

# TScratch Basics

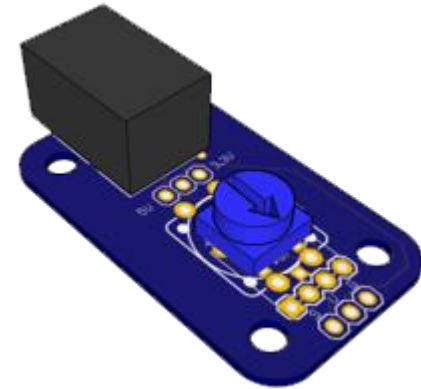
Coding with Arduino IDE (Software)

# Learning Objective

In this lesson you will learn:

**TScratch**

**TSense(Potentiometer)**



- Include an analog input into your project!
- Coding an analog input with Arduino
- Learn how to use the Arduino Serial Monitor

# Arduino Analog Function

## Analog function

Reading the values/state from the sensor inputs

`analogRead(sensor)`

- Different from digital function, the analog function has 1024 values (0 - 1023).
- When required to read a value from the sensor, the `analogRead()` function will be used.

\*\*Note that the word Read is in capital letter

# Arduino Analog Function

## Analog function

Reading the values/state from the sensor inputs

`analogWrite(output)`

- When required to write a value to the output, the `analogWrite()` function will be used.

\*\*Note that the word **Write** is in capital letter

# Arduino Mapping

## What is mapping?

- Mapping is the conversion of data values in proportion.
- i.e. 100cm = 1m or 100g into 0.1kg
- 100cm → 39.3 Inches
- Since we want the LED (digital) to display the value from the potentiometer (analog), we need to convert the value using mapping.

# Analog Vs Digital comparison

## Digital Signal

- Values/reading either high [1] or low [0]
- Readings is simple
- Activation (i.e. On and off) is straightforward

## Analog Signal

- Values/reading either Ranges from 0 to 1023
- Readings may be difficult (number changes)
- Activation (i.e. On and off) requires a condition

# Arduino Serial Monitor

## What is a monitor?

A screen or print out used to give the user instructions when writing a program or using microcontroller.



Serial Monitor  
Screen

# Arduino Serial Monitor

## How to use this screen to monitor the program?

- Declare the serial screen by typing `Serial.begin(9600);` in the setup code.
- In order to print the values in the new line, use `Serial.println(analogvalue);` in the loop code to display the analogvalue



# Code TSense(pot)

- Declare the following in the declaration space
  1. Define POT (potentiometer) as pin A2
  2. Define LED as pin 12
  3. A variable named “potvalue” to store the value of the potentiometer

Note: Comments appear after // or within /\*\*\*/

```
TScratch5_Potentiometer
/*
 * Program name : TScratch5_Potentiometer
 * Description  : This program changes the brightness of the LED when the potentiometer is adjusted
 */
#define POT_PIN A2
#define LED_PIN 12
int potvalue = 0;           //Declares a integer variable to store the value of the potentiometer
```

# Code TSense(pot)

- Declare the input/output in the setup
  1. POT → INPUT
  2. LED → OUTPUT
  3. Serial.begin(9600) to use the monitor screen

```
void setup() {  
  pinMode(POT_PIN, INPUT);      //Initializes the potentiometer as an INPUT  
  pinMode(LED_PIN, OUTPUT);     //Initializes the led as an OUTPUT  
  Serial.begin(9600);          //Initializes the Serial Monitor with a baud rate of 9600  
}
```

# Code TSense(pot)

- Use `analogRead()` to read the value of the potentiometer
- Store it in the “potstate” variable
- Use `Serial.println` to constantly display the value of potstate (and thus the potentiometer) in the serial monitor

```
void loop() {  
  potvalue = analogRead(POT_PIN); //Reads potentiometer and stores its value in the potstate variable  
  Serial.println(potvalue);        //Writes the value of potvalue to the serial monitor  
  analogWrite(LED_PIN, map(potvalue, 0, 1023, 0, 255)); //maps the value of potvalue from (0,1023) to (0,255).  
                                     //The value is then written to the LED_PIN using analogWrite  
}
```

# Code TSense(pot)

- Use mapping to scale the analog values (0 to 1023) to a digital display (0 to 255)
- Use `analogWrite()` to program the LED to display the mapped value

```
void loop() {  
  potvalue = analogRead(POT_PIN); //Reads potentiometer and stores its value in the potstate variable  
  Serial.println(potvalue);       //Writes the value of potvalue to the serial monitor  
  analogWrite(LED_PIN, map(potvalue, 0, 1023, 0, 255)); //maps the value of potvalue from (0,1023) to (0,255).  
                                     //The value is then written to the LED_PIN using analogWrite  
}
```

# Complete Program

```
#define POT_PIN A2
#define LED_PIN 12
int potvalue = 0;           //Declares a integer variable to store the value of the potentiometer

void setup() {
  pinMode(POT_PIN, INPUT); //Initializes the potentiometer as an INPUT
  pinMode(LED_PIN, OUTPUT); //Initializes the led as an OUTPUT
  Serial.begin(9600);      //Initializes the Serial Monitor with a baud rate of 9600
}

void loop() {
  potvalue = analogRead(POT_PIN); //Reads potentiometer and stores its value in the potstate variable
  Serial.println(potvalue);       //Writes the value of potvalue to the serial monitor
  analogWrite(LED_PIN, map(potvalue, 0, 1023, 0, 255)); //maps the value of potvalue from (0,1023) to (0,255).
  //The value is then written to the LED_PIN using analogWrite
}
```

# Try it yourself!

## Use TSense(Pot) to control both the buzzer and LED!

- Connect and declare TSense (buzzer).
- Use pin13 for buzzer
- Program the following:
  1. Change the frequency of the buzzer sound when the potentiometer dial is turned (like a volume control knob)
  2. When the buzzer is on, the LED should be switched off.  
(Hint: you will need to use 2 sets of **[If-Else]**s)

# Solution –TSense(Pot, Buzzer, LED)

```
#define POTENTIOMETER_PIN A2
#define BUZZER_PIN 13
int analogvalue = 0, buzzerfreq = 0;

void setup() {
  pinMode(pot, INPUT);      // Initializes the potentiometer as an input
  pinMode(led, OUTPUT);     // Initializes the led and buzzer as an output
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  Serial.println(analogvalue);
  analogvalue = analogRead(pot);
  buzzerfreq = map(analogvalue, 0, 1023, 40, 20000); // Maps the analog value(0-1023) to buzzer frequency values(40-20000)
  tone(BUZZER_PIN, buzzerfreq); // Plays the buzzer at buzzerfreq
}
```