

J-LINK Frequently Asked Questions (FAQ)

Copyright Debug Innovations 2004. Issue 2, 14 Feb 2004.

General Questions

What is J-LINK and how is it different to JTAG?

J-LINK is a system that allows a JTAG interface to be implemented with just one wire. JTAG is a 5 pin test and debug interface that is defined by the IEEE 1149.1 standard. It is used to test chips and boards with a technique called "Boundary Scan" and to perform higher level operations on certain devices such as programming EEPROMs / PLDs and debugging software running on embedded processors. J-LINK simply enables anything you can do with JTAG to be done with only one wire or device pin. However the most common application of J-LINK is for providing debug access to processors in low pin-count devices.

Background Information – JTAG debug

Modern microprocessors generally support some form of debug connection. There are 2 types in common use: Asynchronous Serial connections with a 'debug monitor' program running on the target CPU, or some form of hardware debug system designed into the CPU core. Debug monitor systems only require one or two device pins but they are slow and the monitor takes up program space. That is to say they are resource hungry. In addition they can interfere with the debug process as the CPU is always running. As a result of these issues, there is a trend towards hardware debug systems. Hardware debug systems are much faster and don't require any target resources.

To meet testability requirements, most modern devices support the industry standard IEEE 1149.1 JTAG boundary scan system. This consists of a 5 pin interface to a piece of test logic known as a TAP controller. The TAP controller is connected to one or more 'scan chains' which are like giant shift registers. Each scan chain cell is connected to a carefully chosen part of the device that needs to be tested. Through the 5 pin interface, a piece of test equipment can set and read the state of each bit in the scan chains and hence test the device thoroughly.

In order to avoid adding extra pins for hardware debug, designers have 'piggy-backed' the debug system on to the JTAG test system by adding new instructions that the TAP controller can recognise as debug information. In this way the device can be tested in production then the same pins can be reused for debugging purposes. Perhaps the best example of this technique can be found in the ubiquitous ARM family of CPUs. All of the ARM7, ARM9, ARM10, ARM11 and Xscale CPU cores use this method for debug.

However, five pins is too many for low pin count designs and the designers of these devices have to choose between providing a serial debug monitor which is resource hungry (hence extra cost) or providing no debug capability at all which is unacceptable to users.

The J-LINK system allows designers to test and debug devices with standard JTAG interfaces using only one device pin. It consists of a small piece of on-chip logic known as the 'interface circuit' and a 'converter box' which interfaces the J-LINK to standard JTAG testers or in-circuit emulators (ICEs). The converter box supports both JTAG and J-LINK standards using the same connector so that the same tools can be used with multiple different device types. In principle any type of device with a JTAG interface will operate with J-LINK including all types of CPUs and DSPs. The system also supports serial chaining of TAP controllers as used with multi-processor or processor + DSP designs.

Is J-LINK an IEEE standard?

Not yet. J-LINK is backward compatible with the IEEE 1149.1 standard so could become an extension to the existing standard.

Which processors can I use J-LINK with?

In theory you can use J-LINK with the vast majority of processors (in fact, any device that has a JTAG interface and a free running clock) because J-LINK faithfully transfers all the JTAG signal information across the link. However commercial considerations will affect the viability of using J-LINK with a specific device e.g. the cost and availability of a suitable ICE or converter box. Also not all devices are suitable for technical reasons e.g. devices with no clock source or a maximum clock frequency below 16MHz. In general J-LINK is suitable for all modern processors.

Is the interface circuit the same for all processors?

Yes, but the board connectors will be different because they will match the existing standard for each different processor enabling ICEs to support both JTAG and J-LINK standards with a single connector.

Is the converter box the same for all processors?

No. The basic circuit is the same but the physical characteristics match the ICE e.g. the connectors are all different. Please ensure you order the correct converter box for the ICE you want to use.

The flyer specifies a maximum J-LINK frequency of 50MHz. Is this number equivalent to a 50MHz JTAG clock rate in a regular 5 pin JTAG system?

No. It is the data rate over the single pin / wire. It is equivalent to one sixth of the standard data rate as J-LINK require 6 clocks to transfer one TCK's worth of data so, at 50MHz, you will get the equivalent of over 8MHz equivalent performance.

In contrast other single pin solutions require more than 6 clocks per TCK so their maximum possible data rates are lower than J-LINK. In addition they do not operate at the same frequency (J-LINK is over 20 times faster than the only other known solution).

What performance can I expect?

In complete debug systems, performance is dominated by the ICE and software (debugger) performance. Unlike other single wire systems J-LINK is so fast that a typical user would not be able to tell the difference between a J-LINK debug session and a JTAG debug session. As an example, Multi-ICE's maximum TCK frequency is 10MHz and it uses less than 50% of the bandwidth available so anything over a 5MHz equivalent will give full performance debug.

In test applications where the JTAG is run at very high TCK clock rates e.g. 100MHz, J-LINK will increase the test time. There is a trade off between the increased cost of extra test time and the reduction in cost of requiring 4 less pins or increased selling price of a device with the same number of pins and extra functionality.

Why is J-LINK so fast?

Unlike previous systems of this sort, J-LINK uses special techniques to achieve extremely high performance. In most cases a debugging session using J-LINK is indistinguishable from the same session using regular 5 pin JTAG.

The key to J-LINK's performance advantage is the low number of clocks required to transfer one TCK's worth of JTAG data. For each TCK sent by the ICE or tester, J-LINK uses only 6 clocks to transfer both the outgoing and incoming JTAG data, the clocking information, link status and flow control information. Furthermore, in most cases, the J-LINK interface circuit clock can be connected directly to the CPU clock. The interface circuit clock frequency determines the data rate on the J-LINK.

As an example, suppose you have a 40MHz CPU clock, then connecting the CPU clock directly to the interface circuit will result in a 40MHz J-LINK data rate. If you have a 70MHz CPU clock then you would run the J-LINK from a divide-by-2 circuit resulting in a 35MHz J-LINK frequency. This is because the maximum data rate for the J-LINK converter box is 50MHz.

	TCK rate	Num pins	Gate Count	Link Integrity	User Proof
UART	0.3 MHz	1-2	High	Unmonitored	No
Manchester Enc.	4 MHz	1	Low	Unmonitored	No
J-LINK	8 MHz	1	Low	Monitored	Yes

The table shows two alternative approaches to this problem and compares them with J-LINK. It is clear that J-LINK not only delivers the best performance but has other clear benefits. The information in the table comes from an analysis of 3 approaches for the same system – a 48MHz CPU requiring debug through a single pin:

With J-LINK, using a 48MHz CPU, the J-LINK data rate would be 48MHz and the maximum TCK rate would be 8MHz because there are 6 bits required to transfer the data for one TCK. Comparing this with other systems shows just how efficient J-LINK is:

System 1: Using a UART. This is the method used with a debug monitor but it could also be used to serialize a JTAG data stream. In this case the UART at the target end has to 'measure' the width of the start bit so that it can sample the data bits with correct timing. This measuring process results in a data rate that is typically 16 times slower than the maximum device frequency (which we shall assume is the same as the CPU clock). With a 48MHz clock, this results in a maximum data rate of 3MHz. We have to send 4 JTAG bits and receive one bit back from the target. Each packet requires a start and stop bit. Thus we need 9 bits to send and receive one TCK's worth of data which results in a maximum TCK rate of 333kHz (24 times slower than J-LINK). In addition we need a large amount of logic (a specially designed UART) at the target end using valuable Silicon area and some way of the user selecting and knowing the correct baud rate.

System 2: Using Manchester encoding. This is considerably better than system 1 but still half as efficient as J-LINK. In this scheme each signal is encoded into a single bit as before but each bit is accompanied by a clock signal. Line speed is limited by signal integrity and shared line timing considerations to around 50MHz and 10-12 clocks are required to transfer the data. That gives a maximum TCK rate of 4-5MHz. One particular disadvantage of Manchester encoding is that the data rate is determined by the sending end, and since data is transferred in both directions for JTAG, that means that both ends determine the data rate. This forces the ICE to have a programmable data rate and the user to understand the data rate requirements of a particular ASIC. This is not very user-friendly and will increase support costs. In contrast J-LINK adapts automatically to the data rate that the ASIC designer has chosen so there is no possibility of the user 'getting the settings wrong'.

Neither system 1 nor system 2 has any way of monitoring the link integrity. Indeed, since they are both asynchronous systems, disconnecting the link would not result in any sort of error until the user tried to establish a debug session. J-LINK continuously monitors the line status and displays it on an LED on the converter box. This simple integrity monitoring helps to reduce debug time and support costs by catching gross errors such as loss of ASIC power.

How long a wire can I use?

That depends on what frequency the J-LINK is operating at. The chip designer will choose the best frequency to operate J-LINK at depending on the requirements for that device. Higher frequencies will result in better performance for systems that can keep up with J-LINK whereas lower frequencies result in more user friendly characteristics such as the ability to use long wires. As a general recommendation, for debugging purposes we suggest the J-LINK should not be run at more than 25MHz which allows the use of about 1 metre of ribbon cable or longer lengths of coax.

Does J-LINK support multi-processor or CPU + DSP systems?

Yes, provided all the cores are on the same chip. You can serially chain devices on a chip in the normal way (using the 5 JTAG wires) then connect the resulting interface to the J-LINK interface circuit. This is the normal way of supporting multi-processor systems.

You cannot take multiple chips with J-LINK interfaces and connect them together into a single J-LINK interface, nor can you mix JTAG and J-LINK interfaces. You must decode the J-LINK using a converter box before connecting any extra JTAG devices.

Can I use ARM's Coresight system with J-LINK?

Yes. Simply run Coresight in 5 pin mode (normal JTAG) and connect a J-LINK interface circuit to it in the normal way. You will then have a J-LINK enabled device that is compatible with a wide range of ICs and has better performance than using Coresight's single pin mode.

How does J-LINK compare with Coresight's single pin mode?

ARM's Coresight system uses Manchester encoding so suffers from all the drawbacks of this technique (see earlier performance discussion). J-LINK is expected to be considerably faster but as Coresight is not yet available, this cannot be tested.

Coresight will initially only be available for ARM9 & ARM11 cores. It is not available for non-ARM cores. There is no indication that it will be offered as IP to ICE manufacturers and ARM are only adding single pin support to RealView ICE. ARM have not announced any plans to make converter boxes which means any chips that use Coresight single pin mode will not be supportable by the majority of ICs.

Coresight is a huge IP block (22k gates for ARM9, 100k for ARM11) compared to less than 1k gates for J-LINK.

In general, for low cost, low pin count devices, you can save on gates by using J-LINK instead of Coresight. For high pin count devices, implement trace using Coresight and save pins on JTAG by adding a J-LINK block to drive the JTAG lines. This should give improved performance over Coresight's single pin mode.

Product selection matrix

CPU type	Small no. pins	Large no. pins	Gate count	Comments
ARM7	J-LINK	J-LINK	0.1k	Coresight not available for ARM7
ARM9	J-LINK	CS + J-LINK	22k+	Add trace on large pin count device. J-LINK improves performance of single pin JTAG
ARM10	J-LINK	J-LINK	0.1k	Coresight not available for ARM10
ARM11	J-LINK	CS + J-LINK	100k	Add trace on large pin count device. J-LINK improves performance of single pin JTAG
XScale	J-LINK	J-LINK	0.1k	Coresight not available for Xscale and may never be (because it's an Intel design)
Non-ARM	J-LINK	J-LINK	0.1k	Coresight does not support non-ARM CPUs

What ICEs does the converter box work with?

You will find an up to date list of supported ICEs on our website www.debuginnovations.com. Many other ICEs will work with our converter boxes if they use the standard connectors / pinouts and support all the signals. Please contact us if your ICE is not on the supported list.

Does the J-LINK Multi-ICE converter box work with RealView ICE?

It hasn't been tested but the interface is the same as Multi-ICE so it should work. Please tell us if you are successfully using an ICE that is not on our supported list so we can keep the list up to date.

Why don't you use a single pin connector for J-LINK?

In order to maintain backward compatibility (to allow an ICE to support both 5-pin and J-LINK standards through a single connector) the converter box uses the same connector as before. For example, for an ARM processor a 20 way connector is currently used. Therefore the J-LINK version is also a 20 way connector. This way there is only one connector required for both standards and a user doesn't have to be aware of what standard a board or chip uses - he simply plugs in the ICE and the J-LINK converter circuitry automatically detects the presence of a J-LINK signal and decides whether to decode it (J-LINK mode) or pass the standard pins through (JTAG mode). However system designers can choose to interface using any connector they like. For example a phone can dedicate a single pin to J-LINK on its base connector.

How can I evaluate J-LINK?

The converter boxes are not yet in production though we have functioning prototypes. Until converter boxes are in production, please contact us to arrange a demo with your systems. By the end of Q1, the converter boxes should be available to buy - you will need a test board with the interface circuit on it as well.

My CPU/ICE/connector is not supported. What are my options?

We undertake to supply adaptor boards/cables or converter boxes for all the popular cases until the ICE manufacturers add J-LINK support into their products. Please contact us to discuss the best way forward.

How much does the J-LINK system cost?

If you want to license J-LINK for use on your Silicon, please contact us for a quote. If you want to buy a converter box so you can connect to a J-LINK enabled chip, the prices are on our website www.debuginnovations.com.

Technical Questions

How do I connect up the J-LINK interface circuit?

Connect TCK, TMS, TDI, TDO and nTRST to your chip's JTAG pins.

Connect REFCLK to a free running clock source that never stops and doesn't change speed. It must have a frequency of between 16 and 50MHz (more info in the next question).

Connect JLINK to the pin of the chip you want to use.

If you are using the large version of the interface circuit, you will have an extra signal called CONNECTED. This is an output that is high when the interface circuit is connected to a converter box and can be used to enable/disable other parts of your design that are not required when not being debugged. For example you could connect ARM's DBGEN to CONNECTED.

What frequency clock should I use?

Basically the requirement is to connect the J-LINK REFCLK input to a free running clock with a frequency of 16 - 50MHz (because that's what the converter box circuit can deal with) and has a 1:1 mark:space ratio. However higher frequencies limit the maximum cable length that can be used. Any clock source is fine provided it meets the requirements above so if you have a 200MHz processor clock then you will have to divide it by at least 4 before feeding it to the J-LINK interface circuit and if you divide it by 6, make sure you first do the divide by 3 followed by the divide by 2 so that the resulting clock is symmetrical.

Another consideration is the time it takes for whatever you are driving with the JTAG lines to respond to each TCK. TCKs from the converter circuit are one REFCLK wide (20ns at 50MHz) and capture the resulting TDO half a REFCLK after the falling edge of TCK (10ns at 50MHz). It is important to give the processor enough time to process each TCK so the J-LINK clock may have to be slowed down for this reason.

This sounds a bit complex but in all cases, you just need to connect the J-LINK clock to a convenient clock source on your chip and it is easy to test that you have implemented it correctly.

Overall system performance will be limited by the rate at which the ICE can drive the converter box and the rate at which the debug software can put data into the ICE. In typical systems using a J-LINK clock rate of above 25MHz will not improve the debug performance. Therefore we recommend that unless you are building a special test system, you use a J-LINK clock rate of no more than 25MHz to allow the maximum cable length and electrical robustness.

Can I use a faster J-LINK clock than my processor clock?

Yes but beware. Some designs need to run the processor at a fixed low clock rate for power consumption or other reasons. However the J-LINK clock needs to be a minimum of 16MHz. So you could for example, run J-LINK at 16MHz then divide down to 4MHz for the CPU clock. However be careful that the rest of your design still works with faster signals presented to it. For example, if your design has a TCK synchronizer running at a low frequency and J-LINK is presenting it with much higher frequency TCK pulses, you may lose information.

Can I use a completely separate J-LINK clock to my processor clock?

Yes but beware. Because the J-LINK interface circuit generates traditional JTAG signals which are asynchronous to your processor clock, there is nothing to stop you using any convenient clock source. However the same restrictions apply as described above.

How do I use an ARM-S core? What do I do with the RTCK signal?

Synthesizable ARMs sample TCK with the processor clock to produce a retimed (delayed) version of TCK called RTCK which is returned to the ICE. The ICE does not issue the next TCK edge until it has received 'acknowledgement' of the previous edge via RTCK. Therefore synthesizable ARMs require 6 JTAG pins.

When using J-LINK with a synthesizable ARM, it is not necessary to bring out RTCK. Since the J-LINK interface circuit runs off a derivative of the processor clock it can simply be run at a frequency that is low enough to allow time for the synchronizers to process the TCKs. Even a completely asynchronous J-LINK clock will work providing enough time is allowed for the synchronizers. In the majority of cases, since synthesizable ARMs are relatively new so run at high clock frequencies, the requirement to divide down the CPU clock to under 50MHz for J-LINK will allow enough time for the synchronizers to do their job.

In short, J-LINK removes the need for RTCK at the debug interface (saving yet another pin for ARM-S devices). The converter box regenerates RTCK for the ICE but it is used for a completely different reason to that on the chip and bears no resemblance to the on-chip signal.

What I/O voltage can I use on my chip?

The converter box will work with signal voltages of between 2.0V and 3.3V. Contact us if you want to operate outside this range.

What are the I/O drive requirements from my chip?

There is a 220R pullup resistor on the J-LINK line outside the chip so the static current for a low output is 15mA. In addition there is the drive requirements of the cable (which varies with cable type, length and clock frequency). We recommend using at least a 24mA driver.

How big is the interface circuit?

There are two versions of the interface circuit:

1. The small version which is always operating (about 150 gates).
2. A low power version with an automatic power down (sleep mode) system. This detects when a converter box is plugged in and 'wakes up'. If the converter box stops sending good frames for a few seconds, the interface circuit goes to sleep. The large counter required to wait a few seconds and the extra power down and clock gating circuits make this version about 600 gates.

Both versions are supplied under the same license.

How much power does the interface circuit use?

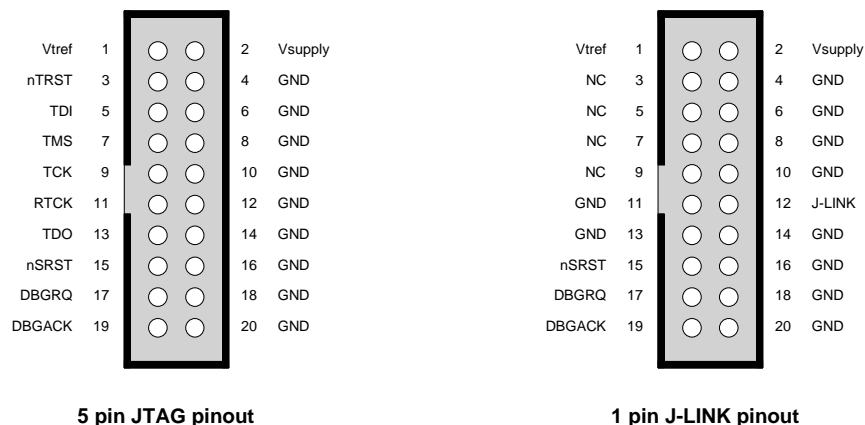
The small version (see above) takes about 3mA at 3.3V when idle. This is due to the low pulses on the J-LINK pin pulling down the 220R pullup resistor outside the chip. The low power version takes leakage current only when not connected to the converter box. Both versions take 3 or 4mA when actively sending JTAG data.

How do I test that I have implemented J-LINK correctly in my chip?

All J-LINK licensees receive test bench code to verify correct operation of the J-LINK interface circuit in a simulation environment. If this is your first J-LINK design, you will have been allocated an engineer to help you integrate the interface logic into your chip.

What is the connector pinout for an ARM system?

There are several ARM connector pinouts in use. There are a number of 14 pin IDC pinouts that are all obsolete. Do not use these for new designs. There is a 20 way IDC pinout that is used by Multi-ICE and most other major ICEs and there is a Mictor connector which combines Trace and JTAG signals in a single (smaller) connector. To accommodate J-LINK we have redefined the pinout of the 20 way and Mictor connectors so as to be backward compatible with existing usage as shown below:



This is how the connector would be on a board. Current boards use the 5 pin JTAG pinout shown. Boards containing J-LINK devices should use the 1 pin J-LINK pinout which no longer has any JTAG signals.

Mictor connector pinout – TBA.

These pinouts simply reflect the requirements of the Multi-ICE converter box. The choice of connector makes it easy to mix JTAG and J-LINK boards without having to worry which interface is being used. However for space or other reasons, it is perfectly possible to use just a single wire (or more likely a signal and ground) to interface J-LINK to a converter box but you will have to use 3.3V signal levels and will lose the functionality of the missing signals. Please contact us for advice on alternative connector schemes.

What happens when I mix different ICs/Converter Boxes with JTAG/J-LINK boards?

In the J-LINK connector pinout, all previous JTAG inputs to the chip/board are open circuit as these may be driven if a user plugs a JTAG ICE into a J-LINK board and both outputs (TDO and RTCK) are grounded. The J-LINK signal replaces an existing GND on the JTAG pinout and is surrounded by GNDs for signal integrity.

When a user plugs a converter box into a board using the 5 pin JTAG pinout, the converter box goes into a pass-through mode where it simply routes all the JTAG signals through unchanged (not even the levels change i.e. 2V signals will still be 2V signals at the ICE, they are not processed in any way). When a user plugs a converter box into a board using the J-LINK pinout, the presence of the J-LINK signal on pin 12 switches the converter into J-LINK mode and the signals are decoded by the box and presented to the ICE as 3.3V JTAG data.

What do I do with all the signals?

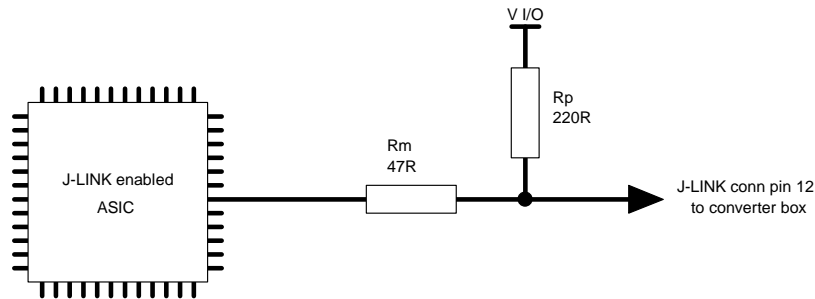
For advice on how to connect up signals for the regular 5 pin JTAG pinout, please see the relevant documentation that came with your ICE (in the case of Multi-ICE it is in the Multi-ICE User Guide).

For the J-LINK pinout, use the same information for the unchanged signals and simply connect the J-LINK chip pin to the J-LINK connector pin via the resistors described in the next question. Here are some points to bear in mind:

1. Even though J-LINK converter boxes do not use Vsupply, there is no guarantee that a future ICE with a J-LINK converter circuit built in will not, so always connect Vsupply.
2. Vtref should be connected to a representative I/O voltage e.g. if you are going to use 2V signal levels on the J-LINK line, connect Vtref to 2.0V.
3. Do not ground the NC signals – you could damage a JTAG ICE if it is accidentally plugged into a J-LINK board.
4. Most chips don't bring out DBGRQ and DBGACK and most ICs don't use them so don't worry if you can't find them, they're probably not there. In this case, ground the DBGACK connector pin and leave DBGRQ unconnected.
5. nSRST is a bi-directional system reset. You need to read up on this in the Multi-ICE User Guide because it can get a little complicated (if you already have a working design for 5 pin JTAG, then keep it because this signal hasn't changed).

What circuitry do I require outside the chip (on the board)?

You need two resistors: a pullup and a series matching resistor. It is important to get them in the right order (not the order you would expect). See diagram below:



Rm is a series matching resistor. Ideally for matching ribbon cable this should be around 100-120R but for reasons that will become apparent, it cannot be larger than about 50R. Also using 47R makes it compatible with a 50R coax cable as well. The converter box also drives J-LINK with a 47R series terminator. Its probably best not to change the value of this resistor, but if you do, **do not increase** its value.

Rp is the pullup resistor. It is required even in your final product (and your test jig) because it is the high level on the J-LINK pin that tells the interface circuit that there is no converter box present. In normal operation it is used to pull up the J-LINK line when the output goes tri-state. The low value is required because of the speed with which the line must be pulled high at maximum frequency (50MHz). Please contact us for advice if you want to use a different value resistor.

Rp is connected directly to the connector pin because it is the capacitance of the cable that is being charged (therefore it rises quicker) and when the output is low, the chip requires less drive (because Rm decreases the current requirement). This form of connection also allows us to use a 47R matching resistor because, if Rp were connected directly to the chip pin, when the converter box drives the J-LINK line low, the chip would not see a valid low ($3.3V * (47+47) / (220+47+47) = 0.98V$). This form of connection allows us to drive the chip pin to < 0.6V.

V I/O should be connected to a representative I/O voltage e.g. if your I/O voltage is 2.7V, connect it to 2.7V.

My question is not answered here. Where can I find out more?

For more information visit www.debuginnovations.com or email info@debuginnovations.com.