



UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO

MATERIA: SISTEMAS EMBEBIDOS

PROFESOR: BENJAMIN PEREZ CLAVEL

INTEGRANTES DEL EQUIPO:

JOSE BRAYANT ALVARES MORALES

ULISES BECERRIL VALDÉS

YANELA MICHELLE TRINIDAD CALIXTO

FECHA DE ENTREGA: 09 DE ABRIL DEL 2024

GRUPO E1

PRÁCTICA		
RUBRO	VALOR	CALIF.
Portada	2.5	
Introducción	10	
Desarrollo	65	
Conclusiones	20	
Referencias / citas	2.5	
TOTAL/10		
Firma del docente		



Universidad Autónoma del Estado de México
UAEM



INTRODUCCION

En el ámbito de la ingeniería de sistemas digitales, el diseño y la implementación de controladores desempeñan un papel fundamental en la gestión y la regulación del flujo de información entre distintos componentes de un sistema. Los controladores, dispositivos electrónicos sofisticados, son hábiles en la recepción de señales de entrada, el procesamiento de estas señales de acuerdo con un conjunto predefinido de reglas, y la generación de señales de salida correspondientes para dirigir y controlar otros dispositivos o procesos en el sistema.

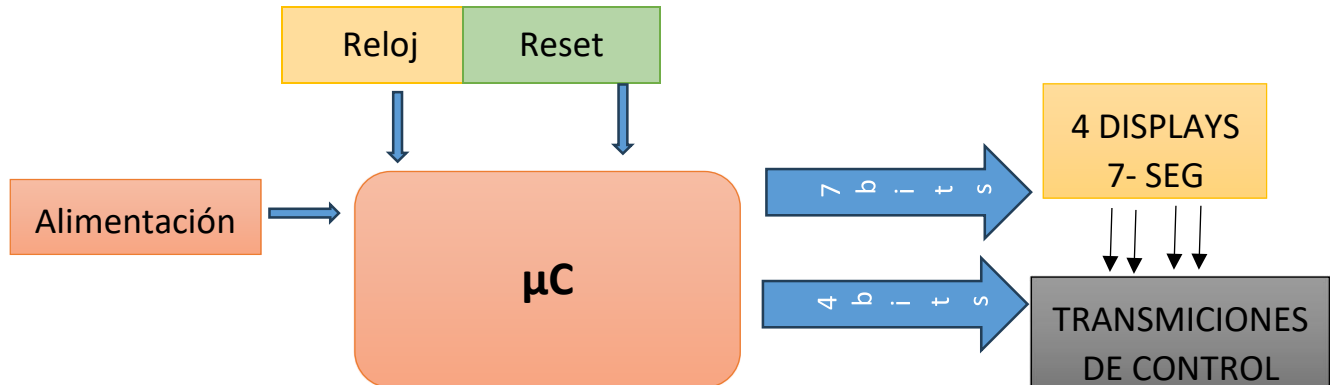
Dentro de esta esfera tecnológica, se destaca el PIC (Programmable Integrated Circuit), un tipo de circuito integrado programable que concentra todos los elementos necesarios para ejecutar y supervisar una tarea específica, otorgándole la categoría de microcontrolador. Los PIC pertenecen a una distinguida familia de microcontroladores de tipo RISC, manufacturados por Microchip Technology Inc. Estos dispositivos, herederos del PIC1650 originalmente desarrollado por la división de microelectrónica de General Instrument, han evolucionado considerablemente desde su concepción inicial. Si bien el nombre actual no constituye un acrónimo, originalmente se refería a "PICmicro", aunque en la práctica se le suele identificar como "Peripheral Interface Controller" (controlador de interfaz periférico), dada su funcionalidad inherente en el manejo y la interacción con diversos periféricos y dispositivos externos.

Tomando como piedra angular este concepto de control y regulación mediante microcontroladores, se ha concebido un circuito específico que aprovecha las capacidades del microcontrolador PIC18F46K22. Este circuito ha sido diseñado siguiendo un esquema meticulosamente planificado, el cual incluye especificaciones detalladas sobre los materiales a utilizar, con el fin de asegurar un funcionamiento óptimo y confiable del sistema.





En base a este concepto clave , se construyo un circuito utilizando el microcontrolador PIC18F46k22 utilizando el siguiente esquema con especificaciones de material a utilizar :



En el contexto de este proyecto, se ha llevado a cabo el diseño, programación y simulación de un sistema que muestra un reloj digital, exhibiendo tanto los minutos como los segundos transcurridos mediante cuatro displays de 7 segmentos con cátodo común. Para lograr este cometido de manera eficiente, se ha elaborado un diagrama esquemático que incorpora una serie de elementos esenciales para el funcionamiento del sistema.

En primer lugar, se ha considerado la alimentación eléctrica del sistema, asegurando un suministro constante y adecuado de energía para todos los componentes involucrados. Este aspecto es crucial para garantizar el correcto funcionamiento del sistema en todo momento.

Asimismo, se ha integrado un componente de reloj en el diagrama esquemático. Este módulo de reloj proporciona la sincronización necesaria para el funcionamiento preciso del sistema, asegurando que los intervalos de tiempo se gestionen de manera adecuada y confiable.

Otro elemento clave en el diseño es el mecanismo de reset, el cual permite restablecer el estado del sistema a un estado inicial predeterminado. Esta funcionalidad es fundamental





para iniciar o reiniciar el conteo del tiempo en el reloj, garantizando la precisión y la coherencia en la visualización de la hora.

En el corazón del sistema se encuentra el controlador, que desempeña un papel crucial en la coordinación y gestión de todas las operaciones. Este controlador, posiblemente un microcontrolador PIC18F46k22, se encarga de procesar la información recibida, calcular los valores correspondientes para los minutos y segundos transcurridos, y generar las señales de salida adecuadas para los displays de 7 segmentos.

Para la conexión de los displays de 7 segmentos al microcontrolador, se ha previsto una entrada de 7 bits que permitirá transmitir la información necesaria para representar los dígitos en cada display. Esta entrada de 7 bits se encargará de activar los segmentos específicos requeridos para cada dígito, permitiendo una visualización clara y precisa del tiempo transcurrido.

Además, se ha incluido una entrada de 4 bits para la transmisión de control. Esta entrada de 4 bits se utiliza para indicar el display activo en un momento dado, permitiendo alternar entre los cuatro displays de 7 segmentos para mostrar tanto los minutos como los segundos transcurridos. En el contexto de esta práctica, el controlador se encargará de interpretar las señales provenientes de dos switches de entrada y generar una secuencia específica de encendido y apagado de LEDs de 8 bits en función de la configuración de los switches.

CONCEPTOS CLAVE:

Puerto Microcontrolador PIC18F46k22: El microcontrolador seleccionado para el proyecto, que actúa como el cerebro del sistema, encargado de procesar la información y controlar la visualización en los displays de 7 segmentos.





Displays de 7 segmentos de cátodo común: Los dispositivos de salida del sistema, utilizados para mostrar los dígitos del reloj. Cada display está compuesto por siete segmentos que pueden ser activados individualmente para representar números del 0 al 9 y algunas letras.

Diagrama esquemático: La representación gráfica del diseño del sistema, que muestra la interconexión de los componentes clave como la alimentación, el reloj, el reset, el controlador y las entradas para los displays de 7 segmentos.

Alimentación eléctrica: El suministro constante de energía eléctrica necesaria para el funcionamiento adecuado de todos los componentes del sistema.

Reloj: El componente encargado de proporcionar la sincronización necesaria para el funcionamiento del reloj digital, asegurando que los intervalos de tiempo se gestionen de manera precisa y confiable.

Reset: La funcionalidad que permite restablecer el estado del sistema a un estado inicial predeterminado, lo que es esencial para iniciar o reiniciar el conteo del tiempo en el reloj.

Entrada de 7 bits para los displays de 7 segmentos: La conexión que permite transmitir la información necesaria desde el microcontrolador a los displays de 7 segmentos, activando los segmentos específicos requeridos para representar los dígitos del tiempo transcurrido.

Entrada de 4 bits para la transmisión de control: La conexión que permite alternar entre los cuatro displays de 7 segmentos, indicando cuál está activo en un momento dado para mostrar tanto los minutos como los segundos transcurridos.





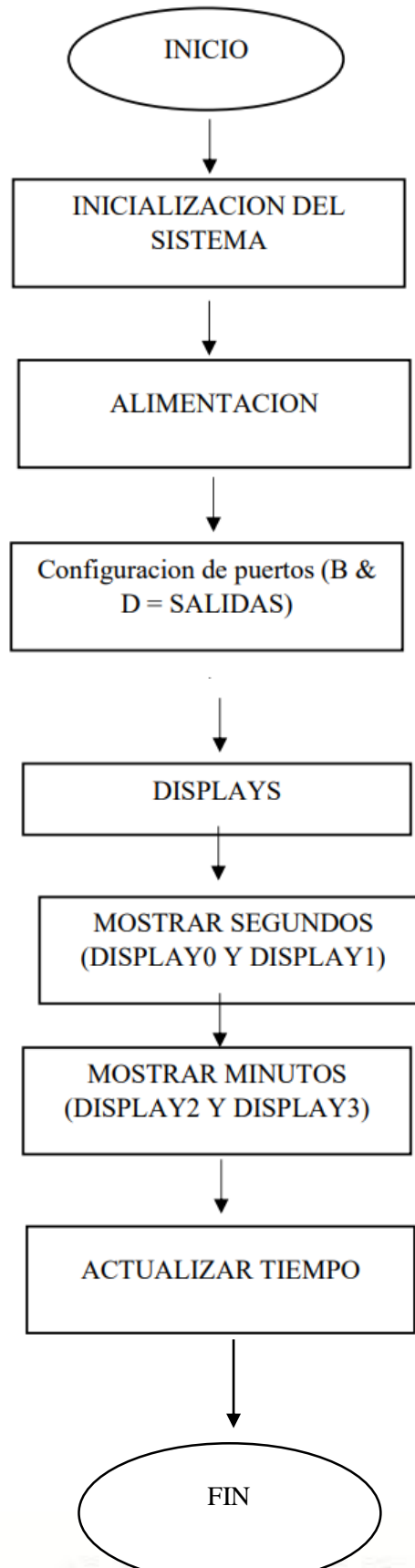
MATERIALES :

- Microcontrolador PIC18F46k22: El corazón del sistema, que actúa como el cerebro para procesar la información y controlar la visualización en los displays de 7 segmentos.
- Displays de 7 segmentos de cátodo común: Los dispositivos de salida del sistema, utilizados para mostrar los dígitos del reloj. Cada display está compuesto por siete segmentos que pueden ser activados individualmente para representar números del 0 al 9 y algunas letras.
- Cristal de reloj (oscilador): Componente utilizado para proporcionar una señal de frecuencia estable al microcontrolador, asegurando la precisión en la medición del tiempo.
- Resistencias: Componentes electrónicos que limitan el flujo de corriente en un circuito. Se utilizan para limitar la corriente a través de los segmentos de los displays de 7 segmentos y para la protección de los componentes.
- Condensadores: Componentes utilizados para almacenar energía en un campo eléctrico. Se utilizan en el circuito para estabilizar la alimentación y filtrar las fluctuaciones de voltaje.
- Interruptores (switches): Dispositivos que permiten abrir o cerrar un circuito eléctrico. Se utilizan para activar funciones como el reset o para controlar el modo de operación del reloj.
- Protoboard o placa de pruebas: Placa de circuito impreso con una disposición de contactos que permite la interconexión de componentes electrónicos sin necesidad de soldadura. Se utiliza para montar y probar el circuito antes de su implementación definitiva.
- Cables de conexión: Utilizados para conectar los diferentes componentes entre sí y con el microcontrolador, permitiendo la transmisión de señales eléctricas y la interacción entre ellos.
- Botón RESET: Este botón proporciona una forma de reiniciar el circuito o el microcontrolador en caso de ser necesario. Puedes conectarlo a una entrada específica del microcontrolador para que pueda detectar cuándo se presiona el botón y realizar la acción de reinicio correspondiente.





DIAGRAMA DE FLUJO 1



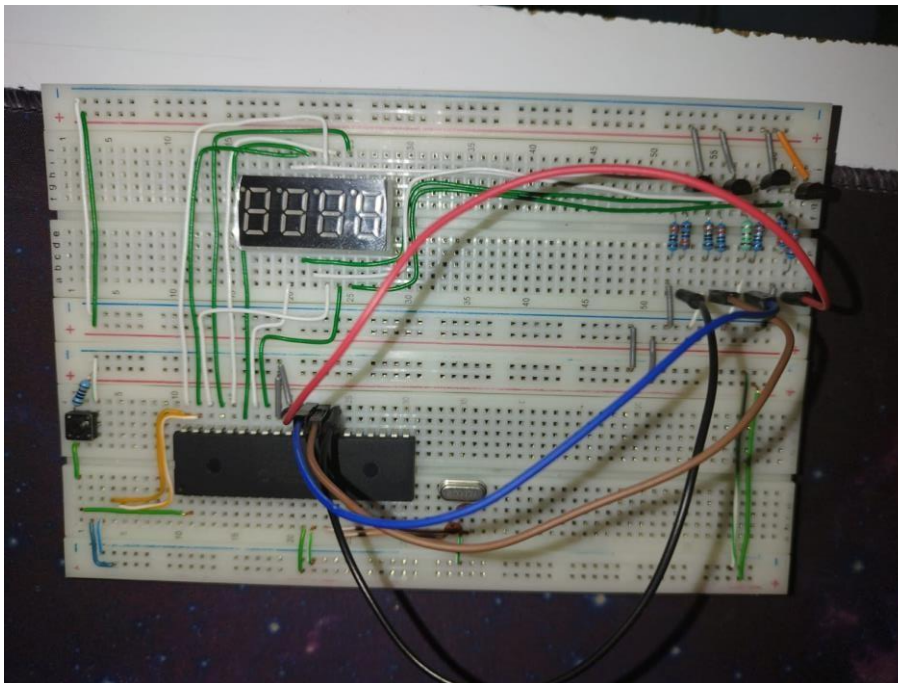
DESARROLLO

1. Configuración del Entorno de Desarrollo:

Se procedió a abrir el entorno de desarrollo MPLAB X IDE y se creó un nuevo proyecto específico para el microcontrolador PIC18F46k22. Se verificó que el compilador XC8 estuviera correctamente configurado para el proyecto en cuestión.

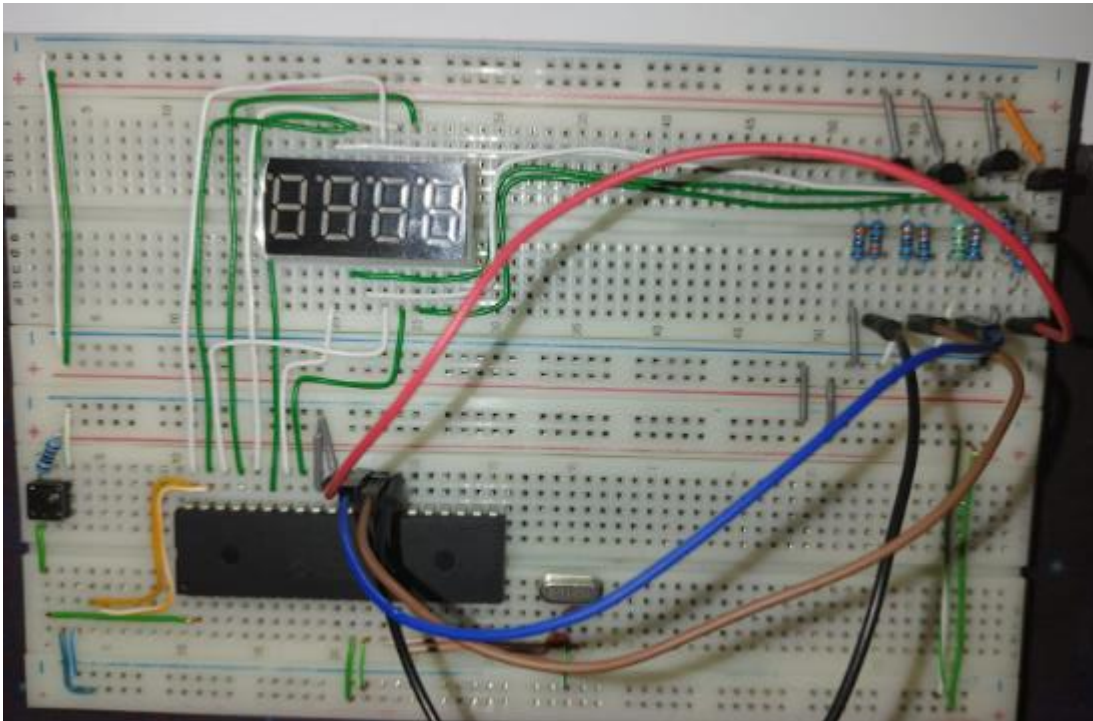
2. Diseño del Circuito:

Se colocó el microcontrolador PIC18F46k22 en la protoboard siguiendo las especificaciones del datasheet. Se conectaron los pines de alimentación (VDD y VSS) y se establecieron las conexiones necesarias para la programación del microcontrolador. Además, se insertaron los cuatro displays de 7 segmentos de cátodo común en la protoboard y se realizaron las conexiones correspondientes entre los pines de los displays y los pines de salida del microcontrolador. Se utilizaron resistencias para limitar la corriente y garantizar un funcionamiento adecuado.



3. Implementación de los Displays de 7 Segmentos:

Los displays de 7 segmentos se colocaron de manera adecuada en la protoboard y se conectaron según las especificaciones del datasheet. Se verificó que todas las conexiones estuvieran correctamente establecidas.



4. Programación del Microcontrolador:

Se escribió el código en lenguaje C utilizando MPLAB X IDE. Se implementaron las funciones necesarias para controlar los displays de 7 segmentos y para calcular los minutos y segundos transcurridos. Se utilizó el compilador XC8 para compilar el código y generar el archivo de código objeto (.hex).



```
C reloj.c X
C: > Users > joseb > Downloads > C reloj.c
1 /*
2  * File:   reloj.c
3  * Author: ubecerril
4  *
5  * Created on 1 de abril de 2024, 10:02 AM
6  */
7 // CONFIG1H
8 #pragma config FOSC = HSHP           // Oscillator Selection bits (HS oscillator (HS))
9
10 // CONFIG2L
11 #pragma config PWRTEN = ON           // Power-up Timer Enable bit (PWRT enabled)
12
13 // CONFIG2H
14 #pragma config WDTEEN = OFF          // Watchdog Timer Enable bit (WDT disabled (control is placed on the SWDTEN bit))
15
16 // CONFIG3H
17 #pragma config MCLRE = INTMCLR       // MCLR Pin Enable bit (RE3 input pin enabled; MCLR disabled)
18
19 #include <xc.h>
20 #define _XTAL_FREQ 8000000
21 //VARIABLES PARA LOS NUMEROS
22 unsigned char CERO = 0x3F;
23 unsigned char UNO = 0x06;
24 unsigned char DOS = 0x5B;
25 unsigned char TRES = 0x4F;
26 unsigned char CUATRO = 0x66;
27 unsigned char CINCO = 0x6D;
28 unsigned char SEIS = 0x7D;
```

```
C reloj.c X
C: > Users > joseb > Downloads > C reloj.c
28 unsigned char SEIS = 0x7D;
29 unsigned char SIETE = 0x07;
30 unsigned char OCHO = 0x7F;
31 unsigned char NUEVE = 0x6F;
32
33 unsigned char CEROP = 0xBF;
34 unsigned char UNOP = 0x86;
35 unsigned char DOSP = 0x0B;
36 unsigned char TRESP = 0xCF;
37 unsigned char CUATROP = 0xE6;
38 unsigned char CINCO = 0xED;
39 unsigned char SEISP = 0xFD;
40 unsigned char SIETEP = 0x87;
41 unsigned char OCHOP = 0xFF;
42 unsigned char NUEVEP = 0xEF;
43
44 //VARIABLES PARA LOS DISPLAY
45 unsigned char D3 = 0x7F;
46 unsigned char D2 = 0xBF;
47 unsigned char D1 = 0x0F;
48 unsigned char D0 = 0xEF;
49
50 //FUNCIONES PROTOTIPO
51 void display0(int);
52 void display1(int);
53 void display2(int);
54 void display3(int);

C reloj.c X
C: > Users > joseb > Downloads > C reloj.c
55
56 //VARIABLES MINUTOS Y SEGUNDOS
57 static int minutos, segundos, vueltas;
58
59 void main(void) {
60     TRISB = 0x00; //PUERTO B EN MODO SALIDA
61     TRISD = 0x0F; //PUERTO D EN MODO SALIDA
62
63     while (1)
64     {
65         display0(segundos);
66
67         display1(segundos);
68
69         display2(minutos);
70
71         display3(minutos);
72
73         vueltas++;
74         if(vueltas == 48)
75         {
76             segundos ++;
77             vueltas = 0;
78         }
79         if (segundos == 60)
80         {
81             minutos++;
82             segundos = 0;
```





```
C reloj.c
C: > Users > joseb > Downloads > C reloj.c
59 void main(void) {
64 {
80 {
82 segundos =0;
83 vueltas=0;
84 }
85 if(minutos==60)
86 {
87 vueltas=0;
88 segundos=0;
89 minutos=0;
90 }
91 }
92 return;
93 }
94 }
95 }
96 void display0 (int s)
97 {
98 s /= 10;
99 LATD = D0;
100 switch (s)
101 {
102 case 0:
103 LATB=CERO;
104 break;
105
106 case 1:
107
108 case 1:
109 LATB=UNO;
110 break;
111
112 case 2:
113 LATB=DOS;
114 break;
115
116 case 3:
117 LATB=TRES;
118 break;
119
120 case 4:
121 LATB=CUATRO;
122 break;
123
124 case 5:
125 LATB=CINCO;
126 break;
127
128 case 6:
129 LATB=SEIS;
130 break;
131
132 case 7:
133 LATB=SIETE;
134 break;
135
136 case 8:
137 LATB=OCHO;
138 break;
139
140 case 9:
141 LATB=NUEVE;
142 break;
143 }
144 __delay_ms(5);
145 }
146 void display1 (int s)
147 {
148 s /= 10;
149 LATD = D1;
150 switch (s)
151 {
152 case 0:
153 LATB=CERO;
154 break;
155
156 case 1:
157
158 case 1:
159 LATB=UNO;
160 break;
161
162 case 2:
163 LATB=DOS;
164 break;
165
166 case 3:
167 LATB=TRES;
168 break;
169
170 case 4:
171 LATB=CUATRO;
172 break;
173
174 case 5:
175 LATB=CINCO;
176 break;
177
178 case 6:
179 LATB=SEIS;
180 break;
181
182 case 7:
183 LATB=SIETE;
184 break;
185
186 case 8:
187 LATB=OCHO;
188 break;
189
190 case 9:
191 LATB=NUEVE;
192 break;
193 }
194 __delay_ms(5);
195 }
196 }
```





```
C reloj.c
C: > Users > joseb > Downloads > C reloj.c
146 {
180 }
181 void display2 (int m)
182 {
183     m %= 10;
184     LATD = D2;
185     switch (m)
186     {
187         case 0:
188             LATB=CEROP;
189             break;
190
191         case 1:
192             LATB=UNOP;
193             break;
194
195         case 2:
196             LATB=DOSP;
197             break;
198
199         case 3:
200             LATB=TRESP;
201             break;
202
203         case 4:
204             LATB=CUATROP;
205             break;
206
207         case 5:
208             LATB=CINCOP;
209             break;
210
211         case 6:
212             LATB=SEISP;
213             break;
214
215         case 7:
216             LATB=SIETEP;
217             break;
218
219         case 8:
220             LATB=OCHOP;
221             break;
222
223         case 9:
224             LATB=NUEVEP;
225             break;
226     }
227     __delay_ms(5);
228 }
229 void display3 (int m)
230 {
231 }
```

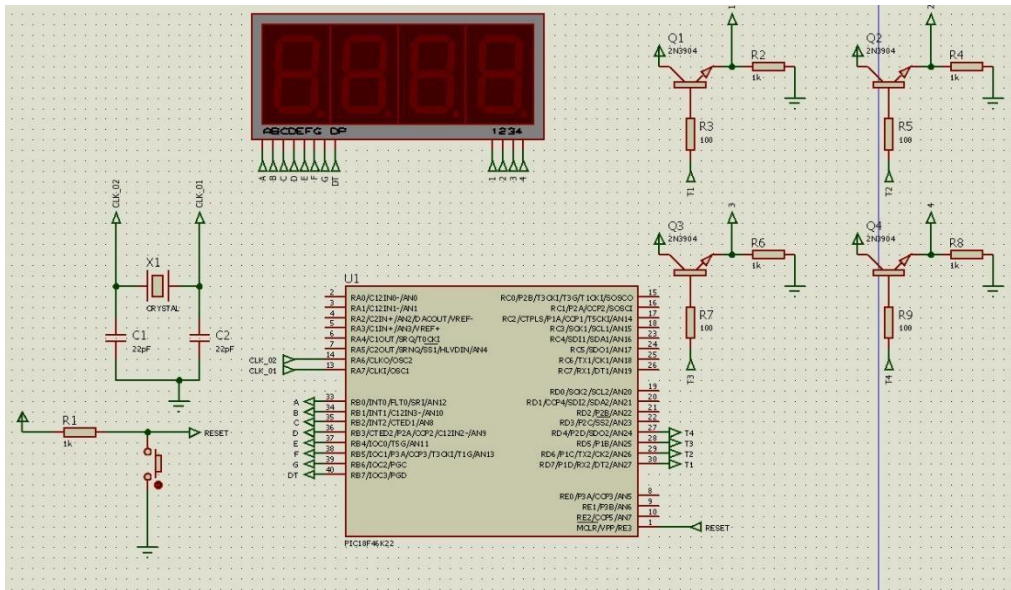
```
C reloj.c
C: > Users > joseb > Downloads > C reloj.c
231 {
232     m /= 10;
233     LATD = D3;
234     switch (m)
235     {
236         case 0:
237             LATB=CERO;
238             break;
239
240         case 1:
241             LATB=UNO;
242             break;
243
244         case 2:
245             LATB=DOS;
246             break;
247
248         case 3:
249             LATB=TRES;
250             break;
251
252         case 4:
253             LATB=CUATRO;
254             break;
255
256         case 5:
257             LATB=CINCO;
258             break;
259
260         case 6:
261             LATB=SEIS;
262             break;
263
264         case 7:
265             LATB=SIETE;
266             break;
267
268         case 8:
269             LATB=OCHO;
270             break;
271
272         case 9:
273             LATB=NOVE;
274             break;
275     }
276     __delay_ms(5);
277 }
```





5. Simulación del Circuito en Proteus:

Se abrió el software Proteus y se diseñó el circuito, incluyendo el microcontrolador PIC18F46k22 y los displays de 7 segmentos. Se realizaron las conexiones según el diseño previamente realizado en la protoboard. Se cargó el archivo de código objeto generado por MPLAB X IDE en el microcontrolador simulado y se ejecutó la simulación para verificar el funcionamiento del sistema de reloj digital.

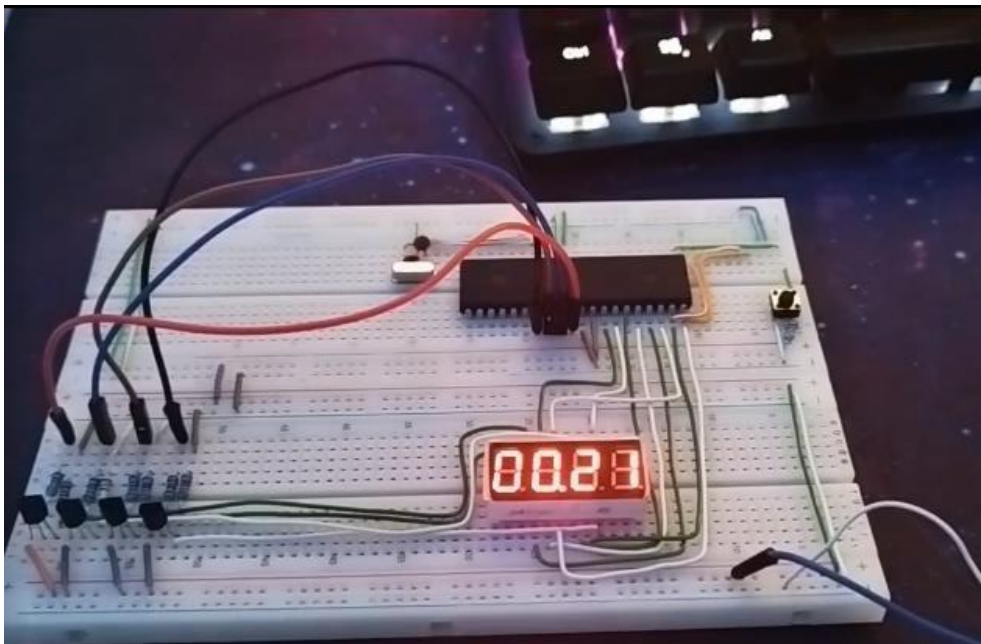
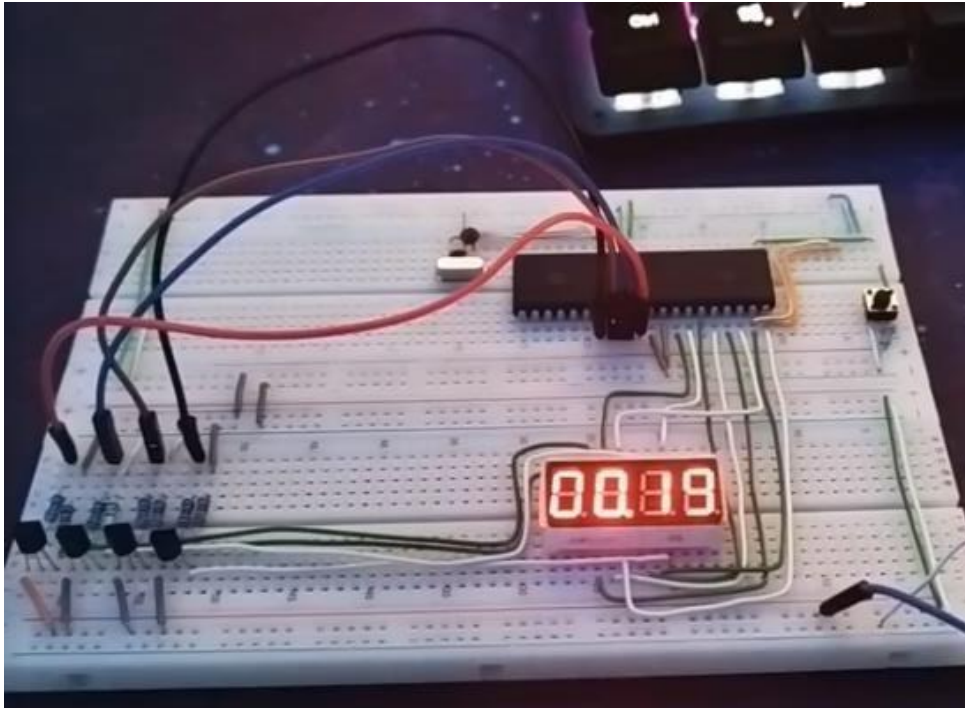


6. Resultados:

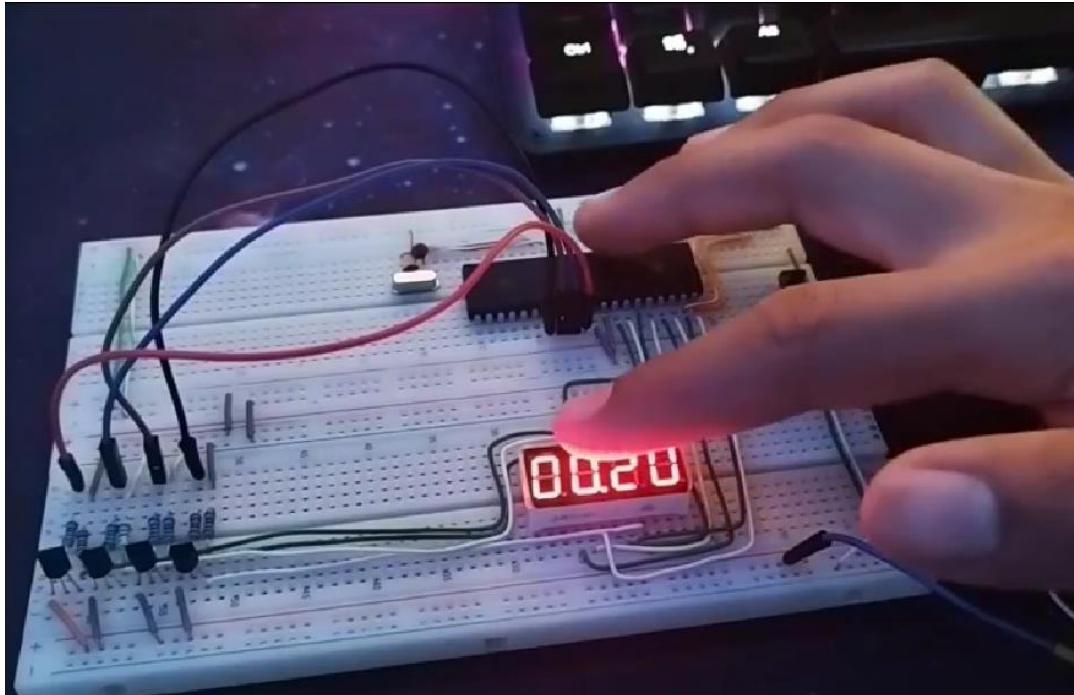
La simulación en Proteus mostró que el sistema de reloj digital funcionaba correctamente, mostrando los minutos y segundos transcurridos de manera precisa en los displays de 7 segmentos. Todas las funcionalidades programadas fueron exitosas, y el circuito respondió como se esperaba durante la simulación.



Universidad Autónoma del Estado de México
UAEM



Universidad Autónoma del Estado de México
UAEM



Conclusiones:

La realización de la práctica permitió al equipo verificar la viabilidad y eficacia del diseño del sistema de reloj digital utilizando el microcontrolador PIC18F46K22 y displays de 7 segmentos de cátodo común. Se confirmó que la programación del microcontrolador y la configuración del circuito fueron realizadas de manera adecuada, lo que resultó en un sistema funcional y preciso. Este ejercicio proporcionó una invaluable experiencia en el desarrollo de proyectos de electrónica digital, fortaleciendo el conocimiento teórico y práctico del equipo en el área de materiales mencionados.



REFERENCIAS

Introducción a la programación de microcontroladores PIC. (s/f). TECmikro

Ecuador. Recuperado el 9 de marzo de 2024, de

<https://tecmikro.com/content/8-programacion-microcontroladores-pic>

Hubor. (s/f). *¿Qué es proteus?* Hubor-proteus.com. Recuperado el 13 de febrero de

2024, de [https://www.hubor-proteus.com/proteus-pcb/proteus-](https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2proteus.html)

[pcb/2proteus.html](https://www.hubor-proteus.com/proteus-pcb/proteus-pcb/2proteus.html)

Abuelo, E. (2020, febrero 21). *Parpadeo de un LED.* Reset.

<https://www.reset.etsii.upm.es/servicios/tutoriales/tutorial-1-blink/>



Universidad Autónoma del Estado de México
UAEM