

NOTA DE APLICACION

“MODULOS RF-433MHZ WENSHING”

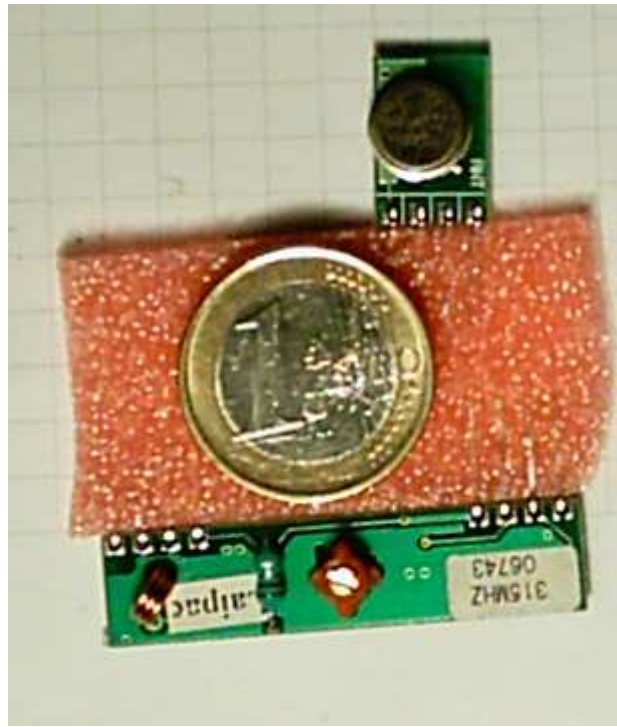
PROFESOR TITULAR: Ing. Oros, Ramón

CURSO: 5R1

Introducción

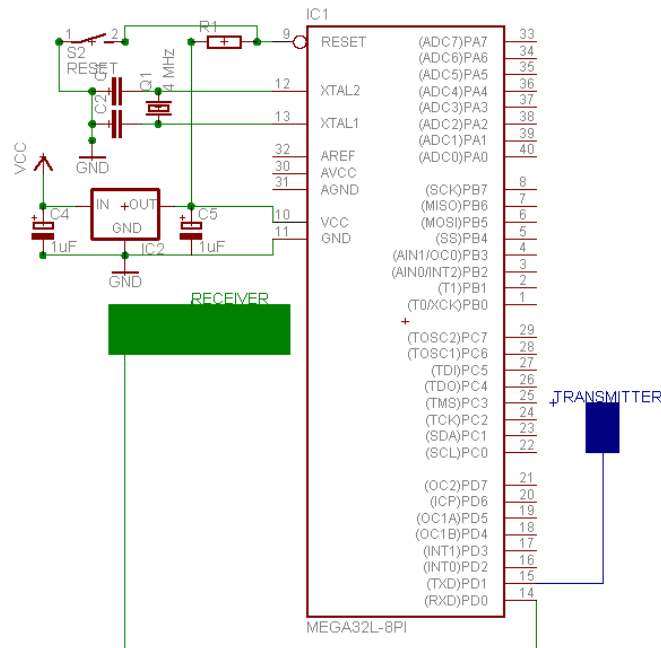
La presente nota de aplicación es sobre la utilización de módulos rf que funcionan a 433Mhz UHF (Ultra High Frequency). Estos son ideales para aplicaciones donde la simpleza debe ser máxima ya que son extremadamente fáciles de utilizar. Salvándonos así de los numerosos inconvenientes en la parte de RF.

Estos módulos no son muy rápidos (max 10kbps, con Manchester 5kbps) pero son muy económicos (10u\$ emisor + receptor) y de fácil utilización. El código implementado tiene detección de errores una trama y codificado esto también se podría adaptar a otros medios de comunicación.



A continuación se presentan dos códigos el del receptor y el emisor. Se utiliza un protocolo muy simple, aunque tiene control de errores por checksum y timeouts, devuelve un código por el puerto serie dependiendo de cual sea el error. Todo es codificado usando código Manchester

En este ejemplo se utiliza un microcontrolador atmel, pero podría ser cualquiera ya que los requisitos de procesamiento no son muy grandes. Usa un cristal de 4Mhz (esto es importante ya que todos los tiempos están basados en esto, habría que arreglarlo para que se pueda configurar).



Transmisor:

Fue utilizado el TWS-BS3 (433.92 MHz)

Al principio se empiezan a recibir 1 y 0, de cualquier duración debido a que el receptor estaba apagado, y hasta que se estabilice pueden pasar varios ms esto puede ser de hasta 300ms, probado 50ms, estoy hay que volver a hacerlo cuando no se haya transmitido por un tiempo. Esto lo hace la función RF_TX_Header.

Luego la función RF_TxSendFrame envia el paquete con el siguiente protocolo
 <STX><STX><DR><DT><LEN><DATA><CHKSUM><ETX>

Envia la cantidad de Bytes RF_TX_Len desde la posicion RF_TX_Buffer de la memoria, solo manda ascii (en realidad no se puede mandar un byte que sea igual a stx o etx), en el check sum no se considera el stop ni el mismo checksum.

En el .h RF_TX_macs.h se puede configurar la pata y el puerto que se va a usar para enviar los datos, Data In en este caso PORTC, 4. También se puede configurar cuanto es el tiempo de un bit con RF_Unidad en us. Ej .200 seria 200us por bit.



pin 1 : GND
pin 2 : Data in
pin 3 : VCC
pin 4 : ANT

Ejemplo de uso (desde el main por ejemplo)

```
call      RF_TX_Header

bankssel  RF_TX_Buffer    ;selecciona el banco de buffer
movlw     0xaa             ;va a mandar 0xaa
movwf     RF_TX_Buffer+0x04
movlw     0xbb             ;va a mandar 0xbb
movwf     RF_TX_Buffer+0x05
movlw     0xcc
movwf     RF_TX_Buffer+0x06
movlw     0xdd
movwf     RF_TX_Buffer+0x07
movlw     0xff
movwf     RF_TX_Buffer+0x08
movlw     0x55
movwf     RF_TX_Buffer+0x09
movlw     0x07
movwf     RF_TX_Buffer+0x0a
bankssel  STATUS           ;banco cero
movlw     .6
movwf     RF_TX_Len        ;cantidad a mandar
call      RF_TxSendFrame
```

Receptor:

Se utilizo un RWS-371-6 (433.92 MHZ)

Al principio se empiezan a recibir 1 y 0, de cualquier duracion debido a que el receptor estaba apagado, y hasta que se estabilice pueden pasar varios ms esto puede ser de hasta 300ms, probado 50ms y anda, Luego de esto se recibe un pulso bajo de 1 ms y a contignación la trama.

Usa RB0 como entrada de datos conectado a la pata 3 del receptor (lineal output).

Cuando RB0 cambia de nivel se dispara la interrupción.

Se usa tmr0 con prescaler de 16 para los timeouts.

Cuando aparece un 1 en rbo, se activa la interrupción, entonces llama a la rutina RF_RxWorUP,

en esta rutina, primero espera que vuelva a 0, esto tiene un time out de 4ms en caso que se clave en 1 por una falla, después verifica que quede en cero 1 ms, si viene un 1 antes del ms vuelve al principio de la rutina y hace todo de nuevo, si viene después retorna con 1 en el carry indicando exito.

En la interrupción verifica el carry es 1, si es cierto indica que es un header valido, y se dispone

a recibir mas datos, si es cero, era ruido, o que todavía se esta estabilizando o algo esta mal.

A continuación se llama a la rutina RF_RxGetFrame, esta espera un frame del siguiente formato:

<STX><DR><DT><LEN><DATA><CHKSUM><ETX>, cada byte es obtenido por medio de RF_RxGetByte. Se hace un control de checksum.

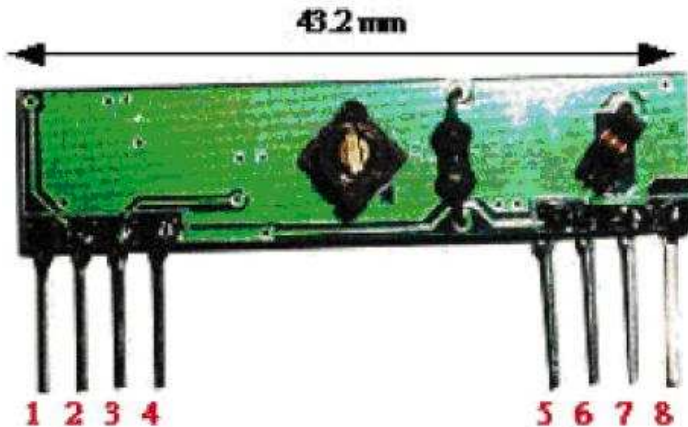
Cada dato es ubicado en RF_RX_Buffer que esta en el banco 1.

Si esta función devuelve carry 0, es error y el codigo esta en la variable RF_RxErrorCode.

Función RF_RxGetByte:

Cada byte consta de un byte de start, 8 de datos y uno de stop. Cada bit esta codificado en Manchester

El valor es devuelto de en RF_RxRXReg.



pin 1 : Gnd
 pin 2 : Digitak Output
 pin 3 : Linear Output
 pin 4 : Vcc
 pin 5 : Vcc
 pin 6 : Gnd
 pin 7 : Gnd
 pin 8 : ANT (About 30 - 35 cm)

Supply Voltage : 5v dc

RB0 esta conectada pata 3 del receptor (lineal output)

Ejemplo del receptor (tiene que ir en la rutina de interrupción por cambio de flanco de RB0):

```

ServiceRB0
    call    RF_RxWorUP      ;verifica si vino un header valido
    BNC     Salir           ;si no es carry sale por un error
    call    RF_RxGetFrame   ;recibe los datos
    BC     Acepto           ;si no es carry Salir por un error
    call    EnviarCR        ;retorno de carro
    movf    RF_RxErrorCode,w
    call    EnviarBinAscii0X ;envio el error
    goto    Salir
;si llega aca ya estan recibidos los datos, solo los muestra
Acepto
    bsf     PORTC,0
    Mensaje Table1         ;envia "los datos son:"

    banksel RF_RX_Buffer   ;bank 1
    movf    RF_RX_LEN,w    ;saca cuantos bytes hay
    banksel STATUS
    call    EnviarBinAscii0X ;envia en forma de hexa
                           ;por el uart
    call    EnviarCR        ;retorno de carro

    banksel RF_RX_Buffer
    RF_RX_DATA    0        ;macro que pone en w, el valor
                           ;de rx_buffer + 0
    banksel STATUS        ;bank 0
    call    EnviarBinAscii0X ;envia en forma de hexa

Salir:    bcf     INTCON,INTF ;resetea inter flag
    goto    EndISR        ;vuelve
  
```

Se usa código "relocatable" para que no se mezcle todo, el es necesario el archivo *.lkr para compilarlo, este dice en que sección de código va cada cosa.
Para usarlo hay que incluir RF_tx.inc y RF_rx.inc, además del anterior código en la interrupción.

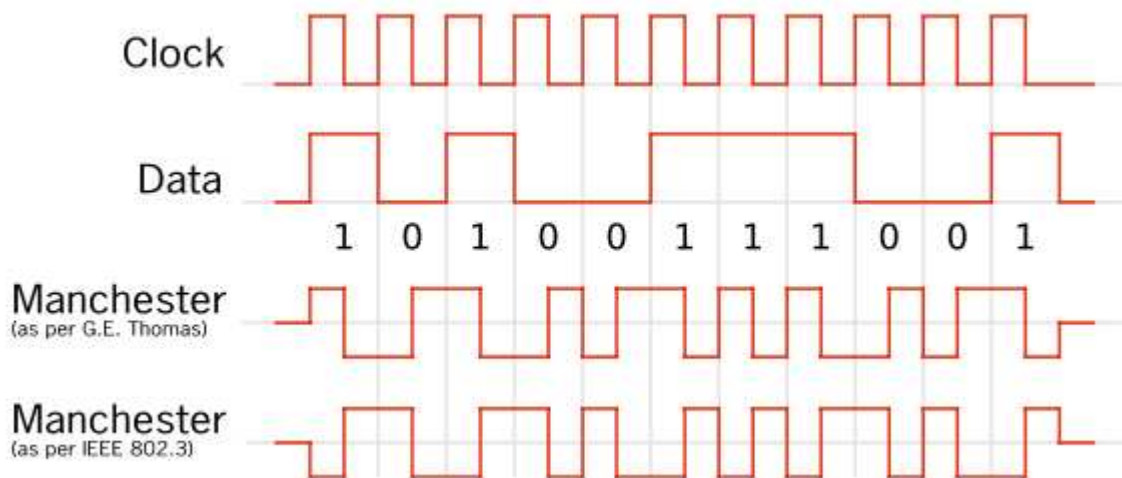
Codificación Manchester

La codificación Manchester, también denominada codificación bifase-L, es un método de codificación eléctrica de una señal binaria en el que en cada tiempo de bit hay una transición entre dos niveles de señal. Es una codificación autosincronizada, ya que en cada bit se puede obtener la señal de reloj, lo que hace posible una sincronización precisa del flujo de datos. Una desventaja es que consume el doble de ancho de banda que una transmisión asíncrona. Hoy en día hay numerosas codificaciones (8B/10B) que logran el mismo resultado pero consumiendo menor ancho de banda que la codificación Manchester.

La codificación Manchester se usa en muchos estándares de telecomunicaciones, como por ejemplo Ethernet.

Descripción

- Las señales de datos y de reloj, se combinan en una sola que auto-sincroniza el flujo de datos.
- Cada bit codificado contiene una transición en la mitad del intervalo de duración de los bits.
- Una transición de negativo a positivo representa un 1 y una transición de positivo a negativo representa un 0.



Ejemplo de codificación Manchester, de acuerdo con las convenciones Ethernet

Los códigos Manchester tienen una transición en la mitad del periodo de cada bit. Cuando se tienen bits iguales y consecutivos se produce una transición al inicio del segundo bit, la cual no es tenida en cuenta por el receptor al momento de decodificar, solo las transiciones separadas uniformemente en el tiempo son las que son consideradas por el receptor. Hay algunas transiciones que no ocurren a mitad de bit. Estas transiciones no llevan información útil, y solo se usan para colocar la señal en el siguiente estado donde se llevará a cabo la siguiente transición. Aunque esto permite a la señal auto-sincronizarse, en realidad lo que hace es doblar el requerimiento de ancho de banda, en comparación con otros códigos como por ejemplo los Códigos NRZ.

La codificación Manchester como Modulación por desplazamiento de fase

La codificación Manchester es solo un caso especial de la Modulación por desplazamiento de fase, donde los datos que van a ser transmitidos controlan la fase de una onda rectangular portadora. Para controlar la cantidad de ancho de banda consumida, se puede usar un filtro para reducir el ancho de banda hasta un valor bajo como 1Hz por bit/segundo, y mantenerlo para no perder información durante la transmisión.

Ventajas y desventajas del uso de la codificación Manchester

Como ventajas principales se pueden destacar las siguientes:

* La codificación Manchester o codificación bifase-L es autosincronizada: provee una forma simple de codificar secuencias de bits, incluso cuando hay largas secuencias de periodos sin transiciones de nivel que puedan significar la pérdida de sincronización, o incluso errores en las secuencias de bits. Por ello es altamente fiable.

* Detección de retardos: directamente relacionado con la característica anterior, a primera vista podría parecer que un periodo de error de medio bit conduciría a una salida invertida en el extremo receptor, pero una consideración más cuidadosa revela que para datos típicos esto llevaría a violaciones de código. El hardware usado puede detectar esas violaciones de código, y usar esta información para sincronizar adecuadamente en la interpretación correcta de los datos.

* Esta codificación también nos asegura que la componente continua de las señales es cero si se emplean valores positivos y negativos para representar los niveles de la señal, haciendo más fácil la regeneración de la señal, y evitando las pérdidas de energía de las señales.

Las principales desventajas asociadas son las siguientes:

- Ancho de banda del doble de la señal de datos: una consecuencia de las transiciones para cada bit es que el requerimiento del ancho de banda para la codificación Manchester es el doble comparado en las comunicaciones asíncronas, y el espectro de la señal es considerablemente más ancho. La mayoría de los sistemas modernos de comunicación están hechos con protocolos con líneas de codificación que persiguen las mismas metas, pero optimizan mejor el ancho de banda, haciéndolo menor.

Pruebas de laboratorio

Según las hojas de datos se puede obtener una tasa de transmisión de hasta 9600 baudios, sin embargo no se logro superar los 4800. Teniendo en cuenta que un baudio es la cantidad de veces que la señal cambia por segundo y debido a que se utilizo codificación Manchester, se logro transmitir solamente 2400bps.



Antena

Debido a que la antena debe ser de al menos 0.25 de la longitud de onda (433Mhz), teniendo en cuenta que la longitud de onda para esta frecuencia es 0.7m, el largo de la antena debe ser de aproximadamente 17cm.

Posible utilización con decodificadores integrados

A continuación se presenta la utilización con codificadores integrados, estos se utilizan por ejemplo en las alarmas de los autos. Son mas simples que con un microcontrolador pero presentan el inconveniente de tener que agregar otro CI mas, y que no es posible encriptar la señal. Debido a esto es que por ejemplo existen casos en que el usuario quiere activar la alarma de su auto y abre el portón automático de su casa.

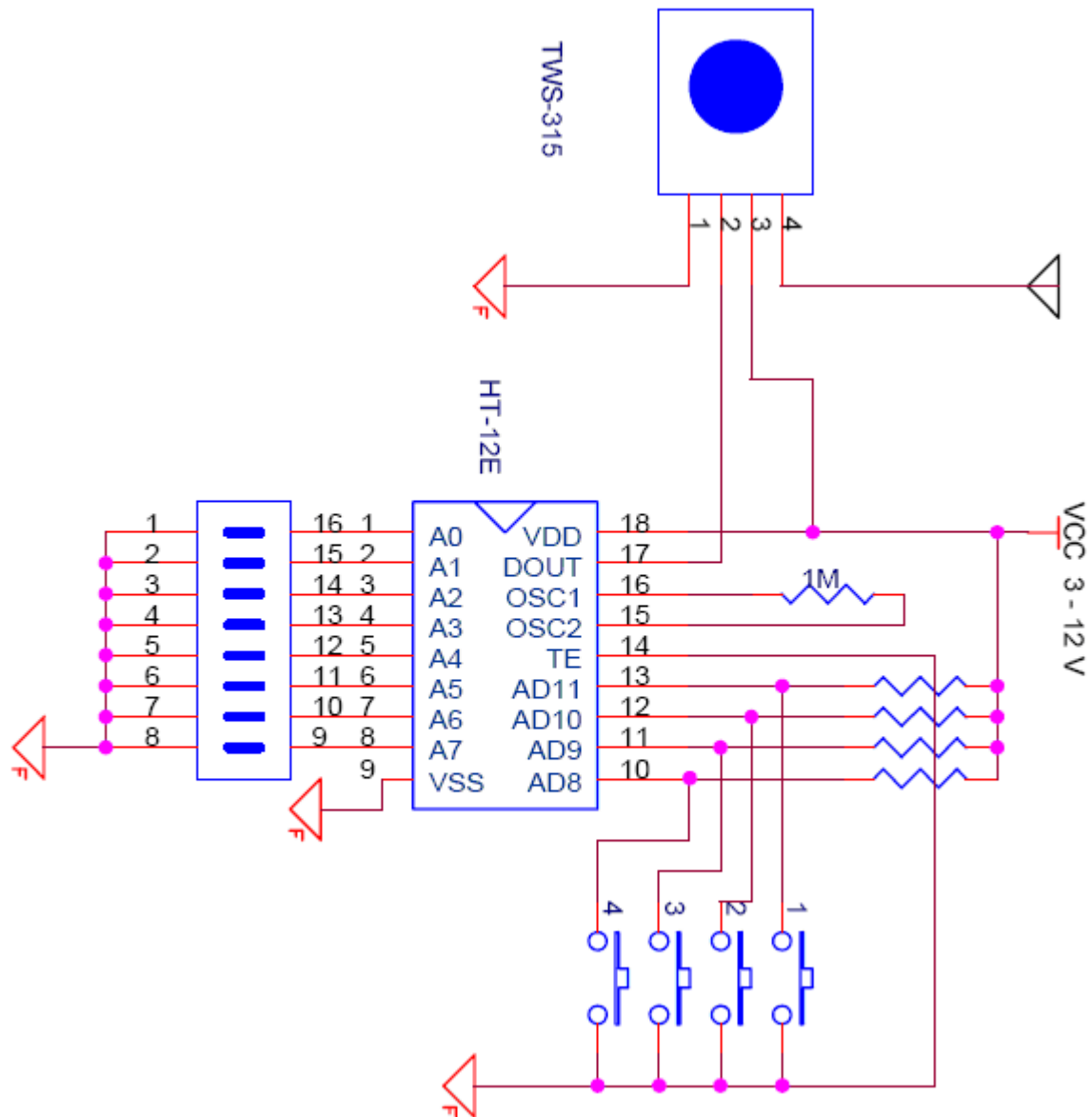


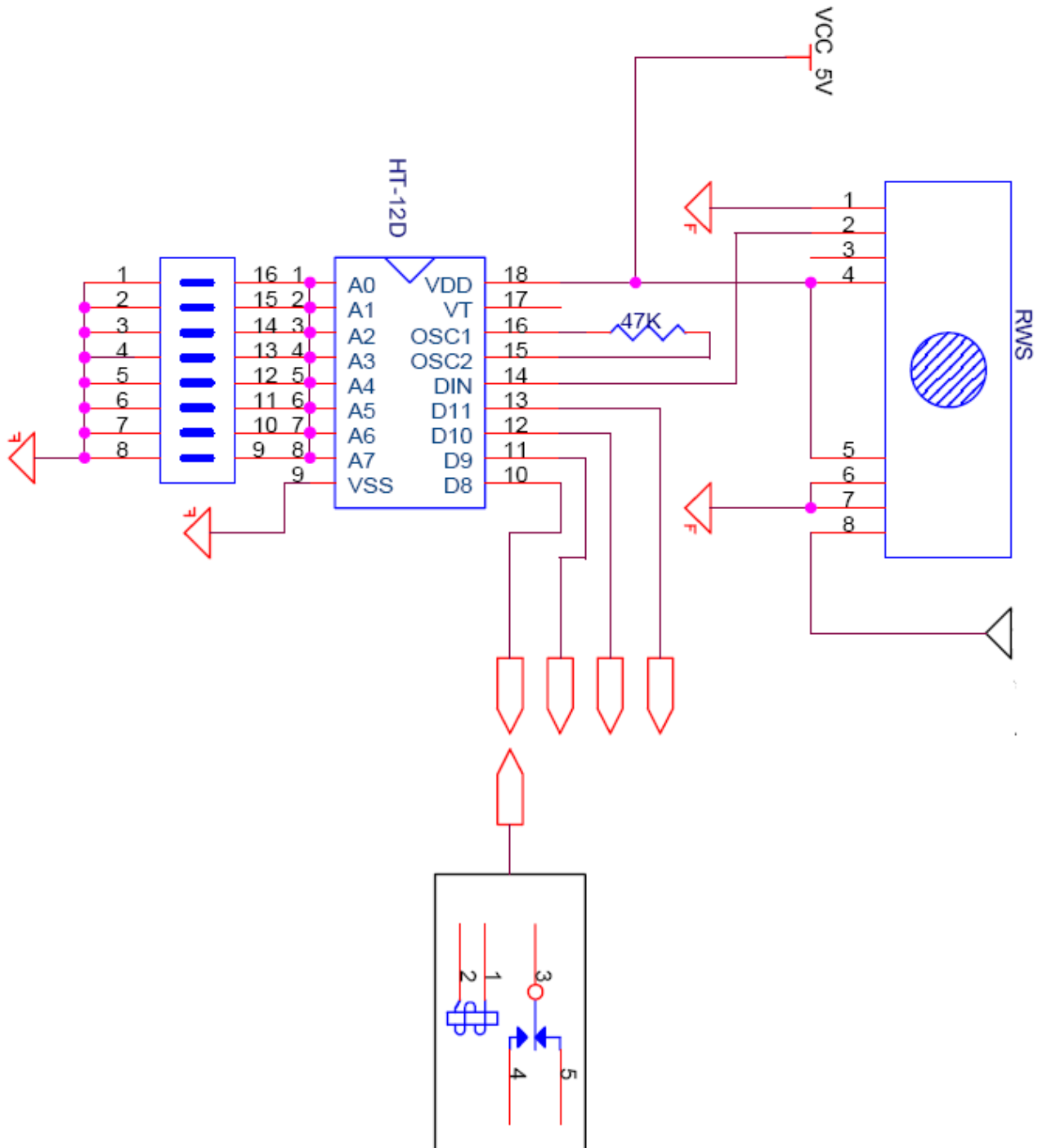
Electrónica Aplicada III

Gusella, Angel Maximiliano (47851)

Negri, Matias (49844)

Remonda, Adrian (48366)





Conclusión

Utilizar este tipo de módulos simplifica notablemente el trabajo ya que a estos solo hay que enviarles la información en forma digital a una baja frecuencia evitando así el uso de poderosos y caros procesadores. Estos se encargan de modular y transmitir dicha información, para luego ser recibida por el receptor. Ahorrándose así los inconvenientes del diseño de placas RF.