

## Apéndices

### Apéndice A. Componentes de un sistema de procesamiento óptico

#### *El Láser*

Láser proviene del acrónimo “Light Amplification by Stimulated Emission of Radiation” (Luz amplificada por emisión estimulada de radiación). El láser es un haz de luz cuasi-monocromático, que posee una longitud de onda casi única y coherente (Andrews & Phillips, 2005).

Un láser consta de 5 elementos fundamentales: Una cavidad resonante, dos espejos, un dispositivo de bombeo y un medio activo. La cavidad resonante es el espacio por donde el haz de luz se mueve libremente, mientras rebota en dos espejos ubicados a sus extremos, uno de ellos con alta reflectancia y otro parcialmente reflector, que es por donde saldrá el láser (Muñoz, 2013).

El proceso de generación se produce a partir de los siguientes pasos:

1. Para generar un haz de luz, primero se necesita una fuente externa de energía, que funciona como estimulador. Los electrones del átomo tienden a estar en el nivel estable o más bajo de energía. Dicha fuente se encarga de aportar energía al átomo, excitando los electrones en forma de energía electromagnética por medio de descargas eléctricas o fotones. Funciona similar a una lámpara de xenón, aportando energía de forma intermitente.
2. Los paquetes de energía, se aportan al cristal de rubí, caso de un medio activo sólido, en forma de fotones. Así pues, los átomos en el cristal, visualizados en el diagrama por puntos grises claros, absorben esta energía haciendo que sus electrones salten a un nivel de energía más alto. Después de cierto tiempo, los electrones vuelven a su nivel de energía original (estado fundamental) emitiendo un fotón de luz. Esto se llama *emisión espontánea*.

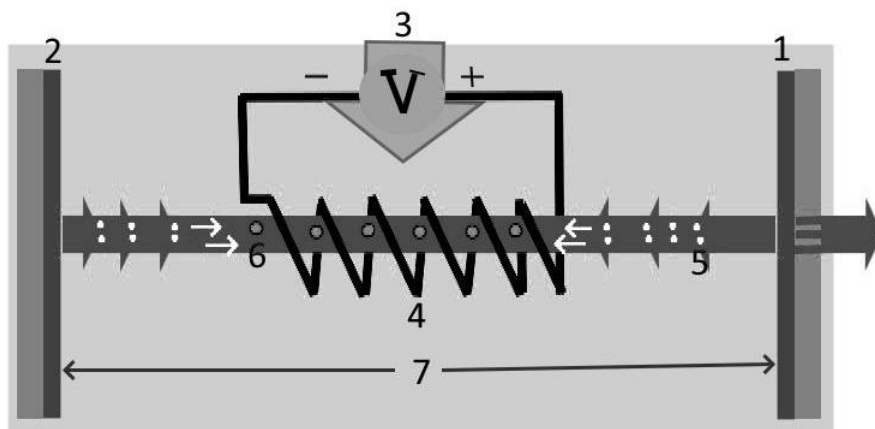
3. Los fotones emitidos por los átomos se acercan y alejan dentro del cristal de rubí (resonador), viajando a la velocidad de la luz, y es allí cuando uno de estos fotones estimula un átomo ya excitado emitiendo dos fotones: el fotón original con el que el átomo fue excitado y un fotón generado. Esto se llama *emisión estimulada*. Es de notar que un fotón de luz ha producido dos fotones, así que ésta se ha amplificado. En otras palabras, la *amplificación de luz* implica un aumento en la cantidad de fotones o luz, generada por la *amplificación de luz por emisión estimulada de radiación*, de ahí el nombre *láser*, de su acrónimo.

4. Todo este proceso sucede en medio de dos espejos ubicados en paralelo, uno con reflectancia del 100% o altamente reflectante y otro espejo parcialmente reflectante, en cada extremo del tubo láser mantiene los fotones rebotando hacia adelante y hacia atrás dentro del cristal. Así pues, el espejo parcialmente reflectante deja escapar algunos fotones que conforman el haz de láser generado para su utilización.

Las partes detalladas del láser se encuentran a continuación, en la figura 16.

### Figura 17.

*Partes del láser*



Fuente, autor.

Dichas partes corresponden a: (1) Espejo de reflectancia promedio del 95%, (2) Espejo de reflectancia cercana al 100%, (3) Fuente de energía de alto voltaje, (4) Dispositivo que “bombea” energía al cristal de rubí, (5) Fotones rebotando entre los dos espejos, (6) Átomos del cristal absorbiendo energía. Los electrones en dichos átomos saltan a un nivel de energía más alto y (7) Cavity láser o resonador.

En ocasiones dentro del resonador, hay un gas o una mezcla de gases, usados como medio activo. Estos gases se usan por algunas características como su bajo costo y su importante aporte como medio transportador de la energía en la emisión de la luz.

Dentro de los láseres gaseosos se pueden distinguir los siguientes:

- Vapor metálico: El medio activo contiene algún metal calentado hasta el vapor y son altamente utilizados en aplicaciones industriales. Un vapor metálico común es el vapor de cobre.
- Gas ionizado: Los átomos del medio activo no son neutros, sino como bien lo menciona su nombre, han perdido uno o varios electrones convirtiéndose en iones cargados positivamente (láser de argón).
- Gas molecular: El medio activo está lleno, por ejemplo, de CO<sub>2</sub>.
- Gas neutro: láser de helio-neón.

El láser utilizado durante el proyecto es de He-Ne.

El láser de helio-neón fue uno de los primeros láseres inventados y todavía se utiliza ampliamente en ramas específicas de la industria como interferometría, impresión laser y alineamiento, principalmente.

Dicho dispositivo es un láser compacto, portátil y sencillo que genera un haz con una longitud de onda de 632,8 nm. Su potencia varía desde menos de 1 mW hasta unas pocas decenas de mW y tiene una vida media larga, que puede alcanzar decenas de miles de horas de funcionamiento. El medio activo es una mezcla de helio y neón en un tubo de vidrio y la excitación se produce mediante una descarga entre un cátodo y un ánodo. El helio bombea energía al neón, creando inversión de población y generando la transición láser. La proporción usualmente es del 85% - 15%, para el helio y el neón respectivamente.

Los láseres de helio-neón emiten una luz roja y tienen una longitud de onda de 632,8 nanómetros. Son láseres de gas que utilizan la mezcla de helio y neón para producir la luz amplificada mediante la descarga de una corriente eléctrica. Debido a su potencia de salida relativamente baja, pero se usan comúnmente en interferometría, impresión láser y alineación.

En la mezcla de gases, el helio actúa únicamente para conseguir el bombeo, transfiriendo energía por colisiones al neón, siendo entre dos niveles de este último entre los que se genera la inversión de población y la transición laser. El tamaño del dispositivo puede ir de unos pocos centímetros a un par de metros, y está determinado en función de la potencia de salida del láser.

### ***El beam splitter o divisor de haz***

Un divisor de haz es un dispositivo óptico que divide la luz incidente en diferentes direcciones. Consta de dos prismas de vidrio triangulares pegados en la parte inferior de dicho dispositivo, donde el grosor de la capa de resina se adapta a una longitud de onda específica. Hay dos tipos de divisores de haz disponibles en el mercado. Uno para polarización y otro para intensidad.

Los dos tipos de divisores de haz se muestran en las figuras 17 (izq.) y 17 (der.).

**Figura 18.**

*Beam splitter de polarización (Izq.) y de intensidad (Der.)*



El beam splitter de polarización, al igual que el prisma de Wollaston, divide el haz de luz en rayos de diferente polarización ortogonal.

#### ***El atenuador o filtro de densidad neutra***

Los filtros de densidad neutra son dispositivos ópticos capaces de controlar la cantidad de energía que ingresa al sensor o lente de la cámara, lo que permite capturar imágenes en entornos muy brillantes o de alto contraste.

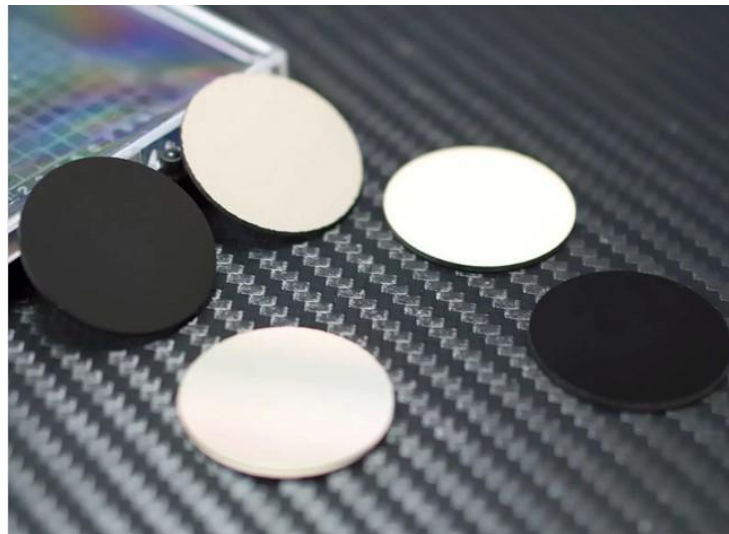
La principal característica de dicho dispositivo es la capacidad de filtrar la luz sin afectar la calidad cromática de la imagen, esto es, filtrar todos los colores por igual obteniendo una solución óptica sencilla a problemas de saturación o distorsión de colores.

Dichos dispositivos suelen tener la forma de espejos redondos de algunas pulgadas de diámetro.

Tienen una amplia gama de longitudes de onda de 400 a 2000 nm, lo que los hace adecuados para su uso con láseres de potencia moderada, tal como se muestra en la figura 18.

**Figura 19.**

*Filtro de densidad neutra.*



*Nota* Zspingzhi. (2023). *Nanomacro optics.*

<http://zspingzhi.com/chanpinzhongxin/zhaidailvguangpian/20210609/65.html>

Así pues, al elegir el filtro de densidad neutra adecuado, se consideraron factores como la potencia del filtro o el porcentaje de reducción de la luz a través de él, el tipo de sensor CMOS usado, las condiciones de iluminación específicas. Algunos filtros están diseñados para usarse bajo luz solar intensa o una alta potencia del haz de luz utilizado, mientras que otros son más adecuados para situaciones de poca potencia o exposiciones prolongadas (Andovercorp Optical, Standard & Custom Optical Filters and Coatings, 2021, pág. 31)

En el desarrollo del presente trabajo de grado se utilizó un beam splitter de intensidad (Figura 3), así pues, el rayo de luz incidente (rayo a) se divide en 2 rayos, uno reflejado (rayo b) y otro transmitido a través de la resina que une a los dos prismas (rayo c).

**Apéndice B. Microcontrolador ARM – *Córtex Mx***

La serie de núcleos ARM *Córtex* abarca una amplia gama de dispositivos con diferentes opciones de consumo y procesamiento, ofreciendo una gran variedad de opciones para elegir según los requerimientos de cada proyecto. De forma general, la familia *Córtex* se divide en tres ramas principales:

- ***Córtex-A***: Estos procesadores proporcionan una gama de soluciones para dispositivos que utilizan sistemas operativos de código abierto u open source como Linux o Android y se utilizan en una amplia gama de aplicaciones, desde teléfonos de bajo costo hasta teléfonos inteligentes.
- ***Córtex-R***: Estos procesadores son la oferta ARM más pequeña y menos conocida, en términos de derivados y aplicaciones comerciales. Los procesadores *Córtex-R* están diseñados para aplicaciones de alto rendimiento en tiempo real, como controladores de disco duro (o controladores de unidades de estado sólido), equipos de red e impresoras de tipo industrial, reproductores Blu-ray, en aplicaciones como airbags, sistemas de frenado y gestión del motor.
- ***Córtex-M***: La serie de MCU *Córtex-M* son la apuesta comercial de la empresa, por lo que son de propósito general, tienen tiempos de respuesta cortos y bajo consumo de potencia. La serie *Córtex-M* se basa en la arquitectura ARMv7-M (utilizada para *Cortex-M3* y *Cortex-M4*), y el *Cortex-M0+* (el más pequeño de esta familia) se basa en la arquitectura ARMv6-M. Los procesadores ARM *Córtex-Mx* han tenido gran relevancia en el mercado de dispositivos integrados durante los últimos 15 años (ARM, 1995). La gama de dichos productos va desde pequeños microcontroladores hasta procesadores diseñados para

aplicaciones más potentes, la ventaja principal que ofrecen los núcleos CórteX-Mx, es el desarrollo de diseños de bajo consumo y alto rendimiento en aplicaciones industriales, automotriz, de electrónica de consumo y de dispositivos capaces de conectarse a la nube, o Internet de las cosas (IoT).

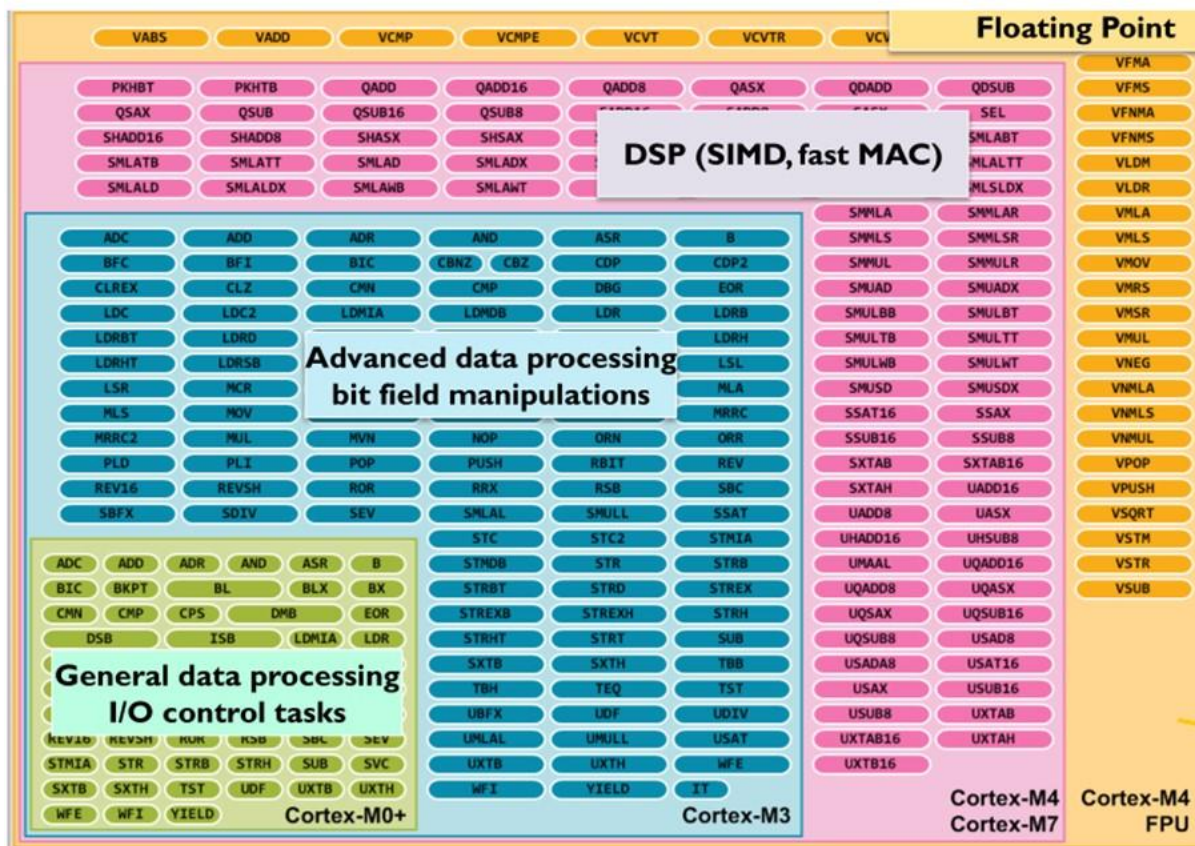
Algunas características de la arquitectura ARM son:

- El procesamiento de datos se hace en base a la unidad aritmética lógica o ALU.
- Las instrucciones se ejecutan en un solo ciclo de reloj.
- El sistema almacena las instrucciones, aumentando el rendimiento.
- El costo del dispositivo es bajo y su tamaño es pequeño, comparado con dispositivos de similares características.

A medida que la tecnología CórteX Mx iba ampliando su capacidad, su conjunto de instrucciones también debía ser ampliada. En la figura 19 se muestra la evolución del set de instrucciones, empezando con la familia M0 que incluía funciones básicas, su sucesora, la familia M3 triplica el número de instrucciones. Los dispositivos M4 también hacen un salto en la cantidad y variedad de funciones, siendo un cambio tan amplio, que el posterior M7 incluía casi la misma cantidad de funciones e instrucciones (Developer, 1995).

Figura 20.

Tipos de procesadores familia Cortex.



Nota: Instituto NCB, Conociendo el Microcontrolador ARM MSP432P401, un Cortex-M4f de Texas Instruments. Consultado en Internet el 28 de noviembre de 2023.

<https://newtoncbraga.com.mx/index.php/component/content/article?id=1722:conociendo-el-microcontrolador-arm-msp432p401-un-cortex-m4f-de-texas-instruments-mic012s>

El procesador CórteX-M7 es un procesador de alto rendimiento altamente eficiente que presenta:

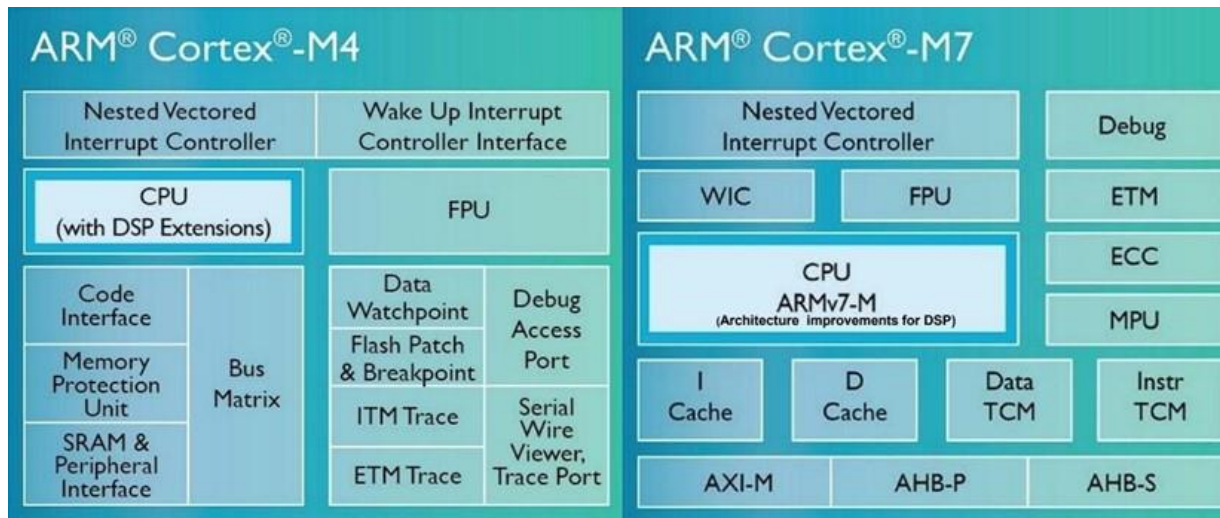
- Interfaz ITCM de 64 bits
- 2 interfaces DTCM de 32 bits
- Interfaz maestra AXI (AXIM).
- Interfaz periférica AHB-Lite (AHBP) dedicada de baja latencia

- Interfaz esclava AHB-Lite (AHBS) que proporciona acceso DMA a los TCM.

Ahora bien, las instrucciones es sólo parte de las diferencias entre los MCU, así que después de analizar la arquitectura de la familia M4 y M7, se encontró las siguientes similitudes y diferencias:

**Figura 21.**

*Comparación arquitecturas Cortex M4 y M7.*



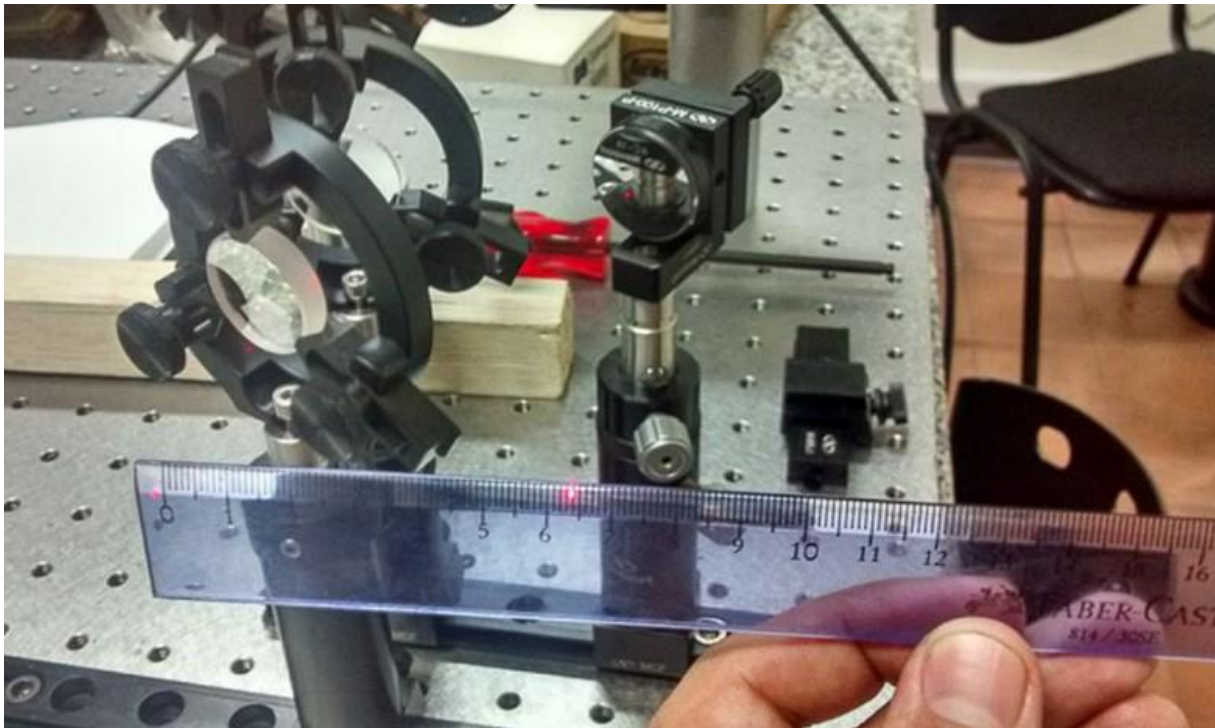
*Nota:* ARM Developer, Consultado en Internet el 28 de noviembre de 2023.

<https://developer.arm.com/documentation/102787/latest>

A partir de la Figura 20, se puede deducir que el CórteX M7 se encuentra un paso más cerca al procesamiento digital, ya que el dispositivo almacena y carga los datos en paralelo, así como reduce los bucles innecesarios. Otra diferencia son las etapas segmentadas y la arquitectura super escalar. De esta sección, se puede destacar que la elección del dispositivo usado en el proyecto de grado, el STM32F746 ha sido seleccionado, tomando en cuenta sus pares y versiones anteriores CórteX Mx, de acuerdo a los requerimientos de procesamiento, respuesta en tiempo real, precio y facilidad para portar e implementar, tomando como referencia (Reyes Resta, 2017, págs. 27-57) y (Tresanchez, y otros, 2018).

**Apéndice C. Medición de la variable  $S$  en el montaje óptico****Figura 22.**

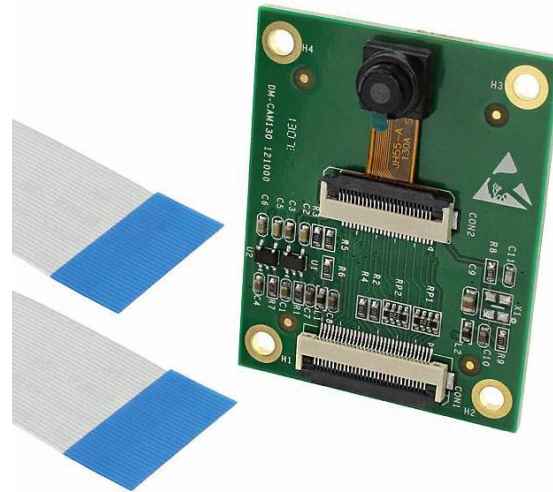
*Distancia entre espejos, montaje óptico.*



Fuente: Autor.

**Sensor de imagen**

El sensor de imagen utilizado es el OV 9655, sensor tipo CMOS de Omnivision. El sensor tiene una resolución de 1.3 Megapíxeles y está integrado en un módulo periférico (Figura 22) junto con una lente óptica, un conector de tipo FPC de 24 pines para la comunicación con el microcontrolador (MCU) que es donde va conectada la cinta flexible impresa, de color azul en la imagen. La referencia del módulo anteriormente mencionado es stm32f4dis cam.

**Figura 23.***Stm32f4dis-cam.*

*Nota:* ArduCAM. (s.f.). OV9655 1/4" CMOS UXGA (1632x1232) image sensor with OmniPixel technology. [https://www.arducam.com/downloads/modules/OV9655/ov9655\\_full.pdf](https://www.arducam.com/downloads/modules/OV9655/ov9655_full.pdf)

El sensor de imagen OV 9655 está configurado para recopilar datos en modo RGB565 (cada píxel consta de 16 bits) a 30 fotogramas por segundo y enviarlos a través de un FPC de 24 pines como un flujo constante de información. Los datos se envían un byte a la vez en una línea paralela de 8 bits junto con las señales de Sincronización Horizontal (HSYNC), Sincronización Vertical (VSYNC) y Reloj de Píxeles (PCLK), y son recibidos por el DCMI, donde se empaquetan en un registro de 32 bits llamado Registro de datos (Data Register - DR).

El módulo de memoria de acceso directo o Direct Memory access - DMA genera una solicitud de recepción de información, cada vez que la interfaz de la cámara recibe un bloque de datos completo de 32 bits en su registro de datos y se envía al DMA2, que inicia una transferencia desde el registro de datos a un búfer interno de tipo FIFO, primer registro en entrar, primer registro en salir a través del bus periférico AHB.

Este sensor de imagen tiene dos buses de comunicación:

- 1) Un bus de control de cámara en serie (SCCB - Serial Camera Control Bus) para inicializar y configurar los parámetros de la cámara
- 2) Un Bus de datos de píxeles de salida paralela de hasta 10 líneas, que transmite imágenes continuamente. El formato de salida es configurable a RGB sin formato, RGB (RGB 4:2:2, RGB 565, RGB 555), YUV(4:2:2) y YCbCr (4:2:2).

El sensor tiene una Matriz activa de 1280x1024 píxeles, pero el tamaño de la imagen es configurable, entre SXGA [1280 × 960], VGA [640 x 480], QQVGA [160 x 120], QVGA [320 x 240], CIF con un escalamiento múltiplo de 40 x 30 píxeles. Todas las configuraciones tienen un máximo de velocidad de transferencia de 30 fotogramas por segundo (30 fps), excepto SXGA que es 15 fotogramas por segundo (15 fps). El tiempo de exposición, la ganancia, el balance de blancos y negros, el filtro de banda y otros parámetros de imagen se pueden configurar a través de SCCB (Serial Camera Control Bus).

Del mismo modo, los parámetros de calidad de imagen como saturación de color, gamma, nitidez, corrección de lente, blanco, cancelación de píxeles, cancelación de ruido y detección de luminancia de 50/60 Hz, también se pueden configurar.

El sensor de la cámara funciona con diferentes fuentes de alimentación. Principalmente necesita 1,8 V si el regulador interno está desactivado. Sus líneas análogas necesitan una fuente de 2.45 a 3 V que pueda usarse para las líneas IO. Cuando la cámara está en condiciones normales de trabajo, el consumo de corriente del periférico es de alrededor de 20 mA y en modo de espera puede requerir tan solo 1  $\mu$ A.

Es importante destacar que el fabricante del módulo de la cámara (OV9655) y el del microcontrolador (STM32F746) son diferentes. Por ello, la implementación de este trabajo de

grado requirió la integración y programación de dicho periférico, comenzando a partir de un ejemplo de implementación para la pantalla LCD. Gracias a la estandarización de los protocolos de bus de campo, ahora es posible “mezclar y combinar” productos de diferentes fabricantes en una misma máquina o sistema. Además, la industria de la automatización ha adoptado Ethernet como la capa física estándar para la lectura y escritura de datos en los dispositivos (Dixson, 2011).

**Apéndice D.** *Reconocimiento de imágenes planteado por Takeda.*

Se hizo el análisis de la información contextual, mediante las características extraídas durante el reconocimiento de la imagen, donde se tomó en cuenta el trabajo realizado por Matsuo Takeda, por tanto, el análisis ha sido tomado de la referencia (Takeda, Ina, & Kobayashi, 1982).

Takeda comienza el análisis planteando el problema, basado en la importancia de la obtención de información a partir de unas franjas en una señal cuasi periódica modulada temporal o espacialmente.

Así pues, Takeda menciona que la obtención precisa y veloz de datos de franjas en una señal casi periódica es esencial, ya sea moduladas en el tiempo o en el espacio. Esto se logra mediante el empleo de métodos de generación y análisis de franjas, los cuales se dividen en dos categorías: portadoras temporales y portadoras espaciales. Las técnicas de portadora temporal, como la interferometría heterodina y la técnica del desplazamiento de fase, detectan la fase y la amplitud mediante la variación temporal periódica de franjas en una señal. Por otro lado, las técnicas de portadoras espaciales codifican la información de fase en variaciones espaciales casi periódicas de franjas en una señal, es decir su equivalente en el dominio espacial.

Takeda realiza una revisión de las técnicas de demodulación de franjas de Fourier, inicialmente desarrolladas para franjas de portadoras espaciales y aplicadas luego a una variedad más amplia de señales de franja espacio-temporal (Álvarez, y otros, 2007).

Luego el autor centra su trabajo en el análisis de franjas de Fourier (FFA) o método de transformada de Fourier (FTM). El FTM, propuesto en la década de 1980, ha sido mejorado y perfeccionado a lo largo de cuatro décadas, encontrando diversas aplicaciones. Takeda describe

sus características, fortalezas y debilidades en comparación con otras técnicas, así como algunas de sus aplicaciones prácticas y cómo se aprovechan sus ventajas en estas.

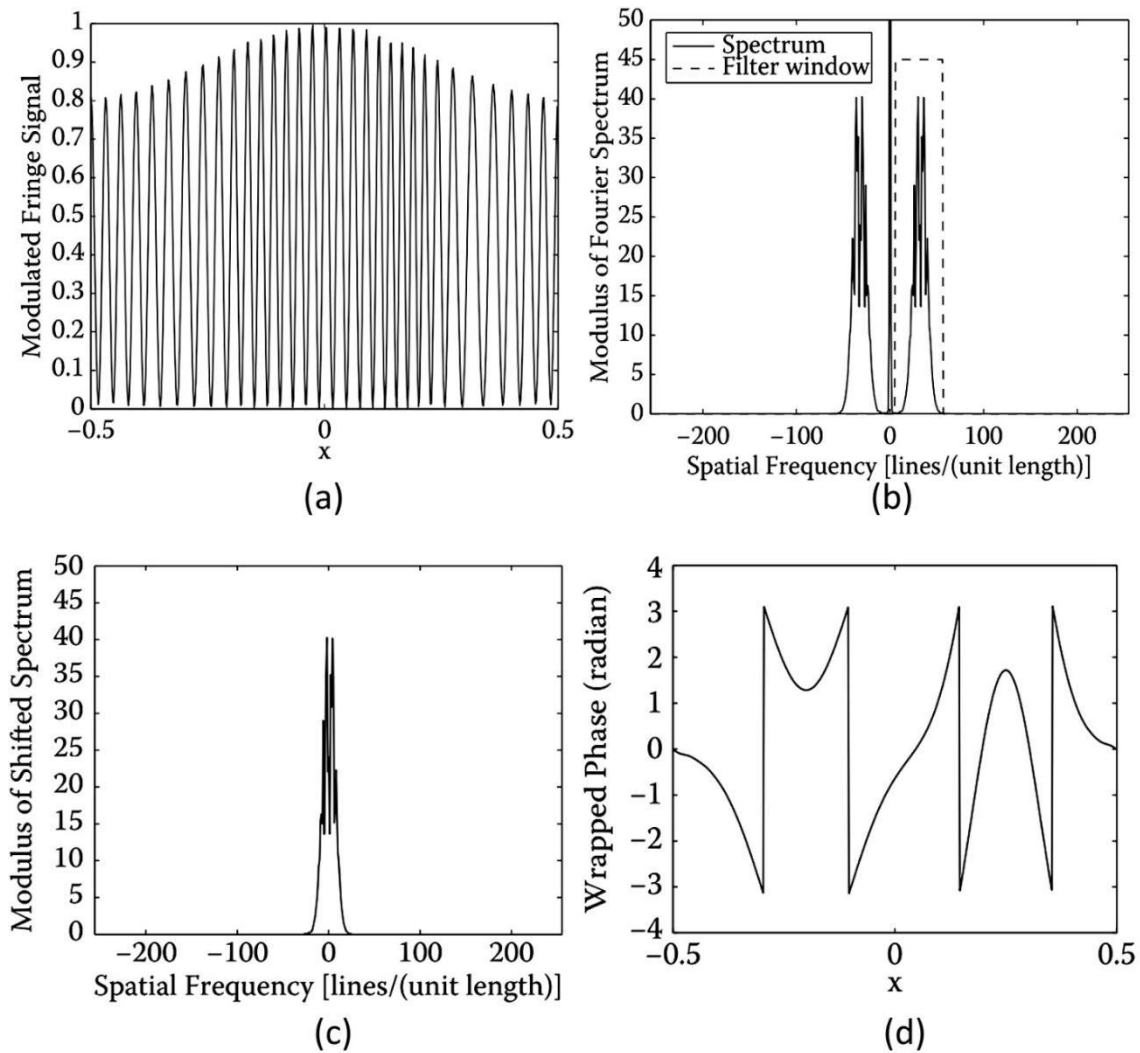
Takeda menciona que una señal con franjas, en el dominio del espacio normalizado tiene la forma

$$g(x) = a(x) + b(x) \cos [2\pi f_0 * x + \phi(x)] \quad (-0.5 \leq x \leq 0.5) \quad (3)$$

La Figura 23.a muestra una señal con franjas, implementada a partir de valores específicos en la ecuación (3) que por motivos prácticos no hay interés en mencionar. El módulo del espectro de Fourier de la señal  $G(x)$  se muestra en la Figura 23.b, así como un filtro pasa banda señalado por la línea punteada, con el que Takeda desea separar la señal envolvente de la señal a identificar, desplazando posteriormente el espectro filtrado hacia el origen para eliminar la frecuencia portadora, tal como se observa en la Figura 23.c. Por último, la figura 23.d muestra la fase envuelta, de la señal  $g(x)$  ya filtrada.

**Figura 23.**

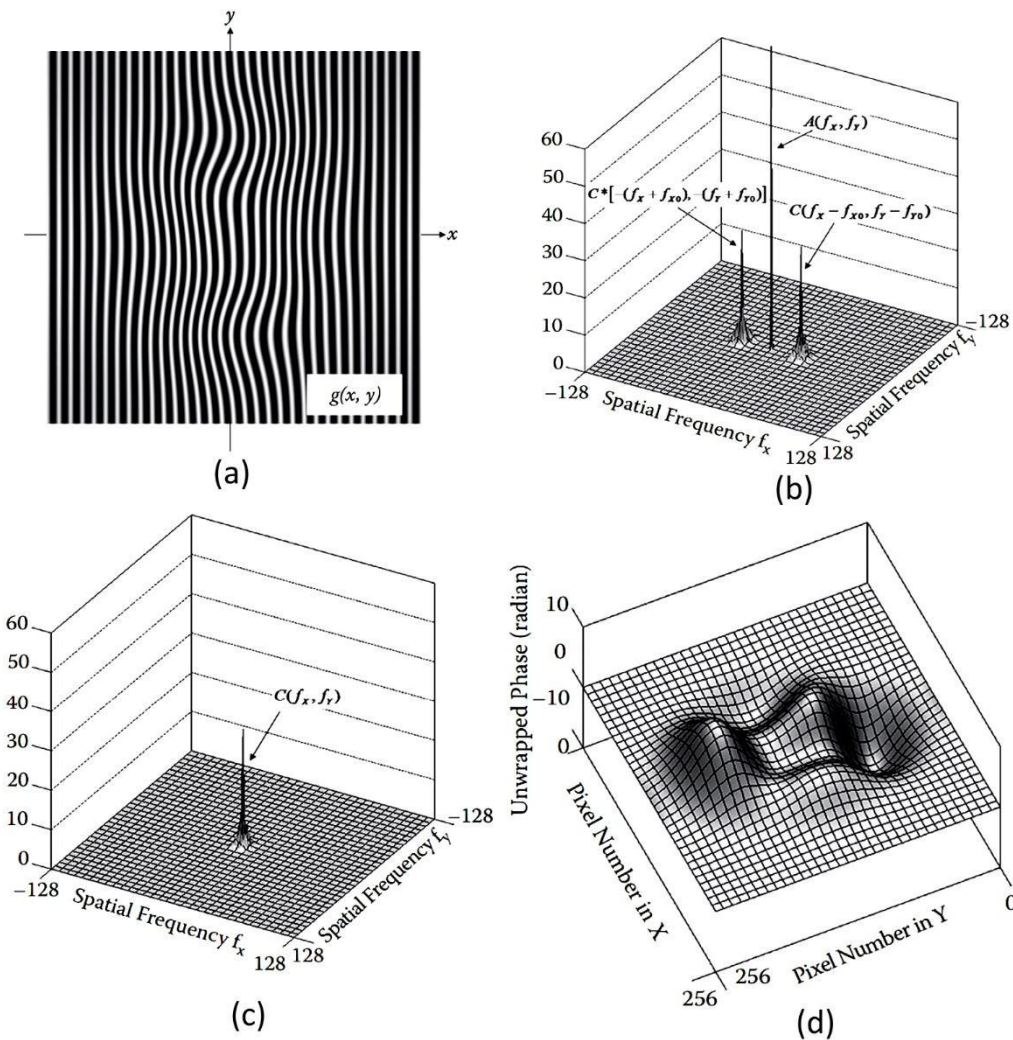
*Tratamiento de imágenes por M. Takeda,*



Nota: Takeda, Mitsuo; Ina, Hideki; Kobayashi, Seiji. (1982). Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. *Josa A*, vol. 72, no 1, p. 156-160.

**Figura 24.**

*Tratamiento de imágenes por M. Takeda,*



*Nota:* Takeda, Mitsuo; Ina, Hideki; Kobayashi, Seiji. (1982). Fourier-transform method of fringe-pattern analysis for computer-based topography and interferometry. Josa A, vol. 72, no 1, p. 156-160.

Un análisis similar es desarrollado para un interferograma  $g(x,y)$ , con una distribución de fase  $\phi(x,y)$  no ideal, tal como se muestra en la Figura 24.a. Dicha imagen busca simular un sistema de franjas no ideales, intentando acercarse a posibles resultados experimentales.

La frecuencia portadora espacial  $fx_0$  se introduce en la dirección  $x$ , con  $fy_0 = 0$ . La Figura 24.b muestra el módulo de la transformada de Fourier 2-D del interferograma. Los espectros están separados entre sí por la frecuencia portadora espacial  $fx_0$ , lo que permite un filtrado selectivo del espectro, es decir se aplica un filtro pasa banda nuevamente, al igual que con el análisis 1-D. El espectro filtrado se desplaza hacia el origen para eliminar la frecuencia portadora, como se muestra en la parte Figura 24.c, y luego se le aplica la transformada inversa de Fourier. La Figura 24.d muestra el resultado final obtenido después de desenvolverlo entre el rango  $[-\pi, \pi]$ .

Así pues, durante el presente proyecto de grado, se implementó un tratamiento de imágenes, basado en (Takeda, Ina, & Kobayashi, 1982).

1. Lectura de la imagen y opcional, selección del canal de color de interés (imread)
2. Cambio de la imagen a blanco y negro.
3. Binarización de la imagen para evitar zonas grises
4. Transformada 2D de Fourier (fft2) y posterior desplazamiento (fftshift) del componente de frecuencia cero al centro del arreglo. De aquí se deduce que hay que hacerle un corrimiento al ángulo de  $-90^\circ$ , debido a cómo Matlab hace las transformaciones.
5. Cálculo del centroide o centro de masa.
6. Se busca una copia para llevar sólo los cambios de la frecuencia fundamental al espacio directo.
7. Se le hace un filtro de Hanning al centroide, para restringir el análisis al área de interés.

### **Calculo de la desviación estándar**

Dentro del procesamiento de imágenes hecho en el presente proyecto, un punto importante fue el análisis a los ángulos de inclinación para posterior cálculo de la desviación estándar, variable de interés en el cálculo de los parámetros de la turbulencia. Dicha desviación se calculó a partir del código a continuación:

```

F_C = sprintf('Resolución de la imagen: %dX%d',M, N);%Show resolución
disp(F_C); %Anjher2
%L_orientacion = lenght(orientacion);
% Media de las orientaciones
Mean_orientacion = mean(orientacion);
F_D = sprintf('Media en radianes: %d radianes', Mean_orientacion);
disp(F_D);
Orient_grados = rad2deg(Mean_orientacion);
disp(['Media de los ángulos en grados: ', num2str(Orient_grados)]);
figure(15); plot(orientacion); colormap('gray')
title(['Angulo de inclinación en cada línea'])

%Cálculo de la desviación estándar
diferenciasAlCuadrado = (orientacion - Mean_orientacion).^2;
varianza = mean(diferenciasAlCuadrado);
desviacionEstandar = sqrt(varianza);
disp(['La desviación estándar 1 del patrón de franjas es: ', num2str(desviacionEstandar)]);

desviacionEstandar2 = std(orientacion);
disp(['La desviación estándar 2 del patrón de franjas es: ', num2str(desviacionEstandar2)]);

```

### Generación de franjas interferométricas Matlab

```

clc
clear all
close all

Nx = 20;
Ny = 20;
Res_x = 320;
Res_y = 240;

%DeltaX = 3e-6;
DeltaX = 0.15; %Valores entre 0.1 y 0.2 son válidos, spacial frequency

% Definir el ángulo A en radianes y la longitud X
A = 90; %Angulo en grados
A_R = deg2rad(A); % Ángulo A en radianes

% Calcular la longitud Y y posterior DeltaY
DeltaY = DeltaX * tan(A_R);

x = -Nx/2:Nx/Res_x:Nx/2; % X- Axis
y = -Ny/2:Ny/Res_y:Ny/2; % Y- Axis

[X Y] = meshgrid(x,y);

```

```
Tx = (2*DeltaX);           % Fringes period along X-axis
Ty = (2*DeltaY);           % Fringes period along Y-axis

Fx = ((1)/Tx);             % #frinjes/longitude -- Adjustable Espatial Frequency X-Axis
Fy = ((1)/Ty);             % #frinjes/longitude -- Adjustable Espatial Frequency Y-Axis

fxy = 2.*cos(2*Fx.*X + 2*Fy.*Y); % Objective Function
figure(1); imshow(fxy);

%%%%%% Anjher test franjas sin inclinación %%%
clear all
clc
%Rejilla con dimensiones 1600x1200
x = linspace(-0.01, 0.01, 320); %distancia, 1600 pix in X
% T = 0.8E-3 %Periodo de la rejilla de difraccion
T = 0.25E-2 %Periodo de la rejilla de difraccion
y = cos(6*pi*(1/T)*x);
y = y >= 0;
y = repmat(y,240,1); %Las dimensiones se hacen con linspace y repmat
imshow(y);
```