

Generador de patrones de video

Introducción

El objetivo de este proyecto ha sido la construcción de un generador de patrones de video compuesto, cuyas aplicaciones se extienden desde el campo de la enseñanza de la señal de video hasta la reparación y ajuste de equipos receptores de televisión. No se ha pretendido la realización de un equipo con fines comerciales, ya que no se garantizan las especificaciones de un generador de video profesional, si bien se logrado un equipo de muy alta calidad.

Especificaciones

| | |
|----------------------------|--|
| Patrones: | Barras de color, Raster, Cross-hatch, Puntos. |
| Controles: | R, G, B, luminancia, crominancia y burst de color independientes. |
| Salida de video: | video compuesto, 1 Vp-p en carga de 75 ohms. |
| Sistema de color: | PAL-N (opcional PAL-B/G/I cambiando el oscilador de croma). |
| Sistema de barrido: | entrelazado (Barras y Raster), no entrelazado (Cross-hatch y Puntos). |
| Alimentación: | 12 Vdc (8 x AA pilas alcalinas). |
| Consumo máximo: | 70 mA (Raster blanco). |
| Aplicaciones: | ajustes de receptores de televisión (pureza, convergencia, linealidad, etc.), localización de fallas en el sector de video de televisores y videocaseteras, enseñanza de la generación de video compuesto. |

Generalidades

El ojo humano es capaz de percibir imágenes mediante receptores ubicados en la retina. Hay dos tipos de receptores, según su función: los bastones, encargados de percibir imágenes en blanco y negro, y los conos, a cargo de la percepción del color. Si nos concentramos en el estudio de los conos, veremos que hay de tres tipos: los que reaccionan frente a la luz roja, los que lo hacen frente a luz verde y finalmente los que son sensibles a la luz azul. Solo se perciben tres colores, sin embargo nosotros “vemos” todos los colores que nos rodean.

Aquí se hace evidente una regla básica del color: **para conocer la información de color de un objeto basta con tener la proporción de los tres colores básicos: Rojo, Verde y Azul**. Por esta razón a dichos colores se los conoce como **Colores Primarios**, ya que con la combinación de los mismos se pueden obtener todos los demás.

Cabe hacer una aclaración, ya que tal vez alguna vez se nos haya dicho que los primarios son el rojo, amarillo y azul. Es más, si observamos los colores utilizados por las impresoras de tinta, muy comunes en la actualidad, veremos que utilizan tres cartuchos, magenta, amarillo y cian. No utilizan los primarios, y sin embargo imprimen “a todo color”. Alguien debe estar equivocado...

La “confusión” resulta de la existencia de dos grupos de colores primarios: los **Primarios Aditivos** y los **Primarios Sustractivos**. Para entender la diferencia entre ambos veamos un par de ejemplos.

Supóngase que iluminamos una pared blanca con una luz verde. Obviamente se verá verde, ya que este es el color que refleja la pared y llega a nuestros ojos. Si ahora cambiamos la luz a rojo, igualmente veremos el color rojo. Pero si iluminamos al mismo tiempo con verde y rojo, el color que percibiremos será el amarillo, el cual resulta de la suma o adición de los colores verde y rojo. Como se observa, la suma de estos colores resulta en la generación de un color secundario, por esto se denominan primarios aditivos.

Ahora supongamos que pintamos un papel blanco con pintura amarilla, y lo iluminamos con luz blanca (contiene todos los colores). Obviamente se ve amarillo. ¿Por qué?. Porque la pintura retiene (sustrae) todos los componentes de la luz blanca que la ilumina y devuelve solo el amarillo, el cual llega a nuestros ojos. Si ahora agregamos pintura cian (celestes intenso), el color resultante es verde. Quiere decir que la mezcla ha retenido todos los colores de la luz blanca menos el verde. ¿Cómo se explica esto?. Veamos algunas ecuaciones.

Si sumamos todos los colores primarios aditivos obtenemos blanco:

LUZ ROJA + LUZ VERDE + LUZ AZUL = LUZ BLANCA

Si los sumamos de a dos:

LUZ ROJA + LUZ VERDE = LUZ AMARILLA

LUZ ROJA + LUZ AZUL = LUZ MAGENTA

LUZ VERDE + LUZ AZUL = LUZ CIAN

De aquí surge una relación clave en nuestro análisis: los llamados “colores primarios sustractivos” resultan de combinar dos a dos los primarios aditivos.

Entonces, cuando vemos la pintura amarilla en realidad estamos viendo luz roja y luz verde combinadas. Quiere decir que la pintura amarilla sustrae el azul de la luz blanca y devuelve los otros dos primarios.

Por otra parte, la pintura cian retiene el rojo y devuelve luz verde y luz azul.

Entonces, ahora es evidente el resultado de nuestro experimento. Si mezclamos pintura amarilla (retiene el azul) con pintura cian (retiene el rojo), el único color que puede salir de esta mezcla es el verde, el cual es justamente el color que vemos.

¿Qué ocurre si mezclamos los tres colores primarios sustractivos?. Obviamente se ve negro, ya que todas la “luces” son retenidas:

CIAN + MAGENTA + AMARILLO = NEGRO

Queda aún una pregunta sin resolver. ¿Por qué habitualmente se dice que los colores primarios son rojo, amarillo y azul?. El motivo es sencillo. Desde niños se nos enseña el color mediante la pintura, y en este campo ya sabemos que trabajamos con los primarios sustractivos, amarillo, magenta y cian. A estos últimos, por ser rojizo y azulado, respectivamente, se los llama “rojo” y “azul”.

Los colores en televisión

La imagen en un televisor a colores se forma mediante la emisión de luz resultante de la excitación de la película de fósforo, que recubre internamente la pantalla, al ser alcanzada por un haz de electrones que barre periódicamente la superficie visible. Si hablamos de “emisión de luz”, inmediatamente debemos pensar en procesos “aditivos”, lo cual nos lleva a concluir que en televisión los colores primarios son el Rojo, Verde y Azul (RVA o en inglés RGB). Efectivamente, dentro del tubo de televisión se emiten tres haces de electrones, destinados cada uno a excitar una franja de fósforo en la pantalla, la cual responderá emitiendo un color característico al fósforo empleado. Naturalmente, como no podía ser de otro modo, estos colores son Rojo, Verde y Azul.

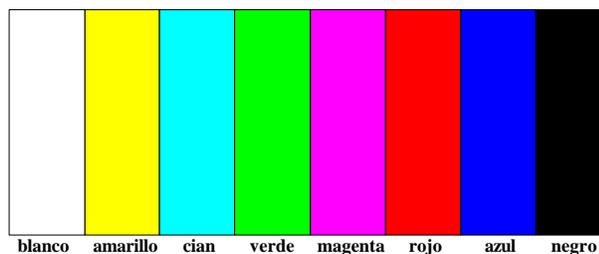
Todos los demás colores (y digo realmente “todos”) se pueden obtener combinado estos tres primarios en distintas proporciones.

Generador de barras básico

Un generador de barras de color básico podría hacer lo siguiente:

- tener tres salidas, una para cada color primario
- cada una de estas salidas se conecta a la correspondiente entrada del televisor
- el equipo generará combinaciones de sus salidas, según la siguiente tabla:

| | | | | | | | | |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Azul | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Rojo | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| Verde | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |



En esta tabla un “1” significa “presencia” del color, en tanto que un “0” es su “ausencia”. En la práctica, estos “unos” y “ceros” se representan por niveles de tensión, por ejemplo 5V y 0V respectivamente.

Como se observa, es muy sencillo construir un generador de este tipo, ya que basta un mínimo de electrónica digital para obtener estas barras. Entonces, ¿por qué complicarse más?.

La mayoría de los equipos de televisión y video no poseen entradas directas de Rojo, Verde y Azul, estas quedan reservadas para monitores destinados al campo profesional. Lo habitual es que los equipos hogareños tengan una entrada de “Video Compuesto”, denominada usualmente como “VIDEO IN”. Por esta razón, nuestro generador debe poder convertir los componentes Rojo, Verde y Azul en ese “Video Compuesto”.

Video compuesto

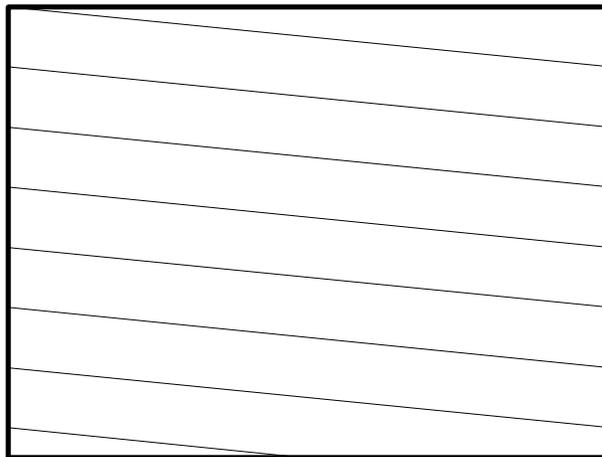
Las señales de Rojo, Verde y Azul (de ahora en más la llamaremos simplemente RGB) contienen toda la información de la imagen, pero ocuparían un ancho de banda considerable si se transmitieran, lo cual justamente es el objetivo de una “transmisión” de televisión. Para reducir este ancho de banda, además de mantener la compatibilidad entre transmisiones en “Blanco y Negro” y “Color”, se creó la señal de Video Compuesto.

En esta señal va la información de “luminosidad” (Luminancia) de una imagen, su color (Crominancia), y además todos los sincronismos necesarios para generar correctamente la imagen en la pantalla del televisor.

¿Cuáles son estos sincronismos?. Para contestar esta pregunta veamos primero como se forma una imagen en el televisor.

Un haz de electrones (consideremos uno solo, sabemos que son tres) recorre la pantalla de izquierda a derecha y de arriba a abajo, según lo muestra el esquema adjunto. A medida que recorre la pantalla excita en mayor o menor grado al fósforo que la recubre, generando una imagen.

Como se ve, si bien la imagen aparece en una pantalla de dos coordenadas (un plano), en realidad se genera mediante líneas sucesivas. Del mismo modo ingresa al equipo, línea a línea. Por lo tanto, es indudable que se necesita “sincronizar” el



haz de electrones que barre la pantalla con el barrido generado en el estudio de televisión. Si no se hiciera esto, las imágenes aparecerían cortadas, con barras inclinadas, con colores incorrectos (basta con ver un “canal codificado” para tener una idea de lo que quiero decir, ya que una manera de codificar la señal es quitarle los sincronismos).

Sincronicemos entonces el barrido del haz. Hay que informarle donde empieza una nueva imagen (sincronismo Vertical o “V”) y donde comienza cada línea (sincronismo Horizontal o “H”). Evidentemente el sincronismo Horizontal es un componente de mayor frecuencia que el Vertical, ya que dentro de cada imagen hay muchas líneas de barrido.

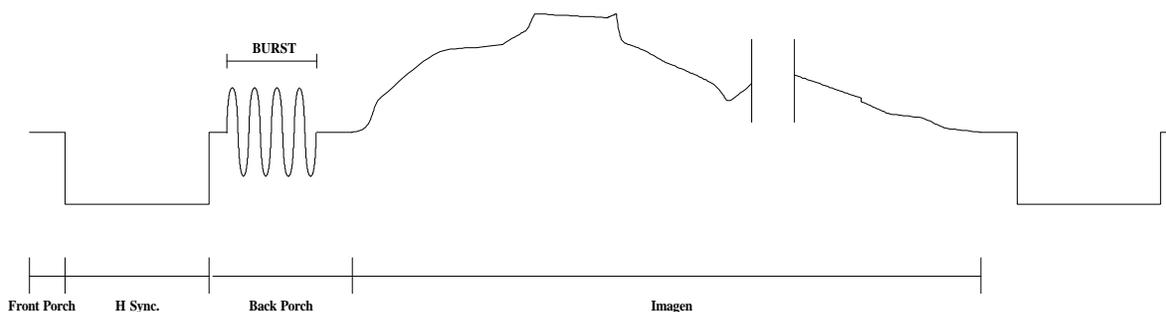
¿Cuántas líneas hay dentro de una imagen?. La respuesta es: “depende de la norma de transmisión que estemos considerando”.

Normas de transmisión

Una norma es un conjunto de parámetros adoptados como regla dentro de determinado grupo o región a fin de mantener una relación clara y sin ambigüedades entre las partes. En televisión es exactamente eso. Las normas establecen los parámetros que deben seguir tanto los equipos transmisores de señal como los receptores, a fin de que se establezca una comunicación segura y sin errores entre ambos. Entre los muchos parámetros que se fijan consideraremos solo los que atañen a nuestro proyecto.

| | |
|---|--|
| Frecuencia Horizontal (H): | frecuencia a la que se repiten las líneas de imagen. |
| Sincronismo Horizontal (H Sync): | pulso que indica el comienzo de una línea de imagen. |
| Frecuencia Vertical (V): | frecuencia a la que se repiten las imágenes (campos). |
| Sincronismo Vertical (V Sync.): | pulso que indica el comienzo de una imagen (campo). |
| Líneas Horizontales: | cantidad de líneas que forman una imagen completa (cuadro). |
| Subportadora de Color (SC): | frecuencia a la que se modula para enviar la información de color. |
| “BURST” de Color: | ráfaga de la SC que “sincroniza” la demodulación del color. |
| “Front Porch”: | intervalo de señal sin información de imagen previo al H. Sync. |
| “Back Porch”: | intervalo de señal posterior al H Sync., donde se coloca el BURST. |

Todo esto se resume en el siguiente diagrama:



En la información anterior se hizo referencia a dos palabras que aún no han sido definidas: Campo y Cuadro. Para comprender su definición debemos primero aclarar el concepto de “Barrido Entrelazado”.

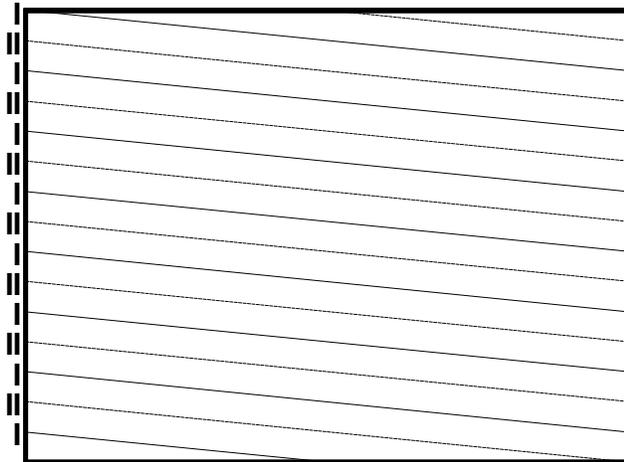
Como ya se dijo, para reconstruir la imagen en la pantalla del televisor, el haz de electrones recorre la misma línea a línea de arriba a abajo. Bastaría una pasada completa del haz de electrones para tener una imagen formada en pantalla. Sin embargo no es así. Se requiere que el haz de electrones “barra” dos veces la pantalla para formar una imagen completa. Veamos esto con más detenimiento.

Cuando la cámara de video en el estudio de televisión “escanea” la imagen que va a transmitir, lo hace dividiendo la imagen en líneas horizontales. Estas líneas serán posteriormente transmitidas al receptor de televisión para que este las reproduzca secuencialmente en la pantalla. Sin embargo, no se transmiten las líneas consecutivamente (línea 1, 2, 3, 4, ...) sino que primero se transmiten las impares (línea 1, 3, 5, ...) y luego las pares (línea 2, 4, 6, ...). Del mismo modo recorre el haz de electrones la pantalla, reproduciendo primero las líneas impares y luego las pares. Por este motivo se requiere que el haz recorra 2 veces toda la pantalla para formar una imagen completa. A las dos “semi-imagenes” se les denomina Campos (Impar y Par respectivamente) y a la imagen completa Cuadro. Esto se esquematiza en la figura adjunta.

Campo

En esta figura se ha denominado I al Campo Impar y II al Campo Par.

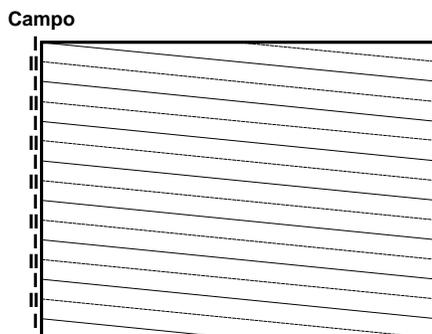
En el diagrama se puede observar que el campo impar termina en media línea horizontal y el par comienza con media línea horizontal. Esta característica la veremos a continuación cuando analicemos las dos formas de barrido que pueden encontrarse: Barrido Entrelazado y Barrido No Entrelazado. El ejemplo que acabamos de ver corresponde a una imagen generada con Barrido Entrelazado. Su nombre



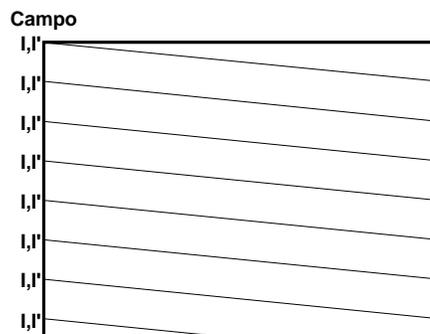
proviene del “entrelazamiento” de las líneas de barrido de dos campos sucesivos. ¿Qué ventaja tiene?. Obtener una mayor frecuencia de repetición de imágenes sin aumentar el ancho de banda. ¿Cómo es esto?. Por el canal de transmisión se envían una cierta cantidad de imágenes por segundo (digamos por ejemplo 25). En el televisor se reproducen estas 25 imágenes por segundo, pero barriendo la pantalla 50 veces por segundo. Para nuestro ojo es como si se hubieran generado 50 imágenes en un segundo, haciendo imperceptible el “parpadeo” de la imagen. Como dato adicional, en el cine ocurre lo mismo. Las películas antiguas corrían a 16 cuadros por segundo, y en ellas era muy notorio el parpadeo. Actualmente se filma a 24 cuadros por segundo, y el obturador se abre dos veces en cada cuadro durante la proyección, con el mismo propósito de disminuir el “parpadeo”.

Volviendo a la televisión, si este método es tan bueno, no tiene sentido insistir con el Barrido No Entrelazado. Sin embargo, el entrelazamiento, que ha probado ser insustituible en imágenes en movimiento, falla en imágenes fijas. ¿Por qué?. Imaginemos una línea blanca horizontal quieta sobre un fondo negro en medio de la pantalla. Cuando se barre el campo impar la línea aparecerá a determinada altura, en tanto que cuando se barra el campo par la misma aparecerá naturalmente una línea horizontal más arriba o más abajo que la anterior. Al sucederse los campos, la línea se verá “temblar” en sentido vertical, en forma poco perceptible, pero muy molesta si uno debe fijar la vista en la pantalla. ¿Cómo se soluciona?. Fácil. Se utiliza el Barrido No Entrelazado. Este es mucho más sencillo. Simplemente, las líneas de barrido se superponen campo tras campo. Esta técnica es muy utilizada en monitores de computación y en generadores de patrones cuando se utilizan señales como el “Cross-hatch” (líneas) y “Puntos”.

Barrido entrelazado



Barrido no entrelazado



Norma N

Actualmente en el mundo hay muchas normas de transmisión de televisión, denominadas con letras desde la “A” a la “N”. Algunas ya no se utilizan, pero la mayoría siguen vigentes. Veamos ahora los valores que se han establecido para los parámetros que hemos definido en la norma N, utilizada en solo tres países del mundo: Argentina, Uruguay y Paraguay.

| | |
|---|--|
| Frecuencia Horizontal (H): | 15626 Hz (duración de una línea: 64 µseg.) |
| Sincronismo Horizontal (H Sync): | 4.8 µseg. |
| Frecuencia Vertical (V): | 50 Hz (duración de un campo: 20 mseg.) |
| Sincronismo Vertical (V Sync.): | 2.5 líneas horizontales |
| Líneas Horizontales: | 625 por cuadro (312.5 por campo) |
| Subportadora de Color (SC): | 3.582056 MHz |
| “BURST” de Color: | 9 a 11 ciclos de SC |
| “Front Porch”: | 1.9 µseg. |
| “Back Porch”: | 5 µseg. |

Si comparamos estos valores con normas en uso en Europa (B, G, I) veremos que hay una gran similitud, excepto por el valor de la subportadora de color (en estos sistemas es 4.43 MHz).

Para concluir con el tema de las normas, veamos un esquema que representa una señal real de video norma N, utilizando los dos métodos de barrido ya analizados (ver figura en página siguiente).

Concentrémonos solamente en el método entrelazado. Se ve claramente como el pulso de sincronismo vertical abarca 2½ líneas horizontales (la 1, 2, y mitad de la 3 en el Campo I). Pero además se ve algo no mencionado hasta ahora: aparecieron unos pulsos de Pre y Pos-Ecualización. ¿Qué es esto?. La detección del sincronismo vertical en el televisor se realiza integrando la señal de video que llega, y cuando esta alcanza un determinado valor se dispara el barrido vertical. La integración se lleva a cabo mediante un circuito RC, y se monitorea la carga del capacitor para decidir cuando disparar el vertical. Dependiendo de la imagen previa al pulso de sincronismo vertical, el capacitor podrá estar más o menos cargado, haciendo que los barridos verticales no se disparen siempre en un mismo punto, provocando inestabilidades en la imagen (temblor vertical). Para evitar esto se colocan los pulsos de pre y pos-ecualización, cuya función es llevar la carga del capacitor a valores fijos antes y después del pulso vertical. El hecho de hacerlo también después evita disparos erráticos del sistema.

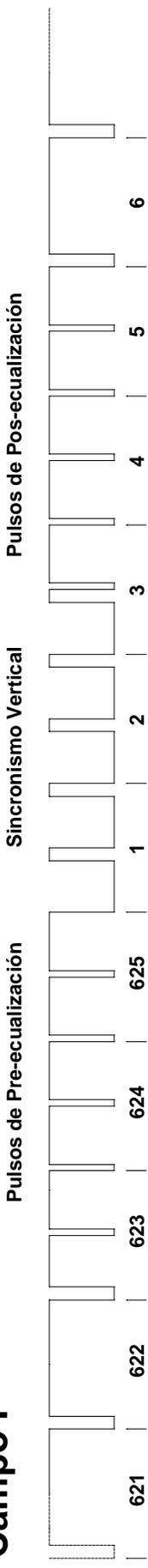
Los pulsos de pre y pos-ecualización tienen una frecuencia igual al doble de la horizontal y su duración es la mitad de la del H Sync. Se aplican durante un tiempo igual a 2½ líneas horizontales.

Finalmente, los pulsos positivos que aparecen dentro del pulso de sincronismo vertical se denominan “Serrated Pulses”, y su función es mantener enganchado al oscilador horizontal durante este período. Su duración es igual al H Sync.

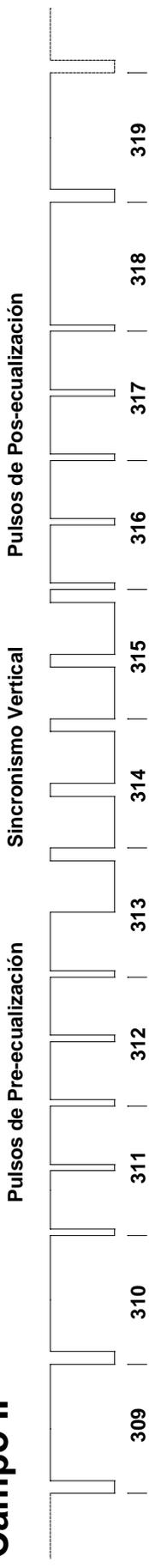
Métodos de barrido en TV - sistema PAL-N

Barrido entrelazado

Campo I

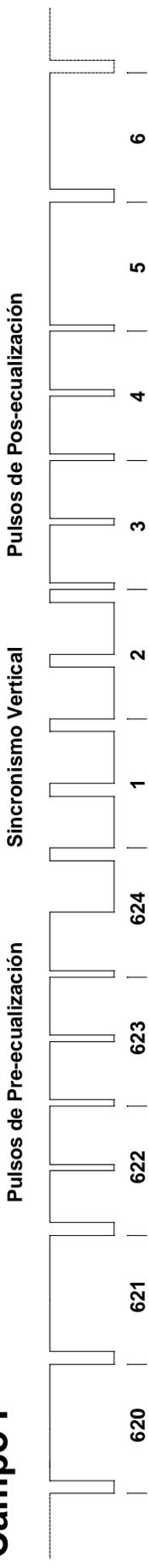


Campo II

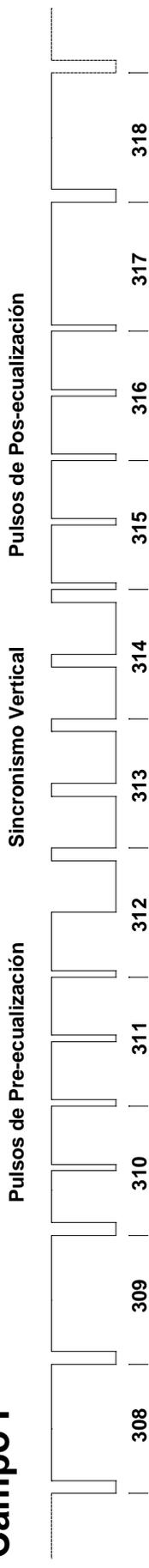


Barrido no entrelazado

Campo I



Campo I'



Sistemas de color

La televisión cromática llegó varios años después de inventado el sistema “blanco y negro”, y por lo tanto tuvo que “adaptarse” al sistema en uso. Fue necesario desarrollar sistemas de color compatibles con los receptores monocromáticos del momento y además los nuevos aparatos reproductores de color debían seguir captando las señales de origen “blanco y negro”. Fue así que aparecieron y perduraron tres sistemas de color: NTSC, PAL y SECAM. Trataremos del PAL que es el sistema que nos concierne.

El sistema PAL se basa en enviar la información de color modulando la fase de una portadora con dos señales en cuadratura (a 90°). Veamos el proceso paso a paso.

Dijimos que toda la información de color de un objeto está en sus componentes Rojo, Verde y Azul. Esos son los colores que captan nuestros ojos, y si logramos transmitirlos, habremos transmitido el color real del objeto. Lo que no dijimos es que nuestros ojos tienen distinta sensibilidad a cada uno de los colores primarios. Son muy sensibles al verde, menos al rojo y mucho menos al azul. Si lo ponemos en porcentajes relativos de sensibilidad:

VERDE: 59%
ROJO: 30%
AZUL: 11%

O sea que la luminosidad de un objeto (LUMINANCIA o Y), entendida como la suma total de luz que emite, puede representarse así:

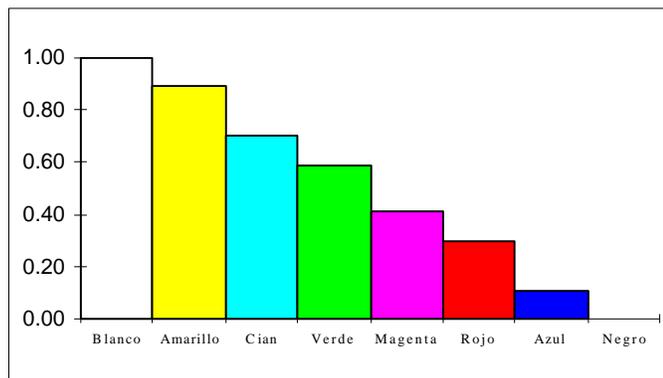
$$Y = 0.30\text{Rojo} + 0.59\text{Verde} + 0.11\text{Azul}$$

De ahora en más denominaremos R al Rojo, G al Verde y B al Azul (tomados del Inglés), por lo tanto:

$$Y = 0.30R + 0.59G + 0.11B$$

Esta información de luminosidad (Y) es la que nos da el “brillo” de cada combinación de colores en la pantalla, y es la responsable de la imagen monocromática. Veamos gráficamente el brillo relativo de los colores primarios y secundarios:

| Color: | R | G | B | Y |
|----------|---|---|---|------|
| Blanco | 1 | 1 | 1 | 1.00 |
| Amarillo | 1 | 1 | 0 | 0.89 |
| Cian | 0 | 1 | 1 | 0.70 |
| Verde | 0 | 1 | 0 | 0.59 |
| Magenta | 1 | 0 | 1 | 0.41 |
| Rojo | 1 | 0 | 0 | 0.30 |
| Azul | 0 | 0 | 1 | 0.11 |
| Negro | 0 | 0 | 0 | 0.00 |



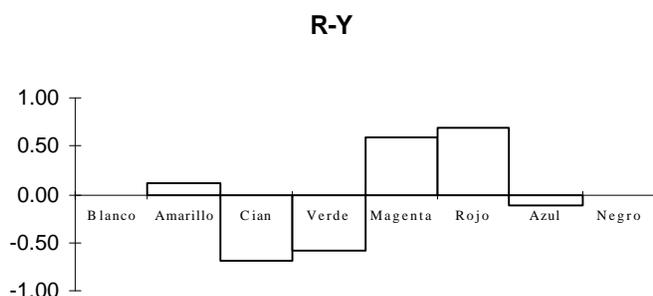
La luminancia (Y) es uno de los componentes fundamentales de la señal de video, y como vemos, tiene información sobre la suma total de los componentes RGB de la imagen. De este modo, bastaría con enviar la información de dos de los componentes RGB junto con Y, ya que en el receptor se podría recomponer el componente faltante mediante operaciones sencillas. Como el componente G (Verde) es predominante en Y, este será el que no se enviará por separado, y se reconstruirá en el receptor.

Recordemos que toda esta complicación surge porque el enviar RGB como tales ocupa mucho ancho de banda, lo cual no está permitido (esto es porque cuando llegó la televisión a color ya se habían asignado los anchos de banda correspondientes a transmisiones monocromáticas).

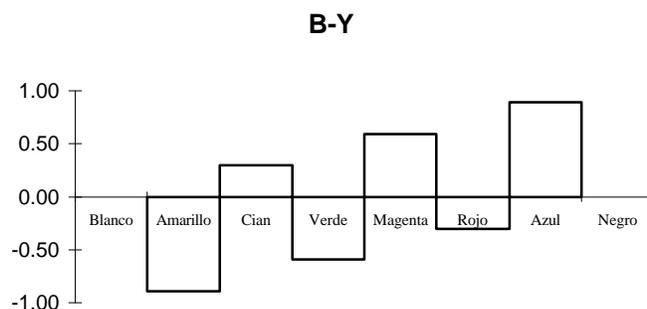
Entonces enviaremos Y, R y B. Pero R y B tampoco serán enviados como tales, sino que se enviará su diferencia respecto a Y: **R-Y** y **B-Y**

Veamos como se ven estas señales:

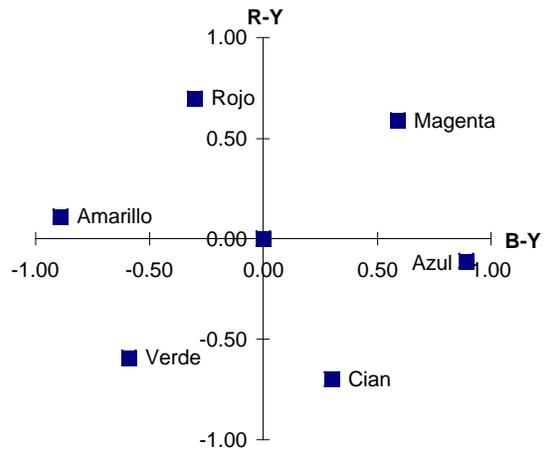
| Color: | R | G | B | R-Y |
|----------|---|---|---|-------|
| Blanco | 1 | 1 | 1 | 0.00 |
| Amarillo | 1 | 1 | 0 | 0.11 |
| Cian | 0 | 1 | 1 | -0.70 |
| Verde | 0 | 1 | 0 | -0.59 |
| Magenta | 1 | 0 | 1 | 0.59 |
| Rojo | 1 | 0 | 0 | 0.70 |
| Azul | 0 | 0 | 1 | -0.11 |
| Negro | 0 | 0 | 0 | 0.00 |



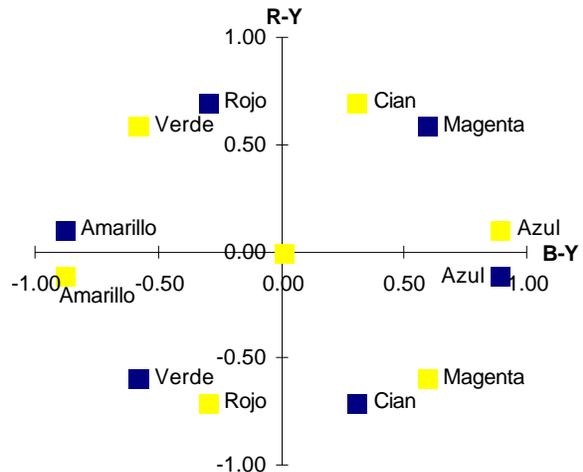
| Color: | R | G | B | B-Y |
|----------|---|---|---|-------|
| Blanco | 1 | 1 | 1 | 0.00 |
| Amarillo | 1 | 1 | 0 | -0.89 |
| Cian | 0 | 1 | 1 | 0.30 |
| Verde | 0 | 1 | 0 | -0.59 |
| Magenta | 1 | 0 | 1 | 0.59 |
| Rojo | 1 | 0 | 0 | -0.30 |
| Azul | 0 | 0 | 1 | 0.89 |
| Negro | 0 | 0 | 0 | 0.00 |



Estas señales, denominadas “Señales de Diferencia de Color” serán las que modularán en fase a la subportadora de color. La señal B-Y modulará a SC a 0° , en tanto que R-Y lo hará a 90° . De este modo podemos obtener un diagrama de fase de los distintos colores, como se ve en la figura adjunta.



A su vez, y esto es una característica del sistema PAL, la portadora en cuadratura (90°) rotará 180° de una línea a la otra, es decir que si en una línea horizontal R-Y modula a SC a 90° , en la línea siguiente lo hará a 270° , y así sucesivamente. De aquí obtiene su nombre el sistema: PAL = Phase Alternation by Line (Alternancia de Fase por Línea).



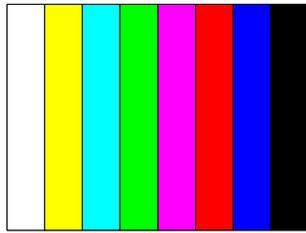
Una vez modulada la subportadora de color pasa a llamarse señal de “Crominancia” o señal de Color (C). Sumando la señal de Luminancia (Y) más la de Crominancia (C) obtenemos finalmente la señal buscada, la SEÑAL DE VIDEO COMPUESTO.

Construcción de un generador de patrones

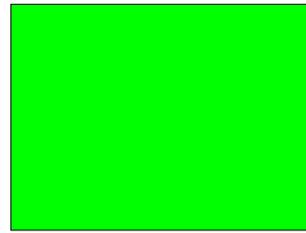
Un generador de patrones no solo debe entregar los componentes RGB de la señal, sino que debe generar video compuesto, con todo lo que esto implica: generar sincronismos, obtener la señal de luminancia, generar una subportadora de color y modularla, etc.

Lo primero que debe definirse es la cantidad y tipo de patrones que deberá generar el equipo, ya que esto determinará las características, y por tanto la complejidad, del sistema a desarrollar.

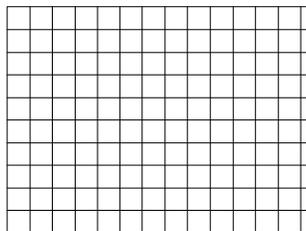
En nuestro caso el equipo será capaz de generar cuatro patrones básicos:



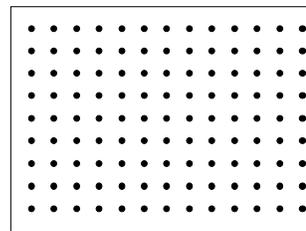
Barras



Raster

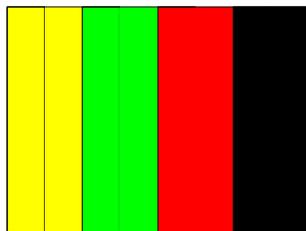


Cross-hatch

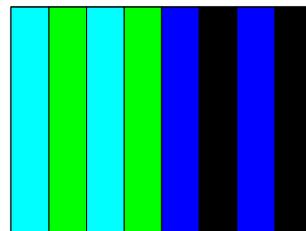


Puntos

Además permitirá control independiente de los tres colores R, G y B, así como de las señales de luminancia (Y) y crominancia (C). De este modo se amplía la cantidad de patrones que pueden ser generados, ya que el Raster se podrá hacer con cualquiera de los 8 colores, las barras podrán ser monocromáticas o adoptar diferentes combinaciones de color (ver ejemplos adjuntos), etc.



Barras: B off



Barras: R off

Como control adicional se permite la supresión del BURST de color, herramienta útil en la detección de fallas relacionadas con los circuitos de proceso de color.

Para seleccionar cual de los cuatro patrones básicos generará el equipo se utilizan dos llaves (S4 y S5), cuya combinación determinará el patrón según la siguiente tabla:

| | | |
|----------------|--------------------|---------------|
| Llaves: | S4 OFF | S4 ON |
| S5 OFF | BARRAS | RASTER |
| S5 ON | CROSS-HATCH | PUNTOS |

Los estados OFF (o “cero”) y ON (o “uno”) se refieren a que el punto medio de la llave se conecta a masa (0V) o a VCC (5V) respectivamente.

Una vez definido que va a hacer el equipo veamos como implementarlo.

Generación de sincronismos y patrones

La generación de la base de tiempo, los sincronismos y los cuatro patrones básicos estarán a cargo de un microcontrolador (PIC16F84-10), por lo que toda esta sección consistirá en desarrollar el programa (Software) adecuado. Al terminar esta etapa, el microcontrolador deberá hacer lo siguiente:

- Generar una base de tiempos estable, de donde obtener todos los tiempos requeridos por los sincronismos.
- Generar en uno de sus terminales, el correspondiente al **Bit 0 del PORTB**, todos los **sincronismos** requeridos por la norma de televisión adoptada (N), sin agregar video a esta señal (sincronismos puros).
- Generar en tres terminales las señales R, G y B, que correspondan con el patrón que deba mostrarse a la salida. Estos terminales no tendrán sincronismos (video puro). La designación de terminales es la siguiente:

PORTB (2) = B (Azul)

PORTB (3) = R (Rojo)

PORTB (4) = G (Verde)

(Entre paréntesis se indica el Bit correspondiente del PORTB)

- Aceptar en dos de sus terminales, configurados como entradas, las órdenes provenientes de las llaves S4 y S5, de modo de poder seleccionar el patrón a generar. Estas entradas corresponden a dos Bits del PORTA, los siguientes:

PORTA (2) = S4

PORTA (3) = S5

Una vez claros los objetivos, veamos como los lleva a cabo el programa.

Básicamente el mismo se compone de cuatro bloques independientes de generación de señal, realizándose en cada uno todo lo necesario para la generación de una imagen completa. En la sección de diagramas se incluye un diagrama de flujo del programa, el cual ayudará a seguir la explicación.

Luego de una primera instancia de definición de variables e inicialización de las mismas, se pasa a leer el estado de las llaves S4 y S5. Según que combinación se encuentre activada en ese momento, el programa se dirigirá a uno de los cuatro bloques de video mencionados, donde se generará uno de los patrones básicos.

En cada uno de estos bloques se comienza por generar los pulsos de pre-ecualización, luego el sincronismo vertical con sus correspondientes “Serrated Pulses”, seguido de los pulsos de pos-ecualización.

A continuación se realiza la selección de campo par/impar. Esto es muy importante, ya que, como trabajamos con barrido entrelazado, en uno de los campos la primera línea horizontal luego del sincronismo vertical es completa, mientras que en el otro campo debe ser solo media línea (recordar que comienza en medio de la pantalla). Si no se hiciera esto la imagen aparecería temblorosa en el sector superior.

Es de destacar que en dos de los patrones (Cross-hatch y Puntos) se trabaja con barrido no entrelazado, para evitar el fenómeno de temblor vertical (o “flicker”) de las líneas fijas. En estos casos la primer línea horizontal es siempre entera, y para compensar esto se quita un pulso de pre-equalización (según ya se mostró en los diagramas de la señal de video, métodos de barrido).

Luego de esto se hacen 3 o 4 líneas horizontales sin video (según el campo), pero respetando correctamente los tiempos de sincronismo.

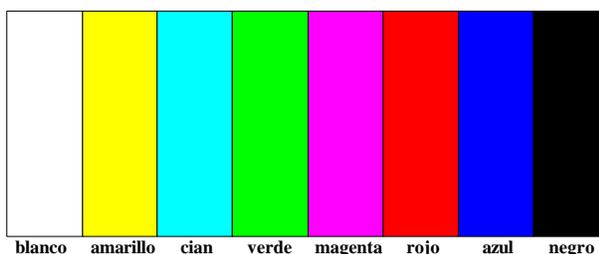
Ahora es el momento en que entran en actividad la líneas R, G y B. Luego de generar el sincronismo horizontal y respetar el tiempo de back porch, en las líneas RGB aparece la información que corresponda a la señal mostrada. ¿Cómo es esto?. Veamos un ejemplo.

Supongamos que se está generando una señal de barras. Son ocho barras, por lo tanto debemos dividir el tiempo útil de video en ocho intervalos iguales.

Antes de seguir, cabe aclarar que el tiempo útil de video es aquel tiempo en que efectivamente la información generada se ve en pantalla. Recordemos que en PAL-N cada línea horizontal dura un tiempo total de 64 μ seg., donde se incluyen 4.8 μ seg. de H Sync, 1.9 μ seg. de Front Porch y 5 μ seg. de Back Porch. Por lo tanto solo nos quedan 52.3 μ seg. para mostrar video, y ese es nuestro tiempo útil.

Volviendo a la generación de barras, ya tenemos los ocho intervalos. Veamos como debemos enviar las señales R (Rojo), G (Verde) y B (Azul) en cada uno de ellos:

| | | | | | | | | |
|--------------|----------|----------|----------|----------|----------|----------|----------|----------|
| Azul | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 |
| Rojo | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| Verde | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |



Como verán, esta ya es una figura conocida. Se utilizó para describir lo que debía hacer un generador de barras sencillo con salida RGB, y eso es justamente lo que estamos haciendo.

Veamos ahora como se genera el Raster. Este es aún más sencillo: se envía todo el tiempo un nivel alto en las tres líneas de RGB. Pero, si RGB están los tres activos, solo generaremos Raster blanco. Es cierto. La selección de color se realiza controlando R, G o B externamente al microcontrolador.

Para la generación de líneas y puntos se requieren rutinas algo más elaboradas, ya que no solo hay que contar tiempos en sentido horizontal sino también líneas horizontales, a fin de fijar exactamente la separación vertical de las líneas o puntos. Pero igualmente se trata de contar, ahora utilizando dos variables.

Con respecto a las líneas RGB, todas se ponen activas en el momento de dibujar líneas o puntos (o sea, son blancos).

Si se analiza con detenimiento el programa se verá que en cada uno de los bloques de video la generación de líneas horizontales y sus correspondientes señales de video se realiza en tres bloques, básicamente iguales. La razón de esto es sencilla. En cada pasada dentro de un bloque de video se barre un campo completo, o sea 312.5 líneas horizontales. Para hacerlo con precisión se debe llevar la cuenta de las líneas generadas en algún registro. Como solo se dispone de registros de 8 bits con signo (o sea que el número máximo que alcanzan es 127) se precisa cargar tres veces el registro a fin de llegar al número de líneas requerido.

Para finalizar con el bloque de video, luego de completado cada campo se evalúa el teclado (S4 y S5). Si no han cambiado, se continúa en el mismo bloque; si hay algún cambio, se vuelve a la rutina inicial de lectura de teclado, y el programa se dirige al bloque de video que le indique el estado de S4 y S5.

Y eso es todo. Puede quedar una duda... ¿cómo calculo tiempos dentro de un programa?.

El uso de un microcontrolador (PIC16F84-10) facilita mucho esta tarea, ya que basta con “contar ciclos de reloj” para obtener todos los tiempos correctos. Utilizando un cristal de 10 MHz, y sabiendo que cada ciclo de instrucción son 4 ciclos de reloj, obtenemos el tiempo de un ciclo de instrucción:

$$T_{osc} = 1/f_{osc}$$

$$T_{ins} = T_{osc} \times 4$$

$$T_{ins} = 1/10 \text{ MHz} \times 4 = 0.4 \text{ } \mu\text{seg.}$$

Si cada ciclo de instrucción dura 0.4 μ seg., entonces para obtener el pulso de sincronismo horizontal basta con contar 12 ciclos:

$$12 \times 0.4 \text{ } \mu\text{seg.} = 4.8 \text{ } \mu\text{seg.}$$

Del mismo modo obtenemos que la duración de una línea horizontal completa es de 160 ciclos de instrucción:

$$160 \times 0.4 \text{ } \mu\text{seg.} = 64 \text{ } \mu\text{seg.}$$

Básicamente, esto es lo que hace el programa. Cuenta instrucciones y pone a nivel alto o bajo, según corresponda, el Bit 0 del PORTB. Se estableció que durante el pulso de sincronismo (H o V) este bit estará a nivel bajo (0V) y el resto del tiempo a nivel alto (5V).

Generación de Video Compuesto

Como ya habíamos visto, no basta generar RGB para tener un generador de patrones útil, que pueda ser conectado a un receptor de televisión o a un videograbador. Debemos combinar esta señal RGB con la señal de sincronismos y con ambas generar Video Compuesto, señal que sí puede inyectarse a los equipos mencionados.

Ya vimos todo el proceso requerido para obtener Video Compuesto a partir de RGB, así que no lo repetiremos. Menos aún, teniendo en cuenta que hay un circuito integrado diseñado por Motorola® que cumple con las siguientes especificaciones:

- Posee cuatro entradas de señal: Sincronismo, R, G y B
- A partir de RGB genera la señal de luminancia (Y)
- Posee un circuito oscilador, que con el cristal adecuado genera la Subportadora de Color
- Genera las señales B-Y y R-Y, con la alternancia de fase requerida por el sistema PAL
- A partir de B-Y y R-Y genera la señal de crominancia (C)
- Mezcla Y con C para obtener Video Compuesto

Como ven, un solo integrado hace exactamente lo que necesitamos. Y además, requiere exactamente las señales que ya hemos generado con el microcontrolador.

Este integrado es el MC1377, RGB ENCODER, y bastan muy pocos componentes externos para realizar el circuito completo. De hecho, se ha utilizado la configuración sugerida en sus hojas de datos, con algunas modificaciones empíricas a fin de mejorar aún más su rendimiento.

Se ha utilizado un cristal de 3.582056 MHz para que el equipo genere señal en el sistema PAL-N. Nada impide reemplazar este cristal por uno de 4.43 MHz y hacer un pequeño ajuste al TRIMMER CV1 para obtener una señal en los sistemas PAL-B/G/I, de uso actualmente en Europa.

En esta etapa es que se realizan los controles de RGB, Y, C y BURST. Básicamente se han colocado llaves que derivan la señal a masa, directamente (RGB) o a través de un capacitor (Y, C). En el caso del BURST, para anularlo, la llave (S8) saca del circuito al capacitor C04, el cual es responsable de generar el tiempo de permanencia del mismo.

Veamos ahora un resumen de las llaves de comando del equipo y sus funciones:

| Llave | Función |
|-------|--------------|
| S1 | G ON/OFF |
| S2 | R ON/OFF |
| S3 | B ON/OFF |
| S4 | PROGRAMA |
| S5 | PROGRAMA |
| S6 | Y ON/OFF |
| S7 | C ON/OFF |
| S8 | BURST ON/OFF |
| S9 | ALIMENTACION |

Una vez obtenida la señal de Video Compuesto se ajusta el nivel y la impedancia de la misma pasando por un circuito buffer, conformado por Q1, R14 y R15.

Con esto concluye la generación de señal, y prácticamente la descripción de nuestro circuito. Solo queda por mencionar que ambos integrados se alimentan con tensiones diferentes, por lo que se puede observar una alimentación principal de 12V (8 pilas alcalinas tipo AA, se ha pensado en un equipo portátil), destinada al sector de video (U2 y Q1), y una alimentación secundaria, 5V, derivada de la primera, destinada al microcontrolador (U1).

Realización práctica del generador de video

A continuación se dan todos los diagramas y dibujos necesarios para realizar la construcción del equipo. Se ha incluido el diseño del circuito impreso a escala real, por lo que basta imprimirlo en transparencia y pasarlo a una placa sensibilizada para obtener el impreso real. Prestar atención al hecho de que impreso está invertido, a fin de que coincida el diagrama de ubicación de componentes. En el impreso final, el texto “Generador de video” debe quedar “al derecho”.

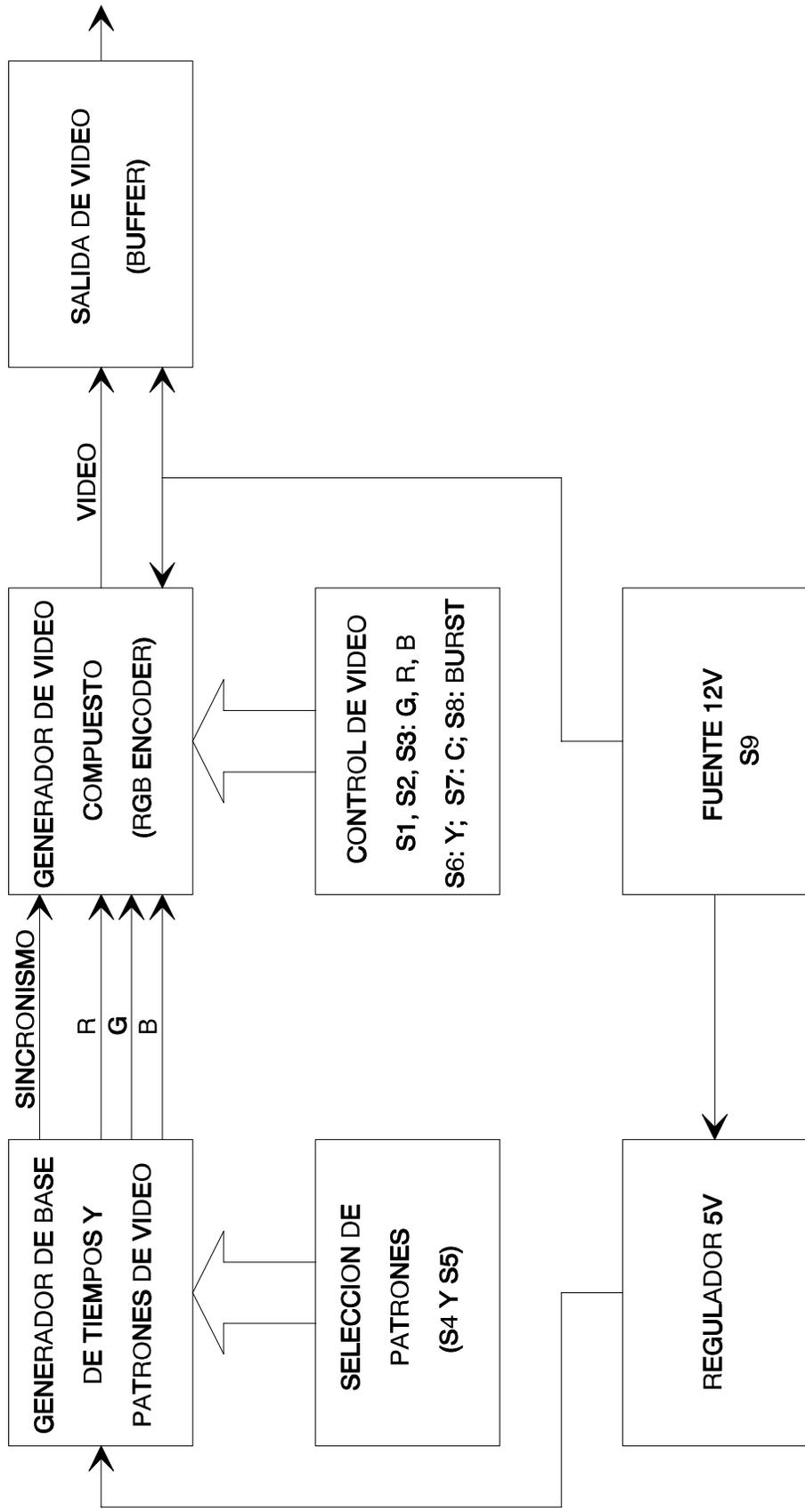
Con respecto al programa, se encuentra en su totalidad en las páginas finales. Basta copiarlo en un editor de texto, ensamblarlo y cargarlo en el PIC, utilizando la herramientas provistas por Microchip® o mediante el método que habitualmente se utilice.

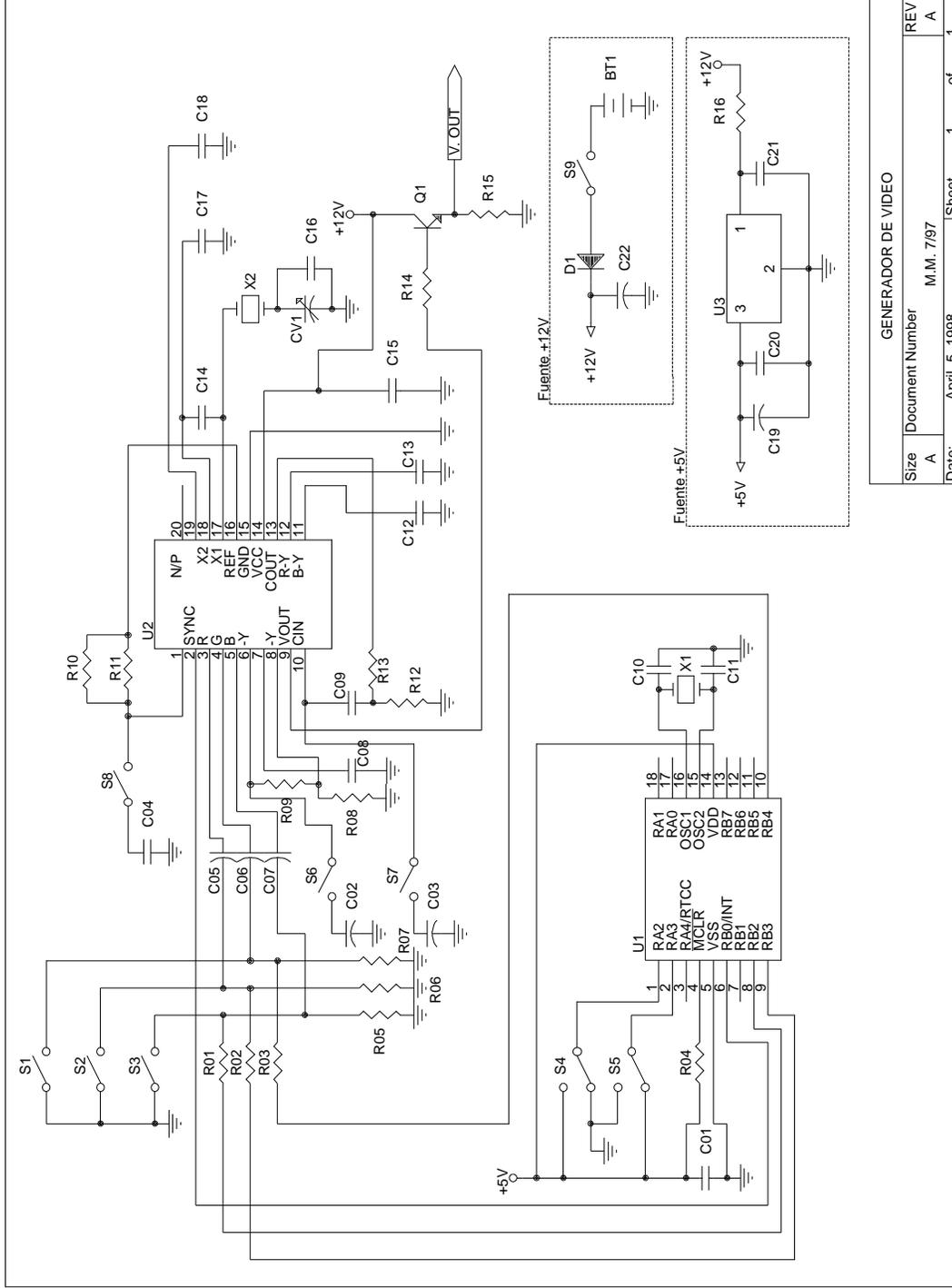
NOTA IMPORTANTE: En el momento de volcar el programa en el PIC no olvidar poner la opción de operación con cristal (XT). De otro modo, el cristal no oscilará.

Y eso es todo. Si todos los componentes se han ubicado correctamente el equipo funcionará según lo esperado desde el comienzo. El único ajuste que puede realizarse es mover CV1 a fin de mejorar la reproducción de color, lo cual es muy sencillo.

Espero que este proyecto sea de utilidad y quedo a la espera de comentarios, sugerencias y posibles mejoras, así como preguntas e inquietudes al respecto.

Marcelo F. Maggi - Abril, 1998
mmaggi@hotmail.com

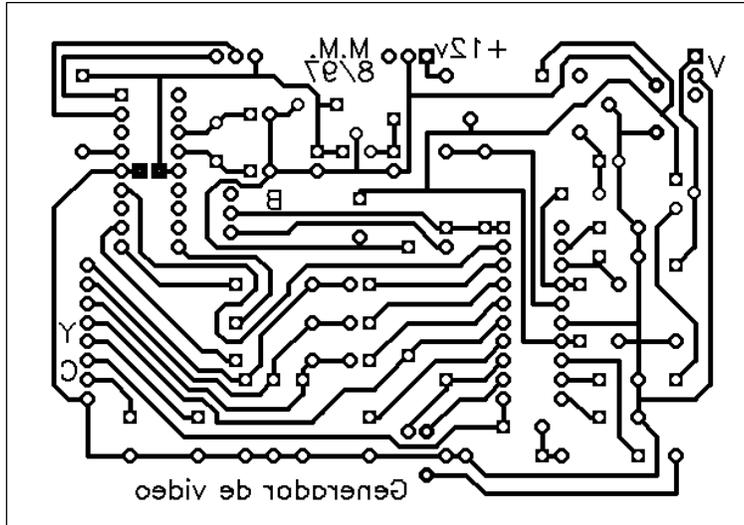




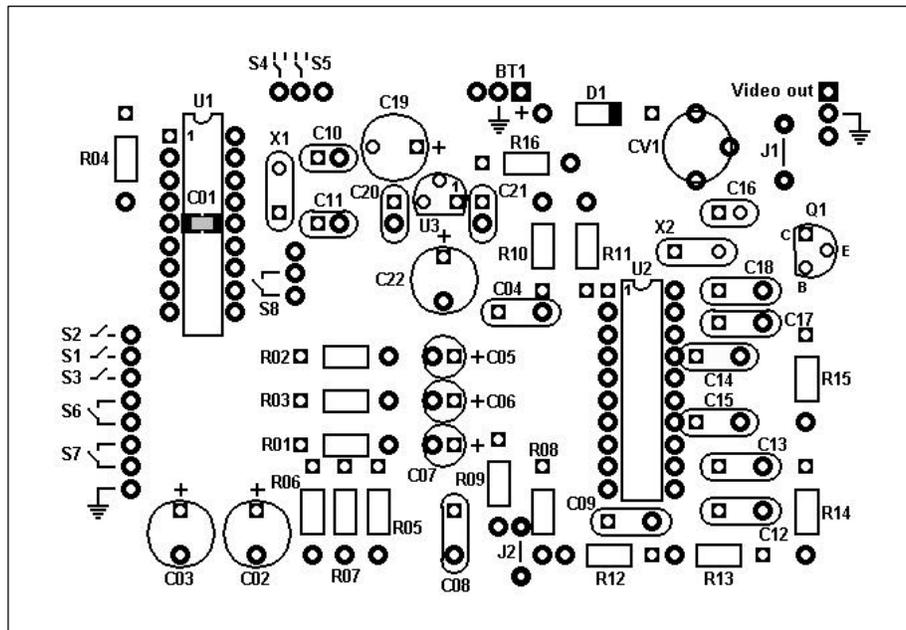
| | | |
|--------------------|-----------------|--------------|
| GENERADOR DE VIDEO | | |
| Size | Document Number | REV |
| A | M.M. 7/97 | A |
| Date: | April 5, 1998 | Sheet 1 of 1 |

| | |
|-----|----------|
| R01 | 3K9 |
| R02 | 3K9 |
| R03 | 3K9 |
| R04 | 1K |
| R05 | 1K |
| R06 | 1K |
| R07 | 1K |
| R08 | 1K |
| R09 | 1K |
| R10 | 68K |
| R11 | 82K |
| R12 | 10K |
| R13 | 2K2 |
| R14 | 4K7 |
| R15 | 2K7 |
| R16 | 100 |
| C01 | 0.1μ |
| C02 | 100μ/16V |
| C03 | 100μ/16V |
| C04 | 1500p |
| C05 | 10μ/25V |
| C06 | 10μ/25V |
| C07 | 10μ/25V |
| C08 | .02μ |
| C09 | .01μ |
| C10 | 15p |
| C11 | 15p |
| C12 | 0.1μ |

| | |
|-----|------------------------|
| C13 | 0.1μ |
| C14 | 220p |
| C15 | 0.1μ |
| C16 | 18p |
| C17 | 150p |
| C18 | .02μ |
| C19 | 100μ/16V |
| C20 | 0.1μ |
| C21 | 0.1μ |
| C22 | 100μ/16V |
| CV1 | TRIMMER 5-45p |
| D1 | 1N4007 |
| Q1 | BF494C |
| U1 | PIC16F84-10 |
| U2 | MC1377 |
| U3 | LM78L05 |
| X1 | 10.000MHz |
| X2 | 3.582056MHz |
| S1 | LLAVE 2 POSICIONES |
| S2 | LLAVE 2 POSICIONES |
| S3 | LLAVE 2 POSICIONES |
| S4 | LLAVE 2 POSICIONES |
| S5 | LLAVE 2 POSICIONES |
| S6 | LLAVE 2 POSICIONES |
| S7 | LLAVE 2 POSICIONES |
| S8 | LLAVE 2 POSICIONES |
| S9 | LLAVE 2 POSICIONES |
| BT1 | 8 x AA PILAS ALCALINAS |



5 cm



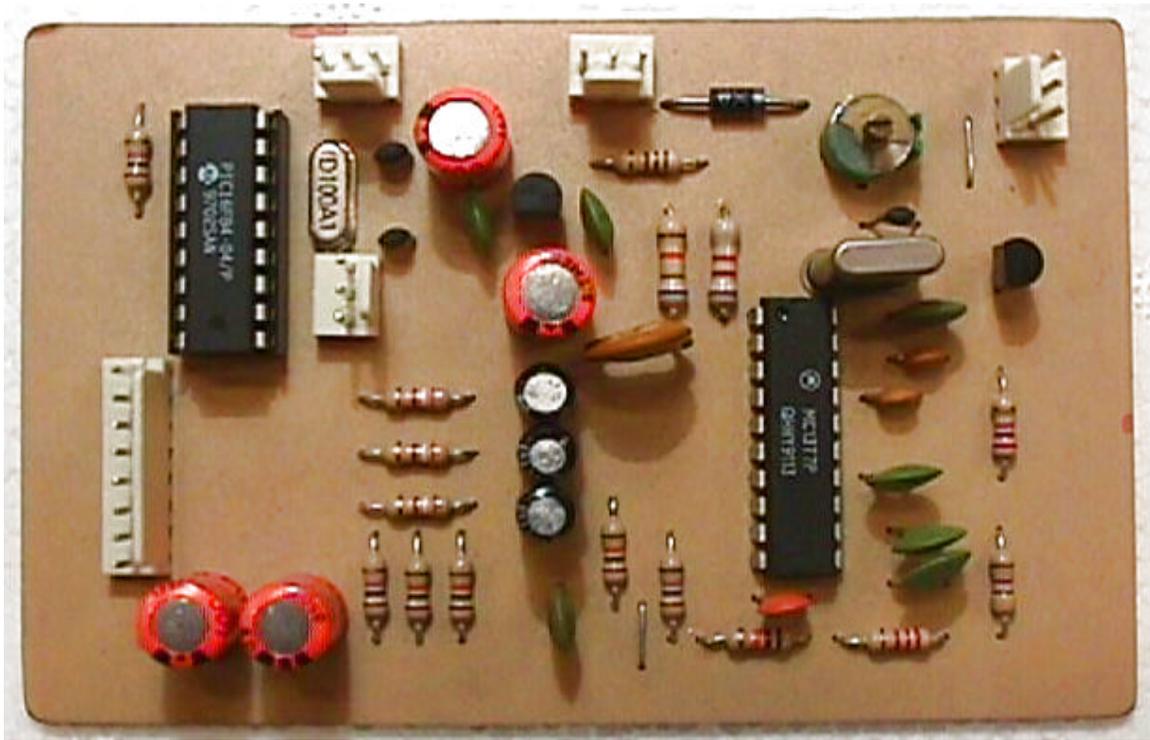


Diagrama de flujo general

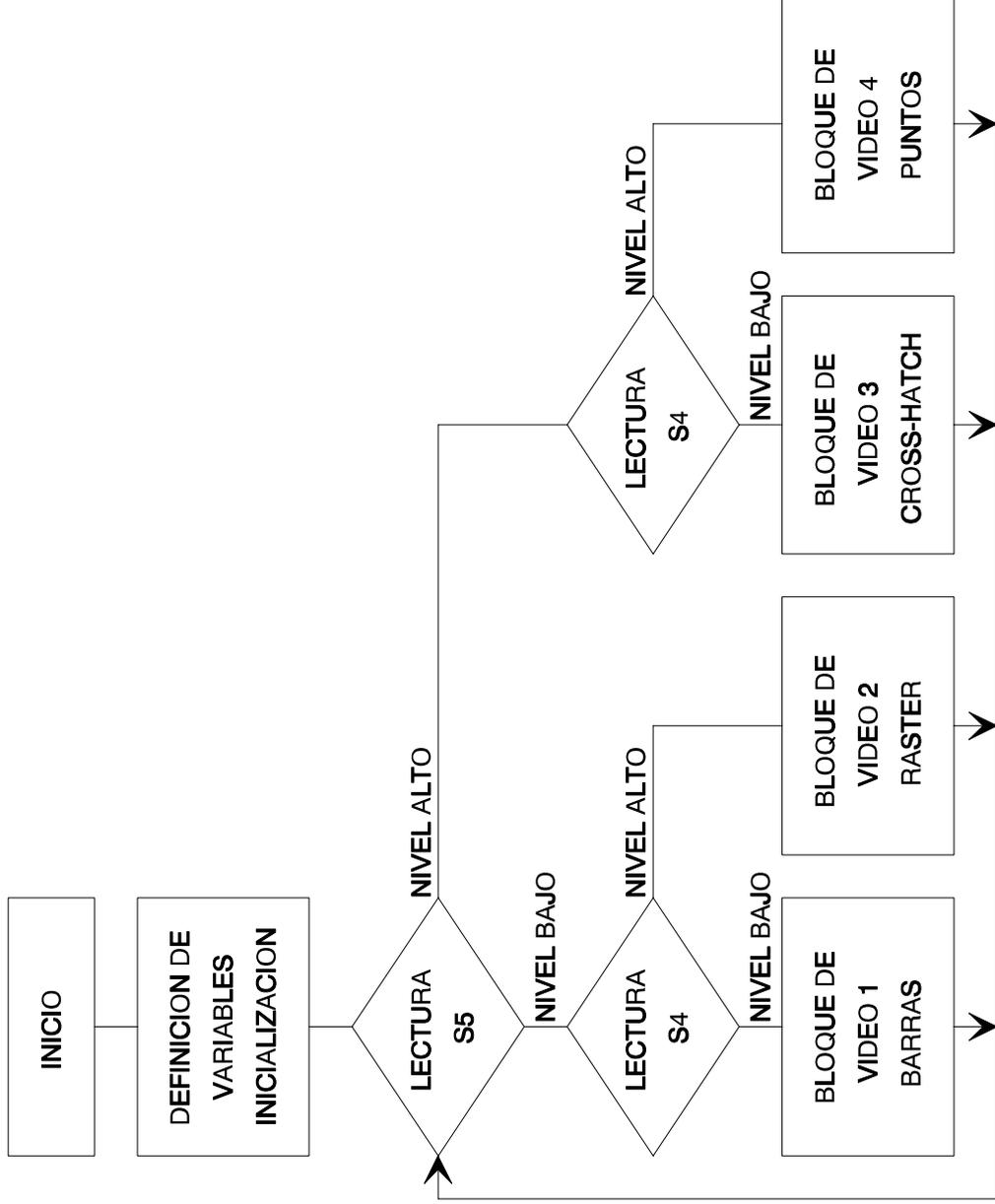
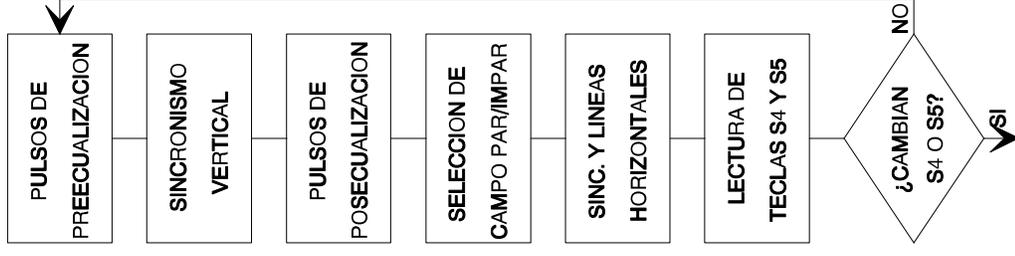


Diagrama de flujo del bloque de video



```

; ***** GENERADOR DE PATRONES PARA VIDEO *****
; VERSION 2.01
; GEN201.ASM
; (C) M. MAGGI - 30/08/1997

        list      p=16f84

;DEFINICION DE PUERTOS:
;PORTB(0): SYNC
;PORTB(2): AZUL
;PORTB(3): ROJO
;PORTB(4): VERDE
;NO USAR EL BIT 1 DEL PORTB
;PARA LOS PATRONES DE RASTER Y BARRAS EL VIDEO ES ENTRELAZADO
;LOS PUNTOS Y EL CROSSHATCH SE HACEN CON VIDEO NO ENTRELAZADO PARA EVITAR EL
;"FLICKER"

CBLOCK  0X0C                      ;VARIABLES
        DURHOR, CANTHB1, CANTHB2, BLKLIN, CANTPRE, DUREQU, CANTVER, DURVER, CANTPOS
        TIEMPO, FIELD, CARRY
        WHITE, YELLOW, CYAN, GREEN, MAGEN, RED, BLUE, BLACK, CANTLIN
ENDC

PORTA   EQU      5
TRISA   EQU      85H
PORTB   EQU      6
TRISB   EQU      86H
STATUS  EQU      3
RP0     EQU      5
BLANCO  EQU      B'00011101'
AMARIL  EQU      B'00011001'
CYANO   EQU      B'00010101'
VERDE   EQU      B'00010001'
MAGENT  EQU      B'00001101'
ROJO    EQU      B'00001001'
AZUL    EQU      B'00000101'
NEGRO   EQU      B'00000001'
;
        CLRF     PORTA           ;TODOS LOS BITS EN 0
        CLRF     PORTB           ;TODOS LOS BITS EN 0
        BSF      STATUS,RP0      ;SELECCIONA BANCO DE REGISTROS 1
        MOVLW    B'11111111'
        MOVWF    TRISA           ;TODOS LOS BITS DEL PUERTO A COMO ENTRADAS
        CLRF     TRISB^80H       ;TODOS LOS BITS DEL PUERTO B COMO SALIDA
        BCF      STATUS,RP0      ;SELECCIONA BANCO DE REGISTROS 0
;
        MOVLW    0
        MOVWF    CARRY           ;VARIABLE CONTROLAR EL ESTADO DEL CARRY
        RRF      CARRY           ;CARRY FLAG A "0"
        MOVLW    B'10101010'
        MOVWF    FIELD          ;CONTROL DEL CAMPO

LECTURA  BTFSS   PORTA,3        ;SE LEE EL TECLADO
          GOTO    LECT1         ;SE USAN LOS BITS 2 Y 3 DEL PUERTO A
          BTFSC   PORTA,2        ;FUNCION:          BIT3     BIT2
          GOTO    INICIO3        ;BARRAS           0       0
          GOTO    INICIO2        ;RASTER           0       1
LECT1     BTFSC   PORTA,2        ;CROSSHATCH       1       0
          GOTO    INICIO1        ;PUNTOS           1       1

;***** BARRAS DE COLOR *****

INICIO   RRF      FIELD          ;CARRY PASA AL BIT 7 DE FIELD, BIT 0 AL CARRY
          MOVLW   D'3'          ;LINEAS SIN VIDEO LUEGO DE LA POSECUALIZACION

```

```

        BTFSS    FIELD,0          ;SI ES EL CAMPO 1 SE HACEN SOLO 3 LINEAS
        MOVLW   D'4'              ;4 LINEAS EN EL CAMPO 2
        MOVWF   BLKLIN
        MOVLW   D'99'
        MOVWF   CANTHB1          ;CANTIDAD DE LINEAS HORIZONTALES EN UN BLOQUE
        MOVLW   D'3'
        MOVWF   CANTHB2          ;CANTIDAD DE BLOQUES (3)
        MOVLW   5
        MOVWF   CANTPRE          ;PULSOS DE PREECUALIZACION
        MOVLW   5
        MOVWF   CANTVER          ;PULSOS DE SINCRONISMO VERTICAL
        MOVLW   5
        MOVWF   CANTPOS          ;PULSOS DE POSECUALIZACION

PREEQU  BCF     PORTB,0          ;DURACION: 2,6µS ABAJO
        MOVLW   D'23'
        MOVWF   DUREQU
        NOP
        NOP
        NOP
        BSF     PORTB,0
LOOP1   DECFSZ  DUREQU          ;SE COMPLETAN LOS 32µS ARRIBA
        GOTO    LOOP1
        NOP
        NOP
        DECFSZ  CANTPRE
        GOTO    PREEQU
        NOP
VERT    BCF     PORTB,0
        MOVLW   D'22'
        MOVWF   DURVER
LOOP2   DECFSZ  DURVER
        GOTO    LOOP2
        BSF     PORTB,0          ;DURACION: 4.8µS ARRIBA ("SERRATED PULSES")
        MOVLW   2
        MOVWF   TIEMPO
TIME    DECFSZ  TIEMPO
        GOTO    TIME
        NOP
        DECFSZ  CANTVER
        GOTO    VERT
        NOP
POSEQU  BCF     PORTB,0
        MOVLW   D'23'
        MOVWF   DUREQU
        NOP
        NOP
        NOP
        BSF     PORTB,0
LOOP3   DECFSZ  DUREQU
        GOTO    LOOP3
        NOP
        NOP
        DECFSZ  CANTPOS
        GOTO    POSEQU
        NOP

;SE EMPIEZAN A BARRER LAS LINEAS HORIZONTALES
;LA PRIMERA LINEA ES COMPLETA EN EL CAMPO 1, EN TANTO QUE ES SOLO MEDIA LINEA
;EN EL CAMPO 2, Y NO COMIENZA CON UN PULSO DE SINCRONISMO

        RLF     PORTB           ;1 O 1/2 LINEA H SEGUN EL CAMPO
        NOP
        NOP                     ;SE PASA EL CARRY AL BIT 0 DEL PUERTO B
        NOP                     ;CAMPO 1: 1 LINEA Y PULSO DE SINC (CARRY=0)

```

```

NOP                                ;CAMPO 2: 1/2 LINEA SIN PULSO DE SINC (CARRY=1)
NOP
NOP
NOP
NOP
MOVLW    D'21'                    ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
BTFSS    PORTB,0                  ;SI HAY H SYNC (CAMPO 1) SE AGREGA MAS TIEMPO
ADDLW    D'27'                    ;TIEMPO PARA 1 H (160 CICLOS TOTAL)
MOVWF    DURHOR
BSF      PORTB,0
BTFSS    FIELD,0
GOTO     NEXT                    ;SE PIERDE 1 CICLO MAS (SOLO 1/2 H)
NEXT     BCF      PORTB,1
NOP
LOOP     DECFSZ   DURHOR
GOTO     LOOP
NOP

```

;SE HACEN 3 O 4 LINEAS EN BLANCO PARA CUMPLIR CON LAS 625 LINEAS DE LA NORMA N
;SI ES EL CAMPO 1 SE HACEN SOLO 3 LINEAS, YA QUE ANTES SE HIZO 1 DE MAS

```

HORIZ    BCF      PORTB,0
MOVLW    2
MOVWF    TIEMPO                    ;PIERDO TIEMPO PARA
TIME3    DECFSZ   TIEMPO            ;HACER LOS 4,8µS
GOTO     TIME3
NOP
NOP
MOVLW    D'48'
MOVWF    DURHOR
BSF      PORTB,0                    ;BIT 0 ALTO
LOOPH3   DECFSZ   DURHOR
GOTO     LOOPH3
NOP
DECFSZ   BLKLIN
GOTO     HORIZ
NOP

```

;SE HACEN 3 BLOQUES DE 99 LINEAS HORIZONTALES
;3*(99+1)=300 LINEAS

```

HORIZ1   BCF      PORTB,0
MOVLW    2
MOVWF    TIEMPO                    ;PIERDO TIEMPO PARA
TIME1    DECFSZ   TIEMPO            ;HACER LOS 4,8µS
GOTO     TIME1
NOP
NOP
MOVLW    D'31'
MOVWF    DURHOR
BSF      PORTB,0                    ;BIT 0 ALTO
NOP
NOP
NOP
MOVLW    5
MOVWF    WHITE
MOVWF    YELLOW
MOVWF    CYAN
MOVWF    GREEN
MOVWF    MAGEN
MOVWF    RED
MOVWF    BLUE
MOVWF    BLACK
MOVLW    BLANCO

```

```

MOVWF PORTB
WHITE1 DECFSZ WHITE
GOTO WHITE1
MOVLW AMARIL
MOVWF PORTB
YELLOW1 DECFSZ YELLOW
GOTO YELLOW1
MOVLW CYANO
MOVWF PORTB
CYAN1 DECFSZ CYAN
GOTO CYAN1
MOVLW VERDE
MOVWF PORTB
GREEN1 DECFSZ GREEN
GOTO GREEN1
MOVLW MAGENT
MOVWF PORTB
MAGEN1 DECFSZ MAGEN
GOTO MAGEN1
MOVLW ROJO
MOVWF PORTB
RED1 DECFSZ RED
GOTO RED1
MOVLW AZUL
MOVWF PORTB
BLUE1 DECFSZ BLUE
GOTO BLUE1
MOVLW NEGRO
MOVWF PORTB
BLACK1 DECFSZ BLACK
GOTO BLACK1
NOP
NOP
NOP
NOP
DECFSZ CANTHB1
GOTO HORIZ1
NOP

HORIZ2 BCF PORTB,0 ;BIT 0 BAJO
MOVLW 2
MOVWF TIEMPO ;PIERDO TIEMPO PARA
TIME2 DECFSZ TIEMPO ;HACER LOS 4,8µS
GOTO TIME2
MOVLW D'99'
MOVWF CANTHB1
MOVLW D'31'
MOVWF DURHOR
BSF PORTB,0 ;BIT 0 ALTO
NOP
NOP
NOP
MOVLW 5
MOVWF WHITE
MOVWF YELLOW
MOVWF CYAN
MOVWF GREEN
MOVWF MAGEN
MOVWF RED
MOVWF BLUE
MOVWF BLACK
MOVLW BLANCO
MOVWF PORTB
WHITE2 DECFSZ WHITE

```

```

        GOTO    WHITE2
        MOVLW  AMARIL
        MOVWF  PORTB
YELLO2  DECFSZ  YELLOW
        GOTO    YELLO2
        MOVLW  CYANO
        MOVWF  PORTB
CYAN2   DECFSZ  CYAN
        GOTO    CYAN2
        MOVLW  VERDE
        MOVWF  PORTB
GREEN2  DECFSZ  GREEN
        GOTO    GREEN2
        MOVLW  MAGENT
        MOVWF  PORTB
MAGEN2  DECFSZ  MAGEN
        GOTO    MAGEN2
        MOVLW  ROJO
        MOVWF  PORTB
RED2    DECFSZ  RED
        GOTO    RED2
        MOVLW  AZUL
        MOVWF  PORTB
BLUE2   DECFSZ  BLUE
        GOTO    BLUE2
        MOVLW  NEGRO
        MOVWF  PORTB
BLACK2  DECFSZ  BLACK
        GOTO    BLACK2
        NOP
        NOP
        NOP
        NOP
        DECFSZ  CANTHB2
        GOTO    HORIZ1
        NOP

```

;ESTA ULTIMA LINEA/MEDIA LINEA, LA 305, LA USO PARA CARGAR VARIABLES

```

        BCF    PORTB,0          ;BIT 0 PASA A NIVEL BAJO
        NOP
        NOP                    ;PIERDO TIEMPO PARA
        MOVWLW 0                ;HACER LOS 4,8µS
        BTFSC  FIELD,0          ;NO USAR EL BIT 1 DEL PORTB, BIT 0 = SYNC
        MOVLW 1
        MOVWF  CARRY
        NOP
        MOVLW D'15'
        BTFSC  FIELD,0
        ADDLW D'24'
        MOVWF  DURHOR
        BSF    PORTB,0          ;BIT 0 PASA A NIVEL ALTO
        BTFSS  FIELD,0
        GOTO  NEXT1
NEXT1   NOP
        NOP
LOOPH5  DECFSZ  DURHOR
        GOTO  LOOPH5
        RRF   CARRY              ;CARRY = 1 SI 1 H, CARRY = 0 SI 1/2 H
        BTFSC  PORTA,2
        GOTO  LECTURA
        BTFSC  PORTA,3
        GOTO  LECTURA
        NOP

```

```
    NOP
    NOP
    NOP
    NOP
    GOTO    INICIO
```

```
;***** RASTER *****
```

```
INICIO1  RRF      FIELD
          MOVLW   D'3'
          BTFSS   FIELD,0
          MOVLW   D'4'
          MOVWF   BLKLIN
          MOVLW   D'99'
          MOVWF   CANTHB1
          MOVLW   D'3'
          MOVWF   CANTHB2
          MOVLW   5
          MOVWF   CANTPRE
          MOVLW   5
          MOVWF   CANTVER
          MOVLW   5
          MOVWF   CANTPOS
```

```
APREEQU  BCF      PORTB,0          ;DURACION: 2,6µS ABAJO
          MOVLW   D'23'
          MOVWF   DUREQU
          NOP
          NOP
          NOP
```

```
ALOOP1   BSF      PORTB,0
          DECFSZ  DUREQU          ;SE COMPLETAN LOS 32µS ARRIBA
          GOTO    ALOOP1
          NOP
          NOP
          DECFSZ  CANTPRE
          GOTO    APREEQU
          NOP
```

```
AVERT    BCF      PORTB,0
          MOVLW   D'22'
          MOVWF   DURVER
```

```
ALOOP2   DECFSZ  DURVER
          GOTO    ALOOP2
          BSF      PORTB,0          ;DURACION: 4.8µS ARRIBA ("SERRATED PULSES")
          MOVLW   2
```

```
MOVWF   TIEMPO
ATIME    DECFSZ  TIEMPO
          GOTO    ATIME
          NOP
          DECFSZ  CANTVER
          GOTO    AVERT
          NOP
```

```
APOSEQU  BCF      PORTB,0
          MOVLW   D'23'
          MOVWF   DUREQU
          NOP
          NOP
          NOP
```

```
ALOOP3   BSF      PORTB,0
          DECFSZ  DUREQU
          GOTO    ALOOP3
          NOP
          NOP
          DECFSZ  CANTPOS
```

```

GOTO    APOSEQU
NOP

RLF     PORTB           ;1 O 1/2 LINEA H SEGUN EL CAMPO
NOP
NOP
NOP
NOP
NOP
NOP
MOV LW  D'21'           ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
BTFS   PORTB,0
ADD LW  D'27'           ;TIEMPO PARA 1 H (160 CICLOS TOTAL)
MOV WF  DURHOR
BSF    PORTB,0
BTFS   FIELD,0
GOTO   ANEXT           ;SE PIERDE 1 CICLO MAS (SOLO 1/2 H)
ANEXT  BCF    PORTB,1
NOP
ALOOP  DECFSZ  DURHOR
GOTO   ALOOP
NOP

AHORIZ BCF    PORTB,0
MOV LW  2
MOV WF  TIEMPO         ;PIERDO TIEMPO PARA
ATIME3  DECFSZ  TIEMPO   ;HACER LOS 4,8µS
GOTO   ATIME3
NOP
NOP
MOV LW  D'48'
MOV WF  DURHOR
BSF    PORTB,0         ;BIT 0 ALTO
ALOOP3 DECFSZ  DURHOR
GOTO   ALOOP3
NOP
DECFSZ  BLKLIN
GOTO   AHORIZ
NOP

AHORIZ1 BCF    PORTB,0
MOV LW  2
MOV WF  TIEMPO         ;PIERDO TIEMPO PARA
ATIME1  DECFSZ  TIEMPO   ;HACER LOS 4,8µS
GOTO   ATIME1
NOP
NOP
MOV LW  D'44'
MOV WF  DURHOR
BSF    PORTB,0         ;BIT 0 ALTO
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
MOV LW  B'00011101'
MOV WF  PORTB
ALOOP4 DECFSZ  DURHOR
GOTO   ALOOP4

```

```

        MOVLW    B'00000001'
        MOVWF    PORTB
        DECFSZ   CANTHB1
        GOTO     AHORIZ1
        NOP

        BCF      PORTB,0
        MOVLW    2
        MOVWF    TIEMPO           ;PIERDO TIEMPO PARA
ATIME2  DECFSZ   TIEMPO           ;HACER LOS 4,8µS
        GOTO     ATIME2
        MOVLW    D'99'
        MOVWF    CANTHB1
        MOVLW    D'44'
        MOVWF    DURHOR
        BSF      PORTB,0         ;BIT 0 ALTO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        MOVLW    B'00011101'
ALOOPH5 MOVWF    PORTB
        DECFSZ   DURHOR
        GOTO     ALOOPH5
        MOVLW    B'00000001'
        MOVWF    PORTB
        DECFSZ   CANTHB2
        GOTO     AHORIZ1
        NOP

;ESTA ULTIMA LINEA/MEDIA LINEA, LA 305, LA USO PARA CARGAR VARIABLES

        BCF      PORTB,0         ;BIT 0 PASA A NIVEL BAJO
        NOP           ;PIERDO TIEMPO PARA
        NOP           ;HACER LOS 4,8µS
        MOVLW    0           ;NO USAR EL BIT 1 DEL PORTB, BIT 0 = SYNC
        BTFSC   FIELD,0
        MOVLW    1
        MOVWF    CARRY
        NOP
        MOVLW    D'15'
        BTFSC   FIELD,0
        ADDLW   D'24'
        MOVWF    DURHOR
        BSF      PORTB,0         ;BIT 0 PASA A NIVEL ALTO
        BTFSS   FIELD,0
        GOTO     ANEXT1
ANEXT1  NOP
        NOP
ALOOPH6 DECFSZ   DURHOR
        GOTO     ALOOPH6
        RRF      CARRY           ;CARRY = 1 SI 1 H, CARRY = 0 SI 1/2 H
        BTFSS   PORTA,2
        GOTO     LECTURA
        BTFSC   PORTA,3
        GOTO     LECTURA
        NOP
        NOP
        NOP

```

```

NOP
NOP
GOTO     INICIO1

;***** CROSSHATCH *****

INICIO2  RRF      FIELD
NOP
NOP
MOVLW   D'4'
MOVWF   BLKLIN
MOVLW   D'28'
MOVWF   CANTHB1
MOVLW   D'10'
MOVWF   CANTHB2
MOVLW   4
MOVWF   CANTPRE           ;SOLO 4 PULSOS POR SER VIDEO NO ENTRELAZADO
MOVLW   5
MOVWF   CANTVER
MOVLW   5
MOVWF   CANTPOS

BPREEQU  BCF      PORTB,0           ;DURACION: 2,6µS ABAJO
MOVLW   D'23'
MOVWF   DUREQU
NOP
NOP
NOP
BSF     PORTB,0
BLOOP1  DECFSZ   DUREQU           ;SE COMPLETAN LOS 32µS ARRIBA
GOTO    BLOOP1
NOP
NOP
DECFSZ  CANTPRE
GOTO    BPREEQU
NOP
BVERT   BCF      PORTB,0
MOVLW   D'22'
MOVWF   DURVER
BLOOP2  DECFSZ   DURVER
GOTO    BLOOP2
BSF     PORTB,0           ;DURACION: 4.8µS ARRIBA ("SERRATED PULSES")
MOVLW   2
MOVWF   TIEMPO
BTIME   DECFSZ   TIEMPO
GOTO    BTIME
NOP
DECFSZ  CANTVER
GOTO    BVERT
NOP
BPOSEQU BCF      PORTB,0
MOVLW   D'23'
MOVWF   DUREQU
NOP
NOP
NOP
BSF     PORTB,0
BLOOP3  DECFSZ   DUREQU
GOTO    BLOOP3
NOP
NOP
DECFSZ  CANTPOS
GOTO    BPOSEQU
NOP

```

```

NOP                                ;1/2 LINEA H (NO ENTRELAZADO)
NOP
NOP
NOP
NOP
NOP
NOP
NOP
MOV LW    D'21'                    ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
NOP
MOV WF    DURHOR
NOP
NOP
NOP
BCF       PORTB,1
NOP
BLOOP    DECFSZ  DURHOR
        GOTO    BLOOP
        NOP

BHORIZ   BCF     PORTB,0
        MOV LW   2
        MOV WF   TIEMPO          ;PIERDO TIEMPO PARA
BTIME3   DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    BTIME3
        NOP
        MOV LW   D'48'
        MOV WF   DURHOR
        BSF     PORTB,0          ;BIT 0 ALTO
BLOOPH3  DECFSZ  DURHOR
        GOTO    BLOOPH3
        NOP
        DECFSZ  BLKLIN
        GOTO    BHORIZ
        NOP

BHORIZ1  BCF     PORTB,0
        MOV LW   2
        MOV WF   TIEMPO          ;PIERDO TIEMPO PARA
BTIME1   DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    BTIME1
        NOP
        MOV LW   9
        MOV WF   CANTLIN
        BSF     PORTB,0          ;BIT 0 ALTO
        NOP
        NOP
BLOOPHA  MOV LW   B'00011100'
        ADD WF   PORTB
        SUB WF   PORTB
        NOP
        MOV LW   2

```

```

MOVWF    DURHOR
BLOOPH4  DECFSZ  DURHOR
GOTO     BLOOPH4
DECFSZ   CANTLIN
GOTO     BLOOPHA
NOP
MOVLW    B'00011100'
ADDWF    PORTB
SUBWF    PORTB
NOP
NOP
NOP
NOP
NOP
NOP
DECFSZ   CANTHB1
GOTO     BHORIZ1
NOP

BCF      PORTB,0
MOVLW    2
MOVWF    TIEMPO           ;PIERDO TIEMPO PARA
BTIMEZ   DECFSZ  TIEMPO   ;HACER LOS 4,8µS
GOTO     BTIMEZ
NOP
NOP
MOVLW    D'44'
MOVWF    DURHOR
BSF      PORTB,0         ;BIT 0 ALTO
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
MOVWF    DURHOR
BLOOPHZ  DECFSZ  DURHOR
GOTO     BLOOPHZ
MOVLW    B'00011101'
MOVWF    PORTB
MOVWF    DURHOR
BLOOPHZ  DECFSZ  DURHOR
GOTO     BLOOPHZ
MOVLW    B'00000001'
MOVWF    PORTB
NOP
NOP
NOP

BCF      PORTB,0
MOVLW    2
MOVWF    TIEMPO           ;PIERDO TIEMPO PARA
BTIME2   DECFSZ  TIEMPO   ;HACER LOS 4,8µS
GOTO     BTIME2
MOVLW    D'28'
MOVWF    CANTHB1
MOVLW    D'44'
MOVWF    DURHOR
BSF      PORTB,0         ;BIT 0 ALTO
NOP
NOP
NOP
NOP
NOP
NOP

```

```

NOP
NOP
NOP
MOVW  B'00011101'
MOVWF  PORTB
BLOOPH5  DECFSZ  DURHOR
GOTO   BLOOPH5
MOVW  B'00000001'
MOVWF  PORTB
DECFSZ  CANTHB2
GOTO   BHORIZ1
NOP

```

;ESTA ULTIMA MEDIA LINEA, LA 305, LA USO PARA CARGAR VARIABLES

```

BCF    PORTB,0          ;BIT 0 PASA A NIVEL BAJO
NOP
NOP                    ;PIERDO TIEMPO PARA
NOP                    ;HACER LOS 4,8µS
MOVW   0                ;NO USAR EL BIT 1 DEL PORTB, BIT 0 = SYNC
BTFSC  FIELD,0
MOVW   1
MOVWF  CARRY
NOP
MOVW   D'15'
NOP
NOP
MOVWF  DURHOR
BSF    PORTB,0          ;BIT 0 PASA A NIVEL ALTO
NOP
NOP
NOP
NOP
BLOOPH6  DECFSZ  DURHOR
GOTO   BLOOPH6
RRF    CARRY            ;CARRY = 1 SI 1 H, CARRY = 0 SI 1/2 H
BTFSC  PORTA,2
GOTO   LECTURA
BTFSS  PORTA,3
GOTO   LECTURA
NOP
NOP
NOP
NOP
GOTO   INICIO2

```

;***** PUNTOS *****

```

INICIO3  RRF    FIELD
NOP
NOP
MOVW   D'4'
MOVWF  BLKLIN
MOVW   D'28'
MOVWF  CANTHB1
MOVW   D'10'
MOVWF  CANTHB2
MOVW   4
MOVWF  CANTPRE
MOVW   5
MOVWF  CANTVER
MOVW   5
MOVWF  CANTPOS

```

```

CPREEQU BCF      PORTB,0      ;DURACION: 2,6µS ABAJO
        MOVLW    D'23'
        MOVWF    DUREQU
        NOP
        NOP
        BSF      PORTB,0
CLOOP1  DECFSZ   DUREQU      ;SE COMPLETAN LOS 32µS ARRIBA
        GOTO     CLOOP1
        NOP
        NOP
        DECFSZ   CANTPRE
        GOTO     CPREEQU
        NOP
CVERT   BCF      PORTB,0
        MOVLW    D'22'
        MOVWF    DURVER
CLOOP2  DECFSZ   DURVER
        GOTO     CLOOP2
        BSF      PORTB,0      ;DURACION: 4.8µS ARRIBA ("SERRATED PULSES")
        MOVLW    2
        MOVWF    TIEMPO
CTIME   DECFSZ   TIEMPO
        GOTO     CTIME
        NOP
        DECFSZ   CANTVER
        GOTO     CVERT
        NOP
CPOSEQU BCF      PORTB,0
        MOVLW    D'23'
        MOVWF    DUREQU
        NOP
        NOP
        NOP
        BSF      PORTB,0
CLOOP3  DECFSZ   DUREQU
        GOTO     CLOOP3
        NOP
        NOP
        DECFSZ   CANTPOS
        GOTO     CPOSEQU
        NOP

        NOP      ;1/2 LINEA H
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        MOVLW    D'21'      ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
        NOP
        NOP
        MOVWF    DURHOR
        NOP
        NOP
        NOP
        NOP
        BCF      PORTB,1
        NOP
CLOOP   DECFSZ   DURHOR
        GOTO     CLOOP

```

```

NOP

CHORIZ BCF PORTB,0
        MOVLW 2
        MOVWF TIEMPO ;PIERDO TIEMPO PARA
CTIME3 DECFSZ TIEMPO ;HACER LOS 4,8µS
        GOTO CTIME3
        NOP
        NOP
        MOVLW D'48'
        MOVWF DURHOR
        BSF PORTB,0 ;BIT 0 ALTO
CLOOPH3 DECFSZ DURHOR
        GOTO CLOOPH3
        NOP
        DECFSZ BLKLIN
        GOTO CHORIZ
        NOP

CHORIZ1 BCF PORTB,0
        MOVLW 2
        MOVWF TIEMPO ;PIERDO TIEMPO PARA
CTIME1 DECFSZ TIEMPO ;HACER LOS 4,8µS
        GOTO CTIME1
        NOP
        NOP
        MOVLW D'48'
        MOVWF DURHOR
        BSF PORTB,0 ;BIT 0 ALTO
CLOOPHZ DECFSZ DURHOR
        GOTO CLOOPHZ
        NOP
        DECFSZ CANTHB1
        GOTO CHORIZ1
        NOP

        BCF PORTB,0
        MOVLW 2
        MOVWF TIEMPO ;PIERDO TIEMPO PARA
CTIMEZ DECFSZ TIEMPO ;HACER LOS 4,8µS
        GOTO CTIMEZ
        NOP
        NOP
        MOVLW 9
        MOVWF CANTLIN
        BSF PORTB,0 ;BIT 0 ALTO
        NOP
        CLOOPHA MOVLW B'00011100'
        ADDWF PORTB
        SUBWF PORTB
        NOP
        MOVLW 2
        MOVWF DURHOR
CLOOPH4 DECFSZ DURHOR
        GOTO CLOOPH4
        DECFSZ CANTLIN

```



```

    BTFSC    FIELD,0
    MOVLW   1
    MOVWF   CARRY
    NOP
    MOVLW   D'15'
    NOP
    NOP
    MOVWF   DURHOR
    BSF     PORTB,0           ;BIT 0 PASA A NIVEL ALTO
    NOP
    NOP
    NOP
    NOP
CLOOPH6  DECFSZ  DURHOR
         GOTO   CLOOPH6
         RRF    CARRY           ;CARRY = 1 SI 1 H, CARRY = 0 SI 1/2 H
         BTFSS  PORTA,2
         GOTO   LECTURA
         BTFSS  PORTA,3
         GOTO   LECTURA
    NOP
    NOP
    NOP
    NOP
    GOTO   INICIO3

END

```