

# **Módulo educativo para el aprendizaje en procesamiento de imágenes**

Especificación de casos de uso

Pablo Jesús Carrizo  
([pablo.carrizoj@gmail.com](mailto:pablo.carrizoj@gmail.com))

23/10/2021

Versión A

## Historial de cambios

<b>Versión</b>	<b>Fecha</b>	<b>Descripción</b>	<b>Autor</b>	<b>Revisores</b>
A	23/10/21	Versión Original	Pablo Jesús Carrizo	

## Índice de contenido

1. Introducción	4
1.1. Propósito	4
1.2. Ámbito del Sistema	<b>Error! Bookmark not defined.</b>
1.3. Definiciones, Acrónimos y Abreviaturas	4
1.4. Referencias	4
2. Descripción General	5
2.1. Perspectiva del Producto	5
2.2. Funciones del Producto	5
3. Casos de uso	<b>Error! Bookmark not defined.</b>
3.1. Captar imágenes en tiempo real	6
3.2. Comunicación con terminal (PC) vía USB	7
3.3. Uso de librerías y aplicaciones	7
3.4. Manejo de motores PAP	8
4. Apéndices	9

# 1. Introducción

## 1.1. Propósito

1. Este documento representa una especificación de casos de uso para un módulo educativo en procesamiento de imágenes.
2. Está dirigido a desarrolladores que se ocupen del análisis, diseño e implementación en procesamiento de imágenes.

## 1.2. Definiciones, Acrónimos y Abreviaturas

1. ARM      Advanced RISC Machine (máquina RISC avanzada)
2. N/A      No aplica
3. PAP      Paso a Paso
4. MCU      Microcontroller Unit (Unidad de microcontrolador)

## 1.3. Referencias

1. MEPAPI-ER-0002      Especificación de requerimientos de sistema para un módulo educativo para el aprendizaje en procesamiento de imágenes.

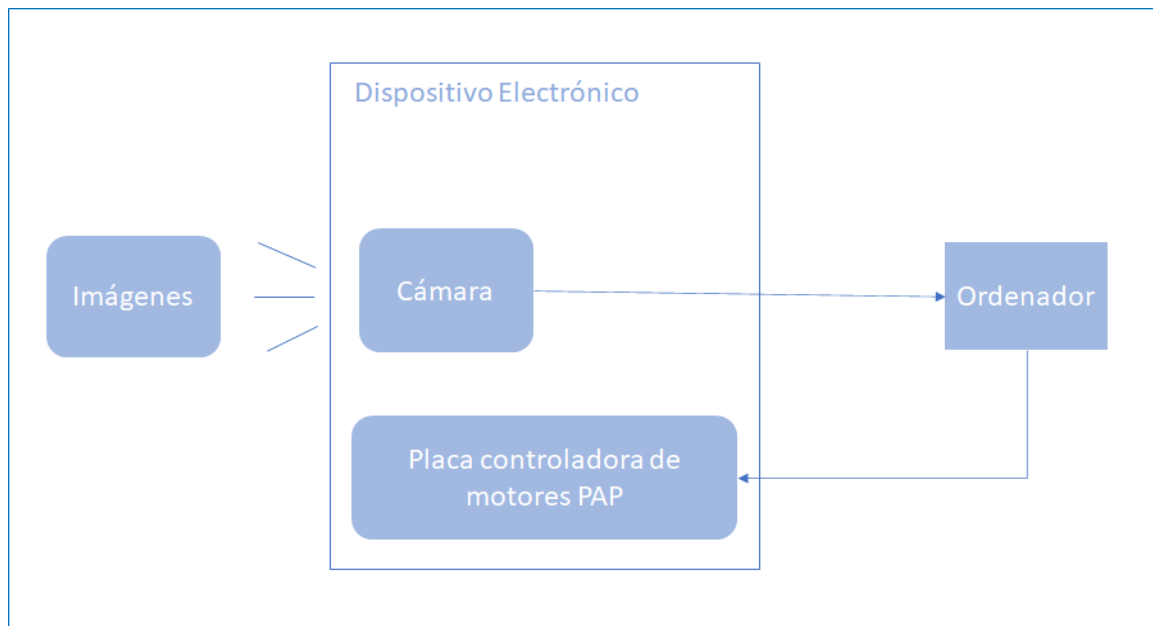
## 1.4. Visión General del Documento

1. Este documento se realiza siguiendo el estándar IEEE Std. 830-1998

## 2. Descripción General

### 2.1. Perspectiva del Producto

1. El módulo aquí especificado deberá captar imágenes en tiempo real y transmitir las a un ordenador. En su función secundaria debe permitir el movimiento en dos ejes. Para ello, la placa controladora de la cámara propuesta es OpenMV Cam H7 plus y la placa controladora de los motores estará formada por un microcontrolador y un array de transistores.



### 2.2. Funciones del Producto

1. El módulo aquí especificado brindará las siguientes funcionalidades.
  - a) Captar imágenes en tiempo real
  - b) Comunicación con ordenador para su posterior procesamiento
  - c) Proveer un entorno de desarrollo para el procesamiento digital de las imágenes
  - d) Transmitir datos desde el ordenador hacia el módulo para el control de los motores
2. El producto aquí especificado no brindará los servicios de
  - a) Comunicación inalámbrica con el ordenador

### 3. Casos de uso

#### 3.1. [MEPAPI-ECU-001-CU001] Captar imágenes en tiempo real

1. Captar imágenes en tiempo real	
1.1 Descripción	1.1.1. La cámara del módulo deberá ser capaz de tomar imágenes de 640x480 8 bit en escala de grises o 640x480 8 bit imágenes Bayer a 40 FPS.
1.2 Actor Principal	1.2.1. Profesores y alumnos de la universidad.
1.3 Disparadores	1.3.1. El evento se inicia en cuanto se habilita transmisión de datos en la toma de imágenes.
2. Flujo de eventos	
2.1 Flujo básico	2.1.1. Se energiza el zócalo
	2.1.2. Se energiza microcontrolador
	2.1.3. Firmware de MCU STM32H7 inicializa sensor de cámara
	2.1.4. Microcontrolador inicializa comunicación con CPU
	2.1.5. Se envía flujo de imágenes.
2.2 Flujo alternativo	2.2.1 En caso de fallas, no hay inicialización de sensor y se envía alerta a la terminal indicando que se encuentra en modo de fallo.
3. Requerimientos especiales	
	3.1. Para un funcionamiento con bajo ruido del sensor de imágenes, MT9M114, requiere fuentes de alimentación separadas para secciones analógicas y digitales.
	3.2. Es importante proveer de capacitores para desacoplar ruido. Hoja de datos del sensor MT9M114 provee las recomendaciones óptimas para lograr el desacoplado.
4. Pre-condiciones	
	4.1. El sistema no debe estar en estado de falla
	4.2. Sensor de imágenes montado en placa (el diseño permite montar multiples sensores de camaras)
5. Post-condiciones	
	5. 1. Desde Open MV se deberá finalizar la sesión de captura de imágenes.

### 3.2. [MEPAPI-ECU-001-CU002] Comunicación con terminal (PC) vía USB

2. Comunicación con terminal (PC) vía USB	
1.1 Descripción	1.1.1. Interfaz mediante USB (12Mbs) de velocidad completa para conexión con la computadora.
1.2 Actor Principal	1.2.1. Profesores y alumnos de la universidad.
1.3 Disparadores	1.3.1. El evento se inicia en cuanto se habilita transmisión de datos en la toma de imágenes.
2. Flujo de eventos	
2.1 Flujo básico	2.1.1. Energización de CPU
	2.1.2. Conexión de placa a puerto USB
	2.1.3. OpenMV Cam aparecerá como un puerto COM virtual y una unidad flash USB cuando se conecte.
	2.1.4. Microcontrolador inicializa comunicación con CPU
	2.1.5. Se envía flujo de imágenes.
2.2 Flujo alternativo	2.2.1 En caso de fallas, no hay inicialización de sensor y se envía alerta a la terminal indicando que se encuentra en modo de fallo.
3. Requerimientos especiales	
	N/A
4. Pre-condiciones	
	4.1. El sistema no debe estar en estado de falla.
	4.2. Disponibilidad de conexión USB Full speed.
5. Post-condiciones	
	N/A

### 3.3. [MEPAPI-ECU-001-CU003] Uso de librerías y aplicaciones

3. Uso de librerías y aplicaciones	
1.1 Descripción	1.1.1. La plataforma OpenMV Cam viene incorporada con una biblioteca RPC (Remote Python / Procedure Call) que facilita la conexión de OpenMV Cam mediante librerías de implementación.
1.2 Actor Principal	1.2.1. Profesores y alumnos de la universidad.

1.3 Disparadores	1.3.1. El evento relacionado a las necesidades de implementación de aplicaciones en la práctica de la cátedra. Estas están relacionadas con la conexión de OpenMV Cam a su computadora, una SBC (computadora de placa única) como RaspberryPi o Beaglebone, o un microcontrolador como Arduino o ESP8266 / 32.
2. Flujo de eventos	
2.1 Flujo básico	2.1.1. Se energiza el zócalo
	2.1.2. Se energiza microcontrolador
	2.1.3. Firmware de MCU STM32H7 inicializa sensor de cámara
	2.1.4. MCU inicializa aplicación y los protocolos de comunicación según los periféricos a ser usados por la aplicación.
	2.1.5. Inicio y desarrollo de aplicaciones.
2.2 Flujo alternativo	2.2.1 En caso de fallas, no hay inicialización de sensor y se envía alerta a la terminal indicando que se encuentra en modo de fallo.
3. Requerimientos especiales	
	N/A
4. Pre-condiciones	
	4.1. El sistema no debe estar en estado de falla
	4.2. Es necesario descargar las librerías para el desarrollo de aplicaciones.
5. Post-condiciones	
	N/A

#### [MEPAPI-ECU-001-CU004] Manejo de motores paso a paso

4. Manejo de motores PAP	
1.1 Descripción	1.1.1. El módulo hará uso de dos motores PAP para el movimiento en dos ejes, permitiendo flexibilidad en las aplicaciones de <b>tracking (seguimiento)</b>
1.2 Actor Principal	1.2.1. Profesores y alumnos de la universidad.
1.3 Disparadores	1.3.1. El evento relacionado a las necesidades de implementación de aplicaciones en la práctica de la cátedra. Estas están relacionadas con la aplicación implementada ( <b>Line detection, Eye Tracking, Color Tracking,...</b> )
2. Flujo de eventos	
2.1 Flujo básico	2.1.1. Se energiza el zócalo y MCU
	2.1.2. Se descarga firmware de tracking en microcontrolador
	2.1.3. Se inicializa los Motores PAP
	2.1.4. Inicio de aplicación en el IDE de Open MV



2.2 Flujo alternativo	2.2.1 En caso de fallas, no hay inicialización de Motores PAP y se envía alerta a la terminal indicando que se encuentra en modo de fallo.
3. Requerimientos especiales	
	3.1. Se deberá descargar firmware (driver) para el manejo de los Motores PAP.
4. Pre-condiciones	
	4.1. El sistema no debe estar en estado de falla
	4.2. Conectar el módulo de potencia de los motores PAP a la placa Open MV
	4.3. Los motores PAP deben estar conectados a la placa de potencia
	4.4. Es necesario descargar las librerías para el desarrollo de aplicaciones.
5. Post-condiciones	
	N/A

## 4. Apéndices

1. N/A