Publicaciones Técnicas
pubtecia@astro.unam.mx

Instituto de Astronomía
UNAM

Reporte Técnico RT-2016-03

aceptación: 18-03-2016
recepción: 20-04-2015

Nuevo control electrónico para el espectrógrafo Mezcal.

F. Murillo, H. Serrano, S. Zazueta, J.M. Murillo, F. Quirós, J. Herrera, E. Colorado, E. López, G. Guisa, J.A. López, J. Nuñez, L. Gutiérrez.

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

RESUMEN:

Este trabajo presenta el diseño e implementación de un nuevo control electrónico para el instrumento Mezcal utilizado en el telescopio de 2.1m del OAN. El nuevo diseño está basado en una microcomputadora de tecnología reciente, que facilita la incorporación de nuevas funcionalidades, asi como la operación y monitoreo remoto del instrumento.

Contenido

1. INTRODUCCIÓN

El instrumento Mezcal es un espectrógrafo de alta resolución utilizado de manera regular en el telescopio de 2.1 m del Observatorio Astronómico Nacional. En el año 2000 se hizo un trabajo muy completo de automatización computarizada para aumentar la eficiencia en su operación; como resultado, en los años siguientes, se contó con un instrumento robusto, y con un índice de fallas muy bajo. Hasta el año 2014 se siguió utilizando la misma infraestructura electrónica y de programación, desarrollada en el año 2000, funcionando correctamente. El programa de interfaz de usuario fue desarrollado en lenguaje TCL y la electrónica se basó en la implementación de un microcontrolador ATMEL AT89C52.

En la actualidad, el lenguaje TCL ha caído en desuso debido a que en los últimos cinco años no ha tenido el soporte suficiente. No ha sido así el caso de otros lenguajes como Python, que tienen un desarrollo constante, con módulos actualizados que minimizan el tiempo de desarrollo de aplicaciones.

En el año 2014 surgió el interés por actualizar el programa de usuario. La nueva versión se desarrollaría en lenguaje Python, explotando las nuevas capacidades del lenguaje en la incorporación de nuevas herramientas disponibles para el usuario. Para ello, se trasladaría el instrumento al laboratorio de electrónica de la sede del Instituto de Astronomía en Ensenada.

Dado que el procesador ATMEL AT89C52 ya es obsoleto y su funcionalidad es limitada, se propuso actualizar la electrónica de control utilizando una microcomputadora moderna con sistema operativo y conectividad a red Ethernet. En junio de 2014 se inició el diseño de la nueva electrónica, se construyó en el laboratorio de electrónica del Instituto de Astronomía de Ensenada y fue probada en el instrumento en septiembre del mismo año.

En este documento se describe el diseño de la nueva electrónica y los programas de control desarrollados.

2. EL INSTRUMENTO MEZCAL

El instrumento Mezcal es un espectrógrafo Echelle nebular de alta resolución; corresponde a una configuración de doble paso, de tal forma que el sistema óptico de colimación con el cual se ilumina la rejilla de difracción, funciona como cámara a su regreso. A continuación se listan los elementos que componen al espectrógrafo, desde el telescopio hasta el CCD. La *Figura 1* muestra el esquéma óptico del instrumento.

- Rueda de polarizadores y filtro Red Cutoff (Wheel).
- Carro de Rendijas (Slits).
- Carro de Filtros (Filters).
- Espejo diagonal (Tilting mirror).
- Colimador-Cámara (Collimator).
- Rejilla de difracción o Espejo plano para imagen directa (Grating, Mirror).
- Detector (CCD).

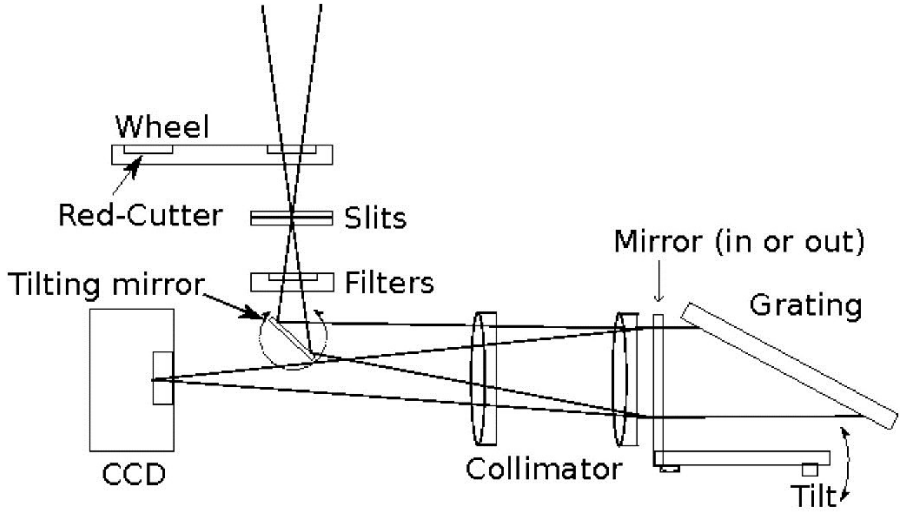


Figura 1: *Esquema del espectrógrafo Mezcal.*

El espectrógrafo posee siete ejes de movimiento: difusor, espejo, filtros, rendijas, polarizadores, rejilla y foco. Cada uno de ellos es impulsado por un motor de corriente directa de 24V.

Los ejes del difusor y del espejo son mecanismos de dos posiciones: dentro y fuera; sus posiciones están retroalimentadas por interruptores límite.

Los ejes de filtros, rendijas y polarizadores son mecanismos de ginebra de 3, 4 y 5 posiciones, respectivamente; su posición es retroalimentada utilizando sensores de proximidad e interruptores de inicializado.

El eje de rejilla cambia la inclinación de la rejilla; su posición está retroalimentada por codificador incremental de cuadratura e interruptor de inicializado.

Por último, el eje de foco se mueve linealmente y su posición se retroalimenta por un potenciómetro de vueltas múltiples.

3. DISEÑO DE LA NUEVA ELECTRÓNICA DE CONTROL

El diseño de la nueva electrónica se apoyó en la infraestructura desarrollada en el año 2000 [1] y que se muestra en el diagrama de la *Figura 2*. Consta de una tarjeta de control basada en el microcontrolador AT89C52 de la compañía Atmel y una tarjeta de relevadores para manejar la potencia de los motores. La tarjeta del microcontrolador cuenta con lo necesario para acondicionar las señales de retroalimentación de la posición de cada eje.

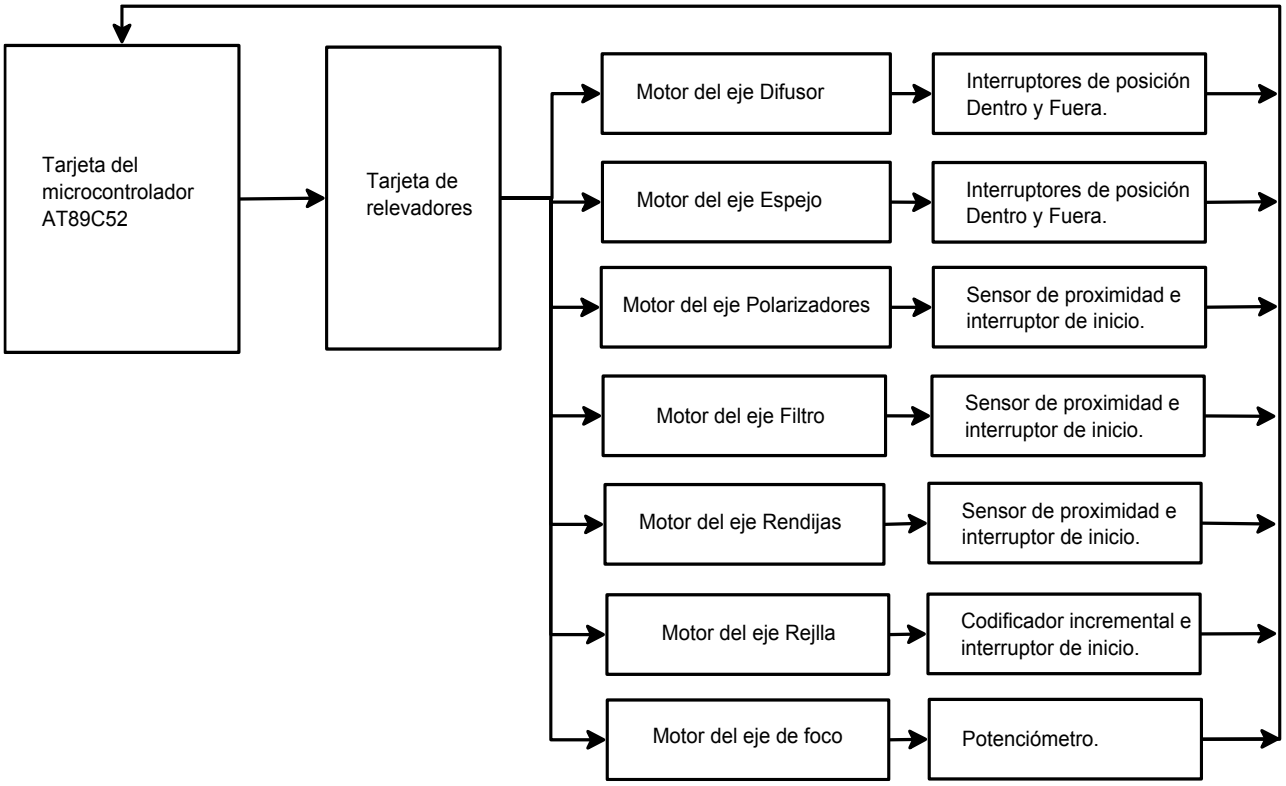


Figura 2: *Diagrama a bloques del control de Mezcal basado en el microcontrolador AT89C52.*

La actualización que se realizó consistió en reemplazar la tarjeta del microcontrolador AT89C52 por una nueva tarjeta basada en una microcomputadora Beaglebone denominada "**Tarjeta de interfaz**". No fue necesario reemplazar lo demás ya que, tanto la tarjeta de relevadores como los sensores, funcionan correctamente.

La *Figura 3* muestra un diagrama a bloques de la estructura de la **Tarjeta de interfaz** y la *Figura 4* muestra una fotografía de la tarjeta instalada en el instrumento. Básicamente se trata de una tarjeta donde se inserta el microcontrolador Beaglebone y posee reforzadores para señales de salida y circuitos de acondicionamiento de señales de entrada. El diseño completo se muestra en el Apéndice A. Se conservaron las medidas, el tipo de conectores y los comandos de control utilizados en la tarjeta del microcontrolador AT89C52, con el fin de intercambiarlas en caso de ser necesario.

En este nuevo diseño se utilizaron los recursos de hardware disponibles en el Beaglebone, como son: el decodificador de cuadratura utilizado para leer la posición del eje de rejilla y el convertidor analógico a digital utilizado para leer la posición del eje de foco. Además, el número de entradas y salidas disponibles en el Beaglebone fueron suficientes para esta aplicación, eliminando la necesidad de utilizar puertos de expansión adicionales. El aprovechamiento de estos recuros facilitó el diseño del circuito impreso.

Otra ventaja del nuevo diseño es la conexión directa a la red Ethernet, eliminando el convertidor Ehternet-RS232 que se utilizaba anteriormente.

El sistema operativo Linux, instalado en el Beaglebone, permite depurar, modificar y respaldar el programa de control de manera remota, lo que también representa una ventaja con respecto a la electrónica anterior, donde cualquier cambio en el programa de control requería el uso de una herramienta de programación para el microcontrolador AT89C52. Para ello era necesario extraer físicamente el microcontrolador de su tarjeta e insertarlo en la herramienta de programación.

Con la implementación del nuevo microcontrolador es posible ejecutar tareas en paralelo, de tal manera que una tarea puede atender la comunicación mientras otra se encarga del control y supervisión de movimientos. Esto representa otra ventaja con respecto al desarrollo anterior, ya que no era posible hacer las dos tareas en paralelo y, si existía una falla en el control de movimientos, se perdía la comunicación y la única forma de reestablecerla era apagando y encendiendo el microcontrolador. El nuevo microcontrolador no se pierde en esta circunstancia y reporta al usuario la ocurrencia de una falla en el control de movimientos.

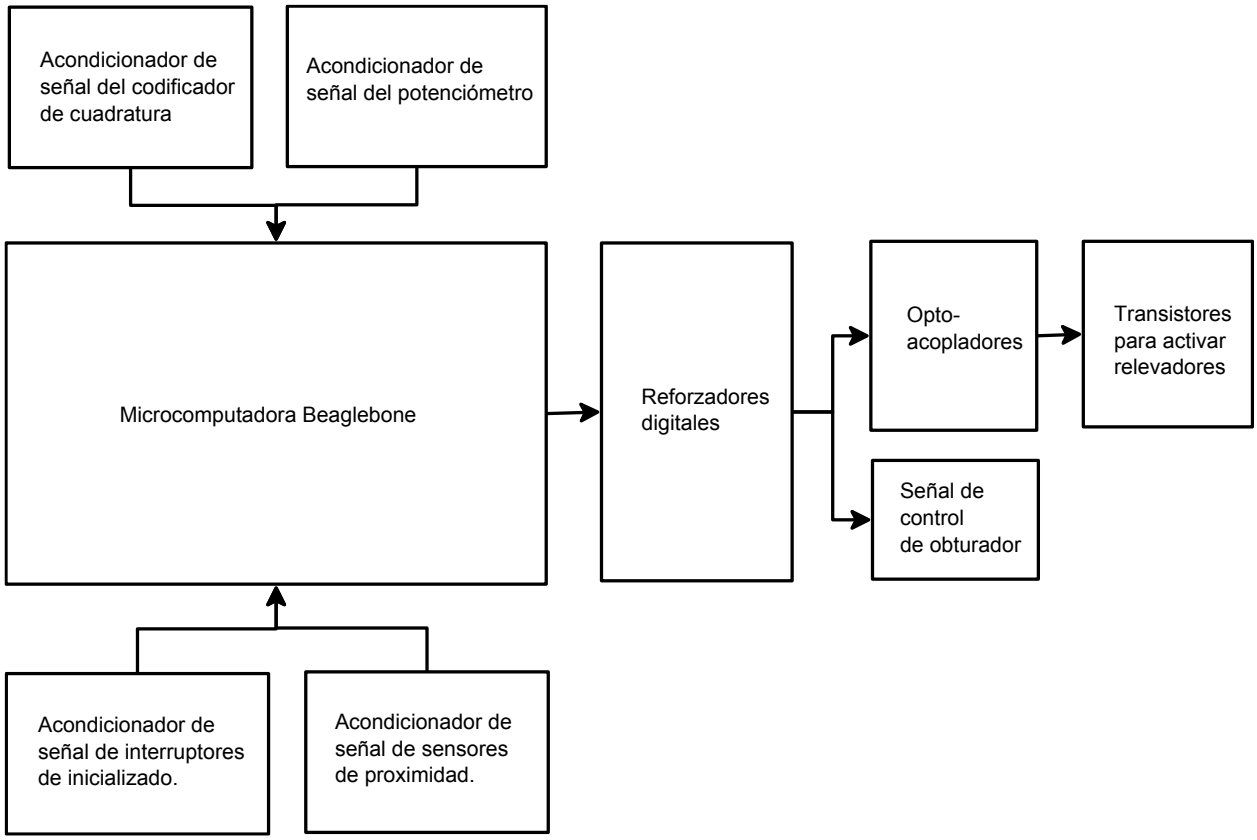


Figura 3: *Tarjeta de interfaz para la microcomputadora Beaglebone.*

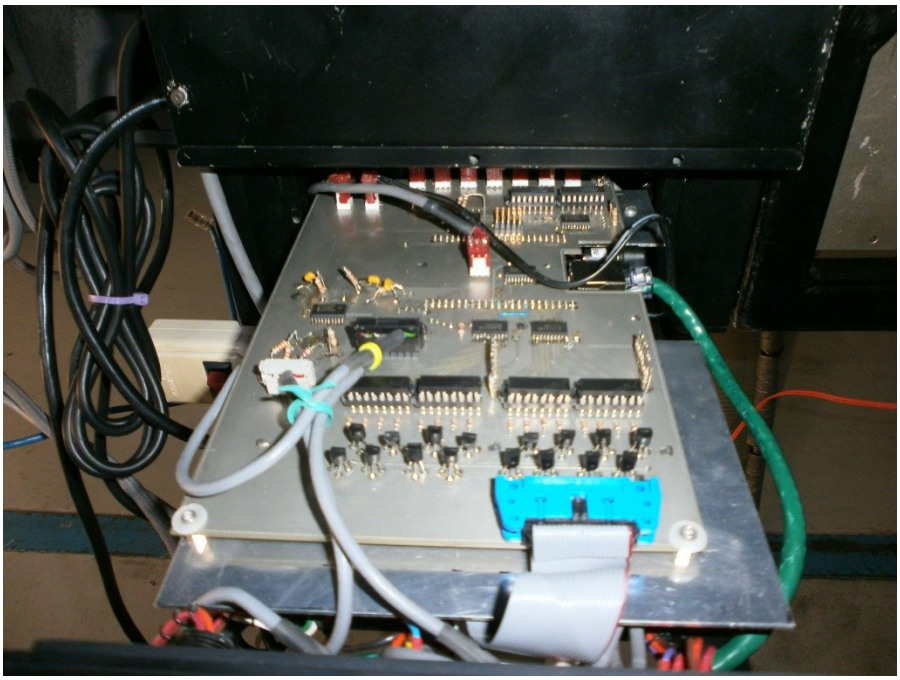


Figura 4: *Vista de la tarjeta de interfaz instalada en el instrumento.*

4. PROGRAMAS DESARROLLADOS

Para el control de los ejes de movimiento del instrumento se desarrollaron dos programas: el programa de control "**mezcal.py**", en lenguaje Python, y el manejador de bajo nivel "**Driver**", en lenguaje C. Ambos programas se ejecutan en el microcontrolador Beaglebone y se describen a continuación.

4.1 PROGRAMA DE CONTROL "mezcal.py"

El progama de control denominado **mezcal.py** se desarrolló en lenguaje Python y se utilizó la biblioteca de manejo de puertos "Adafruit", de código abierto, para accesar los puertos de entrada y salida directamente desde Python.

La *Figura 5* muestra la estructura del programa que consta de ocho hilos de ejecución: uno de comunicación, seis de control de movimiento y uno de actualización de estado. Cada hilo de ejecución funciona como un programa independiente, con sus propias instrucciones y variables; los ocho hilos se arrancan al inicio del programa y se ejecutan en paralelo de manera continua. El hilo de comunicación abre un *socket* de red en el puerto 10001 por donde recibe instrucciones. La Tabla 1 muestra la lista de instrucciones que fueron programadas. Se conservaron las instrucciones definidas en el sistema anterior por compatibilidad; solamente la instrucción `dame_error` es nueva, y es introducida para informar al programa de usuario sobre errores detectados en el posicionamiento de algún eje. En el Apéndice B se muestra una lista de los errores que pueden presentarse.

Cada instrucción recibida es analizada y colocada en la cola de instrucciones del hilo de control correspondiente. Cada hilo de ejecución monitorea continuamente su cola de instrucciones, y cuando detecta la presencia de una instrucción, la extrae y la procesa.

Las instrucciones que procesan los hilos de control son de dos tipos: inicialización y posicionamiento. Si algún error es detectado en un hilo de control, éste inserta un mensaje en la cola de errores para ser enviado a la interfaz de usuario cuando ésta lo requiera.

El hilo que actualiza el estado del sistema forma constantemente una cadena de caracteres con la información de posición de todos los ejes de movimiento. El formato de la cadena, así como la información de los campos que contiene, se muestra en el Apéndice F.

El programa crea un archivo "log" con mensajes generados por todos sus hilos de ejecución. Este archivo es un registro de todos los movimientos realizados. Si se presentan errores de funcionalidad, se registran en el archivo de tal manera que, analizando su contenido, se puede inferir la causa del problema. Esta funcionalidad representa una ventaja con respecto al sistema anterior, donde al presentarse un problema simplemente se apagaba y encendía la electrónica, y no había forma de saber la causa de la falla.

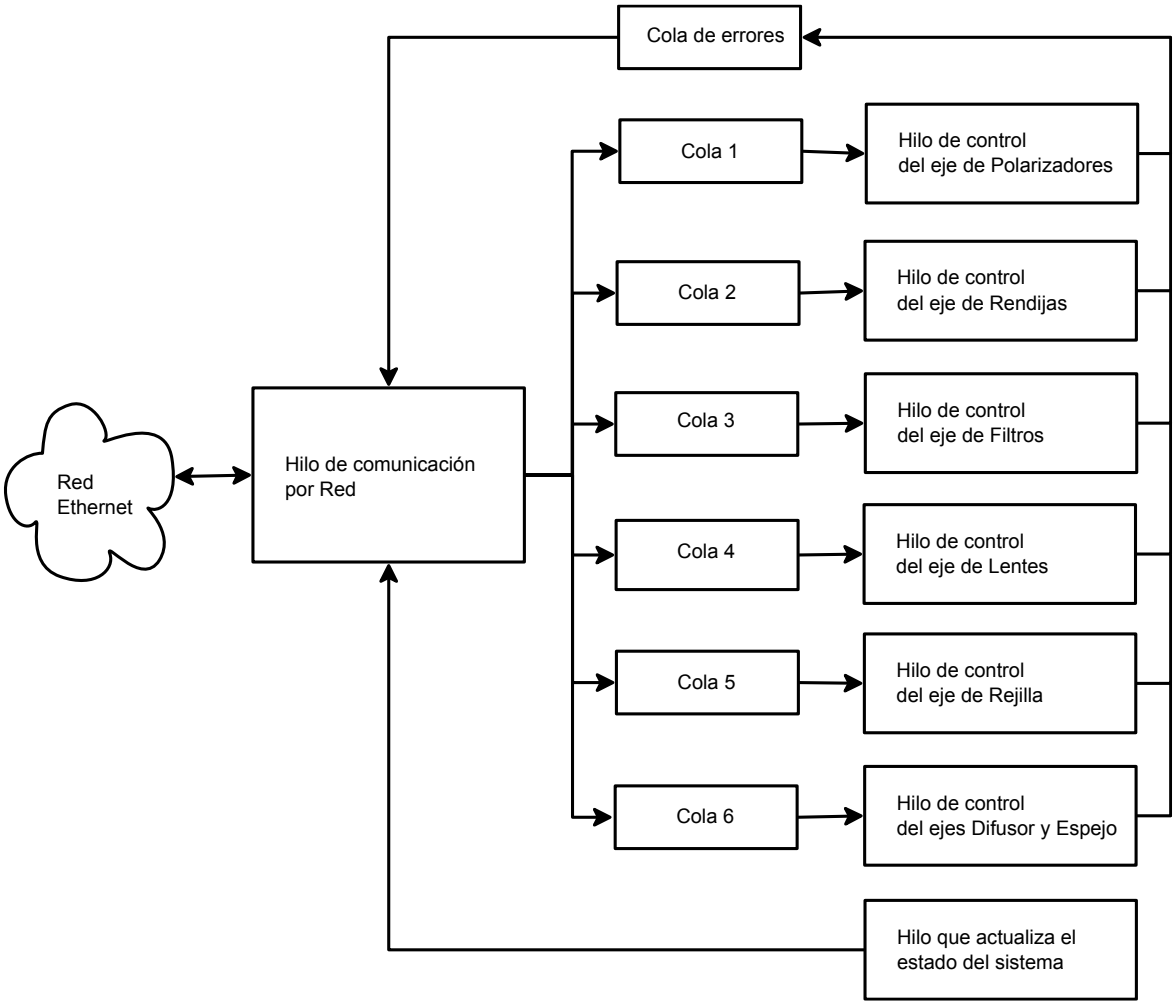


Figura 5: *Estructura del programa de control "mezcal.py".*

TABLA 1

Lista de instrucciones del programa mezcal.py.

:A10;	Envía la cadena de estado del sistema.
:A11;	Alto, detiene el movimiento de la rueda de polarizadores.
:A12;	Inicializa rueda de polarizadores.
:A13000N;	Mueve la rueda de polarizadores a la posición N, donde N es un número del 1 al 5.
:A21;	Alto, detiene el movimiento del eje de Rendijas.
:A22;	Inicializa eje de rendijas.
:A23000N;	Mueve el carro de rendijas a la posición N, donde N es un número del 1 al 3.
:A31;	Alto, detiene el movimiento del eje de Filtros.
:A32;	Inicializa Eje de filtros.
:A33000N;	Mueve el carro de filtros a la posición N, donde N es un número del 1 al 4.
:A41;	Alto, detiene el movimiento del eje de foco,
:A43mnop;	Mueve el eje de foco a la posición “mnop” en hexadecimal.
:A44;	Mueve el eje de foco hacia la derecha hasta que se recibe el mando de Alto.
:A45;	Mueve el eje de foco hacia la izquierda hasta que se recibe el mando de Alto.
:A51;	Alto: Detiene el movimiento del eje de rejilla.
:A52;	Inicializa eje de Rejilla
:A53mnop;	Mueve el eje de Rejilla a la posición “mnop” en hexadecimal.
:A54;	Mueve rejilla hacia la derecha hasta que se recibe el mando de Alto.
:A55;	Mueve rejilla hacia la izquierda hasta que se recibe el mando de Alto.
:A620010;	Configura el control del obturador en Local y abre el obturador.
:A620011;	Configura el control del obturador en Local y cierra el obturador.
:A62000	Configura el control del obturador en Remoto.
:A72ABCD;	Mueve difusor, espejo y controla lámparas de acuerdo a lo siguiente: A: 0 Enciende Lámpara 1, 1 Apaga Lámpara 1. B: 0 Enciende Lámpara 2, 1 Apaga Lámpara 2 C: 0 Mete Espejo, 1 Saca Espejo. D: 0 Mete difusor, 1 Saca difusor.
dame_error	Extrae una cadena de la cola de errores y la envía. Si no hay errores, envía una cadena “OK”.

4.2 PROGRAMA MANEJADOR "Driver" DEL MÓDULO DECODIFICADOR DE CUADRATURA.

La posición del eje de Rejilla se cuantifica con un codificador de tipo incremental con señales de salida en cuadratura. Para obtener la información de posición de este codificador, se utiliza uno de los módulos decodificadores de cuadratura EQEP disponibles dentro de los recursos de *hardware* del Beaglebone. El módulo EQEP es configurable a través de una serie de registros, esto se hace desde funciones desarrolladas y agrupadas en una biblioteca denominada "**mezcal_lib.c**", escrita en lenguaje C. Esta biblioteca constituye el manejador o "**Driver**" de bajo nivel del módulo. La Tabla 2 muestra las funciones disponibles en la bilioteca y su funcionalidad.

Esta bibioteca se compila como un objeto que se llama desde el programa de aplicación "**mezcal.py**".

TABLA 2

Funciones programadas en la biblioteca **mezcal_lib.c**.

configura()	Habilita el módulo EQEP1 escribiendo en los registros QEPCTL y QDECCTL.
lee_codificador()	Regresa el valor del contador del módulo codificador de cuadratura, para esto lee el registro QPOSCNT.
lee_banderas_codificador()	Regresa el valor del registro de banderas del módulo contador de cuadratura, para esto lee el registro QFLG.
limpia_banderas_codificador()	Limpia el registro de banderas del módulo contador de cuadratura, para esto modifica el registro QCLR.
habilita_inicio_codificador()	Habilita la opción de inicializado del contador de posición con flanco de subida en la señal STROBE, para esto modifica el registro QEPCTL.
des_habilita_inicio_codificador()	Deshabilita el inicializado del contador, para esto modifica el registro QEPCTL.

5. CONFIGURACIÓN DE LA MICROCOMPUTADORA BEAGLEBONE

En esta sección se describen los archivos de configuración del Beaglebone para esta aplicación. Se implementó el esquema de disco en memoria RAM que evita que el sistema operativo Linux se corrompa al apagar el microcontrolador.

5.1 CONFIGURACIÓN DE PUERTOS

La Tabla 4 muestra la asignación de pines de la microcomputadora Beaglebone para esta aplicación. La primera columna muestra los nombres asignados a las señales y la segunda columna muestra el nombre del puerto de expansión que corresponde a cada señal. Cada pin es configurable y puede adoptar uno de siete modos de configuración, de acuerdo a lo señalado en el Manual de Referencia para el Beaglebone [2]. La configuración se realiza a través del registro multiplexor de cada pin. En la Tabla 4, la columna "**multiplexor**" muestra la dirección del registro multiplexor correspondiente a cada pin, y la columna "**Dato**" muestra el valor en hexadecimal que se escribió en cada registro multiplexor para su configuración. Este valor se

forma tomando en cuenta la información mostrada en la Tabla 9-1 del Manual del Microprocesador [3].

TABLA 4

Asignación de pines de la microcomputadora Beaglebone.

SEÑAL	PUERTO	MULTIPLEXOR	DATO
EJE1_D_33	P8-7	0X094	0X0f
EJE2_D_33	P8-9	0X09C	0X0f
EJE2_L_33	P8-10	0X098	0x0f
EJE3_D_33	P8-11	0X034	0X0f
EJE3_L_33	P8-12	0X030	0X0f
EJE4_D_33	P8-13	0X024	0X0f
EJE4_L_33	P8-14	0X028	0X0f
EJE5_D_33	P8-15	0X03C	0X0f
EJE5_L_33	P8-16	0X038	0X0f
LAMP2_33	P8-17	0X02C	0X0f
LAMP1_33	P8-18	0X08C	0X0f
ESP_OFF_33	P8-19	0X020	0X0f
ESP_ON_33	P8-20	0X084	0X0f
DIF_OF_33	P8-21	0X080	0X0f
DIF_ON_33	P8-22	0X014	0X0f
E_OBT_33	P8-23	0X010	0X0f
OBT_33	P8-24	0X004	0X0f
/OE	P8-26	0X07C	0X0f
SW5_I_33	P8-38	0X0C4	0X2f
SW5_S_33	P8-40	0X0BC	0X2f
SW4_S_33	P8-41	0X0B0	0X2f
SW4_SEG_33	P8-42	0X0B4	0X2f
SW3_I_33	P8-43	0X0A8	0X2f
SW4_I_33	P8-44	0X0AC	0X2f
SW2_I_33	P8-45	0X0A0	0X2f
SW1_I_33	P8-46	0X0A4	0X2f
LEVA3_33	P9-11	0X070	0X2f
ESP_OUT_33	P9-12	0X078	0x2f
LEVA1_33	P9-13	0X074	0x2f
LEVA2_33	P9-14	0X048	0x2f
ESP_IN_33	P9-16	0X04C	0x2f
DIF_OUT_33	P9-27	0X1A4	0x2f
SW_TTL_1_33	P9-28	0X19C	0x2f
SW_ESP_S_33	P9-30	0X198	0x2f
DIF_IN_33	P9-31	0X190	0x2f
SW_ORIGEN	P8-31	0X0D8	0x2a
SW5_I_33	P8-38	0X0C4	0x2a
A_33	P8-35	0X0D0	0x2a
B_33	P8-33	0X0D4	0x2a
SW_ORIGEN	P8-32	0X0DC	0x2a

El procedimiento para configurar los multiplexores es mediante un archivo "script" donde se colocan los datos de las columnas 3 y 4 de la Tabla 4. El archivo tiene terminación ".dts". En el Apéndice C se muestra el listado del archivo generado para la tarjeta de nuestra aplicación denominado "mezcal-ooAo.dts". Este archivo se compiló e instaló en el núcleo del sistema operativo para configurar los multiplexores. Se compiló con un programa llamado "dtc" que se incluye en la distribución de Linux para el Beaglebone. La compilación se realizó tecleando la siguiente instrucción desde una ventana de comandos:

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

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

```
echo mezcal > /sys/devices/bone_capemgr.9/slots
```

Cabe mencionar que cada vez 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.

5.2 CONFIGURACIÓN DE DIRECCIÓN IP ESTÁTICA

Los datos de red configurados en el Beaglebone se muestran en la Tabla 3.

TABLA 3

Datos de red configurados en el Beaglebone.

IP	192.168.0.26
Gateway	192.168.0.254
Mask	255.255.255.0

5.3 CONFIGURACIÓN DEL OPERATIVO PARA SU EJECUCIÓN EN UN DISCO EN MEMORIA RAM

En una microcomputadora Beaglebone, el Sistema operativo Linux normalmente es ejecutado desde una memoria flash tipo tarjeta microSD, lo que permite usarla como si tuviera instalado un disco duro. Sin embargo, las memorias flash tienen varias desventajas que afectan el desempeño de un sistema de control dedicado. Dos inconvenientes sobresalen: el primer problema es el número finito de ciclos de escritura de la memoria flash; esta condición a la larga causa que el sistema no inicie por daño en el disco duro. El segundo problema, que no es inherente a las memorias flash pero que es muy grave para un sistema de control dedicado, es la corrupción del sistema de archivos que se da con las posibles fallas de la energía eléctrica [4].

Dado lo anterior, se decidió configurar la microcomputadora para que la ejecución del operativo y de los programas se hiciera desde un disco en memoria RAM. Para ello se siguió al pie de la letra el procedimiento descrito en [4]. A continuación se describen las adecuaciones que se hicieron al procedimiento para nuestra aplicación.

5.3.1 PARTICIONES EN LA MEMORIA MICROSD

Se eligió una memoria microSD de 4GB de capacidad en la que se instaló la imagen de Linux Debian. En el espacio libre de 2GB que quedó disponible, se crearon dos particiones: "trabajo" y "log". La partición "trabajo" contiene los programas y archivos de configuración de nuestra aplicación; en el esquema de disco en RAM se monta como sólo lectura. La partición "log" se monta como escritura para que el programa de aplicación pueda escribir ahí el archivo "mezcal.log" con los mensajes de salida.

5.3.2 ARCHIVOS EN LA PARTICIÓN "TRABAJO"

En la partición "trabajo" se encuentra el archivo "aplica.sh" y tres directorios: *app*, *compila* y *mezcal*. En el directorio *compila* se encuentra la imagen "initrd2.img" que fue modificada para que el sistema operativo se ejecute solamente en memoria RAM. En el directorio *app* se encuentran los scripts: "ap_dropbear.sh", "ap_mezcal.sh" y "pon_hora.sh", desde los que se ejecutan los programas necesarios para la aplicación. Por último, en el directorio *mezcal* se encuentran los programas de configuración de puertos, control del instrumento y biblioteca de funciones.

En la imagen modificada se monta la partición "trabajo" como sólo lectura y se ejecuta el script "aplica.sh". En éste se configura la dirección de IP y se llama a la ejecución de los scripts: "ap_dropbear.sh", "pon_hora.sh" y "ap_mezcal.sh". El primero arranca la aplicación Dropbear, que es un servidor de ssh que permite accesar al microcontrolador desde otra computadora. El segundo obtiene la hora del servidor de tiempo y actualiza el reloj del microcontrolador. El último ejecuta el programa de control del instrumento. El contenido de estos scripts se encuentra en el Apéndice D.

5.3.3 ACCESO AL MICROCONTROLADOR DESDE OTRA PC

El acceso al microcontrolador se hace via ssh de la siguiente forma:

```
ssh root@192.168.0.26
passwd: root
```

Dentro del sistema en RAM-DISK no se tienen muchas de las aplicaciones que existen en la ejecución normal. A continuación se mencionan los pasos a seguir para realizar algunas tareas:

Para montar el sistema de archivos y tener acceso a los archivos de Mezcal, es necesario montar la partición donde está el sistema de archivos original de Linux Debian. Generalmente es la partición 2, y en la configuración de ram-disk se etiquetó como "oldroot". Se puede montar como lectura-escritura para hacer cambios:

```
mount -o remount,rw /oldroot
```

Los archivos de Mezcal están en el directorio:

/oldroot/home/machinekit/mezcal/

Si se quieren tener las aplicaciones del sistema de archivos original, es necesario hacer un "chroot":

```
chroot /oldroot/ /bin/bash -login
```

Para reinicializar el microcontrolador, es necesario escribir en el registro del "watchdog" y después de un minuto se reinicializará:

```
"cat > /dev/watchdog"
```

6. PRUEBAS Y RESULTADOS

El nuevo control ha sido utilizado de manera regular en las temporadas de observación, siendo la primera en octubre de 2014. En esta temporada se tuvo como principal problema la suspensión inesperada en la ejecución del hilo que actualiza el estado. Para corregirlo se hicieron modificaciones al programa para robustecerlo.

La segunda temporada fue en febrero de 2015, donde el nuevo control trabajó de manera aceptable; se logró identificar un problema mecánico en el eje 3 revisando el archivo ".log" generado por el programa de control.

En febrero de 2016 se utilizó por 13 noches consecutivas sin presentar ningún problema.

7. REFERENCIAS

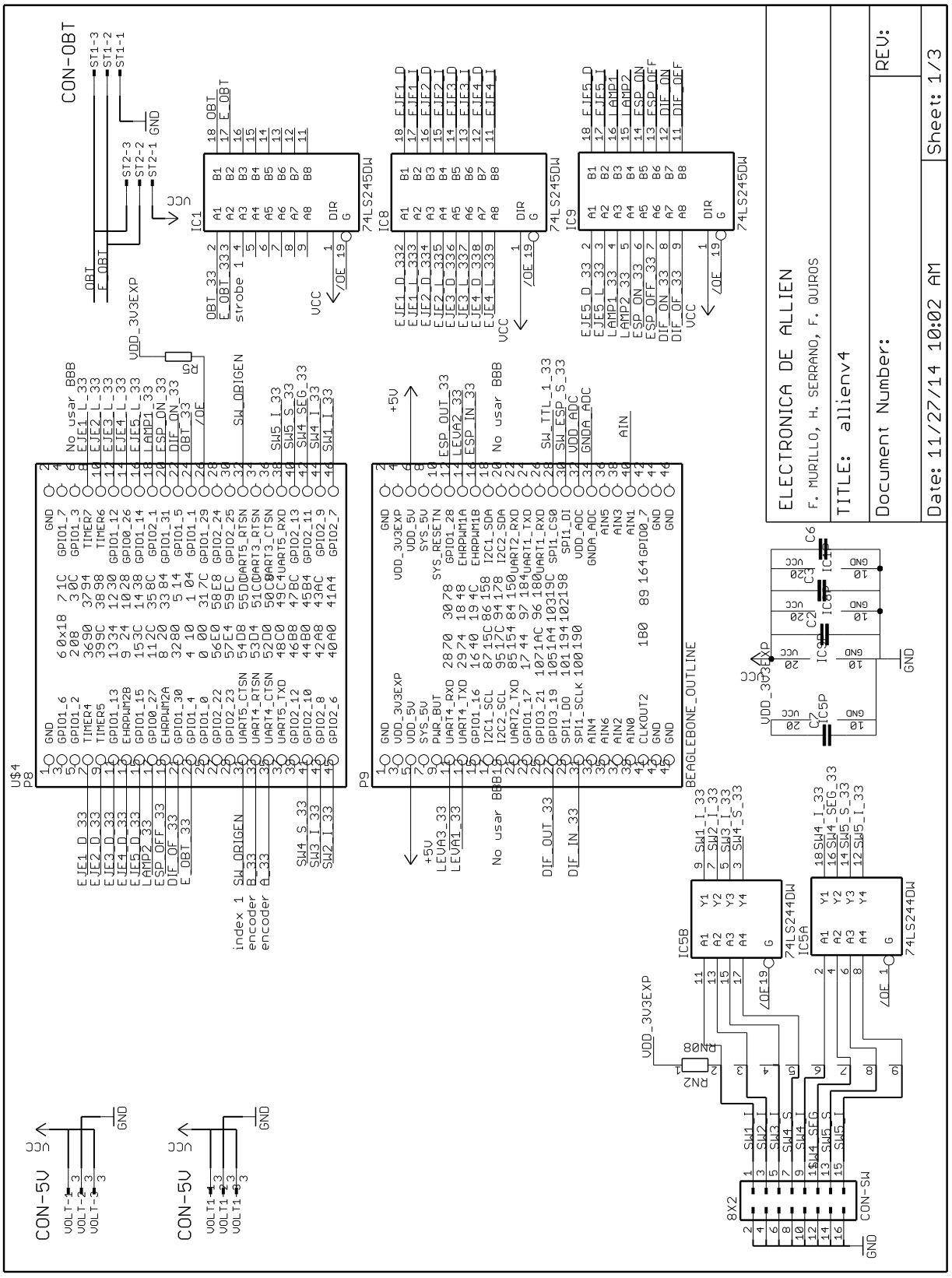
[1] MEZCAL: Un espectrógrafo computarizado para su uso en el Observatorio Astronómico Nacional.
L. Gutiérrez, J.M. Murillo, F. Quirós, M. Pedrayes, J. López, J. Meaburn.
Disponible en:
http://www.astrossp.unam.mx/~sectec/web/instrumentos/mezcal/jal/Gutierrez01.pdf

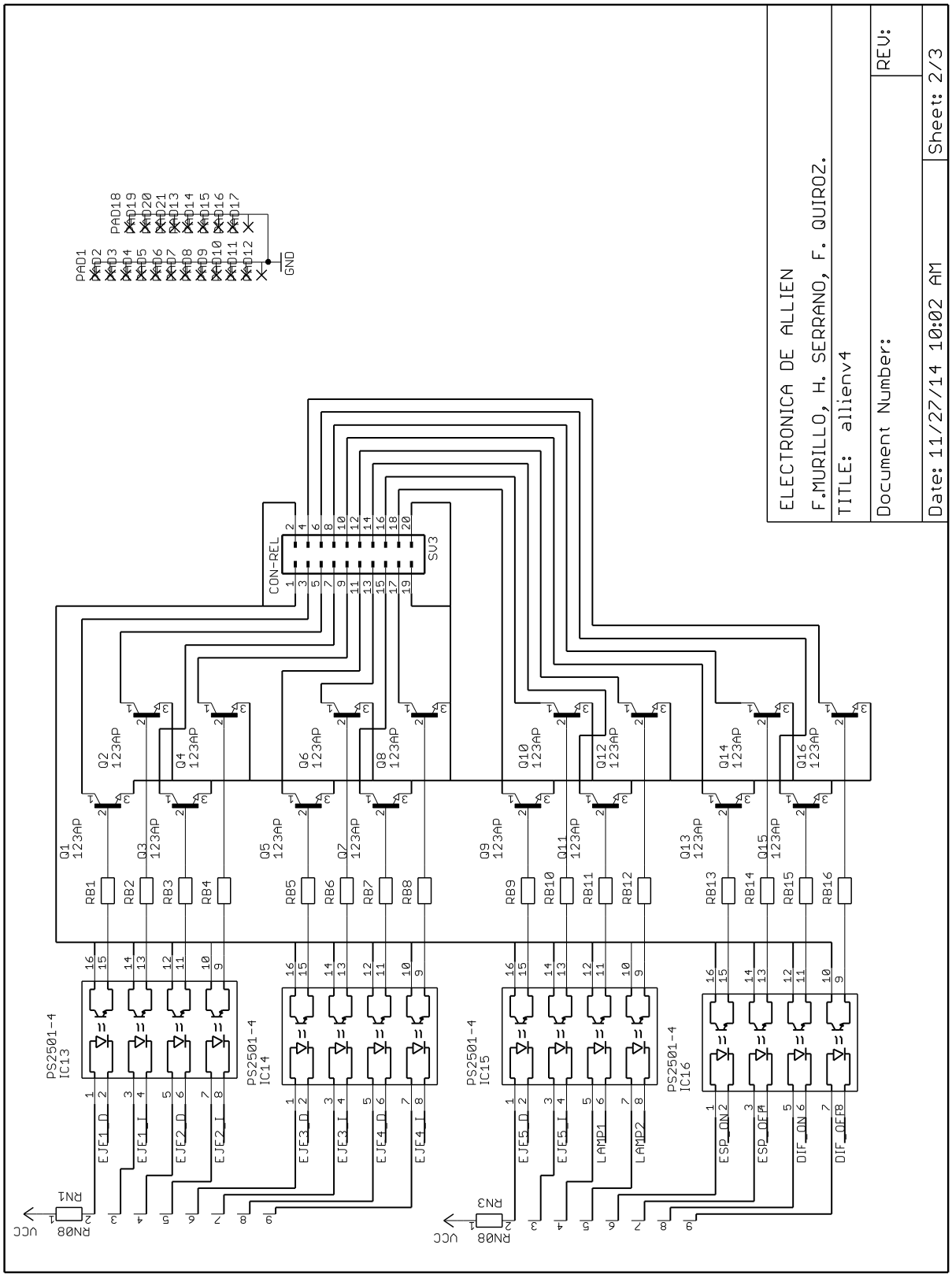
[2] BeagleBone Black System Reference Manual.
Revision A5.6.

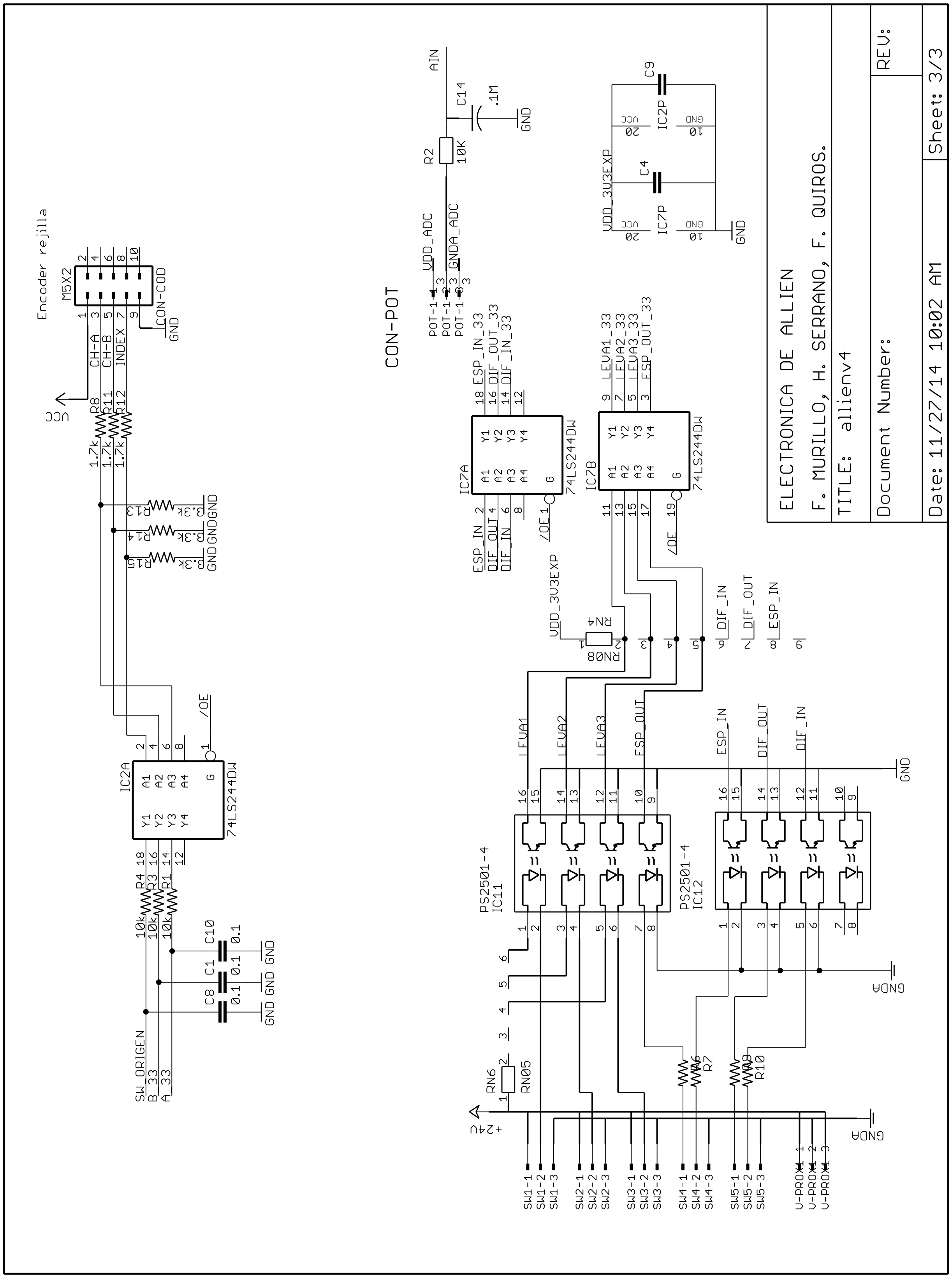
[3] AM335x ARM Cortex-A8 Microprocessors (MPUs).
Technical Reference Manual.

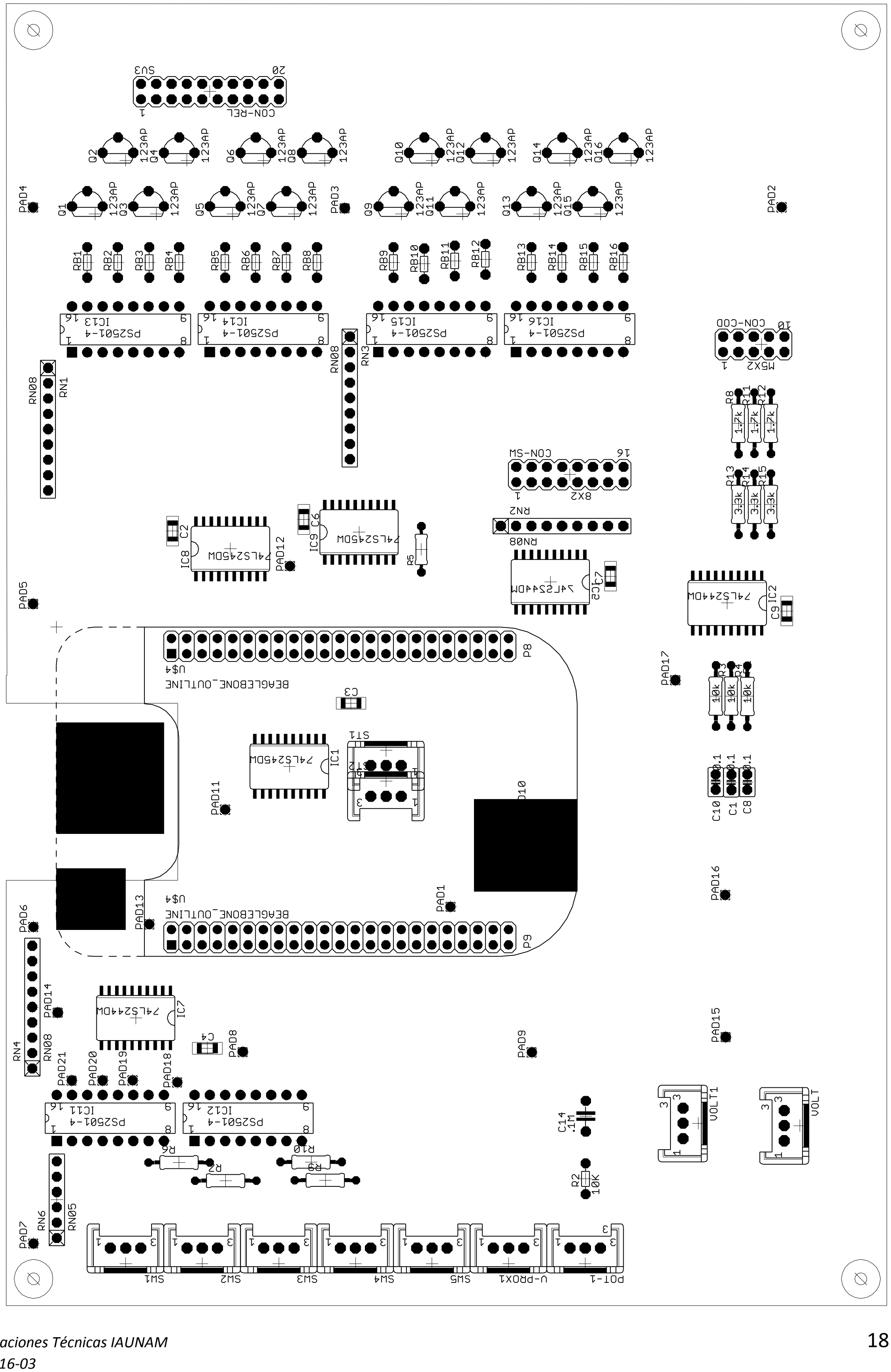
[4] Zazueta, S.
"Procedimiento de configuración de la microcomputadora BeagleBone para que ejecute una aplicación dedicada desde memoria RAM".
Publicaciones Técnicas del Instituto de Astronomía, UNAM.
Comunicación Interna. CI-2014-05.
México, Abril, 2014.

APÉNDICE A. DIAGRAMA ESQUEMÁTICO Y MAPA DE COMPONENTES DE LA TARJETA DE CONTROL




ELECTRONICA DE ALLIEN
F.MURILLO, H. SERRANO, F. QUIROZ.
TITLE: allienv4
Document Number:
REV:
Date: 11/27/14 10:02 AM
Sheet: 2/3
CON-REL
SV3
PAD1
PAD2
PAD3
PAD4
PAD5
PAD6
PAD7
PAD8
PAD9
PAD10
PAD11
PAD12
PAD13
PAD14
PAD15
PAD16
PAD17
PAD18
PAD19
PAD20
PAD21
GND
Q1 123AP
Q2 123AP
Q3 123AP
Q4 123AP
Q5 123AP
Q6 123AP
Q7 123AP
Q8 123AP
Q9 123AP
Q10 123AP
Q11 123AP
Q12 123AP
Q13 123AP
Q14 123AP
Q15 123AP
Q16 123AP
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RB8
RB9
RB10
RB11
RB12
RB13
RB14
RB15
RB16
PS2501-4 IC13
PS2501-4 IC14
PS2501-4 IC15
PS2501-4 IC16
RN1
RN2
RN08
VCC
EJE1_D
EJE1_I
EJE2_D
EJE2_I
EJE3_D
EJE3_I
EJE4_D
EJE4_I
EJE5_D
EJE5_I
LAMP1
LAMP2
ESP_ON2
ESP_OFF
DIF_ON
DIF_OFF


ELECTRONICA DE ALLIEN
F. MURILLO, H. SERRANO, F. QUIROS.
TITLE: allienv4
Document Number:
REV:
Date: 11/27/14 10:02 AM
Sheet: 3/3


CON-REL
SU3
CON-COD
M5X2
CON-SM
8X2
PS2501-4
74LS245DW
74LS244DW
BEAGLEBONE_OUTLINE
P8
P9
VOLT1
VOLT
SW1
SW2
SW3
SW4
SW5
U-PROX1
POT-1

APÉNDICE B. MENSAJES DE ERROR DEL SISTEMA

Eje_x Error: No encontré SW de proximidad leva_x

Este error se puede presentar en los ejes 1,2 y 3. Indica que pasó un tiempo de 10 segundos y no se detectó la leva que activa el sensor de proximidad del mecanismo de ginebra.

Eje_x Error: No encontré el SW x.

Este error se puede presentar en los ejes 1,2 y 3. Indica que no se encontró el interruptor de inicio del mecanismo de ginebra.

Eje2: Error de inicializado, no llegó el SW de inicio

Eje2: Error de inicializado, no llegó la Leva

Eje3: Error de inicializado, no llegó el SW de inicio

Eje3: Error de inicializado, no llegó la Leva

Al inicializar los ejes 2 y 3, primero se detecta el SW de inicio, posteriormente se detecta la leva. Si alguno de los dos no se detecta, se genera alguno de los errores anteriores.

Intentaré posicionar de nuevo la rendija

Intentaré posicionar de nuevo el filtro

Si ocurre un error de posicionado en los carros de rendija y filtros, el programa inicializará el eje e intentará de nuevo.

Eje4 Error: No llegué a la posición

Este error se genera cuando se manda mover el eje de foco, indica que pasó un tiempo mayor a 60 segundos, y no alcanzó la posición deseada.

Eje5 Error: No encontré la señal INDEX.

Este error se genera cuando el eje de rejilla se mueve en busca de la señal Index del codificador y pasa un tiempo mayor a 60 segundos y no la encuentra.

Eje5 Error: No llegué a la posición.

Este error indica que transcurrio un tiempo mayor a 60 segundos y no se alcanzó la posición deseada.

Error: Hubo un error al sacar el espejo

Este error se genera cuando pasa un tiempo mayor a 5 segundos y no se detecta la señal del interruptor que indica que el espejo está fuera.

Error: Hubo un error al meter el espejo

Este error se genera cuando pasa un tiempo mayor a 5 segundos y no se detecta la señal del interruptor que indica que el espejo está dentro.

Error: Hubo un error al meter el difusor

Este error se genera cuando pasa un tiempo mayor a 120 segundos y no se detecta la señal del interruptor que indica que el difusor está dentro.

Error: Hubo un error al sacar el difusor

Este error se genera cuando pasa un tiempo mayor a 120 segundos y no se detecta la señal del interruptor que indica que el difusor está fuera.

APÉNDICE C. ARCHIVO DE CONFIGURACIÓN DE PINES 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 = "MEZCAL";
    version = "00A0";

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

    fragment@0 {
        target = <&am33xx_pinmux>;
        __overlay__ {
            MEZCAL: pinmux_MEZCAL_pins {
                pinctrl-single,pins = <
                    0x090 0x0f    /* P8-7 */
                    0x094 0x0f    /* P8-8 */
                    0x09C 0x0f    /* P8-9 */
                    0x098 0x0f    /* P8-10 */
                    0x034 0x0f    /* P8-11 */
                    0x030 0x0f    /* P8-12 */
                    0x024 0x0f    /* P8-13 */
                    0x028 0x0f    /* P8-14 */
                    0x03C 0x0f    /* P8-15 */
                    0x038 0x0f    /* P8-16 */
                    0x02C 0x0f    /* P8-17 */
                    0x08C 0x0f    /* P8-18 */
                    0x020 0x0f    /* P8-19 */
                    0x084 0x0f    /* P8-20 */
```

```
                                0x080 0x0f      /* P8-21 */
                                0x014 0x0f      /* P8-22 */
                                0x010 0x0f      /* P8-23 */
                                0x004 0x0f      /* P8-23 */
                                0x07C 0x0f      /* P8-26 */
                                0x0C4 0x2f      /* P8-38 */
                                0x0BC 0x2f      /* P8-40 */
                                0x0B0 0x2f      /* P8-41 */
                                0x0B4 0x2f      /* P8-42 */
                                0x0A8 0x2f      /* P8-43 */
                                0x0AC 0x2f      /* P8-44 */
                                0x0A0 0x2f      /* P8-45 */
                                0x0A4 0x2f      /* P8-46 */
                                0x070 0x2f      /* P9-11 */
                                0x078 0x2f      /* P9-12 */
                                0x074 0x2f      /* P9-13 */
                                0x048 0x2f      /* P9-14 */
                                0x04C 0x2f      /* P9-16 */
                                0x1A4 0x2f      /* P9-27 */
                                0x19C 0x2f      /* P9-28 */
                                0x194 0x2f      /* P9-29 */
                                0x198 0x2f      /* P9-30 */
                                0x190 0x2f      /* P9-31 */
                                0x0d8 0x2a      /* P8-31 */
                                0x0d4 0x2a      /* P8-38 */
                                0x0d0 0x2a      /* P8-35 */
                                0x0dc 0x2a>;  /* P8-32 */
                        };
                };
        };

        fragment@1 {
                target = <&ocp>;
                __overlay__ {
                        test_MEZCAL {
                                compatible      = "bone-pinmux-helper";
                            pinctrl-names       = "default";
                            pinctrl-0 = <&MEZCAL>;
                                status          = "okay";
                        };
                };
        };
};
```

APÉNDICE D. ARCHIVOS DE CONFIGURACIÓN Y ARRANQUE DE LOS PROGRAMAS

D1. APLICA.SH.

```
#!/bin/sh
ifconfig etho 192.168.0.26
busybox nc -l -l -p 9090 -e /bin/sh &
sh /app/app/ap_dropbear.sh
sh /app/app/pon_hora.sh
sleep 5
sh /app/app/ap_mezcal.sh > /dev/null &
## fin del listado de aplica.sh
```

D2. AP_DROPBEAR.SH.

```
#!/bin/sh
mkdir /oldroot
mount -t ext4 -o ro /dev/mmcblkop2 /oldroot
nn=$( cat /oldroot/etc/shadow | grep root|cut -f2 -d':' )
cp /oldroot/etc/shadow /etc
cp /oldroot/etc/passwd /tmp
sed -e 's/bash/sh/g' /tmp/passwd > /tmp/passwd1
mv /tmp/passwd1 /etc/passwd
echo "/oldroot/lib" >> /etc/ld.so.conf
echo "/oldroot/lib/arm-linux-gnueabihf/" >> /etc/ld.so.conf
cp -dpa /oldroot/etc/dropbear /etc
/oldroot/sbin/ldconfig
/oldroot/usr/sbin/dropbear -E &
```

D3. PON_HORA.SH.

```
#!/bin/sh
echo "/oldroot/usr/lib/arm-linux-gnueabihf/" >> /etc/ld.so.conf
echo "/oldroot/lib/arm-linux-gnueabihf" >> /etc/ld.so.conf
echo "/oldroot/usr/lib" >>  /etc/ld.so.conf
echo "/oldroot/lib" >>  /etc/ld.so.conf
cp /oldroot/etc/services /etc
/oldroot/sbin/ldconfig
route add default gw 192.168.0.253
/oldroot/usr/sbin/ntpdate -4 -v 192.168.3.1
```

D4. AP_MEZCAL.SH.

```
#!/bin/sh
mkdir -p  /media/log
mount -t ext4 -o rw /dev/mmcblkop4 /media/log
export PATH=$PATH:.:/oldroot/usr/bin/:/oldroot/bin/
cd /app/mezcal/
sh ./configura_mux.sh
python ./mezcal.py &
```

D5. CONFIGURA_MUX.SH.

```
#!/bin/sh
ln -s /oldroot/lib/firmware /lib/firmware
echo mezcal > /sys/devices/bone_capemgr.9/slots
```

APÉNDICE F. CADENA DEL ESTADO DEL SISTEMA

La cadena tiene el siguiente formato:

;[A][o][a3][a2][a1][ao][-][b2][b1][bo][-][c3][c2][c1][-][d4][d3][d2][d1][do][;]

Donde:

;[a3][a2][a1][ao]	Posicion de Eje 5
;[b2][b1][bo]	Posicion de Eje 4
;[c3]	Posicion de Gin 3 [1,4]
;[c2]	Posicion de Gin 2 [1,3]
;[c1]	Posicion de Gin 1 [1,5]

; [d4]
; [bit3] Leva1
; [bit2] Ind1
; [bit1] Leva2
; [bito] Ind2
; [d3]
; [bit3] Leva3
; [bit2] Ind3
; [bit1] Sup4
; [bito] Inf4
; [d3]
; [bit3] Sup5
; [bit2] Inf5
; [bit1] Seguro
; [bito] Obt
; [d1]
; [bit3] E_Obt
; [bit2] Esp_Dentro
; [bit1] Esp_Fuera
; [bito] Dif_Dentro
; [do]
; [bit3] Dif_Fuera
; [bit2] Lamp1
; [bit1] Lamp2
; [bito] Spare