

## RTBx Data Logger

### Introduction to RTBx Data Logger

The RTBx data logger connects to output ports on your target hardware and collects timing data as your application runs. As the instrumentation point (Ipoint) identifiers appear on the output port, RTBx timestamps and records them, producing a trace of Ipoint/timestamp pairs.

The benefits RTBx provides include:

- Reduce on-target instrumentation;
- Support testing and data collection over weeks;
- Quick set up and easy remote management.

RTBx is currently available in two variants:



- RTBx 1210 - captures data traces using instrumentation point identifiers of up to 16 bits;
- RTBx 1220 - captures data at higher rates, in larger volumes and using instrumentation points of up to 32 bits.

### Getting the most out of the RTBx Data Logger

You could use a debugger or logic analyzer to collect timing data. However, the RTBx Data Logger is a cost-effective and easy-to-use alternative because:

- RTBx is a generic solution which you can reuse across projects, unlike some debuggers, which are restricted to a single CPU family.
- While some logic analyzers are complex to set up for both data capture and transfer, RTBx collects and exports timing data easily and quickly;
- RTBx has a massive data storage capacity which logic analyzers lack. Equipping logic analyzers to store large amounts of data can become extremely expensive.

### Maximizing the benefits of using the RTBx

- Reducing on-target instrumentation: By using RTBx you can minimize the instrumentation overhead in your application. Each instrumentation point could be implemented as one or two assembler instructions, which write a constant value to the output port. RTBx does the trace data time-stamping for you.
- Continuous data collection: You can use the RTBx to collect and store data from an entire test programme. Even when connected to sources that generate trace data at the highest rates, RTBx can still collect trace data for 19 days or more without interruption.
- You can use RTBx straight out of the box with your target system and you can control it via a GUI running on a Windows or Linux host.

## Product and data specifications

Model/mode	RTBx1210/ 16 bit	RTBx1210/ 8 bit	RTBx1220/ 32 bit	RTBx1220/ 16 bit	RTBx1220/ 8 bit
<b>Width of data sample</b>	Up to 16 bits	Up to 8 bits	Up to 32 bits	Up to 16 bits	Up to 8 bits
<b>Maximum sustained trace rate (million lpoints/second)</b>	20	30	20	40	60
<b>Minimum separation between lpoints (typical instrumentation)</b>	33ns	33ns	40ns	33ns	33ns
<b>Minimum separation between lpoints (typical instrumentation)</b>	12ns	12ns	12ns	12ns	12ns
<b>Time resolution</b>	10ns	10ns	20ns	10ns	20ns
<b>RTBx capacity</b>	25	33	33	50	67
<b>Maximum duration of trace at maximum sustained trace rate</b>	374 hours (14 days)	308 hours (12 days)	462 hours (19 days)	462 hours (19 days)	308 hours (12 days)
<b>Maximum duration of trace at typical tracing rate*</b>	66 days	75 days	135 days	133 days	151 days

\* Data for ARM9 running Dhrystone tests at 96MHz

### FAQs

*Will RTBx support my processor running at x MHz?*

RTBx will support lpoints output on an I/O port with a minimum separation of 12ns (83MHz). The answer depends upon the number of CPU cycles it takes to output successive lpoints. Obtain this number, then divide it by the CPU speed of the processor.

*Does 60M instrumentation points/second mean RTBx is limited to 60MHz CPUs?*

No. The sample rate represents the *sustained* rate at which it is possible to read from an output port. A 600MHz CPU that outputs values to the output port once every 10 cycles can be supported. Although the highest sustained rate is 60M instrumentation points per second, RTBx can handle a faster rate for short periods of time, provided that a) The minimum separation between instrumentation points is 12ns and b) the average instrumentation point rate is 60M points per second.

*What if I don't have a spare I/O port?*

RTBx can be connected to an address bus that runs up to 33MHz. A certain region of the address space needs to be dedicated to instrumentation points (for example 0x8000000 to 0x8000fff).

The instrumentation point routine simply writes a value to a specific address in that region to denote a specific lpoint. This approach will reduce the maximum trace duration of RTBx.

For more information or to purchase RTBx contact Andrew Coombes by:

Email: [acoombes@rapitasystems.com](mailto:acoombes@rapitasystems.com)

Tel: +44 (0)1904 567747



IT Centre  
York Science Park  
Heslington  
York YO10 5DG  
United Kingdom

Tel No: +44 (0)1904 567747  
Fax No: +44 (0) 1904 567719  
Email: [enquiries@rapitasystems.com](mailto:enquiries@rapitasystems.com)  
Website: [www.rapitasystems.com](http://www.rapitasystems.com)  
Registered in England & Wales 5011090