Escuela Superior Politécnica del Litoral

Facultad de Ingeniería en Electricidad y Computación

Integración y optimización de la Configuración e Interfaces del Bote Batimétrico Autónomo del CTDI en ESPOL

Proyecto Integrador

Previo la obtención del Título de:

Ingeniero en Electrónica y Automatización

Presentado por:

Deriot Santiago Zambrano Coronel

Guayaquil - Ecuador

Año: 2023-2024

Dedicatoria

Me gustaría dedicar este documento primero que nada a mi abuela Mariana Estupiñán Nieves que siempre ha sido mi pilar en quien yo me suelo reflejar bastante y quien admiro por la fuerte mujer, madre, abuela, tía y un enorme etcétera que ella es, y a mi abuelo Proscopio Zambrano Cedeño por todo el cariño y apoyo que me ha demostrado en mi vida.

También a mis padres Andrea Coronel Reyna y Ruben Zambrano Estupiñán como al resto de mi familia y amigos quienes me han alentado siempre a superarme y quienes me proporcionaron un alivio y comodidad en mi vida universitaria para que yo disfrute y aprenda en el mejor ambiente posible.

Agradecimientos

Un enorme agradecimiento a todas las personas que me ayudaron a poder cumplir con la responsabilidad de este proyecto, pero principalmente a la Ing. Doménica Barreiro Palacios, al Ing. Jorge Vulgarin Punguil y al Ing. Jorge Bravo Lino. Me gustaría agradecer también al Ph.D. Daniel Ochoa por otorgarme la oportunidad de trabajar como ayudante en sus instalaciones, lo cual, influyo mucho en mi decisión de querer ser parte de los estudiantes que hayan logrado aportar algo en el desarrollo de los equipos del laboratorio, lo que dio paso solicitar realizar mi tesis en el CTDI.

Declaración Expresa

Yo *Deriot Santiago Zambrano Coronel* acuerdo y reconozco que la titularidad total y exclusiva sobre los derechos patrimoniales de patente de invención, modelo de utilidad, diseño industrial, información no divulgada y cualquier otro derecho o tipo de Propiedad Intelectual que corresponda o pueda corresponder respecto de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, incluyendo cualquier derecho de participación de beneficios o de valor sobre titularidad de derechos, pertenecerán de forma total, perpetua, exclusiva e indivisible a LA ESPOL, sin limitación de ningún tipo. Se deja además expresa constancia de que lo aquí establecido constituye un "previo acuerdo", así como de ser posible bajo la normativa vigente de transferencia o cesión a favor de la ESPOL de todo derecho o porcentaje de titularidad que pueda existir.

Sin perjuicio de lo anterior el alumno firmante de la presente declaración recibe en este acto una licencia de uso gratuita e intransferible de plazo indefinido para el uso no comercial de cualquier investigación, desarrollo tecnológico o invención realizada durante el desarrollo de su trabajo de titulación, sin perjuicio de lo cual deberán contar con una autorización previa expresa de la ESPOL para difundir públicamente el contenido de la investigación, desarrollo tecnológico o invención.

Así también autorizo expresamente a que la ESPOL realice la comunicación pública de la obra o invento, por cualquier medio con el fin de promover la consulta, difusión y uso público de la producción intelectual.

Guayaquil, 2 de octubre del 2023.

Deriot Zambrano Coronel

Evaluadores

JONATHAN ABRAHAM AVILES CEDENO
I have reviewed this document

Firmado electrónicamente por:
DENNYS DICK CORTEZ ALVAREZ

Dennys Dick Cortez Alvarez

Profesor de Materia

Jonathan Abraham Avilés Cedeño

Tutor de proyecto

Resumen

El proyecto se centra en la integración del sistema de control del bote batimétrico y los softwares que lo gestionan en un dispositivo portátil. Se aborda la comprensión del módulo transmisor y receptor del control remoto 3DR X8+ y el protocolo SBUS, con el objetivo de crear una interfaz electrónica que simplifique el manejo del bote para usuarios no especializados y se amplíe sus aplicaciones. Algunas de las herramientas utilizadas incluyeron un convertidor de señal SBUS a USB para analizar la señal del control, una aplicación para Android que permitió la recepción de instrucciones y envió de datos, y el uso de ventanas flotantes en teléfonos modernos. La interfaz electrónica establece la comunicación inalámbrica vía Bluetooth entre el gamepad y el teléfono, el teléfono lee y envía las instrucciones a un Arduino UNO vía USB, este interpreta las instrucciones y recrea una señal en el protocolo SBUS, enviándola al bote a través de un transmisor de radio al receptor que está conectado al controlador, mientras la antena de telemetría recopila la información de navegación que se visualiza en el programa Mission Planner. Se logró comprender como se envían los comandos de navegación, reproducir los comandos de maniobra, replicar la señal en protocolo SBUS, y reducir el tiempo de puesta en marcha del bote.

Palabras Clave: Bote batimétrico, Telemetría, Control, SBUS, Arduino.

Abstract

This project focuses on integrating the control system of the bathymetric boat and its management software into a portable device. This involves understanding the transmitter and receiver modules of the 3DR X8+ remote control and the SBUS to create an electronic interface that simplifies boat operation for non-expert users and broadens its applications. The approach utilized an SBUS to USB signal converter for analyzing control signals, an Android app for signal reception and management, and floating windows on modern smartphones for data visualization and instruction transmission. The electronic interface enables wireless communication via Bluetooth between the gamepad and the phone. The phone then reads and sends instructions to an Arduino UNO via USB. The Arduino interprets the instructions and recreates a signal in the SBUS protocol and sends it to the boat through a radio transmitter to the receiver connected to the controller. Meanwhile, the telemetry antenna collects navigation data displayed in the Mission Planner program. The understanding of navigation commands, reproduction of maneuver commands, replication of the signal in the SBUS protocol, and reduction of the boat's startup time were successfully achieved.

Keywords: *Bathymetric boat, Telemetry, Control, SBUS, Arduino.*

Índice general

Abreviaturas

ESPOL Escuela Superior Politécnica del Litoral

CTDI Centro de Transformación Digital Industrial

RC Radio Controller (Controlador de radio)

USV Unmanned Surface Vehicle (Vehículo de superficie no tripulado)

GPS Global Positioning System (Sistema de posicionamiento global)

ESC Electronic Stability Control (Control de estabilidad electrónica)

PWM Pulse Width Modulation (Modulación de ancho de pulso)

PCB Printed Circuit Board (Placa de circuito impreso)

OTG On-The-Go

USB Universal Serial Bus (Bus universal en serie)

Control remoto Radio control de la marca de dron 3DR X8+

Simbología

bit	Numero binario 0 o 1
s	Segundo
byte	Conjunto de 8 bits
ms	Microsegundos
MHz	Megahercio
mm	Milímetro
Bd	Baudio

Índice de ilustraciones

Índice de tablas

Índice de planos

Capítulo 1

1.1 Introducción

La continua mejora de los equipos de captura de datos del Centro de Transformación Digital Industrial (CTDI) es de alta importancia ya que se utilizan en proyectos de investigación para el desarrollo de nuevas tecnologías.

Además, el bote batimétrico ha demostrado su utilidad capturando datos batimétricos en la piscina de la ESPOL haciendo uso de su sonar, con gran precisión en posicionamiento gracias a la tecnología GPS con RTK, y ahora se busca mejorar su puesta en marcha y manejo operativo.

La mejora del sistema de maniobra y comunicación del bote autónomo permitirá su manejo por usuarios sin conocimientos técnicos especializados y de esta forma atraer también a una gama más amplia de clientes, desde investigadores académicos hasta empresas privadas relacionadas con construcciones marinas.

1.2 Descripción del problema

La configuración actual del bote batimétrico autónomo del Centro de Transformación Digital Industrial (CTDI) presenta dificultades operativas al necesitar un computador portátil y un control remoto, cada uno con su respectivo módulo de radio para su operación. Esta complejidad consume tiempo que se puede reflejar en dinero al ralentizar las futuras pruebas de campo del bote. También, debido a que los componentes electrónicos se encuentran sueltos dentro de la carcasa del bote, existe ruido en los datos de posicionamiento de sus GPS al moverse el bote, y por último, se requería que los propulsores tengan una mayor potencia.

1.3 Justificación del problema

El correcto funcionamiento del bote batimétrico autónomo es esencial para obtener mediciones precisas y confiables, especialmente en investigaciones y aplicaciones técnicas. La configuración actual del bote del Centro de Transformación Digital Industrial (CTDI) podría no ser intuitiva debido que el control remoto está hecho para manejar un dron y los softwares utilizados para la puesta en marcha del bote pueden resultar difíciles de entender, lo cual podría traducirse en una mala manipulación por parte del usuario.

Este tipo de inconvenientes no solo limita que los usuarios puedan hacer uso del bote, sino que también puede implicar costos adicionales en términos de tiempo y recursos en cada operación. Por lo tanto, es imperativo abordar y rectificar estas dificultades, haciendo uso de software más amigable para su puesta en marcha además de un mejor arreglo y sujeción de los componentes del bote para evitar que en movimientos bruscos, resultado del cambio de motores más potentes, se afecten los datos de posicionamiento.

1.4 Objetivos

1.4.1 Objetivo general

Centralizar los sistemas de configuración y control del bote autónomo a través de una interfaz inalámbrica de control de navegación, para disminuir tiempo de puesta en marcha y facilitar el uso operativo del bote.

1.4.2 Objetivos específicos

1. Analizar los protocolos y señales de los actuales módulos de radiocontrol identificando los comandos de navegación y control del sistema embebido del bote.
2. Diseñar la interfaz electrónica que facilite la conexión del módulo de radiocontrol a dispositivos digitales, como tabletas, mediante comunicación serial.
3. Implementar la nueva interfaz de usuario basado en software realizando pruebas comparativas, evaluando la eficacia, facilidad de uso y precisión de la nueva configuración en relación con el sistema anterior.

1.5 Marco teórico

1.5.1 Vehículo de Superficie No Tripulado

El Vehículo de Superficie No Tripulado (USV) (ilustración 1) es un tipo de vehículo que opera en la superficie marina sin transportar personas y se ha vuelto popular desde la Segunda Guerra Mundial debido a sus aplicaciones militares, científicas y civiles. Puede ser autónomo o controlado remotamente, lo que amplía su alcance de uso. Su estructura incluye un armazón resistente al agua, cascos estables y un sistema de control que procesa información y toma decisiones [11].

Ilustración 1 Vehículo de superficie no tripulado

Una estación terrestre permite el control remoto y la operación autónoma del USV. El sistema de posicionamiento GPS se utiliza para determinar la ubicación. Un sensor batimétrico mide la profundidad del fondo marino, y el sistema de alimentación proporciona energía a través de baterías. El sistema de comunicación utiliza antenas de telemetría, un transmisor y un receptor para conectar el vehículo al sistema de control. Finalmente, el sistema de propulsión incluye motores sin escobillas manejados por un controlador de velocidad electrónico (Electronic Speed Controller, ESC). El ESC utiliza modulación por ancho de pulso (Pulse Width Modulation, PWM) para controlar la energía enviada al motor (ilustración 2) [11].

Ilustración 2 Componentes del Vehículo de Superficie No Tripulado

1.5.2 Gamepad iPega PG-9083S

Este controlador de juegos es compatible con la reproducción de juegos en tabletas y teléfonos inteligentes Android e iOS habilitados para Bluetooth. Los botones del reproductor multimedia "volumen +, -", "anterior", "siguiente" y "reproducir/pausa" se pueden utilizar en dispositivos con sistema operativo Android [8]. El gamepad de la ilustración 3 es el que remplaza al control remoto del bote batimétrico ya que los periféricos que tiene son suficientes para ejecutar las instrucciones que el bote necesita.

Ilustración 3 Gamepad

1.5.3 Módulos de comunicación serial

Un puerto serie constituye una interfaz de comunicación de datos digitales que se utiliza con frecuencia en computadoras y dispositivos periféricos. En esta interfaz, la información se transmite secuencialmente bit por bit, transmitiendo un solo bit a la vez, en contraste con los puertos paralelos que envían varios bits simultáneamente [4]. El módulo de transmisor DJT (ilustración 4) de FrSky utiliza comunicación serial para actualización de firmware, configuración de alarma y visualización de datos de telemetría que se podían observar en el control remoto. El transmisor DJT es el que se encarga de transmitir la señal SBUS hacia al receptor, quien se la transmite finalmente al controlador.

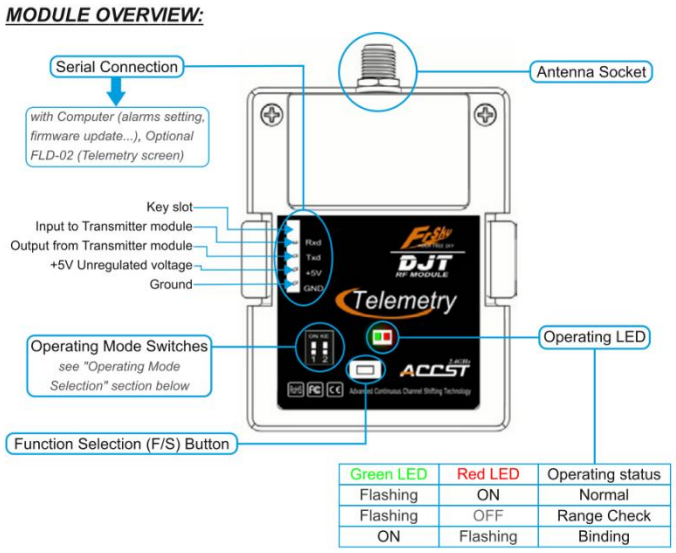


Ilustración 4 Características del módulo transmisor DJT [10]

1.5.4 Protocolo S-Bus

Conocido bajo el nombre de SBUS o Bus Serie, este protocolo es ampliamente empleado por Futaba y FrSky. Permite la transmisión de hasta 16 canales a través de un único cable de señal. Es importante tener en cuenta que la señal SBUS en los receptores de FrSky se encuentra invertida [12]. El S-Bus es el protocolo que utiliza el receptor X8R (ilustración 5) que es traducido a USB por el convertidor y poder analizar las instrucciones del control remoto y el gamepad después.

Ilustración 5 Receptor X8R

1.5.5 Circuito de inversión serial

Cumple la función de invertir una cadena de bits o datos binarios. Este tipo de circuito toma una entrada digital y produce una salida con los niveles lógicos invertidos, es decir, los bits de 0 se convierten en 1 y viceversa. Dado que la señal S-Bus del receptor X8R está invertida [13]. Se debe realizar el circuito de la ilustración 6 para que los datos que entren en la placa STM32F1 del convertidor sean los correctos.

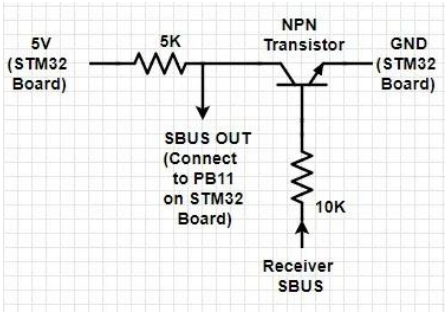


Ilustración 6 Esquema del circuito de inversión serial para el convertidor de SBUS a USB [13]

1.5.6 Arduino UNO

Arduino es una placa electrónica de hardware de código abierto que incorpora un microcontrolador reprogramable y una serie de conectores que facilitan la conexión de sensores y actuadores, en su mayoría mediante cables Dupont [1]. La variedad de herramientas que ofrece la interfaz de software de Arduino IDE para manipulación de datos y generación de señales con una velocidad de reloj de 16 MHz permiten que el Arduino UNO (ilustración 7) sea un componente capaz de trabajar con las instrucciones que envía el gamepad.

Ilustración 7 Arduino UNO R3

1.5.7 Placa de desarrollo STM32F1

La gama de microcontroladores STM32F1 de ST, abarca una amplia gama de aplicaciones en los sectores industrial, médico y de consumo, satisfaciendo diversas necesidades en estos mercados [15]. La placa de desarrollo se muestra en la ilustración 8 y es crucial para la construcción del convertidor SBUS a USB, ya que se encarga de leer los datos que le llegan del receptor X8R y enviarlos a la computadora para analizar el comportamiento de la señal del control remoto y del gamepad.

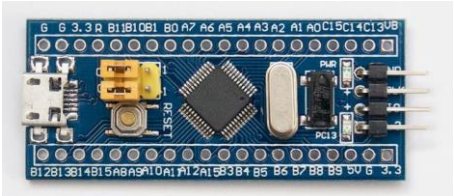


Ilustración 8 Placa de desarrollo STM32F1

1.5.8 FTDI232I chip

El chip FTDI232, desarrollado por la empresa Future Technology Devices International (FTDI), es un controlador de interfaz USB a UART que desempeña un papel crucial en la comunicación entre dispositivos electrónicos y computadoras mediante el estándar USB [16]. El chip de la ilustración 9 es el encargado de formatear la placa de desarrollo STM32F1 para que funcione como convertidor de protocolo S-Bus USB.

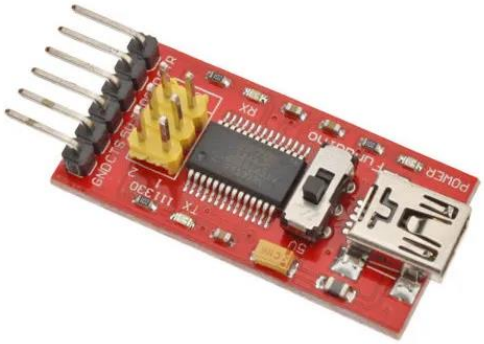
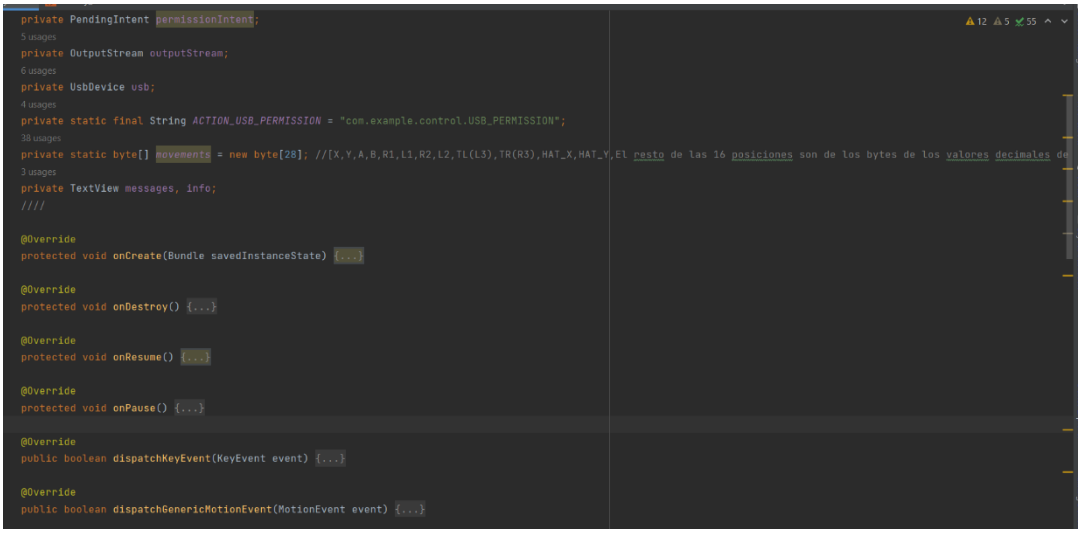


Ilustración 9 FTDI Chip

1.5.9 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) diseñado específicamente para la creación de aplicaciones Android. Utiliza el lenguaje de programación Java o Kotlin y se integra con el Android Software Development Kit (SDK), facilitando la implementación de funcionalidades complejas y la adaptabilidad a las constantes actualizaciones del sistema operativo Android [6]. El uso de las funciones (ilustración 10) para interpretar las interacciones del usuario con el gamepad son las que permiten leer las instrucciones de los botones y de los valores de los ejes de los joysticks, para después dependiendo de estos valores programar el traductor y que ejecute las instrucciones del control remoto con el gamepad.



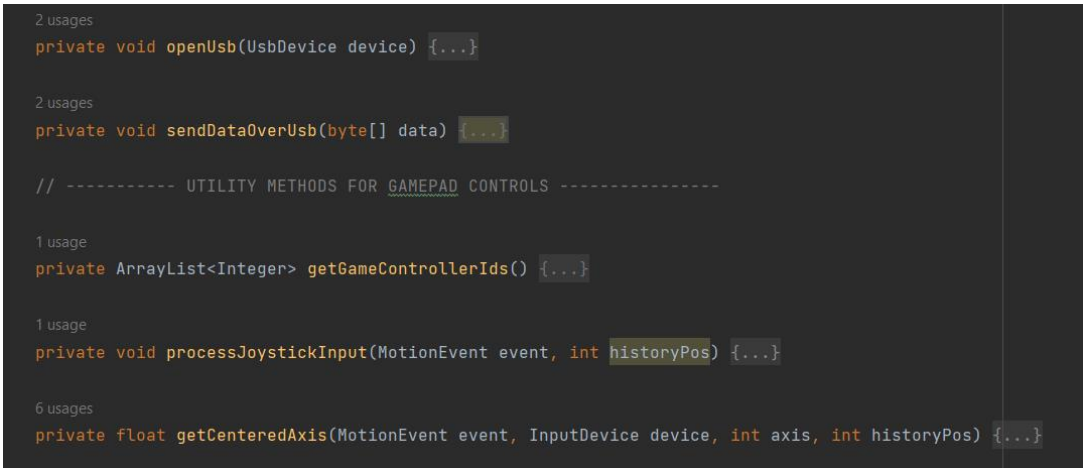


Ilustración 10 Funciones utilizadas en Android Studio

1.5.10 Mission Planner

Mission Planner es una aplicación de software de código abierto que permite a los usuarios planificar rutas de vuelo, establecer puntos de referencia, monitorear la telemetría en tiempo real y realizar ajustes en la configuración del vehículo [2]. A través de la aplicación de Mission Planner (ilustración 11) se comprueba que varios factores importantes al poner en marcha el bote como es modo de maniobra del bote ya sea manual o automático, si los GPS se encuentran en buen estado, si el bote batimétrico está posicionado correctamente con respecto a la base (APÉNDICE B), si la batería tiene carga, entre otras características.

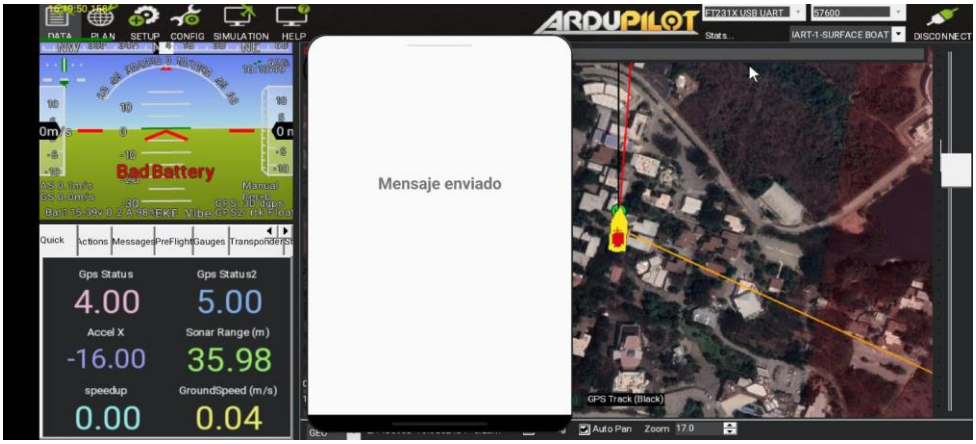


Ilustración 11 Programas Mission Planner y Control ejecutandose en paralelo

Capítulo 2

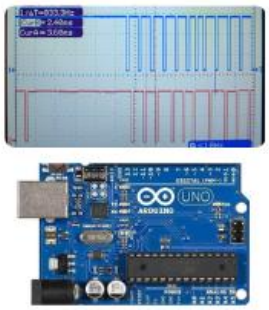
2.1 Metodología.

1) Formatear la placa STM32F1 con un adaptador serial FTDI

2) Realizar un Convertidor SBUS a USB

3) Analizar la señal del control remoto

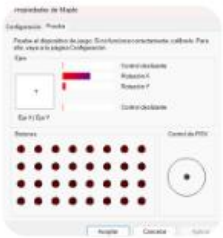
4) Replicar la señal de instrucciones del control remoto



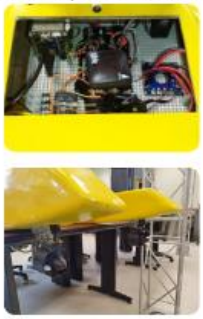
5) Leer las instrucciones del gamepad

6) Relacionar las instrucciones del gamepad con las del control remoto

7) Comprobar que ambos controles ejecuten la misma acción por la misma instrucción



8) Fijar los componentes electrónicos y cambio de motores



9) Pruebas de toda la interfaz electrónica

Ilustración 12 Resumen del desarrollo del proyecto

En la ilustración 12 se muestra de forma general como fue el desarrollo del presente.

Previamente, el bote se manipulaba con el control remoto para las instrucciones manuales y armado/desarmado (habilitar o deshabilitar los motores de las turbinas respectivamente) del mismo, utilizando el transmisor DJT para mandar estas instrucciones al receptor que a su vez está conectado al controlador de bote.

El problema principal fue adaptar el transmisor del control remoto al gamepad y que las señales que este envía se puedan comunicar con el receptor, ya que el hardware del transmisor está fabricado para un modelo en específico del control remoto de la marca de dron 3DX8+ [14]. Al final se obtuvo una mezcla entre ambas opciones conectando por bluetooth el gamepad y el teléfono, luego las instrucciones del gamepad que llegan a la tableta se envían por cable vía USB haciendo uso de un módulo OTG (ilustración 15) hacia el traductor, que replica la señal enviada por el transmisor al receptor, y finalmente al controlador que maneja el bote y la antena de telemetría se conectan a la tableta por medio de un cable separador tipo c (ilustración 13) para así poder visualizar el envío y recepción de la información del bote batimétrico en Mission Planner.

Con respecto a la parte electrónica se realizaron nuevos diseños de la ubicación y dimensión de los componentes del bote dado que se los ordenó y compactó de mejor forma en una plancha de plástico (ilustración 14) y con una distribución distinta dentro del bote. También se cambiaron los motores de propelas por unos de mayor potencia que permitieron una mayor velocidad del bote.

Ilustración 13 Separador de tipo C macho a 2 tipos c hembra

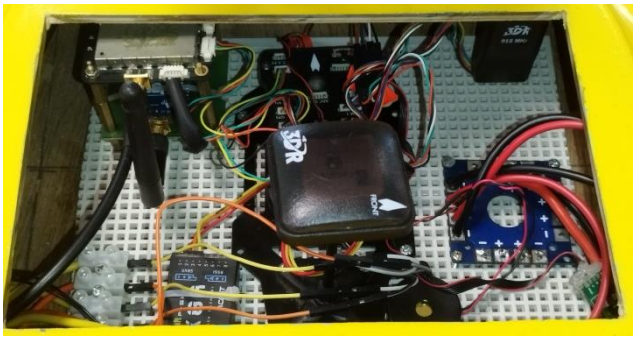


Ilustración 14 Componentes fijados en una plancha de plástico

2.1.1 Comunicación entre los componentes

Las comunicaciones entre los componentes incluyen señales inalámbricas y conexiones por cable. El controlador es el que se encarga de recibir las instrucciones, como se muestra en la ilustración 15. La estación de tierra permite el uso manual del bote en el programa creado en Android Studio llamado “Control”, haciendo usos de los joysticks. También permite realizar las maniobras de ruta automáticas que se envían por las antenas de telemetría con un software de la marca del controlador del dron llamado “Mission Planner”. Este mismo software provee información visual sobre el estado del sistema de posicionamiento o la cantidad de satélites que hayan receptado la base que es el equipo que configura la posición inicial del bote batimétrico (APÉNDICE B) [11].

El conjunto de estación de tierra, traductor y transmisor son los elementos involucrados en la maniobra del bote. Las instrucciones que manda el gamepad vía bluetooth son receptadas en la tableta para ser reenviadas por cable USB tipo AB, utilizando un módulo OTG [9] en el periférico del teléfono, las cuales llegan al Arduino UNO donde son codificadas a una señal que el receptor asimila y que llega a través del transmisor al bote.

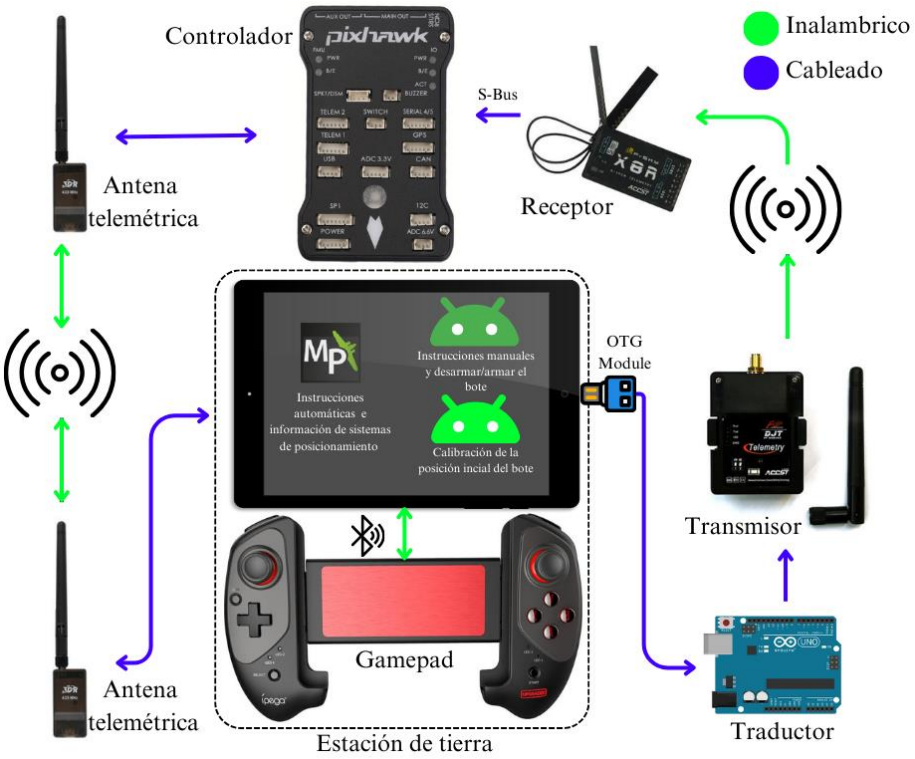


Ilustración 15 Esquema de conexiones de los componentes de maniobra y comunicación

2.1.2 Replicar la señal del control remoto

Para lograr replicar la señal se realizó una investigación acerca del modelo 3DX8+ del control remoto, el protocolo de sus instrucciones y sus puertos de comunicación. Para ello, se usó el osciloscopio colocando las puntas de prueba en los pines que se ubican en la parte trasera del control remoto (ilustración 16). Se descubrió que por el primer pin se envía de forma constante una trama con forma PWM de 17 pulsos altos separados entre ellos por un tiempo de 300us (ilustración 17).

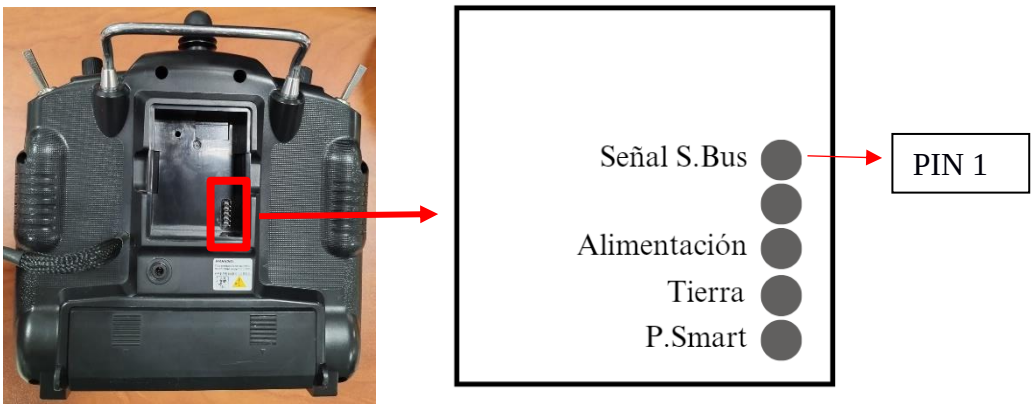
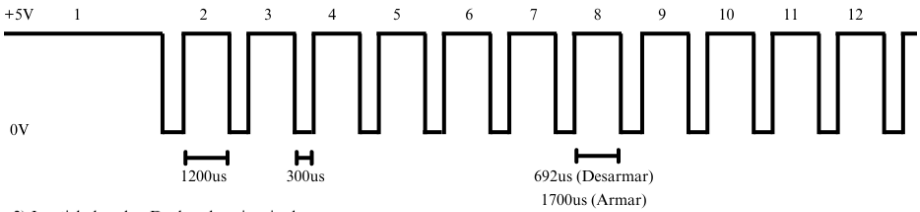


Ilustración 16 Descripción de los pines de la parte de atrás del control remoto antiguo [14]

Partiendo de la función de interrupciones de Arduino, que ocurren cuando existe un flanco positivo o uno negativo entre los pulsos, se empleó un programa capaz de obtener un vector de números enteros que representan el tiempo en microsegundos de cada uno de los 17 pulsos altos separados entre ellos por un tiempo de 300us.



2) Joystick derecho: De derecha a izquierda
3) Joystick derecho: De arriba a abajo
4) Joystick izquierdo: De arriba a abajo
5) Joystick derecho: De derecha a izquierda

8) X/Y: Armar/Desarmar el controlador

Ilustración 17 Señal PWM que sale del PIN 1 del control remoto

Como muestra la ilustración 17 los pulsos del 2 al 5 son los que se modifican al manipular los joysticks del control remoto. También se aprecia que el pulso 8 es el que se encarga de armar o desarmar el controlador y que dependiendo de qué instrucción se quiera mandar el ancho de pulso es diferente.

En base a eso se modificó el vector de tiempo en microsegundos (ilustración 18), para que los valores en sus respectivos pulsos cambien cuando se esté realizando una u otra acción.

```
// Vector de tiempo en microsegundos que recrea la señal SBUS y se modifica dependiendo del decimal de la informacion de los Joystciks del Gamepad
volatile unsigned int ConfG[] = {14800, 300, 1200, 300, 1200, 300, 1200, 300, 1200, 300, 1024, 300, 688, 300, 692, 300, 1200, 300, 1200, 300, 1200,
```

Ilustración 18 Vector de tiempo en microsegundos para recrear la señal del control remoto

Las señales valueZ, valueRZ, valueY, y valueX son números flotantes entre 1 y -1 que envía el gamepad al teléfono vía bluetooth al manipular los joysticks, y son los responsables en modificar el tiempo del vector en sus respectivas posiciones 2, 4, 6, 8 como se muestra en la ilustración 19. Además, se muestra como la señal de armado y desarmado del bote depende del valor que tenga el buffer (vector de bytes) que le llega al Arduino desde el teléfono (ilustración 20). Este contiene los datos de todas las instrucciones que se hayan generado al manipular el gamepad.

Con esta información ya se dio paso a replicar la señal del control remoto y que así el gamepad realice las mismas acciones con las mismas instrucciones.

```
// Actualizar el vector ConfG según los nuevos valores de los Joysticks
ConfG[2] = valueZ * 480 + 1200;
ConfG[4] = valueRZ * 480 + 1200;
ConfG[6] = valueY * -480 + 1200;
ConfG[8] = valueX * 480 + 1200;

//Condicion para armar o desarmar el bote usando los botones X = ARMAR e Y = DESARMAR
if (int (buffer[0]) == 1){
  ConfG[14] = 1700;
}else{
  ConfG[14] = 692;
```

Ilustración 19 Parte del código del Arduino que modifica el vector de tiempo

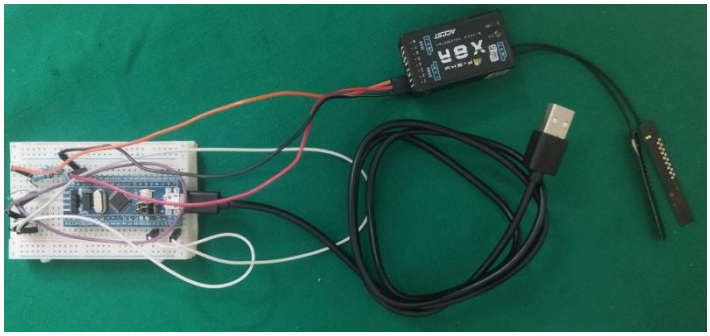
```
void loop() {
  // Se lee el serial al momento que llega el bufer de 28 bytes que contiene la información de todas las funciones del Gamepad
  if (Serial.available() > 0) {
    Serial.readBytes(buffer, sizeof(buffer)); // Se lee el  array de bytes que tiene la información de las acciones que se esten realizando en el gamepad
```

Ilustración 20 Sección de código para la lectura del vector de bytes que llega al Arduino desde el teléfono.

2.1.3 Convertidor de una señal SBUS a USB

Para poder analizar si la señal replicada es correcta, se construyó un circuito con el microcontrolador STM32F1 (ilustración 21), que permite realizar la conversión de una señal SBUS a una USB. Con el chip FTDI232 se instaló el firmware "SBUSJoystick Initial reléase" en la placa STM32F1 con el software "Flash loader demonstrator" [13]. También, debido a que la señal SBUS es "inverted" [13], se construyó un pequeño circuito inversor serial para que el microcontrolador pueda leer los datos S-Bus del receptor, procesarlos y codificarlos al protocolo USB.

Al conectar el convertidor al computador, el receptor X8R mostrará una conexión exitosa con el transmisor por el correspondiente led indicador en color verde [5]. Se analizó la rotación en X, en Y y en el movimiento del "+" al manipular los joysticks del control remoto en la ventana de "Propiedades del control" (ilustración 22) predeterminado que ofrece Windows y se las comparó con la señal replicada del Arduino, dado que en base a esto se plantearon las fórmulas de la ilustración 19 [13].



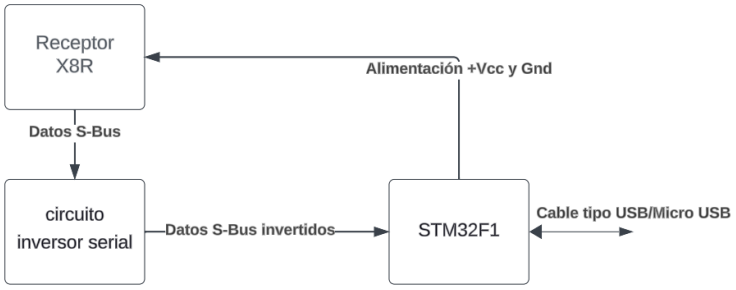


Ilustración 21 Convertidor de la señal SBUS a USB

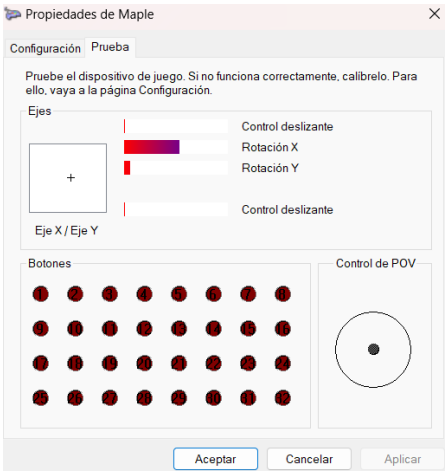


Ilustración 22 Interfaz en la que se reflejan las instrucciones realizadas con el control remoto o con el gamepad [13]

2.1.4 Leer la señal del gamepad a través de la señal bluetooth

Para comunicarse con el bote usando protocolo SBUS, y dado que el gamepad se comunica vía bluetooth con celulares Android, se programó una aplicación móvil que permite leer las instrucciones bluetooth del gamepad que llegan a la tableta y visualizarlas en pantalla para así poder comprobar que las instrucciones de los botones y joysticks del gamepad sean las correspondientes a las ya conocidas con el control remoto. Después, esas instrucciones son enviadas vía USB con un cable tipo AB desde el teléfono usando un módulo OTG conectado al puerto serial del Arduino. La información que se envía es un vector de bytes en el que cada byte o conjunto de bytes son valores de movimiento de los joysticks o de los botones presionados en el gamepad [9].

Para programar esto se usó el programa de Android Studio Giraffe que permite crear una aplicación móvil usando funciones propias del programa como el procedimiento de permisos de acceso con el celular, comunicación de transmisión/emisión en el protocolo USB, verificación e identificación de dispositivos conectados (en este caso ya sea un “joystick” o “gamepad”) y la respectiva calibración de los joysticks [5].

En la ilustración 23 se muestra la interfaz de usuario de la aplicación “Control”. En la parte superior se muestran los valores de los ejes X, Y, Z, RZ de los dos joysticks, cuál se está usando y en qué dirección, por ejemplo, el joystick derecho (RJ), dirección de movimiento general “DOWN” (abajo) y los valores correspondientes de los ejes. También en caso de presionar un botón se mostrará en pantalla cuál se presionó. Y, por último, existe un mensaje para alertar de dos posibles situaciones. Primero, si la aplicación detecta al Arduino y se realiza una instrucción desde el gamepad mostrará “Mensaje enviado”. Segundo, en caso de no detectar al Arduino y se intenta enviar alguna instrucción mostrará “Serial Null”.

Con esto se pudo validar el envío de la señal por el cable USB al Arduino al momento de realizar alguna acción con el gamepad, ya que todo se muestra en pantalla.

Ilustración 23 Entorno de la aplicación para leer la señal del gamepad

2.1.5 Cambio del equipo de propulsión

Para mejorar la potencia mecánica del bote batimétrico se le han agregado nuevos motores (ilustración 24). Se cambiaron los antiguos motores sin escobilla por unos más modernos y con mejores prestaciones. Los nuevos motores tienen turbinas con una mayor potencia de salida que permitió que el bote se mueva más rápido.

Los antiguos motores tenían una potencia de 140W [11], mientras que los nuevos tienen una potencia máxima de 312W [17].

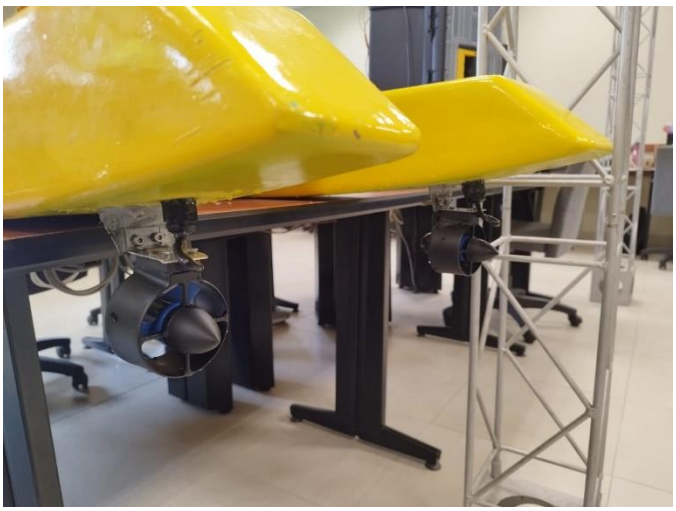


Ilustración 24 Nuevos motores con placas de acero inoxidable colocados en el bote

2.1.6 Construcción de Soporte

Con el objetivo de que todo el sistema de maniobra esté unificado se debe tener los componentes que involucran la interfaz electrónica a la mano. Para manipular e inicializar los componentes se diseñó una estructura (ilustración 25) para contener el Arduino, el transmisor de radio, y la antena de telemetría y el celular o tableta.

La estructura se realizó con una impresora 3D, la parte superior (color azul) tiene unas dimensiones de 102x200 mm y la parte inferior (color verde) tiene unas dimensiones de 102x186 mm. Las pequeñas columnas verdes son las encargadas de conectar las dos piezas al atornillarlas con la pieza azul.

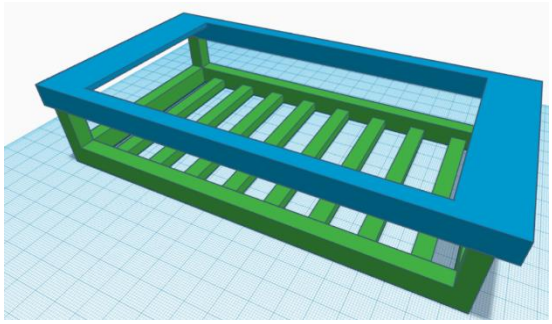


Ilustración 25 Modelo que se adhiere al gamepad

Capítulo 3

3.1 Resultados y análisis

El análisis del funcionamiento de los joysticks, botones y periféricos del control remoto permitió entender y comprender cómo se comunica con el receptor y cómo las instrucciones enviadas interactúan con el controlador del bote batimétrico.

Se usaron varias herramientas tanto de hardware como de software para que al momento de interactuar con los joysticks y botones del gamepad, poder visualizar esas instrucciones y compararlas con las instrucciones que manda el control remoto al maniobrar el bote.

Estas herramientas requerían tener un dispositivo que permitiera la observación de las funciones del control remoto y el gamepad sin necesidad de tener que involucrar al bote en el proceso, como el convertidor de la señal SBUS a USB. También se programó una aplicación para celular en Android Studio que permitió la recepción y visualización de las instrucciones del gamepad, para posteriormente ser enviada a un Arduino que captaba el vector de bytes de esas instrucciones vía USB y en base a ellas permitía control el bote en modo manual.

Para poder visualizar los dos softwares en paralelo (Mission Planner y Control), se hizo uso de la función de ventanas flotantes de los teléfonos Android. Esto era necesario para poder observar los datos e instrucciones automáticas enviadas con “Mission Planner” y el envío de instrucciones manuales con la aplicación creada “Control”. “Mission Planner” permanecía en el plano principal, mientras que la aplicación “Control” se encontraba disponible en cualquier momento que se necesite

3.2 Comparación de la señal del control remoto y el gamepad

En el proceso de la creación de la señal del control remoto se intentó usar la función de escritura digital (digitalWrite) y la función de interrupción de microsegundos (delayMicroseconds) de Arduino IDE, pero estas daban problemas porque interferían con el periodo de lectura de la señal que llegaba vía USB. Para solucionar este inconveniente se tuvo que independizar el proceso de lectura y creación de la señal haciendo uso de los comparadores y temporizadores propios del Arduino UNO (ilustración 26).

```
//Configurar Timer1
cli();  // Deshabilitar interrupciones globales
TCCR1A = 1 <<  COM1A0 | 0 << COM1B0 | 0 << WGM10;
TCCR1B = 1 << WGM12 | 5 << CS10;
TCNT1 = 0;
OCR1A = 16000;
TIMSK1 |= (1 << OCIE1A);
sei();  // Habilitar interrupciones globales
```

```
//Se recrea la señal SBUS para controlar el bote batimétrico, la señal se ve por el pin digital 9
ISR(TIMER1_COMPA_vect) {
  OCR1A = ConfG[idx]/64; // Se lo divide para 64 porque calculando la frecuencia con la que se leen los datos ese el valor minimo de tiempo en microsegundos
  idx ++;
  if(idx >= 34){
    idx = 0;
    }
}
```

Ilustración 26 Sección de código del Arduino que se encarga de replicar la señal

Para verificar que la señal replicada se parezca a la que envía el control remoto, primero se realizó una revisión visual utilizando las puntas de prueba del osciloscopio. Se ubicó una punta de prueba en la salida SBUS del control remoto y la otra en la salida digital 9 que es por donde se envía la señal generada por la función “OCR1A”. Finalmente se compararon las señales y se verificó que sean lo más parecidas posibles.

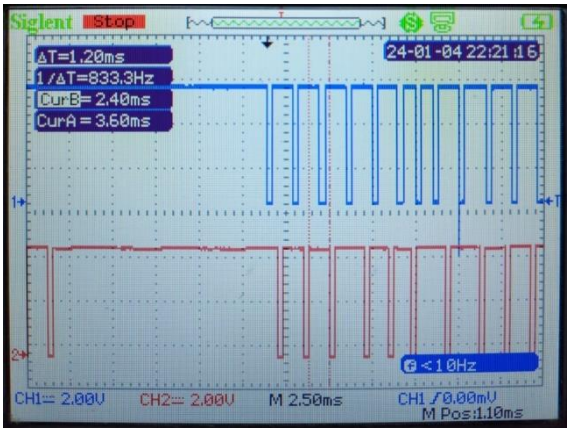


Ilustración 27 Imagen de las señales del control remoto y el gamepad en el osciloscopio cuando no se está enviando ninguna instrucción

La ilustración 27 muestra la señal que sale del Arduino (color azul) y la del control remoto (color rojo). Se pueden observar en las señales los cuatro primeros segmentos altos de 1200µs que representan los joysticks. El octavo segmento de ambas señales es el que desarma/arma el controlador y tiene un tiempo de 692µs, siendo ambas señales muy parecidas entre sí. Se distingue que el tercer segmento de la señal roja es un poco más ancho, pero a niveles prácticos ambos segmentos ejecutan la misma maniobra. Es importante recalcar que ambas señales están en un estado neutro.

Al enviar la señal azul al transmisor se verificó que se podía manipular el movimiento de los motores al alterar los cuatro primeros segmentos y armar/desarmar el controlador mediante el octavo segmento.

3.3 Funcionamiento de los motores con los joysticks

Para realizar una comparación del movimiento que tendrán los motores al ejecutar las mismas instrucciones tanto en el control remoto como en el gamepad, se utilizó el convertidor de la señal SUB a USB. Se tomaron como referencia 8 instrucciones que representan posiciones de puntos extremos de los joysticks en el control remoto (ilustración 28) y para ver que representaban esas configuraciones se usó la ventana de "Propiedades del control" que está disponible al conectar el convertidor a la laptop.

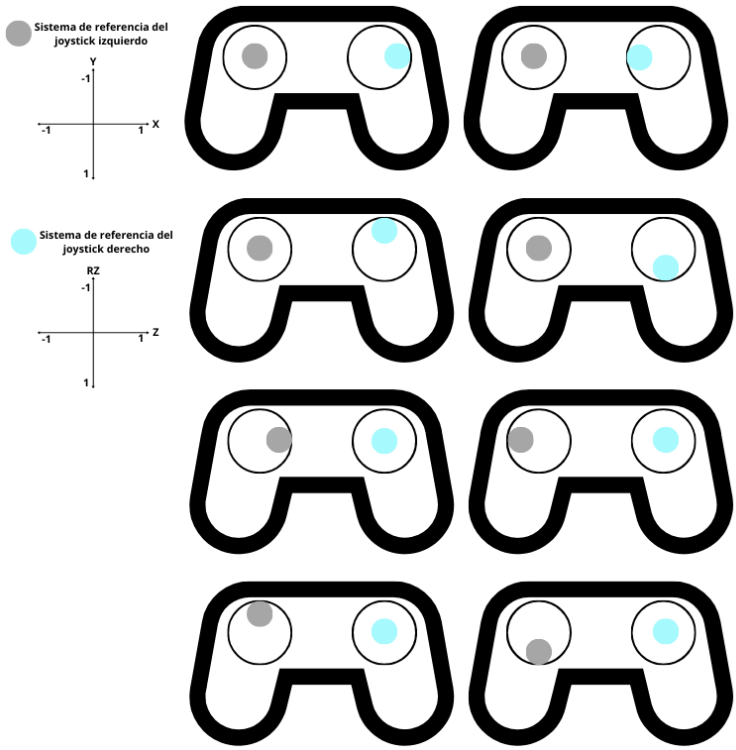


Ilustración 28 Imagen de las 8 instrucciones que se probaron en el control remoto y en el gamepad

Se observó el comportamiento de la rotación en X y en Y, y en el movimiento del "+" en las "Propiedades del control" y se realizó las modificaciones en el código del Arduino para que las acciones del gamepad sean las mismas.

3.4 Funcionamiento de la acción de habilitar o deshabilitar los motores

El controlador tiene la función de armarse (habilitar) o desarmarse (deshabilitar), que es la que permite que al momento de mover los joysticks o presionar un botón esas instrucciones se ejecuten o no.

Debido a que esta acción se habilitaba o deshabilitaba por medio de un switch en el control remoto, se usaron 2 botones del gamepad para ejecutar esa acción. El botón “X” se usó para armar y poder maniobrar el bote y el botón “Y” para desarmar (ilustración 29) y que no se pueda realizar ninguna acción. Además, para saber si el bote esta armado o desarmado hay dos indicadores, uno visual en la aplicación de Mission Planner y el segundo es que el controlador emite un sonido cuando se cambia de estados.

Ilustración 29 Botones XYAB de la parte derecha del control

3.5 Funcionamiento de los componentes de maniobra con el gamepad y comunicación

El gamepad en conjunto con la tableta/teléfono trabajan con componentes diferentes tanto de software como de hardware y se tenía que verificar que todo funcione de forma correcta antes de poder realizar las pruebas en agua (APÉNDICE B).

Se debió tener en cuenta que los componentes de maniobra y de telemetría operan con diferentes velocidades de transmisión de datos (baudios Bd) y ambos usan aplicaciones diferentes por lo que se verificó que no exista una interferencia entre ellos o entre las señales de telemetría.

Ilustración 30 Componentes de maniobra del bote conectados

En la ilustración 30 se encuentra los componentes antiguos y nuevos que involucran la puesta en marcha del bote (APÉNDICE B) conectados y en funcionamiento, ya que al momento de enviar instrucciones manuales o automáticas al controlador del bote no presentó ningún inconveniente.

3.6 Prueba de campo con los componentes de maniobra y comunicación del bote

Se llevó a cabo pruebas de campo para verificar que los componentes del bote batimétrico funcionen (ilustración 31), dado que en el proceso de creación del convertidor de una señal SBUS a USB (ilustración 21) y de la fijación de los componentes del bote en una plancha de plastico (ilustración 1) se manipularon componentes como el receptor X8R y el receptor Piksi fuera del bote y se desoldó la placa de separación de voltaje.

Ilustración 31 Prueba de campo del bote con el gamepad y los componentes de maniobra y comunicación

3.6.1 Implementación del soporte

El soporte facilita la puesta en marcha del bote al tener los componentes cohesivos, lo que también permite la capacidad de trasportar todo el sistema de maniobra de forma segura al proporcionar estabilidad y compactibilidad (ilustración 32).

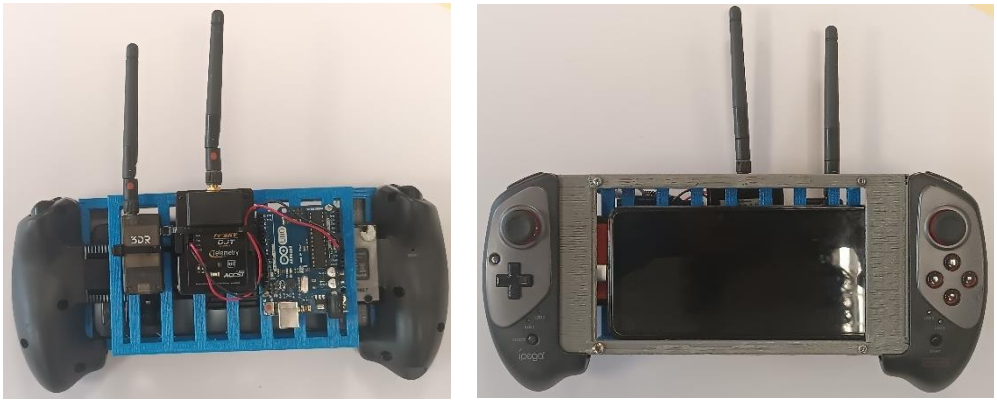


Ilustración 32 Soporte armado en el gamepad con componentes de maniobra y comunicación

3.6.2 Uso de la aplicación Geo-Locator (APÉNDICE B)

Antes de poder maniobrar el bote se debe inicializar su posición, para que al momento de encenderlo exista comunicación entre la base (ilustración 33) y el receptor Piksi que se encuentra en el bote y se muestre en Mission Planner un punto conocido desde el cual empezar a enviar instrucciones manuales o automáticas.

Ilustración 33 Base del bote donde se inicializa la posición

3.6.3 Visualización de datos

Después de haber inicializado la posición se comprueba que exista comunicación entre la base y el receptor Piksi del bote verificando que el led rojo esté encendido (ilustración 34)). Ahora se puede visualizar la posición del bote en el mapa de Mission Planner, el estado de los GPS, el modo en el cual el bote inicia (manual o automático) y el estado en el que se encuentra ya sea armado o desarmado, entre otras características (ilustración 35).

Ilustración 34 Rover del bote comunicándose con la base

Ilustración 35 Programa Mission Planner con la antena de telemetría conectada

3.6.4 Envió de instrucciones

Para poder maniobrar con instrucciones manuales, el estado del controlador debe estar armado y el modo debe ser Manual (el estado y el modo se pueden comprobar en la pantalla del Mission Planner). Ya con el controlador armado, el modo en Manual y con la aplicación de "Control" abierta como ventana flotante, se pueden enviar instrucciones con los joysticks del gamepad (ilustración 36). Para maniobras automáticas se desarma el controlador con el gamepad, luego se traza una ruta en la ventana de "Plan" del Mission Planner y se sube esa ruta al controlador con el botón "Write" (ilustración 37). Finalmente, al armar el controlador con el gamepad el bote empezará a recorrer la ruta subida.

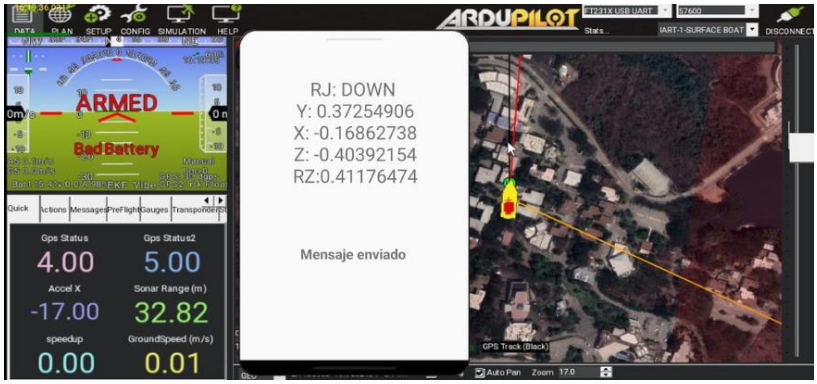


Ilustración 36 Visualización de las instrucciones manuales enviadas al bote desde el gamepad

Ilustración 37 Interfaz de la ventana "Plan" del Mission Planner

3.6.5 Tiempo de autonomía

La estimación inicial de la autonomía del bote, utilizando los motores antiguos de 280 [W], fue de aproximadamente 1 hora y 4 minutos (1.06 horas). Este cálculo se basó en una demanda de corriente total del sistema de 18.82 [A], según la referencia [11]. No obstante, la introducción de nuevos motores, con un consumo de 312 [W] [17], implica una reducción en el tiempo máximo de operación del bote debido al incremento en el consumo de energía.

La potencia total cambió de 282.35 W [11] a 314.35 [W]. Este valor se obtuvo utilizando la ecuación 3.1 donde la potencia total nueva es igual a la suma de la potencia total anterior de 282.35 [W] con la diferencia entre la potencia que consumen los motores actualmente ($P_{nuevos\_motores}$ = 312 [W]) y la potencia de los motores anteriores ($P_{antiguos\_motores}$ = 280 [W]).

$$P_{total\_nueva} = 282.35 + (P_{nuevos\_motores} - P_{antiguos\_motores}) \quad (3.1)$$

Donde:

P_{total_nueva} es la potencia total del sistema después de cambiar los motores en Vatios.

$P_{nuevos\_motores}$ es la potencia nominal de los nuevos motores en Vatios.

$P_{antiguos\_motores}$ es la potencia nominal de los antiguos motores en Vatios.

$$P_{total\_nueva} = V_{alimentación} * I_{sistema\_nueva} \quad (3.2)$$

Donde:

P_{total_nueva} es la potencia total después de cambiar los motores en Vatios.

$V_{alimentación}$ es el voltaje que se suministra al bote batimétrico en Voltios.

$I_{sistema\_nuevo}$ es la corriente del sistema con el cálculo de la nueva potencia en Amperios.

Usando la ecuación 3.2, donde $V_{alimentación}$=15 [V], se obtiene una nueva corriente del sistema $I_{sistema\_nueva}$ = 20.95 [A].

Finalmente, utilizando la ecuación 3.3, se calculó el nuevo tiempo estimado de autonomía del bote. Se dividió la capacidad total de las baterías en paralelo ($C_{apacidad\_baterías}$ = 20 Ah [11]) por la nueva corriente del sistema ($I_{sistema\_nuevo}$ = 20.95 [A]), obteniendo así una autonomía aproximada de 57 minutos (0.95 horas):

$$t_{autonomía\_nuevo} = C_{apacidad\_baterías} / I_{sistema\_nuevo} \quad (3.3)$$

Donde:

$t_{autonomía\_nuevo}$ es el tiempo de autonomía del bote batimétrico en horas.

$C_{apacidad\_baterías}$ es la capacidad de las baterías en paralelo en Amperios-hora.

$I_{sistema\_nuevo}$ es la corriente del sistema con el cálculo de la nueva potencia en Amperios.

3.7 Análisis de costos

Nombre del equipo	Descripción	Cantidad	Valor unitario	Total
Arduino UNO	Microcontrolador utilizado para recrear la señal del control remoto con las instrucciones del nuevo	1	$30	$30
Cable de comunicación tipo B/A	Cable para conectar el Arduino con la tableta	1	$1.5	$1.5
Equipo OTG tipo C/B	Equipo que permite la comunicación de la tableta con el Arduino	1	$3.5	$3.5
Control Remoto Ipega 9083S	Control para el manejo manual del bote	1	$40	$40
Placa STM31F1	Microcontrolador que sirve de cerebro para el convertidor de SBUS a USB	1	$8	$8
FTDI232	Chip para modificar el software del STM31F1	1	$2	$2
Cable divisor USB C a doble USB C	Cable que divide un USB C macho a dos USB C hembra	1	$9	$9
Turbinas	Equipo de propulsión del bote	2	$30	$60
	Total			154$

Tabla 1 Equipo nuevo utilizado para el desarrollo del proyecto

El microcontrolador STM32F1 permitió realizar la conversión de una señal SBUS a una USB al configurarla con el chip FTDI232 y juntos solo representaron un gasto de $10.

Los elementos de hardware más importantes, que se encuentran en la interfaz electrónica como el ArduinoUNO y el control remoto Ipega 9083S, son los que representan unos de los mayores gastos ya que suman $70 ($30 y $40 respectivamente). Los componentes para la comunicación y energización como el cable de comunicación tipo B/A, el equipo OTG tipo C/B y el cable divisor USB C a doble USB C solo representan un gasto de $14.

Además, la adquisición del nuevo equipo de propulsión incurre en un gasto considerable. Cada unidad se adquirió a un costo de 30 dólares, lo que elevó el costo total a 60 dólares. Este monto es comparable al costo de los componentes de hardware de la interfaz electrónica.

La tabla 1 no incluye los elementos pasivos como resistencias, capacitores o cables comprados para la implementación de algún circuito (Convertidor de señal SUB a USB) o ampliación de la red eléctrica (aumentar la longitud del cableado de las turbinas).

Ya incluyendo los cambios realizados y todas las herramientas utilizadas el costo total del bote batimétrico del CTDI de la ESPOL tiene un valor de $3,117.87. Además, este costo total podría disminuir al momento de replicar el proyecto dado que si se adquieren nuevos equipos para otros botes batimétricos se puede obviar comprar el control remoto 3DR X8+ y directamente remplazarlo con el gamepad.

3.7.1 Comparación de costos

Nombre	Descripción	Total
Bote batimétrico CTDI ESPOL	Equipo USV con sistema de batimetría con un ecosonda ping2 Manejo manual y automático con Mission Planner Técnica RTK de posicionamiento.	$3,117.87
Bathycal Survey Catamaran	El BathyCat es un catamarán de reconocimiento portátil no tripulado diseñado para completar estudios batimétricos con un ecosonda. Compartimento interno: Transmisor: 2,4 ghz FUTABA T6K Alcance del transmisor: 2 km Ecosonda: Bathylogger BL200 o bl700 AutoPilot: control autónomo del BathyCat.	$3,700.00
HyDrone	HyDrone-ASV es un bote con forma de catamarán portátil, controlada de manera autónoma y remota, desarrollada para aplicaciones de levantamiento hidrográfico. Control remoto: RCU 2.4Ghz de largo alcance Alcance remoto: 2 km Ecosonda: Fondo marino HydroLite™-, HydroLite-DFX	$7,500.00
POSEIDON SU30	SOUTH Poseidon SU30 USV es una nueva generación de USV de alta inteligencia. El pequeño físico hace que el levantamiento hidrográfico vaya a todas partes. Los campos de topografía y cartografía, hidrogeología y gestión del agua se pueden cubrir fácilmente. Métodos: Puente de red 5.8G R/C 2.4G Rango R/C: 2KM Estándar: Ecosonda SDE-18S Modo de enfoque: Automático/ Semiautomático/ manual	$27,950.00

Tabla 2 Modelos de botes existentes en el mercado en comparación con el bote del CTDI

Para saber si es ventajoso replicar el proyecto o no, se analizaron otras marcas de botes autónomos como se muestra en la tabla 2, algunos realizan funciones parecidas que el bote del CTDI y otros tienen diferentes tecnologías a su disposición dependiendo de las necesidades de los usuarios y de la industria.

Como se mencionó en el análisis de costos, el valor total del bote batimétrico del CTDI de la ESPOL con los cambios realizados fue de $3,117.87 y es mucho más barato en comparación con otros botes en el mercado que cumplen con funcionalidades parecidas como el Bathycal Survey Catamaran de $3,700.00 y el HyDrone de $7,500.00. En cambio, para otros modelos de botes con diferentes tecnologías como el POSEIDON SU30 que permite realizar topografía, cartografía, hidrogeología, etc., ya la diferencia de precio es mucho más significativa dado que este modelo vale $27,950.00.

Capítulo 4

4.1 Conclusiones y recomendaciones

4.1.1 Conclusiones

1. Se unificó los softwares que involucran la puesta en marcha del bote batimétrico en un dispositivo portátil utilizando Android Studio y Arduino IDE. Esto agilizó el proceso de inicialización de la base del bote y redujo el número de equipos que se usan para el envío de instrucciones manuales y automáticas.
2. Se comprendió el funcionamiento del transmisor, el receptor y del protocolo SBUS, que utiliza el control remoto para comunicarse con el controlador del bote batimétrico. Esto permitió el análisis de la señal que este enviaba y en consecuencia fue posible su recreación en el Arduino UNO para el envío de instrucciones con el gamepad hacia el controlador del bote.
3. Se implementó el soporte para los componentes de la interfaz electrónica que involucra el Arduino, el transmisor de radio, la antena de telemetría, el dispositivo portátil y el gamepad. Esto facilitó el transporte y la operación de los componentes de maniobra y comunicación del bote batimétrico.
4. El cambio de los motores otorgó una mayor potencia al sistema de propulsión del bote, pero redujo el tiempo de operación de 1h 4min a 57min, lo que equivale a una reducción del 11% aproximadamente.
5. Se fijaron los componentes a una plancha de plástico que permitió visualizar las conexiones de los elementos que contiene el bote batimétrico, además de evitar que exista ruido ocasionado por el movimiento del bote en operación en el receptor Piksi y el GPS del controlador.

4.1.2 Recomendaciones

1. Permitir que la aplicación creada para leer las acciones realizadas en el gamepad, y que es la que se encarga de mandar esa información al Arduino UNO, pueda trabajar en segundo plano con otras aplicaciones abiertas y no tener que usar la función de ventana flotante de los teléfonos con sistema operativo Android.
2. Investigar los motivos por los que la aplicación de Mission Planner falla al momento de conectar el cable divisor USB C a doble USB C al teléfono para que los otros dos equipos como el Arduino UNO y la antena telemétrica estén conectados al mismo tiempo.
3. Manipular el gamepad con delicadeza. No se deben presionar los botones de forma brusca o manipular los joysticks de forma arbitraria o muy rápida ya que podría causar que el Arduino colapse y se tengan que reiniciar los componentes.
4. Conectar y utilizar los componentes de la interfaz electrónica tanto de hardware como de software en el orden especificado en el manual para que el sistema de puesta en marcha del bote funcione de forma correcta. (APÉNDICE B)
5. Es crucial evitar movimientos bruscos al acelerar y maniobrar en modo manual, así como abstenerse de realizar giros de 90 grados en modo automático, para prevenir el riesgo de volcadura. Dado que el bote carece de tapas de sellado al vacío, podría entrar agua, causando daños significativos a los componentes internos.

Referencias

[1] Arduino. (s.f.). What is Arduino? [En linea]. Disponible en: https://www.arduino.cc/en/Guide/Introduction

[2] Ardupilot. (s.f.). Mission Planner Overview [En linea]. Disponible en: https://ardupilot.org/planner/docs/mission-planner-overview.html

[3] Atmel. (s.f.). 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash DATASHEET [En linea]. Disponible en: https://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[4] Axelson, J. (2007). Serial Port Complete. Estados Unidos: Lakeview Research.

[5] Developers. (s.f.). Handle controller actions [En linea]. Disponible en: https://developer.android.com/develop/ui/views/touch-and-input/game-controllers/controller-input

[6] Developers. (s.f.). Meet Android Studio [En linea]. Disponible en: https://developer.android.com/studio/intro

[7] Github. (s.f.). USBSerial [En linea]. Disponible en: https://github.com/felHR85/USBSerial?tab=readme-ov-file

[8] Ípega. (s.f.). Products IPEGA [En linea]. Disponible en: http://ipega.hk/gamehandle/53-180.html

[9] Hingxyu. (2019, Enero 08). Arduino Android Serial Communication [En linea]. Disponible en: https://hingxyu.medium.com/arduino-android-serial-communication-b72b124142fb

[10] Laburthe, D. (2012, Mayo 05). FrSky_DJT_Manual_v0.1 [En linea]. Disponible en: https://es.scribd.com/document/373477754/FrSky-DJT-Manual-v0-1

[11] Leones, D. Zambrano, I. (2023). Diseño e implementación de un bote autónomo para realizar un análisis batimétrico en el lago de la ESPOL [Tesis de grado Escuela Superior Politécnica del Litoral].

[12] Liang, O. (2021, Abril 01). FPV Protocols Explained (CRSF, SBUS, DSHOT, ACCST, PPM, PWM and more) [En linea]. Disponible en: https://oscarliang.com/rc-protocols/#SBUS

[13] Liang, O. (2019, Enero 23). DIY SBUS to USB Converter for FPV Simulators [En linea]. Disponible en: https://oscarliang.com/diy-SBUS-USB-converter/

[14] Manual.ec. (s.f.). 3DR X8 plus manual [En linea]. Disponible en: https://www.manual.ec/3dr/x8-plus/manual

[15] ST. (s.f.). STM32F1 Series [En linea]. Disponible en: https://www.st.com/en/microcontrollers-microprocessors/stm32f1-series.html

[16] FTDI chip. (s.f.). About us [En linea]. Disponible en: https://ftdichip.com/corporate-profile/

[17] Amazon. (s.f.). Simlug Hélice de barco [En linea]. Disponible en: https://www.amazon.com/-/es/Simlug-potencia-propulsor-profunda-impermeable/dp/B08G4BTZ89/ref=sr_1_4?__mk_es_us=%C3%85M%C3%85%C5%BD%C3%95%C3%91&crid=2IS3X5MBY6S2Q&keywords=dilwe+h%C3%A9lice+de+agua+profunda%2C+560kv+300m&qid=1705744498&sprefix=dilwe+h%C3%A9lice+de+agua+profunda%2C+560kv+300m+%2Caps%2C145&sr=8-4&ufe=app_do%3Aamzn1.fos.006c50ae-5d4c-4777-9bc0-4513d670b6bc

APÉNDICE A

VISTAS DEL DISEÑO QUE SE ANCLA AL CONTROL

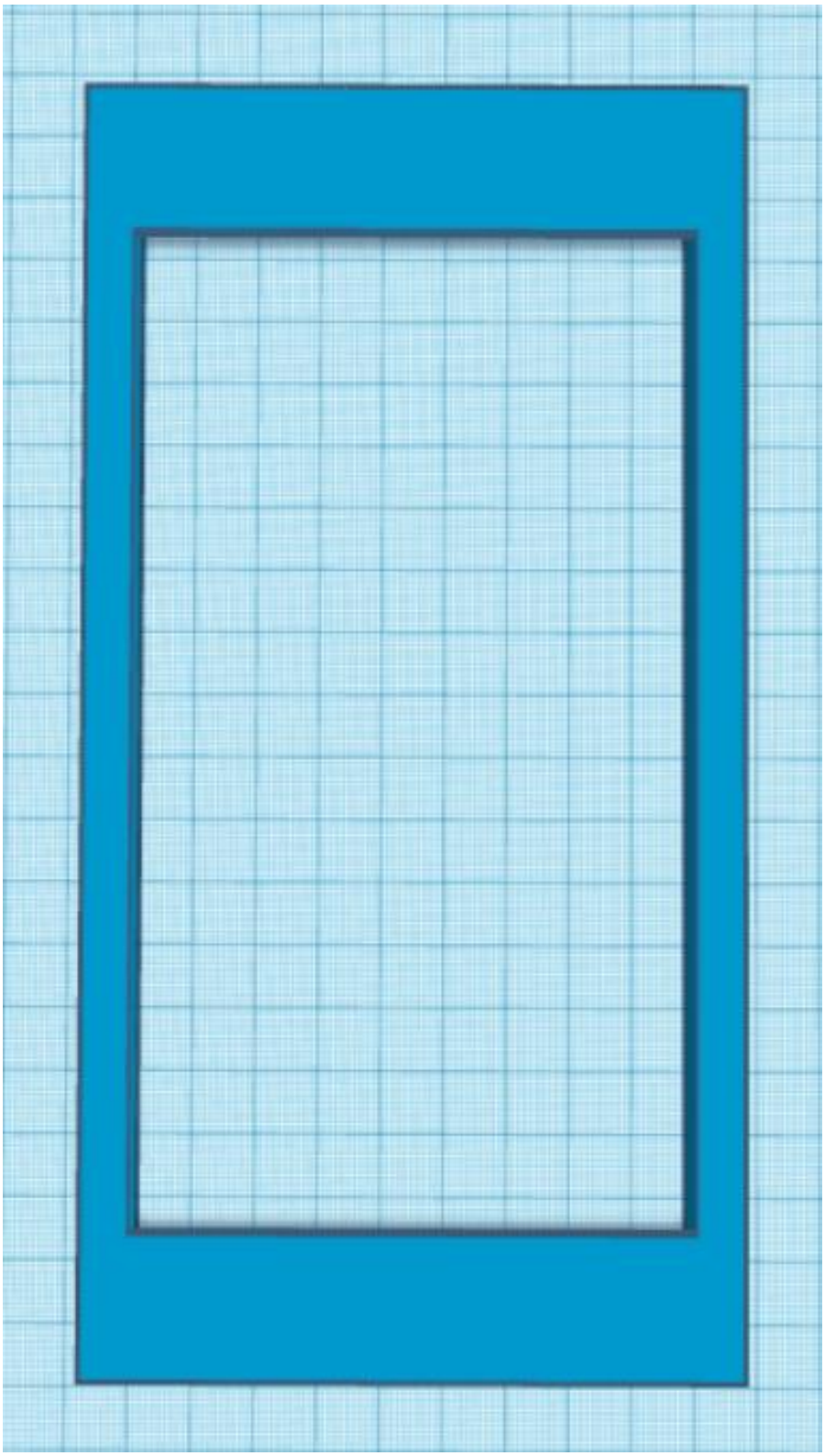
PLANO 1 Vista superior de la primera pieza del modelo

PLANO 2 Vista inferior de la primera pieza del modelo

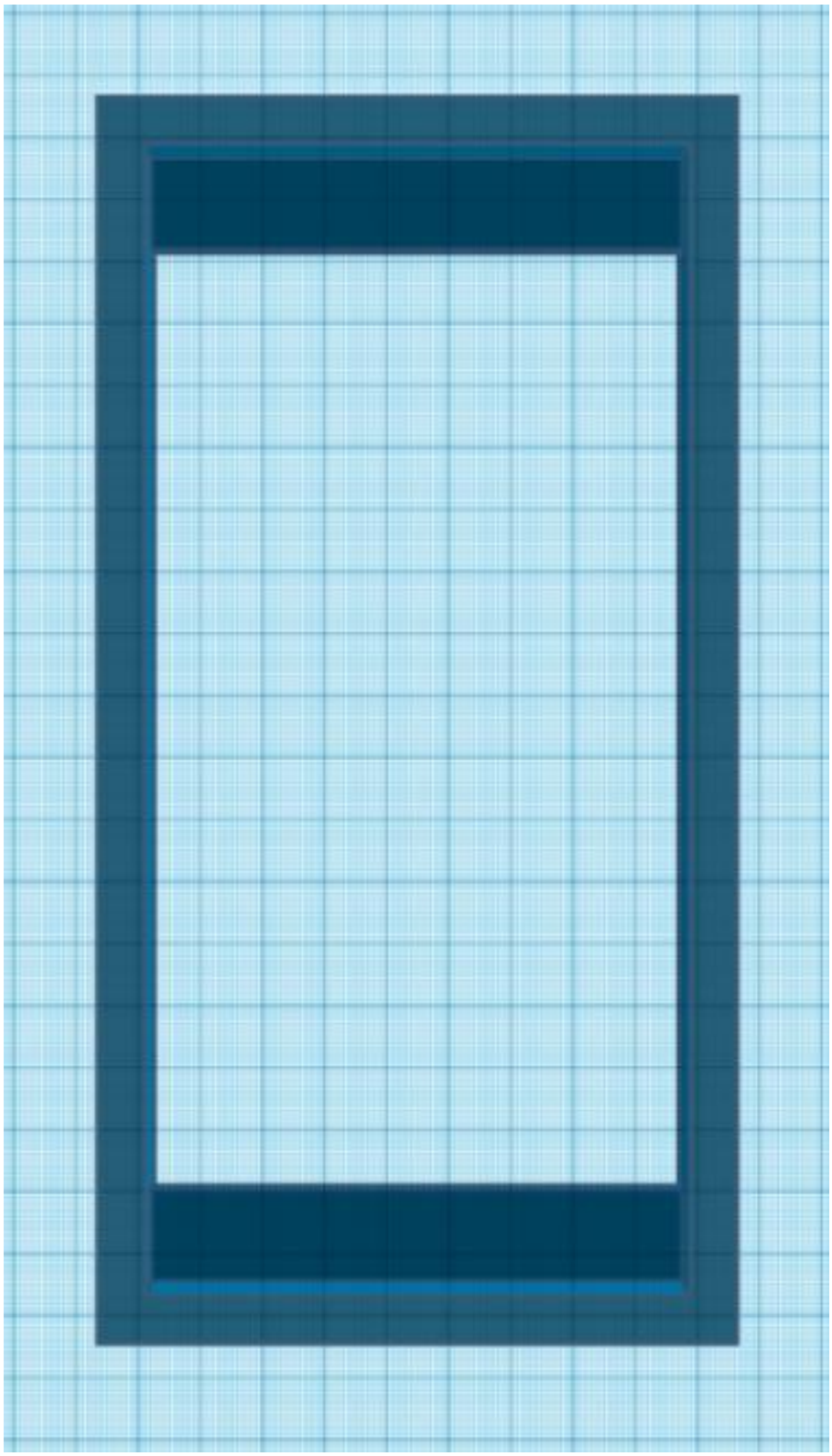
PLANO 3 Vista superior de la segunda pieza del modelo

PLANO 4 Vista inferior de la segunda pieza del modelo

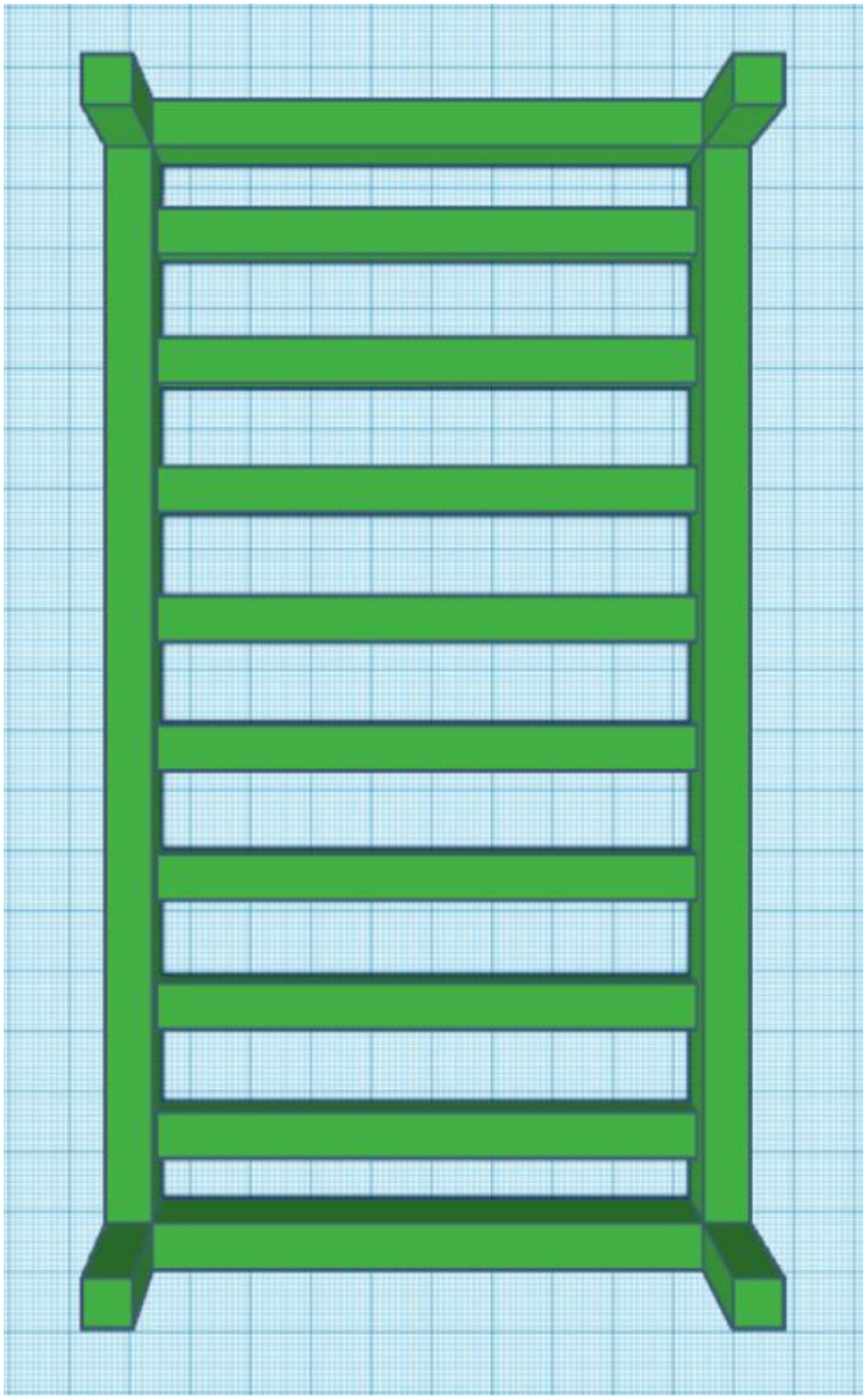
Plano 1



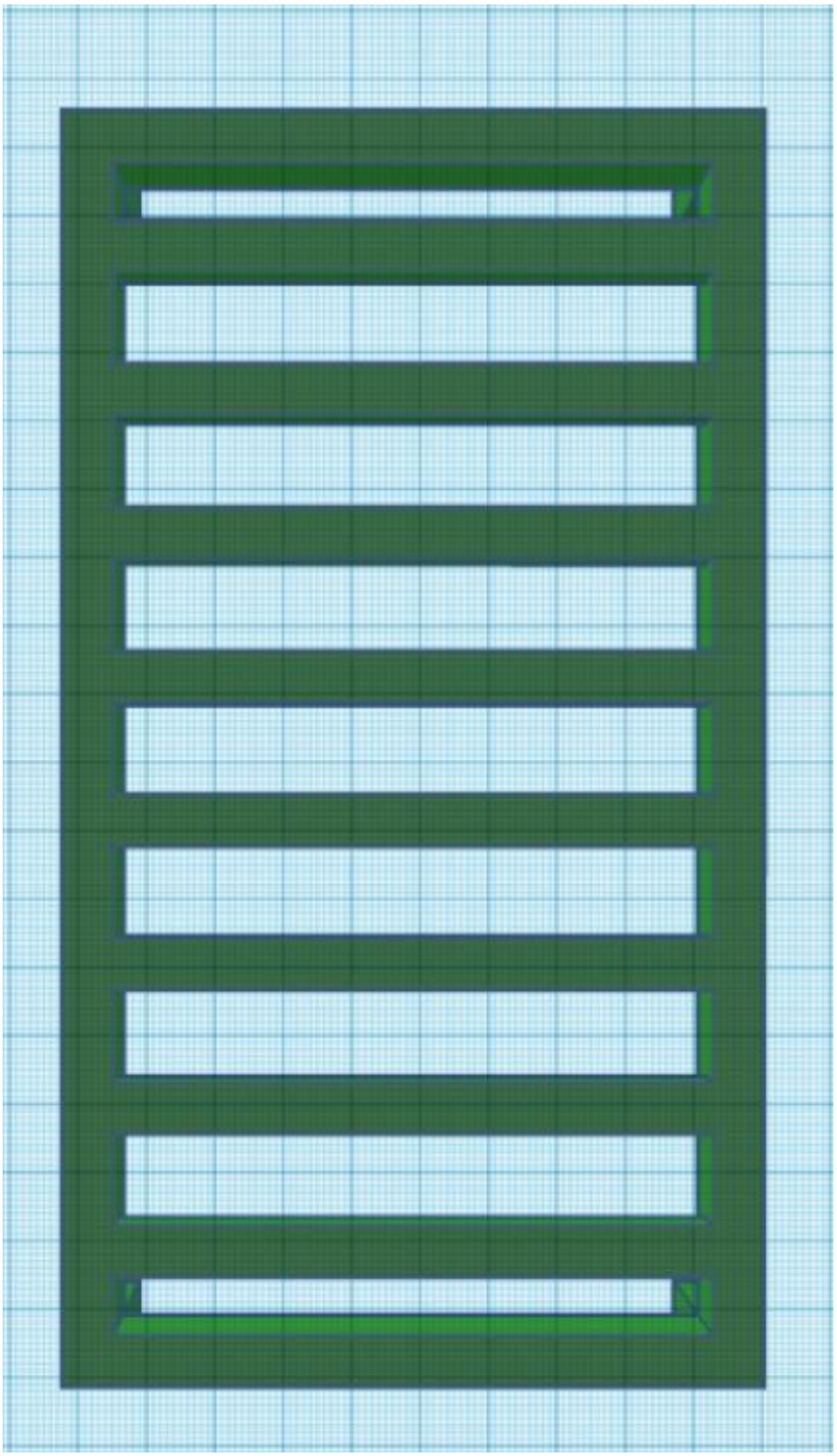
Plano 2



Plano 3



Plano 4



APÉNDICE B

START UP MANUAL

Calibración de la posición inicial del bote

1) Conectar el receptor de GPD Piksi de la base con un cable que permita la comunicación y su energización como el cable que se muestra en la ilustración 38. Un exremo va a una batería portátil que permite alimentar el receptor Piksi y el otro extremo va al teléfono. El cable que va al teléfono tiene en su punta un módulo OTG para permitir la transferencia de datos. También recordar conectar las dos antenas que utiliza el receptor Piksi como se muestra en la ilustración 33.

Ilustración 38 Base

2) Una vez esté conectado, encender la batería portátil y esperar a que uno de los leds del receptor Piksi empiece a parpadear lentamente de color verde. Esto quiere decir que el receptor Piksi está buscando señales satelitales, una vez que el led este parpadeando lentamente de color verde se puede conectar el cable al teléfono.

3) Ya conectado el teléfono si se tiene instalado Mission Planner le aparecerá una ventana emergente que le indica si desea abrir la aplicación cuando se conecta el receptor Piksi (ilustración 39), le da clic a cancelar, y se procede a abrir la aplicación de Geo-Locator.

Ilustración 39 Ventana emergente al conectar el teléfono a la base

4) Al abrir la aplicación le da clic en la opción de USB, selecciona el dispositivo que le pertenece al receptor Piksi como se muestra en la ilustración 40, después le aparecerá la opción de seleccionar drivers en este caso se seleccionará el controlador del receptor Piksi como se muestra en la ilustración 40 y se le da clic en guardar.

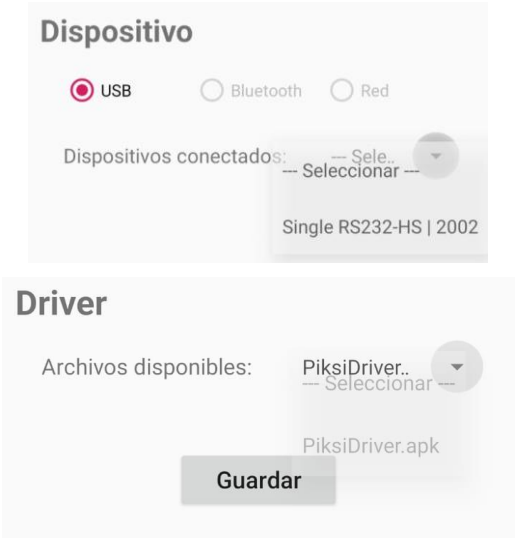


Ilustración 40 Ventana de dispositivos y controladores del Geo-Locator

5) Después le saldrá una ventana que indica como se desea configurar el receptor Piksi, como deseamos localizar la ubicación de la base seleccionamos la opción de Rover.

6) Al seleccionar la opción de Rover le aparecerá una ventana emergente que le preguntará si desea permitir que Geo-Locator acceda a Single RS232-HS y le da aceptar (ilustración 41).

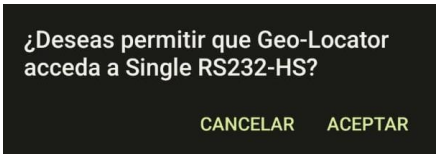


Ilustración 41 Ventana emergente al seleccionar la opción rover

7) Al aceptar ya se pueden realizar cambios en la configuración RTK-Rover, pero en este caso no se deben realizar ningún cambio solo darle clic en Guardar. Si al dar clic en guardar la aplicación bota un mensaje de "dispositivo no conectado", se soluciona cambiando de ventana a otra como se muestra en la ilustración 42. Al momento de volver a la ventana donde estaba la configuración del RTK-Rover volver a dar clic en guardar y ya pasará a la sección de Generar Hito.

Ilustración 42 Ventana de información

8) Ya en la ventana de Generar Hito solo se debe colocar el tiempo que el receptor Piksi va a tomar para recolectar datos, a la derecha de dónde dice "Tiempo de captura (min):" colocar 3 y darle clic en iniciar como se muestra en la ilustración 43.

Ilustración 43 Ventana de Generar Hito

9) Al darle clic en iniciar el receptor Piksi empezará a captar satélites que tenga cerca y empezar a triangular su posición, una vez finalizado el proceso aparecerán los resultados del Hito en la parte inferior de la pantalla en la que se encontraran las coordenadas en formato UTM. Se recomienda tomar una captura para que al momento de ya realizar la configuración del receptor Piksi como base en vez de como rover colocar esas coordenadas, y por último dar clic a la pestaña incluir lecturas RTK y dar clic en guardar (ilustración 44).

Ilustración 44 Proceso de generación de hito finalizado

10) Ya una vez obtenida la ubicación se puede cerrar y volver a abrir la aplicación de Geo-Locator o dar clic en el icono que se muestra en la ilustración 45.

Ilustración 45 Icono para configurar de nuevo el receptor Piksi

11) Repetir el paso 4.

12) En la ventana de “cómo se desea configurar” seleccionar Base.

13) En la ventana de “configurar la posición de la base” se coloca la zona UTM, en este caso al estar en ecuador-guayaquil se pone 17 y cambiar la “n” por la “s”, después colocar las coordenadas obtenidas en el hito, debería quedar como se muestra en la ilustración 46. Darle clic en guardar, para verificar que la posición fue guardada correctamente se pude dar clic en “Ver config actual” y saldrá una ventana emergente con la posición guardada que tiene el receptor Piksi.

Ilustración 46 Ventana de configuración de la base

14) Listo ya se puede desconectar el teléfono, el receptor Piksi debe estar energizado todo el tiempo.

Puesta en marcha del bote

1) Una vez configurada la base ya se puede dar paso a encender el bote batimétrico presionando el interruptor.

2) Una vez encendido el bote lo primero que se debe verificar es que exista comunicación entre el receptor Piksi del bote y el receptor Piksi que está en la base, se lo puede verificar si al momento de encender el bote un de los leds del receptor Piksi del bote está parpadeando de color rojo mientras el otro led está parpadeando de color verde como se muestra en la ilustración 34.

3) Conectar vía Bluetooth el gamepad y el teléfono.

4) Conectar el cable divisor de tipo USB C macho a USB C hembra al teléfono.

5) A unos de los extremos del divisor conectar primero la antena de telemetría, luego abrir el Misión Planner. Arriba a la derecha se deber dar clic en conectar para empezar la comunicación entre el bote y el teléfono, verificar de poner los mismos parámetros que se muestran en la ilustración 47.

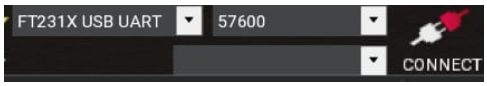


Ilustración 47 Configuración para conectar la antena de telemetría

6) Una vez conectados ya podemos ver la información que nos llega desde el controlador del bote tales como la posición en la que se encuentra, el estado de los GPS, si el bote está armado o desarmado, entre otras cosas (ilustración 36).

7) Antes de seguir se debe verificar que el bote se encuentre en manual como se muestra en la ilustración 48.

Ilustración 48 Indicadores del estado del bote en Mission Planner

8) Ahora para proceder con el control manual ya podemos conectar el Arduino al otro extremo del divisor USB.

9) Al conectar el Arduino le saldrá la una ventana emergente le dan clic en cancelar (ilustración 49), después proceden a abrir la aplicación “Control”.

Ilustración 49 Ventana emergente al conectar el Arduino

10) Una vez abierta la aplicación les aparecerá otra ventana emergente (ilustración 50), le dan clic en aceptar y después para que se guarde esa configuración proceden a cerrar la aplicación, esto quiere decir que no debe aparecer en las ventanas recientes como se muestra en la ilustración 51.

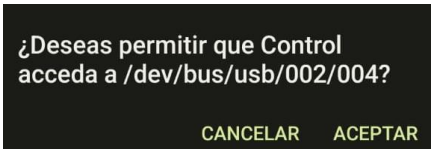


Ilustración 51 Ventana emergente al abrir la aplicación "Control"

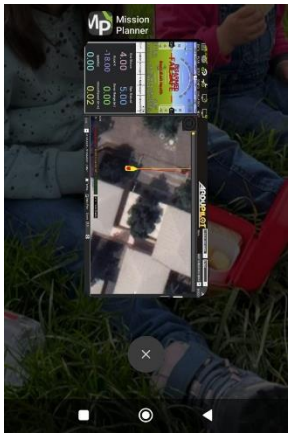


Ilustración 50 Ventana de recientes del telefono

11) Después se procede a abrir la aplicación del control de nuevo y se presiona X para armar el controlador, se deberá escuchar un pitido largo y se procede a mover los joysticks para ver si funcionan los motores, luego se presiona Y para desarmar el controlador y se comprueba que los motores no funcionen al mover los joysticks de nuevo.

Nota - Para poder realizar el control automático en Mission Planner, tener aplicación de "Control" abierta al mismo tiempo y poder manipular el bote con el gamepad de forma manual en caso de ser necesario, se debe realizar los siguientes pasos:

12) Se debe abrir la ventana de recientes y mantener presionado la aplicación del control y dar clic en el icono que se muestra en la ilustración 52, esto hará que la aplicación del control sea una ventana flotante.

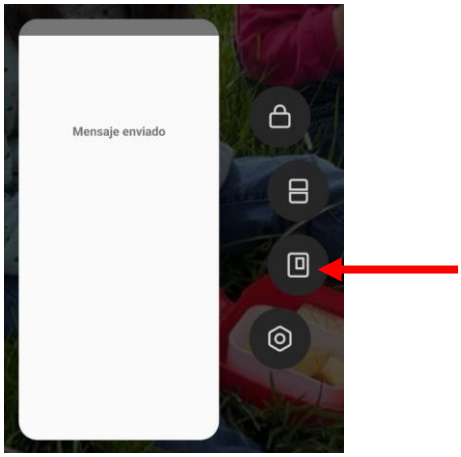


Ilustración 52 Configurando la aplicación "Control" como ventana flotante

13) Teniendo ya la aplicación del control como ventana flotante se procede a abrir el Mission Planner y ya se puede interactuar con las dos aplicaciones como se muestra en la ilustración 36.

14) Así ya se pueden realizar instrucciones manuales con el gamepad y automáticas con el Mission Planner.