

PROYECTO FINAL

Sistema de Automatización de Telescopio



Alumnos

Arluna, Pedro
Caruso, David
Rojas, Diego
Ruiz, Gonzalo
Vieytes, Sebastián

Tutores

Ing. Gerardo Battaglia
Ing. Jorge Cabrera

Docentes

Ing. Edgardo De Fina
Ing. Mario D'Indio

6 de octubre de 2013

Índice

1. Breve Descripción	5
2. Integrantes del Grupo	6
3. Descripción Detallada	7
3.1. Introducción al Sistema	7
3.2. HandSet	10
3.2.1. Criterio de Selección de la Unidad Central	10
3.2.2. Posición de los Objetos Celestes	11
3.2.3. Estimación del Campo Visible	12
3.2.4. Real Time Clock (RTC)	12
3.2.5. Teclado	13
3.2.6. Display LCD Gráfico	14
3.2.7. Memoria SD	15
3.2.8. Comunicación SPI	17
3.2.9. Comunicación con la PC	20
3.3. Motor Driver	22
3.3.1. Introducción	22
3.3.2. Criterio de Selección de la unidad Central	22
3.3.3. Análisis de recorrido y pasos necesarios	23
3.3.4. Control de Avance	25
3.4. Fuente de Alimentación	26
4. Memoria de Cálculos	27
4.1. Calculo de resolución mínima de cada eje	27
4.2. Cálculo de velocidad sidérea y de desplazamiento	28
4.3. Cálculo Memoria del Microcontrolador	30
5. Lista de Componentes	31
5.1. Lista de Componentes del HandSet	31
5.2. Lista de Componentes del Motor Driver	32
5.3. Lista de Componentes de la Fuente de Alimentación	33
5.4. Lista de Componentes del adaptador Serie MAX232	33
6. Planos eléctricos y mecánicos	34
6.1. Esquemático HandSet	34
6.2. Circuito impreso del HandSet	35
6.2.1. Placa Principal	35
6.2.2. Placa del Teclado	36
6.2.3. Placa adaptador Serie	36
6.3. Esquemático Motor Driver	37
6.4. Circuito Impreso del Motor Driver	38
6.5. Esquemático Fuente de Alimentación	39
6.6. Circuito Impreso de la Fuente de Alimentación	40

7. Métodos de ajuste y control	41
7.1. Ajuste y Control del HandSet	41
7.1.1. LCD Gráfico	41
7.1.2. Comunicación SPI y teclado	42
7.1.3. Real Time Clock (RTC)	42
7.1.4. Lectura de la SD	43
7.2. Circuito Motor Driver	45
7.2.1. Procedimiento de Control	45
7.2.2. Posible Fallas	45
7.3. Comunicación con la PC	46
7.3.1. Procedimiento de Control	46
7.3.2. Posibles Fallas	46
8. Métodos de fabricación y mantenimiento	47
8.1. Proceso de Fabricación de Placas	47
8.2. Gabinetes	47
8.3. Ensamble	47
8.4. Mantenimiento	47
9. Métodos de prueba y envejecimiento	48
9.1. Circuitos Impresos	48
9.2. Gabinetes	48
9.2.1. Comportamiento Mecánico	48
9.2.2. Envejecimiento acelerado	48
10. Estudio de garantías	49
10.1. Análisis de Garantías para el HandSet	50
10.2. Análisis de Garantías para el Motor Driver	52
10.3. Análisis de Garantías para la Fuente de Alimentación	53
10.4. Análisis de Garantías para Placa de Comunicación Serie, MAX232	54
10.5. Calculos de Confiabilidad	54
10.5.1. Confiabilidad del HandSet	54
10.5.2. Confiabilidad del Motor Driver	54
10.5.3. Confiabilidad de la Fuente de Alimentación:	55
10.5.4. Confiabilidad del Adaptador Serie:	55
10.5.5. Confiabilidad del Sistema	55
11. Legislación	56
12. Comercialización	57
12.1. Presentación e Introducción del producto desde un enfoque Comercial	57
12.2. Vida Util, uso del producto	58
12.2.1. Producto Básico	58
12.2.2. Producto Real	58
12.2.3. Producto Aumentado o Final	58
12.3. Aplicaciones o Mercado del Producto	58

12.4. Determinación del Consumo	58
12.5. Participacion de la Competencia	60
12.6. Determinacion del crecimiento del mercado	60
12.7. Plan de Ventas	61
12.7.1. Tipo de Mercado	61
12.7.2. Tipo de distribución	61
12.8. Organigrama de Comercialización	62
12.9. Estrategia de ventas, participación y políticas para ingresar al mercado	62
12.10Layout	63
12.10.1Proceso productivo	63
12.10.2Diagrama de Recorrido	64
13. Referencias	65
A. Características del Sistema Ecuatorial	66
B. Algoritmo Cálculo de Posición de Planetas	67
C. Estimación de campo visible	68
D. Tiempo Solar y Tiempo Sidéreo	72

1. Breve Descripción

El proyecto consiste en el desarrollo de un sistema para el posicionamiento, búsqueda y seguimiento de objetos celestes (estrellas, planetas y galaxias) para telescopios. Este sistema está diseñado para utilizarse particularmente sobre monturas del tipo ecuatorial y con la mecánica necesaria para su acoplamiento. El equipo diseñado presenta las siguientes características:

- Puesta en estación asistida (guía en el menú gráfico)
- Búsqueda de objetos celestes en forma autónoma
 - Planetas de nuestro sistema solar
 - Luna y el Sol
 - Mediante catálogos: Messier, IC, NGC almacenados en memoria SD
- Seguimiento automático (Tracking) de los objetos observados.
- Interacción de movimientos con software en PC (Compatible protocolo Meade Lx200)

2. Integrantes del Grupo

Foto	Apellido y Nombre	Mail
	Arluna , Pedro	pedroarluna@gmail.com
	Caruso , David	carusodvd@gmail.com
	Rojas , Diego	diego_rojas_utn@hotmail.com
	Ruiz , Gonzalo	gonzaloruiz88@gmail.com
	Vieytes , Sebastián	sv.1810@gmail.com

Tabla 1: Integrantes del grupo

3. Descripción Detallada

3.1. Introducción al Sistema

El sistema fue implementado y probado en un telescopio tipo newtoniano con una longitud focal de 1504mm y un diámetro de 300mm, elaborada su mecánica por el Ing. Gerardo Battaglia. Para poder cumplir con las tareas y funciones mencionadas anteriormente (ver: Breve Descripción), el sistema de automatización para telescopios presenta un diseño modular, lo cual permite una simple interpretación de las funciones desempeñadas por cada uno de los bloques que lo componen.

A modo de resumen, a continuación se presenta un diagrama general del sistema en la siguiente Figura Diagrama en Bloques.

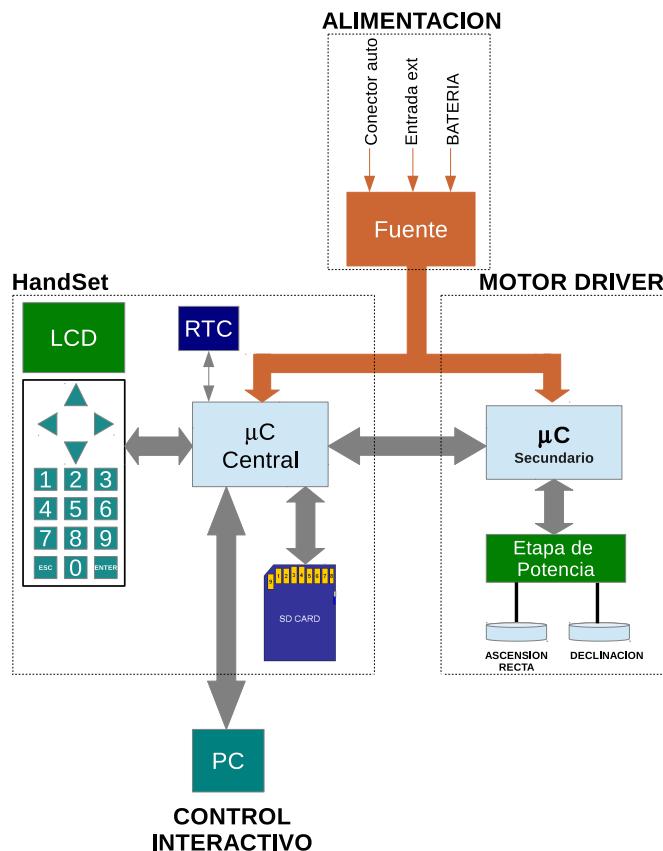


Figura 1: Diagrama en Bloques

En el sistema de automatización desarrollado, visualmente se pueden destacar dos partes, las cuales son: el mando de control (HandSet) y un gabinete que irá montado sobre la montura del telescopio, donde se encuentran la placa de control de motores (Motor Driver) y la Fuente de Alimentación. El Motor Driver se conecta directamente a la Fuente de Alimentación y el HandSet se alimenta a través de una conexión al primero. Por medio de dicha conexión también se establece la comunicación entre ambos.

A continuación se detallan los módulos principales que componen al proyecto y se enumeran las tareas más relevantes a realizar por cada uno de ellos.

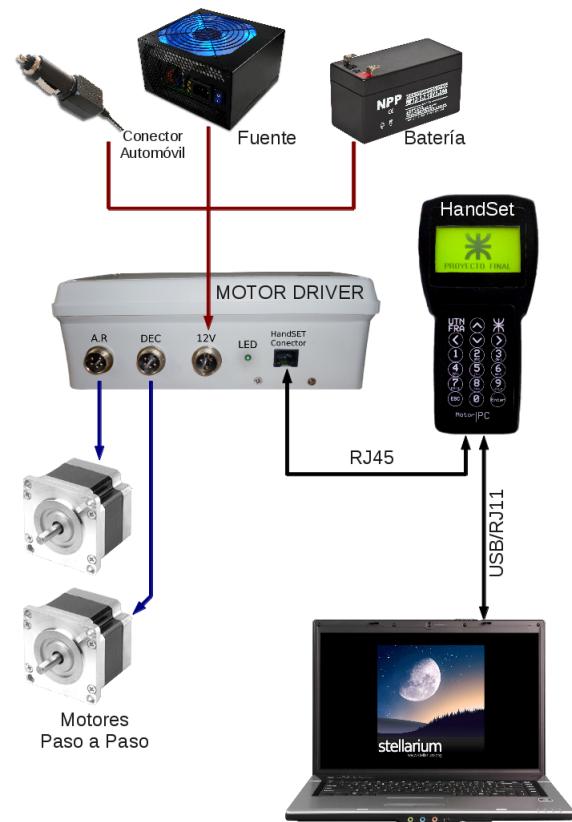


Figura 2: Esquema de conexión del Sistema

■ HandSet (Mando)

- Cálculo de posición de elementos del sistema Solar
- Control del RTC (Real Time Clock)
- Control del teclado matricial
- Comunicación con Motor Driver (Protocolo SPI)
- Visualización en lcd display gráfico.
- Lectura de catálogos de la memoria SD.

- Control interactivo con Software en PC (Compatible con Meade Lx200)
- Motor Driver (Controlador de Motores)
 - Traducción de coordenadas absolutas (posición) a relativas (Pasos del Motor).
 - Mantenimiento de la calibración del sistema.
 - Control de avance (cantidad de pasos, velocidad y torque) de los motores Paso a Paso.
 - Comunicación con placa del HandSet (Protocolo SPI).
- Fuente de Alimentación

Para una mayor comprensión sobre el funcionamiento del sistema de automatización, se iran explicando detalladamente las funciones desempeñadas por cada módulo.

3.2. HandSet

Este módulo es el que permite la interacción del usuario con el sistema, que se encarga de controlar y dar órdenes al resto del equipo.

Principalmente, éste se encuentra compuesto por el microcontrolador ATmega128, responsable de centralizar todas las funciones que ya se describieron del HandSet. En la Figura 3 se observa su diagrama en bloques.

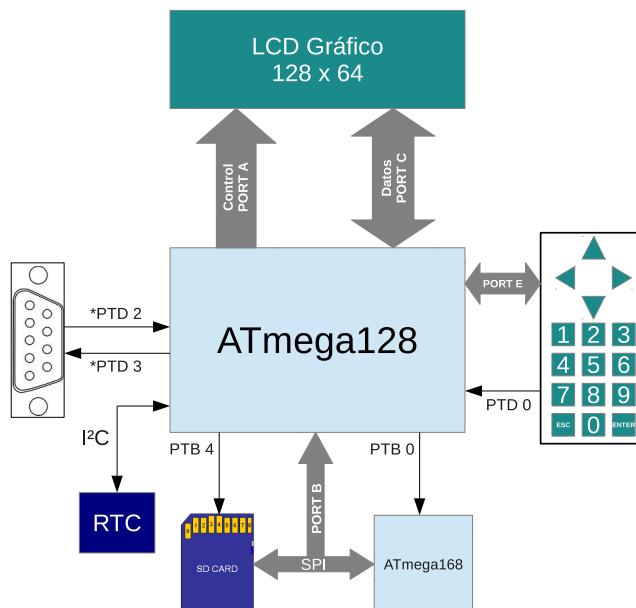


Figura 3: Diagrama en Bloques del HandSet

3.2.1. Criterio de Selección de la Unidad Central

Se sabía que las tareas a resolver por el HandSet eran variadas y complejas: desde la interacción con la PC hasta problemas de cálculo en punto flotante. Por lo que el microcontrolador que debía resolver las mismas debía tener:

- Compilador de C optimizado para cálculos en punto flotante
- Memoria RAM suficiente para los cálculos y anidamiento de funciones e interrupciones (2KB)
- Memoria ROM suficiente para el código (100KB)
- Alta disponibilidad de puertos
- Bajo Consumo

Considerando las características requeridas para solventar las necesidades del diseño en el HandSet, se determinó que el microcontrolador a utilizar debería ser de la familia de los ATmega de ATMEL. Dado que soportan un compilador de C Open Source basado en GCC denominado avr-gcc, empaquetado en AVR LibC [7]. También esta familia posee un programador USB Open Hardware denominado USBTiny [8] de bajo costo, el cual fabricamos y hemos utilizado para el desarrollo.

Luego el microcontrolador capaz de cubrir las necesidades presentadas es el ATmega128. El cual es un microcontrolador que posee:

- 128KB de Memoria Flash
- 4KB de Memoria RAM
- Interfaces SPI
- 2 x Timers de 8 bit
- 2 x Timers de 16 bit
- Frecuencia de operación hasta 16MHz
- Doble UART Programable
- Más de 50 pines programables de puerto de propósito general

3.2.2. Posición de los Objetos Celestes

Como se comentó anteriormente en Breve Descripción, el sistema por si sólo (sin necesidad de una PC) debe ser capaz de localizar la posición en coordenadas absolutas ecuatoriales de los siguientes elementos:

- Planetas del Sistema Solar
- Luna y el Sol
- De Catálogos: Messier, IC y NGC

Esta es una de las tareas más complejas e importantes que debe desarrollar el microcontrolador de esta unidad. Para comprender la complejidad de esta labor, a continuación se expresa una breve explicación sobre cómo se estiman las posiciones de los distintos elementos.

Todos los elementos que se encuentran a una distancia considerable del observador (La Tierra) son considerados como estáticos en el sistema de coordenadas ecuatoriales. Para determinar la posición de los mismos no se necesita ningún algoritmo y estos se pueden encontrar en los catálogos antes mencionados. Los catálogos son validos por 50 años aproximadamente.

Para el cálculo de la posición de los planetas en el espacio, se utilizó el método de Paul Schlyter [2]. Puede verse una breve explicación del mismo en el apéndice B

Para el cálculo de la posición de la Luna se utilizó el método de Jean Meeus [3]. En este se utilizan cientos de términos periódicos para determinar cada parámetro orbital. La precisión del cálculo depende de la cantidad de términos considerados.

El cálculo de la posición de la Luna presenta un inconveniente en comparación con el de los

planetas. En estos últimos, debido a que la distancia entre la Tierra y estos es mucho mayor que el radio de la Tierra, se podría suponer que el observador se encuentra en el centro de la misma y los cálculos se referían a ese punto sin inducir grandes errores.

En el cálculo de la Luna, su posición depende en una pequeña medida de la del observador sobre la superficie de la Tierra, efecto que se llama Paralaje (ver [4]), y que tiene mayor importancia cuanto más cercano esté el astro a la Tierra. En este caso, luego de obtener las coordenadas geocéntricas de la Luna, se tendrán que pasar a coordenadas topo-céntricas es decir, con el observador como centro de referencia teniendo en cuenta el radio terrestre.

3.2.3. Estimación del Campo Visible

Esta claro que el horizonte es uno de los grandes limitadores de visión del cielo nocturno, ya que en todo momento nos deja ver sólo la mitad de la esfera celeste. Por lo tanto, es necesario evitar que el telescopio alcance las coordenadas que están ocultas por el horizonte, ya que puede que el tubo se golpee contra el suelo en el intento de alcanzarlas. Por esta razón es necesario estimar cuales son las coordenadas ocultas por el horizonte e identificarlas antes de hacer movimiento alguno. El problema que surge es que la Tierra rota en todo momento, por lo que la media esfera visible se va desplazando y por otro lado, el cielo visible depende de la posición geográfica del observador. Por esta razón debió elaborarse un algoritmo que toma todos estos parámetros y estima si la coordenada solicitada pertenece al campo visible o no en el momento y lugar indicados. Este algoritmo es aplicado por el HandSet al momento que se le solicita una coordenada. También se utiliza para filtrar el listado de estrellas disponibles en el momento de la alineación. En el Apéndice: Estimación de campo visible, se resume como funciona el algoritmo.

3.2.4. Real Time Clock (RTC)

Es muy importante para el sistema de automatización contar, en todo momento, con una referencia de tiempo (segundos, minutos, horas, días, meses y años). Esta información es necesaria ya que es utilizada para coordinar el sistema relativo con el absoluto de coordenadas. La tarea de conteo de tiempo es llevada a cabo por el integrado DS1307, el cual tiene las siguientes características:

- Mantenimiento y conteo de Fecha y Hora.
- Programable vía protocolo I2C.
- Memoria RAM de 56 bytes para almacenar datos.
- Señal Cuadrada de salida programable.
- Capacidad de almacenar los datos por más de 10 años.

Inicialmente, cuando se comienza con la observación de objetos, se debe cargar al sistema las coordenadas de posición donde se encuentra el observador (latitud y longitud) así como también la fecha y hora. Estos últimos datos son el punto de inicio para el conteo del RTC.

Mediante la comunicación entre el micro-controlador ATmega128 y el RTC a través del protocolo I2C, se realiza la consulta de fecha y hora para completar los cálculos de posición.

Una función importante que lleva a cabo el RTC es la de guardar la información de la posición geográfica del telescopio. También, puede guardar si la ultima posición a la que se envió el telescopio es la de inicio o no. Esto es útil para el modo de Parking del telescopio, es decir, el modo de guardarlo en la posición de inicio (Polo Celeste) a fin de evitar el procedimiento de Alineación nuevamente.

Los datos almacenados en la RAM del RTC son:

- Huso Horario
- Latitud (Grados)
- Latitud (Minutos)
- Latitud (Hemisferio)
- Longitud (Grados)
- Longitud (Minutos)
- Longitud (Este/Oeste)
- Parking (1=Parking, 0=No Parking)

3.2.5. Teclado

Con el fin de lograr la interfaz con el usuario, el HandSet posee un teclado alfa-numérico para el ingreso de datos, junto con teclas direccionales, de validación (ENTER) y de cancelación (ESCAPE) para poder desplazarse a través de los distintos menús del sistema.

En la Figura 4 puede observarse la distribución de las teclas mencionadas.



Figura 4: Diseño del teclado del HandSet

Para cumplir con las características y tareas mencionadas, el teclado utilizado es del tipo matricial 4x4, en el cual la lectura de las teclas pulsadas se realiza mediante el método de barrido de filas y columnas. Todas las teclas tienen la misma prioridad, dado que todas actúan sobre

una única entrada de interrupción que se encarga de determinar cual fue pulsada.

En el manejo del teclado se distinguen los siguientes estados:

- IDLE (Reposo) Este estado se mantiene hasta que se pulse una tecla.
- DEBOUNCING (Anti-Rebote) Esto se logra mediante una demora de tiempo fija de 10mseg.
- SCANNING (Barrido) Esto sucede cuando se presionó por lo menos una tecla, y dentro del mismo se determina que tecla fue pulsada. Esto ocurre dentro de la interrupción.
- WAIT (Espera) Aquí se espera a que se suelte la tecla. Hasta que esto no suceda, no se abandona el vector de interrupción, lo cual garantiza la correcta lectura del próximo pulsado. La misma tiene la demora del Anti-rebote de 10mseg para evitar lecturas del rebote de la tecla.

3.2.6. Display LCD Gráfico

El display LCD es el medio por el cual el sistema se comunica con el usuario, ya sea para mostrar los distintos menús, para pedir el ingreso de datos o mostrar distintos mensajes. Los controles convencionales utilizan, en su gran mayoría, displays de 16x2, pero debido a la poca resolución que poseen, la información por pantalla es muy reducida por lo que en general quedan pantallas y menús confusos y poco entendibles por el usuario.

En nuestro caso, el display LCD utilizado, es un display gráfico (128x64) de la marca WINSTAR. El mismo sólo permite graficar de a 8 pixeles verticales por vez. Es decir, que para barrer los 128 x 64 pixels de la pantalla se tienen: 8 filas de 128 columnas de pixels. Dado que el display no cuenta con ningún dispositivo por Hardware que permita cargar la información fácilmente, se debieron implementar en el firmware del Atmega128 una serie de rutinas para facilitar su acceso. Las mismas se plantearon en 3 niveles distintos de operación y por medio del anidamiento (cada nivel superior, anida a uno inferior), se logró un código eficiente.

Nivel	Nombre	Función	Tarea
Bajo	lcd_wrdat	Escritura	Escribe una columna de 8 pixels
Bajo	lcd_rdat	Lectura	Lee una columna de 8 pixels
Bajo	lcd_wrcom	Escritura	Envía un comando al LCD
Medio	lcd_init	Control	Inicializa el LCD
Medio	lcd_setcursor	Control	Mueve el cursor a una posición
Medio	lcd_clr	Control	Borra el LCD
Medio	lcd_putchar	Escritura	Escribe un caracter
Alto	lcd_string	Escritura	Escribe una frase
Alto	lcd_bmp	Escritura	Dibuja una imagen

Tabla 2: Lista de Funciones implementadas para el LCD

Las imágenes mostradas en el menú se diseñaron con un editor de imágenes tipo BMP. Luego con una herramienta que convierte de BMP a una tabla en C se obtiene la imagen codificada (ver Referencia [9]). Luego las imágenes se componen en un archivo (*images.h*) mediante un script perl elaborado por nosotros.

Con el display gráfico empleado se logró un menú con mayor cantidad de opciones e información por pantalla además de animaciones simples que producen una mejor interacción con el usuario. El display con la pantalla de presentación como muestra, es el siguiente:



Figura 5: Pantalla de muestra del LCD Gráfico

3.2.7. Memoria SD

La función de la memoria SD es almacenar los catálogos IC, NGC y MESSIER, a través de los cuales se podrán visualizar un gran número de objetos, tales como estrellas, constelaciones, nebulosas, etc, que no necesitan de cálculos para su ubicación ya que sus coordenadas se mantienen fijas a lo largo de algunas décadas (50 años aproximadamente). Los catálogos mencionados son los de uso más frecuente en los telescopios comerciales, en los cuales se presenta la siguiente información:

NI	RH	RM	RS	V	DG	DM	DS
1	20	10	10	+	80	33	20
...
1001	33	12	02	-	72	44	21

Tabla 3: Formato de texto del archivo de catalogo

Donde:

- **NI:** Número o Nombre del objeto.

- **RH:** Horas Coordenada Ascensión Recta
- **RM:** Minutos Coordenada Ascensión Recta
- **RS:** Segundos Coordenada Ascensión Recta
- **V:** Signo (+/-) Coordenada Declinación
- **DG:** Grados Coordenada Declinación
- **DM:** Minutos Coordenada Declinación
- **DS:** Segundos Coordenada Declinación

Mediante el menú que se dispone en el HandSet podemos seleccionar el objeto que deseamos observar, almacenado en la memoria SD.

Una de las principales ventajas sobre el uso de memoria SD, es la practicidad para el usuario de poder modificar los archivos de catálogos a criterio y deseo de cada uno, a través de un editor de hojas de cálculo (Excel, OpenCalc, etc) o un editor de texto, ya que los catálogos se encuentran en un formato tipo **CSV** (*Comma Separated Value*). En este caso el separador utilizado es un espacio y los caracteres no poseen comillas, para reducir la información y simplificar la búsqueda. Además, el uso de esta memoria permite la flexibilidad de utilizar sin ningún problema otra con mayor capacidad o inclusive el uso de memorias Micro SD, que con la ayuda de un adaptador, puede ser leída sin problemas por el sistema.

La memoria SD utilizada se encuentra formateada en FAT 32, en ella se almacenan los catálogos de estrellas en archivos con extensión **.csv**.

Como adaptador SD a SPI se utilizó un kit comercial (ver [11]).

3.2.7.1. Lectura de la Memoria SD

Dada la gran cantidad de objetos que se encuentran en los catálogos, era necesario la utilización de un método que permitiera tener acceso a cualquier objeto de manera rápida, por lo que una búsqueda lineal no era adecuada, ya que en caso de buscar el último objeto de la lista debería recorrerse la misma desde principio a fin, pudiendo llegar a tardar varios minutos. Por ejemplo el archivo de IC tiene 5387 líneas, por lo que con el método lineal tardaría 5387 ciclos de lectura. Por esta razón se decidió realizar la búsqueda de objetos a través del método binario. En este método, el algoritmo va dividiendo la lista en dos hasta llegar al objeto buscado. Lógicamente este método requiere tener todos los elementos ordenados en forma creciente según el nombre asignado que puede ser alfabético o numérico.

Para comprender la secuencia en la que procede el algoritmo se plantea el siguiente ejemplo. Cualquiera sea el elemento solicitado, se comienza la búsqueda desde la mitad del archivo. Se inspecciona el elemento de esa posición y se analiza si el nombre del mismo es igual, mayor o menor al que está buscando. Si se cumple el primer caso, toma los datos de esa línea. En caso de ser la segunda condición se queda con la mitad superior del archivo y direcciona al elemento que está en la mitad de la misma. La ultima condición es similar a la anterior, pero toma la mitad inferior del archivo. Claramente el proceso continua hasta que la cantidad de elementos restantes es 1, o sea, indivisible por dos y ese elemento, el siguiente o el anterior (porque puede que la cantidad de elementos no sea potencia de dos) es el buscado.

De esta forma, con el método binario, el tiempo máximo de búsqueda (dado en ciclos de lectura) es de:

$$\text{round}(\log_2(N))$$

Donde **N** es la cantidad de elementos o líneas del archivo y el redondeo se hace siempre hacia el próximo entero. Para el caso del catálogo IC el tiempo máximo de lectura es de 13 ciclos de lectura.

3.2.8. Comunicación SPI

El HandSet controla el movimiento del telescopio desde un nivel superior, es decir, que sólo le indica las coordenadas a las que desea que este se mueva al Motor Driver y este (desde un nivel inferior) actúa sobre los motores. Esta comunicación entre los dispositivos mencionados se realiza mediante la interfaz serie SPI. Se eligió este tipo de interfaz dado que permite implementar protocolo full-duplex y ambos microcontroladores poseen por defecto hardware dedicado para esta. De ahora en más se usarán términos como: maestro y esclavo. El primero se aplica al microcontrolador del HandSet (ATmega128) y el segundo al del Motor Driver (ATmega168).

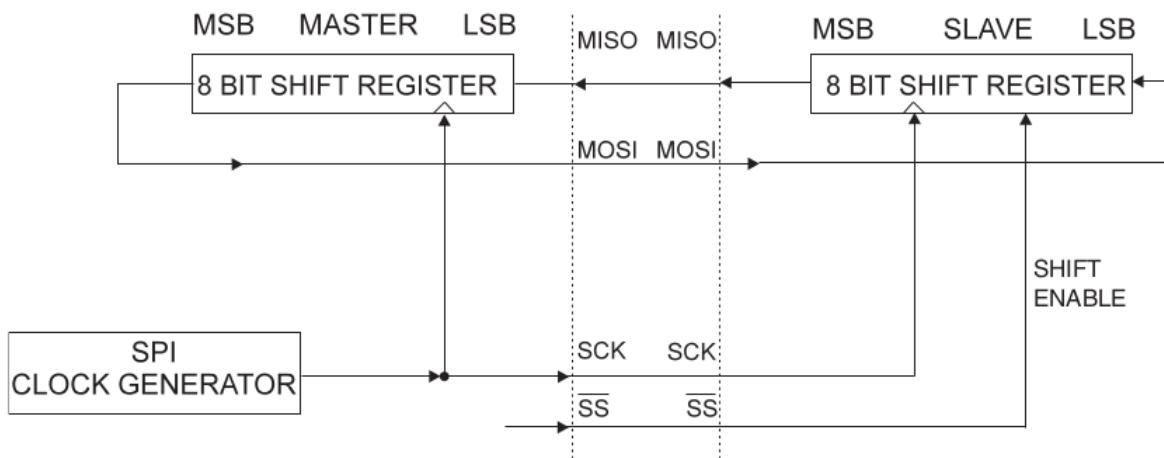


Figura 6: Diagrama de Interconexión SPI Maestro-Esclavo

La comunicación se rige por medio de 4 líneas (o hilos):

- **MOSI:** Master Output Slave Input. Datos enviados por el maestro al esclavo
- **MISO:** Master Input Slave Output. Datos enviados por el esclavo al maestro
- **SS:** Slave Select. Usado para seleccionar un esclavo en particular, cuando se tiene un sistema de múltiples esclavos.
- **SCK:** Clock Signal. Señal de Clock, controlada por el maestro.

En la Figura 6 se ve como el mecanismo consiste en un registro de desplazamiento tanto en el maestro como en el esclavo. Cuando el maestro quiere transmitir, pone un comando o dato en su registro interno y lo envía en forma serial por el pin **MOSI**. El esclavo, se sincroniza a la

señal **SCK**, toma los datos por **MOSI** y transmite lo que hay en su buffer interno por **MISO**. Ambos procesadores transmiten simultáneamente por lo que el protocolo es full duplex. La velocidad de comunicación está establecida por el maestro.

En el HandSet el bus SPI es compartido para lectura de la memoria SD. Es decir, que ambas comunicaciones comparten las señales **MOSI**, **MISO** y **SCK** y cada una tiene su propio **SS**, de forma que el maestro (ATmega128) habilita uno u otro según se requiera.

3.2.8.1. Protocolo de Comunicación

Básicamente la trama de comunicación se divide en tres partes que siempre tienen el mismo orden:

- Comando de inicio: marca el inicio de trama
- Datos: conjunto de información intercambiada
- Comando de finalización: cierra la comunicación e indica fallas

Aprovechando la comunicación full-duplex, por cada dato enviado se realiza un chequeo de la información transmitida hasta el momento (Acknowledge, **ACK**). Este **ACK** enviado consiste en retransmitir lo que acaba de llegar para que el que emitió el mensaje verifique la información. De esta forma el microprocesador transmisor tendrá dos campos que pertenecen a lo que envió y a lo que recibió y en caso de ser iguales indica que lo enviado es correcto. Si por el contrario difieren, significa que hubo un error en la comunicación, la trama entera se descarta y se vuelve a transmitir. También, los comandos de inicio de trama, tienen información sobre la cantidad de datos que deben enviarse, por lo que si esto no se cumple, se identifica como otra condición de error.

Si cualquiera de estas situaciones ocurre 3 veces consecutivas sobre la misma trama, el sistema interpreta que ocurrió un problema en la conexión, como que el cable está desconectado o roto, y escribe un mensaje de error en pantalla.

El comando de finalización sirve como indicador para saber si la trama que se ha enviado contiene errores. El mismo puede tener dos valores posibles: 0xEE (trama OK) o 0xFF (trama con error).

3.2.8.2. Estructura del Comando de inicio

El protocolo básicamente se definió para operar, desde el maestro, en dos modos: indicar una acción al esclavo o leer un parámetro del mismo. A través del comando de inicio de trama se establece si la trama corresponde a alguna de las acciones mencionadas. Por lo que existen dos clases de comandos de inicio de trama para un modo u otro.

Estos comandos que inician la trama se componen de un BYTE y en la tabla 4 se resumen el orden de los bits que lo componen.

- **S/R**: Set / Read (1/0). Define si el maestro intenta setear o leer algo en el esclavo.
- **A[2-0]**: Dirección del comando. Este campo define una distancia entre los distintos comandos, para identificarlos.

Campo de Bits							
D7	D6	D5	D4	D3	D2	D1	D0
S/R	A2	A1	A0	C	S2	S1	S0

Tabla 4: Campos del comando de inicio

- **C:** Parámetros variantes del comando. Permite incorporar información adicional que modifique la interpretación de los datos subsiguientes por parte del esclavo.
- **S[2-0]:** Tamaño de la trama de datos del comando (Cantidad de Datos - 1). Véase que como máximo la trama puede tener 9 bytes (Comando de inicio+ 8 datos + Comando de finalización)

3.2.8.3. Comandos de Seteo

En la Figura 7 puede verse como resulta la secuencia de la trama para los comandos de seteo. En dicha figura, se observan las líneas SPI del maestro (**MOSI**) y las del esclavo (**MISO**), especificando qué dato envían en cada ciclo de la comunicación. Puede observarse como el esclavo envía la información (**ACK**) para que el maestro la verifique. El signo **X** indica que es un dato que se descarta, esto ocurre porque el esclavo no sabe en qué momento el maestro enviará un comando y lo que tiene en su buffer no tiene nada que ver con la comunicación a establecerse.

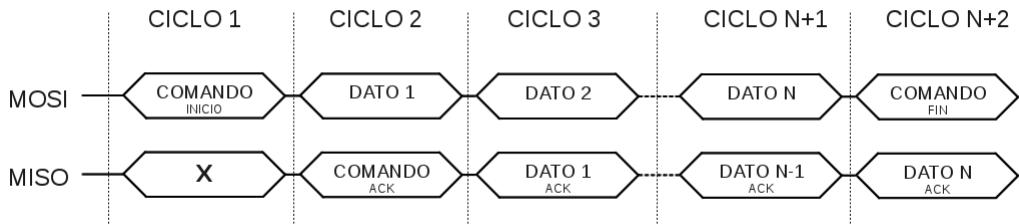


Figura 7: Secuencia para Comandos de Seteo

En la tabla 5 se resume la lista de comandos definidos para operar con el esclavo.

Nombre	Bytes	Función
SMOVMOV	4	Mueve al motor un n° de pasos indefinidos
SPOSORG	8	Establece la posición de origen
SGOTO	8	Envía al telescopio a una coordenada

Tabla 5: Comandos de Seteo por SPI

3.2.8.4. Comandos de Lectura

Para el caso de los comandos que pretenden extraer información del esclavo, la secuencia es similar, pero hay detalles a tener en cuenta. Como el maestro es el que lleva adelante la comunicación y él mismo es quien debe enviar los **ACK** en esta oportunidad, es necesario un ciclo

más que en el caso de los de seteo. Véase la Figura 8 para comprender esta cuestión.

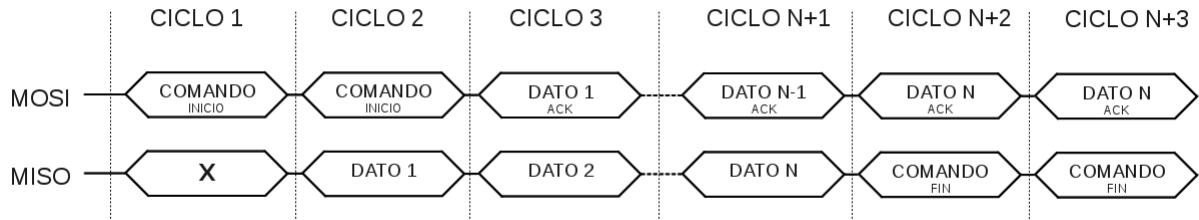


Figura 8: Secuencia para Comandos de Lectura

En la Tabla 6 se muestran los comandos que utiliza el dispositivo maestro para recibir parámetros del esclavo, detallando el nombre del comando, el código, cantidad de bytes enviados en la trama y la función.

Nombre	Bytes	Función
RMOTST	5	Lee el estado de los motores
RPOS	9	Lee la posición actual del telescopio

Tabla 6: Comandos de Lectura por SPI

3.2.9. Comunicación con la PC

Para realizar un control interactivo desde la PC, se utilizó un adaptador USB/232 comercial y un MAX232. De manera que el microcontrolador maneja una interfaz serie mediante una UART a 9600 baudios 8N1 por hardware y la PC recibe los comandos por USB.

El protocolo de comunicación utilizado es el de un telescopio comercial (Meade LX200 [6]). Por lo que la interactividad entre la PC y el telescopio puede resolverse mediante el uso de cualquier software que soporte este protocolo. De manera que se obtienen otras variantes de uso del equipo.

El protocolo es simple, la PC envía comandos para realizar acciones sobre el telescopio y este responde a ellos. Es muy similar al protocolo SPI implementado en nuestro sistema.

La comunicación es half-duplex, es decir, que luego de cada comando enviado por la PC (la cual inicia toda comunicación) prosigue una respuesta por parte del HandSet. Todos los comandos comienzan con el carácter ASCII ':'. Luego lo prosiguen una o dos letras que definen el nombre del comando y la trama finaliza con el carácter ASCII '#'.

En la Tabla 7 se muestra un resumen de los comandos implementados.

Comando	Retorno	Función
:GR#	HH:MM:SS#	Pide la posición actual de AR
:GD#	sDD β MM:SS#	Pide la posición actual de DEC
:SrHH:MM:SS#	R	Setea la posición de AR del objeto a ver
:SdsDD:MM:SS#	R	Setea la posición de DEC del objeto a ver
:MS#	Nada	Mueve el telescopio a la coordenada segun 'Sr' y 'Sd'
:Q#	Nada	Detiene el movimiento del telescopio

Tabla 7: Comandos del protocolo Meade LX200

Donde:

- HH: Horas (BCD ASCII)
- MM: Minutos (BCD ASCII)
- SS: Segundos (BCD ASCII)
- DD: Grados (BCD ASCII)
- s: signo de declinación ('+' o '-')
- R: Respuesta del HandSet ('1'=objeto en campo visible, '0'=objeto fuera de campo visible)
- β : Separador de grados. Utiliza este ASCII en vez de 'o'

El software que se seleccionó para realizar las pruebas fue el Stellarium [5], ya que presenta una de las interfaces más completas y con mucho realismo. Además es Open Source, permite introducir variaciones, crear distintos pluggins y cuenta con mucho soporte online.

3.3. Motor Driver

3.3.1. Introducción

El controlador de los motores se encuentra en otro de los módulos que integran el sistema. Dicho controlador, junto con la fuente de alimentación, se encuentra alojado en un gabinete sobre la montura del telescopio y no tiene ninguna interacción directa con el usuario.

El microcontrolador de esta unidad es el encargado de traducir las coordenadas ecuatoriales absolutas, enviadas desde el HandSet, a coordenadas relativas mediante las relaciones mecánicas, transformándolas en pasos de motor. Por lo tanto, esto permite que con un único firmware en el HandSet se puedan manejar distintas configuraciones mecánicas (distintas relaciones de engranajes, coronas, tornillos sin fin o tipos de monturas) ya que sólo habrá que adaptar el firmware del controlador de esta unidad. De esta forma, el firmware del HandSet es independiente de la mecánica a la cual debe manejar, obteniendo así una gran versatilidad del sistema a distintas configuraciones mecánicas. Además, se reparte el trabajo de control entre 2 microcontroladores y se separa del ruido de potencia al HandSet.

También el controlador de esta unidad debe controlar el avance de los pasos del motor, llevando un conteo de los pasos realizados, realizando a la vez, un control de velocidad y torque sobre los mismos y manteniendo el sincronismo con el sistema ecuatorial celeste. Sobre esto también se buscaba que el control de cada eje sea independiente, es decir, que se pudieran controlar los avances de los ejes al mismo tiempo, de forma de reducir los tiempos de búsqueda.

En cuanto al algoritmo de control desarrollado para este módulo se encuentra planteado por niveles. Primero se analiza como se realizará el recorrido del telescopio evitando los puntos ciegos (ver Análisis de recorrido y pasos necesarios), luego se configura como se realizará el avance y mediante interrupciones se controla el movimiento (ver Control de Avance).

Otra de las tareas realizadas por este microcontrolador, es la de mantener la comunicación vía SPI, explicada en 3.2.8.

3.3.2. Criterio de Selección de la unidad Central

Para realizar esta labor se sabía que no se necesitaba un microcontrolador tan complejo como el utilizado en el HandSet. Un requisito importante era que debía pertenecer a la misma familia (ATmega), para facilitar su uso (mismo entorno de programación y programador) y sobre todo el mecanismo de comunicación por SPI. A continuación se detalla la lista de necesidades para el controlador de esta unidad:

- Perteneciente a la familia de los ATmega
- Memoria ROM suficiente para el código (10KB)
- Memoria RAM suficiente para cálculos y anidamiento de funciones e interrupciones (500B)
- 1 Timers para el control de torque.
- 2 Timers idénticos para el control de velocidad y avance de los motores en forma independiente.
- Interfaz SPI para comunicación

- Disponibilidad de puertos para control de motores (10) y para comunicación SPI (4).

Considerando los puntos planteados, se llegó a la conclusión de utilizar el microcontrolador ATmega168, que posee las siguientes características:

- 16KB de Memoria Flash
- 1KB de Memoria RAM
- Interfaces SPI
- 2 x Timers de 8 bit
- 1 x Timer de 16 bit
- Frecuencia de operación hasta 20MHz
- Más de 20 pines programables de puerto de propósito general

3.3.3. Análisis de recorrido y pasos necesarios

Aquí se explica el nivel superior del algoritmo, el mismo opera a nivel de coordenadas y solo determina el recorrido que ha de realizar el telescopio.

El Motor Driver, recibe por SPI la coordenada que el HandSet le solicita alcanzar, la cual pertenece al campo visible estimado por la unidad antes mencionada. Luego el controlador de motores debe determinar cuál es el recorrido óptimo para llegar a la coordenada solicitada. El algoritmo, siempre que pueda, selecciona el recorrido de menor cantidad de pasos, es decir, el que recorre menor distancia para alcanzarlo. Esto se calcula simplemente con una diferencia de posiciones:

$$C_{AR} - P_{AR} = d_{AR}$$

$$C_{DEC} - P_{DEC} = d_{DEC}$$

Donde:

- **C**: Coordenada a alcanzar
- **P**: Posición actual del telescopio
- **d**: Diferencia proporcional a la cantidad de pasos a realizar.
- Término AR: se refiere al eje de Ascensión recta
- Término DEC: se refiere al eje de Declinación

En la Figura 9 puede observarse cómo es este movimiento recorriendo el mínimo espacio. Si bien la figura muestra primero la coordenada Ascensión Recta y luego en Declinación, en la realidad es una composición de ambos movimientos al mismo tiempo.

También en dicha figura puede observarse la posición del Polo Sur Celeste (PS) el cual es el punto de referencia inicial para el telescopio. El mismo tiene coordenada -90° de Declinación, que es el valor mínimo de este eje.

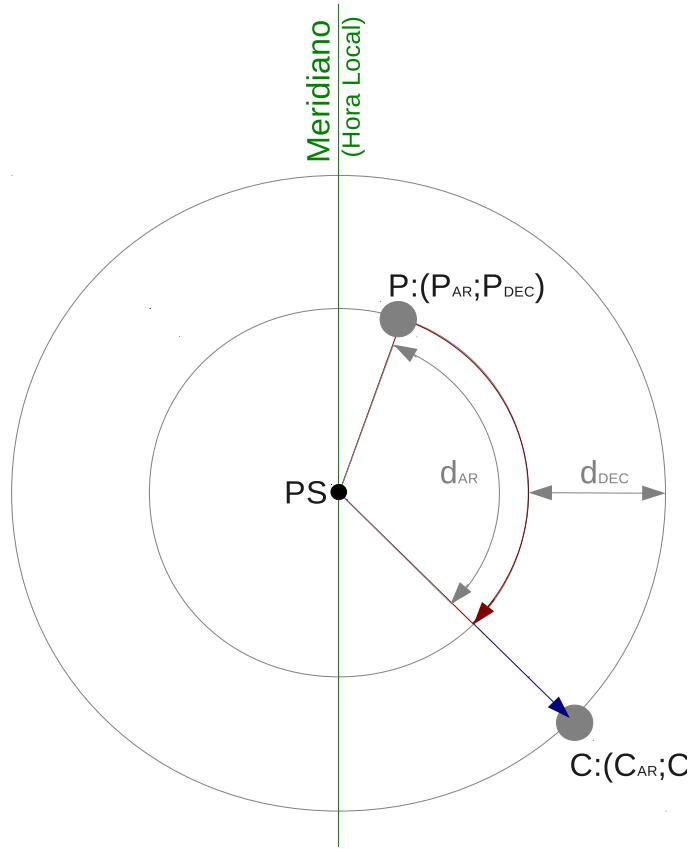


Figura 9: Recorrido del telescopio por mínima distancia

Las circunferencias concéntricas representan las curvas de nivel de Declinación constante. Los radios de esas curvas concéntricas son las curvas de nivel de los meridianos o Ascensiones Rectas constantes.

Este movimiento no puede realizarse siempre, dado que existe un rango ciego en el telescopio debido a la montura. La misma limita el recorrido del eje de Ascensión Recta y genera una zona que es imposible de acceder si no se cambia la estrategia.

En los casos en los que el telescopio deba pasar por esta zona ciega, se cambia la estrategia pasando por un camino alternativo. En la Figura 10, puede observarse un ejemplo al respecto. Donde el MODO 1 muestra la estrategia que evita la zona ciega y el MODO 2 recorre por mínima distancia quedando más cerca de la zona mencionada.

Puede verse que para evitar la zona ciega el telescopio deberá pasar por el polo celeste y así cambiar la estrategia. Esto obliga al eje de declinación a recorrer una distancia mayor a la calculada por el método anterior. También véase que el sentido de movimiento del eje de Ascensión Recta se ha invertido.

Para evitar los puntos ciegos y establecer un mecanismo automático de cambio de estrategia, se estableció un punto límite: el meridiano local (en la Figura 10 línea de 12hs). Es decir, que el Motor Driver estima si el movimiento que se está solicitando cruza el meridiano local y decide si es necesario cambiar la estrategia de movimiento por la mencionada.

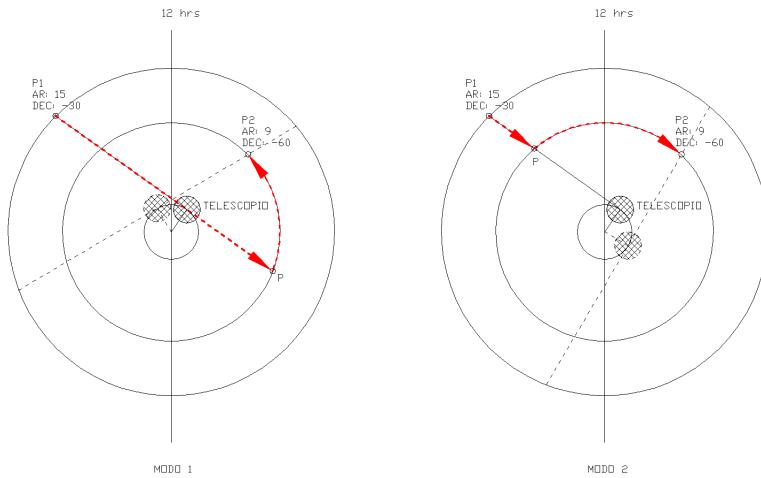


Figura 10: Movimientos alternativos del telescopio

3.3.4. Control de Avance

Luego de determinar el recorrido, se calculan la cantidad de pasos necesarios para alcanzar dicha coordenada. Para esto se realiza el siguiente cociente por cada eje:

$$S_{AR} = \frac{d_{AR}}{M_{AR}}$$

$$S_{DEC} = \frac{d_{DEC}}{M_{DEC}}$$

Donde:

- S_{AR} : Cantidad de pasos eje Ascensión Recta.
- d_{AR} : Distancia a recorrer para el eje Ascensión Recta.
- M_{AR} : Constante mecánica para el eje Ascensión Recta.
- S_{DEC} : Cantidad de pasos eje Declinación.
- d_{DEC} : Distancia a recorrer para el eje Declinación.
- M_{DEC} : Constante mecánica para el eje Declinación.

Los motores paso a paso requieren, para lograr un avance parejo y pseudo-continuo, que se exciten las bobinas en una secuencia determinada, sin perder ning n estado intermedio porque eso generaría una p rdida de pasos en el motor. La secuencia de avance se realiza por medio de una tabla en ROM que contiene los estados del motor para el control de a medio paso. Entonces, seg n el sentido en el que se recorra la tabla, se consigue un movimiento en una u otra direcci n. Existe una puntero a la tabla por cada motor, para lograr la independencia solicitada entre los mismos.

El control de avance se hace dentro de una interrupci n de un timer para poder tener una velocidad controlada.

3.3.4.1. Control de velocidad y Rampa de velocidad

Las velocidades de movimiento del motor tolerables por el sistema son múltiplos de la velocidad sidérea (velocidad de giro de la Tierra). Ya que esto facilita la estimación del corrimiento de coordenadas por el giro de la Tierra. El controlador de motores posee un sistema independiente para agregar o quitar pasos según si se está moviendo en contra o a favor de la rotación de la Tierra, respectivamente.

La demora de tiempo necesaria para lograr estas velocidades se explica en la sección 4.2. Por otro lado, arranques o paradas bruscas pueden hacer que el telescopio se vaya corriendo de la calibración. Para evitar eso, se le agrego al control una rampa de velocidad de pendiente fija, pero valor máximo configurable automáticamente según la cantidad de pasos a recorrer. De esta forma se lograron arranques y paradas suaves.

3.3.4.2. Control de Torque

Las velocidades máximas alcanzadas son del orden de 250 veces la velocidad sidérea, es decir, unos 500 medios pasos por segundo aproximadamente. Esto hace que se esté forzando al motor paso a paso a trabajar en el límite de lo que puede realizar, según su curva torque vs. velocidad para 5V. Es decir, que el sistema tenía un límite máximo de velocidad posible, ya que comenzaría a perder pasos por no tener el torque suficiente. Este límite se encontraba muy por debajo de lo que se esperaba unos 100 medios pasos por segundo, por lo que recorría 12 horas de Ascensión Recta en 15 minutos aproximadamente.

Para aumentar dicho límite, se diseño un control de torque el cual simplemente realiza una modulación tipo PWM sobre las bobinas del motor y con una tensión pico de 12V. El controlador si desea aumentar la velocidad, debe incrementar el ancho del pulso, entregando mayor energía a las bobinas del motor, aumentando el torque y evitando la pérdida de pasos.

De esta forma y con esta velocidad máxima alcanzada, se pueden recorrer 12 horas de Ascensión Recta en menos de 3 minutos.

3.4. Fuente de Alimentación

Para simplificar el mecanismo de alimentación del sistema se diseño un regulador switching utilizando como base el integrado L4962EA. La idea es sencilla, el sistema debe alimentarse con 12V/3A, de la cual se extrae la tensión para alimentar a los motores paso a paso. Luego el regulador permite una reducción de tensión a 5V con buen rendimiento para la alimentación de toda la lógica.

La línea de 12V posee una protección contra inversión de conexión dada por el diodo de potencia SBL2040.

4. Memoria de Cálculos

4.1. Calculo de resolución mínima de cada eje

Se pretende determinar que avance se tiene por cada 1/2 paso del motor paso a paso utilizado. Los ejes del telescopio utilizados son mecánicamente idénticos. Cada uno tiene adosado una corona de $C = 205$ dientes, la cual se acopla a un sin fin. Este último tiene una polea que mediante una correa se acopla al eje del motor. La relación de reducción es de $\frac{21,7}{47,8}$. Es decir, que por cada grado que gira el motor, el sin fin gira $R_M = \frac{21,7}{47,8}(\circ)$. En la Figura 11 puede observarse esta cuestión.

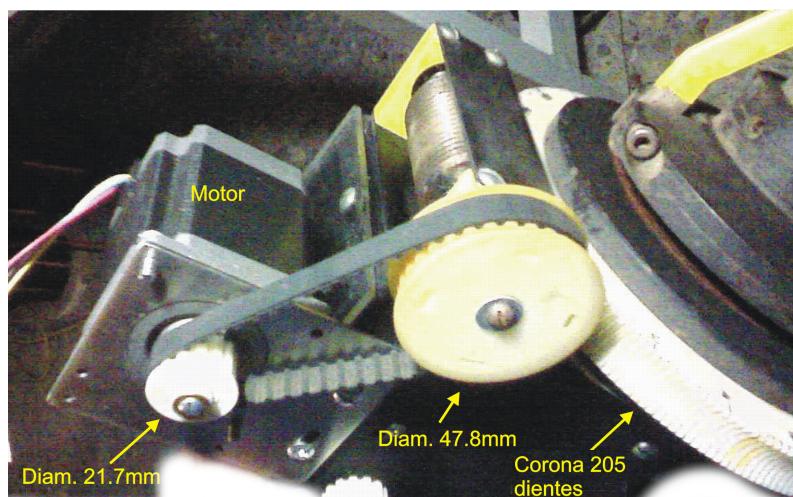


Figura 11: Acople del motor al eje

Cada motor tiene un paso mínimo de $1,8^\circ$, pero moviéndolo de a medio paso por vez, conseguimos una resolución de $0,9^\circ = S_m$. Entonces para estimar qué ángulo se realizará por cada medio paso, se realiza la siguiente ecuación:

$$S_{DECmin}[arcseg] = S_m \frac{R_M}{C} 3600$$

$$S_{DECmin} = 0,9^\circ \frac{21,7}{47,8} \frac{1}{205} 3600 = 7,175017859[arcseg]$$

Esto está expresado en segundos de arco en grados, es decir, que es la resolución mínima para el eje de Declinación. Para calcular el Ascensión Recta, hay que pasarlo a la escala de horas mediante la relación $R = \frac{24[hsl]}{360^\circ} = \frac{1}{15}$. Por lo la relación mecánica para este eje será:

$$S_{ARmin} = 0,478334523[arcseg]$$

4.2. Cálculo de velocidad sidérea y de desplazamiento

El tiempo sidéreo está íntimamente ligado a la rotación de la Tierra. El mismo indica el tiempo real en que la Tierra da una vuelta completa, que es aproximadamente en $23^h56^m04^s$. Para comprender más esta cuestión remitirse al apéndice D.

Para mantener el sistema en estación, es necesario tener un reloj sidéreo calculado según la resolución mínima de los pasos del motor calculadas en la sección Calculo de resolución mínima de cada eje. Ya que, para hacer seguimiento de un objeto, es necesario compensar el giro de la Tierra moviendo el eje de Ascensión Recta a una velocidad contraria e igual al giro de la Tierra.

Entonces, la Tierra recorre $C_{AR} = 24^h = 86400[\text{arcseg}]$ del eje de Ascensión Recta en un día sidéreo, es decir en $t_{sid} = 23^h56^m04^s = 86164[\text{seg}]$. Con esto se entiende como en este sistema de coordenadas las estrellas mantienen la posición.

La relación $v_{sid} = \frac{C_{AR}}{t_{sid}}$ indica la velocidad sidérea de la Tierra.

Por otro lado, se sabe que cada paso mínimo del eje de Ascensión Recta es de

$$S_{ARmin} = 0,478334523[\text{arcseg}]$$

. Con estos datos puede estimarse el tiempo entre pasos como:

$$t_s = \frac{S_{ARmin}}{v_{sid}}$$

$$t_s = 0,477027961[\text{seg}]$$

Quiere decir, que para realizar seguimiento el sistema tiene que realizar medio paso cada el tiempo t_s calculado.

Los timers del microcontrolador, que son los encargados de realizar esta demora en forma pareja y continua, tienen una resolución discreta que no permite realizar esta demora exacta. Por otro lado, era necesario generar demoras de tiempo que eran partes enteras de t_s para lograr avances a velocidades proporcionales a la sidérea mediante un único timer, dado que eran escasos en el ATmega168. Estas dos problemáticas se combinaron para encontrar una solución común.

Se utilizó un único timer y se utilizó una variable como divisor de velocidad (D). Es decir, que el timer genera una demora fija de tiempo (menor a t_s), calculada para una velocidad máxima y con el divisor se agregan múltiplos de ese tiempo hasta lograr una velocidad semejante a la sidérea. Es decir, que si $D=1$, generaría una demora de tiempo para un avance a velocidad máxima y si vale el máximo genera un tiempo semejante a t_s . La ecuación para calcular el tiempo es la siguiente:

$$t_{s_{DISC}} = D_{max} \frac{P_r(T + 1)}{f_{clk}}$$

Donde:

- D_{max} : Divisor máximo de velocidad. A este valor genera t_s
- P_r : Prescaler del timer. Valores posibles: 1, 8, 64, 256 y 1024.
- T : Registro de cuenta máxima del timer: [0-255]
- f_{clk} : Frecuencia de Clock: 20MHz

Se pretendía lograr una velocidad máxima de 250 veces la sidérea, por lo que la demora de tiempo debería ser 250 veces más pequeña que el tiempo t_s antes calculado. Para esto se procedió a buscar la relación numérica entre las variables mostradas mediante un script. Sabiendo que la demora de tiempo nunca podría cumplir con el tiempo t_s calculado, se procedió a calcular la que introdujera el mínimo error o mejor dicho la que requiriese una corrección cada tiempos prolongados.

Se tomaron los siguientes valores:

- $D_{max} = 1000$
- $P_r = 64$
- $T = 148$

Con estos valores, se consigue una demora de tiempo de $t_{s_{DISC}} = 0,4768[seg]$. El error que se tiene es de:

$$e_t = t_s - t_{s_{DISC}} = 2,27961[mseg]$$

Puede verse, que el sistema avanza más rápido de lo que gira la Tierra, por lo que esto introduciría un problema de deriva, es decir, que el objeto encontrado comenzaría a desplazarse lentamente del ocular. Para corregir esto, deber saltarse un paso cada cierto tiempo. Si cada paso requiere a velocidad sidérea de una demora de t_s , teniendo un error e_t en el tiempo, debo saltar un paso cada P_s pasos, según la relación:

$$P_s = \frac{t_s}{e_t}$$

$$P_s = 2093[pasos]$$

4.3. Cálculo Memoria del Microcontrolador

Mediante este cálculo, se pretende estimar la cantidad de memoria ROM utilizada por el microcontrolador central de la unidad HandSet y ver que elementos conviene llevarlos a una memoria externa.

Básicamente se estima que la memoria estará en gran parte por 3 cosas: código, imágenes del LCD y catálogos.

Se estima que el código del microcontrolador ocupará unos 70KB de memoria.

Las imágenes del LCD poseen 128 columnas por 8 filas de 1 Byte de longitud. Por lo que cada imagen pesará :

$$\text{Imagen} = 128[\text{col}]8[\text{fil}]1[B] = 1024B = 1KB$$

Si se tienen alrededor de 31 imágenes para almacenar, ocuparán 31KB en la memoria.

Dado que el microcontrolador posee 128KB de ROM, el espacio restante para los catálogos, suponiendo que cada elemento del catálogo requiere 14B (3B para Ascensión Recta, 3 para Declinación y 8 para el nombre):

$$C = \frac{128[KB] - 70[KB] - 31[KB]}{14B} = 1974[obj]$$

De acuerdo a lo relevado, en el mercado actual los telescopios de una gama media cuentan con una base de datos de 13000 objetos.

Considerando esta situación, optamos por utilizar una memoria SD externa para cubrir la necesidad de mayor capacidad. Tomando como base 13000 objetos necesitaríamos:

$$M = 14[B] 13000 = 182000[B]$$

Por lo que con una memoria de 1MB entrarían todos los objetos y más.

5. Lista de Componentes

5.1. Lista de Componentes del HandSet

Tipo de componente		
Descripción	Modelo	Cantidad
Placa Principal		
Microcontrolador	ATMEGA128	1
Reloj tiempo real	DS1307	1
Resistencia Variable	10KΩ	1
Resistencia	2,2 Ω	1
Resistencia	15 Ω	1
Resistencia	1KΩ	1
Resistencia	10KΩ	4
Cristal	8MHz	1
Cristal	32.768Hz	1
Capacitor Electrolítico	220 μF	1
Capacitor Cerámico	10 nF	1
Capacitor Cerámico	100 nF	5
Capacitor Cerámico	22 pF	2
Transistor	BC548	1
Buzzer	(genérico)	1
Conecotor (J11)	Tipo peine 5 pines	1
Conecotor (J2,J6,SD)	Tipo peine 6 pines	3
Conecotor (J1,J9,J12,SPI,GND)	Tipo peine 8 pines	5
Lector de SD	(genérico)	1
Memoria SD	(genérica)	1
Display LCD Gráfico		
LCD Gráfico 128x64	WG12864A-YYK-VN	1
Conecotor (J4,J5)	Tipo peine 6 pines	2
Conecotor (J3)	Tipo peine 8 pines	1
Teclado		
Resistencia	10KΩ	1
Diodo	1N4148	4
Capacitor Cerámico	100 nF	1
Pulsador	(genérico)	16
Conecotor Serial	RJ45 (hembra)	1
Conecotor Motor	RJ11 (hembra)	1
Conecotor (J8)	Tipo peine 8 pines	1
Conecotor (J7)	Tipo peine 5 pines	1

Tabla 8: Lista de Componentes del HandSet

5.2. Lista de Componentes del Motor Driver

Tipo de componente		
Descripción	Modelo	Cantidad
Microcontrolador	ATMEGA168	1
Doble Puente H	L298	2
Resistencia	0,47 Ω	4
Resistencia	330 Ω	1
Resistencia	2,7KΩ	1
Resistencia	8,2KΩ	1
Resistencia	10KΩ	1
Cristal	20MHz	1
Diodo	1N4148	1
Diodo	LED	1
Diodo	1N5408	16
Capacitor Electrolítico	1000 μF	1
Capacitor Electrolítico	470 μF	1
Capacitor Electrolítico	100 μF	2
Capacitor Electrolítico	47 μF	1
Capacitor Electrolítico	22 μF	1
Capacitor Electrolítico	10 μF	1
Capacitor Electrolítico	1 μF	1
Capacitor Cerámico	100 nF	7
Conector (J3,J4)	Tipo peine 5 pines	2
Conector (J2)	Tipo peine 7 pines	1
Conector (J1)	Tipo peine 15 pines	1

Tabla 9: Lista de Componentes del Motor Driver

5.3. Lista de Componentes de la Fuente de Alimentación

Tipo de componente		
Descripción	Modelo	Cantidad
Pwr. Switching Reg.	L4962	1
Resistencia	15 Ω	1
Resistencia	330 Ω	1
Resistencia	4,7KΩ	1
Diodo	UF4007	1
Diodo	SBL2040	1
Diodo	LED 5mm	1
Capacitor Electrolítico	470 μF	1
Capacitor Electrolítico	220 μF	1
Capacitor Electrolítico	2,2 μF	1
Capacitor Cerámico	100 nF	2
Capacitor Cerámico	33 nf	1
Capacitor Cerámico	2,2 nF	1
Inductor	220 μHy	1
Conector	Tipo peine 5 pines	1
Conector	Tipo peine 10 pines	1
Bornera	2 polos genérica	1

Tabla 10: Lista de Componentes de la Fuente de Alimentación

5.4. Lista de Componentes del adaptador Serie MAX232

Tipo de componente		
Descripción	Modelo	Cantidad
Multichannel RS-232	MAX232	1
Capacitor Electrolítico	1μF	4
Capacitor Electrolítico	10μF	1
Conector	Tipo peine 5 pines	1
Conector	Tipo peine 10 pines	1

Tabla 11: Lista de Componentes del adaptador Serie MAX232

6. Planos eléctricos y mecánicos

6.1. Esquemático HandSet

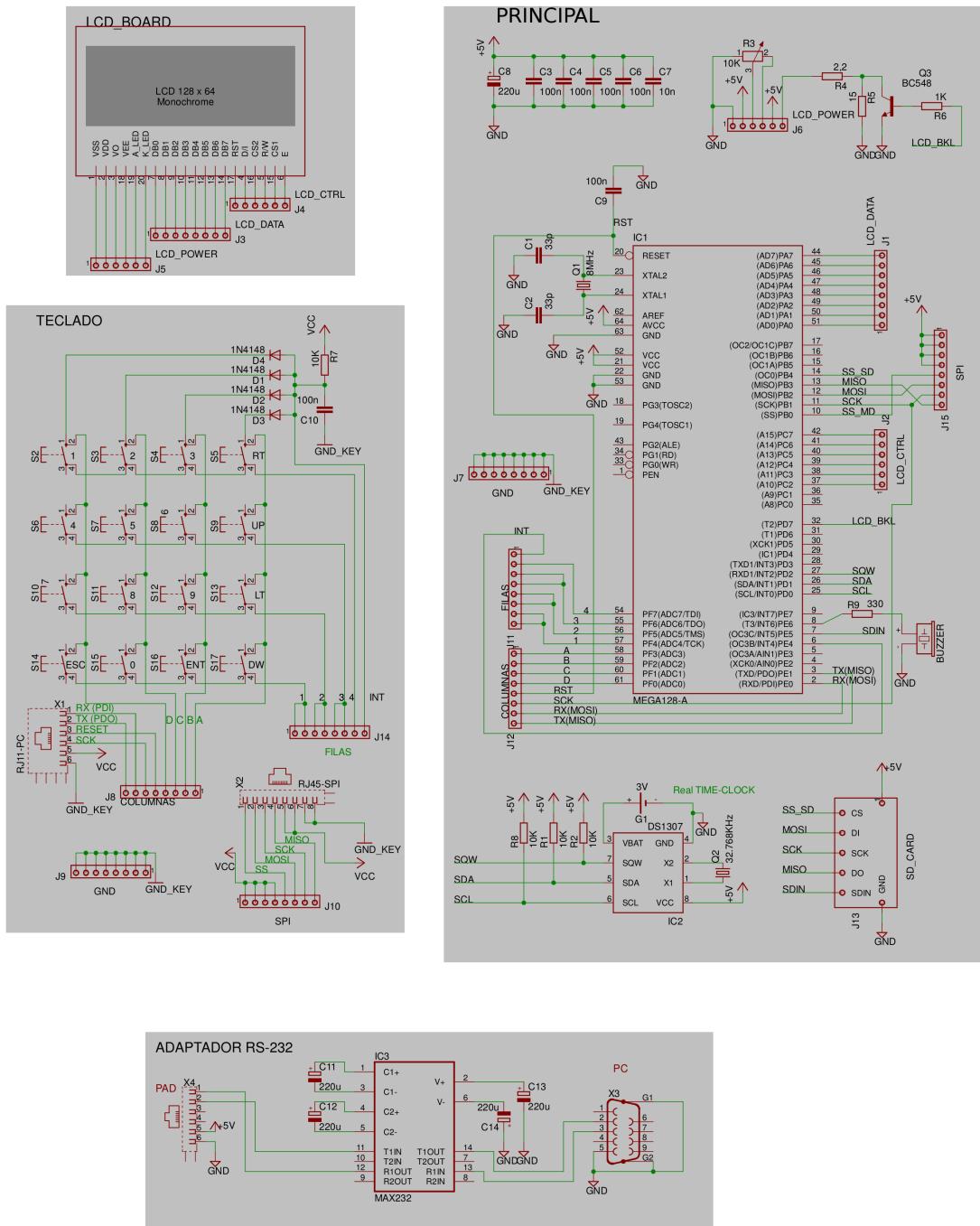


Figura 12: Diagrama esquemático del HandSet

6.2. Circuito impreso del HandSet

6.2.1. Placa Principal

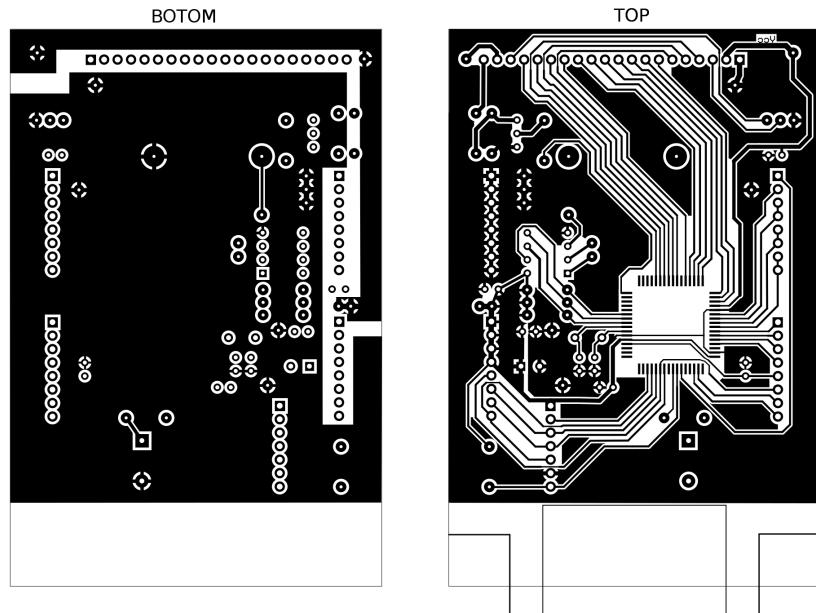


Figura 13: Circuito Impreso de la placa principal del HandSet

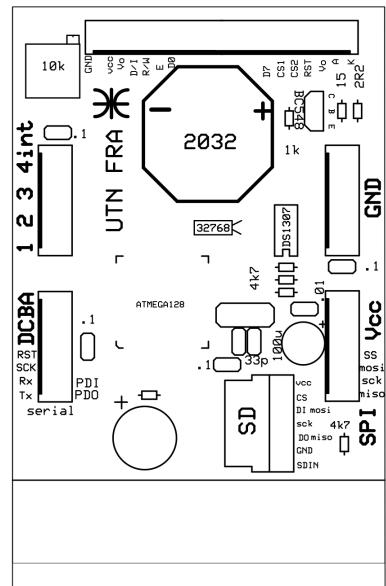


Figura 14: Referencias del Circuito Impreso de la placa principal del HandSet

6.2.2. Placa del Teclado

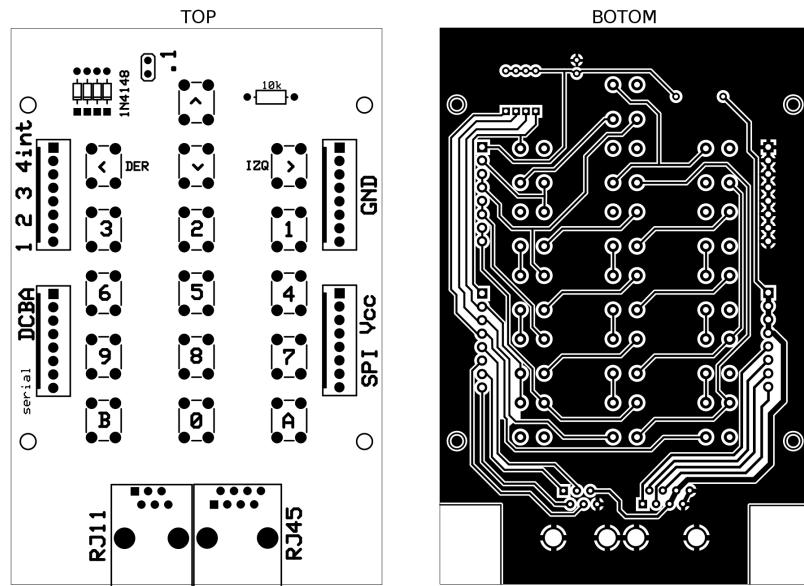


Figura 15: Circuito Impreso del Teclado del HandSet

6.2.3. Placa adaptador Serie

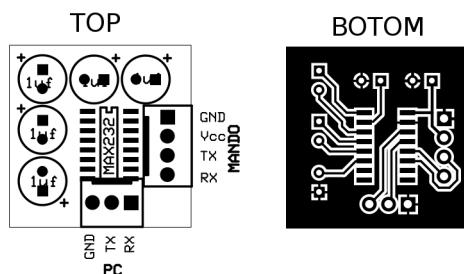


Figura 16: Circuito Impreso del Adaptador MAX232 serie del cable RJ11/USB

6.3. Esquemático Motor Driver

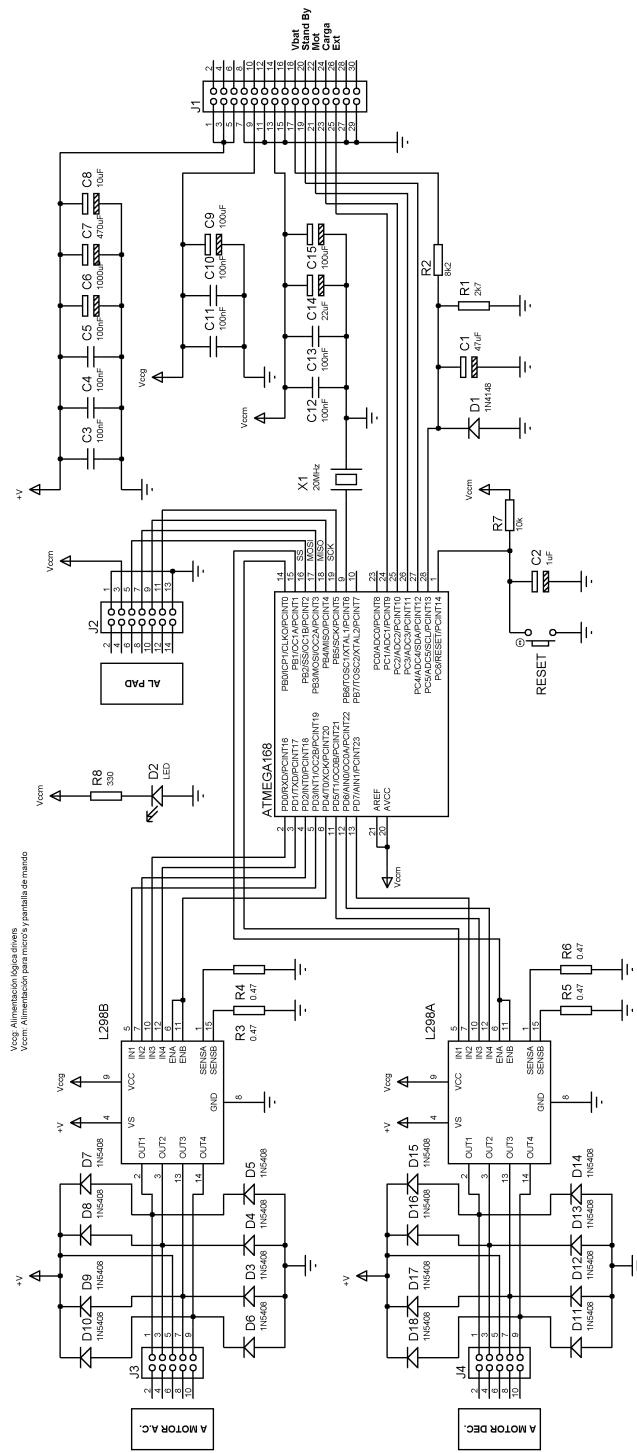


Figura 17: Diagrama esquemático del Motor Driver

6.4. Circuito Impreso del Motor Driver

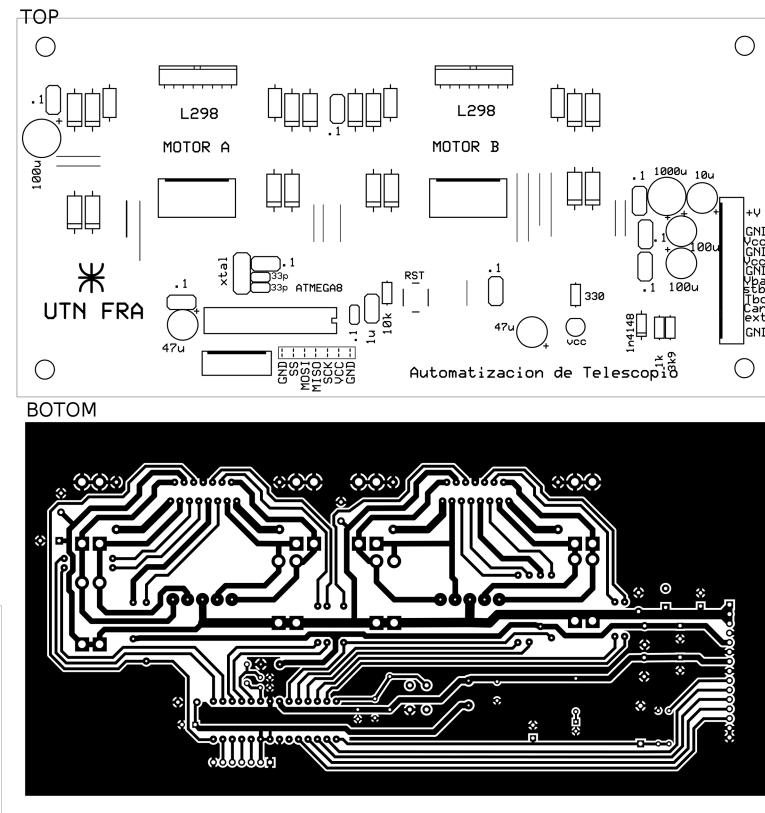


Figura 18: Circuito Impreso del Motor Driver

6.5. Esquemático Fuente de Alimentación

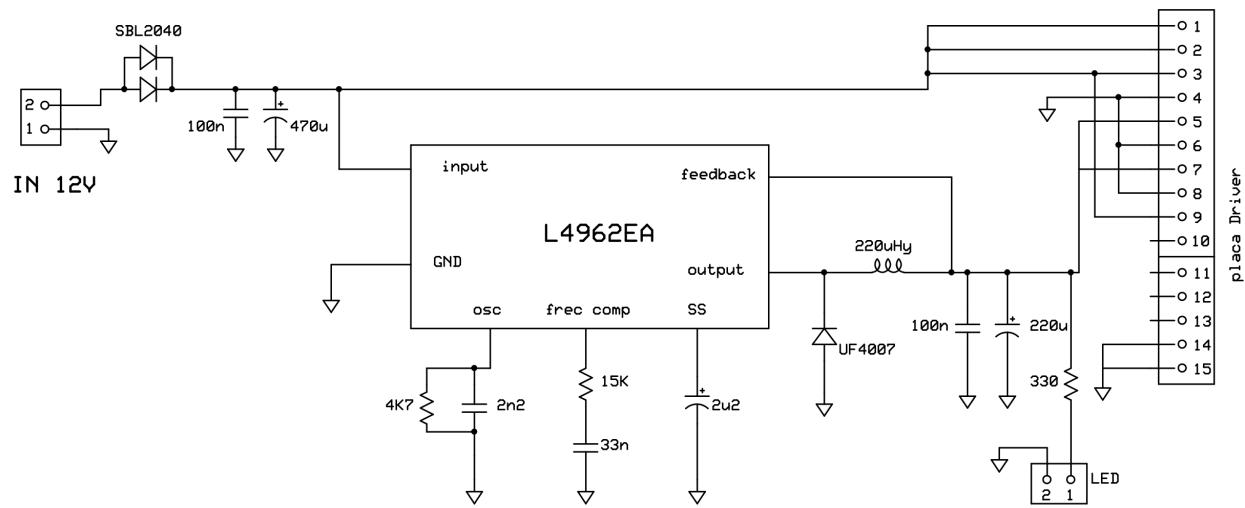


Figura 19: Diagrama esquemático de la Fuente de Alimentación

6.6. Circuito Impreso de la Fuente de Alimentación

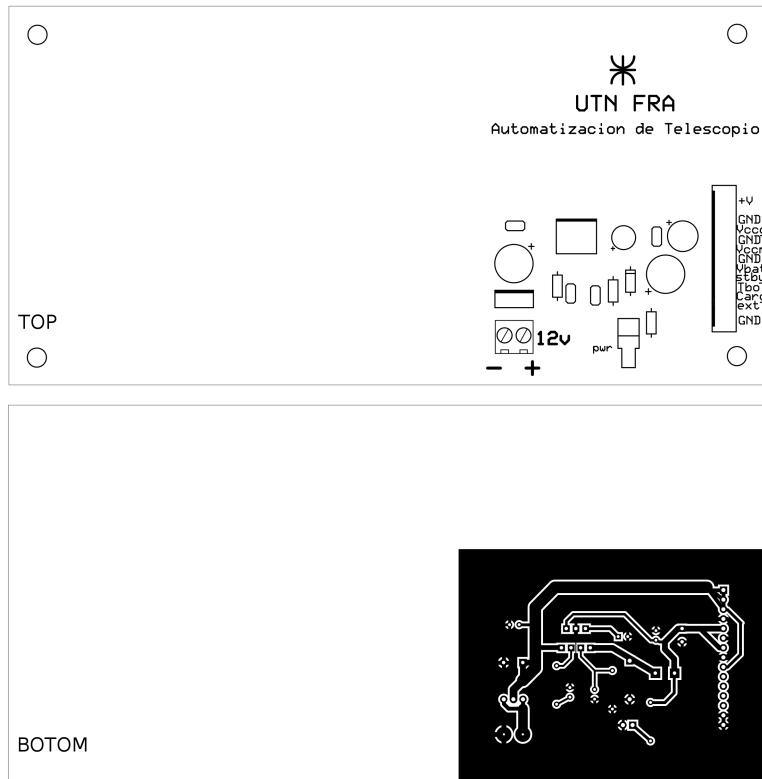


Figura 20: Circuito Impreso de la Fuente de Alimentación

7. Métodos de ajuste y control

Debido a que el sistema cuanta con diferentes etapas, se evaluarán cada una de forma separada. Dependiendo el módulo analizado, se llevarán a cabo distintos ajustes y métodos de verificación.

7.1. Ajuste y Control del HandSet

7.1.1. LCD Gráfico

El control sobre el display gráfico se realiza previo a cerrar el gabinete del HandSet habiendo montado la placa principal y la del teclado. Para este test, se le carga un firmware de prueba al microcontrolador ATmega128 que envía imágenes al LCD. Este Test pone en evidencia cualquier tipo de falla en los pixels del LCD, contraste, brillo, etc.

7.1.1.1. Procedimiento de Ajuste y Control

Previo a terminar el ensamble del HandSet, es necesario el ajuste del contraste del display LCD, para esto es necesaria una fuente regulada en 5V con un conector RJ45.

1. Colocar el LCD en el HandSet
2. Alimentar con 5V a través del conector RJ45 del HandSet.
3. Cargar firmware de prueba en el ATmege128
4. Con un destornillador plano variar el valor de el preset R3. Hasta hacer visible la imagen.
5. Verificar el correcto funcionamiento del LCD

Luego de este procedimiento, puede cargarse la versión definitiva del firmware del ATmega128.

7.1.1.2. Posibles fallas

El LCD no muestra información.

- Revisar las conexiones entre el micro y el LCD.
- Revisar el preset R3 y su conexión.
- Probar con otro LCD.
- Cambiar la placa del microcontrolador y repetir el procedimiento.

El LCD no prende todos los pixels.

- Cambiar el LCD.

7.1.2. Comunicación SPI y teclado

El control sobre la comunicación SPI y el teclado se realizan en forma simultánea. Para ello es necesario disponer de una placa de emisión y recepción de datos vía SPI, que emule la funcionalidad del Motor Driver. La misma posee un display que muestra el comando y los datos enviados e información general de la comunicación: como cantidad de errores, tiempos, etc.

Para realizar estos controles se ingresan, a través del teclado, las coordenadas de algunos objetos, navegando por el menú del equipo requiriendo un uso absoluto del teclado.

7.1.2.1. Procedimiento de Control

Con este mecanismo se corrobora el correcto funcionamiento de la comunicación SPI desde el HandSet y el funcionamiento del teclado.

1. Se conecta el HandSet a la placa auxiliar de Test.
2. Se alimenta al sistema.
3. Se recorre el menú mediante el teclado.
4. Se ingresan unas coordenadas cualesquiera mediante teclado.
5. Se verifica, en el display de la placa auxiliar de Test, que la información enviada sea coherente con lo que se pretendió hacer.

7.1.2.2. Posibles fallas

El teclado no responde a lo que se pretende.

- Identificar las teclas.
- Reponer las mismas.

En la pantalla del HandSet aparece el error de "falla de comunicación"

- Revisar las conexiones y resistencias de pull-up.
- Medir las señales del microcontrolador.
- Cambiar el microcontrolador.

La pantalla de la placa auxiliar muestra información distinta a la que fue enviada.

- Revisar las conexiones y resistencias de pull-up.
- Medir las señales del microcontrolador.
- Cambiar el microcontrolador.

7.1.3. Real Time Clock (RTC)

Se verifica que el reloj de tiempo real realice en forma correcta el conteo de tiempo y que almacene en forma correcta los datos de la fecha, hora, posición geográfica, de la última posición del telescopio.

7.1.3.1. Procedimiento de Control

1. Alimentar con 5V a través del conector RJ45 del HandSet.
2. Se ingresan todos los datos necesarios en el menú de configuración: fecha, hora, huso horario, posición geográfica.
3. Se observa que el separador Hora y minutos (:) parpadee una vez por segundo. Esto garantiza el funcionamiento del RTC.
4. Se le quita la alimentación al HandSet, tomando nota de la Hora .
5. Transcurridos 5 minutos se vuelve a alimentar el HandSet.
6. Se corrobora que el tiempo haya sufrido un cambio de 5 minutos.
7. Se corrobora que los datos ingresados anteriormente: fecha, huso horario, posición geográfica; no hayan sufrido modificación alguna.

7.1.3.2. Posibles Fallas

No se observa el parpadeo del separador de Hora y Minutos (:)

- Revisar el integrado DS1307.
- Revisar el oscilador de 32.768Hz, oscile a la frecuencia deseada.
- Revisar la conexión entre el RTC y el ATmega128.

El reloj no mantiene los datos.

- Hay un problema con la pila que lo alimenta.
- Revisar el integrado DS1307.

El reloj no cuenta el tiempo en forma correspondiente

- Revisar el oscilador de 32.768Hz, oscile a la frecuencia deseada.
- Hay un problema con la pila que lo alimenta.

7.1.4. Lectura de la SD

Para controlar la lectura de la memoria SD, se procede a la búsqueda aleatoria de un objeto perteneciente a los catálogos almacenados en la misma.

El sistema posee un mecanismo de verificación para determinar si hay una memoria SD conectada o no en el HandSet, en caso de no reconocerla lo indica por el Display LCD Gráfico.

7.1.4.1. Procedimiento de Control

1. Conectar una memoria SD en el zócalo del HandSet.
2. Alimentar con 5V a través del conector RJ45 del HandSet.
3. Observar si en la inicialización informa si la SD no fué encontrada.
4. Recorrer el menú hasta: Búsqueda → Catálogos Base → Messier.
5. Buscar un objeto cualquiera del catálogo.
6. Verificar la posición del objeto contra el archivo desde una PC.
7. Repetir el procedimiento con el catálogo: IC y NGC.

7.1.4.2. Posibles Fallas La memoria SD no fue encontrada.

- Verificar el conector del HandSet de la SD.
- Verificar las conexiones entre el conector y el ATmega128.
- Verificar la integridad de los archivos de la memoria SD en una PC.

Los datos leídos por el HandSet no coinciden con los vistos en la PC.

- Verificar la integridad de los archivos de la memoria SD en la PC.

7.2. Circuito Motor Driver

Tener precargado una secuencia de movimientos en la memoria, la cual es utilizada para la verificación del comportamiento de los motores. La idea es generar combinaciones tanto en la dirección del movimiento como la velocidad de desplazamiento, y así de esta forma poder descartar fallas mecánicas y eléctricas en los mismos. Los ensayos deben realizarse en vacío como también en carga.

7.2.1. Procedimiento de Control

1. Conectar la placa de los motores a la PC (mediante una placa simuladora de comunicación SPI).
2. Vincular el eje de los motores a los encoder situados en la placa de banco de prueba. Los encoder, a su vez están vinculados a la PC.
3. Se alimenta a todo el sistema.
4. Se ejecuta el software de prueba que realiza una secuencia de los motores, para corroborar: velocidad de giro, pasos (para chequear si pierde alguno), velocidad de respuesta, etc.
5. En la PC se visualiza un mensaje que indica si el test dio correcto o no.

7.2.2. Posible Fallas

- Si no se verifica el giro en los motores, la falla puede radicar en la alimentación, así como también en la conexión entre el motor y la placa.
- Si no se logra la comunicación con la PC, posiblemente existe un problema en el micro-controlador Atmega168 (encargado de la comunicación SPI), o los conectores encargados de vincular la placa con la PC.
- Si las velocidades o pasos no son las correctas, se debe revisar la etapa de potencia.

7.3. Comunicación con la PC

Para verificar el correcto funcionamiento de la comunicación con la PC es necesario una computadora con el Software Stellarium o uno compatible con el protocolo Meade LX200, el HandSet y el Motor Driver.

Con esta prueba, además de chequear la comunicación con la PC, se testea al sistema en forma completa

7.3.1. Procedimiento de Control

1. Se conecta el HandSet al Motor Driver.
2. Vincular el eje de los motores a los encoder situados en la placa de banco de prueba. Los encoder, a su vez están vinculados a la PC.
3. Se alimenta al sistema.
4. Se recorre el menú mediante el teclado hasta el menú de la conexión a la PC.
5. Conectar con el Adaptador USB/serie el HandSet a la PC.
6. Configurar el Stellarium o Software compatible para el telescopio.
7. Iniciar la comunicación desde la PC.
8. Verificar si la PC identificó la posición del telescopio.
9. Solicitar que el telescopio vaya a una posición determinada desde el Software en la PC.
10. Verificar que en el Display del HandSet se observe las coordenadas de la posición a la que fue enviado el telescopio.
11. Verificar que el telescopio comience el recorrido.
12. Verificar que el telescopio llegue a la posición solicitada, tanto en la simulación como en la medición realizada con los encoder.

7.3.2. Posibles Fallas

No se reconoce el telescopio en la PC.

- Verificar si el COM (Windows) o el USBx (Linux) está ocupado por este sistema.
- Verificar el cable adaptador USB/serie.
- Verificar las conexiones del ATmega128 al conector RJ11.

8. Métodos de fabricación y mantenimiento

8.1. Proceso de Fabricación de Placas

Para la fabricación en serie del producto, se comienza a partir de la fabricación de los circuitos impresos tomando como base los diseños del equipo de prueba o prototipo.

Los circuitos impresos se pueden terciarizar o, dependiendo de los costos que ello conlleva y de que nuestro sistema no pertenece a un mercado de consumo masivo, podemos optar por la fabricación propia de los mismos debido a la amplia disponibilidad de herramientas y métodos actuales que nos permiten su realización.

Una vez obtenidos los circuitos impresos se procede al soldado de los componentes.

8.2. Gabinetes

Dado que para el HandSet son necesarios gabinetes que se adapten a nuestras necesidades particulares, éstos deben ser fabricados especialmente para nuestro sistema y la mejor opción para su obtención es tercerizarlos.

En cuanto a los gabinetes para alojar la placa de los drivers y la fuente, éstos son de fabricación estándar, por lo que pueden ser adquiridos sin inconvenientes.

8.3. Ensamble

Con las placas y los gabinetes listos, se procede al ensamblaje de las placas en los gabinetes correspondientes.

En el gabinete del HandSet se ensamblan la placa del teclado, la placa de la unidad central, el display LCD y el lector de memorias SD.

En el otro gabinete se ensamblan las placas restantes, que son la placa de los drivers de los motores y de la fuente de alimentación.

8.4. Mantenimiento

Dado que nuestro sistema carece de partes móviles y/o que sufran desgaste mecánico, no se requiere ningún tipo de mantenimiento en cuanto lubricación o similares.

El único mantenimiento que requiere el sistema es el cambio de la pila de litio de 3,3 V que mantiene en funcionamiento el RTC (Reloj de Tiempo Real). El cambio de esta pila es cuando ésta se agote, cada 10 años aproximadamente (según la hoja de datos del mismo).

9. Métodos de prueba y envejecimiento

9.1. Circuitos Impresos

Circuitalmente, el fenómeno al que debemos atender para que no se produzca un envejecimiento prematuro y posterior deterioro del sistema, es la temperatura. Por lo tanto, es necesario ensayar los circuitos que poseen los componentes que deben disipar mayores temperaturas debido a los valores de corriente que manejan, como ser los drivers de los motores (L298) y en la fuente, el integrado L4962EA y el diodo UF-4007.

Dado que el sistema se utilizará en lugares abiertos y de noche, la temperatura ambiente no es un factor determinante para el ensayo, por lo que no es crucial realizar dicho ensayo en una cámara de simulación medioambiental y sólo se hace un ensayo a temperatura ambiente sobreexigiendo los componentes. Las lecturas del ensayo se realizan, a través de una termocupla adecuada, en una pc.

A partir de los resultados obtenidos se determina si con la colocación de disipadores es suficiente o si es necesario el agregado de planos de conducción térmica, ventiladores, etc.

9.2. Gabinetes

Tomando en cuenta que tanto el HandSet de control del sistema como la fuente de alimentación y las placas que contienen los Drivers de los motores, se encuentran montados sobre gabinetes plásticos, es importante evaluar su respuesta frente a las altas temperaturas, humedad, impactos, etc. Con tales fines se llevarán a cabo los siguientes ensayos para poder evaluarlos:

9.2.1. Comportamiento Mecánico

Se efectúa un ensayo de impacto (Izod, Charpy, impacto a alta velocidad o ensayos de caída) donde podemos determinar la cantidad de energía que es capaz de absorber el material cuando recibe un golpe. Principalmente este ensayo debe realizarse ya que los gabinetes utilizados se acoplan a la montura, con lo cual existe el riesgo de que accidentalmente caigan al suelo por una mala maniobra.

9.2.2. Envejecimiento acelerado

Estos ensayos sirven de gran ayuda para tener información sobre la duración y comportamiento de los gabinetes empleados cuando están sometidos a un ambiente o condiciones determinadas. Para ensayarlos se utilizan cámaras especiales (cámara climática y de envejecimiento a la luz) donde se puede reproducir la influencia de la temperatura, la lluvia, la humedad, etc. De esta forma comprobamos si el material empleado se degrada.

10. Estudio de garantías

Para el estudio de confiabilidad del equipo, nos basamos en el manual de predicción de confiabilidad MIL-HDBK-217 PLUS, edición 2006, elaborado por el RIAC (Reliability Information Analysis Center) y en la información brindada por los fabricantes de semiconductores para el caso de los circuitos integrados. Para el análisis dividiremos el equipo en tres partes y calcularemos la confiabilidad de cada una de ellas en forma separada. Cada una de estas partes serán: HandSet de comando, placa de los drivers de los motores, placa de alimentación del equipo. Para el cálculo de la confiabilidad total, asumiremos que las tres partes forman un sistema en serie, ya que la falla de alguna de las partes hará que el equipo deje de funcionar. Véase el ejemplo en la Figura 21.

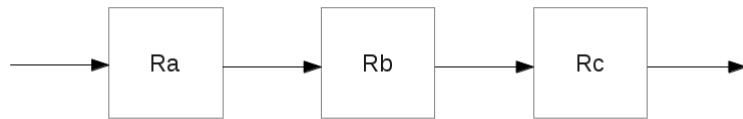


Figura 21: Esquema de un sistema serie

$$R_{TOTAL} = R_a R_b R_c$$

$$R_i = e^{tFR_i}$$

Realizaremos una memoria de cálculos detallando cada uno de los componentes, la cantidad de los mismos y su régimen de fallas (FR, Failure Rates).

El procedimiento de cálculo para el FR de cada componente se encuentra detallado en el MIL-HDBK-217PLUS, pero para facilitar los cálculos, tomaremos los valores que se encuentran en la tabla 2.3-2 de la página 69, de dicho manual.

10.1. Análisis de Garantías para el HandSet

Tipo de componente					
Ítem	Descripción	Modelo	Cantidad	FR (Unidad)	FR (Sumatoria)
Placa Principal					
1	Microcont. ¹	ATMEGA128	1	0,059171598	0,059171598
2	RTC ²	DS1307	1	0,000677204	0,000677204
3	Resist. Var.	10KΩ	1	0,04278	0,04278
4	Resist.	2,2Ω	1	0,000603	0,000603
5	Resist.	15Ω	1	0,000603	0,000603
6	Resist.	1KΩ	1	0,000603	0,000603
7	Resist.	10KΩ	4	0,000603	0,002412
8	Cristal	8MHz	1	0,13212792	0,13212792
9	Cristal	32.768Hz	1	0,18174748	0,18174748
10	Cap. Electro.	220 μF	1	0,000866	0,000866
11	Cap. Cerám.	10nF	1	0,004165	0,004165
12	Cap. Cerám.	100nF	5	0,004165	0,020825
13	Cap. Cerám.	22pF	2	0,004165	0,00833
14	Transist.	BC548	1	0,0129	0,0129
15	Buzzer	(genérico)	1	0,13212792	0,13212792
16	Soldaduras	Trough Hole	183	0,0002	0,0366
17	Con. (J11)	Tipo peine 5 pines	1	0,00084	0,00084
18	Con. (J2,J6,SD)	Tipo peine 6 pines	3	0,001008	0,003024
19	Con. (J1,J9,J12,SPI,GND)	Tipo peine 8 pines	5	0,001344	0,00672
Display					
1	LCD Gráf ³	WG12864A-YYK-VN	1	0,839376847	0,839376847
2	Con. (J4,J5)	Tipo peine 6 pines	2	0,001008	0,002016
3	Con. (J3)	Tipo peine 8 pines	1	0,001344	0,001344

Teclado					
1	Resist.	10KΩ	1	0,000603	0,000603
2	Diodo	1N4148	4	0,004577	0,018308
3	Diodo	LED	1	0,000135	0,000135
4	Cap. Cerám.	100nF	1	0,004165	0,004165
5	Pulsador	(genérico)	16	0,301404	4,822464
6	Soldaduras	Trough Hole	109	0,0002	0,0218
7	Con. Serial	RJ45 (hembra)	1	0,000741	0,000741
8	Con. Motor	RJ11 (hembra)	1	0,000741	0,000741
9	Con. (J8)	Tipo peine 8 pines	1	0,001344	0,001344
10	Con. (J7)	Tipo peine 5 pines	1	0,00084	0,00084
Lector de Memoria ⁶					
1	Reg. Lineal (LDO)	LM117 ⁵	1	0,000150103	0,000150103
2	Resist.	10KΩ	10	0,000603	0,00603
3	Cap. Electr.	10μF	1	0,000866	0,000866
4	Cap. Ceram.	100nF	1	0,004165	0,004165
5	Transist.	MosFET	4	0,0129	0,0516
6	Mem. SD ⁴	(genérico)	1	1	1
7	Switch	(genérico)	1	0,004392	0,004392
8	Soldaduras	Trough Hole	65	0,0002	0,013
9	Con.	Tipo peine 6 pines	1	0,001008	0,001008
$FR_{HandSet}$				7,442212071	

Tabla 12: Análisis de Garantías del HandSet

1. Dato tomado de la página de Atmel <http://support.atmel.no/bin/customer.exe?&action=viewKbEntry&id=266>
2. Dato tomado de informe presentado por Dallas Semiconductor para éste dispositivo <http://www.maximintegrated.com/reliability/product/DS1307.pdf>
3. Se tomó el valor de MTBF de otros display LCD al no disponer de los datos del display utilizado
4. Dato tomado de la hoja de datos de una memoria SD.
5. Dato tomado del documento Reliability Data, 21/08/2006, presentado por Linear Technology.
6. Debido a que no se encontró el FR del módulo lector de SD, se lo calculó a partir de su circuito, pero éste módulo no forma parte de nuestro desarrollo ya que se consigue armado.

El tiempo medio entre fallas para el HandSet será:

$$MTBF_{HandSet} = \frac{1}{FR_{HandSet}} = \frac{10^6 [hrs]}{7,44221} = 134368,68 [hr]$$

Aproximando: $MTBF_{HandSet} \simeq 15,34 [años]$

10.2. Análisis de Garantías para el Motor Driver

Ítem	Tipo de componente	Modelo	Cantidad	FR (Unidad)	FR (Sumatoria)
1	Microcont. ¹	ATMEGA168	1	0,059171598	0,059171598
2	Doble Puente H	L298	2	0,186915888	0,373831776
3	Resist.	0,47Ω	4	0,000603	0,002412
4	Resist.	330Ω	1	0,000603	0,000603
5	Resist.	2,7KΩ	1	0,000603	0,000603
6	Resist.	8,2KΩ	1	0,000603	0,000603
7	Resist.	10KΩ	1	0,000603	0,000603
8	Cristal	20MHz	1	0,163125104	0,163125104
9	Diodo	1N4148	1	0,004577	0,004577
10	Diodo	LED	1	0,000135	0,000135
11	Diodo	1N5408	16	0,001362	0,021792
12	Cap. Electro.	10μF	1	0,000866	0,000866
13	Cap. Electro.	22μF	1	0,000866	0,000866
14	Cap. Electro.	47μF	1	0,000866	0,000866
15	Cap. Electro.	100μF	2	0,000866	0,001732
16	Cap. Electro.	470μF	1	0,000866	0,000866
17	Cap. Electro.	1000μF	1	0,000866	0,000866
18	Cap. Electro.	1μF	1	0,000866	0,000866
19	Cap. Cerám.	100nF	7	0,004165	0,029155
20	Soldaduras	Trough Hole	180	0,0002	0,036
21	Con. (J3,J4)	Tipo peine 5 pines	2	0,00084	0,00168
22	Con. (J2)	Tipo peine 7 pines	1	0,001176	0,001176
23	Con. (J1)	Tipo peine 15 pines	1	0,00252	0,00252
$FR_{MDrivers}$				0,708214477	

Tabla 13: Análisis de Garantías del Motor Driver

1. Dato tomado de la página de Atmel [http://support.atmel.no/bin/customer.exe?
&action=viewKbEntry&id=266](http://support.atmel.no/bin/customer.exe?&action=viewKbEntry&id=266)

El tiempo medio entre fallas para el HandSet será:

$$MTBF_{MDrivers} = \frac{1}{FR_{MDrivers}} = \frac{10^6 [hrs]}{0,708214477} = 1412010 [hr]$$

Aproximando: $MTBF_{MDrivers} \simeq 161 [\text{años}]$

10.3. Análisis de Garantías para la Fuente de Alimentación

Ítem	Tipo de componente	Descripción	Modelo	Cantidad	FR (Unidad)	FR (Sumatoria)
1	Pwr. Switch. Reg		L4962	1	0,18691589	0,18691589
2	Resist.		15Ω	1	0,000603	0,000603
3	Resist.		330Ω	1	0,000603	0,000603
4	Resist.		4,7KΩ	1	0,000603	0,000603
5	Diodo		UF4007	1	0,027975	0,027975
6	Diodo		SBL2040	1	0,003925	0,003925
7	Diodo		LED	1	0,000135	0,000135
8	Cap. Electro.		2,2μF	1	0,000866	0,000866
10	Cap. Electro.		220μF	1	0,000866	0,000866
11	Cap. Electro.		470μF	1	0,000866	0,000866
12	Cap. Cerám.		2,2nF	1	0,004165	0,004165
13	Cap. Cerám.		33nF	1	0,004165	0,004165
14	Cap. Cerám.		100nF	2	0,004165	0,00833
15	Inductor		220μH	1	3,32E-06	3,32E-06
16	Soldaduras		Trough Hole	60	0,0002	0,012
17	Con.		Tipo peine 5 pines	1	0,00084	0,00084
18	Con.		Tipo peine 10 pines	1	0,00168	0,00168
19	Bornera		2 polos genérica	1	0,000336	0,000336
<i>FR_{Fuente}</i>					0,254877208	

Tabla 14: Análisis de Garantías de la Fuente de Alimentación

El tiempo medio entre fallas para el HandSet será:

$$MTBF_{Fuente} = \frac{1}{FR_{Fuente}} = \frac{10^6 [hrs]}{0,254877208} = 3923568 [hr]$$

Aproximando: $MTBF_{Fuente} \simeq 447 [\text{años}]$

10.4. Análisis de Garantías para Placa de Comunicación Serie, MAX232

Ítem	Tipo de componente	Descripción	Modelo	Cantidad	FR (Unidad)	FR (Sumatoria)
1	Multich. RS-232		MAX232 ¹	1	0,00419	0,00419
2	Cap. Electro.		1 μ F	4	0,000866	0,003464
3	Cap. Electro.		10 μ F	1	0,000866	0,000866
4	Soldaduras		Trough Hole	25	0,0002	0,005
5	Con.		Tipo peine 5 pines	1	0,00084	0,00084
6	Con.		Tipo peine 10 pines	1	0,00168	0,00168
FR_{ASerie}					0,0152	

Tabla 15: Análisis de Garantías de la Placa de Comunicación Serie, MAX232

1. Dato tomado de la página de Maxim Integrated:<http://www.maximintegrated.com/qa/reliability/fit.cfm?pn=MAX232>

El tiempo medio entre fallas para el HandSet será:

$$MTBF_{ASerie} = \frac{1}{FR_{ASerie}} = \frac{10^6 [hrs]}{0,0152} = 65789473,68 [hr]$$

Aproximando: $MTBF_{ASerie} \simeq 7510 [años]$

10.5. Calculos de Confiabilidad

El menor tiempo entre fallas se encuentra en el HandSet (15años). Por lo que este es el módulo más crítico del todo el sistema.

Calcularemos la confiabilidad de cada una de las partes para 1 año de vida (8760 hr de vida).

10.5.1. Confiabilidad del HandSet

Se calcula la confiabilidad del HandSet según su Tiempo Medio entre Fallas (MTBF), calculado previamente.

$$R_{HandSet} = e^{-\frac{t}{MTBF_{HandSet}}} = e^{-(\frac{8760}{134368})}$$

$$R_{HandSet} = 0,9369 = 93,69 \%$$

10.5.2. Confiabilidad del Motor Driver

Se calcula la confiabilidad del Motor Driver según su Tiempo Medio entre Fallas (MTBF), calculado previamente.

$$R_{MDriver} = e^{-\frac{t}{MTBF_{MDriver}}} = e^{-(\frac{8760}{1412010})}$$

$$R_{MDriver} = 0,9938 = 99,38 \%$$

10.5.3. Confiabilidad de la Fuente de Alimentación:

Se calcula la confiabilidad de la Fuente de Alimentación según su Tiempo Medio entre Fallas (MTBF), calculado previamente.

$$R_{Fuente} = e^{-\frac{t}{MTBF_{Fuente}}} = e^{-(\frac{8760}{3823568})}$$

$$R_{Fuente} = 0,9977 = 99,77\%$$

10.5.4. Confiabilidad del Adaptador Serie:

Se calcula la confiabilidad del Adaptador Serie según su Tiempo Medio entre Fallas (MTBF), calculado previamente.

$$R_{ASerie} = e^{-\frac{t}{MTBF_{ASerie}}} = e^{-(\frac{8760}{65789473,68})}$$

$$R_{ASerie} = 0,99999 = 99,999\%$$

10.5.5. Confiabilidad del Sistema

Con los valores de confiabilidad de cada una de las partes, obtenemos la confiabilidad total del equipo:

$$R_{TOTAL} = 0,9369 \cdot 0,9938 \cdot 0,9977 \cdot 0,99999 = 0,9288$$

$$R_{TOTAL} = 92,88\%$$

De esta forma podemos ver que la confiabilidad del equipo, para un año, es de 92,88% por lo que podemos ofrecer una garantía de un año sin correr grandes riesgos.

11. Legislación

Para este producto se considerará la verificación de su seguridad eléctrica mediante las normas IEC. Estas normas son escritas y publicadas por el Comité Electrotécnico Internacional y su alcance es para equipos hogareños y de oficina, semiconductores, fibra óptica, equipos de generación y transmisión de energía y nanotecnología, entre otros. Casi la totalidad de los países del mundo están afiliados a este comité, por lo que el cumplimiento del sistema automatizado del telescopio en estas normas nos garantiza una comercialización a futuro a nivel internacional, más rápido y seguro.

En nuestro producto, la norma a cumplir es la IEC 60950-1:2009, que es aplicable a equipos electrónicos de uso hogareño y de oficina con una tensión nominal menor a 600V. Los diseños del equipo se realizarán en función de esta norma y su cumplimiento será verificado mediante ensayos completos en algún laboratorio especializado.

Para lograr la certificación de nuestro producto, se realizará el proceso establecido por la Secretaría de Comercio. En primer lugar se contactará a alguna certificadora nacional como IRAM, TUV, etc. Estos emitirán un plan de ensayo con la norma correspondiente. Este pedido se lo envía junto al equipo a un Laboratorio de Ensayos auditado por el Organismo Argentino de Acreditación (OAA) y por el CB Scheme para la realización de los ensayos establecidos por la norma. En el caso de cumplimiento, se obtendrá la certificación y aprobación del producto, y estaremos en condiciones de comercializarlo.

Por otra parte, debemos considerar que para el desarrollo de nuestro sistema de automatización, se utilizaron algoritmos para el cálculo de la posición de los planetas desarrollados hace más de 30 años por T. van Flandern's y K. Pulkkinen y publicados en el paper *Low precision formulae for planetary positions* del **Astrophysical Journal Supplement Series** en el año 1980.

Con respecto a la comunicación serial entre el sistema de control y la PC/notebook, elegimos utilizar como interfaz con el usuario el software Stellarium. Stellarium es un software planetario gratis, con licencia bajo los términos de GNU (General Public License), disponible para Linux, Windows, y Mac OS X. Este usa OpenGL (Open Graphics Library) para lograr una proyección realística del cielo nocturno en tiempo real.

Stellarium fue desarrollado por el programador francés Fabien Chéreau, quién lanzó el proyecto en el verano de 2001. Otros desarrolladores involucrados fueron Robert Spearman, Johannes Gajdosik, Matthew Gates, Nigel Kerr, y Johan Meuris, quién fue el responsable del trabajo de arte.

Por último, los catálogos de estrellas, constelaciones y nebulosas fueron extraídos de sitios de internet donde se los puede encontrar con mucha facilidad. Algunos de ellos son el catálogo Messier, NGC, etc.

Los catálogos son de suma importancia para el sistema, ya que estos son una base de datos (con coordenadas, posiciones y nombres) de los objetos visibles en la esfera celeste.

12. Comercialización

12.1. Presentación e Introducción del producto desde un enfoque Comercial

El producto que deseamos comercializar es el sistema de automatización para telescopios, el cual tiene presencia en el mercado local hace unos 15 años. Básicamente este consta de un HandSet (interfaz con el usuario) donde uno puede indicar el objeto que desea observar y realizar el seguimiento. El sistema también tiene incluido la etapa de control de los motores responsables del movimiento, sobre la montura del telescopio.

Dentro del ámbito de la astronomía, este sistema de automatización tiene mucha aplicación y utilidad. Pueden ser encontrados tanto en aplicaciones de telescopios aficionados como también a niveles más profesionales y científicos, abarcando tanto jóvenes como adultos. Es por esa razón que nos encontramos con un espectro variado de fabricantes que se dedican a la comercialización de los mismos. Por ejemplo: Celestron, Meade, Galileo, etc.

Es claro que al querer comercializar este producto nos encontramos con una competencia ya establecida en el mercado, lo que nos hace optar por una estrategia un tanto agresiva para ingresar al mismo.

Una ventaja importante a destacar es el bajo costo de producción que tiene acompañado el sistema, con lo que se puede generar una fuerte inserción en el mercado con precios muy competitivos. Obviamente, tomando en cuenta la calidad de los productos de la competencia, no debemos dejar de lado ese aspecto y tomar como base la fidelidad y calidad para un comienzo lo más grato posible.

El objetivo considerado como ideal es, una vez ya asentados en el mercado, desarrollar variantes en el producto y tener una iniciativa de expansión y crecimiento para no perder el ritmo con respecto a la competencia, así como también llegar a ser una reconocida entidad en el rubro.

12.2. Vida Util, uso del producto

12.2.1. Producto Básico

Automatización para telescopios.

12.2.2. Producto Real

El sistema está compuesto principalmente por un HandSet o control de mando, el cual está montado sobre un gabinete plástico con un display gráfico y un teclado matricial. Como adicional consta con un puerto para tarjetas SD y Micro SD y un conector serial para la comunicación con la PC (uso de software Stellarium).

El HandSet se vincula físicamente mediante cableado a la etapa de control de motores, formado por drivers encargados de la parte de potencia del circuito. Se adicionará un manual de uso y descripciones técnicas para comprender el funcionamiento.

12.2.3. Producto Aumentado o Final

Se tendrá, por lo tanto, un sistema automatizado para control posicional de telescopios, con su gabinete y controles correspondientes, además de su manual de funcionamiento. Como beneficios post-venta se brindará una garantía adecuada y vías de comunicación (teléfonos, mail y redes sociales) para efectuar consultas. Podrá enviarse el producto a domicilio, considerando los gastos de envío, según la distancia.

El producto es una variante o modificación de los sistemas ya existentes, pero con el objetivo de introducirse en el mercado debido a su bajo precio, obteniéndose así la principal ventaja del mismo, por ser de producción nacional.

12.3. Aplicaciones o Mercado del Producto

En un comienzo la distribución del producto será dentro de la provincia de Bs.As.

Podrá abastecerse tanto al por mayor (casas dedicadas al rubro u ópticas) o individualmente (aficionados a astronomía o consumidores finales).

Como principales competidores o productos sustitutos se tendrá a las empresas reconocidas en el mercado dedicadas a la producción de artículos astronómicos u ópticos.

Los proveedores serán variados, dependiendo de los distintos componentes del producto (cables, gabinetes, plaquetas y componentes electrónicos, etc).

12.4. Determinación del Consumo

Para la determinación del consumo se utiliza el Método Deductivo:

El producto no pretende ser consumido en demasía al principio, por lo tanto nuestra estrategia

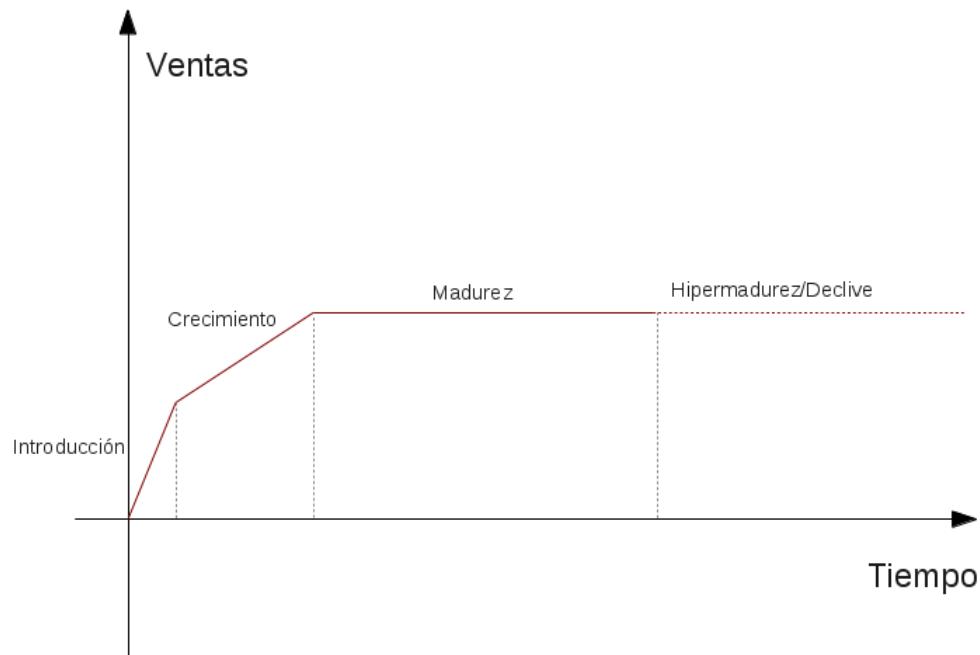


Figura 22: Vida del producto en el mercado

será en base a ofertas de varios tipos, en cuanto a precio y tiempos de prueba para los consumidores. Una vez reconocido en el mercado se tomará una actitud de expansión y más agresiva, como también para abarcar al resto del país, y quizás luego la exportación.

12.5. Participacion de la Competencia

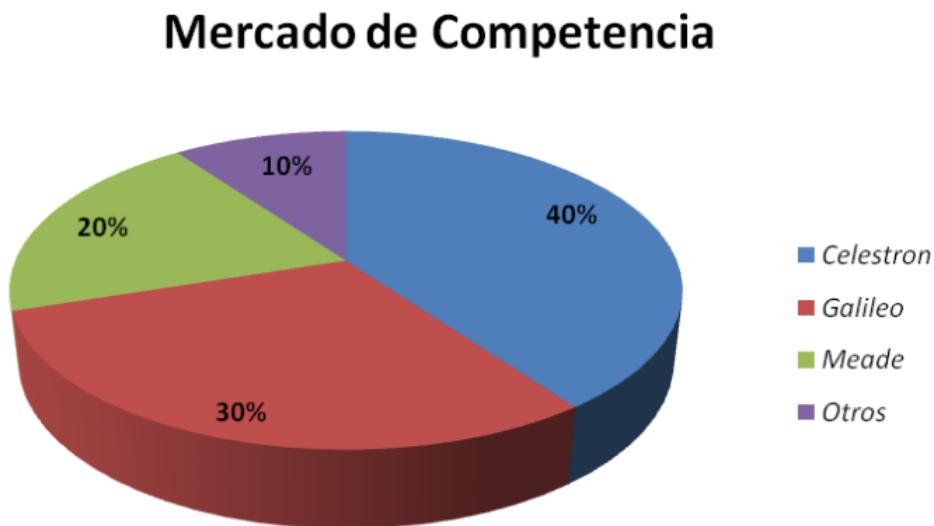


Figura 23: Mercado de la Competencia

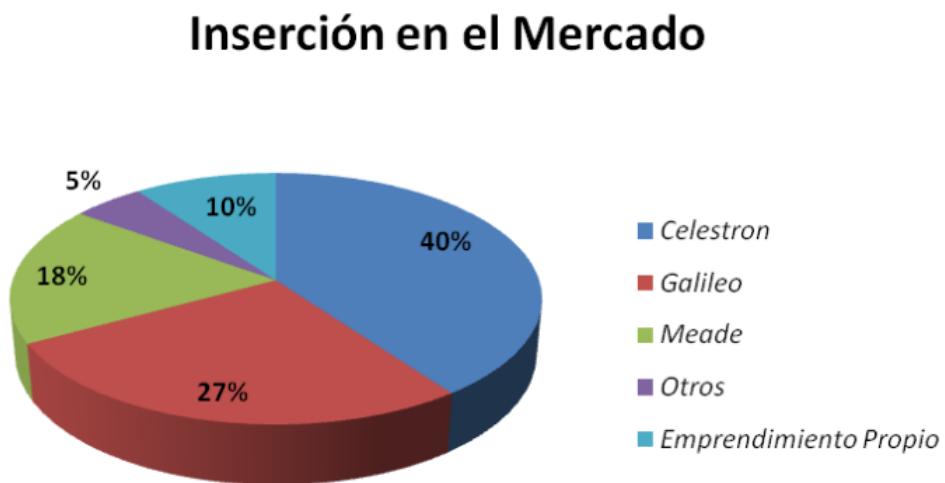


Figura 24: Inserción en el Mercado

12.6. Determinacion del crecimiento del mercado

El mercado al cual está orientado el producto se encuentra en una situación estable sin sobresaltos ya que el producto no es de consumo masivo.

Considerando la situación actual del país, el mercado se caracteriza por los costos elevados debidos a la inflación, razón por la cual creemos que nuestro producto de construcción puramente nacional va a tener una buena aceptación en el mercado debido a su bajo costo en relación a los competidores (mayormente extranjeros, por lo tanto se estaría sustituyendo importaciones), pero siempre manteniendo una calidad similar.

12.7. Plan de Ventas

12.7.1. Tipo de Mercado

En cuanto a consumidores se harán reducciones de precios u ofertas una vez consolidado el producto, es decir, superada la etapa de crecimiento. Convenio o descuento a Universidades que estudien la temática. Para los comerciantes, se ofrecerán productos con el fin de exponerlos en los distintos locales, y así posibilitar promocionarlos y que el consumidor conozca y pueda probar nuestros productos.

12.7.2. Tipo de distribución

Se distribuirá directamente al consumidor, o a intermediarios/minoristas como pueden ser locales de venta especializados u ópticas. Adicionalmente se proveerá de catálogos con nuestra variedad de productos, ya sea de mayor o menor gama, e información de contactos para consultas y servicio técnico.

12.8. Organigrama de Comercialización

En la figura 25 se muestra el organigrama de la empresa para la comercialización.



Figura 25: Organigrama de Comercialización

12.9. Estrategia de ventas, participación y políticas para ingresar al mercado

Como se trata de un mercado ya existente, la estrategia que tendremos para ingresar al mismo, será mediante bajos precios de venta, y que nuestros productos sean a su vez de calidad similar a lo ofrecido por la competencia. Al mismo tiempo, se intentará desarrollar un producto distinto, ya sea mediante la adición de nuevas funcionalidades, nuevos comandos de operación, mayor cantidad de catálogos, etc. La idea principal es lograr productos innovadores.

12.10. Layout

12.10.1. Proceso productivo

El equipamiento básico que necesitaremos para efectuar nuestro proceso productivo es el siguiente:

Componentes y Herramientas	Cantidad
Fuentes	10
Osciloscopio	3
Soldadores	5
Generador de funciones	3
Agujereadora	2
Componentes electrónicos	N/D
Estaño (Rollo)	10
Pinzas	4
Alicates	4
Destornilladores	5
Cables y conectores	N/D

Tabla 16: Lista de Componentes y Herramientas

Con respecto a la seguridad, debemos contemplar el siguiente equipamiento (especialmente al personal que se dedica a la soldadura):

Equipamiento de Seguridad	Cantidad
Matafuegos	1
Par de guantes	2
Lentes de seguridad	3
Maletín primeros auxilios	1

Tabla 17: Lista de Equipamiento de Seguridad

Considerando la materia prima tenemos:

- Placas con el circuito impreso.
- Gabinetes plásticos y metálicos.
- Pintura.
- Etiquetas.

El flujo de proceso que se adoptará para nuestro emprendimiento será la producción en línea, donde se soldaran componentes y se preparará la placa de manera de insertarla finalmente en el gabinete y hacer el detallado final. Es decir, el proceso será convergente desde los sectores de soldadura y de gabinetes.

12.10.2. Diagrama de Recorrido

A continuación se muestra el diagrama de recorrido dentro de la empresa.

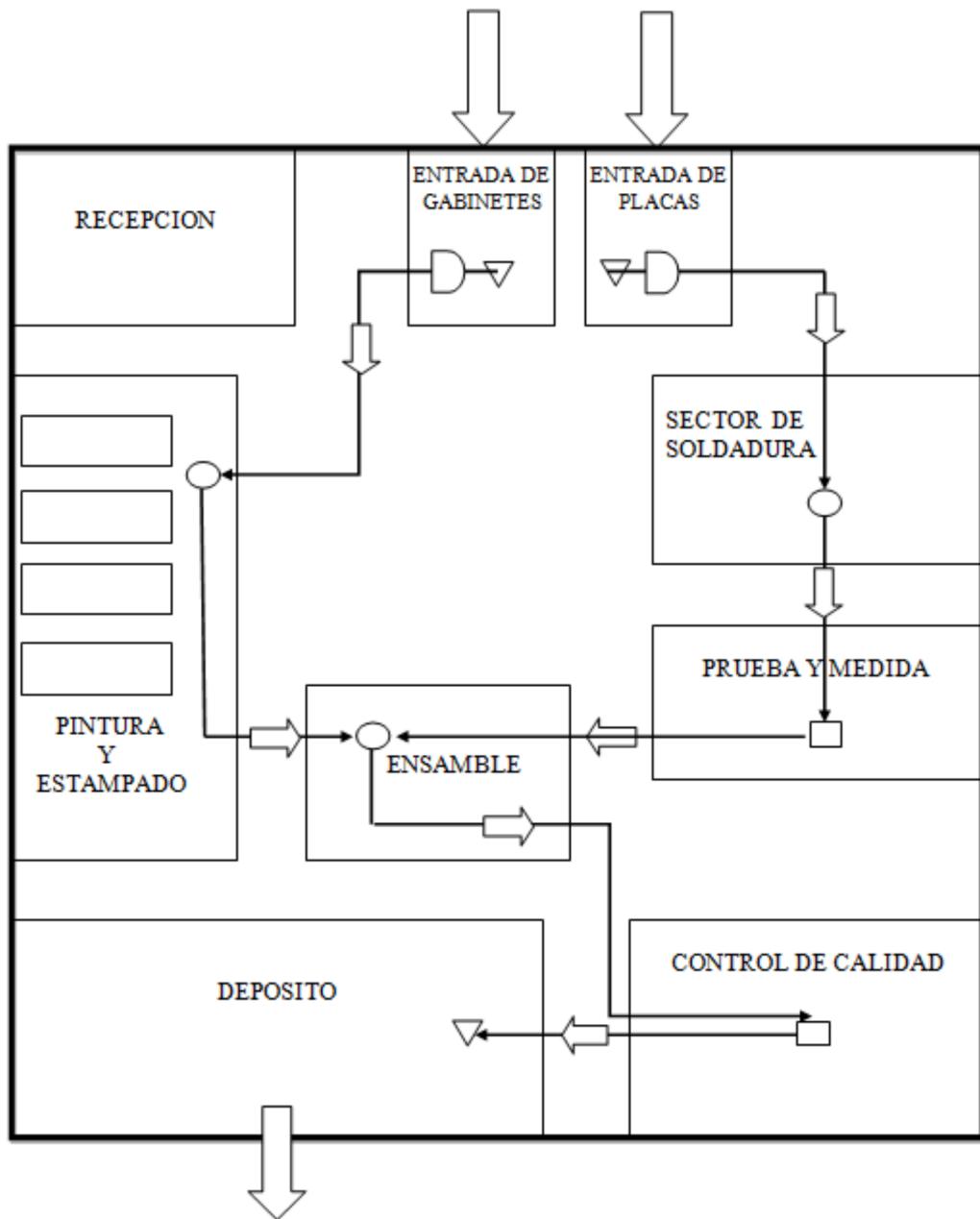


Figura 26: Diagrama de Recorrido

13. Referencias

Referencias

- [1] Muhammad Ali Mazidi, *The AVR microcontroller and embedded system*. Pretince Hall, 2009.
- [2] Paul Schlyter, *How to compute planetary positions*, <http://stjarnhimlen.se/comp/ppcomp.html>
- [3] Jean Meeuss *Moon position to within 10 arcsec*, <http://www.stargazing.net/kepler/moon3.html>
- [4] AstroRed, *Modificación de las coordenadas ecuatoriales por paralaje*, <http://www.astrored.org/usuarios/xgarciaf/paralaje1.html>
- [5] Stellarium, *Software for astronomy*, <http://www.stellarium.org/es/>
- [6] Meade, *The Autostar/LX200 Command Set*, <http://www.meade.com/support/CommandSet.html>
- [7] AVR-GCC, *GCC Compiler for AVR*, <http://www.nongnu.org/avr-libc>.
- [8] USBTiny, *AVR Programmer and SPI Interface*, <http://www.ladyada.net/make/usbtinyisp/>.
- [9] BMP2ASASM, *BMP converter*, <http://www.ticalc.org/archives/files/fileinfo/134/13438.html>
- [10] Skywatcher *POINTING AN EQUATORIAL TELESCOPE*, http://skywatcher.com/downloads/Instruction_Manuals/Pointing.pdf
- [11] Patagonia Tec Electrónica *Lector De Tarjetas SD Micro SD Spi para microcontroladores*, http://articulo.mercadolibre.com.ar/MLA-474712860-lector-de-tarjetas-sd-micro-sd-spi-arduino-arm-pic-_JM
- [12] Wikibook LaTeX *Guide to the LaTeX markup language*, <http://en.wikibooks.org/wiki/LaTeX/> El manual de Usuario y el informe fueron escritos bajo LaTeX.

A. Características del Sistema Ecuatorial

Este tipo de coordenadas presupone que todos los elementos observables en el cielo pueden proyectarse sobre una esfera de radio infinito denominada esfera celeste. Mediante dos ejes denominados: Ascensión recta (α) y Declinación (δ) pueden recorrerse todos los puntos de dicha esfera. La particularidad de este sistema es que el eje de rotación de la esfera celeste es el mismo que el de rotación de la Tierra, por lo que para seguir a los objetos celeste es necesario mover un sólo eje (Ascensión Recta (α)). En la Figura 27 puede observarse como se aplica este sistema de coordenadas para encontrar un objeto proyectado sobre esta esfera celeste.

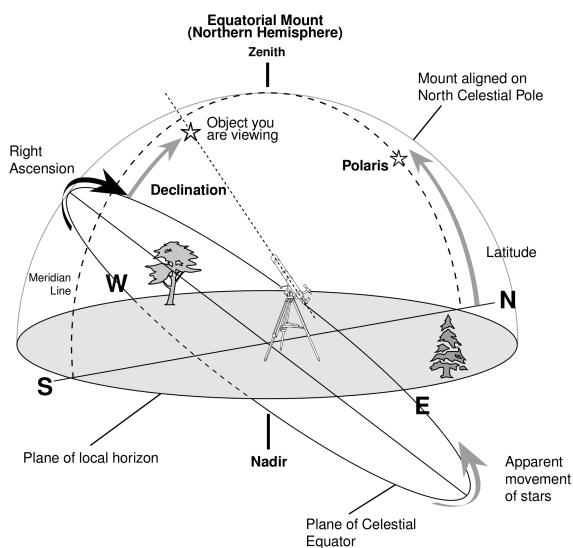


Figura 27: Sistema de Coordenadas ecuatoriales

B. Algoritmo Cálculo de Posición de Planetas

Todo elemento perteneciente al sistema solar, al tener un movimiento de traslación alrededor del Sol, el cual se combina con el de la Tierra (roto-traslación) hace que la posición en el sistema de coordenadas no sea fija y debe estimarse en todo momento. El cálculo de dicha posición, depende de varios factores, como pueden ser la fecha y la posición geográfica del observador. Este se basa en una serie de simplificaciones de otros métodos más complejos pero manteniendo una precisión aceptable para los telescopios de uso aficionado. Entre las simplificaciones realizadas, se pueden mencionar que no se tienen en cuenta todos los movimientos de la Tierra (son ocho en total) y sólo se considera la traslación y la rotación de esta, tampoco se tienen en cuenta las perturbaciones en las órbitas elípticas que generan todos los planetas por acción de la gravedad. Sólo son consideradas las perturbaciones de los planetas mayores como Júpiter y Saturno.

La precisión obtenida es de menos de un arco de minuto para el Sol y los planetas menores y, entre uno y dos arcos de minuto, para los demás cuerpos celestes.

El método utiliza seis parámetros que definen a una órbita de un cuerpo celeste. Algunos de estos parámetros son la distancia media al Sol, la excentricidad de la órbita, inclinación del plano de la órbita, etc. Con estos datos se obtienen las coordenadas rectangulares de un cuerpo celeste para un tiempo establecido, referidas al Sol (coordenadas heliocéntricas). Luego solo resta hacer un pasaje a coordenadas ecuatoriales y referirlas a la Tierra (coordenadas geocéntricas). Para mejorar la precisión se suman a las coordenadas obtenidas, términos de series que representan las perturbaciones de los planetas y otros fenómenos no considerados en un principio por el método.

C. Estimación de campo visible

Se resumen los cálculos de como estimar el campo visible por un usuario en un tiempo y lugar determinados. Entonces en la Figura puede observarse el efecto que produce el horizonte en la esfera celeste, como el mismo limita el campo de visión. También puede observarse como el hecho de cambiar de latitud, modifica a este campo visible.

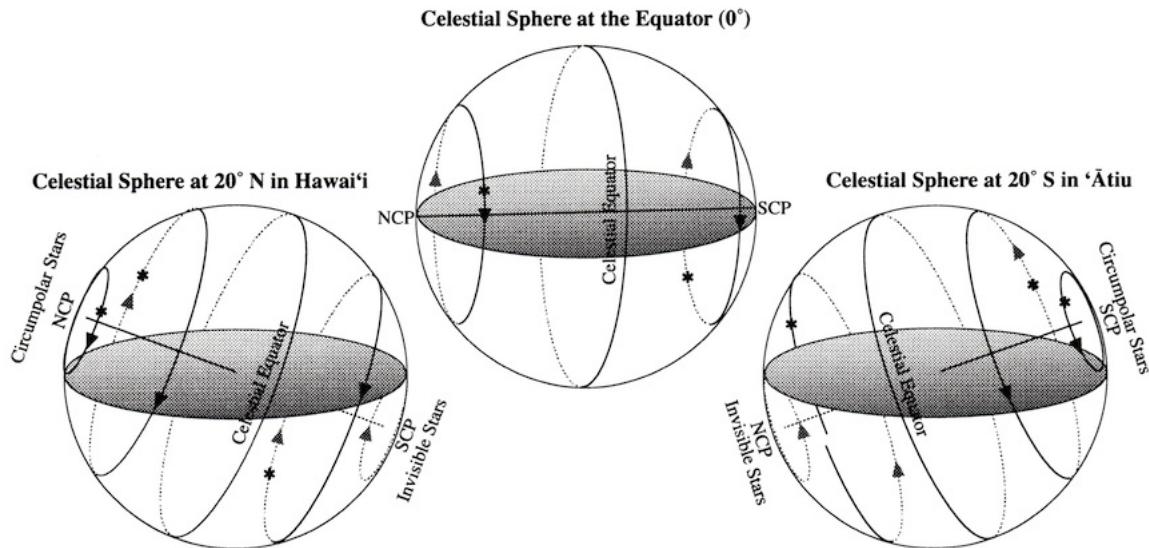


Figura 28: Campo Visible según latitud del observador

Partimos de la base que la esfera celeste posee un radio R , el cual es mucho mayor que el radio de la Tierra. Por esta razón puede considerarse al observador como situado en el centro de la Tierra el cual coincide con el centro de la esfera celeste (dado que es una proyección aparente). Se parte de la ecuación de la esfera celeste en coordenadas rectangulares

$$x^2 + y^2 + z^2 = R^2$$

El horizonte puede modelarse como un plano, el cual siempre pasa por el origen de coordenadas (observador)

$$ax + by + cz = 0$$

Para simplificar, tomo el plano paralelo al eje $x \Rightarrow a = 0$. Si se proyecta sobre el plano yz a la esfera celeste y se analiza. En la Figura 29 puede observarse esta proyección.

De la Figura 29 se puede ver que cuando:

$$y = -\frac{1}{b} \rightarrow z = -\frac{1}{c}$$

Además como:

$$\tan(\theta) = -\frac{c}{b}$$

Entonces:

$$y = -z \frac{c}{b} = z \tan(\theta)$$

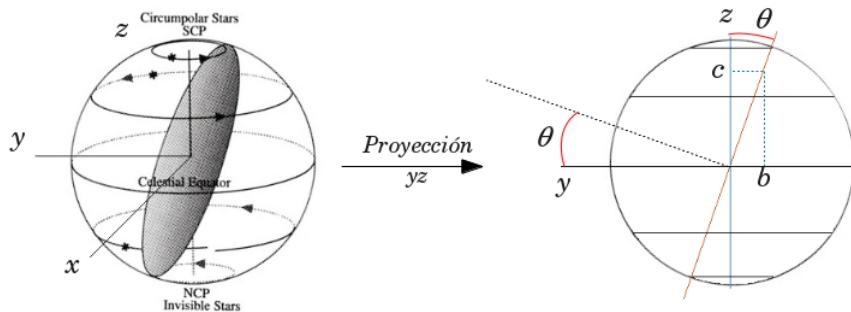


Figura 29: Esfera Celeste proyectada

El algoritmo intenta buscar los puntos límite del campo de visión. Los mismos son los que resultan de la intersección de la esfera con el plano, lo que genera una circunferencia. La ecuación de la misma es:

$$x^2 + z^2 \tan^2(\theta) + z^2 = R^2$$

Acomodando

$$x^2 + (1 + \tan^2(\theta))z^2 = R^2$$

Como:

$$(1 + \tan^2(\theta)) = \frac{1}{\cos^2(\theta)}$$

Entonces:

$$x^2 + \frac{z^2}{\cos^2(\theta)} = R^2$$

Si pasamos a coordenadas esféricas (se toma la transformación clásica), en la Figura 30 puede observarse como se miden las coordenadas en este sistema.

La transformación de coordenadas se basa en las siguientes expresiones

$$x = R \sin(\gamma) \cos(\varphi)$$

$$z = R \cos(\gamma)$$

Entonces transformando la esfera resulta:

$$R^2 \sin^2(\gamma) \cos^2(\varphi) + R^2 \frac{\cos^2(\gamma)}{\cos^2(\theta)} = R^2$$

Simplificando R^2

$$\sin^2(\gamma) \cos^2(\varphi) + \frac{\cos^2(\gamma)}{\cos^2(\theta)} = 1$$

Despejando el ángulo φ , se obtiene:

$$\cos^2(\varphi) = (1 - \frac{\cos^2(\gamma)}{\cos^2(\theta)}) \frac{1}{\sin^2(\gamma)}$$

Debe comprenderse, que sólo se aplicó una transformación de coordenadas que es semejante al sistema ecuatorial, pero si no se establecen los puntos de referencia adecuados, se estaría

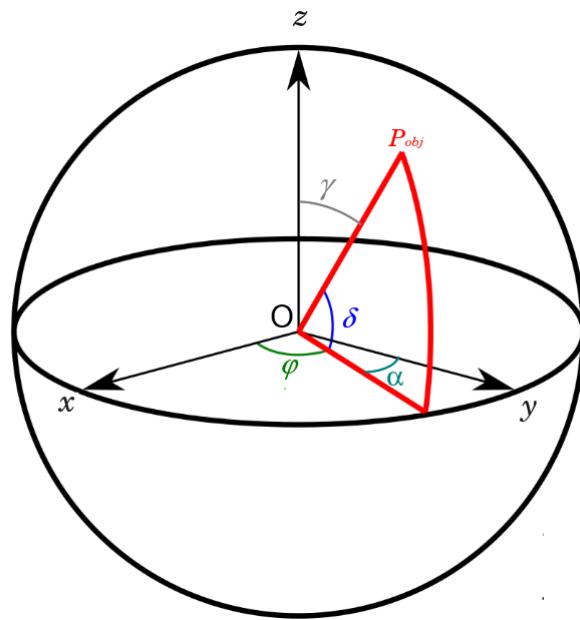


Figura 30: Esfera celeste en coordenadas esféricas

tomando mal la posición del objeto. Entonces si se mide la Ascensión Recta desde el meridiano local (α) (coincide con el eje y dado que el plano del horizonte es paralelo al eje x), entonces mido el complemento de φ . Luego la declinación (δ) se mide desde el plano ecuatorial (xy) por lo que es el complemento de γ . Entonces:

$$\operatorname{sen}^2(\alpha) = \left(1 - \frac{\operatorname{sen}^2(\delta)}{\cos^2(\theta)}\right) \frac{1}{\cos^2(\delta)}$$

Despejando la Ascensión recta, se obtiene lo que se busca:

$$\alpha = \pm \arcsin\left(\frac{1}{\cos(\delta)} \sqrt{1 - \frac{\operatorname{sen}^2(\delta)}{\cos^2(\theta)}}\right)$$

Esta ecuación es la que tiene el algoritmo para determinar si una coordenada esta dentro del campo visible. La misma entrega como parámetros la declinación del objeto (δ) y la latitud del observador (θ). Con esto calcula los valores de Ascensión recta límites (α). Si el objeto a visualizar tiene una coordenada de Ascensión recta (α_{OBJ}), medida desde el meridiano local, que se encuentra entre:

$$-\alpha \leq \alpha_{OBJ} \leq +\alpha$$

Entonces significa que el objeto está oculto por el plano del horizonte. Si no se cumple la desigualdad, el objeto es visible.

Por otro lado, existen condiciones de calculo que deben salvarse, por ejemplo:

$$\frac{\operatorname{sen}^2(\delta)}{\cos^2(\theta)} \geq 1$$

Esa condición haría que el resultado de la raíz sea imaginario. Esto se cumple para los elementos circumpolares, los cuales son siempre visibles.

Otra condición a salvar, es si el observador se encuentra en el polo ($\theta = \pm 90^\circ$), con lo que cualquier Ascensión Recta es visible si la Declinación es del mismo signo que la latitud y es mayor a cero.

La última condición a salvar es si el usuario desea ver el polo ($\delta = \pm 90^\circ$) en forma similar al caso anterior, la latitud debe ser del mismo signo que la Declinación que desea verse.

D. Tiempo Solar y Tiempo Sidéreo

El tiempo solar se mide por el movimiento diurno aparente del Sol y el mediodía local se define como el momento en que el Sol se encuentra en su cenit. Por definición, el tiempo que tarda el Sol en volver a su punto más alto es exactamente 24 horas, esto define un día tradicional como lo conocemos.

Sin embargo, las estrellas tienen un movimiento aparente ligeramente distinto. Durante el transcurso de un día, la Tierra se habrá movido un poco a lo largo de su órbita alrededor del Sol, por lo que debe girar una pequeña distancia angular extra antes de que el Sol alcance su punto más alto. En cambio las estrellas están tan alejadas que el movimiento de la Tierra a lo largo de su órbita genera una diferencia apenas apreciable con respecto a su dirección aparente (véase, en cualquier caso, paralaje), por lo que vuelven a su punto más alto en algo menos de 24 hs o día solar. Un día sidéreo medio ocupa alrededor de $23^h56^m04^s$ (es casi 4 minutos más corto que el día solar). Como se muestra en la 31, el tiempo transcurrido entre culminaciones sucesivas no es el mismo para el Sol que para las estrellas distantes. Al trasladarse la Tierra de B a C la estrella culmina de nuevo pero el Sol no, y se dice que retrasa el ángulo DCA que es lo que le falta para repetir su culminación. El tiempo correspondiente al arco BC es un tiempo sidéreo.

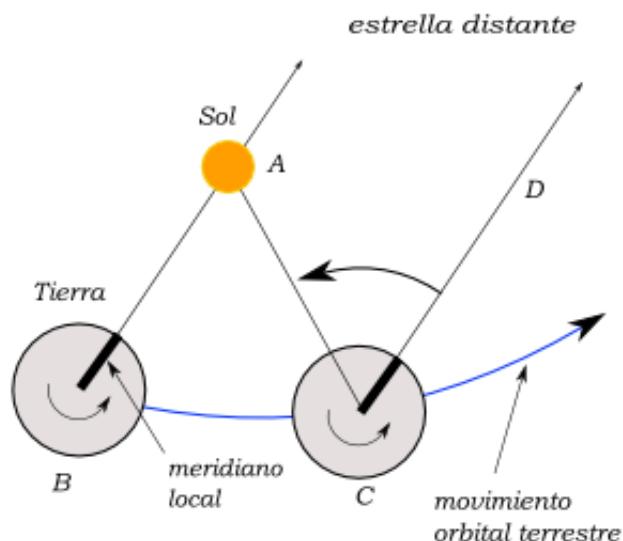


Figura 31: Desplazamiento de la Tierra: tiempo sidéreo y solar