Frequently Asked Questions

Title: 1553Px family, Major Frames, Minor Frames
Date: 29 April 2015
Card/Board/Module: 1553Px family (M4K1553Px module, PCMCIA/Px and /EP, ExCARD, UNet/Px)
Operating System: all

Question:

I need help understanding the definitions of:

- Major frame
- Minor frame
- Frame time
- Intermessage gaptime

And how all these things work together.

Answer:

1. **Major frame** is the list of all the messages in the current frame. We place a bus list of messages in a FRAME struct, as defined in file bcrun.h, and used in all the BC demo programs: demo_bc1, demo_bc2, demo_2crd, demo_2md, demo_async, demo_cmon, demo_minor.

You can create multiple major frames (call Create_Frame_Px to create each individual frame), and then decide which one is the (current) frame to transmit, using Start_Frame_Px.

You can create an asynchronous frame which can then be transmitted at will, in the middle of the transmission of the major frame, when the current message completes being transmitted over the bus (using functions Select_Async_Frame_Px and Send_Async_Frame_Px). When the async frame completes transmission, control is returned to the next message of the major frame.

2. Within any list of messages, each message is created using function Create_1553_Message_Px. This assigns an id to the message. Each message is added to its desired frame into the FRAME structure, with two parameters: id and intermessage gaptime. The **intermessage gaptime** is the time the firmware is to wait between the completion of one message and the beginning of transmission of the next message.

3. The major frame time (set by calling function Set_Frame_Time_Px) sets the total frame time for all of the messages in our major frame. The firmware processes & transmits all the messages in the major frame, all the while keeping an accounting of how much time was spent from the beginning of the first message transmission (by setting a counter equal to the frame time, and counting down).

If all the messages have been transmitted and the counter is not yet 0, then the firmware waits until the counter gets to 0 before continuing. That is, if the frame is scheduled to be transmitted more than once, either continuous using Run_BC_Px(handle, num_times), where num_times=0, or N-times where num_times is a number between 2-255, the firmware will wait until the counter counts down to 0 before going back to retransmit the frame.

4. **Minor frames** are designed to be used to create groups of messages that need to be transmitted at specific intervals, say every 10 or 20 msec.

For example, if you have

- group A of messages that need to be transmitted every 20 msec,
- group B of messages that need to be transmitted every 10 msec,

- group C of messages that need to be transmitted every 40 msec,

then you determine the lowest common denominator of the time periodicity (in this case 10 msec) and group the messages using Minor Frames. In this case, your Minor Frame (MF) time will be 10 msec. You will group the messages with a "marker" (MF) message in between groups.

The MF message (Create_1553_Message_Px with message type MINOR_FRAME) is used to tell the firmware how long to count down before allowing the firmware to move to the next group of messages. Hence, we use MF markers between every set of messages that form a group. One MF marker message at the beginning of the major frame list, and then groups of messages delineated by MF messages. At the end of the major frame we do **not** place a MF message.

Since we want the MF time to drive the timing, we set the major frame to (Set_Frame_Time_Px) to a very small number, say 10 (this parameter is in microsec).

For the example I started above, your major frame would look like this:
MF-B-MF-B-A-MF-B-MF-B-A-C

5. You can read more about minor frames in the User's Manual (hardware manual), section 3.5.