measuring is 2.5mA.

- **Interfacing Diagram**



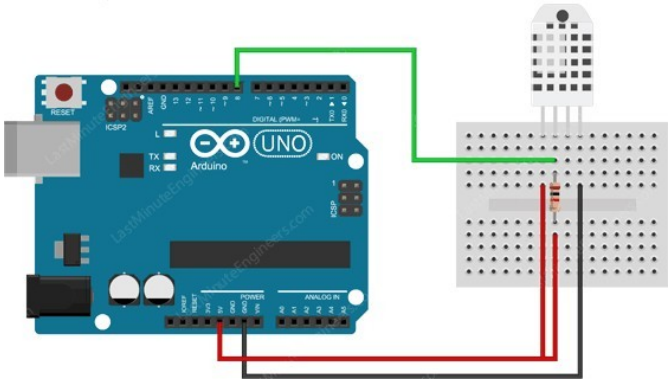


Figure 2.2: DHT11 and DHT 22 Sensor interfacing with Arduino Uno

Procedure:

1. Connect the DHT 11/DHT 22 with Arduino Uno board as per the given connection diagram
2. Write program in Arduino IDE
3. Build the program and Run.
4. Check output in terms of temperature and humidity values on Serial monitor.

Conclusion:

Post Lab Questions:

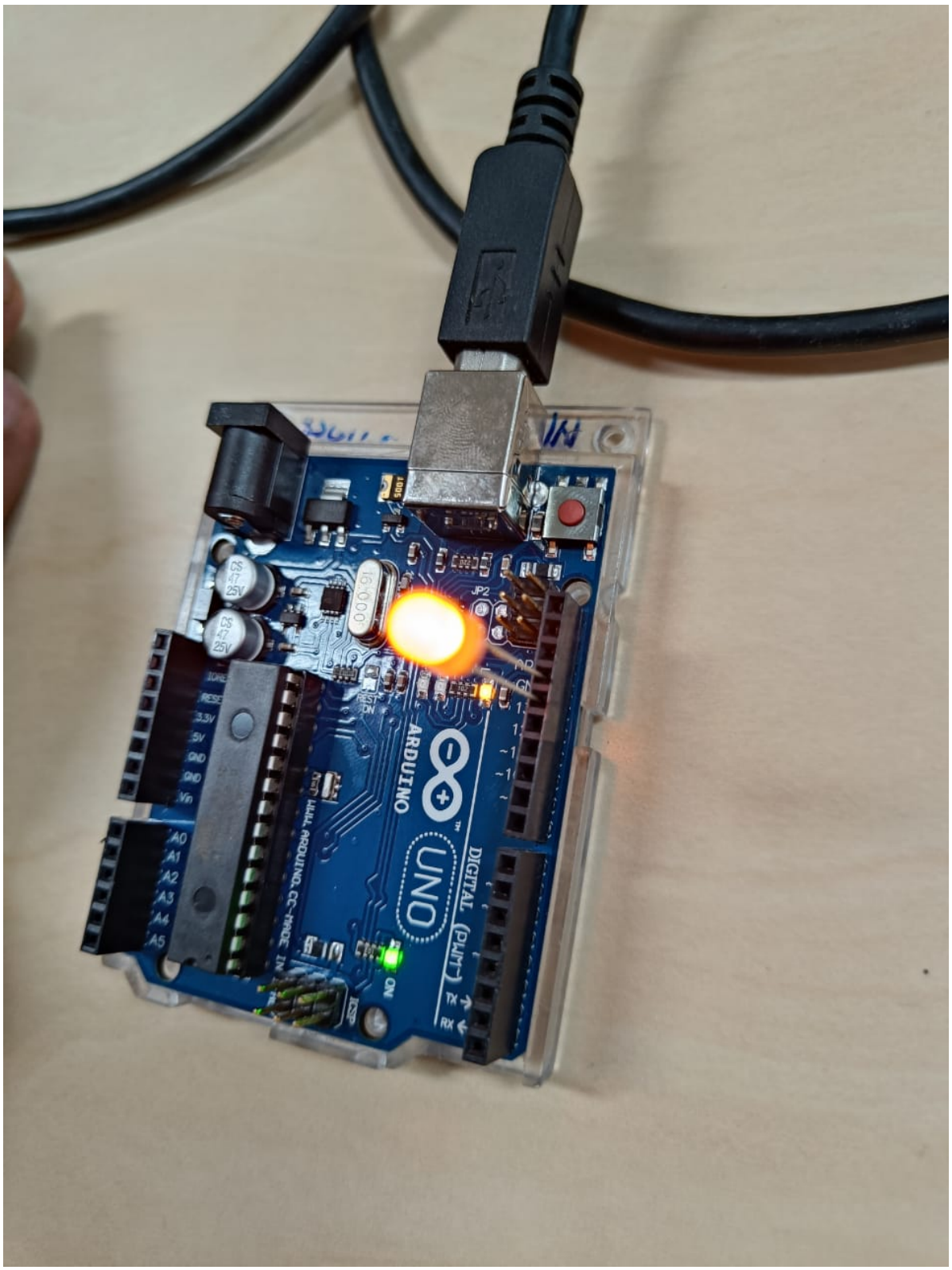
1. List out various analog sensors and explain any one in brief
2. List out various digital sensors and explain any one in brief
3. State the applications of any one sensors
4. Explain the commonly used command of Arduino IDE

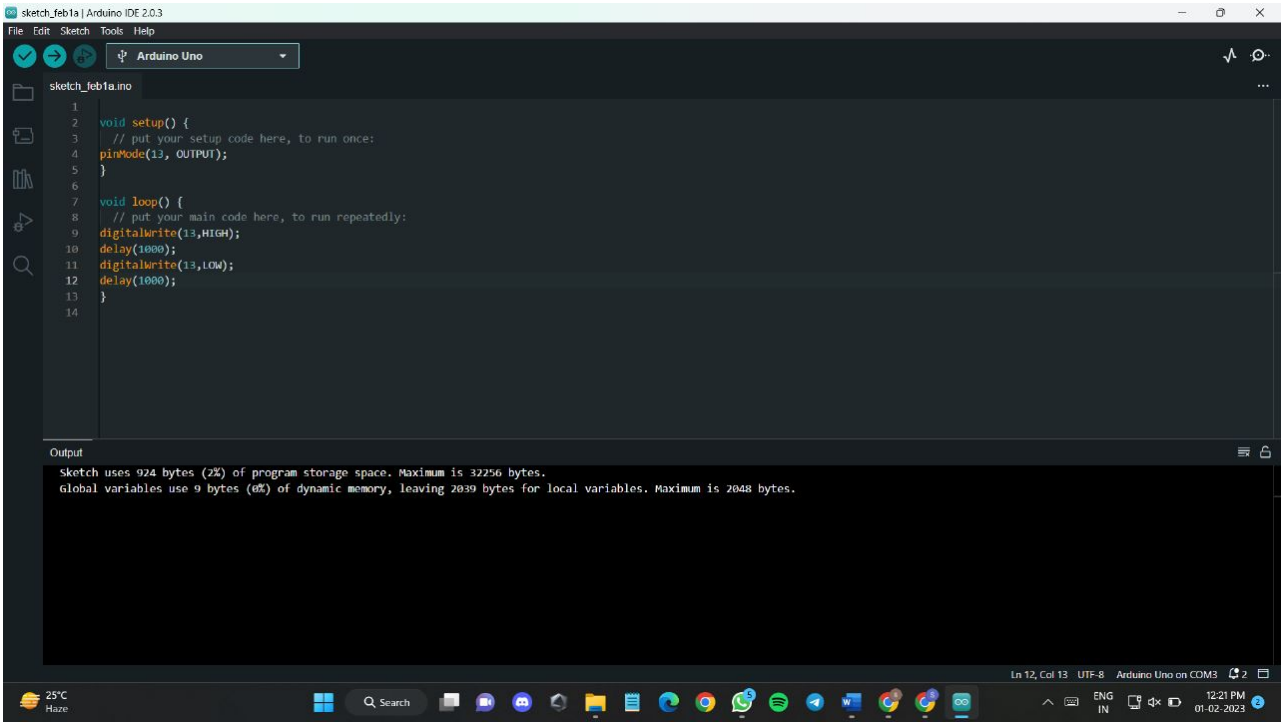
Additional links for more information:

1. **Getting started with Arduino with Spoken-Tutorial**
   https://spoken-tutorial.org/tutorial-

search/?search_foss=Arduino&search_language=English

2. **Using Sensors with Arduino-Arduino Project Hub**
https://create.arduino.cc/projecthub/JANAK13/using-sensors-with-arduino-eab1ec


ARDUINO
UNO
DIGITAL (PWM~)
POWER
ANALOG IN


sketch_feb1a | Arduino IDE 2.0.3
File Edit Sketch Tools Help
Arduino Uno
sketch_feb1a.ino
void setup() {
// put your setup code here, to run once:
pinMode(13, OUTPUT);
}
void loop() {
// put your main code here, to run repeatedly:
digitalWrite(13,HIGH);
delay(1000);
digitalWrite(13,LOW);
delay(1000);
}
Output
Sketch uses 924 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables. Maximum is 2048 bytes.
Ln 12, Col 13 UTF-8 Arduino Uno on COM3
25°C Haze
Search
ENG IN
12:21 PM 01-02-2023

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

Exp no. 2

* ~~Procedure~~ Post Lab Questions -

Q1. List out various analog sensors and explain any one in brief

→ Analog sensors are sensors that generate an output signal that varies continously in response to change in the input parameter being measured-
Some analog sensors are -
1) Temperature sensors
2) Pressure sensors
3) Light sensors
4) Accelerometers
5) Strain gauges
6) Proximity sensors.

· Thermistor - A thermistor is a type of temp sensor that uses the change in electrical resistance of a material to measure temperature. As the temperature of the thermistor increases, its electrical resistance decreases, and vice versa. This change in resistance can be measured using a simple voltage divider circuit, where the thermistor ~~ther~~ is placed in series with a known resistance.

Q2) List out varias digital sensors and explain any one in brief.

→ Digital sensors, unlike analog sensors, provide a discrete output signal that is either a 0 or 1.

Some digital sensors are -

1) Motion sensors
2) Light sensors
3) Proximity sensors
4) Magnetic sensors
5) Humidity sensors
6) Gas sensors.

Passive Infrared motion sensor -

This sensor detects the presence of a person or an object by measuring the infrared radiation emitted by them. The PIR sensor consits of a pyroelectric sensor and a Fresnel lens, which are both sensitive to infrared radiation. When an object moves within the field, ~~of~~ views of the sensor it causes a change in the infrared radiation detected by the pyroelectric sensor.

Q3) State the applications of any one sensors.

→ Accelerometer - It is used to measure acceleration or changes in velocity. Some applications are -

(1) Automotive industry -
Accelerometers are used in cars and other vehicles to detect changes in acceleration and determine the orientation.

(2) Aerospace industry -
Accelerometers are used in aircraft, rockets and satellites to measure changes in acceleration & determine the orientation of aircraft.

(3) Industrial automation:-
Accelerometers are used in industrial automation systems to monitor the vibration levels of machinery.

(4) Consumer electronics -
Accelerometers are used in smartphones, tables and other portable electronic devices to detect changes in orientation and adjust the display accordingly.

Q4) Explain the commonly used command of Arduino IDE.

→ Arduino IDE is an integrated development environment that is used to program and upload code to Arduino boards.

① Verify / compile -
This command is used to check the code for any syntax errors and compile it into a machine-readable format.

② Upload -
This command is used to upload the compiled code to the Arduino board.

③ Serial Monitor -
This command opens a serial communication window that allows you to send and receive data between the Arduino board and the computer.

④ New sketch -
This command opens a new sketch window where you can write your code.

Second Year B. Tech (EL & CE)

Semester: IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY**

Roll No: 29 **Batch: A2**

Experiment No: 03

Name of the Experiment: Interfacing stepper motor/DC Motor and relay with hardware platforms.

Performed on:

Submitted on:

Marks	Teacher's Signature with date

Aim: Interfacing of Actuators with hardware platforms

Prerequisite: Arduino board, Arduino IDE, Basics of Actuators.

Objectives:

1. To understand usefulness of Actuators
2. To understand Actuators specification, working and applications
3. To get the output using actuator and interface with Arduino Uno
4. To understand and experience PWM concept with Arduino Uno.

Components and equipment required:

Arduino Uno Board, USB cable, Arduino IDE, 5V DC motor, 270Ω Resistor, BC548 NPN transistor, 1N4001 Diode, Bread board, Jumper wires, 10k Ω Potentiometer, Push button and L293D Motor Driver IC, etc.

Theory:

DC motors are electric motors that are powered by direct current (DC), such as from a battery or DC power supply. Their commutation can be brushed or brushless. The speed of a brushed DC motor can be controlled by changing the voltage alone. This makes DC motors better suited for equipment ranging from 12VDC systems in automobiles to conveyor motors, both which require fine speed control for a range of speeds above and below the rated speeds.

DC motor basically consists of two main parts. The rotating part is called the rotor and the stationary part is also called the stator. The rotor rotates with respect to the stator.

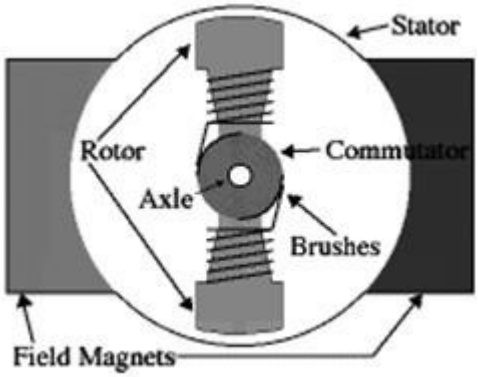


Figure 3.1: DC motor construction

The armature coil is the rotating part whereas the stationary part is the stator. In this, the armature coil is connected toward the DC supply which includes the brushes as well as the commutators. The main function of the commutator is to convert the AC to DC which is induced in the armature. The flow of current can be supplied by using the brush from the motor's rotary part toward the inactive outside load. The arrangement of the armature can be done in between the two poles of the electromagnet or permanent.

Working Principle

An electrical machine that is used to convert the energy from electrical to mechanical is known as a DC motor. The DC motor working principle is that when a current-carrying conductor is located within the magnetic field, then it experiences a mechanical force. This force direction can be decided through Flemming's left-hand rule as well as its magnitude.

Figure 3.2: DC Motor

If the first finger is extended, the second finger, as well as the left hand's thumb, will be vertical to each other & primary finger signifies the magnetic field's direction, the next finger signifies the current direction & the third finger-like thumb signifies the force direction which is experienced through the conductor.

$$F = B*I*L \text{ Newtons}$$

Where,

'B' is the magnetic flux density,
'I' is current
'L' is the conductor's length in the magnetic field.
Whenever an armature winding is given toward a DC supply, then the flow of current will be set up within the winding. Field winding or permanent magnets will provide the magnetic field. So, armature conductors will experience a force because of the magnetic field based on the above-stated principle.
The Commutator is designed like sections to attain uni-directional torque or the path of force would have overturned each time once the way of the conductor's movement is upturned within the magnetic field. So, this is the working principle of the DC motor. When selecting DC motors, industrial buyers need to identify the key performance specifications, determine design and size requirements, and consider the environmental requirements of their application.
Shaft speed: A DC motor applies a voltage (V) to rotate a shaft at a proportional rotational speed (ω). Shaft speed specs generally refer to the no-load speed, which is the maximum speed the motor can reach when no torque is applied. Typically, shaft speed is given in rotations or revolutions per minute (rpm). These rotations or revolutions can be related to the number of radians to express the motor speed in radians per second (rad/s). For numerical calculations, this unit of rotational speed is more convenient. The following formula describes the relationship between radians per second and rotations or revolutions per minute.

$$\omega \text{ rad/s} = \omega \text{ rpm} \cdot (2\pi/60)$$

For an ideal DC motor, the rotational speed is proportional to the supplied voltage, or

$$\omega = j \cdot V$$

where j is the constant of proportionality, with units rad/(s-V).
Circuit Connections

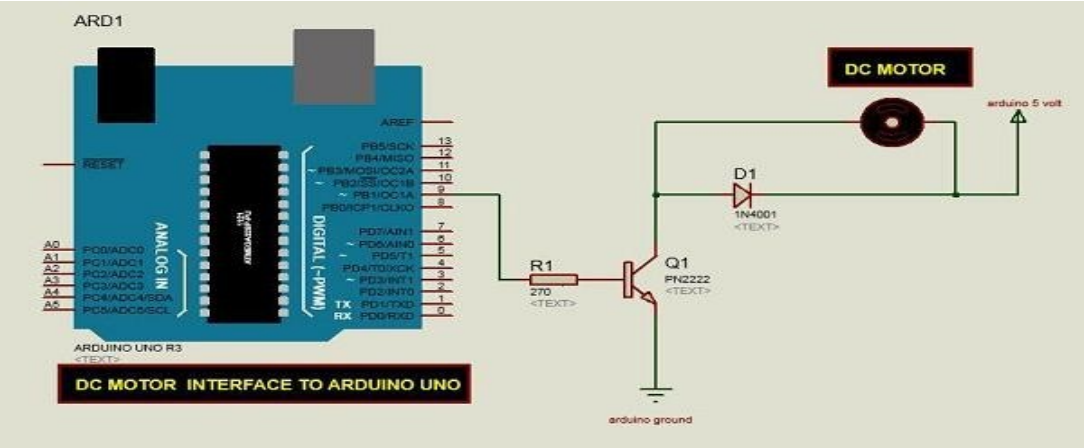


Figure 3.3: Schematic diagram of a DC motor, connected to the Arduino board.

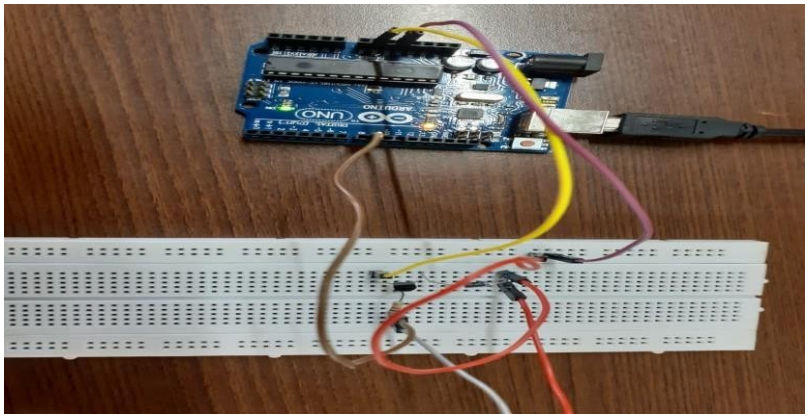


Figure 3.4: Arduino interfacing with Driver circuit implemented in lab

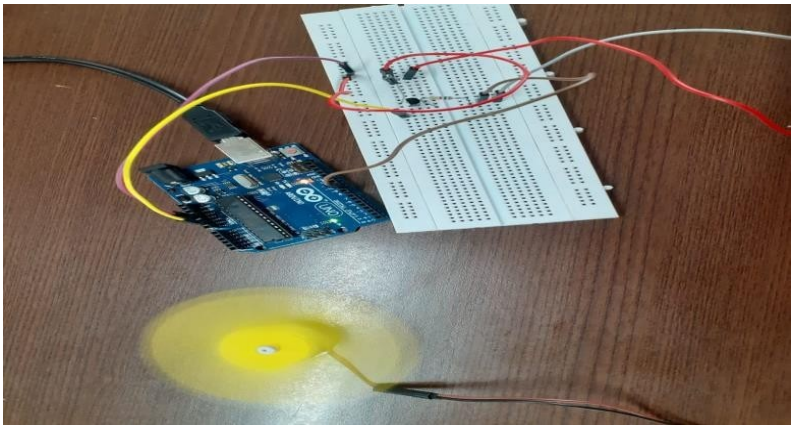


Figure 3.5: Arduino interfacing with Driver circuit and DC motor implemented

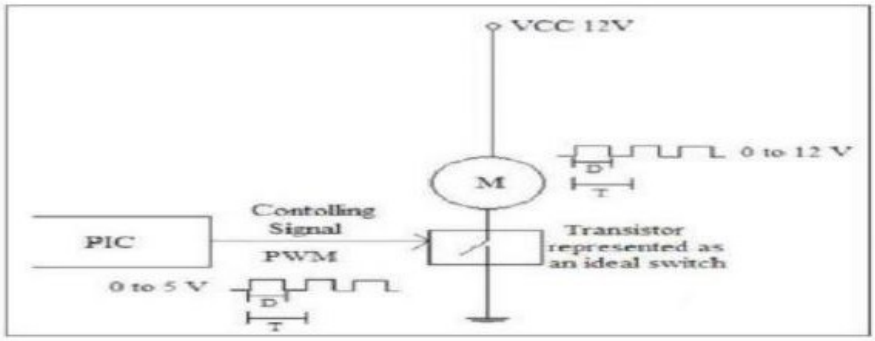


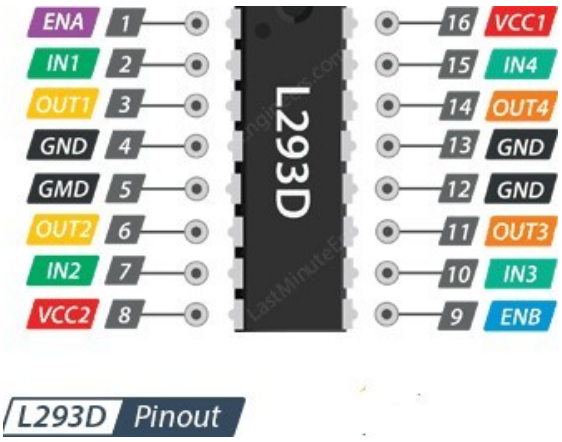
Figure 3.6: Speed control of DC motor Using PWM signal

L293D Driver IC:

In order to have a complete control over DC motor, we have to control its speed and rotation direction. This can be achieved by combining these two techniques.

PWM – For controlling speed
H-Bridge – For controlling rotation direction

The L293D is a dual-channel H-Bridge motor driver capable of driving a pair of DC motors or one stepper motor. That means it can individually drive up to two motors making it ideal for building two-wheel robot platforms.



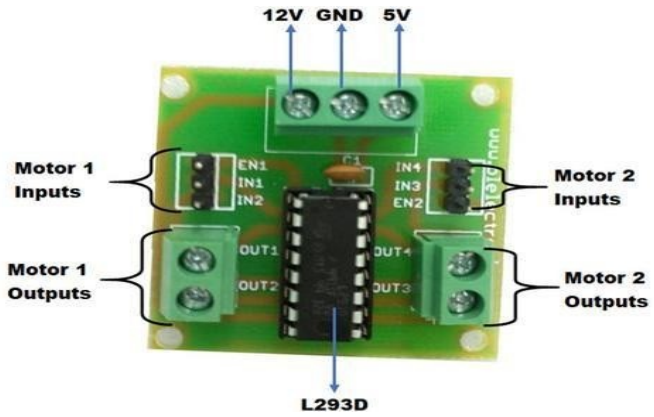


Figure 3.7: Pin Layout of L293D Motor Driver IC

Circuit Connections

- A button and a potentiometer are used to control the direction of rotation and speed of the motor
- A button is connected to Pin 12 of Arduino for driving the motor in forward and reverse direction with the other terminals of the button connected to GND.
- A potentiometer i.e. the wiper terminal of the pot is connected to analog input pin A0 of the Arduino UNO. The other terminals of the potentiometer are connected to 5V supply and ground
- Pin 1 of L293D IC is used to enable the driver channels 1 and 2 i.e. inputs of motor. It is an active high pin and hence it is connected to 5V supply.
- Pins 2 and 7 of L293D are inputs of drivers associated with motor 1. They are connected to Pins 11 and 10 of Arduino UNO respectively.
- Pins 3 and 6 of L293D are the output pins of first driver channel. They must be connected to the motor we are going to control.

- Pins 4, 5, 12 and 13 of the L293D IC are ground pins.
- Pin 16 of L293D IC is the supply pin for internal operations and is connected to a 5V supply.
- Pin 8 of L293D IC is the supply for driving the motor and is connected to a 5V supply.

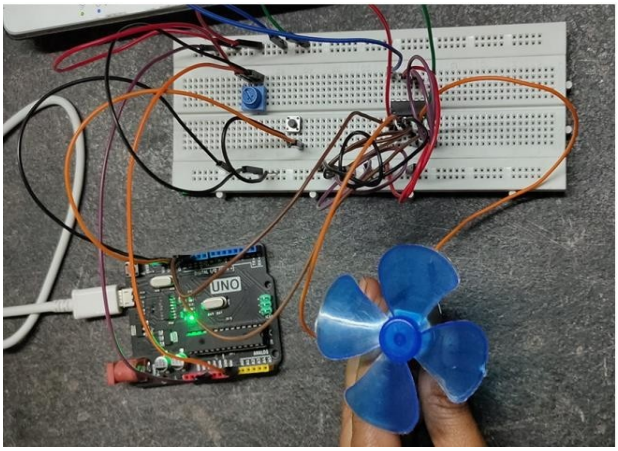


Figure 3.8: Direction and Speed control of DC Motor using L293D Motor Driver IC

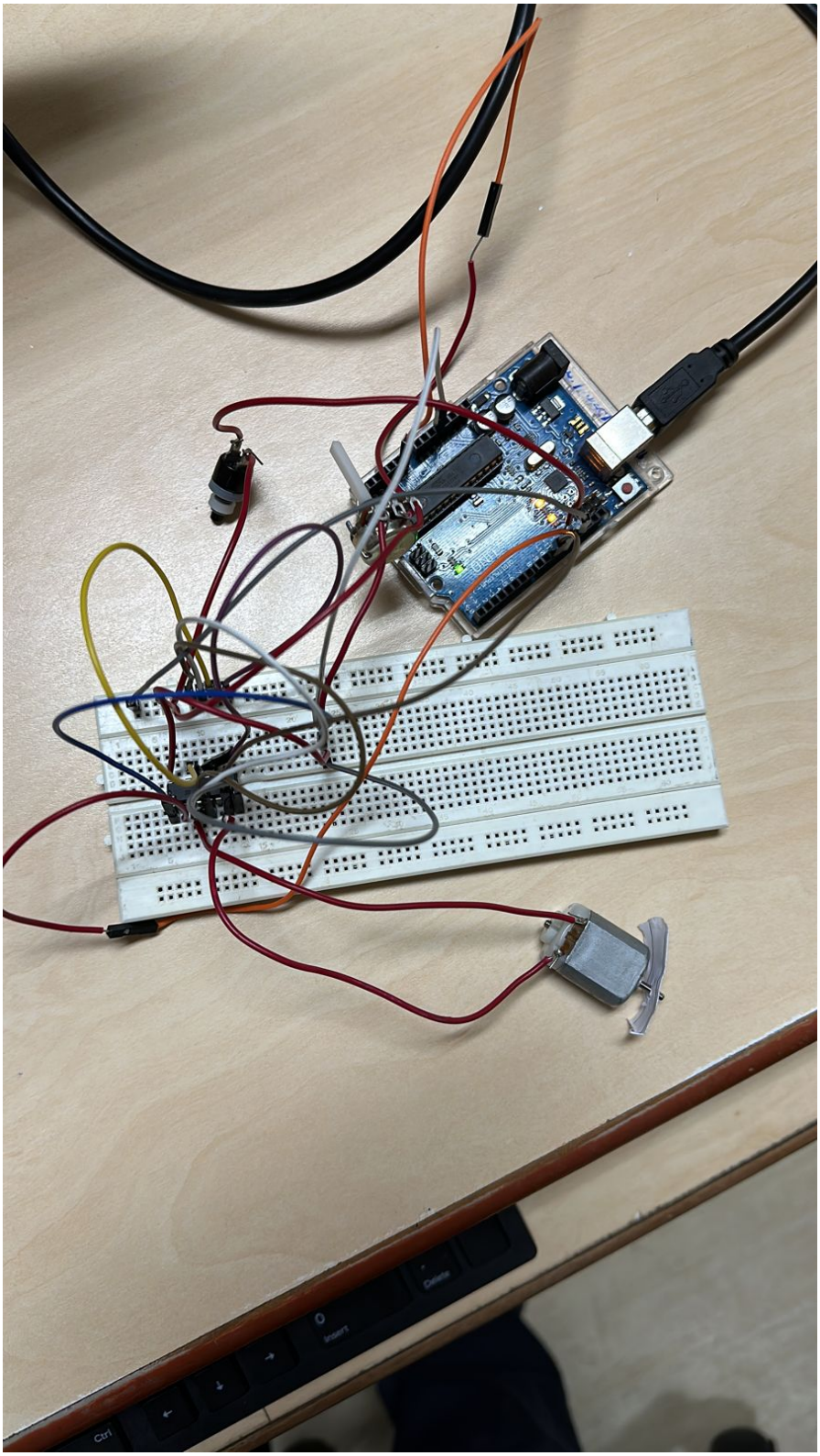
Conclusion:

Post Lab Questions:

1. List and state Different types of DC motor.
2. What is significance PWM? How to control speed using PWM pin? Explain in detail.
3. List out various Actuators used for IoT and explain anyone other than DC Motor.
4. What are different applications of DC motor? Explain any one in detail.

Additional links for more information:

1. Introduction to Actuators
   https://nptel.ac.in/content/storage2/courses/112104158/lecture36.pdf
2. Arduino-DC Motor
   https://www.tutorialspoint.com/arduino/arduino_dc_motor.htm
3. DC Motor Arduino Tutorial
   https://lastminuteengineers.com/l293d-dc-motor-arduino-tutorial/



Name: Shreerang Mhatre
Roll no: 29
Batch: A2

Exp no. 2

MIT-WPU
॥ विद्यशान्तिर्युवं धुवा ॥
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

* ~~Procedure~~ Post Lab Questions -

Q1. List out various analog sensors and explain any one in brief

→ Analog sensors are sensors that generate an output signal that varies continously in response to change in the input parameter being measured.
Some analog sensors are -
1) Temperature sensors
2) Pressure sensors
3) Light sensors
4) Accelerometers
5) Strain gauges
6) Proximity sensors.

· Thermistor - A thermistor is a type of temp sensor that uses the change in electrical resistance of a material to measure temperature. As the temperature of the thermistor increases, its electrical resistance decreases, and vice versa. This change in resistance can be measured using a simple voltage divider circuit, where the thermistor ~~ther~~ is placed in series with a known resistance.

www.mitwpu.edu.in

MIT-WPU
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

Q2) List out varias digital sensors and explain any one in brief.

→ Digital sensors, unlike analog sensors, provide a discrete output signal that is either a 0 or 1.

Some digital sensors are -

1) Motion sensors
2) Light sensors
3) Proximity sensors
4) Magnetic sensors
5) Humidity sensors
6) Gas sensors.

Passive Infrared motion sensor -

This sensor detects the presence of a person or an object by measuring the infrared radiation emitted by them. The PIR sensor consits of a pyroelectric sensor and a Fresnel lens, which are both sensitive to infrared radiation. When an object moves within the field, ~~of~~ view, of the sensor it causes a change in the infrared radiation detected by the pyroelectric sensor.

www.mitwpu.edu.in

Q3) State the applications of any one sensors.

→ Accelerometer - It is used to measure acceleration or changes in velocity. Some applications are -

① Automotive industry -
Accelerometers are used in cars and other vehicles to detect changes in acceleration and determine the orientation.

② Aerospace industry -
Accelerometers are used in aircraft, rockets and satellites to measure changes in acceleration & determine the orientation of aircraft.

③ Industrial automation:-
Accelerometers are used in industrial automation systems to monitor the vibration levels of machinery.

④ Consumer electronics -
Accelerometers are used in smartphones, tables and other portable electronic devices to detect changes in orientation and adjust the display accordingly.

Q4) Explain the commonly used command of Arduino IDE.

→ Arduino IDE is an integrated development environment that is used to program and upload code to Arduino boards.

(1) Verify / compile -
This command is used to check the code for any syntax errors and compile it into a machine-readable format.

(2) Upload -
This command is used to upload the compiled code to the Arduino board.

(3) Serial Monitor -
This command opens a serial communication window that allows you to send and receive data between the Arduino board and the computer.

(4) New sketch -
This command opens a new sketch window where you can write your code.

Second Year B. Tech (EL&CE)

Semester: IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY**

Roll No: 29 **Batch: A2**

Experiment No: 04

Name of the Experiment: Understanding Node MCU as development platform and connecting to Wi-Fi network through Arduino IDE.

Performed on: 22/02/2023

Submitted on: 04/03/2023

Marks	Teacher's Signature with date

Aim: Understanding Node MCU as development platform and connecting to Wi-Fi network through Arduino IDE.

Prerequisite: C or C++ Programming, NodeMCU board layout, NodeMCU pin configuration, Arduino IDE

Objectives:

1. To understand integration of NodeMCU board with Arduino IDE
2. To Configure and upload code on ESP8266 module
3. To interface sensor with NodeMCU and check output with Arduino IDE.
4. To connect the NodeMCU ESP8266 to the Wi-Fi router using SSID and password of our Home network Wi-Fi

Components and equipment required:

NodeMCU Board, USB cable, Arduino IDE, PIR Sensor (HC-SR501), LED, Buzzer, Bread board, Jumper wires etc.

Theory:

PIR sensors allows to sense motion, generally used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low-power, easy to use and don't wear out. They are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion" sensors. When any body (human, animal or body emitting infrared radiations) emitting heat waves comes close to PIR sensor, PIR motion sensor will instantly detect the presence of it and rings a buzzer alarm to notify about an intrusion. Every body emits heat waves and heat waves contains infrared

rations. Generally the living bodies emit more heat than the raw materials. So PIR sensor can be used to detect the motion of living bodies.

PIR stands for passive infrared sensor. From passive it means it did not emit anything by its own to detect presence of a body, rather it receives the infrared waves emitted by the bodies and generates a presence alert as output. Infrared radiations cannot be seen by naked human eye. PIR motion detection sensors has a pyroelectric sensor in them. Pyroeletric sensor is enclosed in a round shaped lens or plastic material. The lens or plastic material focuses the received infrared radiations from body on a single point on pyroelectric sensor. Presence alert by PIR sensor depends on the strength of the infrared radiations received on the pyroelectric sensor. The sensitivity and range of PIR motion sensors can be varied.

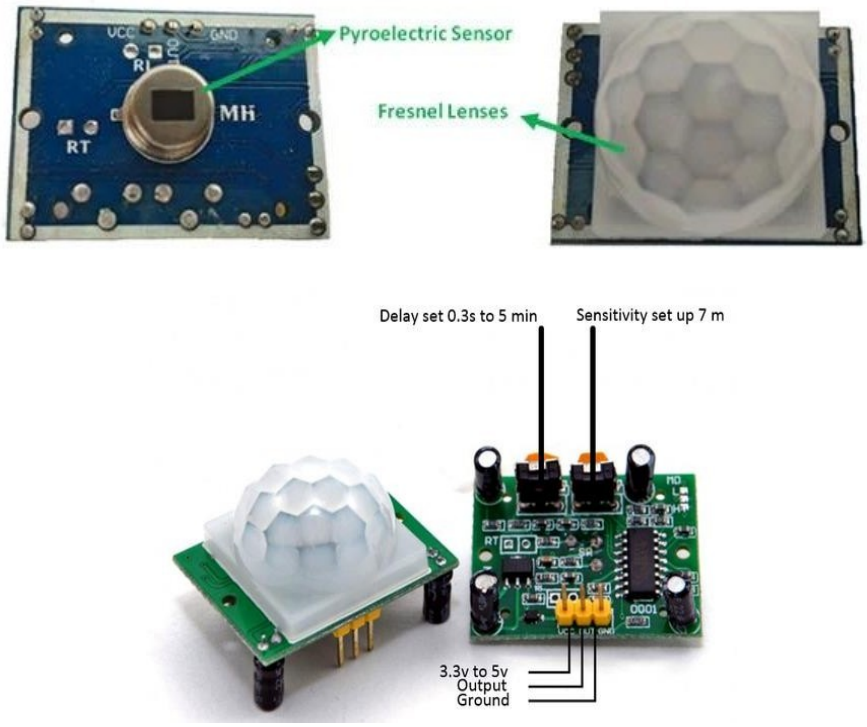
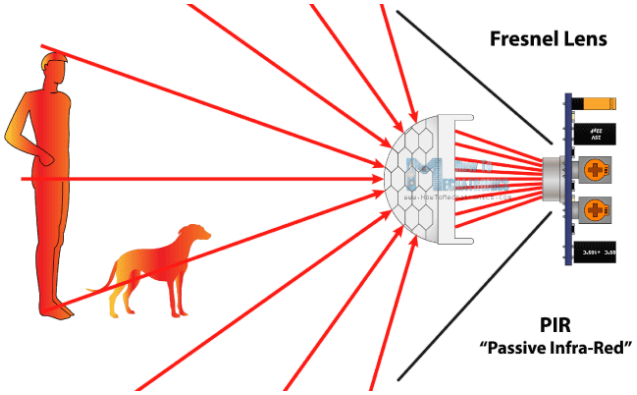


Figure 4.1: PIR Sensor Pin Out

Working Principle:

Generally, PIR sensor can detect animal/human movement in a requirement range. PIR is made of a pyroelectric sensor, which is able to detect different levels of infrared radiation. The detector itself does not emit any energy but passively receives it.



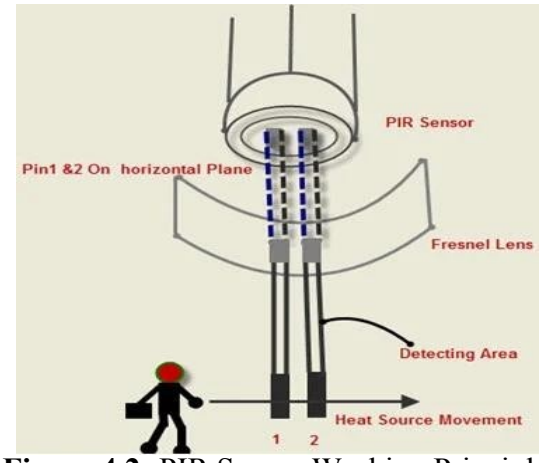


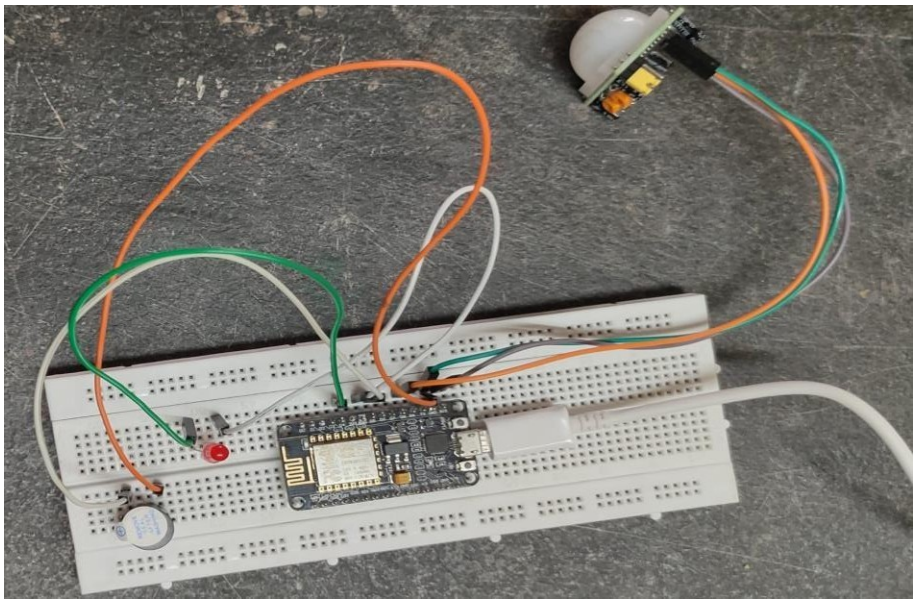
Figure 4.2: PIR Sensor Working Principle

It detects infrared radiation from the environment. Once there is infrared radiation from the human body particle with temperature, focusing on the optical system causes the pyroelectric device to generate a sudden electrical signal. Passive infrared alarms classified into infrared detectors (infrared probes) and alarm control sections. The most widely used infrared detector is a pyroelectric detector. It uses as a sensor for converting human infrared radiation into electricity. If the human infrared radiation is directly irradiated on the detector, it will, of course, cause a temperature change to output a signal. But in doing all this, the detection distance will not be more. In order to lengthen the detection distance of the detector, an optical system must be added to collect the infrared radiation. Usually, plastic optical reflection system or plastic Fresnel lens used as a focusing system for infrared radiation.

Circuit Connections:

- Anode terminal of LED is connected to D4 pin of NodeMCU and cathode terminal is connected to GND of NodeMCU
- Long terminal (+ve) of buzzer is connected to D5 pin of NodeMCU and short terminal (-ve) of buzzer is connected to GND pin of NodeMCU.

- OUT pin of PIR sensor is connected to D8 pin of NodeMCU, GND pin of PIR sensor is connected to GND of NodeMCU and VCC pin of PIR sensor is connected to 3.3V pin of NodeMCU



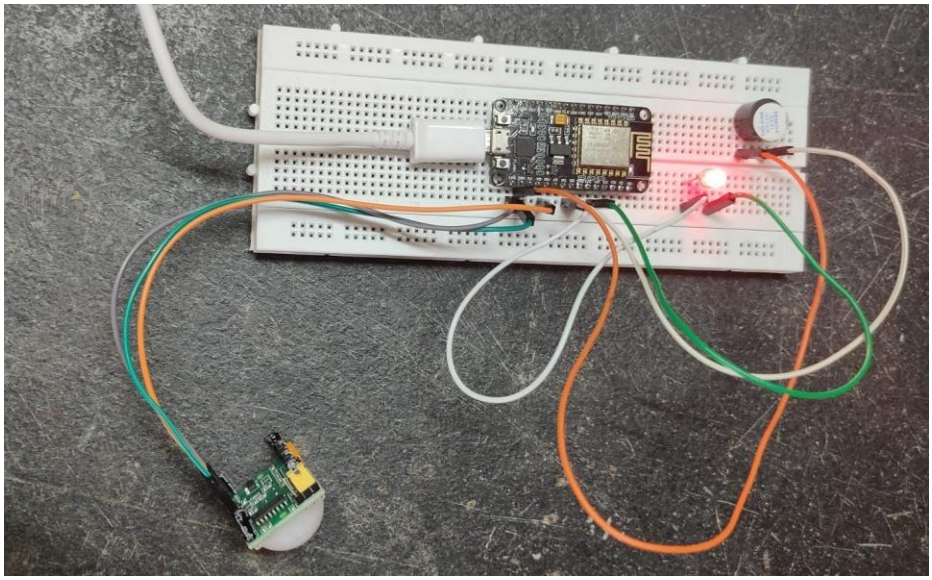


Figure 4.3: Circuit Connection

Connecting NodeMCU to Wi-Fi network through Arduino IDE:

When it comes to connecting your Arduino to the internet there are many options, but one of the most popular and widely available is the ESP8266. NodeMCU is an open source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc, it can solve many of the project's needs alone. The ESP8266 is a microcontroller

developed by Espressif Systems. Known as a WiFi module, this microcontroller can be used to perform various WiFi-related activities, with applications in home automation and beyond.

Devices that connect to Wi-Fi networks are called stations (STA). Connection to Wi-Fi is provided by an access point (AP) that acts as a hub for one or more stations. The access point on the other end is connected to a wired network. An access point is usually integrated with a router to provide access from a Wi-Fi network to the internet. Each access point is recognized by a SSID (Service Set IDentifier) that essentially is the name of network you select when connecting a device (station) to the Wi-Fi.

ESP8266 modules can operate as a station, so we can connect it to the Wi-Fi network. It can also operate as a soft access point (soft-AP), to establish its own Wi-Fi network. When the ESP8266 module is operating as a soft access point, we can connect other stations to the ESP module. ESP8266 is also able to operate as both a station and a soft access point mode. This provides the possibility of building e.g. mesh networks.

Circuit Connections:

- Anode terminal of LED is connected to D7 pin of NodeMCU and cathode terminal is connected to GND of NodeMCU
- Write sketch for accessing WiFi connectivity
- Once connected to WiFi, reflect LED ON/OFF status on Serial Monitor

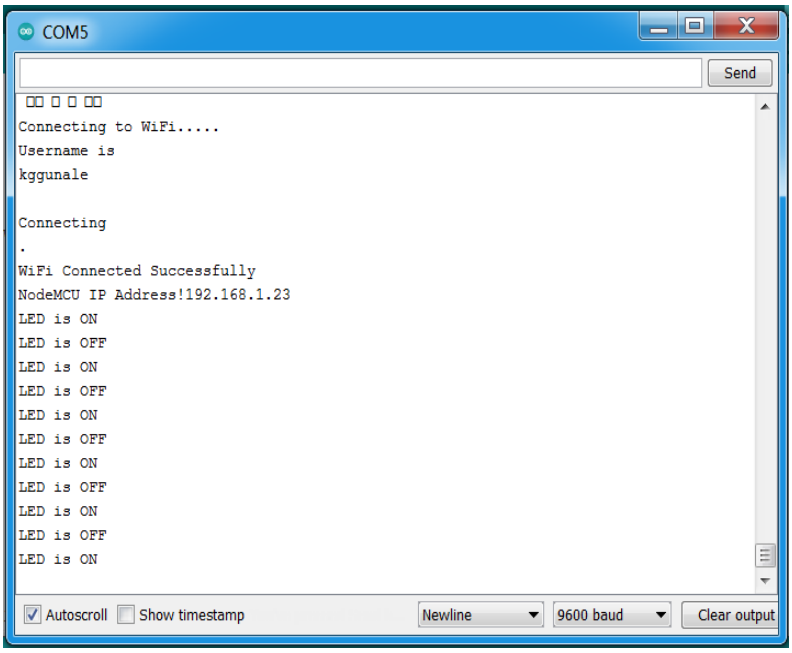


Figure 4.4: WiFi Connectivity status and LED Status on Serial Monitor

Conclusion:

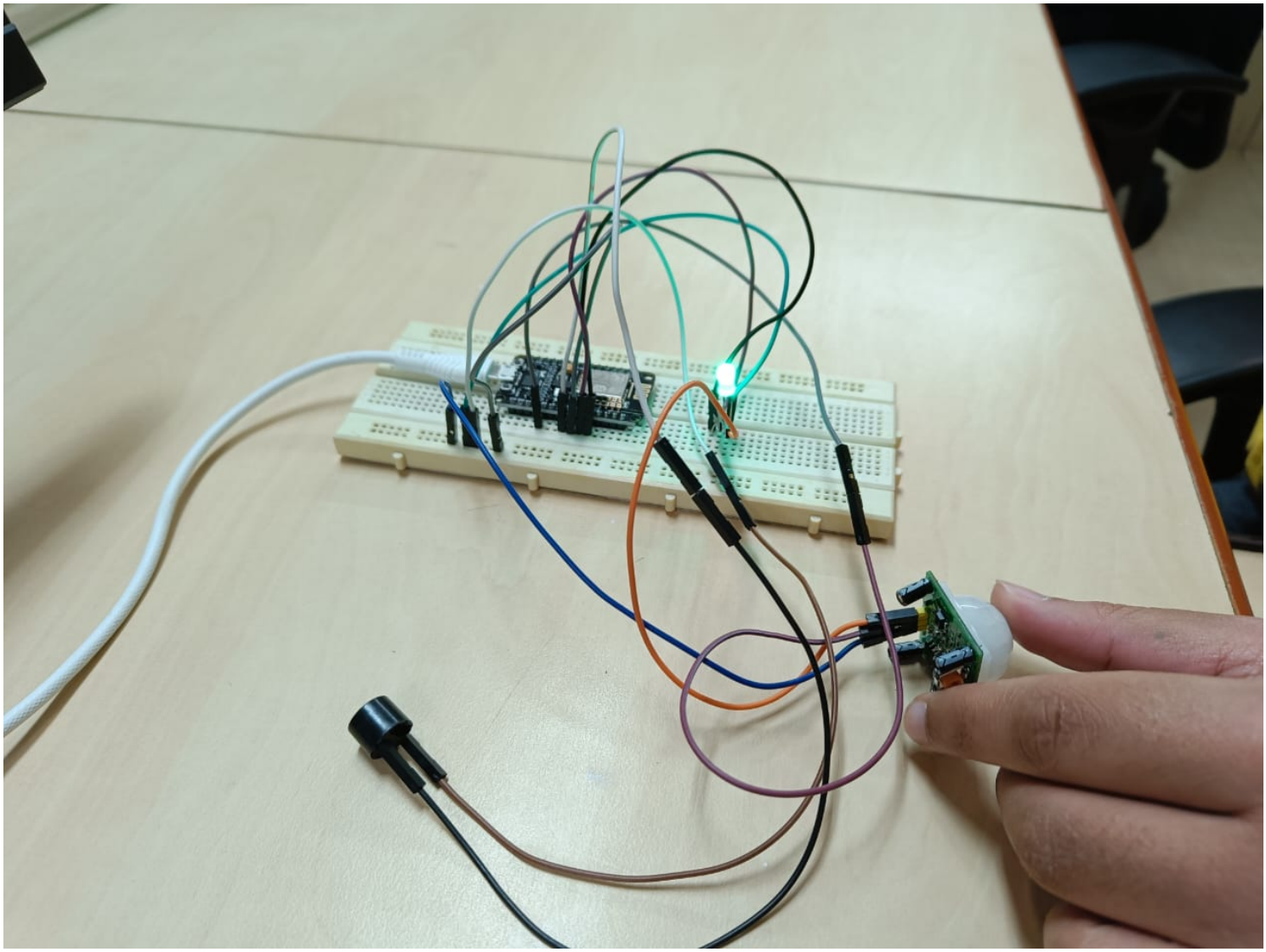
Post Lab Questions:

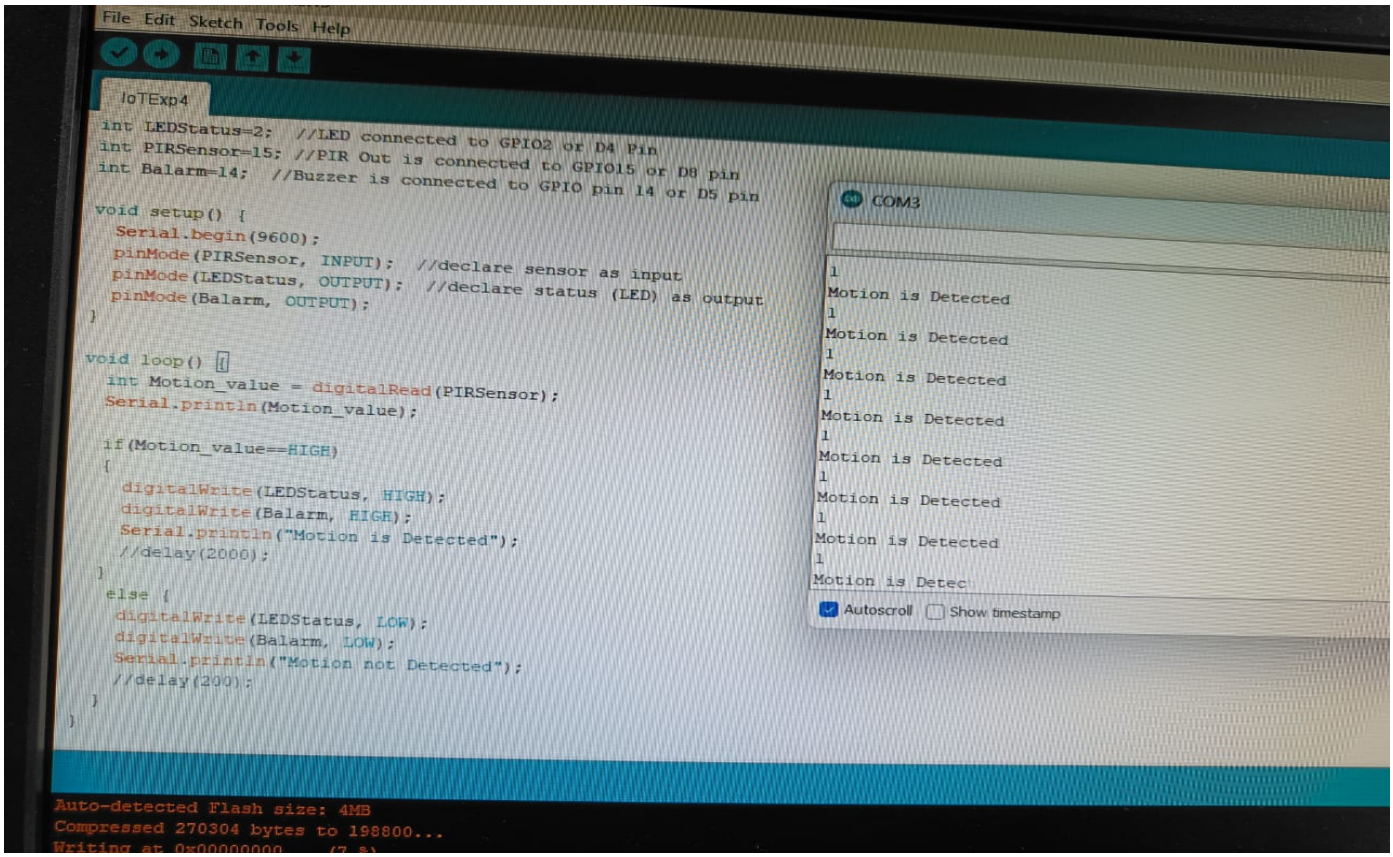
1. List Different motion sensors available and explain any one.
2. Compare between ESP32 and ESP8266
3. List out various applications of PIR sensor and explain any one in detail
4. Mention various pins of ESP8266 NodeMCU and explain in detail all the Pins.

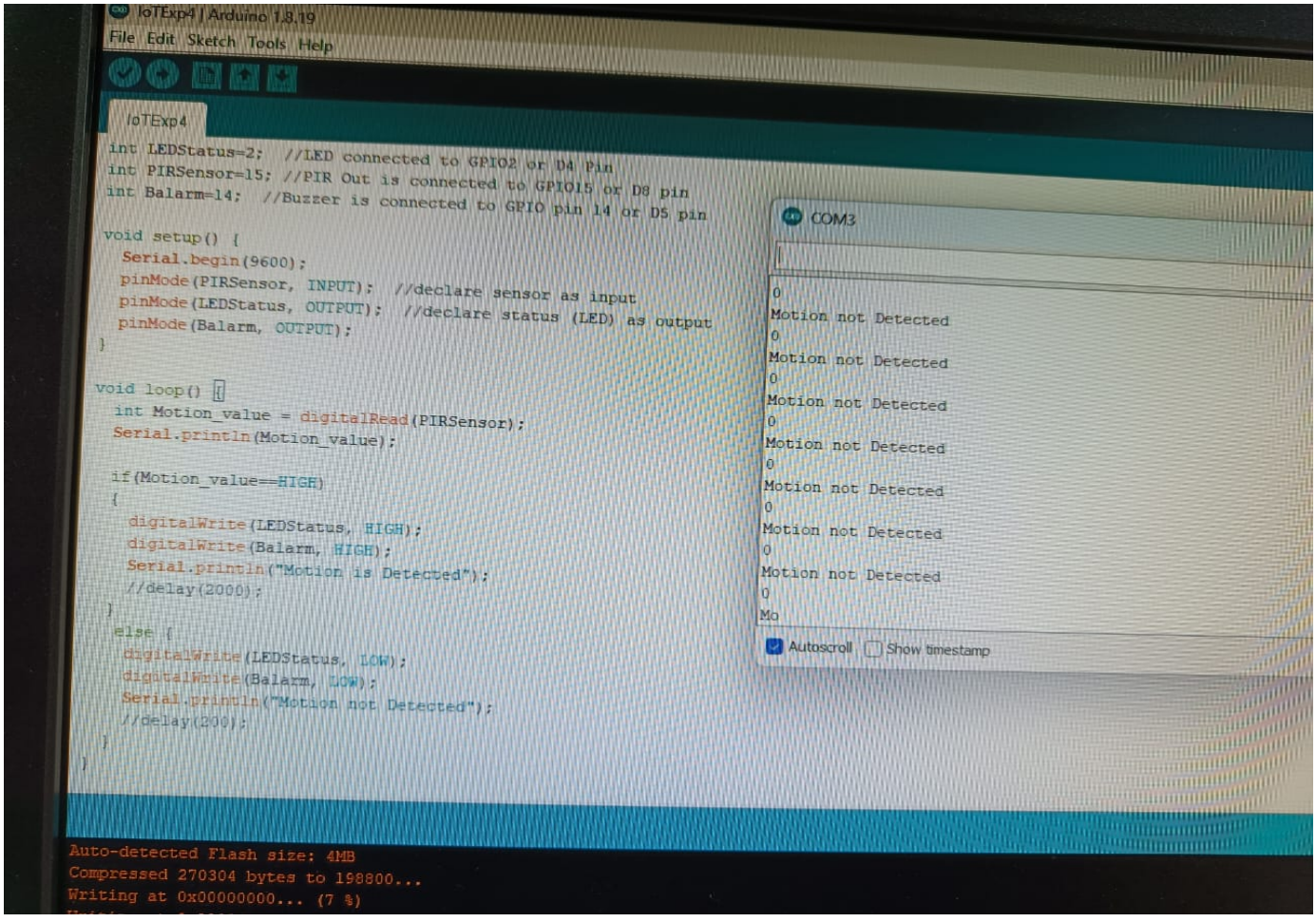
Additional links for more information:

1. IoT Automation with ESP8266
   https://www.udemy.com/course/iot-internet-of-things-automation-with-esp8266/
2. Establishing a WiFi Connection
   https://tttapa.github.io/ESP8266/Chap07%20-%20Wi-Fi%20Connections.html
3. IoT based Home automation using PIR sensor and Blynk App
   https://www.ijeat.org/wp-content/uploads/papers/v9i4/D7583049420.pdf
4. Insight into ESP8266 NodeMCU Features
   https://lastminuteengineers.com/esp8266-nodemcu-arduino-tutorial/

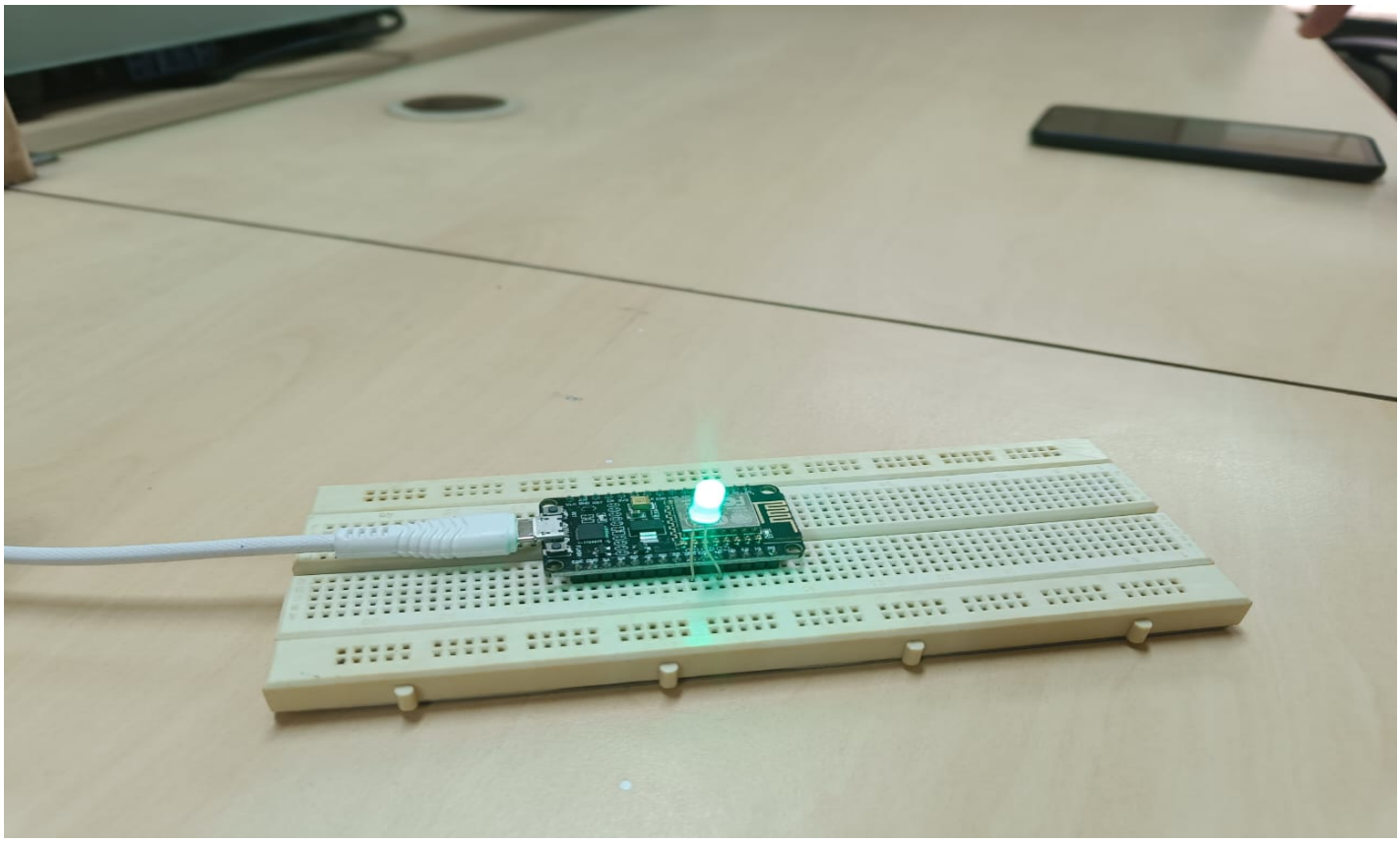
PIR sensor without WIFI

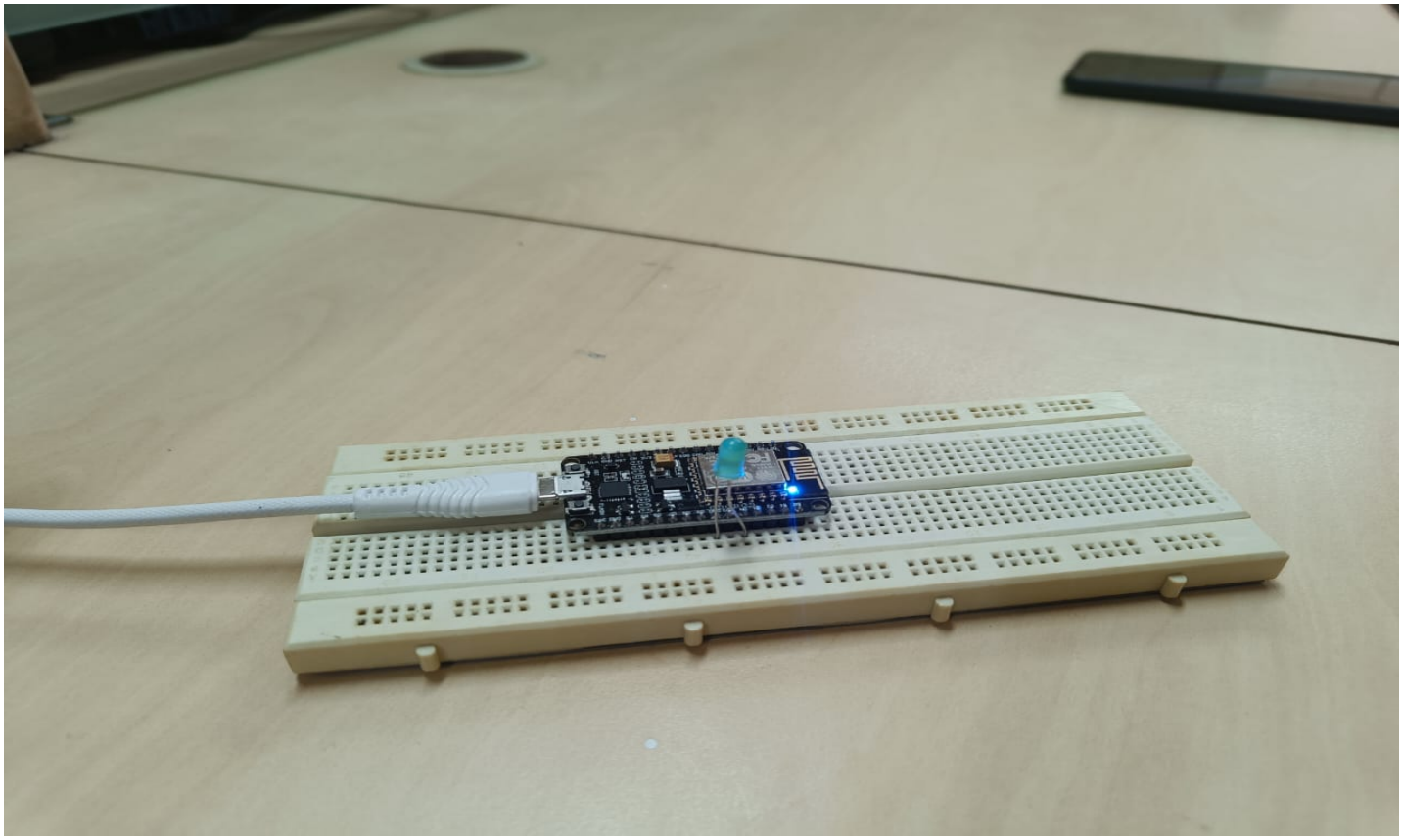


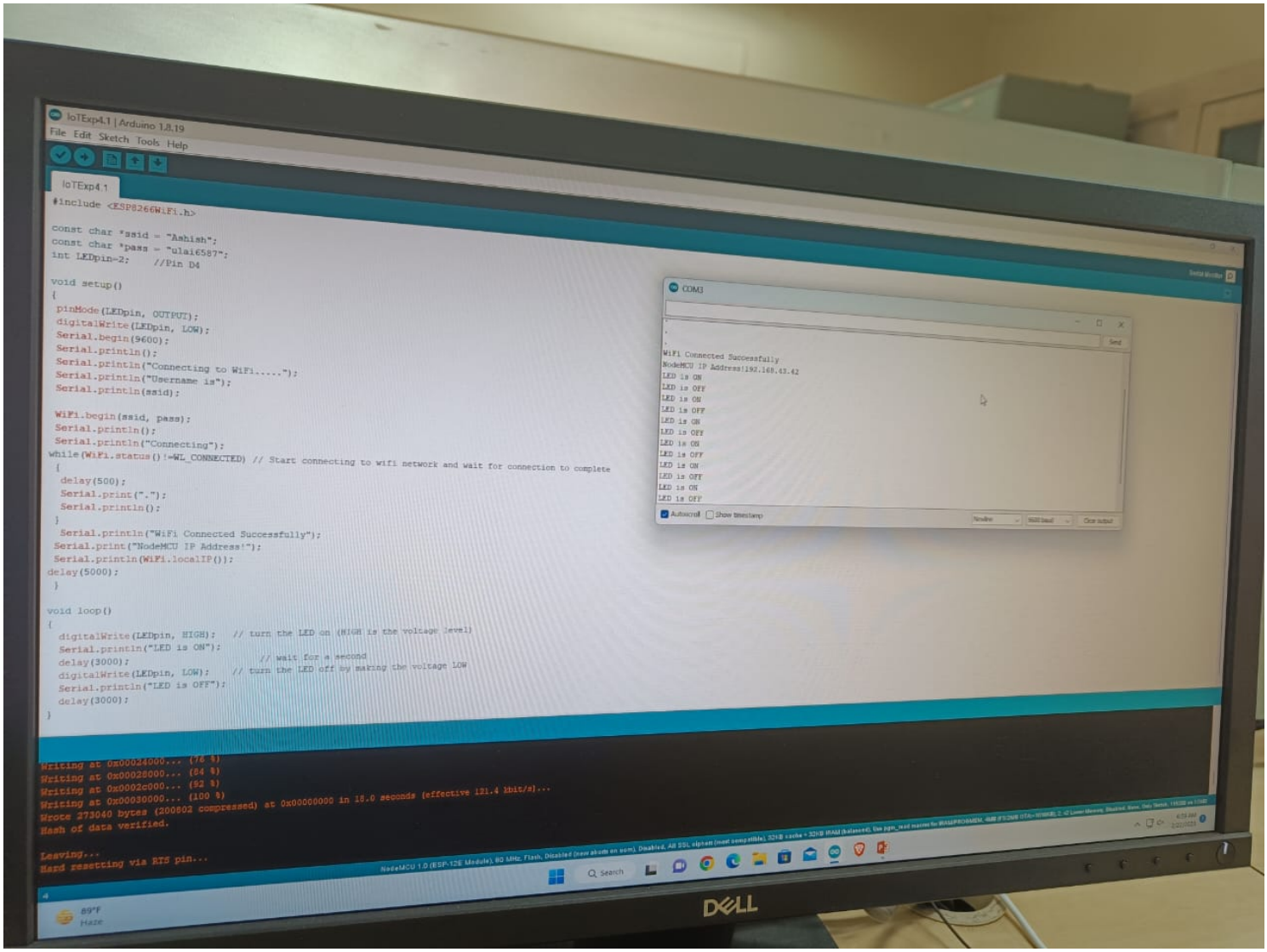

File Edit Sketch Tools Help
IoTExp4
int LEDStatus=2; //LED connected to GPIO2 or D4 Pin
int PIRSensor=15; //PIR Out is connected to GPIO15 or D8 pin
int Balarm=14; //Buzzer is connected to GPIO pin 14 or D5 pin
void setup() {
Serial.begin(9600);
pinMode(PIRSensor, INPUT); //declare sensor as input
pinMode(LEDStatus, OUTPUT); //declare status (LED) as output
pinMode(Balarm, OUTPUT);
}
void loop() {
int Motion_value = digitalRead(PIRSensor);
Serial.println(Motion_value);
if(Motion_value==HIGH)
{
digitalWrite(LEDStatus, HIGH);
digitalWrite(Balarm, HIGH);
Serial.println("Motion is Detected");
//delay(2000);
}
else {
digitalWrite(LEDStatus, LOW);
digitalWrite(Balarm, LOW);
Serial.println("Motion not Detected");
//delay(200);
}
}
COM3
1
Motion is Detected
1
Motion is Detected
1
Motion is Detected
1
Motion is Detected
1
Motion is Detected
1
Motion is Detected
1
Motion is Detected
1
Motion is Detec
Autoscroll
Show timestamp
Auto-detected Flash size: 4MB
Compressed 270304 bytes to 198800...
Writing at 0x00000000 (7 %)


IoTExp4 | Arduino 1.8.19
File Edit Sketch Tools Help
IoTExp4
int LEDStatus=2; //LED connected to GPIO2 or D4 Pin
int PIRSensor=15; //PIR Out is connected to GPIO15 or D8 pin
int Balarm=14; //Buzzer is connected to GPIO pin 14 or D5 pin
void setup() {
Serial.begin(9600);
pinMode(PIRSensor, INPUT); //declare sensor as input
pinMode(LEDStatus, OUTPUT); //declare status (LED) as output
pinMode(Balarm, OUTPUT);
}
void loop() {
int Motion_value = digitalRead(PIRSensor);
Serial.println(Motion_value);
if(Motion_value==HIGH)
{
digitalWrite(LEDStatus, HIGH);
digitalWrite(Balarm, HIGH);
Serial.println("Motion is Detected");
//delay(2000);
}
else {
digitalWrite(LEDStatus, LOW);
digitalWrite(Balarm, LOW);
Serial.println("Motion not Detected");
//delay(200);
}
}
COM3
0
Motion not Detected
0
Motion not Detected
0
Motion not Detected
0
Motion not Detected
0
Motion not Detected
0
Motion not Detected
0
Motion not Detected
0
Mo
Autoscroll
Show timestamp
Auto-detected Flash size: 4MB
Compressed 270304 bytes to 198800...
Writing at 0x00000000... (7 %)

With WIFI






IoTExp4.1 | Arduino 1.8.19
File Edit Sketch Tools Help
IoTExp4.1
#include <ESP8266WiFi.h>
const char *ssid = "Aahish";
const char *pass = "ulai6587";
int LEDpin=2; //Pin D4
void setup()
{
pinMode(LEDpin, OUTPUT);
digitalWrite(LEDpin, LOW);
Serial.begin(9600);
Serial.println();
Serial.println("Connecting to WiFi.....");
Serial.println("Username is");
Serial.println(ssid);
WiFi.begin(ssid, pass);
Serial.println();
Serial.println("Connecting");
while(WiFi.status()!=WL_CONNECTED) // Start connecting to wifi network and wait for connection to complete
{
delay(500);
Serial.print(".");
Serial.println();
}
Serial.println("WiFi Connected Successfully");
Serial.print("NodeMCU IP Address:");
Serial.println(WiFi.localIP());
delay(5000);
}
void loop()
{
digitalWrite(LEDpin, HIGH); // turn the LED on (HIGH is the voltage level)
Serial.println("LED is ON");
delay(3000);   // wait for a second
digitalWrite(LEDpin, LOW);   // turn the LED off by making the voltage LOW
Serial.println("LED is OFF");
delay(3000);
}
COM3
Wifi Connected Successfully
NodeMCU IP Address:192.168.43.42
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
LED is ON
LED is OFF
Autoscroll
Show timestamp
Writing at 0x00024000... (76 %)
Writing at 0x00028000... (84 %)
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 273040 bytes (200802 compressed) at 0x00000000 in 18.0 seconds (effective 121.4 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
NodeMCU 1.0 (ESP-12E Module), 80 MHz, Flash, Disabled (new aborts on oom), Disabled, All SSL ciphers (most compatible), 32KB cache + 32KB IRAM (balanced), Use pgm_read macros for IRAM/PROGMEM, 4MB (FS:2MB OTA:~1019KB), 2, v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM3
89°F
Haze
Search
DELL

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

22/02/2023

Exp no: 4

MIT-WPU
॥ विश्वशान्तिर्भुवं धुवा ॥
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

* Post Lab Questions -

Q1) List different motion sensors available & explain any one.

→ The common motion sensors are -

1. Passive Infrared (PIR) sensors
2. Ultrasonic sensors
3. Microwave sensors
4. Tomographic sensors
5. Video Motion Detection (VMD)

• PIR sensor -

PIR sensors detect motion by measuring changes in the infrared (IR) energy in their field of view. IR energy is emitted by all objects that have a temperature above absolute zero. When a warm-blooded animal, a human for instance, enters a PIR sensor's field of view, it radiates IR energy that is detected by the sensor.

Q2) Compare between ESP 32 & ESP 8266

→ The ESP32 & ESP8266 are both - low-cost, low-power wifi & bluetooth enabled microcontrollers developed by Espressif sys.

① CPU & Memory -
The ESP32 has a dual-core processor that runs at up to 240 MHz, while the ESP8266 has a single-core processor that runs up to 80 MHz.

② Connectivity -
Both the ESP32 & ESP8266 have wifi & bluetooth capabilities

③ Power Consumption
The ESP32 is designed to be more power efficient than the ESP8266.

④ Price -
The ESP8266 is generally cheaper than the ESP32, making it a popular choice for hobbyists & makers on a budget.

MIT-WPU
॥ विश्वशान्तिर्भुवं भुवा ॥
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

Q3) List ~~different~~ out various applications of PIR sensor & explain any one in detail

-> Applications of PIR sensor are -

1. Security systems
2. Lighting controls
3. Home automation
4. Industrial automation
5. Health monitoring

One of the most common application of PIR sensors is in security systems. In this application, the PIR sensor is used to detect motion in a restricted area, such as a room or hallway. When the sensor detects motion, it sends a signal to the control unit, which triggers an alarm or sends a notification to the security personnel.

->

MIT-WPU
MIT WORLD PEACE UNIVERSITY | PUNE

(Q4) Mention various pins of ESP 8266 Node MCU & explain in detail all the pins -

→ Various pins on the Node MCU board -

① VIN: This pin is used to supply power to the Node MCU board. It can accept a voltage range of 5-12VDC

② GND: This pin is the ground reference for the Node MCU board.

③ 3V3: This pin provides a regulated 3.3v output that can be used to power other peripherals connected to the board

④ EN: This is the enable pin, which is used to enable or disable the Node MCU board.

⑤ RST: This is the reset pin, which is used to reset the Node MCU board.

Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE
TECHNOLOGY, RESEARCH, SOCIAL INNOVATION & PARTNERSHIPS

Second Year B. Tech (EL & CE)

Semester: IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY**

Roll No: 29 **Batch: A2**

Experiment No: 05

Name of the Experiment: **Exploring cloud infrastructure for connecting IoT devices and sending and visualizing sensor data to open source cloud via Arduino IDE.**

Performed on: 02/03/2023

Submitted on: 04/03/2023

Marks	Teacher's Signature with date

Aim: Exploring cloud infrastructure for connecting IoT devices and sending and visualizing sensor data to open source cloud via Arduino IDE.

Prerequisite: Basics of NodeMCU Model, Cloud concepts.

.

Objective:

1. Understand DHT11/DHT22 sensor
2. Sensor interfacing with NodeMCU (DHT11/DHT22)
3. Display Temperature/Humidity values on serial Monitor
4. Connect to WiFi, display WiFi IP and Display Temperature/Humidity values on serial Monitor
5. Understand Thinger.io cloud platform
6. Display Temperature/Humidity values on cloud platform

Components and equipment required:

NodeMCU, DHT11/22 sensor, Breadboard, Connecting Wires etc.

Theory:

Temperature sensor: It is a device, a thermocouple or RTD, that provides temperature measurement through an electrical signal.

Thermocouple: It is made from two dissimilar metals that generate electrical voltage in direct proportion to changes in temperature. The wires are joined together to form measuring junction and reference junction.

RTD: Resistor temperature detection is variable resistor that will change its electrical resistance in direct proportion to changes in temperature in precise, repeatable & linear manner.

The DHT22 sensor is used to measure the temperature and humidity. It is also known as AM2302. This sensor is cheap and also has better accuracy.

Specifications of DHT22

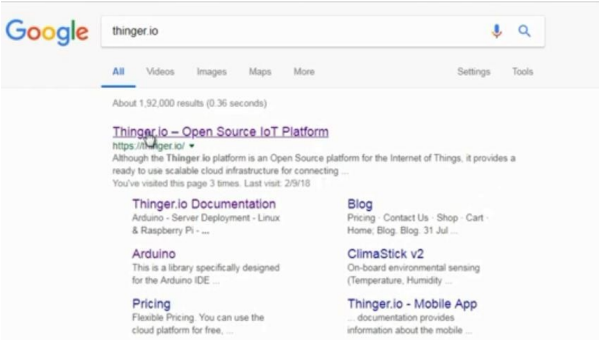
The specifications of the temperature and humidity sensor DHT22 are as follows:

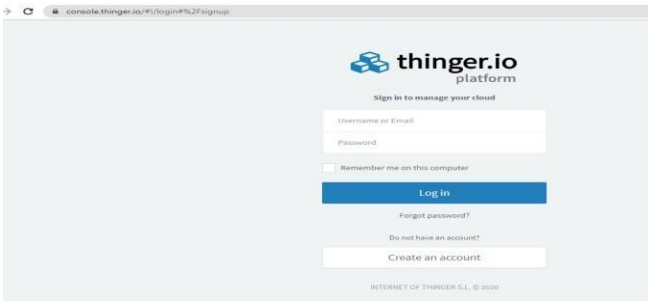
- Temperature range is from -40 to 125 degree Centigrade with accuracy of ±0.5 °C.
- Humidity range is from 0 to 100% with accuracy of ± 2-5%.
- Sampling rate is 0.5 Hz.
- Operating Voltage is 3-5V.
- Maximum Current while measuring is 2.5mA.

Setting up the Thinger.io Account:

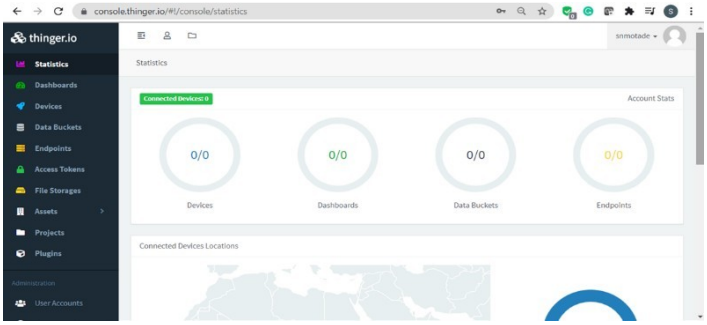
Thinger.io is a cloud IoT Platform that provides every needed tool to prototype, scale and manage connected products in a very simple way.

- **Free IoT platform:** Thinger.io provides a lifetime freemium account with only few limitations to start learning and prototyping when your product becomes ready to scale, you can deploy a Premium Server with full capacities within minutes.
- **Simple but Powerful:** Just a couple code lines to connect a device and start retrieving data or controlling its functionalities with our web-based Console, able to connect and manage thousands of devices in a simple way.
- **Hardware agnostic:** Any device from any manufacturer can be easily integrated with Thinger.io's infrastructure.
- **Extremely scalable & efficient infrastructure:** thanks to our unique communication paradigm, in which the IoT server subscribes device resources to retrieve data only when it is necessary, a single Thinger.io instance is able to manage thousands of IoT devices with low computational load, bandwidth and latencies.
- **Open-Source:** most of the platform modules, libraries and APP source code are available in our Github repository to be downloaded and modified with MIT license.



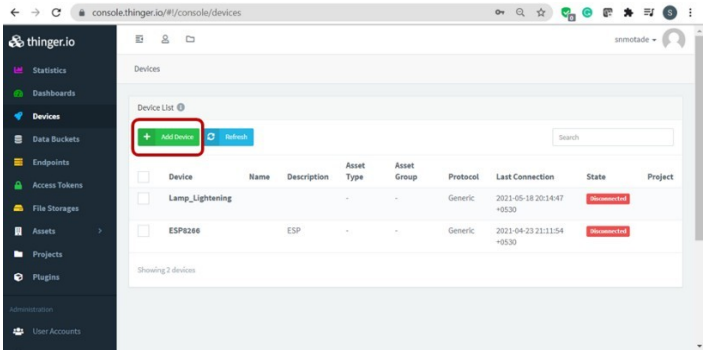


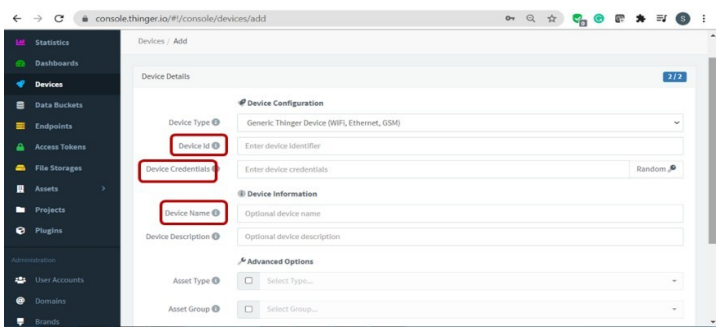
Thinger.io Cloud Platform:



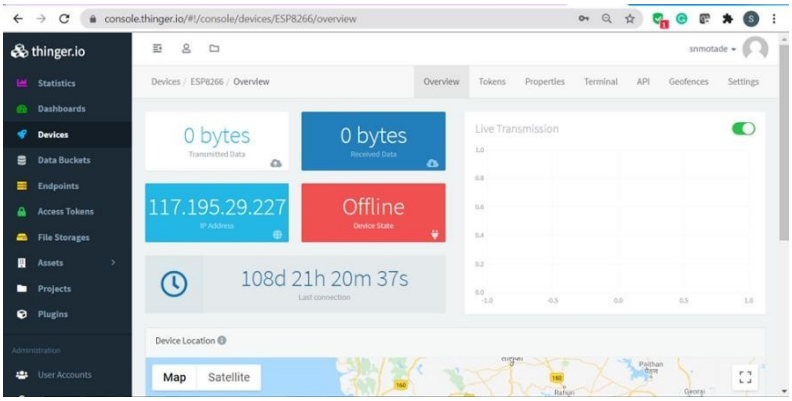
Create the device:

The first step with Thinger.io (except for not connected devices like Sigfox) is creating a device profile, which will relate the hardware device with the user account. Any device in Thinger.io must be registered to get access to the cloud. Each one has its own identifier and credentials and is related to the user account. All device creation and management processes are performed from the devices tab in the main menu.

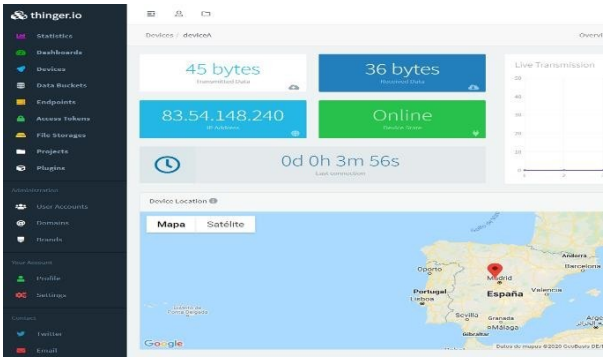




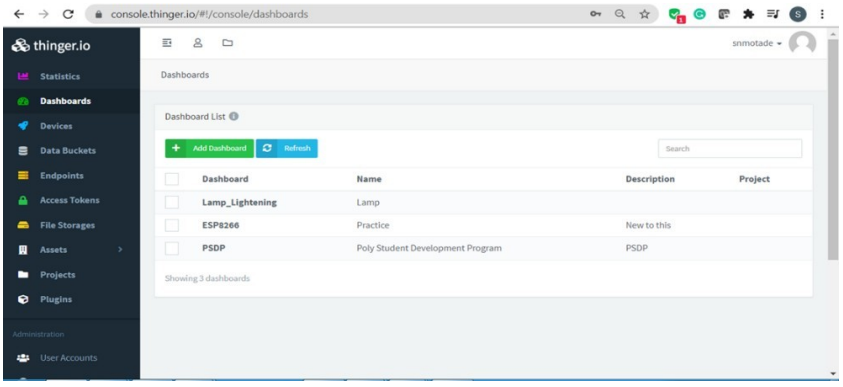
Once in the code our account identifier, device identifier, and device credentials have established, we can compile and flash the program. Meanwhile, we can open our device in the cloud console, just by clicking its identifier in the devices list. In the device screen, you will be able to see some information about your device, like its IP address, connection status, or sent/received information in real time. By default, our device will appear as disconnected, as shown in the picture below.

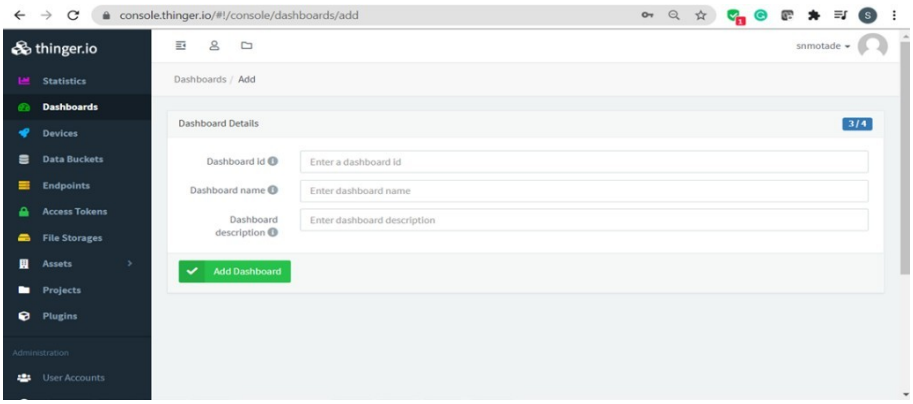


Once the device gets connected to the account, the interface will change its status, showing "Online" status, and some connection data like the IP address or the upload/download data amount:



Add Dashboard:





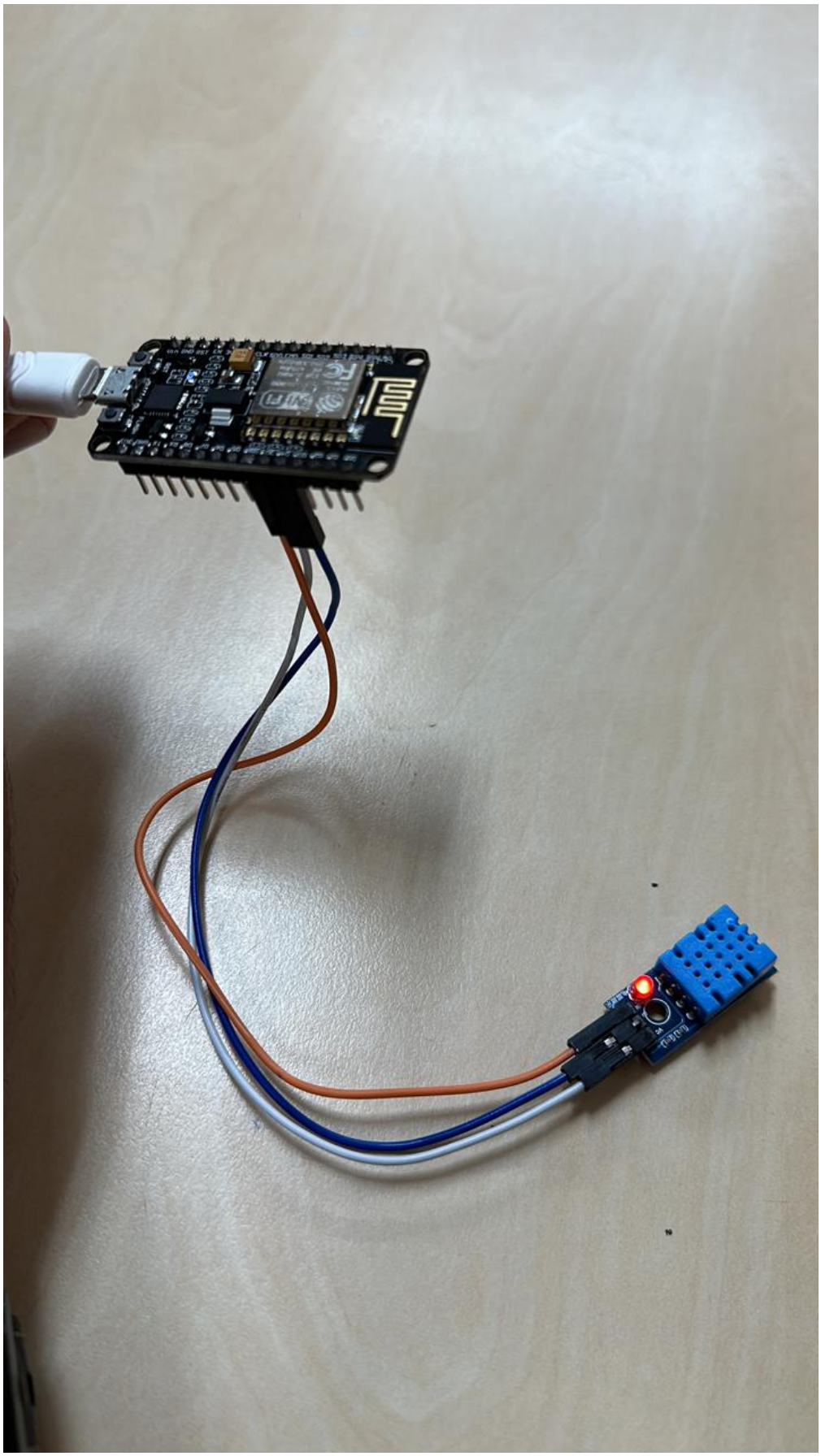
Conclusion:

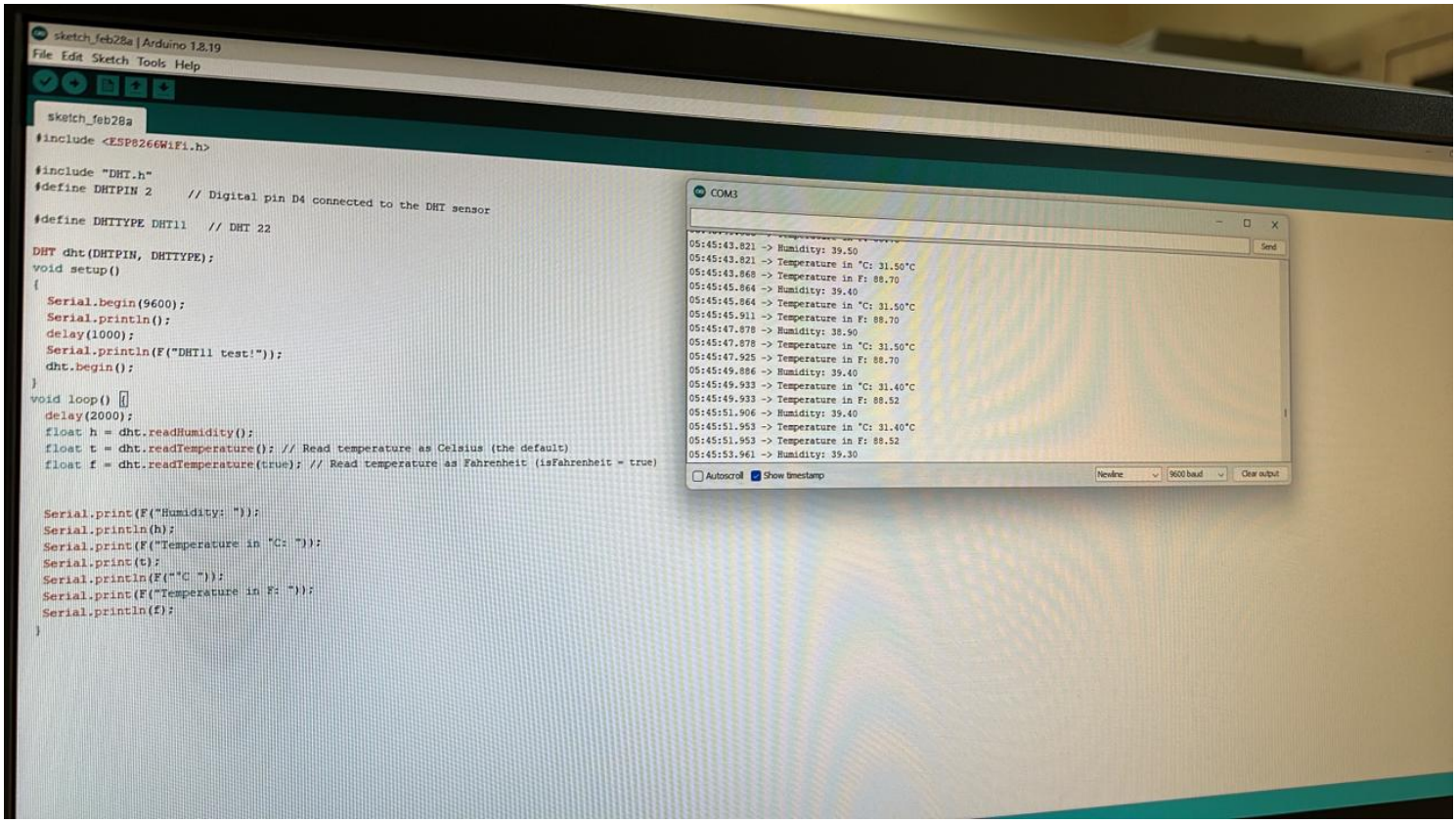
Post Lab Questions:

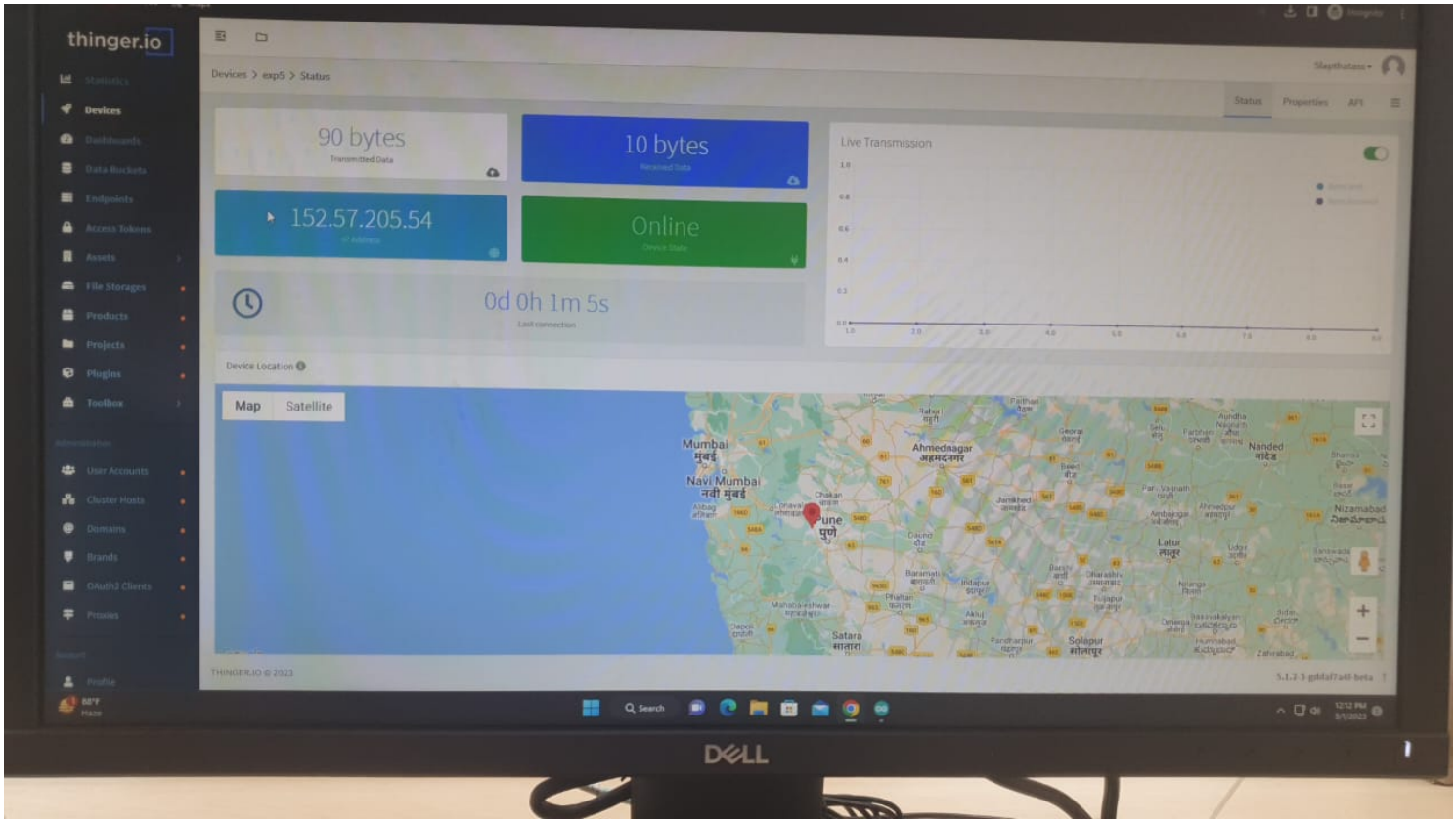
1. List and write features of any one IoT Cloud platforms.
2. Compare between DHT11 and DHT 22
3. What is the role of cloud in IoT?
4. What are the features of Thinger.io

Additional links for more information:

1. Monitoring DHT11 Sensor Data with NodeMCU ESP8266 and Thinger IO
   https://www.youtube.com/watch?v=nWRtz7jRvVE
2. Installing DHT11/DHT22 sensor libraries for Arduino/NodeMCU
   https://iot4beginners.com/installing-dht11-dht22-libraries-for-arduino-nodemcu/
3. DHT11 Temperature & Humidity sensor on NodeMCU using Arduino IDE
   https://roboindia.com/tutorials/dht11-nodemcu-arduino/
4. Thinger.io Documentation
   https://docs.thinger.io/features




sketch_feb28a | Arduino 1.8.19
File Edit Sketch Tools Help
sketch_feb28a
#include <ESP8266WiFi.h>
#include "DHT.h"
#define DHTPIN 2 // Digital pin D4 connected to the DHT sensor
#define DHTTYPE DHT11 // DHT 22
DHT dht(DHTPIN, DHTTYPE);
void setup()
{
Serial.begin(9600);
Serial.println();
delay(1000);
Serial.println(F("DHT11 test!"));
dht.begin();
}
void loop() {
delay(2000);
float h = dht.readHumidity();
float t = dht.readTemperature(); // Read temperature as Celsius (the default)
float f = dht.readTemperature(true); // Read temperature as Fahrenheit (isFahrenheit = true)
Serial.print(F("Humidity: "));
Serial.println(h);
Serial.print(F("Temperature in °C: "));
Serial.print(t);
Serial.println(F("°C "));
Serial.print(F("Temperature in F: "));
Serial.println(f);
}
COM3
Send
05:45:43.821 -> Humidity: 39.50
05:45:43.821 -> Temperature in °C: 31.50°C
05:45:43.868 -> Temperature in F: 88.70
05:45:45.864 -> Humidity: 39.40
05:45:45.864 -> Temperature in °C: 31.50°C
05:45:45.911 -> Temperature in F: 88.70
05:45:47.878 -> Humidity: 38.90
05:45:47.878 -> Temperature in °C: 31.50°C
05:45:47.925 -> Temperature in F: 88.70
05:45:49.886 -> Humidity: 39.40
05:45:49.933 -> Temperature in °C: 31.40°C
05:45:49.933 -> Temperature in F: 88.52
05:45:51.906 -> Humidity: 39.40
05:45:51.953 -> Temperature in °C: 31.40°C
05:45:51.953 -> Temperature in F: 88.52
05:45:53.961 -> Humidity: 39.30
Autoscroll
Show timestamp
Newline
9600 baud
Clear output


thinger.io
Devices > exp5 > Status
Status
Properties
API
90 bytes
Transmitted Data
10 bytes
Received Data
152.57.205.54
IP Address
Online
Device State
0d 0h 1m 5s
Last connection
Live Transmission
Device Location
Map
Satellite
Mumbai
मुंबई
Navi Mumbai
नवी मुंबई
Ahmednagar
अहमदनगर
Pune
पुणे
Satara
सातारा
Solapur
सोलापूर
Latur
लातूर
Nanded
नांदेड
Nizamabad
Devices
Dashboards
Data Buckets
Endpoints
Access Tokens
Assets
File Storages
Products
Projects
Plugins
Toolbox
User Accounts
Cluster Hosts
Domains
Brands
OAuth2 Clients
Proxies
Profile
THINGER.IO © 2023
DELL

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

01/03/2023

MIT-WPU
MIT WORLD PEACE UNIVERSITY | PUNE

Exp no 5

* Post Lab Questions -

Q1) List & write features of any one IoT cloud platforms -

→ Device Management -

AWS IoT provides device management features that enable users to securely register, organize, & manage their IoT devices at scale. It also provides features such as over-the-air updates, remote device monitoring, and management of device configurations & policies. AWS IoT is a comprehensive IoT cloud platform that provides a range of features and capabilities for developing and deploying IoT solutions. Its device management, data management security & compliance & edge computing capabilities make it a powerful & flexible platform for building scalable & secure IoT solutions.

Q2) Compare between DHT11 & DHT22

	Parameter	DHT11	DHT22
①	Temp range	0° to 50°C	−40° to 80°C
②	Temp accuracy	±1-2°C	±0.5°C
③	Relative humidity range	30% to 90%	0% to 100%
④	Humidity accuracy	±4 to 5%	±2 to 5%
⑤	Resolution	8bits	16bits
⑥	Sampling period	≥1s	≥2s
⑦	Price / cost	Low cost	Higher cost

Q3) What is the role of cloud in IoT?

→ The Internet of things generates a huge amount of data per second. cloud computing helps in storing & analyzing this data to optimise IoT infrastructure. It also helps in the modernisation of operations by connecting legacy & smart devices, and machines to the internet & reducing the barriers between IT & OT teams with a unified view of systems. & data. It also reduces the extensive deployment costs for hardware or configure & manage networks & infrastructure relating to IoT.

Q4) What are the features of Thinger.io?

-> Features of thinger.io are -

1. Private cloud: It deploys in a flexible way and adapts to each project needs
2. Projects managements: All assets on thinger.io can be organised in terms of projects.
3. Allows many IoT device connection: It accepts almost any device in the market
4. store, view and analyse'.
One can see real time and historic data on its dashboard which can be further customised according to user requirement for analysis.

Second Year B. Tech (EL & CE)

Semester: IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY**

Roll No: 29 **Batch: A2**

Experiment No: 06

Name of the Experiment: **Understanding Raspberry-Pi as single board Computer and exploring GPIO.**

Performed on: 15/03/2023

Submitted on: 19/03/2023

Marks	Teacher's Signature with date

Aim: Understanding Raspberry-Pi as single board Computer and exploring GPIO.

Prerequisite: Basics of Raspberry Pi.

.

Objective:

1. Understand basics of Raspberry Pi as single board computer
2. Understand the layout of Raspberry pi
3. Understand the GPIO pin out of Raspberry Pi

Components and equipment required:

Raspberry Pi Board.

Theory:

Raspberry Pis are single-board computers (SBCs), meaning the memory, interface as well as processor are soldered into one circuit board that numbers 4 revisions and a minimalistic zero variant. The pocket-size computers are built to make learning programming languages fun. Raspberry Pi is a series of SBCs developed in the United Kingdom by the Raspberry Pi Foundation in association with Broadcom. The Raspberry Pi project originally leaned towards the promotion of teaching basic computer science in schools and in developing countries. The original model became more popular than anticipated, selling outside its target market for uses such as robotics. It is widely used in many areas, such as for weather monitoring, because of its low cost, modularity, and open design. It is typically used by computer and electronic hobbyists, due to its adoption of HDMI and USB devices. The main difference between a Raspberry Pi (tiny computer) and other computers is the GPIO (General Purpose Input Output) pins. The GPIO pins are one way in which the Raspberry Pi can control and monitor the outside world by being connected to electronic circuits.

Hardware

- A Raspberry Pi computer with an SD card or micro SD card
- A monitor with a cable (and, if needed, an HDMI adaptor)
- A USB keyboard and mouse
- A power supply
- Headphones or speakers (optional)
- An ethernet cable (optional)

Software

Raspberry Pi OS, installed using the Raspberry Pi Imager

Series and generations:

There are three series of Raspberry Pi, and several generations of each have been released. Raspberry Pi SBCs feature a Broadcom system on a chip (SoC) with an integrated ARM-compatible central processing unit (CPU) and on-chip graphics processing unit (GPU).

Figure 6.1: Raspberry Pi 4 Model B

Raspberry Pi Family with specification

Family	Model	SoC	Memory	Form Factor	Ethernet	Wireless	GPIO	Released	Discontinued
Raspberry Pi	B	BCM2835	256 MB	Standard	Yes	No	26-pin	2012	Yes
Raspberry Pi	A	BCM2835	256 MB	Standard	No	No	26-pin	2013	No
Raspberry Pi	B+	BCM2835	512 MB	Standard	Yes	No	40-pin	2014	No
Raspberry Pi	A+	BCM2835	512 MB	Compact	No	No	40-pin	2014	No

Raspberry Pi Zero	Zero	BCM2835	512 MB	Ultra-Compact	No	No	40-pin	2015	No
Raspberry Pi 2	B	BCM2836/7	1 GB	Standard	Yes	No	40-pin	2015	No
Raspberry Pi 3	B	BCM2837A0/B0	1 GB	Standard	Yes	Yes	40-pin	2016	No
Raspberry Pi Zero	W/WH	BCM2835	512 MB	Ultra-Compact	No	Yes	40-pin	2017	No
Raspberry Pi 3	A+	BCM2837B0	512 MB	Compact	No	Yes (dual band)	40-pin	2018	No
Raspberry Pi 3	B+	BCM2837B0	1 GB	Standard]	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2018	No
Raspberry Pi 4	B	BCM2711	1 GB	Standard	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2019[43]	March 2020[1]
Raspberry Pi 4	B	BCM2711	2 GB	Standard	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2019[43]	No
Raspberry Pi 4	B	BCM2711	4 GB	Standard	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2019[43]	No
Raspberry Pi 4	B	BCM2711	8 GB	Standard	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2020	No

Raspberry Pi 4	400	BCM2711	4 GB	Keyboard	Yes (Gigabit Ethernet)	Yes (dual band)	40-pin	2020	No
Raspberry Pi Pico	N/A	RP2040	264 KB	Pico (21 mm × 51 mm)	No	No	26-pin	2021	?

The Pi can control LEDs, turning them on or off, drive motors, and interact with many other objects. It can also detect the pressing of a switch, change in temperature, or light, etc, by attaching kinds of sensors. These pins are a physical interface between the Raspberry Pi and the outside world. Using them, you can program the Raspberry Pi to switch devices on and off (output), or receive data from sensors and switches (input). Of the 40 pins, 26 are GPIO pins and the others are power or ground pins (plus two ID EEPROM pins which you should not play with unless you know your stuff!)

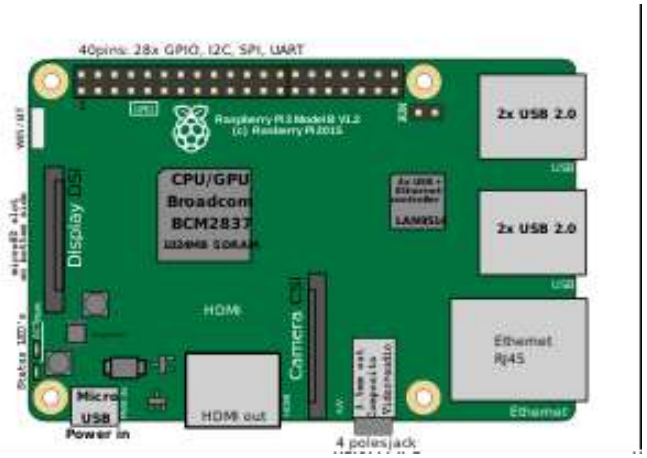


Figure 6.2: Location of Connectors & main ICs on Raspberry Pi 3

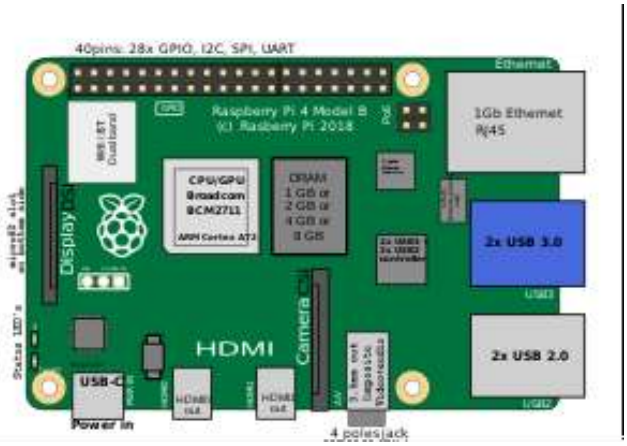


Figure 6.3: Location of Connectors & main ICs on Raspberry Pi 4

General purpose input-output (GPIO) connector

Raspberry Pi 1 Models A+ and B+, Pi 2 Model B, Pi 3 Models A+, B and B+, Pi 4, and Pi Zero, Zero W, and Zero WH GPIO J8 have a 40-pin pinout. Raspberry Pi 1 Models A and B have only the first 26 pins.

In the Pi Zero and Zero W, the 40 GPIO pins are unpopulated, having the through-holes exposed for soldering instead. The Zero WH (Wireless + Header) has the header pins preinstalled.

GPIO#	2nd func.	Pin#	Pin#	2nd func.	GPIO#
	+3.3 V	1	2	+5 V	
2	SDA1 (I^2C)	3	4	+5 V	
3	SCL1 (I^2C)	5	6	GND	
4	GCLK	7	8	TXD0 (UART)	14
	GND	9	10	RXD0 (UART)	15
17	GEN0	11	12	GEN1	18
27	GEN2	13	14	GND	
22	GEN3	15	16	GEN4	23
	+3.3 V	17	18	GEN5	24
10	MOSI (SPI)	19	20	GND	
9	MISO (SPI)	21	22	GEN6	25
11	SCLK (SPI)	23	24	CE0_N (SPI)	8
	GND	25	26	CE1_N (SPI)	7
(Pi 1 Models A and B stop here)					
0	ID_SD (I^2C)	27	28	ID_SC (I^2C)	1
5	N/A	29	30	GND	
6	N/A	31	32	N/A	12
13	N/A	33	34	GND	
19	N/A	35	36	N/A	16
26	N/A	37	38	Digital IN	20
	GND	39	40	Digital OUT	21

Model B rev. 2 also has a pad (called P5 on the board and P6 on the schematics) of 8 pins offering access to an additional 4 GPIO connections. These GPIO pins were freed when the four board version identification links present in revision 1.0 were removed.

GPIO#	2nd func.	Pin#		Pin#	2nd func.	GPIO#
	+5 V	1		2	+3.3 V	
28	GPIO_GEN7	3		4	GPIO_GEN8	29
30	GPIO_GEN9	5		6	GPIO_GEN10	31
	GND	7		8	GND	

Raspberry Pi Tutorial: Raspberry Pi Components

Now, let's have a look at the different components of the Raspberry Pi 3 – B model:

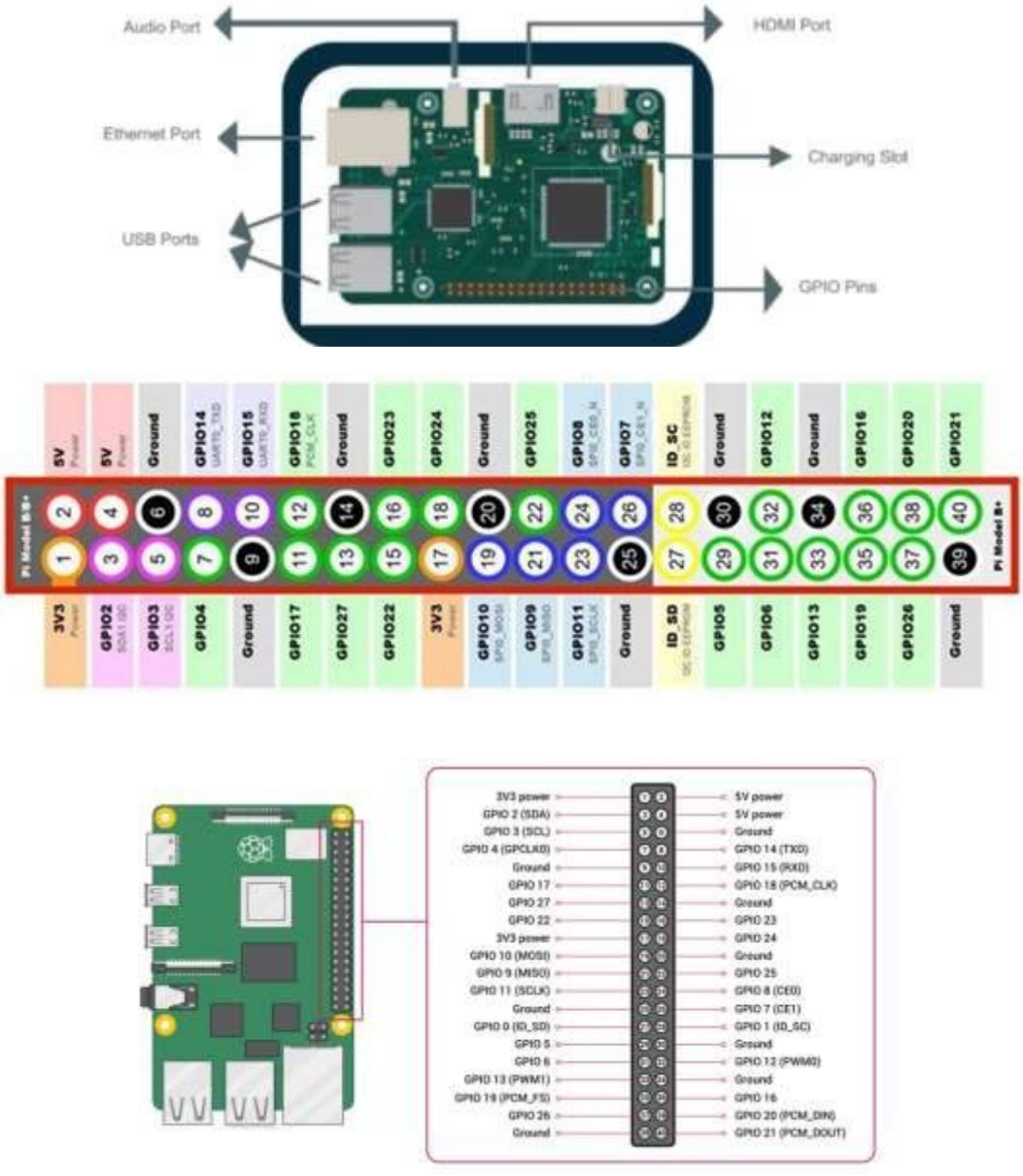


Figure 6.4: GPIO Pinout Diagram

GPIO Functions

Both the Raspberry Pi3 and Pi4 offer these functions:

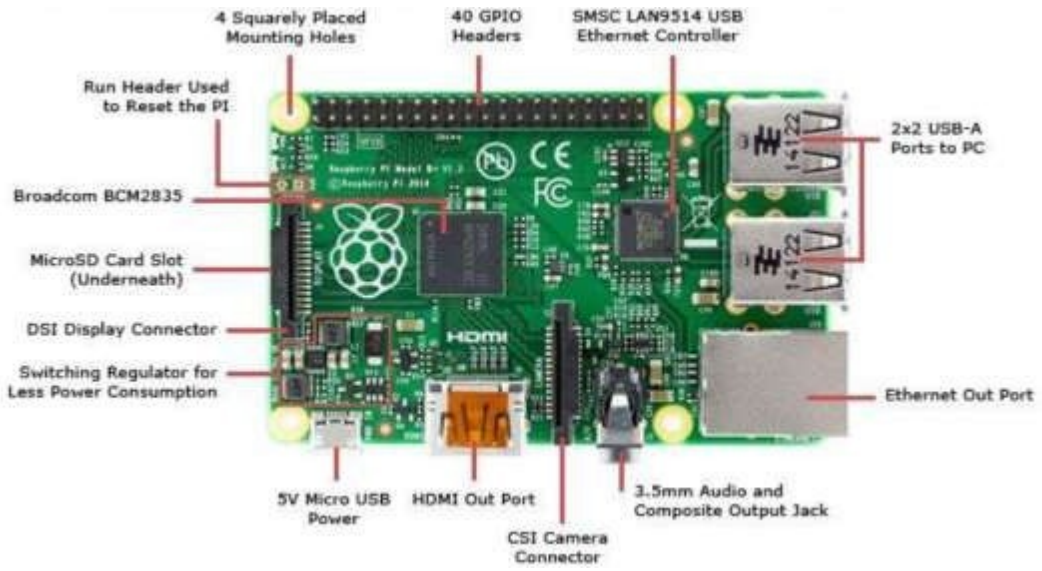


Figure 6.5: Raspberry Pi Board Layout

- **GPIO** is your standard pins that simply be used to turn devices on and off. For example, a LED.
- **I2C** (Inter-Integrated Circuit) pins allow you to connect and talk to hardware modules that support this protocol (I2C Protocol). This protocol will typically take up two pins. The GPIO 2 SDA and GPIO 3 SCL allow the connection via the I2C protocol. I2C creates a serial, synchronous communication with multiple server/client connection of up to 128 devices.
- **SPI** (Serial Peripheral Interface Bus) pins can be used to connect and talk to SPI devices. Pretty much the same as I2C but makes use of a different protocol. The SPI protocol is a synchronous serial communication between one server and several clients, where both can actively send data. To make an SPI outbound connection, 4 pins are needed. The Raspberry Pi allows to make either 2 separate Server connections, or one server, one client connections. For the 1st GPIO subsystem, GPIO 9, 19, 11, 8 and 16 are used, and for the 2nd GPIO subsystem its 19, 20, 21 and 16.
- **UART** (Universal asynchronous receiver/transmitter) is the serial pins used to communicate with other devices. GPIO 10 (RX) and GPIO 8 TX can be used to directly connect two devices with the UART protocol, which enables serial, asynchronous communication.
- **PWM:** Pulse width modulation is a signaling technique in which a digital signal is switching rapidly between two states high and low in a certain frequency. The Raspberry pi has no native support for PWM, but there are Libraries that provide Software PWM, although with restrictions compared to Arduino.
- **DNC** stands for do not connect, this is pretty self-explanatory.

- The **power** pins pull power directly from the Raspberry Pi.
- **GND** are the pins you use to ground your devices. It doesn't matter which pin you use as they are all connected to the same line.

Connect your Raspberry Pi

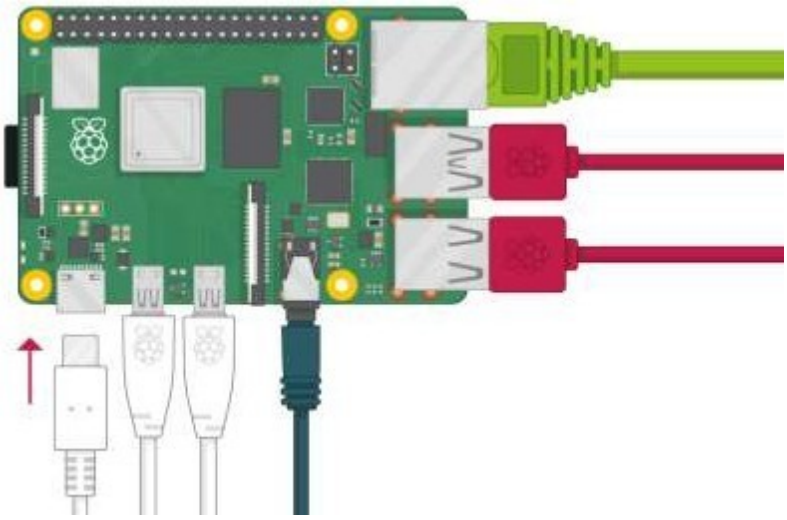


Figure 6.6: Raspberry Pi Connectors

- Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside. If no SD card is there, then insert an SD card with Raspbian installed (via NOOBS).
- Find the USB connector end of your mouse's cable, and connect the mouse to a USB port on your Raspberry Pi (it doesn't matter which port you use).
- Connect the keyboard in the same way.
- Make sure your screen is plugged into a wall socket and switched on.
- Look at the HDMI port(s) on your Raspberry Pi — notice that they have a flat side on top.
- Use a cable to connect the screen to the Raspberry Pi's HDMI port — use an adapter if necessary.
- Connect your screen to the single HDMI port.
- If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket on the wall or on your internet router. You don't need to do this if you want to use wireless connectivity, or if you don't want to connect to the internet.
- If your screen has speakers, your Raspberry Pi can play sound through these. Or you could connect headphones or speakers to the audio port.
- Plug the power supply into a socket and then connect it to your Raspberry Pi's USB power port.
- You should see a red light on your Raspberry Pi and raspberries on the monitor.

Figure 6.7: Raspberry Pi Graphical Desktop

- Your Raspberry Pi then boots up into a graphical desktop.

4M%veIGWi!ilnEsx

Conclusion:

Post Lab Questions:

1. What is Raspberry Pi? How does it works?
2. What are the different components of the Raspberry Pi board?
3. How is Raspberry Pi used in IoT?
4. How is Raspberry pi different from Arduino and NodeMCU?

Additional links for more information:

1. Raspberry Pi Documentation
   https://www.raspberrypi.com/documentation/computers/os.html
2. Getting Started with Raspberry Pi
   https://projects.raspberrypi.org/en/projects/raspberry-pi-getting-started/3
3. The Raspberry Pi Platform and Python Programming for the Raspberry Pi
   https://www.coursera.org/learn/raspberry-pi-platform#about

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

15/03/2023

Exp no 6

* Post Lab Questions -

Q1) What is Raspberry Pi? How does it works?

→ Raspberry Pi is a small, single-board computer that was created with the goal of promoting computer scierre education and programming skills among people of all ages. It is a credit card-sized computer that can be connected to a keyboard, mouse, and monitor or TV. The Raspberry Pi is based on ARM processors and is designed to run various Linux-based operating systems It has a set of GPIO pins that allow it to interact with external devices and sensors, making it ideal for projects involving robotics, automation, and IOT.

The Raspberry Pi works by running code written in programming languages such as Python, C++, and scratch. The code can be written on the Raspberry Pi itself or on another computer and transferred to the Raspberry Pi via the internet or a USB flash drive. Once the code is loaded onto the Raspberry Pi, it can be executed to perform various tasks.

Q2) what are the different components of the Raspberry Pi board?

→ ① Processor - The Raspberry Pi uses a Broadcom system-on-chip (SoC) that includes a CPU, GPU, and other components.

② Memory - The board has between 1 GB to 8GB of RAM, depending on the model.

③ Storage - Uses a microSD card to store the operating system and user data.

④ Input/output (I/O) ports - The board has several I/O ports, including USB, Ethernet, HDMI, audio and a 40-pin GPIO header for interfacing

⑤ Power supply - It can be powered using a micro-USB port or GPIO pins.

⑥ Ethernet & wifi - Some models have Ethernet and/or wifi built-in for network connectivity.

⑦ Camera & display connectors - The board has a dedicated connector for connecting a Raspberry Pi camera module, as well as a connector for connecting a display via the Display Serial Interface (DSI).

Q3) How is Raspberry used in IoT?

→ Uses -

① Home automation -

The Raspberry Pi can be used to control smart home devices, such as lights, thermostats, and security cameras. It can be connected to sensors and actuators to monitor and control various aspects of home.

② Industrial automation -

The Raspberry Pi can be used to control machinery, monitor environmental conditions, and automate production lines in industrial settings.

③ Environmental monitoring -

The Raspberry Pi can be used to collect data from sensors that measure environmental conditions, such as temp, humidity & air quality.

④ Smart agriculture -

The Raspberry Pi can be used to monitor soil moisture levels, temp, & humidity in agricultural settings. This data can be used to optimize crop yield and reduce water usage.

Q4) How is Raspberry Pi different from Arduino and NodeMCU?

→ Differences -

① Processing power -

The Raspberry Pi have significantly more processing power than Arduino or NodeMCU, as it is based on a full-fledged CPU and can run a full operating system. This makes it well-suited for applications that require complex computations or multitasking

② GPIO pins -

All three platforms have GPIO pins that can be used to connect sensors, actuators, and other devices. However, the Raspberry Pi has a larger number of GPIO pins (40) than Arduino or NodeMCU, which have 20 and 11 pins, respectively.

③ Programming languages:

Arduino uses a simplified version of C++, while NodeMCU uses Lua, a lightweight scripting language. The Raspberry Pi can run a wide variety of programming languages including Python, C++, Java, and more.

Second Year B. Tech (EL & CE)

Semester: IV **Subject:** Basic IoT Laboratory

Name: Shreerang Mhatre **Class: SY B.Tech**

Roll No: 29 **Batch: A2**

Experiment No: 07

Name of the Experiment: Interfacing the LCD screen with hardware platforms.

Performed on: 24/03/2023

Submitted on: 10/04/2023

Marks	Teacher's Signature with date

Aim: Interfacing of LCD screen with hardware platforms

Prerequisite: Arduino board, Arduino IDE, Basics LCD screen.

Objectives:

1. To understand usefulness of the LCD screen

Components and equipment required:

Arduino Uno Board, USB cable, Arduino IDE, LCD screen, 270Ω Resistor, 10 K Potentiometer, Bread board, Jumper wires

Theory:

LCD means Liquid Crystal Display. We use LCD technology today in watches, digit code display and in TV and advert screens. There exist two big families of LCD displays:
1-Character LCD is based on a matrix of characters (columns x rows).
2- Graphical LCD it is based on a pixel matrix.
We can find many printed circuit boards that include an LCD and the connectors to interface them with Arduino and other systems such that Raspberry pi or PIC microcontroller, nowadays.

There is now a library included in the Arduino Core that is really so easy to use. Its name is **"Liquid Crystal"**, and it works with all LCD displays that are compatible with the Hitachi HD44780 driver. This driver is common. Hitachi developed it as a

very dedicated driver, that includes a micro-controller itself, specifically to drive alphanumeric characters LCDs and to connect to the external world easily too, which can be done by a specific link using, usually, 16 connectors, including power supply for the external circuit itself and the backlightsupply too.

These instructions presence in the library of the display:

#include<LiquidCrystal.h>

- **lcd.begin(16,2):** This instruction use to set up the LCD's number of columns and rows.
- **lcd.print("Message"):** This instruction used for print a message on the LCD if need to display numbers must remove the double quotationmark like this {lcd.print(var)}.
- **lcd.setCursor(j, i):** This instruct used for determine site scripting as row and column where,(j) represent the column and (i) represent the row. The figure below shows us the locations of rows and columns in the 2x16 Liquid Crystal Display. The figure (2.1) below shows the LCD row and column arrangements.

- **lcd.clear( ):** This instruction used to clear the screen .
- **Delay (n):** This instruct used to give delay time where, (n) is an integer number in millisecond. This can used without having to call the library of the screen.
-

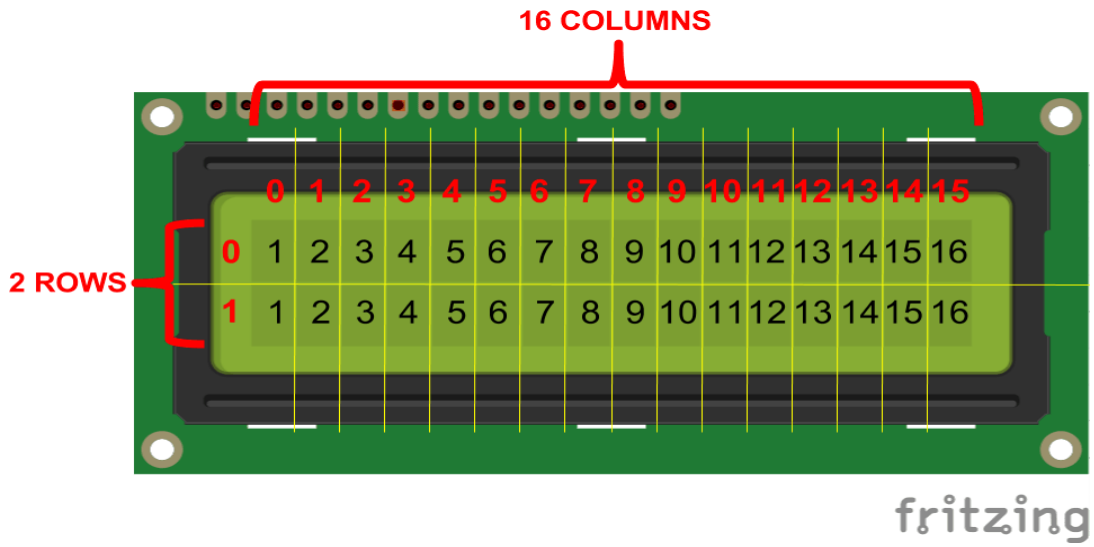


Fig 7.1

- **LCD pin configuration:**

- The connections of the LCD screen are describe as the following below:

- PIN1 or VSS to ground
- PIN2 or VDD or VCC to +5v power
- PIN3 or VEE to ground (gives maximum contrast best for a beginner)
- PIN4 or RS (Register Selection) to PIN0 of ARDUINO UNO
- PIN5 or RW (Read/Write) to ground (puts LCD in read mode easesthe communication for user)
- PIN6 or E (Enable) to PIN1 of ARDUINO UNO
- PIN11 or D4 to PIN8 of ARDUINO UNO
- PIN12 or D5 to PIN9 of ARDUINO UNO
- PIN13 or D6 to PIN10 of ARDUINO UNO
- PIN14 or D7 to PIN11 of ARDUINO UNO
- PIN 15and 16 for background light.

Apparatus:

1- Breadboard.

2- ArduinoUNO.

3- Potentiometer 1 KΩ.4-

Jumper wires.

5-LCD 2x16.

Procedure:

1-Connect the circuit shown in fig.7.2 shown below.

2-write Arduino program to display the phrase “System DEP." on the LCD at the First row.

3- Verify the Arduino sketch.

4-Download the Arduino sketch to the Arduino UNO board.

5-Test the circuit and write down the Arduino code on result paper.

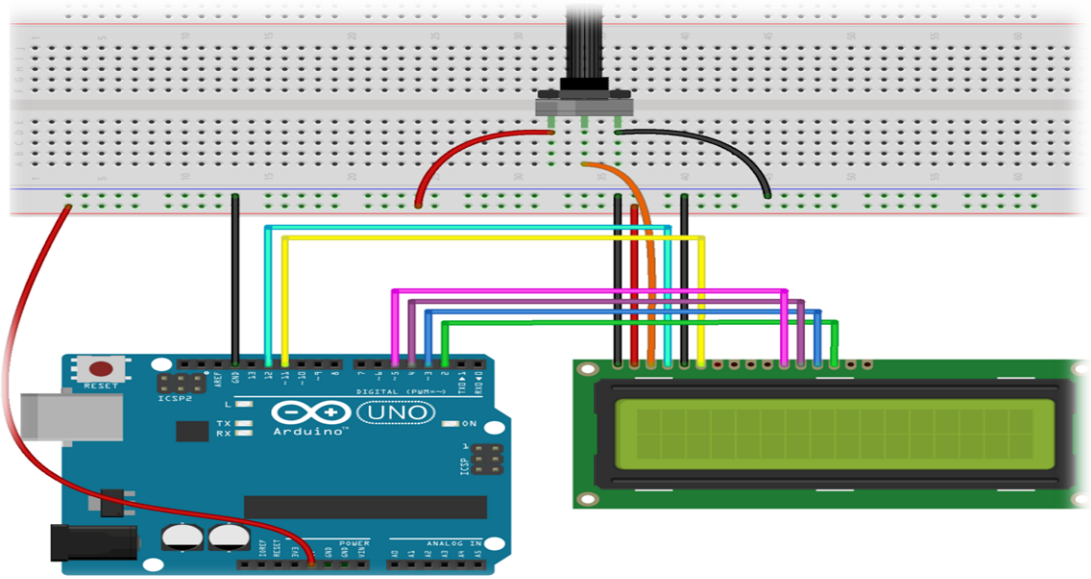


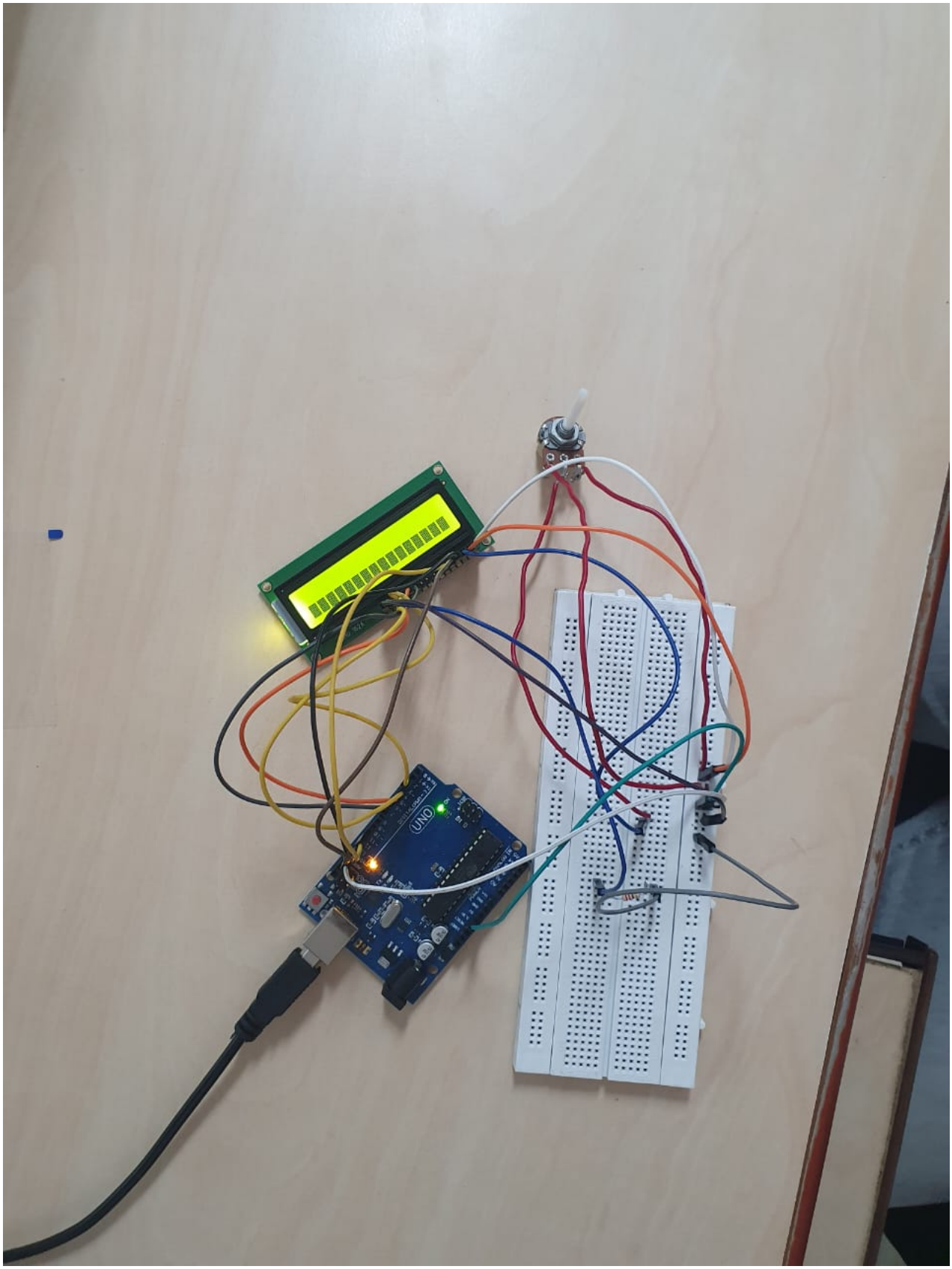
Figure (7.2) practical circuit diagram

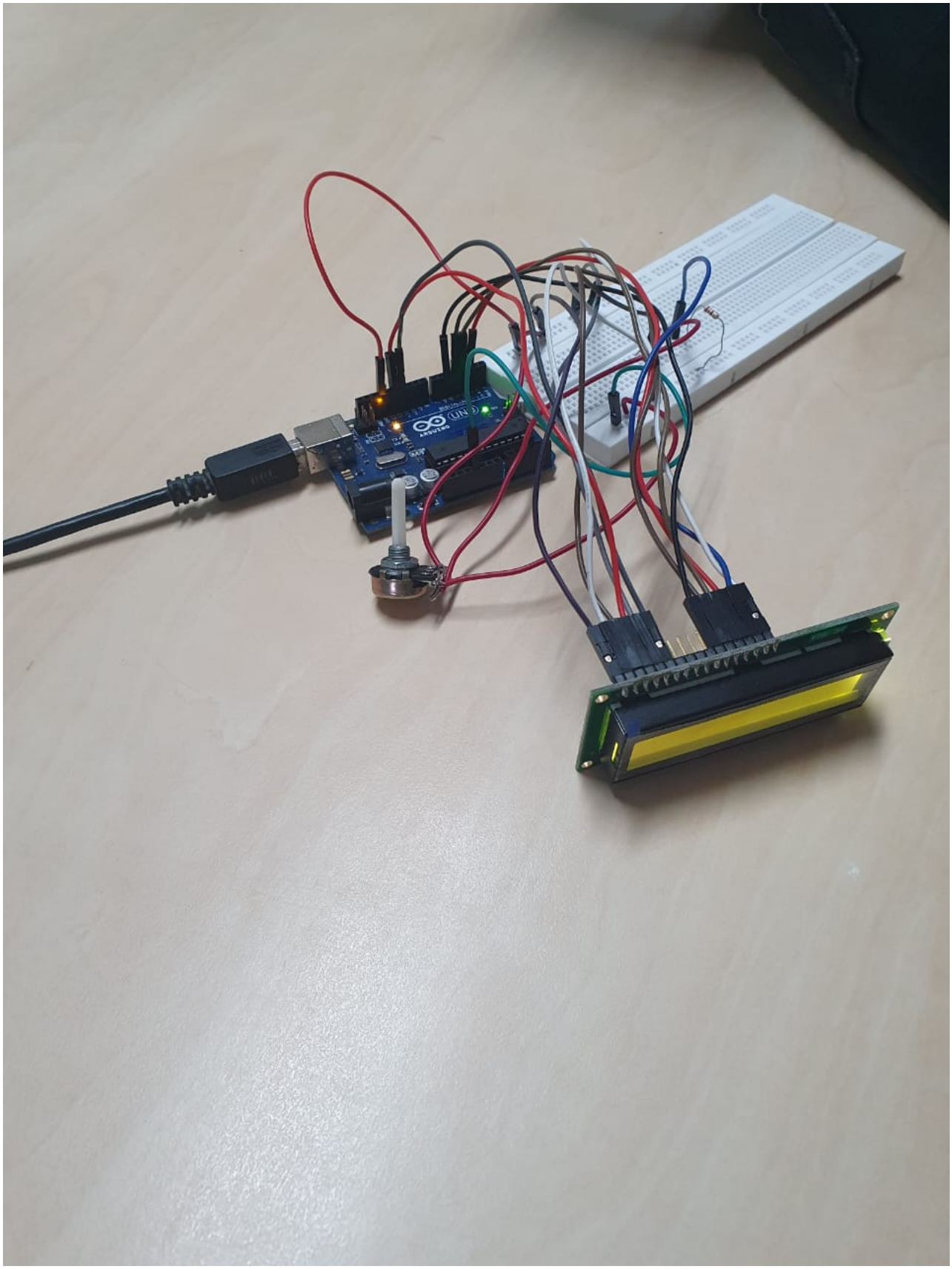
Program:

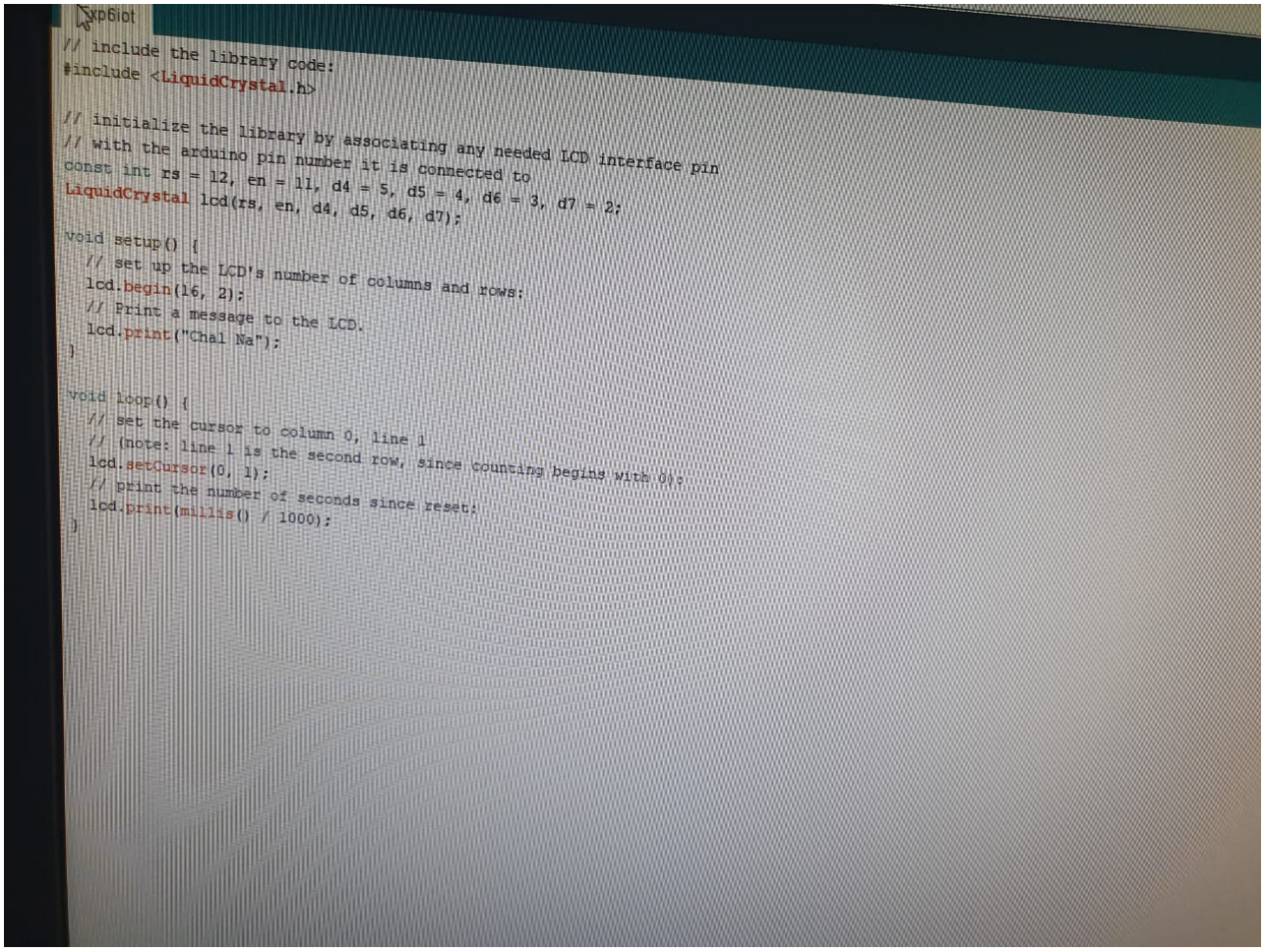
Conclusion:

Post Lab Questions:

1. What is the benefits of using potentiometer that connected on the pin(3)of the LCD?
2. Write down the pin configuration of LCD with diagram.






xp6iot
// include the library code:
#include <LiquidCrystal.h>
// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
void setup() {
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
// Print a message to the LCD.
lcd.print("Chal Na");
}
void loop() {
// set the cursor to column 0, line 1
// (note: line 1 is the second row, since counting begins with 0):
lcd.setCursor(0, 1);
// print the number of seconds since reset:
lcd.print(millis() / 1000);
}

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

08/04/2023

Exp no. 7

* Post Lab Questions -

Q1) What is the benefits of using potentiometer that connected on the pin (3) of the LCD?

→ Benefits of using potentiometer that is connected to the pin (3) of an LCD -

① Better readability -
By adjusting the contrast, you can make the display more readable, especially in different lighting conditions.

② Energy-saving -
By adjusting the contrast you can reduce the power consumption of the LCD module

③ Easy to use -
The potentiometer connected to pin 3 is easy to use, and you can adjust the contrast by simply turning the knob of the potentiometer.

Q2) write down the pin configuration of LCD with diagram.

→ * Pin configuration -

1 → Vss (Ground): Connect to ground
2 → VDD (+5V): Connect to +5V power supply
3 → VO (contrast): connect to a potentiometer to adjust contrast
4 → RS (Register Select): used to select between data and instruction registers
5 → RW (Read/write): used to select between read and write modes
6 → E (Enable): Used to enable data red/write
7 → D0 (Data Bit 0): Used in 8-bit mode
8 → D1 (Data Bit 1): Used in 8-bit mode
9 → D2 (Data Bit 2): Used in 8-bit mode
10 → D3 (Data Bit 3): Used in 8-bit mode
11 → D4 (Data Bit 4): Used in 4-bit & 8-bit modes
12 → D5 (Data Bit 5): Used in 4bit & 8-bit modes
13 → D6 (Data Bit 6): Used in 4bit & 8-bit modes
14 → D7 (Data Bit 7): Used in 4bit & 8-bit modes
15 → A (Anode): connect to +5V power supply for backlight (if available)
16 → K (cathode): Connect to Ground for back light (if available)

Second Year B. Tech (EL&CE)

Semester: IV **Subject:** Basic IoT Lab

Name: Shreerang Mhatre **Class: Sy**

Roll No: 29 **Batch: A2**

Experiment No: 08

Name of the Experiment: To introduce new hardware platforms TinyML

Marks	Teacher's Signature with date

Performed on: 19/04/23

Submitted on: 28/04/23

Aim: To introduce hardware platforms for IoT based design TinyML

Pre-requisite: Knowledge of Electronics and circuits, Knowledge of Assembly and C Programming language.

Objective:

1. To introduce new hardware platforms for IoT based design.
2. To understand different features of new hardware platforms.

Components and equipment required:

Arduino Uno Board, TinyML

Theory:

IoT hardware platforms are considered as the most significant component of the IoT environment. One IoT device can be connected to other IoT device to enable the information exchange using suitable Internet protocols. IoT platforms works as a bridge between the device sensors and data networks. It gives data connectivity to the sensor system and helps to understand using back-end applications and helps to analyse the huge data generated by sensors.

What is TinyML

TinyML is a field of study in Machine Learning and Embedded Systems that explores the types of models you can run on small, low-powered devices like microcontrollers. It enables low-latency, low power and low bandwidth model inference at edge devices. While a standard consumer CPUs consume between 65 watts and 85 watts and standard consumer GPU consumes anywhere between 200 watts to 500 watts, a typical microcontroller consumes power in the order of milliwatts or microwatts. That is around a thousand times less power consumption. This low power consumption enables the TinyML devices to run unplugged on batteries for weeks, months, and in some cases, even years, while running ML applications on edge.

Advantages of TinyML

1. **Low Latency:** Since the model runs on the edge, the data doesn't have to be sent to a server to run inference. This reduces the latency of the output.
2. **Low Power Consumption:** As we discussed before, microcontrollers consume very little power. This enables them to run without being charged for a really long time.
3. **Low Bandwidth:** As the data doesn’t have to be sent to the server constantly, less internet bandwidth is used.
4. **Privacy:** Since the model is running on the edge, your data is not stored in any servers.

.

Applications of TinyML

By summarizing and analyzing data at the edge on low power devices, TinyML offers many unique solutions. Even though TinyML is an emerging field, it has been used in production for years. The “OK Google”, “Alexa”, “Hey Siri” wake words are an example of TinyML. Here, the devices are always on and are analyzing your voice to detect the wake word. I’ll add some more applications of TinyML here.

1. **Industrial Predictive Maintenance:** Machines are prone to fault. Using TinyML on low powered devices, it is possible to monitor the machine and predict faults ahead of time constantly. This predictive maintenance can lead to significant cost savings. Ping Services, an Australian startup, has introduced an IoT device that autonomously monitors wind turbines by magnetically attaching to the outside of the turbine and analyzing detailed data at the edge. This device can alert the authorities regarding potential issues even before it occurs.

2. **Healthcare:** The Solar Scare Mosquito project uses TinyML to curb the spread of mosquito-borne diseases like Dengue, Malaria, Zika Virus, Chikungunya, etc. It works by detecting the mosquito breeding conditions and agitates the water to prevent mosquito breeding. It runs on solar power and can thus run indefinitely.

3. **Agriculture:** The Nuru app helps farmers detect diseases in plants just by taking a picture of it by running Machine Learning models on the device using TensorFlow Lite. Since it works on the device, there is no need for an internet connection. This is a crucial requirement for remote farmers since they might not have proper internet connection in their place.

4. **Ocean Life Conservation:** Smart ML-powered devices are used to monitor whales in real-time in waterways around Seattle and Vancouver to avoid whale strikes in busy shipping lanes.

Description :

1. **Hardware:** The **Arduino Nano 33 BLE Sense** is the suggested hardware for deploying Machine Learning models on edge. It contains a 32-bit ARM Cortex-M4F microcontroller running at 64MHz with 1MB of program memory and 256KB RAM. This microcontroller provides enough horsepower to run TinyML models. The Arduino Nano 33 BLE Sense also contains colour, brightness, proximity, gesture, motion, vibration, orientation, temperature, humidity, and pressure sensors. It also contains a digital microphone and a Bluetooth low energy(BLE) module. This sensor suite will be more than enough for most applications.

2. **Machine Learning Framework:** There are only a handful of frameworks that cater to TinyML needs. Of that, **TensorFlow Lite** is the most popular and has the most community support. Using TensorFlow Lite Micro, we can deploy models on microcontrollers.

3. **Learning Resources:** Since TinyML is an emerging field, there aren't many learning materials as of today. But there are a few excellent materials like Pete Warden and Daniel Situnayake's book, "TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power", Harvard University's Course on TinyML by Vijay Janapa Reddi, and Digikey's blogs and videos on TinyML.

Diagram of TinyML:


SYNTIANT
TinyML
T126A
SYNTIANT
MDP101809F
USAA 2008
ST1003N001
SHA
2050
SFEJH
P1_210805

Key Features

- Neural Decision Processor: NDP101
- Host processor: SAMD21 Cortex-M0+ 32bit low power ARM MCU, including:
- 256KB flash memory
- 32KB host processor SRAM
- Board power supply: 5V micro-USB or 3.7V LiPo battery
- 5 Digital I/Os compatible with Arduino MKR series boards
- 1 UART interface (included in the digital I/O Pins
- 1 I2C interface (included in the digital I/O Pins)
- 2MB on-board serial flash
- 48MHz system clock
- One user defined RGB LED
- uSD card slot (uSD card not included)
- BMI160 6 axis motion sensor
- SPH0641LM4H microphone

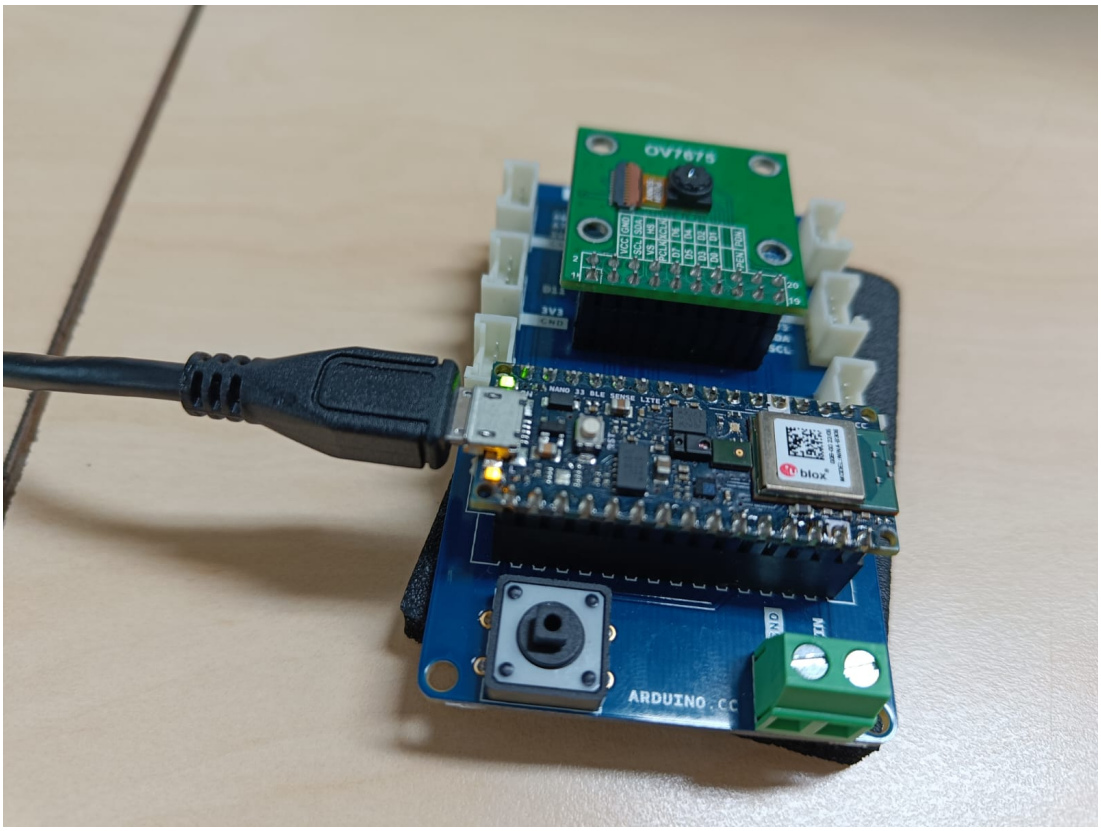
Program CODE :

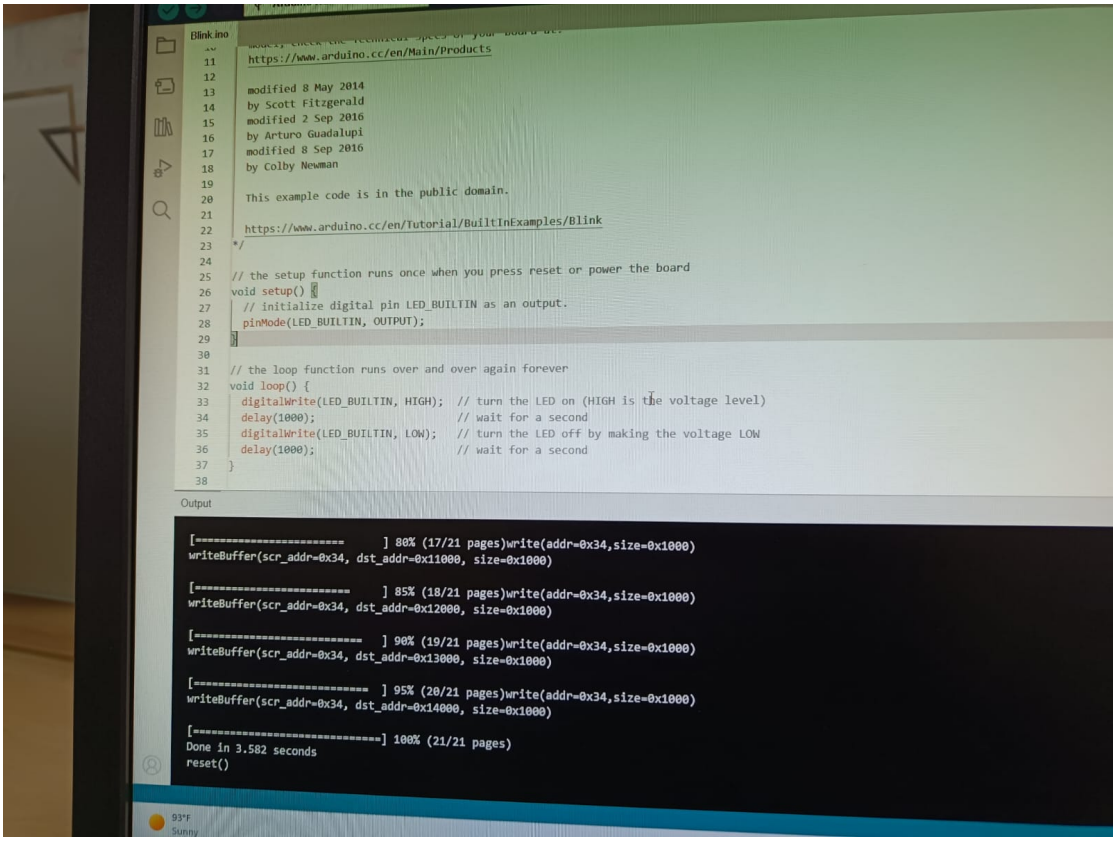
Output :

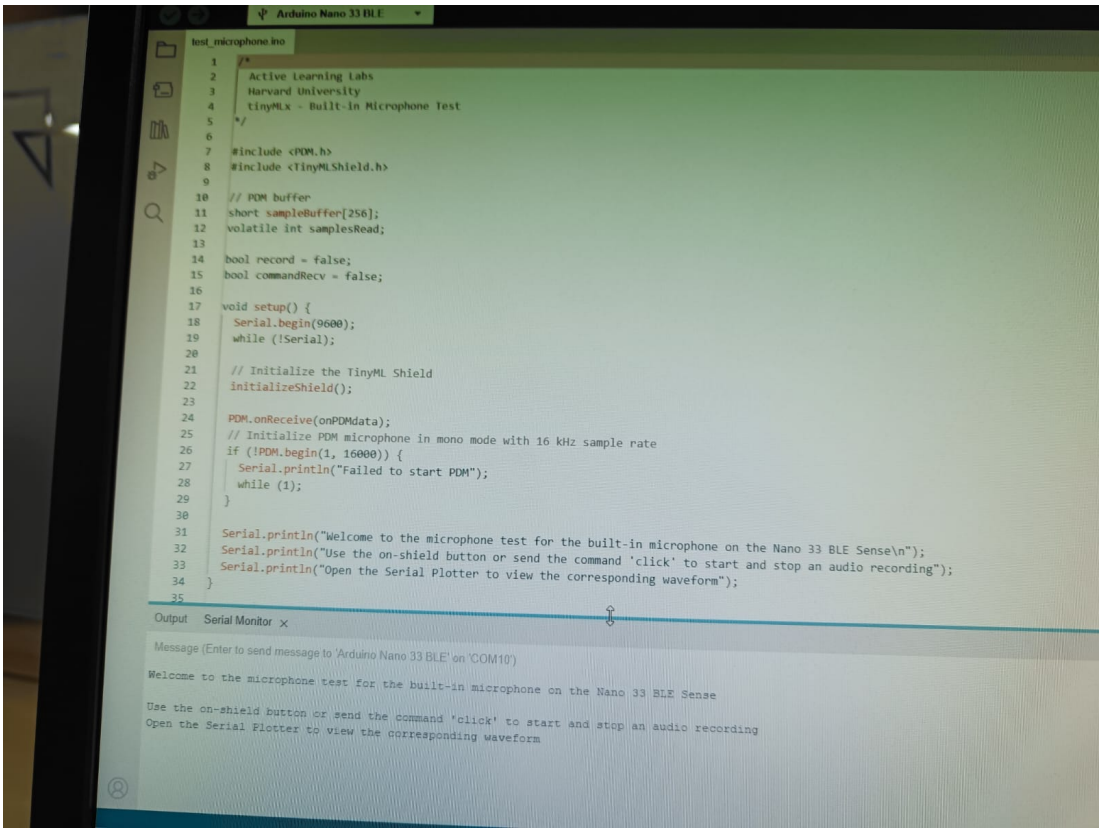
Conclusion :

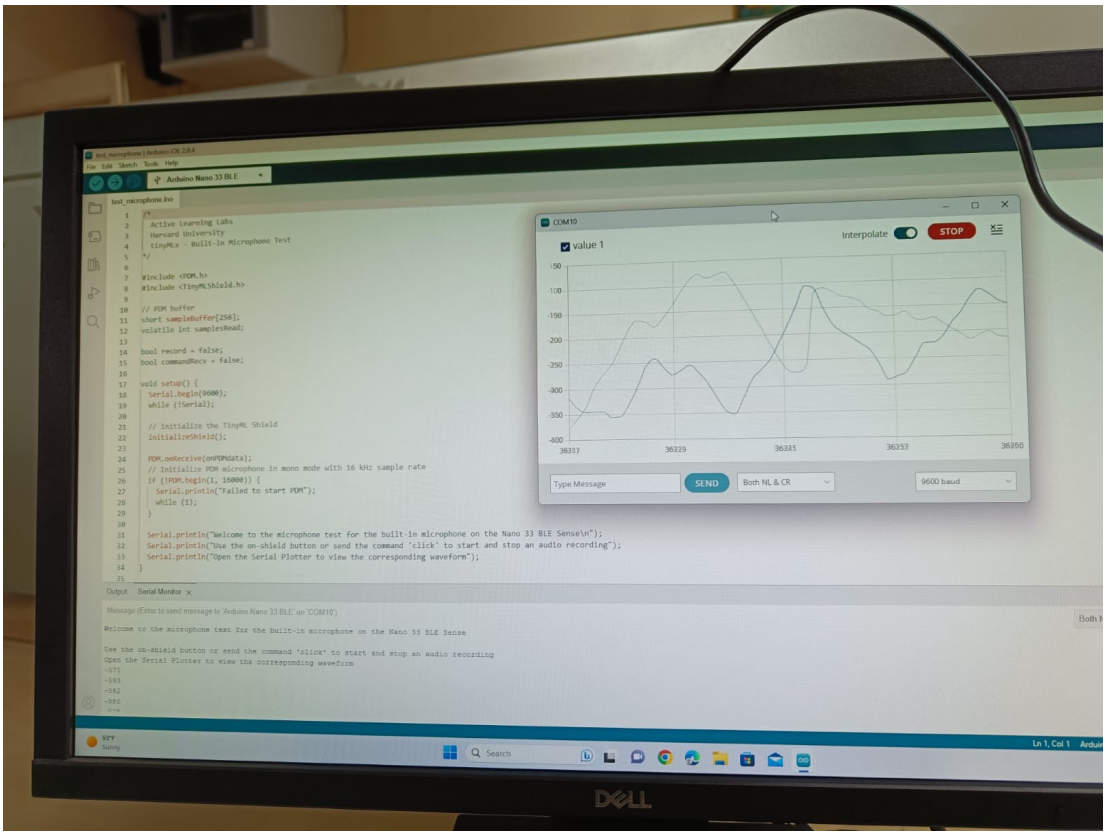
Post lab questions :

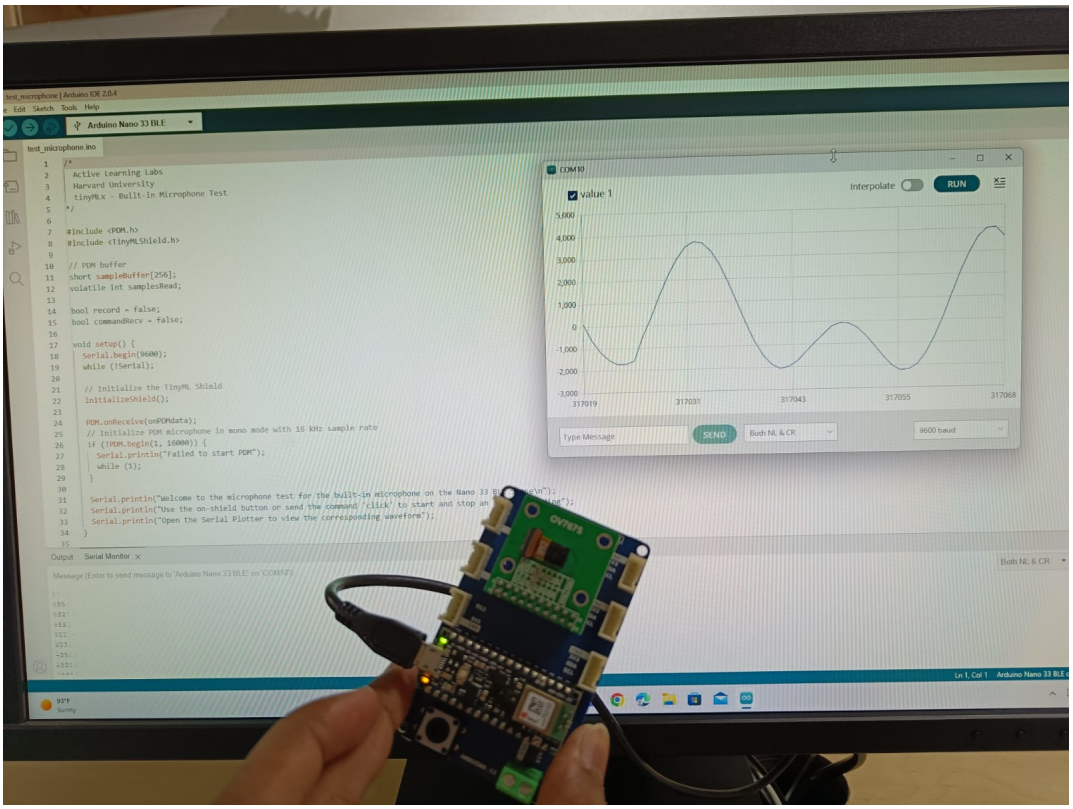
1. Describe different features present in the TinyML kit .

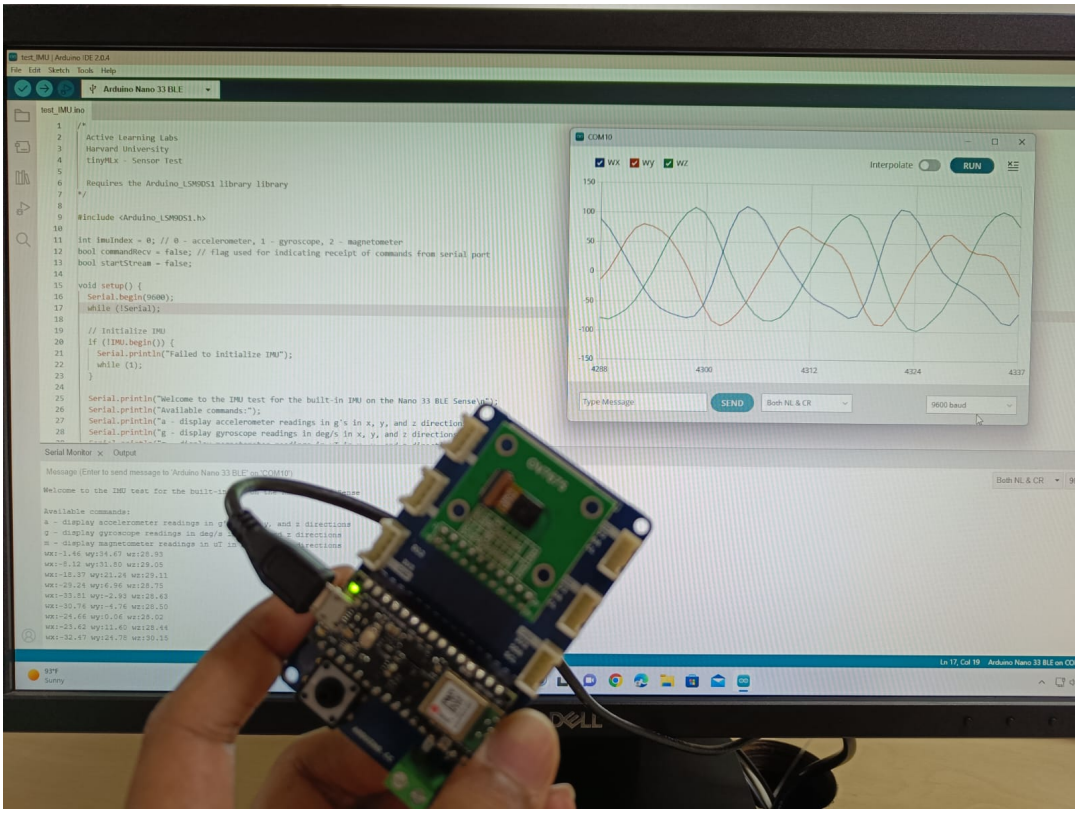

OV7675
NANO 33 BLE SENSE LITE
ARDUINO.CC


Blink.ino
https://www.arduino.cc/en/Main/Products
modified 8 May 2014
by Scott Fitzgerald
modified 2 Sep 2016
by Arturo Guadalupi
modified 8 Sep 2016
by Colby Newman
This example code is in the public domain.
https://www.arduino.cc/en/Tutorial/BuiltInExamples/Blink
// the setup function runs once when you press reset or power the board
void setup() {
// initialize digital pin LED_BUILTIN as an output.
pinMode(LED_BUILTIN, OUTPUT);
}
// the loop function runs over and over again forever
void loop() {
digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
delay(1000); // wait for a second
digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
delay(1000); // wait for a second
}
Output
[==========] 80% (17/21 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x11000, size=0x1000)
[==========] 85% (18/21 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x12000, size=0x1000)
[==========] 90% (19/21 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x13000, size=0x1000)
[==========] 95% (20/21 pages)write(addr=0x34,size=0x1000)
writeBuffer(scr_addr=0x34, dst_addr=0x14000, size=0x1000)
[==========] 100% (21/21 pages)
Done in 3.582 seconds
reset()
93°F
Sunny


Arduino Nano 33 BLE
test_microphone.ino
/*
Active Learning Labs
Harvard University
tinyMLx - Built-in Microphone Test
*/
#include <PDM.h>
#include <TinyMLShield.h>
// PDM buffer
short sampleBuffer[256];
volatile int samplesRead;
bool record = false;
bool commandRecv = false;
void setup() {
Serial.begin(9600);
while (!Serial);
// Initialize the TinyML Shield
initializeShield();
PDM.onReceive(onPDMdata);
// Initialize PDM microphone in mono mode with 16 kHz sample rate
if (!PDM.begin(1, 16000)) {
Serial.println("Failed to start PDM");
while (1);
}
Serial.println("Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense\n");
Serial.println("Use the on-shield button or send the command 'click' to start and stop an audio recording");
Serial.println("Open the Serial Plotter to view the corresponding waveform");
}
Output
Serial Monitor
Message (Enter to send message to 'Arduino Nano 33 BLE' on 'COM10')
Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense
Use the on-shield button or send the command 'click' to start and stop an audio recording
Open the Serial Plotter to view the corresponding waveform


COM10
value 1
Interpolate
STOP
SEND
Type Message
Both NL & CR
9600 baud
Arduino Nano 33 BLE
test_microphone.ino
Output
Serial Monitor
Welcome to the microphone test for the built-in microphone on the Nano 33 BLE Sense
Use the on-shield button or send the command 'click' to start and stop an audio recording
Open the Serial Plotter to view the corresponding waveform


test_microphone | Arduino IDE 2.0.4
Arduino Nano 33 BLE
test_microphone.ino
COM10
value 1
Interpolate
RUN
SEND
Type Message
Both NL & CR
9600 baud
Serial Monitor
Output


test_IMU | Arduino IDE 2.0.4
Arduino Nano 33 BLE
test_IMU.ino
COM10
wx
wy
wz
Interpolate
RUN
SEND
Type Message
Both NL & CR
9600 baud
Serial Monitor
Output

Name: Shreerang Mhatre
Roll no: 29
Batch: A2

19/04/2023

MIT-WPU
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

Exp No. 8

* Past lab questions:

Q1) Describe different features present in the Tiny ML kit.

-> Different features present in the Tiny ML kit are -

1. Microcontroller -
A microcontroller is a small computer that can be embedded in a device to control its functions. A Tiny ML kit usually includes a microcontroller that is optimized for running machine learning algorithms.

2. Sensors -
Sensors are devices that can detect and measure physical phenomena such as temperature motion, and light. Tiny ML kits often come with sensors that can be used to collect data for training and testing machine learning models

www.mitwpu.edu.in

MIT-WPU
Dr. Vishwanath Karad
MIT WORLD PEACE UNIVERSITY | PUNE

(3) Machine learning algorithms -

Tiny ML kits typically include pre trained machine learning algorithms that can be run on the microcontroller. These algorithms are usually optimized for running on low-power devices with limited resources.

(4) Development tools -

Tiny ML kits often come with software development tools that make it easier to develop, test, and deploy machine learning models. These tools may include libraries.

(5) Connectivity -

Tiny ML kits may include features that allow them to connect to other devices or the internet. This can be useful for sending data to cloud-based machine learning models for training or for receiving updates to the device's machine learning algorithms

SSP
28/4/23

www.mitwpu.edu.in