



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y
REDES INDUSTRIALES

**“DISEÑO EXPERIMENTAL DE UN DISPOSITIVO QUE POR
MEDIO DE UNA FUENTE DE RADIACIÓN INFRARROJA
EXTERNA PERMITA VISUALIZAR UNA IMAGEN TÉRMICA
DE UN OBJETIVO”**

TRABAJO DE TITULACIÓN
TIPO: DISPOSITIVO TECNOLÓGICO
Presentado para optar al grado académico de:
INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTOR: LUIS GONZALO MALDONADO GARCIA

TUTOR: ING. PABLO LOZADA

Riobamba-Ecuador

2018

2018©, Luis Gonzalo Maldonado Garcia

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERIA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El tribunal de Trabajo de Titulación certifica que: El trabajo de investigación: DISEÑO EXPERIMENTAL DE UN DISPOSITIVO QUE POR MEDIO DE UNA FUENTE DE RADIACIÓN INFRARROJA EXTERNA PERMITA VISUALIZAR UNA IMAGEN TÉRMICA DE UN OBJETIVO, de responsabilidad del señor Luis Gonzalo Maldonado Garcia, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

Ing. Washington Gilberto Luna E.

**DECANO DE LA FACULTAD
DE INFORMÁTICA
Y ELECTRÓNICA**

Ing. Freddy Enrique Chávez V.

**DIRECTOR DE ESCUELA DE
INGENIERÍA ELECTRÓNICA
EN CONTROL Y REDES
INDUSTRIALES**

Ing. Pablo Eduardo Lozada Y.

**DIRECTOR DEL TRABAJO DE
TITULACIÓN**

Ing. Jesús Alberto Rodríguez F.

MIEMBRO DE TRIBUNAL

Yo, Luis Gonzalo Maldonado Garcia soy responsable de todos los resultados, ideas y modos expuestos en este Trabajo de Titulación y del patrimonio intelectual del trabajo de titulación pertenece a la Escuela Superior Politécnica de Chimborazo.

LUIS GONZALO MALDONADO GARCIA

DEDICATORIA

El presente trabajo de titulación lo dedico a Dios por darme la paciencia y las fuerzas para seguir adelante a pesar de las adversidades que me ha puesto la vida. A mis padres Miriam Garcia y Serafin Maldonado quienes además se ser un apoyo económico y moral, sus éxitos y errores me han servido como guía. A la Lic. Mónica Mora y Paulina Mora quienes han formado parte de mi familia. A mis hermanos Jessica, Ricardo, Nathaly, Kristel, Alison y mis sobrinos a pesar de la distancia y otras circunstancias ellos han sido mi motivación para seguir adelante, ellos hicieron que todo este sacrificio haya valido la pena. A mis abuelos y tíos ya que gracias con su apoyo y enseñanzas pude entender el significado del trabajo duro y sus recompensas. Gracias a todos ellos pude cumplir la meta de convertirme en un profesional.

Luis

AGRADECIMIENTO

A mi familia por ser el pilar fundamental en todo lo que soy, por siempre estar a mi lado, darme su apoyo y sobre todo por confiar en mí, a mis maestros, quienes de una forma u otra supieron transmitirme sus conocimientos que me servirán en mi vida profesional. A mis amigos, a los que durante el transcurso de mi vida tuve el privilegio de conocerlos, ellos se convirtieron en mi segunda familia, su convivencia, experiencias y sabiduría me ayudaron en parte a convertirme en la persona que soy ahora. Además, quiero agradecer a la Ing. Noemi Ovando, si en algún momento llego a prescindir de ella quiero darle las gracias por su amistad, paciencia y apoyo ya que el desarrollo de este trabajo ha sido posible gracias a ella.

Luis

ÍNDICE GENERAL

INDICE DE ANEXOS	xi
ÍNDICE DE TABLAS.....	xii
ÍNDICE DE FIGURAS.....	xiii
ABREVIATURAS.....	xv
RESUMEN	xviii
ABSTRACT.....	xix
INTRODUCCIÓN	1
CAPÍTULO I	
1. MARCO TEÓRICO CONCEPTUAL.....	6
1.1 Termografía.....	6
1.1.1 Definición	6
1.1.2 Emisividad, Reflexión y Transmisión	6
1.2 El Espectro Electromagnético	7
1.3 Onda.....	7
1.4 Calor y Temperatura	9
1.5 Sensor:.....	10
1.6 Cámara Digital	10
1.6.1 Funcionamiento	10
1.7 Cámara Termográfica.....	10
1.8 Imagen Radiométrica.....	11
1.8.1 Importancia de la Sensibilidad (Resolución)	11
1.9 Pixel, FOV y IFOV	11
1.10 Reglas para capturar una Imagen Térmica	12
1.11 Visión Artificial	13
1.11.1 Definición	13
1.11.2 Aplicaciones	13

1.11.3	<i>Niveles de la visión artificial</i>	14
1.11.4	<i>Etapas de un sistema de visión artificial</i>	14
1.11.5	<i>Componentes de un sistema de visión artificial</i>	15
1.12	Raspberry Pi	16
1.12.1	<i>Definición</i>	16
1.12.2	<i>Características</i>	16
1.12.3	<i>Distribución de Pines</i>	18
1.12.4	<i>Cámaras Compatible con Raspberry Pi</i>	19
1.12.5	<i>Dispositivos de Almacenamiento compatible con Raspberry Pi</i>	22
1.12.6	<i>Pantalla y teclado compatible con Raspberry Pi</i>	23
1.13	Buzzer	23
1.13.1	<i>Definición</i>	24
1.14	Entornos de Programación	24
1.14.1	<i>Python</i>	24
1.14.1.1	<i>Definición</i>	24
1.14.1.2	<i>Características</i>	24
1.14.2	Qt	26
1.14.2.1	<i>Definición</i>	26
1.14.2.1	<i>Características</i>	26
 CAPÍTULO II		
2.	MARCO METODOLÓGICO	27
2.1	Componentes del Dispositivo	27
2.2	Selección del Hardware	28
2.3	Selección del Software	29
2.4	Desarrollo del Algoritmo	30
2.4.1	<i>Instalación del Software necesario en la Raspberry Pi</i>	30
2.4.1.1	<i>Instalación del Sistema Operativo</i>	30
2.4.2	<i>Consideraciones para el diseño del dispositivo</i>	31

2.4.2.1	<i>Área de Trabajo</i>	31
2.4.2.2	<i>Objetivos a Detectar</i>	31
2.4.2.3	<i>Variables de Entrada</i>	32
4.1.2.4	<i>Variables de Salida</i>	32
4.1.2.4	<i>Funciones de la Interfaz</i>	32
2.4.3	<i>Diseño de la Interfaz Gráfica</i>	33
2.4.4	<i>Diseño de Códigos de los Entornos de Programación</i>	35
2.4.4.1	<i>Diseño del Algoritmo</i>	35
2.4.4.1	<i>Desarrollo del Código para el Algoritmo</i>	37
2.5	Implementación del Dispositivo	41
 CAPÍTULO III		
3.	PRUEBAS Y RESULTADOS	44
3.1	Pruebas de envío y adquisición de datos	44
3.1.1	<i>Interfaz I²C</i>	44
3.1.2	<i>Puertos GPIO</i>	44
3.2	Ajustes de las Variables	45
3.3	Configuración de las Componentes	46
3.4	Pruebas de Funcionamiento	47
3.4.1	<i>Prueba Inicial de Funcionamiento</i>	47
3.4.1.1	<i>Visualización de la Imagen y Temperatura</i>	47
3.4.1.2	<i>Almacenamiento de la Imagen</i>	48
3.5	Errores en Lectura de Datos	48
3.5.1	<i>Visualización de las Variables</i>	48
3.5.2	<i>Uso de la Pantalla HDMI</i>	49
3.6	Corrección y Ajustes de Parámetros	49
3.6.1	<i>Pantalla</i>	49
3.6.2	<i>Actualización de los Datos</i>	50
3.6.3	<i>Escala y Datos de la Imagen</i>	51

3.6.4	<i>Recursos Adicionales</i>	51
3.7	Resultados del Proyecto	52
3.7.1	<i>Dispositivo Implementado</i>	52
3.7.2	<i>Interfaz Gráfica</i>	53
3.7.3	<i>Pruebas Finales</i>	53
3.7.2.1	<i>Área Doméstica</i>	53
3.7.2.2	<i>Área Industrial</i>	54
3.7.2.3	<i>Objetivos a Corta Distancia</i>	55
3.7.2.4	<i>Objetivos a Larga Distancia</i>	56
3.8.3	<i>Análisis de Resultados</i>	57
	CONCLUSIONES	58
	RECOMENDACIONES	59
	GLOSARIO	
	BIBLIOGRAFÍA	
	ANEXOS	

INDICE DE ANEXOS

ANEXO A: Instalación del Sistema Operativo Raspbian en la Tarjeta Micro-SD

ANEXO B: Configuración del Sistema Operativo Raspbian

ANEXO C: Instalación de Programas Necesarios

ANEXO D: Instalación de Librerías

ANEXO E: Prueba de Funcionamiento de Cámaras

ANEXO F: Configuración de Qt Creator

ANEXO G: Nuevo Proyecto en Qt Creator

ANEXO H: Valores almacenados

ÍNDICE DE TABLAS

Tabla 1-1: Aplicaciones de la Visión Artificial	13
Tabla 2-1: Características de la Raspberry Pi 3B	16
Tabla 3-1: Características Cámara Raspberry Pi Camera Board V2	19
Tabla 4-1: Características Cámara Radiométrica FLIR Lepton 2.5	20
Tabla 5-1: Clases de Tarjetas SD según su clase	23
Tabla 1-2: Conexión Cámara Flir Lepton a Raspberry	41
Tabla 2-2: Conexión Módulo Buzzer a Raspberry	42
Tabla 1-3: Valores Máximo y Mínimo Detectados	45
Tabla 2-3: Configuración de Componentes	46

ÍNDICE DE FIGURAS

Figura 1-1: Emisividad, Reflexión, Transmisión	6
Figura 2-1: Espectro Electromagnético	8
Figura 3-1: Energía emitida por un objeto a diferente temperatura	8
Figura 4-1: Escalas de Temperatura.....	9
Figura 5-1: Funcionamiento de una Cámara Digital	10
Figura 6-1: Forma gráfica: Pixel, FOV, IFOV	12
Figura 7-1: Etapas de un Sistema de Visión Artificial	14
Figura 8-1: Componentes de un Sistema de Visión Artificial	15
Figura 9-1: Partes de una Raspberry Pi.....	16
Figura 10-1: Pines de la Raspberry Pi 3B	18
Figura 11-1: Pi Camera Board V2	19
Figura 12-1: Cámara Radiométrica FLIR Lepton 2.5, Breakout Board v1.4.....	20
Figura 13-1: Tarjeta MicroSD.....	22
Figura 14-1: Pantalla para Raspberry.....	23
Figura 15-1: Tipos de Buzzer.....	24
Figura 16-1: Características de Python	25
Figura 17-1: Entorno de Programación Qt.....	26
Figura 1-2: Componentes del Dispositivo.....	27
Figura 2-2: Pantalla táctil HDMI de 5”.....	28
Figura 3-2: Raspberry Pi 3 Modelo B	29
Figura 4-2: Módulo Buzzer Activo	29
Figura 5-2: Instalación del S.O. Raspbian.....	30
Figura 6-2: Iniciando Raspbian.....	31
Figura 7-2: Configurando Raspbian.....	31
Figura 8-2: Diseño de la Interfaz Gráfica.....	33
Figura 9-2: Ejecutando Qt Creator	33
Figura 10-2: Diseño de la GUI en Qt Creator	35
Figura 11-2: Diagrama de flujo base para los algoritmos.....	37
Figura 12-2: Archivos fuente y cabecera del proyecto	40
Figura 13-2: Conexión Cámara Flir Lepton	41
Figura 14-2: Conexión Cámara Board V2	42
Figura 15-2: Conexión del módulo buzzer.....	42
Figura 16-2: Conexión de Pantalla HDMI de 5”	43

Figura 17-2: Módulo de carga para Raspberry	43
Figura 1-3: Lectura Interfaz PC	44
Figura 2-3: Lectura de pines de la Raspberry Pi	45
Figura 3-3: Configuración de Componentes	47
Figura 4-3: Prueba Inicial del Algoritmo	47
Figura 5-3: Prueba Inicial del Algoritmo	48
Figura 6-3: Error Visualización de las Variables	48
Figura 7-3: Error Visualización de las Variables	49
Figura 8-3: Pantalla oficial para Raspberry Pi	50
Figura 9-3: Imagen Térmica 320x240 pixeles.....	51
Figura 10-3: Imagen Espectro Infrarrojo.....	51
Figura 11-3: Vista Lateral del Dispositivo	52
Figura 12-3: Vista Posterior del Dispositivo	52
Figura 13-3: GUI del Dispositivo	53
Figura 14-3: Prueba en personas	54
Figura 15-3: Prueba en animales.....	54
Figura 16-3: Prueba en mezcladora con doble pared.....	54
Figura 17-3: Prueba en instalación eléctrica	55
Figura 18-3: Prueba en cableado.....	55
Figura 19-3: Prueba en objetivos a Corta distancia	55
Figura 20-3: Relación Irradiancia-Temperatura	56
Figura 21-3: Prueba en objetivo a larga distancia	56
Figura 22-3: Prueba en objetivo FCC actualizada.....	56
Figura 23-3: Lectura de Corriente del Dispositivo	57

ABREVIATURAS

μK:	micro Kelvin, unidad de medida de temperatura.
μm:	micro metros, unidad de medida de distancia.
AGC:	Automatic Gain Control, Control de Ganancia Automático.
C:	Centígrado, unidad de medida de temperatura.
CCD:	Charge Couple Device, Dispositivo de Carga Acoplada.
CMOS:	Complementary Metal Oxide Semiconductor.
CPU:	Central Processing Unit, Unidad Central de procesamiento.
CSI:	Camera Serial Interface, Interfaz serie de la cámara.
DC:	Direct Current.
DSI:	Database of Scientific Illustrators, Archivo de Ilustradores Científicos.
F:	Fahrenheit, unidad de medida de temperatura.
FFC:	Flat Field Correction, Corrección de Campo Plano.
FLIR:	Forward Looking InfraRed, Vista Infrarroja Frontal.
FOV:	Field of View, Campo de Visión.
FPA:	Focal Plane Array, Matriz de Plano Focal.
FPS:	Frames Per Second, Cuadros por Segundo.
GND:	Ground, Tierra.
GPIO:	General Purpose Input/Output, Entrada/Salida de Propósito General.
GUI:	Graphical User Interface, Interfaz gráfica del usuario.
HD:	High Definition.
HDMI:	High Definition Multimedia Interface, Interfaz Multimedia de Alta Definición.
HSV:	Hue Saturation Value.
Hz:	Hertz, unidad de medida de frecuencia.

I²C:	Inter-Integrated Circuit, Circuito Inter-Integrado.
IDE:	Integrated Development Environment, Entorno de Desarrollo Integrado.
IFOV:	Instantaneous Field Of View Campo de Visión Instantáneo.
IO:	In/Out, Entrada/Salida.
ISO:	International Organization for Standardization, Organización Internacional para la Estandarización.
J:	Julios, unidad de medida de energía.
JPG:	Joint Photographic Group, formato de imagen.
K:	Kelvin, unidad de medida de temperatura.
LWIR:	Long Wave Infrared, Infrarrojo de Onda Larga.
mA:	mili Amperio, unidad de medida de corriente.
MB:	Megabyte, unidad de medida de cantidad de datos informáticos.
MIPI:	Mobile Industry Processor Interface, Interfaz de Procesador de la Industria Móvil.
mm:	milímetro, unidad de medida de distancia.
ms:	milisegundo, unidad de medida de tiempo.
mW:	Unidad de medida de potencia.
PAL:	Phase Alternating Line, Línea alterna de fase.
PWM:	Pulse-Width Modulation, Modulación de Ancho de Pulso.
RAM:	Random Access Memory, Memoria de Acceso Aleatorio.
RCA:	Radio Corporation of America, Radio Corporación de America.
RGB:	Red Green Blue, tipo de imagen para formatos.
s:	segundo, unidad de medida de tiempo.
SD:	Secure Digital, Seguro Digital.
SDIO:	Secure Digital Input / Output, Seguro Digital de Entrada / Salida.

- SDK:** Software Development Kit, Kit de Desarrollo de Software.
- SPI:** Serial Peripheral Interface, Interfaz Periférica Serial.
- UART:** Universal Asynchronous Receiver-Transmitter, Receptor-Transmisor Asincrónico Universal.
- USB:** Universal Serial Bus.
- V:** voltio, unidad de medida de tensión.

RESUMEN

En este proyecto se propuso implementar un dispositivo experimental que a partir de las emisiones infrarrojas del cuerpo detectado permita medir remotamente la temperatura de un área en específico, para ello se realizó un estudio a varios proyectos elaborados anteriormente y relacionados directamente con el desarrollo de la tecnología. Para ello se investigaron los diferentes métodos para detectar la radiación infrarroja, en el mercado existen sensores con la capacidad de detectar dicha radiación, pero se optó por utilizar la cámara radiométrica Flir Leptón 2.5 ya que integra un conjunto de lente de foco fijo, un conjunto de sensores de microbolómetro de onda larga infrarroja de 80x60 (LWIR) y componentes electrónicos de procesamiento de señales. Estas señales se envían a una mini computadora llamada Raspberry Pi por medio de la interfaz I²C y SPI. Para establecer estas comunicaciones, se diseñó un algoritmo y una interfaz gráfica para enlazar las componentes de la misma con el código de programación mediante el entorno de programación Qt Creator. Adicionalmente se utilizaron varios periféricos para implementar un dispositivo y usarlo ya sea de forma fija o portátil. Durante las pruebas iniciales se pudieron corregir errores y ajustar algunos parámetros y en las pruebas finales se tomaron imágenes térmicas en áreas tanto domésticas e industriales. Después de realizar las pruebas en las diferentes áreas se pudo observar que al presentarse factores como el humo y el polvo, la función creada que actualiza la matriz de campo focal es de gran utilidad ya que permite una actualización de los términos de corrección interna utilizados para mejorar la calidad de la imagen. Otros factores como la distancia entre la cámara térmica y el objetivo a detectar, así como también sus dimensiones y la temperatura del entorno servirán para tener una mejor respuesta en la adquisición y visualización de los datos obtenidos.

PALABRAS CLAVES: <INGENIERÍA Y TECNOLOGÍA ELECTRÓNICA>, <VISIÓN ARTIFICIAL>, <CÁMARA TÉRMICA>, <TERMOGRAFÍA>, <INFRARROJO>, <RASPBERRY PI (SOFTWARE-HARDWARE)>, <INTERFAZ GRÁFICA>.

ABSTRACT

In this project, it was proposed to implement an experimental device that, based on the infrared emissions of the detected body, allows to be measured remotely the temperature of a specific area, for which a study was made of several projects previously developed and directly related to the development of technology. To do this, different methods to detect infrared radiation were investigated on the market, there are sensors with the ability to detect this radiation, but the FLIR Lepton 2.5 radiometric camera was chosen because it integrates a fixed focus lens assembly, a set of infrared long-wave microbolometer sensors of 80x60 (LWIR) and electronic components of signal processing. These signals are sent to a mini computer called Raspberry Pi through the graphic interface I²C y SPI. To establish these communications, it was designed an algorithm and a graphic interface to link the components of the same code through the environment of programming Qt Creator. Additionally, several peripherals were used to implement a device and use it either fixed or portable. During the initial tests it was possible to correct errors and adjust some parameters and in the final tests, thermal images were taken in both domestic and industrial areas. After carrying out the tests in the different areas it was observed that when factors such as smoke and dust appear, the created function that updates the focal field matrix is very useful since it allows an update of the internal correction terms used to improve the quality of the image. Other factors such as the distance between the thermal chamber and the objective to be detected, as well as its dimensions and the temperature of the environment, will serve to have a better response in the acquisition and visualization of the obtained data.

KEYWORDS: <ELECTRONIC ENGINEERING AND TECHNOLOGY>, <ARTIFICIAL VISION>, <THERMAL CHAMBER>, <THERMOGRAPHY>, <INFRARED>, <RASPBERRY PI (SOFTWARE-HARDWARE)>, <GRAPHICAL INTERFACE>.

INTRODUCCIÓN

En la década de los 60 con la idea básica de conectar una cámara de video a un computador, mediante el proceso digital de imágenes y su análisis se origina la visión artificial. El resultado de este trabajo se define el inicio de esta ciencia derivada de la Inteligencia Artificial y su progreso, durante las últimas décadas se ha desarrollado una gran velocidad. Los primeros sistemas se basaron en imágenes binarias que se procesaban en bloques, ventanas o píxeles y mediante el desarrollo de nuevos algoritmos se pudo reconocer el contorno de elementos y su lugar dentro de una imagen, pero estos algoritmos poseían límites al no poder operar en diversos tipos de iluminación.

La radiación infrarroja fue descubierta durante el siglo XIX por Herschel, los primeros usos técnicos se realizaron alrededor del año 1920 donde se pudo realizar la medición de la radiación y después de un tiempo se transformó la lectura de los niveles de radiación infrarroja en temperatura.

Para visualizar la imagen térmica de un objeto en un área sin iluminación se creó la primera cámara termográfica en 1958 en Suecia para aplicaciones militares y así detectar amenazas durante la noche, la primera cámara aplicada a actividades comerciales se implementó en 1965 para poder inspeccionar líneas de alta tensión. Los inconvenientes de las primeras cámaras era su gran tamaño y no fue hasta 1997 que apareció en el mercado una cámara portátil. Esto fue posible al crear un detector no refrigerado llamado microbolómetro y mediante el mismo se pudieron crear cámaras más económicas y menos susceptibles de averías. Fue a partir de aquí cuando se extendió el uso de las cámaras termográficas a campos más comerciales en sus distintas áreas.

Para el presente trabajo se realizó un estudio a varios proyectos elaborados anteriormente y relacionados directamente con el desarrollo de la tecnología a futuro, también de nuevos recursos en el mercado que permiten el desarrollo de aplicaciones orientadas al presente tema, obteniendo experiencias que tienen relación con el problema objeto de investigación.

FORMULACIÓN DEL PROBLEMA

¿Se podrá implementar un dispositivo que permita visualizar la imagen térmica de un objetivo mediante su fuente de radiación infrarroja?

SISTEMATIZACIÓN DEL PROBLEMA

¿Qué métodos tecnológicos existen para la captura y visualización de imágenes térmicas?

¿Cuáles serán las limitaciones de operación de la cámara Flir Lepton y del dispositivo de visualización de imágenes térmicas al momento de detectar el objetivo?

¿Cuál será la plataforma hardware y software que permita implementar el dispositivo de visualización de imágenes térmicas usando la cámara Flir Lepton?

¿Cuáles serán los escenarios de operación y funcionalidad del dispositivo de visualización de imágenes térmicas implementado?

JUSTIFICACIÓN

JUSTIFICACIÓN TEÓRICA

Por medio de este proyecto se pretende implementar un dispositivo experimental que a partir de las emisiones infrarrojas del cuerpo detectado permita medir remotamente la temperatura de un área en específico.

El diseño del dispositivo permitirá tener una visión de un objetivo en función de la temperatura, con ella se podrá obtener una termografía del estado del mismo. Esta termografía ofrecerá no solo una imagen visual, sino que contendrá la información sobre los diferentes niveles de temperatura.

Para el hardware del dispositivo se pretende utilizar una matriz de sensores infrarrojos el cual se va a encargar de medir la temperatura del objetivo y enviar las señales para usarlas como datos que serán adquiridos por una computadora con periféricos de entrada y salida para el ingreso y visualización de información y un actuador que realizará una función según su configuración.

Para el software, por medio de un entorno de programación se diseñará el algoritmo para procesar la matriz de sensores infrarrojos y un interfaz con varias funciones entre ellas visualizar los datos procesados por el algoritmo.

Para la comunicación entre el software y hardware se realizará un estudio para elegir el más adecuado para la adquisición y transmisión de los datos dependiendo de las capacidades cada uno de los elementos que conforman el dispositivo.

El uso de una cámara digital se utilizará únicamente para visualizar el área del objetivo a detectar en función de su radiación lumínica para compararla con la imagen térmica de la misma área y realizar un análisis de sus resultados.

JUSTIFICACIÓN APLICATIVA

La radiación infrarroja lleva información acerca de la distribución de la temperatura de los objetos estudiados dependiendo de la potencia de los sensores y la adquisición de sus datos frente a diversos factores como el humo, polvo, ruido entre otros no permiten tomar imágenes digitales en función de su radiación lumínica en áreas que el usuario necesita lo que hace este dispositivo una herramienta útil.

El uso de este dispositivo, ya sea como herramienta única o combinada con otros equipos, puede agilizar el trabajo de algún proceso de forma notable y al diseñar el algoritmo con un código abierto esto permitirá al usuario modificarlo para crear sus propias funciones.

El infrarrojo es básicamente radiación caliente y la implementación de este dispositivo permitirá visualizar termografías localizadas en un área específica casi con exactitud dónde se pueden detectar en un objetivo múltiples pérdidas de energía, sin necesidad de efectuar ninguna prueba destructiva.

Este dispositivo abre un amplio campo de aplicaciones, entre ellas sistemas mecánicos donde por medio de la evaluación de equipos de aire acondicionado y calefacción, detección de fugas en el aislamiento en equipos de refrigeración y la inspección de instalaciones eléctricas para localizar defectos en las instalaciones lográndose detectar las pérdidas energéticas.

OBJETIVOS

Objetivo General

Diseñar e implementar un dispositivo que permita visualizar una imagen térmica de un objetivo específico mediante su fuente de radiación infrarroja.

Objetivos Específicos

- Estudiar los métodos que existen para escanear una fuente de radiación infrarroja y elegir el más adecuado.
- Analizar los límites del dispositivo de visualización de imagen térmica usando la cámara Flir Lepton frente a diversos factores como distancia, superficie, cantidad de luz entre otros.
- Implementar una interfaz hardware y software que permita visualizar una imagen térmica capturada usando la cámara Flir Lepton.
- Evaluar en diferentes escenarios la funcionalidad del dispositivo de captura y visualización de imágenes térmicas implementado.

CAPÍTULO I

1. MARCO TEÓRICO CONCEPTUAL

1.1 Termografía

1.1.1 Definición

La termografía es un método de medición pasivo, sin la necesidad de estar en contacto con el objeto mediante la detección de radiación infrarroja del espectro electromagnético. Con el uso de cámaras termográficas se puede registrar dicha radiación y convertirla en información.

El principio de la termografía define que “*Cualquier objeto cuya temperatura sea superior al cero absoluto (0 Kelvin = -273.15 °C) emite radiación infrarroja, invisible al ojo humano*”. (Melgosa, 2011a: p.20)

1.1.2 Emisividad, Reflexión y Transmisión

Mediante una cámara termográfica se puede registrar la radiación de onda larga, emitida, reflejada y transmitida a factores que siempre están presentes en un objeto al momento de estar en el campo de visión de la cámara. A continuación, se explican las diferentes formas en las que nos podemos encontrar estas radiaciones de onda según la Guía de Termografía Testo.

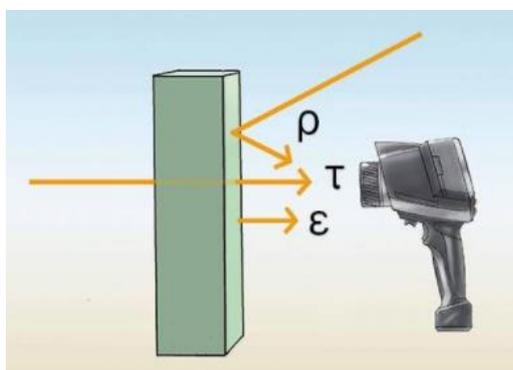


Figura 1-1: Emisividad, Reflexión, Transmisión
Fuente: Testo, 2008

- **Emisividad (ϵ):** Es la medida de la capacidad que posee un material para emitir radiación infrarroja. La emisividad puede variar dependiendo las propiedades del material, de su superficie y la temperatura del objeto que se va a medir. En teoría su

valor máximo es igual a uno, pero ya que en la realidad existen cuerpos que también reflejan y transmiten radiación el valor máximo es menor que uno es decir $\mathcal{E} < 1$. (Testo, 2008 b: p.6)

- **Reflexión (ρ):** Es la medida de la capacidad que posee un objeto para reflejar la radiación infrarroja. La medida de reflexión de un material va a depender de factores como su tipo, temperatura y la superficie del mismo. Los materiales que tienen superficies lisas y pulidas reflejan con mayor intensidad la radiación infrarroja que un material que no esté pulido apropiadamente y que posea superficies irregulares. (Testo, 2008c: p.7)
- **Transmisión (τ):** Es la medida de la capacidad que posee un material de transmitir de la radiación infrarroja. La mayoría de los materiales no poseen una propiedad transmisiva lo que causa una impermeabilidad a la radiación infrarroja, esto va a depender del grosor de material y su tipo. (Testo, 2008d: p.8)

1.2 El Espectro Electromagnético

El espectro electromagnético es el rango de todos los tipos de radiación electromagnética clasificados por longitud de onda. (Melgosa, 2011, p.21)

1.3 Onda

Una onda es la propagación de una perturbación que transfiere energía progresivamente de un punto a otro a través de un medio y que puede tener la forma de deformación elástica, una variación de presión, intensidad magnética, eléctrica o de temperatura. Las ondas distribuyen, en función de su energía a lo largo de espectro electromagnetismo desde la radiación de menor longitud de onda, como los rayos gama y los rayos X, pasando por la luz ultravioleta, la luz visible y los rayos infrarrojos, hasta las ondas electromagnéticas de mayor longitud de onda, como son las ondas de radio. (Melgosa, 2011, p.22)

Las longitudes de onda no tienen los límites tan marcados como los que se representan en la figura 2-1. Sin embargo, en la práctica los límites y fronteras son muy evidentes. La cantidad de luz perceptible tiene una similitud a la infrarroja ya que ésta compuesta de una mezcla de longitudes de onda. A través del ojo humano, se pueden percibir colores correspondientes a

un rango de longitudes de onda muy estrecho, entre $0,4 \mu\text{m}$ a $0,7 \mu\text{m}$. En el espectro, existe una franja en donde la radiación tiene la capacidad de poder transmitir calor sea por emisión o absorción.

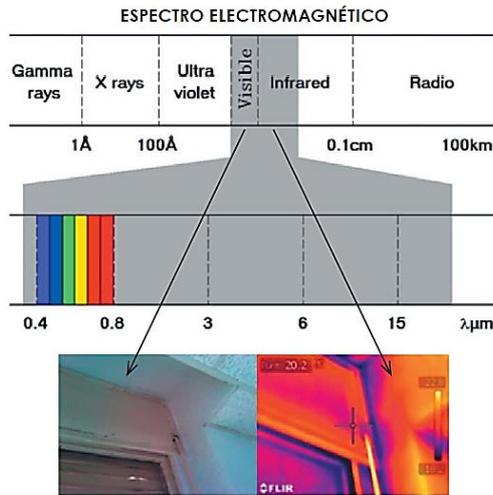


Figura 2-1: Espectro Electromagnético
Fuente: Melgosa, S. 2011

En la gráfica de la figura 3-1, se observa la energía que emite un objeto a diferentes niveles de temperatura. Se puede ver que, mientras más grande es el nivel de temperatura, mayor es el pico de energía.

A medida que la temperatura aumenta el pico de energía se torna en una onda corta gradualmente y a cuando la temperatura decremanta el pico de energía se convierte en una onda larga.

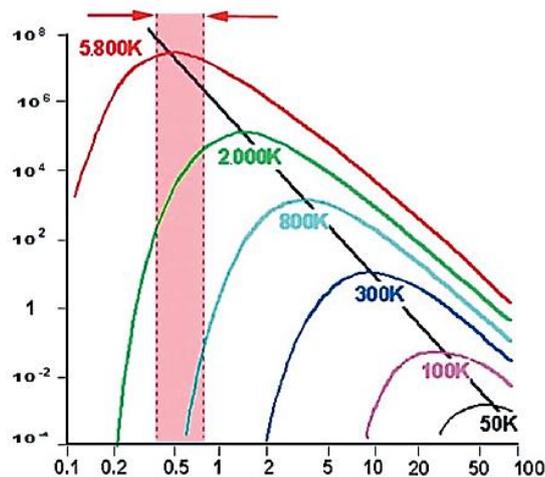


Figura 3-1: Energía emitida por un objeto a diferente temperatura
Fuente: Melgosa, S. 2011

1.4 Calor y Temperatura

A pesar de que tienen relación el calor y la temperatura son temas que tienen conceptos distintos.

- **Calor:** Es la transferencia de energía entre diferentes cuerpos o diferentes zonas de un mismo cuerpo que se encuentran a distintas temperaturas. Este flujo siempre ocurre desde el cuerpo de mayor temperatura hacia el cuerpo de menor temperatura, ocurriendo la transferencia de calor hasta que ambos cuerpos se encuentren en equilibrio térmico. (Melgosa, 2011, p.43)

El calor es generado por medio de la transformación de la energía ya sea por un proceso de combustión, movimiento o rozamiento y su unidad de medida son los Julios (J).

- **Temperatura:** Es una magnitud a un sistema termodinámico y referida a la energía interna asociada a las partículas del sistema y su movimiento. Se mide en Kelvin (K) o en grados Celsius ($^{\circ}\text{C}$). (Melgosa, 2011, p.44)

Existen varias escalas de temperatura como se muestra en la figura 4-1 la cuales se dividen en dos grupos, las absolutas (Kelvin) y las relativas (Celsius, Fahrenheit).

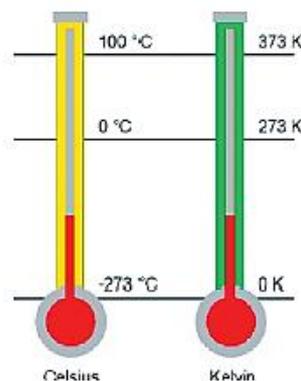


Figura 4-1: Escalas de Temperatura

Fuente: Melgosa, S. 2011

En resumen, se puede decir que la temperatura es una medida relativa que permite la comparación de un objeto con otro mientras que el calor es una magnitud cuantificable y medible.

1.5 Sensor:

Es un dispositivo con la capacidad de detectar magnitudes físicas denominadas variables de instrumentación y las convierte en variables eléctricas emitiendo una señal analógica o digital, según la aplicación se utiliza un sensor en específico. (Ortega, 2013)

Mediante un conjunto de sensores que transforman una señal luminosa en una señal eléctrica se conectan en forma de matriz de tal manera que cada nivel de luz que detecten se pueda convertir en información la cual servirá para un proceso posterior.

1.6 Cámara Digital

1.6.1 Funcionamiento

La óptica de la cámara permite detectar la radiación lumínica que es emitida por el objetivo o área en específico, cada punto o pixel de la cámara envía un cambio de tensión o resistencia eléctrica que será leída por los elementos electrónicos de la cámara. Esa señal es convertida en una imagen electrónica en la pantalla, donde los distintos tonos de color se corresponden con distintas señales o lumínica procedentes del objeto de estudio.



Figura 5-1: Funcionamiento de una Cámara Digital
Fuente: Melgosa, S. 2011

1.7 Cámara Termográfica

Una cámara termográfica es un dispositivo que detecta el patrón térmico del objetivo al que se apunta, en el espectro de la longitud de onda infrarroja y sin entrar en contacto con ese cuerpo. La radiación registrada por la cámara termográfica consiste en la radiación de onda larga, emitida, reflejada y transmitida que surge de los objetos presentes en el campo de visión de la cámara. (Melgosa, 2011b: p.20)

1.8 Imagen Radiométrica

Una imagen radiométrica es una imagen térmica que contiene información de las medidas de temperatura en todos los puntos de la imagen. La información térmica corresponde a un patrón, un estado puntual en cuanto a su temperatura.

Se dice que es puntual ya que no se considera el objeto como algo aislado, al contrario, estará bajo condiciones cambiantes rodeado de otros objetos que le influyan unas actuaciones, etc. (Melgosa, 2011c: p.20)

1.8.1 Importancia de la Sensibilidad (Resolución)

La resolución visual afecta a la claridad con la cual se pueden ver los componentes o equipos bajo estudio. En ocasiones, cuando se diagnostica un problema, se debe distinguir la fuente real del problema, entre dos o más posibles fuentes, localizadas una cerca de otra.

Cuanto más claro se pueda ver cada componente individual, más claramente se podrá distinguir la fuente del problema. Esto en termografía dependerá de varios factores, del diseño del detector, del número de píxeles que es capaz de generar la imagen y de la calidad de la óptica.

1.9 Pixel, FOV y IFOV

A continuación, se van a definir algunos conceptos según la Guía de Termografía Infrarroja:

- **Pixel:** *Es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico. En termografía, es la menor unidad que pueda detectar energía infrarroja.* (Melgosa, 2011a: p.31)
- **FOV (Campo de Visión):** *Ángulo horizontal visible a través de una lente de infrarrojos.* (Melgosa, 2011b: p.31)
- **IFOV (Campo de Visión Instantáneo):** *Medida de resolución geométrica de una cámara de infrarrojos.* (Melgosa, 2011c: p.31)

De una forma gráfica sería lo siguiente:

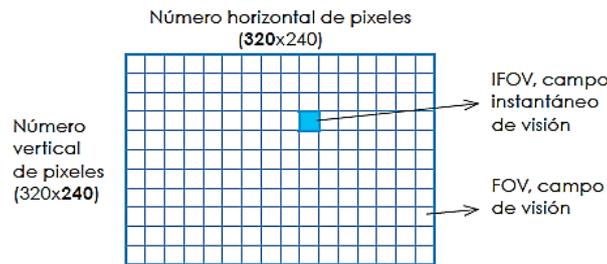


Figura 6-1: Forma gráfica: Pixel, FOV, IFOV
Fuente: Melgosa, S. 2011

1.10 Reglas para capturar una Imagen Térmica

Existen cuatro reglas para capturar una imagen térmica según la Guía de Termografía Infrarroja:

- **Encadre de la imagen:** Para medir un objeto se debe colocar la cámara en una posición en el que el objeto se encuentre de la forma más visible y en posición que otorgue mayor información a la cámara térmica. (Melgosa, 2011a: p.39)
- **Enfoque:** El tomar una imagen de forma desenfocada puede mostrar medidas erróneas para ello se deben tener los equipos de la cámara en un buen estado manteniendo la pantalla limpia y la lente calibrada. (Melgosa, 2011b: p.39)
- **Ajuste del rango de temperatura:** No todos los equipos trabajan a la misma temperatura así que, si existe la posibilidad de fijar los rangos mínimos y máximos a detectar de modo que la temperatura del objetivo se encuentre dentro de este rango. Si se establece un rango muy amplio dará como resultado una precisión menor en los niveles de medida. (Melgosa, 2011c: p.39)
- **Cuidado con las condiciones meteorológicas:** La temperatura en un ambiente exterior, así como la hora en la que se realiza una inspección termográfica podría dar como resultado un cambio en los datos obtenido de un día a otro, para evitar estos problemas se debe elegir un momento adecuado para ya las condiciones climáticas es un factor que se debe tomar en cuenta. (Melgosa, 2011d: p.39)

1.11 Visión Artificial

1.11.1 Definición

La visión artificial es una rama de la Inteligencia Artificial, consiste en captar imágenes en línea mediante receptores con transferencia de carga CCD y su posterior tratamiento a través de técnicas de procesamiento avanzadas. (Santiago, 2007)

1.11.2 Aplicaciones

Existen diferentes aplicaciones en las que se puede aplicar visión artificial como se detalla en la tabla 1-1.

Tabla 1-1: Aplicaciones de la Visión Artificial

Inspección industrial y control de calidad	<ul style="list-style-type: none">• Verificación de etiquetado y códigos.• Inspección de soldaduras, circuitos impresos, motores.• Clasificación de piezas
Vigilancia y seguridad	<ul style="list-style-type: none">• Control de accesos.• Control de abandonos.
Identificación	<ul style="list-style-type: none">• Identificación biométrica: huellas, pisadas, firmas, iris.• Reconocimiento de caras, de gestos.
Control	<ul style="list-style-type: none">• Reconocimiento de matrículas, peaje por volumen, control de flujo.• Sistemas de ayuda a la conducción.
Operación de Robots	<ul style="list-style-type: none">• Industriales• Vehículos autónomos
Análisis de imágenes	<ul style="list-style-type: none">• Satélite
Aplicaciones militares	<ul style="list-style-type: none">• Detección de objetivos• Guiado balístico
Bioingeniería	<ul style="list-style-type: none">• Ayuda al diagnóstico

Fuente: http://arantxa.ii.uam.es/~jms/seminarios_doctorado/abstracts2006-2007/20070503LSalgado.pdf

1.11.3 Niveles de la visión artificial

Según (Valencia & Brito, 2015, p.22-23) los niveles de la visión artificial son:

- **Nivel Bajo:** se trabaja directamente con los puntos (píxeles) para extraer propiedades como el gradiente, profundidad, textura, color, etc.
- **Nivel Intermedio:** agrupa los elementos obtenidos en el nivel bajo para obtener bordes, líneas, regiones, generalmente con el propósito de segmentar la imagen.
- **Nivel Alto:** orientado al proceso de interpretación de los entes obtenidos en los niveles inferiores y se utilizan modelos o conocimiento a prioridad del problema.

1.11.4 Etapas de un sistema de visión artificial

En la figura 7-1 se detallan las etapas de un sistema de visión artificial donde en muchas ocasiones es necesario una retroalimentación entre las fases:

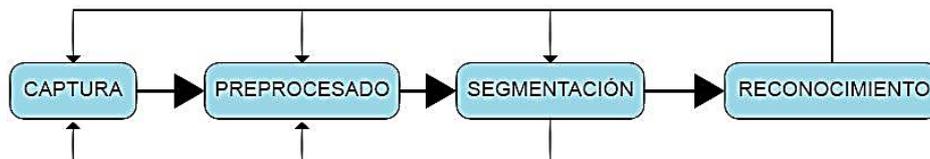


Figura 7-1: Etapas de un Sistema de Visión Artificial
Fuente: Maldonado, L. 2018

- **Captura:** En la etapa de captura se obtiene una adquisición digital de imágenes mediante la utilización de un sensor. (Gómez and Domínguez, 2016a: p.5)
- **Preprocesado:** En la etapa de preprocesado elimina las partes que no son útiles y/o realza lo más significativo de cada imagen. (Gómez and Domínguez, 2016b: p.5)
- **Segmentación:** Esta etapa consiste en dividir una imagen digital en regiones iguales con respecto a una o más características. (Gómez and Domínguez, 2016c: p.5)

- **Reconocimiento:** Nos permite distinguir los diferentes objetos segmentados en función de sus características. (Gómez and Domínguez, 2016d: p.5)

1.11.5 Componentes de un sistema de visión artificial

La figura 8-1 muestra los componentes de un sistema de visión artificial los mismos que se detallan a continuación:

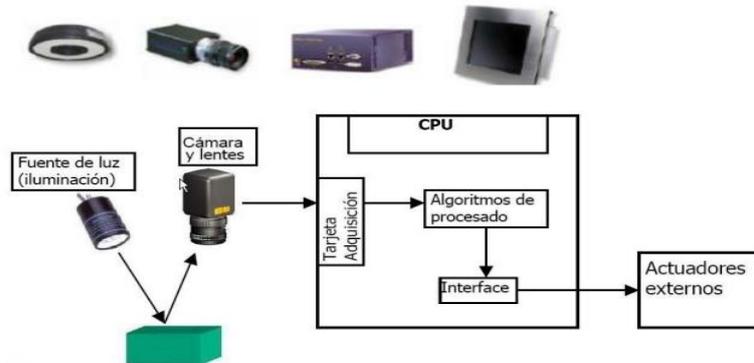


Figura 8-1: Componentes de un Sistema de Visión Artificial

Fuente: <https://bit.ly/2AEZ0ee>

- **Iluminación:** Según su aplicación se utiliza en la Visión Artificial para controlar la forma en que la cámara ve el objeto.
- **Cámara:** Su función es capturar la imagen proyectada a través del sensor para poder transferirla a un sistema electrónico.
- **CPU:** Su función no solo es recoger las imágenes guardadas en la tarjeta de adquisición y mostrarlas en una pantalla mediante una interfaz, también de procesarlas para llevar a cabo su cometido según el algoritmo de procesamiento.
- **Actuadores Externos:** Éstos se sincronizan con el proceso, para adquirir la imagen en el momento adecuado.

1.12 Raspberry Pi

1.12.1 Definición

Raspberry Pi es un mini ordenador de pequeño tamaño el cual consume pocos recursos para su funcionamiento, por lo general éste trabaja con sistemas operativos basados en Linux y también incorpora funciones de electrónica como pines GPIO (General Purpose Input/Output), de comunicación como UART (Universal Asynchronous Receiver-Transmitter), SPI (Serial Peripheral Interface), I²C (Inter-Integrated Circuit). Todas estas funciones lo hacen ideal para poder usarlo en proyectos de electrónica interactuando con sensores y actuadores que se van a utilizar. (Llamas, 2017)

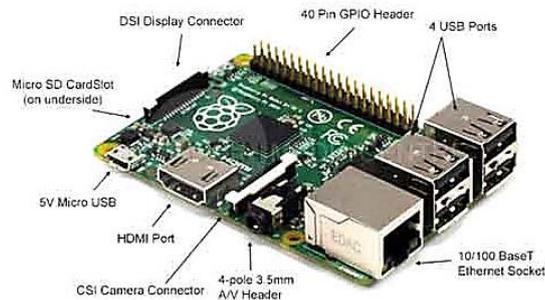


Figura 9-1: Partes de una Raspberry Pi
Fuente: <https://bit.ly/2yKoa9Z>

Existen varios modelos de Raspberry Pi, para el dispositivo se va a utilizar una Raspberry Pi 3 modelo B, su procesador, memoria RAM y otras características hacen a este mini ordenador el ideal para la implementación del proyecto.

1.12.2 Características

Las características de la Raspberry Pi 3B se detallan a continuación:

Tabla 2-1: Características de la Raspberry Pi 3B

Procesador	<ul style="list-style-type: none">• Chipset Broadcom BCM2387.• 1,2 GHz de cuatro núcleos ARM Cortex-A53
------------	--

GPU	<ul style="list-style-type: none"> • Dual Core VideoCore IV ® Multimedia Co-procesador. Proporciona Open GL ES 2.0, OpenVG acelerado por hardware, y 1080p30 H.264 de alto perfil de decodificación. • Capaz de 1 Gpixel / s, 1.5Gtexel / s o 24 GFLOPs con el filtrado de texturas y la infraestructura DMA
RAM	<ul style="list-style-type: none"> • 1GB LPDDR2
Conectividad	<ul style="list-style-type: none"> • Ethernet socket Ethernet 10/100 BaseT • 802.11 b / g / n LAN inalámbrica y Bluetooth 4.1 (Classic Bluetooth y LE) • Salida de vídeo <ul style="list-style-type: none"> ○ HDMI rev 1.3 y 1.4 ○ RCA compuesto (PAL y NTSC) • Salida de audio <ul style="list-style-type: none"> ○ jack de 3,5 mm de salida de audio, HDMI ○ USB 4 x Conector USB 2.0 • Conector GPIO <ul style="list-style-type: none"> ○ 40-clavijas de 2,54 mm (100 milésimas de pulgada) de expansión: 2x20 tira ○ Proporcionar 27 pines GPIO, así como 3,3 V, +5 V y GND líneas de suministro • Conector de la cámara de 15 pines cámara MIPI interfaz en serie (CSI-2) • Pantalla de visualización Conector de la interfaz de serie (DSI) Conector de 15 vías plana flex cable con dos carriles de datos y un carril de reloj • Ranura de tarjeta de memoria Empuje / tire Micro SDIO

Fuente: <https://www.pccomponentes.com/raspberry-pi-3-modelo-b>

1.12.3 Distribución de Pines

Los pines de la Raspberry Pi 3B se dividen en pines de alimentación, pines de propósito general, y pines de interfaz digital. La distribución se muestra a continuación en la figura 10-1:

GPIO#	2nd fun	pin#	pin#	2nd fun	GPIO#
-	+3V3	1	2	+5V	-
GPIO2	SDA1 (I2C)	3	4	+5V	-
GPIO3	SCL1 (I2C)	5	6	GND	-
GPIO4	GCLK	7	8	TXD0 (UART)	GPIO14
-	GND	9	10	RXD0 (UART)	GPIO15
GPIO17	GEN0	11	12	GEN1	GPIO18
GPIO27	GEN2	13	14	GND	-
GPIO22	GEN3	15	16	GEN4	GPIO23
-	+3V3	17	18	GEN5	GPIO24
GPIO10	MOSI (SPI)	19	20	GND	-
GPIO9	MISO (SPI)	21	22	GEN6	GPIO25
GPIO11	SCLK (SPI)	23	24	CE0_N (SPI)	GPIO8
-	GND	25	26	CE1_N (SPI)	GPIO7
EEPROM	ID_SD	27	28	ID_SC	EEPROM
GPIO5	-	29	30	GND	-
GPIO6	-	31	32	-	GPIO12
GPIO13	-	33	34	GND	-
GPIO19	-	35	36	-	GPIO16
GPIO26	-	37	38	-	GPIO20
-	GND	39	40	-	GPIO21

Figura 10-1: Pines de la Raspberry Pi 3B
Fuente: <https://bit.ly/2ADNOyw>

- Los pines de alimentación proporcionan los niveles de voltaje de 5V, 3.3V y GND estos pines a diferencia del resto pueden proporcionar corrientes de hasta 200mA. (Valencia & Brito, 2015a: p.47)
- Los pines de propósito general son pines cuyo hardware está dedicado para control y señalización ON-OFF y salidas PWM por software. (Valencia & Brito, 2015b: p.47)
- Los pines de interfaz digital se dividen en pines de comunicación I²C, SPI, Serial UART, los mismos tienen hardware dedicado para las tareas de comunicación. (Valencia & Brito, 2015c: p.47)

1.12.4 Cámaras Compatible con Raspberry Pi

Existen varias cámaras que son compatibles con la Raspberry Pi, para el dispositivo se van a utilizar dos tipos de cámaras la Camera Board V2 y la cámara Radiométrica FLIR Lepton 2.5 ya sus características que se detallan a continuación las hacen ideales para el dispositivo.

- **Cámara Raspberry Pi Camera Board V2:** Esta cámara ideal para usar en la Raspberry mediante el bus CSI diseñado particularmente para la interfaz de la cámara y lleva únicamente datos de píxeles al procesador. Sus características se definen en la tabla 3-1.

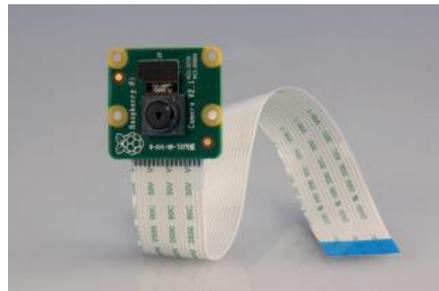


Figura 11-1: Pi Camera Board V2

Fuente: <https://bit.ly/2Of08Mc>

Tabla 3-1: Características Cámara Raspberry Pi Camera Board V2

Sensor de imagen	Sensibilidad térmica <50 mK
Resolución	8 megapíxeles
Resolución de imagen fija	3280 x 2464
Velocidad máxima de transferencia de imágenes	1080p: 30 fps (codificar y decodificar) 720p: 60 fps
Conexión a Raspberry Pi	Cable plano de 15 pines, a la interfaz serie de la cámara MIPI de 15 pines dedicada (CSI-2).
Funciones de control de imagen	Control de exposición automático Balance de blancos automático

	Filtro de banda automático Detección de luminancia automática de 50/60 Hz Calibración automática del nivel de negro
Rango de temperatura	Funcionamiento: -20° a 60° Imagen estable: -20° a 60°
Tamaño de la lente	1/4"
Dimensiones	23.86 x 25 x 9mm
Peso	3g

Fuente: <http://docs-europe.electrocomponents.com/webdocs/14db/0900766b814db308.pdf>

- Cámara Radiométrica FLIR Lepton:** Es una cámara de infrarrojos de onda larga (LWIR) completo, diseñado para interactuar fácilmente con interfaces nativas de dispositivos móviles y otros dispositivos electrónicos de consumo. La cámara se puede conectar a un módulo de arranque el cual facilita la comunicación con la Raspberry Pi.



Figura 12-1: Cámara Radiométrica FLIR Lepton 2.5, Breakout Board v1.4

Fuente: <https://bit.ly/2qldR7P>

Tabla 4-1: Características Cámara Radiométrica FLIR Lepton 2.5

Visión de conjunto	
Tecnología de sensores	Microbolómetro VOx no refrigerado
Rango espectral	Infrarrojo de onda larga, 8 μm a 14 μm

Formato de matriz	80 × 60, escaneo progresivo
Tamaño de píxel	17 μm
Velocidad de fotogramas efectiva	8,6 Hz (aplicación comercial exportable)
Sensibilidad térmica	<50 mK (0.050 ° C)
Compensación de temperatura	Automático. Imagen de salida independiente de la temperatura de la cámara.
Rango dinámico de escena	0° a 120°C
Precisión radiométrica	N / A
FOV - horizontal	51 °
FOV - diagonal	63.5 °
Formato de salida	Seleccionable por el usuario de 14 bits, 8 bits (AGC aplicado) o 24 bits RGB (AGC y color aplicada)
Protección solar	Integral
Eléctricas	
Reloj de entrada	Niveles de tensión IOS CMOS de 25 MHz nominales
Interfaz de datos de video	Video sobre SPI
Puerto de control	CCI (I2C-like), CMOS IO Voltage Levels
Voltaje de suministro de entrada (nominal)	2.8 V, 1.2 V, 2.5 V a 3.1 V IO
Disipación de potencia (Típico, Ta)	(Típico, Ta) 150 mW (funcionamiento), 650 mW (durante el evento del obturador), 4 mW (modo de espera)

Mecánicas	
Dimensiones del paquete: versión de socket (w x l x h)	10.5 x 11.7 x 6.4 mm
Peso	0.9 gramos
Ambiental	
Rango de temperatura para un óptimo funcionamiento	-10 ° C a +65 ° C
Rango de temperatura no operativo	-40 ° C a +80 ° C
Shock	1500 G @ 0,4 ms

Fuente: <https://groupgets.com/manufacturers/flir/products/radiometric-lepton-2-5>

1.12.5 Dispositivos de Almacenamiento compatible con Raspberry Pi

Debido a los puertos que posee la Raspberry Pi es compatible con diversos dispositivos de almacenamiento, para el proyecto se escogió la tarjeta MicroSD ya que cuenta con una ranura dedicada a la misma y es apropiada para las necesidades del dispositivo.



Figura 13-1: Tarjeta MicroSD

Fuente: <https://bit.ly/2Jumzcz>

A continuación, se detalla en la tabla 5-1 los tipos de tarjetas SD según su clase.

Tabla 5-1: Tipos de Tarjetas SD según su clase

CLASE	VELOCIDAD MIN.	APLICACIONES
Clase 2	2 MB/s	Fotos y grabar vídeos
Clase 4	4 MB/s	Grabar vídeo de alta definición HD
Clase 6	6 MB/s	Grabar vídeo de alta definición HD
Clase 10	10 MB/s	Grabar video Full HD y tomar fotos HD posee un bus de alta velocidad
UHS Clase 1 (U1)	10 MB/s	Graba en tiempo real y vídeos de alta definición posee un bus de Ultra Alta Velocidad
UHS Clase 3 (U3)	30 MB/s	Archivos de vídeo de resolución 4K posee un bus de Ultra Alta Velocidad

Fuente: <http://tarjetasd.com/guia/>

1.12.6 Pantalla y teclado compatible con Raspberry Pi

La Raspberry Pi cuenta con varios puertos de entrada como USB, HDMI y GPIO de modo que cualquier pantalla que cuente con uno de estos tres tipos de puertos de salida se pueden utilizar, igualmente para el teclado hoy en día la mayoría de estos poseen una alimentación USB. Para el dispositivo se va utilizar una pantalla HDMI de 5 pulgadas ya que su tamaño es apropiado para conectarlo a la Raspberry y poder utilizarlo de forma portátil.

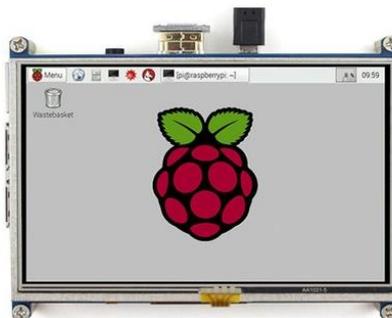


Figura 14-1: Pantalla para Raspberry

Fuente: <https://bit.ly/2P0b4zJ>

1.13 Buzzer

1.13.1 Definición

Los buzzers son transductores electroacústicos, es decir, dispositivos que convierten señales eléctricas en sonido. Existen dos tipos de buzzers los pasivos los cuales es necesario proporcionar una señal eléctrica para conseguir el sonido deseado y los activos disponen de un oscilador interno, por lo que únicamente se debe alimentar el dispositivo para que emita el sonido.



Figura 15-1: Tipos de Buzzer
Fuente: <https://bit.ly/2zjdY7V>

1.14 Entornos de Programación

1.14.1 Python

1.14.1.1 Definición

Python es un software abierto que está dedicado a la programación orientada a objetos, es una herramienta simple y versátil ya que nos provee de rapidez al momento de desarrollar aplicaciones. Es un lenguaje scripting desarrollado para hacer cualquier tipo de programa, este viene a ser un lenguaje interpretado esto quiere decir que no necesariamente debe ser compilado para que este sea ejecutado. (Valencia & Brito, 2015, p.37)

1.14.1.2 Características

El lenguaje de programación Python presenta algunas características como se muestra en la figura 16-1 las cuales se detallan a continuación:

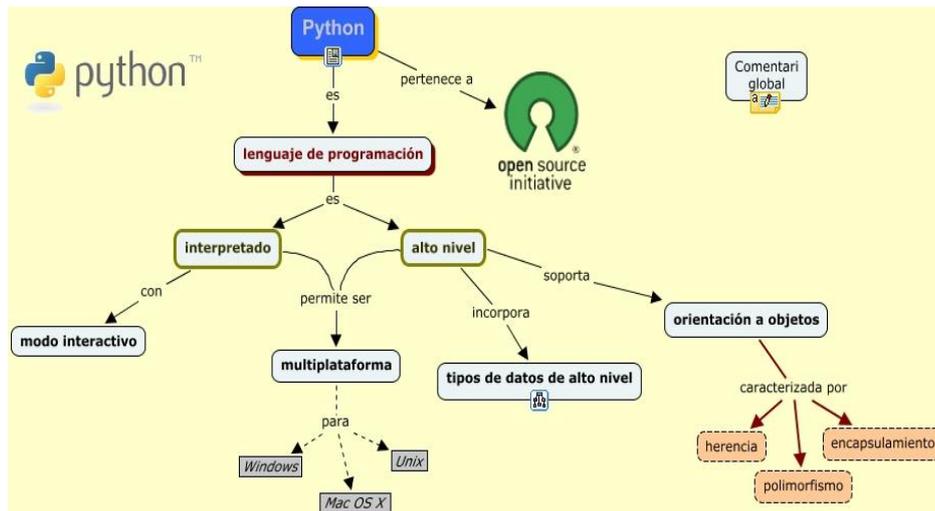


Figura 16-1: Características de Python

Fuente: <https://bit.ly/2DcgMrX>

- **Propósito general:** Es un lenguaje de programación que permite crear cualquier tipo de programas este no está orientado solo a la web. (Valencia & Brito, 2015a: p.38)
- **Multiplataforma:** Está desarrollado para sistemas informáticos diferentes siempre y cuando tenga un intérprete programado. (Valencia & Brito, 2015b: p.38)
- **Interpretado:** Python brinda la posibilidad de no compilar el código antes de ejecutarlo. (Valencia & Brito, 2015c: p.38)
- **Interactivo:** Cada sentencia se ejecuta y produce un resultado que puede ser visto por el programador ayudando a probar los resultados del código por porciones. (Valencia & Brito, 2015d: p.38)
- **Orientado a Objetos:** Python facilita crear programas con componentes reutilizables. (Valencia & Brito, 2015e: p.38)
- **Funciones y librerías:** Posee una variedad de librerías que se puede utilizar en los programas para tratar cosas puntuales y funciones específicamente para el manejo de strings, números, archivos etc. (Valencia & Brito, 2015f: p.38)

1.14.2 Qt

1.14.2.1 Definición

Qt es un conjunto de librerías o bibliotecas que se caracterizan por ser independientes de la plataforma donde se ejecutan y que permiten generar interfaces gráficas (GUIs) de forma rápida y sencilla para muchos sistemas operativos incluyendo sistemas embebidos como Raspberry Pi o Beagle Board. Dispone de un entorno de desarrollo integrado (IDE) de software libre y multiplataforma llamado Qt Creator.



Figura 17-1: Entorno de Programación Qt
Fuente: <https://bit.ly/2yHSEcN>

1.14.2.1 Características

Entre sus características destacan:

- Posee un avanzado editor de código C++.
- Aporta un GUI integrado y un diseñador de formularios.
- Herramientas para la administración de proyectos.
- Ayuda sensible al contexto.
- Contiene un depurador visual
- Resaltado y autocompletado de código

CAPÍTULO II

2. MARCO METODOLÓGICO

2.1 Componentes del Dispositivo

La selección de los componentes del dispositivo se realizó en base al método tecnológico para validar los resultados de un proceso de creación de un dispositivo tecnológico. Este proceso está compuesto de etapas secuenciales: Necesidad, Análisis, Diseño, Prototipo, Evaluación, Memoria Técnica y Construcción, las mismas facilitan la obtención de resultados y adaptarlo a los requerimientos iniciales del proyecto.

Para el diseño e implementación del dispositivo se establecen especificaciones para el software y hardware, teniendo en cuenta las etapas que se muestran en la figura 1-2 con sus componentes.

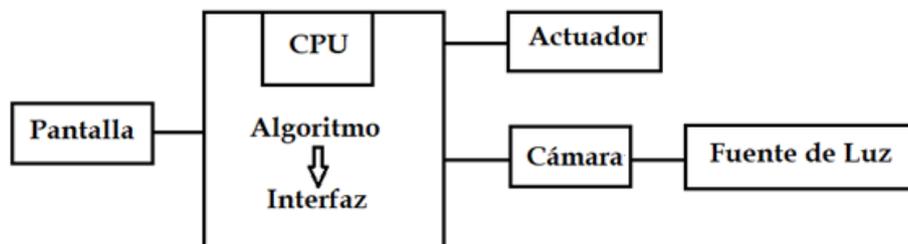


Figura 1-2: Componentes del Dispositivo
Realizado por: Maldonado, L. 2018

- **Pantalla:** Por medio de ésta se tendrá acceso a la interfaz gráfica en donde se selecciona la función que se desea según la necesidad del usuario.
- **CPU:** Ejecuta el algoritmo escogido por el usuario en la interfaz gráfica y controla los periféricos de entrada y salida. Previo al diseño del algoritmo se instala el sistema operativo herramientas y programas necesarios para su ejecución.
- **Cámara:** En una base se colocará la cámara para capturar la imagen del objetivo según la fuente de luz a la que este expuesto. La comunicación entre las cámaras y la Raspberry será I²C para obtener y configurar el registro de comando, control y SPI para capturar imágenes.

- **Actuador:** Según la función seleccionada por la interfaz gráfica se establece su funcionamiento dependiendo de la programación.
- **Fuente de Luz:** La iluminación es un factor importante ya que de ésta dependerá el nivel de radiación a la que estará expuesto el objetivo y así poder registrar los datos de cada pixel y así obtener una imagen de alta calidad.

2.2 Selección del Hardware

Existen cámaras termográficas disponibles en el mercado, pero su alto precio no permite que esté al alcance de todos los usuarios, por eso, este dispositivo tratará de emular una. Para la cual se necesita utilizar equipos y materiales de menor tamaño posible que sean de fácil acceso y que no sobrepasen el precio de una cámara térmica existente en el mercado. El dispositivo no tiene una aplicación específica así que se dejará abierto el tema acerca del objetivo a detectar.

- **Pantalla:** Se utilizará una pantalla táctil de 5” como se muestra en la figura 2-2 pero, en el diseño del algoritmo y la interfaz gráfica se utilizará una pantalla de 14” y un teclado USB para tener una mejor comodidad al momento de ingresar la información y la visualización de los mismos.



Figura 2-2: Pantalla táctil HDMI de 5”
Fuente: <https://bit.ly/2qmDDIw>

- **Computadora:** Se usará una Raspberry Pi 3 Modelo B donde se instalará el sistema operativo, se desarrollará el algoritmo y diseñará la interfaz gráfica para controlar los datos de entrada y salida. Sus características, como se muestran en la tabla 2-1 lo hacen ideal para el dispositivo.



Figura 3-2: Raspberry Pi 3 Modelo B
Fuente: <https://bit.ly/2PvXNOM>

- **Cámaras:** Para el dispositivo se van a utilizar dos tipos de cámaras, la Camera Board V2 y la cámara Radiométrica FLIR Lepton 2.5 como se mencionó anteriormente.
- **Módulo Buzzer:** Se va a utilizar un buzzer activo ya que posee un circuito integrado que permite emitir un sonido sólo con activar el pin de señal, el propósito será alertar al usuario cuando los sensores de la cámara térmica FLIR Lepton 2.5 lleguen a los límites de su sensibilidad.



Figura 4-2: Módulo Buzzer Activo
Fuente: <https://bit.ly/2SD1DEz>

2.3 Selección del Software

El algoritmo se va a diseñar con el entorno de programación Qt, mediante el uso de librerías propias del programa se puede diseñar la interfaz gráfica para el dispositivo. Entre estas librerías esta leptonSDKEmb32PUB el cual está dedicado a procesos que puede realizar la cámara Flir Lepton 2.5 que añade capacidades de procesamiento de imágenes.

2.4 Desarrollo del Algoritmo

2.4.1 Instalación del Software necesario en la Raspberry Pi

2.4.1.1 Instalación del Sistema Operativo

Existen varios sistemas operativos compatibles con la Raspberry Pi, pero se va utilizar Raspbian, un sistema operativo basado en Linux ya que este sistema ocupa menos recursos y es óptimo para el modelo de la Raspberry Pi.

Una vez que se descarga la imagen ISO del Sistema Operativo mediante el programa Etcher permite escribir una imagen de disco sin formato en un dispositivo extraíble en el cual se instalará el sistema en la Tarjeta Micro-SD desde la computadora.

Cuando se finaliza el proceso la tarjeta de memoria estará lista para ser utilizada en la Raspberry Pi.

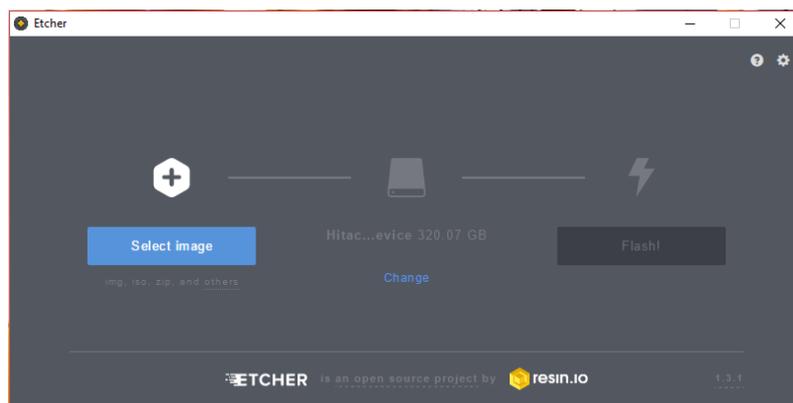


Figura 5-2: Instalación del S.O. Raspbian
Realizado por: Maldonado, L. 2018

Al iniciar el sistema operativo en la Raspberry Pi es necesario realizar algunas configuraciones como el idioma, región, teclado y habilitar el uso de la cámara Board V2 e interfaces que permiten la comunicación entre la cámara Radiométrica FLIR Lepton 2.5 y los puertos de la Raspberry Pi. Para ello se puede acceder a la configuración ingresando el comando “sudo raspi-config” desde la terminal.



Figura 6-2: Iniciando Raspbian
Realizado por: Maldonado, L. 2018

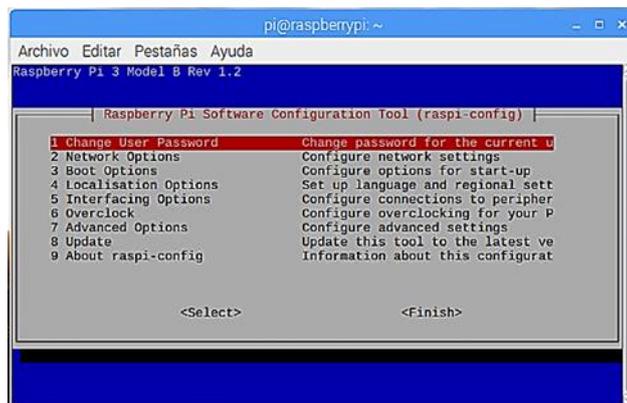


Figura 7-2: Configurando Raspbian
Realizado por: Maldonado, L. 2018

2.4.2 Consideraciones para el diseño del dispositivo

2.4.2.1 Área de Trabajo

El dispositivo se puede utilizar en áreas domesticas e industriales mientras que la temperatura ambiente de trabajo se encuentra entre los -10° y 65°C , en este rango las cámaras pueden trabajar conjuntamente.

2.4.2.2 Objetivos a Detectar

El dispositivo es capaz de detectar cualquier objetivo en la escena siempre y cuando la fuente de radiación a detectar este dentro de los límites de temperatura de la cámara infrarroja es decir entre -10° y 180°C

2.4.2.3 Variables de Entrada

Para el dispositivo existen dos variables de entrada para que sean procesadas por el algoritmo:

Luminosidad: En cada pixel de la cámara Board V2 detectará una señal de luz que se convertirá en una señal eléctrica.

Radiación Infrarroja: En cada pixel de la cámara FLIR Lepton 2.5 se detectará una señal de radiación infrarroja que se convertirá en una señal eléctrica.

4.1.2.4 Variables de Salida

Los datos obtenidos a través de la señal registrada por la cámara Board V2, los datos de radiación térmica obtenidos por la cámara Flir Lepton 2.5 y la señal que activa el módulo buzzer.

4.1.2.4 Funciones de la Interfaz

Antes de Diseñar el código primero se establece las funciones que tendrá la interfaz:

1. Un display en donde se pueda visualizar la imagen del objetivo en la escena.
2. Un botón que permita guardar la imagen térmica del objetivo en la escena.
3. Un botón que permita corregir el campo plano ya que según las condiciones ambientales a la que está expuesto el objetivo, es necesario actualizar el campo de visión y corregir ciertas perturbaciones.
4. Un display en donde se muestre la temperatura de la Matriz de Campo Focal (FPA) de la cámara.
5. Un display en donde se muestra la temperatura de la cámara.
6. Un display en donde se muestra la temperatura máxima de la escena.
7. Un display en donde se muestra la temperatura mínima de la escena.

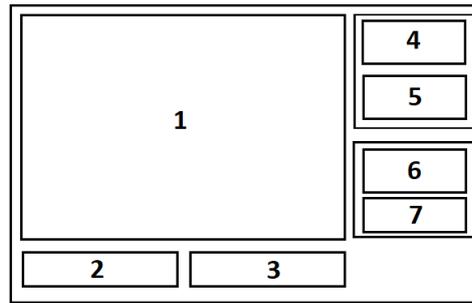


Figura 8-2: Diseño de la Interfaz Gráfica
Realizado por: Maldonado, L. 2018

2.4.3 Diseño de la Interfaz Gráfica

Para empezar a trabajar con Qt se necesita dos prerequisites: saber programar en C++ y conocer la técnica de programación orientada a objetos.

Para usar Qt en la Raspberry Pi es necesario descargar desde la terminal ingresando los siguientes comandos:

- sudo apt-get update
- sudo apt-get upgrade
- sudo apt-get install g++
- sudo apt-get install qt4-dev-tools
- sudo apt-get install qtcreator

Después se reinicia el sistema y Qt Creator se puede abrir desde la sección de programación la versión qt4 para su configuración.

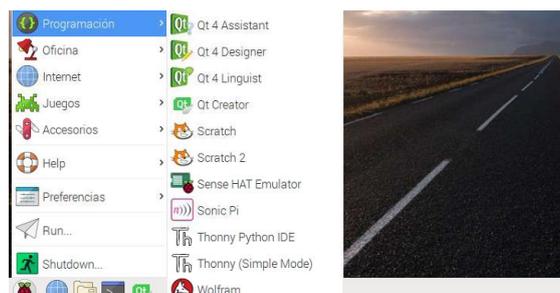


Figura 9-2: Ejecutando Qt Creator
Realizado por: Maldonado, L. 2018

Qt permite crear fácilmente interfaces gráficas por el método de arrastrar y soltar componentes como botones, labels, textbox, etc. Además, permite la depuración visual de los programas por medio de la simulación del programa paso a paso y al utilizar C++ como lenguaje de programación se puede acceder directamente a los puertos de la Raspberry Pi a través de librerías.

Para comenzar se inicia un nuevo proyecto con el nombre “Lepton” en donde se crean cuatro tipos de archivos con el nombre del proyecto por defecto:

- Lepton.pro: Un archivo en donde se especifican los formularios, clases y librerías que se van a utilizar en este proyecto. Por defecto se añade la librería del núcleo de Qt mediante la notación “+=” seguido de la librería a añadir si se trabaja con librerías externa como en este caso se deben añadir en este archivo para que no acurran errores al momento de compilar el proyecto.
- Lepton.h: Son archivos de cabecera que contienen elementos específicos de funciones y variables del proyecto que deben procesar, estos archivos se almacenan en una carpeta llamada “Sources”.
- Lepton.cpp: Son archivos fuente que sirven para compilar en el proyecto, aquí se elaboran las funciones y procesos que se van a llamar desde los archivos cabecera, estos archivos se almacenan en una carpeta llamada “Headers”.
- Lepton.ui: Este archivo está ubicada dentro de una carpeta llamada “Forms”, es aquí en donde se definen los componentes de la GUI, no es necesario manipular este código ya que la herramienta Qt Designer se encarga de ello automáticamente. A continuación, se detallan los elementos que se utilizaron para diseñar la GUI como se muestra en la figura 2-11:
 - La GUI se compone de un Witget en donde se muestra la imagen térmica del objetivo dentro de la escena.
 - Un Frame que contiene un PushButtom que, al momento de ser presionado pueda almacenar la imagen térmica de la escena.

- Un Frame que contiene un PushButton que, al momento de ser presionado pueda corregir y actualizar el FFC.
- Un Frame con el nombre “Temp. Camara” que contiene un GroupBox con un conjunto de Labels y dos PlainTextEdit que van a mostrar los valores de la Temperatura de la Cámara y la temperatura de la FPA.
- Un Frame con el nombre “Temp. Escena” que contiene un GroupBox con un conjunto de Labels y dos PlainTextEdit que van a mostrar los valores de temperatura máxima y mínima detectada en la escena.

Todos estos componentes estarán en un Widget principal con un tamaño definido de modo que no se pueda expandir la ventana del mismo sin que se alteren la posición de sus elementos los cuales se van a enlazar con los archivos de cabecera y fuente para que cada elemento pueda mostrar los procesos a los que fueron asignados.

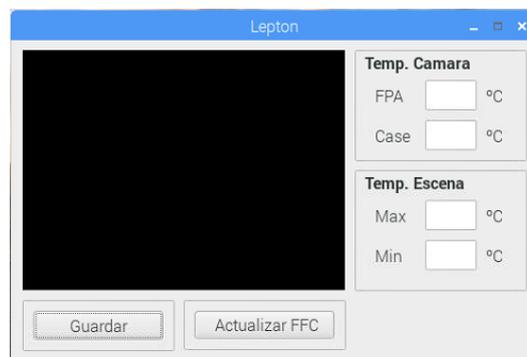


Figura 10-2: Diseño de la GUI en Qt Creator
Realizado por: Maldonado, L. 2018

2.4.4 Diseño de Códigos de los Entornos de Programación

2.4.4.1 Diseño del Algoritmo

Una vez establecidos los elementos que va a contener la interfaz se procede a realizar el algoritmo como se muestra en la figura 2-11, el diagrama de flujo con sus etapas que se detallan continuación.

- **Lectura de Datos:** En esta etapa mientras el puerto SPI este habilitado se procederá a leer los datos del puerto I²C para su procesamiento correspondiente.
- **Procesamiento de Datos:** Cada dato toma un valor de tipo UNIT16 según su fila y columna en donde se almacenan en un vector, mientras estos datos se almacenan se comparan para obtener el valor máximo y mínimo de temperatura y se calculan los valores de cada pixel para el formato de la imagen en RGB de modo que pueda formar la imagen de la escena.
- **Visualización de la Imagen y Variables:** Se observa la imagen del objetivo con los valores de temperatura tanto de la cámara como de la escena.
- **Corregir FFC:** Se corrige el campo plano de la escena si no es necesario sigue a la siguiente etapa.
- **Guardar:** Se almacena la imagen en formato JPG si no, sigue a la siguiente etapa.
- **Salir:** La Aplicación se cierra, si no cumple lo establecido regresa a la lectura de datos.

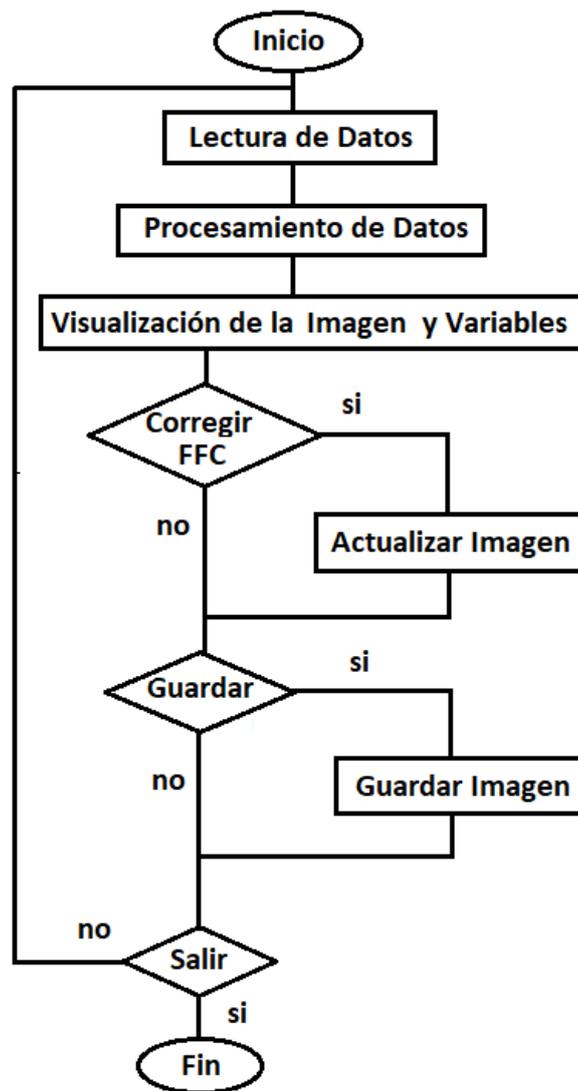


Figura 11-2: Diagrama de flujo base para los algoritmos
 Realizado por: Maldonado, L. 2018

2.4.4.1 Desarrollo del Código para el Algoritmo

Con el diagrama de flujo definido se puede desarrollar el código en los archivos fuente y de cabecera del proyecto. La cámara radiométrica Flir Lepton 2.5 cuenta con un archivo que permite crear aplicaciones con el uso de su librería este se puede descargar del siguiente enlace en la opción de documentos:

<https://lepton.flir.com/sdk/>

Los fabricantes que desarrollaron la cámara Flir Lepton 2.5 crearon una librería para utilizar comandos para poder comunicar la cámara con las interfaces I²C y SPI de la Raspberry Pi. La librería se puede obtener del siguiente enlace:

<https://lepton.flir.com/sdk/>

Los archivos se descargan y colocan en una carpeta con el nombre “leptónSDKEmb32PUB”, se copian a la carpeta donde se creó el proyecto y se compila desde la terminal con el comando “make”.

Para utilizar buzzer en Qt Creator es necesario descargar una librería llamada “wiringPi” la cual se puede descargar desde la terminal con los siguientes comandos:

- sudo apt-get install git-core
- git clone git://git.drogon.net/wiringPi
- cd wiringPi
- ./build

Esta librería nos permite habilitar los puertos GPIO de la Raspberry Pi y manipularlos como salida para alguna señal.

Previo a la declaración de las librerías externas que se van utilizar en el archivo “Lepton.pro” se desarrolló del algoritmo del dispositivo se crearon nuevos archivos fuente y de cabecera los cuales se detallan a continuación:

- **SPI:** Habilita la Interfaz Serial Periférica donde se establece principalmente el modo, el número de bit por palabra y la velocidad de lectura de datos. En el proyecto se eligió el modo 3 el cual inhabilita el reloj o pin CKL en un nivel alto y envía un 0 para que habilite la interfaz SPI y la deshabilite cuando envíe un 1 con una lectura de 8 bits por palabra a una velocidad de 10MHz es decir 1 μ S por bit. Adicionalmente se crearon varias funciones que muestran un mensaje de alerta si se presenta algún error.
- **Lepton_I2C:** Ya que para comunicarse con la cámara Flir Lepton requiere la apertura de un puerto de comunicación compatible se elabora un código que permita realizar la apertura del puerto donde se especifica la interfaz, velocidad en baudios deseada y

devuelve un descriptor de puerto para usar con todas las demás API de la librería SDK. Las funciones que permiten leer la temperatura de la cámara y la temperatura de la matriz de campo focal (FPA) se crean para ejecutarlas más adelante en la GUI.

- La cámara Flir Lepton posee algunas funciones que vienen habilitadas por defecto como la radiometría, el control automático de ganancia (AGC).
- El algoritmo AGC convierte la imagen térmica de resolución completa (14 bits) en una imagen mejorada en contraste adecuada para la visualización es una función de mapeo no lineal basada en histograma.
- Adicionalmente se creó una función que permita normalizar la imagen térmica mediante la corrección del campo plano (FFC). Este proceso es necesario, para tener una cierta independencia de las propiedades de la imagen, como lo son el brillo y el contraste. La ecuación para este proceso se muestra continuación:

$$N(x,y) = \frac{(N^{\circ} \text{ Niveles} - 1)}{(\max(I) - \min(I))} * (I(x,y) - \min(I))$$

Donde:

$I(x,y)$: Nivel de color de la imagen en la coordenada (x,y).

$\min(I)$, $\max(I)$: Mínimo y máximo nivel de color en la imagen respectivamente.

$N(x,y)$: Nivel de color de la imagen normalizada en la coordenada (x,y).

Estas variables serán especificadas más adelante para poder llamar la función.

- **LeptonThread**: Este archivo fuente contiene el algoritmo principal del proyecto ya que aquí se construye la imagen y se establece el formato para ello, se habilita el puerto SPI, se leen datos obtenidos por la interfaz I²C para que se puedan almacenar en una matriz digital de 80x60. Para realizar una nueva lectura datos se debe establecer un tiempo de espera el cual es de 750 ms.
- Estos datos se modifican de tal manera que se puedan comparar con los valores máximos y mínimos definidos asignando un valor por cada pixel y se almacenan junto con otras variables para realizar el proceso de normalización de la imagen. Existen

diferentes técnicas para representar una imagen digital para este proyecto se eligió procedimiento para generar una imagen de tipo RGB en donde cada pixel consta de tres valores (Rojo, Verde y Azul). Mientras se almacena los datos se comparan entre sí para determinar el valor máximo y mínimo de temperatura detectada en la escena, además, estos valores se usarán para activar el módulo buzzer según su programación. Adicionalmente se crearon las funciones para guardar la imagen y ejecutar las funciones que se crearon el archivo fuente “Lepton_I2C”.

- **MyLabel:** Posee una función que crea el espacio para que la imagen generada por el archivo fuente “LeptonThread” pueda observarse en la pantalla de la GUI y se actualiza cada vez que la matriz de 80x60 genera una nueva imagen.
- **Lepton:** En este archivo se llaman las funciones y variables que se crearon en los archivos anteriores para que interactúen con los elementos que se diseñaron en la interfaz gráfica.

El uso de los comandos de las librerías junto con los códigos programación en lenguaje en C++ de los archivos fuente y cabecera permiten visualizar los datos de las variables en la consola para que después se puedan enlazar con la GUI.

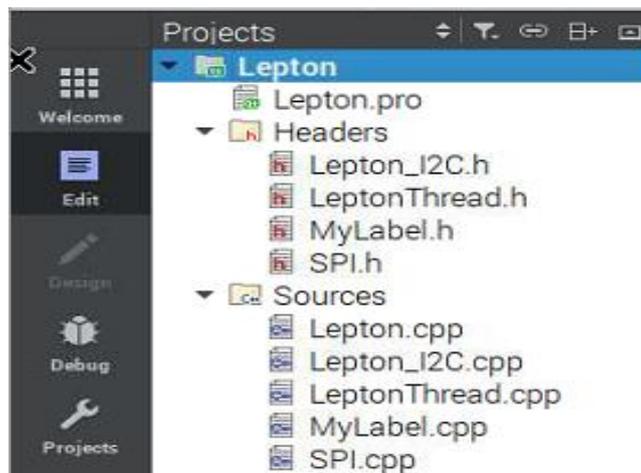


Figura 12-2: Archivos fuente y cabecera del proyecto
Realizado por: Maldonado, L. 2018

2.5 Implementación del Dispositivo

Una vez habilitada la cámara y las interfaces I²C y SPI se conecta el módulo de la cámara Flir Lepton 2.5 con los pines de la Raspberry Pi 3B para su comunicación como se muestran en la tabla 1-2 y figura 13-2 a continuación:

Tabla 1-2: Conexión Cámara Flir Lepton a Raspberry

Pin Cámara	Pin Raspberry	Nombre Pin
CS	24	CE0
MOSI	19	MOSI
MISO	21	MISO
CLK	23	CLK
GND	6	GND
VIN	1	3V3
SDA	3	SDA
SCL	5	SCL

Realizado por: Maldonado, L. 2018

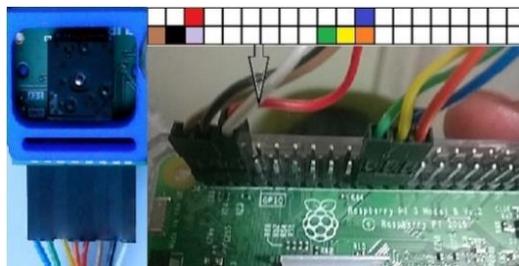


Figura 13-2: Conexión Cámara Flir Lepton
Realizado por: Maldonado, L. 2018

Se conecta la Cámara Raspberry Pi Camera Board V2 al puerto IO.



Figura 14-2: Conexión Cámara Board V2
Realizado por: Maldonado, L. 2018

Los pines del módulo buzzer se conectan en los pines que se van a configurar en el archivo fuente “LeptonThread” como se muestra en la tabla 2-2 y figura 15-2.

Tabla 2-2: Conexión Módulo Buzzer a Raspberry

Pin M. Buzzer	Pin Raspberry	Nombre Pin
VCC	2	5V
GND	34	GND
SIGNAL	32	GPIO 12

Realizado por: Maldonado, L. 2018

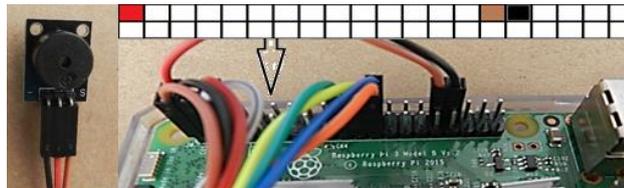


Figura 15-2: Conexión del módulo buzzer
Realizado por: Maldonado, L. 2018

Se conecta la pantalla HDMI de 5”, la cual para utilizarla con la Raspberry se debe instalar previamente los drivers e ingresar algunos comandos en el siguiente orden a través de la terminal.

- `git clone https://github.com/goodtft/LCD-show.git`
- `chmod -R 755 LCD-show`
- `cd LCD-show/`
- `sudo ./LCD5-show`

Una vez ingresado los comandos a la Raspberry se reinicia y configura para utilizarlo como dispositivo de salida.



Figura 16-2: Conexión de Pantalla HDMI de 5”
Realizado por: Maldonado, L. 2018

Debido a que la cámara Flir Lepton utiliza algunos puertos de la Raspberry la función táctil de la pantalla no se puede utilizar de modo que se debe utilizar un mouse inalámbrico como dispositivo de entrada.

Para la alimentación de la Raspberry y la Pantalla se utiliza un cargador de 5V a 2.5A conectada a un módulo de carga que alimenta una batería para que se pueda usar el dispositivo ya sea de forma fija o portátil.

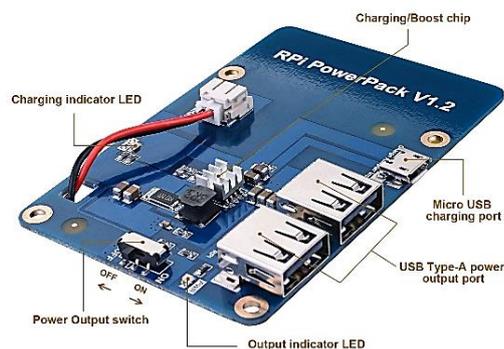


Figura 17-2: Módulo de carga para Raspberry
Fuente: <https://bit.ly/2PuXtzQ>

CAPÍTULO III

3. PRUEBAS Y RESULTADOS

3.1 Pruebas de envío y adquisición de datos

3.1.1 Interfaz I²C

Para comprobar que los datos son adquiridos por la interfaz I²C es necesario instalar una herramienta que ayudara a depurar el bus ingresado desde la terminal el comando “sudo apt-get install i2c-tools”. Después de reiniciar el sistema se abre nuevamente la terminal y se ingresa el comando “sudo i2cdetect -y 1” para comprobar la comunicación entre la interfaz I²C y la Raspberry Pi



```
pi@raspberrypi: ~
Archivo Editar Pestañas Ayuda
pi@raspberrypi:~$ sudo i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- 2a -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
pi@raspberrypi:~$
```

Figura 1-3: Lectura Interfaz I²C
Realizado por: Maldonado, L. 2018

3.1.2 Puertos GPIO

Para poder utilizar el módulo buzzer se debe descargar la librería “wiringPi” la misma que se declara en el archivo “Lepton.pro”, dentro de la función para crear la imagen térmica del objetivo se incluye un pequeño código definiendo el pin 32 o GPIO 12 de la Raspberry Pi como salida para que active el módulo buzzer. La activación se dará cuando en la escena se detecten niveles que estén por encima de los límites de temperatura de la cámara infrarroja es decir menor de -10°C y mayor de 180°C. Para comprobar que se realizó la configuración del pin de salida ingresando el comando “gpio readall” como se muestra en la imagen en la figura 2-3 confirmando así el resultado esperado.

```

pi@raspberrypi:~$ sudo gpio readall
-----Pi 3-----
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 2 | 8 | 3.3v | | | 1 | 2 | | | 5v | | |
| 3 | 8 | SDA.1 | ALTO | 1 | 3 | 4 | | | 5v | | |
| 4 | 9 | SCL.1 | ALTO | 1 | 5 | 6 | | | 0v | | |
| 7 | 7 | GPIO.7 | IN | 1 | 7 | 8 | 0 | IN | TxD | 15 | 14 |
| | | | | | 9 | 10 | 1 | IN | RxD | 16 | 15 |
| 17 | 0 | GPIO.0 | IN | 0 | 11 | 12 | 0 | IN | GPIO.1 | 1 | 18 |
| 27 | 2 | GPIO.2 | IN | 0 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO.3 | IN | 0 | 15 | 16 | 0 | IN | GPIO.4 | 4 | 23 |
| | | | | | 17 | 18 | 0 | IN | GPIO.5 | 5 | 24 |
| 10 | 12 | 3.3v | | | 19 | 20 | | | 0v | | |
| 9 | 13 | MOSI | ALTO | 0 | 21 | 22 | 0 | IN | GPIO.6 | 6 | 25 |
| 11 | 14 | MISO | ALTO | 1 | 23 | 24 | 1 | OUT | CEO | 10 | 8 |
| | | | | | 25 | 26 | 1 | OUT | CE1 | 11 | 7 |
| 0 | 30 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 5 | 21 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 6 | 22 | GPIO.22 | IN | 1 | 31 | 32 | 0 | OUT | GPIO.26 | 26 | 12 |
| 13 | 23 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| 19 | 24 | GPIO.24 | IN | 0 | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| 26 | 25 | GPIO.25 | IN | 0 | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

Figura 2-3: Lectura de pines de la Raspberry Pi
Realizado por: Maldonado, L. 2018

3.2 Ajustes de las Variables

A través de la interfaz I²C la cámara envía valores enteros comprendidos en 0 y 65535, éstos se guardan en una variable llamada “value” para compararlas con las variables “maxValue” y “minValue” para obtener los valores máximos y mínimos de temperatura en la escena. Con una lectura de 1036 valores almacenados adjuntos en el Anexo H se establecieron sus límites de modo que se puedan obtener datos que estén dentro del rango de detección de la cámara térmica como se muestra en la tabla 1-3.

Tabla 1-3: Valores Máximo y Mínimo Detectados

Valor Máximo Detectado	45464
Valor Mínimo Detectado	26300
Num. Valores Almacenados	1036

Realizado por: Maldonado, L. 2018

Ya que estos valores se expresan en Kelvin en una escala por 100 se deben transformar de tal forma que se muestren los valores en grados Celsius en la consola para después enlazarlos con la GUI. Para ello, se utilizaron las siguientes ecuaciones respectivamente:

$$x = (maxValue * 0.01) - 273$$

$$y = (minValue * 0.01) - 273$$

Donde:

x: es el valor de la Teperatura Máxima
y: es el valor de la Temperatura Mínima
maxValue: es el valor máximo detectado
minValue: es el valor mínimo detectado

Por medio de la librería SDK de la cámara se pueden obtener la temperatura de la Matriz Focal de Campo y la Cámara Térmica con las funciones que se crearon desde los archivos fuente y cabecera “Lepton_I2C” y se ejecutan desde la función los archivos fuente y cabecera “LepronThread” para que se pueda visualizar en la consola los valores en grados Celsius. Para la función que guarda la imagen térmica del objetivo se crearon las variables “*i_width*” y “*i_eight*” con los valores de 80 y 60 respectivamente de modo que al momento de almacenar se pueda observar a través el visor de imágenes del sistema operativo con las dimensiones definidas anteriormente.

3.3 Configuración de las Componentes

Para enlazar los displays de temperatura y los botones diseñados en la GUI se les asignarán un nombre para identificarlos fácilmente como se muestra en la tabla 2-3.

Tabla 2-3: Configuración de Componentes

Componente	Nombre	Función
PlainTextEdit	FPAtemp	Mostrar la temperatura de la FPA
PlainTextEdit	AuxTemp	Mostrar la temperatura de la cámara
PlainTextEdit	Tmax	Mostrar la temperatura máxima de la escena
PlainTextEdit	Tmin	Mostrar la temperatura mínima de la escena
PushButton	UpdateFFC	Corregir y actualizar la imagen térmica
PushButton	SaveButton	Almacenar la imagen térmica

Realizado por: Maldonado, L. 2018

Para el caso de los Label no es necesario asignar un nombre en particular ya que su función es mostrar una cadena de caracteres fijas. El mismo caso es para los Frames y GoupBox ya que éstos solo sirven para contener los PlainTextEdit y Label, a continuación, se muestra en la figura 3-3 la configuración de los componentes.

Object	Class
Lepton	QWidget
Buttons	QFrame
UpdateFFC	QPushButton
CameraTemp	QFrame
CameraTemperature	QGroupBox
AuxTemp	QPlainTextEdit
FPATemp	QPlainTextEdit
label_3	QLabel
label_4	QLabel
label_5	QLabel
label_6	QLabel
CameraTemp_2	QFrame
CameraTemperature_2	QGroupBox
Tmax	QPlainTextEdit
Tmin	QPlainTextEdit
label_10	QLabel
label_7	QLabel
label_8	QLabel
label_9	QLabel

Figura 3-3: Configuración de Componentes
Realizado por: Maldonado, L. 2018

3.4 Pruebas de Funcionamiento

3.4.1 Prueba Inicial de Funcionamiento

3.4.1.1 Visualización de la Imagen y Temperatura

Con el arreglo de las variables se ejecutó el algoritmo dando como resultado una imagen en la GUI, la visualización en la consola de la temperatura de la cámara FCC y la temperatura maxima y minima detectada en la escena como se muestra en la figura 4-3. Estas pruebas se realizaron con una pantalla externa de 14” y un teclado USB.

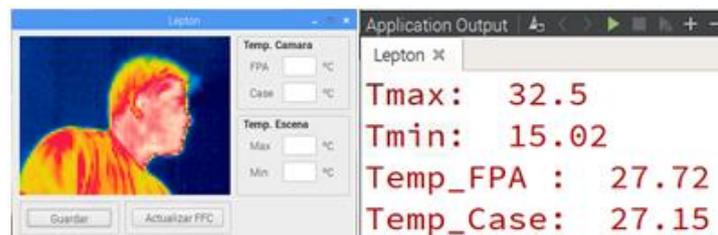


Figura 4-3: Prueba Inicial del Algoritmo
Realizado por: Maldonado, L. 2018

3.4.1.2 Almacenamiento de la Imagen

Para almacenar la imagen de la escena se debe enlazar la función que se creó en los archivos fuente y cabecera “LeptonThread” con el botón “SaveButton”, esto se puede realizar mediante el uso los componentes “*SIGNAL*” el cual detecta un cambio de estado en alguna de las propiedades de un PushButton en este caso al hacer clic sobre el mismo y “*SLOT*” que es una función C++ que es llamada en respuesta a un “*SIGNAL*”, para que esto sea posible estos componentes se deben conectar con una función llamada “*Object::connect()*”. A continuación, se muestra en la figura 5-3 el resultado del enlace.

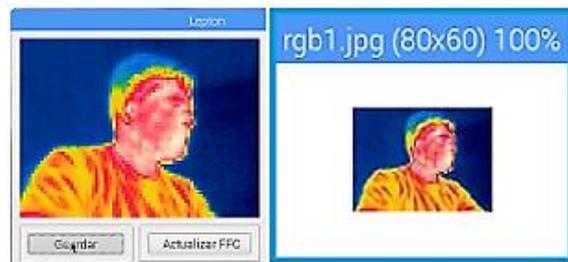


Figura 5-3: Prueba Inicial del Algoritmo
Realizado por: Maldonado, L. 2018

3.5 Errores en Lectura de Datos

3.5.1 Visualización de las Variables

Al momento de enlazar las variables con los displays que muestran los diferentes niveles de temperatura y con la existencia de un cambio en la temperatura máxima y mínima de la escena al igual que la temperatura de la cámara y de la FPA estos valores no se actualizaban en los displays, únicamente mostraba los primeros valores como se muestra en la figura 6-3.

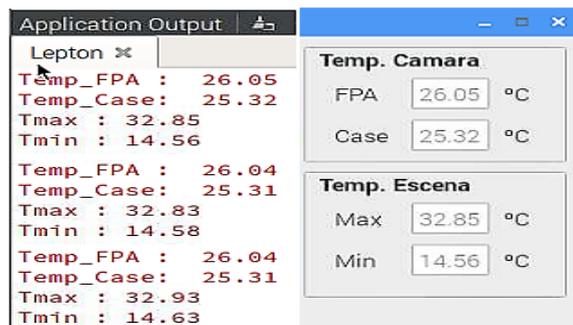


Figura 6-3: Error Visualización de las Variables
Realizado por: Maldonado, L. 2018

3.5.2 Uso de la Pantalla HDMI

Al probar el algoritmo con la pantalla HDMI de 5 pulgadas conectada a la Raspberry Pi se presentó un error, al instalar los controladores de la pantalla creó un conflicto con la cámara Flir Lepton. En el manual de usuario de la pantalla indicaba que si no se deseaba utilizar la función táctil sólo había que desconectarla de los pines de la Raspberry Pi y únicamente conectar el cable HDMI. Para el dispositivo si bien la función táctil brindaba mayor comodidad al momento de manipular la GUI no se podía utilizar ya que usa algunos puertos e interfaces que utiliza la Cámara Flir Lepton y aunque no se conecte la pantalla a los pines de la Raspberry Pi el sistema negaba la comunicación de la Cámara debido a la configuración de los controladores de la pantalla. El mensaje de error aparece al ejecutar el algoritmo como se muestra en la figura 7-3.

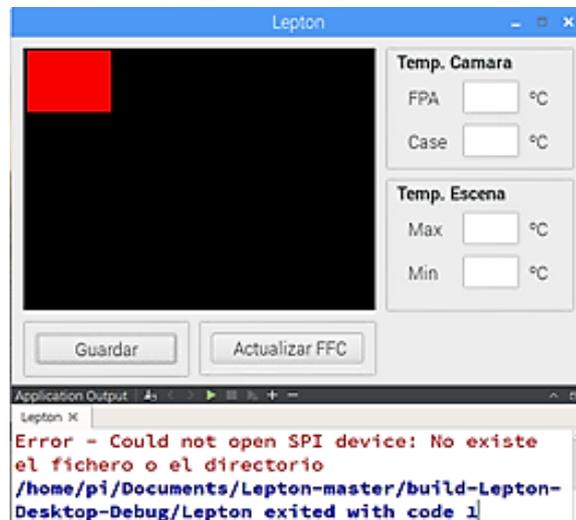


Figura 7-3: Error Visualización de las Variables
Realizado por: Maldonado, L. 2018

3.6 Corrección y Ajustes de Parámetros

3.6.1 Pantalla

Se consideró utilizar la pantalla de 5 pulgadas, pero debido a los errores que causaba en el algoritmo con la instalación de los controladores hubo la necesidad de reemplazarla de 5 pulgadas por una de 7 pulgadas con una resolución de 800x400. Esta pantalla es de uso exclusivo para Raspberry Pi por lo que no necesita instalar algún controlador ya que éstos vienen instalados en el sistema operativo lo que garantiza una compatibilidad evitando

conflictos con los pines de la Raspberry Pi y permitiendo la comunicación con la cámara Flir Lepton por medio del algoritmo. Para su uso necesita dos conexiones una a través del puerto DSI y la otra con dos terminales para alimentar la placa controladora de la pantalla.

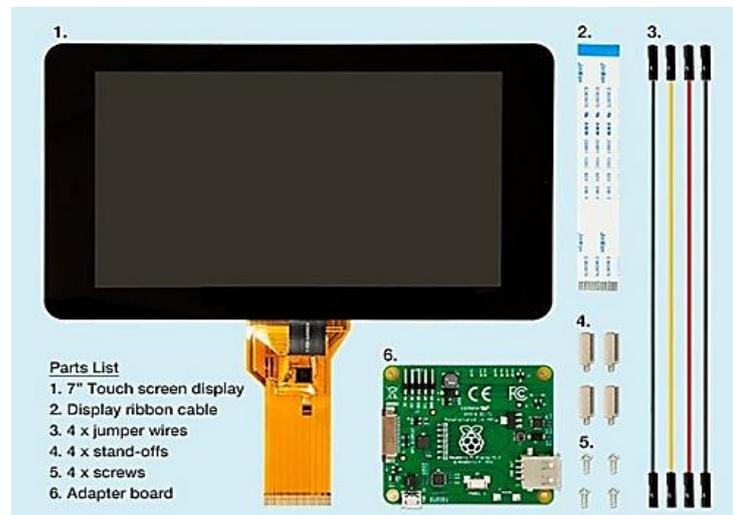


Figura 8-3: Pantalla oficial para Raspberry Pi
Fuente: <https://bit.ly/2yILTHI>

3.6.2 Actualización de los Datos

Para corregir el error de la actualización de los datos de temperatura se propuso desarrollar una función que utiliza parte del método de actualización de la imagen y la función “*Object::connect()*”. En este caso para la componente “SIGNAL” no funciona con una acción si no que debe emitir un valor de tipo String al igual que la componente “SLOT”.

Para utilizar las variables de temperatura se deben convertir sus datos y almacenarlos en nuevas variables que se podrán emitir como componentes “SIGNAL” después se deben declarar en el archivo de cabecera en donde se crearon las variables es decir el archivo “LeptonThread.ccp” como funciones de tipo String para en enlazarlo con los PlainTextEdit donde se visualizaran estos valores. Se realizó la prueba con una variable “*tmax*” la cual contiene los valores de “x” convertidos en una variable de tipo String para que se pueda emitir como función obteniendo el resultado esperado, después se aplicó el mismo procedimiento para las variables de la temperatura mínima, FPA y de la cámara.

3.6.3 Escala y Datos de la Imagen

La imagen que se almacena tiene una dimensión de 80x60 píxeles al momento de abrirla en el visor de imágenes se muestra de un tamaño que no se puede apreciar muy bien además de que no muestra ningún dato. Por eso se modificó el código de la función para que la imagen térmica sea almacenada a una escala 4 veces mayor a su tamaño real para que se pueda observar en el visor de imágenes mejorando su calidad con los datos de la temperatura máxima y mínima de la escena como se muestra en la figura 9-3.



Figura 9-3: Imagen Térmica 320x240 píxeles
Realizado por: Maldonado, L. 2018

3.6.4 Recursos Adicionales

Hasta ahora la GUI presenta las funciones que se establecieron con anterioridad, para que el usuario pueda comprender de mejor manera los datos de la imagen térmica se pretende ingresar una imagen con los tonos de color de menor a mayor de modo que el usuario pueda identificar los puntos con mayor y menor nivel de temperatura. Para ello se agrega en el proyecto un nuevo archivo de recursos en donde se configura y selecciona la imagen que se va a utilizar dando como resultado un archivo llamado “imagen.qrc” que se crea dentro de una carpeta “Resources”.

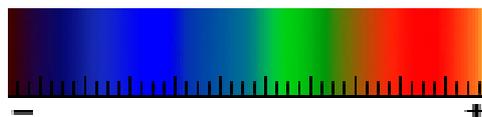


Figura 10-3: Imagen Espectro Infrarrojo
Realizado por: Maldonado, L. 2018

3.7 Resultados del Proyecto

3.7.1 Dispositivo Implementado

Una vez corregido los errores de los códigos de programación la Raspberry Pi, la cámara Flir Lepton, la cámara Board V2, la pantalla HDMI el módulo Buzzer y el módulo de carga se colocaron cuidadosamente en un estuche plástico de modo que el usuario pueda utilizar el dispositivo de forma fija o portátil. Adicionalmente se instaló un par de ventiladores para disipar el calor ya que al momento de trabajar con procesamiento de imágenes por un tiempo prolongado hace que aumente la temperatura del procesador de la Raspberry Pi.



Figura 11-3: Vista Lateral del Dispositivo
Realizado por: Maldonado, L. 2018



Figura 12-3: Vista Posterior del Dispositivo
Realizado por: Maldonado, L. 2018

3.7.2 Interfaz Gráfica

Con las correcciones, ajustes de parámetros y el uso de recursos adicionales la GUI definitiva que permite visualizar la imagen térmica de la escena se puede observar a continuación en la figura 13-3.



Figura 13-3: GUI del Dispositivo
Realizado por: Maldonado, L. 2018

3.7.3 Pruebas Finales

Por medio del entorno de programación Python diseñó un pequeño algoritmo que al ejecutarlo almacena una imagen en función de su radiación lumínica con la Cámara Board 2V con el fin de compararlas con la imagen tomada por la cámara Flir Lepton en función de su radiación infrarroja. A continuación, se detallan las pruebas que se realizaron en distintas condiciones climáticas y de trabajo.

3.7.2.1 Área Doméstica

Las cámaras normales no permiten tener una buena visión de los objetivos en la oscuridad y mediante la cámara térmica se puede visualizar varios objetivos ya sean seres vivos como personas y animales casi siempre con una temperatura corporal diferente al entorno que los rodea dando una opción alternativa como método de seguridad y vigilancia. Los resultados se muestran a continuación.



Figura 14-3: Prueba en personas
Realizado por: Maldonado, L. 2018

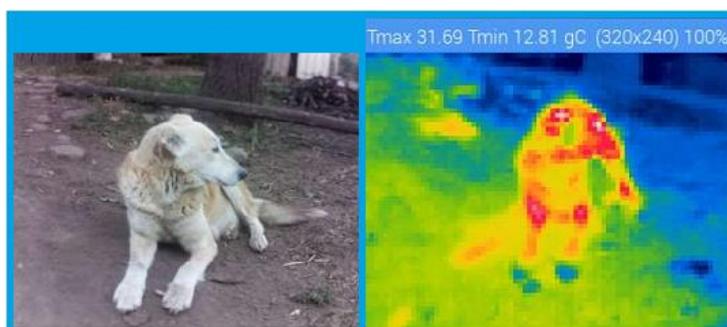


Figura 15-3: Prueba en animales
Realizado por: Maldonado, L. 2018

3.7.2.2 Área Industrial

Esta prueba se realizó mezcladora con doble pared calentada por resistencias y agua caliente en donde su temperatura era controlada por una termocupla. El objetivo de esta prueba era el comprobar el sistema de control de temperatura en la mezcladora estaba calentándose a la temperatura establecida por el usuario es decir 80 °C. Adicionalmente se realizó una inspección en la instalación eléctrica la cual mostraba un aumento de temperatura tanto en los cables, así como también en la conexión de la caja de breakers. Los Resultados de las pruebas de la mezcladora y la instalación eléctrica se muestran a continuación



Figura 16-3: Prueba en mezcladora con doble pared
Realizado por: Maldonado, L. 2018

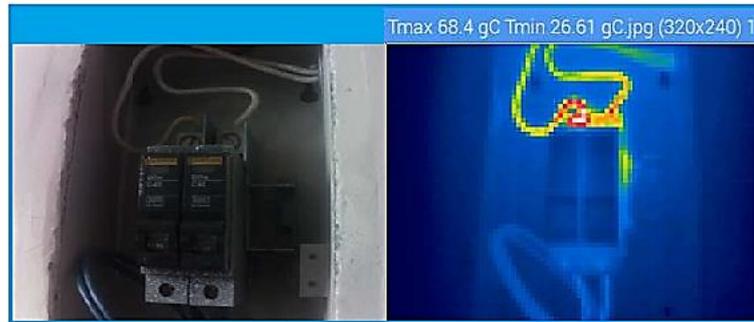


Figura 17-3: Prueba en instalación eléctrica
Realizado por: Maldonado, L. 2018

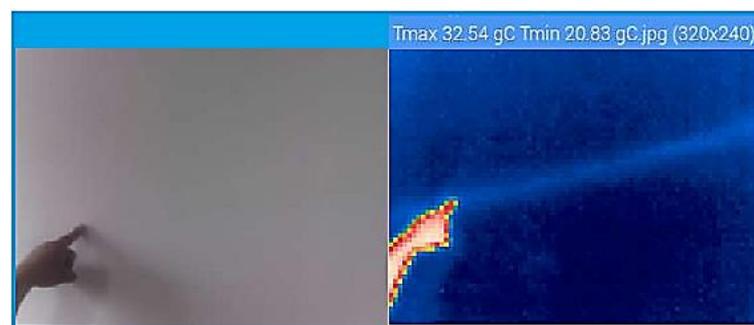


Figura 18-3: Prueba en cableado
Realizado por: Maldonado, L. 2018

3.7.2.3 *Objetivos a Corta Distancia*

Se colocaron un cuerpo frío y un cuerpo caliente con temperatura variable a una distancia de 0.5 m de la cámara térmica en espacio cerrado además de que la base en donde estaban situados los cuerpos tenía propiedades reflectivas. Por medio de la interfaz gráfica se pudo visualizar y almacenar la imagen con la temperatura máxima del cuerpo caliente y la temperatura mínima del cuerpo frío como se muestra en la figura 19-3.

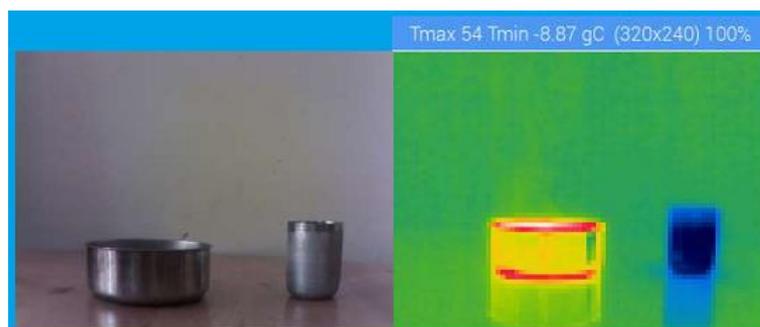


Figura 19-3: Prueba en objetivos a Corta distancia
Realizado por: Maldonado, L. 2018

3.7.2.4 Objetivos a Larga Distancia

Para esta prueba se utilizó un vehículo como objetivo a visualizar a una distancia de 10 m en un espacio abierto ya que los sensores en la cámara FLir Lepton posee una constante de tiempo de 12 ms de detección, es decir que un cambio instantáneo de irradiancia dará como resultado un cambio de temperatura del detector como se muestra en la figura 20-3.

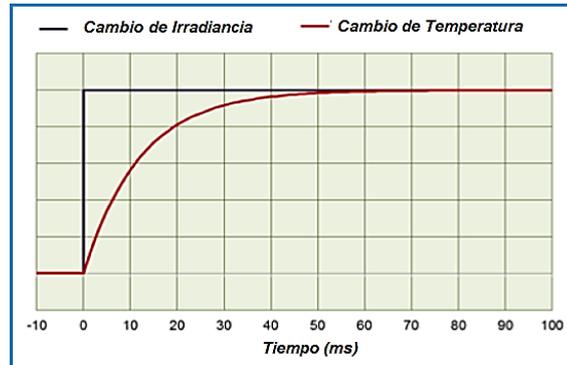


Figura 20-3: Relación Irradiancia-Temperatura
Realizado por: Maldonado, L. 2018

Para observar los valores de la temperatura máxima y mínima era necesario actualizar la Matriz de Campo Focal, además, la temperatura del entorno mantenía la temperatura máxima en la escena en vez del objetivo como se muestra a continuación.



Figura 21-3: Prueba en objetivo a larga distancia
Realizado por: Maldonado, L. 2018

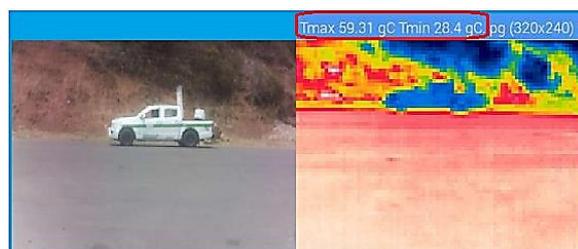


Figura 22-3: Prueba en objetivo FCC actualizada
Realizado por: Maldonado, L. 2018

3.8.3 Análisis de Resultados

Después de realizar las pruebas en las diferentes áreas se pudo observar que al presentarse factores como el humo y el polvo la función que actualiza la FPA es de gran utilidad ya que permite una actualización de los términos de corrección interna utilizados para mejorar la calidad de la imagen.

Otros factores como la distancia entre la cámara térmica y el objetivo a detectar, así como también sus dimensiones y la temperatura del entorno servirán para tener una mejor respuesta en la adquisición y visualización de los datos obtenidos.

Con el dispositivo conectado a una batería de 2 Ah con un módulo de carga que alimenta a la Raspberry pi y sus componentes que consumen una corriente de 1,05 A como se muestra en la figura 20-3 dando como resultado una autonomía de 1,9 horas.

$$\text{Autonomía} = \frac{2 \text{ Ah}}{1,05 \text{ A}}$$

$$\text{Autonomía} = 1,9 \text{ h}$$

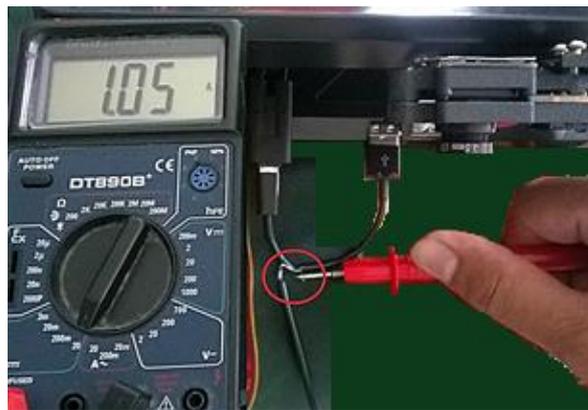


Figura 23-3: Lectura de Corriente del Dispositivo
Realizado por: Maldonado, L. 2018

CONCLUSIONES

- La selección de la Cámara Radiométrica Flir Lepton 2.5 y la Raspberry Pi 3B como componentes principales del dispositivo en base a la aproximación del método tecnológico evaluando y corrigiendo aspectos en el diseño al momento de detectar una fuente de radiación infrarroja y procesar sus datos.
- Mediante las pruebas se analizaron los límites del dispositivo y factores como la emisividad de la escena y la señal no deseada de fuentes distintas, las cuales influyen en la precisión de la medición de una escena real.
- El uso de la Raspberry Pi como unidad central de proceso en conjunto con el entorno de programación Qt permitió implementar un sistema de visión artificial de nivel bajo facilitando la comunicación entre los componentes del dispositivo.
- Al evaluar el dispositivo en diferentes escenarios se pudieron observar varias aplicaciones en el área industrial ya que se según la necesidad del usuario se pueden realizar pruebas en base a las imágenes capturadas o a los valores de temperatura alanceados.

RECOMENDACIONES

- Se deben tener medidas de seguridad al momento de utilizar el dispositivo ya que la cámara radiométrica Flir Lepton es susceptible a descargas electroestáticas y la adquisición de un repuesto llevará tiempo ya que no se comercializan dentro del país.
- Para aumentar el campo de visión del dispositivo se puede colocar la cámara en una base con servos con una función que permita controlar su posición final e inicial.
- Si el dispositivo se utiliza de manera fija se puede aprovechar la señal que emite la alarma para controlar un proceso adicional y agregarlo como una función a la interfaz gráfica.
- El usar el entorno de programación Qt como método de aprendizaje puede ser de gran utilidad para proyectos futuros ya que junto con la Raspberry Pi poseen varias aplicaciones entre ellas sistemas embebidos.

GLOSARIO

Baudios:	Unidad de medida de la velocidad de transmisión de señales.
Buzzers:	Dispositivos que convierten señales eléctricas en sonido.
Infrarrojo:	Tipo de radiación del espectro luminoso.
Irradiancia	Magnitud utilizada para describir la potencia incidente por unidad de superficie de todo tipo de radiación electromagnética.
Label:	Componente para diseño de interfaces gráficas.
Oscilador:	Dispositivo capaz de convertir la energía de corriente continua en corriente alterna a una determinada frecuencia.
Raspbian:	Sistema Operativo dedicado a Raspberry Pi
Raspberry Pi:	Mini computadora que por lo general se utiliza en proyectos de electrónica para interactuar con sensores y actuadores
Scripting:	Lenguaje de Programación
Sistema embebido:	Sistema de cómputo dedicado a realizar una o varias funciones en específico.
String:	Cadena de caracteres.

BIBLIOGRAFÍA

Gómez, F. y Domínguez, M. «Fundamentos de la Visión Artificial» (2016), pp. 1-22. (Consulta: 13 de marzo de 2018) Disponible en: <http://docplayer.es/13643465-Fundamentos-de-la-vision-artificial-prof-dr-francisco-gomez-rodriguez-prof-manuel-j-dominguez-morales-1.html>.

Llamas, L. *¿Qué es Raspberry Pi?* (2017) (Consulta: 31 de enero de 2018) Disponible en: <https://www.luisllamas.es/que-es-raspberry-pi/>.

Melgosa, S. «Guía de la termografía infrarroja: aplicaciones en ahorro y eficiencia energética» (2011), *Fenercom*, pp. 1-188. (Consulta: 2 de febrero de 2018) Disponible en: <https://www.fenercom.com/pdf/publicaciones/Guia-de-la-Termografia-Infrarroja-fenercom-2011.pdf>.

Ortega, E. «Clasificación de Sensores» (2013), pp. 1-20. (Consulta: 16 de enero de 2018) Disponible en: <http://ingemecanica.com/tutoriales/CategoriasdeVehiculos.html>.

Santiago, A. «Sistema de detección de objetos mediante cámaras. Aplicación en Espacios Inteligentes.» (2007), p. 244. (Consulta: 31 de enero de 2018) Disponible en: <http://www.geintra-uah.org/system/files/private/Proyecto.pdf>.

Testo «Termografía, guía de bolsillo Teoría – Aplicación Práctica – Consejos y Trucos» (2008), pp. 1-56. (Consulta: 16 de enero de 2018). Disponible en: http://www.ait-orsonor.com/testo_guia.pdf

Valencia, G. y Brito, V. «Implementación de un Control de Acceso con Reconocimiento Facial Mediante Opencv para la Sala de Profesores de la FIE.» Escuela Superior Politécnica de Chimborazo, (2015) pp. 1-139. (Consulta: 13 de marzo de 2018) Disponible en: <http://dspace.esPOCH.edu.ec/handle/123456789/5020>.

ANEXOS

ANEXO A

Instalación del Sistema Operativo Raspbian en la Tarjeta Micro-SD

Se ejecuta el programa “Etcher” desde la PC para seleccionar la imagen ISO del sistema operativo.

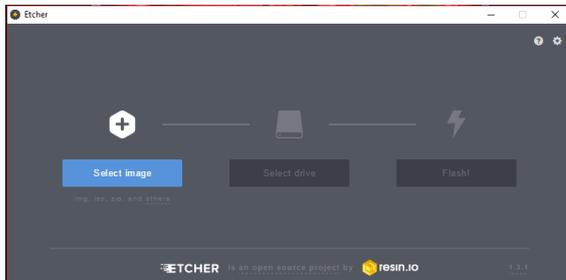


Figura 1: Selección de la Imagen ISO.

Se selecciona el dispositivo donde se desea instalar el Sistema Operativo, en este caso la Tarjeta Micro-SD.

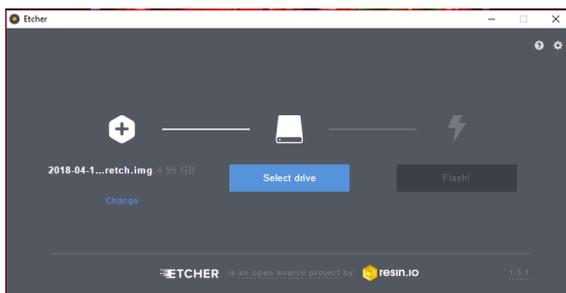


Figura 2: Selección del Dispositivo Extraíble.

Con el Sistema Operativo y el dispositivo seleccionado se presiona el botón “Flash”.



Figura 3: Proceso de Grabación.

Cuando finalice el proceso se mostrará la siguiente imagen para retirar la tarje Micro-SD de la PC y colocarla en la Raspberry Pi.

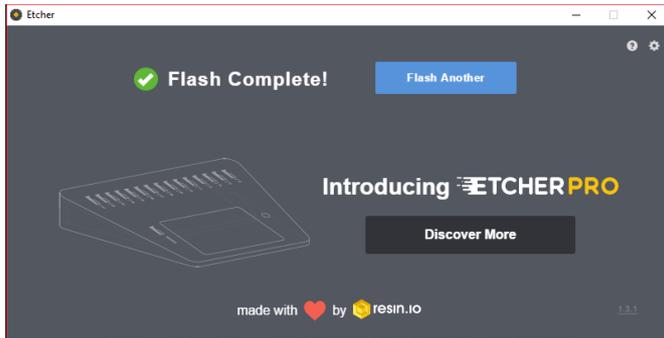


Figura 4: Flash Completo

ANEXO B

Configuración del Sistema Operativo Raspbian

Una vez que se inicia el escritorio ejecutar desde la barra de tareas la terminal

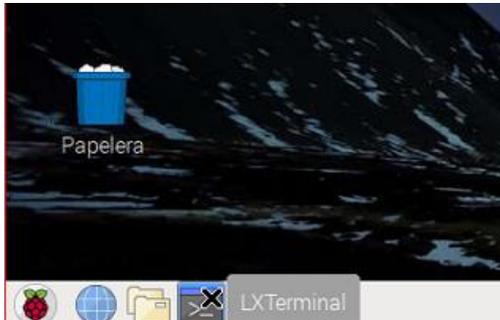


Figura 1: Ingreso a la Terminal

A continuación, para cambiar algunas de las configuraciones de Raspberry pi, se ingresa el siguiente comando para acceder a este menú: “sudo raspi-config”.

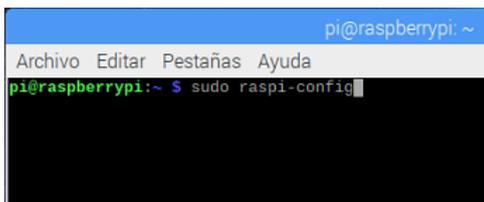


Figura 2: Configuración Raspbia

Se selecciona la opción 5 “Interfacing Options”.

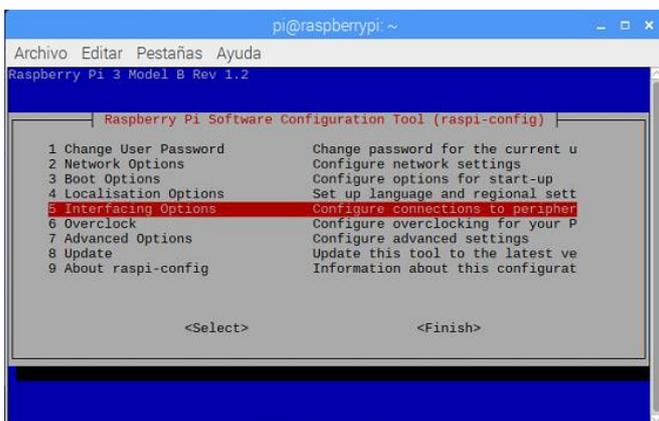


Figura 3: Opciones de Interfaz

Se habilitan las interfaces desde P1 a P5.

- **Cámara**

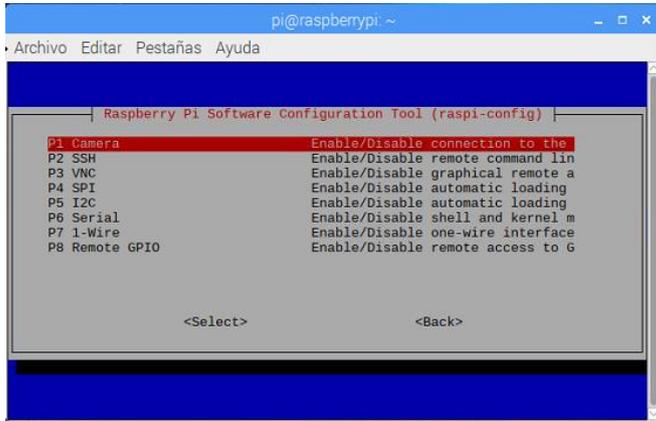


Figura 4: P1 Camera

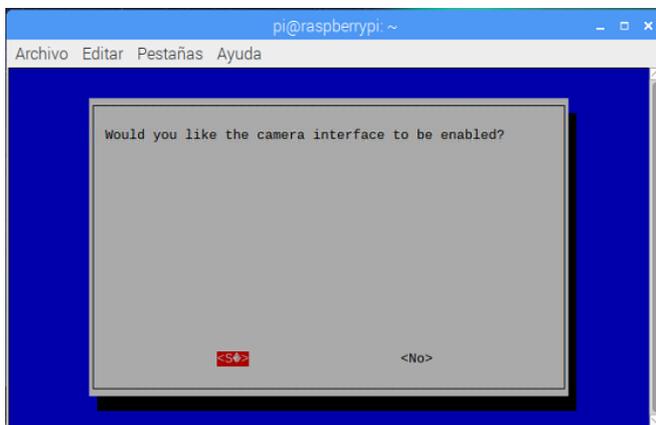


Figura 5: Habilitando Interfaz Cámara

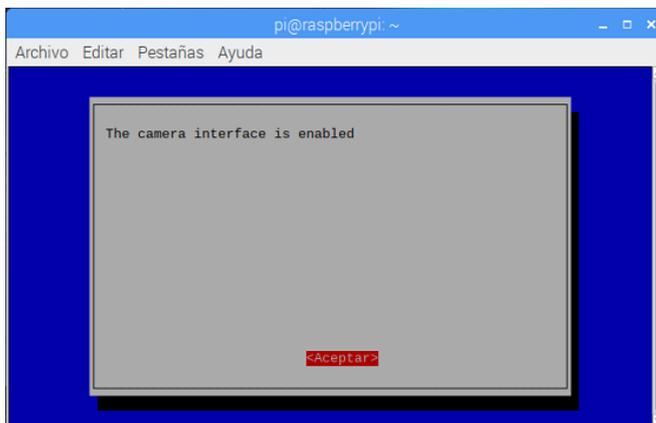


Figura 6: Interfaz Cámara Habilitada

- SSH

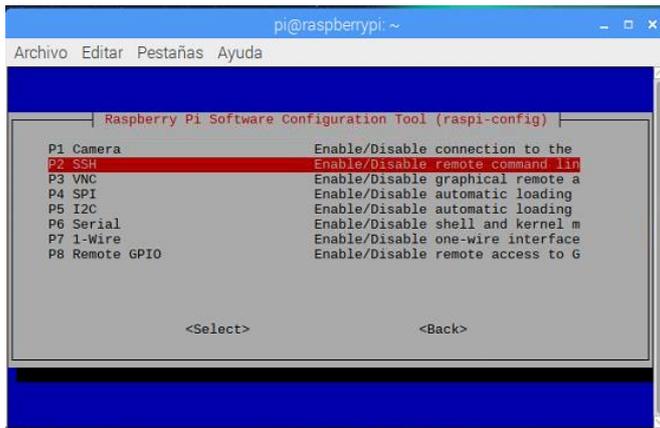


Figura 7: P2 SSH

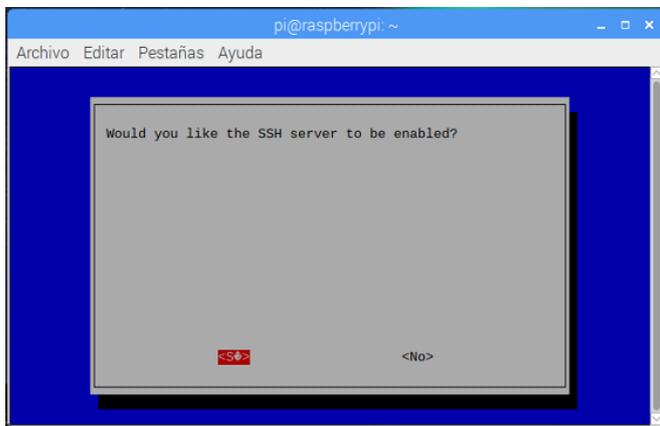


Figura 8: Habilitando Interfaz SSH

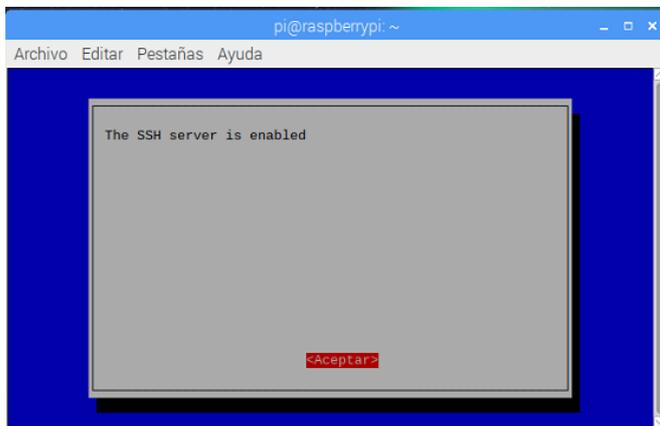


Figura 9: Interfaz SSH Habilitada

- VNC

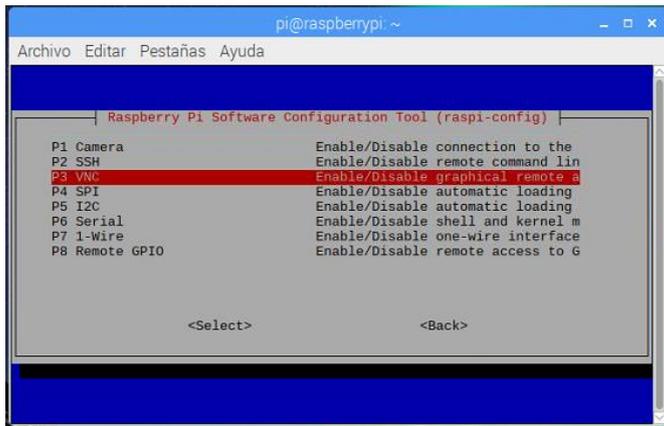


Figura 10: P3 VNC

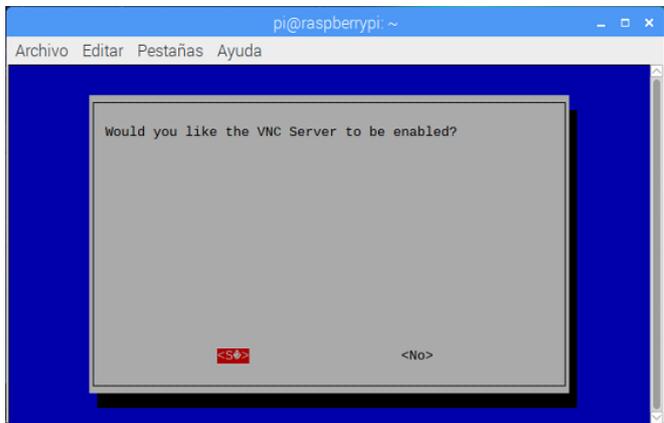


Figura 11: Habilitando Interfaz VNC

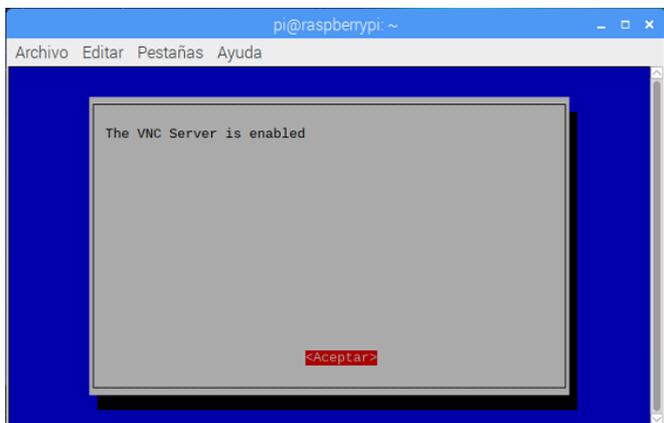


Figura 12: Interfaz VNC Habilitada

- SPI

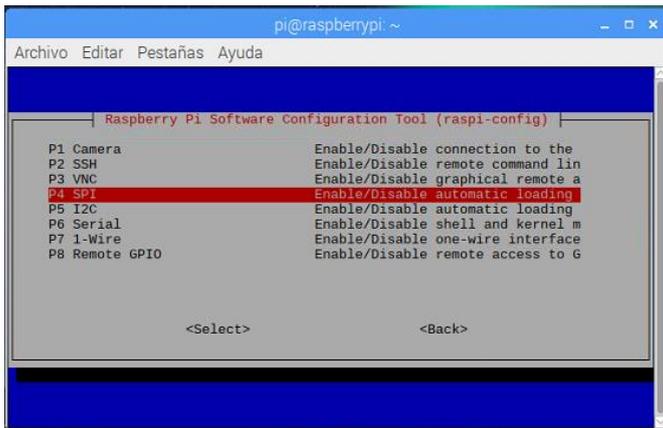


Figura 13: P4 SPI

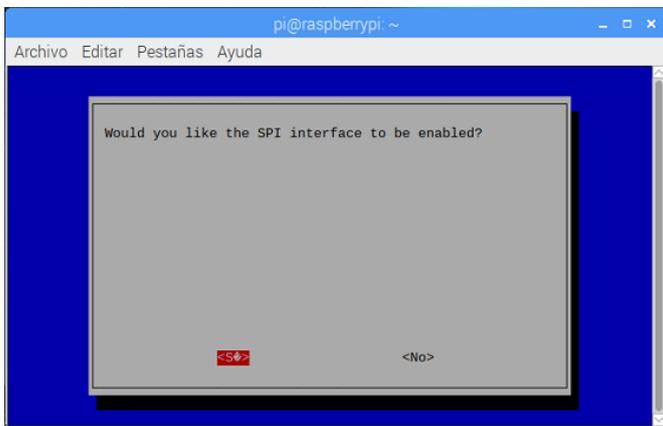


Figura 14: Habilitando Interfaz SPI

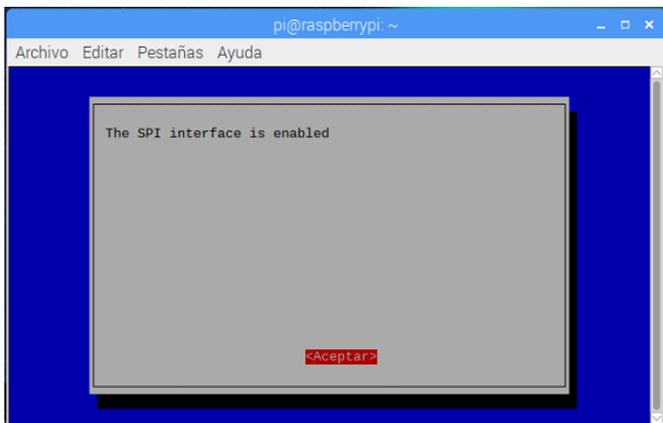


Figura 15: Interfaz SPI Habilitada

- I²C

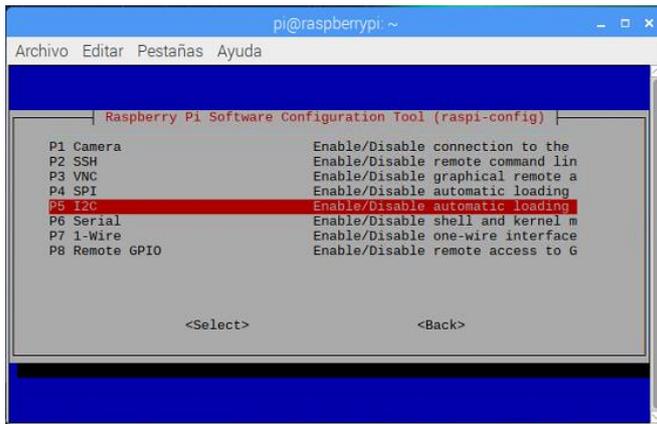


Figura 16: 15 I²C

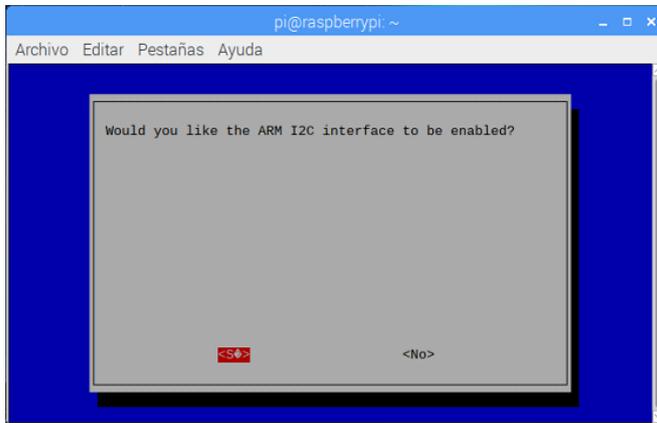


Figura 17: Habilitando Interfaz I²C

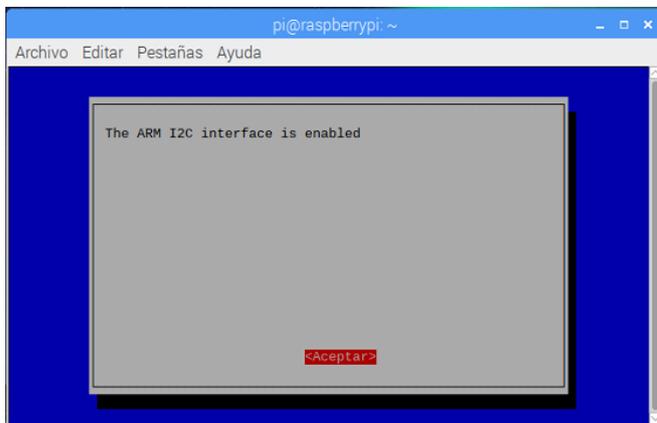


Figura 18: Interfaz I²C Habilitada

Una vez terminada las configuraciones seleccionar “Finish”.

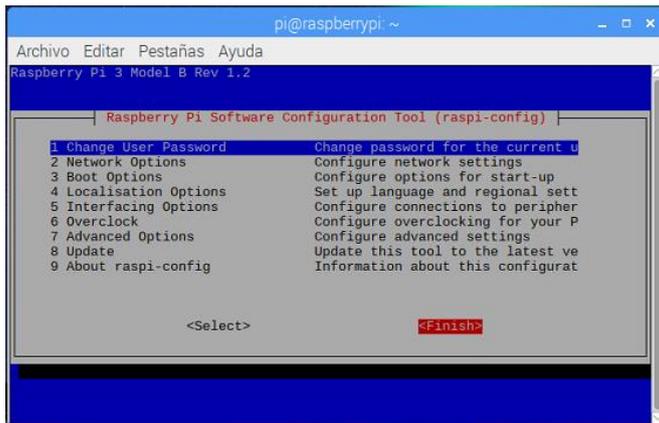


Figura 19: Configuración Raspbian Terminado

Se ingresa el comando “sudo reboot” para reiniciar el sistema.

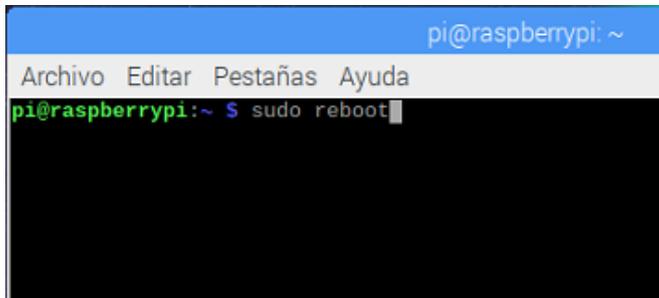
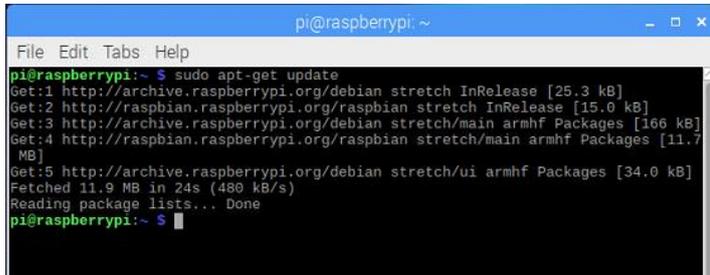


Figura 20: Reinicio del sistema

ANEXO C

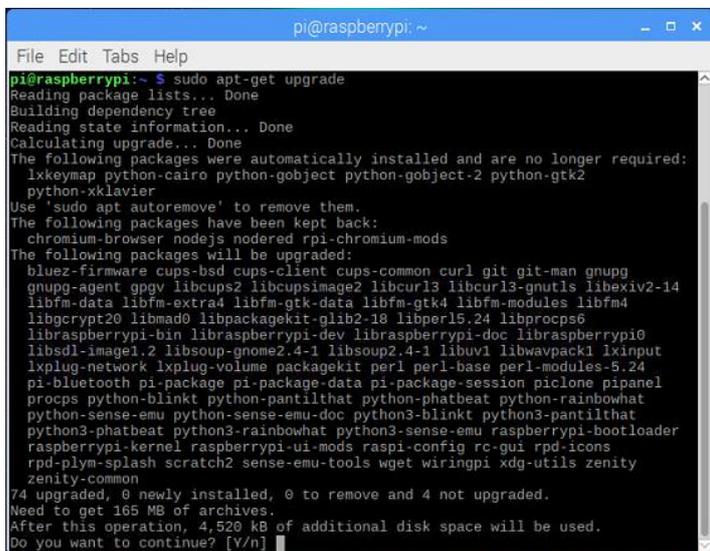
Instalación de Programas Necesarios

Antes de instalar los programas necesarios se deben buscar e instalar actualizaciones ingresando los comandos “sudo apt-get update” y “sudo apt-get upgrade” repectivamente:



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get update  
Get:1 http://archive.raspberrypi.org/debian stretch InRelease [25.3 kB]  
Get:2 http://raspbian.raspberrypi.org/raspbian stretch InRelease [15.0 kB]  
Get:3 http://archive.raspberrypi.org/debian stretch/main armhf Packages [166 kB]  
Get:4 http://raspbian.raspberrypi.org/raspbian stretch/main armhf Packages [11.7 MB]  
Get:5 http://archive.raspberrypi.org/debian stretch/ui armhf Packages [34.0 kB]  
Fetched 11.9 MB in 24s (480 kB/s)  
Reading package lists... Done  
pi@raspberrypi:~$
```

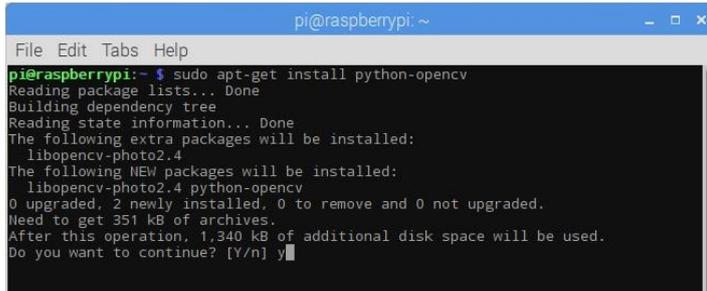
Figura 1: Ingreso “sudo apt-get update”



```
pi@raspberrypi:~$ sudo apt-get upgrade  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Calculating upgrade... Done  
The following packages were automatically installed and are no longer required:  
  lxkeymap python-cairo python-gobject python-gobject-2 python-gtk2  
  python-xklavier  
Use 'sudo apt autoremove' to remove them.  
The following packages have been kept back:  
  chromium-browser nodejs nodered rpi-chromium-mods  
The following packages will be upgraded:  
  bluez-firmware cups-bsd cups-client cups-common curl git git-man gnupg  
  gnupg-agent gpgv libcups2 libcupsimage2 libcurl3 libcurl3-gnutls libexiv2-14  
  libfm-data libfm-extra4 libfm-gtk-data libfm-gtk4 libfm-modules libfm4  
  libgcrpt20 libmad0 libpackagekit-glib2-18 libperl5.24 libprocps6  
  libraspberrypi-bin libraspberrypi-dev libraspberrypi-doc libraspberrypi0  
  libsdl-image1.2 libsoup-gnome2.4-1 libsoup2.4-1 libuv1 libwavpack1 lxinput  
  lxplug-network lxplug-volume packagekit perl perl-base perl-modules-5.24  
  pi-bluetooth pi-package pi-package-data pi-package-session piclone pipanel  
  procps python-blinkt python-pantilthat python-phatbeat python-rainbowhat  
  python-sense-emu python-sense-emu-doc python3-blinkt python3-pantilthat  
  python3-phatbeat python3-rainbowhat python3-sense-emu raspberrypi-bootloader  
  raspberrypi-kernel raspberrypi-ui-mods raspi-config rc-gui rpd-icons  
  rpd-plym-splash scratch2 sense-emu-tools wget wiringpi xdg-utils zenity  
  zenity-common  
74 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.  
Need to get 165 MB of archives.  
After this operation, 4,520 kB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Figura 2: Proceso de actualización

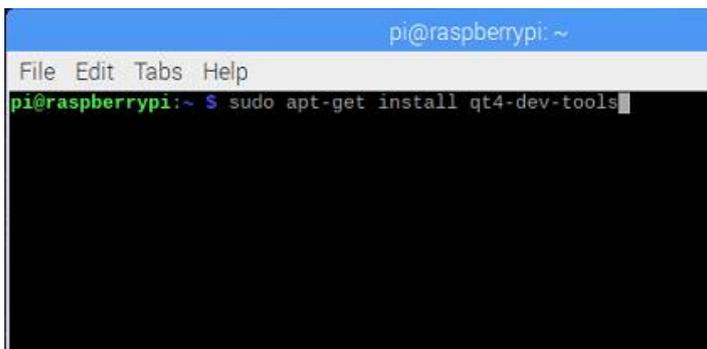
Para comprobar el funcionamiento de las cámaras se procede con la instalación de python-open cv con el comando “sudo apt-get install python-opencv”.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get install python-opencv  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following extra packages will be installed:  
  libopencv-photo2.4  
The following NEW packages will be installed:  
  libopencv-photo2.4 python-opencv  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 351 kB of archives.  
After this operation, 1,340 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

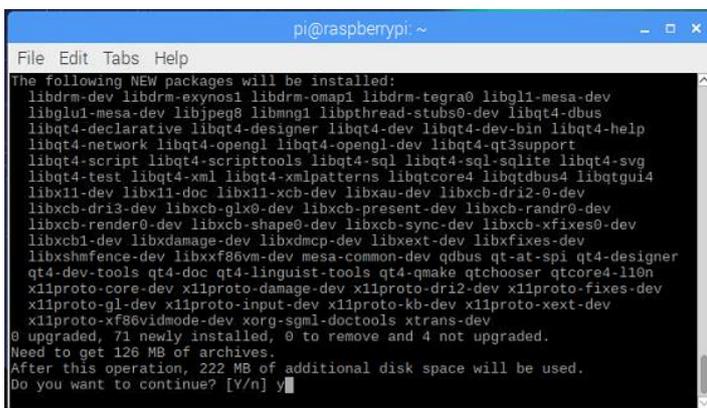
Figura 3: Instalando python-opencv

Para instalar el Entorno de Programación Qt se ingresan los comandos “sudo apt-get install qt4-dev-tools” y “sudo apt-get install qtcreator” respectivamente:



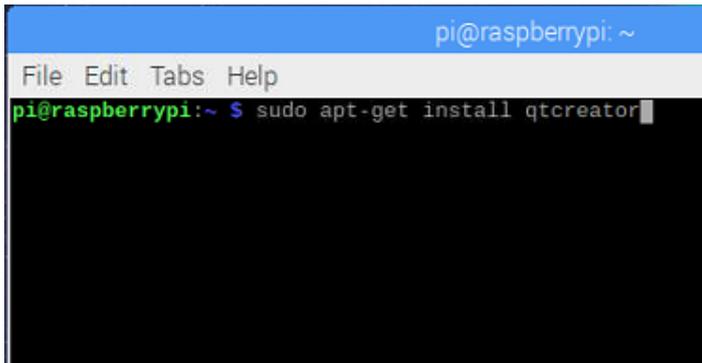
```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get install qt4-dev-tools
```

Figura 4: Instalando qt4-dev-tools



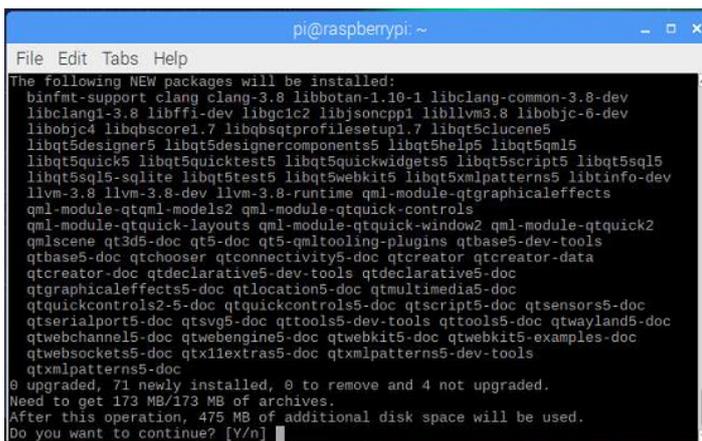
```
pi@raspberrypi: ~  
File Edit Tabs Help  
The following NEW packages will be installed:  
  libdrm-dev libdrm-exynos1 libdrm-omap1 libdrm-tegra0 libgl1-mesa-dev  
  libglu1-mesa-dev libjpeg8 libmng1 libpthread-stubs0-dev libqt4-dbus  
  libqt4-declarative libqt4-designer libqt4-dev libqt4-dev-bin libqt4-help  
  libqt4-network libqt4-opengl libqt4-opengl-dev libqt4-qt3support  
  libqt4-script libqt4-scripttools libqt4-sql libqt4-sql-sqlite libqt4-svg  
  libqt4-test libqt4-xml libqt4-xmlpatterns libqtcore4 libqtdbus4 libqtgui4  
  libx11-dev libx11-doc libx11-xcb-dev libxau-dev libxcb-dri2-0-dev  
  libxcb-dri3-dev libxcb-glx0-dev libxcb-present-dev libxcb-randr0-dev  
  libxcb-render0-dev libxcb-shape0-dev libxcb-sync-dev libxcb-xfixes0-dev  
  libxcb1-dev libxdamage-dev libxdmcp-dev libxext-dev libxfixes-dev  
  libxshmfence-dev libxxf86vm-dev mesa-common-dev qdbus qt-at-spi qt4-designer  
  qt4-dev-tools qt4-doc qt4-linguist-tools qt4-qmake qtchooser qtcore4-l10n  
  x11proto-core-dev x11proto-damage-dev x11proto-dri2-dev x11proto-fixes-dev  
  x11proto-gl-dev x11proto-input-dev x11proto-kb-dev x11proto-xext-dev  
  x11proto-xf86vidmode-dev xorg-sgml-doctools xtrans-dev  
0 upgraded, 71 newly installed, 0 to remove and 4 not upgraded.  
Need to get 126 MB of archives.  
After this operation, 222 MB of additional disk space will be used.  
Do you want to continue? [Y/n] y
```

Figura 5: Proceso de instalación qt4-dev-tools



```
pi@raspberrypi: ~  
File Edit Tabs Help  
pi@raspberrypi:~$ sudo apt-get install qtcreator
```

Figura 6: Instalando Qt Creator



```
pi@raspberrypi: ~  
File Edit Tabs Help  
The following NEW packages will be installed:  
binfmt-support clang clang-3.8 libbotan-1.10-1 libclang-common-3.8-dev  
libclang1-3.8 libffi-dev libgc1c2 libjsoncpp1 libllvm3.8 libobjc-6-dev  
libobjc4 libqbscore1.7 libqbsqtprofilessetup1.7 libqt5clucene5  
libqt5designer5 libqt5designercomponents5 libqt5help5 libqt5qml5  
libqt5quick5 libqt5quicktest5 libqt5quickwidgets5 libqt5script5 libqt5sql5  
libqt5sql5-sqlite libqt5test5 libqt5webkit5 libqt5xmlpatterns5 libtinfo-dev  
llvm-3.8 llvm-3.8-dev llvm-3.8-runtime qml-module-qtgraphicaleffects  
qml-module-qtqml-models2 qml-module-qtquick-controls  
qml-module-qtquick-layouts qml-module-qtquick-window2 qml-module-qtquick2  
qmlscene qt3d5-doc qt5-doc qt5-qmltooling-plugins qtbase5-dev-tools  
qtbase5-doc qtchooser qtconnectivity5-doc qtcreator qtcreator-data  
qtcreator-doc qtdeclarative5-dev-tools qtdeclarative5-doc  
qtgraphicaleffects5-doc qtlocation5-doc qtmultimedia5-doc  
qtquickcontrols2-5-doc qtquickcontrols5-doc qtscript5-doc qtsensors5-doc  
qtserialport5-doc qtsvg5-doc qttools5-dev-tools qttools5-doc qtwayland5-doc  
qtwebchannel5-doc qtwebengine5-doc qtwebkit5-doc qtwebkit5-examples-doc  
qtwebsockets5-doc qtx11extras5-doc qtxmlpatterns5-dev-tools  
qtxmlpatterns5-doc  
0 upgraded, 71 newly installed, 0 to remove and 4 not upgraded.  
Need to get 173 MB/173 MB of archives.  
After this operation, 475 MB of additional disk space will be used.  
Do you want to continue? [Y/n]
```

Figura 7: Proceso de instalación de Qt Creator

Después de Terminar con la instalación se reinicia el sistema.

ANEXO D

Instalación de Librerías

La librería para controlar la cámara Flir Lepton se puede obtener del siguiente enlace:

<https://lepton.flir.com/sdk/>

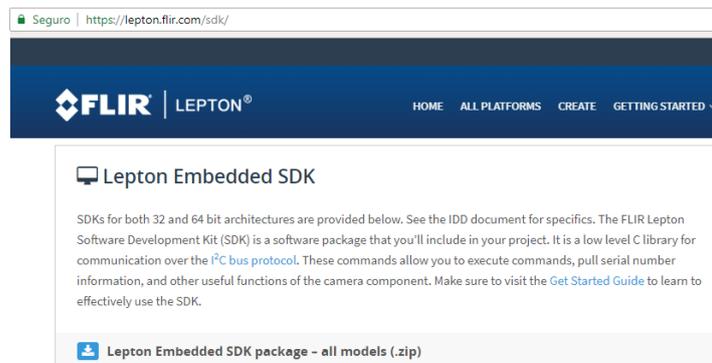


Figura 1: Descarga de Librería leptónSDKEmb32PUB

Los archivos se descargan y colocan en una carpeta con el nombre “leptónSDKEmb32PUB”, luego se accede a la carpeta desde la Terminal y se compila con el comando “make” como se muestra a continuación:

```
pi@raspberrypi: ~/leptonSDKEmb32PUB
File Edit Tabs Help
pi@raspberrypi:~ $ cd /home/pi/leptonSDKEmb32PUB
pi@raspberrypi:~/leptonSDKEmb32PUB $ make
mkdir -p "Debug"
gcc -fpermissive -Dlinux=1 -c -v -g -o "Debug/raspi_I2C.o" -I/cydrive/c/WinAVR-20090313/avr/include "raspi_I2C.c"
Using built-in specs.
COLLECT_GCC=gcc
Target: arm-linux-gnueabi
Configured with: ../src/configure -v --with-pkgversion='Raspbian 6.3.0-18-rp1+d
eb9u1' --with-bugurl=file:///usr/share/doc/gcc-6/README.Bugs --enable-languages=
c,ada,c++,java,go,d,fortran,objc,obj-c++ --prefix=/usr --program-suffix=-6 --pro
gram-prefix=arm-linux-gnueabi --enable-shared --enable-linker-build-id --libe
xecdir=/usr/lib --without-included-gettext --enable-threads=posix --libdir=/usr/
lib --enable-nls --with-sysroot=/ --enable-clocale=gnu --enable-libstdcxx-debug
--enable-libstdcxx-timeyes --with-default-libstdcxx-abi=new --enable-gnu-unique
-object --disable-libitm --disable-libquadmath --enable-plugin --with-system-zli
b --disable-browser-plugin --enable-java-awt=gtk --enable-gtk-cairo --with-java-
home=/usr/lib/jvm/java-1.5.0-gcj-6-armhf/jre --enable-java-home --with-jvm-root-
dir=/usr/lib/jvm/java-1.5.0-gcj-6-armhf --with-jvm-jar-dir=/usr/lib/jvm-exports/
java-1.5.0-gcj-6-armhf --with-arch-directory=arm --with-ecj-jar=/usr/share/java/
eclipse-ecj.jar --with-target-system-zlib --enable-objc-gc=auto --enable-multiar
ch --disable-sjlj-exceptions --with-arch=armv6 --with-fpu=vfp --with-float=hard
--enable-checking=release --build=arm-linux-gnueabi --host=arm-linux-gnueabi --
target=arm-linux-gnueabi
```

Figura 2: Acceso y compilación de librería leptónSDKEmb32PUB

Se debe descargar la librería wiringPi para poder configurar los puertos GPIO con el comando: “git clone git://git.drogon.net/wiringPi”.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ git clone git://git.drogon.net/wiringPi
Cloning into 'wiringPi'...
remote: Counting objects: 1177, done.
remote: Compressing objects: 100% (980/980), done.
remote: Total 1177 (delta 820), reused 213 (delta 142)
Receiving objects: 100% (1177/1177), 370.25 KiB | 283.00 KiB/s, done.
Resolving deltas: 100% (820/820), done.
pi@raspberrypi:~$

```

Figura 3: Instalando librería wiringPi

Los archivos se descargan y colocan en una carpeta con el nombre “wiringPi”, luego se accede a la carpeta desde la Terminal y se compila con el comando “./build” como se muestra a continuación:

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ cd wiringPi
pi@raspberrypi:~/wiringPi$ ./build
wiringPi Build script
=====

WiringPi Library
[UnInstall]
[Compile] wiringPi.c
[Compile] wiringSerial.c
[Compile] wiringShift.c

```

Figura 4: Compilando librería wiringPi

Para comprobar la instalación de la librería se ingresa el comando “gpio readall” y se observan las configuraciones de entrada y salida de los pines de la Raspberry Pi.

```

pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~$ gpio readall
-----Pi 3-----
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 2 | 8 | SDA.1 | ALTO | 1 | 3 | 4 | | | 5v | | |
| 3 | 9 | SCL.1 | ALTO | 1 | 5 | 6 | | | 0v | | |
| 4 | 7 | GPIO.7 | IN | 1 | 7 | 8 | 0 | IN | TXD | 15 | 14 |
| | | | | | 9 | 10 | 1 | IN | RXD | 16 | 15 |
| 17 | 0 | GPIO.0 | IN | 0 | 11 | 12 | 0 | IN | GPIO.1 | 1 | 18 |
| 27 | 2 | GPIO.2 | IN | 0 | 13 | 14 | | | 0v | | |
| 22 | 3 | GPIO.3 | IN | 0 | 15 | 16 | 0 | IN | GPIO.4 | 4 | 23 |
| | | | | | 17 | 18 | 0 | IN | GPIO.5 | 5 | 24 |
| 10 | 12 | MOSI | ALTO | 0 | 19 | 20 | | | 0v | | |
| 9 | 13 | MISO | ALTO | 0 | 21 | 22 | 0 | IN | GPIO.6 | 6 | 25 |
| 11 | 14 | SCLK | ALTO | 0 | 23 | 24 | 1 | OUT | CE0 | 10 | 8 |
| | | | | | 25 | 26 | 1 | OUT | CE1 | 11 | 7 |
| 0 | 30 | SDA.0 | IN | 1 | 27 | 28 | 1 | IN | SCL.0 | 31 | 1 |
| 5 | 21 | GPIO.21 | IN | 1 | 29 | 30 | | | 0v | | |
| 6 | 22 | GPIO.22 | IN | 1 | 31 | 32 | 0 | IN | GPIO.26 | 26 | 12 |
| 13 | 23 | GPIO.23 | IN | 0 | 33 | 34 | | | 0v | | |
| 19 | 24 | GPIO.24 | IN | 0 | 35 | 36 | 0 | IN | GPIO.27 | 27 | 16 |
| 26 | 25 | GPIO.25 | IN | 0 | 37 | 38 | 0 | IN | GPIO.28 | 28 | 20 |
| | | | | | 39 | 40 | 0 | IN | GPIO.29 | 29 | 21 |
|-----|-----|-----|-----|---|-----|---|-----|-----|-----|-----|
| BCM | wPi | Name | Mode | V | Physical | V | Mode | Name | wPi | BCM |
-----Pi 3-----
pi@raspberrypi:~$

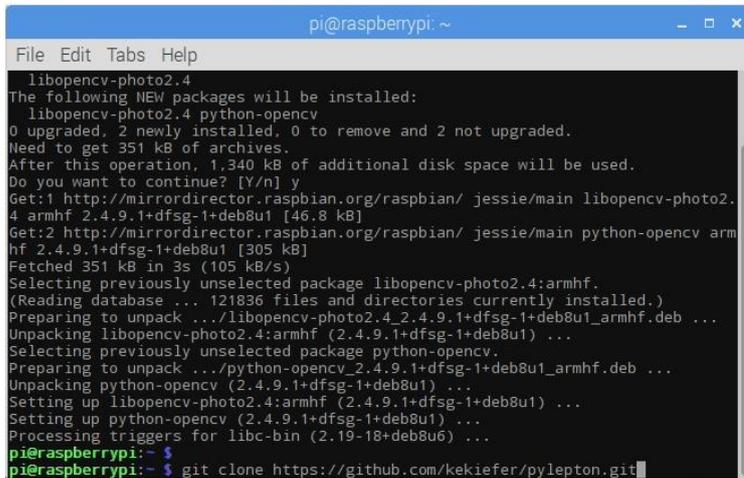
```

Figura 5: Prueba librería wiringPi

ANEXO E

Prueba de Funcionamiento de Cámaras

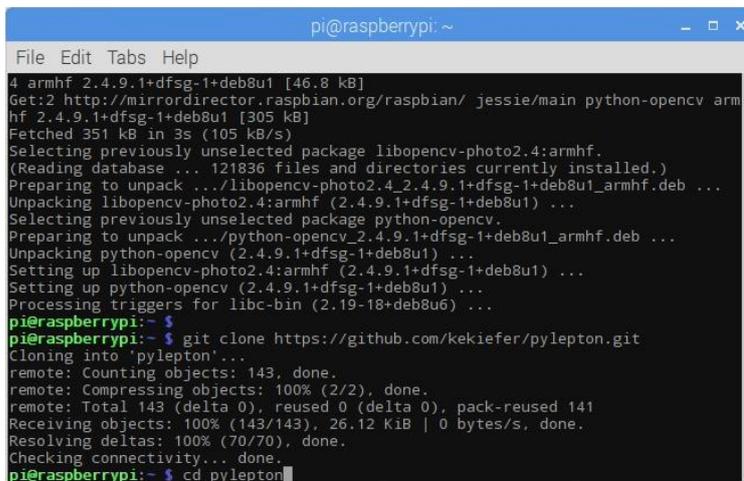
Para probar el funcionamiento de las cámaras Flir Lepton y Board 2V se descarga un archivo con el comando: “git clone https://github.com/kekiefer/pylepton.git”.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
libopencv-photo2.4  
The following NEW packages will be installed:  
  libopencv-photo2.4 python-opencv  
0 upgraded, 2 newly installed, 0 to remove and 2 not upgraded.  
Need to get 351 kB of archives.  
After this operation, 1,340 kB of additional disk space will be used.  
Do you want to continue? [Y/n] y  
Get:1 http://mirrordirector.raspbian.org/raspbian/ jessie/main libopencv-photo2.4 armhf 2.4.9.1+dfsg-1+deb8u1 [46.8 kB]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main python-opencv armhf 2.4.9.1+dfsg-1+deb8u1 [305 kB]  
Fetched 351 kB in 3s (105 kB/s)  
Selecting previously unselected package libopencv-photo2.4:armhf.  
(Reading database ... 121836 files and directories currently installed.)  
Preparing to unpack .../libopencv-photo2.4_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...  
Unpacking libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...  
Selecting previously unselected package python-opencv.  
Preparing to unpack .../python-opencv_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...  
Unpacking python-opencv (2.4.9.1+dfsg-1+deb8u1) ...  
Setting up libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...  
Setting up python-opencv (2.4.9.1+dfsg-1+deb8u1) ...  
Processing triggers for libc-bin (2.19-18+deb8u6) ...  
pi@raspberrypi:~$  
pi@raspberrypi:~$ git clone https://github.com/kekiefer/pylepton.git
```

Figura 1: Descarga del archivo de prueba

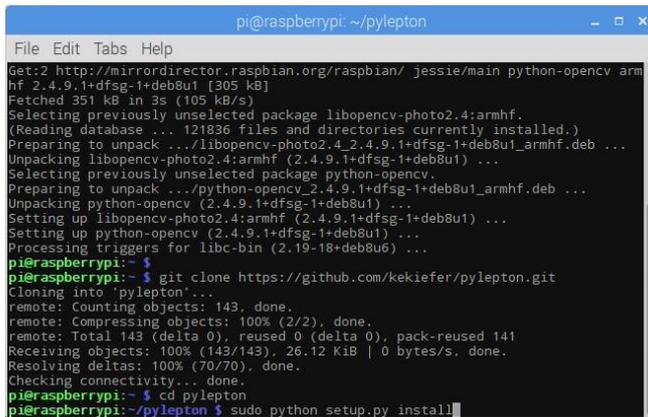
Luego cd a la carpeta de pylepton recién descargada: “cd pylepton”.



```
pi@raspberrypi: ~  
File Edit Tabs Help  
4 armhf 2.4.9.1+dfsg-1+deb8u1 [46.8 kB]  
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main python-opencv armhf 2.4.9.1+dfsg-1+deb8u1 [305 kB]  
Fetched 351 kB in 3s (105 kB/s)  
Selecting previously unselected package libopencv-photo2.4:armhf.  
(Reading database ... 121836 files and directories currently installed.)  
Preparing to unpack .../libopencv-photo2.4_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...  
Unpacking libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...  
Selecting previously unselected package python-opencv.  
Preparing to unpack .../python-opencv_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...  
Unpacking python-opencv (2.4.9.1+dfsg-1+deb8u1) ...  
Setting up libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...  
Setting up python-opencv (2.4.9.1+dfsg-1+deb8u1) ...  
Processing triggers for libc-bin (2.19-18+deb8u6) ...  
pi@raspberrypi:~$  
pi@raspberrypi:~$ git clone https://github.com/kekiefer/pylepton.git  
Cloning into 'pylepton'...  
remote: Counting objects: 143, done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 143 (delta 0), reused 0 (delta 0), pack-reused 141  
Receiving objects: 100% (143/143), 26.12 KiB | 0 bytes/s, done.  
Resolving deltas: 100% (70/70), done.  
Checking connectivity... done.  
pi@raspberrypi:~$ cd pylepton
```

Figura 2: acceso a la carpeta del archivo

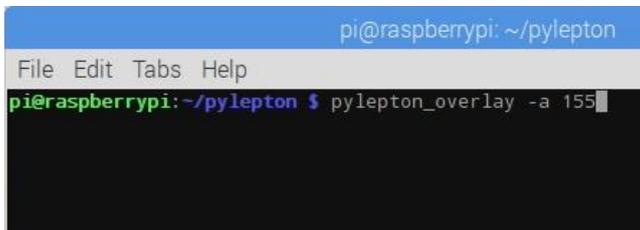
Ahora se instala el software pylepton: “sudo python setup.py install”.



```
pi@raspberrypi: ~/pylepton
File Edit Tabs Help
Get:2 http://mirrordirector.raspbian.org/raspbian/ jessie/main python-opencv armhf 2.4.9.1+dfsg-1+deb8u1 [305 kB]
Fetched 351 kB in 3s (105 kB/s)
Selecting previously unselected package libopencv-photo2.4:armhf.
(Reading database ... 121836 files and directories currently installed.)
Preparing to unpack .../libopencv-photo2.4_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...
Unpacking libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...
Selecting previously unselected package python-opencv.
Preparing to unpack .../python-opencv_2.4.9.1+dfsg-1+deb8u1_armhf.deb ...
Unpacking python-opencv (2.4.9.1+dfsg-1+deb8u1) ...
Setting up libopencv-photo2.4:armhf (2.4.9.1+dfsg-1+deb8u1) ...
Setting up python-opencv (2.4.9.1+dfsg-1+deb8u1) ...
Processing triggers for libc-bin (2.19-18+deb8u6) ...
pi@raspberrypi:~$
pi@raspberrypi:~$ git clone https://github.com/kekiefier/pylepton.git
Cloning into 'pylepton'...
remote: Counting objects: 143, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 143 (delta 0), reused 0 (delta 0), pack-reused 141
Receiving objects: 100% (143/143), 26.12 KiB | 0 bytes/s, done.
Resolving deltas: 100% (70/70), done.
Checking connectivity... done.
pi@raspberrypi:~$ cd pylepton
pi@raspberrypi:~/pylepton$ sudo python setup.py install
```

Figura 3: Instalación programa setup.py

Para probar el programa se escribe lo siguiente: “pylepton_overlay -a 155”.



```
pi@raspberrypi: ~/pylepton
File Edit Tabs Help
pi@raspberrypi:~/pylepton$ pylepton_overlay -a 155
```

Figura 4: Prueba código pylepton_overlay

Se ve una imagen de superposición térmica en la pantalla y para salir del programa se presiona “ctrl + c”.



Figura 5: Funcionamiento de las cámaras

ANEXO F

Configuración de Qt Creator

Al ejecutar el programa Qt Creator en la barra de herramientas se elige “Tools” en la opción “Options...”.

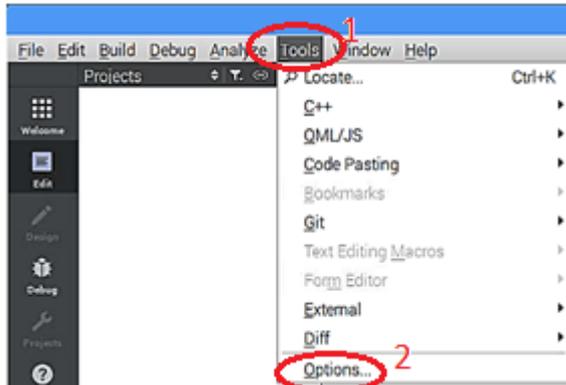


Figura 1: Configurando Qt Creator

En la opción “Build & Run” se selecciona la pestaña “Compilers” donde se crea el compilador “GCC” desde la ruta “/usr/bin/gcc”.

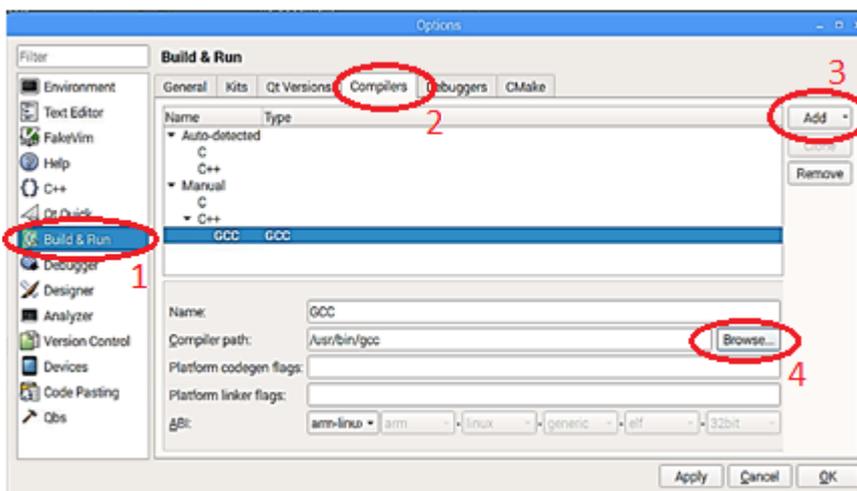


Figura 2: Agregando compilador

En la opción “Build & Run” se selecciona “Kits” en donde agrega el compilador y la versión de Qt.

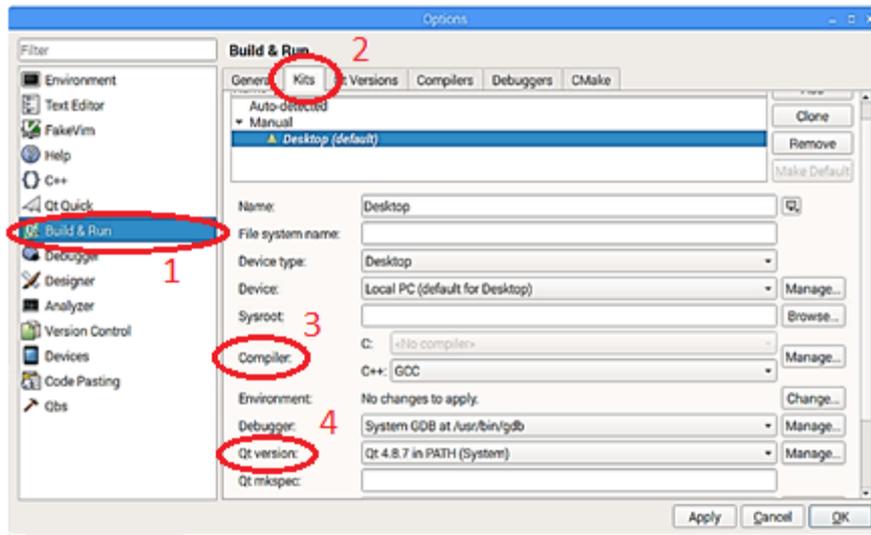


Figura 3: Agregando compilador y versión del sistema

ANEXO G

Nuevo Proyecto en Qt Creator

Al ejecutar el programa Qt Creator en la barra de herramientas se elige “File” en la opción “New File or Project...”.

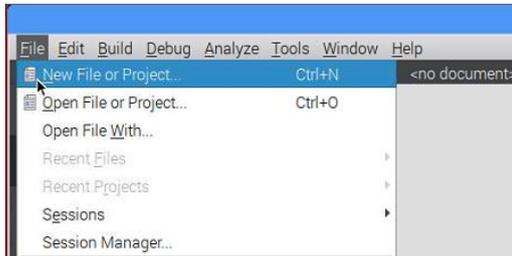


Figura 1: Creando Nuevo Proyecto

En el tipo de proyecto se selecciona “Application”, “Qt Widgets Application” y luego “next”.

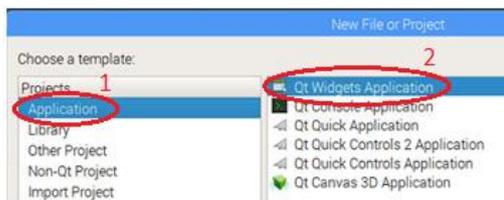


Figura 2: Selección tipo de proyecto

Se ingresa el nombre del proyecto y su ubicación, después se presiona “Next>” y en la siguiente ventana “Next>”.

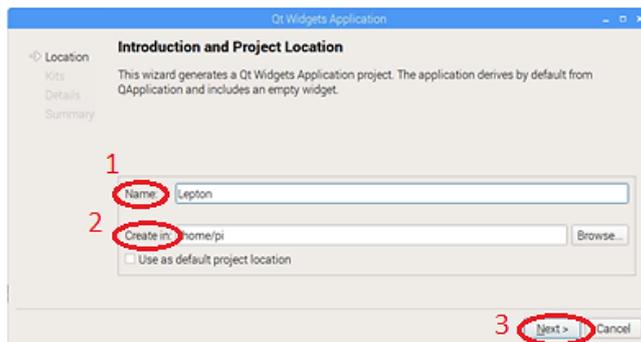


Figura 3: Selección Nombre y ubicación del proyecto

Se cambian los nombres por defecto de los archivos y luego se presiona “Next”.

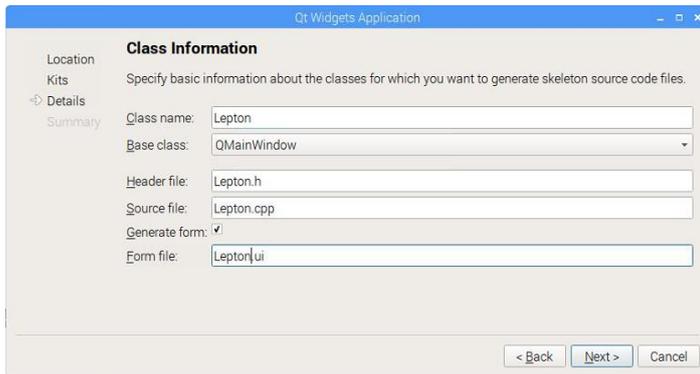


Figura 4: Selección nombre y ubicación del proyecto

Por últimos se muestran el nombre de los archivos y la ubicación del proyecto si la información es correcta se presiona “Finish” y el proyecto está listo para diseñar el código en los archivos fuente y cabecera.

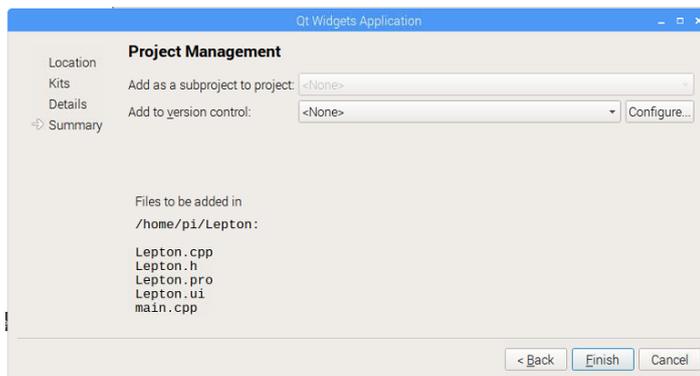


Figura 5: Creación del proyecto finalizado

ANEXO H

Valores almacenados para hallar el Máximo y Mínimo estableciendo los límites de detección en los sensores infrarrojos.

30491	30499	30507	30513	30499	30493	30505	30493
29391	29386	29371	29397	29380	29378	29391	29373
30491	30499	30507	30513	30499	30493	30505	30495
29391	29386	29371	29397	29380	29378	29395	29373
30491	30499	30507	30513	30499	30499	30505	30493
29393	29386	29371	29397	29380	29386	29395	29371
30493	30497	30507	30513	30499	30499	30505	30493
29391	29384	29384	29404	29382	29386	29395	29371
30493	30497	30507	30513	30499	30497	30501	30493
29391	29384	29384	29404	29382	29386	29391	29371
30493	30497	30507	30513	30499	30497	30501	30495
29391	29384	29384	29404	29382	29386	29391	29369
30511	30497	30507	30509	30497	30497	30499	30495
29389	29382	29389	29404	29380	29386	29389	29369
30511	30497	30507	30509	30497	30495	30499	30495
29389	29382	29389	29404	29380	29371	29389	29369
30511	30497	30507	30509	30497	30495	30497	30493
29389	29382	29389	29404	29380	29371	29384	29369
30509	30497	30509	30509	30495	30495	30497	30493
29382	29384	29393	29399	29378	29371	29384	29369
30509	30497	30509	30509	30495	30491	30497	30493
29382	29384	29393	29399	29378	29384	29384	29369
30509	30497	30509	30509	30495	30503	30497	30495
29382	29384	29393	29399	29378	29391	29380	29369
30505	30495	30507	30507	30495	30503	30497	30495
29389	29380	29393	29399	29378	29395	29380	29369
30505	30495	30507	30507	30495	30503	30497	30495
29389	29380	29393	29399	29380	29395	29380	29369
30505	30495	30507	30505	30495	30503	30497	30523
29389	29380	29393	29399	29380	29395	29378	29369
30503	30493	30505	30505	30495	30503	30497	30523
29384	29375	29386	29399	29380	29391	29378	29369
30501	30495	30505	30501	30495	30507	30497	30523
29384	29375	29386	29397	29382	29391	29378	29369
30501	30495	30505	30501	30495	30507	30495	30519
29384	29375	29391	29397	29382	29391	29380	29373
30501	30495	30505	30501	30495	30507	30495	30519
29384	29375	29391	29397	29382	29391	29380	29373
30499	30507	30507	30501	30495	30505	30495	30519
29384	29373	29397	29375	29386	29391	29380	29373
30499	30507	30507	30501	30495	30505	30493	30493
29384	29373	29397	29375	29386	29391	29373	29371
30499	30507	30507	30501	30495	30505	30493	30493
29384	29373	29397	29375	29386	29391	29373	29371

30493	26194	30505	26156	30248	29356	30819	29378
29371	30301	26072	30513	26194	30763	29347	45464
30529	26121	30505	26159	30248	29356	31002	29378
29373	30301	26072	30404	26194	30759	29360	45464
30531	26121	30536	26165	30248	29347	31002	29378
29375	30301	26082	30404	26194	30779	29360	45464
30531	26121	30536	26165	30485	29349	31002	29378
29375	30141	26082	30404	26213	30779	29360	45464
30531	26127	30536	26165	30485	29349	31954	29393
29375	30141	26082	30281	26213	30779	29439	45464
30525	26127	30560	26165	30485	29349	31954	29393
29371	30141	26085	30281	26213	30788	29439	45464
30525	26127	30560	26165	30707	29349	31954	29402
29371	30076	26085	30281	26407	30788	29439	45464
30525	26146	30560	26165	30707	29349	35071	29402
29371	30076	26085	30287	26407	30788	29397	45464
30542	26146	30581	26188	30707	29349	35071	29391
29362	30076	26117	30287	26407	30788	29397	45464
30542	26146	30581	26300	30734	29343	35071	29391
29362	30145	26117	30287	29319	30788	29397	45464
30542	26146	30581	26188	30727	29343	36931	29397
29362	30143	26117	30287	29329	30788	29369	45464
30544	26159	30566	26188	30727	29343	36931	29397
29364	30143	26121	30287	29329	30794	29369	45464
30544	26159	30570	26188	30727	29345	36931	29397
29364	30143	26143	30287	29329	30794	29369	45464
30544	26159	30570	26188	30733	29345	45464	29397
29364	30096	26143	30283	29343	30794	29384	45464
30544	26159	30570	26191	30733	29345	45464	29397
29362	30096	26143	30283	29343	30798	29384	45464
30544	26159	30570	26191	30733	29351	45464	29380
29362	30096	26162	30283	29343	30798	29389	45464
30544	26159	30570	26191	30731	29351	45464	29380
29362	29371	26162	30264	29338	30798	29389	45464
30517	26146	30570	26188	30731	29351	45464	29373
29360	29371	26162	30264	29338	30798	29384	45464
30517	26146	30566	26188	30731	29356	45464	29373
29360	29371	26168	30256	29338	30798	29384	45464
30517	26146	30566	26191	30752	29356	45464	29373
29360	29624	26168	30256	29345	30798	29384	45464
30552	26076	30566	26191	30752	29356	45464	29373
29364	29624	26168	30256	29345	30811	29384	45464
30552	26076	30564	26191	30752	29354	45464	29373
29364	29624	26156	30256	29345	30820	29375	45464
30552	26076	30564	26191	30756	29351	45464	29364
29364	30450	26156	30256	29349	30820	29375	45464
30548	26072	30564	26191	30756	29351	45464	29364
28391	30450	26156	30266	29349	30820	29375	45464
30501	26072	30554	26181	30756	29351	45464	29364
26194	30450	26156	30266	29349	30819	29378	45464
30501	26072	30554	26181	30763	29347	45464	29367
26194	30505	26156	30266	29356	30819	29378	45464
30501	26072	30554	26181	30763	29347	45464	29367

45464	29382	30851	29351	30824	29364	30649	29369
29367	45464	29354	30813	29354	30834	29356	30855
45464	29382	30851	29351	30805	29347	30649	29364
29362	45464	29354	30813	29349	30834	29356	30855
45464	29382	30801	29351	30805	29347	30649	29364
29362	45464	29354	30828	29349	30834	29356	30855
45464	29389	30855	29349	30805	29347	30715	29378
29362	45464	29356	30828	29349	30744	29362	30855
45464	29389	30855	29349	30748	29358	30715	29378
29362	45464	29356	30828	29354	30744	29362	30887
45464	29389	30855	29349	30748	29358	30715	29386
29364	40558	29356	30857	29354	30744	29362	30887
45464	29354	30826	29349	30748	29358	30838	29386
29364	30847	29373	30857	29354	30655	29362	30874
45464	29354	30826	29349	30597	29358	30838	29443
29364	30847	29373	30857	29354	30655	29362	30874
45464	29354	30826	29349	30597	29358	30838	29443
29371	30847	29373	30758	29354	30655	29362	30857
45464	29354	30807	29349	30597	29358	30874	29397
29371	30847	29351	30758	29354	30647	29354	30857
45464	29354	30807	29349	30581	29356	30843	29397
29371	30847	29351	30758	29364	30647	29356	30857
45464	29354	30807	29349	30587	29356	30843	29397
29373	30847	29351	30665	29364	30647	29356	30895
45464	29354	30824	29349	30587	29356	30843	29406
29373	30843	29351	30665	29364	30661	29356	30878
45464	29351	30824	29349	30587	29354	30830	29386
29373	30843	29351	30665	29364	30661	29364	30878
45464	29351	30824	29349	30744	29354	30836	29386
29373	30843	29351	30824	29364	30661	29369	
45464	29351	30876	29354	30744	29354	30836	
29382	30851	29349	30824	29364	30663	29369	
45464	29354	30813	29354	30744	29356	30836	

Valor Máximo Detectado	45464
Valor Mínimo Detectado	29347

2018

Manual de Usuario



Luis Maldonado Garcia

ESPOCH

10-7-2018

Contenido

Introducción	3
Precauciones	3
Hardware	4
Software	5
Lepton:	5
Camera.py:	6
Conexiones	6

Introducción

En el presente manual se detalla el hardware del Dispositivo que permite visualizar y almacenar la imagen térmica de un objetivo.

El Dispositivo consta de un procesador central conformado por una mini computadora Raspberry Pi 3B la cual tiene instalado un Sistema Operativo Raspbian con el entorno de programación Qt, Open Source y Python localizados en la memoria Micro SD, la cámara radiométrica Flir Lepton 2.5, ventiladores y el módulo buzzer están conectados en los pines de la Raspberry Pi. La cámara Board 2V y la pantalla de 7 pulgadas están conectadas en los puertos DSI respectivamente.

Precauciones

- Se deben tener medidas de seguridad al momento de utilizar el dispositivo ya que la cámara radiométrica Flir Lepton es susceptible a descargas electroestáticas.
- En caso de eliminar los archivos por accidente se encuentra una copia en la ubicación “home/pi”.
- En el archivo “camera.py” se puede modificar la ruta de almacenamiento de las imágenes mientras que las imágenes térmicas se almacenaran en la ubicación donde se ejecuta el archivo “Lepton”.
- Considere el ambiente de trabajo para el uso de las cámaras ya que los límites no son las mismas, los rangos se muestran en la tabla 1:

Cámara	Rango de Temperatura
Board 2V	-20 °C a 60 °C
Flir Lepton	-10 °C a 80 °C

Tabla 1: Rango de Temperatura de Funcionamiento

Hardware



Figura 1: Vista Lateral del Dispositivo



Figura 2: Vista Posterior del Dispositivo

Software

La GUI (Interfaz Gráfica de Usuario) del dispositivo se puede ejecutar un archivo con el nombre “Lepton” desde el escritorio. Si desea utilizar la cámara Board 2V puede ejecutar un acceso directo con el nombre “camera.py” el cual se ejecuta con el programa “Thony (simple mode)”

Lepton:

Al ejecutar el archivo “Lepton” aparecerá una ventana de opciones, seleccione la opción “Execute” como se muestra en la figura 2:

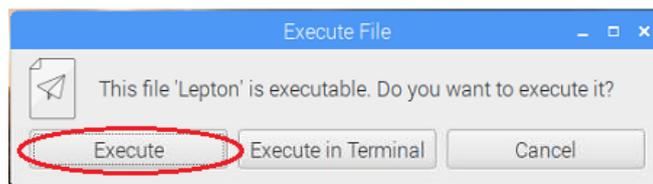


Figura 3: Opciones de ejecución

Después se abrirá la GUI cuyos componentes se detallan a continuación:

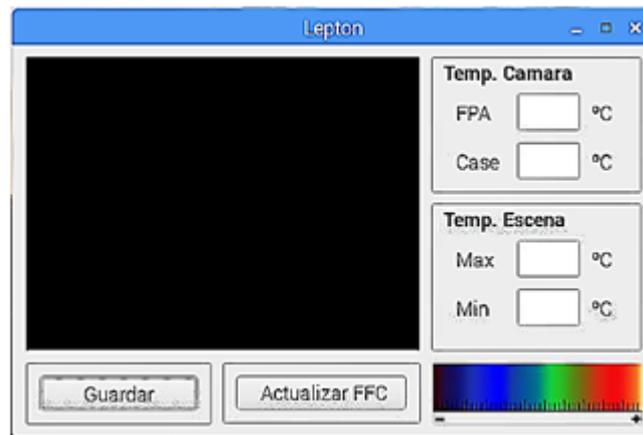


Figura 4: GUI Lepton

- **FPA:** Muestra la temperatura de la FPA (Matriz de Campo Focal).
- **Case:** Muestra la temperatura de la cámara.
- **Max:** Muestra la temperatura máxima detectada en la escena.
- **Min:** Muestra la temperatura mínima detectada en la escena.
- **Guardar:** Almacena la imagen térmica del objetivo en la ubicación de archivo ejecutable.
- **Actualizar:** FFC: Corregir el campo plano y actualiza el campo de visión.

Camera.py:

Este archivo posee un código que permite almacenar la imagen captada por la cámara Board 2V en la ruta especificada al momento de presionar el botón “Guardar” de la GUI Lepton.

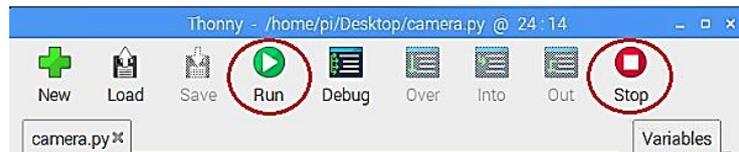


Figura 5: Archivo camera.py

- **Run:** Ejecuta el código del programa.
- **Stop:** Detiene el programa “camera.py”.

Una vez que se cierran los programas se deben copiar los archivos en otra ubicación ya que si se vuelven a ejecutar cualquiera de los programas los archivos almacenados se van a sobrescribir.

Conexiones

En caso de desconectarse los cables con los pines de la Raspberry Pi a continuación se muestra como conectar los componentes en sus respectivas tablas:

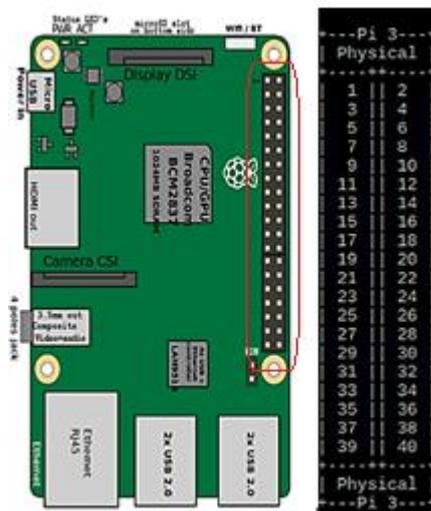


Figura 6: Distribución de los pines de la Raspberry Pi 3B+

Pin Cámara	Pin Raspberry	Nombre Pin
CS	24	CE0
MOSI	19	MOSI
MISO	21	MISO
CLK	23	CLK
GND	9	GND
VIN	1	3V3
SDA	3	SDA
SCL	5	SCL

Tabla 2: Conexión Cámara Flir Lepton a Raspberry

Pin M. Buzzer	Pin Raspberry	Nombre Pin
VCC	17	5V
GND	30	GND
SIGNAL	32	GPIO 12

Tabla 3: Conexión Módulo Buzzer a Raspberry Pi

Pin Buzzer	Pin Raspberry	Nombre Pin
SIGNAL	36	5V
GND	34	GND

Tabla 4: Conexión Buzzer a Raspberry Pi

Ventiladores	Pin Raspberry	Nombre Pin
VCC	4	5V
GND	6	GND

Tabla 5: Conexión Ventiladores a Raspberry Pi

Ventiladores	Pin Raspberry	Nombre Pin
VCC	2	5V
GND	14	GND

Tabla 6: Conexión Pantalla a Raspberry Pi