दीन दयाल उपाध्याय कॉलेज
DEEN DAYAL UPADHYAYA COLLEGE
(दिल्ली विश्वविद्यालय) (UNIVERSITY OF DELHI)
दिल्ली रा. रा. क्षेत्र सरकार द्वारा 100% वित्त पोषित, 100% funded by Govt. of NCT of Delhi
सेक्टर -3, द्वारका, नई दिल्ली Sector-3, Dwarka, New Delhi – 110078
दूरभाष/Tel. 011- 41805580, 45051037, Website: https://dducollegedu.ac.in

Internet of Things Practical file

Ram Tripathi
Roll no: 22HEL2231
Course: BSc(H) Electronics
Year and Semester: 2^{nd} *Year* , 4^{th} *sem*

Submitted To:
Dr. Poonam Kasturi Mam

Date: 26th April 2024

Index Table

S no.	Experiment Name	Sign
1	Connect an LED to GPIO pin 24 and a Switch to GPIO 25 and control the LED with the switch. The state of LED should toggle with every press of the switch.	
2	To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings	
3	To interface motor using relay with Arduino/Raspberry Pi and write a program to turn ON motor when push button is pressed	
4	Create a traffic light signal with three colored lights (Red, Orange and Green) with a duty cycle of 5-2-10 seconds	
5	To install MySQL database on Raspberry Pi and perform basic SQL queries	
6	Write a program to create TCP server on Arduino/Raspberry Pi and respond with humidity data to TCP client when requested.	
7	Interfacing 7-Segment LED with Raspberry	
8	Write a program on Arduino/Raspberry Pi to upload/retrieve temperature and humidity data using ThingSpeak cloud.	

Raspberry Pi Experiments Codes

Ram Tripathi

April 26th 2024

Abstract

This document contains the code and descriptions of various experiments conducted using a Raspberry Pi. Each section describes a separate experiment with the respective Python code.

Practical 1

```python
from gpiozero import LED, Button
from signal import pause

led = LED(24)
button = Button(25)

button.when_pressed = led.on
button.when_released = led.off

pause()
```

Practical 2

```python

import Adafruit_DHT
import time

DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 4

while True:
    humidity, temperature = Adafruit_DHT.read(DHT_SENSOR, DHT_PIN)
    if humidity is not None and temperature is not None:
        print("Temp={0:0.1f}C Humidity={1:0.1f}%".format(
            temperature, humidity))
    else:
        print("Sensor failure. Check wiring.");
    time.sleep(2);
```

Practical 3

```
from gpiozero import LED, Button
from signal import pause

motor = LED(24)
button = Button(25)

button.when_pressed = motor.on
button.when_released = motor.off

pause()
```

Practical 6

```
from gpiozero import TrafficLights
from time import sleep
from signal import pause

lights = TrafficLights(2, 3, 4)

def traffic_light_sequence():
    while True:
        yield (0, 0, 1) # green
        sleep(10)
        yield (0, 1, 0) # amber
        sleep(1)
        yield (1, 0, 0) # red
        sleep(10)
        yield (1, 1, 0) # red+amber
        sleep(1)

lights.source = traffic_light_sequence()

pause()
```

Practical 11

Ram Tripathi

May 2, 2024

1 Aim

The aim of this practical is to demonstrate setting up Basic UDP server and client and them communicating to each other

2 Server and Client Sides communicating

```
ram@ram:~/python $ sudo python udpprac.py
Server is up and listening......
INC
Client_Address 192.168.29.65
1
Howdy Server , from your client
Client_Address 192.168.29.65
1
```

Figure 1: Server side up and running

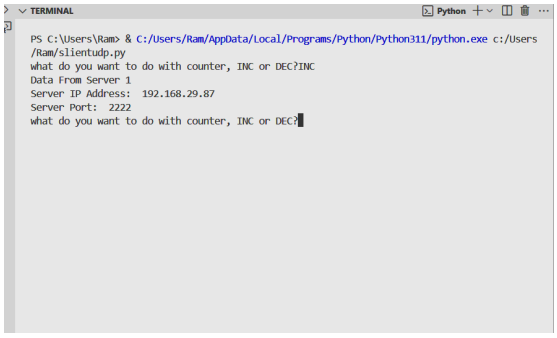

Figure 2: Client side up and running and sending data

C: > Users > Ram > slientudp.py > ...

```python
import socket
msgFromClient = "Howdy Server , from your client"
bytesToSend = msgFromClient.encode('utf-8')
serverAddress=('192.168.29.87',2222)
bufferSize= 1024
UDPClient = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
while True:
    cmd= input('what do you want to do with counter, INC or DEC?')
    cmd = cmd.encode('utf-8')
    UDPClient.sendto(cmd,serverAddress)

    UDPClient.sendto(bytesToSend, serverAddress)
    data,address=UDPClient.recvfrom(bufferSize)
    data = data.decode('utf-8')
    print("Data From Server", data)
    print("Server IP Address: ", address[0])
    print("Server Port: ", address[1])

```

Figure 3: Client side code

ram@ram: ~/python

GNU nano 7.2 udpprac.py

```python
import socket
import time
bufferSize = 1024
msgFromServer = 'Howdy Client, Happy to be Your Server'
ServerPort = 2222
ServerIP= '192.168.29.87'
bytesToSend = msgFromServer.encode('utf-8')
RPIsocket = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
RPIsocket.bind((ServerIP, ServerPort))
print('Server is up and listening.....')
cnt = 0
while True:
        message,address = RPIsocket.recvfrom(bufferSize)
        message = message.decode('utf-8')
        print(message)
        print('Client_Address', address[0])
        if message=='INC':
                cnt = cnt+1
        if message=='DEC':
                cnt = cnt-1

        print(str(cnt))
        response =  str(cnt)


        bytestoSend = response.encode('utf-8')


        RPIsocket.sendto(bytestoSend, address)
```

Figure 4: Server side code

Interfacing 7 segment display with Raspberry pi

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

May 2, 2024

1 Aim

To interface 7 segment display with Raspberry Pi

2 Circuit diagram

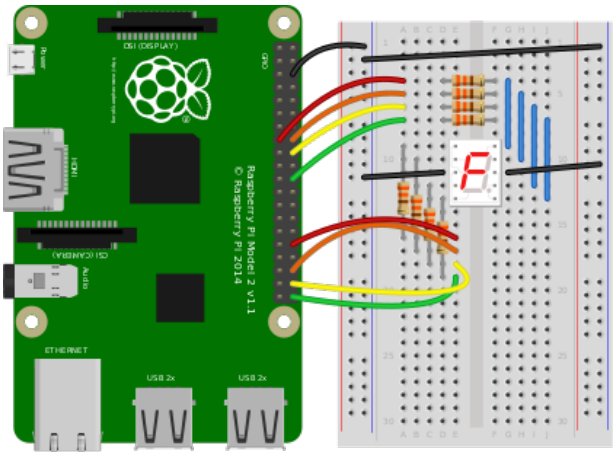

Figure 1: Circuit Diagram.

3 Procedure

To interface a 7-segment LED with a Raspberry Pi using gpiozero, begin by connecting the LED to GPIO pins, assigning each segment and digit to separate pins. Install the gpiozero library using pip install gpiozero. Write a Python script importing LED from gpiozero and sleep from time. Define lists for segments and digits, assigning GPIO pins accordingly.

4 Results

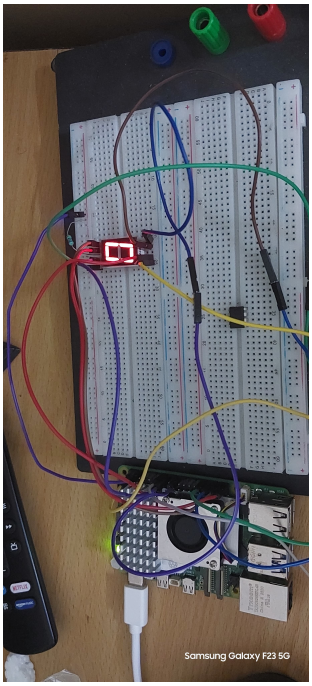

Figure 2: 7 segment display setup displaying number 0

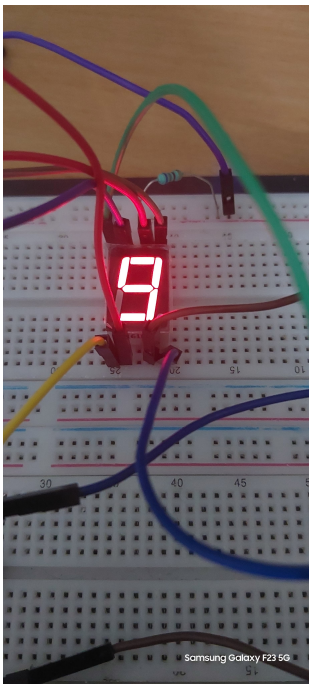

Figure 3: 7 segment display setup and displaying number 9

5 Code Used

```
from gpiozero import LEDCharDisplay
from time import sleep

display = LEDCharDisplay(21, 20, 16, 22, 23, 24, 12, dp=25)

for char in '987654321':
    display.value = char
    sleep(1)

display.off()
```

Practical 8

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

May 2, 2024

1 Aim

To acquire Temperature and humidity data from a dht11 sensor and publish it to the ThingSpeak cloud

2 Circuit diagram

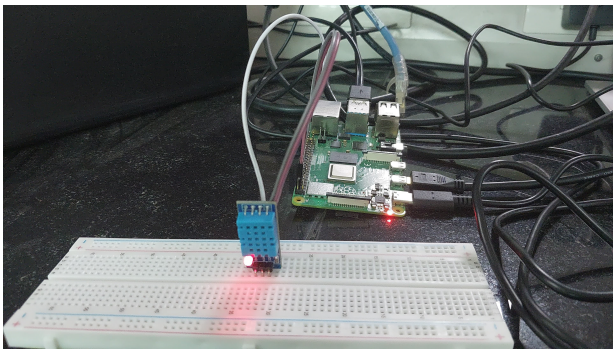

Figure 1: Circuit Diagram.

3 Procedure

The present practical aims to acquire data from dht11 sensor and publishes it to the thingspeak cloud using post command.

4 Results

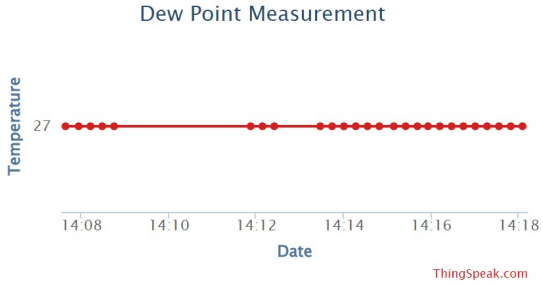

Figure 2: Temperature data being published to thingspeak cloud

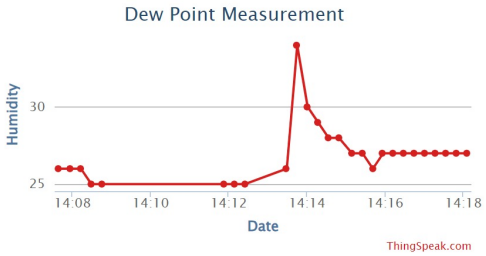

Figure 3: Humidity data being published to thingspeak cloud

5 Code Used

```
import time
import board
import adafruit_dht
import requests

API_KEY = '402QV2OVIBHQ9N1L'
URL = 'https://api.thingspeak.com/update'

sensor = adafruit_dht.DHT11(board.D14)

print("Reading data from DHT11 sensor and uploading to ThingSpeak....")
try:
    while True:
        Temperature = sensor.temperature
        Humidity = sensor.humidity

        print("Temp= {0:0.1f}C ,Humidity = {1:0.1f}%".format(Temperature, Humidity))

        if Humidity is not None and Temperature is not None:
            payload = {
                'api_key': API_KEY,
                'field1': Temperature,
                'field2': Humidity
            }
            try:
                response = requests.post(URL, params=payload)
                if response.status_code == 200:
                    print("Data uploaded successfully to ThingSpeak. Status Code:", respons
                else:
                    print("Failed to upload to ThingSpeak. Status Code:", response.status_c
            except Exception as e:
                print("Error occurred while uploading to ThingSpeak:", e)

        time.sleep(4)
# Adjust the sleep time according to your update interval

except KeyboardInterrupt:
    print("Exiting...")
```

Practical 1

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

April 30, 2024

1 Aim

Connect an LED to GPIO pin 24 and a Switch to GPIO 25 and control the LED with the switch. The state of LED should toggle with every press of the switch

2 Circuit diagram

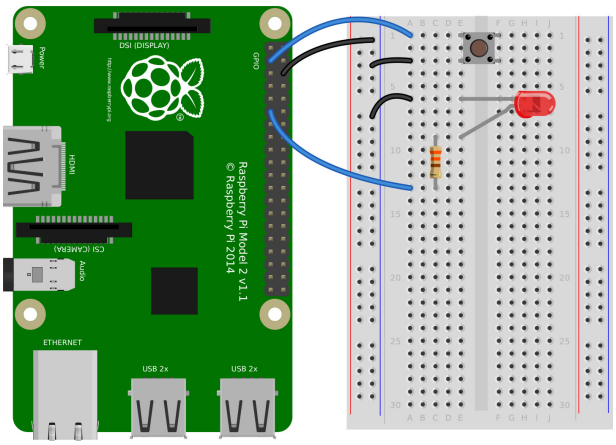

Figure 1: Circuit Diagram.

3 Procedure

In this experiment, GPIO pin 24 will be designated to connect with an LED, serving as the output channel for controlling its illumination state. Additionally, GPIO pin 25 will interface with a switch, acting as the input channel to detect user actions. The primary goal is to create a toggle functionality, where pressing the switch alternates the state of the LED between on and off.

4 Results

Figure 2: Before pressing the button.

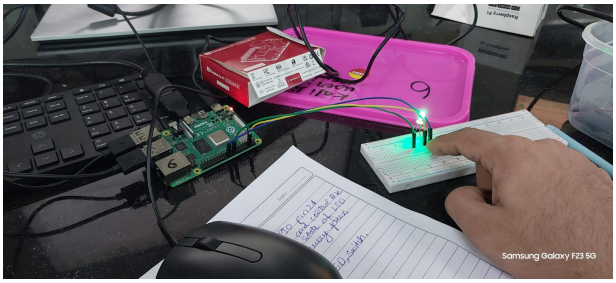

Figure 3: After pressing the button.

Practical 2

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

April 30, 2024

1 Aim

To interface DHT11 sensor with Arduino/Raspberry Pi and write a program to print temperature and humidity readings.

2 Circuit diagram

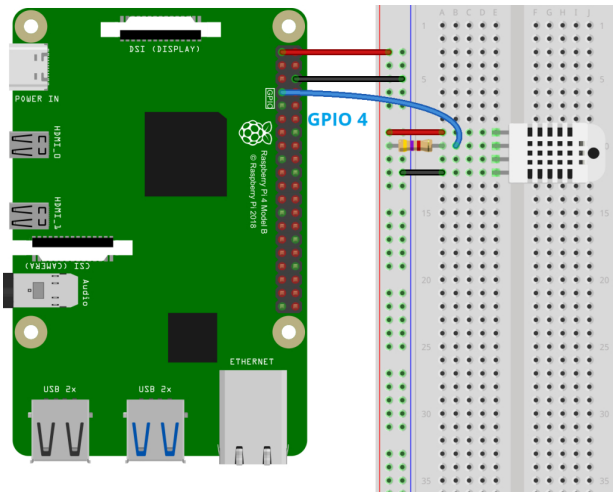

Figure 1: Circuit Diagram.

3 Procedure

In this experiment, we will explore the process of interfacing the DHT11 sensor with a Raspberry Pi and writing a Python program to print temperature and humidity readings to the terminal. We will discuss the hardware connections required to establish communication between the sensor and the Raspberry Pi, as well as the software setup needed to read data from the sensor using Python.

4 Results

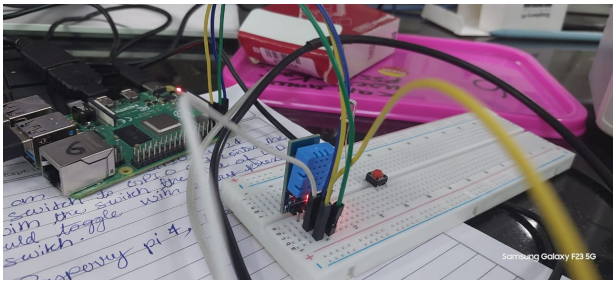

Figure 2: Circuit Assembled

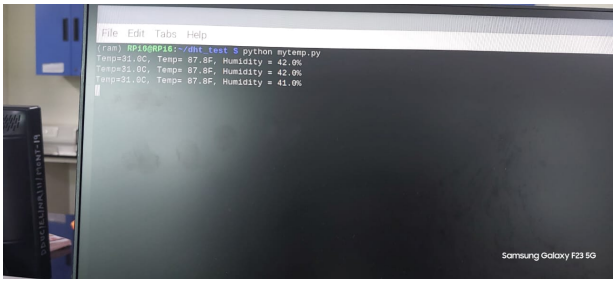

Figure 3: Temperature and humidity readings.

Practical 3

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

April 30, 2024

1 Aim

To interface motor using relay with Raspberry Pi and write a program to turn ON motor when push button is pressed.

2 Circuit diagram

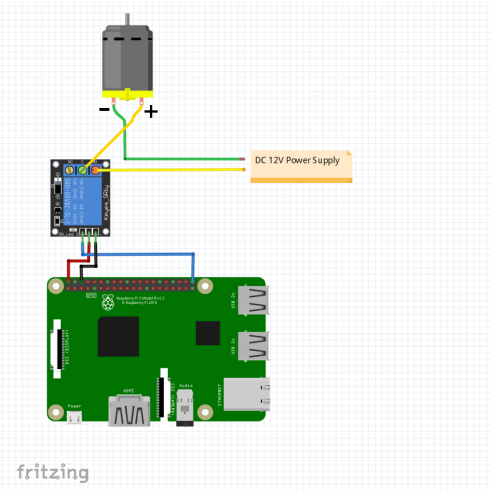

Figure 1: Circuit Diagram.

3 Procedure

In this experiment, we will explore the process of interfacing a motor with a Raspberry Pi using a relay module. We will discuss the hardware connections required to establish communication between the Raspberry Pi, the relay module, and the motor.

4 Results

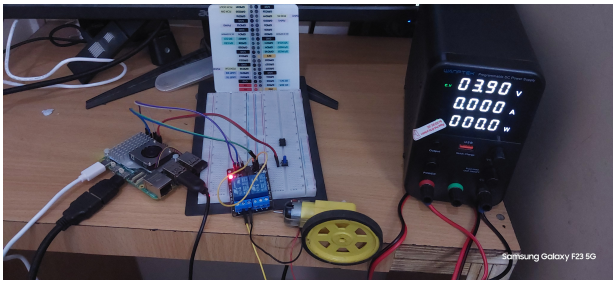

Figure 2: Circuit Assembled

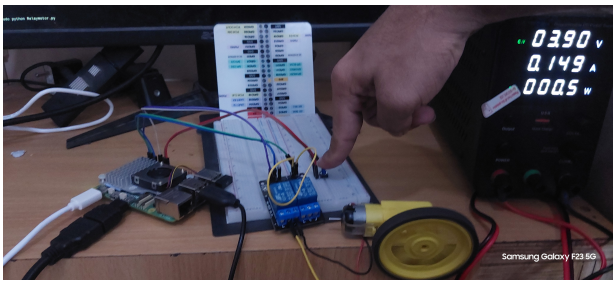

Figure 3: Motor running on the press of a button

Practical 6

Ram Tripathi

Electronics Department, Deen Dayal Upadhyaya College, Delhi University

April 30, 2024

1 Aim

Create a traffic light signal with three colored lights (Red, Orange and Green) with a duty cycle of 5-2-10 seconds

2 Circuit diagram

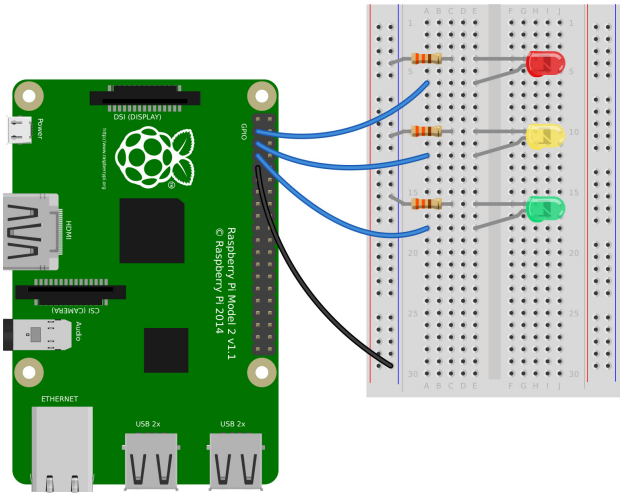

Figure 1: Circuit Diagram.

3 Procedure

We will implement a duty cycle of 5-2-10 seconds for the Red, Orange, and Green lights, respectively. This means that the Red light will remain illuminated for 5 seconds, followed by the Orange light for 2 seconds, and finally, the Green light for 10 seconds before looping back to the Red light.

4 Results

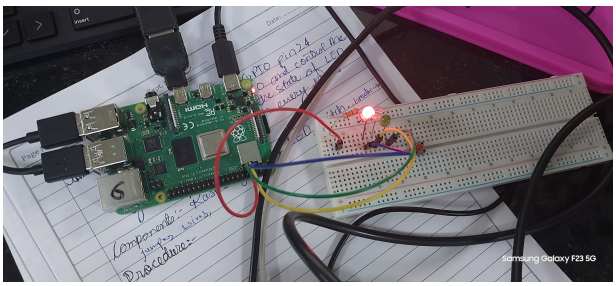

Figure 2: Red Light On

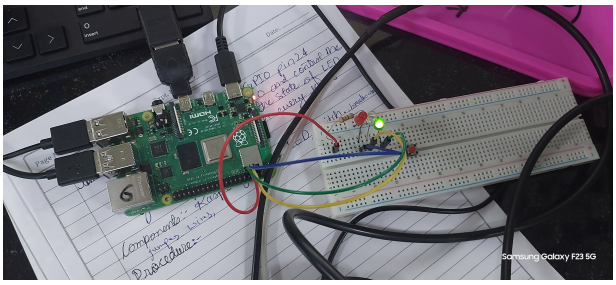

Figure 3: Green Light On

Practical 10

Ram Tripathi

April 30, 2024

1 Aim

The aim of this practical is to demonstrate basic operations in MySQL/MariaDB, including creating databases, tables, inserting data, and querying data.

2 Start the MySQL/MariaDB Service

```
ram@ram:~ $ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
Aborting!

Cleaning up...
ram@ram:~ $ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 40
Server version: 10.11.6-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> SELECT user, host FROM mysql.user;
+-------------+-----------+
| User        | Host      |
+-------------+-----------+
| mariadb.sys | localhost |
| mysql       | localhost |
| root        | localhost |
| Ram         | ram       |
+-------------+-----------+
4 rows in set (0.001 sec)
```

Figure 1: MySql installation and root user access

3 Various operations in MySql

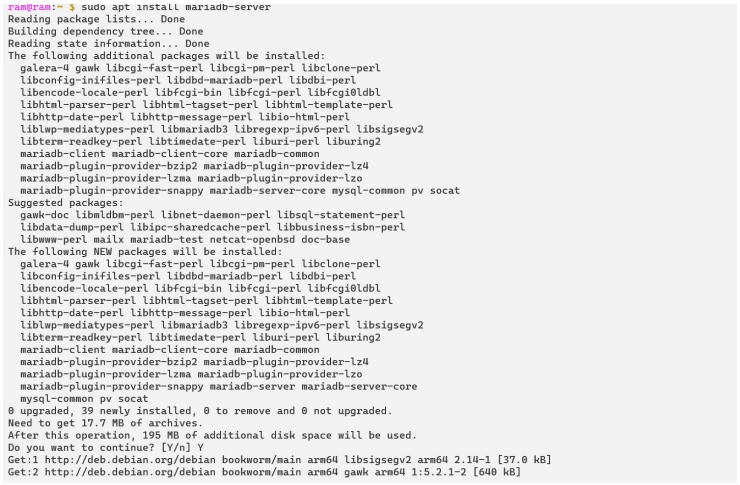

Figure 2: Output of accessing the MySQL/MariaDB command-line interface.

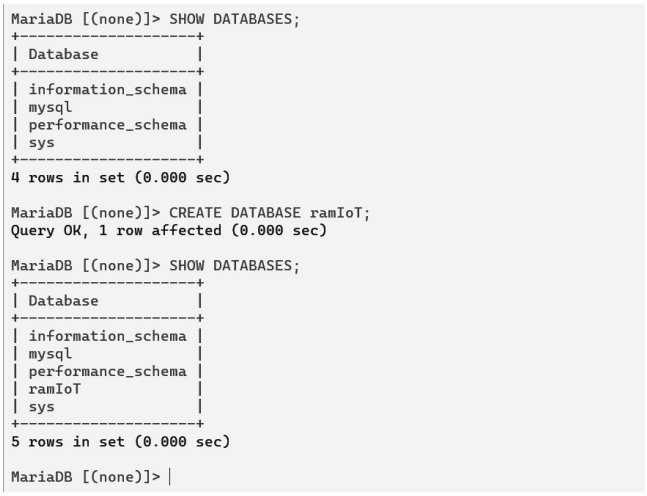

Figure 3: Output of creating a database.

```
MariaDB [ramIoT]> CREATE TABLE people (
    -> name VARCHAR(100),
    -> phone_number VARCHAR(20)
    -> );
Query OK, 0 rows affected (0.020 sec)

MariaDB [ramIoT]> SHOW TABLES;
+------------------+
| Tables_in_ramIoT |
+------------------+
| people           |
+------------------+
1 row in set (0.000 sec)
```

Figure 4: Output of creating a table.

```
MariaDB [ramIoT]> DESCRIBE people;
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| name         | varchar(100) | YES  |     | NULL    |       |
| phone_number | varchar(20)  | YES  |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
2 rows in set (0.001 sec)
```

Figure 5: Output of inserting data into the table.