IoT Physical Devices and Raspberry Pi with Python

By,

Dr. Vidya Rao
Assistant Professor,
Dept of DSCA, MIT, MAHE

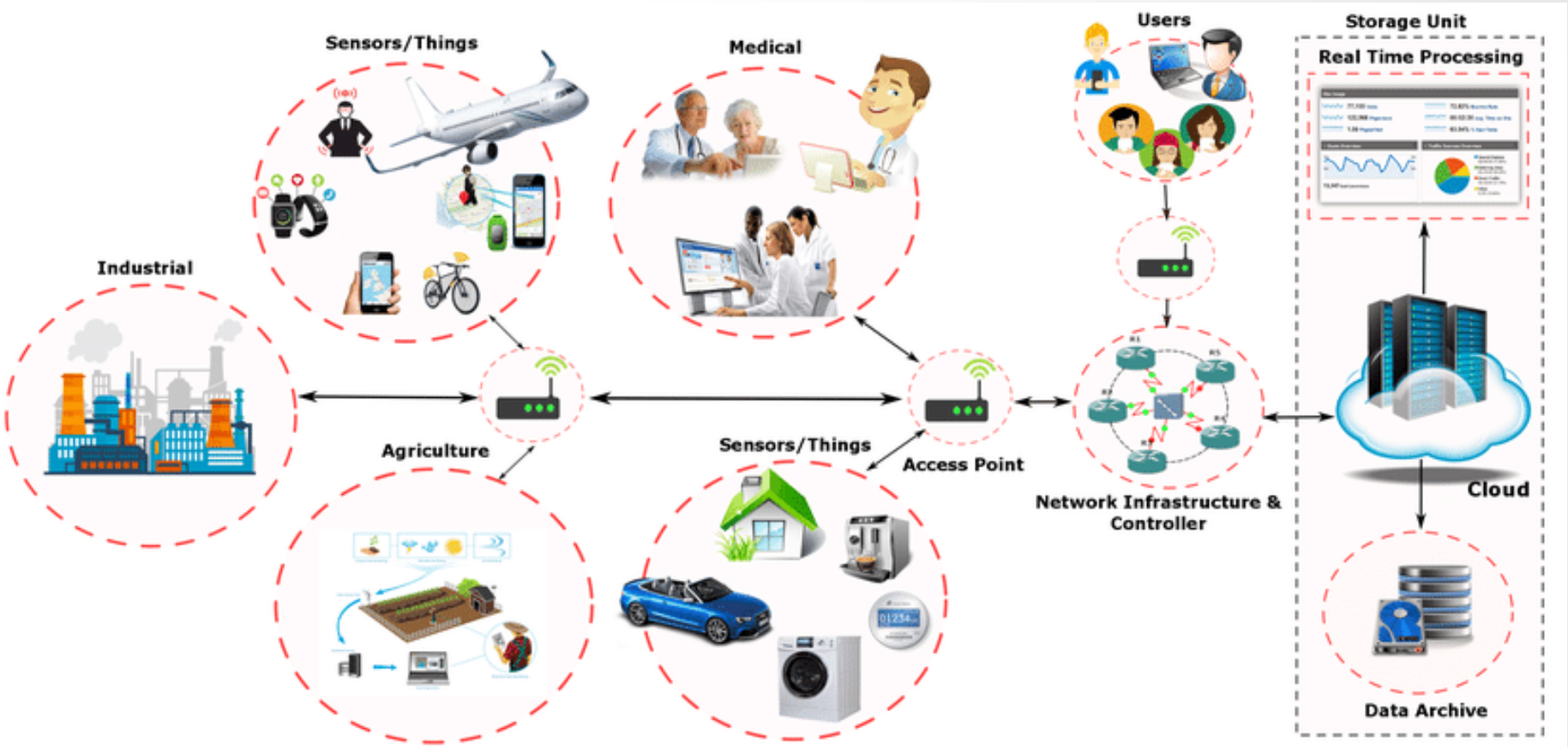
Outline

- Basic building blocks of an IoT Device
- Exemplary Device: Raspberry Pi
- Raspberry Pi interfaces
- Programming Raspberry Pi with Python
- Other IoT devices

What is an IoT Devices?

- A "Thing" in the Internet of Things (IoT) can be any object that has a unique identifier and which can send/receive data (including user data) over a network (e.g., smartphone, smart TV, computer, refrigerator, car, etc.).
- IoT devices are connected to the Internet and send information about themselves or about their surroundings (e.g. information sensed by the connected sensors) over a network (to other devices or servers/storage) or allow actuation upon the physical entities/environment around them remotely.

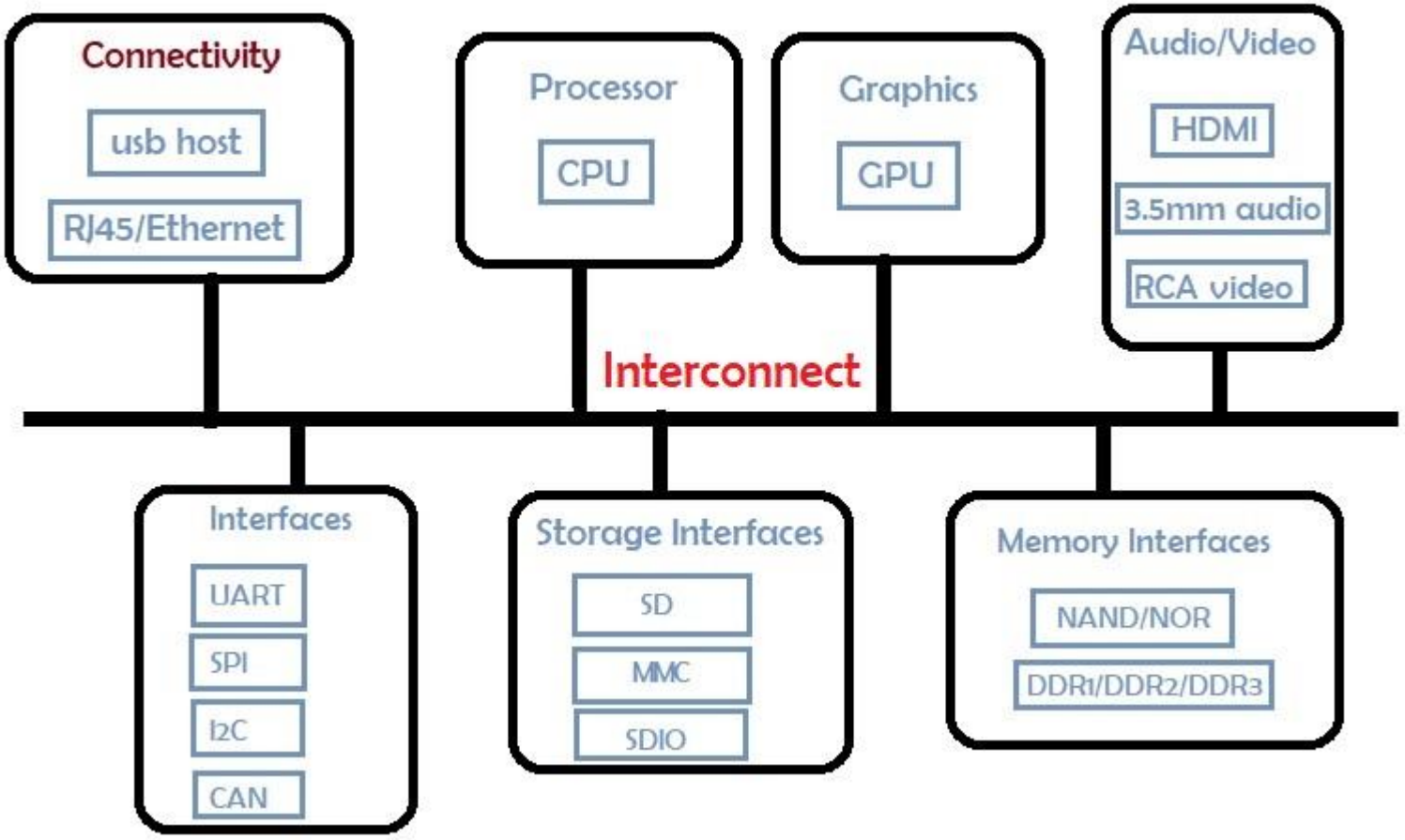
IoT Device Examples



Basic building blocks of an IoT Device

- **Sensing**
  - Sensors can be either on board the IoT device or attached to the device.
- **Actuation**
  - IoT devices can have various types of actuators attached that allow taking
  - actions upon the physical entities in the vicinity of the device.
- **Communication**
  - Communication modules are responsible for sending collected data to other devices or cloud-based servers/storage and receiving data from other devices and commands from remote applications.
- **Analysis & Processing**
  - Analysis and processing modules are responsible for making sense of the collected data.

Block diagram of an IoT Device

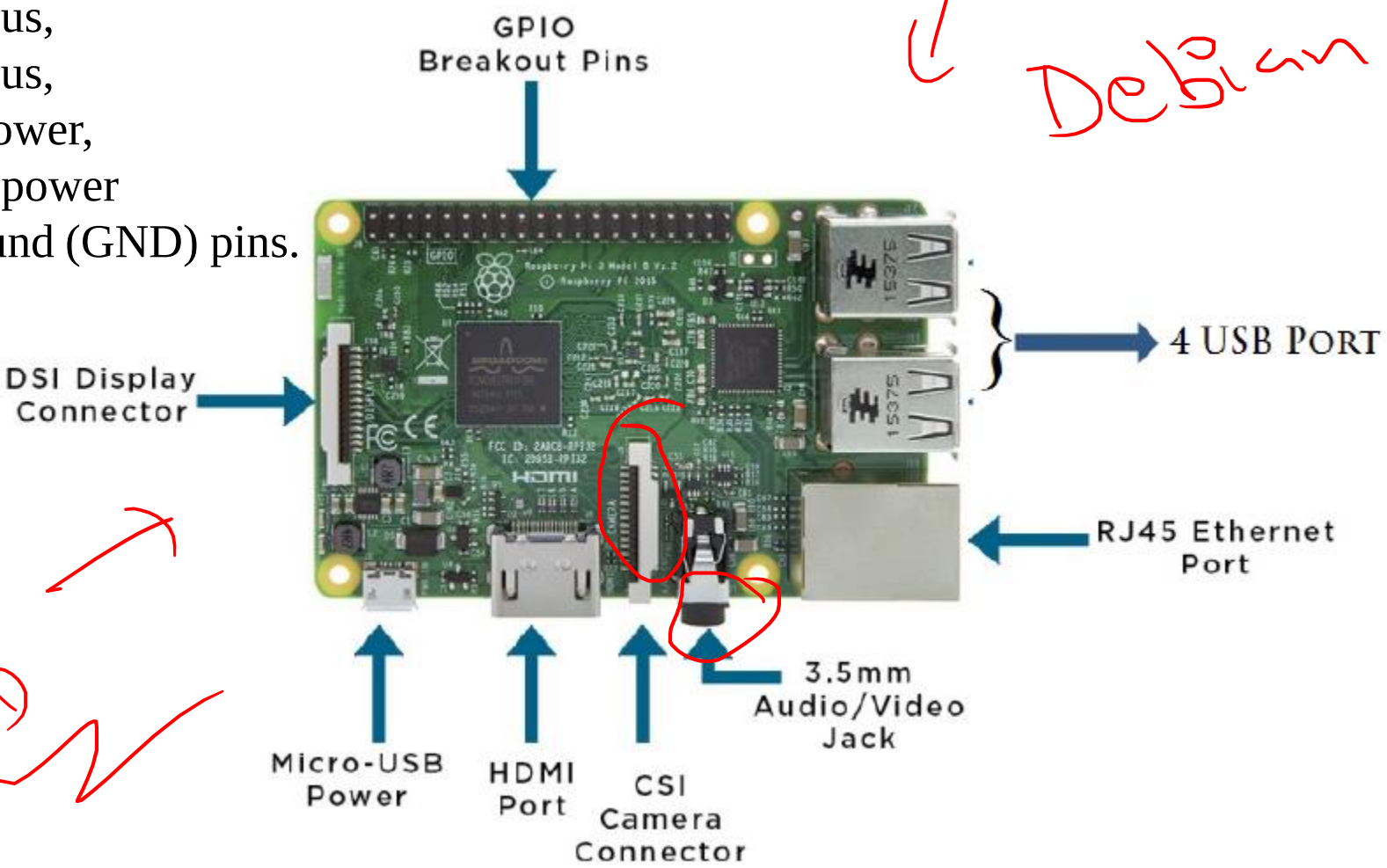


Exemplary Device: Raspberry Pi

- Raspberry Pi is a low-cost mini-computer with the physical size of a credit card.
- Raspberry Pi runs various flavors of Linux and can perform almost all tasks that a normal desktop computer can do.
- Raspberry Pi also allows interfacing sensors and actuators through the general-purpose I/O pins.
- Since Raspberry Pi runs Linux operating system, it supports Python "out of the box".

Raspberry Pi 3

- RPi3 has twenty four GPIO pins,
- one serial UART,
- two SPI bus,
- one I2C bus,
- two 5V power,
- two 3.3V power
- eight ground (GND) pins.



Raspberry Pi 3 (contd..)

- **RAM & processor:** RPi3 has 1GB RAM with 700 MHz Low power ARM1176JZ-F processor.
- **USB Ports:** It has four 2.0 USB ports thus providing all the required peripherals like keyboards, monitor and mouse to be connected at a time.
- **Ethernet Port:** It has one Ethernet port that can be connected with the LAN adapter else it can also use in-built WiFi adapter.
- **Display Serial Interface(DSI):** It is used to connect LCD screen to RPi3.
- **Camera Serial Interface(CSI):** It is used to connect camera to RPi3.
- **SD card slot:** A minimum 8GB storage SD is required to store the image of Linux OS that is used to run the RPi3.

Raspberry Pi 3 (contd..)

- **General Purpose Input-Output (GPIO) Pins:** These pins enable RPi3 to establish connections with various sensors and actuators. It contains true GPIO pins, I2C interface pins, SPI interface pins and one of serial Rx pin for receiving and a Tx pin for transmitting.
- **High-Definition Multimedia Interface (HDMI) output**: It helps RPi3 to provide both video and audio output. Also, RPi3 can be connected to PC monitor using HDMI cable adapter.
- **Composite video output:** This helps RPi3 to connect to composite video output with RCA jack that supports PAL and NTSC video input.
- **Composite video output**: This helps RPi3 to connect to composite video output with RCA jack that supports PAL and NTSC video input.
- **Audio output:** It has 3.5mm audio jack which can be connected to old RCA television to get better quality than the current monitors.

Raspberry Pi 3 (contd..)

- **Power Input:** A micro USB power input is used for connecting power that takes 5.1V at 2.5A.
- **Status LED:** RPi3 has five status LED with functionalities shown in the Table 1.
- Rest of RPi3 specifications are shown in the Table 2.

Status LED	Function
ACT	SD card access
PWR	3.3V Power
FDX	Full Duplex LAN
LNK	Link/Network activity
100	100 Mbit LAN connected

Table 1: Status information

Table 2: Specification

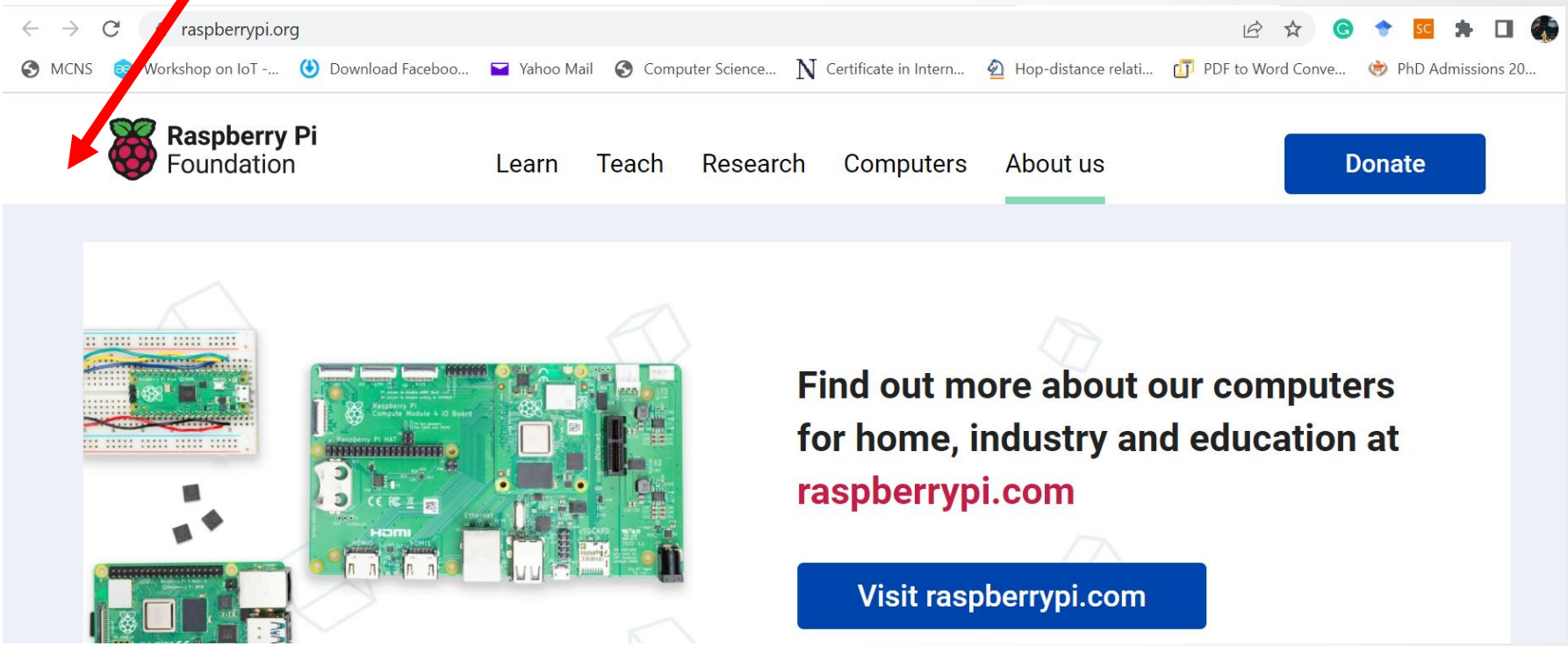
Board	Specification Pi-3	Architecture	ARMv8-A
SoC	Broadcom	CPU	1.2 GHz quad-core
Core frequency	250 MHz frequency	ARM	600 MHz
On-board network	Ethernet, WiFi 802.11 Bluetooth 4.1	Core OS	1.3062 V

Raspberry Pi 3 Pin Configuration

3.3V	1	2	5V
GPIO 2	3	4	5V
GPIO 3	5	6	Ground
GPIO 4	7	8	GPIO 14
Ground	9	10	GPIO 15
GPIO 17	11	12	GPIO 18
GPIO 27	13	14	Ground
GPIO 22	15	16	GPIO 23
3.3V	17	18	GPIO 24
GPIO 10	19	20	Ground
GPIO 9	21	22	GPIO 25
GPIO 11	23	24	GPIO 8
Ground	25	26	GPIO 7
---	27	28	---
GPIO 5	29	30	Ground
GPIO 6	31	32	GPIO 12
GPIO 13	33	34	Ground
GPIO 19	35	36	GPIO 16
GPIO 26	37	38	GPIO 20
Ground	39	40	GPIO 21

Raspberry Pi Foundations

- Raspberry Pi web page: https://www.raspberrypi.com/
- Documentation: https://www.raspberrypi.com/documentation/
- Foundation: https://www.raspberrypi.org/
- Feature courses: https://www.raspberrypi.org/courses/featured

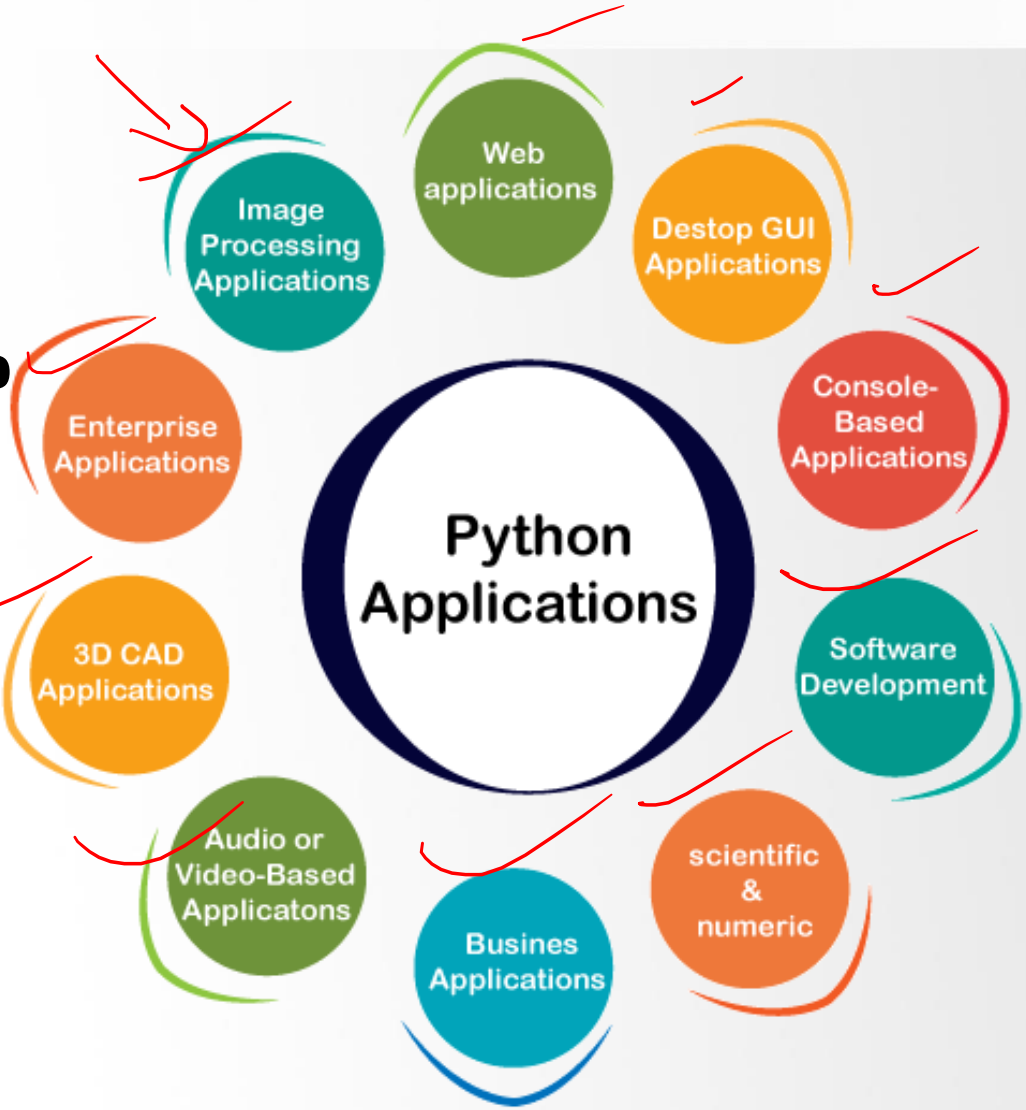


Linux on Raspberry Pi

- **Raspbian**
  - Raspbian Linux is a Debian Wheezy port optimized for Raspberry Pi.
- **Arch**
  - Arch is an Arch Linux port for AMD devices.
- **Pidora**
  - Pidora Linux is a Fedora Linux optimized for Raspberry Pi.
- **RaspBMC**
  - RaspBMC is an XBMC media-center distribution for Raspberry Pi.
- **OpenELEC**
  - OpenELEC is a fast and user-friendly XBMC media-center distribution.
- **RISC OS**
  - RISC OS is a very fast and compact operating system.

Python

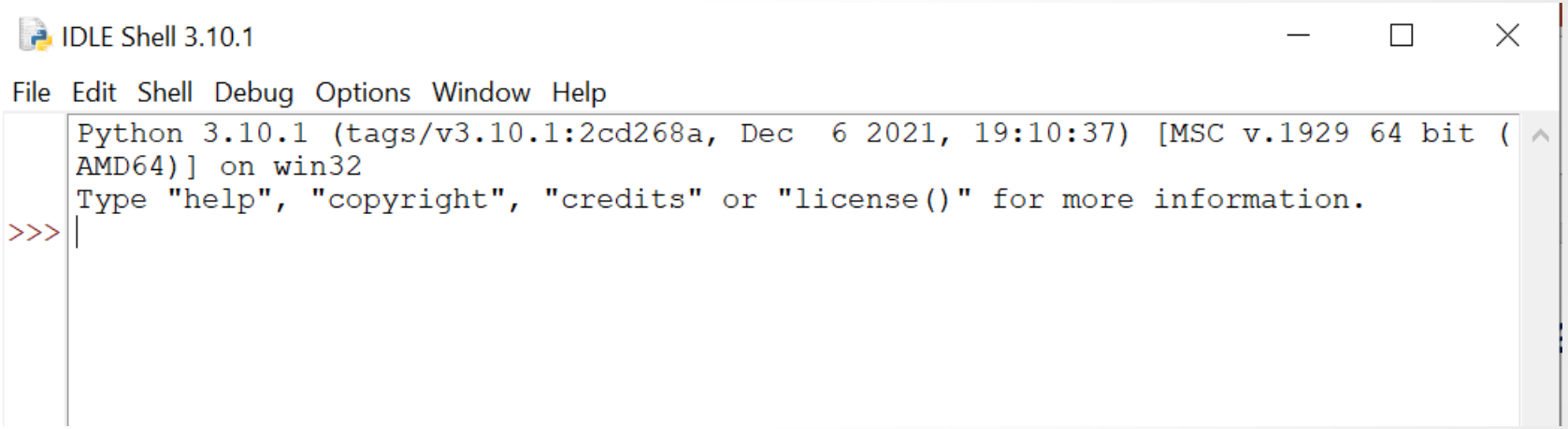
- Python is a widely used high-level programming language created by **Guido van Rossum in the late 1980s.**
- The language places a strong emphasis on code readability and simplicity, making it possible for programmers to develop applications rapidly.



Installing Python

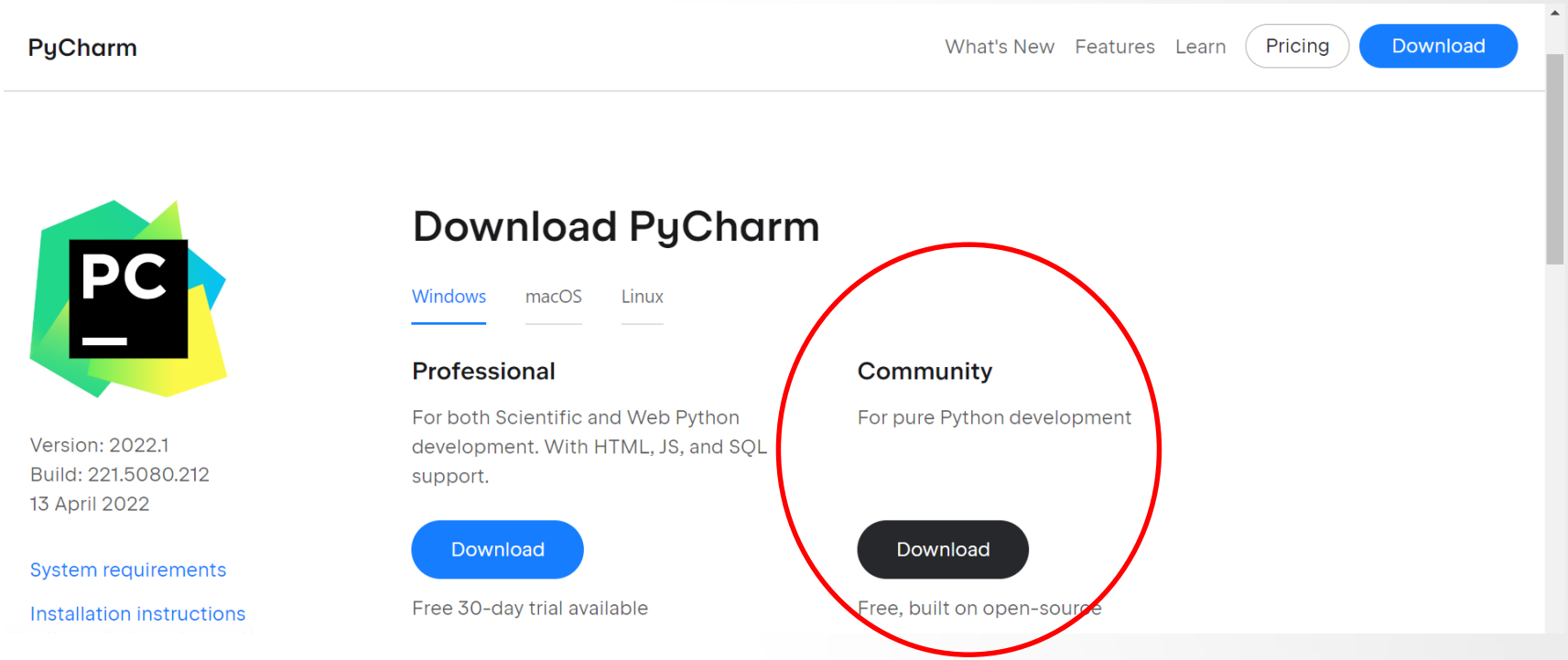
- To install the interpreter for Python 3, head over to https://www.python.org/downloads/

- Python code uses the IDLE program that comes bundled with our Python interpreter.



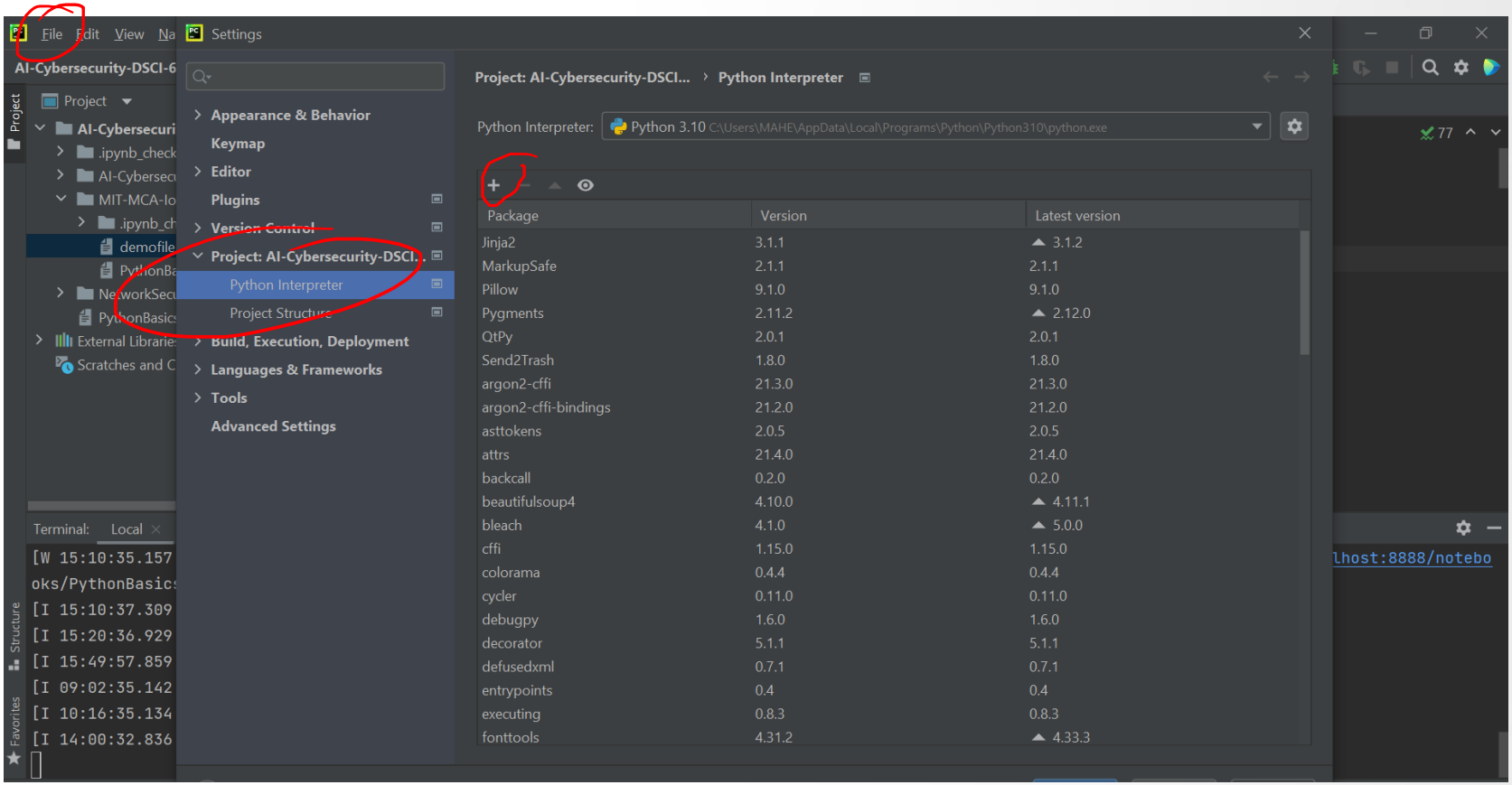
Installing PyCharm

- https://www.jetbrains.com/pycharm/download/#section=windows



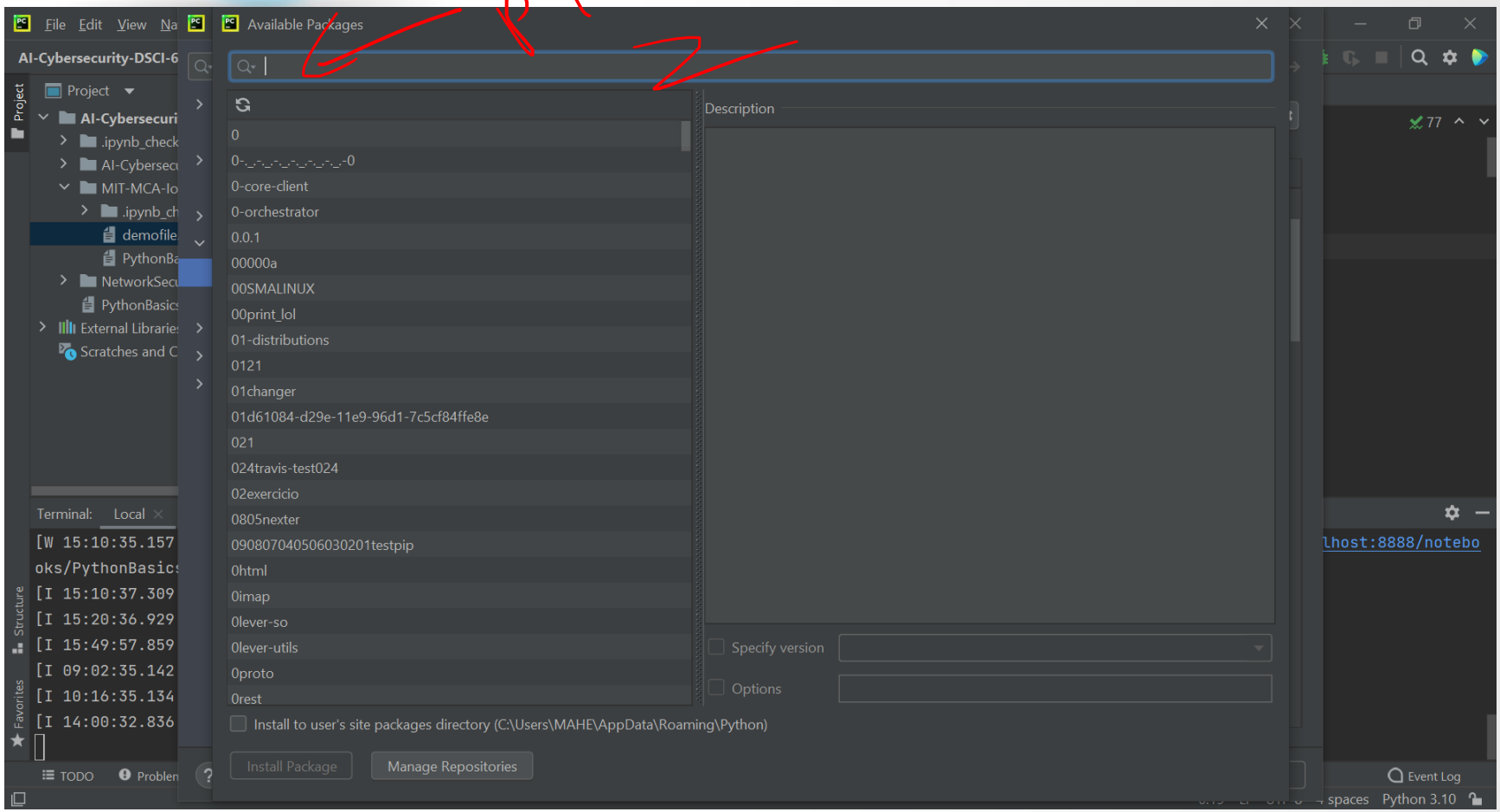
PyCharm Environment

- File → Settings →YourProjectName →Python Interpreter → “+” → search and install packages.



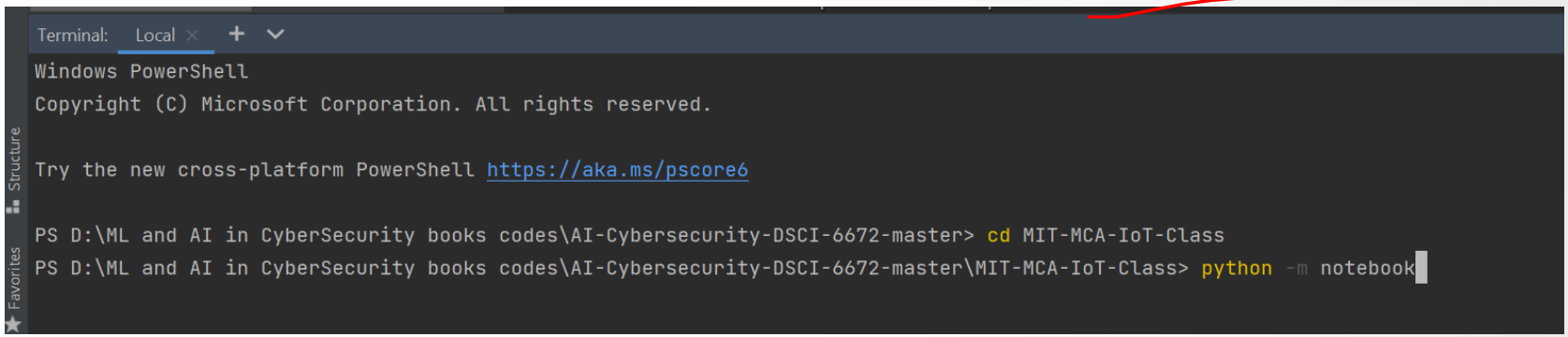
PyCharm Environment

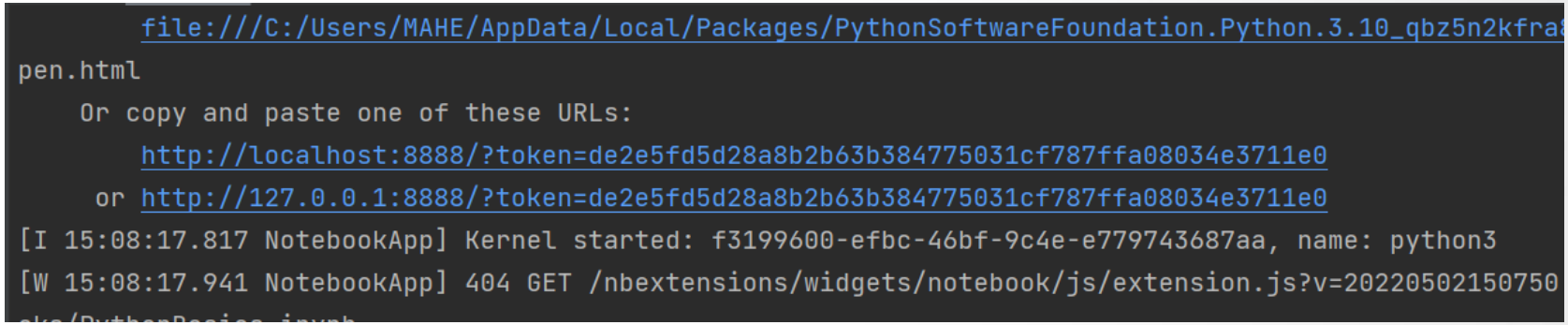
(contd..)



Installing Jupyter NoteBook on PyCharm

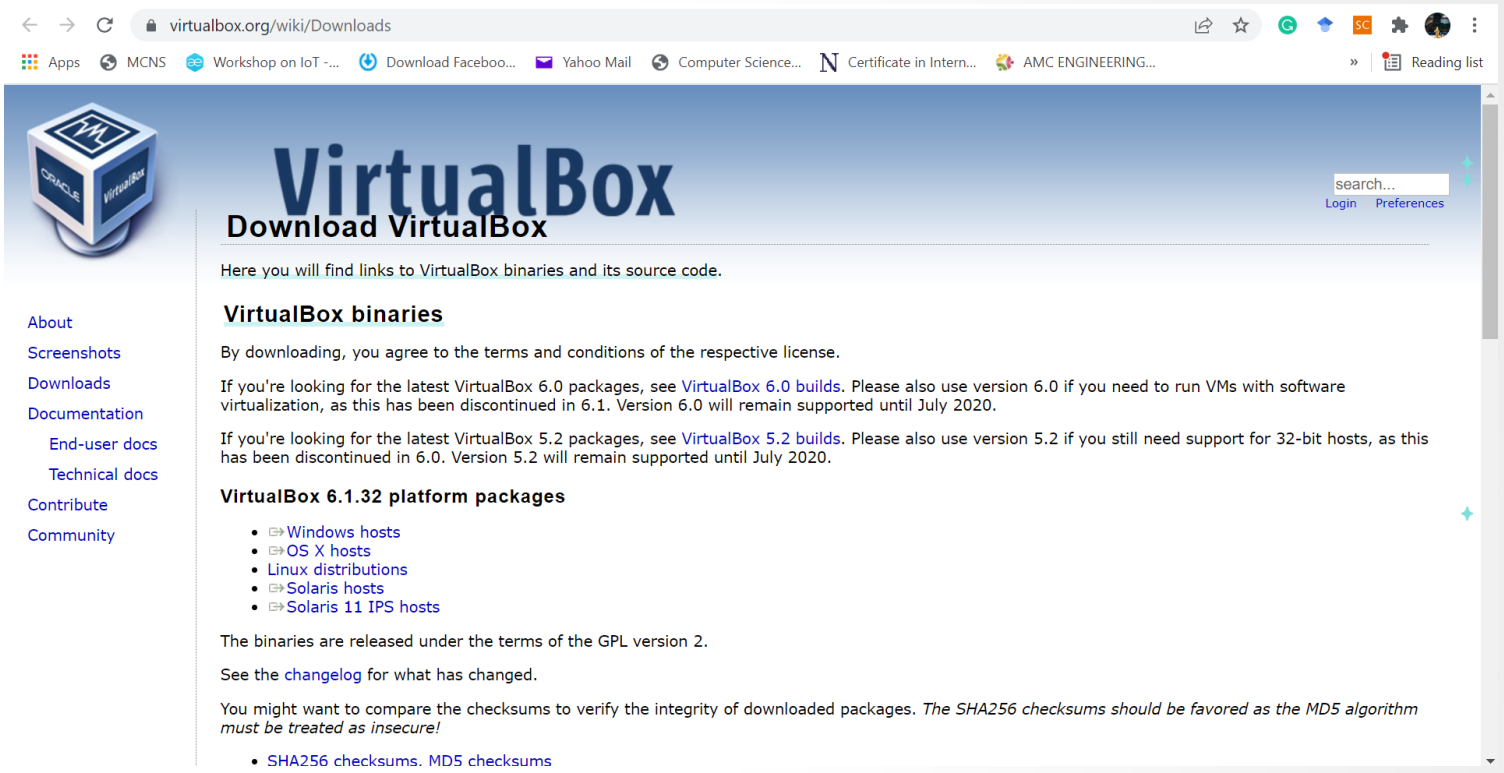
- Jupyter Notebook on PyCharm---- https://jupyter.org/install
  - Install Jupyter-lab → pip install juypter-lab
  - Install notebook → jupyter install notebook
  - Run Jupyter Notebook → python -m notebook





Virtual Home Lab

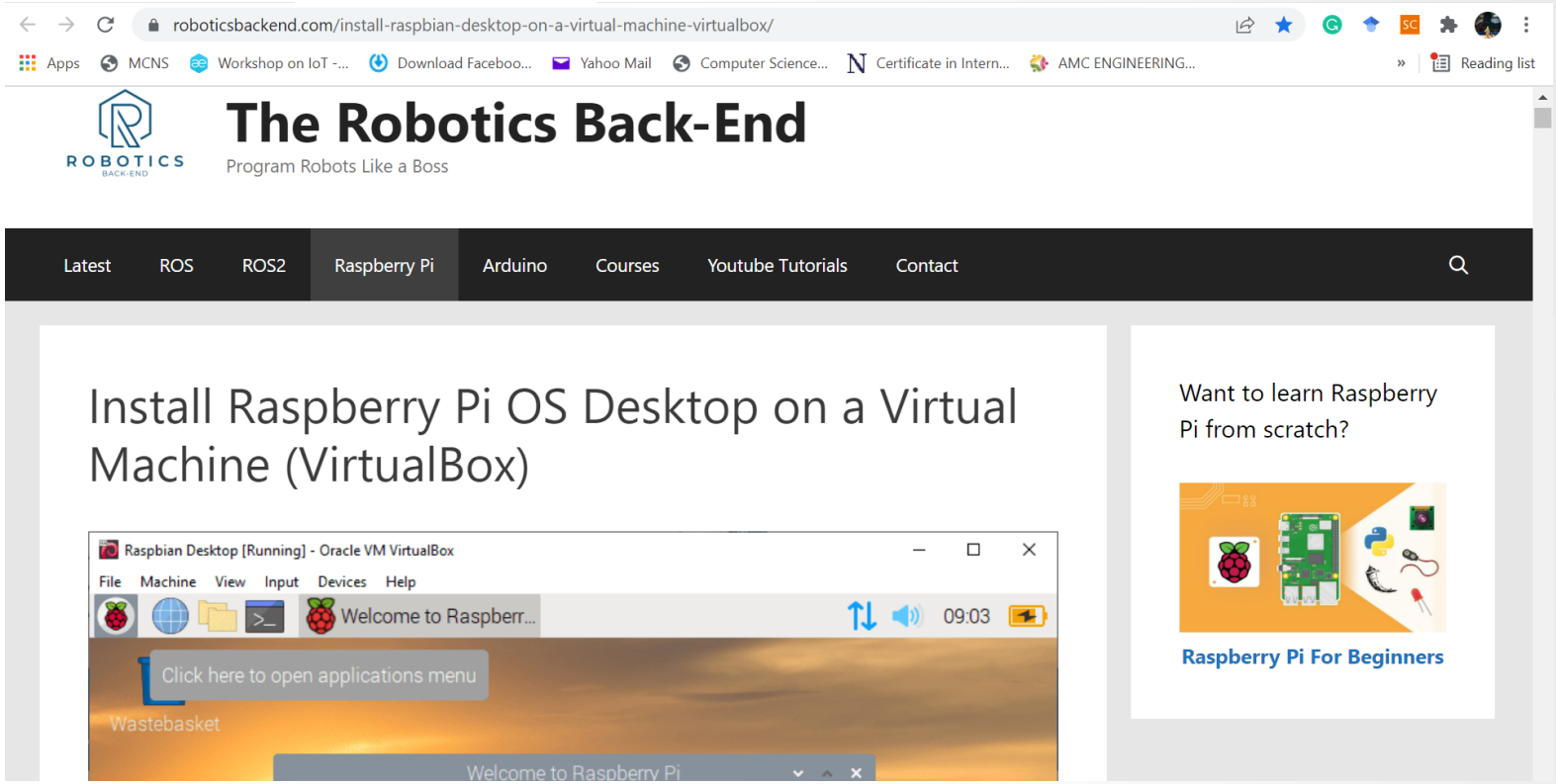
Virtual Box



https://www.virtualbox.org/wiki/Downloads

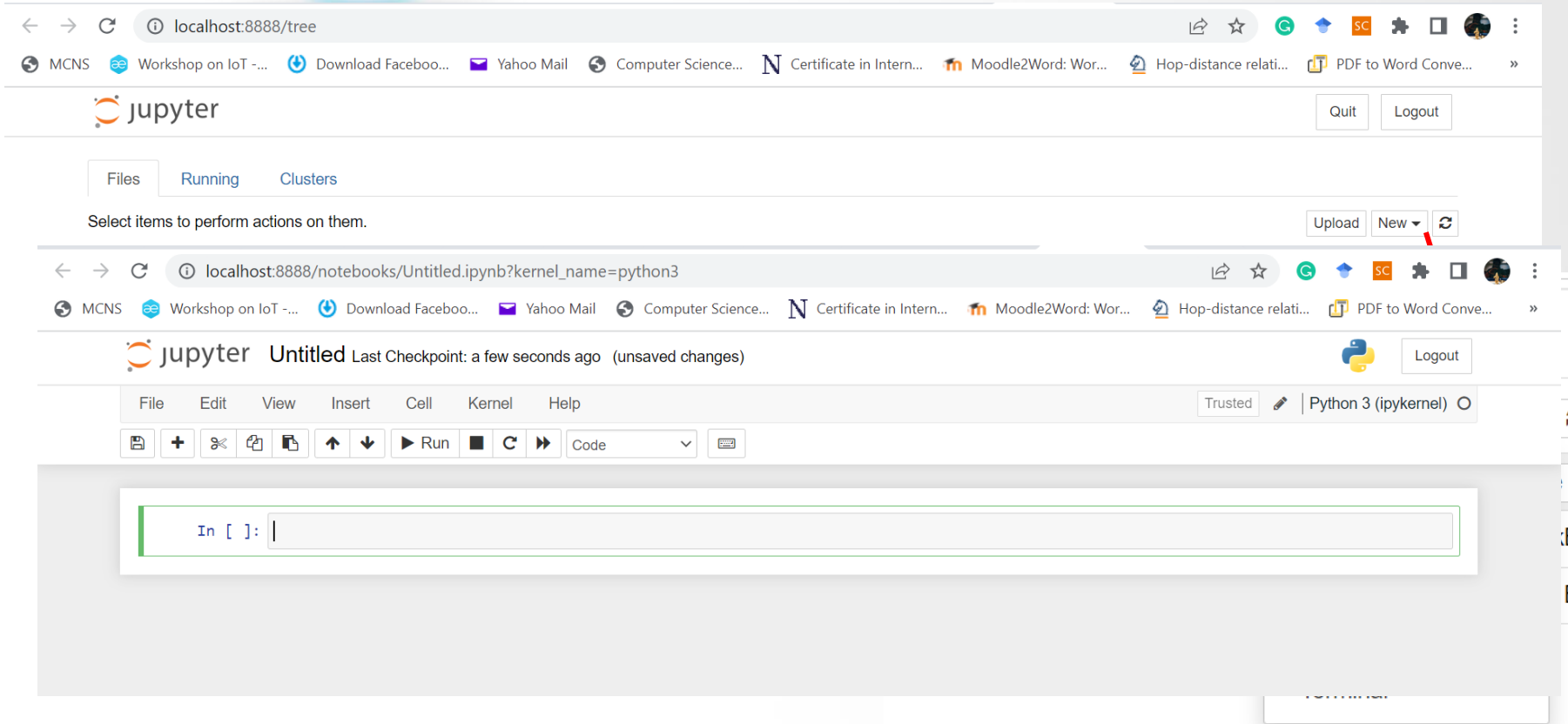
Virtual Home Lab

Raspberry Pi for virtual machine



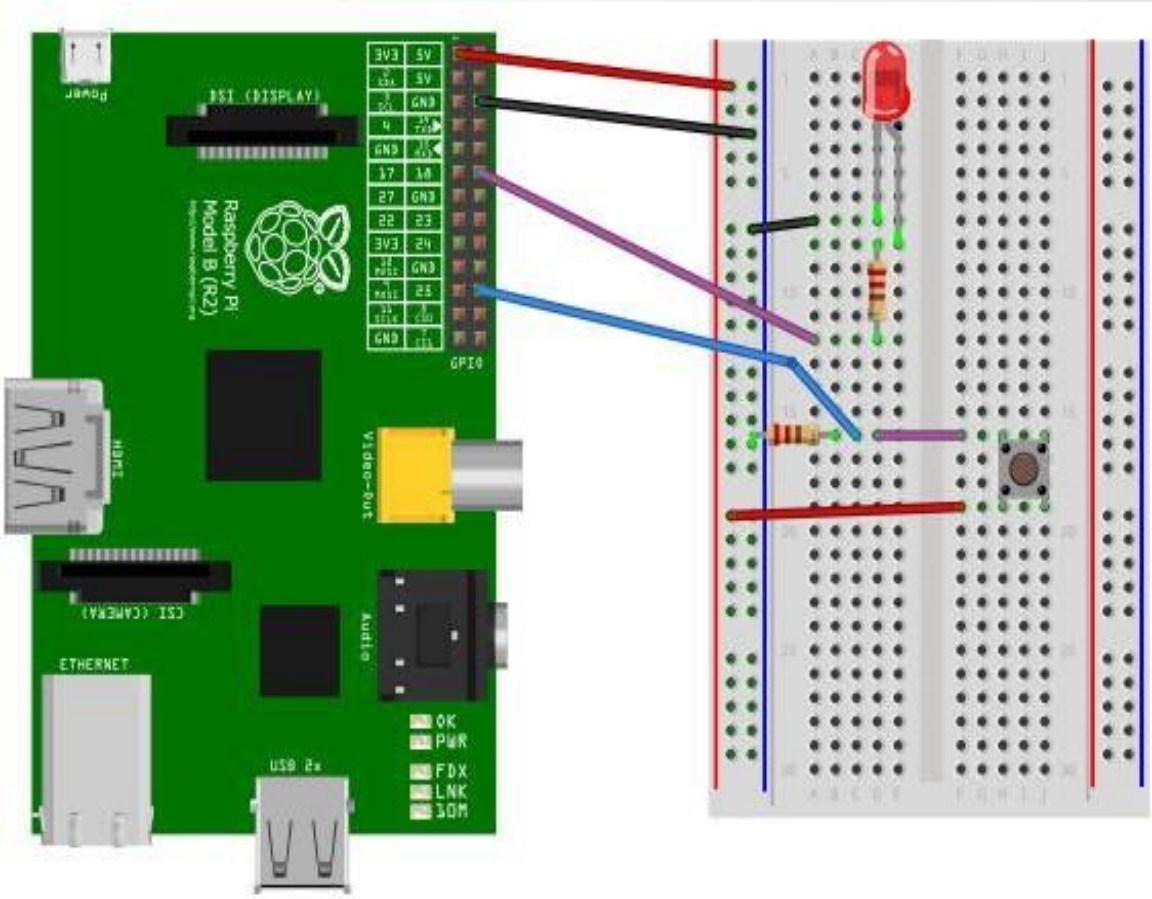
https://roboticsbackend.com/install-raspbian-desktop-on-a-virtual-machine-virtualbox/

Installing Jupyter NoteBook on PyCharm

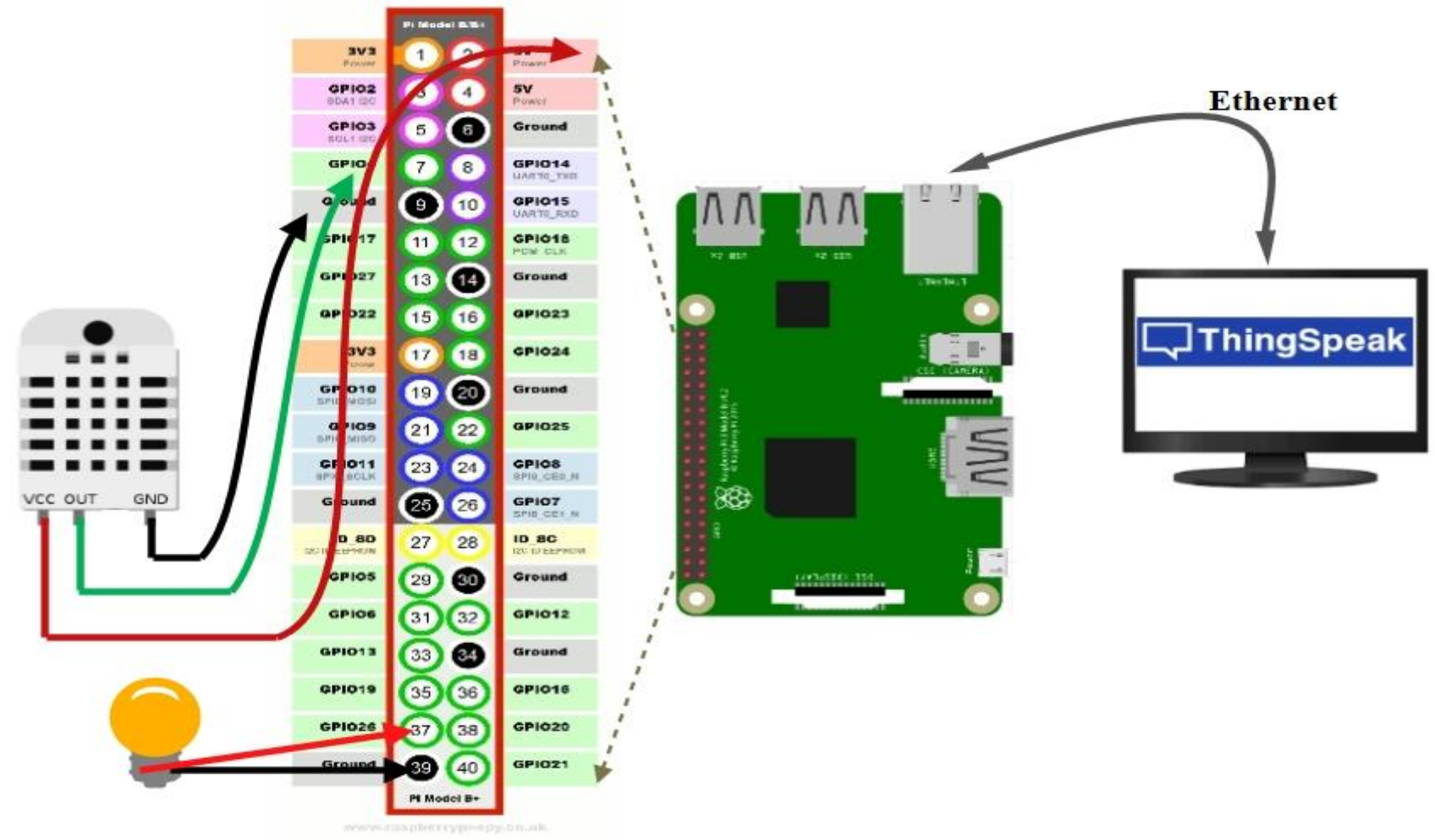


Raspberry Pi Example: Interfacing LED and switch with Raspberry Pi

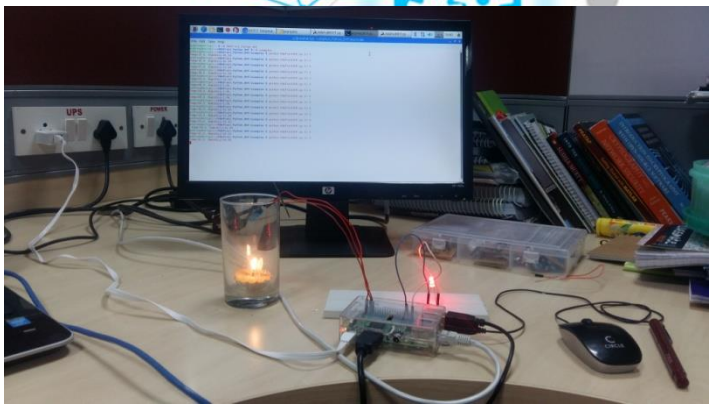
```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)

#Switch Pin  GPIO.setup(25,
GPIO.IN) #LED Pin
GPIO.setup(18,  GPIO.OUT)
state=false

def  toggleLED(pin):
        state = not state
        GPIO.output(pin,  state)

while  True:
        try:
                if (GPIO.input(25) == True):
                        toggleLED(pin)
                sleep(.01)
                except  KeyboardInterrupt:
                        exit()
```

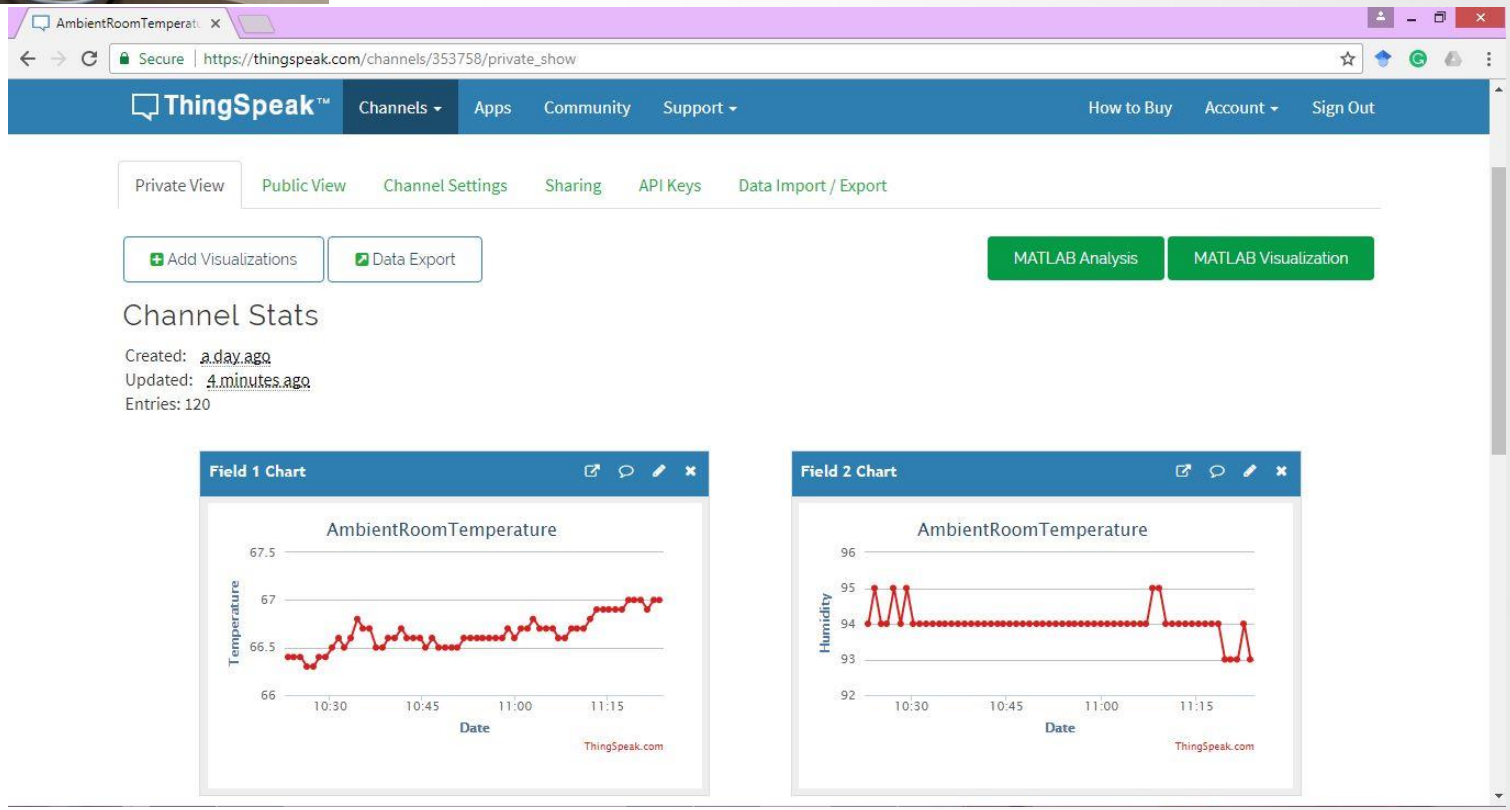


Temperature Monitoring using RPi

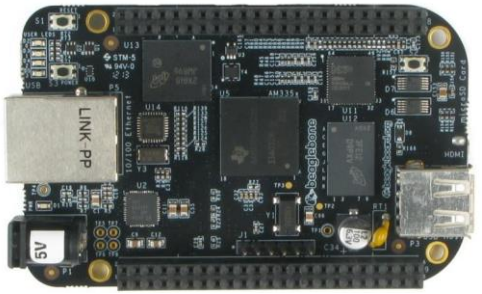


Temperature Monitoring using RPi



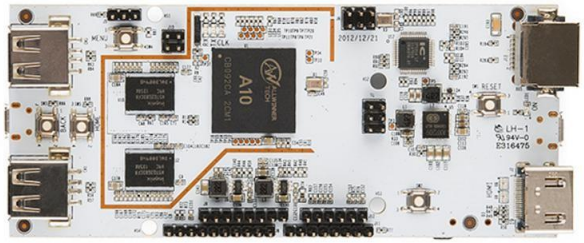


Other Devices

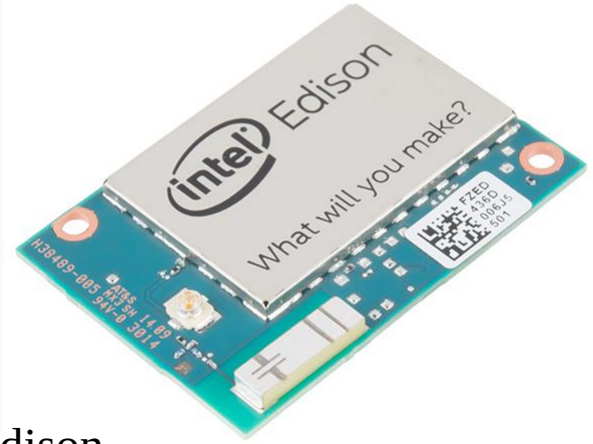


pcDuino

Cubieboard



BeagleBone Black



Intel Edison


WHAT
NEXT