A WEARABLE MUSICAL INTERFACE USING A WIRELESS SENSOR NETWORK

Francesco Corucci – f.corucci@gmail.com – 2012

Mobile and pervasive systems class – MSc in Computer Engineering

Introduction

- **Goal**: implement a platform that allows the use of a wireless sensor network as a **wearable interface** that can be used for **producing music**
- **Requirements:**
  - Modular
  - Flexible
  - MIDI-compliant

What is MIDI

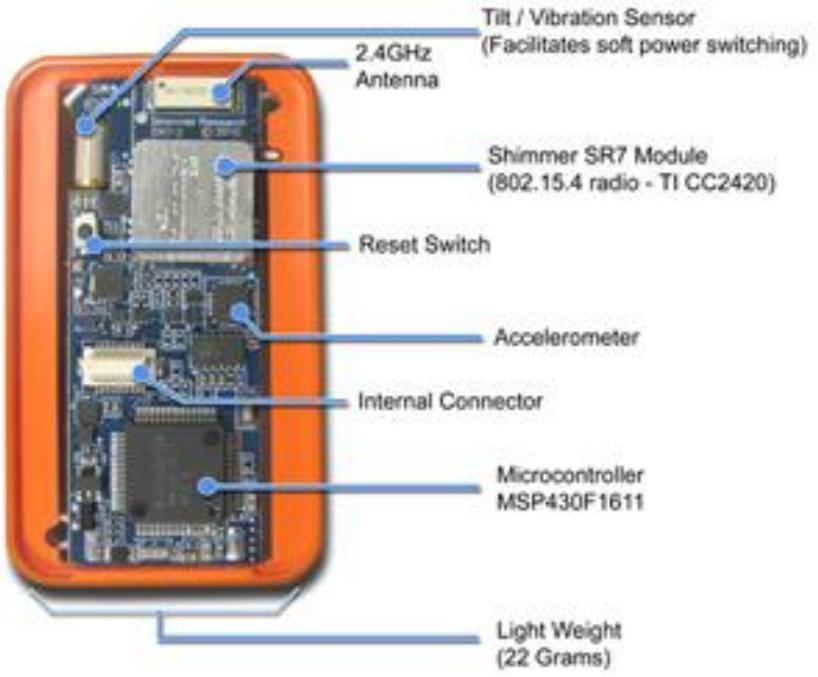
- **MIDI** *(Musical Instrument Digital Interface)* is a standard technology for the **interaction between musical devices**
- It standardizes:
  - A communication protocol
  - An hardware interface
- Versatile technology, widely used since the 80's and still reliable after more than 30 years

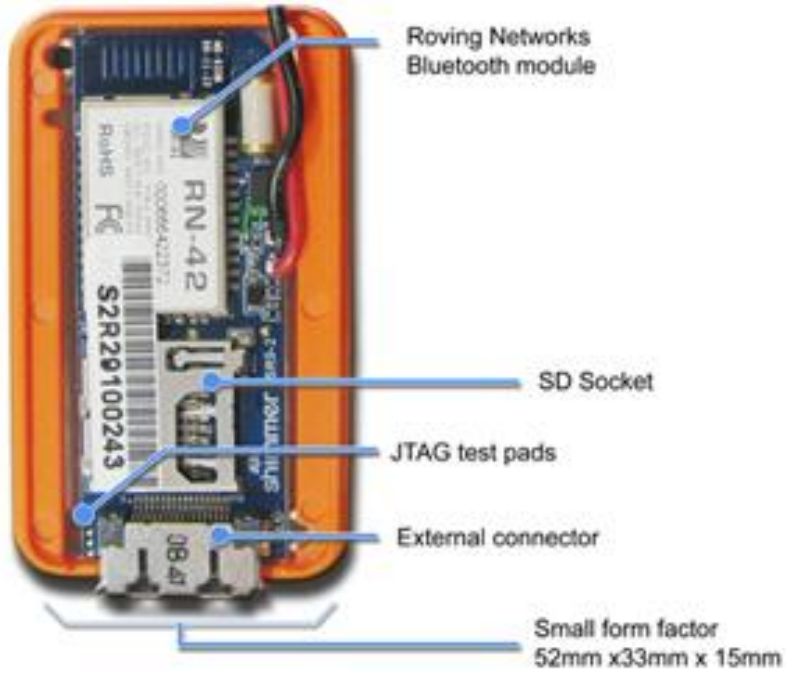
Shimmer sensor nodes

- For setting up the WSN we choose ***Shimmer*** sensor nodes :
  - Suitable for wearable applications
  - Integrate a **3-axis accelerometer**
  - Optional gyroscope shield (not used in this project)
  - Bluetooth or 802.15.4 communication
  - Support **TinyOS**

Shimmer sensor nodes

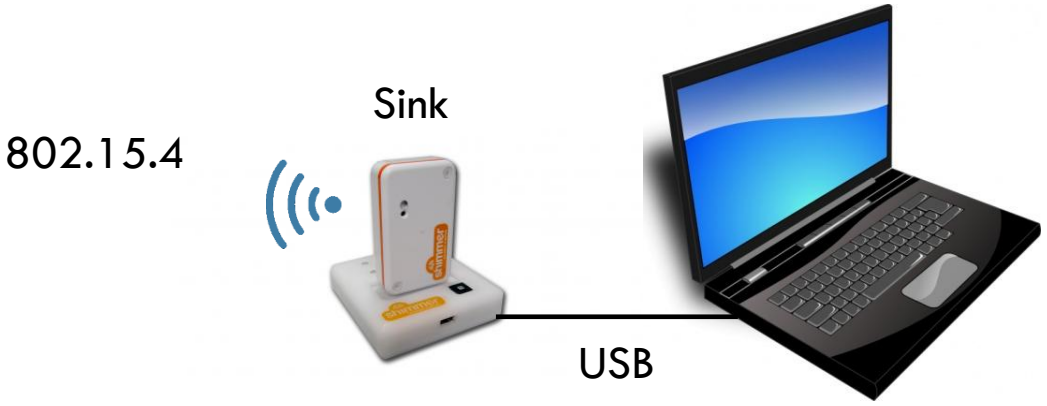
shimmer
DISCOVERY IN MOTION



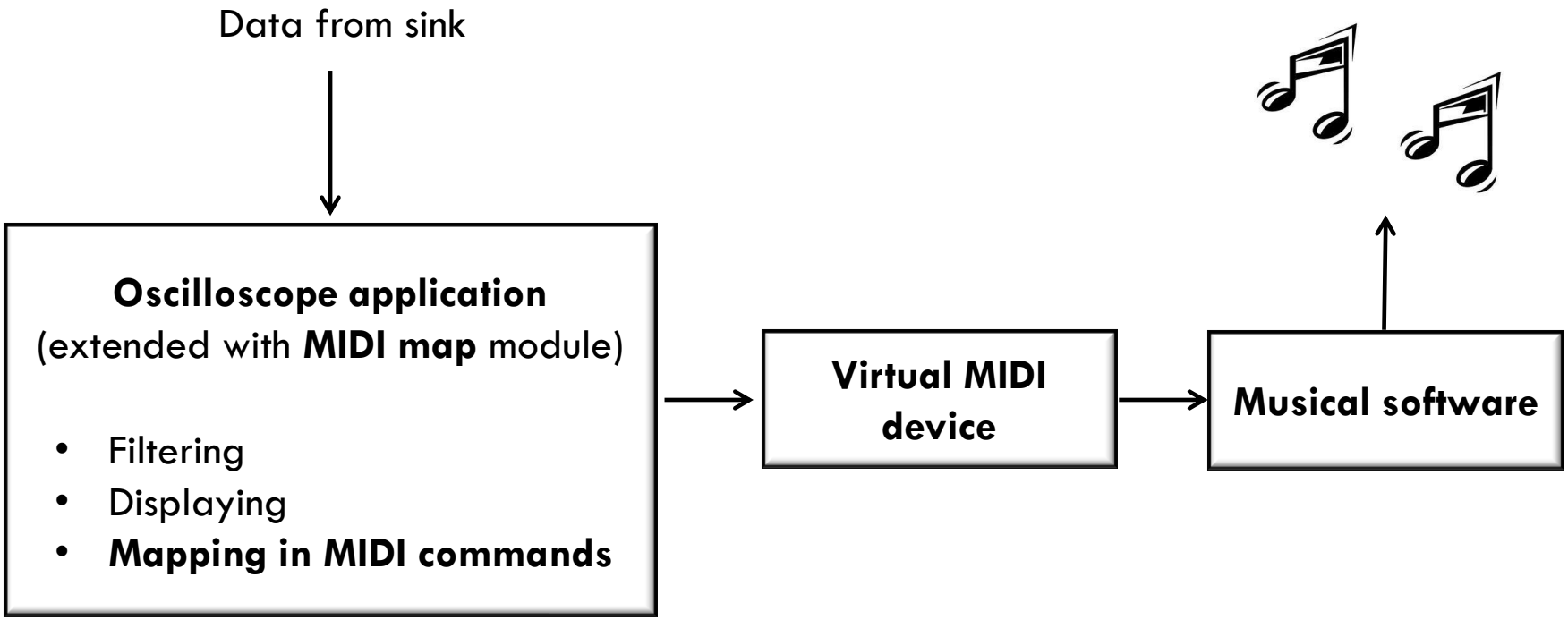


Overview of the system

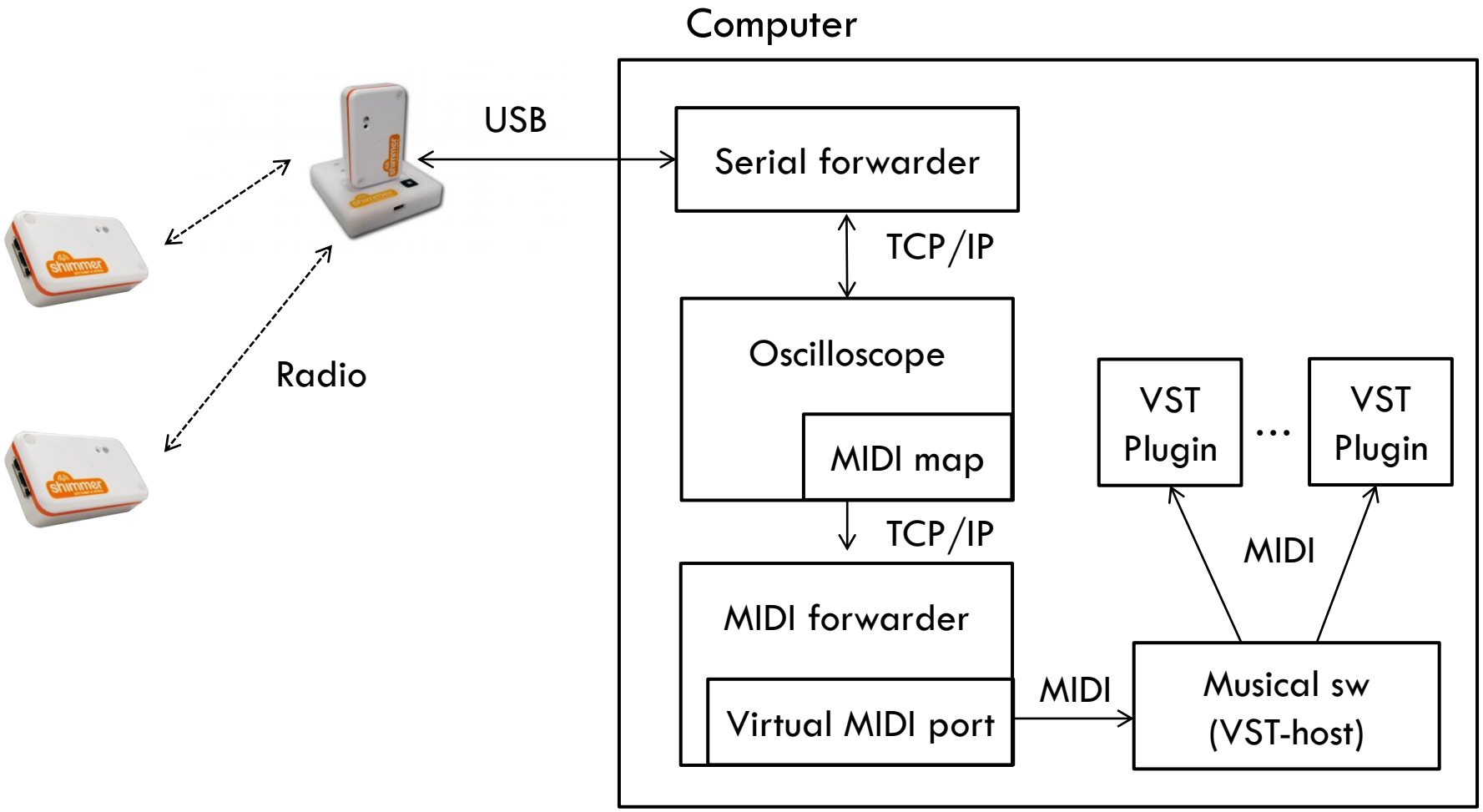
- Two or more wireless sensors on the body
- The sensors sample accelerometric data associated with movements
- Data are then collected from the sink and forwarded to the computer



Overview of the system (cont'd)



Detailed architecture



MIDI controller

This way the WSN become a standard **MIDI controller,** that can interoperate with every musical software

How to choose the mapping

- The way to map sensed data into MIDI messages is absolutely arbitrary, and can lead to a **multitude of expressive results**
- Every data channel can be mapped in a different
  - instrument
  - effect
  - note/chord
  - etc…
- …and this can be done in a variety of ways!

Demonstration time

Features

- **Modular, extensible:**
  - Is it possible to use an arbitrary number of sensor nodes, and choose which on-board sensors must be used as data sources
- **Flexible**:
  - Many degrees of freedom in the mapping
    → Many expressive possibilities
  - Sound generation is left to specialized and mature softwares
- **Standard:**
  - Since the WSN becomes a MIDI controller, it is able to interoperate with a moltitude of software platforms

Possible applications

The platform can be used for several purposes:

- New expressive forms based on gestuality
- In combination with traditional instruments
  (e.g. influence of on-stage movements on played music)
- Any application that asks for *sonification* of human movements:
  - Sport training
  - Rehabilitation
  - …

References

- Shimmer sensor nodes
  - http://www.shimmer-research.com/
- General information about MIDI
  - http://en.wikipedia.org/wiki/MIDI
- VST
  - http://en.wikipedia.org/wiki/Virtual_Studio_Technology
- rtMidi, a cross-platform C++ library for MIDI
  - http://www.music.mcgill.ca/~gary/rtmidi/
- Sonification
  - http://en.wikipedia.org/wiki/Sonification

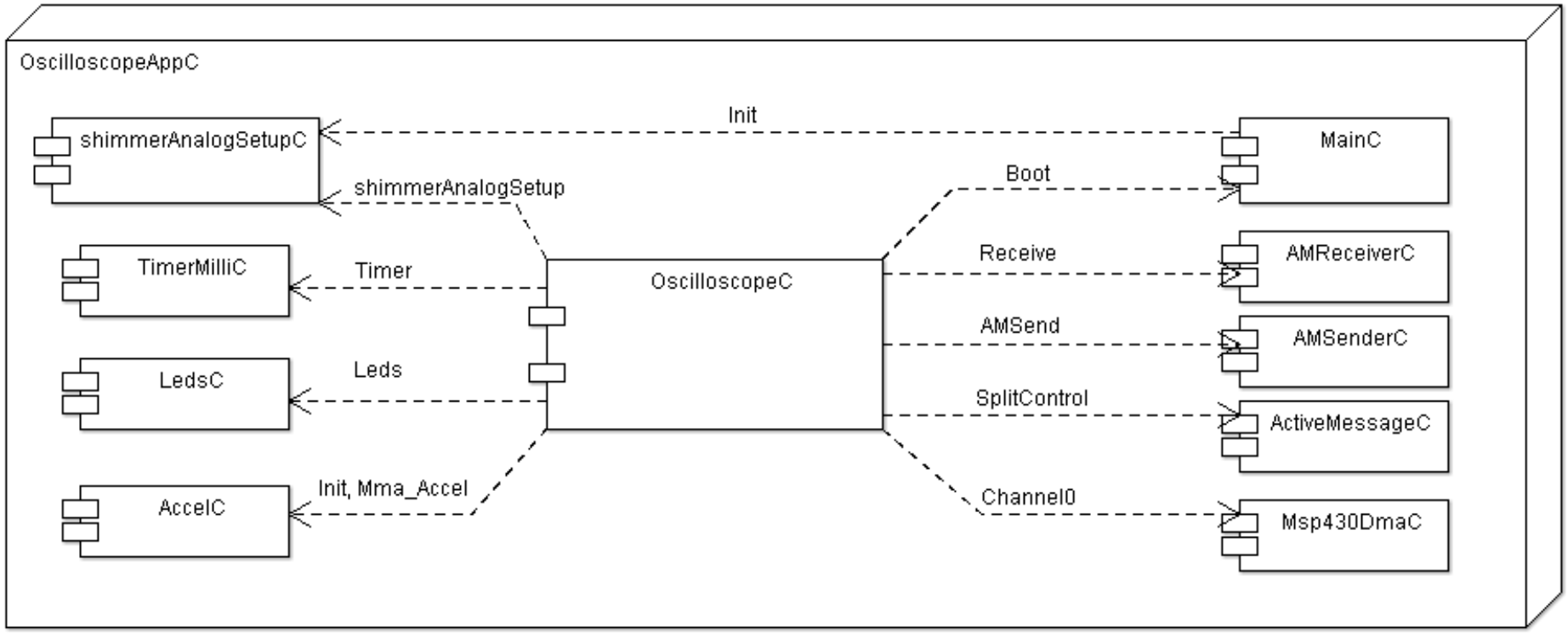
Thank you for your attention :-)

Implementation details

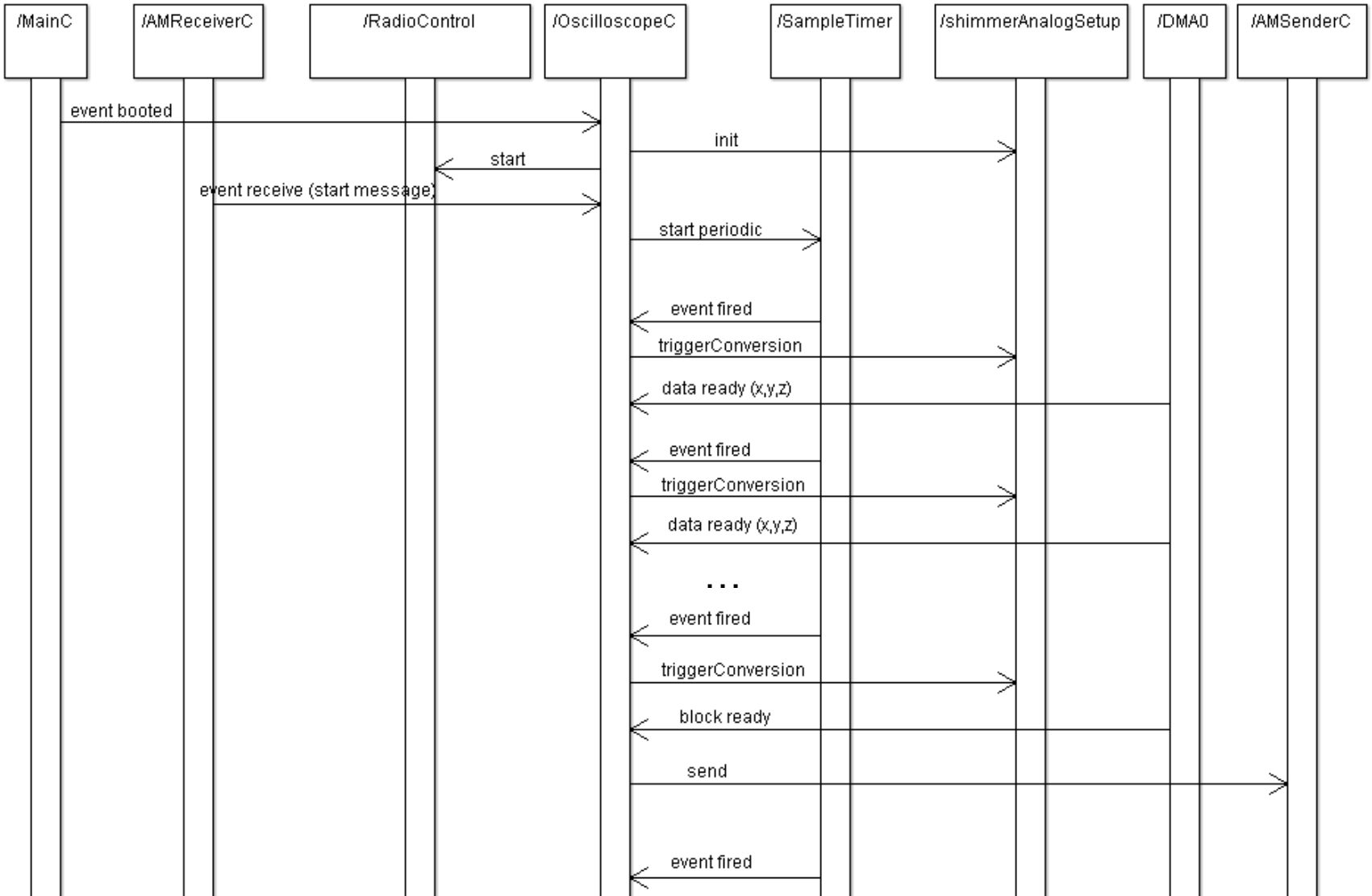
OscilloscopeAppC - overview

- *OscilloscopeAppC* is the TinyOS application running on sensor nodes
- Receives commands from the sink: start/stop/change_freq
- Samples the on-board tri-axis accelerometer using three ADC channels in DMA mode
- When a block of samples has been collected, the sensor node sends it to the sink via radio

OscilloscopeAppC - architecture



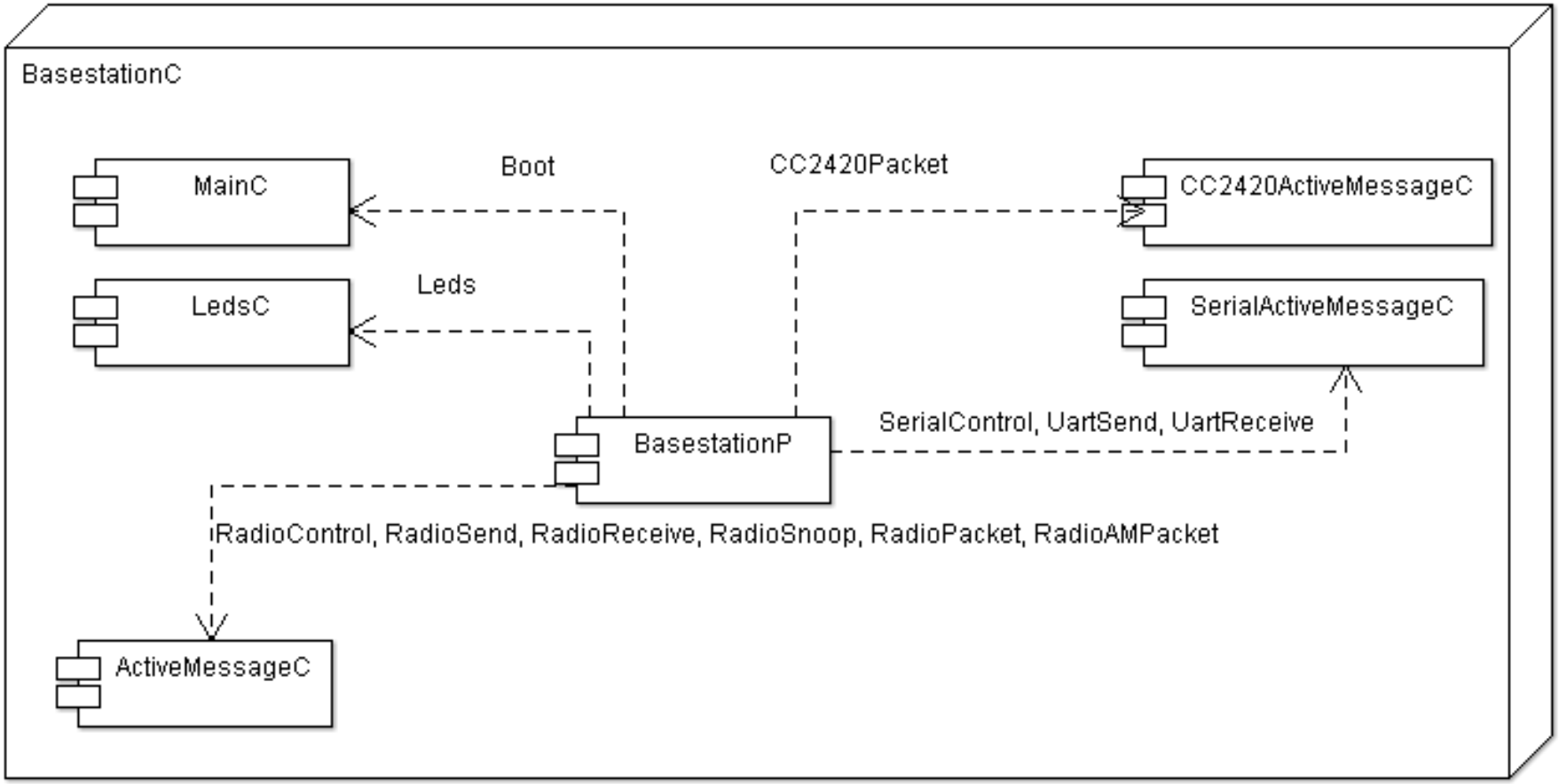
OscilloscopeAppC - interactions



BasestationC - overview

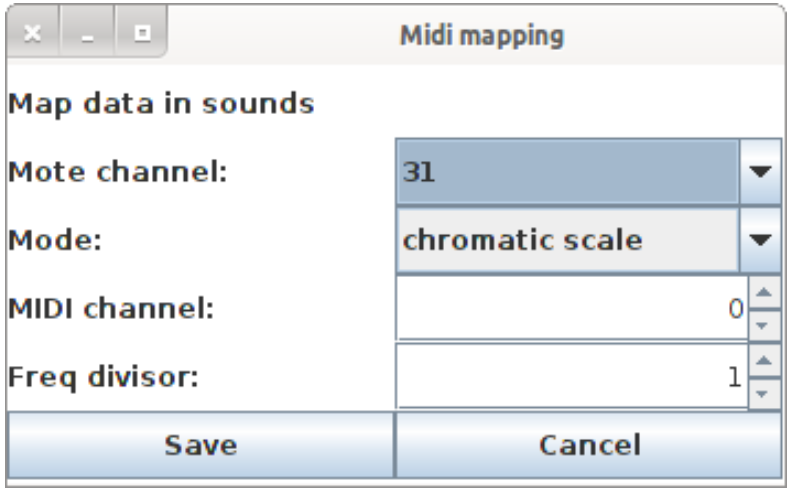
- *Basestation*C is the TinyOS application running on the sink node
- Acts as a **bridge** between the serial and the radio link
- Implements queues in both direction to handle traffic spikes

BasestationC - architecture



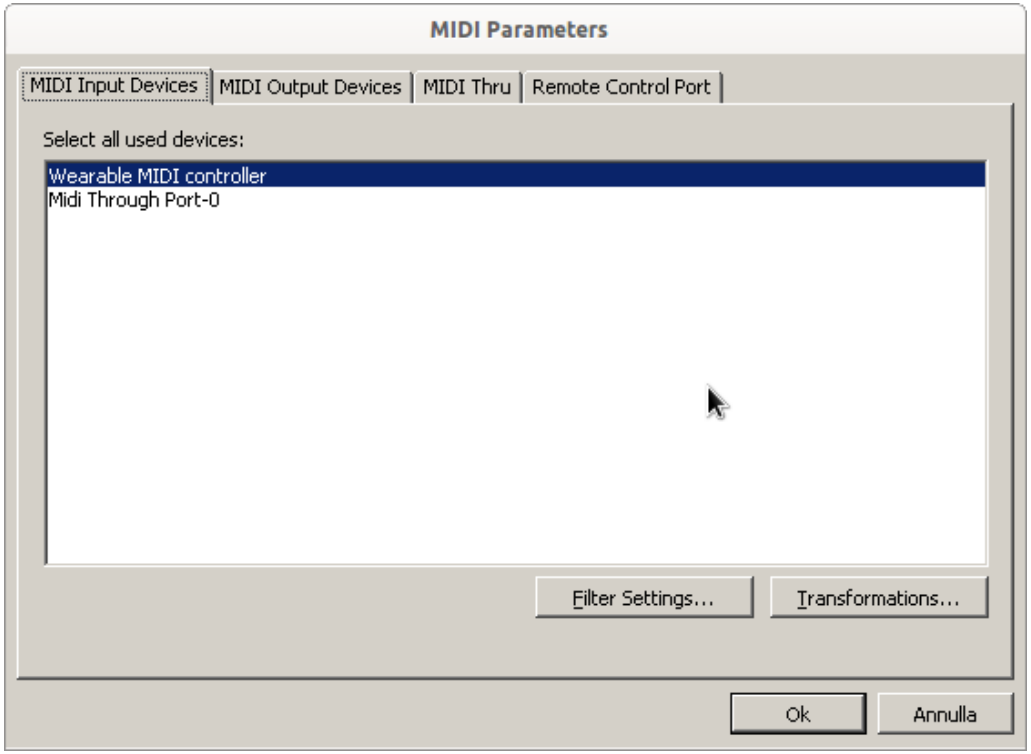
MIDI map module

- Offers a simple GUI: the user can choose how to map the active data channels into MIDI messages from a predefined set of mappings
- The module then catches incoming data, maps it into MIDI command according to user-defined preferences, and forwards them via TCP/IP socket toward the MIDI forwarder

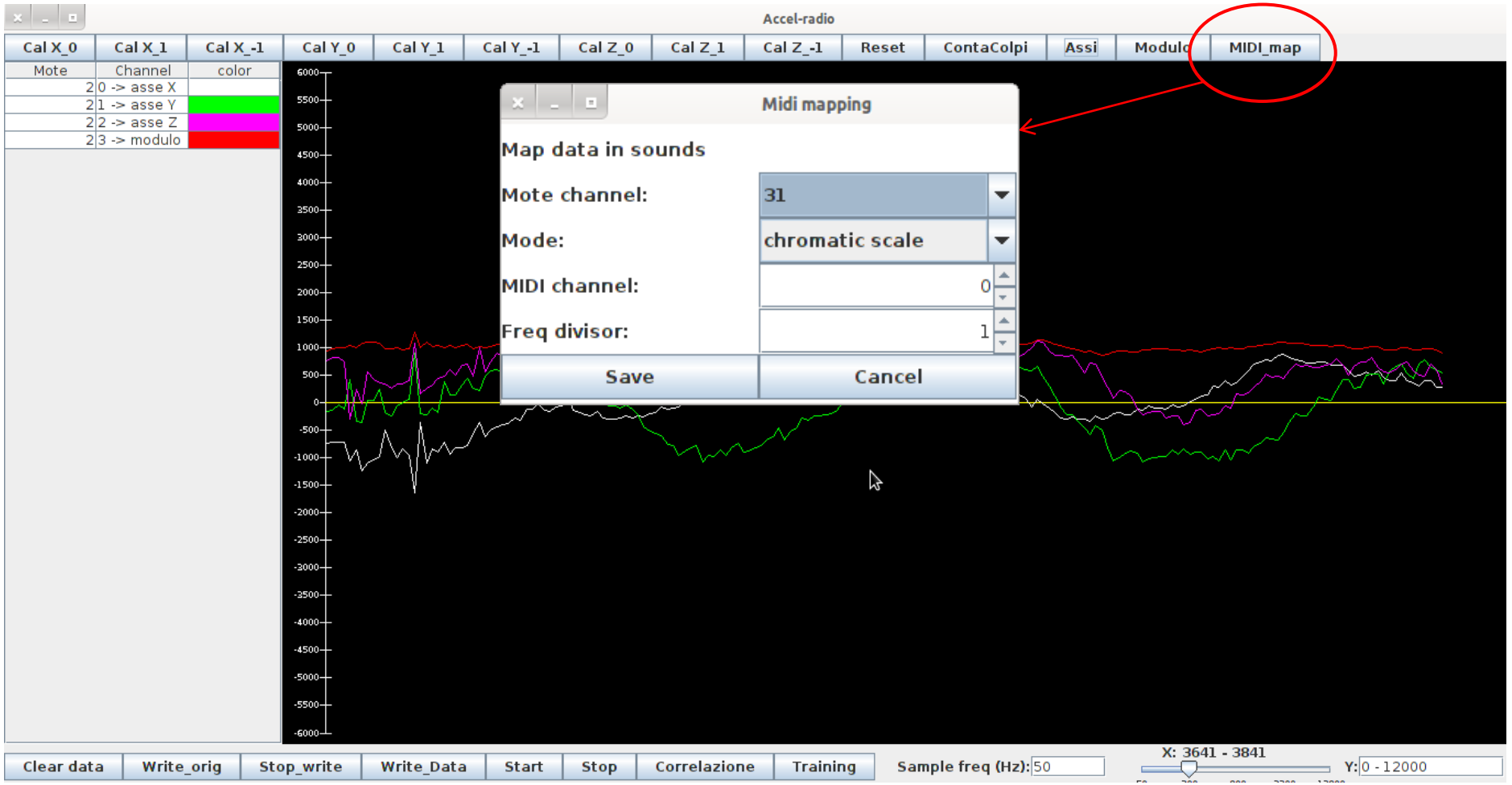


MIDI forwarder

- Implements a virtual MIDI device: the WSN is seen as an input MIDI peripheral plugged into the computer
- It then acts as a bridge:
  - Waits for incoming MIDI messages from a TCP/IP socket
  - Forwards MIDI messages to the virtual MIDI port



Screenshots: Oscilloscope + MidiMap



Screenshots