



# **CooCox software with Olimex JTAGS**

**(ARM-USB-TINY;ARM-USB-TINY-H; ARM-USB-OCD; ARM-USB-OCD-H)**

## **Step-by-step guide**

**Initial release, April 2012**



All boards produced by Olimex LTD are ROHS compliant

## 1. Introduction to CooCox

What is **CooCox**?

**CooCox** is a company that provides free software that can match the software of the giants in the business. The software is split into modules and two of the major ones are:

**CooCox CoIDE** is a new, free and highly-integrated software development environment for ARM based microcontrollers, which includes all the tools necessary to develop high-quality software solutions in a timely and cost effective manner. It has a built-in debugger.

What is the good about **CooCox CoIDE**?

- Free to use
- Full functional IDE
- Component-oriented development platform
- Internet-based, efficient integration of network resources
- Integrates CoOS
- Peripheral registers
- Growing community and support

**CooCox CoFlash** is a stand-alone Cortex M3 & Cortex M0 Flash Programming software for PCs running Microsoft Windows. CoFlash has an intuitive user interface with a traditional Windows graphical user interface (GUI), and a command line mode as well. CoFlash requires a CoLink. It is able to work with most common Cortex-M3 MCUs. Users can define his own algorithm to support more flash devices, regardless of internal or external flash.

What is the good about **CooCox CoFlash**?

- Free
- Most common Cortex-M3 MCUs supported
- Nuvoton Cortex-M0 MCUs supported
- User-defined flash program algorithm supported
- Intuitive user interface with GUI and Command line mode
- Image file (currently .bin or .elf ) supported
- Keep user's last configuration

## 2. Requirements and preparations

### 2.1 CooCox Software

The best way to get CooCox modules is via a master control module called CoCenter. Go to the [www.coocox.org](http://www.coocox.org) and navigate to the downloads section (<http://www.coocox.org/Downloads.htm>). Click on CoIDE and at the bottom of the page there would be option Download through CoCenter which will download CoCenter executable automatically. Remember the download path we will use it in the next chapter.

### 2.2 GCC Toolchain

Latest version (at the time of writing - 1.41) of CooCox CoIDE doesn't integrate GCC toolchain in it. The suggested toolchain is the ARM GCC, and has to be downloaded separately from the following link (available at the time of writing):

<https://launchpad.net/gcc-arm-embedded/4.6/2011-q4-major>

Choose appropriate one according to your operating system. The common user should stick with Installer packages (available both for Linux and Windows).

Install it and remember the install folder.

Additional info on the compiler: [http://www.coocox.org/CoIDE/Compiler\\_Settings.html](http://www.coocox.org/CoIDE/Compiler_Settings.html).

### 2.3 Compatible Olimex ARM JTAG (ARM-USB-OCD; ARM-USB-OCD-H; ARM-USB-TINY; ARM-USB-TINY-H)

You will need one of those.

### 2.4 OlimexFTDI driver

When getting the drivers for the Olimex debugger be sure to download the FTDI drivers, and NOT the libUSB/OpenOCD ones. CooCox CoIDE and CooCox CoFlash doesn't work with libUSB/OpenOCD drivers. Double-check that. When you download them extract them in a folder and remember the path to its contents.

The link for the drivers is: [https://www.olimex.com/Products/ARM/JTAG/\\_resources/driver-ftd2xx.zip](https://www.olimex.com/Products/ARM/JTAG/_resources/driver-ftd2xx.zip)

### 2.5 Recommended additional hardware/software

It is recommended to have a Cortex M3 board for testing. I used OLIMEX STM32-H107 for the example below.

## 3. Setting up the things together and example program/debug

I used Windows 7 x64; ARM-USB-TINY-H; CoIDE 1.41;

### 3.1 First connect your Olimex debugger to your PC via the USB-A - USB-B cable.

### 3.2 Open Device Manager and check if you have the following devices:

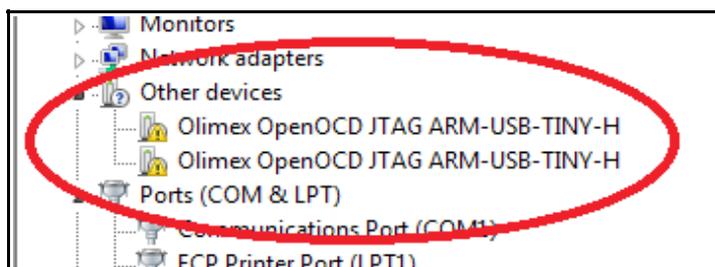
-In Ports (COM & LPT) section : USB Serial Port (COMxx), where XX is a different COM port depending on your devices

-In Universal Serial Bus controllers: USB Composite Device; USB Serial Converter A; USB Serial Converter B

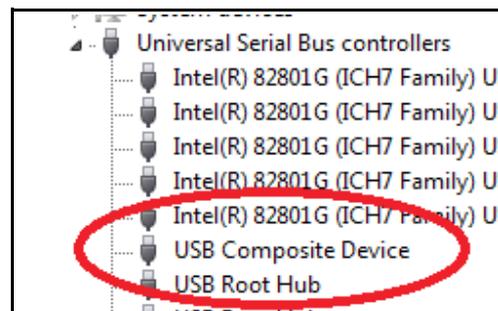
If you have the above mentioned devices with the same names you can skip the uninstall old drivers/install FTDI drivers section, if not continue to the next:

### 3.3 Uninstall Old Drivers/Install FTDI drivers

If you plug your OLIMEX ARM debugger for the first time your computer should show the following devices:



In case you see other devices when you plug the Olimex debugger you have to unistall the software associated with them from the device manager.



For instance if you have installed the libUSB/OpenOCD drivers you right-click over them in the corresponding tab and choose uninstall driver, remember to also mark the check box so that the files are also deleted from the computer:



Repeat the uninstall procedure for every other device in device manager that shows up when you plug the Olimex debugger. For example, the other two I have to uninstall are located in Universal Serial Bus controllers tab - USB Serial Converter B and just in case USB Composite Device:



After all is uninstalled replug the debugger (skip automatic search for drivers) and your Device Manager should look like the screens in the beginning of chapter 3.3. Right-click over each device and point manually (do not use the automatic search for drivers) to the folder you extracted the archive with the FTDI drivers. If the installer pops a warning about software being untrusted software click "Install anyway". This warning shows up upon installing software by companies unwilling or unable to pay the yearly fee for this feature to Microsoft.

Repeat the right-click-point-to-FTDI-drivers procedure for every unrecognized device until there

are no icons with yellow triangle. I had to repeat it 3 times. After the installation you will have the devices mentioned in 3.2.

### 3.4 Installing the CoCoX products is very easy if you have already downloaded and installed CoCenter

Run CoCenter, then click Download and then click Install. Be sure that there isn't firewall blocking the access of CoCenter to the world wide web.



The one required for the featured guide is CoCoX CoIDE and the recommended one is the CoCoX CoFlash. The other downloads/installs are not mandatory.

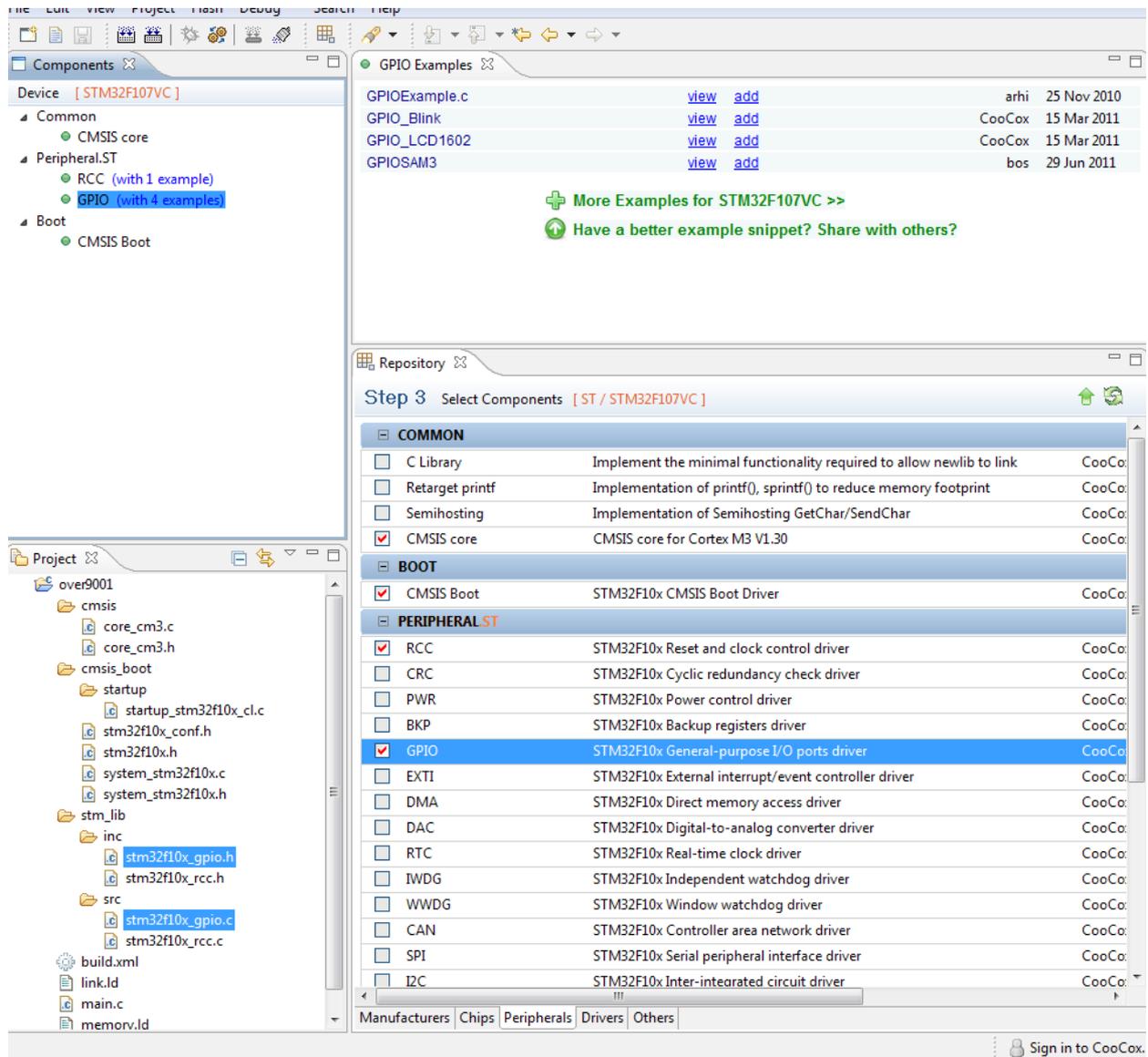
### 3.5 Run CoCoX CoIDE either from the CoCenter or from a desktop shortcut and set the path to the toolchain you downloaded and installed earlier. This is done from Project - > Select Toolchain Path. In my case it is: F:\Program Files (x86)\arm-none-eabi-gcc-4\_6\bin but in your case it will probably differ.

For detailed instructions check here too: [http://www.cocox.org/CoIDE/Compiler\\_Settings.html](http://www.cocox.org/CoIDE/Compiler_Settings.html)

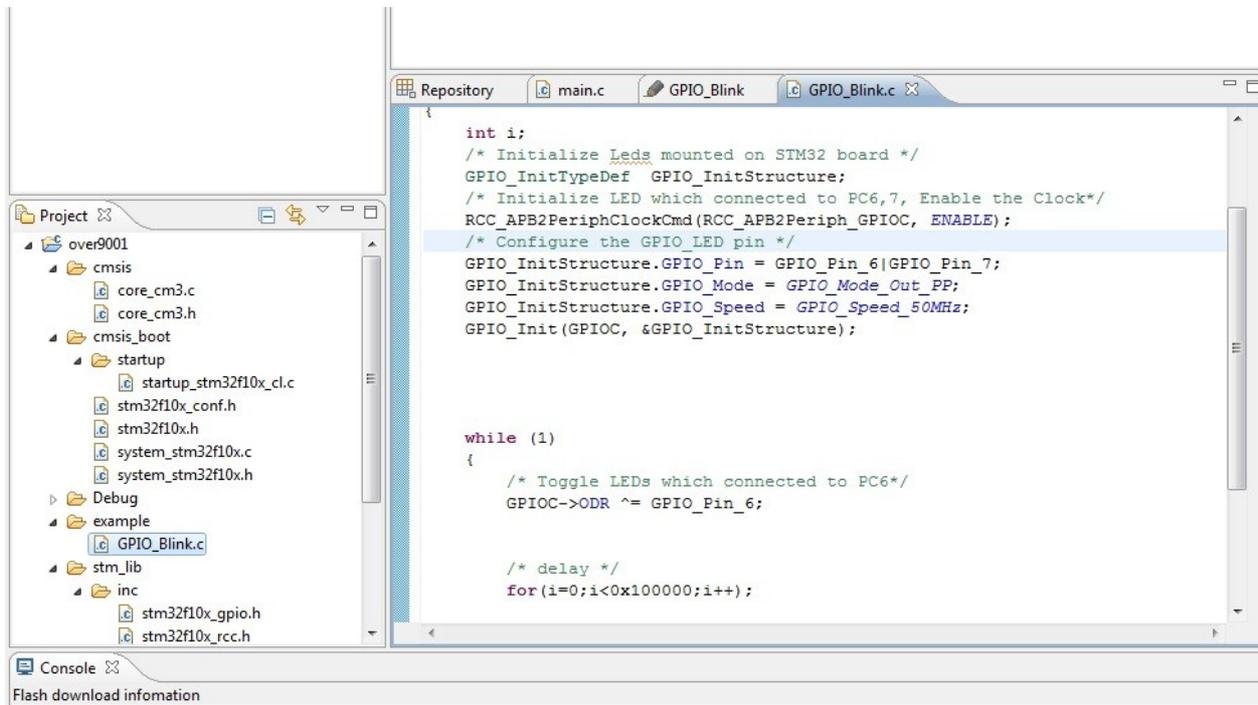
FOR NEXT CHAPTERS 3.6 AND 3.7 THERE IS A GOOD VIDEO AVAILABLE JUST REMEMBER THAT YOU HAVE TO CHANGE THE PORT FOR DEBUGGING TO 2009  
[http://www.youtube.com/watch?v=cb\\_kAUdt01w](http://www.youtube.com/watch?v=cb_kAUdt01w)

### 3.6 You need to create new project to be able to set the debug settings. For the board I have at hand STM32-H107 it is really simple to modify one of the examples CoCoX

provides. I click **Project -> New Project** and name it **over9000** then a screen named **Step 1** appears. I click **ST** and find the processor my board has. I mark **GPIO library** which will also auto mark. In the **Components** windows click on **Gpio**:



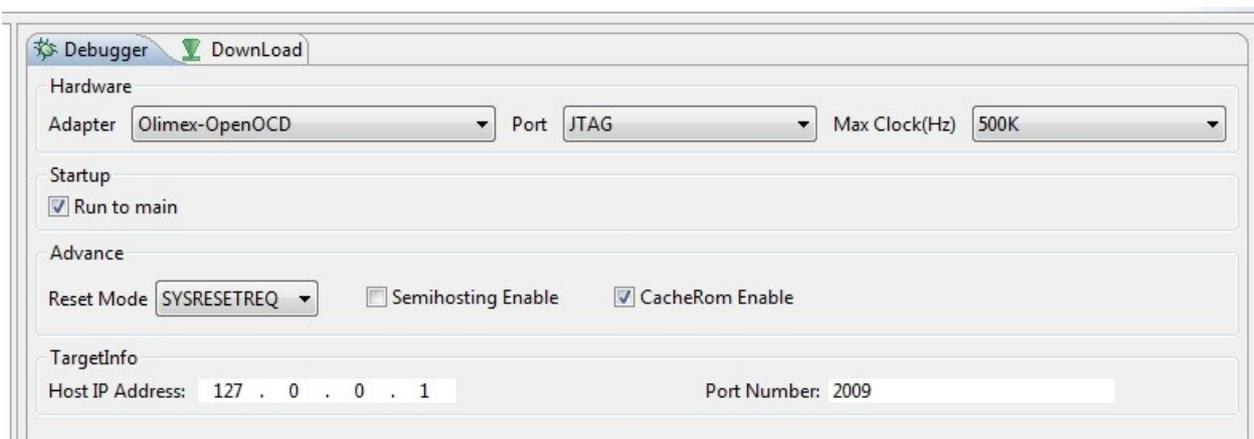
Then add the **GPIO\_Blink** from **GPIO Examples Window** and agree to be added to the project. Double-click on the added **GPIO\_Blink.c** file and edit the pin/port number according to the hardware you have while referring to the schematic of your device (or the pinout table). In my case with **STM32-H107** the two **GPIO LEDs** are on ports **PC6** and **PC7**:

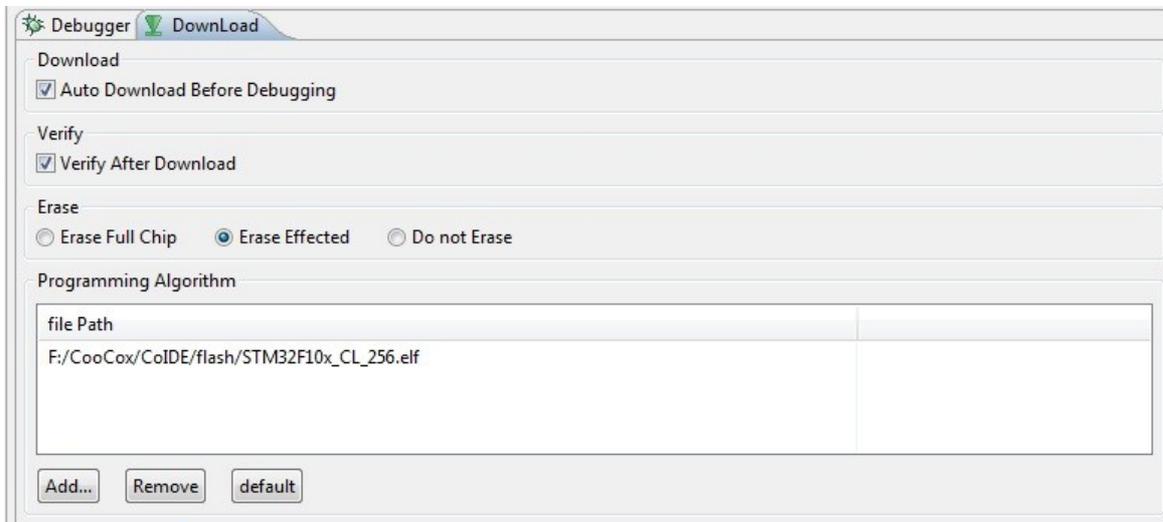


After that from Project -> Rebuild All to compile and build the project.

### 3.7 Click Debug -> Debug Configuration and set the settings in the following order:

Adapter: Olimex-Open OCD, Port: JTAG, Max Clock – 500K Hz, Startup – check Run to main, Reset Mode: SYSRESETREQ; Disable the Semi-hosting since it isn't stable yet; Enable caching for faster debugging; the host address for the GDB is local host e.g. 127.0.0.1 and the port is 2009 (!).





Voila! Now you should be able to both download to flash and debug with CooCox and Olimex Arm Debuggers.

For more information visit the CooCox web site: <http://www.coocox.org>; for questions regarding the CooCox software visit their forums: <http://www.coocox.org/forum/forum.php?id=1>

For Olimex products visit: <http://olimex.com/>