

Reporte de prácticas tema 4

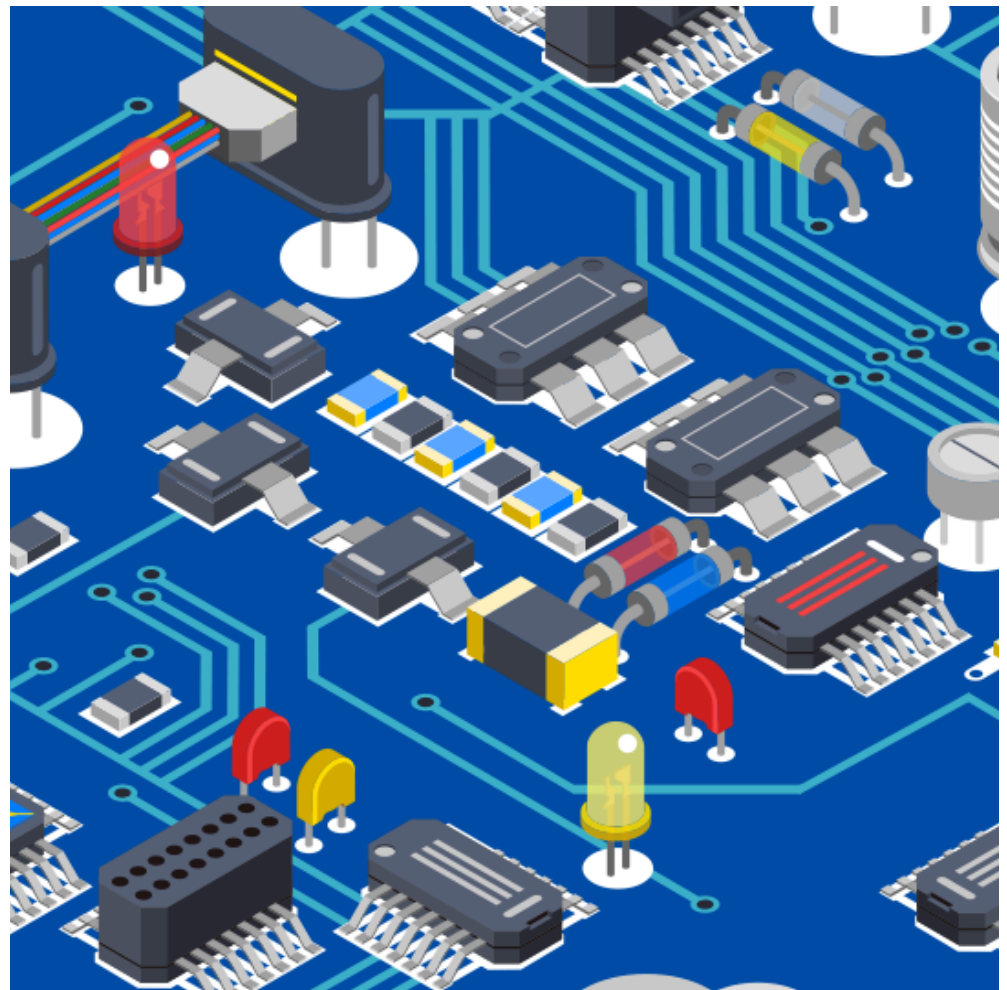
Catedrático: Ing. Germán Godínez Cardoza

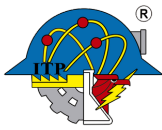
Alumnos:

- | | |
|-----------------------------------|----------------|
| • Arciniega Tlapaya Alfredo | N.C.: 21200576 |
| • Garrido Virgen Jessel Alejandro | N.C.: 21200602 |
| • Morales Mateos Johanna | N.C.: 21200619 |

Fecha: 21 de junio 2024

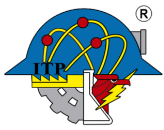
L e n g u a j e s d e I n t e r f a z





Guía de observación proyecto

| Lenguajes de Interfaz | | | | | | | | | | | | |
|--|--|------------|------------------------|----|-------------------|---------------------------|-------------------|-----------------|-----------------------|---|---|---|
| Guía de Observación | | | | | | | | | | | | |
| Programa Educativo: | Ingeniería en Sistemas Computacionales. | | | | | Fecha: | 17/06/2024 | | | | | |
| Nombre del docente: | Germán Godínez Cardoza. | | | | | | | | | | | |
| Nombre los integrantes del equipo: | 1- Alfredo Arriaga Tapia 2- Johana Morales Mejias 3- José Alejandro Garrido V. | | | | | Semestre: | 6 | Número de Tema: | 1 | | | |
| | | | | | | Clave: | 6X6 | Tema: | | | | |
| | | | | | | Grupo: | B | | | | | |
| Nombre de la(s) práctica(s): | Proyecto Final | | | | | Número(s) de práctica(s): | | | | | | |
| Instrucciones: | | | | | | | | | | | | |
| 1. El alumno deberá completar los campos: Nombre de los integrantes del equipo, fecha, tema, número de tema, número(s) de práctica(s) y nombre(s) de la práctica(s). | | | | | | | | | | | | |
| 2. El docente llenará la guía de observación en función de la calidad del desempeño observado por el (los) estudiantes. | | | | | | | | | | | | |
| 3. Se marca con una "X" si cumple o no el criterio. | | | | | | | | | | | | |
| 4. Se llenará el apartado "Puntos" con los puntos que considere correspondan a la calidad del desempeño. | | | | | | | | | | | | |
| 5. El puntaje máximo de la evaluación es de 100 puntos. | | | | | | | | | | | | |
| 6. Realizar la sumatoria. | | | | | | | | | | | | |
| No. | Criterio | Valor | IND | SI | No | Puntos | Retroalimentación | | | | | |
| 1 | Cuenta con los componentes requeridos para realizar la práctica y termina a tiempo. | 20 | | | | | | | | | | |
| 2 | La información y resultados de la práctica se encuentran registrados en su cuaderno. | 15 | | | | | | | | | | |
| 3 | Trabaja de manera ordenada el desarrollo y ensamble de su práctica. | 10 | | | | | | | | | | |
| 4 | Utiliza de manera correcta las herramientas e instrumentos de medición. | 15 | | | | | | | | | | |
| 6 | Desarrolla la práctica colaborando con su equipo de trabajo. | 10 | | A | | | | | | | | |
| 7 | Conoce y expone correctamente los fundamentos teóricos de la práctica. | 20 | | B | | | | | | | | |
| 8 | Trabaja de manera autónoma las actividades de la práctica. | 10 | | F | | | | | | | | |
| Puntaje Máximo: | | 100 puntos | | | Puntaje obtenido: | | 80 | | | | | |
| Evidencia de aprendizaje | | % | Indicadores de alcance | | | | | | Métodos de evaluación | | | |
| | | | A | B | C | D | E | F | Instrumento | P | C | A |
| Práctica | | | 5 | 10 | | | | 5 | Guía de Observación | X | X | X |
| Nombre y firma del estudiante o integrantes del equipo: | | | | | | | | | | | | |
| José Alejandro Garrido V. | | | | | | | | | | | | |
| Johana Morales Mejias | | | | | | | | | | | | |
| Alfredo Arriaga Tapia | | | | | | | | | | | | |
| Nombre y firma del docente: | | | | | | | | | | | | |
| Germán Godínez Cardoza | | | | | | | | | | | | |

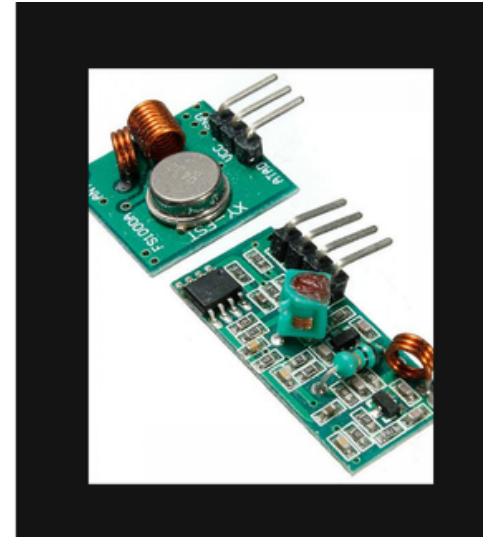


Presentación del proyecto

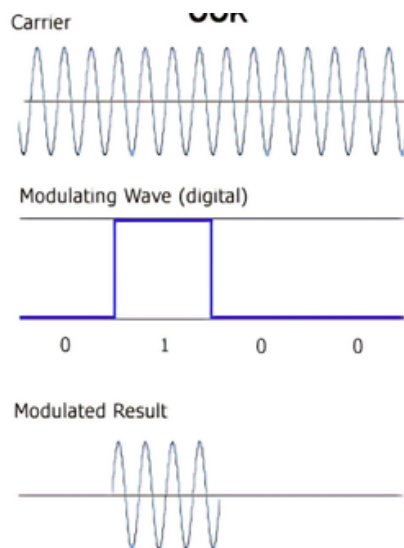


Comunicación inalámbrica con los módulos RF 433MHz

Equipo,
Lenguajes de Interfaz.
Ing. Germán Godínez Cardoza



Portada del proyecto



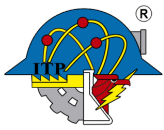
Introducción del proyecto

Proyecto

Proyecto de comunicación inalámbrica con los módulos RF de modulación **OOK**.

Es una **forma simple** de modulación de amplitud (AM) en la que la **presencia o ausencia** de una portadora de RF se utiliza para transmitir datos binarios (0s y 1s).

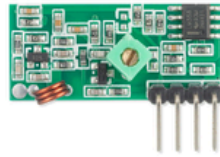
Utiliza un protocolo de comunicación que implica la transmisión de datos mediante señales de control y tiempos de retardo específicos para sincronizar la transmisión y recepción de bits. Este puede ser considerado una forma de **comunicación asincrónica simple**.



Presentación del proyecto

01

Receptor



02

Emisor



03

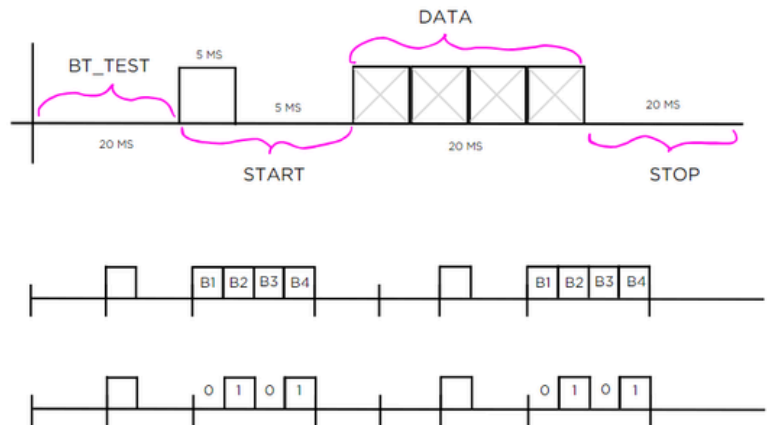
Microcontroladores



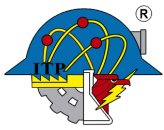
Componentes

Componentes del proyecto

Lógica



Lógica para implementar el proyecto



Nombre: Proyecto final

Procedimiento:

Para nuestro proyecto decidimos realizar una comunicación inalámbrica con un con módulos de frecuencia F1 y F2. El F1 tiene una frecuencia de 100 kHz y el modulo F2 una frecuencia de 150 kHz. Se conectaron los dispositivos al microcontrolador, configurando el puerto A como entrada para recibir señales y el puerto B como salida para transmitir datos.

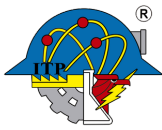
Con las configuraciones ya echas hicimos el programa en lenguaje ensamblador con el IDE MPLAB. A grandes rasgos el propósito del código es esperar una señal de inicio, leer los datos de 4 bits del puerto A, y enviarlos en serio por el puerto B..

En nuestro bucle principal el código monitoriza los cambios de estado en los pines PBO y PB1 para generar y contar los pulsos de entrada.

```
Principal
    movlw    0x08
    movwf    cuenta
    movlw    0x04
    movwf    contador
    btfsc    PORTA,0
    goto     Principal
;hay un tiempo aleatorio que debo esperar para obtener el uno del dato star.
```

Como el programa solo funciona si recibió una señal debemos confirmar que no realmente se recibió una señal y no solo ruido.

```
EsperoUno
    btfss    PORTA,0
    goto     EsperoUno
```



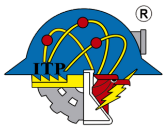
Por eso agregamos líneas de código como las de la imagen para que fuera comprobando si realmente había una señal o si la señal se mantenía para seguir con su funcionamiento.

También implementamos unas rutinas de interrupción para el Timer Oy Timer1 para controlar el tiempo y la frecuencia de las señales generadas.

```
Retardo_200ms      ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'200'    ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_100ms      ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'100'    ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_50ms       ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'50'     ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_20ms       ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'20'     ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_10ms       ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'10'     ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_5ms        ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'5'      ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_2ms        ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'2'      ; Aporta 1 ciclo máquina. Este es el valor de "M".
    goto Retardos_ms ; Aporta 2 ciclos máquina.
Retardo_1ms        ; La llamada "call" aporta 2 ciclos máquina.
    movlw d'1'      ; Aporta 1 ciclo máquina. Este es el valor de "M".
;
```

Para el procesamiento de datos creamos el bucle LOOPDATOS. El código dentro de este bucle lee y procesa datos del puerto A y realiza manipulaciones bit a bit en datoserie, después transfiere esos datos al puerto B. Ocupamos los retardos y comparaciones para asegurar que los datos sean leídos y procesados correctamente antes de pasar a la siguiente parte del programa.

```
LOOPDATOS
    call Retardo_1ms
    movf PORTA,0
    movwf Count1
    call Retardo_1ms
    movf Count1,0
    subwf PORTA,0
    btfss STATUS,2
    goto Fin
    call Retardo_1ms
    movf Count1,0
    subwf PORTA,0
    btfss STATUS,2
    goto Fin
    bsf STATUS,0
    btfss Count1,0
    bcf STATUS,0
    rrf datoserie,1
    call Retardo_2ms
    decfsz contador,1
    goto LOOPDATOS
;continua. El swapf es para obtener el verdadero dato DATO -> DATOSERIE
;se divide en 2 segmentos de 4 bits, el swapf cambia, luego (DSERIE1) (DSERIE2) -> (DSERIE2) (DSERIE1)
    swapf datoserie,1
    movf datoserie,0
    movwf PORTB
;espero un tiempo menor a 40ms
    call Retardo_10ms
    goto Principal
Fin
```

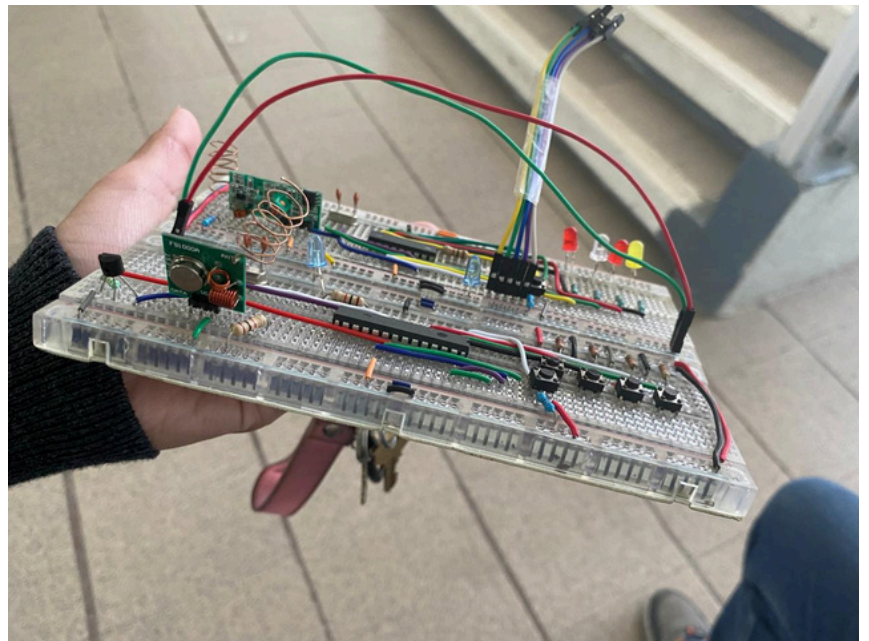
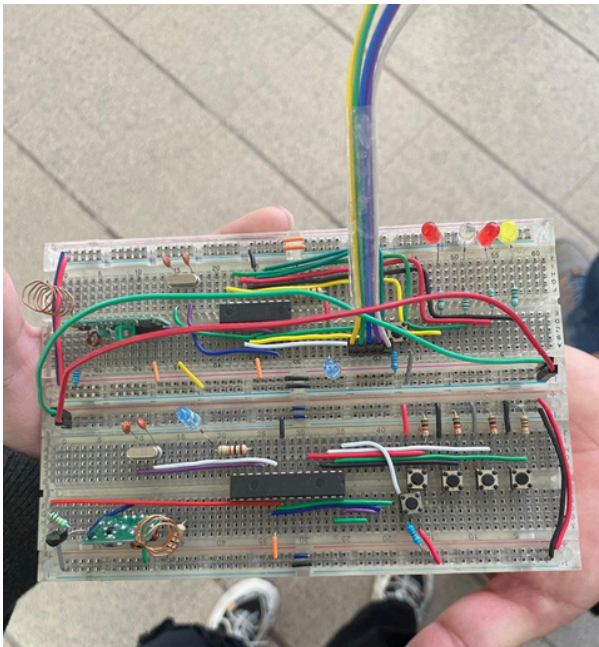



Este código funciona cuando ocurre una interrupción del Timer0. Verifica si el primer bit de PORTA es 0; si lo es, borra PORTB. Luego cambia el valor del primer bit de PORTB, reinicia la interrupción y vuelve al programa principal

```
;Temporizacion por desbordamiento del Timer0
Timer0_interrupcion
    btfss    PORTA,0
    clrf     PORTB
    ;TEST////////////////////////////////////
    movlw   0x80
    xorwf    PORTB,F
    ;////////////////////////////////////|
    bcf      INTCON,T0IF
    retfie
```

Memoria fotográfica:

Fotografías del nuestro proyecto:



Diferentes vistas del proyecto