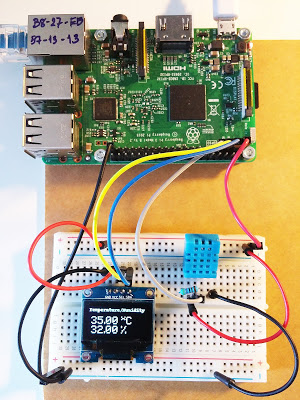
**PRACTICAL – 9**

**Write a program to understand the use of Firebase with Raepberry Pie to control sensors.**

|  |  |
| --- | --- |
|  | import smbus |
|  | import time |
|  | from firebase import firebase |
|  |  |
|  |  |
|  | bus = smbus.SMBus(1) #If I2C library detects the pull up then initialize the register using sensors |
|  |  |
|  |
|  |
|  |
|  | bus.write\_i2c\_block\_data(0x44, 0x2C, [0x06]) |
|  |  |
|  | time.sleep(0.5) |
|  |  |
|  | data = bus.read\_i2c\_block\_data(0x44, 0x00, 6) |
|  |  |
|  |  |
|  | temp = data[0] \* 256 + data[1] #shifting data[0] to left side and adding data[1] xisting in right side |
|  | cTemp = -45 + (175 \* temp / 65535.0) #formula mentioned in datasheet |
|  | fTemp = -49 + (315 \* temp / 65535.0) #formula mentioned in datasheet |
|  | humidity = 100 \* (data[3] \* 256 + data[4]) / 65535.0 |
|  |  |
|  |  |
|  | print "Temperature in Celsius is : %.2f C" %cTemp |
|  | print "Temperature in Fahrenheit is : %.2f F" %fTemp |
|  | print "Relative Humidity is : %.2f %%RH" %humidity |
|  |  |
|  | time.sleep(5) #5milliseconds |
|  |  |
|  |  |
|  | firebase= firebase.FirebaseApplication('HOST ID') |
|  |  |
|  |  |
|  | result = firebase.post('Project Name', {'cTemp':str(cTemp),'ftemp':str(fTemp), 'humidity':str(humidity)}) |
|  | print(result) |

**Output:-**

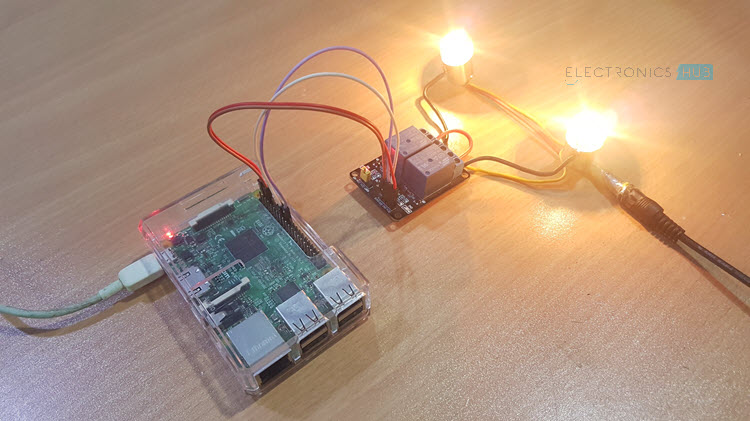
**[](https://2.bp.blogspot.com/-JcPernpaB8U/WLZ8WJgaEGI/AAAAAAAAnrk/ZzTQe71jmJczyaaa0jfUuphaQe8WXxvQQCLcB/s1600/RaspberryPi_DHT_OLED.jpg)**

**PRACTICAL – 10**

**Write a program to use Relay to control electrical equipment.**

|  |
| --- |
| import RPi.GPIO  as GPIO  import time |
| in1 = 16 |
| in2 = 18 |
|  |
| GPIO.setmode(GPIO.BOARD) |
| GPIO.setup(in1, GPIO.OUT) |
| GPIO.setup(in2, GPIO.OUT) |
|  |
| GPIO.output(in1, False) |
| GPIO.output(in2, False) |
|  |
| try: |
| while True: |
| for x in range(5): |
| GPIO.output(in1, True) |
| time.sleep(0.1) |
| GPIO.output(in1, False) |
| GPIO.output(in2, True) |
| time.sleep(0.1) |
| GPIO.output(in2, False) |
|  |
| GPIO.output(in1,True) |
| GPIO.output(in2,True) |
|  |
| for x in range(4): |
| GPIO.output(in1, True) |
| time.sleep(0.05) |
| GPIO.output(in1, False) |
| time.sleep(0.05) |
| GPIO.output(in1,True) |
|  |
| for x in range(4): |
| GPIO.output(in2, True) |
| time.sleep(0.05) |
| GPIO.output(in2, False) |
| time.sleep(0.05) |
| GPIO.output(in2,True) |
|  |
|  |
|  |
| except KeyboardInterrupt: |
| GPIO.cleanup() |

**Output:-**



|  |
| --- |
|  |
|  |  |