



The Ground Control Station Design for Can-Sized Satellite (CanSAT) System

Noor Thuwaibah Abdul Razak^a, Huda A Majid^{*a}, Faiz Asraf Saparuddin^a, Muhamad Fitry Abdul Jalil^a, Muhamad Shakry Jamaluddin Jalil^a, Mohd Hazmi Mokhtar^b

^aUniversiti Tun Hussein Onn Malaysia, Faculty of Engineering Technology, Johor, Malaysia

^bCommunication Engineering Department, Faculty of Electrical Engineering, Universiti Teknologi Malaysia, Johor, Malaysia

Abstract. Nowadays, Natural disaster tragedy is now one of the world's biggest concerns. Can-sized satellite, MedSAT: Location-Aware CanSAT for On-Site Emergency Medical Supplies develop a platform for finding direction and accurately locating an emergency patient and providing emergency medical supplies such as bandages, antiseptic wipes, sterile gauze pads of various sizes, insulin, pills, syringe and antivenom, as well as real-time visual feed for medical diagnosis during and after landing. This project focuses on the design of MedSAT and provides a real-time system to capture MedSAT's real-time data during descent. The objective of the real-time system is to improve the accuracy and location speed of MedSAT data collection which can provide readings of altitude, latitude and longitude to help MedSAT navigate to the patient location. Hardware design (flight controller, GPS module and telemetry kit), software design (Mission Planner) and real-time system (RTS) are the main components of this platform. In addition, the ground station was developed to communicate with users via wireless telemetry communication using MAVLink protocol. Based on the overall findings, MedSAT and ground station's compact and lightweight design was developed in search and rescue operations for emergency location.

Keywords: satellite; CanSat; telemetry; ground control station

DOI: [10.37869/ijatec.v1i2.24](https://doi.org/10.37869/ijatec.v1i2.24)

Received 15 July 2020; Accepted 6 August 2020; Available online 7 August 2020

© 2020 IRIS Publisher. All rights reserved.

1. Introduction

The first can-sized satellite concept (CanSAT) was started by Robert J. Twiggs at Stanford University in the late 1990's. All structure and components shall fit in a cylindrical envelope with diameter of 7cm and height of 11cm. The CanSAT shall fit inside a release mechanism pod which is drone during launch (20cm length x 15cm width x 15cm height) and the total weight of the CanSAT must not exceed 1000 grams. The CanSAT were renamed as MedSAT to fit with the mission. MedSAT is a can-sized satellite that can provide medical support for emergency purpose and real-time visual feed during and after landing for medical diagnosis. MedSAT required to communicate with operator at Ground Control Station (GCS) and sharing live video feed to show the condition and situation at the scene plus allocate a real time direction and location of the patient. Hence, GCS is designed to collect and analyze the real time data of MedSAT. The main purpose of MedSAT is to give an early medication such as insulin, patches or vaccine injection to the patient that hardly can be located by medical officer in the shortest time possible. The victims can be saved and the possibility to survive is higher with the help of MedSAT. Therefore, the collection of data and real time information from MedSAT is important. GCS are also able to display data on Graphical User Interface (GUI) for analysis purpose.

*Corresponding author: mhuda@uthm.edu.my

Plentiful studies have been made regarding can-sized satellite (MedSAT) that collect telemetries data and provide live video feed when launching in search and rescue operation. A can-sized satellite is built in the size of a soda can and also can be called as nano-satellite even though the functionality of a can-sized is far away to be compared to a real satellite. Nanosatellite was built to have the ability to collect data from environmental services and also providing a live video feed when it navigates itself to the target point. Telemetries data should be transmitted on a real-time basis while Internet-based streaming is used for the live video feed and 433 MHz transceivers are used to allow the collection of data to be received at GCS. The ground control station used to interact with MedSAT fundamentally is a software application in the ground-based computer that uses wireless telemetry communication to communicate with MedSAT. In addition, GCS has many types of equipment that are almost identical to the actual satellite. The telemetry data that the ground controller can observe such as airspeed, altitude, humidity, etc even the live location of the MedSAT in a map. Moreover, ground station is responsible for navigation (real-time estimation of the MedSAT situation, speed and altitude) and the control (maneuvering and other high-level behaviors). The Communication Protocol for Micro Air Vehicle Link (MAVLink) enables entities to communicate via a wireless channel. MAVLink is a lightweight communication protocol that can enables up to 256 unmanned air vehicles to communicate at the same frequency is discuss in [1]. This communication protocol used between MedSAT and (GSC) for bidirectional communication. The GCS sends commands and controls to the drone whereas the drone sends telemetry and status information. The MAVLink protocol is tested on multiple unmanned air vehicle platforms and several GCS software applications running on many operating system types. Besides, The MAVLink messages carry signals transmitted from GCS to the MedSAT and provide feedback from the MedSAT to the GCS, such as telemetry, heat beat and system status, allowing the pilot to keep aircraft under control. In another paper, a GCS was designed using Ardupilot Mega APM2 as presented in [2]. The first reason for selecting Ardupilot Mega APM2 as a controller for CanSAT is to maximize the light data acquisition system. The APM2 is sturdier and more stable than handmade cards. The APM2 involves accelerometer, gyroscope, barometer, humidity sensor, magnetometer, logic level converter, microprocessor ATMEGA3560 and voltage controller. With this setting, the GPS and the SD card can be easily integrated. The system's telecommunication part consists of the Zigbee network, which has been implemented with the communication modules of the XBee Pro S2B. The network is set up with one communication module connected to the satellite and another module linked to the ground station terminal. Satellite module receives all the analyzed data from the microcontroller via serial interface and sends it via radio waves to the ground station module. Both modules operate at a frequency of 2.4 GHz. For both devices, congestion control is enabled when communicating with the API mode setting on.

The proposed system in [3] uses as an antenna Arduino Uno 8-Bit Microcontroller Development Platform, GPS sensor, Xbee RF Modem and Shield, and as a pressure sensor BMP 085 I2C. It is straightforward to assemble the components and more emphasis can be placed on the firmware. The ground station design GUI is one of the most important aspects for CanSAT and design is briefly mentioned in this article. Many programming languages and software such as MATLAB, Java and LabView are investigated and analyzed for this study. C #programming language and Visual Studio are chosen for ground station GUI design because C #is an advanced, object-oriented high-level programming language and Visual Studio is a user-friendly platform for development. In addition to these functionalities, there are many benefits of using C #programming language for learners and some of them are; it comes with a large framework of pre-developed components (also called the. Net Framework), it is full of characteristics that make development quicker and easier, usually at the cost of flexibility and/or runtime performance. It is much smoother and has easier syntax to use, Visual Studio provides the ability to use advanced and easy interface. It is free and open source for small-sized applications.

The hardware design in [4] is using 434.43 MHz frequency. The CanSat transceiver module featured a $1/4 \pi$ -antenna wire-based. This type of antenna is well suited for the mission as it does not take up a lot of space and during the CanSat ejection is not easily damaged. A Yagi-antenna was used on the ground to achieve a long-range telemetry setup and to achieve a true two-way communication connection at the respective distances the antenna located on the CanSat and the corresponding amplifier had to be more sensitive than the $1/4 \pi$ -antenna used. One program, the Listener, is responsible for reading the antenna data and storing the data along with a permanent log

file in a temporary communication file. Another program, the Feeder, reads the file and filters, analyzes and processes the read data. The data processed will then be forwarded to any connected user, the Clients. Simultaneously, the ground station runs a web server, the presenter, which serves the requested web page for any connected client. The Presenter can define a connection between a Client and the Feeder and inform the browser of the Client to render the graphs found at the endpoints of `"/live / graph"` and `"/graph/"`. The Presenter also hosts a personalized blog at the endpoint of `"/blog/,"` a page describing the project at the endpoint of `"/about,"` and a live map rendered at the endpoint of `"/live / map"` which use Google Maps. Of course, information from previous missions can also be found on the website of the Presenter. A wireless hotspot could also be set up by the ground station. This is especially useful when launching in areas without access to the internet, as it allows bystanders to connect to the hotspot and view the data in real time without any connection to the internet.

In order to broadcast a live video, two main parameters in the communication link are needed; a large bandwidth and fast data transfer for both hardware and software configurations. Instead of trying to solve these issues using the Atmega AVR microcontroller, an unbiased microprocessor board is utilized to launch a Linux distribution that was also backed by other hardware components such as webcams. Next, with a 700~800 MHz microprocessor, 512 MB RAM and a modest VGA setting, Raspberry Pi model B development board is used. With these attributes, the "avconf" Linux library are able to be installed, the upgraded version of "ffmpeg" for recording, taking static images and streaming online videos. A handheld internet access provider is also needed which could be used for the satellite body, so on the Raspberry Pi board, the 3G USB modem is used. The 3G is not automatically identified on the Linux kernel, so the option is to either write a device driver for it from scratch or find an appropriate pre-written driver. The driver used is a tool called "sakis3 g," perfectly suited to the set-up needs. There are also two extra digital cameras installed, one for taking snapshots and storing them in the storage of SD cards, and the other for footage and storing videos in the storage of SD cards [5].

The ground station consists of a laptop, an antenna and a module for the receiver. The receiver module gathers the data transmitted and using MATLAB software, these data are plotted. The data transmitted to the computer via USB to Serial Converter after being received from the system. Then the received data is analyzed using MATLAB software, specific data is extracted, and the parameters of the sensor were plotted versus time. Iran's permitted frequency is about 2.4 GHz at 1-watt maximum RF power. Using 2.4 GHz module, 5 dB antennas could be used for transmitting data to a receiver in transmitter and receiver module with 100 mW RF power. The data is coded and then broadcasted to a predefined receiver node for security issues. The received data is plotted in MATLAB software after decoding. A Zigbee communication module is used for data transmission and the transmitter follows the IEEE 802.15.4-compliant coprocessor [6].

2. Design & Methodology

There are a few steps to complete this research and this project is coordinated into three sections which are the method of hardware design, the method of software design and the development of real time systems. First and foremost, the system starts by hardware components setup such as hardware design of a MedSAT payload, assembling the parachute with MedSAT and others. Second, GCS' development to interact and control MedSAT. Thirdly, MedSAT extracts real-time telemetry data. Finally, MedSAT design after the real-time system has been completed. The software and hardware development details of the project and the RTS system functioning process will be discussed in the next section.

Mission Planner is the ground station software selected for this research. Mission Planner is a comprehensive and most appropriate navigation control for the Ardupilot APM2.8 GCS application. This software will help the MedSat team navigate in the second phase when the MedSat payload rovers are deployed. With simple point-and-click way-point entry on Google or other maps, it can also plan, save and load autonomous missions into MedSat [12]. The GCS is able to access and analyze real-time application mission records created by MedSat. This project's flight controller is ArduPilot Mega V2.8, which works alongside Mission Planner and is suitable for GUI. The rife of features when

compared to the other software is the reason to consider Mission Planner as ground station software. It loads the firmware into the autopilot flight controller before connecting to the flight controller.

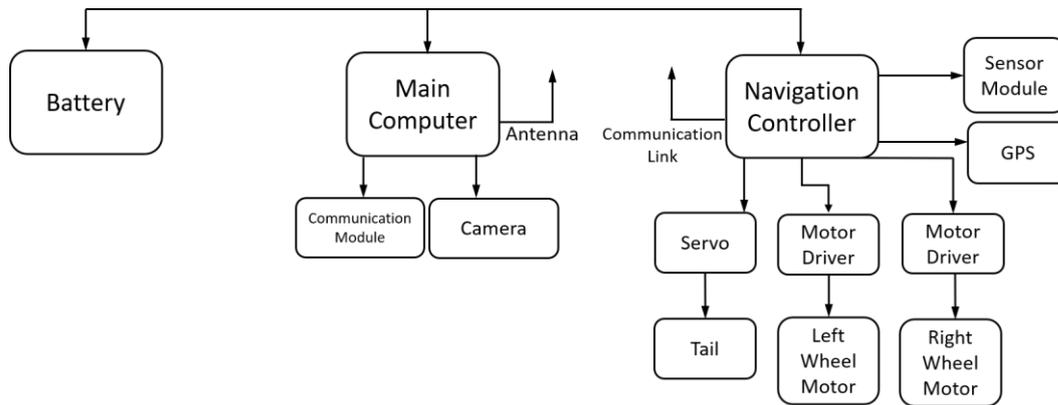


Figure 1. Block diagram of MedSAT

The uses of Mozilla FireFox Web Browser is to display the real time data system of MedSAT when its launched from 200 meter altitude. The data is from Mission Planner when MAVLink transferred the data from MedSAT to ground station. MAVLink IP address (<http://127.0.0.1:56781/mavlink/>) is being refreshed in Mozilla FireFox Web Browser as the data being transferred. Every time the web browser being refreshed, the data keep changing according to movement of MedSAT. The Mozilla FireFox Web Browser were chosen out of all web browser available because it has JSON Viewer, that helps ground station to have better GUI display to present the data collected by MedSAT.



Figure 2. ArduPilot V2.8



Figure 3. 3DR Radio Telemetry Kit



Figure 4. Raspberry Pi Zero W

The physical components of this project are described based on research requirements, including reasoning for choosing a microcontroller, aircraft controller, radio receptor, telemetry kit, GPS module, and battery. Ardupilot APM2 V.8 as shown in Figure 2 is a self-control navigation rover used when MedSAT lands on the ground. The flight data screen of Mission Planner will show you on a google satellite map where the MedSAT is and all the status information including a detailed heads up display. It allows the user to turn any fixed, rotary wing or multirotor vehicle (even cars and boats) into a fully autonomous vehicle; capable of performing programmed GPS missions with waypoints. It includes 3-axis gyro, accelerometer, along with a high-performance barometer. Furthermore, APM V2.8 consists of onboard 4 MegaByte Dataflash chip for automatic datalogging, and optional off-board GPS, uBlox 7M module with Compass. This kit came with a transmitter and receiver radio with 433MHz frequency band as shown in Figure 3. The transmitter kit will be attached at MedSAT when launching along with other payloads whereas the receiver kit will be attached at GCS to receive all information transmitted in real time basis. The telemetry kit act as a link between MedSAT and GCS. The telemetry data includes location of MedSAT (altitude, longitude and latitude), speed of MedSAT, humidity and pressure of surrounding atmosphere. The Global Positioning System (GPS) Module is a GPS navigation device that can receive information from satellites and then calculate the device's geographical position. Based on the research, real time system is required to capture the location of emergency beacon during search and rescue operation. The GPS module selected from Ubox 7M also comes with compass. This type of GPS module is suitable to work with flight controller. As an

independent on-board microcontroller, Raspberry Pi Zero W has opened more options than ever for the development of small computer. The W stands for wireless as it uses the wireless Bluetooth and Wi-Fi technology. Instead of using antenna, it utilizes Wi-Fi, enabling the computer to communicate with GCS. Low-cost and new connectivity for is perfect MedSAT. Raspberry Pi Zero W is a new Raspberry Pi Zero version that comes with the latest functions and smaller size including a built-in wireless internet. Raspberry Pi Zero W's low-level peripherals are 17 GPIOs plus the same specific functions and HAT ID buses. Raspberry Pi Zero W is shown in Figure 4.

In this study, a can-sized body that is a 7 cm diameter and 11 cm height cylindrical shell is required for carrying devices like flight controller, Raspberry Pi Zero W, GPS unit, 3DR Radio Telemetry Kit and other parts. In order to optimize the MedSAT design, the placement of components must be considered for the other component, so that there is no waste of space on the body. All the necessary components have no fixed position on this air frame except for the motors and rovers that had to be mounted at the end of each side of the MedSAT. This is because at the end of each arm the motors must be assembled to allow MedSAT to move freely on the ground.

As shown in Figure 5 is the incomplete MedSAT chassis, the chassis have two motors attached on both sides, and connection of both motors to the motor driver to regulate power for the motors to rotate. Figure 6 shows the MedSAT's rover tire that has been made by cutting of polymerizing vinyl chloride (PVC) pipe and round shape of aluminum cover from Pringles tub. The PVC spokes are tide by cable tie to secure the position of the spokes when it glued to the motors and to avoid the spokes from loose. The complete MedSAT with all components attached to each other as shown in Figure 6. Measuring tape shows in Figure 9 was as a tail for MedSAT to balance out MedSAT while it is moving on the ground. The first cover was layered by clear tape to ensure and secure the position of the components are not moving when launching and rover to target point. The outer layer of MedSAT was covered by manila card which is thicker than a normal paper to cover all the components as shown in Figure 6.

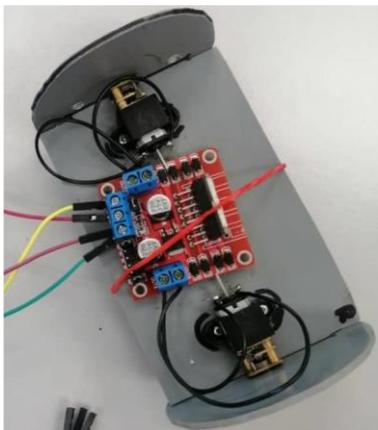


Figure 5. Incomplete MedSAT chassis



Figure 6. Complete MedSAT with all components attached



Figure 7. MedSAT's parachute

As shown in Figure 7 is a piece of equipment made of special cloth that is fastened to a MedSAT that is dropped from drone in order to to make MedSAT fall slowly and safely to the ground is called parachute. The parachute is used to slow down MedSAT when it is launched. There is no specific requirement for parachute selection, and for MedSAT launching, the big size of parachute was chosen to lessen the impact of MedSAT when it hits the ground. 0.5 kilograms weight of MedSAT were attached to the parachute when launching.

3. Results and Discussions

The MedSAT flight graph log as shown in Figure 8 shows MedSAT's airspeed from launching until it landed, MedSAT's altitude from the ground until it reached 200 meters and released, and finally MedSAT's pressure reading during the mission. The altitude reading in Figure 8 shown 204.17 meters above the ground means that MedSAT are ready for launching. Another telemetry data shown is pressure which is 98805 Pascal when it is on 204.17 meters above the ground. Airspeed of MedSAT is 5.63 m/s when MedSAT are drop freely from drone and when parachute takes over to slow down the MedSAT until it reached ground. Figure 9 shows the Mavlink Log Graph of MedSAT when it is connected between MedSAT and GCS, the red line data indicated the airspeed movement of MedSAT, in meter/second unit, altitude of MedSAT were specify as the green line data to show the altitude when MedSAT is launched and falling. Telemetry data which is pressure that collected by Ardupilot Mega V2.8 in the MedSAT recorded as shown in yellow line data in the Figure 9.



Figure 8. MedSAT at altitude 200 meters above ground

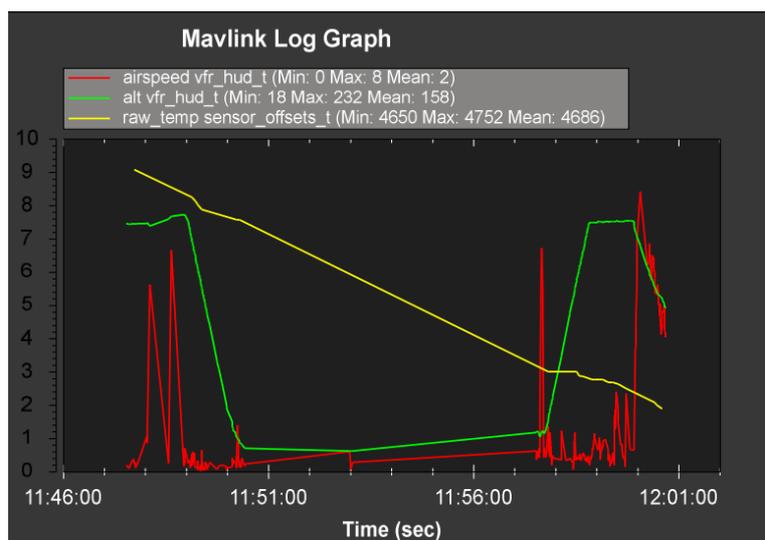


Figure 9. Graph log of MedSAT

JSON Viewer is a user-friendly program whose main task is to provide a convenient way of viewing JSON files' content and information. The Mozilla FireFox Web Browser is one of the popular

JSON Viewers that allows you to paste the JSON in the text space region and then look at the tree structure in the viewer section. Figure 10 shows the raw data gathered by MedSAT along the mission.

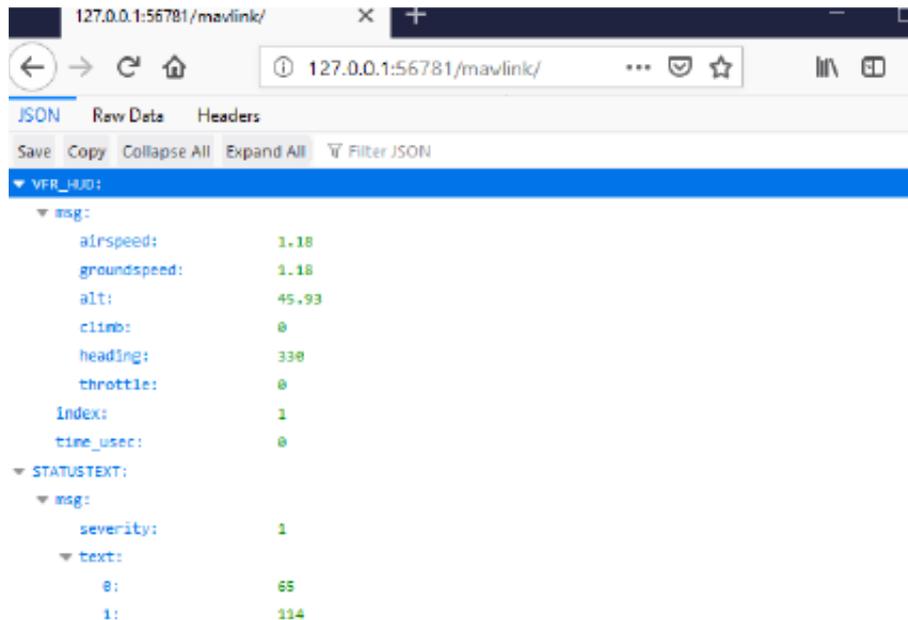


Figure 10. Data viewed in JSON Viewer

4. Conclusions

In conclude, the plan was successfully developed and achieved all the objectives. The project's real-time data will reduce unnecessary human risks and be able to quickly find victims. MedSAT's real-time system can locate MedSAT's real-time location and transmit data such as pressure on telemetry. In addition, all sensors provided reliable data, leading to scientific findings, and the GPS system provided a reliable and stable location for MedSAT. Last but not least, the parachute used to attach MedSAT is works perfectly.

References

- [1] S. Atoev and S. Lee, "Data Analysis of the MAVLink Communication Protocol," pp. 1-3, 2017.
- [2] C. Beker and A. Kirli, "Model Satellite Design for CanSat Competition," no. June, 2013.
- [3] M. E. Aydemir and R. C. Dursun, "GROUND STATION DESIGN PROCEDURES FOR," no. 1, pp. 909-912, 2013.
- [4] N. A. Dyrbye et al., "CFP-15-SG Can Science 2015 A solid CanSat taking on demand photos with simultaneously position determination - an approach to two- way communications and an interactive online presentation of data for teaching purposes," pp. 1-10, 2015.
- [5] M. Abo-arais et al., "Approaching a nano-satellite using CAN-SAT systems," pp. 813-817, 2015.
- [6] V. Bohlouri, A. Reza, and M. R. M. Aliha, "Systematic design of an atmospheric data acquisition flying vehicle telemetry system," vol. 2, pp. 265-276, 2014.