

Control de un Sintonizador con PIC

Control de un Sintonizador con PIC



CAMILO MONETTA

Ingeniero Tecnológico en Electrónica

cmprod@adinet.com.uy

Salto ,Republica Oriental del Uruguay

En este artículo describiremos la manera en que se comunica el microprocesador utilizado en TV con el sintonizador de canales. Centramos el artículo sobre los sintonizadores de 3 hilos.

Los microprocesadores hoy en día se colocaron como centro de atención, su utilización se los pueden encontrar en las más variadas aplicaciones. Desplazando un sin fin de integrados lógicos utilizados hace algunos años atrás. Como no podría ser de otra forma los televisores no se escaparon de su inserción dotándolos así de comodidades y funciones para el usuario final. Trataremos en este artículo de describir el funcionamiento del microprocesador para la sintonía de los canales, estos le comunican al sintonizador la banda que deben seleccionar así como la división que debe hacer el PLL interno del sintonizador, logrando así que el oscilador local desplace su frecuencia para ubicarse en la frecuencia adecuada en la cual al mezclarse la señal de entrada con la del oscilador local den como resultado a la salida del mezclador cuatro frecuencias F_1, F_2, F_3, F_4 las cuales son:

F_1 = frecuencia de entrada

F_2 = frecuencia generada por el oscilador local

$F_3 = F_1 + F_2$ (suma de la frecuencia del oscilador y la frecuencia de entrada)

$F_4 = F_1 - F_2$ (resta de la frecuencia del oscilador y la frecuencia de entrada)

Luego a la salida del mezclador se le coloca un filtro pasa bajos y se obtiene una única frecuencia F_4 que se convierte en la llamada $F.I$ frecuencia intermedia. Para que la idea sea más notoria ver la figura 1 representación del proceso sufrido por la señal de entrada

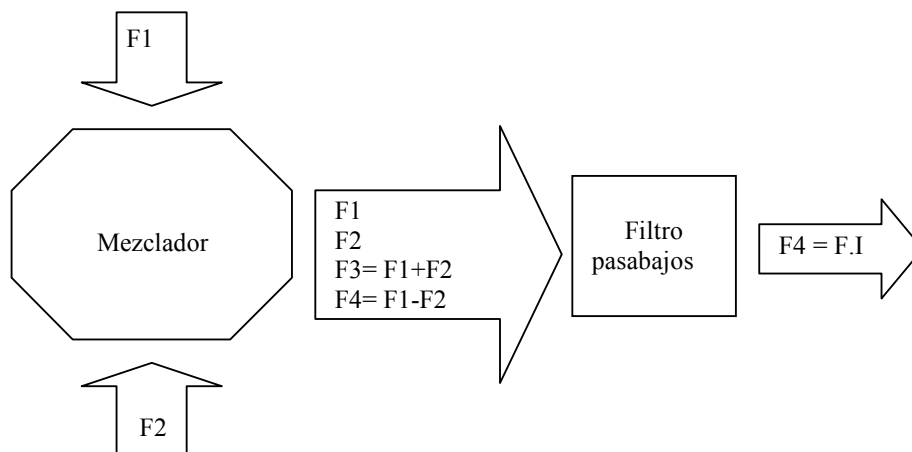


Figura 1

Construcción interna del sintonizador TUGH9 – A04M

Para llevar a cabo nuestro proyecto hemos de hacer uso de un sintonizador de tv, un convertor de frecuencia, un display donde indicaremos el canal seleccionado, dos pulsadores con los cuales podremos recorrer los canales y un control remoto Philips, y la figura central de nuestro circuito es el microcontrolador de microchip el 16C84.

El sintonizador que utilizaremos en este artículo es el TUGH9-A04M haremos una descripción de la circuitería interna del mismo para poder comprender su funcionamiento. Este sintonizador está compuesto de un PLL fabricado por Motorola es el MC44817 de montaje superficial (ver figura 2) el mismo requiere de un cristal externo de 4Mhz que se conecta en el Pin 3 que luego se divide la frecuencia obteniendo una frecuencia de referencia para el comparador de fase. La frecuencia de entrada proporcionada por el oscilador local entra por el Pin 8 es amplificada y pasa por un divisor por 8, luego esta ingresa a un divisor programable de 15 bit, su salida proporciona la frecuencia que se debe comparar en fase con la frecuencia de referencia para obtener el error y así hacer la corrección para el VCO. El mismo integrado dispone de un selector de bandas ubicados en los pines 10, 11, 12, 13. La selección de banda como del divisor programable es controlado por el bus de 3 hilos en los cuales se hacen presente en los siguientes pines:

Pin 1 _ **Dato** ingresa el dato al sintonizador con la información de la banda y de la división

Pin 2 _ **Reloj** este actúa como sincronizador de los datos de no estar presente la señal que ingresa por el pin 1 no debe ser validada

Pin 16_ **Habilitación** señal que como su nombre lo indica habilita o lo deshabilita al sintonizador para recibir datos

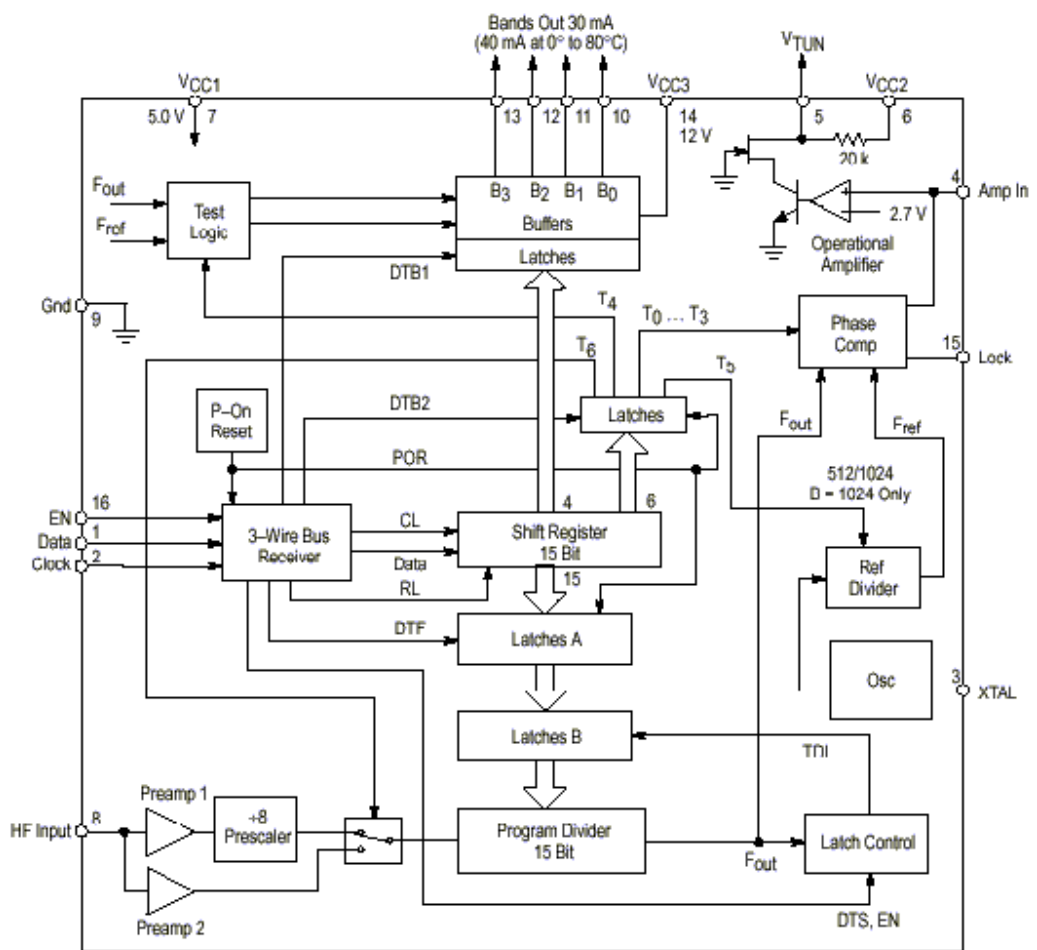


Figura 2

En la Figura 3 apreciamos la manera en que debemos establecer la comunicación con el integrado, los primeros cuatro bit enviados corresponde a la selección de banda, y los restantes 15 bit se utilizan para programar el divisor, claro está que a los datos lo acompañan en cada ciclo el reloj, y para validar estos datos debe estar enabled en estado alto de lo contrario se descartaran los datos enviados

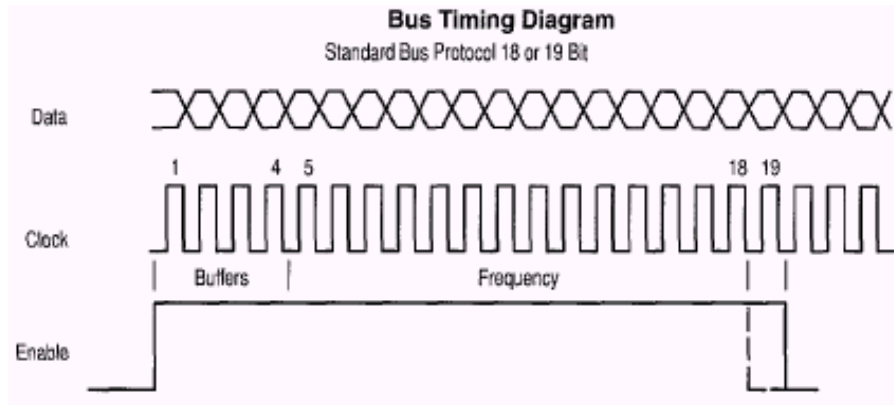


Figura 3

Encapsulado del MC44817

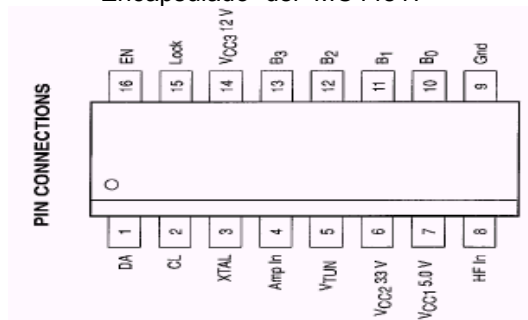


Figura 4

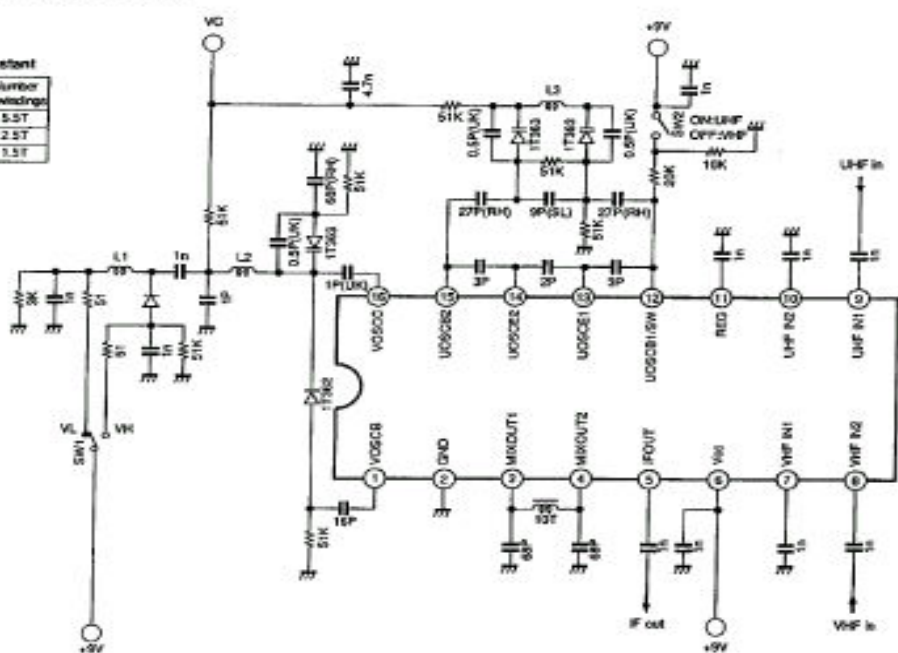
MC44817

El VCO que utiliza este sintonizador es fabricado Sony es el CXA1665M que esta provisto de oscilador ,mezclador para VHF/CATV/UHF y también de un amplificador de F.I En la figura 5 veremos una aplicación típica de este integrado con los componentes externos a el

Electrical Characteristics Test Circuit

Table 1. Inductance constant

Wire diameter	Coil diameter	Number of windings
L1	0.5	3.2
L2	0.5	3.2
L3	0.5	3.2



Típica aplicación del CXA 1665

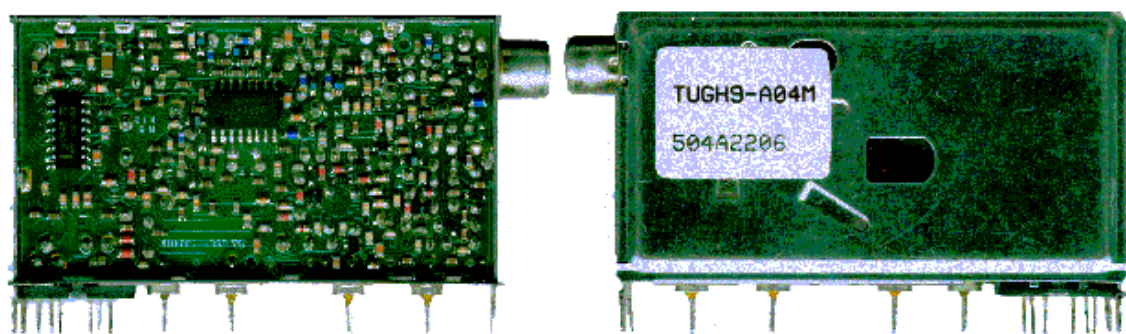
Figura 5

Para hacer el cambio de banda debemos tener presente el valor de la frecuencia de entrada (F), según sea la misma, cambia el número a enviarle al MC44817 para que actualice la banda, este cambio se presenta en los pines 10,11,12,13 informándole al CXA 1665 que debe de efectuar el cambio.

Si $F < 165$ MHz el valor de la banda = 1 (B3=0 B2=0 B1=0 B0=1) 0 0 0 1

Si $165\text{MHz} < F < 390$ Mhz el valor de banda = 2 (B3=0 B2=0 B1=1 B0=0) 0 0 1 0

Si $F > 390$ MHz el valor de la banda = 8 (B3=1 B2=0 B1=0 B0=0) 1 0 0 0



Como hemos visto hasta ahora la comunicación que debemos entablar con el Mc44817 es en forma serial de 3 hilos, ahora veremos la forma de llevar esto a cabo. Para esta tarea necesitaremos de la ayuda del microcontrolador 16c84, podría ser otro pero en este momento es muy popular y de un costo muy razonable, además de encontrar información abundante sobre el mismo.

Usaremos dos pulsadores conectados a el micro para informarle que debe de hacer un cambio de canal ,implementaremos uno para aumentar el canal y el segundo de ellos para disminuir el canal y un pin dedicado a la recepción del control remoto.La distribución de pines que usaremos es la siguiente :

PUERTO A

RA2 = Pulsador + (IN)
 RA3 = Datos del control remoto (IN)
 RA4 = Pulsador – (IN)

PUERTO B

RB0 = Data MC14499
 RB1 = Enable MC14499
 RB2 = Clock MC14499
 RB3 = Enable del Sintonizador
 RB4 = Clock del Sintonizador
 RB5 = Data del Sintonizador

Por lo tanto programaremos el puerto A como entrada y el puerto B como salida . Tomaremos el proyecto y lo repartiremos en tres tareas a realizar .La primera de ellas es el manejo de 4 display de siete segmentos ,otra es el micro controlando el sintonizador y la tercera y ultima es el conversor de frecuencia con salida en el canal 4

Manejando Display de 7 Segmentos con el MC14499

Necesitaremos 4 display de 7 segmentos de cátodo común ,en nuestro caso uno de ellos le haremos las conexiones necesarias para que nos muestre una C de canal en forma permanente, los tres restantes nos mostrara el numero del canal sintonizado que estará comprendido desde el canal 2 al 125 de la banda de cable CATV.

Para el control y manejo del display haremos uso de un integrado diseñado para este fin como lo es el integrado fabricado por Motorola el MC14499 el esta provisto con comunicación con microprocesadores y microcontroladores.

El integrado acepta 20 bit de entrada ,de los cuales 16 son para los 4 display y los restantes 4 bit para el manejo del punto decimal.El orden de entrada de los datos es la siguiente ,debe estar ENABLE\ en nivel bajo luego se envía el dato serial acompañado de una señal de reloj ,estos ingresan a un registro de desplazamiento que transforman los datos seriales en paralelos y son almacenados cuando ENABLED\ vuelve al estado alto.Podemos fijar la frecuencia de refresco de los display mediante el condensador conectado al pin 6 OSC ,se recomienda operar en el rango de los 200 a 800 Hz.

Este integrado nos facilita mucho a la hora de pensar en la construcción del programa ya que de no utilizarlo tendríamos que implementar en el software un continuo envió de información tanto del numero a mostrar como del digito a habilitar consumiendo un gran esfuerzo tanto de nuestra parte como de parte del microcontrolador ya que la mayor parte de su tiempo lo utilizaría en ello ,en cambio de esta forma solo tendremos que implementar la rutina de manejo del MC14499 ,enviándole los 20 bit y luego queda todo a cargo del mismo del manejo de los display tanto de indicar el numero y de el refresco liberándolo al mico para atender otras tareas.

Nº BIT	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
	LSE		MSE		DIGITO 3				DIGITO 2				DIGITO 1				DIV	DH	DH	DH	DI
																	PUNTO DECIMAL				

Secuencia de ingreso de datos al MC 14499

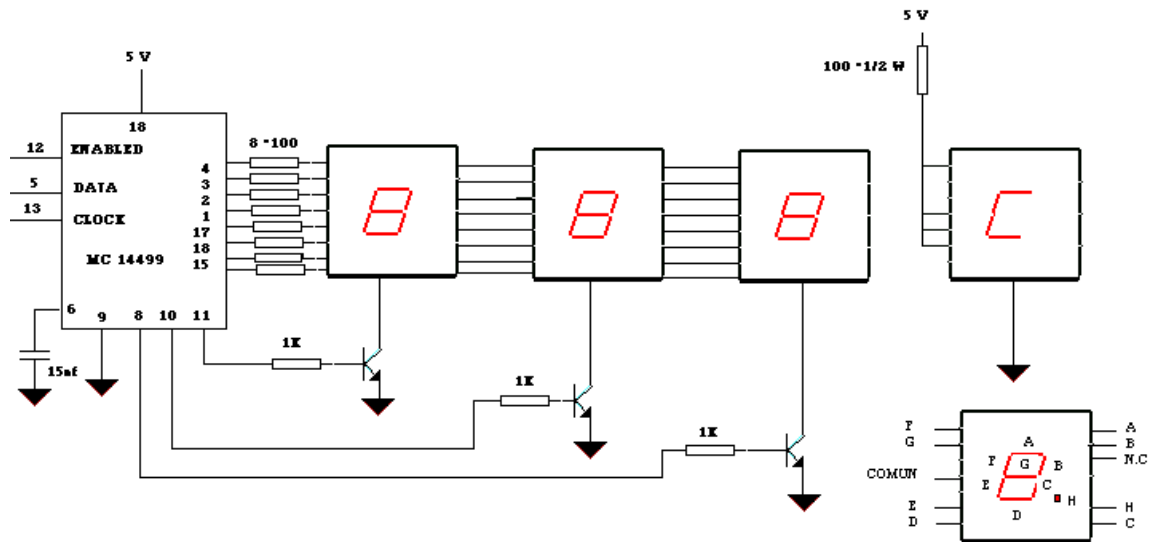


Figura 6

Como podemos observar en la figura 6 es de gran simpleza su circuiteria pero de grandes prestaciones para nuestros propósitos ,ya que con solo 3 pines del microcontrolador obtenemos el control total de 4 display.

Lista de Materiales Para el Display

- 8_ Resistencias de 100 ohmios $\frac{1}{4}$ W
- 3_ Resistencias de 1K $\frac{1}{4}$ W
- 1_ Resistencia de 220 ohmios $\frac{1}{2}$ W
- 1_ Condensador cerámico de 15 nf
- 4_ Display de 7 segmentos cátodo común
- 3_ Transistores BC 547
- 1_ Circuito Integrado MC 14499

MICROCONTROLADOR (16C84)

En la figura 7 podemos observar el circuito que debemos construir. Como podemos ver tenemos la fuente de alimentación que se compone por un transformador de tensión de línea y nos entrega 2 tensiones 30+30 VAC y 12+12 VC que luego de ser rectificadas y filtradas pasan a sus correspondientes reguladores de tensión, obteniendo 33V, +9V y +5V.

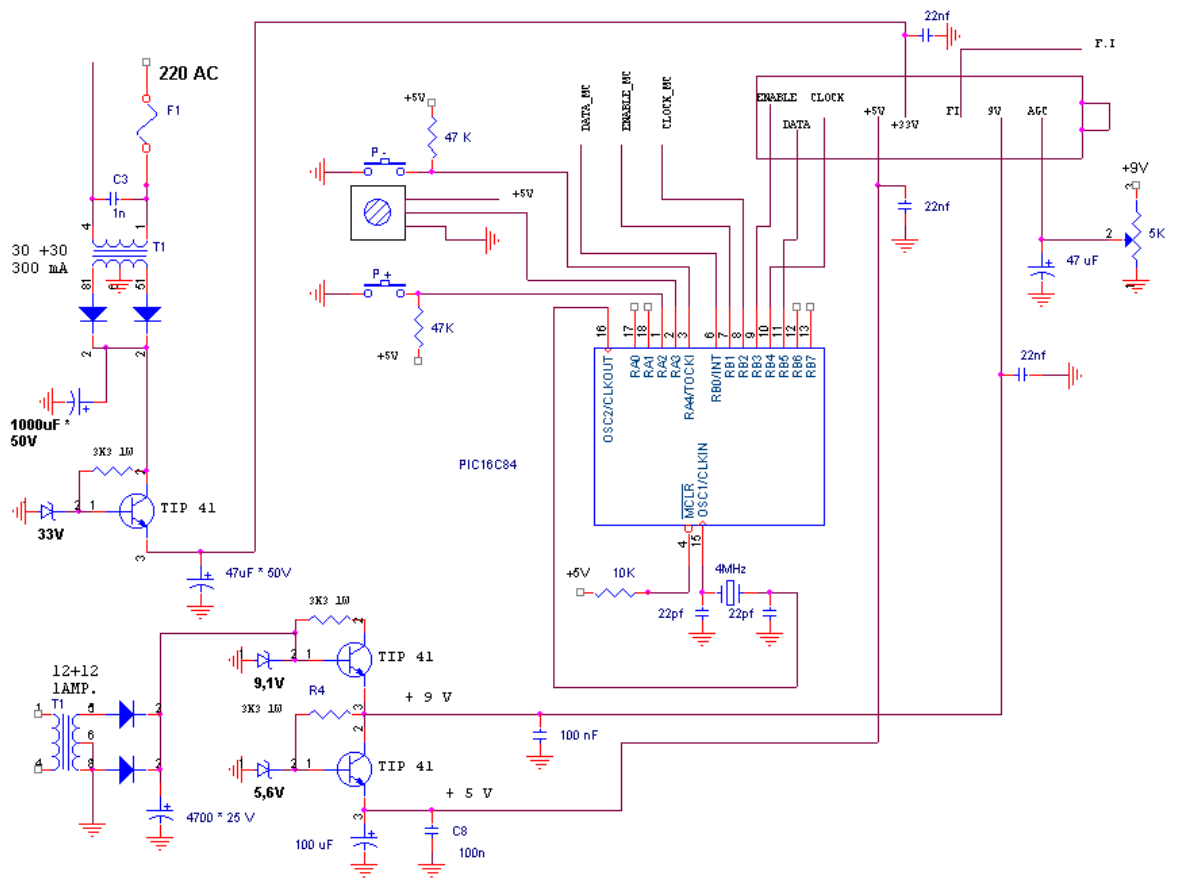
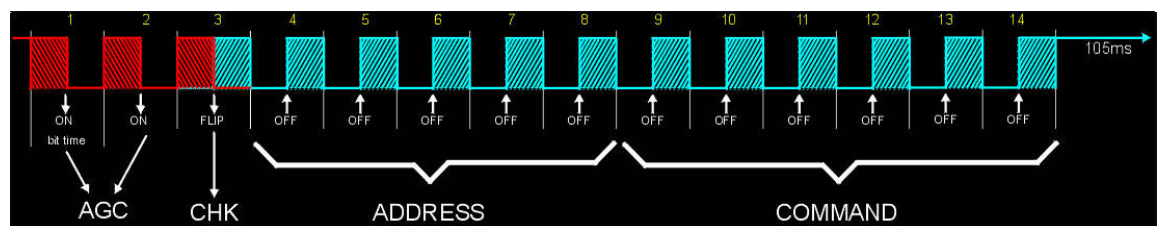


Figura 7

ANEXANDO UN CONTROL REMOTO

Porque no implementarle la posibilidad de comandar nuestro sintonizador de canales a distancia ,para esto pensé utilizar un control remoto universal de la marca Philips ya que se lo encuentra a un precio bastante razonable y además se puede conseguir información del código RC5 ,formato con el cual se envía la información del control remoto hacia la TV ,que en nuestro caso será el sintonizador

El formato del RC5 consta de 14 bit y esta compuesto de la siguiente manera:



Los dos primeros bit se les llama calibración de AGC,el siguiente bit es utilizado para saber si se presiono nuevamente una tecla,los siguientes 5 bit corresponden a que aparato queremos manejar es decir para TV el código a enviar es cero,en nuestro caso lo modificaremos la conexión de SAA3010 y lo pasaremos a siete que corresponde a Experimental,con lo cual si tenemos un Tv Philips en nuestra casa no interferirá en nuestros sintonizador, la Figura 9 podemos ver la tabla con los distintos Sistemas que puede controlar ,los siguientes 6 bit corresponde al comando a ejecutar por el aparato receptor ,ver figura 10

Vídeo		Audio		Otro	
0	TV1	17	Sintonizador	27	
1	TV2	18	Registrador 1	28	
2	Teletexto	19	Preamplificación	29	Iluminación
3	Vídeo	20	Lector de cd	30	Iluminación
4	LV1	21	Phono	31	Teléfono
5	VCR1	22	Sentó A		
6	VCR2	23	Registrador 2		
7	Experimental	24			
8	SAT1	25			
9	Cámara	26	CDR		
10	fotográfica				
11	SAT2				
12					
13	CDV				
14	Camcorder				
15					
16	Preamplificación				

Figura 9

Descripción de los comandos Utilizados

COMANDO (en decimal)	DESCRIPCIÓN de la FUNCIÓN
0-9	NUMEROS del 0 - 9
38	SLEEP (1- -)
32	CANAL +
33	CANAL -

Figura 10

Como el control remoto universal Philips viene para trabajar en TV debemos reformarlo para que envíe el código de experimental (7) ,para ello debemos fijarnos en la hoja de datos del integrado que utilizan estos controles ,el mismo es el SAA3010,allí encontramos toda la información para realizar dicho cambio ,el pin 3 que corresponde a Z0 viene conectado al pin 17 (DR0) así se lo utiliza en TV dirección=0 ,ahora si queremos que envíe en la dirección un 7 debemos abrir la conexión pin3 con pin 17 y conectar el pin 3 (Z0) a el pin 9 (DR7)

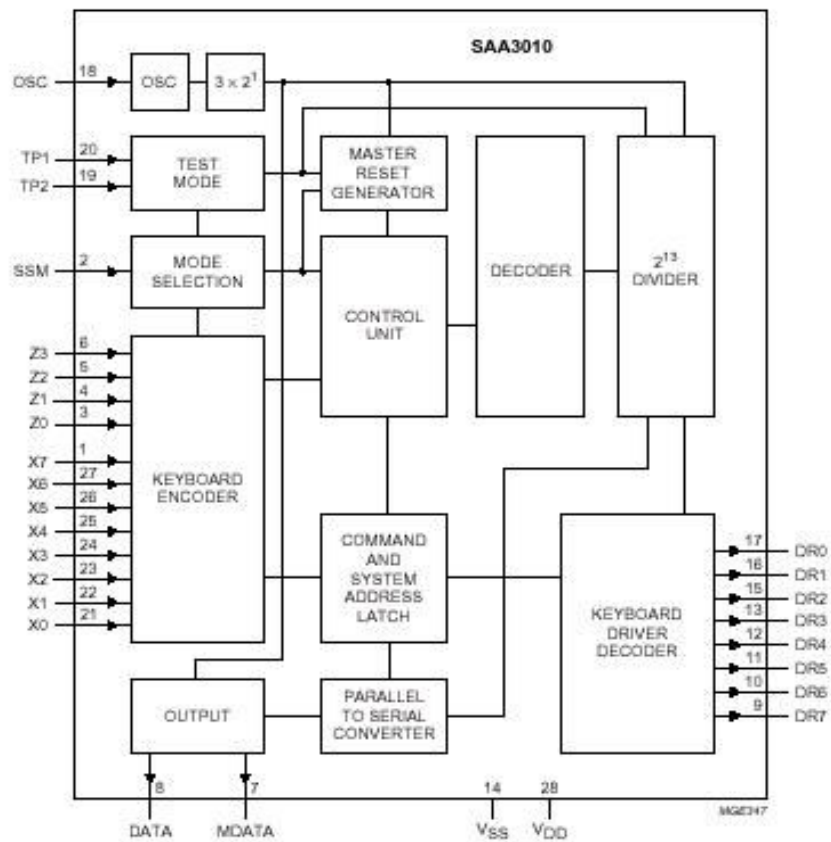
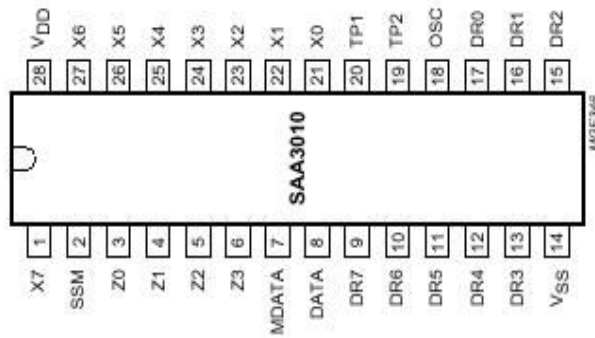
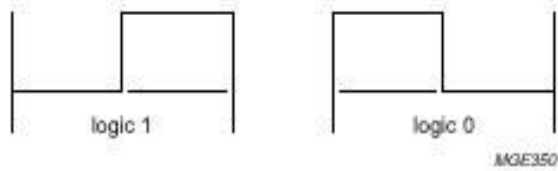


Diagrama del SAA3010



Encapsulado del SAA3010



Where:

$$1 \text{ bit-time} = 3.2^a \times T_{\text{OCC}} = 1.778 \text{ ms (typ.)}$$

Tiempo de duración de un bit

Tabla con los distintos sistemas

System matrix (Z-DR)

SYST. NO.	Z-LINES								DR-LINES								SYSTEM BITS				
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	4	3	2	1	0
0	•								•								0	0	0	0	0
1	•									•							0	0	0	0	1
2	•										•						0	0	0	1	0
3	•											•					0	0	0	1	1
4	•												•				0	0	1	0	0
5	•													•			0	0	1	0	1
6	•														•		0	0	1	1	0
7	•															•	0	0	1	1	1

En nuestro programa dedicado a la recepción del control remoto no tendremos en cuenta los primeros 3 bit ,luego recibiremos la dirección y el comando ,comparamos si la dirección es igual a siete ,si no lo es abortamos los datos recibidos ,en caso contrario ejecutamos el comando enviado.

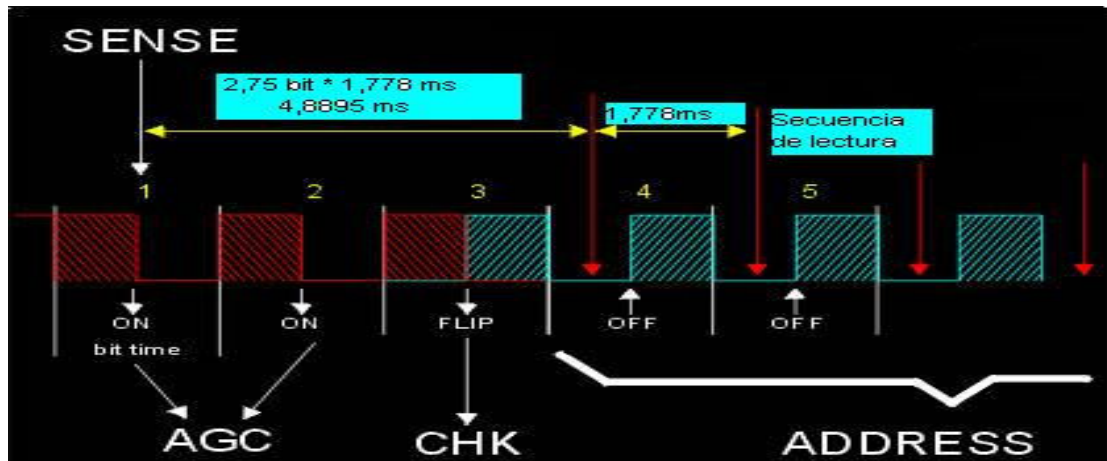


Figura 11

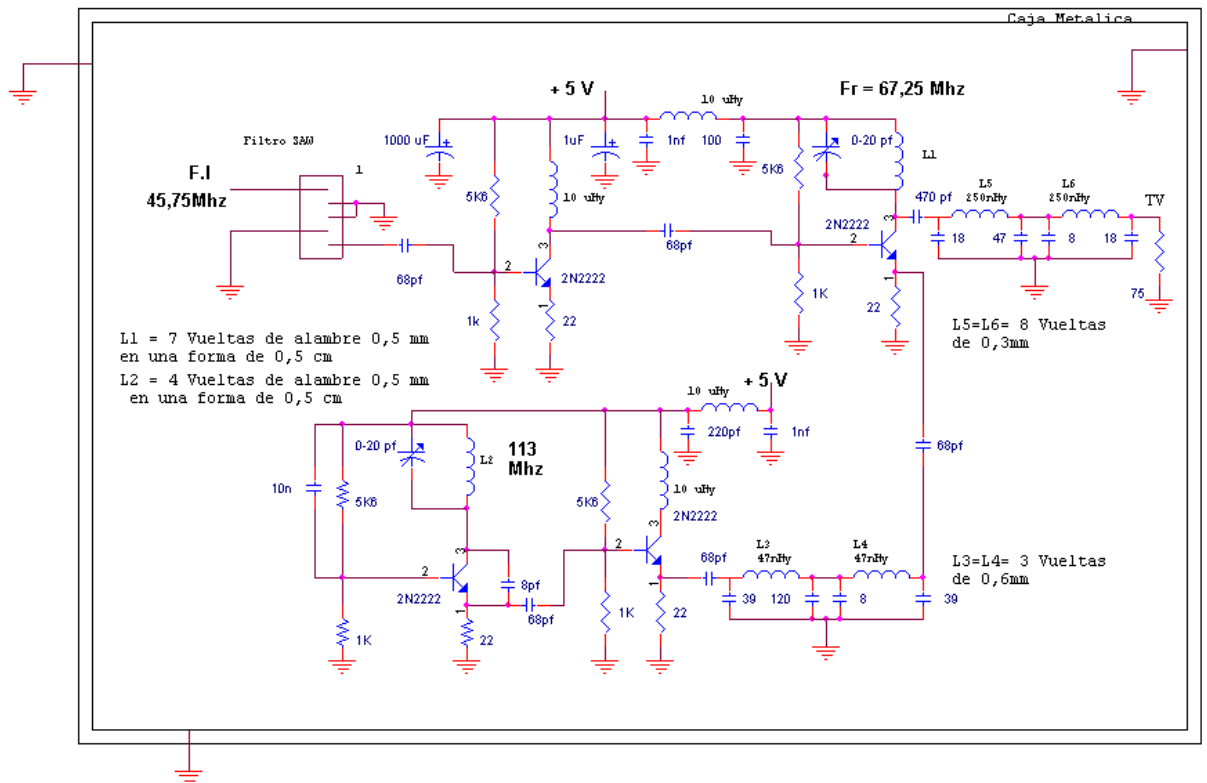
Luego de esperar el tiempo que corresponde a 2,75 bit ($2,75 * 1,778\text{mseg} = 4,8895\text{mseg}$) nos ubicamos en el lugar para leer los datos que corresponden a la dirección ,luego de leer esta retardamos 1,778 mseg y nos encontramos nuevamente en condiciones de leer el nuevo bit ,así hasta completar la lectura del comando.Luego de verificar que la dirección corresponda a 7 que es nuestro caso ejecutamos el comando enviado ,en caso de que la dirección enviada difiera de 7 se rechaza el comando recibido ,poniéndose a la espera de uno nuevo.

En nuestro caso particular usaremos las teclas que corresponden a los números del 0 al 9, canal + canal - y usaremos la tecla SLEEP para poner los canales superiores al 99 la llamaremos tecla 1XX ,las demás teclas de este control las anularemos pegándole cinta adhesiva en los contactos ,quedando así sin ningún efecto .

Ahora solo tendremos que centrar nuestra atención en la construcción del convertor de F.I a la frecuencia del canal 4 ,siendo este el ultimo escollo que tendremos que sortear para poder sintonizar nuestro programa favorito ,y así dar por terminada nuestra tarea

CONVERSIÓN DE FRECUENCIA

La señal que nos entrega el sintonizador no la podemos aplicar directamente al TV ,debemos de trasladarla de frecuencia para que corresponda a un canal que si puede sintonizar el TV,En este caso optamos por el traslado al canal 4 ,con lo cual el tv quedara sintonizando el canal 4 de forma permanente y los cambios de canales lo efectuaremos nosotros mediante los contoles.En el circuito ingresa la frecuencia intermedia proporcionada por el sintonizador pasa por el filtro saw y es amplificada por un transistor ingresa a la base del transistor mezclador .Por otra parte tenemos un transistor que actúa de oscilador proporcionando una frecuencia fija de 113 Mhz ,la señal generada entra en un transistor que actúa como driver y entrega la señal por emisor ,la cual pasa por un filtro P I que atenúa las armónicas generadas por el oscilador ,dando paso a la frecuencia fundamental (113Mhz) esta ingresa por emisor al mezclador .El transistor mezclador en su colector tiene un circuito resonante LC en la frecuencia de 67,25 Mhz atenuando las demás frecuencias ,con lo cual obtenemos la salida que corresponde al canal 4 mediante un condensador de desacople pasa a un filtro PI que atenúa las demás frecuencias superiores a la buscada ,ahora si podemos conectarla al TV



Ajustes del conversor

Luego de armar la plaqueta y montar todos sus componentes ,llega el momento de ajustarlo para un buen funcionamiento,para ello tenemos que recurrir a un frecuencimetro que nos permitirá asegurarnos de poner el oscilador en la frecuencia adecuada,luego de alimentar el circuito debemos de variar el trimer del oscilador hasta observar en la lectura del frecuencimetro 113 Mhz y testear si a la salida del filtro tenemos la misma frecuencia ,el circuito tanque que esta conectado en el colector del transistor mezclador se lo ajusta hasta obtener la mejor imagen sin distorsiones en pantalla.

Hay que tener en cuenta a la hora del diseño y armado de la plaqueta que estamos trabajando en radio frecuencia y por tanto tomar los recaudos necesarios .Es conveniente que el conversor quede lo mas separado posible del resto del circuito y además de proporcionarle un blindaje metálico adecuado ,para que no afecte a los demás circuitos y que no sea perturbado el en su funcionamiento .La conexión que trae la señal de F.I debe ser con cable brindado .

Forma de obtener las tablas para los canales

En nuestro programa debimos de implementar las tablas para que se puede sintonizar cada uno de los canales ,ahora explicare la forma como se las obtuvo
Para ellos supondremos que queremos sintonizar el canal 2 cuya frecuencia intermedia de video es 55,25MHZ

Como el MC44817 al recibir tensión ,el bit T5 se pone en cero ,por lo tanto selecciona que el oscilador de 4Mhz sera dividida por 512 .

$4000000 \text{ Hz} / 512 = 7812.5 \text{ Hz}$ frecuencia de referencia a comparar

Seguiremos con el ejemplo del canal 2 de TV

Frec. para sintonizar el canal 2 = frecuencia intermedia + frecuencia de video del canal 2

$F_{osc} = 45.75 \text{ Mhz} + 55.25 \text{ Mhz} = 101 \text{ Mhz}$

Si suponemos que el PLL esta enganchado, y que la señal del oscilador se hace pasar por el prescalador +8 la frecuencia de entrada al divisor programable será de:

$101000000 \text{ Hz} / 8 = 12625000 \text{ HZ}$ o sea 12.625 Mhz

El divisor tiene 15 bits o sea 32767 posibles divisiones programables (n-1) y es un contador descendente.

Si queremos que se enganche, deberán ser iguales las frecuencias, es decir tendremos que lograr que la salida del divisor programable sea de 7812.5Hz.

Para obtener el divisor hacemos los calculaos al revés:

Frecuencia de entrada al divisor programable = $101000000 \text{ Hz} / 8 = 12625000 \text{ Hz}$

Numero divisor = $12625000 \text{ Hz} / 7812.5 \text{ Hz} = 1616$

Entonces tendríamos $7812.5 \text{ Hz} * 1616 * 8 = 101000000 \text{ Hz}$

O sea que para sintonizar el canal 2 debemos dividir por 1616. así debemos hacer los calculaos para cada uno de los canales ,en nuestro caso como las tablas no aceptan valores mayores de 255 tuvimos que dividirla en dos de 8 bit ,encontrándose en una la parte alta **divh** y en la otra tabla la parte baja **divl** , expresadas en hexadecimal.

En la siguiente tabla podemos observar la frecuencia que le corresponde a cada canal la frecuencia del oscilador la división que debemos efectuarle y la banda que le corresponde

CANAL	Frec.Video(MHZ)	Frec. OSC(MHZ)	DIVISION	BANDA	divh	Divl
1	73.25	119	1904	1	7	70
2	55.25	101	1616	1	6	50
3	61.25	107	1712	1	6	B0
4	67.25	113	1808	1	7	10
5	77.25	123	1968	1	7	B0
6	83.25	129	2064	1	8	10
7	175.25	221	3536	2	D	D0
8	181.25	227	3632	2	E	30
9	187.25	233	3728	2	E	90
10	193.25	239	3824	2	E	F0
11	199.25	245	3920	2	F	50
12	205.25	251	4016	2	F	B0
13	211.25	257	4112	2	10	10
14	121.25	167	2672	2	A	70
15	127.25	173	2768	2	A	D0
16	133.25	179	2864	2	B	30
17	139.25	185	2960	2	B	90
18	145.25	191	3056	2	B	F0
19	151.25	197	3152	2	C	50
20	157.25	203	3248	2	C	B0
21	163.25	209	3344	2	D	10
22	169.25	215	3440	2	D	70
23	217.25	263	4208	2	10	70
24	223.25	269	4304	2	10	D0
25	229.25	275	4400	2	11	30
26	235.25	281	4496	2	11	90
27	241.25	287	4592	2	11	F0
28	247.25	293	4688	2	12	50
29	253.25	299	4784	2	12	B0
30	259.25	305	4880	2	13	10
31	265.25	311	4976	2	13	70
32	271.25	317	5072	2	13	D0
33	277.25	323	5168	2	14	30
34	283.25	329	5264	2	14	90
35	289.25	335	5360	2	14	F0
36	295.25	341	5456	2	15	50

37	301.25	347	5552	2	15	B0
38	307.25	353	5648	2	16	10
39	313.25	359	5744	2	16	70
40	319.25	365	5840	8	16	D0
41	325.25	371	5936	8	17	30
42	331.25	377	6032	8	17	90
43	337.25	383	6128	8	17	F0
44	343.25	389	6224	8	18	50
45	349.25	395	6320	8	18	B0
46	355.25	401	6416	8	19	10
47	361.25	407	6512	8	19	70
48	367.25	413	6608	8	19	D0
49	373.25	419	6704	8	1A	30
50	379.25	425	6800	8	1A	90
51	385.25	431	6896	8	1A	F0
52	391.25	437	6992	8	1B	50
53	397.25	443	7088	8	1B	B0
54	403.25	449	7184	8	1C	10
55	409.25	455	7280	8	1C	70
56	415.25	461	7376	8	1C	D0
57	421.25	467	7472	8	1D	30
58	427.25	473	7568	8	1D	90
59	433.25	479	7664	8	1D	F0
60	439.25	485	7760	8	1E	50
61	445.25	491	7856	8	1E	B0
62	451.25	497	7952	8	1F	10
63	457.25	503	8048	8	1F	70
64	463.25	509	8144	8	1F	D0
65	469.25	515	8240	8	20	30
66	475.25	521	8336	8	20	90
67	481.25	527	8432	8	20	F0
68	487.25	533	8528	8	21	50
69	493.25	539	8624	8	21	B0
70	499.25	545	8720	8	22	10
71	505.25	551	8816	8	22	70
72	511.25	557	8912	8	22	D0
73	517.25	563	9008	8	23	30
74	523.25	569	9104	8	23	90
75	529.25	575	9200	8	23	F0
76	535.25	581	9296	8	24	50
77	541.25	587	9392	8	24	B0
78	547.25	593	9488	8	25	10
79	553.25	599	9584	8	25	70
80	559.25	605	9680	8	25	D0
81	565.25	611	9776	8	26	30
82	571.25	617	9872	8	26	90
83	577.25	623	9968	8	26	F0

84	583.25	629	10064	8	27	50
85	589.25	635	10160	8	27	B0
86	595.25	641	10256	8	28	10
87	601.25	647	10352	8	28	70
88	607.25	653	10448	8	28	D0
89	613.25	659	10544	8	29	30
90	619.25	665	10640	8	29	90
91	625.25	671	10736	8	29	F0
92	631.25	677	10832	8	2A	50
93	637.25	683	10928	8	2A	B0
94	643.25	689	11024	8	2B	10
95	91.25	137	2192	2	B	90
96	97.25	143	2288	2	B	F0
97	103.25	149	2384	2	9	50
98	109.25	155	2480	2	9	B0
99	115.25	161	2576	2	A	10
100	649.25	695	11120	8	2B	70
101	655.25	701	11216	8	2B	D0
102	661.25	707	11312	8	2C	30
103	667.25	713	11408	8	2C	90
104	673.25	719	11504	8	2C	F0
105	679.25	725	11600	8	2D	50
106	685.25	731	11696	8	2D	B0
107	691.25	737	11792	8	2E	10
108	697.25	743	11888	8	2E	70
109	703.25	749	11984	8	2E	D0
110	709.25	755	12080	8	2F	30
111	715.25	761	12176	8	2F	90
112	721.25	767	12272	8	2F	F0
113	727.25	773	12368	8	30	50
114	733.25	779	12464	8	30	B0
115	739.25	785	12560	8	31	10
116	745.25	791	12656	8	31	70
117	751.25	797	12752	8	31	D0
118	757.25	803	12848	8	32	30
119	763.25	809	12944	8	32	90
120	769.25	815	13040	8	32	F0
121	775.25	821	13136	8	33	50
122	781.25	827	13232	8	33	B0
123	787.25	833	13328	8	34	10
124	793.25	839	13424	8	34	70
125	799.25	845	13520	8	34	D0

```

;
;           Programa que controla un sintonizador de tv
;   maneja el MC14499 dos pulsadores y un control remoto Philips en modo Experimental
;
;           El ultimo canal utilizado queda grabado en la EEprom
;
;           Camilo Monetta
;           cmprod@adinet.com.uy
;           Salto,Uruguay
;           22/10/01 17:16
;

```

```

;XT=4MHZ
;WDT=OFF
;PWRTE=ON

```

```

list    p=16c84
#include<p16c84.inc>
__config _XT_OSC & _WDT_OFF & _PWRTE_ON

```

```

estado equ    0x03
pcl     equ    0x02
pb      equ    0x06 ;puerto B
pa      equ    0x05 ;puerto A
dat_con1 equ 0x08 ;dato lectura/escritura de eeprom
adr_con2 equ 0x09 ;direcciona acceder a la eeprom
d1      equ    0x10 ;digito a mostrar 1
d2      equ    0x11 ;digito a mostrar 2
d3      equ    0x12 ;digito a mostrar 3
aux     equ    0x13 ;contador auxiliar
aux1    equ    0x14 ;contador auxiliar 1
dato    equ    0x15 ;dato a mandar
rotar   equ    0x16 ;numero de veces a rotar
canal   equ    0x17 ;canal
banda   equ    0x18 ;banda a sintonizar
nbit    equ    0x19 ;numero de bit a enviar
datomc  equ    0x20 ;dato para el mc14499
dividendo equ 0x21
divisor  equ    0x22
resto   equ    0x23
cociente equ 0x24
ir_dir  equ    0x25 ;direccion de identificación del control remoto Philips (7 Experimental)
ir_dat  equ    0x26 ;comando enviado por el control remoto
n       equ    0x27 ;variable
eeif    equ    4   ;indica estado de la escritura
wrer    equ    3   ;señalizado de error de escritura
wren    equ    2   ;activación de escritura
wr      equ    1   ;control de escritura
rd      equ    0   ;control de lectura

```

```

;=====
;=====
;===== Variables del MC14499P =====
;=====
mc_dat equ    0   ;linea de datos
mc_en  equ    1   ;enable del MC14499
mc_ck  equ    2   ;pin de la señal de clock
;=====
;=====

```

```

;===== Variables del Sintonizador TUGH9-A04M =====
s_en equ 3 ;enable del sintonizador
s_ck equ 4 ;clock del sintonizador
s_dat equ 5 ;dato del sintonizador
;=====
;=====

org 0
bsf estado,5 ;pone bit 5 de status = 1 . Ir al Banco 1
clrf pb ;selecciona el puerto B como salida
movlw 0xff
movwf pa ;puerto A como entrada
bcf estado,5 ;pone bit 5 de status = 0 . Ir al banco 0
clrf pb
bsf pb,mc_en ;enable=1 del MC
clrf d3 ;d3=0
clrf d2 ;d2=0
clrf d1 ;d1=0
;=====

call eelect ;lee la eeprom y trae el ultimo canal utilizado
movlw .255 ;w=255
xorwf canal,w ;w= canal xor w
btfss estado,2 ;el resultado =0?
goto chay ;hay grabado un canal
movlw .2 ;w=2
movwf canal ;canal=2
chay call display ;va a la rutina para mostrar el canal
call sintonia ;sintoniza el canal

inicio btfss pa,2 ;compara si el bit 2 del pa es uno
call pmas ;si es cero va a pmas
btfss pa,3 ;compara si el bit 3 del pa es uno
call rx3 ;va a la recepci3n del infrarrojo
btfss pa,4 ;compara si el bit 4 del pa es uno
call pmenos ;si es cero va a pmenos
goto inicio ;no se presiono ning3n pulsador
;=====

;=====Secuencia a seguir para sintonizar un canal =====
sintonia
bsf pb,s_en ;pone en alto enabled del sintonizador
nop
call enbanda ;va a la subrutina de envio de banda
call endivh ;va a la subrutina de envio de division alta
call endivl ;va a subrutina de envio de division baja
bcf pb,s_en ;pone en bajo enabled del sintonizador
return ;retorna a rutina principal
;=====

```

```

;===== Secuencia para enviar bit =====
enviabit
    btfss dato,7    ;pregunta si el bit 7 de dato es 1
    bcf pb,s_dat    ;dato=0
    btfsc dato,7    ;pregunto si dato es cero
    bsf pb,s_dat    ;si es cero
    bsf pb,s_ck     ;clock =1
    nop
    bcf pb,s_ck     ;clock=0
    return

;=====
;===== Envia la banda =====
enbanda
    movlw HIGH divl
    movwf PCLATH
    movf canal,w    ;w=canal
    call divl       ;Busca el valor de la banda
    movwf dato
    clrf PCLATH
    movlw b'00001111' ;w=00001111
    andwf dato,w    ;w= dato and 00001111 elimino los 4 bit de mas peso
    movwf dato      ;dato=w
    rlf dato,1
    rlf dato,1
    rlf dato,1
    rlf dato,1
    movlw .4        ;w=4
    movwf rotar     ;rotar=w=4
cban  call enviabit
    rlf dato,1
    decfsz rotar,1
    goto cban
    return

;=====

;===== Envia la parte alta de la division =====
endivh
    movlw HIGH divh
    movwf PCLATH
    movf canal,w    ;w=canal
    call divh
    movwf dato      ;dato=valor traído de la tabla
    clrf PCLATH
    rlf dato,1
    rlf dato,1     ;rote 2 veces el dato de la tabla divh
    movlw .6
    movwf rotar     ;rotar=w=7
cdivh call enviabit
    rlf dato,1
    decfsz rotar,1
    goto cdivh
    return

;=====

```

```

;===== Envia la parte baja de la division =====
enddivl
    movlw HIGH divl
    movwf PCLATH
    movf canal,w          ;w=canal
    call divl
    movwf dato           ;valor traído de la tabla
    clrf PCLATH
    movlw b'11110000'    ;w=11110000 b
    andwf dato,w         ;w= dato and 1111000 elimino los 4 bit de menos peso
    movwf dato           ;dato=w
    movlw .8
    movwf rotar          ;rotar=w=8
cdivl  call enviabit
    rlf dato,1
    decfsz rotar,1
    goto cdivl
    return
;=====

```

```

;=====
;===== RUTINAS PARA EL MC =====
;=====

```

```

;===== SE PULSO LA TECLA - =====
pmenos  decf canal,1    ;canal=canal-1
        movlw .1        ;w=1
        xorwf canal,w
        btfss estado,2 ;canal=1?
        goto no1        ;canal>1
        movlw .125
        movwf canal
no1     call pp
        return
;=====

```

```

;===== SE PULSO LA TECLA + =====
pmas    incf canal,1    ;canal=canal+1
        movlw .126     ;w=126
        xorwf canal,w
        btfss estado,2 ;canal=126?
        goto no126
        movlw .2       ;w=2
        movwf canal
no126   call pp
        return
;=====

```

```

;===== Rutina comun a p+ y p- =====
pp    clrf    cociente
      call display    ;muestra el nuevo canal
      call sintonia  ;sintoniza el canal
      call eegrab    ;graba el canal en la EEprom
      call retardo
      return

;=====

;===== Separa de canal cada digito a mostrar D3 D2 D1 =====
d3d2d1
      movf canal,w      ;w=canal
      movwf dividendo  ;dividendo=canal
      movlw .100       ;w=100
      movwf divisor    ;divisor=100
      call divide      ;divide canal/100
      movf cociente,w  ;w=cociente
      movwf d3         ;d3=cociente
      movf resto,w     ;w=resto
      movwf dividendo  ;dividendo=resto
      movlw .10        ;w=10
      movwf divisor    ;divisor=10
      call divide
      movf cociente,w  ;w=cociente
      movwf d2         ;d2=cociente
      movf resto,w     ;w=resto
      movwf d1         ;d1=resto
      return

;=====

;===== DISPLAY =====
display
      call d3d2d1      ;separa cada digito
      bcf pb,mc_en     ;enable=0 MC14499 habilitado para recibir datos
      clrf datomc      ;dato=0=punto
      call mc_envia
      movf d1,w        ;w=d1
      movwf datomc     ;dato=d1
      call mc_envia
      movf d2,w        ;w=d2
      movwf datomc     ;dato=d2
      call mc_envia
      movf d3,w        ;w=d3
      movwf datomc     ;dato=d3
      call mc_envia
      clrf datomc      ;dato=0=4º digito no usado en este caso
      call mc_envia
      bsf pb,mc_en     ;enable=1 MC14499 no recibe mas datos
      bsf pb,mc_ck
      return

;=====

```

```

;===== Secuencia para enviar bit =====
mc_envia
    movlw .4      ;w=4
    movwf rotar   ;rotar=4
sigue  call mc_enbit
       rlf datomc,1 ;roto el dato 1 lugar a la izquierda
       decfsz rotar,1 ;rotar=rotar-1
       goto sigue
       return
;=====

;=====
mc_enbit
    btfss datomc,3 ;pregunta si el bit 3 de dato es 1
    bcf pb,mc_dat ;dato=0
    btfsc datomc,3 ;pregunto si dato es cero
    bsf pb,mc_dat ;si es cero
    bsf pb,mc_ck ;clock =1
    nop
    bcf pb,mc_ck ;clock=0
    return
;=====

;===== Retardo =====
retardo movlw .155      ;w=155
        movwf aux      ;aux=255
p44     movlw .255      ;w=255
        movwf aux1     ;aux1=255
redo    nop            ;1 uSEG
        nop            ;1 uSEG
        nop            ;1 uSEG
        nop            ;1 uSEG
        nop            ;1 uSEG
        nop            ;1 uSEG
        nop            ;1 uSEG
        decfsz aux1,1  ;decrementa aux1 y salta la siguiente instruccion si es cero
        goto redo     ;si aux1 es distinto de cero va a redo
        decfsz aux,1  ;decrementa aux y salta la siguiente instruccion si es cero
        goto p44      ;si aux es distinto de cero va a p44
        return        ;retorna a la rutina que lo llamo
;=====

;===== Rutina que divide IN(dividendo/divisor ) OUT(cociente,resto) =====
divide:
    clrf resto      ;borra el resto
    movlw 8         ;Para dividir
    movwf aux       ;8 bits

divloop: rlf dividendo,1 ;Correr el dividendo
         rlf resto,1
         movf divisor,w
         subwf resto,w

```

```

CHECKLESS: BNC   NOSUB      ;Si la porcion corrida del dividendo fu,
           movf  divisor,W   ;De otro modo, REMAINDER = REMAINDER - DIVISOR.
           subwf resto

```

```

NOSUB:    rlf   cociente,1
           decfsz aux,1
           goto  divloop
           return

```

```

;=====

```

```

;===== Graba el canal en la EEprom =====

```

```

eegrab
           clrf  adr_con2      ;eeadr=0 direcciona escribir
           movf  canal,w       ;w=canal
           movwf dat_con1      ;eedata=canal valor a grabar

           bsf  estado,5       ;paso al banco 1
           clrf dat_con1       ;borra el dato
           bsf  dat_con1,wren   ;Permite escrituras
           movlw 0x55           ;w=55H
           movwf adr_con2
           movlw 0xAA           ;w=AAH
           movwf adr_con2
           bsf  dat_con1,wr     ;Inicia un ciclo de escritura
ll        btfsz dat_con1,wr    ;pregunta si termino la escritura
           goto ll             ;no termino
           bcf  dat_con1,wren   ;No permite mas escrituras
           bcf  dat_con1,eEIF   ;Borra el bit indicador de escritura terminada
           bcf  estado,5       ;Paso al banco 0
           return

```

```

;=====

```

```

;===== Lectura del canal grabado en la EEprom =====

```

```

eelect
           clrf  adr_con2      ;eeadr=0 direccion a leer
           bsf  estado,5       ;paso al banco 1
           bsf  dat_con1,rd     ;Comienzo de lectura
           bcf  estado,5       ;paso al banco 0
           movf  dat_con1,w     ;w=valor leído
           movwf canal         ;canal=w
           return

```

```

;=====

```

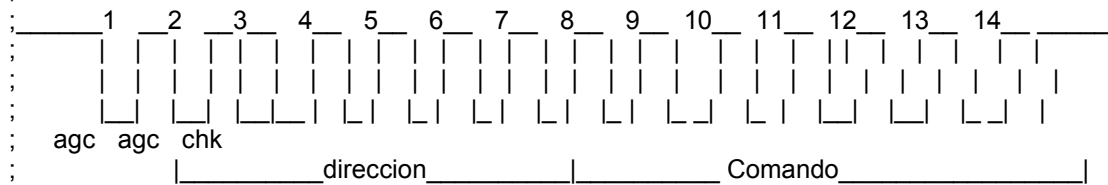


```

;=====
;===== RUTINAS PARA EL CONTROL REMOTO =====
;=====

```

FORMATO DE LA SEÑAL RECIBIDA



```

;Direccion= 0 para TV
;Comandos
;0 al 9 numeros
;20H canal +
;21H canal -
;10H volumen +
;11H volumen -

```

```

;===== RX3 =====

```

```

rx3  call rx          ;va a leer el primer dato
      movf ir_dat,w
      movwf d3

      movlw .255      ;w=255
      xorwf d3,w      ;w= d3 xor w
      btfss estado,2 ;el resultado=0?
      goto p2         ;sigue preguntando
      clrf d3
      return          ;Tengo error en la recepcion

p2    movlw .38       ;w=38 SLEEP (1--)
      xorwf d3,w      ;w=d3 xor w
      btfss estado,2 ;el resultado=0?
      goto chd        ;Se pregunta por ch+ ch- o d2 y d2
      movlw .1
      movwf d3        ;d3=1 el canal esta comprendido entre el 100 al 125
      goto ddd

chd   movlw .32       ;CHANNEL +
      xorwf d3,w      ;w= d3 xor d3
      btfss estado,2 ;Se presiono Chanel +
      goto chm        ;NO se presiono
      incf canal,1    ;canal=canal+1
      movlw .126     ;w=126
      xorwf canal,w
      btfss estado,2 ;el resultado=0?
      goto oich       ;canal<126
      movlw .2        ;w=2
      movwf canal     ;canal=2
      goto oich       ;muestra el canal

chm   movlw .33       ;w=33 CHANNEL -
      xorwf d3,w      ;w= d3 xor w

```

```

        btfss estado,2 ;Se presiono Chanel -
        goto d1d2      ;NO se presiono
        decf canal,1   ;canal=canal-1
        movlw .1       ;w=1
        xorwf canal,w
        btfss estado,2 ;el resultado=0?
        goto oich      ;Canal>1
        movlw .125     ;canal=1
        movwf canal    ;canal=125
        goto oich

d1d2    movf d3,w
        movwf d2       ;d2=d3
        clrf d3
        goto ud1

ddd
q        call retardo
        btfss pa,3
        goto t
        goto q

t        call rx
        movf ir_dat,w
        movwf d2       ;d2 = ok

ud1
r        call retardo
        btfss pa,3
        goto ds
        goto r

ds       call rx
        movf ir_dat,w
        movwf d1       ;d1 = ok

;===== Obtengo el Canal =====
oi       clrf canal
        movlw .0       ;w=0
        xorwf d3,w     ;w= d3 xor w
        btfss estado,2 ;El resultado es =0?
        call mul100    ;d3>0
        movlw .0
        xorwf d2,w     ;w= d2 xor w
        btfss estado,2 ;el resultado=0?
        call mul10     ;d2>0
        movf d1,w     ;w=d1
        addwf canal,1  ;canal=canal+d1

oich     call pp ;rtuina que muestra ,sintoniza y graba el canal
        return

;=====

```

```

;===== MUL100 =====
mul100
    movlw .100
    movwf aux
mul1    incf canal,1    ;canal=canal+1
        decfsz aux,1    ;aux=aux-1
        goto mul1
        return
;=====

;===== MUL10 =====
mul10   movf d2,w      ;w=d2
        movwf aux1
mul2    movlw .10
        movwf aux
mul     incf canal,1    ;canal=canal+1
        decfsz aux,1    ;aux=aux-1
        goto mul
        decfsz aux1,1   ;aux1=aux1-1
        goto mul2
        return
;=====

;===== RX =====
;Espera a que pasen los 3 primeros bit que son de AGC CHECK

rx      clrf ir_dat
        clrf ir_dir
        call ret4_7      ;descarto los 2,75 bit de inicio

;Descarto los 5 bit de direccion
        movlw .5         ;w=5
        movwf n          ;n=5
nudir   rlf ir_dir,1
        btfss pa,3      ;pregunta si el pin RA3 es 1
        goto dircer     ;RA=0
        bsf ir_dir,0
dircer  call ret_17      ;retardo 1,778 milisegundos
        decfsz n,1      ;n=n-1
        goto nudir      ;n<>0

;Recepcion de COMANDO 6 BIT
        movlw .6         ;w=6
        movwf n          ;n=6
nucom   rlf ir_dat,1
        btfss pa,3      ;pregunta si el pin RA3 es 1
        goto cero       ;RA=0
        bsf ir_dat,0    ;pone en 1 el bit 0
cero    call ret_17      ;retardo 1,778 milisegundos
        decfsz n,1      ;n=n-1
        goto nucom      ;n<>0

```

```

;Compara si la direccion es 7
    movlw .7
    xorwf ir_dir,w
    btfsz estado,2
    goto err      ;No es igual
    return        ;Es igual
err    movlw .255
        movwf ir_dat
        return

;=====

;===== Retardo de 1,778 milisegundos =====
ret_17    movlw .254      ;w=42
           movwf aux1     ;aux1=82
redir     nop
           nop
           nop
           nop
           decfsz aux1,1  ;decrementa aux1 y salta la siguiente instruccion si es cero
           goto redir    ;si aux1 es distinto de cero va a redo
           return        ;retorna a la rutina que lo llamo

;=====

;===== Retardo de (2,75* 1,778)=4,889 milisegundos =====
ret4_7    movlw .4        ;w=7
           movwf aux      ;aux=7
p444      movlw .244      ;w=42
           movwf aux1     ;aux1=82
redo4     nop
           nop
           decfsz aux1,1  ;decrementa aux1 y salta la siguiente instruccion si es cero
           goto redo4    ;si aux1 es distinto de cero va a redo
           decfsz aux,1   ;decrementa aux y salta la siguiente instruccion si es cero
           goto p444     ;si aux es distinto de cero va a p44
           return        ;retorna a la rutina que lo llamo

;=====

                                org h'300'
;===== Tabla con la parte alta de la division =====
divh
    addwf pcl,f
    DT H'00',H'07',H'06',H'06',H'07',H'07',H'08',H'0D',H'0E',H'0E'
    DT H'0E',H'0F',H'0F',H'10',H'0A',H'0A',H'0B',H'0B',H'0B',H'0C'
    DT H'0C',H'0D',H'0D',H'10',H'10',H'11',H'11',H'11',H'12',H'12'
    DT H'13',H'13',H'13',H'14',H'14',H'14',H'15',H'15',H'16',H'16'
    DT H'16',H'17',H'17',H'17',H'18',H'18',H'19',H'19',H'19',H'1A'
    DT H'1A',H'1A',H'1B',H'1B',H'1C',H'1C',H'1C',H'1D',H'1D',H'1D'
    DT H'1E',H'1E',H'1F',H'1F',H'1F',H'20',H'20',H'20',H'21',H'21'
    DT H'22',H'22',H'22',H'23',H'23',H'23',H'24',H'24',H'25',H'25'
    DT H'25',H'26',H'26',H'26',H'27',H'27',H'28',H'28',H'28',H'29'
    Dt H'29',H'29',H'2A',H'2A',H'2B',H'08',H'08',H'09',H'09',H'0A'
    DT H'2B',H'2B',H'2C',H'2C',H'2C',H'2D',H'2D',H'2E',H'2E',H'2E'
    DT H'2F',H'2F',H'2F',H'30',H'30',H'31',H'31',H'31',H'32',H'32'
    Dt H'32',H'33',H'33',H'34',H'34',H'34'

;=====

```

;==== Tabla con la parte baja de la division y en los 4 bit menos significativos la banda ====

divl

```
addwf pcl,f
DT H'00',H'71',H'51',H'B1',H'11',H'B1',H'11',H'D2',H'32',H'92'
DT H'F2',H'52',H'B2',H'12',H'72',H'D2',H'32',H'92',H'F2',H'52'
DT H'B2',H'12',H'72',H'72',H'D2',H'32',H'92',H'F2',H'52',H'B2'
DT H'12',H'72',H'D2',H'32',H'92',H'F2',H'52',H'B2',H'12',H'72'
DT H'D2',H'32',H'92',H'F2',H'52',H'B8',H'18',H'78',H'D8',H'38'
DT H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38',H'98',H'F8'
DT H'58',H'B8',H'18',H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8'
DT H'18',H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78'
DT H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38'
DT H'98',H'F8',H'58',H'B8',H'18',H'91',H'F1',H'51',H'B1',H'11'
DT H'78',H'D8',H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8'
DT H'38',H'98',H'F8',H'58',H'B8',H'18',H'78',H'D8',H'38',H'98'
DT H'F8',H'58',H'B8',H'18',H'78',H'D8'
```

=====

end ;fin del programa

Programa

El programa en esta dividido en módulos que se encargan de una tarea específica para ello tenemos uno que se encarga de sintonizar el canal ,otro para el manejo del display otro para el para la recepción del control remoto y otro para lectura y escritura de la eeprom.

Este al comenzar hace una lectura de la eeprom y comprueba si hay un canal grabado en el caso de ser la primera vez que va a funcionar no encuentra ningún canal grabado y por lo tanto leerá el valor 255 (FF)

En este caso setea el canal 2 y luego sintoniza y muestra el canal en el display y queda a la espera de una nueva orden tanto de parte del control remoto o de los pulsadores ,al efectuar un cambio este actualiza el canal y el display y lo graba en la eeprom .En el caso de corte de energía o que lo apagamos quedo almacenado el ultimo canal utilizado.Al encenderlo nuevamente se inicia en este canal