USER'S MANUAL

FOR

RSAPDUINO LAB

Manufactured By

LOGSUN SYSTEMS

4A, Shree Sadgurunivas Soc.,16/5/1 Hingane
Khurd, Sinhgad Road, Pune-411051.
Ph.-020 24356456 / +91 8956711765

Email: - info@logsun.com

www.logsun.com

RASPDUINO DEVELOPMENT BOARD

INDEX

INTRODUCTION:

Thank you for purchasing the Logsun System's RASPDUINO LAB DEVELOPMENT Board. You will find it useful in developing RASPBERRY PI and ARDUINO based applications.

We are providing you on board interfacing facilities for
1) Trafic light control,
2) Lift elevator simulator
3) XBEE interfacing,
4) 16*2 LCD interfacing
5) 4*4 Keboard interfacing
6) LM35 Temperature sensor
7) IR sensor,relay interfacing
8) ADC/DAC
9) DC/stepper motor Interfacing(5V)
10) LED switch logic

This development board also supports IOT environment

INTRODUCTION TO INTERNET OF THINGS (IOT):

IOT is short for Internet of Things. The Internet of Things refers to the ever-growing network of physical objects that feature an IP address for internet connectivity, and the communication that occurs between these objects and other Internet-enabled devices and systems.

The internet of things, or **IoT**, **is** a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers (UIDs) and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.

WORKING OF IOT:

The Internet of Things (IoT), also sometimes referred to as the Internet of Everything (IoE), consists of all the web-enabled devices that collect, send and act on data they acquire from their surrounding environments using embedded sensors, processors and communication hardware. These devices, often called "connected" or "smart" devices, can sometimes talk to other related devices, a process called **machine-to-machine** (M2M) communication, and act on the information they get from one another.

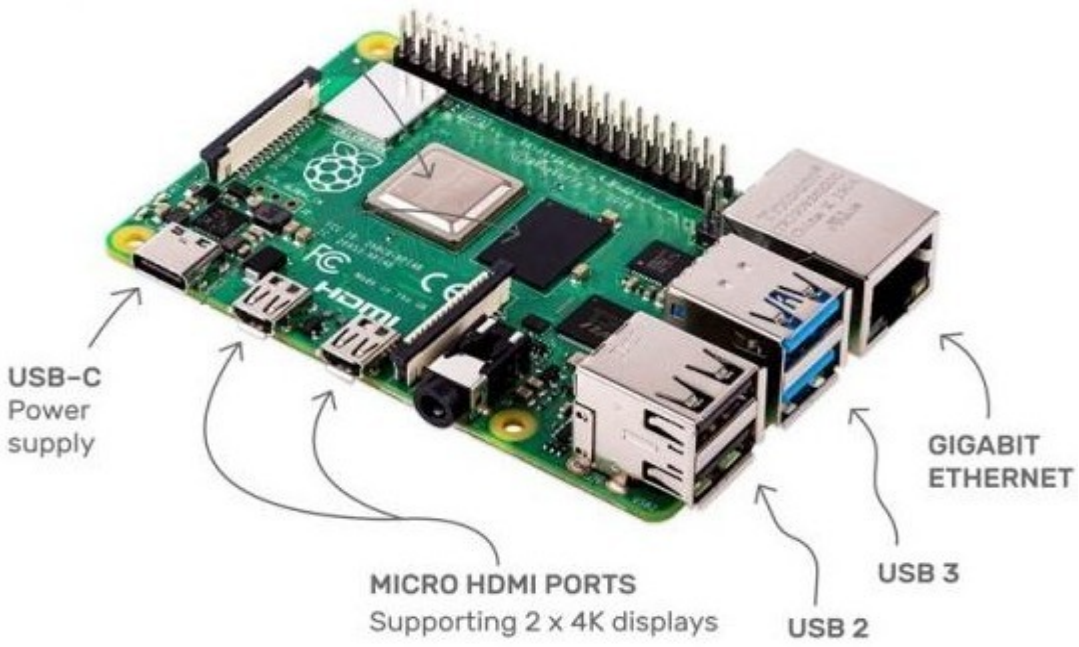
Humans can interact with the gadgets to set them up, give them instructions or access

the data, but the devices do most of the work on their own without human intervention. Their existence has been made possible by all the tiny mobile components that are available these days, as well as the always-online nature of our home and business networks.

APPLICATIONS OF IOT:

- Smart home. Smart Home clearly stands out, ranking as highest Internet of Things application on all measured channels.
- Wearables
- Smart City
- Smart grids
- Industrial internet
- Connected car
- Connected Health (Digital health/Telehealth/Telemedicine)
- Smart retail

1. Introduction of Raspberry Pi



Raspberry Pi-4

The Raspberry Pi 4 Model B] with a 1.5 GHz 64-bit quad core ARM Cortex-A72 processor, on-board 802.11ac Wi-Fi, Bluetooth 5, full gigabit Ethernet (throughput not limited), two USB 2.0 ports, two USB 3.0 ports, 1–8 GB of RAM, and dual-monitor support via a pair of micro HDMI (HDMI Type D) ports for up to 4K resolution.

Specification

Processor: Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz

Memory: 1GB, 2GB, 4GB or 8GB LPDDR4 (depending on model) with on-die ECC

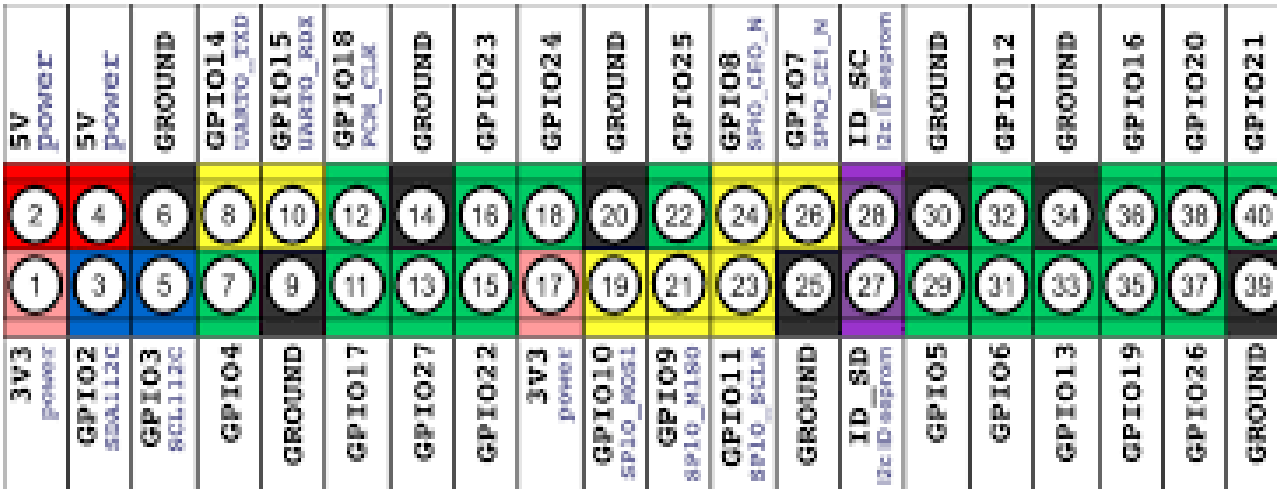
Connectivity: 2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports. GPIO: Standard 40-pin GPIO header (fully backwards-compatible with previous boards)

Video & sound: 2 × micro HDMI ports (up to 4Kp60 supported) 2-lane MIPI DSI display port 2-lane MIPI CSI camera port 4-pole stereo audio and composite video port

Multimedia: H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 graphics

SD card support: Micro SD card slot for loading operating system and data storage Input power: 5V DC via USB-C connector (minimum 3A1) 5V DC via GPIO header (minimum 3A1) Power over Ethernet (PoE)–enabled (requires separate PoE HAT) Environment: Operating temperature 0–50ºC

GPIO DETAIL



Configuration setting of Raspberry Pi

How to Install Operating System in Raspberry Pi SD Card

This Raspberry Pi is the best way to get started on working with the IoT (Internet Of Things) and to build your knowledge to expand it to other applications. So, we made a step-by-step guide for Pi beginners so you can learn various methods that can be used to make your application of choice.

We have divided the whole Raspberry Pi IoT project into two parts. The first part consists of setting up the Raspberry Pi and interfacing it with the sensors & actuators. The second part covers building the various applications and connecting it to the server.

If you aren't familiar with Linux and terminal, check out Basic Linu Commands for Beginners.

Basic Commands -

1. ls — Use the "Is" command to know what files are in the directory you are in. You can see all the hidden files by using the command “ls”.

```
nayso@Alok-Aspire:~$ ls
Desktop           itsuserguide.desktop  reset-settings     VCD_Copy
Documents         Music                 School_Resources   Videos
Downloads         Pictures              Students_Works_10
examples.desktop  Public                Templates
GplatesProject    Qgis Projects         TuxPaint-Pictures
```

1. cd — Use the "cd" command to go to a directory. Remember, this command is case sensitive, and you have to type in the name of the folder exactly as it is. But there is a problem with these commands. Imagine you have a folder named “Raspberry Pi”. In this case, when you type in “cd Raspberry Pi”, the shell will take the second argument of the command as a different one, so you will get an error saying that the directory does not exist. Here, you can use a backward slash. That is, you can use “cd Raspberry\ Pi” in this case. Spaces are denoted like this: If you just type “cd” and press enter, it takes you to the home directory. To go back from a folder to the folder before that, you can type “cd ..” . The two dots represent back.

```
nayso@Alok-Aspire:~$ cd Downloads
nayso@Alok-Aspire:~/Downloads$ cd
nayso@Alok-Aspire:~$ cd Raspberry\ Pi
nayso@Alok-Aspire:~/Raspberry Pi$ cd ..
nayso@Alok-Aspire:~$
```

Intermediate Commands-

1. **echo** — The "**echo**" command helps us move some data, usually text into a file.
2. **cat —** Use the **cat** command to display the contents of a file. It is usually used to easily view programs.

```
nayso@Alok-Aspire:~/Desktop$ echo hello, my name is alok >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
nayso@Alok-Aspire:~/Desktop$ echo this is another line >> new.txt
nayso@Alok-Aspire:~/Desktop$ cat new.txt
hello, my name is alok
this is another line
```

3. **nano, vi, jed — nano** and **vi** are already installed text editors in the Linux command line. The **nano** command is a good text editor that denotes keywords with color and can recognize most languages. And **vi** is simpler than **nano**. You can create a new file or modify a file using this editor. You can save your files after editing by using the sequence Ctrl+X, then Y (or N for no).

4. **sudo** — A widely used command in the Linux command line, **sudo** stands for "SuperUser Do". So, if you want any command to be done with administrative or root privileges, you can use the **sudo** command.

Note: We'll be remotely accessing the Raspberry Pi's terminal through SSH, so you won't need a dedicated monitor, mouse or keyboard in any part of this tutorial. You would also want to download the following software:

- Win32 Disk Imager
- PuTTY
- Advanced IP Scanner

Tips & Tricks:

You can use the **clear** command to clear the terminal if it gets filled up with too many commands.

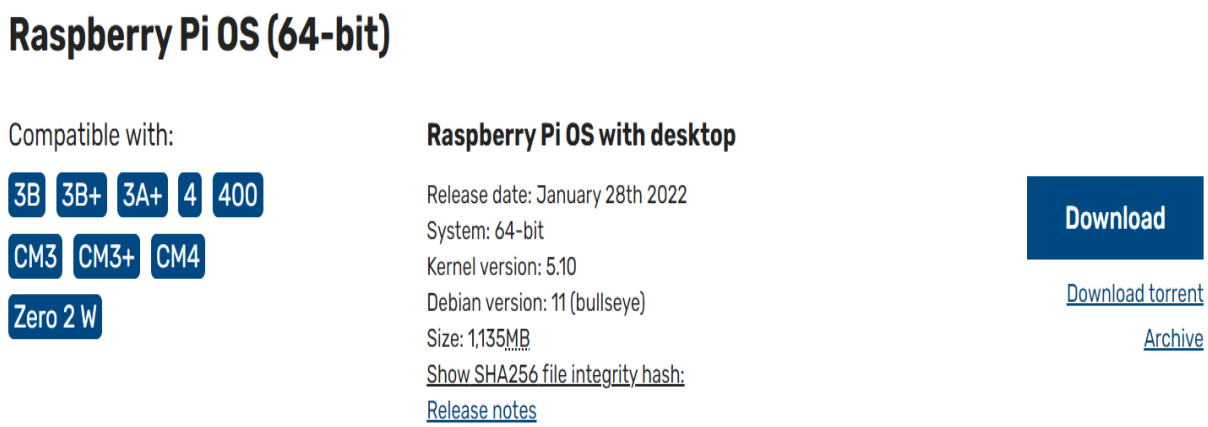
TAB can be used to fill up in terminal. For example, you just need to type "**cd Doc**" and then **TAB** and the terminal fills the rest up and makes it "**cd Documents**".

- **Ctrl+C** can be used to stop any command in terminal safely. If it doesn't stop with that, then **Ctrl+Z** can be used to force stop it.
- You can exit from the terminal by using the **exit** command.
  You can power off or reboot the computer by using the command **sudo halt** and **sudo reboot**.

Setting Up the Pi for the Raspberry Pi IoT Project:

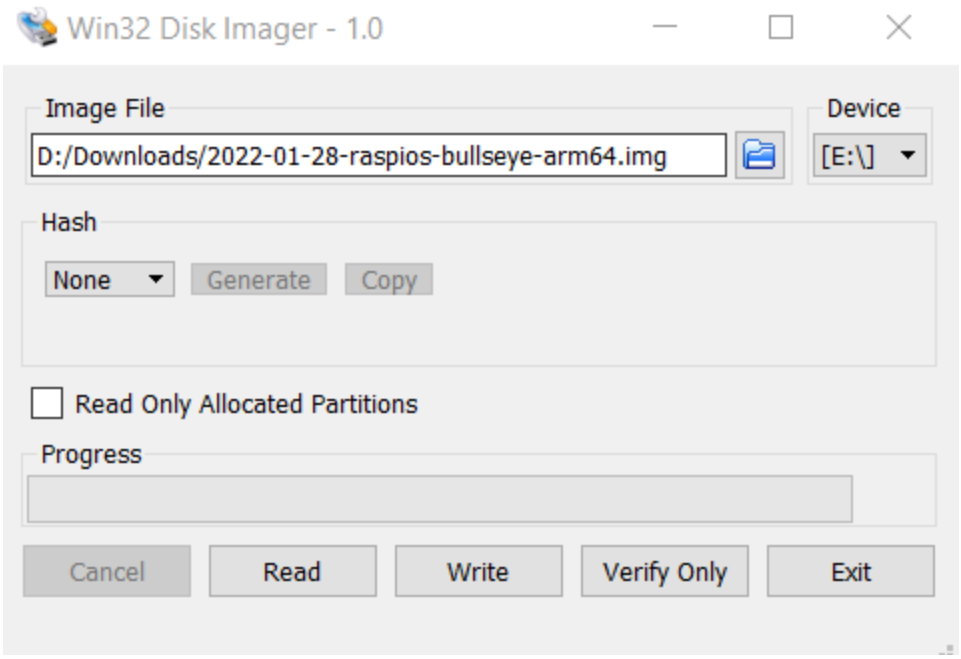
If you already have a Raspberry Pi set up, move over to the next step to start with your Raspberry Pi IoT project. Otherwise, download the Raspbian OS for your Pi. There are many other distributions you can use, but Raspbian remains the most common and convenient for beginners. Visit https://www.raspberrypi.org/downloads/raspbian/ to download the Raspbian OS.

Extract the *.img file from the downloaded zip folder and write it to your SD card.

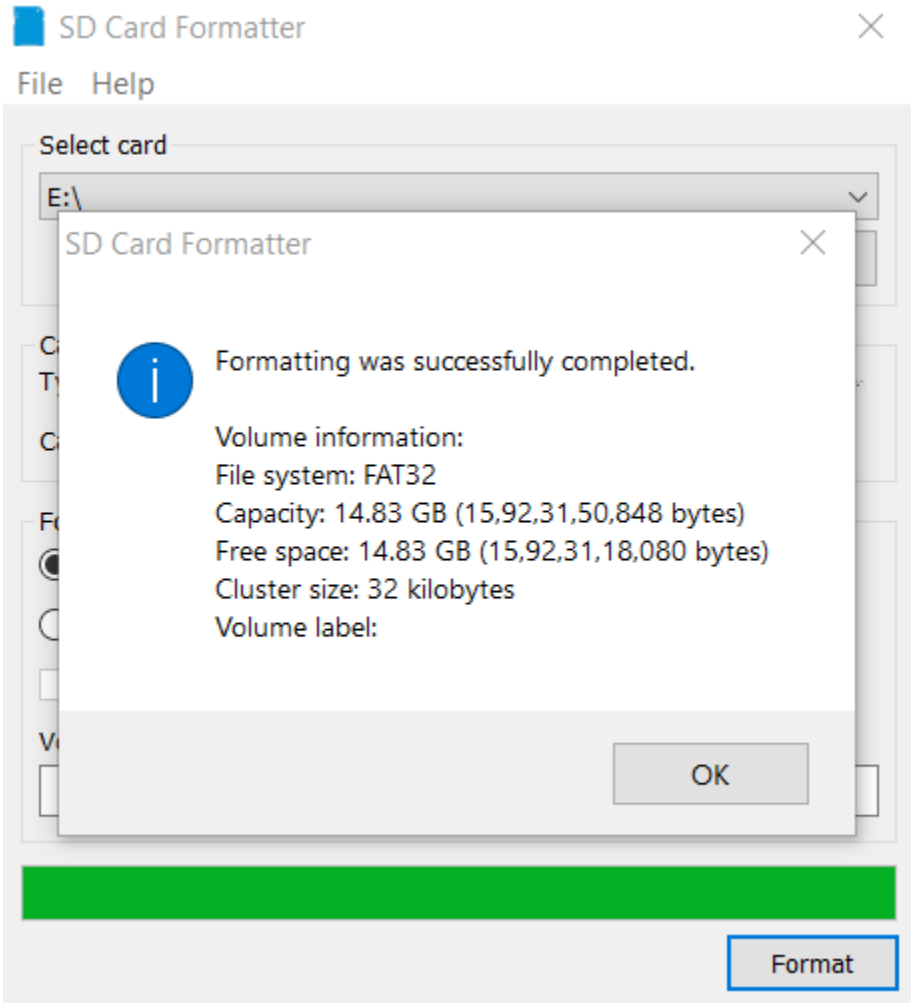


For Windows Users:

1. Insert the SD card reader with the SD card in it. Figure out the Drive assigned to it.
2. Run Win32 Disk Imager (you may have to run this as administrator), and select the extracted image file and the drive letter. Be very careful to select the correct drive—you do not want to unintentionally destroy other data. It should look something like this:



3. Click "Write". Wait for the process to complete and eject the SD card.



How to Connect Raspberry Pi on Windows Platform

Booting Your Raspberry Pi IoT System:

Getting a dedicated monitor, mouse, and keyboard to use your Raspberry Pi might become an unnecessary hassle. Access to the terminal is sufficient to get most things done. So we eliminate the need for extra hardware by logging into the Pi using your personal laptop through SSH. Latest versions of Raspbian come with SSH enabled by default, so you can run Pi remotely even while setting it up for the first time.

Things to do right after the Starting Raspberry Pi 1st time.

1. Select location, language and time location.
2. Click on US keyboard and English Language.
3. Set a new password and don't keep it as "raspberry" as that's the default password and it won't change.
4. If you have internet access select your network and enter your password to connect.
5. Let the system check for software updates and install those updates*.
6. Click on Finish
7. Click on the Pi Menu > Preferences click on 'Raspberry Pi configuration'
8. Go to interfaces.
    1. All the interfaces are disabled by default.
    2. Enable all the interfaces except for Remote GPIO (Enable if you really need it).
    3. Click OK and it will ask you to reboot.
    4. Click on reboot.

*If the software update fails, then open terminal (Ctrl + Alt + T)
Type "sudo apt-get update && sudo apt-get upgrade"
Press 'Y' and press enter if asked.
'update' will update the repositories and 'upgrade' will install all the new updates.
Reboot once the process is complete.
Complete step 8 > 2 before reboot to avoid rebooting too many times.

Raspberry Pi 4

Install AdaFruit Library for DHT22 Practical it is required.

1. Open Terminal (Ctrl + Alt + T).
2. Type “sudo apt-get update && sudo apt-get upgrade”
3. Type “sudo apt-get install python3-dev python3-pip”
4. Type “sudo python3 -m pip install --upgrade pip setup tools wheel”
5. Type “sudo pip3 install Adafruit_DHT”

Don’t forget to add ‘sudo’ as it gives you root access.

Enable SSH (ttyS) for MQTT*

1. Type “sudo apt-get install python3-serial”

2. Copy “dmesg | grep tty” and paste.
    1. This will show you all the available ports.

    2. ttyAMA0 might be the only enabled port.

3. Type “sudo rasp-config”
    1. Use arrow pointer to select the correct option, then press tab to select Yes or No option.

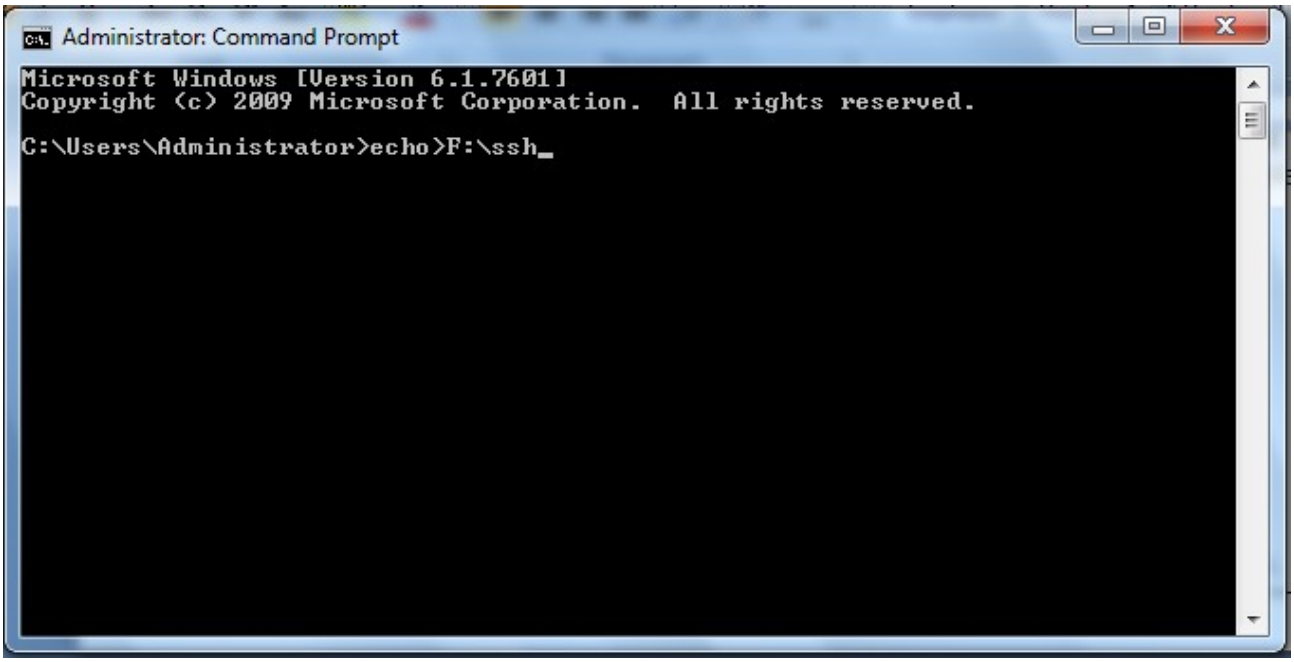
    2. Select Interfacing Options

    3. Select P6 Serial
        1. Select *NO* for “Login shell access over serial”
        2. Select *YES* for “Enable serial port hardware”
        3. Select *YES* to Reboot now.

If SSH is not enabled then do the following procedure:

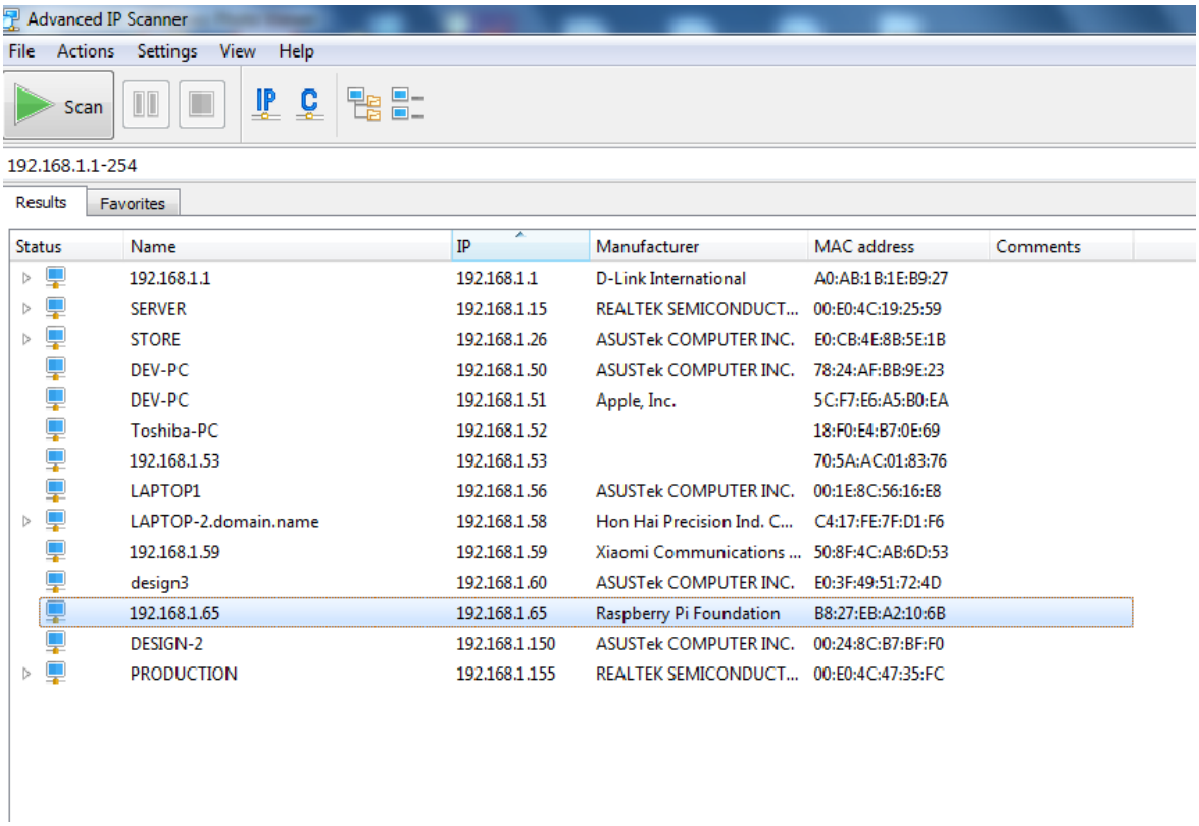
- Click on Start - All Programs – Accessories - Command Prompt
- Enter command : *echo>F:\ssh<Enter>*

You will see the window like this:

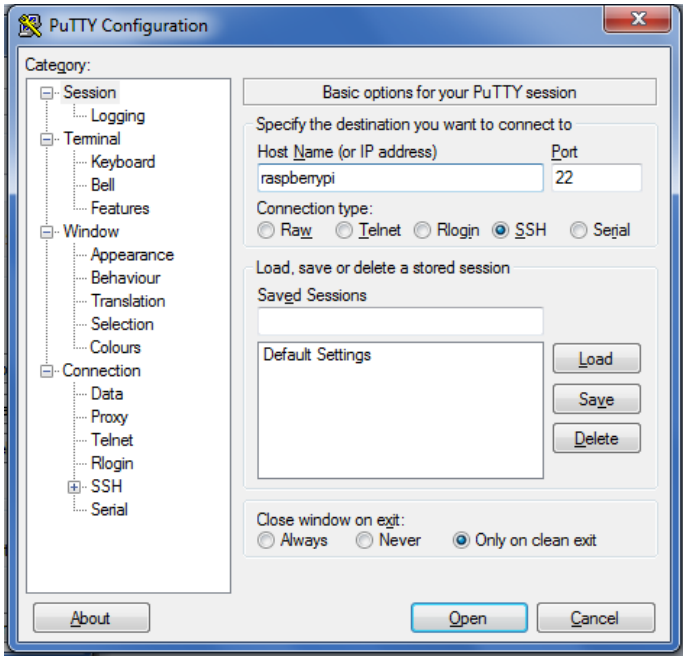


Note: *Here F is Removable Disk Drive Letter like of SD Card. Ensure your SD Card drive letter & make changes in above command accordingly.*

- **Get a router with DHCP enabled**. This is necessary because we want our Pi to have a unique IP address to be able to connect to it. Connect both your laptop and your Pi to the router via Ethernet cables. Your laptop and Pi now share a local area network and can identify each
- other using their unique IP address.
- To use SSH, you'll need the **IP address of your Pi**.
- Connect Raspberry pi with 5V power supply
- Open the Advance IP Scanner Software
- *Run the Advanced IP Scanner. This will list out the IP addresses of all devices on your network and their manufacturer. You'll see something like this:*

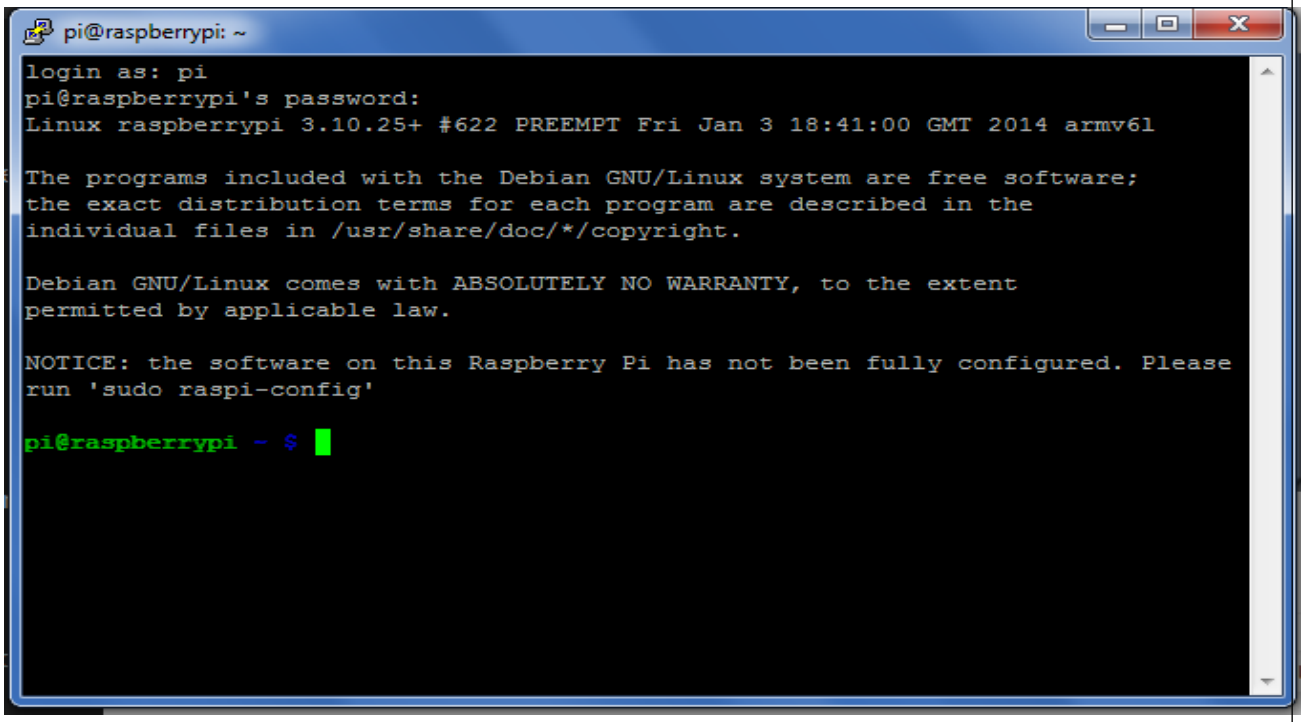


Use PuTTY to access Pi's Terminal. Run PuTTY and simply enter the IP address determined in the above step.

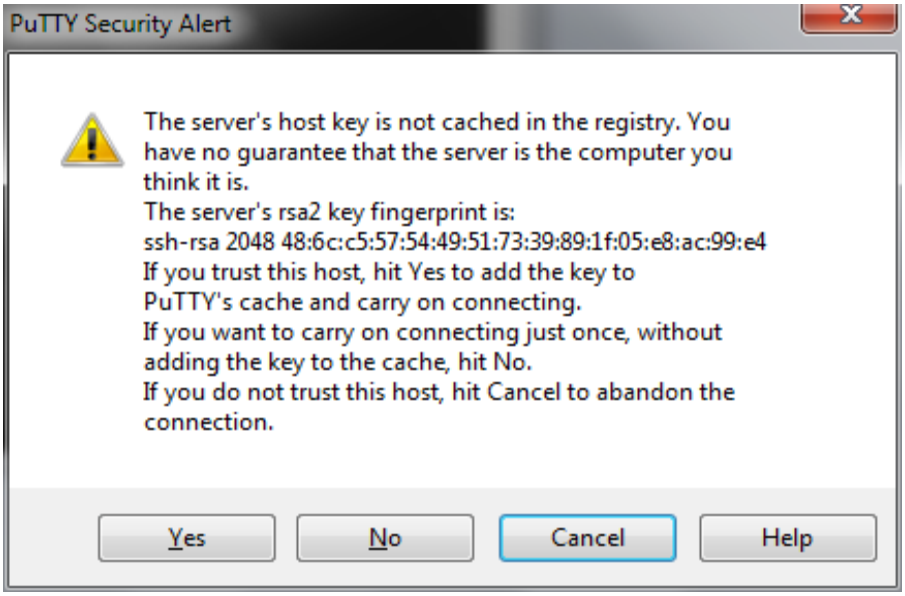
- 

- Subsequently, you will receive a login prompt. Use login id as **pi** and password

as **raspberry**.



- **Side Note:** For first-time login, you'll receive a warning for a security alert. Click on "Yes" and proceed. You have now opened a terminal session on your Raspberry Pi, which can be accessed through your Windows laptop.
-



-

How to Connect Raspberry Pi on Linux Platform

- To know your Raspberry Pi IP address, turn ON your laptop/desktop Wi-Fi.
- Connect Ethernet cable between Raspberry Pi & laptop/desktop.
- Click on Wi-Fi icon.
- Click on VPN connection – configure VPN
- Click on add
- Choose connection type as Ethernet
- Click on create
- Connection name- Raspberry Pi (You can assign any name)
- Go to IP4 settings tab.
- Click on Method & select as share to other computers.
- Click on save.
- Yu will see Raspberry Pi Connected notification.
- Open your terminal window
- enter the following command: cat /var/lib/misc/dnsmasq.leases<Enter>
- You will get Raspberry Pi IP address.
- Now enter the following command : ssh pi@10.42.0.66 <Enter>
- Enter the password as raspberry.
- You'll receive a warning for a security alert. Click on "Yes" and proceed. You have now opened a terminal session on your Raspberry Pi, which can be accessed through your Windows laptop.
- Please note that the cursor won't move forward while entering the password due to default settings.

EXPERIMENT No 1-

LED LOGIC

AIM:- To study the interfacing of LED and SWITCH to the Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to "XBEE-LED-SWITCHES" section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder "PYTHON PROGRAMMS"

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls.**

8) To run this specific program enter the command >> **python ledlogic.py**

9) If you want to see program enter the command >> **sudo nano ledlogic.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that LEDs will glow when you **ON** the switches and OFF when you **OFF** the switches.

Note: FRC port connection must be proper to run the program.

EXPERIMENT No 2.

LCD DISPLAY

AIM: - To study the interfacing of LCD (16*2) with the Raspberry pi.(4 bit)

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, LCD.

PROCEDURE:

1) Connect Raspberry pi board.
2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.
3) Switch ON the power supply.
4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “LCD-KEYPAD” section.
5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**
6) Now we are into the folder PYTHON PROGRAMMS
7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**
8) To run this specific program enter the command >> **python lcd.py**
9) If you want to see program enter the command >> **sudo nano lcd.py**
10) To exit from the program press **Ctrl-Z.**
11) You will observe that it displays **“LOGSUN SYSTEMS”** on LCD display.

THEORY:

Liquid crystal display:

LCD can be connected to the Raspberry pi through the GPIO pins. LCD is connected in the 4-bit mode or 8-bit mode. And the standard subroutine is given with the development board.

So that the application can be easily demonstrated and also for further implementation the subroutine can be easily embedded for which one has to do very few changes. Wide range of instruction functions: Clear displays, cursor home, display ON/OFF, cursor ON/OFF, cursor shift, display shift.

4-bit mode

RS, RW, EN are 1&2 are short.

8-bit mode

RS, RW, EN are 2&3 are short

- **INSTALLING THE RPLCD LIBRARY**

The RPLCD library can be installed from the Python Packages Index, or PIP. It might already be installed on your Pi, but if not enter this at the command prompt to install it.

sudo apt-get install python-pip

After you get PIP installed, install the RPLCD library by entering:

sudo pip install RPLCD

EXPERIMENT No 3.

LCD WITH KEYPAD INTERFACING

AIM: To study the interfacing of LCD and Keypad with Raspberry Pi.(4 bit)

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, LCD, 4*4 matrix keypad.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “LCD-KEYPAD” section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS.

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python keypad.py**

9) If you want to see program enter the command >> **sudo nano keypad.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that when you press a key (0-F) it will show on LCD display.

EXPERIMENT No 4.
TRAFFIC LIGHT SIGNAL

AIM: To study the interfacing of LED to make traffic light signal with Raspberry Pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “TRAFFIC LIGHT” section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS.

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python traffic.py**

9) If you want to see program enter the command >> **sudo nano traffic.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that red led’s of 1st and 3rd stage is ON for 10 sec. Green led’s of both sides are on for 5 sec. Then yellow led’s 5 times blinks. And red led’s OFF. And same action happens for section 2nd and 4th.

EXPERIMENT No 5.

INTERFACING IR SENSOR WITH RASPBERRY PI

AIM: To study the interfacing of IR sensor with Raspberry Pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on board. And connect 26 pin FRC cable from the board to “ADC-DAC-TMP-IR” section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS.

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python ir.py**

9) If you want to see program enter the command >> **sudo nano ir.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that when IR is on or obstacle is detected that time “Ir is detected” prints on the terminal, or relay will switched on. And red led is on when ir Output is high.

Experiment No 6.

INTERFACING DC MOTOR TO RASPBERRY PI

AIM: To study the interfacing of dc motor to Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, DC motor, external12v dc power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “DC & STEPPER MOTOR” section, also connect the 12V power supply externally.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS.

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python dcmotor.py**

9) If you want to see program enter the command >> **sudo nano dcmotor.py**

10) To exit from the program press **Ctrl-Z**.

11) You will observe that when you press the switch **START** motor will start rotating in forward direction. And when you press the switch **REV** motor will rotate in opposite or reverse direction.

12) When you press switch **INC** the speed of motor increase slowly, as you press the same switch again. And when you press switch **DEC** the speed of motor decrease slowly, as you press same switch again. When you press switch **STOP** the motor will stop rotating.

NOTE: - Connect both jumpers (JP2 & JP3) on DC side.

Experiment No 7.

INTERFACING STEPPER MOTOR TO RASPBERRY PI

AIM: To study the interfacing of stepper motor to Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, stepper motor, external12v dc power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “DC & STEPPER MOTOR” section, also connect the 12V power supply externally.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python stepper.py**

9) If you want to see program enter the command >> **sudo nano stepper.py**

10) To exit from the program press **Ctrl-Z**.

11) You will observe that when you press the switch **START** motor will start rotating stepwise in forward direction. And when you press the switch **REV** motor will rotate in opposite or reverse direction.

12) When you press switch **INC** the speed of motor increase slowly, as you press the same switch again. And when you press switch **DEC** the speed of motor decrease slowly, as you press same switch again.

13) When you press switch **STOP** the motor will stop rotating.

NOTE:- Connect both jumpers(JP2 &JP3) on STP side.

Experiment No 8.

INTERFACING RELAY WITH RASPBERRY PI

AIM: To study interfacing of relay with Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to "ADC-DAC-TMP-IR" section.

5) Open the Terminal window of Raspberry pi and enter the command >>
cd /home/pi/PYTHON PROGRAMMS.

6) Now we are into the folder PYTHON PROGRAMMS

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python relay.py**

9) If you want to see program enter the command >> **sudo nano relay.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that relay is ON for 5sec and OFF for 5 sec.

Experiment No 9.

LIFT ELEVATOR SIMULATOR

AIM: To study Lift Elevator Simulator.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to "LIFT ELEVATOR SIMULATOR" section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python lift_rasp.py**

9) If you want to see program enter the command >> **sudo nano lift_rasp.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that when you press the switch **Ground** the **zero** will display on 7-segment display and green led is on. And when you press the switch **Top** the **one** will display on 2nd display and yellow led is high, then after some delay **two** display on 3rd display and yellow led is high. Again after some delay **three** displays on 4th display and green led is high.

12) The above procedure is same for 2nd floor and 1st floor.

NOTE- Connect jumper to vcc or ground as per requirement for pull up or pull down the switches.

Experiment No 10.

INTERFACING LM-35 WITH RASPBERRY PI

AIM: Interface LM-35 to Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, LM-35(Temperature sensor).

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to "ADC-DAC-TMP-IR" section.

5) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS.**

6) Now we are into the folder PYTHON PROGRAMMS

7) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

8) To run this specific program enter the command >> **python LM35.py**

9) If you want to see program enter the command >> **sudo nano LM35.py**

10) To exit from the program press **Ctrl-Z.**

11) You will observe that when resistor is heated LM-35 sensor sense the temperature and the output will shows on serial monitor.

12) Also you can see the output on LED's by varying the pot.

NOTE- the LM-35 sensor gives the Analog output. So we use the PC8951 ADC IC to convert Analog output into Digital. You can also use a POT as a analog input and shows the output on led's for study of ADC with raspberry pi.

CONFIGURATION

- The I2C peripheral is not turned on by default. To enable it, do the following.

1. Run `sudo raspi-config`.
2. Use the down arrow to select `9 Advanced Options`
3. Arrow down to `A7 I2C`.
4. Select `yes` when it asks you to enable I2C,
5. Also select `yes` when it asks about automatically loading the kernel module.
6. Use the right arrow to select the `<Finish>` button.
7. Select `yes` when it asks to reboot.

- Now install the i2c-tools package by:

sudo apt-get install i2c-tools

- If you get a 404 error do an update first:

sudo apt-get update

- Then run the install the i2c-tools again.

Note : The installation could take a few minutes to do, depend on how busy the server is.

- After the reboot test to see any device connected by:

sudo i2cdetect -y 1

Experiment No 11.

INTERFACING ZIGBEE WITH RASPBERRY PI

AIM: Zigbee interface to Raspberry pi.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply, LM-35(Temperature sensor).

PROCEDURE:

1) Connect Raspberry pi board.

2) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

3) Switch ON the power supply.

4) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to “ZIGBEE” section.

5) Configure ZigBee modules first as transmitter and second as a receiver. Configuration procedure will show the link given below. If once configured there is no need to configure again and again.

6) For this experiment we need two Raspberry Pi board.

7) Open the Terminal window of Raspberry pi and enter the command >> **cd /home/pi/PYTHON PROGRAMMS**.

8) Now we are into the folder PYTHON PROGRAMMS.

9) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

10) **I**) First load (**Receiver**) program to Onboard Raspberry Pi enter command >> **python recexbee.py**
II) Then load (**transmitter**)program in other Raspberry Pi enter command >> **python transxbee.py**

11) If you want to see program enter the command >> **sudo nano xbee.py**

12) To exit from the program press **Ctrl-Z.**

13) You will observe that whatever data will transmitted by transmitter will received on terminal window through receiver.

14)

NOTE-

- **Here we need two Raspberry Pi if in case two Raspberry Pi are not available then you can also use another microcontroller board e.g. Arduino**

- **We need 2 ZigBee modules for transmit and receive data. And 1st we have to configure those modules as a transmitter and receiver using XCTU software. For ZigBee configuration refer**
  http://www.libelium.com/development/waspmote/documentation/x-ctu-tutorial/
  https://alselectro.wordpress.com/category/xbee-radios/

Experiment No 12.

Interfacing of LM35 Temperature sensor with RASPBERRY PI and computing data using things board.

REQUIREMENTS:- Raspberry pi board, 40 pin FRC cable, 26 pin FRC cable, 5v,2A power supply.

PROCEDURE:

13) Connect Raspberry pi board.

14) Connect the HDMI screen, mouse and keyboard to Raspberry pi.

15) Switch ON the power supply.

16) Connect 40 pin FRC cable from Raspberry pi to the 40 pin connector which is placed on the board. And connect 26 pin FRC cable from the board to "LIFT ELEVATOR SIMULATOR" section.

17) Open the Terminal window of Raspberry pi and enterthe command >>
cd /home/pi/PYTHON PROGRAMS.

18) Now we are into the folder PYTHON PROGRAMMS

19) To see the list of files into PYTHON PROGRAMMS folder enter command >> **ls**

20) To run this specific program enter the command >> **python LM35_IOT.py**

21) If you want to see program enter the command >> **sudo nano LM35_IOT.py**

22) To exit from the program press **Ctrl-Z.**

23) You will observe the respective temperature sensed by LM35 temperature sensor on python window and the same data can be observed on things board which provides you the perfect IOT environment for your projects.

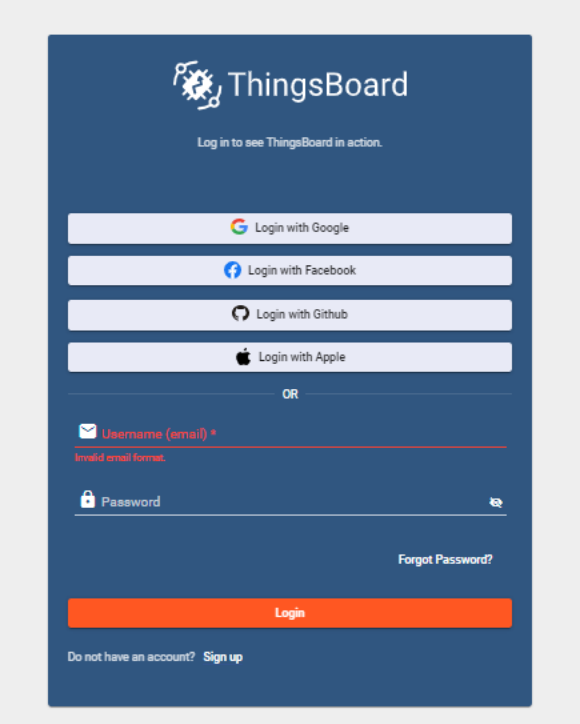
NOTE: Please note we are using evaluation version of things board here,which is free of cost for demo.

5. How to see your data on things board:

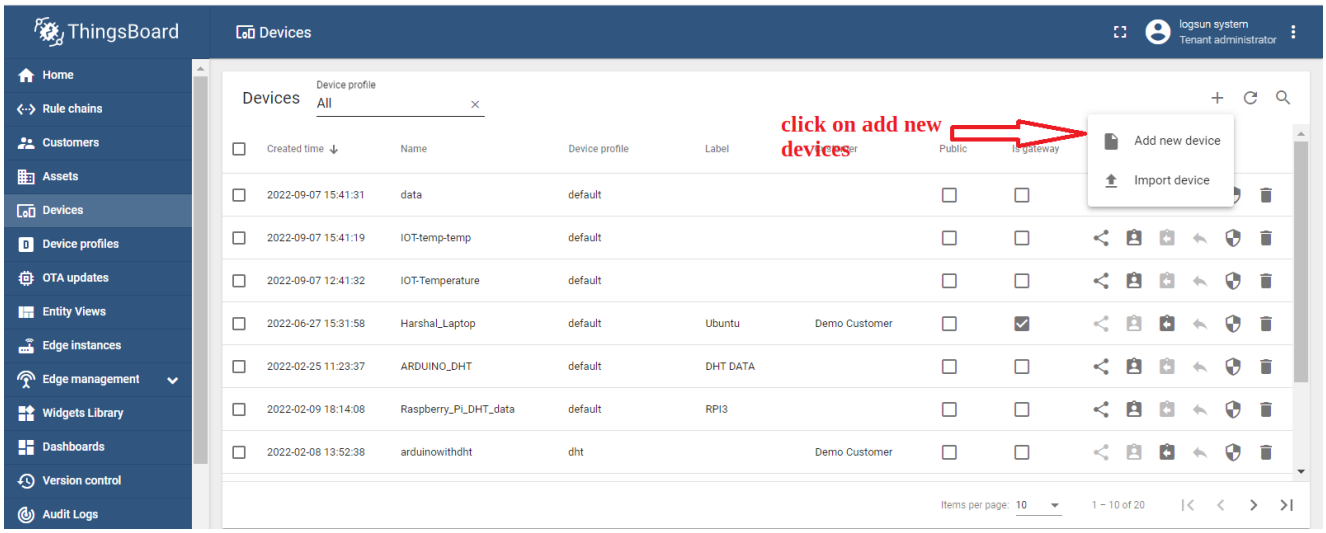
ThingsBoard is an open-source IoT platform for data collection, processing, visualization, and device management.

1. Go to https://demo.thingsboard.io/home

2. Sinup or login using gmail

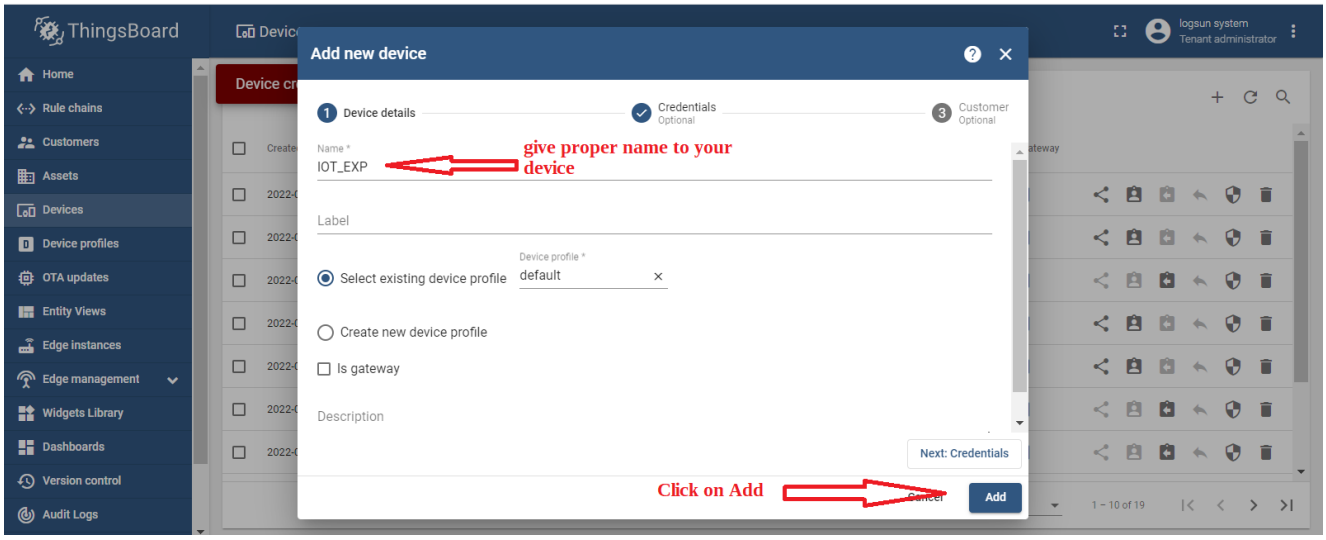


3. Ater sinup or login you will able to see following window.

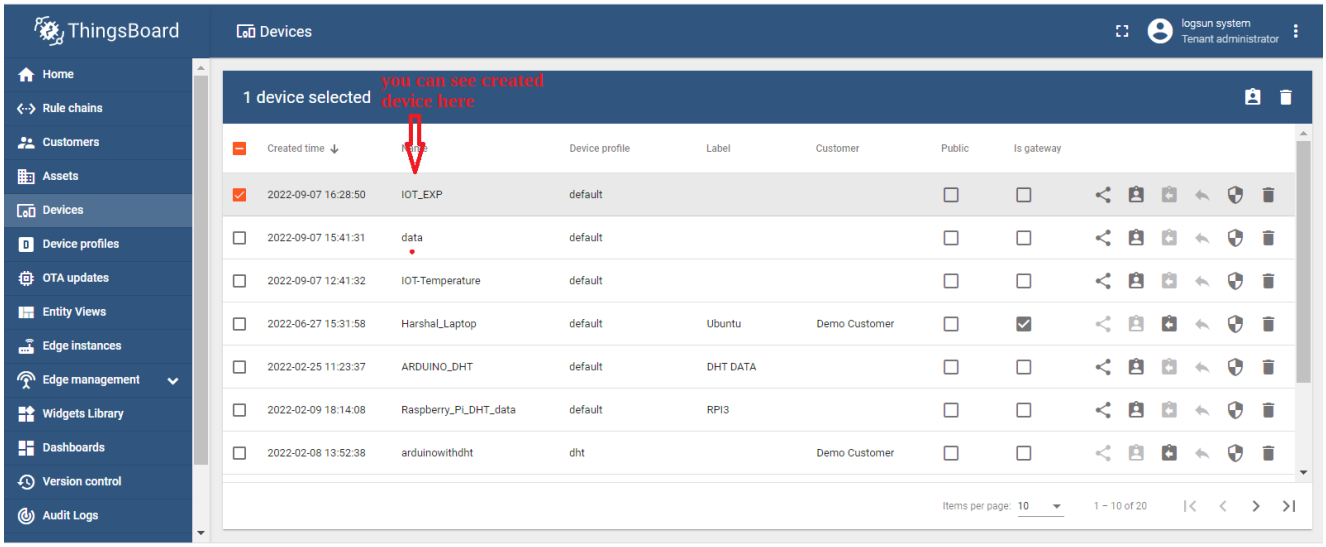


Here you can see some devices are already added for reference ,to add your device which you want to monitor through things board.
Click on “+ “sign =>click on Add new device

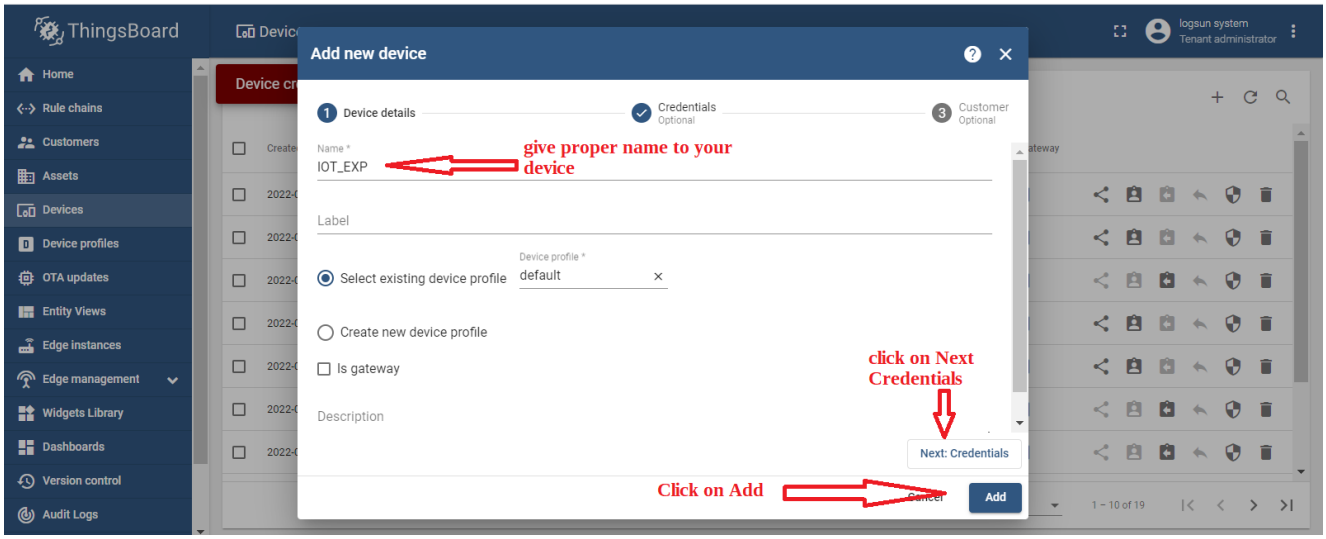
following window will appear



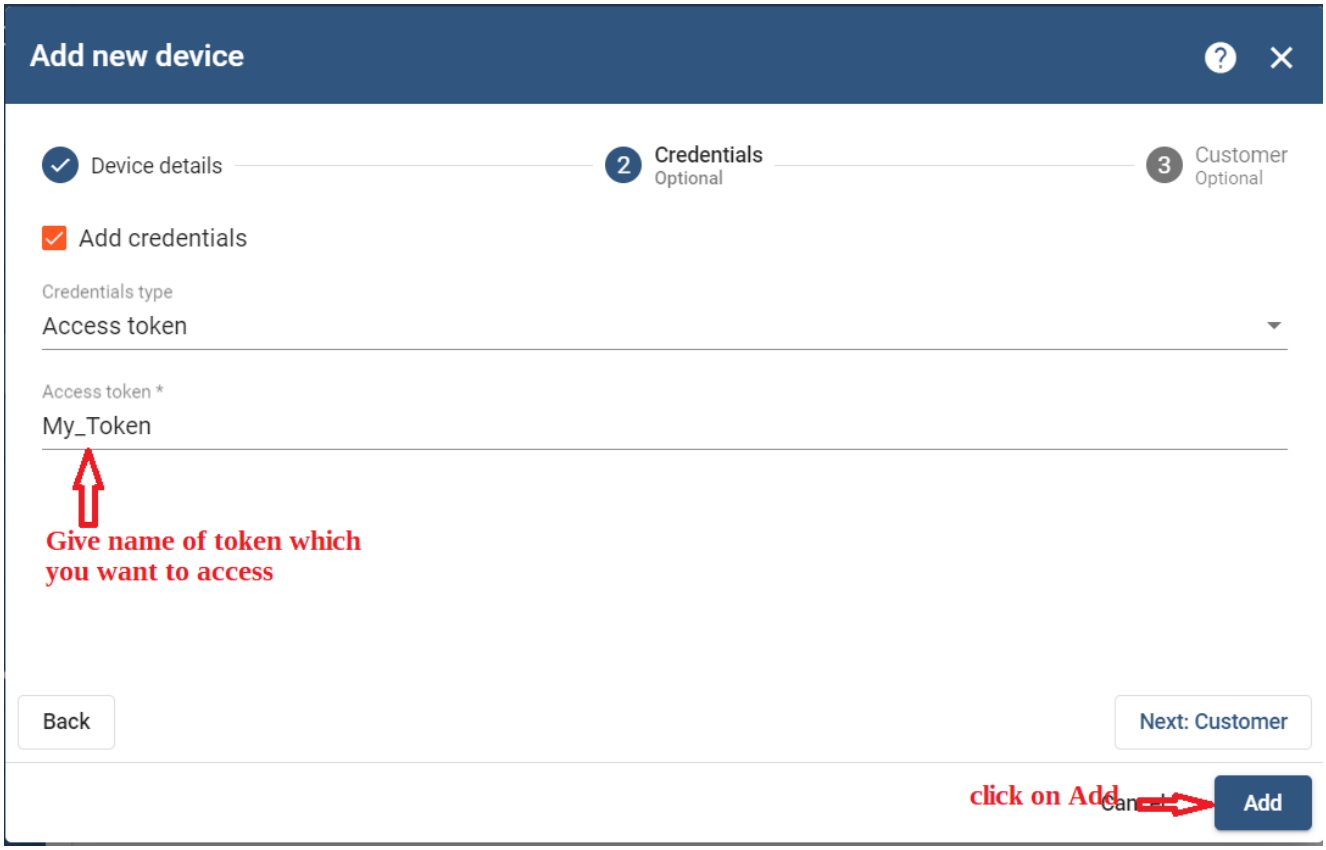
Give proper Name to your device, click on “Add’
your added device



if you want to access token created before then follow the below steps



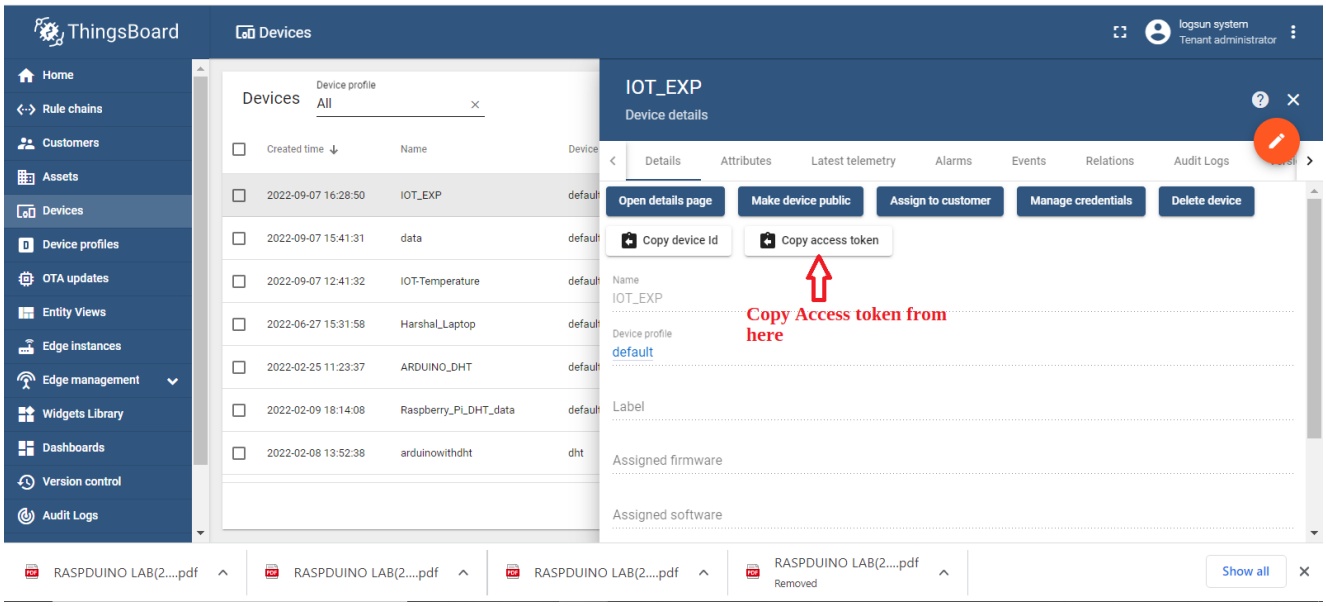
then following window will appear.



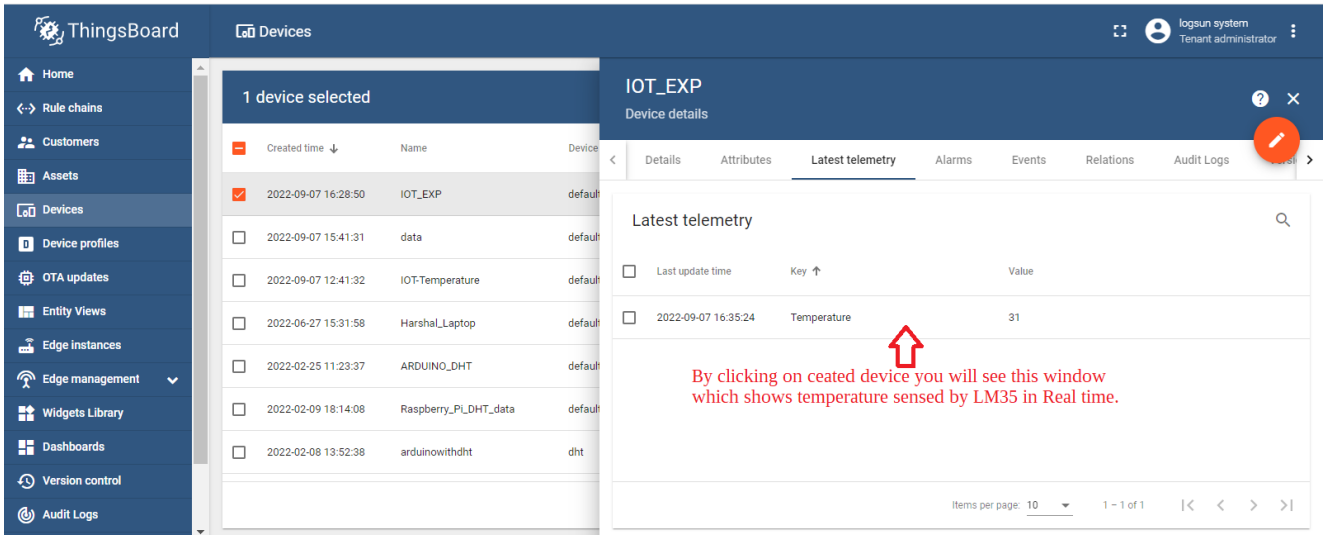
```python
import paho.mqtt.client as mqtt
import json
import smbus
import time

address = 0x48                # To store data of LM35
A0 = 0x40                     # To read LM35
bus = smbus.SMBus(1)

THINGSBOARD_HOST = "demo.thingsboard.io"
ACCESS_TOKEN = "My_Token"

INTERVAL = 2

TempData = {'Temperature':0}
next_reading = time.time()

client = mqtt.Client()

client.username_pw_set(ACCESS_TOKEN)

client.connect(THINGSBOARD_HOST, 1883, 60)
client.loop_start()

try:
    while True:
        bus.write_byte(address, A0)
        value = bus.read_byte(address)
        temperatureData = round(value, 2)
        print("Temperature is:-", str(value), "Degrees C")
```

Paste here access token copied from things board

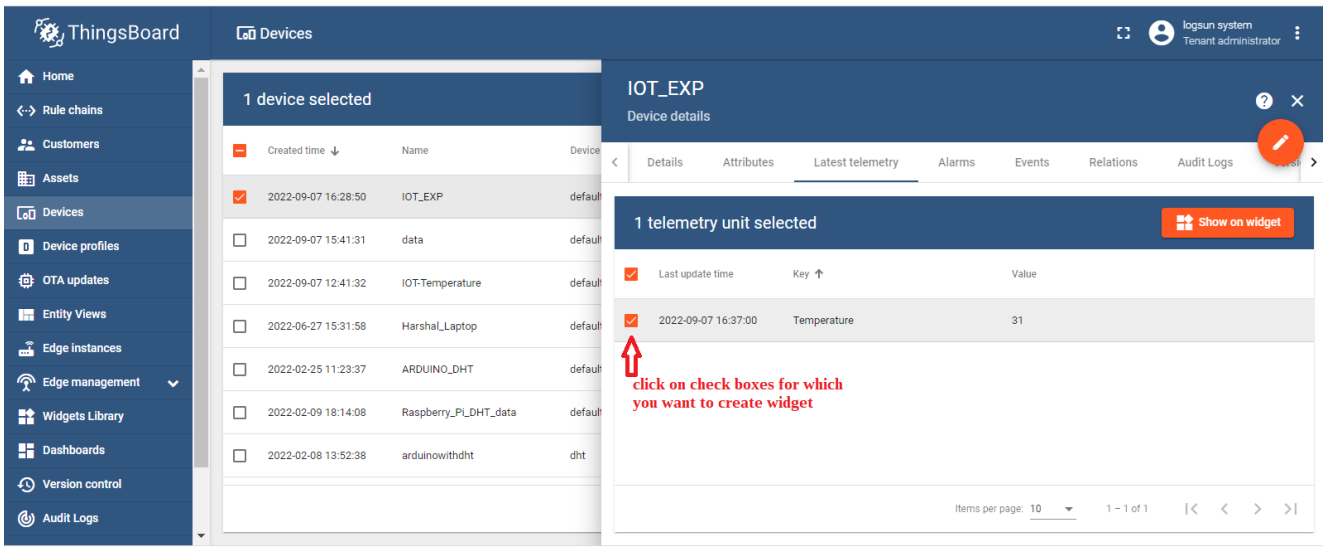
Now copy the token form copy access token bar,and paste this token into your program given location in above image.
And run your code.



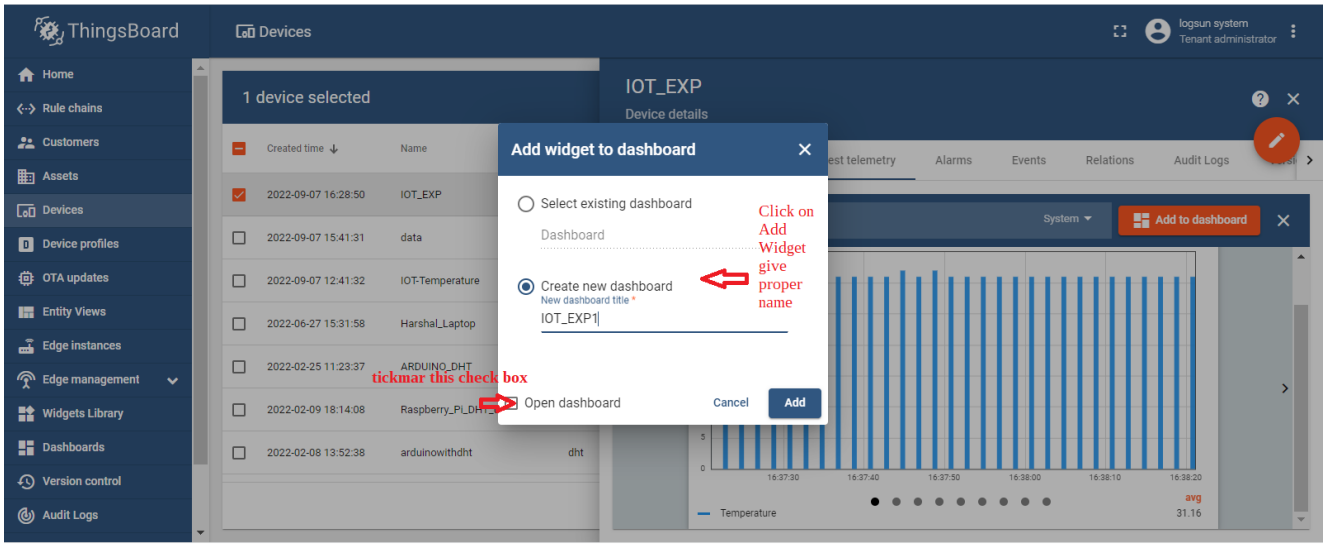
4. click on added device, you will see following window.
Here click on **“Latest telemetry”** here you can see computed data in Real time.
Make sure that program is in running mode with connected connected hardware,(here it is LM35 Temperature) while checking your data here.



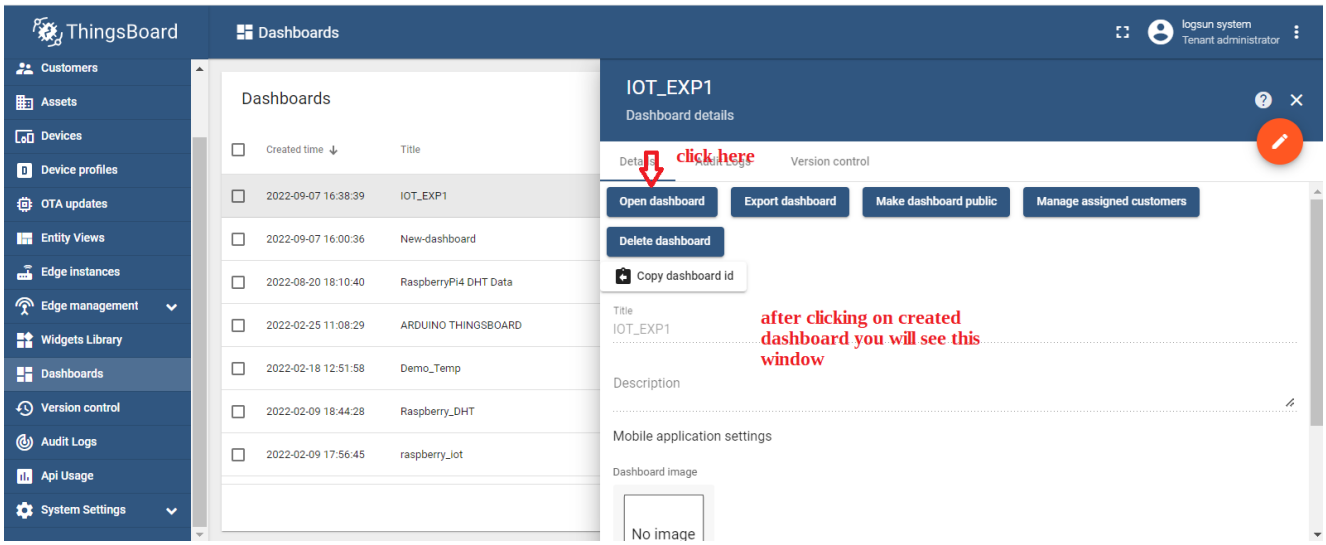
4. To see the same data on graphical format, create widget.
Select the device for which you want to create widget, and click on show on Widget symbol at Top right corner

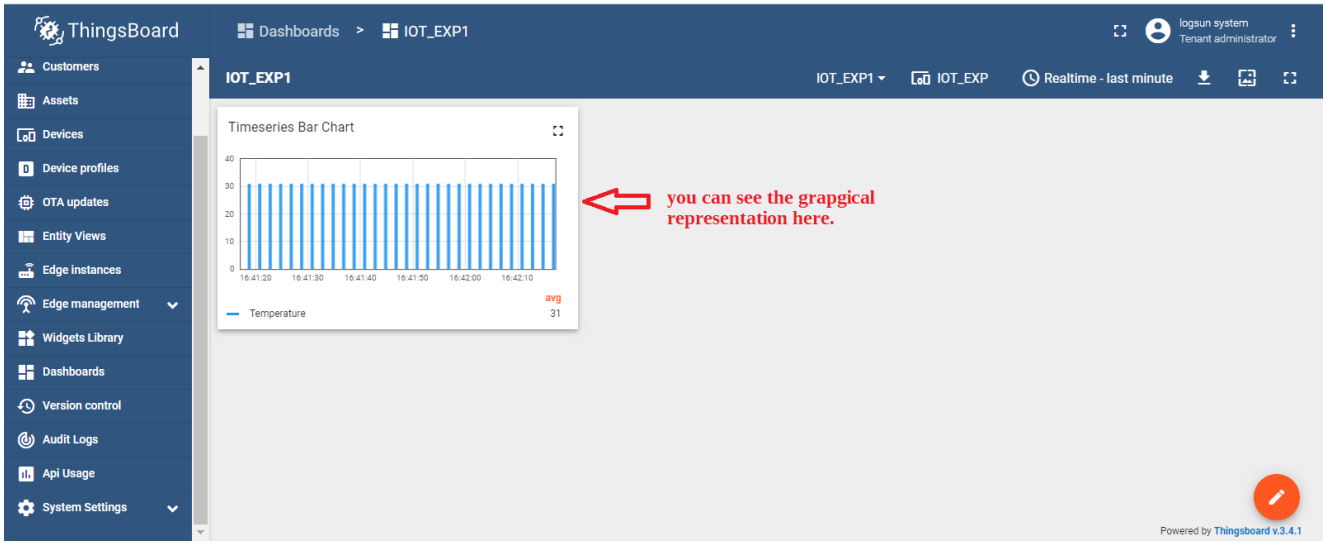


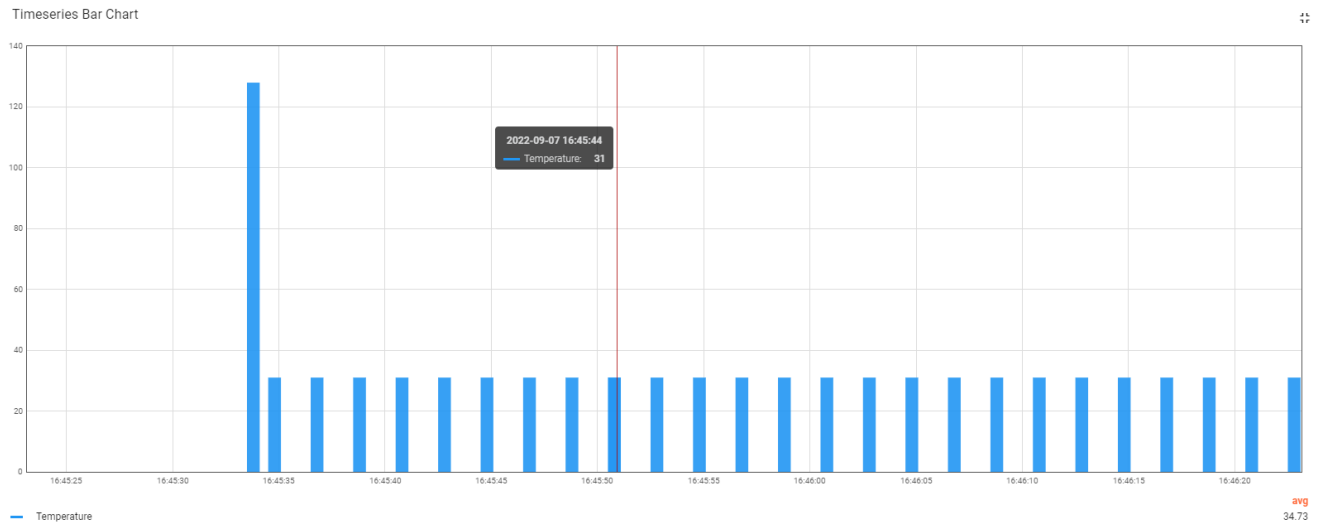
following window with graphical representation will appear here,
click on Add to Dashboard icon shown at top right corner



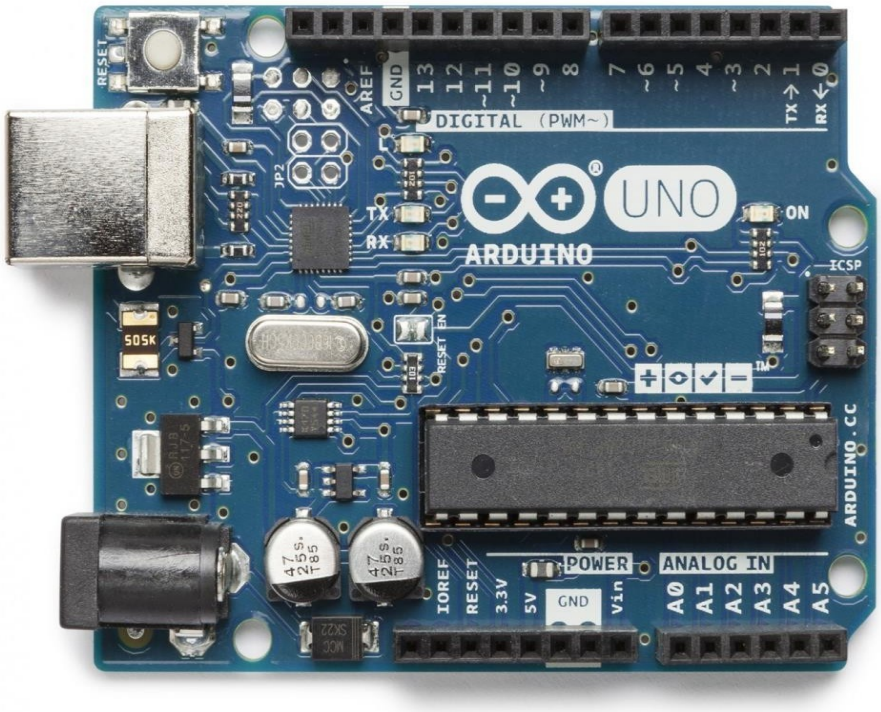
5. after adding it to dashboard
go to=> dashboard => open dashboard=> select the Widget added,you will able to see the following window.




ThingsBoard
Dashboards > IOT_EXP1
logsun system
Tenant administrator
Customers
Assets
Devices
Device profiles
OTA updates
Entity Views
Edge instances
Edge management
Widgets Library
Dashboards
Version control
Audit Logs
Api Usage
System Settings
IOT_EXP1
IOT_EXP1
IOT_EXP
Realtime - last minute
Timeseries Bar Chart
40
30
20
10
0
16:41:20
16:41:30
16:41:40
16:41:50
16:42:00
16:42:10
avg
Temperature
31
you can see the grapgical representation here.
Powered by Thingsboard v.3.4.1


Timeseries Bar Chart
140
120
100
80
60
40
20
0
2022-09-07 16:45:44
Temperature: 31
16:45:25
16:45:30
16:45:35
16:45:40
16:45:45
16:45:50
16:45:55
16:46:00
16:46:05
16:46:10
16:46:15
16:46:20
avg
Temperature
34.73

6. INTRODUCTION TO ARDUINO UNO



Arduino uno

The **Arduino Uno** is a microcontroller board based on the ATmega328. Arduino is an open-source, prototyping platform and its simplicity makes it ideal for hobbyists to use as well as professionals. The Arduino Uno has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, 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 a AC-to-DC adapter or battery to get started.

Power :-

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically. External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector. The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

VIN-

The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.

5V-

This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.

3V3-

A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND- Ground pins.

Memory-

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output-

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX)-

Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3-

These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.

PWM-

3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.

SPI-

10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.

LED: 13-

There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analog Reference() function. Additionally, some pins have specialized functionality:

TWI: A4 or SDA pin and A5 or SCL pin- Support TWI communication using the Wire library.

There are a couple of other pins on the board:

AREF-

Reference voltage for the analog inputs. Used with analogReference().

Reset-

Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

SPECIFICATION

Microcontroller- ATmega328

Operating Voltage- 5V

Input Voltage (recommended)- 7-12V

Input Voltage (limits)- 6-20V.

Digital I/O Pins- 14 (of which 6 provide PWM output)

Analog Input Pins- 6

DC Current per I/O Pin- 40 mA .

DC Current for 3.3V Pin- 50 mA

Flash Memory- 32KB (ATmega328) of which 0.5 KB used by boot loader

SRAM- 2 KB (ATmega328)

EEPROM- 1 KB (ATmega328)

Clock Speed- 16 MHz

SETTING FOR Arduino Uno :

INSTALL Arduino 1.8.3 -

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, Linux. This software can be used with any Arduino board.

1) DOWNLOAD-

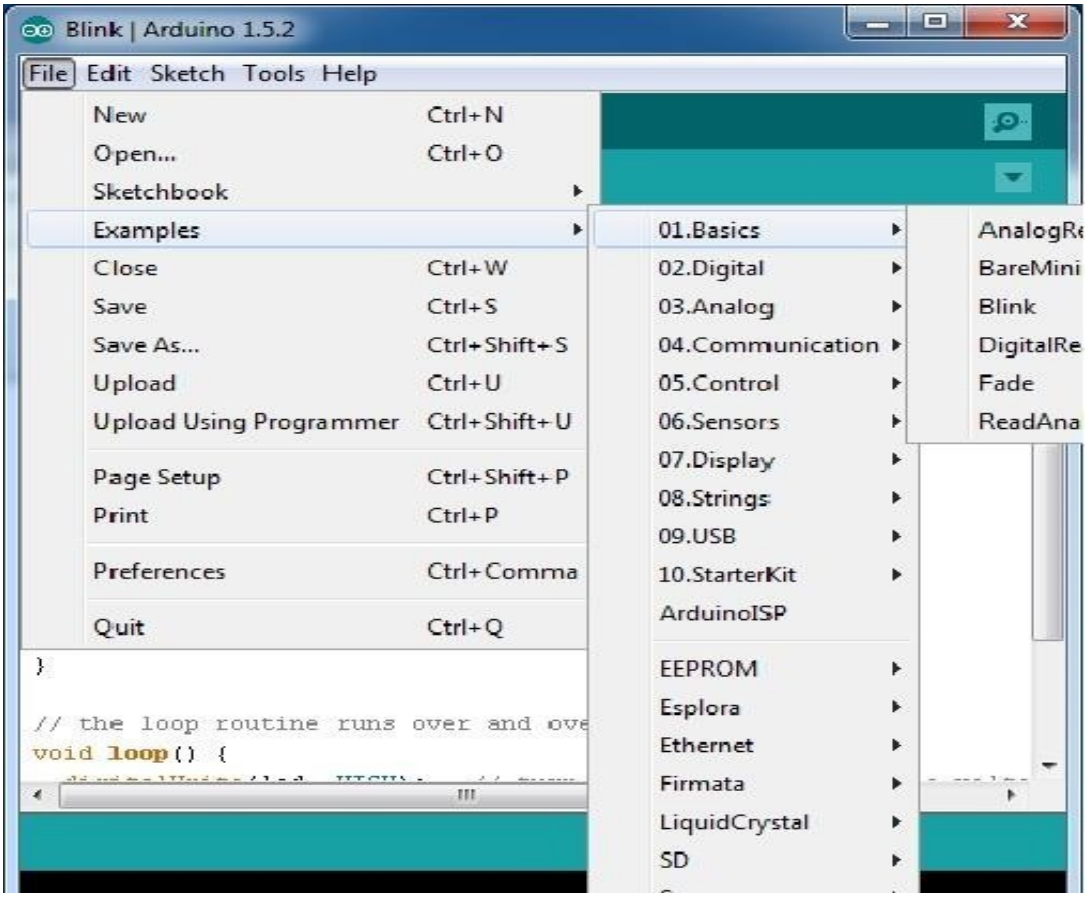
Download the software from-

https://www.arduino.cc/en/Main/Software

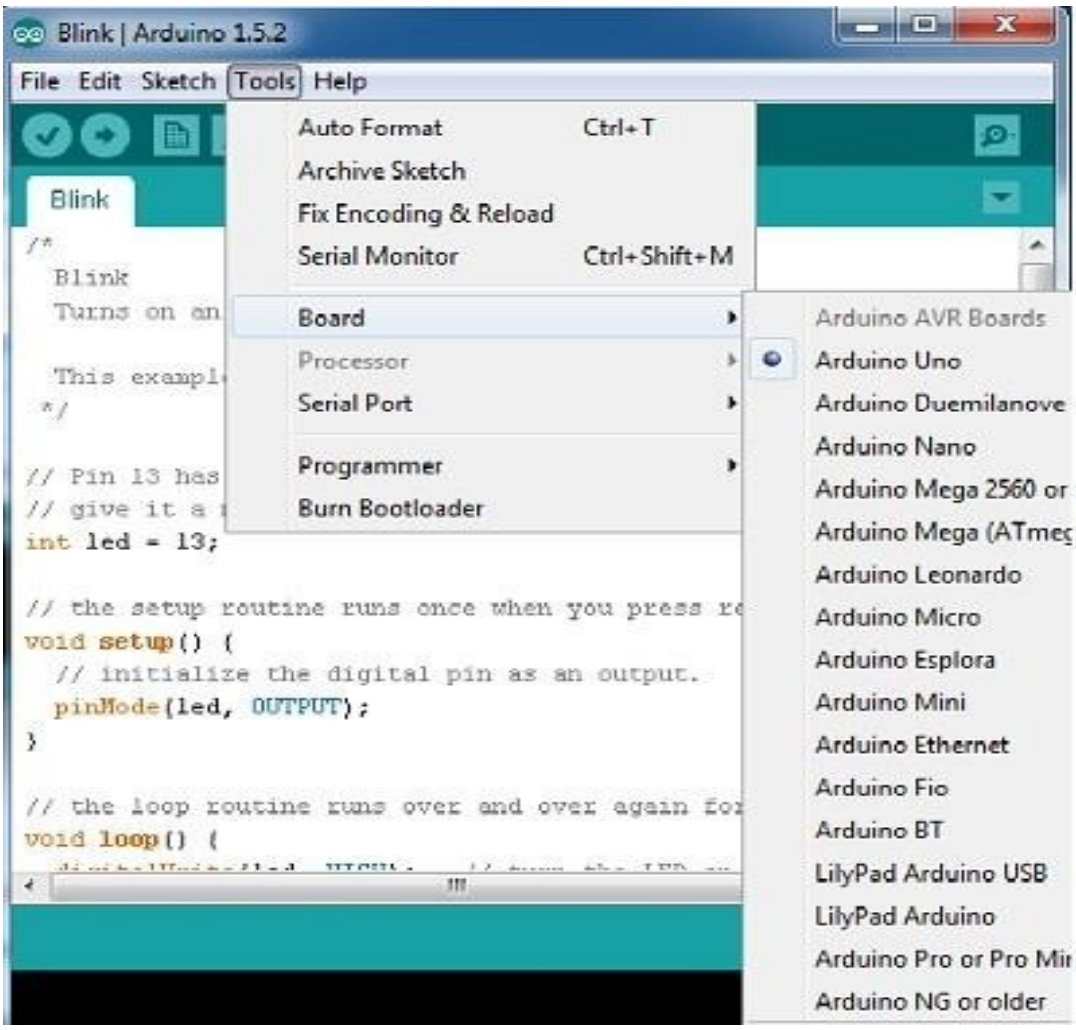
Launch and Blink-

After following the appropriate steps for your software install, we are now ready to test your first program with your Arduino board!

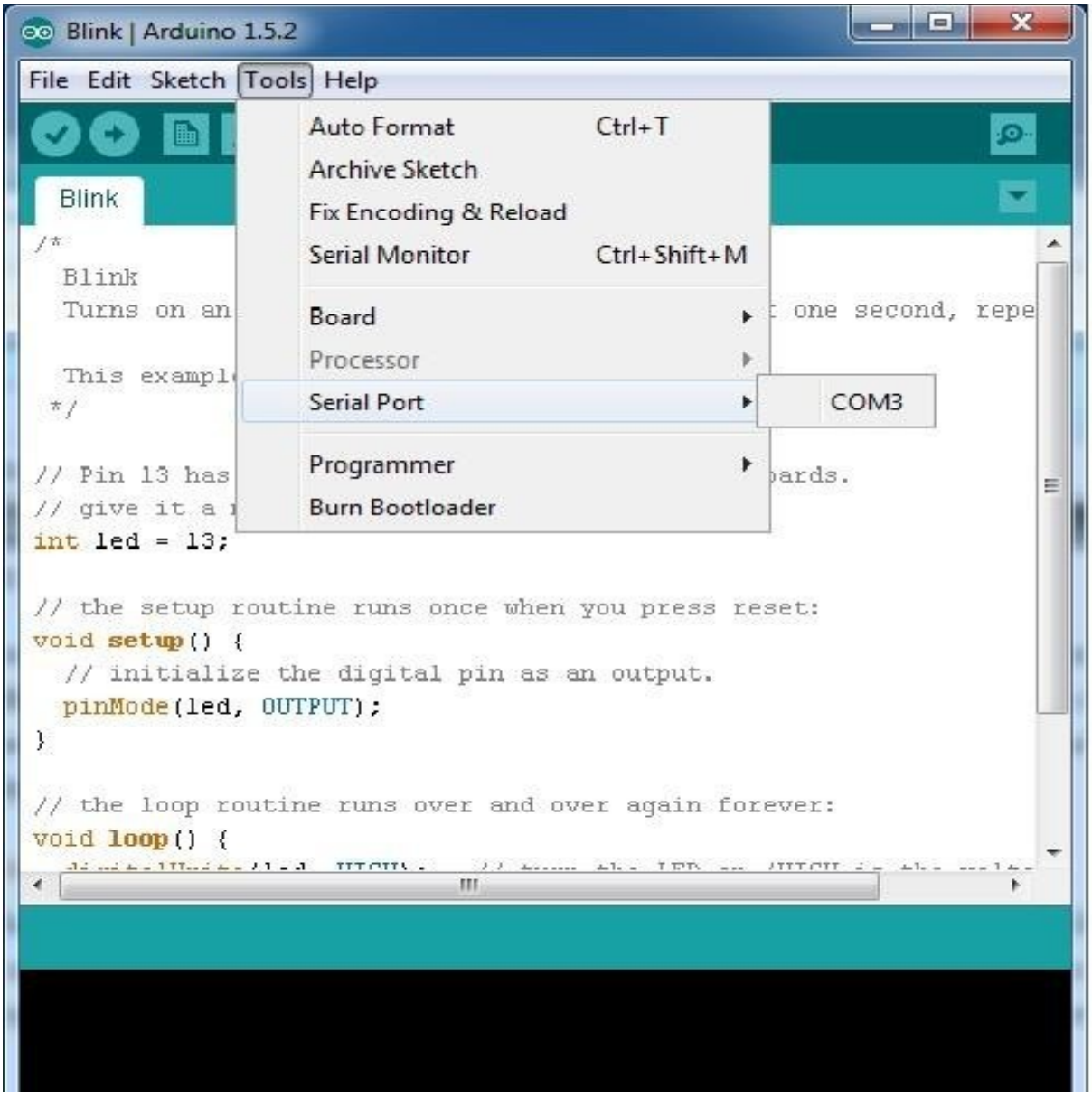
- Launch the Arduino application
- If you disconnected your board, plug it back in
- Open the Blink example sketch by going to: File > Examples > 1.Basics > Blink



- Select the type of Arduino board you're using: Tools > Board > your board type

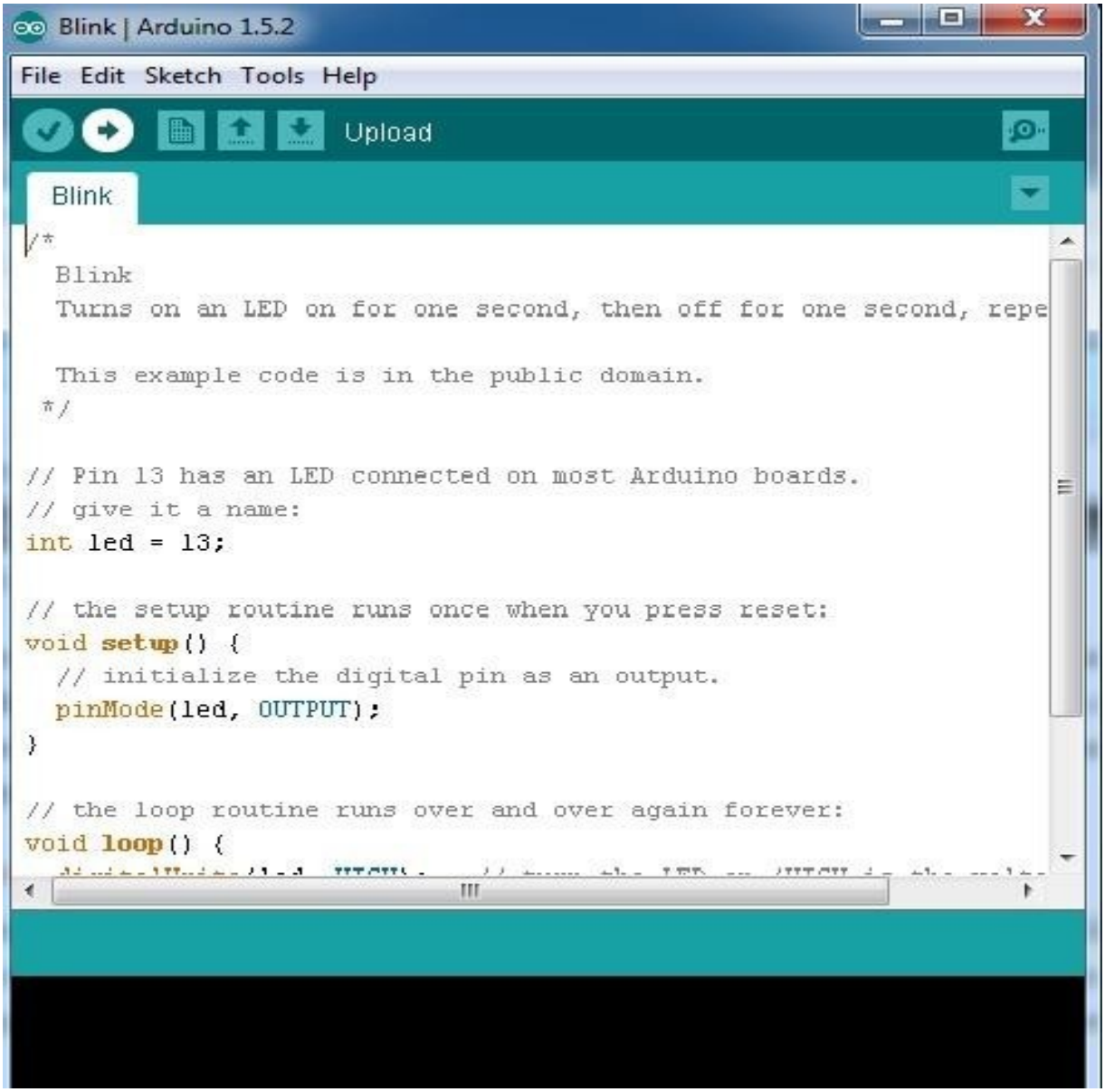


- Select the serial/COM port that your Arduino is attached to: Tools > Port > COMxx



- If you’re not sure which serial device is your Arduino, take a look at the available ports, then unplug your Arduino and look again. The one that disappeared is your Arduino.

- With your Arduino board connected, and the Blink sketch open, press the ‘Upload’ button



- After a second, you should see some LEDs flashing on your Arduino, followed by the message ‘Done Uploading’ in the status bar of the Blink sketch.
- If everything worked, the onboard LED on your Arduino should now be blinking! You just programmed your first Arduino!

EXPERIMENT No . 13

LED Logic

AIM: To study the interfacing of LED and SWITCH to the Arduino.

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v, 2A power supply and USB cable.

PROCEDURE:

12) Open the Arduino IDE. Connect USB cable to Arduino Uno.

13) Select Arduino Uno and COM port from **TOOL** section.

14) Now go to file ► open particular "**.ino**" file of led logic program.

15) Compile the program and then upload it.

16) After uploading connect 26 pin FRC cable from the board to "XBEE-LED-SWITCHES" section.

17) You will observe that led's will **ON** when you **ON** the switches and **OFF** when you **OFF** the switches.

Note: FRC port connection must be proper to run the program.

EXPERIMENT No 14.

LCD DISPLAY

AIM: To study the interfacing of LCD(16*2) with the Arduino Uno.

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of LCD program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "LCD-KEYPAD" section.
6) You will observe that it displays "**LOGSUN SYSTEMS**" on LCD display.

THEORY:

Liquid crystal display:

LCD can be connected to the Raspberry pi through the GPIO pins. LCD is connected in the 4-bit mode or 8-bit mode. And the standard subroutine is given with the development board. So that the application can be easily demonstrated and also for further implementation the subroutine can be easily embedded for which one has to do very few changes. Wide range of instruction functions: Clear displays, cursor home, display ON/OFF, cursor ON/OFF, cursor shift, display shift.

4-bit mode

RS, RW, EN are 1&2 are short.

8-bit mode

RS, RW, EN are 2&3 are short

EXPERIMENT No 15.

LCD WITH KEYPAD INTERFACING

AIM: To study the interfacing of LCD and Keypad with Arduino.

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v, 2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of LCD Keypad program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "LCD-KEYPAD" section.
6) You will observe that when you press a key (0-F) it will show on LCD display.

EXPERIMENT No 16.
Traffic Light Signal

AIM: To study the interfacing of LED to make traffic light signal with Arduino

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of traffic light signal program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "TRAFFIC LIGHT" section.

12) You will observe that red led's of 1st and 3rd stage is ON for 10 sec. Green led's of both opposite sides are on for 5 sec. Then yellow led's 5 times blinks. And red led's OFF. And same action happens for section 2nd and 4th.

EXPERIMENT No 17.

INTERFACING IR SENSOR WITH ARDUINO

AIM: To study the interfacing of IR sensor with Arduino.

REQUIREMENTS:- Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.

2) Select Arduino Uno and COM port from **TOOL** section.

3) Now go to file ⟶ open particular "**.ino**" file of IR Sensor program.

4) Compile the program and then upload it.

5) After uploading connect 26 pin FRC cable from the board to "ADC-DAC-TEMP-IR" section.

12) You will observe that when obstacle is detected that time "IR is detected" prints on the serial monitor, relay will activated and red led is ON.

Experiment No 18.

INTERFACING DC MOTOR TO ARDUINO

AIM: To study the interfacing of dc motor to Arduino.

REQUIREMENTS:- Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable, Dc motor, 12v dc external power supply.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of DC motor program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "DC-STEPPER MOTOR" section.
6) You will observe that when you press the switch **START** motor will start rotating in forward direction. And when you press the switch **REV** motor will rotate in opposite or reverse direction
7) When you press switch **INC** the speed of motor increase slowly, as you press the same switch again. And when you press switch **DEC** the speed of motor decrease slowly, as you press same switch again.
8) When you press switch **STOP** the motor will stop rotating.

NOTE:- Connect both jumpers(JP2 & JP3) on DC side.

Experiment No 19.

INTERFACING STEPPER MOTOR TO ARDUINO

AIM: To study the interfacing of stepper motor to Arduino.

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable, Dc motor,12v dc external power supply.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of Stepper motor program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "DC-STEPPER MOTOR" section.
6) You will observe that when you press the switch **START** motor will start rotating in forward direction. And when you press the switch **REV** motor will rotate in opposite or reverse direction
7) When you press switch **INC** the speed of motor increase slowly, as you press the same switch again. And when you press switch **DEC** the speed of motor decrease slowly, as you press same switch again.
8) When you press switch **STOP** the motor will stop rotating.

NOTE:- Connect both jumpers(JP2 &JP3) on STP side.

Experiment No 20.

INTERFACING RELAY WITH ARDUINO

AIM: To study interfacing of relay with Arduino

REQUIREMENTS:- Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of relay program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "ADC-DAC-TEMP-IR" section.

12) You will observe that relay is ON for 5sec and OFF for 5 sec.

Experiment No 21.

LIFT ELEVATOR SIMULATOR

AIM: To study Lift Elevator Simulator.

REQUIREMENTS:- Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable.

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of Lift Elevator Simulator program.
4) Compile the program and then upload it.
5) After uploading connect 26 pin FRC cable from the board to "LIFT ELEVATOR SIMULATOR" section.

1) You will observe that when you press the switch **Ground** the **zero** will display on 7-segment display and green led is on. And when you press the switch **Top** the **one** will display on 2nd display and yellow led is high, then after some delay **two** display on 3rd display and yellow led is high. Then again after some delay **three** display on 4th display and green led is high.
2) The above procedure is same for 2nd floor and 1st floor.

NOTE- Connect jumper to VCC or GND as per requirement for pull up or pull down the switches.

Experiment No 22.

INTERFACING LM-35 WITH ARDUINO

AIM: Interface LM-35 to Arduino.

REQUIREMENTS:- Arduino,26 pin FRC cable, 5v,2A power supply, LM-35(Temperature sensor).

PROCEDURE:

1) Open the Arduino IDE. Connect USB cable to Arduino Uno.
2) Select Arduino Uno and COM port from **TOOL** section.
3) Now go to file ⟶ open particular "**.ino**" file of LM-35 program.
4) Compile the program and then upload it.
5) After uploading keep JP8 jumper on LM-35 side.
6) Now connect 26 pin FRC cable from the board to "ADC-DAC-TEMP-IR" section.
7) You will observe that when resistor is heated LM-35 sense the temperature and the output will shows on serial monitor.
8) Also you can see the output on LED's by varying the pot.

NOTE- The LM-35 sensor gives the Analog output. So we use the ADC of arduino to convert Analog output into Digital. You can also use a POT as a analog input and shows the output on led's for study of ADC with Arduino.

Experiment No 23.

INTERFACING ZIGBEE WITH ARDUINO

AIM: ZigBee interface to Arduino.

REQUIREMENTS: Arduino, 26 pin FRC cable, 5v,2A power supply or USB cable ZigBee modules.

PROCEDURE:

14) Open the Arduino IDE. Connect USB cable to Arduino Uno.

15) Select Arduino Uno and COM port from **TOOL** section.

16) Now go to file→ open particular "**.ino**" file of Lift Elevator Simulator program.

17) Configure ZigBee modules first as transmitter and second as a receiver. Configuration procedure will show the link given below.

18) For this experiment we need two Arduino uno board.

19) Upload **receiver.ino** program to onboard Arduino and **transmitter.ino** to the other Arduino.

20) Now open serial monitor window in Arduino software on pc.

21) Connect 26 pin FRC cable from the board to "XBEE-LED-SWITCHES" section.

22) Whatever data will transmitted by transmitter will received on serial monitor through receiver.

NOTE- We need 2 ZigBee modules for transmit and receive data. And 1st we have to configure that modules as a transmitter and receiver using XCTU software.

ForZigBee configuration refer
http://www.libelium.com/development/waspmote/documentation/x-ctu-tutorial/
https://alselectro.wordpress.com/category/xbee-radios/

8. FRC Connector Details

A) 26 PIN FRC CONNECTOR DETAILS RASPBERRY PI (J7) :

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	SCL	2	SDA
3	GP27	4	GP22
5	GP16	6	GP10
7	GP12	8	GP20
9	GP7	10	GP21
11	GP8	12	GP26
13	GP25	14	GP19
15	GP24	15	GP13
17	GP23	18	GP6
19	TX	20	GP5
21	RX	22	GP11
23	GP18	24	GP9
25	GND	26	5V

B) 26 PIN FRC CONNECTOR DETAILS ARDUINO UNO (J8):

PIN NO.	DESCRIPTION	PIN NO.	DESCRIPTION
1	A5	2	A4
3	D10	4	D11
5	D9	6	D12
7	D8	8	D13
9	D7	10	A0
11	D6	12	A1
13	D5	14	A2
15	D4	15	A3
17	D3	18	
19	D1 TX	20	
21	D0 RX	22	
23	D2	24	
25	GND	26	5V