

TITLE: Write a program for arithmetic operation in Python.

GPRECD/EEE/IoT(P)-1

Date: 10-08-2020

OBJECTIVE

Write a program for performing arithmetic operations in Python programming using IDLE.

Apparatus:

1. PC (Core i3, 4GB RAM, 500GB HDD).
2. IDLE - 2.0 or 3.0 software.

Algorithm

1. Read inputs using commands `input()` or `raw_input()`.

2. Add two numbers

Syntax :

Add= num1+num2

All the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

3. Subtracting num2 from num1

Syntax :

sub = num1 - num2

4. Multiply num1 with num2

Syntax:

multi = num1 * num2

5. Divide num1 by num2

Syntax:

div = num1 / num2

Prepared By:

Approved By:

Page no. 1 of 2

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for arithmetic operation in Python.

GPRED/EEE/IoT(P)-1

Date: 10-08-2020

6. Modulus of num1 and num2

Syntax:

```
mod = num1 % num2
```

7. Exponent of num1 and num2

Syntax:

```
expo = num1 ** num2
```

8. Print the result of each operation by using print command.

Result:

The output for the different loops appears in editor.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Write a program for looping statements in Python. (Use While, For, If-else) .

GPRECD/EEE/IoT(P)-2

Date: 10-08-2020

OBJECTIVE

Write a program for performing looping statement operations in Python programming using IDLE.

Apparatus:

1. PC (Core i3, 4GB RAM, 500GB HDD).
2. IDLE - 2.0 or 3.0 software.

Algorithm

1. Read two input integer(s) using input() or raw_input().
2. While Loop:

In python, while loop is used to execute a block of statements repeatedly until a given a condition is satisfied. And when the condition becomes false, the line immediately after the loop in program is executed.

Syntax :

while expression:

statement(s)

All the statements indented by the same number of character spaces after a programming construct are considered to be part of a single block of code. Python uses indentation as its method of grouping statements.

3. for - Loop:

For loops are used for sequential traversal. For example: traversing a list or string or array etc. In Python, there is no C style for loop, i.e., for (i=0; i<n; i++). There is “for in” loop which is similar to for each loop in other languages. Let us learn how to use for in loop for sequential traversals.

Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for looping statements in Python. (Use While, For, If-else) .

GPRECD/EEE/IoT(P)-2

Date: 10-08-2020

Syntax:

for iterator_var in sequence:

statements(s)

It can be used to iterate over a range and iterators.

4. Loop Control Statements: Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed. Python supports the following control statements.
 - Continue Statement: It returns the control to the beginning of the loop.
 - Break Statement: It brings control out of the loop
 - Pass Statement: We use pass statement to write empty loops. Pass is also used for empty control statement, function and classes.
5. Print the result of each operation.

Programs may execute

1. Print Prime Numbers
2. Prints all letters except 'e' and 's' in given statement.
3. Find Factorial of a Number

Result:

The output for the different loops appears in editor.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for LED blink using Raspberry Pi.

GPRECD/EEE/IoT(P)-3

Date: 10-08-2020

OBJECTIVE

Write a program for LED blink using Raspberry Pi.

Apparatus:

1. Raspberry Pi board
2. SD card at least 4GB
3. HDMI cable
4. Monitor
5. USB Keyboard
6. USB mouse.
7. LED
8. Connecting Wires

Prerequisites

Python should already be installed on your Raspberry Pi, as it comes standard with Raspbian.

If it is not installed, please install/update it by connecting to internet.

Setting up the circuit

The first step in this project is to design a simple LED circuit. Then we will make the LED circuit controllable from the Raspberry Pi by connecting the circuit to the general purpose input/output (GPIO) pins on the Raspberry Pi.

A simple LED circuit consists of a LED and resistor. The resistor is used to limit the current that is being drawn and is called a current limiting resistor. Without the resistor the LED would run at too high of a voltage, resulting in too much current being drawn which in turn would instantly burn the LED, and likely also the GPIO port on the Raspberry Pi.

Prepared By:

Approved By:

Page no. 1 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for LED blink using Raspberry Pi.

GPRECD/EEE/IoT(P)-3

Date: 10-08-2020

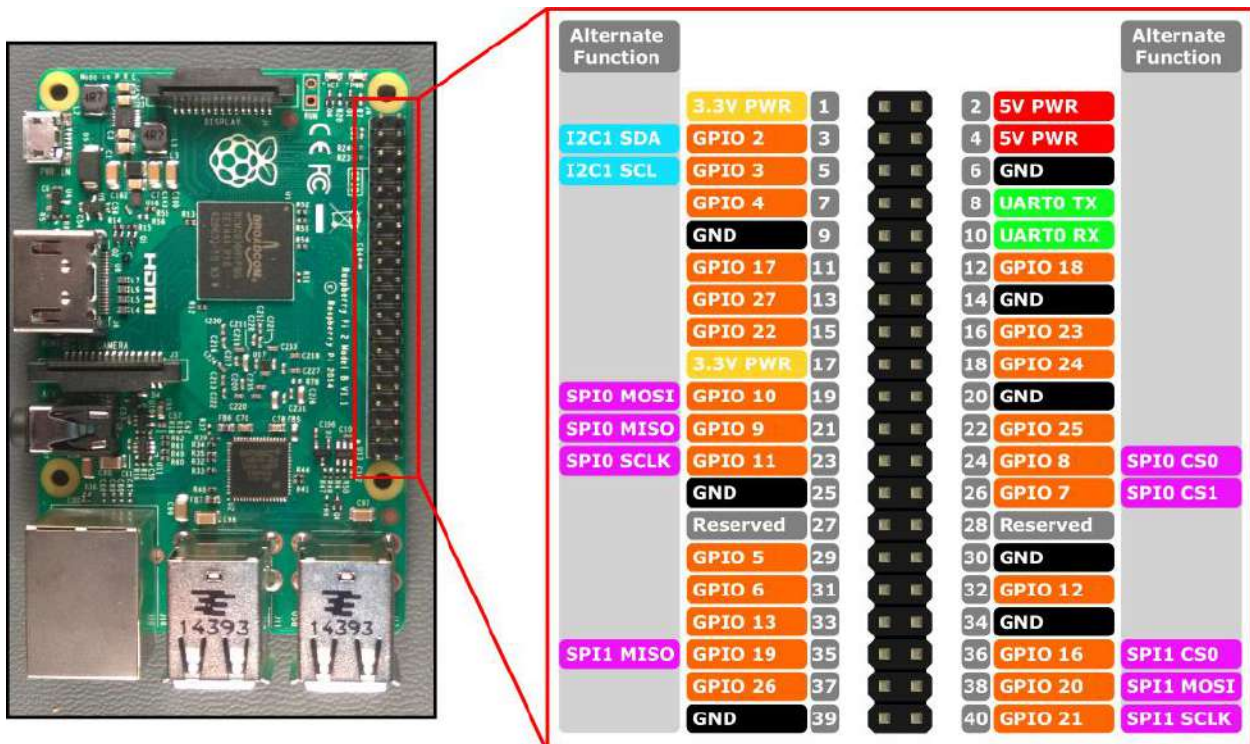
To calculate the resistor value we need to examine the specifications of the LED. Specifically we need to find the forward voltage (VF) and the forward current (IF). A regular red LED has a forward voltage (VF) of 1.7V and forward current of 20mA (IF). Additionally we need to know the output voltage of the Raspberry Pi which is 3.3V.

Calculate the resistor size needed to limit the current to the LED's maximum forward current (IF) using ohm's law like this:

$$R\Omega = \frac{V}{I} = \frac{3.3 - VF}{IF} = \frac{3.3 - 1.7}{20mA} = 80\Omega$$

Unfortunately 80 ohm is not a standard size of a resistor. To solve this it can either combine multiple resistors, or round up to a standard size. In this case it would round up to 100 ohm.

Pin Out diagram of Raspberry pi



Prepared By:

Approved By:

Page no. 2 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

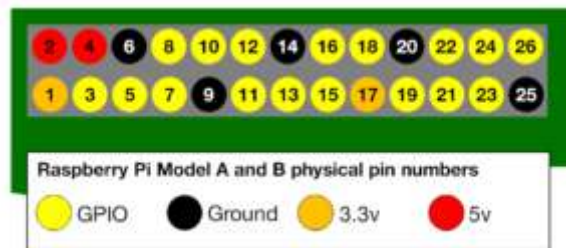
G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for LED blink using Raspberry Pi.

GPRECD/EEE/IoT(P)-3

Date: 10-08-2020

When hooking up the circuit note the polarity of the LED. You will notice that the LED has a long and short lead. The long lead is the positive side also called the anode, the short lead is the negative side called the cathode. The long should be connected to the resistor and the short lead should be connected to ground via the blue jumper wire and pin 6 on the Raspberry Pi as shown on the diagram. To find the pin number refer to pinout diagram showing the physical pin numbers on the Raspberry Pi.



With the circuit created, now to write the Python script to blink the LED. Before start writing the software, first of all install the Raspberry Pi GPIO Python module. This is a library that allows access the GPIO port directly from Python.

To install the Python library open a terminal and execute the following

```
$ sudo apt-get install python-rpi.gpio python3-rpi.gpio
```

Algorithm

1. Import Raspberry Pi GPIO library
2. Import the sleep function from the time module
3. Use physical pin numbering by using setmode
4. Set pin **number** to be an output pin and set initial value to low (off)
5. Turn on for 1 second
6. Sleep for 1 second
7. Turn off for 1 second
8. Sleep for 1 second

Prepared By:

Approved By:

Page no. 3 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Write a program for LED blink using Raspberry Pi.

GPRECD/EEE/IoT(P)-3

Date: 10-08-2020

9. Repeat the loop.
10. Save the program as `blinking_led.py`
11. Run the program by using the command

\$ python blinking_led.py

Result:

LED connected to the Raspberry pi should blink with one second delay.

Prepared By:

Approved By:

Page no. 4 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing of IR sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-4

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing of IR sensor to Raspberry pi.

Apparatus:

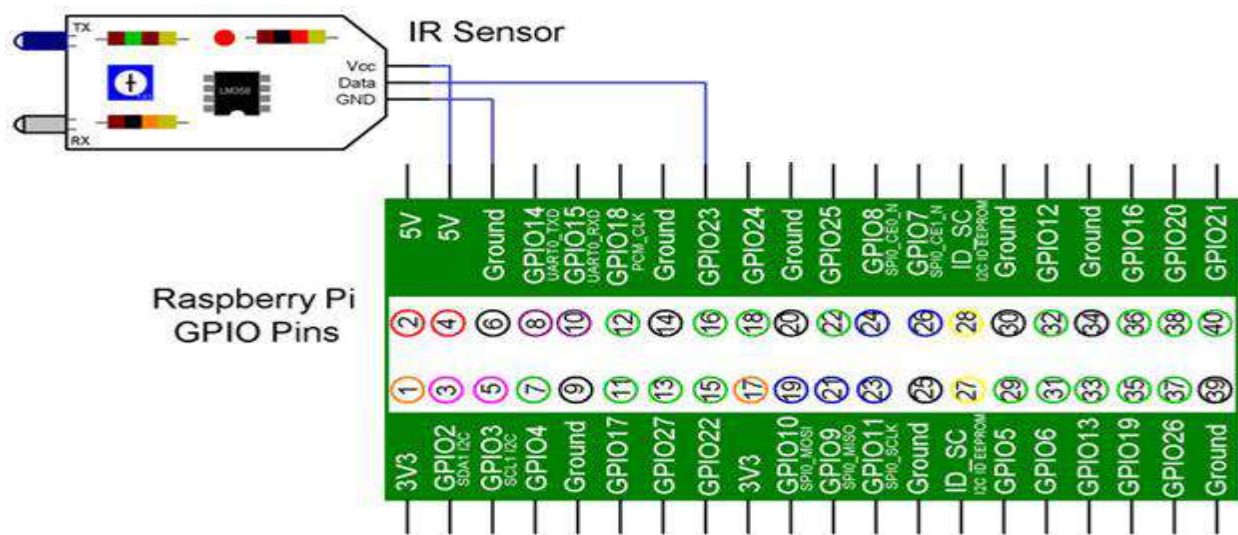
1. Raspberry Pi board
2. SD card at least 4GB
3. HDMI cable
4. Monitor
5. USB Keyboard
6. USB mouse.
7. IR sensor
8. Connecting Wires

Prerequisites

Python should already be installed on your Raspberry Pi, as it comes standard with Raspbian.

If it is not installed, please install/update it by connecting to internet.

Setting up the circuit



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing of IR sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-4

Date: 10-08-2020

The IR Sensor Module has only three Pins: VCC, GND and Data. Connect the VCC and GND pins of the IR Sensor to +5V and GND pins of the Raspberry Pi.

Then connect the Data pin of the IR Sensor to GPIO23 i.e. Physical Pin 16 of the Raspberry Pi.

Algorithm

1. Import Raspberry Pi GPIO library
2. Import the sleep function from the time module
3. Use physical pin numbering by using setmode
4. Set pin *number* to be an input pin and read the value of the sensor output.
5. Delay for 0.5 second
6. If sensor output is high then print the output as object found/detected else no output is printed.
7. Repeat the loop.
8. Save the program as pi_ir.py
9. Run the program by using the command
\$ python pi_ir.py

Result:

Print the output as Object found/detected on the screen when sensor output is High.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement Interfacing of PIR sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-5

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing of PIR sensor to Raspberry pi.

Apparatus:

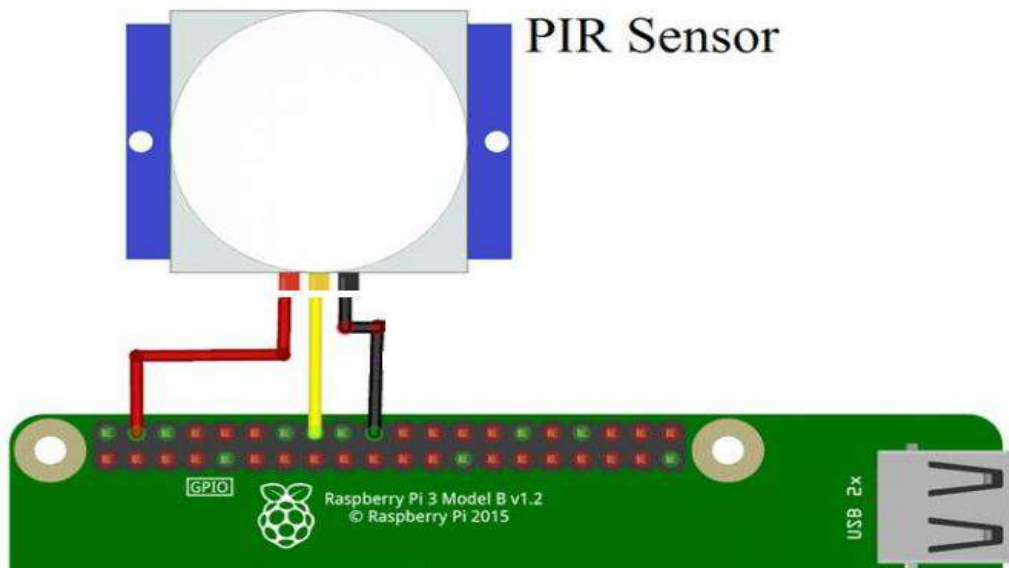
1. Raspberry Pi board
2. SD card at least 4GB
3. HDMI cable
4. Monitor
5. USB Keyboard
6. USB mouse.
7. PIR sensor
8. Connecting Wires

Prerequisites

Python should already be installed on your Raspberry Pi, as it comes standard with Raspbian.

If it is not installed, please install/update it by connecting to internet.

Setting up the circuit



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing of PIR sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-5

Date: 10-08-2020

Connect the VCC and GND pins of the PIR Motion Sensor to +5V and GND pins of the Raspberry Pi. Connect the DATA Pin of the PIR Sensor to GPIO23 i.e. Physical Pin 16 of the Raspberry Pi.

Algorithm

1. Import Raspberry Pi GPIO library
2. Import the sleep function from the time module
3. Use physical pin numbering by using setmode
4. Set pin *number* to be an input pin and read the value of the sensor output.
5. Delay for 0.5 second
6. If sensor output is high then print the output as Motion Detected else no output is printed.
7. Repeat the loop.
8. Save the program as pi_pir.py
9. Run the program by using the command

\$ python pi_pir.py

Result:

Print the output as Motion Detected on the screen when the sensor output is high.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement Interfacing of ultrasonic sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-6

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing of ultrasonic sensor to Raspberry pi.

Apparatus:

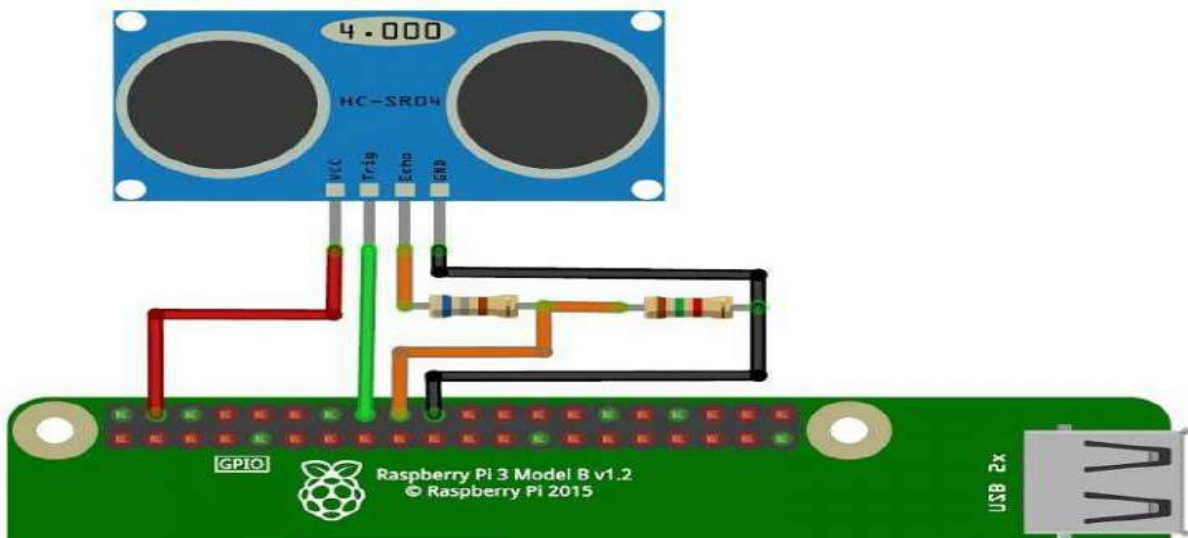
1. Raspberry Pi board
2. SD card at least 4GB
3. HDMI cable
4. Monitor
5. USB Keyboard
6. USB mouse.
7. Ultrasonic sensor
8. Connecting Wires

Prerequisites

Python should already be installed on your Raspberry Pi, as it comes standard with Raspbian.

If it is not installed, please install/update it by connecting to internet.

Setting up the circuit



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing of ultrasonic sensor to Raspberry pi.

GPRECD/EEE/IoT(P)-6

Date: 10-08-2020

Connect the Trig Pin of the HC-SR04 Ultrasonic Sensor to the Physical Pin 16 i.e. GPIO23 of the Raspberry Pi. Use a combination of 680Ω and 1.5 KΩ Resistor to convert the Echo pin to 3.3V Logic (approximately) and connect it to Physical Pin 18 i.e. GPIO24 of the Raspberry Pi.

Algorithm

1. Import Raspberry Pi GPIO library
2. Import the sleep function from the time module
3. Use physical pin numbering by using setmode
4. Set pin **number** to be an input pin and read the value of the sensor output.
5. Delay for 0.5 second
6. If sensor output is high then print the output as distance in centimeters on the screen else no output is printed.
7. Repeat the loop.
8. Save the program as pi_us.py
9. Run the program by using the command
\$ python pi_us.py

Result:

Print the distance in centimeters on the screen when the sensor output is high.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Zigbee Protocol using Raspberry Pi.

GPRECD/EEE/IoT(P)-7

Date: 10-08-2020

OBJECTIVE

Study and Implement Zigbee Protocol using Raspberry Pi.

Apparatus:

1. Raspberry Pi board
2. SD card at least 4GB
3. HDMI cable
4. Monitor
5. USB Keyboard
6. USB mouse.
7. XBee 1mW Wire Antenna- Series 1 2 No
8. Connecting Wires

Prerequisites

Python should already be installed on your Raspberry Pi, as it comes standard with Raspbian.

If it is not installed, please install/update it by connecting to internet.

Setting up the circuit



Attach the two Zigbee's to the two dongle's and connect one pair on the USB port of the Raspberry Pi2. Connect the other pair to the USB port of a computer or a Laptop. Install the

Prepared By:

Approved By:

Page no. 1 of 3

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

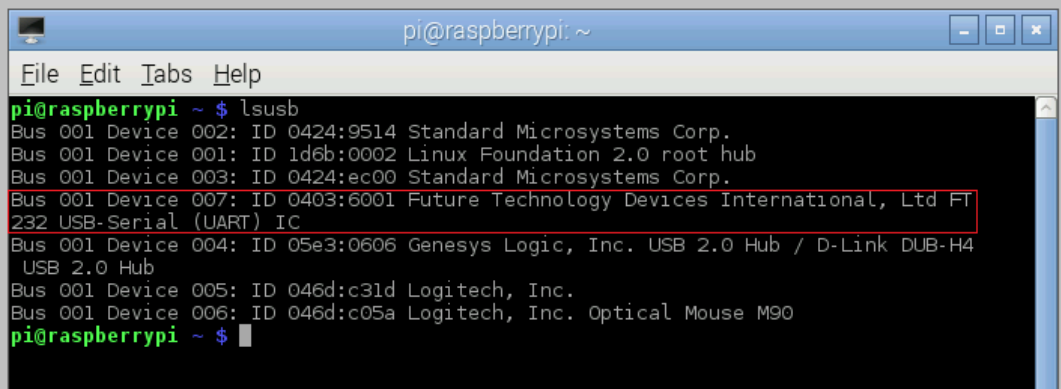
G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Zigbee Protocol using Raspberry Pi.

GPRECD/EEE/IoT(P)-7

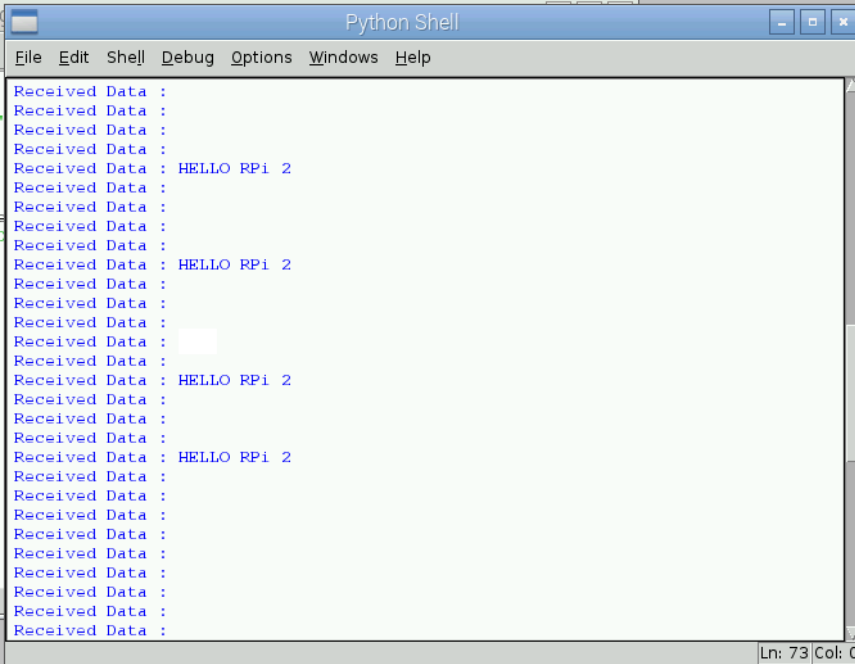
Date: 10-08-2020

FTDI Driver on the Computer /laptop to provide usb communication. Create a hyperterminal, select the corresponding com port and the baudrate for the connected device.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi ~ $ lsusb  
Bus 001 Device 002: ID 0424:9514 Standard Microsystems Corp.  
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub  
Bus 001 Device 003: ID 0424:ec00 Standard Microsystems Corp.  
Bus 001 Device 007: ID 0403:6001 Future Technology Devices International, Ltd FT232 USB-Serial (UART) IC  
Bus 001 Device 004: ID 05e3:0606 Genesys Logic, Inc. USB 2.0 Hub / D-Link DUB-H4 USB 2.0 Hub  
Bus 001 Device 005: ID 046d:c31d Logitech, Inc.  
Bus 001 Device 006: ID 046d:c05a Logitech, Inc. Optical Mouse M90  
pi@raspberrypi ~ $
```

The two zigbee must be in a line of sight and check the results in the Python shell and in the hyper terminal of the computer.



```
Python Shell  
File Edit Shell Debug Options Windows Help  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data : HELLO RPi 2  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data : HELLO RPi 2  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data : HELLO RPi 2  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Received Data :  
Ln: 73 Col: 0
```

Prepared By:

Approved By:

Page no. 2 of 3

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Zigbee Protocol using Raspberry Pi.

GPRECD/EEE/IoT(P)-7

Date: 10-08-2020

Algorithm

1. Import Raspberry Pi GPIO library
2. Import the sleep function from the time module
3. Import serial communication.
4. Enable USB serial communication
5. Send “Hello RPI 2” as output when the combination is detected using write command.
6. Repeat the loop.
7. Save the program as pi_zig.py
8. Run the program by using the command

\$ python pi_zig.py

Result:

On python shell Hello RPI2 is printed.

Prepared By:

Approved By:

Page no. 3 of 3

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing an LED to NodeMCU.

GPRECD/EEE/IoT(P)-8

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing an LED to NodeMCU.

Apparatus:

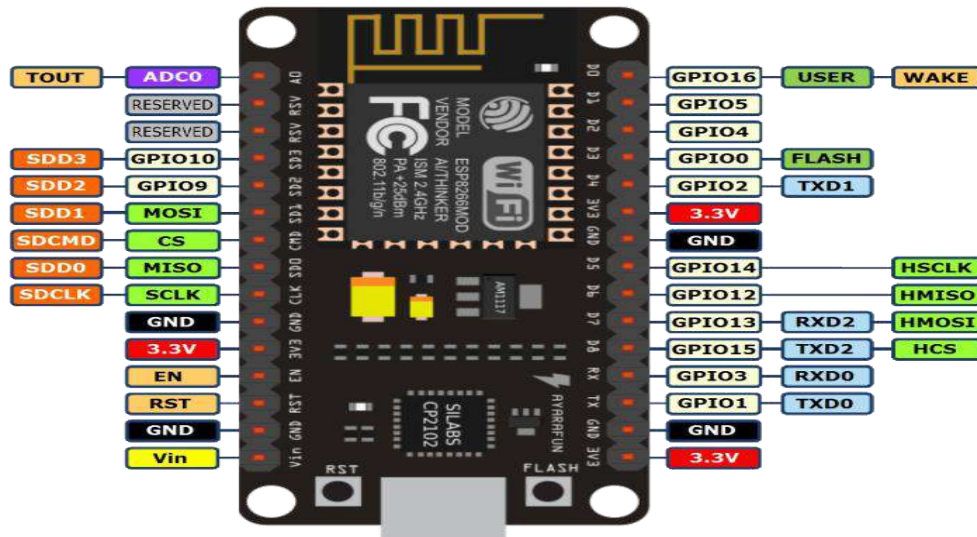
1. NodeMCU ESP8266
2. LED
3. Resistor 220 Ohms
4. Breadboard
5. Jumper Wire
6. Micro USB cable

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit

Anode of the LED is connected to the D1 pin of the NodeMCU, the cathode of the LED is connected with the one terminal of the resistor and another terminal of the resistor is connected to the ground pin.



Pinout Diagram of Node MCU

Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing an LED to NodeMCU.

GPRECD/EEE/IoT(P)-8

Date: 10-08-2020

Algorithm

1. Assign LED pin i.e: D1 on NodeMCU
2. Initialize GPIO 5 as an output using is pin MODE
3. Turn the LED on using digitalWrite command.
4. Provide Delay for 1000 milliseconds.
5. Turn the LED off using digitalWrite command.
6. Provide Delay for 1000 milliseconds.
7. Setup the loop function runs over and over again forever.
8. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module").

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

9. Run the program on Arduino IDE.

Result:

LED connected to the NodeMCU should blink with one second delay.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement LED brightness controller with NodeMCU.

GPRECD/EEE/IoT(P)-9

Date: 10-08-2020

OBJECTIVE

Study and Implement LED brightness controller with NodeMCU.

Apparatus:

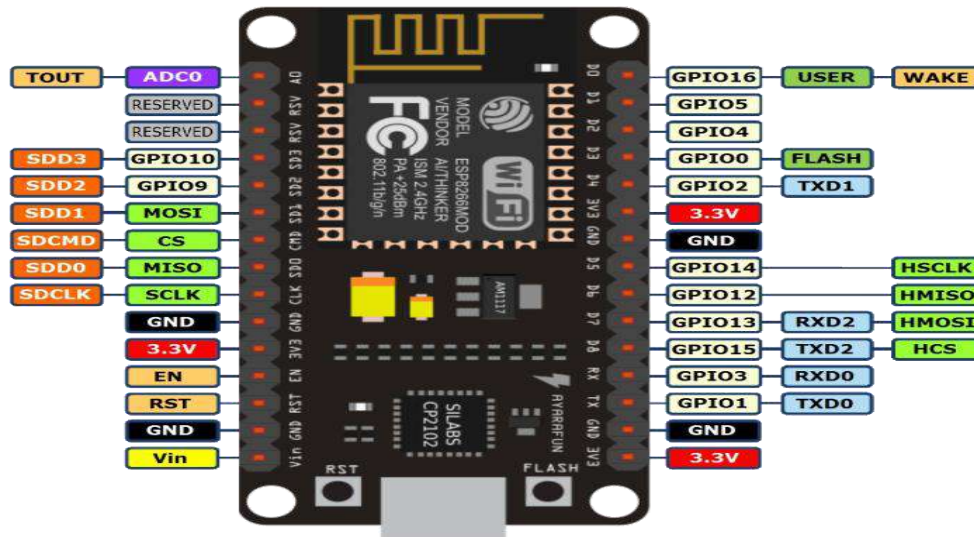
1. NodeMCU ESP8266
2. LED
3. Resistor 220 Ohms
4. Breadboard
5. Jumper Wire
6. Micro USB cable

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit

Anode of the LED is connected to the D4 pin of the NodeMCU, the cathode of the LED is connected with the one terminal of the resistor and another terminal of the resistor is connected to the ground pin.



Pinout Diagram of Node MCU

Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement LED brightness controller with NodeMCU.

GPRECD/EEE/IoT(P)-9

Date: 10-08-2020

Algorithm

1. Assign LED pin i.e: D4 on NodeMCU
2. Initialize GPIO as output using is pin MODE
3. In the loop(), vary the duty cycle between 0 and 1023 to increase the LED brightness.
4. To set the LED brightness use analogWrite() function.
5. Provide Delay for 1 milliseconds.
6. In the loop(), vary the duty cycle between 1023 and 0 to decrease the LED brightness.
7. Provide Delay for 1 milliseconds.
8. Setup the loop function runs over and over again forever.
9. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module") .

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

10. Run the program on Arduino IDE.

Result:

LED connected to the NodeMCU should vary the brightness with one second delay.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing the temperature and humidity sensor to NodeMCU.

GPRECD/EEE/IoT(P)-10

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing the temperature and humidity sensor to NodeMCU.

Apparatus:

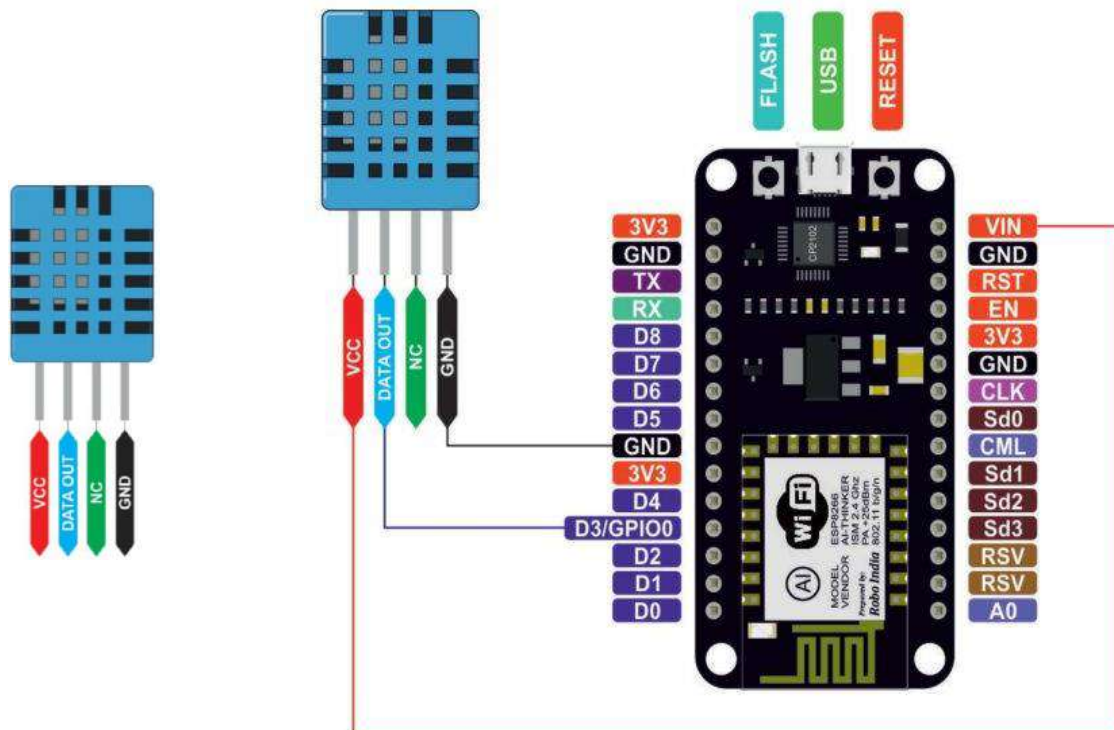
1. NodeMCU ESP8266
2. DHT11 sensor.
3. Jumper Wire
4. Micro USB cable

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit

DHT11 sensor gives humidity and temperature data. It has got following pin interface.



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing the temperature and humidity sensor to NodeMCU.

GPRECD/EEE/IoT(P)-10

Date: 10-08-2020

Algorithm

1. including the library of DHT11 temperature and humidity sensor
2. Assign LED pin i.e: D3 on NodeMCU
3. Initialize GPIO as output using is pin MODE
4. Setup the serial monitor at baud rate 9600
5. In the loop(), print the humidity and temperature using serial print commands
6. Provide Delay for 600 milliseconds.
7. Setup the loop function runs over and over again forever.
8. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module").

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

9. Run the program on Arduino IDE.

Result:

Humidity and temperature should be displayed on serial monitor.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement Interfacing the moisture sensor to NodeMCU.

GPRECD/EEE/IoT(P)-11

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing the moisture sensor to NodeMCU.

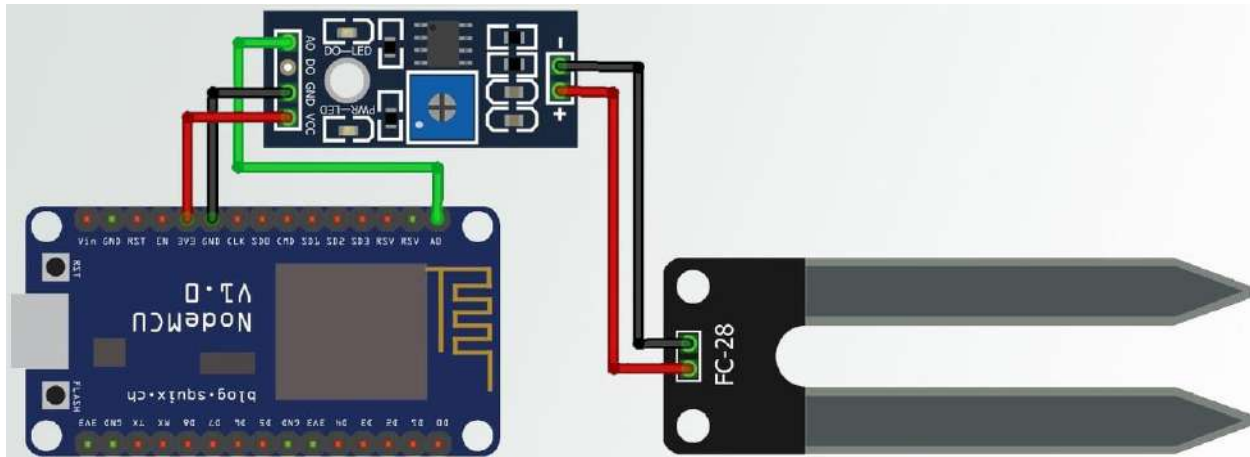
Apparatus:

1. NodeMCU ESP8266
2. Moisture sensor.
3. Jumper Wire.
4. Micro USB cable.

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit



The soil moisture sensor output is analog mode. So, connect the analog output pin to A0 of NodeMCU. So, connect its SDA SCL pin to D2 & D1 of NodeMCU. Both the OLED & Soil Moisture Sensor work at 3.3V. So connect their VCC pin to 3.3V of NodeMCU.

Algorithm

1. Assign input pin i.e: A0 on NodeMCU
2. Initialize GPIO as output using is pin MODE
3. Setup the serial monitor at baud rate 9600
4. In the loop(), print the moisture using serial print commands

Prepared By:

Approved By:

Page no. 1 of 2

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing the moisture sensor to NodeMCU.

GPRECD/EEE/IoT(P)-11

Date: 10-08-2020

5. Provide Delay for 600 milliseconds.
6. Setup the loop function runs over and over again forever.
7. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module").

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

8. Run the program on Arduino IDE.

Result:

Moisture of the soil should be displayed on serial monitor.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement to control the speed of a DC motor using NodeMCU.

GPRECD/EEE/IoT(P)-12

Date: 10-08-2020

OBJECTIVE

Study and Implement to control the speed of a DC motor using NodeMCU.

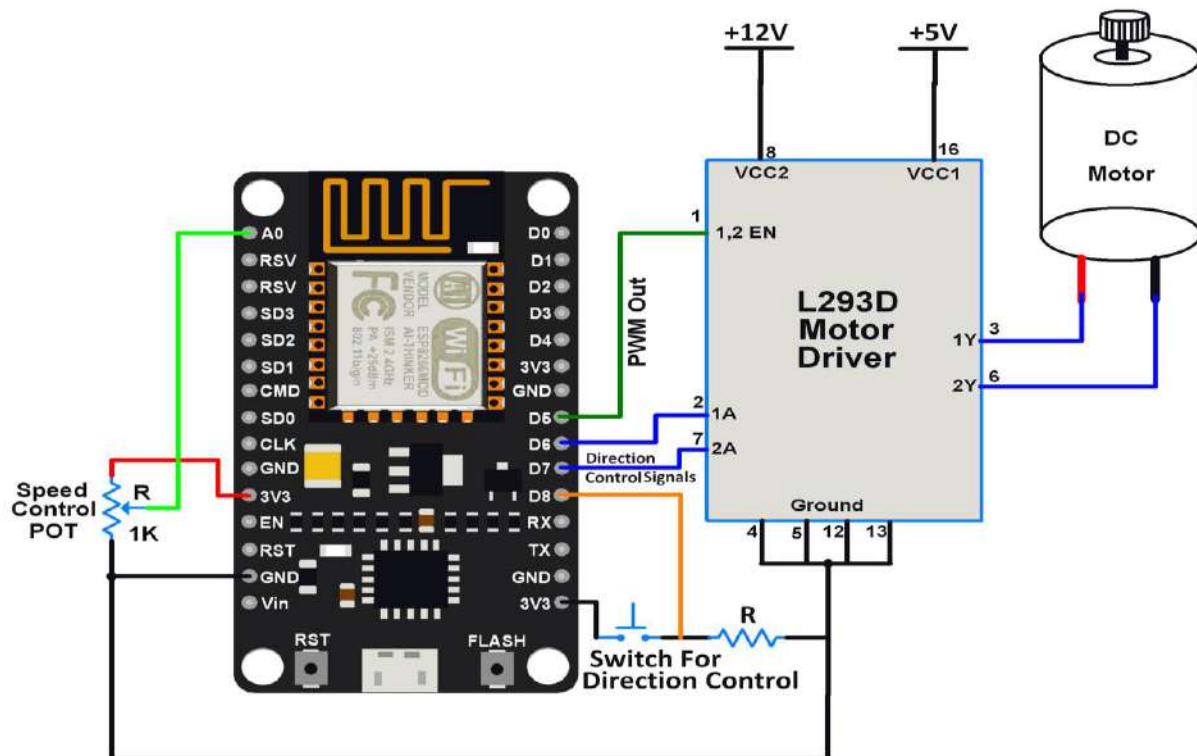
Apparatus:

1. NodeMCU ESP8266
2. DC Motor.
3. L293 Driver.
4. RPS-DC Supply.
5. Jumper Wire.
6. Micro USB cable.

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement to control the speed of a DC motor using NodeMCU.

GPRECD/EEE/IoT(P)-12

Date: 10-08-2020

Algorithm

1. Assign input pin i.e: A0 and D8 on NodeMCU
2. Assign Output pins i.e: D6,D7.
3. Set PWM pin as D5.
4. Initialize GPIO as input/output using is pin MODE.
5. Set PWM frequency
6. Set PWM duty cycle in between 0-1023.
7. Write direction control pin initial states.
8. Read pot using ADC
9. Limit PWM to max of duty cycle
10. Send the PWM to motor pins
11. Provide Delay for 200 milliseconds.
12. Setup the loop function runs over and over again forever.
13. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module") .

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

14. Run the program on Arduino IDE.

Result:

Motor should be run on either direction based on the push button selection.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement to control the speed of a DC motor using NodeMCU.

GPRECD/EEE/IoT(P)-12

Date: 10-08-2020

Prepared By:

Approved By:

Page no. 3 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement Interfacing Ultrasonic Sensor with Blynk and NodeMCU.

GPRECD/EEE/IoT(P)-13

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing Ultrasonic Sensor with Blynk and NodeMCU.

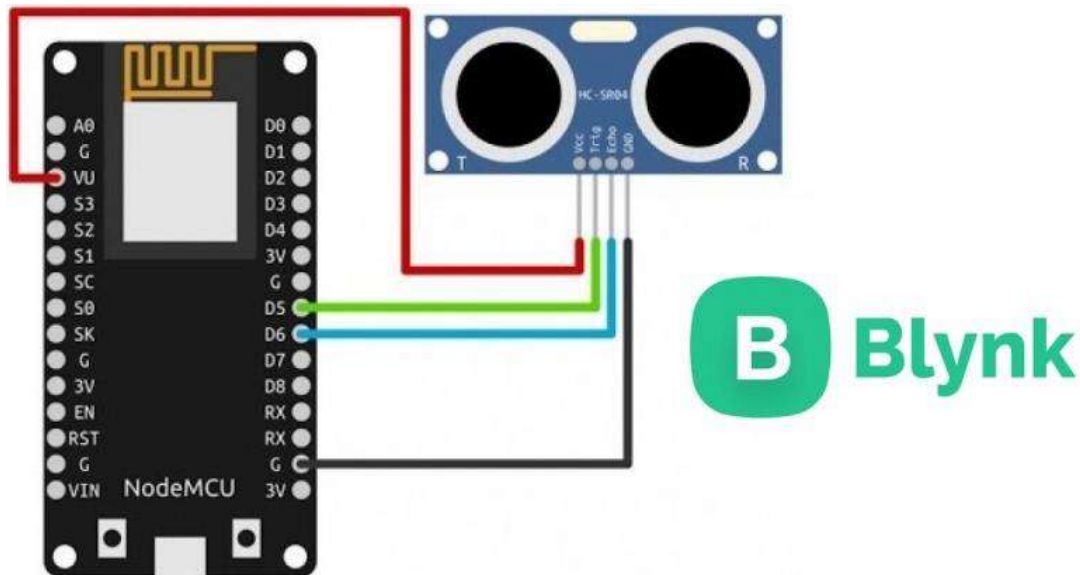
Apparatus:

1. NodeMCU ESP8266
2. Ultrasonic sensor.
3. Smartphone/Tab.
4. Jumper Wire.
5. Micro USB cable.

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit



Setting Up the Blynk App:

create a Project use Blynk apps Following Instruction, First LogIn your Blynk App Then Flow the Step By Step.

Prepared By:

Approved By:

Page no. 1 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing Ultrasonic Sensor with Blynk and NodeMCU.

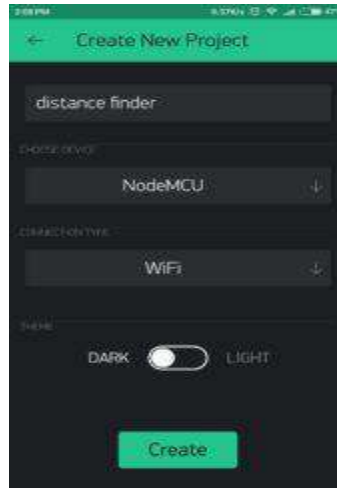
GPRECD/EEE/IoT(P)-13

Date: 10-08-2020

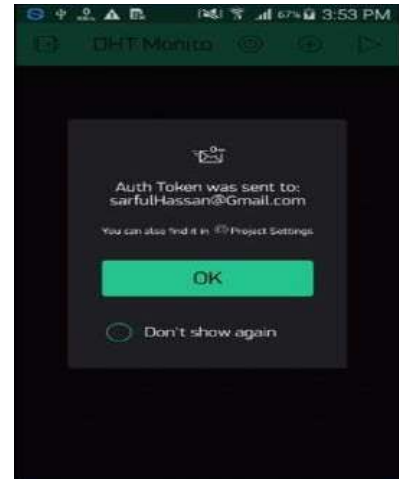
Step1 -Create A New Project



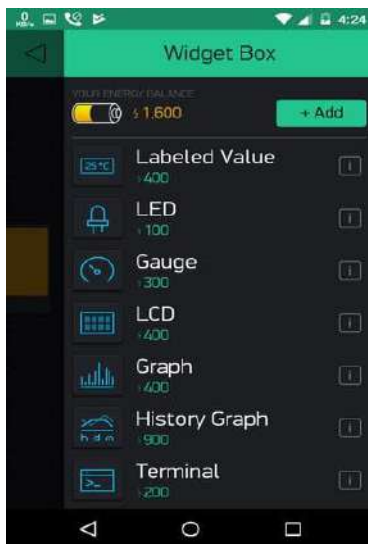
Step2- Select Your project name and Board then Press Create.



Step3 -This is very important Step Hear you got your AUTH TOKEN it send to your email address. Press OK



Step4-Now go to Widget Box And select LCD



Step5- Go to LCD Settings Turn the switch from simple to advanced and the input variable to "V1".Set to colours to "Text"



Prepared By:

Approved By:

Page no. 2 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

TITLE: Study and Implement Interfacing Ultrasonic Sensor with Blynk and NodeMCU.

GPRECD/EEE/IoT(P)-13

Date: 10-08-2020

Firstly the BLYNK_PRINT Serial will define which serial port will be used for the Blynk Debug later in the background. Then the header file <ESP8266Wifi> will utilize all the functions needed for operating the NodeMCU over the internet. Then the header <BlynkSimpleEsp8266> is added for the working of Blynk app with the NodeMCU(Esp8266) over Wifi.

Then, the authentication token is inserted in the code, it is required to ensure security as to make sure that the hardware connected with the PC is communicating with the correct Blynk Project and its user. Afterwards your network credentials (ssid and password) are needed to establish link between the hardware and the Blynk app over your network.

WidgetLCD lcd(v1) is used to define a lcd widget in the blink app and define it to variable v1 Then the D5 and D6 pin of the NodeMCU are defined as trigPin and echoPin respectively, as constants with integer data type.

Algorithm

1. Pin modes for trigPin and echoPin are set.
2. Serial communication between the PC and the NodeMCU is initialized at a baud rate of 9600 bits per second.
3. Blynk.begin is used to start communicating with blynk app.
4. firstly the trigPin is cleared by setting it LOW.
5. Delay of 2 milli-seconds
6. firstly the trigPin is set to HIGH for 10 milli-seconds
7. The <Blynk.run()> function ensures continuous communication between the Blynk app and the NodeMCU.
8. Dump the code to NodeMCU.

In your Arduino IDE, go to **Tools > Board** and select your ESP8266 model (If you're using an ESP-01, select "Generic ESP8266 Module").

Go to **Tools > Port** and select COM port the ESP8266 is connected to.

Prepared By:

Approved By:

Page no. 3 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing Ultrasonic Sensor with Blynk and NodeMCU.

GPRECD/EEE/IoT(P)-13

Date: 10-08-2020

If you're using an ESP-01, you need an FTDI programmer or Serial Adapter to upload code.

9. Run the program on Arduino IDE.

Result:

Distance between the ultrasonic sensor and the object placed in front of it in centimeters in blynk app.

Prepared By:

Approved By:

Page no. 4 of 4

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No: 0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing an LED to ESP32.

GPRECD/EEE/IoT(P)-14

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing an LED to ESP32.

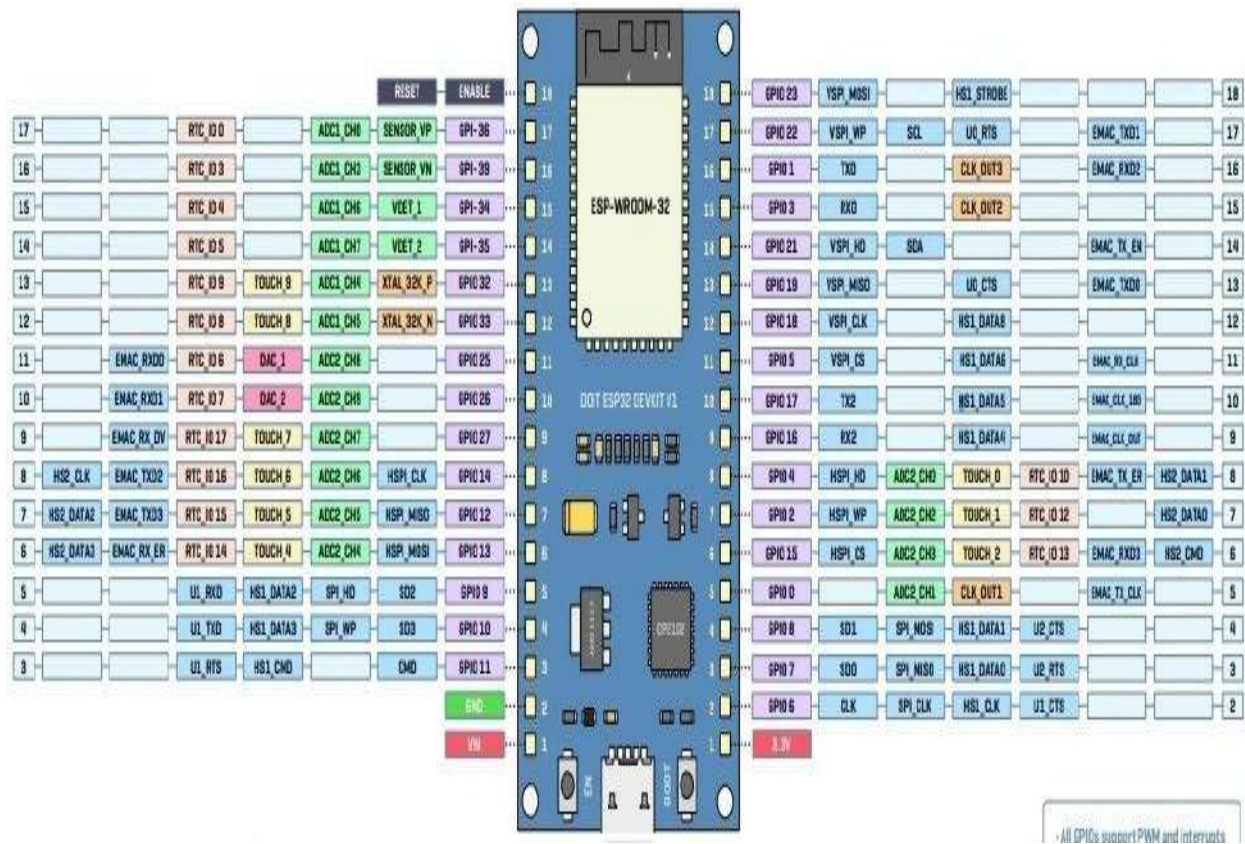
Apparatus:

1. ESP32
2. LED
3. Bread Board
4. Jumper Wire
5. Micro USB cable

Prerequisites

Arduino IDE should already be installed on PC.

Pin out for ESP 32



Prepared By:

Approved By:

Page no. 1 of 2

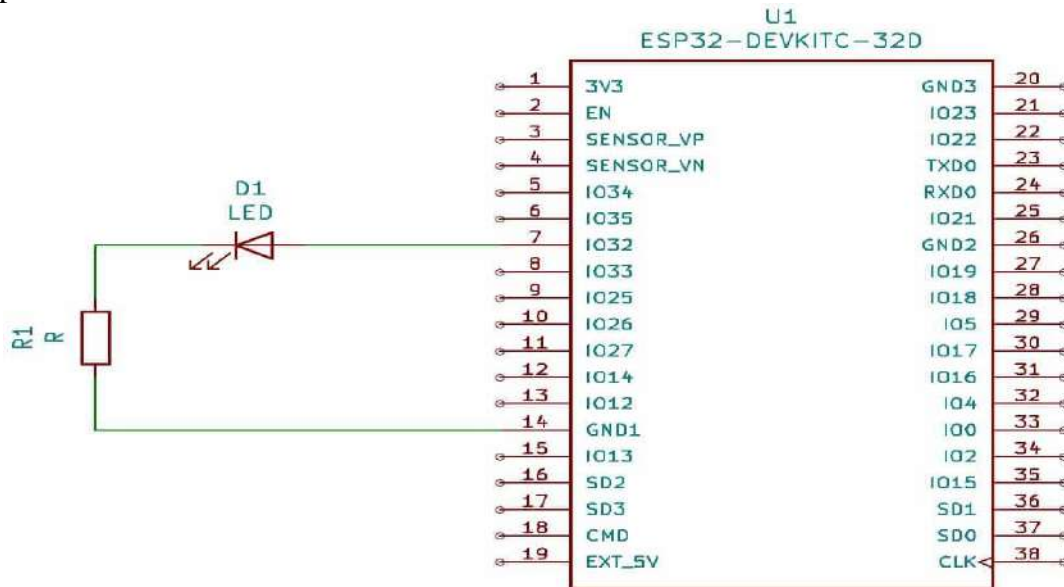
TITLE: Study and Implement Interfacing an LED to ESP32.

GPRECD/EEE/IoT(P)-14

Date: 10-08-2020

Setting up the circuit

a single LED connected in series to a 230 Ω resistor. Connect the anode of the LED to one of the GPIOs on the ESP32, and the cathode to the blue rail, which is connected to one of the GND pins of the ESP32 dev kit.



Algorithm

1. Assign LED pin on ESP 32
2. Initialize GPIO as output using is pin MODE
3. Turn the LED on using digitalWrite command.
4. Provide Delay for 1000 milliseconds.
5. Turn the LED off using digitalWrite command.
6. Provide Delay for 1000 milliseconds.
7. Setup the loop function runs over and over again forever.
8. Dump the code to ESP32.
9. Run the program on Arduino IDE.

Result:

LED connected to the ESP32 should blink with one second delay.

Prepared By:

Approved By:

Page no. 2 of 2

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing the temperature and humidity sensor to ESP32.

GPRECD/EEE/IoT(P)-15

Date: 10-08-2020

OBJECTIVE

Study and Implement Interfacing the temperature and humidity sensor to ESP32.

Apparatus:

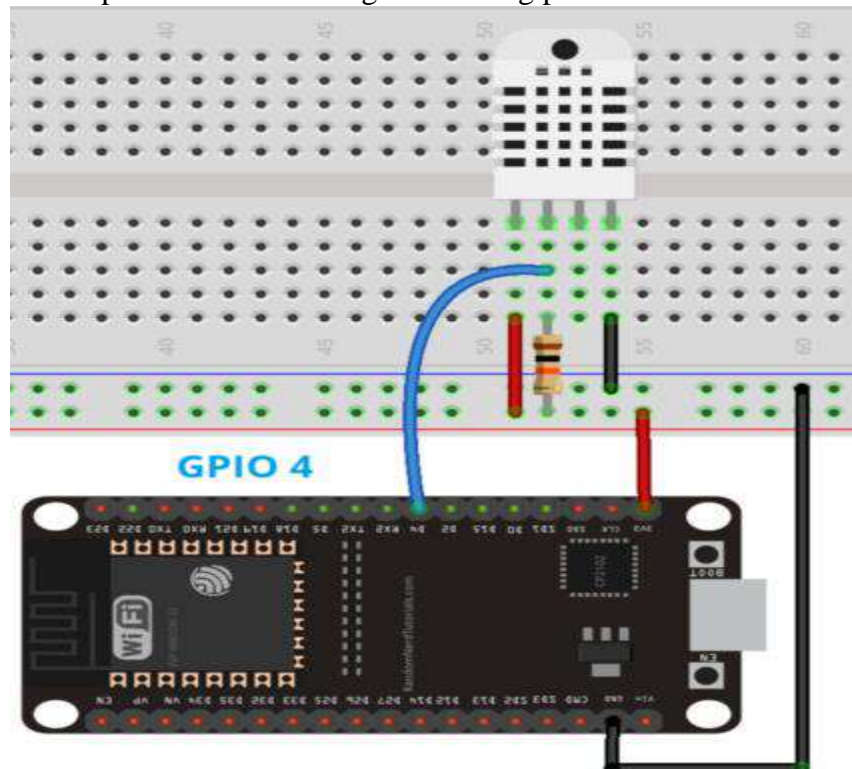
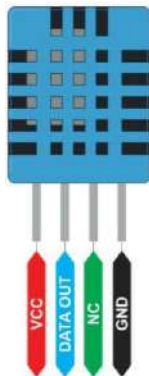
1. ESP32
2. DHT11 sensor.
3. Jumper Wire
4. Micro USB cable

Prerequisites

Arduino IDE should already be installed on PC.

Setting up the circuit

DHT11 sensor gives humidity and temperature data. It has got following pin interface.



Prepared By:

Approved By:

Page no. 1 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No:0

G.PULLA REDDY ENGINEERING COLLEGE (AUTONOMOUS): KURNOOL
ELECTRICAL & ELECTRONICS ENGINEERING DEPARTMENT
Internet of Things Laboratory (IoT (P))

TITLE: Study and Implement Interfacing the temperature and humidity sensor to ESP32.

GPRECD/EEE/IoT(P)-15

Date: 10-08-2020

Algorithm

1. including the library of DHT11 temperature and humidity sensor
2. Assign LED pin on ESP32
3. Initialize GPIO as output using is pin MODE
4. Setup the serial monitor at baud rate 9600
5. In the loop(), print the humidity and temperature using serial print commands
6. Provide Delay for 600 milliseconds.
7. Setup the loop function runs over and over again forever.
8. Dump the code to ESP32.
9. Run the program on Arduino IDE.

Result:

Humidity and temperature should be displayed on serial monitor.

Prepared By:

Approved By:

Page no. 2 of 2

Dr. G. Satheesh

Dr.T.Bramhananda Reddy
Head of the Department

Revision No:0