DIY Reaction Timer

Introduction

Mental chronometry is the study of how fast humans react to different inputs. It takes a few hundred milliseconds for the signal to get from your eyes, to your brain, out to your limbs to respond. The reaction timer is a great project to demonstrate this time delay. It also makes for a fun game between friends!

According to the team at Human Benchmark\(^1\), the average human reaction time is about 215 ms. How fast are you?

The principal of the reaction timer is simple: when the user sees the light turn on, press the button! A microcontroller is perfect for this because it can time milliseconds very accurately.

For this project, we will use an Arduino to be the time-keeper. An Arduino is a small, low-cost, and fast microcontroller that is capable of performing command / instructions at a rate of 16 MHz or roughly 62.5 ns per instruction.

Wiring / Hook-up Guide (Fritzing)

The wiring for this circuit is simple, it requires a single LED and a single button. Connect the LED between pin 12 and GND using the solderless breadboard. Remember that each row of 5 holes are connected together by an internal metal clip. Make sure that the LED leg and the connecting wire are in the same row. Also, LEDs have a polarity. Make sure that the short leg of the LED is connected to GND.

Similarly, connect the button up between pins 7 and GND.

Example Code

https://codebender.cc/sketch:62303

```c
int sound = false;    // enables the buzzer

int ledPin = 12;      // connect LED between pin 12 and GND
int buttonPin = 7;     // connect push button between pin 7 and GND
int gameNum = 0;       // counter for the games played
unsigned long waitTime; // Random "waiting" time before turning on the light
unsigned long zeroTime; // Zero reference time
unsigned long reactTime; // Reaction Time

void setup()
{
  Serial.begin(9600);  // setups up communication to talk back to the computer

  pinMode(ledPin, OUTPUT);  // sets up the ledPin to be an OUTPUT
  pinMode(buttonPin, INPUT_PULLUP);  // Sets the pull-up resistor for the button
  digitalWrite(ledPin, LOW);    // sets pin LOW (GND)

  // prints out the header.
  Serial.print("Iter");
  Serial.print("\t");
  Serial.print("React (ms)");
  Serial.println();
  Serial.println("================");
  Serial.println();
}

void loop()
{
  randomSeed(analogRead(A5));    // Get noise to seed the random number generator
  // Use an un-used pin for the random noise.

  waitTime = random(2000, 3500); // randomTime from 2 to 3.5 seconds
  delay(waitTime);               // delay randomTime
  digitalWrite(ledPin, HIGH);    // turn on LED!

  zeroTime = millis();            // set zeroTime reference

  while(digitalRead(buttonPin) == HIGH) // holding loop until button is pressed.
  {
  }
}
reactTime = millis() - zeroTime; // calculation of reaction time

digitalWrite(ledPin, LOW); // turn off LED!

// Display information to Serial Monitor //

Serial.print(gameNum);
Serial.print(" ");
Serial.print(reactTime, 1);
Serial.println();
delay(1000); // short delay before starting again.

}