

Feature Overview

DT10 MULTI-WAVE SCOPE

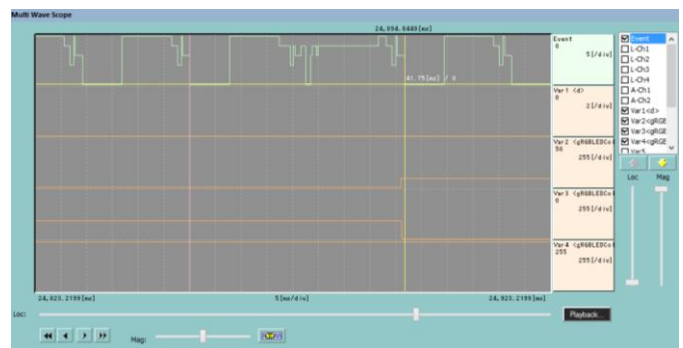
Feature Introduction

A Logic Analyzer can be a very useful tool for troubleshooting embedded software when a traditional debugger is not suitable for the job. Check out this article which illustrates step-by-step the setup required for such a use-case:

[Troubleshooting-real-time-software-issues-using-a-logic-analyzer](#)

But, a logic analyzer is a tool traditionally used by hardware engineers to debug digital circuits. So as evident from the referenced article above, the embedded software engineer has a lot of setup work to do in order to use the analyzer, because the tool is not purpose-built for troubleshooting software.

DT10's **Multi-Wave Scope + Analog Box** have been created precisely to provide capabilities for troubleshooting embedded software with features similar to what you would see in a logic analyzer or oscilloscope. It captures and displays characteristics of the running software with precision time graphs updated on a scope in real-time.



Just some of the features of DT10 that automate much of the work for the embedded engineer include:

- ✓ Automatic Insertion of Test Point macros into the source code (no tedious coding or scripting is necessary)
- ✓ Integrates seamlessly with your existing host communication mechanism (GPIO, UART, Ethernet, etc.)
- ✓ Capture and display a trace of Kernel Events (e.g. task scheduler, Interrupt routines, etc.)
- ✓ Capture and display up to 10 local or global variables simultaneously
- ✓ Supports up to **32 days** of continuous execution trace (for that difficult-to-reproduce bug)
- ✓ Offline Playback feature, to review the transition of variables as occurred during live execution
- ✓ A 2-way point-and-click link between the Scope data points and the real-time Execution Path Trace of the code
- ✓ Auto-Generates a suite of Analysis Reports to assist with Troubleshooting runtime issues

The DT10 Analog Box

In addition to capturing characteristics of the source code in real-time, the DT10 Analog Box component can **capture Logic and Analog signals from your target device**. Data from both hardware and software is synchronized by the Dynamic Tracer and displayed together on the same time-graph on the Multi-Wave Scope in real time.



This combination of DT10 Dynamic Tracer with Analog Box provides the Embedded Engineer with significant advantages for pin-pointing the root cause of difficult-to-reproduce software bugs.

This really is a Power-Tool for Embedded Software troubleshooting!

Feature Demonstration

A Proximity Detector device uses a sensor to calculate the distance of an approaching object. The software running on the device is used to process the distance calculation and uses the result to drive either one, two or three LED signals to an ON state, depending on the proximity of the object.

In this example DT10 is used to monitor various characteristics of the embedded software execution, and correlate it with the behavior of the LED's at the same time, so that at any instance we can pinpoint the state of the software for a given event in the hardware. Likewise, with DT10 we can monitor characteristics of the executing software and pinpoint the state of logic or analog signals from the hardware at the same instance.



DT10 Dynamic Tracer Setup

1. Test Point Insertion

Using the DT10 software Test Point macros are automatically inserted into the source code, to capture and output characteristics of the software as it executes on the target device; including precise timing information, coverage information, event and variable monitoring data, etc.

2. Configuring Test Point Output

In this example the Test Points are configured to send trace data from the executing software out to the Dynamic Tracer unit over GPIO. The GPIO pins on the target device are connected to the DT10 Dynamic Tracer's GPIO port.

3. Capturing LED Digital Signals

The DT10 Analog Box is also connected to the target to capture logic signals from the three LED's on the device. This allows the HIGH and LOW transitions to be traced and monitored with precision as the LED's on the device turn on or off in response to some object in close proximity. The Dynamic tracer will automatically synchronize both sets of captured data using a single clock source.



This completes the setup.

The target device is left to run for as long as necessary to reproduce any defective behavior. All data captured through the DT10 Dynamic Tracer is synchronized and sent to the DT10 Software on a connected host machine. The data can be monitored on the Multi-Wave Scope in real-time and also played back at a later time.

Once the test is complete the DT10 Analysis Engine will scan the data and automatically generate a suite of analysis reports to help troubleshoot the embedded software.

This report generation step is where DT10 effectively does a lot of the “heavy lifting” for the engineer!



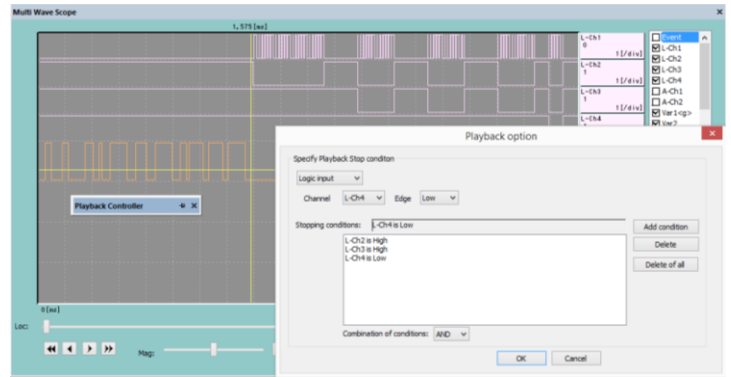
The Analysis

Part 1: Pin-pointing the Software Execution state, based on the LED logic signal pattern

The Multi-Wave Scope can be used to examine the logic or analog signal trace for patterns suggestive of a problem with the sensors LED operation, and from there pinpoint the exact instance in the embedded software execution when the pattern occurred. Once that point is identified a synchronized suite of analysis reports generated by DT10 can be used to examine various characteristics and viewpoints of the running software.

To identify the pattern of interest in the LED's logic signals the user can zoom-out in the scope and manually scroll through the trace execution, zooming in again to examine the display in more detail at any point of interest.

Alternatively, with the DT10 Playback feature and Auto-Stop conditions, the user can effectively replay the execution in the scope and DT10 will automatically pause the playback each time the Auto-Stop conditions are met (e.g. LED1 On, LED2 On, LED3 Off)



Now that we have identified the pattern of interest we can zoom-in on the scope display, and by clicking on any point on the scope DT10 can synchronize that particular instance of execution with the state of the software at the same point in time, through a suite of software analysis reports that have been generated by DT10.

The Screenshot below illustrates **just some** of the views generated by DT10 to assist with software troubleshooting:

No.	Source	Function	Memory	Value of a variable	Elapsed time(ns)	Difference(ns)
10410 (10762)	Task_Hardware	getDistance		10,335,166,950	69,500	
10411 (10763)	Task_LED.c	Task_LED		10,335,396,110	72,099	
10412 (10765)	Task_LED.c	Task_LED		10,335,396,110	157,670	
10413 (10766)	Task_LED.c	Task_LED		10,335,264,100	89,999	
10414 (10767)	Task_Hardware	setLEDState		10,335,537,440	71,340	
10415 (10768)	Task_Hardware	setLEDState		10,335,607,100	69,660	

Now the engineer can, for example, examine the transition of code variables before, during and after the pattern occurred, or can step through the execution path and see which lines of code were executed, what branches were executed, etc. prior to a particular behavior occurring in the hardware.

There is a collection of other very useful and insightful reports and charts available to the engineer to assist in root-cause analysis of runtime software issues.



Part 2: Pin-pointing the LED state, based on Characteristics of the Executing Software

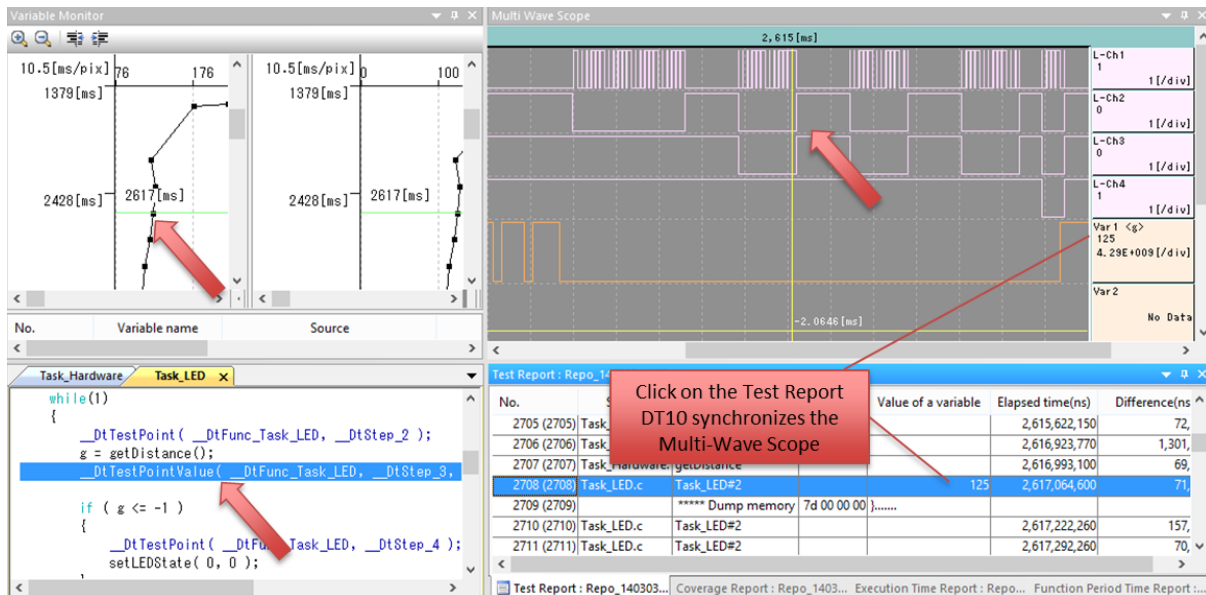
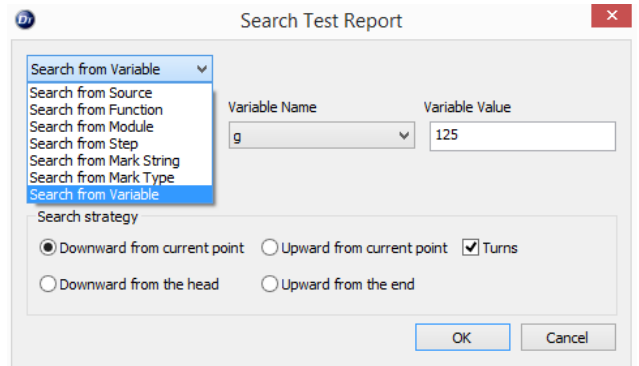
In the previous example we started in the Multi-Wave Scope by identifying a pattern of the LED logic signal trace and using DT10 to correlate this with characteristics of the running software at the same instance in time.

You can also use DT10 very effectively in the opposite direction, to search for specific characteristics of the executing software (e.g. the state of a variable, the occurrence of a software event, line # of executing code) and correlate this with the LED signal state at the same time. So you can see what the LED logic signals are doing when the software is in a particular state.

In the software for this example, the variable 'g' stores the calculated distance between the sensor and an approaching object. Based on the value of this variable, the software will drive 1, 2 or 3 of the device LEDs to an ON state.

We can, for example, search through the DT10 Test Report to find any instances where the variable took on a specific value.

Then by simply clicking on the Test Report at that particular step, DT10 will update the other active windows so that each display is focused at the same recorded execution time. From the Multi-Wave Scope then, we can zoom-in and view the state of logic and analog signals representing the state of the LED's at that same point in time.



Summary

The DT10 Multi-Wave Scope and Analog Box provide unique capabilities to troubleshoot embedded software. They offer the software engineer a view of the executing code with features similar to tools like logic analyzers and oscilloscopes, but purpose-built for embedded-software, which makes it easy to setup and configure for this purpose.

This solution from DT10 is used to track down and troubleshoot some of the most difficult-to-reproduce bugs that are often encountered over the course of an embedded development project.

