

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 1 de 20

Revisión	Fecha	Comentario	Autor
0	31/12/2008	<ul style="list-style-type: none"> <li>- La presente nota consiste en el resumen y divulgación de un trabajo de cátedra realizado sobre la base de un módulo RCM3720.</li> <li>- El modelo realizado, objeto del trabajo original (v1.0), fue presentado durante el 2º semestre de 2008 en varias muestras de tecnología.</li> <li>- El documento original completo y el material bibliográfico referenciado podrán ser accedidos como documentos anexos a este trabajo.</li> <li>- Si el lector estuviese interesado en algún aspecto de la implementación de este proyecto puede comunicarse con nosotros a fines de solicitar más información.</li> </ul>	Ulises Bigliati

 <p><b>CONTINEA</b> Microprocesamiento modular + Conectividad</p>	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 2 de 20

## índice

<b>INTRODUCCIÓN.....</b>	<b>3</b>
EL ROBOSCOPIO .....	3
<i>Que es?</i> .....	3
<i>Que hace?</i> .....	3
<i>Como lo hace?</i> .....	3
<i>Para que?</i> .....	3
EL PROYECTO (OVERVIEW) .....	4
<b>EL SOFTWARE DE ASTRONOMÍA.....</b>	<b>5</b>
SKYMAP, EL SOFTWARE UTILIZADO EN EL PROYECTO.....	5
EL DRIVER DE COMUNICACIONES SKYMAP/ROBOSCOPIO .....	6
<i>Introducción</i> .....	6
<i>Arquitectura del esquema de interacción de los drivers con Skymap</i> .....	6
<i>Telescope control DLL (scope.dll)</i> .....	6
EL DRIVER ROBOSCOPE.DRV .....	7
<b>EL HARDWARE DEL ROBOSCOPIO .....</b>	<b>9</b>
VISTA GLOBAL DEL SISTEMA .....	9
<b>EL FIRMWARE DEL ROBOSCOPIO .....</b>	<b>10</b>
ESTRUCTURA DE ARCHIVOS DEL PROYECTO.....	11
LOS MÓDULOS DEL PROYECTO ROBOSCOPIO .....	11
<i>Biblioteca de funciones para astronomía</i> .....	11
<i>Biblioteca de funciones de dispositivos de hardware</i> .....	12
<i>Biblioteca de funciones accesorias</i> .....	12
<i>Biblioteca de funciones para comunicación</i> .....	12
<i>Roboscopio_v1.11, el programa</i> .....	13
El comando SLEW .....	13
La función Auto-Track .....	13
<i>La Interfaz Web</i> .....	13
<b>ESQUEMA DEL MODELO MECÁNICO A CONSTRUIR .....</b>	<b>15</b>
<b>PRIMER MODELO CONSTRUIDO .....</b>	<b>16</b>
<b>CONCLUSIONES.....</b>	<b>17</b>
EVOLUCIÓN EN TRES EJES DE OPTIMIZACIÓN.....	17
<i>Mecánica: Trenes de engranajes.</i> .....	17
<i>Electrónica: Sensores magnéticos, auto-alineación</i> .....	17
<i>Software embebido: Conectividad TCP/IP, Roboscopio on-line.</i> .....	17
LIMITACIONES DETECTADAS .....	18
CONSIDERACIONES FINALES .....	18
<b>REFERENCIAS.....</b>	<b>19</b>
EL SOFTWARE DE ASTRONOMÍA .....	19
EL HARDWARE DEL ROBOSCOPIO .....	19
EL FIRMWARE DEL ROBOSCOPIO .....	20
EVOLUCIÓN[...]SENSORES MAGNÉTICOS.....	20

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 3 de 20

## Introducción

El presente trabajo fue realizado en el marco de la robótica aplicada, en función de lo cual debimos elegir una temática práctica sobre la cual elaborar nuestro proyecto. Esta temática tenía que presentar algún aspecto que permitiera la articulación de nuestro trabajo con algún sistema preexistente dentro del área de aplicación, lo cual implicó el estudio y comprensión de las características de la disciplina seleccionada, es decir, la astronomía.

Nos hemos propuesto entonces, aplicar la robótica a la astronomía y como resultado de esto surgió nuestro RoboScopio y como una primera aproximación a este, continuación definiremos su alcance.

## *El Roboscopio*

### Que es?

Roboscopio: contracción de los términos Robot y Telescopio. El Roboscopio, fue el nombre con el cual bautizamos a nuestro proyecto y fue el nombre del primer prototipo. Como su nombre lo indica, reúne en un mismo dispositivo a un telescopio y a un robot. Es decir, un telescopio robotizado. Por supuesto que esto no se trata de nada nuevo, pero intentaremos crear un prototipo dotado de las capacidades necesarias para operar eficazmente en la práctica y además, dejaremos el camino listo para incorporar funcionalidades si no novedosas, al menos atractivas, para nuestro modelo.

### Que hace?

El modelo construido es capaz de interactuar con un software profesional de astronomía.

Puede recibir informáticamente coordenadas de objetos celestes y ubicarlos con precisión por sus propios medios.

Puede reportar al software su orientación actualizada en tiempo real

Puede realizar el seguimiento de la trayectoria aparente del objeto celeste indicado compensando el movimiento diurno terrestre.

Posee una interfaz Web desde la cual pueden monitorearse sus variables de operación, y también comandar sus movimientos.

### Como lo hace?

Un microprocesador realiza los cálculos matemáticos involucrados en las operaciones de observación astronómica. Maneja las comunicaciones con el software mediante un puerto serie, y expone la interfaz Web mediante una conexión Ethernet.


Para mover el telescopio, se vale de sus puertos de entrada/salida para comandar motores paso a paso. Un software de computadora especializado en la cartografía astronómica es quien envía los comandos y las coordenadas.

Un tercer elemento debe entrar en juego: el driver de dispositivo para Windows que construimos para interfacear el software y nuestro prototipo. Este driver fue escrito en C++.

### Para que?

La asistencia de la robótica en la astronomía tiene varias ventajas:

- Acercar al aficionado al facilitar enormemente la ubicación de los objetos.
- Asistir en la observación mediante el seguimiento autónomo de los cuerpos celestes. Esto último permite: Facilitar el estudio de un objeto celeste y realizar fotografías astronómicas, ya que estas normalmente requieren una prolongada exposición.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 4 de 20

También ahorra tiempo porque permite ir directo al punto de estudio, evitando tiempos de búsqueda mucho mas largos en comparación con el uso tradicional de las cartas astronómicas, que por otro lado, deben hacerse en medio de la oscuridad.

## ***El proyecto (overview)***

Hecha la introducción, solo resta brindar un panorama general del trabajo antes de comenzar a recorrer el proyecto. Este se organiza en torno a tres ejes fundamentales que serán abordados no necesariamente en este orden:

- ❑ El software de astronomía, que se compone de:
  - Software de aplicación
  - Driver de dispositivo
- ❑ Los sistemas de coordenadas.
- ❑ El prototipo, que a su vez involucra:
  - Hardware
  - Firmware


Esto implica:

- ❑ Indagar en el mercado y en el ambiente de la astronomía sobre que programas de aplicación existen relacionados con esta ciencia y decidir cual sería conveniente utilizar para nuestro propósito, y por supuesto aprender a utilizarlo.
- ❑ Construir un driver para gestionar la interfaz entre el software utilizado y nuestro prototipo.
- ❑ Estudiar los fundamentos clave de la observación astronómica. Lo cual requiere la familiarización con sus métodos, instrumental, herramientas y conceptos teóricos básicos.
- ❑ Construir un prototipo electromecánico adecuado para modelizar adecuadamente nuestros conceptos.
- ❑ Escribir el firmware que soporte todas las funcionalidades buscadas.

El presente informe se ha constituido como una documento central y compacto con el fin de ofrecerle al lector una descripción que le permita tomar un rápido conocimiento del proyecto.

Una serie de referencias bibliográficas mas el acceso al documento completo del trabajo permitirá la ampliación necesaria para completar debida e integramente la descripción del proyecto.

A continuación pasamos a realizar una reseña sobre los bloques que conforman la realización del Roboscopio.

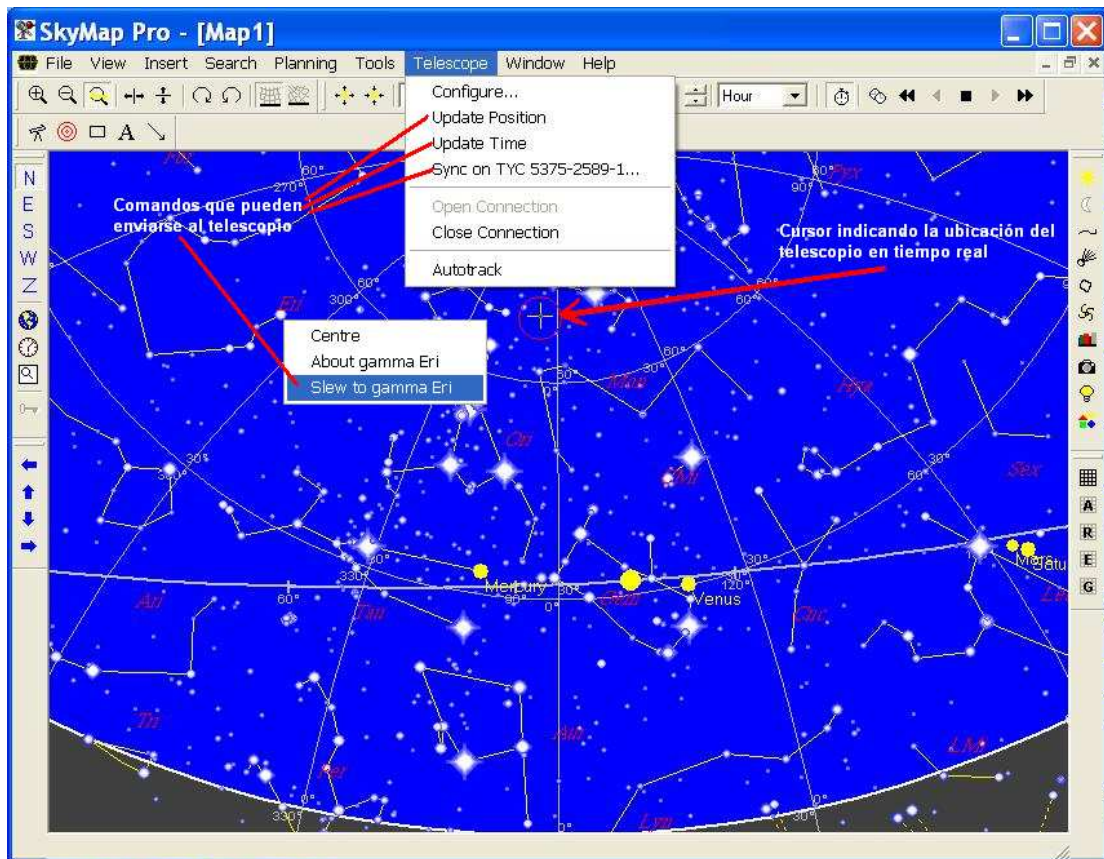
	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 5 de 20

## El software de astronomía


Existen en el mercado una gran variedad de software de apoyo a la astronomía, estas aplicaciones permiten construir cartas astronómicas que ofrecen la posibilidad de realizar con extremada precisión la identificación y posterior ubicación de cuerpos celestes en el firmamento en función de la ubicación geográfica y de la fecha y hora de observación.

### ***Skymap, el software utilizado en el proyecto***

El software de astronomía SkyMap [SKY] puede comunicarse con una amplia variedad de telescopios controlados por computadora. Al conectarse a uno de estos dispositivos, entre otras funciones, SkyMap muestra un cursor parpadeando sobre el mapa estelar que indica a donde está apuntando actualmente el telescopio. Si el aparato posee otras habilidades, como por ejemplo la función "slew" que consiste en posicionar por software el telescopio en un destino particular, pueden ser controladas desde el Skymap. La forma en que se utilizan las funciones del telescopio desde el software es a través de drivers de dispositivo, que físicamente están ubicados en el directorio "Drivers" en el directorio de instalación de SkyMap. Cada driver provee las particularidades necesarias para controlar el dispositivo para el que fuera creado; nuevos drivers pueden ser fácilmente agregados para controlar nuevos dispositivos. Para esto, una biblioteca de funciones para C++ junto con toda la documentación necesaria es provista a través del web site de SkyMap.



Screen shot del software SkyMap

	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 6 de 20

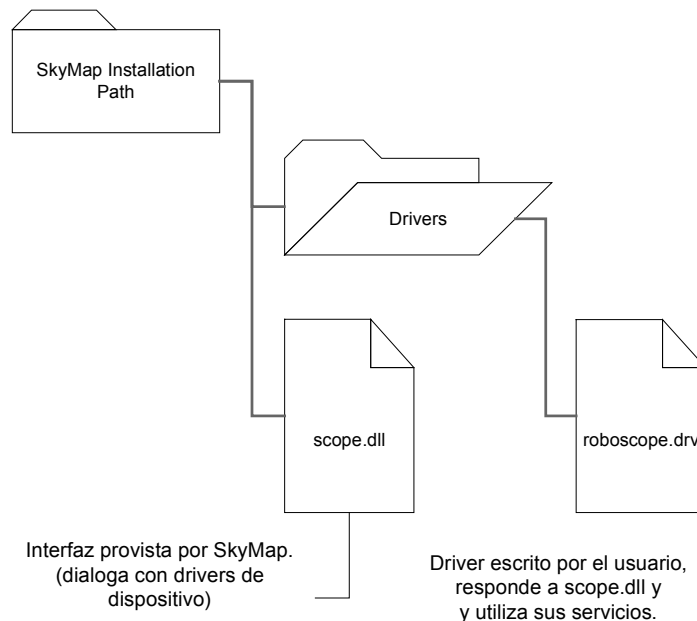
## ***El driver de comunicaciones Skymap/Roboscopio***

### **Introducción**

El fabricante del software Skymap le ofrece al usuario la posibilidad de articular esta la aplicación con su telescopio mediante la intermediación de un driver de dispositivo que básicamente consiste en un archivo DLL escrito para Win32. Normalmente los telescopios motorizable que se encuentran en el mercado proveen su propio driver para interactuar con diferentes programas. En este caso, el fabricante nos proporciona una interfaz con su programa (scope.dll) [SKY-DRV] desde la cual podemos aprovechar todas las funcionalidades y a la cual podemos acceder fácilmente desde nuestro driver escrito en C++.

### **Arquitectura del esquema de interacción de los drivers con Skymap**

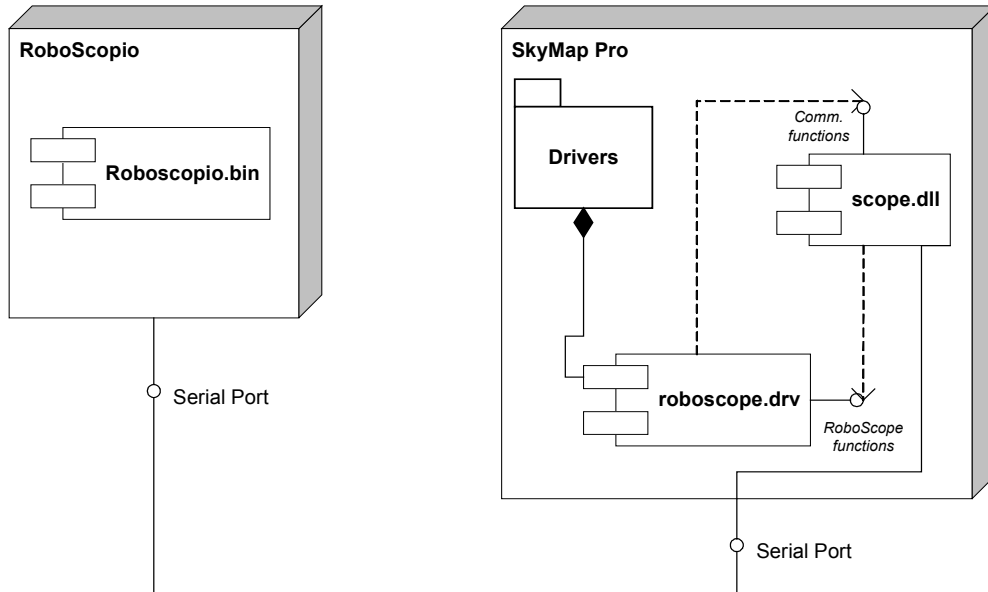
Un driver para SkyMap Pro [SKY-DRV] es una DLL (dynamic link library) para Windows de 32-bit renombrada con la extensión ".drv", y que no requiere instalación ni registro en el sistema operativo, simplemente se debe copiar en el subdirectorio "drivers" de la ruta de instalación del SkyMap Pro. El driver DLL debe "exportar" una serie de funciones que deben poseer unos nombres especiales para poder ser llamados desde el software SkyMap Pro. Cuando SkyMap Pro carga un driver para un telescopio interroga al driver para descubrir que funciones admite y cuales no. Esto hace realmente muy fácil escribir un driver con funcionalidades básicas e ir sumandoles gradualmente nuevas características.



### **Telescope control DLL (scope.dll)**

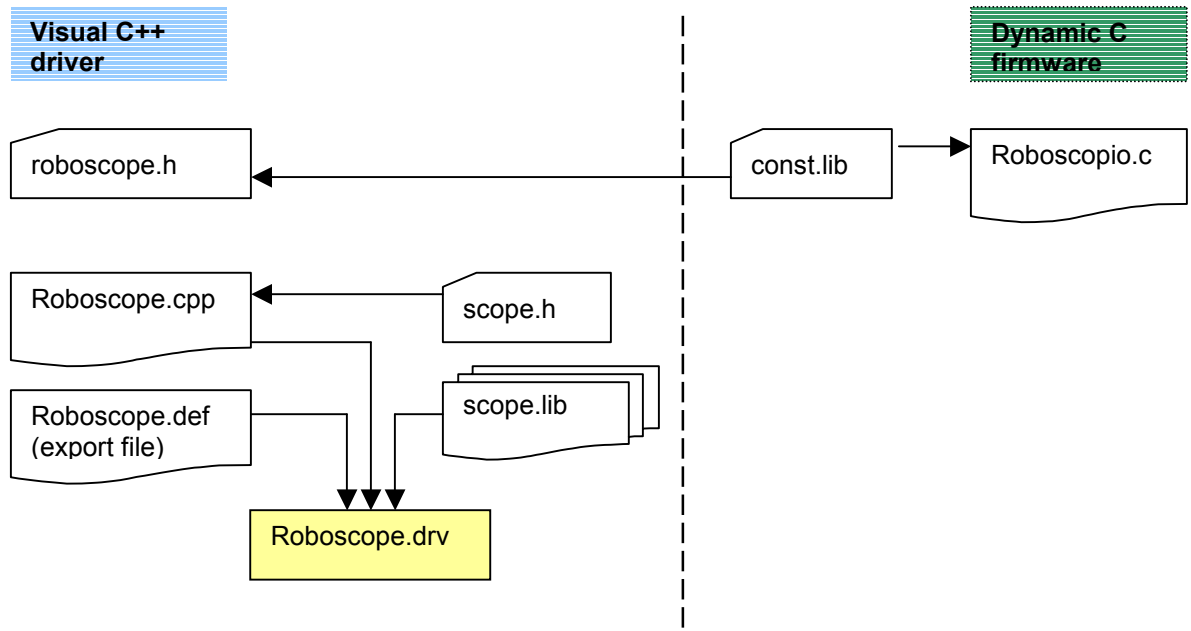
Toda las funciones de control del telescopio en SkyMap Pro son manejadas via la DLL, "scope.dll", que está localizada en el subdirectorio "system" del directorio de instalación del SkyMap Pro. Cuando desde el driver que nosotros escribimos debemos ejecutar alguna función de control sobre el telescopio, en realidad estamos pasando a través de la interfaz que provee scope.dll.

Scope.dll llama a nuestras funciones, y nuestras funciones utilizan los servicios, a su vez, de scope.dll que exporta una serie de funciones de alto nivel que manejan las comunicaciones seriales y que el driver puede utilizar para conversar con el dispositivo.




## El driver RoboScope.drv

La construcción del Roboscope.drv se realizó sobre la base del driver de ejemplo LX200.drv provisto por el fabricante del software SkyMap. La arquitectura modular que tendrá nuestro propio driver es la siguiente:



La clase *Roboscope.cpp* será el corazón del driver *Roboscope.drv* que necesitamos para intermediar entre nuestro Roboscopio (hardware) y el software *SkyMap*, y será la encargada de implementar las funciones para un objeto de interfaz que define un sencillo protocolo de comunicación que transporta una

 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 8 de 20

breve interfaz de comandos. Conceptualmente podría esquematizarse el sistema de comunicación con una estructura en capas:

<b>COMMANDS</b> (aplicación)
<b>ASCII PROTOCOL</b> (transporte)
<b>RS232</b> (física)

Este protocolo responde a un esquema en el cual hay un master que interroga (el driver Roboscope.driv) y un slave que responde (el firmware Roboscopio.bin) .

El protocolo adopta un formato ASCII y se definen los siguientes elementos en su estructura:

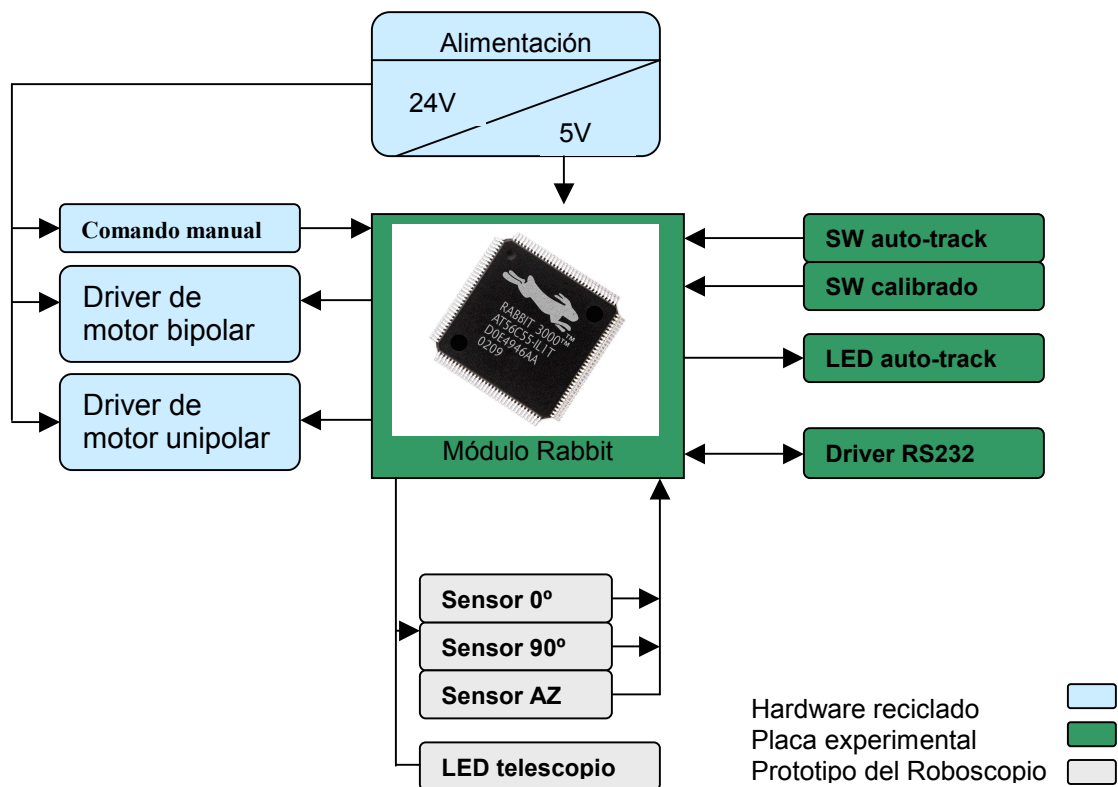
Función	Representación	Código ASCII
Tokens		
Separador uso grl.	:	0x3A
Separador fechas	/	0x2F
Fin de bloque	#	0x23
Fin de paquete	<b>ENTER</b>	0x0D + 0x0A
Respuestas		
Acknowledge	<b>ACK</b>	0x06
Not Acknowledg	<b>NAK</b>	0x15
Cancel	<b>CAN</b>	0x18
Comandos		
Set time	<b>H</b>	0x48
Set Location	<b>L</b>	0x4C
Ping	<b>P</b>	0x50
Get RA-Dec	<b>R</b>	0x52
Slew	<b>W</b>	0x57
Sync	<b>Y</b>	0x59
Get Alt-Az	<b>Z</b>	0x5A




## El hardware del Roboscopio

### *Vista global del sistema*

A continuación podemos observar una vista integral de todo el sistema de hardware. La parte de la electrónica de potencia que incluye las etapas de control de los motores paso a paso está aprovechada de una placa madre de una vieja impresora, como así también los motores y los engranajes utilizados en el modelo. La otra parte esta implementada corre por cuenta de la placa de prototipos de Rabbit y el centro neurálgico del proyecto está focalizado en un módulo Rabbit RCM3720 [RCM]. Por supuesto, si existiera una siguiente versión del prototipo, este reunirá en un solo PCB a toda la electrónica, o se hará de forma modular según análisis pertinente.



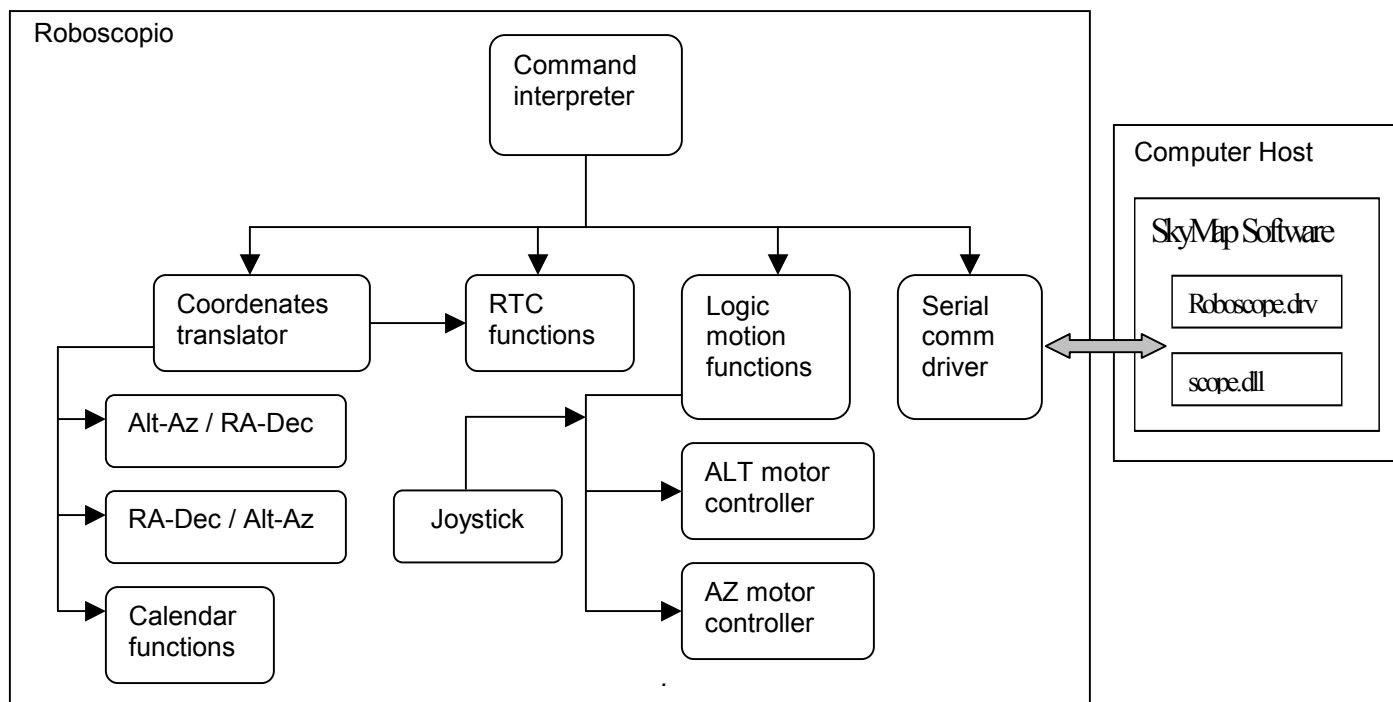
	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 10 de 20

## El firmware del Roboscopio

Dado que ya tenemos una descripción básica del sistema, podemos delinear el esquema general del sistema embebido que será construido mediante el entorno de desarrollo Dynamic C 9.62 [DYNC]:

- Es indispensable el control de dos motores, por lo cual escribimos un módulo encargado del control en bajo nivel de estos dispositivos.
- Manejamos un canal de comunicaciones serial para luego implementar una pequeña interfaz de comandos, el corazón de esto serán funciones de comunicación serial asíncrona provistas por el entorno de desarrollo y la cláusula switch del lenguaje C para construir la interfaz de comandos.
- Será fundamental el manejo de algoritmos de conversión para las coordenadas, funciones de calendario y trigonométricas, será conveniente disponer todas estas funcionalidades en un módulo a parte que pueda ser reutilizado.
- Manejamos tres sensores, el RTC, algunos LEDs y algunos switches, que responderán a funciones específicas que iremos definiendo.
- Incorporamos un Joystick implementado mediante un pequeño teclado matricial.


Cabe destacar que como era necesario compartir algunas constantes entre el firmware y el driver roboscope.driv del software SkyMap existirá un módulo destinado para este fin. A continuación vemos un esquema en bloques del firmware en su conjunto:



*Esquema del software embebido del Roboscopio*

### Nota:

La descripción de arriba corresponde a la versión inicial del proyecto. Al momento de escribir este informe se han incorporado varias mejoras y modificaciones en el firmware, la más notoria es la incorporación de la interfaz Web.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 11 de 20

## ***Estructura de archivos del proyecto***

Físicamente el proyecto de software del roboscopio está organizado según puede apreciarse en el siguiente esquema.

**Importante:** los archivos listados a continuación corresponden a la versión del proyecto vigente a la fecha de ser escrito el presente informe, la cual difiere de la versión inicial del trabajo. Sin embargo las modificaciones fueron mayormente estructurales. Siendo el mayor cambio registrado la incorporación de la interfaz Web de monitoreo y control al sistema.

```

myProjects__Roboscopio_1.11
|__Roboscopio_1.11.c
|__Roboscopio_1.11.dcp
|__lib.dir
|__myLibs
|   |__altaz_radec.lib
|   |__calibrate.lib
|   |__command.lib
|   |__const.lib
|   |__iweb.lib
|   |__joystick.lib
|   |__juliandate.lib
|   |__keyboard.lib
|   |__motors.lib
|   |__sensors.lib
|__iWeb
|   |__index.zhtml
|   |__setup.zhtml
|   |__style.css
|   |__< ...
|   |__archivos de imagen >

```

## ***Los módulos del proyecto Roboscopio***

A continuación exponemos una reseña de las particularidades de cada uno de los módulos que se construyeron para este proyecto haciendo una distinguiéndolos según una categorización que los agrupa según su naturaleza.

### **Biblioteca de funciones para astronomía**

#### **Altaz-radec.lib**


Esta librería representa el corazón del proyecto Roboscopio. Es responsable de computar todas las operaciones relativas a los cálculos astronómicos que nos permitirán luego establecer los valores correctos para movilizar los motores en forma adecuada y precisa.

En el código fuente de este módulo se cristalizan los conceptos teóricos investigados y estudiados que figuran en las páginas precedentes.

En base a esta librería se especializa el proyecto y se aplica la robótica a la astronomía.

En ella se exponen las estructuras definidas para las operaciones de las funciones de conversión de coordenadas que provee el módulo.

Por ejemplo, dentro de esta library se decide si un objeto que quiere apuntarse se encuentra visible o por el contrario si la coordenada RA-Dec corresponde a un objeto que está bajo el horizonte.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 12 de 20

### **Juliandate.lib**

Contiene funciones de calendario que complementan las tareas que son desarrolladas por el módulo anterior. Precisamente este módulo proporciona el importante dato del Día Juliano actual, a partir del cual el módulo anterior calcula la Hora Sidérea Local, en el trabajo original se ha visto la importancia de estos datos y conceptos para los cálculos que involucran coordenadas de objetos celestes en tiempo real.

## **Biblioteca de funciones de dispositivos de hardware**

### **Motors.lib**

Implementan todas las funciones para el control en bajo nivel de los motores paso a paso, y se expone las funciones de alto nivel para ser utilizadas desde el programa principal de la forma que la aplicación lo requiera. Las funciones que el módulo provee, están divididas en dos grandes grupos, las funciones de control de motor unipolar y las del motor bipolar:

### **Keyboard.lib**

Es una pieza de software para el manejo de un teclado matricial en forma muy genérica. Concretamente se aplica al control de un Joystick para comando manual del telescopio. Hemos elegido hacerlo de esta manera pensando en versiones futuras que pudiera incorporar mayor cantidad de teclas.

### **sensors.lib**

Como su nombre lo indica, este módulo fue responsable de encapsular y exponer las constantes, macros y función de inicialización de los sensores utilizados en el modelo.

## **Biblioteca de funciones accesorias**

### **calibrate.lib**

Este módulo reúne varias funciones de calibración relacionadas con los motores paso a paso y algunas rutinas de inicialización del modelo, además incluye una rutina que desarrolla una secuencia de movimientos con fines demostrativos.


### **joystick.lib**

Fue creada en las recientes versiones con la finalidad de organizar el código, liberando al programa principal de funciones accesorias. Así fue que todas las funciones y variables de manejo del Joystick fueron agrupadas bajo este módulo.

## **Biblioteca de funciones para comunicación**

### **command.lib**

Este módulo encapsula todo el código que resuelve la interfaz de comandos, que además está implementándose sobre un sencillo protocolo ASCII definido para el proyecto, como se pudo apreciar en secciones precedentes.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 13 de 20

## Const.lib

Contiene las constantes utilizadas por el protocolo de comunicación que se emplea para transmitir datos entre el driver y el firmware, por lo tanto es accedido por sendos proyectos: Roboscope.CPP (driver C++) y Roboscopio.C (Dynamic C).

## Roboscopio\_v1.11, el programa

Podemos resumir el comportamiento del programa en una secuencia de acciones de inicialización seguidas por un bucle infinito, en el cual se atienden los diferentes estados en los que puede ingresar el programa. En otro orden de cosas, el programa también ejecuta un servidor http que despliega la interfaz Web de monitoreo y control del sistema.

Dado el alcance de este documento, no se realiza una descripción exhaustiva del programa ni del código fuente. En lugar de eso, se describirán las funciones fundamentales que dan sentido al Roboscopio, y que son implementadas con éxito por el programa:

### El comando SLEW

El comando SLEW es la función mas interesante que nos proporciona el software SkyMap. Se trata de recibir un mensaje por el puerto serie, parsear su contenido, reconocer el comando SLEW, mediante el cual se obtienen la coordenadas RA (ascensión recta) y Dec (declinación) las cuales se almacenan en una estructura de datos temporal (newRaDec). Posteriormente, en forma asíncrona, SkyMap envía el comando GO, que desencadena una secuencia de acciones que incluyen las importantes funciones de conversión de coordenadas que lleva a posicionar el telescopio en el objeto indicado por el software. Para finalizar, se actualiza la posición curRADec, y el software SkyMap recibe la nueva posición del telescopio, con la cual actualizará el cursor sobre su mapa celeste.

### La función Auto-Track

No obstante la importancia de esta funcionalidad, esta resultó construida prácticamente sin esfuerzo adicional, ya que surgió naturalmente de la lógica de los módulos de conversión de coordenadas y de movimiento de los motores.

La característica añadida por esta función es la de ejecutar periódicamente el calculo de la coordenada local para el último objeto apuntado, tras lo cual se actualiza la posición consiguiendose fácilmente el seguimiento del objeto.

## La Interfaz Web

En las últimas mejoras realizadas sobre el Roboscopio se incorporó conectividad de red y tecnología web. El dispositivo es capaz de implementar el stack TCP/IP sobre una interfaz Ethernet y de correr un servidor web, que es capaz de ejecutar scripts del lado del servidor (tecnología "RabbitWeb" [RWEB]). Se presenta una interfaz web de dos niveles, de los cuales, uno de ellos está protegido por contraseña. La conexión con Skymap continúa realizándose a través de puerto serie, pero esta interfaz web proporciona un estadio inicial a partir del cual se podrá reemplazar en forma total el vínculo serial punto a punto por el vínculo de red vía Ethernet.

### Página de monitoreo

La interfaz principal que presenta el Roboscopio es de monitoreo. Mediante aquella es posible visualizar en tiempo real todos los parámetros necesarios para el funcionamiento del dispositivo y para la información del usuario, a saber:

- Coordenadas geográficas del lugar:

- Latitud y longitud en grados.
- Coordenadas ecuatoriales del objeto apuntado:
  - Ascensión recta y declinación en horas.
- Hora local en horas.
- Hora sidérea en horas.
- Coordenadas horizontales del objeto apuntado:
  - Altitud y azimut en grados.
- Modo de funcionamiento:
  - Autotracking, Fast-Autotracking o Normal.

Esta interfaz se actualiza automáticamente cada aproximadamente 3 segundos, solicitando los nuevos datos al dispositivo. Puede verse un screenshot de la página que despliega el dispositivo (*figura iweb-m*)

### Página de comando manual

Desde la página principal podemos acceder a la página de control manual a través de un link presente en el menú izquierdo.

En esta página el Roboscopio ofrece la posibilidad de controlar manualmente sus movimientos.

Existen dos formas de hacerlo: mediante la virtualización del Joystick, o bien mediante la especificación directa de una nueva coordenada RA-Dec.

Una vez ejecutado el comando, cualquiera sea este, el Roboscopio intentará moverse según lo indicado y reportará los resultados al operador.

Cabe destacar que cuando se utilice el Joystick virtual, se debe indicar al Roboscopio la finalización de la operatoria manual mediante la tecla identificada con la letra 'E' (END), de lo contrario, el Roboscopio se moverá, pero no actualizará sus datos de posición, tal como sucede en la operatoria manual por hardware. Puede verse un screenshot de la página que despliega el dispositivo (*figura iweb-c*)



Parámetro	Valor	Unidad
RA	154.3507	grados
Dec	25.3111	grados
Altitud	30.0000	grados
Azimut	358.3403	grados
Resolucion vertical	0.5294	grados
Resolucion horizontal	0.1901	grados
Julian Day	2454680.22381	Días
Sidereal Time	10:11:2.67	hh:mm:ss.ss
Fecha local	01/08/2008	dd/mm/aaaa
Hora local	14:22:18	hh:mm:ss
Latitud	-34.6700	grados
Longitud	-58.5000	grados

*Figura iweb-m*

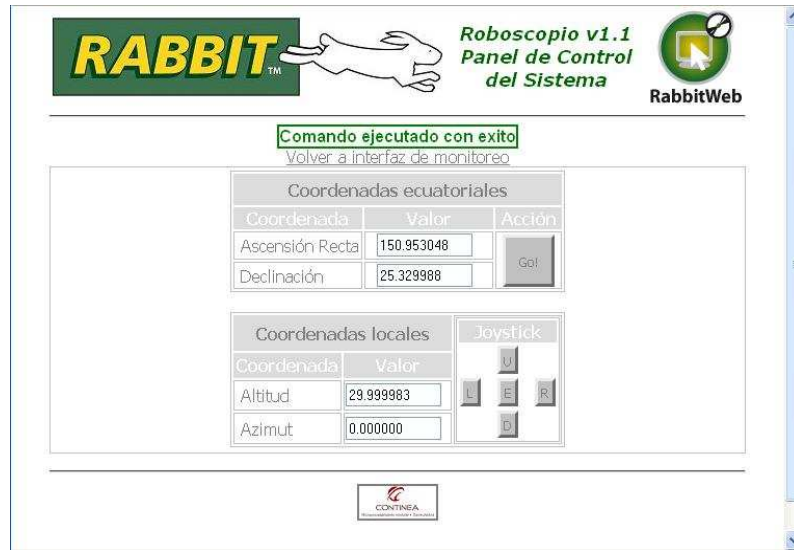









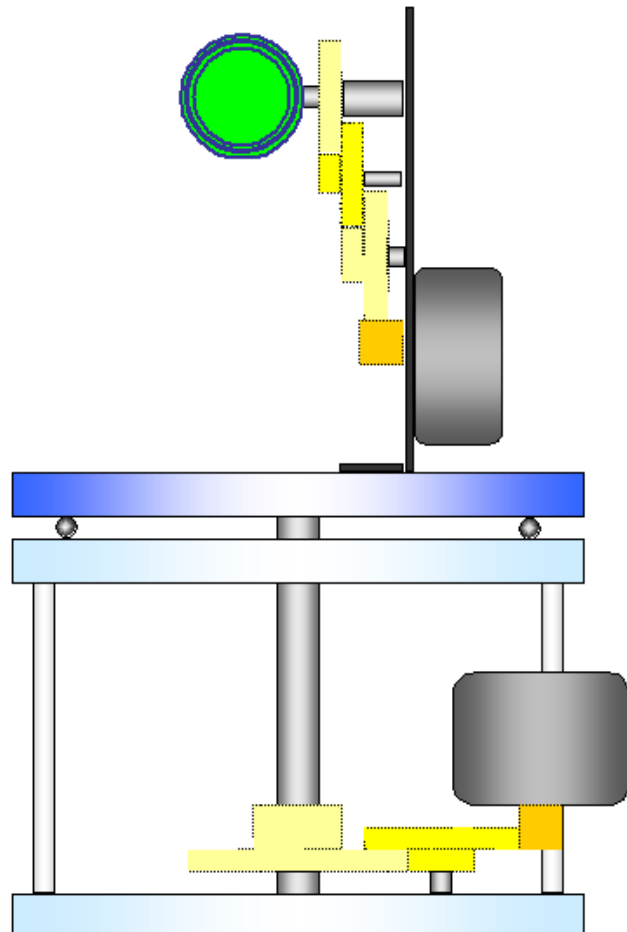


Figura iweb-c

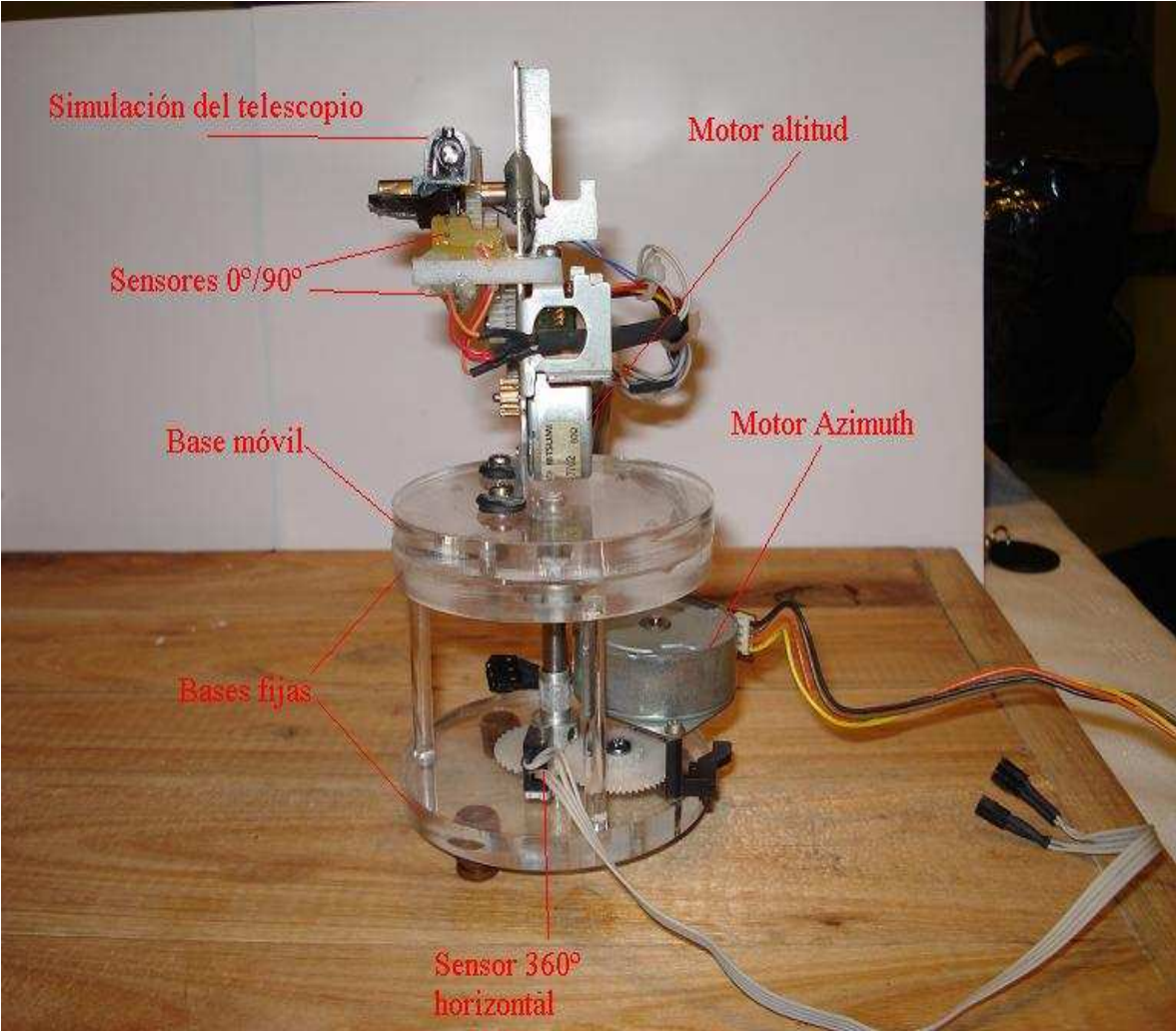
## Esquema del modelo mecánico a construir

No menos importante que cualquiera de los componentes descritos es el cuerpo estructural que dará soporte a todo el conjunto de partes utilizadas. Dicha estructura estará básicamente conformada en acrílico, y constará de tres bases circulares apiladas, separadas convenientemente para alojar las diferentes partes funcionales. Una vista de perfil de la estructura puede apreciarse en el esquema siguiente:


- Motores 
- Engranajes 
- Piñón 
- Rodamiento 
- Eje principal 
- Bases fijas 
- Base giratoria 
- Barras soporte 
- Torre vertical "Telescopio" 



## Primer modelo construido





	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 17 de 20

## Conclusiones

En función de las aspiraciones con las que hemos planteado el proyecto, se ha cumplido razonablemente con las expectativas ya que se ha logrado obtener la totalidad de las funcionalidades buscadas. Cerramos estos capítulos con un puntual análisis de la evolución futura del proyecto y finalmente agregamos algunas palabras a como de conclusión final del trabajo realizado.

### ***Evolución en tres ejes de optimización***

Si bien las mejoras que pueden realizarse sobre el proyecto son muchas y muy variadas existen tres ejes fundamentales que en torno a los cuales se articulan las optimizaciones furas que podría sufrir en su conjunto el modelo construido.

#### **Mecánica: Trenes de engranajes.**

Durante el desarrollo y pruebas realizadas notamos, que la pérdida mayor de eficiencia de todo el sistema, y esto se traduce inmediatamente en pérdida de precisión, está relacionado con los engranajes utilizados, y por supuesto con la resolución angular de los motores.

Para este proyecto, utilizamos motores de 7.5° y engranajes plásticos que nos daban resoluciones cercanas a los 12" y 30" de grado aprox. para azimut y altitud respectivamente.

Si la relación de engranajes fuera más conveniente, y los engranajes fueran de bronce o alguna combinación de materiales mas eficiente, o si se empleara una combinación de tornillo sinfín y corona, los resultados podría ser muy superiores a los obtenidos en el proyecto, y por otro lado, los movimientos sería menos bruscos que los logrados con la combinación actual de engranajes.

Otro aspecto a tener en cuenta, sería el reemplazo de los motores paso a paso, por servo-motores, motores de corriente continua exitados por PWM y gobernados por un sistema de posicionamiento de lazo cerrado, lo cual también ofrecería un comportamiento de orden superior.

#### **Electrónica: Sensores magnéticos, auto-alineación.**


La intención inicial al comenzar con este proyecto era la de dotar al Roboscopio con la capacidad de detectar por sí mismo el norte magnético.

Para ello, identificamos una serie de módulos de la firma Honeywell. Concretamente nos atrajo el modelo HMC6352 [HMC], por poseer la funcionalidad de brújula electrónica en una solución modular, con comunicación I2C, que provee una resolución angular de 0.5° con una precisión de +/-2.5°.

Dados los tiempos acotados para la finalización del proyecto se resolvió prescindir de este dispositivo, dejándolo para una futura evolución del modelo del Roboscopio.

#### **Software embebido: Conectividad TCP/IP, Roboscopio on-line.**

En cuanto al software, ya se ha implementado una mejora importante en una version posterior a la inicial, esto es, la de incorporar la interfaz Web, sin embargo todavía resta conectar el dispositivo con el software SkyMap mediante TCP/IP modificando el driver de comunicaciones Roboscope.driv, lo cual evitaría la conexión serial, y daría muchísima flexibilidad en las comunicaciones Sklymap - Roboscopio.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 18 de 20

## ***Limitaciones detectadas***

No podemos dejar de mencionar los inconvenientes, limitaciones y/o problemas de eficiencia advertidos en el modelo:

- ❑ Un punto fundamental a mejorar en este primer modelo en particular, está relacionado con la disposición de los cables. Obviamente la base es giratoria y lleva consigo los cables del motor de movimiento vertical, y los sensores, y estos pueden enroscarse y atascarse fácilmente en el mecanismo. Para contrarrestar este problema realizamos un algoritmo que evita que la base gire mas de 180° en el plano horizontal, buscando siempre el camino mas corto, de esta manera los cables nunca se llegan a enroscar. De todas formas, el esquema de cableado no es satisfactorio.
- ❑ Otro aspecto a mejorar tal como se ha dicho en la sección anterior, es la resolución angular vertical, al menos en lo particular de este primer modelo. Y en general, se debe estudiar la mejora sustancial de los trenes de engranajes y la elección de motores mas eficientes. Inferimos que los movimientos bruscos son provocados en parte, por la mala calidad del acoplamiento de los engranajes y por la baja resolución, ya que al comparar, ya que si el movimiento fuera mas fluido


## ***Consideraciones Finales***

Antes de concluir, queremos resaltar los logros de este proyecto:

- ❑ Conocimos un software profesional de astronomía y aprendimos a utilizarlo.
- ❑ Aprendimos sobre las posibilidades que brinda la interacción entre telescopios robotizados y el software astronómico.
- ❑ Aprendimos los fundamentos matemáticos que rigen la utilización de los diferentes sistemas de coordenadas.
- ❑ Conocimos los diferentes tipos de telescopios y sus monturas.
- ❑ Realizamos un driver escrito en C++ para vincular al software y al dispositivo.
- ❑ Desarrollamos un software embebido que controla sensores, motores paso a paso, implementa cálculos astronómicos, implementa un protocolo de comunicación, maneja un puerto serie y un teclado matricial.

En particular, la mayor satisfacción reside en que el modelo desarrollado es perfectamente aplicable a la observación astronómica práctica, dado que los desafíos algorítmicos y matemáticos fueron superados, la única condición para llevar a la práctica y exponer en un ámbito profesional a un primer prototipo utilizable reside en la correcta construcción de los trenes de engranajes y de un cableado adecuado.

Hechos que ya estarían resueltos en algunos modelos de telescopios concebidos para ser motorizados, por lo tanto el desarrollo de esta primera versión que aquí está concluyendo quedará a la espera de un telescopio real al cual dotar de inteligencia.

	<b>Roboscopio V1.11</b>	Nota de Aplicación
		CoAN-009
	<b>Robotización de telescopio con RCM3720</b>	Publicado: 00/00/0000
		Página 19 de 20

## Referencias

### *El software de astronomía*

El fabricante proporciona en forma gratuita el código fuente listado a continuación que puede descargarse libremente desde su sitio Web:

**[SKY-DRV]**

DLL de control (header file e import library) "scope.h" y "scope.lib" son archivos requeridos para construir un nuevo driver de telescopio.

<http://www.skymap.com/files/drivers/source/scope.zip>

Driver de ejemplo (Telescopio Meade LX200 )

Complete source code (written for the Microsoft Visual C++5 compiler) for the SkyMap Pro Meade LX200 telescope driver. This is the source code used to build the "lx200.drv" driver supplied as standard with SkyMap Pro.

<http://www.skymap.com/files/drivers/source/lx200.zip>

**[SKY]**

Versión de demostración del software SkyMap Pro 11, que fuera utilizada durante todo el desarrollo del proyecto.

<http://www.skymap.com/>

Este proyecto ha sido realizado sobre la base del código mencionado para proporcionar la interacción entre el firmware que hemos desarrollado y el software SkyMap con la finalidad de robotizar un telescopio en forma experimental.

### *El hardware del Roboscopio*

**[MOTU]**

Hoja de datos de motor paso a paso unipolar.

\\Documentación\Hardware\Motores\M35SP-9.pdf

**[MOTB]**

Hoja de datos de motor paso a paso bipolar.

\\Documentación\Hardware\Motores\M42SP-7.pdf


**[DRVU]**

Hoja de datos del driver del motor paso a paso unipolar.

\\Documentación\Hardware\Drivers\LB1845.pdf

**[DRVB]** Hoja de datos del driver del motor paso a paso bipolar.

\\Documentación\Hardware\Drivers\STA471.pdf

 <b>CONTINEA</b> <small>Microprocesamiento modular + Conectividad</small>	<b>Roboscopio V1.11</b>	Nota de Aplicación
	<b>Robotización de telescopio con RCM3720</b>	CoAN-009
		Publicado: 00/00/0000
		Página 20 de 20

## ***El firmware del Roboscopio.***

### **[DYNC]**

Rabbit Semiconductors, Manual Dynamic C versión 9.62.

<http://www.rabbit.com/documentation/docs/manuals/DC/DCUserManual9/index.htm>

### **[RWEB]**

Rabbit Semiconductors, Manual TCP/IP Vol.II, Rabbit Web

Sección 4: HTTP Server

<http://www.rabbit.com/documentation/docs/manuals/TCPIP/UsersManualV2/http.html#1000626>

Sección 5: Rabbit Web

<http://www.rabbit.com/documentation/docs/manuals/TCPIP/UsersManualV2/rabbitweb.html#1002674>

## ***Evolución[...]Sensores magnéticos***

### **[HMC]**

Compás electrónico modular Honeywell modelo HMC6352.pdf

\Documentación\Hardware\Otros\HMC6352.pdf