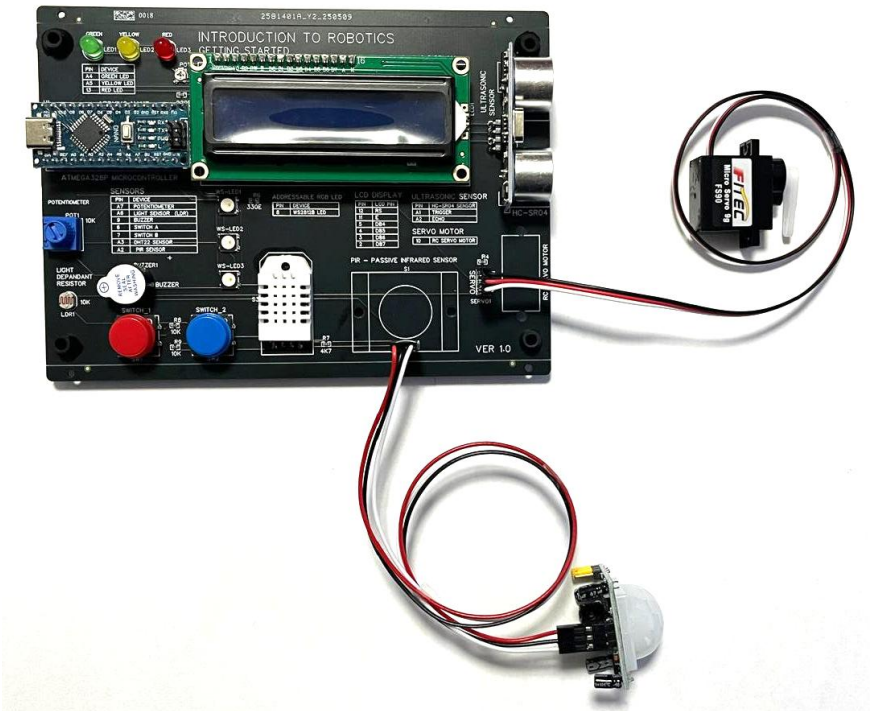
GETTING STARTERD WITH ROBOTICS

Switch detection with the Arduino



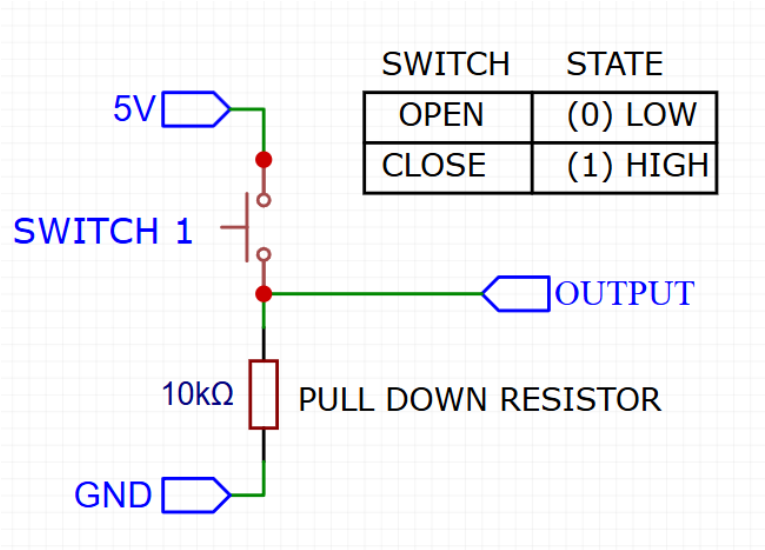
Background and what we have learnt so far

Up to this point, we have focused on writing digital outputs to control LEDs (HIGH or LOW). In this section, we will shift our focus to reading digital inputs by detecting the states of two switches (Switch 1 and Switch 2).

By the end of this exercise, you will understand how to read digital input signals and use them to control outputs.

Important Concept – PULL-DOWN resistor

The mainboard has two switches, each connected to ground through a 10kΩ pull-down resistor.



PULL-DOWN resistors keep the inputs LOW (0V) when the switch is not pressed. Pressing the switch applies 5V (HIGH).

Why do we use PULL-DOWN resistors

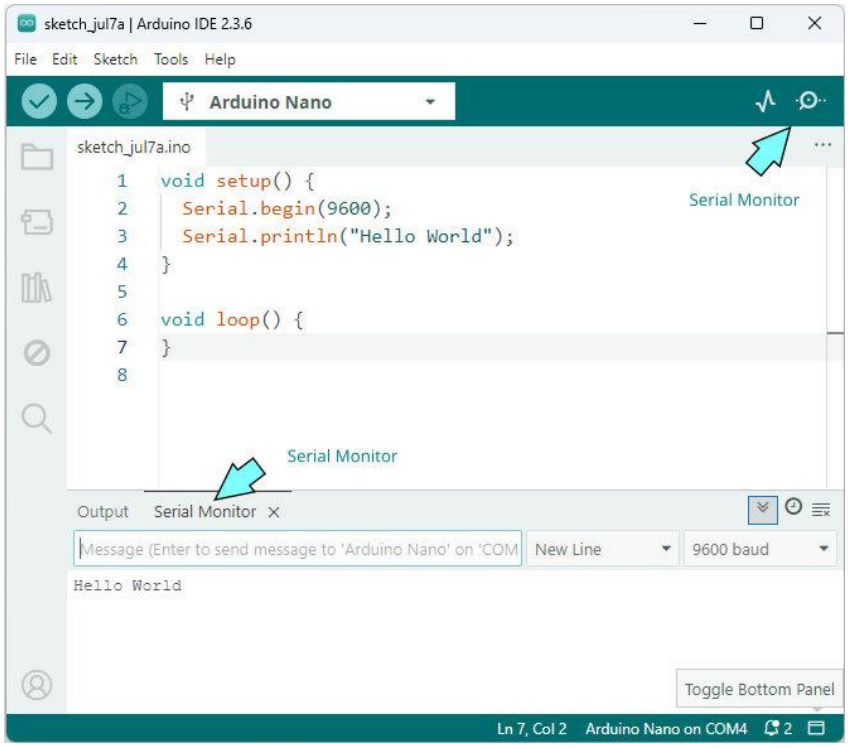
Without the pull-down resistor, the input may float and give unstable readings.

If a default HIGH (5V) state is required, a PULL-UP resistor can be used?

What is the Serial Port Monitor in the Arduino IDE?

The Serial Monitor is a tool in the Arduino IDE that shows data sent from the Arduino board to your computer over USB.

You open it by clicking the magnifying glass icon in the IDE.



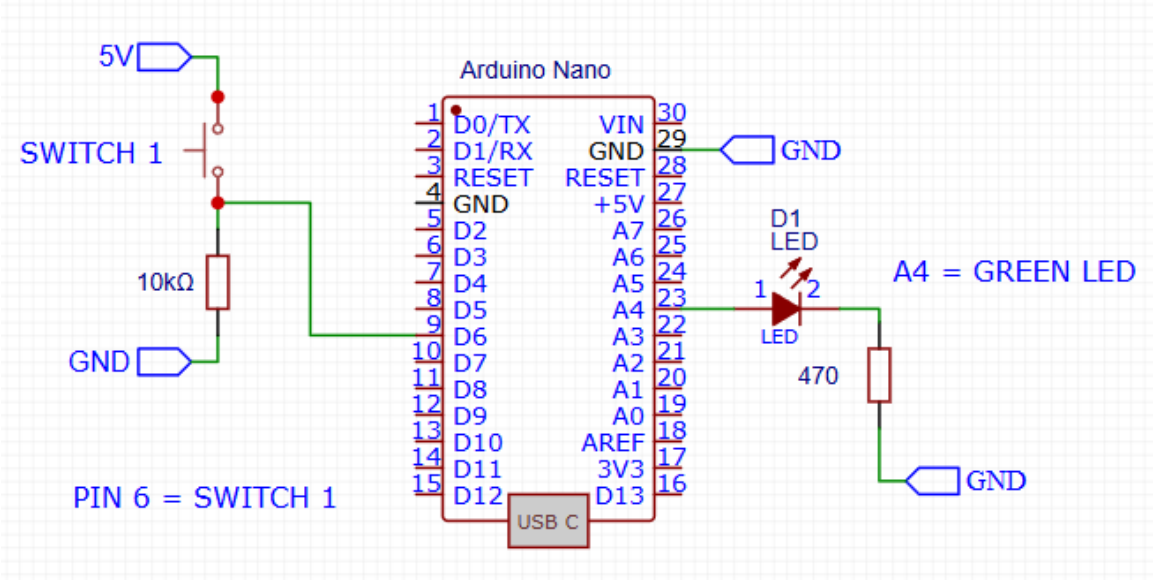
It's used to display messages, debug code, or view sensor values in real time.

Exercise 1 – Write data to the Serial Monitor

```
void setup() {
  Serial.begin(9600);             // Start serial comms
  Serial.println("Hello World"); // Print to Serial Monitor
}

void loop() {
  // Nothing needed here
}
```

Exercise 2

Write code to turn ON the Green LED on pin A4, only when Switch 1 (Pin 6)is pressed.




```
#define SWITCH1 6         // Switch 1 connected to Pin 6
#define GREEN_LED A4      // Green LED connected to Pin A4

void setup() {

  pinMode(SWITCH1, INPUT);     // Set Pin 6 to input
  pinMode(GREEN_LED, OUTPUT); // Set Pin A4 to output
}

void loop() {

  boolean state;
  state = digitalRead(SWITCH1);   // Read switch state
  digitalWrite(GREEN_LED, state); // Control LED
}
```

boolean -> declares a variable named state that can hold only two values: true or false.

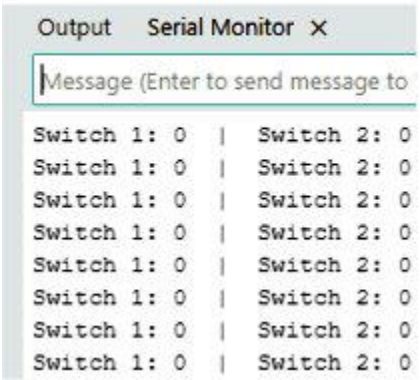
Exercise 2

Display the state of Switch 1 (Pin 6) and Switch 2 (Pin 7) on the Serial Terminal

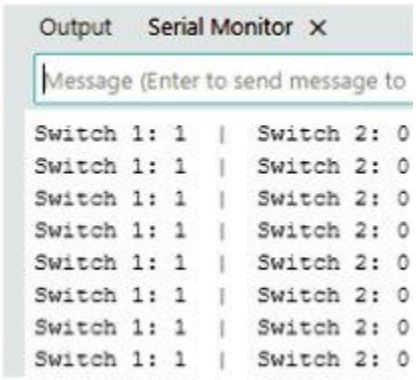
```
#define SWITCH1_PIN 6  // Switch 1
#define SWITCH2_PIN 7  // Switch 2

void setup() {
  pinMode(SWITCH1_PIN, INPUT);  // Set pin 6 as input
  pinMode(SWITCH2_PIN, INPUT);  // Set pin 7 as input
  Serial.begin(9600);           // Start serial comms
}

void loop() {

  int state1 = digitalRead(SWITCH1_PIN);  // Read switch 1
  int state2 = digitalRead(SWITCH2_PIN);  // Read switch 2

  // Print both switch states to the Serial Monitor

  Serial.print("Switch 1: ");
  Serial.print(state1);
  Serial.print("  |  Switch 2: ");
  Serial.println(state2);
  delay(200);  // Wait 200ms
}
```

Serial Monitor Output – No Switches Pressed

No Switches Pressed

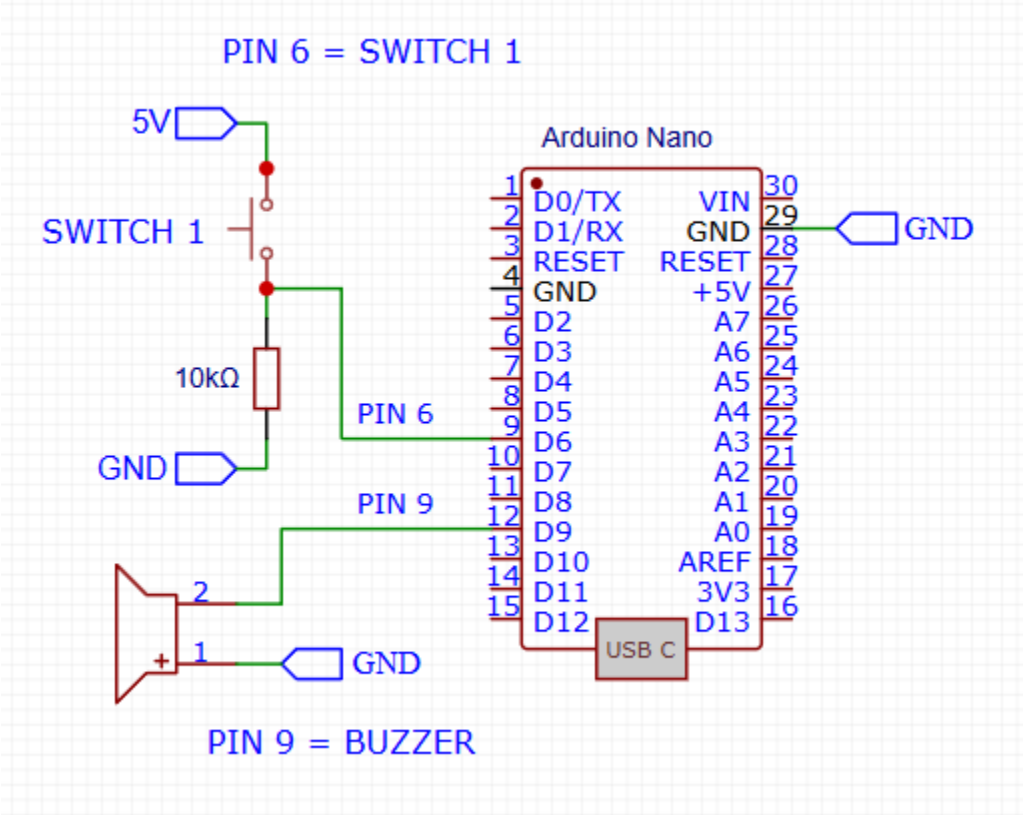


Switch 1 Pressed



Exercise 3

Now that we understand inputs (switches) and outputs LEDs), let's write a program to turn on a buzzer on pin 9 only when both switches are pressed. This demonstrates how digital inputs can control digital outputs using logic conditions.




```
#define SWITCH1_PIN 6    // Switch 1 input
#define SWITCH2_PIN 7    // Switch 2 input
#define BUZZER_PIN 9     // Buzzer output

void setup() {
  pinMode(SWITCH1_PIN, INPUT);
  pinMode(SWITCH2_PIN, INPUT);
  pinMode(BUZZER_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
```

```
  int state1 = digitalRead(SWITCH1_PIN);
  int state2 = digitalRead(SWITCH2_PIN);

  Serial.print("Switch 1: ");
  Serial.print(state1);
  Serial.print("  |  Switch 2: ");
  Serial.println(state2);

  if (state1 == HIGH && state2 == HIGH) {
    digitalWrite(BUZZER_PIN, HIGH);  // Turn on buzzer
  } else {
    digitalWrite(BUZZER_PIN, LOW);   // Turn off buzzer
  }

  delay(200);
}
```

What is important

```
pinMode(13, OUTPUT);          //Set Pin 13 to Output mode
pinMode(6, INPUT);            //Set Pin 6 to Input mode

Serial Terminal
---------------
Serial.begin(9600);           //Initialize Serial
Serial.print("Text");         //Write text to Serial

Check one conditions
--------------------

if (PIN6 == HIGH)

     {
      //do something
     }

Check 2 conditions
------------------

if (state1 == HIGH && state2 == HIGH)

PIN6 == HIGH)
```

```
{
 //do something
}
```