

Actualización del control de cúpula del telescopio de 84cm.

F. Murillo, F. Quirós, S. Zazueta, J.M. Murillo, G. Guisa, A. Córdova, B. García.

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

RESUMEN:

Este trabajo documenta la actualización realizada a los controles de cúpula y cortina del telescopio de 84cm que consistió en la unificación de ambos controles y la implementación de una micro-computadora de

tecnología reciente para realizar las tareas. También se sustituyó el manejador del motor de cúpula por una versión con rampa de velocidad para un arranque suave.

Contenido

1. INTRODUCCIÓN-----	3
2. DESCRIPCIÓN DEL SISTEMA-----	3
2.1 EL MOTOR DE LA CÚPULA -----	3
2.2 EL CODIFICADOR DE POSICIÓN-----	4
2.3 LA CORTINA Y EL GAJO -----	5
2.4 ESPECIFICACIONES DEL SISTEMA -----	5
3. CONTROL DE CÚPULA Y CORTINA-----	6
3.1 USO DE LA BOTONERA MANUAL -----	9
4. LA MICROCOMPUTADORA BEAGLEBONE-----	10
4.1 ASIGNACIÓN DE PATAS -----	10
4.2 PROGRAMAS DESARROLLADOS-----	12
4.2.1 LA BIBLIOTECA “CUPULA_84_LIB.C”-----	12
4.2.2 EL PROGRAMA “CUPULA_84.PY”-----	14
4.3 CONFIGURACIÓN DE LA MICROCOMPUTADORA BEAGLEBONE -----	15
4.3.1 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA OPERATIVO-----	16
4.3.2 CONFIGURACIÓN DE UNA DIRECCIÓN DE IP FIJA -----	16
4.3.3 CONFIGURACIÓN PARA EJECUTAR EL PROGRAMA DE APLICACIÓN AL ARRANCAR EL OPERATIVO	17
5. REFERENCIAS-----	18
APÉNDICE A. DIAGRAMA ESQUEMÁTICO Y PCB DE LA TARJETA DEL CONTROLADOR-----	19

APÉNDICE B: DIAGRAMA ESQUEMÁTICO Y MAPA DE COMPONENTES DE LA TARJETA DE RELEVADORES Y CODIFICADOR DE CÚPULA -----	23
APÉNDICE C: DIAGRAMA ESQUEMÁTICO DEL CABLEADO DE RELEVADORES Y MANEJADOR MARCA LEESON MODELO M1740006 DEL MOTOR DE CÚPULA-----	25
APÉNDICE D. DIAGRAMA ESQUEMÁTICO Y MAPA DE COMPONENTES DE LA TARJETA DE RELEVADORES Y SENSORES DE CORTINA Y GAJO-----	26
APÉNDICE E. CONFIGURACIÓN DE LOS REGISTROS DEL BEAGLEBONE-----	28

1. INTRODUCCIÓN

El telescopio de 84cm del OAN es uno de los tres telescopios instalados en la Sierra de San Pedro Mártir, en el municipio de Ensenada. Posee una cúpula móvil que lo protege del clima, la cual tiene una cortina y un gajo que se abren al inicio de las observaciones y se cierran al finalizar. Al abrirse, la cortina deja una ventana por donde el telescopio capta la luz de los objetos celestes; la ventana tiene que posicionarse en la dirección donde se desea observar con el telescopio y para ello cuenta con un control automático de posicionamiento de cúpula, desarrollado en el IA-UNAM Ensenada.

La última actualización al control de posicionamiento de la cúpula del telescopio de 84cm, previa a la que se describe en el presente reporte, se realizó en el año 2009 [1]. En esa versión se utilizó un microcontrolador Rabbit con Red Ethernet para realizar el control. En ese mismo año se instaló un control automático de apertura de cortina y gajo [2] que utilizaba también un microcontrolador Rabbit para realizar la tarea. Para llevar energía a los motores de la cortina y el gajo, que están sobre la cúpula, se instaló un segmento de contactos deslizantes, de tal manera que la cúpula debe posicionarse en el segmento para realizar la apertura de la cortina y el gajo. En este esquema, el posicionamiento se realizaba utilizando el sistema de control de la cúpula, y la apertura se hacía utilizando el sistema de control de la cortina y el gajo. Ambos se coordinaban mediante el programa de control robótico instalado en la computadora de control de instrumentos de la sala de observación, denominada 'Grulla' en la jerga del Observatorio. Este esquema funcionaba de una forma no robusta, con frecuencia la cortina no cerraba o abría por errores atribuidos principalmente al posicionamiento de la cúpula.

Para solucionar el problema se propuso unificar ambos controles de tal manera que, para abrir la cortina, sea el control de la cúpula el que lleve la cúpula al segmento utilizando un interruptor de referencia y, posteriormente, realice la apertura. Además, se utilizaría una microcomputadora de tecnología reciente y un manejador de motor con rampa para un arranque suave. La propuesta fue aprobada y se inició la construcción del nuevo control, que se puso en operación en agosto de 2013.

Las secciones siguientes documentan el nuevo control de la cúpula y la cortina desarrollado para el telescopio de 84cm del OAN SPM.

2. DESCRIPCIÓN DEL SISTEMA

2.1 EL MOTOR DE LA CÚPULA

La cúpula del telescopio de 84cm gira en ambas direcciones impulsada por un motor de corriente continua que tiene las siguientes características:

TABLA 1

Características del motor de la cúpula.

Marca:	Toledo Gearmotor Company.
Modelo:	TC43
Potencia:	¼ HP
Voltaje:	90V
Corriente:	3 A

La cúpula cuenta con una cremallera en su periferia a la cual se acopla el motor a través de un engrane.

2.2 EL CODIFICADOR DE POSICIÓN

El codificador de posición de la cúpula es de tipo incremental rotatorio con las siguientes características:

TABLA 2

Características del codificador de cuadratura.

Número de serie	TRD-NH500-RZWD
Pulsos por revolución	500
Voltaje de alimentación	5-30VDC
Nivel de las Salidas	TTL

El codificador se encuentra acoplado a la cremallera de la cúpula a través de un engrane, de tal manera que cuando gira la cúpula, el codificador se mueve. El acoplamiento se realiza utilizando una caja de reducción para que el eje del codificador gire una revolución cuando la cúpula da una vuelta completa (Ver *Figura 1*).

La caja de engranes no es indispensable en el arreglo, se construyó en el pasado para un codificador absoluto. La reducción se hizo a la medida, considerando el número de perforaciones de la cremallera a lo largo de la circunferencia de la cúpula. Después de un mantenimiento realizado a la cúpula, el número de perforaciones de la cremallera se redujo en 4 y la reducción dejó de ser exacta. La posición de la cúpula reportada por el codificador absoluto se desfasaba con el tiempo y era necesario reajustar periódicamente la posición del cero. La solución más rápida fue cambiar el codificador por uno de tipo incremental.

El codificador genera dos señales en cuadratura para indicar movimiento y sentido. Estas señales se leen utilizando un módulo decodificador que incrementa la resolución en un factor de 4, de tal manera que el codificador genera 2000 pulsos por revolución. La resolución en la

posición de la cúpula es de 0.18 grados, que resulta basta pero suficiente para la aplicación. Dada la dimensión transversal de la ventana de la cúpula (2.31 m) comparada con la apertura del telescopio (0.84m), se pueden tolerar errores de posicionamiento de 1 grado sin obstruir al telescopio.

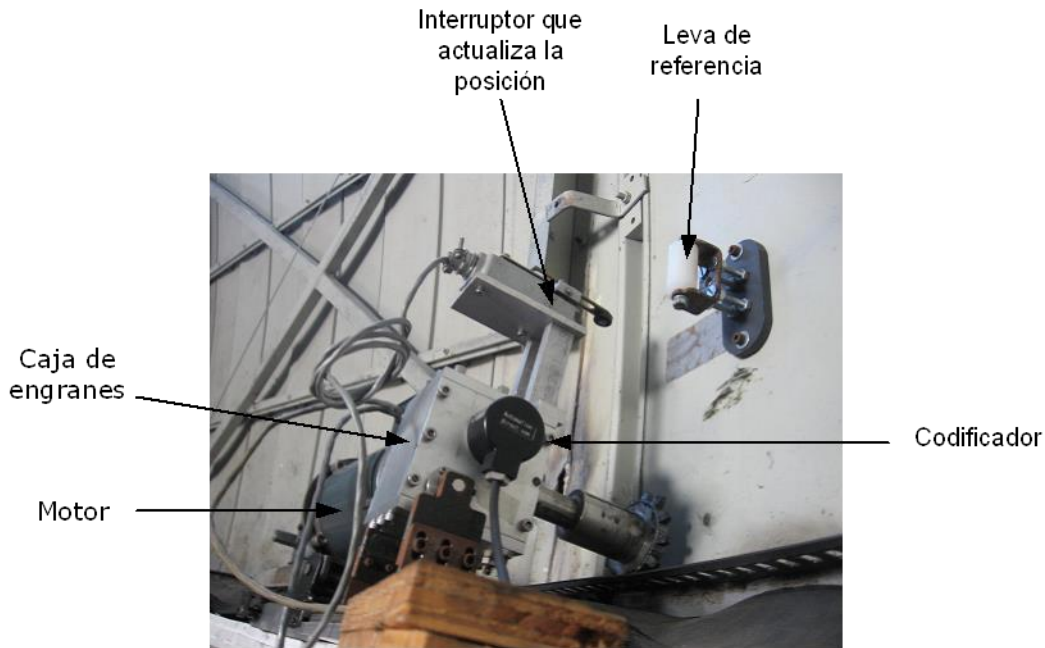


Figura 1: Codificador, motor e interruptor de inicio para el posicionamiento de la cúpula.

2.3 LA CORTINA Y EL GAJO

La **cortina** sube y baja impulsada por un motor de corriente alterna que puede girar en ambas direcciones, dependiendo de la alimentación de su bobina de arranque. Posee interruptores límite para detener el movimiento al llegar a las posiciones extremas.

El **gajo** abre y cierra como una puerta mediante la acción de un pistón impulsado por un motor de corriente alterna. Al igual que la cortina, posee interruptores límite que detienen el movimiento cuando se alcanzan las posiciones extremas.

El cableado de los motores e interruptores límite del sistema de gajo y cortina llega hasta una caja de control eléctrico que está sobre la cúpula [2].

2.4 ESPECIFICACIONES DEL SISTEMA

Las especificaciones del sistema completo se muestran en la Tabla 3. Los movimientos de la cortina y el gajo se realizan de forma secuencial; en esa tabla se muestran los tiempos individuales y combinados. Los últimos dos renglones muestran los tiempos máximos de apertura y cierre tomando en cuenta que la cúpula gira media vuelta para llegar al contacto deslizante.

TABLA 3
Especificaciones del sistema.

Resolución.	0.18° por pulso
Velocidad final	15.9 pulsos por segundo
Tiempo en alcanzar la velocidad final	5 segundos
Tiempo de frenado	0.5 segundos
Tiempo de apertura de cortina	139 segundos
Tiempo de apertura de gajo	15 segundos
Tiempo de apertura de cortina y gajo	154 segundos
Tiempo de cierre de cortina	141 segundos
Tiempo de cierre de gajo	15 segundos
Tiempo de cierre de cortina y gajo	156 segundos
Tiempo de giro de una vuelta completa	126 segundos
Tiempo máximo de apertura	217 segundos
Tiempo máximo de cierre	219 segundos

3. CONTROL DE CÚPULA Y CORTINA

La *Figura 2* muestra la forma en que está organizada la electrónica de control. Básicamente está formada por tres tarjetas:

- Tarjeta del Controlador
- Tarjeta de Relevadores y Codificador de Cúpula
- Tarjeta de Relevadores y Sensores de Cortina y Gajo

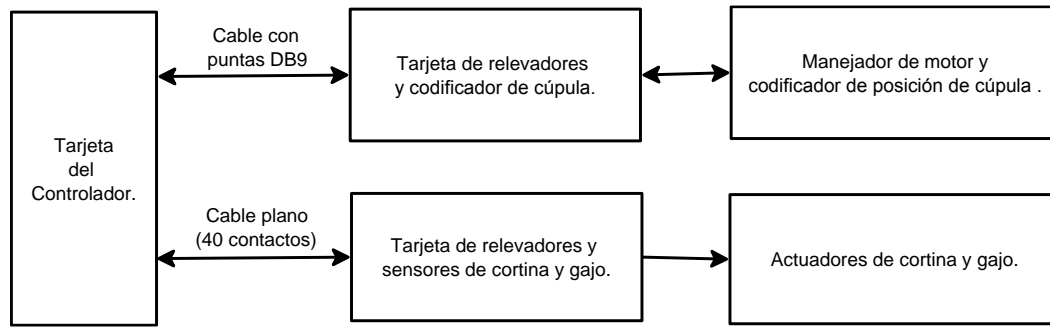


Figura 2: Organización del control de cúpula, cortina y gajo.

La Tarjeta del Controlador contiene una microcomputadora modelo “Beaglebone” de la compañía Texas Instruments; se diseñó para interconectar sus puertos de entrada-salida con las dos tarjetas de relevadores. Su diagrama esquemático y mapa de componentes se muestran en el Apéndice A. Básicamente contiene reforzadores, optoacopladores y un manejador de línea para recibir las señales del codificador de la cúpula.

Esta tarjeta se construyó en el laboratorio de electrónica del IA-UNAM Ensenada. La *Figura 3* muestra su instalación dentro del gabinete de Control de Cortina ubicado en el lado noroeste del piso del telescopio de 84cm.

Tarjeta del Controlador



Tarjeta de relevadores y sensores de cortina y gajo

Figura 3: Vista de la tarjeta del Controlador instalada en el interior del gabinete de control de la cortina y gajo.

Para el manejo de la corriente del motor de la cúpula se adquirió un manejador de motores comercial modelo LESSON que tiene la capacidad de generar una rampa de aceleración para el arranque, y de desaceleración para detener el movimiento. Este manejador está construido para ser operado manualmente, de fábrica viene con un par de interruptores: uno para encendido y otro para dirección. Para acoplarlo a la Tarjeta del Controlador se sustituyeron los interruptores manuales por relevadores activados eléctricamente. Se diseñó la Tarjeta de Relevadores y Codificador de Cúpula, la cual cuenta con transistores para activar los relevadores del manejador del motor, un circuito de retardo para activar el relevador de dirección y, un instante después, el de encendido, posee un manejador de línea para enviar las señales del codificador de posición de la cúpula a la Tarjeta del Controlador.

Su diagrama esquemático y mapa de componentes se muestran en el Apéndice B. Esta tarjeta también se construyó en el laboratorio de electrónica del IA-UNAM, Ensenada. La *Figura 4* muestra la vista interior del gabinete donde se instalaron y alambraron la Tarjeta de Relevadores y Codificador de Cúpula, el manejador del motor de la cúpula y los relevadores. La *Figura 5* ilustra los conectores instalados en el exterior del gabinete.

La Tarjeta de Relevadores y Sensores de la Cortina y Gajo se reutilizó del control original de la cortina, y su diagrama y mapa de componentes pueden verse en el Apéndice D. Básicamente contiene relevadores para mover los contactores que energizan los motores de la cortina y gajo y, además, cuenta con sensores de corriente para verificar si los motores se están moviendo. La comunicación entre la Tarjeta del Controlador y esta tarjeta se realiza mediante un cable plano de 40 contactos (Ver *Figura 3*).

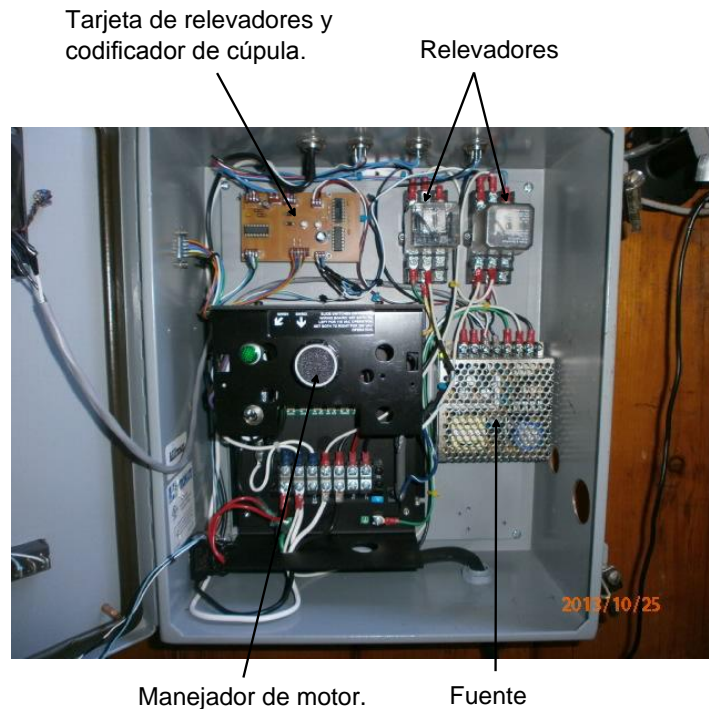


Figura 4: Vista interior del gabinete del manejador del motor de la cúpula y tarjeta de relevadores.



Figura 5: Vista exterior del gabinete del manejador de motor de cúpula y tarjeta de relevadores.

3.1 USO DE LA BOTONERA MANUAL

Sobre la tapa del gabinete del control de la cortina y gajo se colocaron 4 botones desde donde puede operarse el control (ver *Figura 6*). La funcionalidad de los botones se describe en la *Tabla 4*.

TABLA 4

Funcionalidad de los botones

Botón	Función
Abrir	Posiciona la cúpula en el contacto deslizante, abre la cortina totalmente y posteriormente el gajo.
Cerrar	Posiciona la cúpula en el contacto deslizante, cierra el gajo y posteriormente la cortina.
Abre gajo	Abre la cortina durante 5 segundos, se detiene y después abre el gajo.
Alto	Detiene la operación actual, que puede ser cualquiera de las anteriores.



Figura 6: Vista de la botonera del control de cortina y gajo.

4. LA MICROCOMPUTADORA BEAGLEBONE

La microcomputadora Beaglebone es de tecnología reciente y tiene las siguientes características: es de tamaño pequeño 3.25”x2.25”, corre el sistema operativo Linux, contiene un procesador ARM AM355x que corre a 720MHz, y tiene 256MB de RAM DDR2, 70 líneas de entrada-salida, conexión a red Ethernet y una entrada para codificador de cuadratura (como el utilizado para posicionar la cúpula), por lo que resulta adecuada para nuestra aplicación.

El sistema operativo Linux es fácil de mantener, y es más fácil depurar los programas desarrollados. Además ofrece la posibilidad de utilizar lenguajes de programación de alto nivel, como Python, que reducen el tiempo de desarrollo de los programas.

4.1 ASIGNACIÓN DE PATAS

La Tabla 5 muestra la asignación de patas de la computadora Beaglebone para esta aplicación. La primera columna muestra los nombres asignados a las señales utilizadas en la aplicación; la segunda columna muestra el nombre de la pata del procesador, como está indicado en su diagrama esquemático [3], que corresponde a cada señal. La columna “multiplexor” muestra el nombre del multiplexor correspondiente a la pata del procesador; cada pata es configurable y puede adoptar uno de siete modos de configuración, de acuerdo con lo señalado en el Manual del Procesador Cortex-A8 [4]. La columna “Dato Hex” muestra el valor en hexadecimal que hay que escribir en el archivo de configuración de cada multiplexor; estos archivos tienen el nombre del multiplexor correspondiente y se encuentran en el sistema de archivos de la computadora Beaglebone, en el directorio: “/sys/kernel/debug/omap_mux/”. El dato a escribir depende del modo de configuración y de si es entrada o salida, de acuerdo con lo señalado en el Manual de Referencia del Procesador Cortex-A8 [5].

TABLA 5

Asignación de patas de la computadora Beaglebone.

Serial	Pata	Multiplexor	Modo	Dato Hex
LED1_33	GPIO1_0	GPMC_AD0	GPIO1_0	0xF
LED2_33	GPIO1_4	GPMC_AD4	GPIO1_4	0xF
LED3_33	GPIO1_30	GPMC_CSN1	GPIO1_30	0xF
LED4_33	EHRPWM2A	GPMC_AD8	GPIO0_22	0xF
LED5_33	GPIO1_33	GPMC_AD13	GPIO1_13	0xF
LED6_33	EHRPWM2B	GPMC_AD9	GPIO0_23	0xF
LUZ1_33	GPIO0_26	GPMC_AD10	GPIO0_26	0xF
LUZ2_33	GPIO1_15	GPMC_AD15	GPIO1_15	0xF
COR_UP_33	GPIO1_29	GPMC_CSN0	GPIO0_29	0xF
COR_DW_33	TIMER7	GPMC_GPMC_OEN_REN	GPIO2_3	0xF
GAJ_DW_33	GPIO1_14	GPMC_CSN2	GPIO1_31	0xF
GAJ_UP_33	TIMER6	GPMC_GPMC_WEN	GPIO2_4	0xF
ST3-2	GPIO1_7	GPMC_AD7	GPIO1_7	0x2F
ST3-3	GPIO1_2	GPMC_AD2	GPIO1_2	0x2F
ST3-4	GPIO1_12	GPMC_AD12	GPIO1_12	0x2F
ST3-5	GPIO1_6	GPMC_AD6	GPIO1_6	0x2F
SENA_33	GPIO2_6	LCD_DATA0	GPIO2_6	0x2F
SC_A_33	GPIO2_9	LCD_DATA3	GPIO2_9	0x2F
SC_B_33	GPIO2_11	LCD_DATA5	GPIO2_11	0x2F
SC_C_33	GPIO2_23	LCD_HSYNC	GPIO2_23	0x2F
SC_D_33	GPIO2_12	LCD_DATA6	GPIO2_12	0x2F
Codificador A	UART4_CTSN	LCD_DATA12	eQEP1A_in	0x2A
Codificador B	UART4_RTSN	LCD_DATA13	eQEP1B_in	0x2A
CUP_DER	EHRPWM1A	GPMC_A2	GPIO1_18	0xF
CUP_IZQ	EHRPWM1B	GPMC_A3	GPIO1_19	0xF
SW_ORIGEN	UART2_TXD	SPIo_Do	GPIO0_3	0x2F
STROBE	UART5_RTSN	LCD_DATA15	eQEP1_strobe	0x2A
INDEX	UART5_CTSN	LCD_DATA14	eQEP1_index	0x2A

4.2 PROGRAMAS DESARROLLADOS

Para el posicionamiento de la cúpula y para la apertura de la cortina y el gajo, se desarrolló una biblioteca de funciones y un programa de aplicación.

La biblioteca denominada “cupula_84_lib.c” está hecha en lenguaje C y contiene las funciones básicas que interactúan directamente con los puertos de entrada-salida, el codificador de posición de la cúpula y los registros internos de configuración.

El programa de aplicación se denomina “cupula_84.py”, está hecho en lenguaje Python y contiene las funciones que manejan el socket de comunicación, así como las funciones que realizan secuencias para hacer tareas más complicadas. Por ejemplo: para abrir la cortina, el programa posiciona la cúpula en el contacto deslizante, posteriormente abre la cortina y supervisa que el movimiento se realice satisfactoriamente.

El programa “cupula_84.py” utiliza las funciones disponibles en la biblioteca “cupula_84_lib.c”; esto es posible compilando la biblioteca como un Objeto y llamándola desde el programa en Python. La razón de utilizar ambos lenguajes se debe a que es más sencillo acceder los registros de 16 bits del codificador de cuadratura desde una función en C, y el manejo de la comunicación con hilos (Threads) es más sencillo de programar en Python.

4.2.1 LA BIBLIOTECA “CUPULA_84_LIB.C”

La biblioteca “cupula_84_lib.c” es una biblioteca de 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 y al módulo de lectura del codificador de cuadratura. El programa realiza un mapeo de memoria para direccionar los registros de configuración. En el Apéndice F se muestra la sección del programa que realiza la configuración. Las funciones disponibles en la biblioteca se listan en la Tabla 6.

TABLA 6

Funciones disponibles en la biblioteca “cúpula_84_lib.c”

Función	Descripción
configura ()	Configura multiplexores y registros del codificador de cuadratura.
Enciende_led(int cual)	Enciende el led indicado por la variable “cual” que es un número entre 1 y 5.
Apaga_led(int cual)	Apaga el led indicado por la variable “cual” que es un número entre 1 y 5
enciende_luz(int cual)	Enciende la luz indicada por la variable “cual” que es un número entre 1 y 2
apaga_luz(int cual)	Apaga la luz indicada por la variable “cual” que es un número entre 1 y 2
abre_cortina()	Acciona el relevador que abre la cortina.

Cierra_cortina()	Acciona el relevador que cierra la cortina.
Abre_gajo()	Acciona el relevador que abre el gajo.
Cierra_gajo()	Acciona el relevador que cierra el gajo.
Alto_cortina()	Detiene el movimiento de cortina abriendo o cerrando.
Alto_gajo()	Detiene el movimiento de gajo abriendo o cerrando.
Alto()	Detiene el movimiento de gajo, cortina y cúpula.
Lee_boton_open()	Lee botón “Abrir” instalado en la caja de control.
Lee_boton_close()	Lee botón “Cerrar” instalado en la caja de control.
Lee_boton_gajos()	Lee botón “Abre gajo” instalado en la caja de control.
Lee_boton_stop()	Lee botón “Alto” instalado en la caja de control.
Lee_sw_trolley()	Lee el interruptor que indica que el contacto deslizante está en posición.
Lee_corriente_a()	Lee el estado del sensor de corriente A.
lee_corriente_b()	Lee el estado del sensor de corriente B.
lee_corriente_c()	Lee el estado del sensor de corriente C.
lee_corriente_d()	Lee el estado del sensor de corriente D.
gira_cup_der()	Gira la cúpula hacia la derecha.
Gira_cup_izq()	Gira la cúpula hacia la izquierda.
Alto_cupula()	Detiene el movimiento de la cúpula.
Lee_sw_origen()	Lee el interruptor de referencia de posición de la cúpula.
Posicion_cupula()	Regresa el valor del contador de posición de cúpula.
Limpia_banderas_codificador()	Limpia el registro de banderas del codificador de posición de cúpula.
Revisa_si_toco_sw_origen()	Regresa el estado de la bandera que indica que el interruptor de referencia de posición de cúpula fue activado.

4.2.2 EL PROGRAMA “CUPULA_84.PY”

El programa “cupula_84.py” se hizo en lenguaje Python y su estructura se muestra en la *Figura 7*. Básicamente está formado por tres hilos de programa que se ejecutan en paralelo: Comunicación, Control y Atención a botones. La comunicación entre hilos se realiza mediante la “Cola de instrucciones” y dos variables globales: “alto” y “estatus”.

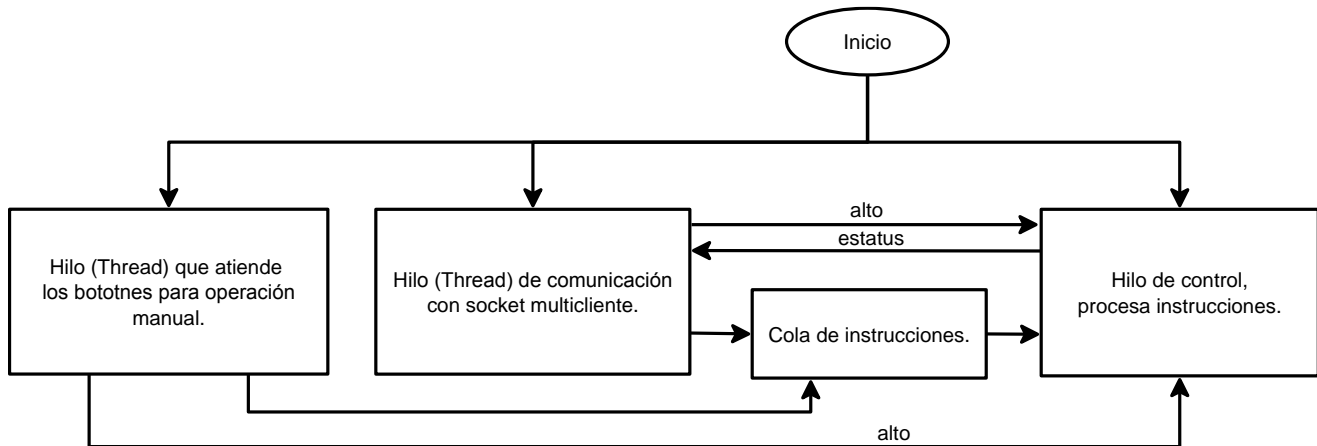


Figura 7: Estructura del programa “Cúpula_84.py”.

El hilo de comunicación abre un socket TCP servidor en la dirección 192.168.0.109 y puerto 4545. El socket es multicliente, 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 que pueden ser recibidas por este socket se listan en la Tabla 7.

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. Si se presiona el botón “Alto” simplemente se levanta la variable global “alto” para que el hilo de control detenga lo que esté ejecutando.

El hilo de control saca instrucciones de la cola y las ejecuta, para ello utiliza las instrucciones de la biblioteca “cupula_84_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 cambia a un estado activo se suspende cualquier tarea que se esté realizando y se vacía la cola de instrucciones.

TABLA 7

Instrucciones que recibe el programa “cupula_84.py”.

Instrucción	Descripción
:MXXX;	Mueve la cúpula a la posición XXX en grados.
:O;	Inicializa la cúpula.
:S;	Detiene el movimiento de cúpula.
:J+;	Gira la cúpula hacia la derecha hasta que se recibe el mando :S;.
:J-;	Gira la cúpula hacia la izquierda hasta que se recibe el mando :S;.
abre_cortina	Abre la Cortina.
Cierra_cortina	Cierra la Cortina.
Abre_gajo	Abre el gajo.
Cierra_gajo	Cierra el gajo.
Busca_trolley	Gira la cúpula hacia la derecha hasta que toca el interruptor de inicio. Enviar “alto” para detener la operación.
Open	Posiciona la cúpula en el contacto deslizante, abre la cortina totalmente y posteriormente el gajo.
Close	Posiciona la cúpula en el contacto deslizante, cierra el gajo y posteriormente la cortina.
enciende_luz	Enciende la luz del piso de telescopio.
apaga_luz	Apaga la luz del piso de telescopio.
Alto	Detiene las operaciones.

4.3 CONFIGURACIÓN DE LA MICROCOMPUTADORA BEAGLEBONE

En caso de que la microcomputadora Beaglebone se dañe, será necesario sustituirla por otra igual. En esta sección se indican los pasos necesarios para configurar la nueva Beaglebone.

4.3.1 INSTALACIÓN Y CONFIGURACIÓN DEL SISTEMA OPERATIVO

La instalación del sistema operativo Linux Angstrom se hace con los siguientes pasos:

1. Descargar del sitio:

`downloads.angstrom-distribution.org/demo/beaglebone/archive/`

la siguiente imagen:

`Angstrom-Cloud9-IDE-eglibc-ipk-v2012.04-core-beaglebone-ro.img.xz`

2. Insertar la memoria SD card en la PC y descomprimir la imagen en el dispositivo:

```
$sudo -s
```

```
# xz -dkc Angstrom-Cloud9-IDE-eglibc-ipk-v2012.04-core-beaglebone-ro.img.xz > /dev/sdX
```

```
#exit
```

3. Insertar la SD Card en el Beaglebone.

4.3.2 CONFIGURACIÓN DE UNA DIRECCIÓN DE IP FIJA

Para asignar al Beaglebone una dirección de IP fija que se mantenga al reinicializar el microcontrolador, es necesario seguir los siguientes pasos:

1. En el directorio `/etc/init.d` crear un script llamado "fixnetwork" con el siguiente contenido:

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

```
/etc/init.d/networking restart
```

```
cp /etc/resolv.ok /etc/resolv.conf
```

El archivo `resolv.ok` viene en el sistema operativo Angstrom.

2. Hacerlo ejecutable:

```
chmod +x fixnetwork
```

3. Agregarlo al "startup":

```
update-rc.d fixnetwork defaults
```

4. Agregar los datos de la red en el archivo `/etc/networks/interfaces`. En el caso del control de la cúpula del 84cm, agregar lo siguiente en el archivo:

```
# Wired or wireless interfaces
```

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.0.109
```

```
netmask 255.255.255.0
```

```
network 192.168.0.0
```


4.3.3 CONFIGURACIÓN PARA EJECUTAR EL PROGRAMA DE APLICACIÓN AL ARRANCAR EL OPERATIVO

Para configurar el operativo para que ejecute un programa al arranque del Beaglebone, seguir los siguientes pasos:

1. Copiar en el directorio raíz los archivos: `cupula_84.lib` y `cupula_84_lib.o`
2. En el directorio `/usr/bin` crear un archivo con el nombre "`autoejecuta.sh`" con el siguiente contenido:

```
#!/bin/bash
python /home/root/cupula_84.py
```

3. Hacerlo ejecutable:
`chmod u+x /usr/bin/autoejecuta.sh`
4. En el directorio `/lib/systemd/` crear el archivo de servicio con el nombre "`autoejecuta.service`" con el siguiente contenido:

```
[Unit]
Description=Run the Cupula84 pattern script
After=syslog.target network.target
```

```
[Service]
Type=simple
ExecStart=/usr/bin/autoejecuta.sh
```

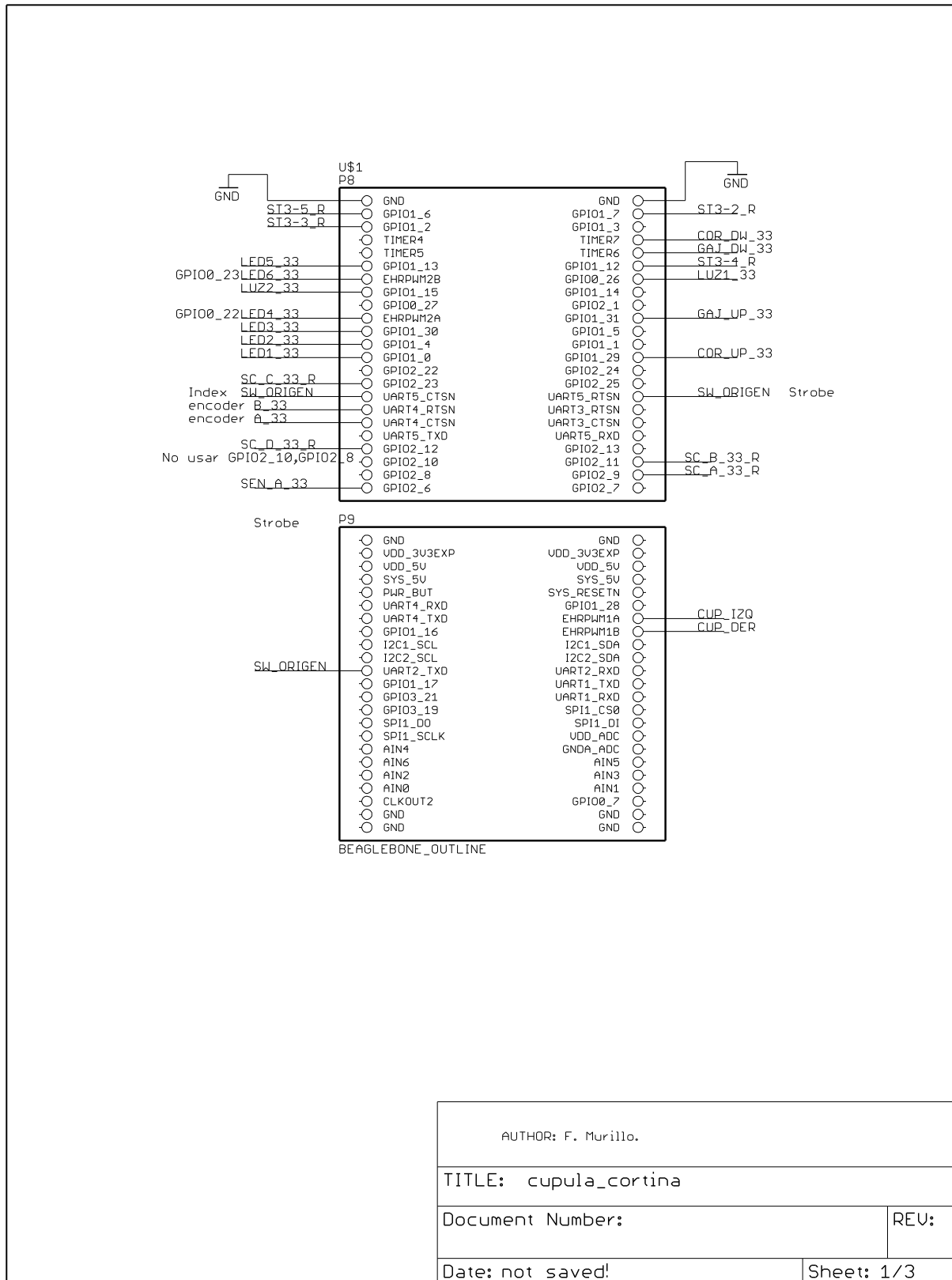
```
[Install]
WantedBy=multi-user.target
```

5. Crear una liga:
`cd /etc/systemd/system`
`ln /lib/systemd/autoejecuta.service autoejecuta.service`
6. Habilitar el servicio para que se ejecute al arrancar el sistema:
`systemctl enable autoejecuta.service`

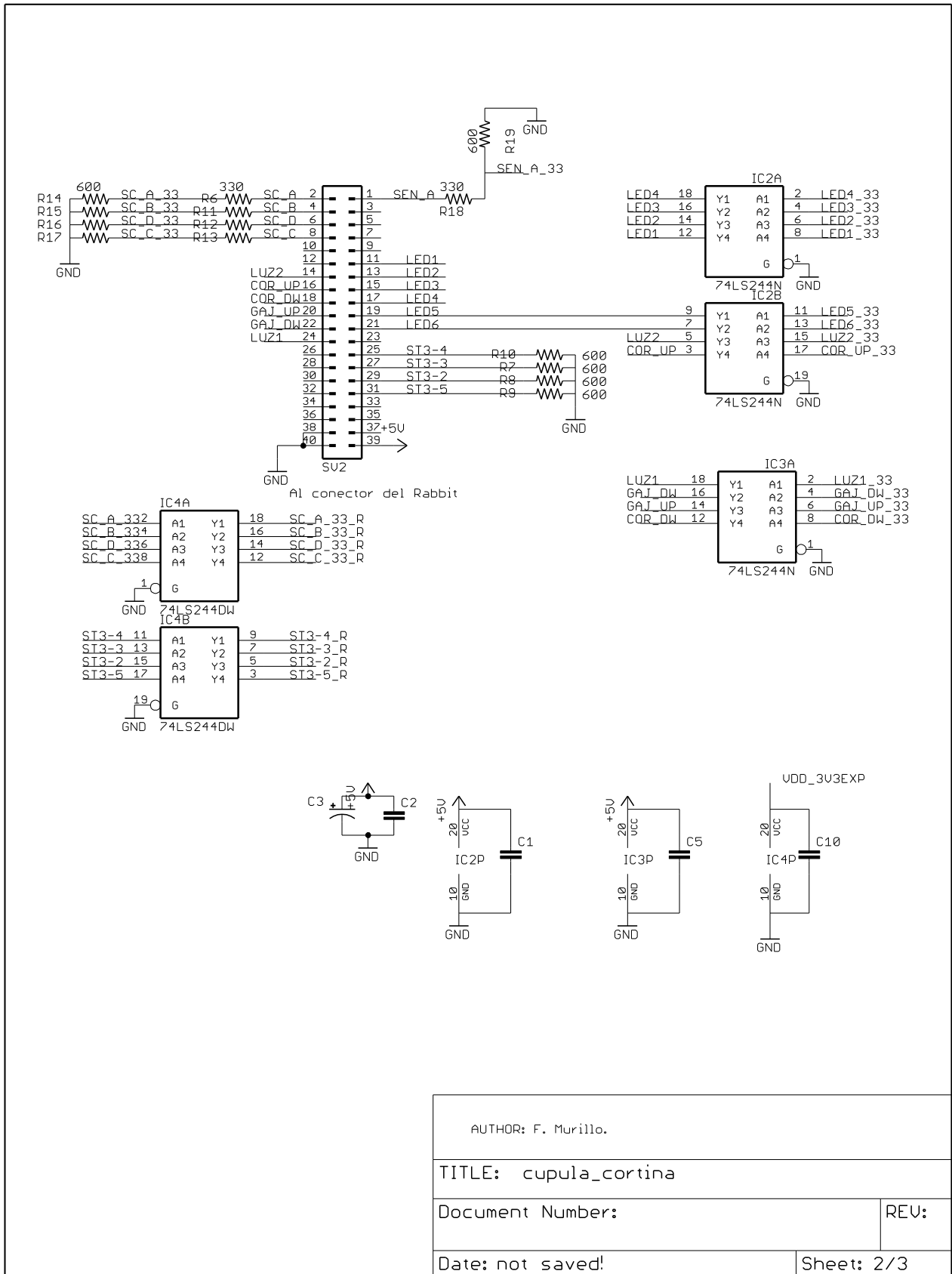
5. REFERENCIAS

- [1] Lazo, F., Zazueta, S., Murillo, F., Ochoa, J.L., García, B., Murillo, J.M.
“Control de movimiento y posicionado para los domos de los telescopios de 1.5m y 84cm. Versión Ethernet”.
Publicaciones Técnicas del Instituto de Astronomía, UNAM.
Reporte Técnico. RT-2012-02.
México, Junio, 2012.
- [2] Quirós, F., Hiriart, D., Martínez, B., Colorado, E., Murillo, J.M., Ochoa, J.L., Murillo, F., Córdoba, A., Guisa, G., López, E.
“Automatización de apertura de cortina y gajo del Telescopio de 84cm y 1.5 m del OAN-SPM”.
Publicaciones Técnicas del Instituto de Astronomía, UNAM.
Reporte Técnico. RT-2010-04.
México.
- [3] BONE_SCH Beaglebone Rev. a5a. Document Number 450-5500-001. Página 11.
- [4] AM335x ARM Cortex-A8 Microprocessors (MPUs) (Rev D).
- [5] AM335x ARM Cortex-A8 Microprocessors (MPUs)
Technical Reference Manual. Página 876.

APÉNDICE A. DIAGRAMA ESQUEMÁTICO Y PCB DE LA TARJETA DEL CONTROLADOR



AUTHOR: F. Murillo.	
TITLE: cupula_cortina	
Document Number:	REV:
Date: not saved!	Sheet: 1/3



AUTHOR: F. Murillo.

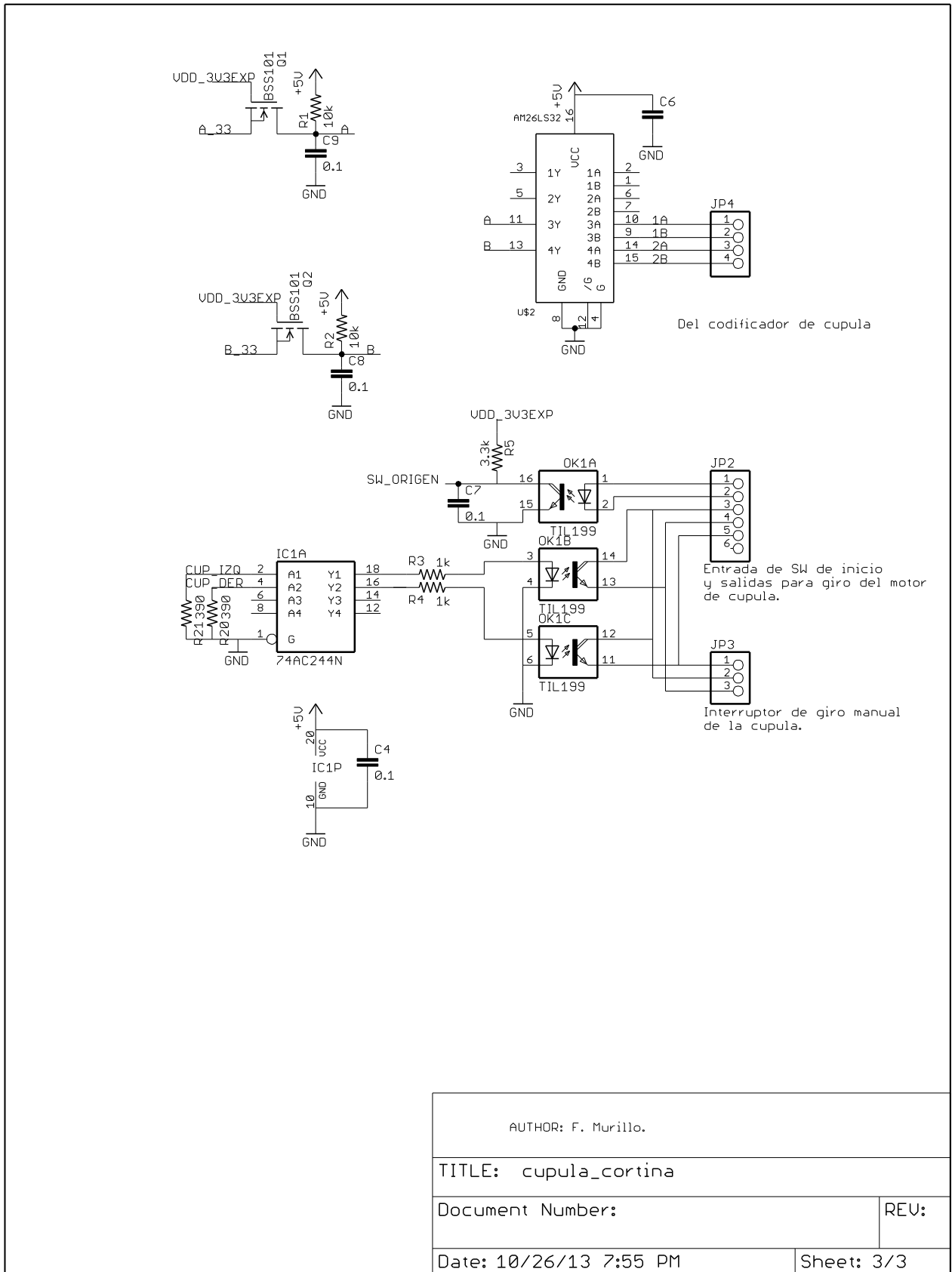
TITLE: cupula_cortina

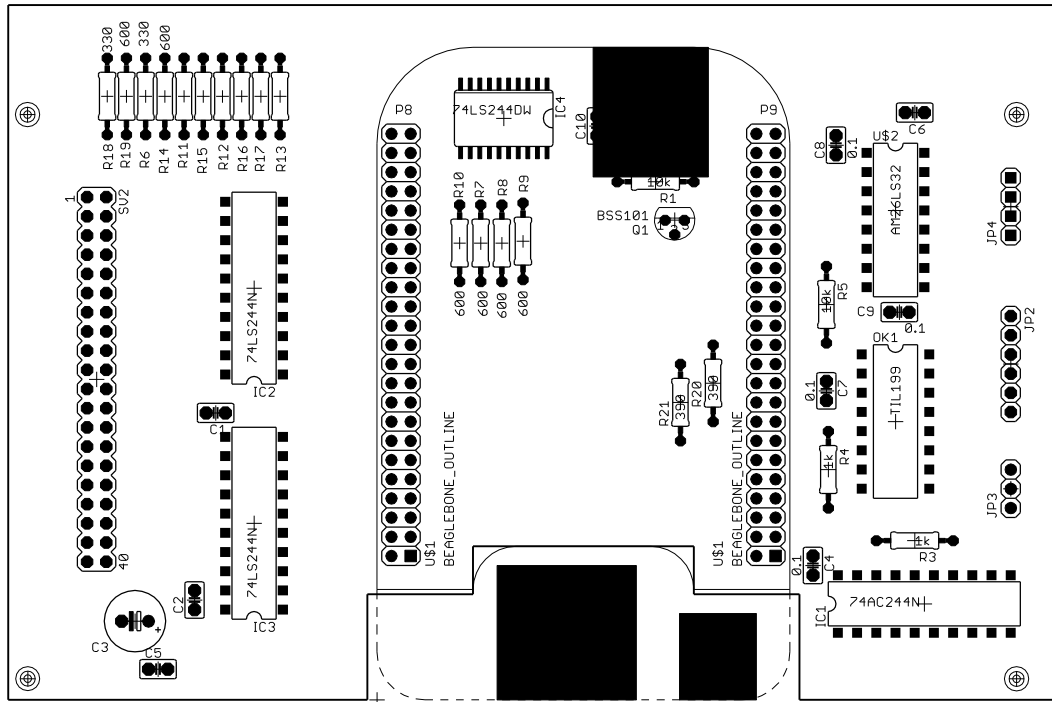
Document Number:

REV:

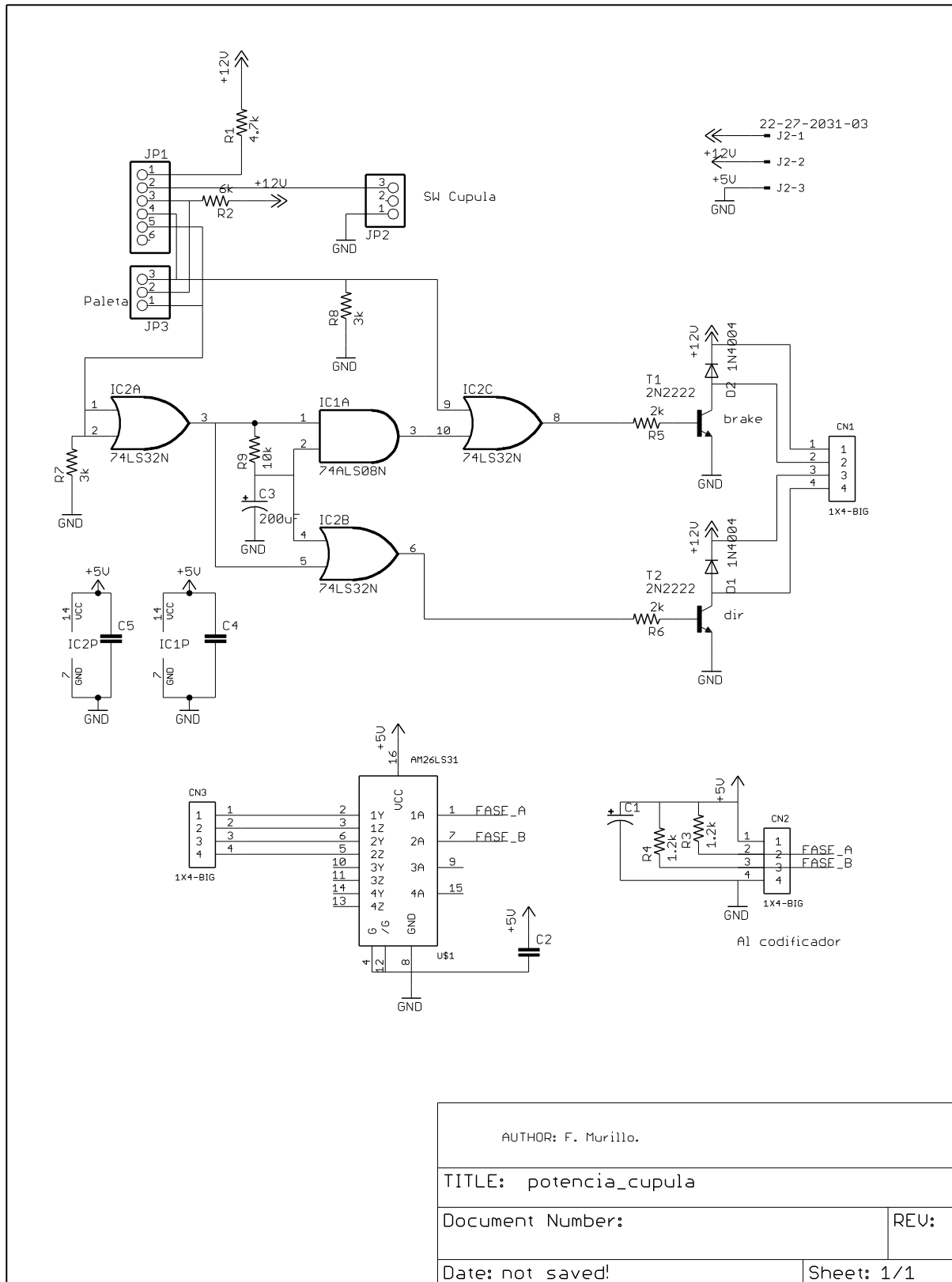
Date: not saved!

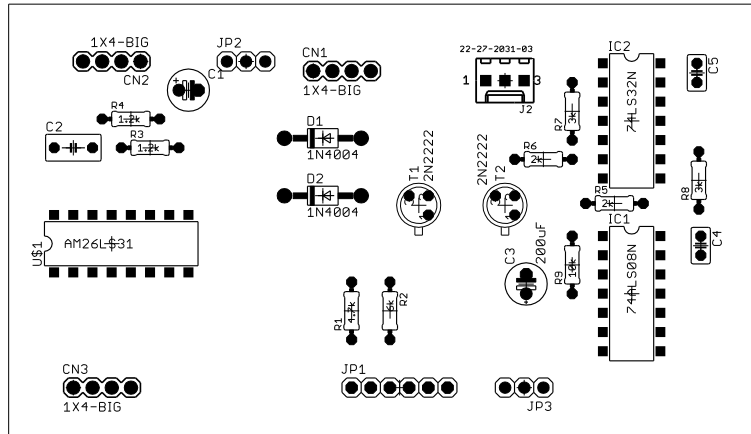
Sheet: 2/3



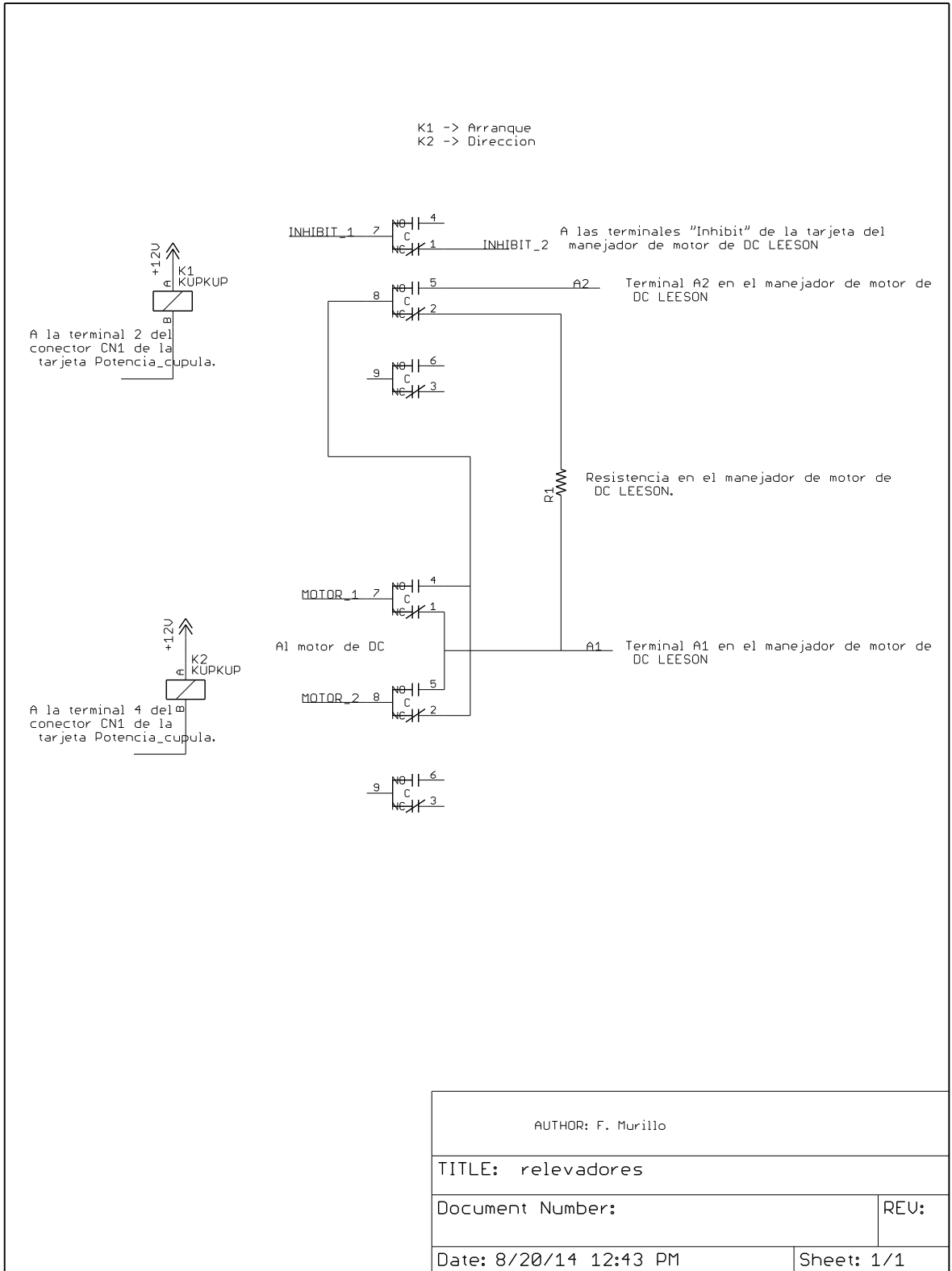


APÉNDICE B: DIAGRAMA ESQUEMÁTICO Y MAPA DE COMPONENTES DE LA TARJETA DE RELEVADORES Y CODIFICADOR DE CÚPULA





APÉNDICE C: DIAGRAMA ESQUEMÁTICO DEL CABLEADO DE RELEVADORES Y MANEJADOR MARCA LEESON MODELO M1740006 DEL MOTOR DE CÚPULA



APÉNDICE D. DIAGRAMA ESQUEMÁTICO Y MAPA DE COMPONENTES DE LA TARJETA DE RELEVADORES Y SENSORES DE CORTINA Y GAJO

10

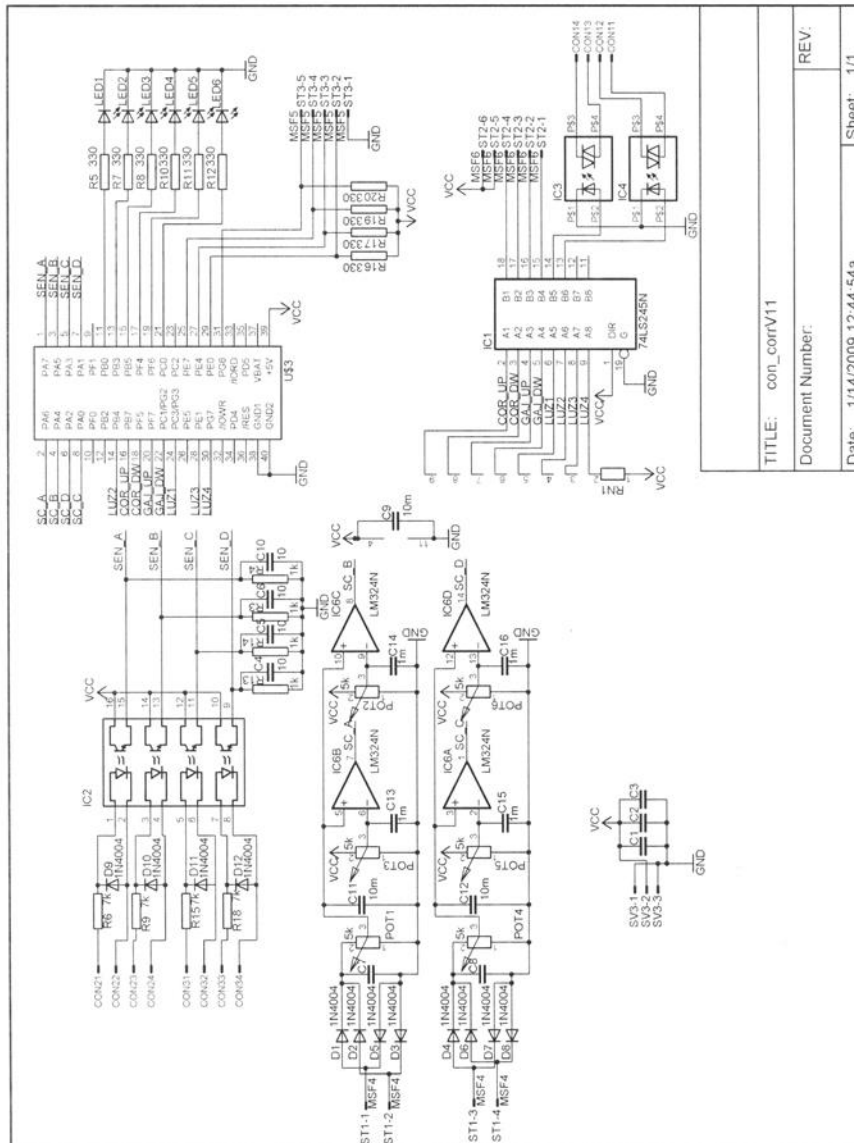


Figura A.1. Diagrama esquemático.

Departamento de Instrumentación, Ensenada, IA-UNAM

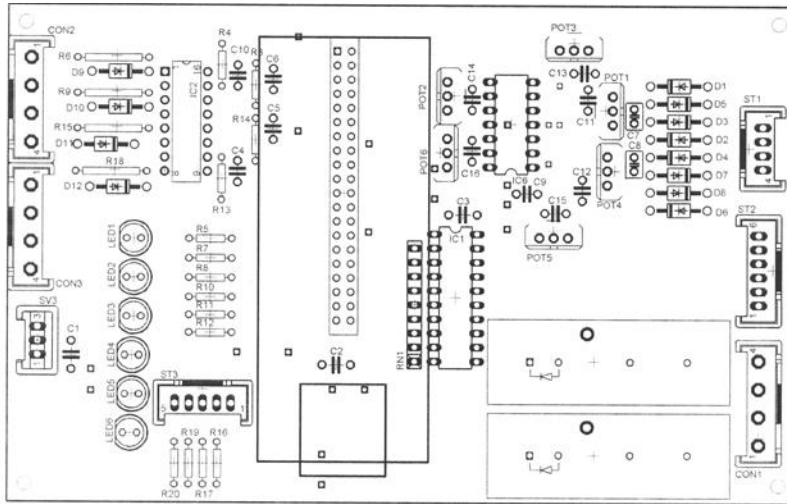


Figura A.2. Distribución de componentes.

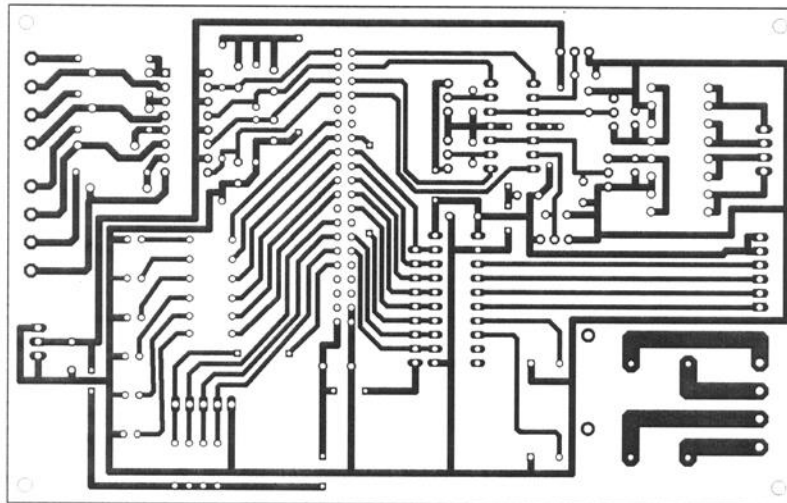


Figura A.3. Mascara del circuito impreso del lado de soldadura.

APÉNDICE E. CONFIGURACIÓN DE LOS REGISTROS DEL BEAGLEBONE

```
void configura()
{
    // Declaracion de apuntadores para las regiones de memoria correspondientes
    // a los registros de configuracion:

    char *map_GPIO0, *map_GPIO1, *map_GPIO2, *map_PWMSS1, *map;
    int fd;

    configura_multiplexores();

    // Abre el archivo para mapeo de memoria
    fd = open("/dev/mem", O_RDWR);

    // Realiza los mapeos de memoria de los registros de entrada-salida y modulo PWMSS1, que
    // contiene el decodificador de cuadratura:

    map_GPIO0 = mmap(0, MMAP_GPIO0_size, PROT_WRITE | PROT_READ, MAP_SHARED, fd, MMAP_GPIO0_offset);
    map_GPIO1 = mmap(0, MMAP_GPIO1_size, PROT_WRITE | PROT_READ, MAP_SHARED, fd, MMAP_GPIO1_offset);
    map_GPIO2 = mmap(0, MMAP_GPIO2_size, PROT_WRITE | PROT_READ, MAP_SHARED, fd, MMAP_GPIO2_offset);
    map_PWMSS1 = mmap(0, MMAP_PWMSS1_size, PROT_WRITE, MAP_SHARED, fd, MMAP_PWMSS1_offset);
    map = mmap(0, MMAP_size, PROT_WRITE | PROT_READ, MAP_SHARED, fd, MMAP_offset);

    // Configura los puertos de salida

    *(int *)(map_GPIO0 + GPIO_OE) = ~(1 << LED4) & ~(1 << LED6) & ~(1 << LUZ1);
    *(int *)(map_GPIO1 + GPIO_OE) = ~(1 << LED1) & ~(1 << LED2) & ~(1 << LED3) & ~(1 << LED5) & ~(1 <<
    LUZ2) & ~(1 << CUP_IJQ) & ~(1 << CUP_DER) & ~(1 << CORT_UP) & ~(1 << GAJ_UP);
    *(int *)(map_GPIO2 + GPIO_OE) = ~(1 << CORT_DW) & ~(1 << GAJ_DW);

    // Configura los registros del decodificador de cuadratura:

    *(unsigned short *)(map_PWMSS1 + QEPCTL)=0x1808;
    *(unsigned short *)(map_PWMSS1 + QDECCTL)=0x400;
    *(int *)(map_PWMSS1 + QPOSMAX)=2000;    // Cuenta maxima del codificador
    *(int *)(map_PWMSS1 + QPOSINIT)=482;    // Posicion del switch de origen
    *(int *)(map_PWMSS1 + QCLR)=0X200;

}
}
```

