IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Introduction to Raspberry Pi – Part I

Dr. Sudip Misra
Associate Professor
Department of Computer Science and Engineering
IIT KHARAGPUR
Email: smisra@sit.iitkgp.ernet.in
Website: http://cse.iitkgp.ac.in/~smisra/

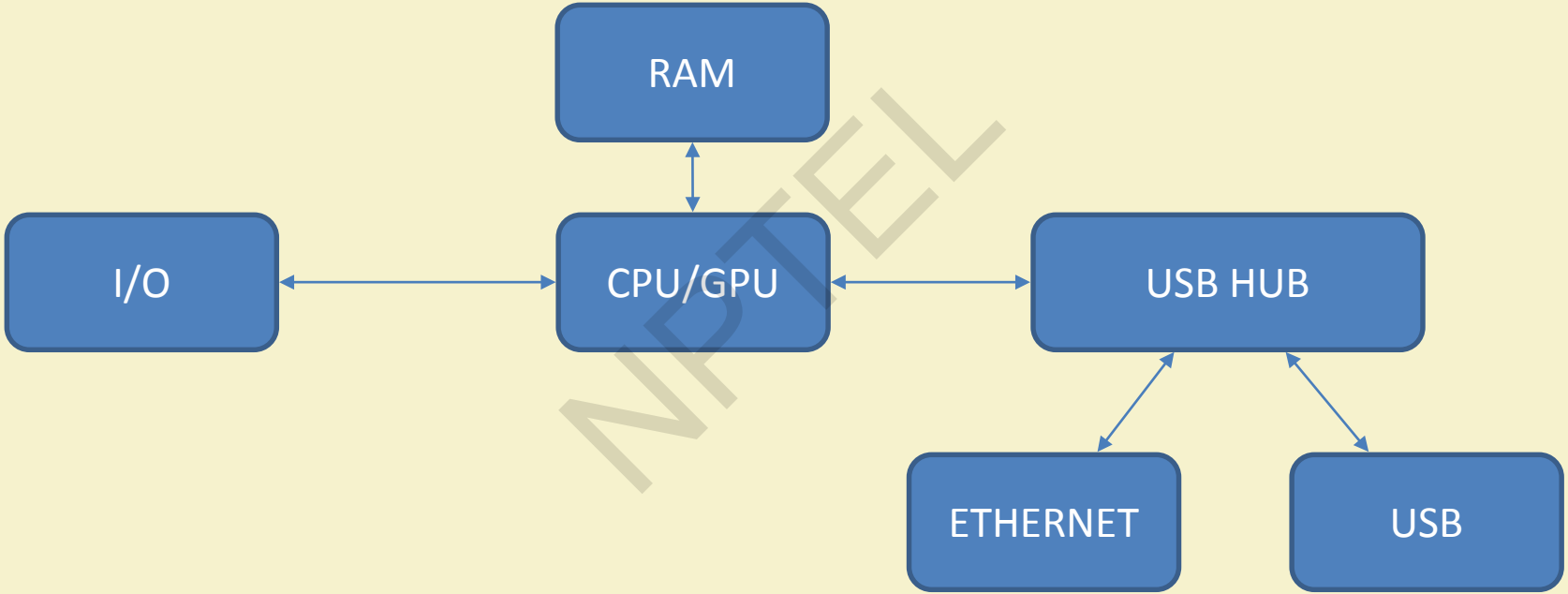
What is Raspberry Pi?

- Computer in your palm.
- Single-board computer.
- Low cost.
- Easy to access.

Specifications

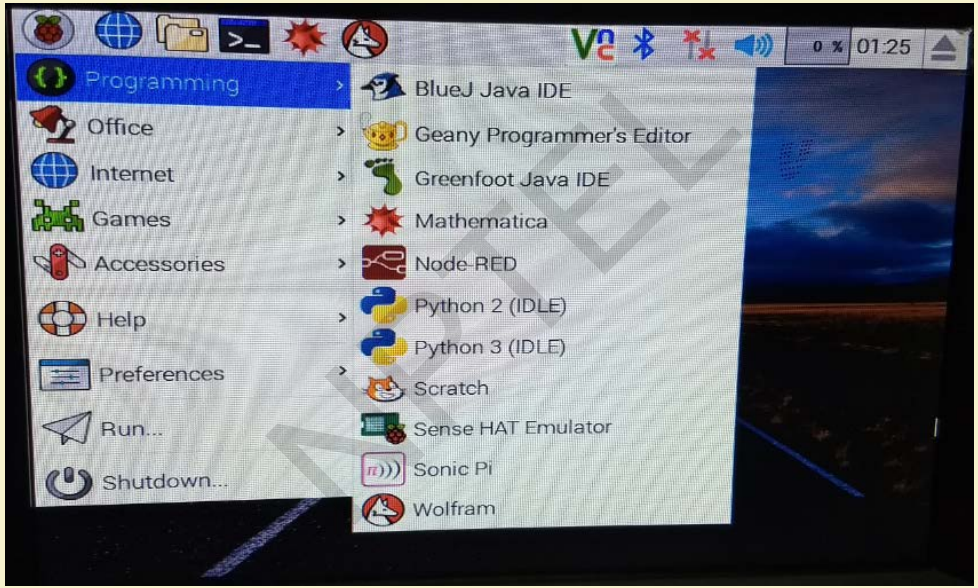
Key features	Raspberry pi 3 model B	Raspberry pi 2 model B	Raspberry Pi zero
RAM	*1GB SDRAM*	*1GB SDRAM*	*512 MB SDRAM*
CPU	*Quad cortex A53@1.2GHz*	*Quad cortex A53@900MHz*	*ARM 11@ 1GHz*
GPU	*400 MHz video core IV*	*250 MHz video core IV*	*250 MHz video core IV*
Ethernet	*10/100*	*10/100*	*None*
Wireless	*802.11/Bluetooth 4.0*	*None*	*None*
Video output	*HDMI/Composite*	*HDMI/Composite*	*HDMI/Composite*
GPIO	*40*	*40*	*40*

Basic Architecture



Raspberry Pi

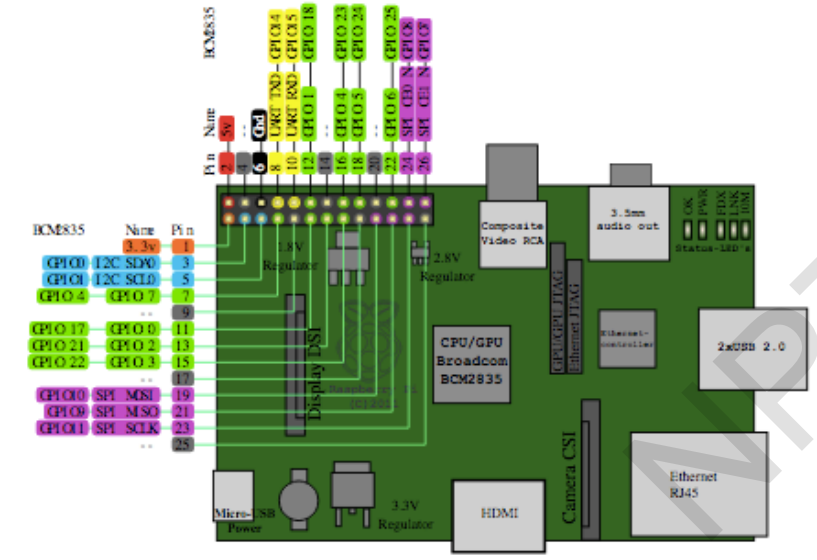
Start up raspberry pi



Raspberry Pi GPIO

- Act as both digital output and digital input.
- **Output**: turn a GPIO pin high or low.
- **Input**: detect a GPIO pin high or low.

Raspberry Pi pin configuration



Source: Raspberry Pi PCB Pin Overview, Wikimedia Commons (Online)

Source: Raspberry Pi GPIO, Wikimedia Commons (Online)

Basic Set up for Raspberry Pi

- HDMI cable.
- Monitor.
- Key board.
- Mouse.
- 5volt power adapter for raspberry pi.
- LAN cable .
- Min- 2GB micro sd card

Basic Set up for Raspberry Pi

Operating System

Official Supported OS :

- Raspbian
- NOOBS

Some of the third party OS :

- UBUNTU mate
- Snappy Ubuntu core
- Windows 10 core
- Pinet
- Risc OS

Source: Downloads, Raspberry Pi Foundation

Raspberry Pi Setup

Download Raspbian:

- Download latest Raspbian image from raspberry pi official site: https://www.raspberrypi.org/downloads/

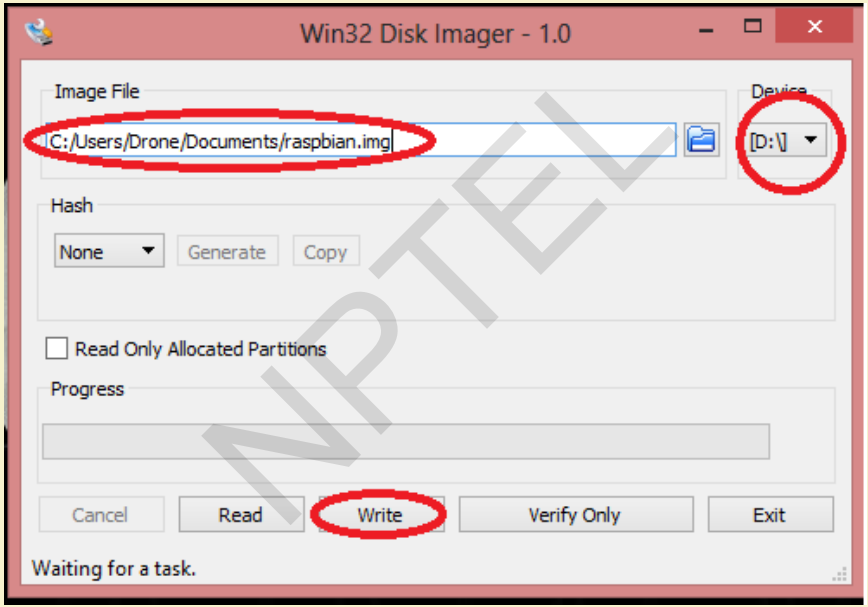
- Unzip the file and end up with an .img file.

Raspberry Pi OS Setup

Write Raspbian in SD card :

- Install “Win32 Disk Imager” software in windows machine .
- Run Win32 Disk Imager
- Plug SD card into your PC
- Select the “Device”
- Browse the “Image File”(Raspbian image)
- Write

Raspberry Pi OS Setup



Basic Initial Configuration

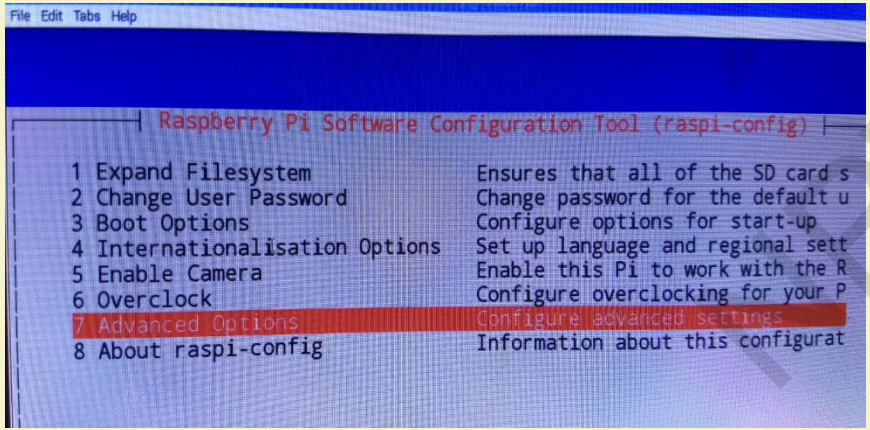
Enable SSH

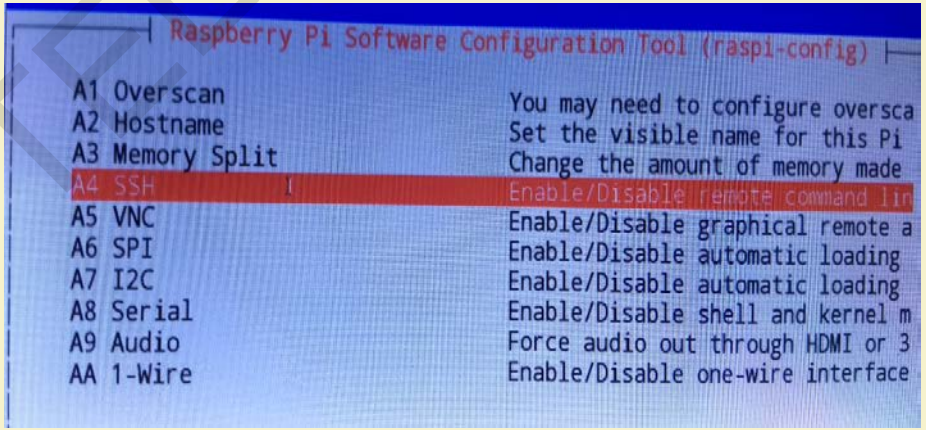
Step1 : Open command prompt and type **sudo raspi-config** and press enter.

Step2: Navigate to SSH in the Advance option.

Step3: Enable SSH

Basic Initial Configuration





Basic Initial Configuration contd.

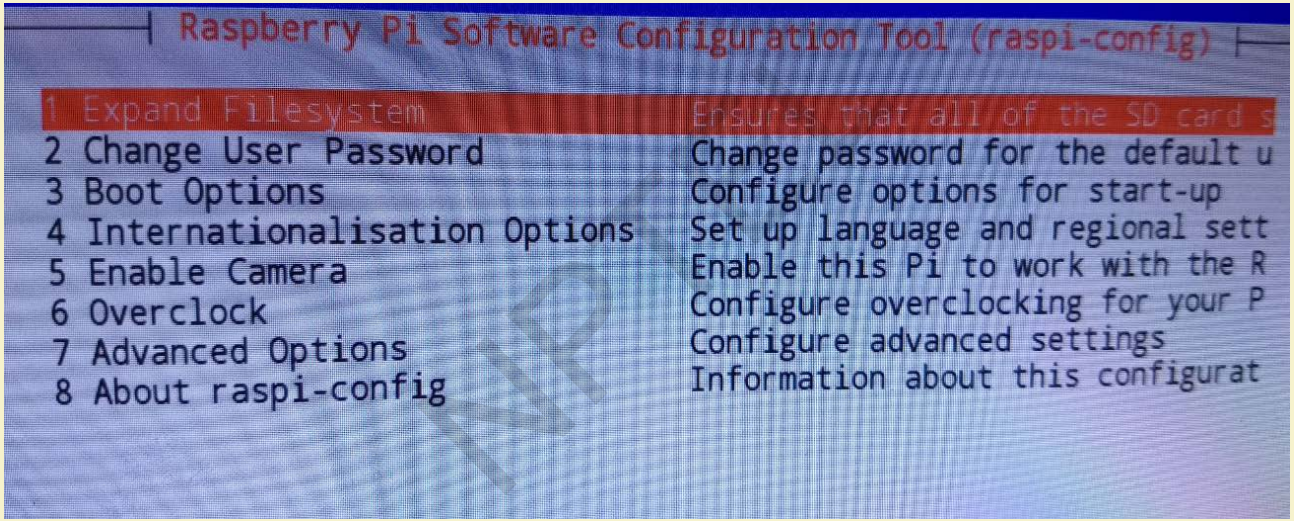
Expand file system :

Step 1: Open command prompt and type **sudo raspi-config** and press enter.

Step 2: Navigate to Expand Filesystem

Step 3: Press enter to expand it.

Basic Initial Configuration contd.



Programming

Default installed :

- Python
- C
- C++
- Java
- Scratch
- Ruby

Note : Any language that will compile for ARMv6 can be used with raspberry pi.

Source: Programming languages for Raspberry Pi, eProseed, Lonneke Dikmans, August 07, 2015

Popular Applications

- Media streamer
- Home automation
- Controlling BOT
- VPN
- Light weight web server for IOT
- Tablet computer

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Introduction to Raspberry Pi – Part II

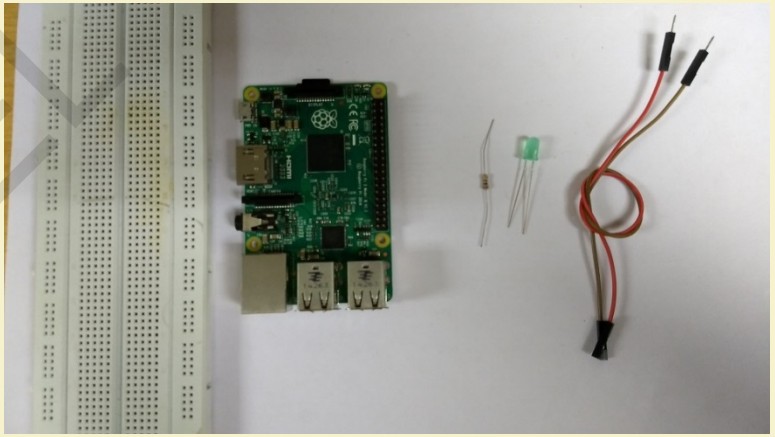
Dr. Sudip Misra
Associate Professor
Department of Computer Science and Engineering
IIT KHARAGPUR
Email: smisra@sit.iitkgp.ernet.in
Website: http://cse.iitkgp.ac.in/~smisra/

Topics Covered

- Using GPIO pins
- Taking pictures using PiCam

Blinking LED

- Requirement:
- Raspberry pi
- LED
- 100 ohm resistor
- Bread board
- Jumper cables



Blinking LED (contd..)

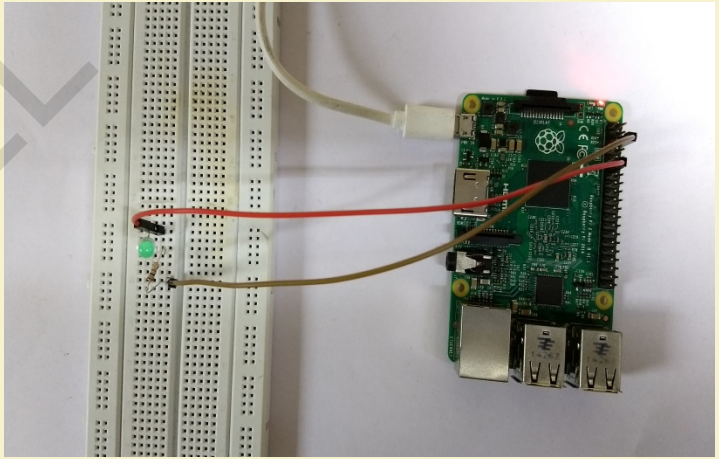
Installing GPIO library:

- Open terminal
- Enter the command “sudo apt-get install python-dev” to install python development
- Enter the command “sudo apt-get install python-rpi.gpio” to install GPIO library.

Blinking LED (contd..)

Connection:

- Connect the negative terminal of the LED to the ground pin of Pi
- Connect the positive terminal of the LED to the output pin of Pi



Blinking LED (contd..)

Basic python coding:

- Open terminal enter the command

  sudo nano filename.py

- This will open the nano editor where you can write your code
- Ctrl+O : Writes the code to the file
- Ctrl+X : Exits the editor

Blinking LED (contd..)

Code:

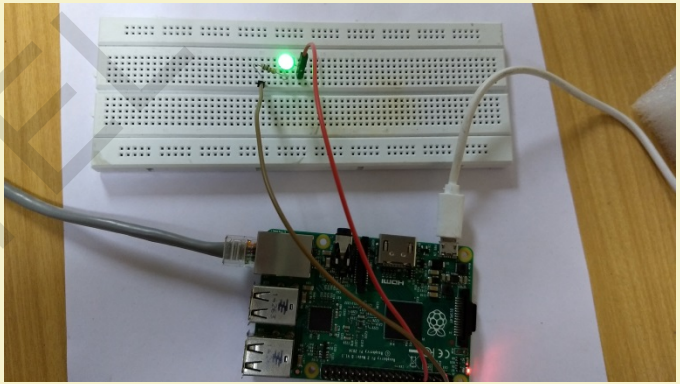
```
import RPi.GPIO as GPIO          #GPIO library
import time
GPIO.setmode(GPIO.BOARD)         # Set the type of board for pin numbering
GPIO.setup(11, GPIO.OUT)         # Set GPIO pin 11as output pin
for i in range (0,5):
   GPIO.output(11,True)          # Turn on GPIO pin 11
   time.sleep(1)
   GPIO.output(11,False)
   time.sleep(2)
   GPIO.output(11,True)
GPIO.cleanup()
```

Blinking LED (contd..)

```
  GNU nano 2.2.6                                    File: BLINK_LED.py

import RPi.GPIO as GPIO ## GPIO library
import time
GPIO.setmode(GPIO.BOARD) ## Set the type of board for pin numbering
GPIO.setup(11, GPIO.OUT) ## Set GPIO pin 11 as output pin
for i in range (0,5):
    GPIO.output(11,True) ## Turn on GPIO pin 11
    time.sleep(1)
    GPIO.output(11,False)
    time.sleep(2)
    GPIO.output(11,True)
GPIO.cleanup()
```

Blinking LED (contd..)

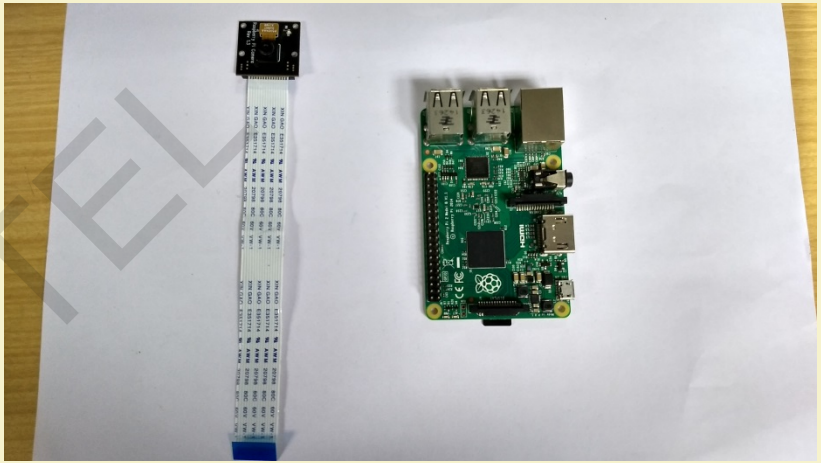
The LED blinks in a loop with delay of 1 and 2 seconds.



Capture Image using Raspberry Pi

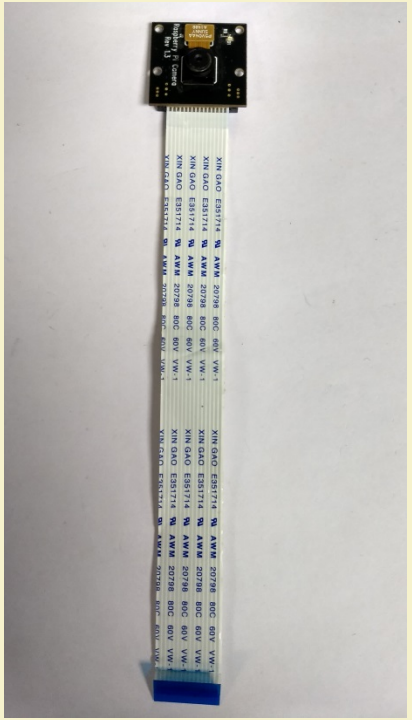
Requirement

- Raspberry Pi
- Raspberry Pi Camera



Raspberry Pi Camera

- Raspberry Pi specific camera module
- Dedicated CSI slot in Pi for connection
- The cable slot is placed between Ethernet port and HDMI port



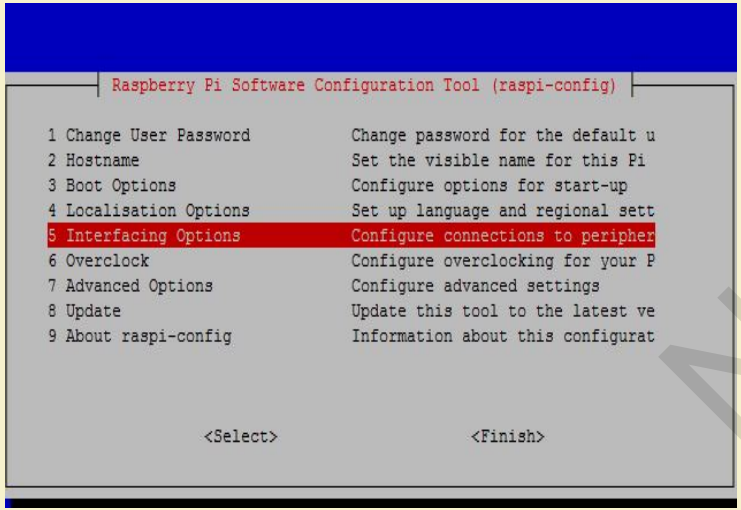
Connection

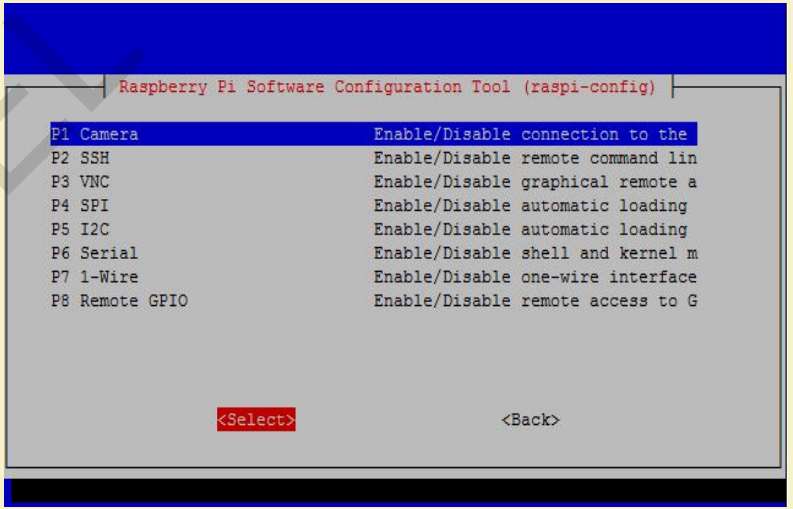
Boot the Pi once the camera is connected to Pi

Configuring Pi for Camera

- In the terminal run the command “sudo raspi-config” and press enter.
- Navigate to “Interfacing Options” option and press enter.
- Navigate to “Camera” option.
- Enable the camera.
- Reboot Raspberry pi.

Configuring Pi for Camera (contd..)





Capture Image

- Open terminal and enter the command-

  raspistill -o image.jpg

- This will store the image as ‘image.jpg’

Capture Image (contd..)

PiCam can also be processed using Python camera module python-picamera

```
sudo apt-get install python-picamera
```

Python Code:

```
Import picamera
camera = picamera.PiCamera()
camera.capture('image.jpg')
```

Source: PYTHON PICAMERA, Raspberry Pi Foundation

Capture Image (contd..)

```
pi@raspberrypi:~ $ raspistill -o image.jpg
pi@raspberrypi:~ $
```

Thank You!!

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

Implementation of IoT with Raspberry Pi: Part 1

Dr. Sudip Misra
Associate Professor
Department of Computer Science and Engineering
IIT KHARAGPUR
Email: smisra@sit.iitkgp.ernet.in
Website: http://cse.iitkgp.ac.in/~smisra/

IOT

Internet Of Things

- Creating an interactive environment
- Network of devices connected together

Sensor

- Electronic element
- Converts physical quantity into electrical signals
- Can be analog or digital

Actuator

- Mechanical/Electro-mechanical device
- Converts energy into motion
- Mainly used to provide controlled motion to other components

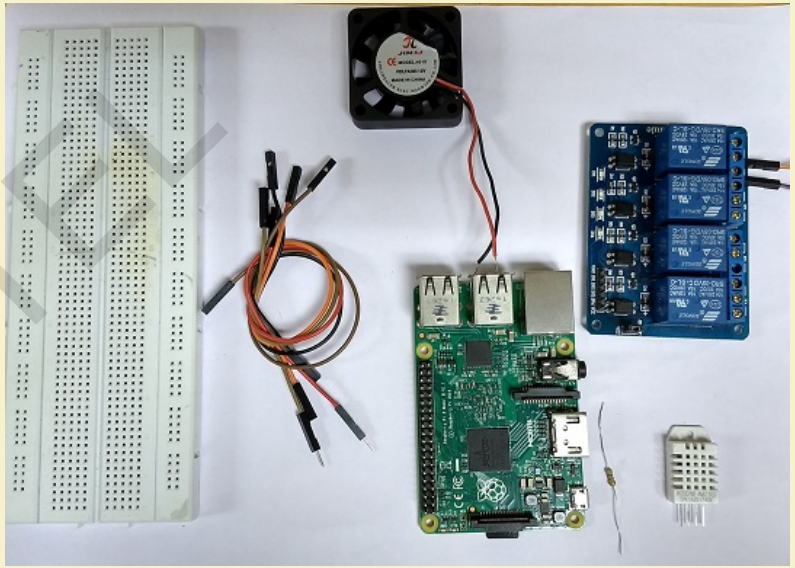
System Overview

- Sensor and actuator interfaced with Raspberry Pi
- Read data from the sensor
- Control the actuator according to the reading from the sensor
- Connect the actuator to a device

System Overview (contd..)

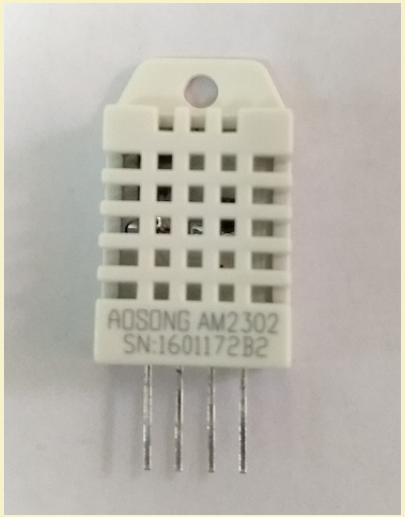
Requirements

- DHT Sensor
- 4.7K ohm resistor
- Relay
- Jumper wires
- Raspberry Pi
- Mini fan



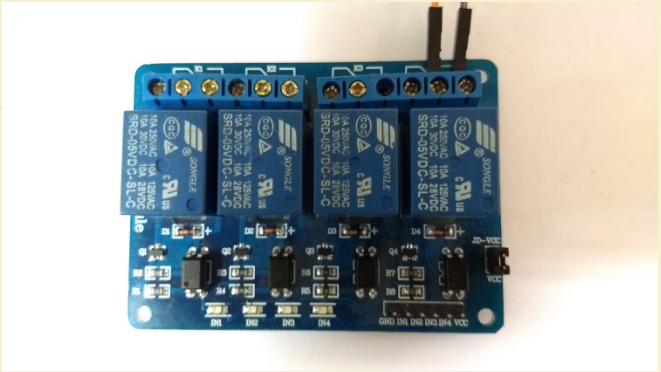
DHT Sensor

- Digital Humidity and Temperature Sensor (DHT)
- PIN 1, 2, 3, 4 (from left to right)
  - PIN 1- 3.3V-5V Power supply
  - PIN 2- Data
  - PIN 3- Null
  - PIN 4- Ground



Relay

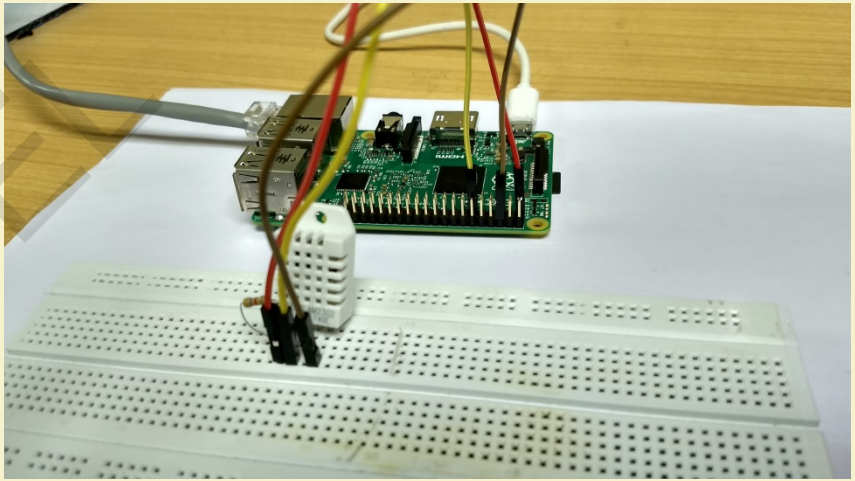
- Mechanical/electromechanical switch
- 3 output terminals (left to right)
  - NO (normal open):
  - Common
  - NC (normal close)



Temperature Dependent Auto Cooling System

Sensor interface with Raspberry Pi

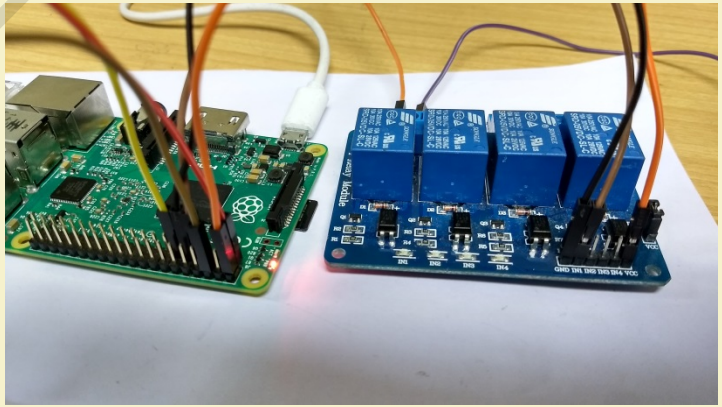
- Connect pin 1 of DHT sensor to the 3.3V pin of Raspberry Pi
- Connect pin 2 of DHT sensor to any input pins of Raspberry Pi, here we have used pin 11
- Connect pin 4 of DHT sensor to the ground pin of the Raspberry Pi



Temperature Dependent Auto Cooling System (contd..)

Relay interface with Raspberry Pi

- Connect the VCC pin of relay to the 5V supply pin of Raspberry Pi
- Connect the GND (ground) pin of relay to the ground pin of Raspberry Pi
- Connect the input/signal pin of Relay to the assigned output pin of Raspberry Pi (Here we have used pin 7)



Temperature Dependent Auto Cooling System (contd..)

Adafruit provides a library to work with the DHT22 sensor

- Install the library in your Pi-
  - Get the clone from GIT
    git clone https://github.com/adafruit/Adafruit_Python_DHT.g...
  - Go to folder Adafruit_Python_DHT
    cd Adafruit_Python_DHT
  - Install the library
    sudo python setup.py install

Source: ADAFRUIT DHTXX SENSORS, Lady Ada, 2012-07-29

Program: DHT22 with Pi

```
import RPi.GPIO as GPIO
from time import sleep
import Adafruit_DHT                                  #importing the Adafruit library

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
sensor = Adafruit_DHT.AM2302                         # create an instance of the sensor type
print ('Getting data from the sensor')
#humidity and temperature are 2 variables that store the values received from the sensor

humidity, temperature = Adafruit_DHT.read_retry(sensor,17)
print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
```

Program: DHT22 interfaced with Raspberry Pi

Code

```
GNU nano 2.2.6                File: IOTSR.py

import RPi.GPIO as GPIO
from time import sleep

import Adafruit_DHT

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

sensor = Adafruit_DHT.AM2302  # create an instance of the sensor type

print ('Getting data from the sensor')

#humidity and temperature are 2 variables that store the values received from the sensor
humidity, temperature = Adafruit_DHT.read_retry(sensor,17)

print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
```

Output

```
pi@raspberrypi:~ $ python IOTSR.py
Getting data from the sensor
Temp=26.1*C humidity=65.9%
pi@raspberrypi:~ $
```

Connection: Relay

- Connect the relay pins with the Raspberry Pi as mentioned in previous slides

- Set the GPIO pin connected with the relay's input pin as output in the sketch

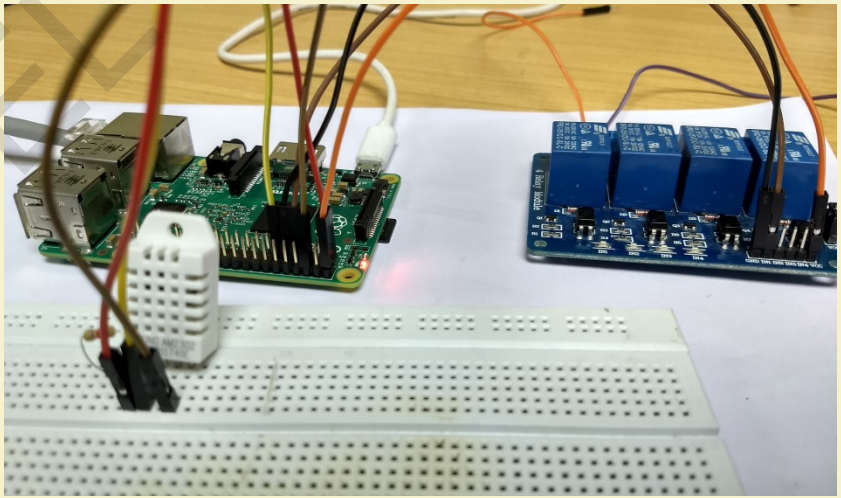
  GPIO.setup(13,GPIO.OUT)

- Set the relay pin high when the temperature is greater than 30

```
if temperature > 30:
    GPIO.output(13,0) # Relay is active low
    print('Relay is on')
    sleep(5)
    GPIO.output(13,1) # Relay is turned off after delay of 5 seconds
```

Connection: Relay (contd..)

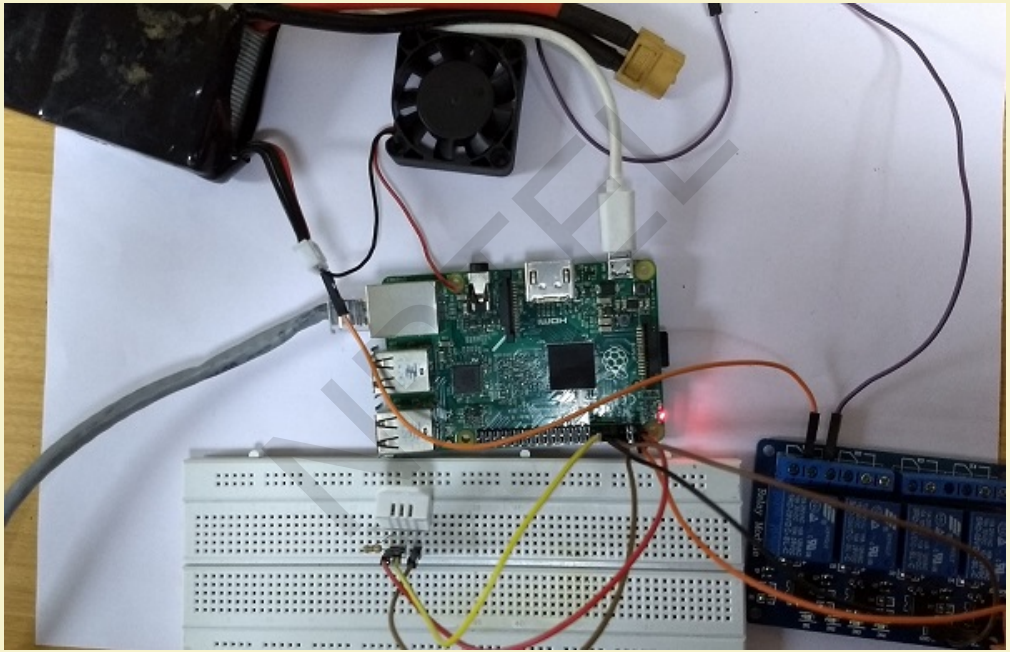
```
  GNU nano 2.2.6                                File: IOTSR.py

import RPi.GPIO as GPIO
from time import sleep

import Adafruit_DHT

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


sensor = Adafruit_DHT.AM2302  # create an instance of the sensor type

print ('Getting data from the sensor')

#humidity and temperature are 2 variables that store the values received from the sensor
humidity, temperature = Adafruit_DHT.read_retry(sensor,17)

print ('Temp={0:0.1f}*C humidity={1:0.1f}%'.format(temperature, humidity))
if temperature > 20:
    GPIO.output(7,0) # Relay is active low
    print('Relay is on')
    sleep(5)
    GPIO.output(7,1) # Relay is turned off after delay of 5 seconds
```



Connection: Fan

- Connect the Li-po battery in series with the fan
  - NO terminal of the relay -> positive terminal of the Fan.
  - Common terminal of the relay -> Positive terminal of the battery
  - Negative terminal of the battery -> Negative terminal of the fan.
- Run the existing code. The fan should operate when the surrounding temperature is greater than the threshold value in the sketch

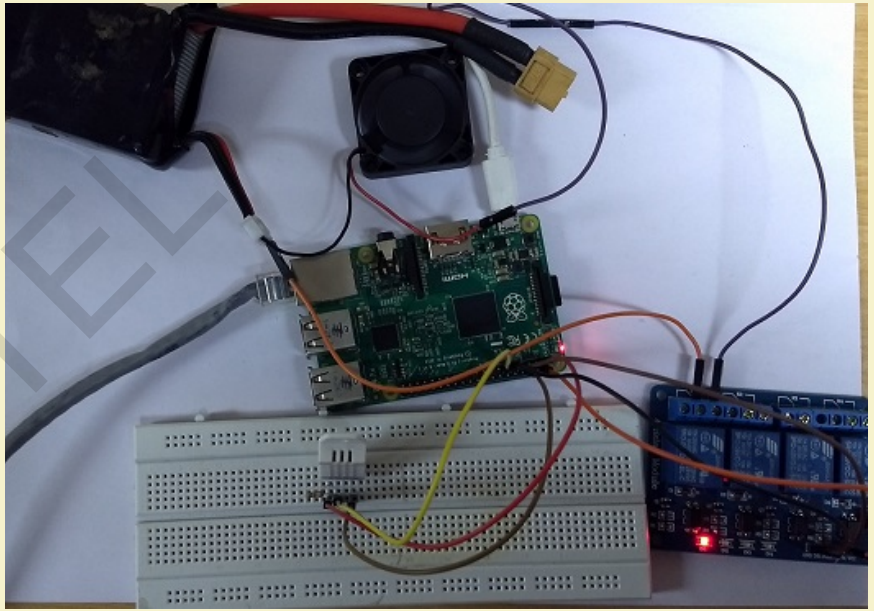
Connection: Fan (contd..)



Result

The fan is switched on whenever the temperature is above the threshold value set in the code.

Notice the relay indicator turned on.



Thank You!!