19/01/23

Al motor “THOR ELL18” se le entrega instrucciones a partir del software disponible en la página “ELL”. Cada instrucción tiene un equivalente hexadecimal que se muestra en la pantalla, una vez ejecutada la instrucción el motor devuelve su posición, igual en hexadecimal, por lo cual se requiere traducir dicho código con el fin de entender qué es lo que nos dice la máquina.

Para traducir dicho código se tiene la siguiente trama:

| byte 1 | byte 2 | byte 3 | byte 4 … byte N |
| --- | --- | --- | --- |
| Cabecera | ID del comando | | HEX ASCII DATA (Probablemente la posición del motor en hexadecimal) |

Se tienen los códigos en hexadecimal de la comunicación serial. Si se los convierten en decimal devuelve un número el cual es igual :

Se realizaron varios ajustes para encontrar el valor de “”, el que dio mejor resultado, y con el cual se trabajará posteriormente, es en donde se usaron 25 datos de diversas escalas de magnitudes con el fin de obtener un mejor ajuste. El valor de “” para dicho ajuste fue:

24/01/23

Con el “Communication Protocol” del motor “THOR ELL18”, se investigó los diversos ID de los comandos que se pueden ingresar. Con dicha información se realizó un código en “Python”, usando las librerías: “pyserial” para establecer la comunicación serial.

Igualmente, con el modelo realizado se realiza un código para transformar los datos de ángulo a decimal, y posteriormente a hexadecimal. Aquello permitirá al usuario ingresar directamente los ángulos, sin necesidad de conocer los valores en hexadecimal, e irlos guardando en una hoja de datos.

Para ingresar datos mediante la librería pyserial es necesario que los datos se encuentren codificados mediante en formato UTF-8, por tanto, en el programa se incluye dicha codificación de datos.

**Problemas a resolver:**

Se detecta el puerto e incluso se ejecuta la apertura de este, sin embargo, cuando se trata de enviar alguna instrucción no se establece la comunicación.

**A revisar:**

Ricardo se comunicará con alguien que conozca a mayor profundidad la comunicación serial para tener asesoramiento, y poder solucionar el problema.

26/01/23

Se revisó el código anteriormente escrito al igual que la documentación disponible sobre los diversos comandos dentro de la comunicación serial, logrando resolver el problema con el establecimiento de la comunicación. El problema se debía tanto a cómo se definió al puerto, con sus diversas características, al igual de cómo se envían los comandos al puerto, ya que se los estaba enviando en formato de codificación UTF-16, mientras que el motor requería comandos UTF-8.

Se procedió a escribir el código para que el motor realice los movimientos que el usuario desee, en el proceso de escritura se notaron varios, errores, peculiaridades y posibles mejoras, varias de estas son:

* El valor mínimo de movimiento del motor debe ser 0.5° (grados) con el fin de minimizar los errores de medición.
* Al escribir un programa que realice movimientos por iteraciones falla en el valor de 0.5° (grados), sin embargo, para el valor de 0.1° (grados) funciona medianamente “bien” con respecto a lo esperado.
* Se podría aumentar el tiempo de escritura para que al momento de realizar códigos iterativos el motor realice movimientos más sutiles, lo cual puede ayudar a minimizar el error.
* Se debe investigar cómo obtener la respuesta del motor a enviar una instrucción, dentro de ELLO se podría observar como este devolvió la posición angular “verdadera” del motor. Esto guarda suma importancia pues es esta posición la que debe ser considerada en el análisis de datos.

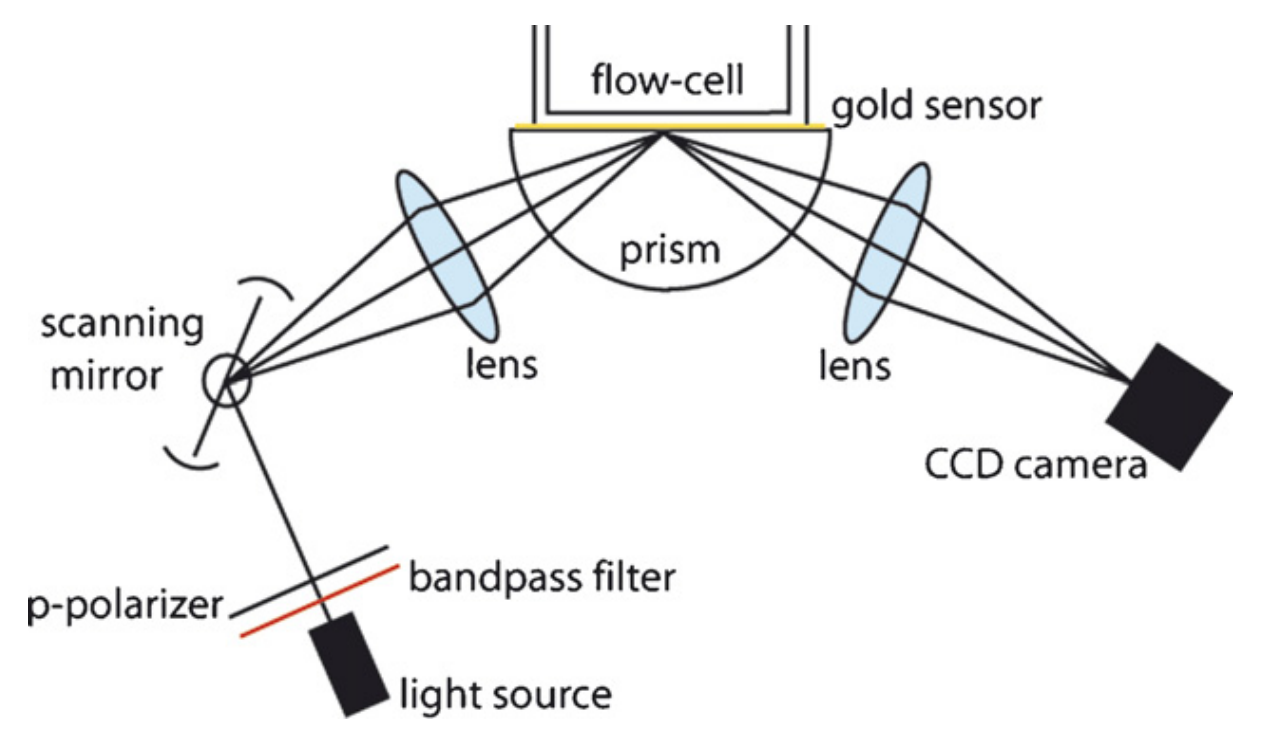
Asimismo, dentro del código se incluyó una sección que permite encerar al motor, debido a que se el motor interpreta la posición cero, al realizar movimiento absolutos, en la posición en la que se enciende. Para encerar se usó el comando "0ho0”.

Se proyecta continuar con el desarrollo del programa y al mismo tiempo con el desarrollo de la parte mecánica del proyecto: Cómo adaptar el Led al motor, y que su movimiento relativo poseen una relación constante.

31/01/23

Se inició con la planificación del montaje experimental. Con Ricardo se analizó la posibilidad de aplicar varios montajes que constan en la *Review* titulada “Recent Advances in Surface Plasmon Resonance Imaging Sensors” (archivo disponible en la carpeta “Repositorio”). Ricardo comentó el interés de que el montaje que vaya a ser aplicado sea horizontal, debido a su mayor estabilidad, sirviendo este parámetro como filtro para encontrar el montaje a aplicar.

Tomando en cuenta que se dispone del motor THOR ELL8 y su arquitectura, se eligió un montaje experimental análogo al siguiente.



En el montaje se tiene un espejo que refleja la luz incidente de la fuente, este espejo estará conectado al motor, controlando su ángulo de desplazamiento, la luz reflejada pasará por un lente para posteriormente incidir en el prisma y la lámina de oro, para, posteriormente, reflejar y pasar por un segundo lente (que probablemente deberá tener el mismo foco del primer lente. **Por revisar**) que desviará al haz de luz a un detector (Probablemente un fotodiodo).

En caso de ser un fotodiodo, se deberá realizar el acondicionamiento de la señal, mediante un circuito acon. de transimpedancia, y luego adquirir los datos por medio de un microcontrolador (“ARDUINO UNO”).

07/02/23

Se discutió sobre los diferentes programas que permiten el desarrollo de interfaces para el programa que permitirá: mover el motor, recolectar los datos (de posición angular y del fotodiodo), y graficar. Como se realizó el programa de comunicación serial mediante python, se podría tener problemas al momento de concatenar con el programa en ARDUINO para la recolección de datos del fotodiodo. Por tanto, se pensó en unificar ambos programas para evitar los posibles problemas.

Se continuó con la investigación de diversas plataformas que permitan el desarrollo de interfaces como:

* GTK (open source, varios lenguajes: c++, c, fortran, python, etc )
* MegunoLink (Pagado, prueba gratis, ARDUINO)
* Tkinter (Python)
* ECLIPSE (c++,c)