

OpenSensIOT
User Guide

Version 1.3

4.12.2018

History of changes

Version #	Date	Modified paragraph	Reason
0.1	14.10.2018	All	Created new document.
0.2	28.10.2018	3	Added testing procedures.
1.0	31.10.2018	All	Release of the document.
1.1	11.11.2018	2.3, 3	Minor cosmetic changes of diagrams.
1.2	26.11.2018	All	Minor cosmetic changes.
1.3	4.12.2018	All	Some content changes, larger cosmetic changes, added getting started chapter, file name changed to User Guide.

Table of contents

1	Introduction.....	5
2	Hardware Architecture.....	6
2.1	Features.....	6
2.2	Daughter board connector.....	7
2.3	Daughter board.....	8
2.4	Power supply and backup battery.....	8
3	Software architecture.....	9
3.1	Template firmware startup.....	9
3.2	Command line interface.....	10
3.3	Self-test procedure.....	11
3.4	IDE guidelines.....	11
3.4.1	IDE setup.....	11
3.4.2	Opening the project.....	11
3.4.3	Debugging the project.....	12
4	Getting started.....	13
4.1	Command line interface.....	13
4.2	Programming and debugging.....	13

List of figures

Figure 1:	Main board.....	5
Figure 3:	Daughter board to main board connector pinout.....	7
Figure 2:	Daughter board with 2-pin Molex picoBlade connector and two U.FL coaxial connectors for ECG input.....	8
Figure 4:	After peripherals and components initialization, program enters the main loop.....	9
Figure 5:	Project open window.....	11
Figure 6:	Debugger configuration window.....	12
Figure 7:	Main board prepared for testing.....	13

Abbreviations

ADC	Analog-Digital Converter
CLI	Command Line Interface
CMRR	Common-Mode Rejection Ratio
DAC	Digital-Analog Converter
DMA	Direct Memory Access
DSP	Digital Signal Processing
ECG	Electrocardiography
EEPROM	Electrically Erasable Programmable Read-Only Memory
EMI	Electromagnetic Interference
FPU	Floating Point Unit
GPS	Global Positioning System
GSM	Global System for Mobile communication
IDE	Integrated Development Environment
IoT	Internet of Things
MCU	Microcontroller Unit
MPU	Memory Protection Unit
PWM	Pulse Width modulation
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
RTC	Real Time Clock
USART	Universal Synchronous and Asynchronous Receiver-Transmitter
USB	Universal Serial Bus

1 Introduction

OpenSensIoT board is state of the art open-source development platform. With its powerful 32-bit ARM®Cortex® processor, low-power sensors with local telemetry acquisition and comprehensive collection of radios, OpenSensIoT development board was designed to provide its users quick prototyping and low-cost development for variety of embedded IoT applications in a small size.

User-friendly OpenSensIoT development kit comes equipped with numerous C-code examples and complete low-level drivers for both basic and advanced prototyping and development. Software package comes with various projects which are primarily designed for Atollic True Studio IDE for STM32.

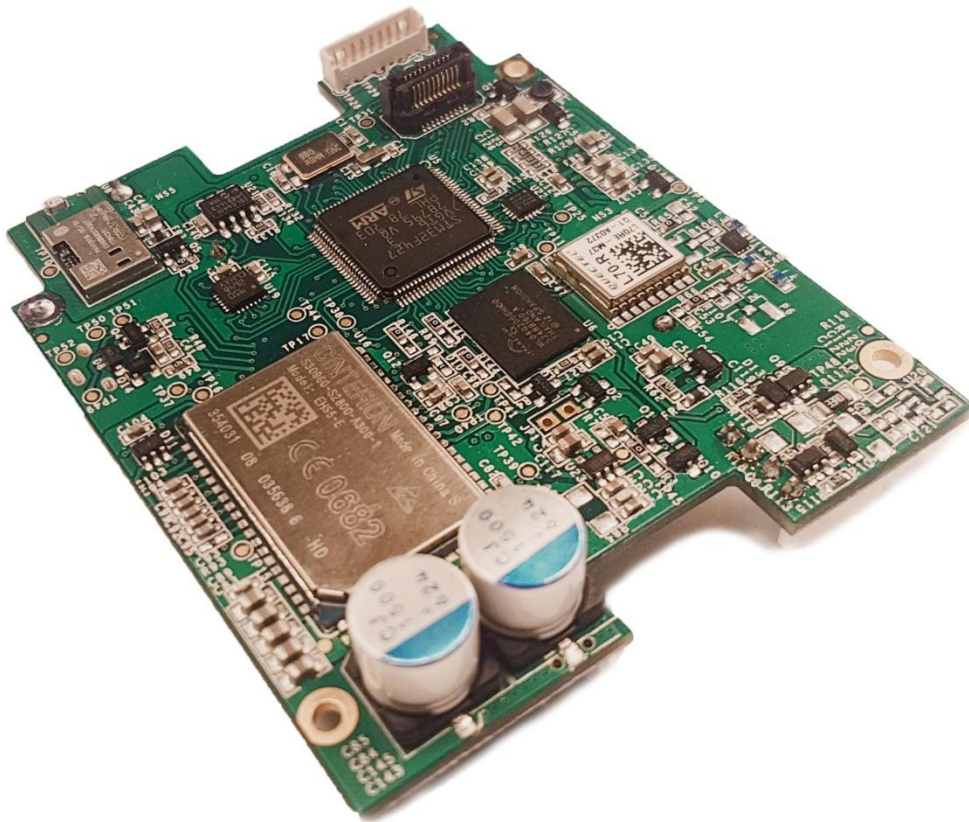


Figure 1: Main board

2 Hardware Architecture

Heart of OpenSensIOT main board is STM32F4 microcontroller which is based on Arm Cortex-M4 32-bit RISC core, operating at frequency up to 180 MHz. Core features FPU single precision, supports all Arm single-precision instructions and implements full set of DSP instructions and a MPU. MCU offers seventeen general purpose 16-bit timers with PWM functionality, three 12-bit ADCs, two 12-bit DACs, general purpose DMA (16 channels) and twenty-one communication interfaces (SPI, I2C, USART ...).

These peripherals are connected to gyroscope, accelerometer, magnetometer, EEPROM, NAND flash, GPS module and GSM module on the main board. Daughter board is extension to the main board which is connected to it via 20-pin connector. Daughter board additionally incorporates pressure sensor, pulse oximeter front-end, ECG front-end and 24-bit ADC. Most of these have enable signals connected through one 8-bit and one 16-bit IO expander.

Besides on-board EEPROM and NAND flash, MCU has 2 MB flash memory and 256+4 kB SRAM (4 kB as backup).

2.1 Features

- High-performance 180 MHz 32-bit ARM®Cortex®-M4 with FPU core, 2 MB Flash memory and 256+4 kB RAM
- 24-Bit High-precision AD converter with analog input from Wheatstone bridge
 - Internal amplifier with gain of 128
 - Low input noise (35 nVrms)
 - Input EMI filter
 - Sample rate: 80 samples per second
- 24-Bit Analog Front-End for Biopotential Measurements with ultra-small coaxial connectors
 - Programmable gain of 1, 2, 3, 4, 6, 8 or 12
 - Low input noise (8 μ Vrms)
 - Programmable data rate from 125 to 8k samples per second
 - Low power: 335 μ W/channel
 - CMRR: -105 dB
- Full 3D-axis Inertial System
 - Accelerometer $\pm 2g/\pm 4g/\pm 6g/\pm 8g/\pm 16g$
 - Gyroscope $\pm 245/\pm 500/\pm 2000$ dps
 - Magnetometer $\pm 2/\pm 4/\pm 8/\pm 12$ gauss
 - 16-bit data output
 - Temperature sensor (12-bit data)
- Piezoresistive Absolute Pressure Sensor
 - 260-1260 hPa absolute pressure range
 - High-resolution mode: 0.01 hPa RMS
 - Low power consumption of 4 μ A
 - 24-bit data output
 - Programmable output data rate from 1 to 25 Hz
- GPS

- Supports autonomous GPS C/A, SBAS function (including WAAS and EGNOS) and AGPS (EASY function)
- High-Efficient 3G Wireless Module
 - EHS5-US (850/1900 MHz) for North America, and EHS5-E (900/2100 MHz) rest of the world. USB, Embedded TCP/IP with TLS 1.2

2.2 Daughter board connector

Daughter board connector is Samtec 20 pin connector from series LSS. It has routed two I2C buses, two SPI buses and some GPIO pins. One SPI bus is initially menth for ADS1292R interface which is actually a simple serial interface that shifts data on each ADS1929_SCLK positive edge clock signal.

+3.0V	1	2	GND
VBAT	3	4	GPIO12
I2C1_CLK	5	6	ADS1292R_SCLK
I2C1_SDA	7	8	ADS1292R_CS
I2C2_SDA	9	10	ADS1292R_DRDY
I2C2_CLK	11	12	ADS1292R_DOUT
AFE4490_CS	13	14	ADS1292R_DIN
SPI1_MOSI	15	16	SB_IO_EXT_CS
SPI1_MISO	17	18	SB_IO_EXT_RESET
SPI1_SCK	19	20	SB_IO_EXT_INT

Figure 2: Daughter board to main board connector pinout.

User can easily design its own daughter board. Both I2C and SPI buses are shared with other components on the main board, but all GPIOs may be used as independent inputs or outputs.

2.3 Daughter board

Daughter board is functional extension of the main board. It is connected to it via 20-pin small-print connector described in chapter 2.2, which acts as a power-supply and communication link. User can access daughter board extension functionality with I2C or SPI buses.

To use 24-bit ADC with Wheatstone bridge input, just connect the resistance sensor to Molex picoBlade connector P22 as shown in the Figure 3. For ECG measurements connect ECG probes to the two ultra-small coaxial connectors (one to channel 1 and another to channel 2).

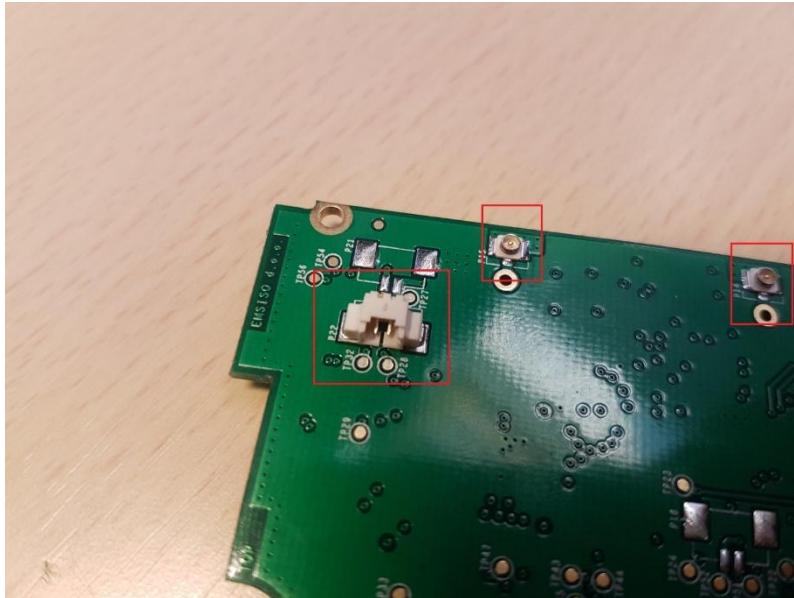


Figure 3: Daughter board with 2-pin Molex picoBlade connector and two U.FL coaxial connectors for ECG input.

2.4 Power supply and backup battery

Main board and its daughter board are powered by a single cell Li-Ion battery connected via JST XH 4-pin crimp style connector. If Li-Ion battery is applied, user has to put a jumper on P12 pins. If power supply is used instead of Li-Ion battery, note to provide 3,5 – 4.0 V and at least 0,5 A.

CR1632 battery is soldered on the main board as a backup power supply for RTC.

3 Software architecture

OpenSensIOT template firmware provides basic low level drivers for most of the periphery. It was developed with an emphasis on open-source, meaning that everyone can share, modify and use the code as is, according to his needs. It has MIT license and does not provide any warranty.

OpenSensIOT template firmware was developed in Atollic True Studio for STM32. Project is organized into the following folder structure:

Drivers	CMSIS and STM32F4 HAL drivers. (Do not edit)
Inc	Device drivers and project specific code (Header files).
Src	Device drivers and project specific code (Source files).
Startup	MCU specific start-up code.

The template firmware uses HAL drivers for peripherals and MCU specific functions. Project uses nano-lib standard library configuration.

3.1 Template firmware startup

Provided template firmware has a non-deterministic approach, meaning that the infinite loop pauses executing once the interrupt triggers. Note that USART interrupts only put received data into FIFO buffers, which are later processed in infinite loop. Handling all received data directly in the interrupt routines would result in character losses.

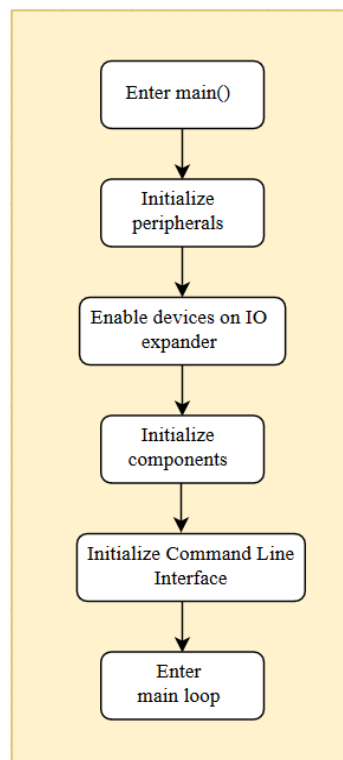


Figure 4: After peripherals and components initialization, program enters the main loop

3.2 Command line interface

Template firmware has inbuilt command line interface for easy debugging and basic interactions with the main and daughter board. There are two varieties of communication implemented to communicate with the PC or other serial device. One is via USART and the other is via USB. Both types can work simultaneously, but USB debugging can be disabled by commenting the define `CLI_OVER_USB` in `main.h`. Note that debug USART interface is hardly reachable on the PCB.

Terminal settings

Baud	115200 bps
Data	8 b
Stop	1 b
Parity	None

CLI supports basic set of commands, which were specifically added for the OpenSensIOT project. By sending command 'help' from the terminal, the board will print basic set of commands back to the terminal.

Note: When using USB for serial communication the procedure for successfully connecting to PC terminal is to always first unplug USB cable and turn off the board supply, then turn on the board supply and wait approximately 10s for board initialization, after that plug the USB cable to PC and finally open terminal and COM port on it. CLI over USB should be working now.

OpenSens Commands

- OS+GET_VERSION (returns version of the firmware)
- OS+GET_BATTERY (returns battery voltage level)
- OS+GET_TEMP (returns temperature from inertial sensor – this can easily be fused and averaged with pressure sensor IC temperature sensor)
- OS+EINK_REFRESH (refreshes E-Ink display with demo print out)
- OS+SELF_TEST (initiates self-test and returns status of each component)

Inertial Sensors Commands

- IS+GET_GYRO (returns gyroscope x, y, z values in mdps)
- IS+GET_ACC (returns accelerometer x, y, z values in mg)
- IS+GET_MAG (returns magnetometer x, y, z values in mgauss)

Pressure Sensor Commands

- PS+GET_PRES (returns pressure value in hPa)

Strain Gauge ADC Commands

- SG+GET_STRAIN (returns raw value from 24-bit ADC)

ECG ADC Commands

- EC+GET_ECG (returns raw value from 24-bit ADC front-end for ECG measurements)

GPS Commands

- GP+GET_GPS (returns latitude and longitude values in degrees)

GSM Commands

- GS+[AT command] (returns data from GSM module for specific AT command which is sent)

3.3 Self-test procedure

OpenSensIOT template firmware has a built-in self-test procedure which probes the components on the board and prints the results to the CLI. It can be initiated from the CLI with the command “OS+SELF_TEST”. Self-test procedure currently does not support any recovery or error handling i.e. re-initialization of peripherals or components which is application specific and can be easily implemented by the end user.

3.4 IDE guidelines

3.4.1 IDE setup

- 1.) Download and install Atollic TrueSTUDIO for STM32 from Atollic webpage (<https://atollic.com/resources/download/>, version 9.1.0, build 20181011-1241 was used for the development).

3.4.2 Opening the project

- 1.) Open template or other project file with Atollic TrueSTUDIO IDE by clicking “File -> Open Projects from File System”.
- 2.) Navigate to the Directory button and search for folder with the existing project and click “Finish”.

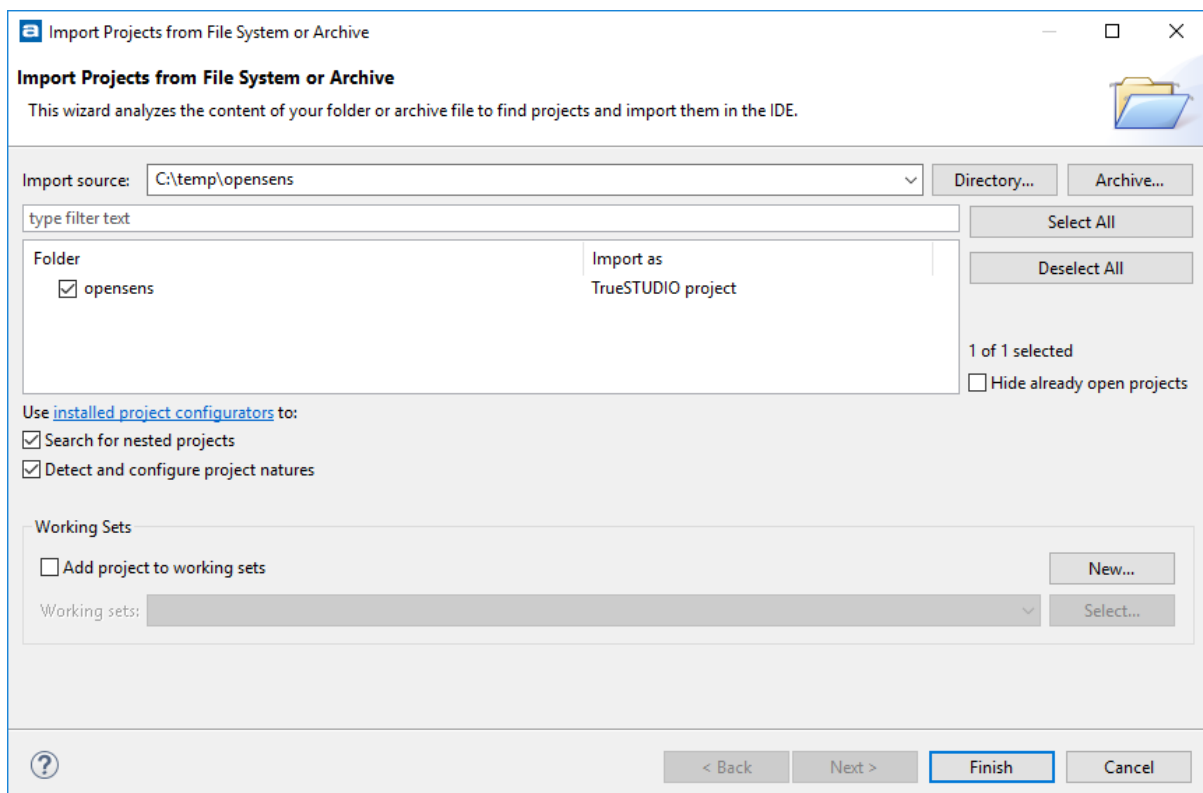


Figure 5: Project open window.

- 3.) Build the project by clicking “Run -> Build”, there should be no errors reported after build is done.

3.4.3 Debugging the project

- 1.) Setup debug configuration by clicking “Run -> Debug Configurations”.
- 2.) Under tab “Embedded C/C++ Application”, configuration as seen on image below should be set (select “ST-LINK” debug probe and “SWD” GDB Server Command Line Option).

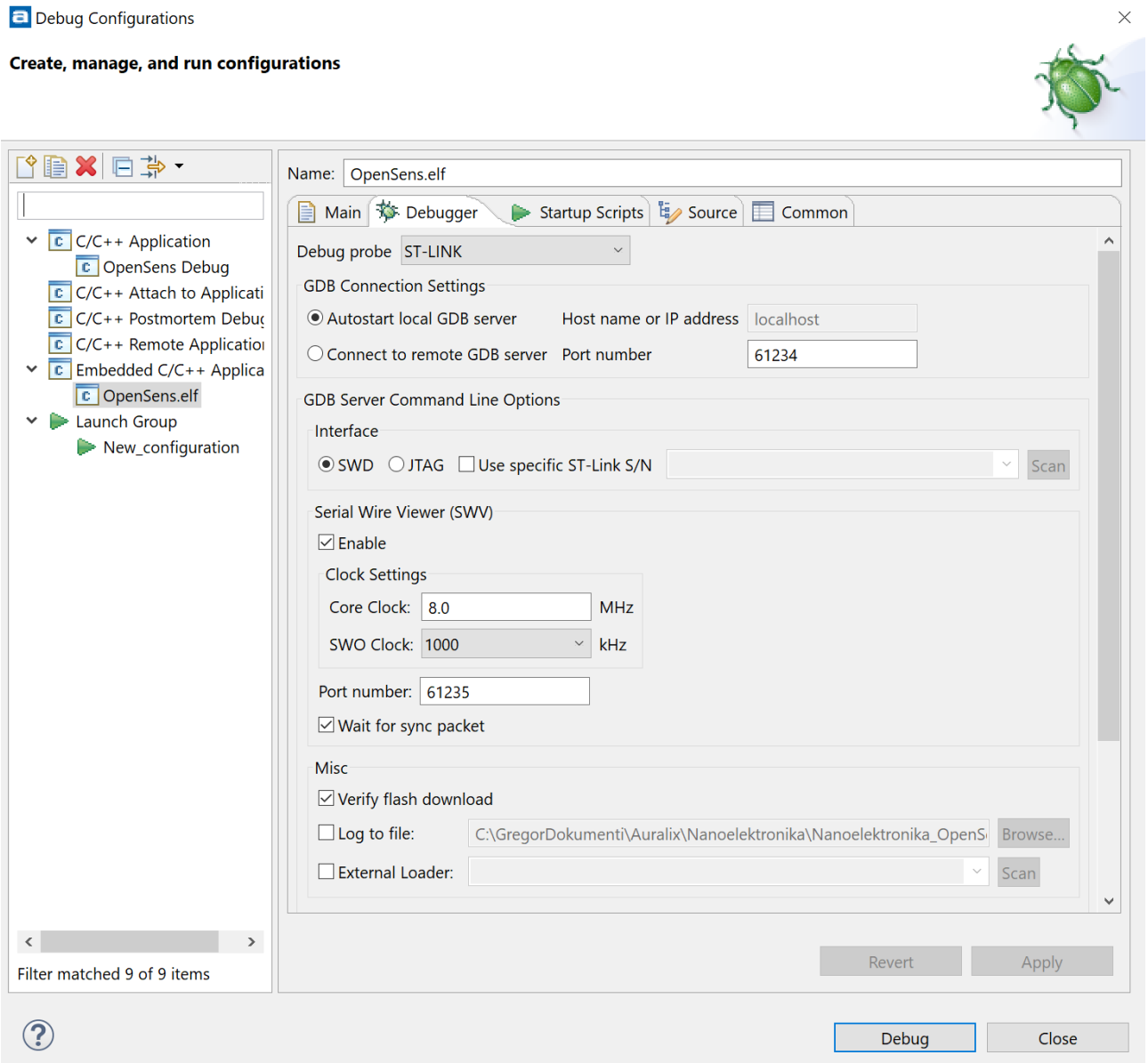


Figure 6: Debugger configuration window

- 3.) Start debugging by clicking on “Run -> Debug”, program should stop at the breakpoint at the beginning of the main function.

4 Getting started

4.1 Command line interface

- 1.) Make sure to have any COM port terminal software installed on the computer (example; [YAT terminal](#)).
- 2.) Connect the battery or external power supply (3,7 V and 0,5 A) to power up the OpenSenseIOT board via connector P11.
- 3.) Aproximately 10 seconds after power up, connect the USB for debugging to the computer USB port and open the terminal software.
- 4.) In the terminal software select correct COM port and open it.
- 5.) Use CLI commands described in chapter 3.2 to run any functions implemented into the template firmware or implement you own functions. Type "help" command for example to get a printout of available commands.

4.2 Programming and debugging

- 1.) Connect the ST-Link programmer to the OpenSenseIOT board, like shown on the Figure 7.
- 2.) Connect the battery or external power supply (3,7 V and 0,5 A) to power up the OpenSenseIOT board via connector P11.

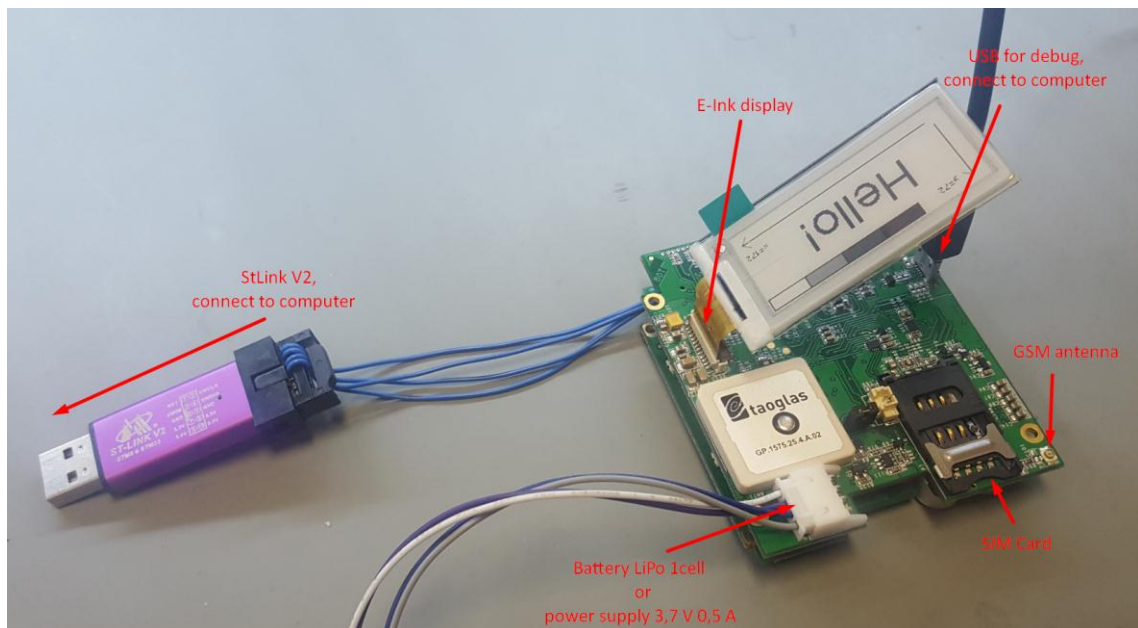


Figure 7: Main board prepared for testing

- 3.) Make sure to have template firmware transferred from GitHub or from provided USB stick to your computer.
- 4.) Make sure to have IDE installed like described in the chapter 3.4.1.
- 5.) Open the template project or any other project like described in the chapter 3.4.2.
- 6.) Start debugging the project like described in the chapter 3.4.3.