Diyala Journal of Engineering Sciences

ISSN 1999-8716
Printed in Iraq

Second Engineering Scientific Conference
College of Engineering –University of Diyala
16-17 December. 2015, pp. 574-588

DEVELOPMENT OF A SERIAL COMMUNICATION PROTOCOL FOR SATELLITE ATTITUDE DETERMINATION AND CONTROL SYSTEM SIMULATOR

[1] Mousa Kadhim Wali, [2] Md. Hussein Baqir [3] Md. Hussein Baqir
[1, 2, 3] Foundation of Technical Education, Electrical and Electronic Technical College, Iraq
E-mail: [1]musawali@yahoo.com, [2] musawali@yahoo.com, [3]msnengb99@yahoo.com

ABSTRACT: - The purpose of this research is to design a valid serial communication protocol for satellite and developing Attitude Determination Control System (ADCS) simulator and on Hardware in the Loop (HIL) simulator by using digital signal processing dsPIC30F4013 board. ADCS subsystem consists of actuator, controller, sensor subsystem, and On Board Computer (OBC). These subsystem need to communicate between each other to control and determine the satellite attitude. The ADCS simulator is needed Because of the difficulties of satellite ADCS study in the space.

The development of this simulator is divided into microcontroller (MCU) integration in linear bus, HIL simulator, and analog to digital convertor (ADC). The HIL simulation is done through MATLAB to generate raw data which represents satellite attitude. A PID controller has been used as the control system for the satellite actuator.

ADC feature of the MCU is used to convert potentiometer analog value to digital which represents either satellite temperature or pressure value. Four MCU has been integrated together using RS485 bus with the implementation of token bus access. Cyclic Redundancy Check and checksum had been tested in the transmission.

The entire simulator design program has been done in C language and successful test results show that the research objectives has been achieved with 0.14 as an angle error in degrees with settling time 20s based on simple and low cost hardware with an overall 54.9s for data transfer, therefore these features considered as a contribution regarding too simple, fast and accurate satellite PID controller.

Keywords: SAT, ADCS simulator, RS-485, Hardware in the Loop, Communication protocol

1. INTRODUCTION

Sputnik was the first artificial satellite, launched in 1958 by the Soviet Union. Since that time, the space race began between United State (US) and the Union of Soviet Socialist Republics (USSR) [1]. A primary requirement of any satellite is to exchange information to it’s on board computer, and this information might be send to ground station. Basically, a satellite consists of five major subsystems; on board computer (OBC), thermal control subsystem, power Subsystem, attitude determination and control subsystem (ADCS), and payload [2]. Each subsystem can be represented as an individual microcontroller distributed system. All the subsystems communicate via serial communication on various buses.

The satellite is exposed to rays and noises from various sources, thereby the serial communication should have capabilities to handle the noisy environment, because of the noise can corrupt the data travels in a serial bus. The constructed data must reach the destination without any error and the system should be able to identify the corrupted frame using check sequence algorithm. As an example in RazakSAT, which shown in fig.(1), cyclic redundancy check (CRC) is being used [3]. On any satellite, several data networking is being used within subsystems or between subsystems which depends on the required characteristics of the interaction between those entities. The actual networks that are being used can be categorized as multi drop bus, point to point serial interface, local area networks (LANs), and point to point sensor and actuator interfaces [4]. Multi drop bus provides connection to components that are connected on a same set of bus like a controller area network (CAN bus), RS-485, RS422, etc. Point to point serial interface at satellite used for bulk data transfer and usually used for debugging and testing. LAN is used for larger infrastructure usage were each host has equally computing power and it is based on peer to peer type communication. Point to point sensor and actuator interfaces in satellite used for gathering data from sensors and to control actuators.

Several threats appeared during designing communication network for satellite system like; interference, ground, or space threat [5]. Space based threats happen at space environment due to solar eruption, and temperature variation and also space objects such as debris. Interference based threats which due to solar disturbance can cause damage to signals at satellite data links and subsystem buses. The interference can cause data loss, packet error and this faulty data make the ADCS wrongly interpret data and become unstable at space. To overcome those failures, the communication bus and protocol on the ADCS must have capabilities to handle all these expected incidents. In order to avoid misinterpretation of error frame, error check mechanism is used to validate each frame. The common error check mechanism used in satellite is checksum and CRC [6].

Serial communication protocol is essential for satellite in order to exchange data among its subsystems. Currently varieties of communication protocols are available for satellite application such as TDMA (time division multiple access), space wire and etc. However, most of the protocols are very expensive and complex to implement at university level. Therefore, this research is presented in order to implement a suitable, easy and reliable serial communication protocol between various subsystems with a common linear bus able to withstand high noise ratio and compatible to space environment. In addition to Simulation of Hardware in the loop (HIL) through MATLAB.

Research and development in serial communication protocol in ADCS hardware in the loop simulation for spacecraft is being done since a few decades back. HIL is a simulation technique used to test prototype or final hardware for simple or complex systems with real time constraints [9]. The Fire wire (type 1394) protocol has been determined as the most suitable protocol for satellite if compared with RS422, Space wire, and TAXI.

However, the complexity due to external hub and networking for TAXI, routing switches and server for Space wire application made the overall cost of implementation is higher compare with RS 422 [10].The frame structure which has data length and frame module on its header was constructed to collect housekeeping data, sensor data, and to run the camera capturing device based on CRC for error checking mechanism [11]. RS-485 Serial Port has the advantage of far distance communication and anti-jamming. Although, RS-485 transceiver has one transmit and receive end on both side with the same cable, it can't realize the function of full-duplex communication [12].

The rest of this paper is organized as follows. In Section II, we summarize the research methodology by elucidating the ADCS simulator development procedure and architecture, HIL simulation development, and communication software development. Section III illustrates the overview of the results and discussion of this present work, and finally conclusions are given in Section IV.

2. METHODOLOGY

2.1 ADCS Simulator Development Procedure

The first main part of the ADCS simulator is the four microcontroller hardware connected through serial communication protocol. The second part, explained in the next section, is the hardware in the loop simulation in MATLAB that sends data to the MCU controller of station2 (the controller which connected to the MATLAB through the RS232). The procedure shown in fig.(2) explain how to design the serial communication protocol and hardware in the loop simulation of the ADCS simulator. This flowchart indicate that the communication bus of the satellite systems should be studied carefully in addition to the microcontroller architecture, and the exact hardware interface of the hyper terminal in order to show satellite attitude and temperature. The communication serial bus RS485 is responsible of determination of transmit and receive interrupts. This system can be used only with one station or more (hardware in loop (HIL)) depending on the satellite requirements and connections. In this flowchart the software algorithms required are mentioned like; framing, error correction code and debugging for test.

2.2 Architecture of ADCS Simulator

This simulator contains controller system, which acquires data from HIL MATLAB simulation, sensor system processing data, on board computer that handle and verify data and monitoring terminal to monitor the serial bus activity through hyper terminal configuration. In real satellite, ADCS system receives pressure and thermal data from their equivalent sensors. However in this research, the real satellite sensors data are simulated using potentiometer that vary an analog signal read by the ADC of station 3 as shown in the whole block diagram in fig.(3) with another three stations requires RS-485 protocol. RS-485 is used so that the stations can share a common bus to send and receive data.

On this configuration, transmit and receive ports of the MCU is connected to the RS-485 transceiver. All the RS-485 transceiver shared ground connection to send data on its line. The data that is being sent from one station can be received by all stations but it must be programmed correctly so that only the addressed microcontroller would receive the data while the other discarded it. The RS-485 chip has driver enable ports that used when the specified MCU started to transmit data. Monitoring terminal station send type and number of interrupt to be shown by the hyper terminal with help of on board computer station. The hardware connection is shown in fig.(4).The main part of the simulator is the digital signal processor in DSPIC30F board shown in fig.(5) which designed to accommodate MCU in order to execute the commands .

This board supports 40 I/O ports, 48KB program memory, up to 5 Timers, two UART's, and 13 channel of ADC [8]. This research required data transmission to several stations using a common bus were multi point communication has been considered as shown in fig.(6). In this type of communication, there is one sender and multiple recipients. The disadvantage of this type of communication is only one can transmit at a time.

Four stations configuration requires RS-485 protocol. RS-485 is used so that the stations can share a common bus to send and receive data. On this configuration, transmit and receive ports of the MCU is connected to the RS-485 transceiver. All the RS-485 transceiver shared ground connection to send data on its line. The data, that is being sent from one station, can be received by all stations, but it must be programmed correctly, so that only the addressed microcontroller would receive the data, while the other discarded it. The RS-485 chip has driver enable ports that used when the specified MCU started to transmit data. Fig. (7) Shows the RS-485 transceiver chip. Fig. (7) Shows the RS-485 transceiver chip.

The development of packet based on bits of information prepared to be sent is called as framing [7]. Whereby, the data packet size determined by the designer. In general a frame consists of header, message and trailer. The frame travels through a bus is expose to radiation and high noise environment. Thus error detection mechanism is needed, like parity bit, cyclic redundancy check (CRC) and check sum.

2.3 HIL Simulation Development

This subsection describes the HIL simulation to generate a dummy data to be send to the stations. This dummy data is generated through MATLAB. HIL is a technique for performing system level testing of embedded systems in a comprehensive, repeatable manner, and cost-effective. ADCS requires real time modeling of a spacecraft, therefore this simulation is used to replace the expensive real environment of spacecraft. The HIL MATLAB simulation shown in fig.(8) is consist of constant input, slider gain, PID, actuator, satellite dynamic, sensor gain, data sampling, digital display and serial configuration for data communication. There is a need of data sampling as this system is running in continuous time signal.

The slider block is used to vary the constant input to the system. So, to generate different input, the slider can be used during running the simulation. The actuator transfer function is based on a very simple model and the satellite dynamics transfer function is based on the rigid body dynamic. Gain block is used to replicate sensor feedback value.

Rate of data sampling is chosen to be 0.1sec in order to read values of sensors within this period that expected for these values to be changed. The sampling value is converted to two decimal point values before sending to output display, MATLAB workspace via simout block, this function to interface MATLAB Simulink with workspace area to save the resulted data. Finally this sampled data must be send to the computer through the serial port via COM3 block as a communication serial bus functioning in request to send order with baud rate 1200 bit / sec.

The mathematical model of the compensator gain formula (G) of this system is represented by eq. (1.0)

$$G\text{=P }[1+\text{ I}\,\frac{1}{s}+\text{ D }\{\frac{N}{1+N\,\frac{1}{s}}\}]\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1.0)$$

Where, Proportional (p) = 0.092, Integral (I) = 0.008, Derivative (D) = 11.55, Filter coefficient (N) = 2.12

2.3.1 PID Controller

PID stands for Proportional, Integral, and derivatives. For satellite HIL subsystem, PID is the common controller which has the ability to control steady state error (SSE) and response time. In this research, the controller used in the HIL simulation is shown in fig.(9). The PID block in MATLAB has few parameters to be set such as PID type, form of controller, time domain. To get valid PID controller in parallel form, K_p, K_i, K_d and T_f in eq.(2) must be real and finite.

$$K_p + K_i * \frac{1}{s} + K_d * \frac{1}{T_f+\frac{1}{s}}\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.0)$$

Where $\frac{1}{s}$ is the Laplace transform of the derivative process.

K_p: Proportional gain, K_d : derivatives gain, K_i : Integral gain, T_f : filter time constant.

In addition to that, T_f must be a non-negative value. Tuning the PID controller is done through the built in function. The default value for PID parameters are $K_p = 1$, $K_i = 0$, $K_d = 0$ and $T_f = 0$.

The input signal to the PID controller in fig. (9) must be affected in parallel by three gains (proportional, derivative and integral). To get the output signal the proportional gain of the signal should be summed with both; the integrator of the integral gain and the filtered derivative gain.

2.4 Software Development

This subsection discusses the development of software routine for the designed ADCS simulator. The flow chart of important software routines have been constructed and explained in detail. There are mainly five routines; main routine, transmit routine, receive routine, data processing routine and ADC routine.

In embedded serial communication, transmission of data should be handle very carefully so that the data packet can be send without missing any bytes. In this research, the transmission will start whenever data is written to the transmit buffer of the MCU posses the token. For power consumption aware, transmission is based on software interrupt routine. Before sending the data to the transmit shift writing register TXREG, buffer is used to save these data. Fig.(10) explains how transmission and receive interrupt routines works. Figure (10.a) explains how transmission routine works when transmit buffer have data. In embedded serial communication, transmission of data should be handle very carefully so that the data packet can be send without any left over bytes. For this research, the transmission will start whenever data is written to the transmit buffer but of course the MCU should have the token. Interrupt based transmission is establish in software routine as interrupt is much more better in term of power consumption, and timing. Buffer is used to write data into it before send to transmit shift register and TXREG.

Figure (10.b) explains the receiving routine in detail. Receiving routine also based on interrupt method as the advantage over pooling method has been mentioned in above section. In this research, receiving routine is a bit complex as the receiver must detect the SOF byte in order to receive data until it reaches EOT byte. In addition, receiving routine in software can detect the address on each frame, and if the address does not match the MCU address, it will discard the message and will not accept any other byte until it receives the next SOF.

Figure (10.c) explains the receive data processing that is applied in the software. Processing data on data reception is very essential, so that the MCU can respond to the received message. Data processing involves copying data from receive buffer to processing buffer, validating minimum length, and validating frame based on checksum value. Apart from that, data processing is useful to identify token or data based on frame ID.

2.5 Communication Protocol

Communication development is necessary to transfer data between subsystems in a simulator. This subsection briefly explains the framing design, token bus implementation and error checking mechanism applied in RS-485 communication bus. A complete frame for this project contains SOF, destination address, source address, message followed by frame ID to identify whether the received frame is either token or data and trailer. Trailer contains error check bytes and end of transmission (EOT) byte. Fig.(11) shows the structure of the frame in general.The complete sample frame of data and token that is transmitted over serial bus is explained as in fig. (12) and fig. (13) respectively.

Token bus protocol is used as the media access in this simulator. In token bus protocol the network is connected through a linear bus. The operation of token bus is quite similar to a token ring. In this research, token generated at station 1. It sends token to station 2 by addressing it. If data is ready at station 2, the data is send to station 1 and the token pass to station 3 by addressing the token to it. If station 3 is ready, it sends data to station 1 and the token back to station 1.

Error detection mechanism is applied during constructing the frame. Checksum is a very simple error checking mechanism. During checksum calculation, all the byte at the frame is added and the checksum value is appended to the frame. For check sum calculation, only LSB byte is appended to the frame. CRC is used after the frame check sequence checksum.

2. RESULT AND DISCUSSION

The control maintenance mode used a PID controller as detailed earlier in the HIL components. The result in table (1) shows the satellite pointing error simulations of degrees accuracy during 10 second. There is no specific method to tune PID controller. Its output value solely depends on the gain values that have been set earlier. However, tuning very essential to make the response of the system to be less overshot with shorter rising time as shown in fig. (14). In order to apply the PID tuning for this simulator, MATLAB built in functions has been used. HIL simulation was put on running mode and the output has been displayed on the output display. The output value then is sent to station 2 (MCU) via serial cable RS232. Station 2 is used to obtain magnetometer data to locate the position and orientation of satellite. The stimulation data has been used to replace the real magnetometer data. These stimulus data is generated using HIL simulation through MATLAB.

As shown by the schematic in fig.(15), all the station can emits token or data. Upon receiving the token, the data will be send to station 1 followed by the token to next station. Token initialization is start from station 1 which used to receive data from other stations. This station acting as the on board computer in real satellite application and has the ability to detect and check data error because this error checking mechanism is important to validate the data. Fig.(16a) illustrates the timing diagram for token transmission at station 1 with total time taken for the whole frame about 3.96ms. Fig.(16b) shows the data timing diagram at station 2 with total time 9.20ms for transmitting the data. Oscilloscope reading shows that the total time for the data to transmit via station 3 is about 9.442ms as shown in fig.(17a).

Fig. (17b) shows the RS-485 driver enable period when station 1 transmits the token were the driver is set high for about 8.116ms. The best value for RS-485 line baud rate is 9600bps. Fig. 18 show the overall RS-485 directions and stations activity. Direction 1, Direction 2 and Direction 3 on the diagram are represents the RS-485 driver direction for Station 1, Station 2 and Station 3 respectively. There is one character delay before the direction of the driver goes low to let enough time for the byte to be transmitted over the bus.

Table 2 shows the time required for data transfer in the ADCS. Block Diagram shown in fig.(19) is the debugging system configuration. PIC kit 3 is used as the debugger to monitor the software flow in the MCU. The error in the software flow can be identified by monitoring the debugger running in the MPLABX software. Apart from that, the hyper terminal would help to show lastly received data which can be used to identify which MCU is in problem. In this figure, four PC is suggested for better debugging environment on the ADCS simulator.

Hyper terminal has been used as the monitoring terminal to display bus activity within microcontrollers. In windows based operating system, hyper terminal built in software. The best value of baud rate for the monitoring terminal is 19200bps. If the value is adjusted to higher rate of data transfer, the hyper terminal would show rubbish characters. Fig.(20) shows data transfer between microcontrollers.

Table(3) shows the comparison between the settling time, error angle in degrees, transceiver total time and complexity of the previous researchers work and the present work. From table 3, the minimum settling time is 18s is achieved on getting 0.16 degrees as an error angle [*Zhang, 2012*] but with high complexity. By low complexity, 0.2 degrees is achieved with settling time 25s (Honglei, *2013*), while Jorgensen, (2011) got 0.5 degrees with 50s settling time. However the present system achieves the error angle of 0.14 degrees with settling time 20s based on low complexity.

3. CONCLUSION

The simulation of the ADCS is done based on 4 MCU representing 4 stations; OBC, controller, sensor, mentoring terminal. In addition to HIL simulation in MATLAB to provide satellite attitude and position. These stations are connected through RS-48 protocol with very low cost, efficient and reliable serial communication established at the university level. In this research, 0.14 error angles in degrees is achieved during 10s with settling time 0.14s and an overall 54.9s for data transfer. The designed and developed protocol can be used as the guidance in the future to develop a better communication in the satellites. RS-485 has been chosen as the serial bus because it supports multipoint communication and let the designer choose the frame size and data contents unlike others which only supports 8 bytes of data per frame. In addition to that, RS-485 has the capability to be used at space to withstand high ratio of radiation and noises as it uses differential line to transfer data. Grounding is important when using RS-485 protocol, therefore all the chips must be grounded together at same line because of using balanced and differential signals. In future, this research can be upgraded by applying multiple bus line as the fault tolerance setup. Bus arbitrary system to monitor bus activity would help to reduce power consumption by making the idle line go to sleep mode. Future works include reducing the steady-state angular velocity errors, overall data transfer time, testing the sensors with the actuators and implementing the designed systems in a satellite prototype.

REFERENCES

1) Launius, R. D. (n.d.). *The History of Satellites*. Retrieved March 23, 2012, from Inventors: http://inventors.about.com/library/inventors/blsatellite.htm.
2) Arif, D. T. (2010). *Design of Low-cost Telecommunications CubeSat-class Spacecraft.* Crotia: Aerospace Technologies Advancements.
3) Arshad, S. A., & Syed Omar, A, S. (2004). *RazakSAT Concepy, Design and Development.* Selangor,Malaysia: Astronautic Technology (M) Sdn. Bhd.
4) Secretariat, C. (2007). *CCSDS Report Concerning Spacecraft Onboard Interface Services.* Washington, USA: National Aeronautics and Space Administration.
5) Akir, Z. I. (2006). *Space Security:Possible Issues & Potential Solutions.* Athens, Ohio: College of Communication,Ohio University
6) Doering, T. J. (2009). *Development Of A Reusable Cubesat Satellite Bus Architecture For The Kysat-1.* Lexington, Kentucky: University of Kentucky.
7) Samson Ag Inc., (1999), *Serial Data Transmission Technical Information,* Frankfurt, Germany.
8) Microchip Technology Inc.,(2010), *dsPIC30F4013/3014 Application Notes*, USA.
9) Renato Miyagusuku, M. A. (2009). *Low cost validation test-bed for small satellite's attitude determination and control.* Retrieved March 20, 2012, from Universidad Nacional de Ingenieria, Lima, Peru: http://www.nanosat.jp/.../[NSS-03-0402]_Low_cost_validation_test_bed.
10) Wolfram, K. D., & Bloom, H. J. (2004). New Radiation-Hardened High-Speed Serial Data Bus For Satellite Onboard Communication. *IEEE* , 156-159.
11) Clausen, T. B. (2006). Designing On Board Computer and Payload for AAU CubeSAT. *IEEE* , 1-9.
12) Su Xunwen, W. S. (2010). RS-485 Serial Port Pseudo-full-duplex Communication Research and Application. *Prognost ics & System Health Management Conference* (p. MU3097). Macau: IEEE.
13) Kristiansen, R. & Nicklasson, P. J. (2005). 'Satellite Attitude Control by Quaternion-Based Backstepping'. *American Control Conference June 8-10, 2005*. Portland, OR, USA.
14) Rgensen, U. J., & Gravdahl J.T. (2011),' Observer Based Sliding Mode Attitude Control: Theoretical and Experimental Results'. *Modeling, Identication and Control*, Vol. 32, No. 3, 113:121.
15) Zhang, F., Duan, G., & Hou, M. (2012),' Integrated relative postion and attitude control of spacecraft in proximity operation missions with control saturation'. *International Journal of Innovitave computing, information & technolgy*, Vol. 8, No. 5B, 3537-3551.
16) Honglei, A., Jie, L., Jian, W., Jianwen, W., & Hongxu, M., (2013),' Backstepping-Based Inverse Optimal Attitude Control of Quadrotor', *International Journal of Advanced Robotic Systems*, Vol. 10, 223:2013.

Table (1): Satellite angle error verses gain

Sensor Gain	10	20	30	50	75	100
Error Angle (degrees)	0.14	0.38	0.57	0.95	1.43	1.90

Table (2): The Token and Data Transfer in Milliseconds.

Time(ms)	Master /Station 1	Station 2	Station 3
0	-	-	-
8.5	-	token	-
18.3	data	-	-
26.8	-	-	token
36.6	data	-	-
45.1	-	token	-
54.9	data	-	-

Table (3): comparison of settling and transceiver total time, error angle and complexity of present work with that of earlier works

Name of Author, year [Ref. No.]	*Settling time (second)*	Error Angle (degrees)	Transceiver total time (s)	Complexity& cost of hardware
Kristiansen, *2005 [13]*	*28*	*0.37*	*86*	*High*
Jorgensen,*2011[14]*	*50*	*0.5*	*71*	-------
Zhang, 2012[15]	*18*	*0.16*	------	*High*
Honglei, *2013[16]*	*25*	*0.2*	*54*	*Low*
Present work	*20*	*0.14*	*54.9*	*Low*

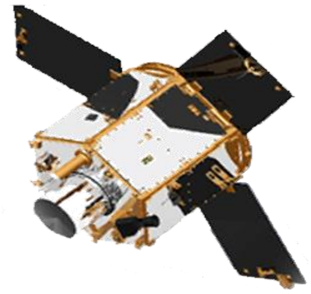


Figure (1): RazakSAT with Solar Panel Deployment

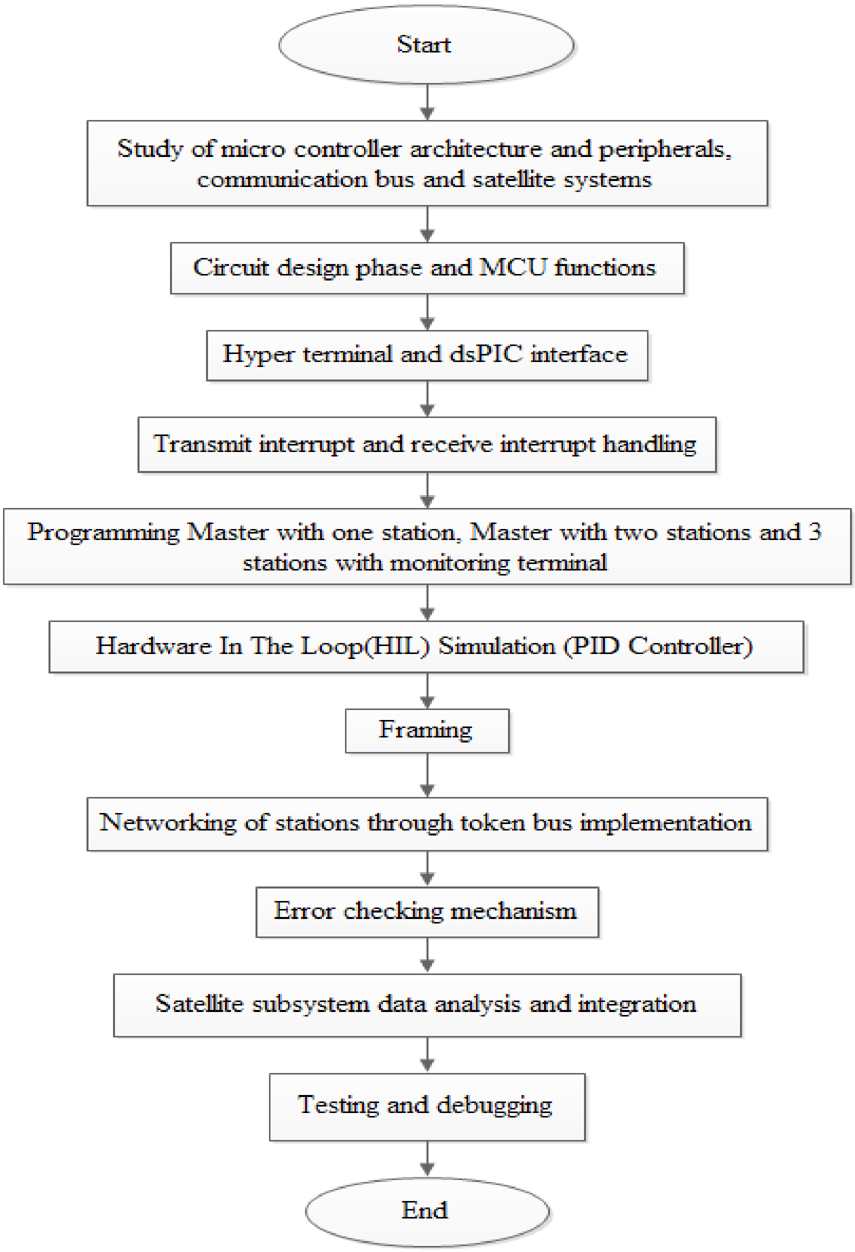

Start
Study of micro controller architecture and peripherals, communication bus and satellite systems
Circuit design phase and MCU functions
Hyper terminal and dsPIC interface
Transmit interrupt and receive interrupt handling
Programming Master with one station, Master with two stations and 3 stations with monitoring terminal
Hardware In The Loop(HIL) Simulation (PID Controller)
Framing
Networking of stations through token bus implementation
Error checking mechanism
Satellite subsystem data analysis and integration
Testing and debugging
End

Figure (2): Procedure of Developing ADCS Simulator

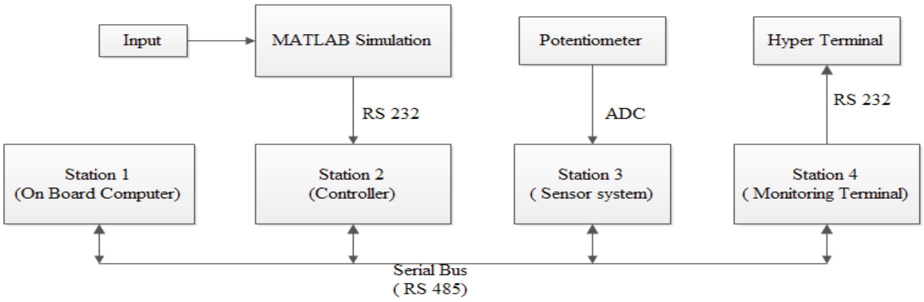


Figure (3): Architecture of ADCS Simulator with HIL

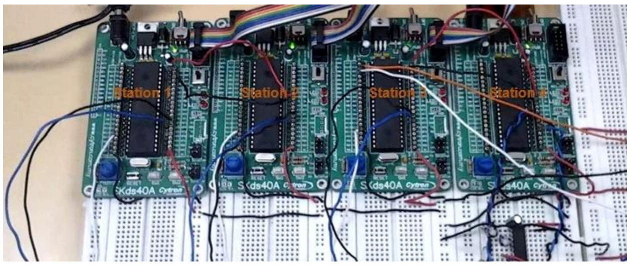


Figure (4): Practical circuit of Microcontroller Connection of the ADCS

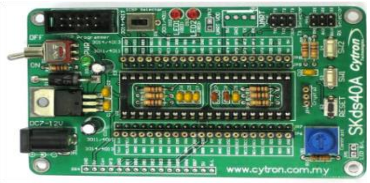


Figure (5): dsPIC30F Board

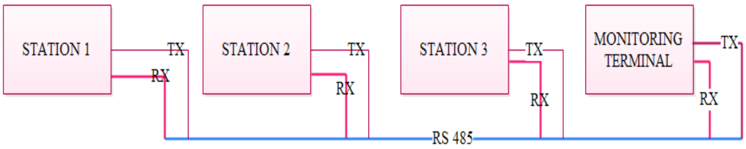


Figure (6): Four Station Configurations

Figure (7): SN75176 Chip (RS-485 Transceiver)

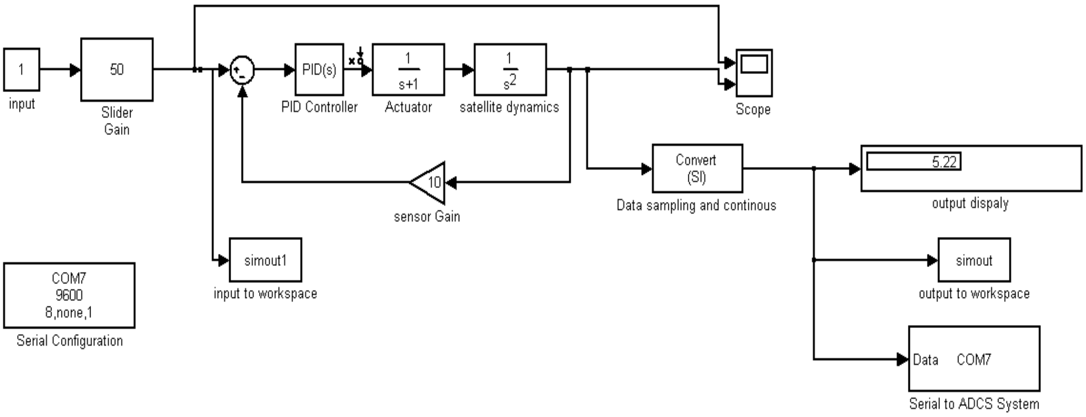


Figure (8): Simulink Diagram of HIL Control System

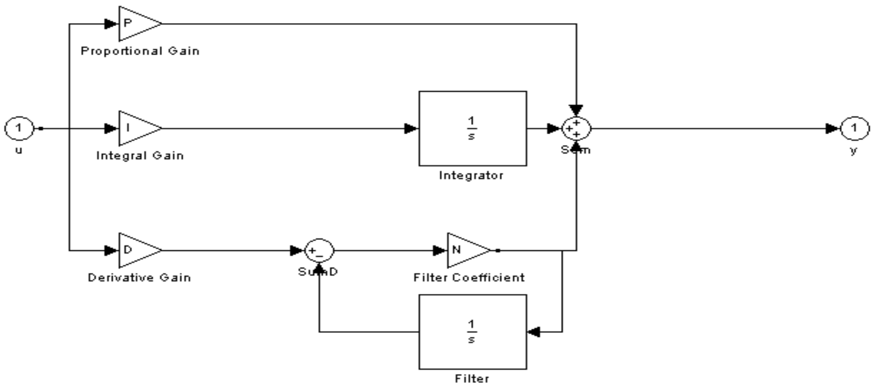


Figure (9): PID Controller Block diagram of HIL in MATLAB

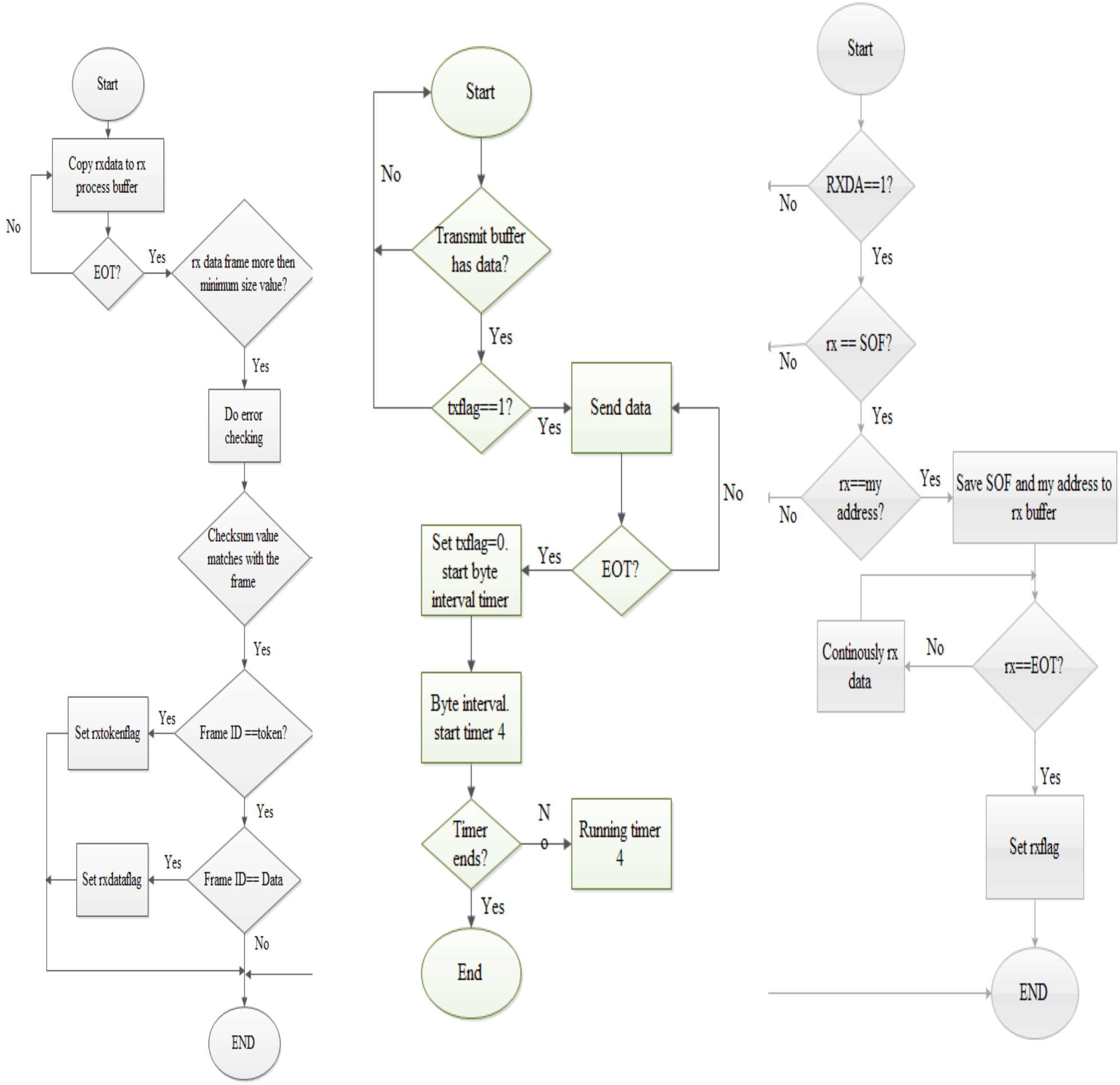


Figure (10): Flow Chart of (a) Transmit Interrupt Routine (b) Receive Interrupt Routine (c) Receive Data Processing Routine

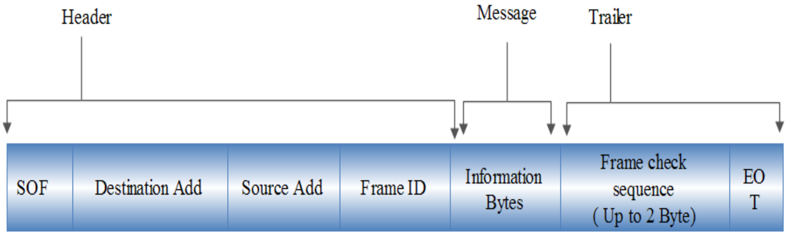


Figure (11): General Structure of Frame

~	3	1	**0x94**	T	2	0	.	3	2	P	1	0	2	.	6	7	*

Figure (12): Sample of Data Frame with SOF

~	3	1	0x94	7	*

Figure (13): Sample of Token Frame with SOF

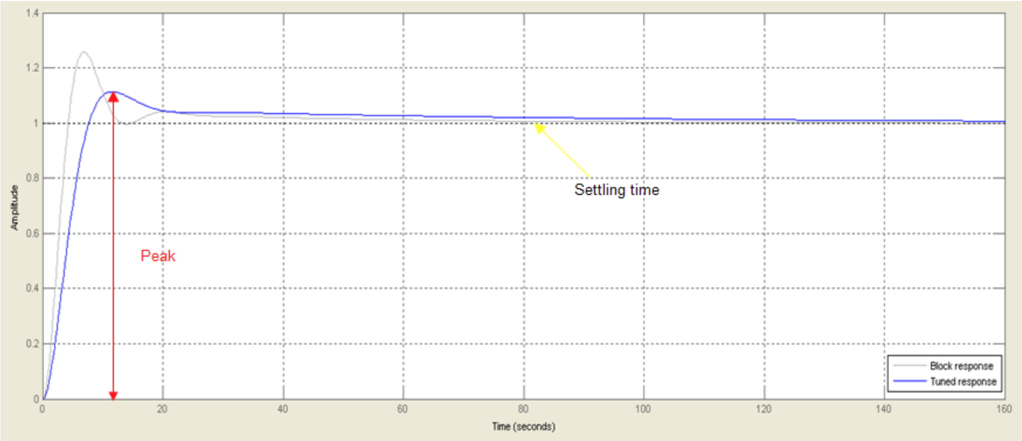


Figure (14): PID settling time graph

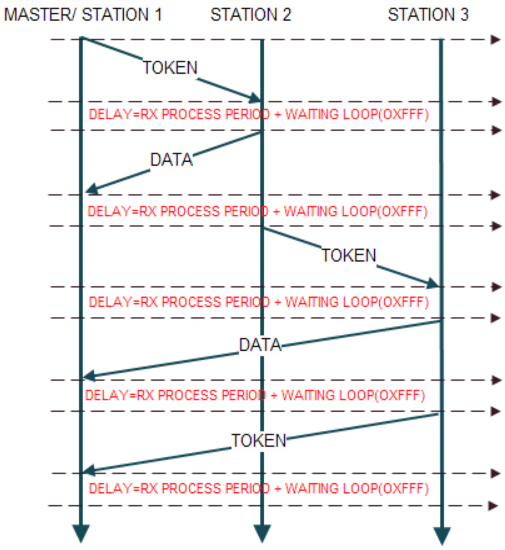
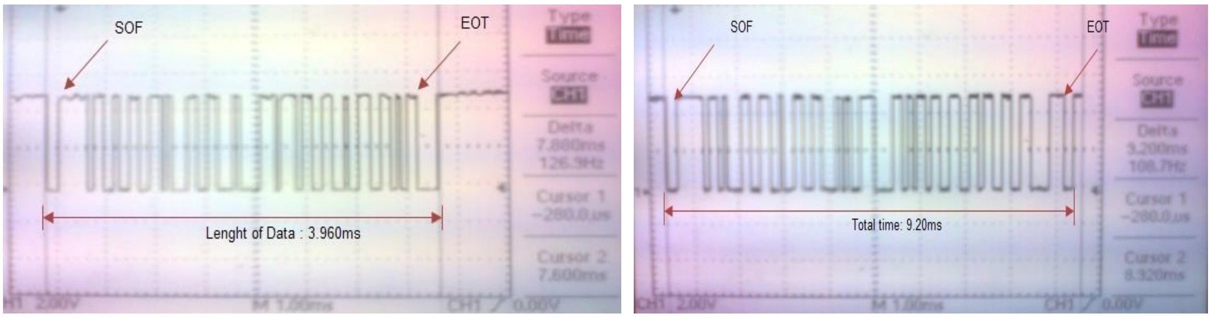

MASTER/ STATION 1
STATION 2
STATION 3
TOKEN
DELAY=RX PROCESS PERIOD + WAITING LOOP(OXFFF)
DATA
DELAY=RX PROCESS PERIOD + WAITING LOOP(OXFFF)
TOKEN
DELAY=RX PROCESS PERIOD + WAITING LOOP(OXFFF)
DATA
DELAY=RX PROCESS PERIOD + WAITING LOOP(OXFFF)
TOKEN
DELAY=RX PROCESS PERIOD + WAITING LOOP(OXFFF)

Figure (15): Data and token Flow in RS-485 Bus



(a) (b)

Figure (16): Timing Diagram of (a) Token at Station 1 (b) Data at station 2

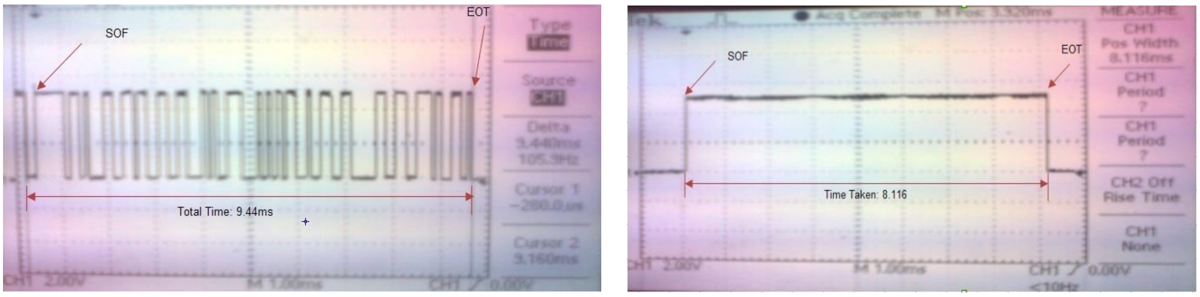


Figure (17): Timing Diagram of (a) Data at Station 3 (b) RS-485 Direction Driver

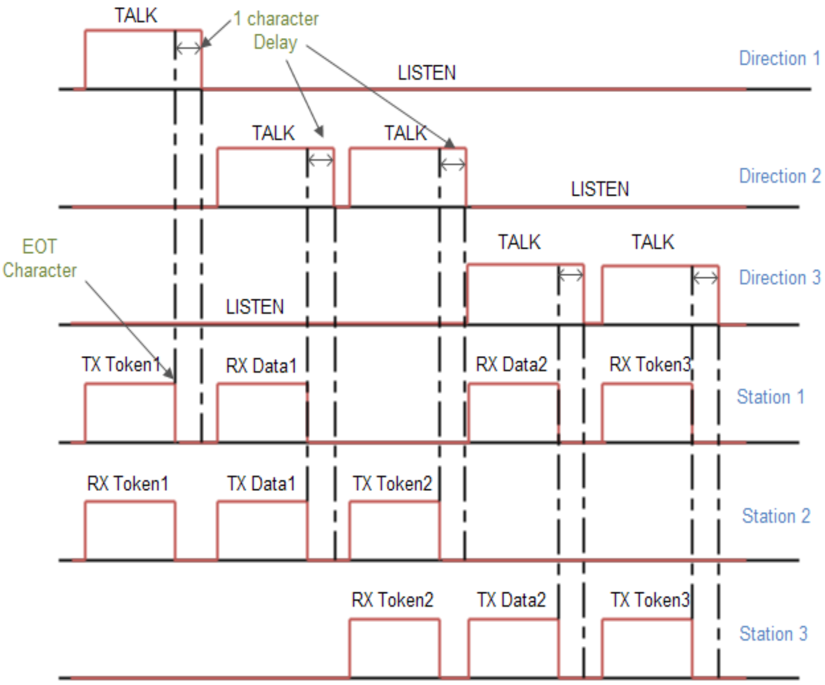


Figure (18): Timing Diagram of the RS-485 Driver Directions over the stations

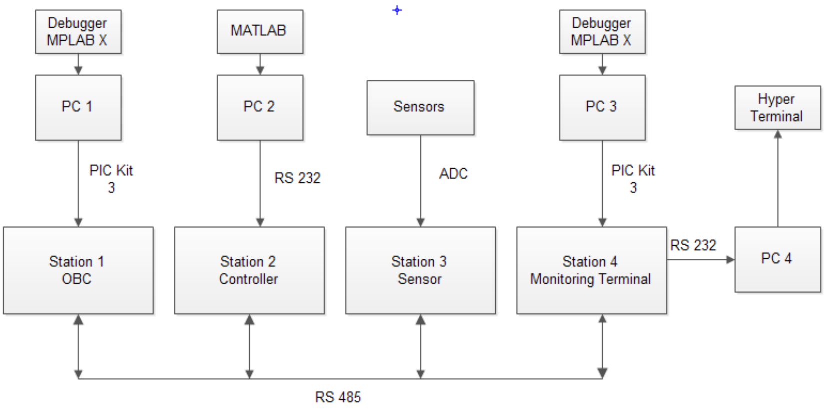
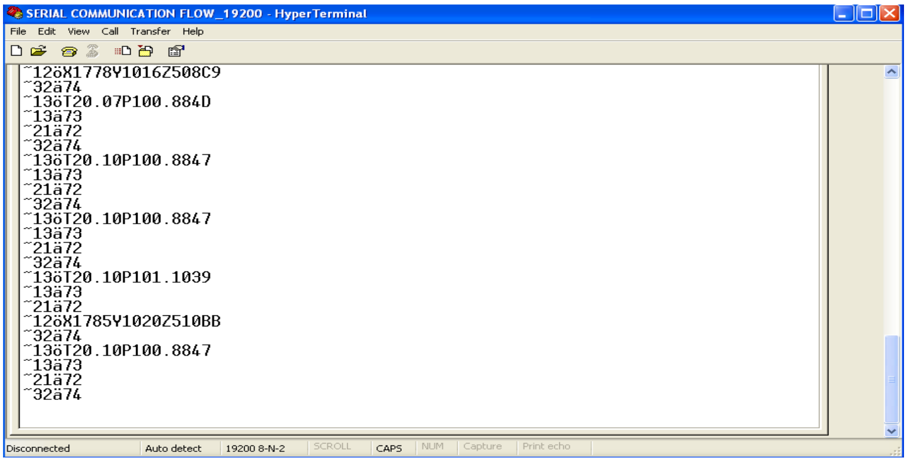


Figure (19): Block Diagram of Debugging System Configuration


SERIAL COMMUNICATION FLOW_19200 - HyperTerminal
File Edit View Call Transfer Help

```
~12öX1778Y1016Z508C9
~32ä74
~13öT20.07P100.884D
~13ä73
~21ä72
~32ä74
~13öT20.10P100.8847
~13ä73
~21ä72
~32ä74
~13öT20.10P100.8847
~13ä73
~21ä72
~32ä74
~13öT20.10P101.1039
~13ä73
~21ä72
~12öX1785Y1020Z510BB
~32ä74
~13öT20.10P100.8847
~13ä73
~21ä72
~32ä74
```

Disconnected | Auto detect | 19200 8-N-2 | SCROLL | CAPS | NUM | Capture | Print echo

Figure (20): Hyper terminal display received data from master MCU

تطوير بروتكول المنفذ المتوالي لمحاكاة منظومة ألسيطره والتحديد لموقع ألستلايت

د.موسى كاظم والي ، محمد حسين باقر ، د.ماجد صلال نغمش
هيئة التعليم التقني- الجامعه التقنيه ألوسطى ، كلية التقنيات الهندسية الكهربائية والاكترونية

الخلاصة:

الغرض من هذا ألبحث لتصميم بروتكول أتصالات متوالي موثوق به للستلايت وذلك بتطوير محاكاة تحدبد ألموقع للستلايت بمنظومة ألسيطره (*ADCS*) ألمعتمده على محاكاة ألكيان ألمادي في ألحلقه(*HIL*) بأستعمال بورد معالج ألشاره الرقمي *dsPIC30F4013*. منظومة *ADCS* تتألف من ألمشغل ألميكانيكي, ألمسيطر, ألمتحسسسات, وبورد ألحاسوب. هذه ألأجزاء تحتاج ألى ألأتصال فيما بينها لغرض ألسبطره على موقع ألستلايت. نظرا لصعوية دراسة منظومة *ADCS* للستلايت في ألجو, لذلك دعت ألحاجه ألى محاكاتها. تطوير هذا ألمحاكاة يقسم الى مجموعة من مسيطر دقيق في ناقل خطي, محاكاة ألكيان ألمادي *HIL* ومحول تماثلي ألى رقمي. محاكاة *HIL* ينجز من خلال ماتلاب ليولد بيانات متواليه والتي تمثل موقع ألستلايت وألسيطره عليه. ألمسيطر *PID* أستخدم كنظام سيطره على ألمشغل ألميكانيكي للستلايت. مميزات ألمحول ألنماثلي ألى رقمي ألخاص بألمسيطر ألدقيق أستخدم لتحويل فرق ألجهد ألتماثلي وألذي يمثل قيمة ضغط وحرارة ألستلايت. أربعة من ألمسيطرات ألدقيقه أستخدمت سويه بأستخدام ناقل *RS485* بتنفيذ جلب رمز ألناقل. تم أستخدام ألتحقق ألدوري ألفائض وتحقيق ألمجموع في ألنقل. بقية برنامج ألمحاكاة صمم معتمدا على لغة *C* وأن ألنتائج بينت بأن أهداف ألبحث قد تم ألحصول عليها بزاوية خطأ *0.14* درجه وبزمن أستقرار *20s* معتمدا على تصميم يسيط ورخيص ألكلفه وبوقت كلي *54.9s* لنقل ألبيانات.