Date available online: 20-09-2024
Vol. 2, Issue 1 (January – June 2024)

Floating Satellite with Ultrasonic Radar Payload for Debris Detection

Khubaib Ahmad[1*], M. Kamran Saleem[1], Sergio Montenegro[2], Muhammad Faisal[2], Awais A. Khan[3]

[1]Electrical Engineering Department, University of Central Punjab Lahore, Pakistan
[2]Aerospace Information Technology Department University of Wuerzburg Wuerzburg, Germany
[3]Mechanical Engineering Department University of Engineering and Technology Lahore, Pakistan

[*]Corresponding author: Khubaib Ahmad (e-mail: khubaibahmad@ucp.edu.pk)

ABSTRACT An educational platform designed for assessing and testing control algorithms for pico and nano satellites in a near-frictionless environment is presented. The system provides access to test different space application algorithms. The application provides here is the debris detection mission for space surveillance. Different payloads are added on FloatSat, integrated with the Real-Time operating system (RODOS) and STM32F407G as the main controller to detect the debris around the satellite in a specific range.

INDEX TERMS Satellite subsystem, RODOS, Aerospace, Satellite mission, Satellite test, Evaluation, Payload integration.

I. INTRODUCTION

In the field of space technology, many test benches were established to simulate and evaluate Nano/Pico satellites like a three-degree-of-freedom platform for small satellites [1]. CubeSat simulator [2], the NanoSAT platform [3]. A FloatSat [4] is a single-degree-of-freedom test bench to test and evaluate real satellite algorithms. The FloatSat platform is especially designed for engineering students to assess the performance of satellite control algorithms in a real-time in space like environment [5]. This educational platform helps to simulate different space missions and check the algorithm's stability. As dozens of satellites send to space to orbit around the Earth, to prevent collision between the spacecrafts the space surveillance is crucial. The MeerKAT radar is especially designed to predict and perform sensor scheduling for orbit determination accuracy [6]. This article's main focus is the application of FloatSat, the prototype space mission for detecting debris in the orbits around the Earth. The proposed platform for the prototype of the space Application is shown in Fig 1. This platform is a combination of the electrical and mechanical sections. The mechanical support provides a frictionless environment and the electrical section provides the control of a satellite's attitude and payloads attached for specific space applications. This algorithm for debris detection system is implemented on the real-time operating system (RODOS) that rotates the sensor 360 degrees to detect the object just like the observatory satellites in the lower earth orbits.

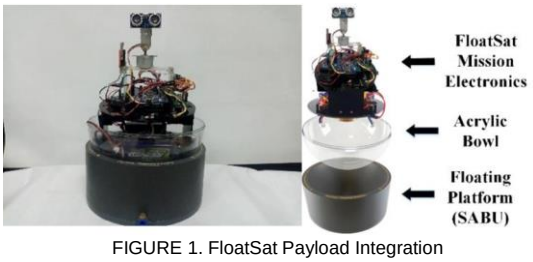

FIGURE 1. FloatSat Payload Integration

In Section II the modified FloatSAT hardware for the Debris Detection mission is described. Section III briefly describes RODOS, processing IDE, and Arduino IDE which is followed by the functionality of the Application such as Debris Detection with FloatSat, wireless data extraction, and visualization on H-term and Processing IDE respectively along with the results in Section IV. Finally, Section V concludes the FloatSat mission.

II. HARDWARE PALTFORM

One of the important subsystems in FloatSat platform is its spherical air-bearing unit (SABU) that provides a frictionless environment to simulate a space like environment. The other components include sensors for the Attitude Heading Reference System to implement the satellite simulator[7].

Figure 2 shows the radar formation as an initial step for the FloatSat mission using Arduino, Stepper motor & ultrasonic sensor. The visualization is on processing IDE. First, the system moves towards the limit switch to take the initial position, then moves 360 degrees back and forth repeatedly.

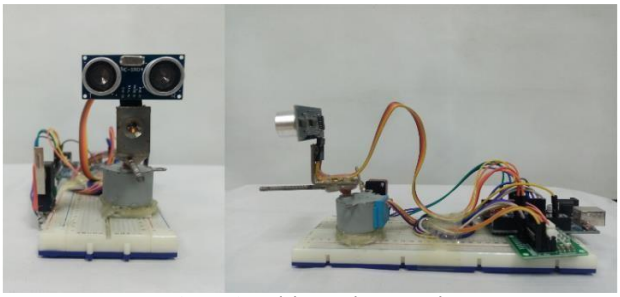

FIGURE 2. Arduino Radar Formation

Figure 3. Shows the basic components to implement the debris detection Radar system application on the Floating Satellite. Two Lithium Iron Phosphate

(LiFePo4) batteries are installed in FloatSAT to provide power to various subsystems in the platform. These batteries generally provide heavy current, so to protect the FloatSat components, a DC-DC Step-Down voltage regulator with a constant 5V output is installed in the way to power up the electronics. HC-06 Bluetooth module operating at 2.4 GHzfrequency is simply used for wireless telemetry of FloatSat overthe range of 100 meters.

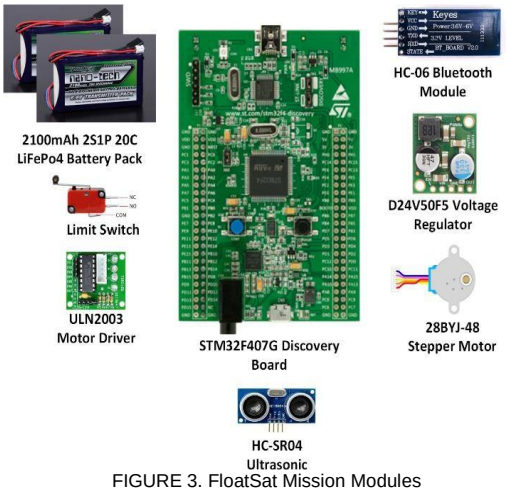

FIGURE 3. FloatSat Mission Modules

RODOS is running on STM32F407G, a controller that handles all the functionality of FloatSat and its applications. A 28BYJ-48 stepper is a 5-wire unipolar motor that runs on 5 volts. It canbe precisely positioned one step at a time with 4096 steps per revolution. The RPM of the motor can be changed by providinga delay in each step. ULN2003 is the driver module used to drive a stepper motor. The input for this controller is given by the microcontroller. This input is in the sequence mentioned torun the motor in the datasheet. The output is then given to the motor accordingly. HC-SR04 is an ultrasonic sensor used to measure the distance of a distant object. It provides 2- to 400cmnon-contact measurement with an Ultrasonic transmitter, receiver, and control circuit. A limit Switch at a fixed location is used to locate the initial position of for detection sensor.

Figure 4. Demonstrates the different views of mission structure implemented on Floatsat.

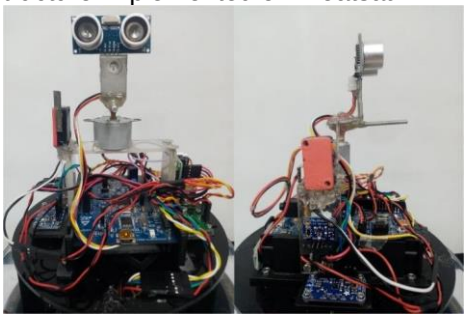

FIGURE 4. FloatSat Mission (Debris Detection) Formation

III. SOFTWARE PLATFORM

The Real-Time On-Board Dependable Operating System (RODOS) is the software framework for FloatSAT. Parallel running threads make the RODOS highly efficient for systems that demand high dependability. The RODOS is developed by DLR for the STM32F407G discoveryboard. RODOS is operational in almost 10 satellites that are orbiting around the Earth. The software components in RODOS adjust each other to provide dependable computing [8].RODOS controls both the operating system (OS) and microelectromechanical system (MEM). The RODOS operating system is governed by 48 library header files. The Application Programming Interface (API) retrieves data and communicates with external hardware and software components.

The initial step for the FloatSat application is to implementit on Arduino IDE. Arduino IDE is open-source software that ismainly used for writing and compiling code into Microcontrollers [9]. Processing IDE is used to display the measured data in graphical form in real time. It usually uses Java to write the code for a graphical user interface.

IV. APPLICATIONS

This section describes the space mission implemented on FloatSat. The details of each step that leads to the Application are discussed below:

A. Radar Implementation on Arduino UNO

A radar is a sensor used to locate or track a distant object. The prototype of the small radar system is first implemented onan Arduino board [10]. This system also includes Stepper Motor with the motor driver, an Ultrasonic sensor, and a limit switch.

The ultrasonic sensor is mounted on the stepper motor that rotates 360 degrees and then processed by Arduino. A long screw is also fixed with the motor to take the initial position. Thelimit switch is placed in the initial position. When the system isswitched ON, the motor rotates towards the limit switch, and by pressing the limit switch the motor recognizes the initial position and rotates 360 degrees back and forth until the system shut down. The ultrasonic sensor measures the distance of the objectwithin the range.

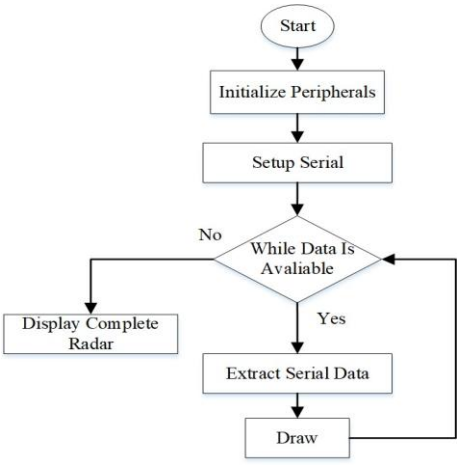

FIGURE 5. 360 Degree Radar on Arduino IDI

Figure 5. Describes the functionality of the program to implement radar as follows:

- After switching ON the system, the radar is set to its initial position until the limit switch is pressed.
- If the ultrasonic and stepper pins are initialized completely, it moves toward the main loop, otherwise,end the program.
- The loop is running until the system shuts down.
- Radar is rotated in both clockwise and anticlockwise directions with the calculation of distance and angle, and print data on a serial monitor.

Processing IDE is used to capture the real-time display. Figure 6. illustrates the algorithm running on processing IDE asfollows:

- After running the program different peripherals are initialized in the library file included in the program.
- Setup Screen Resolution & Serial Information.
- Data is not extracted for a while when the program starts. This is done to display the complete radar onthe screen.
- If data is available on the serial port program extracts the serial information data that receives on the communication port and integrates this information with the draw functionality i.e radar, text, line, and object.

Figure 7. Demonstrate the real-time obstacle detection system. After taking the reset position, the motor with an Ultrasonic sensor rotates 360 degrees. If the obstacle is in the range theradar system shows the red mark with the distance of the object. The total range is about 50cm for the current visualization system.

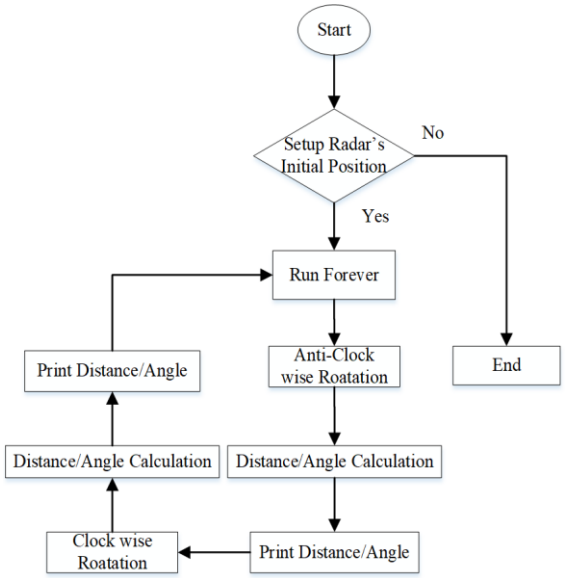

FIGURE 6. Algorithm of Processing IDE

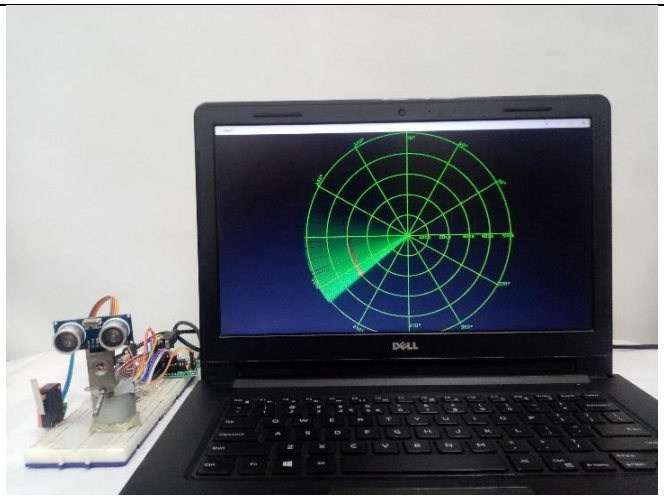

FIGURE 7. Radar Display on Processing IDE

B. *Radar Implementation with STM32F407G*

As STM32F407G uses RODOS, radar implementation withthis controller is crucial to replicate the space debris detection mission that requires high dependability. Two threads are running on the RODOS framework to implement the functionality i.e. Ultrasonic and Radar_360. The Description for these threads is as follows:

1) Ultrasonic Thread

The flow chart associated with the ultrasonic thread is shown in figure 8.

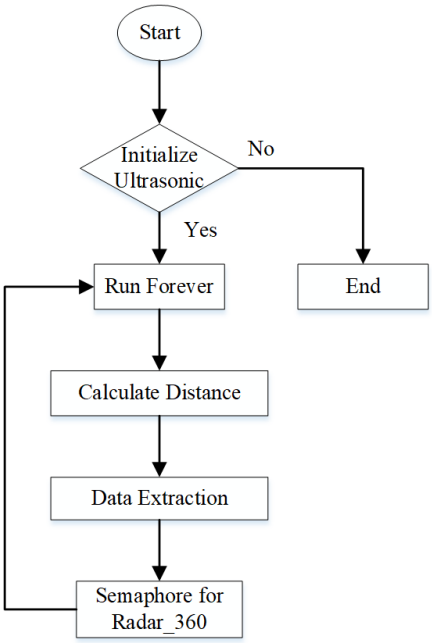

FIGURE 8. Algorithm of Ultrasonic Thread

- Ultrasonic thread class computes distance parameters.
- Initialize system pins i.e. Echo & Trigger for Ultrasonic sensor. This is necessary to link ultrasonic with the RODOS library.
- If the sensor is initialized completely, run the loop, otherwise, end the program.

- Distance is calculated by sending the 10-microsecond signal on the trigger pin and capturing the reflected waveon the Echo pin.
- Loop is running to extract the distance measurement data.
- Make a semaphore for inter-thread communication. Thedistance measurement data is then utilized in Radar_360Thread.
- Radar_360 thread implements the prototype mission fordebris detection.
- Initialize peripherals and reset the radar to its initialposition.
- If initialization is complete move to the main loop,otherwise end the program.
- First radar rotates in an anticlockwise direction.
- Angle & Distance is calculated and sent to the wirelessserial port.
- Clockwise drive with angle/distance measurement andsend data to wireless UART.
- Distance data comes from the ultrasonic thread viasemaphore.

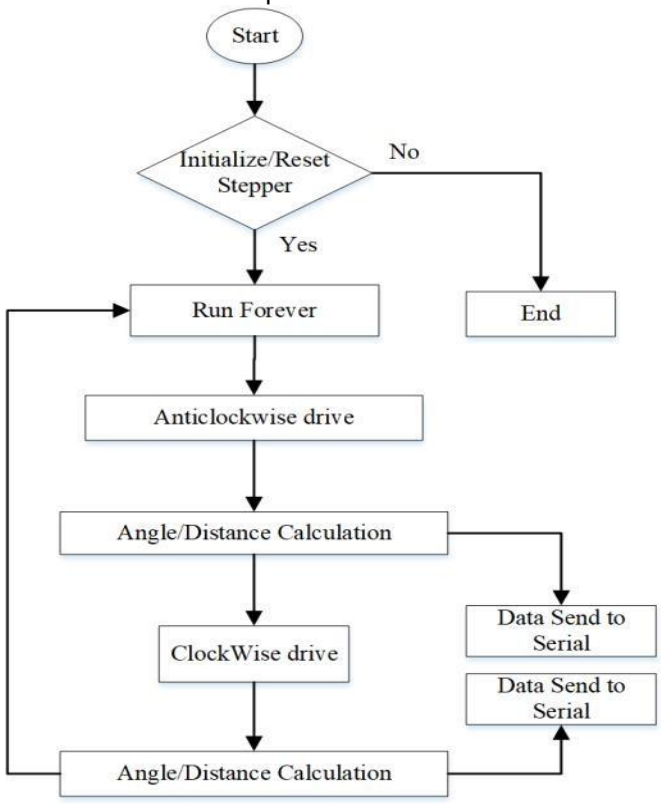

FIGURE 9. Algorithm of Radar_360 Thread

2) Radar_360 Thread

The purpose to implement this thread is to make a round angleof 360 degrees for the stepper motor and extract angle and distance measurement. The flow chart for Radar_360 Thread isshown in figure 9.

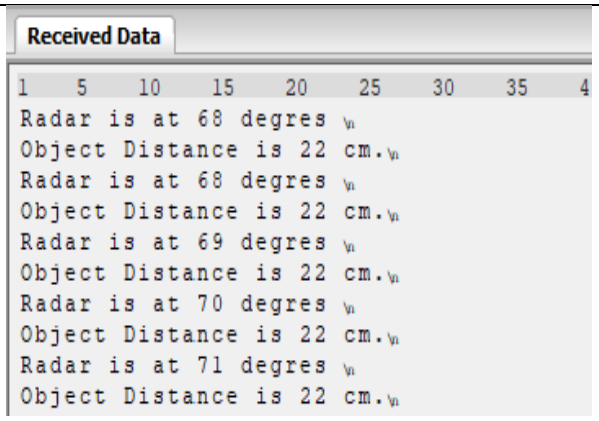

FIGURE 10. Telemetry on Hyper-Terminal

Figure 10 shows the radar output containing the Radar angle andobject distance. The data is sent via Bluetooth module on the Hyper-terminal.

Figure 11 Describe the prototype scenario for the debris detectionmission. Make a 360-degree radar on the chart paper.

FIGURE 11. Prototype mission Scenario

Each circle is 10 cm apart, the same as the one on the graphical interface in processing IDE. Different obstacles are placed at different distances from the radar to measure the distance. When the system is switched ON, the Sensor goes back to its initial zero degrees position. It finds the first object at 30cm, makes a trip of 180 degrees, and finds the second one at 20 cm.The third is placed at 270 degrees and the fourth is just before 360 degrees with a distance of 45cm and 40cm respectively.

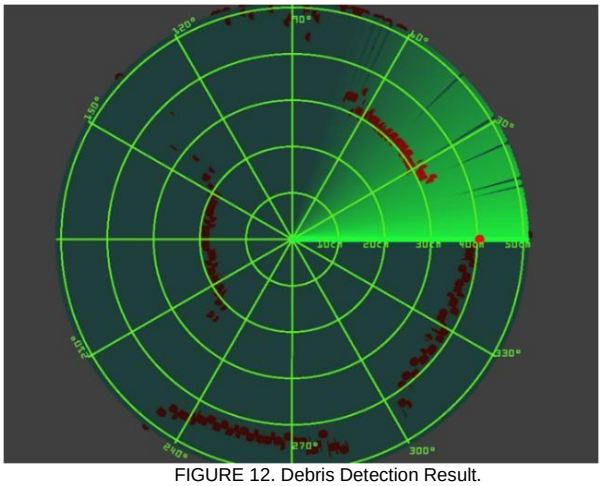

FIGURE 12. Debris Detection Result.

Figure 12 shows the visualization of the debris detection mission implemented on a processing IDE. First Radar moves towards its initial set point and then complete 360-degree rotation in both clockwise and anticlockwise direction. The detection criteria depend upon the resolution of the detecting sensor. This graphical visualization matches the prototype scenario on the chart paper.

V. CONCLUSION

The FloatSAT platform is modified for a debris detection mission. The modified platform can be utilized to detect the nearby objects in vicinity of 50 cm with full azimuth plane coverage. The modified FloatSAT platform can be very efficiently utilized by students to develop basic understanding of space debris detection.

VI. ACKNOWLEDGMENT

The authors would like to extend their sincere appreciation to the chair of computer science VIII, Aerospace information technology, University of Wurzburg, Germany for their support and for providing the FloatSat hardware and RODOS operatingsystem.

REFERENCES

[1] I. Gavrilovich, S. Krut, M. Gouttefarde, F. O. Pierrot,´L. Dusseau, “Test Bench For Nanosatellite Attitude Determination And Control System Ground Tests,” *4S: Small Satellites Systems and Services Symposium, May 2014, Porto Petro, Spain*.

[2] Suhandinata and M. Setiawan, "Development of 1U CubeSat attitude determination and control system simulator", *THE 8TH INTERNATIONAL SEMINAR ON AEROSPACE SCIENCE AND TECHNOLOGY – ISAST 2020*, 2021.

[3] J. Berk, J. Straub and D. Whalen, “The open prototype for educational Nanosats: Fixing the other side of the small satellite cost equation,” 2013 IEEE Aerospace Conference, Big Sky, MT, pp. 1-16, 2013.

[4] Redah, M. Faisal and S. Montenegro, "The Floating Satellite System as an Educational Platform for Space Applications", *2020 IEEE 32nd Conference on Software Engineering Education and Training (CSEE&T)*, 2020.

[5] K. Ahmad, A. Redah, A. Arif, A. A. Khan, M. K. Saleem, and S. Montenegro. "An Educational Platform for Testing and Evaluating Satellite Control Algorithms in a Real-Time and Frictionless Environment.",” *19th International Bhurban Conference on Applied Sciences & Technology, August 2022.*

[6] A. Dhondea and M. Inggs, *"*Mission Planning Tool for Space Debris Detection and Tracking with the MeerKAT Radar,*" 2019 IEEE Radar Conference (RadarConf), pp. 1-6, 2019.*

[7] K. Ahmad, A. Redah, A. Arif, A. A. Khan, M. K. Saleem, and S. Montenegro. "An Educational Platform for Testing and Evaluating Satellite Control Algorithms in a Real-Time and Frictionless Environment.",” *19th International Bhurban Conference on Applied Sciences & Technology, August 2022.*

[8] Dr. S. Montenegro, F. Dannemann ”RODOS Real Time Kernel Design for Dependability” *DLR, Institute of Space Systems, Robert-Hooke-Str. 7,28359 Bremen, Germany*.

[9] Fezari, Mohamed, and A. A. Dahoud. "Integrated development environment “IDE” for Arduino." *WSN applications* (2018): 1-12.

[10] Hameed, Sarmad, N. Jafri, D. Rashid, and F. Shoaib. "Arduino based radar system." *3c Tecnología: glosas de innovación aplicadas a la pyme* 7, no. 1 (2018): 157-166.