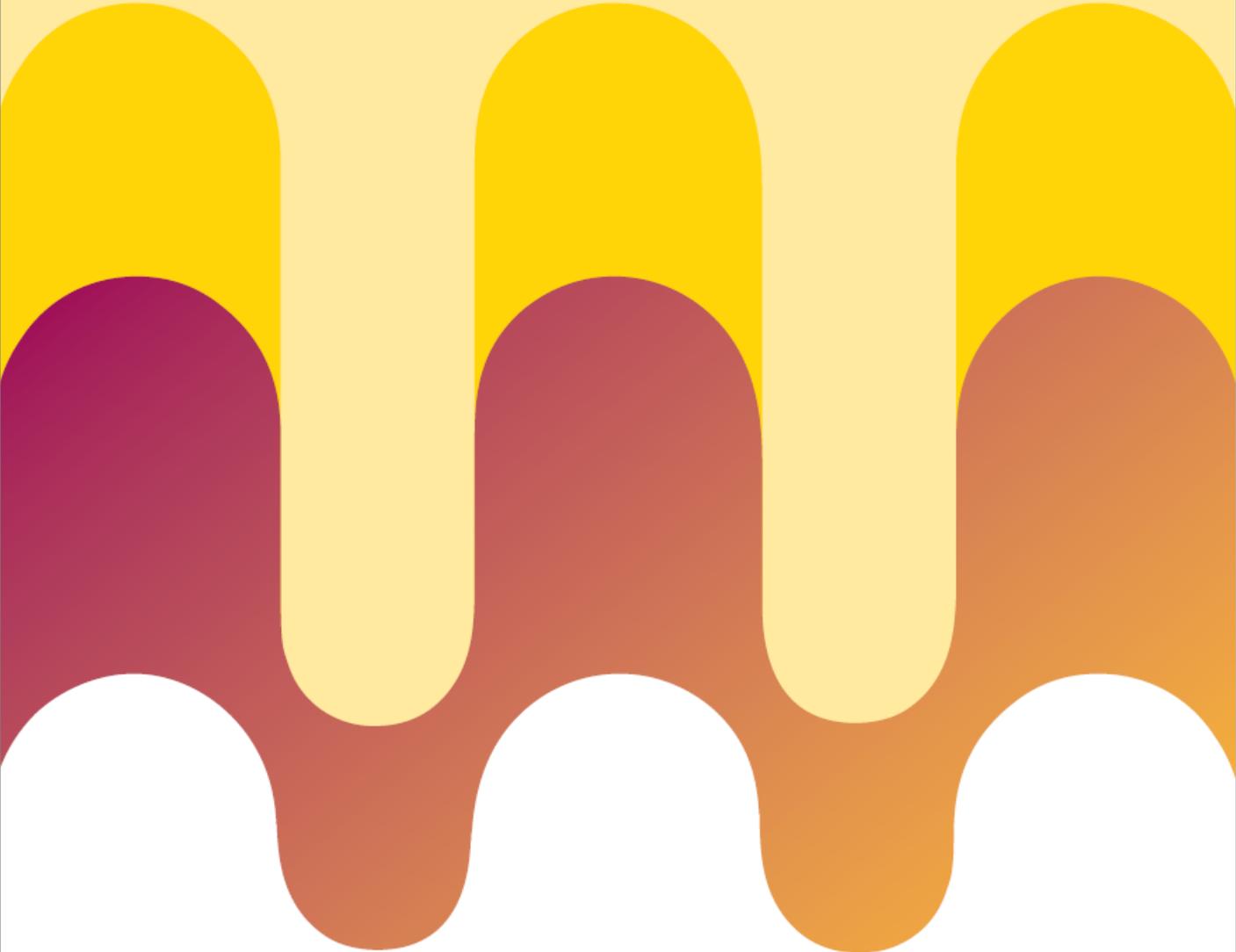


Getting started with Contiki on SEED-EYE Board

Version 0.1



EVIDENCE[®]
EMBEDDING TECHNOLOGY

Document History

| Version | Date | Author | Description |
|---------|------------|-----------------|-------------|
| 0.1 | 13/06/2013 | Errico Guidieri | First draft |
| | | | |

Tabella 1: Document History

Document References

| Reference | Title | Author | Date | Version |
|---|------------------------------|--------|------|---------|
| http://contiki.sourceforge.net/docs/2.6/ | The Contiki Operating System | | | |
| | | | | |

Tabella 2: Document References

Table of contents

| | |
|--|---|
| Document History | 2 |
| Document References | 2 |
| Table of contents | 3 |
| List of tables..... | 3 |
| Abbreviations..... | 3 |
| 1 SEED-EYE Board Overview | 4 |
| 2 Setting up a Contiki building environment for SEED-EYE on windows..... | 4 |
| 2.1 Installing Cygwin..... | 4 |
| 2.2 Microchip PIC32 Compiler and Tools | 4 |

List of tables

| | |
|--------------------------------------|---|
| Tabella 1: Document History..... | 2 |
| Tabella 2: Document References | 2 |
| Tabella 3: Abbreviations | 3 |

Abbreviations

| Abbreviation | Description |
|--------------|-------------|
| | |
| | |
| | |

Tabella 3: Abbreviations

SEED-EYE Board Overview

SEED-EYE is an advanced board for implementing low-cost multimedia Wireless Sensor Network.

It hosts a powerful Microchip PIC32, and has a full set of communication interfaces such as Ethernet, IEEE802.15.4 / Zigbee, and USB. Moreover, it hosts a CMOS Camera, making it an ideal board for implementing next generation Imaging Wireless Sensor networks.

The SEED-EYE Board comes with Full software support, including porting for [Contiki OS](#) and [ERIKA Enterprise](#).

SEED-EYE has been designed by Scuola Superiore Sant'Anna and Evidence Srl.

1 Setting up a Contiki building environment for SEED-EYE on windows

1.1 Installing Cygwin

The **Contiki** development environment under Windows uses the Cygwin environment. Cygwin is a Linux-like environment for Windows. Cygwin can be found at <http://www.cygwin.com>. Click on the icon "Install Cygwin Now" to the right to get the installation started.

Choose "Install from Internet" or download the **setup.exe** and then specify where you want to install cygwin (recommended installation path: **C:\cygwin**). Continue with the installation until you are asked to select packages. Most packages can be left as "Default" but there is one package that are not installed by default. Install the following package by clicking at "Default" until it changes to "Install":

Devel - contains things for developers (make, etc).

When cygwin is installed there should be a cygwin icon that starts up a cygwin bash when clicked on. Whenever it is time to compile and send programs to the SEED-EYE nodes it will be done from a **cygwin shell**.

1.2 Microchip PIC32 Compiler and Tools

To program SEED-EYE board you can use any PIC programmer as ICD3 or picKit3. Just be aware that the connector on the board is thought for **picKit3**, for other programmers you might need to build/buy an adapter.

To interact with the programmer you need a **Microchip IPE** (Integrated Programming Environment). The easiest way to get such software is to install a **Microchip IDE** (Integrated Development Environment).

Although the new **MPLABX IDE** is available for Windows platform too, we still suggest to use the traditional Microchip **MPLAB IDE for Windows**, the following paragraph that describe how load an image on a SEED-EYE assume to work with the latter tool.

You can find the last available one at the end of [MPLAB Integrated Development Environment](#) page.

The Contiki porting for SEED-EYE platform support **xc32** Microchip compiler. You can download the free version from the following address <http://www.microchip.com/mplabxc32windows> . The free version contains a60-day PRO evaluation that can be started at any time in the software.

1.3 Get a copy of Contiki

After setup the environment you need to get a copy of Contiki OS source. Contiki official page suggest to start with a copy of [Instant Contiki](#) that is a a VMware Ubuntu virtual machine contains the most commons tools used to develop with contiki.

But we prefer to access directly the source code downloading the last release from here:

<http://sourceforge.net/projects/contiki/files/Contiki>

or cloning their public repository in Git Hub:

git clone <https://github.com/contiki-os/contiki.git> git

2 The Contiki build system

Contiki SEED-EYE board support is fully integrated in **Contiki build system**. The **Contiki build system** is designed to make it easy to compile Contiki applications for either to a hardware platform or into a simulation platform by simply supplying different parameters to the make command, without having to edit makefiles or modify the application code.

The file example project in **examples/hello-world/** shows how the Contiki build system works. The hello-world.c application can be built into a complete Contiki system by running **make** in the **examples/hello-world/** directory.

Running **make without parameters** will build a Contiki system using the **native target**. The **native target** is a special Contiki platform that builds an entire Contiki system as a program that runs on the development system. After compiling the application for the native target it is possible to run the Contiki system with the application by running the file **hello-world.native**.

To compile the application and a Contiki system for the platform use the command:

```
make TARGET=seedeye
```

This produces a **hello-world.seedeye** file that can be loaded into an SEED-EYE board.

To compile the **hello-world application** into a loadable entity to be loaded by **contiki program loader** into a running Contiki system, use the the command;

```
make TARGET=seedeye hello-world.ce
```

To avoid having to type **TARGET=** every time make is run, it is possible to run:
make TARGET=seedeye savetarget

to save the selected target as the default target platform for subsequent invocations of make. A file called **Makefile.target** containing the currently saved target is saved in the project's directory.

3 Loading the Program on SEED-EYE

As said before, to load a program on SEED-EYE board you should rely on **MPLAB IDE**.

Start the tool and import the product a previous built artifact selecting **import** action from **file menu**. To make this step easier you can rename the artifact extension from **.seedeye** to **.elf**.

Select the right MCU with the **Select Device...** action device from **Configure menu**. SEED-EYE MCU device is: **PIC32MX795F512L**

Select the right programmer with **Select Programmer** action from **programmer menu**, e.g. **PICKit3**

Now you are ready to flash the board just clicking on **program button** on **toolbar**