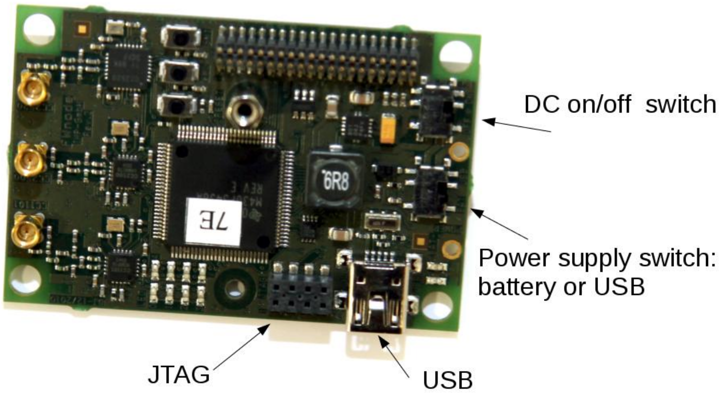

Figure 1: FWnode

1 FeuwerWhere motes programming

1.1 Introduction

The FeuwerWhere sensor node (FWnode) is based on the MSP430 micro-controller (MCU). In short, this MCU includes both RAM and flash memories, and programs are stored in flash. Therefore, to write new software to FWnodes, the flash memory must be overwritten. There are two common ways to write programs to the flash memory of FWnodes: USB and JTAG.

To write the flash memory, the FWnode must be powered up. FWnode can either be powered from an external battery or from the USB port. There is a switch close to the USB port that selects the power source: battery of USB (see Figure 1). Another switch (DC ON/OFF) enables the feature of providing 3V to the MSP430, even when the battery voltage is below 3V. However, it results in extra power consumption.

1.2 USB interface

Software can be written to the flash memory of the FWnode using the USB interface. To do this you need to connect the FWnode to a computer with an USB cable.

You will also need an extra adapter for USB programming (see Figure 2). In short, to start USB programming the MSP430 must enter BSL (Bootstrap Loader) mode. To enter this mode, the BSL lines must be switched very quickly, and it is not possible with this USB configuration. Therefore, the extra adapter includes another MSP430

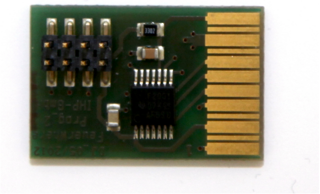

Figure 2: Adapter for flashing FWnode via USB

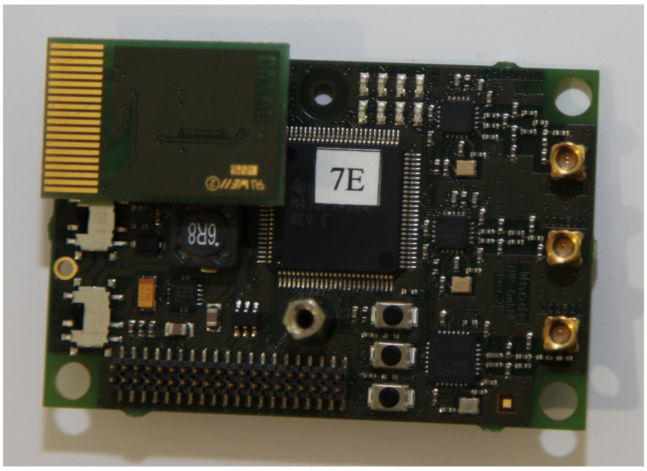

Figure 3: FWnode with the adapter for USB programming

that monitors these BSL lines and switches them quickly enough to enter the BSL mode. FWnode with the adapter is showed in Figure 3.

To program FWnodes from a PC you will also a Python script for loading HEX binary files into the flash memory of FWnode. The same Python script works on both Linux and Windows, and was tested with Python 2.7. When running the script you have to give the port the FWnode is connected to, and the filename of the image. For example, the following command write *image.hex* to the FWnode connected to the port */dev/ttyUSB1*:

```
mb_program.py -p /dev/ttyUSB1 -f image.hex
```

Please ensure that FWnode is powered with battery or USB (power supply switch in Figure 1).

1.3 JTAG interface

Programs can be written to the flash memory with the JTAG interface (see JTAG socket in Figure 1). You will an extra interface (cable) to connect a PC with the FWnode. We used the model MSP-FETU430IF of Texas Instruments but it is not available any more.

There is a newer version at Texas Instruments but it does not work with the older Code Composer version (version 5), which we mainly use here.

JTAG interface can also be used to debug software running on FWnodes (set breakpoints, read registers, values, etc.). Both features - programming and debugging - can be easily done in the Code Composer software of Texas Instruments, which is based on the Eclipse platform. As we experience some problems with the JTAG interface under Linux, we use Code Composer for Windows operating system.

1.4 Example (USB programming)

To check if USB programming works, you may try out an example with files attached to this tutorial: *mb_program.py* and *read_id.hex*. The former is the Python script used for flashing FWnode via USB. The latter is an image file, which prints every second the unique identifier of the FWnode and flashes the blue LED. Here are the steps:

1. Put the USB adapter (Figure 2 and Figure 3) onto the FWnode

2. Connect the FWnode to the PC with an USB cable

3. Find out the port name the FWnode is connected to (/dev/ttyUSBn in Linux and COMn in Windows). In Windows you may need to install FTDI drivers:
   http://www.ftdichip.com/Drivers/VCP.htm

4. Ensure you have Python installed (prefereably version 2.7, but it may work with other version as well). You will also *pyserial* Python package
   (https://pyserial.readthedocs.org/en/latest/)

5. Program FWnode with the following command:
   python mb_program.py -p PORT_NAME -f read_id.hex

6. While the FWnode is being programmed, you should see two LEDs close to the USB port at the FWnode flashing. Once a while you get an error just after issuing the command. In this case, just try again.

7. After finishing, the FWnode should flash its LED and print its ID on the USB port. You can read the id of the node by reading the USB port (115200 bps), which works as a serial port. For example, you can use our *terminal.py* program to read it, or any other terminal program.