

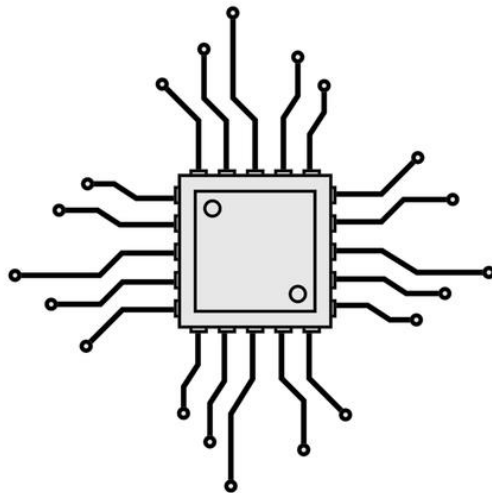


UANL
UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN



FIME
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA



**Producto
aprendizaje**

Integrador de

CONTROLADORES Y
MICROCONTROLADORES PROGRAMABLES

Nombre	Matricula	Carrera	# Lista
Lizeth Saraí Tapia Martínez	1734046	ITS	40
Santiago Valenzuela Federico	1943524	ITS	41

DIA: JUEVES

HORA: M4

INSTRUCTOR: JESUS DANIEL GARZA CAMARENA

PERIODO: ENERO – JUNIO 2022

Cd. Universitaria, San Nicolás de los Garza, Nuevo León, a 13 de mayo de 2022.

Objetivo

Aplicar el conocimiento adquirido durante el semestre y aplicarlo en un proyecto relacionado a la asignatura

Instrucciones

Diseñe, efectúe la simulación y construya un prototipo de un sistema electrónico que muestre varias funciones mediante dos display de 7 segmentos, el funcionamiento debe de estar basado en una máquina de estados cada vez que se de clic a un botón estos cambiarán de modalidad.

El sistema cuenta con 4 modos que deben de trabajar bajo las siguientes condiciones:

- Cuando se conecte el sistema este debe de mostrar un contador automático ascendente de 0 a 59, su incremento debe de ser por TIMER y no por delay.
- Si se presiona el botón "modo" el sistema deberá de cambiar de modalidad y el contador 0 a 30 deberá de aumentar de forma manual desde un botón "ascendente".
- Si se presiona de nuevo el botón "modo" el sistema deberá de cambiar de modalidad y el contador 30 a 0 deberá de descender de forma manual desde un botón "descendente".
- Si se presiona de nuevo el botón "modo" el sistema deberá de cambiar de modalidad y los display deberán de mostrar la la lectura de un potenciómetro ADC (0V \rightarrow 00, 5V \rightarrow 99)
- Se pueden utilizar una técnica de barrido para mostrar los displays, decodificadores o pueden estar conectados directamente a los puertos del MCU.

Introducción

En este proyecto usaremos un chip del cual se nos introdujo en clase. Este es el Decodificador de BCD, Binary Coded Decimal, el cual nos permite hacer entrada de números binarios para convertirlo a decimal. Lo cual a nuestro beneficio nos permite mostrar números del 0 al 9 en nuestro display, con solamente 4 pines de salida del microcontrolador ATMEGA328P al BCD y una conexión de pin a pin hacia el display de 7 Segmentos.

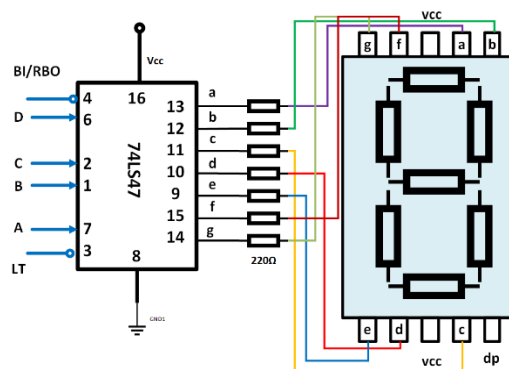
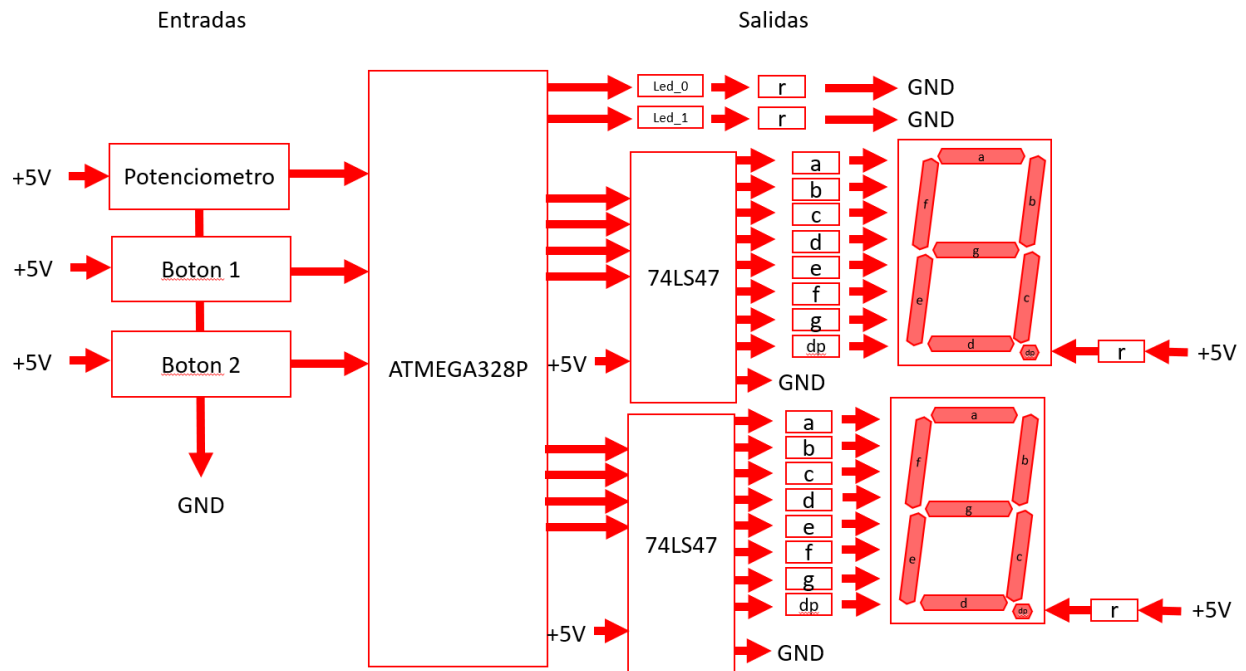

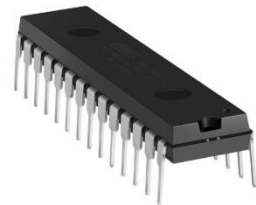



Diagrama de bloques



Lista de materiales

Nombre	Cantidad	Total	
Potenciómetro	1	\$9	
Atmega	1	\$119	
Bcd 7447	2	\$70 (\$35 c/u)	

Display de 7seg	2	\$24 (\$12 c/u)	
Botón	2	\$14 (\$7c/u)	
Resistencias	6	\$6 (\$1 c/u)	
BlockT de alimentacion usb	1	\$49	

Marco teórico

Para la implementación de este sistema electrónico se hizo de varios materiales, de los cuales explicare a continuación:

- Un potenciómetro es un instrumento utilizado generalmente para poder determinar la diferencia de potencial eléctrico entre dos terminales eléctricas. La diferencia de potencial es lo que comúnmente se conoce como voltaje.
- Atmega es un circuito integrado que es el componente principal de una aplicación embebida. Funciona como una mini PC. Su función es la de automatizar procesos y procesar información.
- Bcd 7447 es un decodificador es un circuito integrado que convierte el código binario de entrada en formato BCD a niveles lógicos que permiten activar display de 7 segmentos de ánodo común en donde la posición de cada barra forma el número decodificado
- . • Los displays de 7 segmentos son dispositivos electrónicos de visualización utilizados como una forma fácil de representar numerales decimales y una alternativa a los displays de matriz de puntos más complejos.
- El botón de un dispositivo electrónico funciona por lo general como un interruptor eléctrico, es decir en su interior tiene dos contactos, al ser pulsado uno, se activará la función inversa de la que en ese momento este realizando.
- Una resistencia sirve para limitar la corriente que circula por una rama de un circuito eléctrico. Dado que la placa base gestiona al resto de componentes, es esencial contar con específicas resistencias de alta calidad en circuitos clave.
- BlockT de alimentación usb es el cable que da corriente al circuito.

Imagen del Diagrama Esquemático

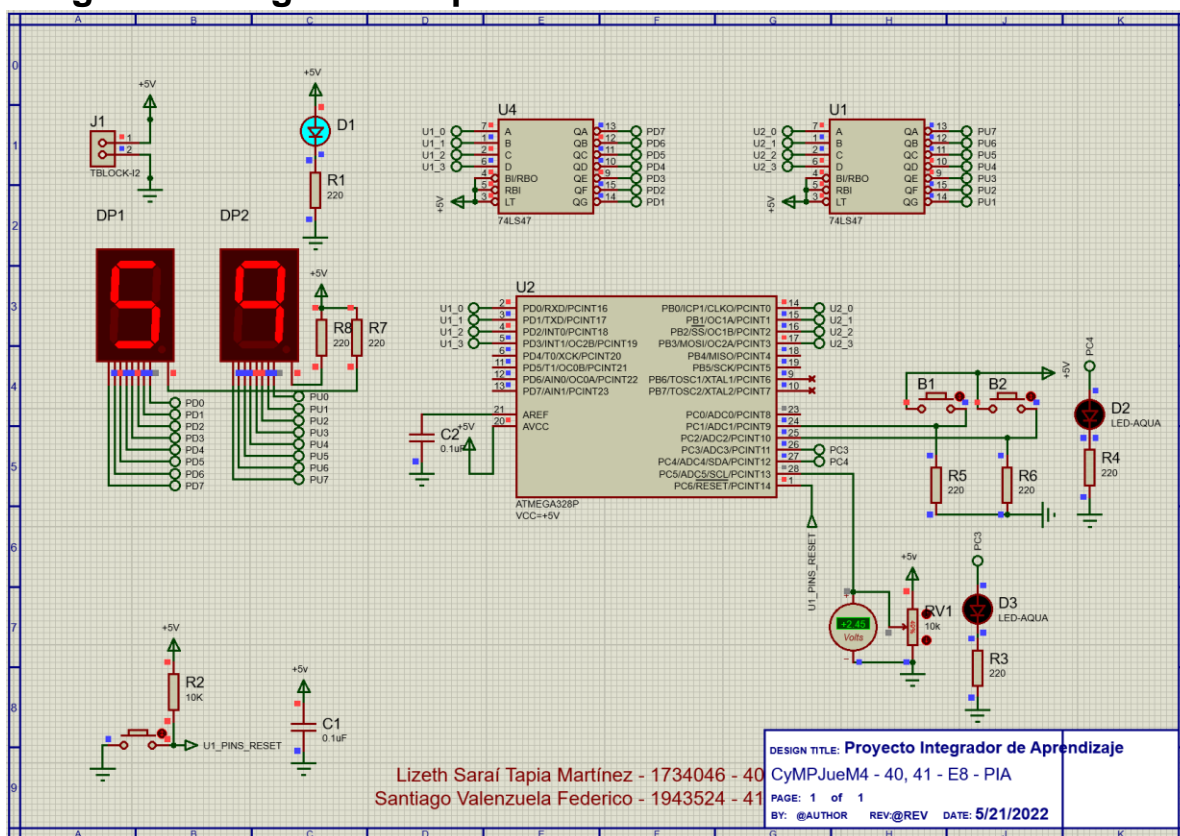


Tabla de estado siguiente

Estado Actual	Estado Siguiente	Entradas	Salidas
Modo 1	Modo 2	Btn 1	Timer (0 a 60)
Modo 2	Modo 3	Btn 1 & Btn 2	Contador (0 a 30)
Modo 3	Modo 4	Btn 1 & Btn 2	Contador desc(0 a 30)
Modo 4	Modo 1	Btn 1 & Pot 2	Rango de V(+5v). (0 a 99)

Diagrama de transición de estados finitos

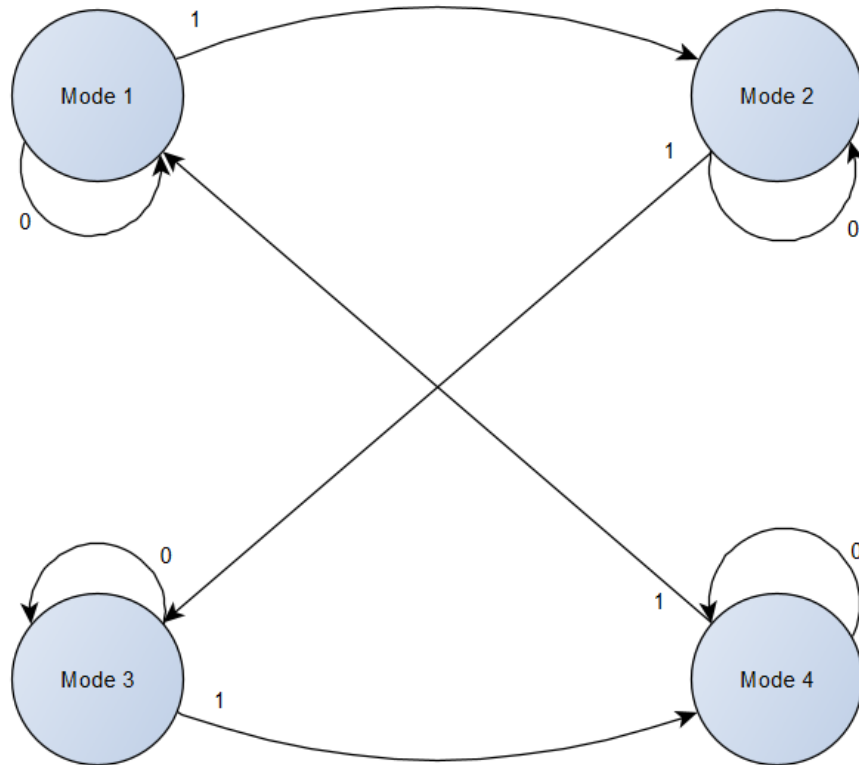
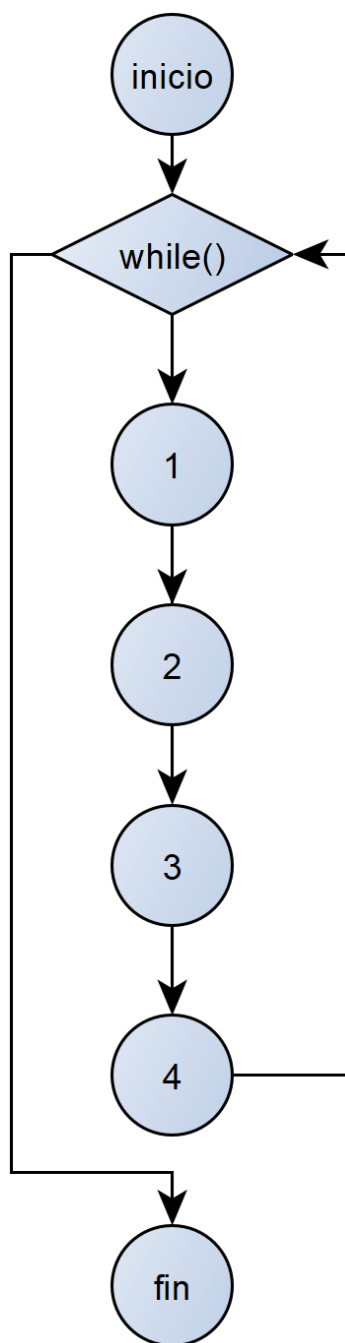
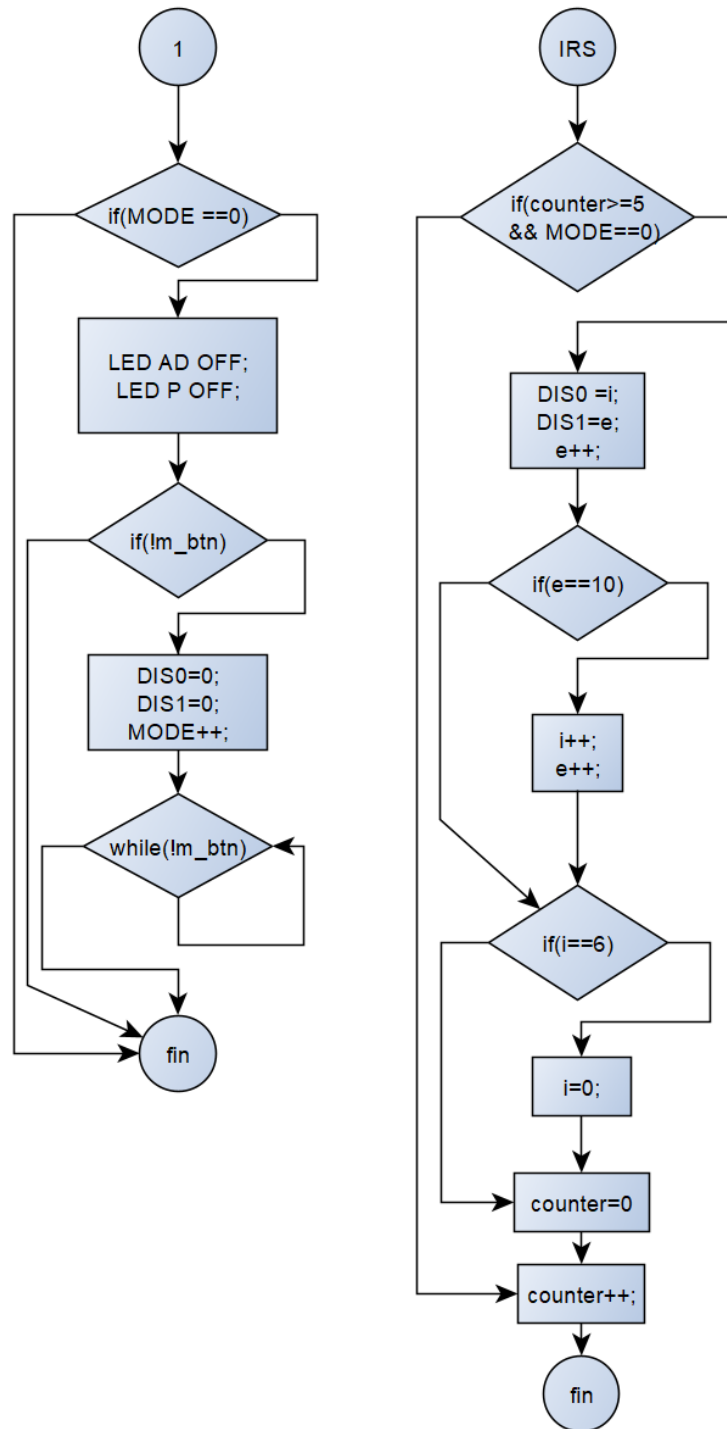


Diagrama de flujo del código en alto nivel

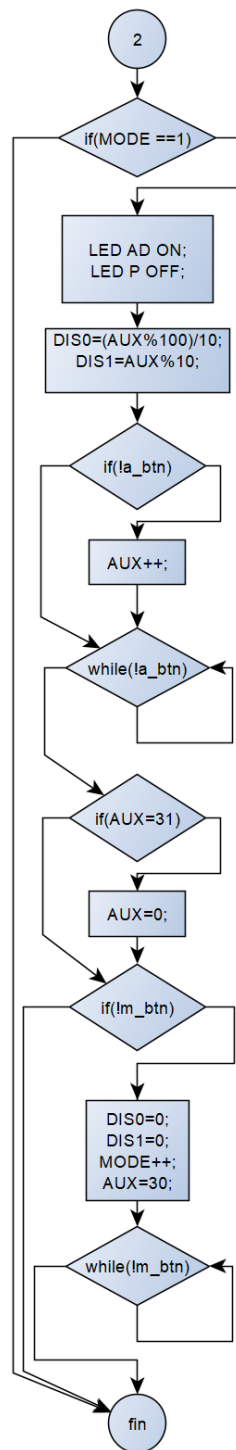
General



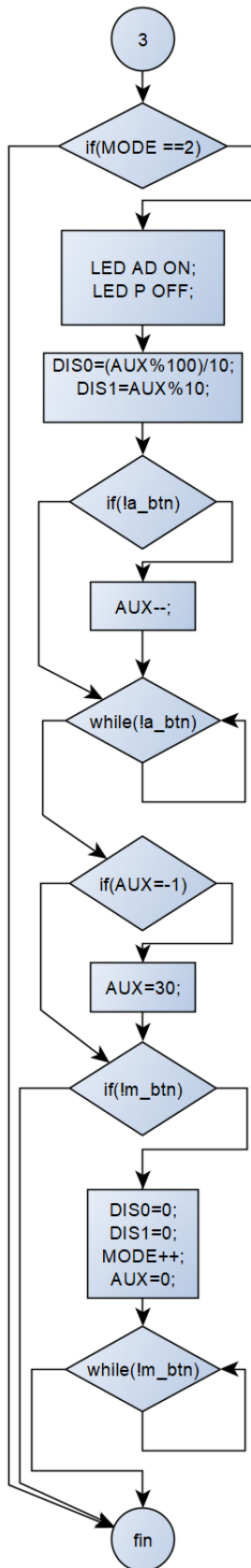
Seccion del Modo 1 y Timer



Seccion del Modo 2



Seccion del Modo 3



Seccion del Modo 4

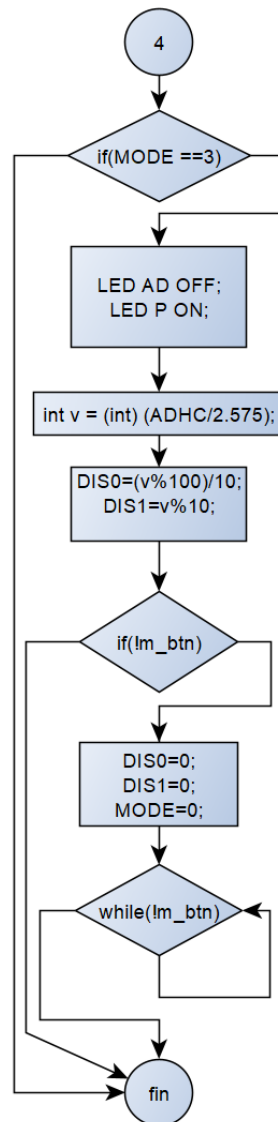


Imagen del código en lenguaje C o C++ con los nombres de sus integrantes

```
/*
 * Producto Integrador de Aprendizaje.c
 *
 * Created: 5/11/2022 8:37:29 PM
 * Equipo :
 *
 *      +-----+-----+-----+
 *      |Lizeth Saraí Tapia Martínez | 1734046 | 40 |
 *      |Santiago Valenzuela Federico | 1943524 | 41 |
 *      +-----+-----+-----+
 */
#include <avr/io.h>
#define F_CPU 1000000UL
#include <util/delay.h>
#include <stdio.h>
#include <avr/interrupt.h> //Biblioteca de interrupciones
//Salidas
//--Display

#define DIS0 PORTD
#define DIS1 PORTB

#define LED4 PORTC4
#define LED_AD_ON  PORTC |= _BV(LED4)
#define LED_AD_OFF PORTC &= ~_BV(LED4)

#define LED3 PORTC3
#define LED_P_ON   PORTC |= _BV(LED3)
#define LED_P_OFF PORTC &= ~_BV(LED3)

#define BUTTONS PINC
//--BTN0
#define BUTTON0 PINC1
#define BUTTON0_PRESS bit_is_clear(BUTTONS,BUTTON0)
//--BTN1
#define BUTTON1 PINC2
#define BUTTON1_PRESS bit_is_clear(BUTTONS,BUTTON1)

#define abtn BUTTON1_PRESS
#define mbtn BUTTON0_PRESS

int MODE = 0;
int AUX = 0;
int i = 0;
int e = 0;
int counter=0;

//Entradas
//--Declaracion de funciones
void init_ports(void);
void init_timer0(void);
void on_timer0(void);
void off_timer0(void);
```

```

void init_ADC (void);
void on_ADC   (void);
void off_ADC  (void);
//Variable global

//Array
uint8_t numeros[10] = {0,1,2,3,4,5,6,7,8,9};

//-----MAIN-----
int main(void)
{
    cli(); // deten interrupciones
    init_ports();
    init_timer0();
    on_timer0();
    init_ADC();
    on_ADC();
    sei(); // activa interrupciones
    while(1)
    {
        if (MODE==0)
        {
            LED_AD_OFF;
            LED_P_OFF;
            if (!mbtn)
            {
                LED_AD_OFF;
                DIS0=0;
                DIS1=0;
                MODE++;
                AUX=0;
                while(!mbtn){}
            }
        }
        if (MODE==1)
        {
            LED_AD_ON;
            LED_P_OFF;
            DIS0=(AUX%100)/10;
            DIS1=AUX%10;
            if (!abtn)
            {
                AUX++;
                while(!abtn){}
            }
            if (AUX==31)
            {
                AUX=0;
            }
            if (!mbtn)
            {
                DIS0=0;
                DIS1=0;
                AUX=30;
                MODE++;
                while(!mbtn){}
            }
        }
    }
}

```

```

}
if (MODE==2)
{
LED_AD_ON;
LED_P_OFF;
DIS0=(AUX%100)/10;
DIS1=AUX%10;
if (!abtn)
{
AUX--;
while(!abtn){}
}
if (AUX==-1)
{
AUX=30;
}
if (!mbtn)
{
DIS0=0;
DIS1=0;
MODE++;
while(!mbtn){}
}
}
if (MODE==3)
{
LED_AD_OFF;
LED_P_ON;
int ax = (int)(ADCH/2.575);
DIS0 = (ax%100)/10;
DIS1 = ax%10;
if (!mbtn)
{
DIS0=0;
DIS1=0;
i = 0;
e = 0;
MODE=0;
while(!mbtn){}
}
}
/**/
} //Fin_while
}
// Fin_main
//--Funciones
/*****
/* init_ports() inicializa por puertos E/S
*****/
void init_ports(void)
{
//--OUTPUTS
//DISPLAY
DDRD |= _BV(DDD0) | _BV(DDD1) | _BV(DDD2) | _BV(DDD3);
DDB0 |= _BV(DDB0) | _BV(DDB1) | _BV(DDB2) | _BV(DDB3);
DDRC |= _BV(LED4) | _BV(LED3);

//Leds

```

```

    DDRD = 0xFF;
    DDRB = 0xFF;
    //--INPUTS
    DDRC  &= ~(1<<BUTTON1)|(1<<BUTTON0));
}
void init_ADC (void)
{
    //Ref AVCC
    ADMUX &= ~_BV(REFS1);
    ADMUX |=  _BV(REFS0);

    //Ajustar a 8 bits
    ADMUX |=  _BV(ADLAR);

    //Escoger el PIN ADC5
    ADMUX &= ~_BV(MUX3);
    ADMUX |=  _BV(MUX2);
    ADMUX &= ~_BV(MUX1);
    ADMUX |=  _BV(MUX0);

    //Freerunning
    ADCSRA |=  _BV(ADSCF);

    //Enable interrupt
    ADCSRA |=  _BV(ADIF);

    //velocidad de muestreo entre 8
    ADCSRA &= ~_BV(ADPS2);
    ADCSRA |=  _BV(ADPS1);
    ADCSRA |=  _BV(ADPS0);
}
void on_ADC (void)
{
    //Encender ADC
    ADCSRA |=  _BV(ADEN);
    //Inicias conver
    ADCSRA |=  _BV(ADSC);
}

ISR (TIMER0_COMPA_vect)
{
    if (counter>=5 && MODE==0)
    {
        DIS0=i;
        DIS1=e;
        e++;
        if (e==10)
        {
            i++;
            e=0;
        }
        if (i==6)
        {
            i=0;
        }
        counter=0;
    }
}

```



```

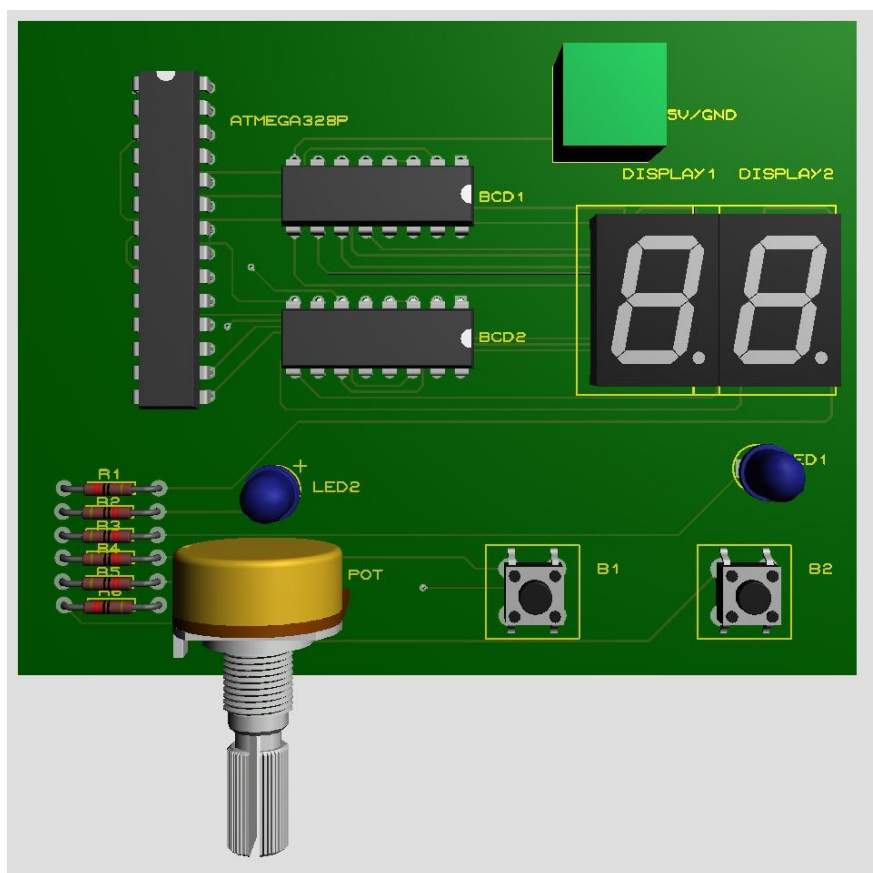
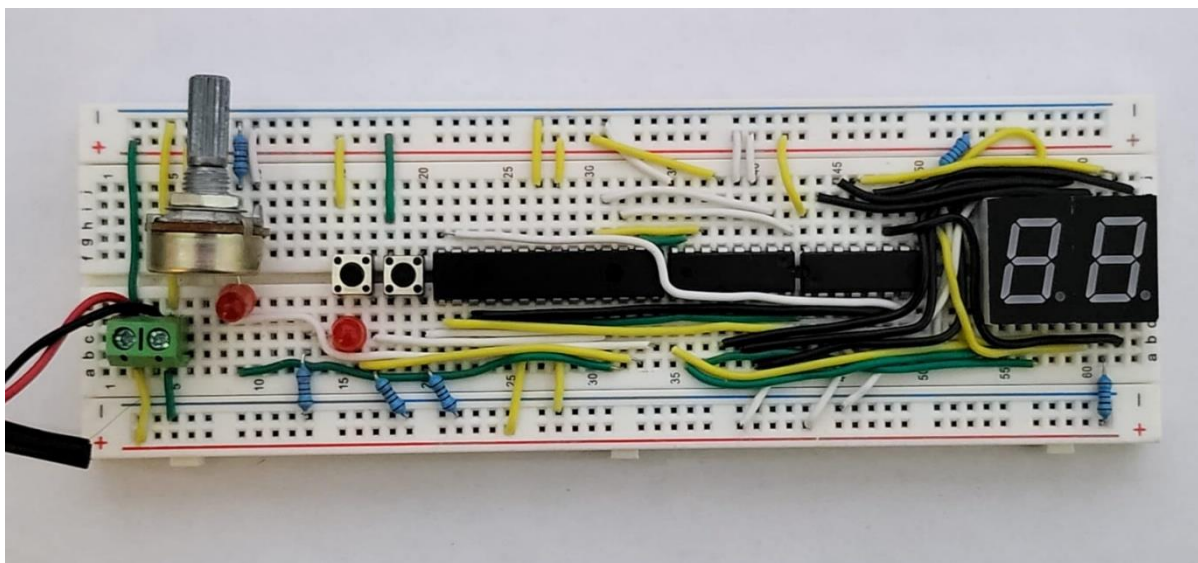
counter++;
}
ISR(ADC_vect)
{
}
void init_timer0(void)
{
//Modo CTC
TCCR0A &= ~_BV(WGM00);
TCCR0A |= _BV(WGM01);
TCCR0B &= ~_BV(WGM02);
//Tope
OCR0A = 97 ;// 0.1s
//Enable
TIMSK0 |= _BV(OCIE0A);
}
void on_timer0(void)
{
//Reiniciar conteo
TCNT0 = 0;
//Prescaler 1024
TCCR0B |= _BV(CS02);
TCCR0B &= ~_BV(CS01);
TCCR0B |= _BV(CS00);
}
void off_timer0(void)
{
//Stop
TCCR0B &= ~_BV(CS02);
TCCR0B &= ~_BV(CS01);
TCCR0B &= ~_BV(CS00);
}

```

Simulación funcionando de la actividad en un video (.mp4) en un archivo aparte

Proteus: [Simulacion Proteus PIA.mp4](#)

Fotografía del prototipo armado solo el área útil



Video del circuito físico funcionando (.mp4) anexo a la entrega

Prototipo: [Video Prototipo - PIA.mp4](#)

Explicación del funcionamiento

El sistema se divide por 4 modos y el timer:

Modo1: habilita la condición del timer, el cual funciona cuando solo cuando el modo 1 esta activado. Este muestra un contador de 0 a 59 el cual cambia cada .5 segundos.

Modo2: Presionando el botón de modo este desactiva la condición del timer e inicia un contador manual el cual iniciara en 0 y se deberá presionar el botón de agregar o restar al contador. Después se harán incrementos del 0 al 30.

Modo3: Presionando de nuevo el botón de modo este cambiara de modo, en este se reiniciarán los valores y se establecerán en 30 para entonces proceder a decrementos individuales de 30 a 0.

Modo4: Al presionar el botón de modo nuevamente estaremos en el modo 4. Aquí se inicia el ADCH, el cual leerá los valores de entrada del potenciómetro que oscilan desde 0 a 255 y los transformara a su equivalente del rango 0 a 99. También presionando nuevamente el botón de modo volveríamos al modo 1, así reiniciando el proceso.

Conclusión personal

Lizeth Sarai Tapia Martinez 1734046

En este proyecto hicimos un sistema electrónico que muestre varias funciones mediante dos display de 7 segmentos, el uso de dos display no se habia realizado anteriormente, por lo cual es una practica "desbloqueada". De igual manera tambien reforzamos el conocimiento del BCD (descodificador), al volver a trabajar con el.

Santiago Valenzuela Federico – 1943524

En este proyecto aplicamos varios temas vistos en clase y aplicarlos de manera bastante especifica. Cada los modos 1 2 y 4 aplican diferentes temas vistos en clase. En el caso del modo 1 aplicamos TIMER y el uso del BCD, en el modo 2 y 3 aplicamos uno que vemos viendo desde el inicio, es el uso de botones. En el modo 4 aplicamos uno de los últimos temas que vimos fue el ADCH este transforma la entrada analógica de voltaje para expresarlo en un valor decimal.

Referencias

S.T.A.N.N.O. (2021, 10 agosto). *One Digit 7-Segment LED Display*. Arduino

Project Hub. <https://create.arduino.cc/projecthub/stannano/one-digit-7-segment-led-display-70b1a0>

Wikipedia. (2022, 20 febrero). *Pinout of ARDUINO Board and ATmega328PU*.

Wikipedio.com.

https://commons.wikimedia.org/wiki/File:Pinout_of_ARDUINO_Board_and_ATMega328PU.svg

Texas Instruments. (s. f.). *SN74LS47N - BCD-to-Seven-Segment Decoders/Drivers*. Texas

Instruments - DataSheet. Recuperado 13 de mayo de 2022, de

<https://www.ti.com/store/ti/en/p/product/?p=SN74LS47N&HQS=ti-null-null-ds360-invf-store-ds360-wwe#>

Steren. (s. f.). *Display de 7 segmentos, ánodo común, de 12,7 mm (0,5 pulgadas)*.

Recuperado 15 de mayo de 2022, de <https://www.steren.com.mx/display-de-7-segmentos-anodo-comun-de-12-7-mm-0-5-pulgadas.html>

Lang, J. (2018, 16 junio). *Potentiometer - CAD*. GRABCAD. Recuperado 18 de mayo de

2022, de <https://grabcad.com/library/potentiometer-14>

Loïc, T. (2020, 15 noviembre). *7 Segment display*. GRABCAD. Recuperado 18 de mayo de

2022, de <https://grabcad.com/library/7-segment-display-9>

J. (2018, 30 octubre). *Tactile Switch - SMD (6mm Series)*. GRABCAD. Recuperado 18 de

mayo de 2022, de <https://grabcad.com/library/tactile-switch-smd-6mm-series-1>

