Duplicate Timers

Data Access Table Scheme

Robert Goodwin

Mon, May 12, 2008

The objective was to have two timers such that the second timer tracks the first timer. The reason was to get more signal drive for the output signal, as it needed to drive a heavy load. This note describes a simple means of providing this using two Data Access Table entries. Note that the scheme is not limited for use with timers, but could also be used for D/A's.

Use the RDATA entry type 0x24 for monitoring "counter word differences." Monitor the first timer reading (or setting) field, to see whether it changes. The difference is placed in the reading word of a dummy channel, and the setting word of that channel is used to recall the previous value. It sets a Bit to indicate that a change occurred. The following entry is an auto-setting type 0x0D entry. It takes the first channel's (new) value and performs a setting to the second channel, only if the Bit set by the first entry is set. The second channel tracks the first.

To illustrate this with an example, the following two entries were used for a test of this approach in node0590.

```
90:480014A0 2400 003F 4801 03E0
90:480014A8 0000 0000 0083 0001

90:480014B0 0D81 8083 4801 03E0
90:480014B8 0590 003D 0000 0001
```

For this test, channel 0x003E was monitored as the first channel. The reading field of channel 0x003F is then set to the difference seen in the memory word at 0x480103E0, which is actually the reading field of channel 0x003E. Bit 0x0083 is set whenever this difference is nonzero; otherwise, it is cleared.

The second RDATA entry is an auto-setting entry type 0x0D. It is executed only when Bit 0x0083 is set; otherwise, it does nothing. With these two RDATA entries running, a change made to channel 0x003E is replicated in channel 0x003D, soon after the change is made. The sign bit of the Bit# (0x8083) is set, so that the auto-setting only works if the Bit is set.

Note that in the auto-setting entry, it is usually a good idea to set the sign bit of the second byte, which carries the listype#, in order to prevent the generic destination table# interpretation of this byte. (Valid system table#s only range from 0x00—0x1F.)

For doing the same thing to two channels, both entry types need a memory delta, which in this case, for consecutive channels being monitored as well as consecutive channels being set, might be 0x0010. For the 0x24 entry type, the delta is a long word, where the 0000 0000 is above. For the 0x0D entry type, it is a short, where the 0000 is above. Two consecutive dummy channels would be needed for this case, and two consecutive Bit#s would be used.