Arduino Assignement Report

Spring 2018

Alexandre Allani

April 28, 2018

This is the report of my Arduino Assignement

In the zipfile, my arduino code is named _20186024_arduino.ino because arduino doesn't accept file name starting with numbers.

Arduino Starter Kit ID: 14

Maximum bit rate: As you can see in my code, the value DELAY is basically on what I rest upon to calculate the bit rate. I delayed the main function, which is the function "choose" two times by $\frac{\text{DELAY}}{2}$. So it takes $2 * \frac{\text{DELAY}}{2} = \text{DELAY} = 2\text{ms}$ to send 1 bit. Then I have a theoretical maximum bitrate of **500 bits/sec**.

Calibration of sensor: Concerning sensor's calibration, this is done each time the Arduino is powered on.
In my Arduino code, in the setup() function, we let 7 seconds to the sensor to calibrate the high value, and 7 seconds to calibrate the low value. Afterward, the sensor is not calibrated anymore, so that we don't loose bitrate.
However, knowing that values will probably change, if we have negative values or too high values after the map function (**choose() function, line 57 or 71**), meaning that the value is out of bound, we're invoking the function translate. This will just map the detected value to the highest value, or the lowest value according to the situation.
However, I wrote another code for this project, where the sensor calibrates after each loop of loop() function. I just took the calibration part of the setup() function which is :

```
int a = millis();
while (millis()-a < 7000) {
        // record the maximum sensor value
        sensorValue = analogRead(A0);
        if (sensorValue > sensorHigh) {
                sensorHigh = sensorValue;
        }
}
digitalWrite(ledPin, LOW);
while (millis()-a < 14000) {
        // record the maximum sensor value
        sensorValue = analogRead(A0);
        if (sensorValue < sensorLow) {
                sensorLow = sensorValue;
        }
}
```

And I placed this in the loop() function. (I changed little things in the code above and what you can find in the setup() function. For instance, the condition of the while loop is different. Indeed, millis() count the time since the start of Arduino board. Then in order to know if 7 seconds passed, we're just saving millis() in a variable).
However, this totally kills the bitrate. The time spent to calibrate the sensor is much more higher than the time taken to send the message. Hence for my ID, we I have a bitrate of 9bit/14second ie, 0.64 bits/sec, which is incredibly low. So that's why I kept the calibration part in the setup() function.

The most difficult thing: I already did a lot of Arduino last year for a project, and I've also coded a lot in C++. So the coding part was not really difficult for me. However, I lost like one hour just because the sensor is quite sensitive to light.I did all my experiments in a dark place, i.e. without much noises. That's why I have a quite high bitrate. However, in a noisy place, I had to increase the DELAY value, so that the sensor could keep up. I think it was a lot easier than the Android assignment since I did not know how to code in Android.