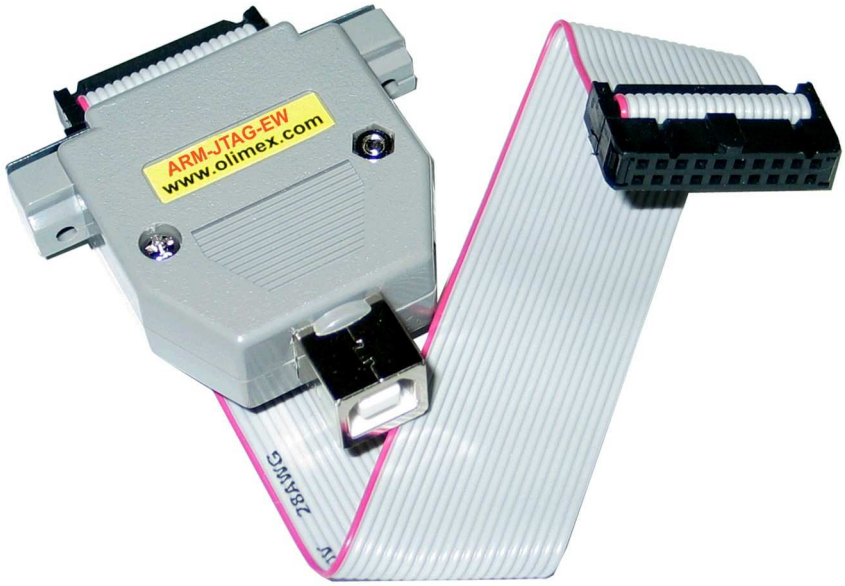


ARM-JTAG-EW

User Manual

All boards produced by Olimex are ROHS compliant

Rev.B, March 2009

1. Introduction

ARM-JTAG-EW is a JTAG probe for debugging ARM microcontrollers. It's a product by joined efforts of Olimex and IAR Systems, aiming not just to release yet another fast USB JTAG compatible with Embedded Workbench, but to make low cost JTAG with features available only in the expensive logic analyzers. For this purpose ARM-JTAG-EW offers external breakpoints in addition to ordinary software and hardware ones. User can configure ARM-JTAG-EW to stop the target program on an external trigger, defined by target power consumption, rising/falling edge of external signals, analogue signal window value.

ARM-JTAG-EW emulates the IAR Systems' J-LINK API so it works like normal J-LINK debugger, yet it adds some unique features which are available only in the very high end and expensive debugger tools on the market.

1.1. Hardware Features

- JTAG connector with ARM 2x10 pin layout for target programming and debugging
- supports ARM targets working in voltage range 2.0–5.0V DC [1]
- USB full-speed connection to the PC
- bi-color status LED
- can provide 5.0V DC power to target via pin 19 of the JTAG connector
- measurement of target current consumption
- measurement of target MCU voltage and output voltage (pins 1 and 19 of the JTAG connector)
- JTAG and SWD TCK frequency range 6kHz – 12MHz
- SWO frequency range 1kHz – 3MHz
- dimensions 50x40 mm (2x1.6") + 20 cm (8") JTAG cable

1.2. Software Features

- DLL mostly compatible with original *jlinkarm.dll* from IAR-EW[2]
- works with IAR Embedded Workbench 5.30 from IAR Systems
- supports ARM7TDMI targets (e.g. SAM7, LPC2000, STR7)
- supports Cortex M3 targets (e.g. STM32, LPC1000)
- ability to put “external” breakpoints that trigger on a user-defined event
- external event sources are the target MCU voltage, target supply voltage and target consumption current

1 ARM-JTAG-EW outputs have 3.3V levels and will work with 5V targets that have TTL-level inputs.

2 DLL compatible means that we supply our own jlinkarm.dll. The original IAR-EW DLL will not work with the ARM-JTAG-EW device because ARM-JTAG-EW and IAR J-Link use different USB protocols.

1.3. *Electrostatic Warning*

The ARM-JTAG-EW device is shipped in protective anti-static packaging. The device must not be subject to high electrostatic potentials. General practice for working with static sensitive devices should be applied when working with this device.

1.4. *ARM-JTAG-EW Usage Requirements*

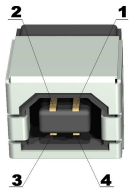
Cables: 1.8 meter USB A-B cable.

Software: EW-ARM 5.30 from IAR Systems AB

2. Hardware

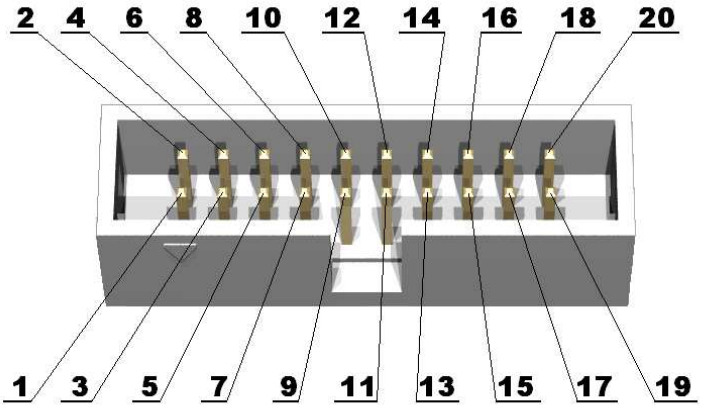
2.1. *USB Connector*

ARM-JTAG-EW has a standard USB device connector that requires a standard USB A-B cable for connecting to a PC.



2.2. *JTAG Connector*

The JTAG connector pin description is given below.



Pin №	Signal	Direction[3]	Description
1	U_{TG}	I	Target reference voltage. Target board must connect

3 Pin direction is from the side of ARM-JTAG-EW. I stands for Input (Target to ARM-JTAG-EW), and O for output (ARM-JTAG-EW to Target).

Pin №	Signal	Direction	Description
			it to the MCU supply line that drives the JTAG pins. Input has resistance 4 kΩ.
2	$U_{TG,2}$	I	Target board should connect it to the MCU supply line that drives the JTAG pins.
3	nTRST	O (open drain)	Target JTAG TAP reset. ARM-JTAG-EW has 100 Ω resistor in series with this output. Driven as open drain output.
4	GND	-	Ground.
5	TDI	O	Target JTAG Data IN. ARM-JTAG-EW has 100 Ω resistor in series with this output.
6	GND	-	Ground.
7	TMS/ SWDIO	O	Target JTAG Mode Select and Serial Wire Data Input/Output. ARM-JTAG-EW has 100 Ω resistor in series with this output.
8	GND	-	Ground.
9	TCK/ SWCLK	O	Target JTAG clock and Serial Wire Clock. ARM-JTAG-EW has 100 Ω resistor in series with this output.
10	GND	-	Ground.
11	RTCK	I	Target JTAG return clock. ARM-JTAG-EW has 100 Ω resistor in series with this input.
12	GND	-	Ground.
13	TDO/ SWO	I	Target JTAG data output and Serial Wire Output. ARM-JTAG-EW has 100 Ω resistor in series with this input.
14	GND	-	Ground.
15	nSRST	O (open drain)	Target system reset. ARM-JTAG-EW has 100 Ω resistor in series with this output. Driven as open drain output.
16	GND	-	Ground.
17	NC	-	Not connected in ARM-JTAG-EW.
18	GND	-	Ground.
19	U_{TGPWR}	O	Target power supply voltage provided by ARM-JTAG-EW. Supply is taken from USB and is switched by a MOSFET transistor. There is 4 kΩ resistor connected between this pin and ground.
20	GND	-	Ground.

2.3. *Providing Target Power Via JTAG Connector*

By default ARM-JTAG-EW connects the USB supply voltage to pin 19 of the JTAG connector via MOSFET switch. Depending on the current consumption and the used USB host and hubs the supplied voltage can vary between 4.0V and 5.25V.

WARNING: In case of a target current consumption exceeding 400 mA, including a short circuit on the target board, ARM-JTAG-EW will shutdown its power output U_{TGPWR}. Some USB hosts and hubs, however, have faster short-circuit-detectors and will shutdown the power to ARM-JTAG-EW before it has a chance to react.

Target power can be turned ON and OFF using the C-SPY macro __jlinkExecCommand() like this:

```
//Turn power ON
execUserPreload() {
  __jlinkExecCommand("SupplyPower = 1");
}
//Turn power OFF
execUserPreload() {
  __jlinkExecCommand("SupplyPower = 0");
}
```

2.4. *LED Indication*

ARM-JTAG-EW has a single dual color LED. Its states are shown below. Slow blinking is with a period of 1s, and fast blinking is with a period of 128ms.

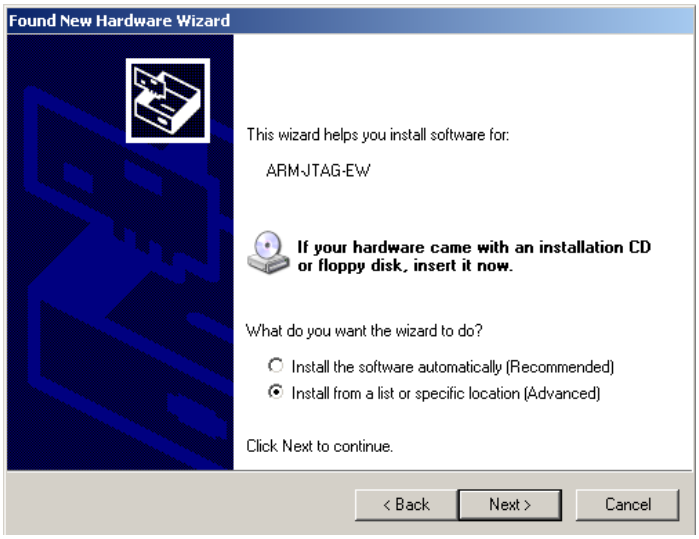
LED State	Description
OFF	ARM-JTAG-EW is not connected to USB, or it is in USB suspended state.
Slow RED blinking	No target connected.
GREEN constantly on	Target connected.
Fast RED blinking	Target error caused by target current overconsumption or MCU voltage out of bounds.
GREEN blinking	USB communication is taking place.
Slow blinking sequence GREEN→RED→OFF	ARM-JTAG-EW is in boot loader mode.

3. Software Installation

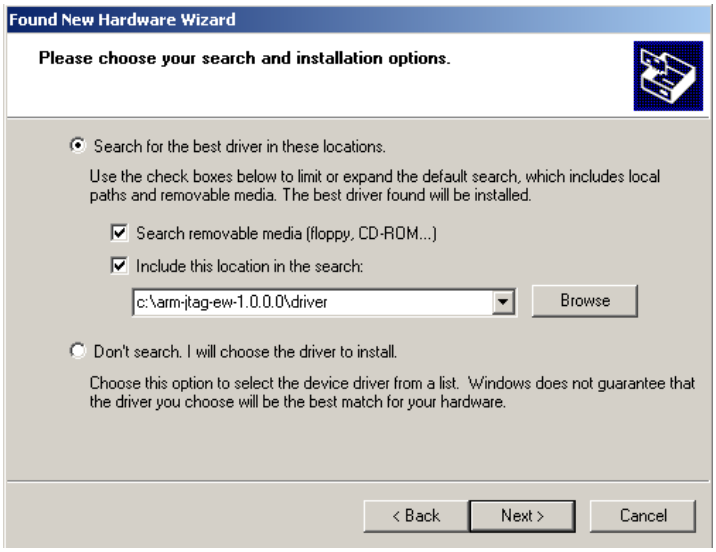
Please follow the following procedure exactly in the order it is written:

1. Install the Visual Studio 2008 SP1 Redistributable Package *vcredist_x86.exe* freely available from Microsoft: http://www.microsoft.com/downloads/details.aspx?familyid=A5C84275-3B97-4AB7-A40D-3802B2AF5FC2&displaylang=en
2. Install the .NET framework available freely from Microsoft: http://msdn2.microsoft.com/en-us/netframework/default.aspx.

3. Download the ARM-JTAG-EW software package from http://www.olimex.com/dev. Extract it to a temporary directory on your hard drive using the password found in the box along with the ARM-JTAG-EW device.
4. Plug ARM-JTAG-EW in your PC and wait Windows to ask you for drivers.



5. Next point the driver installation wizard to the *"\driver"* subdirectory of the previously extracted software package.



6. If Windows complains that the driver is not signed then click to continue and ignore the warning.

4. Using ARM-JTAG-EW with IAR-EW 5.30 ARM

Using ARM-JTAG-EW under IAR-EW is straightforward. ARM-JTAG-EW behaves just like IAR J-Link, provided that the original IAR-EW *jlinkarm.dll* is replaced by ARM-JTAG-EW's one.

4.1. *Installing ARM-JTAG-EW DLL*

Download the ARM-JTAG-EW software package from http://www.olimex.com/dev. Extract it to a temporary directory on your hard drive using the password found in the box along with the ARM-JTAG-EW device. The software package contains the file *jlinkarm.dll*. Find it and copy it to the IAR-ARM *"arm\bin"* directory, like this:

C:\> ***copy c:\arm-jtag-ew-1.0.0.0\jlinkarm.dll "c:\Program Files\IAR Systems\Embedded Workbench 5.20\ARM\bin"***

Overwrite c:\Program Files\IAR Systems\Embedded Workbench 5.20\ARM\bin\jlinkarm.dll? (Yes/No/All): yes

1 file(s) copied.

5. External event breakpoints

5.1. Overview

ARM-JTAG-EW has the ability to halt the target CPU upon detecting an external event like a current/voltage spike/drop, external input trigger, external input pulse detection.

External Breakpoints Block Diagram is given on page 9.

External events can be defined with a simple C-like expression like this (see page 9 for more information):

(3000<U_tg && U_tg<3600)

So when ARM-JTAG-EW detects that U_{TG}>3000mV AND U_{TG}<3600mV, it will shift a request to the target CPU to halt.

Multiple inputs can be mixed into one event:

(3000<U_tg && U_tg<3600) || I_tgpwr>200

WARNING: There is always some time between an actual event and the corresponding action by ARM-JTAG-EW. Due to the limited sampling rate of analogue signals, ARM-JTAG-EW cannot detect instantly changes on its inputs. Also note that shifting the JTAG sequence for requesting a target CPU halt also takes time, depending on the TCK frequency. Heavy USB traffic can also influence the response time.

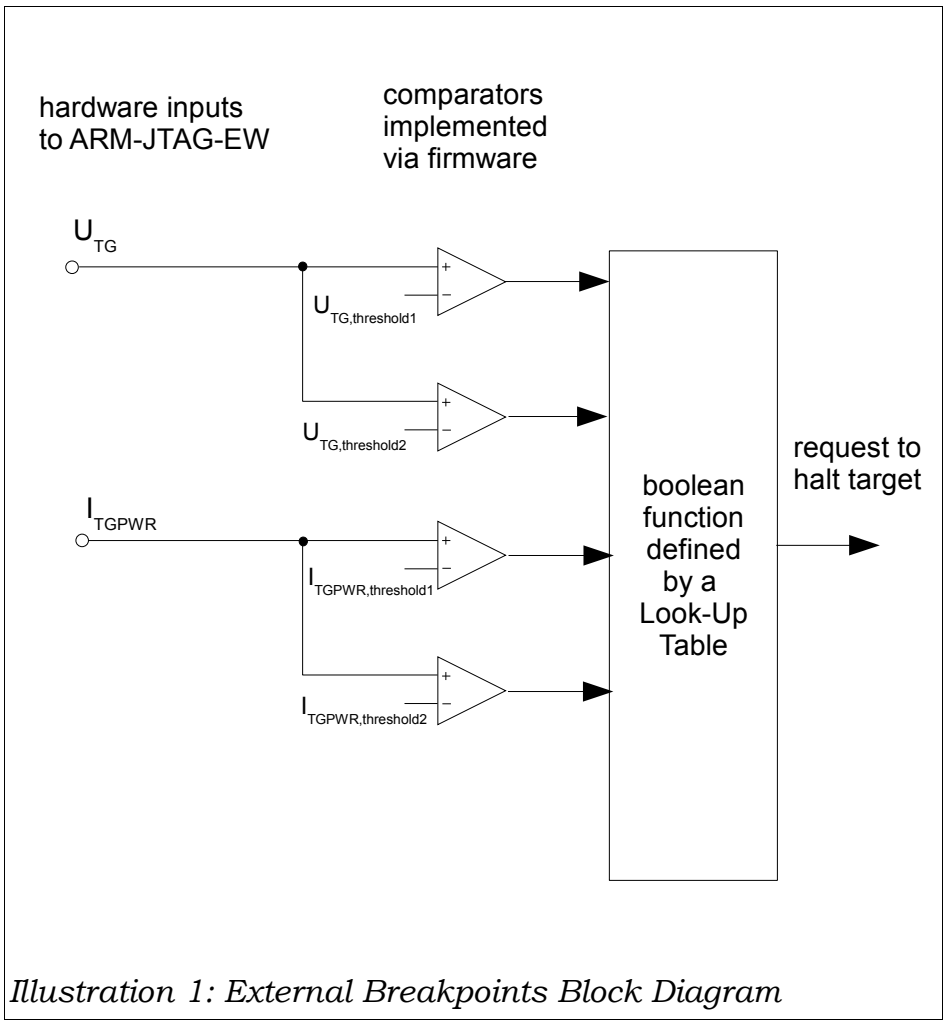


Illustration 1: External Breakpoints Block Diagram

5.2. *Formal event syntax*

The grammar for defining events is very C-like. The following identifiers are recognized:

Identifier	Type	Description
U_tg	analogue	Target power voltage. Expressed in mV.
I_tgpwr	analogue	Target current consumption. Expressed in mA.
true	boolean	Constant high (one)
false	boolean	Constant low (zero)

Analogue variables can only be compared to constant values (mV for voltages, mA for currents). Every analogue variable can be compared against at most two constants, due to the availability of only two comparators in firmware.

A formal definition of the grammar is given below.

```
PRIMARY_BOOLEAN ::= 'true' | 'false' | ('(' EXPR ')')
PRIMARY_ANALOG ::= ['0'-'9']+ | 'U_tg' | 'I_tgpwr'
RELATIONAL_ANALOG ::=    (PRIMARY_ANALOG '<'  PRIMARY_ANALOG)
```