

Análisis y decodificación de protocolos IR para control de dispositivos con microcontrolador STM32

Integrantes: Monica Julieth Paez,
Lizeth Camila Sánchez,
Neyder Alberto Vargas Guerrero

Resumen—Este informe detalla el diseño y la implementación de un sistema de control remoto utilizando un microcontrolador STM32, un receptor IR VS1838B, un LED transmisor IR y un display LCD 16x2. El sistema opera en dos modos: SET y RUN. En el modo SET, los usuarios pueden capturar y almacenar hasta 15 comandos IR de un control remoto estándar, asignando un número consecutivo a cada uno. La pantalla LCD muestra el número de comandos almacenados. En el modo RUN, el sistema decodifica e interpreta cualquier comando almacenado, mostrando su número correspondiente. Además, los usuarios pueden retransmitir comandos almacenados mediante pulsadores dedicados (hasta 5). El informe describe las conexiones del hardware, el desarrollo del software para el funcionamiento de los modos, la implementación de la interfaz de usuario y la modulación de 38KHz para la transmisión IR.

Keywords— Control remoto, Protocolo IR, Microcontrolador STM32, Receptor IR VS1838B, LED IR transmisor, Display LCD 16x2, Decodificación de comandos

I. INTRODUCCIÓN

Este informe presenta el diseño e implementación de un sistema de control remoto utilizando un microcontrolador STM32. El objetivo del proyecto es crear un sistema que permita controlar dispositivos electrónicos de forma remota mediante la recepción, decodificación e interpretación de comandos enviados por un control remoto estándar.

El informe describe en detalle los diferentes aspectos del proyecto, incluyendo el análisis del protocolo IR, utilizado por el control remoto para transmitir comandos, incluyendo la estructura de las tramas, los tiempos de bit y la modulación, una descripción del diseño del circuito electrónico que interconecta el microcontrolador, el receptor IR, el LED IR y el display LCD. Se presenta el desarrollo del programa para el microcontrolador, incluyendo la implementación de los modos SET y RUN, la decodificación de comandos IR, la gestión del display LCD y la retransmisión de comandos.

II. MARCO TEORICO

La placa STM32F401CEUx es una plataforma de desarrollo de microcontroladores de la serie STM32 de la empresa STMicroelectronics. Los microcontroladores de esta serie se basan en la arquitectura ARM Cortex-M y son conocidos por su alta eficiencia energética y su alto rendimiento. El microcontrolador STM32F401CEU6 que se encuentra en la placa cuenta con una arquitectura de procesador ARM CortexM4 de 32 bits y una velocidad de reloj de hasta 84 MHz. Esta arquitectura permite un alto rendimiento de procesamiento de datos y la posibilidad de realizar tareas complejas en tiempo real[1]

La placa ofrece una amplia gama de características, incluyendo interfaces de comunicación como USB, UART, SPI y I2C. Estas interfaces permiten la comunicación con otros dispositivos y la transmisión de datos. Además, la placa cuenta con una gran cantidad de pines GPIO que permiten la conexión de sensores y actuadores, lo que permite la implementación de proyectos de electrónica y robótica, también cuenta con una memoria Flash de 512 KB y SRAM de 128

KB, lo que permite el almacenamiento de programas y datos. Esto es especialmente importante en proyectos donde se requiere una gran cantidad de memoria para el procesamiento de datos complejos y la implementación de algoritmos de control.

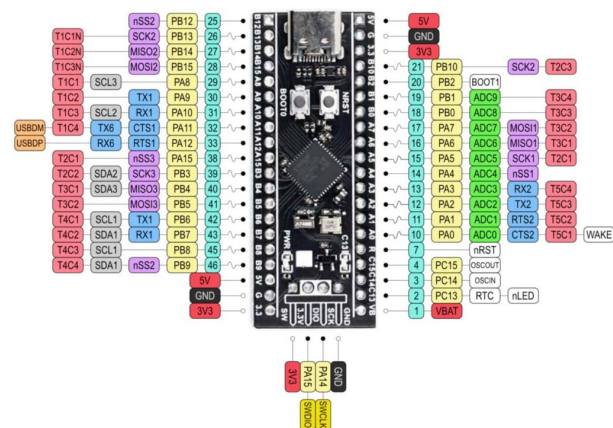


Figura 1. Placa STM32F401CEUx

El VS1838B es un receptor infrarrojo universal que detecta pulsos de 38 KHz de mandos a distancia comunes. Es ideal para proyectos de electrónica que implementen control remoto inalámbrico de corto alcance mediante infrarrojos. Se puede usar con Arduino, microcontroladores PIC, Raspberry Pi o cualquier circuito digital. También es un reemplazo ideal si el receptor infrarrojo de tu circuito o dispositivo se ha dañado.

Especificaciones:

- Tipo de receptor: Luz infrarroja
- Número de pines: 3
- Voltaje Vcc: 2.7V a 5.5V DC
- Corriente Icc: Min. 0.4mA y Max. 1.5mA
- Distancia de recepción: Min. 18m y Typ. 20m
- Frecuencia: 38 KHz
- Ángulo de recepción: $\pm 45^\circ$ Atenuación de distancia 1/2

Aplicaciones:

- Control remoto inalámbrico
- Proyectos y prácticas de electrónica
- Módulos de control IR
- Equipo visual (Audio, TV, Reproductores)
- Electrodomésticos (Aire acondicionado, Luz eléctrica)
- Otros equipos de recepción IR



Figura 2. Sensor Receptor Infrarrojo IR – VS1838B

Un protocolo de infrarrojos (IR) es un conjunto de reglas y estándares que define cómo los dispositivos transmiten y reciben datos a través de señales infrarrojas. Estos protocolos son utilizados comúnmente en sistemas de control remoto, como los que se encuentran en televisores, reproductores de DVD, acondicionadores de aire, sistemas de audio, etc.

Algunos de los protocolos de infrarrojos más comunes incluyen:

1. RC-5: Desarrollado por Philips, es uno de los protocolos más antiguos y comunes. Define una serie de códigos de pulsos y espacios para representar comandos de control remoto.
2. NEC (National Electronics Code): Otro protocolo comúnmente utilizado, especialmente en aplicaciones de electrónica de consumo. Utiliza una modulación de pulsos en serie para codificar datos.
3. Sony SIRC (Sony Infrared Remote Control): Utilizado en muchos dispositivos de Sony, este protocolo tiene varias variantes y se caracteriza por su simplicidad y eficiencia.
4. RC-6: Una evolución del protocolo RC-5 de Philips, que ofrece características adicionales como la transmisión de datos bidireccionales y soporte para dispositivos de control remoto universales.
5. Samsung: Utilizado por los dispositivos de Samsung, este protocolo también tiene sus propias especificaciones de codificación de señales infrarrojas.

Estos protocolos definen aspectos como la longitud de onda de la señal infrarroja, la modulación de la señal, la codificación de los datos y los códigos específicos para representar diferentes comandos. Es importante que los dispositivos que interactúan entre sí utilicen el mismo protocolo o sean capaces de interpretar los protocolos utilizados por los otros dispositivos.

El protocolo NEC establece que primero se emite 9ms de una señal alta, y luego 4.5ms de señal baja para empezar el envío de una señal de 8 bits, empezando por el bit más bajo LSB hasta el más alto MSB y luego su complemento. Todo en una señal de 38kHz donde se modula primero la dirección y luego el comando



Figura 3. Trama de bits protocolo NEC

Una señal con el protocolo NEC es una especie de señal PWM donde un '0' es un ciclo de 1.125ms la emisión de una señal de

0.565ms y un '1' lógico o mismo pero en un intervalo de 2.25ms, se ve mejor con un dibujo:

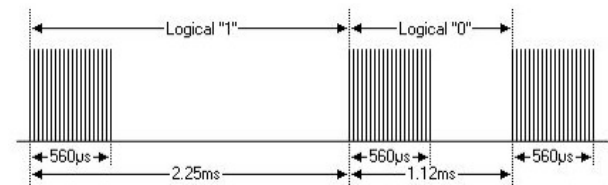


Figura 4. Señal protocolo NEC

III. DIAGRAMA DE FLUJO

El diagrama de flujo se concentra en el paso a paso de los dos modos utilizados, modo run y modo set, teniendo en cuenta la configuración de las tramas para cualquier control que se vaya a utilizar.

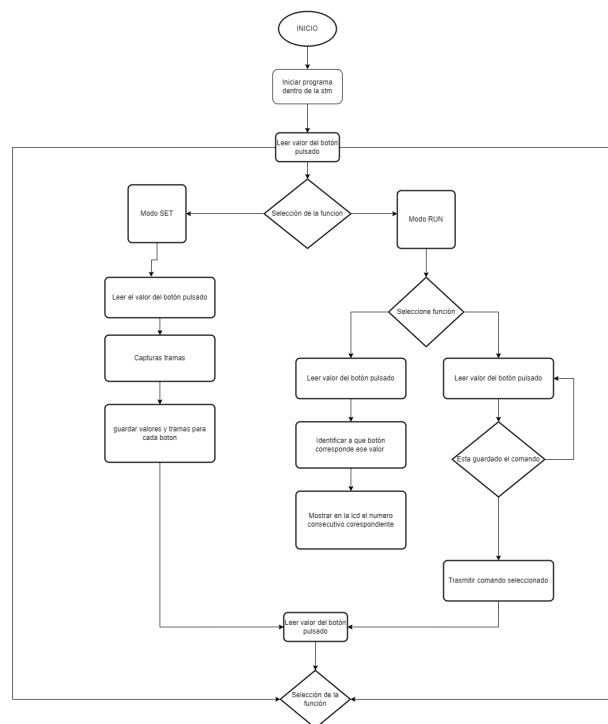


Figura 5. Diagrama de flujo – Cambios de modo

IV. PROCEDIMIENTO

Comenzando con la comprensión del laboratorio, inicialmente se llevó a cabo la verificación de la conexión del TSOP1838, probando cada una de sus entradas para determinar dónde se encuentra su salida. Una vez identificada la configuración, se procede a conectarla para permitir la lectura del pin del receptor y comenzar a identificar sus tramas.

Para poder identificar sus tramas, se tuvo en cuenta el tipo de protocolo a los que pertenecen los controles elegidos, para esto utilizamos el osciloscopio, medimos el tiempo del start para medir los tiempos de bit y otros parámetros relevantes de las tramas y se obtiene el protocolo NEC en ambos controles.

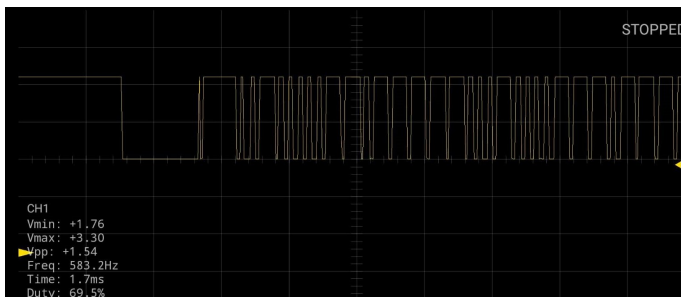


Figura 6. Onda osciloscopio

En la implementación del código, se configuro el módulo TIM del microcontrolador para contar los tiempos de bit, se utilizo el módulo TIM para capturar y clasificar los comandos enviados por el control remoto y de esta forma obtener el modo set, se almacena la información de los comandos y para mostrar los botones guardados en el modo run, se visualiza cada boton que se presiona con el numero consecutivo que se guardo.

En esto código miramos las tramas, los ceros y unos que obtenemos.

```
switch(estado)
{
case 0://para inicializar el tiempo de subida y guardar
uint16_t x = __HAL_TIM_GET_COUNTER(&htim2);
if(x > 6)//guarda un 1 o 0
dependiendo de cuanto tiempo se demore
{
    cero [posicion]=1;
}
else
{
    cero [posicion]=0;
}
//cero [posicion]=__HAL_TIM_GET_COUNTER(&htim2);

__HAL_TIM_SET_COUNTER(&htim2,0);//reinicia el counter

estado = 1;//envia a estado 0 para guardar y recetear el counter
break;

case 1://para inicializar el tiempo de bajada y guardar

uno [posicion]= __HAL_TIM_GET_COUNTER(&htim2);
__HAL_TIM_SET_COUNTER(&htim2,0);//reiniciar el counter
//if(posicion < 40)
//{
    posicion ++;//cambia la posicion en el arreglo
//}

estado = 0;//envia a estado 1 para guardar y recetear el counter
break;
}
```

En general las principales características y funcionalidades del código son, en primer lugar, como debe hacerse en todos los proyectos se configuran los periféricos y se configura el sistema, incluida la configuración del reloj del microcontrolador, para que se genere el código de inicialización de los timers y las interrupciones.

En segundo lugar, se definen algunas macros y variables globales necesarias para el funcionamiento del programa, incluidos los arreglos para almacenar datos de comandos IR de la trama completa, las variables de estado para cambiar de modos o controlar y limitar el programa y una estructura para controlar un LCD.

Luego se inicializan los pines GPIO y el temporizador TIM2, que se usan para manejar los tiempos en las lecturas de los pulsos IR y se definen algunas funciones auxiliares, como customPow() para calcular la potencia de 2 para los bits y las funciones modo_set() y modo_run(). En el programa principal se ejecuta un bucle infinito en el que se monitorea el estado del modo de operación, osea el set y el run y se realizan las acciones correspondientes según el estado y las entradas recibidas.

Específicamente el programa maneja las interrupciones generadas por cambios en los pines GPIO, especialmente el pin IR, que se usa para detectar pulsos de un control remoto IR.

V. CONCLUSIONES

- Nos fijamos que si no se tiene en cuenta las banderas de inicio y de final, se podrian tener muchos mas controles, pero en estos cada uno de ellos tiene tiempo y espera de formas diferentes, por lo cual provocando errores al momento de la toma de datos de los controles, restandole o sumandole información que no es necesaria.
- Considerando diferentes soluciones, se exploraron algunas soluciones para minimizar la cantidad de datos, aunque esto implicaria la pérdida de información de uno u otro control. Otra opción considerada fue el uso de una bandera, pero si el control no se cambia, sería necesario realizar un análisis del nuevo control en caso de cambio.
- Se definio un control para trabajar mejor el reconocimiento de los botones y disminuir los errores al configurar los comandos, definiendo banderas de inicio y final, en donde se soluciono la toma de los botones y la recepción del comando.
- Se debe tener en cuenta la información de los controles, ya que, la lectura obedece al tiempo en que se mantenga oprimido cada comando, esto dependeria del fabricante o la construcción del control.
- Finalmente, no se logro la transmisión de los comandos guardados al televisor, debido a varios problemas de conexión y de logica en la parte de transmisión.

VI. ANEXOS

VI-A. Esquemático

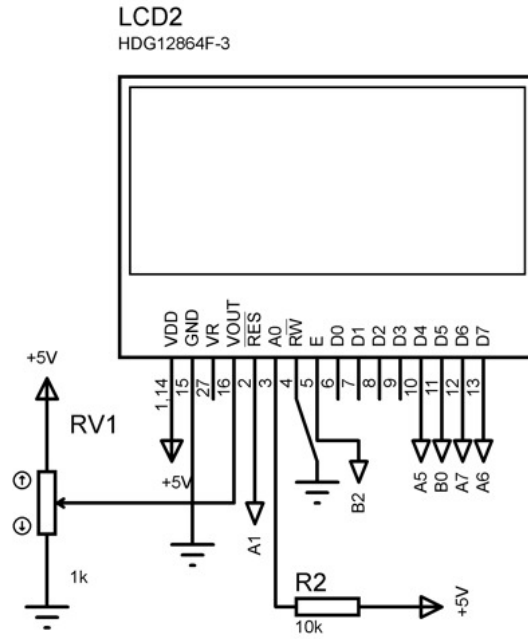


Figura 7. Esquemático conexión de la lcd

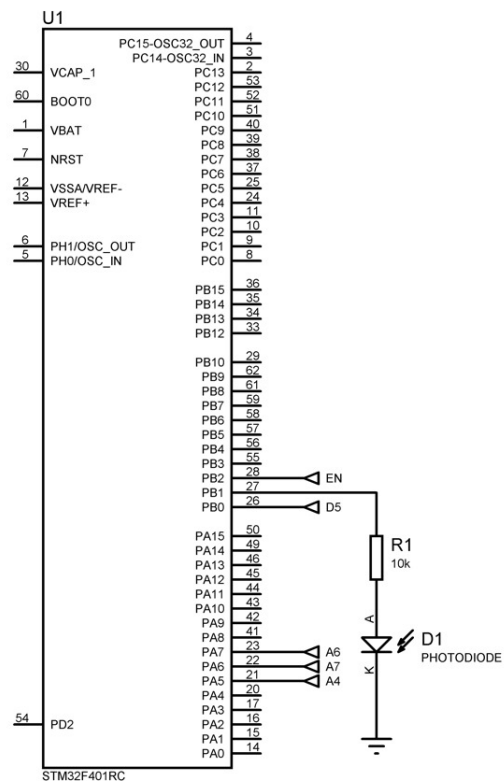


Figura 8. Esquemático stm32

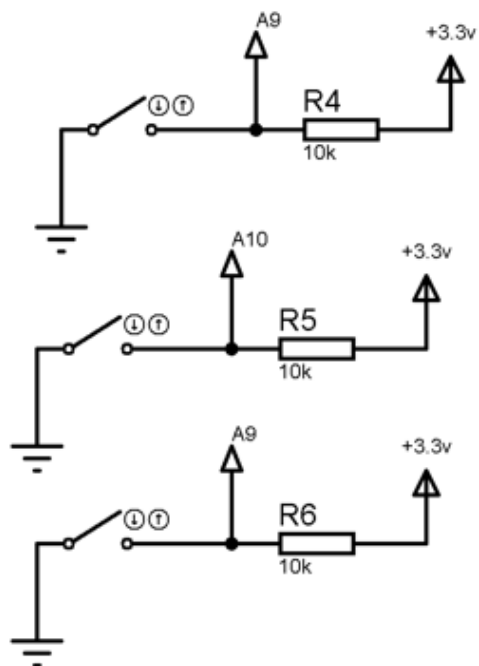


Figura 9. Esquemático botones

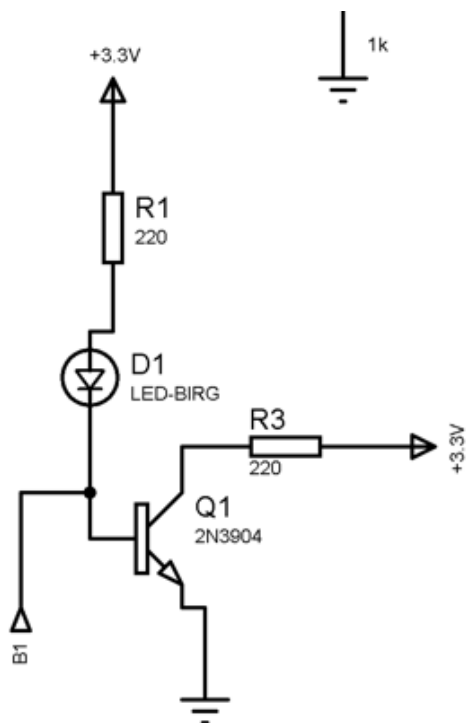


Figura 10. Esquemático TSOP1838

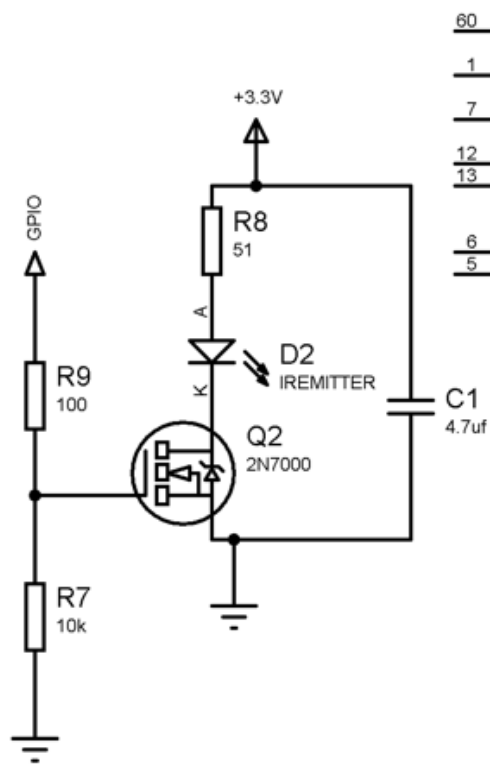


Figura 11. Esquemático IR emisor

VI-B. Diagrama de flujo

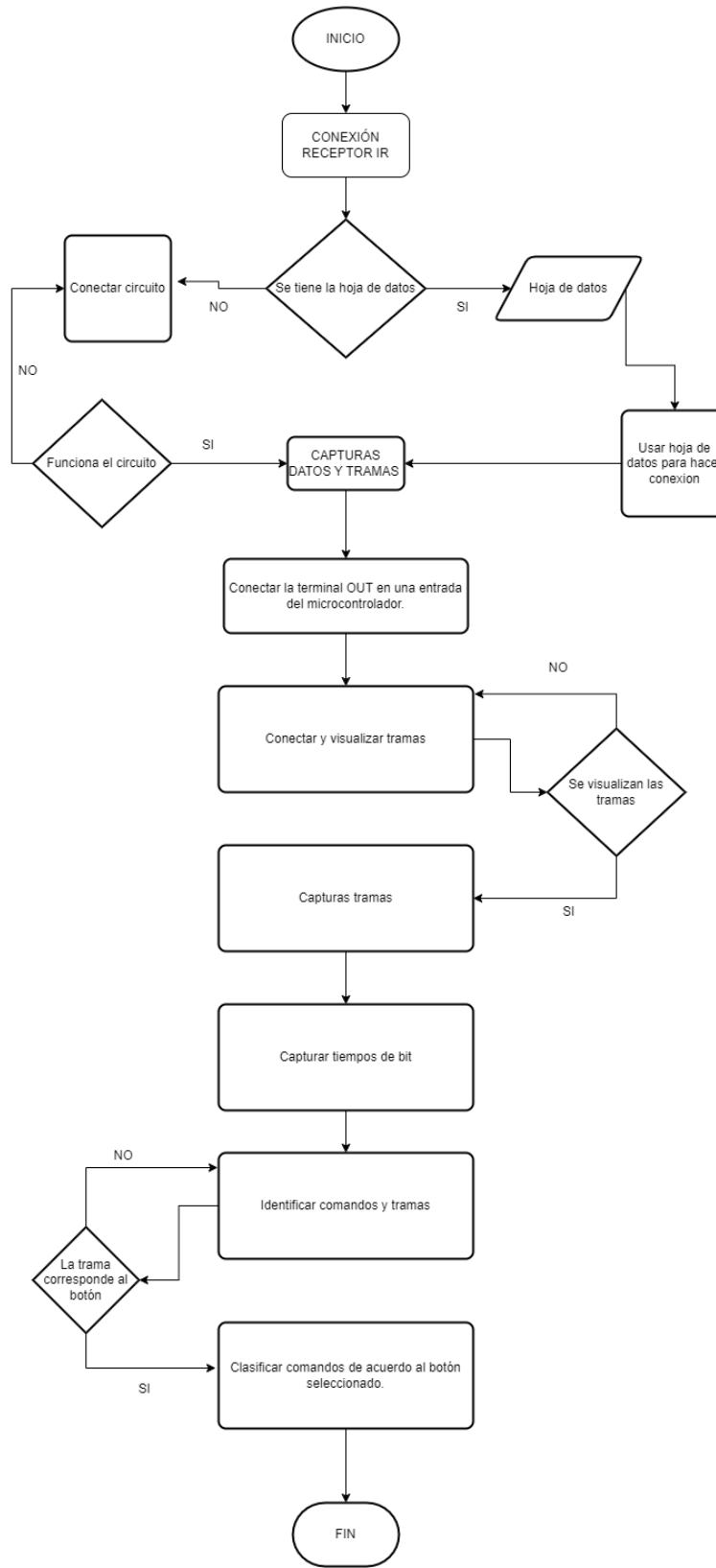


Figura 12. Diagrama de flujo configuración tramas y botones

VII. REFERENCIAS

[1] (S/f). Uelectronics.com. Recuperado el 9 de marzo de 2024, de <https://uelectronics.com/producto/receptor-infrarrojo-vs1838b-ir-fototransistor/>