SIMULACIÓN DE UNA INTERFAZ PARA EL CONTROL DE PROYECTORES PLANETARIOS ZKP-2

Guillermo Álvarez Bestard[1] *quille@icmf.inf.cu*
Miguel A. Machirán Simón[1] *machiran@icmf.inf.cu*
Rafael Acosta Julián[1] *rafael@icmf.inf.cu*

RESUMEN

Se aborda la simulación del funcionamiento de una interfaz basada en un microcontrolador PIC, que se encarga de controlar un Proyector Planetario. Se describen las herramientas empleadas para el diseño y la simulación de los esquemas eléctricos, así como las características básicas de la interfaz. Se selecciona el MPLAB y el Proteus, para la programación del microcontrolador y la simulación del circuito. Se muestran los resultados obtenidos en la simulación y que contemplan la ejecución de dos secuencias de operación, el control de la iluminación y la comunicación serie. Se anexan las secuencias de operación que se utilizan para probar la interfaz y los esquemas eléctricos de la misma.

Palabras Clave: Simulación, microcontroladores PIC, control secuencial, Proyector planetario.

ABSTRACT

This work presents the simulation of an interface based in PIC microcontroller, which takes charge of controlling a Planetary Projector. The paper describes the tools for design and simulation, and the basic characteristics of the interface. The microcontroller programming and circuit simulation is made with MPLAB and Proteus. The simulation results are shown, including the execution of two operation sequences, the control of the illumination and series communication. The operation sequences and schemes are annexed.

Key Words: Simulation***,*** PIC microcontrollers, sequential control, Planetarium projector.

[1] Instituto de Cibernética, Matemática y Física (ICIMAF), AENTA-CITMA, Cuba

1. INTRODUCCIÓN

El diseño de circuitos electrónicos ha sido un reto desde el surgimiento mismo de esta disciplina. Tanto los desarrollos analógicos como los digitales, requieren de la evaluación de numerosos parámetros, que muchas veces no se conocen con exactitud y se asumen. Es por ello que en el éxito de cualquier desarrollo, en este campo, está muy presente la experiencia y el cuidado de sus diseñadores.

Con el surgimiento de las computadoras se abrió una nueva era para estos especialistas: ya podían simular el funcionamiento de sus diseños y conocer con alguna veracidad si estos eran correctos. Pero el uso de los simuladores de circuitos electrónicos, en sus inicios, estaba limitado a unos pocos usuarios y a escasas aplicaciones. Esto era debido a que la computadora era un instrumento caro, escaso, requería gran especialización y el desarrollo o adquisición de estos simuladores no era habitual.

Estos programas se especializaban en la simulación analógica o digital, en muchos casos no era posible llevar las dos en un mismo diseño. Tampoco tenían interfaces gráficas amigables, y la preparación y revisión del diseño podía llevar mucho más tiempo que el propio análisis de los resultados. En la mayoría de los casos, el esquema eléctrico debía transferirse a una lista de nodos y de componentes, cuyo modelo físico debía introducir el usuario (1).

Ya en la era en que la computadora pasa a ser una herramienta más en la "maleta" del técnico, existen numerosos programas de simulación (2) (3) (4) (5) (6) (7) (8) (9) (10) (11). La mayoría permite un diseño gráfico de los circuitos y cuentan con variadas bibliotecas de componentes, organizadas por tipos y fabricantes. Los desarrolladores de simuladores establecen un vínculo con los fabricantes de componentes, manteniendo así actualizadas las bibliotecas.

Adicionalmente ofrecen posibilidades de simulación interactiva (8)(6), donde el usuario puede modificar el estado de interruptores, selectores y potenciómetros. Puede observarse el estado y el valor de las salidas, y realizan animaciones de elementos como motores y relés, incluso con modelos tridimensionales (6). Ofrecen una gama amplia de instrumentos de medición como voltímetros, amperímetros, frecuencímetros, contadores, osciloscopios y analizadores lógicos. Igualmente permiten la generación de señales con diversas formas de ondas o patrones. Todo esto emplea instrumentación virtual que garantiza que el técnico se encuentre familiarizado con este ambiente.

El objetivo de este trabajo es precisamente la simulación de una interfaz empleada para el control de proyectores planetarios, específicamente el modelo ZKP-2 de la Carl Zeiss. Esta emplea un microcontrolador como elemento de control y la componen varias tarjetas electrónicas.

Un Proyector Planetario es un equipo ubicado en una instalación dedicada a la presentación de espectáculos astronómicos. Muestran proyecciones del cielo nocturno, observadas desde diversos lugares de la Tierra y en diferentes momentos en el tiempo. Consta de una pantalla de proyección en forma de cúpula o semiesfera, y de un proyector cuya movilidad permite representar diversos elementos astronómicos. A su alrededor se disponen asientos que pueden acomodar a los espectadores como en un teatro (12).

Estos sistemas constituyen un medio de difundir la cultura a grupos heterogéneos de la sociedad, y por su interesante espectáculo consiguen atraer la atención de niños, jóvenes y adultos. Se emplean además, en la enseñanza y en investigaciones de diversas ramas de la ciencia y la técnica (13). Nuestra interfaz pretende contribuir al aumento de las prestaciones del proyector.

2. MÉTODOS Y HERRAMIENTAS

En el trabajo se emplean un conjunto de herramientas para el diseño de los circuitos y la simulación de los mismos. También se describirán otras utilidades para la configuración del sistema.

Los métodos de validación empleados se basan en la comparación del funcionamiento en simulación con el esperado según los parámetros obtenidos mediante estas herramientas.

2.1. Qué comprobar en simulación.

Mediante la simulación se debe comprobar el funcionamiento básico del sistema, tanto a nivel de registros como en puntos escogidos apropiadamente en el esquema eléctrico (ver anexo 9).

Primeramente se realiza la comprobación a nivel de registros, empleando MPLAB (ver epígrafe 2.2), y se verifica paso a paso o con puntos de ruptura cada operación dentro del microcontrolador. La simulación de los componentes de hardware externos y de algunos internos no es posible con este simulador.

Posteriormente se utiliza Proteus para comprobar el funcionamiento de la mayor parte del esquema eléctrico incluyendo el microcontrolador. En este simulador el acceso a los registros y su modificación es menos factible que en el MPLAB.

También enlazaremos el MPLAB con el Proteus para manipular los registros y los componentes externos de microcontrolador, y observar la repuesta del circuito en su conjunto.

Este proceso se repetirá hasta obtener el comportamiento deseado y se comprobarán específicamente las tareas de:

- Reloj de Tiempo Real.
- Funcionamiento de los pulsadores Play y Stop.
- Ejecución de las secuencias de comandos almacenados en memoria Flash.
- Detección del cruce por cero de la señal de alimentación.
- Control del nivel de iluminación.
- Ejecución de las secuencias de Amanecer y de Atardecer.
- Funcionamiento de la comunicación serie RS232.

2.1.1. Reloj de Tiempo Real.

Se emplea un cristal de 38.768KHz conectado a las entradas T1OSO y T1OSI del Timer1 (ver anexo 9) para generar una interrupción cada 1 segundo. En la subrutina de atención a esta interrupción se incrementa consecuentemente el tiempo y la fecha, así como el día de la semana. Se chequean además si hay alguna alarma programada para comenzar a ejecutar la secuencia de operación correspondiente.

El chequeo de su funcionamiento se puede realizar completamente en Proteus VSM MPALB Driver, a través de los registros asociados a este subsistema.

2.1.2. Pulsadores Play y Stop.

Al pulsar o liberar cualquiera de estos interruptores se provoca un cambio en la entrada digital del microcontrolador (ver anexo 9). La acción correspondiente no se ejecutará hasta que sea liberado, o sea cuando ocurra un cambio de 0 a 1 en la entrada digital.

Esto se puede comprobar en MPLAB mediante puntos de ruptura y el generador de estímulos asincrónicos con que cuenta este simulador.

En Proteus se puede observar si al pulsar y liberar los interruptores se ejecuta la acción correspondiente (ejecución de secuencia, encendido del Led asociado, etc.) También es posible colocar puntos de ruptura en el programa.

2.1.3. Secuencias de comandos.

La interfaz puede almacenar hasta 11 secuencias de operación en los casi 4KWord (14 bits por Word) destinados

para ello en memoria Flash. Mediante estas secuencias se podrán representar los espectáculos en el proyector planetario, ya que están formadas por una serie de comandos que definen estados de las salidas digitales.

Para el diseño de las conferencias o espectáculos se desarrolló un sencillo lenguaje de programación del tipo Lista de Instrucciones, fácil de utilizar por operadores del planetario y por los especialistas en Astronomía. La sintaxis utilizada se adecúa al proyector, por ejemplo: SOL=1 enciende el proyector del Sol, CAZUL=50 entrega el 50% de la potencia a las lámparas azules y ESPERA=5 mantiene el último estado durante 5 segundos (ver anexo 1). Para un uso más general de la interfaz se definió un comando que modifica cualquier salida digital, por ejemplo SD4=1 activa la salida 4.

También se desarrolló un compilador que convierte la lista de instrucciones almacenada en un fichero texto, en un código más eficiente y que el firmware de la interfaz puede interpretar. Detecta y alerta de errores en la programación, como pueden ser estados repetidos o inoperables, o secuencias muy largas. Transfiere las secuencias a la interfaz, muestra el uso de memoria y crea un fichero para su simulación en MPLAB y Proteus.

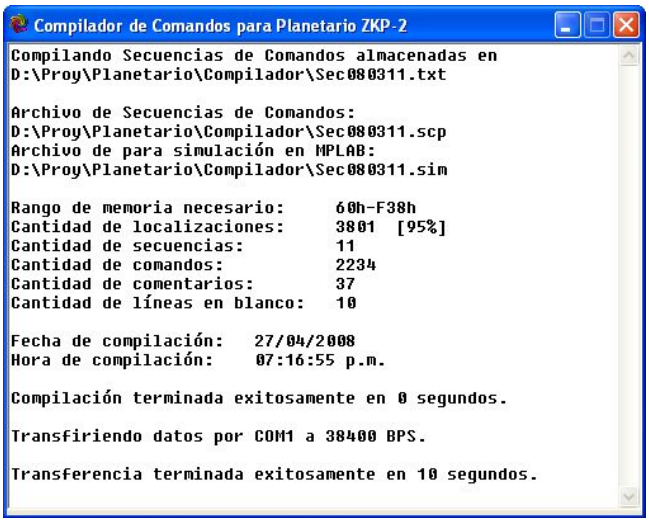


Fig. 1 Compilador de secuencias de comandos.

La correcta ejecución de las secuencias de comandos se verificará en Proteus. Para ello se definió una secuencia especial (ver anexo 2 secuencia 1) que activará cada 1 segundo una salida digital y después las desactivará en orden inverso.

2.1.4. Detección del cruce por cero de la señal de alimentación.

Nuestro esquema de regulación requiere conocer el instante exacto del cruce por cero de la señal de alimentación, para controlar el ángulo de disparo de los tiristores que manejan las lámparas que representan el amanecer y el atardecer (en anexo 9 ver T0, T1 y T2). Para ello se emplea un optoacoplador con entrada bipolar y salida a transistor. Este colocará un 1 lógico en la entrada RB0/INT del microcontrolador cuando ocurra el evento. De esta manera se generará una interrupción que ejecutará las rutinas de control correspondientes.

También es posible detectar la ausencia de la señal de corriente alterna que alimenta a la interfaz, si no se genera la interrupción de cruce por cero durante un tiempo predefinido, y tomar las medidas correspondientes para pasar al modo de bajo consumo o sleep. Para ello se inicializa un contador cada vez que se detecta el cruce por cero, se emplea otro temporizador (TMR2) para decrementarlo y determinar la ausencia de alimentación.

Esta función será comprobada en Proteus VSM MPALB Driver para poder inyectar la señal de 120Hz correspondiente a la alimentación rectificada con onda completa.

2.1.5. Comunicación Serie.

La comunicación serie se empleará para transferir las secuencias de comandos a la memoria Flash del microcontrolador, la secuencia de amanecer a la EEPROM y para la supervisión y control del funcionamiento de la interfaz.

Su comprobación se puede realizar direc-

tamente con el Terminal Virtual de Proteus o en MPLAB con los ficheros de comandos para el UART (menú Debugger/Settings/Uart1).

La simulación en MPLAB requiere de un punto de ruptura en la interrupción de atención a la recepción serie y de modificar los registros adecuados. Es una comprobación de bajo nivel que se realizó en la etapa de programación.

En esta etapa del proyecto es suficiente con la simulación en Proteus y del empleo de su Terminal Virtual para enviar y recibir los datos. Se pueden verificar también los cambios, en los registros afectados y en las salidas digitales correspondientes.

2.1.6. Control del nivel de iluminación.

Para el control de la iluminación de las estrellas y de las luces azules y blancas que alumbran la cúpula, se gobierna el ángulo de disparo de los tiristores que manipulan estas cargas. Este es un método ampliamente utilizado para controlar cargas alimentadas con corriente alterna (motores, resistencias, lámparas, etc.) En nuestro caso la señal de alimentación es alterna rectificada con onda completa, lo cual garantiza el paso de los tiristores al estado de no conducción cuando esta señal se acerca al valor cero (14).

Como el ángulo de disparo no es una magnitud que represente de forma lineal la potencia entregada al filamento de la lámpara, se crearon una serie de herramientas y algoritmos de control que facilitaran mediante un valor entre 0 y 100% definir la potencia entregada a la lámpara y por tanto su nivel de iluminación. Algo análogo al porciento de ciclo útil en los moduladores de ancho de pulso (PWM) empleados en los sistemas de corriente directa.

La herramienta CUAC (ver figura 2) es un programa que calcula, a partir de la frecuencia de la línea de alimentación (60Hz) y la resolución deseada para el control (cantidad de divisiones), los tiempos de disparo necesarios para cantidades iguales de potencia. Para ello se asume una carga constante. Genera además una gráfica con esta distribución de tiempos (ver figura 4).

En un segundo paso se crea el código necesario para que el microcontrolador genere las señales de disparo de los tiristores (ver figura 3). La herramienta calcula el valor que debe tener un contador para esperar el tiempo necesario para el disparo de cada tiristor (ver columna 2 y 4 de la figura 3).

Dentro del microcontrolador se emplea un temporizador (TRM0), ajustado al período recomendado por la herramienta antes mencionada (32.55μs), que generará una interrupción con ese período. De esta forma se cubren los 256 valores posibles de disparo (8.33ms)

En la subrutina de atención a esta interrupción se incrementa un contador (desde 0 hasta 255) y se compara su valor con el valor definido en la tabla para ese nivel de iluminación. Si coincide se activa el tiristor mediante la salida digital correspondiente (T0, T1 o T2). Se emplean 3 variables para almacenar y comparar el nivel de iluminación deseado para las tres salidas a tiristor. Todas las señales de disparo se desactivan con el cruce por cero de la alimentación.

El funcionamiento de estos algoritmos se puede comprobar con un osciloscopio virtual en Proteus y empleando los comandos de comunicación serie adecuados.

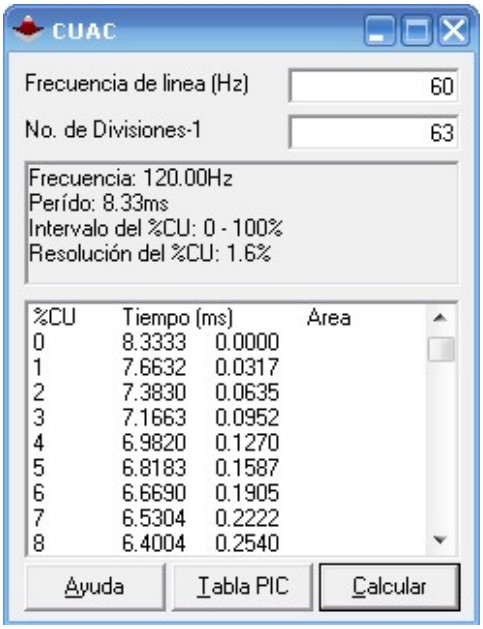


Fig. 2 Herramienta para el cálculo del tiempo de disparo de tiristores.

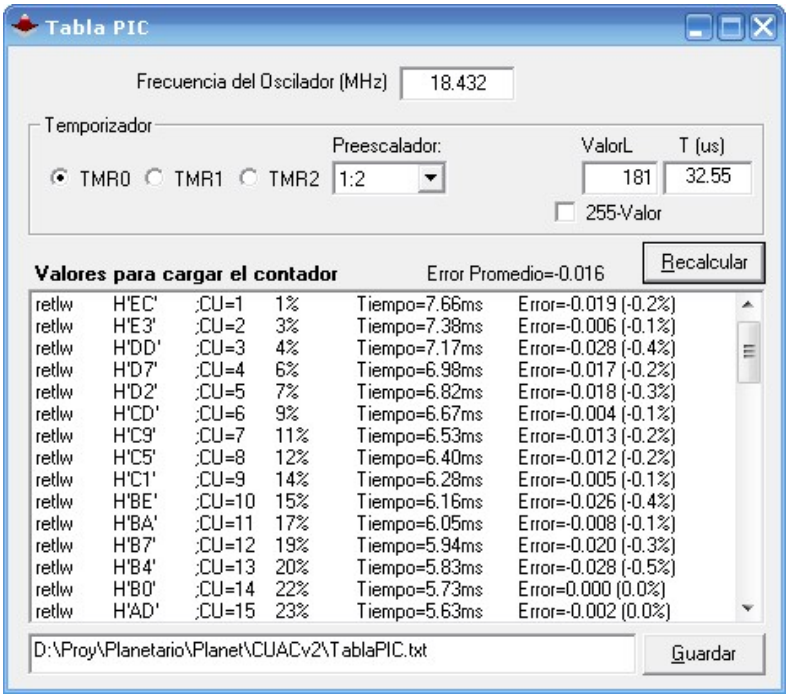


Fig. 3 Herramienta para la generación del código necesario para el control del disparo de los tiristores.

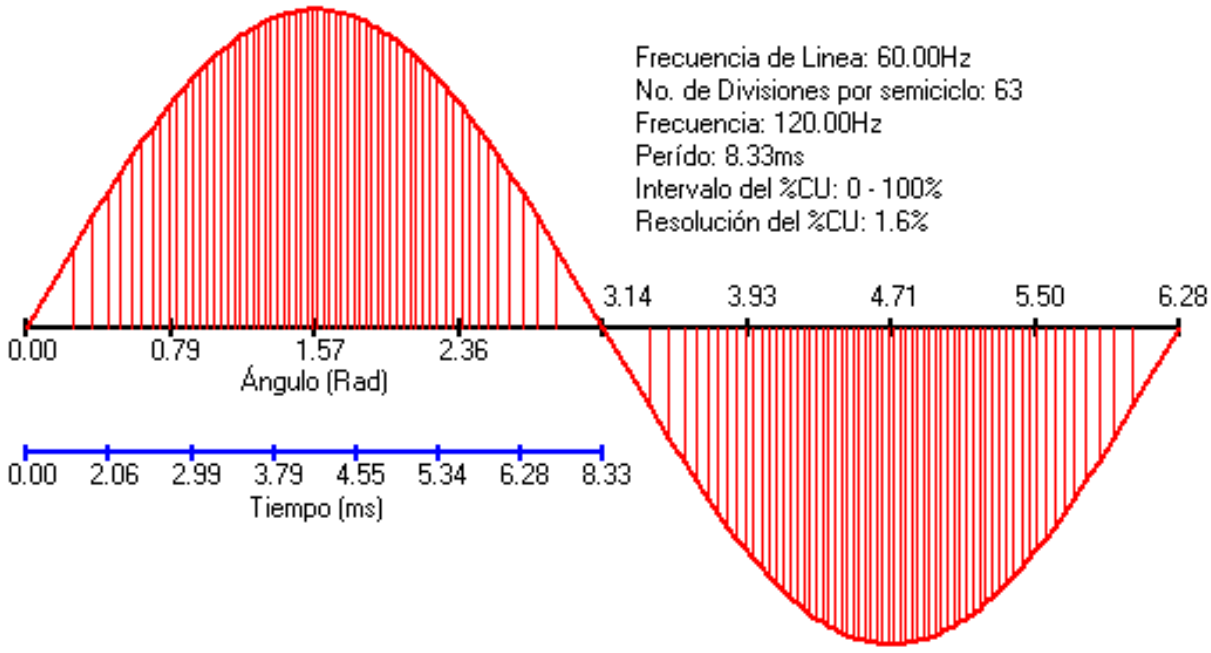


Fig. 4 Gráfica de ángulos y tiempos de disparo de tiristores para obtener intervalos iguales de potencia.

2.1.7. Secuencias de Amanecer y de Atardecer.

La representación de la transición de la noche al día, denominada amanecer, y el proceso contrario llamado atardecer, es una de las funciones más importantes del proyector planetario. Son simulaciones interesantes de un fenómeno natural, que además preparan de forma gradual al espectador para pasar de un nivel de iluminación a otro. El atardecer permite la adaptación de un ambiente con abundante iluminación a uno muy oscuro y así se puede apreciar sin dificultad el espectáculo que comienza. El amanecer minimiza el impacto con la luz exterior al salir del salón.

Para conseguir estos efectos se preparó una secuencia para el amanecer, que cambia los niveles de iluminación de las tres lámparas involucradas: estrellas, luces azules y luces blancas. Esta secuencia se ejecuta leyendo una tabla ubicada en la EEPROM y que permite seguir las curvas de iluminación características de estos fenómenos (ver figura 5). Para representar el atardecer se ejecuta la misma secuencia pero en sentido inverso.

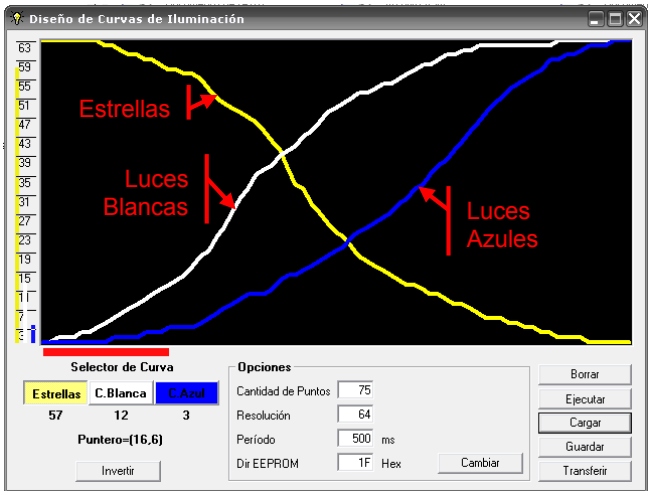


Fig. 5 Curva de iluminación para el amanecer.

Fue necesario desarrollar una aplicación que permitiera, de forma gráfica, diseñar estas curvas y transferir sus valores a la EEPROM del microcontrolador.

La comprobación de esta secuencia se realizará en Proteus. Para ello se pasará cada una de las tres señales digitales por un filtro pasa bajo, obteniendo una medición analógica de cada una. Estas curvas se compararán con las deseadas.

2.2. Simuladores

Se emplean los simuladores MPLAB y Proteus, los cuales se describirán brevemente a continuación.

2.2.1. MPLAB IDE

MPLAB IDE es el entorno de desarrollo que ofrece Microchip (15) para programar y simular el funcionamiento de sus microcontroladores y procesadores de señales digitales. Admite módulos de otros fabricantes para programar en lenguaje C, Basic, entre otros. Posee útiles herramientas de simulación entre las que se destaca el generador de estímulos, que permite cambiar valores de registros, entradas digitales y periféricos, de forma sincrónica, asincrónica y al cumplirse determinadas condiciones.

Ofrece un rápido acceso a los registros de funciones especiales y de usuario, en cualquiera de los formatos numéricos empleados. Permite editar y modificar la memoria de datos RAM y EEPROM y de programa. Cuenta también con un contador de tiempo muy útil en las tareas de temporización.

Posee un entorno de programación similar al de los lenguajes de alto nivel, facilitando el indexado, la identificación de comandos y del tipo de datos. Durante la simulación visualiza el valor de las variables y registros al pasar el ratón sobre ellos. Cuenta con un Tracer, un Analizador Lógico y puede manejar varios programadores y emuladores. Posee un módulo que puede interactuar con el MATLAB Simulink y otro para el Proteus VSM. En nuestro trabajo se empleó MPLAB IDE 7.52.

2.2.2. Proteus.

Proteus es un software de diseño electrónico desarrollado por Labcenter Electronics(8) que consta de dos módulos: Ares e Isis y que incluye un tercer módulo opcional denominado Electra.

Con Isis podemos diseñar el circuito electrónico con gran cantidad y variedad de componentes en sus bibliotecas, incluyendo varias familias de microcontroladores. Mediante el módulo de aplicación VSM podemos simular en tiempo real y de forma interactiva, el funcionamiento del circuito, incluyendo el programa del microcontrolador.

Ares es una herramienta de diseño de circuitos impresos, que se complementa con Electra para el ruteo automático de las conexiones. En nuestro trabajo se empleó Protel 99 SE (16) para el diseño del esquema y del PCB, y el módulo Isis de Proteus 7.1 SP2 para la simulación.

El diseño del esquema electrónico realizado en Protel 99 SE se transfirió a 3 diseños en Proteus, optimizado para la simulación de cada aspecto de interés.

2.2.3. Proteus VSM MPLAB Driver.

Estos dos simuladores unen sus potencialidades mediante un módulo o "plug in" denominado Proteus VSM MPALB Driver. Este permite utilizar muchas de las facilidades del MPLAB para la puesta a punto del programa del microcontrolador, en conjunto con la simulación en Proteus del resto del esquema eléctrico. Para ello se debe activar en el menú Debugger/Select tool la opción Proteus VSM. Es posible ejecutar ambos simuladores en una misma computadora o en dos máquinas enlazadas por una red local. A partir de ahí trabajarán de forma coordinada.

3. RESULTADOS

A continuación se muestran los principales resultados obtenidos mediante la simulación de la interfaz.

3.1. Simulación en MPLAB.

En la simulación con MPLAB se utilizó el mismo espacio de trabajo empleado para la programación y la puesta a punto del programa del microcontrolador. Los aspectos verificados se describen a continuación.

3.1.1. Pulsadores Play y Stop.

Para comprobar la detección del cambio de estado, en las entradas digitales asociadas a los pulsadores Play y Stop, se colocó un punto de ruptura en la subrutina de atención a la interrupción por cambio de RB. Se configuró también el generador de estímulos asincrónicos como se muestra en la figura 6.

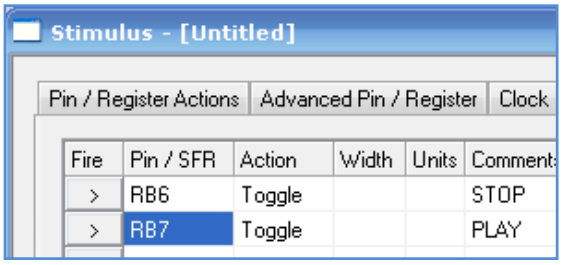


Fig. 6 Generador de estímulos asincrónicos.

Al pulsar RB7 se genera una interrupción y se observa el cambio en los registros de funciones especiales (ver figura 7). Se comprobó también la correcta ejecución de la subrutina.

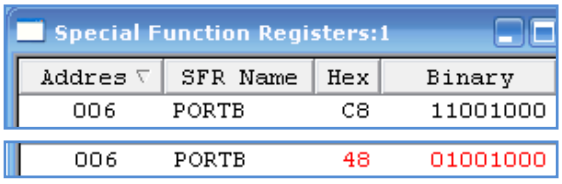


Fig. 7 Estado de las entradas digitales RB6 y RB7 antes y después de pulsar Play.

3.2. Simulación con Proteus VSM MPLAB Driver.

En la simulación con este módulo se utilizó el mismo espacio de trabajo empleado para la simulación con MPLAB, pero seleccionando el simulador Proteus VSM. Esto nos permite inyectar las señales de estímulos de los osciladores y pulsado-

res, y acceder a los registros fácilmente. Los aspectos verificados se describen a continuación.

3.2.1. Reloj de Tiempo Real.

Para comprobar el Reloj de Tiempo Real (RTC) se colocó un punto de ruptura en la subrutina de atención a la interrupción de TMR1 y se cargó en la ventana Proteus VSM MPALB Viewer del esquema 1 mostrado en el anexo 3.

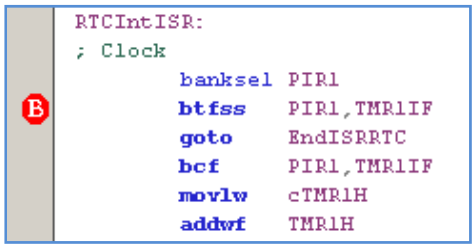


Fig. 8 Punto de ruptura en la subrutina del RTC.

Se registró durante varias iteraciones el tiempo en que se detenía la simulación en el punto de ruptura, el cual se puede observar en la zona izquierda de la barra de estado del Proteus VSM MPALB Viewer. La figura 9 muestra las mediciones realizadas en 60 iteraciones (60 segundos) que son suficientes para comprobar que la generación de esa interrupción se realiza cada 1 seg, con suficiente precisión.

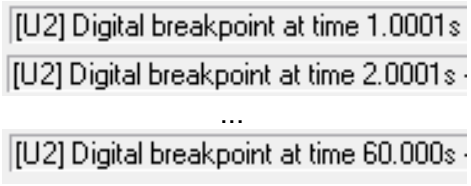


Fig. 9 Mediciones de tiempo en 60 iteraciones.

El incremento de la hora y la fecha se verificó en los registros del MPLAB. Para reducir el número de iteraciones y se cambiaron varias veces las condiciones iniciales a valores fronteras como son: para el cambio de día, la hora 23:59; para el cambio de mes el 28 de febrero en años bisiestos y no bisiestos; y el 31 de diciembre para el cambio de año. Todas las pruebas resultaron satisfactorias.

3.2.2. Detección del cruce por cero.

La comprobación de la detección del cruce por cero se realizó con el esquema 1 (ver anexo 3). Se verificó que a partir del estímulo proporcionado al pin RB0/Int por el oscilador de 120Hz, se genera la interrupción en el instante de tiempo correcto. Para ello se colocó un punto de ruptura en la subrutina de atención a la interrupción INT y se midió el tiempo de cada iteración. Los resultados se muestran en la siguiente figura.

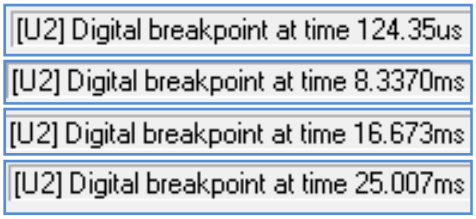


Fig. 10 Mediciones de tiempo entre pulsos.

Las mediciones muestran que entre cada llamada a la subrutina de atención hay un intervalo de tiempo aproximadamente igual a 8.33ms, lo cual corresponde con el período de un semiciclo de la señal de alimentación.

Se pudo comprobar también a nivel de registros el correcto funcionamiento de la subrutina.

3.2.3. Comunicación Serie.

La comunicación serie se verificó a través del Terminal Virtual y se probaron todos los comandos definidos para la interfaz. El Terminal se configuró a una velocidad de 38400 BPS, 8 bits de datos, sin paridad, sin control de flujo y polaridad normal para los pines RX/TX.

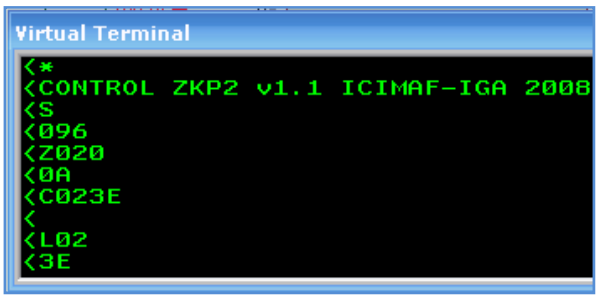


Fig. 11 Transmisión y recepción serie.

En la figura 11 se observan algunos comandos y sus respuestas. Al enviar "*" se solicitó la identificación de la interfaz, por lo que esta respondió con una trama que contiene una descripción propia.

El caracter "S" pide el estado de la alarma y de la batería, y recibe como respuesta 096. Esto significa que no se está en alarma por fecha u hora, y que la batería está en su valor máximo (96h).

Al enviar "Z020" estamos solicitando el valor de la dirección 20h en la RAM. Recibimos como respuesta 0Ah, lo cual se corrobora en la figura 12.

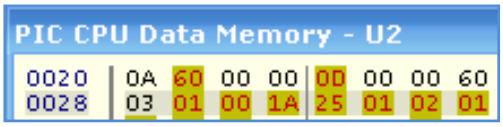


Fig. 12 Valor en RAM.

El comando "C023E" escribe en la dirección 02h de la EEPROM el valor 3Eh y recibimos confirmación de su ejecución con "<". Esto se verifica después en la tabla mostrada en la figura 13.

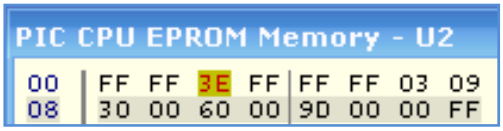


Fig. 13 Escritura en EEPROM.

Se puede observar, en los comandos de la figura 11, el carácter de inicio de trama "<", pero el fin de trama correspondiente al retorno de carro (0Ch) que no es visible.

3.3. Simulación en Proteus.

Los elementos comprobados mediante Proteus se describen a continuación.

3.3.1. Pulsadores Play y Stop.

Empleando el esquema 1 mostrado en el anexo 3, se simuló el cambio de estado de los pulsadores Play y Stop. Se pudo comprobar que al pulsar Play se comenzaba a ejecutar la secuencia seleccionada en el potenciómetro RV1, y al presionar Stop se detenía su ejecución. También se observó la activación de los Leds Rojo y Verde en cada etapa de trabajo.

3.3.2. Secuencias de comandos.

Para comprobar la ejecución de la secuencia especial (ver epígrafe 2.1.3) se utilizó el esquema 2, donde se fijó en la entrada analógica para la selección de la secuencia el valor 0V (ver anexo 5). También se conectaron dos generadores de pulso para activar la secuencia y una punta de prueba por cada salida digital. El funcionamiento se graficó mediante un gráfico digital. Se puede apreciar en el anexo 6 que la secuencia se ejecuta correctamente y en el tiempo esperado.

3.3.3. Control del nivel de iluminación.

Para comprobar el funcionamiento del algoritmo de control, que regula el disparo de los tiristores en función de la potencia que se le debe entregar a las lámparas, se utilizó el esquema 1 (ver anexo 3). En el podemos apreciar un osciloscopio digital de 4 canales y un terminal de comunicación serie.

Con el terminal de comunicación se le transmitió a la interfaz las órdenes para cambiar el valor de disparo de los tiristores en un rango de 0 a 63 (equivalente a 0 y 100%). Con el osciloscopio se observó la señal de detección de cruce por cero y las de disparo de los 3 tiristores. Una de estas pruebas se muestra en el anexo 4.

Al enviar el comando "X0810" se pone a T0 al 25% y debe disparar al tiristor a los 5.53ms del cruce por cero. Con "X0860" T1 va al 50% por lo que se debe activar a 4.12ms del cruce. Finalmente se envía "X08B6" y T2 va al 85% ocurriendo la activación a los 2.06ms. Esto se puede apreciar en la figura anexada donde la escala de tiempo del osciloscopio es 1ms por división.

3.3.4. Secuencias de Amanecer y de Atardecer.

Para comprobar la ejecución del amane-

cer y del atardecer se diseñó una secuencia de comandos especial (ver anexo 1 secuencia 2) que activará ambos procesos. Esta se colocó en la zona de memoria correspondiente a la secuencia 2 y se fijó en la entrada analógica de selección de secuencia el valor 0.5V. El esquema cargado se observa en el anexo 7.

Se hicieron pasar las señales de control de los tiristores por filtros pasa-bajos y a la salida de estos se colocó una punta de prueba. Estas tres salidas analógicas se graficaron mediante un gráfico analógico.

Se puede apreciar las curvas del amanecer en la figura del anexo 8, y se observa una gráfica similar a la de la figura 5. El tiempo corresponde a los 15 seg definidos por el comando AMANECER 1. Se observan algunas irregularidades en la señal, debido a la resolución empleada para el control. Un resultado similar se obtuvo con el atardecer.

4. CONCLUSIONES

Mediante la simulación en MPLAB y Proteus, de la interfaz de supervisión y control para proyectores planetarios ZKP-2, se pudo validar el funcionamiento de la mayor parte del circuito.

Se comprobaron varios elementos importantes del diseño y se obtuvieron resultados satisfactorios, lo cual nos permitió continuar hacia la etapa de fabricación del prototipo.

Al usar las herramientas de simulación, durante la puesta a punto, pudimos detectar errores en el diseño y corregirlos antes de fabricar la interfaz. La fortaleza de estas herramientas garantiza una simulación interactiva y un fácil acceso a los registros y periféricos del microcontrolador.

5. REFERENCIAS BIBLIOGRÁFICAS

1. **eCircuit Center.** eCircuit Center SPICE Topics. *A BRIEF HISTORY OF SPICE.* [En línea] 2003. http://www.ecircuitcenter.com/History%20SPICE.htm.

2. AIM-Spice. [En línea] http://www.aimspice.com/.

3. AKNM Circuit magic. [En línea] http://www.geocities.com/ecalcsys/.

4. AnaSoft Spice Analog Simulation with Schematic Capture. [Online] http://www.anasoft.co.uk/.

5. Circuit Magic. [Online] http://www.circuit-magic.com/.

6. **Crocodile Clips Ltd.** Crocodile Thecnology 3D. [En línea] http://www.crocodile-clips.com.

7. Electronics Workbench. [En línea] http://www.interactiv.com/.

8. **Labcenter Electronics.** Proteus. [En línea] 2008. http://www.labcenter.co.uk/.

9. Micro-Cap. [En línea] http://www.cdquickcache.com/pcecap.htm.

10. VisualSpice. [Online] http://www.islandlogix.com/.

11. eCircuit Center. [En línea] http://www.ecircuitcenter.com/.

12. **Carl Zeiss.** What is a Planetarium. [En línea] 2008. http://www.zeiss.de/de/planetarium/home_e.nsf/.

13. **Wikimedia Foundation Inc.** Wikipedia. [En línea] 2006. http://es.wikipedia.org/wiki/Planetario.

14. **Bates, Martin.** *Interfacing PIC Microcontrollers Embedded Design by Interactive Simulation.* 2006.

15. **Microchip Technology Inc.** Microchip. [En línea] 2008. http://www.microchip.com.

16. **Protel International Limited Ltd.** *Design Explorer 99 SE SP6 v6.6.7.* 2000.

17. CAD-Migos Software Tools. [En línea] http://www.cadmigos.com/.

18. Dolphin Integration SMASH . [En línea] http://www.dolphin.fr/medal/smash/smash_overview.html.

19. SpiceCreator. [En línea] http:http://www.cadmigos.com/Products/AMS/spicecreator.asp.

ANEXO 1. Lista de Instrucciones

Comando	***Descripción***
SD#=V	Asigna valor V a la salida digital #.
SOL=V	Asigna valor V a salida digital correspondiente al Sol.
LUNA=V	Asigna valor V a salida digital correspondiente a la Luna.
PLANETAS=V	Asigna valor V a salida digital correspondiente a los planetas.
ECLIPTICA=V	Asigna valor V a salida digital correspondiente a la eclíptica.
MERIDIANO=V	Asigna valor V a salida digital correspondiente al meridiano.
METEORO=V	Asigna valor V a salida digital correspondiente a los meteoros.
JUPITER=V	Asigna valor V a salida digital correspondiente a Júpiter.
SATELITES	Asigna valor V a salida digital correspondiente a los satélites.
MOVANUAL=W	Define el estado y el sentido del motor del movimiento anual.
MOVDIARIO=W	Define el estado y el sentido del motor del movimiento diario.
MOVPOLAR=W	Define el estado y el sentido del motor del movimiento polar.
T#=<	Asigna ángulo de disparo < al tiristor #.
ESTRELLAS=A	Define disparo A para las lámparas de las estrellas.
CAZUL=A	Define disparo A para las lámparas azules de la cúpula.
CBLANCA=A	Define disparo A para las lámparas blancas de la cúpula.
AMANECER P	Ejecuta secuencia de transición del amanecer con período P.
ATARDECER P	Ejecuta secuencia de transición del atardecer con período P.
ESPERA S	Espera un tiempo de S seg. sin ejecutar nuevas instrucciones.
TIEMPO= HH:MM	Actualizar tiempo en hora (H) y minutos (M).
FECHA=DD/MM/AA	Actualiza la fecha.
INI	Inicio de secuencia.
FIN	Fin de secuencia.
;	Comentarios.
‘	Comentarios.

Variable	**Valor**	**Descripción**
V	0	Apagado
	1	Encendido
W	0	Detenido
	1	Movimiento en sentido positivo
	-1	Movimiento en sentido negativo
S	1 a 256	Tiempo en segundos
P	1 a 256	Coeficiente de la duración del amanecer o atardecer. Duración= P*15 seg
#	0 a 13	Señales Digitales
	0 a 2	Tiristores

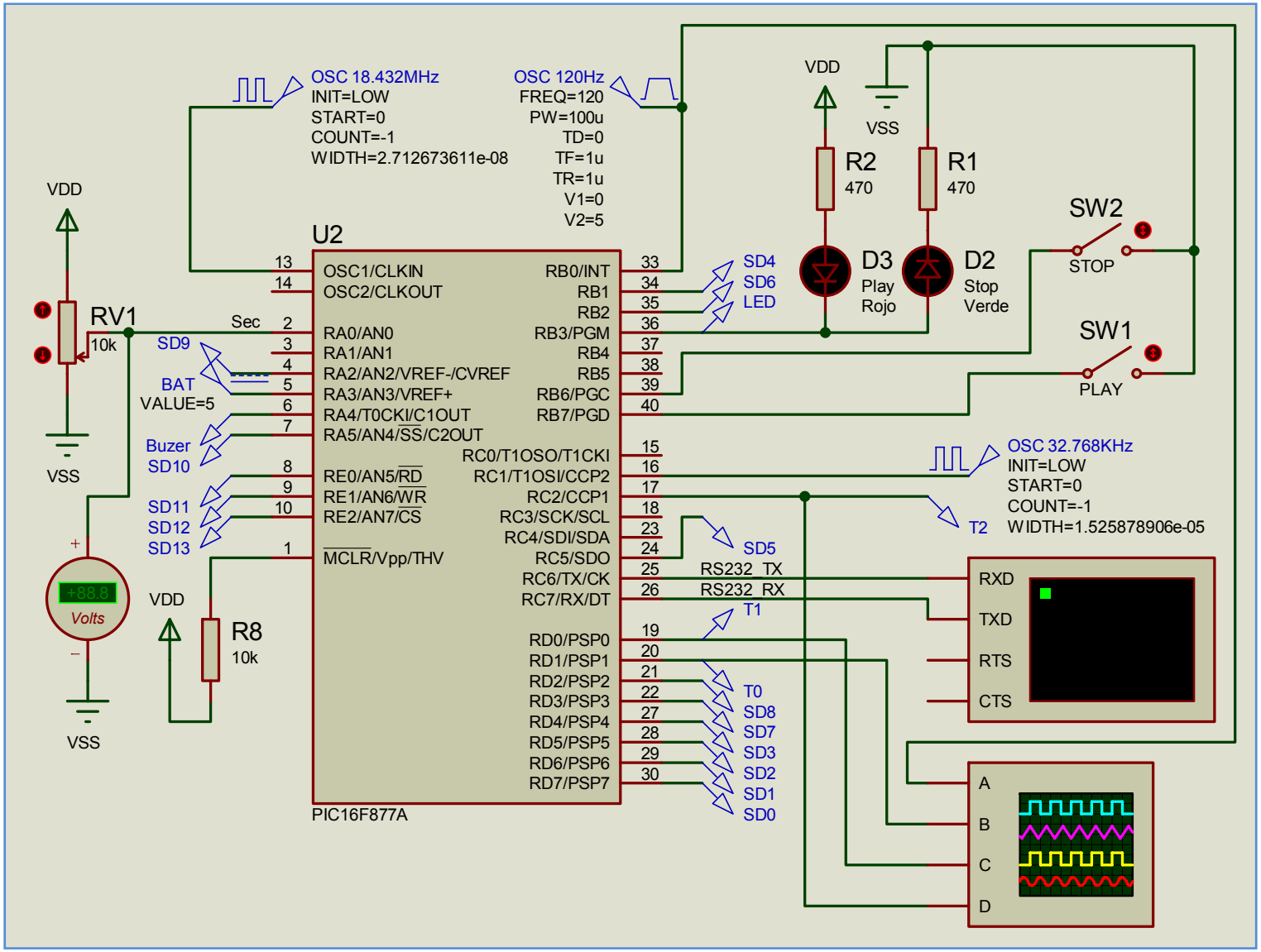
Variable	**Valor**	**Descripción**
<	0 a 63	Corresponde a cantidades iguales de potencia entregada al tiristor. Es semejante al % de ciclo útil en motores de corriente directa.
HH	0 a 23	Hora
MM	0 a 59	Minutos
DD	1 a 31	Día
MM	1 a 12	Mes
AA	0 a 99	Año

ANEXO 2. Secuencias 1 y 2

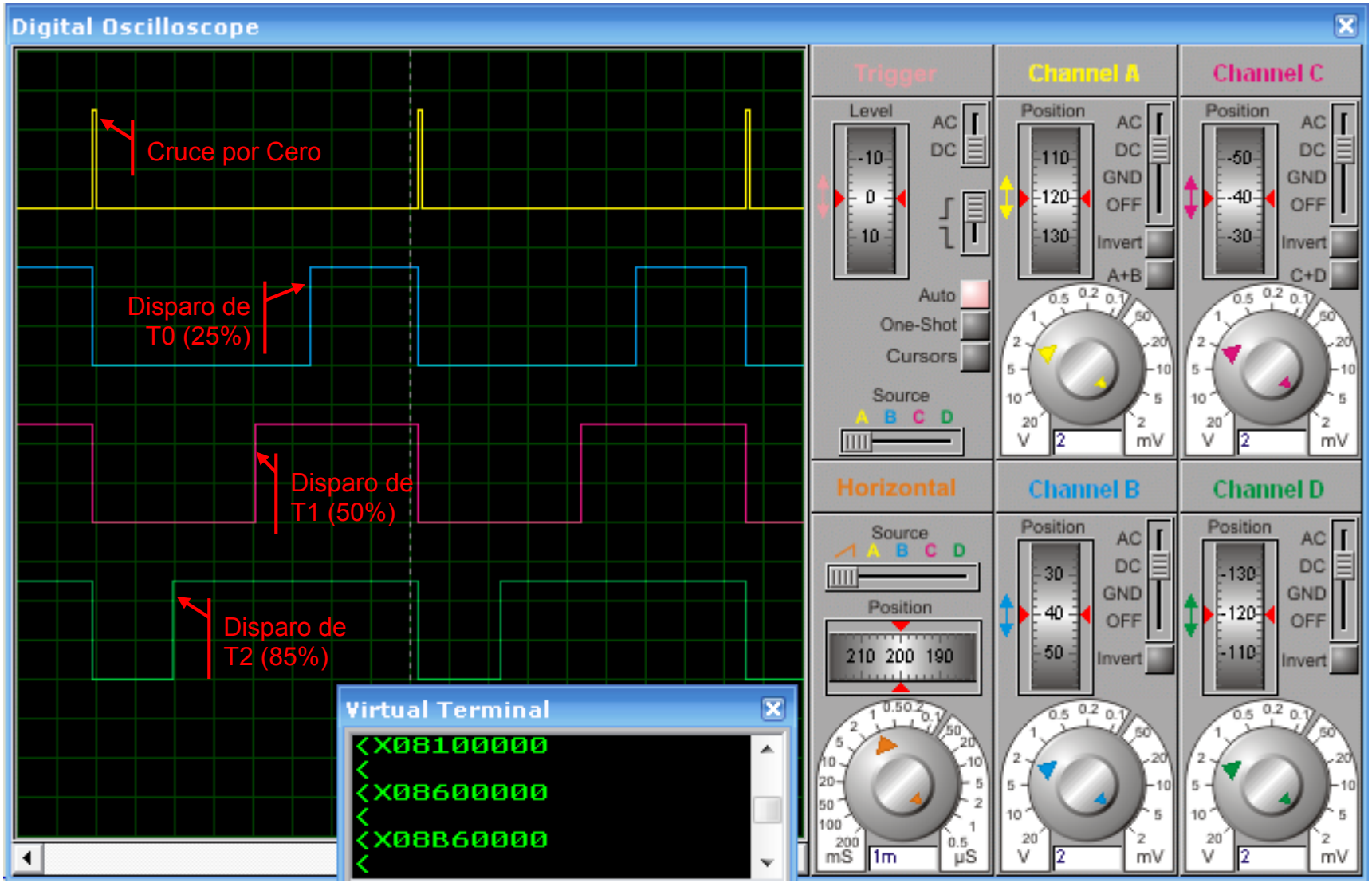
```
INI          ; Secuencia 1
SD=0         ; Todas las SD van a 0
ESPERA 1     ; Espera 1 seg
SD0=1        ; Salida Digital 0 = 1
ESPERA 1
SD1=1
ESPERA 1s
SD2=1
ESPERA 1
SD3=1
ESPERA 1
SD4=1
ESPERA 1
SD5=1
ESPERA 1
SD6=1
ESPERA 1
SD7=1
ESPERA 1
SD8=1
ESPERA 1
SD9=1
ESPERA 1
SD10=1
ESPERA 1
SD11=1
ESPERA 1
SD12=1
ESPERA 1
SD13=1
ESPERA 2
SD13=0
ESPERA 1
SD12=0
ESPERA 1
SD11=0
ESPERA 1
SD10=0
ESPERA 1
SD9=0
ESPERA 1
SD8=0
ESPERA 1
SD7=0
ESPERA 1
SD6=0
ESPERA 1
SD5=0
ESPERA 1
SD4=0
ESPERA 1
SD3=0
ESPERA 1
SD2=0
ESPERA 1
SD1=0
ESPERA 1
SD0=0
ESPERA 1
FIN
```

```
INI              ; Secuencia 1
AMANECER 1       ; Amanecer en 15seg
ESPERA 16        ; Espera 16 seg
ATARDECER 1      ; Amanecer en 15seg
ESPERA 15        ; Espera 15 seg
FIN
```

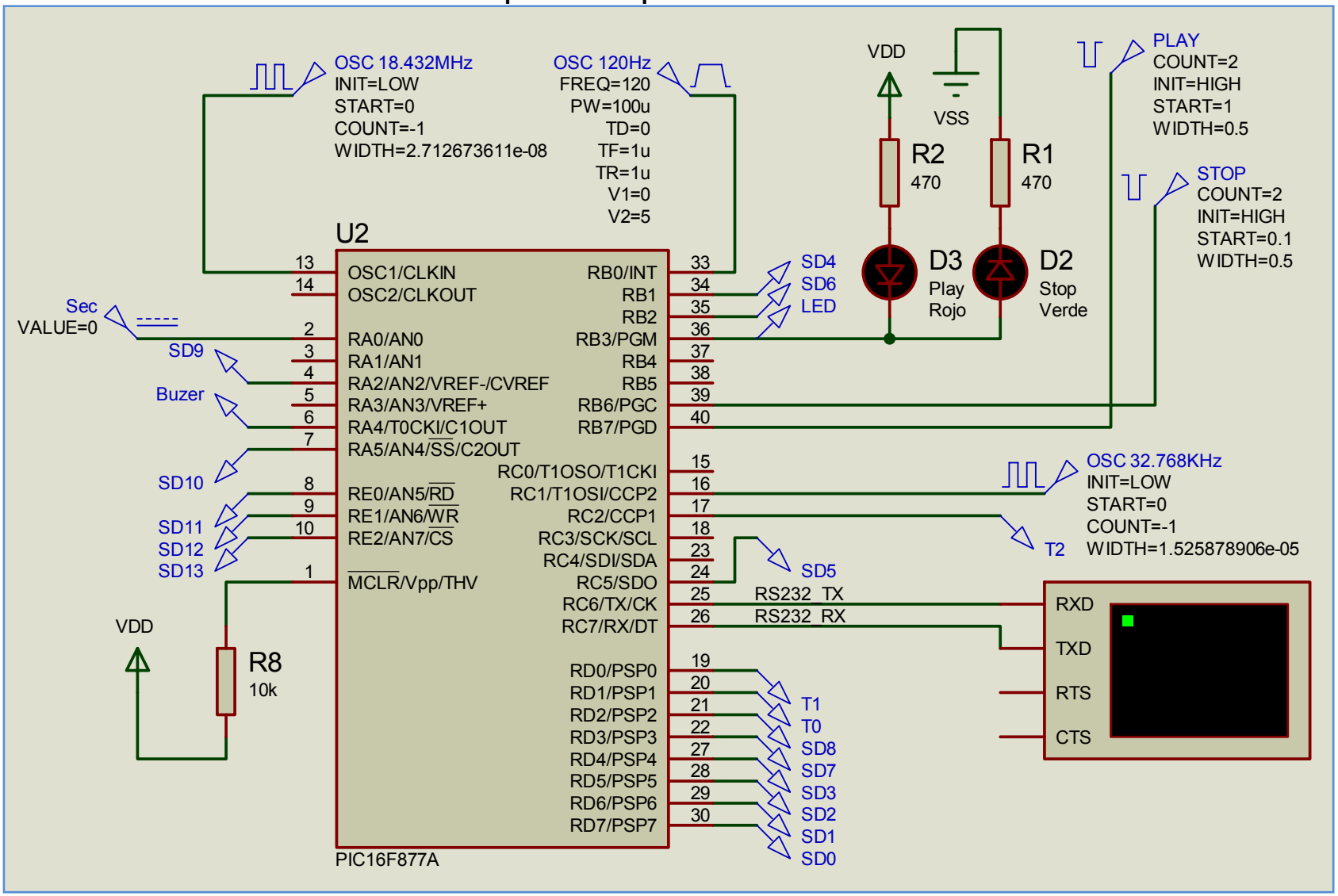
ANEXO 3. Esquema No. 1 para la simulación en Proteus.



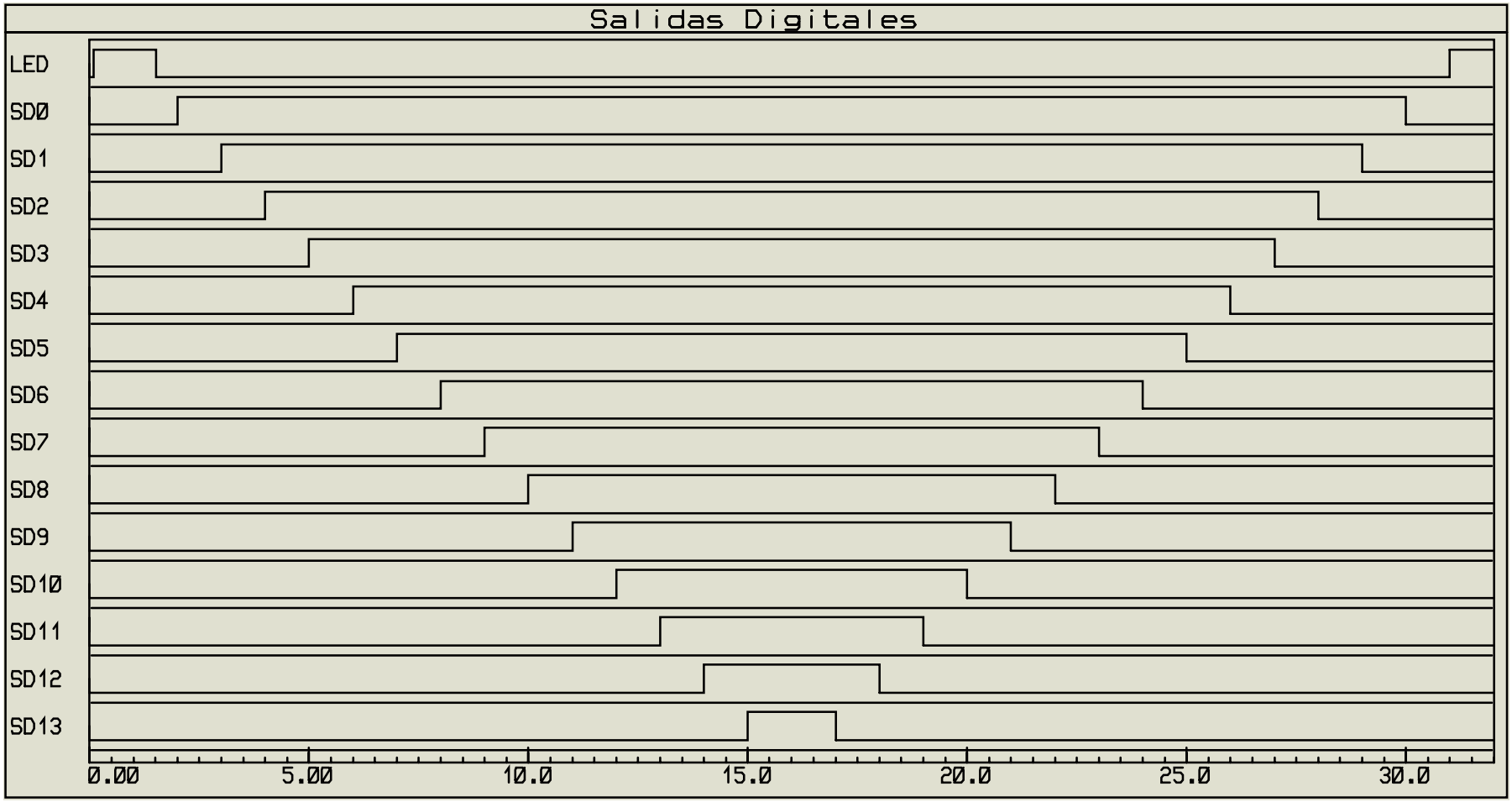
ANEXO 4. Simulación en Proteus del control de iluminación.



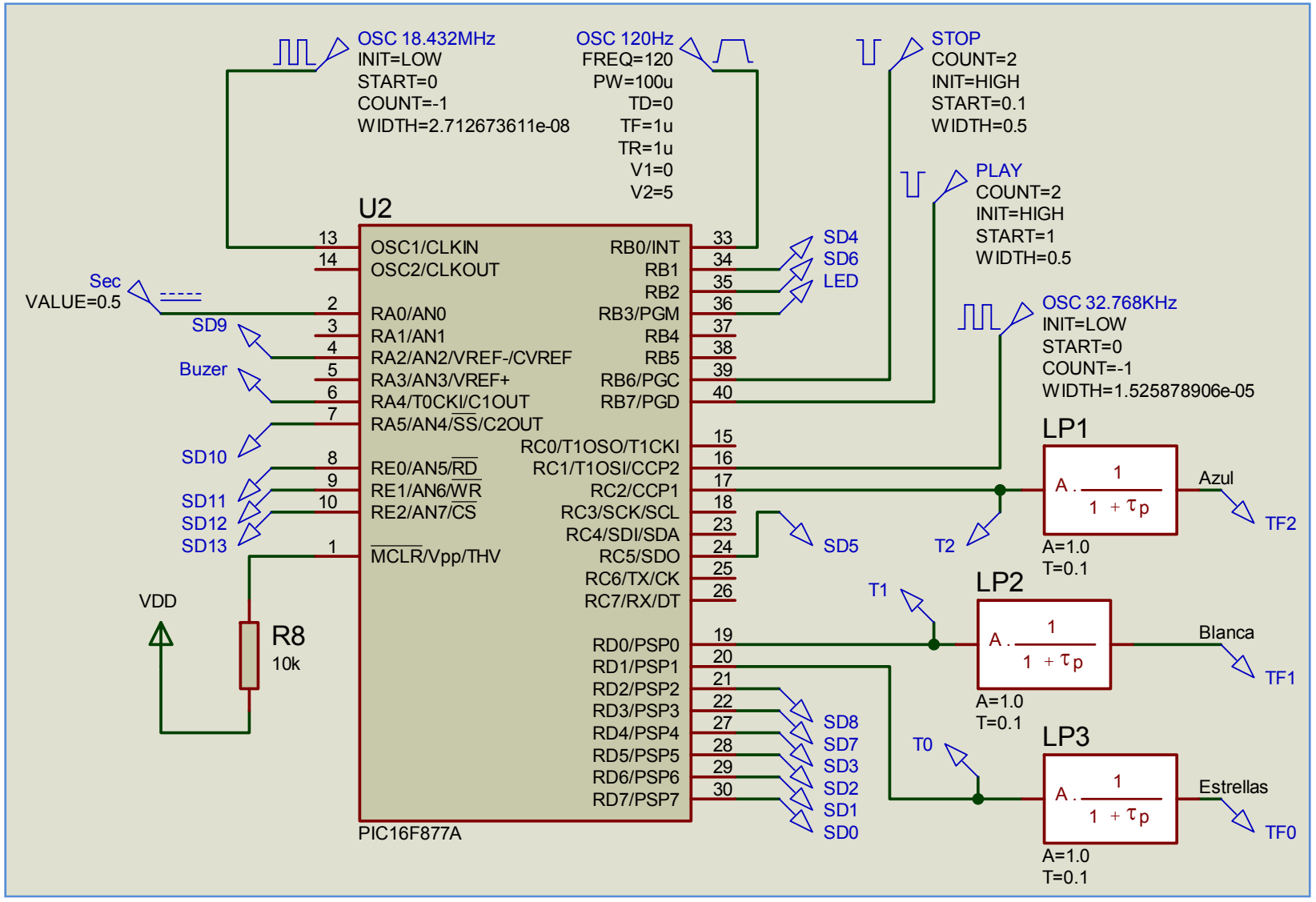
ANEXO 5. Esquema No. 2 para la simulación en Proteus.



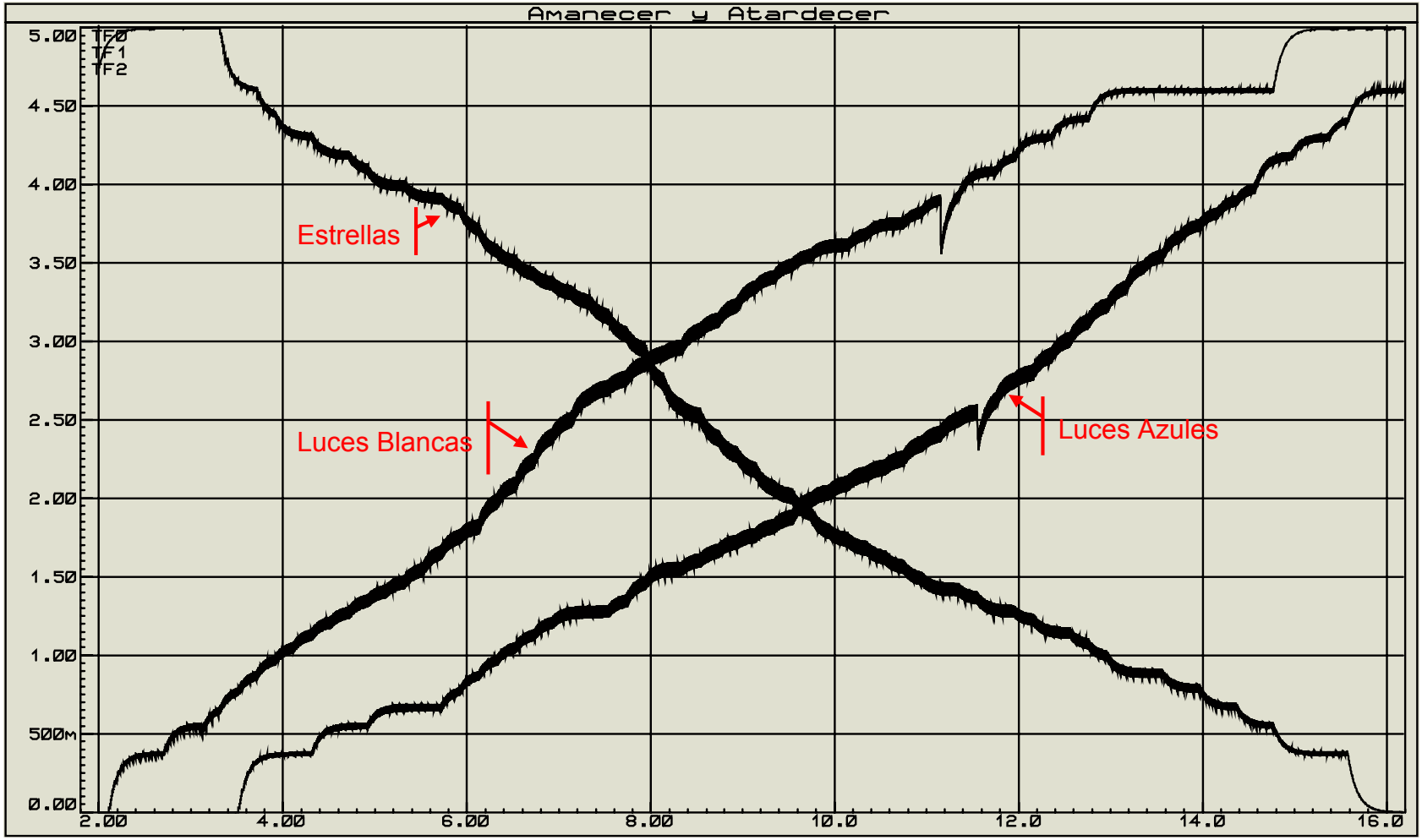
ANEXO 6. Resultados de la simulación de la secuencia de comandos 1 en Proteus .



ANEXO 7. Esquema No. 3 para la simulación en Proteus.



ANEXO 8. Resultados de la simulación de la secuencia de amanecer en Proteus.



ANEXO 9. Esquema eléctrico de la tarjeta principal realizado en Protel 99 SE.