

# AUTOMATIZACIÓN DE LA RENDIJA DEL ESPECTRÓGRAFO ECHELLE.

G. Sierra, F. Murillo, J.M. Murillo, G. Guisa, B. García.

Instituto de Astronomía. Universidad Nacional Autónoma de México.  
Km. 103 Carretera Tijuana-Ensenada, Ensenada, B.C., México.

## RESUMEN

El presente trabajo documenta el mecanismo, sistema electrónico y programas desarrollados para automatizar la rendija del espectrógrafo Echelle. El desarrollo permite

una operación más eficiente del instrumento durante las noches de observación en el telescopio de 2.1m del OAN.

## Contenido

---

1. INTRODUCCIÓN.-----	3
2. SISTEMA MECÁNICO.-----	3
2.2.1 MECANISMO ACTUAL -----	3
2.2.2 MECANISMO PROPUESTO.-----	4
3. ELECTRÓNICA DE CONTROL.-----	9
3.1 EL MICROCONTROLADOR BEAGLEBONE. -----	11
3.1.1 ASIGNACION DE PATAS -----	11
3.1.2 PROGRAMAS DESARROLLADOS.-----	13
3.1.2.1 LA BIBLIOTECA DE FUNCIONES. -----	13
3.1.2.2 EL PROGRAMA DE CONTROL EN LENGUAJE PYTHON.-----	14
3.1.3 EL SERVICIO DE AUTOARRANQUE. -----	16
3.2 LA TARJETA DE POTENCIA PARA MOTORES.-----	16
3.3 LA TARJETA DE RELEVADOR Y ENTRADAS OPTOACOPLADAS. -----	16
4. LA INTERFAZ DE USUARIO.-----	17
5. PRUEBAS Y RESULTADOS.-----	17
6. APÉNDICE A. DIAGRAMAS ELECTRÓNICOS.-----	18
6.1 TARJETA DEL MICROCONTROLADOR BEAGLEBONE. -----	18
6.1.1 ESQUEMÁTICO.-----	18

6.1.2 MAPA DE COMPONENTES. -----	19
6.2 TARJETA DE POTENCIA PARA EL MECANISMO DE RENDIJA. -----	20
6.2.1 ESQUEMÁTICO. -----	20
6.2.2 MAPA DE COMPONENTES. -----	21
6.3 TARJETA DE POTENCIA PARA EL MECANISMO DE LAMPARA. -----	22
6.3.1 ESQUEMÁTICO. -----	22
6.3.2 MAPA DE COMPONENTES. -----	23
6.4 TARJETA DE RELEVADOR Y ENTRADAS OPTOACOPLADAS. -----	24
6.4.1 ESQUEMÁTICO. -----	24
6.4.2 MAPA DE COMPONENTES -----	24
7. APENDICE B. ARCHIVO DE CONFIGURACIÓN DE PATAS DE LA MICROCOMPUTADORA BEAGLEBONE. ---	25
8. APENDICE C. SEÑALES EN LOS CONECTORES. -----	27
7.1 CONECTOR DEL MECANISMO DE RENDIJA. -----	27
7.2 CONECTOR DEL MECANISMO DE LÁMPARA. -----	27
9. APENDICE D. SERVICIO DE AUTOARRANQUE DEL SISTEMA. -----	28
8.1 CONTENIDO DEL ARCHIVO "RUN_ECHELLE.SERVICE" -----	28
8.2 CONTENIDO DEL ARCHIVO "RUN_ECHELLE.SH". -----	28
10. REFERENCIAS. -----	28

## 1. INTRODUCCIÓN.

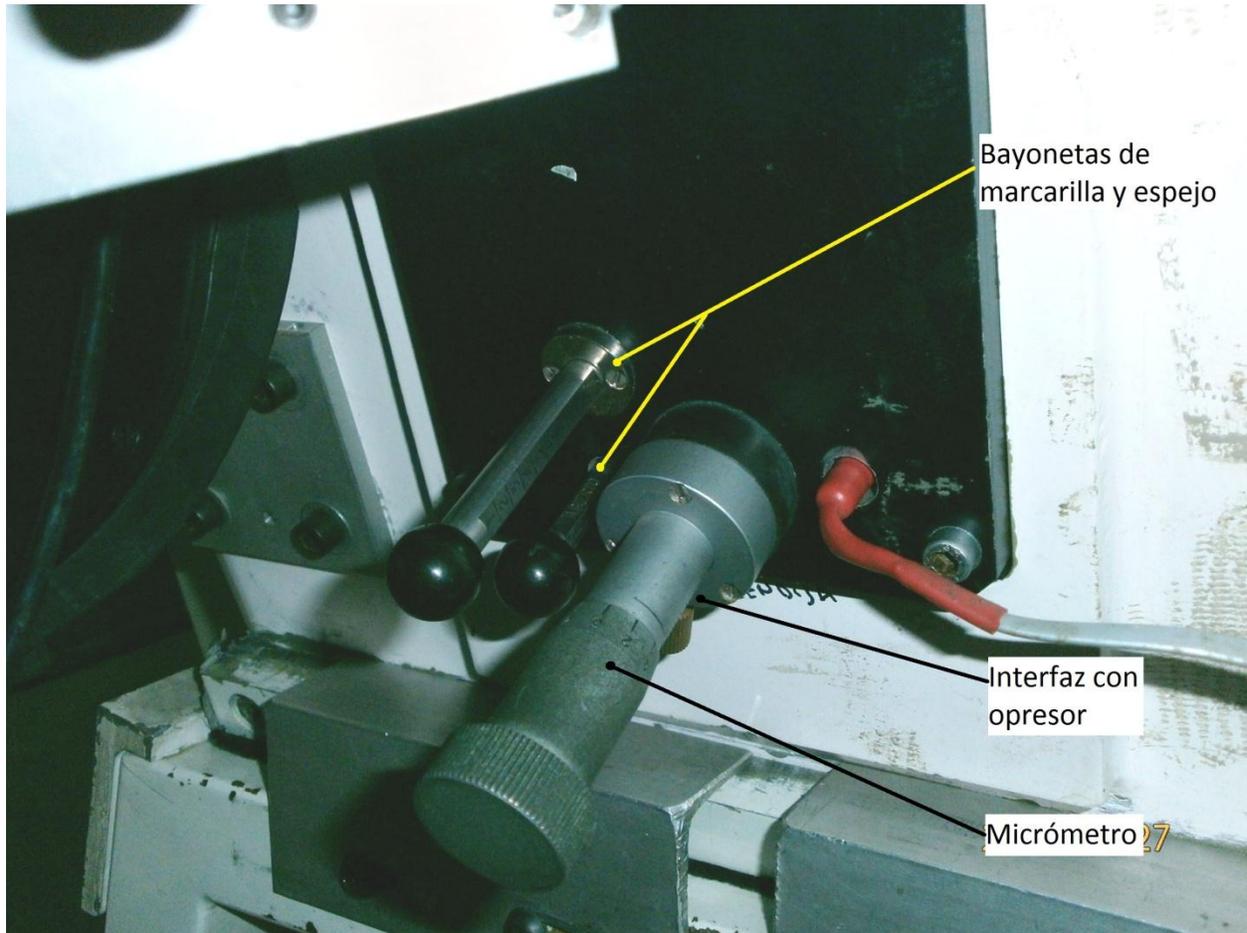
La automatización de la rendija del espectrógrafo Echelle surge de la necesidad de hacer más eficiente la operación del instrumento, los movimientos de apertura de la rendija durante una noche de observación suelen ser frecuentes, esta tarea es realizada por el operador de telescopio quien sube dos niveles en el edificio de telescopio, usando el elevador o escalera, para después colocar una escalera portátil que le permita alcanzar el micrómetro que mueve la rendija. Esta tarea manual toma del orden de 3 a 5 minutos en realizarse. El equipo de instrumentación se dio a la tarea de automatizar este procedimiento. Los requerimientos del diseño mecánico para la automatización fueron que el mecanismo fuera robusto, sencillo, fácil de montar y barato. Con la implementación del sistema automático el observador simplemente presiona un botón en una interfaz gráfica en la computadora de usuario y una microcomputadora conectada a un motor se encarga de posicionar la rendija, con esto la tarea toma 5 segundos.

En las siguientes secciones se presenta el diseño del mecanismo que mueve el micrómetro utilizando un motor de pasos, la electrónica de control basada en una microcomputadora de tecnología reciente y el programa de interfaz de usuario desarrollado en lenguaje Python.

## 2. SISTEMA MECÁNICO.

### 2.2.1 MECANISMO ACTUAL

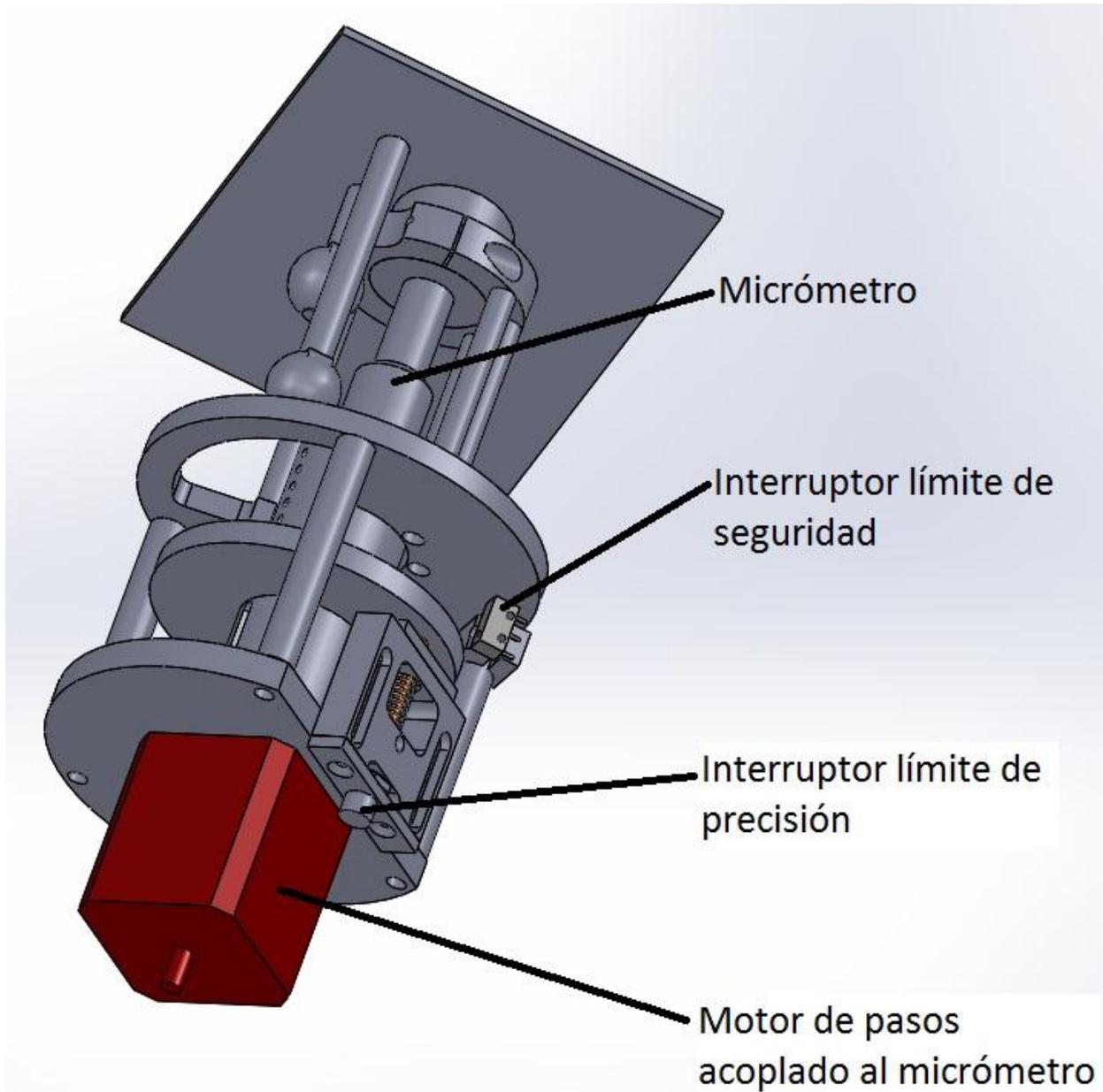
Actualmente el mecanismo consiste en un micrómetro que está ensamblado al instrumento por medio de una interfaz con opresor, muy cerca a las bayonetas de marcarillas. Para modificar la apertura de la rendija es necesario girar el micrómetro manualmente, y la forma de regular la separación de las navajas es por medio de la lectura del micrómetro. La precisión de apertura depende de la correcta lectura y manipulación del micrómetro que en ocasiones no es fácil de operar, dependiendo de la posición del telescopio. Véase figura 1.



*Figura 1. Mecanismo actual para modificar la apertura de la rendija del Espectrógrafo Echelle.*

### 2.2.2 MECANISMO PROPUESTO.

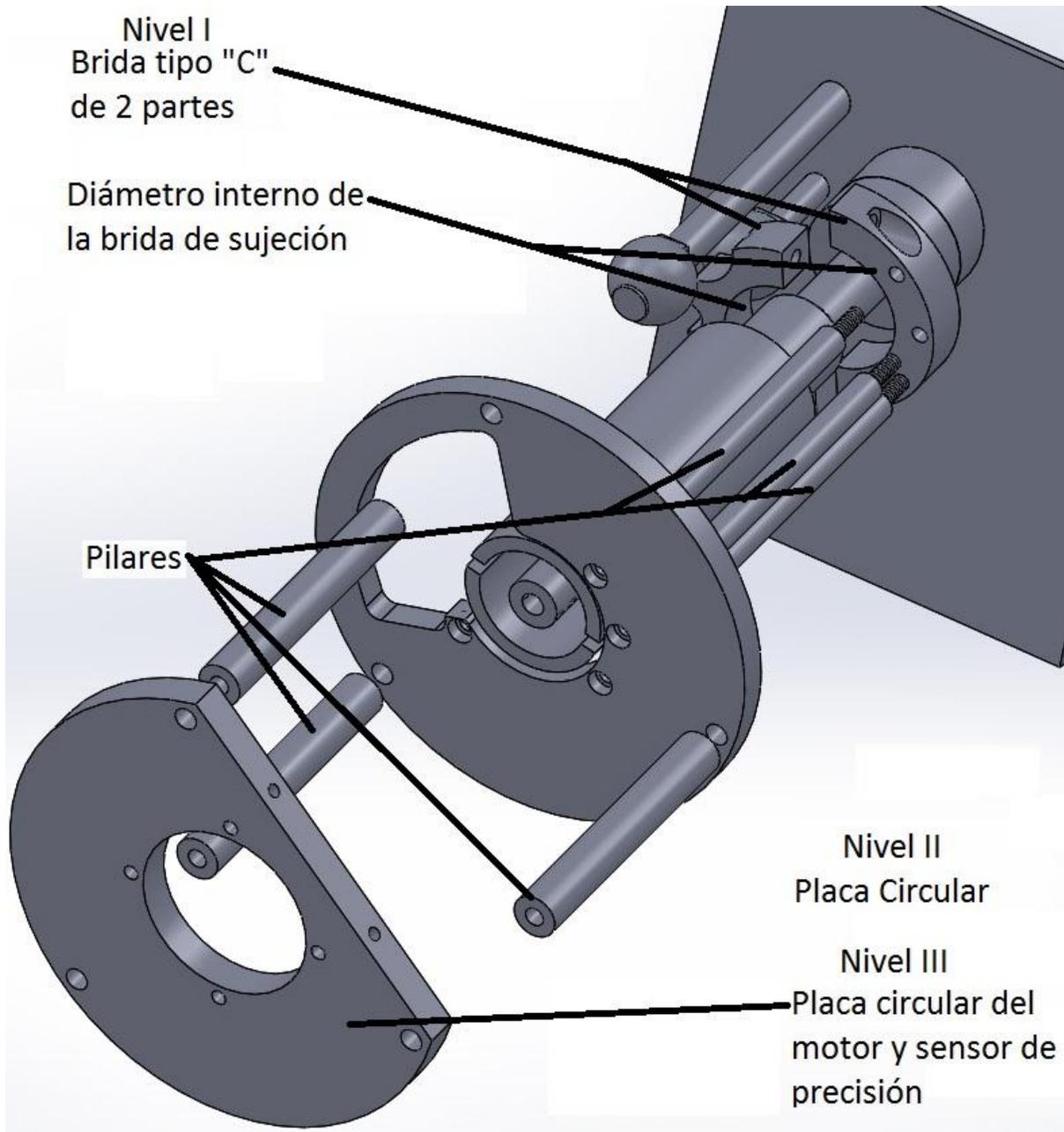
La propuesta del nuevo mecanismo consiste en acoplar un motor de pasos al micrómetro, por medio de una montura concéntrica al micrómetro. El motor de pasos es codificado por medio de la electrónica que controla los pasos. Tiene dos interruptores límites; uno de precisión que inicializa el sistema de control y otro de seguridad ubicado al final del recorrido del micrómetro. Véase figura 2



**Figura 2. Propuesta del mecanismo concéntrico al micrómetro para controlar la apertura de las navajas de la rendija del Espectrógrafo Echelle.**

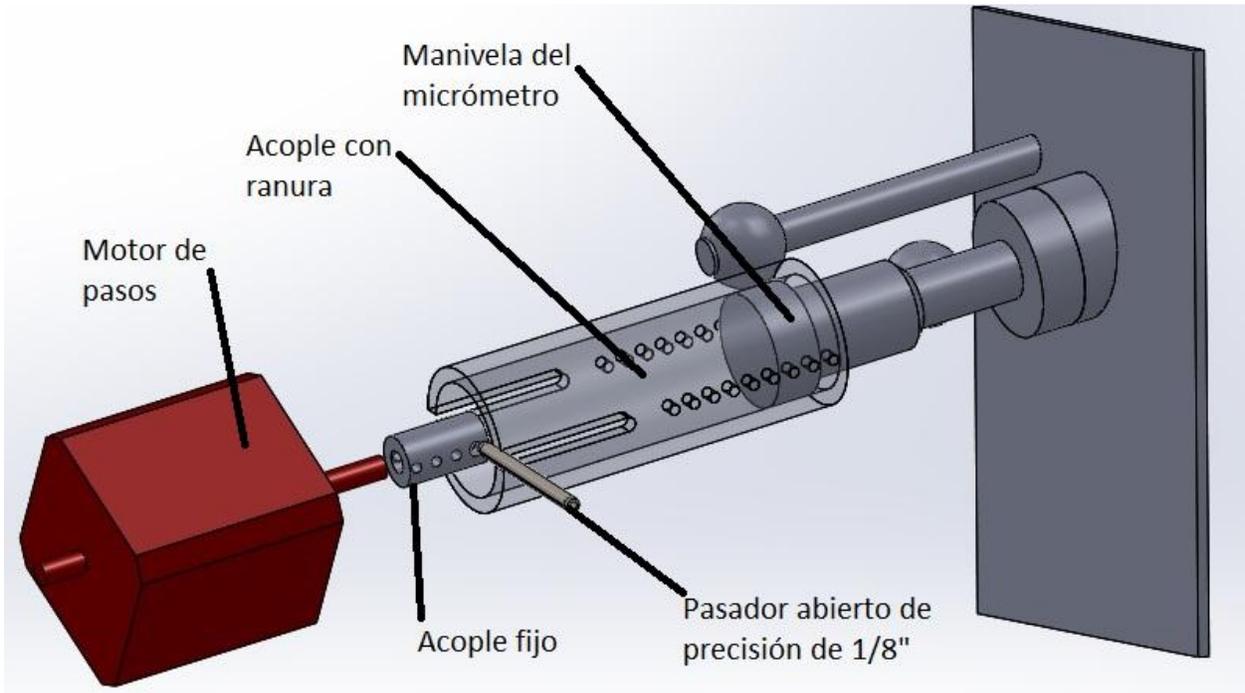
La estructura que soporta al motor es de aluminio. La descripción de esta se presenta en las figura 3, donde se aprecia que la montura es del tipo estructural con tres niveles. En el primer nivel se encuentra la interfaz con el espectrógrafo que es donde se sujeta el mecanismo al espectrógrafo Echelle con una brida tipo "C" de 2 partes, la cual se aprieta a la interfaz por medio de dos tornillos 6-32. La sección de la brida con rosca cuenta con 2 muescas circulares para permitir el libre paso a las bayonetas de las máscaras. La concentricidad de las monturas al micrómetro se obtiene por medio del diámetro interno de la brida de sujeción, dado que es rectificadas y maquinadas con precisión. A la brida se le atornillan cuatro pilares tipo barras circulares que sostienen a la placa circular del segundo nivel de donde salen otros tres pilares tipo barras circulares hacia el tercer y último nivel donde se encuentra la placa de ensamble del motor de pasos y del mecanismo del interruptor de precisión. El interruptor de seguridad se

coloca sobre unas de las barras circulares que se encuentran entre los niveles dos y tres. Véase figura 3.



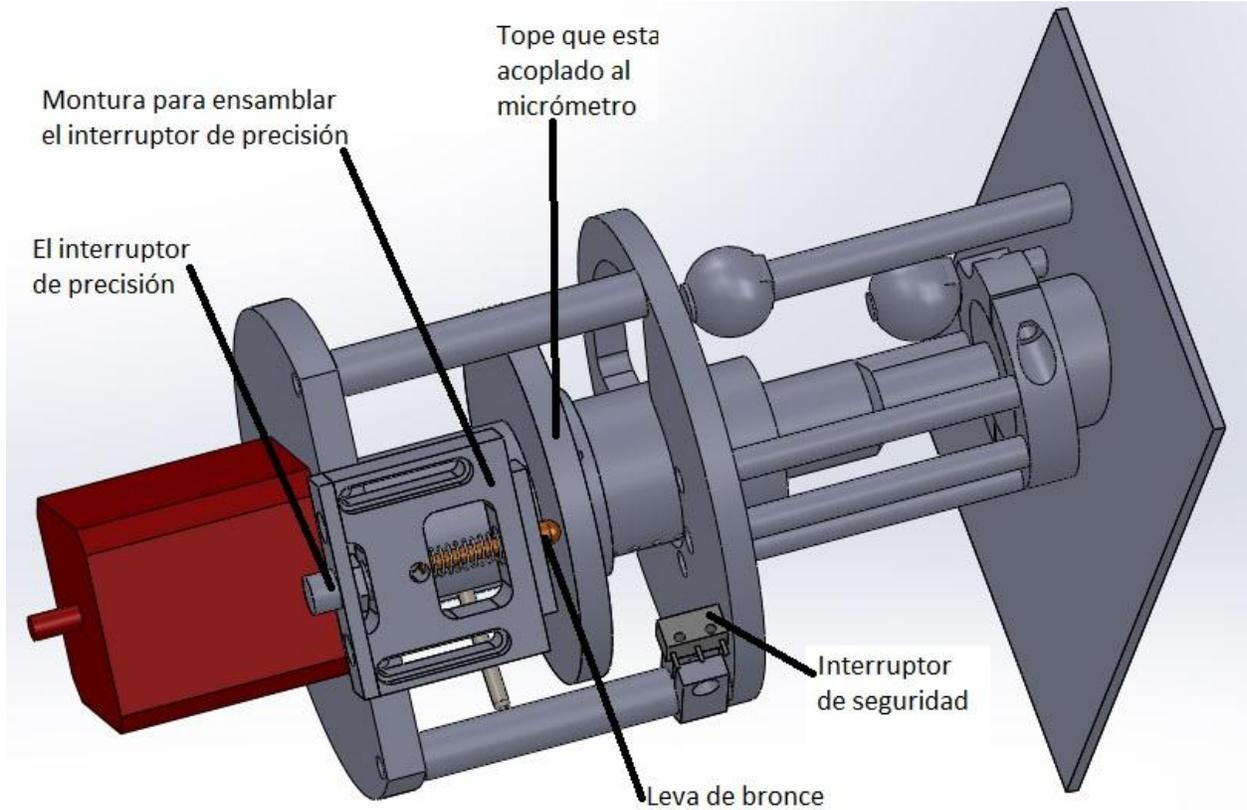
*Figura 3. Estructura de soporte del mecanismo. Presentado los tres niveles.*

Sobre la manivela del micrómetro se le monta un acople con ranura para poder transmitir la torca del motor al mismo por medio de un pasador abierto de precisión de 1/8 de pulgada que desliza por la ranura y que está sujeto al eje del motor por un acople fijo. Este mecanismo de acoplamiento absorbe desalineaciones que pudieran existir entre los ejes del motor y el del micrómetro. Véase figura 4



*Figura 4. Acoples de transmisión de potencia del motor al micrómetro. Se presenta el acople con ranura transparente para visualizar el ensamblaje.*

El interruptor de precisión es el que inicializa el sistema. Este componente comercial es muy delicado en su operación. Dado que nos da un micrón de repetitividad. Así que para robustecerlo se diseñó un mecanismo que lo protege y lo actúa de la manera adecuada. Este mecanismo consiste en una montura que protege el interruptor de precisión y cuenta con una leva de bronce que lo activa al ser presionada por el tope que esta acoplado al acople del micrómetro. Este mismo tope activa el interruptor de seguridad. Véase figura 5



*Figura 5. Los dos interruptores que limitan el movimiento de desplazamiento del micrómetro. El primero de precisión dentro de su mecanismo especial y el segundo de seguridad.*

En la figura 6 se muestran dos fotos del mecanismo ensamblado sobre el espectrógrafo cuando se estaban realizando las pruebas. Cabe mencionar que los diseños; conceptuales, preliminares, críticos y de detalle así como los planos de manufactura se desarrollaron en el IA-UNAM Ensenada. Así mismo, la manufactura de todas las piezas mecánicas en el taller mecánico de precisión del mismo instituto sede Ensenada.



*Figura 6. Fotos del mecanismo para controlar la apertura de la rejilla ensamblado sobre el instrumento cuando se realizaba pruebas.*

### **3. ELECTRÓNICA DE CONTROL.**

El control electrónico cumple con las tareas de posicionar la rendija utilizando un motor de pasos, supervisa el estado de los interruptores límite y de origen del mecanismo, además por razones de espacio en el instrumento se incorporó en este control lo necesario para controlar el sistema de iluminación del espectrógrafo: se agregó una tarjeta de potencia adicional para controlar el motor que mueve la bayoneta del espejo del sistema de lámparas y una tarjeta de relevadores para encender la lámpara. Por lo que esta electrónica sustituye a la desarrollada previamente para el sistema de lámparas y que fue reportada en [1].

El diseño del este control es similar al desarrollado para el espectrógrafo Boller& Ch [2], las tarjetas de potencia de motores y de relevadores son idénticas, la diferencia en este control es que se utilizó una microcomputadora más moderna modelo “Beaglebone” que cuenta con sistema operativo Linux y un esquema de programación totalmente distinto al utilizado por los microcontroladores “Rabbit”. La ventaja de utilizar el “Beaglebone” es que el sistema operativo Linux permite depurar los programas y diagnosticar fallas a distancia, además la programación se realiza utilizando lenguajes de alto nivel como Python que reducen el tiempo de desarrollo.

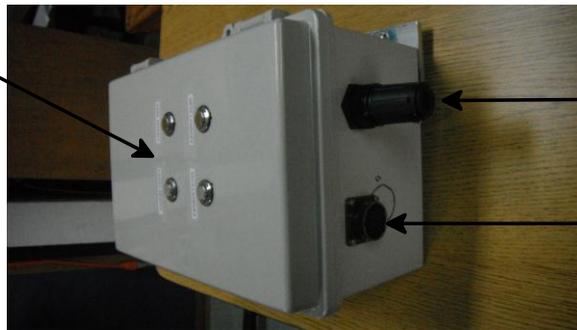
El diseño consta de tres tarjetas electrónicas: la primera es una tarjeta para manejar la potencia de los motores, de la cual se utilizaron dos en el control; la segunda para manejar el relevador que enciende la lámpara de comparación y lee la botonera manual y la tercera es una tarjeta de acoplamiento que interconecta a la microcomputadora con las otras tarjetas. La figura 7 muestra fotografías del control, las tarjetas electrónicas se manufacturaron en el laboratorio de electrónica de Ensenada. El maquinado de la caja y los soportes para las tarjetas se construyeron en el taller mecánico de Ensenada.



- Tarjeta de relevadores y entradas opto-acopladas.
- Tarjeta de potencia del motor de lámpara.
- Tarjeta de potencia del motor de rendija.
- Tarjeta de la microcomputadora Beaglebone.
- Fuente +5V, +12V

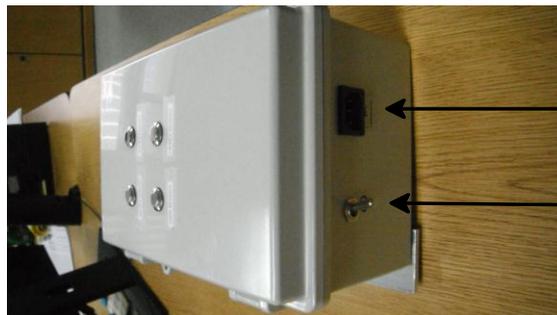
a)

Botones para operación manual.



- Conector del mecanismo de lámpara
- Conector del mecanismo de rendija

b)



- Conector para 110V
- Interruptor de encendido

c)

**Figura 7. Vistas del control de rendija y lámpara del espectrógrafo Echelle. A) Interior, b) Conectores y botonera, c) Interruptor y conexión de alimentación.**

Sobre la caja del control se colocaron cuatro botones para mover los mecanismos de manera manual, dos conectores de señales uno para el mecanismo de rendija y el otro para el mecanismo de lámpara, la información sobre las señales en estos conectores se muestra en el apéndice C.

Las siguientes secciones detallan el diseño de cada una de las tarjetas y los programas desarrollados.

### 3.1 EL MICROCONTROLADOR BEAGLEBONE.

El diseño del controlador está basado en una microcomputadora “Beaglebone” que cuenta con las siguientes características:

1. Procesador ARM Cortex-A8 a 720MHz con 256 MB de RAM.
2. Puerto Ethernet.
3. Puerto USB.
4. Conectores de expansion con 70 líneas de entrada-salida.
5. MicroSD Card Slot.
6. Puerto HDMI.
7. Memoria Flash.
8. Sistema operativo Linux.

De estas características, las más importantes para nuestra aplicación son: el puerto Ethernet, el sistema operativo y los conectores de expansión.

Para interconectar la Microcomputadora con las tarjetas de manejo de potencia de motores y relevadores se diseñó una tarjeta de expansión, su diagrama esquemático y mapa de componentes se muestra en la sección 6.1 del apéndice A. Básicamente se trata de una tarjeta de distribución de señales que contiene solo conectores. En el esquemático se muestran los puertos de expansión de la microcomputadora P8 y P9 así como las señales asignadas a los contactos que fueron ocupados en esta aplicación.

#### 3.1.1 ASIGNACION DE PATAS

La tabla 1 muestra la asignación de patas de la microcomputadora Beaglebone para esta aplicación. La primera columna muestra los nombres asignados a las señales, la segunda columna muestra el nombre del puerto de expansión que corresponde a cada señal. Cada pata es configurable y puede adoptar uno de siete modos de configuración de acuerdo a lo señalado en el manual de referencia para el Beaglebone [3], la configuración se realiza a través del registro multiplexor de cada pata. En la tabla 1 la columna “multiplexor” muestra la dirección del registro multiplexor correspondiente a cada pata y la columna “Dato” muestra el valor en hexadecimal que se escribió en cada registro multiplexor para la configuración, este valor se forma tomando en cuenta la información mostrada en la tabla 9-1 del manual del microprocesador [4].

Tabla 1. Asignación de patas de la microcomputadora Beaglebone.

SEÑAL	PUERTO	MULTIPLEXOR	DATO
REL1	P8_19	oX20	oXoF
REL2	P8_13	oX24	oXoF
SW0_2	P8_37	oXC0	ox2F
SW1_2	P8_17	oX2C	oX2F
SW2_2	P8_29	oXE4	oX2F
SW3_2	P8_18	oX8C	oX2F
SW0_3	P8_31	oXD8	oX2F
SW1_3	P8_35	oXDo	oX2F
SW2_3	P8_32	oXDC	oX2F
SW3_3	P8_33	oXD4	oX2F
BRAKE1_2	P8_39	oXB8	oXoF
BRAKE2_2	P8_40	oXBC	oXoF
DIR1_2	P8_42	oXB4	oXoF
DIR2_2	P8_41	oXB0	oXoF
SW0_1	P9_26	oX180	oX2F
SW1_1	P9_11	oX70	oX2F
SW2_1	P9_13	oX74	oX2F
SW3_1	P9_17	oX15C	oX2F
DIR1_1	P9_22	oX150	oXoF
DIR2_1	P9_21	oX154	oXoF
PWM1_1	P9_14	oX48	oXoF
PWM2_1	P9_16	oX4C	oXo6
PWM1_2	P9_31	oX190	oXo1
PWM2_2	P9_29	oX194	oXo1
BRAKE1_1	P9_18	oX158	oXoF
BRAKE2_1	P9_24	oX184	oXoF

El procedimiento para configurar los multiplexores es mediante un script donde se colocan los datos de las columnas 3 y 4 de la tabla 1. El script tiene terminación “.dts”, en el apéndice B se muestra el listado del script generado para la tarjeta de expansión de nuestra aplicación denominado “echelle-ooAo.dts”. Este scrip debe compilarse e instalarse en el núcleo del sistema operativo para que se reflejen los cambios en los multiplexores. El script se compila con un programa llamado “dts” y que se incluye en la distribución de Linux para el Beaglebone. La compilación se realiza tecleando la siguiente instrucción desde una ventana de comandos:

```
dts -O dtb -o /lib/firmware/echelle-00A0.dtbo -b 0 -@ echelle-00A0.dts
```

Y se inserta en el núcleo del sistema operativo tecleando la siguiente instrucción:

```
echo echelle > /sys/devices/bone_capemgr.8/slots
```

Cabe mencionar que cada que se reinicializa el Beaglebone es necesario ejecutar el comando anterior. El Beaglebone está configurado para hacer esto de manera automática al arranque del sistema operativo (Ver la sección: Servicio de autoarranque).

### 3.1.2 PROGRAMAS DESARROLLADOS.

Para el manejo de los mecanismos de rendija y lámpara se desarrolló una biblioteca de funciones y un programa de aplicación. La biblioteca denominada “echelle\_lib.c” esta echa en lenguaje C y contiene las funciones básicas que interaccionan directamente con los puertos de entrada salida y los registros internos de configuración. El programa de aplicación se denomina “echelle.py”, está hecho en lenguaje Python y contiene las funciones que manejan el socket de comunicación y las funciones que realizan secuencias para realizar tareas elaboradas. Por ejemplo para encender la lámpara primero hay que introducir la bayoneta y esperar a que este dentro para después encender la lámpara. El programa echelle.py utiliza las funciones disponibles en la biblioteca echelle\_lib.c, esto es posible compilando la biblioteca como un objeto y llamándola desde el programa de Python.

#### 3.1.2.1 LA BIBLIOTECA DE FUNCIONES.

La biblioteca denominada “echelle\_lib.c” contiene funciones independientes que realizan tareas específicas, interactúan directamente con los registros de 16 y 32 bits del controlador asociados a la configuración y manejo de los puertos de entrada-salida. El programa realiza un mapeo de memoria para direccionar los registros de configuración. Las funciones disponibles en la biblioteca se listan en la tabla 2.

Tabla 2. Funciones disponibles en el programa de bajo nivel “echelle\_lib.c”.

Función	Descripción
configura()	Realiza la configuración de los puertos de entrada-salida, y los módulos que generan las señales moduladas en ancho de pulso “PWM” para regular la potencia de los motores de lámpara y rendija.
mueve_un_paso(int sentido)	Mueve el motor de rendija en un paso en el sentido indicado por el argumento <i>sentido</i> , si <i>sentido</i> es igual a 1 el motor gira en las manecillas del reloj y en sentido contrario en otro caso.
gira_motor_lampara(int sentido)	Gira el motor de la lámpara en el sentido indicado por el argumento <i>sentido</i> , si <i>sentido</i> es igual a 1 el motor gira en las manecillas del reloj y en sentido contrario en otro caso.
alto_motor_rendija()	Apaga el motor de rendija.
alto_motor_lampara()	Apaga el motor de lámpara.
pwm1_frec(int frec)	Define la frecuencia de la señal modulada en ancho de pulso PWM que controla la potencia del motor de rendija.
pwm0_frec(int frec)	Define la frecuencia de la señal modulada en ancho de pulso PWM que controla la potencia del motor de lámpara.
pwm0_setA(int percent)	Ajusta el voltaje aplicado al motor de la lámpara, el argumento <i>percent</i> va de 0 a 100 donde 0 es voltaje cero y 100 es 12V.

pwm1_setA(int percent)	Ajusta el voltaje aplicado a la primer bobina del motor de rendija, el argumento <i>percent</i> va de 0 a 100 donde 0 es voltaje cero y 100 es 12V.
pwm1_setB(int percent)	Ajusta el voltaje aplicado a la segunda bobina del motor de rendija, el argumento <i>percent</i> va de 0 a 100 donde 0 es voltaje cero y 100 es 12V.
lee_sw_origen()	Regresa el estado del interruptor de inicio del mecanismo de rendija.
lee_sw_limite()	Regresa el estado del interruptor límite del mecanismo de rendija.
lee_sw_dentro()	Regresa el estado del interruptor que limita la carrera al meter la bayoneta del espejo del sistema de lámpara.
lee_sw_fuera()	Regresa el estado del interruptor que limita la carrera al sacar la bayoneta del espejo del sistema de lámpara.
lee_boton_rendija_mas()	Regresa el estado del botón etiquetado como “ABRE RENDIJA”.
lee_boton_rendija_menos()	Regresa el estado del botón etiquetado como “CIERRA RENDIJA”.
lee_boton_fuera()	Regresa el estado del botón etiquetado como “SACA LAMPARA”.
lee_boton_dentro()	Regresa el estado del botón etiquetado como “METE LAMPARA”.
enciende_lampara()	Activa el relevador que enciende la fuente de la lámpara.
apaga_lampara()	Desactiva el relevador que enciende la fuente de la lámpara.
estatus_lampara()	Regresa el estado de la señal que activa el relevador que enciende la fuente de la lámpara.

### 3.1.2.2 EL PROGRAMA DE CONTROL EN LENGUAJE PYTHON.

El programa “echelle.py” se hizo en lenguaje Python, su estructura se muestra en la figura 8. Básicamente está formado por cuatro hilos de programa “Threads” que se ejecutan en paralelo: Comunicación, Control, Atención a botones y Actualización de estado, la comunicación entre “Threads” se realiza mediante la “Cola de instrucciones” y dos variables globales: “alto” y “estatus”.

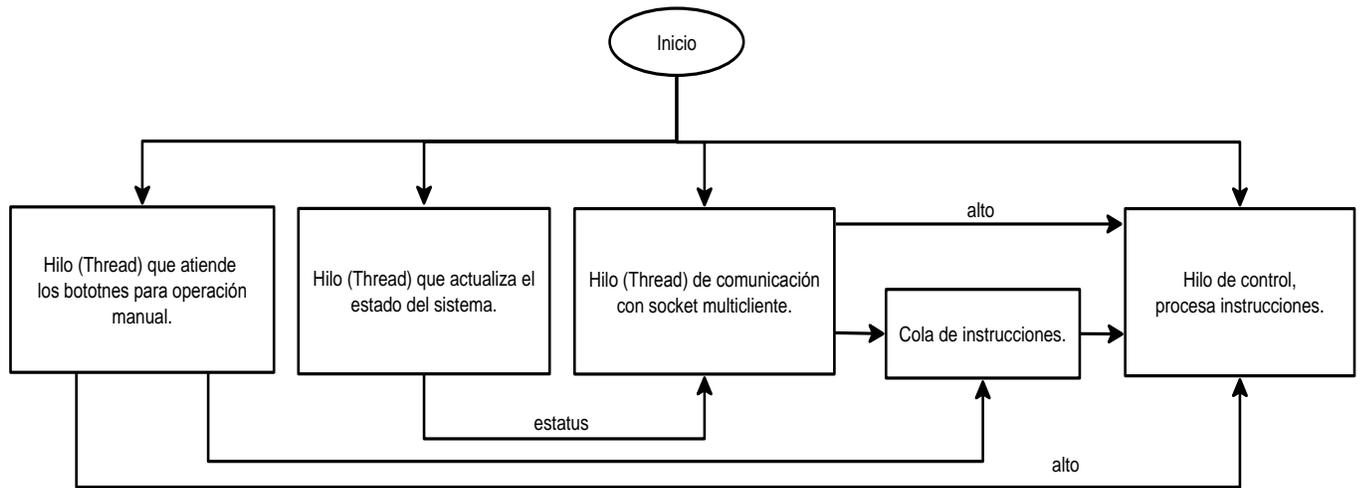


Figura 8. Estructura del programa “echelle.py”.

El hilo de comunicación abre un socket TCP servidor en la dirección 192.168.0.24 y puerto 2020, el socket es multi-cliente de tal manera que puede aceptar conexiones de varios clientes a la vez. Las instrucciones recibidas por el socket son puestas en la cola de instrucciones para ser ejecutadas por el hilo de control. Las instrucciones programadas y que pueden ser recibidas por este socket se listan en la tabla 3.

El hilo de Atención a botones revisa continuamente si se presionó un botón, una vez que detecta un evento, se coloca la instrucción correspondiente en la cola de instrucciones para que el hilo de control la ejecute.

El hilo de control saca instrucciones de la cola y las ejecuta, para ello utiliza las instrucciones de la biblioteca “echelle\_lib.c” para activar y supervisar la ejecución de los movimientos. En todos los ciclos de espera se programaron temporizadores para evitar que el programa se pierda en un ciclo infinito. La bandera “alto” siempre es monitoreada y cuando se recibe se suspende cualquier tarea que se esté realizando y se vacía la cola de instrucciones.

Tabla 3. Instrucciones que recibe el programa echelle.py.

Mando	Descripción
MUEVE_RENDIJA xxx	Lleva la rendija a la apertura especificada por xxx en micrómetros.
ENCIENDE_LAMPARA	Mete la Bayoneta del espejo y enciende la lámpara.
APAGA_LAMPARA	Saca la bayoneta del espejo y apaga la lámpara.
ALTO	Cancela cualquier operación que se esté ejecutando y vacía la cola de entrada.
ESTATUS	Regresa la cadena con el estado del sistema. La cadena tiene el siguiente formato “LAMPARA ON   RENDIJA xxx   ERROR”. La cadena se divide en tres secciones separadas por el caracter ‘ ’. La primer sección indica el estado de la lámpara que puede ser ON u OFF; la segunda sección indica la posición de la rendija xxx, la tercer sección es un mensaje de error que puede tomar una de las tres siguientes posibilidades: “NORMAL”, “ERROR INICIO” y “ERROR de SW”. “NORMAL” indica que no hay problema, “ERROR de INICIO” indica que el interruptor de inicio del mecanismo de rendija no fue detectado, y “ERROR de SW” indica que alguno de los dos

	interruptores que limitan la carrera de la bayoneta del espejo no fue detectado.
--	--

### 3.1.3 EL SERVICIO DE AUTOARRANQUE.

El sistema operativo Linux distribución Angstrom que viene instalado en las microcomputadoras Beaglebone requiere de la elaboración de un script para que se ejecute el programa de aplicación al arranque del sistema. Este script se denomina servicio del sistema “System Service”. Angstrom utiliza el demonio de administración del sistema “systemd” para manejar todos los servicios, para cambiar la configuración del systemd se utiliza el comando “systemctl”, mediante el cual es posible: habilitar, deshabilitar, arrancar, parar, restablecer y consultar el estado de los servicios del sistema.

En el apéndice D se muestra el contenido del archivo de servicio denominado “run\_echelle.service”, este archivo se debe colocar en el directorio “/lib/systemd/system” y se habilita con la siguiente instrucción:

```
systemctl enable run_cupula_2m.service
```

De esta forma al reinicializar el Beaglebone el programa se ejecuta al arranque del sistema.

### 3.2 LA TARJETA DE POTENCIA PARA MOTORES.

Para el manejo de los motores de rendija y lámpara se utilizó una tarjeta desarrollada previamente, con capacidad de manejo de dos motores de corriente directa, la tarjeta cuenta con dos puentes H modelo LMD18200 con entrada para señal modulada en ancho de pulso “PWM” y contiene opto-acopladores para recibir 4 entradas de interruptor límite. Las señales de entrada y salida digitales de esta tarjeta se canalizan mediante un conector 5X2 para cable plano, de tal manera que se puede controlar utilizando cualquier tipo de microcontrolador, lo que hace de esta tarjeta una herramienta versátil cuando se trata de controlar mecanismos motorizados.

En el control de rendija y lámpara del Echelle se utilizó un par de estas tarjetas, la primera para mover el motor de pasos del mecanismo de rendija y para sensar el estado de los interruptores límite y de posición de origen, su diagrama esquemático y mapa de componentes se muestra en la sección 6.2 del apéndice A, en el esquemático se han identificado con etiquetas las señales del conector de salida que se utilizan en el mecanismo de rendija. La segunda tarjeta se utilizó para mover el motor de la bayoneta de la lámpara y sensar el estado de los interruptores de posición dentro y fuera de la bayoneta, su diagrama esquemático y mapa de componentes se muestra en la sección 6.3 del apéndice A. En este esquemático también se identificó con etiquetas las señales que se utilizaron en el mecanismo de lámpara. En el esquemático también es evidente que solo se está utilizando la mitad de la tarjeta, la otra mitad se deja disponible para la futura incorporación de nuevos mecanismos.

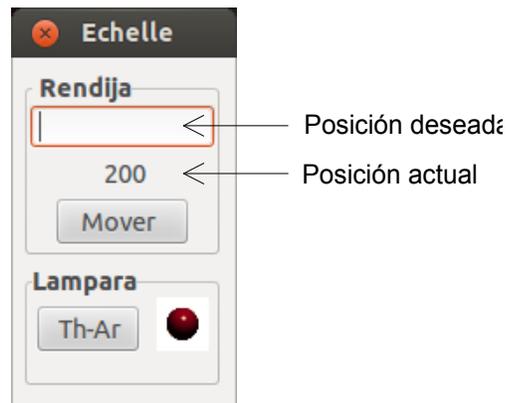
### 3.3 LA TARJETA DE RELEVADOR Y ENTRADAS OPTOACOPLADAS.

La tarjeta de relevadores utilizada, al igual que la tarjeta para el manejo de motores, es una tarjeta desarrollada previamente con capacidad para manejar dos relevadores y recibir 4 entradas opto-acopladas, las señales digitales de entrada y salida se canalizan a través de un

conector 5x2 para cable plano, por lo que puede ser acoplada a cualquier microcontrolador. Su diagrama esquemático y mapa de componentes se muestra en la sección 6.4 del apéndice A. De esta tarjeta se utiliza un relevador para el encendido de la fuente de la lámpara y las cuatro entradas para el sensado de los botones de operación manual, en el esquemático se ha señalado con etiquetas la asignación de señales utilizadas en esta aplicación.

#### 4. LA INTERFAZ DE USUARIO.

Se desarrolló un programa gráfico para la operación amigable desde la PC de usuario del telescopio de 2.1m que en la actualidad es Sonaja. El programa gráfico se hizo en GTK y se utilizó el módulo Pygtk para desarrollar el código en lenguaje Python. La figura 9 muestra la ventana del programa. La sección superior es para manejar la rendija y la inferior para la lámpara. Si se requiere mover la rendija solo hay que insertar la posición deseada y presionar el botón “Mover”. Para encender la lámpara se presiona el botón “Th-Ar”, cuando el foquito enciende significa que la lámpara está encendida, para apagarla se presiona nuevamente el botón “Th-Ar”.



*Figura 9. Interfaz de usuario.*

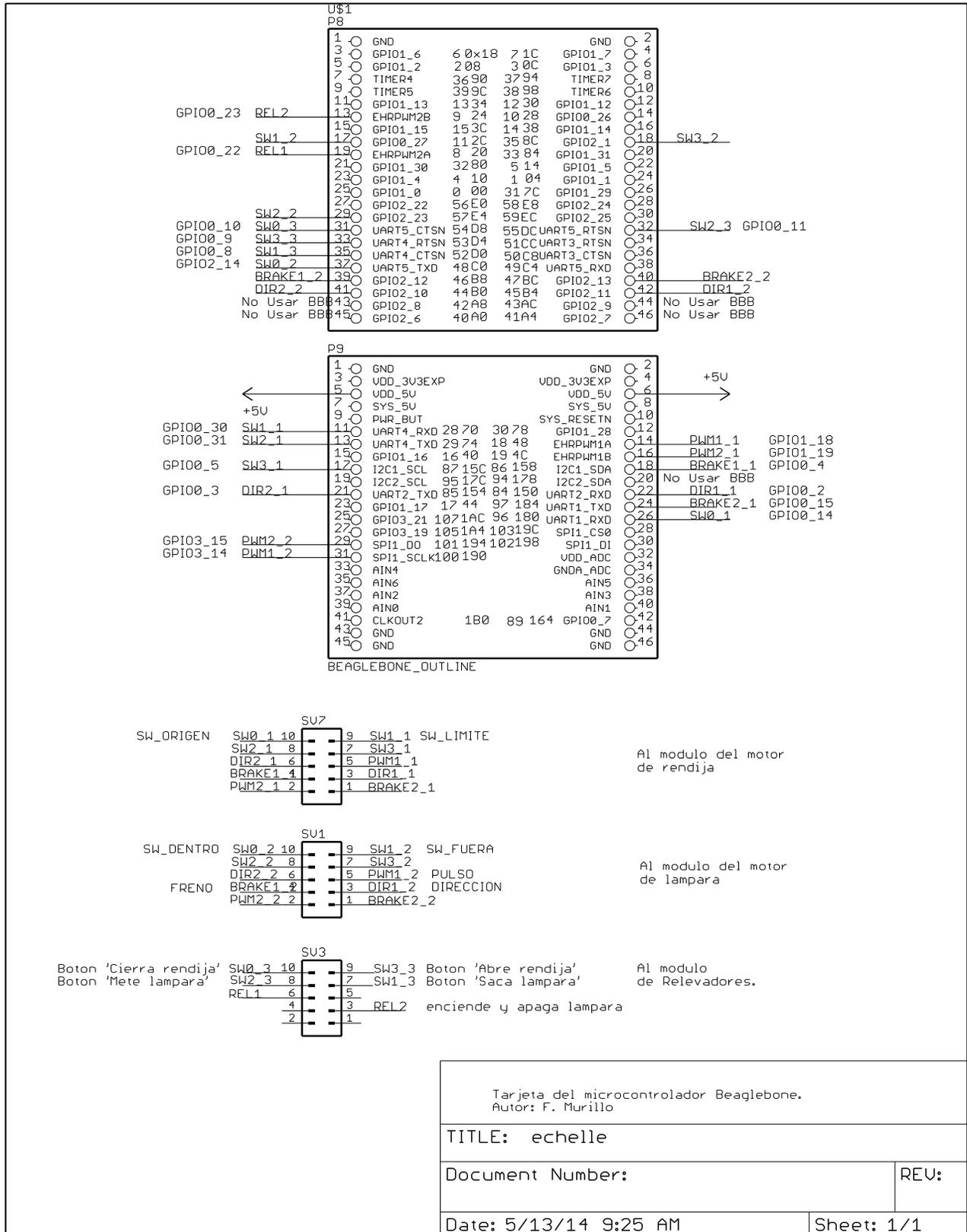
#### 5. PRUEBAS Y RESULTADOS.

El mecanismo fue instalado y probado en el instrumento funcionando de manera satisfactoria, la precisión de posicionado de la rendija fue del  $\pm 1 \mu\text{m}$ , medido con la escala del micrómetro de apertura de rendija. Este documento forma parte del protocolo de entrega y puesta en marcha de desarrollos en el Observatorio, por lo que se pondrá en funcionamiento próximamente.

## 6. APÉNDICE A. DIAGRAMAS ELECTRÓNICOS.

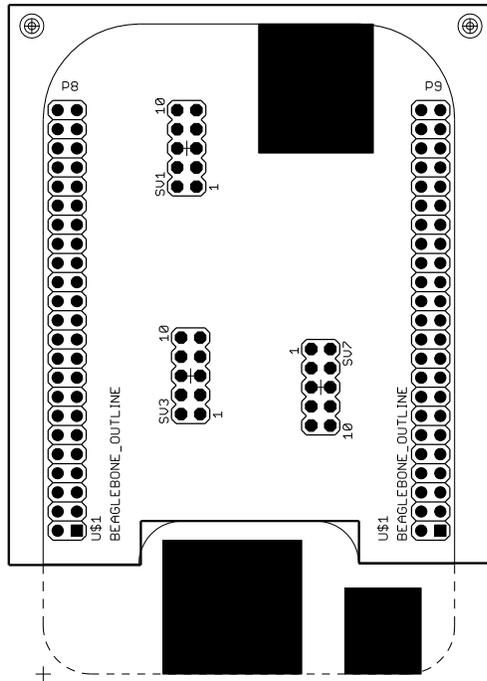
### 6.1 TARJETA DEL MICROCONTROLADOR BEAGLEBONE.

#### 6.1.1 ESQUEMÁTICO.



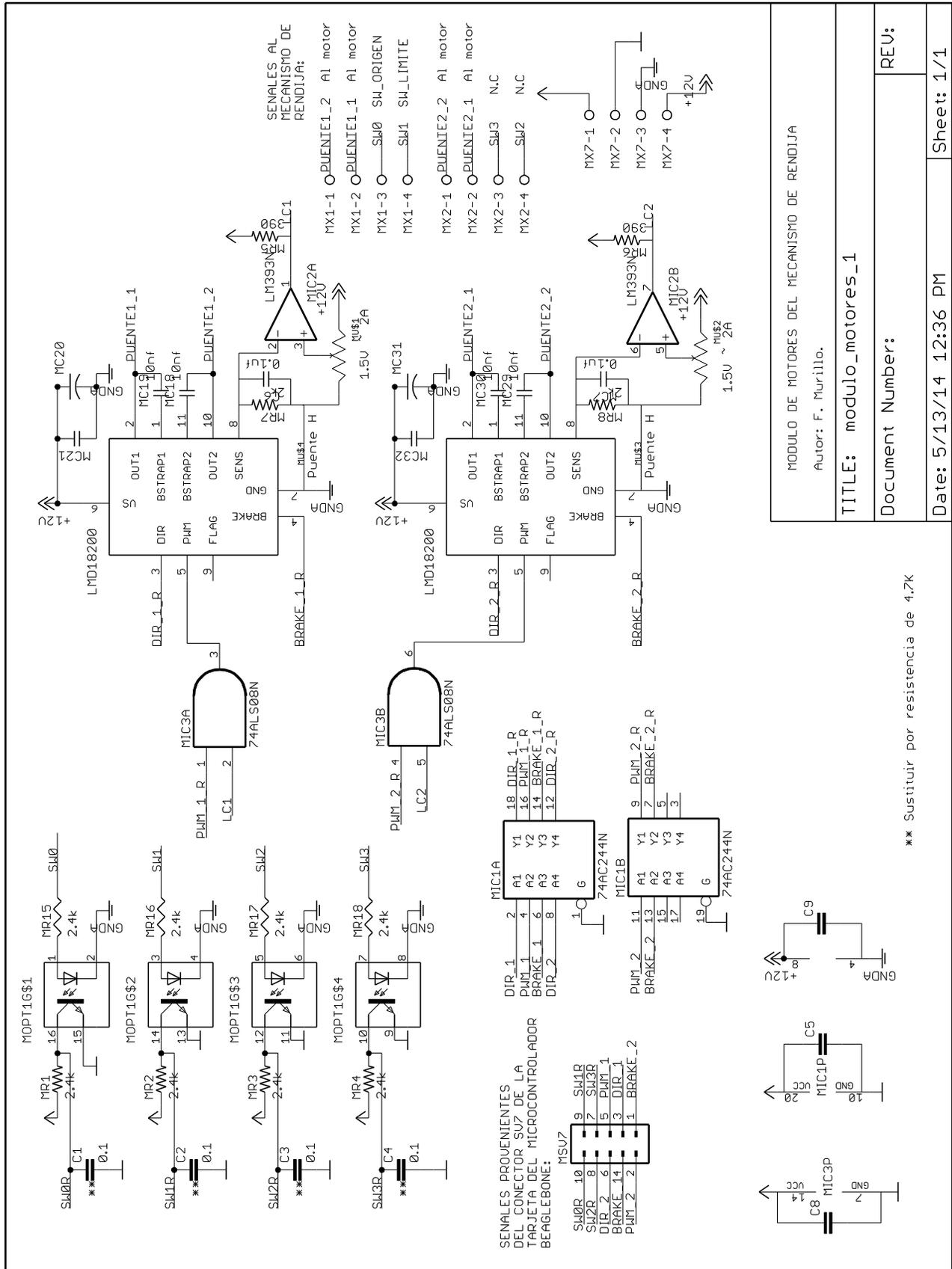
Tarjeta del microcontrolador Beaglebone. Autor: F. Murillo	
TITLE: echelle	
Document Number:	REV:
Date: 5/13/14 9:25 AM	Sheet: 1/1

### 6.1.2 MAPA DE COMPONENTES.

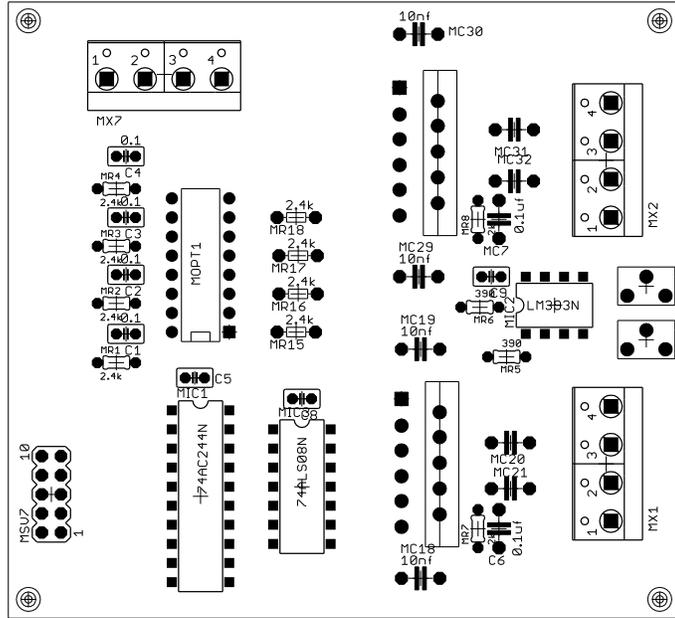


## 6.2 TARJETA DE POTENCIA PARA EL MECANISMO DE RENDIJA.

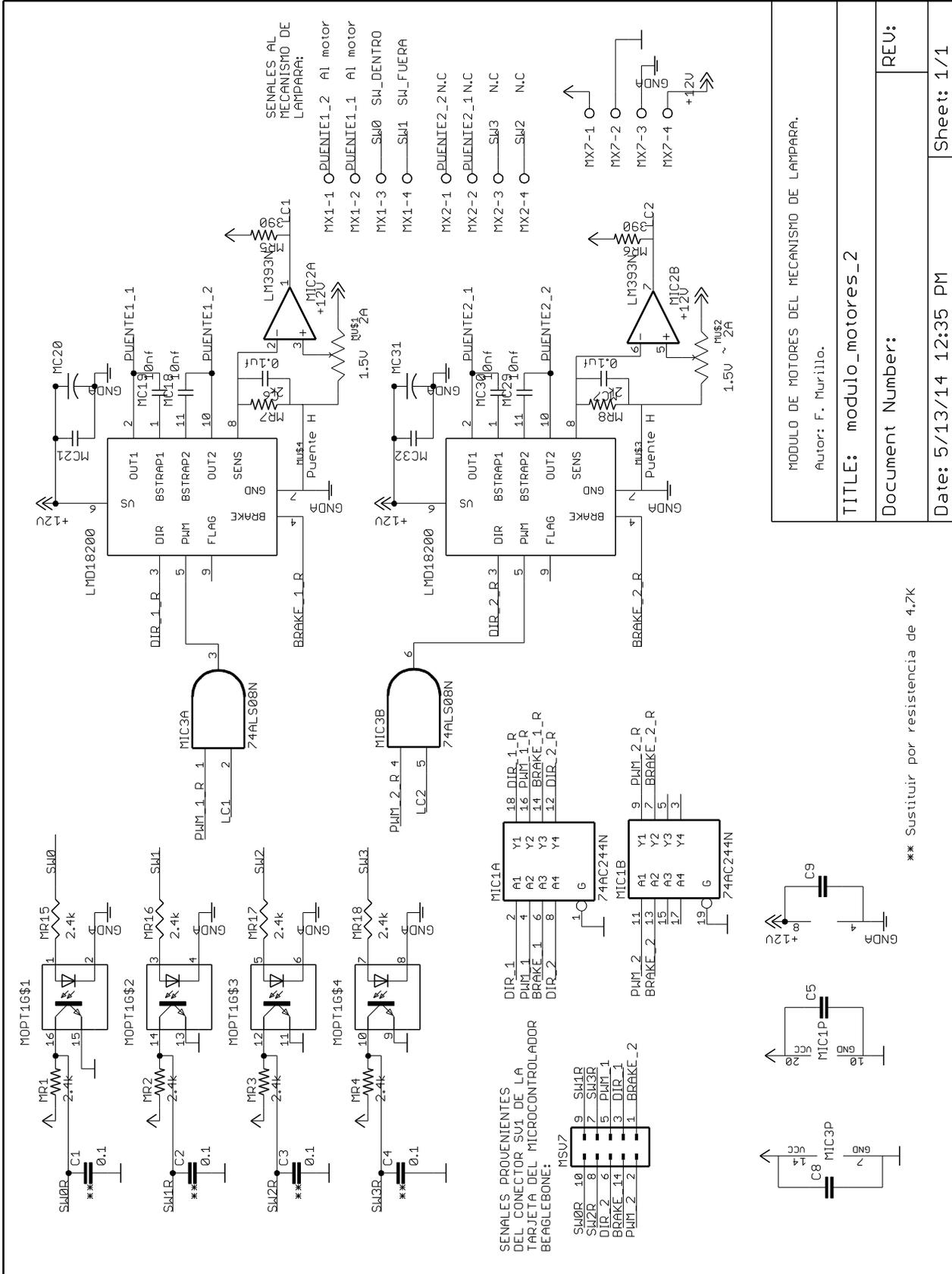
### 6.2.1 ESQUEMÁTICO.



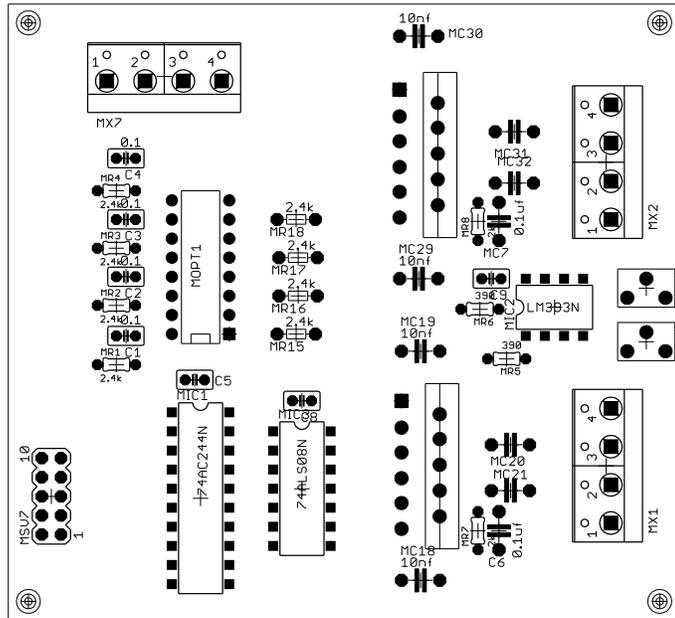
### 6.2.2 MAPA DE COMPONENTES.



### 6.3 TARJETA DE POTENCIA PARA EL MECANISMO DE LAMPARA. 6.3.1 ESQUEMÁTICO.

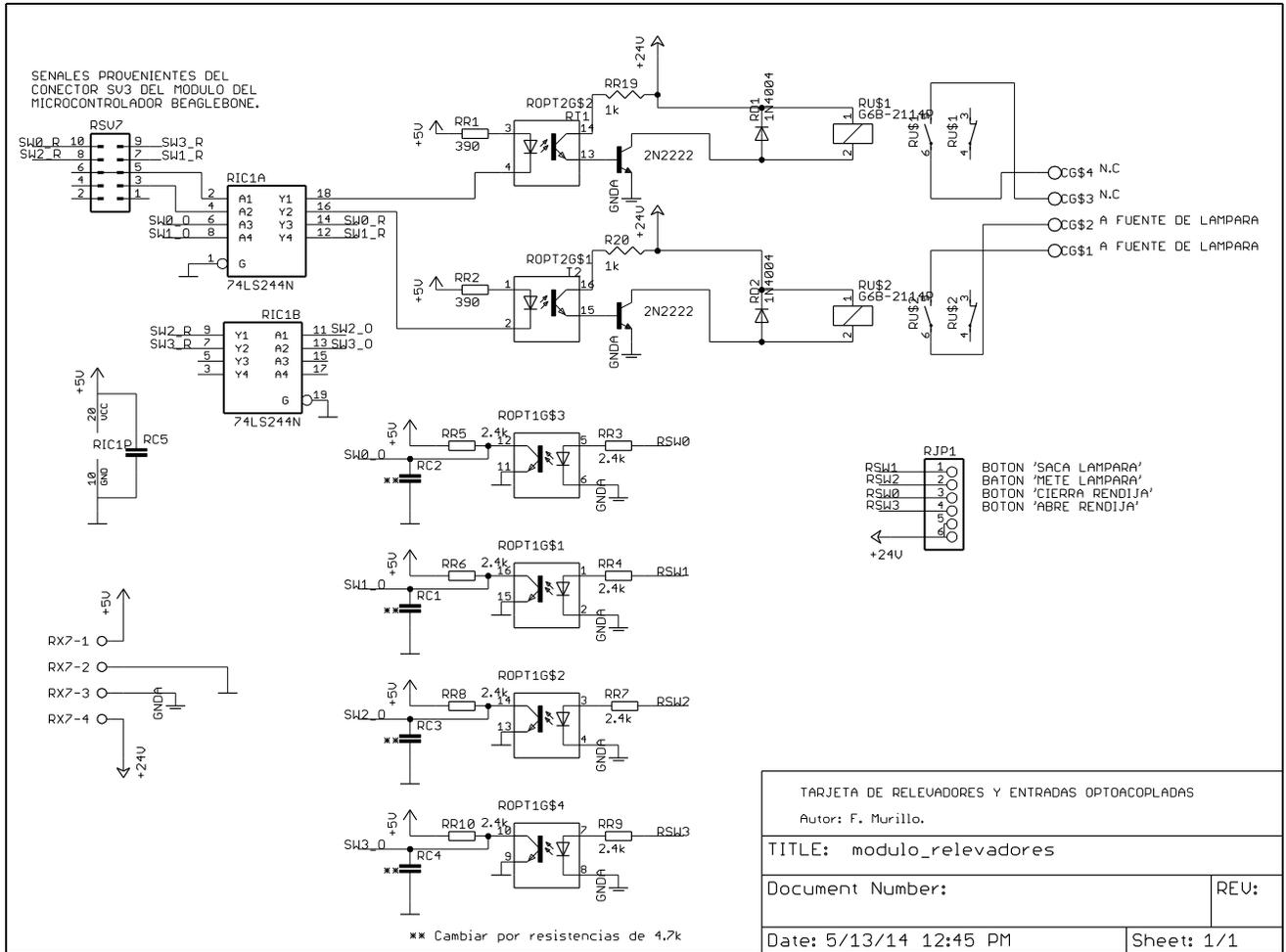


### 6.3.2 MAPA DE COMPONENTES.

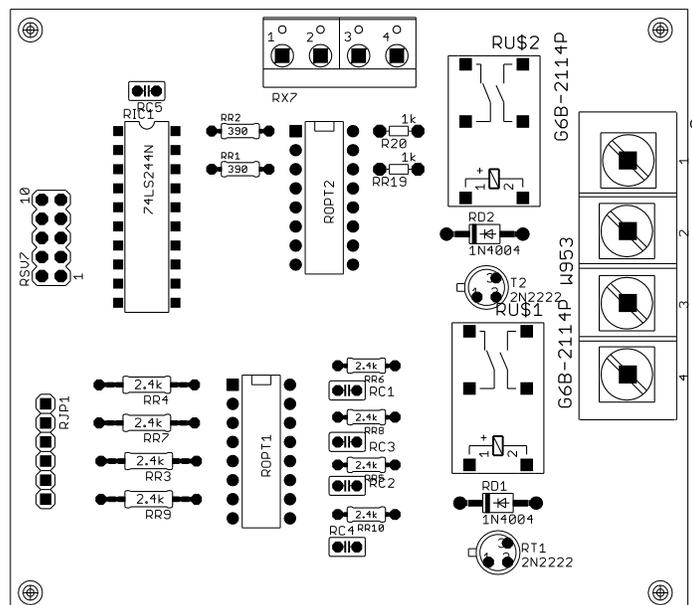


## 6.4 TARJETA DE RELEVADOR Y ENTRADAS OPTOACOPLADAS.

### 6.4.1 ESQUEMÁTICO.



### 6.4.2 MAPA DE COMPONENTES



7. **APENDICE B. ARCHIVO DE CONFIGURACIÓN DE PATAS DE LA MICROCOMPUTADORA BEAGLEBONE.**

```
/*
 * Copyright (C) 2013 CircuitCo
 * Copyright (C) 2013 Texas Instruments
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License version 2 as
 * published by the Free Software Foundation.
 */
/dts-v1/;
/plugin/;

/ {
    compatible = "ti,beaglebone", "ti,beaglebone-black";

    /* identification */
    part-number = "ECHELLE";
    version = "00A0";

    /* state the resources this cape uses */
    exclusive-use =
        /* the pin header uses */
        "P9.20";

    fragment@0 {
        target = <&am33xx_pinmux>;
        __overlay__ {
            ECHELLE: pinmux_ECHELLE_pins {
                pinctrl-single,pins = <
                    0x024 0x0f
                    0x020 0x0f
                    0x0d8 0x2f
                    0x0d4 0x2f
                    0x0d0 0x2f
                    0x0b8 0x0f
                    0x0b0 0x0f
                    0x0e4 0x2f
                    0x0a0 0x2f
                    0x0dc 0x2f
                    0x0bc 0x0f
                    0x0b4 0x0f
                    0x08c 0x2f
                    0x02c 0x2f
                    0x190 0x01
                    0x194 0x01
                >
            };
        };
    };
};
```

```
                                0x04c 0x06
                                0x184 0x0f
                                0x150 0x0f
                                0x048 0x06
                                0x154 0x0f
                                0x158 0x0f
                                0x180 0x2f
                                0x070 0x2f
                                0x074 0x2f
                                0x15c 0x2f>;
                                };
                                };
};

fragment@1 {
    target = <&ocp>;
    __overlay__ {
        test_ECHELLE {
            compatible = "bone-pinmux-helper";
            pinctrl-names = "default";
            pinctrl-o = <&ECHELLE>;
            status = "okay";
        };
    };
};
};
```

## 8. APENDICE C. SEÑALES EN LOS CONECTORES.

### 7.1 CONECTOR DEL MECANISMO DE RENDIJA.

Tabla 4. Señales en el conector del mecanismo de rendija.

Contacto	Señal
A	Puente1_2, Salida a la bobina 1 del motor de pasos.
B	Puente1_1, Salida a la bobina 1 del motor de pasos.
C	Swo, Entrada de señal del interruptor de inicio.
D	Sw1, Entrada de señal del interruptor límite.
E	Puente2_2, Salida a la bobina 2 del motor de pasos.
F	Puente2_1, Salida a la bobina 2 del motor de pasos.
G	Salida de voltaje +12V para el interruptor de inicio.
H	Salida de voltaje +12V para el interruptor de límite.

### 7.2 CONECTOR DEL MECANISMO DE LÁMPARA.

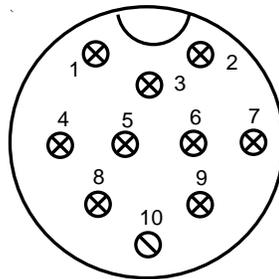


Figura 10. Distribución de contactos del conector del mecanismo de lámpara.

Tabla 5. Señales en el conector del mecanismo de lámpara.

Contacto	Señal
1	A la fuente de la lámpara.
2	A La fuente de la lámpara.
3	N.C
4	Puente1_1, al motor de bayoneta.
5	Swo, Interruptor límite, bayoneta Dentro.
6	Sw1, Interruptor límite, bayoneta Fuera.
7	Puente1_2, al motor de bayoneta.
8	N.C
9	N.C
10	N.C

## 9. APENDICE D. SERVICIO DE AUTOARRANQUE DEL SISTEMA.

### 8.1 CONTENIDO DEL ARCHIVO “RUN\_ECHELLE.SERVICE”

[Unit]

Description=echelle

[Service]

WorkingDirectory=/home/root/

ExecStart=/home/root/run\_echelle.sh

SyslogIdentifier=echelle

Restart=on-failure

RestartSec=5

[Install]

WantedBy=multi-user.target

### 8.2 CONTENIDO DEL ARCHIVO “RUN\_ECHELLE.SH”.

```
#!/bin/sh
```

```
echo echelle > /sys/devices/bone_capemgr.8/slots
```

```
python echelle.py
```

## 10. REFERENCIAS.

[1] Automatización del Sistema de Iluminación del Espectrógrafo Echelle. F. Murillo, G. Sierra, B. Martínez, J.M. Murillo, G. Guisa, A. Córdova, B. García, E. López, F. Quirós, J.L. Ochoa. CI-2010-09.

[2] Automatización de la rendija del espectrógrafo Boller & Chivens del OAN. F. Murillo, G. Sierra, J.M. Murillo, G. Guisa, B. Martínez, F. Quirós, J.L. Ochoa. RT-2013-01.

[3] BeagleBone Black System Reference Manual. Revision A5.6.

[4] AM335x ARM Cortex-A8 Microprocessors (MPUs), Technical Reference Manual.