



INSTITUTO TECNOLÓGICO DE COSTA RICA

ÁREA DE INGENIERÍA EN COMPUTADORES

CE-4202: TALLER DE DISEÑO ANALÓGICO

Proyecto #2: Control de acceso mediante frecuencias

Estudiantes:

Arturo CHINCHILLA S.
Alejandro CALVO P.

Profesor:

Rodolfo TACSAN CHAVES

I Semestre, 12 de abril de 2019

Índice

1. Introducción	2
2. Objetivos Generales	2
3. Objetivos Específicos	2
4. Descripción del problema	2
5. Descripción de la solución	2
5.1. Código fuente Arduino	3
5.2. Explicación del código	4
6. Diagrama del Diseño	5
7. Resultados Obtenidos	6
8. Análisis de los Resultados	7
9. Conclusiones y Recomendaciones	7

Índice de figuras

1. Circuito esquemático de la solución	5
2. Circuito esquemático para la alimentación con la red eléctrica	5
3. Pantalla del Osciloscopio para la señal de 2.5KHz. Verde: señal de entrada del filtro, rojo: salida del filtro	6
4. Pantalla del Osciloscopio para la señal de 1.5KHz. Verde: señal de entrada del filtro, rojo: salida del filtro	6
5. Pantalla del Osciloscopio para la señal de 500Hz. Verde: señal de entrada del filtro, rojo: salida del filtro	7

Índice de cuadros

1. Introducción

La seguridad se ha vuelto parte fundamental en la vida del ser humano. Motivos como crecientes olas de delitos o la necesidad de tener dispositivos más seguros hace que continuamente se quiera innovar en esta área.

En esta ocasión se plantea el diseño e implementación de un sistema para una cerradura electrónica, el cuál cuenta con 3 modos: abierto, cerrado y un modo de pánico donde se emitirá una alerta continua (pitidos). Cada uno de estos modos es accedido mediante tres frecuencias de entrada específicas. La cerradura se abre en 1500KHz, la cerradura se cierra en 2500KHz y el modo de pánico se activa con una frecuencia de entrada de 500Hz. En la vida real estas frecuencias pueden ser obtenidas desde distintos aparatos, como un mando a distancia, o un circuito especializado.

Primeramente se realiza un proceso de filtrado de la señal, que en caso de coincidir con alguna de las tres antes mencionadas, esta señal alterna se acondiciona utilizando un rectificador, y es enviada a un microcontrolador ATMEGA320P en el cuál utilizando el lenguaje de Arduino se programa cómo se deben comportar las salidas, en 1500Hz aparte de abrir la cerradura debe sonar un pitido, en 2500Hz aparte de cerrar la cerradura se deben sonar dos pitidos y en 500Hz debe permanecer la alarma sonando continuamente.

El proyecto se puede dividir en 5 etapas básicas: alimentación, filtrado, acondicionamiento de la señal (rectificadores), manejo de la señal (ATMEGA320P) y por último las salidas (Buzzer y cerradura).

2. Objetivos Generales

- Aplicar conceptos básicos vistos en clase sobre filtros .

3. Objetivos Específicos

1. Construir un dispositivo capaz de detectar distintas frecuencias y controlar varias señales para tomar una determinada acción.
2. Implementar una alarma de manera que dependiendo la frecuencia de entrada tome uno de tres modos.
3. Diseñar y construir un circuito de alimentación para el circuito de manera que sea alimentado con 110V.

4. Descripción del problema

El sistema recibirá como entrada una onda proveniente del generador de funciones, el cuál va a ir variando en frecuencia. El circuito diseñado deberá de detectar las frecuencias de 500Hz, 1500Hz, 2500Hz y enviar una señal de pánico, apertura y cerrado. Estas señales se utilizan para controlar una cerradura y una alerta sonora.

Cuando se tenga una señal de apertura se debe abrir la cerradura y se tiene que escuchar un pitido. Cuando se tenga una señal de cerrado se debe cerrar la cerradura y se tiene que escuchar dos pitidos. Para la señal de pánico se debe de generar una alarma sonora.

El sistema tiene que estar alimentado únicamente por una conexión a 110V, sin requerir de fuentes de poder externas.

5. Descripción de la solución

Para clasificar las distintas frecuencias en las cuales se tenían que ejecutar acciones, se decidió utilizar tres filtros pasa banda angosta con la frecuencias de corte en 500Hz, 1500Hz y 2500Hz. De esta manera cuando la frecuencia de entrada del sistema está en alguna de las frecuencias de corte de los filtros, esta puede ser clasificada y procesada. Luego de los filtros, cada señal se rectifica para poder trabajar con tensiones constantes. Posteriormente, las señales se conecta a las entrada analógicas del microcontrolador ATMEGA328P. Las salidas al buzzer y al solenoide son controladas por el ATMEGA328P. El código utilizado para el microcontrolador es el siguiente:

5.1. Código fuente Arduino

```
int pin2k5 = A0;
int pin1k5 = A1;
int pin500 = A2;
int pinSolenoid = 9;
int pinBuzzer = 10;
int i = 0;
int state = 0;
int ps = 0;
bool done = false;
bool alarm = false;
int f2k5 [] = {0,0,0,0,0};
int f1k5 [] = {0,0,0,0,0};
int f500 [] = {0,0,0,0,0};
unsigned long previousMillis = 0;
unsigned long currentMillis = 0;

void setup() {
  pinMode(pin2k5, INPUT);
  pinMode(pin1k5, INPUT);
  pinMode(pin500, INPUT);
  pinMode(pinSolenoid, OUTPUT);
  pinMode(pinBuzzer, OUTPUT);
}

void loop() {
  readFilters();
  state = getState();
  switch(state){
    case 0: //Standby
      if(alarm){
        digitalWrite(pinBuzzer, LOW);
        alarm = false;
      }
      break;
    case 1: //Panico
      if(!done){
        digitalWrite(pinSolenoid, LOW);
        done = true;
      }
      currentMillis = millis();
      if (currentMillis - previousMillis >= 200) {
        previousMillis = currentMillis;
        digitalWrite(pinBuzzer, alarm);
        alarm = !alarm;
      }
      break;
    case 2: //Abrir
      if(!done){
        digitalWrite(pinSolenoid, HIGH);
        digitalWrite(pinBuzzer, HIGH);
        delay(300);
        digitalWrite(pinBuzzer, LOW);
        done = true;
      }
      break;
    case 3: //Cerrar
```

```

        if (!done){
            digitalWrite(pinSolenoid , LOW);
            digitalWrite(pinBuzzer , HIGH);
            delay(200);
            digitalWrite(pinBuzzer , LOW);
            delay(100);
            digitalWrite(pinBuzzer , HIGH);
            delay(200);
            digitalWrite(pinBuzzer , LOW);
            done = true;
        }
        break;
    default:
        break;
    }
}

//Funcion para leer la salida de los filtros
void readFilters(){
    f2k5[i] = analogRead(pin2k5);
    f1k5[i] = analogRead(pin1k5);
    f500[i] = analogRead(pin500);
    i = (i+1)%5;
}

//Funcion para calcular el promedio
int promedio(int arreglo[]){
    int prom = 0;
    for (int j = 0; j < 5; j++){
        prom += arreglo[j];
    }
    return prom/5;
}

//Funcion para obtener el estado a partir de los filtros
int getState()
{
    int p2k5 = promedio(f2k5);
    int p1k5 = promedio(f1k5);
    int p500 = promedio(f500);
    int s = 0;
    if(p2k5 >= 115 && p2k5 <= 125 && p1k5 < 80){
        s = 3;
    } else if(p1k5 >= 135 && p1k5 <= 145 && p500 < 180){
        s = 2;
    } else if(p500 >= 350 && p500 <= 370){
        s = 1;
    }
    if (ps != s && s != 0){
        done = false;
        ps = s;
    }
    return s;
}

```

5.2. Explicación del código

El código lo que hace es leer los datos analógicos y promediarlos cada 5 valores. Esto se hace para tener una lectura más limpia del comportamiento cambiante de los filtros. Seguidamente se compara que los valores de las entradas analógicas estén dentro de ciertos valores, que serían los

valores de la salida de los filtros cuando se está en sus frecuencias de corte. Posteriormente, a partir de estos valores se crea una máquina de cuatro estados: standby, abrir, cerrar, pánico. Para así poder ejecutar cada una de las funciones establecidas. En pánico se genera una alarma, en abrir se abre el solenoide y se genera un beep, en cerrar se cierra el solenoide y se generan dos beeps y por último el estado de standby es un estado de espera, no se realiza ninguna acción.

6. Diagrama del Diseño

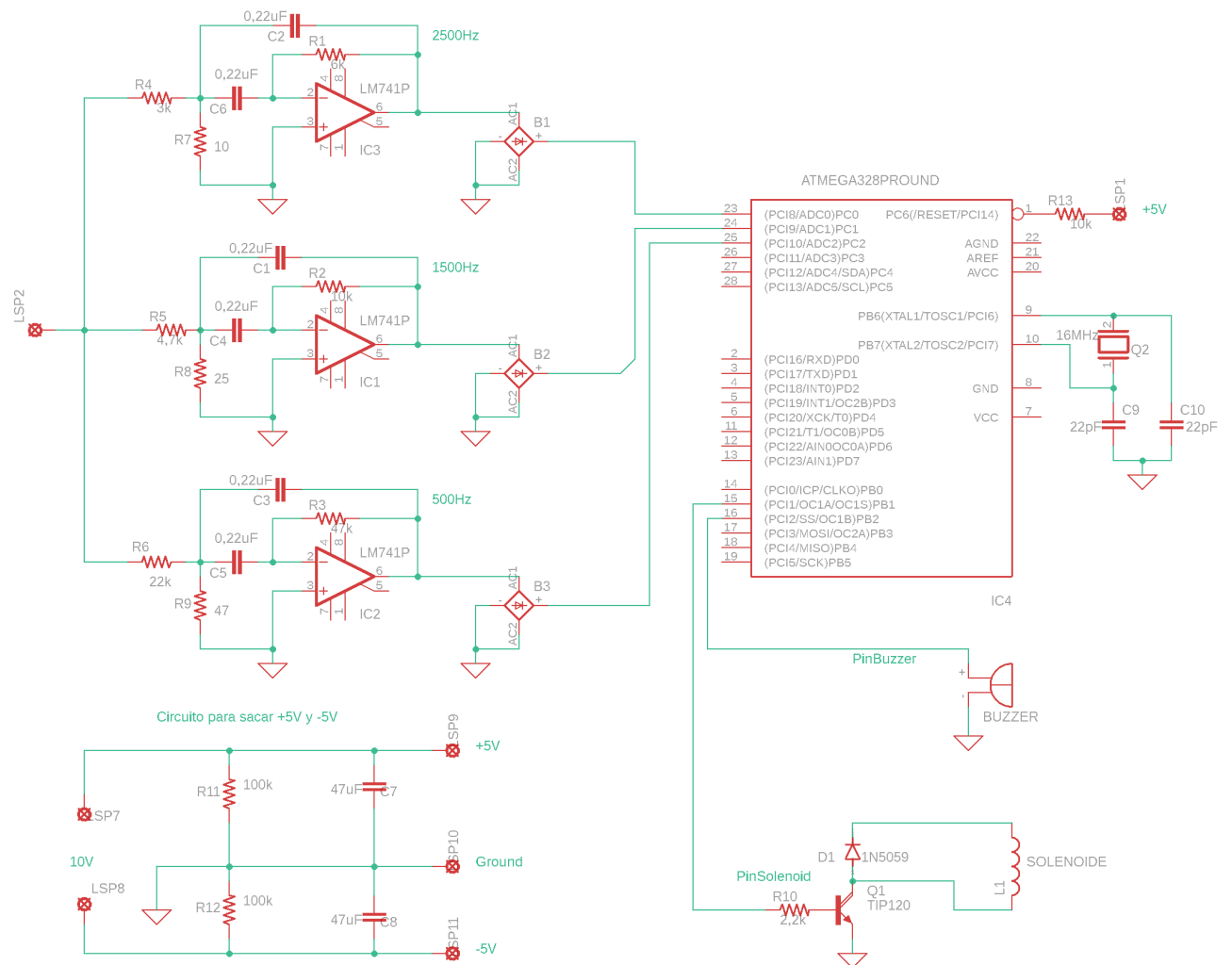


Figura 1: Circuito esquemático de la solución

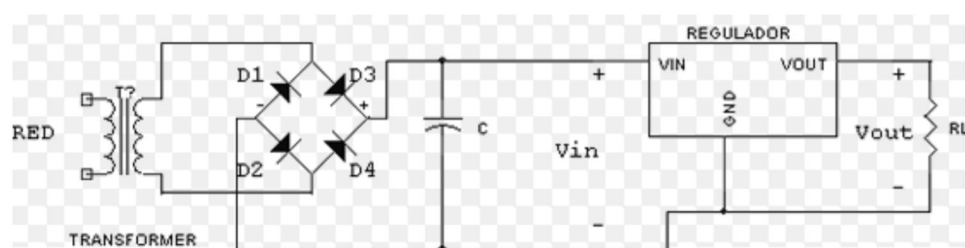


Figura 2: Circuito esquemático para la alimentación con la red eléctrica

7. Resultados Obtenidos

En la figura 3 se observan las señales de entrada y salida en el filtro diseñado para la señal de 2500Hz (Cerrado de la cerradura).

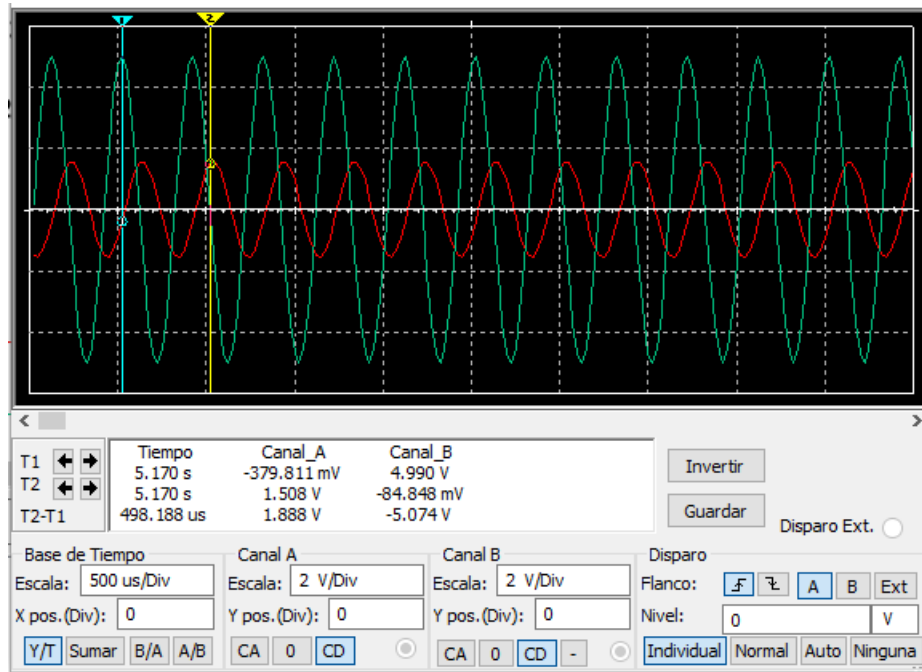


Figura 3: Pantalla del Osciloscopio para la señal de 2.5KHz. Verde: señal de entrada del filtro, rojo: salida del filtro

En la figura 4 se observan las señales de entrada y salida en el filtro diseñado para la señal de 1500Hz (Apertura de la cerradura).

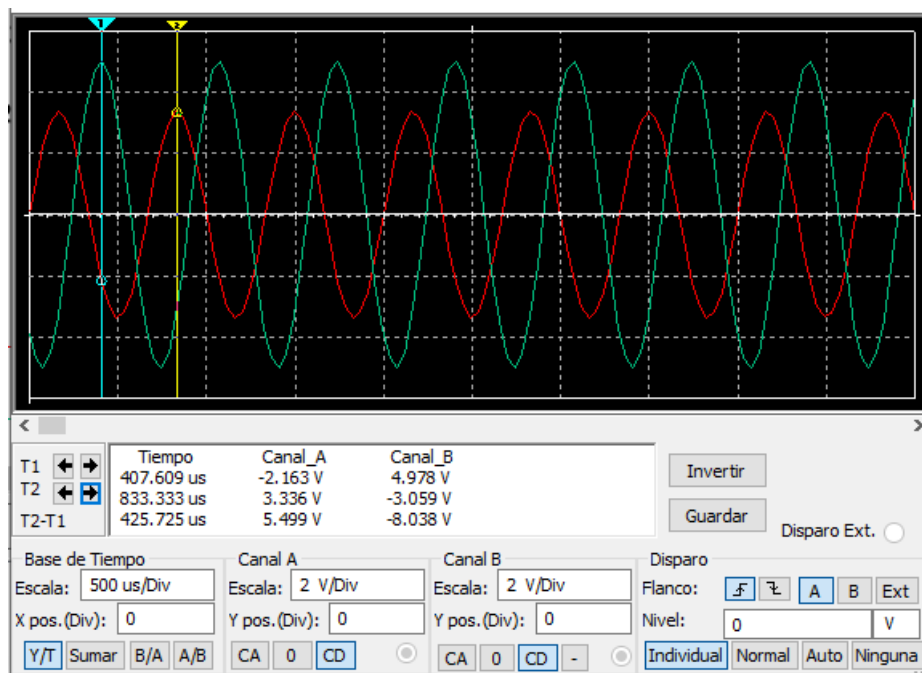


Figura 4: Pantalla del Osciloscopio para la señal de 1.5KHz. Verde: señal de entrada del filtro, rojo: salida del filtro

En la figura 5 se observan las señales de entrada y salida en el filtro diseñado para la señal de 500Hz (Modo de pánico).

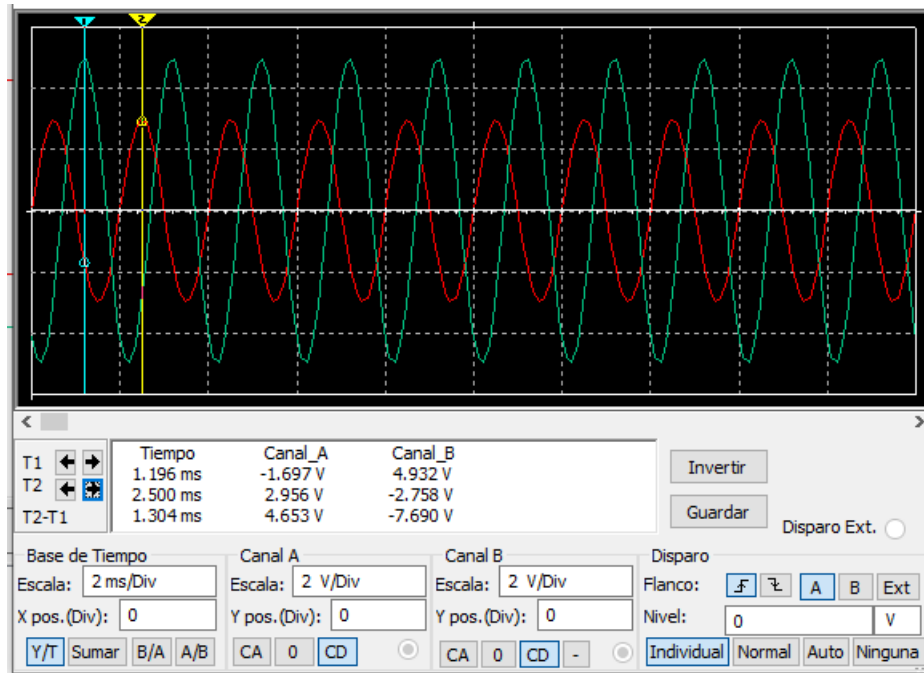


Figura 5: Pantalla del Osciloscopio para la señal de 500Hz. Verde: señal de entrada del filtro, rojo: salida del filtro

8. Análisis de los Resultados

Los valores de las resistencias tuvieron que ser ajustados a los valores comerciales más cercanos, ocasionado también cierto desajuste en los filtros.

En la figura 3 se puede observar una atenuación de poco más del 50 % de la señal, a pesar de esto, el filtro deja pasar correctamente una gran parte de la misma cumpliendo su función.

En las figuras 4 y 5 se observa un mejor comportamiento en la salida de los filtros, ya que la atenuación disminuye y las ondas pasan por los filtros de manera correcta.

9. Conclusiones y Recomendaciones

Una vez las señales analógicas han sido acondicionadas y clasificadas adecuadamente, es útil utilizar un microcontrolador para analizar y procesar las señales. Esto disminuye el espacio requerido por el circuito y simplifica algunas operaciones que son realizadas de manera digital.

Es importante verificar el correcto funcionamiento del equipo utilizado para realizar las pruebas, ya que de ello depende que el circuito no tenga comportamientos indeseados y que los resultados adquiridos sean correctos.

En la realización de los cálculos para los filtros, es importante obtener primero los capacitores, para que con estos poder calcular los valores de las resistencias. Esto se hace de esta forma, ya que es más fácil conseguir valores comerciales de resistencias que se adecúen a las necesidades que valores de capacitores.

Referencias

- [1] LM741 "<http://html.alldatasheet.com/html-pdf/53589/FAIRCHILD/LM741/813/2/LM741.html>"
- [2] Arduino Reference, Arduino.cc. <https://www.arduino.cc/reference/en/>.