

TScratch Basics

Coding with Arduino IDE (Software)

Learning Objective

In this lesson you will learn:

TScratch (TPad)



- Include a Button-based controller to your project!
- Coding another Analog input with Arduino

What is a TPad?

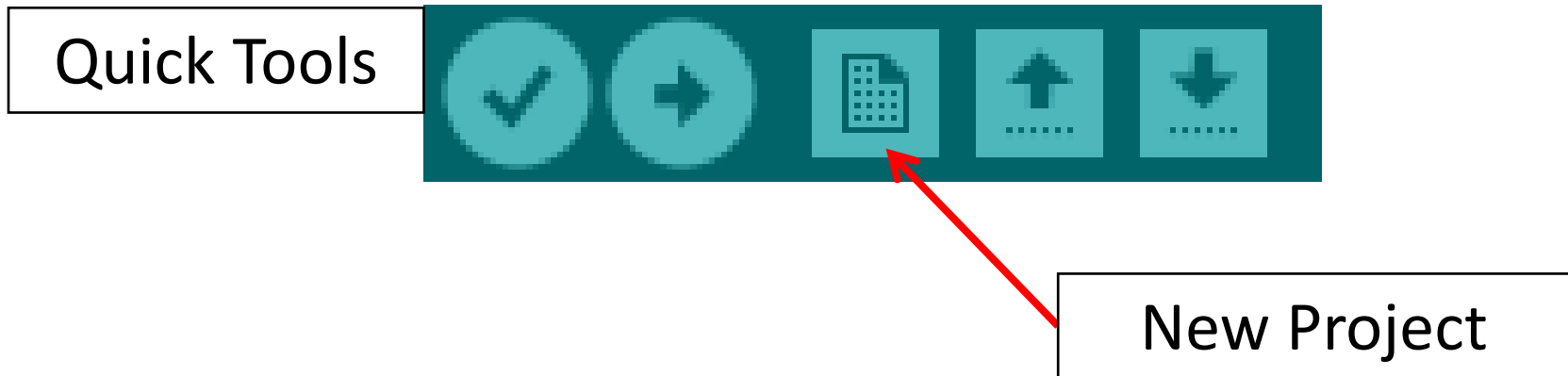
- TPad is made up of many resistors strung together
- It is a “switch” that works on analog readings
- When different switches are pressed, each activates a different number of resistors.
- Readings is based on voltage readouts.
- This simple layout can be found on game controllers, remote controls, brick phones, etc.

Code TSense(TPad)

Write a controller with TScratch!

In a simple step, connect the TScratch with TSense(LED) and TSense(TPad)

- Create a new project on the QUICK TOOLS



Code TSense(TPad)

- First, use a continuous sensing program in the loop() function to display the value.

The image shows a screenshot of an IDE with a code editor and a serial monitor window. The code editor displays the following code:

```

cont.sensing
int tpad = A1;
int led = 12;
int analogvalue = 0;

// the setup function runs once when you press reset or power the board
void setup() {
  Serial.begin(9600);
  // initialize
  // initialize
  pinMode(tpad,
  pinMode(led, 0
}
// the loop fu
void loop() {
  analogvalue =
  Serial.println
  delay(50);
}
    
```

The serial monitor window, titled "/dev/cu.wchusbserial1410", shows the output of the program. The output consists of a series of numbers: 694, 694, 694, 694, 695, 695, 694, 695, 640, 640, 640, 640, 0. The serial monitor also has a "Send" button and a "9600 baud" rate selected. At the bottom of the IDE, there is a status bar with "Autoscroll" checked, "No line ending" selected, and "9600 baud" selected.

Code TSense(TPad)

- Declare the following in the declaration space
 1. Define TPAD as pin A1
 2. Define LED as pin 12
 3. A variable named “tpadvalue” to store the sensed value

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

```
TScratch7_TPad
/~
* Program name : TScratch7_TPad
* Description  : This program turns the led on when a button on the tpad is pressed
*/

#define TPAD_PIN A1
#define LED_PIN 12
int tpadvalue = 0;
```

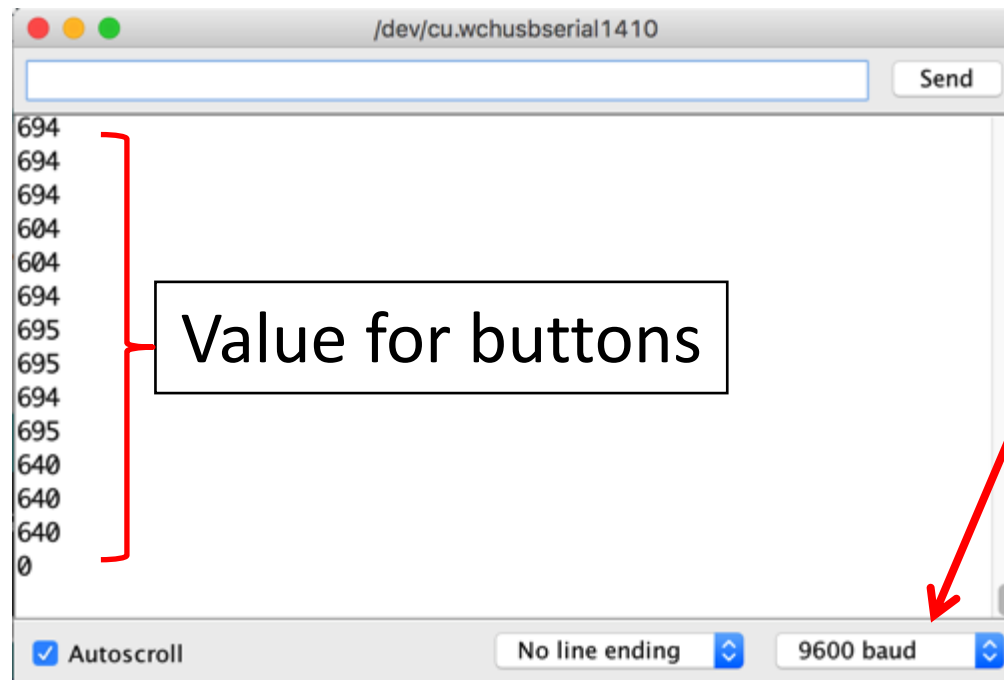
Code TSense(TPad)

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

```
void setup() {  
    pinMode(TPAD_PIN, INPUT);  
    pinMode(LED_PIN, OUTPUT);  
    Serial.begin(9600);  
}
```

Code TSense(TPad)

- Upload the code onto TScratch.
- Open the Serial Screen
- Test and **RECORD DOWN** the buttons' values



Code TSense(TPad)

- Sample values for Tpad

Note: Your values may differ as different Tpads may have slightly different resistance values (ie. off by 1 or 2)

| Button | Value |
|--------|-------|
| Red | 640 |
| Blue | 787 |
| Yellow | 605 |
| Green | 695 |
| White | 553 |
| Black | 518 |

Code TSense(TPad)

- In the loop(), program the following
 1. Write the sensed Tpad values to “tpadvalue”
 2. Use Serial.println to display the values in the serial monitor

```
void loop() {  
    tpadvalue = analogRead(TPAD_PIN);  
    Serial.println(tpadvalue);  
    if (tpadvalue > 630 && tpadvalue < 650) {  
        digitalWrite (LED_PIN, HIGH);  
    }  
    else {  
        digitalWrite(LED_PIN, LOW);  
    }  
}
```

Code TSense(TPad)

3. Using conditional programming, if a button that has a value between 630 and 650 on the TPad is pressed (eg red), the LED will turn on, else the LED will turn off

```
void loop() {  
  tpadvalue = analogRead(TPAD_PIN);  
  Serial.println(tpadvalue);  
  if (tpadvalue > 630 && tpadvalue < 650) {  
    digitalWrite (LED_PIN, HIGH);  
  }  
  else {  
    digitalWrite(LED_PIN, LOW);  
  }  
}
```

Complete Program

```
#define TPAD_PIN A1
#define LED_PIN 12
int tpadvalue = 0;

void setup() {
  pinMode(TPAD_PIN, INPUT);
  pinMode(LED_PIN, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  tpadvalue = analogRead(TPAD_PIN);
  Serial.println(tpadvalue);
  if (tpadvalue > 630 && tpadvalue < 650) {
    digitalWrite(LED_PIN, HIGH);
  }
  else {
    digitalWrite(LED_PIN, LOW);
  }
}
```

Try it yourself!

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

- Connect and declare TSense (buzzer).
- Use pin13 for buzzer
- Use 2 separate button for LED and Buzzer to turn on and off.

Solution – TSense(TPad, buzzer, LED)

```
#define TPAD_PIN A1
#define LED_PIN 12
#define BUZZER_PIN 13
int analogvalue = 0;

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

void loop() {
  analogvalue = analogRead(TPAD_PIN);      // Turns the led/buzzer on/off
  Serial.println(analogvalue);
  if (analogvalue > 630 && analogvalue < 650) { // When left button is pressed
    digitalWrite(LED_PIN, HIGH);
  }
  if ( analogvalue > 595 && analogvalue < 615) { // When right button is pressed
    digitalWrite(LED_PIN, LOW);
  }
  if (analogvalue > 777 && analogvalue < 797) { // When up button is pressed
    tone(BUZZER_PIN, 2000);
  }
  if ( analogvalue > 685 && analogvalue < 705) { // When down button is pressed
    noTone(BUZZER_PIN);
  }
}
```