



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA**

ESCUELA DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES

**“Trabajo de grado previo a la obtención del Título de Ingeniero en
Electrónica y Telecomunicaciones”**

TRABAJO DE GRADUACIÓN

Título del proyecto:

**DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA GRAFICADOR 3D EN
MOVIMIENTO.**

Autores:

**Freddy Fernando Mora Hidalgo
Christian Humberto Pizán Cárdenas**

Director:

Ing. Fabián Gunsha

**Riobamba – Ecuador
Año 2015**

Los miembros del Tribunal de Graduación del proyecto de investigación de título: **DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA GRAFICADOR 3D EN MOVIMIENTO** presentado por: **Freddy Fernando Mora Hidalgo y Christian Humberto Pizanán Cárdenas** y dirigida por: **Ingeniero Fabián Gunsha**.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Ing. Carlos Peñafiel
Presidente de Tribunal



Firma

Ing. Fabián Gunsha
Director del Proyecto



Firma

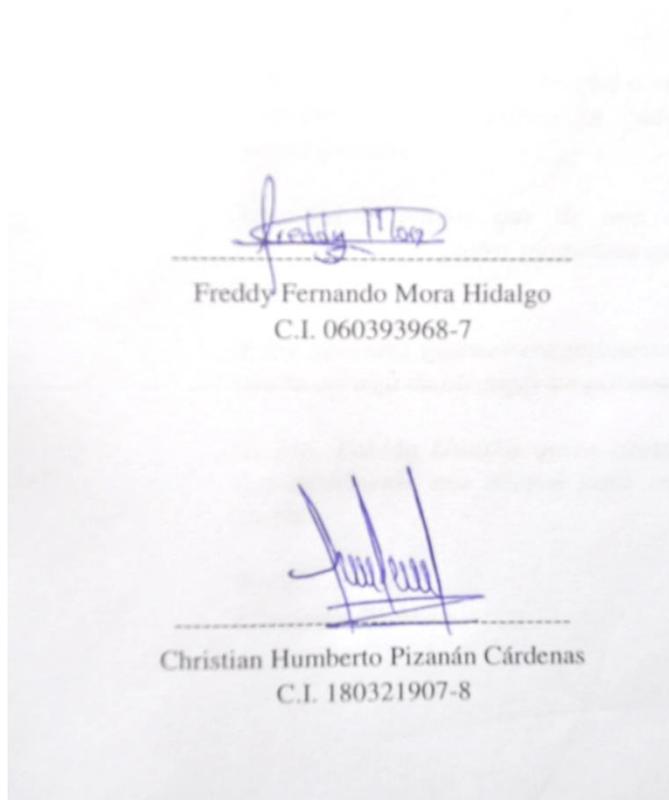
Ing. Aníbal Llanga
Miembro del Tribunal



Firma

AUTORÍA DE INVESTIGACIÓN

La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: **Freddy Mora, Christian Pizanán e Ingeniero Fabián Gunsha**; y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.



AGRADECIMIENTO

Este presente trabajo es la muestra de haber llegado al final de nuestra vida universitaria.

Agradecemos a Dios y a todas aquellas personas quienes hemos llegado a conocer en esta etapa tan importante de nuestras vidas.

A nuestras familias y en especial a nuestros padres y hermanos, por apoyarnos en cada decisión que hemos tomado.

A nuestros amigos que de una u otra manera correlacionaron gratos momentos que encaminaron nuestra vida.

A los docentes quienes encaminaron nuestras vidas con la certeza de alcanzar un porvenir alentador.

Al Ing. Fabián Gunsha quien asesoró esta tesis y constantemente nos motivó para realizar un buen trabajo.

Freddy Mora
Christian Pizanán

DEDICATORIA

A mis padres y hermana, quienes han sido el pilar fundamental en mi desarrollo integral y me han dado el ejemplo de superación y perseverancia para alcanzar una meta más en mi vida.

Freddy Mora

Quiero dedicar este trabajo a Dios por ayudarme en todos los momentos de mi vida, por darme la fuerza necesaria para salir adelante;

A mi familia que ha estado a mi lado en todo este tiempo brindándome su apoyo ilimitado.

Y a todas aquellas personas que siempre me han incentivado, a través de acciones y palabras alentadoras.

Christian Pizanán

ÍNDICE GENERAL

ÍNDICE DE TABLAS.....	xii
ÍNDICE DE ANEXOS.....	xiii
RESUMEN.....	xiv
SUMMARY	¡Error! Marcador no definido.
INTRODUCCIÓN.....	1
CAPÍTULO I.....	2
1. FUNDAMENTACIÓN TEÓRICA	2
1.1. Antecedentes	2
1.2. Sensores de proximidad.....	3
1.2.1. Láser.....	4
1.2.2. Propiedades de láser.....	5
1.2.3. Sensor láser LIDAR Lite v2.....	7
1.2.4. Características del láser del sensor LIDAR lite v2	8
1.2.5. Dimensiones sensor láser LIDAR lite v2.....	8
1.2.6. Descripción pines del sensor LIDAR lite v2.....	9
1.2.7. Cálculo de distancia del sensor LIDAR lite v2.....	10
1.3. Servomotores	11
1.3.1. Funcionamiento de un servomotor.....	11
1.3.2. Control de un servomotor	12
1.3.3. Servo estándar Hitec HS-311	14
1.4. Controlador de servomotores.....	15
1.4.1. Controlador Micro Maestro de 6 canales	16
1.4.1.1. Distribución de pines del controlador de 6 servos	16
1.5. Microcontroladores.....	17
1.5.1. Arquitectura de un microcontrolador	18
1.5.2. Arquitectura interna de un microcontrolador.....	19
1.6. Procesador ARM Cortex-M4.....	20
1.6.1. Familia STM32	21
1.6.2. Familia STM32F4	22

1.7. Descripción periféricos para el desarrollo de la aplicación.....	27
1.7.1. Entradas y salidas de propósito general (GPIOs).....	27
1.7.2. Circuitos Inter-Integrados (I2C).....	28
1.7.3. Universal Asynchronous Receivers/Transmitters (UARTs).....	29
1.8. Utilización de microcontrolador STM32F429IG	31
1.8.1. Características técnicas de placa STM32F429IG	32
1.8.2. ST-Link - Circuito Programmer /Debugger.....	36
1.8.3. Conexión con aplicaciones STM32	37
1.9. Software de programación	38
1.9.1. Programación en C con KEIL μ Vision5	40
1.9.2. Depuración de programas	44
CAPÍTULO II.....	45
2. METODOLOGÍA	45
2.1. Tipo de estudio	45
2.1.1. Descriptivo	45
2.2. Métodos, Técnicas e Instrumentos	45
2.2.1. Métodos.....	45
2.2.1.1. Analítico/Deductivo	45
2.2.2. Técnicas.....	45
2.2.2.1. Observación.....	45
2.2.3. Instrumentos.....	45
2.3. Población y muestra.....	46
2.4. Hipótesis.....	47
2.5. Operacionalización de variables	47
2.6. Procedimientos	47
2.7. Procedimiento y análisis	49
2.7.1. Diseño	49
2.7.1.1. Control PAN&TITL.....	49
2.7.1.2. Descripción de requerimientos de precisión del sistema	50
2.7.1.3. Resolución y precisión del sistema graficador 3D.....	54
2.7.1.4. Esquema electrónico del sistema	58
2.7.1.5. Descripciones del código software.....	59
2.7.2. Visualización en 3D utilizando MATLAB	67
2.7.3. Comprobación de Hipótesis	72

2.7.3.1. Planteamiento de hipótesis estadística	72
2.7.3.2. Establecimiento del nivel de significancia	72
2.7.3.3. Determinación del valor estadístico de prueba	72
CAPÍTULO III.....	85
3. RESULTADOS.....	85
CAPÍTULO IV	86
4. DISCUSIÓN.....	86
CAPÍTULO V.....	88
5. CONCLUSIONES Y RECOMENDACIONES	88
5.1. Conclusiones	88
5.2. Recomendaciones.....	89
CAPÍTULO VI.....	90
6. PROPUESTA.....	90
6.1. Título de la propuesta	90
6.2. Introducción	90
6.3. Objetivos	91
6.3.1. Objetivo General	91
6.3.2. Objetivos Específicos.....	91
6.4. Fundamentación Científico-Técnica.....	91
6.5. Descripción de la propuesta	92
6.6. Diseño Organizacional.....	93
6.7. Monitoreo y Evaluación de la propuesta	93
7. BIBLIOGRAFÍA	94
8. APÉNDICES Y ANEXOS	96

ÍNDICE DE GRÁFICOS

Figura 1.- Diodo láser rojo.	4
Figura 2.- Superficie Difusa Reflectante.....	5
Figura 3.- Superficie Especular Reflectante.....	6
Figura 4.- LIDAR lite v2.....	7
Figura 5.- Dimensiones LIDAR lite v2.....	9
Figura 6.- Servomotor.	11
Figura 7.- Elementos internos de un servomotor.	12
Figura 8.- Relación de ancho de pulso y posición angular.	13
Figura 9.- Servomotor Hitec HS-311.	14
Figura 10.- Controladores de servomotores.	15
Figura 11.- Distribución de componentes del controlador.	16
Figura 12.- Arquitectura Neumann de un microcontrolador.....	18
Figura 13.- Arquitectura Harvard de un microcontrolador.	19
Figura 14.- Diagrama de bloques series STM32F4.	24
Figura 15.- Series STM32F4.....	26
Figura 16.- Configuración de pines.....	28
Figura 17.- Configuración GPIO.....	28
Figura 18.- Diagrama de bloques I2C.	29
Figura 19.- Flujo de datos I2C.	29
Figura 20.- Bloques USARTs.	30
Figura 21.- Los USARTs soportan una comunicación Half-Huplex.	31
Figura 22.- Comunicación entre USARTS y SPI.	31
Figura 23.- Core429I.....	34
Figura 24.- Componentes Core429IG.	34
Figura 25.- Conexión PC-Placa.	36
Figura 26.- Interfaz del software STM32 ST-Link Utility.....	37
Figura 27.- Diseño del conector debugging ST-LINK y cable plano.	38
Figura 28.- Entorno Integrado de Desarrollo uVision5.	38
Figura 29.- Componentes del entorno de desarrollo Keil µVision.	39
Figura 30.- Pasos en el diseño de un programa para microcontrolador.	41
Figura 31.- Creación de un nuevo proyecto.	42
Figura 32.- Selección microcontrolador STM32F429IG de la Familia STMicroelectronics.	42
Figura 33.- Copia de archivo “startup_stm32f429_439xxx.s” en carpeta del proyecto.....	43
Figura 34.- Agregar archivos en Source Group.	43
Figura 35.- Diagrama de conexiones sistema graficador 3D.	48
Figura 36.- Estructura para Control PAN y TILT.....	49
Figura 37.- Controlador de servos de 6 canales.	49
Figura 38.- Representación del sistema mecánico que sostiene el sensor láser....	50
Figura 39.- División de cuadrantes y ubicación del sensor láser.	51
Figura 40.- Divisiones y ángulos del sistema.....	52
Figura 41.- Ubicación de puntos a detectar en el sistema.	52
Figura 42.- Entorno de prueba.	53

Figura 43.- Entorno de prueba sobrepuesto en el sistema.....	53
Figura 44.- Reconocimiento del sistema de un entorno de prueba.	54
Figura 45.- Reconocimiento de barrido horizontal.	54
Figura 46.- Reconocimiento de barrido vertical.	55
Figura 47.- Reconocimiento del área cubierta por el sistema.	55
Figura 48.- Precisión Lineal Sistema.	57
Figura 49.- Diagrama esquemático-electrónico del sistema.	58
Figura 50.- Diagrama funcional del proceso ejecutado en el microcontrolador...	59
Figura 51- Diagrama de flujo del almacenamiento de datos en la matriz.....	66
Figura 52.- Representación de la matriz de graficación.....	69
Figura 53.- Representación con interpolación de datos.	70
Figura 54.- Representación con mapa de colores blanco-negro.	70
Figura 55.-Representación con mapa de colores negro-blanco.	71
Figura 56.- Zonas de aceptación y rechazo.....	72

ÍNDICE DE TABLAS

Tabla 1.- Características del sensor láser LIDAR lite v2.....	7
Tabla 2.- Características del láser del sensor LIDAR lite v2.....	8
Tabla 3.- Comparación de especificaciones de servomotores según el fabricante	13
Tabla 4.- Especificaciones técnicas Hitec HS-311.	14
Tabla 5.- Clasificación de la familia STM32.	22
Tabla 6.- Clasificación de la familia STM32F4.....	23
Tabla 7.- Operacionalización de variables.	47
Tabla 8.- Área cubierta por el sistema graficador 3D.	56
Tabla 9.- Precisión del sistema graficador 3D.	57
Tabla 10.- Configuración de velocidad del sensor Lidar Lite v2.....	64
Tabla 11.- Configuración de número de lecturas del sensor Lidar Lite v2.....	65
Tabla 12.- Valores de error para la mediciones en un ambiente plano.	75
Tabla 13.- Estadísticos para una muestra para prueba N°.1.....	76
Tabla 14.- Prueba para una muestra para prueba N°.1.....	76
Tabla 15.- Valor estadístico Z para prueba N°.1.....	76
Tabla 16.- Valores de error para las mediciones en un ambiente con gradas.	79
Tabla 17.- Estadísticos para una muestra para prueba N°.2.....	80
Tabla 18.- Prueba para una muestra para prueba N°.2.....	80
Tabla 19.- Valor estadístico Z para prueba N°.2.....	80
Tabla 20.- Valores de error para las mediciones en un ambiente con obstáculos. 83	
Tabla 21.- Estadísticos para una muestra para prueba N°.3.....	84
Tabla 22.- Prueba para una muestra para prueba N°.3.....	84
Tabla 23.- Valor estadístico Z para prueba N°.3.....	84

ÍNDICE DE ANEXOS

Anexo 1.- Sistema graficador 3D.....	96
Anexo 2.- Evidencias de pruebas iniciales.....	97
Anexo 3.- Evidencias de pruebas realizadas en un ambiente plano.....	102
Anexo 4.- Evidencias de pruebas realizadas en un ambiente con gradadas.	103
Anexo 5.- Evidencias de pruebas realizadas en un ambiente con obstáculos.	104
Anexo 6.- Datasheet Sensor láser LIDAR lite v2.	1
Anexo 7.- Diagrama de bloques microcontrolador STM32F429xx.....	6
Anexo 8.- Pines procesador ARM Cortex-M4.....	9
Anexo 9. - Core429IG Schematic.	10

RESUMEN

El presente proyecto describe el diseño y la construcción de un sistema graficador en 3D mediante la integración de dispositivos capaces de establecer la ubicación y parámetros de distancias de objetos en un perímetro de reconocimiento, para posteriormente realizar la graficación de ese perímetro a través de la utilización del software MATLAB.

El sistema consta de un bloque de adquisición de datos de distancia en el que se utiliza un sensor láser LIDAR lite v2, que por sus características de velocidad en la recolección de muestras es óptimo para esta aplicación. El sensor será manipulado por medio de un sistema mecánico con un arreglo de servomotores permitiendo realizar un barrido PAN de 60° y un barrido TILT de 32° del área a explorar.

Para la interpretación de los datos obtenidos del sensor láser se utilizó un microcontrolador STM32F429IG de la familia STMicroelectronics, que incorpora un procesador ARM Cortex-M4, utilizado en dispositivos de alto desempeño. Las pruebas se realizaron en tres distintos escenarios con ambientes controlados para determinar la precisión y funcionamiento del sistema; ambiente plano, ambiente con gradas y ambiente con obstáculos.

El procesamiento realizado por el microcontrolador se lo acondiciona para ser interpretado por el software MATLAB y posteriormente realizar la representación gráfica en 3D.



UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CENTRO DE IDIOMAS



Lic. Rosa Fernández

12 de Enero de 2016

SUMMARY

This project describes the design and construction of a 3D graphic system using integration of devices which are able to set location and range features of objects into a recognition perimeter, after that, the system will plot this perimeter using MATLAB software.

The system has a block of acquiring distance data where LIDAR lite v2 laser sensor is used; this sensor is optimum for this application because it has great speed features in data collection. This sensor will be manipulated by a mechanical system with a servomotor system, therefore it is possible to make 60 degrees PAN sweeping and 32 degrees TILT sweeping from the area to be explored.

For the data interpretation from the laser sensor, it employed a STM32F429IG microcontroller from STMicroelectronics family which contains the ARM Cortex-M4 processor used in high performance devices. The proof was tested in three different sceneries with controlled environments to determine the accuracy and performance system: flat surface, environment with stairs and environment with obstacles.

The processing was made by the microcontroller and it was conditioned to be interpreted by MATLAB software and finally, it'll make the 3D graphic representation.



INTRODUCCIÓN

Los seres humanos somos capaces de percibir los objetos que se encuentran a nuestro alrededor de manera tridimensional, es decir que identificamos las relaciones de profundidad con respecto a un plano y determinamos los puntos que conforman los objetos. El sistema visual humano ha sido tratado de ser emulado por muchos dispositivos desarrollados en la actualidad y ser aplicados para brindar soluciones a la vida cotidiana.

Se han conseguido por medio de procesamiento de ordenador la representación tridimensional de objetos puntuales con gran precisión. A su vez se han creado dispositivos capaces de establecer la ubicación y parámetros de distancias de objetos en un perímetro de reconocimiento. Sin embargo son pocos los sistemas que convergen estas características de reconocimiento de contorno con su respectiva representación gráfica tridimensional.

La determinación de la ubicación y las mediciones de distancia de un objeto dado con respecto a un punto de origen, son tareas que se realizan de manera manual y prescinden de herramientas que no pueden brindar facilidad e independencia a la persona que intenta realizar la medición. Por otra parte, los dispositivos que permiten la detección de obstáculos son de difícil adquisición y acceso en el mercado, debido a que sus costos son altamente elevados

Una vez determinados los factores que inciden en la adquisición y reconocimiento de objetos en el espacio, el presente proyecto propone el diseño de un sistema graficador 3D que permitirá determinar la ubicación, distancia y representación gráfica de obstáculos en una superficie.

CAPÍTULO I

1. FUNDAMENTACIÓN TEÓRICA

1.1. Antecedentes

El presente proyecto de investigación toma como punto de partida el diseño de un sistema graficador 3D, para lo cual se ha investigado si existen proyectos similares o relacionados al tema. Es muy importante obtener un fundamento previo al desarrollo del proyecto, con el fin de encontrar soluciones prácticas y adecuadas en base a la experiencia de diferentes autores; por tal motivo se detallan a continuación los siguientes temas:

“RECONSTRUCCIÓN ACTIVA DE OBJETOS 3D MEDIANTE ESCANEEO LÁSER Y GENERACIÓN DE LA VISTA POR MEDIO DEL SOFTWARE MATLAB”. - El proyecto elaborado por Luis Geovanny Ilvay Paca, Escuela Politécnica Nacional, consiste en el diseño del hardware y software de una plataforma que permite reconstruir objetos 3D mediante escaneo láser. Para lo cual se usó un sensor láser URG-04-LX-UG01. Esta plataforma permite rotar el objeto 360° sobre un eje haciendo que el láser adquiera los datos para la reconstrucción del objeto.

“INTEGRACIÓN DE UN LÁSER 3D EN EL MANIPULADOR MÓVIL MANFRED”. - El proyecto elaborado por Raúl Villajos Rayo, Universidad Carlos III de Madrid, se basa en la utilización de un sensor láser Hokuyo UTM-30lx y un motor Dynamixel EX-106+ para hacer girar el láser, y la creación del software necesario para hacer que estos cumplan las funciones necesarias, con el fin de obtener mapas tridimensionales para la posterior utilización del robot manipulador MANFRED.

“DISEÑO DE UN ESCÁNER LÁSER 3D A PARTIR DE UN SISTEMA DE MEDICIÓN LÁSER 2D SICK LMS 200”.- El proyecto elaborado por MsC. Jaime Andrés Velásquez M., Universidad Manuella Beltrán , presenta el diseño de un escáner 3D empleando un escáner 2D LMS 200 de la marca SICK y un servo como actuador mecánico. Con este diseño se pretende obtener mejores tiempos y área de sensado que la mayoría de escáneres láser 3D comerciales. Se presentan diferentes métodos de escaneo, la caracterización de sus distribuciones de densidades de mediciones y una idea para corregir errores sistemáticos. Por último se plantean mejoras al diseño que permitan emplearlo para el sensado de ambientes dinámicos.

1.2. Sensores de proximidad

Los sensores de proximidad son dispositivos orientados a la detección de objetos por diversos mecanismos; las prestaciones de estos sensores se han convertido en una parte indispensable para la industria debido a que permiten determinar la ubicación de un objeto con gran precisión y sus aplicaciones son numerosas, desde el conteo de piezas, movilización o accionamiento de mecanismos.

Existen diversos tipos de sensores de proximidad, sin embargo se los puede agrupar dentro de tres categorías:

- Tipo Inductivo
- Tipo Magnético
- Tipo Capacitivo

Sensor Tipo Inductivo

Este tipo de sensor produce o induce un campo magnético cuando circula corriente por la bobina, produciéndose un campo magnético de distinta dirección con respecto al campo de la bobina, cuando se acerca un objeto metálico.

Sensor Tipo Magnético

El principio de funcionamiento de este tipo de sensores es basado en el efecto magneto-resistivo, que produce una modificación en el comportamiento del valor de la resistencia de un material, al ser expuesto a un campo magnético externo.

Sensor Tipo Capacitivo

Los sensores capacitivos utilizan la propiedad que poseen los cuerpos para mantener una carga eléctrica; esta propiedad se ve reflejada cuando se acerca un objeto al sensor, se aumenta la capacitancia, mientras que cuando se aleja se disminuye el valor capacitivo.

1.2.1. Láser

Es un dispositivo que emite un haz de luz potente. El término láser, proviene de la sigla en inglés **LASER**¹, que significa Amplificación de Luz por Emisión Estimulada de Radiación.

Estas características hacen que el láser resulte muy útil tecnológicamente. Actualmente, existen dispositivos láser de amplia variedad y se los utiliza en la informática como en reproductores de audio o vídeo. Y sin duda los mayores beneficios son los que se ha brindado en la medicina, especialmente en las operaciones quirúrgicas. (Batanga, 2015)



Figura 1.- Diodo láser rojo.
Fuente: Robotshop.

¹ Light Amplification by Stimulated Emission of Radiation

1.2.2. Propiedades de láser

Reflectividad

Las características reflectantes de la superficie de un objeto se pueden dividir en tres categorías:

- Difuso Reflectante
- Especular Reflectante
- Retro reflectante

Difuso Reflectante

En el caso de superficies puramente difusas, comprende los materiales que tienen una textura en la que la energía se refleja y se dispersa uniformemente. Esta característica tiene como resultado un porcentaje relativamente alto para que la energía láser se disperse y retorne al receptor; consecuentemente, estos materiales tienden a reflejar de manera eficiente. Los materiales que forman esta categoría son: papel, paredes, y granito. Es importante tener en cuenta que los materiales que pertenecen en esta categoría, debido a la reflexión observada en longitudes de onda de luz visible, pueden presentar interferencia inesperada en otras longitudes de onda.

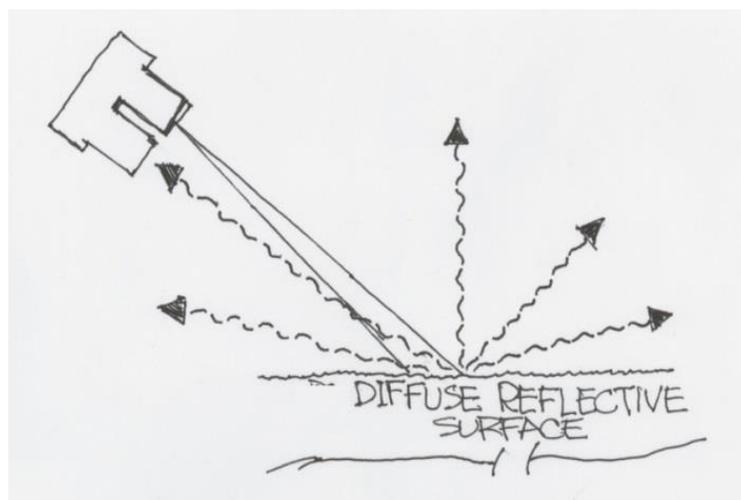


Figura 2.- Superficie Difusa Reflectante.
Fuente: pulsedlight3d.

Especlar Reflectante

Las superficies especulares o de espejo son difíciles o imposibles de reconocer por el láser, porque la energía radiada no se dispersa. Las reflexiones fuera de las superficies especulares tienden a reflejar con poca dispersión, produciendo que el haz reflejado sea pequeño y no se refleje directamente de vuelta a un receptor, perdiendo por completo la señal emitida. El láser no puede detectar un objeto especlar de frente, a menos que el objeto se posicione perpendicularmente. Las superficies especulares más comunes son: espejos y vidrios.

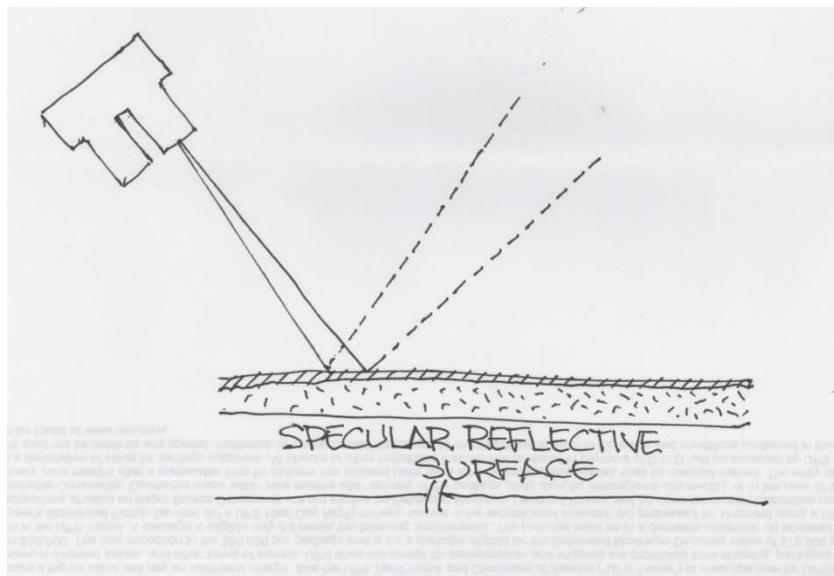


Figura 3.- Superficie Especlar Reflectante.

Fuente: pulsedlight3d.

Retro reflectante

Las superficies retro reflectantes devuelven un porcentaje muy alto de la energía radiada al receptor, debido a sus propiedades de reflexión. Si la luz golpea una superficie retro reflectante, ésta regresa al receptor sin mayor pérdida de señal. Las superficies consideradas retro reflectantes son: la pintura utilizada para marcar los caminos, los ojos de los animales, placas y señales de tráfico.

1.2.3. Sensor láser LIDAR Lite v2

El sensor láser LIDAR Lite v2, como se aprecia en la Figura 4, es un nuevo dispositivo con características adecuadas para mediciones ópticas de distancia. El sensor Lidar Lite es ideal para ser usado en drones, robots o vehículos no tripulados, donde se necesita una gran precisión en las mediciones y en condiciones de poco espacio.



Figura 4.- LIDAR lite v2.
Fuente: Sparkfun.

En cuanto a las especificaciones del sensor láser LIDAR lite v2 se encuentran las siguientes características proporcionadas en la Tabla 1:

Alimentación	4.75 – 5.5 V DC Nominal
Consumo de corriente	<100 mA (Mediciones continuas)
Interface I2C	100 kb, Dirección esclavo 0XC4.
Salidas de sensor	PWM I2C
Rango máximo	- 40 m
Precisión	+/- 2.5 cm
Rep. rate por defecto	- 50 Hz

Tabla 1.- Características del sensor láser LIDAR lite v2.
Fuente: Sparkfun.

1.2.4. Características del láser del sensor LIDAR lite v2

Las características del láser incorporado corresponden a la Tabla 2, mostrada a continuación:

Longitud de Onda	905 nm (Nominal)
Potencia Total	1.3 Watts
Modo de Operación	Pulsos (256 pulsos máximo)
Ancho de Pulso	0.5 us (50% ciclo de trabajo)
Frecuencia de Repetición de Pulsos	10-20 KHz
Energía por Pulso	<280 nJ
Apertura del Diámetro del Haz Láser	12mm x 2mm

Tabla 2.- Características del láser del sensor LIDAR lite v2.
Fuente: Sparkfun.

El sensor LIDAR Lite v2., es designado como un producto láser de Clase 1, que según las normativas UNE EN 60825-1/A2-2002², determinan que este tipo de láser son seguros bajo todas las condiciones de funcionamiento, incluyendo la utilización de instrumentos de forma directa. Sin embargo es recomendable evitar la exposición directa de los ojos hacia el haz del láser, por motivos de seguridad y prevención.

1.2.5. Dimensiones sensor láser LIDAR lite v2

Las dimensiones correspondientes al sensor LIDAR lite v2 se pueden observar en la Figura 5:

² Norma UNE (Una Norma Española) EN 60825-1/A2-2002, elaborada por el Comité Tecnológico de Normalización 209 de AENOR

PulsedLight, Inc. - LIDAR-Lite Housing

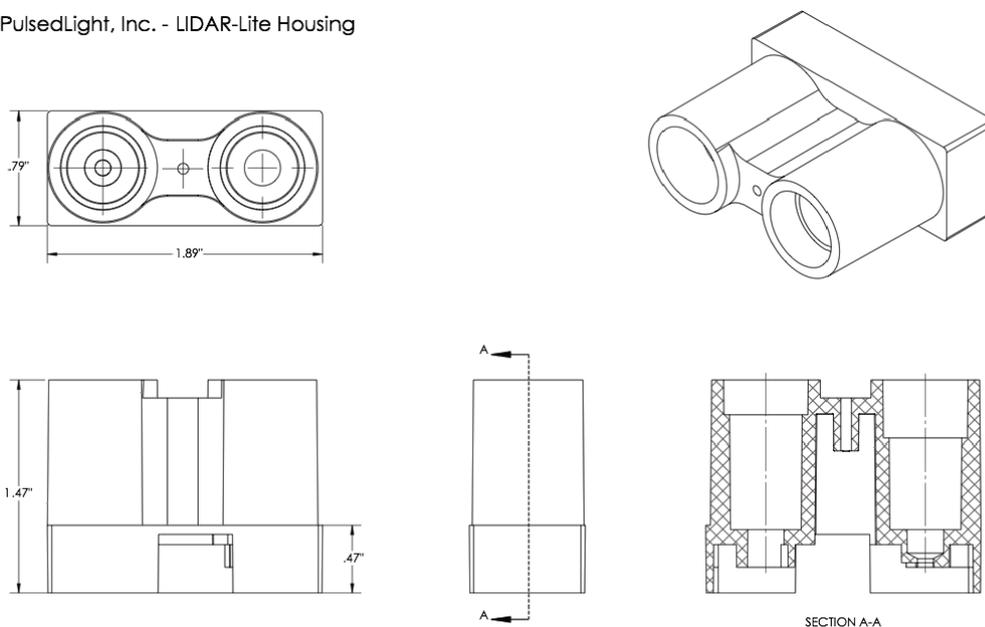


Figura 5.- Dimensiones LIDAR lite v2.
Fuente: PulsedLight.

1.2.6. Descripción pines del sensor LIDAR lite v2

- **Pin 1** - POWER_IN – 4.75-5.5V DC Nominal, Máximo 6V DC. Picos de corriente que ocurre en esta entrada (se produce durante el periodo de adquisición) son típicamente < 100 mA durante un periodo de 4 a 20 ms dependiendo de la fuerza de la señal recibida.
- **Pin 2** - POWER_EN – Activo en alto, permite la operación del regulador del microcontrolador de 3.3 V. En estado bajo pone a la tarjeta en modo sleep, draws <40 μ A. (Resistencia Interna 100K pull-up).
- **Pin 3** - Mode – Select – Proporciona disparo (Nivel alto a bajo) Salida PWM (Nivel alto).
- **Pin 4** - I2C Clock (SLC).
- **Pin 5** - I2C Data (SDA).
- **Pin 6** - Signal/power ground.

1.2.7. Cálculo de distancia del sensor LIDAR lite v2

El sensor láser transmite un haz direccionado para reflejar en un objetivo, con una porción de señal de retorno que pueda ser reflejada al receptor. La distancia es calculada tomando la diferencia entre el tiempo de la transmisión de la señal y el tiempo de recepción de la señal. Sin embargo la recepción correcta de la señal es altamente influenciada por algunos factores como: distancia y tamaño del objetivo, reflexión de superficies.

La relación de la distancia (D) con la intensidad de la señal de retorno es inversamente proporcional al cuadrado de la distancia, de tal manera que con el incremento de la distancia la intensidad de la señal de retorno disminuye:

$$\text{señal de retorno} = \frac{1}{D^2} \quad (1)$$

Además, la relación existente de la sección transversal del objetivo (C) a reconocer con respecto a la intensidad de la señal de retorno es inversamente proporcional a la cuarta potencia de la sección transversal:

$$\text{intensidad de señal de retorno} = \frac{1}{C^4} \quad (2)$$

El sensor transmite un haz láser direccionado que conforme a la distancia se sigue ampliando; alrededor de un metro de distancia, el haz del láser es del tamaño de la lente del sensor; al sobrepasar la barrera de un metro, el haz se amplía y puede ser calculado aproximadamente, dividiendo la distancia para el factor de 100 u 8 miliradianes.

La orientación del objetivo así como el sensor, afecta la sección transversal del objeto a reconocer y consecuentemente, la cantidad de la señal de retorno disminuye conforme la perpendicularidad del objetivo.

1.3. Servomotores

Los servomotores, representados en la Figura 6, son dispositivos que incorporan un eje de movimiento controlado por un sistema electrónico, que envía una señal de carácter analógica o digital para fijar una posición angular dentro del dispositivo.



Figura 6.- Servomotor.
Fuente: Tower Pro.

Los servomotores son útiles en diversas aplicaciones como: robótica, control, mecánica, modelismo; debido a que sus características de tamaño, funcionamiento y movimiento son muy variadas en el mercado.

1.3.1. Funcionamiento de un servomotor

El principio de funcionamiento de un servomotor está determinado por el motor interno que posee algunos circuitos de control y un potenciómetro, como se representa en la Figura 7. Ambos elementos son conectados al eje central del servomotor. El circuito de control es capaz de supervisar el ángulo actual del servomotor por medio del potenciómetro, de tal forma que si el eje está en el ángulo correcto, entonces el motor está apagado. Si el circuito verifica que el ángulo no es el correcto, el motor gira en la dirección adecuada hasta llegar al ángulo correcto. El eje del servo es capaz de llegar alrededor de los 180 grados; sin embargo este rango de movimiento varía según el fabricante y en algunos

casos el rango se amplía alrededor de los 210 grados. Un servomotor normal tiene un movimiento angular de entre 0 y 180 grados.

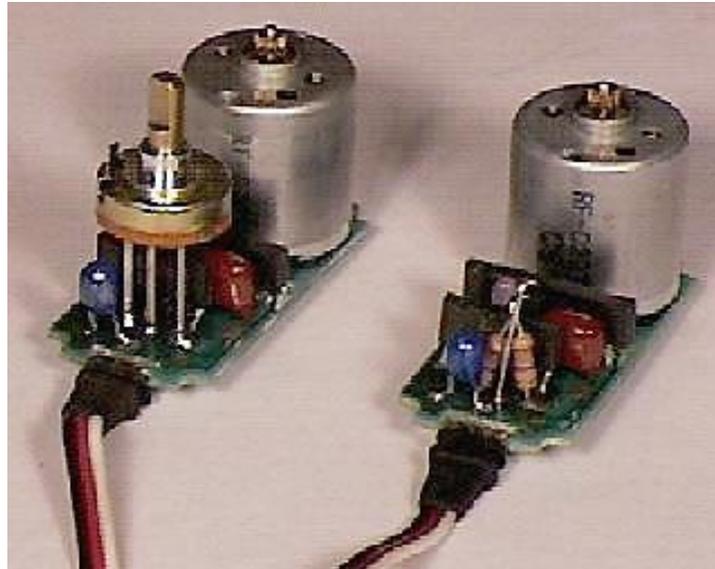


Figura 7.- Elementos internos de un servomotor.
Fuente: Tecnoficio.

Con respecto a la cantidad de voltaje que es necesario suministrar al motor, corresponde a una relación de control proporcional de la distancia que necesita recorrer, es decir que para recorrer distancias grandes el motor se mueve a toda velocidad, mientras que para recorrer pequeñas distancias el motor utiliza una velocidad inferior.

1.3.2. Control de un servomotor

Un servomotor es controlado por medio de señales recibidas que indican la posición exacta a la que se sitúa el eje del motor. Generalmente los servomotores tienen tres cables externos usados como entradas de: voltaje positivo, tierra y datos. El control que se aplica en este tipo de servomotores se genera con una señal que modula un ancho de pulsos (PWM). Se debe aplicar un pulso de duración y frecuencia específicas, como se observa en la Figura 8, teniendo una relación de posición angular y el valor del ancho del pulso enviado al control del servo.

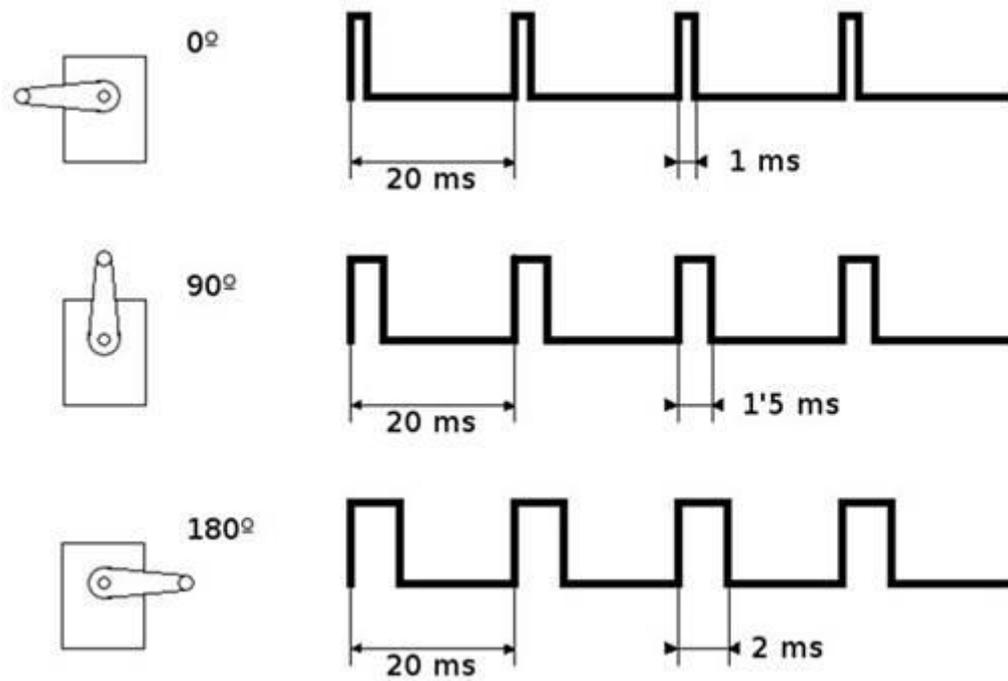


Figura 8.- Relación de ancho de pulso y posición angular.
Fuente: Wordpress.

Es importante conocer que la disposición de cables y el tiempo de duración de los pulsos, son determinados por el fabricante como se puede visualizar en la Tabla 3, a continuación:

Fabricante	Duración pulso (ms)				disposición de cables		
	min.	neutral.	máx..				
Futaba	0.9	1.5	2.1	50	rojo	negro	blanco
Hitech	0.9	1.5	2.1	50	rojo	negro	amarillo
Graupner/Jr	0.8	1.5	2.2	50	rojo	marrón	naranja
Multiplex	1.05	1.6	2.15	40	rojo	negro	amarillo
Robbe	0.65	1.3	1.95	50	rojo	negro	blanco
Simprop	1.2	1.7	2.2	50	rojo	azul	negro

Tabla 3.- Comparación de especificaciones de servomotores según el fabricante
Fuente: Pesadillo.

1.3.3. Servo estándar Hitec HS-311

El servo Hitec HS-311 es un servomotor estándar, como se muestra en la Figura 9, utilizado en muchas aplicaciones ya que sus características son de altas prestaciones con respecto al factor precio/funcionalidad. Entre sus principales características se incluyen el material incorporado en sus mecanismos que ofrecen una alta durabilidad.



Figura 9.- Servomotor Hitec HS-311.
Fuente: Jrcmodels.

Las especificaciones técnicas son detalladas en la Tabla 4, representada de la siguiente manera:

Modulación	Analógica
Torque	4.8V: 3.02 kg/cm 6.0V: 3.53 kg/cm
Velocidad	4.8V: 0.19 s/60 grados 6.0V: 0.15 s/60 grados
Peso	42.8 gramos
Dimensiones	Largo: 39.9 mm Ancho: 19.8 mm Altura: 36.3 mm
Tipo de motor	3 polos
Tipo de engranaje	Plástico
Rango de rotación	180 grados
Ciclo de pulso	20 ms
Ancho de pulso	900-2100 us
Tipo de conector	Universal

Tabla 4.- Especificaciones técnicas Hitec HS-311.
Fuente: Autores.

1.4. Controlador de servomotores

La existencia de circuitos especializados en el control de servomotores permite generar las señales directamente con la instrucción de la posición en la que se requiere fijar el servomotor. En la actualidad existen muchos controladores de servos que varían sus características para adaptarse a las necesidades de la aplicación como se aprecia en la Figura 10; principalmente dependen del número de canales a usarse (generalmente desde 4 canales hasta 84 canales), la programación interna o externa que necesite el dispositivo y del protocolo para establecer la comunicación.

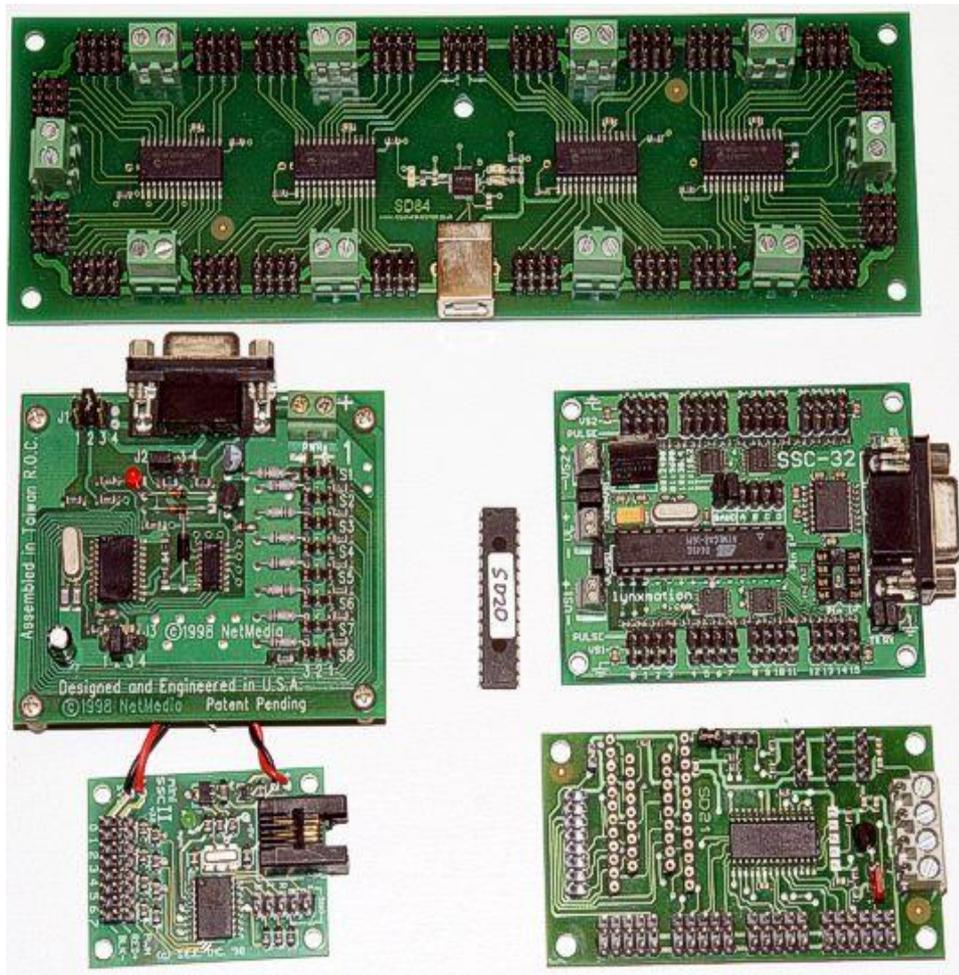


Figura 10.- Controladores de servomotores.

Fuente: Superrobotica.

1.4.1. Controlador Micro Maestro de 6 canales

El controlador Micro Maestro de 6 canales es el dispositivo más pequeño en lo que corresponde a la segunda generación de controladores USB, perteneciente a la empresa desarrolladora Pololu. Tiene como principal función el control de servomotores, a través del incremento del desempeño interno con características de interfaz USB nativa y control interno por medio de programación almacenada.

Los canales pueden ser configurados como salidas para servos de radio control o de control electrónico de velocidad (ESC), también como salidas digitales o entradas analógicas. La exacta resolución de los pulsos comprende una precisión menor a 200 ns, razón por la que este dispositivo es frecuentemente usado en aplicaciones de alto rendimiento.

1.4.1.1. Distribución de pines del controlador de 6 servos

El controlador de servos de 6 canales puede ser conectado a una computadora por vía puerto USB para configurar el dispositivo, además puede ser utilizado mediante solamente por medio del envío de comandos de manera serial. La distribución de los componentes se representa en la Figura 11, mostrada a continuación:

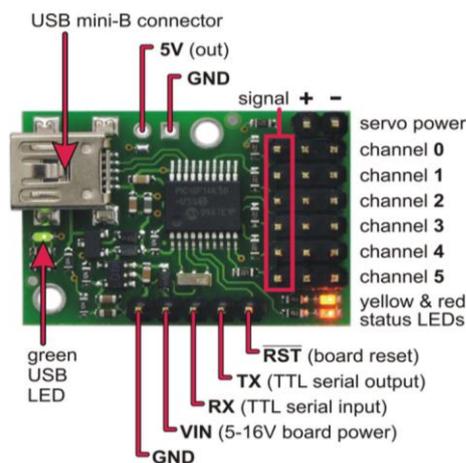


Figura 11.- Distribución de componentes del controlador.

Fuente: Pololu.

- **Alimentación de la placa:** La alimentación puede provenir desde USB o de una fuente externa entre 5 y 16V conectada a las entradas VIN y GND. El procesador interno que tiene la placa es energizado por medio de una fuente externa.
- **Alimentación de servos:** Las conexiones están provistas de una entrada externa para la alimentación de los servomotores, directamente sin el uso de un regulador, por lo que es importante considerar las especificaciones de cada servomotor.
- **Salida 5V:** La placa incluye una salida de 5V para conectar dispositivos externos, con una corriente de hasta 50mA.
- **Líneas de señal:** Envían los pulsos hacia los servomotores. Estos elementos son protegidos con resistencias de 220Ω , limitando la corriente hasta 60 mA.
- **RX:** A través de esta línea se reciben los bytes seriales como señales TTL no invertidas, generalmente usado por protocolos UART. Los bytes pueden ser comandos seriales o bytes arbitrarios para ser retornados a la computadora vía USB.
- **TX:** La línea de transmisión del controlador de servos envía bytes seriales, producto como respuesta de los comandos recibidos por medio de USB.
- **Reset:** La placa incorpora un pin para efectuar un reseteo por medio de un pulso bajo. Esta línea contiene una resistencia pull-high, por lo que no es necesario conectar este pin de manera obligatoria para su funcionamiento.

1.5. Microcontroladores

Un microcontrolador (abreviado uC, UC o MCU) es básicamente una pequeña computadora en un circuito integrado, con característica de contener un núcleo de

procesamiento, memoria y puertos de entrada/salida programables. Los microcontroladores han sido diseñados principalmente para el uso en aplicaciones embebidas, a diferencia de los microprocesadores que son usados en otras funciones de propósito general, teniendo como ejemplo el desempeño de las computadoras personales.

1.5.1. Arquitectura de un microcontrolador

La arquitectura tradicional que presenta un microcontrolador es basada en una propuesta elaborada por John Von Neumann, en la que se hace referencia a la conexión de una unidad central de procesamiento con una memoria que contiene las instrucciones para su funcionamiento y el flujo de programas y datos. El diagrama de la Figura 12 muestra la arquitectura utilizada.

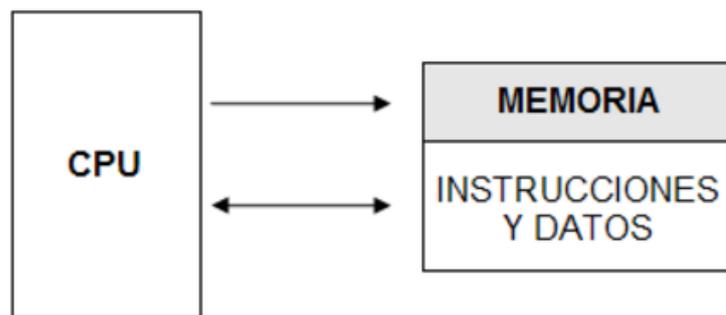


Figura 12.- Arquitectura Neumann de un microcontrolador.
Fuente: Wordpress.

Otro tipo de arquitectura implementada en los microcontroladores es denominada Harvard, en la que la unidad central de procesamiento es conectada a dos memorias diferentes a través de buses distintos, como se aprecia en la Figura 13. La primera memoria contiene exclusivamente las instrucciones del programa mientras que la segunda memoria se reserva solamente a lo que son datos. Claramente se produce una ventaja sobre la arquitectura tradicional al poder leer los datos de una instrucción inicializada al mismo tiempo que pueda leer la siguiente instrucción a efectuarse.

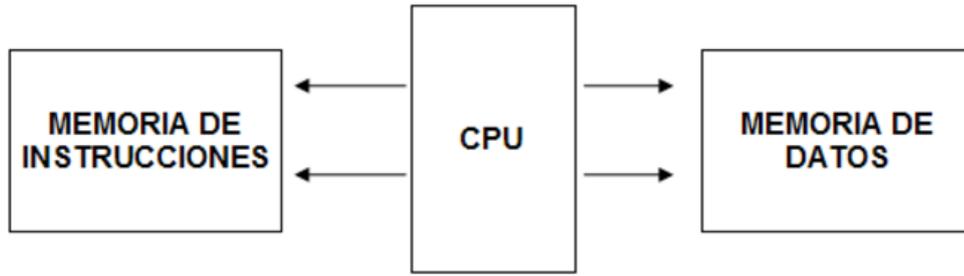


Figura 13.- Arquitectura Harvard de un microcontrolador.
Fuente: Wordpress.

1.5.2. Arquitectura interna de un microcontrolador

Internamente un microcontrolador dispone de tres elementos característicos: una unidad central de procesamiento, una memoria y las unidades de entrada y salida.

Unidad Central de Procesamiento (CPU)

La CPU es el núcleo del microcontrolador encargado de realizar todas las funciones al interpretar las instrucciones provenientes de los distintos programas o entradas al dispositivo, ejecutando la recopilación de datos conjuntamente a un procesamiento.

Las tareas definidas para la CPU se han clasificado en cuatro procesos:

- **Lectura de datos:** Se leen los datos o instrucciones provenientes de la memoria o de una unidad de entrada/salida.
- **Interpretación de instrucciones:** Se decodifica la instrucción para ejecutar la tarea asignada.
- **Procesamiento de datos:** Durante el proceso de las tareas asignadas se puede requerir el uso de operaciones aritméticas o lógicas de los datos.
- **Escritura de datos:** Se pueden escribir datos hacia la memoria o a las unidades de salida.

Memoria

La información que se obtiene dentro del mismo microcontrolador o de diferentes unidades de entrada/salida, pueden ser almacenada en la memoria durante un tiempo determinado, teniendo dos tipos de memoria: memoria del programa y memoria de datos.

Además es importante manifestar la existencia de varios tipos de memoria, cada una referente a una actividad específica y una función determinada, como por ejemplo la memoria no volátil que permite mantener las instrucciones de operación a un microcontrolador cada vez que se inicializa.

Unidades de entrada/salida.

Un microcontrolador es capaz de establecer un intercambio de datos con el exterior, por lo que emplea las unidades de entrada y salida, definidas para lectura y escritura de datos respectivamente

1.6. Procesador ARM Cortex-M4

El procesador ARM Cortex-M4 es un procesador de bajo consumo de energía, baja latencia en interrupción, y depuración de bajo costo. El Cortex - M4F es un procesador con la misma capacidad y similares funciones que el procesador Cortex - M4, e incluye funcionalidad aritmética de punto flotante (Floating Point Unit). Ambos procesadores están diseñados para aplicaciones de características embebidas que requieren parámetros de respuesta de interrupción rápida.

Características

El procesador Cortex- M4 incorpora las siguientes características:

- Un núcleo.

- Un Nested Vectored Interrupt Controller (NVIC) reducidamente integrado con el core del procesador para lograr procesamiento de interrupciones de baja latencia.
- Múltiples interfaces de bus de alto rendimiento.
- Una solución de depuración de bajo costo con la capacidad opcional de:
 - Implementar breakpoints and code patches
 - Implementar watchpoints, tracing, y system profiling
 - Soportar estilo de depuración printf().
 - Vincular hacia un Trace Port Analyzer (TPA).
- Una unidad de memoria opcional - Memory Protection Unit (MPU).
- Una unidad de punto flotante (FPU).

Interfaces

El procesador tiene las siguientes interfaces externas:

- Memoria múltiple y dispositivos con interface bus.
- Interfaz ETM.
- Interfaz de puerto de seguimiento.
- Interfaz de puerto de depuración.

1.6.1. Familia STM32

ST es una empresa líder a nivel mundial que provee soluciones utilizando semiconductores y dispositivos integrados. Dentro de sus productos se encuentran disponibles los microcontroladores de 32 bits o conocidos también como familia STM32.

El desarrollo y funcionamiento de STM32 se basa en los procesadores Cortex-M de ARM, implementados para ofrecer una mayor libertad a los usuarios combinando su alto rendimiento, procesamiento digital de señales, bajo consumo y utilidades en tiempo real.

Los microcontroladores que pertenecen a esta familia se distribuyen de acuerdo a las características establecidas conforme a la Tabla 5:

La amplia gama de los microprocesadores STM32 son basados en estándares industriales, por lo que se ha dado prioridad a su núcleo en lo que refiere a desempeño y mediante la existencia de software y herramientas, son dispositivos aptos para pequeños y grandes proyectos de desarrollo e implementación.



Tabla 5.- Clasificación de la familia STM32.

Fuente: ST MicroElectronics.

1.6.2. Familia STM32F4

La serie STM32 contiene una subfamilia que se identifica con la serie STM32F4, que incorpora el núcleo Cortex™ - M4, como se representa en la Tabla 6. Esta serie introduce al mercado y a disposición de los usuarios un DSC (Digital Signal Controller).

 Product	F _{cpu} (MHz)	Flash (bytes)	RAM (KB)	Ethernet I/F IEEE 1588	Camera I/F	SDRAM I/F		SAI ³ I/F	Chrom-ART Graphic Accelerator™	TFT LCD controller	MPI DSI
				2x CAN		Dual Quad SPI	SPDIF RX				
STM32F469 ²	180	512 K to 2 M	384	• •	•	• •	•	•	•	•	•
STM32F429 ²	180	512 K to 2 M	256	• •	•	•	•	•	•	•	
STM32F427 ²	180	1 to 2 M	256	• •	•	•	•	•	•		
STM32F446	180	256 K to 512 K	128	• •	•	• •	• •	• •			
STM32F407 ²	168	512 K to 1 M	192	• •	•						
STM32F405 ²	168	512 K to 1 M	192	•							

Tabla 6.- Clasificación de la familia STM32F4.

Fuente: ST MicroElectronics.

Con una frecuencia máxima en la CPU de 180 MHz, las series STM32F429 y STM32F439 aumentan el desempeño en rendimiento de la serie STM32F4. El acelerador ART permite la latencia 0 en la ejecución desde flash, y entregan 225 DMIPS (Dhrystone MIPS) y 606 coremark score (EEMBC Coremark). Estas utilidades permite a STM32 ofrecer dispositivos pin-to-pin y software compatible con la serie STM32F2, pero con un mayor rendimiento, capacidad de DSP, una unidad de punto flotante, SRAM superior y mejoras en periféricos como controlador LCD TFT con aceleración gráfica, SDRAM, como se observa en la Figura 14.

Adicionalmente tiene una interfaz de audio serial, a menos de 1 μ A RTC y 2,4 Msp ADCs. Las características principales de ARM® Cortex-M4 incorporados se definen básicamente en el single-cycle multiply accumulate (MAC) instructions y el optimizado aritmético SIMD.

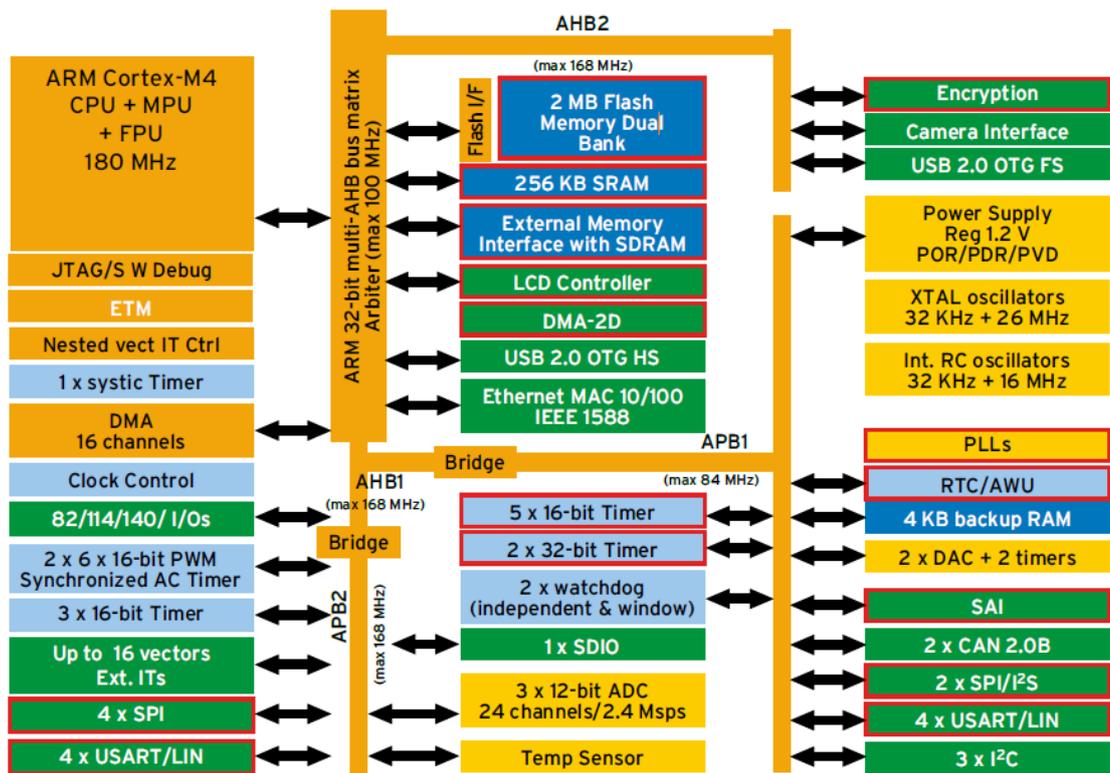


Figura 14.- Diagrama de bloques series STM32F4.
Fuente: EBV's Cortex™ Collection.

Características principales de la serie STM32F4 basada en Cortex-M4.

Las principales características que posee la serie STM32F4 se centran en su arquitectura y periféricos de entrada y salida disponibles:

- Acelerador ST ART: Desempeño: 225 DMips ejecutando el código desde la flash a 180 MHz, el cuál es el máximo desempeño de un core Cortex-M4 que puede adquirir a esta frecuencia.
- Matriz de bus Multi AHB: Mejor ancho de banda disponible para el CPU y los controladores DMA.
- Memorias: hasta 2 MB de Flash y 256 KB de SRAM incluyendo 4 bloques separados para más desempeño y flexibilidad.
- 4 KB + 80 Bytes de backup SRAM.

- Controlador de memoria flexible: con hasta 32-bit de bus de datos apoyado con SRAM, PSRAM/SDRAM/Compact
- Flash/ Memorias NOR y NAND
- Controlador LCD - TFT resolución VGA con acelerador gráfico ChromART.
- Conectividad: Interfaz de Cámara, procesador Crypto/Hash HW, Ethernet MAC10/100 con soporte IEEE 1588 v2, 2 USB OTG (uno con apoyo HS).
- Audio: Audio dedicado PLL, 2 I2S full duplex, 1 Interfaz de audio serial con I2S soporte TDM.
- Hasta 20 interfaces de comunicación (incluyendo 8 USART) con velocidad hasta 10.5 Mbit/s, 6 SPI con velocidad de 37.5 Mbit/s, 3 I2C (con filtros digitales), 2 CAN, SDIO).
- Análogo: Dos DACs de 12-bit, Tres ADC de 12-bit alcanzando 7.2 MSPS en modo interleaved.
- Hasta 17 timers: 16-bit y 32-bit todos trabajan a 168 MHz.
- VDD: desde 1.7V hasta 3.6V

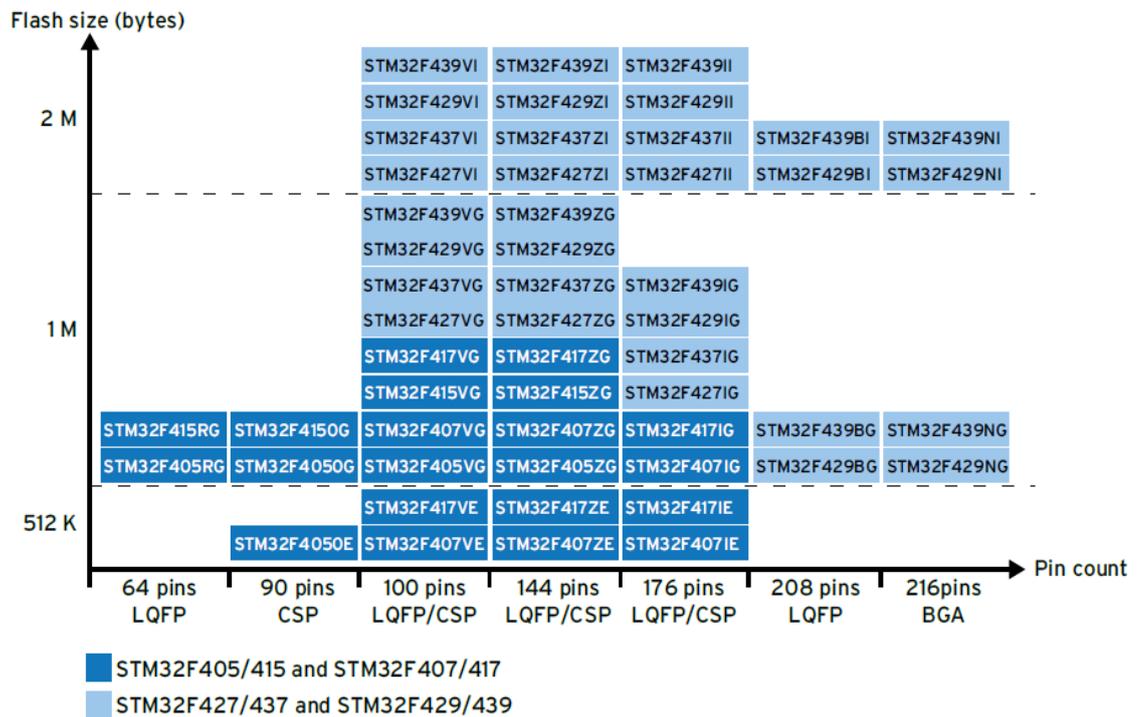


Figura 15. - Series STM32F4.
Fuente: EBV's Cortex™ Collection.

Aplicaciones de la serie STM32F4.

El campo de aplicación STM32F4 es muy amplio al brindar muchas prestaciones y versatilidad al mismo tiempo, por lo que las áreas más destacadas son las siguientes:

- Industrial
- Impresoras
- Redes industriales
- Inversores solares
- Sistemas de alarma
- Control de accesos
- Regulación térmica
- Medidores de energía
- Medidores de glucosa

- Cuidado médico portátil
- Aire acondicionado
- Respiradores Artificiales
- Monitoreo de pacientes
- Controladores motores trifásicos
- Aplicaciones de control
- Cocinas de inducción
- Audio
- Juegos
- Periféricos de PCs
- Cámaras digitales, GPS

1.7. Descripción periféricos para el desarrollo de la aplicación.

Los periféricos de uso general en el microcontrolador STM32 consisten en: Entradas y salidas de uso general, controlador de interrupción externa, convertidores análogo digital, unidades de temporizador avanzadas y de uso general; convertidores digital análogo y reloj en tiempo real con registros de backup y pin de anti-tamper.

1.7.1. Entradas y salidas de propósito general (GPIOs)

STM32 está configurado con pines GPIOs, teniendo 80 pines de entrada y salida unidireccionales. Los pines están distribuidos en 5 puertos cada uno con 16 pines de entrada y salida. Cada pin digital puede ser configurado como GPIO o con una función alterna. Cada pin puede simultáneamente ser configurado como 1 de 16 líneas de interrupción externa.



Figura 16.- Configuración de pines.
Fuente: EBV's Cortex™ Collection.

Estos puertos son nombrados de A-E y soportan un voltaje de 5 voltios. Muchos de los pines externos pueden ser conmutados como GPIO y ser entradas y salidas para los periféricos, por ejemplo USART o I2C. Adicionalmente se encuentra la unidad de interrupción externa la cual permite 16 líneas de interrupción externa para ser mapeadas sobre cualquier combinación de líneas GPIO. Cada Puerto GPIO puede tener pines configurados individualmente como entrada o salida.



Figura 17.- Configuración GPIO.
Fuente: EBV's Cortex™ Collection.

1.7.2. Circuitos Inter-Integrados (I2C)

La interfaz I2C es capaz de operar como un bus esclavo o maestro y también manejar un bus en un sistema multi-maestro. La interface I2C soporta velocidades de bus estándar hasta 100kHz y altas velocidades hasta 400 KHz. El periférico también soporta modos de direccionamiento de siete y diez bits. El periférico I2C provee dos interrupciones al procesador Cortex: uno para contención de error y el otro para control de la dirección de comunicación y transmisión de datos.

Adicionalmente, la unidad DMA provee dos canales DMA que pueden leer y escribir datos al buffer de transmisión I2C. Así, una vez que la dirección inicial y los datos transferidos han sido realizados, los datos pueden ser enviados desde el microcontrolador STM32 bajo un control de hardware.

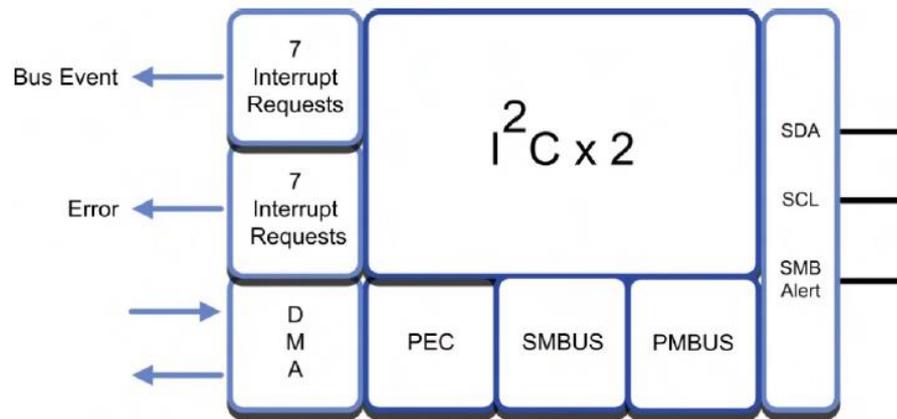


Figura 18.- Diagrama de bloques I2C.
Fuente: Hitex.

Los dos periféricos I2C tienen soporte mejorado para el bus de administración de sistema y de alimentación. Estos incluyen corrección de errores de paquetes mediante hardware.

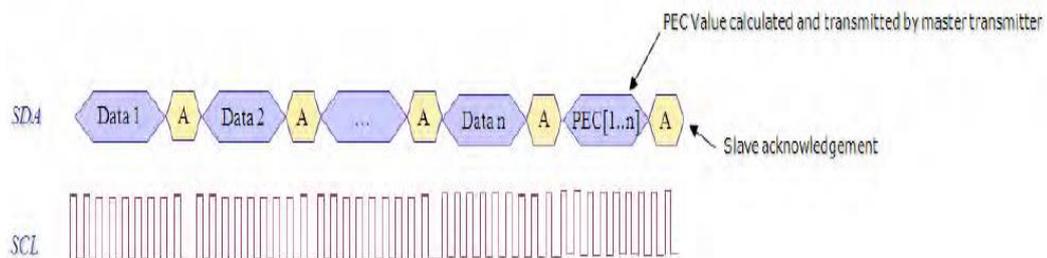


Figura 19.- Flujo de datos I2C.
Fuente: Hitex.

1.7.3. Universal Asynchronous Receivers/Transmitters (UARTs)

Aunque los puertos de comunicaciones serial ya estén desapareciendo de las PCs, estos son todavía ampliamente usados en muchas aplicaciones embebidas como una interface de comunicación simple. El STM32 posee cinco USARTs, cada uno con mejoras en los modos de operación. Cada uno de los USARTs son capaces de alcanzar 4.5 Mbps de comunicación. Además tiene una interface serial

completamente programable con un tamaño de datos de 8 o 9 bits, bit de paridad y de parada y baud rate. Un USART está localizado en el bus APB2 el cual trabaja a 72 MHz, mientras que los otros están localizados en el bus APB1 que trabaja a 36MHz.

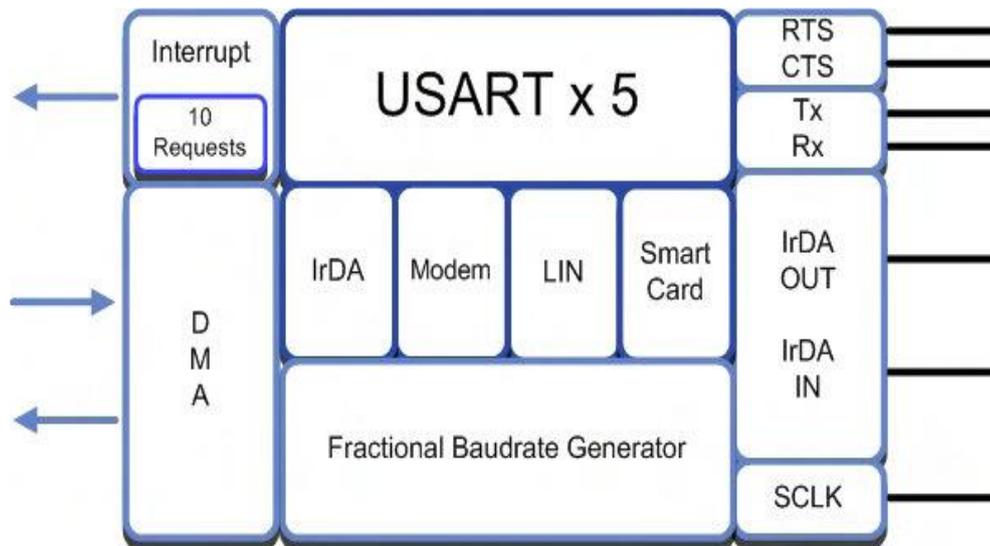


Figura 20.- Bloques USARTs.
Fuente: Hitex.

El generador de baud rate de cada USART es un fraccional del generador de baud rate. Este es más sofisticado que un simple reloj divisor y permite baud rates estándar para ser derivado de cualquier bus de frecuencia.

Parecido a otros periféricos de comunicación serial, cada USART tiene dos canales DMA que son utilizados para transferir datos de transmisión y recepción a la memoria. Cuando se usa como un UART, el USART soporta un número de modos especiales de comunicación. El USART es capaz de operar una comunicación half-duplex mediante un solo cable, usando solo el pin de transmisión. Para el control de flujo en una comunicación entre un modem y un hardware cada USART posee líneas de control adicionales que son CST y RTS.

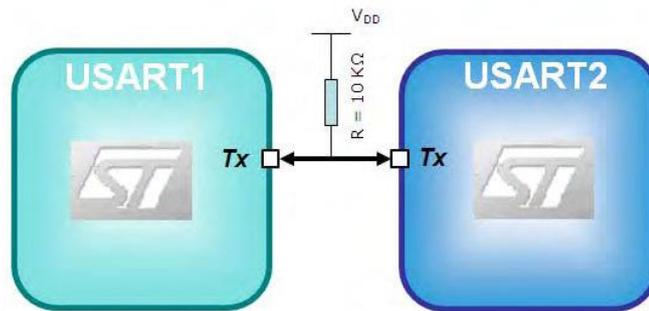


Figura 21.- Los USARTs soportan una comunicación Half-Duplex.
Fuente: Hitex.

El USART puede ser configurado para una comunicación sincrónica, el cual permite una conexión con tres cables a periféricos SPI. Cuando en este modo, La función de un USART como un SPI Master y poseer un reloj programado en polaridad y fase, tan solo de este modo se puede comunicar con cualquier SPI Slave.

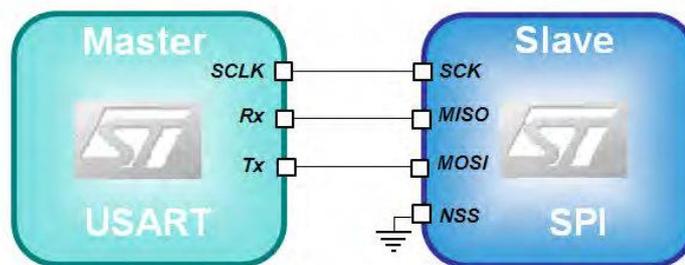


Figura 22.- Comunicación entre USARTS y SPI.
Fuente: Hitex.

1.8. Utilización de microcontrolador STM32F429IG

El microcontrolador seleccionado para llevar a cabo el sistema de adquisición y datos de distancia provenientes del sensor láser debe tener fundamentalmente tres características básicas para cumplir con los objetivos del proyecto:

- Interfaz I2C o PWM
- Processor: ARM Cortex-M4, FPU, MPU, 180 MHz
- USART:4 x 11 Mbit/s USART
- UART:4 x 11 Mbit/s UART

La placa STM32F429IG, ofrece todas las características anteriormente descritas, además de tratarse de una plataforma compacta y versátil que permite la evaluación rápida de prototipos de desarrollo y la creación de múltiples aplicaciones específicas. El kit también incluye una amplia fuente de ejemplos de código y librerías, lo que le permite empezar la creación de aplicaciones en código C de manera simple y sencilla.

Todos los miembros de la familia de productos STM32F4 ®, incluyendo el microcontrolador STM32F429IG, están diseñados en base al núcleo TM ARM Cortex-M4. El procesador ARM Cortex-M4 satisface las necesidades de la aplicación, ya que ofrece un alto rendimiento, a un bajo coste y consumo de energía.

1.8.1. Características técnicas de placa STM32F429IG

La familia STM32F4 incorpora memorias embebidas de alta velocidad y un amplio rango de I/Os mejoradas y periféricos conectados a dos buses APB y tres buses AHB y una matriz de bus multi-AHB de 32 bits.

- 64-Kbyte of CCM (core coupled memory) data RAM
- LCD interfaz paralela, 8080/6800 modos.
- Timer con cuadratura (incremental) encoder input
- 5 V-tolerancia I/Os
- Interfaz paralela de cámara.
- True random number generator.
- RTC: precisión subsecond, hardware calendar
- 96-bit unique ID

Núcleo

ARM Cortex-M4 FPU MPU 180 MHz

- Procesador: ARM Cortex-M4, FPU, MPU, 180 MHz

Temporizador / Contador / PWM

- Timer/Counter: 2 x 32-bit Timer/Counter Module

- Watchdog: 2 x Watchdog
- Timer/Counter: 12 x 16-bit Timer/Counter Module

Reloj y Energía

- RTC: 32 kHz RTC
- Fuente Voltaje: 1.80 V - 3.60 V Voltaje de operación
- Frecuencia máxima de reloj: 180 MHz

Comunicaciones

- I2S: 2 x I2S
- SPI: 6 x 42 Mbit/s SPI
- Other: 1 x SAI Interface
- I2C: 3 x I2C
- USART: 4 x 11 Mbit/s USART
- UART: 4 x 11 Mbit/s UART
- USB, OTG: 2 x USB OTG
- CAN: 2 x CAN
- SDIO: 1 x 8-bit Secure Digital IO
- Ethernet: 1 x 100 Mbit/s Ethernet

Entradas y Salidas

- I/Os: 140 Entradas/Salidas

Análogos

- DAC: 2-canales x 12-bit DAC
- ADC: 24-canales x 12-bit ADC

Otros

- Cámara Interfaz: 1 x 14-bit Cámara Interfaz

Memorias

- ROM: 1024 kB, RAM: 192 kB

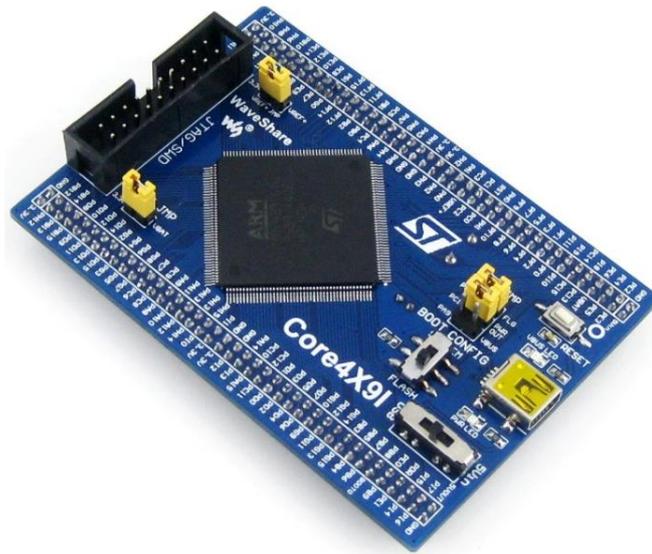


Figura 23.- Core429I.
Fuente: Waveshare Electronics.

En la Figura 24 se puede apreciar una imagen del core429IG con la disposición de sus componentes externos.

Componentes Core429I

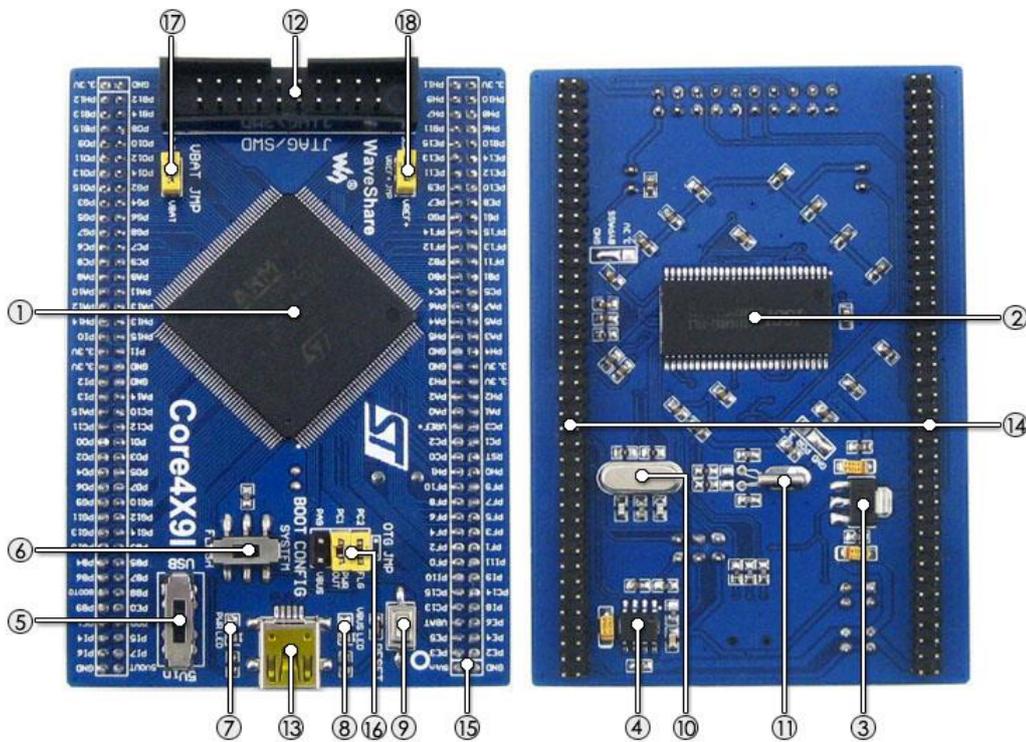


Figura 24.- Componentes Core429IG.
Fuente: Waveshare Electronics.

Descripción detallada de los componentes que se encuentran en la placa.

1. STM32F429IGT6: STM32 microcontrolador:

- **Core:** Cortex-M4 32-bit RISC
- **Característica:** Instrucciones de ciclo único DSP
- **Frecuencia de Operación:** 180MHz, 225 DMIPS/1.25 DMIPS/MHz
- **Voltaje de Operación:** 1.8V-3.6V
- **Empaquetado:** LQFP176
- **Memorias:** 1024kB Flash, 256+4kB SRAM
- **MCU Interfaces de comunicación:**
 - 6 x SPI, 4 x USART, 4 x UART, 2 x I2S, 1 x SAI, 3 x I2C
 - 1 x FMC, 1 x SDIO, 2 x CAN
 - 1 x LCD-TFT
 - 1 x USB 2.0 HS/FS controlador (con DMA dedicado)
 - 1 x USB HS ULPI (PHY externo requerido)
 - 1 x 10/100 Ethernet MAC
 - 1 x 8 to 14-bit camera interfaz
- **AD & DA converters:** 3 x AD (12-bit, 1 μ s, 24 canales compartidos); 2 x DA (12-bit)
- **Depurado/Programación:** soporta interfaces JTAG/SWD, soporta IAP

2. IS42S16400J: SDRAM 1 Meg Bits x 16 Bits x 4 Bancos (64-MBIT)

3. AMS1117-3.3: 3.3V regulador de voltaje

4. MIC2075: onboard USB power management device

5. Power supply switch, alimentado con 5Vin o conexión USB

6. Selección de Boot mode, para configurar pin BOOT0

7. Indicador de encendido

8. VBUS LED

9. Botón de Reseteo

10. 8Mhz cristal

11. 32.768K cristal, para RTC interno con calibración

12. **JTAG/SWD interfaz:** para debugging/programming
13. **USB conector,** soporta Device y/o Host
14. **MCU pines expensor,** VCC, GND y todos los pines I/O son accesibles en los expansores para futuras conexiones.
15. **5Vin pinheader,** Fuente de alimentación de 5V es requerida cuando se usa USB HOST/OTG
16. **USB OTG/HOST jumper**
 - dejar el jumper cuando se usa USB OTG/HOST
 - abrir el jumper para desconectar del puerto I/O relacionado
17. **VBAT selección de jumper**
 - dejar el jumper cuando se usa un sistema de alimentación
 - abrir el jumper para conectar VBAT a una alimentación externa, como una batería.
18. **VREF selección de jumper**
 - dejar el jumper para conectar VREF+ to VCC
 - abrir el jumper para conectar VREF+ a otro pin.

1.8.2. ST-Link - Circuito Programmer /Debugger

Se muestra el diagrama de conexión entre una PC con una placa de evaluación STM32F429IG mediante cable USB, además de la disposición de conexión de cable SWIM/JTAG/SWD y la disposición del Programmer/Debugger, como se representa en la Figura 25.

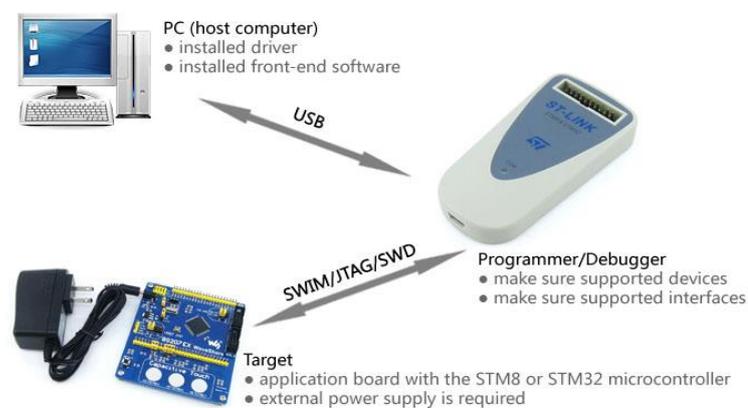


Figura 25.- Conexión PC-Placa.

Fuente: Waveshare Electronics.

La aplicación que permite la conexión de las interfaces tanto física como lógica de los microcontroladores viene en un software denominado ST32 ST-Link Utility, por medio del cual se puede cargar los archivos compilados a la memoria Flash de los dispositivos. La interfaz generada por la aplicación se puede evidenciar en la Figura 26.

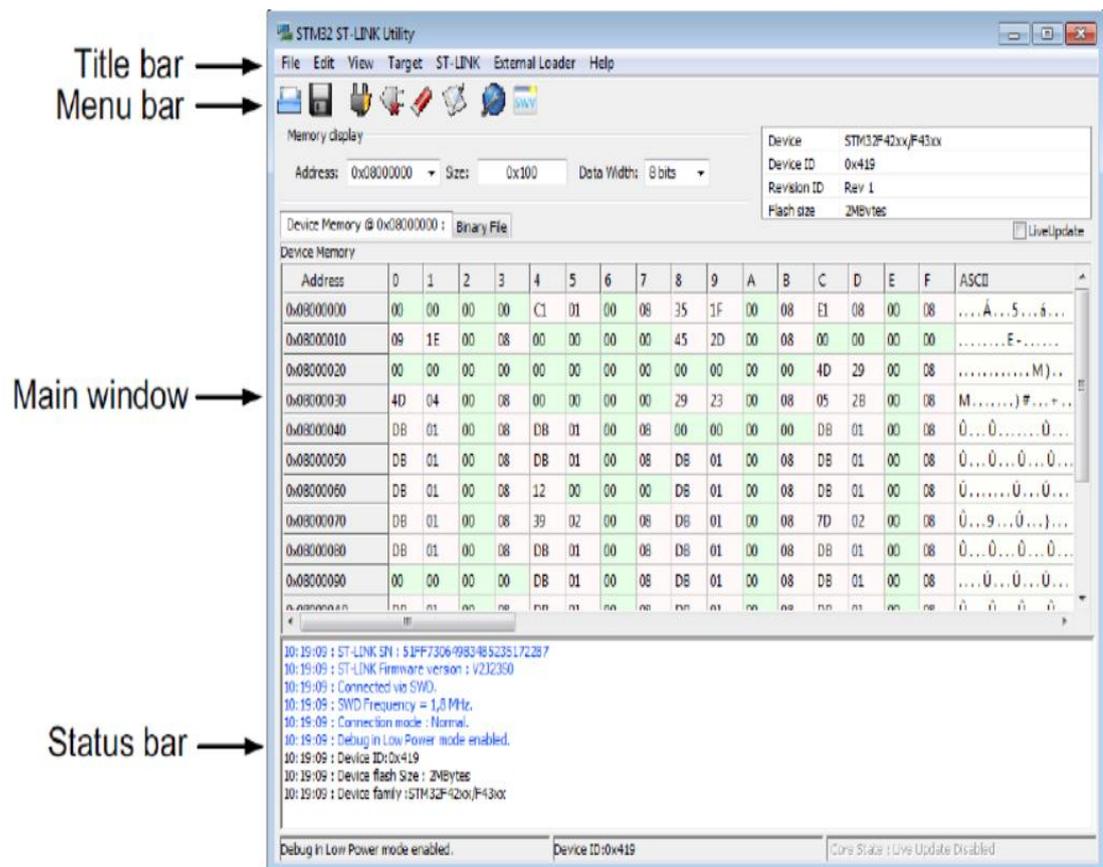


Figura 26. - Interfaz del software STM32 ST-Link Utility
Fuente: ST-Link.

1.8.3. Conexión con aplicaciones STM32

Para desarrollos STM32 la ST -LINK debe estar conectado a la aplicación utilizando la cinta estándar de 20 pines hembra-hembra JTAG plana proporcionada.

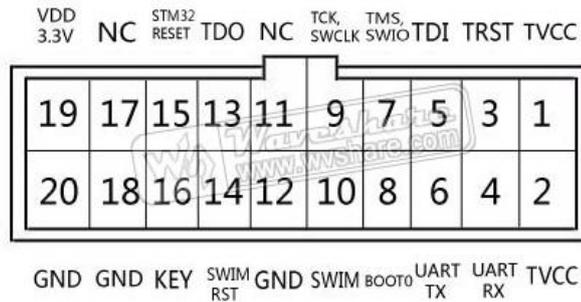


Figura 27.- Diseño del conector debugging ST-LINK y cable plano.
Fuente: Waveshare Electronics.

1.9. Software de programación

La compañía ARM posee el software KEIL MDK-ARM el cual es un entorno de desarrollo de dispositivos basados en procesadores Cortex-M, Cortex-R4, ARM7 y ARM9. Las herramientas incluyen un compilador de C y de C++, e IDE (Entorno Integrado de Desarrollo) llamado uVision5 el cual también sirve para depurar los programas (Debugging) y se muestra a continuación en la Figura 28.

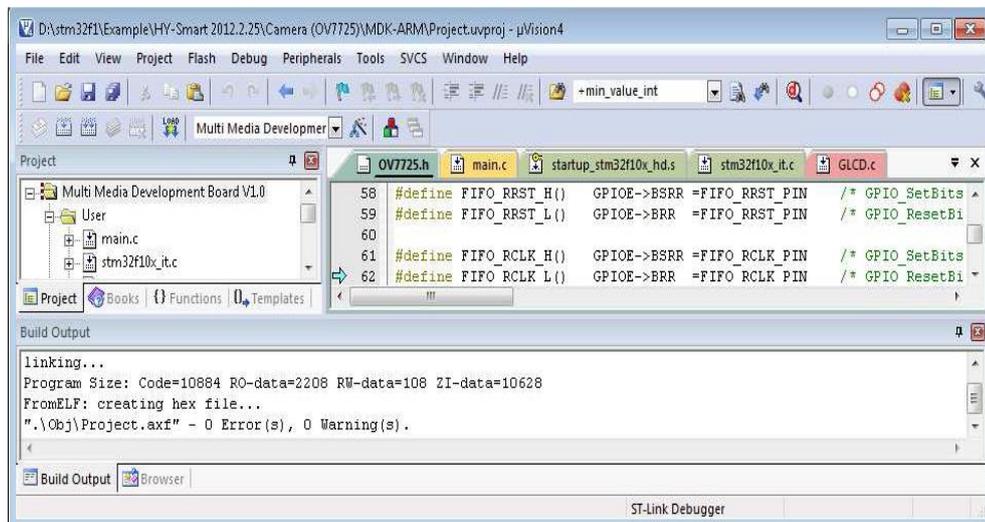


Figura 28.- Entorno Integrado de Desarrollo uVision5.
Fuente: uVision4.

Para el desarrollo de la aplicación ha sido necesario el uso de los siguientes programas: uVision Kiel IDE, ST-Link Programmer/Debugger, HyperTerminal y Matlab.

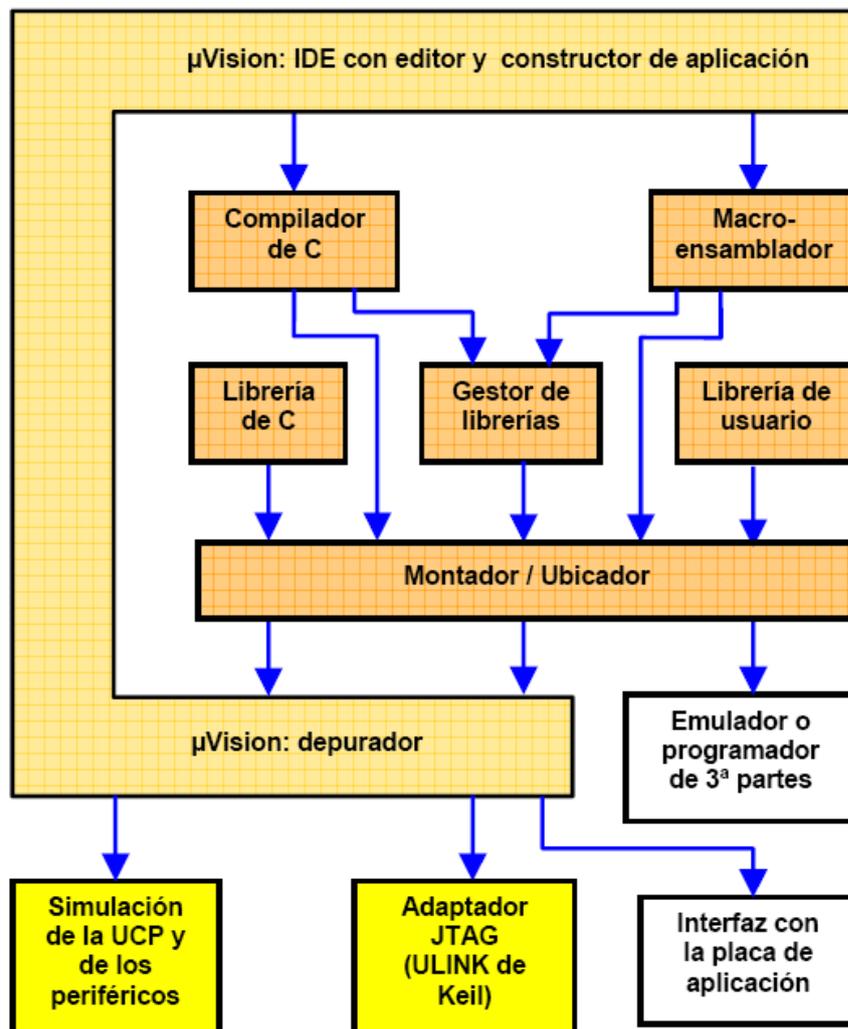


Figura 29.- Componentes del entorno de desarrollo Keil µVision.
Fuente: Eliseo García – UAH.

La interfaz del entorno IDE permite definir las características principales del proyecto como: tipo de procesador utilizado, tipo de optimización en el proceso de compilación, tipos de ficheros al ensamblar (o compilar) y montar. µVision permite a su vez, depurar el código desarrollado. Para esta función, dispone del modo depuración, mediante el cual se carga el código en la placa de desarrollo, ejecutándose el código en la misma y comunicándose con µVision. Para cumplir con este proceso, el PC y la placa de desarrollo se conectan mediante un cable de conversión USB a JTAG.

Los ficheros creados con el IDE de μ Vision se pasan al compilador o al macroensamblador para ser procesados y obtener los ficheros de tipo objeto reubicables, que luego pasan al módulo montador para conseguir el fichero ejecutable. Los ficheros ejecutables (.HEX) se utilizan, por ejemplo, para programar la memoria flash del microcontrolador.

El gestor de librerías permite la utilización de librerías de módulos-objeto previamente generados con el compilador. μ Vision también incluye un conjunto de librerías, que contienen por ejemplo ciertas funciones matemáticas.

1.9.1. Programación en C con KEIL μ Vision5

El manejo del entorno de desarrollo μ Vision de Keil, se presenta como una herramienta software para el desarrollo de proyectos en lenguaje C basados en microcontroladores. Entre otras funciones, permite compilar, simular, depurar y cargar el código en el microcontrolador STM32F429IG.

Se puede introducir a este entorno de desarrollo a través de un ejemplo sencillo que describe la creación de un proyecto, su simulación y depuración para visualizar variables, así como el empleo de los breakpoints.

Creación de un proyecto en C.

Para la creación de un proyecto en lenguaje C, es fundamental conocer, como se muestra en la Figura 30, la esquematización de manera general que corresponde a los pasos que se deben seguir en el diseño de un programa para resolver un determinado problema con un microcontrolador.

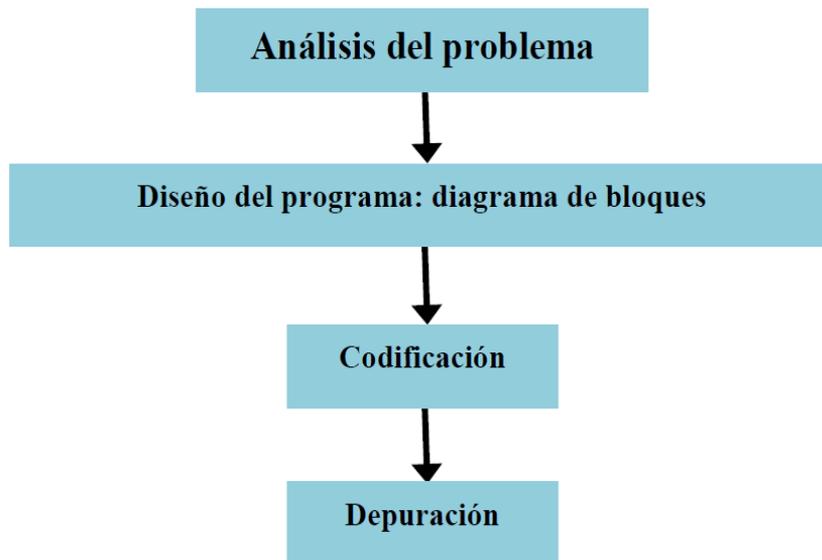


Figura 30.- Pasos en el diseño de un programa para microcontrolador.
Fuente: Eliseo García – UAH.

A continuación, se indica cómo crear un proyecto en μ Vision con el programa fuente que previamente se ha escrito con un editor de texto o desde el propio editor de μ Vision. Luego se detalla el proceso de compilación del programa y su simulación para proseguir con su depuración.

La creación de un nuevo proyecto, se lo realiza siguiendo los pasos indicados en la Figura 30. A continuación se abre una ventana donde se escribe el nombre del proyecto (New Project) y se guarda como se muestra en la Figura 31. Posteriormente se abre otra ventana en la que se selecciona el microcontrolador que se vaya a utilizar, en este caso sería el STM32F429IG como se grafica en la Figura 32. Como requisito se selecciona el fabricante STMicroelectronics y se hace clic en el símbolo '+' a la izquierda, con el objetivo de que aparezcan todos los modelos que corresponden a ese fabricante.

En la siguiente ventana que se abre (mostrada en la Figura 33, se pregunta si se desea copiar al proyecto un fichero con una plantilla para iniciar el código que se vaya a escribir, confirmando esta petición al realizar clic en el botón SI.

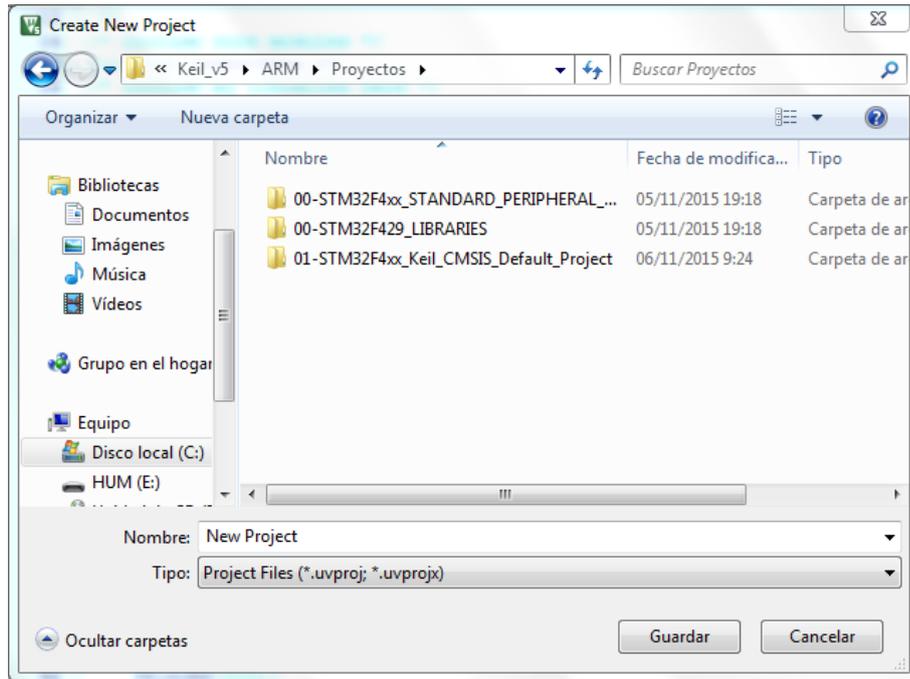


Figura 31.- Creación de un nuevo proyecto.
Fuente: Autores.

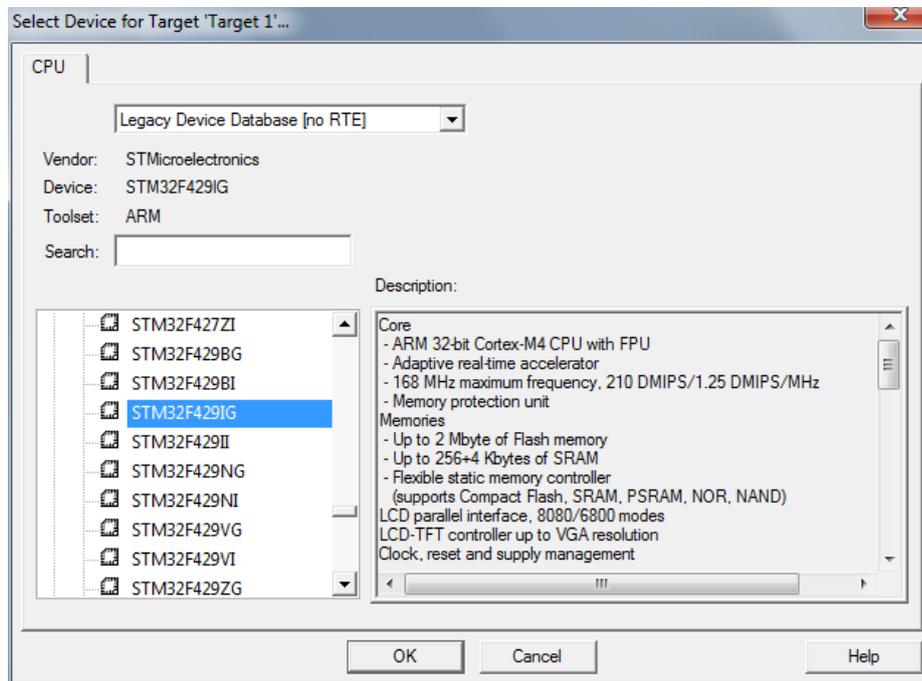


Figura 32.- Selección microcontrolador STM32F429IG de la Familia STMicroelectronics.
Fuente: Autores.

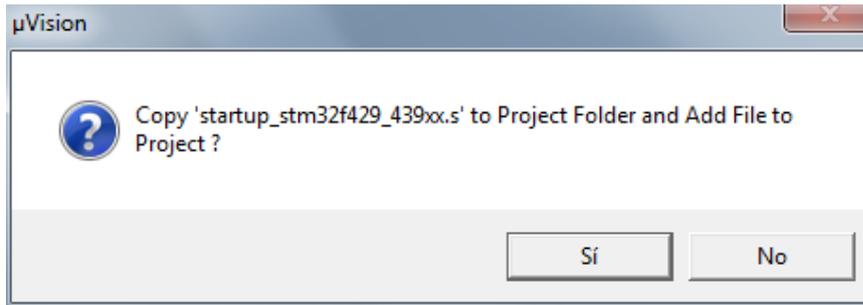


Figura 33.- Copia de archivo “startup_stm32f429_439xxx.s” en carpeta del proyecto.
Fuente: Autores.

El fichero startup_stm32f429_439xxx.s realiza una configuración inicial del microcontrolador, que permite al usuario trabajar con un nivel mayor de abstracción. Una vez realizada esta configuración, se llama a la función main (), que será la función principal del programa en C.

Agregar ficheros al proyecto

Se debe renombrar la entrada del árbol de proyecto Source Group 1 como Startup. Además de contener esta entrada de proyecto al fichero startup_stm32f4xx.s, es importante incluir otro fichero llamado system_stm32f4xx.c, necesario para la configuración inicial del microcontrolador.

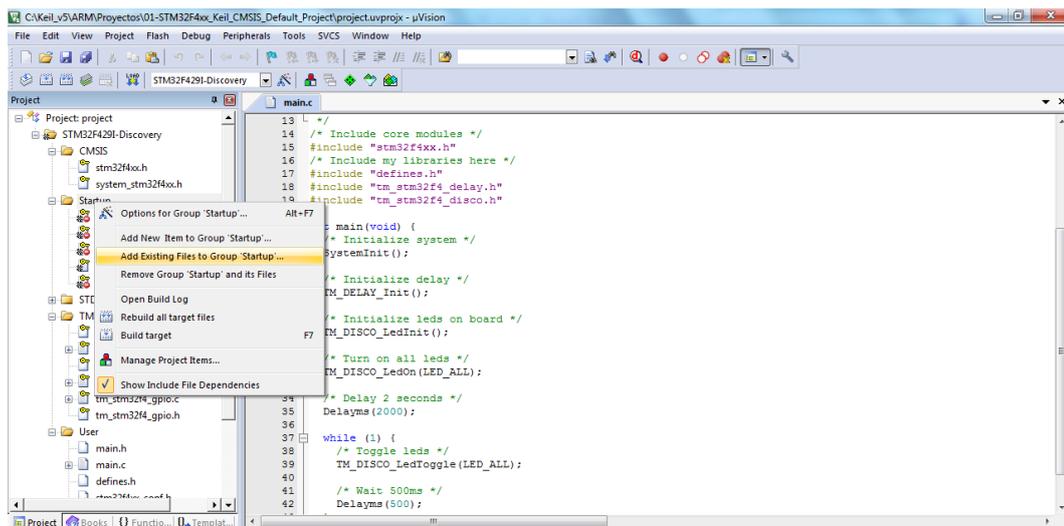


Figura 34.- Agregar archivos en Source Group.
Fuente: Autores.

1.9.2. Depuración de programas

La depuración de programas es de gran utilidad, se puede detener el programa en un punto específico, realizar la ejecución paso a paso, e incluso se puede observar el valor de distintas direcciones de memoria en tiempo real. En el caso específico de los microcontroladores fabricados por ST se tiene una herramienta llamada ST-LINK/v2 la cual sirve tanto para programar como para depurar programas y es compatible con el software KEIL MDK.ARM.

CAPÍTULO II

2. METODOLOGÍA

2.1. Tipo de estudio

2.1.1. Descriptivo

Este proyecto recurre al tipo de estudio descriptivo, debido a que implica observar y describir el comportamiento del fenómeno de propagación de una onda de luz, así como también de cada uno de los componentes electrónicos que conforman el sistema. Finalmente se puede conocer los detalles de funcionamiento del sistema completo por medio de la descripción y análisis.

2.2. Métodos, Técnicas e Instrumentos

2.2.1. Métodos

2.2.1.1. Analítico/Deductivo

Se utilizó el método analítico-deductivo en el proyecto por la razón de que se inicia con un análisis particular del funcionamiento de los componentes necesarios para el desarrollo del sistema y la forma de interactuar entre sí para que el sistema desempeñe su función de la manera más óptima.

2.2.2. Técnicas

2.2.2.1. Observación

Esta técnica consiste en la recolección de información que sea de apoyo para el desarrollo del proyecto, dando las pautas necesarias para el diseño e implementación de sistema graficador 3D en movimiento.

2.2.3. Instrumentos

Los instrumentos necesarios son libros, folletos, archivos, páginas web, blogs, cursos, datasheet, que son útiles para el diseño e implementación.

2.3. Población y muestra

La población corresponde al total de puntos en los que se obtendrá las mediciones de la distancia en los tres ambientes en los que se va a realizar las pruebas del sistema graficador en 3D, pudiendo identificar las mediciones de distancia entre lo real y lo adquirido. Estos ambientes serán los siguientes:

- Ambiente plano
- Ambiente con gradas
- Ambiente con obstáculos

Para determinar el tamaño de la muestra se aplica la siguiente ecuación que necesita datos, como lo es el tamaño de la población N. Para calcular el tamaño de la muestra se utiliza la siguiente fórmula:

$$n = \frac{PQ \times N}{(N - 1) \times \frac{E^2}{K^2} + PQ}$$

$$n = \frac{(0.25) \times (1920)}{(1920 - 1) \times \frac{(0.10)^2}{(2)^2} + 0.25}$$

$$n = 95$$

Dónde:

n = Tamaño de la muestra

N = Tamaño de la población = 1920

PQ = Constante de la varianza población (0.25)

E = Error máximo admisible (al 1% = 0.01; al 2% = 0.02; al 5% = 0.05; al 10% = 0.10) a mayor error probable, menor tamaño de la muestra

K = Coeficiente de corrección del error k=2

Al aplicar esta fórmula se obtiene el número de puntos que conforman la muestra, dando como resultado 95 muestras, con las cuales se deben trabajar en cada ambiente.

2.4. Hipótesis

El diseño de un sistema graficador 3D permitirá la representación gráfica de una superficie con una precisión mayor al 90%.

2.5. Operacionalización de variables

VARIABLES	CONCEPTO	DIMENSIONES	INDICADORES	INSTRUMENTOS
Independiente: Sistema graficador 3D	Dispositivo, aparato o mecanismo que desarrolla determinadas acciones. Su nombre está vinculado a que dicho artefacto está dispuesto para cumplir con su objetivo.	Hardware	Determinación de distancia Procesamiento y envío de datos	Sensores láser Tarjeta STM32F429IG
		Software	Herramienta de software matemático	MATLAB
Dependiente: Ubicación y distancia	La distancia es el trayecto espacial o el periodo temporal que separa dos acontecimientos o cosas. Se trata de la proximidad o lejanía que existe entre objetos o eventos.	Coordenadas	Espacio	Recolección de datos : Mediciones de distancia
			Trayecto espacial	

Tabla 7.- Operacionalización de variables.
Elaborado: Autores.

2.6. Procedimientos

Para el desarrollo de este sistema graficador 3D, se requirió seguir una serie de procesos, que en conjunto, cumplirían con el objetivo planteado. Cada uno desempeña una función específica, y sirviendo así de apoyo para la siguiente etapa, haciendo que el sistema implementado sea funcional.

- Lectura datos de distancia del sensor láser.

- Sistema mecánico de control de ángulo PAN&TILT controlador por el microcontrolador.
- Conversión de datos desde el sensor láser hasta el microcontrolador.
- Interpretación datos de distancia y ubicación por microcontrolador.
- Envío de datos desde el microcontrolador de una forma adecuada para su interpretación por el software MATLAB.
- Graficación en 3D en MATLAB.

Diagrama de conexiones

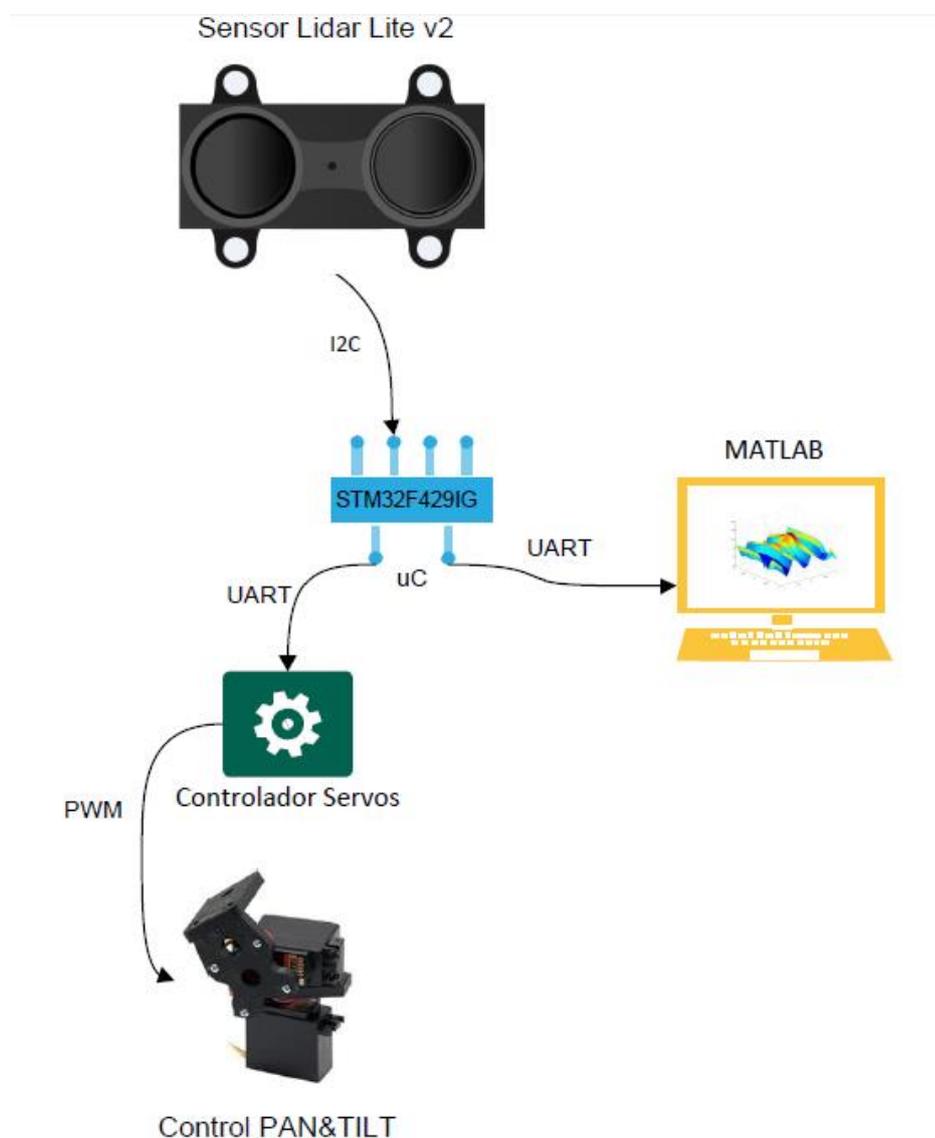


Figura 35.- Diagrama de conexiones sistema graficador 3D.

Fuente: Autores.

2.7. Procedimiento y análisis

2.7.1. Diseño

2.7.1.1. Control PAN&TITL

El sistema mecánico donde se ubica el sensor láser, con el fin de realizar el apuntamiento respectivo, es controlado por dos servomotores (Ver Figura 36), que tiene sus correspondientes grados de libertad para realizar los movimientos de rotación y guiñada (PAN & TILT). Para el control de los servomotores se utilizara un controlador de servos de 6 canales (Ver Figura 37), que será comandado vía UART desde la placa STM32F429IG. Este controla los servomotores HS311 ubicados en la parte superior e inferior de la plataforma de control.



Figura 36.- Estructura para Control PAN y TILT.
Fuente: Servocity.



Figura 37.- Controlador de servos de 6 canales.
Fuente: Pololu.

La medición de las distancias por medio de un sistema mecánico se puede visualizar a través de una representación 3D como se observa en la Figura 38. Mediante la exploración de un área donde se implemente el sistema, se incorpora el sistema mecánico con un arreglo de servomotores que permita hacer un barrido de PAN de 60 y un barrido de TILT de 32° del área a explorar.

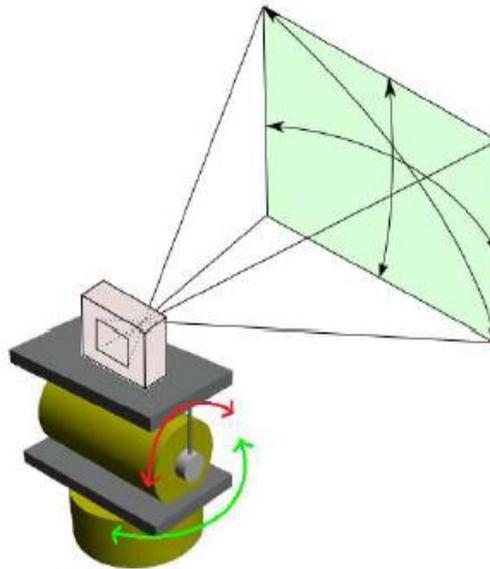


Figura 38.- Representación del sistema mecánico que sostiene el sensor láser.
Fuente: Control PTZ de un Sistema Electro Óptico - EPN.

Con el movimiento de los servomotores varía la posición angular de los mismos, a su vez los obstáculos vistos en pantalla con respecto al sensor láser. En la Figura 38 se representa a los servomotores en color amarillo, las flechas rojas indican como el sensor láser varían en posición vertical, y las rojas en posición horizontal.

2.7.1.2. Descripción de requerimientos de precisión del sistema

El sistema de adquisición de datos del graficador parte del plano de observación, que cubre 60 grados en sentido horizontal y 32 grados en sentido vertical; la división del plano de observación se ha diseñado a partir del número de sensores láser utilizados, siendo implementado un sólo dispositivo de medición. El sensor ha sido focalizado hacia el centro del plano de observación y realiza un envío de datos cada 8 filas que constituyen un bloque o conjunto de filas, como se puede

ver en la Figura 39, siendo de esta forma que el plano de observación está constituido por cuatro bloques. Finalmente el plano de observación está representado en su totalidad por 32 grados verticales y 60 grados horizontales.

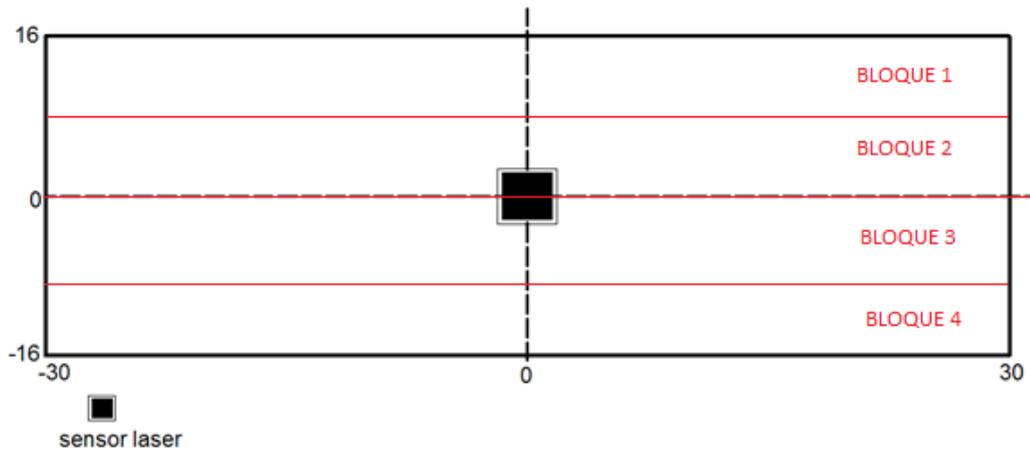


Figura 39.- División de cuadrantes y ubicación del sensor láser.

Fuente: Autores.

Al disponer el plano de observación con 60 grados horizontales y 32 grados verticales, es importante determinar el número de divisiones adecuadas para realizar la medición con el sensor láser, de esta manera se ha considerado proceder a fragmentar el plano horizontal y el plano vertical cada grado, obteniendo un total de 60 divisiones horizontales y 32 divisiones verticales respectivamente, representado en la Figura 40 y determinado mediante las ecuaciones siguientes:

$$\begin{aligned} \# \text{Divisiones horizontales} &= 60^{\circ} / 1^{\circ} \\ \# \text{Divisiones horizontales} &= 60 \text{ divisiones} \end{aligned} \quad (3)$$

$$\begin{aligned} \# \text{Divisiones verticales} &= 32^{\circ} / 1^{\circ} \\ \# \text{Divisiones verticales} &= 32 \text{ divisiones} \end{aligned} \quad (4)$$

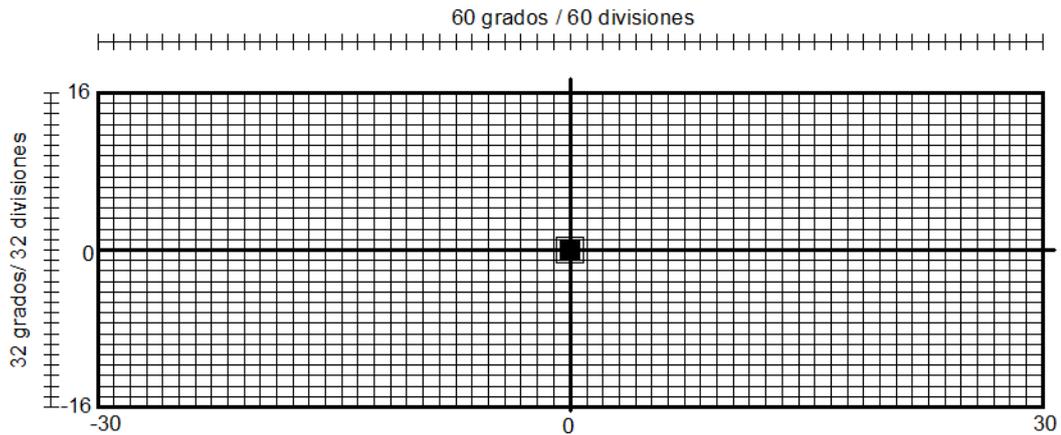


Figura 40.- Divisiones y ángulos del sistema.
Fuente: Autores.

Una vez obtenidas las divisiones en el plano de observación se procedió a determinar la ubicación de los puntos correspondientes en los que los sensores muestrean los datos de distancia y cubren el plano de observación, como se aprecia en la Figura 41.

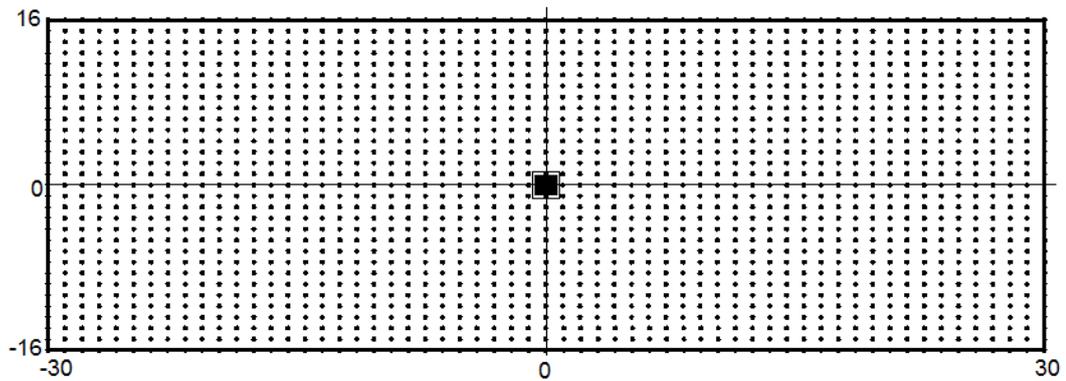


Figura 41.- Ubicación de puntos a detectar en el sistema.
Fuente: Autores.

El sistema al emplear una detección de obstáculos por puntos debe realizar una aproximación de los contornos que se analizan. En la Figura 42 se puede observar el perfil de un contorno de prueba, que presenta irregularidades al contemplar objetos de alturas variables, mientras que en la Figura 43 se sobrepone dicho perfil en el sistema de detección ya dividido y que contiene el plano de observación.

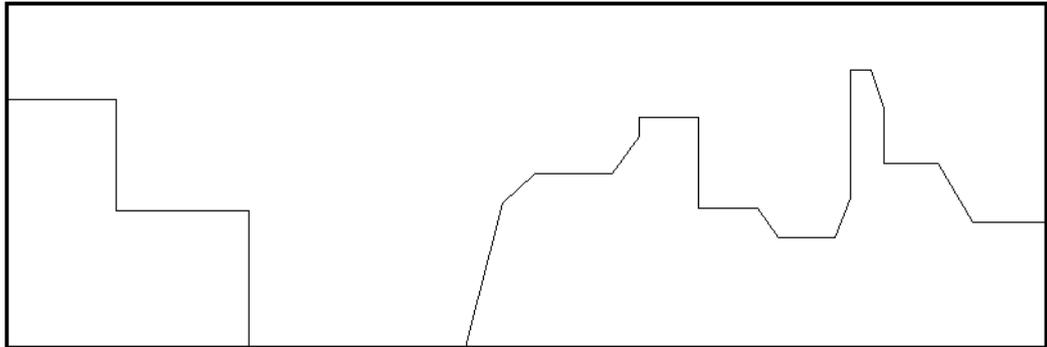


Figura 42.- Entorno de prueba.
Fuente: Autores.

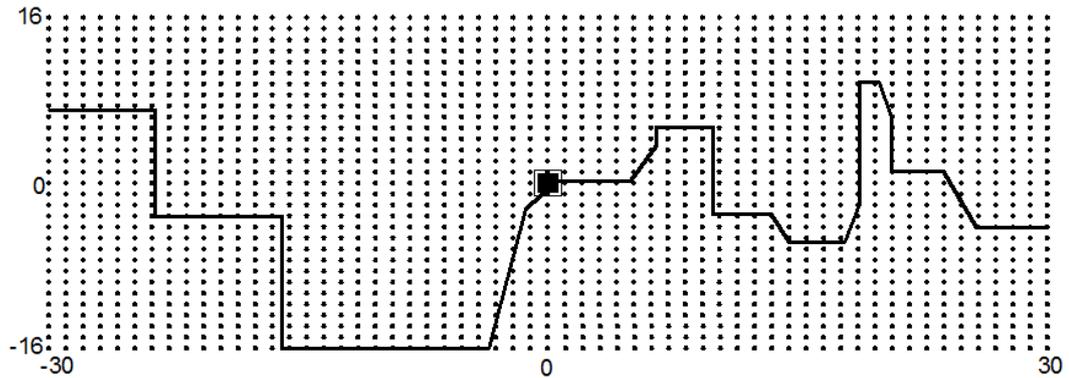


Figura 43.- Entorno de prueba sobrepuesto en el sistema.
Fuente: Autores.

Finalmente, los perfiles de los objetos que coinciden con los puntos del sistema de detección son procesados para graficarse, mientras que las partes que no son reconocidas al no coincidir con los puntos, son descartadas; se puede visualizar en la Figura 44 como se realiza la detección de los objetos que coinciden con los puntos, al mismo tiempo que se ilustra el contorno real de una superficie y el contorno reconocido por el sistema graficador.

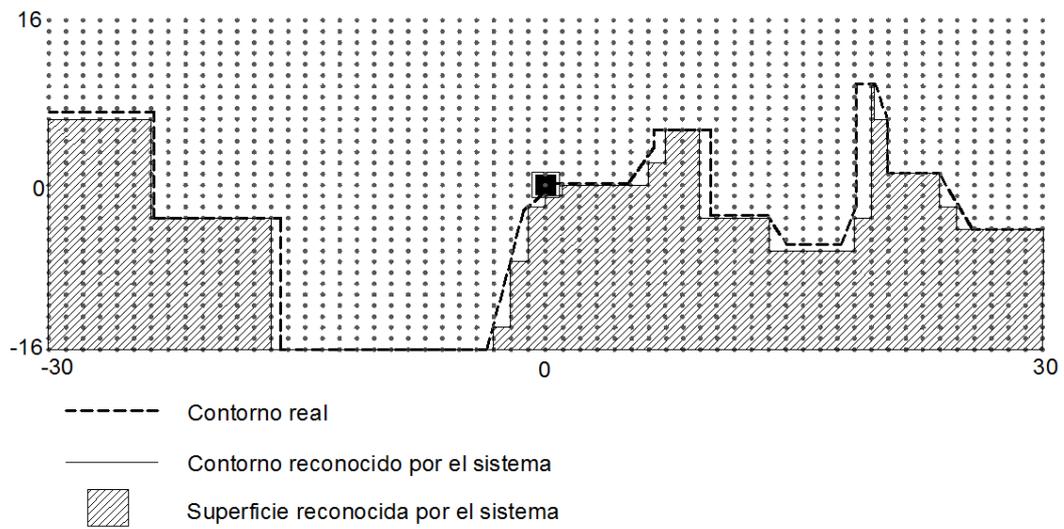


Figura 44.- Reconocimiento del sistema de un entorno de prueba.
Fuente: Autores.

2.7.1.3. Resolución y precisión del sistema graficador 3D

Resolución del sistema

Barrido Horizontal

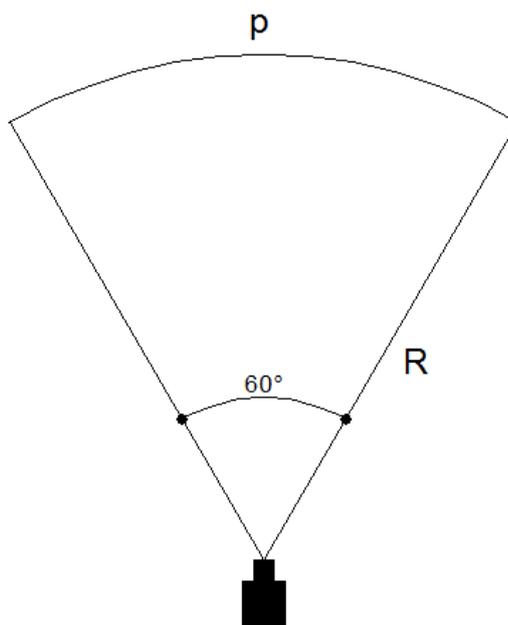


Figura 45.- Reconocimiento de barrido horizontal.
Fuente: Autores.

$$p = \text{radianes} \times R$$

$$p = \left(\frac{60^\circ \times \pi}{180^\circ} \right) \times R \quad (5)$$

Barrido Vertical

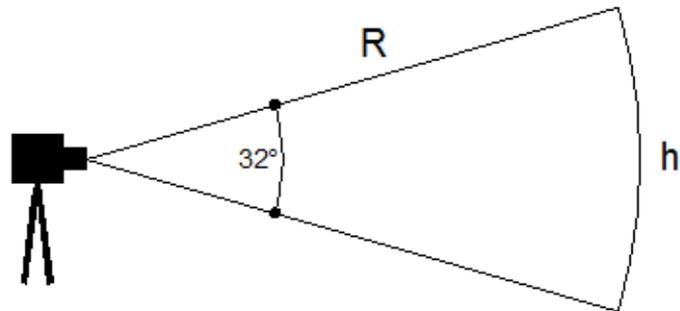


Figura 46.- Reconocimiento de barrido vertical.

Fuente: Autores.

$$h = \text{radianes} \times R$$

$$h = \left(\frac{32^\circ \times \pi}{180^\circ} \right) \times R \quad (6)$$

Área cubierta por el sistema.

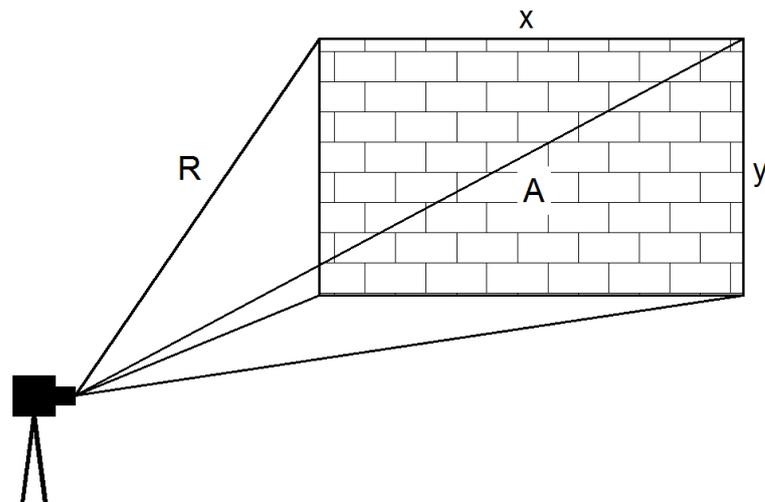


Figura 47.- Reconocimiento del área cubierta por el sistema.

Fuente: Autores.

R (m)	Ancho (m)	Altura (m)	Área (m ²)
1	1,05	0,56	0,58
2	2,09	1,12	2,34
3	3,14	1,68	5,26
4	4,19	2,23	9,36
5	5,24	2,79	14,62
6	6,28	3,35	21,06
7	7,33	3,91	28,66
8	8,38	4,47	37,43
9	9,42	5,03	47,37
10	10,47	5,59	58,49

Tabla 8.- Área cubierta por el sistema graficador 3D.
Fuente: Autores.

Pixeles

$$x = 60 \text{ muestras} = 60 \text{ pixeles}$$

$$y = 32 \text{ muestras} = 32 \text{ pixeles}$$

$$\text{Resolución Pixeles} = x \cdot y \quad (7)$$

$$\text{Resolución Pixeles} = 60 \times 32$$

$$\text{Resolución Pixeles} = 1920 \text{ pixeles}$$

Precisión del sistema

La precisión del láser para la aplicación es más que suficiente. Si transformamos la precisión angular en precisión lineal según la ecuación, como se puede apreciar en la Figura 48 y los resultados en la Tabla 9, vemos que para hacer mapas una sala grande, la precisión lineal a 5 metros es de unos 8,73 cm. Con esta precisión podremos mapear paredes, techos, puertas, ventanas e incluso mesas, sillas, y personas.

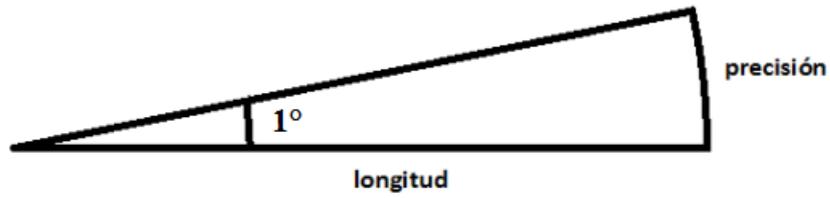


Figura 48.- Precisión Lineal Sistema.
Fuente: Autores

$$\text{Precisión} = \text{radianes} \times R$$

$$\text{Precisión} = \frac{(2\pi \times 1^\circ)}{360} \times R \quad (8)$$

R (m)	Precisión Lineal (cm)
1	1,75
1,5	2,62
2	3,49
2,5	4,36
3	5,24
3,5	6,11
4	6,98
4,5	7,85
5	8,73
5,5	9,60
6	10,47
6,5	11,34
7	12,22
7,5	13,09
8	13,96
8,5	14,84
9	15,71
9,5	16,58
10	17,45

Tabla 9.- Precisión del sistema graficador 3D.
Fuente: Autores.

2.7.1.4. Esquema electrónico del sistema

El sistema de graficación 3D está constituido principalmente por cinco bloques funcionales, interconectados entre sí, como se muestra en la Figura 49. Los bloques del sistema son los siguientes:

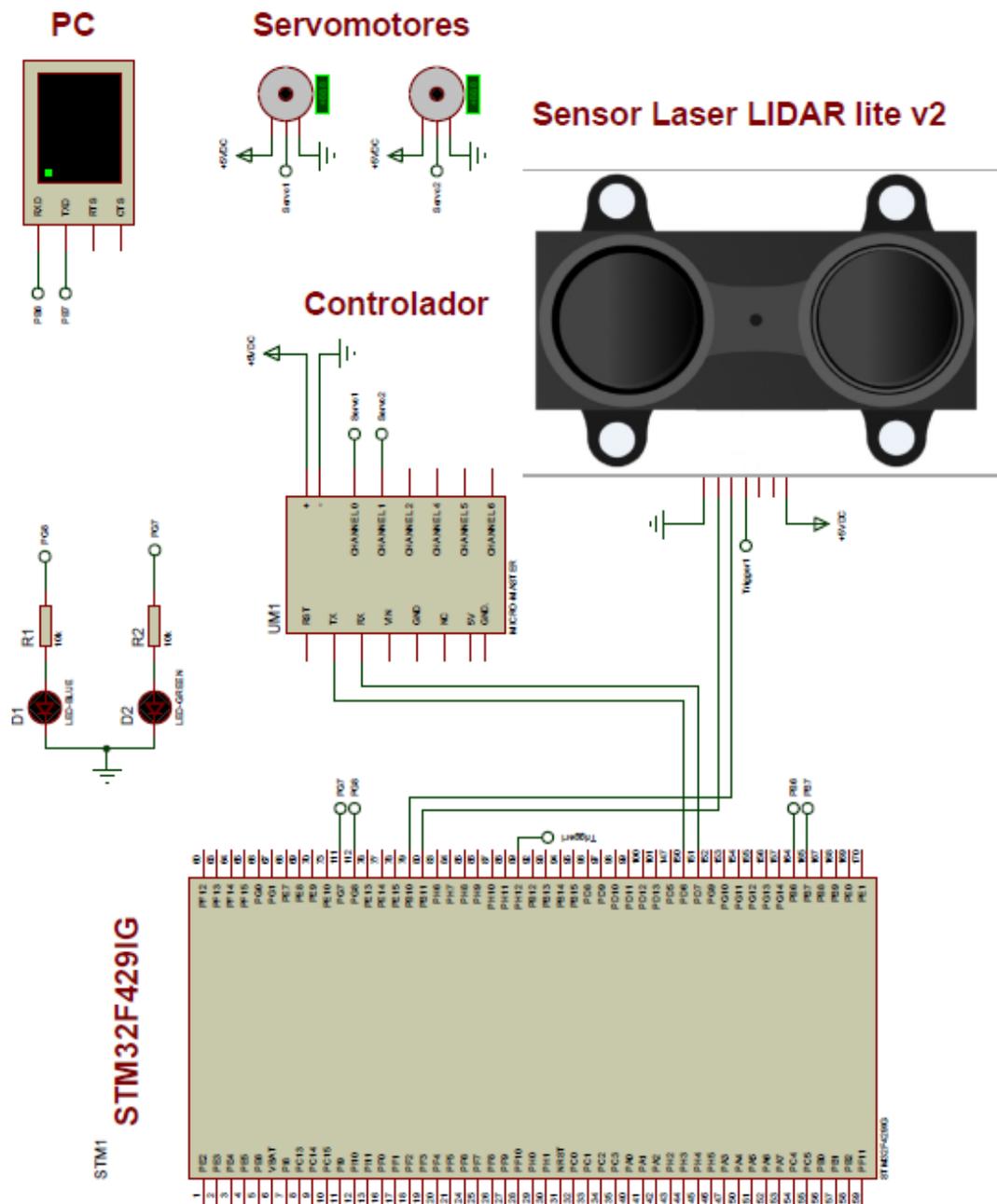


Figura 49.- Diagrama esquemático-electrónico del sistema.
Fuente: Autores.

2.7.1.5. Descripciones del código software

El microcontrolador STM32F429I ha sido programado de tal manera que se orienta a ejecutar los procesos necesarios para poder realizar funciones como: interfaz I2C, procesamiento simultáneo de datos y envío de información, control de servos y mecanismos PAN-TILT.

Todos estos procesos son realizados de acuerdo a un orden preestablecido, con fines de optimización para aprovechar las ventajas del microcontrolador y demás dispositivos incorporados en el sistema de graficación. A continuación se estructura el proceso de operación a seguir, como se aprecia en la Figura 50:

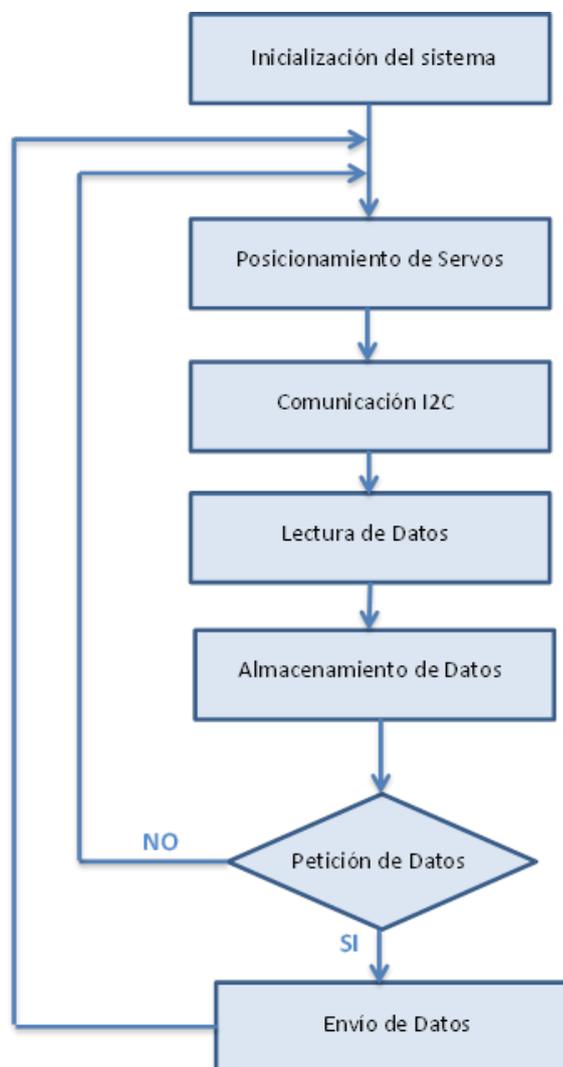


Figura 50.- Diagrama funcional del proceso ejecutado en el microcontrolador.

Fuente: Autores.

Inicialización del Sistema

El sistema debe ser inicializado a través de la declaración de librerías y variables en el microcontrolador SMT32F429I, es decir que se definen los parámetros iniciales que comprende también la puntualización de las entradas y salidas (GPIOs).

Es importante al momento de empezar a trabajar con un microcontrolador STM32, incluir la librería que contiene la declaración específica de registros de periféricos, bits definidos y memorias; para este proceso se utiliza la instrucción:

```
#include "stm32f4xx.h"
```

De esta forma se define el uso dirigido a periféricos que pertenecen a la familia STM32F4. Las demás librerías son declaradas conjuntamente con la anterior librería y las instrucciones correspondientes son:

```
#include "tm_stm32f4_usart.h"  
#include "tm_stm32f4_delay.h"  
#include "tm_stm32f4_i2c.h"
```

Se utilizan las librerías relacionadas al módulo de transmisión y recepción serial, la generación de pausas en el sistema y el uso del módulo I2C respectivamente.

La declaración de variables se lo realiza dentro del programa principal o función principal dada por la estructura:

```
int main (void) {
```

El sistema requiere de algunos tipos de variables que cumplen una función específica y que se adaptan de acuerdo al proceso a ejecutarse, por lo que se incorporan variables de tipo entero sin signo estándar, variando solamente su longitud y en algunos casos se las modifica para obtener una matriz de datos. La declaración según su funcionalidad y condiciones iniciales viene dado por las instrucciones:

```
uint16_t matriz[8][60];
```

```
int i,j;
int i1=1;
uint16_t dato_i2c;
int filas=0, columnas=0;
uint8_t datorecibido;
uint8_t numero[2];
```

El otro proceso que se debe cumplir es la inicialización tanto del sistema del microcontrolador como de los otros módulos a utilizarse, definiendo los parámetros que comprende cada uno de ellos de manera individual, como se indica en las líneas de programación siguientes:

```
/* Inicialización del sistema*/
    SystemInit();
/* Inicializar Delay Library */
    TM_DELAY_Init();
/* Inicializar USART1, 57600baud, TX:PB6, RX:PB7 */
    TM_USART_Init(USART1, TM_USART_PinsPack_2, 57600);
/* Inicializar I2C, SCL: PB10 and SDA: PB11 with 100kHz serial
clock*/
    TM_I2C_Init(I2C2, TM_I2C_PinsPack_1, 100000);
```

Posicionamiento de Servos

El posicionamiento de los servos se ejecuta por el movimiento sincronizado proveniente de las señales que emite el microcontrolador STM32F429I hacia el controlador de servos, encargado de convertir las instrucciones seriales a pulsos PWM para el funcionamiento de los servos. Al iniciar el proceso de adquisición de datos, los servos parten de una posición de origen para empezar el barrido y luego realizan un incremento angular. La instrucción que se envía a cada servo es representada por:

```
TM_USART_Send(USART1, servo, 5);
```

En donde la variable servo está compuesta por 5 bytes de información correspondientes a la siguiente estructura:

Byte 1= 0xFF, comando para establecer el baud rate del protocolo.

Byte 2= 0x84, comando para configurar los datos de posición del servo.

Byte 3= especifica el número de canal.

Bytes 4 y 5= contienen la información de la señal PWM a generar. Esta información está dividida en dos bytes correspondientes a los bits más significativos y menos significativos respectivamente.

Lectura de Datos

El sensor láser permite la adquisición de los datos de distancia. Debido a que la frecuencia de operación del sensor LIDAR lite v2 de 500Hz (2ms) es necesario también incluir un retardo mayor o igual a este tiempo. Además se sincroniza la adquisición del valor de la distancia mediante un pin indicador de disponibilidad de datos, entregado por el sensor láser.

Las señales generadas para las funciones de salida y Leds indicadores se ejecutan por medio de una declaración de pines de salida (GPIOs), en donde se definen los pines a usarse y las características de funcionamiento como:

Declaración de la función de GPIO:

```
GPIO_InitTypeDef GPIO_InitDef;
```

Habilitar el reloj de los pines:

```
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOG, ENABLE);
```

Declaración de pines a utilizar:

```
GPIO_InitDef.GPIO_Pin = GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 |  
GPIO_Pin_3;
```

Definir pines como salidas:

```
GPIO_InitDef.GPIO_Mode = GPIO_Mode_OUT;
```

Definir salida como push-pull:

```
GPIO_InitDef.GPIO_OType = GPIO_OType_PP;
```

Deshabilitar resistencias pull internas de los pines:

```
GPIO_InitDef.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

Definir la máxima velocidad de reloj de los GPIOs:

```
GPIO_InitDef.GPIO_Speed = GPIO_Speed_100MHz;
```

Inicializar pines

```
GPIO_Init(GPIOG, &GPIO_InitDef);
```

El código de programación que nos permite enviar las señales están constituidos por las estructuras siguientes:

```
GPIO_SetBits(GPIOG, GPIO_Pin_0 | GPIO_Pin_1 | GPIO_Pin_2 |  
GPIO_Pin_3);  
Delayms(50);
```

La declaración para la señal de disponibilidad de datos, se declara como entrada y se rige a la siguiente línea de programación:

```
TM_GPIO_Init(GPIOH, GPIO_Pin_12, TM_GPIO_Mode_IN, TM_GPIO_OType_PP,  
TM_GPIO_PuPd_NOPULL, 100);
```

Comunicación I2C

Para iniciar una adquisición de datos por parte del sensor Lidar Lite v2., se utiliza la interfaz de comunicación I2C, siendo necesario configurar los parámetros iniciales tanto del dispositivo maestro como esclavo.

El primer parámetro a configurar es la inicialización del sensor, mediante la instrucción de escritura I2C, declarando el número de periférico I2C a utilizar, la dirección del dispositivo a comunicar, el registro a usar y finalmente el valor a escribir en el registro, de la siguiente manera:

```

/*****Configuracion Sensor*****/
TM_I2C_Write(I2C2, 0xc4, 0x00, 0x00);/***** Inicio Sensor*****/

```

El sensor láser es configurado en modo continuo para obtener las lecturas de datos de manera ininterrumpida; se debe escribir en el registro 0x45, el valor de la velocidad entre cada medición, de la siguiente forma:

```

TM_I2C_Write(I2C2, 0xc4, 0x45, 0x14);/*****Velocidad*****/

```

El valor de la velocidad seleccionada, corresponde a la configuración de la Tabla 10 a continuación:

Periodo de Medición (ms)	Velocidad (m/s)	Valor a cargar en el registro 0x45
100	0.10 m/s	0xC8
40	0.25 m/s	0x50
20	0.50 m/s	0X28
10	1.00 m/s	0x14

Tabla 10.- Configuración de velocidad del sensor Lidar Lite v2.
Fuente: pulsedlight3d.

Para habilitar el pin de disponibilidad de datos, que cambia de estado alto a bajo cuando una nueva medición está lista, se escribe el valor 0x20 en el registro 0x04:

```

TM_I2C_Write(I2C2, 0xc4, 0x04, 0x20);/*****Pin Mode *****/

```

El número de lecturas a ejecutarse, es configurado por medio del registro 0x11, siendo los valores establecidos por la instrucción de escritura y según la Tabla 11, mostrados a continuación:

```
TM_I2C_Write(I2C2, 0xc4, 0x11, 0xFF);/*****Numero de Lecturas*****/
```

Número de Lecturas	Valor a cargar en el registro 0x11
1	0x01
100	0x64
254	0xFE
Continuas	0xFF

Tabla 11.- Configuración de número de lecturas del sensor Lidar Lite v2.
Fuente: pulsedlight3d.

Para iniciar el proceso de adquisición de lecturas se ejecuta la siguiente instrucción, escribiendo el valor 0x04 al registro 0x00:

```
TM_I2C_Write(I2C2, 0xc4, 0x00, 0x04);/*****Inicio de Lectura*****/
```

Almacenamiento de Datos

El proceso para almacenar los valores de distancia en una matriz de datos inicia cuando está disponible una nueva medición indicado por el sensor y se procede a la lectura del registro 0x8F, que contiene el valor de la distancia, repartido en dos bytes, correspondientes a un registro alto y bajo, por tal motivo se realiza una reconstrucción del valor de distancia para ser almacenado en un solo dato y se ejecuta conforme al diagrama de flujo de la Figura 51:

```
/*Leer Dato de Laser*/
while (GPIO_ReadInputDataBit(GPIOH, GPIO_Pin_12)){
    /*espera hasta encontrar nueva medición*/
}
}
```

```

TM_I2C_ReadMulti(I2C2, 0xc4, 0x8f, distancia_sensor, 2 );
    Delays(1);

distancia_final=(distancia_sensor[0]<<8)+distancia_sensor[1];
matriz[columna_primaria][cont1]=distancia_final;

```

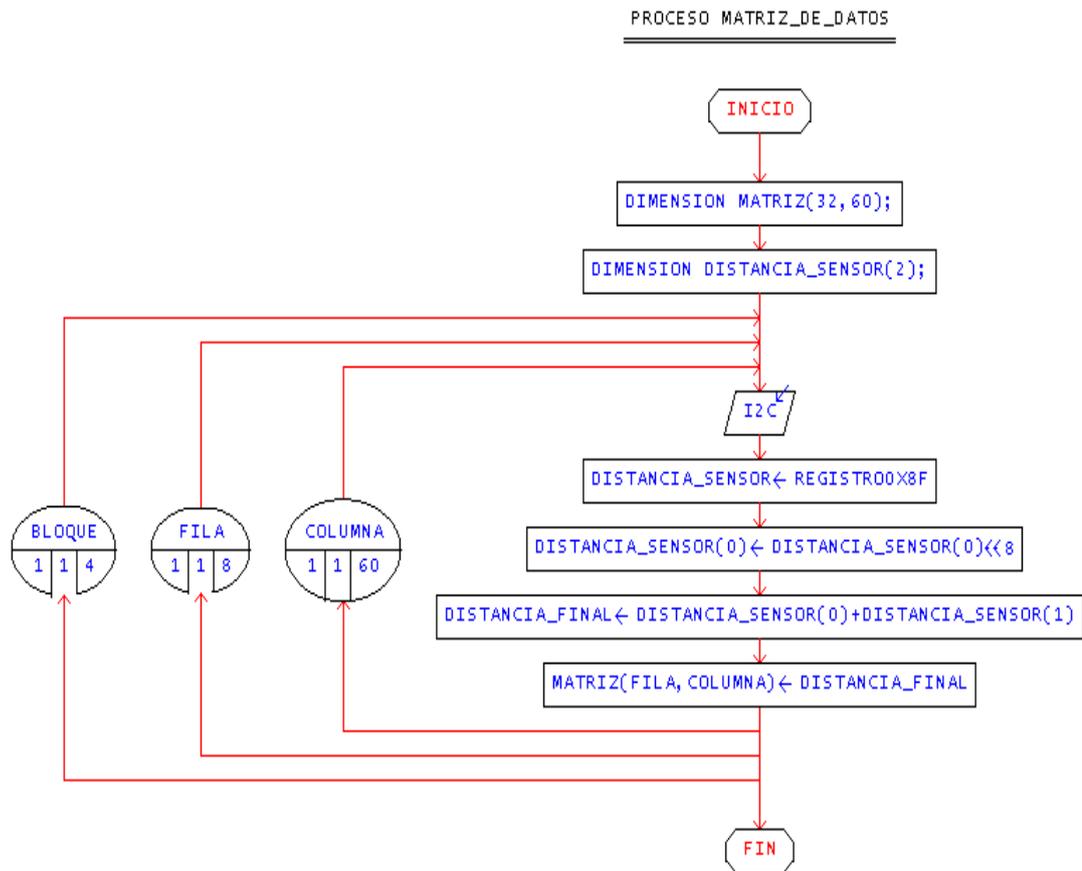


Figura 51- Diagrama de flujo del almacenamiento de datos en la matriz.
Fuente: Autores.

Envío de Datos

El envío de los datos almacenados se genera de manera serial mediante una solicitud de transmisión proveniente de la PC. Se verifica que el dato recibido sea correcto para empezar el proceso. La instrucción que recoge la recepción de la solicitud corresponde a:

```

datorecibido = TM_USART_Getc(USART1);

```

El proceso para identificar el bloque de datos o conjunto de filas correspondiente para ser enviado, se realiza mediante la transmisión de una cadena de caracteres que contienen el orden para ser reconocidos por la interfaz de recepción:

```
if(bloque==2){
    TM_USART_Send(USART1,solicitud,1);
    TM_USART_Send(USART1,solicitud3,1);
    TM_USART_Send(USART1,solicitud3,1);
}
/**El valor de la solicitud depende del número de bloque***/
```

En lo que respecta al envío de la información de distancias, es importante realizar un desplazamiento de bits, con el propósito de obtener de manera separada los bits más significativos y menos significativos, debido a que la transmisión serial sólo permite el envío de 8 bits por trama y los datos almacenados tienen una dimensión de 12 bits. La instrucción USART tiene la opción de definir el número de bytes a transmitir.

```
numero[1]=(matriz[filas][columnas]>>8) & 0x00FF;
numero[0]=matriz[filas][columnas] & 0x00FF;
TM_USART_Send(USART1,numero,2);
```

2.7.2. Visualización en 3D utilizando MATLAB

La visualización de los datos enviados por el microcontrolador se efectúa por medio del software MATLAB. La recepción de datos es de forma serial, por esta razón se deben declarar las instrucciones para habilitar el puerto serial con sus respectivas configuraciones de parámetros de transmisión y recepción:

```
% declarar el puerto serial
PS=serial('COM5');
% configuración de la velocidad a 57600 Baudios
set(PS,'Baudrate',57600);
% configuración bit de parada a uno
set(PS,'StopBits',1);
% configuración del tamaño del dato, 8 bits
```

```

set(PS, 'DataBits', 8);
% configuración sin paridad
set(PS, 'Parity', 'none');

% "n" es el número de bytes a enviar
set(PS, 'OutputBufferSize', 1);
% "n" es el número de bytes a recibir
set(PS, 'InputBufferSize' , 720);
% 5 segundos de tiempo de espera
set(PS, 'Timeout', 50);
% Abrir el puerto serial
fopen(PS);

```

Para realizar la petición de envío de datos hacia el microcontrolador STM32F429I, se transmite un carácter desde el puerto serial de la PC que representa el inicio de la recepción de datos en el programa:

```
fprintf(PS, '%c', '0');
```

Al recibir los datos seriales se los almacena directamente en una matriz de graficación, por lo que es importante precisar la posición correspondiente de cada valor de distancia a la posición dentro de la matriz. Se realiza una comparación de los valores recibidos en el puerto serial y se guardan las distancias de acuerdo al número de bloque respectivo. La instrucción de lectura serial de MATLAB es configurada para obtener directamente datos de 2 bytes y convertidos a formato decimal a través de la instrucción uint16:

```

read = fread(PS, 3, 'uint8');

if (read(1) == hex2dec('AA') && read(2) == hex2dec('B1') &&
read(3) == hex2dec('B1'))

    for i=1:8
        for j=1:60
            matriz(i,j) = fread(PS,1,'uint16');
        end
    end
end

```

```
end

if (read(1) == hex2dec('AA') && read(2) == hex2dec('B2') &&
read(3) == hex2dec('B2'))

    for i=9:16
        for j=1:60
            matriz(i,j) = fread(PS,1,'uint16');
        end
    end
end
end
```

La graficación como se puede ver en la Figura 52, una vez adquirido los datos de distancia y almacenados en la matriz, se realiza por la siguiente instrucción:

```
surf(matriz)
```

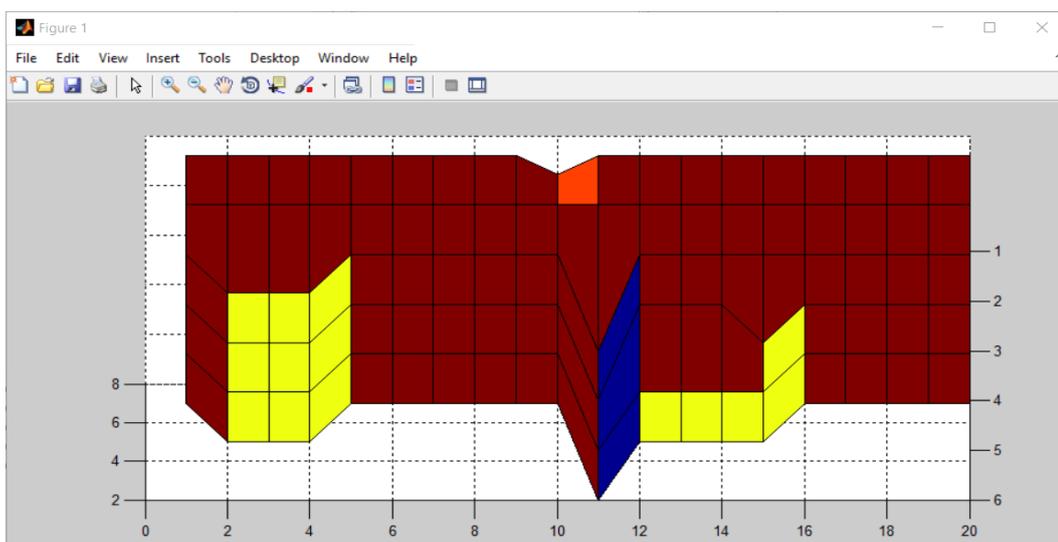


Figura 52.- Representación de la matriz de graficación.
Fuente: Autores.

Para obtener una visualización óptima con una interpolación de los datos representados como se aprecia en la Figura 53, se configura a través del siguiente comando:

```
shading interp
```

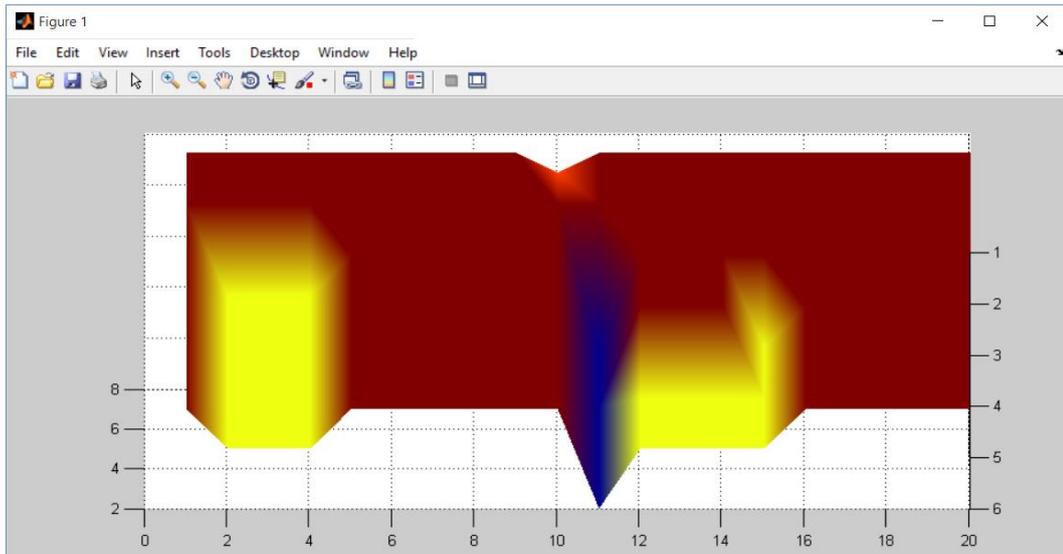


Figura 53.- Representación con interpolación de datos.
Fuente: Autores.

La configuración del color, se establece por medio de archivos guardados previamente con extensión MAT-File. La creación de dos archivos permite la visualización en dos diferentes matices como se observan en las Figuras 54 y 55:

```
set (figure(1), 'Colormap', blanconeuro);
```

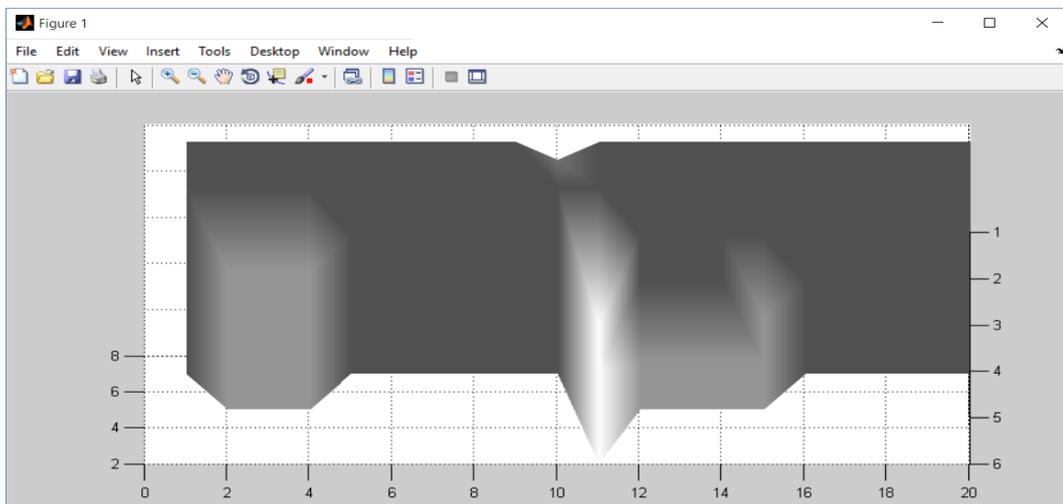


Figura 54.- Representación con mapa de colores blanco-negro.
Fuente: Autores.

```
set (figure(1), 'Colormap', negroblando);
```

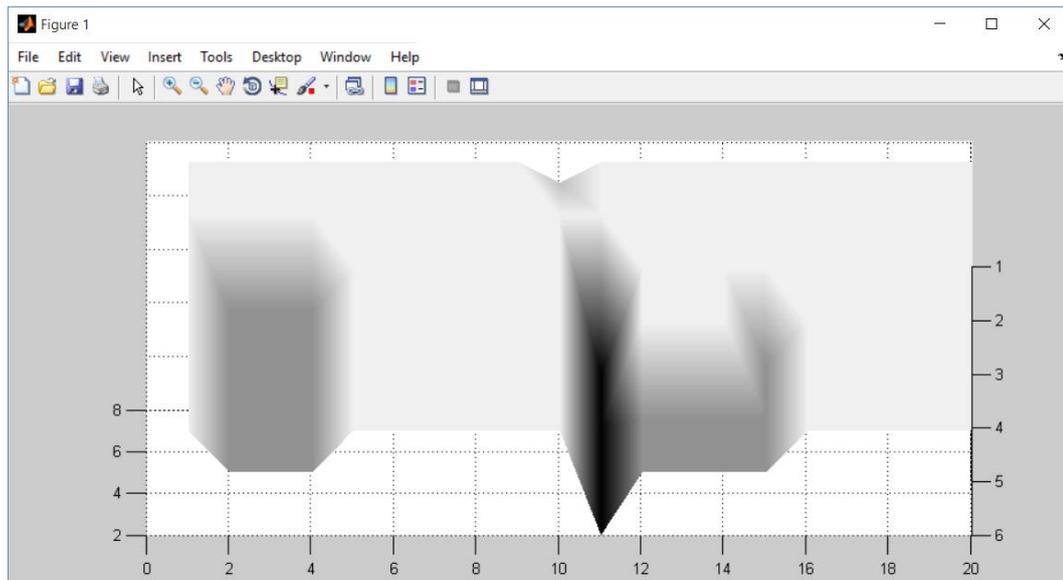


Figura 55.-Representación con mapa de colores negro-blanco.
Fuente: Autores.

Agregado de texturas

Para una mejor visualización, MATLAB permite cargar texturas y mapas de colores a una superficie, por medio de imágenes con extensiones de tipo: bmp, jpg, png,gif, entre otras. La adquisición de las imágenes se genera a través de una cámara web configurada de la siguiente manera:

```
vid=videoinput('winvideo',2,'YUY2_640x480');
set(vid, 'ReturnedColorSpace', 'RGB');
preview(vid);
img = getsnapshot(vid);
```

El almacenamiento de la imagen es posteriormente realizada y se procede a extraer la información contenida:

```
imwrite(img,'imagen1.png','png');
a=imread('imagen1.png');
```

Al obtener los datos de la imagen se forma nuevamente la graficación de la superficie y además se sobrepone una textura que corresponde a la imagen adquirida anteriormente:

```
surf(matriz,'CData',a,'FaceColor','texturemap')
view(0,-65)
```

2.7.3. Comprobación de Hipótesis

2.7.3.1. Planteamiento de hipótesis estadística

$$H_0: \text{Parametro} \leq x$$

$$H_1: \text{Parametro} > x$$

2.7.3.2. Establecimiento del nivel de significancia

Las pruebas se realizaron con un 95% de confiabilidad, es decir se trabajó con un nivel de significancia de $\alpha = 0,05$.

2.7.3.3. Determinación del valor estadístico de prueba

Al obtener un valor de muestras equivalente a 95 se utiliza el estadístico z.

Pruebas unilateral izquierda: la hipótesis planteada se formula con \geq o \leq

$$H_0: \mu \geq 10\%$$

$$H_1: \mu < 10\%$$

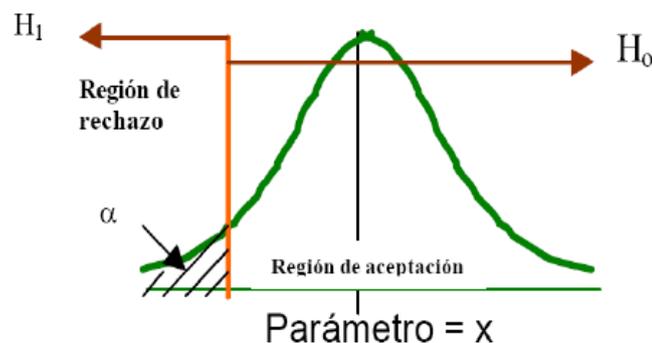


Figura 56.- Zonas de aceptación y rechazo.

Fuente: Apuntes de Estadística - Universidad de Puerto Rico.

Regla de decisión

Si $z \geq z_\alpha$ no se rechaza H_0 .

Si $z < z_\alpha$ se rechaza H_0 .

Por este motivo la regla de decisión precisa rechazar la hipótesis nula y aceptar la hipótesis alternativa, si el valor Z calculado no queda en la región comprendida entre $-x \leq z \leq x$. En caso contrario no se rechaza la hipótesis nula si Z queda entre $-x \leq z \leq x$.

PRUEBA N° 1
AMBIENTE PLANO

Datos:

Nro.	Distancia [cm]		% Error
	Real	Medida	
1	401	410	2,24
2	403	411	1,99
3	402	410	1,99
4	405	412	1,73
5	404	410	1,49
6	403	411	1,99
7	402	410	1,99
8	399	412	3,26
9	397	405	2,02
10	401	408	1,75
11	399	406	1,75
12	403	411	1,99
13	402	410	1,99
14	399	406	1,75
15	405	402	0,74
16	399	405	1,50
17	400	408	2,00
18	400	407	1,75
19	403	407	0,99
20	405	409	0,99
21	404	410	1,49
22	403	404	0,25
23	402	408	1,49
24	399	410	2,76
25	397	411	3,53
26	401	410	2,24
27	399	412	3,26
28	403	410	1,74
29	402	411	2,24
30	405	409	0,99
31	399	403	1,00
32	403	410	1,74

33	402	411	2,24
34	399	410	2,76
35	405	412	1,73
36	399	410	2,76
37	400	411	2,75
38	400	406	1,50
39	403	409	1,49
40	405	410	1,23
41	405	408	0,74
42	404	409	1,24
43	403	410	1,74
44	402	408	1,49
45	399	410	2,76
46	397	412	3,78
47	401	409	2,00
48	399	406	1,75
49	403	407	0,99
50	402	408	1,49
51	399	406	1,75
52	405	407	0,49
53	399	406	1,75
54	400	409	2,25
55	400	407	1,75
56	403	410	1,74
57	405	409	0,99
58	404	410	1,49
59	403	404	0,25
60	402	409	1,74
61	399	410	2,76
62	397	411	3,53
63	403	410	1,74
64	402	409	1,74
65	399	408	2,26
66	405	409	0,99
67	399	406	1,75
68	400	407	1,75
69	400	407	1,75
70	403	411	1,99

71	405	409	0,99
72	404	408	0,99
73	403	410	1,74
74	397	411	3,53
75	403	410	1,74
76	402	412	2,49
77	399	408	2,26
78	405	411	1,48
79	403	410	1,74
80	403	411	1,99
81	402	410	1,99
82	399	412	3,26
83	397	405	2,02
84	401	408	1,75
85	399	406	1,75
86	403	411	1,99
87	402	410	1,99
88	399	406	1,75
89	405	402	0,74
90	399	405	1,50
91	400	408	2,00
92	400	407	1,75
93	403	407	0,99
94	405	409	0,99
95	404	410	1,49

Tabla 12.- Valores de error para la mediciones en un ambiente plano.
Fuente: Autores.

La selección de la hipótesis nula y la hipótesis alternativa corresponde a:

$$H_0: \mu \geq 10\%$$

$$H_1: \mu < 10\%$$

El cálculo de la desviación estándar muestral y la media de la muestra empleando SPSS Statistics, devuelve los valores como se muestra el resultado en la siguiente tabla:

Estadísticos para una muestra

	N	Media	Desviación típica	Error típico de la media
Plano	95	1,8152	,70128	,07195

Tabla 13.- Estadísticos para una muestra para prueba N°.1.
Fuente: Autores.

Prueba para una muestra

	Valor de prueba = 10					
	z	Grados de libertad (gl)	Sig. (unilateral)	Diferencia de medias	95% Intervalo de confianza para la diferencia	
					Inferior	Superior
Plano	-113.758	94	,000	-8.18484	-8.3277	-8.0420

Tabla 14.- Prueba para una muestra para prueba N°.1.
Fuente: Autores.

1. Ho			
2. H1			
3. α	0,05		
4. Grados de Libertad	gl=n-1	95-1	94
5. Valor estadístico Z	1,985250956	Z α	1,9853

Tabla 15.- Valor estadístico Z para prueba N°.1.
Fuente: Autores.

Decisión y justificación

Como $-113.758 < 1.9853$ se rechaza Ho y se concluye, al tener un nivel de significancia del 0.05, que los valores de error medidos por el sistema graficador 3D tiene un error menor al 10% en las pruebas en un ambiente plano.

PRUEBA N° 2
AMBIENTE CON GRADAS

Datos:

Nro.	Distancia [cm]		% Error
	Real	Medida	
1	290	287	1,03
2	290	287	1,03
3	290	287	1,03
4	290	287	1,03
5	290	287	1,03
6	290	287	1,03
7	290	287	1,03
8	290	287	1,03
9	290	287	1,03
10	290	287	1,03
11	291	288	1,03
12	291	288	1,03
13	291	288	1,03
14	291	288	1,03
15	291	288	1,03
16	291	288	1,03
17	291	288	1,03
18	291	288	1,03
19	291	288	1,03
20	291	288	1,03
21	260	256	1,54
22	260	256	1,54
23	260	256	1,54
24	260	256	1,54
25	260	256	1,54
26	260	256	1,54
27	260	256	1,54
28	260	256	1,54
29	260	256	1,54
30	260	256	1,54
31	262	257	1,91
32	262	257	1,91

33	262	257	1,91
34	262	257	1,91
35	262	257	1,91
36	262	257	1,91
37	262	257	1,91
38	262	257	1,91
39	262	257	1,91
40	265	257	3,02
41	230	225	2,17
42	230	225	2,17
43	230	225	2,17
44	230	225	2,17
45	230	225	2,17
46	230	225	2,17
47	230	225	2,17
48	230	225	2,17
49	230	225	2,17
50	230	225	2,17
51	230	225	2,17
52	234	227	2,99
53	234	227	2,99
54	234	227	2,99
55	234	227	2,99
56	234	227	2,99
57	234	227	2,99
58	234	227	2,99
59	234	227	2,99
60	234	227	2,99
61	200	198	1,00
62	200	198	1,00
63	200	198	1,00
64	200	198	1,00
65	200	198	1,00
66	200	198	1,00
67	200	198	1,00
68	200	198	1,00
69	200	198	1,00
70	200	198	1,00

71	203	200	1,48
72	203	200	1,48
73	203	200	1,48
74	203	200	1,48
75	203	200	1,48
76	203	200	1,48
77	203	200	1,48
78	203	200	1,48
79	190	192	1,05
80	190	192	1,05
81	190	192	1,05
82	190	193	1,58
83	190	194	2,11
84	190	192	1,05
85	190	192	1,05
86	190	193	1,58
87	190	192	1,05
88	190	193	1,58
89	194	197	1,55
90	194	198	2,06
91	194	196	1,03
92	194	200	3,09
93	194	197	1,55
94	194	199	2,58
95	194	195	0,52

Tabla 16.- Valores de error para las mediciones en un ambiente con gradas.
Fuente: Autores.

La selección de la hipótesis nula y la hipótesis alternativa corresponde a:

$$H_0: \mu \geq 10\%$$

$$H_1: \mu < 10\%$$

El cálculo de la desviación estándar muestral y la media de la muestra empleando SPSS Statistics, devuelve los valores como se muestra el resultado en la siguiente tabla:

Estadísticos para una muestra

	N	Media	Desviación típica	Error típico de la media
Gradas	95	1,6248	,66043	,06776

Tabla 17.- Estadísticos para una muestra para prueba N°.2.
Fuente: Autores.

Prueba para una muestra

	Valor de prueba = 10					
	z	gl	Sig. (unilateral)	Diferencia de medias	95% Intervalo de confianza para la diferencia	
					Inferior	Superior
Gradas	-123.603	94	,000	-8.37516	-8.5097	-8.2406

Tabla 18.- Prueba para una muestra para prueba N°.2.
Fuente: Autores.

1. Ho			
2. H1			
3. α	0,05		
4. Grados de Libertad	gl=n-1	95-1	94
5. Valor estadístico Z	1,985250956	Z α	1,9853

Tabla 19.- Valor estadístico Z para prueba N°.2.
Fuente: Autores.

Decisión y justificación

Como $-123.603 < 1.9853$ se rechaza Ho y se concluye, al tener un nivel de significancia del 0.05, que los valores de error medidos por el sistema graficador 3D tiene un error menor al 10% en las pruebas en un ambiente con gradas.

PRUEBA N° 3
AMBIENTE CON OBSTÁCULOS

Datos:

Nro.	Distancia [cm]		% Error
	Real	Medida	
1	645	643	0,31
2	645	630	2,33
3	645	643	0,31
4	645	630	2,33
5	645	643	0,31
6	645	630	2,33
7	645	643	0,31
8	645	643	0,31
9	645	643	0,31
10	645	643	0,31
11	645	643	0,31
12	645	643	0,31
13	645	643	0,31
14	645	643	0,31
15	645	643	0,31
16	645	643	0,31
17	645	643	0,31
18	645	643	0,31
19	645	643	0,31
20	645	643	0,31
21	645	643	0,31
22	645	643	0,31
23	645	643	0,31
24	645	643	0,31
25	645	643	0,31
26	645	643	0,31
27	645	643	0,31
28	645	643	0,31
29	645	643	0,31
30	645	643	0,31
31	645	643	0,31
32	645	643	0,31

33	645	643	0,31
34	645	643	0,31
35	645	643	0,31
36	645	643	0,31
37	645	643	0,31
38	645	643	0,31
39	645	643	0,31
40	645	643	0,31
41	645	643	0,31
42	645	643	0,31
43	645	630	2,33
44	645	643	0,31
45	645	643	0,31
46	645	643	0,31
47	645	643	0,31
48	645	643	0,31
49	400	398	0,50
50	400	398	0,50
51	400	398	0,50
52	400	398	0,50
53	645	643	0,31
54	645	643	0,31
55	645	643	0,31
56	645	643	0,31
57	645	643	0,31
58	645	643	0,31
59	645	643	0,31
60	340	337	0,88
61	645	643	0,31
62	645	643	0,31
63	645	643	0,31
64	645	630	2,33
65	645	643	0,31
66	300	295	1,67
67	300	295	1,67
68	300	295	1,67
69	300	295	1,67
70	300	295	1,67

71	300	295	1,67
72	300	295	1,67
73	300	295	1,67
74	300	295	1,67
75	300	295	1,67
76	300	295	1,67
77	300	295	1,67
78	300	295	1,67
79	400	398	0,50
80	400	398	0,50
81	400	398	0,50
82	400	398	0,50
83	645	630	2,33
84	645	643	0,31
85	645	630	2,33
86	645	643	0,31
87	645	643	0,31
88	645	643	0,31
89	645	630	2,33
90	340	337	0,88
91	645	643	0,31
92	645	630	2,33
93	645	630	2,33
94	645	630	2,33
95	645	643	0,31

Tabla 20.- Valores de error para las mediciones en un ambiente con obstáculos.
Fuente: Autores.

La selección de la hipótesis nula y la hipótesis alternativa corresponde a:

$$H_0: \mu \geq 10\%$$

$$H_1: \mu < 10\%$$

El cálculo de la desviación estándar muestral y la media de la muestra empleando SPSS Statistics, devuelve los valores como se muestra el resultado en la siguiente tabla:

Estadísticos para una muestra

	N	Media	Desviación típica	Error típico de la media
Obstáculos	95	,7580	,73514	,07542

Tabla 21.- Estadísticos para una muestra para prueba N°.3.
Fuente: Autores.

Prueba para una muestra

	Valor de prueba = 10					
	z	gl	Sig. (unilateral)	Diferencia de medias	95% Intervalo de confianza para la diferencia	
					Inferior	Superior
Obstáculos	-122.534	94	,000	-9.24200	-9.3918	-9.0922

Tabla 22.- Prueba para una muestra para prueba N°.3.
Fuente: Autores.

1. Ho			
2. H1			
3. α	0,05		
4. Grados de Libertad	gl=n-1	95-1	94
5. Valor estadístico Z	1,985250956	Z α	1,9853

Tabla 23.- Valor estadístico Z para prueba N°.3.
Fuente: Autores.

Decisión y justificación

Como $-122.534 < 1.9853$ se rechaza Ho y se concluye, al tener un nivel de significancia del 0.05, que los valores de error medidos por el sistema graficador 3D tiene un error menor al 10% en las pruebas en un ambiente con obstáculos.

CAPÍTULO III

3. RESULTADOS

- Los datos obtenidos por medio de las pruebas, efectuadas en tres ambientes diferentes, representan el error entre la distancia real y la adquirida por el sensor LIDAR lite v2 incorporado sobre el sistema graficador 3D. Las pruebas fueron realizadas en un ambiente plano (pared), con gradas y obstáculos, permitiendo verificar el comportamiento del sistema graficador 3D para poder realizar el análisis de los datos reales y medidos, por consiguiente el error registrado en las mediciones fue menor significativamente al 10% en todos los entornos de prueba.
- El análisis estadístico de los datos obtenidos se comprobó a través de una prueba de valor Z para los tres distintos ambientes en estudio. Con los datos estadísticos obtenidos en las tres pruebas realizadas, se comprobó que el valor de Z calculado no se sitúa en la región comprendida entre $-x \leq z \leq x$ en la campana de Gauss, siendo por esta razón que la hipótesis nula se rechaza y se acepta la hipótesis alternativa que determina el valor del error en las mediciones considerado menor al 10%, es decir que el graficador funciona con una precisión superior al 90%.
- Se pudo constatar con los datos obtenidos que a menor distancia la representación gráfica presenta una mejor resolución, por el motivo de que las distancias entre los puntos que conforman un objeto disminuyen y se obtiene como resultado una mayor precisión.

CAPÍTULO IV

4. DISCUSIÓN

El sistema graficador 3D es un dispositivo capaz de analizar un entorno para reunir datos de formas y distancias. Esta información adquirida se puede emplear para producir modelos tridimensionales que se usan en diferentes aplicaciones. El propósito final es crear una nube de puntos. Estas nubes de puntos describen la posición en el espacio tridimensional de cada punto obtenido por el sistema graficador 3D.

Los principales usos de esta tecnología comprenden la medición de distancias, caracterización interna de materiales, la detección de posibles anomalías en campos relacionados con ingeniería civil y medicina; por ejemplo, en mapas de elevación.

En los últimos años se ha dado un incremento en las aplicaciones correspondientes en la industria automotriz. Los sensores se incorporan en la parte superior de los automóviles con el fin de controlar las distancias en las maniobras de estacionamiento y prevenir los choques con los obstáculos cuando se conduce marcha atrás (Villajos Rayo).

Las aplicaciones más importantes en las que se utilizan estos tipos de sistemas de graficación tridimensional son las siguientes:

En la Industria es utilizado para el control dimensional en la fabricación de componentes que requieren tolerancias muy estrictas y mecanizados de gran precisión. Las piezas se escanean y la nube de puntos se compara con el modelo teórico, permitiendo así un control minucioso de la producción.

Se realiza Ingeniería inversa a componentes mecánicos que requieren un modelo digital preciso de los objetos para ser reproducidos. Mediante el escaneo

tridimensional se consigue el mapa del objeto para después reproducirlo con fresadoras y tornos CNC.

Utilizado también para documentar patrimonios culturales, escaneando sitios y artefactos históricos. Esto favorece a la documentación y mantenimiento de edificaciones, monumentos y elementos históricos.

Finalmente en la actualidad, en la rama de robótica se consigue dar al robot un mapa tridimensional del entorno en el que se encuentra, con el fin de facilitar la navegación y el reconocimiento de objetos, generando mapas y modelos 3D.

CAPÍTULO V

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

Al terminar la presente investigación se concluye lo siguiente:

- ✓ El sistema es capaz de reconocer obstáculos que se encuentren a una distancia de hasta 10 metros y se los representó gráficamente en 3D mediante MATLAB. Sin embargo las características del sensor láser Lidar Lite v2 puede ser utilizado para distancias de hasta 40 metros.
- ✓ Para la recolección de los datos de distancia a una velocidad adecuada, se optimizó el proceso mediante el uso del protocolo de comunicación I2C entre el sensor láser Lidar Lite v2 y el microcontrolador STM32F429I a una velocidad de 100KHz.
- ✓ Las distancias obtenidas y graficadas en la interfaz de MATLAB son representadas por un mapa de colores monocromático y se relacionan directamente con el valor real de las distancias.
- ✓ Las medidas de distancia obtenidas por el sensor representan un porcentaje por debajo del 10% de error.
- ✓ Las superficies especulares provocan una lectura errónea de la medida de distancia por las propiedades de reflexión del láser.

5.2. Recomendaciones

Luego de haber realizado las conclusiones de la investigación se proponen las siguientes recomendaciones:

- ✓ Aumentar el número de sensores láser para disminuir el tiempo de adquisición de datos y aumentar la precisión del sistema.

- ✓ Reemplazar la interfaz ofrecida por un ordenador con un LCD TFT, para obtener un sistema de tamaño compacto, aprovechando las características del microcontrolador STM32F4.

- ✓ Este sistema propuesto puede ser modificado tanto en hardware como software, pudiendo ser utilizado para la creación de mapas de elevación y modelado 3D.

- ✓ Los sensores ultrasónicos no son recomendables para realizar un barrido de reconocimiento de obstáculos que exija una gran precisión en la recolección de datos.

CAPÍTULO VI

6. PROPUESTA

6.1. Título de la propuesta

IMPLEMENTACIÓN DE UN SISTEMA 3D EN MOVIMIENTO PARA NAVEGACIÓN VEHICULAR

6.2. Introducción

El difícil acceso y adquisición de dispositivos de parecidas características en el mercado es un factor limitante para poder desarrollar más prestaciones de estos dispositivos, así como también los costos elevados representan una restricción importante, por este motivo el presente proyecto es desarrollado con un bajo costo en implementación.

La posibilidad de desarrollar un graficador 3D que cumpla con las características del diseño y con la tecnología que utiliza reflexión de ondas en su funcionamiento, permite presentar una solución con un dispositivo capaz de poder determinar la ubicación y distancia de los obstáculos con respecto a un punto de origen, presentado la ventaja de adquirir tales parámetros de manera autónoma, sin la necesidad de usar herramientas manuales, permitiendo de esta forma la incorporación del sistema en un vehículo como una ayuda en la navegación.

Además, la rapidez y precisión de la obtención de los datos permite establecer la ubicación de los obstáculos con un diseño mecánico que permita realizar: 180° en giro (PAN) y 60° en inclinación (TILT), optimizando de esta manera la aplicación del graficador 3D.

6.3. Objetivos

6.3.1. Objetivo General

- Diseñar e implementar un sistema graficador 3D en movimiento incorporado en un vehículo.

6.3.2. Objetivos Específicos

- Analizar las características y los distintos elementos que van a conformar un sistema graficador 3D.
- Utilizar los dispositivos adecuados existentes en el mercado para desarrollar la adquisición de datos.
- Desarrollar un software que permita realizar los cálculos de detección ubicación y graficación 3D en malla, de obstáculos mediante el uso de MATLAB.
- Determinar su funcionamiento en diferentes superficies.

6.4. Fundamentación Científico-Técnica

El sistema graficador en 3D permite dentro de sus utilidades, una gran cantidad de aplicaciones futuras en lo que refiere la detección de obstáculos, por lo que su funcionamiento está orientado para la implementación posterior en vehículos en movimiento o cualquier otro medio móvil, limitándose a la determinación de la ubicación y distancia de los objetos que se encuentren en alcance con respecto a un punto de origen con el propósito de permitir una navegación segura mediante el reconocimiento y detección de objetos en un rango de corto alcance alrededor de 5 metros, proporcionando un sistema de guía autónomo.

6.5. Descripción de la propuesta

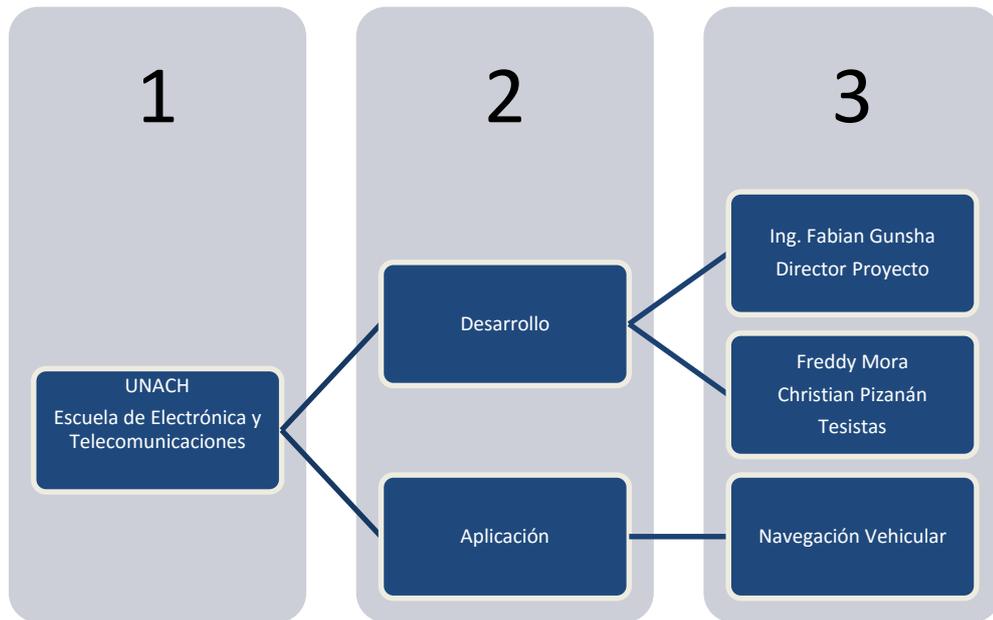
Este proyecto consiste en medir las distancias y por lo tanto su ubicación que será representado en grados PAN y TILT, mediante la programación se podrá procesar la información del sensor láser en que su funcionamiento se basa en la estimación del tiempo que transcurre entre la emisión de una señal láser y su respectiva señal de recepción después de haber sido reflejado por algún objeto u obstáculo.

El sistema graficador 3D consiste en la medición de las distancias y por medio de un sistema mecánico se pueda crear una representación 3D mediante la exploración de una área donde se implemente el sistema para esto se va a realizar el sistema mecánico con un arreglo de servomotores que nos permita hacer un barrido de PAN de 180° y un barrido de TILT de 60° del área a explorar. Esta área a explorar con el sistema se lo va a dividir en cuatro sub áreas para facilitar el procesamiento de datos una vez hecho la exploración cada control va a cubrir 90° de PAN y 30° de TILT.

El sistema ha de ser capaz de manejar entrada y salida (I/O), se necesitara de un circuito microcontrolador, quien controla los servomotores para el control de PAN y TILT para la determinación de distancias.

La adquisición de datos se la realizara con un núcleo STM32F429IG, que contiene un procesador ARM Cortex-4M de alta gama, en el que se realice el procesamiento de la información y estos datos de ubicación y distancia serán posteriormente enviados a la herramienta de software MATLAB y finalmente la representación gráfica en 3D.

6.6. Diseño Organizacional



6.7. Monitoreo y Evaluación de la propuesta

El monitoreo y la evaluación de la propuesta se la realizará a través de pruebas del sistema graficador variando los entornos y verificando que las distancias de los objetos sean o se aproximen a la realidad.

El impacto producido al implementar el sistema graficador 3D en movimiento es beneficioso, al poder establecer una navegación segura en vehículos que se encuentren circulando en ambientes con condiciones de poca luminosidad o escasa visión.

7. BIBLIOGRAFÍA

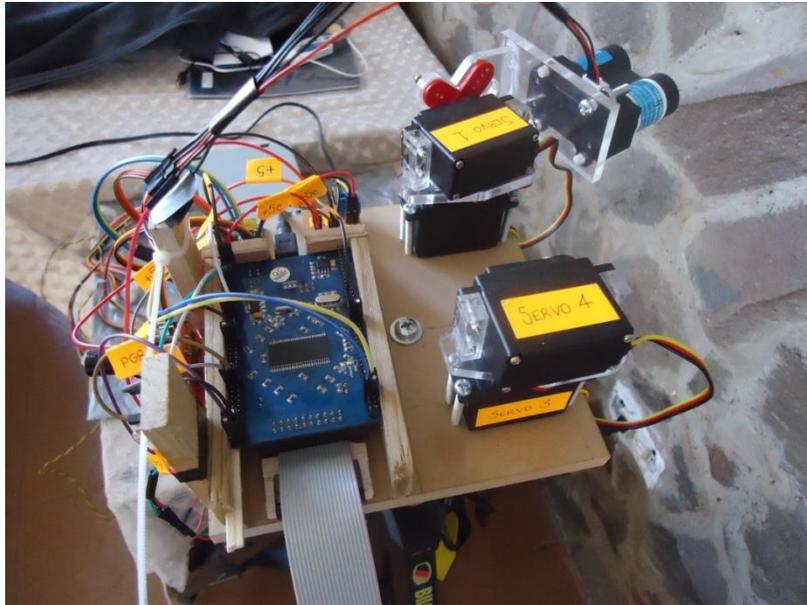
- [1] Aldaz Santana, L. (2005). *Análisis, Estudio y Simulación de los Parámetros del radar en el programa MATLAB*. Latacunga, Ecuador.
- [2] Almeida Hernández, I. L., & Ochoa Urgiles, J. A. (Abril de 2013). *Diseño y construcción de un robot explorador de terreno*. Guayaquil, Guayas, Ecuador.
- [3] Arrambí Díaz, M. (Junio de 2005). *Modelado de un Radar Doppler de Pulsos (PDR)*. Tijuana, México.
- [4] Ataurina Arellano, M. (2013). *MATLAB & Simulink para Ingeniería Nivel I*. Peru: Universidad de Ciencias y Humanidades.
- [5] Borja Macías, V. (Abril de 2012). *Herramientas computacionales para matemática - MATLAB Gráficas 3D*.
- [6] Caballero Escribano, J., & García Lorenz, M. (Octubre de 2011). *Creación de un entorno de desarrollo para aplicaciones basadas en microcontroladores STM32L Cortex-M3*. Madrid, España.
- [7] Castilla Gallardo, M. A. (2012). *Seguimiento virtual en tiempo real de maniobras de estabilización de un simulador de vuelo satelital*. Mexico DF, Mexico , Mexico.
- [8] Corcuera, P. (2012). *Gráficos 3D en MATLAB*. Cantabria, España.
- [9] Corona Nakamura, M. A., & Ancona Valdez, M. d. (2011). *Diseño de algoritmos y su codificación en lenguaje C*. Guadalajara, Mexico: McGraw-Hill.
- [10] Cortes Valencia, J. H., Castaño Acosta, L. A., & Morales Aristazabal, W. H. (2010). *Prototipo experimental de un sistema de detección de objetos por medio de imagenes y sensores para invidentes (DOMIS)*. Pereira, Risabalda, Colombia.
- [11] Fernández Saucedo, M. A. (Agosto de 2007). *Graficación*. Morelia, Mich.
- [12] Garcia, E. (2012). *Web08*. Recuperado el 1 de Noviembre de 2015, de <https://web08.uah.es/>
- [13] Hitex. (2009). *The Insiders Guide to the STM32*. Coventry: Hitex (UK) Ltd.
- [14] Martínez Arias, B. R. (2007). *Arquitectura de procesadores ARM de 32 bits, Circuitos integrados y Entornos de desarrollo*. Sangolquí, Cotopaxi, Ecuador.
- [15] Panata Cumbajin, C. D. (Marzo de 2008). *Diseño, construcción y prueba de un prototipo de un sonar para navegacion vehicular*. Quito, Pichincha, Ecuador.
- [16] Raúl, C. N. (Noviembre de 2007). *Modelados de registros de geo-radar en 2d y 3d*. Madrid, España.
- [17] Rodríguez Astol, J. G. (11 de Febrero de 2010). *Programación de un registrador de datos sísmicos autónomo de bajo consumo*. Cataluña, España.
- [18] Rubio Segovia, S. N., & Villao Carrillo, K. S. (2010). *Medidor de distancia con sensores ultrasonicos utilizando microcontrolador avanzado*. Guayaquil, Guayas, Ecuador.
- [19] Villacís Rosales, M. R., & Telenchana Manobamba, B. R. (Mayo de 2013). *Diseño y construcción de una plataforma robótica para control ptz de un sistema electro-óptico*. Quito, Pichincha, Ecuador.
- [20] Villao Carrillo, K. S., & Rubio Segovia, S. N. (2010). *Medidor de distancia con sensores ultrasónicos utilizando un microcontrolador avanzado con comunicación serial a datalogger e interfaz gráfica*. Guayaquil, Guayas, Ecuador.

ENLACES WEB

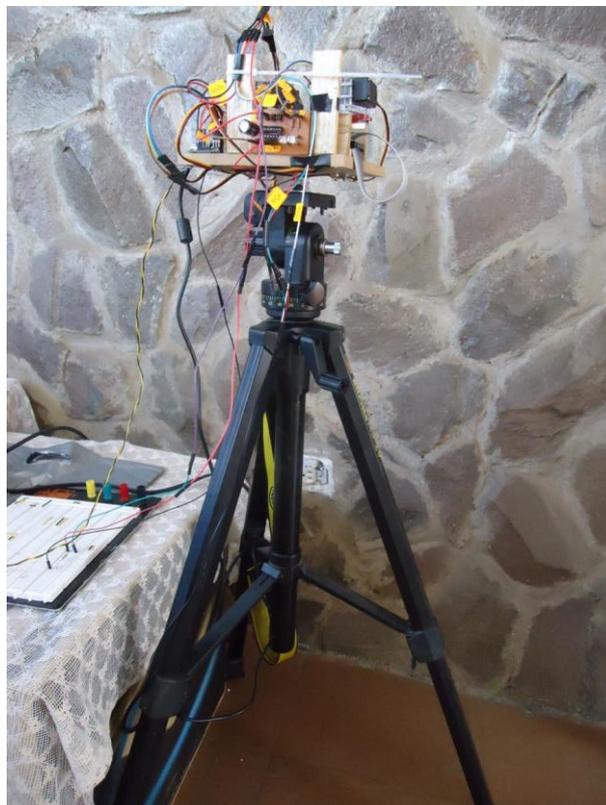
- [1] <http://electronicatk.blogspot.com/2015/06/placa-stm32f4-discovery.html>
- [2] <http://www.emcu.it/STM32F4xx/STM32F4-Library/STM32F4-Library.html>
- [3] <http://www2.keil.com/mdk5/install/>
- [4] <http://www.keil.com/support/man/docs/uv4/>
- [5] <http://www.keil.com/download/list/arm.htm>
- [6] http://www.keil.com/uvision/db_sim_prf_peripherals.asp
- [7] <http://www.electronics-homemade.com/STM32F4-Turn-on-LED.html>
- [8] <http://www.industriaembedahoy.com/kit-de-desarrollo-para-stm32f4/>
- [9] <http://trisuliswanto.blogspot.com/2014/08/instalasi-keil-uvision-5-mdk-arm.html>
- [10] <https://microcontroladoresv.wordpress.com/arquitectura-de-los-microcontroladores/>
- [11] <http://www.fidemar.com.uy/site/www/productos/sensores-de-proximidad.c7.html>
- [12] <http://dominion.com.mx/descargas/sensores-de-proximidad.pdf>
- [13] http://www.academia.edu/8800268/como_funcionan_los_sensores_magnetoresistivo_funciona
- [14] http://www.st.com/web/en/catalog/mmc/FM141/SC1169?s_searchtype=reco

8. APÉNDICES Y ANEXOS

Anexo 1.- Sistema graficador 3D.



Base del sistema graficador.

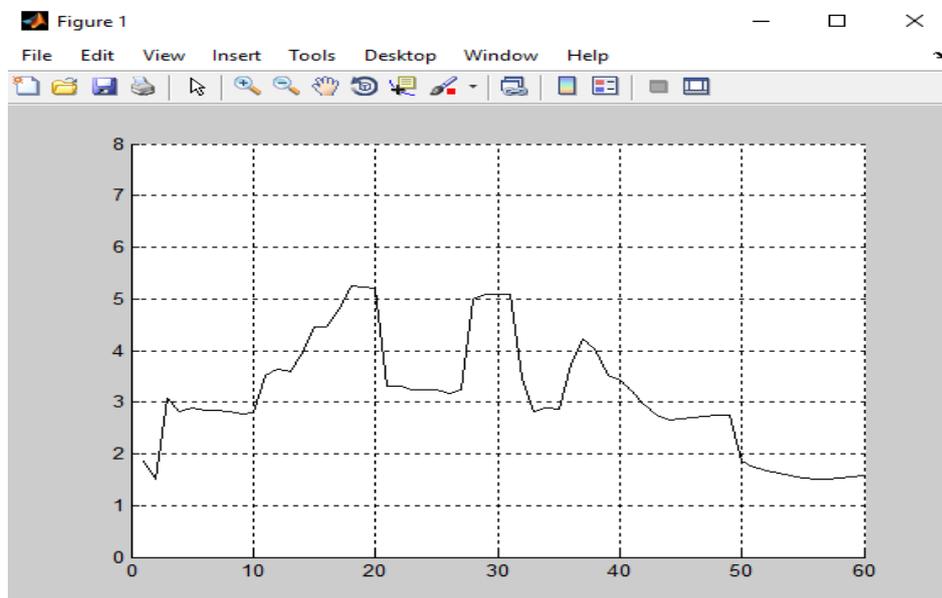


Montaje del sistema en trípode con altura ajustable.

Anexo 2.- Evidencias de pruebas iniciales.



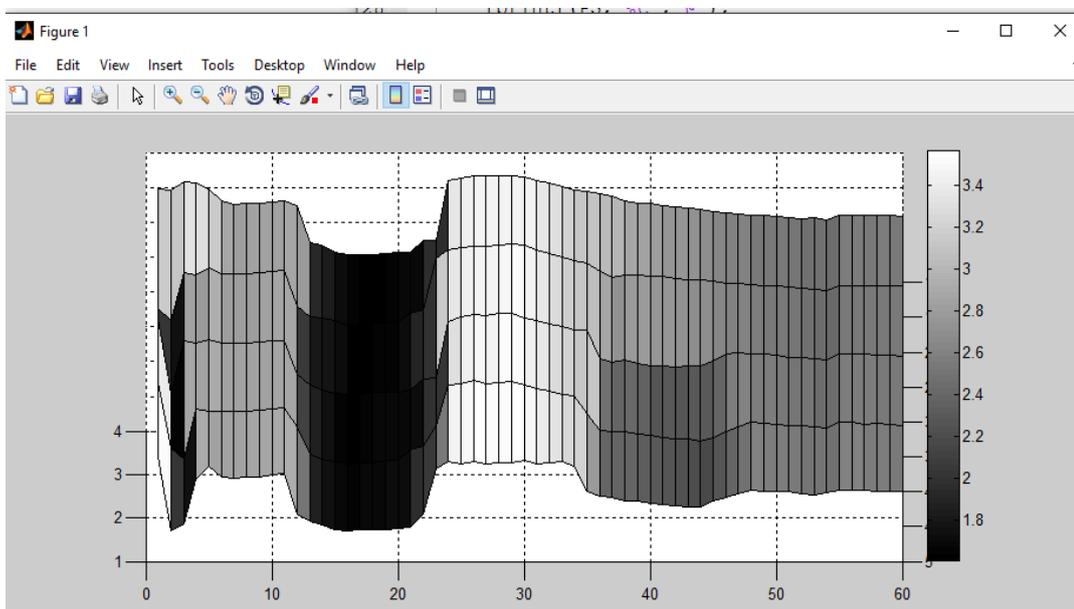
Entorno de prueba 1.



Perfil de entorno de prueba 1, vista superior.



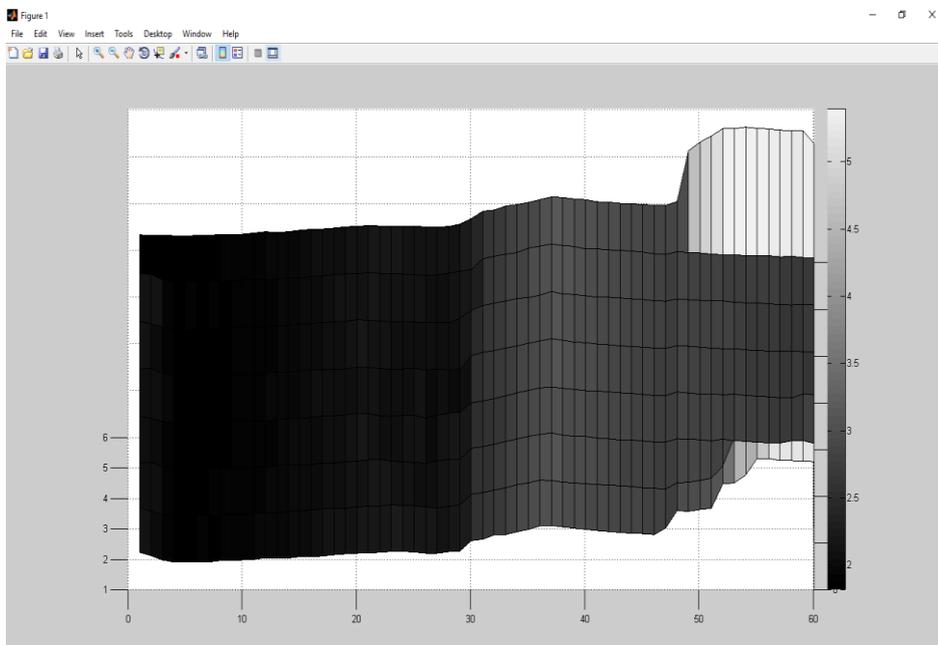
Entorno de prueba 2.



Prueba de reconocimiento de entorno 2.



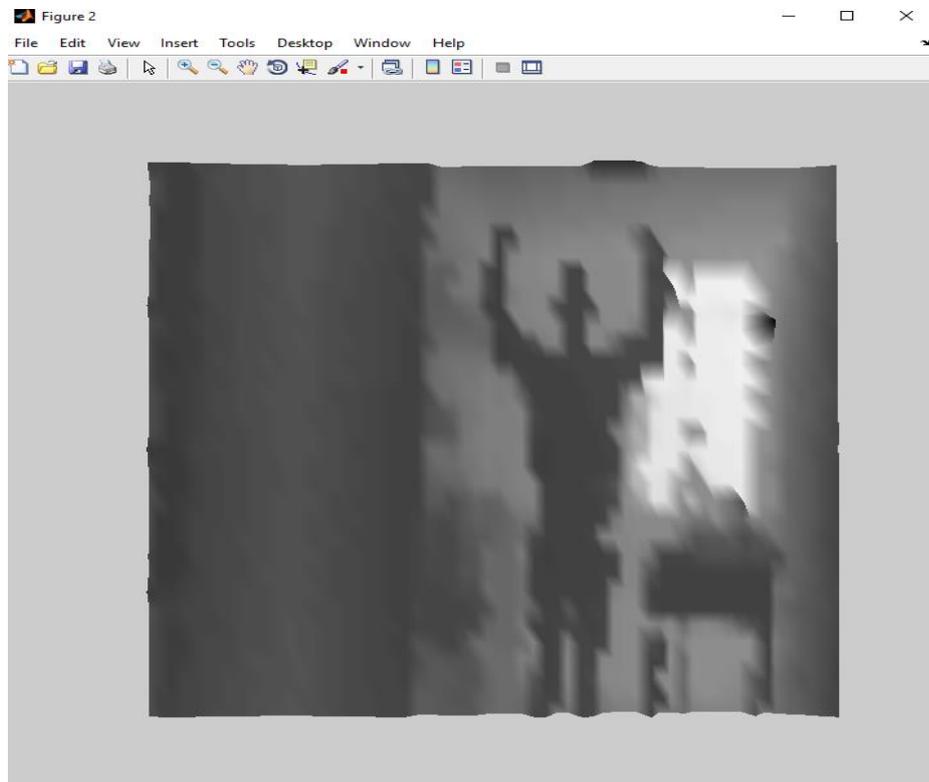
Entorno de prueba 3.



Prueba de reconocimiento de entorno 3.



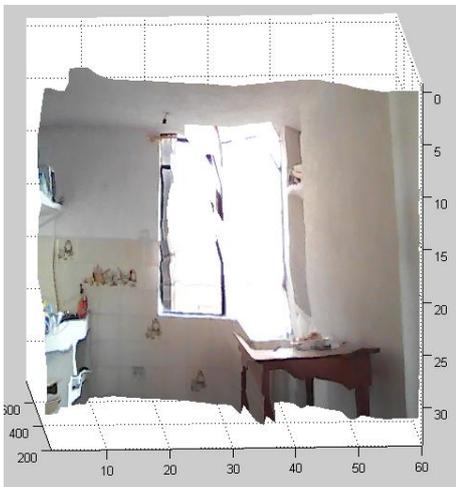
Entorno de prueba 4.



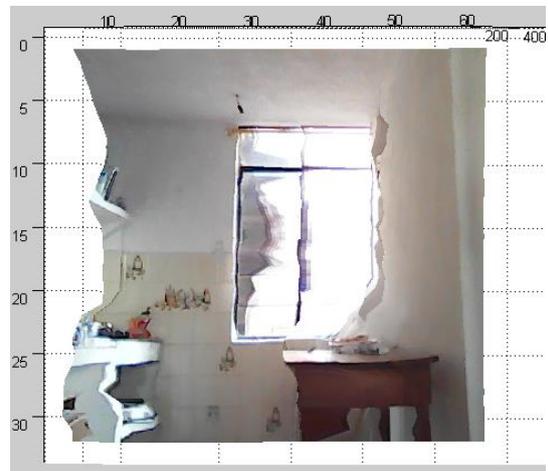
Prueba de reconocimiento de entorno 4.



Entorno de prueba de textura.



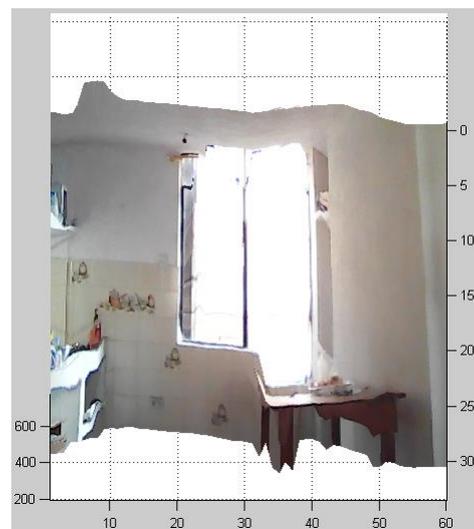
Vista lateral izquierda



Vista lateral derecha

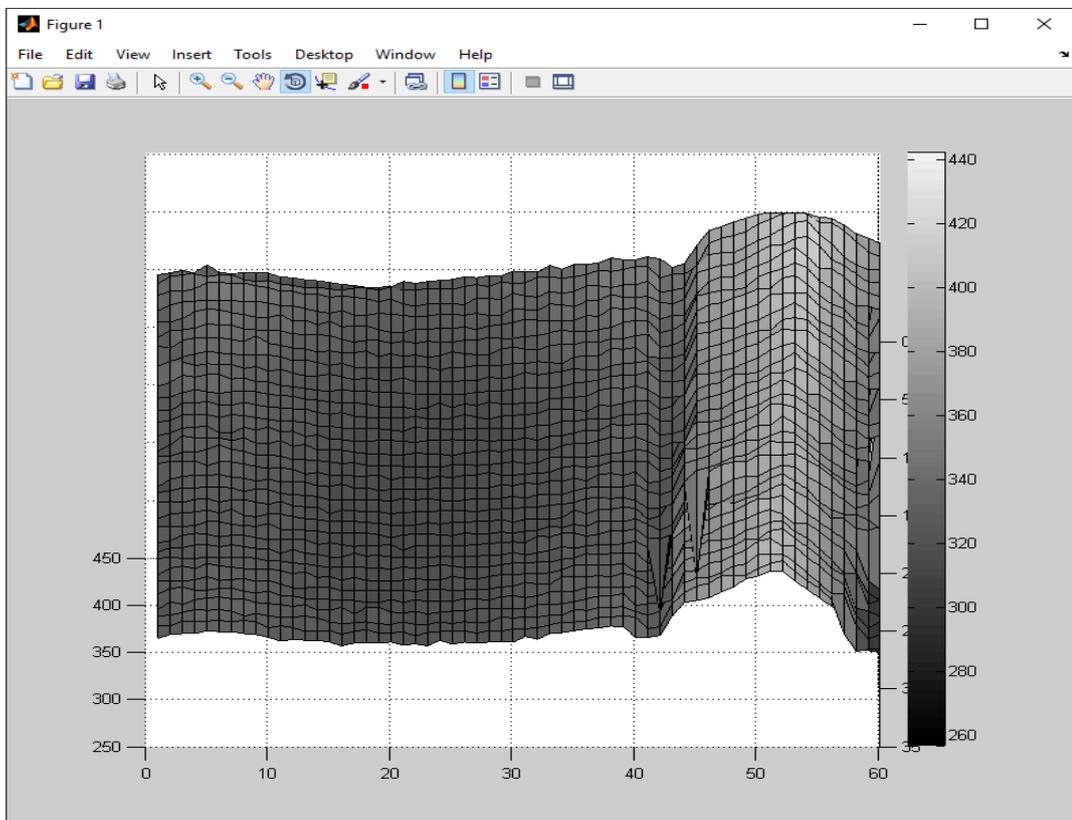


Vista inferior

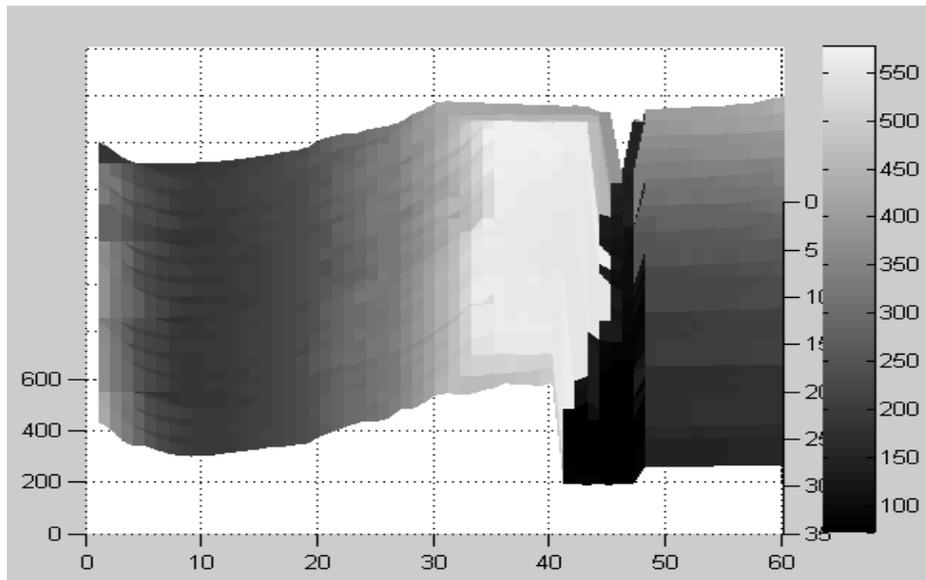


Vista superior

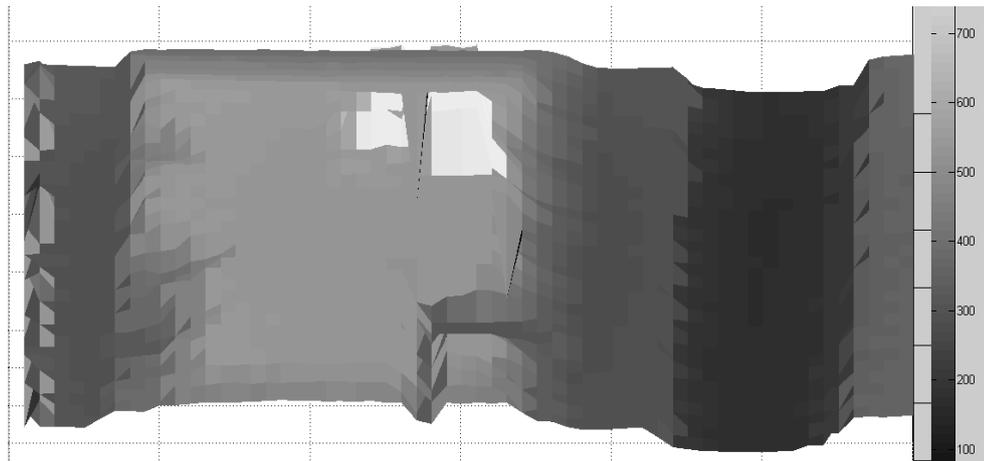
Anexo 3.- Evidencias de pruebas realizadas en un ambiente plano



Anexo 4.- Evidencias de pruebas realizadas en un ambiente con gradas.



Anexo 5.- Evidencias de pruebas realizadas en un ambiente con obstáculos.

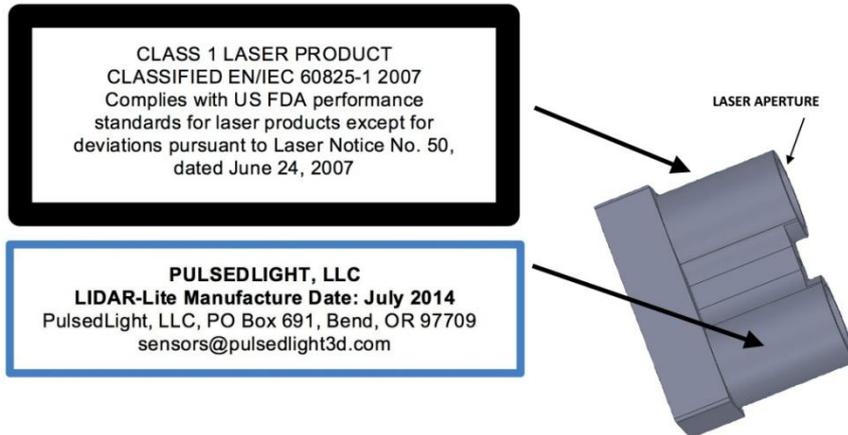


Anexo 6.- Datasheet Sensor láser LIDAR lite v2.

LIDAR-Lite Specifications

General	Technical Specifications
Power	4.75-5.5V DC Nominal, Maximum 6V DC
Weight	PCB 4.5 grams, Module 22 grams with optics and housing
Size	PCB 44.5 X 16.5mm (1.75" by .65")
Housing	20 X 48 X 40mm (.8" X 1.9" X 1.6")
Current Consumption	<2mA @ 1Hz (shutdown between measurements), <100mA (continuous operation)
Max Operating Temp.	70° C
External Trigger	3.3V logic, high-low edge triggered
PWM Range Output	PWM (Pulse Width Modulation) signal proportional to range, 1msec/meter, 10µsec step size
I2C Machine Interface	100Kb – Fixed, 0xC4 slave address. Internal register access & control.
Supported I2C Commands	Single distance measurement, velocity, signal strength
Mode Control	Busy status using I2C, External Trigger input / PWM outputs
Max Range under typical conditions	~40m
Accuracy	+/- 2.5cm, or +/- ~1"
Default Rep Rate	~50 Hz.

Laser Safety



LIDAR-Lite is a laser rangefinder that emits laser radiation. This Laser Product is designated Class 1 during all procedures of operation. This means that the laser is safe to look at with the unaided eye. However, it is very advisable to avoid looking into the beam and power the module off when not in use.

No regular maintenance is required for LIDAR-Lite. In the event that the unit becomes damaged or is inoperable, repair or service of LIDAR-Lite is only to be handled by authorized, factory-trained technicians. No service of LIDAR-Lite by the user is allowed. Attempting to repair or service the unit on your own can result in direct exposure to laser radiation and the risk of permanent eye damage. For repair or service please contact PulsedLight directly for a return authorization.

No user should modify LIDAR-Lite or operate it without its housing or optics. The operation of LIDAR-Lite without a housing and optics or modification of the housing or optics that exposes the laser source may result in direct exposure to laser radiation and the risk of permanent eye damage. Removal or modification of the diffuser in front of the laser optic may result in the risk of permanent eye damage.

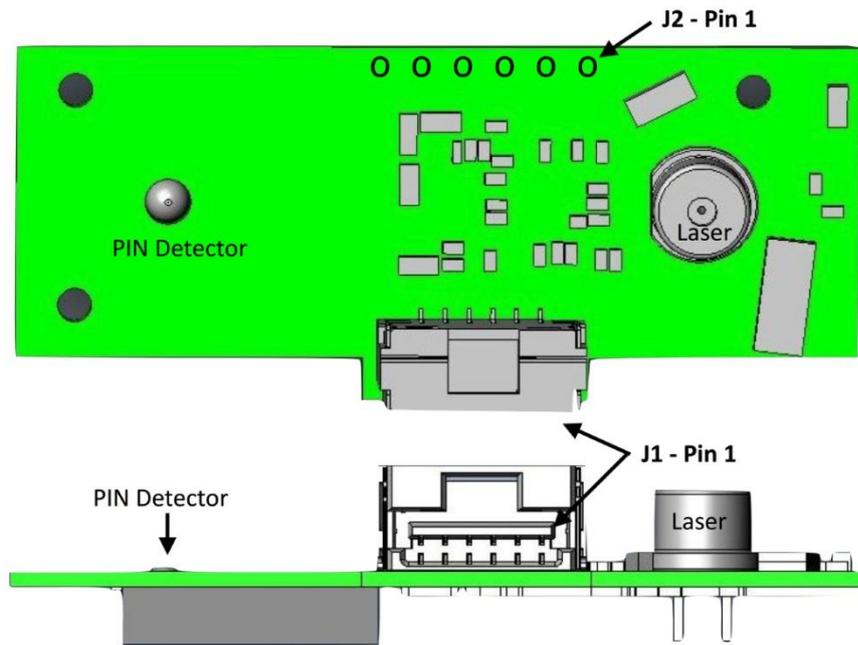
Caution – Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure. PulsedLight is not responsible for injuries caused through the improper use or operation of this product.

Class 1 Laser Product

This Laser Product is designated Class 1 during all procedures of operation.

Parameters	Laser Value
Wavelength	905nm (nominal)
Total Laser Power - Peak	1.3Watts
Mode of operation	Pulsed (max pulse train 256 pulses)
Pulse Width	0.5 μ Sec (50% duty Cycle)
Pulse Repetition Frequency	10-20KHz nominal
Energy per Pulse	<280nj
Beam Diameter at laser aperture	12mm x 2mm
Divergence	4mRadian x 2mRadian (Approx)

LIDAR-Lite Signal & Power Interface Definitions



J1 - Primary interface

Board Connector: Molex part #5023860670 (DigiKey Part #: WM3917CT-ND)

Mating Connector: Molex # 5023800600 PLUG HSG 6POS (DigiKey Part #: WM2271-ND)

Pin	Description
PIN1	POWER_IN – 4.75-5.5V DC Nominal, Maximum 6V DC. Peak current draw from this input (which occurs during acquisition period) is typically < 100 mA over a duration from 4 to 20ms depending on received signal strength. Unless you use power management, the unit will draw 80 mA between acquisition times.
PIN2	POWER_EN - Active high, enables operation of the 3.3V micro-controller regulator. Low puts board to sleep, draws <40 μ A. (Internal 100K pull-up)
PIN3	Mode Select – Provides trigger (high-low edge) PWM out (high)
PIN4	I2C Clock (SCL)

PIN5	I2C Data (SDA)
PIN6	Signal/power ground.

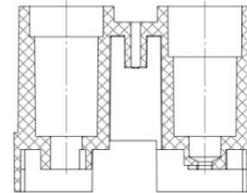
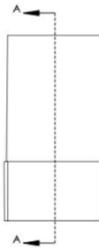
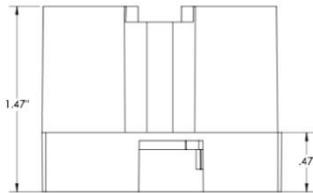
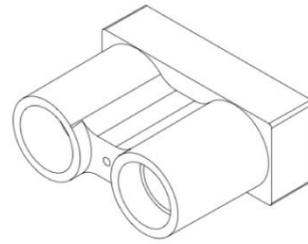
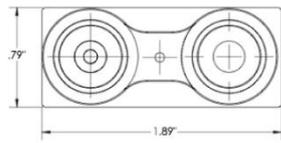
J2 - Secondary signal/power - .1" spacing Molex style through hole (Factory Option Only)

Pin	Description
PIN1	Laser Bypass 5-20V max (nominally connected to pin 2 through inductor L8 – removed for external power)
PIN2	POWER_IN – 4.75-5.5V DC Nominal, Maximum 6V DC
PIN3	POWER_EN - Active high
PIN4	External reference clock input (Factory Option – Consult Factory)
PIN5	Signal/power ground.
PIN6	Detector bias – up to 25V external bias for PIN, external bias input 200V for APD (consult factory)

Module Mechanical Drawings & Dimensions

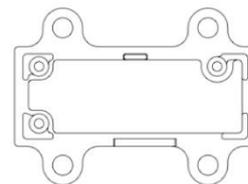
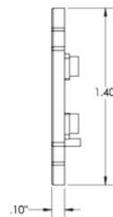
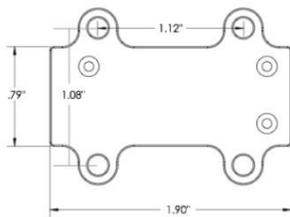
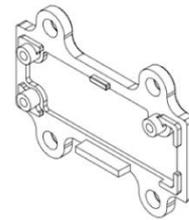
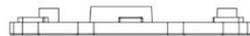
[Download Housing and Cover in PNG, STL, STEP and SLDPRPT formats on GitHub](#)

PulsedLight, Inc. - LIDAR-Lite Housing



SECTION A-A

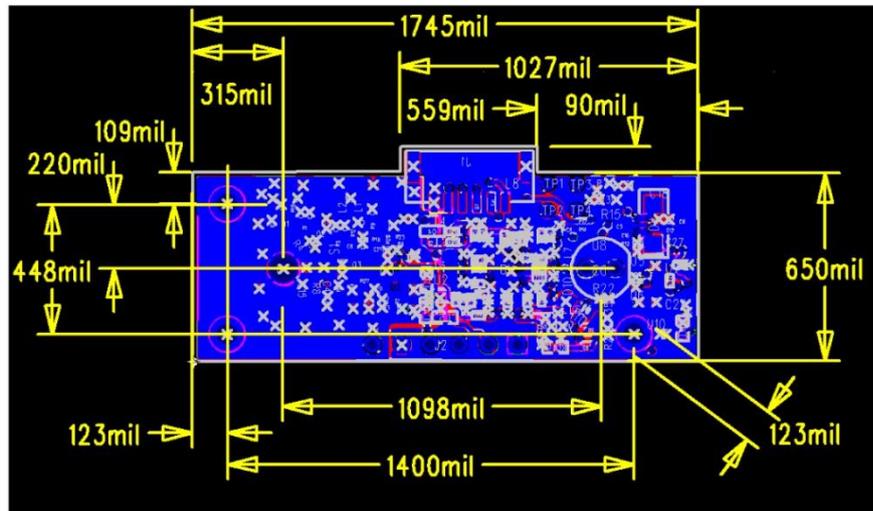
PulsedLight Inc. - LIDAR-Lite Housing Cover



Notes: - The hole diameters at their smallest are 0.15" or 0.385 cm with a little rounding erring small

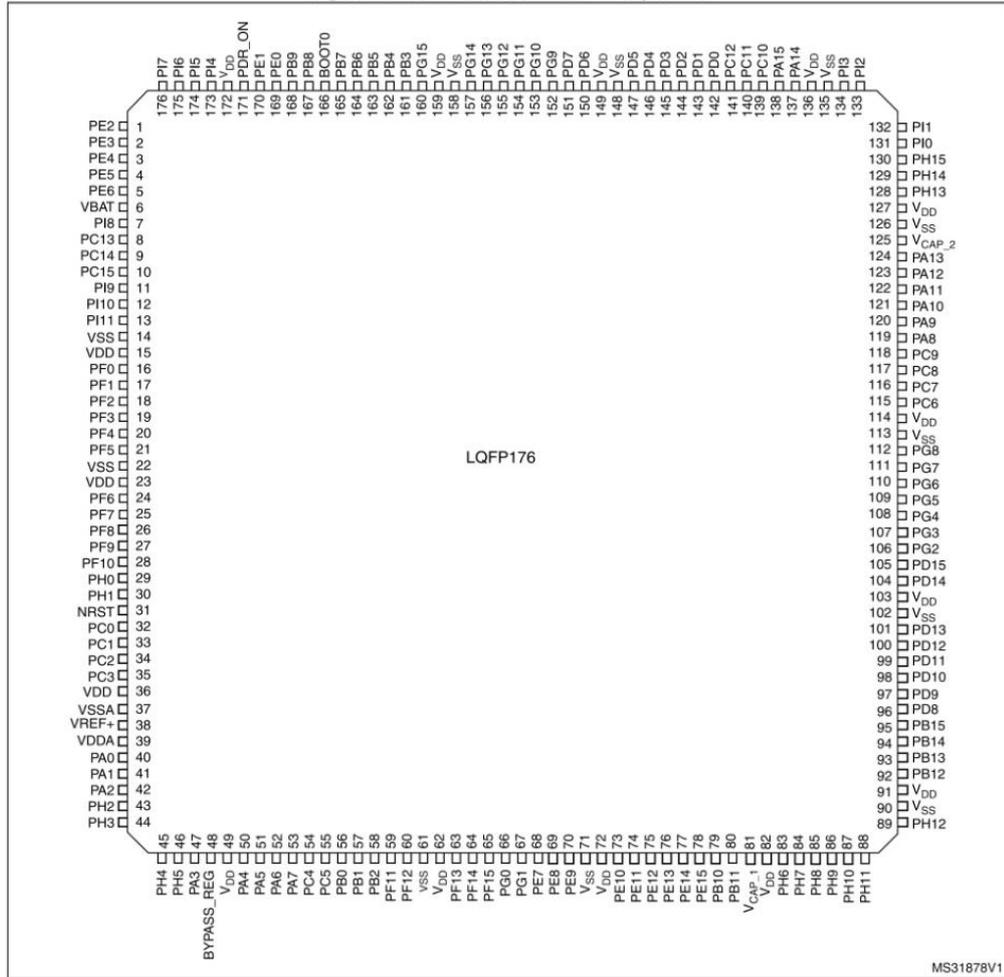
PCB Dimensions

Backside viewed through board



Anexo 8.- Pines procesador ARM Cortex-M4.

Figure 14. STM32F42x LQFP176 pinout



1. The above figure shows the package top view.

Anexo 9. - Core429IG Schematic.

