Laboratorio de microcomputadoras 66.09
Laboratorio de microprocesadores 86.07

Guía de Trabajos Prácticos

2° cuatrimestre

Año 2020

INDICE

Introducción

Antes de empezar a leer esta guía, es recomendable leer la guía de usuario del programador, en la que se explica cómo usar el Programador y como instalar los programas necesarios para poder empezar a programar.

Para realizar estas prácticas deberán conseguir algunos materiales.

Las prácticas pueden hacerse con cualquier microcontrolador de la familia ATmega, en esta guía están explicadas para el microcontrolador ATmega328 o compatible, de 28 pines. Si usan otro micro, por ejemplo si usan un ATmega32 que tiene 40 pines, deberán ver a que pines tienen que conectar los elementos, leds, resistencias, etc.

En esta guía nos referimos también a **micro** en forma abreviada haciendo referencia al microcontrolador.

Pueden usar la placa de desarrollo Arduino que contiene un micro

ATmega 328p, deben ver que pines del conector del arduino conectan con los pines del micro pedidos en la práctica.

Al final de esta guía se adjunta imagen de pinouts de varios micros y arduino uno.

Trabajo Práctico N° 1
-Manejo de Puertos

Objetivo:

Usar los registros de los puertos, ver utilidad de R pullup interna

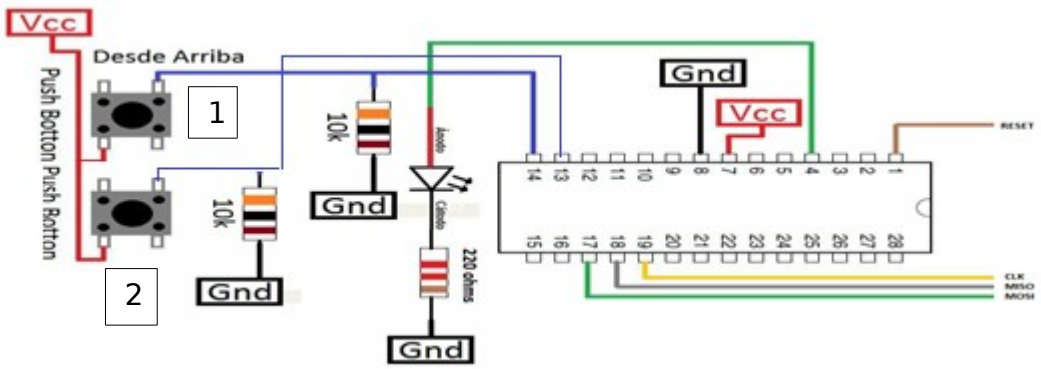
Practica:

1- Hacer un programa que haga parpadear un LED conectado en el PIN 2 (Usar la rutina de retardo Dada)
2- Modificar programa para que prenda un LED cuando se presiona el pulsador 1 y quede parpadeando hasta que se apague cuando se presiona el botón 2. El LED está conectado a un pin del microcontrolador y los pulsadores a otros dos pines.
3- Como modifica el circuito y el programa, para usar la resistencia de pullup interna de los ports al conectar el pulsador. Se ahorra algún componente?

Materiales

1 LED
1 Resistencia de 220 Ohms
2 Resistencia de 10Kohms
2 Pulsador
1 Microcontrolador ATmega328p
Programador USBasp V3.0

Diagrama Esquemático



Lectura recomendada

“The AVR microcontroller and embedded systems. Embedded system using Assembly and C”. Autores: MUHAMMAD ALI MAZIDI, SARMAD NAIMI, SEPEHR NAIMI

Ejemplos de generación de un retardo

```
------------------------------------------------------------
; Rutina  de Retardo de 5ms
;------------------------------------------------------------

delay5ms:
        ldi Temp1, 66              ; para 8mhz
                                     ; 1 ciclo
LOOP0:
        ldi temp2, 200             ; 1 ciclo
LOOP1:
        dec temp2                   ; 1 ciclo
        brne LOOP1                 ; 1 si es falso
2 si es
                                           ;verdade
ro
         dec Temp1 ; 1
        brne LOOP0 ; 2
ret
```

```
;------------------------------------------------------------
Calculo del tiempo de retardo)

Llamada 5ms                    * 1
Ldi  Temp1                     * 1
LDI Temp 2                     * 66
dec  temp2                     * 200
brne loop 2                    * 200
dec  temp1                     * 66
brne Loop2                     * 66
ret                              * 1
Tiempo=
(5+1+66+(200+400)*66+66+132+4)/8000
000= 0,00498seg ~= 5ms
```

```
; Assembly code
; Delay 8 000 000 cycles
; 1s at 8.0 MHz
    ldi  r18, 41
    ldi  r19, 150
    ldi  r20, 128
L1: dec  r20
    brne L1
    dec  r19
    brne L1
    dec  r18
    brne L1
```

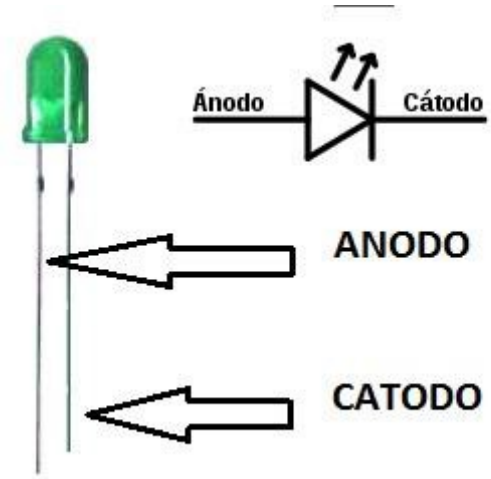
Resumen teórico

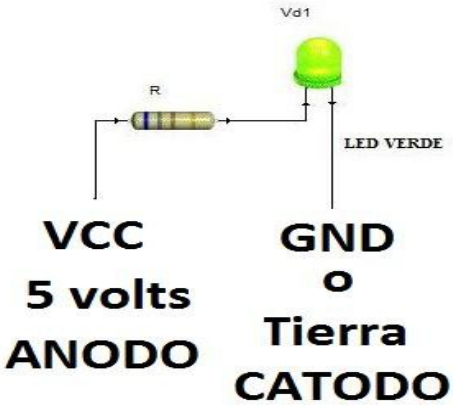
LED

LED acrónimo de Light Emitting Diode o Diodo Emisor de Luz, es un dispositivo semiconductor que emite luz al circular a través de él una corriente eléctrica.

Los LEDs como los diodos normales tienen un ánodo y un cátodo, los cuales se pueden identificar de manera fácil, siendo el ánodo la pata más larga como se observa en la figura.

Parpadeo de un LED





Como se puede apreciar en la figura superior, al LED se le coloca una resistencia en serie para limitar la corriente del mismo, en este caso que se alimentará con 5v se sugieren usar resistencias de 220 Ohms.

Para calcular las resistencias exactas y obtener un desempeño óptimo se realiza a través de la siguiente fórmula:

$$R = \frac{Vcc - V_{LED}}{I_{LED}}$$

Color	Caida de tensión (VLED) V	Intensidad máxima (ILED) mA	Intensidad media (ILED)mA
Rojo	1.6	20	5 – 10
Verde	2.4	20	5 – 10
Amarillo	2.4	20	5 – 10
Naranja	1.7	20	5 – 10

Caída de tensión e intensidad.

Tensión de alimentación, VCC , es el voltaje aplicado al circuito

Caída de tensión del **LED**,Vled es el voltaje necesario para el funcionamiento del **LED**, generalmente está entre 1.7 y 3.3 voltios, depende del color del **diodo** y de la composición de metales.

El microcontrolador tiene varios puertos de los cuales podemos hacer uso, estos puertos los podemos configurar como nosotros queramos, como entrada o como salida, para poder hacer esto es necesario escribir en los registros del puerto para darle las instrucciones necesarias.

Existen tres principales formas de controlar los puertos, tomamos como ejemplo el puerto B.

DDR

Para hacer que el puerto B se comporte como entrada o salida, es necesario setear el DDR, este registro no activará ni desactivará ningún pin del microcontrolador, simplemente le indicará al puerto si este será entrada o salida.

The Port B Data Direction Register – DDRB

Bit	7	6	5	4	3	2	1	0	
	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Para indicarle al DDR si el puerto será de entrada o salida, el 1 indica salida, y el 0 entrada, se le puede escribir como hexadecimal, decimal, o binario, por ejemplo, si queremos que todos los bits del puerto sean salidas lo podemos escribir como sigue:

DDRB=0xFF; //Como Hexadecimal, DDRB=255; //Como Decimal, DDRB=0b11111111; //Como Binario

Si queremos que algunos bits funcionen como entradas:

DDRB=0x8C; //Como Hexadecimal, DDRB=140; //Como Decimal, DDRB=0b10001100; //Como Binario

Se puede escribir en cualquiera de los tres tipos (binario hexadecimal y decimal), en todos los registros.

PORT El PORT controla la salida del puerto, este se usa en caso de que el DDR haya sido seleccionado como salida, un 1 en el PORT indica un nivel alto en el puerto como salida, un 0 indica que el pin estará en nivel bajo. Veamos algunas configuraciones de ejemplo para el PORT:

PORTB=0xFF; Todos los pines están en 1 lógico (high), PORTB=0x00; Todos los pines están en 0 lógico (low)

PORTB=0x03; //Solo los primeros dos bits del puerto están **high**

The Port B Data Register – PORTB

Bit	7	6	5	4	3	2	1	0	
	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PIN

El PIN es un registro de lectura (notar en la imagen del registro donde dice Read/Write, todos son R), este registro nos da un 1 si el pin del microcontrolador tiene una tensión mayor a Vih (limite de tensión de entrada por arriba del cual se considera 1 lógico) , y un cero si el pin presenta una tensión menor a ViL (la tensión de entrada low por debajo de la cual se considera 0 lógico.

En este caso el valor del PIN se le puede asignar a una variable la cual guardará el valor del mismo, al momento de ejecutar la instrucción. ej.

valor=PINB; //El valor de PINB

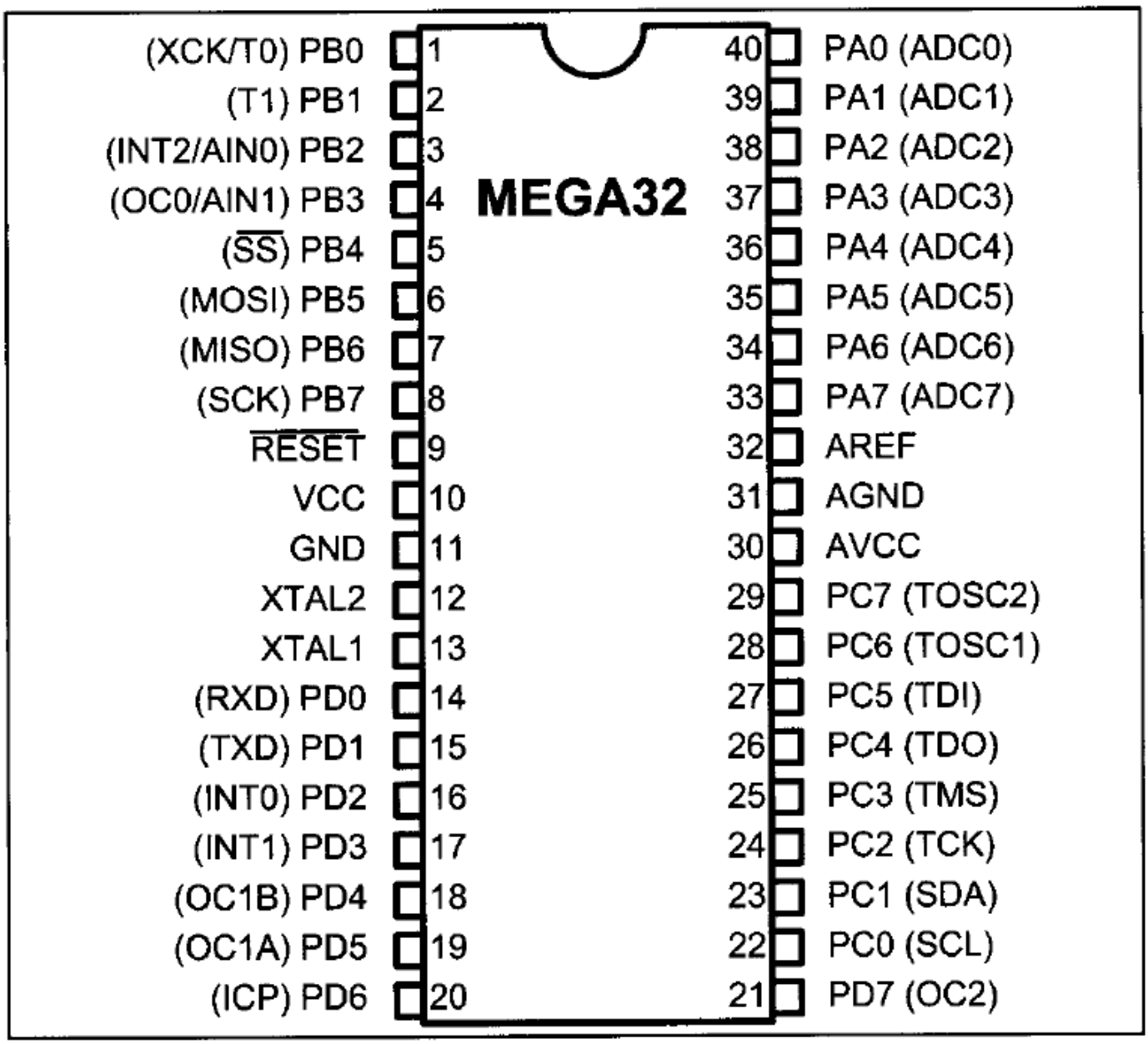
The Port B Input Pins Address – PINB

Bit	7	6	5	4	3	2	1	0	
	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

Aclaración: Cuando el micro resetea, es decir empieza a funcionar, los puertos están seteados como entrada(como valor inicial). DDR valor inicial = 0

Apéndice

Pinout ATmega32



Pinout ATmega328/p

Pinout

Figure 5-1. 28-pin PDIP

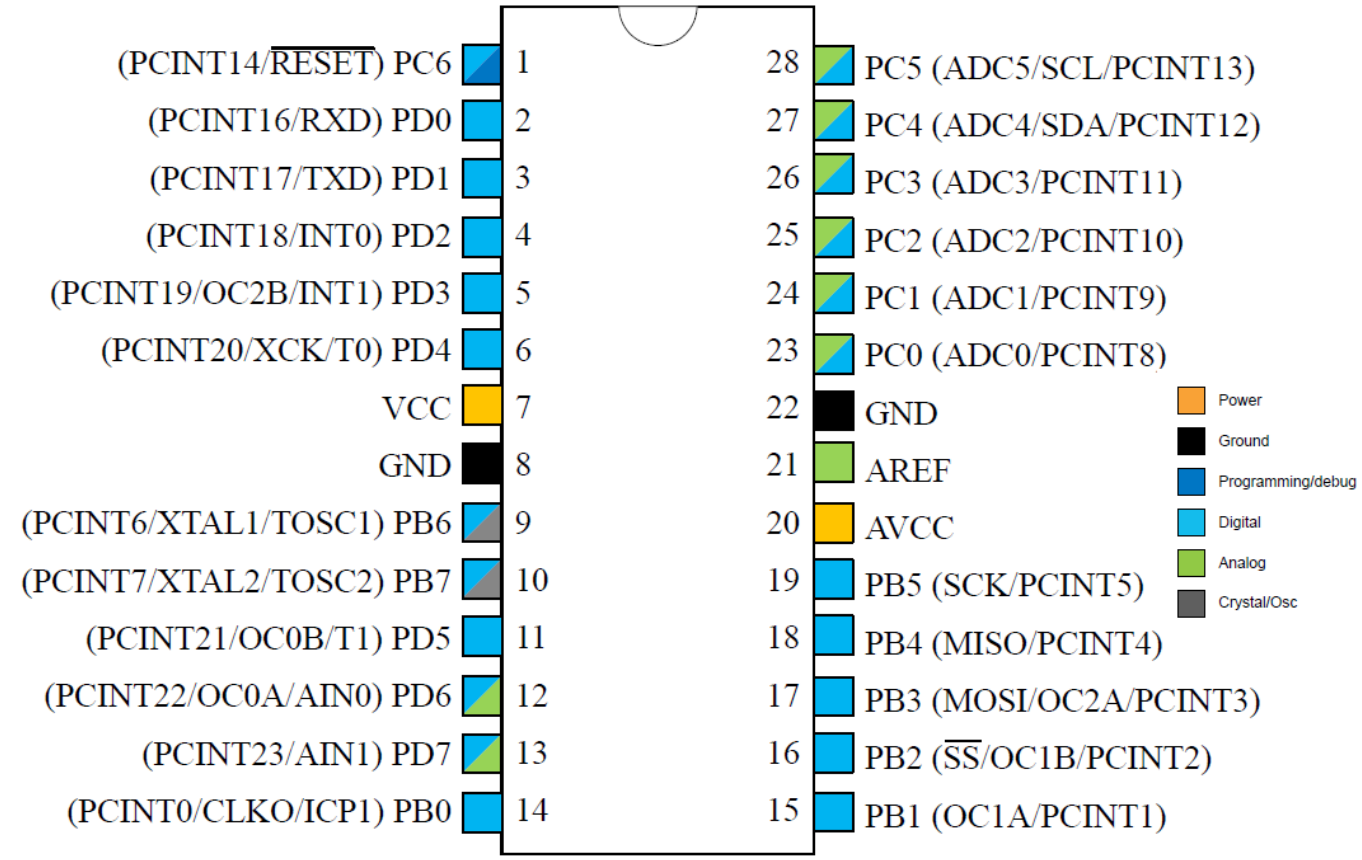


Diagrama de Arduino Uno