Frequently Asked Questions

Title: How Does 1553 Work – A Simple Tutorial
Date: 22 July 2010
Card/Board/Module: M4K1553Px, PCMCIA/Px, ExCARD/Px, UNet/Px
Operating System: n/a

Question:

When an RT sends a message, how does the BC or Monitor know that the data is for a Transmit message ?

How does the Merlin GUI application get this information that it displays on the screen ?

Answer:

This requires a short tutorial on the concepts of 1553 :

We start with a bus, over which the 1553 traffic flows. There are three players: BC (Bus Controller), RT (Remote Terminal), Mon (Bus Monitor).

The BC controls the whole flow of messages. The BC is active, and the other players (RT and Mon) are passive. All messages are initiated by the BC, who sends out the command word. All players on the bus listen for a command word, and determine if it is for them to "pick up."

The **command word** includes information as follows:

- target **RT-SA** (Remote Terminal number, SubAddress number)
- **word count**
- in which **direction** is the data traveling, from the perspective of the target RT: Receive (BC2RT) means that the RT receives the data words (and the BC sends the data words); Transmit (RT2BC) means that the RT transmits the data words

[See Appendix B MIL-STD-1553 Message Formats in the manuals for a description of the different message types in 1553.]

RT status word (see Appendix A MIL-STD-1553 Word Formats in the manuals)

- is either the first word returned by the RT (followed by data), or is sent by the RT after receiving all the data word (depending on the message type)
- contains the RT address, acting as an ACKnowledge to say "I am here"; has a number of other information bits (not mandatory)
- has one bit which can indicate to the BC that the RT wants service (SRQ bit), for which the BC will make a new message requesting this information from the RT

The firmware for the different players (in the given card or box) is built according to the rules of the 1553 protocol. Hence, any player on the bus who receives the command word knows exactly what to expect next (command word, data word, status word), and will report an error if the unexpected occurs.

Our 1553 GUI application, **MerlinPlus**, in each of the modes, takes the command word, parses it, and decides how to display the message based on the parsed result. It displays command word, command type, target RT-SA, word count, error or not.