

PROYECTO PRÁCTICO ARDUINO

Institución: Universidad Tecnológica Nacional Facultad Regional de Villa Maria

Carrera: Ingeniería en Sistemas de Información

Cátedra: Teoría de Control, Ciclo Lectivo 2023

Fecha de Entrega: 24 de Noviembre de 2023

Docentes del Curso

- Ing. Jose Luis Catalano
- Ing. Nicolas Oliva

Integrantes del Grupo

- Calvo Nazarena - 13925 - nazacalvo15@gmail.com
- Corrá Delfina - 14115 - delfinacorra@gmail.com
- Fumero Paula - 14160 - paulitafumero@gmail.com
- Venier Aldana - 14112 - aldanavenier@gmail.com

ÍNDICE

ÍNDICE.....	1
INFORME.....	2
Resumen.....	2
Descripción del Proyecto.....	2
Objetivo Principal.....	2
Objetivo General.....	2
Introducción.....	3
Contexto y justificación del Proyecto.....	3
Problemática a Resolver.....	3
Marco Teórico.....	3
Diseño y Metodología.....	4
Esquema de Conexionado.....	6
Desarrollo del Código.....	7
Código.....	7
Funcionamiento del Código.....	9
Resultados Obtenidos.....	9
Conclusiones.....	10

INFORME

Resumen

Descripción del Proyecto

En el contexto de la cátedra Teoría de Control, Ciclo Lectivo 2023, se llevará a cabo el desarrollo de un proyecto práctico con Arduino. Este, tiene la finalidad de lograr diseñar, construir y poner en funcionamiento un circuito, con un módulo en particular, para sensar algún dato o métrica de nuestro entorno.

Para el desarrollo del mismo, contamos con los siguientes elementos: una placa de Arduino uno, una placa Ethernet, una protoboard, un Relay, un sensor de sonido modelo HW-485, un diodo emisor de luz y una resistencia. Teniendo en cuenta los elementos para realizar las conexiones entre los elementos del circuito y la computadora.

Objetivo Principal

Nuestro objetivo principal para este proyecto fue lograr la implementación correcta del módulo particular en el circuito Arduino, para censar los distintos niveles de sonido del ambiente y dar una señal al sobrepasar cierto nivel.

La implementación incluye el modelado del circuito, como su construcción física, programación y configuración de la conexión para recibir los datos en una computadora por medio de paquetes enviados por el arduino.

Objetivo General

Lograr construir el circuito Arduino, con el módulo particular seleccionado, y lograr la implementación del código para capturar los datos de manera adecuada y comandar a través de un sistema.

Los principales elementos del circuito Arduino son el sensor HW-485, un LED, un microcontrolador y una placa de red.

Introducción

Contexto y justificación del Proyecto

El proyecto se presentó a modo de desafío con la finalidad de poner en práctica conceptos teóricos sobre sistemas de control, vistos en la cátedra, aplicarlos sobre una situación real y poder evidenciarlos físicamente.

Problemática a Resolver

La problemática que se nos plantea a resolver es lograr construir el circuito físicamente realizando las conexiones adecuadas, programar el dispositivo mediante la aplicación arduino IDE para poder utilizar el sensor, que en nuestro caso es un sensor de sonido, y poder capturar datos del ambiente.

Programar el dispositivo refiere a la construcción de un código que nos permita inicializar el arduino y utilizar de forma correcta el sensor, el cual captará las ondas sonoras del entorno y las convertirá en señales eléctricas que el Arduino podrá interpretar. El código se encarga de recibir los e interpretar los decibelios mediante la aplicación y, si se supera cierta cantidad de decibelios, se le indicará al arduino que, por medio del Relay, la acción de controlar la cual en este caso es el encendido del LED. De esta manera cuando hay ruido en el ambiente, este se evidencia mediante el led encendido.

Con la construcción de este circuito podremos crear, por ejemplo, aplicaciones como sistemas de seguridad activados por sonido, luces que se encienden al detectar aplausos, o incluso proyectos de música interactiva.

Marco Teórico

En el proyecto se aplicaron algunos de los conceptos vistos sobre sistemas de control a lo largo de la cursada de la cátedra.

Se aplican conceptos claves sobre sistemas dinámicos, comprendiendo la importancia de mantener o modificar el comportamiento de estos sistemas a través de la retroalimentación y la acción de un controlador.

Un sistema dinámico es un sistema cuyo estado evoluciona con el tiempo, mientras que un sistema de control es una combinación de componentes físicos conectados de una manera tal que puedan comandar, dirigir o regular así mismo o a otro sistema. Esto lo hacen a través de una señal de retroalimentación, la cual a partir de los resultados derivados de una

actividad, estos se reintroduce de nuevo en el sistema con el objetivo de mantener un control y una optimización de su comportamiento.

Los sensores son herramientas que detectan y responden a algún tipo de información del entorno físico. Un sensor de sonido sensa la intensidad del sonido o ejecutar alguna instrucción posterior a detectar un ruido. Un sensor de sonido o micrófono, generalmente proporciona una señal analógica al Arduino.

Una señal analógica es una señal generada por algún tipo de fenómeno electromagnético, la cual es representable por una función matemática continua en la que es variable su amplitud y periodo en función del tiempo. En este caso, la señal analógica representa la amplitud de la onda sonora captada por el micrófono. Es importante tener en cuenta que la señal analógica del micrófono puede variar en amplitud y frecuencia dependiendo del sonido captado.

El Arduino Uno tiene pines de entrada analógica que le permiten leer esta señal analógica y convertirla en un valor digital que puede ser utilizado por el código del Arduino.

En un principio, se decidió trabajar con una señal analógica, pero durante el desarrollo del proyecto, se terminó optando por cambiar el tipo de enfoque para poder trabajar con una señal de entrada digital.

Esto nos permitió ser más consistentes con los datos capturados, y las acciones de control, y menos susceptibles a interferencias externas.

Diseño y Metodología

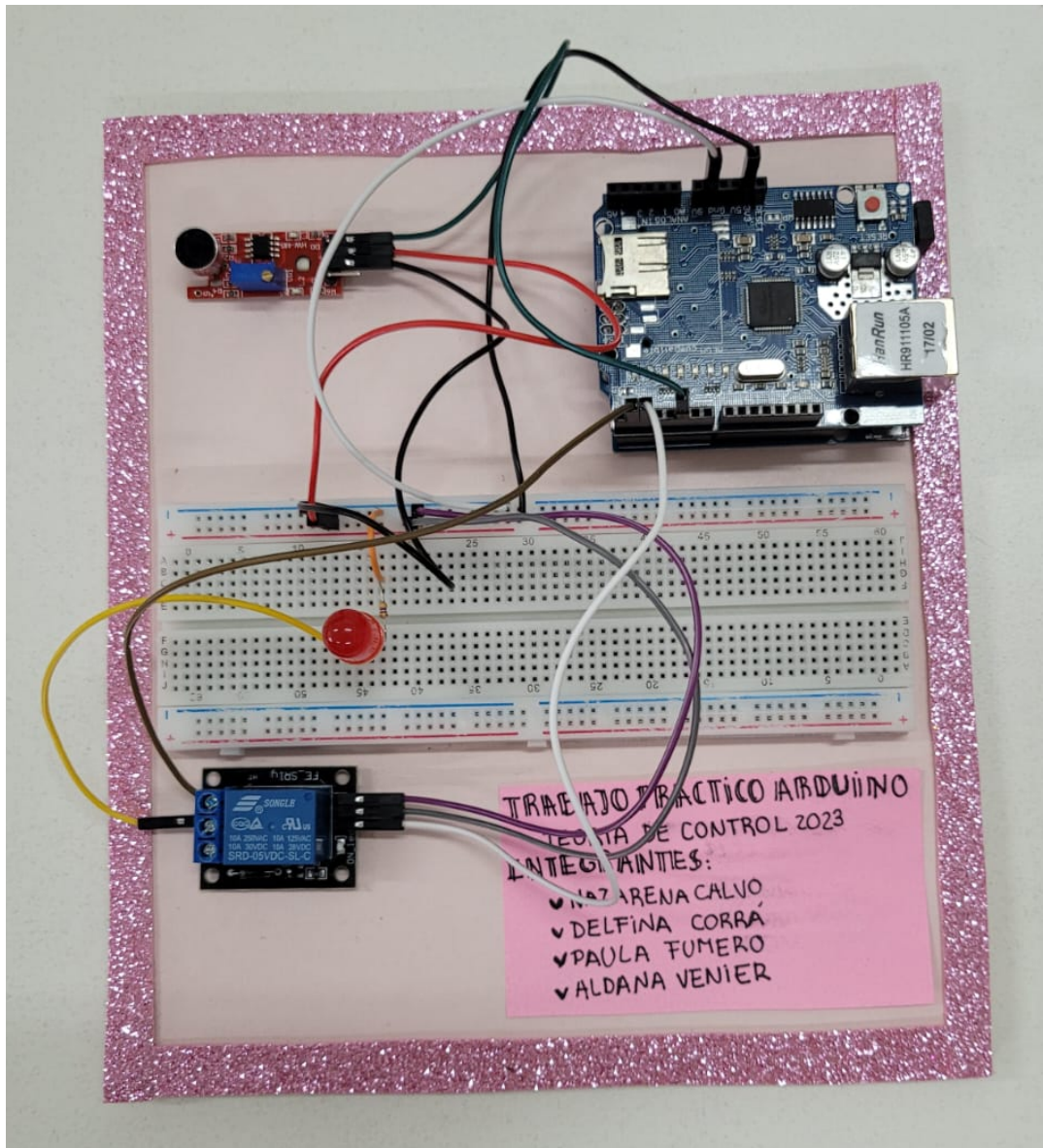
Antes de trabajar con el sensor de sonido, se realizó el circuito implementando un dispositivo pulsador conectado al Arduino. La problemática de utilizar dicho componente fue que, para el proyecto, se debía de sensar algún dato del entorno y en base a este aplicar una acción de control. Cosa que, con la implementación del pulsador, no fue posible.

Debido a esto se decidió cambiar de dispositivo, por un micrófono el cual sensara sonido y sus niveles en decibelios. Para la construcción de este nuevo circuito, se siguieron los siguientes pasos:

1. Se colocó el LED en la protoboard. El ánodo o el lado más largo del LED, se conectó al riel positivo, mientras que el lado más corto se conectó al riel negativo.

2. Para limitar la corriente que ingresa al LED, colocamos una resistencia conectada en serie con el ánodo del LED. Es decir, desde la misma columna donde se colocó la pata del led positiva , se conecta la resistencia hacia el riel negativo.
3. Antes de seguir con las conexiones entre los elementos, desde el Arduino conectamos dos cables a la protoboard para establecer la conexión de energía. Conectamos uno usando un cable jumper y lo conectamos desde el pin de alimentación de 5V del Arduino a un riel positivo de la protoboard. La otra conexión, también con un cable jumper, conectamos el pin de tierra (GND) del Arduino a un riel negativo. Esta conexión proporcionará energía a los componentes.
4. Incorporamos un Relay, el mismo funciona como intermediario entre el Arduino que recibe la señal del micrófono y el LED. Los pines VCC y GND del Relay se conectan a las líneas de alimentación de la protoboard, generalmente en el riel positivo y negativo respectivamente, asegurando una fuente de alimentación adecuada para el Relay. El pin de control del Relay (IN) se conecta a un pin digital del Arduino (2), permitiendo al Arduino activar o desactivar el Relay según sea necesario.
5. Para incorporar el Led al circuito, se realiza la conexión con el relay. En primer lugar con un cable jumper, desde una de la terminal común hasta el canal de la protoboard donde está el led. Desde la terminal abierta del Relé se conecta a un pin digital del arduino (1).
6. Se realiza la conexión del micrófono, considerando que no se utilizará la conexión analógica. La conexión VCC se conecta a la protoboard a una línea de 5V y el GND a una de tierra. La señal de salida del micrófono (OUT) se conecta a un pin digital del Arduino (4).
7. La última incorporación del sistema fue el módulo Ethernet. Este se coloca encima del Arduino, por lo que las conexiones que tenía esta placa, se trasladaron al módulo ethernet. La configuración de la conexión con esta placa se realiza sobre la computadora, al que se conectará por cable cruzado. A través del protocolo IPV4 se configuran las IPs de los dispositivos.
8. Por último, se programó el dispositivo para que este pudiera iniciarse, capturar los datos del ambiente por medio del sensor, enviarlos a la computadora y, según el valor de estos, aplicar la acción de control que en este caso, es el encendido del LED y el envío de mensaje a través del protocolo UDP.

Esquema de Conexionado



Componentes Conectados

- Micrófono HW-485 conectado al Arduino y a la protoboard para detectar niveles de sonido, utilizando un pin digital para la lectura de la señal.
- Luz LED conectado al Relé, que está controlado por el Arduino. El Relé actúa como un interruptor para encender o apagar el LED.
- Placa de Red Shield Ethernet HR911105A colocada encima del Arduino, implicando que el Arduino y sus conexiones ahora están enlazados a este módulo para la comunicación Ethernet.

Funcionalidades Asociadas

- El Arduino recibe la señal del micrófono y, dependiendo de los niveles de sonido detectados, controla el Relé para encender o apagar el LED.
- El módulo Ethernet habilita la comunicación en red, permitiendo la interacción con otros dispositivos o la conexión a internet.

Desarrollo del Código

Código

```
#include <SPI.h>
#include <Ethernet.h>
#include <EthernetUdp.h>

byte mac[] = {0xDA, 0xAB, 0xBE, 0xEF, 0xFE, 0xED};
IPAddress ip(192, 168, 1, 201);
IPAddress ipServer(192, 168, 1, 200);
unsigned int localPort = 5005;
unsigned int appPortM = 8080;

String paquete = "";

// Buffer de envío y recepción
char packetBuffer[UDP_TX_PACKET_MAX_SIZE] = "";
char ReplyBuffer[UDP_TX_PACKET_MAX_SIZE] = "";

// Ethernet Udp Packet
EthernetUDP Udp;
#include <SD.h> // PARA LECTURA/ESCRITURA SD
File myFile;

const int pinRelay = 2; // El pin del relé
const int pinMicrofono = 5; // El pin del módulo de micrófono

void setup() {
  Serial.begin(9600);
  pinMode(pinRelay, OUTPUT); // Configura el pin del relé como salida
  pinMode(pinMicrofono, INPUT); // Configura el pin del micrófono como entrada
  Ethernet.begin(mac, ip);
  Udp.begin(localPort);
  Serial.print("SD");
  if (!SD.begin(4)) {
```



```

        return;
    }
    Serial.println("OK");
    SPI.begin();
}

void loop() {
    recibirUDP();
    // Lee el valor del micrófono
    int estadoMicrofono = digitalRead(pinMicrofono);

    Serial.print("Valor del micrófono: ");
    Serial.println(estadoMicrofono);

    // Si la señal de salida digital del módulo del micrófono es
    // 1 (escucho un ruido significativo), enciende el relé
    if (estadoMicrofono == 1) {
        String message = "RUIDO; LED ON"; //mensaje a enviar por
        ethernet
        const char* charArray = message.c_str(); //lo convertimos a
        char
        enviarUDP(charArray, ipServer, appPortM); //lo enviamos
        digitalWrite(pinRelay, HIGH); // Enciende el relé
    } else {
        digitalWrite(pinRelay, LOW); // Apaga el relé
    }
    delay(1000);
    Serial.println("Waiting");
}

void enviarUDP(char paquete[], IPAddress ip, unsigned int
port) {
    Udp.beginPacket(ip, port); //Start building up a packet to
    send to the remote host specific in host and port
    Udp.write(paquete);
    int resultado = Udp.endPacket(); //returns 1 if successful,
    0 if there was a problem resolving the hostname or port
    Serial.print("Resultado del envío: ");
    if (resultado == 1){
        Serial.println("enviado con éxito");
    } else {
        Serial.println("error");
    }
}
}

```

Funcionamiento del Código

El resultado de la programación del Arduino se puede observar en el fragmento de código presentado anteriormente. Parte de este, fue proporcionado como material del cual guiarnos y adaptar a necesidad para trabajar con conexión vía Ethernet.

Comenzamos con la función “setup()” iniciando la conexión mediante Ethernet con el arduino y escuchamos el puerto indicado con la operación “begin” en las respectivas instancias. También configuramos los pines del Relé y el módulo de sonido como “OUTPUT” (salida) e “INPUT” (entrada) respectivamente.

Luego, dentro del ciclo definido con “loop()”, leemos el valor de la señal de salida digital del módulo de sonido, que devolverá un 1 si el sonido excede el valor fijado con el potenciómetro y 0 en caso contrario.

Si la señal digital leída es 1, le indicamos al Relay que aplique la acción de control, la cual prenderá el LED que recibe su señal. Por medio de ethernet, enviaremos un mensaje a la pc conectada a la placa de red “Shield Ethernet HR911105A” colocada sobre el Arduino.

Resultados Obtenidos

Luego del desarrollo del proyecto descrito a lo largo del documento, obtuvimos como resultado un circuito arduino que implementa un micrófono (HW-485), mediante el cual podemos detectar y censar la presencia de sonido en el ambiente que lo rodea. Con la particularidad de que solo se aplicará la acción de control, el encendido del LED, cuando se detecten sonidos que se encuentren dentro de un rango definido de decibelios.

Lo que se buscó fue ajustar la calibración del micrófono en diferentes niveles. El mismo fue censando de manera variada, a pesar de colocar diferentes valores de Delay para que las mediciones no se realizarán tan seguidas y modificar la calibración. En las pruebas realizadas, el micrófono logró detectar aquellas voces en el rango definido para que prenda el led si se trataba de voces graves.

El circuito se encargará de prender la luz led siempre y cuando esté detectando sonidos, esto le permitirá al circuito avisarnos que hay un sonido en el ambiente que está dentro del rango definido de decibelios que el sensor puede detectar.

Al trabajar con una señal digital solo podemos identificar la presencia de ruido que afecta al sensor de sonido, pero no nos permite identificar cual es la intensidad exacta de ese sonido.

Conclusiones

Una vez confeccionado el circuito, surgieron inconvenientes respecto a la conexión ethernet entre el arduino y la computadora ya que no se lograba establecer la comunicación. Descubrimos que esto era debido al cable que estábamos utilizando ya que no era cruzado, por lo que durante las clases tuvimos que confeccionarlo, y debido a que en el código, no estaba declarada y definida la conexión de manera correcta, lo que nos llevó tener que modificarlo en repetidas ocasiones.

Al momento de lograr la conexión y poner en funcionamiento el circuito, observamos que al tratar de censar el sonido del ambiente, el micrófono no detectaba todos los sonidos que realizábamos. Para solucionar esta problemática, tuvimos que manipular el regulador de sensibilidad de manera manual en el dispositivo físico, el micrófono. Esto provocó que de los distintos sonidos realizados, sólo algunos eran reconocidos. Debido a esto, tuvimos que especificar al momento del funcionamiento del circuito, que el micrófono no detectara todos los sonidos, sino que sólo aquellos que alcanzaran el rango de sensibilidad en decibelios del micrófono.

Entonces, podemos concluir con que los objetivos planteados al principio del proyecto, se llevaron a cabo de manera satisfactoria y que, a pesar de las dificultades que surgieron durante el desarrollo del circuito, logramos cumplimentar las metas propuestas en tiempo y forma. Llevándonos el aprendizaje de la confección de un circuito utilizando los materiales que se mencionan en este trabajo, la construcción de un cable cruzado, pero lo más importante, poder poner en práctica todo los conceptos que se estudiaron de manera teórica durante el transcurso de la cursada de la cátedra.