

TELÉMETRO LASER MOTORIZADO



Carlos Fernández Bailador

17/10/2018

Índice

1	INTRODUCCIÓN.....	5
1.1	Entorno y motivación	5
1.2	Precedentes	5
1.3	Objetivo	6
2	PROPUESTA DE DISEÑO Y DESCRIPCIÓN DEL SISTEMA ..	6
2.1	Especificaciones del sistema	6
2.2	Propuesta de diseño, diagrama de bloques	7
3	FUNDAMENTOS TEÓRICOS	8
4	DESARROLLO DEL PROYECTO.....	9
4.1	Bloque de control.....	9
4.1.1	Objetivo.....	9
4.1.2	Especificaciones	9
4.1.3	Alternativas	10
4.1.4	Desarrollo	11
4.2	Bloque actuadores.....	12
4.2.1	Objetivo.....	12
4.2.2	Especificaciones	13
4.2.3	Alternativas	13
4.2.4	Desarrollo	15
4.3	Bloque de sensores	16
4.3.1	Objetivo.....	16
4.3.2	Especificaciones	16
4.3.3	Alternativas	16
4.3.4	Desarrollo	18
5	MONTAJE Y PUESTA EN FUNCIONAMIENTO	20
5.1	MONTAJE.....	20
5.1.1	FUNCIONAMIENTO.....	23

CONCLUSIONES.....23

WEBGRAFÍA.....25

AGRADECIMIENTOS

En primer lugar quiero agradecer a mis gatos su habilidad para “Testear” la dureza de las piezas impresas. También agradecer al tutor de este curso, Joan, por su habilidad para enseñar y transmitir conocimientos de manera sencilla y paciente y su habilidad de adaptación a la persona. Por último agradecer a mis compañeros por enseñarme conocimientos de sus respectivas materias y mostrarme sus puntos de vista.

RESUMEN

Este proyecto se basa en diseñar, montar y programar un sistema combinado que permita obtener mediciones sobre los planos X,Y y Z. Las mediciones obtenidas pueden ser empleadas con una gran diversidad de fines, siendo las más comunes : Autolocalización de vehículos no tripulados, escaneo 3D de entornos, posicionamiento en aplicaciones de robotica, etc...

Para montar un sistema de escaneo 3D con movimiento mediante motores se requieren ciertos componentes que por motivos de tiempo no se han podido conseguir, tales componentes se muestran en el apartado:

Debido a la imposibilidad de montar el sistema de escaneo 3D a tiempo para la entrega, se ha optado por montar un sistema que obtenga medidas en solo dos planos X y Y y sin rotación continua

1 INTRODUCCIÓN

1.1 Entorno y motivación

La idea parte de un proyecto propio en desarrollo el cual se ve beneficiado por la realización del curso de programación e impresión 3d, tal proyecto requiere diseñar e imprimir varias piezas, de modo que la elección de este proyecto da lugar a la puesta en práctica de los conocimientos adquiridos en este curso.

1.2 Precedentes

Actualmente existen sistemas profesionales LIDAR (*Laser Imaging Detection and Ranging*) compactos que efectúan medidas rápidas y precisas pero también existen sistemas creados en el ámbito personal empleando tecnologías al alcance de todos. Algunos ejemplos son los mostrados a continuación:

- <https://www.youtube.com/watch?v=gCpCGkwwy8I&t=9s>

- <https://www.youtube.com/watch?v=RHPAJ3kjc3Q>
- <https://www.youtube.com/watch?v=o3Yh3oAf-KQ>

1.3 Objetivo

Los objetivos son:

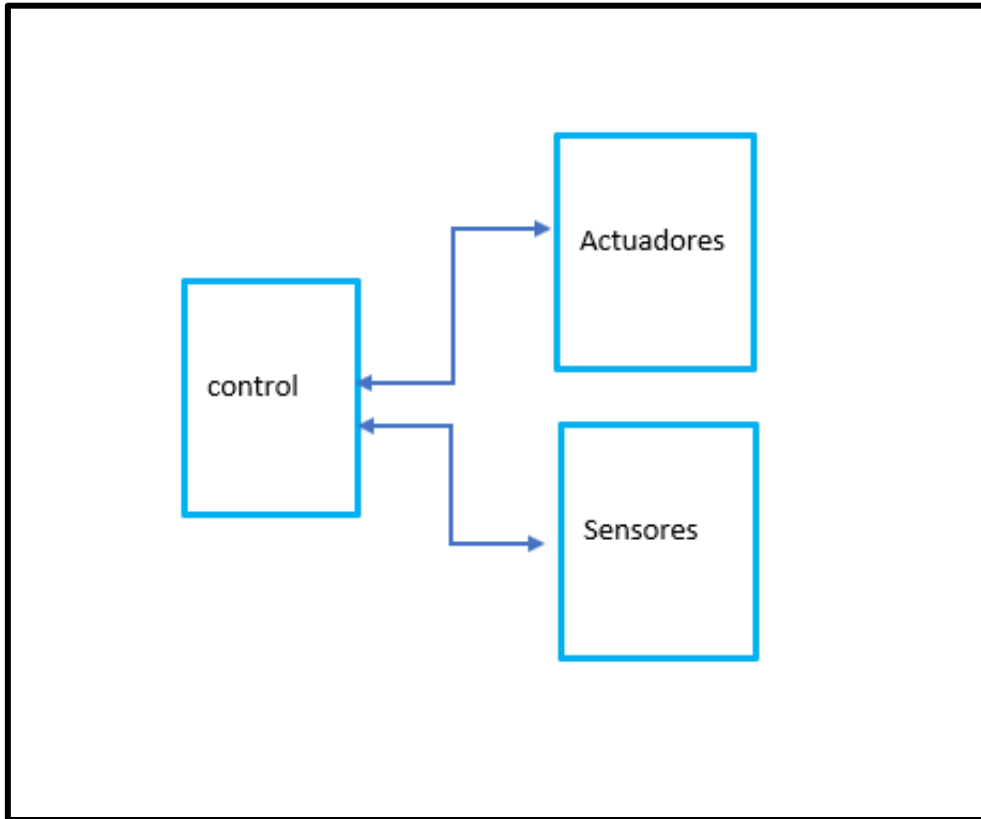
- Logar el movimiento del Telémetro laser mediante un motor paso a paso, debido a que este tiene una mayor resolución en cuanto a grados que un servomotor, además de la posibilidad de rotar continuamente y no solo 180 grados, limitación propia de los servos.
- Diseñar e imprimir las piezas necesarias que conformaran los soportes de los componentes del sistema motorizado LIDAR.
- Aprender y llevar a cabo el montaje del sistema del motor, es decir, conexionado, regulación del driver y control del motor.
- Aprender y elegir el método mas adecuado de mediciones que ofrece el telémetro empleado en este proyecto.
- Crear un código que implemente el sistema de control del motor y el sistema de mediciones del telémetro.

2 PROPUESTA DE DISEÑO Y DESCRIPCIÓN DEL SISTEMA

2.1 Especificaciones del sistema

- Alimentación de 12v en parte de potencia y 5v en parte de potencia y control.
- Control mediante microcontrolador programable.
- Control del motor mediante driver externo al microcontrolador.
- Soportes fabricados en material PLA
- Resolución en pasos del motor para posicionamiento: pasos de 0.9° en full steep o movimiento en micropasos, dividiendo cada paso en hasta 32 micropasos.
- Mediciones de hasta 40 metros a una velocidad de 500 hz.

2.2 Propuesta de diseño, diagrama de bloques



3 **FUNDAMENTOS TEÓRICOS**

Hay varias formas de controlar el motor y el telemetro.

Control motor:

- **Control mediante driver externo**

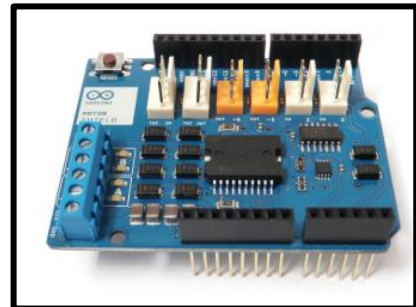
Existen varios modelos de driver en el mercado, siendo los mas comunes el A4988 y el DRV8825, ambos muy similares en cuanto a funcionamiento, arquitectura, programación y coste.

La principal diferencia reside en que el A4988 ofrece una resolución de hasta 16 micropasos mientras que el DRV8825 ofrece hasta 32, de modo que por la poca diferencia de coste se suele emplear este último.



- **Control mediante shield motor.**

Existe la posibilidad de controlar un motor paso a paso mediante un shield de control para motores acoplado directamente en Arduino, estos tipos de Shields ofrecen una solución cómoda y rápida para el control de motores de corriente continua y servos pero para el control del motores paso a paso prescinden de algo fundamental como es la posibilidad de control mediante micropasos.



Control telémetro:

- **Control mediante comunicación I2C.**

Este método establece una comunicación serie entre el microcontrolador y el telémetro que permite la transmisión de ordenes del microcontrolador y el envio de mediciones del telémetro. De los dos métodos de comunicación que permite el telémetro este es el que permite un funcionamiento mas rápido.

- **Control mediante PWM.**

Este método permite la comunicación mediante ancho de pulso entre el telemetro y el microcontrolador. El telemetro emite una señal de ancho de pulso proporcional a la distancia medida.

4 DESARROLLO DEL PROYECTO

4.1 Bloque de control

4.1.1 Objetivo

El objetivo del presente bloque, es enviar las ordenes de movimiento al motor, enviar las ordenes de medición al telemetro y almacenar y tratar los resultados obtenidos de las mediciones.

4.1.2 Especificaciones

El controlador seleccionado debe contar con las siguientes características:

- 4 entradas mínimo de las cuales deben ser digitales.
- 3 salidas mínimo, las cuales deben ser digitales
- 2 pines mínimos de Rx/Tx
- 2 pines mínimos de comunicación SDA/CLK
- Tensión de alimentación menor o igual a 12V.
- Tensión de entrada (límites): +6 a + 20 V.
- Comunicación serie I2C.
- Frecuencia de reloj mínima 100kHz
- Dimensiones inferiores a: 1" x 2".

4.1.3 Alternativas

Para el control de este proceso hay varios controladores aptos, de entre ellos se destacan:

- ***Controladores tipo PLC***

Aunque cuentan con recursos de sobra para este control, se han Descartado debido al coste y tamaño.

Algunos ejemplos son:

- SIEMENS S7 1200 CPU 1214DC/DC/DC 14DI, 2AI, 10DQ
- Allen-Bradley MICRO800, módulos enchufable E/S (20 output, 20 input y 2 canales AI)

- ***Raspberry Pi***

Este controlador permite el control del sistema en cuestión pero respecto al control del telemetro existe un volumen de documentación mas amplio respecto a su control mediante arduino, a demás de la simplicidad del control de este mediante Arduino.

La solución más eficiente y optimizada tanto a nivel económico como de dificultad técnica es Arduino, debido a su pequeño coste y facilidad de interconexión con otros módulos y por supuesto, cumple sobradamente con las especificaciones.

Especificaciones técnicas Arduino nano.

Microcontroller	Atmel ATmega328
Operating Voltage (logic level)	5 V
Input Voltage (recommended)	7-12 V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	8
DC Current per I/O Pin	40 mA
Flash Memory	32 KB (of which 2KB used by bootloader)
SRAM	2 KB
EEPROM	1 KB
Clock Speed	16 MHz
Dimensions	0.70" x 1.70"

4.1.4 Desarrollo

Debido a problemas en la puesta en práctica del sistema no se ha podido elaborar el código optimizado todavía, el problema es el siguiente:

El movimiento de un paso a otro del motor se realiza enviando pulsos de la siguiente manera:

El bucle FOR se repite tantas veces como el valor de la variable steps, esta variable corresponde a los pasos que queremos que haga el motor.

En cada ejecución de este bucle, se dan las siguientes ordenes: -avanzar un paso estableciendo un pin vinculado al motor en HIGH, -esperar unos microsegundos determinados para dar tiempo a que el motor avance un

```
for (int x = 0; x< steps ; x++) {    //Moverme paso a paso y coger medidas_iniciales
  digitalWrite(stepPin, HIGH);
  delayMicroseconds(microPausa);
  digitalWrite(stepPin, LOW);
  delayMicroseconds(microPausa);

  dist = lidarLite.distance(false); // Without bias correction

  medidas_iniciales[x]=dist;
}
```

paso, -Poner el pin en estado LOW para poder emitir otro pulso a continuación, -Esperar unos microsegundos determinados para no solapar los pulsos.

Al ejecutar este código el movimiento del motor es rápido y fluido, el problema viene al incluir la orden de medición del telemetro, correspondiente a una función propia de la librería del telemetro.

Al incluir la línea de código correspondiente a la medición del telemetro se provoca un retraso mínimo en la ejecución del bucle debido a la ejecución de la función de medición. Este mínimo retraso provoca que el movimiento de un paso a otro del motor no sea tan continuo como debería dando lugar a un sonido de traqueteo del motor, este suceso no es un problema para el funcionamiento ya que el movimiento sigue siendo rápido y aceptable para la toma de medidas pero al no ser un funcionamiento optimizado se ha

optado por “investigar” los modos de funcionamiento del telemetro y encontrar el adecuado que solventa este error.

El telemetro cuenta con varios métodos de toma de medias, estos métodos están orientados según el uso que se le quiera dar al telemetro, por ejemplo, uno de los métodos efectúa 100 medidas y tras un filtrado de estas devuelve un resultado, este método ofrece una medición totalmente blindada a costa de tardar más tiempo, otro método por ejemplo ofrece mediciones a máxima velocidad a costa de un menor filtrado de errores.

El problema en cuestión es que el código probado hasta ahora corresponde a un código de ejemplo que efectúa un tipo de medición “lenta”. La manera de resolverlo será determinar cual es el mejor y método de medición ofrecido en las librerías del telemetro e implementarlo.

4.2 Bloque actuadores

4.2.1 Objetivo

El bloque de actuadores se compone básicamente del motor, tiene el objetivo de permitir el movimiento rotacional del telemetro, permitiendo de esta manera efectuar mediciones en distintas direcciones.

4.2.2 Especificaciones

Con respecto a este bloque se precisa un motor preciso y relativamente rápido capaz de posicionarse en distintos grados, los requerimientos son los siguientes:

- Funcionamiento a corriente continua de 12v
- Alta resolución de pasos.
- Bajo valor de inercia.

4.2.3 Alternativas

Las alternativas existentes más viables son las siguientes:

Motor paso a paso

- Alta resolución de pasos.
- Tensión nominal 12 v cc.
- Alto grado de precisión y repetibilidad.

Ejemplos:

- Motor nema 17.

Servomotor

- Resolución de hasta 270 grados.
- Tensión nominal de 6 a 9 v cc.
- Alto torque y control simple.

Ejemplos:

- DSSERVO DS3218MG

Motor cc con encoder

Esta es otra alternativa a los servos y a los motores paso a paso pero implica un nivel del control mas avanzado y sin profundizar en su control se aprecia un control mas simple y eficaz en los otros motores normbrados.

De entre las alternativas estudiadas escogemos el motor paso a paso NEMA 17, el cual consta de las siguientes especificaciones:

Model: NEMA17 Stepper Motor - JK42HM (0.9 degree)

General Specification:

Step Accuracy ----- $\pm 5\%$
 Resistance Accuracy ----- $\pm 10\%$
 Inductance Accuracy ----- $\pm 20\%$
 Temperature Rise ----- 80°C MAX.
 Ambient Temperature Range ----- $-20^{\circ}\text{C} \sim +50^{\circ}\text{C}$
 Storage Temperature Range ----- $-30^{\circ}\text{C} \sim +60^{\circ}\text{C}$
 Insulation Resistance ----- $100\text{M } \Omega \text{ MIN. } 500\text{V DC}$
 Dielectric Strength ----- $500\text{V AC } 1\text{min}$
 Radial Play ----- $0.02\text{mm MAX. (450g Load)}$
 End Play ----- $0.08\text{mm MAX. (450g Load)}$
 Max. radial force ----- 28N
 Max. axial force ----- 10N

Electrical Specification:

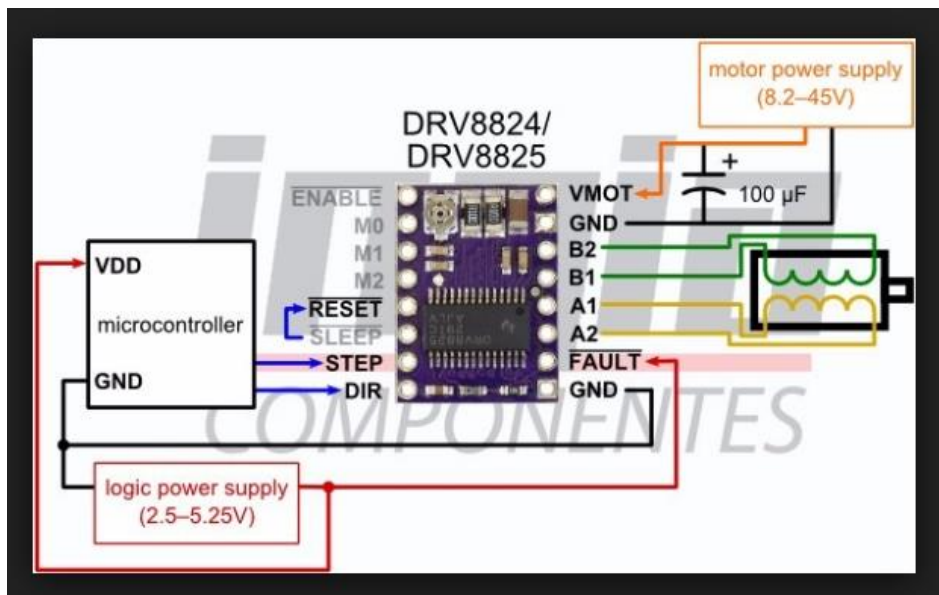
Model No.	Step Angle ($^{\circ}$)	Motor Length (L)mm	Current /Phase A	Resistance /Phase Ω	Inductance /Phase mH	Holding Torque kg.cm	Lead wires No.	Detent Torque g.cm	Rotor Inertia g.cm ²	Motor weight Kg
JK42HM34-0956	0.9	34	0.95	4.2	4	1.58	6	200	35	0.22
JK42HM34-0606	0.9	34	0.6	10	9.5	1.58	6	200	35	0.22
JK42HM34-0316	0.9	34	0.31	38.5	33	1.58	6	200	35	0.22
JK42HM34-1334	0.9	34	1.33	2.1	4.2	2.2	4	200	35	0.22
JK42HM40-1206	0.9	40	1.2	3.3	3.4	2.59	6	220	54	0.28
JK42HM40-0806	0.9	40	0.8	7.5	6.7	2.59	6	220	54	0.28
JK42HM40-0406	0.9	40	0.4	30	30	2.59	6	220	54	0.28
JK42HM40-1684	0.9	40	1.68	1.65	3.2	3.3	4	220	54	0.28
JK42HM48-1206	0.9	48	1.2	3.3	4	3.17	6	250	68	0.35
JK42HM48-0806	0.9	48	0.8	7.5	10	3.17	6	250	68	0.35
JK42HM48-0406	0.9	48	0.4	30	38	3.17	6	250	68	0.35
JK42HM48-1684	0.9	48	1.68	1.65	4.1	4.4	4	250	68	0.35

*Note: We can also manufacture products according to customer's requirements.

4.2.4 Desarrollo

Para lograr el objetivo y correcto funcionamiento del motor se ha seguido el siguiente esquema:

Esquema conexión motor.



En este esquema se muestra la conexión del microcontrolador con el driver del motor y la conexión de este con el motor, cabe destacar que para este caso no se ha conectado nada en los terminales M0,M1,M2 debido a que en esta práctica no se hace mover el motor en micropasos sino en modo full steep.

4.3 Bloque de sensores

4.3.1 Objetivo

El objetivo principal de este bloque es tomar medidas de forma rápida a una distancia considerable, considerándose aceptable una distancia superior a 20 metros.

4.3.2 Especificaciones

El sensor debe contar con las siguientes características:

- Comunicación I2C
- Voltaje funcionamiento inferior a 12v
- Distancia de medición 25 metros.
- Velocidad de medición igual o superior a 500 hz.

4.3.3 Alternativas

Algunos de los sensores que hay en el mercado son los siguientes:

- **LDB6**
 - Voltaje: DC3V ~ 3,3 V
 - Actual: 100mA
 - Rango de medición: 0,01-80 m
 - De medición FRequencia: 5Hz, 10Hz, 20Hz (ajustable)
 - Velocidad medible de los objetos móviles: menos de 2 m/s
 - Precisión de la medición (desviación estándar): mm
 - Unidad de distancia: m
- **GP2**
 - Tensión de alimentación: dc2.5v-2.8 v.
 - 2 mA flujo eléctrico: dc3v, en espera, medición 120 mA;
 - Tipo de láser: 635nm, 1 MW. (rojo)
 - Rango: 0.02 a 80 metros;
 - Velocidad de medición: 0.1 S, 0.1 S, 0.3 S-4 S, 4 S.

- Precisión típica: +/-2mm (0 a 40, interior 20 metros paredes blancas;
- mediante corrección puede ser hasta +/-1.5mm)
- Temperatura de trabajo:-10-40 temperatura de trabajo:-

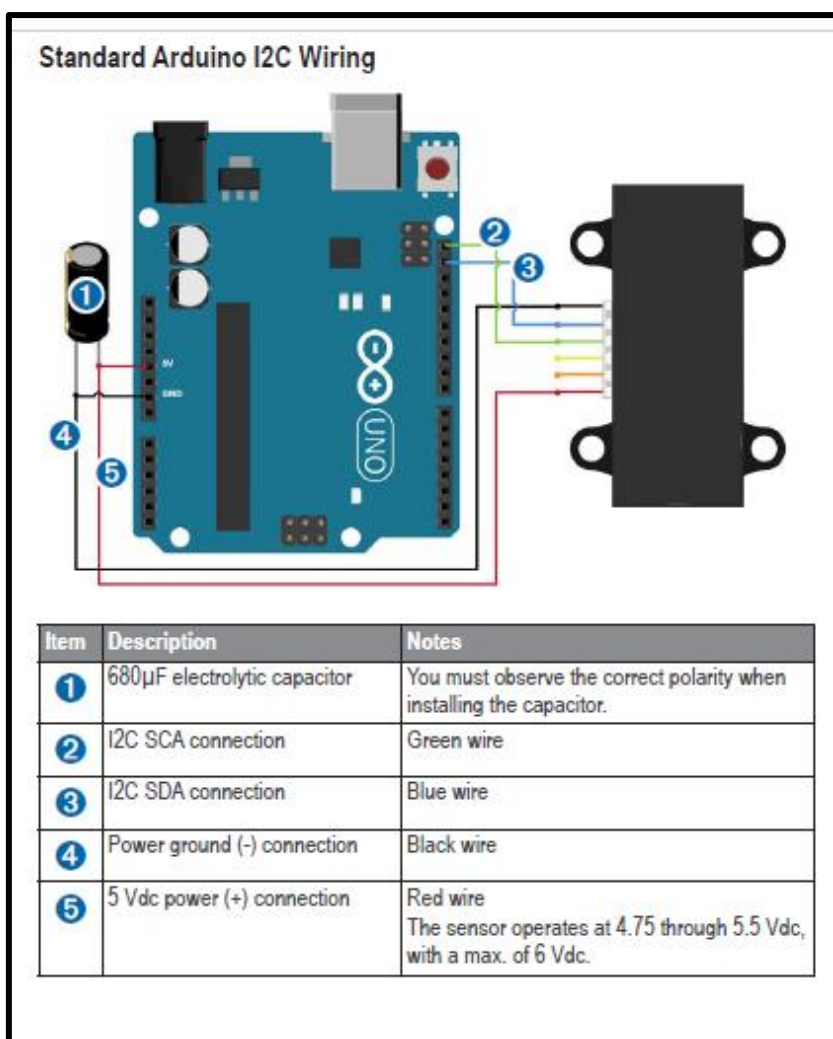
- ***LIDAR lite v3***

- Tamaño de la unidad (Al. x An. x Prof.): 20 x 48 x 40 mm
- Peso: 22 g
- Resolución: 1 cm
- Precisión: +/- 2,5 cm a distancias superiores a 1 m Consulta el manual de funcionamiento, donde encontrarás las especificaciones de funcionamiento completas.
- Distancia: 5 cm - 40 m
- Frecuencia de actualización: hasta 500 Hz
- Interfaz: I2C o PWM
- Potencia (voltaje de funcionamiento): 4,75-5,5 V de CC; 6 V máx.
- Consumo eléctrico: 105 mA inactivo; 130 mA con un funcionamiento continuo
- Temperatura de funcionamiento: de -20 a 60 °C
- Longitud de onda láser/potencia máxima: 905 nm/1,3 vatios
- Divergencia de haz: 8 m/radián
- Apertura óptica: 12,5 mm


De todas las opciones la más adecuada para nuestro proyecto es el Lidar lite v3, el cual cuenta con comunicación I2C y tiene una velocidad y rango de medida aceptable.

4.3.4 Desarrollo

La conexión del Telémetro es la siguiente:



En la siguiente captura se muestra un ejemplo simple de la programación del sensor:

A screenshot of an Arduino IDE window showing a C++ sketch. The title bar at the top indicates the file is named 'sketch_oct19a'. The sketch includes the following code:

```
serial.begin(9600); // Initialize serial connection to display distance readings

lidarLite.begin(0, true); // Set configuration to default and I2C to 400 kHz
lidarLite.configure(0); // Change this number to try out alternate configurations
}

void loop()
{
  int dist;

  // At the beginning of every 100 readings,
  // take a measurement with receiver bias correction
  if ( cal_cnt == 0 ) {
    dist = lidarLite.distance(); // With bias correction
  } else {
    dist = lidarLite.distance(false); // Without bias correction
  }

  // Increment reading counter
  cal_cnt++;
  cal_cnt = cal_cnt % 100;

  // Display distance
  Serial.print(dist);
  Serial.println(" cm");

  delay(10);
}
```

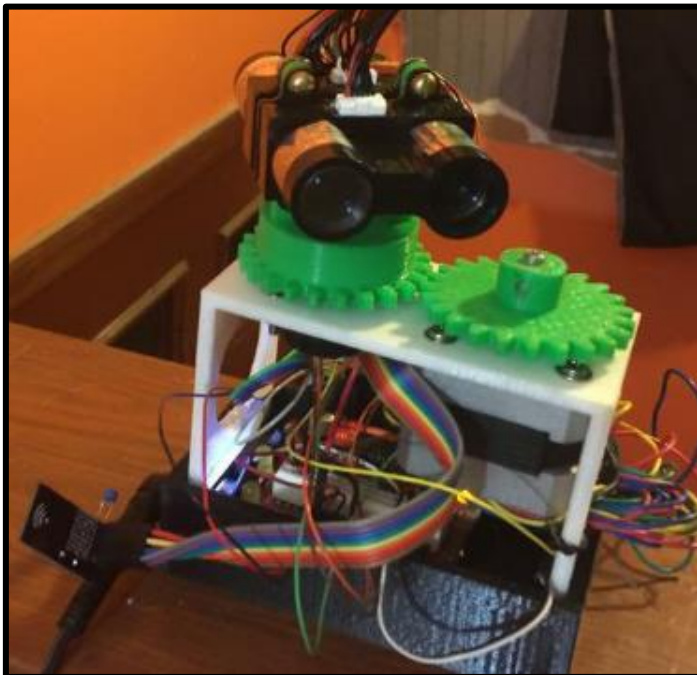
Como se muestra en la captura anterior, ejecutando la función distance, se ordena al sensor efectuar una medición. En este ejemplo efectúan la medición en el modo configurado por defecto 0. El sensor cuenta con 5 configuraciones diferentes, las cuales aún están pendientes de probarse.

Además, en este ejemplo se puede apreciar como independientemente del modo configurado, se puede efectuar una medición simple o cada 100 mediciones invocar a la función distance pero recibiendo un resultado previamente filtrado en cuanto a falsos resultados debidos a ruido, iluminación y otros problemas.

5 MONTAJE Y PUESTA EN FUNCIONAMIENTO

5.1 MONTAJE

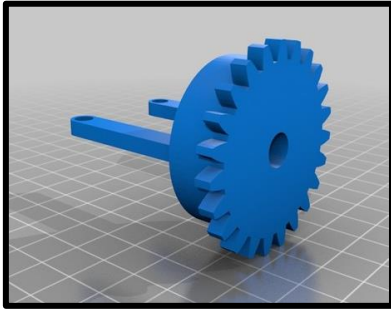
El montaje elegido para el proyecto es una variante del siguiente:



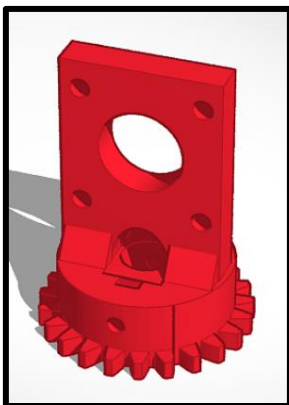
El montaje mostrado no se ha podido realizar debido a que para lograr una rotación completa del telemetro, hace falta incorporar un elemento llamado Anillo deslizante, el cual permite la conexión eléctrica de elementos que giran mas de 360 grados. Este elemento ha sido comprado pero no ha llegado a tiempo.

Debido a esto, el plan ha pasado a escanear 180 grados en lugar de 360. Para realizar el montaje, se han descargado las siguientes piezas:

- Pieza para soportar el telémetro y ser movida mediante transmisión dentada por otra rueda.

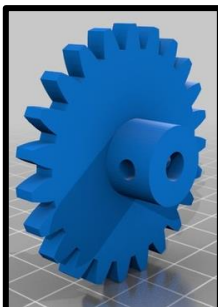


Esta pieza ofrecía una sujeción débil del telémetro y el diámetro de eje interior no era el adecuado para mi montaje de modo que mediante el programa tinkercad, se ha modificado tal pieza, dando como resultado la siguiente:

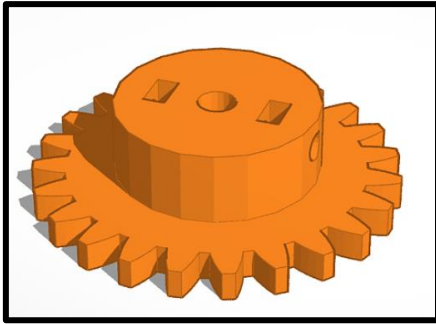


Además, se ha añadido un agujero pasante de un lado a otro de la ruda, con dos huecos en cada lado. Este agujero combinado con los huecos, permiten la sujeción de la pieza al eje interior mediante tornillos y tuercas métrica 4.

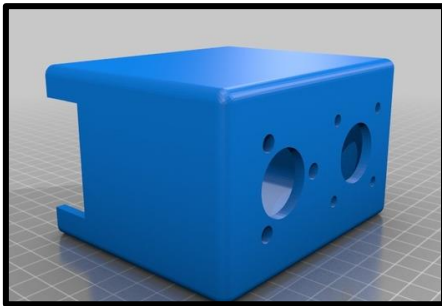
- Pieza de transmisión motor-telémetro



Esta pieza es valida para el montaje, el inconveniente es que tal y como se aprecia en la fotografía, no ofrece un sistema de fijación al eje decente por lo cual se ha modificado dando el siguiente resultado:

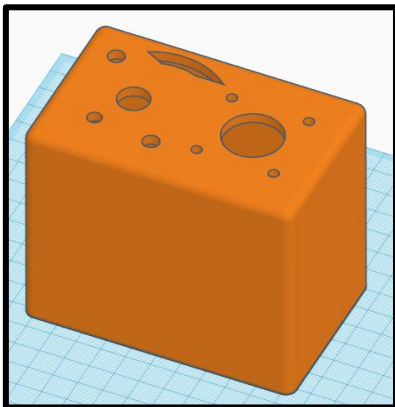


- Pieza sujeción principal



Esta pieza ha sido modificada para añadir un rodamiento destinado a sustituir el sistema de anillos deslizantes y permitir el paso del cable mediante un corte semicircular.

La modificación ha dado lugar a la siguiente pieza:



La impresión de las piezas se ha realizado con una impresora 3D prusa p3steel con PLA como material de aportación. Se han aplicado los conceptos aprendidos en el curso en cuanto a la preparación de impresión mediante el software cura.

5.1.1 FUNCIONAMIENTO

La idea de funcionamiento es la de ordenar al motor paso a paso avanzar un paso y realizar una medición mediante el telémetro, estas dos ordenes deben ejecutarse sucesivamente para lograr un escaneo de tantos grados como pasos se configuren en el programa.

A la hora de probar el funcionamiento se ha detectado el problema descrito en el apartado. (4.1.4).

Al detectarse este problema no se ha podido avanzar con el proyecto sino que en su lugar se ha intentado determinar el problema y encontrar su solución.

CONCLUSIONES

Este proyecto sigue abierto y se seguirá desarrollando de modo que es pronto para emitir unas conclusiones finales pero si expondré las conclusiones a las que he llegado hasta este punto.

- El programa Tinkercad se me mostró como un programa excesivamente simple para la elaboración de las piezas necesarias pero al aprender a utilizarlo me ha servido como una herramienta muy útil para elaborar piezas sencillas y lo más importante, modificar piezas complejas descargadas y adaptarlas a mis necesidades, esta posibilidad es de gran ayuda teniendo en cuenta que muchos programas de diseño 3D no me admitían formatos STL.
- Tras consultar especificaciones técnicas del sensor y aprender sobre su programación me he dado cuenta de que el problema no viene de su velocidad, la cual es de 250 mediciones por segundo sino de mi manera de gestionar el movimiento del motor, por lo cual mi siguiente

paso será el de consultar como otras personas lo han hecho e implementarlo en mi código.

- La transmisión de ruedas dentadas para escaneos 3D es muy útil y fácil de lograr mediante la impresión 3d pero requiere de una precisión demasiado grande para aplicaciones que requieran volver a un grado en concreto y volver a efectuar una medición en ese mismo punto.
- He aprendido a “Testear” la velocidad de medición real del telémetro mediante un código muy simple, algo muy útil.
- Es importante pensar si el diseño de la pieza ofrecida por terceros es el adecuado, en mi caso la pieza de soportación del telémetro duro menos de un día.

Este proyecto se seguirá desarrollando aplicando las correcciones necesarias para evitar los problemas surgidos hasta ahora.

Gracias a este proyecto he adquirido diversos conocimientos en los campos del diseño 3d, la impresión y la programación, sin lugar a duda ha sido provechoso y ha asentado una base solida para su posterior desarrollo.

WEBGRAFÍA

- **Montaje motor y programación**

<https://www.luisllamas.es/motores-paso-paso-arduino-driver-a4988-drv8825/>

<http://carlini.es/manejar-un-motor-stepper-con-un-driver-drv8825-y-arduino/>

- **Telémetro**

<https://buy.garmin.com/es-ES/ES/p/557294>

<https://www.sparkfun.com/products/14032>

- **Montaje**

<https://hackaday.io/project/4087-360-degree-lidar-lite-scanner>