Design and Implementation of Attitude Determination and Control Subsystem for Cube Satellites

Ahmed Ehab Elsantawy, Mostafa ElFallal, Ibrahim Abuaitta, Shrief Elafify, Mohamed Zain Aldin, Eman Elbanna, Mennatallah Eldeeb, and Farida Alaa Okasha
Tanta University, Egypt, ahmedehaba192@gmail.com

Supervisor: Dr. Salah Khamis, Associate Professor
Supervisor: Dr. Hussein E. Seleem, Assistant Professor
Faculty of Engineering, Tanta University, Egypt

Abstract*– In this paper, we present a completely designed, programmed, manufactured, and tested attitude determination and control subsystem (ADCS) for the 3U form cube satellite (CubeSat). The proposed ADCS utilizes gyros, magnetometers, and sun sensors for attitude determination, and was designed to be a simple, low-cost solution to properly meet the attitude determination requirements. Meanwhile, a magnetorquer and a reaction wheel were utilized as the actuators for the control process. The accomplished work comes in forms of completely designed hardware, developed software, testing environment, and manufactured structure for the CubeSat. Firstly, regarding the designed hardware, we selected the suitable controller and electronic components for the subsystem board, implemented the required circuits, designed and fabricated the Printed Circuit Board (PCB) then assembled the components. Secondly, we built software drivers for all peripherals that are used for ADCS. These drivers include communications between the microcontroller and external devices using an agreed protocol in all subsystems, communication error detection Cyclic redundancy check (CRC), and building software to control the CubeSat actuators. Moreover, all software was combined in a real-time operating system (FreeRTOS) to be ready for the algorithm that is responsible for the amount of required movement. Thirdly, we used LabVIEW software and MATLAB software for verification, checkout, and testing to make sure all functions work properly. Finally, we designed and manufactured the structure of the 3U CubeSat.

***Keywords*— Attitude Determination, Attitude Control, CubeSat Nanosatellite, ADCS, Sensor, Actuator.**

I. Introduction

Owing to their affordability and low time of deployment, CubeSat technologies have been diversely studied and developed by universities, companies, and space organizations all over the world. CubeSats provide a relatively cost-effective and rapidly developed means to perform technological and scientific studies in space. Over the past two decades, the aerospace industry has experienced massive growth and a tremendous boost in the number of CubeSats developed. This growing interest is encouraged by their proven ability to provide and allow a wide range of services and applications, such as earth observation, remote sensing and imaging, mobile communications and IoT, geolocation and navigation, and a broad range of other scientific applications [1, 2].

Despite its small size, a CubeSat contains all of the sub-systems found in the largest satellites. It is comprised of multiple subsystems that are responsible for carrying out its mission. Fig. 1 displays an exploded view of a typical standard CubeSat model revealing all its subsystems [3]. In order to maintain, monitor, and control the satellite's attitude in space, the satellite needs an Attitude Determination and Control System (ADCS). Moreover, in remote sensing or imaging satellites, the ADCS is particularly critical to allow taking images with high quality and to guarantee achieving better communication [4].

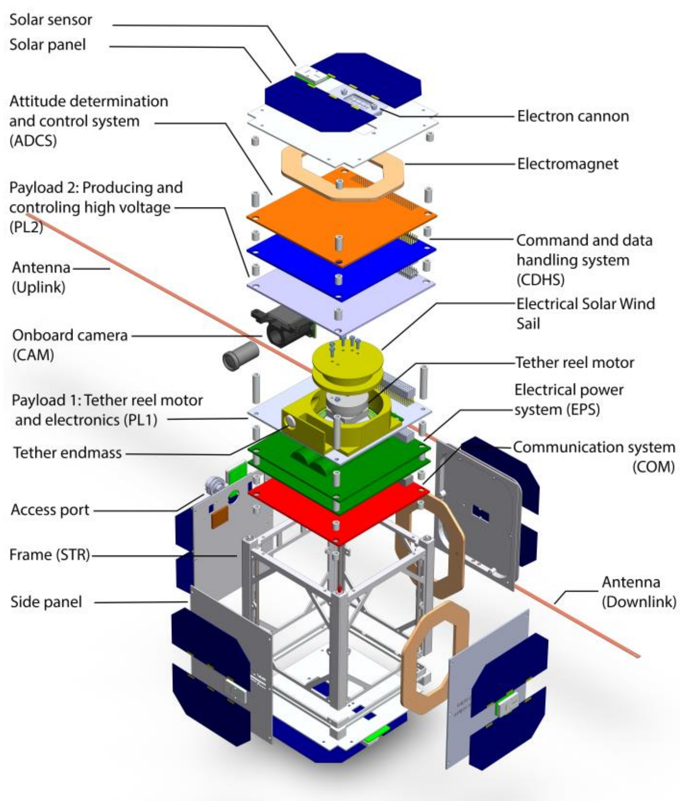

Fig. 1 Exploded view of a typical standard CubeSat model displaying all its subsystems [3].

The ADCS comprises hardware such as sensors, actuators, and the ADCS board which acts as the ADCS processing unit. The ADCS also includes software that has been coded or programmed into the ADCS processor. This software contains all the control schemes or algorithms of the ADCS [5]. So, on a broad scale, it can be observed that there are two major components that make up the ADC subsystem, which are hardware and software. Due to the unreliable nature of the inexpensive hardware and software that may be used in such a project, it is necessary to conduct simulations of hardware, software, and the entire ADCS to ensure that the system can cope with likely software and hardware failures. Therefore, a possible software and hardware testing approach will be outlined here. Areas of testing that could be considered in these tests are situations of flawed sensor readings and actuator failures for instance. Also, the overall ADCS system operation should eventually be tested using LABVIEW or MATLAB, such as the one being designed in this project, to better recreate conditions in the actual orbit or flight environment.

In this paper, we present and describe the complete design, development, implementation, and testing of the ADCS. A detailed description of the complete work is presented in subsequent sections. Section II describes the designed and selected hardware including all its peripherals. Section III presents the developed software. Section IV describes the verification and testing process. Section V shows simulations and testing results of the overall manufactured design. Finally, Section VI presents the conclusion.

II. HARDWARE

The ADCS functionality can be described as two processes, estimating the current attitude or orientation and giving a command to control. These processes require a number of peripherals including sensors and actuators. The first task of the ADCS Board is to estimate the current attitude (or orientation) of the satellite by a set of sensors including magnetometers, gyroscopes, and sun sensors. Secondly, the attitude controllers command the magnetorquers to control the satellite's attitude to a certain desired orientation. A more detailed description of these peripherals will be covered in the next subsections. Thereafter we present a closer look at the configuration and circuit design for each one of them.

Our ADCS board has three types of attitude sensors onboard. The magnetometer is used to aid in detumbling. The gyroscope is mainly used for attitude computation of the CubeSat, once it is detumbled and stable [6]. Finally, the sun sensors are replaced by photodiodes due to the cost and availability of sun sensors [7]. The main function of sun sensors is to determine the orientation of the satellite in relation to the sun.

A magnetometer is used to measure magnetic field strength. The ADCS board contains LIS2MDL and LIS3MDL magnetometers with 3-axis coils. Moreover, the ADCS board includes IAM-20380 and A3G4250DTR 3-Axis gyroscopes. The photodiodes supply the board with values that reflect the intensity of sunlight. These values are used to determine the orientation of the satellite in relation to the sun.

After we discussed the ADCS peripherals previously in detail, we give a closer look at the designed circuits of the whole system. The first step in the design process is performing the schematic design of the system and then converting it into a Printed Circuit Board (PCB) layout. Due to the size of the project, the schematic design is divided into six parts or sheets. Each sheet of these sheets contains a part of the project or several parts related to each other. These sheets are divided as follows:

A. *Microcontroller and J-Tag sheet*

The microcontroller sheet represents the bulk of the project because it contains a number of auxiliary circuits to ensure that the microcontroller operates correctly and reliably. This sheet is divided into several parts, which are Microcontroller IC, Power Stabilization circuit, Crystal Oscillator circuit, J-Tag circuit, SWD circuit, and BOOT circuit.

B. *Interface Sheet*

The interface sheet contains two parts which are a 52 Pin Stack Header and an RS-422/RS-485 Transceiver. The previous two sheets are displayed together in Fig. 2.

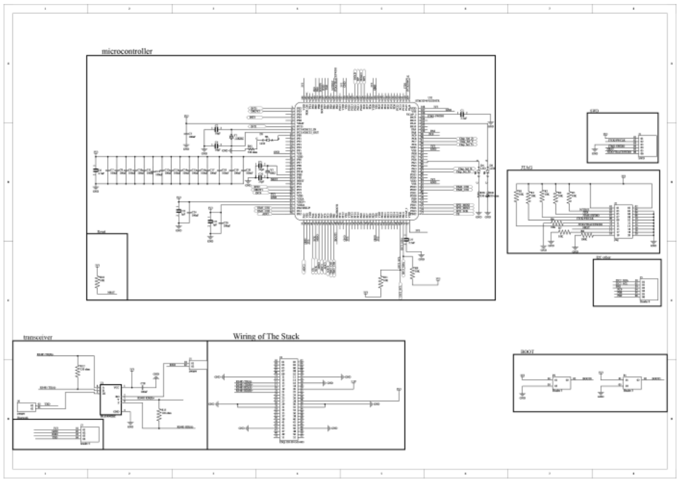

Fig. 2 Microcontroller and Interface sheets.

C. Sensors Sheet

This sheet contains the circuits of five sensors, a temperature sensor, two gyroscope sensors, and two magnetometer sensors, all shown together in Fig. 3.

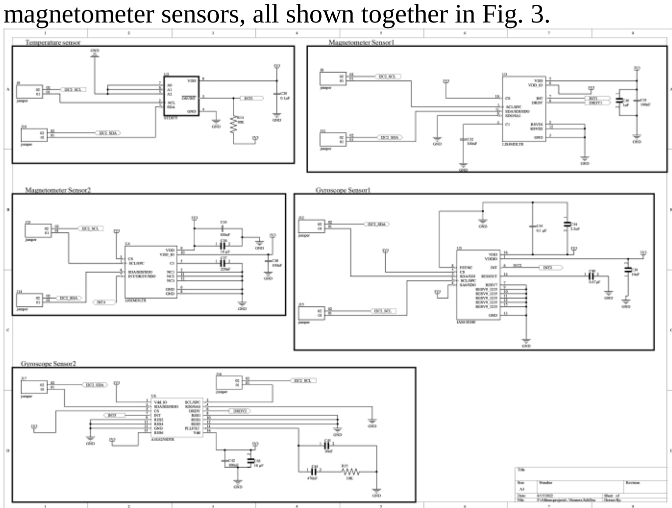

Fig. 3 Sensors sheets.

D. Photodiode Circuit Sheet

This design uses a 256-element linear photodiode array, which is both flexible in operation, by allowing the distribution and magnitude of light to be determined, and efficient by utilizing only a single line of sensing elements rather than a matrix such as a Charge-Coupled Device (CCD) array. Here we used just one module as an amplifier which is LMP7704MT and designed its schematic as shown in Fig. 4.

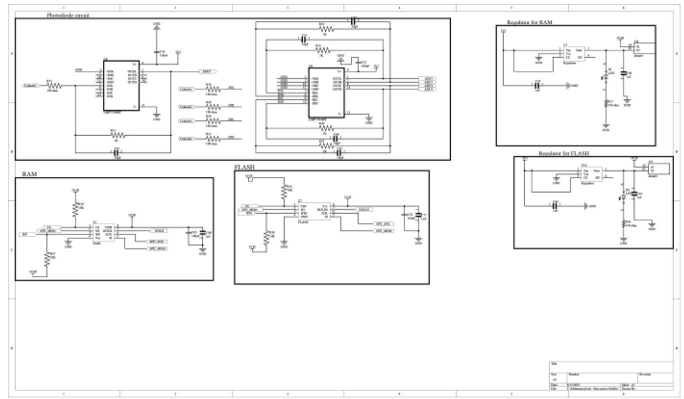

Fig. 4 Photodiode circuits and Storage system sheet.

E. H-bridge circuit and Solar Panels Interface Sheet

This sheet, shown in Fig. 5 has three types of components mainly used to control the magnetorquers, which are H-Bridge circuits, Picoplade 12-pin connectors, and Magnetorquers header.

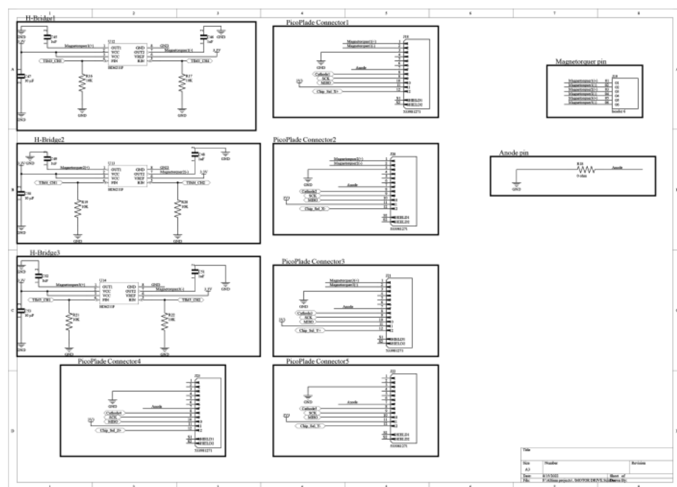

Fig. 5 H-bridge sheet.

F. Storage Sheet

This sheet has three types of components mainly used to store data. These components are RAM, FLASH, and SDRAM. Their circuits were displayed in Fig. 4.

Finally, after finishing all sheets, the Printed Circuit Board (PCB) is designed. The design process is carried out in different steps which are, designing the shape and dimensions of the board, layer stack management, adding footprint to all components, checking design rules for PCB, Placement of components, routing of the board, and checking error.

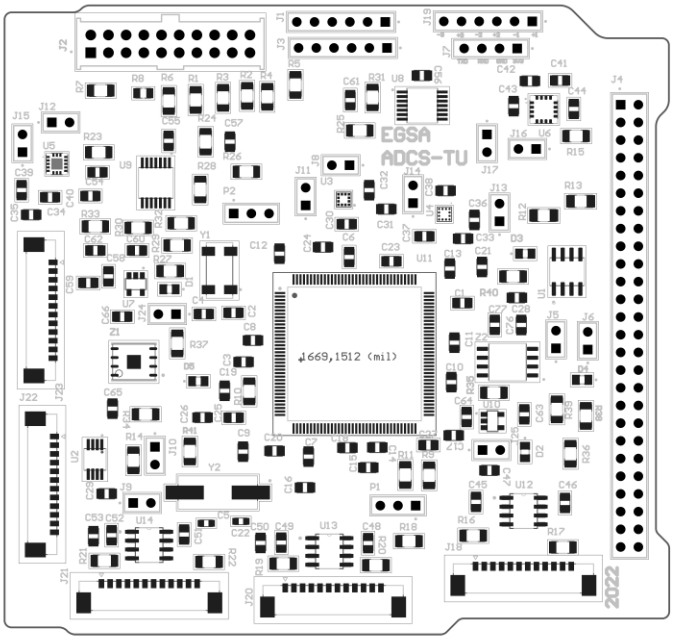

Fig. 6 Placement of components.

Using Altium Designer software, the design process was carried out successfully as described above. The design process was completed by the placement of components as depicted in Fig. 6, then routing was performed developing the final board as displayed in Fig. 7.

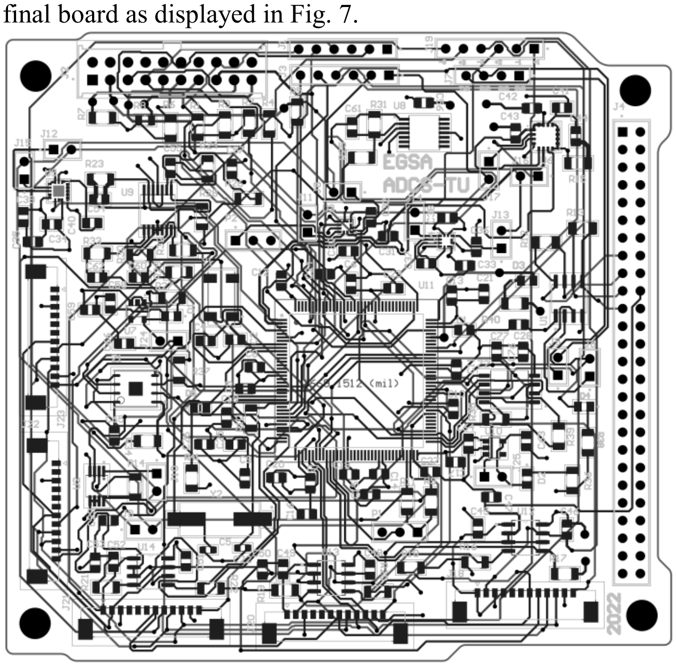

Fig. 7 The designed PCB after routing.

Finalizing our work, a 3u form CubeSat structure was designed and fabricated to integrate the system and its peripherals within it. Assembling all the system parts in the designed structure as shown in Fig.8, has facilitated the testing process which will be detailed in the next sections.

Fig. 8 The fabricated 3u form CubeSat structure assembling the overall system parts.

IV. Software

In this section, we discuss the ADCS software that runs on the microcontroller. We will describe how our system is communicating with the external environment, explain the way the requirements of the ADCS software are achieved, and go through the design describing the higher layers such as the application layer and services layer. The layers will be described in a top-down view, meaning that the lower layers like Microcontroller Abstraction Layer (MCAL) and Hardware Abstraction Layer (HAL) will be discussed along with hardware protocols and their functionalities.

A. *Software Design*

In this subsection, the important point is to understand how the program flows, the steps may be a little complicated however, the whole program can be simplified into these general steps: receiving the command frame, checking the validity of the frame, executing the command and transmitting reply frame. Meanwhile, there are other tasks in the background doing some more calculations. The mentioned frame is a part of a software protocol called Simple Serial Protocol (SSP), as the whole program is essentially depending on the SSP it is the first point that will be discussed.

a) SSP: SSP is used in our subsystem to communicate with other subsystems. It is built over the UART and as shown in the hardware section the UART is converted to RS-485 which is a multi-drop bus. SSP transmission and bus handling, framing, transactions and bus access, forwarding, and packet format, all have been studied and investigated. SSP frames are variable length with byte stuffing framing technique. The following table shows its frame structure.

TABLE I
Basic Format of an SSP Packet.

FEND	***DEST***	***SRC***	***Type***	***Data***	***…***	***CRC1***	***CRC0***	***FEND***

Every field is a byte where the FEND is the hex value “0xC0” which is replaced to two bytes “0xDB 0xDC” if it appears in data fields, where the value “0xDB” is called FESC and is replaced with the two bytes “0xDB 0xDD” if it appears in data fields. Frames may have no data fields and there is no limit for the maximum, however, our design limited the maximum to 100 bytes of data.

b) Application Layer: The application layer is the highest layer in the software, containing UART Handler, Receiver Task, Transmitter Task, Log Saver, Update Telemetry Task, Latch Task, and Frame Provider. Each of them is responsible for a valuable function in software. All of them are combined using Real Time Operating System (FreeRTOS). Giving them equal priorities makes them run in Round-Robin algorithms. So, they appear as if they are working in parallel. This gives the software the ability to have a foreground application that is running in front of users like interacting with frames and a background application that is used to do some more calculations. Starting with the function, UART Handler which is not a task it is an Interrupt Service Routine (ISR) This function is executed only if there is a received byte over UART.

V. Checkout and Testing

Appropriate system integration is essential to ensure compatibility among all CubeSat subsystems, which defines the importance of the verification and testing process to make sure that communication between the developed ADCS and the rest of the system is achieved [8]. After the design and fabrication of the ADCS board, we try to test the board, its sensors, and its response with other subsystems. As known the On-Board Computer (OBC) subsystem is responsible for managing everything in the CubeSat, so we tried to simulate the OBC function and the way it is working by using software like LabVIEW and MATLAB.

A. LabVIEW

Using LabVIEW software, an interactive front panel was designed and programmed with needed algorithms and functions to control and monitor the ADCS response during the checkout and testing process. The designed front panel provides a serial setting that allows various options such as controlling the baud rate to determine the speed of communication over the specified port and the serial wires, error detection, and flow control which manages the rate of data transmission between two nodes to prevent a fast sender from overwhelming a slow receiver. Also, to get control over the ADCS board, a set of commands is programmed such as:

1) Ping: to check the connection with the ADCS board.
2) Read Magnetometer: to get Magnetometer reads.
3) Read Gyroscope: to get Gyroscope reads.
4) Read Temperature: to get Temperature sensor reads.
5) Read Sun Sensors: to get Photodiodes reads.
6) Telemetry: to get all sensors reads.
7) Write Pulses: to send the values of PWM to motors.
8) Get Motors: to read the speed and directions of motors.
9) Write Memory: to send and store data in the RAM.
10) Read Memory: to read data from the memory.

In addition to these mentioned options and commands, a Cyclic Redundancy Check (CRC) is added for data validity check.

B. MATLAB

System testing and validating are core elements besides the development process [9]. Therefore, our team developed CTE software to test every point in the system and validate that the performance is what is meant to be. The Checkout and Testing Equipment (CTE) is developed using MATLAB app designer.

First of all, the connection between the CTE and the CubeSat must be checked before proceeding to the actual testing page. The connection is performed via a Bluetooth module which is connected to the specific serial port in the testing machine, to complete the connection process, we must choose the serial port and the baud rate. This connection sends Ping frame and receives the reply if it arrived properly the connection status becomes green with the word “Connected” and then proceeds to the main page. If it didn’t get a reply, it times out and asks to check the serial port. After that it asks the system which sensors are active, the CubeSat replies with the Who I Am register value of each active sensor.

IV. Simulations and Testing Results

The performance of the overall system is verified through the prementioned developed checkout and testing software to simulate the board response. Users can control the CubeSat and/or monitor the data from the MATLAB apps of the developed MATLAB CTE.

A. Simulation app

This app simulates Euler’s angles in three dimensions (Pitch, Yaw, and Roll). This is done by requesting the angles from the CubeSat and rotating the cube Fig according to these angles.

B. Gyroscope app

The gyroscope app plots the three angular velocities with time.

C. Magnetometer app

This is like the Gyroscope APP but this plots the magnetic field.

D. Sun Sensors app

This Plots the voltage across the photo resistors and displays it as a function of time.

E. LOG file

This app is used to monitor all the commands sent to the ADCS, it’s useful for debugging sessions and checking the performance over the last 256 commands.

F. *Memory APP*

By providing 131,071(0x1 FFFF) Bytes of non-volatile memory, the OBC can store any data in these bytes and use it again in case the power is down. The ADCS provides 5 functions for this memory which are read, write, erase, lock and unlock. These functions give the OBC full control over the memory.

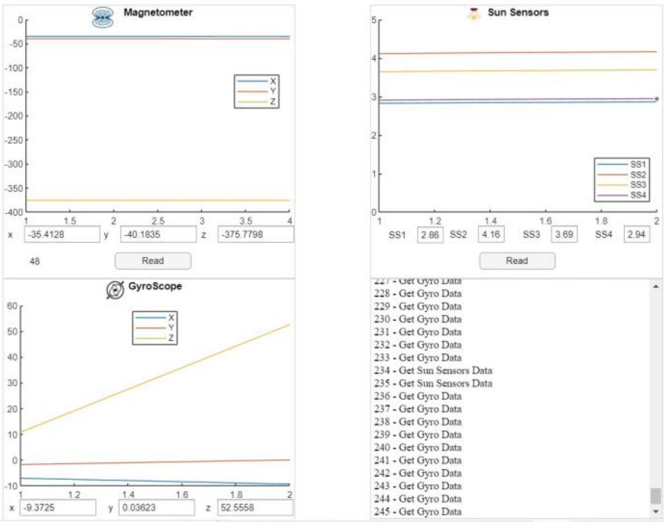

Fig. 9 Sensors reading during a steady state of the CubeSat.

F. *Memory APP*

By providing 131,071(0x1 FFFF) Bytes of non-volatile memory, the OBC can store any data in these bytes and use it again in case the power is down. The ADCS provides 5 functions for this memory which are read, write, erase, lock and unlock. These functions give the OBC full control over the memory.

G. *Actuators APP*

This APP allows the user to control the 3 actuators, each of them can be controlled with a speed percentage and direction. Each wheel can be set to a value from -100 to 100 which specifies the speed and direction of the actuators.

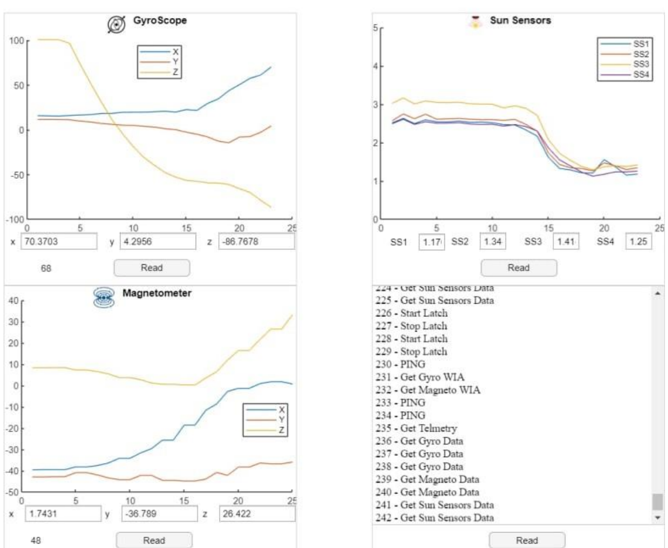

Fig. 10 Sensors reading during a rotation movement of the CubeSat.

For a clear investigation of the system response using the developed CTE, sensors readings were observed during a steady state of the CubeSat as shown in Fig. 9, and also, during a rotation movement as depicted in Fig. 10. On the other hand, the actuators were tested to control the CubeSat and execute specified actions.

VII. Conclusion

In this paper, a completely designed, manufactured, and tested attitude determination and control subsystem for CubeSats was presented. The proposed ADCS makes use of different types of sensors and actuators as it utilizes gyros, magnetometers, and sun sensors for attitude determination while using a magnetorquer and a reaction wheel for the control mission. The design, development, implementation, and testing processes were described in detail and with a logical flow. Starting with the designed hardware, we presented a complete description of all its selected peripherals of sensors and actuators, besides presenting the design process of developing the PCB. Also, we introduced our work regarding the developed software that achieves the communication of our system with the external environment and integrates all software in a real-time operating system (FreeRTOS). Then going through the verification and testing process of the entire implemented system, the compatibility of the used sensors and actuators into the ADC subsystem, and the integration and interaction between our system and other CubeSat systems all have been totally verified. Finally, we presented the simulation and testing results of the manufactured system board.

Acknowledgment

We would like to express our deep thanks and great appreciation to our major advisors Dr. Salah Khamis and Dr. Hussein E. Saleem for their constant encouragement, patience, and supervision and for giving us the opportunity to carry out such innovative and challenging research. We also acknowledge the funding of our project by the Egyptian Space Agency (EgSA).

References

[1] Cappelletti, Chantal, and Daniel Robson. "Cubesat missions and applications." In Cubesat Handbook, pp. 53-65. Academic Press, 2021.

[2] Camps, Adriano. "Nanosatellites and applications to commercial and scientific missions." Satell. Mission. Technol. Geosci (2020).

[3] Lätt, Silver, Andris Slavinskis, Erik Ilbis, Urmas Kvell, Kaupo Voormansik, Erik Kulu, Mihkel Pajusalu et al. "ESTCube-1 nanosatellite for electric solar wind sail in-orbit technology demonstration." Proceedings of the Estonian Academy of Sciences 63, no. 2 (2014): 200.

[4] Faizullin, D. "Attitude Determination and Control System for the first SAR Satellite in a Constellation of iQPS." In *Proceedings of the 71st International Astronautical Congress (IAC), Dubai, United Arab Emirates*, pp. 12-14. 2020.

[5] H. A. A. Alarazah and P. Teofilatto, "Theoretical and numerical analysis of the attitude control of a 3u cubesat-iraqi satellite (tigrisat)," Association of Arab Universities Journal of Engineering Sciences, vol. 25, no. 4, pp. 236–257, 2018.

[6] He, Liang, Wenjie Ma, Pengyu Guo, and Tao Sheng. "Developments of attitude determination and control system of microsats: A survey." Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 235, no. 10 (2021): 1733-1750.
[7] Jones, Windell. The Design and Simulated Performance of the Attitude Determination and Control System of a Gravity Gradient Stabilized Cube Satellite. Lulu Press, Inc, 2015.
[8] Y. Rahnamai, Kourosh, Thomas Searles, and Ryan Parker. "Design and development of a low-cost CubeSat attitude control system testing platform." In 2018 IEEE Aerospace Conference, pp. 1-6. IEEE, 2018.
[9] Olsen, Jørgen Anker. "Attitude Determination and Control System Testbed for Hardware and Software Testing and Verification for Small Satellites." Master's thesis, NTNU, 2021.