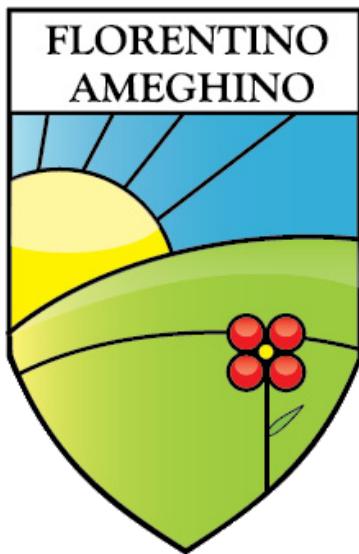


Escuela Secundaria Técnica Florentino Ameghino



**Instituto Argentino
de Radioastronomía**



Demostrador Tecnológico de un Sistema de Apuntamiento

Alumnos: Victoria Salomón, Nicolás Bahl, Franco Calcagno y Ludmila Clemente

Profesor: Tomás Arce

Curso: 7ºElectrónica

Tutor a cargo: Gastón Valdez

Año: 2022

Índice

1. Definición del proyecto	2
1.1. Posicionador para antena	2
1.2. Selección de hardware	2
1.2.1. Microcontrolador utilizado	2
1.2.2. Interfaz de red	3
2. Desarrollo de software	4
2.1. Interfaz de usuario	4
2.1.1. Software Gpredict	4
2.1.1.1. Protocolo de comunicación	4
2.1.1.2. Configuración de la antena en Gpredict	4
2.1.2. Software Stellarium	9
2.1.2.1. Protocolo de comunicación	9
2.1.2.2. Configuración del Stellarium	10
2.2. Software del microcontrolador	12
2.2.1. Diagrama del sistema	12
2.2.2. Esquema eléctrico de los componentes	13
2.2.3. Conexión del software con Gpredict y Stellarium	14
2.2.3.1. Conexión a la red mediante DHCP	14
2.2.3.2. Programación para conectarse con Gpredict	15
2.2.3.3. Programación para conectarse con Stellarium	16
2.2.3.4. Programación para comunicarse con la página web	17
3. Resultados obtenidos	18
3.1. Conexión con Gpredict	18
3.2. Conexión con Stellarium	19
3.3. Conexión con Página Web	20
4. Conclusiones	20
5. Bibliografía	22

1. Definición del proyecto

1.1. Posicionador para antena

El presente proyecto tiene como fin la realización de un demostrador tecnológico de un sistema de apuntamiento para la maqueta a escala de la antena del instituto IAR (Instituto Argentino de Radioastronomía).

La antena cuenta con dos motores, uno por cada eje de movimiento, altitud y azimut.

En el presente trabajo se va a desarrollar un sistema que sea capaz de realizar el movimiento de la misma, recibiendo la posición desde los programas utilizados. Estas coordenadas son las posiciones de los satélites.

En cuanto a los programas utilizados, se encuentran el Gpredict, el Stellarium y una página web, realizada por alumnos de informática, para el modo ingeniería del sistema.

1.2. Selección de hardware

1.2.1. Microcontrolador utilizado

De acuerdo a los requerimientos del sistema, el microcontrolador a utilizar debe tener las siguientes características:

- Cantidad de pines disponibles: 4 como mínimo
- Puerto SPI: conexión con el chip Ethernet

El hardware seleccionado es la placa de desarrollo Arduino Uno, la cual se basa en el microcontrolador ATMega328P. La placa está equipada con conjuntos de pines de E/S digitales y analógicas que pueden conectarse a varias placas de expansión y otros circuitos.

Sus características principales son las siguientes:

- Microcontrolador: Microchip ATmega328P6
- Voltaje de funcionamiento: 5 voltios
- Voltaje de entrada: 7 a 20 voltios
- Pines de E/S digitales: 14 (de los cuales 6 proporcionan salida PWM)
- Pines de entrada analógica: 6
- Corriente DC por Pin de E/S: 20 mA
- Corriente CC para Pin de 3.3V: 50 mA
- Memoria Flash: 32 KB de los cuales 0.5 KB utilizados por el gestor de arranque
- SRAM: 2 KB
- EEPROM: 1 KB
- Velocidad del reloj: 16 MHz

1.2.2. Interfaz de red

Uno de los requerimientos del sistema es que no debe utilizar redes inalámbricas por lo cual necesitamos un módulo ethernet. La placa disponible es el chip Ethernet W5100.

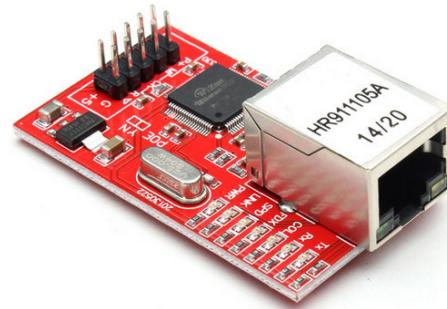


Figura 1. Chip Ethernet W5100

La misma utiliza el protocolo de comunicación SPI (Serial Peripheral Interface) y se programa mediante la librería Ethernet.h en el IDE de Arduino.

2. Desarrollo de software

2.1. Interfaz de usuario

2.1.1. Software Gpredict

Gpredict es una aplicación que permite rastrear y seguir la órbita de los satélites en el espacio en tiempo real. Se puede realizar el seguimiento de un número ilimitado de satélites y mostrar su posición y otros datos, en listas, tablas, mapas y diagramas polares. Gpredict basa su comunicación en la librería HAMLIB, la cual tiene su propio protocolo de comunicación.

2.1.1.1. Protocolo de comunicación

La librería HAMLIB a su vez utiliza el programa “rotctld” para el control de rotadores a través de sockets TCP. Dado que la mayoría de las operaciones de Hamlib tienen un método set y get, se utiliza una letra mayúscula para los métodos set , mientras que la letra minúscula correspondiente se refiere al método get .

Alguno de los comandos utilizados por el programa son los siguientes:

Métodos Get		Métodos Set	
Comando p	Obtener posición actual de la antena	Comando P	Enviar posición a la antena
Comando q	Desconectar la antena	Comando S	Detener la antena

Tabla 2. Comandos de rotctld usados por Gpredict

2.1.1.2. Configuración de la antena en Gpredict

Para realizar el seguimiento de satélites, el programa utiliza “módulos” en los cuales se puede seleccionar los satélites a seguir.

Para crear un módulo debe dirigirse a File → New module, como se muestra en la imagen.

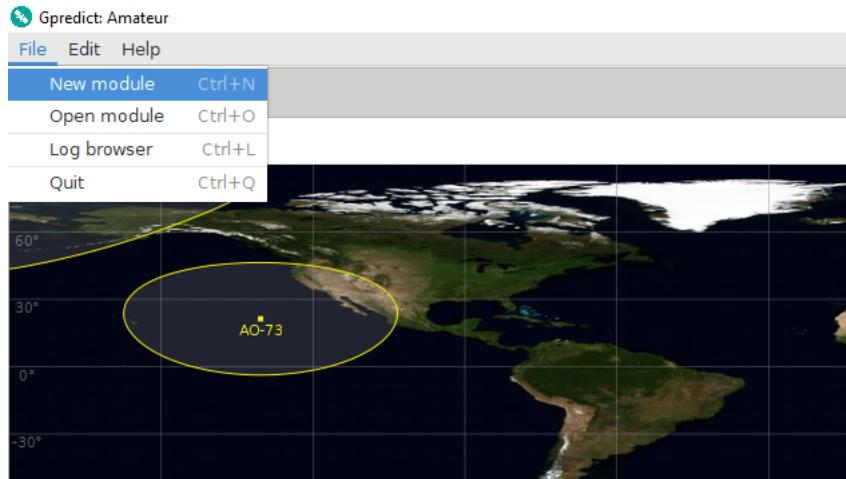


Figura 3.1. Creación de un módulo en Gpredict

Luego se abre una ventana, donde debe seleccionar los satélites a seguir, y nombrar el módulo. Además debe dirigirse a “Ground station” para configurar la latitud y longitud de la antena.

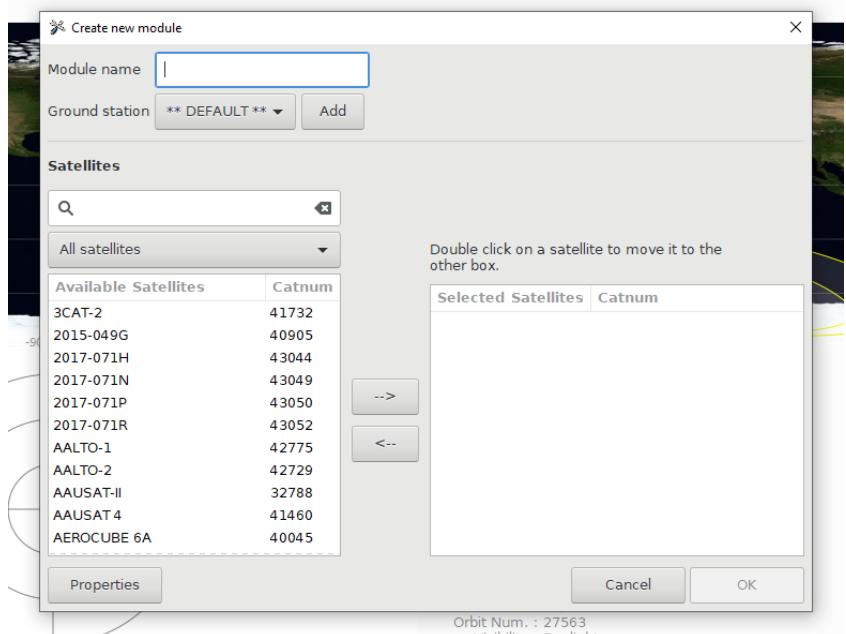


Figura 3.2. Creación de un módulo en Gpredict

Una vez que creó el módulo, debe configurar el rotador. Para ello debe dirigirse a Edit → Preferences. Al realizar esto se abre la siguiente ventana:

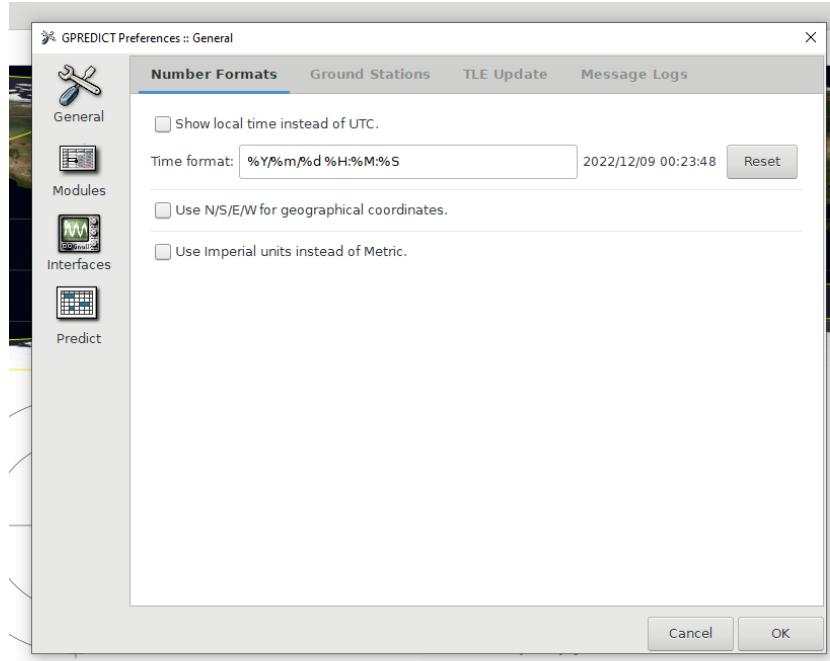


Figura 3.3. Configuración de la antena en Gpredict

Luego debe presionar donde dice “Interfaces” y luego en la pestaña “Rotators”.

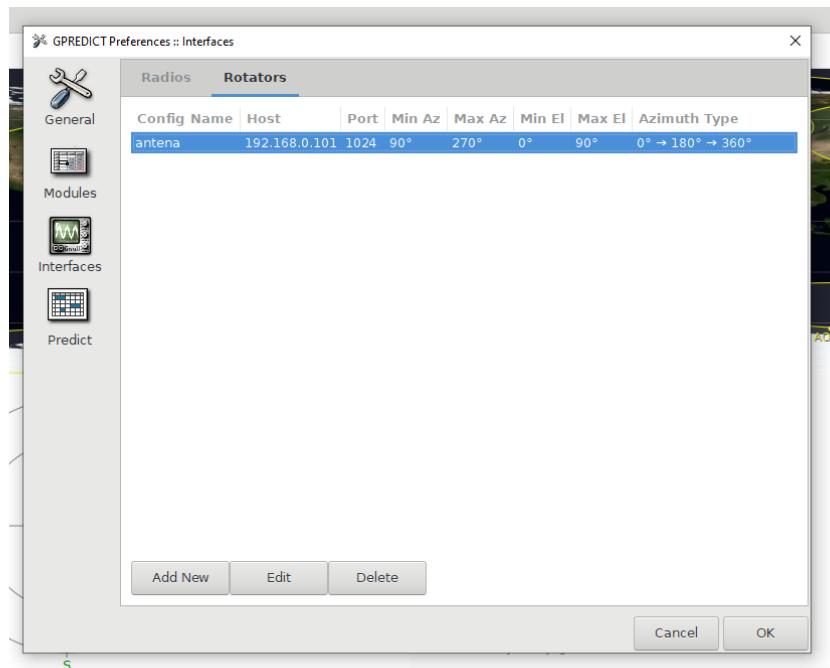


Figura 3.4. Configuración de la antena en Gpredict

Una vez ahí, debe presionar donde dice “add new” y se abre la siguiente ventana:

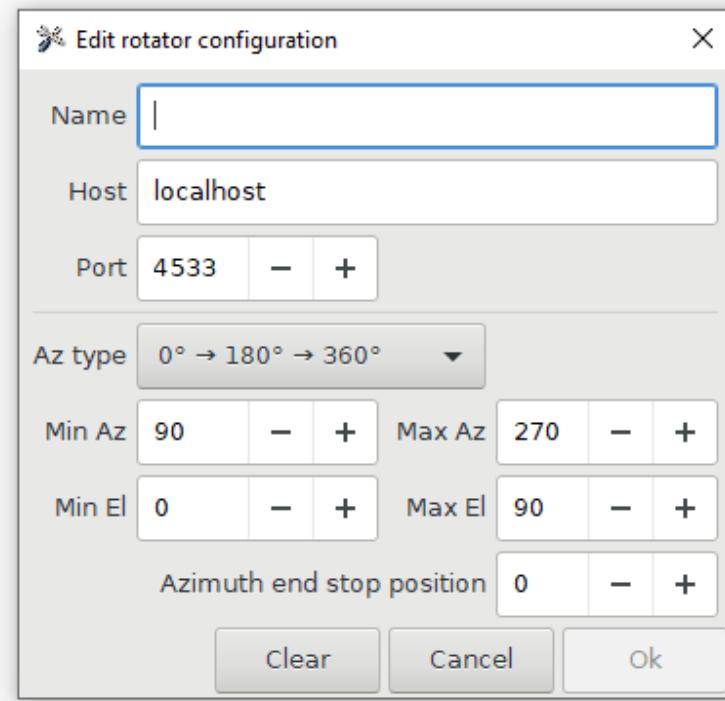


Figura 3.5. Configuración de la antena en Gpredict

Los parámetros a configurar son el nombre (elegido por el usuario), ángulos máximos y mínimos de la antena, el puerto a utilizar y donde dice “host” debe ir la IP del dispositivo. En nuestro caso utilizaremos el puerto 1024.

Por último, el Gpredict toma el eje azimutal en el polo norte geográfico (90°) y va aumentando en sentido horario hasta 360° , por lo que debe configurarse el eje azimutal entre 90° y 270° .

Luego de completar los datos debe presionar OK para que se guarde la configuración.

Una vez creado el rotador, debe dirigirse a la esquina superior derecha de la pantalla principal y presionar donde dice “antenna control”

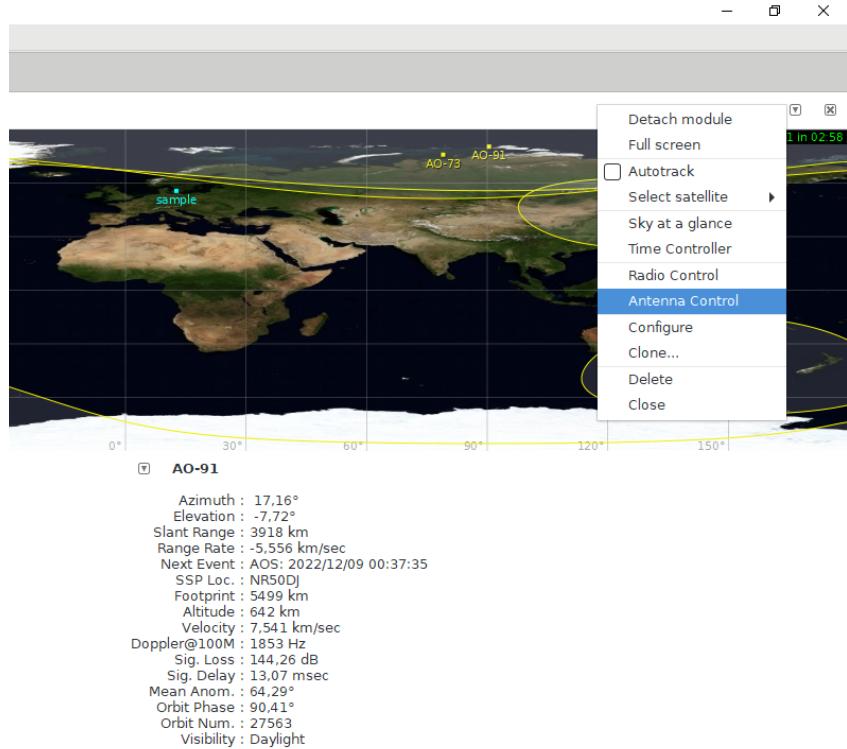


Figura 3.6. Ver el rotador creado

En la siguiente ventana se observan dos botones. Estos dos botones son “track” y “engage”. En el menú de track, realizamos el seguimiento automático. El menú engage, sirve para realizar la calibración de la antena.

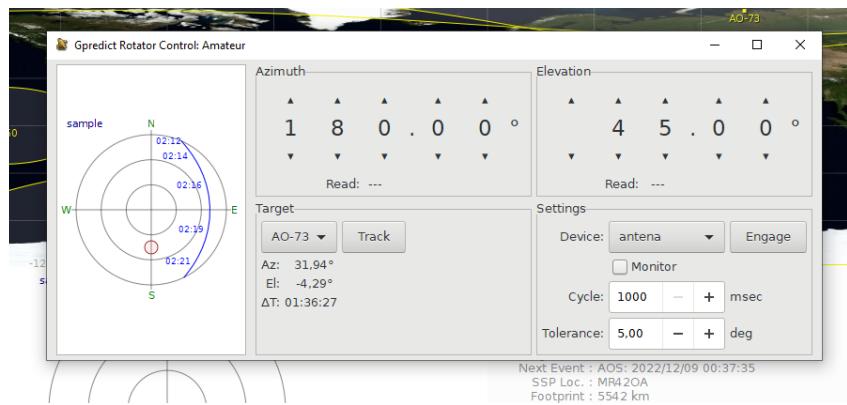


Figura 3.7. Panel de control

2.1.2. Software Stellarium

El software Stellarium, posee un seguimiento en tiempo real de estrellas, y está pensado para realizar el seguimiento usando telescopios. Los datos, pueden enviarse a través de la red, para que los reciba un telescopio, y realice el apuntado del telescopio. Además existen plugins que permiten enviar las coordenadas a dispositivos mediante la red.

2.1.2.1. Protocolo de comunicación

El Stellarium utiliza un programa denominado “telescope server”, el cual trabaja sobre TCP/IP. Los mensajes se basan en bytes, donde están agrupados del siguiente modo:

- LENGTH : 2 bytes - Indica la longitud total del mensaje
- TIME : 8 bytes. Tiempo UT a partir de 01/01/1970 en microsegundos. Actualmente en desuso
- RA: 4 bytes(sin signo) - ascensión recta:
- DEC : 4 bytes,con signo
- STATUS: 4 bytes con signo.

Por otro lado, internet o las redes, son protocolos Big-Endian, mientras que el microcontrolador atMEGA328p (arduino UNO) es Little-Endian. Big-Endian y Little Endian, se refiere al

modo de leer los datos dentro de un registro o posición de memoria dentro de un microprocesador. Por este motivo, los bits que llegan al microcontrolador, se deben reorganizar para que se interpreten correctamente.

2.1.2.2. Configuración del Stellarium

Para empezar se debe configurar la posición geográfica dentro del software, debido a que el apuntamiento de telescopios depende de la posición del observador en la tierra.

Para elegir la posición debemos oprimir el botón F6, donde se abrirá la siguiente ventana:



Figura 4.1. Configuración de ubicación en Stellarium

En esa ventana, debe seleccionarse, la latitud, longitud y altitud del lugar en el que se encuentra el telescopio o antena.

Por otro lado, para poder conectar el dispositivo con la PC, se debe configurar el puerto por el cual se realizará el intercambio de mensajes, y la dirección IP del dispositivo.

Para ello debe dirigirse a Configuración (F2) → Plugins → Control del telescopio.



Figura 4.2. Abrir el plugin de telescopio en Stellarium

Una vez en esta ventana, se debe presionar el botón “Configurar” y se abrirá la ventana que se muestra a continuación:



Figura 4.3. Añadir un telescopio en Stellarium

Luego debe presionar para añadir un nuevo dispositivo. Se abrirá una ventana donde deben elegirse los parámetros de configuración.

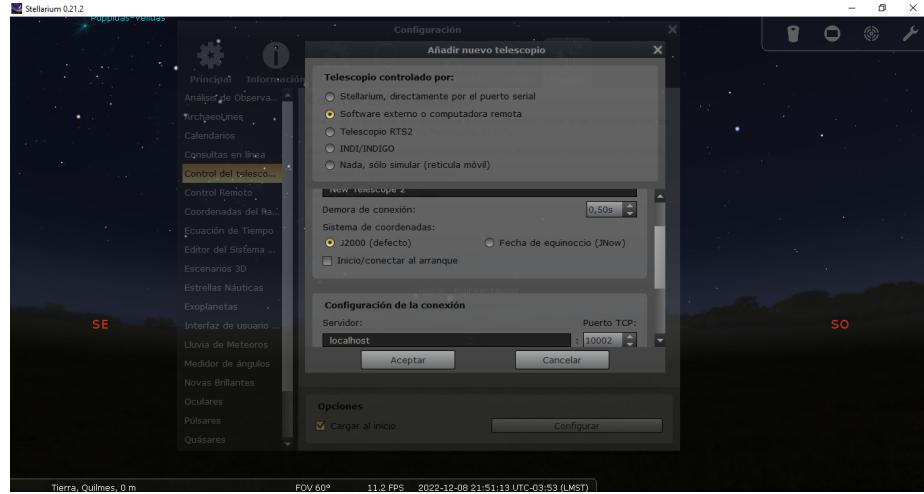


Figura 4.4. Configuración de red en Stellarium

Donde dice “localhost” debe introducir la IP del dispositivo, y donde dice “puerto TCP” debe seleccionar el puerto a utilizar. En nuestro caso utilizaremos el 10000 que es el utilizado por defecto por el Stellarium.

2.2. Software del microcontrolador

2.2.1. Diagrama del sistema

De acuerdo a los componentes seleccionados, estos deben interconectarse entre sí mediante sus protocolos de comunicación. El diagrama de bloques presente en la siguiente figura muestra cómo se compone el sistema de control.

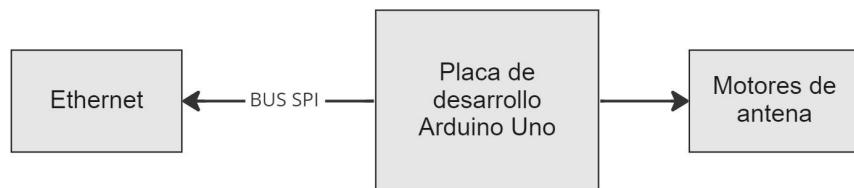


Figura 5. Diagrama en bloques general del sistema

El diagrama en bloques muestra el protocolo de comunicación utilizado por cada dispositivo. En el caso del módulo Ethernet, se comunica con la placa arduino mediante el bus SPI. A su vez, la placa arduino se comunica con los motores servo de la antena mediante un pin digital seleccionado por el usuario.

2.2.2. Esquema eléctrico de los componentes

En el esquema podemos ver cómo se conectan los componentes entre sí. En el caso del chip ethernet W5100, necesitamos utilizar los puertos SPI de la placa de desarrollo Arduino, los cuales son los pines 11 (MOSI), 12 (MISO) y 13 (SCK). El puerto de Slave Select (SS) y el Reset se seleccionan mediante software, en nuestro caso elegimos el pin 4 para el SS y el pin 3 para el reset.

Además del módulo ethernet, se encuentran conectados a la placa dos motores servo correspondientes a los ejes de la antena. Estos motores pueden ir conectados en cualquier pin digital de la placa arduino. En nuestro caso, el motor correspondiente a la altitud, lo conectamos al pin digital 9, y el correspondiente al acimut, en el pin digital 10.

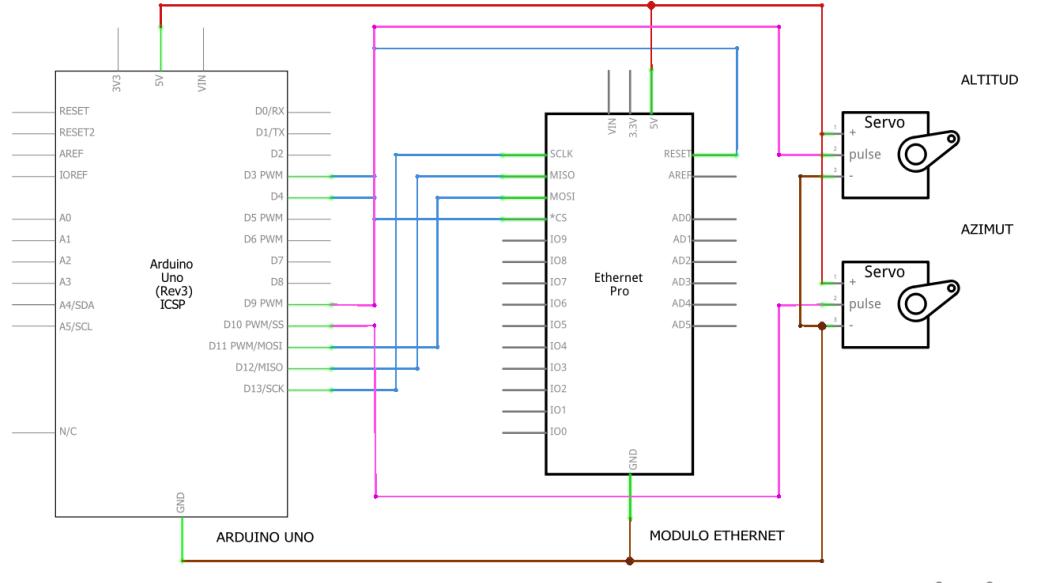


Figura 6. Esquema de conexión entre componentes

2.2.3. Conexión del software con Gpredict y Stellarium

En esta sección se muestran los componentes principales del software para comunicarse con estos programas.

2.2.3.1. Conexión a la red mediante DHCP

Para empezar, debemos conectar el chip w5100, mediante un cable UTP, a un switch o router. Luego, para obtener los parámetros de red utilizamos la librería “ethernet.h” que nos provee arduino.

Esta librería la utilizaremos para obtener la IP mediante el protocolo DHCP, como se muestra en el siguiente código.

```
#include <Ethernet.h>
#include <SPI.h>

#define PINSS 4
#define PINRESET 3
```

```

void setup() {

    Serial.begin(9600);
    Ethernet.init(PINSS);
    byte mac[] = { 0xBC, 0xCB, 0xAF, 0xAB, 0xAA,
    0x01 }; // dirmac elegida por el usuario
    if (Ethernet.begin(mac) == 0) {

        Serial.print(F("Fallo DHCP"));

    } else {
        Serial.println(Ethernet.localIP());
    }
}

```

Código 1. Obtención de IP mediante DHCP

Una vez que obtuvimos los parámetros, debemos configurar los programas con la IP obtenida.

2.2.3.2. Programación para conectarse con Gpredict

Se utiliza la librería “ethernet.h” vista anteriormente, para crear un objeto servidor con la siguiente sentencia.

```

#define PORT_GPREDICT 1024
EthernetServer gpr(PORT_GPREDICT);

```

Código 2. Definición del objeto servidor Gpredict

En donde se encuentra definido PORT_GPREDICT , es el puerto utilizado para la comunicación mediante el protocolo TCP/IP.

Luego, dentro del código principal, se realiza la recepción y envío de datos, como se muestra en el siguiente código.

```

EthernetClient cliente_g = gpr.available();
if (cliente_g) {

    while (cliente_g.connected()) {

        if (cliente_g.available()) {
            {
                char c = cliente_g.read() ;
            }
        }
    }
}

```

Código 3. Captura de datos dentro del bucle principal

Luego de la lectura de lo recibido, debe ir la acción a realizar según el tipo de comando recibido. Por ejemplo, supongamos que leemos “P 180,00 45,00”, esto significa que la antena debe apuntar a 180° en azimut y 45° en altitud. En cambio, si leemos una “p”, significa que debemos responder la posición de la antena al programa, supongamos que se encuentra en posición cenit (90° azimut y 90° altura), la respuesta será “125\n 39°\n”.

2.2.3.3. Programación para conectarse con Stellarium

Al igual que con el Gpredict, se debe crear un nuevo objeto servidor.

```

#define PORT_STELLARIUM 10000
EthernetServer stell(PORT_STELLARIUM);

```

Código 4. Definición del objeto servidor Stellarium

En la sección 2.1.2.1 se explica como envía los datos el Stellarium. De acuerdo a esta organización es que debemos guardar los datos recibidos.

Además, dado que los bits llegan desordenados (Big Endian y Little Endian), se deben reorganizar de la siguiente manera.

```
dec = 0x00000000 | (long (data[19])<<24) |
      (long (data[18])<<16) | (long
      (data[17])<<8) | (long (data[16])<<0);
RA = 0x00000000 | (long (data[15])<<24) |
      (long (data[14])<<16) | (long
      (data[13])<<8) | (long (data[12])<<0);
```

Código 5. Reorganización de datos

Luego, al igual que con el Gpredict, se realiza dentro del código principal, la recepción y envío de datos.

```
EthernetClient cliente_s =
stell.available();

if (cliente_s) {

    while (cliente_s.connected()) {

        if (cliente_s.available()) {

            }

        }
    }
}
```

Código 6. Captura de datos dentro del bucle principal

2.2.3.4. Programación para comunicarse con la página web

Los datos se envían desde la página web en un archivo tipo .json, por lo cual, para leerlos y poder manipularlos, debemos decodificarlos o deserializarlos. Para ello, se utilizaron las librerías que provee arduino “ArduinoJson.hpp” y “ArduinoJson.h”.

```

if (client.available())
{
    jsonp [i] = ((char)client.read());
    i++;
    String json = jsonp;
    StaticJsonDocument<300> doc;
    DeserializationError error =
deserializeJson(doc, json);
    if (error) {
        return;
    }
    Serial.println(jsonp);

    int azimuth = doc["x"];
    int altitud = doc["y"];

    Serial.println(azimuth);
    Serial.println(altitud);

}

```

Código 7. Decodificación de datos recibidos

3. Resultados obtenidos

En esta sección se comprueba la conexión con los dos programas utilizados en el proyecto (Gpredict y Stellarium), y con la página web diseñada por los alumnos de informática.

Todos los archivos de las programaciones se encuentran disponibles en un repositorio en GitHub. El enlace al proyecto es <https://github.com/victoriasalomon/antena>

3.1. Conexión con Gpredict

Una vez configurado el programa, debemos abrir el panel de control y oprimir el botón “engage” para calibrar la antena. Si se encuentra conectado, aparecerán las coordenadas enviadas al programa en donde dice “read”.

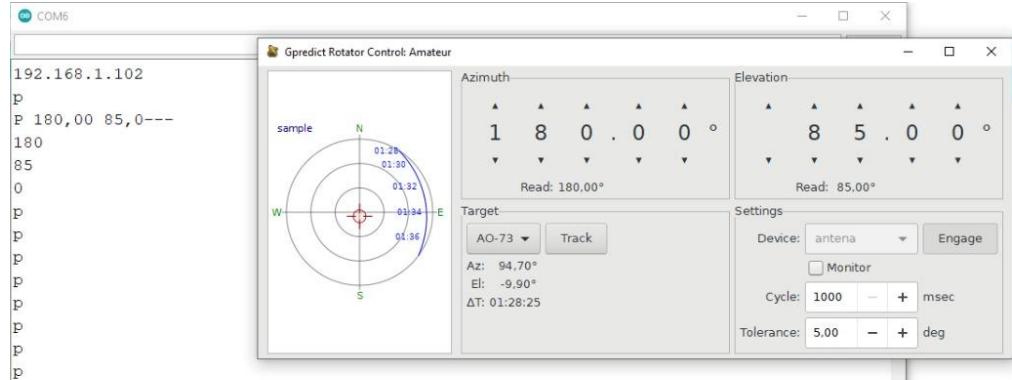


Figura 7. Coordenadas recibidas por el Gpredict

En nuestro caso, las coordenadas enviadas al programa fueron las mismas que recibimos. Como podemos observar, el programa recibió los datos correctamente.

3.2. Conexión con Stellarium

En cuanto al Stellarium, de acuerdo con el protocolo de comunicación utilizado, tuvimos que guardar los datos en distintas variables para poder manipularlos después. Para comprobar su correcto funcionamiento, se probó rotando el telescopio a unas coordenadas seleccionadas, y se verificó que el programa recibió los datos correctamente. Además, debido a que un dato que manda el stellarium es el tiempo unix, para poder comunicarnos debemos obtener el mismo desde la red, como se observa en la imagen.

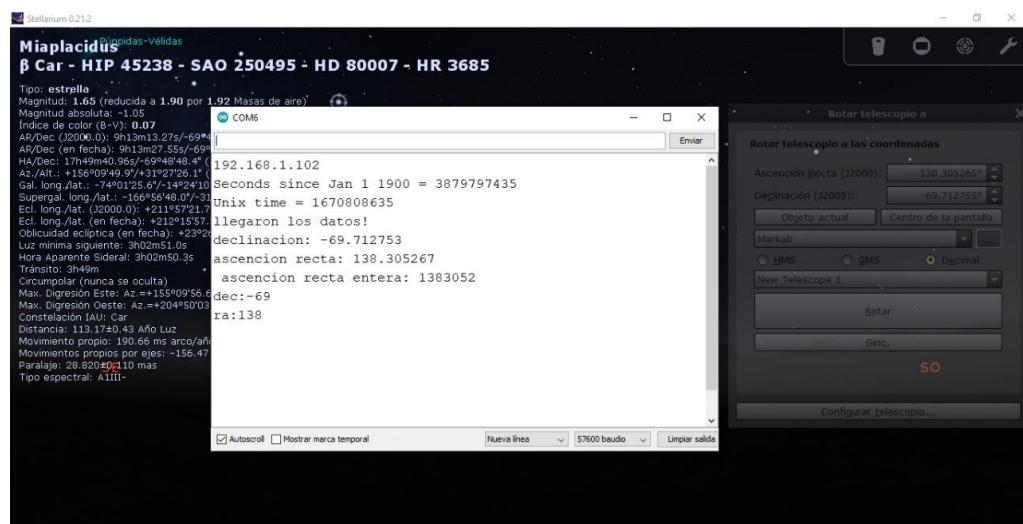


Figura 8. Coordenadas recibidas por el Stellarium

3.3. Conexión con Página Web

En el caso de la página web, luego de crear el servidor y conectarnos con la misma, pudimos comprobar que los datos llegaban correctamente y pudimos decodificarlos para luego poder trabajar con ellos.

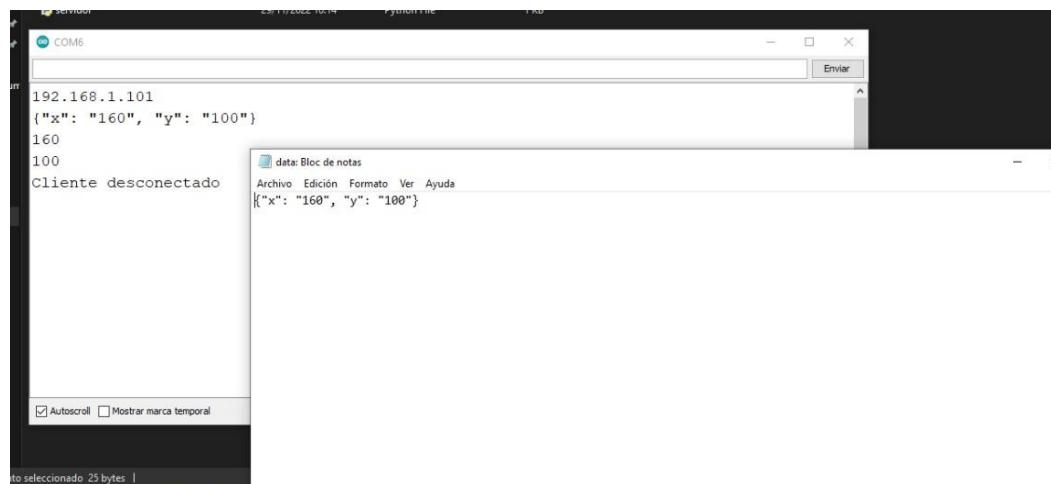


Figura 9. Coordenadas recibidas por la página

4. Conclusiones

Como conclusión de este proyecto, se logró realizar las conexiones entre el microcontrolador y los tres programas solicitados en este trabajo (Gpredict, Stellarium, Página Web). Además, en el caso de Stellarium, se realizó la transformación de coordenadas ecuatoriales locales a horizontales, necesaria para el direccionamiento de la antena.

Como trabajo a futuro, queda pendiente la unificación de los programas, con el fin de poder conectar la antena a tres programas diferentes sin la necesidad de cambiar la programación del microcontrolador.

En cuanto a saberes obtenidos, se destaca la utilización de Stellarium y Gpredict, la programación del módulo ethernet para la conexión con los programas, conocimientos en cuanto a coordenadas ecuatoriales y horizontales, y refuerzo de conocimientos en lenguaje C++ (Arduino).

5. Bibliografía

- Manual de usuario Stellarium, disponible en <https://stellarium.org/>
- Documentación telescope server, disponible en
http://svn.code.sf.net/p/stellarium/code/trunk/telescope_server/stellarium_telescope_protocol.txt
- Documentacion sobre rotctld, disponible en <https://www.mankier.com/1/rotctld>
- Documentación sobre librerías de Arduino, disponible en
<https://www.arduino.cc/reference/en/>