

Universidad Autonoma de Sinaloa

INGENIERIA EN
TELECOMUNICACIONES, SISTEMAS Y
ELECTRONICA

Asignatura:
Microcontroladores

REPORTE

Asesor:
José Gabriel Zúñiga Tapia

Autores:
Angel Daniel Zanahua Soto
Cristian Alexis Lopez Bernal
José Luis Rubio Manjarrez

Grupo 4-1

JUNIO 2024

Contents

1	Descripcion de proyecto	2
2	Documentacion de componentes	2
2.1	PIC16F628A	2
2.2	PIC18F4550	3
2.3	Modulos RF de 433Mhz	4
2.3.1	FS1000A	4
2.3.2	XY-MK-5V	5
2.3.3	Modulación por desplazamiento de amplitud	5
3	Implementacion	6
3.1	Transmisor	6
3.2	Receptor	7
3.3	Simulacion en Proteus	14
3.4	Implementacion Fisica	15
3.5	Transmisor	15
3.6	Receptor	15
4	Referencias	16

1 Descripción de proyecto

Consiste en dos módulos, transmisor y receptor. El módulo consiste en un PIC16F628A enviando datos seriales a 4800 baudios usando un transmisor RF. El módulo receptor consiste en un PIC18F4550 que se encarga de manejar 8 display de 7 segmentos por medio de multiplexación, lo que representa cada display y sus funciones se hace interpretando los datos que recibe por serial, los cuales se interpretan como valores que cambian o para inicio de contadores

2 Documentación de componentes

2.1 PIC16F628A

El Microcontrolador PIC16F628A cuenta con 8 Bits, 3.5KB, 224 RAM, 4MHz, 18 Pines. Compatible con el PIC16F628, PIC16C62XA, PIC16C5X y PIC12CXXX. Tecnología en Flash/EEPROM CMOS.

Cuenta con modo de ahorro de energía en modo sueño, resistencias programable pul-ups del PORTB, multiplexado del pin reset/Entrada-pin, temporizador Watchdog

- Familia : PIC16F
- Voltaje de operación: 3.5V a 5.5V
- Capacidad de memoria de programa: 3.5KB (2K a 14 bits)
- RAM: 224 bytes
- SRAM: 256 bytes
- EEPROM: 128 bytes a 8 bits
- I/O disponibles: 16 pines
- ADC: No
- Comparadores : 2
- Timer: 2 de 8 bit y 1 de 16 bit

- Frecuencia Máxima : 20 MHz
- Oscilador Interno: 4 MHz
- Comunicación: USART/UART

2.2 PIC18F4550

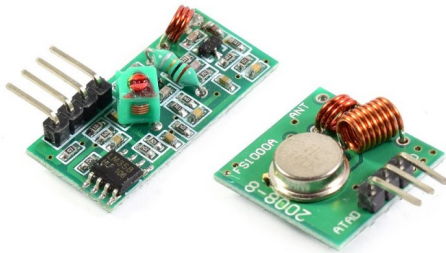
El Microcontrolador PIC18F4550 es un circuito integrado programable capaz de poder realizar y controlar tareas las veces que desees gracias a la memoria flash de alta resistencia. El MCU cuenta con 8 Bits, 48 MHz, 32 KB, 2 KB, 40 Pines y pertenece a la familia de microcontroladores PIC18.

Especificaciones

- Familia : PIC18
- Modelo: PIC18F4550-I/P
- Voltaje de operación: 4.2V a 5.5V
- Comunicación : UART, A/E/USART, SPI, I²C, MSSP (SPI/I²C)
- Interfaz : USB 2.0 de alta velocidad (12 Mbit/s)
- Pines: 40
- ADC de 10 bits: 13 canales
- I/O disponibles: 35 pines
- Comparadores Analógicos: 2
- EEPROM: 256 Bytes
- Memoria Flash: 32Kb
- Memoria de datos SRAM: 2048 Bytes
- Máxima frecuencia de trabajo: 48 MHz

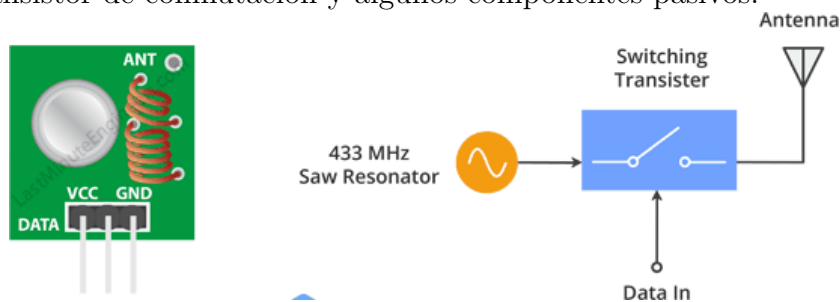
2.3 Módulos RF de 433Mhz

Los módulos RF de 433Mhz de bajo costo y fácil uso. Vienen en pareja, emisor (FS1000A) y receptor (XY-MK-5V), el tipo de comunicación es simplex, es decir en un solo canal y unidireccional, son de baja velocidad de transmisión pero para aplicaciones básicas son muy útiles.



2.3.1 FS1000A

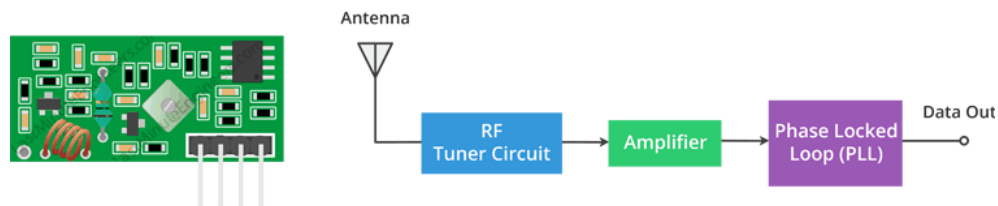
Este módulo sirve como transmisor. En el núcleo del módulo hay un resonador SAW sintonizado para funcionar a 433 MHz. Aparte de eso, tiene un transistor de conmutación y algunos componentes pasivos.



Cuando la entrada de DATOS es alta, el oscilador genera una onda portadora de salida de RF constante a 433 MHz, y cuando la entrada de DATOS es baja, el oscilador deja de funcionar; dando como resultado una onda de amplitud modulada. Esta técnica se conoce como modulación por desplazamiento de amplitud.

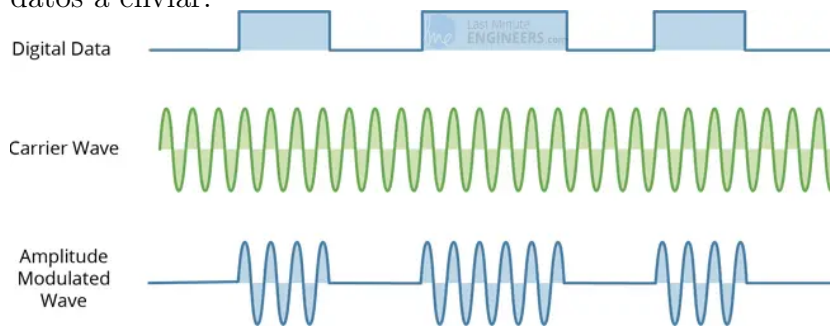
2.3.2 XY-MK-5V

Este módulo es el receptor. Consiste en un circuito sintonizado por RF y un par de amplificadores operacionales (OP Amps) que amplifican la onda portadora recibida. Luego, la señal amplificada se envía a un PLL, que permite que el decodificador se "bloquee" en un flujo de bits digitales, lo que da como resultado una salida decodificada mejorada y una inmunidad al ruido.



2.3.3 Modulación por desplazamiento de amplitud

La modulación por desplazamiento de amplitud, en inglés Amplitude-shift keying (ASK), es una forma de modulación en la cual se representan los datos digitales como variaciones de amplitud de la onda portadora en función de los datos a enviar.



La amplitud de una señal portadora analógica varía conforme a la corriente de bit (modulando la señal), manteniendo la frecuencia y la fase constante. El nivel de amplitud puede ser usado para representar los valores binarios 0 y 1. Podemos pensar en la señal portadora como un interruptor ON/OFF. En la señal modulada, el valor lógico 0 es representado por la ausencia de una portadora, así que da ON/OFF la operación de pulsación y de ahí el nombre dado.

3 Implementacion

3.1 Transmisor

```
1 #include <16F628A.h>
2 #device ADC=16
3 #FUSES NOWDT, NOMCLR, NOBROWNOUT, NOLVP
4 #use delay(internal=4MHz)
5 #define boton1    PIN_A0
6 #define boton2    PIN_A1
7 #define boton3    PIN_A2
8 #define boton4    PIN_A3
9 #define boton5    PIN_A4
10 #define boton6    PIN_A5
11 #define boton7    PIN_A6
12 // #define boton8    PIN_A7
13
14 #use rs232(baud=4800,parity=N,xmit=PIN_B2,rcv=PIN_B1,bits=8,
15           stream=PORT1,errors)
16
17 void main() {
18     while(TRUE){
19         if(input(boton1)){
20             putc('a');
21         }
22         if(input(boton2)){
23             putc('b');
24         }
25         if(input(boton3)){
26             putc('c');
27         }
28         if(input(boton4)){
29             putc('d');
30         }
31         if(input(boton5)){
32             putc('e');
33         }
34         if(input(boton6)){
35             putc('f');
36         }
37         if(input(boton7)){
38             putc('z');
39         }
40     }
```

3.2 Receptor

```
1 #include <18F4550.h>
2 #device ADC=16
3
4 #FUSES NOWDT //No Watch Dog Timer
5 #FUSES WDT128 //Watch Dog Timer uses 1:128
6 #FUSES POSTSCALE Postscale
7 #FUSES NOFCMEN //Fail-safe clock monitor
8 #FUSES disabled
9 #FUSES NOBROWNOUT //No brownout reset
10 #FUSES NOMCLR //Master Clear pin used for I/O
11 #FUSES NOLVP //No low voltage prgming, B3(PIC16) or B5(PIC18) used for I/O
12 #FUSES NOXINST //Extended set extension and Indexed Addressing mode disabled (Legacy mode)
13
14 #use delay(crystal =20Mhz)
15 #use FIXED_IO( D_outputs=PIN_D7,PIN_D6,PIN_D5,PIN_D4,PIN_D3,PIN_D2,PIN_D1,PIN_D0 )
16 #use FIXED_IO( A_outputs=PIN_A1)
17
18 #define dig1a PIN_D0 //local
19 #define dig2a PIN_D1 // sdec --
20 #define dig1b PIN_D2 //vis
21 #define dig2b PIN_D3 //per
22 #define dig1c PIN_D4 // suni --
23 #define dig2c PIN_D5
24 #define dig3c PIN_D6
25 #define dig4c PIN_D7
26
27 #define led PIN_A1
28
29 #use rs232(baud=4800,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8,stream=PORT1,errors) //xmit=PIN_C1,rcv=PIN_C0
30
31 int timer = 0;
32 int reloj = 0;
33 int min = 0;
34 int per = 0;
35 int shot = 0;
36 int local = 0;
37 int vis = 0;
38 int unidades = 0;
```

```

37 int decenas = 0;
38 int sreloj = 0;
39 int suni = 0;
40 int sdec = 0;
41 int reset = 0;
42
43 #int_timer0
44 void timer0_isr() {
45
46     if(timer>0){
47         reloj++;
48         per = min + 1;
49         if(shot>0 && shot<2){
50             sreloj++;
51         }
52     }
53     set_timer0(26473);
54 }
55
56 int display[10] = {0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0
    x7F,0x6F};
57 int display2[10] = {0xC0,0xF9,0xA4,0xB0,0x99,0x92,0x82,0xF8,0
    x80,0x90};
58
59 char dato_rx;
60
61 void main() {
62     setup_timer_0(RTCC_INTERNAL|RTCC_DIV_128);           //1600 ms
        overflow
63
64     enable_interrupts(INT_TIMER0);
65     enable_interrupts(GLOBAL);
66     set_timer0(26473);
67
68     set_tris_b(0x00);
69
70     while(TRUE){
71         if(kbhit()){
72             dato_rx = getch(); //if(kbhit())>0) Verifica si ha
                recibido un dato por el puerto serial
73             switch(dato_rx)
74             {
75                 case 'a':
76                     local++;
77                     putc('a');

```

```

78         delay_ms(16);
79         break;
80
81         case 'b':
82             local--;
83             delay_ms(16);
84             break;
85
86         case 'c':
87             vis++;
88             delay_ms(16);
89             break;
90
91         case 'd':
92             vis--;
93             delay_ms(16);
94             break;
95
96         case 'e':
97             timer++;
98             break;
99
100        case 'f':
101            reset++;
102            timer = 0;
103            local = 0;
104            vis = 0;
105            per = 0;
106            min = 0;
107            reloj= 0;
108            unidades = 0;
109            decenas = 0;
110            shot=0;
111            sreloj =0;
112            break;
113
114        case 'z':
115            shot++;
116            delay_ms(16);
117            break;
118    }
119 }
120 unidades=reloj%10;
121 decenas=(reloj/10)%10;
122

```

```

123     output_low(dig1a);
124     output_high(dig1b);
125     output_high(dig2b);
126     output_low(dig2c);
127     output_low(dig3c);
128     output_low(dig4c);
129     output_high(dig2a);
130     output_high(dig1c);
131     output_b(display[local]);
132     delay_ms(3);
133
134     output_high(dig1a);
135     output_low(dig1b);
136     output_high(dig2b);
137     output_low(dig2c);
138     output_low(dig3c);
139     output_low(dig4c);
140     output_high(dig2a);
141     output_high(dig1c);
142     output_b(display[vis]);
143     delay_ms(3);
144
145     output_high(dig1a);
146     output_high(dig1b);
147     output_low(dig2b);
148     output_low(dig2c);
149     output_low(dig3c);
150     output_low(dig4c);
151     output_high(dig2a);
152     output_high(dig1c);
153     output_b(display[per]);
154     delay_ms(3);
155
156     output_high(dig1a);
157     output_high(dig1b);
158     output_high(dig2b);
159     output_high(dig2c);
160     output_low(dig3c);
161     output_low(dig4c);
162     output_high(dig2a);
163     output_high(dig1c);
164     output_b(display2[min]);
165     delay_ms(3);
166
167     output_high(dig1a);

```

```

168     output_high(dig1b);
169     output_high(dig2b);
170     output_low(dig2c);
171     output_high(dig3c);
172     output_low(dig4c);
173     output_high(dig2a);
174     output_high(dig1c);
175     output_b(display2[decenas]);
176     delay_ms(3);
177
178     output_high(dig1a);
179     output_high(dig1b);
180     output_high(dig2b);
181     output_low(dig2c);
182     output_low(dig3c);
183     output_high(dig4c);
184     output_high(dig2a);
185     output_high(dig1c);
186     output_b(display2[unidades]);
187     delay_ms(3);
188
189     suni=sreloj%10;
190     sdec=(sreloj/10)%10;
191
192     output_high(dig1a);
193     output_high(dig1b);
194     output_high(dig2b);
195     output_low(dig2c);
196     output_low(dig3c);
197     output_low(dig4c);
198     output_low(dig2a);
199     output_high(dig1c);
200     output_b(display[suni]);
201     delay_ms(3);
202
203     output_high(dig1a);
204     output_high(dig1b);
205     output_high(dig2b);
206     output_low(dig2c);
207     output_low(dig3c);
208     output_low(dig4c);
209     output_high(dig2a);
210     output_low(dig1c);
211     output_b(display[sdec]);
212     delay_ms(3);

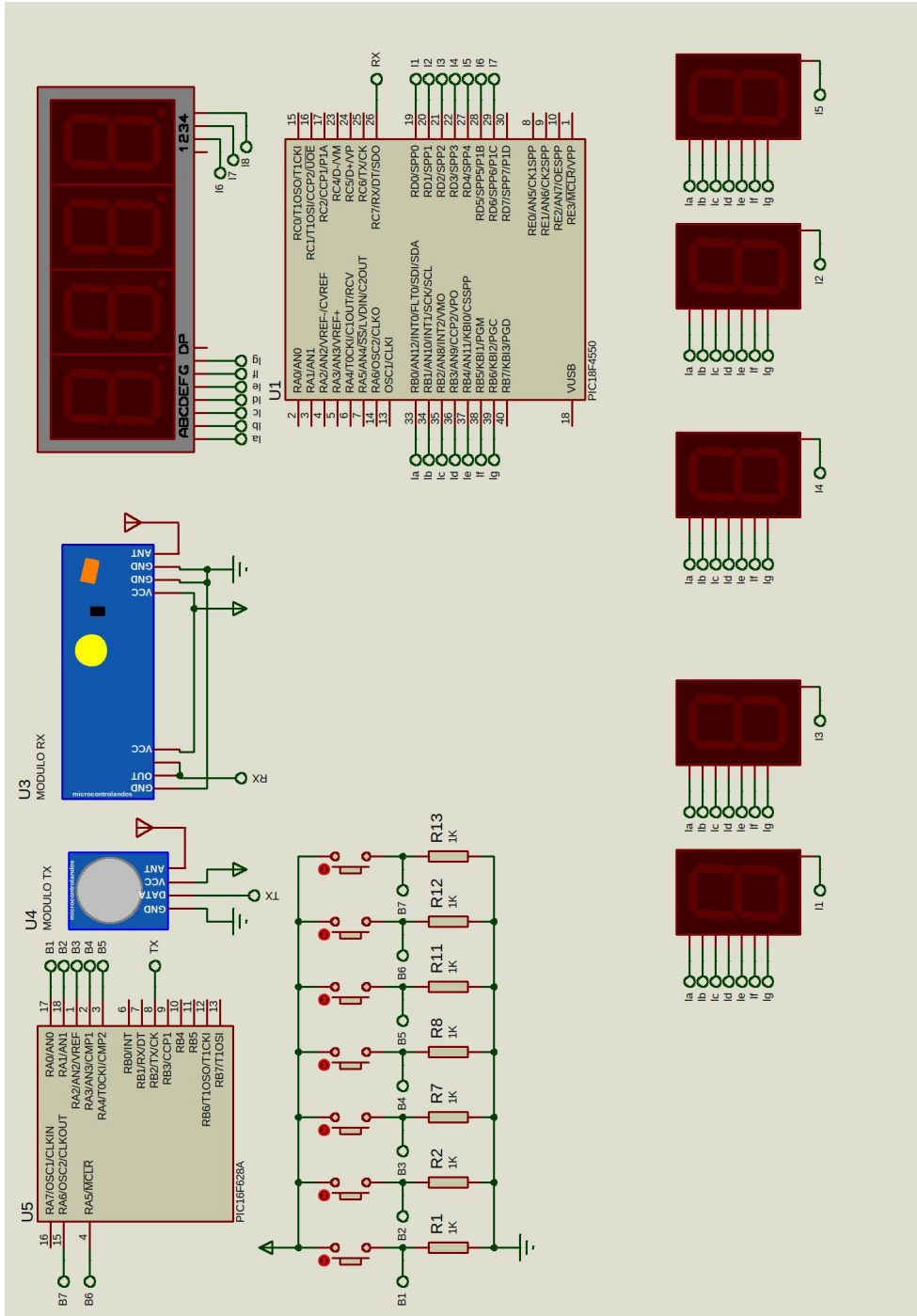
```

```

213
214     if(local > 9){
215         local = 0;
216     }
217     if(vis > 9){
218         vis = 0;
219     }
220     if(per > 5){
221         per = 0;
222         min = 0;
223         timer = 0;
224     }
225     if(sreloj > 24){
226         sreloj=0;
227         sreloj=88;
228     }
229     if(shot>1){
230         shot=0;
231         sreloj =0;
232         suni = 0;
233         sdec =0;
234     }
235     if(reloj > 59){
236         reloj=0;
237         min++;
238         per = min + 1;
239     }
240     if(reset>0){
241         reset = 0;
242         timer = 0;
243         local = 0;
244         vis = 0;
245         per = 0;
246         min = 0;
247         reloj= 0;
248         unidades = 0;
249         decenas = 0;
250         shot = 0;
251         sreloj =0;
252         suni = 0;
253         sdec =0;
254     }
255 }
256 }

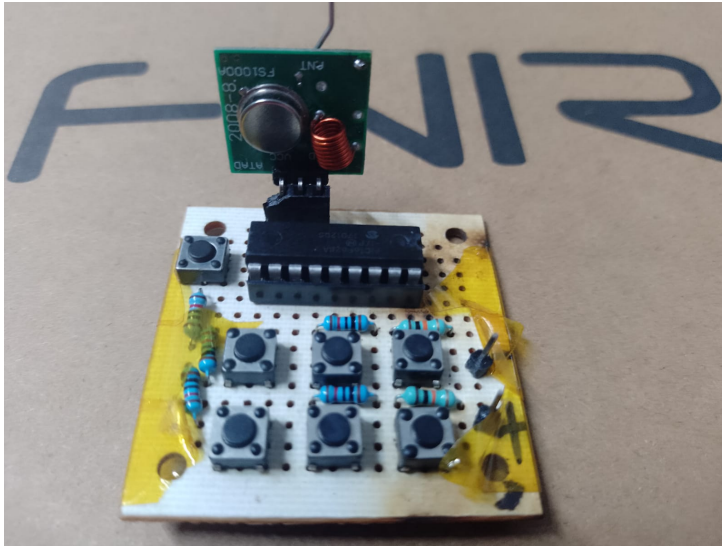
```

3.3 Simulacion en Proteus

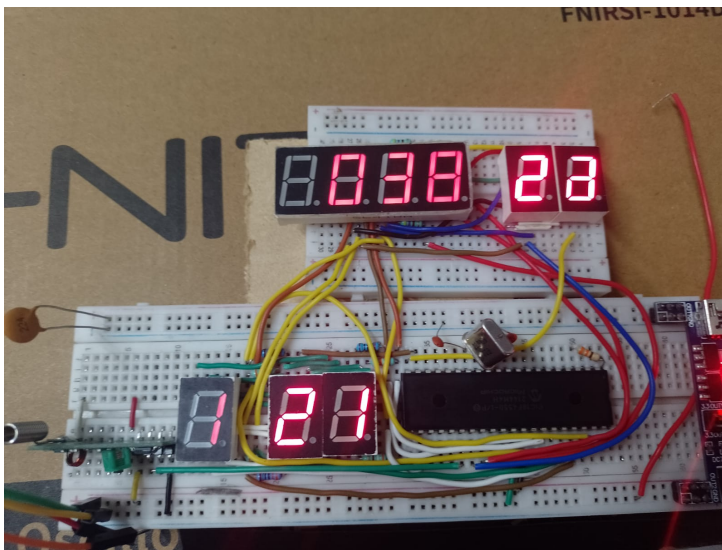


3.4 Implementacion Fisica

3.5 Transmisor



3.6 Receptor



4 Referencias

<https://lastminuteengineers.com/433mhz-rf-wireless-arduino-tutorial/>
https://www.youtube.com/watch?v=9-JsrJU3cwU&list=PLONPO-iVba9nbY_KTCHt9GGj9dGSj1qYo&index=31&pp=iAQB
https://www.youtube.com/watch?v=MEIuWy9-nn4&list=PLONPO-iVba9nbY_KTCHt9GGj9dGSj1qYo&index=32&t=247s&pp=iAQB