ISIM 2021

International Symposium on
IoT and ML for
Ecosystem Restoration &
Multihazard Resilience
05th to 09th of June 2021

Introduction to Git & GitHub

Sai Shibu

Note: Contents are from publicly available data

About Me

Research Assistant III at Center for Wireless Networks and Applications, Amrita Vishwa Vidyapeetham

Research TAG: Intelligent Infrastructure & Pervasive Mobile Computing

Area of Interest: Future Mobility, Blockchain Technology, IoT & ML

Social Profiles

Git & Github

Git is an example of version control

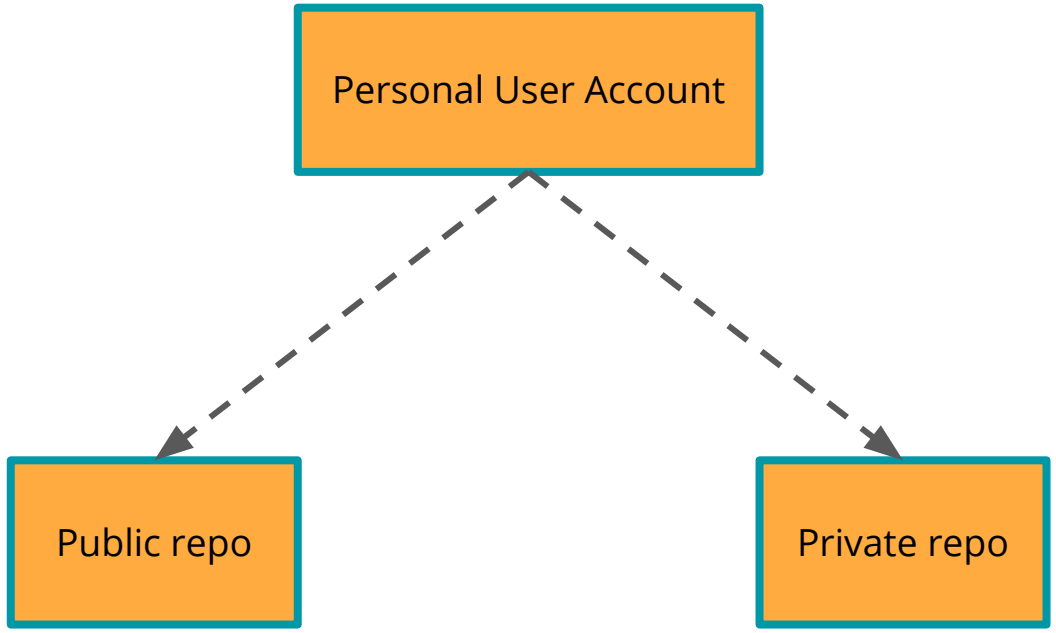
Version control is a system that records changes to a file or set of files and helps us recall specific versions later if needed. E.g. Subversion (SVN), CVS etc It allows you to :

- Revert files or the whole project to an earlier state
- Compare changes over time
- See who modified what?
- Control modifications by collaborators with the permission of admin/owners

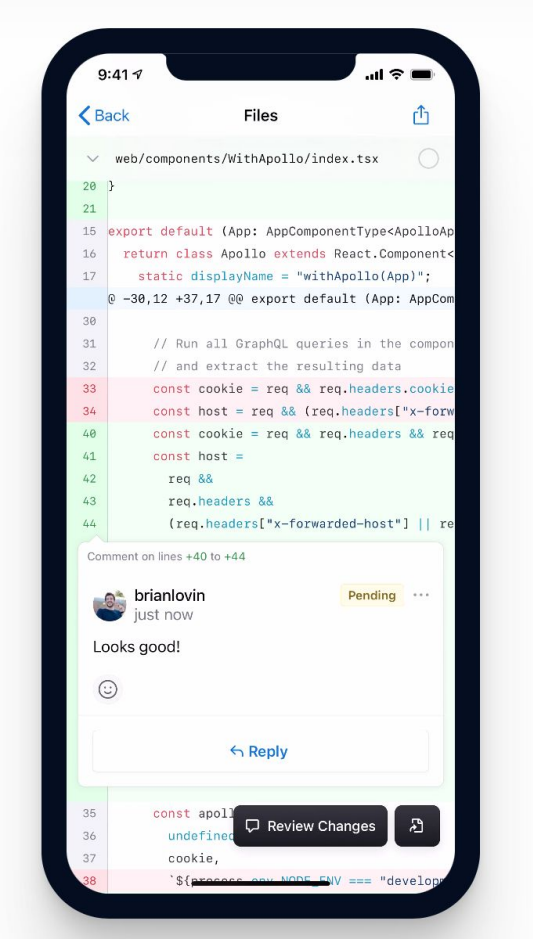
Github is a repository hosting service for Git

Github

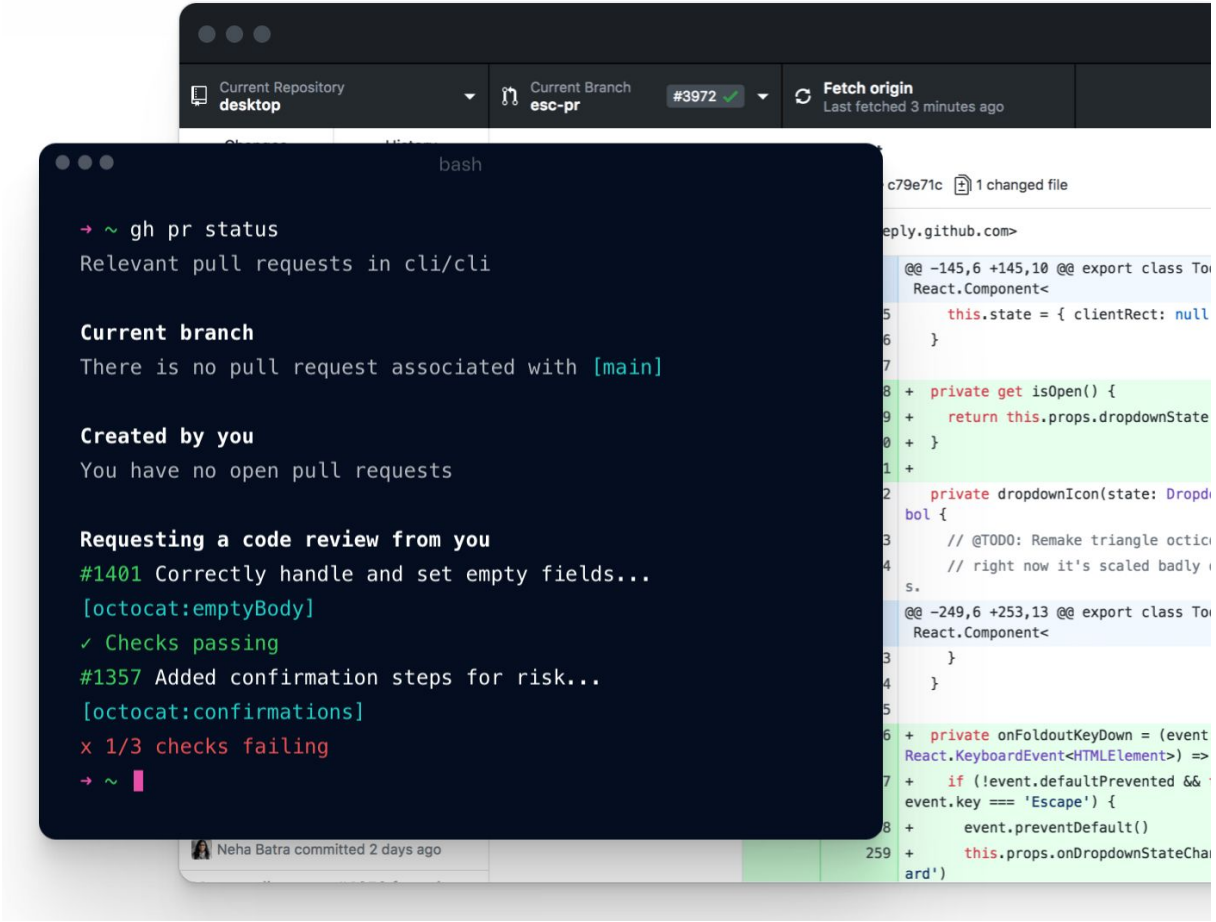
- Unlimited public repositories and collaborators on all plans
- Limited Private repositories
- Ability to add unlimited repository collaborators
- Public repositories are open to view and copy but not commit changes.



GitHub Clients



Mobile App



CLI & GUI Desktop App

ISIM 2021

International Symposium on
IoT and ML for
Ecosystem Restoration &
Multihazard Resilience
05th to 09th of June 2021

AMRITA VISHWA VIDYAPEETHAM
School for Sustainable Development

United Nations Educational, Scientific and Cultural Organization
UNESCO Chair on Experential Learning for Sustainable Innovation and Development, Amrita Vishwa Vidyapeetham, Kerala, India

esri India
THE SCIENCE OF WHERE

AMRITA CENTER FOR WIRELESS NETWORKS & APPLICATIONS

Quick Demo


iSiM 2021
International Symposium on
IoT and ML for
Ecosystem Restoration &
Multihazard Resilience
05th to 09th of June 2021
AMRITA
School for Sustainable Development
esri India
Thank You!

ISIM 2021

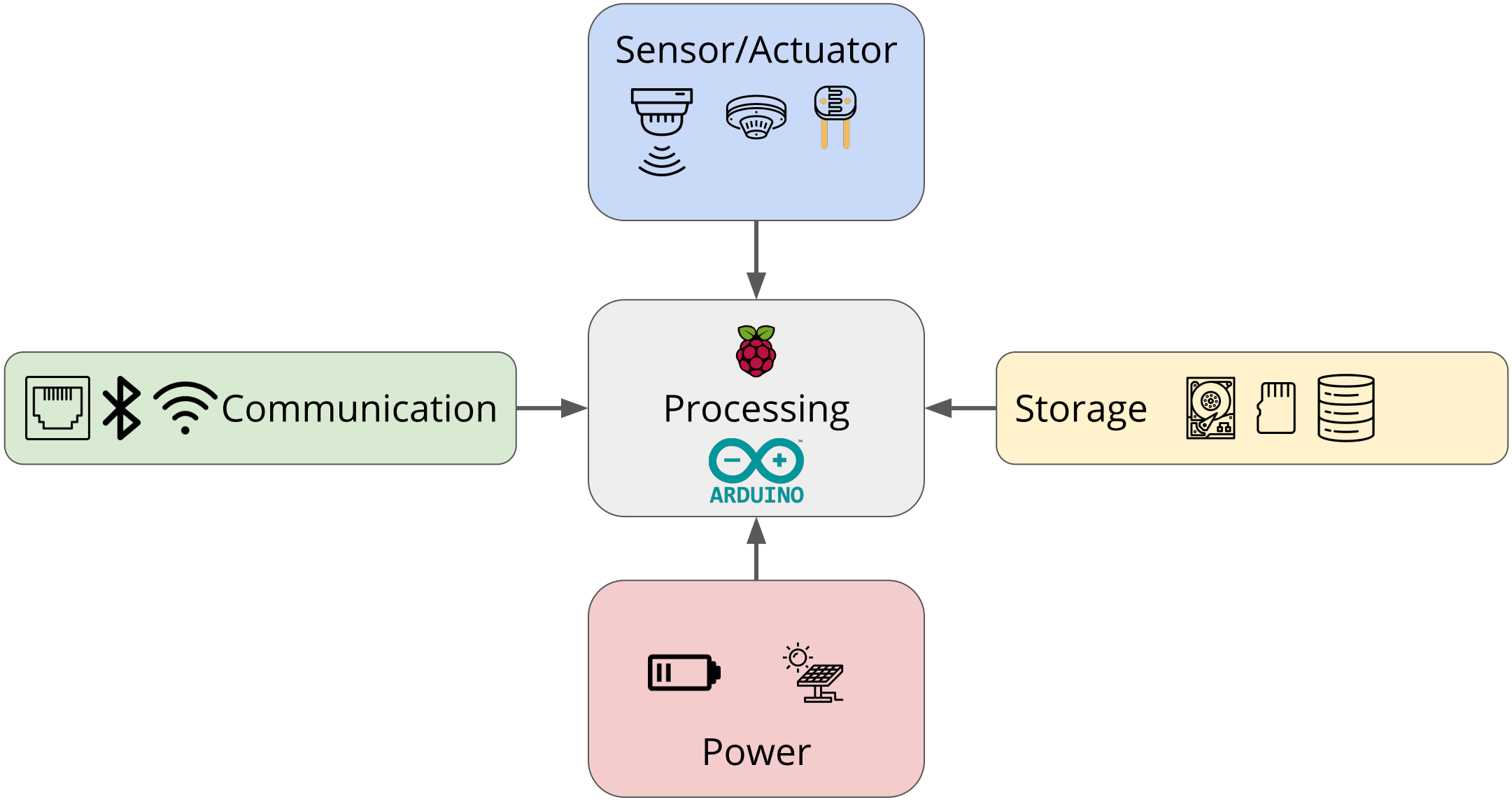
International Symposium on
IoT and ML for
Ecosystem Restoration &
Multihazard Resilience
05th to 09th of June 2021

Introduction to Single Board Computers

Sai Shibu

Note: Contents are from publicly available data

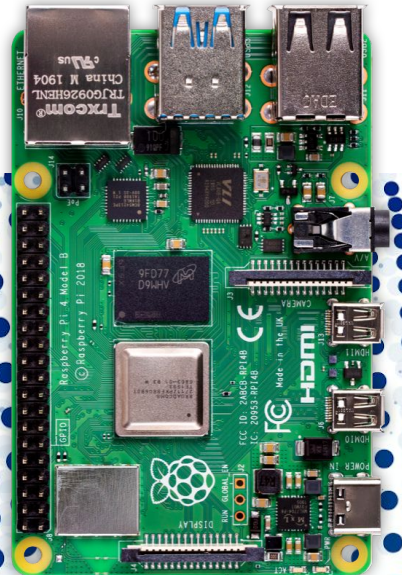
General IoT Architecture - Simplified form



Raspberry Pi

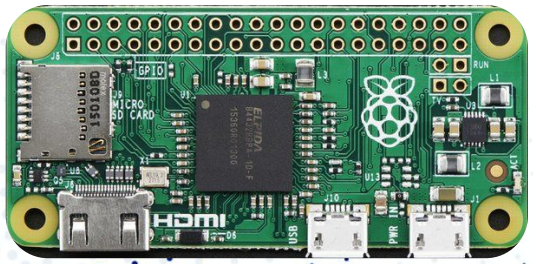
The **Raspberry Pi** is a credit-card-sized single-board computer developed in the UK by the Raspberry Pi Foundation for promoting the teaching of basic computer science in schools.
Makers, tinkerers and hackers adapted the R-Pi to act as a tiny computer that they can easily embed into their projects.

RPi 4

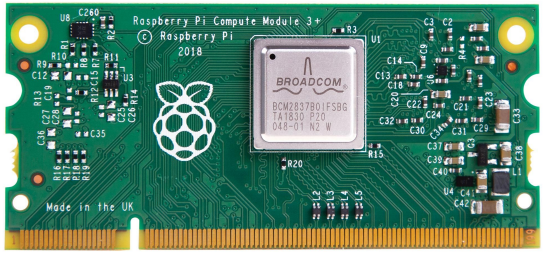


RPi 3A

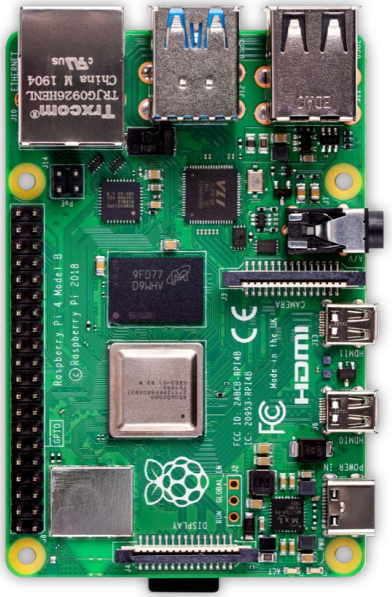
RPi ZERO W



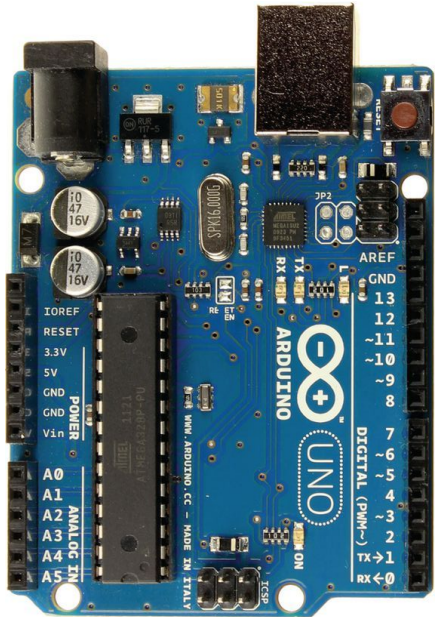
RPi COMPUTE MODULE



Raspberry Pi vs Arduino

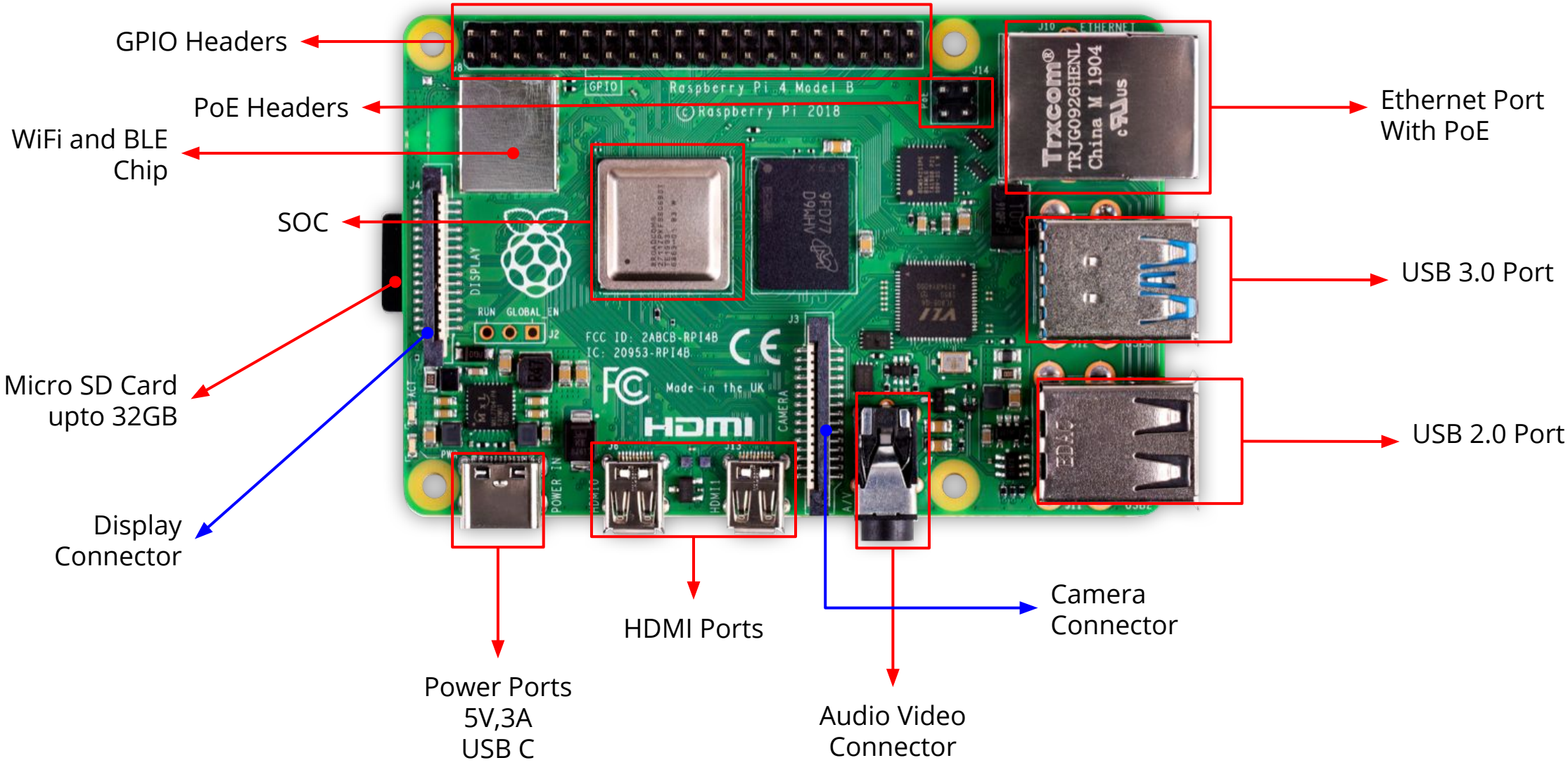


- OS Support
- Onboard WiFi & Bluetooth
- Parallel Processing
- Storage
- Multiple language support
- Sensitive to current
- No ADC
- 3.3V GPIOs



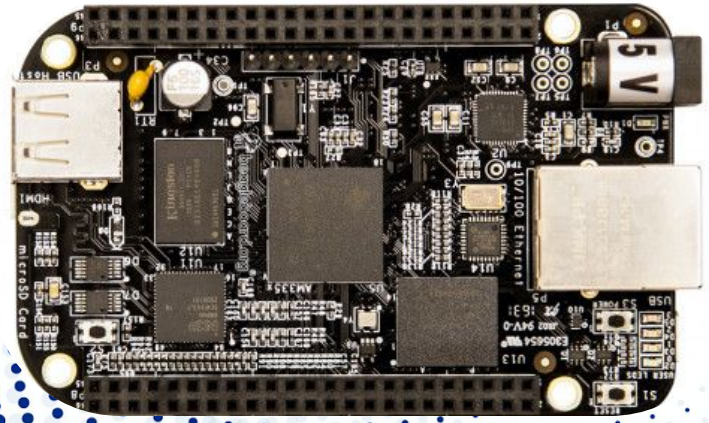
- Onboard ADC
- Low cost
- Less sensitive to current
- 5V I/O Ports
- Single programming Language
- low memory & Storage
- less number of I/Os

Raspberry Pi - Board Layout

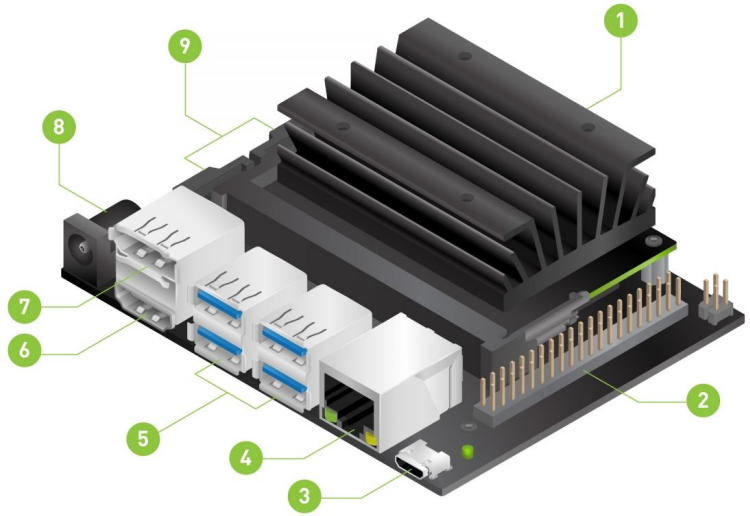


Do we have any other SBCs other than RPi?

Yes

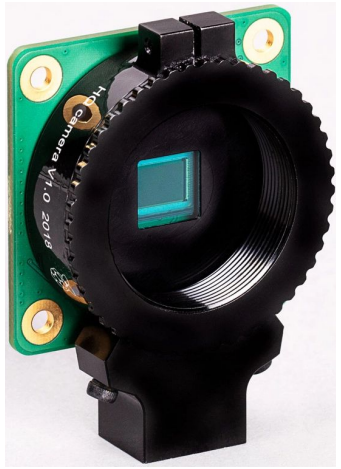


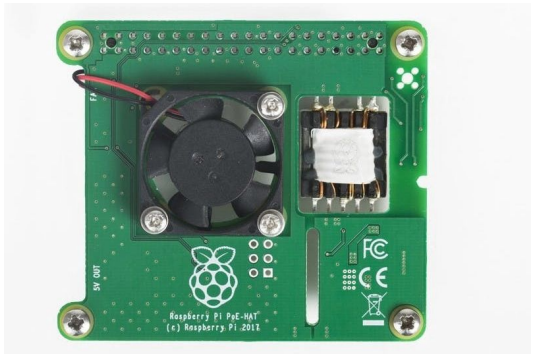
Beagle Bone

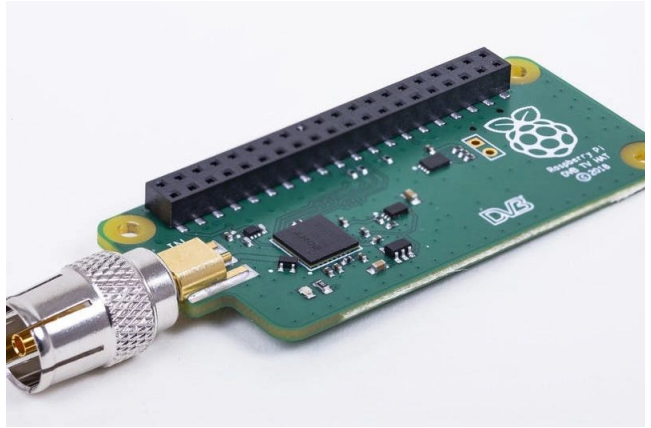


Nvidia Jetson Nano

Accessories for Raspberry Pi

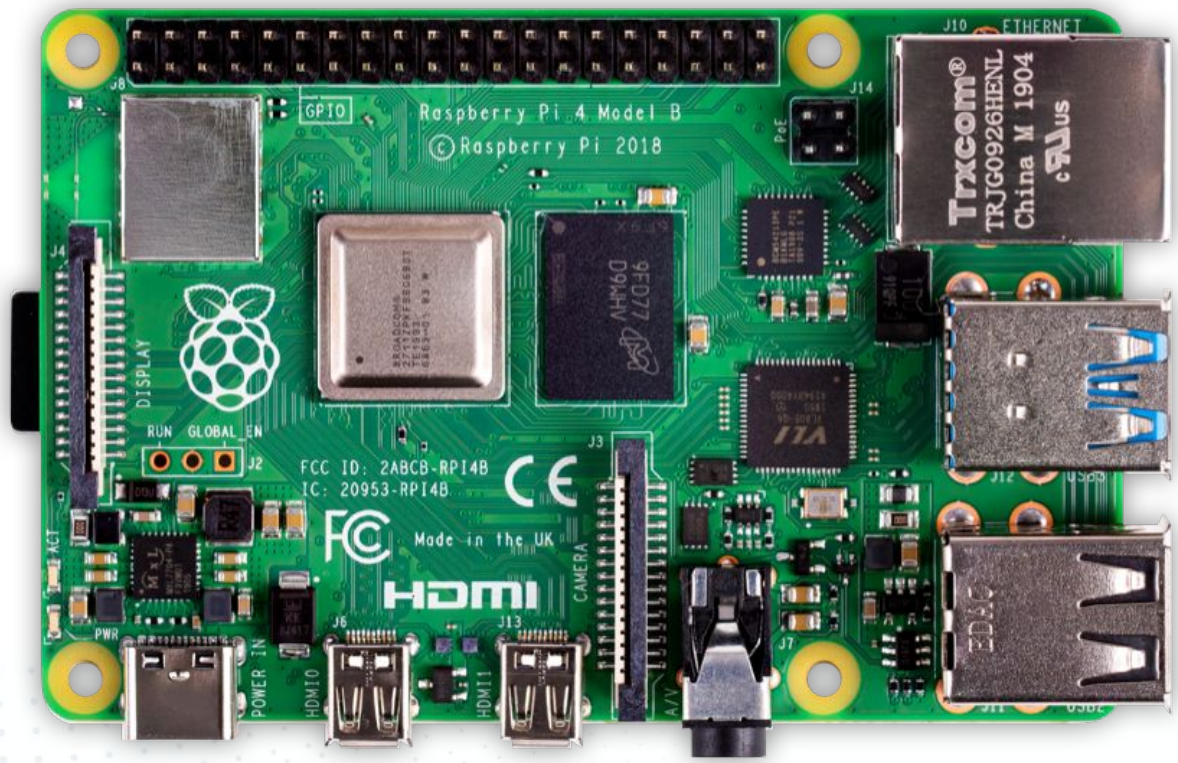






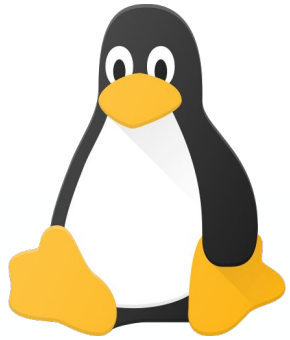
Let's get started

1. Input devices - Keyboard & Mouse
2. Display
3. SD with OS on it



What is OS? Why do we need this?

- Intermediate between user application and hardware
- Provides, file system, I/O Operation, program control, memory management



OS for Raspberry Pi

Raspberry Pi OS - 32 bit operating system

Light version and Full version

- Stripped down Linux with required packages for Raspberry Pi
- Python and other programming languages included
- Support to access GPIOs

Any other OS available for RPI?

Ubuntu MATE

Ubuntu Core

Ubuntu Server

OSMC

LibreELEC

Mozilla WebThings

PiNet

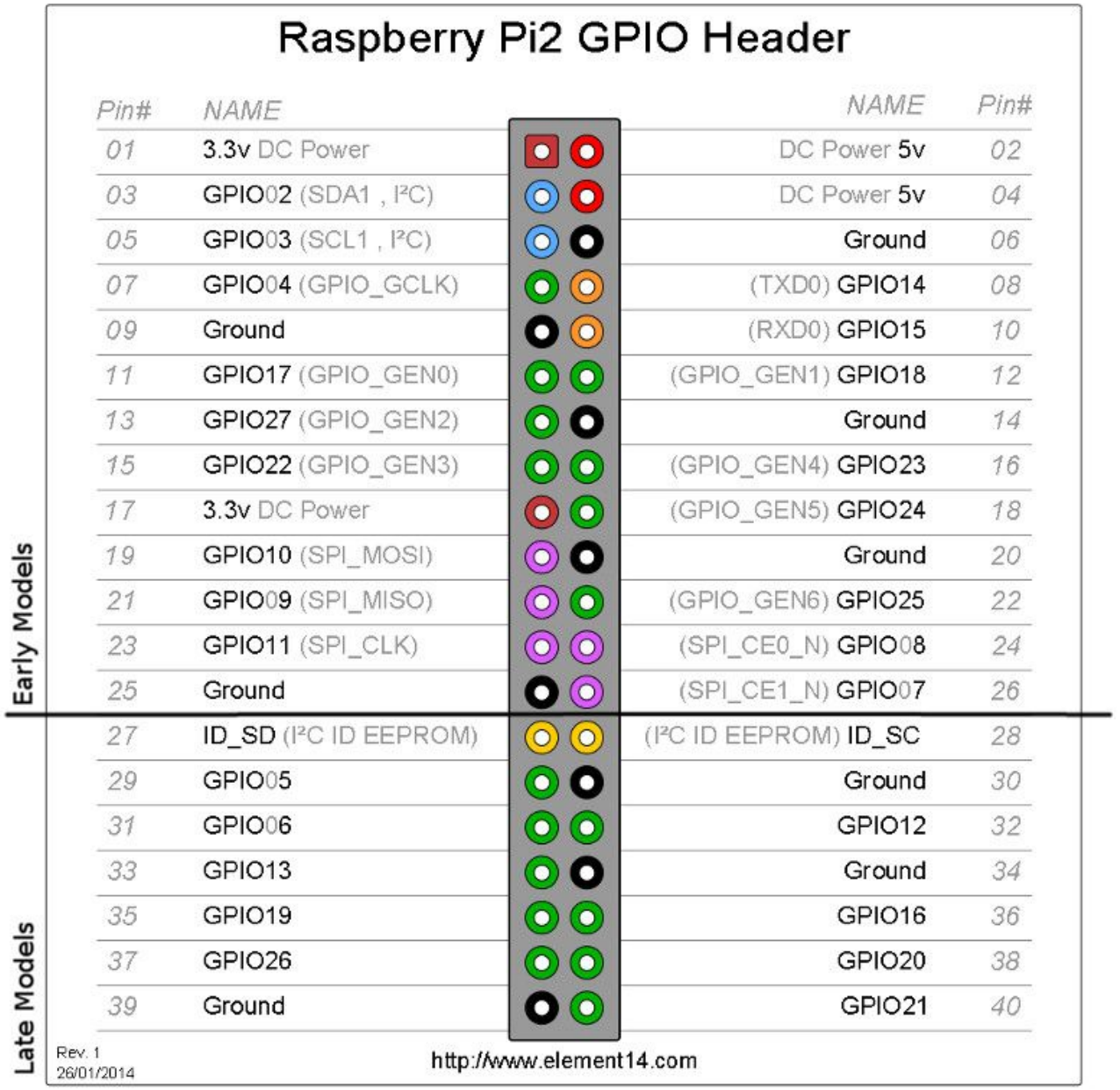
RISC OS

Weather Station

IchigoJam RPi

GPIO

- Connect Sensors and Actuators
- Digital Input and output similar to arduino. But 3.3v range & 10mA
- lack of ADC
- Possible to connect external ADCs through SPI or I2C
- PWM for motor/LED Control
- UART for communication



Using the GPIOs (Python based)

GPIO Library

Very similar to Arduino Digital Input & Output

Set GPIO to Input / Output

If input - read the specific pin to get the value

if output - set high/low to control

More reading: https://www.raspberrypi.org/documentation/usage/gpio/

UART, SPI & I2C have to enabled through raspi-config

specific libraries allow data read/write operations through these protocols.

Modbus protocol is used for industry M2M communications, eg: reading data from a smart meter. No native support available, but RS232 to Modbus converter is possible

Other SBC - Beaglebone supports CANBUS

Using GPIO

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7,GPIO.OUT)

for i in range(50):
    GPIO.output(7,True)
    time.sleep(1)
    GPIO.output(7,False)
    time.sleep(1)

GPIO.cleanup()
```

Import the GPIO Library

Choose GPIO Pinmode

Config GPIO as output

Set GPIO to high

Set GPIO to low

Clean GPIO configs

Using GPIO - Input

```
#buttonInput.py
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)

sleepTime = .1

#GPIO Pin of the component
lightPin = 4
buttonPin = 17

GPIO.setup(lightPin, GPIO.OUT)
GPIO.setup(buttonPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)
GPIO.output(lightPin, False)

try:
    while True:
        GPIO.output(lightPin, GPIO.input(buttonPin))
        sleep(.1)
finally:
    GPIO.output(lightPin, False)
    GPIO.cleanup()
```

Import the GPIO Library

Choose GPIO Pinmode

Config GPIO as Input

Read GPIO Input

Clean GPIO configs

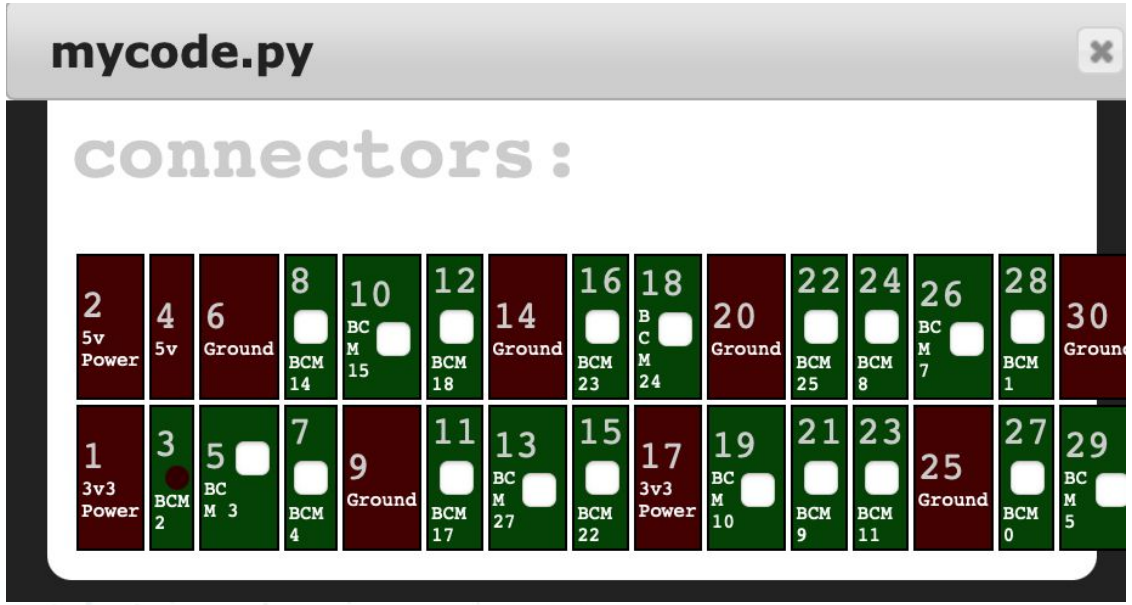
Simulators (1)

https://www.iot4smes.eu/en/demonstrator.aspx?id=4.full

UNED R-Pi Simulator

mycode.py

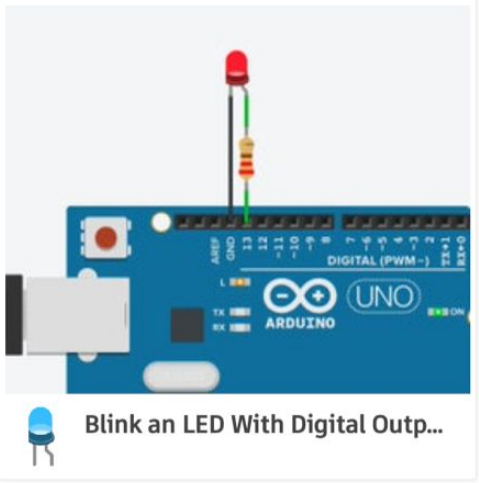
```
1  import RPi.GPIO as GPIO
2  import time
3
4  GPIO.setmode(GPIO.BOARD)
5  GPIO.setup(3, GPIO.OUT)
6  GPIO.setup(5, GPIO.IN)
7
8  # flash pin 3 five times
9  for i in range(5):
10     GPIO.output(3, GPIO.HIGH)
11     time.sleep(1)
```

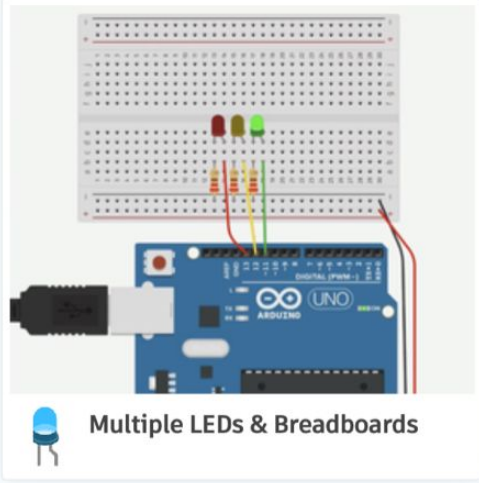


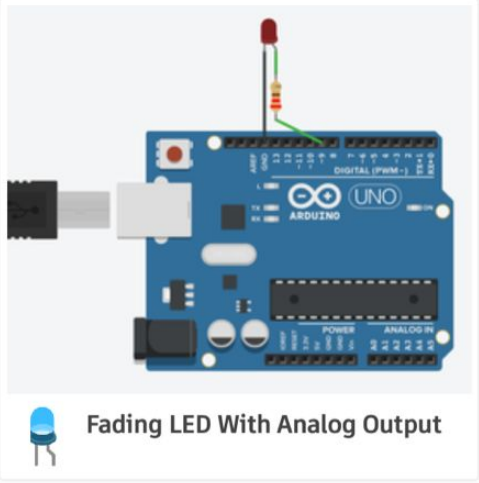
Simulators (2)

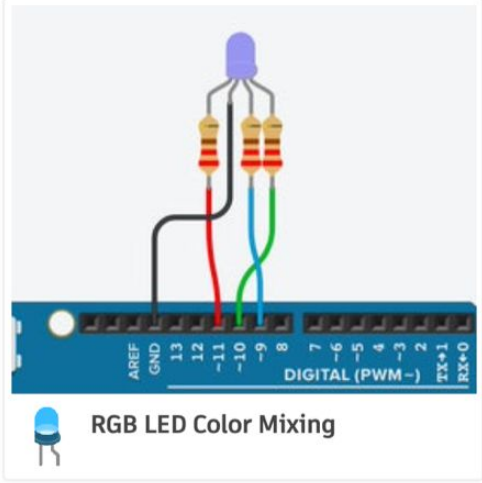
https://www.tinkercad.com

Learn Arduino

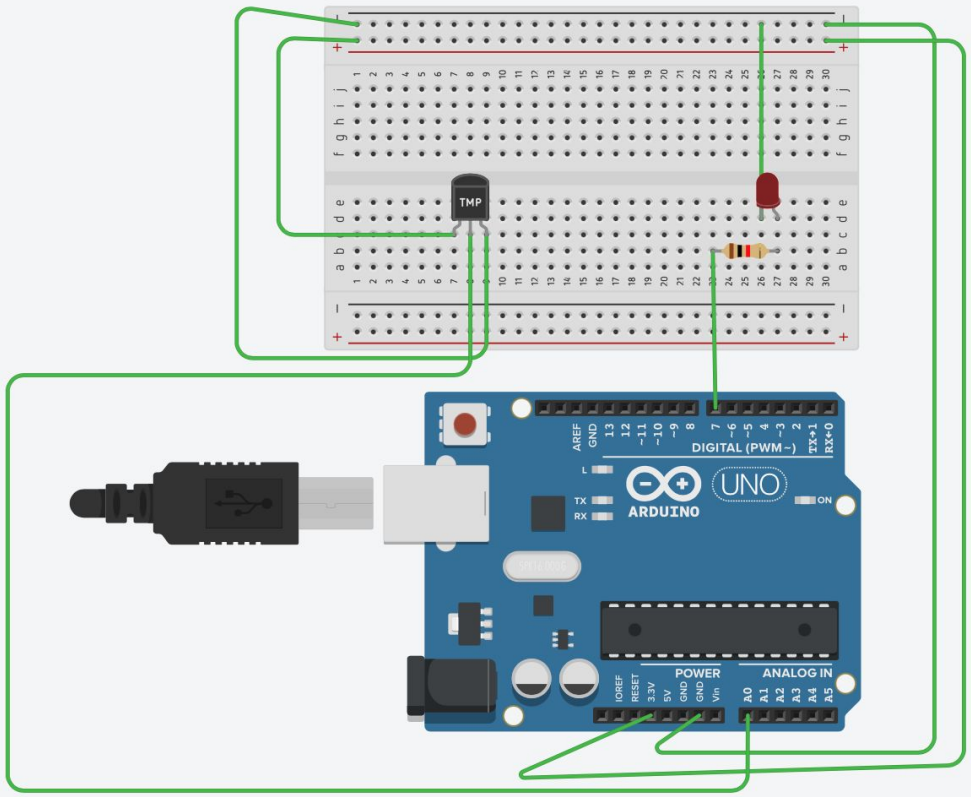








Show all Arduino



Blocks

Output
Input
Notation
Control
Math
Variables

set built-in LED to HIGH

pin 0 to HIGH

pin 3 to 0

ate servo on pin 0 to 0 degr

y speaker on pin 0 with tone 6

n off speaker on pin 0

nt to serial monitor hello world wit

RGB LED in pins 3 3

1 (Arduino Uno R3)

```
set Temp to read analog pin A0
print to serial monitor Temp with newline
set Temp to map Temp to range -40 to
print to serial monitor Temp with newline
if Temp > 25 then
    set pin 7 to HIGH
else
    set pin 7 to LOW
```

Serial Monitor

ACKNOWLEDGMENTS / REFERENCES

- http://pibeginners.com/
- https://www.youtube.com/user/RaspberryPiBeginners
- https://www.coursehero.com/file/21672571/rpi-lec/
- https://www.raspberrypi.org/
- https://www.amrita.edu/research/project/advanced-integrated-wireless-sensor-networks-real-time-monitoring-and-detection
- https://www.amrita.edu/research/project/micronet-mobile-infrastructure-coastal-region-offshore-communications-networks
- https://www.amrita.edu/research/project/amrita-wireless-smart-grid