Versatile Sensor Node (VSN) platform - design of hardware and software

Marko Mihelin

AgroSense

Introduction

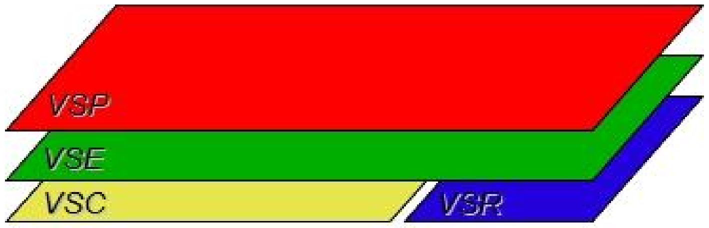
- A new WSN node - The Versatile Sensor Node
- Designed and developed in collaboration with ISOTEL d.o.o.
- Multiple purpose design, can be used in many different applications
- Open source software

Concept of the Versatile Sensor Node

- Modular base, which can be quickly adapted to various applications
- Powerful micro-controller with a ARM Cortex-M3 core
- Supports basic digital interfaces such as I^2C, SPI, UART, IrDA and analog interfaces ADC and DAC
- Dedicated radio module that supports ISM band frequencies (315 MHz, 433 MHz, 868 MHz, 915 MHz and 2400 MHz), Xbee and Bluetooth modules
- Various options for powering the nod

VSN Modules

- VSN comprises of at least two different modules
- The core module – VSC supports the micro-controller and power supply
- The radio module – VSR houses the low power radio interfaces
- The expansion modules – VSE



- The power module – VSP provides additional power supply options

VSC Module

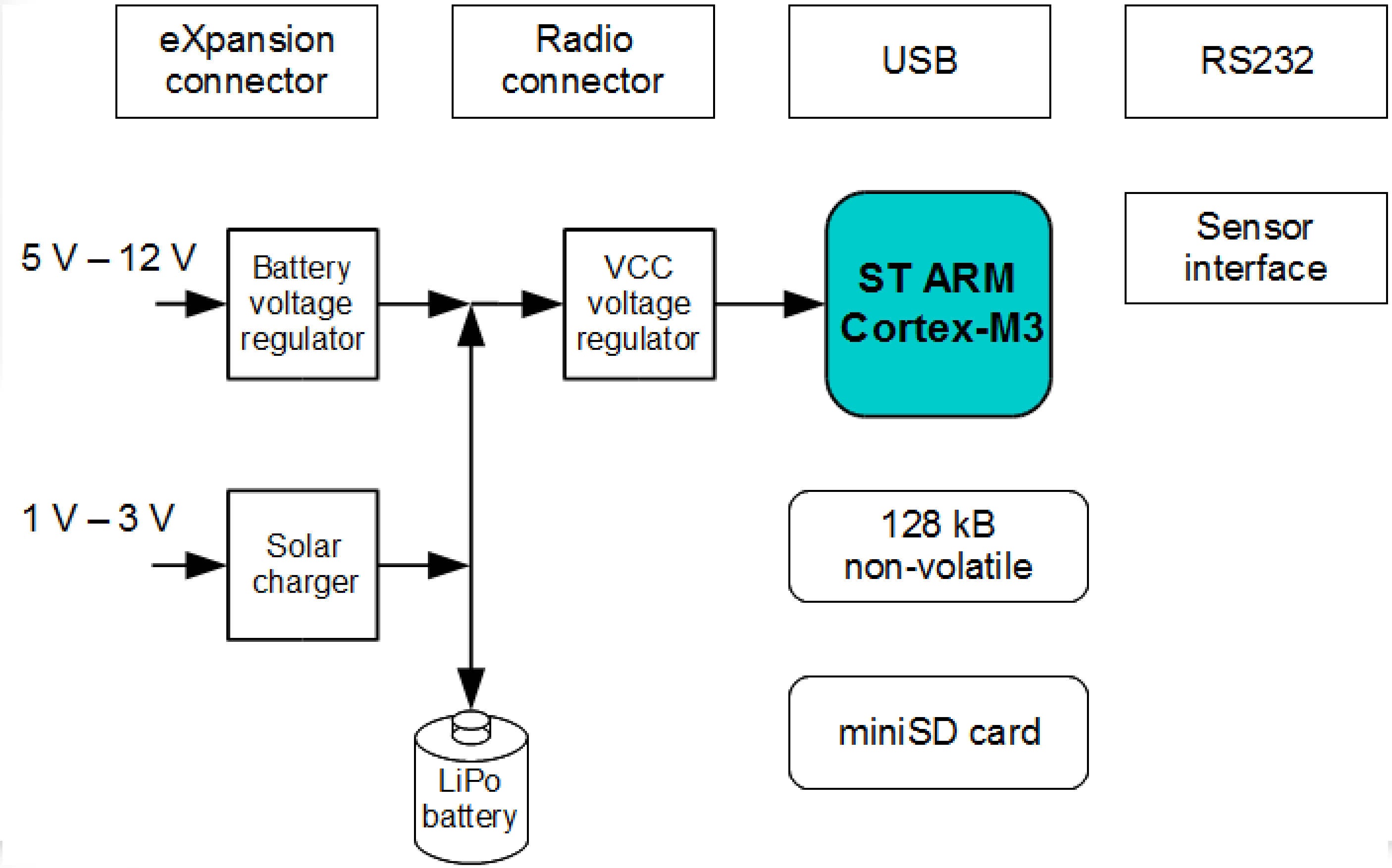
- Used micro-controller is STM32F103 with 512 kB flash, 64 kB RAM and maximum clock of 72 MHz
- Additional memory 128 kB FRAM, miniSD card
- Supported interfaces USB, RS232, UART, IrDA, SPI, I^2C, 12 bit ADC, DAC, instrumental differential amplifier with settable gain
- Selectable system voltage (3,3 V, 3 V, 2,3 V)
- Battery charger and solar battery charger with MPPT (Maximum Power Point Tracker)

VSC Module power states

- The micro-controller has a few standard low power states
- Supported low power modes: Sleep, Stop and Standby
- Additional low power mode : Deep Hibernation
- Power consumption in deep hibernation is below 7 µA
- The node can be woken from deep hibernation using RTC alarm
- System state can be recalled from external high speed non volatile memory

Jožef Stefan Institute
Department of
Communication Systems

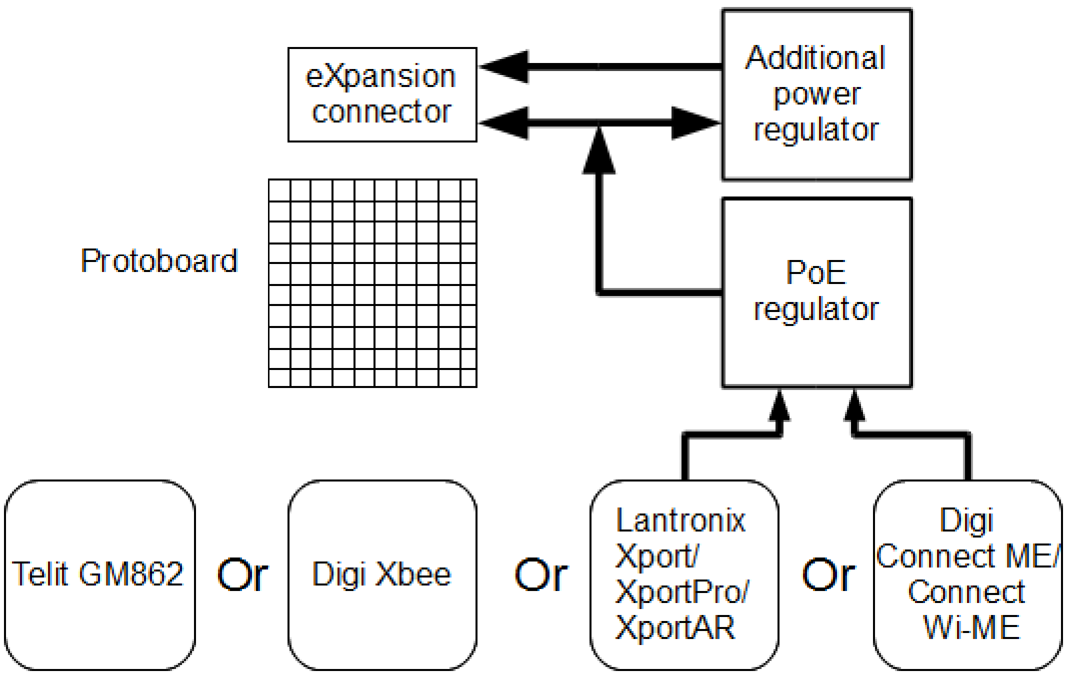
VSC module block schematic



VSE Module

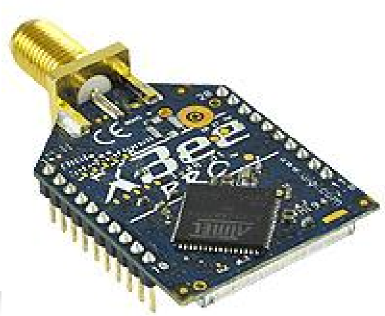
- Communication expansion module which supports:
  - Serial to Ethernet (Xport, Xport Pro, XportAR, Connect-Me) converters from Lantronix and Digi with PoE power supply
  - Serial to Wi-Fi (Connect-Wi-Me) converter from Digi
  - Telit GM862
  - Xbee or Bluetooth modules
- Development module with a standard 20 pin JTAG connector, and small protoboard
- Full size protoboard for application testing

Communication VSE module block schematic



VSR Module

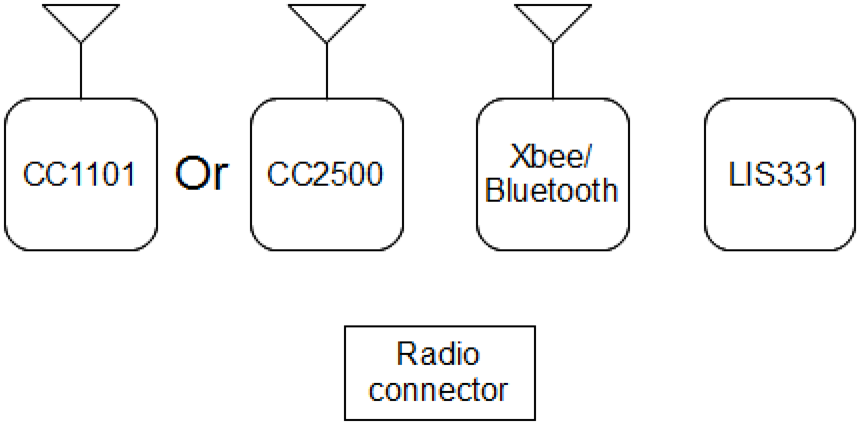
- Low power radio module
- Supports ISM band frequencies from 315 MHz to 2,4 GHz
- Sub GHz frequencies CC1101 from Texas Instruments
- 2,4 GHz frequency band CC2500 from Texas Instrument
- Optionally a Xbee or Bluetooth module can be attached



AgroSense

Jožef Stefan Institute
Department of
Communication Systems

VSR Module block schematic



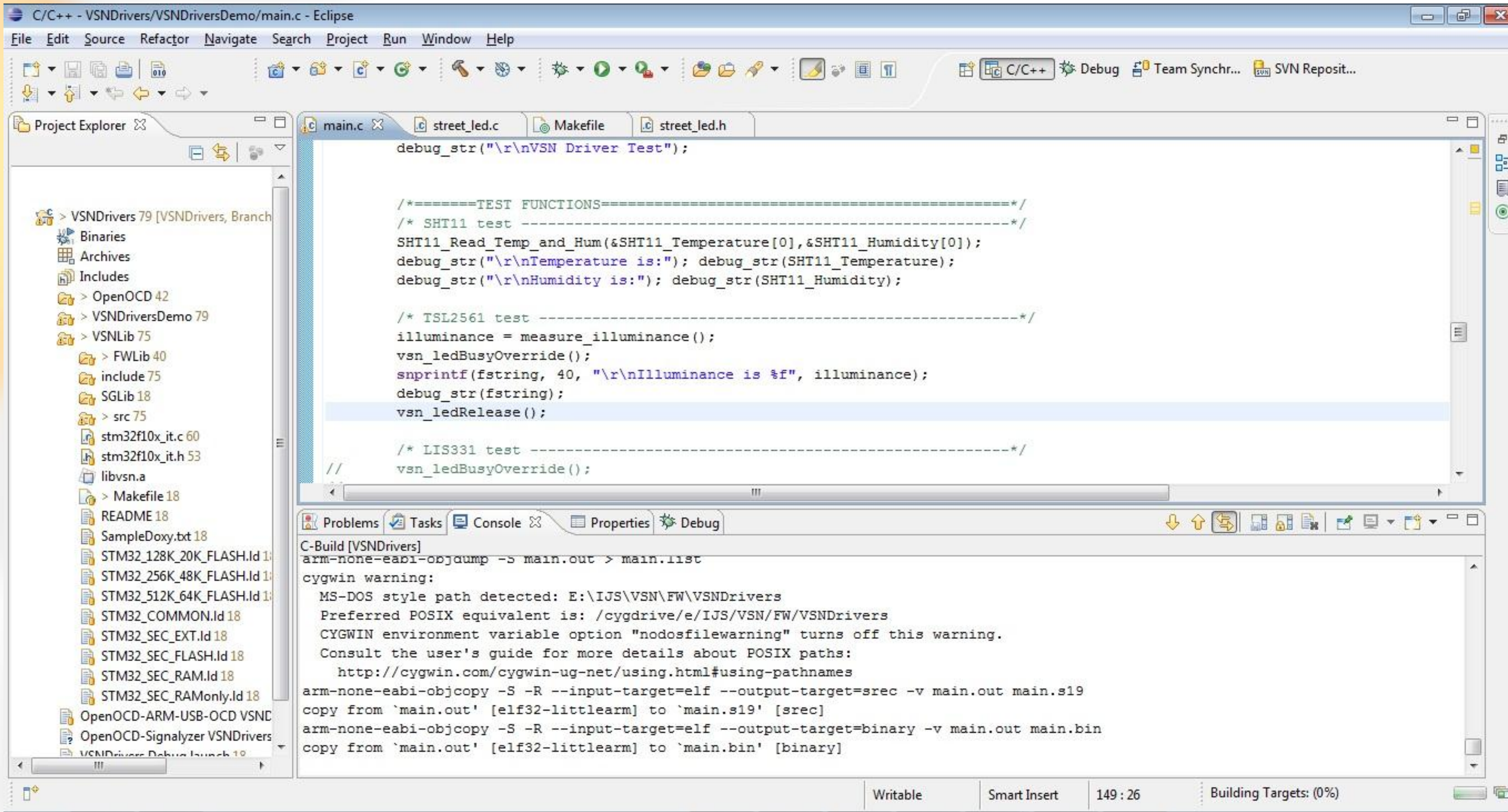
Tools and principles used in VSN design process

- For developing two layer PCBs we used CadSoft Eagle Student edition
- For hardware simulation we used SpiceOpus, a free Spice package
- Most of circuits can be designed with the help of application notes and fine tuned with simulation and trial and error

Software development tools

- Professional
  - Keil, full development system
- Open Source
  - Development environment: Eclipse IDE
  - Tool-chain: Sourcery G++ Lite
  - Cygwin, Linux environment for Windows
  - JTAG server: OpenOCD
  - JTAG hardware interface: Olimex ARM-USB-OCD

Eclipse IDE C/C++



Project build system

- Fully customizable build system based on makefiles
- Requires knowledge of how the building system works
- Provides full control of the build process

http://imgs.xkcd.com/comics/compiling.png

The ST Standard Peripheral Library

- We use the ST Standard Peripheral Library V3.1.2
- The library deals with register and bit peripheral access so we don't have to
- The library provides macros for various register addresses and states
- All other software including the device drivers are build on top of this library

Driver development general pointers

- The STM32F103 Reference manual is your best friend
- The sensor/device datasheet is your second best friend
- If your best friends fail to provide you with enough information ask Google
- Have patience

Driver development procedure

1) Identify the sensor/device interface
2) Connect the sensor/device to appropriate pins on VSN
3) Initialize the identified clocks and peripherals of the micro-controller
4) Write a test driver to check if all is working correctly
5) Complete the driver, use interrupts and DMA to maximize the performance

Thank you for your attention.

Questions?

AgroSense

Jožef Stefan Institute
Department of
Communication Systems