Experiment: Study of Raspberry-Pi, Beagle board, Arduino, and different operating systems for Raspberry Pi/Beagle board/Arduino. Understanding the process of OS installation on Raspberry Pi/Beagle board/Arduino

Lab setup: Raspberry Pi -3, SD Card, SD card formatter, Raspbian OS latest image, 5 Volts 2 Amperes adapter, VGA /HDMI cable converter

Step by Step Execution:

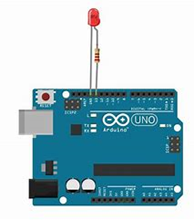
1. Download the latest image of Raspbian OS from [Raspberry Pi OS – Raspberry Pi](https://www.raspberrypi.com/software/)
2. Install Raspberry Pi Imager from the same link.
3. Format the existing SD card using Raspberry Pi Imager and copy the downloaded Raspbian OS with the same tool
4. Post to successful writing of OS over SD card, remove SD card safely from the reader
5. Insert SD card in raspberry pi 3 slot
6. Connect the HDMI port of raspberry pi 3 to monitor using VGA /HDMI cable converter
7. Power on the Raspberry pi 3 using adapter
8. Observe booting process
9. After the successful login set up the time, network and serial port configurations using raspi-config command in terminal

Experiment: Open-source prototype platform- Raspberry-Pi/Beagle board/Arduino -Simple program digital read/write using LED and Switch -Analog read/write using sensor and actuators.

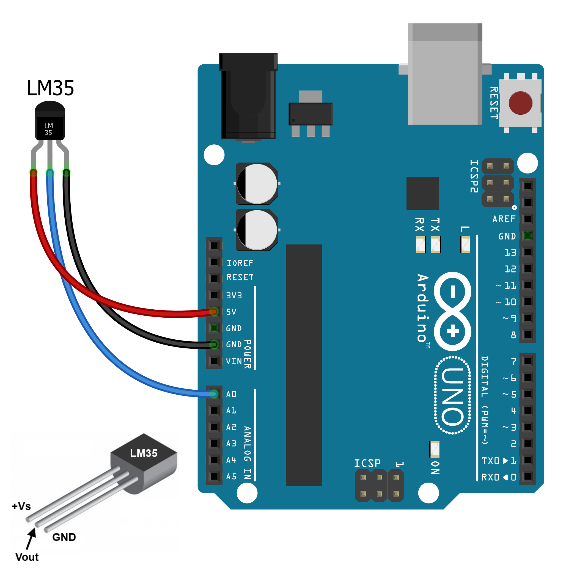
Lab setup: Arduino Uno/ Nano, LED, 220 Ohm resistors, LM 35 sensor, Arduino IDE

Step by Step Execution:

1. Connect LED to GPIO pins of Arduino UNO/Nano with 220 Ohm resistor Ref Link [LED Interface Arduino Uno](https://www.bing.com/images/search?view=detailV2&ccid=hR4qsjfs&id=7A0DC4AA098373281BED64C1713C3627FE2B9B1D&thid=OIP.hR4qsjfs-httuZ9tvto6-QHaKr&mediaurl=https%3a%2f%2f1.bp.blogspot.com%2f-7mXR6vJD_uc%2fVLA-N7ZtUdI%2fAAAAAAAAM3s%2fhhXKp5HUMlc%2fs1600%2fLedBlink_bb.png&exph=1323&expw=918&q=led+blink+arduino&simid=608021851839686634&FORM=IRPRST&ck=7B0A02AB67A079B1E5E6FD7487131952&selectedIndex=6)



1. Connect LM 35 sensor to Arduino Uno/ Nano as shown below. Refer link [LM 35 Arduino Uno Interface](https://beatyourbit.files.wordpress.com/2018/01/lm35-arduino-2.png)



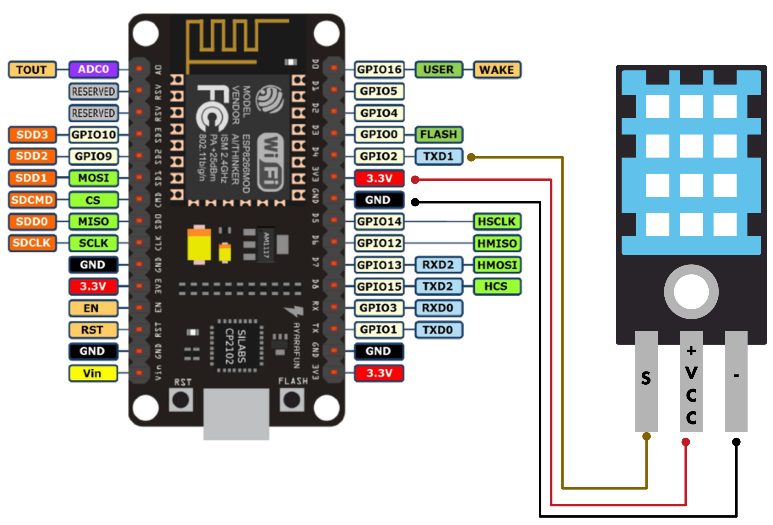
1. Using AnalogRead() function in Arduino start reading temperature from LM35 sensor and display the temperature in degree Celsius over serial port on PC
2. Set the threshold of 36-degree Celsius temperature in programming and turn on LED if temperature crosses the threshold value.

Experiment: Interfacing sensors and actuators with Arduino/Raspberry-pi

Lab setup: ESP8266 Node-MCU, DHT11, 10 K Pull Up resistor, USB to Serial cable, Arduino IDE with ESP8266 libraries

Step by Step Execution:

1. Connect DHT 11 to ESP8266 Node MCU as shown in the below figure. Reference Link [ESP8266 - DHT 11 Interface](https://miro.medium.com/max/5166/1*gIMRuDbEumGs8rN7nu8chg.png)



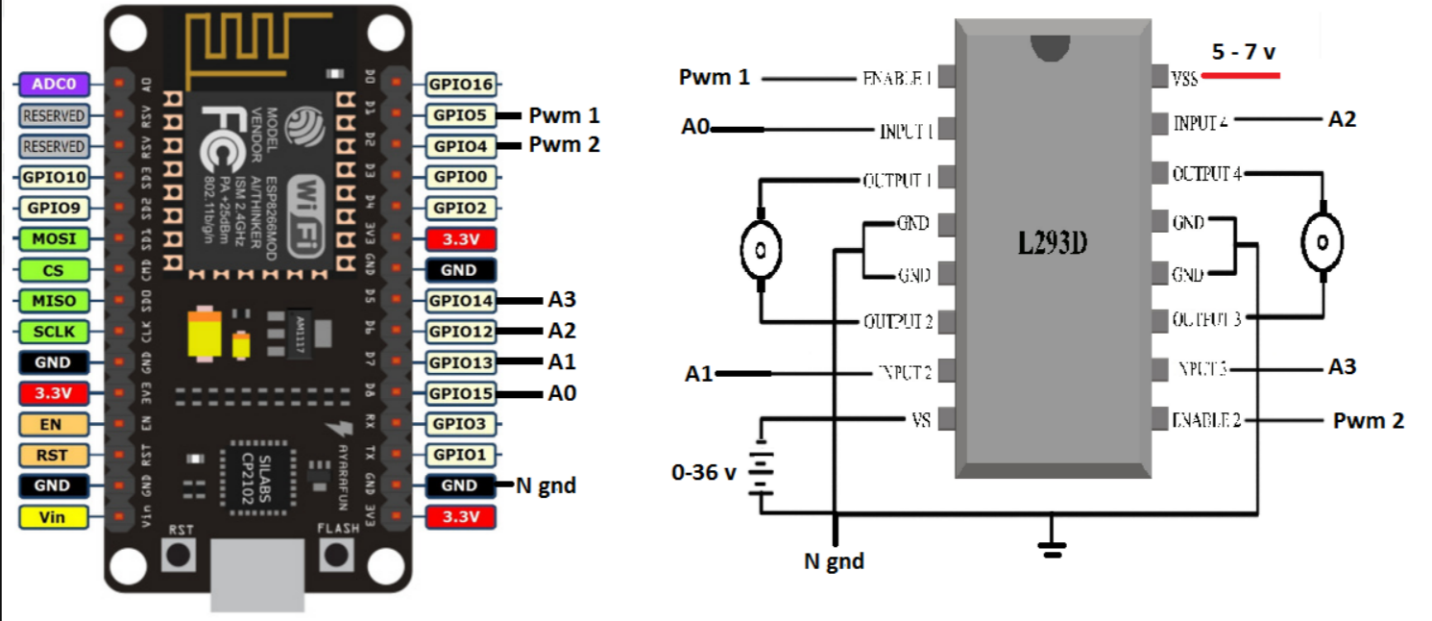
1. Install DHT 11 Arduino Libraries in Arduino IDE
2. Create ESP8266 as access point using SOFTAP () function
3. Add HTML script of Web page for displaying temperature and humidity
4. Read the temperature and humidity values from DHT11, parse the string in separate fields
5. Once system is boot up, check its local IP address over serial port
6. Connect to the access point hosted by ESP8266 and log into HTTP portal of IP address to observe the temperature and humidity

Experiment: IoT based Stepper Motor/DC Motor Control with Arduino/Raspberry Pi

Lab setup: ESP8266, DC motor, L293D

Step by Step Execution:

1. Connect the DC motor to ESP8266 using L293D as shown below. Reference Link [ESP8266 - Motor Control](https://www.engineersgarage.com/nodemcu-and-l293d-motor-driver-controlling-dc-motor/)



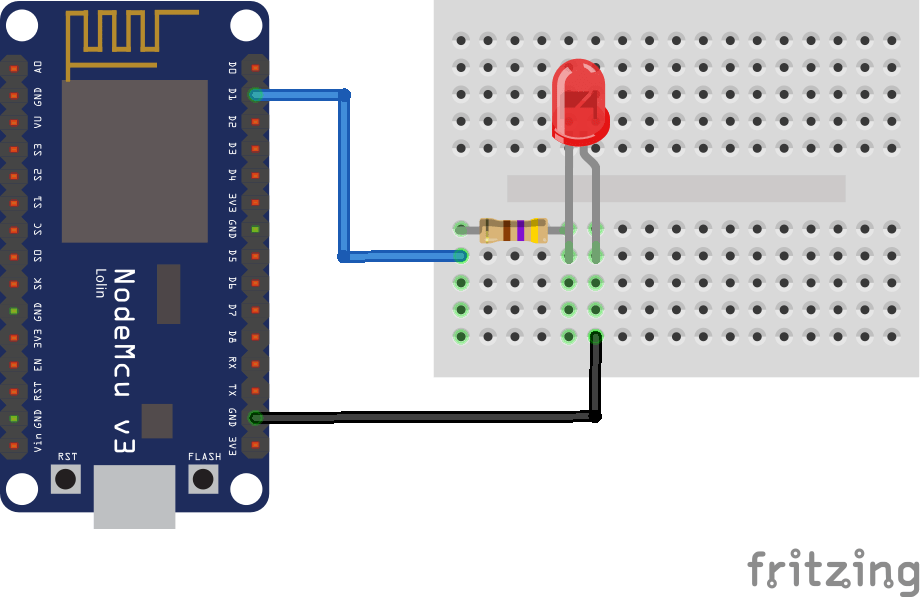
1. Configure ESP8266 as station and connect it to Wi-Fi access point with SSID and Password
2. In the HTML part configure the buttons viz. Start motor, Stop motor, Toggle direction and PWM speed control
3. Once the esp8266 boots up, find out the current IP address over serial port
4. Control the motor as mentioned in the above points

Experiment: Get the status of a bulb at a remote place (on the LAN) through web.

Lab setup: ESP8266 nodemcu, LED, 220 Ohm resistors

Step by Step Execution:

1. Connect simple LED to physical pin D1 and D2 of ESP8266 nodemcu. Reference link [ESP8266 LED](https://www.google.com/url?sa=i&url=http%3A%2F%2Fwww.esp8266learning.com%2Fblink-led-nodemcu.php&psig=AOvVaw2IjerZJPMgnt6MQ_kl1rPl&ust=1667290601387000&source=images&cd=vfe&ved=2ahUKEwizpMTRg4r7AhVVTPUHHSZ_AcwQjRx6BAgAEAw)



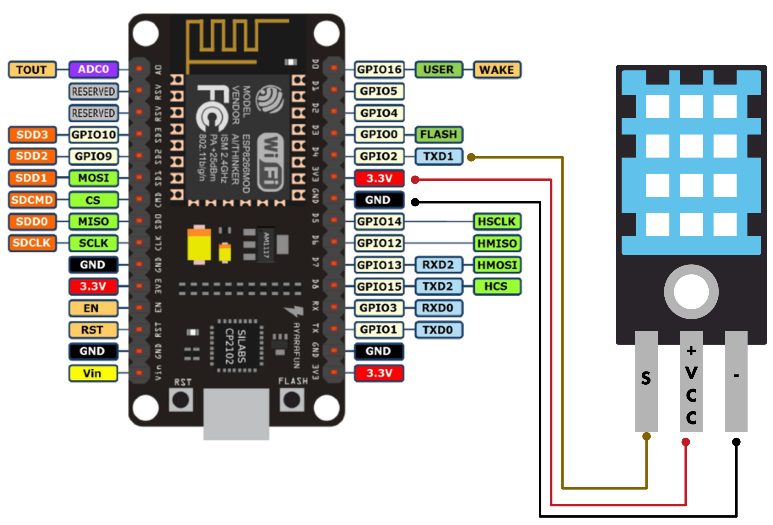
1. Configure ESP8266 as station connected to access point. The hardcoded SSID and Password of access point mentioned in the code.
2. ESP8266 is hosting a server.
3. Once connected to Wi-Fi, the serial port will provide Local IP address of ESP8266
4. Connect your client device either mobile phone or laptop to the same access point, enter the IP address of ESP8266 with port address as 8888 in the browser
5. The secure web browser will ask for username and password which has been base64 encoded in the CSS/HTML scrip of the code with last character as ‘=’
6. Enter base 64 decoded username and password
7. Once web page has been granted the access, control the LED from client and observe the status

Experiment: Introduction to MQTT/ CoAP and sending sensor data to cloud using Raspberry-Pi/Beagle board/Arduino.

Lab setup: ESP8266 Node-MCU, DHT11, 10 K Pull Up resistor, USB to Serial cable, Arduino IDE with ESP8266 libraries

Step by Step Execution:

1. Connect DHT 11 to ESP8266 Node MCU as shown in the below figure. Reference Link [ESP8266 - DHT 11 Interface](https://miro.medium.com/max/5166/1*gIMRuDbEumGs8rN7nu8chg.png)



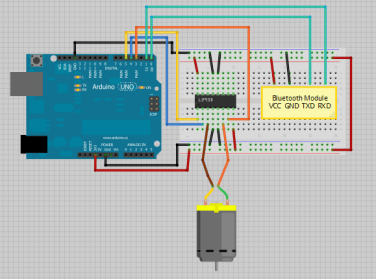
1. Create account on things.io with your valid mail address. Refer the link [ThingsIO set up](https://www.hackster.io/detox/mqtt-with-the-esp8266-on-the-thethings-io-1ba119)
2. Create JSON packet format and get TOKEN/API key from things.io
3. Insert the same in code to configure ESP8266 as MQTT client. The things.io will act as MQTT broker
4. The DHT11 sensor data will be available on MQTT protocol over things.io

Experiment: Interfacing Arduino to Bluetooth Module.

Lab setup: Arduino UNO, HC 05 Bluetooth Module, L293D, DC motor, BlueTerm application over mobile phone

Step by Step Execution:

1. Connect the HC 05 to Arduino UNO as shown in figure below. Reference image [HC05-Arduino Uno](https://i0.wp.com/randomnerdtutorials.com/wp-content/uploads/2013/02/schematics.png?quality=100&strip=all&ssl=1)



1. HC 05 communicates serially to Arduino Uno
2. To control the motor using Bluetooth module send the following commands using Blueterm application from mobile phone
   1. ‘0’ – Turns off the DC motor
   2. 1’ – DC motor rotates to right
   3. ‘2’ – DC motor rotates to left

Experiment: IoT based small project implementation on home automation system.

Problem statement: Students need to design home automation system. The design can be wi-fi based MCU such as ESP8266 connected to access point controlling lights through relays using Web interface. The web can be accessible using port forwarding techniques.

Steps:

1. Students need to select components and create block diagram, schematic and code shall be maintained
2. Hardware can be built either on breadboard or general purpose PCB