Terrain Sensing Shoes

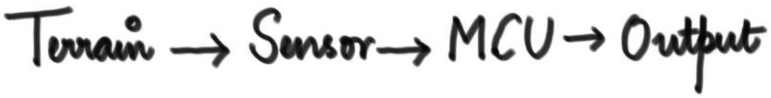
ON THE PRINCIPLE OF COLOURS OF THE TERRAIN

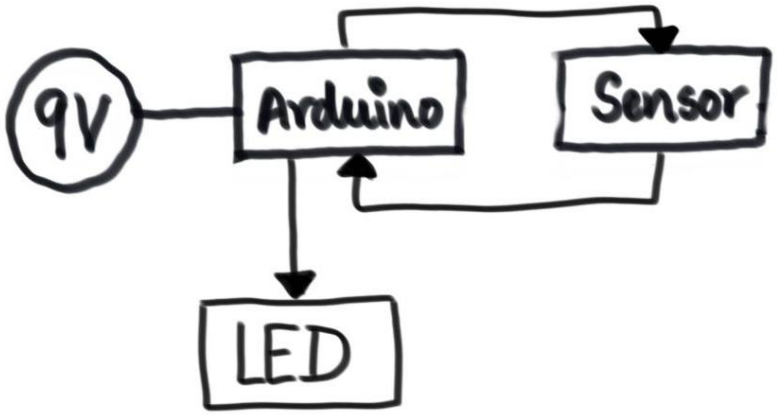
Apala Chakrabarti | Dhruvi Singh |Mayank Shekar |
Sukarn Pahuja | Tony Jacob

UNDER THE GUIDANCE OF
DR. ELIZABETH RUFUS

Overview

As a group, we had been assigned with the task of making a wearable device that requires any basic sensor. With the proper guidance from Professor Elizabeth Rufus, we planned to proceed with terrain sensing shoes that senses terrain based on the colour that is emitted. For this, we used a colour sensor to match out the values of the RGB scale of different terrains namely; sand, snow, and grass. We calibrated the values onto a microcontroller which when detected, gave the output of the respective terrain as programmed. Here we use LED strips as the output. This system of sensor, microcontroller, and LED are implemented onto a shoe as to achieve this on a wearable scale.





Apparatus required

1. ***SHOES***

2. ***LED'S (3 COLOURS)- used to deliver the output***

3. ***ARDUINO GENUINO UNO- used as a microcontroller***

4. ***9V CELLS***

5. ***COLOUR SENSOR –used to detect colour by its wavelength***

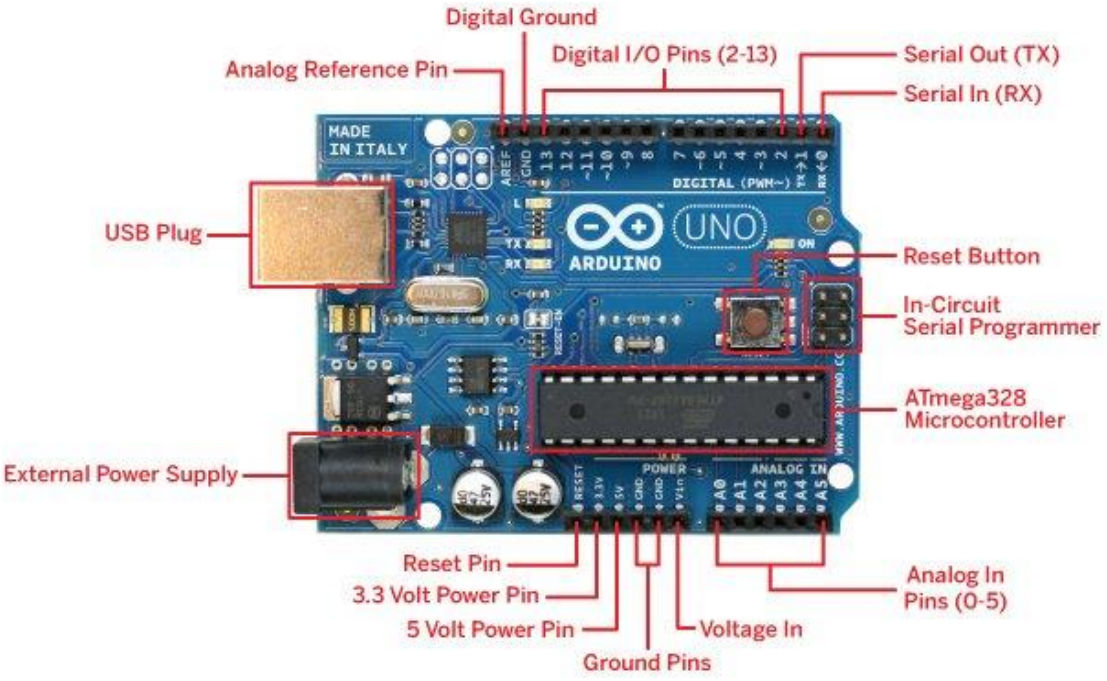
6. ***Jump wires (3 types):*** ***- used to supply current***

a) MALE-MALE

b) FEMALE-FEMALE

c) MALE-FEMALE

Arduino Uno Pin Configuration



Powering the Arduino Uno:

There are totally three ways by which you can power your Uno:

USB Jack: Connect the mini USB jack to a phone charger or computer through a cable and it will draw power required for the board to function

Vin Pin: The Vin pin can be supplied with a unregulated 6-12V to power the board. The on-board voltage regulator regulates it to +5V

+5V Pin: If you have a regulated +5V supply then you can directly provide this o the +5V pin of the Arduino.

Input/output:

The 14 digital input/output pins can be used as input or output pins by using pinMode(), digitalRead() and digitalWrite() functions in arduino programming. Each pin operate at 5V and can provide or receive a maximum of 40mA current, and has an internal pull-up resistor of 20-50 KOhms which are disconnected by default. Out of these 14 pins, some pins have specific functions as listed below:

- **Serial Pins 0 (Rx) and 1 (Tx):** Rx and Tx pins are used to receive and transmit TTL serial data. They are connected with the corresponding ATmega328P USB to TTL serial chip.
- **External Interrupt Pins 2 and 3:** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value.
- **PWM Pins 3, 5, 6, 9 and 11:** These pins provide an 8-bit PWM output by using analogWrite() function.
- **SPI Pins 10 (SS), 11 (MOSI), 12 (MISO) and 13 (SCK):** These pins are used for SPI communication.
- **In-built LED Pin 13:** This pin is connected with an built-in LED, when pin 13 is HIGH – LED is on and when pin 13 is LOW, its off.

Along with 14 Digital pins, there are 6 analog input pins, each of which provide 10 bits of resolution, i.e. 1024 different values. They measure from 0 to 5 volts but this limit can be increased by using AREF pin with analog Reference() function.

- Analog pin 4 (SDA) and pin 5 (SCA) also used for TWI communication using Wire library.

Arduino Uno has a couple of other pins as explained below:

- **AREF:** Used to provide reference voltage for analog inputs with analogReference() function.
- **Reset Pin:** Making this pin LOW, resets the microcontroller.

TC3200 Colour Sensor

The TCS3200 programmable colour light-to-frequency converters that combine configurable silicon photodiodes and a current-to-frequency converter on a single monolithic CMOS integrated circuit. The output is a square wave (50% duty cycle) with frequency directly proportional to light intensity (irradiance). The full-scale output frequency can be scaled by one of three pre-set values via two control input pins. Digital inputs and digital output allow direct interface to a microcontroller or other logic circuitry. An output enables (OE) places the output in the high-impedance state for multiple-unit sharing of a microcontroller input line.

In the TCS3200, the light-to-frequency converter reads an 8 x 8 array of photodiodes. Sixteen photodiodes have blue filters, 16 photodiodes have green filters, 16 photodiodes have red filters, and 16 photodiodes are clear with no filters.

The four types (colours) of photodiodes are interdigitated to minimize the effect of non-uniformity of incident irradiance. All photodiodes of

the same colour are connected in parallel. Pins S2 and S3 are used to select which group of photodiodes (red, green, blue, clear) are active. Photodiodes are 110 µm x 110 µm in size and are on 134-µm centers.

Absolute Maximum Ratings over operating free-air temperature range (unless otherwise noted)

Supply voltage, VDD = 6 V

Input voltage range, all inputs, VI = –0.3 V to VDD + 0.3 V

Operating free-air temperature range, TA =–40°C to 85°C

Storage temperature range =–40°C to 85°C

Solder conditions in accordance with JEDEC J–STD–020A, maximum temperature =260°C

Recommended Operating Conditions:

High-level Input voltage-VDD=2.7V to 5.5V

Low-level input voltage-2.7V to 5.5V

Operating free-air temperature:- 40 to 70 degree Celsius

1.Power supply considerations

Power-supply lines must be decoupled by a 0.01-µF to the 0.1-µF capacitor with short leads mounted close to the device package.

2.Input interface

A low-impedance electrical connection between the device OE pin and the device GND pin is required for

improved noise immunity. All input pins must be either driven by a logic signal or connected to VDD or GND —they should not be left unconnected (floating).

3.Output interface

The output of the device is designed to drive a standard TTL or CMOS logic input over short distances. If lines greater than 12 inches are used on the output, a buffer or line driver is recommended. A high state on Output Enable (OE) places the output in a high-impedance state for multiple-unit sharing of a microcontroller input line.

4.Power down

Powering down the sensor using S0/S1 (L/L) will cause the output to be held in a high-impedance state. This is similar to the behavior of the output enable pin, however powering down the sensor saves significantly more power than disabling the sensor with the output enable pin.

5.Photodiode type (colour) selection

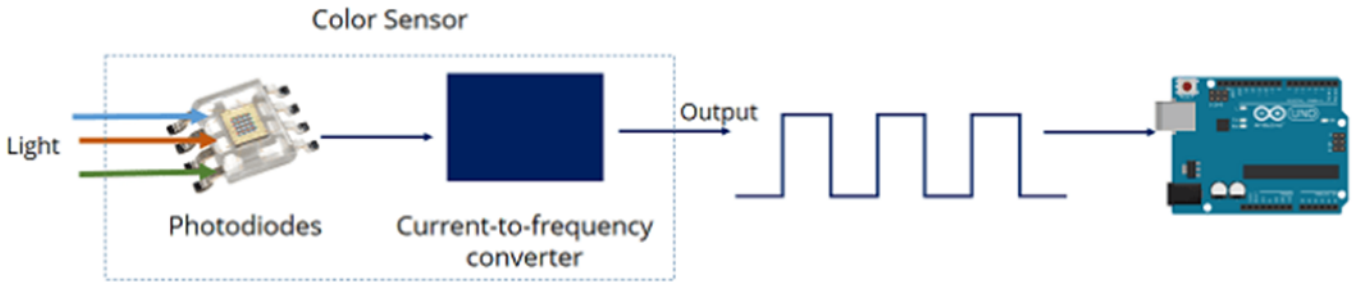
The type of photodiode (blue, green, red, or clear) used by the device is controlled by two logic inputs, S2 and S3.

6.Output frequency scaling

Output-frequency scaling is controlled by two logic inputs, So and S1. The internal light-to-frequency converter generates a fixed-pulse width pulse train. Scaling is accomplished by internally connecting the pulse-train output of the converter to a series of frequency dividers. Divided outputs are 50%-duty cycle square waves with relative frequency values of 100%, 20%, and 2%. Because division of the output frequency is accomplished by counting pulses of the principal internal frequency,

the final-output period represents an average of the multiple periods of the principal frequency.

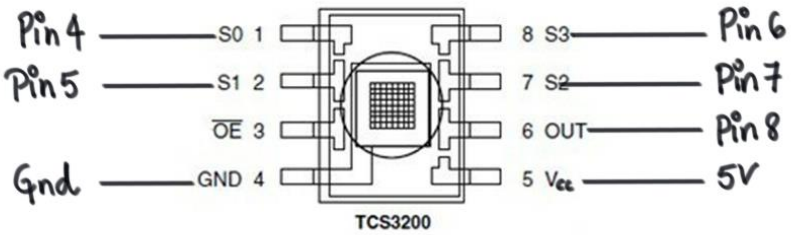
The output-scaling counter registers are cleared upon the next pulse of the principal frequency after any transition of the S0, S1, S2, S3, and OE lines. The output goes high upon the next subsequent pulse of the principal frequency, beginning a new valid period. This minimizes the time delay between a change on the input lines and the resulting new output period. The response time to an input programming change or to an irradiance step change is one period of new frequency plus 1 µs. The scaled output changes both the full-scale frequency and the dark frequency by the selected scale factor. The frequency-scaling function allows the output range to be optimized for a variety of measurement techniques. The scaled-down outputs may be used where only a slower frequency counter is available, such as the low-cost microcontroller, or where period measurement techniques are used.



MECHANICAL INFORMATION

This SOIC package consists of an integrated circuit mounted on a lead frame and encapsulated with an electrically nonconductive clear plastic compound. The TCS3200 has an 8 × 8 array of photodiodes with a total size of 1 mm by 1 mm. The photodiodes are 110 µm × 110 µm in size and are positioned on 134 µm centers.

Connections



The Arduino code

The arduino code logic is divided into two parts. There are two codes. One code reads the RGB values for various colours. This data is then implemented in the second code to control the LEDs. The sequential order of the process involves is as follows.

1. The connections between the arduino and the sensor is made.
2. The arduino code is developed. The code is uploaded to the memory of the arduino from the computer device.
3. The arduino powers the sensor.
4. Coloured objects are placed in front of the sensor.
5. The sensor reads the RGB values of the colours placed
6. The sensor sends the digital signals to the arduino (this is treated as the input signal)
7. The serial monitor on the computer screen shows the RGB values of the colours respectively.
8. The RGB values for the colours are noted down.
9. The second code for the arduino is developed.
10. The code is uploaded to the arduino memory.
11. The colour is sensed by the TC3200 module and the signals are sent back to the arduino.

12. The arduino checks the values of the RGB and compares it with the previously noted values.
13. According to the conditions, the arduino powers the specific pin connected to the suitable coloured LEDs.
14. The shoe is then lit up with the respective coloured LED.

CODE 1 – Finding the required RGB values

To select the colour read by the photodiode, you use the control pins S2 and S3. As the photodiodes are connected in parallel, setting the S2 and S3 LOW and HIGH in different combinations allows you to select different photodiodes.

Photodiode type	**S2**	**S3**
Red	LOW	LOW
Blue	LOW	HIGH
No filter (clear)	HIGH	LOW
Green	HIGH	HIGH

From the datasheet, we get the above conventions.

We read the colours after pre-defining these constraints.

Example:

```
// Setting RED (R) filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);

//for leds
  pinMode(rpin,OUTPUT);
  pinMode(gpin,OUTPUT);
  pinMode(bpin,OUTPUT);

 // Reading the output frequency
  redFrequency = pulseIn(sensorOut, LOW);

  // Printing the RED (R) value
  Serial.print("R = ");
  Serial.print(redFrequency);
  delay(100);
```

Similarly, we write the code for green and blue. Once the signal is detected, this code gives an output of RGB on the serial monitor. We found the range of the frequency for each colour as sensed using the TCS3200.

COLOUR	LOWER LIMIT	UPPER LIMIT

RED	50	80
GREEN	49	90
BLUE	50	80

The above tabulated values are experimental values. We further used these values to detect the colour and power up the LEDs.

Code 2- mapping the frequencies

Example:

```
// Setting RED (R) filtered photodiodes to be read
  digitalWrite(S2,LOW);
  digitalWrite(S3,LOW);
 // Reading the output frequency
  redFrequency = pulseIn(sensorOut, LOW);
  // Remaping the value of the RED (R) frequency from 0 to 255
  // We replace it with the values we noted from the previous code. Here's an example:
  // redColour = map(redFrequency, 70, 120, 255,0);
  redColour = map(redFrequency, XX, XX, 255,0);
// Here we replace values from the previous table respective to each colour.
  // Printing the RED (R) value
  Serial.print("R = ");
```

```
Serial.print(redColour);
delay(100);
```

Similarly, we write the code for green and blue.

Adding to code 2- the command to light the LEDs up

```
// once the colour is detected, its respective pin is powered, hence lighting the LEDs up.
if(redColour > greenColour && redColour > blueColour){
    digitalWrite(rpin,HIGH);
    digitalWrite(gpin,LOW);
    digitalWrite(bpin,LOW);
    Serial.println(" - RED detected!");
```

Similarly, we write the code for blue and green.

Fabrication

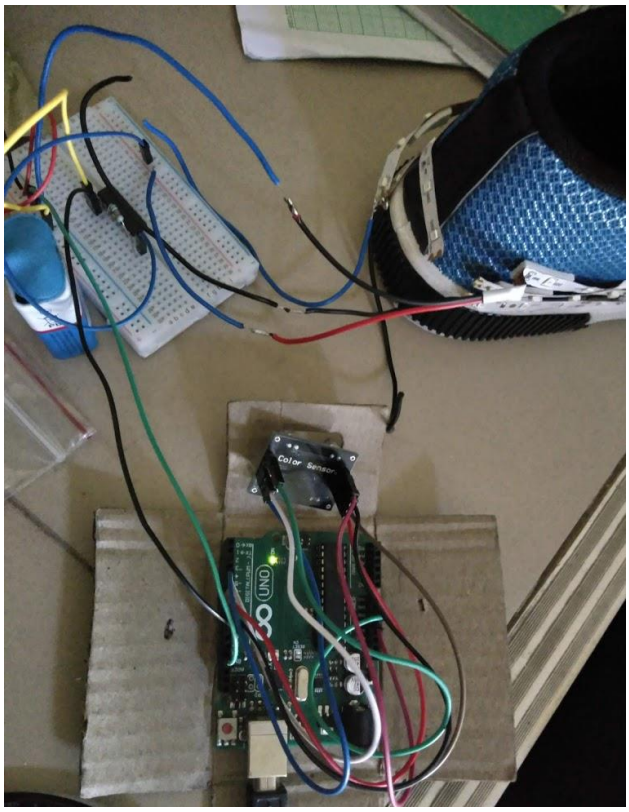
We created a housing for the arduino and the sensor using cardboard. We used the jumper wires to make the connections between the arduino and the sensor. the three output pins from the arduino for the three led strips were connected to an AND gate onto the breadboard. The two inputs for the and gate are from the 9V battery and the output pin from the arduino. The output of the and gate when high, powers up the respective coloured LED. The arduino is directly powered by

another 9V cell. The 9V cell powering the arduino is placed outside the housing to restrict damage due to excessive weight.

THE SHOE FITTED WITH LED STRIPS

https://photos.app.goo.gl/q77RTPgBopmM9AkQ7

(THE HOUSING)



THE CONNECTIONS

https://photos.app.goo.gl/o9EbG3xp2kMWQGQc8

(THE WORKING)

Observations

When a green terrain is sense, the green led strips light up. When the blue colour is sensed, the blue led strip lights up. Similarly, when the red colour is sensed the red led strips glow.

Applications

The terrain-sensing shoes can be used in various places such as the military, the fashion industry. It can be further modified to have various other functions such as a GPS connection, other lights of various colours, vibrations etc.