

Creative Electronics

Arduino Practice

1-06-2016

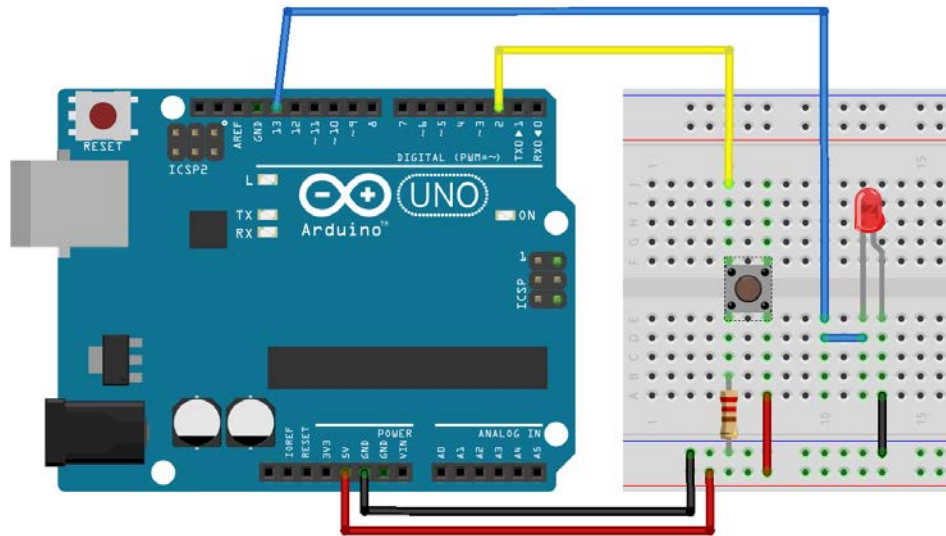
TU / **e**

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

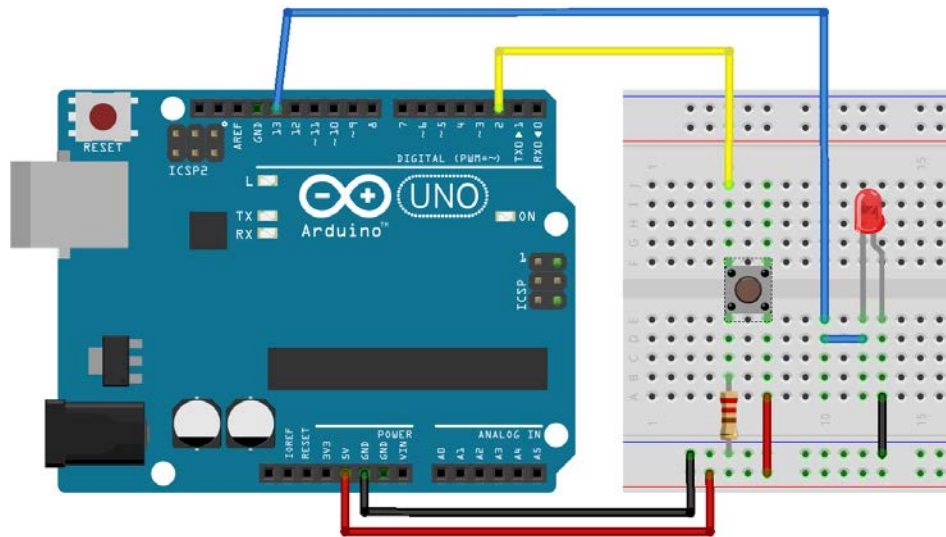
Button

- Button presses should switch the led on and off.
 - (1st press: LED on, next press: LED off, etc...)



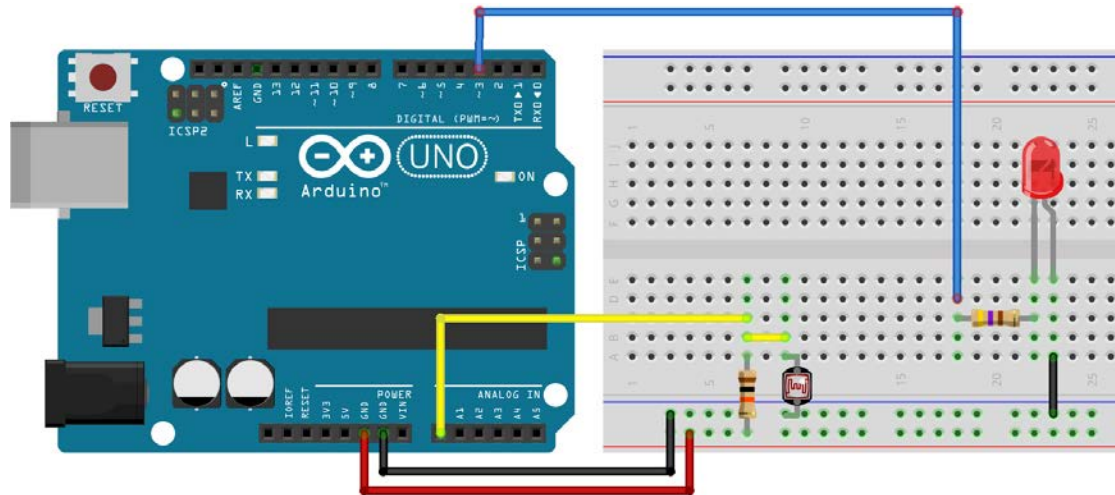
More Button

- 1st button press switches the LED on for 10 seconds, second press within these 10 seconds switches the LED off again.



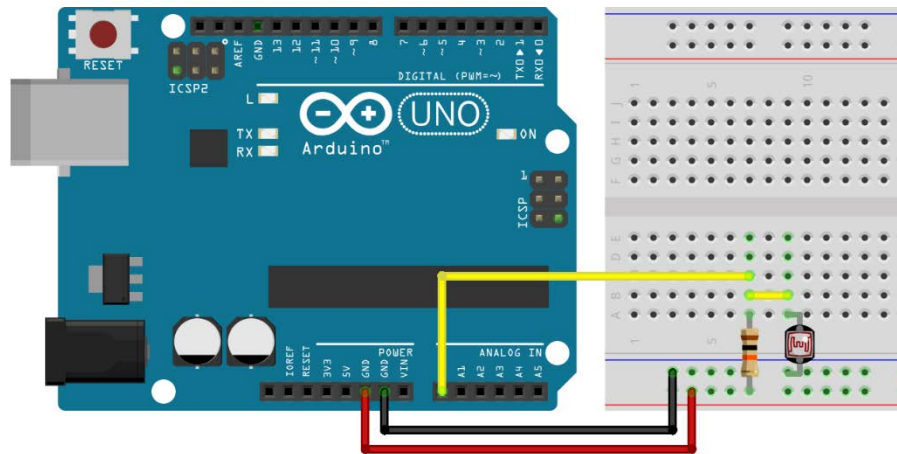
Fade LED

- Lowering the light intensity should trigger the led to fade on and off. You can use the 'fade_good' example as a starting point.



Communication (includes using Processing)

- Measure light intensity and communicate the measured value to a Processing program to display a changing color background.



Processing skeleton

```
// use serial communication
import processing.serial.*;

// character indicating end-of-line
byte linefeed = 10;
// background color
int bckGnd = 0;

// the serial connection
Serial myPort;

void setup() {
  // canvas size
  size(500, 350);
  // create serial connection
  myPort = new Serial(this, Serial.list()[index], 115200);
}

void draw() {
  background(bckGnd);
  if (checkForNewData()) {
    // handle new data`
  }
}
```

```
// check if new data is available and translate it to the data array
boolean checkForNewData() {
  // if data available
  if (myPort.available() > 0) {
    // you know you received data, so read it
    .....
    // process it
    ....
    return true; // say we have a value
  }
}
return false; // say we do not have a value
}
```

Button to control a LED

```
// constants won't change. They're used here to
// set pin numbers:
const int buttonPin = 2; // the number of the pushbutton pin
const int ledPin = 13; // the number of the LED pin

// variables will change:
int buttonState = LOW; // variable for reading the pushbutton status
int ledState = LOW; // variable remembering the LED status LOW = off, HIGH = on

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}
```

Button to control a LED

```
void loop() {  
  // read the state of the pushbutton value:  
  buttonState = digitalRead(buttonPin);  
  
  // check if the pushbutton is pressed.  
  // if it is, the buttonState is HIGH:  
  if (buttonState == HIGH) {  
    // change LED:  
    if (ledStatus == LOW) {  
      ledStatus = HIGH;  
    } else {  
      ledStatus = LOW;  
    }  
    digitalWrite(ledPin, ledStatus);  
  }  
}
```

That happens more than once....

Button LED Timer

```
long endTime = 0;

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

  // if the pushbutton is pressed.
  if (buttonState == HIGH) {
    // if timer is already running
    if (endTime > millis()) {
      // switch off led and stop timer
      digitalWrite(ledPin, LOW);
      endTime = 0;
    } else {
      // switch on LED and start timer
      digitalWrite(ledPin, HIGH);
      endTime = millis() + 10000;
    }
  }
}
```

Fade LED

- **Note:**
 - **Calibrate LED level at start, otherwise the light detection will only work for a specific condition.**
 - **Tip: The program can be in one of two states: detecting or fading.**