Indian Journal of Science and Technology, *Vol 11(23), DOI: 10.17485/ijst/2018/v11i23/123174,* June *2018*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

COTS Based Implementation of Data Handling Unit for Micro-Satellites

Mohsin Ahmed[1], Bilal Sheikh[1] and Omer Mohsin Mubarak[2*]

[1]Department of Electronics Engineering, Iqra University, Pakistan
mohsinhundal@hotmail.com, bilal3051992@hotmail.co.uk
[2]Department of Electrical Engineering, Jouf University,
Kingdom of Saudi Arabia; ommubarak@ju.edu.sa

Abstract

Background: Data Handling Unit (DHU) is a one of the most important units of Command and Data Handling System (CDHS) in a satellite. The unit is responsible for Telemetry (TM) acquisition and Tele commands (TC) Execution and on-board communication management through shared and dedicated links. **Objectives:** This paper presents an effective COTS based implementation of DHU for micro satellites. **Methods:** Controller Area Network (CAN) is used in the proposed system as shared communication link. It is preferred over I^1C because of higher immunity to losses. Redundancy is very important in a satellite system, since on-site maintenance is not possible after its launch. In order to add redundancy to this system, direct links and dual redundant CAN is added to the system. **Findings:** Two different designs are presented for Data Handling Unit. In the first design FPGA has been used as central processor and its advantages and disadvantages analyzed. The second design based on PIC microcontroller is proposed. Paper also presents the software design for TM, TC, CAN, and Analog to Digital Converter (ADC) and UART. Results presented in the paper show successful testing of individual modules as well as the complete system. Simulation based testing was done for hardware having simulation models. In other cases, hardware based testing was performed by displaying result on onboard LCD and on PC using UART. **Application:** The proposed DHU can be used in weather monitoring satellites, aircrafts for communication of flight sensors data and automobiles for effective communication between sub-systems.

Keywords: Micro-Satellite, CAN, DHU

1. Introduction

A micro satellite is being developed in joint academia industry collaboration. This paper is focused on Data Handling Unit (DHU) of this satellite. DHU is one of the most important units in a microsatellite and can be considered to be a brain of a satellite[2]. One of its functions is to monitor the system through different sensors (e.g. current, voltage, temperature etc.). DHU is a sub unit of Command and Data Handling Subsystem (CDHS). The on-board CDHS of a satellite is the unit which carries and stores data between the various electronics units and the ground segment, via the Telemetry, Telecommand and Communication subsystem (TTCS). The health data is called telemetry (TM). Telemetry data is collected from all the subsystems and sent to the ground station through RF communication chain. Commands for different functions to be executed on-board a satellite are called telecommands (TC). Telecommands are generated from Ground stations and executed by the relevant subsystem through DHU. Other subsystems connected to DHU in a satellite are Attitude Control Subsystem (ACS), Payload, Electrical Power Subsystem (EPS) and Telemetry, Telecommand & Communication subsystem (TTCS). TTCS subsystem contains antennas for ground transmission. TTCS receives packetized TM data and then modulate it for onward ground transmission, whereas EPS contains a Power Conditioning Unit (PCU)

*Author for correspondence

which continuously monitors the state of Battery Voltage, Solar Power output and the loads. Power distribution unit (PDU) is also part of EPS. A PDU is a unit fitted with multiple outputs designed to distribute electric power to all active units of a satellite. ACS consists of all navigation equipment, which is used to maintain or change the direction at which the spacecraft is pointing.

Redundancy is very important in satellite systems, since on-site maintenance is not possible. Therefore, a dual redundant Controller Area Network (CAN) and RS422 is used to transfer data between DHU and other subsystems. DHU also has direct TM and TC connections with PDU, PCU in EPS 1 and Antennas of TTCS. After acquiring TM data, it packetizes it according to AX.25 and sends it to TTCS for modulation and onward ground transmission. Similarly, it receives ground TC from TTCS, decode it and execute the command directly or forward it to respective subsystem as required. It has synchronous interface (i.e. Data, Clock) with TTCS. DHU controller should have high clock speed because it has to complete this entire process in given time interval, which is 1 sec in our case. DHU block diagram is shown in Figure 1.

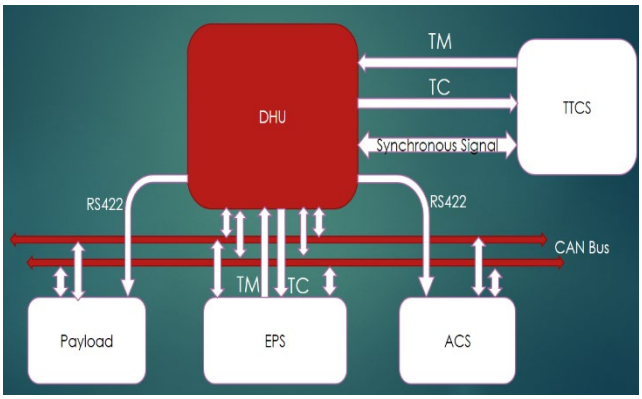

Figure 1. Block diagram of Data Handling Unit of a satellite.

1.1 Quick Sat

Computer CPU was used in QuickSat which has a clock speed of 1GHz and it was embedded onto the motherboard. Clock speed was well above the minimum requirement of 15MHz. Mvix Nubbin small-sized USB2.0 Wireless LAN adapter was used for communication[1]. In this satellite design, wireless communication was used, which is significantly slower than the wired network.

1.2 Delfi-C3

Delfi-C3 is a nanosatellite program of Delft University of technology, Netherlands. DelfC3 was launched from India in 2008. DelfC3 is in orbit and still operational. The objective of this project was to provide students an opportunity to obtain knowledge about satellite technology. PIC microcontroller was used as central processor and I2C for bus communication. DelfC3 was the first university-class satellite[3].

One of the problems with Delfi-C3 was data loss in CDHS. The design team proposed better node protection, watchdog functionality and structural software testing method to avoid data loss[3].

1.3 Hermes Cubesat

University of Colorado, USA used 16-bit PIC microcontroller PIC24HJ256GP610 for CDHS system in their Hermes Cubesat system. Explorer 16 development board was used for PIC development and driver implementation. Internal watchdog timer was used to help prevent run time errors and I[1]C for bus communication[4].

2. Motivation

This Paper presents a design for data handling unit in micro satellite with following parameters for improved performance:

- Wired network communication is used, since wireless network communication results in higher losses.
- Watchdog timer functionality is added to prevent run time errors.
- CAN is used, as it uses differential signal to transmit data and results in higher immunity to losses. Table 1 shows the comparison between CAN and I[1]C.
- In earlier proposed designs, PIC microcontroller was used in CDHS systems as central processor and CAN for bus communication. However, they haven't yet been used together in CDHS of a satellite. In this design both microcontroller and CAN are used together for improved performance.

3. Design Requirements

DHU is designed as per following requirements:

Table 1. Comparison between CAN and I2C

S. No.	CAN	I[1]C
1	CAN uses differential signal to communicate which makes it more immune to losses.	I[2]C does not use differential signal for message transmission.
2	A single CAN node can have different Ids i.e. it can receive messages with different identifiers from different CAN nodes.	This functionality is not present in I[1]C.
3	CAN have error detection capabilities.	It has less error detection capabilities as compare to CAN.
4	It requires higher power as compare to I2C	Power consumption is lower in microcontroller with build in I2C as compare to microcontroller with build in CAN
5	Baud rate of CAN can go only up to 1Mbit/s.	The baud rate of I2C can go up to 3-4Mbit/s.

- Extended temperature version should be available of the IC's used in designing.
- Redundancy should be provided at every level.
- High Clock speed of central processor.
- Dual Redundant CAN interface.
- Telemetry and Telecommands interfaces.
- Four RS422 interfaces.
- Selected Software and tools should be used.
- Power consumption should not be more than 2 Watts.
- DHU shall operate on +12V and +5V, which are received from PDU
- Support for Real Time Operating System.
- Sufficient memory for TM and TC storage (greater than 512KB).

4. Hardware Design and Algorithm Development

This paper first presents FPGA based DHU, which was initially designed for this project. However, later a microcontroller based system was designed. Both these designs are discussed and analyzed in this section.

4.1 FPGA based Design

A Spartan FPGA XC3S500E was selected as central processor. The major reason for the selection of this FPGA was better adaptability and reduced complexity of circuit board. It also reduces the implementation cycles. Verilog Language was used for programming this FPGA. Block diagram of FPGA based design is shown in Figure 2.

The design used Philips Semiconductors SJA1000 as a Standalone CAN controller and NXP semiconductors PCA82C250 as a CAN transceiver. As mentioned in design requirements, PDU provides DHU with +12V and +5V. However, Spartan S3E FPGA works at 3.3V and

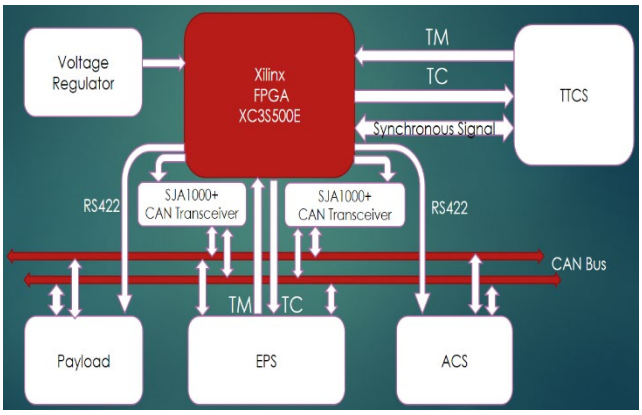

Figure 2. Block diagram of proposed FPGA based design.

there is no internal voltage regulator present in it. Thus, an external voltage regulator (LM317) was incorporated in the design.

Four software based Universal Asynchronous Receiver/Transmitter (UART's) were used in this design to fulfil the requirement of four RS422 interfaces.

4.1.1 Advantages

Advantages of FPGA XC3S500E based design were:

- FPGA has very high speed (up to 300MHz).
- Extended temperature version of the FOGA is also available.
- Low cost.

4.1.2 Disadvantages

This design also had few disadvantages, which are listed below:

- Clock speed of SJA1000[5] is limited to 24 MHz. This made advantage of high speed FPGA useless.

- No open source RTOS was available for XC3S500E.
- Design does not include circuitry for watchdog timer, TM and TC.
- No internal CAN controller and ADC was available, so external CAN controller and ADC are required resulting in more power consumption and additional hardware.
- Software based UARTs are used, which may result in data loss.
- Memory requirements are not fulfilled.

This design was modified to incorporate these flaws using a microcontroller instead of FPGA.

4.2 Microcontroller Based Design

Microcontroller was used as central processing unit in this design. Microcontroller model was selected after comparing number of features, i.e. build in CAN controller, availability of extended temperature version, speed and UART. A Microchip microcontroller PIC18f66k80 was selected after comparing various options. Figure 3 shows the block diagram of modified design.

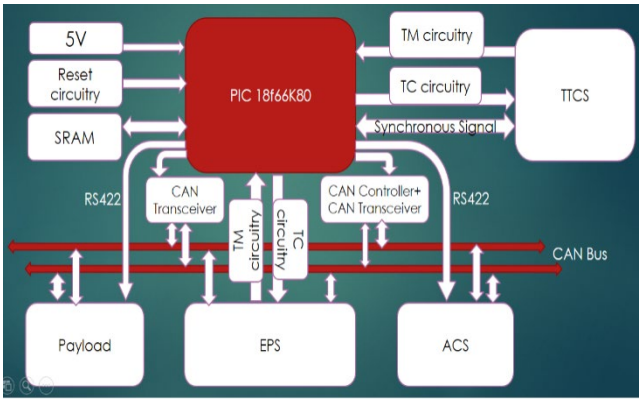

Figure 3. Block diagram of proposed microcontroller based design.

4.2.1 CAN Interfacing

Two CAN interfaces are required, i.e. nominal CAN and redundant CAN, as per the design requirements. One external CAN controller MCP2515[6] was used with PIC18f66k80 for the interfacing of Redundant CAN, whereas for nominal CAN a build in CAN controller of PIC18f66k80 was used. Both the CAN interfaces contain a separate CAN transceiver as shown in Figure 4. In this design MCP2551 is used as CAN transceiver[7].

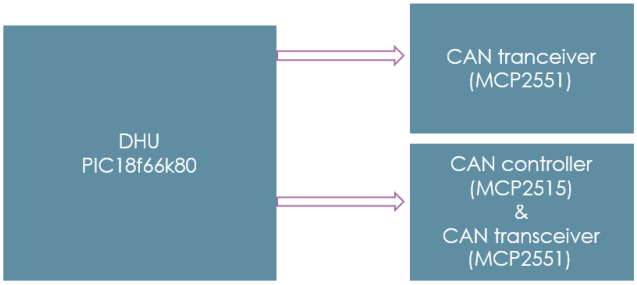

Figure 4. Block diagram of DHU Interfacing for dual redundant CAN communication.

4.2.2 Telemetry and Telecommand Circuitry

At least 40 direct TM channels are required for acquisition of direct telemetries. The telemetry interface specifications are as follows:

- Voltage levels for Bi-level Telemetries:
- Low (0): 0V and High (1): 5V
- For Analog Telemetry: 0 to 5V

Figure 5 shows the block diagram of TM interfacing. DHU is required to collect 40 direct TM's consisting of 24 Bi-level telemetries and 16 Analog telemetries. It was not possible to directly interface these with microcontroller, so multiple mixes have been used to get that data from other subsystems. Only one pin of DHU was used to collect digital TM and the ground terminal for all Digital telemetries remained common. However, two pins of DHU were used for analog TM's, since they do not have a common ground. Analog telemetries were digitized by build in ADC of PIC18f66k80[8].

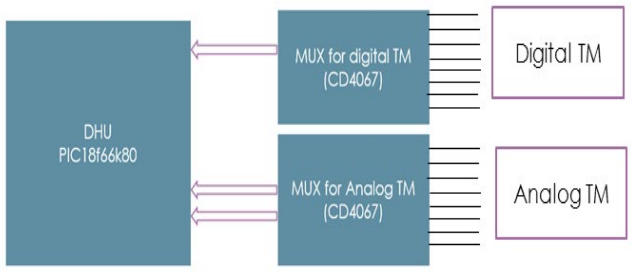

Figure 5. Block diagram of DHU interfacing for Telemetry data.

At least 70 direct Telecommand (TC) channels are also required for execution of critical TCs. The TC interface specifications are as follows:

- Voltage levels for Bi level Telecommands:
- Low (0): 0 to 0.2V and High (1): 11V ± 1V
- For Analog TCs: 0 to 11V

Proposed system has been provided with 72 TC's. They included 40 Bi-level telemetries (0 and 11V) and 32 Analog telemetries (0-11V). As these levels cannot be generated directly by DHU, external OP-amps and Demux were used. Figure 6 shows the block diagram of TC interfacing. In order to achieve the required voltage level for analog TC, first digital to analog converter (DAC) was used for converting digital signal from DHU to analog signal and then Op-amp was used as voltage amplifier followed by demuxes to generate analog TC. However, to generate digital TCs, Op-amp and Demux are directly used with DHU.

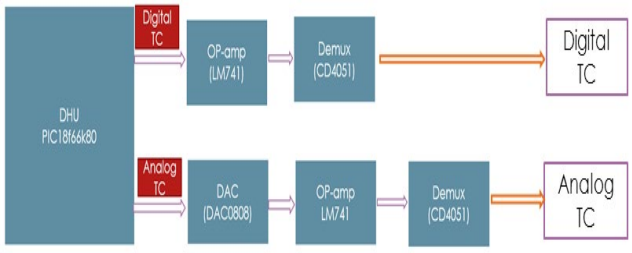

Figure 6. Block diagram of DHU interfacing for Telecommand data.

4.2.3 QUAD UART

Four RS422 interfaces are required, as per the design requirements. PIC18f66k80 has only two build in UART's. In the initial FPGA based design, four software based UART were used. Software based UART has a drawback that if it receives data from two subsystems simultaneously, data from one of the subsystems gets lost. Interrupt based Soft UART also does not solve this problem. Thus, a hardware based QUAD UART IC TL16C754B[9] is used in this design.

4.2.4 SRAM

Design requirements require 512 kB of memory storage. The microcontroller PIC18f66k80 is capable of supporting up to 512kB of SRAM. Hence an external SRAM was used in the design.

4.2.5 RTOS

A RTOS separates the program functions into self-contained tasks and implements an on-demand scheduling of their execution. Some of the open source RTOS available for PIC18 microcontroller are OSA[10], PICOS18[11] and FreeRTOS[12].

4.2.6 Advantages of Modified Design

It fulfills nearly all the design requirements of Data Handling unit. Following are the few advantages of the using of PIC18f66k80 as DHU central processor.

- Build in Watchdog timer and CAN Controller
- Open source RTOS is available
- Build in analog to digital converter
- On-chip 3.3V voltage regulator
- Operating voltage range from 1.8V to 5V
- Up to 64kB Flash memory

5. Software Development

This section describes the software development for the hardware described above. Software was developed for CAN interfacing with DHU. Since, there is no simulation model available for CAN and selected microcontroller, CAN software was directly tested on hardware. Software was also developed for Interfacing of LCD, Digital TM and TC circuitry. Since, simulation model for PIC18f66k80 is not available; software was first built for PIC16f877a as its simulation model is available. PIC16f877a is also similar and software wise compatible with PIC18f66k80. Figure 7 shows the testing of TM software using simulation, whereas Figure 8 shows the flow chart of software implementation of DHU on microcontroller. At start, LCD, ADC and CAN modules were initialized and CAN's buffer register started collecting the TMs. A timer

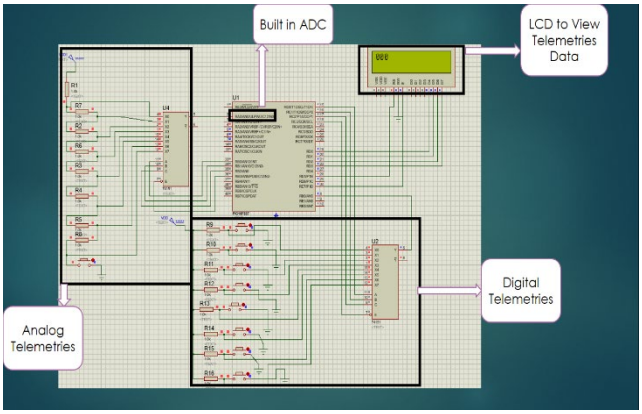

Figure 7. Simulation based testing for telemetry signals.

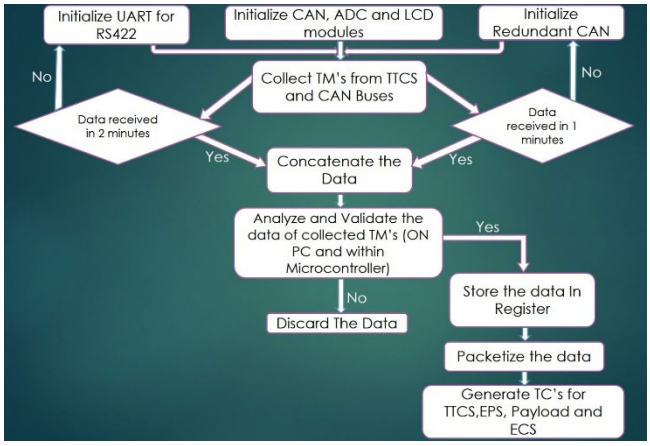

Figure 8. Software Flow Chart of DHU on microcontroller.

was implemented for monitoring activity of communication links. If the system stayed idle for 60 seconds, the redundant CAN turned on. If no data is received even on the redundant CAN for 120 seconds, then UART module is initialized as the last resort. After receiving, data was concatenated and analyzed for validation. If the data was found to be correct, it was saved in register and later on it was packetized using AX.25 protocol. This packetized data is known as TC and transmitted to relevant subsystem.

In the simulation based testing, a test environment was created consisting of circuitry for Digital and Analog Telemetries, microcontroller and LCD. Analog and digital telemetries data were displayed on LCD. In case of digital telemetries data was collected bit by bit using mux, however to collect analog telemetries in built ADC was used along with muxes. Interfacing of in built ADC was done with DHU for collecting Analog Telemetries from other sub systems. Similarly, software was developed for the interfacing of UART to analyze the data of TM's and TC's on PC using hyper terminal software. DHU software was operating at a clock speed of 50MHz. 24 bi levels digital telemetries were stored in 3 bytes register after concateation, whereas 16 analog telemetries were stored in 16 registers of 1 byte each. This DHU software analyzes the received T data and then packetizes it according to AX.25 protocol for ground transmission through TTCS. Watchdog timer was also used n DHU software. This was used to reset the PIC microcontroller if the program gets stuck somewhere or the PIC gets caught into an infinite loop.

6. Results

6.1 Modules Testing

Testing was first performed individually for each module. Data on CAN bus was first tested using an oscilloscope. Figure 9 shows the data analysis of CAN on oscilloscope. A square waveform between 0 and 5V confirmed that data is being carried on the bus. CAN communication was further tested using two test boards each having LCD, PIC18fXXk80 series microcontroller (with build in CAN controller and crystal oscillator) and CAN transceiver. Test boards are shown in Figure 10. LCD was working in 4-bit mode. One board contained PIC18f46k80 microcontroller, which has 34 I/O ports required for telemetry collection and other operations. The other board contained smaller size PIC18f26k80, since only LCD needs to be connected with it. PIC18f46k80 collected telemetries, displayed the telemetry data on LCD, packetize it as per AX.25 and transmitted them on CAN bus. This transmitted data was collected by the other microcontroller (PIC18f26k80) and that microcontroller displayed the data on the other LCD. Both LCDs had same data, which confirmed successful testing. Another test board was used for testing TC and TM circuit. This test board contained switches and variable registers with digital and analog Muxes. In order to generate bi level telemetries on off switches were used, whereas to generate analog telemetries variable resisters were used. In the proposed system, UART is used for communication if dual redundant CAN

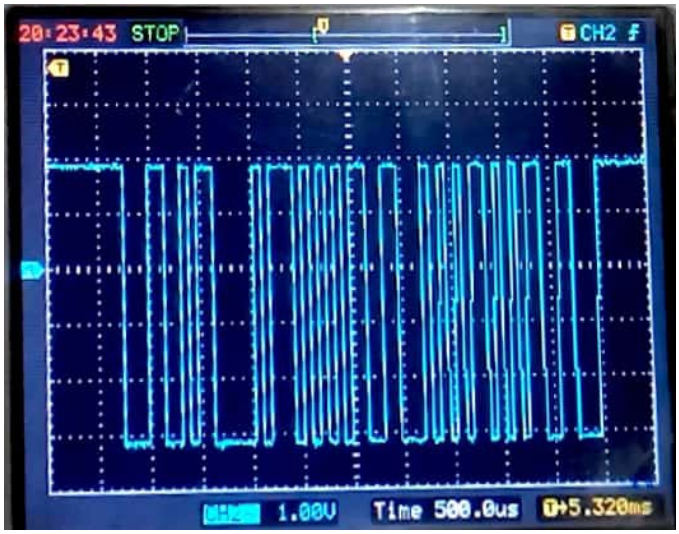

Figure 9. CAN data displayed on an oscilloscope.

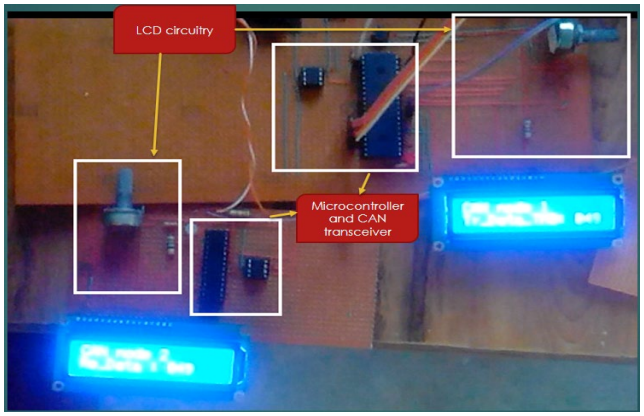

Figure 10. Boards developed for modules testing.

fails. Thus, data of telemetries and telecommands were also analyzed using UART. It was tested using prolific USB to serial cable. Data was transmitted from microcontroller at a baud rate of 9600. This data was analyzed on PC using hyper terminal software. Correct data on hyper terminal confirmed successful testing.

6.2 System Testing

After the successfully testing individual modules, system testing was performed. Telemetries were collected directly by DHU using bi level switches, variable registers and via CAN. On turning on the DHU, it initialized all the modules i.e. Redundant CAN, ADC, LCD and UART. Initially DHU read data from direct telemetries only but it starts using data from redundant CAN in case direct telemetries failed. Capability of DHU software to use the system redundancy was checked. In order to check the redundancy of the system, connection between DHU and CAN was disconnected for more than 60 seconds. After 60 seconds DHU stopped reading data from direct TM's and used redundant CAN data. Redundant CAN received data was analyzed and found to be correct in each case. Similarly, when redundant CAN was disconnected for more than 120 seconds, it started sending data on RS422 and that data was analyzed on hyper terminal software. Similar to the previous cases, correct data was received in this case as well.

A similar methodology was used to test the TC's communication. Its data was also analyzed using hyper terminal on PC and LCD. Communications through each module i.e. CAN, Redundant CAN and UART, was analyzed and found to be correct.

Thus, the proposed system has been successfully tested along with all the incorporated redundancies.

7. Conclusion

Data Handling Unit of CDHS subsystem was developed for a microsatellite. CDHS subsystems of existing satellite were first analyzed. On the basis of analysis, wired network with CAN protocol was selected. Initially, an FPGA based design was proposed and analyzed in light of the system requirements. However, the design did not meet RTOS requirement. Therefore, the design was modified to microcontroller based design. PIC18f66k80 was selected as central processor in this design. Software was developed for TM, TC and redundant CAN. Watchdog timer functionality was also added to prevent run time errors. First module wise testing was performed and each module was individually tested. Two test boards were developed to test the software operation. These test boards consisted of LCD, PIC microcontroller (with build in CAN controller) and CAN transceiver. Finally, entire system testing was performed, along with system redundancy by disconnecting system connections.

The proposed system can be further improved as follows:

- Quad UART and SRAM interfaces are not presented in this paper. This can be designed and included in the proposed system.
- The proposed DHU can be used for data handling purpose in variety of satellite systems. It has the capability to complete the process of receiving telemetries and distributing TCs within a second, so it can be used in weather monitoring satellites resulting in efficient response to emergencies. It can also be used for data handling in aircrafts for communication of flight sensors and navigation data. Similarly, it can also be used in automobiles for effective communication between sub-systems.

8. Acknowledgement

This paper is based upon work supported by Space and Upper Atmosphere Research Commission (SUPARCO), Lahore, Pakistan.

9. References

1. Scholz A. Command and Data Handling System Design for the Compass S-1 Picosatellite. Proceedings of the 5th IAA Symposium on Small Satellite for Earth Observation, Berlin, Germany. 2005; p. 1-5.PMid:15841397.
2. Comsa J, Sharma N, Rolle P, Khan O. QuickSAT Command and Data Handling Unit Preliminary Design Review. Jupitron, Toronto, Canada. 2010.PMCid:PMC2979832.
3. Hamann RJ, Bouwmeester J, Brouwer GF. Delfi-C3 Preliminary Mission Results. Proceedings of the 23rd Annual AIAA/USU Conference on Small Satellites, Logan, UT, USA. 2009; p. 1-11.PMid:19246323.
4. Command and Data Handling (CDH). Available from: http://elfin.igpp.ucla.edu/spacecraft-subsystems/cdh/. Date accessed: 30/04/2018.
5. Datasheet for SJA1000 Stand-alone CAN controller. Available from: https://www.nxp.com/docs/en/data-sheet/SJA1000.pdf. Date accessed: 30/04/2018.
6. Datasheet for MCP2515 CAN controller. Available from: http://ww1.microchip.com/downloads/en/DeviceDoc/20001801H.pdf. Date accessed: 30/04/2018.
7. Datasheet for MCP2551 High speed CAN Transceiver. Available from: http://ww1.microchip.com/ downloads/en/DeviceDoc/21667E.pdf. Date accessed: 30/04/2018.
8. Datasheet for PIC18fxxK80 Family. Available from: http://ww1.microchip.com/ downloads/en/DeviceDoc /39977e.pdf. Date accessed: 30/04/2018.
9. Datasheet for TL16C754B QUAD UART with 64-byte FIFO. Available from: http://www.ti.com/lit/ds/symlink/tl16c754b.pdf. Date accessed: 30/04/2018.
10. OSA RTOS. Available from: http://www.pic24.ru/doku.php/en/osa/ref/download/intro. Date accessed: 30/04/2018.
11. PICOS18. RTOS for PIC18 µC. Available from: http://www.microchip.com/forums/m43743.aspx. Date accessed: 30/04/2018.
12. FreeRTOS. Available from: https://sourceforge.net/projects/freertos/. Date accessed: 30/04/2018.