

Addiel Martinez-Perez

Ruian Pan

Jihyun Lee

12/12/2016

ECE 110 Honors Lab Final Project:

Remote Key-less Entry(Lock) for Bicycle

Introduction)

Problem Description

A four year undergraduate student has an estimated 17.75% chance of having his or her bike stolen. With such a high rate of bike theft and the high number of people riding bikes on campus, improving the system and making it more convenient seemed very important for college students. Thus, the goal of our project was to build a key-less bicycle lock, designed to add fashion and automation into the rather outdated, but still an effective way of transportation: the bicycle. We intend to add a keyless system that works with an infrared remote and an alarm system for the bicycle lock, making it convenient and more deterrent to theft.

Design Concept

The basic structure of our design includes a car remote control system with a CPU, an infrared light receiver and transmitter, a command module, and a pushbutton. We certainly did not need something to "fire" a bicycle up, like what we would need to do with a car, so what we basically needed was a set of infrared equipment, an arduino board as the CPU and the command module,

a push button in the lab kit, a speaker as an alarm during emergency, and a circuit designed to make the bike even safer.

The bike lock is based on a normal bike lock, meaning one can also open it in a normal way in case of an emergency. The infrared transmitter and receiver work to deliver message from the remote to the system. The protection circuit stays with the bike lock itself in order to prevent the lock from being cut and the wheels from being taken off.

Although we also intend to create a retracting mechanism for the bike lock; that is, the bicycle will also carry a box that after the lock disengages will coil the lock's cable into the box, allowing for rapid unlocking of the bike and for faster "take off", to the destination of the user, we were unable to work on the part because the mechanics involved was extremely challenging and we had to design another locking mechanism or retrofit our design onto an existing lock.

The expected challenges that we predicted were that the protection circuit needs to be sturdy and sensitive, and that the Arduino board needs to be well detective and have clear communication with the remote.

Analysis of Components)

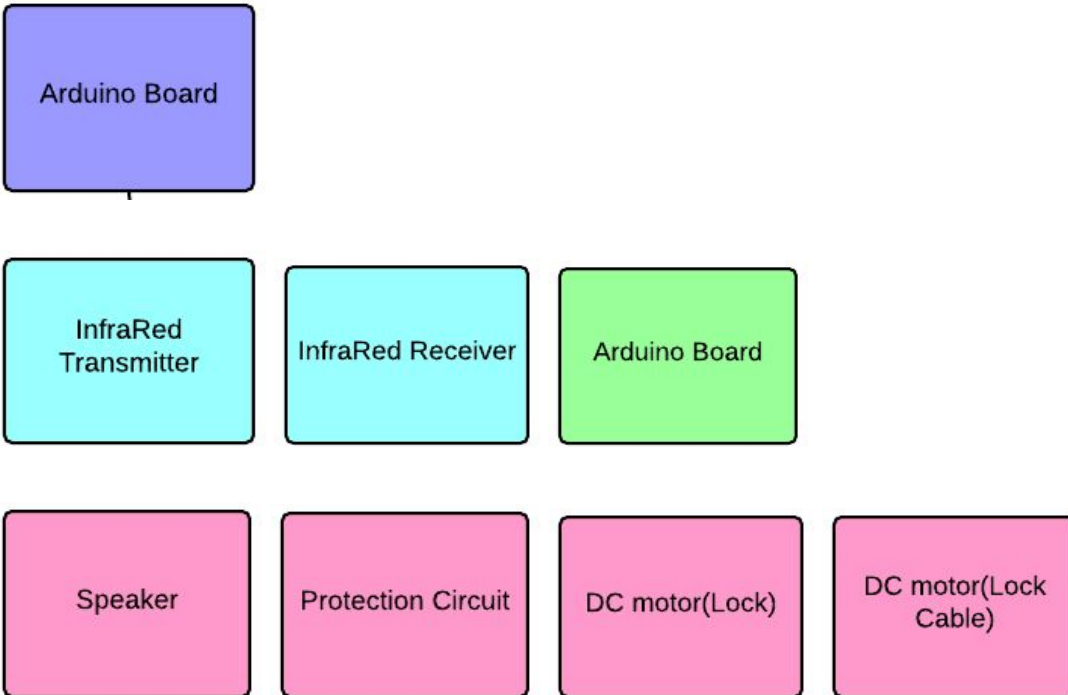
| Symbol | Parameter | Rating | Units |
|---------------|-----------------------|---------------|--------------|
| T_{OPR} | Operating Temperature | -40 to +85 | °C |
| T_{STG} | Storage Temperature | -55 to +100 | °C |

| | | | |
|--------------------------|----------------------------|---------------|-------------|
| T_{SOL} | Soldering Temperature | 260 for 5 sec | $^{\circ}C$ |
| Infrared Emitter | | | |
| I_F | Continuous Forward Current | 50 | mA |
| V_R | Reverse Voltage | 5 | V |
| I_{FP} | Peak Forward Current | 1 | A |
| P_D | Power Dissipation | 75 | mW |
| Infrared Detector | | | |
| V_{CEO} | Collector-Emitter Voltage | 30 | V |
| V_{ECO} | Emitter-Collector Voltage | 5 | V |
| I_C | Collector Current | 20 | mA |
| P_D | Power Dissipation | 100 | mW |

Motor

Design Description

-Block Diagram



-Circuit Schematic

```
void setup() {  
  // put your setup code here, to run once:  
  pinMode(A1, INPUT);  
  pinMode(11, OUTPUT);  
  Serial.begin(9600);  
}
```

```
void loop() {
  // put your main code here, to run repeatedly:
  int sensorValue = analogRead(A1);
  float voltage = sensorValue * (5.0 / 1023.0);
  if (voltage<=2)
  {
    analogWrite(11, 150);
  }
}

#define NOTE_G3 196
#define NOTE_C4 262
int signalPin =2;
int receivePin =3;
int val=0;
// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3
};
// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 4
};
void setup() {
  Serial.begin(9600);
  pinMode(signalPin, OUTPUT);
  pinMode(receivePin, INPUT);
  digitalWrite(signalPin, HIGH);
}
```

```
void loop() {
val = digitalRead (receivePin);
Serial.println(val);
if (val == HIGH){
  delay(0);
}
else{
{
// iterate over the notes of the melody:
for (int thisNote = 0; thisNote < 2; thisNote++) {
  // to calculate the note duration, take one second
  // divided by the note type.
  //e.g. quarter note = 1000 / 4, eighth note = 1000/8,
etc.
  int noteDuration = 1000 / noteDurations[thisNote];
  tone(8, melody[thisNote], noteDuration);
  // to distinguish the notes, set a minimum time
between them.
  // the note's duration + 30% seems to work well:
  int pauseBetweenNotes = noteDuration * 1.30;
  delay(pauseBetweenNotes);}
}
}
```

Conclusion and Self Assessment

References

"College students beware!," in *National Bike Registry*. [Online]. Available:

<https://www.nationalbikeregistry.com/college.html>. Accessed: Sep. 19, 2016.

O. 21, "Remote keyless entry systems overview - application note - maxim," in *Maxim Integrated*, 2016. [Online]. Available:

<https://www.maximintegrated.com/en/app-notes/index.mvp/id/1774>. Accessed: Sep.

<https://www.sparkfun.com/datasheets/Components/LTR-301.pdf>

<https://www.sparkfun.com/datasheets/Components/LTE-302.pdf>