



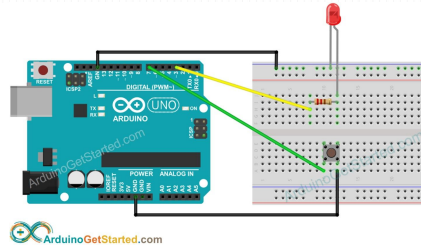
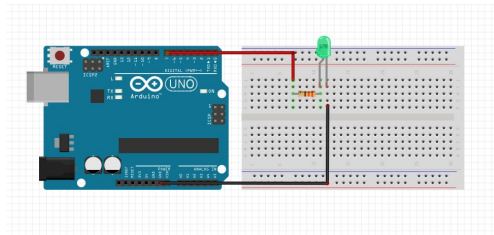
E-TEXTILES

The image features the text "E-TEXTILES" in a bold, blocky, sans-serif font. The letters are white with a dashed outline, giving them a stitched or embroidered appearance. This text is centered within a horizontal, slightly curved border that also has a dashed, stitched look, resembling the edge of a piece of fabric or a patch. The entire graphic is contained within a simple black rectangular frame.



Recap over Day 6: Into to Arduino

Blinking LED



Recap over Day 6: Intro to Arduino

Code

```
const int BUTTON_PIN = 7; // the number of the pushbutton pin
const int LED_PIN = 3; // the number of the LED pin

int buttonState = 0; // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(LED_PIN, OUTPUT);
  // initialize the pushbutton pin as a pull-up input:
  // the pull-up input pin will be HIGH when the switch is open and LOW
  // when the switch is closed.
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(BUTTON_PIN);

  // control LED according to the state of button
  if(buttonState == LOW) // If button is pressing
    digitalWrite(LED_PIN, HIGH); // turn on LED
  else // otherwise, button is not pressing
    digitalWrite(LED_PIN, LOW); // turn off LED
}
```

Recap over the task!

This was the code that you were supposed to adapt

Code

```
// Defining button pins
const int BUTTON1_PIN = 7; // the number of the pushbutton pin
const int BUTTON2_PIN = 8;
const int BUTTON3_PIN = 9;

// Defining LED pins
const int LED_RED_PIN = 3; // the number of the LED pin
const int LED_GREEN_PIN = 4;
const int LED_WHITE_PIN = 5;

// Creating variables for button states
int buttonState1 = 0; // variable for reading the pushbutton status
int buttonState2 = 0;
int buttonState3 = 0;

void setup() {
  // initialize the LED pins as outputs:
  pinMode(LED_RED_PIN, OUTPUT);
  pinMode(LED_GREEN_PIN, OUTPUT);
  pinMode(LED_WHITE_PIN, OUTPUT);
}
```

The following slides include the correct code for the Task.

Code

```
// initialize the pushbutton pin as
an pull-up input:
// the pull-up input pin will be
HIGH when the switch is open and LOW
when the switch is closed.
pinMode(BUTTON1_PIN, INPUT_PULLUP);
pinMode(BUTTON2_PIN, INPUT_PULLUP);
pinMode(BUTTON3_PIN, INPUT_PULLUP);
}

void loop() {
  // read the states of the
pushbuttons:
  buttonState1 =
digitalRead(BUTTON1_PIN);
  buttonState2 =
digitalRead(BUTTON2_PIN);
  buttonState3 =
digitalRead(BUTTON3_PIN);

  // control LED according to the
state of button
```

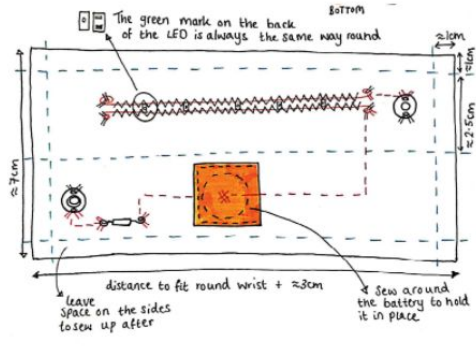
```
// control LED according to the state of button

//if button 1 and button 2 are pressed together - Red
LED lights up
if ((buttonState1 == LOW) && (buttonState2 == LOW))
    digitalWrite(LED_RED_PIN, HIGH);

//if button 3 or button 2 are pressed - Green LED
lights up
else if ((buttonState3 == LOW) || (buttonState2 ==
LOW))
    digitalWrite(LED_GREEN_PIN, HIGH);

//if button 1 by itself is pressed - White LED lights
up
else if (buttonState1 == LOW)
    digitalWrite(LED_WHITE_PIN), HIGH);

//if nothing is pushed - no LED lights up
else // otherwise, button
is not pressing
    digitalWrite(LED_RED_PIN, LOW); // turn off LED
    digitalWrite(LED_GREEN_PIN, LOW);
    digitalWrite(LED_WHITE_PIN, LOW);
}
```



This session, we looked over the construction of the LED bracelet and how it uses a simple circuit with LEDs in parallel

- Watch the video for more detail