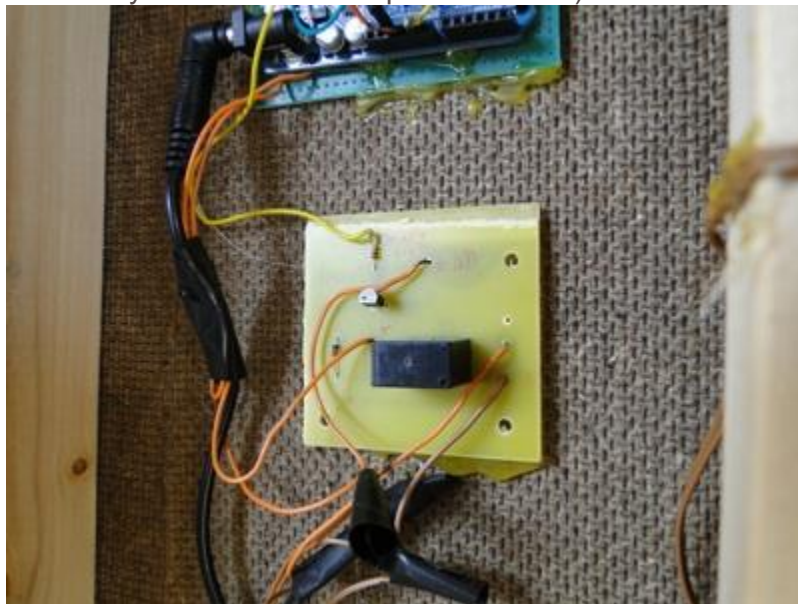


- **Arduino Door Lock**

Recently, while renovating at headquarters we discussed the purchase of combination door locks. You know-the ones with the keypad and the 4 digit PIN that unlocks the door. I decided to come up with a plan to build these locks. Rather than buy the expensive pre-manufactured door locks, here's how I did it:

First the parts. I knew I would use an Arduino as the main board and from there, the concept quickly unfolded. The Arduino would take the inputs from the various buttons, and each button that was used in the PIN would cause a math operation in the program on the Arduino. Very much like a "checksum." Once the checksum reached a certain value, the Arduino would output an unlock signal. The electronic door strike required much more current than an Arduino could supply, so a relay circuit was also needed. I ended up building two of them one for each of the doors we needed to lock. (My first prototype was a mess as you can see in the picture below.)





I built the first relay circuit from scratch. The circuit diagram is available in the Arduino inventors kit, but I don't recommend this route. In the second system, not only was it much more organized, but I also found a better solution to the relay circuit, our CPS156. I also used an Arduino ProtoShield instead of a bulky eurocard (see below).

The Build:

First lets gather the materials. If you don't already have an Arduino, I recommend the Arduino inventors kit and that you also check out our Arduino Video Learning Series to get started.

So you'll need:

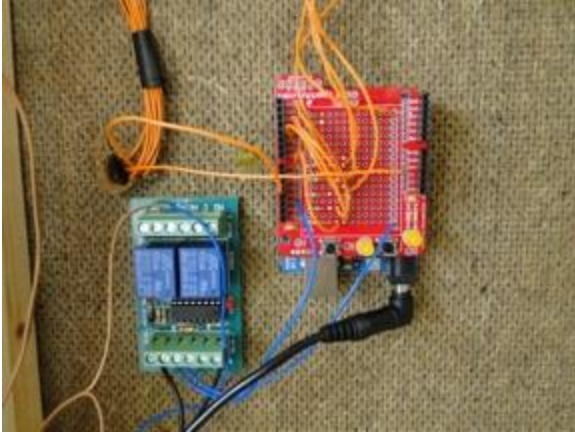
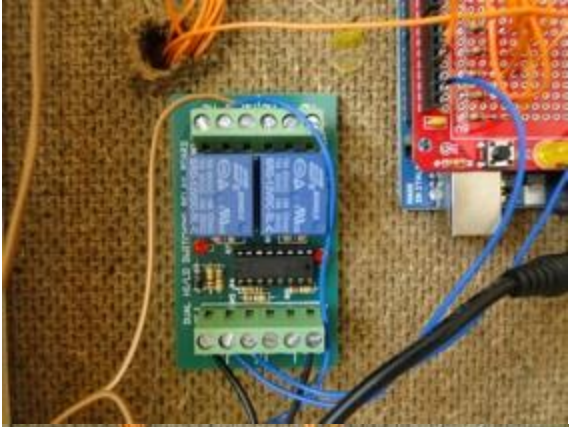
- Arduino Uno** x1
- Arduino ProtoShield** x1
- Keypad** x1
- Dual HI/LO relay board -- CPS156** x1

-Electronic door strike -- I found these on ebay for about \$15 Make sure you buy a 12V DC strike.

Some other small components such as:

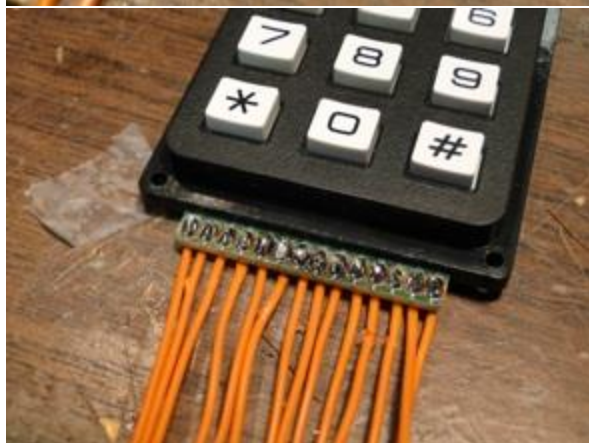
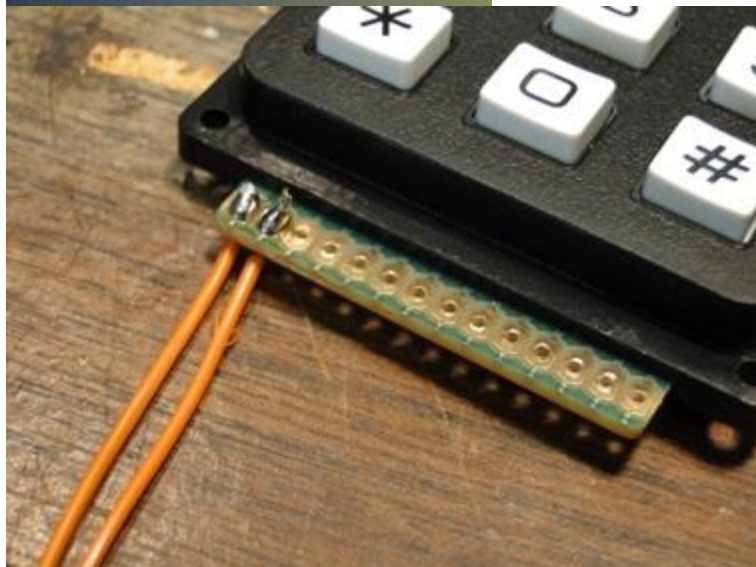
- A handful of 10k Ohm resistors
- A 330 Ohm resistor
- An LED
- A power supply, or two. You'll need 12VDC and at least 2A to power the door strike.

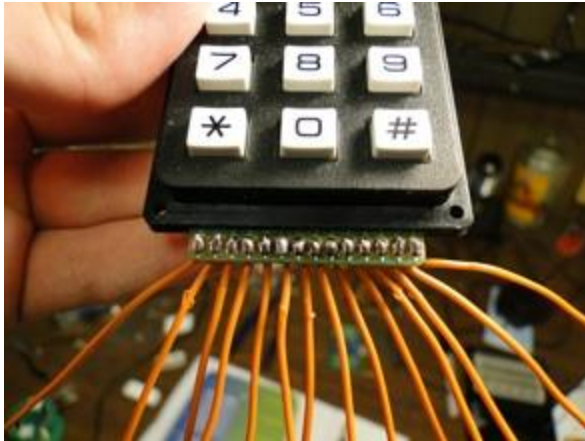
The first step is to solder wires to all the pins on the keypad. This is a common ground keypad and the pinouts are on the back of the packaging.



The Project Story

OUTPUT PIN NO.	SYMBOL
1	COM
2	
3	*
4	7
5	4
6	1
7	0
8	8
9	5
10	2
11	#
12	9
13	6
14	3

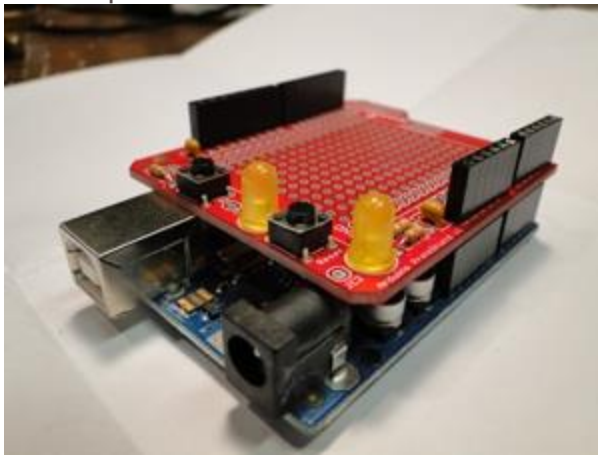




Next we move onto something a little more complex. Grab your Arduino ProtoShield! It is relatively easy to assemble. You can see how by [clicking here](#).

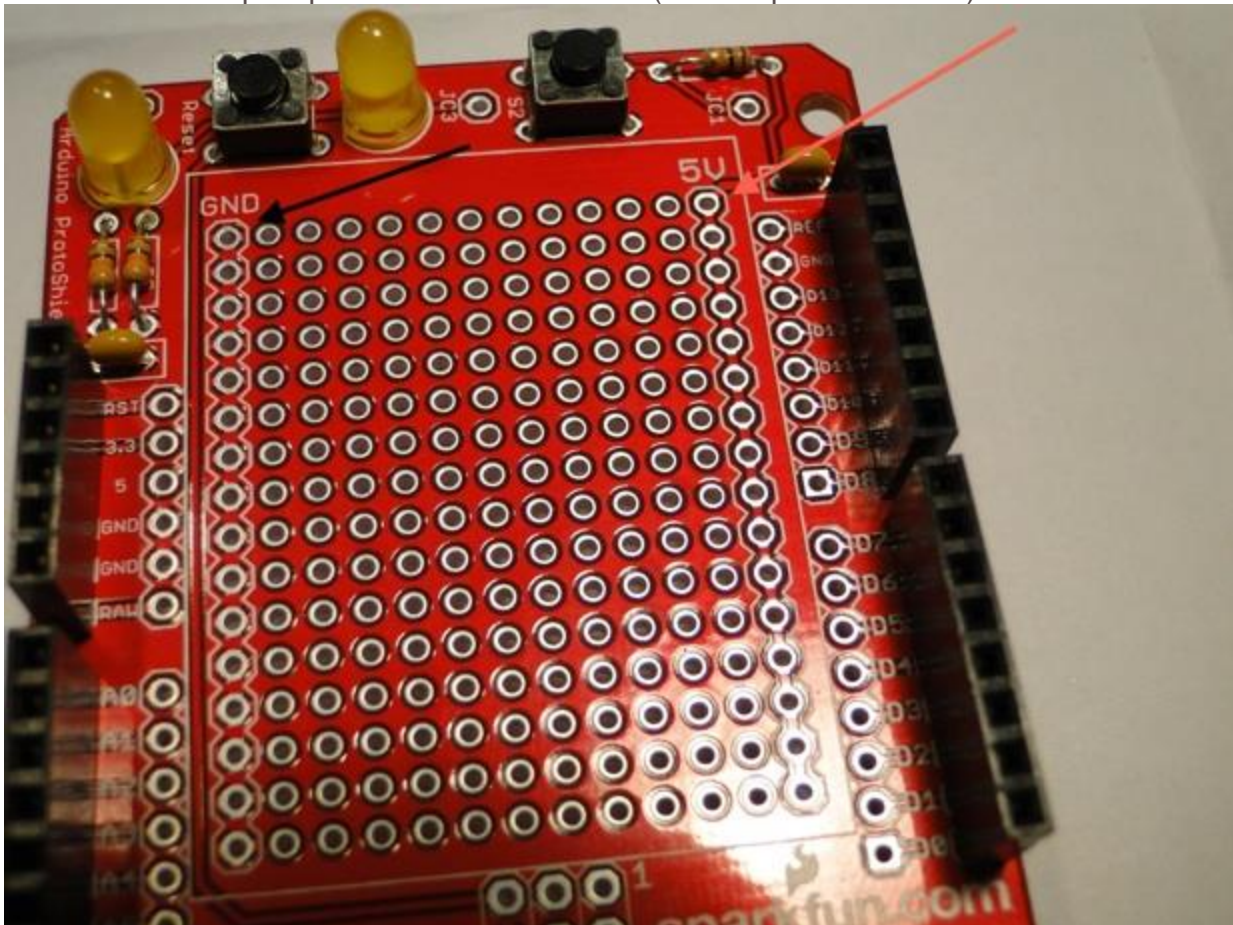
Once you've assembled it completely, its time to make our door lock circuit. First you'll need to figure out what your PIN is going to be since it will be somewhat hardwired. For the purpose of this tutorial, I'm going to use "1234" We'll need to figure out what the pins are that are operated by these numbers. So follow the chart on the keypad packaging and mark the wires on the keypad accordingly.

Whats cool about the an Arduino shield is that it stacks on top of the arduino using header pins.

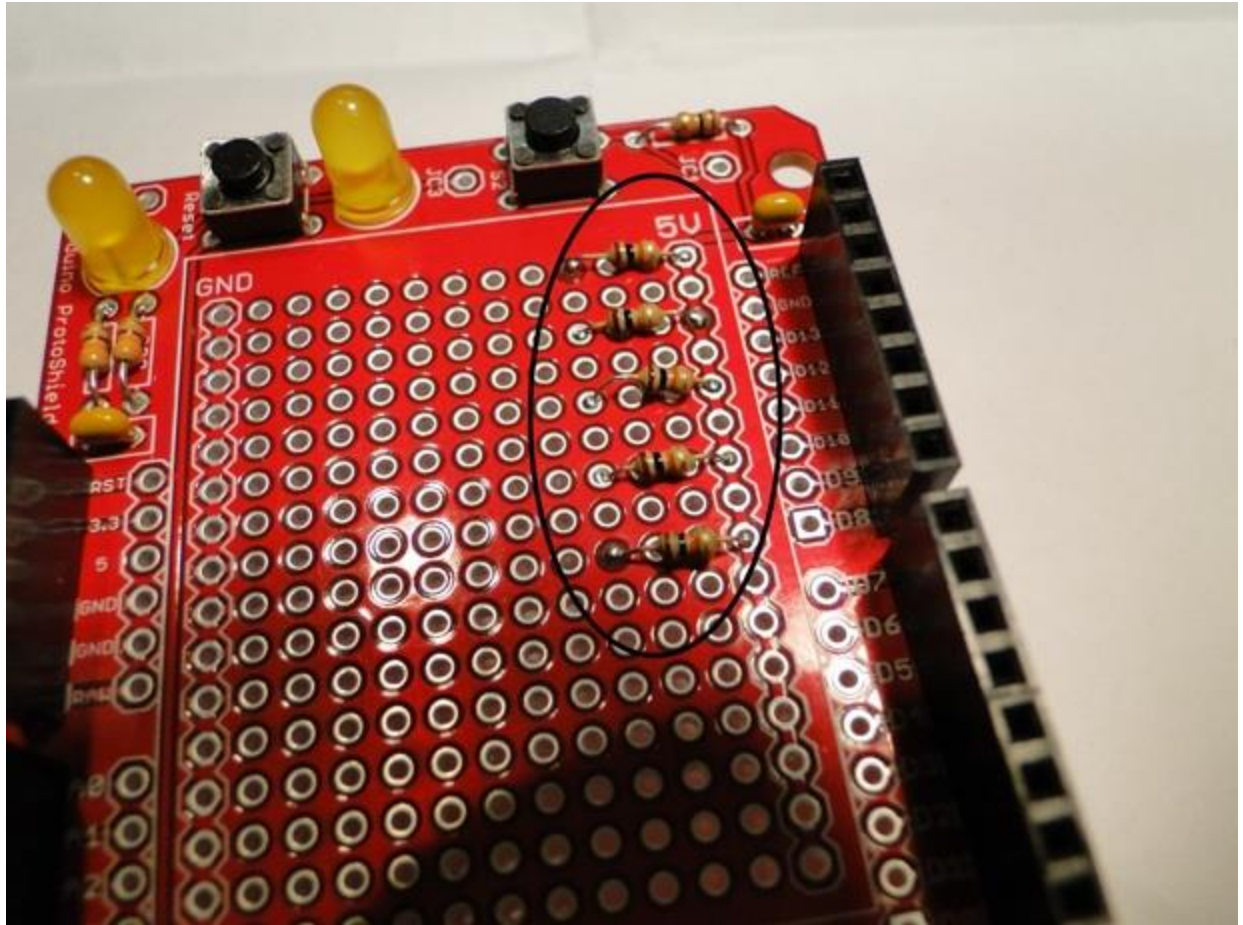


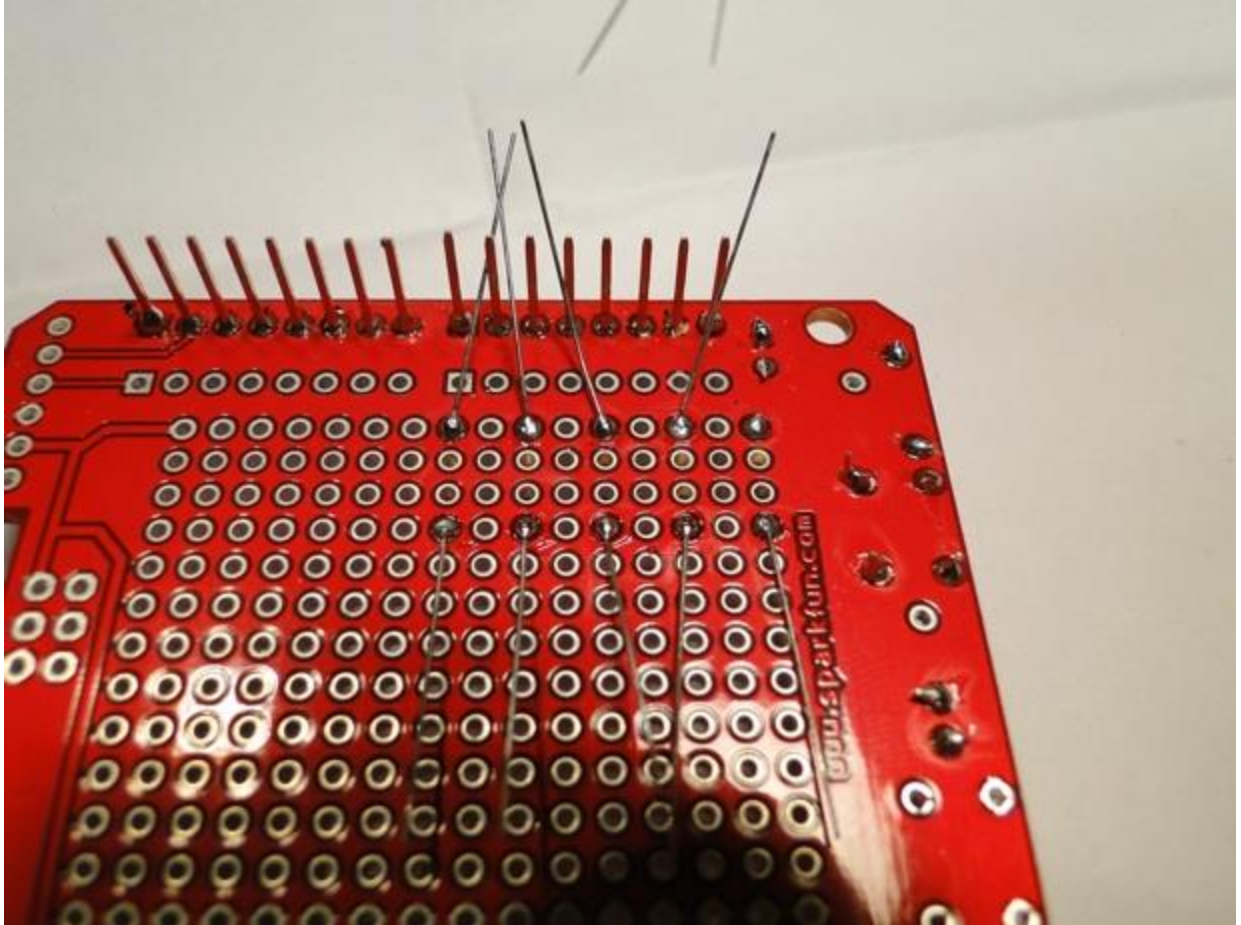


You will need to setup the circuit that accepts inputs from the keypad. Attach five 10k Ohm resistors from the +5V section on the right hand column (red arrow) to a spot on the open board. I would skip a space between each one. (see the pictures below)

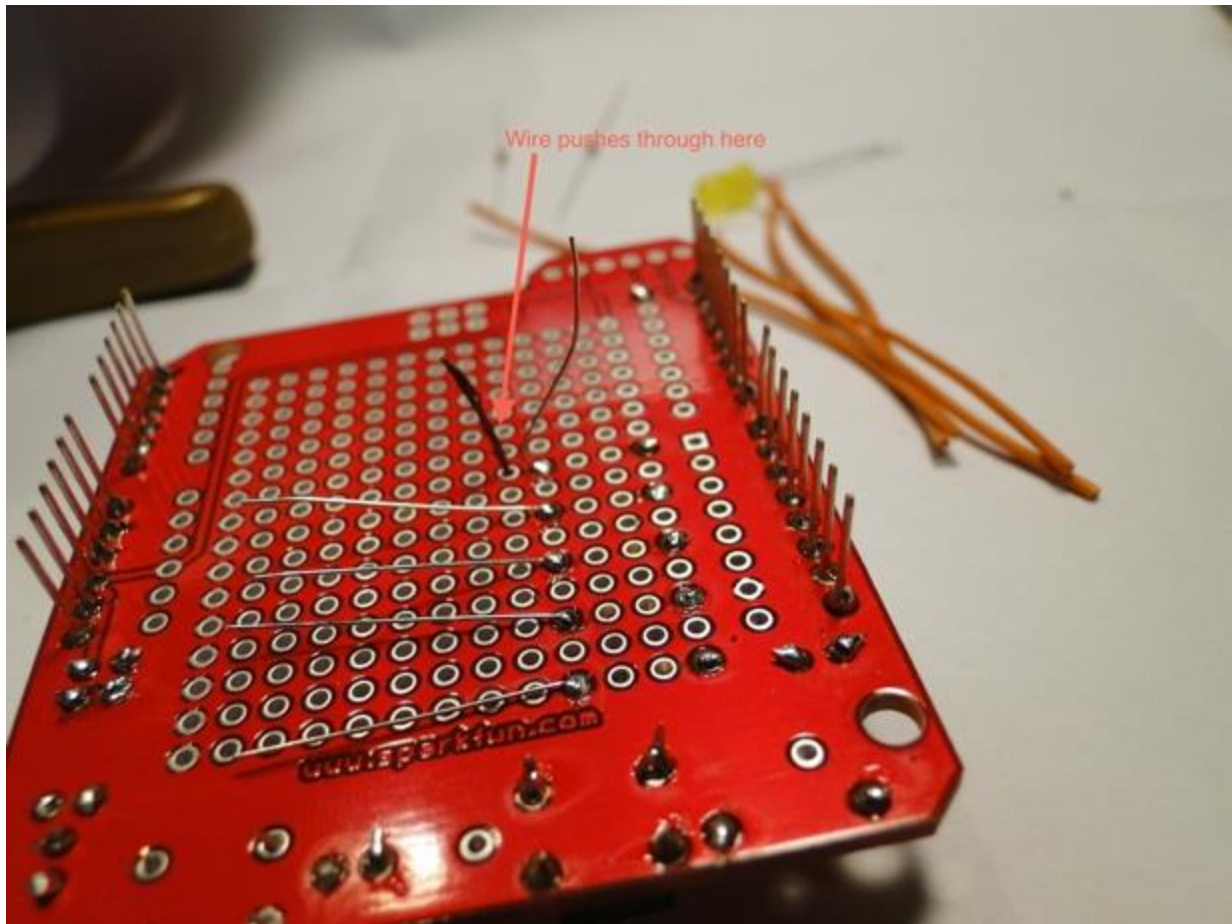


Here are the resistors in place:

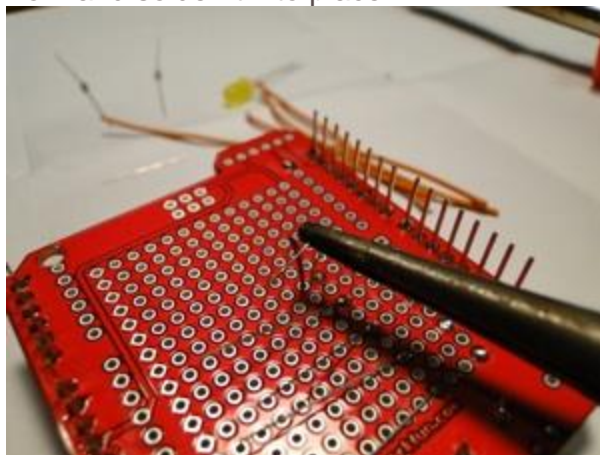


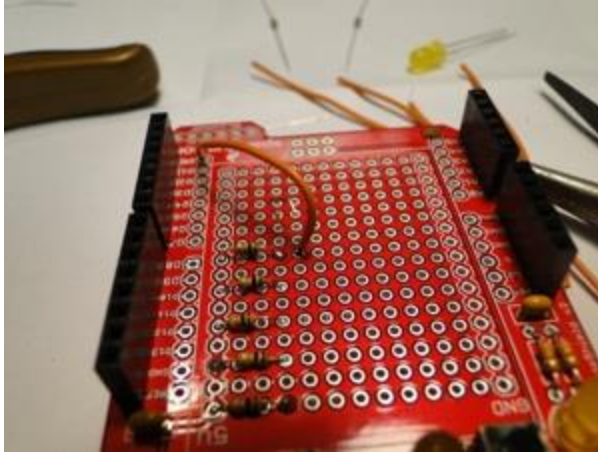


The resistor leads closest to the header pins can be removed, but leave the leads on the opposite side attached for now. Now its time to connect the four wires to the positive end of these resistors.

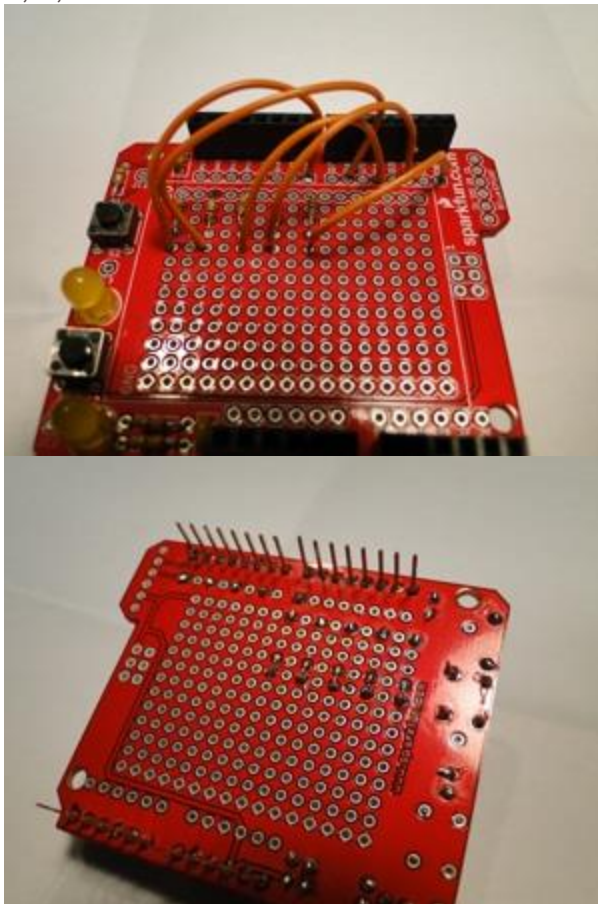


Make the remaining resistor lead push back up through the hole that the wire came down from and solder it into place.



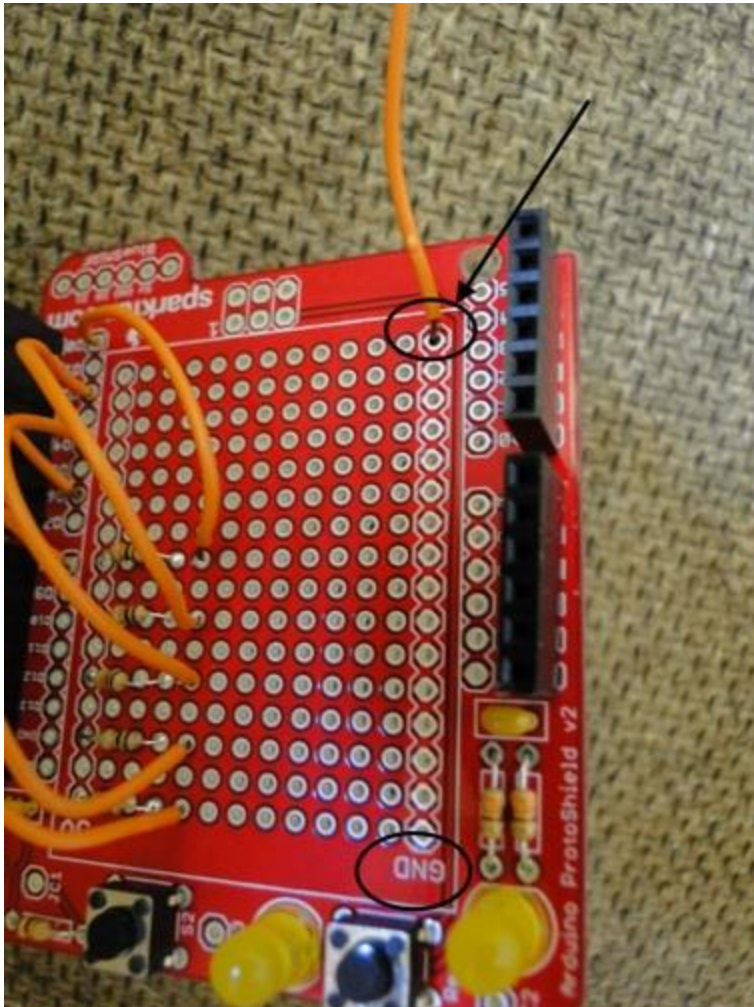


Connect the other end of this wire to Digital pin 0 on the Arduino. You will connect other wires to each resistor in the same fashion except they will connect to Arduino digital pins 2, 4, 6, and 8.

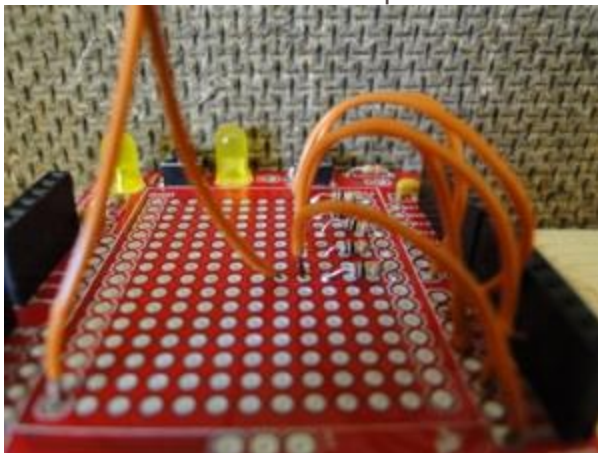


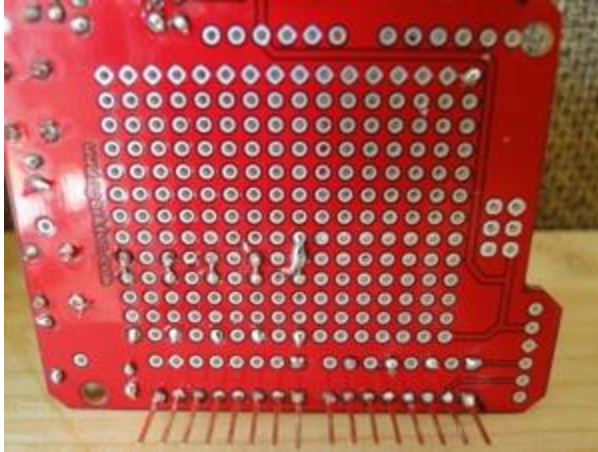
You are now ready to mount the key pad and setup the keypad wiring. Ours was mounted on the opposite side of the wall from the Arduino so I needed to drill a hole and feed the wires through. If you are also doing this, I recommend feeding the wires through one at a time and starting with the wires that are actually used in your combination.

First feed the common wire through. (the first pin on the keypad) On the other side of the wall it will attach to the GND on the Arduino.



Now start with the wire for the first button you will press, in this case "1" This will attach to the ProtoShield in the same place the wire for pin 1 does. (see below picture)



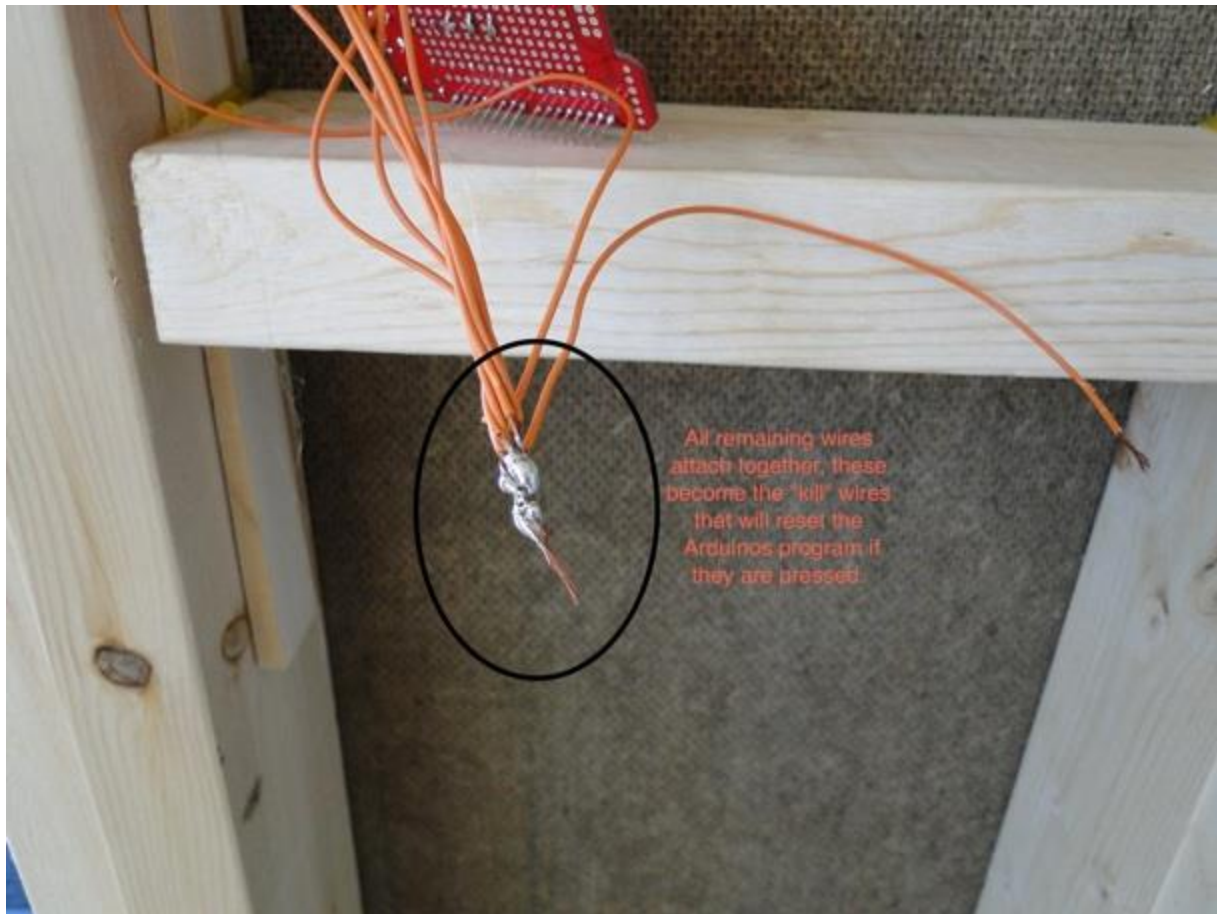


Solder it to the connection of the resistor and the Pin 0 wire.
Do this for all of the connections the same way except for number 8.

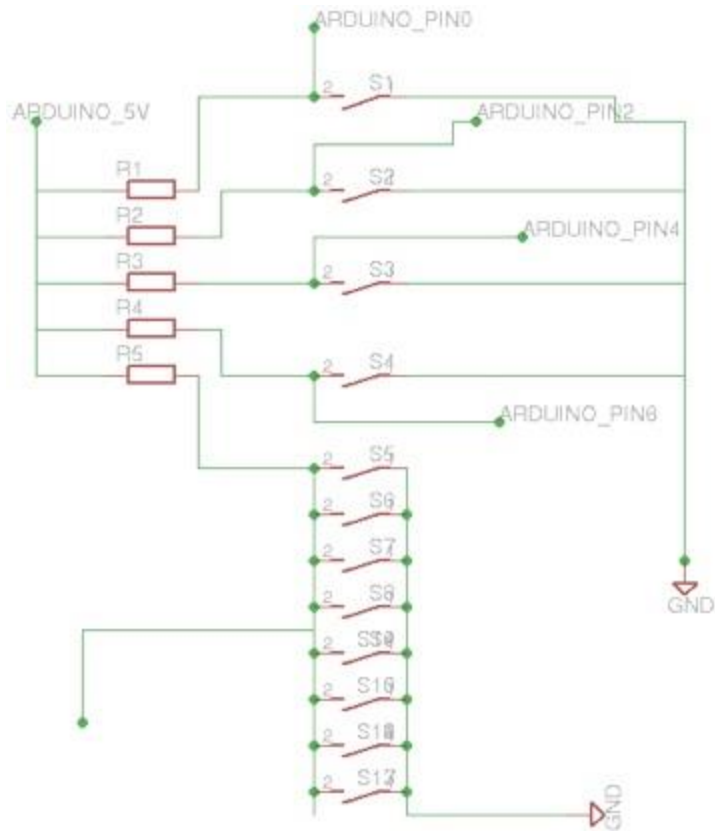
For example, my PIN number is "1234," so the wire for "1" which is pin 6 on the keypad connects to the first Arduino pin 0. The wire for "2" is pin 10 on the keypad and attaches to pin 2 on the Arduino.

Button	Keypad pin	Arduino pin
1	6	0
2	10	2
3	14	4
4	5	6

Once all the "used" connections are attached to the correct pins solder all the remaining wires from the keypad together and add an additional wire. (see below)



These wires connect to the Arduino's 8th pin. When we go over the code later on, this will make more sense. Everything should match the wiring diagram below:



CPS156

We need a device that can register the digital “high or low” output from the Arduino and convert it to an analog “on-off” signal. For this we will use the [CPS156](#). It is easy to assemble and only takes about 15-30 minutes, depending on your experience. You can download the instructions [here](#).

For the final wiring, you will need to solder a wire to pin 13 on the Arduino. The other end of this wire will attach to the L1 terminal block on the CPS156. Then run a wire from the H1 terminal block on the CPS156 to the Arduino’s ground connection.

You will need to connect the 12v power supply to the V+ and GND on the CPS156. Also run a wire from the power supply GND to the common (C) on the relay side of the CPS156.



Then connect the positive terminal of the power supply to the positive connection on your electronic door strike. Connect the negative connection on the door strike to the normally open (NO) terminal block on the CPS156. The build is now complete. Now you are ready to configure the sketch and install in on the Arduino.

The Software

Here I will explain how the software works so that you can easily change it to meet your needs.

Here is a link to download the: [Arduino Door Lock Sketch](#).

Now lets look at the code:

//You'll see that we start with some constants.

```
const int buttonPin1 = 0;
const int buttonPin2 = 2;
const int buttonPin3 = 4;
const int buttonPin4 = 6;
const int buttonPinKill = 8;
```

```
const int RelayPin = 13;
const int LEDPin = 12;
```

*/*These set up constant number values for the pins. They cannot be changed anywhere by the program.*

Then we have variables:/*

```
int buttonState1 = 0;    // variable for reading the pushbutton status
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int buttonState8 = 0;
int combChecksum = 0;
```

//They can change

/ Next is the main setup method, it prepares the Arduino for the rest of the program by telling it which pins are used for what.*

**/*

```
void setup() {
    // initialize the Relay pin and LED as outputs:
    pinMode(RelayPin, OUTPUT);
    pinMode(LEDPin, OUTPUT);
    // initialize the pushbutton pins as inputs:
    pinMode(buttonPin1, INPUT);
    pinMode(buttonPin2, INPUT);
    pinMode(buttonPin3, INPUT);
    pinMode(buttonPin4, INPUT);
    pinMode(buttonPinKill, INPUT);
}
```

//After the setup method comes the Main loop. This method gets run repeatedly thousands of times per minute. There is a lot to cover here so I will place comments throughout the code.

```
void loop(){
// read the state of the pushbuttons value:
    buttonState1 = digitalRead(buttonPin2);
    buttonState2 = digitalRead(buttonPin6);
    buttonState3 = digitalRead(buttonPin7);
    buttonState4 = digitalRead(buttonPin0);
    buttonState8 = digitalRead(buttonPinKill);
```



```
// check if the pushbuttons are pressed.  
// if it is, the buttonState is HIGH:
```

```
/*The following waits for the first button to be pressed when pressed, it changes the value of  
the "combCheckSum" variable from 0 to 1000.*/
```

```
if (buttonState1 == LOW) {  
    combCheckSum = 1000;  
}
```

```
/*then we check to see if any other buttons have been pressed, if so the checksum value is  
reset.*/
```

```
if (buttonState8 == LOW) {  
    combCheckSum = 0;  
}
```

```
/*same as above only for button 2, it also checks to make sure button 1 is not still pressed, if  
all these statements return "TRUE" it makes the checksum 1200*/
```

```
if (buttonState2 == LOW) {  
    if (buttonState1 == HIGH) {  
        if (combCheckSum == 1000) {  
            combCheckSum = 1200;  
        }  
    }  
    else {  
        combCheckSum = 0;  
    }  
}
```

```
if (buttonState8 == LOW) {  
    combCheckSum = 0;  
}
```

```
/*same as before, checking the state of button 3 this time. if these return TRUE checksum  
becomes 1230*/
```

```
if (buttonState3 == LOW) {  
    if (buttonState2 == HIGH) {  
        if (combCheckSum == 1200) {  
            combCheckSum = 1230;  
        }  
    }  
    else {  
        combCheckSum = 0;  
    }  
}
```

```
if (buttonState8 == LOW) {
```

```

    combChecksum = 0;
}
/*final check, looks at button 4, if all all the statements return TRUE the value of checksum
becomes 1234.*/
if (buttonState4 == LOW) {
    if (buttonState3 == HIGH) {
        if (combChecksum == 1230) {
            combChecksum = 1234;
        }
    }
}
else {
    combChecksum = 0;
}
}
if (buttonState8 == LOW) {
    combChecksum = 0;
}
}

/*unlock the door. if the following returns TRUE, the relay pin will become active and the
door will unlock. It will stay unlocked for 3 seconds, or 3000 milliseconds, then close
again.*/
if (combChecksum == 1234) {
    digitalWrite(RelayPin, HIGH);
    delay(3000);
    digitalWrite(RelayPin, LOW);
    combChecksum = 0;
}
}
}

```

And thats the code. Relatively simple right? Upload it to your Arduino and if you did everything right you will have a working door lock.

-Tanner Ewing
Support Manager

Leave your questions or comments in the forum box below.

