UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

TFG del Grado en Ingeniería Informática

Detección de objetos con láser de Seguridad

Presentado por Álvaro Ruifernández Palacios
en Universidad de Burgos — 19 de enero de 2019
Tutor: Jesús Enrique García Sierra

UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática

D. Jesús Enrique García Sierra, profesor del departamento de Ingeniería Civil, área de Sistemas informáticos.

Expone:

Que el alumno D. Álvaro Ruifernández Palacios, con DNI 71365383V, ha realizado el Trabajo final de Grado en Ingeniería Informática titulado Detección de objetos con láser de Seguridad.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 19 de enero de 2019

Vº. Bº. del Tutor:	Vº. Bº. del co-tutor:
D. nombre tutor	D. nombre co-tutor

Resumen

Los vehículos autoguiados o AGVs son una de las grandes innovaciones en el sector de la industria de estos últimos años. Estos son capaces de cumplir trabajos de traslado y colocación de mercancías en los distintos sectores de la industria en la que se encuentran instalados sin la necesidad de operarios que los manejen.

Para realizar estas tareas, además de los sistemas de guiado, necesitan un sistema de detección de objetos en su entorno, tanto para localizar los elementos a transportar como para detectar obstáculos . Para esta tarea los estos sistemas llevan incorporado un láser de seguridad el cual hace lecturas periódicas de su entorno.

Dos de los problemas que poseen estos láseres son: el número de campos de seguridad posibles y la exactitud de las detecciones. Estos láseres tienen como característica la posibilidad un número limitado de campos de área en los cuales solo se detectan las variaciones en cuanto a los puntos detectados por el láser pero sin especificar donde se encuentra el objeto exactamente, es decir, devuelven la detección de un objeto en un campo pero no en que parte del mismo se encuentra.

En este proyecto se ha desarrollado un software destinado al láser para detectar objetos de manera mucho más precisa de lo que lo hacía hasta ahora los láseres y poder detectar estos objetos en un número de campos de área casi ilimitado, eliminando también la primera de las restricciones descritas anteriormente.

Descriptores

Láser, AGV, vehículos autoguiados,detección de objetos, Python,Spyder,Jupyter Notebook...

Abstract

Automated guided vehicle o AGVs are one of the most important innovations in the industry of the last few years. These are able to complete labors of transfer and placement of goods in the different sectors of the industry they are installed without any driver.

To perform these tasks, apart from the guidance systems, they need a object detection system in their environment, both to locate the goods to be transported and to detect obstacles. Thus, these systems incorporate a safety laser which makes periodic measurements of their surroundings.

Some of the problems or aspects to improve these lasers have are 2: the limited possible areas and the accuracy of measurements.One of the problems these lasers have is the limited number of areas which only detect variations in the measurements detected by the laser with no specification about where exactly the object is.

In this project software was developed for a laser to detect objects far more accurate than current lasers and be able to detect these objects in a almost unlimited number of areas, also deleting the first of the restrictions previously described.

Keywords

Láser, AGV,automated guided vehicle,object detection, Python,Spyder,Jupyter Notebook...

Índice general

Índice de figuras

Índice de tablas

Introducción

Los AGV [1] son una de las últimas innovaciones en el mundo de la industria. Son robots autónomos que sustituyen al trabajo humano en el sentido de transporte de objetos o mercancías entre las distintas secciones de las industrias.

Para la realización de sus propósitos estos autómatas, además de sus sistemas de guiado, necesitan estructuras que les permitan reconocer si en su camino existe algún obstáculo o si el objeto con el que trabaja (mercancía) se encuentra cerca de ellos. Para esto, los AGVs llevan incorporado, como norma general, un láser de seguridad, el cual recoge un gran conjunto de datos sobre el entorno que lo rodea.

Actualmente, esta tecnología se basa en la configuración de áreas en las cuales el láser detecta las variaciones de puntos que se localizan dentro de estas, representado estas variaciones de forma booleana a través de una alarma. Los láseres de seguridad poseen un número limitado de estas áreas, también denominadas campos de área, las cuales, como bien se ha dicho antes, solo ofrecen no ofrecen ninguna información de donde se encuentra un objeto debido al hecho de detectar únicamente variaciones en áreas de un gran tamaño.

En este proyecto se desarrollará un sistema con el que se pretende solventar los problemas anteriormente descritos, ya que permitirá un número de áreas prácticamente ilimitado y la posibilidad de conocer de forma exacta la posición de los objetos que el láser detecte en su entorno, lo cual no se ha hecho hasta ahora con este tipo de láseres.

Objetivos del proyecto

En este apartado se explican los distintos objetivos identificados en este proyecto, distinguiendo entre los objetivos generales del proyecto y los objetivos técnicos.

2.1. Objetivos generales

Se tiene por objetivo crear un software capaz de realizar un análisis de los datos recibidos del láser para identificar la posición de los objetos de su entorno. Por lo tanto, los objetivos generales de este proyecto son:

- Crear una conexión para la comunicación con el láser
- Intercambiar mensajes con el láser para obtener sus datos de lectura.
- Realizar un análisis de dichos datos para comprobar si existe o no un objeto en las áreas requeridas por el usuario.

2.2. Objetivos funcionales

Los objetivos funcionales estimados para este proyecto son:

- El usuario podrá escoger el número de zonas que desea analizar, siendo este número tan grande como desee.
- El usuario decidirá cada una de las coordenadas en las que se encuentran los límites de las áreas que pretende analizar.

- El usuario podrá ejecutar cuantas veces quiera el programa para poder observar diferentes lecturas.
- El usuario podrá observar una serie de mensajes "true.º "false"(en el orden de inserción de las áreas anteriormente establecidas) para comprobar en qué áreas se ha detectado un objeto y en cuales no.
- El usuario podrá observar de forma gráfica (más concretamente en un gráfico de puntos) los resultados del análisis y la posición de las áreas establecidas.
- El usuario será capaz de observar la distancia a la que se encuentran los objetos detectados en el análisis.

2.3. Objetivos técnicos

Los objetivos técnicos planteados para este proyecto son:

- Aprender el uso de Python para la creación del código del proyecto, a través de todas sus funcionalidades previstas con el objetivo de la lectura de datos de las lecturas del láser.
- Usar librerías de Python no empleadas hasta el momento en ninguna asignatura del grado como binascii o re, las cuales han servido para poder manejar los datos recibidos y desarrollar el comportamiento deseado.
- Encontrar todos los elementos hardware necesarios para la creación del sistema hardware necesario para el proyecto.
- Crear un proyecto el cual implementa la funcionalidad que se requiere en los objetivos del proyecto, es decir, que realice la lectura, tratamiento y análisis de los datos del láser.
- Utilizar un sistema de control de versiones como es Git, más concretamente uno de sus servicios centrales más utilizados conocido como GitHub. Esta herramienta es útil en el caso de producirse algún error grave en alguna de las partes del proyecto ya que se puede restaurar la versión previa (la cual se considera estable).

Conceptos teóricos

En este proyecto se deben de emplear una gran cantidad de conocimientos adquiridos a lo largo de las asignaturas del grado y otros que se han ido aprendiendo a lo largo del desarrollo del mismo. En esta sección se describirán cuales son esos conceptos además de la forma en la que se aplican en este proyecto.

3.1. Organización de la infraestructura

Para la implementación del sistema que nos va a permitir el desarrollo del programa que tiene como objetivo este proyecto se necesitarán tres elementos principales:

- Láser: esta parte es la que se encarga de obtener los datos de su entorno. Estos datos serán enviados a través de una interfaz hacia la unidad de tratamiento de datos. En este caso, el láser que se va a emplear es un LIDAR (o Laser Imaging Detection and Ranging). Estos dispositivos tienen un funcionamiento muy sencillo, ya que se basan en la emisión de una gran cantidad de rayos infrarrojos a través de su foco emisor, los cuales son observados gracias a una lente infrarroja. Algunos de estos dispositivos poseen una característica que podría ser tomada como una desventaja, ya que son caaces de recoger información a cerca de su entorno pero solo en un ángulo de 270º. Es por esto que algunos de estos se encuentran instalados en soportes donde se les permita girar para poder obtener información del entorno completo. Tras heber analizado algunos de los LIDAR presentes hoy en día en el mercado (Sick S300, Benewake CE30-D....) se ha decidido utilizar el Hokuyo Safety Laser Scanner (UAM-05LP-T301). Este dispositivo es capaz de desarrollar tres áreas de detección dependiendo de la distancia a la

que detecte un elemento (puede ser de 20, 10 y 5 metros).

Figura 3.1: Rango de visión del Hokuyo

Para poder comparar con otro láser, en la siguiente imagen se puede observar un lidar de la marca Sick, concretamente el modemo S300, en el cual podemos observar la diferencia en cuanto a la forma de las áreas, la cual posee una forma más rectangular. lo cual es debido a que este tipo de láseres posee una configuración dedicada a una dirección específica.

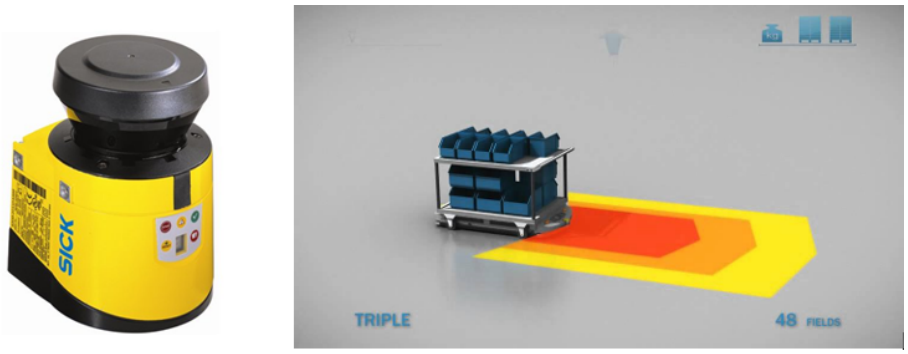

Figura 3.2: Lidar Sick y representación de sus campos de seguridad

- Cable ethernet: es el soporte a través del cual se transportan tanto los mensajes enviado por el ordenador hacia el láser como las respuestas y los datos de lectura que este manda al PC demandante. Esta es la que, en la descripción del anterior elemento se conoce como la interfaz. En el caso de este proyecto se usará un UAM-NET, un cable Ethrernet de 3 metros de longitud desarrollado por Hokuyo, misma empresa desarrolladora del láser empleado, lo que hace que resulte idóneo para evitar problemas de incompatibilidad y asegurar así el correcto funcionamiento del sistema.

- PC: es la parte principal del proyecto, ya que es la encargada de permitir al usuario tanto introducir las áreas donde se necesita detectar los objetos, como las órdenes que el usuario desea transmitir al láser para recibir la información. Además también es la encargada de recibir y procesar las respuestas del láser para poder mostrar al usuario los resultados de la comparación de los datos y las coordenadas de las áreas introducidas. La utilización de este aparato es debida a que este proyecto sería destinado a ser introducido dentro de un AGV pero se necesita hacer visible al usuario a través de una pantalla.

Figura 3.3: Sistema hardware del proyecto

3.2. Procesado de datos

Cuando el programa desarrollado se comunica con el láser, este debe recibir las tramas que el láser crea al analizar su área de observación, las cuales deben de ser tratadas para separar la información importante de las cabeceras y demás datos que el láser usa para interactuar con su aplicación. Una vez hemos conseguido extraer ese tipo de datos, hemos de descartar aquellos datos correspondientes a las zonas donde el láser no detecta ningún objeto, quedándonos solo con aquellas lecturas con valor significativo para la función principal del programa. Tras esto, se analizarán los datos para ver si se encuentran dentro de los límites establecidos por cada una de las áreas a analizar según las preferencias del usuario.

Comandos disponibles

Para que el láser devuelva los datos de sus lecturas se le debe de mandar un comando específico dependiendo de la forma de lectura que deseamos que realice. Hay muchos tipos de comandos disponibles para enviar al láser

pero para los objetivos de este proyecto vamos a emplear un único tipo, los comandos AR.

Estos comandos son los que le indican al láser la orden de enviar los datos de lectura. Dependiendo del comando AR enviado, se enviará un determinado conjunto de datos u otro. Algunos de estos comandos sirven para detener el funcionamiento provocado por otros comandos del mismo tipo. Los comandos de este tipo son 6 (Los cuales se encuentran especificados en el manual del protocolo de comunicación del láser, suministrado por el tutor, accesible desde aquí `https://bit.ly/2APyvm0`):

1. AR00: envía la medición de las distintas distancias de una única lectura del entorno.

2. AR01: envía las mediciones de las distintas distancias e intensidades de una única lectura del entorno.

3. AR02: envía las mediciones de las distintas distancias de cada una de las lecturas del entorno que realiza de forma contínua.

4. AR03: sirve para detener el comportamiento continuado del comando AR02, se recibe después un mensaje de confirmación.

5. AR04: envía las mediciones de las distintas distancias e intensidades de cada una de las lecturas del entorno que realiza de forma contínua.

6. AR05: sirve para detener el comportamiento continuado del comando AR04, se recibe después un mensaje de confirmación.

Comandos	Contínuo	Único	Distancia	Distancia/Intensidad	Parada
AR00		X	X		
AR01		X		X	
AR02	X		X		
AR03					X
AR04	X			X	
AR05					X

Tabla 3.1: Características de los comandos

Envío y recepción de mensajes

Tras haber escogido el comando a utilizar, se debe construir el mensaje que se va a enviar desde el PC hacia el láser. Para esta construcción se debe de seguir una estructura específica basada en estos campos.

Host » UAM

STX	Command Size	Header	Sub Header	CRC	ETX
1 char	4 char	2 char	2 chars	4 char	1 char

Figura 3.4: Mensaje de petición al láser

- STX: Carácter, habitualmente correspondiente a "2çodificado como Bytes, que indica al láser el comienzo del mensaje que se le desea enviar.
- Command size: como su propio nombre indica, es el tamaño del mensaje que el láser va a recibir incluyendo los caracteres de inicio y fin. Para los comandos que se van a emplear en este proyecto, los 4 caracteres que representan esta parte del mensaje van a ser siempre "000Eza que siempre tienen la misma extensión de 14 caracteres.
- Header: o cabecera, es la parte del mensaje que indica el tipo de comando que se está mandando desde el ordenador. En este caso .ᴬR".

- Subheader: después de haber expresado el tipo de comando, se especifica en te campo que comando del tipo escogido se va a enviar.
- CRC (Cyclical Redundancy Checking): o verificación de redundancia cíclica, es un código que se añade al mensaje para que e láser verifique la ausencia de fallos en la creación o transmisión del mensaje recibido. Existen muchos tipos de CRC, pero en el caso del láser empleado se usa el CRC Kermit.
- EXT: Carácter, habitualmente correspondiente a "3çodificado como Bytes, que indica al láser el final del mensaje que se le desea enviar.

Tras recibir este mensaje, el láser analiza este mensaje, para comprobar que no falte ninguna de las partes descritas anteriormente y la corrección del CRC (evitando fallos de trasmisión). El mensaje de respuesta también obedece a una estructura muy estricta:

Host « UAM

STX	Reply Size	Header	Sub Header	Data*	Status	CRC	ETX
1 char	4 char	2 char	2 chars	N char	2 char	4 char	1 char

Figura 3.5: Mensaje de respuesta del láser

- STX y ETX: Son los caracteres de inicio y final del mensaje enviado pro el láser. su definición es identica a la de sus homólogos en el anterior mensaje descrito.
- Reply size: al igual que en el anterior se especifica el tamaño del mensaje en 4 caracteres que representan este valor en hexadecimal aunque a diferencia con esta, este tamaño si que es variable ya que no siempre se van a mandar la misma cantidad de datos.
- Header y Subheader: son la cabecera y el número del comando recibido por el láser en el anterior mensaje. Sirve de confirmación al usuario de la correcta comprensión del láser a cerca del comando enviado.
- Data: como indica su nombre, son los datos de las lecturas realizadas por el láser. Este campo es el único en toda la comunicación el cual no es obligatorio ya que hay casos en los que se puede omitir. Se destacan dos casos:

  - Mensaje de error: en los mensajes de error (cuando alguna de las comprobaciones no se ha verificado o la lectura realizada por el láser no se ha terminado con éxito) este campo se omite ya que no hay datos a enviar.
  - Comandos continuos: en los comandos que provocan un comportamiento de lectura del láser contínua (AR02 o AR04) se envía primero un mensaje con la respuesta que se está describiendo y después se envian mensajes sucesivos (con el mismo comienzo y fin que los otros) donde se muestran los datos.

- Status: es el estado del mensaje. A no ser que el mensaje sea un mensaje de error, el resto debería de ser "00.º similar. En caso de que sea un error, estos dos caracteres cambiarán dependiendo del error que ocurra.
- CRC: es el código de verificación que demuestra que no ha habido errores en la transmisión, de forma identica a su homólogo en el mensaje enviado en la otra dirección.

En el comando empleados en este proyecto (AR02) la forma que posee estos mensajes de respuesta es la siguiente:

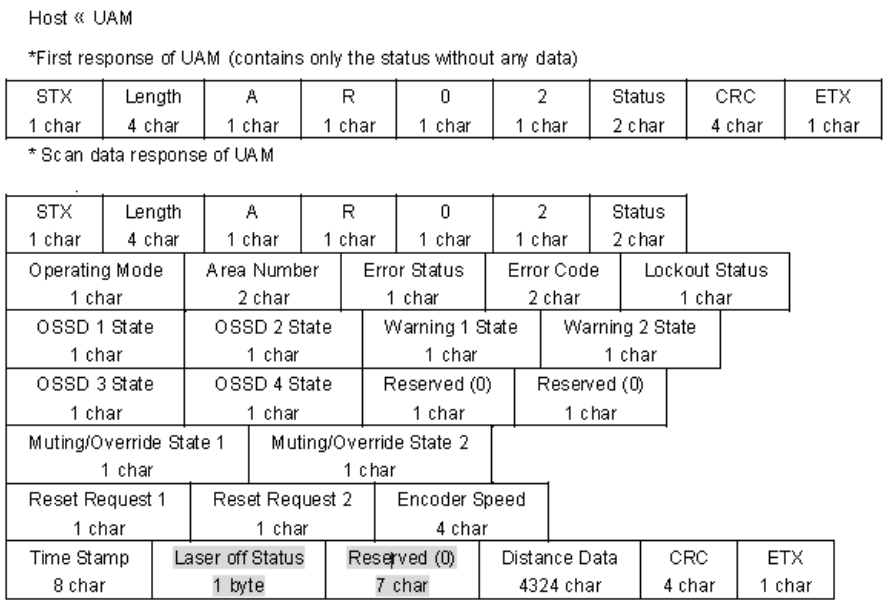

Figura 3.6: Mensaje de respuesta del láser ante comando AR02

Como se puede observar, dentro del mensaje que posee los datos de las lecturas se pueden encontrar una grán cantidad de datos que indican otra información relevante sobre la lectura realizada aunque irrelevante para los objetivos de este proyecto, como puede ser el estado de las áreas predeterminadas, los estados de alguno de los puertos de salida, el tiempo que se ha tardado en realizar la lectura, etc.

Por todo esto se debe analizar cada trama para extraer la parte en la que exclusivamente aparecen los datos, los cuales después se deben traducir.

Traducción de los datos

Después de haber aislado los datos y haberlos dividido en sus correspondientes partes (ya que cada lectura de entorno es enviada por el láser como una cadena de gran longitud) se debe traducir cada uno de los datos (que posee una longitud de 4 caracteres lo cual se deduce de los 4324 caracteres recibidos divididos entre las 1081 lecturas realizadas por cada lectura de entorno del láser) según una serie de pasos a seguir:

1. Se extra el equivalente en hexadecimal de cada uno de los 4 caracteres.

2. Si el dato hexadecimal está entre los 30_h y 39_h, se le resta 30_h. Por le contrario, si el dato se encuentra entre los 41_h y 46_h, se le resta 37_h.
3. Se traduce cada dato transformado a codificación binaria.
4. El comjunto de caracteres binarios transformados en una cadena conjunta se traduce a decimal para extraer el dato expresado en milímetros.

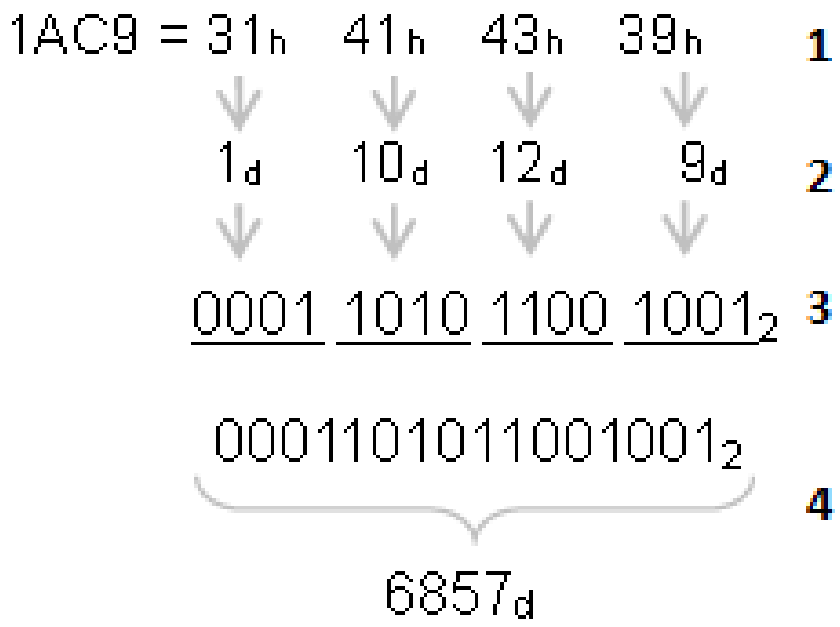

Figura 3.7: Traducción de datos a decimal

Una vez realizados estos cálculos se crea una lista con valores de ángulos expresados en grados del mismo tamaño que la lista de datos decodificados para poder asignar a cada uno de los datos su ángulo correspondiente para poder crear así una lista puntos expresados en coordenadas polares.

Esta lista de puntos es traducida elemento a elemento a coordenadas cartesianas a través de dos sencillas operaciones.Teniendo en cuenta un par de coordenadas polares (r,θ)donde rrepresenta el radio y θ representa el ángulo, la traducción a coordenadas cartesianas (x,y) se realiza así:
x = r * cos(θ)
y = r * sin(θ)

Una vez hechas estas traducciones se posee una lista de puntos cartesianos los cuales podemos representar en una gráfica para poder observar las lecturas ofrecidas por estos datos.

Con estos datos, unidos a las coordenadas de cada uno de los límites de cada una de las áreas, s realiza una comprobación para determinar si alguno de los puntos de lectura se encuentra dentro de cada área. En caso positivo, el sistema devolverá un mensaje con la posición del primero de los puntos encontrados traducida a indicaciones de lenguaje coloquial (hacia delante, hacia atrás, derecha o izquierda).

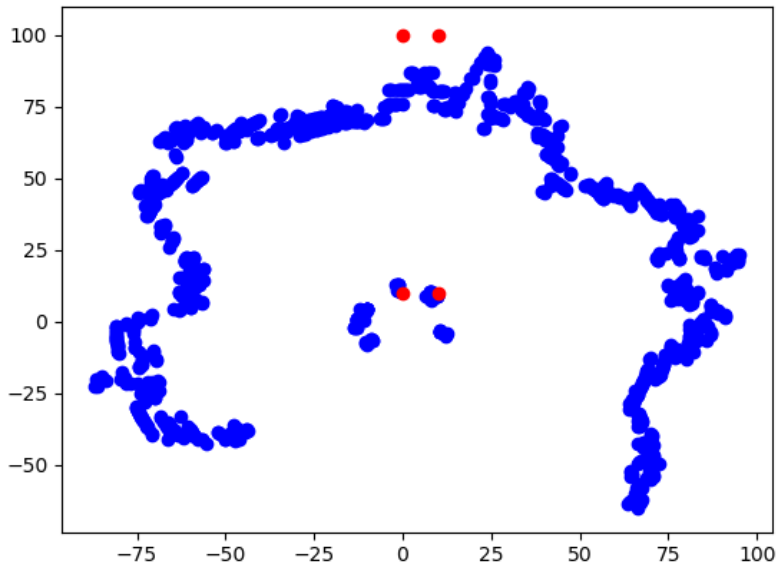

Figura 3.8: Representación de lecturas y áreas

Como se puede ver en esta imagen, el área configurada por el usuario abarca algunos de los puntos de lectura del láser, por lo tanto el sistema realizará la traducción anteriormente mencionada para indicar al usuario esa posición.

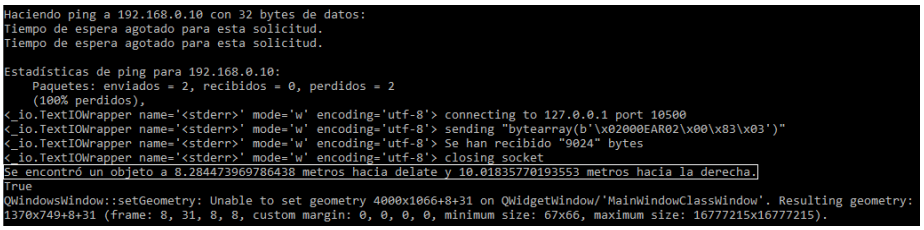

Figura 3.9: Salida por pantalla al encontrar un objeto

En caso de que en el área que establece el usuario no haya puntos, se emite el mensaje "No se encontró ningún objeto en este área". Estos mensajes aparecen en el mismo orden en el que se introdujeron las áreas.

Técnicas y herramientas

Esta parte de la memoria tiene como objetivo presentar las técnicas metodológicas y las herramientas de desarrollo que se han utilizado para llevar a cabo el proyecto. Si se han estudiado diferentes alternativas de metodologías, herramientas, bibliotecas se puede hacer un resumen de los aspectos más destacados de cada alternativa, incluyendo comparativas entre las distintas opciones y una justificación de las elecciones realizadas.

No se pretende que este apartado se convierta en un capítulo de un libro dedicado a cada una de las alternativas, sino comentar los aspectos más destacados de cada opción, con un repaso somero a los fundamentos esenciales y referencias bibliográficas para que el lector pueda ampliar su conocimiento sobre el tema.

4.1. Desarrollo de la memoria

En este aspecto se ha querido emplear LaTeX [5] en contraposición a otros programas de propósito similar como pueden ser Microsoft Word o Open Office Writer ya que esta aplicación nos permite una mayor cantidad de posibles acciones a realizar en el documento crear.

Por poner un ejemplo de estas ventajas antes mencionadas, la presentación del documento puede quedar más limpia ya que el usuario puede manejar el espacio de cada hoja a su antojo y poder disponer de todo él ya que se puede variar los márgenes de las páginas de una forma sencilla (introducción de un comando), lo que en los otros programas antes mencionados resultaría demasiado complicado a demás de ser potencialmente peligroso debido a que, gracias a las acciones automáticas que poseen, podrían desajustar todo

el contenido de la hoja en sí haciendo que el usuario tenga que ajustar cada uno de los elementos que antes poseía. LaTeX permite saltarse los márgenes establecidos, por ejemplo, para insertar una imagen. en cuanto a los encabezados, pies de página y numeración de página también se puede realizar con un simple comando.

Con todo esto y con la característica de que permite guardar el archivo directamente en PDF sin realizar ninguna conversión (aunque para ello necesite crear algún archivo debido a la compilación) es por la que se ha escogido para este

4.2. Desarrollo del código

Jupyter Notebook

Debido a que este entorno de desarrollo ya se ha empleado en algunas asignaturas del grado, lo cual lo hace más amigable y facil de usar, se ha decidido emplear Jupyter Notebook para el desarrollo del código. Además de ser compatible con el sistema operativo, este entorno tiene la ventaja de poder obtener la conexión con el láser, y por lo tanto el intercambio de mensajes y datos con el mismo, sin necesidad de instalar ningún complemento para poder desarrollar el código sin dificultades añadidas.

Este entorno, además de las ventajas antes descritas, permite desarrollar y ejecutar (y por lo tanto testar) partes del código por separado. Por este motivo, es más fácil comprobar el correcto funcionamiento del código desarrollado antes de realizar otras partes del código siendo después más difícil rectificar dicho error.

Spyder

Este programa, desarrollado también por Anaconda (misma empresa que desarrolla Jupyter) también ha sido empleado a la hora de crear el código del proyecto ya que, además de no usar un soporte web para funcionar (lo que lo hace más eficiente) permite al usuario poder ejecutar archivos con extensión .py (propios de Python), lo cual permite desarrollar y probar código que otros usuarios pueden utilizar en sus propias plataformas, al contrario que Jupyter el cual usa archivos de extensión exclusiva para dicho programa.

Eclipse

Este programa fue el elegido en las primeras fases del proyecto para el desarrollo del código debido a muchos factores.
Eclipse es uno de los programas más utilizados en el grado para el desarrollo del código en muchas de las asignaturas, lo que hace que su forma de uso y sus características sean amigables y fáciles de comprender debido a al experiencia obtenida.
También es un programa que permite desarrollar código en una amplia variedad de lenguajes, entre los cuales se encuentra aquel que fue elegido para desarrollar en esas fases del proyecto, C++ .

4.3. Planteamiento de las tareas

Trello

Para organizar las actividades a realizar se ha escogido la aplicación Trello. Esta aplicación se ha escogido debido a su facilidad de uso (debido a que su interfaz de usuario es muy intuitiva), a demás de otros muchos aspectos como el hecho de que a un tablero (unidades organizativas en las que se gestionan las tareas de los diferentes proyectos que se puede gestionar desde una misma cuenta de usuario) se puede acceder más de un usuario. Esta característica permite que tanto el alumno como el profesor pueden acceder al mismo tablero para gestionar las tareas a realizar.

Figura 4.10: Interfaz de Trello

4.4. Metodología de gestión y herramientas asociadas

Para este proyecto se ha decidido utilizar la metodología de SCRUM. Esta metodología esta basada en entregas incrementales pero funcionales. Para la realización de esta metodología, perteneciente a las denominadas metodologías ágiles, se va a utilizar la aplicación ZenHub. Esta aplicación se puede emplear para presentar los sprints y se puede planificar la fecha de comienzo y final cada una de las tareas.

4.5. Herramienta de manejo de láser

UAM Proyect Designer

Aunque el manejo de este elemento del proyecto se va a realizar a través de funciones presentes en Eclipse y el código implementado para el proyecto, se va a utilizar una aplicación instalada vía CD llamada UAM Proyect Designer. Esta aplicación, cuando el equipo en el que se instala está conectado al láser, permite observar los datos relacionados con la nube de puntos creada por este aparato al realizar sus operaciones de lectura de su entorno.

En este caso esta herramienta será utilizada para la comprobación de que el código desarrollado para alcanzar los objetos del proyecto funciona de forma correcta comparando los resultados obtenidos por el código y la aplicación, los cuales deberían de coincidir.

Como bien se ha dicho al principio de esta sección, la aplicación se ha instalado utilizando el CD presente en el paquete en el que el láser se encuentra almacenado.

RealTerm

Este programa, consistente en una terminal para TCP y puerto serie (TCP/Serial port terminal) consiste en, como suu propio nombre indica, transmitir mensajes via TCP a través del puesto serie. Su funcionamiento principal consta de 3 fases:

- Elegir la forma de transmisión de mensajes entre los dispositivos conectados por este puesto (duplex, full duplex...).

- Configurar los parámetros como la frecuencia, el puesto, la dirección IP....
- Escribir el mensaje a transmitir separando por comas (',') cada elemento y escoger entre la transmisión como número o como carácter ASCII

4.6. Lenguajes de programación

C++

Es uno de los lenguajes de programación más antiguos que existen. Surgido como modificación del lenguaje C para adaptarlo a la programación orientada a objetos, es uno de los lenguajes más versátiles ya que gracias a sus diferentes versiones, en la más actual hasta el momento (C+ + 17) este lenguaje posee facilidades de programación genérica, programación estructurada y programación orientada a objetos.[2]

Es por esto por lo que este lenguaje es uno de los más empleados en el mundo industrial y por lo que se decidió escoger en las primeras fases de este proyecto como lenguaje para el desarrollo del código.

Por otra parte, a pesar de estas ventajas, es te lenguaje tiene sus inconvenientes. Uno de los más relevantes es la dificultad que presenta la configuración de algunos procesos. Esto, unido a la poca experiencia que se tenía en el desarrollo de código con este lenguaje en contraposición con otros, fue lo que propició la decisión de cabiar de lenguaje para pasar a utilizar otro con el que se tuviese más experiencia.

Python

El lenguaje de programación con este nombre es aquel desarrollado en 1991 con el objetivo de crear código con una sintaxis que favorezca un código legible.
Además de esta característica, es importante destacar el hecho de que se trata de un lenguaje multiparadigma, ya que puede desarrollarse código orientado a objetos, programación imperativa y, en menor medida, programación funcional.[3]

Ha sido seleccionado debido a la sencillez que proporciona a la hora de programar distintos procedimientos que en otros lenguajes (Java, C,C++

...) resultarían mucho más complicados como es la implementación de un socket para comunicarse (enviar mensajes y recoger datos) con el láser.

4.7. Cableado

Para que el proyecto pueda ser desarrollado y ejecutado para la comprobación de su correcto funcionamiento es necesario suministrar corriente eléctrica al láser y comunicarlo de forma física con el PC. Para ello se han tenido que buscar dos cables diferentes:

- Comunicación col el PC: para este aspecto del proyecto se ha empleado un cable ethernet 100BASE-TX con conexión resistente al agua, el cual es desarrollado por la misma empresa que desarrolla el láser empleado lo cual lo hace idóneo para comunicar ambos dispositivos.

- corriente eléctrica: para este aparato se han utilizado un adaptador a corriente alterna ya que este láser necesita alimentarse con 24 V de este tipo de corriente. Este cable ha sido buscado y encontrado por el alumno.

Aspectos relevantes del desarrollo del proyecto

5.1. Definición del trabajo como proyecto de investigación

Un aspecto que hay que destacar de este proyecto es que su enfoque, además de ser el de crear un sistema funcional que cumpla los requisitos establecidos en puntos anteriores, es el de crear una simulación del entorno en el que posteriormente se implementará este sistema, buscando los elementos tanto hardware como software (programas, lenguajes, librerías, recursos...) necesarios para que un sistema ya implantado en su entrono de trabajo pueda utilizar el sistema desarrollado para realizar algunas de sus tareas. Como resumen, se podría decir que es un trabajo que pretende desarrollar un sistema para la investigación de la implementación de un nuevo sistema de detección de objetos.

5.2. Conocimientos utilizados aprendidos en el grado

Algunos de los conocimientos que se han empleado en este proyecto se han adquirido a lo largo de las diferentes asignaturas de los 4 cursos del grado.

Codificación en Python

El conocimiento principal que ha permitido desarrollar el código funcional de este proyecto es el conocimiento a cerca de la creación de código en lenguaje Python. Gracias a algunas asignaturas como son Sistemas Inteligentes (1^{er} semestre del 3^{er} curso), Algoritmia (2^{o} semestre del 3^{er} curso), Nuevas Tecnologías y empresa y Minería de Datos (ambas del 2^{o} semestre del 4^{o} curso).

Gracias a haber recibido clases de estas asignaturas se han podido conocer y aprender a manejar algunas de las librerías que se han empleado en la creación de este sistema software, como pueden ser Numpy, Matplotlib o Math, además de los conocimientos a cerca de las características principales de este lenguaje de programación.

Conocimiento sobre sockets

Aunque en este sistema se ha tratado este asunto con poco detalle, el manejo de sockets también es un concepto que se ha sabido emplear gracias a alguna asignatura del grado. Un ejemplo de estas asignaturas es Sistemas Distribuidos (2^{o} semestre del 4^{o} curso).

Gracias a los conocimientos adquiridos en esta asignatura se ha podido configurar y gestionar el comportamiento del socket lógico creado para la comunicación entre el láser y el ordenador.

Conocimiento a cerca de la gestión del proyecto

Este aspecto es uno de los más relevantes ya que es la parte del proyecto la cual ayuda a gestionar el tiempo que se usa para llegar al objetivo marcado por los requisitos del proyecto. Los conceptos de la definición, aspectos y formas de aplicación de las metodologías ágiles, SCRUM en el caso de este proyecto, han sido adquiridos gracias a haber recibido las clases de la asignatura de Gestión de Proyectos (1^{er} semestre del 3^{er} curso).

5.3. Conocimientos externos a lo aprendido en el grado

También existen conocimientos aprendidos gracias al desarrollo de este proyecto los cuales no se han adquirido a lo largo del grado.

Codificación de mensajes

En cuanto a este proyecto, se ha de destacar uno solo de estos conocimientos, la creación de los mensajes. Como ya se ha explicado en apartados anteriores, para comunicarse con el láser, el PC debe enviar una serie muy concreta de comandos, los cuales deben de temen una codificación específica. Esta codificación se basa en que algunas de las partes del mensaje (inicio y final del mensaje y CRC) se deben enviar codificadas como Bytes mientras que el resto deben ser enviados como código ASCII. Ninguna de estas codificaciones ha sido impartida en ninguna de las asignaturas que imparten código en Python mencionadas en el punto anterior.

Estos conocimientos tienen la ventaja de que, al ser conceptos sobre uno de los lenguajes de programación más usados, es sencillo encontrar mucha información al respecto por internet, por lo que se pueden encontrar diferentes formas de obtener el mismo resultado (utilizando diferentes librerias o creando métodos con las librerías que están por defecto en el propio lenguaje) con lo que el programador puede escoger la que le parezca más idónea para su propósito.

Lectura de pinout del láser

Debido a que en el grado no se imparte ninguna asignatura obligatoria (ya que solo existe Mantenimiento de Equipos Informáticos en el 1^{er} semestre del 4º curso y es una asignatura optativa) la cual tenga el hardware como base de los conocimientos impartidos en la misma, se ha obtenido conocimiento a cerca de este aspecto gracias a la necesidad de configuración del sistema a nivel físico.

En este proyecto se ha necesitado estudiar y comprender el pinout del láser con el objetivo de poder usarlo. Se define pinout como el conjunto de cables que posee un sistema para conectarse con otros elementos del

exterior. En el caso que se ocupa aquí, se ha necesitado conocer la función de cada uno de estos cables para saber cuales de ellos son los que abastecen de corriente eléctrica al láser, los cuales se empleará par conectarse con el cargador fabricado y poder tener el láser plenamente operativo.

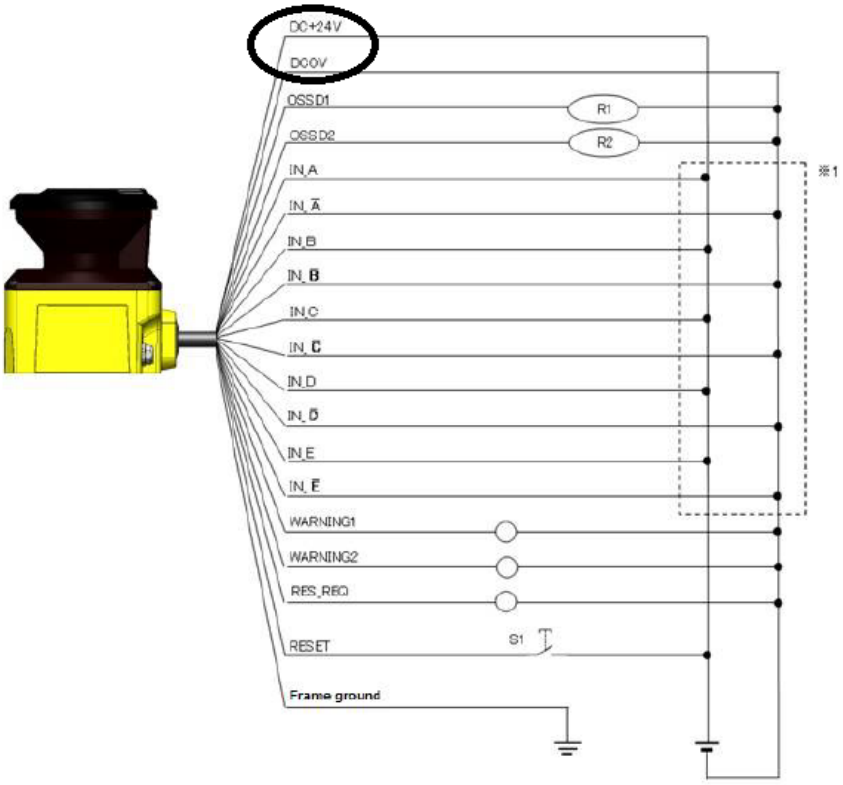

Figura 5.11: Conjunto cables (pinout) del láser.

Tras observar esta imagen se puede observar que son los dos primeros cables (descritos como DC 24V y DC 0V) los que se necesitan para el propósito antes descrito. Después de haberlo descubierto solo se tuvo que localizar los cables en el láser (de colores marrón y azul respectivamente) y conectarlos al cargador.

Conocimientos sobre el láser

Además de lo expresado en la sección anterior, también se ha adquirido conocimiento a cerca de otros aspectos del láser, los cuales, debido a los

mismos motivos que en el punto anterior, no se han dado a lo largo del grado.

En este desarrollo se ha aprendido las diferentes formas con las que un operador se puede comunicar con el láser además de los diferentes procedimientos previos que se deben realizar para poder usar cada una de estas formas de conexión. Estos procedimientos son muy distintos dependiendo de la forma de conexión:

- USB: Para poder realizar esta conexión se necesitó previamente instalar los drivers necesarios (estos se pueden encontrar en el repositorio en la carpeta *Drivers*), los cuales no eran drivers firmados por lo que, de forma temporal se habilitó el uso de este tipo de archivos en Windows para poder instalarlos para, tanto poder usar este driver como para evitar la instalación de divers maliciosos (de ahí que al volver a encender el ordenador se pueda seguir usando este archivo específico pero no otros drivers no firmados).
- Ethernet: con este tipo de conexión los preparativos son más sencillos ya que el único paso a realizar es el cambio de dirección IP para hacer que el láser y el PC se encuentren en la misma red lógica para poder comunicarse.

Trabajos relacionados

Los AGVs existen desde 1953, cuando su inventor diseñó un remolque que transportaba comestibles hasta un almacén recorriendo el camino marcado cpn una cable de acero el cual tiraba de él hasta su destino.

Tras muchos años de innovación se han ido mejorando las estructuras de estos robots así como los elementos incorporados para su funcionamiento. Uno de los aspectos en los aspectos en los que más ha mejorado es en los sistemas de guiado. En la actualidad existen infinidad de sistemas de guiado para estos sistemas y uno de los más utilizados es el guiado por láser.

Este sistema de guiado se basa en la detección de placas reflectantes con las que el AGV es capaz de orientarse para recorrer su camino gracias al análisis de los datos recibidos del láser e identificar dichas placas.

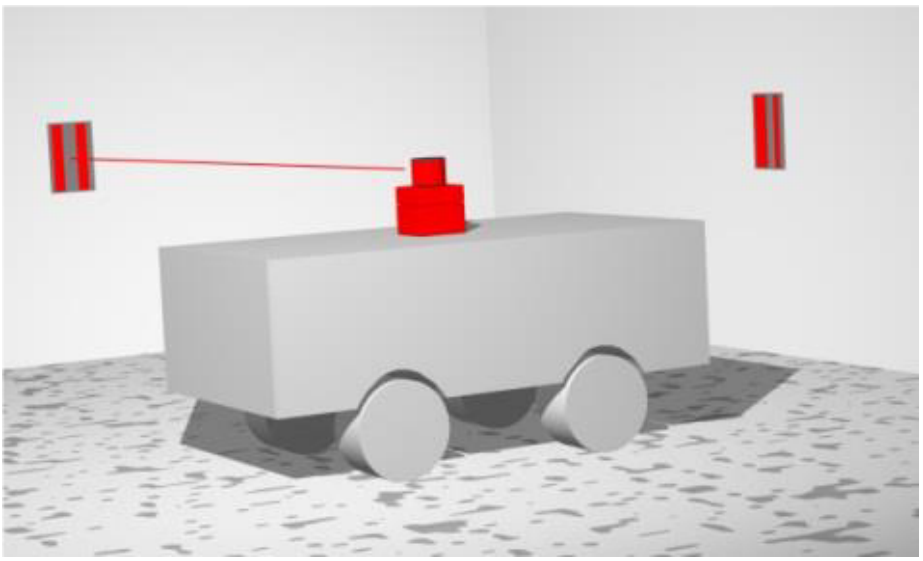

Figura 6.12: AGV guiado por láser

Otra de las aplicaciones en las cuales se está aplicando la tecnología láser es en la automovilística donde se está empezando a aplicar para sistemas de frenado de seguridad en cuanto a la medida de distancias.

Figura 6.13: Medida de la distancia de seguridad vía láser

Un proyecto no oficial es el desarrollado por el usuario de la plataforma GitHub iliasam. Este trabajo es en realidad mucho más mecánico, ya que se basa en la creación de un PCB en el cual se va a implementar un sistema de rotación y se va a conectar in lidar, el cual irá realizando lecturas para crear un mapa de puntos del entono detectado.

Para poder obtener más información a cerca de este proyecto hay que visitar la URl: `https://github.com/iliasam/OpenSimpleLidar`

Conclusiones y Líneas de trabajo futuras

Todo proyecto debe incluir las conclusiones que se derivan de su desarrollo. Éstas pueden ser de diferente índole, dependiendo de la tipología del proyecto, pero normalmente van a estar presentes un conjunto de conclusiones relacionadas con los resultados del proyecto y un conjunto de conclusiones técnicas. Además, resulta muy útil realizar un informe crítico indicando cómo se puede mejorar el proyecto, o cómo se puede continuar trabajando en la línea del proyecto realizado.

7.1. Introducción

En este apartado se expondrán las conclusiones derivadas del desarrollo del proyecto y de los resultados finales obtenidos. Por otra parte, se expondrán una serie de posibles mejoras y líneas de trabajo futuras con respecto a este proyecto.

7.2. Conclusiones

Finalizando la creación de esta memoria, se expondrán las conclusiones obtenidas del desarrollo de este proyecto:

- Como primera de las conclusiones se ha de destacar la parte del proyecto que más ha costado conseguir o más bien la que más ha costado de conseguir de forma estable,el entorno de programación. Esta parte es una de las que más tiempo ha llevado si no es la que

más debido a los problemas surgidos debido a programas como Eclipse, el cual fallo impidiendo continuar el proyecto durante un periodo de tiempo muy extenso, al igual que el uso de sistemas virtualizados como es el caso del uso del sistema operativo Ubuntu en el programa Oracle VirtualBox que al no encontrar solución posible para el problema encontrado se tuvo que rehacer varias veces. Aunque haya supuesto un problema, ha servido también para conocer estas aplicaciones en profundidad y poder investigar para solucionar estos fallos o similares en un futuro.

- Otra conclusión a tener en cuenta es la cantidad de nuevos conocimientos obtenida gracias al desarrollo de este software. Se ha obtenido nueva información a cerca de comunicación entre dispositivos, sockets, formas de codificación de diferentes conjuntos de datos, diferentes formas de implementación de un sistema para obtener los mismos resultados (ya sea variando los elementos hardware o software utilizados), las ventajas y desventajas que supone crear una determinada parte del sistema dependiendo del lenguaje de programación utilizado para ello, etc.

- Y hablando de las investigaciones, este proyecto ha servido también para conocer como realizar un proyecto software basado en investigaciones a cerca de temas de los que se tenía un idea muy superficial pero se ha podido descubrir más aspectos los cuales han servido para aumentar conocimientos y hacer saber la dinámica llevada a cabo en proyectos de investigación.

En resumen, la experiencia en el desarrollo de este proyecto ha sido por lo general satisfactoria ya que, aún siendo una actividad realmente costosa en algunos aspectos, ha sido útil para dar a conocer la realidad sobre el proceso que este tipo de proyectos siguen para llevarse a cabo y conocer también las facilidades y dificultades que estos suponen para las personas encargadas de su desarrollo.

7.3. Mejoras y líneas de trabajo

Mejora del simulador

Una de las últimas partes desarrolladas de este proyecto fue la creación de un servidor para poder simular el uso del láser cuando el usuario que desea utilizar el software no posee o no tiene acceso a un láser.

Esta parte del sistema software es una parte en la que se puede extraer una linea de trabajo debido a la cantidad de mejoras que se le pueden realizar.Las más destacadas son:

- Información a enviar: en la versión actual, el servidor posee una variable en la cual almacena lecturas que se han recogido de lecturas realizadas por el láser en ejecuciones del sistema principal. Esta variable (de tipo lista, *list()* en Python) es recorrida de forma continua y transmite una de sus cadenas de lectura por cada mensaje que recibe de cada cliente. Esta es una manera muy poco eficiente además de no ser la forma que el láser tiene de funcionar. Por este motivo una de las mejoras a realizar en el servidor es la implementación de un método o métodos con los cuales se puedan generar de forma automática datos de lectura aleatorios.

- Seguridad: otra de las características de la versión actual de esta parte del proyecto es la nula seguridad a la hora de recibir los mensajes, es decir, el usuario puede enviar cualquier tipo de mensaje (independientemente de la estructura del mismo) y el servidor le va a seguir devolviendo una lectura de su lista. En una futura versión se puede implementar un sistema que, al igual que hace el láser, compruebe que el mensaje está formado por todos los campos, cada uno de ellos con su contenido correcto (longitud, valor). En esta futura versión el servidor podría calcular el CRC para poder compararlo con el recibido en el mensaje.

Ejecución remota

Esta mejora viene motivada por, como ya se ha explicado en otros apartados, el hecho de no haber conseguido implementarlo durante el desarrollo de este proyecto.

Para el desarrollo de este aspecto se necesita la inclusión en el sistema del MTX-GTW [**?**]. Este instrumento es capaz de simular los recursos, y por tanto el rendimiento, presentes en un AGV. Para comunicarse con él se debe usar varios cables adicionales:

- Cable de red: para realizar la conexión entre el MTX-GTW y el ordenador se necesita un cable de red estándar.

- Cable con conector RS232: este cable es necesario para poder conectar el nuevo dispositivo con el láser.
- Cable de alimentación: se necesitará tambien un cable que suministre 12 V de conrriente continua sin cabezal ya que posee uno propio al cual se han de acoplar dos hilos individuales.

Tras haber obtenido este sistema hardware se ha de comprobar si en el MTX-GTW se puede ejecutar código en Python, ya que si no es así sería necesario traducir el sistema a lenguajes como C++, el cual si que es capaz de comprender. En la versión de este dispositivo utilizada en las primeras fases de este proyecto, si que es capaz de reconocer y ejecutar código Python pero si se desea utilizar otra versión u otro equipo hay que leer su manual para saberlo.

Prueba en un AGV

Otra de las mejoras o trabajos derivados que pueden surgir de esta versión del software es la implantación de este software en un AGV real. Este sistema, como ya se ha descrito, detecta objetos y los localiza, lo cual puede ser usado por un AGV para su funcionamiento más habitual. En la actualidad, gracias a la tecnología láser, los AGV pueden detectar si en su entorno se encuentras algún objeto, aunque no pueden saber en que parte del entorno exactamente. Con la implantación del sistema de este proyecto en el AGV se podría ver y testar si actúa de una manera más eficiente, es decir, si puede detectar la posición del objeto y continuar con su funcionamiento habitual si esa posición no se lo impide.

Para poder implementar estas pruebas se puede probar a implementar este sistema en un microprocesador, el cual irá conectado más tarde al vehículo y gestionará el análisis de estos datos. Esta implementación hará que el trabajo de análisis de datos sea ejecutado por este procesador, lo cual liberará de trabajo al procesador principal del AGV, lo que supondrá una mejora en la eficiencia del sistema completo de estos vehículos.

Bibliografía

[1] ASTI Automatismos y Sistemas de Transporte Interno, S.A.U. Agv, los vehículos industriales inteligentes. https://bit.ly/2FIbCUN, 2010 (accessed September 25, 2018).

[2] cplusplus. The c++ resources network. http://www.cplusplus.com/, 2000(accessed September 25, 2018).

[3] Python Software Fundation. Python sofware fundation. https://www.python.org/, 2001(accessed September 20, 2018).

[4] hokuyo Automatic CO. LTD. *Safety Laser Scanner UAM-05LP User's Manual*, 2015 (accessed October 3, 2017).

[5] Wikipedia. Latex — wikipedia, la enciclopedia libre. https://en.wikibooks.org/wiki/LaTeX, 2015. [Internet; descargado 30-diciembre-2017].