
CAPITULO VI Actuator Lineal

6.1 FIRGELLI

Con los experimentos y los trabajos realizados sobre el PPA 20 y con resultados no favorables para esta actividad propuesta, se decidió realizar una mejora a un proyecto de locomoción de robot, y aunque este no estuviera ligado con materiales piezoeléctricos, podríamos realizar mejoras considerables con los estudios que veníamos realizando previamente.

El objetivo de este proyecto sería lograr el desplazamiento de forma ascendente en un ángulo de 90° dentro del tubo, además de poder desplazarse por tubos con un cierto radio de curvatura. El principio de funcionamiento será tomar el actuador lineal, del cual hablaremos más adelante, y construir sus bases de algún material ligero pero a su vez rígido para poder soportar la fricción generada por el rozamiento del tubo (plástico), y el sistema de avance (cerdas de cepillo de dientes).

El actuador que utilizaremos para este experimento se tenía inactivo y fue el actuador lineal FIRGELLI PQ 12 de la marca FIRGUELLI.



Fig. 44. Actuador Lineal Firgelli PQ 12

Cabe mencionar que no fue un actuador de cual se tuvo una discusión sobre su compra, ya que para este proyecto no estaba planeado, sin embargo era un dispositivo que se tenía disponible y a su vez inactivo. Por lo cual se decidió trabajar en él, y con el poco tiempo que se tenía para poder concluir estos proyectos, no teníamos mucha opción para elegir. Este actuador está controlado por un motor interno, que activa un pistón que se expande o retrae dependiendo del voltaje que a este se le esté aplicando, además cuenta con un potenciómetro interno que a base de diferencia de resistencia nos puede proporcionar la posición del pistón cuando esta está activo o no lo está.

Stroke	20 mm
Input Voltage	6 or 12 VDC
Stall Current	550mA @ 6V, 220mA @ 12V
Mass	15g
Operating Temperature	-10°C to +50°C
Positional Accuracy	±0.1mm
Lifetime	20,000 strokes, 20% Duty Cycle
Audible Noise	55dB @ 45cm
Ingress Protection	IP-54
Feedback Potentiometer	1/8W Non-Buffered 10kΩ Potentiometer
Limit Switches	Max. Current Leakage: 8uA

Fig. 45. Datasheet Actuador Lineal PQ 12

Lo primero que se realizó fue poner en marcha el mecanismo del pistón, crear su fuente de alimentación.

Para esto realizamos las pruebas principales de funcionamiento, para esto se requirió el siguiente material:

- Fuente de alimentación de 6 V con Corriente variable
- 2 Cables Banana-Pin
- 1 Tabla de pruebas o Protoboard

- 5 cables de conexión
- Adaptador 5-5 del PQ12
- Actuador Lineal PQ12

Al saber el actuador soportaba un margen de voltaje de 6 a 12 VDC, se optó por protección empezar solo con el voltaje mínimo, y solo probar el funcionamiento del pistón. El actuador lineal PQ 12 al recibir un voltaje positivo en sus terminales 2 y 3, respetivamente siendo la segunda la referencia o la tierra, este expandía el pisto hasta el punto máximo o al corte de voltaje. Para el caso contrario si su alimentación era negativa el PQ 12 actuador lineal contraria el pistón bajo las mismas condiciones que el primer caso.

Comprando el funcionamiento adecuado de y correcto del PQ 12 actuador lineal, pasamos a la fase de hacer un programa que pudiera controlar este movimiento de forma simultánea.

Para lo cual se decidió por un control PWM, el cual se programaría en un arduino.



Fig. 47. Controlador arduino duemilanove

Esta plataforma electrónica y de control, permite hacer programas al estilo C, C++, por lo cual se requería conocimientos básicos de programación, además de conocimiento de lo que es el control por ancho de pulso. A demás opte por utilizar un *motor shield* que es un complemento de esta plataforma.

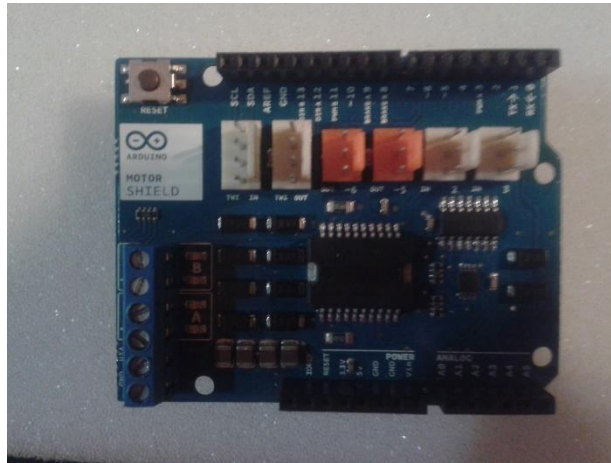


Fig. 48. Plataforma Motor Shield arduino

Este dispositivo funciona de igual forma que un puente H electrónico, de hecho lleva interno un puente H, y como sabemos y a demás utilizamos en el proyecto pasado, estos puentes H son muy utilizados cuando se requiere controlar el sentido de un motor. En este caso en particular es lo que realmente necesitábamos, control de dirección de un motor.

6.2 CONTROL PWM

Ya obtenidos los dos dispositivos que se requerían para la control del motor, se procedió con la creación del programa para el control del movimiento. A continuación se presenta el programa completo y seguido de esto se ira explicando paso a paso, la función de cada línea de programación.

```

Archivo Editar Sketch Herramientas Ayuda
firgellipq12_MOD $
#define SalidamotorA 3 // Canal A
#define DireccionmotorA 12 // Canal A
#define pinSelecc A1 // Pin de salida analoga A1

int limiteInferior = 10;
int limiteSuperior = 16;
int direccion = HIGH; // Ponemos la direccion en ALTO para que el Piston se expanda

int posicion = 0; // Ponemos posicion en 0 para empezar a comparar
void setup()
{
  Serial.begin(9600);
  pinMode(SalidamotorA, OUTPUT);
  pinMode(DireccionmotorA, OUTPUT);

  analogWrite( SalidamotorA,0 );
}

void loop()
{
  posicion = analogRead( pinSelecc );

  if ( posicion >= limiteSuperior ) // Comparamos la posicion en la cual se encuentra el piston,
    // si ya paso la poscion Superior, la salida se pone en BAJO y esta contrae el piston.
  {
    direccion = LOW;
  }

  if( posicion <= limiteInferior ) // Caso contrario a la linea 23, si se sobrepaso el limite inferior, el piston se extiende.
  {
    direccion = HIGH;
    Serial.print ("posicion:");
    Serial.println(posicion);
  }

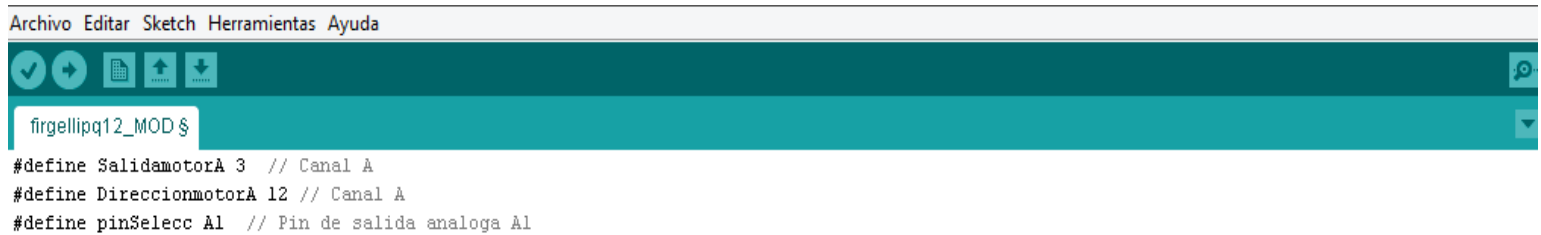
  analogWrite( SalidamotorA,255 );
  digitalWrite( DireccionmotorA,direccion );

  delay( 300 ); // Un tiempo de reaccion o de retraso de 300 ms es suficiente para que el programa pueda evaluar la posicion del piston.
}
Guardado Terminado.

```

Fig. 49. Programa completo para arduino

Ahora explicaremos el código:



```

Archivo  Editar  Sketch  Herramientas  Ayuda
firgellipq12_MOD $
#define SalidamotorA 3 // Canal A
#define DireccionmotorA 12 // Canal A
#define pinSelecc A1 // Pin de salida analoga A1

```

Fig. 50. Declaración de pines

En estas líneas definimos los pines que controlaran al motor, que en este caso son los pines 3 y 12 del *motor shield*, estos mismos pertenecen al canal de salidas A. En la tercera línea definimos el pin de feedback, que es la salida análoga A1.

```

int limiteInferior = 10;
int limiteSuperior = 16;
int direccion = HIGH; // Ponemos la direccion en ALTO para que el Piston se expanda

```

Fig. 51. Declaración de variables enteras

En estas líneas definimos las variables de tipo entero de los límites de la posición del pistón, esta misma es regresada por el potenciómetro que esta interno en el PQ 12 actuador lineal. La variable *dirección* se establece en alto desde un principio para expandir el pistón.

```

int posicion = 0; // Ponemos posicion en 0 para empezar a comparar

```

Fig. 52. Valor inicial de potenciómetro

Establecemos el valor entero leído por el potenciómetro, acerca de la posición del pistón en cada ciclo de trabajo.

```

void setup()
{
  Serial.begin(9600);

```

Fig. 53. Velocidad de bits por segundo

Establecemos la velocidad de bits por segundo que queremos que transmita el arduino, la cual debe estar síncrona con el monitor IDE de arduino que nos mostrara los datos, si este valor fuera diferente en alguno de los dos, no podríamos mostrar los datos.

```
pinMode(SalidamotorA, OUTPUT);
pinMode(DireccionmotorA, OUTPUT);
```

Fig. 54. Pines de salida

Ponemos en modo de salida los pines que contienen a las variables "SalidamotorA" y la dirección del motor "DireccionmotorA".

```
    analogWrite( SalidamotorA,0 );
}
```

Fig. 55. Velocidad del actuador

Posterior a esto ponemos a cero la velocidad del actuador, mientras la configuración está siendo cargada.

```
void loop()
{
    posicion = analogRead( pinSelecc );
```

Fig. 56. Lectura del pin A1

En esta instrucción simplemente leemos el valor de nuestro pin de salida "pinSelecc", en pin A1 de las salidas análogas.

```
if ( posicion >= limiteSuperior ) // Comparamos la posicion en la cual se encuentra el piston,
    // si ya paso la poscion Superior, la salida se pone en BAJO y esta contrae el piston.
{
    direccion = LOW;
}
```

Fig. 57. Posición del pistón

En esta instrucción hacemos una comparación de la posición del pistón, con respecto al límite superior previamente establecido. De cumplirse la condición, este valor cambiara de sentido "LOW", para retraer el pistón ya que este ha llegado al límite máximo permitido.

```
if( posicion <= limiteInferior ) // Caso contrario a la linea 23, si se sobrepaso el limite inferior, el piston se extiende.
{
  direccion = HIGH;
```

Fig. 58. Comparación de posición

Al igual que en la instrucción pasada se hace una comparación de la posición del pistón, y en caso de cumplirse la condición el valor cambiara de sentido "HIGH", para poder extender nuevamente el pistón y continuar con su ciclo.

```
Serial.print ("posicion:");
Serial.println(posicion);
}
```

Fig. 59. Imprime posición

Lo que realiza al final de este ciclo es una impresión en el monitor del valor de la posición del pistón, se colocó al final del ciclo solo por cuestiones de evaluación, y saber el valor del límite inferior para poder establecer los parámetros iniciales. Sin embargo, esta instrucción podría ser colocada al inicio o entre las dos instrucciones sin algún problema, todo depende de que sea lo que se requiera monitorear.

```
analogWrite( SalidamotorA,255 );
digitalWrite( DireccionmotorA,direccion );
```

Fig. 60. Ejecución de comandos

Aquí ejecuta los comandos del control de arduino, basándose en la velocidad y dirección establecidas por las instrucciones anteriores.

```
delay( 300 ); // Un tiempo de reaccion o de retraso de 300 ms es suficiente para que el programa pueda evaluar la posicion del piston.  
}
```

Fig. 61. Tiempo de espera

Se estable el tiempo de espera entre cada ciclo, el cual se propuso que sea de 300 milésimas de segundos.

6.3 ANÁLISIS DE DISEÑO MECÁNICO

Una vez creado el programa se procedió al diseño mecánico que tendría este robot, este debería considerar las dimensiones internas del tubo que era de 50 mm, además de obtener cierta centralidad para no girar en su trayectoria. Cabe mencionar que el pistón solo cuenta con una carrera de 20 mm. Y la única opción para un acoplamiento sugerido por el proveedor es un barreno de 3 mm. Esto siendo la parte delantera del PQ 12, aun debíamos idear un soporte en la parte media del actuador. La parte media fue elegida en vez de la parte final debido a que así tendríamos un mejor radio de curvatura, en un tubo con dichas geometrías.

También se debería considerar el peso del material con el cual fabricaríamos estos soportes. Este material debería ser manufacturable a pequeña escala y con el herramental que se tiene en el taller mecánico. A continuación se presenta el diseño del PQ 12 y posteriormente algunos de los diseños propuestos así como el diseño final.

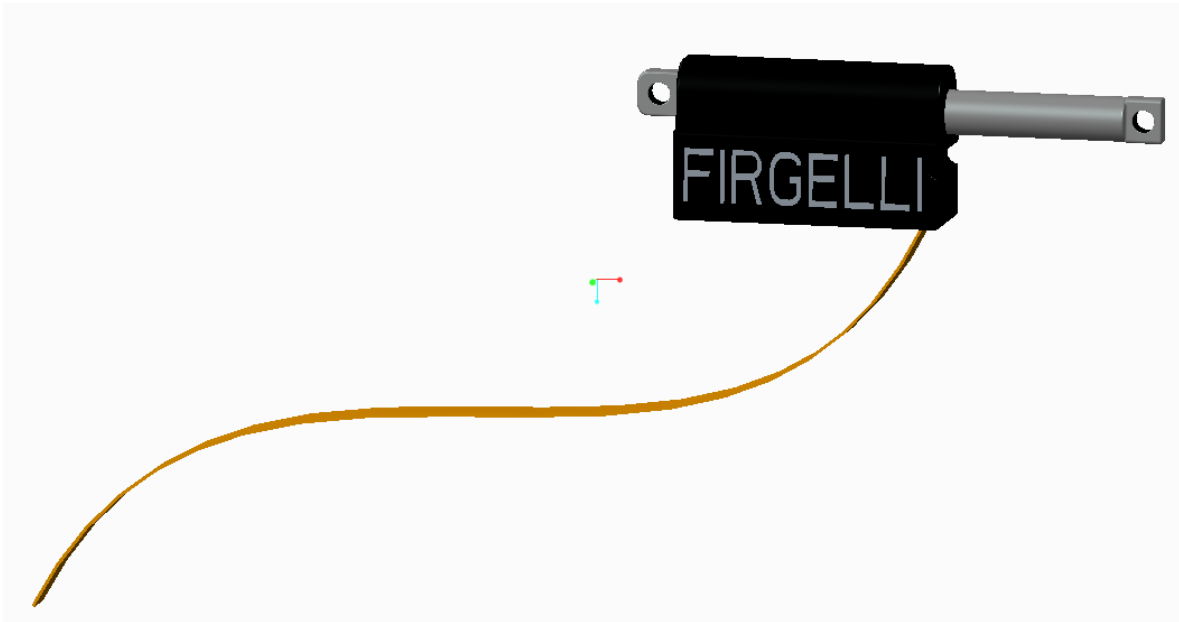


Fig. 62. Modelo del PQ 12 en CAD

Aquí presentamos el modelo en CAD, realizado en el programa CREO Parametric 2.0. Se realizó la réplica para poder simular los posibles diseños mecánicos antes de su construcción. Con esto, podrían ser evaluados por el responsable del proyecto y el tesista en cuestión. Aquí apreciamos el actuador en su estado de expulsión del pistón. En condiciones normales y sin estar activo el pistón (en color gris), se encontraría retraído dentro del cascaron (color negro). Ahora basándonos en las posibles áreas de sujeción, empezaremos con nuestros prototipos.

Mostramos a continuación las cotas del actuador PQ 12 Firgelli
Actuador Lineal:

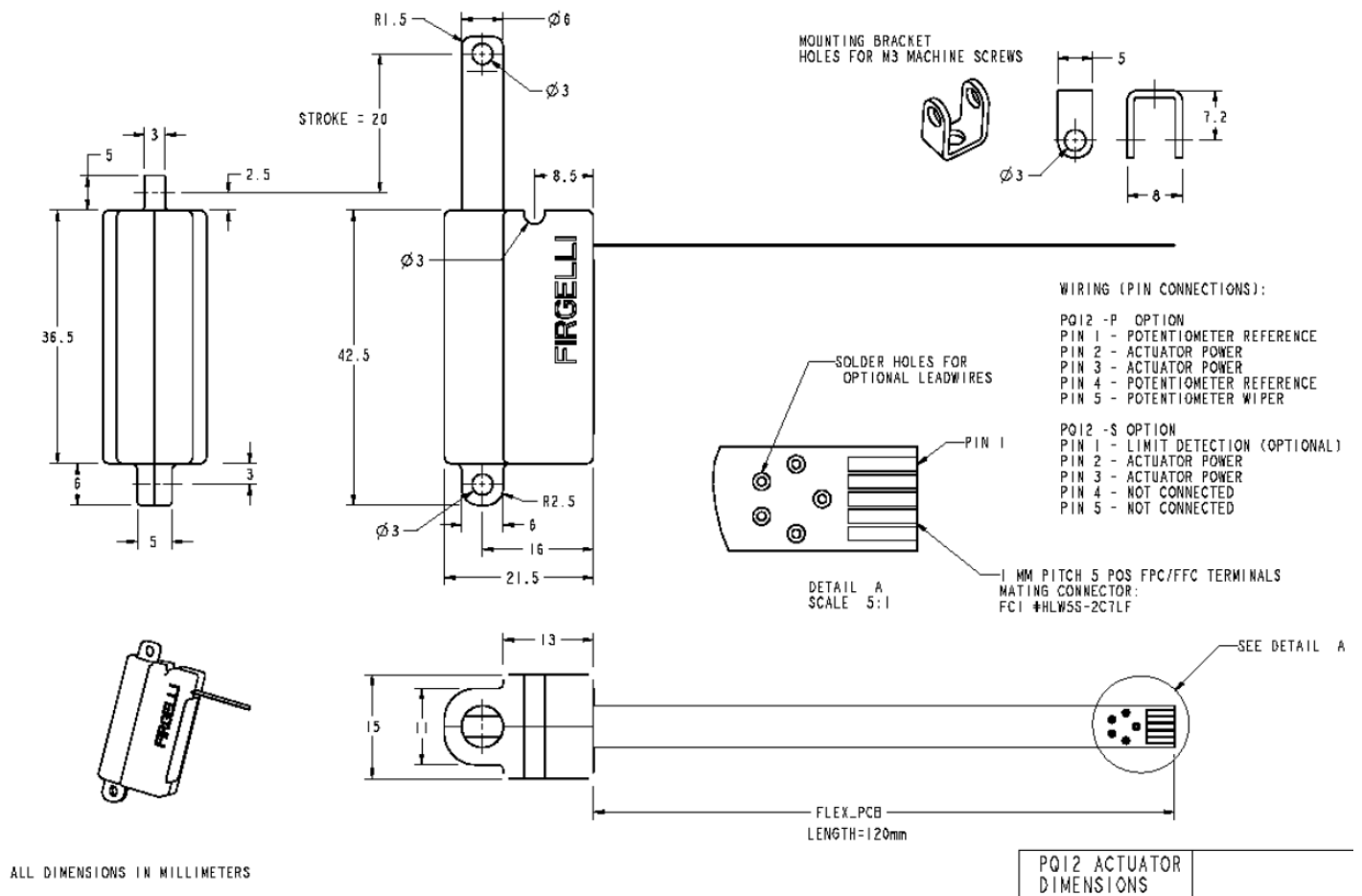


Fig. 63. Dimensiones del PQ 12

Ahora trabajaremos sobre los diseños de CAD para poder analizar las posibles opciones viables para el proyecto.

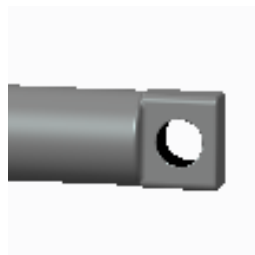


Fig. 64. Barreno de 3mm delantero de sujeción

Primero trabajaremos sobre la parte activa del actuador, que será la que nos proporcionara el desplazamiento y la fuerza necesaria para la locomoción del mismo robot. En la figura 60, mostramos la parte final del pistón, el cual cuenta con un barreno de 3 mm de diámetro, este barreno nos servirá para fijar el mecanismo que tendrá contacto con el tubo y proporcionara la fricción necesaria para hacer la locomoción del robot.

No podemos colocar o pegar algún mecanismo a lo largo del pistón ya que este al retraerse podría ser obstaculizado por dicho mecanismo, además que se requiere cuidar y no destruir el PQ 12 actuador lineal, esto con el fin de poder ser utilizado más adelante para futuros proyectos. No se puede simplemente pegar algo sobre el cascaron del PQ 12 debido a que el pegamento podría dañar al PQ 12 actuador lineal. Entonces determinado esto, pasamos a las propuestas de diseño.

CAPITULO VII Propuestas de Diseño

7.1 DISEÑO 1

Sistema de tres soportes:

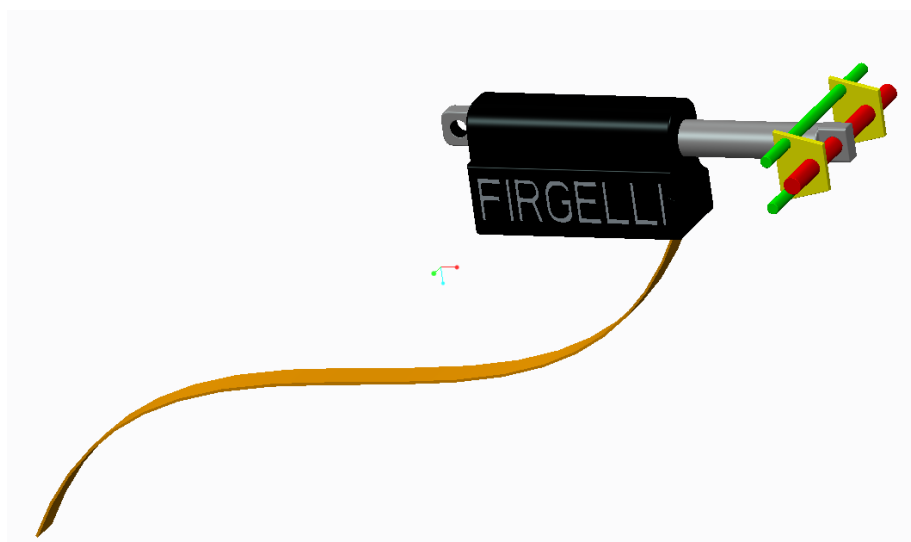


Fig. 65. Propuesta 1 del PQ 12