



Introduction to Movesense Programming

14.02.2019

Petri Lipponen

Content

- Introduction
- Movesense system overview
- Movesense sensor overview
- How to start developing?
- Data acquisition
- Sensor simulator
- Sensor programming basics
- Sensor programming: DEMO
- Wrap-up

Introduction – Petri Lipponen

- "Who am I?"
 - Lead Software Engineer in Movesense team
 - 14 years as a SW consultant
 - 3 years developing Movesense sensor software
- Movesense SW Development Team:
 - Internal Suunto team since June
 - Testing, Mobile and Embedded development in-house

Movesense System Overview

- Movesense sensor
- Sensor software platform
 - REST like sensor API's
- Mobile connectivity solution for iOS & Android
 - Easy to access all sensor features via same API's
 - JSON & REST
- Support for other frameworks
 - Unity3D
 - Xamarin

Movesense Sensor Overview

- CR 2025 Coincell battery
- 64MHz NordicSemiconductor MCU (RAM: 64kB, FLASH 512kB)
- Data memory: 384kB
- 9-axis IMU (Accelerometer, Gyroscope, Magnetometer)
- Temperature measurement
- Maxim ECG Analog Frontend (ECG, HeartRate, RR-intervals, stud contact detection)
- Maxim 1-wire Master
 - Smart connector detection
 - 1-Wire communication support (coming in v1.8)

How to start developing?

- *Have a clear idea on what you want to measure*
- Get to know the sensor using the mobile "Showcase App"
- Record some data and take a good look at it
 - What does it show?
 - What does it not show?
 - Noise signals?
- Make a simple mobile software that "does the trick"
 - Easier to debug & find coding help!
- Start development on the simulator and only when that works continue on the sensor device

How to start developing?

DEMO

<https://bitbucket.org/suunto/movesense-docs/wiki/Home>

Data acquisition

- Understand what you are measuring
- Speed & capacity limits by sensor, datamemory and BLE connections!
 - Sensor: G-ranges, sensitivity, etc.
 - Data memory: 400kbps bandwidth, limited size
 - BLE: 12kB/s theoretical maximum
- Measure with Showcase App if possible, using DataLogger as a fallback (Android *samples/DataLoggerSample*)

Data aquisition

DEMO

Sensor Simulator

- "Movesense sensor software on Windows & Visual Studio"
- Easier debugging and faster development cycle
- Simulated sensors with data import
- Whiteboard communication using *wbcmd.exe*

- Limitations:
 - No BLE
 - No Mobile communication
 - Not 100% accurate

Sensor Simulator

DEMO

Sensor programming basics

- C/C++ with some limitations:
 - No dynamic memory (there is but...)
 - No STL
 - Limited resources (thread stack, cstack, RAM)
- Asynchronous API's
 - Code *MUST NOT* hog the execution => No busy-loops!
 - Automatic power optimization
 - Call – callback structure
- REST-like with some additions (Publish-subscribe pattern)

Sensor programming basics: *Whiteboard*

- Services, clients, timers, threading and external communication
- ExecutionContext: Whiteboard threads
- LaunchableModule: "wb-aware module"
 - Runs in an ExecutionContext (WB thread)
 - Lifecycle callbacks (initModule, startModule, stopModule, deinitModule)
- ResourceProvider: WB REST service
 - API defined using Swagger 2.0 notation (yaml-file)
 - Request callbacks: onGetRequest, onPutRequest,...
- ResourceClient: WB REST client
 - Make requests to internal and external whiteboard services
 - Request methods: asyncGet, asyncPut,...

Sensor Programming: Demo

- What: Simple app that blinks when shaken
- Plan:
 - Measure "shaking" with Showcase app
 - Insert data to simulator
 - Create software that measures data and blinks if shaking using simulator
 - When works, try on sensor

.... Let's see how it goes 😊

Measuring data

- Connect to sensor
- Open Accelerometer
- Find saved data and save it to simulator build folder

Prepare data for simulator

- Edit so that the data is in correct format:
 - Comma (,) separator
 - Period (.) as decimal
 - Header row with *LoopingTimestamp* (optional)
 - 1st column is "Timestamp" (ms since start of the sensor)
 - Rest of columns with data. ColumnHeader from simulator output
- Confirm that simulator reads the file correctly
- Confirm that data comes (wbcmd)

Modify blink_app and try on simulator

- Remove blink timer
- Add asyncSubscribe and onNotify
- Try on simulator

Try on sensor

- If everything worked fine:

→ IT BLINKS!!!