INSTITUTO UNIVERSITARIO AERONÁUTICO

DESCRIPCIÓN GENERAL SOFTWARE BASE TERRENA IUA-GCS

Informe Técnico: DMA 002/16 **Revisión:** 1.00

Proyecto: PIDDEF 038/14 - Paracaídas Comandado Autónomo

Fecha: 29/09/2016

Autor/es: Germán Weht

Revisó:

Ing. Andrés Liberatto
Dir. Dpto. Mecánica Aeronáutica

Índice

“DESCRIPCIÓN GENERAL SOFTWARE BASE TERRENA IUA-GCS”

Por

Germán Weht

Resumen

En el marco del PIDDEF 38-14 “Paracaídas Comandado Autónomo para Entrega de Cargas”, se desarrolla un software de una estación de control terrena (Ground Control Station) utilizando Qt Creator 5.

La finalidad del programa es hacer el seguimiento y navegación del vehículo cuando este vuele en condiciones de vuelo manual y automático.

El código esta basado en rutinas propias y librerías OpenSource.

Córdoba, 29 de septiembre de 2016

1. Introducción

Una estación de control terrena (GCS) es un software que se ejecuta en una computadora en tierra que se comunica con un vehículo no tripulado vía wireless. Su función principal es la de mostrar la performance del vehículo en tiempo real. Además funciona como un “cockpit virtual” mostrando los instrumentos de navegación como si fuera un cockpit real. Otra de las aplicaciones buscadas de una GCS es la posibilidad de control de misión en vuelo, esto significa que el software pueda cambiar la misión y/o los parámetros de vuelo.
Debido a las necesidades específicas del proyecto PIDDEF 38-14 “Paracaídas Comandado Autónomo para Entrega de Cargas”, se inició el desarrollo de una base de control terrena (IUA-GCS). Existen softwares OpenSource con la capacidad de hacer seguimiento y navegación, entre ellos se destacan QGroundControl, Simple-GCS, APM - ArduPilot. Sin embargo hay ciertas funcionalidades que son necesarias al momento de modelar la planta y los sistemas de control, que los softwares anteriores, no las tienen.

2. Desarrollo

A nivel general el software consta de 4 ventanas fijas y 2 móviles, cada una de ellas muestra información específica. La ventana 1 *Location* muestra la posición de la aeronave en plano terrestre. En la ventana 2 *Electrical Systems* informa el estado de los sistemas eléctricos, baterías, consumo, etc. Ventana 3 *Serial Port* muestra la comunicación por puerto serie, la configuración se hace a través de una ventana móvil y por último la ventana 4 *Graph* permite graficar en tiempo real variables de interés.

2.1. Ventana 1

La localización de la aeronave en el plano se hace a través del widget “opmapcontrol” [1]. La librería es de uso libre y se la puede bajar desde el link opmapcontrol-ex.
Esta librería tiene la funcionalidad de trabajar offline, mostrando la posición de la aeronave. Permite además la edición de los waypoints en tiempo real. A continuación se muestra una figura (1) de la ventana *Location* implementada en el soft IUA-GCS.

Figura 1: Pestaña *Location* IUA-GCS utilizando la librería “opmapcontrol”.

La librería esta pensada para ser compilada con Qt4, por lo que fue necesario editar el archivo *.pro* del proyecto, reemplazando las librerías de Qt4 por las de Qt5, como fue el caso de la librería *declarative* por *sql, qml y widget*.

2.2. Ventana 2

La pestaña 2 *Electrical Systems*, tiene por finalidad mostrar el estado de los sistemas eléctricos del vehículo. Actualmente tiene implementado un amperímetro, un voltímetro y dos indicadores de carga de baterías. Además se pretende agregar indicadores de posición de servos, intensidad de señal de las antenas e indicadores de estado de la carga paga. En la figura (2) se muestra el estado actual de la ventana de los sistemas eléctricos.

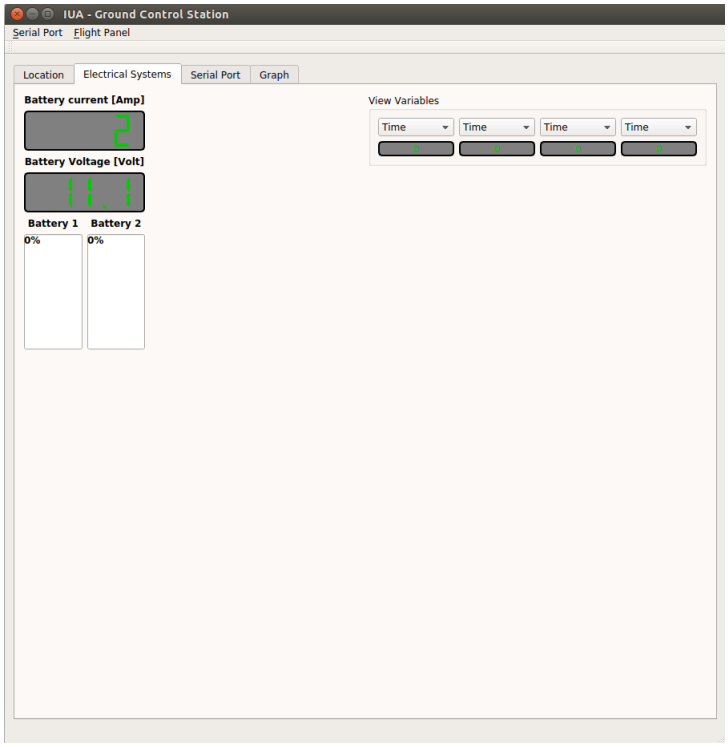

Figura 2: Pestaña 2: sistemas eléctricos.

Sobre la derecha se implementó un visualizador de variables. Se pueden selectar hasta 4 variables. La idea principal es poder mostrar solo las variables de interés.

2.3. Ventana 3

La pestaña *Serial Port*, permite conectarse al vehículo por medio del puerto serie.
Esta ventana consta de dos partes, una fija y otra móvil. La parte fija corresponde a la recepción y envío de datos a través del puerto serie. Por medio de la ventana móvil se configura el puerto serie para establecer la comunicación.
Esta librería es propia del del departamento de Mecánica Aeronáutica. En la figura (3) se muestran las ventanas de recepción y envío de datos (izquierda) y la ventana móvil de configuración del puerto serie (derecha).

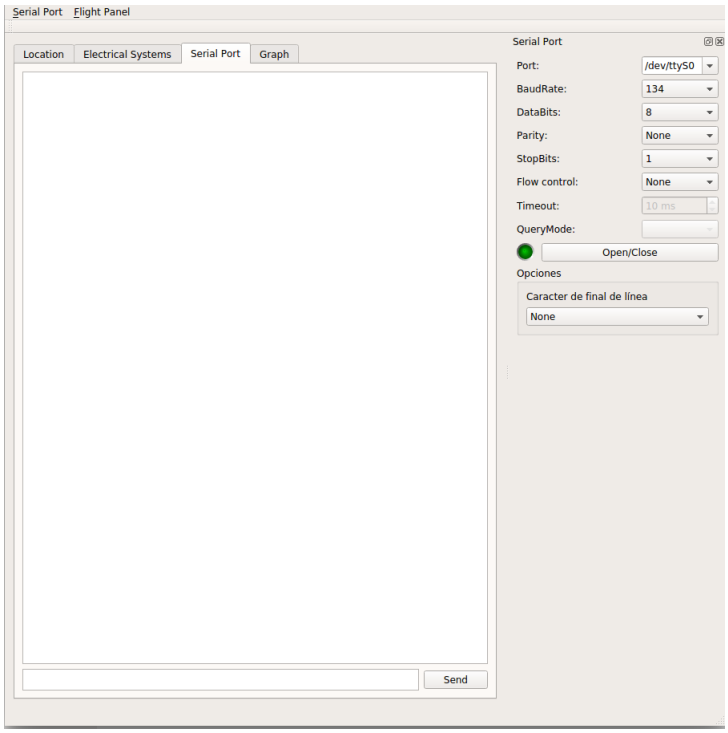

Figura 3: Pestaña *Serial Port*, a la izquierda panel de envío y recepción de datos, a la derecha panel de configuración del puerto.

De la figura (3) se aprecia que la ventana móvil puede ocultarse o desacoplar de la ventana principal. Esto permite, una vez configurado el puerto, tener mayor espacio en las pestañas fijas para visualizar los datos mostrados.

2.4. Ventana 4

Esta pestaña se pensó para tener la capacidad de graficar la performance del vehículo en tiempo real. Esta rutina esta implementada mediante la librearía qcustomplot [2].
Se añadió un panel de configuración sobre la derecha de la ventana. Este panel permite la configuración de las variables a graficar. Una particularidad de lo anterior, es que se pueden graficar dos variables a la vez. Además permite guardar un archivos con los gráfico realizados. En la figura (4) se muestra el estado actual de la ventana:

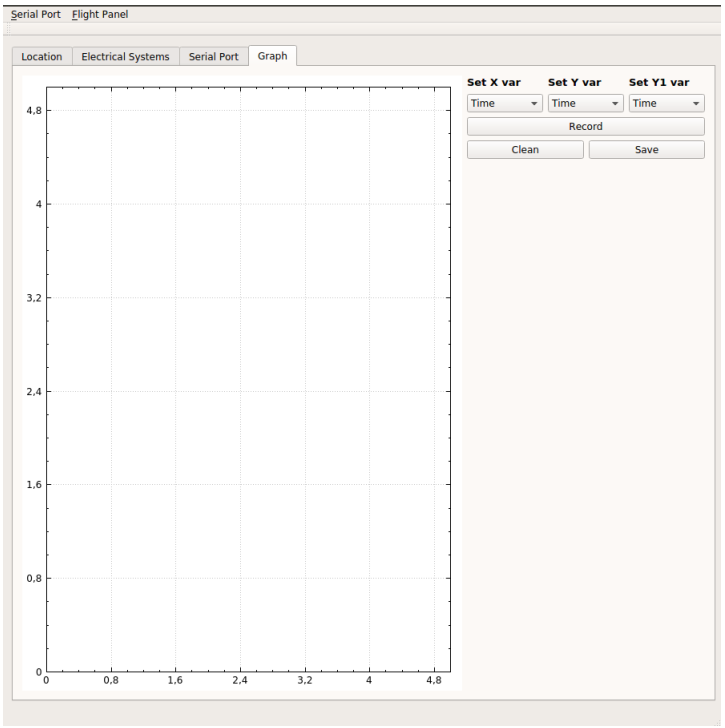

Figura 4: Pestaña *Graph* implementada mediante la librería “qcustomplot”.

2.5. Ventana de instrumentos

La ventana de instrumentos *Flight Instruments*, tiene la opción de poder acoplarse y desacoplarse de la ventana principal, quedando siempre por delante de las ventanas. Esto permite dar prioridad a los instrumentos sobre el resto de las ventanas indicadoras. Actualmente consta de 4 indicadores, PFD (Primary Flight Display), ASI (Airspeed Indicator), ALT (Barometric Altimeter) y VSI (Vertical Speed Indicator) un panel de alarmas y un reloj que indica el tiempo de vuelo. Los indicadores se implementaron utilizando la librería Qt Flight instrument [3].

2.6. Protocolo de transmisión de datos

Actualmente el protocolo de transmisión de datos es el estándar de APM 1.01. Al corto plazo se pretende reemplazar el protocolo APM 1.01 por el MAVLink, permitiendo mayor modularidad en el envío de datos de cada sistema.

Obtención del código

El código fuente se puede obtener del siguiente repositorio:
//192.168.2.2/dma-fi/Proyectos %20DMA/Paracaídas/programacion/IUA-GCS.git
Haciendo una copia local al repositorio personal mediante:

- git fetch
- git checkout gcs-iua-protocol

3. Bibliografía

[1] S. Bu, "Qt Widget mapping and navigation tool," http://www.adv-ci.com/blog/source/opmapcontrol_ex/, 2016, [Online; accessed August-2016].

[2] E. Eichhammer, "Qt C++ widget for plotting and data visualization," http://www.qcustomplot.com/, 2016, [Online; accessed July-2016].

[3] M. M. Cel, "Qt4 and Qt5 widgets of flight instruments," http://marekcel.pl/, 2016, [Online; accessed August-2016].