International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 04 | Apr 2019 www.irjet.net p-ISSN: 2395-0072

Design and Implementation of Telemetry Encoder for light-weight Balloon Payloads

B.V.N. Kapardhi[1], D. Anand[1], Anurup Ojha[2], Dr. B. Ramachandran[2], Dr. D. K. Ojha[3]

[1]Balloon Facility, Tata Institute of Fundamental Research, Hyderabad, India
[2] Department of Electronics and communication engineering, SRMIST, Chennai, India
[3]Department of Astronomy and Astrophysics, Tata Institute of Fundamental Research, Mumbai, India.

Abstract - *Pulse Code Modulation (PCM) telemetry conforming to Inter Range Instrumentation Group (IRIG-106) standards is widely used in high altitude scientific ballooning. These standards are defined by the Range Commanders Council (RCC) in order to make Telemetry systems interoperable [1]. This paper presents the design of hardware and software to implement PCM Telemetry encoder for light-weight balloon-borne payloads.*

Key Words: Scientific ballooning, Telemetry Encoder, IRIG-106, PCM, Payload, Decoder.

1. INTRODUCTION

For over five decades, the Balloon Facility of Tata Institute of Fundamental Research (BF-TIFR) located at Hyderabad has been conducting balloon flights that go upto stratospheric altitudes for scientific investigations in the areas of astronomy, astrobiology [6] and atmospheric science. Several national and international institutions have carried out balloon experiments from BF-TIFR which is one of the few unique launch sites in the world. Scientific payload along with telemetry, tele-command, ballast, GPS and air-safety devices are carried aloft a plastic balloon upto at an altitude of about 42km (near-space) above mean sea level. The science data along with balloon position and the health parameters of the on-board systems are transmitted down through Telemetry encoder, transmitter and antenna. The Ground Tracking station receives, decodes displays and stores the data for post-flight analysis. Once the balloon flight mission is completed, the balloon is detached and destroyed through uplink telecommand. A parachute connected to the balloon system gets deployed subsequently and the whole instrument descends down to ground safely. The payload and all the electronics is recovered and is reused in future missions [2] making scientific ballooning cost-effective for conducting research. In the recent times, due to the rapid advancements in technology, payload size and weight is getting smaller steadily in order to minimize the cost of a balloon flight. Added to that, the onboard system is made autonomous and data is stored onboard leaving only few essential parameters to be transmitted to the ground station [8]. Therefore, a low-complexity, light-weight telemetry encoder is designed and implemented in order to meet these requirements.

2. BRIEF OVERVIEW OF TELEMETRY SYSTEM

PCM Telemetry is the process by which an object's characteristics are measured and the results are transmitted to a distant station, using PCM and Time division multiplexing. At the Ground Station the received data stream is reconstructed by a bit synchronizer to the original serial square wave train [5]. Then, a decommutator or decoder recognizes the Frame synchronization word, extracts the original measurands and converts the serial digital stream to parallel data for display, analysis and storage. Frame synchronization word is a pseudorandom sequence of 1's and 0's that is unlikely to occur randomly in the acquired word and occupies two or more words in a minor frame [3]. Optimum frame synchronization pattern for various telemetry formats is listed in the Appendix C of IRIG-106 and these standards have class I and class II distinctions based on the data rate and data formats [1][9]. This paper deals with class I format. A simple Telemetry system is shown in Fig. 1.0.

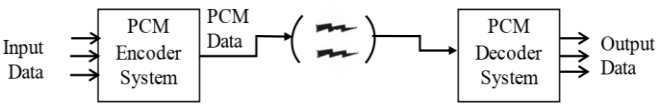

Fig -1: Telemetry system.

3. DESIGN ASPECTS OF TELEMETRY ENCODER

As stated earlier, the Telemetry Encoder (TE) is designed to cater to light weight payloads that require few parameters to be measured. The specifications of the TE designed and implemented are as given under:

Format	: IRIG-106 Class I
Output Data Rate	: 40 kbps
Output Format	: PCM BiØ-L
Word Length	: 8 bits/ word
Minor Frame Length	: 12 words/ Minor Frame
Major Frame Length	: 256 Minor Frames/ Major Frame
Frame Sync Pattern	: 24 bits (0x FAF320)
Sub-Frame Sync method	: Sub-Frame IDentification (SFID)

W0, W1 and W2 are the Frame sync words that appear in every minor frame. W3 is the sub-frame Identifier which is nothing but a minor frame identifier. It has the value of 0x00 for the first minor frame. The value increments for every

minor frame till it reaches a value of 0xFF for the 256th minor frame after which it resets to 0x00 and starts over again. W4 and W5 are digital data words. W6 to W11 are analog data words. The telemetry frame structure of the Encoder implemented is shown in Fig. 3, where W0 to W11 is the minor frame and 256 minor frames having SFID 0x00 to 0xFF form one major frame.

The complete circuit diagram and the timing signals of the TE are shown in Fig. 2 and Fig. 4 respectively.

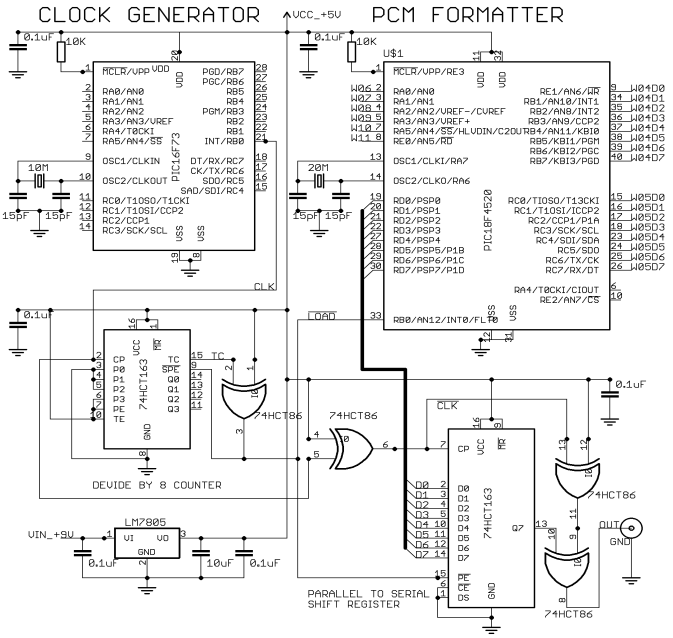

Fig -2: Circuit diagram of the Telemetry Encoder

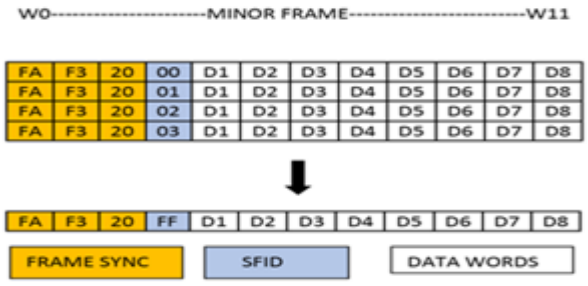

Fig -3: Frame structure of the Telemetry Encoder

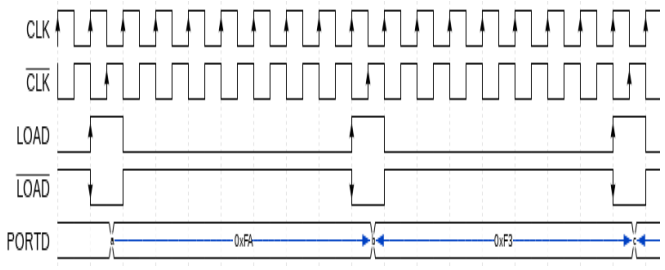

Fig -4: Timing diagram of signals

3.1 Code for Clock generator - PIC16F73

```
1.  #define True 1
2.  volatile unsigned char counter=0; //count 0-255
3.  void main()
4.  {
5.  TRISB=0;              //Set port B as output port
6.  counter=0;            // counts from 0-255
7.  while(True)
8.  {
9.  PORTB = counter;  // move counter value to port B
10. counter++;            //increment counter
11. Delay_us (8);         // Wait for 8 us
12. asm nop;              //  add 400 ns delay
13. asm nop;                    //  add 400 ns delay
14. asm nop;              //  add 400 ns delay
15. }   // total 12.5 us time delay is obtained.
16. }
```

3.2 Code for PCM formatter – PIC18F4520

```
1.  #define SYNC1 0xFA
2.  #define SYNC2 0xF3
3.  #define SYNC3 0x20
4.  #define MINORFRAMELEN 11                    // 0 to 11
5.  #define True 1
6.  volatile char minorframe[12];//Array to store 12 words
7.  volatile unsigned char word_counter=0;     // count 0-11
8.  volatile unsigned char subframe_id=0;      //count 0-255
9.  volatile unsigned char  i;                  // count 0-5
10.volatile unsigned int temp_res;         //holds temporary
                                                ADC value
11. void main ( )
12. {
13. TRISA = 0x3F;              // Initialize  Port A  as INPUT
14. TRISB = 0xFF;                    //  Port B  as INPUT
15. TRISC = 0xFF;                    //  Port C  as INPUT
16. TRISE = 0x07;                    //  Port E  as INPUT
17. TRISD = 0x00;                  //  Port D  as OUTPUT
18. ADCON1= 0x09;           // AN0-AN5 as analog INPUTS
19. ADC_Init();                  // Initialize ADC module
20. memset(minorframe,0x00,12);  //set all words to 0x00
21. minorframe[0]= SYNC1;                  // Load 0xFA
22. minorframe[1]= SYNC2;                   //Load 0xF3
23. minorframe[2]= SYNC3;                   //Load 0x20
24. INT0IF_bit = 0;              // Clear INT0 interrupt flag
25. INTEDG0_bit=0           //INT0 to occur  on falling edge
26. INT0IE_bit = 1;                // Enable INT0 interrupt
27. GIE_bit   = 1;                 // Enable Global interrupt
28. while(True)
29. {
30. for(i=0;i<6;i++)// store ADC words in minorframe array
                                loop here until interrupt occurs
31. {
32. temp_res=ADC_Get_Sample(i);//Store 10bit ADC value
33. minorframe[i+6]=temp_res>>2;//Store 8bit ADC value
                                              for  transmission
34. }                                          //end for loop
35. }                                        // end while loop
36. }                                           // end main
```

```
37. void interrupt( )     //Interrupt INT0 service routine
38. {
39. if (INT0IF_bit)              //Continue if INT0 flag is set
40. {
41. if (word_counter > MINORFRAMELEN)
42. {
43. word_counter = 0;
44. subframe_id++;
45. minorframe [3] =subframe_id;
46. minorframe [4] =(PORTB&0xfe) | (PORTE&0x02)>>1);
47. minorframe [5] =PORTC;
48. }
49. LATD=minorframe[word_counter];  // send to port D
50. word_counter++;     //Increment word counter
51. INT0IF_bit = 0;    //clear INT0 flag to receive interrupt
52. }
53. }                          // end Interrupt service routine
```

3.3 Flow chart of the Program execution

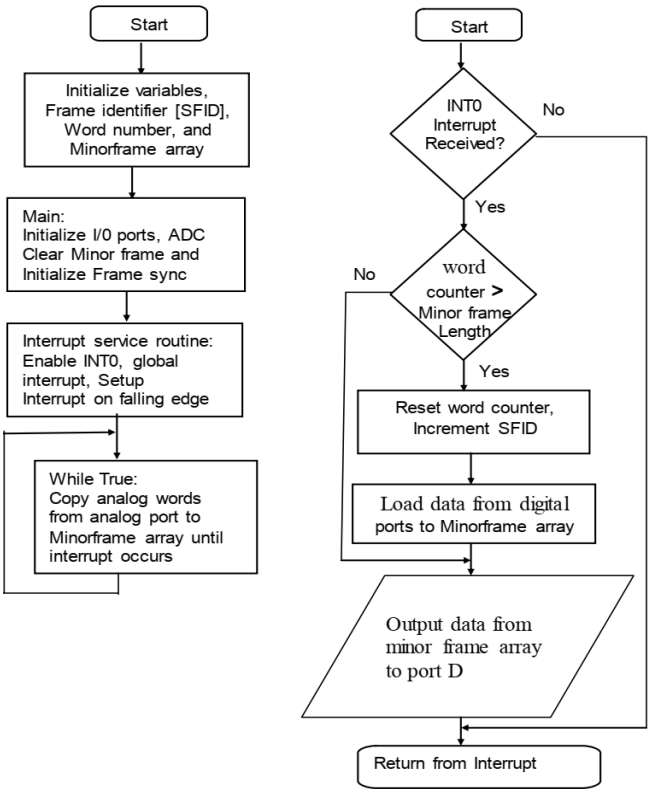

Fig -5: Flow chart of the main program (left) and Interrupt service routine (right).

4. HARDWARE AND SOFTWARE DESCRIPTION

The design was implemented using Peripheral Interface Controller (PIC) PIC18F4520, PIC16F73 and Integrated Circuits 74HCT163 four bit binary counter, 74HCT166 eight bit parallel-in serial-out shift register, 74HCT86 quad two inputs Exclusive-OR gate and a LM7805, 5V regulator to power the circuit. The code for both the PIC Microcontrollers was written and compiled in the mikroC PRO for PIC IDE. mikroC PRO for PIC supports about 808 types of MicroController Units(MCU) and with 1000 library functions the development time and effort is drastically simplified and reduced.

The clock generator code was optimized to generate 40 KHz clock on RBO of PIC16F73 with 10MHz oscillator (RB0 toggles every 12.5 us). 74HCT163 is preset to binary eight to function as a divide-by-8 counter (P3 high, P2, P1, P0 low) resulting in a Load pulse which is inverted by the EX-OR gate to get the load_bar pulse. The parallel enable pin of the counter is active low and hence load_bar is needed to load the preset value for every 8 rising edge of the clock. The same load_bar is used to generate a hardware interrupt on PIC18F4520 which is programmed to accept interrupt on falling edge of a pulse at pin RB0/INT0. RE1 is used in place of RB0 to accept 8 data bits using code (PORTB&0xfe) | (PORTE&0x02)>>1. ADC_Init( ) is the inbuilt function that is used to initialize the Analog to Digital Converter (ADC) module of PIC18F4520. Ports A, B, C and E are initialized as input ports to take digital and analog data. The formatted output is sent through Port D which is initialized as output port.

The main program after initializing the variables, ports, ADC module and interrupt related bits, remains in the While loop reading analog words and storing them in the minor frame array. When INT0 interrupt occurs, control is transferred to the interrupt service routine (ISR) which outputs one word through Port D every time an interrupt is received until all words in the minor frame array are transmitted. Once the word count is greater than minor frame length, word count is made 0 to start counting from 0 to 11 all over again. And the SFID is incremented to indicate transmission of next frame. Once all the 256 minor frames are transmitted the SFID is reset to 0.

The output from Port D is given to 74HCT166, an eight bit parallel-in serial-out shift register which shift the 8 bit data out serially in synchronous with its clock input. The serial output from the shift register is in Non-Return to- Zero (NRZ-L) form. NRZ-L serial stream is given to one input and clock is given to the other input of EX-OR gate to convert it to BiØ-L format. The inbuilt ADC of PIC18F4520 is 10 bit wide. But only the 8 most significant bits are transmitted. At the Decoding end, it is aligned to get the right value.

Arduino boards and PIC microcontrollers, due to their low-complexity are quite popular in High altitude balloon missions [8]. Using PIC Microcontroller was the preferred choice due to the reason that have better drive capability, fast conversion speed inbuilt ADC and are easy to implement.

5. SIMULATION AND RESULTS

The Proteus Design Suite is widely used by electronic engineers for circuit simulation, drawing

schematics and designing PCB's too. It can be simulated with additional electronic circuits connected to the MCU. For simulation with MCU's in Proteus, the compiled hex file is applied to the circuit in the design suite. This design too went through several simulations in Proteus before finalizing the software and hardware. The schematic circuit drawn in Proteus and the simulation of the complete circuit is shown in Fig.6.

The wave forms seen on the Proteus Oscilloscope from top to bottom are Clock, Load_bar, Frame sync in NRZ-L and Frame sync in BiØ-L format. The design was also verified by conventional method such as checking the various pulses on a physical oscilloscope.

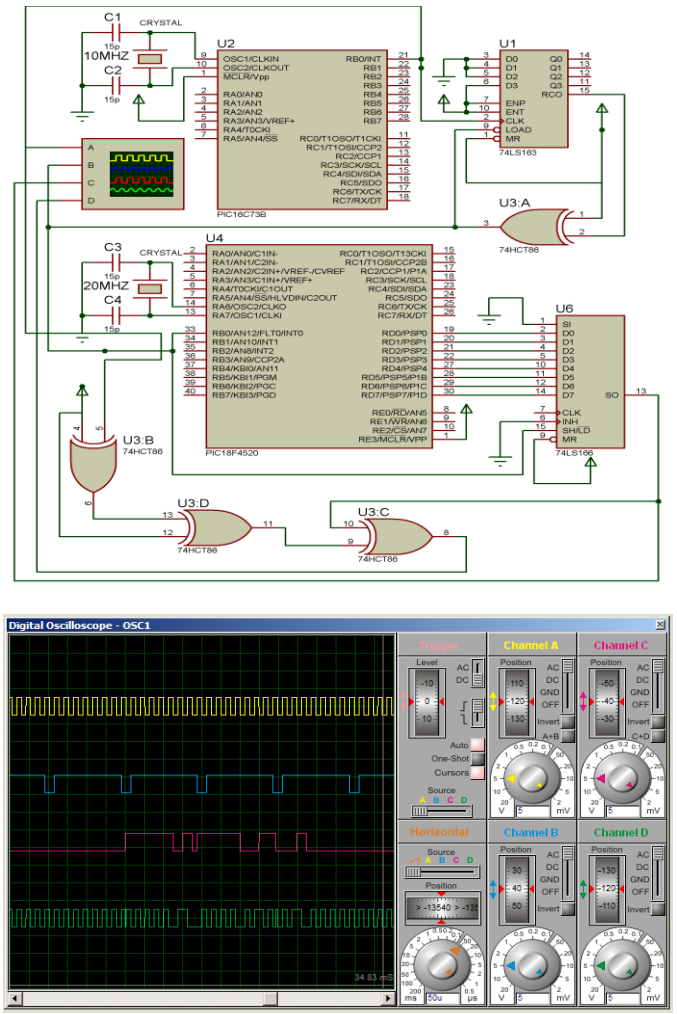

Fig -6: Hardware and hex file loaded in Proteus (top) and simulation results seen on Proteus Oscilloscope (bottom).

Finally the TE was fed with real digital and analog data and its output was given to the standard Decoder at BF-TIFR. The Decoder is programmable and can accept any standard IRIG-106 class I telemetry format [7].

All the words could be extracted, displayed and stored as shown in Fig. 7. The prototype model of the designed Telemetry Encoder is shown in Fig. 8.

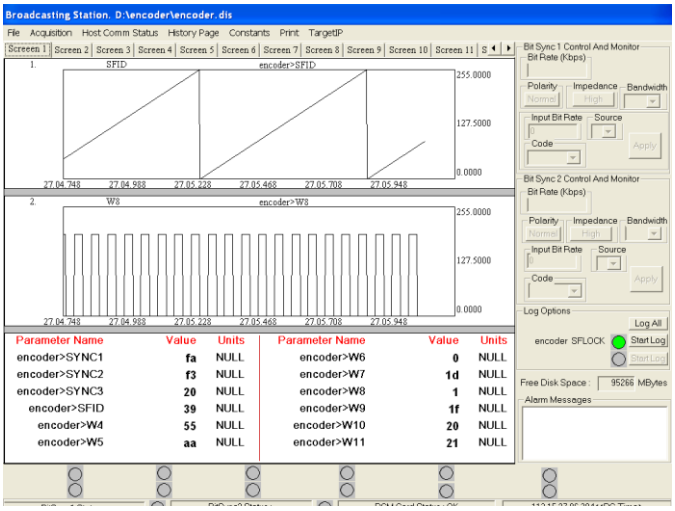

Fig -7: Hardware decoding and display of all the parameters

Fig -8: Prototype model of Telemetry Encoder

6. CONCLUSIONS

A compact PCM Telemetry Encoder for balloon-borne scientific payloads has been designed and implemented successfully. The telemetry encoder is especially designed to cater to ever increasing demands of small light-weight payloads due to the fact that higher altitudes can be achieved with lighter payloads [2].

The Complete design could be simulated in the Proteus software and thus the development cycle time could be reduced. It was tested extensively with the IRIG-106 standard telemetry decoder at BF-TIFR and its performance was found to be highly reliable and robust.

Another noteworthy feature implemented in the design is the use of an independent Micro-controller for generating the bitrate that makes the bitrate of the encoder programmable

by merely changing the code for the microcontroller. No change of hardware is required.

The same design when implemented with surface mount devices will have a very small foot- print and would weigh less than half of this prototype model making it a right candidate for light weight payloads. In addition, a low complexity design approach has been adopted to ensure easy maintainability.

Standard telemetry Encoders at BF-TIFR have large telemetry capability meant for heavy astronomy and atmospheric science payloads. It is obvious that they will be bulky and consume more power. This is, therefore, a positive step forward to design and develop a customized telemetry encoder.

ACKNOWLEDGEMENT

The authors would like to thank Mr. Ashish Rodi, scientific officer, TIFR, Mumbai for his support and work in simulating the design in Proteus.

REFERENCES

[1] Range commanders council Telemetry group, "Telemetry Standards, IRIG Standard 106-13, document number (Part 1)," June 2013.

[2] NASA, "NASA Facts", Goddard Space Flight Centre, Wallops Flight Facility, Va. USA, 2006.

[3] L-3 communications, "Telemetry Tutorial", 9020 Balboa Avenue, San Deigo, CA, USA.

[4] A. Sushko, A.Tedjarati, J. Creus-Costa, S. Maldonado, K. Marshland & M. Pavone, "Low Cost High Endurance, Altitude-Controlled Latex Balloon for Near-Space Research (ValBal)", Proceedings of the 2017 IEEE Aerospace Conference, 2017.

[5] G. Sadhukhan, M. Sandhu, R.P. Singh , "VLSI based implementation of PCM MUX encoder for range telemetry system", in Proceedings of the IEEE INDICON 2004, First India Annual Conference, 2004.

[6] Narlikar J.V., Lloyd D., Wickramasinghe N.C., Harris M.J., Turner M.P., Al-Mufti S., Wallis M.K.,Wainwright M., Rajaratnam P., Shivaji S., Reddy, G.S.N., Ramadurai S., Hoyle F. " A Balloon Experiment to detect Microorganisms in the Outer Space", Astrophysics and Space Science, 285: 555-562, 2003.

[7] R. K. Manchanda, "Current Developments and Future Plans at NBF Hyderabad and Prospects for Long duration Balloon Flight" Trans. JSASS Aerospace Tech. Japan, Vol. 8, No. ists27, pp. Tm_55-Tm_62, 2010.

[8] Paul Clark, Marc Funk, Benjamin Funk, Tobias Funk, Richard E. Meadows, Anthony M. Brown, Lun Li, Richard J. Massey and C. Barth Netterfield, "'An open source toolkit for the tracking, termination and recovery of high altitude balloon flights", Journal of Instrumentation, vol. 14, 2019 (arXiv:1904.04321 [astro-ph.IM]).

[9] Lawrence J. Mueller, "Synchronization of pulse code modulation telemetry", Master's Thesis, Missouri University of science and technology, 1968.

BIOGRAPHIES

B V N Kapardhi received A.M.I.E. degree in Electronics and Communication Engineering. He is working as scientific officer-C at the Balloon Facility of TIFR, Hyderabad in Telemetry division.

D Anand obtained his M.Tech (by research) degree from Anna University and is working as scientific officer-E at Balloon Facility of TIFR, Hyderabad. His research interests include FPGA, GPS, telemetry and telecommand.

Anurup Ojha, a final year student of Electronics and Communication Engineering, SRMIST, participated in the design as part of his final year project.

Dr. B. Ramachandran is a Professor, Dept. of ECE, SRMIST, d Kattankulathur, Chennai. His research interests include Ad hoc and sensor networks, signal processing and antenna design.

Prof. D. K. Ojha obtained his PhD in Astronomy and Astrophysics from Strasbourg University, France. He is currently working in the Infrared Astronomy Group of TIFR, Mumbai.