 **Universidad Autónoma de Nuevo León**

**Facultad de Ingeniería Mecánica y Eléctrica**

#LAB. CONTROLADORES Y MICROCONTROLADORES PROGRAMABLES

#Práctica P4

" Display 7 segmentos "

\*Nombre o nombres de los integrantes junto con su matrícula:

#Verónica Yazmín Gómez Cruz #1884224

#Nahaliel Gamaliel Ríos Martínez #1884244

#Ing. Jesús Daniel Garza Camarena

Semestre Febrero 2021 – Junio 2021

# MN1N2

San Nicolás de los Garza, N.L. #15.04.2021

# Objetivo

Conocer el uso de los displays y el diseño de los contadores utilizando un microcontrolador.

# Introducción.

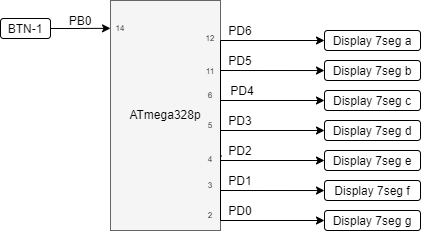
Los displays de siete segmentos es una configuración particularmente versátil. Al iluminar diferentes combinaciones de los siete segmentos, se pueden mostrar todos los dígitos numéricos, así como algunos caracteres alfabéticos. Por lo general, se incluye un punto decimal, como se muestra. Esto significa que hay ocho LED en la pantalla, que necesitan 16 conexiones. Para simplificar las cosas, todos los ánodos LED están conectados entre sí o todos los cátodos LED. Los dos posibles patrones de conexión se denominan cátodo o ánodo comunes. Ahora, en lugar de que se necesiten 16 conexiones, solo hay nueve, una para cada LED y una para la conexión común. Las conexiones de clavijas reales en el ejemplo que se muestra se encuentran en dos filas, en la parte superior e inferior del dígito. Hay diez pines en total, con el ánodo o cátodo común tomando dos pines.

Display de siete segmentos (Kingbright, 12,7 mm).

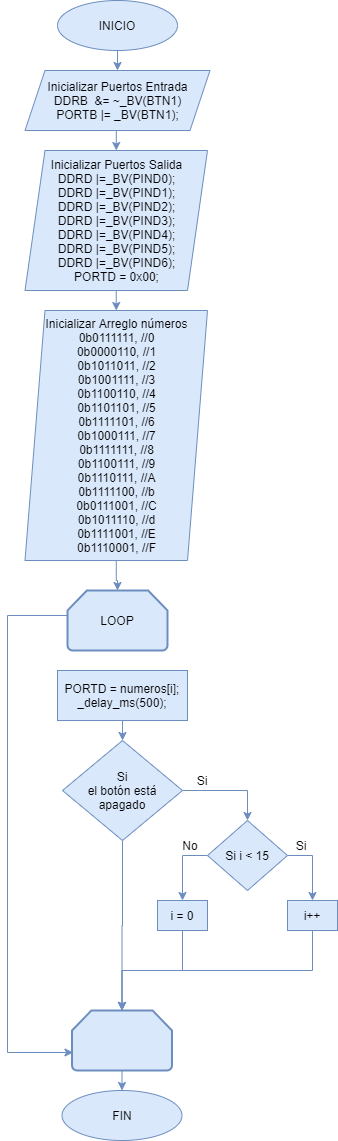
Diagrama, Dibujo de ingeniería

Descripción generada automáticamente

**Diagrama de bloques**

****

# Diagrama de flujo.



# Materiales utilizados

1 ATMEGA328P

1 Display 7 seg cc

2 Push button

9 Resistencia 220

Jumpers de distintos colores

# Código en Atmel.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* LLENAR ESTE ESPACIO CON LOS SIGUIENTES DATOS: \*

\* Nombre: Verónica Yazmín Gómez Cruz \*

\* Nahaliel Gamaliel Rios Martinez \*

\* Hora clase: N1-N2 \*

\* Día: M \*

\* N° de lista: 17, 18 \*

\* N° de Equipo: 7 \*

\* Dispositivo: ATMEGA328P \*

\* Rev: 1.0 \*

\* Propósito de la actividad: \*

\* Hacer un contador hexadecimal (0 a F) automático \*

\* con una velocidad de 0.5 segundos, agregar un \*

\* switch de pausa que al presionarlo detenga \*

\* conteo y se muestre el número donde se quedó y \*

\* al soltarlo el conteo continúe. \*

\* Fecha: 15.04.2021 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*atmega328P PIN - OUT\*/

/\* PIN - OUT

atmega328P

-------

PC6 |1 28| PC5

PD0 |2 27| PC4

PD1 |3 26| PC3

PD2 |4 25| PC2

PD3 |5 24| PC1

PD4 |6 23| PC0

VCC |7 22| GND

GND |8 21| AREF

PB6 |9 20| AVCC

PB7 |10 19| PB5

PD5 |11 18| PB4

PD6 |12 17| PB3

PD7 |13 16| PB2

PB0 |14 15| PB1

--------

\*/

/\*atmega328P PIN FUNCTIONS\*/

/\*

atmega328P PIN FUNCTIONS

pin function name pin function name

1 !RESET/PCINT14 PC6 15 PCINT1/OC1A PB1

2 RxD/PCINT16 PD0 16 PCINT2/OC1B/SS PB2

3 TxD/PCINT17 PD1 17 PCINT3/OC2A/MOSI PB3

4 INT0/PCINT18 PD2 18 PCINT4/MISO PB4

5 INT1/PCINT19/OC2B PD3 19 PCINT5/SCK PB5

6 PCINT20 PD4 20 ANALOG VCC AVCC

7 +5v VCC 21 ANALOG REFERENCE AREF

8 GND GND 22 GND GND

9 XTAL1/PCINT6 PB6 23 PCINT8/ADC0 PC0

10 XTAL2/PCINT7 PB7 24 PCINT9/ADC1 PC1

11 PCINT21/OC0B PD5 25 PCINT10/ADC2 PC2

12 PCINT22/OC0A/AIN0 PD6 26 PCINT11/ADC3 PC3

13 PCINT23/AIN1 PD7 27 PCINT12/ADC4/SDA PC4

14 PCINT0/AIN1 PB0 28 PCINT13/ADC5/SCL PC5

\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Bibliotecas\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <avr/io.h>//se incluyen las Bibliotecas de E/S del AVR atmega328P

#include <util/delay.h> //Biblioteca para usar \_delay\_ms

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Macros y constantes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#define *F\_CPU* 1000000UL //1 Mhz

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Variables globales\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//--Espacio para declarar variables globales

#define BTN1 PINB0

#define a PIND0

#define b PIND1

#define c PIND2

#define d PIND3

#define e PIND4

#define f PIND5

#define g PIND6

*uint8\_t* numeros[16] = {

//gfedcba

0b0111111, //0

0b0000110, //1

0b1011011, //2

0b1001111, //3

0b1100110, //4

0b1101101, //5

0b1111101, //6

0b1000111, //7

0b1111111, //8

0b1100111, //9

0b1110111, //A

0b1111100, //b

0b0111001, //C

0b1011110, //d

0b1111001, //E

0b1110001, //F

};

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Funciones\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//--Espacio para Establecer funciones

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Declaración de Funciones\*\*\*\*\*\*\*\*\*\*\*\*\*/

//--Espacio para declarar funciones

void initialize\_ports(void);

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Programa principal\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int main(void)

{

//--Inicialización

initialize\_ports(); // va hacía la inicialización de puertos

*uint8\_t* i = 0;

*uint8\_t* type = 0;

//--Ejecución

while (1) //loop infinito

{

PORTD = numeros[i];

*\_delay\_ms*(500);

if (bit\_is\_set(PINB,BTN1) ){

if (i < 15) {

i++;

}else{

i = 0;

}

}

} // END loop infinito

} // END MAIN

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Definición de funciones\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//Descripcion de lo que hace la funcion: \*

//initialize\_ports : inicializa los puertos de entrada o \*

//salida \*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initialize\_ports(void)

{

//--Entradas

// Declaramos puerto de entrada para boton

DDRB &= ~\_BV(BTN1);

PORTB |= \_BV(BTN1); //Activamos PULL UP

//--Salidas

DDRD |=\_BV(a);

DDRD |=\_BV(b);

DDRD |=\_BV(c);

DDRD |=\_BV(d);

DDRD |=\_BV(e);

DDRD |=\_BV(f);

DDRD |=\_BV(g);

PORTD = 0x00; //-Por seguridad iniciamos en 0

}

# Diagrama del circuito en PROTEUS.

Diagrama

Descripción generada automáticamente con confianza baja

**Fotografía del Protoboard armado**

Un cable conectado

Descripción generada automáticamente con confianza baja

**Conclusión**

En esta práctica realizamos la conexión de un display de 7 segmentos en simulación y en físico. Lo más retador en esta actividad fue realizar el circuito en físico, pero tras varias pruebas logramos hacer que funcionara como en la simulación. Para hacer que el display mostrara el digito correcto realizamos un arreglo con cada número hexadecimal, y en el loop infinito solo llamamos la posición de dicho array con una variable que llamamos i, después asignamos los valores del array al puerto de salida. La variable i va aumentando de uno en uno hasta llegar a 15, cuando llega 15 se le asigna el valor de 0 para volver a empezar. Para hacer la función de pausa al presionar un botón, solo agregamos una condición para que cuando el botón estuviera presionado la variable i no aumentara.

**Bibliografía**

Toulson, R., & Wilmshurst, T. (2016). Fast and Effective Embedded Systems Design: Applying the Arm Mbed (2nd ed.). Newnes.