 **Universidad Autónoma de Nuevo León**

**Facultad de Ingeniería Mecánica y Eléctrica**

#LAB. CONTROLADORES Y MICROCONTROLADORES PROGRAMABLES

#Práctica P2

"Microcontroladores"

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

#Verónica Yazmín Gómez Cruz #1884224

#Nahaliel Gamaliel Ríos Martínez #1884244

#Ing. Jesus Daniel Garza Camarena

Semestre Febrero 2021 – Junio 2021

# MN1N2

San Nicolás de los Garza, N.L. #05.03.2021

# Objetivo

Entender en qué consisten las funciones en código C y como están formados los retardos del Microcontrolador, practicar el uso de las máscaras de bits mediante compuertas lógicas.

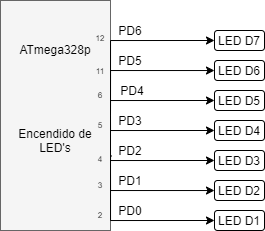
# Introducción.

Cuando necesitamos que transcurra un determinado tiempo de espera antes de que ocurra un evento como por ejemplo el encendido de una luz, LED, activación de una bobina de un relé o lectura de una determinada entrada, se suele recurrir a los retardos. Prácticamente casi todos los programas de microcontroladores PIC usan en algún momento una rutina de retardo.

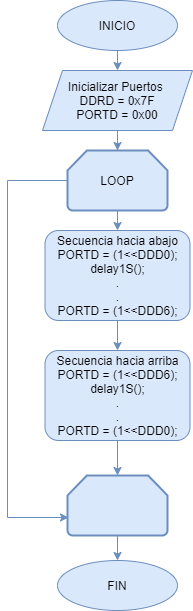
Los retardos en los PIC los podemos generar de dos formas diferentes: por software o por hardware mediante el Registro Timer TMR0.

Los retardos por Software consisten en que el PIC se quede ejecutando unos bucles que van decrementado unos contadores cargados previamente con el tiempo de retardo, cuando los contadores llegan a 0 la rutina de retardo queda terminada y el microcontrolador sigue ejecutando otros procesos.

**Diagrama de bloques**



# Diagrama de flujo.



# Materiales utilizados

1 ATmega328p

7 LED’s Azules

7 resistencias 220

1 resistencia 10K

2 capacitores 0.2F

# 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: \*

\* Crear la secuencia Coche Fantástico con 7 \*

\* Leds a la salida \*

\* Fecha: 5.3.2021 \*

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

/\*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

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

#define F\_CPU 1000000UL //1 Mhz

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

//--Espacio para declarar variables globales

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

//--Espacio para Establecer funciones

void delay1S(){

//Esta funcion sirve para hacer un retardo

volatile unsigned int i;

for (i = 0; i<55555; i++);

}

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

//--Espacio para declarar funciones

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

int main(void)

{

//--Inicialización

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

//--Ejecución

while (1) //loop infinito

{

//Secuencia hacia abajo

PORTD = (1<<DDD0);

delay1S();

PORTD = (1<<DDD1);

delay1S();

PORTD = (1<<DDD2);

delay1S();

PORTD = (1<<DDD3);

delay1S();

PORTD = (1<<DDD4);

delay1S();

PORTD = (1<<DDD5);

delay1S();

PORTD = (1<<DDD6);

delay1S();

//Secuencia hacia arriba

PORTD = (1<<DDD6);

delay1S();

PORTD = (1<<DDD5);

delay1S();

PORTD = (1<<DDD4);

delay1S();

PORTD = (1<<DDD3);

delay1S();

PORTD = (1<<DDD2);

delay1S();

PORTD = (1<<DDD1);

delay1S();

PORTD = (1<<DDD0);

delay1S();

} // 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

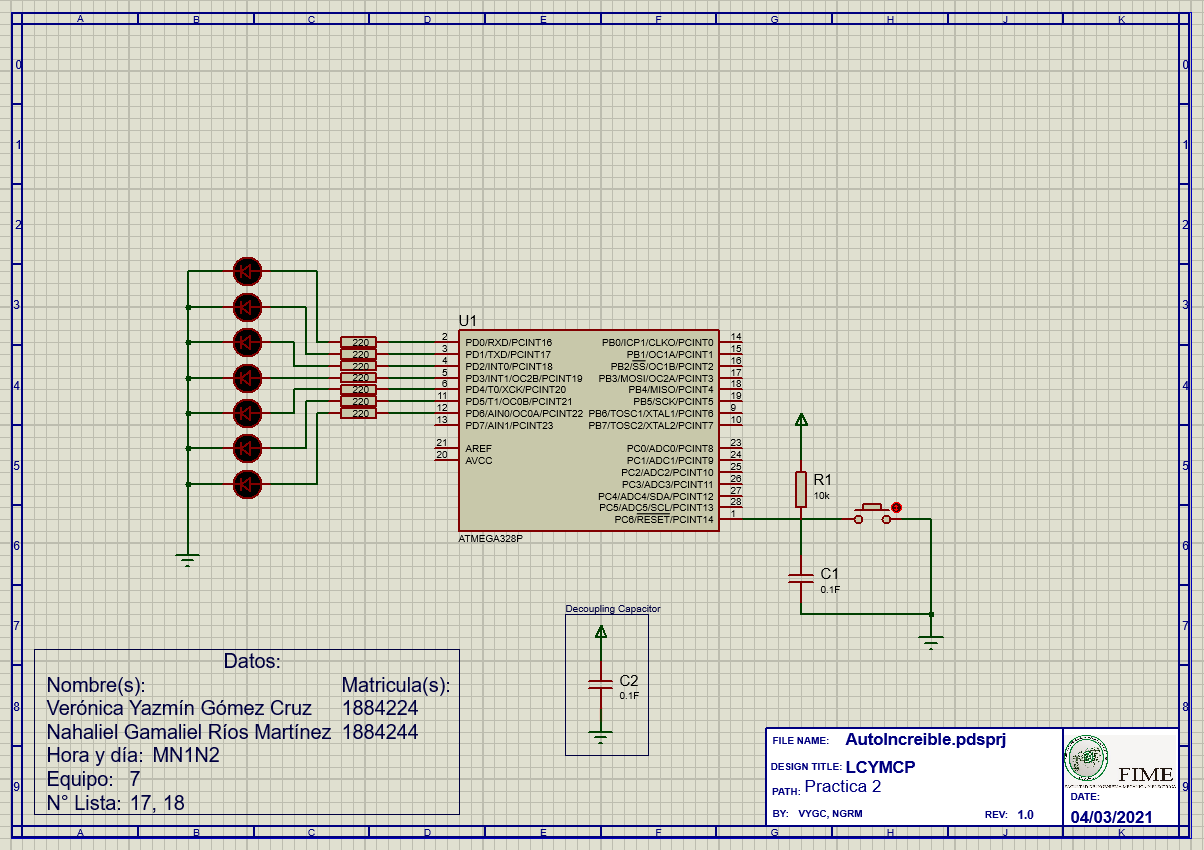
//--Salidas

DDRD = 0x7F; //-Declaramos 7 puertos de salida

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

}

# Diagrama del circuito en PROTEUS.



**Conclusión**

En esta práctica utilizamos Proteus para las conexiones de leds, así como la codificación en MICROCHIP STUDIO, en el código se vio la asignación de pines en modo de salida, la forma de encender y apagar un led, se usó sentencias de ciclos, asignación, condiciones y funciones en C. Con las funciones se creó una rutina para hacer que el led tardara en encender y apagar, dentro de la rutina se ejecuta un ciclo que puede durar n veces, dependiendo del tiempo de espera que queramos configurar.