1 GETTING STARTED

Please read carefully the following notes. These things you *should* know first before proceeding to use our RIFF box JTAG module.

JTAG interface uses 4 main signals: TCK, TDI, TDO and TMS. Additionally, there are often present the TRST signal, the RTCK signal, the NRST signal.

If the TRST signal is present on the schematics of the current device you are trying to connect to, then the TRST must not be emitted, otherwise connection problems may occur.

The NRST signal usually is the reset of the entire hardware of the device. After applying the NRST signal, hardware is reset to some known starting state. The NRST signal is not strictly required, but most of the *RIFF box repair modules* assume NRST is connected. Without NRST it is still OK to connect to the device, though many connection problems may occur and it may be needed many attempts to 'catch' the connection, since in this case it's impossible to know in what state the hardware exactly is at the connection moment.

The RTCK signal, called '*returned clock*' is an optional signal. Using RTCK allows connection at the maximum possible speed synchronized with run-time clock of the device. It should be noted, that RIFF Box hardware has maximum 18.000MHz RTCK sampling frequency.

In most cases RTCK signal dramatically increases connection stability. Moreover, sometimes connection is possible only with adaptive clocking scheme. Thus**,** *always tend to use RTCK signal whenever it is possible*.

Main point that has to be understood: JTAG interface only gives ability to communicate with hardware in just few simple ways:

1. Reading/writing memory;
2. Executing MCU code.

Having a JTAG device in hands absolutely does not mean it is easy now to repair a dead board. In most cases the board has complex hardware, NAND memories and RAM memories which are all connected to the main core chipset which, in its turn, is quite intricate. To get access to these memories, sometimes quite a difficult hardware initialization sequence is to be performed.

Every new board often requires different kinds of hardware initialization. For example if a known Qualcomm chipset is used in the current board, it does not mean at all that settings from some previous board with same chipset onboard will be compatible.

Frequencies (oscillators), memories allocation (in other words chip select signals used), internal power control, etc, etc, **do not have any relation to the chipset type** (to definite measures of course) and are only the pure fantasy and imagination of hardware developers.

So, having some unknown (not yet supported) hardware/board in hands, in many cases it is needed to do following:

1. To find out how the firmware itself performs the initialization of the hardware it runs inside (since if board is dead, there now is no firmware which does these steps, and it is only you now responsible writing proper values at the proper memory locations;
2. To find out how the chipset accesses external memory (the NAND; NOR memory is not mentioned, since it does not require any complicated movements);
3. To write code which will be able to communicate with chipset and to send to it data which is to be written to the flash memory.

4. And of course, to find out what is to be written and to which flash memory locations for the board to be repaired.

These may happen to be unsolvable obstacles for a general customer not familiar with above mentioned problems. The RIFF Box JTAG frees you from this burden due to resurrection feature.

The Resurrection feature is implemented as “Single-Button” solution, thus the customer, having a dead hardware on hands, which has supported repair module installed in the RIFF BOX software, does not have to know anything about hardware. The repair of board is done by single click.

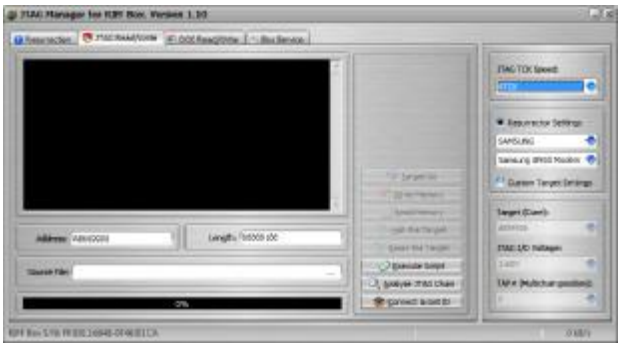
The only actions required from the customer are:

- **v** don’t worry **J**;
- **v** *read resurrection manual* if available for current board;
- **v** disassemble a dead device;
- **v** using schematics find the JTAG interface pads and solder or use external connector to connect RIFF BOX JTAG signals to the board;
- **v** connect battery or just cable to the PC for dead device to get power;
- **v** select device model and manufacturer from the supported list;
- **v** click Resurrect button and wait few seconds until repair is done;
- **v** detach the JTAG connector (or de-solder the wires), assemble device;
- **v** be happy **J**.

Each time the manufacturer and model are selected, the JTAG speed is switched to the recommended speed. The recommended speed means only that during the repair module development stage such speed was found the optimal stable speed depending on the cable used by the module developer.

The TCK speed depends on attached device and the cable used to connect RIFF BOX JTAG to the board’s JTAG pads. The longer cable the more noise is present and the low TCK must be set to have stable data exchange. Using standalone *wire-mess* instead of neat flat cable further on increases the noise and thus increases instability.

Please note, some devices require battery attached for the JTAG communication to be established. For some devices having only USB cable connected to PC is enough to draw power and respond to JTAG. Thus be free to experiment.



The RIFF BOX JTAG offers script execution feature which opens a big field for **advanced users**.

Following are features the RIFF BOX JTAG script manager exports, which may be required for the advanced users/developers:

- – ARM7, ARM9, ARM11, Cortex-A8, PXA3xx, PXA270 cores support;
- – Multiple devices on JTAG chain are supported, thus TAP number selection is available;
- – any custom voltage level selection from range ~1.4V to 3.3V
- – TCK/Adaptive clocking selection
- – Halt core (NRST is not changed)
- – Reset core (NRST is applied before halt)
- – Direct Read memory (by 8/16/32-bit bytes/half-words/words)
- – Direct Write memory (by 8/16/32-bit bytes/half-words/words)
- – Access to the control registers of ARM core (coprocessor 15)
- – Program code breakpoints
- – Run core

Please refer to the *RIFF JTAG Script Manager Specification* for script syntax and implementation details of a script file.

Please refer to the Lauterbach Trace32 PRACTICE language description for details on CMM script files.

1.1 RESURRECTOR

In RIFF BOX JTAG terminology, term *resurrector* means standalone DLL module, which has all information required for a successful firmware repair procedures.

Resurrector contains:

- *DCC Loader*, which has all required code to access (read and write) device's flash memory;
- *Hardware Initialization Data* (script, which is performed by the JTAG Manager software prior the DCC Loader is uploaded into memory and executed);
- *JTAG Pinout Picture* (it is optional, and may be not present). In case the picture is available, there will be ***Interface Pinout*** button visible on the ***Resurrection*** page of the JTAG Manager Software;
- *Short Resurrection Manual* (it is optional, and may be not present). In case the info is available, there will be ***Resurrection Help*** button visible on the ***Resurrection*** page of the JTAG Manager Software.

Please note the ***Interface Pinout*** feature is made solely for your convenience. Whenever this feature (button) is available for a selected device, do check it. Thus you will not need to browse the internet (and in some cases this information is not available on the net at all) for the proper JTAG pinout.

Whenever ***Resurrection Help*** feature is available, *do read its info carefully*. It contains exact information about which shall be your actions for a successful JTAG connection and the following resurrection. *Please don't neglect it*, even if you are a professional. It may contain slight unnoticeable trick information as for example whether it's strictly necessary to use battery instead of external power supply, or are there some peculiar tricks (like hidden back cover switch which has to be pressed in order to establish successful connection); whether it's required to hold power on button during connection or such is not needed; etc.

1.2 RIFF BOX Pinouts

RIFF BOX uses standard ARM 20-pin JTAG interface connector:

1	VCC	2	N.C.
3	TRST	4	GND
5	TDI	6	GND
7	TMS	8	GND
9	TCK	10	GND
11	RTCK	12	GND
13	TDO	14	GND
15	NRST	16	GND
17	N.C.	18	GND
19	N.C.	20	GND

The RIFFBOX RJ-45 Connector:

1	4.2V
2	UART TX
3	UART RX
4	UART TX2
5	MBUS
6	PROBE
7	BSI
8	GND

The 4.2V output voltage can be used for the target power voltage instead of battery or in cases when usage of battery is inconvenient because of disassembled state of device, etc.