



PROYECTO #1
“Propeller Clock (POV)”

Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Universidad del Valle de Guatemala
Microcontroladores aplicados a la industria
Max de León Robles – 13012
Guatemala, 11 de Marzo del 2016

I. Objetivos

A. General

Aplicar los conocimientos aprendidos a lo largo del curso para programar en lenguaje ensamblador un PIC16F84A para realizar un propeller clock (POV) desplegando de manera adecuada la hora deseada.

B. Específicos

- Diseñar el circuito esquemático y luego la placa que será fresada para soldar ahí los componentes necesarios para dicho proyecto.
- Utilizar un fototransistor y un diodo de luz infrarroja para controlar que el reloj se mantenga constante en un lugar sin movimiento oscilante en el mismo.
- Aplicar el concepto de “slip ring” para colocar los botones con los que se cambia la hora afuera de la estructura y así hacer más eficiente y funcional el reloj.

II. Procedimiento y materiales

Para dicho proyecto se siguió el diseño del circuito esquemático presentado en la figura 1 tomando en cuenta que los LED's utilizados para dibujar la hora están todos en el PORTB y los botones de entrada y el fototransistor se encuentran en el PORTA, para ello se programaron las interrupciones en lugar de utilizar las que existen por defecto en el PIC.

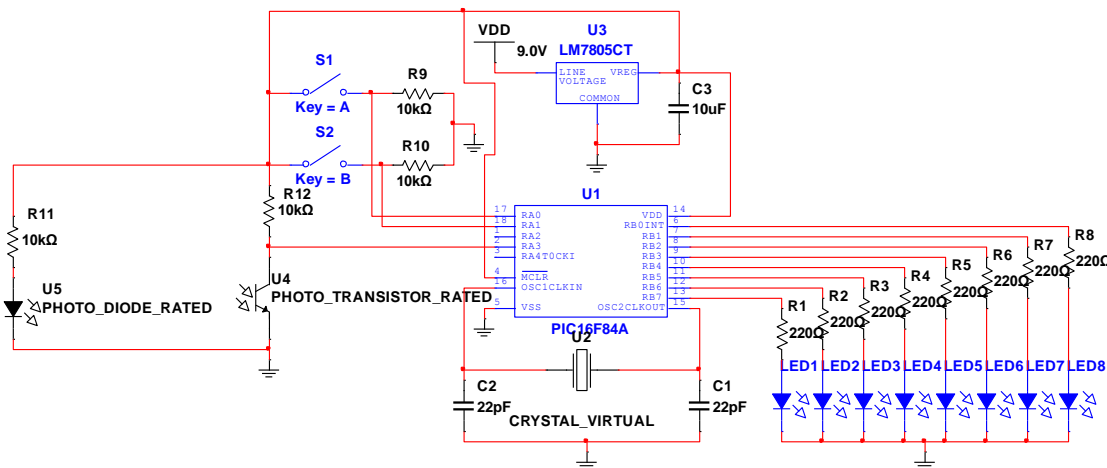


Fig. 1 – Diseño esquemático del circuito utilizado para el Propeller Clock

Los materiales utilizados para dicho proyecto son los siguientes:

- 1 – PIC16F84A
- 1 – Cristal (8 MHz)
- 2 – Capacitores 22pF
- 1 – Capacitores 10 µF
- 1 – Capacitores 100 µF
- 1 – LM7805 (Regulador de voltaje)

- 8 – Resistencias 220 Ω
- 3 – Resistencias 10 K Ω
- 2 – Resistencias 1 K Ω
- 3 – Dip switches (Botones)
- 8 – Led
- 1 – Led infrarrojo
- 1 – Fototransistor
- 1 – Batería 9 V

Para colocar todos los componentes necesarios para el proyecto se diseñaron dos placas que fueron fresadas, en la figura 2 y figura 3 se pueden observar ambos diseños, se respetaron las normas establecidas para el diseño así como se buscó el minimizar el espacio del mismo para reducir peso de todo el sistema y que el motor pudiera funcionar de manera eficiente y sin problemas.

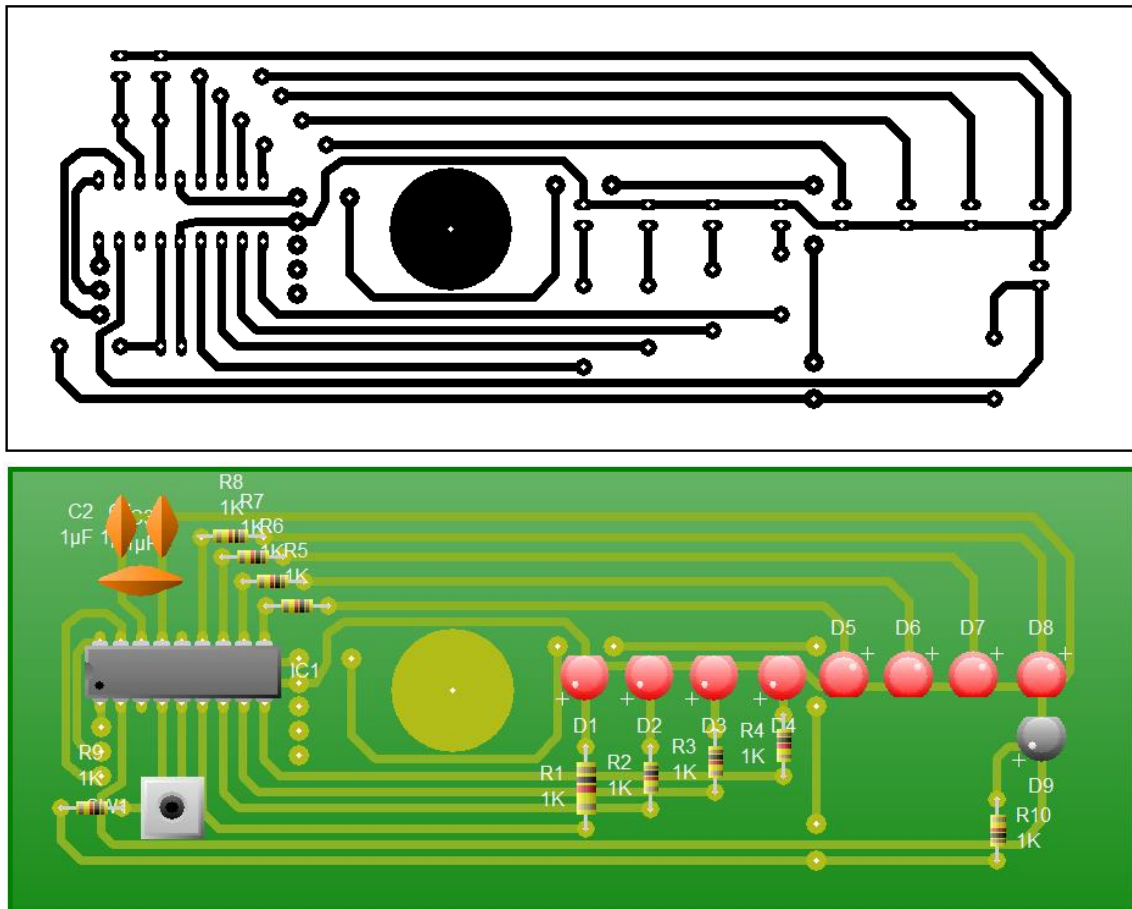


Fig. 2 – Diseño placa circuito Propeller Clock, arriba vista de la placa fresada y abajo vista real de la placa.

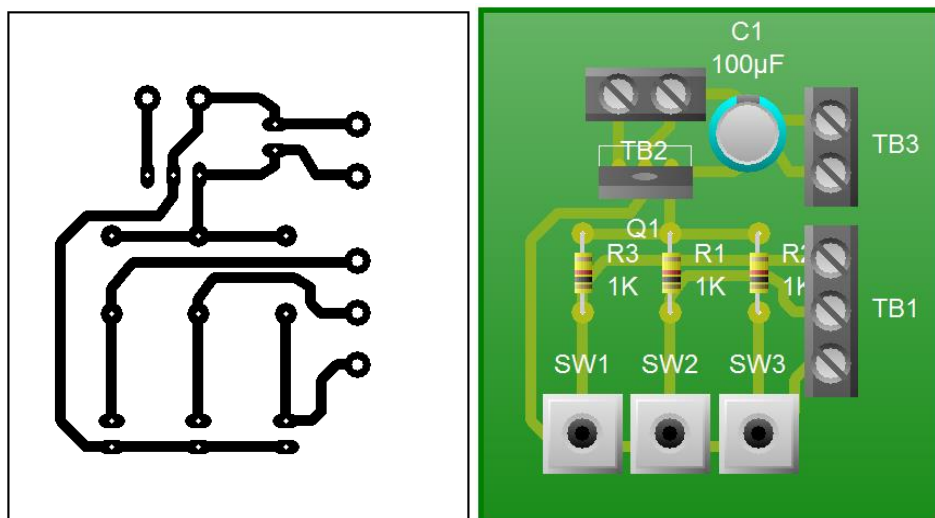


Fig. 3 – Diseño segunda placa donde se ubican los botones y el regulador de voltaje con sus respectivas salidas, a la izquierda el diseño de la placa fresada y a la derecha vista real de la placa.

Se programó en lenguaje ensamblador con los conceptos aprendidos en clase, para llevar el control del tiempo se utilizó la interrupción por TMR0 utilizando como valor inicial del prescaler 248, valor calculado utilizando la herramienta Pictimercalculator, con el que se obtiene un desbordamiento de aproximadamente 1 mS, este ciclo se repitió 100 veces y luego esto 10 veces más, para un total de 1000 repeticiones sumando así un segundo aproximadamente.

Luego de realizar la lógica para llevar el control del tiempo y separar en unidades y decenas las horas, minutos y segundos del reloj se empezó a trabajar en el despliegue de los números utilizando como referencia el orden de los pines utilizados del PORTB como se puede observar en la figura 4 algunos diseños de lo que se presenta en pantalla con el orden de los pines utilizados.

PORTB	CERO	UNO	DOS	TRES	CUATRO
RB0					
RB1					
RB2					
RB3					
RB4					
RB5					
RB6					
RB7					
PORTB	CINCO	SEIS	SIETE	OCHO	NUEVE
RB0					
RB1					
RB2					
RB3					
RB4					
RB5					
RB6					
RB7					

Fig. 4 – Diseño de números para desplegar en reloj con los pines que se utilizaron

Para mantener constante los números y animación del reloj se implementó un fototransistor que recibe la luz infrarroja de un LED y existe una interrupción a cada vuelta

que se utiliza para que el programa solo imprima a partir de ese momento una vez y de esa manera lograr que todo se mantenga en su lugar sin ningún movimiento.

A. Algoritmo narrativo para control del tiempo:

- Revisa si existe desbordamiento del T0IF, si no es así sale de la interrupción.
- Resta uno a cont1, si este aún no es cero sale de la interrupción.
- Si cont1 llega a cero se resta uno a cont2 y reinicia el valor de cont1, si este aún no es cero sale de la interrupción.
- Si cont2 llega a cero se reinicia el valor de cont1 y cont2 y aumenta en uno el valor de unidad segundos.
- Revisa el valor de unidad segundos si es mayor a 9 entonces se regresa a cero y aumenta en uno el valor de decena de segundo, luego se revisa si este valor es mayor a 5 entonces se aumenta en uno el valor de unidad de minutos y se regresa a cero el valor de decena de segundos, de igual manera con decena de minutos, unidad de horas y decenas de horas.

B. Algoritmo narrativo para comparación de números:

- Se carga el valor del número que se va a imprimir a una variable alterna.
- Se incrementa en uno el valor de dicha variable
- Se resta uno y se compara si este valor es cero salta al ciclo del número cero si no sigue a la siguiente revisión que realiza el mismo proceso de restar y comprobar sucesivamente hasta llegar al valor de nueve.

C. Algoritmo narrativo para implementación de botones en PORTA:

- Se crea una bandera para revisión del botón, se inicializa en cero.
- Cada desbordamiento de 100 mS se coloca en uno.
- Se revisa si la bandera es uno si es así salta a revisión de botón.
- Se revisa que pin del PORTA fue activado.
- Salta al ciclo dependiendo del botón presionado y se colocan las banderas de nuevo en cero.
- Si ningún botón fue presionado (ningún pin de PORTA activado) limpia las banderas y sale de la revisión.

D. Algoritmo narrativo para despliegue de números:

- Luego de determinar el número de despliegue salta a subrutina de cada número.
- Carga a PORTB el número en binario respetando el orden de pines de despliegue.
- Se realiza un delay de 100 mS para despliegue y sigue a la siguiente línea, se repite el proceso hasta desplegar todas las líneas.
- Se carga a PORTB '0000000' para que se apaguen los LED's y dejar un espacio entre números.

III. Resultados

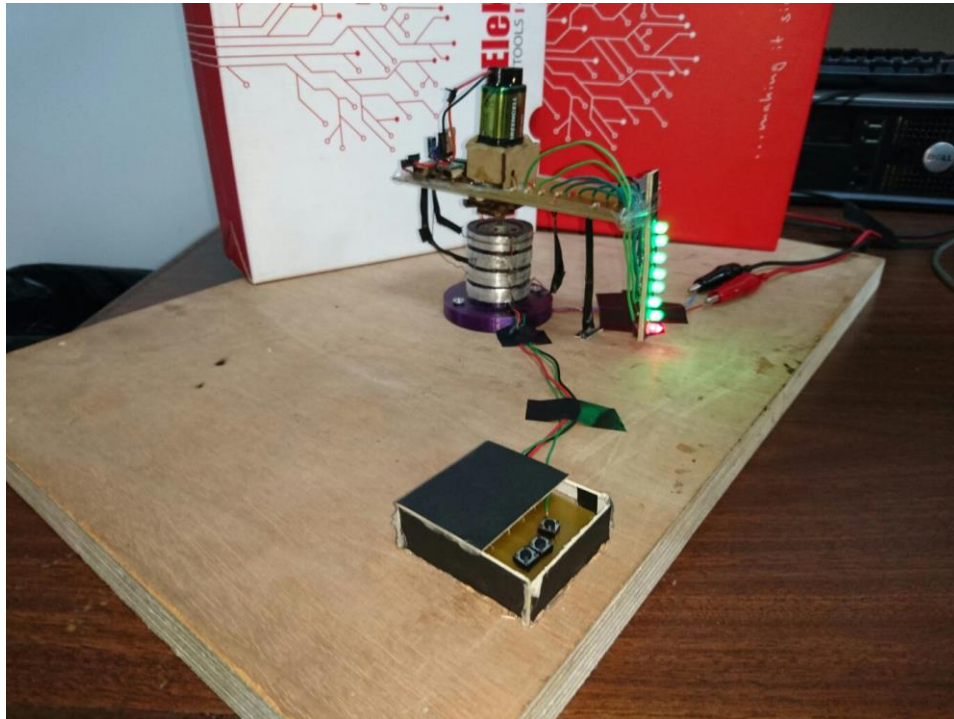


Fig. 6 – Fototransistor y LED infrarrojo montado, se utiliza una pestaña para que suceda la interrupción.

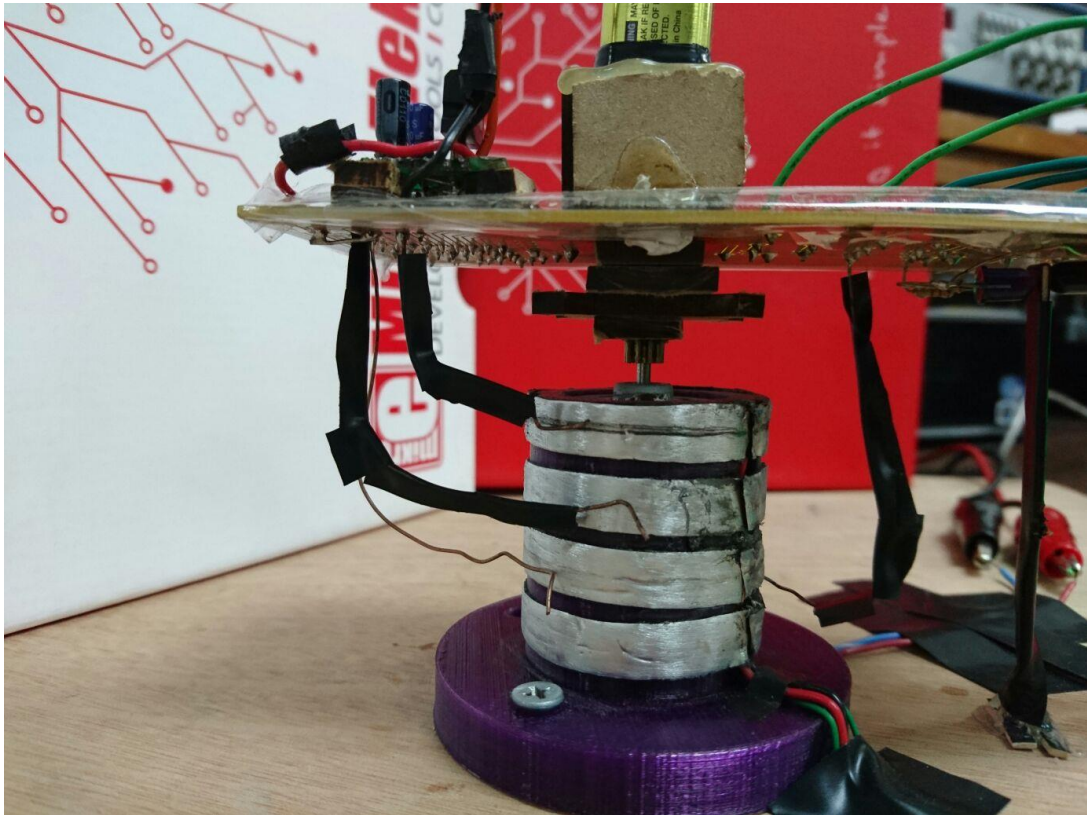


Fig. 7 – Slip ring armado y funcionando se utilizó cinta tapagoteras y como escobilla cable grueso de cobre.



Fig. 8 – Vista aérea de la placa

Link al video: <https://youtu.be/KWkWABOSdrs>