

Introduction to Robot Operating System

---

---

# Let's start with Arduino (1)

— Lec\_2 —

---

---

# Reminder

- pinMode(pin\_number, direction)
- digitalWrite(pin\_number, state)
- Libraries and Modules.

# Summary

- Reading Digital Sensor
- Serial Monitor
- Reading Analog Sensor
- LCD 16x2

# Reading Digital Sensor

Let's try to read a digital sensor using Arduino, we will need:

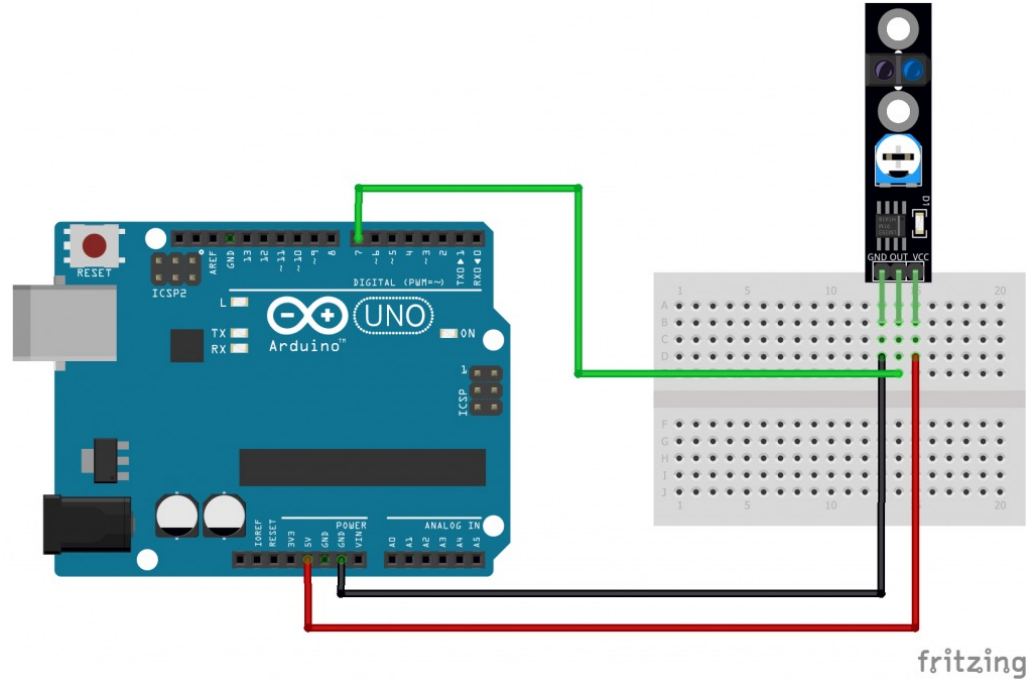
- Arduino Uno or Mega
- Black line sensor (KY-033)
- Jumper wires



KY-033	Arduino
GND	GND
OUT	Pin 7
VCC	5V

Working voltage	3.3V — 5.5V DC
Output signal	TTL level (high level if line detected, low if no line detected)
Board Size	1cm x 4.2cm [0.39in x 1.65in]

# Reading Digital Sensor (Connection Diagram)



# Reading Digital Sensor (Code)

We will use the following functions:

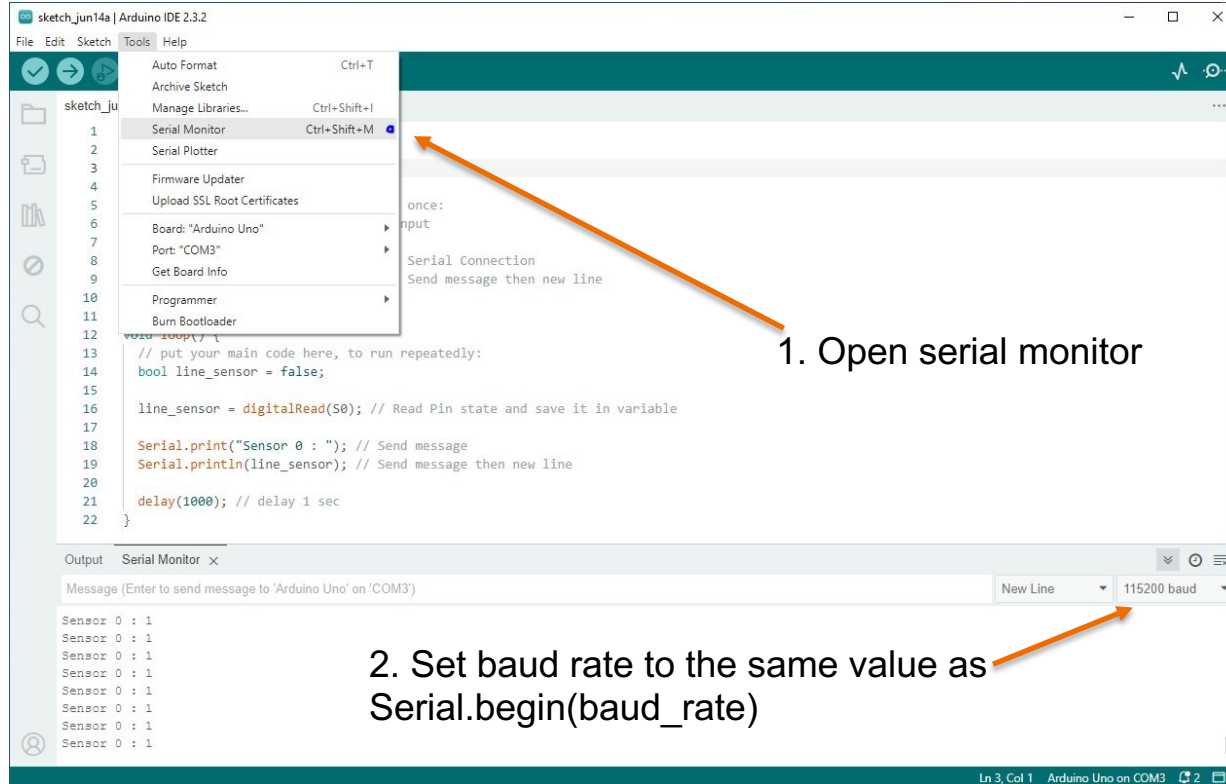
- `pinMode(pin_number, INPUT); // To make pin as input`
- `Serial.begin(baud_rate);`
- `Serial.print(string_message);`
- `Serial.println(string_message);`
- `digitalRead(pin_number); // This function return a boolean value`

**Using serial monitor is an effective way for diagnosing and checking our codes and sensor.**

# Reading Digital Sensor (Code)

```
1  // Sensor attached PIN
2  #define S0 2
3
4  void setup() {
5      // put your setup code here, to run once:
6      pinMode(S0, INPUT); // Set Pin as Input
7
8      Serial.begin(115200); // Initialize Serial Connection
9      Serial.println("Program Begin"); // Send message then new line
10 }
11
12 void loop() {
13     // put your main code here, to run repeatedly:
14     bool line_sensor = false;
15
16     line_sensor = digitalRead(S0); // Read Pin state and save it in variable
17
18     Serial.print("Sensor 0 : "); // Send message
19     Serial.println(line_sensor); // Send message then new line
20
21     delay(1000); // delay 1 sec
22 }
```

# Reading Digital Sensor (Serial Monitor)



The screenshot shows the Arduino IDE 2.3.2 interface. The 'Tools' menu is open, and the 'Serial Monitor' option is highlighted. An orange arrow points from the text '1. Open serial monitor' to this menu item. Below the menu, the code for reading a digital sensor is visible. At the bottom, the 'Serial Monitor' window is open, showing a message input field and a baud rate dropdown set to '115200 baud'. An orange arrow points from the text '2. Set baud rate to the same value as Serial.begin(baud\_rate)' to the baud rate dropdown. The code in the background is as follows:

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
void loop() {  
  // put your main code here, to run repeatedly:  
  bool line_sensor = false;  
  
  line_sensor = digitalRead(S0); // Read Pin state and save it in variable  
  
  Serial.print("Sensor 0 : "); // Send message  
  Serial.println(line_sensor); // Send message then new line  
  
  delay(1000); // delay 1 sec  
}
```

1. Open serial monitor

2. Set baud rate to the same value as `Serial.begin(baud_rate)`

Ln 3, Col 1 Arduino Uno on COM3 2



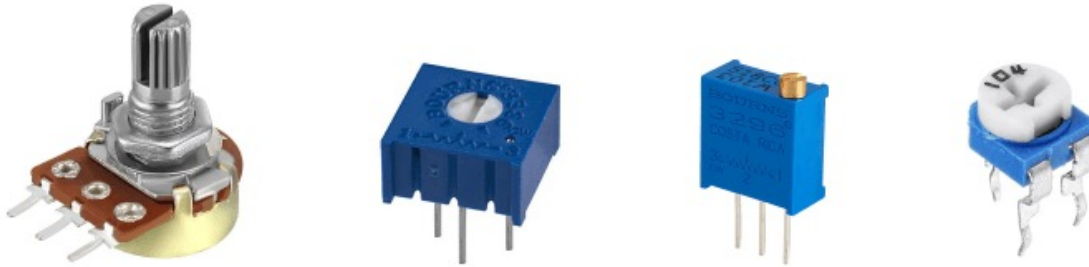
# Practice

- Add another sensor and try to view their states using serial monitor.

# Reading Analog Sensor

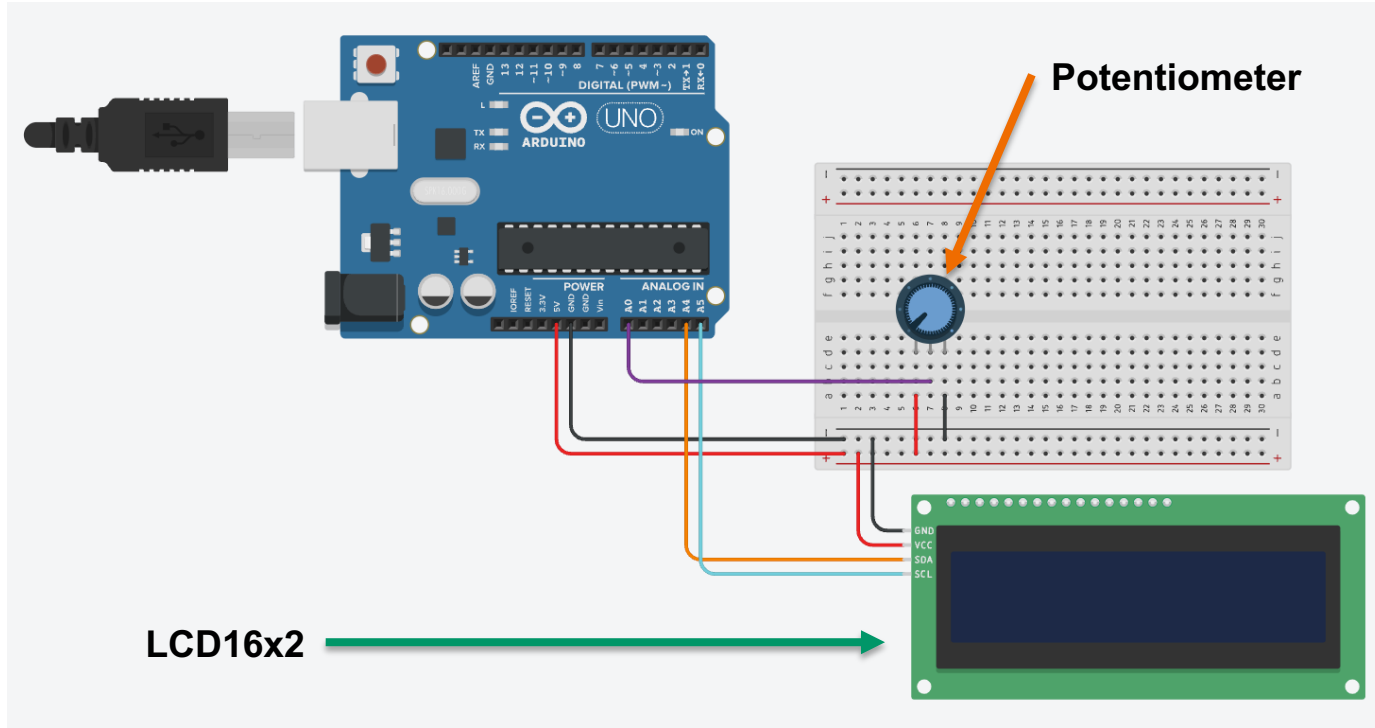
Let's try to read analog sensor using Arduino, we will need:

- Arduino Uno or Mega
- Potentiometer (Variable Resistor)
- LCD16x2
- Jumper wires



Different types of Potentiometers

# Reading Analog Sensor (Connection Diagram)



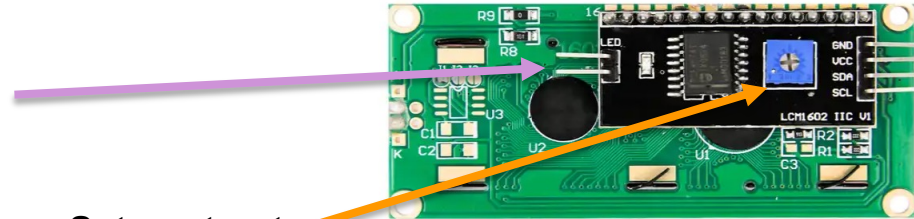
## Reading Analog Sensor (LCD16x2)

LCD is a very simple and powerful device to show data from Arduino:

- Number of columns = 16
- Number of rows = 2
- Use I2C connection protocol then two wires.
- Need library "LCD\_I2C"



## Disconnect LCD backlight



## Set contrast

# Reading Analog Sensor (Code)

We will use the following functions:

- `Lcd.begin();` **// Initialize LCD**
- `Lcd.backlight();` **// Turn on backlight**
- `Lcd.noBacklight;` **// Turn off backlight**
- `Lcd.clear();` **// Clear LCD**
- `Lcd.setCursor(col, row);` **// Set cursor position**
- `Lcd.print(String_message);`
- `analogRead(pin_number);` **// This function return an integer value (0 – 1023).**

# Reading Analog Sensor (Code)

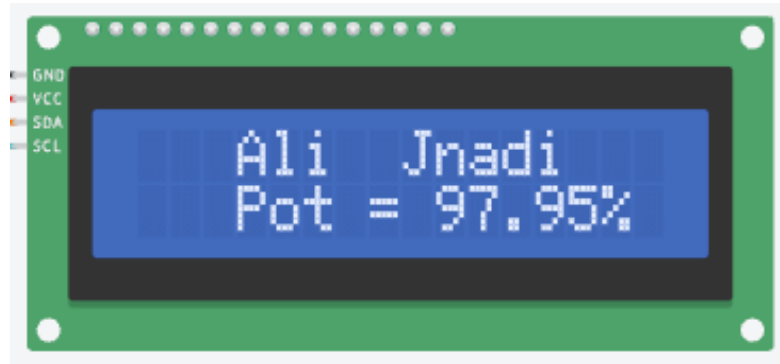
```
1  #include <LCD_I2C.h>
2  LCD_I2C lcd(0x27, 16, 2); // Default address of most PCF8574 modules (Address, col, row)
3
4  // Sensor attached PIN
5  #define Pot A0
6
7  void setup() {
8      // put your setup code here, to run once:
9      pinMode(Pot, INPUT); // Set Pin as Input
10
11     Serial.begin(115200); // Initialize Serial Connection
12     Serial.println("Program Begin"); // Send message then new line
13
14     lcd.begin(); // Initialize LCD
15     lcd.backlight(); // Turn on backlight
16     delay(500);
17     lcd.noBacklight(); // Turn off backlight
18     lcd.clear(); // clear LCD
19 }
20
```

# Reading Analog Sensor (Code)

```
21 void loop() {  
22     // put your main code here, to run repeatedly:  
23     int var_sensor = 0;  
24  
25     var_sensor = analogRead(Pot); // Read variable resistor value  
26  
27     Serial.print("Sensor: "); // Send message  
28     Serial.println(var_sensor); // Send message then new line  
29  
30     lcd.setCursor(0, 0); //setting the cursor in the desired position (col, row).  
31     lcd.print("Sensor");  
32     lcd.setCursor(3, 1);  
33     lcd.print(var_sensor);  
34  
35     delay(20); // delay 20 ms  
36 }  
37
```

# Practice

- Make the LCD display your name and the value of potentiometer as percentage.





**That's All for Today**