PRACTICAL REPORT

For IoT Practical

2022

DARSHAN RAMJIYANI (DSP)

DOCS, KSKV Kachchh University

4.8 Serial Communication – Send Binary data from python and receive them in Arduino

In this practical, the position of mouse pointer in (x, y) coordination will sent to the Arduino board. Arduino process this data and print the decimal value of sent data.

Python Code:

```
from time import sleep
import pyautogui
import serial

PORT: str = "com9"
BAUD_RATE: int = 9600
CONN: serial.Serial = None

def main():
    setup()
    while True:
        loop()

def setup():
    global BAUD_RATE, CONN, PORT

    print("[Python] : Connecting Dwaidh terminal to Arduino. Please wait...")
    while True:
        try:
            CONN = serial.Serial(PORT, BAUD_RATE)
            print(
                f"[Python] : Dwaidh terminal connected to the Arduino via {PORT}
at {BAUD_RATE} bps.")
            break
        except serial.SerialException as e:
            print(
                "[Python] : Can not connect to the port. Try again in 2
mininutes. ", e.args)
            from time import sleep
            sleep(2)

    while True:
        receive_data: str = CONN.readline().decode("ascii")
        if len(receive_data) > 0:
            if "Connected" in receive_data:
```

```
                print("[Python] : Arduino has successfully verify the
connection.")
                break
            else:
                print(f"[Arduino]: {receive_data}")
        continue


def loop():
    while True:
        receive_data: str = CONN.readline().decode("ascii")

        if len(receive_data) > 0:
            if "Provide data" in receive_data:
                x_coordination, y_coordination = pyautogui.position()
                print(f"[Python] : Sending ({x_coordination},{y_coordination}) as
({format(x_coordination, '08b')}, "
                      f"{format(y_coordination, '08b')}) to the Arduino.")
                ""
                CONN.write(bytes(""+format(x_coordination,
'08b')+","+format(y_coordination, '08b'), "ascii"))
            elif "done" in receive_data:
                print("[Arduino] : Process done.")
                x = input("[Python] : Press Y then enter to continue.")
                if x.lower() == 'y':
                    continue
                else:
                    print("[Python] : Closing the connection with Arduino...")
                    CONN.close()
                    print("[Python] : Connection has been closed.")
                    exit(0)
            else:
                print(f"[Arduino]: {receive_data}")
        sleep(1)


if __name__ == '__main__':
    main()
```

Arduino Code:

```
String data, subData1, subData2;
long binaryNumber;

void setup()
{

  int baudRate = 9600;

  /* Established Serial Communication. */
  Serial.begin(baudRate);

  /* Wait until Serial Communication not established. */
  while (!Serial)
  {
  }

  Serial.println("Connected.");
}

void loop()
{
  Serial.println("Provide data.");
  while (Serial.available() == 0)
  {
    ;
  }
  Serial.print("Data Recieved : ");
  data = Serial.readString();
  Serial.println(data);
  splitTheData();
  Serial.print("X-Coordination : ");
  Serial.print(convertBinaryToDecimal((long)subData1.toInt()));
```

```
  Serial.print(" Y-Coordination : ");
  Serial.println(convertBinaryToDecimal((long)subData2.toInt()));
  delay(1000);
  Serial.println("done");
}

void splitTheData()
{
  int sep_pos = data.indexOf(",");
  subData1 = data.substring(0, sep_pos);
  subData2 = data.substring(sep_pos+1, data.length());
}

long convertBinaryToDecimal(long binary) {

  long number = binary;
  long decimalVal = 0;
  long baseVal = 1;
  long tempVal = number;
  long previousDigit;

  while (tempVal) {
    //Converts Binary to Decimal
    previousDigit = tempVal % 10;
    tempVal = tempVal / 10;
    decimalVal += previousDigit * baseVal;
    baseVal = baseVal * 2;
  }
  //Returns the Decimal number
  return decimalVal;
}
```

Output: