Week 22

Synchronization should be done for two subjects in converter systems:

1.Data and variables which controllers have gathered and calculated.

2.Switching signal phase of the modular converters referenced to the supervisory controller (phase synchronization).

In the first part, data is being transferred from master to slaves in the periodic manner after computation of the control loops. In the supervisory controller, the control loop runs based on the switching frequency of the converter (when analog signals have been converted to digital at start of PWM signal). The result is being transferred from the control subsystem (C28) to the master subsystem (m3). This processor is responsible to form a packet frame and send the data to the slave controllers.

The Slave controller has the same process to send/receive data and analyzing them. The difference is that the data flow from master to slave is one to many type (broadcasting), which means sending only one packet is enough to synchronize all the slave controllers, but from slaves to master is not the same and the channel is being shared between controllers. To divide the channel between slave controllers, two principle solutions are available:

1- Time Division Multiple Access: The receive channel can be divided based on time division multiple access (TDMA) technique. Whenever the master send request to slave to get updated data, a scheduler will start counting and based on the module number (assigned automatically or by installer), each module sends out its status and variables. The biggest problem in this method is the error in crystal oscillators of slave controllers and the probability of data frame overlapping. Therefore, a dead time must be allowed between data packets to avoid this problem.

2- Carrier Sense Multiple Access: In this mode, all the slaves listen to the master controller and when the master requested for data from a slave controller, then it starts to send data to the master controller. The other slaves can listen to the transmitting slave and when it ends the process, the next slave can start sending data immediately.