



ELECTRÓNICA DIGITAL II

*Tema: Sistema de seguridad para una
bodega de un almacén fiscal de Aduana.*

Informe

Ing. Luis Isaac Herrera

- *Steven Yosette Mendoza Zapata H.*
- *Jeremy Benjamín Chávez Hernández.*
- *Dayanna Esther Martinez Mejia*
- *José Santiago Chow Rodríguez*

20 de Abril de 2024

Introducción

El sistema de seguridad propuesto para la bodega de un almacén fiscal de Aduana integra tecnología avanzada para garantizar un control efectivo de acceso y protección. Basado en la interacción entre una computadora ejecutando una simulación y un Arduino conectado, el sistema emplea diversos componentes para su funcionamiento. Desde la simulación, controlada por un Arduino Nano, hasta la gestión física a cargo del Arduino real, el sistema se adapta a condiciones específicas para activar la apertura de la puerta, controlar los sensores de movimiento internos y gestionar la alarma ante intrusiones. Este informe detallará la arquitectura, funcionamiento y características del sistema de seguridad, destacando su eficacia y versatilidad en la protección de la bodega del almacén fiscal de Aduana.

Objetivo general:

- Garantizar la protección integral de la bodega de un almacén fiscal de Aduana, mediante la implementación de una solución tecnológica avanzada que controle el acceso autorizado, detecte intrusiones y responda de manera efectiva ante situaciones de riesgo, proporcionando así un ambiente seguro y protegido para los bienes almacenados.

Objetivos específicos

- Configurar el Arduino Nano para recibir las entradas del teclado telefónico y el COMPIM, y enviar las señales correspondientes al Arduino real para el control de acceso.
- Programar el Arduino real para gestionar los display de 7 segmentos, mostrando "CLOSED" en espera y "OPEN" durante el acceso autorizado, con un desplazamiento de letras cada dos segundos.
- Desarrollar un algoritmo para el Arduino real que active y desactive los sensores de movimiento internos según la presencia de personal en la bodega, utilizando los pines de entrada y salida digital.
- Integrar un sistema de alarma sonora en el Arduino real que se active en caso de detección de intrusos por los sensores de movimiento, y pueda ser desactivada únicamente con la llave maestra.

Marco teórico

- **Control de acceso:** Este concepto se refiere a los métodos y dispositivos utilizados para regular quién puede acceder a un determinado espacio. En el caso del sistema propuesto, se emplea un teclado telefónico como método de autenticación de usuarios mediante la introducción de contraseñas.
- **Arduino:** Es una plataforma de hardware de código abierto que utiliza microcontroladores para la creación de dispositivos electrónicos interactivos. En este sistema, se utilizan dos tipos de Arduinos: un Arduino Nano para la simulación y un Arduino real para el control físico de los componentes.
- **Sensores de movimiento:** Son dispositivos que detectan la presencia de personas u objetos en un área determinada. En este caso, se utilizan sensores de movimiento internos en la bodega para detectar intrusiones y activar la alarma correspondiente.
- **Alarma sonora:** Es un dispositivo de seguridad que emite un sonido de advertencia en caso de detectarse una situación de riesgo. En el sistema propuesto, la alarma sonora se activa en respuesta a la detección de intrusos por los sensores de movimiento.
- **Comunicación entre dispositivos:** Se refiere a los métodos y protocolos utilizados para que los dispositivos electrónicos intercambien información entre sí. En este caso, se establece una comunicación entre el Arduino real y la simulación en Proteus para enviar datos sobre la activación de la alarma y el estado de los sensores.
- **Visualización de datos:** Es el proceso de presentar información de manera visual para facilitar su comprensión. En el sistema propuesto, se utiliza una pantalla LCD para mostrar mensajes de estado y alertas de seguridad.
- **Programación:** Se refiere al proceso de escribir instrucciones específicas para que los dispositivos electrónicos realicen determinadas tareas. En este proyecto, se requiere programar los Arduinos para gestionar el control de acceso, la activación de los sensores y la comunicación entre dispositivos.

Desarrollo

Debido a la complejidad del proyecto se fue desarrollando poco a poco, verificando que en cada proceso agregado en el código y la simulación, concordara con el circuito en físico y tuviera una correcta comunicación.

Primeramente, se desarrolló un sistema con estructuras para poder controlar un display de 7 segmentos, luego se definieron datos que contenían que leds tenían que encenderse del display para mostrar en él, cada letra dentro de la palabra “closed” y “open”, luego se definió 2 funciones para mostrar la palabra closed” y “open” desplazando cada letra de la palabra en el display, luego se conectaron 2 display en los mismos pines y se asignaron 2 para controlar su tierra, y poder seleccionar cual energizar, luego se definió una función para alternar entre los valores que se quieran mostrar en cada display y que conmuta los display a una velocidad suficiente para que a la vista parecieran que están energizados en simultaneo, después se ajustaron las funciones para que se desplazaran las palabras en los 2 displays.

Luego se creo un teclado matricial, y se programo un programa en el arduino digital para que dependiendo que columna y fila se energizara detectara que pin se activó, se prosiguió con añadir una funcionalidad a cada tecla cuando la detecte, se asignaron los pines de apertura y cerrado, se asignó la funcionalidad de poder ir guardando los valores en una matriz unidimensional de tamaño 6, de borrar el ultimo numero ingresado y poder enviar el pin para ser verificado. Al ser verificado y ser distinto al pin de apertura al estar cerrado o el pin de cerrado al estar abierto, se toma como intento fallido, llevándose un conteo de esto, al llegar a 3 fallos, se bloquea y se activa una bocina que puede ser desactivada por una llave maestra controlado con otro pin, reiniciando los intentos fallidos a 0. Y al ser aceptada la clave se cierra la cerradura, al estar abierta o se abre al estar cerrada, reiniciando igualmente los intentos fallidos que tenga, mientras se tiene registro si la cerradura esta abierta o no con un pin que energiza un led.

Luego se implemento los sensores de movimiento, siendo que estos al ser activados pasan a activar la alarma, y solo pudiendo ser desactivada con la llave maestra y siguiendo el resto de restricciones propuestas.

Luego se termino de desarrollar el código, la simulación y el circuito físicos, los cuales se comunicaban con comunicación serial mediante la salida USB de la laptop, para compartir datos, y se concluyo que se cumplió con todos los requerimientos

Conclusión

El desarrollo e implementación del sistema de seguridad basado en Arduinos para la protección de la bodega en el almacén fiscal de Aduana representa un paso significativo hacia la mejora de los estándares de seguridad en entornos comerciales y logísticos. A través de la integración de tecnología accesible y funcional, se ha logrado establecer un mecanismo eficaz para controlar el acceso a la bodega, detectar intrusiones y activar alarmas en caso de necesidad.

La combinación de componentes como el teclado telefónico, los displays de 7 segmentos y los sensores de movimiento ha permitido crear un sistema versátil y adaptable a las necesidades específicas del entorno de almacenamiento. La simulación previa ha demostrado la robustez y fiabilidad del sistema, garantizando su efectividad en situaciones reales.

Mediante la entrega del código necesario para la implementación del sistema, se proporciona una herramienta práctica y funcional que puede ser fácilmente adaptada y personalizada según los requerimientos individuales de cada instalación. Esto abre la puerta a futuras mejoras y optimizaciones que puedan surgir a medida que evolucionen las necesidades de seguridad en el almacén.

En resumen, el sistema de seguridad desarrollado representa un avance significativo en la protección de activos y la salvaguarda del personal en entornos comerciales y logísticos, reafirmando el compromiso de la organización con la seguridad y la eficiencia operativa.

Anexos

Código del Arduino Real:

```
int pinA = 2;
int pinB = 3;
int pinC = 4;
int pinD = 5;

int pinE = 6;
int pinF = 7;
int pinG = 8;

int display1_earth = 9;
int display2_earth = 10;

int magneticLocker = 11;

int refreshFrame_Time = 10;
int refreshImage_Time = 2000;

int unlock_Time = 3000;
int open_Time = 6000;

bool open = false;

struct display{
    bool a;
    bool b;
    bool c;
    bool d;

    bool e;
    bool f;
    bool g;
};

struct display dC = {true,false,false,true, true,true,false};
struct display dL = {false,false,false,true, true,true,false};
struct display dE = {true,false,false,true, true,true,true};
struct display dP = {true,true,false,false, true,true,true};
struct display dO = {true, true, true, true, true, true, false};
struct display dD = {false, true, true, true, true, false, true};
struct display dS = {true, false, true, true, false, true, true};
struct display dN = {false, false, true, false, true, false, true};

struct display emptyness = {false,false,false,false, false,false,false};

void displayData(struct display data){
    digitalWrite(pinA, data.a ? HIGH : LOW);
    digitalWrite(pinB, data.b ? HIGH : LOW);
    digitalWrite(pinC, data.c ? HIGH : LOW);
    digitalWrite(pinD, data.d ? HIGH : LOW);

    digitalWrite(pinE, data.e ? HIGH : LOW);
    digitalWrite(pinF, data.f ? HIGH : LOW);
    digitalWrite(pinG, data.g ? HIGH : LOW);
}
```

```

void displayFrames(int ms, struct display d1, struct display d2){
    int frames = (ms / refreshFrame_Time);

    for(int i=0; i<(frames/2) ; i++){
        digitalWrite(display1_earth, LOW);
        digitalWrite(display2_earth, HIGH);
        displayData(d1);

        readSensors();
        delay(refreshFrame_Time);
        readSensors();

        digitalWrite(display1_earth, HIGH);
        digitalWrite(display2_earth, LOW);
        displayData(d2);

        readSensors();
        delay(refreshFrame_Time);
        readSensors();
    }
}

void displayOpen(){
    int ms = 0;
    digitalWrite(magneticLocker, HIGH);

    while( ms < open_Time ){
        displayFrames(refreshImage_Time, dO, dP);
        ms+= refreshImage_Time; if(ms >= open_Time){ break; } else if(ms >= unlock_Time){ digitalWrite(magneticLocker, LOW); };

        displayFrames(refreshImage_Time, dP, dE);
        ms+= refreshImage_Time; if(ms >= open_Time){ break; } else if(ms >= unlock_Time){ digitalWrite(magneticLocker, LOW); };

        displayFrames(refreshImage_Time, dE, dN);
        ms+= refreshImage_Time; if(ms >= open_Time){ break; } else if(ms >= unlock_Time){ digitalWrite(magneticLocker, LOW); };

        displayFrames(refreshImage_Time, dN, dO);
        ms+= refreshImage_Time; if(ms >= unlock_Time){ digitalWrite(magneticLocker, LOW); };
        open = false;
    }
}

void displayClosed(){

    displayFrames(refreshImage_Time, dC, dL); if(open){return;}
    displayFrames(refreshImage_Time, dL, dO); if(open){return;}
    displayFrames(refreshImage_Time, dO, dS); if(open){return;}
    displayFrames(refreshImage_Time, dS, dE); if(open){return;}
    displayFrames(refreshImage_Time, dE, dD); if(open){return;}
    displayFrames(refreshImage_Time, dD, dC); if(open){return;}
}

void setup() {
    Serial.begin(9600);
    for (int i = 2; i <= 13; i++) {
        pinMode(i, OUTPUT);
    }
    pinMode(A0, INPUT);
    pinMode(A1, INPUT);
    pinMode(A2, INPUT);

    displayData(emptyness);
}

```



```
}

void readSensors(){
  if(digitalRead(A0) || digitalRead(A1) || digitalRead(A2)){ Serial.print("0"); }

  digitalWrite(11, digitalRead(A0) ? HIGH : LOW);
  digitalWrite(12, digitalRead(A1) ? HIGH : LOW);
  digitalWrite(13, digitalRead(A2) ? HIGH : LOW);
}

void loop() {
  if (Serial.available() > 0) {
    char data = Serial.read();

    if (data == '1') {
      open = true;
    }
    else if (data == '0') {
      open = false;
    }
  }

  if (open){ displayOpen(); }
  else{ displayClosed(); }
}
```

Código del Arduino de Proteus:

```
int openningPassword[6] = {1,1,1,1,1,1};
int closingPassword[6] = {2,2,2,2,2,2};
int actualNum[6] = {0, 0, 0, 0, 0, 0};
int keyCounter = 0;

bool passwordSelector = true;
int failCounter = 0;

bool sensorSwitch = true;

bool isEqual(const int* a, const int* b){
    for(int i = 0; i<6; i++){
        if(a[i] != b[i]){return false;}
    }
    return true;
}

void checkPassword(){

    if(passwordSelector){
        if( isEqual(openningPassword, actualNum) ){
            //Serial.print(" Entry Accepted. ");
            Serial.print("1");
            passwordSelector = ! passwordSelector;
            digitalWrite(9,LOW);
            failCounter = 0;
            sensorSwitch = false;
        }
        else{
            failCounter += 1;
            //Serial.print(" Entry Denied ");
            //Serial.print(" Fail Counter: ");
            //Serial.print(failCounter);
            //Serial.print(" ");
        }
    }
    else if( isEqual(closingPassword, actualNum) ){
        //Serial.print(" Exit Accepted ");
        passwordSelector = ! passwordSelector;
        digitalWrite(9,HIGH);
        failCounter = 0;
        sensorSwitch = true;
        Serial.print("0");
    }
    else{
        failCounter += 1;
        //Serial.print(" Exit Denied ");
        //Serial.print(" Fail Counter: ");
        //Serial.print(failCounter);
        //Serial.print(" ");
    }
}

actualNum[0] = 0; actualNum[1] = 0; actualNum[2] = 0; actualNum[3] = 0; actualNum[4] = 0; actualNum[5] = 0;
keyCounter = 0;
}

void getKey(){

    if(keyCounter == 6){
        if(digitalRead(5) && digitalRead(8)){
            checkPassword();
        }
    }
}
```

```

    }
    else if(digitalRead(5) && digitalRead(6)){
        actualNum[keyCounter] = 0;
        keyCounter--;
    }
}
else{
    //R1
    if(digitalRead(2)){
        //C1
        if(digitalRead(6)){
            actualNum[keyCounter] = 1;
            keyCounter++;
        }
        //C2
        else if(digitalRead(7)){
            actualNum[keyCounter] = 2;
            keyCounter++;
        }
        //C3
        else if(digitalRead(8)){
            actualNum[keyCounter] = 3;
            keyCounter++;
        }
    }
}

//R2
if(digitalRead(3)){
    //C1
    if(digitalRead(6)){
        actualNum[keyCounter] = 4;
        keyCounter++;
    }
    //C2
    else if(digitalRead(7)){
        actualNum[keyCounter] = 5;
        keyCounter++;
    }
    //C3
    else if(digitalRead(8)){
        actualNum[keyCounter] = 6;
        keyCounter++;
    }
}

//R3
if(digitalRead(4)){
    //C1
    if(digitalRead(6)){
        actualNum[keyCounter] = 7;
        keyCounter++;
    }
    //C2
    else if(digitalRead(7)){
        actualNum[keyCounter] = 8;
        keyCounter++;
    }
    //C3
    else if(digitalRead(8)){
        actualNum[keyCounter] = 9;
        keyCounter++;
    }
}

//R4

```

```

if(digitalRead(5)){
  //C1
  if(digitalRead(6) && keyCounter >= 0){
    actualNum[keyCounter] = 0;
    keyCounter--;
  }
  //C2
  else if(digitalRead(7)){
    actualNum[keyCounter] = 0;
    keyCounter++;
  }
  //C3
  else if(digitalRead(8)){
    checkPassword();
  }
}
}

//print Keys
if( digitalRead(2) || digitalRead(3) || digitalRead(4) || digitalRead(5) ){

  //Serial.print(actualNum[0]);
  //Serial.print(actualNum[1]);
  //Serial.print(actualNum[2]);
  //Serial.print(" ");
  //Serial.print(actualNum[3]);
  //Serial.print(actualNum[4]);
  //Serial.print(actualNum[5]);
  //Serial.print(" ");
  //delay(200);
}

}

void setup() {
  Serial.begin(9600);
  for (int i = 2; i <= 10; i++) {
    pinMode(i, OUTPUT);
  }
  pinMode(11, INPUT);
  digitalWrite(9,HIGH);
}

void loop() {
  if (Serial.available() > 0) {
    char data = Serial.read();

    if (data == '0') {
      failCounter = 3;
    }
  }

  if( digitalRead(11) ){
    failCounter = 0;
    digitalWrite(10, LOW);
  }

  if( (failCounter >= 3) ){
    digitalWrite(10, HIGH);
  }
  else{
    getKey();
  }
}

```

Simulación en Proteus y Montaje en tablas de nodos





