Using Android smartphones to control music software with OpenSoundControl (OSC) using Sensors2OSC

Agenda

- Overview of Sensors2OSC
- Demos
- Practical Tips
- Experimental Support for Bluetooth LE Sensors
- Alligator Bytes: Byproduct of Sensors2OSC

Sensors2OSC

- FLOSS Android app
- Sensor data via OSC over UDP
- Since 2014
- Available via F-Droid

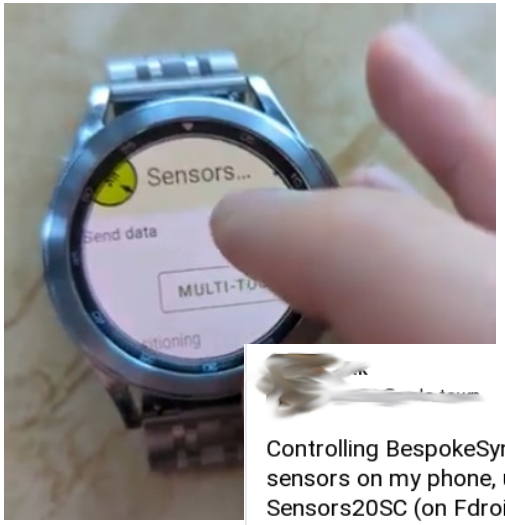
Basic idea

- Cheap Wiimote and Kinect replacement
- Smartphones are obsolete after short time
- Many sensors are included
- Communication via WIFI is ubiquitous

SAFETY

Design Decisions

- Easy to set up
- Support for old hardware (from Android version 4.0)
- As generic as possible
- Use all possible sensors
- UDP only
- Data interpretation on receiver side



Controlling BespokeSynth with the sensors on my phone, using Sensors20SC (on Fdroid) fixed my dislike of sound creation using my computer. Suddenly I have something like a digital theremin and what not in my hand. So fun.

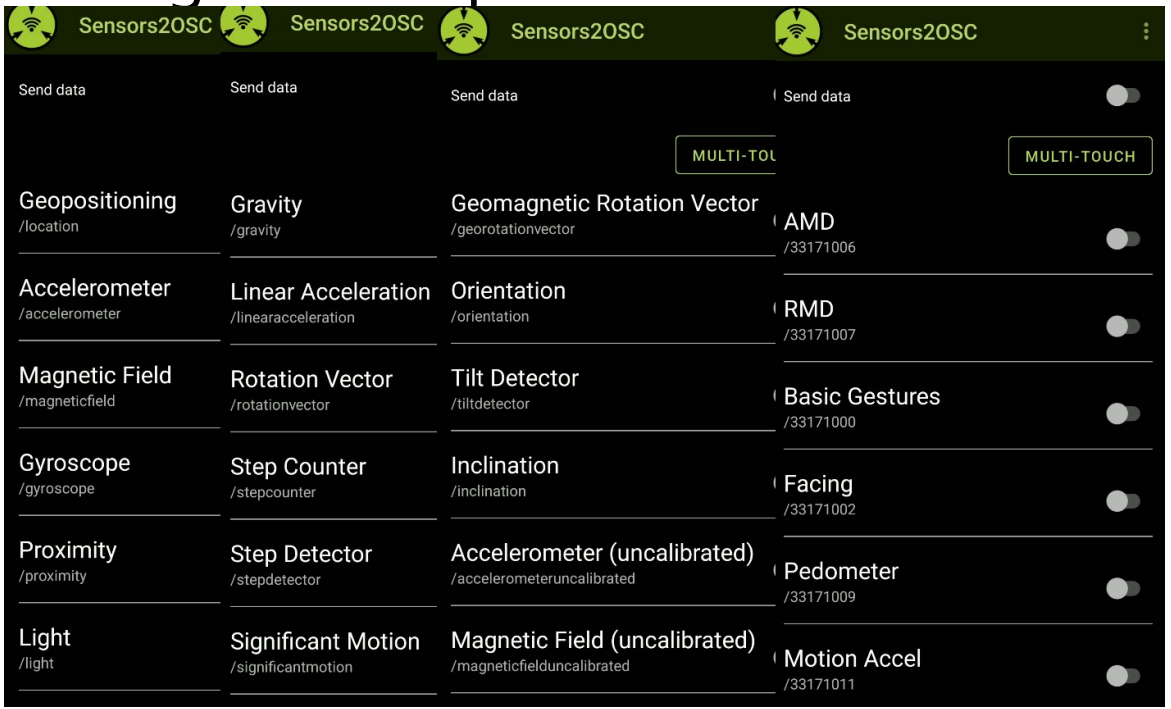
Übersetzen

#BespokeSynth #Sensors20SC #Fdroid

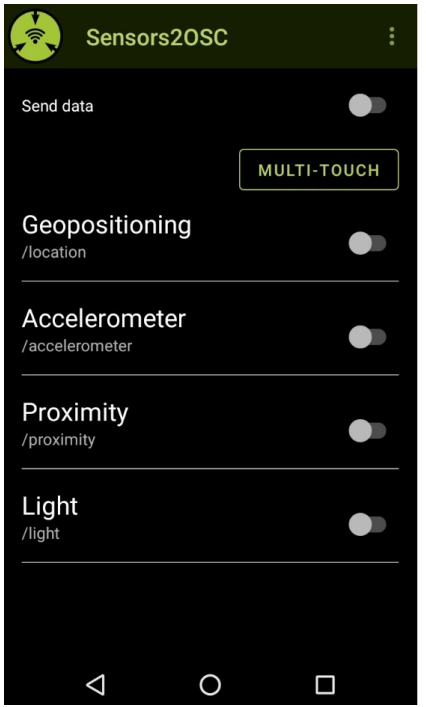
The project is amazing! I have a favour/request that I was wondering if you would be able to help with. I am and accessibility design researcher working on a project with some people that use electric wheelchairs for mobility, to create digital painting as they move around with their wheelchairs.

Sensors on Android

Lineage on Fairphone 3



Stock Moto E 3



Demo Time

LAC 25 Villeurbanne
2025-06-28

Sensors2OSC

Practical Tips: Sensors

- Accelerometer: 3D vector, (value in N along each axis) → changes not independent
- Orientation: combination of accelerometer and magnetic field, 3D vector (azimuth: 0 to 2π, pitch: -π to π, roll: -π/2 to π/2) → changes independent, but discontinuities on endpoints
- Light: wide variation (value in lux), sunny day inside: 60 to 100; cloudy day outside: 1700; night, inside, lights on: 20 → hard to use quantitatively
- Proximity: proximity in cm (in theory, in practice only two values 0 and 5) → works as off/on switch
- Touch: 2D Cartesian coordinates 0 to 1, lifting finger sends -1 for both

Practical Tips (2): Network Latency

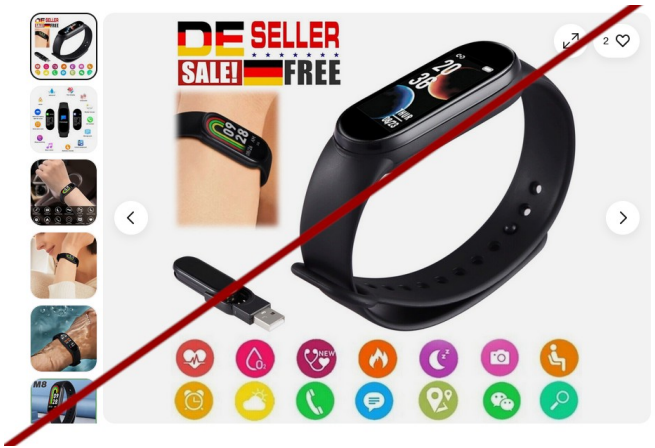
- Dedicated WIFI router for minimal latency
- Data rate should be at least “game speed”
- Only send sensor data that you need
- Multiple devices: Use different ports

Bluetooth LE Sensors

- Experimental
- Currently only sports related sensors implemented (heart rate, cycling cadence, cycling power, step counter)
- Data interpretation copied from Android app OpenTracks
- OSC Address (for now): `/bt/<sensorname>`
- Will not work for “companion devices”

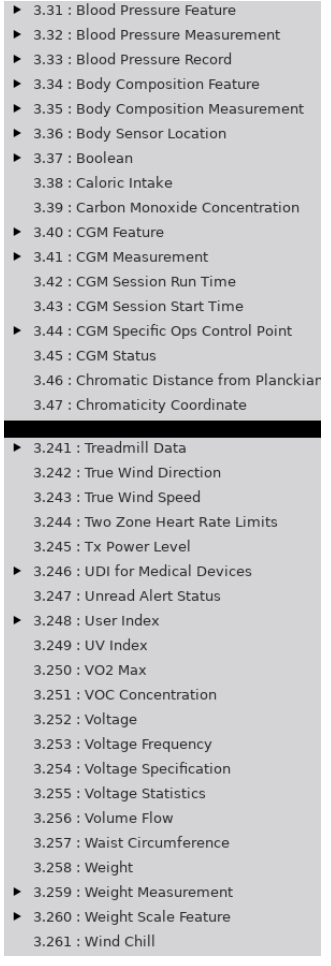
Hi,

Is there a way to read a cadence sensor connected via bluetooth to my android phone as the input source? I have your app sending various phone sensor data to my PC. I want to use the phone as a bridge between the cadence sensor and my pc (in testing to do the same to a quest 3 headset for input to a VRChat world) and so far don't see bluetooth or a cadence sensor as a possible source? Just wanted to make sure I wasn't missing something and/or get your advice.



Open Questions about UX

- General Attribute Profile (GATT) specs list 261 sensor types
- Send raw bytes of any GATT?
- Mix of interpreted data for known sensors and raw data otherwise?
- How should multiple devices with same GATT characteristics be handled?
- Toggle for each sensor?

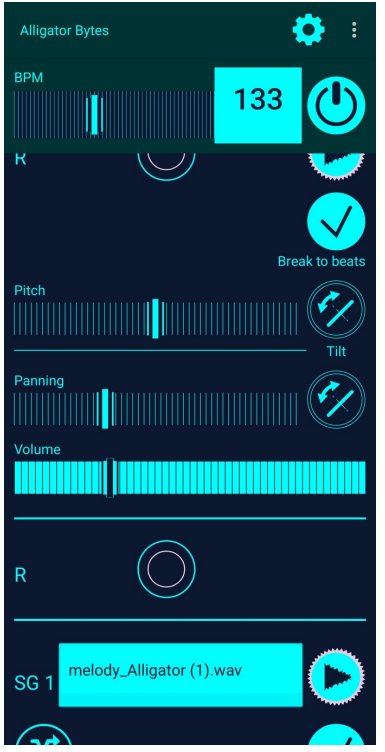


Alligator Bytes

- Made as performance app for Alligator Gozaimasu collective (http://alligator-go.space)
- Sample looper with pitch shift, speed change, panning, breaking up samples
- Implemented using LibPd and orientation sensors

Alligator Bytes (2)

- Loading and exporting samples / sample packs
- Record session
- TODO: UX, help
- Testers wanted for Play store release



Sensors2OSC:

- https://sensors2.org/osc
- https://f-droid.org/packages/org.sensors2.osc/
- https://codeberg.org/Residuum/Sensors2OSC

Alligator Bytes:

- https://f-droid.org/packages/org.residuum.alligator/
- https://codeberg.org/Residuum/alligator-gozaimasu-app

Thomas Mayer:

- https://www.residuum.org/
- thomas@residuum.org
- https://digitalcourage.social/@residuum