Video pattern generator

Introduction

The objective of this project was to construct a Composite Video Pattern Generator, which can be used as a teaching tool in the video field as well as a laboratory instrument, useful for repairing and adjusting television receptors. This instrument is in no way intended to be a commercial set, because I cannot guarantee all the specifications that a professional tool must have. However, the result is a high quality video equipment.

Specifications

Patterns:	Color Bars, Raster, Cross-hatch, Points
Controls:	independent R, G, B, luminance, chrominance and color burst
Video output:	composite video, 1 Vp-p at 75 ohms load
Color system:	PAL-N (option PAL-B/G/I, changing chroma oscillator crystal)
Scanning system:	Interlaced (Bars and Raster), Non-Interlaced (Cross-hatch and Points)
Power supply:	12 Vdc (8 x AA alkaline batteries)
Power consumption:	70 mA maximum (White Raster)
Applications:	television receptors adjustment (purity, convergence, linearity, etc.), troubleshooting of video section in television sets and video recorders, teaching of video generation techniques

General overview

The human eye is capable to “see” images using “receptors” located on the retina. There are two types of receptors, classified by function: the Rods, in charge of black and white perception, and the Cones, responsible for color discrimination. If we concentrate only in the cones, we will discover that there are three types of them: one type reacts to Red light, another to Green light and the last group reacts to Blue light. We only “catch” three colors; however, we actually “see” all the colors.

Here comes a basic rule of color generation: **to know the color information of an object, we only need to know the relative amount of the three basic colors: Red, Green and Blue**. By this reason, these three colors are known as **Primary Colors**, because we can make any other color by combining them in the adequate proportion.

Now, let’s think for a moment. When we learned the colors at school our teacher probably told us that the primaries were Red, Yellow and Blue. Another strange fact: if we observe the way in which ink printers create colors, we will notice that there are only three color cartridges, Magenta, Yellow and Cyan. But they print in full color. Someone must be wrong…

This “confusion” results from the existence of two groups of primary colors: the **Additive Primaries** and the **Subtractive Primaries**. In order to understand the difference between them let’s see a couple of examples.

Suppose that we illuminate a white wall with a green light. Obviously we will see green, because this is the color reflected by the wall. If we now turn the light into red, again we will see red. But, if we illuminate the wall with both colors (green and red) at the same time, the resulting color will be yellow, which comes from the sum or addition of the two original colors. As we can observe, the sum of this colors gives a new, secondary, color. By this reason, the colors Red, Green and Blue are called Additive Primaries.

Now suppose that we paint a white paper with yellow paint, and we illuminate it with white light (it contains all the colors). Obviously we will see a yellow paper. Why?. Because the paint retains (subtract) all the components of the white light except the yellow, which is reflected to our eyes. If we now add cyan paint, the resulting color is green. This fact reveals that the mixture of paints (yellow and cyan) subtracts all the colors except green. How can we explain this?. Let’s see some equations.

If we add all additive primaries we obtain white:

RED LIGHT + GREEN LIGHT + BLUE LIGHT = WHITE LIGHT

If we add only two of them:

RED LIGHT + GREEN LIGHT = YELLOW LIGHT
RED LIGHT + BLUE LIGHT = MAGENTA LIGHT
GREEN LIGHT + BLUE LIGHT = CYAN LIGHT

Here we see the key of our analysis: the so called Subtractive Primaries can be obtained by adding two of the additive primaries.

So, when we see the yellow paint, we are really seeing red and green light added. This means that the yellow paint catches the blue color of the white light and reflects the other two.

In the same way, the cyan paint catches the red light, reflecting green and blue.

Now, the result of our experiment is evident. If we mix yellow paint (captures blue) with cyan paint (captures red), the only color that is effectively reflected is green, which is just the color we saw.

What would happen if we mix all subtractive colors together?. Obviously we will see black, because all “lights” are captured:

CYAN + MAGENTA + YELLOW = BLACK

One question still remains unanswered. Why do we often hear that primary colors are red, yellow and blue?. The answer is simple. We learn the colors through painting at school. We already know that primaries for paint are the subtractive ones, that is yellow, magenta and cyan. These two last colors, due to their “reddish” and “bluish” tones are usually called “red” and “blue”, respectively.

Colors in television

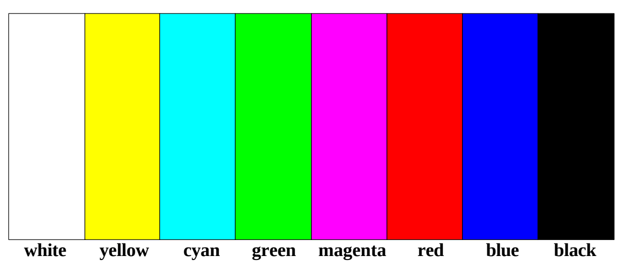
The image in a color television receiver is obtained by the emission of light that results from the excitation of a phosphor layer that covers the interior of the glass screen, when it is hit by an electron beam that periodically scans the visible surface. If we speak about “emission of light” we must immediately think of “Additive Processes”, which leads us to the conclusion that the Primary Colors in television are Red, Green and Blue (RGB). Actually, three electron beams are generated inside a television tube (CRT, Catode Ray Tube), each of them hit a particular kind of phosphor on the screen, which will generate a characteristic color, depending on the chemical structure of that phosphor. Naturally, this three colors are Red, Green and Blue.
All the rest of the colors (and I really mean “all”) can be obtained by the adequate combination of these three primary colors.

Basic color bar generator

A basic color bar generator should be like this:

- it should have three outputs, one for each primary color
- each of these outputs should be connected to the corresponding input of the TV
- the set should generate combinations of its outputs, as shown on the following chart:

Blue	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
Red	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**
Green	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**



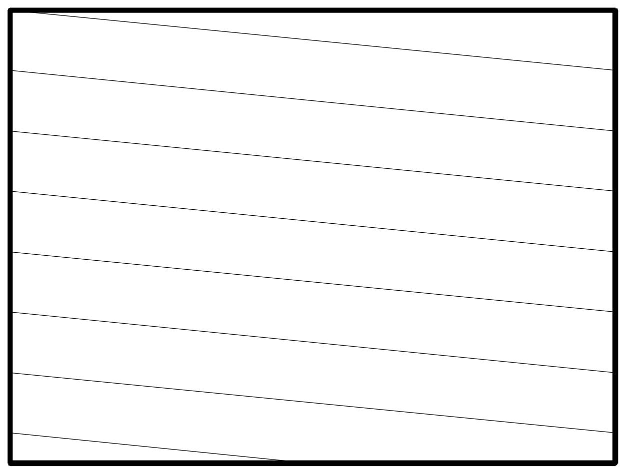
In this chart “1” means “presence” of color, while “0” means “absence”. In practice, these “ones” and “zeros” are represented by voltage levels, e.g. 5V and 0V respectively.
As you can see it is very easy to construct a generator of this kind, because you only need a minimum of digital electronics to obtain this bars. So, why should we complicate ourselves with a more elaborated design?.
Most of TV receivers and VCRs (Video Cassette Recorders) DO NOT have RGB inputs, these can only be found on some video monitors reserved for professional field (Broadcasting). Home appliances usually have a standard input of “Composite Video”, labeled “VIDEO IN”. By this reason, our generator should be able to convert the components, RGB, into this “Composite Video”.

Composite Video

Red, Green and Blue signals (RGB) contain all the information needed to retrieve an image, but they would need a large bandwidth to be transmitted, and television is based on the “transmission of images”. In order to reduce this bandwidth, and at the same time keep compatibility between the original “Black and White” transmissions and the new “Color System”, the “Composite Video Signal” was created.

Into this signal the transmitter sends the information of “brightness” (Luminance) and “color” (Chrominance) of an image, and also all the necessary synchronization pulses to correctly retrieve the image on the TV screen.

Which are these so called “synchronization pulses”?. To answer this question we should first analyze how an image is formed on the TV screen.



An electron beam (let’s consider only one, we know they are actually three) scans the screen from left to right and from top to bottom, as shown on the diagram on the right. While scanning the screen it excites the phosphor layer with variable strength, generating a visible image.

As it is shown, although the image is formed in a two coordinate screen (a plane), it is actually generated by a number of consecutive lines. In this way the signal comes into the TV, line by line. By this reason, it is obvious that we need to synchronize the electron beam that scans the screen with the scanning generated in the TV station. If we did not synchronize the beam, the images would appear cut, with diagonal stripes, or with wrong color information (like it happens with some “coded” channels, in which the synchronization pulses are removed).

So, let’s synchronize the beam scanning. We must tell it where a new image begins (Vertical Synchronism or “V”) and where each line begin (Horizontal Synchronism or “H”). Obviously the H. Sync. has higher frequency components than V. Sync., because there are many lines inside each image.

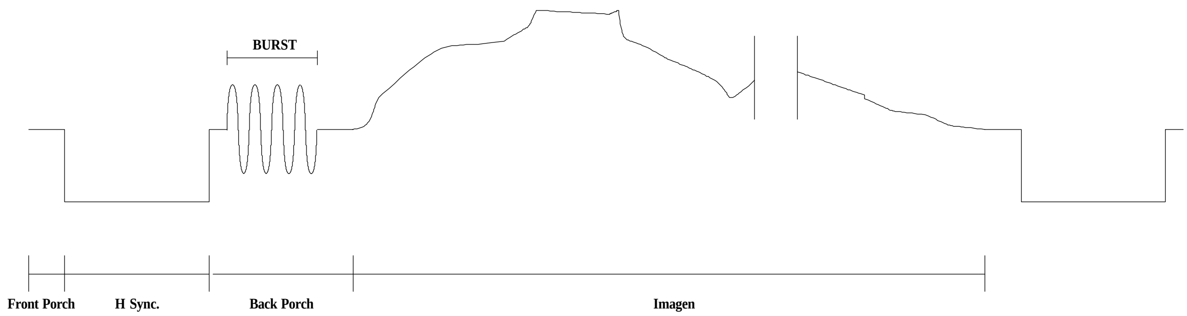
How many lines are there inside an image?. The answer is: “it depends on the transmission norm we are considering”.

Transmission Norms

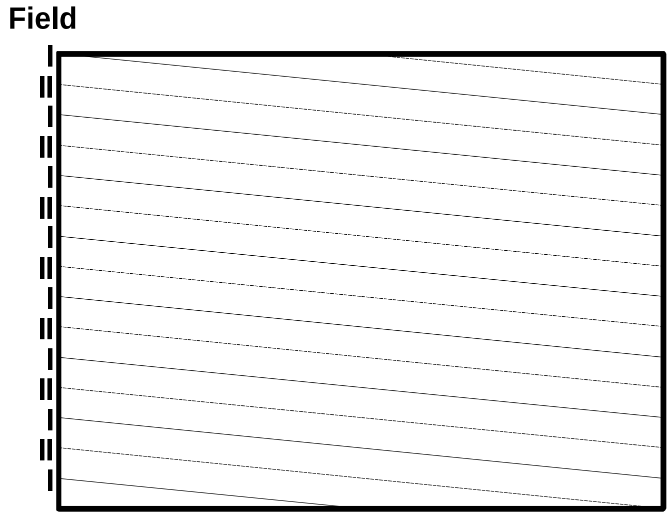
A norm is a collection of parameters adopted as rules inside a particular group or region in order to keep a clear and well-defined relation between the component parts. In television, a norm is exactly that. The transmission norms establish the parameters that transmitters and receivers must comply in order to keep a reliable, error-free communication between them. Each norm establish a lot of parameters; let’s analyze the ones that are directly related to our project.

Horizontal Frequency (H): frequency of line repetition.
Horizontal Synchronism(H Sync): pulse that indicates the beginning of a line.
Vertical Frequency (V): frequency of image repetition (fields).
Vertical Synchronism (V Sync): pulse that indicates the beginning of an image (field).
Horizontal Lines: amount of lines that compose a complete image (frame).
Color Subcarrier (SC): frequency that is modulated to send color information.
Color Burst: SC burst to "synchronize" color demodulation.
Front Porch: image-free interval previous to H. Sync.
Back Porch: image-free interval after H Sync., where color burst lies.

This can be summarized in the following diagram:



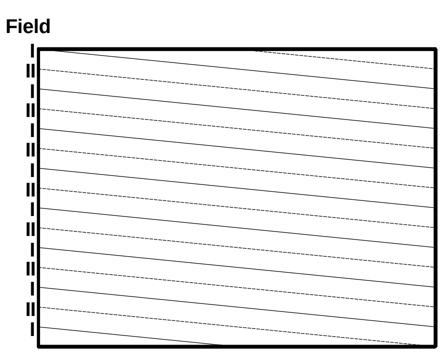
When defining parameters I mentioned two words that I intentionally left unexplained: Field and Frame. Now it is time to explain them, but first we must have a clear idea of what "Interlaced Scanning" means.
As previously stated, in order to reconstruct the image on the TV screen the electron beam scans it line by line, from top to bottom. A single screen scan would be enough to create a complete image. However, this is not true. The beam has to scan the whole screen twice in order to have the image completed. Let's analyze this fact carefully.
When a video camera at the TV studio scans the image to transmit, it does it by dividing the image into horizontal lines. This lines are transmitted to the TV receiver, in order to be sequentially reproduced on the screen. However, the lines are not transmitted consecutively (line 1, 2, 3, 4, …); actually odd lines are transmitted first (line 1, 3, 5, …) and then even lines (line 2, 4, 6, …). In the same way the electron beam scans the screen, generating the odd lines first and then the even ones. This is the reason why the electron beam must scan the screen twice to generate the full image.
The two "half-images" are called "Fields" (Odd and Even, respectively), while the full image is known as "Frame"(see figure).



In this figure the Odd Field is indicated as "I", while the Even Field is "II".
As you can see, the odd field ends with half a line and the even field begins also with half a line. This is a characteristic fact of one of the two scanning systems we will analyze: the Interlaced Scanning and the Non-Interlaced Scanning.

The figure analyzed is a clear example of Interlaced Scanning. Its name comes from the fact that lines of consecutive fields are actually "interlaced". Which is the advantage of this scanning technique?. It permits a higher image repetition frequency without increasing the bandwidth requirements. How?. Through the transmission channel the TV station sends a fixed number of images per second (let's say 25). The TV receiver reproduces these 25 images per second, but it actually scans the screen 50 times per second. Due to our eye characteristics, it looks like 50 images per second, greatly reducing image flicker. As an additional fact, the same thing happens in projection systems (cinema). The first movies were filmed on a 16 frames/second basis, making flicker really visible. Nowadays the standard fixed 24 frames/second for movies, and the shutter opens twice in each "photo", with the same purpose of reducing flicker.
Back to television, if this is such a good method, there is no need to discuss about the Non-Interlaced option. However, the Interlaced Scanning, so good for pictures with movement, fails when trying to reproduce static images, such as charts, lines, etc. Why? Let's imagine a white horizontal line fixed in the middle of the screen over a black background. When the odd field is scanned the line will appear at a certain position on the screen; in the even field this position will naturally change, it will be one line upper or lower than the odd field. The next field will be an odd one, returning to the original position, and so on. So, the line will have vertical flicker, not very noticeable, but really hard if you had to be looking at the screen for quite a long time. How can be solved?. Easy. You must use Non-Interlaced Scanning. This is much easier. Simply, the scanning lines follow the same path in both fields. This technique is widely used in computer monitors and in pattern generators while outputting signals like "Cross-hatch" or "Points".

Interlaced Scanning



Non-Interlaced Scanning

“N” Norm

There are many television transmission norms in the world today, named with letters from “A” to “N”. Some are obsolete, but most of them are still in use.
Let’s see the actual values established for the parameters previously defined, as stated in the “N” norm definition. This norm is only used in three countries: Argentina, Uruguay y Paraguay.

Horizontal Frequency (H):	15626 Hz (line duration: 64 µsec.)
Horizontal Synchronism(H Sync):	4.8 µsec.
Vertical Frequency (V):	50 Hz (field duration: 20 msec.)
Vertical Synchronism (V Sync.):	2.5 horizontal lines
Horizontal Lines:	625 lines per frame (312.5 lines per field)
Color Subcarrier (SC):	3.582056 MHz
Color Burst:	9 to 11 SC cycles
Front Porch:	1.9 µsec.
Back Porch:	5 µsec.

If we compare these values with those established by European norms (B, G, I) we will see that they are quite similar, with the exception of the color subcarrier (European systems use 4.43 MHz).
To conclude with norms, let’s see a diagram that represents an actual video signal (N norm), using the scanning methods previously analyzed (see figure on next page).

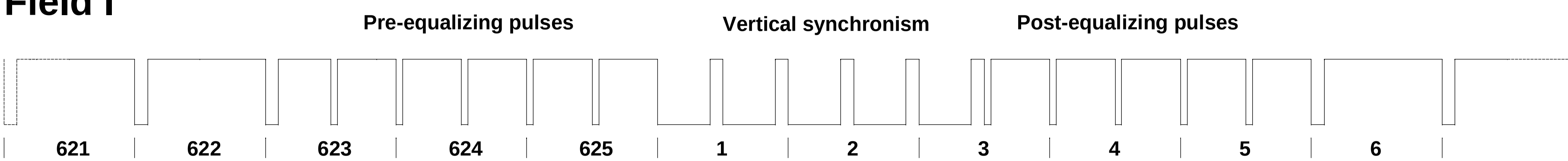
Let’s concentrate only in the interlaced method. It is clear that the V. Sync. pulse covers 2½ horizontal lines (line 1, 2, and half of line 3 in field I). But there is something not mentioned before: there are some Pre and Post-Equalizing pulses. What is this?. The V. Sync. detection in the TV receiver is accomplished by integrating the incoming video signal. When it reaches a preset value it triggers the vertical oscillator. The integration is practically done by mean of an RC circuit. The voltage of the capacitor is used to trigger the system. But, depending on the image previous to the V. Sync. the start voltage of the capacitor could change, making the triggering of the vertical oscillator very image-dependent, thus provoking unstable pictures.
To avoid this we need to make the starting voltage of the capacitor equal from field to field, and that is exactly the reason to include the equalization pulses
This pulses have a frequency twice the horizontal (H), and the duration is half of H. Sync. They are applied during a period equal to 2½ horizontal lines.

Finally, the “positive” pulses that appear inside de V. Sync. pulse are called Serrated Pulses, and the reason to include them is to keep the horizontal oscillator locked during V. Sync. period. The duration of this pulses is equal to H. Sync.

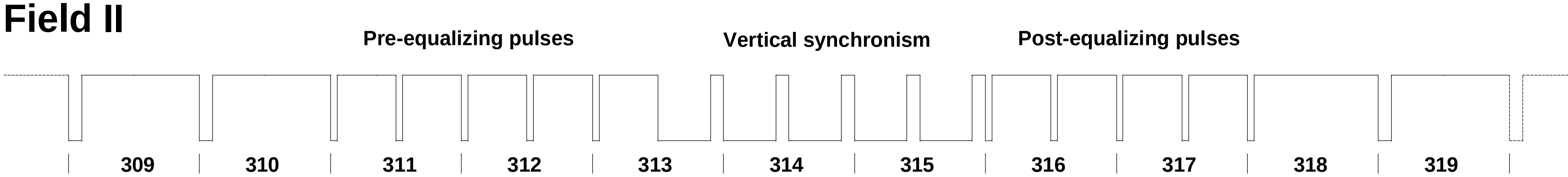
TV scanning methods - PAL-N system

Interlaced scanning

Field I

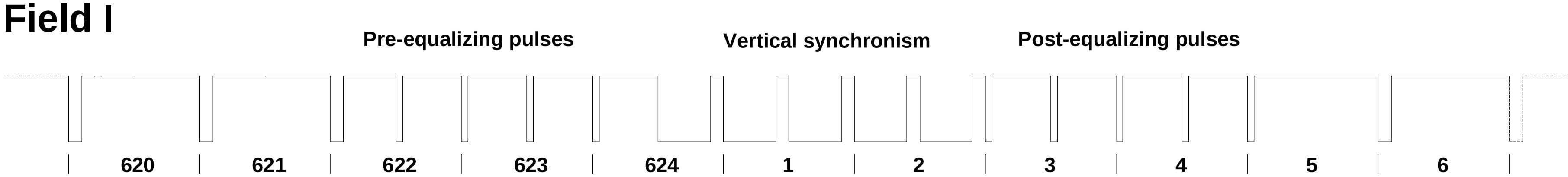


Field II

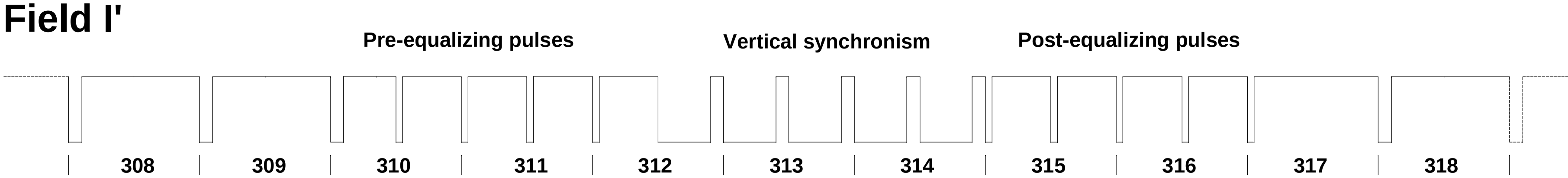


Non-interlaced scanning

Field I



Field I'



Color systems

Color television appeared many years after the "Black and White" system was invented, so it had to adapt itself to the system in use. By this reason, it was necessary to develop compatible color systems, in order to let the old TV receptors reproduce the new color signals (in black and white, obviously) and also the new TV receivers should be able to reproduce signals coming from old monochrome transmitters. Nowadays we can find three color systems spread around the world: NTSC, PAL and SECAM. We will study the PAL system, because we want to generate it.
The PAL system basically sends color information by phase modulating a color carrier with two signals, which have a relative phase shift of 90º. Let's see the process step by step.
We previously stated that the color information of an object is fully contained in the three primary components, Red, Green and Blue. Those are the colors actually "seen" by our eyes and, if we transmit them, then we transmit the real color of the object. What we did not say is the fact that our eyes have different sensibility to each of the three primaries. We are very "receptive" to Green, not so much to Red and less to Blue. If we put it into relative percentages of sensibility:

GREEN: 59%
RED: 30%
BLUE: 11%

This means that the brightness of an object (Luminance or simply "Y"), which is the total amount of light reflected by the object, can be represented like this:

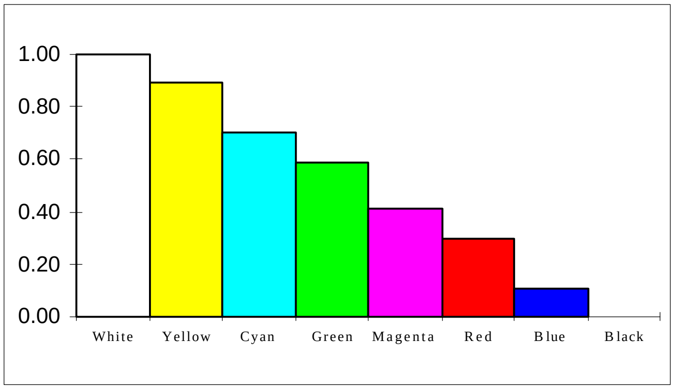
Y = 0.30Red + 0.59Green + 0.11Blue

From now on we will name Red as "R", Green as "G" and "B" will mean Blue, so:

Y = 0.30R + 0.59G + 0.11B

This Luminance information (Y) is the brightness of each combination of colors on the screen, and is the only information used for monochrome transmissions. Let's see the relative brightness of each color, calculated from the previous equation:

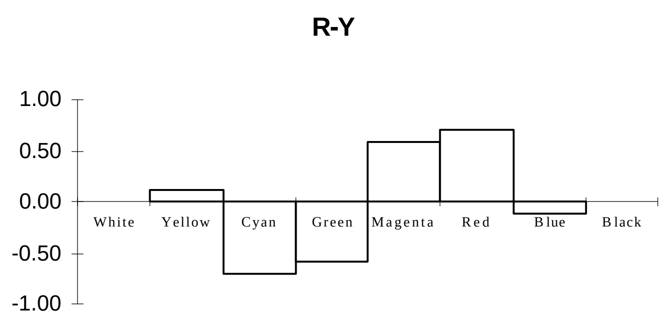
Color:	R	G	B	Y
White	1	1	1	1.00
Yellow	1	1	0	0.89
Cyan	0	1	1	0.70
Green	0	1	0	0.59
Magenta	1	0	1	0.41
Red	1	0	0	0.30
Blue	0	0	1	0.11
Black	0	0	0	0.00



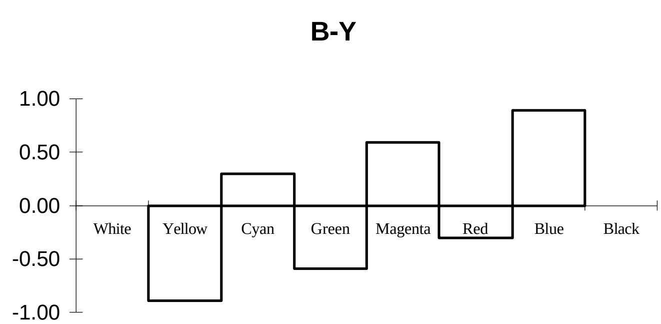
Luminance (Y) is one of the basic components of the video signal and, as we already stated, it is obtained as the sum of the RGB components of the image. So, it would be enough to send the information of only two of the three components (RGB) together with Y, because the missing component can be easily obtained from the Y signal, making simple calculations at the receptor side. Since G is the predominant component in Y it will not be sent separately, and will be recovered from Y in the TV receptor.
Remember that all this complication arose from the fact that sending RGB as separate components would need a considerable bandwidth, which is not allowed because we are trying to accommodate a color transmission in the same channel previously assigned for monochrome signals.
So we will send Y, R and B. But R and B will not be sent as components, we will send the so called “color difference signals”: **R-Y** and **B-Y**.

Let’s see how these signals look like:

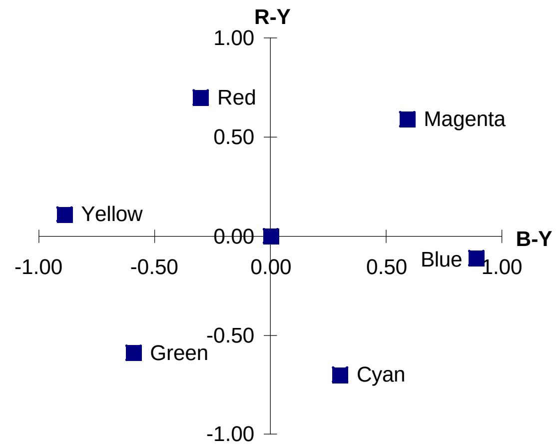
Color:	R	G	B	R-Y
White	1	1	1	0.00
Yellow	1	1	0	0.11
Cyan	0	1	1	-0.70
Green	0	1	0	-0.59
Magenta	1	0	1	0.59
Red	1	0	0	0.70
Blue	0	0	1	-0.11
Black	0	0	0	0.00



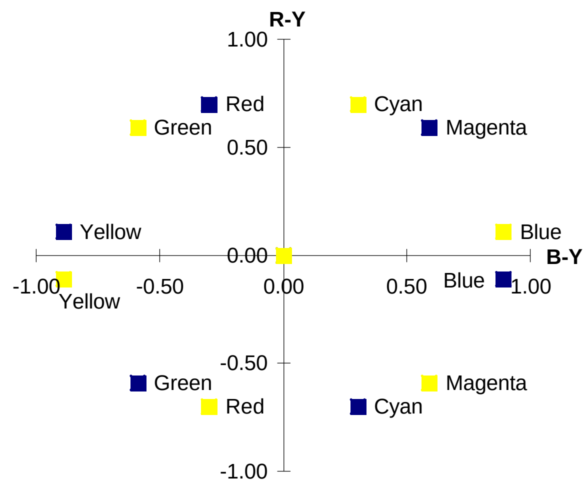
Color:	R	G	B	B-Y
White	1	1	1	0.00
Yellow	1	1	0	-0.89
Cyan	0	1	1	0.30
Green	0	1	0	-0.59
Magenta	1	0	1	0.59
Red	1	0	0	-0.30
Blue	0	0	1	0.89
Black	0	0	0	0.00



This “Color Difference Signals” will be the responsible of phase modulating the color subcarrier. The B-Y signal will modulate the subcarrier with a phase shift of 0º, while the R-Y signal does the same but with a 90º phase shift. So, if we construct an X-Y graph of this two signals we obtain a “Phase Diagram” of the different colors, as seen on the right.



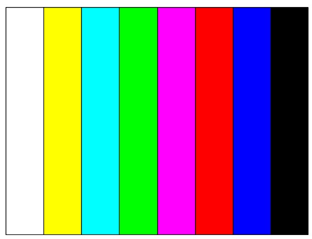
At the same time, and this is characteristic for the PAL system, the 90º shifted subcarrier will have another 180º phase shift from one horizontal line to the next, this means that R-Y modulates the 90º shifted subcarrier in one line, and for the next line it will modulate the subcarrier at 270º, and so on. This characteristic gives the name of the system: PAL = Phase Alternation by Line.

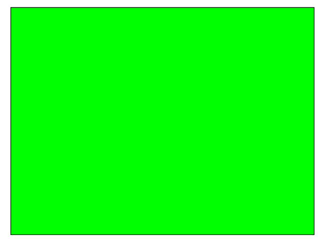


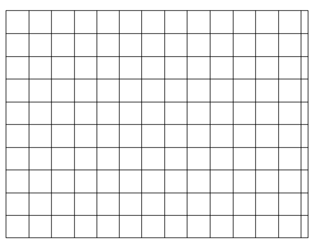
Once modulated, the color subcarrier is called “Chrominance Signal” or “C”. Adding the luminance signal (Y) plus the chrominance signal (C) we finally obtain the signal we were looking for, the COMPOSITE VIDEO SIGNAL.

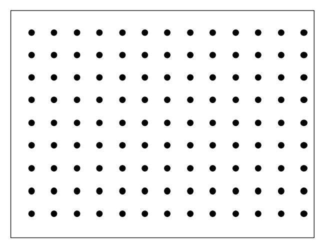
Construction of a video pattern generator

A pattern generator should be capable not only to produce the RGB components but also to generate the corresponding composite video signal, with all its components: synchronism pulses, luminance, modulated color subcarrier, etc.
The first thing we must define is the amount and type of patterns the set should generate, because this will directly determine the characteristics and complexity of the circuit to develop.
Our generator will be capable of producing four basic patterns:


Bars

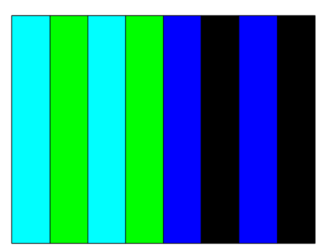

Raster


Cross-hatch


Points

It will also provide independent control for the three components, R, G and B, as well as for luminance (Y) and chrominance (C). This fact greatly increases the pattern generation capability, because the Raster could be of any of the eight colors, the Bars could be monochrome or adopt different color combinations, as shown in the following diagrams:


Bars: B off


Bars: R off

As an additional control we can suppress the color burst, which is very useful to troubleshoot the color processing stages of the TV receiver.
In order to select which of the four basic patterns will be generated we have two switches (S4 y S5), selecting a pattern as show in the following table:

Switches:	S4 OFF	S4 ON
S5 OFF	BARS	RASTER
S5 ON	CROSS-HATCH	POINTS

The OFF (or “zero”) and ON (or “one”) states mean that the middle terminal of the switch is connected to ground (0V) or VCC (5V) potentials respectively.
Once defined the characteristics of the generator let’s see how to do it.

Synchronism and pattern generation

Time base, synchronism and the four basic patterns generation will be accomplished by a microcontroller (PIC16F84-10), so this stage would be basically a Software development. At the end of this stage the microcontroller will be able to do the following:

- Generate a stable time base, from which all required times will be obtained.
- Generate in one of it pins, the one that corresponds to **Bit 0 of PORTB**, all the necessary synchronism pulses to comply with the requirements of the selected television norm (N), without adding video to this signal (pure synchronism signal).
- Generate the R, G and B signals, using three different pins. This signals will have the information required to generate the selected pattern, and will not have added synchronism pulses (pure video signal). The pin designation will be the following:

PORTB (2) = B (Blue)
PORTB (3) = R (Red)
PORTB (4) = G (Green)
(The corresponding bit of PORTB is indicated between brackets)

- Accept in two of its pins, configured as inputs, the commands from switches S4 and S5, in order to let the user select the pattern to generate. This two inputs correspond to two Bits of PORTA, as shown:

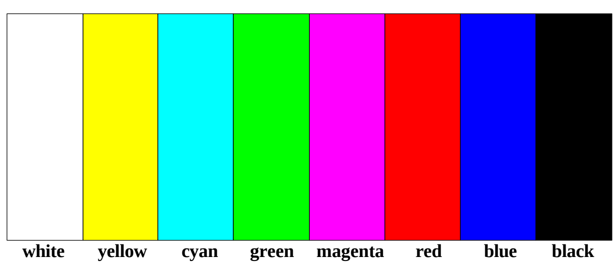
PORTA (2) = S4
PORTA (3) = S5

Now we have the objectives clear; let’s see how the program works.

It is basically composed of four independent blocks, each of them has a complete set of routines in order to generate a complete image. In the section of Diagrams I included a flowchart of the program, which will help to understand the following explanation.
After a first stage, in which all variables are defined and loaded with the initial values, the program reads the status of the two switches, S4 and S5. Depending on the combination of

these switches the program will be addressed to one of the four mentioned blocks, corresponding to one of the four basic patterns.
Each one of these blocks begins with the generation of pre-equalizing pulses, then vertical synchronism with the corresponding serrated pulses, followed by post-equalizing pulses.
After that the field is selected: odd or even. This is very important because we are working with interlaced scanning, which means that the first line of the odd field is a full line, while the even field begins with half line. If we did not take this into account the result would be an unstable image, with a noticeable flicker in the upper side.
Note that in two of the four patterns (Cross-hatch and Points) we will use non-interlaced scanning, in order to avoid the flicker of the fixed horizontal lines or points. In this case the first line is always a full one; in order to compensate for this, we need to eliminate one pre-equalizing pulse (half horizontal line), as previously shown in video signal diagrams.
Then the program generates 3 or 4 horizontal lines without video, depending on the field, so as to compensate for time differences (only in interlaced patterns).
Now it is time to activate the RGB lines. After generating the horizontal synchronism and back porch time, the video signal that correspond to the selected pattern is issues through the RGB lines. How is this achieved?. Let's see an example.
Suppose that we are generating a color bar signal. There are eight bars, so we must divide the usable video time into eight equal intervals.
Before going on we must remember that the usable video time is the time during which the generated information is effectively shown on the TV screen. In PAL-N each line has a duration of 64 µsec.; this time includes 4.8 µsec. of horizontal synchronism, 1.9 µsec. of front porch and 5 µsec. of back porch. So we only have 52.3 µsec. remaining to show video, and this is our usable time.
Back to our example, we already defined the eight intervals. Let's see how we should issue the RGB signals in each of them:

Blue	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**
Red	**1**	**1**	**0**	**0**	**1**	**1**	**0**	**0**
Green	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**



As you can see this is a well known diagram. We already mentioned it to describe what a simple RGB bar generator would do, and that is just what we are doing now.

Let’s analyze the generation of a Raster signal. This is much simpler: all the usable time we must issue a high level signal through the three RGB lines. But, if all lines RGB are active at the same time, the result will always be a white Raster. True. Color selection is achieved by controlling the RGB signals outside the microcontroller, purely by hardware (three switches).
To generate lines or points we need more elaborated routines, because we must control not only the time in the horizontal direction but also the number of lines in the vertical direction, in order to keep an equal distance between lines or points. However, this is not a problem; we just add another variable to keep the count of lines and that’s all.
What about RGB lines?. They are all active when drawing lines or points, so they are white.

If you analyze the program you will probably notice that the generation of horizontal lines and video signals within a video block is repeated three times. There is a simple reason for this. In each run inside a video block a complete field is generated, that is 312.5 horizontal lines. To achieve this we must count the lines and keep this count in a certain register. Since I used an eight bit signed register, the maximum number that can be stored is 127, so I needed to load it three times to achieve the required number of lines.

To finish with the video block, after each field is completed the program evaluates the condition of the two switches (S4 and S5). If they remain unchanged, the program continues within the same block; if there is any change it jumps to the initial evaluation routine, and then goes to the selected video block.

And that’s all. May be there is still an unclear issue… how do I calculate the time inside the program?. When you use a microcontroller this is very easy, you only have to count “instruction cycles”. Using a 10 MHz oscillator and knowing the fact that each instruction cycle needs four oscillator cycles, we can easily calculate the time of one instruction cycle:

$\mathbf{T_{osc} = 1/f_{osc}}$
$\mathbf{T_{ins} = T_{osc} \times 4}$
$\mathbf{T_{ins} = 1/10\ MHz \times 4 = 0.4\ \mu sec.}$

If each instruction cycle lasts for 0.4 µsec., then we need to count 12 cycles in order to obtain the horizontal synchronization pulse:

12 x 0.4 µsec. = 4.8 µsec.

In the same way we can calculate the cycles needed for a complete horizontal line, 160 instruction cycles:

160 x 0.4 µsec. = 64 µsec.

So, this is what the program does. It counts instructions and set or clear, as required, the Bit 0 of PORTB. In our case, during the equalization and synchronization pulses (H or V) this bit will be clear (0V) and the rest of the time it will remain set (5V).

Composite Video generation

As we already stated, it is not enough to generate RGB signals to have a practical video generator, that can be connected to TV receivers or VCRs. We must combine this RGB signal with the synchronization signal and generate Composite Video, which is a practical signal to test receivers.
We already analyzed all the steps needed to obtain Composite Video from RGB, so we will not repeat it here. It is a hard process if you have to do it “manually”. Fortunately, there is an integrated circuit, designed by Motorola®, that complies with the following specifications:

- It has four signal inputs: Synchronism, R, G and B
- From RGB it generates luminance (Y)
- Has an in-circuit oscillator, which generates the color subcarrier
- Generates B-Y and R-Y signals, with the phase alternation required by the PAL system
- From B-Y and R-Y generates chrominance (C)
- Mixes Y with C to obtain Composite Video

As you can see, a single IC does exactly what we need. And it requires exactly the four signal we already generated with the microcontroller.
This IC is the MC1377, RGB ENCODER, and with a few external components it can be fully functional. In fact, I used the configuration suggested in the data sheet, with some modifications to improve its performance.
The crystal used corresponds to PAL-N color subcarrier frequency, 3.582056 MHz. If you want to use this equipment in Europe, in those countries using PAL-B/G/I, you only need to replace the crystal by another one with the proper frequency (4.43 MHz) and make minor adjustments to the TRIMMER CV1.
In this stage we have control of all the signals: RGB, Y, C and color burst. There are basically six switches that derives the signal to ground, directly (RGB) or through a capacitor (Y, C). In the case of color burst, to eliminate it, the switch (S8) disconnect a capacitor (C04), responsible for generating the burst duration time.
Let’s see a summary of the switches and their function:

Switch	Function
S1	**G ON/OFF**
S2	**R ON/OFF**
S3	**B ON/OFF**
S4	**PROGRAM**
S5	**PROGRAM**
S6	**Y ON/OFF**
S7	**C ON/OFF**
S8	**BURST ON/OFF**
S9	**POWER**

Once obtained the Composite Video signal, its level and impedance are adjusted by sending it through a buffer circuit, composed by Q1, R14 and R15.

This concludes the signal generation, and practically the circuit description. I only have to mention that the two main integrated circuits have different supply voltages, so you can see a main power supply of 12V (8 AA alkaline batteries, this is a portable set) for the video sector (U2 and Q1), and a secondary power supply of 5V, obtained from the main one, for the microcontroller (U1).

Practical implementation of the video generator

In the following pages you will find all the necessary information, diagrams and drawings, to construct a really working video generator. I have included the printed circuit board layout in actual size, so you only need to print it on a transparency film and transfer to the board. Note that the circuit is inverted, in order to make easier the mounting stage, using the component layout diagram provided. In the actual board the text “Generador de video” should be in the right direction.
About the program, I put it entirely in the final pages. You only need to copy it into a text editor, assemble it and load it into the PIC, using the tools provided by Microchip® or the ones you may have developed.

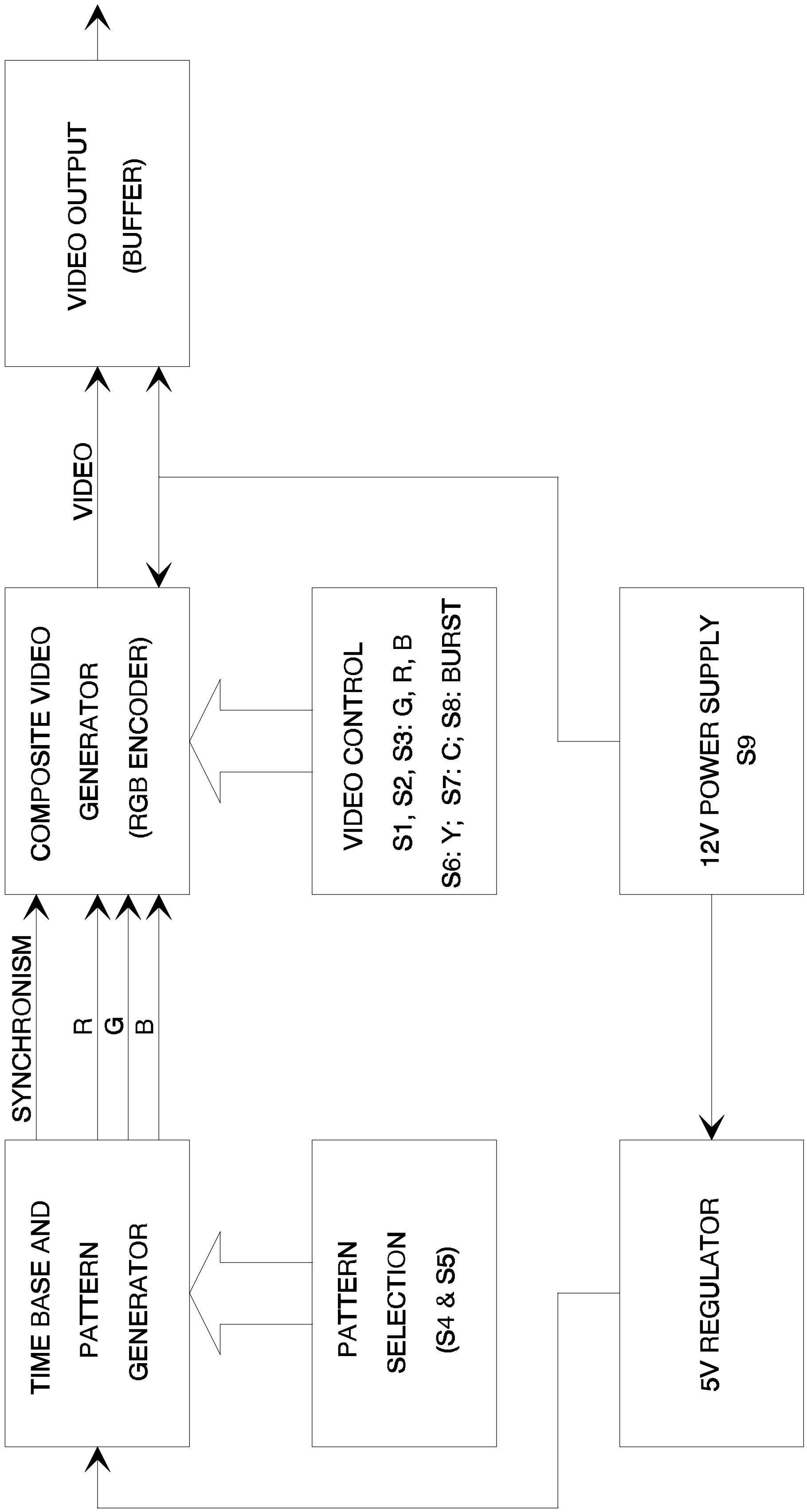
IMPORTANT ADVICE: While loading the program into the PIC, do not forget to set the option for crystal (XT) operation. Otherwise, the crystal will not oscillate.

And that’s all. If everything is correctly placed the set will run as expected from the beginning. The only adjust you may have to do is to move CV1 until you have a clear color reproduction, which is quite simple.

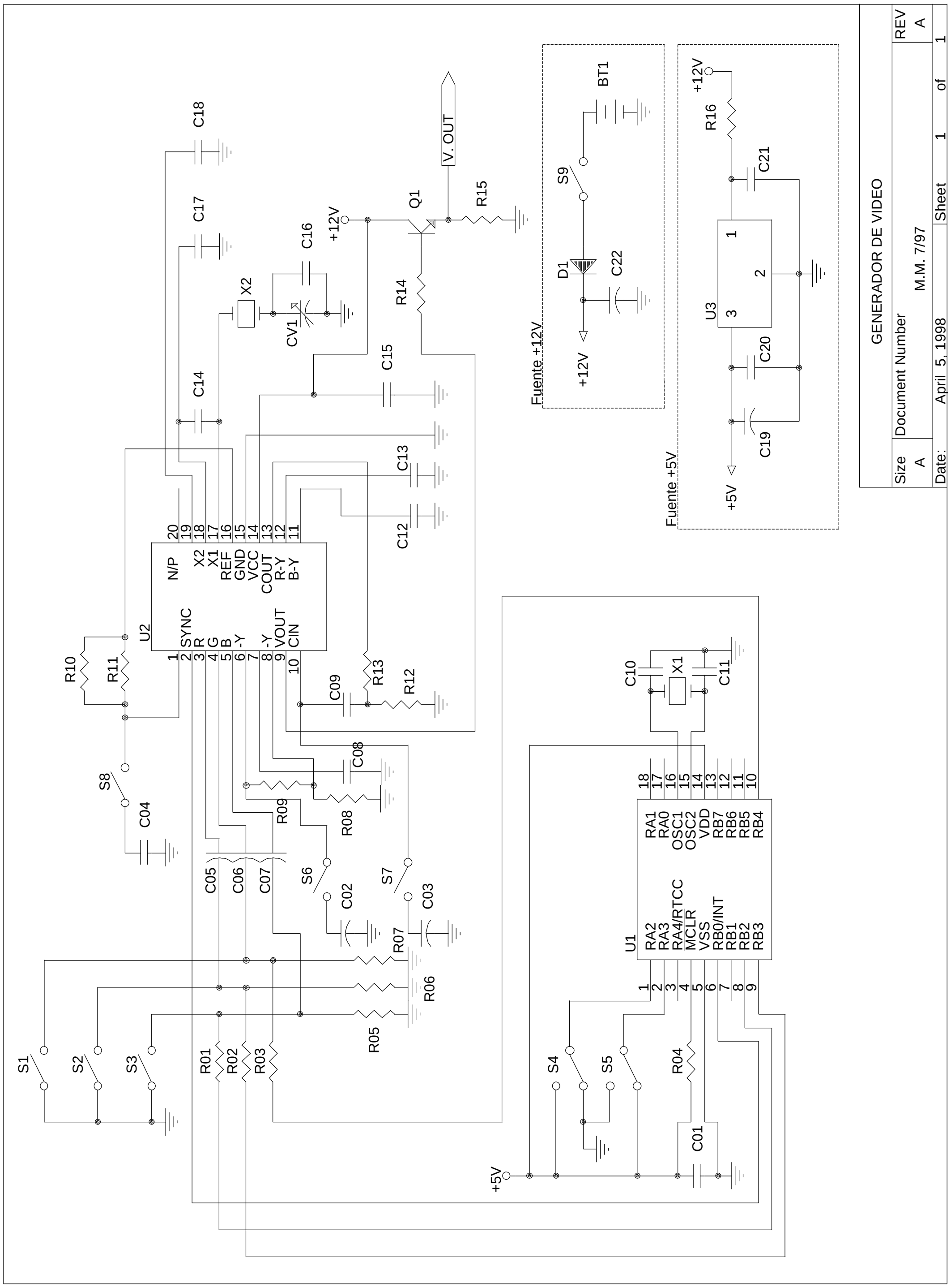
I hope this project could be of use. I’ll be expecting your comments, suggestions and also improvements you may think about. If you want a Spanish copy of this material do not hesitate to contact me.

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

Block diagram



Schematic diagram



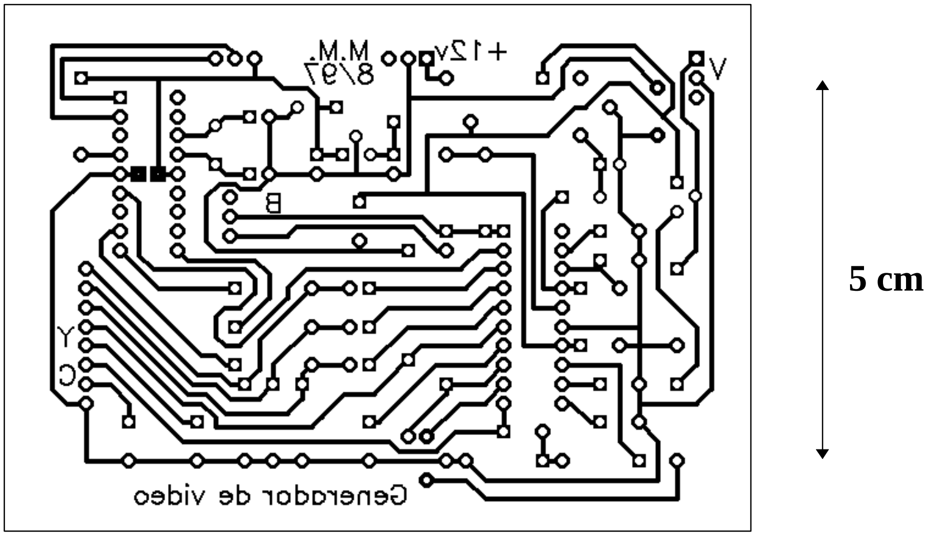
Parts list

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

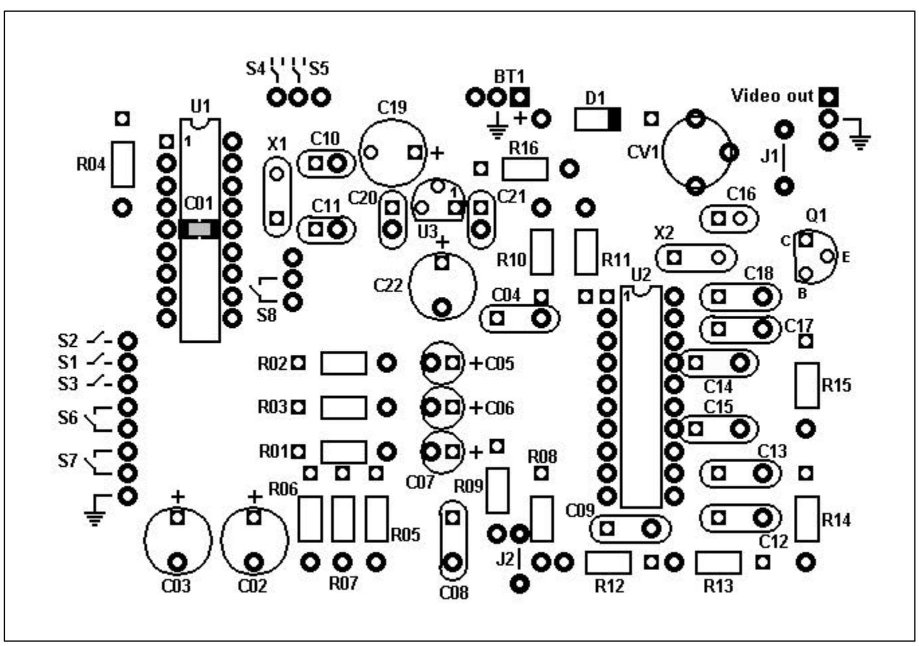
C13	0.1µ
C14	220p
C15	0.1µ
C16	18p
C17	150p
C18	.02µ
C19	100µ/16V
C20	0.1µ
C21	0.1µ
C22	100µ/16V
CV1	TRIMMER 5-45p
D1	1N4007
Q1	BF494C
U1	PIC16F84-10
U2	MC1377
U3	LM78L05
X1	10.000MHz
X2	3.582056MHz
S1	2 POSITION SWITCH
S2	2 POSITION SWITCH
S3	2 POSITION SWITCH
S4	2 POSITION SWITCH
S5	2 POSITION SWITCH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

Note: to change the color system from PAL-N to PAL-B/G/I replace X2 by a 4.43MHz crystal

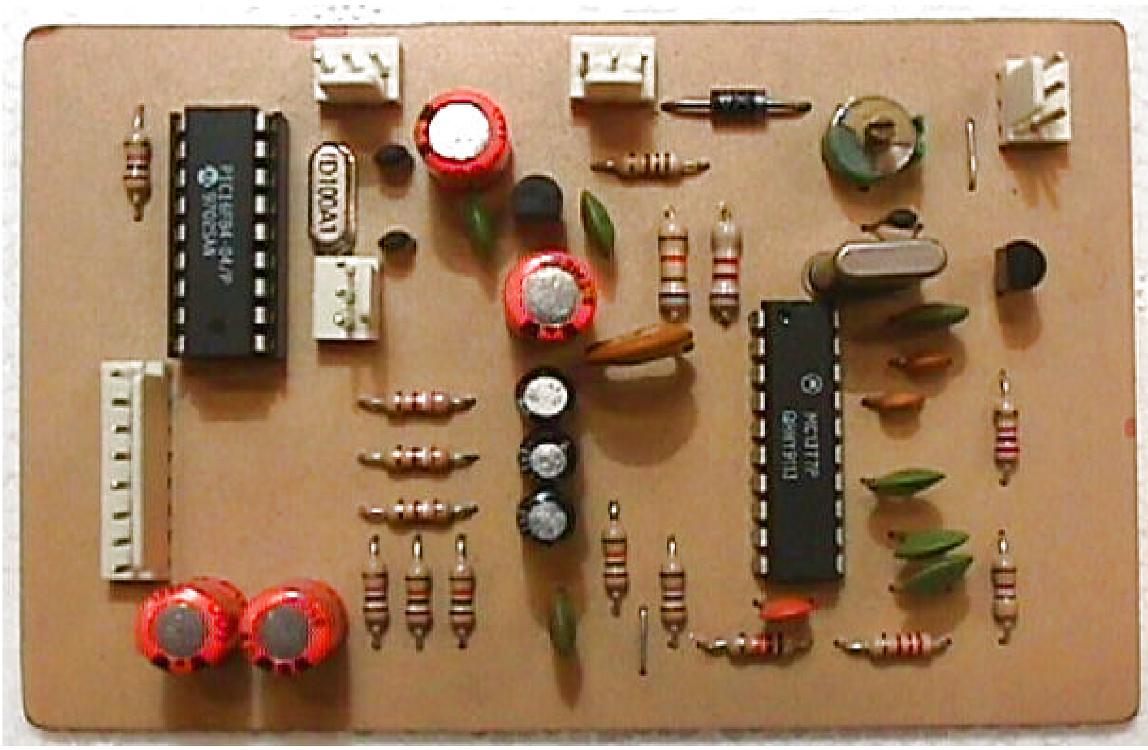
Printed circuit board



Component layout

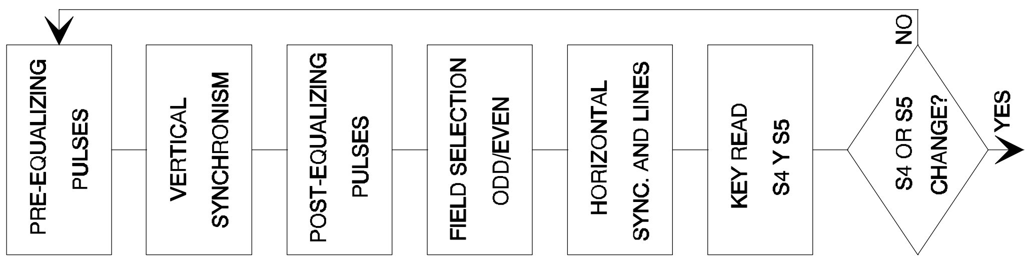


Mounted board

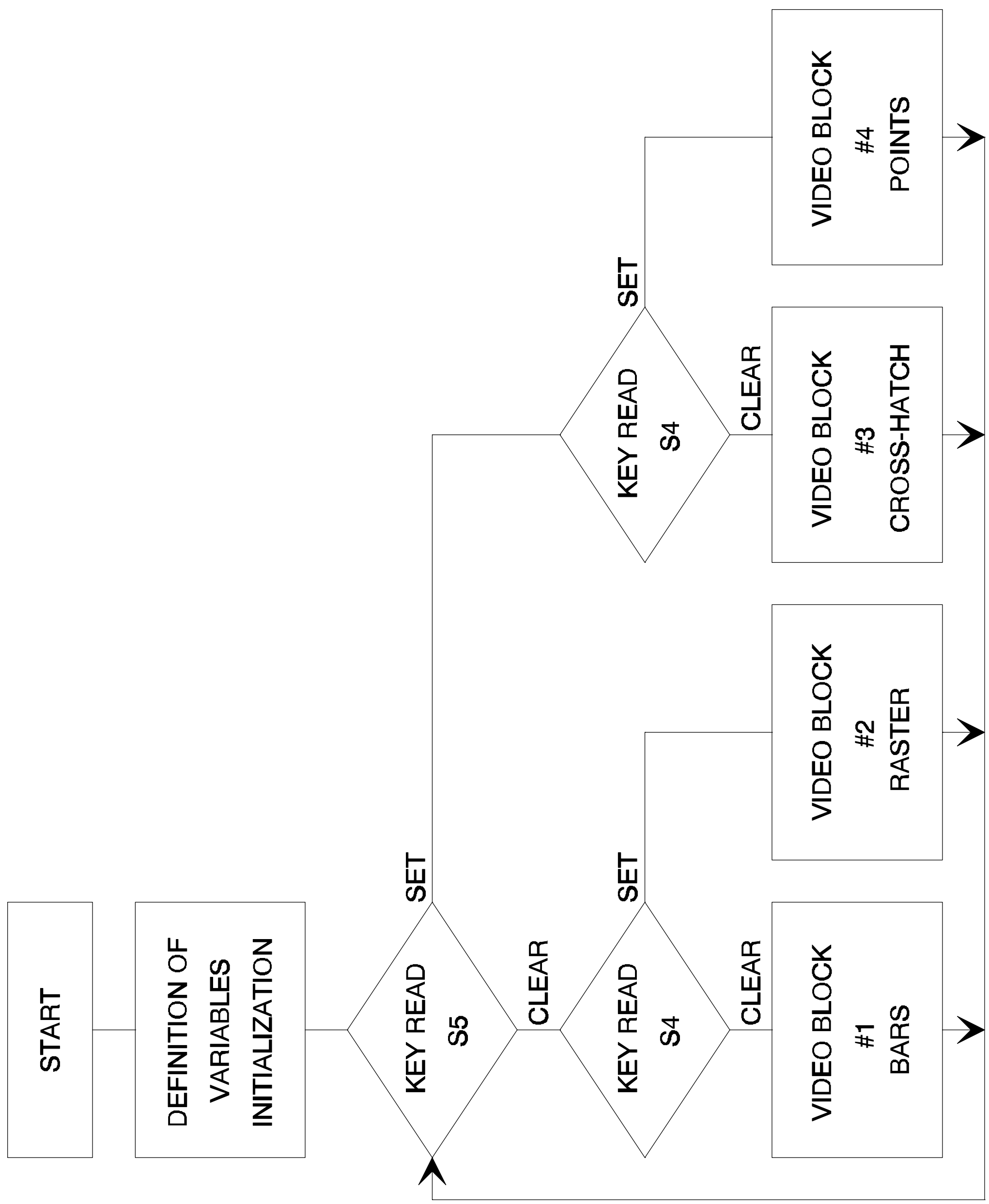


Program

Video block flowchart



General flowchart



```
; ***** 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 O
;
        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                     ;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
YELLO1  DECFSZ  YELLOW
        GOTO    YELLO1
        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                     ;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    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
        BSF     PORTB,0
ALOOP1  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
        BSF     PORTB,0
ALOOP3  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
        NOP
        MOVLW   D'21'           ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
        BTFSS   PORTB,0
        ADDLW   D'27'           ;TIEMPO PARA 1 H (160 CICLOS TOTAL)
        MOVWF   DURHOR
        BSF     PORTB,0
        BTFSS   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
        MOVLW   2
        MOVWF   TIEMPO          ;PIERDO TIEMPO PARA
ATIME3  DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    ATIME3
        NOP
        NOP
        MOVLW   D'48'
        MOVWF   DURHOR
        BSF     PORTB,0         ;BIT 0 ALTO
ALOOPH3 DECFSZ  DURHOR
        GOTO    ALOOPH3
        NOP
        DECFSZ  BLKLIN
        GOTO    AHORIZ
        NOP

AHORIZ1 BCF     PORTB,0
        MOVLW   2
        MOVWF   TIEMPO          ;PIERDO TIEMPO PARA
ATIME1  DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    ATIME1
        NOP
        NOP
        MOVLW   D'44'
        MOVWF   DURHOR
        BSF     PORTB,0         ;BIT 0 ALTO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        MOVLW   B'00011101'
        MOVWF   PORTB
ALOOPH4 DECFSZ  DURHOR
        GOTO    ALOOPH4
```

```
        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
        NOP
        MOVLW   B'00011101'
        MOVWF   PORTB
ALOOPH5 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
        MOVLW   D'21'           ;TIEMPO PARA 1/2 H (80 CICLOS TOTAL)
        NOP
        NOP
        MOVWF   DURHOR
        NOP
        NOP
        NOP
        NOP
        BCF     PORTB,1
        NOP
BLOOP   DECFSZ  DURHOR
        GOTO    BLOOP
        NOP

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

BHORIZ1 BCF     PORTB,0
        MOVLW   2
        MOVWF   TIEMPO          ;PIERDO TIEMPO PARA
BTIME1  DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    BTIME1
        NOP
        NOP
        MOVLW   9
        MOVWF   CANTLIN
        BSF     PORTB,0         ;BIT 0 ALTO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
BLOOPHA MOVLW   B'00011100'
        ADDWF   PORTB
        SUBWF   PORTB
        NOP
        MOVLW   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
        MOVLW   B'00011101'
        MOVWF   PORTB
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
        MOVLW   B'00011101'
        MOVWF   PORTB
BLOOPH5 DECFSZ  DURHOR
        GOTO    BLOOPH5
        MOVLW   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                     ;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'
        NOP
        NOP
        MOVWF   DURHOR
        BSF     PORTB,0         ;BIT 0 PASA A NIVEL ALTO
        NOP
        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
        NOP
        GOTO    INICIO2

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

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

```
CPREEQU BCF     PORTB,0         ;DURACION: 2,6µS ABAJO
        MOVLW   D'23'
        MOVWF   DUREQU
        NOP
        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
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
CLOOPHA MOVLW   B'00011100'
        ADDWF   PORTB
        SUBWF   PORTB
        NOP
        MOVLW   2
        MOVWF   DURHOR
CLOOPH4 DECFSZ  DURHOR
        GOTO    CLOOPH4
        DECFSZ  CANTLIN
```

```
        GOTO    CLOOPHA
        NOP
        MOVLW   B'00011100'
        ADDWF   PORTB
        SUBWF   PORTB
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP

        BCF     PORTB,0
        MOVLW   2
        MOVWF   TIEMPO          ;PIERDO TIEMPO PARA
CTIME2  DECFSZ  TIEMPO          ;HACER LOS 4,8µS
        GOTO    CTIME2
        MOVLW   D'28'
        MOVWF   CANTHB1
        MOVLW   9
        MOVWF   CANTLIN
        BSF     PORTB,0         ;BIT 0 ALTO
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
CLOOPHB MOVLW   B'00011100'
        ADDWF   PORTB
        SUBWF   PORTB
        NOP
        MOVLW   2
        MOVWF   DURHOR
CLOOPH5 DECFSZ  DURHOR
        GOTO    CLOOPH5
        DECFSZ  CANTLIN
        GOTO    CLOOPHB
        NOP
        MOVLW   B'00011100'
        ADDWF   PORTB
        SUBWF   PORTB
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        DECFSZ  CANTHB2
        GOTO    CHORIZ1
        NOP

;ESTA ULTIMA 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'
        NOP
        NOP
        MOVWF   DURHOR
        BSF     PORTB,0         ;BIT 0 PASA A NIVEL ALTO
        NOP
        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
        NOP
        GOTO    INICIO3

        END
```