Keycode Lock

Carlos Callejas Dominguez
Robert Esposito
Ethan Nagelvoort
Ardiana Krasniqi

Summary

Keeping your precious items secure is very important, and one should have an efficient device to help protect them. A lock that utilizes a keycode can be useful in protecting your greatest treasures. For our proposal, the user will enter a keycode using a 4x4 keypad with numbers 0-9, letters A-D and symbols pound and hashtag. The green and red LEDs located on the PCB denote the status of the lock. When the device is locked, a red LED will light up and when the correct keycode is entered, a green LED will light up. The combination consists of four numbers which must be entered in the proper sequence and is pre-programmed into the device and therefore not changeable by the user. A microcontroller will be used to determine if the code entered on the keypad is correct. This device's circuit will implement an ATmega328 microcontroller, two LM339 comparators, two LEDs, and a 4x4 keypad. The ATmega328 microcontroller will be programmed to do its required tasks, as learned in COMPE 375. This mini-project name is "The Keycode Lock" and will be completed in a way for its designers to gain experience in completing projects and working as a team. This project will also meet the Senior Design A requirements as specified below.

Project Description and Block Diagram

A block diagram of the circuit is provided below. This shows the inner workings and generally how this project operates. The LEDs and the LM339's will be powered by the 5V power output of the microcontroller while the microcontroller itself will be powered by a standard 9V battery. This project will use a 4x4 matrix keypad in which the user can input their keycode. This keypad will be connected to the inputs of the microcontroller, which will run a matrix keypad algorithm to determine which key was pressed. Every key press will be saved to a variable inside the microcontroller and compared against the keycode. If more than 10 seconds have elapsed between two key presses, the variable will be reset and the user will have to enter the keycode from the beginning. The green LED will turn ON when the correct keycode is entered, otherwise the red LED will remain ON. To turn ON the correct LED, two of the microcontroller's output pins will be used to control a pair of LM339 comparators that are used to drive each LED. Because a logic 1 output signal will produce 5V and a logic 0 will produce 0V, the positive terminals of the two LM339 comparators will be connected to the 3.3V power output of the microcontroller, while the two negative terminals will each be connected to the two output pins. Until a correct keycode is entered, the first output pin will send a logic 1 signal to the first LM339, turning ON the red LED, and the second output pin will send a logic 0 to the second LM339, turning OFF the green LED. Once the correct keycode has been entered, the first output pin will switch to logic 0 and turn OFF the red LED, while the second output pin will switch to logic 1 and turn ON the green LED. After 3 seconds, the system will return to having the red LED ON and the green LED OFF. Resistors will be attached to each LED to help ensure that the LED is receiving the specified current, determined by the LEDs specs. The LM339 comparators, the LEDs, the resistors, and the connections between these

components will be fixed to a PCB we call an “Arduino Shield”. This shield is designed to fit on top of the microcontroller and connect to all of the pins on it. The traces on the PCB will connect all the components to their correct pins, as described above. This allows for the entire project to be compact and easy to transport.

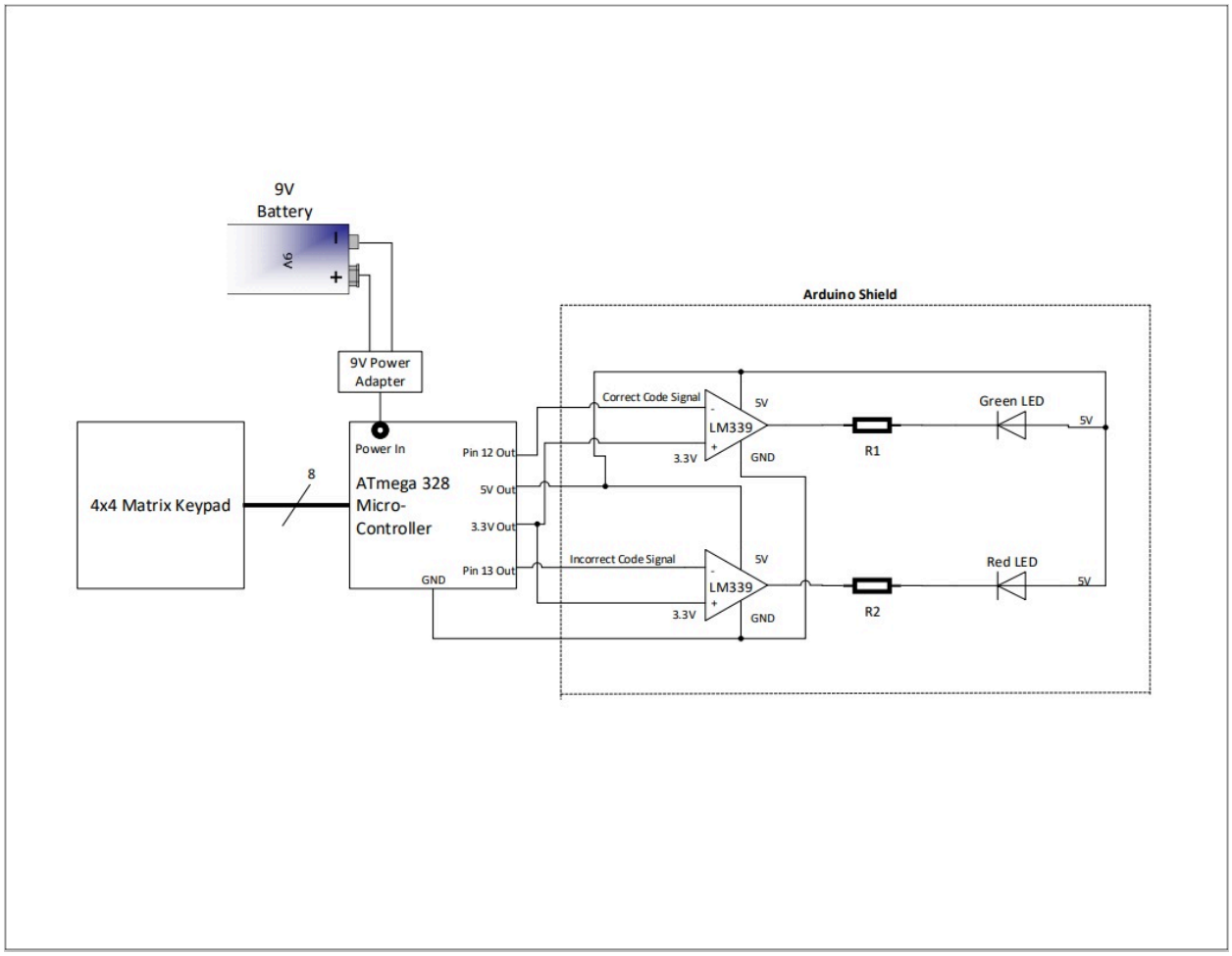


Figure 1-Block Diagram

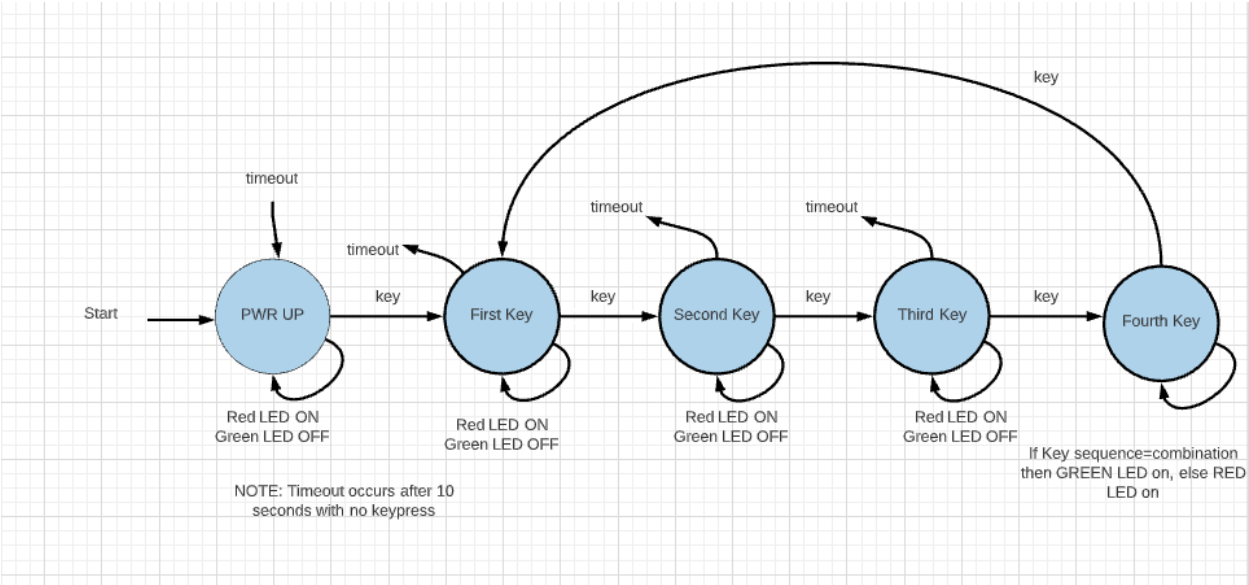


Figure 2-State Diagram

System Test:

For this test, the correct combination of the lock shall be the sequence 1234. Begin with the 9V battery removed.

1) Connect the battery with the battery clip and verify that the red LED lights.
2) Enter the following sequence with less than 10 seconds between kepresses while verifying the red remains lit and the green LED remains off: 0123,4567,89AB.
3) Enter the correct combination, 1234, and , after pressing the last keypress, verify that the red LED turns off and the green LED lights.
4) Wait 15 seconds, verify the green LED remains lit.
5) Enter the character 2 and verify the green LED turns off and the red LED lights.
6) Wait 15 seconds.Enter the correct combination and verify the green LED lights.
7) Enter the sequence 22. Wait 15 seconds. Enter the correct combination and verify the green LED lights.
8) Enter the sequence 222. Wait 15 seconds. Enter the correct combination and verify the green LED lights.

Specification and Validation

ID	Specification	Validation
1	The circuit shall operate from a standard 9-volt battery.	The circuit must pass the system test using the standard 9V battery
2	The keypad has 16 keys; numbers 0-9, letters A-D and symbols * and #.	Inspection
3	The system will have a red LED and green LED.	Inspection
4	Each LED shall have at least 8mA going through it when lit.	Verify using an ammeter on both LEDs while running the system test
5	When the keys are pressed, the system shall follow the state diagram in figure 2.	Refer to System Test

Team Skill Assessment

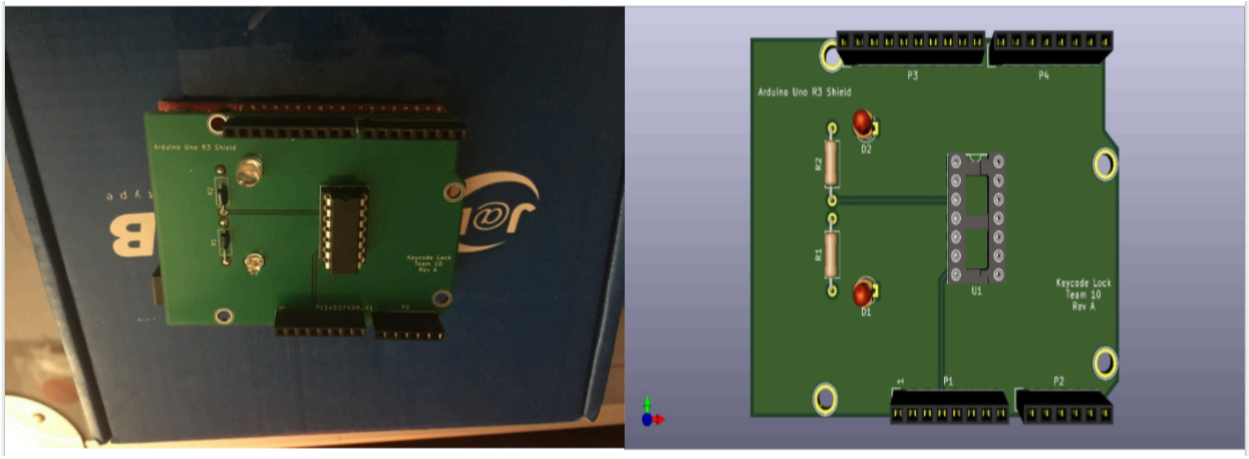
Skill or System Component	Implementation
Install components on PCB	Gained experience from EE330L and 430 lab.
LTSpice Simulation	Gained experience in EE310, EE330, and EE430.
Schematic Capture	Team members will learn KICAD.
Comparator Design	Op-amp circuits taught in EE310 and EE330.
PCB Design	Team members will learn KICAD.
PCB Fabrication	JLCPCB/SDSU will be used.
Microcontroller	Gained experience setting up microcontrollers in COMPE 375.
Thevenin Sources	Gained experience in EE310.
4x4 Keypad	Gained experience in COMPE 375.
Debug and Troubleshooting	Gained experience in EE430 lab.

Cost

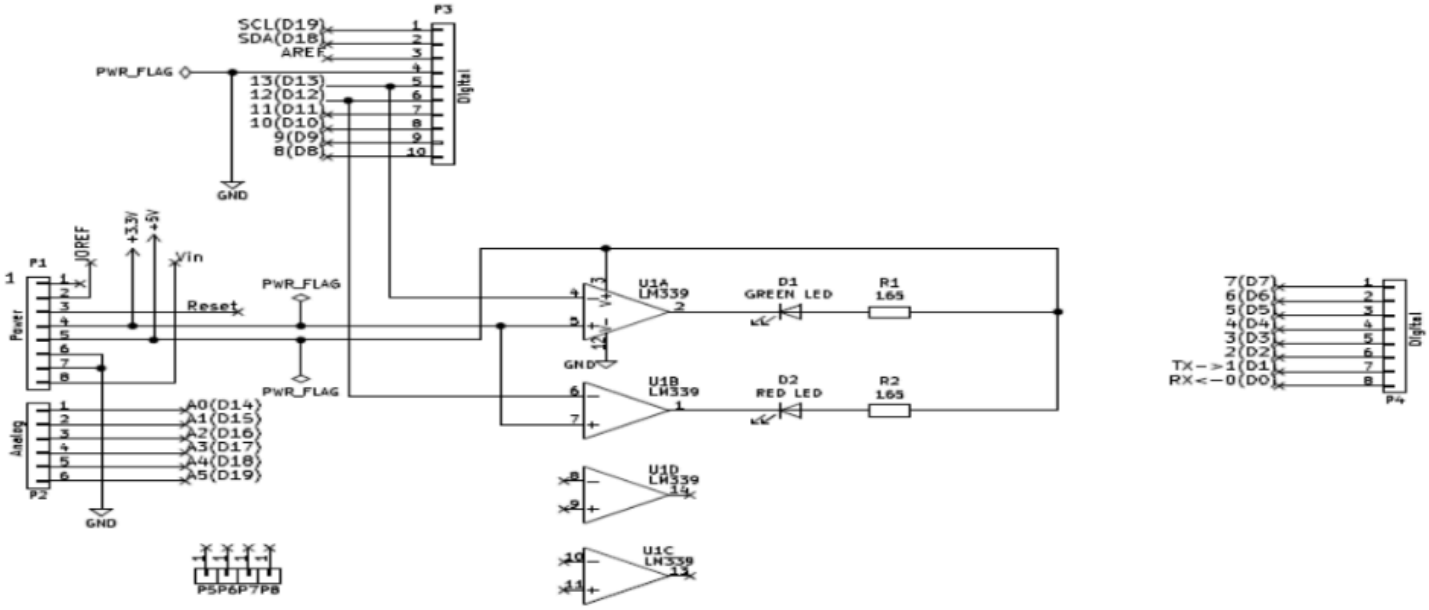
Component Description	Quantity	Cost (per unit)	Cost (total)
Atmega328 Microcontroller	1	$19.95	$19.95
LM339 Comparator	2	$0.43	$0.86
165 Ohm Resistors	2	$0.25	$0.50

Red LED	1	$0.35	$0.35
Green LED	1	$0.35	$0.35
4x4 Keypad	1	$7.95	$7.95
Thevenin Resistors	4	$0.25	$1.00
Total	**10**	**$29.53**	**$30.96**

Printed Circuit Board

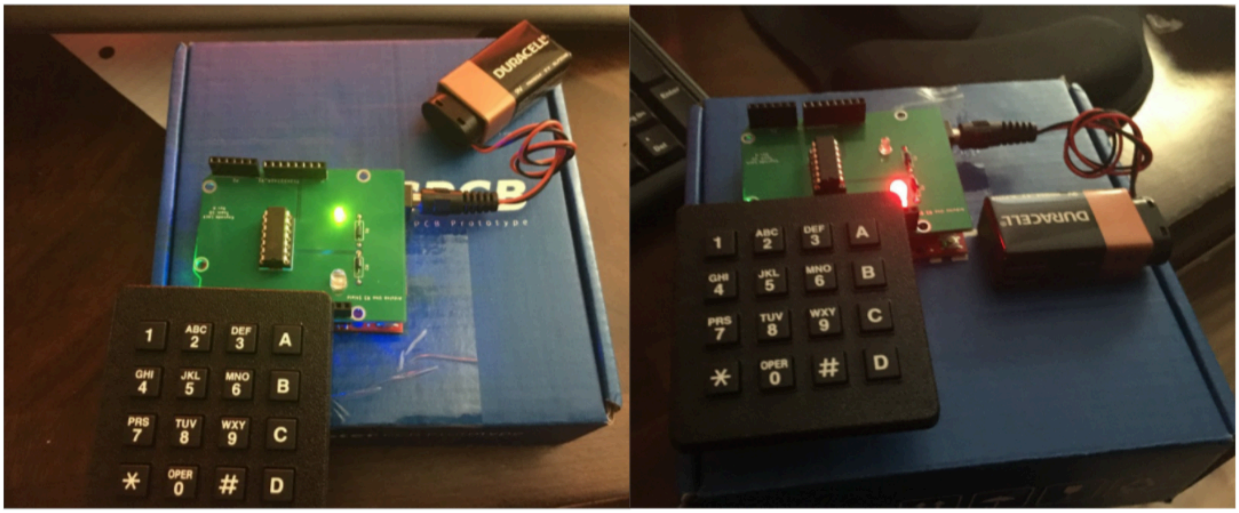


Schematic Diagram



Finished Project Working

Image with red LED lit up means incorrect code was entered and image with green LED lit up means correct code was entered.



Code within Microcontroller

```
#include <Keypad.h>

 const byte rows = 4; //four rows

 const byte cols = 4; //four columns

 char keys[rows][cols] = {

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'*','0','#','D'}

};

 byte rowPins[rows] = {0,1,2,3}; //connect to the row pinouts of the keypad

 byte colPins[cols] = {4,5,6,7}; //connect to the column pinouts of the keypad

 Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, rows, cols );

 char keycode[4] = {'1','2','3','4'};

 char entry[4] = {'0','0','0','0'};

 int count = 0;
```

```
 unsigned long timer = 20000;
 bool timeout;
 bool correct = false;
void setup() {
  pinMode(12, OUTPUT);
  pinMode(13, OUTPUT);
  for (int i = 0; i < 8; i++)
  {
    pinMode(i, INPUT);
  }
  keypad.setHoldTime(50);
  keypad.setDebounceTime(50);
}
void loop() {
  // put your main code here, to run repeatedly:
  if(!correct){
    digitalWrite(12, HIGH);
    digitalWrite(13, LOW);
  }
  timeout = true;
  for (int i = 0; i < 3; i++)
  {
    entry[i] = entry[i+1];
  }
  while (millis() < timer)
  {
    entry[3] = keypad.getKey();
    if (entry[3] != NULL)
    {
```

```
    count++;
    correct = false;
    timeout = false;
    break;
   }
  }
  if (timeout){
   count = 0;
  }
  else {
   if (count == 4){
    if ((entry[0] == keycode[0]) && (entry[1] == keycode[1]) && (entry[2] == keycode[2]) && (entry[3] ==
keycode[3])){
      digitalWrite(13, HIGH);
      digitalWrite(12, LOW);
      correct = true;
    }
    else{
      count = 0;
    }
   }
  }
  timer = millis() + 10000;
}
```