IOT AND CLOUD COMPUTING LAB

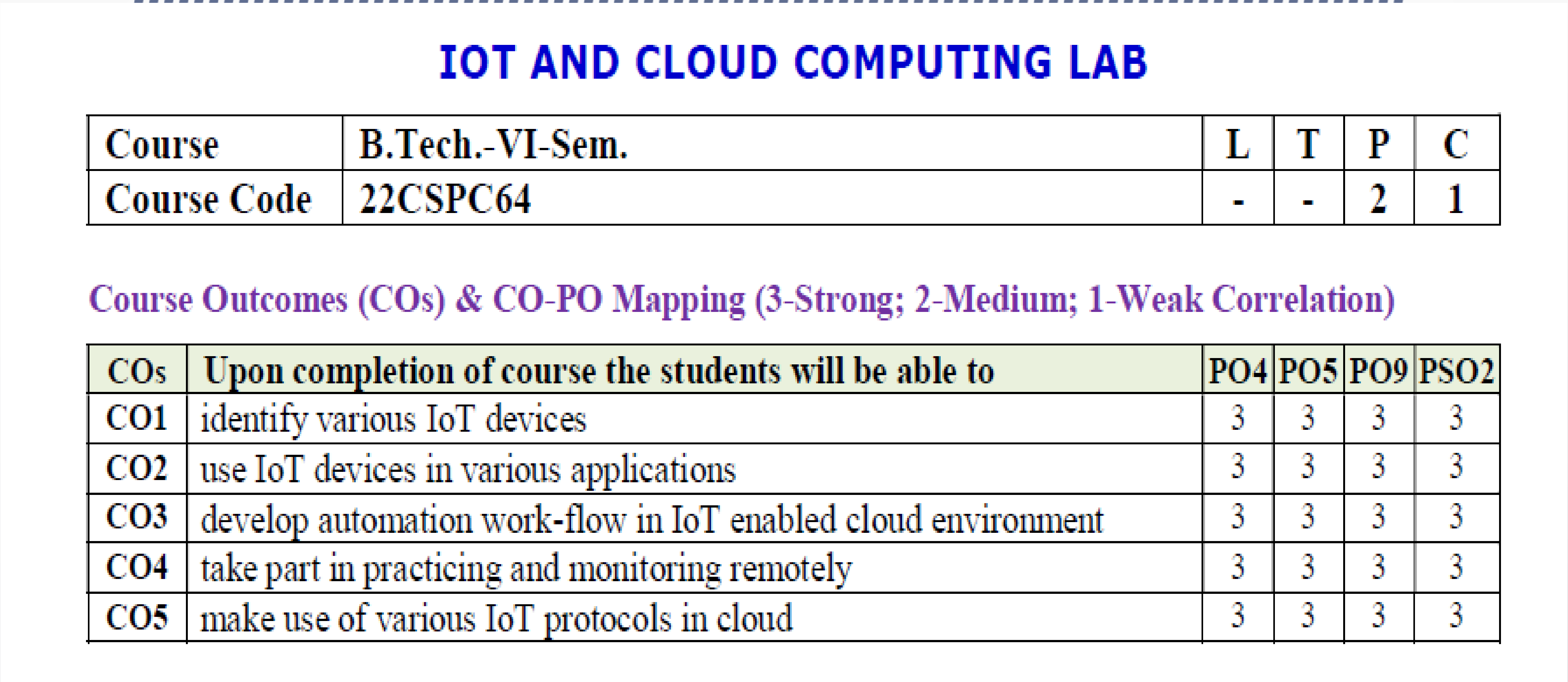
By,

Asst. Prof. S.Asra

M.E(Computer science & Engineering)

B.E(Computer Science & Engineering)

Diploma(Computer Science & Engineering)

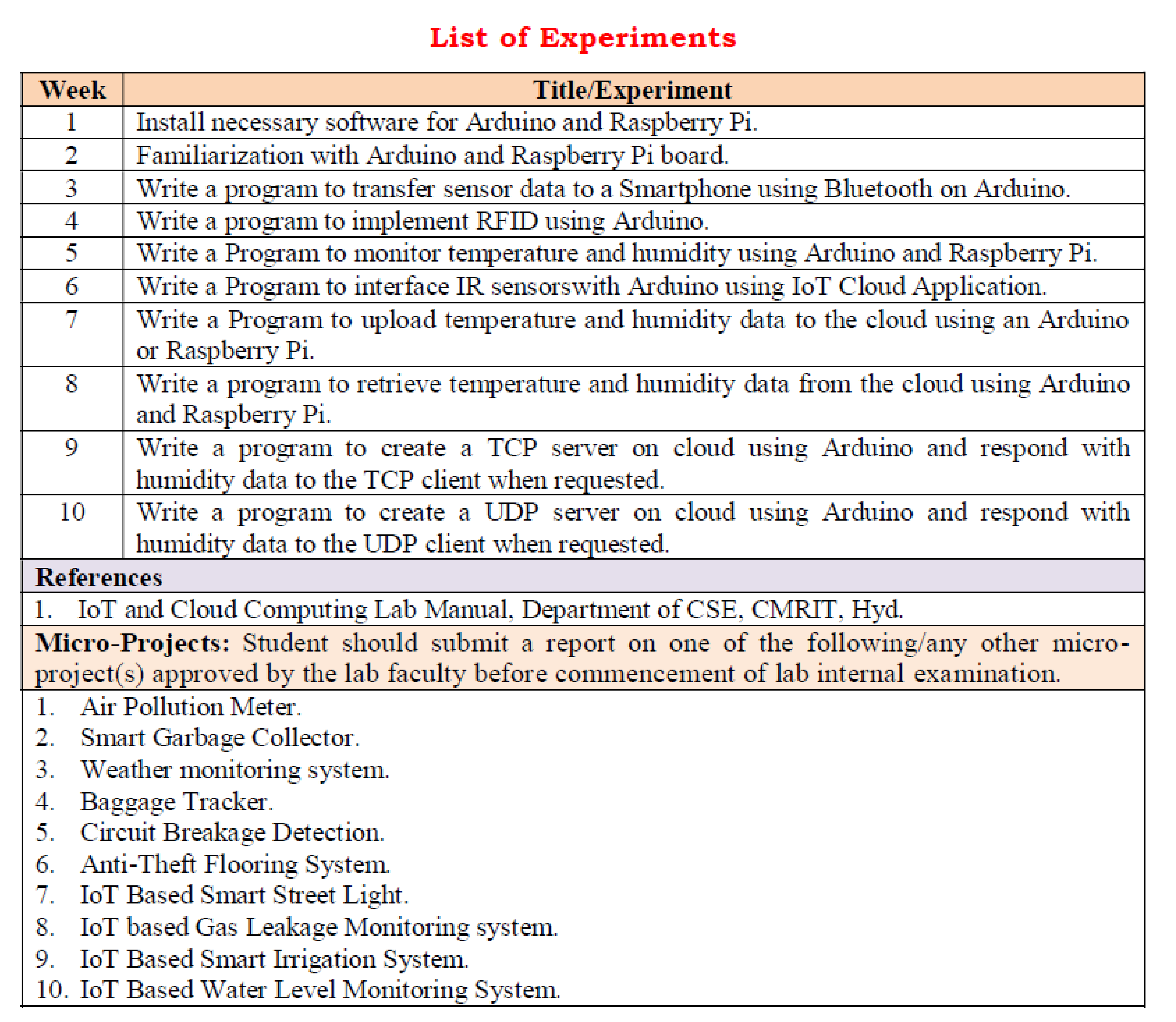


IOT AND CLOUD COMPUTING LAB

Course	B.Tech.-VI-Sem.	L	T	P	C
Course Code	22CSPC64	-	-	2	1

Course Outcomes (COs) & CO-PO Mapping (3-Strong; 2-Medium; 1-Weak Correlation)

COs	Upon completion of course the students will be able to	PO4	PO5	PO9	PSO2
CO1	identify various IoT devices	3	3	3	3
CO2	use IoT devices in various applications	3	3	3	3
CO3	develop automation work-flow in IoT enabled cloud environment	3	3	3	3
CO4	take part in practicing and monitoring remotely	3	3	3	3
CO5	make use of various IoT protocols in cloud	3	3	3	3



List of Experiments

Week	Title/Experiment
1	Install necessary software for Arduino and Raspberry Pi.
2	Familiarization with Arduino and Raspberry Pi board.
3	Write a program to transfer sensor data to a Smartphone using Bluetooth on Arduino.
4	Write a program to implement RFID using Arduino.
5	Write a Program to monitor temperature and humidity using Arduino and Raspberry Pi.
6	Write a Program to interface IR sensorswith Arduino using IoT Cloud Application.
7	Write a Program to upload temperature and humidity data to the cloud using an Arduino or Raspberry Pi.
8	Write a program to retrieve temperature and humidity data from the cloud using Arduino and Raspberry Pi.
9	Write a program to create a TCP server on cloud using Arduino and respond with humidity data to the TCP client when requested.
10	Write a program to create a UDP server on cloud using Arduino and respond with humidity data to the UDP client when requested.

References

1. IoT and Cloud Computing Lab Manual, Department of CSE, CMRIT, Hyd.

Micro-Projects: Student should submit a report on one of the following/any other micro-project(s) approved by the lab faculty before commencement of lab internal examination.

1. Air Pollution Meter.
2. Smart Garbage Collector.
3. Weather monitoring system.
4. Baggage Tracker.
5. Circuit Breakage Detection.
6. Anti-Theft Flooring System.
7. IoT Based Smart Street Light.
8. IoT based Gas Leakage Monitoring system.
9. IoT Based Smart Irrigation System.
10. IoT Based Water Level Monitoring System.

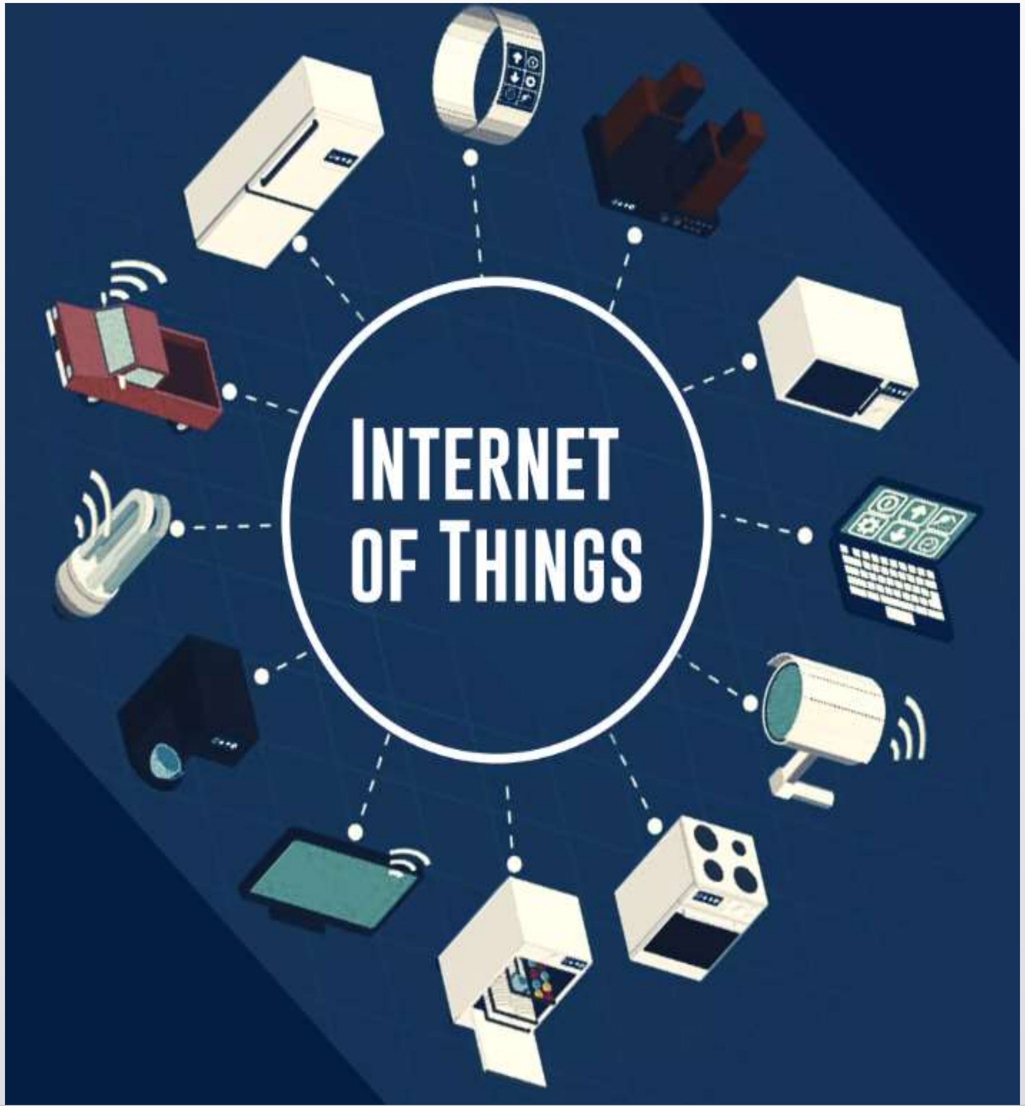
1. Introduction to Internet of Things

- **Definition**:

  The Internet of Things (IoT) refers to **a network of physical objects—devices, vehicles, appliances, and other items embedded with sensors, software, and connectivity to exchange data over the internet**.

- **Some common examples of things or devices**

  1. mobile phones
  2. Home automation digital voice Assistants
  3. computer/Laptops
  4. Arduino/Raspberry pi
  5. Robot



1. Introduction to Internet of Things- HISTORY

Historical Background:
The concept of IoT originated in the late 1990s
when Kevin Ashton coined the term to describe a system where the internet connects to the physical world via sensors.

*It gained momentum with advances in
->wireless communication,
->cloud computing, and
->affordable hardware.

1. Introduction to Internet of Things- HISTORY

History of IOT:

1. **1970-** The actual **idea of connected** devices was proposed.
2. **1980-**first connected device was invented was a **coco-cola vending machine** operated by programmers.
3. **1990-**john romkey -introduces **a toaster connected to the internet with TCP/IP protocol.**
4. **1995-**siemens introduced **first cellular module** build by M2M
5. **1998-**The introduction of **IPV6**
6. **1999-kevin Asthon introduces the term IOT during his work kevin linked the idea of RFID based item identification system.**
7. **2000- LG enables world first internet enabled refrigerators**.
8. **2004- BOOK on iot** was published and title was famous.
9. **2009-** Google starts testing **self driving cars**.
10. **2013-** Google introduced **google glass.**
11. **2016-first IOT malware named 'mirai' was found**.

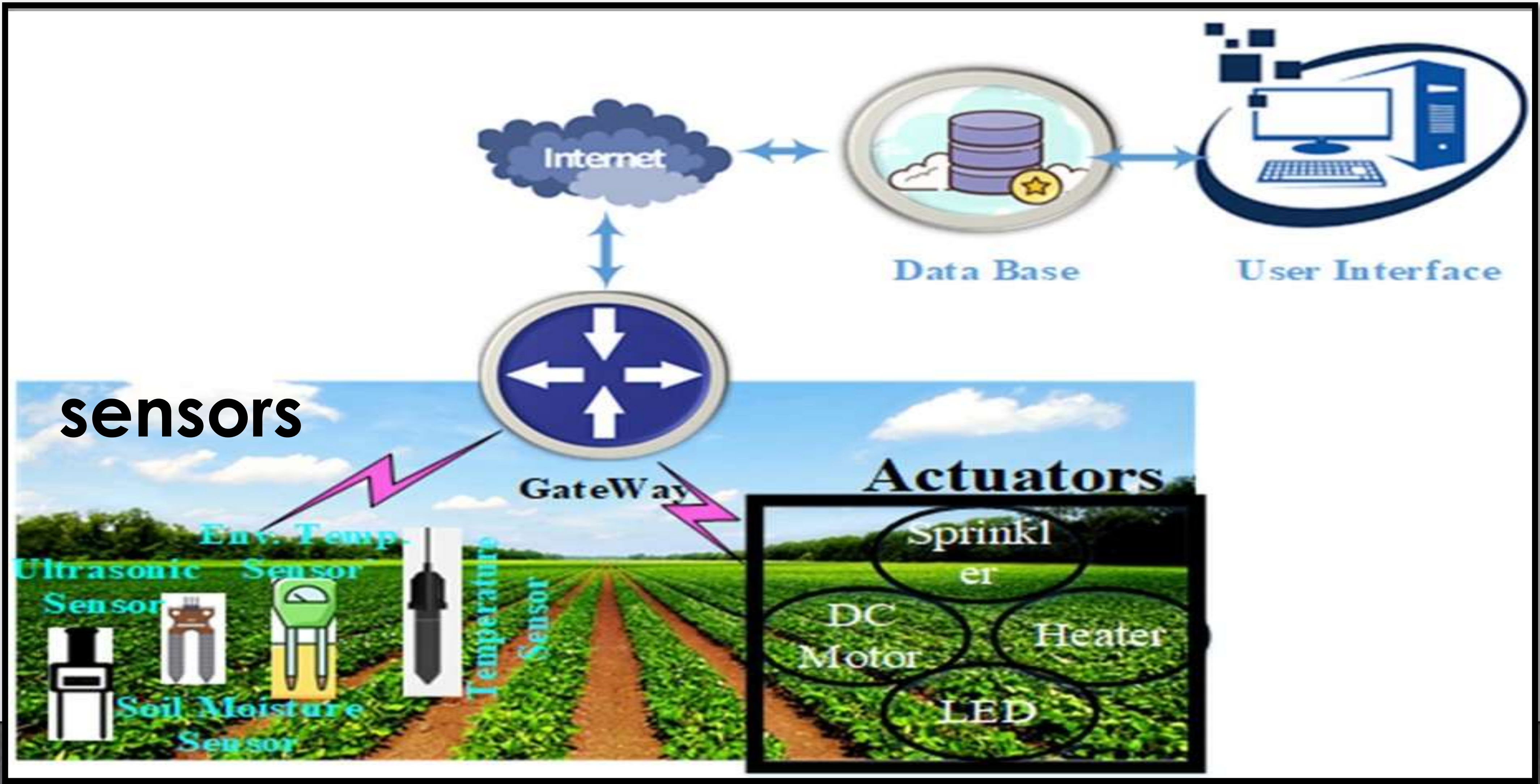
1. Introduction to Internet of Things- HISTORY

Kevin Ashton

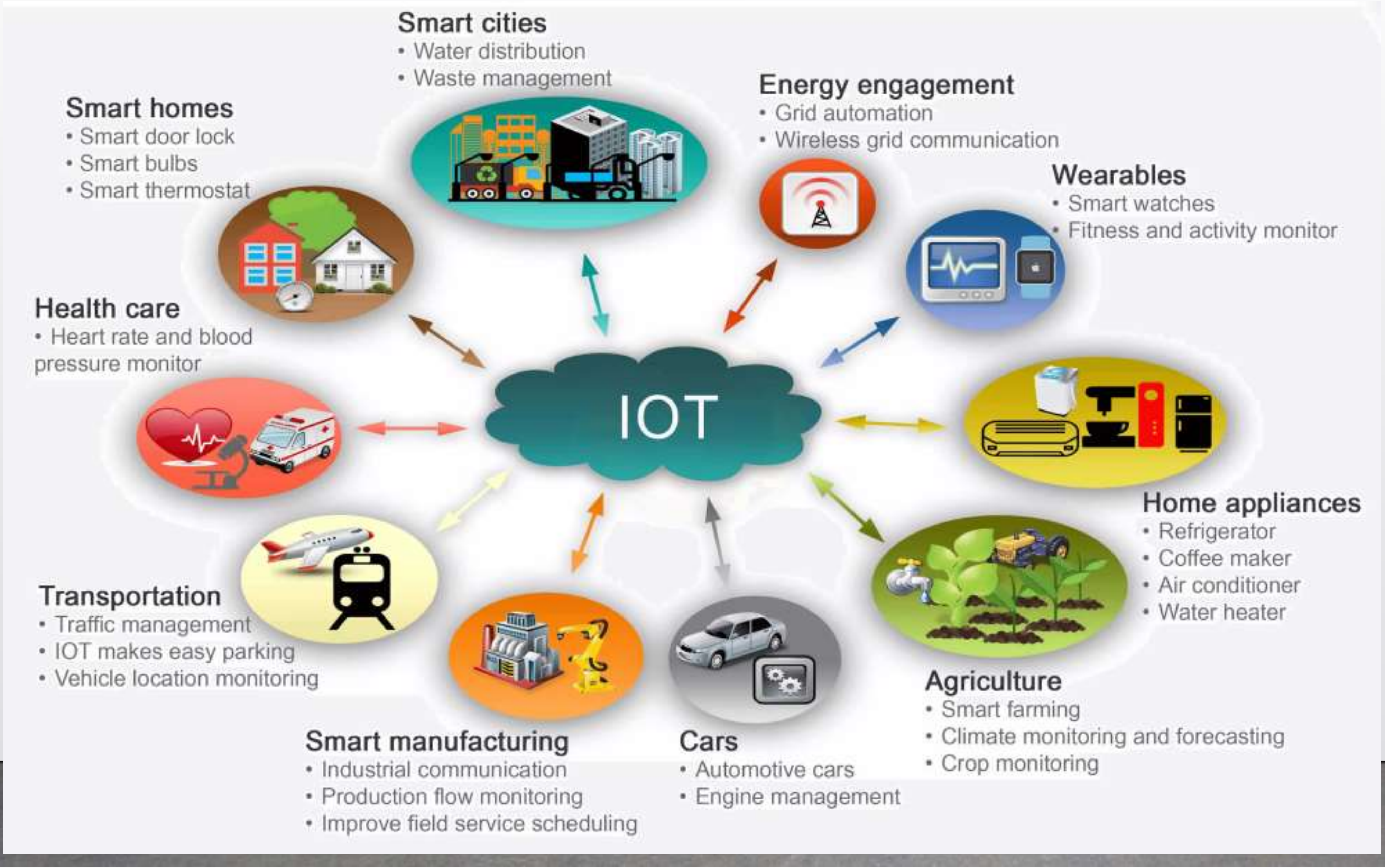
Inventor of the "Internet of Things" | Creator of the WeMo home automation platform | Co-Founder & CEO of leading clean-tech startup Zensi

Kevin Ashton, co-founder of the Auto-ID Center at the Massachusetts Institute of Technology (MIT), first mentioned the "Internet Of Things" in a presentation he made to Procter &Gamble (P&G) in 1999.

IOT based on Smart Agriculture System



Applications of IOT Week 1



Smart cities
- Water distribution
- Waste management

Energy engagement
- Grid automation
- Wireless grid communication

Wearables
- Smart watches
- Fitness and activity monitor

Home appliances
- Refrigerator
- Coffee maker
- Air conditioner
- Water heater

Agriculture
- Smart farming
- Climate monitoring and forecasting
- Crop monitoring

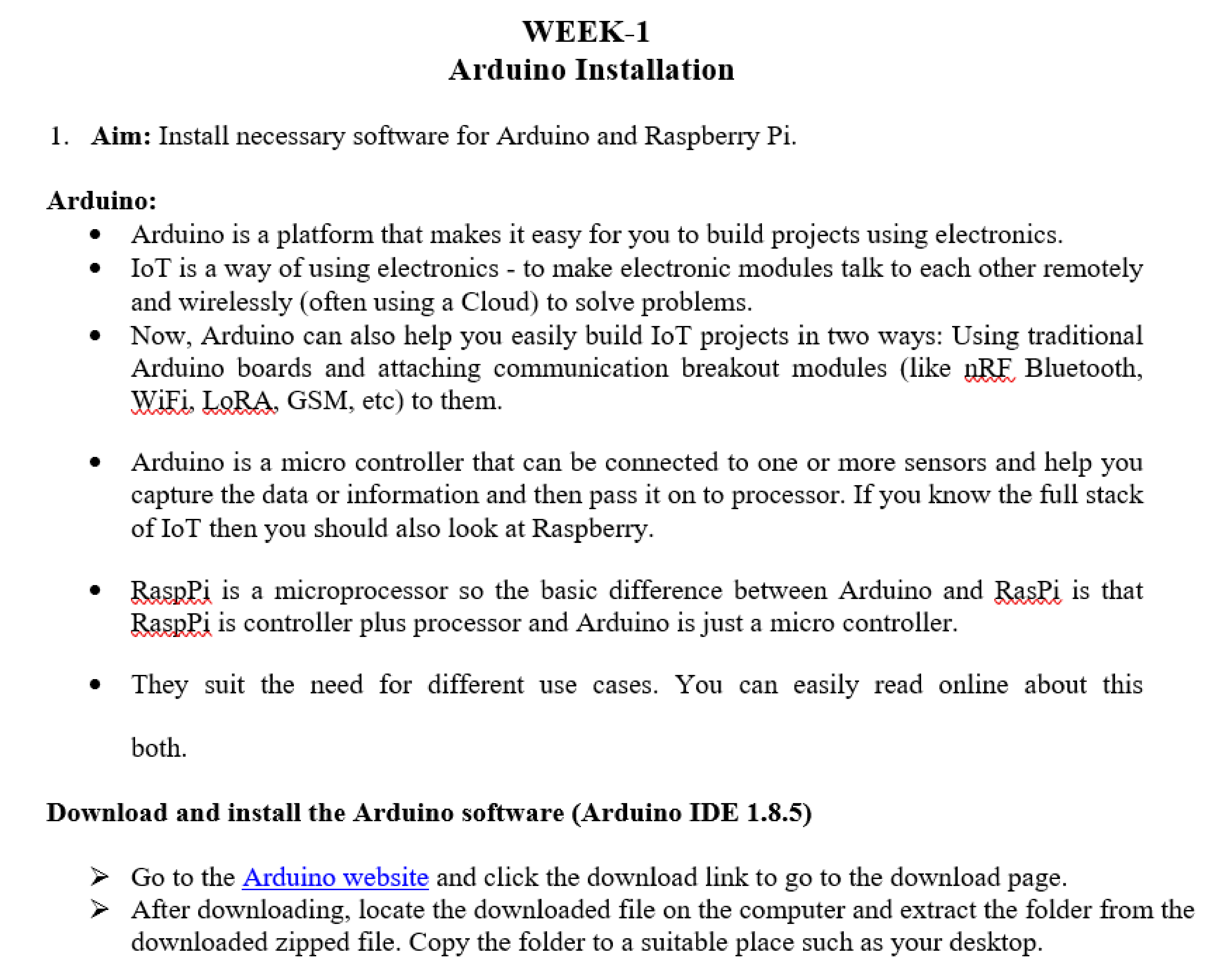
Cars
- Automotive cars
- Engine management

Smart manufacturing
- Industrial communication
- Production flow monitoring
- Improve field service scheduling

Transportation
- Traffic management
- IOT makes easy parking
- Vehicle location monitoring

Health care
- Heart rate and blood pressure monitor

Smart homes
- Smart door lock
- Smart bulbs
- Smart thermostat



WEEK-1
Arduino Installation

1. **Aim:** Install necessary software for Arduino and Raspberry Pi.

Arduino:

- Arduino is a platform that makes it easy for you to build projects using electronics.
- IoT is a way of using electronics - to make electronic modules talk to each other remotely and wirelessly (often using a Cloud) to solve problems.
- Now, Arduino can also help you easily build IoT projects in two ways: Using traditional Arduino boards and attaching communication breakout modules (like nRF Bluetooth, WiFi, LoRA, GSM, etc) to them.
- Arduino is a micro controller that can be connected to one or more sensors and help you capture the data or information and then pass it on to processor. If you know the full stack of IoT then you should also look at Raspberry.
- RaspPi is a microprocessor so the basic difference between Arduino and RasPi is that RaspPi is controller plus processor and Arduino is just a micro controller.
- They suit the need for different use cases. You can easily read online about this both.

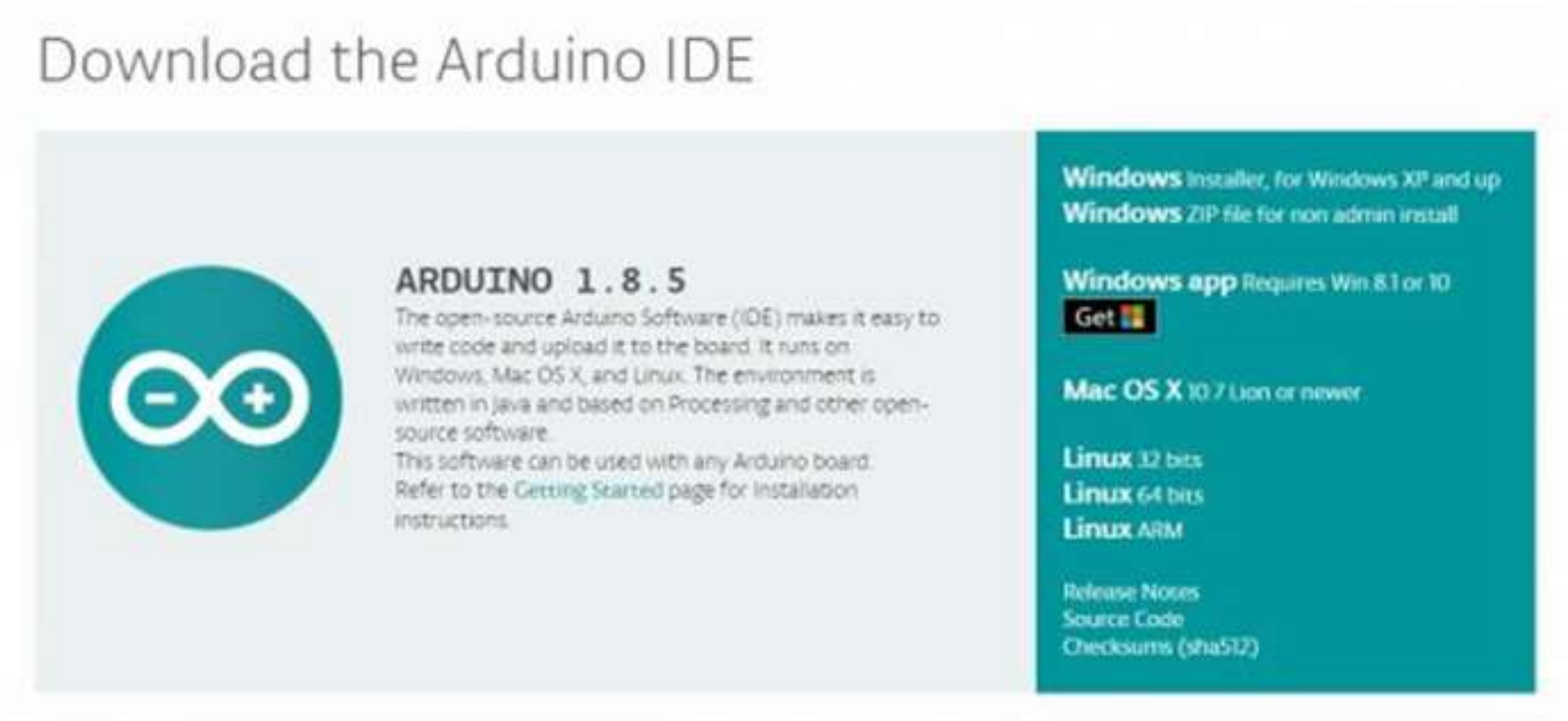
Download and install the Arduino software (Arduino IDE 1.8.5)

- ➢ Go to the Arduino website and click the download link to go to the download page.
- ➢ After downloading, locate the downloaded file on the computer and extract the folder from the downloaded zipped file. Copy the folder to a suitable place such as your desktop.

Week 1

Download and install the Arduino software (Arduino IDE 1.8.5)

- Go to the Arduino website and click the download link to go to the download page.
- After downloading, locate the downloaded file on the computer and extract the folder from the downloaded zipped file. Copy the folder to a suitable place such as your desktop.



Download the Arduino IDE

ARDUINO 1.8.5

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.
This software can be used with any Arduino board.
Refer to the Getting Started page for installation instructions.

Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

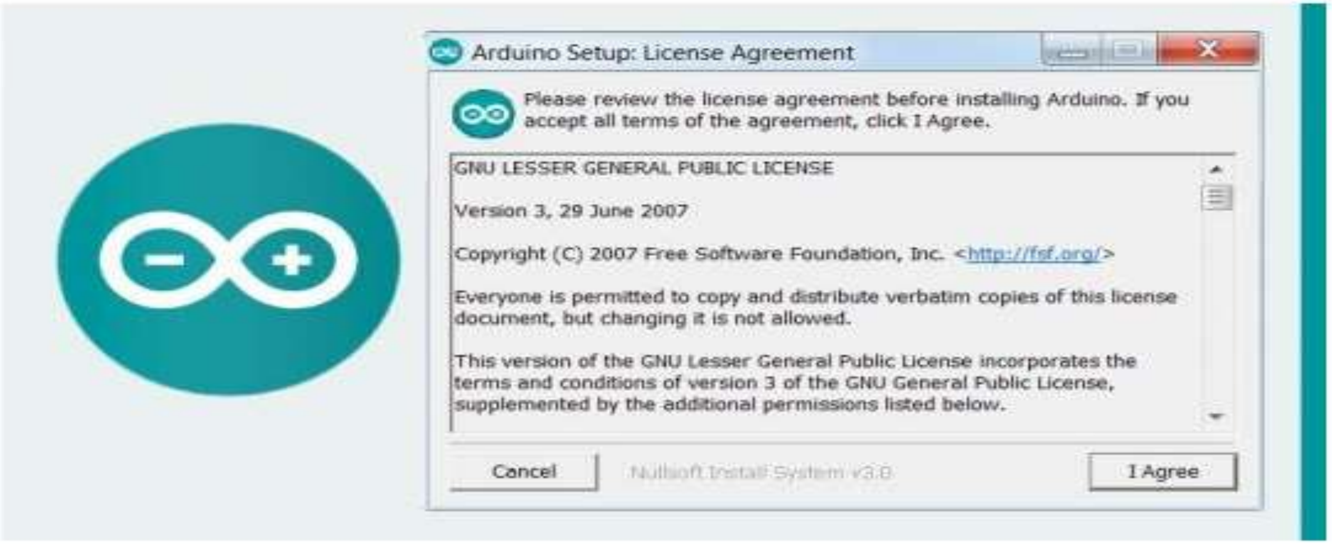
Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

Week 1



Arduino Setup: License Agreement

Please review the license agreement before installing Arduino. If you accept all terms of the agreement, click I Agree.

GNU LESSER GENERAL PUBLIC LICENSE

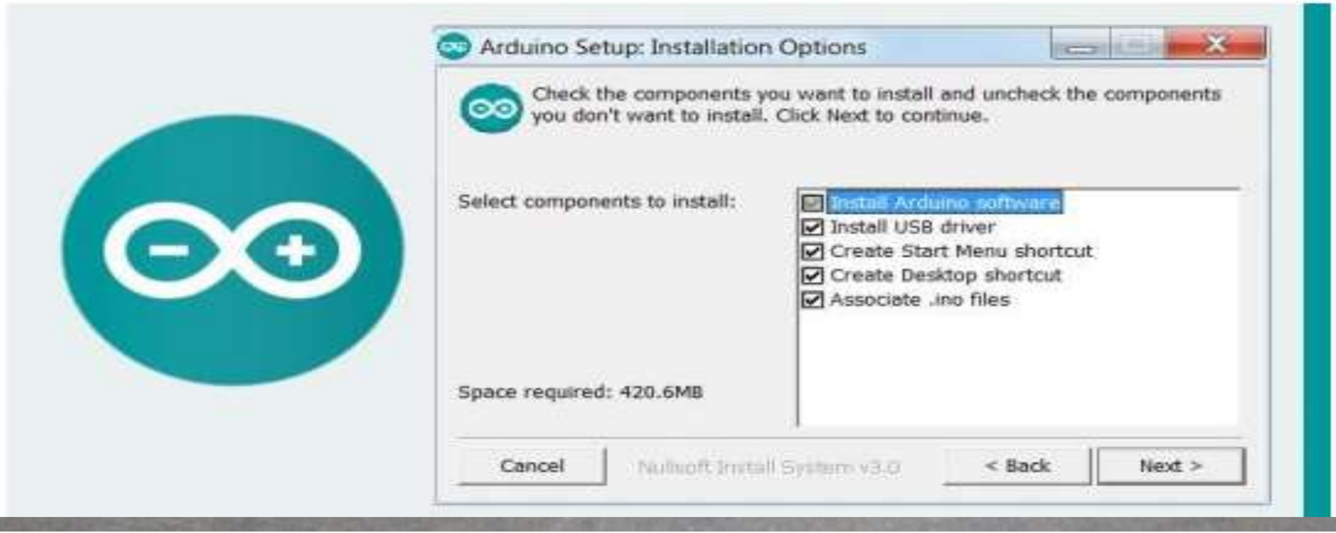
Version 3, 29 June 2007

Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

This version of the GNU Lesser General Public License incorporates the terms and conditions of version 3 of the GNU General Public License, supplemented by the additional permissions listed below.

Read the Arduino License agreement and click the "I Agree" button.



Arduino Setup: Installation Options

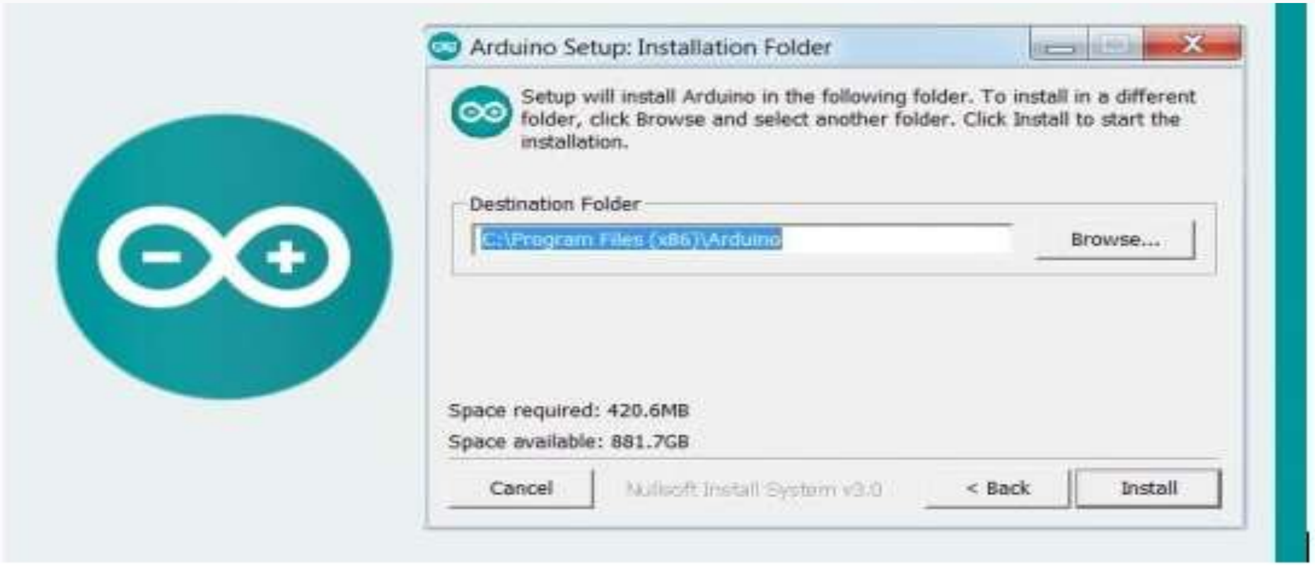
Check the components you want to install and uncheck the components you don't want to install. Click Next to continue.

Select components to install:

- ☑ Install Arduino software
- ☑ Install USB driver
- ☑ Create Start Menu shortcut
- ☑ Create Desktop shortcut
- ☑ Associate .ino files

Space required: 420.6MB

Week 1



Arduino Setup: Installation Folder

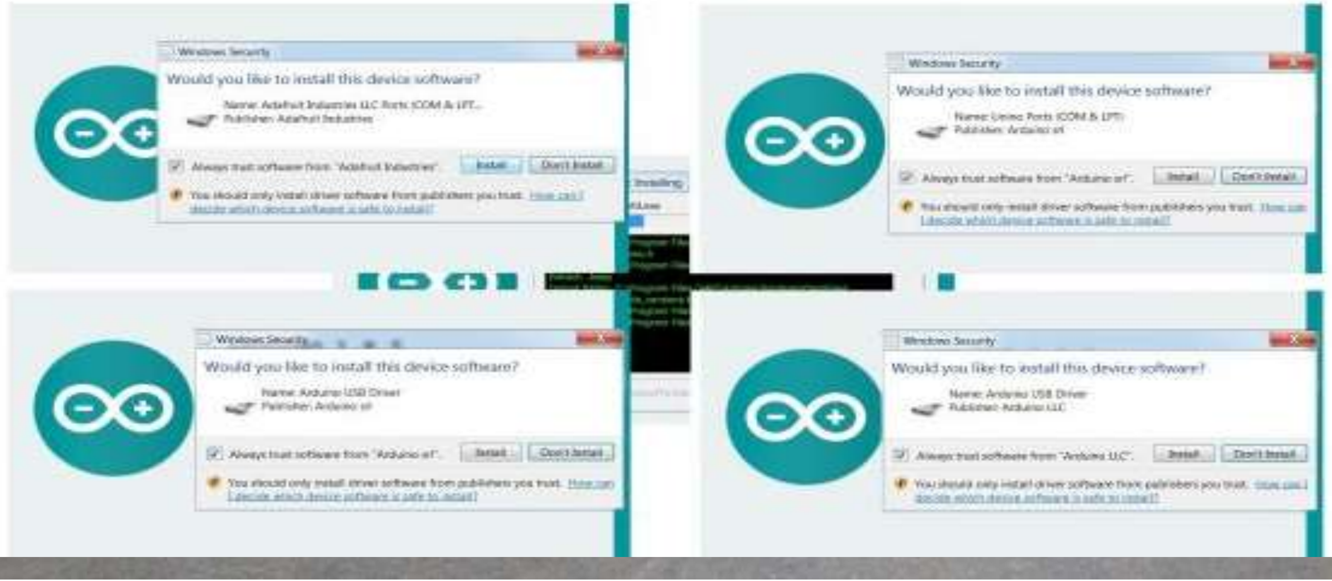
Setup will install Arduino in the following folder. To install in a different folder, click Browse and select another folder. Click Install to start the installation.

Destination Folder: C:\Program Files (x86)\Arduino

Space required: 420.6MB

Space available: 881.7GB

The Arduino software will start to install.



Windows Security

Would you like to install this device software?

Name: Adafruit Industries LLC Ports (COM & LPT...

Publisher: Adafruit Industries

☑ Always trust software from "Adafruit Industries".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Windows Security

Would you like to install this device software?

Name: Unino Ports (COM & LPT)

Publisher: Arduino srl

☑ Always trust software from "Arduino srl".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Windows Security

Would you like to install this device software?

Name: Arduino USB Driver

Publisher: Arduino srl

☑ Always trust software from "Arduino srl".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Windows Security

Would you like to install this device software?

Name: Arduino USB Driver

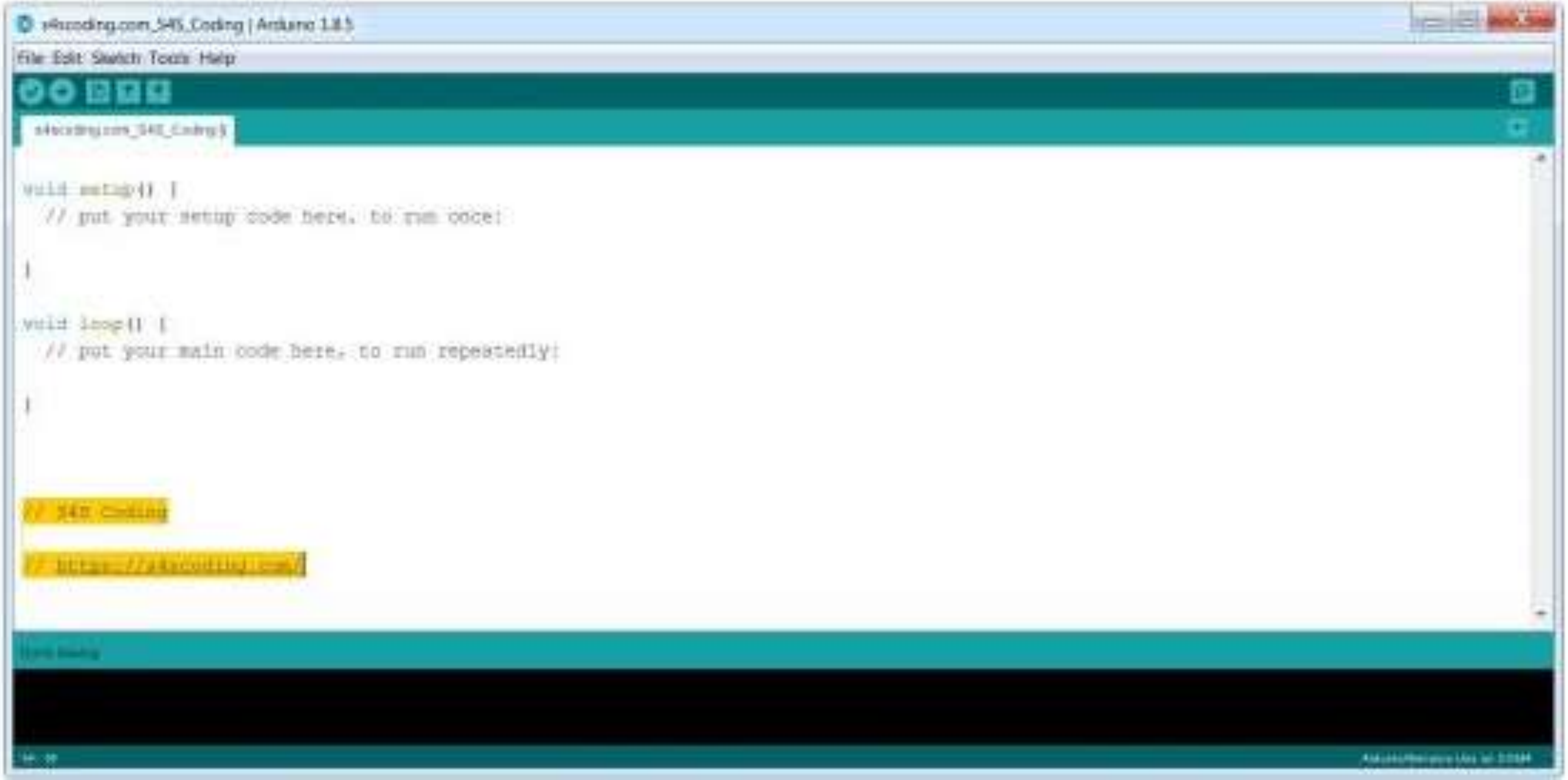
Publisher: Arduino LLC

☑ Always trust software from "Arduino LLC".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Week 1

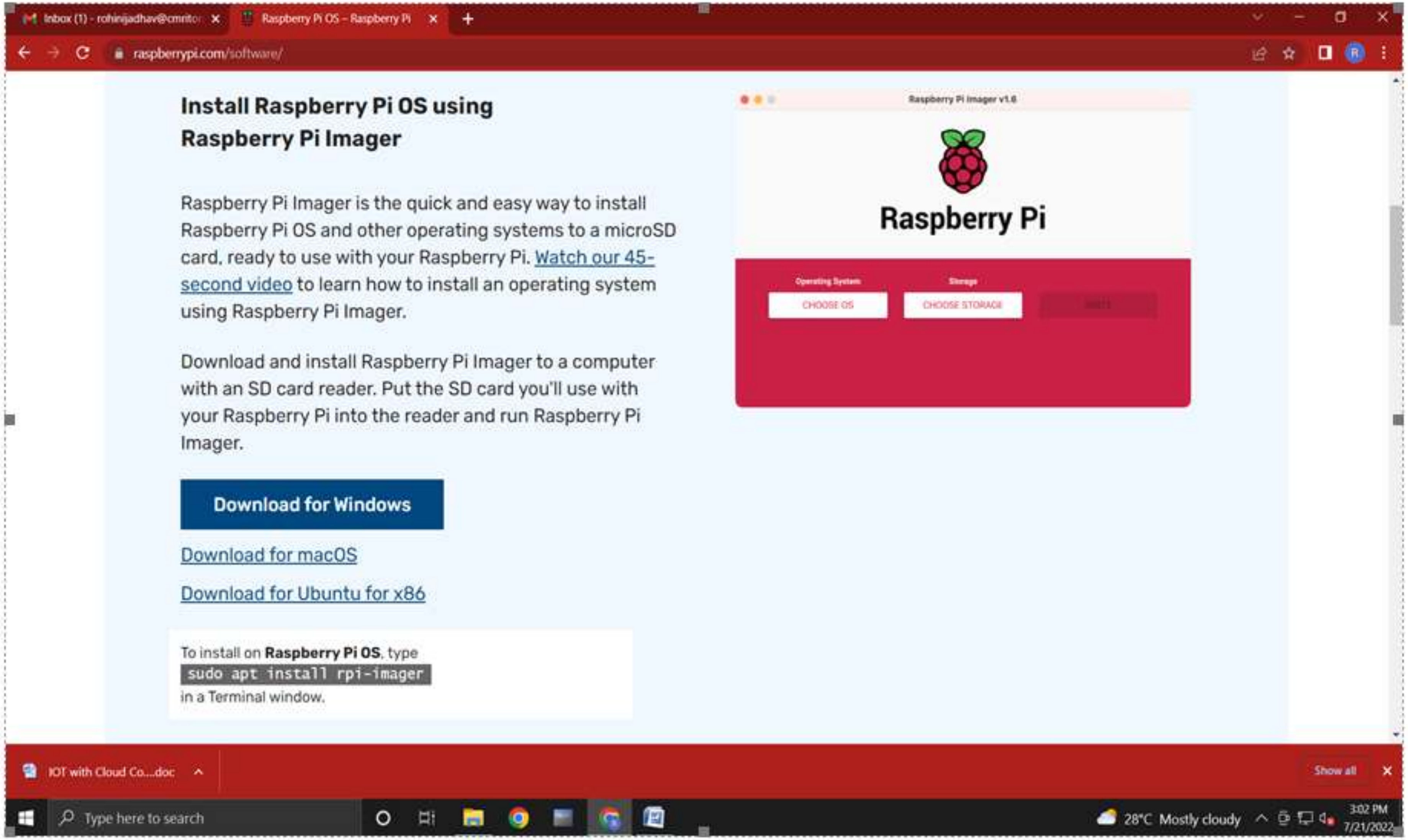
Running the Arduino IDE Software


s4scoding.com_S4S_Coding | Arduino 1.8.5
File Edit Sketch Tools Help
s4scoding.com_S4S_Coding §

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}



// S4S Coding

// https://s4scoding.com/
```

Week 1

Raspberry Pi



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

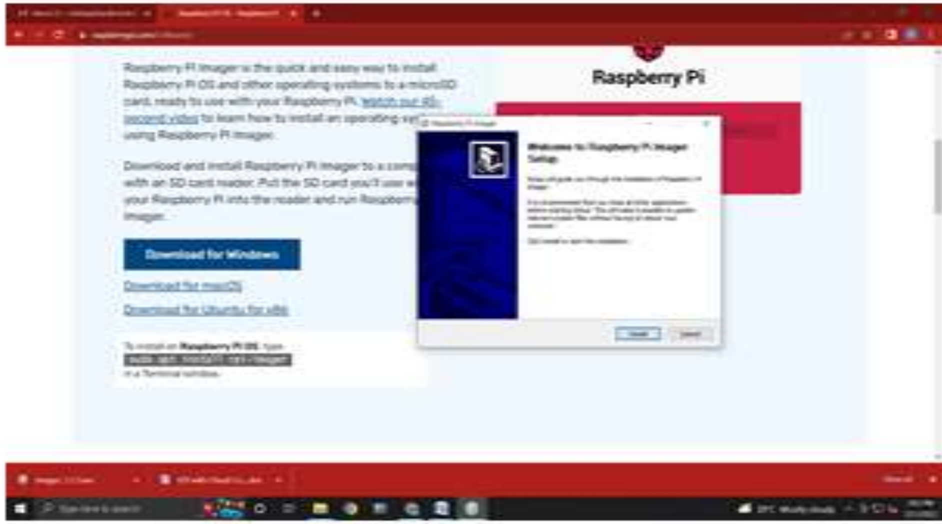
Download for Windows

Download for macOS

Download for Ubuntu for x86

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

Week 1



Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. Watch our 45-second video to learn how to install an operating system using Raspberry Pi Imager.

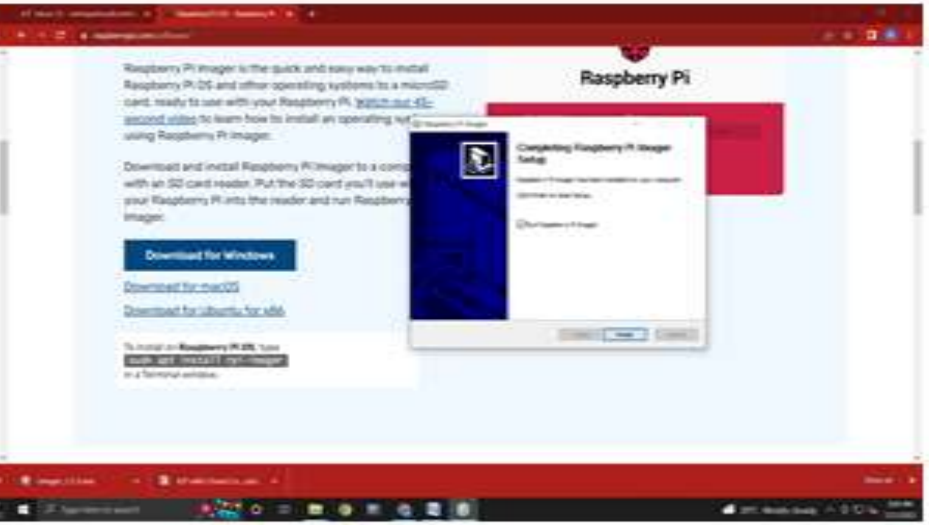
Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

To install on Raspberry Pi OS, type

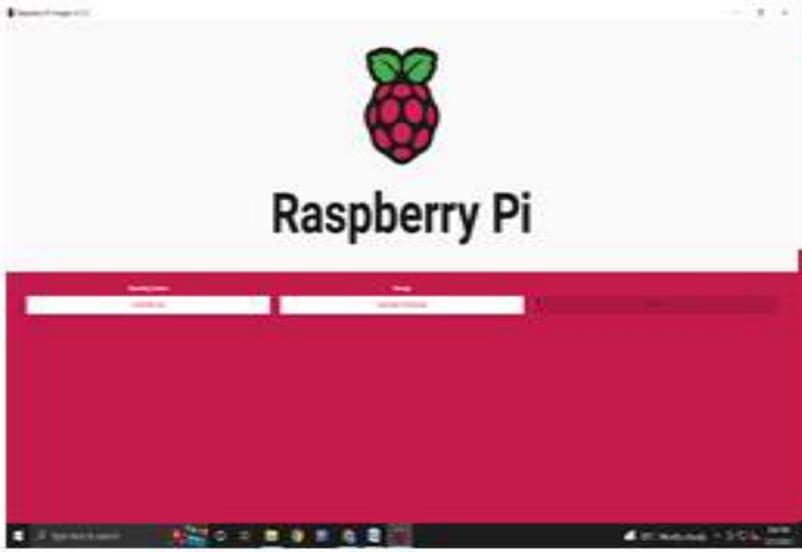
```
sudo apt install rpi-imager
```

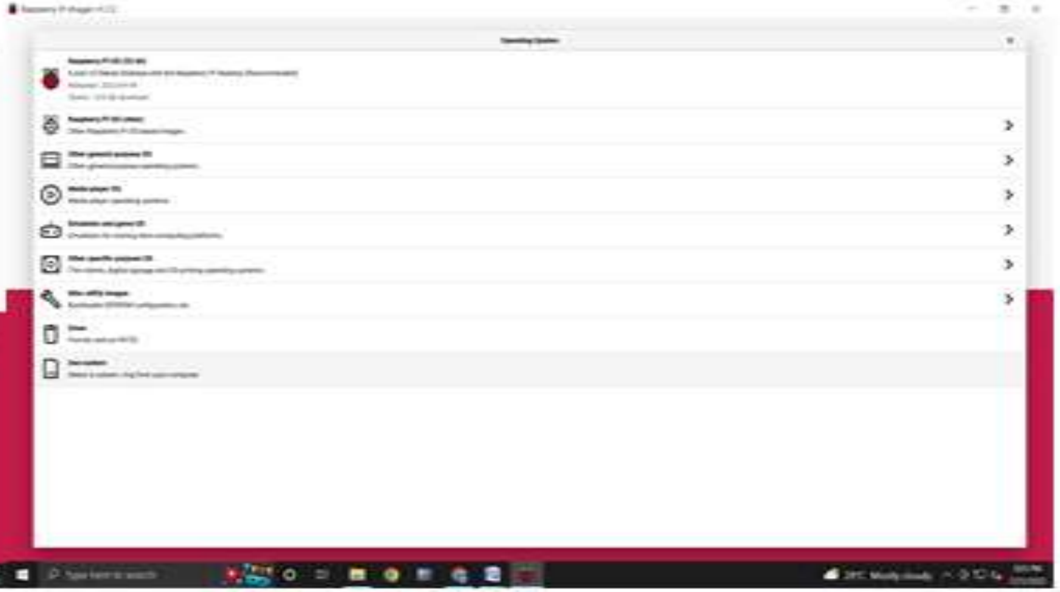
in a Terminal window.

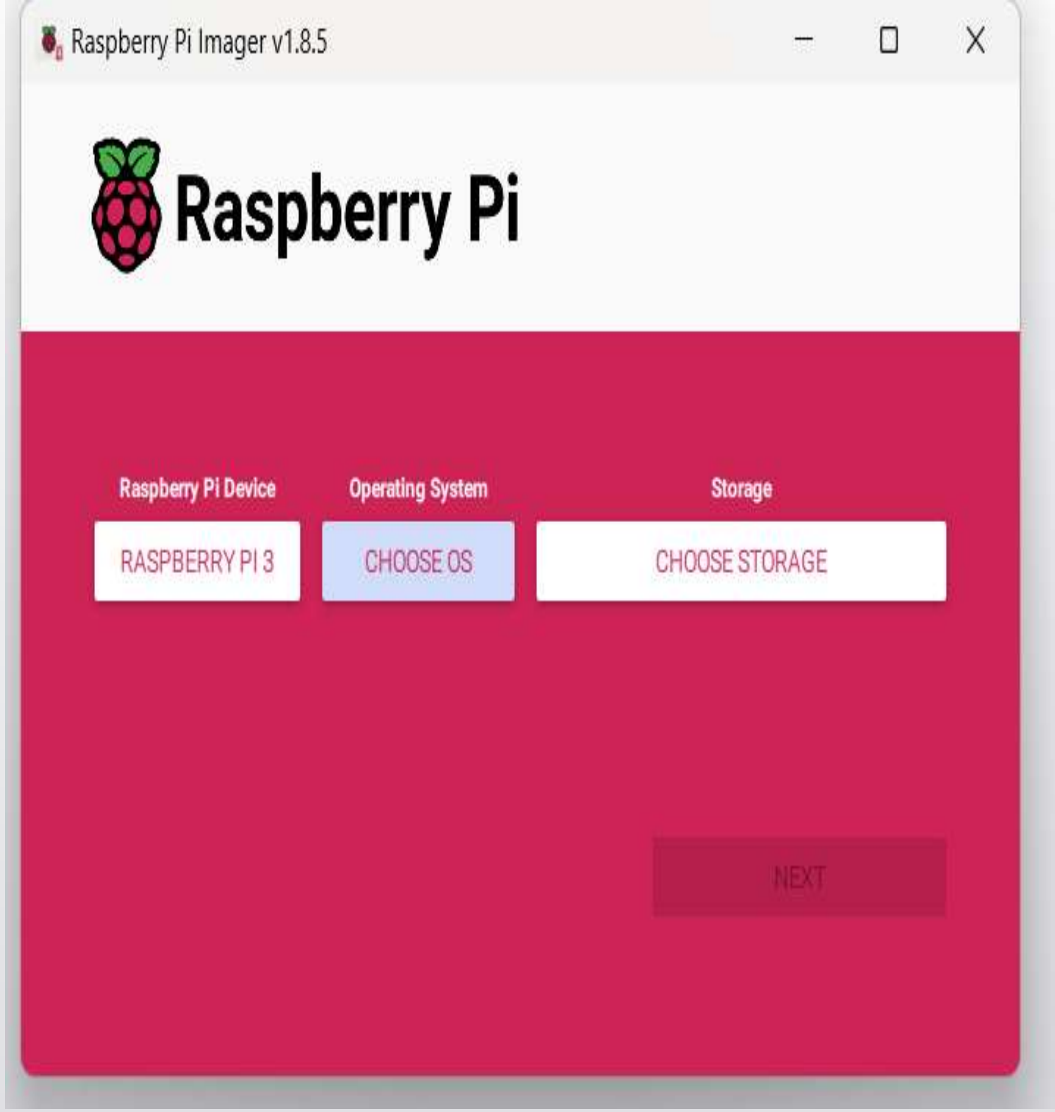
Raspberry Pi
Download for Windows
Download for macOS
Download for Ubuntu for x86
Welcome to Raspberry Pi Imager Setup



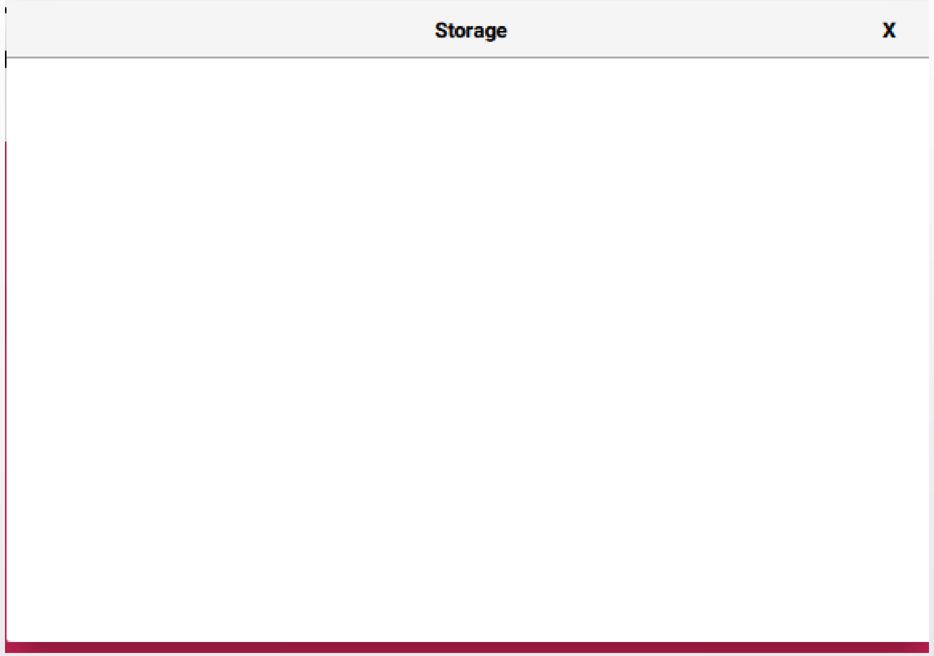
Raspberry Pi
Download for Windows
Download for macOS
Download for Ubuntu for x86
Completing Raspberry Pi Imager Setup





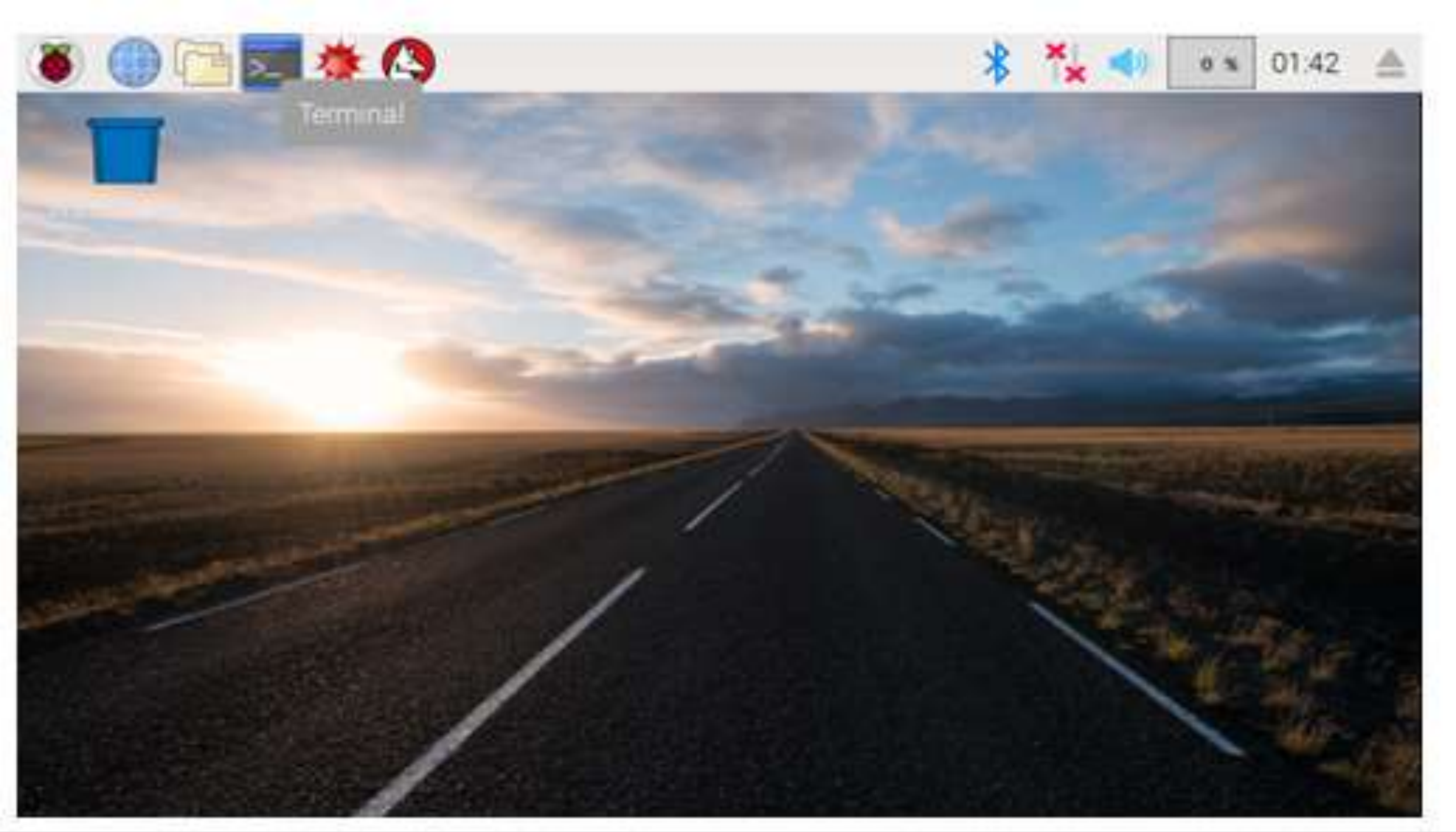


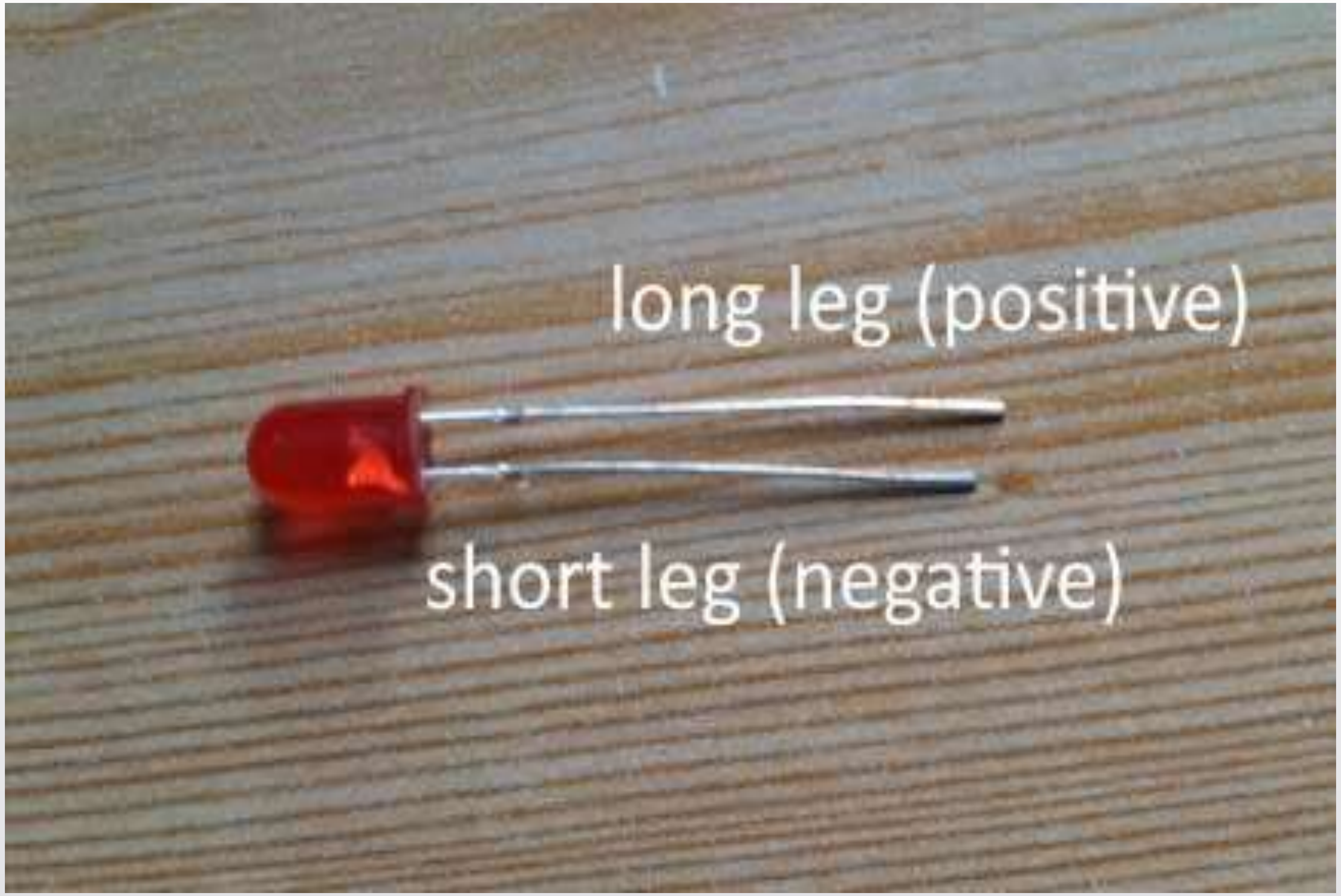
Raspberry Pi Imager v1.8.5
Raspberry Pi
Raspberry Pi Device
RASPBERRY PI 3
Operating System
CHOOSE OS
Storage
CHOOSE STORAGE
NEXT



Storage
X

Week 1



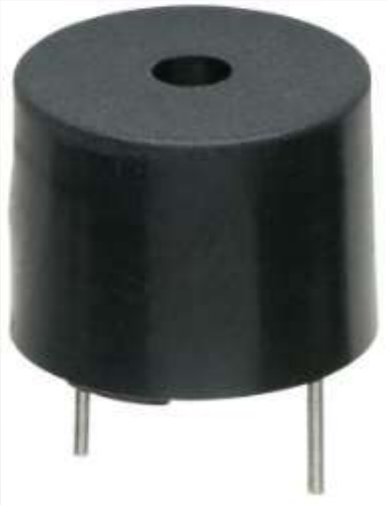


LED Light(Blink Code)

Week 1

Blink light

```
void setup()
 {

  pinMode(LED_BUILTIN, OUTPUT);      // initialize digital pin LED_BUILTIN as an 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
}
```



BUZZER (buzz sound)

Week 1

Buzzer sound

```
void setup()
 {
  // initialize digital pin BUZZER as an output.
  pinMode(BUZZ, OUTPUT);
}

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

```
// Define the pin numbers for the RGB LED
int redPin = 9;
int greenPin = 10;
int bluePin = 11;

void setup() {
  // Set the RGB LED pins as OUTPUT
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  // Red color
  digitalWrite(redPin, HIGH);
  digitalWrite(greenPin, LOW);
  digitalWrite(bluePin, LOW);
  delay(1000); // Wait for 1 second
// Green color
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, HIGH);
  digitalWrite(bluePin, LOW);
  delay(1000); // Wait for 1 second

  // Blue color
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, LOW);
  digitalWrite(bluePin, HIGH);
  delay(1000); // Wait for 1 second

  // All colors off
  digitalWrite(redPin, LOW);
  digitalWrite(greenPin, LOW);
  digitalWrite(bluePin, LOW);
  delay(1000); // Wait for 1 second
}
```

week 2

The **Arduino Uno** is a widely used microcontroller board developed by Arduino. It is designed to make electronics and programming accessible for beginners and professionals alike. Here's an overview of the key features and specifications:

Key Features:

1. **Microcontroller**: ATmega328P (by Atmel/AVR).
2. **Digital I/O Pins**: 14 (of which 6 can be used as PWM outputs).
3. **Analog Input Pins**: 6.
4. **Clock Speed**: 16 MHz.
5. **Flash Memory**: 32 KB (of which 0.5 KB is used by the bootloader).
6. **SRAM**: 2 KB.
7. **EEPROM**: 1 KB.
8. **Operating Voltage**: 5V (input voltage range: 6-20V via barrel jack or 7-12V for optimal operation).
9. **Connectivity**: USB-B port for programming and power, as well as UART, SPI, and I2C communication.
10. **Power Supply Options**: Can be powered via USB or an external power source.

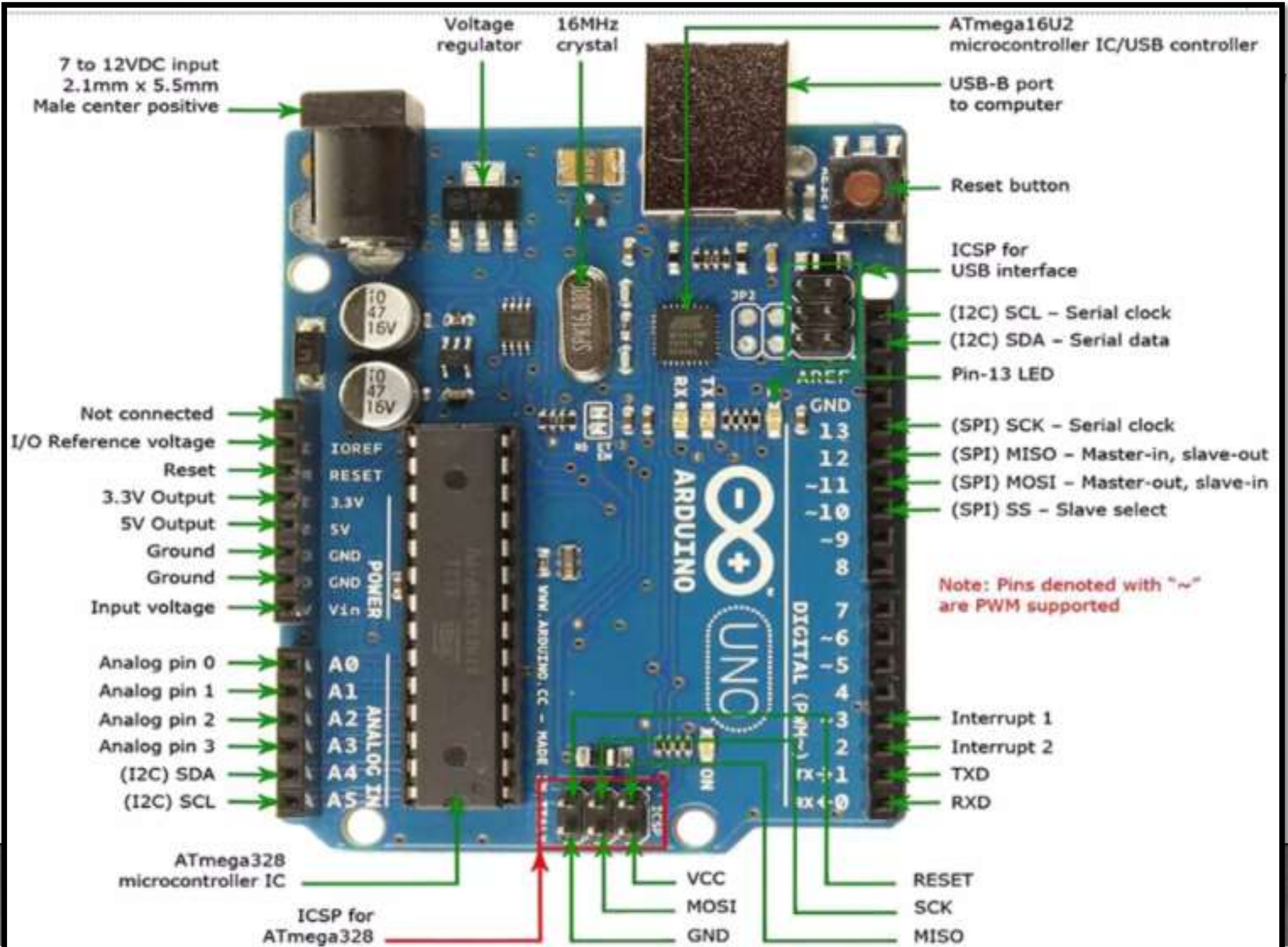
Use Cases:

- Robotics
- IoT projects
- Sensors and data acquisition
- Home automation
- LED and motor control

Advantages:

- Beginner-friendly with extensive community support.
- Compatible with numerous shields (add-on boards) to extend functionality.
- Cross-platform Integrated Development Environment (IDE).
- Open-source hardware and software.

Arduino Pi Board



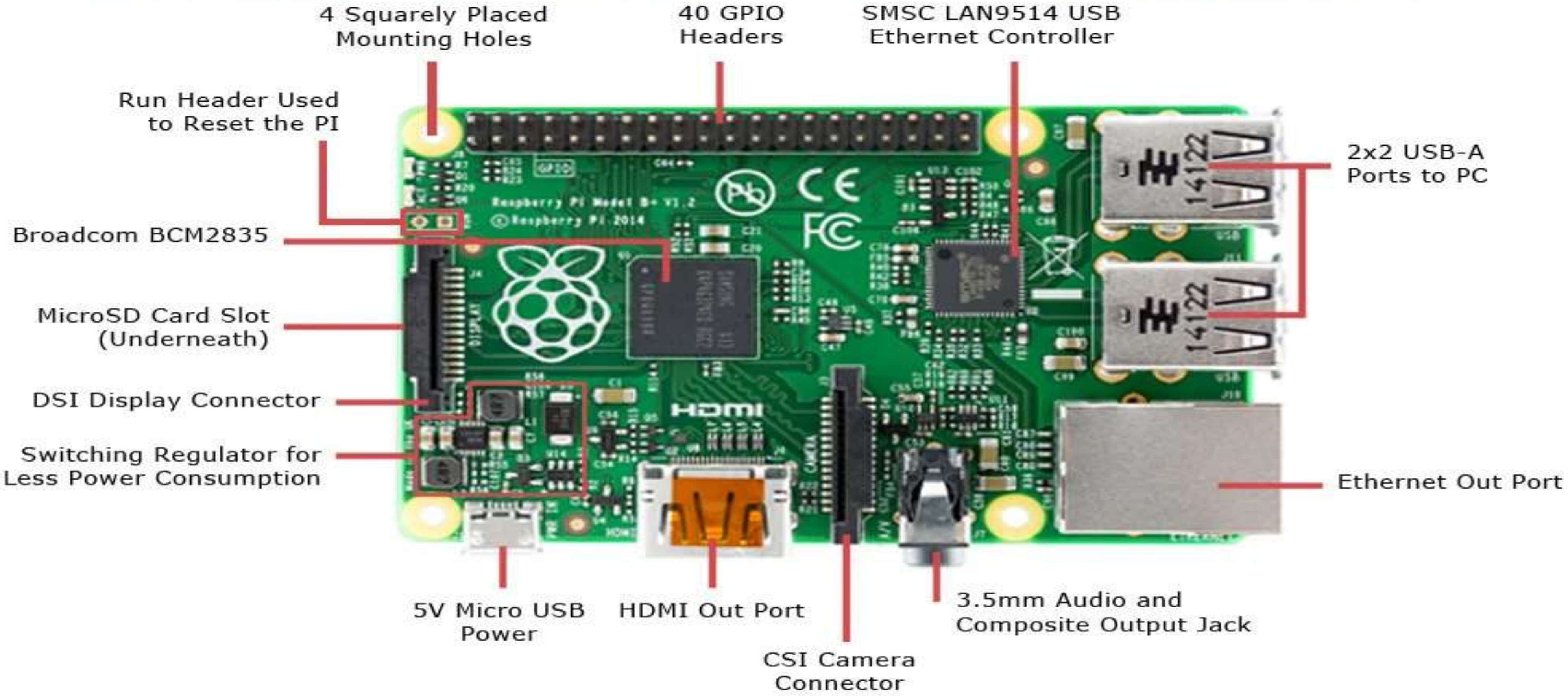
Note: Pins denoted with "~" are PWM supported

Raspberry Pi board

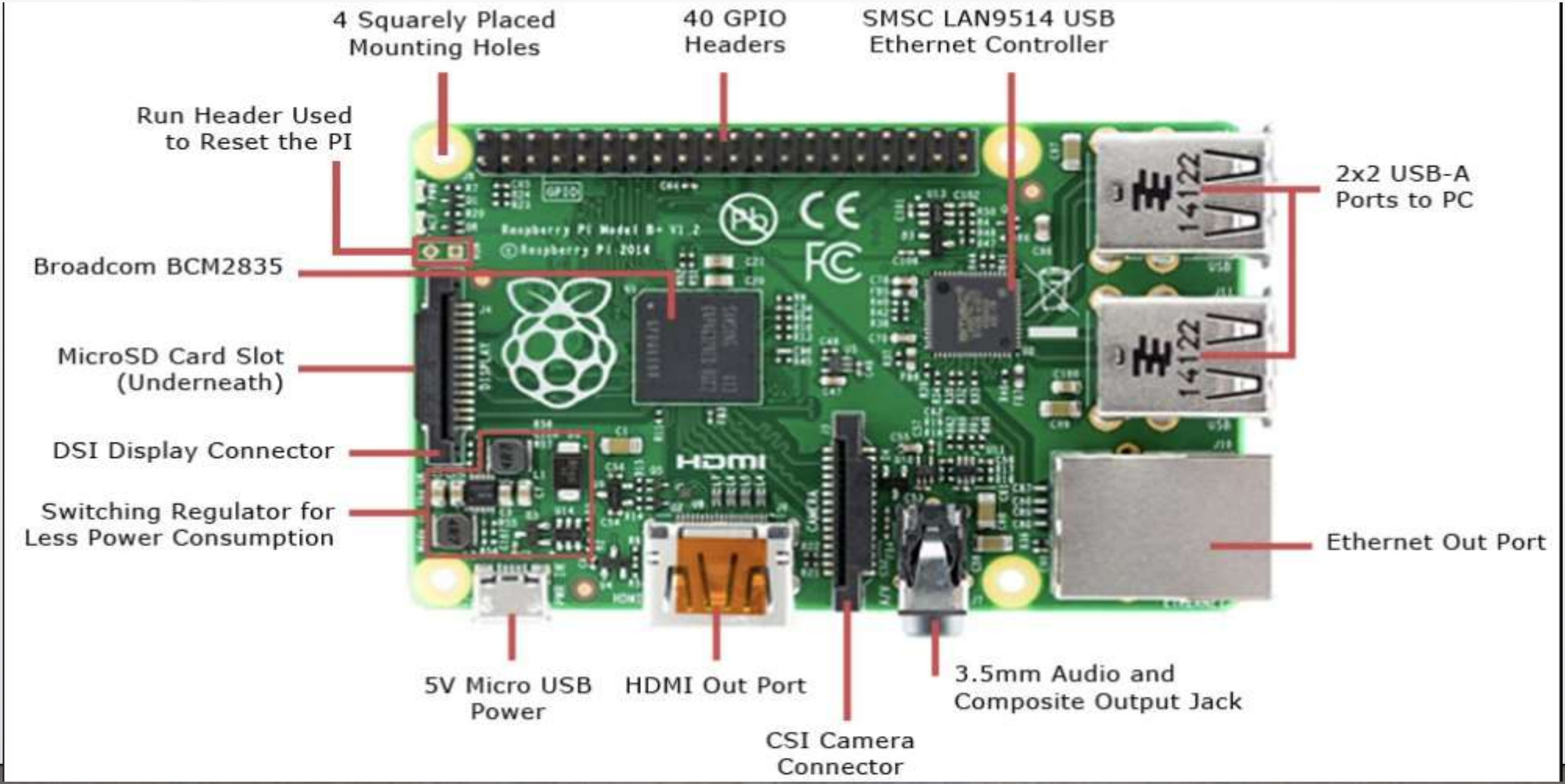
GPIO Pinout Diagram

Pi Model B/B+ — Pi Model B+

5V Power	2	1	3V3 Power
5V Power	4	3	GPIO2 SDA1 I2C
Ground	6	5	GPIO3 SCL1 I2C
GPIO14 UART0_TXD	8	7	GPIO4
GPIO15 UART0_RXD	10	9	Ground
GPIO18 PCM_CLK	12	11	GPIO17
Ground	14	13	GPIO27
GPIO23	16	15	GPIO22
GPIO24	18	17	3V3 Power
Ground	20	19	GPIO10 SPI0_MOSI
GPIO25	22	21	GPIO9 SPI0_MISO
GPIO8 SPI0_CE0_N	24	23	GPIO11 SPI0_SCLK
GPIO7 SPI0_CE1_N	26	25	Ground
ID_SC I2C ID EEPROM	28	27	ID_SD I2C ID EEPROM
Ground	30	29	GPIO5
GPIO12	32	31	GPIO6
Ground	34	33	GPIO13
GPIO16	36	35	GPIO19
GPIO20	38	37	GPIO26
GPIO21	40	39	Ground

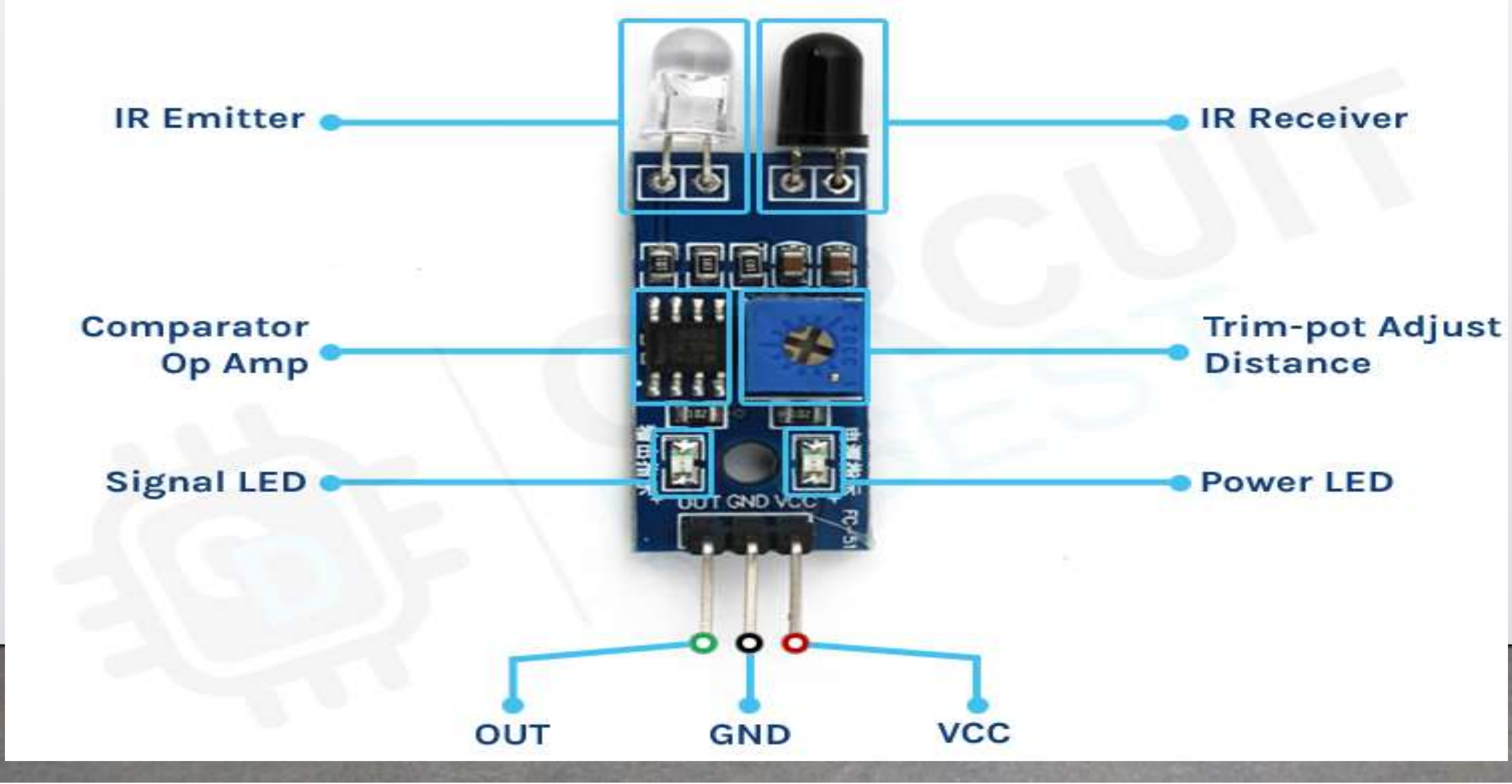

4 Squarely Placed Mounting Holes
40 GPIO Headers
SMSC LAN9514 USB Ethernet Controller
Run Header Used to Reset the PI
2x2 USB-A Ports to PC
Broadcom BCM2835
MicroSD Card Slot (Underneath)
DSI Display Connector
Switching Regulator for Less Power Consumption
Ethernet Out Port
5V Micro USB Power
HDMI Out Port
3.5mm Audio and Composite Output Jack
CSI Camera Connector

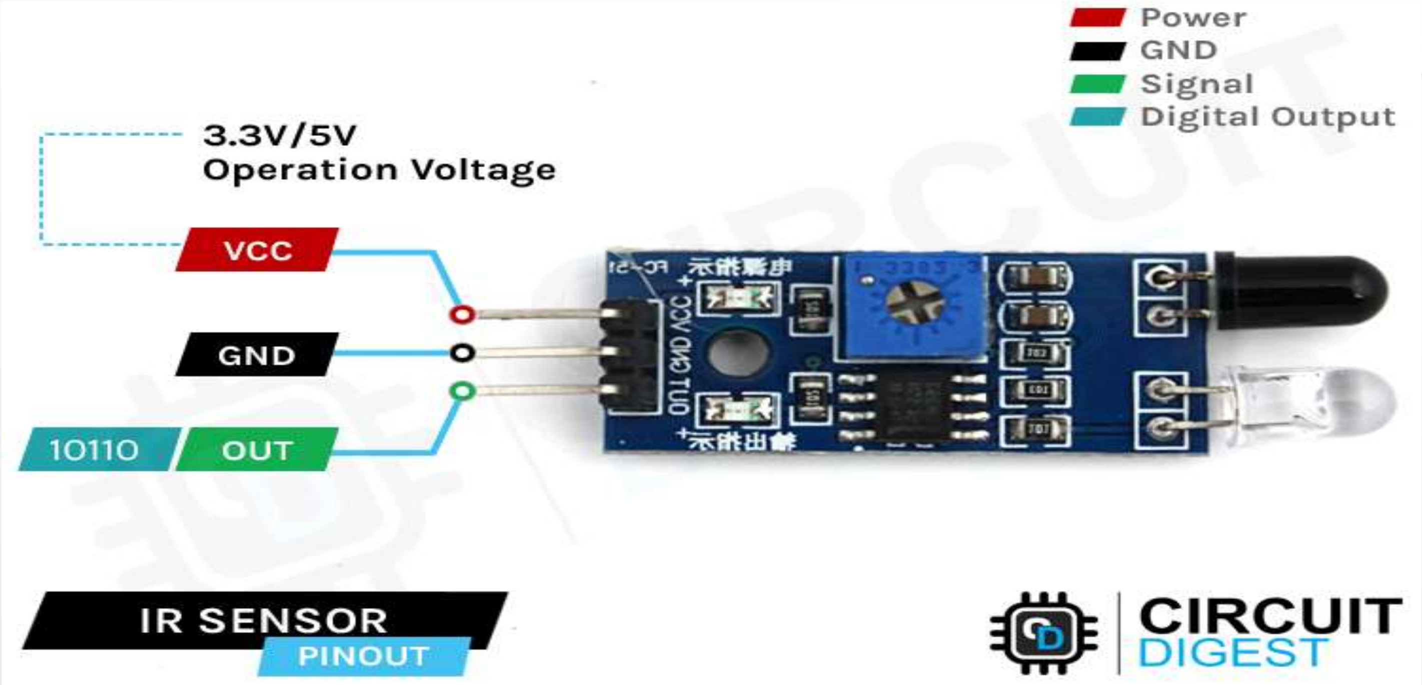
Raspberry pi board


4 Squarely Placed Mounting Holes
40 GPIO Headers
SMSC LAN9514 USB Ethernet Controller
Run Header Used to Reset the PI
2x2 USB-A Ports to PC
Broadcom BCM2835
MicroSD Card Slot (Underneath)
DSI Display Connector
Switching Regulator for Less Power Consumption
Ethernet Out Port
5V Micro USB Power
HDMI Out Port
CSI Camera Connector
3.5mm Audio and Composite Output Jack

Arduino IR Sensor

- An infrared sensor or IR Sensor is an electronic device that emits infrared lights to sense some aspect of the surroundings and can be employed to detect the **motion** of an object.

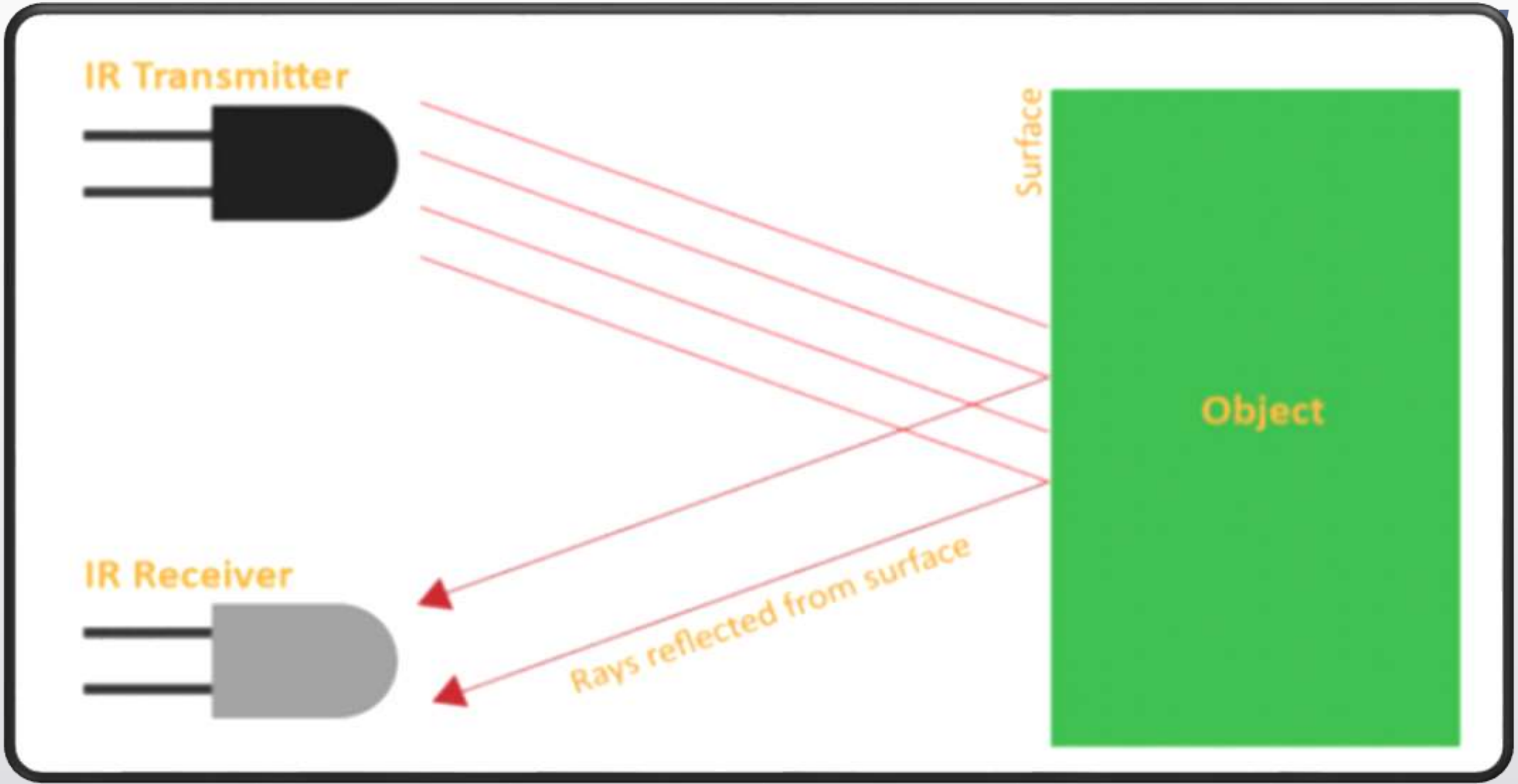




VCC is the power supply pin for the IR sensor which we connect to the 5V pin on the Arduino.

OUT pin is a 5V TTL logic output. LOW indicates no motion is detected; HIGH means motion is detected.

GND Should be connected to the ground of the Arduino.


IR Transmitter
Surface
Object
IR Receiver
Rays reflected from surface

Applications of IR sensor:

1. **Motion detection:** Used in smart home security systems to detect movement.

2. **Proximity sensing:** Detects objects in close range, such as in automatic doors.

3. **Environmental monitoring:** Measures heat levels for applications like smart thermostats or industrial safety.

4. **Remote control:** Enables communication between devices using IR signals.

IR sensor

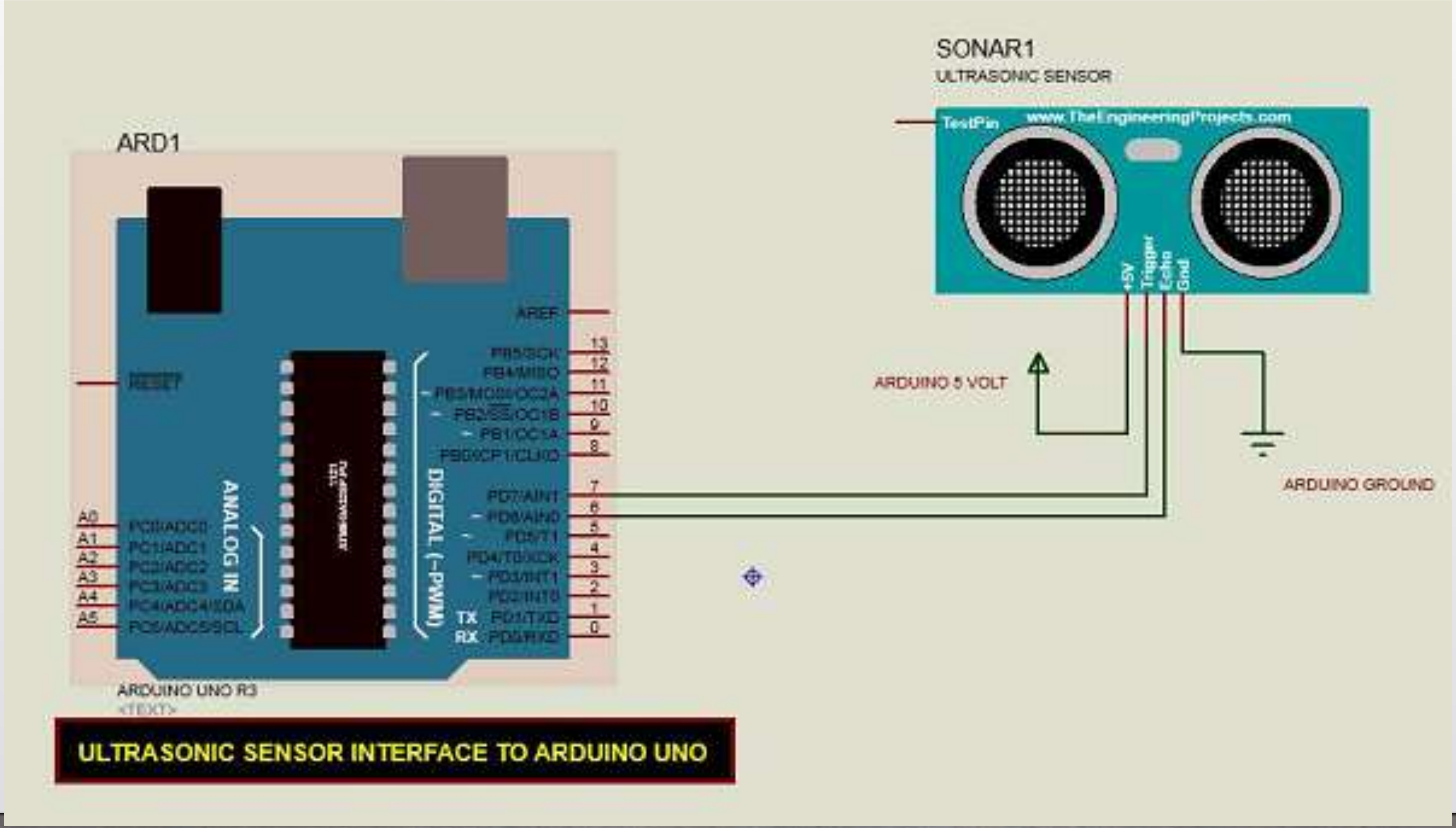
```
int IRPin = 2;
int led = 13;
int value;
voidsetup()
{
  pinMode(IRPin, INPUT);
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}
voidloop()
{
  value = digitalRead(IRPin);
  Serial.println(!value);
  if(digitalRead(IRPin)==0)
{
    digitalWrite(led, HIGH);
    Serial.println("object
detected");
}
```

```
else
{
    digitalWrite(led, LOW);
    Serial.println("object not detected");
    value = 0;
  }
}
```

Getting Started with the HC-SR04 Ultrasonic sensor

The sensor is composed of two ultrasonic transducers. One is transmitter which outputs ultrasonic sound pulses and the other is receiver which listens for reflected waves. It’s basically a SONAR which is used in submarines for detecting underwater objects.

The picture can't be displayed.



ULTRASONIC SENSOR INTERFACE TO ARDUINO UNO

The picture can't be displayed.

Working principles of Ultra sonic Sensor

Applications of Ultra sonic sensor

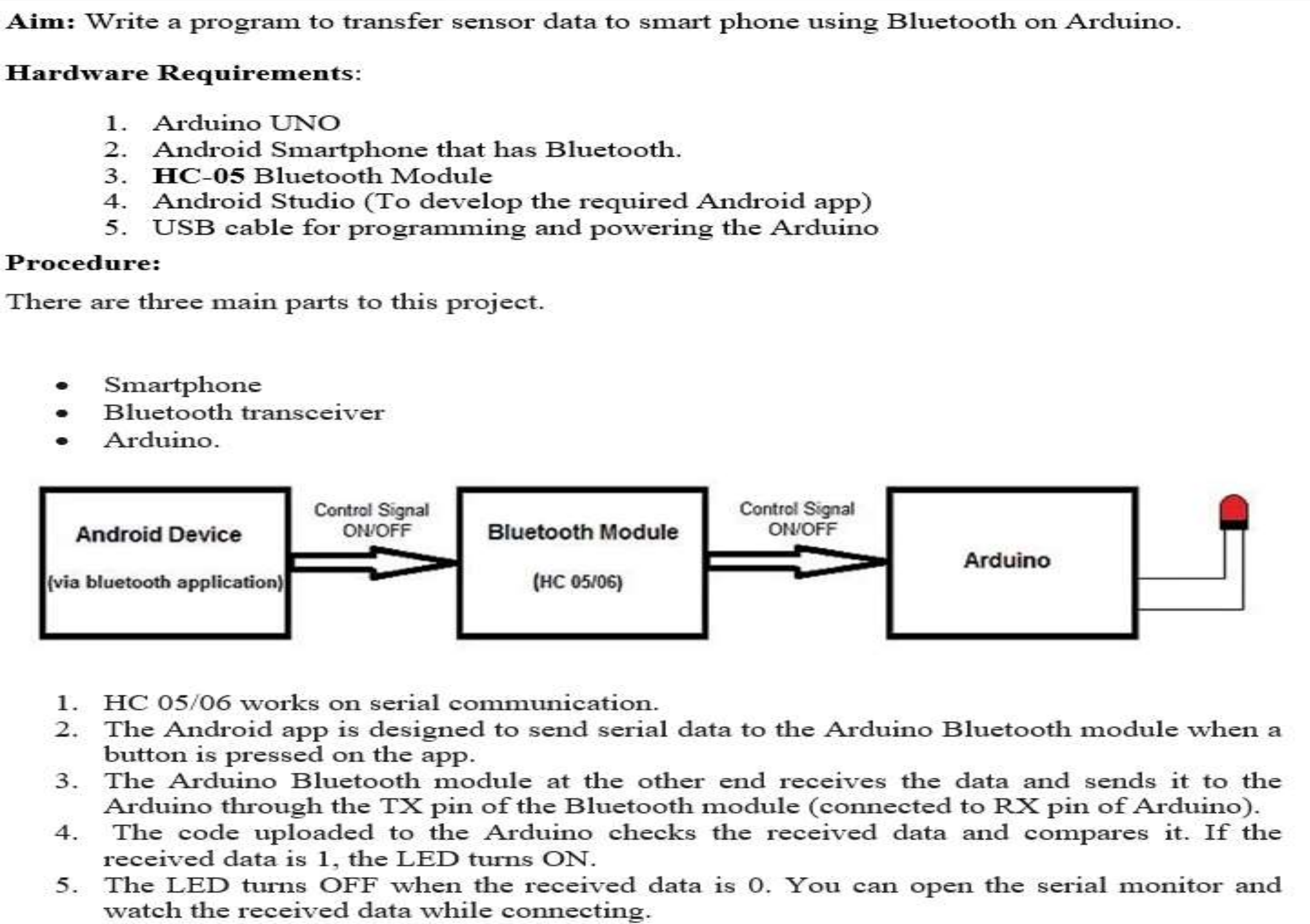
1. **Proximity detection:** Used in obstacle detection for robotics or autonomous vehicles.
2. **Distance measurement:** Measures distance in applications like parking sensors or smart water level monitoring.
3. **Smart agriculture:** Monitors soil levels or tank water levels.
4. **Security systems:** Detects intrusions in restricted areas.

Ultrasonic

```
#define ECHOPIN 7        // Pin to receive echo pulse
#define TRIGPIN 8
int led=12;
int a,b;
void setup()
{
  Serial.begin(9600);
  pinMode(ECHOPIN, INPUT);
  pinMode(TRIGPIN, OUTPUT);
  pinMode(led,OUTPUT);
}
void loop()
{
  digitalWrite(TRIGPIN, LOW);
  delayMicroseconds(2000);
  digitalWrite(TRIGPIN, HIGH);
  delayMicroseconds(1000);
  digitalWrite(TRIGPIN, LOW);
```

```
float a = pulseIn(ECHOPIN, HIGH);
  digitalWrite(led,HIGH);
  b= a*0.0344/2;
  Serial.print(b);
  Serial.println(" cm");
 delay(3000);
}
```

WEEK 3



Aim: Write a program to transfer sensor data to smart phone using Bluetooth on Arduino.

Hardware Requirements:

1. Arduino UNO
2. Android Smartphone that has Bluetooth.
3. **HC-05** Bluetooth Module
4. Android Studio (To develop the required Android app)
5. USB cable for programming and powering the Arduino

Procedure:

There are three main parts to this project.

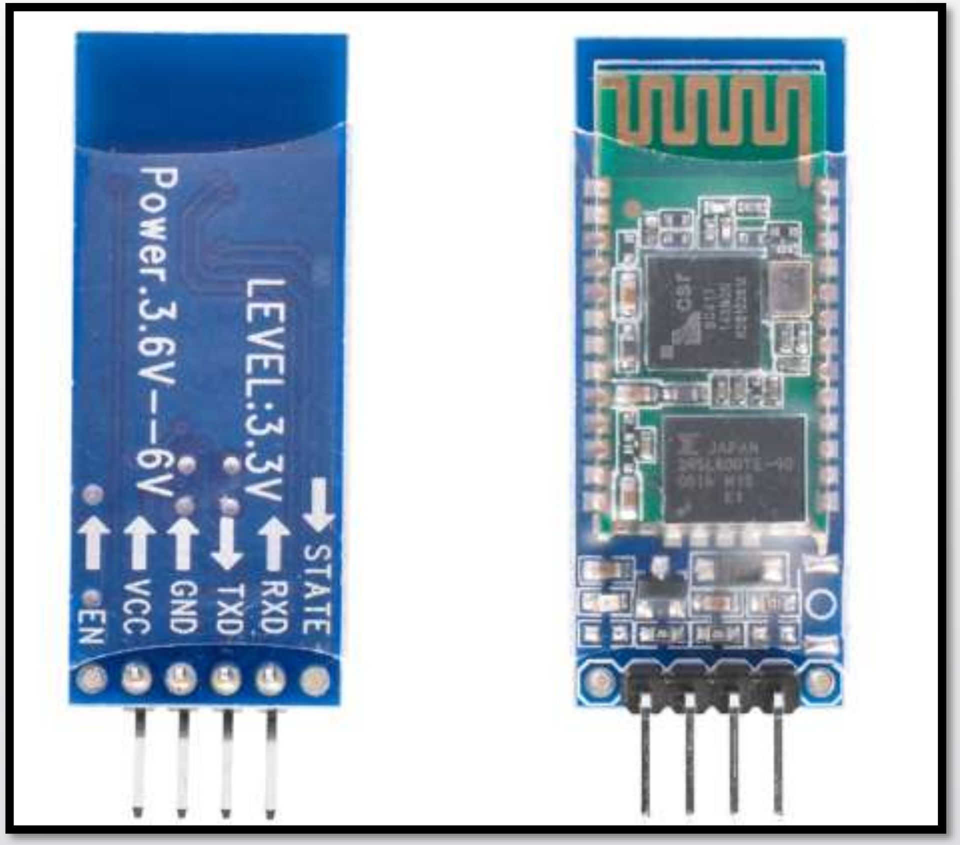
- Smartphone
- Bluetooth transceiver
- Arduino.

1. HC 05/06 works on serial communication.
2. The Android app is designed to send serial data to the Arduino Bluetooth module when a button is pressed on the app.
3. The Arduino Bluetooth module at the other end receives the data and sends it to the Arduino through the TX pin of the Bluetooth module (connected to RX pin of Arduino).
4. The code uploaded to the Arduino checks the received data and compares it. If the received data is 1, the LED turns ON.
5. The LED turns OFF when the received data is 0. You can open the serial monitor and watch the received data while connecting.

WEEK 3

Bluetooth device

The picture can't be displayed.

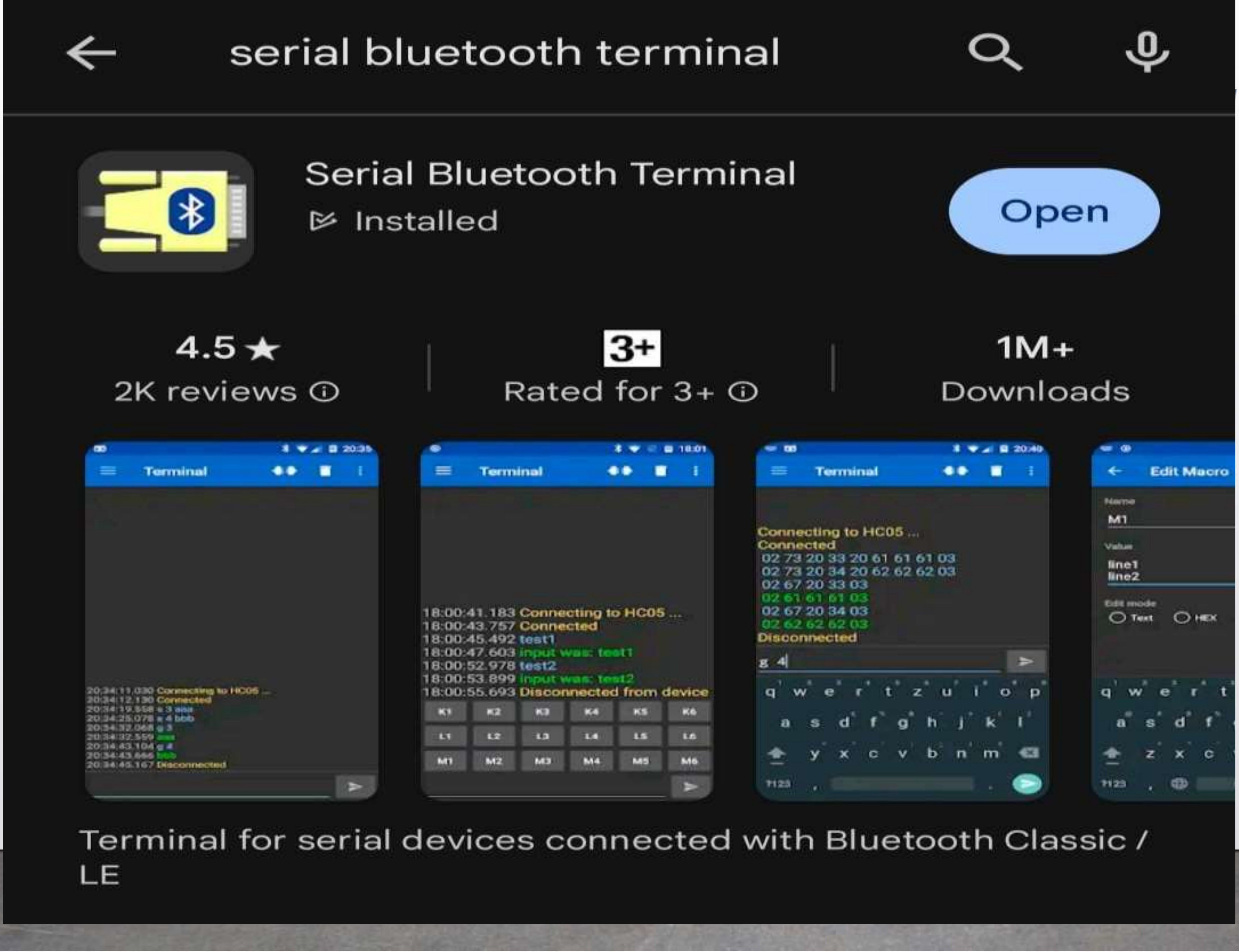


A Bluetooth sensor in IoT is a device that uses Bluetooth technology to wirelessly transmit data over short distances. It is widely used for low-power and low-cost IoT applications due to its ease of integration(separate people or things are brought together) and compatibility(occur together without problems).

- Key IoT applications include:

1. **Wearables:** Fitness trackers and smartwatches use Bluetooth to send health data to smartphones.
2. **Smart home devices:** Connects devices like smart locks, lights, and thermostats for remote control.
3. **Asset tracking:** Monitors and locates objects or vehicles in real-time.
4. **Healthcare:** Transfers data from medical devices like glucose monitors to mobile apps.
5. **Industrial IoT:** Enables machine-to-machine communication for monitoring and automation.

- Bluetooth sensors are essential for creating interconnected IoT ecosystems, enabling seamless communication and data sharing between devices.



```
20:34:11.030 Connecting to HC05 ...
20:34:12.130 Connected
20:34:19.558 s 3 aaa
20:34:25.078 s 4 bbb
20:34:32.068 g 3
20:34:32.559 aaa
20:34:43.104 g 4
20:34:43.666 bbb
20:34:45.167 Disconnected
```

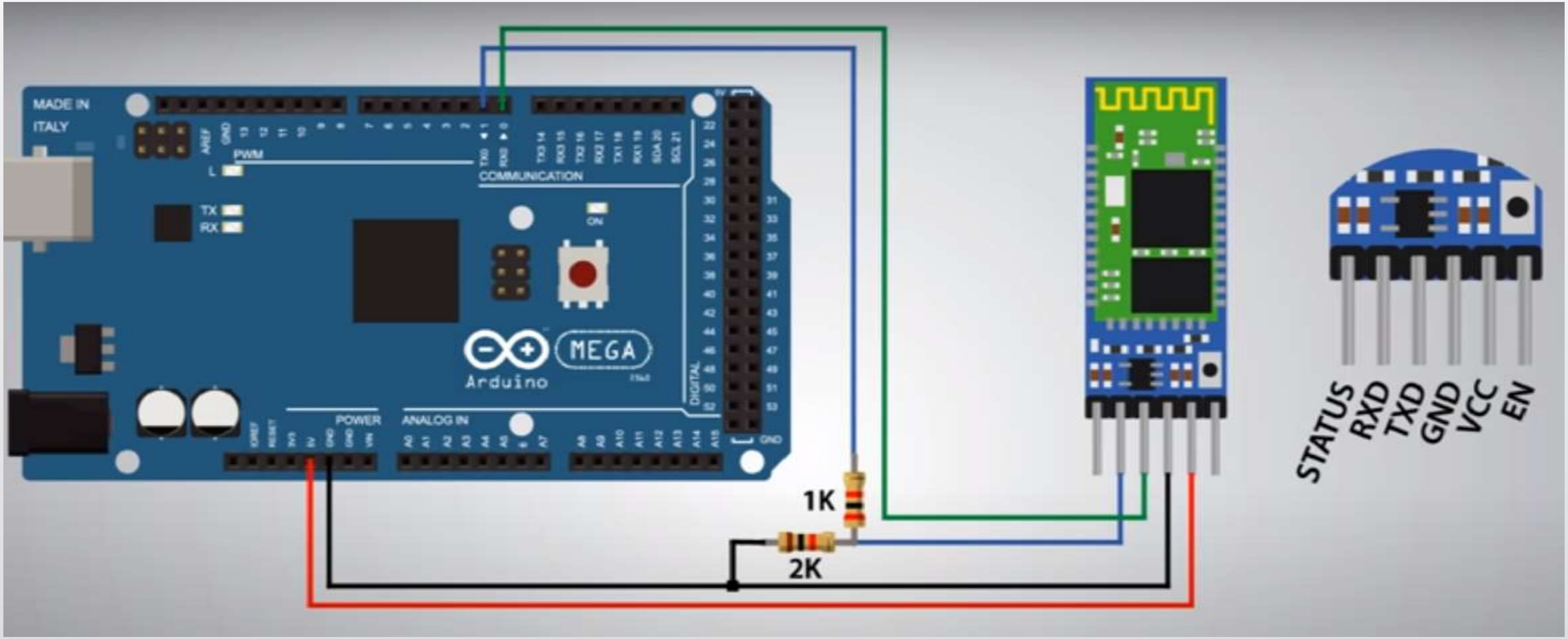
```
18:00:41.183 Connecting to HC05 ...
18:00:43.757 Connected
18:00:45.492 test1
18:00:47.603 input was: test1
18:00:52.978 test2
18:00:53.899 input was: test2
18:00:55.693 Disconnected from device
```

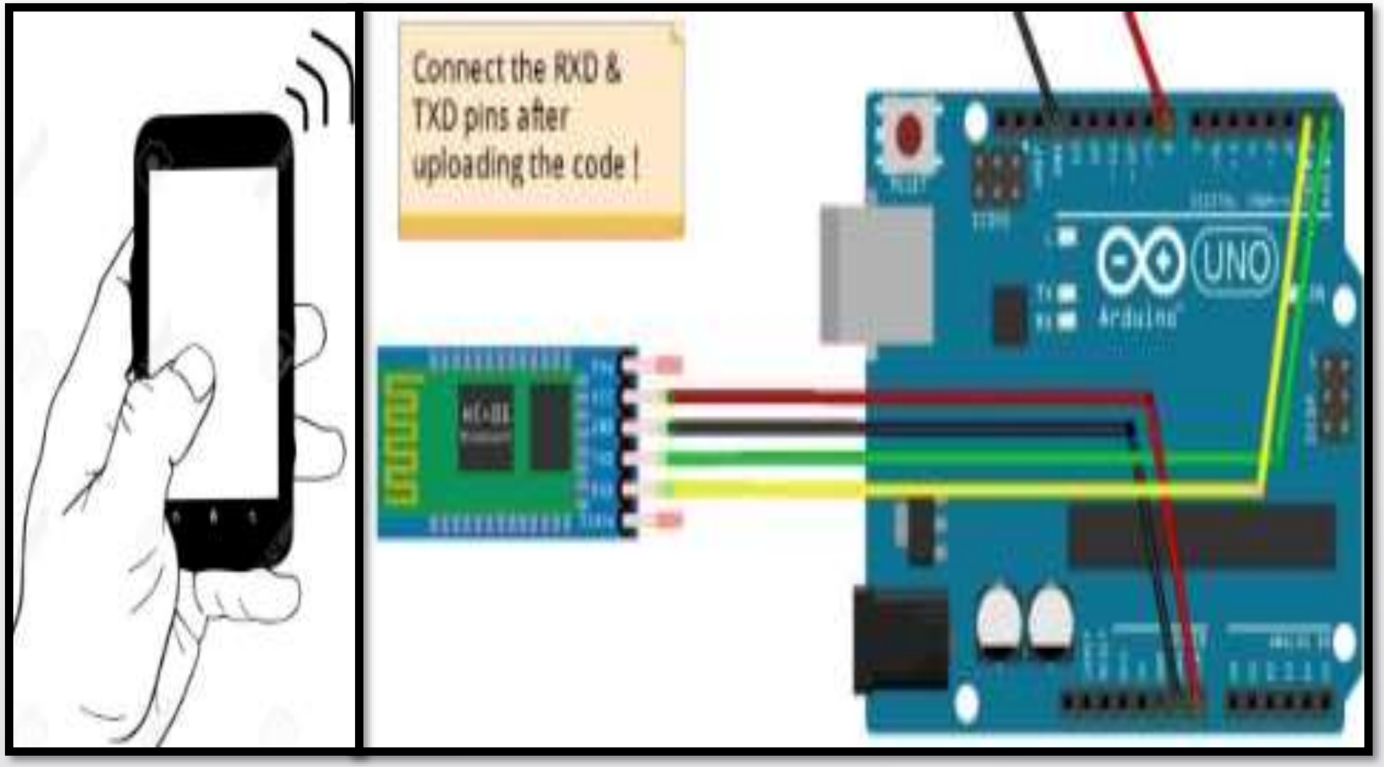
```
Connecting to HC05 ...
Connected
 02 73 20 33 20 61 61 61 03
 02 73 20 34 20 62 62 62 03
 02 67 20 33 03
 02 61 61 61 03
 02 67 20 34 03
 02 62 62 62 03
Disconnected
g 4
```

```
line1
line2
```

Terminal for serial devices connected with Bluetooth Classic / LE

Working Principles of how to transfer sensor data to smart phone using Bluetooth on Arduino





Give inputs from your bluetooth terminal App
0 LED is OFF
1 LED is ON

Source Code:

```
#include <SoftwareSerial.h>
SoftwareSerial Bluetooth(8, 9); // RX, TX (Interchange to 9,
8 on Arduino)
int LED = 13; // the on-board LED
int Data; // the data received
void setup()
 {
  Bluetooth.begin(9600);
  Serial.begin(9600);
  Serial.println("Waiting for command...");
  Bluetooth.println("Send 1 to turn on the LED. Send 0 to
turn Off");
  pinMode(LED,OUTPUT);
}
```

```
void loop()
{
 if (Bluetooth.available())
{ //wait for data received
     Data=Bluetooth.read();
      if(Data=='1'){
     digitalWrite(LED,HIGH);
     Serial.println("LED On!");
     Bluetooth.println("LED On!");
 }
  else if(Data=='0')
{
     digitalWrite(LED,LOW);
      Serial.println("LED Off!");
     Bluetooth.println("LED Off ! ");
 }
      else{;}
 }
     delay(1000);
}
```