



**Universidad Autónoma de Nuevo León**  
**Facultad de Ingeniería Mecánica y Eléctrica**



**#LAB. CONTROLADORES Y  
MICROCONTROLADORES PROGRAMABLES**

**#Práctica P1  
"Microcontroladores"**

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

<b>#Verónica Yazmín Gómez Cruz</b>	<b>#1884224</b>
<b>#Nahaliel Gamaliel Ríos Martínez</b>	<b>#1884244</b>

**#Ing. Jesus Daniel Garza Camarena**

**Semestre Febrero 2021 – Junio 2021**

**# MN1N2**

**San Nicolás de los Garza, N.L.**

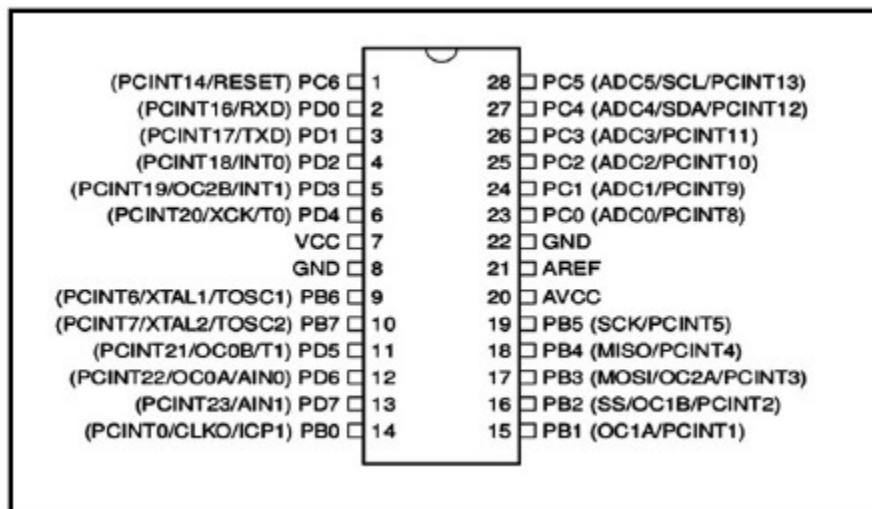
**#25.02.2021**

## Objetivo

Conocer y entender los puertos que se manejan en el Microcontrolador atmega328p

## Introducción.

Es importante conocer la disposición concreta de las patillas (pines) de entrada/salida del microcontrolador, aunque en general todos los pines de E/S sirven para comunicar el microcontrolador con el mundo exterior, es cierto que cada pin suele tener una determinada función específica. Como cada modelo de microcontrolador tiene un número y ubicación de pines diferente, en este caso debemos tener a mano la disposición de pines del ATmega328P. La figura siguiente muestra esta disposición en el encapsulado de tipo DIP.



**VCC:** 1 pin para conexión a corriente

**GND:** 2 pines conectados a tierra

**AVCC:** recibe alimentación suplementaria para el convertidor analógico-digital interno del chip.

**AREF:** recibe la referencia analógica para dicho convertidor.

Pines de E/S: Incluye 3 puertos, todas las terminales con más de una función alterna.

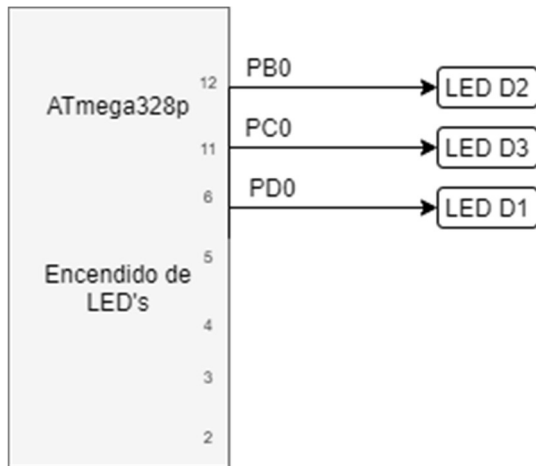
- PBx (8 bytes)
- PCx (7 bytes)
- PDx (8 bytes)

**Se requiere de tres registros I/O para el manejo de cada puerto:**

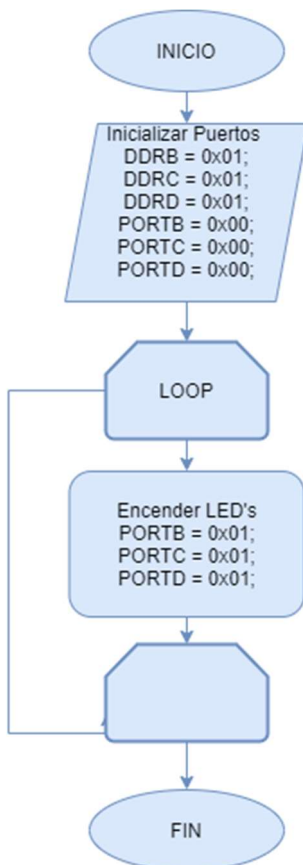
- **DDRx:** Registro que define la dirección del puerto con el apoyo de un buffer de 3 estados. Las direcciones se pueden configurar de manera independiente, para cada terminal.
  - 1 – Salida
  - 0 – Entrada
- **PORTx:** Registro conectado a la terminal del puerto a través del buffer de 3 estados, en PORTx se escribe cuando el puerto está configurado como salida. Si el puerto es entrada sirve para habilitar un resistor de Pull-Up.

- PINx: Si el puerto es entrada, este registro sirve para hacer lecturas directas en las terminales. Cuando es salida, al escribir un 1 lógico se conmuta el valor almacenado.

## Diagrama de bloques



## Diagrama de flujo.



## Materiales utilizados

1 ATmega328p  
1 LED verdes  
1 LED Amarillos  
1 LED Azules  
2 capacitores  
1 botón  
3 resistencias

## Código en Atmel.

```

/*****
 * LLENAR ESTE ESPACIO CON LOS SIGUIENTES DATOS:  *
 * Nombre: Verónica Yazmín Gómez Cruz              *
 *           Nahaliel Gamaliel Rios Martinez         *
 * Hora clase: N1-N2                               *
 * Día: Martes                                     *
 * N° de lista: 17-18                             *
 * N° de Equipo: 7                                 *
 * Dispositivo: ATMEGA328P                         *
 * Rev: 2.0                                         *
 * Propósito de la actividad:                      *
 * Encender un led por puerto                      *
 *                                           Fecha: 14/03/21 *
 *****/
/*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

/*****Programa principal*****/
int main(void)
{
    //Funcion para declarar puertos I/O
    initialize_ports();

    while (1)
    {
        //Encender LED Verde
        PORTB = 0x01;

        //Encender LED Azul
        PORTC = 0x01;

        //Encender LED Amarillo
        PORTD = 0x01;
    }
}

/*****Definición de funciones*****/
/*****
//Descripcion de lo que hace la funcion: *
//initialize_ports : inicializa los puertos de entrada o *
//salida *
// *
// *
/*****
void initialize_ports(void){

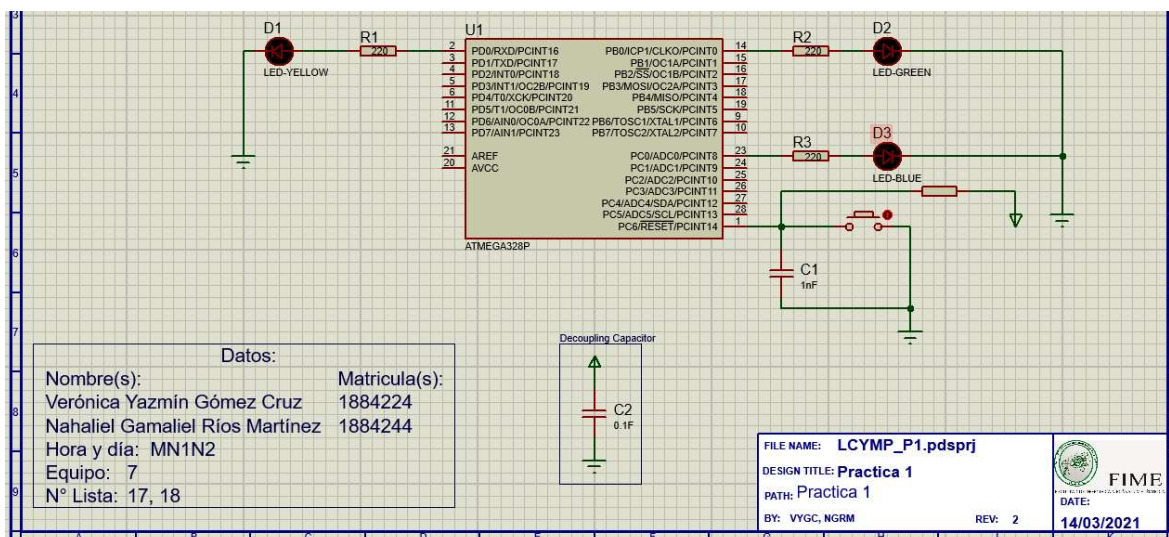
    //Salidas
    DDRB = 0x01;
    DDRC = 0x01;
    DDRD = 0x01;

    //Por seguridad iniciamos en 0

}

```

## Diagrama del circuito en PROTEUS.



## Conclusión

En esta primera práctica, se revisó el entorno de desarrollo con el que estaremos trabajando a lo largo del semestre, que consta de Microchip Studio para desarrollar en C y Proteus para hacer simulaciones de los circuitos. Se crearon los primeros proyectos en cada uno de estos sistemas y se vio como ejecutar código C desde Proteus, además de algunas recomendaciones y generalidades para la organización de nuestros proyectos. Aprendimos a declarar puertos de entrada del ATmega328P, y como hacer que tengan un valor alto o bajo.

## Bibliografía

Oscar Torrente Artero. (2013). Arduino Curso práctico de formación. Madrid, España: Alfaomega Grupo Editor, S.A. de C.V., México.