Implementation of MIL-STD1553 using Microcontroller Atmega 328P

Shivani Shinde[1]

[1]*Under Graduate Student, Electronics and Tele-Communications Department, Vishwakarma Institute of Technology, Pune, Maharashtra, India*

Abstract - *This paper presents a method that implements MIL-STD-1553 point to point transmission of data into a hardware platform and hence, put forward a method to implement it in Arduino Microcontroller. The paper puts a light upon MIL-STD-1553 bus system and also proposes the implementation of the MIL-STD-1553 bus controller and remote terminal in 20 bits word format including command word, data word and status word in Arduino ATmega 328P Microcontroller.*

Key Words: **Arduino ATmega 328P Microcontroller, Bus Controller (BC), Command Word (CW), Data Word (DW), MIL-STD 1553 protocol, Remote Terminal (RT), Status Word (SW).**

1.INTRODUCTION

MIL-STD-1553 is a multiplex bus standard which was developed by U.S Air Force for incorporation of data in the military aircrafts and avionics. There are two specifications of MIL-STD-1553 MIL-STD-1553A in 1975 followed by MIL-STD-1553B in 1978. Considering the initiation in 1973 it has encountered a number of emendations, the first tri-utility version (for Navy, Army and for Air force) is released and then forth, after few changes and developments it has been unbolted for commercial developments as well. At present it has flourished as one amongst the internationally trusted networking standard and an inexorable component used in ships, missiles, satellites and International Space Program and also in advanced avionic applications.

A single bus in MIL-STD-153 comprises of a pair of wire with 70-85 Ω impedance and 1MHz frequency. It uses a circular connector. The center pin of connector is used for Manchester bi-phase signal. The transmitter and receiver braces to the bus through the isolation transformers, and the sub connections diverge off using a couple of resistors and a transformer. The Manchester code is used in attendance to both data and clock on a single wire and to get rid of and DC component in the signal.

This paper presents the design of the standard MIL-STD-1553 protocol controller using Atmega 328P and implementation of the protocol using standard MIL-STD-1553-word format. The implantation and testing of design are done with the help of hardware platform.

Table-1: Descriptions of MIL-STD-1553 protocol

Data Rate	1MHz
Word Length	20 bits
Data Bits / Word	16 bits
Message Length	32 data words (maximum)
Transmission Technique	Half-duplex
Operation	Asynchronous
Encoding	Manchester bi-phase
Protocol	Command and Response
Bus Control	Single or multiple
Fault Tolerance	Dual Redundant
Message Formats	Control to terminal Terminal to terminal Terminal to Controller System control Broadcast
Number of Remote Terminals	31 (Maximum)
Terminal Types	Remote terminal Bus controller Bus monitor
Transmission Mode	Twisted shielded pair
Coupling	Transformer and direct

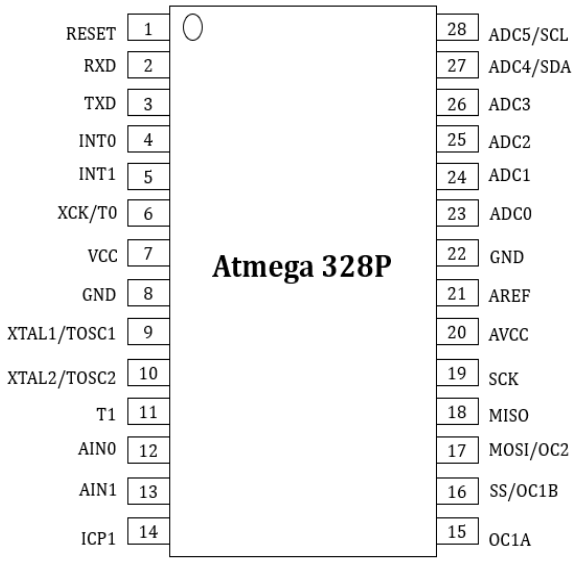

Fig-1: Arduino ATmega 328P pin diagram

2. LITERATURE REVIEW

MIL-STD-1553 controller comprises of bus terminal and remote controller. Based on Bus controller and Remote terminal six types of transactions are allowed. This mainly includes Bus controller to Remote terminal, Remote terminal to Bus controller and Remote terminal to Remote terminal word transfer. Bus controller over the bus, it initiates message communication. Remote terminal mainly provides an interface between the data bus and the sub-system, also it creates a bridge between one bus to another. Bus monitor works like a server. It mainly monitors and records all the bus transactions. Atmega 328P is a very basic single-chip microcontroller developed by Atmel. It has a modified Havard Architecture and 8-bit RISC processor core. The main reason behind choosing this microcontroller is it has 23 I/O pins and can work at 1MHz frequency required for the MIL-STD-1553 protocol. In this paper we have mainly tried to explain the Bus controller to Remote terminal word transfer in Atmega 328P.

3. SYSTEM OVERVIEW

As discussed in the literature review MIL-STD-1553 protocol has Bus controller and Remote terminal. According to the design of MIL-STD-1553 there can be 31 different Remote terminals and Bus terminals which is used for performing different tasks and for controlling a particular activity. For transferring of data is operated using data word.

Earlier MIL-STD-1553A was the only model introduced but to overcome its limitations MIL-STD-1553B model was introduced. The previous model did not comprise of a broadcast option but the later version had the broadcast option which is required for broad casting of messages not only to bus controllers but also to other remote terminals and the bus monitor.

A. Bus Protocol:

MIL-1553 messages on bus accommodate more than 16-bit words, that are classified as command word, data word and the status word. Every word is always leaded up by 3µs synchronization pulse and is always accompanied by an odd party bit. Since, the 3µs synchronization pulse consists of 1.5µs low and accompanied by 1.5µs high pulse Manchester bi-phase signal is used for the working of this protocol. Around 4µs gap is placed between to consecutive messages. If a particular device exceeds the time of 14µs then, it is considered that the command message is not received by that device.

Table-2: Input/output description of Encoder

S. No	Parameter	Input/Output	Description
1	clk	Input	2 MHz
2	rst	Input	1 clock cycle
3	start	Input	2 clock cycles
4	cmd_data	Input	Cmd or data sync
5	info	Input	16-bit hex data
6	man-enc	Output	Serial Manchester encoding
7	busy	Output	Indicates the end of Manchester encoding

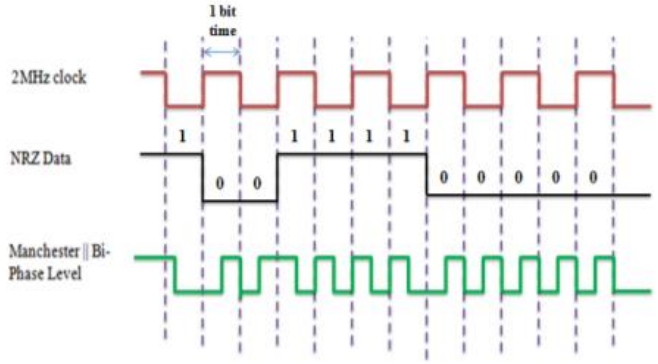

Fig-2: Manchester II Bi-Phase Level Encoding

B. Word Formats:

Every word comprises of a unique format within the ordinary structure. Each word is 20-bits in length who's preceding 3-bits are synchronization bits, which basically sanctions the decoding clock for resynchronization at the start of each word. The next 16-bits contain the data to be transmitted and the last bit is parity bit which always an odd parity for a single word. All encoding of bit is built upon Manchester Bi-phase encoding format. It provides a self-clocking waveform. The signals produced are mainly symmetrical about zero. A logic of 0 and 1 is mainly based upon transition between positive and negative levels. It is logic 0 from negative to positive and 1 from positive to negative. The dedicated waveforms can be understood by the figure.2.

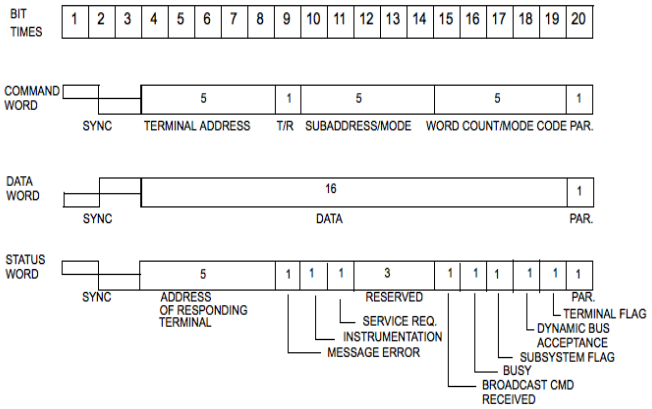

Fig-3: Word Format for MIL-STD-1553

a. Command Word:

A Command Word utilizes the first 3-bits as synchronization bits. The next 5-bits for the address of the RT. The next bit is transmit/receive (1 for Transmit and 0 for receive). The proceeding 5-bits specifies the sub address or mode code bits. The next 5-bits are defined as Word Count bits. They define the word count of the task to be performed and the last bit is parity bit which always defined as odd parity as explained in figure.3.

b. Data Word:

The first 3 bit usually are synchronization bits and the next 16-bits contain the actual data to be transmitted and the las bit is odd parity bit as explained in figure.3.

c. Status word:

The first 3-bits of the status word are also the synchronization bits. The proceeding 5-bits are Terminal address. The next bit defines the message error bit, which is set by terminal when an error is detected. Message error bit is proceeded by instrumentation bit which is always set to 0. The next bit which is service request bit is used to inform BC by terminal that it needs to be received. Bits 12, 13 and 14 are reserved bits for future use. Bit 16 is a busy bit that tells if the terminal is move data or not to other terminals. Bits 17 and 19 are flag bits. Bit 18 defines if terminal has received the word code and has accepted bus control and the last bit is parity bit which is odd parity as explained in figure.3.

4. Methodology

The main method which we have implemented is Bus controller to Remote terminal transfer of data word. In this type of transfer the bus controllers sends the command word, followed by the data word and when the data word is received by terminal the terminal sends the status word back to the controller. In the Remote terminal to Bus controller transfer, the controller sends a command word. The RT after getting command word sends the status word in addition to the data word specified. In remote terminal to remote terminal transfer command word is send to another terminal. Once the command word is received status word is send followed by the data word.

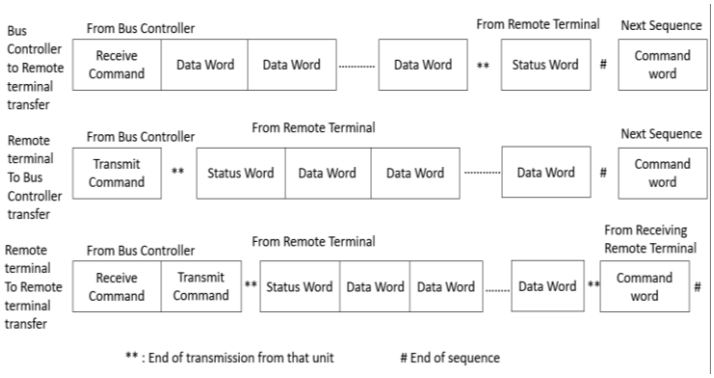

Fig-4: Word Transfer MIL-STD-1553

5. Design and Implementation

The message transfer is based on Mil Std 1533 protocol and communication is obtained using 2 Arduino ATmega 328P microcontroller. Here we are basically constructing a library for Arduino ATmega 328P based on Mil Std 1533 protocol. The digital pins of Arduino ATmega 328P are used to build the communication. One of the Arduino acts as a transmitter and the other acts as a receiver.so the pin configuration is explained in figure.5. The synchronization pin, transmitter, receiver, remote terminal, message error, reserved, instrumentation bit, service request bit, Broadcast received bit, busy bit, flags, dynamic bus control accept pins are defined and address is generated using these pins.by using the synchronization, remote terminal and other pins 20-bit command word is generated. As the data word of mil 1553 is transmitted by Hex coding the data word is constructed by converting a string into a hex address and transmitted and then converted back into string while receiving. When the data word is received the message error bit, instrumentation bit, reserved bit, service request bit, broadcast bit, busy bit, flags, dynamic bus control accept pins are made high and a status word is generated. First is BC to RT and the next one is RT to BC. All of these transfers happen according to the predefined format of the 1553 protocol. In this type of transfer the bus controllers sends the command word, followed by the data word and when the data word is received by terminal the terminal sends the status word back to the controller. The design is done after taking into consideration the working and needs of the 1553 protocol as mentioned in the system structure. Hardware is designed as shown in figure.6.

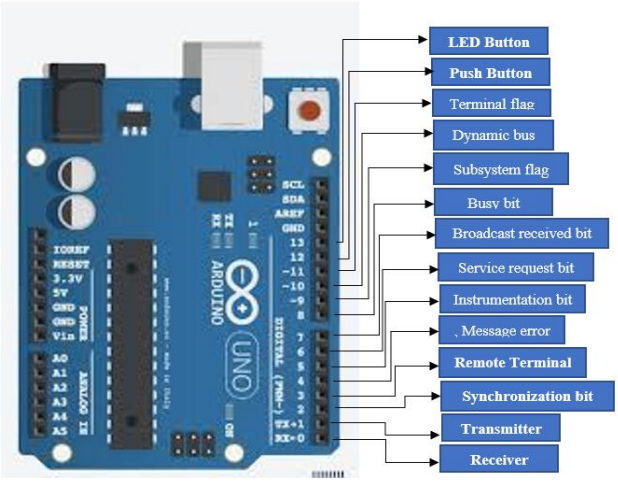

Fig-5: Design of MIL-STD-1553 on ATmega 328P

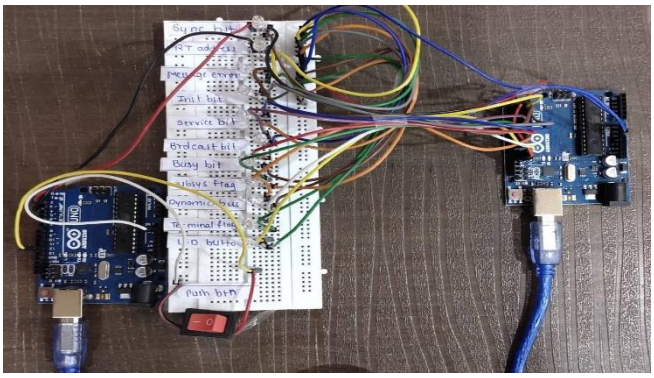

Fig-6: Hardware design of MIL-STD-1553

6. Result

We have tested the protocol in Proteus as seen in figure.6. and also received the results by actually implementing it in hardware the detailed results are shown in figure.7. figure.8. figure.9. and figure.10. To show which pins get in active mode while the communication we have used Led bulbs assure that the only required pins are working and we are getting the output.

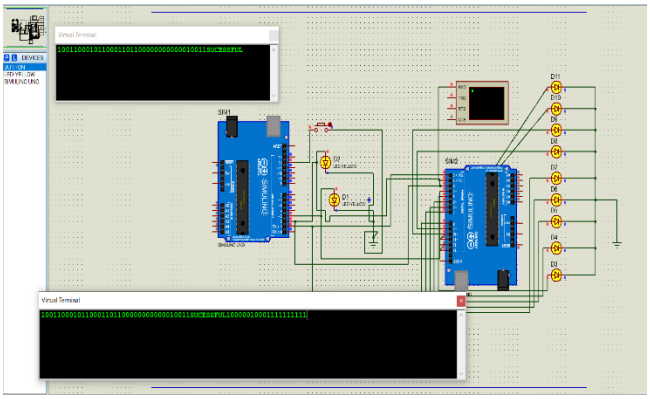

Fig-7: Proteus Simulation of MIL-STD-1553

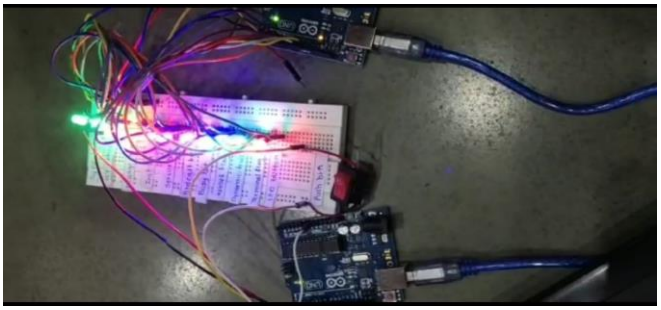

Fig-8: Hardware testing of MIL-STD-1553

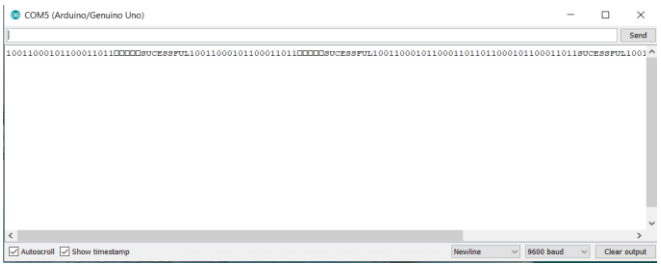

Fig-9: Simulation terminal result of Remote terminal

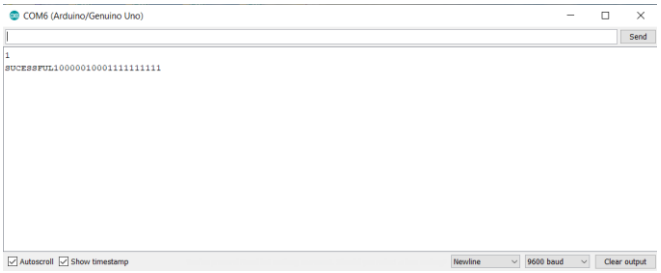

Fig-10: Simulation terminal result of Bus controller

7. Future Scope and Conclusion

In this growing world of development in Artificial Intelligence and other fields MIL-STD-1553 plays a vital role. Implementation of MIL-STD-1553 in other AVR microcontrollers can help in automation and advancement of commercial applications. Development of more advanced libraries based on MIL-STD-1553 has become a necessity as the world is moving towards automation.

This paper describes a new way to design MIL-STD-1553 protocol using ATmega 328P Microcontroller. Since most of the 1553 implementations are on Field Programmable Gate Arrays which are nothing but semiconductor devices, we have designed the Communication channel between the two Arduinos that follows MIL-STD-1553 Protocol. We have tested it ion proteus simulator as well as implemented the hardware. So, it is clear that MIL-STD-1553 will continue to extend its application in more integrated fields and commercial applications too.

References

[1] MIL-STD-1553 (1990) Designer's guide, 3rd edn. Data Device Corporation, Bohemia, New York
[2] MIL-STD-1553 tutorial by Condor Engineering, Inc., USA
[3] MIL-STD-1553 tutorial by AIM GmbH
[4] Department of Defense Interface Standard for Digital Time Division Command/ Response Multiplex Data Bus, MIL-STD-1553B, Notice 4, 1996. Washington, DC: Department of Defense, 1978.
[5] Chen Xi Hui, Leng Xue, Li Wen Ming, Zheng Li Na, "The Remote Terminal Design and Implementation of 1553B Based on DSP", Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences China.
[6] Michael Hegarty, Principal Marketing Engineer Data Device Corporation, "MIL-STD-1553 Goes Commercial" June, 2010
[7] Design and Verification of MIL-STD-1553B Remote Terminal Modules L. Karthik, K.V. Ramana Reddy, Dr. Siva Yellampalli
[8] MIL-STD-1553 DESIGNER'S GUIDE
[9] Arduino Microcontroller Processing for Everyone! Third Edition Steven F. Barrett University of Wyoming, Laramie, WY