

Alumnos

- [Trabajos prácticos](#)
- [Apuntes](#)

Electrónica

- [Tutoriales](#)
- [Componentes](#)
- [Circuitos](#)
- [Descargas](#)
- [Enlaces](#)

Entretenimiento

- [Música](#)
- [Imágenes](#)
- [Humor](#)
- [Juegos](#)

Activación de carga por infrarrojo

Introducción

En esencia se construye un receptor de infrarrojos que capta y decodifica la señal enviada por un mando. El mando puede ser un control remoto de cualquier aparato (tv, video, sonido, etc.) y de cualquier marca, pero en este trabajo el hardware se diseña para ocupar un mando marca Sony de una televisión, sin que en ello exista una razón técnica, salvo el precio de éste.

La etapa decodificadora está compuesta por un microcontrolador, el cual toma la señal del receptor de infrarrojo. Dicha señal, una vez capturada e identificada por el micro, es tratada por éste para llevar a cabo una determinada función dependiendo del botón pulsado en el mando Sony.

El hardware aquí diseñado, es usado para controlar una ampolleta incandescente. Específicamente, el dispositivo se construyó para variar la intensidad de luz, (al igual que un dimmer) encenderla y apagarla.

Es obvio (para los entendidos) que la carga puede ser desde un o unos simples led, hasta un motor, todo depende de la etapa de potencia que se agregue a la etapa de control.

Emisor IR

Primeramente, debemos saber lo que tenemos a la salida del control remoto, que como dije es un Sony de tv. Por ello, lo mejor tanto para el lector como para mí, es que visites [esta página](#) donde encontrarás información general, pero valiosa, sobre como obtener la señal de salida de un control remoto. No obstante, la Figura 1 es un diagrama obtenido con un osciloscopio (generosamente prestado, muchas gracias) conectado al pin de salida del integrado IRM8601S. Mas adelante se encuentra la lista completa de componentes para este sistema embebido.

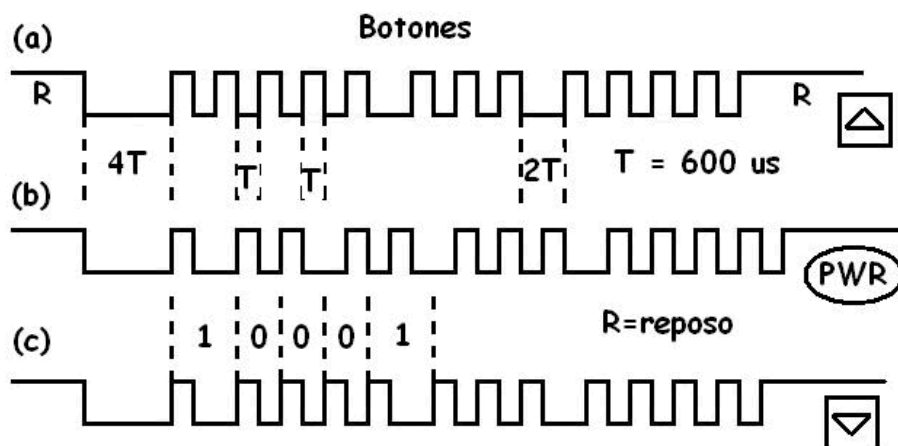


Fig. 1. Diagrama de tiempos

De la Figura 1 se tiene: (a) botón de canal ascendente, (b) botón de power PWR y en (c) el botón de canal descendente.

Y la función que cumplen estos botones en el sistema embebido:

- (a) aumentar intensidad lumínica
- (b) encender o apagar la ampolleta
- (c) disminuir intensidad lumínica

A la salida del Receptor de Infrarrojos tenemos un estado ALTO representado por R=reposo, cuando recibe señal a través del mando cambia a estado BAJO con una duración de $4T=2400\text{ us}$, después viene la trama de unos y ceros con un total de ocho bits, que es la que determina que tecla se pulsa en el mando. Esta trama es la que nos interesa programar en el microcontrolador. Los últimos cuatro bits que son todos ceros no se toman en cuenta en la programación, esto se debe a que todos los botones pulsados en el mando terminan de igual forma. La ventaja o desventaja de este enfoque es que el control de la carga podrá ser activado tanto con un mando de tv como con uno de video mientras sea de SONY.

Por último, en cuanto al tramo de $4T$ al principio del diagrama, éste nos sirve para saber en que momento exacto se ha pulsado un botón en el mando, y así estar atento a capturar la trama que identifica la tecla pulsada.

Receptor IR

Para recibir la señal IR enviada por el control remoto, existen varios dispositivos, tales como el TSOP1736, IRM8601S, gp1u52x, etc. En particular aca usamos el IRM8601S.

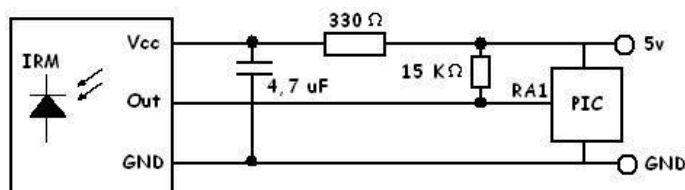


Fig. 2. Conexión del receptor IR

En la Figura 2 se muestra la forma de conectar el dispositivo al microcontrolador. Refiérase al Datasheet para más detalles.

Decodificador

Un microcontrolador, el PIC 16f84, es el encargado de identificar el botón pulsado en el mando Sony. Para esto, el pin PA1 del puerto A del micro se conectó al pin de salida del IRM8601S, es decir, la comunicación entre ambos dispositivos es serial "sólo por si acaso". El programa residente en el PIC [puede verse aquí](#).

No se harán comentarios explicativos del código presentado; pues la idea no es dar clases de programación, sí se destaca, que mediante el programa, este PIC no es el encargado de provocar el disparo para la etapa de potencia, puesto que para ello se usa un segundo PIC.

La necesidad de usar dos PIC 16f84 radica en que el primero, (el decodificador) usa el TMR0 para sincronizar e identificar el tren de pulsos provenientes del pin de salida del IRM8601S, y el segundo PIC (el disparador) usa el TMR0 para variar el ángulo de disparo del triac de potencia conectado a la carga.

Resumiendo; si sólo queremos encender o apagar la ampollita, nos basta con un solo PIC, pero como la idea es también variar la intensidad lumínica, se necesitan dos PIC. Con esto no quiero decir que necesariamente deban usarse dos PIC para resolver el problema, pues talvés alguien que este leyendo esto, pueda hacerlo con uno solo.

Disparador

Esta compuesto por el segundo PIC 16f84 el cual recibe la información del decodificador en forma paralela y es el encargado de excitar al fototriac, el MOC3021, para que este sea finalmente el que envía la señal de disparo a la puerta o gate del triac BTA 08-600B conectado a la carga. El código del disparador [puede verse aquí](#).

Listado de Componentes	
Cantidad	Descripción
6	Resistencia 10K 1/4w
2	Resistencia 330 1/4w
1	Resistencia 15K 1/4w
1	Resistencia 3,3K 1/4w
1	Resistencia 56K 1/4w
1	Resistencia 180 1/4w
1	Resistencia 39 1/4w
2	Diodos 1N4007
1	Condensador 4,7 uF 25v
1	Condensador 0,01 uF 1Kv
2	PIC16F84
1	LM324
1	MOC3021
1	BTA08-600B
1	Trafo tap central 12v

Esquemáticos

En esta sección se muestran los circuitos esquemáticos por partes, el listado anterior dicta todos los componentes salvo la fuente de poder cc de 5v y los componentes asociados a la conexión típica de los PIC como por ejemplo los cristales de 4 MHz, condensadores de 22pF, reset, etc. El conexionado de trabajo del PIC se puede encontrar en los Datasheet o en Internet. La Fig. 2 (mas arriba) muestra la conexión del receptor IR, en la Figura 3, de arriba hacia abajo el decodificador y el disparador respectivamente.

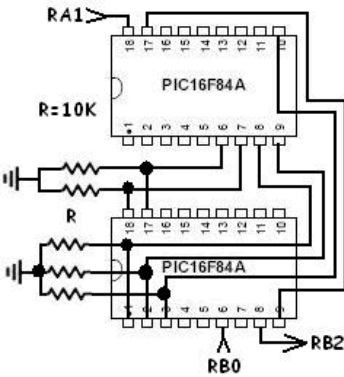


Fig. 3. Pic decodificador y pic disparador

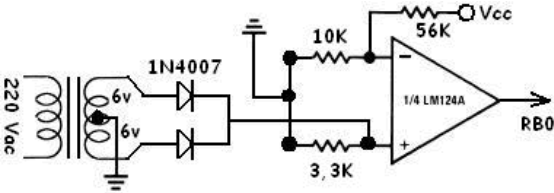


Fig. 4. Detector de cruce por cero con LM324

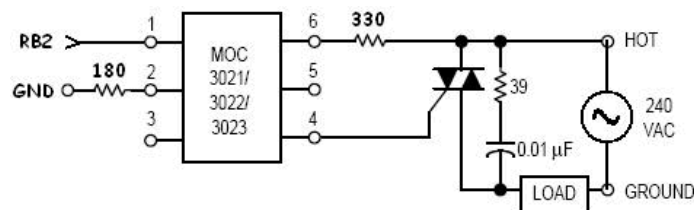


Fig. 5. Etapa de potencia para activar la carga

Conclusión

El lenguaje de programación usado es ensamblador. Fue elegido para tener un control exacto en los tiempos de sincronismo con la señal infrarroja. De igual manera puede abordarse la programación en C, teniendo claro que al momento de compilar el programa, puede resultar en una mayor cantidad de memoria de código y que se vea afectada la sincronía con la señal. La ventaja de programar en ensamblador además del sincronismo, es que si la aplicación demanda rapidez y eficiencia en memoria, éste lenguaje es el adecuado. A modo de información extra, el código fue escrito usando el entorno de programación MPLAB de Microchip y compilado con el MPASM, en todo caso se puede escribir en el notepad o block de notas de windows y después compilarlo usando el MPASM.

Para que el hardware funcione solamente con un tipo de control, es necesario que el programa decodificador sea capaz de capturar la trama completa, desde 4T hasta los últimos 4 bits que son todos ceros como se mostró en el diagrama de tiempos. Para ello, basta con agregar algunas subrutinas mas, pero nada complicado.

La conexión ac para la carga está dada por la red eléctrica, es decir, 50Hz y 220v, datos que son básicos para el cálculo de potencia entregada a la ampolleta en función del tiempo de disparo en el triac. Ya que, como se mencionó anteriormente, se varía el ángulo de disparo para el triac. Si no quisiéramos utilizarlo como dimmer, entonces, se puede reemplazar el MOC3021 por un MOC3041 el cual tiene un detector de cruce por cero incorporado.

El autor

Walter Hernández Angel reside en Chile, donde se desempeña como desarrollador independiente de sistemas embebidos, con distintos microcontroladores. Para cualquier consulta, su dirección de e-mail es xtoqui@hotmail.com

Material adicional

[Hoja de datos del BTA08-600B](#)

[Hoja de datos del IRM8601S](#)

[Hoja de datos del MOC3021/3022/3023](#)

[Hoja de datos del LM324](#)

[Hoja de datos del PIC 16F84](#)

[Trilobytes](#). Sitio citado en este artículo, con información y algunos proyectos.

[Descripción del formato RC5](#)