Experiment No-2

Subject-Computer Laboratory II-Industrial Internet of Things

Class-BE AI & DS

Aim: Write a program for sending alert messages to the user for controlling and interacting with your environment.

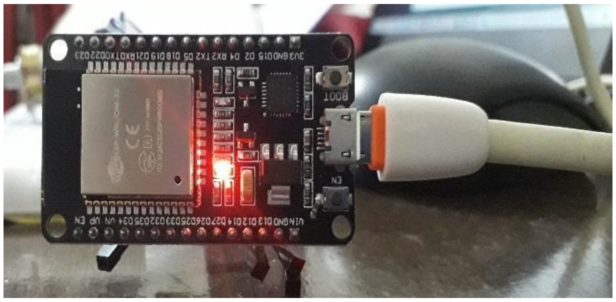
Software Requirement: Arduino IDE

Hardware Requirement: ESP-WROOM 32 board, Micro USB Data Cable, bread board, Potentiometer 10K, Male to female wires, Laptop/PC.

Theory: The program is written in the Arduino programming language and is designed to read the value of a potentiometer connected to GPIO 34 (Analog ADC1_CH6) on an Arduino board. It continuously reads the potentiometer value and prints it to the Serial Monitor. If the potentiometer value exceeds 2000, it also prints "warning" to the Serial Monitor.

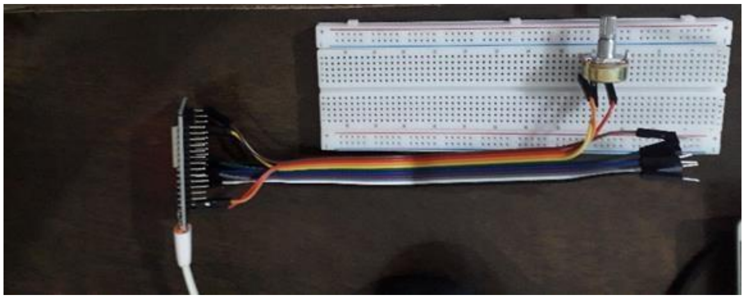
Connections:

1. First connect the data cable to the ESP 32 board, check out the notch and insert the cable in straight manner, without any tilt.

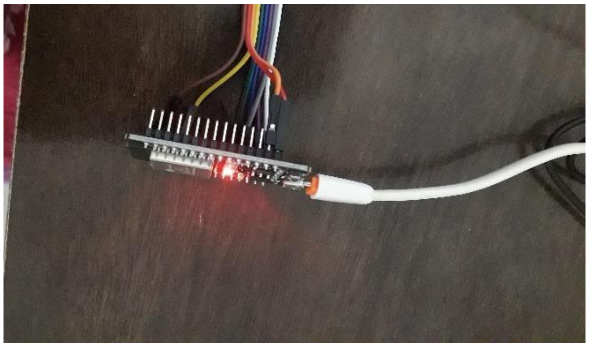


2. Connect the other end of the cable to the USB port of Laptop /PC and you should see a blue light glows.

3. Connect the potentiometer to the bread board as shown. Use the male female cables to connect the potentiometer to ESP 32.



4.The middle pin (yellow) of potentiometer is connected to GPIO 34, the red pin is connected to 3.3 V and the orange pin is connected to ground.



3.Open Arduino IDE, then go to File→Examples→ESP32→Analog Read

4. Analog reading is useful to read values from variable resistors like potentiometers, or analog sensors.

4. Upload the code provided to your ESP32.

5.Before uploading the code check if the right port is selected.

6.If the port is not selected then will get an error message.

4.Then go to Tools→Ports

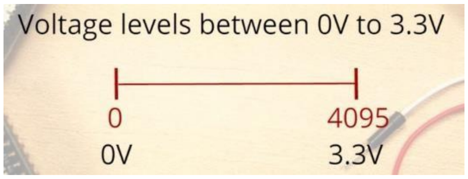
5.Install CH340g Driver from Google (exe file).

6.Now check on the port. It should show com3 or com4.Select the port.

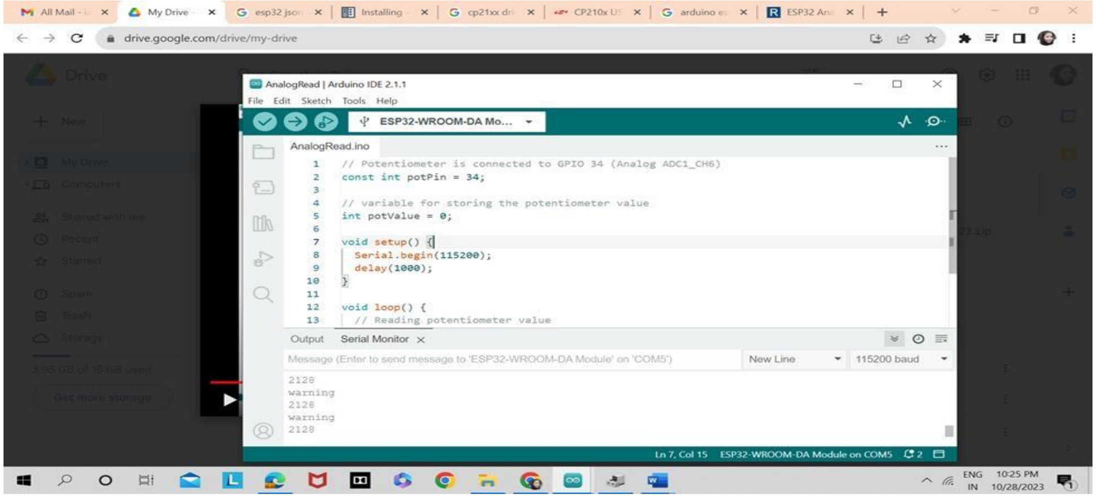
7. After uploading the code and pressing the ESP32 reset button, open the Serial Monitor at a baud rate of 115200. Rotate the potentiometer and see the values changing.

8. Read potentiometer ESP32 analogRead.

9. The maximum value you'll get is 4095 and the minimum value is 0.



10. Now rotate the potentiometer and if the potValue exceeds 2000, it issues a "warning" message and prints the value to the Serial Monitor.


AnalogRead | Arduino IDE 2.1.1
File Edit Sketch Tools Help
ESP32-WROOM-DA Mo...
AnalogRead.ino
Output
Serial Monitor
New Line
115200 baud
Ln 7, Col 15
ESP32-WROOM-DA Module on COM5

```
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value
int potValue = 0;

void setup() {
  Serial.begin(115200);
  delay(1000);
}

void loop() {
  // Reading potentiometer value
```

Message (Enter to send message to 'ESP32-WROOM-DA Module' on 'COM5')

```
2128
warning
2128
warning
2128
```

Procedure:

Explanation of the code:

1. A constant integer variable potPin is defined and assigns it the value 34. It represents the GPIO pin to which the potentiometer is connected.

2. int potValue = 0 declares an integer variable potValue and initializes it to 0. This variable will be used to store the value read from the potentiometer.

3. void setup ()- This is the setup function. It runs once when the Arduino board is powered up or reset.

4. Serial.begin(115200)- Initializes the serial communication with a baud rate of 115,200. This is used for printing messages to the Serial Monitor.

5. delay(1000)- A delay of 1000 milliseconds (1 second) is added to ensure that the Arduino has enough time to initialize before the loop() function starts executing.

6. void loop()- This is the main loop function. It runs continuously after the setup() function is executed.

7.potValue = analogRead(potPin)-Reads the analog voltage on the potPin (GPIO 34) and stores the result in the potValue variable. The analogRead function converts the voltage on the pin to a digital value between 0 and 4095 (assuming a 12-bit ADC resolution).

8.if (potValue > 2000) {Serial.println("warning");}-Checks if the value of potValue is greater than 2000. If it is, a "warning" message is printed to the Serial Monitor. This condition is used to detect when the potentiometer's value exceeds a certain threshold.

9.Serial.println(potValue)-Prints the current value of the potentiometer to the Serial Monitor. This line is executed regardless of the value of the potentiometer.

10.delay(500);: Adds a delay of 500 milliseconds (0.5 seconds) before the next iteration of the loop() function. This helps slow down the rate at which potentiometer values are printed to the Serial Monitor.

Conclusion: This program reads the analog value from a potentiometer, prints the value to the Serial Monitor, and issues a "warning" message if the potentiometer value is greater than 2000. It does this repeatedly in a loop with a 0.5-second delay between readings.

Program code for sending alert msg

```
// Potentiometer is connected to GPIO 34 (Analog ADC1_CH6)
const int potPin = 34;

// variable for storing the potentiometer value int
potValue = 0;

void setup() { Serial.begin(115200);
  delay(1000);
}

void loop() {
  // Reading potentiometer value
  potValue = analogRead(potPin); if
  (potValue>2000)
  {Serial.println("warning");}
  Serial.println(potValue); delay(500);
}
```