UNIVERSIDAD
Blas Pascal

Juego de luces LED con librería de retardos.

Asignatura: Microprocesadores

Fecha: 18/05/2023

Autor: BAIGORRIA, Constanza
CAMPELLONE, Victoria
GOMEZ, Martina

Profesor: Ing. Ezequiel Giovanardi

RESUMEN

En este informe veremos como es posible realizar una secuencia de luces led, pero de una forma menos estructurada, realizando una variación de luces en cuanto a los retardos.

INDICE:

1. INTRODUCCIÓN

En el siguiente informe, daremos a conocer cómo es que, con el programa MPLAB, logramos crear un código para el encendido y apagado secuencial de 8 leds. Este será de una forma menos estructurada, realizando una variación de luces en cuanto a los retardos.

Al finalizar, el mismo será enviado a un pic16f84a que le habilita la placa RASPBERRY a ejecutar dicho código.

2. DESARROLLO

2.1 MATERIALES

Lo mencionado anteriormente, será posible por el lenguaje ASSEMBLER que otorga MPLAB. Con el uso de este lenguaje, y de los siguientes componentes, podremos realizar esta actividad de la que hablaremos más a detalle.

- MPLAB: Programa que nos permite crear un código en lenguaje Assembler.
- PROTEUS: Programa de simulación para crear la placa con el PIC para corroborar que no se queme con el voltaje seleccionado.
- Un PIC 16f84a: Microcontrolador de 8 bits, 18 pines y un conjunto de instrucciones, de la familia PIC perteneciente a la gama media. Este tipo de PIC puede ser programado tanto en ASSEMBLER como en BASIC.
- Una placa RASPBERRY: Serie de ordenadores de placa simple de bajo costo.
- Leds que utilizar.

2.2 CONSTRUCCION

Al momento de comenzar a crear el código, debemos tener en cuenta que tenemos que guardarlo con la extensión *.ams* ya que de esta forma será posible ejecutarlo correctamente. Una vez creado el archivo con esta extensión, es posible comenzar a crear nuestro código en el lenguaje ASSEMBLER.

Para comenzar a crear el código debemos tener en cuenta que es cada cosa que debemos usar. Como esto es una secuencia de encendido/apagado de leds, necesitamos lo siguiente.

- Un controlador de tiempo.
- Un controlador de voltaje.
- Una función que inicie el programa.
- Un encendido y un apagado.
- Una función que permita que dicho proceso se repita indefinidamente.
- Una librería que nos permita demorar al encendido/apagado de nuestros leds.

3. RESULTADOS

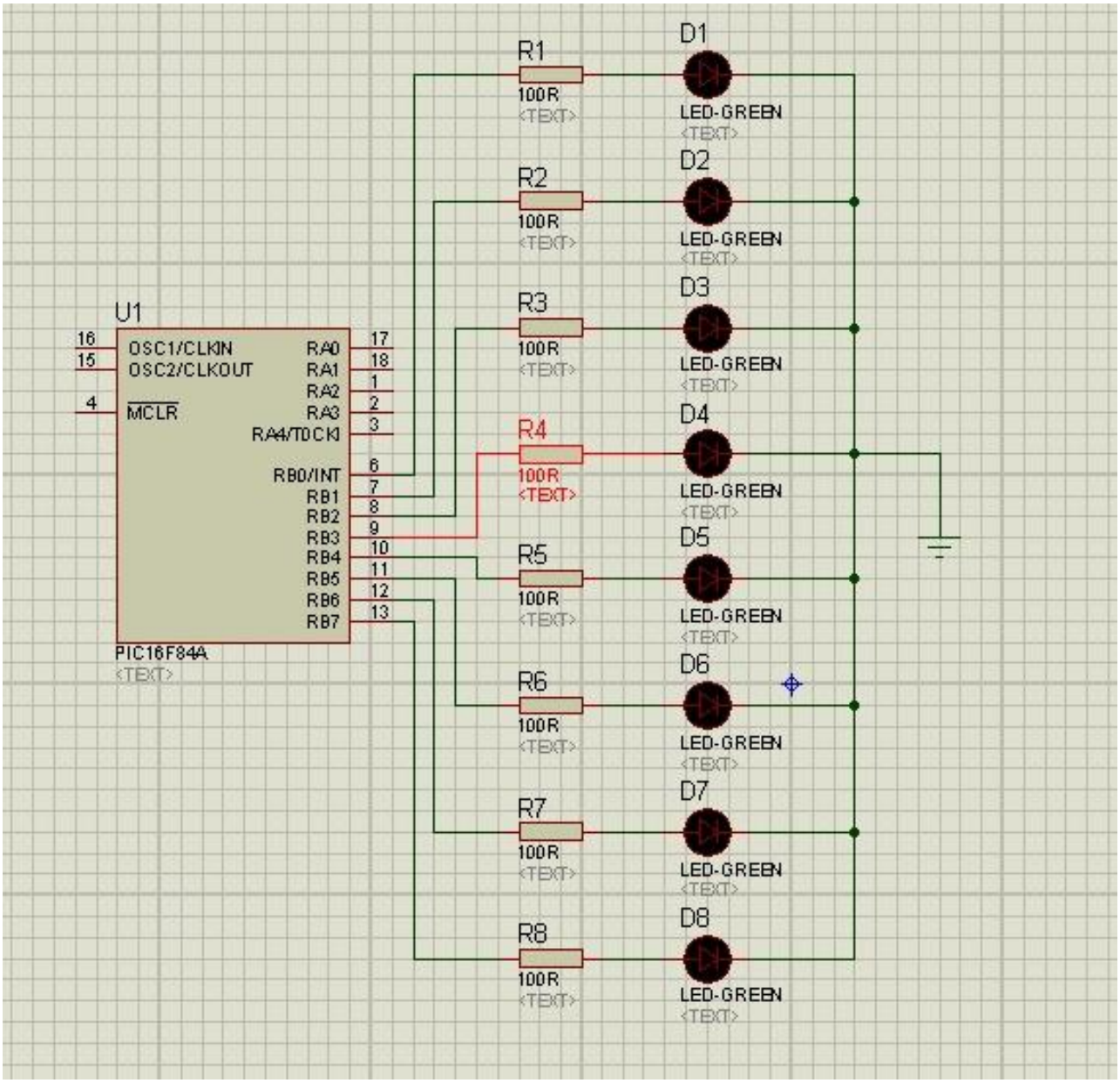
```
list p=16f84
include<P16F84A.inc>
__CONFIG    _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC;
; CONFIGURACION:
;_CP_OFF = Permite que se haga el proceso inverso (desde el pic puedo obtener el codigo)
;_WDT_OFF = Indica si rebaja la pila (si llega al tope de la pila, reinicia el pic)
;_PWRTE_ON = Mide si tiene 5 v el pic (si no lo tiene, no enciende)
;_XT_OSC = Oscilador externo al pic (piedra de cuarzo)

            org     0x00; Define el inicio de la memoria en el 00
            bsf     STATUS,RP0; Pone en 1 el bit 5 (RP0) del registro STATUS (define el output)
            clrf    PORTB; Coloca todos los bits del puerto b en 0
            bcf     STATUS,RP0; Vuelve a 0 el RP0 para trabajar en el banco 1 de la memoria


INICIO
;FUNCION DEL PROGRAMA:  hacer una secuencia de led que encienda del centro hacia afuera
;PRENDE LOS LEDS DE ARRIBA A ABAJO
;LLAMA A LAS SUBRUTINAS (DEFINIDAS AL FINAL) PARA PODER HACERLO
            call PRIMERO
            call BIT2
            call BIT3
            call BIT4
            call BIT5
            call BIT6
            call BIT7
            call ULTIMO
;PRENDE LOS LED DE ABAJO HACIA ARRIBA
;LLAMA A LAS SUBRUTINAS (DEFINIDAS AL FINAL) PARA PODER HACERLO
            call    ULTIMO
            call    BIT7
            call    BIT6
            call    BIT5
            call    BIT4
            call    BIT3
            call    BIT2
            call    PRIMERO
;PRENDE TODOS LOS LEDS
            movlw   B'11111111'
            movwf   PORTB
            call    Retardo_200ms
;APAGA TODOS LOS LEDS
            movlw   B'00000000'
            movwf   PORTB
            call    Retardo_200ms
;PRENDE LOS LEDS IMPARES
;LLAMA A LA SUBRUTINA (DEFINIDA AL FINAL) PARA PODER HACERLO
```

```
            call    IMPARES
;PRENDE LOS LEDS PARES
;LLAMA A LA SUBRUTINA (DEFINIDA AL FINAL) PARA PODER HACERLO
            call    PARES


            goto    INICIC; Genera el loop para que se repita indefinidamente el programa.

PRIMERO     movlw   B'00000001'; Le coloca los valores binarios al registro w
;(los 1 son led encendidos/los 0 led apagados)
            movwf   PORTB; Prende el bit 7
            call    Retardo_200ms; Llama a la subrutina Retardo_200ms de la libreria retardos.inc
            return;da final a la subrutina
            ;call   Retardo_50ms
            ;call   Retardo_50ms

BIT2        movlw   B'00000010'; Le coloca los valores binarios al registro w
;(los 1 son led encendidos/los 0 led apagados)
            movwf   PORTB; Prende el led 6
            call    Retardo_200ms;
            return

BIT3        movlw   B'00000100'
            movwf   PORTB; Prende el led 5
            call    Retardo_200ms
            return

BIT4        movlw   B'00001000'
            movwf   PORTB;Prende el led 4
            call    Retardo_200ms
            return

BIT5        movlw   B'00010000'
            movwf   PORTB; Prende el led 3
            call    Retardo_200ms
            return

BIT6        movlw   B'00100000'
            movwf   PORTB;Prende el led 2
            call    Retardo_200ms
            return

BIT7        movlw   B'01000000'
            movwf   PORTB;Prende el led 1
            call    Retardo_200ms


BIT6        movlw   B'00100000'
            movwf   PORTB;Prende el led 2
            call    Retardo_200ms
            return

BIT7        movlw   B'01000000'
            movwf   PORTB;Prende el led 1
            call    Retardo_200ms
            return

ULTIMO      movlw   B'10000000'
            movwf   PORTB; Prende el led 0
            call    Retardo_200ms
            return

IMPARES     movlw   B'01010101'
            movwf   PORTB; Prende los led 1/3/5/7
            call    Retardo_200ms
            return

PARES       movlw   B'10101010'
            movwf   PORTB ;Prende los led 0/2/4/6/8
            call    Retardo_200ms
            return

include<retardos.inc>

END
```

Lo que obtenemos con nuestro código es una secuencia de encendido y apagado de las luces led de distintas maneras. El programa se encargará de realizar ese juego de luces de forma ascendente, los leds irán encendiendo y apagando de arriba abajo; así como de forma descendente.

Luego, se realizarán diversas pruebas donde se enciendan la mayoría de las luces, aquellas que estén en una posición par, y, por último, leds que se encuentren en una posición impar.

Así, conseguimos un juego de luces con la siguiente conexión de leds.



4. CONCLUSIONES

Logramos ver cómo es posible realizar una secuencia de encendido/apagado de 8 leds conectados a una placa RASPBERRY con un pic16f84a. También debemos tener en cuenta que, a nuestro PIC, debemos conectarlo correctamente a la placa,
No podemos olvidarnos de controlar el voltaje de nuestra placa para que no se quemen los leds y tampoco el PIC.

Con esto aclarado, obtenemos un buen funcionamiento de todo el sistema, y logramos la finalidad del programa.