

UNIVERSIDAD CATÓLICA DE EL SALVADOR



Docente:

Ma. Henry Magari Vanegas Rodríguez

Materia:

Microprocesadores y ensambladores

Alumnos:

Guardado López, José Mauricio

Hernández Alvarado, Ana Gabriela

Pineda Villegas, Joseline Alejandra

Presentación Reloj Binario

Fecha de entrega:

Lunes 07 de junio de 2021

Contenido

Antecedentes y proyectos previos	1
Detalles Técnicos de los componentes utilizados	1
Descripciones	1
Fotos.....	2
Cuadros de datos técnicos	2
Protoboard	2
LED's	3
Resistencias	3
Jumpers	3
Arduino UNO	4
Metodología	4
Análisis Financiero y Propuesta Proyecto Social-Comercial.....	6
Evidencias.....	6
Demostración-Vídeo	6
Archivos códigos fuente	6
Archivos de diseño, planos o esquemas utilizados	8
Fotos evidencia	8
Link GitHub.....	10
Conclusiones y recomendaciones	11

Antecedentes y proyectos previos

Los relojes binarios son más comunes de los que uno puede pensar, de hecho, originalmente, mostraban cada dígito decimal del tiempo sexagesimal como un valor binario, pero actualmente los relojes binarios también muestran horas, minutos y segundos como números binarios. La mayoría de los relojes binarios son digitales, aunque existen versiones analógicas. Muchos de los relojes digitales actuales realmente calculan el tiempo en binario, pero mediante un microchip interno tipo decodificador lo convierte a números del tipo decimal.

El proyecto que nosotros realizaremos demostrara conceptos tan básicos pero importante como el contar con 0 y 1 y el temporizar para ir calculando las horas minutos y segundos, muchos de estos prototipos para demostrar el concepto de los relojes se ha analizado por varias personas con anterioridad y muchos de ellos han encontrado desventajas evidentes tales como que para que el usuario lea la hora debe sumar el valor que se ve representado en cada led iluminado, después se lee de derecha a izquierda.

Ahora bien, las dos primeras columnas representan la hora, las siguientes dos leen los minutos y las dos últimas los segundos. El problema que tiene esto es que Los dígitos cero NO están iluminados por lo cual este reloj presenta la gran desventaja de que no ilumina en la oscuridad.

Al pensar en un reloj binario no debemos preocuparnos mucho por lo que significa y de cómo utilizarlo; pues el mismo se resuelve de manera sencilla para darnos la hora de manera precisa. De esta forma sencilla conocerás la forma correcta de leer un reloj binario:

- a) los relojes binarios no te muestran los números con base en 10 sino que en lugar de esto solo utiliza el número uno y el cero
- b) debido a que solo aparecen dos dígitos se utilizan las luces en lugar de dígitos

Proyectos previos

1. Reloj Con raspberry-pi 3: Reloj decimal que dar la hora mediante led de 7 segmentos
2. Reloj análogo: hecho con decodificadores y elementos no programables, su funcionamiento es el mismo dejando de lado funciones personalizables
3. Calculadora binaria Arduino: suma número binarios y los puede mostrar mediante led en resultado, solo usando el serial y una conexión a computadora

Detalles Técnicos de los componentes utilizados

Descripciones

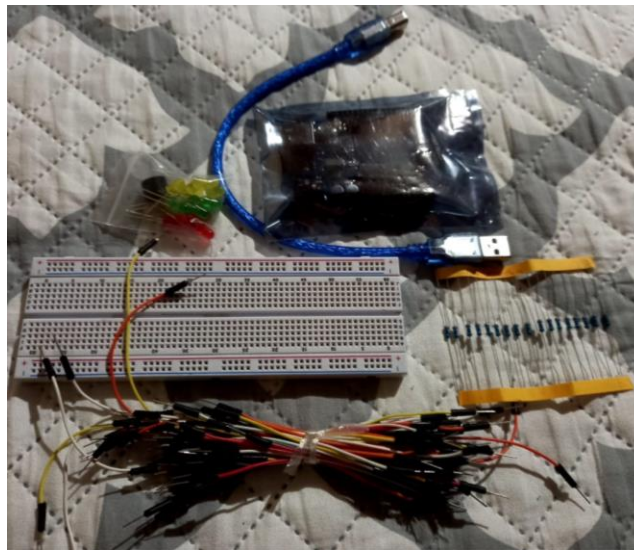
El proyecto está compuesto por los siguientes elementos:

1. Protoboard: pequeñas tablas con perforaciones en toda su área, en las cuales se colocan diversos componentes electrónicos, se distinguen por tener filas y columnas con lo que se puede saber en qué ubicación posicionar cada pieza, también cuentan con 2 rieles a los lados, los cuales se usaran como las líneas Positivas y Negativas de nuestro circuito.
2. 17 LED's (6 Leds Color Rojo, 6 Leds Color Verde, 5 Leds Color Amarillo): diodo que además de permitir el paso de la corriente solo un sentido, en el sentido en el que la

corriente pasa por el diodo, este emite luz. Cuando se conecta un diodo en el sentido que permite el paso de la corriente se dice que está polarizado directamente.

3. 17 Resistencias de 330 ohms: La resistencia es una medida de la oposición al flujo de corriente en un circuito eléctrico.
4. 17 Jumpers: Un jumper o saltador es un elemento que permite cerrar el circuito eléctrico del que forma parte dos conexiones. La función del cable macho-macho es con frecuencia usado en el tablero protoboard haciendo posible la conexión de dos elementos ingresados en dicho tablero.
5. Arduino UNO: Arduino Uno es una placa electrónica basada en el microcontrolador ATmega328. Cuenta con 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas. Además, incluye un resonador cerámico de 16 MHz, un conector USB, un conector de alimentación, una cabecera ICSP y un botón de reseteo. La placa incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

Fotos



Cuadros de datos técnicos

Protoboard

Protoboard o breadboard para la experimentación y prueba de circuitos electrónicos rápido y fácil, sin necesidad de soldadura.

Características:

- 640 puntos de contacto en el área de trabajo principal (Sin incluir buses de alimentación)
- 8 buses para alimentación independientes (Con 25 puntos de contacto c/u)
- 840 puntos de contacto en total
- Compatible con protoboards Wish
- Alambre para puentes recomendado: Calibres AWG 29 a 20 (\varnothing 0.3 a 0.8 mm aprox.)
- Adecuado para la mayoría de circuitos integrados de inserción, con espaciamiento estándar entre pines de 0.1" (2.54mm)
- Tamaño: 17.2cm x 6.8cm aprox.
- Parte inferior con adhesivo

LED's

Selection Guide

Part No.	Dice	Lens Type	Iv (mcd) [2] @ 20mA		Viewing Angle [1]
			Min.	Typ.	2θ1/2
WP7113SRD/D	Super Bright Red (GaAlAs)	RED DIFFUSED	180	250	30°

Notes:

1. θ1/2 is the angle from optical centerline where the luminous intensity is 1/2 the optical centerline value.
2. Luminous intensity/ luminous Flux: +/-15%.

Electrical / Optical Characteristics at TA=25°C

Symbol	Parameter	Device	Typ.	Max.	Units	Test Conditions
λpeak	Peak Wavelength	Super Bright Red	660		nm	If=20mA
λD [1]	Dominant Wavelength	Super Bright Red	640		nm	If=20mA
Δλ1/2	Spectral Line Half-width	Super Bright Red	20		nm	If=20mA
C	Capacitance	Super Bright Red	45		pF	Vf=0V;f=1MHz
Vf [2]	Forward Voltage	Super Bright Red	1.85	2.5	V	If=20mA
IR	Reverse Current	Super Bright Red		10	μA	VR = 5V

Notes:

1. Wavelength: +/-1nm.
2. Forward Voltage: +/-0.1V.

Resistencias

■ RELIABILITY TEST

Load Life in Moisture

Samples: 1/4 Watt, 100Ω, 1kΩ, 10kΩ, 100kΩ, 150 pieces of each, 600 pieces in total.

Conditions: 40°C in 95% RH 1% hours "ON" and 0.5 hours "OFF" cycling for 5,000 hours with DC voltage given by the following ratios:

Criterion (%)		Load Ratio P/Pn (%)	Total Testing Time T(Hrs)	Number of Failures r (pcs)	Failure Ratio		Average Lifetime (60% reliability level) (Hrs)
					λ	λCL (60%)	
ΔR/R	±5	0	2.984 x 10 ⁶	6	0.201	0.244	4.098 x 10 ⁵
		20	2.990 x 10 ⁶	4	0.134	0.176	5.682 x 10 ⁵
		60	2.997 x 10 ⁶	2	0.067	0.104	9.615 x 10 ⁵
		100	2.992 x 10 ⁶	3	0.100	0.139	7.194 x 10 ⁵
		Total	1.196 x 10 ⁷	15	0.125		7.209 x 10 ⁵
	±10	Total	1.20 x 10 ⁷	0	0.0055	<div>0.138 0.0077</div>	1.299 x 10 ⁷

Specifications are subject to change without notice. No liability or warranty implied by this information. Environmental compliance based on producer documentation.

Jumpers

Part Number Table

Description	Usage	Type	Part Number
Jumper Cables	Suitable for Gertboard headers	Female to Female	PSG-JMP150FF
	Suitable for Raspberry Pi breakout (this cables can piggy back one another)	Male to Female	PSG-JMP150MF
	These cables are ideal for development boards such as Arduino™, Beaglebone and many more.	Male to Male	PSG-JMP150MM

Arduino UNO

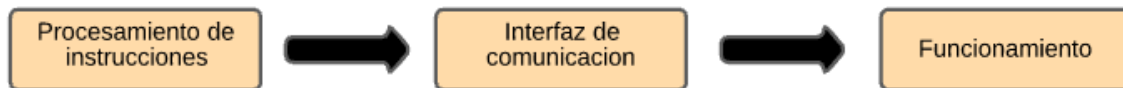
Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB of which 0.5 KB used by bootloader
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz

Metodología

A partir del circuito se pretende que muestre la hora en el sistema binario el cual para lograr este objetivo se hace uso de un Arduino y otros materiales que posteriormente se puedan ver reflejadas en unos leds de colores para la debida interpretación.

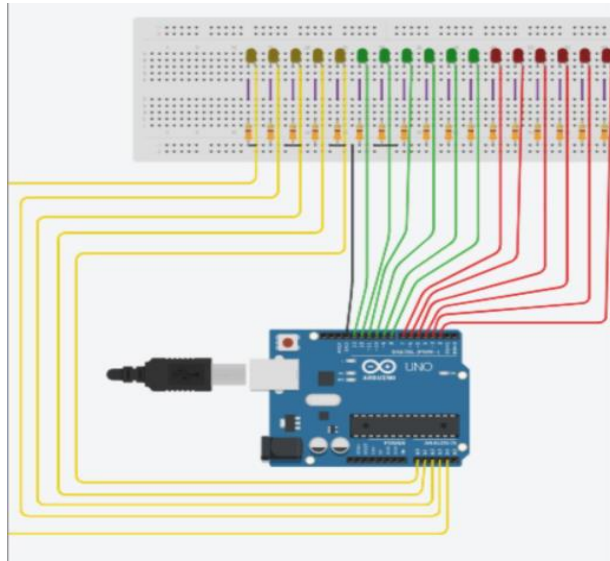
Para poder controlar las luces y hacer que se comporten como el reloj deseado, se debe codificar un programa para el Arduino, el pequeño programa simplemente actúa como un reloj, pero tiene una función que se encarga de encender los leds transformando los valores de minutos y horas (base 10) a binario (base 2) y encender y apagar los LEDs correspondientes.

Para esto se hace la conversión y revisar bit a bit el cambio de base de los números, el equivalente al número en base 10.



Se realiza la integración de los Leds y jumpers entre la Protoboard y el Arduino de manera comparativa al diagrama de conexión que se muestra a continuación para hacer el proyecto funcional mediante las instrucciones que se le darán al Arduino.

En la codificación se pretende implementar métodos relevantes para el debido funcionamiento del Reloj binario.



Sizeof ();

Esta función nos regresa el numero de Bytes en un tipo de variable o el numero bytes ocupados por un arreglo.

PinMode ();

Configura el pin especificado para que se comporte como entrada o salida. Consulte la página Pines Digitales para obtener detalles sobre la funcionalidad de los pines.

If ();

Fueron condicionales necesarios para validar información de funcionamiento.

BitRead ();

Lee un Bit de un número, que es lo que requerimos para nuestro código porque necesitamos leer un bit de un numero entero.

Delay;

Pausa el programa por la cantidad de tiempo (en milisegundos) especificado como parámetro. (Hay 1000 milisegundos en un segundo)

For ();

Para realizar recorridos y almacenamiento de información.

Análisis Financiero y Propuesta Proyecto Social-Comercial

Costos Directos			
Detalle	Cantidad	Precio	Total
Arduino Uno	1	\$11.60	\$11.60
Jumpers	1	\$3	\$3.00
Resistencias 1/4	17	\$0.19	\$3.23
Diodo LED	17	\$0.16	\$2.72
			\$20.55
Costos Indirectos			
Costo de envío	15%		
Local			
Electricidad			
Implementación			
Otros			
Total			\$3.08
Costo mano de obra			
Mano de obra	50%		\$10.28
SubTotal			\$33.91
Margen de ganancia			10%
Precio de venta			\$37.68

Nos enfocamos en el mercado educativo, nuestro nicho en específico serían los colegios, y a su vez como proyección social donaríamos equipos a las instituciones educativas con pocos recursos económicos. De igual forma venderemos nuestro producto a las tiendas de electrónica de todo el país, para todas aquellas personas que quieran tener su reloj binario, puedan adquirirlo.

Por institución educativa: \$ 188.40

Evidencias

Demostración-Vídeo

<https://drive.google.com/file/d/1ZScFHTglu40IyKzgAtFzSvRf-7t-HjLH/view>

Archivos códigos fuente

```
int ledPinsSec[] = {2, 3, 4, 5, 6, 7};
int ledPinsMin[] = {8, 9, 10, 11, 12, 13};
int ledPinsHr[] = {14, 15, 16, 17, 18, 19};
int secSize = sizeof(ledPinsSec) / sizeof(int);
int minSize = sizeof(ledPinsMin) / sizeof(int);
int hourSize = sizeof(ledPinsHr) / sizeof(int);
int seconds = 30;
int minutes = 30;
int hours = 15;
void setup()
{
    Serial.begin(9600);
    //Obtiene el número de elementos de un arreglo
    Serial.println(sizeof(ledPinsSec));
    for(int i = 0; i < secSize; i++)
    {
        pinMode(ledPinsSec[i], OUTPUT);
    }
    for(int i = 0; i < minSize; i++)
```



```

    {
        pinMode(ledPinsMin[i], OUTPUT);
    }
    for(int i = 0; i< hourSize;i++)
    {
        pinMode(ledPinsHr[i], OUTPUT);
    }
}
void loop()
{
    seconds++;
    if(seconds > 59)
    {
        seconds = 0;
        minutes++;
        if(minutes > 59)
        {
            minutes = 0;
            hours++;
            if(hours > 23)
            {
                hours = 0;
            }
        }
    }
    //Serial.println((char) seconds);
    Serial.print("Hours: ");
    Serial.print(hours);
    Serial.print(" Minutes: ");
    Serial.print(minutes);
    Serial.print(" Seconds: ");
    Serial.println(seconds);
    DisplaySeconds();
    DisplayMinutes();
    DisplayHours();
    delay(1000);
/*
    DisplaySeconds();
    DisplayMinutes();
    DisplayHours();*/
}
void DisplaySeconds()
{
    for(int i = secSize - 1; i>= 0; i--)
    {
        int currentSecond = bitRead(seconds, i);
        digitalWrite(ledPinsSec[i], currentSecond);
    }
}
void DisplayMinutes()
{
    for(int i = minSize - 1; i>= 0; i--)
    {
        int currentMinute = bitRead(minutes, i);
        digitalWrite(ledPinsMin[i], currentMinute);
    }
}

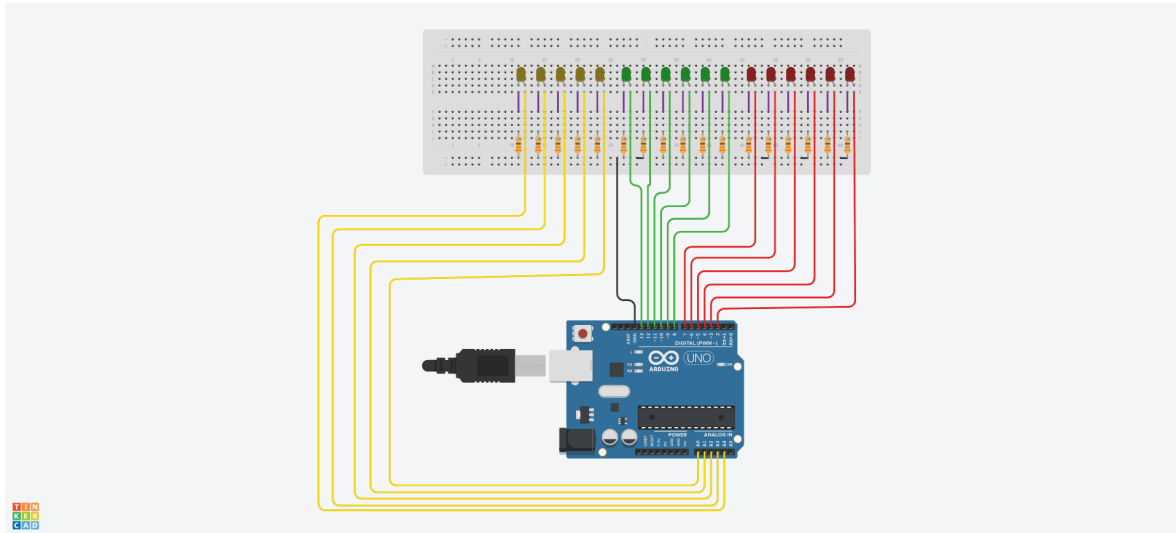
```

```

void DisplayHours()
{
  for(int i = hourSize - 1; i >= 0; i--)
  {
    int currentHour = bitRead(hours, i);
    digitalWrite(ledPinsHr[i], currentHour);
  }
}

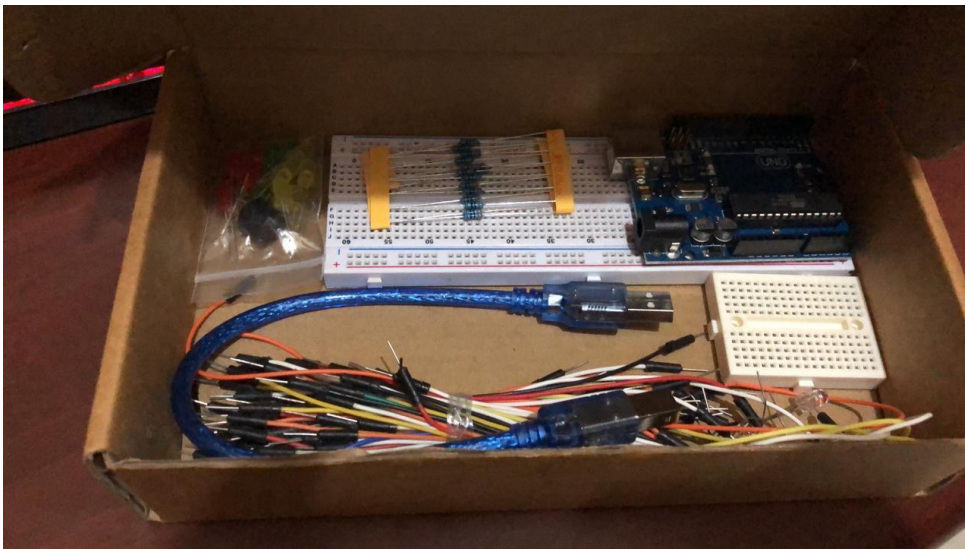
```

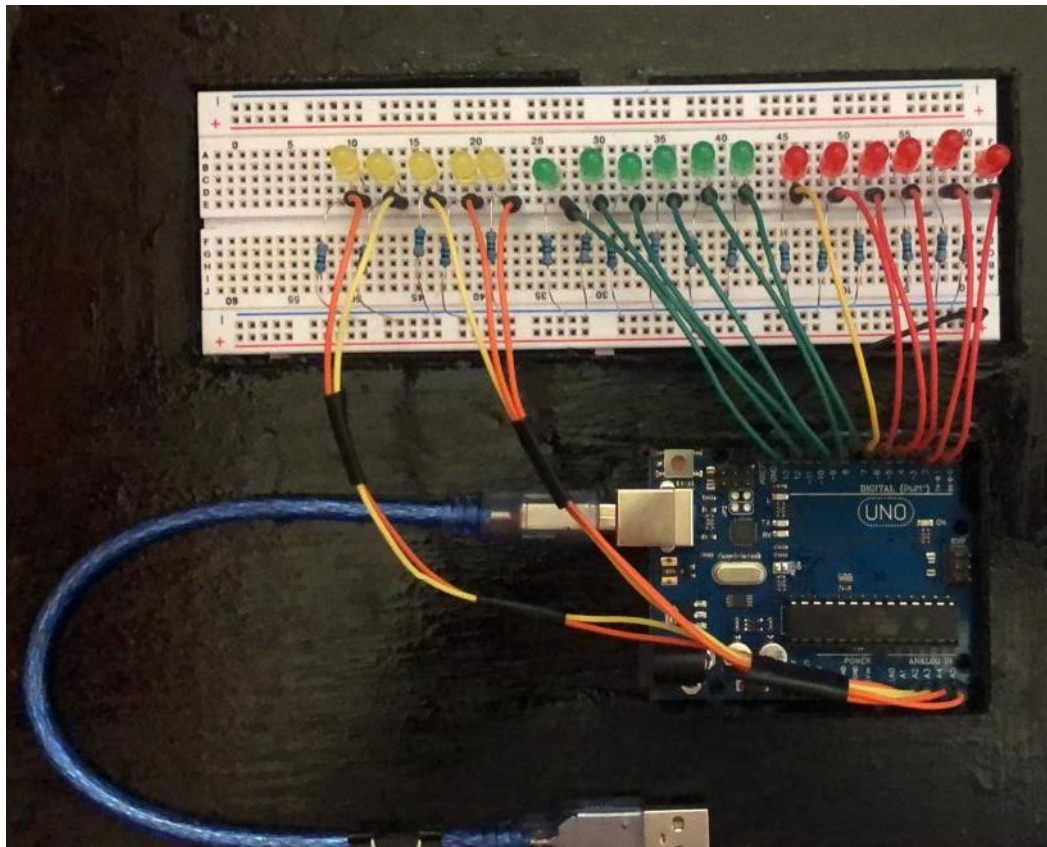
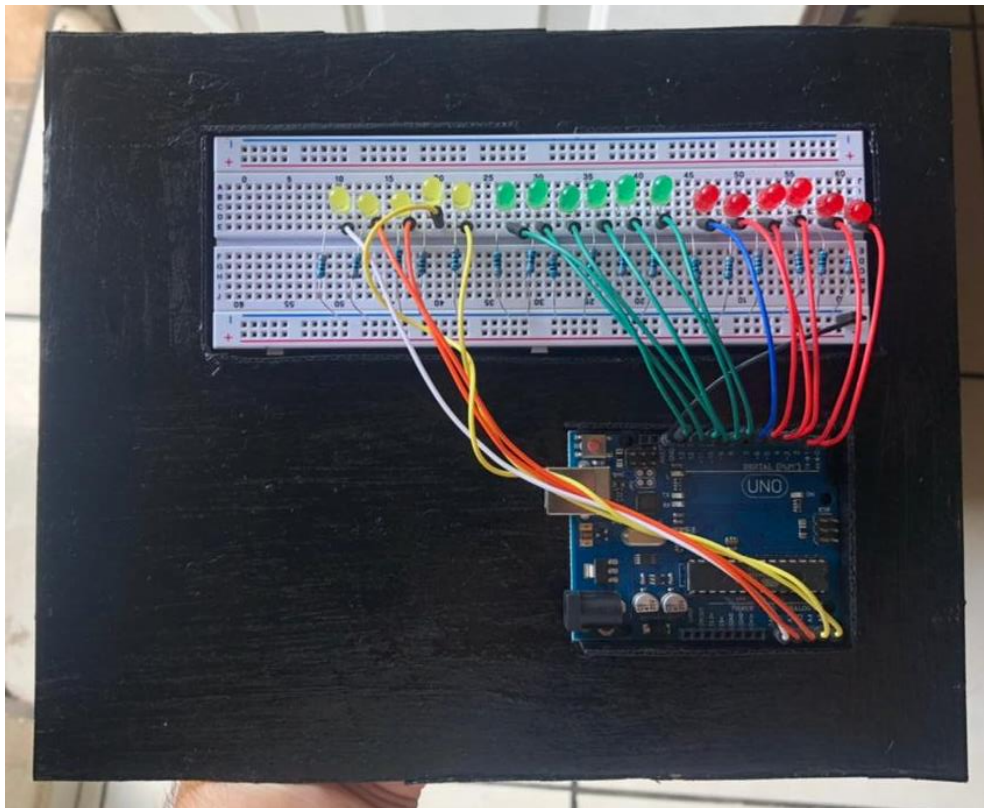
Archivos de diseño, planos o esquemas utilizados

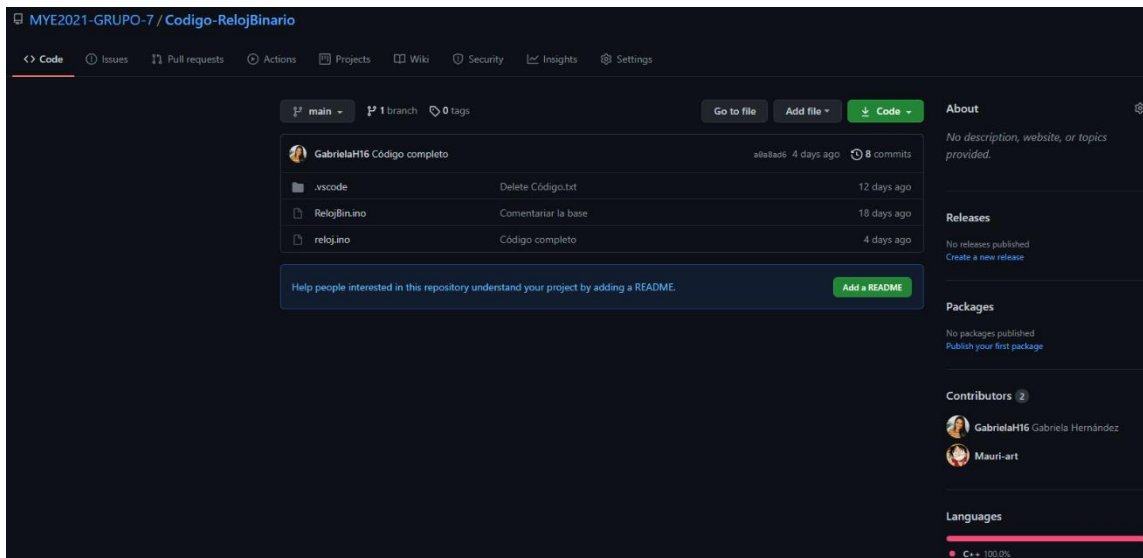
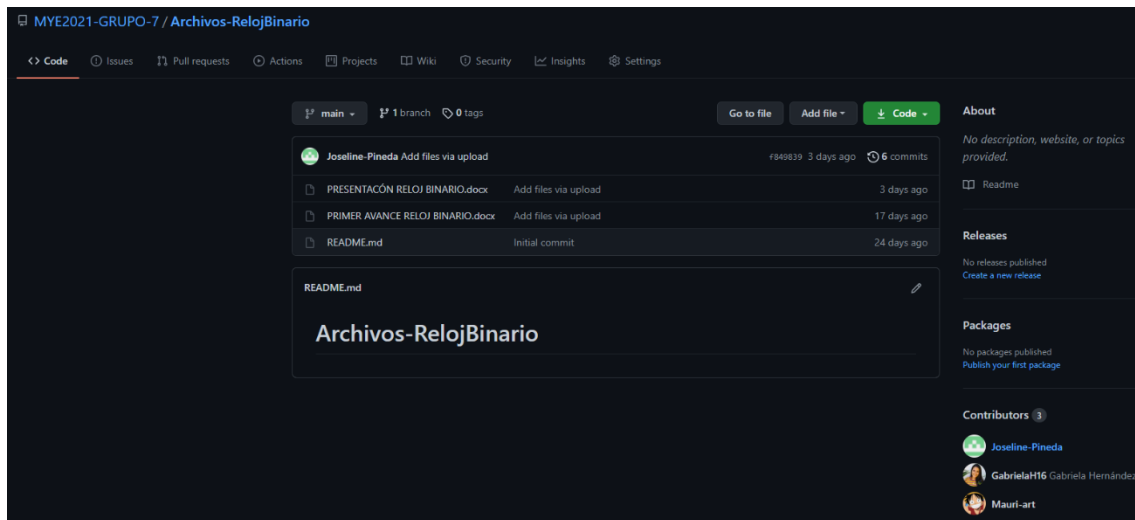


<https://www.tinkercad.com/things/i26K69ge5UK>

Fotos evidencia







Link GitHub

<https://github.com/MYE2021-GRUPO-7>

Conclusiones y recomendaciones

Como conclusión cabe destacar que se han cumplido los objetivos propuestos en el siguiente proyecto, desarrollando el reloj binario y finalizándolo de manera exitosa. En cuanto al desarrollo de este proyecto se ha logrado hacer funcionar el reloj binario brindando la hora, minutos y segundos de forma binaria, para llevar a cabo este proyecto se debe de tener cuidado a la hora de hacer las conexiones respectivas de forma física para evitar que el circuito no funcione o tenga algún desperfecto, tomando como referencia el diagrama de conexión, una recomendación sería utilizar los componentes propuestos, en el caso que no se usaran las resistencias de 300Ω se podrían ocupar otras de una resistencia diferente ya que solo afectaría la iluminación de los led sería más apagada o encendida respecto al valor de las resistencias, se tomó como referencia un código para el funcionamiento del reloj binario y para ajustarlo a la hora actual solo se le deben hacer unas modificaciones en el código.