

EXPERIMENT NUMBER : 1

DATE : 25/03/2022 , FRIDAY

GPIO AND ADC PROGRAMMING USING ARDUINO

* AIM:

Understand both analog and digital sensor interfacing with a programmable hardware platform.

* CODE: (ARDUINO IDE)

(a) LED Blinking Using Arduino -

1 /*

2 LED Blink :

3 Turns on an LED on for a second, then off for a second,
repeatedly.

4 */

5

6 // The setup function runs once when you press reset or power the
board:

7 void setup ()

8 {

9 pinMode (13, OUTPUT); // Initialize digital pin 13 as an output

10 }

11

12 // The loop function runs over and over again forever :

13 void loop ()

14 {

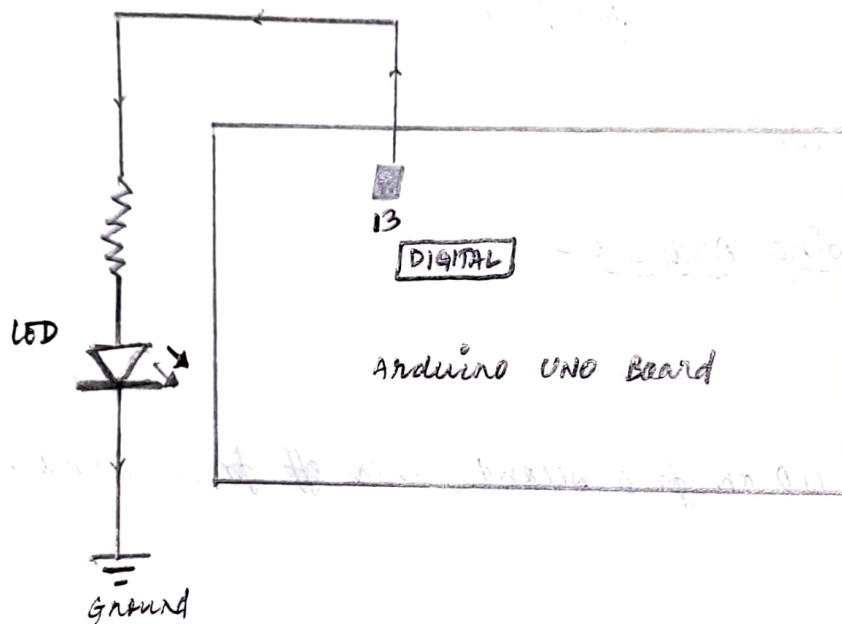
15 digitalWrite (13, HIGH); // Turn the LED on (HIGH is the voltage
level)

16 delay (1000); // wait for a second

17 digitalWrite (13, LOW); // Turn the LED off (LOW)

18 delay (1000); // wait for a second

19 }



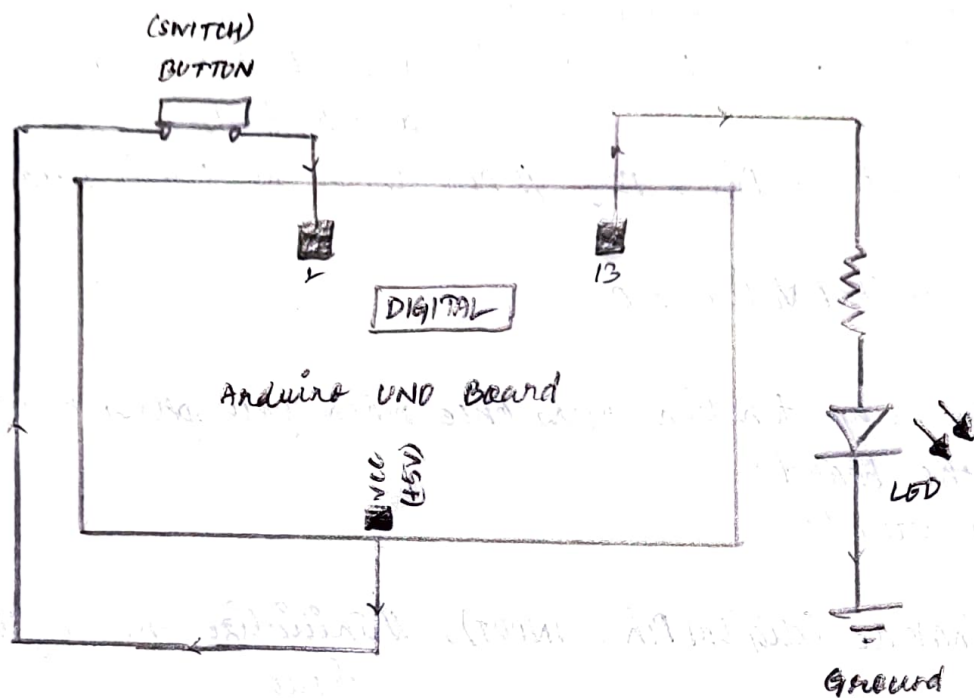
10) LED Blinking Using Arduino

Components used:

- ① Arduino Uno Board
- ② Yellow - LED
- ③ Resistor
- ④ Connecting wires
- ⑤ Power Supply Cable / Battery

(b) Controlling LED With Switch -

```
1 /*
2  * Switch program to control LED:
3  */
4
5 // This constant won't change:
6 const int digitalPin = 2; // Pin that the digital pin is
                          // attached to
7 const int ledPin = 13; // Pin that the LED is attached to
8
9 int digitalValue = 0;
10
11 // The setup function runs once when you press reset or power
12 // the board:
13 void setup()
14 {
15   pinMode(digitalPin, INPUT); // Initialize the digital pin as an
                          // input
16   pinMode(ledPin, OUTPUT); // Initialize the LED pin as an
                          // output
17 }
18
19 // The loop function runs over and over again forever:
20 void loop()
21 {
22   digitalValue = digitalRead(digitalPin); // Read the value
23
24   // If the digital value is equal to 2, turn on the LED:
25   if (digitalValue == HIGH)
26   {
27     digitalWrite(ledPin, HIGH);
28   }
29 }
```



(b) Controlling LED with Switch

Components Used:

- ① Arduino Uno Board
- ② Yellow LED
- ③ Resistor
- ④ Push Button
- ⑤ Connecting wires
- ⑥ Power supply cable / Battery

```

28 else
29 {
30   digitalWrite (ledPin, LOW);
31 }
32
33 }
34

```

(19) Controlling Relay with LDR Module

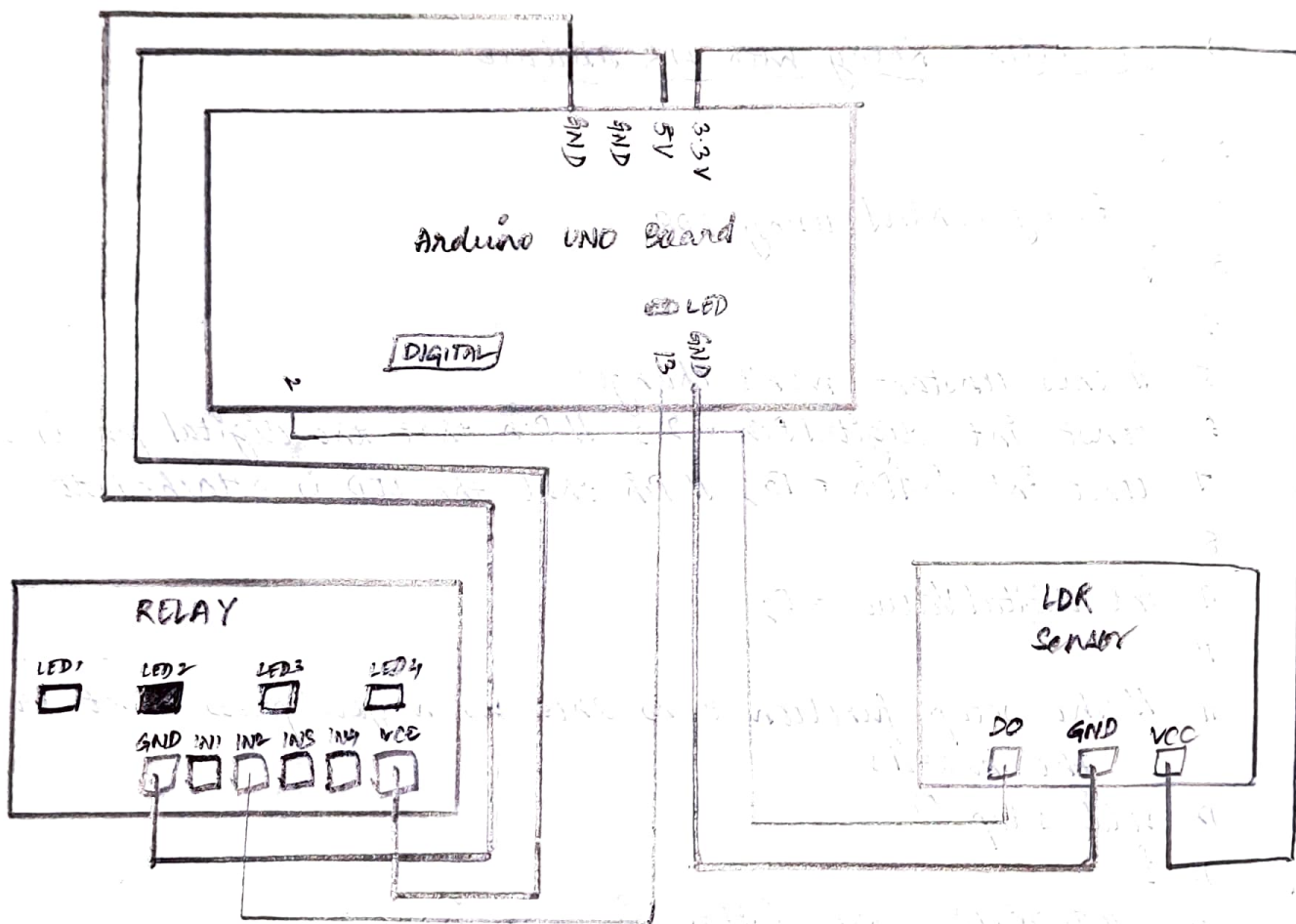
```

1  /*
2   * Relay Control using LDR
3   */
4
5  // This constant won't change:
6  const int digitalPin = 2; // Pin that the digital pin is attached to
7  const int ledPin = 13; // Pin that the LED is attached to
8
9  int digitalValue = 0;
10
11 // The setup function runs once when you press reset or power
12 // the board:
13 void setup ()
14 {
15   // Initialize the digital pin as an input:
16   pinMode (digitalPin, INPUT);
17   // Initialize the LED pin as an output:
18   pinMode (ledPin, OUTPUT);
19 }
20
21 // The loop function runs over and over again forever:
22 void loop ()

```


Components Used :

- ① Arduino UNO Board
- ② Relay
- ③ LDR Sensor
- ④ Connecting Wires
- ⑤ Power Supply cable/Battery



(c) Controlling Relay with LDR Module

```

22 {
23   digitalValue = digitalRead (digitalPin); //Read the value
24
25   // If the digital value is equal to 2, turn on the LED:
26   if (digitalValue == HIGH)
27   {
28     digitalWrite (ledPin, HIGH);
29   }
30   else
31   {
32     digitalWrite (ledPin, LOW);
33   }
34 }
35 }
36

```

10) Controlling Relay with Potentiometer -

1 //

2 It reads the state of a potentiometer (an analog input) and

3 turns on an LED only if the potentiometer goes above a

4 certain threshold level. It prints the analog level

5 regardless of the level.

6

7 Regarding the circuit:

- 8 (i) Potentiometer connected to analog pin 3. [Center Pin]
- 9 (ii) Side pins of the potentiometer go to +5V and ground.
- 10 (iii) LED connected from digital pin 13 to ground.
- 11

12 Note: on most Arduino boards, there is already an LED on

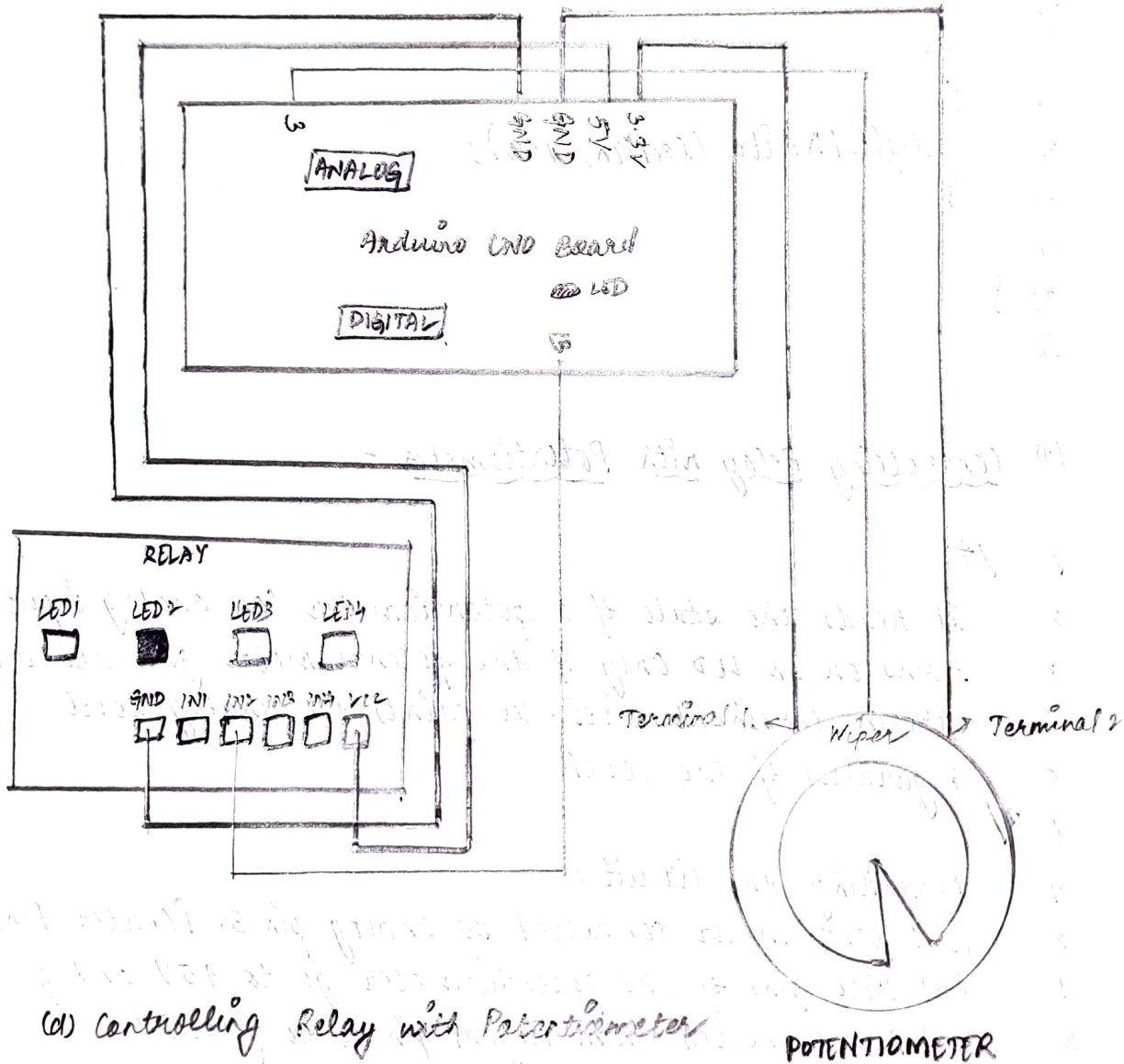
13 the board connected to pin 13, so you don't need any extra

14 components for this example.

15 +/

Components Required:

- ① Arduino UNO Board
- ② Relay
- ③ Potentiometer
- ④ Connecting Wires
- ⑤ Power Supply cable/Battery




```

16
17 // These constants won't change:
18 const int analogPin = A3; // Pin that the sensor is attached to
19 const int ledPin = 13; // Pin that the LED is attached to
20 const int threshold = 400; // An arbitrary threshold level that
    is in the range of the analog input
21
22 void setup()
23 {
24     pinMode(ledPin, OUTPUT); // Initialize the LED pin as an
25     Serial.begin(9600); // Initialize serial communications output
26 }
27
28 void loop()
29 {
30     int analogValue = analogRead(analogPin); // Read the value
        of the potentiometer
31
32     // If the analog value is high enough, turn on the LED:
33     if (analogValue > threshold)
34     {
35         digitalWrite(ledPin, HIGH);
36     }
37     else
38     {
39         digitalWrite(ledPin, LOW);
40     }
41
42     Serial.println(analogValue); // Print the analog value
43     delay(1); // Delay in between reads for stability
44 }
45

```

* INFERENCE:

Analyzed both analog and digital sensor interfacing techniques with a programmable hardware platform and all simulation results were verified successfully.