CATEGORY: LIFE HACKS

RESULT: WINNER

AUTHOR: JAVIER BRIZ & VALENTIN PLUGARU

ARDUINO SECURITY

Project goal:

Design and implement a door lock mechanism controlled by a numerical keypad connected to an Arduino board. We thought of several ways to do it and finally decided to use a sliding lock moved by a servo motor.

Hardware used

- One Arduino NG on a prototyping board
- Servo motor
- Wire (hard but still maleable)
- Numeric keypad with USB-to-PS/2 converter
- PC speaker
- 9V,600mA transformer to power the Arduino and motor
- Sliding lock
- 18kOhm resistor
- Lots of wires

Future improvements

Beyond (some of) the possible improvements stated in our video, we are also thinking about adding another door lock (and servo) to the same Arduino, which would enable us to open two doors depending on the password entered.

Software used

```
#define DEBUG 0
#define VERSION "v0.7 17Apr2008"
//SCANCODES PS/2 KEYBOARDS
#define SCAN_ENTER 0xE0
#define SCAN_BREAK 0xf0

byte scanCodes[10] = {0x70,0x69,0x72,0x7A,0x6B,0x73,0x74,0x6C,
```

```
0x75,0x7D};
char characters[10] = {'0','1','2','3','4','5','6','7','8','9'};
int quantityCodes = 10;

//KEYBOARD HANDLING VARS
byte dataValue;

// PINS
int clockPin = 3; //KEYBOARD CLK
int dataPin = 2;  //KEYBOARD DAT
int soundPin = 9; // TEST
int servoPin = 10;
int closePin = 7;// DOOR "Switch"
int speakerPin = 12; //Optional Speaker VCC

// PASSWORD
char mypass[11]="12345\0";
char pass[11];
int passlen=0;
char letra;

void setup()
{
  pinMode(dataPin, INPUT);
  pinMode(clockPin, INPUT);
  pinMode(closePin, INPUT);
  pinMode(speakerPin, OUTPUT);
  if (DEBUG)
  {
    Serial.begin(9600);
    analogWrite(soundPin,100);
    Serial.println("ARDUINO DOOR OPENER ONLINE");
    Serial.println(VERSION);
  }
 startupSound();
}

void loop()
{
  dataValue = dataRead();
  if (dataValue == SCAN_BREAK)
  {
    dataValue = dataRead(); // Reading the key that was depressed
    dataValue = dataRead(); // Wait for and read a new key
  }
```

```
  pinMode(clockPin,OUTPUT);
  digitalWrite(clockPin,LOW);

  for (int i = 0; i < quantityCodes; i++)
  {
    byte temp = scanCodes[i];
    if (temp == dataValue || dataValue == SCAN_ENTER)
    {
      letra=characters[i];
      if (DEBUG) Serial.println(letra);
      // If enter read
      if(dataValue == SCAN_ENTER)
      {
        digitalWrite(clockPin,HIGH);
        pinMode(clockPin,INPUT);
        dataValue = dataRead();
        pinMode(clockPin,OUTPUT);
        digitalWrite(clockPin,LOW);
        if(dataValue == 0x5A)
        {
          pass[passlen+1]='\0';
          if (strcmp(pass,mypass)==0) {Serial.println("Abriendo");OpenDoor();}
          else {errorSound(); for (int i=0; i<=passlen; i++)
{Serial.print(pass[i]);} }
          Serial.println();
          for (int i=0; i<=10; i++) pass[i]='\0';
          passlen=0;
        }
      }
      //if not enter read
      else
      {
        pass[passlen]=letra;
        if (DEBUG)
        {
          Serial.print(letra); Serial.print(" ");
          Serial.print(pass[passlen]);Serial.print("  ");
          Serial.print(passlen);
          Serial.print("  PASS="); for (int i=0; i<=passlen; i++)
Serial.print(pass[i]); Serial.println();
        }
        passlen++;
        if (passlen==10) passlen=0;
      }
      dataValue = 0;
```

```
      letra=0;
      //delay(200);
    }
    }
  digitalWrite(clockPin,HIGH);
  pinMode(clockPin,INPUT);
}

void OpenDoor()
{
    // OPEN

    analogWrite(servoPin,40);
    openSound();
    delay(3000);

    analogWrite(servoPin,0);
    while (!digitalRead(closePin)) delay(250);

    //CLOSE
    analogWrite(servoPin,220);
    closeSound();
    delay(2000);
    analogWrite(servoPin,0);
}

int dataRead() {
  byte val = 0;
  int v = 0;
  while (digitalRead(clockPin)); // Waiting for LOW
  while (!digitalRead(clockPin));// Waiting for HIGH
  while (digitalRead(clockPin)); // Waiting for LOW
  for (int offset = 0; offset < 8; offset++)
  {
    while (digitalRead(clockPin));        // Waiting for LOW
    val |= digitalRead(dataPin) << offset; // Adding to byte
    while (!digitalRead(clockPin));       // Waiting for HIGH
  }
  while (digitalRead(clockPin));          // Waiting for LOW
  while (!digitalRead(clockPin));         // Waiting for HIGH
  while (digitalRead(clockPin));          // Waiting for LOW
  while (!digitalRead(clockPin));         // Waiting for HIGH
  return val;
}
```

```
void errorSound()
{
  int n = 2000;
  for (int vez=0;vez<2;vez++){
    for (int _=0;_<30;_++){
      delayMicroseconds(n);
      digitalWrite(speakerPin, HIGH);
      delayMicroseconds(n);
      digitalWrite(speakerPin, LOW);
    }
  delay(50);
  }
}

void openSound()
{
  int n = 300;
  for (int vez=0;vez<3;vez++){
    for (int _=0;_<100;_++){
      delayMicroseconds(n);
      digitalWrite(speakerPin, HIGH);
      delayMicroseconds(n);
      digitalWrite(speakerPin, LOW);
    }
  delay(100);
  }
}

void closeSound()
{
  int n = 300;
  for (int vez=0;vez<2;vez++){
    for (int _=0;_<100;_++){
      delayMicroseconds(n);
      digitalWrite(speakerPin, HIGH);
      delayMicroseconds(n);
      digitalWrite(speakerPin, LOW);
    }
  delay(100);
  }

  n = 450;
  for (int _=0;_<110;_++){
    delayMicroseconds(n);
    digitalWrite(speakerPin, HIGH);
```

```
    delayMicroseconds(n);
    digitalWrite(speakerPin, LOW);
  }
}

void startupSound()
{
//RE MI DO DO. SOL
  int note = (1/587.33)*500000;
  for (int _=0;_<100000/note;_++){
    delayMicroseconds(note);
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(note);
    digitalWrite(speakerPin, LOW);
  }
  note = (1/659.26)*500000;
  for (int _=0;_<100000/note;_++){
    delayMicroseconds(note);
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(note);
    digitalWrite(speakerPin, LOW);
  }
  note = (1/523.25)*500000;
  for (int _=0;_<100000/note;_++){
    delayMicroseconds(note);
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(note);
    digitalWrite(speakerPin, LOW);
  }
  note = (1/261.63)*500000;
  for (int _=0;_<100000/note;_++){
    delayMicroseconds(note);
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(note);
    digitalWrite(speakerPin, LOW);
  }
  note = (1/392.00)*500000;
  for (int _=0;_<200000/note;_++){
    delayMicroseconds(note);
    digitalWrite(speakerPin, HIGH);
    delayMicroseconds(note);
    digitalWrite(speakerPin, LOW);
  }
}
```