Librería RoJoPAL

|  |  |
| --- | --- |
| Fecha | 20210201 |
| Autor | Ramón Junquera |

**Índice**

[​ Descripción 1](#__RefHeading___Toc137_2750939115)

[​ Base de la tecnología 1](#__RefHeading___Toc2258_320435842)

[​ Voltajes 2](#__RefHeading___Toc2260_320435842)

[​ Anatomía de una línea 2](#__RefHeading___Toc2262_320435842)

[​ Márgenes 3](#__RefHeading___Toc529_2729259298)

[​ Hardware 4](#__RefHeading___Toc2264_320435842)

# Descripción

RoJoPAL es una librería escrita en C++ para la familia de microprocesadores ESP32 que permite generar una señal de vídeo compuesto con protocolo PAL para conectarlo directamente a una televisión.

Todas las pruebas de esta librería se han realizado con un M5Stack Atom Lite y una televisión SHARP AQUOS LC-22LE430E. Es posible que con otros componentes varíe el comportamiento o la librería deba ser ajustada.

# Base de la tecnología

El objetivo es generar una señal analógica que cumpla el protocolo PAL.

Afortunadamente ESP32 tiene un par de pines (25 y 26) que están asignados a los dos DAC (Digital to Analog Converter) integrados.

Para controlar correctamente la frecuencia utilizaremos I2S configurado como salida y conectado a un DAC.

I2S sólo puede utilizar el canal 0 cuando trabajamos con DAC o ADC.

La máxima frecuencia que puede alcanzar I2S es de 160000000.0/3.0/2.0/2.0=13333333.333Hz=13333.333KHz=13.333MHz

Para conseguirla necesitamos:

* Configurar las interrupciones mediante registros, sobreescribiendo el valor que se define en la configuración de I2S.
* Utilizar la máxima velocidad de las CPUS. La manera más sencilla es incluyendo una referencia en el archivo platformio.ini. Es más limpio que incluir en código el comando rtc\_clk\_cpu\_freq\_set\_config\_fast

Lo que haremos será calcular cuántas muestras debe contener un línea.

Definiremos un array de ese tamaño y lo rellenaremos con los datos que corresponda antes de pedir a I2S que lo envíe al DAC.

Como el tiempo de una línea es fijo, no deberíamos tardar demasiado en componer las muestras de una línea. Sino provocaremos un error de sincronía vertical.

# Voltajes

El rango de voltajes de los DAC integrados es de 0 a 3.3V.

3.3V correspondería al máximo valor DAC (255).

Calcularemos los valores teóricos DAC de los voltajes definidos en el sistema PAL.

Añadimos una columna a la tabla con lo valores que utilizaremos en la librería, obtenidos de la práctica.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Señal** | **Variable** | **Voltaje** | **DAC teórico** | **DAC práctico** |
| Nula | signalNull | 0.000V | 255/3.3\*0.000=0 | 0 |
| Base | signalBase | 0.285V | 255/3.3\*0.285=22.022~22 | 12 |
| Negro puro | signalBlack | 0.339V | 255/3.3\*0.339=26.195~26 | 7 |
| Blanco puro | signalWhite | 1.000V | 255/3.3\*1.000=77.272~77 | 75 |

Notas:

* Por debajo de signalBlack=7 obtenemos fallos de sincronización vertical
* Por debajo de signalBase=11 obtenemos fallos de sincronización horizontal
* Por encima de signalWhite=75 obtenemos saturación cuando la imagen es grande y tiene mucho blanco.
* Aunque en teoría la señal de negro puro está por encima de la base, en la práctica el decodificador de la televisión permite intercambiarlos.
* Todas las tonalidades mostradas deben pertenecer al rango entre el negro puro y el blanco puro. Cuanto más amplio sea este rango más tonos podremos mostrar. El rango DAC teórico es de 77-26+1=52. Con los ajustes prácticos conseguimos 75-7+1=69. Hemos conseguido 69-52=17 nuevas tonalidades. Conclusión: las imágenes representadas tendrán un máximo de 69 tonalidades debido a restricciones de hardware y sistema PAL.

# Anatomía de una línea

El array que contiene las muestras de la línea que vamos a dibujar está ordenado.

Pero sabemos que hay una característica en PAL que nos obliga a intercambiar las columnas pares e impares. Esta tarea puede llevar un precioso tiempo que quizás necesitemos para no tener problemas con la sincronía vertical. Intentaremos evitar este cálculo…

Si una secuencia de muestras iguales tiene un número par de muestras, no es necesario intercambiar columnas. Podemos escribirlas secuencialmente.

Sabemos que las líneas de sincronismo vertical se componen de dos medias líneas.

Y una media línea, se compone a su vez de dos secuencias de señal.

Si una secuencia de señal debe ser par, una media línea será la suma de dos número pares, que también será par.

Dos medias líneas sumarán una línea completa que tendrá un número de muestras divisible por 4.

Teniendo en cuenta esto, comenzaremos con los cálculos.

Antes hemos definido que utilizaremos la máxima frecuencia que permite I2S: 13.333MHz.

El periodo asociado a esta frecuencia es de 1/13.333=0.075µs. Es la duración de una muestra.

Sabemos que una línea tiene una duración fija de 64µs.

Podemos calcular el número de muestras que necesitamos para componer una línea: 64/0.075=853.333~853 muestras/línea

853 no es divisible por 4. Los divisibles por 4 más próximos son 852 y 856. Tomaremos 852 por estar más cerca.

Estructura de una línea con información gráfica:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Secuencia** | **Señal** | **Duración** | **Muestras teóricas** | **Muestras prácticas** |
| 0 | signalNull | 4.7µs | 4.7/0.075=62.666~63 | 64 |
| 1 | signalBase | 5.7µs | 5.7/0.075=76 | 76 |
| 2 | imagen | 51.93µs | 51.93/0.075=692.4~692 | 690 |
| 3 | signalBase | 1.67µs | 1.67/0.075=22.2~22 | 22 |
|  |  | **64µs** | **853** | **852** |

Media señal larga:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Secuencia** | **Señal** | **Duración** | **Muestras teóricas** | **Muestras prácticas** |
| 0 | signalNull | 29.7µs | 29.7/0.075=396 | 396 |
| 1 | signalBase | 2.3µs | 2.3/0.075=30.6~31 | 30 |
|  |  | **32µs** | **427** | **426** |

Media señal corta:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Secuencia** | **Señal** | **Duración** | **Muestras teóricas** | **Muestras prácticas** |
| 0 | signalNull | 2.3µs | 2.3/0.075=30.6~31 | 30 |
| 1 | signalBase | 29.7µs | 29.7/0.075=396 | 396 |
|  |  | **32µs** | **427** | **426** |

Media señal media:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Secuencia** | **Señal** | **Duración** | **Muestras teóricas** | **Muestras prácticas** |
| 0 | signalNull | 4.7µs | 4.7/0.075=62.6~63 | 62 |
| 1 | signalBase | 27.3µs | 27.3/0.075=364 | 364 |
|  |  | **32µs** | **427** | **426** |

# Márgenes

Sabemos que en PAL, un fotograma completo se envía en dos partes. Primero sus líneas impares y después las pares.

Nosotros no descompondremos la imagen en líneas pares e impares. Nos limitaremos a mostrar el mismo fotograma tanto para líneas pares como impares.

Tendremos una tasa de refresco de 50Hz. Como contrapartida, tendremos la mitad de la resolución vertical.

Según el sistema PAL es necesario dejar márgenes verticales.

La máxima resolución vertical completa es de 576 pixels.

La máxima resolución de media imagen es de 288 pixels.

La librería consigue una resolución máxima de 218 pixels. Si lo aumentamos comenzamos con problemas de sincronismo vertical.

Se limita este tamaño. Las líneas sin información gráfica se rellenan de negro y se añaden a los márgenes.

En horizontal ocurre algo parecido. PAL no obliga a definir márgenes horizontales, pero si no lo hacemos la imagen se perderá por el marco de la televisión.

La librería limita la resolución horizontal a 653 pixels. El resto hasta los 690 pixels teóricos se rellenan con pixels negros y se añaden a los márgenes.

# Hardware

La manera más sencilla de sacar la señal de vídeo es conectar el pin DAC (por defecto el 25) y GND a un conector coaxial que vaya directamente a la televisión.

La manera más limpia es montar una interface con:

* Un M5Stack Atom Lite
* Un conector RCA
* El kit M5Stack Atomic DIY