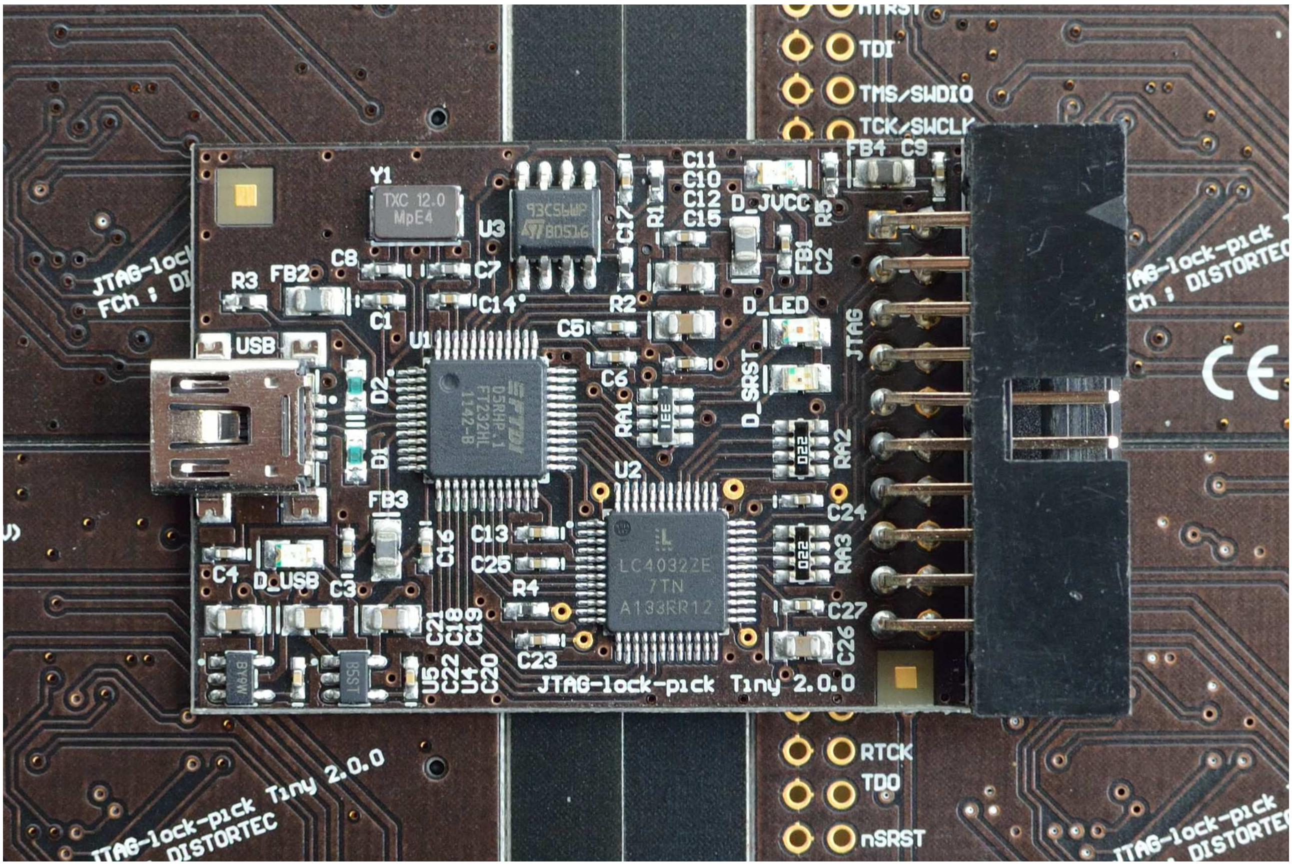
JTAG-lock-pick Tiny 2.x.x
Manual EN 1.0 / 130202

www.distortec.com
www.freddiechopin.info

Table of contents

1. Introduction

JTAG-lock-pick Tiny 2 is an *ARM* core processors' *JTAG* using *USB 2.0* bus to connect to *PC*. The device is based on *FTDI FT232H*[1] chip – a single channel *USB <=> UART/FIFO* converter. Designs goals of **JTAG-lock-pick Tiny 2** project were to create a small and cheap, but at the same time fully functional and fast, interface that would fit various needs.



***Pic. 1:** JTAG-lock-pick Tiny 2 - pcb*

Thanks to *USB* bus the device can be connected to any *PC* on the market – it would not be possible with parallel interface (*LPT*), which is completely obsolete nowadays. Use of more recent standard – *USB 2.0* – allows to increase speed of operation by *30 – 270%* when compared to *USB 1.x*[2].

Maximum frequency of *JTAG* interface's clock is *30MHz* and it is also possible to use *RTCK* mode (so called *Adaptive Clocking*, in which clock frequency adapts itself dynamically to target chip's clock, using feedback connection). Use of advanced *CPLD* chip – *Lattice's ispMACH 4000ZE* series – as line buffers allows connecting target devices with wide range of supply voltage – from *1.4V* to *3.6V*. Inputs of the interface tolerate voltage higher than supply voltage (up to *5.5V*), so it's possible to use *5V* target chips under additional conditions (see **chapter 2.3**).

Additionally **JTAG-lock-pick Tiny 2** supports communication with target chip via new *SWD* (*Serial Wire Debug*) interface, which uses only two lines – bidirectional data line *SWDIO* and clock *SWCLK*.

JTAG-lock-pick Tiny 2 interface can be used to program other types of target chips, such as *FPGA*, *CPLD*, *AVR* or *MIPS*.

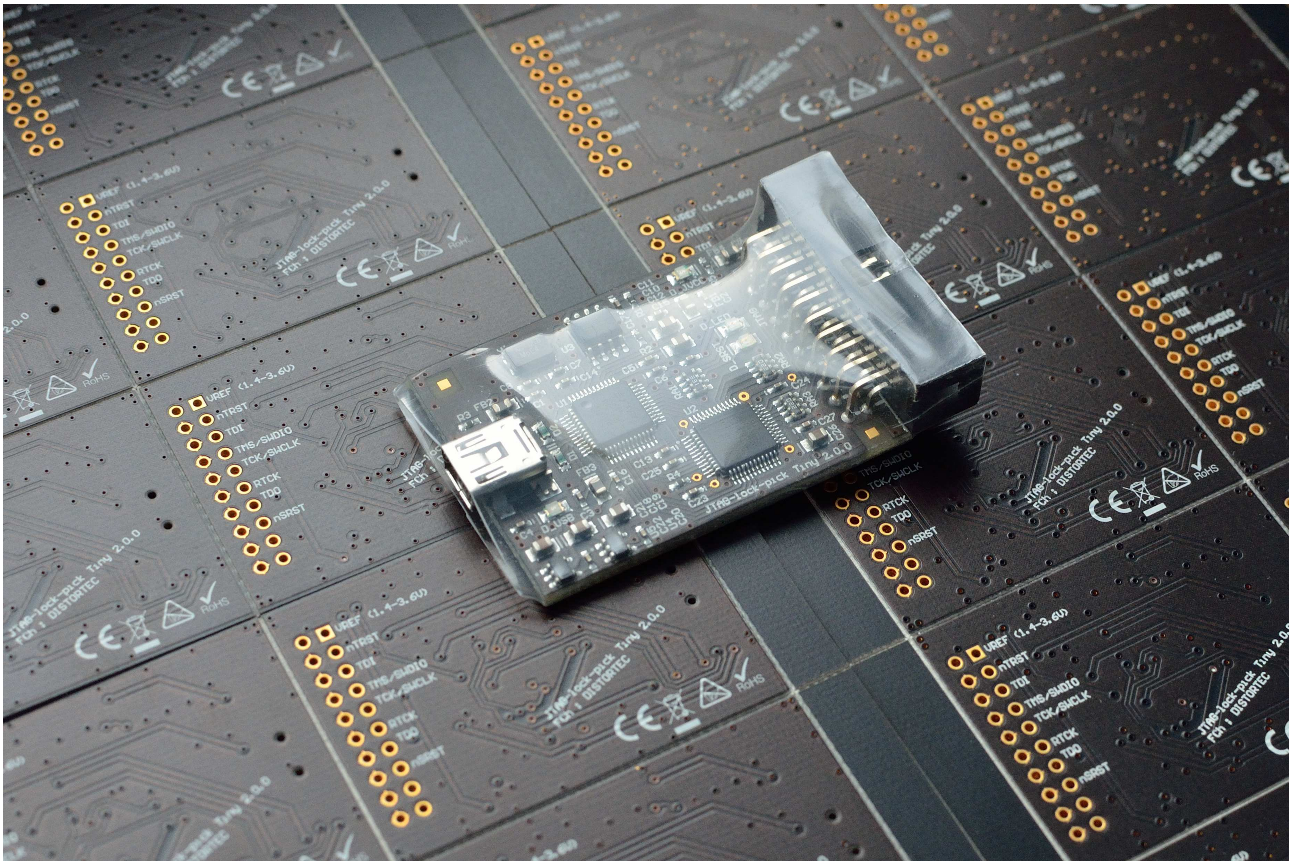
[1] http://www.ftdichip.com/Products/ICs/FT232H.htm

[2] http://permalink.gmane.org/gmane.comp.debugging.openocd.devel/21828

JTAG-lock-pick Tiny 2 is partially compatible with *KT-LINK*[3] interface manufactured by *KrisTech*, thus in many applications it is possible to use existing configurations instead of creating them manually.

JTAG-lock-pick Tiny 2 interface has separate *SRST* and *TRST* lines, which can be independently configured to *push-pull* or *open-drain* mode.

JTAG-lock-pick Tiny 2 project is an improved and miniaturized successor to **JTAG-lock-pick 1.x.x**[4].



Pic. 2: *JTAG-lock-pick Tiny 2*

Picture 1 shows assembled **JTAG-lock-pick Tiny 2** circuit, **pictures 2** and **3** show the circuit in transparent heat-shrink tube which works as an enclosure.

1.1. Supported target chips

JTAG-lock-pick Tiny 2 interface and *PC* software that uses it (among others: *OpenOCD*, *Atollic TrueSTUDIO*, *CooCox CoIDE*, *Keil MDK-ARM*, *IAR Embedded Workbench for ARM*, *Rowley CrossWorks for ARM*, see **chapter 4**) are able to communicate with almost any existing type of ARM processor, including the most popular:

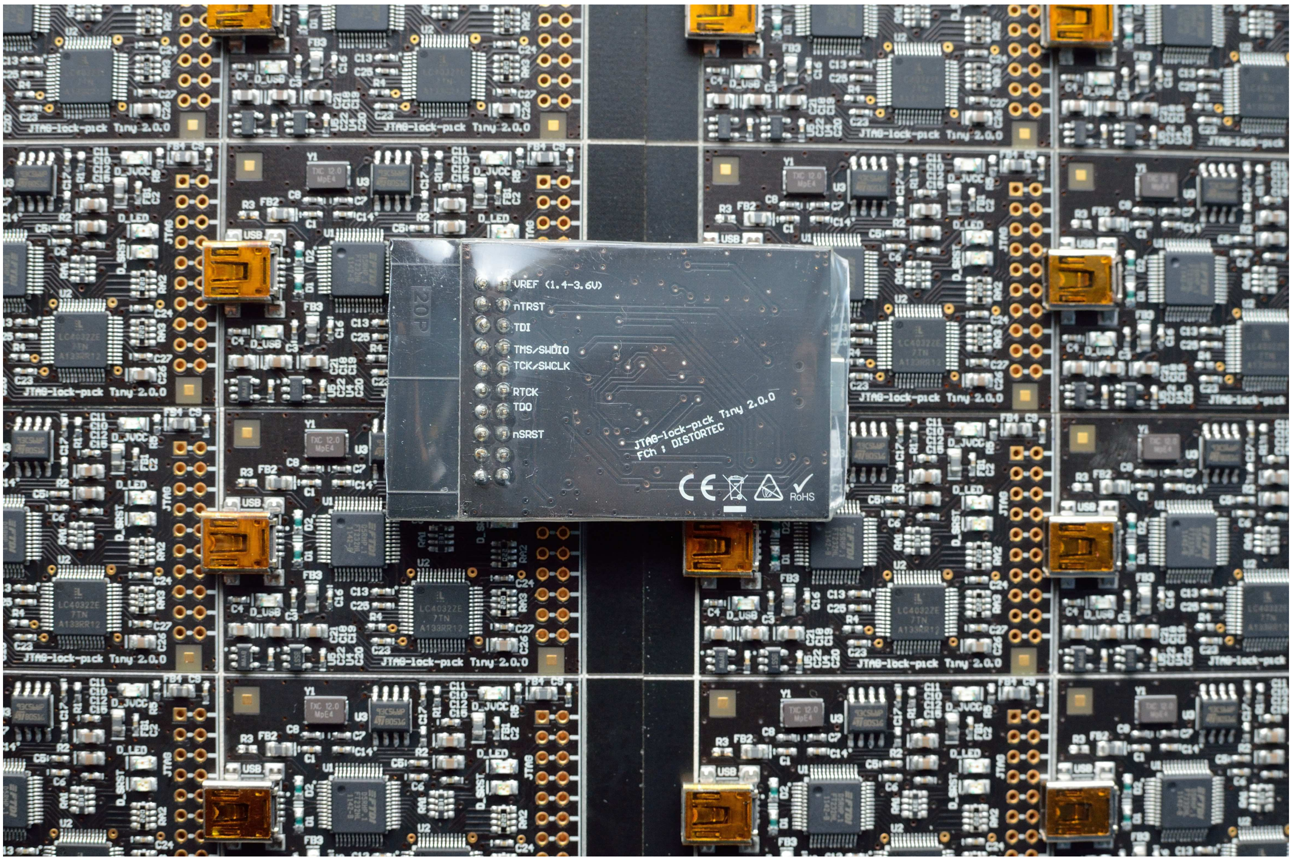
- *Cortex-M3* (*STM32*, *LM3S*, *LPC17xx*, *AT91SAM3*, …),
- *ARM7* (*LPC2xxx*, *AT91SAM7*, *STR7xx*, ...),
- *ARM9* (*LPC3xxx*, *AT91SAM9*, *STR9xx*, ...),
- ...

[3] http://shop.kristech.pl/p/24/257/kt-link-.html

[4] http://www.distortec.com/jtag-lock-pick

Additionally this interface can be used to program other types of target chips, such as *FPGA*, *CPLD*, *AVR* or *MIPS*.

The only limiting factor is support for specific chip in *PC* software.



***Pic. 3:** JTAG-lock-pick Tiny 2*

1.2. “Strong points” of JTAG-lock-pick Tiny 2 project

- *USB 2.0 Hi-Speed 480Mbps* bus,
- support for *SWD* (*Serial Wire Debug*) interface,
- *JTAG* clock frequency up to *30MHz*, support for *Adaptive Clocking* using *RTCK* line,
- safe and reliable communication with target devices with supply voltage in the range from *1.4V* up to *3.6V*, inputs tolerate up to *5.5V* signals, all lines buffered with advanced *CPLD* chip,
- separate *SRST* and *TRST* lines, which can be independently configured to *push-pull* or *open-drain* mode
- transparent heat-shrink tube *“enclosure”*, which protects *JTAG*, connected *PC* and target device from accidental damage,
- small size of whole device (approximate dimensions *55mm x 34mm x 12mm*) and small price,
- ...

1.3. Contents of the package

- **JTAG-lock-pick Tiny 2** debugger / programmer, machine assembled, tested, enclosed in transparent heat-shrink tube (approximate dimensions *55mm x 34mm x 12mm*) (**picture 1**, **2** and **3**),

- *JTAG <=> target* ribbon cable, *20cm*, (**picture 4**),
- *USB mini-B* cable, black, *1,8m*,
- *DVD* with manual, drivers, set of useful software – free (*gcc* toolchains: *CodeSourcery*, *Linaro* and *bleeding-edge-toolchain*; *IDE Eclipse*, *OpenOCD*, *CooCox CoIDE*) and proprietary in evaluation versions (*Atollic TrueSTUDIO for STM32 Lite*, *Keil MDK-ARM* + *CooCox CoMDKPlugin*, *IAR Embedded Workbench for ARM* + *CooCox CoIARPlugin*, *Rowley CrossWorks for ARM*) – all of them in the most recent versions – and other required files.

2. Hardware

2.1. JTAG connector

Signals' assignment in *JTAG* connector (shown in **table 1**) is compatible with the so-called *"standard"*.

Table 1: *Pinout of JTAG connector*

JVCC – 1	*2 – JVCC*
nTRST – 3	*4 – GND*
TDI – 5	*6 – GND*
TMS / SWDIO – 7	*8 – GND*
TCK / SWCLK – 9	*10 – GND*
RTCK – 11	*12 – GND*
TDO – 13	*14 – GND*
nSRST – 15	*16 – GND*
n/c – 17	*18 – GND*
n/c – 19	*20 – GND*

JVCC lines are supply rails for (half of) buffering chip (*CPLD*). **Picture 4** shows typical *20-wire* ribbon connection cable.

2.2. LEDs

On **JTAG-lock-pick Tiny 2's** *PCB* there are four *LEDs*. Their meaning is as follows:

- *D_USB* (green), near *USB* connector – *USB* device enumeration finished, **JTAG-lock-pick Tiny 2** was properly discovered by operating system,
- *D_JVCC* (green), near *JTAG* connector – presence of valid supply voltage for (half of) buffering chip (*CPLD*) of *JTAG* interface (and target circuit),
- *D_SRST* (yellow), near *JTAG* connector, in the center of module – *nSRST* signal is in active (low) state – target device is in reset,
- *D_LED* (red), near *JTAG* connector, in the center of module – state of this *LED* is controlled by *PC* application that is communicating with the interface, for example in case of *OpenOCD* this *LED* blinks when interface is correctly connected with *PC* software.

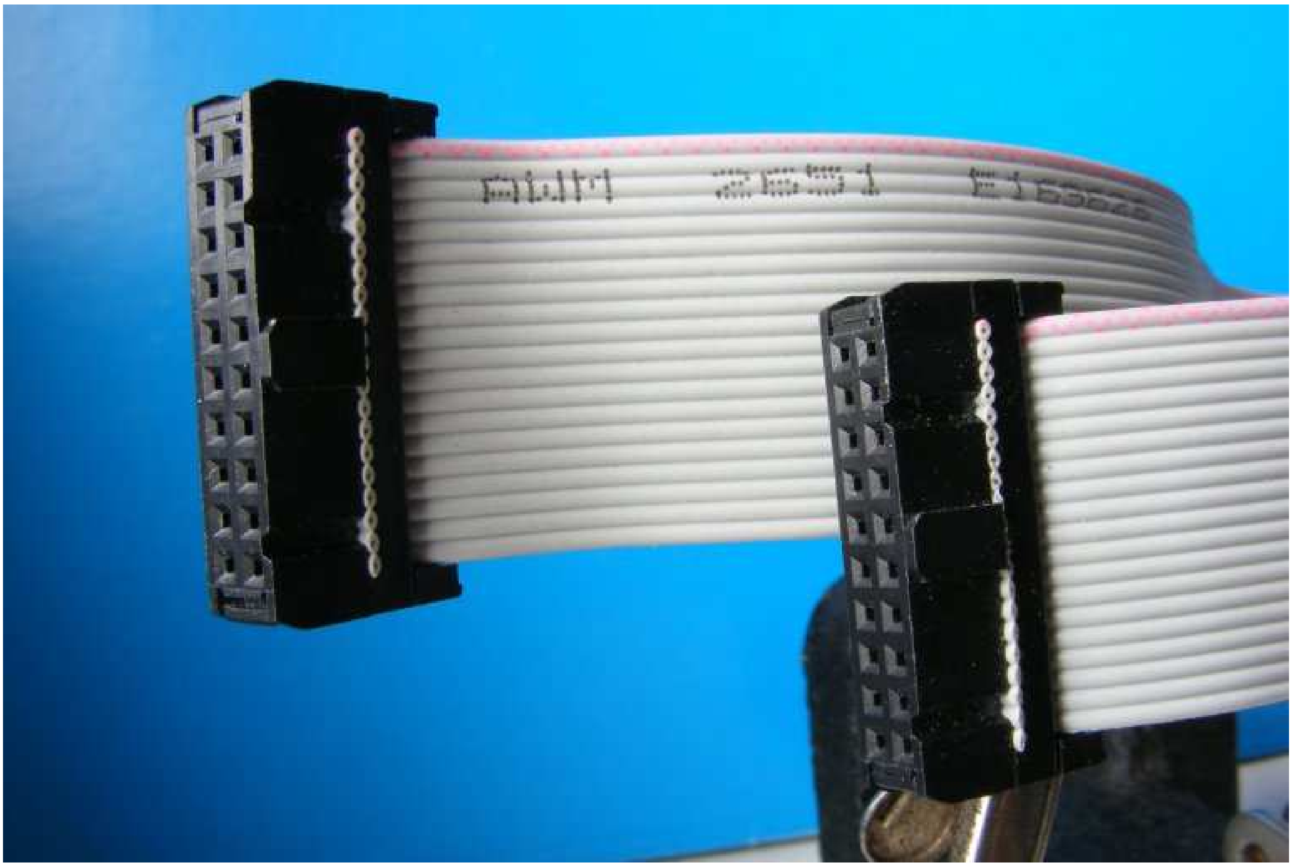
2.3. Interfacing with 5V target chips

JTAG-lock-pick Tiny 2 interface in a typical configuration allows communication with target chips supplied with voltage in range from *1.4V* up to *3.6V*, however under additional conditions and with proper hardware *"configuration"* it is possible to use it for signal levels up to *5.5V*, so with *5V* target chips.

These conditions are as follows:

1. pins *1* and *2* of *JTAG* connector (*JVCC*) should be connected to a voltage in range from *1.4V* up to *3.6V* (for example *3.3V*), **voltage higher than *3.6V* (for example *5V* target chip's supply voltage) must not be connected to these pins in any case**,
2. target chip must be able to correctly interpret signal voltage *(JVCC – 0.4V)* as high level – for example with *3.3V JVCC* output voltage of interface for high level will be in range from *2.9V* up to *3.3V*, so it would not be possible to work with *5V* target chip that states its minimal high level voltage to be *0.7 * VCC* (*3.5V*) – such information can be found in datasheet of target chip in question,
3. Depending on hardware configuration of target device, use of *push-pull* mode for *nSRST* and/or *nTRST* lines may not be possible, so *open-drain* mode should be used with additional pull-up resistors connected to target device's supply voltage.

Proper operation of device in such scenario is possible only if **all** listed conditions are met.



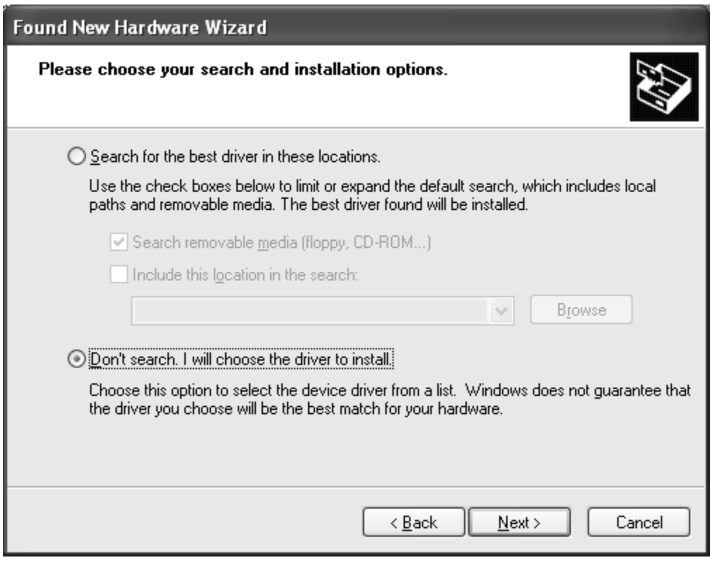
Pic. 4: *Standard JTAG connection ribbon cable*

3. Drivers

First connection of *JTAG* to computer should result in discovery of new *USB* device, for which chosen driver should be installed. Basically there are three options of drivers:

- *WinUSB*,
- *libusb-win32*,
- *FT232H* manufacturer's driver based on *ftd2xx* library.

The choice depends on the software that will be used for debugging with **JTAG-lock-pick Tiny 2**. *OpenOCD* versions that are currently available in the internet use *libusb-1.0* library (configuration files in *scripts/interface/ftdi/* folder) or *libftdi* library (configuration files in *scripts/interface/* folder), so they need (respectively) *WinUSB* or *libusb-win32* driver – it is recommended to choose the first option (*WinUSB*), as it allows for significant increase of speed of operation. Individual compilation of *OpenOCD*, so that it would use slightly faster (in *Windows*) *ftd2xx* driver, is possible but complicated. *CooCox CoIDE* environment and *CooCox* plugins: *CoMDKPlugin* (for *Keil MDK-ARM*) and *CoIARPlugin* (for *IAR Embedded Workbench for ARM*) use *ftd2xx* driver. *Rowley CrossWorks for ARM* environment can use both *ftd2xx* and *libusb-win32*[5] driver.


Found New Hardware Wizard

Please choose your search and installation options.

○ Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

☑ Search removable media (floppy, CD-ROM...)

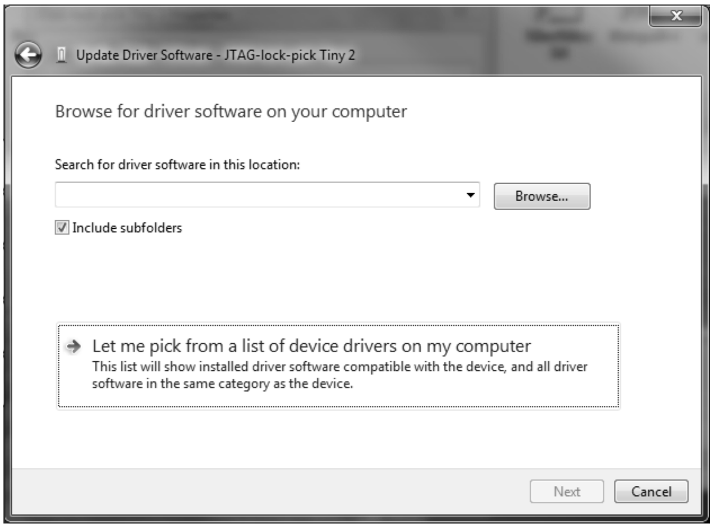
□ Include this location in the search:

● Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

Pic. 5: *Manual installation of drivers in Windows XP*

Archive with all types of drivers for **JTAG-lock-pick Tiny 2** is located on the *DVD* included in the package in *Drivers* folder. The most recent versions of drivers can also be found on *DISTORTEC's* website (www.distortec.com) in *Download* section and on *Freddie Chopin's* website (www.freddiechopin.info) in *Download → Projects → JTAG-lock-pick* section.


Update Driver Software - JTAG-lock-pick Tiny 2

Browse for driver software on your computer

Search for driver software in this location:

☑ Include subfolders

Let me pick from a list of device drivers on my computer

This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

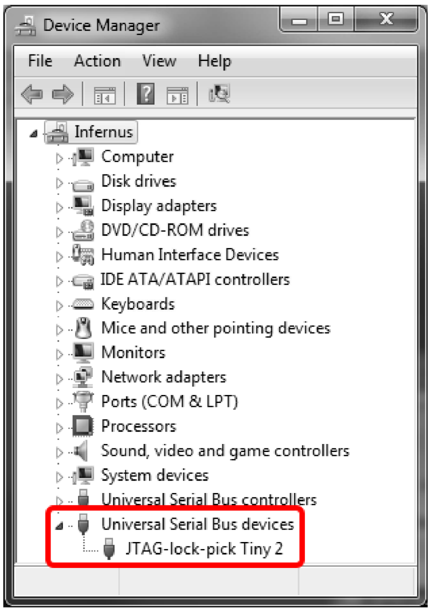
Pic. 6: *Manual installation of drivers in more recent versions of Windows*

In each case it is recommended to install the chosen driver in fully manual mode, in which the driver's file is selected directly, not searched by operating system – during installation this option is called *“Don't search. I will choose the driver to install.”* (see **picture 5**) in case of *Windows XP* or *“Let me pick from a list of device drivers on my computer”* (see **picture 6**) in case of more recent versions of *Windows*. Manual installation is also necessary when changing driver (see **chapter 3.4**).

[5] At the moment of writing this text, use of *libusb-win32* driver with **JTAG-lock-pick Tiny 2** in *CrossWorks* environment (version *2.2.0*) is not possible.

3.1. WinUSB

Correct installation of this driver results in *JTAG-lock-pick Tiny 2* device appearing in *Universal Serial Bus devices* group in *Device Manager*. This situation is shown on **picture 7**.



Infernus
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
- Universal Serial Bus devices
  - JTAG-lock-pick Tiny 2

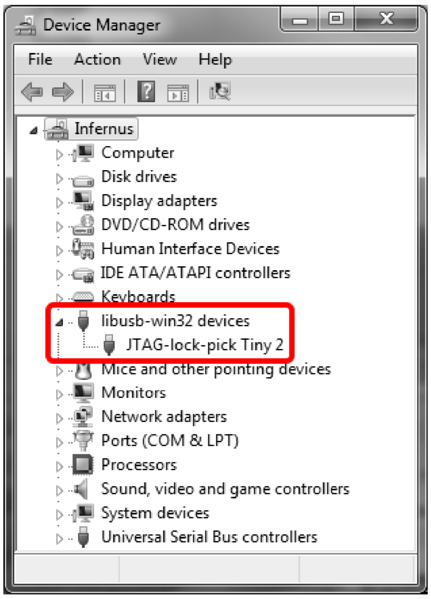
Pic. 7: Installed WinUSB driver

3.2. libusb-win32

Correct installation of this driver results in *JTAG-lock-pick Tiny 2* device appearing in *libusb-win32 devices* group in *Device Manager*. This situation is shown on **picture 8**.

3.3. ftd2xx

Correct installation of this driver results in *JTAG-lock-pick Tiny 2* device appearing in *Universal Serial Bus Controllers* group in *Device Manager*. This situation is shown on **picture 9**.

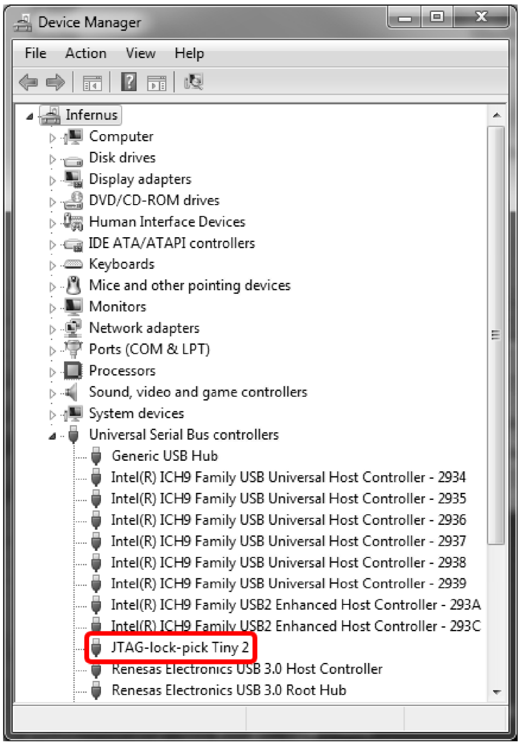


Infernus
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- libusb-win32 devices
  - JTAG-lock-pick Tiny 2
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers

Pic. 8: Installed libusb-win32 driver

3.4. Change of driver

If a need to change the driver occurs (because of – for example – mistake or change of used software), *JTAG-lock-pick Tiny 2* element in *Device Manager* should be located and double-clicked to open its properties. Then *Driver Update...* button in *Driver* tab should be used, which will open driver update wizard. It is important not to let the system search for the driver automatically – *"Don't search. I will choose the driver to install."* option should be used (see **picture 5**) in case of *Windows XP* or *"Let me pick from a list of device drivers on my computer"* (see **picture 6**) in case of more recent versions of *Windows* – otherwise the system will most likely install exactly the same driver's version again, without making any change.



Infernus
- Computer
- Disk drives
- Display adapters
- DVD/CD-ROM drives
- Human Interface Devices
- IDE ATA/ATAPI controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
- Ports (COM & LPT)
- Processors
- Sound, video and game controllers
- System devices
- Universal Serial Bus controllers
  - Generic USB Hub
  - Intel(R) ICH9 Family USB Universal Host Controller - 2934
  - Intel(R) ICH9 Family USB Universal Host Controller - 2935
  - Intel(R) ICH9 Family USB Universal Host Controller - 2936
  - Intel(R) ICH9 Family USB Universal Host Controller - 2937
  - Intel(R) ICH9 Family USB Universal Host Controller - 2938
  - Intel(R) ICH9 Family USB Universal Host Controller - 2939
  - Intel(R) ICH9 Family USB2 Enhanced Host Controller - 293A
  - Intel(R) ICH9 Family USB2 Enhanced Host Controller - 293C
  - JTAG-lock-pick Tiny 2
  - Renesas Electronics USB 3.0 Host Controller
  - Renesas Electronics USB 3.0 Root Hub

Pic. 9: Installed ftd2xx driver

4. Software

List of programs in which **JTAG-lock-pick Tiny 2** can be used contains the most popular tools for *ARM* core processors, among others: *OpenOCD*, *Atollic TrueSTUDIO*, *CooCox CoIDE*, *Keil MDK-ARM*, *IAR Embedded Workbench for ARM* and *Rowley CrossWorks for ARM*. Usage of **JTAG-lock-pick Tiny 2** in these applications is described in details in following sub-chapters.

From software's point of view **JTAG-lock-pick Tiny 2** may be used as *KrisTech KT-LINK* with changed *PID* number of *USB* device – *0x8220*. Because of that, use of this device is very simple – in most cases used software will have available configurations for this *JTAG*.

4.1. OpenOCD[6]

JTAG-lock-pick Tiny 2 interface can be used with *OpenOCD* since version *0.6.1* – use with earlier versions is not possible. Following examples will be based on this exact version, for clarity instead of full executable file's name (like *openocd-x64-0.6.1.exe*) a shortened named – *openocd* – is used.

OpenOCD includes configuration files for **JTAG-lock-pick Tiny 2**:

- *scripts/interface/ftdi/jtag-lock-pick_tiny_2.cfg* – which uses *libusb-1.0* library and *WinUSB* driver,
- *scripts/interface/jtag-lock-pick_tiny_2.cfg* – which uses *libftdi* library and *libusb-win32* driver.

To start *OpenOCD* with **JTAG-lock-pick Tiny 2** one of following commands may be used

```
openocd -f interface/ftdi/jtag-lock-pick_tiny_2.cfg -f target/XXX.cfg
or
openocd -f interface/ftdi/jtag-lock-pick_tiny_2.cfg -f board/XXX.cfg
or
openocd -f interface/jtag-lock-pick_tiny_2.cfg -f target/XXX.cfg
or
openocd -f interface/jtag-lock-pick_tiny_2.cfg -f board/XXX.cfg
```

(depending on which driver was installed and what is the location of the target chip / device configuration file).

For example – running *OpenOCD* for a *STM32F1x* family (*ARM Cortex-M3*) chip with *WinUSB* driver installed:

```
openocd -f interface/ftdi/jtag-lock-pick_tiny_2.cfg -f target/stm32f1x.cfg
```

If there are no issues such as lack of physical connection on the board, lack of support in particular chip etc., it is always recommended to configure use of two separate reset lines. Additionally speed of JTAG should always be configured:

```
openocd -f … -f … -c "adapter_khz XXX; reset_config trst_and_srst;"
```

where *XXX* is frequency of *JTAG* in *kHz* – it should be at least *6x*, but usually at least *8x* lower than frequency of the core after reset – for example in *STM32F1x*, which after reset run at *8MHz*, value *1000* (*1MHz*) should be used.

Example command line shown above should result in similar output as shown below (exact messages will – of course – depend on version of *OpenOCD* and used target device):

[6] http://openocd.sourceforge.net/

```
> openocd-x64-0.6.1.exe -f interface/ftdi/jtag-lock-pick_tiny_2.cfg -f
target/stm32f1x.cfg -c "adapter_khz 1000; reset_config trst_and_srst;"
Open On-Chip Debugger 0.6.1 (2012-10-07-10:39)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.sourceforge.net/doc/doxygen/bugs.html
Info : only one transport option; autoselect 'jtag'
adapter speed: 1000 kHz
adapter_nsrst_delay: 100
jtag_ntrst_delay: 100
cortex_m3 reset_config sysresetreq
adapter speed: 1000 kHz
trst_and_srst separate srst_gates_jtag trst_push_pull srst_open_drain
Info : clock speed 1000 kHz
Info : JTAG tap: stm32f1x.cpu tap/device found: 0x3ba00477 (mfg: 0x23b,
part: 0xba00, ver: 0x3)
Info : JTAG tap: stm32f1x.bs tap/device found: 0x16410041 (mfg: 0x020,
part: 0x6410, ver: 0x1)
Info : stm32f1x.cpu: hardware has 6 breakpoints, 4 watchpoints
```

OpenOCD can also be used only for programming target from command line (or with batch files / scripts) with following command:

```
   openocd -f ... -f ... -c "adapter_khz XXX; reset_config trst_and_srst;
   init; reset init; flash write_image erase YYY.EXT; reset run; shutdown;"
```

where *YYY.EXT* is a filename with extension, extensions handled automatically are *“elf”*, *“hex”* and *“bin”*.

This line consists of several *OpenOCD's* commands:

- *init* – required before following *“executable”* commands (other than configuration) in command line,
- *reset init* – resets the device, halts it and performs optional initialization – halting the chip is required prior to programming it,
- *flash write_image erase YYY.EXT* – writes contents of *YYY.EXT* file (where *EXT* is *hex*, *bin* or *elf* extension) to target's flash memory, erasing it before programming (only affected range),
- *reset run* – resets and resumes target's operation, which results in execution of programmed firmware,
- *shutdown* – closes *OpenOCD's* session.

Example of programming *STM32F1x* chip with *stm32_blink_led.hex* file is shown below.

```
> openocd-x64-0.6.1.exe -f interface/ftdi/jtag-lock-pick_tiny_2.cfg -f
target/stm32f1x.cfg -c "adapter_khz 1000; reset_config trst_and_srst;
init; reset init; flash write_image erase d:/stm32_blink_led.hex; reset
run; shutdown;"
Open On-Chip Debugger 0.6.1 (2012-10-07-10:39)
Licensed under GNU GPL v2
For bug reports, read
        http://openocd.sourceforge.net/doc/doxygen/bugs.html
Info : only one transport option; autoselect 'jtag'
adapter speed: 1000 kHz
adapter_nsrst_delay: 100
jtag_ntrst_delay: 100
cortex_m3 reset_config sysresetreq
adapter speed: 1000 kHz
trst_and_srst separate srst_gates_jtag trst_push_pull srst_open_drain
```

```
Info : clock speed 1000 kHz
Info : JTAG tap: stm32f1x.cpu tap/device found: 0x3ba00477 (mfg: 0x23b,
part: 0xba00, ver: 0x3)
Info : JTAG tap: stm32f1x.bs tap/device found: 0x16410041 (mfg: 0x020,
part: 0x6410, ver: 0x1)
Info : stm32f1x.cpu: hardware has 6 breakpoints, 4 watchpoints
Info : JTAG tap: stm32f1x.cpu tap/device found: 0x3ba00477 (mfg: 0x23b,
part: 0xba00, ver: 0x3)
Info : JTAG tap: stm32f1x.bs tap/device found: 0x16410041 (mfg: 0x020,
part: 0x6410, ver: 0x1)
target state: halted
target halted due to debug-request, current mode: Thread
xPSR: 0x01000000 pc: 0x08000130 msp: 0x20000600
auto erase enabled
Info : device id = 0x20036410
Info : flash size = 128kbytes
wrote 22528 bytes from file d:/stm32_blink_led.hex in 1.382812s (15.910
KiB/s)
Info : JTAG tap: stm32f1x.cpu tap/device found: 0x3ba00477 (mfg: 0x23b,
part: 0xba00, ver: 0x3)
Info : JTAG tap: stm32f1x.bs tap/device found: 0x16410041 (mfg: 0x020,
part: 0x6410, ver: 0x1)
shutdown command invoked
```

4.2. Atollic TrueSTUDIO[7]

In this software **JTAG-lock-pick Tiny 2** can be used via *OpenOCD* (see **chapter 4.1**), but the co-operation is not completely smooth.

Before attempting any configuration the project should be built so that executable file with *.elf* extension has been created. *OpenOCD* should be started in the background – *"externally"* (using operating system's command line) or by configuring it in *Run > External Tools > External Tools Configurations...* menu, using description from chapter dedicated to *OpenOCD* (see **chapter 4.1**).

Debugging configuration can be performed with *Debug Configurations...* option from *Run* menu. In newly opened window new *Embedded C/C++ Application* type configuration should be created and configured in following order:

1. in *Main* tab the name of the project that is meant to be debugged should be typed into (or selected with *Browse...* button) into *Project* field, and then name of executable *.elf* file should by typed into (or selected with *Search Project...* or *Browse...* button) *C/C++ Application* field just above; in most cases these fields will be filled automatically when creating debugging configuration;
2. in *Debugger* tab *Connect to remote GDB server* option should be chosen and *3333* value should be entered into *Port number* field, all other options are ignored – their values do not affect anything and they should be considered non-functional;
3. in *Startup Debug* tab whole content of *Initialization Commands* field should be removed and replaced with following lines:

```
# send "reset init" to OpenOCD
monitor reset init
# load application to target via GDB
load
# send "reset init" to OpenOCD
monitor reset init
```

[7] http://www.atollic.com/index.php/truestudio

```
# set temporary breakpoint at main() and resume target
tbreak main
continue
```

4. all changes should be confirmed with *Apply* button and debugging session can be started right away with *Debug* button.

Problems that should be expected during cooperation of *Atollic TrueSTUDIO* environment with *OpenOCD* are (among others):

- target device cannot be *"reset"* during debugging session with *Restart* button;
- operating system informs about improper termination of *GDB* application when debugging session is closed;

4.3. CooCox CoIDE[8]

In this environment available configuration for *KrisTech KT-LINK* should be used. All options related to *JTAG* interface configuration can be found in *Debug > Debug Configuration* menu, after selecting the debug configuration for active project that will be available there (typically it will be called *project_name.configuration*). *KT-Link* option should be selected from the *Adapter* list in *Hardware* groupbox in the first tab (*Debugger*) and then click *Apply* button and *Close* button.

4.4. IAR Embedded Workbench for ARM[9]

In *IAR's* environment two approaches to using **JTAG-lock-pick Tiny 2** interface are possible – one can use *OpenOCD* (see **chapter 4.1**) or *CooCox CoIARPlugin*.

4.4.1. OpenOCD

First step is – obviously - starting *OpenOCD* in background (via operating system's command prompt) with parameters matching target device that is used, using description from chapter dedicated to *OpenOCD* (see **chapter 4.1**).

Then in *IAR Embedded Workbench for ARM* environment's project options (*Project > Options*) *Debugger* option should be selected from the side menu. In *Setup* tab *GDB Server* option should be selected from *Driver* list and in *Download* tab *Use flash loader(s)* option should be checked. Then select *GDB Server* option from side menu and in *GDB Server* tab enter *localhost* into *TCP/IP address of hostname [.port]* field.

4.4.2. CooCox CoIARPlugin[10]

After installing the plugin select *Debugger* in side menu of project's options (*Project > Options*). In *Setup* tab *RDI* option should be selected from *Driver* list and in *Download* tab *Use flash loader(s)* option should be checked. Then select *RDI* option from side menu, type path to *CoRDI.dll* file located in *CooCox CoIARPlugin* installation folder (typically it will be *c:\Program Files\CooCox\CoIARPlugin\CoRDI.dll*, in *64-bit* system *Program Files (x86)* folder will be used) into *Manufacturer RDI driver* field and check *Allow hardware reset* option. After closing project's options with *OK* button, new menu – *RDI* – will appear in application. The only active option – *Configure* – should be selected from this menu – in newly opened window target device should be

[8] http://www.coocox.org/CooCox_CoIDE.htm
[9] http://www.iar.com/en/Products/IAR-Embedded-Workbench/ARM/
[10] http://www.coocox.org/CoLinkGuide/CoIARPlugin.html

selected from the side list and *KT-Link* option should be selected from the *Adapter* list in *Adapter Config* groupbox.

4.5. Keil MDK-ARM[11]

Use of **JTAG-lock-pick Tiny 2** interface in *Keil MDK-ARM* environment (also known as *μVision* or *RealView*) is possible with *CooCox CoMDKPlugin*[12]. After installing it, in project's options (*Project > Options for Target 'project_name'*, option available only when project is selected in *Projects* window) in *Utilities* tab *CooCox Debugger* should be selected from the list under *Use Target Driver for Flash Programming*, *Update Target before Debugging* option should be checked and *Settings* button should be clicked. In newly opened window go to *Debug* tab and select *KT-Link* from *Adapter* list in *USB Adapter* groupbox. After pressing *OK* button and going back to project's options go to *Debug* tab and select *Use* option on the right side, then – again – select *CooCox Debugger* option from the list next to it – there is no need to configure it again with *Settings* button, because this configuration is shared (the same for *Utilities* and *Debug* tabs). The last options that should be checked are located just below: *Load Application at Startup* and *Run to main()*. Whole configuration should be confirmed with *OK* button.

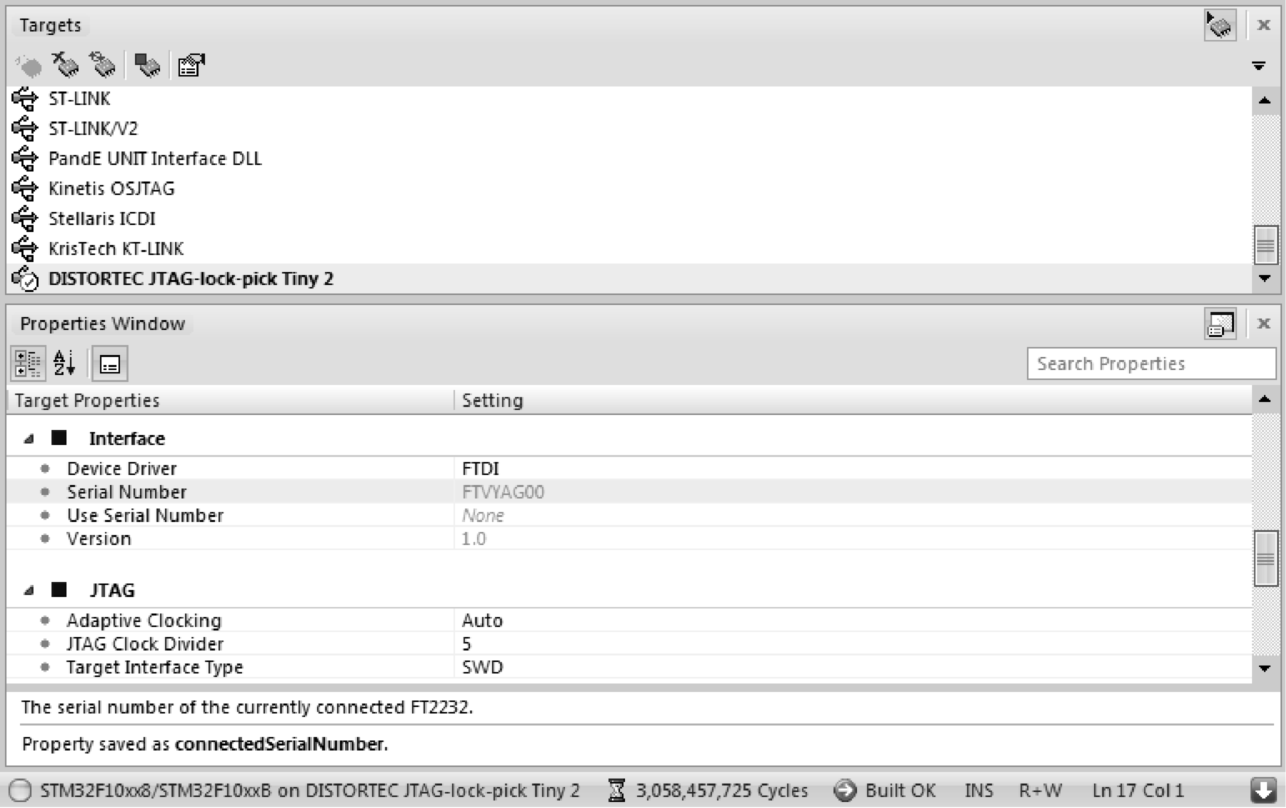
4.6. Rowley CrossWorks for ARM[13]

To enable **JTAG-lock-pick Tiny 2** use in *CrossWorks* environment an interface configuration must be imported from *targets.xml* file which is located in *Rowley CrossWorks for ARM/* folder on the *DVD* included in the package. To do that, click anywhere within *Targets* frame with right mouse button and choose *Import Target Definitions From XML* option and then select *targets.xml* file mentioned previously. After successful import of configuration *DISTORTEC JTAG-lock-pick Tiny 2* option should appear on the list.

Additional configuration options (*JTAG/SWD* interface, clock divider, …) are available in *Properties Window* frame when *DISTORTEC JTAG-lock-pick Tiny 2* interface is selected in the *Targets* frame.

Proper connection with target device is indicated on application's status bar and by appearance of *JTAG's* serial number in *Properties Window* for selected interface – this situation is shown on **picture 10**.

[11] http://www.keil.com/arm/mdk.asp
[12] http://www.coocox.org/CoLinkGuide/CoMDKPlugin.html
[13] http://www.rowley.co.uk/arm/index.htm



Targets

- ST-LINK
- ST-LINK/V2
- PandE UNIT Interface DLL
- Kinetis OSJTAG
- Stellaris ICDI
- KrisTech KT-LINK
- **DISTORTEC JTAG-lock-pick Tiny 2**

Properties Window

Target Properties	Setting
Interface	
Device Driver	FTDI
Serial Number	FTVYAG00
Use Serial Number	*None*
Version	1.0
JTAG	
Adaptive Clocking	Auto
JTAG Clock Divider	5
Target Interface Type	SWD

The serial number of the currently connected FT2232.

Property saved as **connectedSerialNumber.**

STM32F10xx8/STM32F10xxB on DISTORTEC JTAG-lock-pick Tiny 2 3,058,457,725 Cycles Built OK INS R+W Ln 17 Col 1

Pic. 10: *Proper connection with target device in Rowley CrossWorks for ARM environment*

5. Sources of additional information

Additional information and support about *JTAG*, software, debugging and *ARM* core processors can be found on many websites:

- *DISTORTEC's* website (http://www.distortec.com/),
- *Freddie Chopin's* website (http://www.freddiechopin.info/),
- *Elektroda's* forum (http://www.elektroda.pl/rtvforum/),
- *SparkFun's* forum (http://forum.sparkfun.com/viewforum.php?f=18),
- *OpenOCD's* website (http://openocd.sourceforge.net/),
- *Yagarto* toolchain's website (http://www.yagarto.de/),
- *WinARM* toolchain's website (information often out of date!) (http://www.siwawi.arubi.uni-kl.de/avr_projects/arm_projects/),
- *google* (http://www.google.com/).

6. Troubleshooting

Problem: After starting *OpenOCD* following message appears:

```
OpenOCD Error: unable to open ftdi device
```

Cause: *OpenOCD* can use three different kinds of drivers to communicate with *FTDI FT232H* chip – *WinUSB*, *libusb-win32* or *ftd2xx*. The choice is made at the stage of compilation of *OpenOCD*. This message can point to several issues:

1. *JTAG* is not properly connected to the computer;
2. *JTAG* is *"blocked"* by another application or by another session of *OpenOCD*;
3. Wrong driver was installed for *JTAG*;
4. Configuration file which does not match installed drivers was chosen;

Solution: *Ad 1.* Check *JTAG <=> PC* connection.
Ad 2. Close all other *OpenOCD* sessions or other software connected with *JTAG*.
Ad 3. Uninstall wrong and install correct driver (see **chapter 3.4**).
Ad 4. Choose correct interface's configuration file (see **chapter 4.1**).

Problem: Attempt to program / debug target device in *CooCox CoIDE* environment results in following error:

```
Target Chip not found
```

Cause: *CooCox CoIDE* environment in version *1.4.0* does not work correctly with *FTDI FTx232* based *JTAGs*.

Solution: Previous (*1.3.0*) or more recent (*1.5.0*) version of the software should be used.

Problem: After starting the debug session in *CrossWorks* environment the software is unresponsive for a longer period of time or reports various errors.

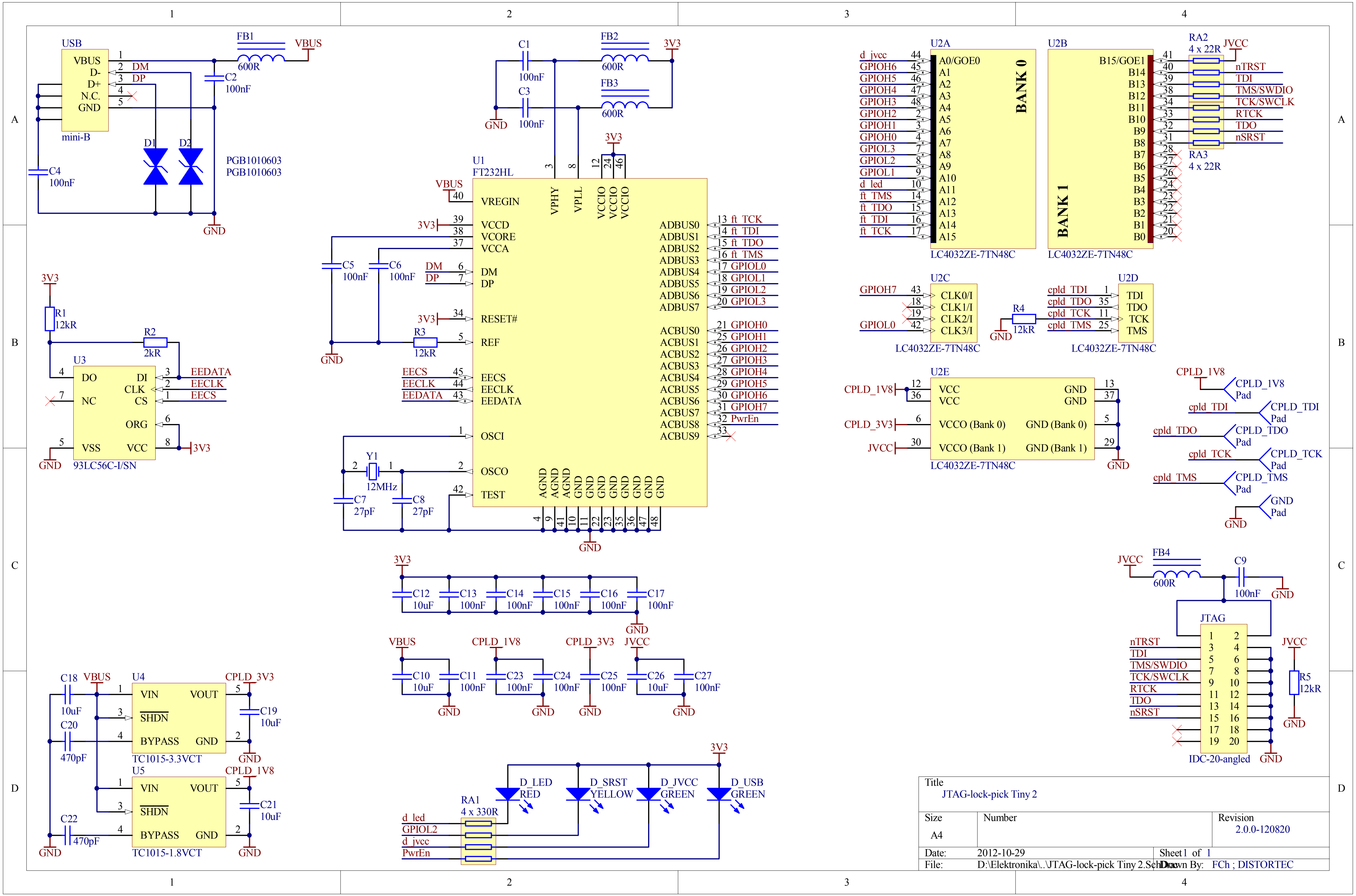
Cause: Even thought this environment allows to use *libusb-win32* drivers (described as *LibUSB* in interface's options), such combination does not work correctly with *FT232H* chip and the most recent (at the moment of writing this text) version of the software (*2.2.0*).

Solution: Install *ftd2xx* driver (see **chapter 3.4**) and select it in interface's options (*Interface > Device Driver > FTDI*).

7. Appendix

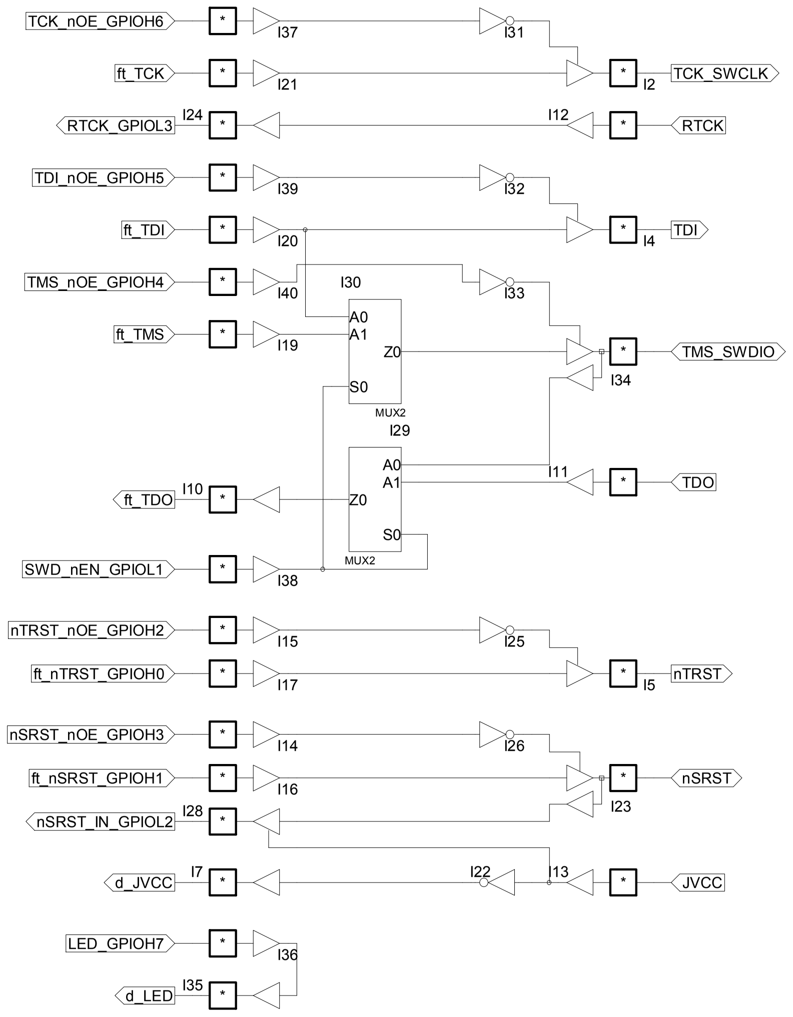
- circuit's schematic diagram (version *2.0.0-120820*), **page 20**
- schematic diagram and configuration of *CPLD* logic (version *2.0.0-120820*), **page 22**
- bill of materials (version *2.0.0-120820*), **page 25**

7.1. Circuit's schematic diagram (version 2.0.0-120820)



Title		
JTAG-lock-pick Tiny 2		
Size A4	Number	Revision 2.0.0-120820
Date: 2012-10-29		Sheet 1 of 1
File: D:\Elektronika\..\JTAG-lock-pick Tiny 2.SchDoc		Drawn By: FCh ; DISTORTEC

7.2. Schematic diagram and configuration of CPLD logic (version 2.0.0-120820)



JTAG-lock-pick Tiny 2.0.0
KT-LINK configuration
120820

	Type	Signal/Group Name	GLB	Macrocell	Pin	Bank	IO Types	Pull	Slewrate	Input registers	Register powerup
1	Input	TDO	B	9	32	1	LVCMOS18	UP	N/A	NONE	N/A
2	Bidirection	nSRST	B	8	31	1	LVCMOS18	UP	FAST	NONE	NONE
3	Input	ft_TDI	A	14	16	0	LVCMOS33	UP	N/A	NONE	N/A
4	Input	ft_TMS	A	12	14	0	LVCMOS33	UP	N/A	NONE	N/A
5	Input	ft_nSRST_GPIOH1	A	6	3	0	LVCMOS33	UP	N/A	NONE	N/A
6	Input	ft_nTRST_GPIOH0	A	7	4	0	LVCMOS33	UP	N/A	NONE	N/A
7	Output	d_JVCC	A	0	44	0	LVCMOS33	UP	FAST	N/A	NONE
8	Output	ft_TDO	A	13	15	0	LVCMOS33	UP	FAST	N/A	NONE
9	Output	nSRST_IN_GPIOL2	A	9	8	0	LVCMOS33	UP	FAST	N/A	NONE
10	Output	TDI	B	13	39	1	LVCMOS18	UP	FAST	N/A	NONE
11	Bidirection	TMS_SWDIO	B	12	38	1	LVCMOS18	UP	FAST	NONE	NONE
12	Output	nTRST	B	14	40	1	LVCMOS18	UP	FAST	N/A	NONE
13	Input	nSRST_nOE_GPIOH3	A	4	48	0	LVCMOS33	UP	N/A	NONE	N/A
14	Input	nTRST_nOE_GPIOH2	A	5	2	0	LVCMOS33	UP	N/A	NONE	N/A
15	Input	LED_GPIOH7		N/A	43	0	LVCMOS33	UP	N/A	NONE	N/A
16	Input	SWD_nEN_GPIOL1	A	10	9	0	LVCMOS33	UP	N/A	NONE	N/A
17	Input	TCK_nOE_GPIOH6	A	1	45	0	LVCMOS33	UP	N/A	NONE	N/A
18	Input	TDI_nOE_GPIOH5	A	2	46	0	LVCMOS33	UP	N/A	NONE	N/A
19	Input	TMS_nOE_GPIOH4	A	3	47	0	LVCMOS33	UP	N/A	NONE	N/A
20	Output	d_LED	A	11	10	0	LVCMOS33	UP	FAST	N/A	NONE
21	Input	JVCC	B	15	41	1	LVCMOS18	DOWN	N/A	NONE	N/A
22	Input	ft_TCK	A	15	17	0	LVCMOS33	DOWN	N/A	NONE	N/A
23	Output	RTCK_GPIOL3	A	8	7	0	LVCMOS33	DOWN	FAST	N/A	NONE
24	Output	TCK_SWCLK	B	11	34	1	LVCMOS18	DOWN	FAST	N/A	NONE
25	Input	RTCK	B	10	33	1	LVCMOS18	DOWN	N/A	NONE	N/A

7.3. Bill of materials (version 2.0.0-120820)

JTAG-lock-pick Tiny 2.0.0-120820 - Bill Of Materials

20.08.2012

Designator	Description	Comment	Footprint	Footprint Description	Value	Quantity
C1, C2, C3, C4, C5, C6, C9, C11, C13, C14, C15, C16, C17, C23, C24, C25, C27	Capacitor	Capacitor	CAPC1005M	Chip Capacitor, Body 1.0x0.5mm, IPC Low Density	100nF	17
C7, C8	Capacitor	Capacitor	CAPC1005M	Chip Capacitor, Body 1.0x0.5mm, IPC Low Density	27pF	2
C10, C12, C18, C19, C21, C26	Capacitor	Capacitor	CAPC2012M	Chip Capacitor, Body 2.0x1.3mm, IPC Low Density	10uF	6
C20, C22	Capacitor	Capacitor	CAPC1005M	Chip Capacitor, Body 1.0x0.5mm, IPC Low Density	470pF	2
D1, D2	Transient Voltage Suppressor diode	PGB1010603	CAPC1608M	Chip Capacitor, Body 1.6x0.8mm, IPC Low Density		2
D_JVCC, D_USB	Typical LED	GREEN	CAPC2012M	Chip Capacitor, Body 2.0x1.3mm, IPC Low Density		2
D_LED	Typical LED	RED	CAPC2012M	Chip Capacitor, Body 2.0x1.3mm, IPC Low Density		1
D_SRST	Typical LED	YELLOW	CAPC2012M	Chip Capacitor, Body 2.0x1.3mm, IPC Low Density		1
FB1, FB2, FB3, FB4	Ferrite Bead	Ferrite Bead	CAPC2012M	Chip Capacitor, Body 2.0x1.3mm, IPC Low Density	600R	4
JTAG	Header, 10-Pin, Dual row, Right Angle	IDC-20-angled	IDC-20, angled	IDC header, angled, 20-pin		1
R1, R3, R4, R5	Resistor	Resistor	RESC1005M	Chip Resistor, Body 1.1x0.5mm, IPC Low Density	12kR	4
R2	Resistor	Resistor	RESC1005M	Chip Resistor, Body 1.1x0.5mm, IPC Low Density	2kR	1
RA1	Quad chip resistor array, 4D03	Resistor array	RESA3216X06M	Chip Resistor Array, 8-Leads, Body 3,2x1,6mm, IPC Low Density	4 x 330R	1
RA2, RA3	Quad chip resistor array, 4D03	Resistor array	RESA3216X06M	Chip Resistor Array, 8-Leads, Body 3,2x1,6mm, IPC Low Density	4 x 22R	2
U1	Single Channel Hi-Speed USB to Multipurpose UART/FIFO IC, LQFP-48, Tape and Reel	FT232HL	TSQFP-50P-900X900X160-48M	TSQFP, 48-Leads, Body 9,0x9,0mm (max), Pitch 0,50mm, IPC Low Density		1
U2	ispMACH 4000ZE Ultra Low Power PLD, 1.8V, 32 User I/Os, 48-Pin TQFP, 7.5ns, Commercial Grade, Pb-Free	LC4032ZE-7TN48C	TSQFP-50P-900X900X160-48M	TSQFP, 48-Leads, Body 9,0x9,0mm (max), Pitch 0,50mm, IPC Low Density		1
U3	2K, 256x8 or 128x16-bit, 2.5V Microwire Serial EEPROM, 8-Pin SOIC 150mil, Industrial Temperature	93LC56C-I/SN	SOIC127P600X175-8M	SOIC, 8-Leads, Body 5,0x4,0mm (max), Pitch 1,27mm, IPC Low Density		1
U4	50 mA, 100 mA and 150 mA CMOS LDOs with Active-Low Shutdown and Reference Bypass, 5-Pin SOT-23, Extended Temperature, Tape and Reel	TC1015-3.3VCT	SOT-95P-284X119-5M	SOT23, 5-Leads, Body 3,0x3,0mm (max), Pitch 0,95mm, IPC Low Density		1
U5	50 mA, 100 mA and 150 mA CMOS LDOs with Active-Low Shutdown and Reference Bypass, 5-Pin SOT-23, Extended Temperature, Tape and Reel	TC1015-1.8VCT	SOT-95P-284X119-5M	SOT23, 5-Leads, Body 3,0x3,0mm (max), Pitch 0,95mm, IPC Low Density		1
USB	USB 2.0, Right Angle, SMT, B Type, Receptacle, 5 Position, Black	mini-B	USB mini-B SMD	Connector; USB2.0, Type B, SM; 5 Position; Right Angle		1
Y1	Crystal Oscillator	12MHz	Quartz SMD 5x3.2	Quartz resonator, SMD, 5x3.2mm		1