Universidad Rey Juan Carlos

PROYECTO FINAL

Cárcel inteligente

Grupo 3

SISTEMAS EMPOTRADOS

Grado en Ingeniería Informática

Curso 2024/2025

Lucía Carmona Butowicz, Ainara Vanesa Tusan Groza, María García Martín y Adriana Hinojar Guzmán

INDICE

PROYECTO

El objetivo de nuestro proyecto es implementar una cárcel del futuro que sea capaz de detectar una fuga al instante y automatizada, es decir, responder sin intervención humana.

Para poder desarrollar nuestros objetivos, hemos establecido dos estados:

- Estado de alarma:
    - Alarma sonora
    - Cierre de la cárcel
    - Búsqueda del preso fugado
- Fin estado de alarma:
    - Pin correcto
    - Vuelta a la normalidad

IMPLEMENTACIÓN

De componentes principales, podemos destacar la placa de Arduino Uno, que controla todos los elementos electrónicos de la cárcel; un Power Supply Module para poder tener más alimentación para el sistema; y dos protoboards, donde los diferentes componentes utilizados estarán conectados mediante claves M-M, H-H y M-H.

Para la torre de vigilancia, hemos usado un led de alta intensidad para poder alumbrar todos los rincones de la cárcel y una resistencia de 10 k para ésta, así como un servomotor que permite que la torre pueda moverse de un lado a lado.

En cuanto a la alarma, hemos usado un sensor ultrasónico para detectar el movimiento en caso de la fuga de un preso, y un buzzer para la alarma sonora.
Por último, para la puerta principal, hemos necesitado un teclado matricial de 3x4 para introducir la contraseña que desactiva el estado de alarma, un led RGB que sirve para indicar el estado del sistema, y un micro servomotor que permite la apertura o cierre de la puerta principal.

PASOS DADOS

Comenzamos haciendo una lluvia de ideas para decidir que queríamos crear y como desarrollarlo fijándonos en proyectos anteriores en el blog para ver que trabajos estaban ya implementados y evitar repeticiones. También tuvimos en cuenta los materiales que nos proporcionó la universidad en nuestra caja de componentes porque queríamos aprovecharlos al máximo.

Una vez decidida la idea, durante las clases prácticas empezamos poco a poco a montar uno a uno los componentes y a programar su código correspondiente. Después, íbamos incorporando cada vez más componentes, y cuando se nos presentaban nuevos problemas en el desarrollo, los hemos ido solucionando, hasta tener todo el funcionamiento conjunto del proyecto.

Una vez teníamos lista la funcionalidad completa comenzamos a crear nuestra maqueta de la cárcel. A construir las diferentes celdas, pintar y colocar cada componente en su sitio.

Finalmente, comprobamos el funcionamiento global e hicimos unos pequeños ajustes en relación con las dimensiones de la maqueta.

REPARTO DE TAREAS

En esta práctica nos hemos repartidos las tareas de manera equitativa. Nuestro grupo ha asistido a todas las clases prácticas que ha habido, por lo que hemos estado durante todo el cuatrimestre trabajando de manera constante en el proyecto.

Además, al haber trabajado siempre juntas en el aula, todas hemos aprendido a realizar correctamente las conexiones de los componentes y a programar el código del proyecto. Esto nos ha permitido comprender de forma global y práctica el funcionamiento del sistema, sin tener que dividir el trabajo en partes, realizarlo cada una en su casa y luego unirlo en el proyecto final. Cada una ha participado activamente en todo el proceso, lo que ha favorecido tanto nuestro aprendizaje como la cohesión del grupo.

COSTES DE MATERIALES

NOMBRE	**COSTE**
Arduino UNO	20 € (PU*)
Protoboards (2 unidades)	3 € (PU* y MP***)
Servomotor	3,66 € (PU*)
Micro-servomotor	1,6 € (PU*)
LED RGB	2 € (PU*)
LED alta intensidad	0,80 €
Resistencia 10k	0,6 € (PU*)
Buzzer	0,75€(PU*)
Sensor ultrasónico	2,39€ (PU*)
Sensor PIR**	2,80 €
Teclado Matricial 3x4	2,66 €
Cables (pack variado M-M, H-M, H-H)	8,48 € (MP***)
Power Supply Module	1 €(PU*)
Power Jack	1 € (MP ***)
Materiales maqueta (cartón, pintura, palos para los barrotes, cartulinas...)	10€ (MP***)
Muñecos LEGO	6,3 €
Pistola de silicona y cola	15 € (MP***)
TOTAL: 82, 04 €	
TOTAL REAL: 12,56 €	

(*PU= proporcionado por la universidad)

(** Como explicaremos en la siguiente sección, compramos este sensor, pero finalmente no pudimos usarlo)

(*** MP= materiales propios)

PROBLEMAS Y SOLUCIONES

El primer problema que tuvimos fue la sensibilidad del sensor PIR. Nuestra primera idea era usar un sensor PIR para la detección del movimiento del muñeco, pero al ver que este era demasiado sensible y detectaba leves movimientos a una distancia de 5 metros (aun poniendo límites de distancia en el código) decidimos no usarlo y solo usar el ultrasónico.

El segundo problema fue la falta de pines ya que al tener una gran cantidad de materiales que necesitaban pines digitales o pines PWM tuvimos que buscar soluciones ya que teníamos un límite de 13 pines digitales.

Una de las primeras soluciones que encontramos fue usar un alargador de pines 74HC95 proporcionado por la universidad (este usaba 3 pines digitales y generaba 8). Pero no resolvió nuestro problema ya que solo podía expandir salidas y no podía leer datos de entrada por lo que no servía ni para el teclado matricial (que ocupaba muchos pines y detecta datos de entrada) ni para el led RGB (que usaba pines PWM) ni para el ultrasónico. Debido que no nos servía para otros componentes seguíamos sin tener esos 3 pines habilitados que necesita el 74HC95.

Finalmente, la solución que encontramos fue intentar utilizar solo los pines digitales que fuesen indispensables en las diferentes componentes. En el teclado matricial que tiene la funcionalidad de 3x4 (3 columnas x 4 filas), decidimos quitar unos de los pines, una fila, y así “ahorrarnos” un pin digital, teniendo en cuenta que una de las filas del teclado no funcionará, por lo que la contraseña utilizada para la vuelta a la normalidad en la cárcel no puede tener los números de esa fila en concreto. También inicialmente teníamos las salidas ECHO y TRIG del ultrasónico cada uno a un pin digital. Conseguimos conectar ECHO a un pin analógico por lo que de esta manera nos “ahorramos” un pin digital más para poder usarlo en los demás componentes.

Pero a pesar de todas estas medidas seguimos sin tener los pines suficientes para incorporar una pantalla LCD que inicialmente queríamos colocar para que fuese informando del estado del sistema y las diferentes acciones que se iban tomando en la cárcel, por lo que al final todos estos mensajes aparecen en el Serial Monitor.

El tercer y último problema que tuvimos fue la falta de voltaje, ya que, al tener tanta cantidad de componentes, los servomotores (de la puerta y el faro) y el LED de alta intensidad no funcionaban con el voltaje de la placa proporcionado a través del USB conectado al ordenador, por lo que la solución ha sido usar una segunda protoboard donde conectamos un Power Supply module proporcionado por la universidad y que funciona mediante un cable Power Jack que hemos enganchado a la corriente. En esta protoboard hemos incluido los dos servomotores y el led de alta intensidad, para que así estos pudiesen obtener la energía necesaria para su correcto funcionamiento.

CÓDIGO:

```
#include <Servo.h>
#include <Keypad.h>

// Definición de pines
#define PIN_TRIG 9                  // Pin de salida del sensor ultrasónico
#define PIN_ECHO A0                 // Pin de entrada del sensor ultrasónico
#define PIN_BUZZER 8                // Pin del buzzer (alarma sonora)
#define PIN_LED_ROJO 2              // LED indicador de alarma activa
#define PIN_LED_VERDE 4             // LED indicador de estado normal
#define PIN_LED_ALTA_INTENSIDAD 3   // LED que se enciende con la alarma
#define PIN_SERVOMOTOR 7            // Servomotor que abre/cierra la puerta
#define PIN_SERVOMOTOR_FARO 10      // Servomotor que mueve el faro

Servo miServo;       // Objeto para controlar el servomotor de la puerta
Servo servoFaro;     // Objeto para controlar el servomotor del faro

bool puertaAbierta = true;   // Estado inicial de la puerta
bool alarmaActiva = false;   // Indica si la alarma está activa
int fondo = 0;  // Almacenará la distancia hasta el fondo de la maqueta

// Configuración del teclado 3x3
const byte FILAS = 3;
const byte COLUMNAS = 3;
char keys[FILAS][COLUMNAS] = {
  {'1','2', '3'},
  {'4','5', '6'},
  {'*','0', '#'}
};
// Pines conectados al teclado
byte pinFilas[FILAS] = {6, 0, 11};
byte pinColumnas[COLUMNAS] = {5, 12, 13};
// Crear objeto para el teclado
Keypad teclado = Keypad(makeKeymap(keys), pinFilas, pinColumnas, FILAS, COLUMNAS);
String codigoIngresado = "";  // Guarda lo que el usuario ingresa desde el teclado
void setup() {
  Serial.begin(9600);  // Inicia la comunicación serial

 // Configura los pines
  pinMode(PIN_TRIG, OUTPUT);
  pinMode(PIN_ECHO, INPUT);
  pinMode(PIN_BUZZER, OUTPUT);
  pinMode(PIN_LED_ALTA_INTENSIDAD, OUTPUT);
  pinMode(PIN_LED_ROJO, OUTPUT);
  pinMode(PIN_LED_VERDE, OUTPUT);

// Asocia servos a sus pines
  miServo.attach(PIN_SERVOMOTOR);
```

```
  servoFaro.attach(PIN_SERVOMOTOR_FARO);

// Posición inicial de los servos
  miServo.write(140);      // Puerta abierta
  servoFaro.write(180);    // Faro apuntando en una dirección

// Medir distancia inicial de fondo
  fondo = medirDistancia();
  Serial.print("Distancia de fondo: ");
  Serial.println(fondo);

// Encender LED verde al inicio
  encenderLEDVerde();
}
void loop() {
  // Leer tecla presionada
  char tecla = teclado.getKey();
  if (tecla) {
    Serial.print("Tecla presionada: ");
    Serial.println(tecla);
    if (tecla == '*') {
      // Reiniciar el código si se presiona '*'
      codigoIngresado = "";
      Serial.println("Código reiniciado.");
    } else {
      // Agregar tecla al código
      codigoIngresado += tecla;
      Serial.print("clave: ");
      Serial.println(codigoIngresado);
      // Verifica si el código ingresado es correcto
      if (codigoIngresado == "2604#") {
        desactivarAlarma();  // Apaga la alarma y abre la puerta
        Serial.println("Código correcto. Puerta abierta.");
        codigoIngresado = ""; // Reinicia el código ingresado
      }
    }
  }
  // Medir distancia con sensor ultrasónico
  int distancia = medirDistancia();
  // Si se detecta algo más cerca que el fondo y no hay alarma activa, la activa
  if (distancia > 0 && distancia < fondo - 10 && !alarmaActiva) {
    activarAlarma();
  }
  // Si la alarma está activa, hacer sonar y mover faro
  if (alarmaActiva) {
    sonarAlarma();
    moverFaro();
    digitalWrite(PIN_LED_ALTA_INTENSIDAD, HIGH);
  } else {
```

```
    digitalWrite(PIN_LED_ALTA_INTENSIDAD, LOW);
  }
  delay(100);  // Pequeño retardo para estabilidad
}
void activarAlarma() {
  alarmaActiva = true;
  encenderLEDRojo();  // LED rojo indica estado de alerta
  // Si la puerta está abierta, cerrarla
  if (puertaAbierta) {
    miServo.write(0);      // Cierra la puerta
    puertaAbierta = false;
    Serial.println("Puerta cerrada.");
  }
}
void desactivarAlarma() {
  alarmaActiva = false;
  miServo.write(140);      // Abre la puerta
  puertaAbierta = true;
  encenderLEDVerde();      // Indica estado seguro
  noTone(PIN_BUZZER);      // Apaga sonido
  digitalWrite(PIN_LED_ALTA_INTENSIDAD, LOW);  // Apaga luz intensa
  servoFaro.write(90);     // Faro se queda en el centro
}
void moverFaro() {
  static int posicion = 180;     // Posición del faro
  static bool direccion = false; // Dirección de movimiento (izq-der)
  // Mueve el faro en zigzag
  if (direccion) {
    posicion += 5;
    if (posicion >= 180) direccion = false;
  } else {
    posicion -= 5;
    if (posicion <= 0) direccion = true;
  }
  servoFaro.write(posicion);  // Mueve faro a nueva posición
  delay(50);  // Espera para dar efecto de movimiento
}
int medirDistancia() {
  // Disparo del sensor ultrasónico
  digitalWrite(PIN_TRIG, LOW);
  delayMicroseconds(2);
  digitalWrite(PIN_TRIG, HIGH);
  delayMicroseconds(10);
  digitalWrite(PIN_TRIG, LOW);
  // Medir el tiempo de retorno del pulso
  long duracion = pulseIn(PIN_ECHO, HIGH, 30000); // 30ms timeout
  if (duracion == 0) return -1; // Si no se detecta nada
  // Convertir tiempo en distancia en cm
```

```
  return (duracion / 2) / 29.1;
}
void sonarAlarma() {
  // Cambia entre tonos para simular una sirena
  static unsigned long ultimaNota = 0;
  static bool tonoAlto = true;
  unsigned long ahora = millis();
  if (ahora - ultimaNota > 300) {
    if (tonoAlto) {
      tone(PIN_BUZZER, 600);  // Tono agudo
    } else {
      tone(PIN_BUZZER, 400);  // Tono grave
    }
    tonoAlto = !tonoAlto;
    ultimaNota = ahora;
  }
}
void encenderLEDVerde() {
  digitalWrite(PIN_LED_ROJO, LOW);
  digitalWrite(PIN_LED_VERDE, HIGH); // Estado normal
}
void encenderLEDRojo() {
  digitalWrite(PIN_LED_ROJO, HIGH);  // Estado de alerta
  digitalWrite(PIN_LED_VERDE, LOW);
}
```

CASOS DE USO

El caso de uso principal que contiene la funcionalidad de nuestro proyecto consiste en la detección de un intento de fuga. El sensor ultrasónico colocado en el pasillo, próximo a las celdas de los presos, detecta el movimiento cuando alguno de los presos intente escapar de su celda.

En ese momento el buzzer lanzará la alarma que sonará de manera ininterrumpida hasta que finalice el estado de alarma y, en el mismo instante, se cerrará la puerta principal de la cárcel para que el preso no pueda salir del recinto y el LED RGB próximo a la puerta cambiará de verde a rojo ya que está asociado al estado de la puerta.

También, al iniciar el estado de alarma, el faro se pondrá en funcionamiento y comenzará a "buscar" al recluso que haya escapado mediante movimientos de 0 a 180º cubriendo toda la maqueta, e iluminando todo con su LED de alta intensidad.

Una vez que el preso haya sido atrapado y llevado de vuelta a la celda pertinente, los funcionarios de la prisión podrán desactivar el estado de alarma y volver a la normalidad introduciendo la contraseña correcta en el teclado matricial. En este se permite borrar la combinación introducida en caso de que se haya producido algún error al introducir los dígitos. Y cuando se pulsa la combinación correcta, toda la maqueta vuelve a su estado inicial. El sonido de alarma cesa, el faro apaga su luz y deja de estar en constante movimiento, la puerta principal se abre, y, por consiguiente, el LED RGB cambia de nuevo al color verde.