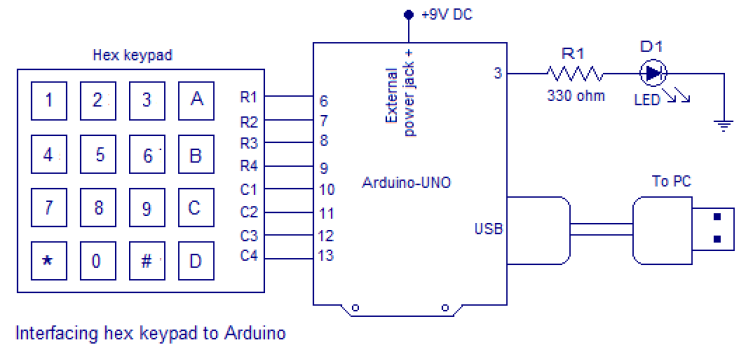
**Simple digital code lock using arduino.**

Digital code lock or digital combination lock are a type of  digital locks where a combination of digits/characters or both are used for unlocking the lock. This article is about a simple digital code lock using arduino. Here the code consists of a combination of  digits from 1 to 6. There are separate keys for locking and unlocking the system. The system can be unlocked by pressing the unlock button after entering the correct combination of digits. A hex key pad is used as the input device. Only the first two rows of key (1, 2, 3, A, 4, 5, 6, B) are used in this project. A is used for locking the system and B is used for unlocking the system. Read this article [**Interfacing hex keypad to arduino**](http://techfeasta.com/interfacing-hex-keypad-to-arduino/)for knowing more about hex keypad and its interfacing to the arduino.  The circuit diagram of the digital code lock using arduino is shown in the figure below.

**Note:-** We have developed an **Advanced Digital Code Lock using Arduino** – which is a much improved version of this project. The advanced version comes with an option to Enter User Defined Password at installation. The Lock is interfaced to LCD module to output status. The password input at installation can be changed any time later on a single key press.

**Circuit diagram.**

[](http://i0.wp.com/techfeasta.com/wp-content/uploads/2015/04/digital-codelock.png)

Row pins R1 to r4 are interfaced to digital pins 6 to 9 of the arduino. Column pins C1 to C4 are interfaced to digital pins 10 to 13 of the arduino. Digital pin 3 of the arduino is configured as the output pin for delivering the control signal for the solenoid lock. The program and circuit is designed on the assumption that the solenoid lock will “lock” for a low signal at its input and “unlock” for a high signal at its input. In this circuit the solenoid is represented by the LED D1. LED ON means “unlocked” and LED OFF means “locked”. 330 ohm resistor R1 limits the current through the LED.

**Program.**

int p[6]; //array for storing the password

int c[6]; // array for storing the input code

int n;

int a=0;

int i=0;

int lock=3;

int r1=6;

int r2=7;

int r3=8;

int r4=9;

int c1=10;

int c2=11;

int c3=12;

int c4=13;

int colm1;

int colm2;

int colm3;

int colm4;

void setup()

{

pinMode(r1,OUTPUT);

pinMode(r2,OUTPUT);

pinMode(r3,OUTPUT);

pinMode(r4,OUTPUT);

pinMode(c1,INPUT);

pinMode(c2,INPUT);

pinMode(c3,INPUT);

pinMode(c4,INPUT);

pinMode(lock,OUTPUT);

Serial.begin(9600); //sets the baud rate at 9600

digitalWrite(c1,HIGH);

digitalWrite(c2,HIGH);

digitalWrite(c3,HIGH);

digitalWrite(c4,HIGH);

digitalWrite(lock,LOW);

p[0]=1; //sets 1st digit of the password

p[1]=2; // sets 2nd digit of the password

p[2]=3; // sets 3rd digit of the password

p[3]=4; // sets 4th digit of the password

p[4]=5; // sets 5th digit of the password

p[5]=6; // sets 6th digit of the password

}

void loop()

{

digitalWrite(r1,LOW);

digitalWrite(r2,HIGH);

digitalWrite(r3,HIGH);

digitalWrite(r4,HIGH);

colm1=digitalRead(c1);

colm2=digitalRead(c2);

colm3=digitalRead(c3);

colm4=digitalRead(c4);

if(colm1==LOW)

{ n=1;

a=1;

Serial.println("1");

delay(200);}

else

{

if(colm2==LOW)

{ n=2;

a=1;

Serial.println("2");

delay(200);}

else

{

if(colm3==LOW)

{Serial.println("3");

n=3;

a=1;

delay(200);}

else

{

if(colm4==LOW)

{Serial.println("LOCKED");

digitalWrite(lock,LOW); //locks

i=0;

delay(200);}

}}}

digitalWrite(r1,HIGH);

digitalWrite(r2,LOW);

digitalWrite(r3,HIGH);

digitalWrite(r4,HIGH);

colm1=digitalRead(c1);

colm2=digitalRead(c2);

colm3=digitalRead(c3);

colm4=digitalRead(c4);

if(colm1==LOW)

{Serial.println("4");

n=4;

a=1;

delay(200);}

else

{

if(colm2==LOW)

{Serial.println("5");

n=5;

a=1;

delay(200);}

else

{

if(colm3==LOW)

{Serial.println("6");

n=6;

a=1;

delay(200);}

else

{

if(colm4==LOW)

{

if(c[0]==p[0]&&c[1]==p[1]&&c[2]==p[2]&&c[3]==p[3]&&c[4]==p[4]&&c[5]==p[5])

{digitalWrite(lock,HIGH); //unlocks

 Serial.println("UNLOCKED");

c[5]=9;} //corrupts the code in array c

else

{Serial.println("WRONG PASSWORD");}

delay(200);}

}}}

if(a==1) // test whether a digit key is pressed

{

c[i]=n; // saves the current digit pressed to array c

i=i+1;

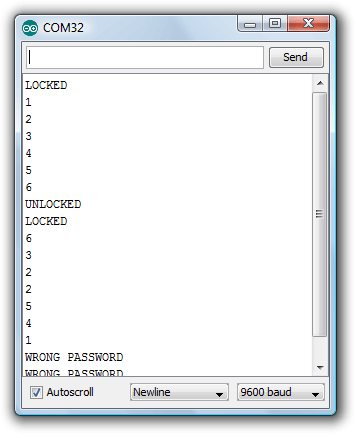
a=0;}

}

**About the program.**

The password which is “123456” is stored in the array “p”. When ever the digit keys are pressed, they are stored in the array “c”. When ever the unlock button is pressed, the contents in the both array are compared and if they are same then digital pin 3 is made high. After this the content of array “c” corrupted by the program. This is done to prevent the correct code from remaining in the memory. If it is not done the system will unlock just on the press of the unlock button(B) after another lock cycle. Pressing the lock button(A) will make the digital pin low. The lock button has to be pressed before you enter the password each time.

The system can be connected to the PC through the USB and the pressed keys can be viewed through the serial monitor window of the arduino. The screen shot of the serial monitor window of this project is shown in the figure below.

[](http://i1.wp.com/www.circuitstoday.com/wp-content/uploads/2014/06/digital-combination-lock.png)When lock button (key A in the hex keypad) is pressed the serial monitor window will display “LOCKED”. The code entered will be also displayed on the window. When unlock button (key b in the hex keypad) is pressed the serial monitor window will display “UNLOCKED”. If the code entered is wrong the serial monitor window will display”WRONG PASSWORD”.

**Notes.**

* For the present configuration ie; no solenoid, there is no need for the 9V external supply. The board can be powered by the PC through the USB.
* The solenoid will consume a good amount of current and the PC’s USB port may be unable supply it. So when you are using a solenoid a separate external supply for powering it is required. The arduino board can be also powered from this external supply if it is 9V.
* The number of digits in the password can be increased by modifying the program.
* I have not shown the solenoid because I do not have one right now. I will add the updated circuit diagram and program as soon as I get one.

**Discuss this topic**