

Controlling a Lock with an Arduino and Bluetooth LE

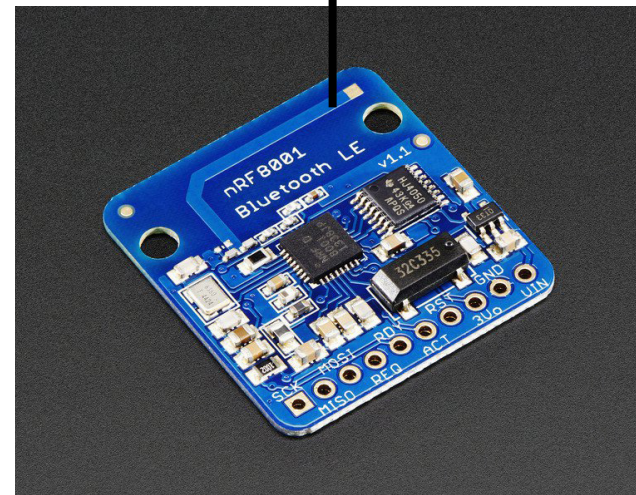
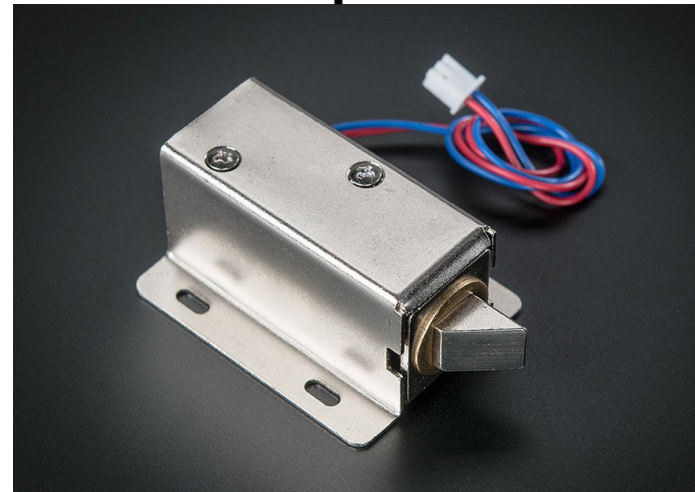
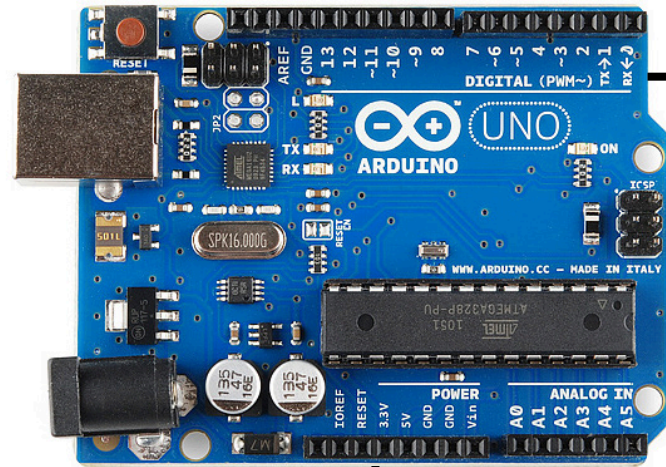
Midterm Project

Kristin Bessette - kristinbessette@gmail.com

This project uses an Arduino, a Bluetooth plugin, and few connections to control the the process. The end result is a system where you can enter in a code on your phone that's transferred over Bluetooth then unlocks the door.

This will help to solve a real-world problem: to prevent yourself from being locked out of your house if you forget your keys. It can be used by multiple people in a household, and has the ability to be shared on multiple doors and between users.

High-Level

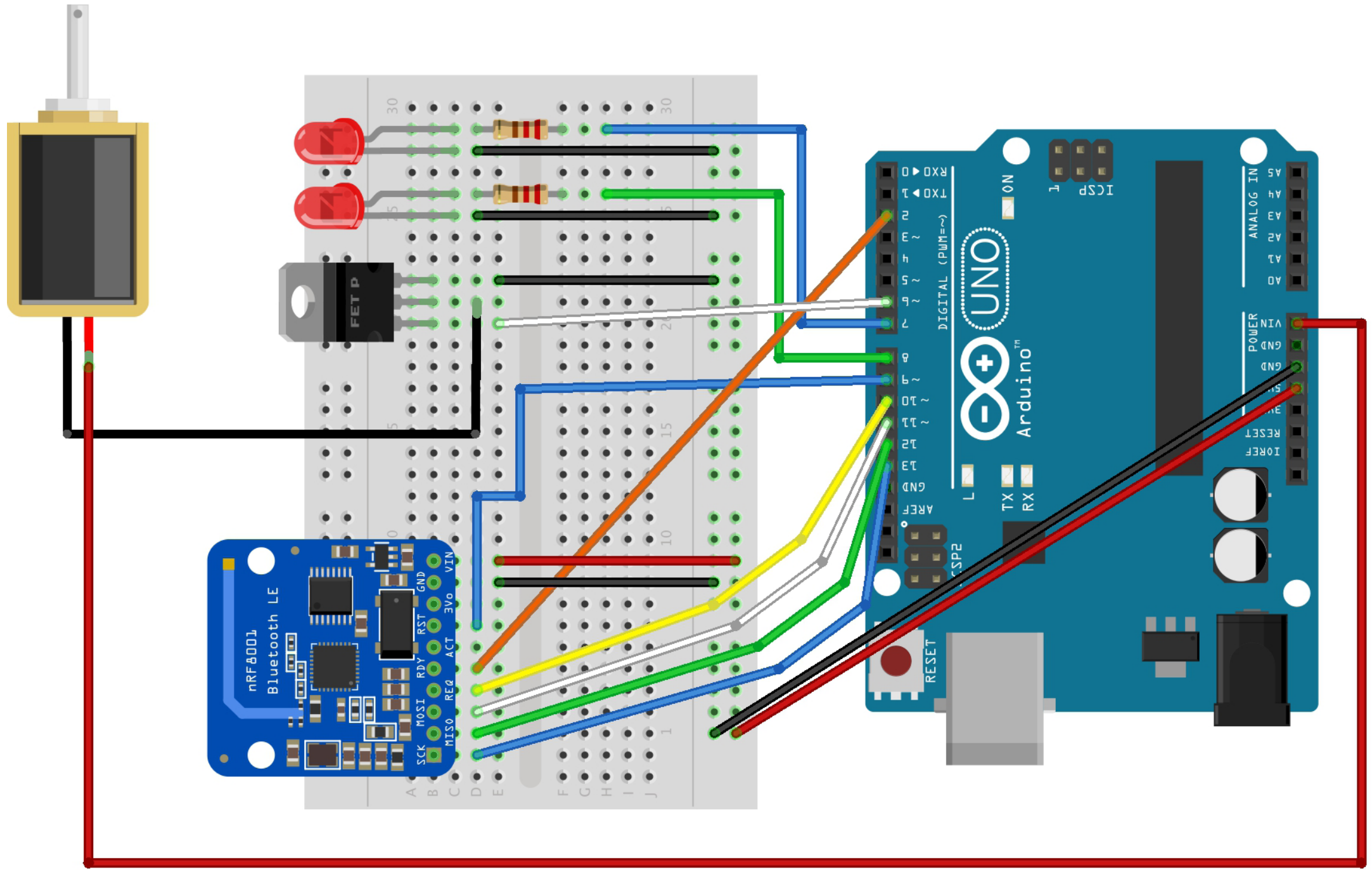


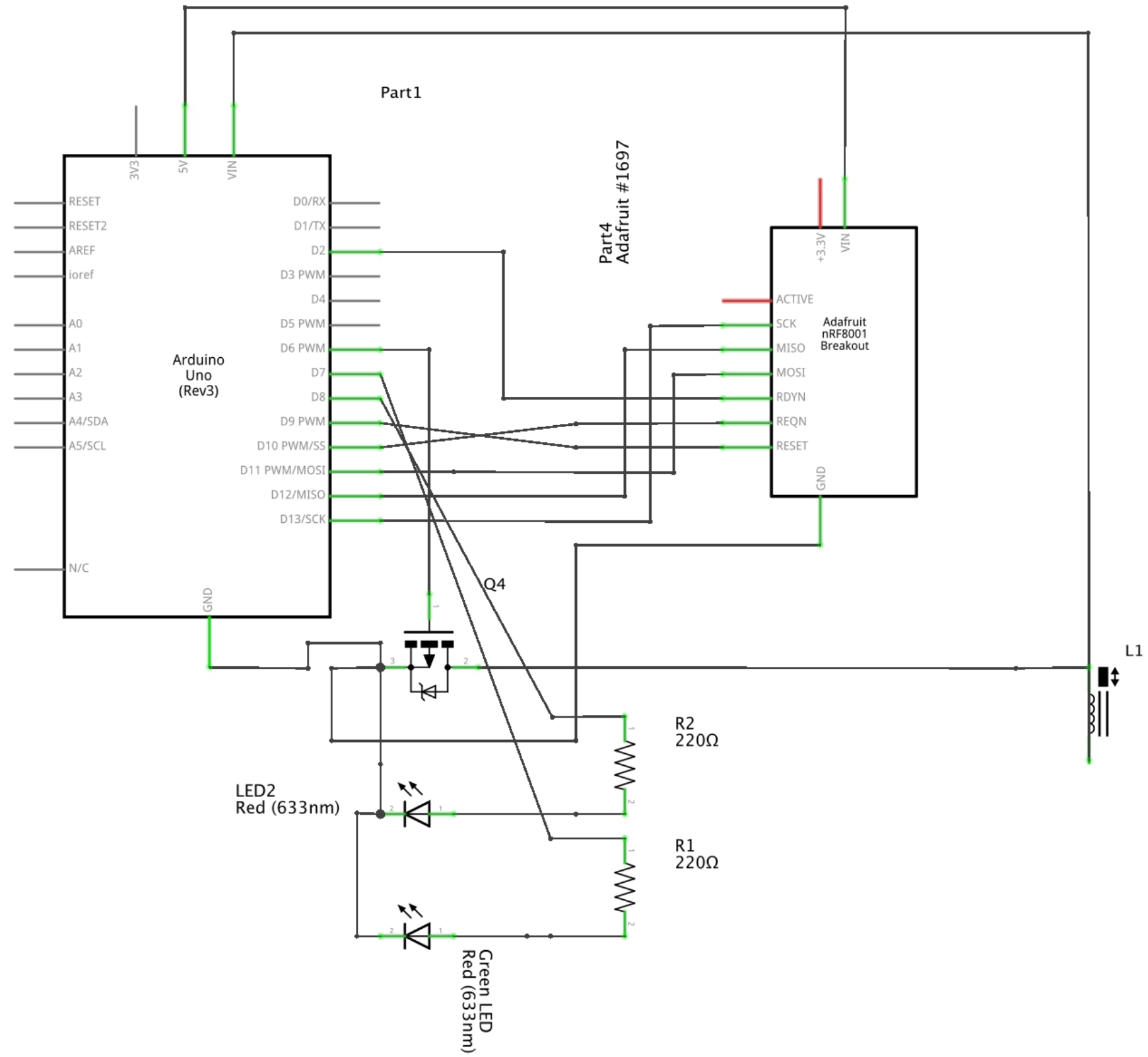
PARTS

- Arduino Uno
- Arduino Uno
- Solenoid Lock
- Bluetooth LE Breakout Board
- Darlington Transistor
- LED (Red)
- LED (Green)
- 220 Ohm Resistor
- 12 V 1000+ mA Power Supply
- Breadboard
- Jumper Wires

TOOLS

- Computer with Arduino IDE software





STEPS

- Wire the project together as shown in the diagram above.
- The solenoid draws a lot of current, can't run off USB power or 9V battery so we use the external power supply. The Darlington transistor allows us to switch the 12V power to the lock using an Arduino pin.
- The Adafruit Bluefruit LE hardware is Bluetooth Low Energy, so we're using it to do serial communication
- Upload the code below.

CREATE DEFINITIONS

Doorlock_viaBluetooth §

```
1 #include <SPI.h>
2 #include "Adafruit_BLE_UART.h"
3
4 #define LOCK_PIN 6
5 #define RED_LED_PIN 7
6 #define GREEN_LED_PIN 8 //Shows Red on diagram
7
8 // Connect CLK/MISO/MOSI to hardware SPI
9 // e.g. On UNO & compatible: CLK=13, MISO = 12, MOSI = 11
10 #define ADAFRUITBLE_REQ 10
11 #define ADAFRUITBLE_RDY 2 // interrupt pin 2 or 3 on UNO
12 #define ADAFRUITBLE_RST 9
13
14 long secret = 12345;
15 long openTime = 0;
16 // Status from the Bluefruit LE driver
17 int lastStatus = ACI_EVT_DISCONNECTED;
18
19 Adafruit_BLE_UART BTLEserial = Adafruit_BLE_UART(ADAFRUITBLE_REQ, ADAFRUITBLE_RDY, ADAFRUITBLE_RST);
20
```


SETUP

```
25  
26 void setup() {  
27   Serial.begin(9600);  
28   Serial.println(F("BLE Safe - Adafruit Bluefruit Low Energy Edition"));  
29   BTLEserial.begin();  
30  
31   pinMode(LOCK_PIN, OUTPUT);  
32   pinMode(RED_LED_PIN, OUTPUT);  
33   pinMode(GREEN_LED_PIN, OUTPUT);  
34   digitalWrite(LOCK_PIN, LOW);  
35   digitalWrite(RED_LED_PIN, LOW);  
36   digitalWrite(GREEN_LED_PIN, LOW);  
37 }  
38
```

LOOP

```
40 |
41 |
42 | void loop() {
43 |
44 |                                     // Tell the nRF8001 to do whatever it should be working on
45 |     BTLEserial.pollACI();
46 |
47 |     int status = BTLEserial.getState();
48 |
49 |     if (status != lastStatus) {
50 |         if (status == ACI_EVT_DEVICE_STARTED) {
51 |             Serial.println(F("* Advertising Started"));
52 |         }
53 |         else if (status == ACI_EVT_CONNECTED) {
54 |             Serial.println(F("* Connected!"));
55 |         }
56 |         else if (status == ACI_EVT_DISCONNECTED) {
57 |             Serial.println(F("* Disconnected or advertising timed out."));
58 |         }
59 |                                     // save for next loop
60 |         lastStatus = status;
61 |     }
62 |
63 |     if (status == ACI_EVT_CONNECTED) {
64 |
65 |                                     // see if there's any data from bluetooth
66 |         if (BTLEserial.available()) {
67 |             Serial.print("* ");
68 |             Serial.print(BTLEserial.available());
69 |             Serial.println(F(" bytes available from BTLE"));
70 |         }
71 |
72 |                                     // keeping u + code for compatibility with the serial api
73 |         if (BTLEserial.find("u")) {
74 |             int code = BTLEserial.parseInt();
75 |             openLock(code);
76 |         }
77 |
78 |     }
79 | }
```

LOOP CONTINUED

```
80 // close lock and reset lights after x seconds
81 if (openTime && millis() - openTime > 4000) {
82     resetLock();
83 }
84
85 }
86
87 void openLock(int code) {
88     openTime = millis();
89     if (code == secret) {
90         // open the lock
91         Serial.println("Code matches, opening lock");
92         digitalWrite(GREEN_LED_PIN, HIGH);
93         digitalWrite(RED_LED_PIN, LOW);
94         digitalWrite(LOCK_PIN, HIGH);
95         BTLEserial.println("unlocked");
96     } else {
97         // bad code, don't open
98         Serial.println("Invalid code " + code);
99         digitalWrite(RED_LED_PIN, HIGH);
100         BTLEserial.println("invalid code");
101     }
102 }
103
104 // closes the lock and resets the lights
105 void resetLock() {
106     // reset the lights
107     digitalWrite(RED_LED_PIN, LOW);
108     digitalWrite(GREEN_LED_PIN, LOW);
109     digitalWrite(LOCK_PIN, LOW); // close the lock
110     BTLEserial.println("locked");
111     openTime = 0;
112 }
```

NEXT STEPS TO TEST ON PHONE

- Install Adafruit Bluefruit LE Connect on your iPhone or iPad.

<https://itunes.apple.com/WebObjects/MZStore.woa/wa/viewSoftware?id=830125974&mt=8>

- Launch Bluefruit LE and choose “UART monitor”
- Once connected, you can send data to the Arduino and receive responses.
- Send u12345 (code). The green light should light up and the lock will open.
- Send an invalid code, u2222 (invalid code), the red light will light up.
- Note that the lock will ignore incoming data without the “u” (unlock) prefix.

INSTALL CORDOVA ON COMPUTER

- Install Cordova on your Mac
- Download and install NodeJS (<http://nodejs.org/>)
- Then install Cordova using NPM, which comes with NodeJS

Open a command prompt or Terminal, and type `npm install -g cordova`.

```
$ npm install -g cordova
```

BUILDING THE CODE IN CORDOVA / XCODE

Both the iOS platform and BluetoothSerial plugin need to be installed into the project before running.

```
$ cd BluetoothLock/cordova  
$ cordova platform add ios  
$ cordova plugin add https://github.com/don/BluetoothSerial  
$ cordova prepare  
$ open platforms/ios/Lock.xcodeproj
```

Open the Xcode project, and choose the target device in Xcode.

Build and deploy through Xcode using the below code.

Code will be uploaded to GitHub (Link to come)