Using Advanced Triggering Methods to Reduce ATE Test Times

INTRODUCTION

With advances in technology, Automatic Test Equipment (ATE) systems are becoming more widely used across a range of industries including manufacturing, avionics, aerospace, military and defense. ATE systems are efficient and can be incredibly useful, allowing quick and accurate testing that communicates across a set of devices. However, it can become a complicated task to properly setup an ATE system to achieve the user's desired outcomes. Developing an ATE system takes time, and there can be different approaches to achieve the same goal. Misusing an approach can cause the system to become less efficient, contributing to more time spent or more resources used to complete the task.

When multiple test instruments are required to work in conjunction with one another to acquire a measurement or sequence of measurements, the ATE architecture becomes even more complex and difficult to design. For example, a test may require triggering a digital multi-meter (DMM) to take a measurement when a relay is closed, triggering a switch module to close a relay when a signal is successfully generated, or triggering a DIO to output a signal after the DMM takes a measurement. Engineers can choose from different triggering methods to find the best approach to achieve their goals. Each triggering method has its own level of efficiency as it relates to execution time and implementation simplicity. Some more deterministic triggering methods can decrease program overhead and latency by allowing devices to communicate directly, outside of the program.

There are various methods for LXI instruments to communicate with one another in a test sequence. An engineer can pace the sequence of events in an ATE system through his or her application code (software triggering). This is very simple to implement in code, however, there is a high degree of dependency on the host to manage the test sequence.

Alternatively, the LXI specification provides various means through extended functions for handshaking between instruments to reduce the amount of host intervention required thereby minimizing any latency attributed to host-instrument communication. VTI has implemented LXI extended functions into its LXI instruments providing a great deal of flexibility in system design options.

LAN events can include trigger messages that are sent over the Ethernet wire from one device to another indicating events such as when an instrument has completed an operation. A second LXI device can be programmed to interpret that message as a trigger event for it to take action.

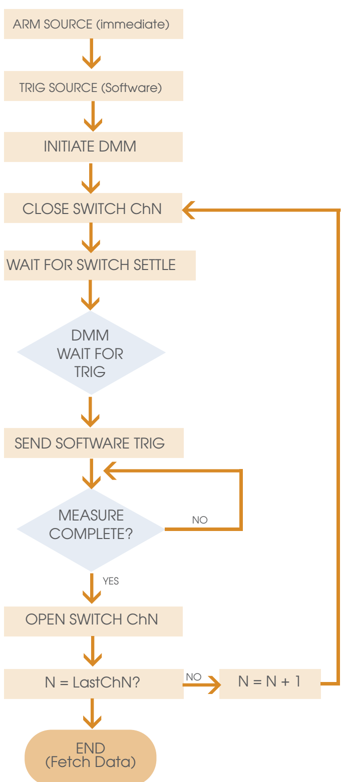
Another approach is to use the LXI wired trigger bus, which is an 8-wire hardware trigger bus analogous to the PXI or VXI backplane trigger bus. The LXI wired trigger bus can be used as a vehicle to send physical signals between one device to others in order to pace the test sequence.

This white paper will discuss a few of the popular methods for programming test sequences and how a test developer can take advantage of VTI's EX1200 series, which implements LXI extended functions, to make test systems operate more efficiently. The EX1200 series (pictured below) is a family of modular, smart, switch and I/O instruments, with embedded scanning capability and an internal analog backplane that reduces external cabling requirements.

EX1206A; WITH 6.5 DIGIT DMM AND SWITCH MODULES

TRIGGERING METHOD SETUP

Each triggering method—software, LAN event and LXI wired trigger bus—can be easily setup and efficiently used by LXI instruments to perform accurate and timely measurements in an ATE system.



SOFTWARE TRIGGERING

In comparison to other triggering methods, the software approach is the easiest but slowest and least efficient method to trigger an action among LXI instruments. The user simply sends a command through a program to trigger an action and repeats these actions in software until all measurements are recorded. The flow chart to the left describes a switch/measure sequence that is paced by the application code where a single measurement device is used to make multiple measurements through a multi-channel switch. Due to the number of commands given by the program, and particularly the need to add a software delay, latency can become an issue.

Another way to illustrate the sequence is shown below, using a DMM as the measure device with an integration time of 1.67 ms, and an electromechanical switch module settling time of 5 ms.

Send Command Close ChN	Variable
Wait for Switch Settle	5 ms
Send DMM Measure Command	Variable
Poll on DMM Measure Complete (or wait)	1.67 ms
Send Command Open ChN	Variable

Removing all variables, and considering an example in which 400 measurements are made through the switch, the best case scenario for execution time is (1.67 ms + 5 ms) = 6.67 ms per measurement. At the rate of 150 readings per second, the test will complete in approximately 2.7 seconds. However, due to the variables in time that is consumed in sending commands across the host-instrument communications bus as well as the variability in wait statements, the efficiency in execution of a software-paced program is much less than the theoretical maximum.

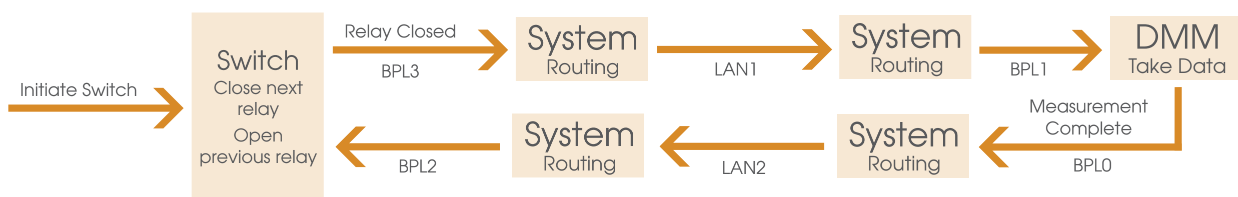
LAN EVENT TRIGGERING

Of the triggering methods available, the LAN event approach supported by the EX1200 series stands out as one of the easiest and most efficient ways to perform triggering. Since all LXI instruments are connected and communicate using an LAN cable, LAN events can be generated without extra effort spent on cabling. Triggering can be performed automatically after a condition is met. To ensure that LAN event triggering is achieving its highest level of efficiency, the test network should be isolated from any non-essential traffic. The following example will describe the application code that is required to create the handshaking mechanism using LAN events between an EX1200 DMM in one instrument mainframe and an EX1200 multi-channel switch housed in another mainframe.

```
//Dmm trigger setup
dmm.Trigger.MultiPoint.MeasurementComplete = VTEXDmmMeasCompleteDestEnum.
VTEXDmmMeasCompleteDestBPL0;
dmm.Trigger.Source = VTEXDmmTriggerSourceEnum.VTEXDmmTriggerSourceBPL1;

//Switch trigger setup
sw.Scan.ConfigureTrigger(0, VTEXSwitchTriggerInputEnum.VTEXSwitchTriggerInputBPL2,
VTEXSwitchAdvancedOutputEnum.VTEXSwitchAdvancedOutputBPL3);

//Route the signal from LAN2 to BPL2
swSystem.InstrumentSpecific.Route.Destinations.Item["BPL2"].SourcesList = "LAN2";
swSystem.InstrumentSpecific.Route.Destinations.Item["BPL2"].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Route the signal from BPL3 to LAN1
swSystem.InstrumentSpecific.Route.Destinations.Item["LAN1"].SourcesList = "BPL3";
swSystem.InstrumentSpecific.Route.Destinations.Item["LAN1"].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Route the signal from BPL0 to LAN2
dmmSystem.InstrumentSpecific.Route.Destinations.Item["LAN2"].SourcesList = "BPL0";
dmmSystem.InstrumentSpecific.Route.Destinations.Item["LAN2"].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;
```

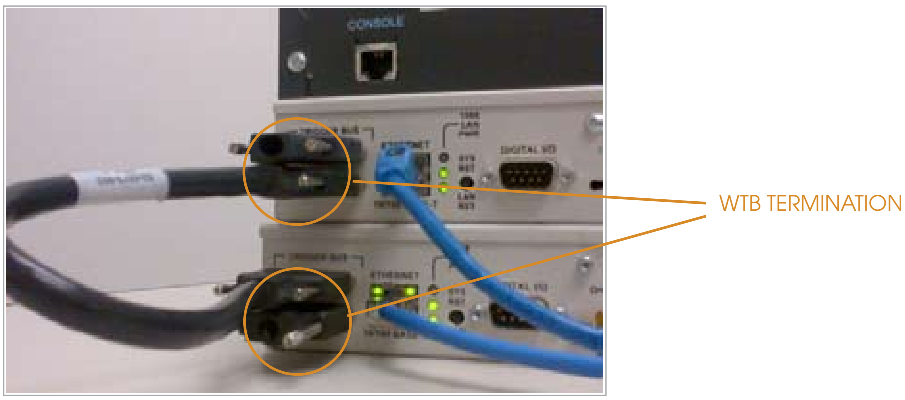
```
//Route the signal from LAN1 to BPL1
dmmSystem.InstrumentSpecific.Route.Destinations.Item["BPL1"].SourcesList =
"LAN1";
dmmSystem.InstrumentSpecific.Route.Destinations.Item["BPL1"].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Initiate switch and dmm
dmm.Measurement.Initiate();
sw.Scan.Initiate();

//Fetch all the data from the FIFO in 1000ms
dmm.Measurement.FetchMultiPointTimeStamped(1000, ref reading, ref second, ref
fraction);
```

The LAN event triggering example above is less intuitive than using a hardware trigger bus, or simple software pacing; however, it does reduce the bus traffic and requires no external hardware. In this example, when the EX1200 DMM has completed taking a measurement, a backplane trigger is generated, and the EX1200 processor sends a LAN message across the Ethernet cable. Any LXI device can be programmed to listen and respond to this message. In this example, an EX1200 switch system is set to close the next channel in sequence and open the previous channel when the DMM has issued a LAN message indicating a measurement is complete. The LAN message is converted to a backplane signal which the switch module interprets as the trigger to close the next switch in sequence and open the previous channel. This cycle continues until all measurements have been made. Reference the figure below:



LXI WIRED TRIGGER BUS TRIGGERING

While LAN event triggering can achieve accuracy in milliseconds, the LXI trigger bus method can achieve accuracy in nanoseconds—making it the most precise way to communicate between instruments. In order to perform triggering using the EX1200 wired trigger bus, the user must physically connect multiple chassis with an LXI wired trigger bus cable and terminating connectors. (See figure below.)



Also, the LXI domains on all the chassis must be the same. By using the EX1200 VTEXSystem driver, the user can easily set the LXI domains. In the following example, we will modify the LAN event example program so that it uses LXI trigger bus to trigger all of the actions instead of LAN events.

```
//Setup LXI domain of each EX1200 system
dmmSystem.LXIDomain = 0;
swSystem.LXIDomain = 0;

//Dmm trigger setup
dmm.Trigger.MultiPoint.MeasurementComplete = VTEXDmmMeasCompleteDestEnum.
VTEXDmmMeasCompleteDestBPL0;
dmm.Trigger.Source = VTEXDmmTriggerSourceEnum.VTEXDmmTriggerSourceBPL1;

//Switch trigger setup
sw.Scan.ConfigureTrigger(0, VTEXSwitchTriggerInputEnum.VTEXSwitchTriggerInputBPL2,
VTEXSwitchAdvancedOutputEnum.VTEXSwitchAdvancedOutputBPL3);
```

```
//Route the signal from LAN2 to BPL2
swSystem.InstrumentSpecific.Route.Destinations.Item[“BPL2”].SourcesList =
“LXI2”;
swSystem.InstrumentSpecific.Route.Destinations.Item[“BPL2”].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Route the signal from BPL3 to LAN1
swSystem.InstrumentSpecific.Route.Destinations.Item[“LXI1”].SourcesList =
“BPL3”;
swSystem.InstrumentSpecific.Route.Destinations.Item[“LXI1”].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Route the signal from BPL0 to LAN2
dmmSystem.InstrumentSpecific.Route.Destinations.Item[“LXI2”].SourcesList =
“BPL0”;
dmmSystem.InstrumentSpecific.Route.Destinations.Item[“LXI2”].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Route the signal from LAN1 to BPL1
dmmSystem.InstrumentSpecific.Route.Destinations.Item[“BPL1”].SourcesList =
“LXI1”;
dmmSystem.InstrumentSpecific.Route.Destinations.Item[“BPL1”].DriveMode =
VTEXSystemEventDriveModeEnum.VTEXSystemEventDriveModeDriven;

//Initiate switch and dmm
dmm.Measurement.Initiate();
sw.Scan.Initiate();

//Fetch all the data from the FIFO in 1000ms
dmm.Measurement.FetchMultiPointTimeStamped(1000, ref reading, ref second, ref
fraction);
```

SUMMARY

Pacing test sequences when using LXI instruments in an ATE system can be straightforward and analogous to other familiar test platforms such as GPIB, VXI and PXI while adding the flexibility to trigger devices over Ethernet. Different methods are available to choose the best approach to reach the user's specific goals, and smart instrumentation, such as VTI Instruments' EX1200 series can be used to increase accuracy and efficiency. If time is not an important factor, software triggering methods can be easily used. However, if accuracy and time are critical to meet the user's ATE system requirements, using extended functions is recommended. LAN event triggering can be efficient for distributed applications, and LXI trigger bus offers backplane-like performance. These LXI instrument extended functions can be used to overcome the impact of program overhead and reduce host intervention in an ATE system in order to create more efficient test programs.