



Departamento de Electrónica
Laboratorio de Microprocesadores - 86.07

Trabajo Práctico Obligatorio N°1: Parpadeo de un Led

Profesor:			Ing. Guillermo Campiglio									
Cuatrimestre/Año:			1°/2020									
Turno de las clases prácticas			Miércoles									
Jefe de trabajos prácticos:			Ing. Pedro Ignacio Martos									
Docente guía:			Ing. Fabricio Baglivo, Ing. Fernando Pucci									
Autor			Seguimiento del proyecto									
Maximiliano	Porta	98800										

Observaciones:

.....

.....

.....

.....

.....

.....

.....

Fecha de aprobación			Firma J.T.P		

19 de mayo de 2020

Índice

1. Objetivo	2
2. Descripción	2
2.1. Introducción	2
2.2. Diagrama de Conexión del Circuito	3
2.3. Esquema en Bloques	3
2.4. Materiales Utilizados	4
2.5. Diagrama de Flujo	4
2.6. Lógica del retardo	4
2.7. Links de funcionamiento del circuito:	4
3. Código	5
3.1. Código activando todo el puerto	5
3.2. Código activando solo un pin	6
4. Resultados y Conclusiones	7
4.1. Resultados	7
4.2. Conclusiones	7

1. Objetivo

Consiste en aprender código básico del lenguaje ensamblador, aprender como trabaja internamente el microcontrolador Atmega328p, en mi caso utilizo una plataforma de Arduino Uno, la misma cuenta con el programador y el microcontrolador, en este trabajo practico se busca hacer parpadear un led mediante el uso de puertos y registros del microcontrolador y hacer un retardo entre el tiempo que esta encendido y apagado del mismo led.

2. Descripción

2.1. Introducción

Para este trabajo practico se utilizo el diagrama de la Figura 1. Se observo en la hoja de datos del Arduino la corriente máxima de salida ($I_{Out} = 20mA$), usando las leyes de ohm se obtiene la ecuación 1, con una Resistencia de 220Ω y la tensión para un diodo led verde, se obtuvo que la corriente de salida es $I_{Out} = 10,9mA$, por lo cual es menor a la máxima permitida por pin del arduino.

Mediante la programación en lenguaje assembler se logra controlar un pin de salida del microcontrolador para controlar un circuito externo, en este caso un diodo led con su resistencia.

Además se aprovecho la alimentación por usb y el terminal de gnd del arduino uno para evitar utilizar una fuente externa. Otro ahorro es que al utilizar una placa arduino uno me ahorra de comprar un programador para programar el microcontrolador Atmega328p ya que la placa cuenta con su propio programador.

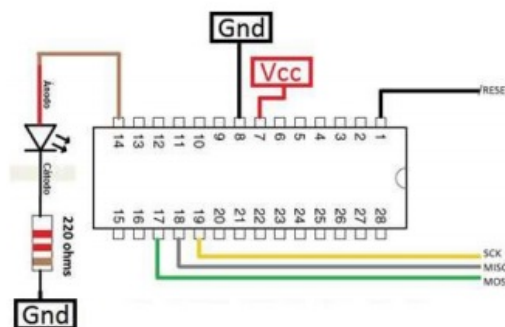


Figura 1: Diagrama de conexión del Diodo Led

Como se observa en la figura 1, este es el diagrama de conexión normal si se dispone del Atmega328p y un programador por separado, pero en mi caso como utilicé una plataforma de arduino uno esta ya tiene todo integrado por lo cual, el diagrama de conexión utilizado se puede apreciar en la figura 2. Esto es mucho mas practico que trabajar con las cosas separadas.

$$I_{Led} = \frac{V_{CC} - V_{Led}}{R_{Led}} \quad (1)$$

2.2. Diagrama de Conexión del Circuito

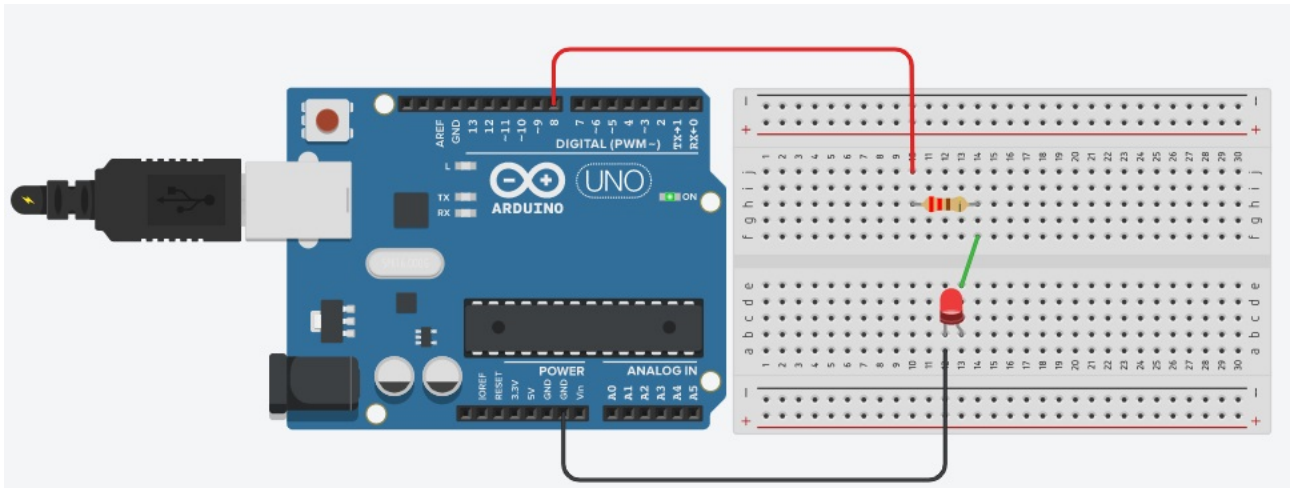


Figura 2: Diagrama de Conexión del Circuito Físico

2.3. Esquema en Bloques

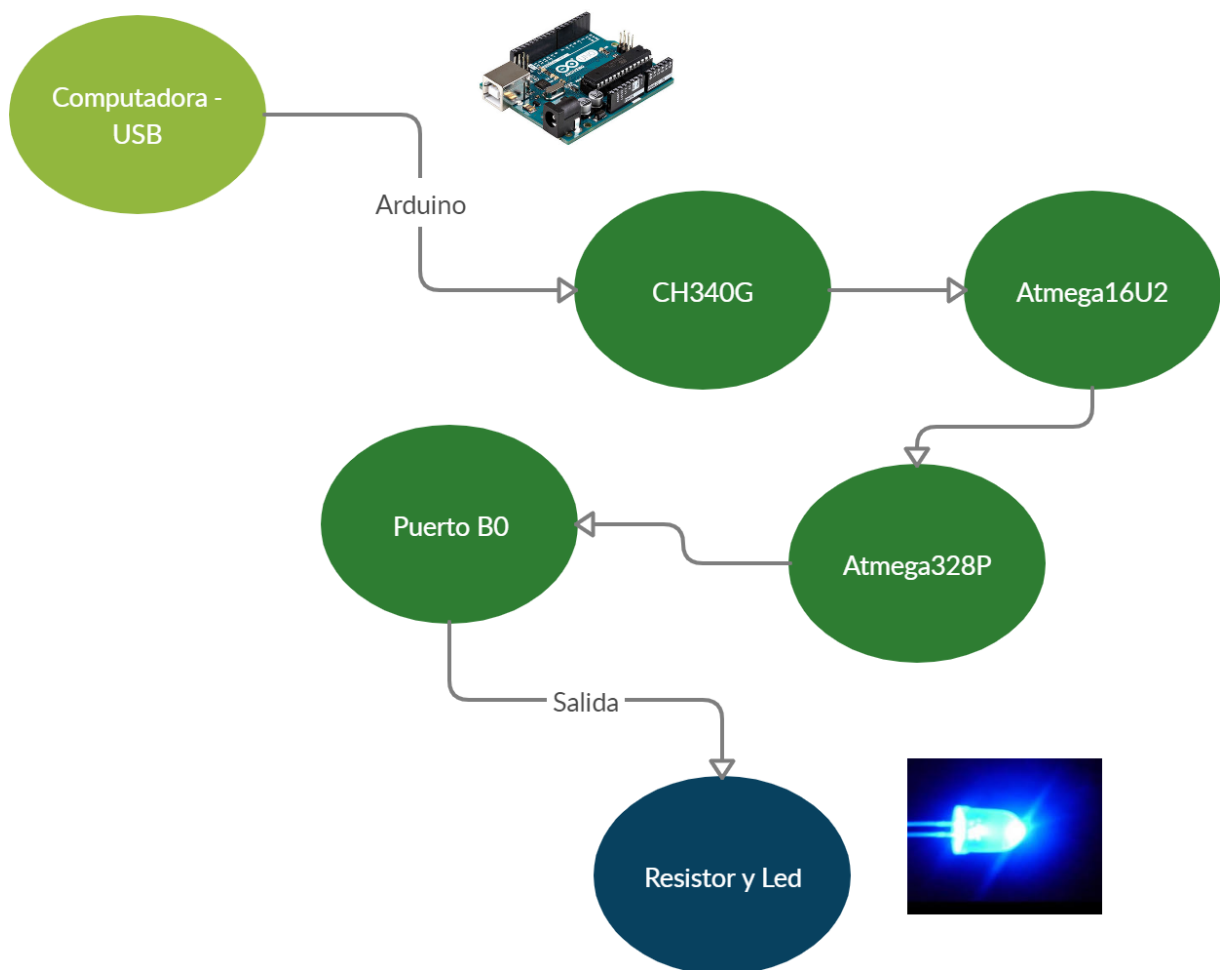


Figura 3: Diagrama en bloques

2.4. Materiales Utilizados

- Diodo Led verde de 3mm (Alta Luminosidad).
- Resistor de 200Ω de carbón 1/4 Watt.
- Placa Arduino Uno con USB.
- Placa Experimental.
- Cables de conexión.

2.5. Diagrama de Flujo

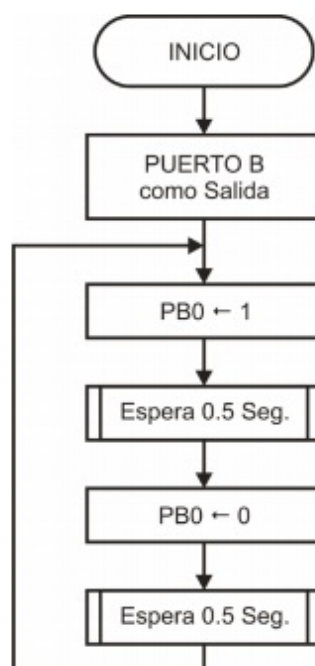


Figura 4: Diagrama de flujo

2.6. Lógica del retardo

Para realizar el retardo entre el tiempo de encendido y apagado del mismo, en vez de aumentar las variables las voy decrementando por que se me hace mas fácil contar, también las escribí en decimal por la misma razón, El microprocesador tiene un frecuencia de 16 Mhz por lo cual hice un bucle como se ve comentado en el código, para obtener un tiempo de espera de medio segundo para que se note bien el parpadeo del led y ver que los cálculos fueran correctos.

2.7. Links de funcionamiento del circuito:

Utilizando solo el pin 0: <https://youtu.be/ZiZ7ANmoCcw>.
Utilizando todo el puerto: https://youtu.be/rFtRU6ik_MA

3. Código

3.1. Código activando todo el puerto

```
; Trabajo_Practico_1.asm
; Encendido de un LED
; Created: 16/5/2020 10:24:57
; Author : Maximiliano Adrian Porta, 98.800
; University : Facultad de Ingenieria, UBA
;

/*Descripcion:
En este programa lo que se realiza es el parpadeo de un led con un periodo
de aproximadamente de 1 segundo, esto se realiza configurando todo los
pines del puerto B como salida y luego se activaran todos los pines mediante
un 1 logico para que el led encienda.
*/

.include "m328pdef.inc"

.cseg ;Escribo el codigo en la Memoria FLASH
.org 0x0000

jmp main

.org INT_VECTORS_SIZE ;Calcula cuanta memoria deja para los perifericos
;Esa cte esta definida en m328pdef.inc

main:

;Configuro el Puerto B del Micro

ldi R16,0xFF ;Carga en el registro 16 todos 1 -> 0b11111111
ldi R17,0x00
out DDRB,R16 ;Configuro todo el puerto B como salida

ciclo_infinito:

out PORTB,R16
rcall tiempo_espera ;Espera el tiempo configurado en esa Bandera
out PORTB,R17
rcall tiempo_espera
rjmp ciclo_infinito ; Vuelve a iniciar este loop infinito

tiempo_espera: ; configuro el tiempo de espera

ldi R20, 32
t3: ldi R19, 250
t2: ldi R18, 250
t1: nop ; hasta aca son 4 ciclos de maquina por iteracion -> 250ns
```

```

dec R18 ; 250 x 250ns = 62.5us
brne t1

dec R19
brne t2 ; 250 x 50us = 15.625ms

dec R20
brne t3 ; 32 x 15.625ms = 500ms
; Con esto logre un retardo de Medio Segundo
ret ; Vuelve a la linea siguiente de la que fue llamado

```

3.2. Código activando solo un pin

```

/*Descripcion: En este codigo se realiza el parpadeo de un led en un periodo
de 1 segundo y configurando un unico pin como salida del puerto b, y activando
con un 1 logico ese pin para que encienda y con un 0 logico para que apague el led
*/
#include "m328pdef.inc"

.cseg      ;Escribo el codigo en la Memoria FLASH
.org 0x0000

jmp main

.org INT_VECTORS_SIZE      ;Calcula cuanta memoria deja para los perifericos
                           ;Esa cte esta definida en m328pdef.inc

main:

;Configuro el Puerto B del Micro

ldi R16,0b00000001 ;Carga en el registro 16 -> 0b00000001
out DDRB,R16      ;Configuro solo el pin 0 del puerto b como salida

ciclo_infinito:

sbi PORTB,0 ;Configura un valor alto en P0 -> Enciende el LED
rcall tiempo_espera ;Espera el tiempo configurado en esa Bandera
cbi PORTB,0 ;Configura un valor bajo en el P0 -> Apaga el LED
rcall tiempo_espera
rjmp ciclo_infinito ;Vuelve a iniciar este loop infinito

tiempo_espera: ;configuro el tiempo de espera clock de 16Mhz

ldi R20, 32
t3: ldi R19, 250
t2: ldi R18, 250
t1: nop ; hasta aca son 4 ciclos de maquina por iteracion -> 250ns

dec R18 ; 250 x 250ns = 62.5us

```

```
brne t1

dec R19
brne t2      ; 250 x 62.5us = 15.625ms

dec R20
brne t3      ; 32 x 15.625us = 500ms
              ; Con esto logre un retardo de Medio Segundo
ret ; Avanza a la linea siguiente de codigo que lo llamo
```

4. Resultados y Conclusiones

4.1. Resultados

Se logro controlar todos los pines del microcontrolador y un pin especifico del mismo para el control del diodo led, se logro a su vez conseguir un retardo deseado decrementando variables. Se logro hacer parpadear al diodo led en la salida del puerto B0.

4.2. Conclusiones

Aprendí a utilizar los programas AtmelStudio y AVRdudess, también los comandos básicos de assembler, como conectar y buscar en que puerto de la computadora esta conectado mi arduino, a cargarle un programa en hexadecimal. Mi arduino es muy chino, por lo cual tuve que adivinar en que puerto usb estaba ya que me figuraba como USB-Serial y no como "arduino uno", ahora lo conecto siempre en el mismo usb que es el puerto de comunicación 3, pude solucionar el problema de que en el AtmelStudio que no me aparecía la ventana de "Build." al abrir un proyecto.