

SSU AND WAP

USER MANUAL



Date: December 23, 2016

Version: 1.0

Author: Gerard Hogenhout, Kees Hogenhout, Jasper Sikken

Support: info@sodaq.com

Contents

1	Introduction	3
2	Specifications	4
3	Using the devices	5
3.1	Setting up	5
3.2	Using	6
A	Settings and debugging	7
A.1	Required hardware to connect the SSU to a PC	7
A.2	Installing the Arduino IDE and drivers	7
A.3	Menu	8
A.4	Calibrating the WAP pressure sensor	9
A.5	Programming the SSU	10
A.5.1	Burning bootloader to the SSU	10
A.5.2	Flashing the SSU	11
A.6	Programming the WAP	12
A.6.1	Burning bootloader and set fuses to the WAP	12
A.6.2	Flashing the WAP	12
A.7	Wiring between SSU and WAP	13
B	Validation	14
B.0.1	Validating SSU-WAP in 2m of water	14
B.0.2	Validating SSU-WAP in 20m of water	15
B.0.3	Validating the WAP alone in 1m water	15
C	Validation 20m	17

Chapter 1

Introduction

The purpose of the WAP (WAter Pressure sensor) is to collect and transfer data about the water pressure in a specific borehole, in millibar or mbar. The WAP sensor is connected with a cable to a second module that measures atmospheric pressure, the SSU (Standard Surface Unit), which reads the data from the WAP and transmits the data through 2G cellular connection to a receiving server. The WAP receives its power from the SSU. Together with data on atmospheric pressure at the same location of the borehole, the level of the water in the borehole can be calculated (in m below the water surface), and thus help water providers and users collect more accurate and more real-time information about the availability of water in a specific borehole.

Chapter 2

Specifications

WAP pressure range	0-14 bar
WAP pressure accuracy	+/-20 mbar (up to 6 bar or 50 meter water)
WAP pressure resolution	1 mbar
WAP pressure stability	-20 mbar/yr
WAP temperature range	-40°C to +85°C
WAP temperature accuracy	1°C (0-10 bar)
WAP temperature resolution	1°C
WAP power consumption	13mA during a measurement
SSU-WAP communication	RS485
SSU-WAP cable length	25 meter
SSU pressure range	300 to 1100 hPa (or mbar)
SSU pressure accuracy	1 hPa (or mbar)
SSU pressure resolution	1 hPa (or mbar)
SSU pressure stability	1 hPa/yr
SSU temperature range	-40°C to +85°C
SSU temperature accuracy	1°C
SSU temperature resolution	1°C
SSU upload interval	24 hours (adjustable)
SSU communication	2G cellular network
SSU operating temperature	0°C to 45°C
SSU Interface to PC	USB2.0 through tag-connector and tag-connect adapter
Power supply	Solar panel
Battery	1200mAh 3.7V Li-Po rechargeable battery
Time to first measurement	2 minutes
Measurement interval	5 minutes (adjustable)

Chapter 3

Using the devices

3.1 Setting up

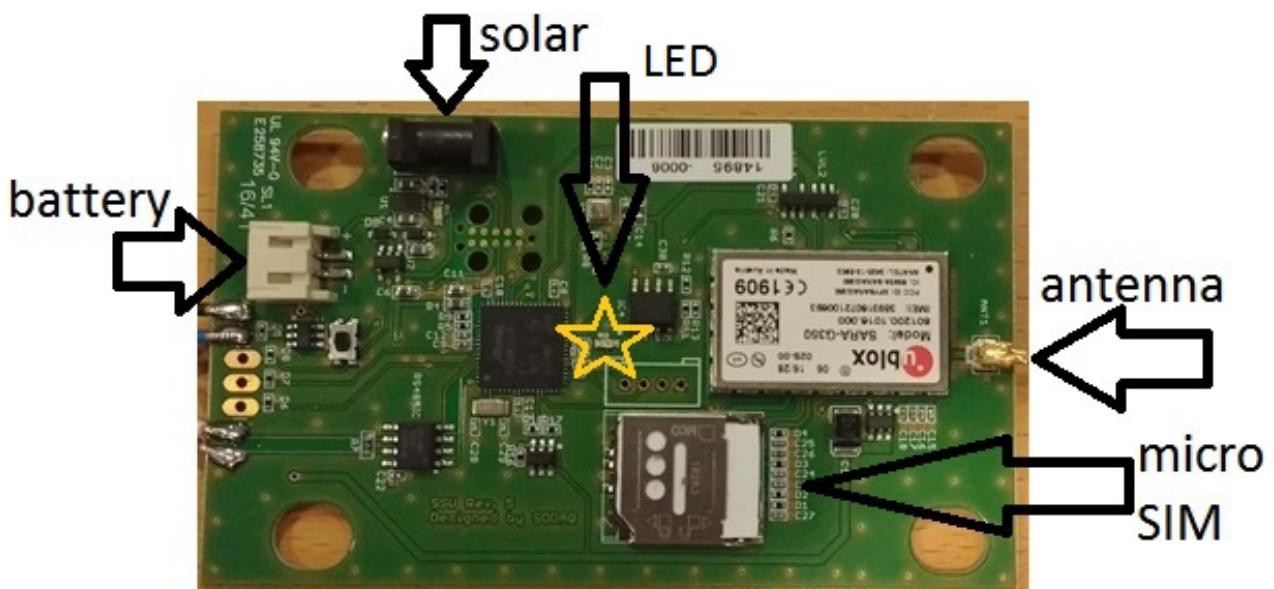


Figure 3.1: Connections on the the SSU board

1. Insert the micro SIM card
2. Connect the antenna to the u.FL connector
3. Connect the battery
4. Connect the solar panel

The SSU needs to be placed above the water/ground surface and with the solar panel directed to the sun and have clear view on the sky. The SSU needs to be placed in an area with sufficient 2G cellular reception. The WAP needs to be placed under water, according to the instructions of the borehole supervisor. The depth of the WAP in the borehole at the location needs to be specified and instructed by the borehole supervisor.

3.2 Using

In normal operation about 90 seconds after power up, the LED on the SSU board starts to blink with 1 sec interval. If measurement interval is set to 5 minutes, the first measurement is sent to the MQTT server after 5 minutes. To configure the measurement interval please see section Menu in Appendix A.

Correct functioning of the SSU can be confirmed by reading the messages from the MQTT server by using a MQTT client.

1. Install a MQTT client, ie. MQTTfx
2. Create a new connection profile, name it SSUWAP (can be anything)
3. Use broker address vps01.m2m4all.com and port 1883
4. Use client id MQTTfxClient (can be anything)
5. Under tab User Credentials enter User Name: *itay* and Password: *sodaq*
6. Use defaults for other settings
7. Connect to MQTT server
8. Under the tab Subscribe you need to type ITAY/SSU and press Subscribe button

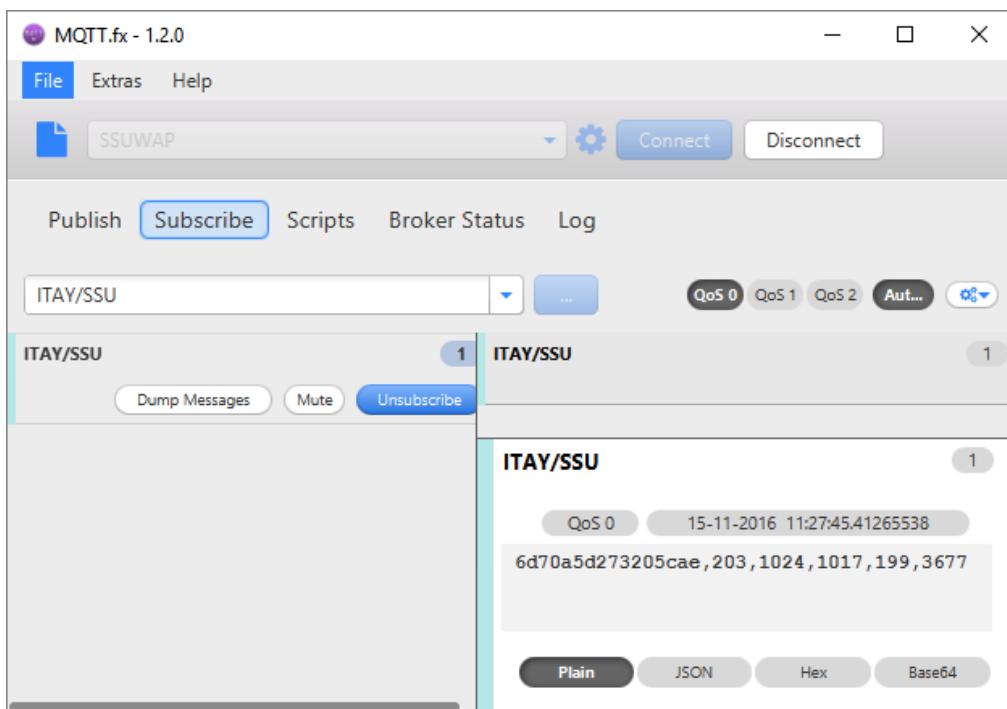


Figure 3.2: Messages received by the MQTT client

As soon as you connect you will see the last (old) message that was stored on the MQTT server.

A good message looks like this 6d70a5d273205cae,203,1024,1017,199,3677

Formatting is: identification, SSU temperature (C/10), SSU pressure (hPa), WAP pressure (millibar), WAP temperature (C/10), battery voltage (mV).

If the SSU-WAP is working correctly, a new message appears at the configured interval in the MQTT client. The first message will appear AFTER the first interval time. To change the interval, please see section menu in Appendix A.

If the WAP is not working correctly WAP pressure and WAP temperature will have value -999.

Appendix A

Settings and debugging

To change SSU settings (ie. measurement interval) and perform simple debugging you need hardware to connect the SSU to a pc, and you need to use a serial monitor on the pc.

A.1 Required hardware to connect the SSU to a PC

You need a tag connector, a tag-connect adapter board and a micro USB cable.

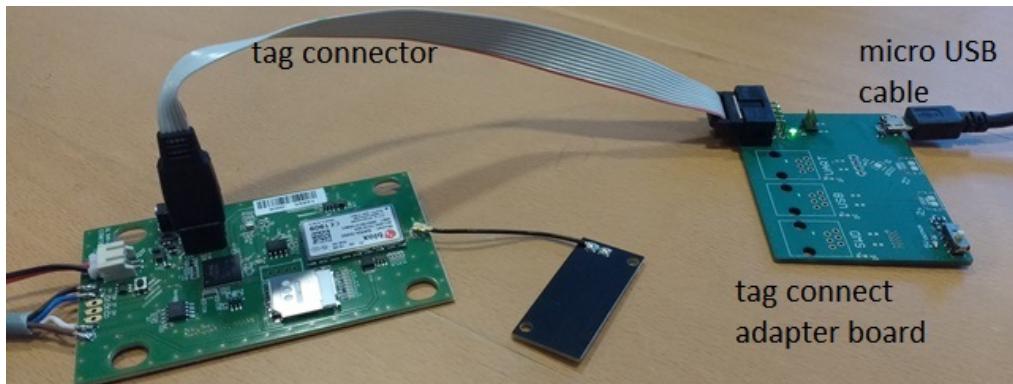


Figure A.1: Connecting the SSU to a pc

A.2 Installing the Arduino IDE and drivers

You also need to install a serial monitor on your PC to be able to make configurations settings to SSU and to read logging/debugging messages from the SSU.

1. Download the Arduino IDE <https://www.arduino.cc/en/Main/Software>
2. Install the Arduino IDE and start it
3. Click on File → Preferences and at the bottom you should see 'Additional Boards Manager URLs'. This is where you need to paste the following URL: http://downloads.sodaq.net/test/package_sodaq_index.json
4. When you have pasted the URL, click 'OK' and 'OK' and you are ready for the next step.
5. Click on Tools → Board → Boards Manager

6. Scroll all the way to the bottom, you should see SODAQ SAMD Boards. Click on it, and install the latest version.

Now it is possible to use the built in serial monitor of the Arduino IDE (Tools > Serial Monitor) or use a different serial monitor i.e. PuTTY.

A.3 Menu

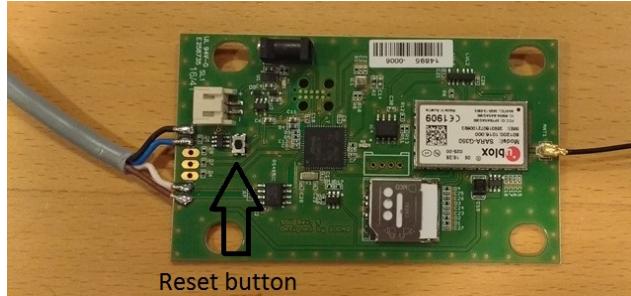


Figure A.2: Press reset button first

After reset of the SSU a COM port with name Arduino Zero will appear in Windows Device Manager. Use the same COM port number in Arduino Serial Monitor. The following will be visible on the Serial Monitor

```

SSU_WAP v0.2.0-17-g12f5bf2
Board reset by (16):
External
[DEVICE ID]SAMD Serial Number
[DEVICE ID]Device ID: d268c04371235cae
INFO: Detected BME280 at 0x76
WAP Communication setup...

```



```

Configuration:
Sample      (sm=): 60
station name (nm=): SSU_WAP
APN         (apn=): public4.m2minternet.com
APN user    (apnusr=):
APN password (apnpw=):
MQTT server (srv=): vps01.m2m4all.com
MQTT port   (port=): 1883
MQTT user   (mqusr=): itay
MQTT password (mqpw=): sodaq
MQTT topic prefix (mqtop=): ITAY/SSU
WAP press offset (press=): 0
Enable Diag (ed=): 1
Keep USB on (usb=): 1
WAP pressure: 1019
BME280 pressure: 1023
WAP press offset: 4

```



```

Enter command: 3

```

Figure A.3: SSU menu

1. General information
 - (a) The device name and software version

- (b) Board reset: cause for the last reset: External
- (c) DEVICE ID: a unique code for each of the SSU-s
- (d) INFO: debugging information
- (e) WAP: initialization of communication with WAP.

2. Menu configuration:

- (a) Sample: time interval for measurements, in minutes
- (b) Station name: name of this SSU station, may be changed
- (c) APN: the APN of the network/operator
- (d) APN user: username for the network APN, usually left blank
- (e) APN password: password for the network APN, usually left blank
- (f) MQTT server: address of the MQTT server
- (g) MQTT port: port of the MQTT server, ussually 1883
- (h) MQTT user: user name for MQTT server
- (i) MQTT password: password for MQTT server
- (j) MQTT topic prefix: enter topic for publishing
- (k) WAP pressure offset: shows the programmed offset applied to WAP pressure
- (l) Enable Diag: if set to 0, less diagnostic messages will printed to the serial monitor
- (m) Keep USB On: if set to 0, USB will be disabled about 30s after restart
- (n) WAP pressure: displays the WAP pressure, used for setting the WAP offset (mbar)
- (o) BME280 pressure: displays the SSU pressure, used for setting the WAP offset (mbar)
- (p) WAP press offset: shows the difference between the above two lines

3. Extra commands:

- (a) OK : Save the configuration parameters and start the program now

If no command is given the program will continue after 30 seconds.

A.4 Calibrating the WAP pressure sensor

The SSU and WAP pressure sensor usually don't have the same value. The SSU pressure sensor, the BME280, is more accurate than the WAP pressure sensor. To calibrate the WAP pressure against the SSU pressure sensor type *pres=<the value from WAP press offset>* and press enter. The entered value will be visible in the menu after *WAP press offset*. After configuring the measurement interval, the corrected WAP pressure will be sent to the MQTT broker. In the MQTT client you will see that the WAP and SSU pressure are the same.

A.5 Programming the SSU

A.5.1 Burning bootloader to the SSU

A bootloader must be burned to the SSU microcontroller to be able to program it over USB serial port from the Arduino IDE. The bootloader is burned using the Atmel-ICE debugger/programmer. In addition the tag connector, the tagconnect adapter board and a USB cable are required.

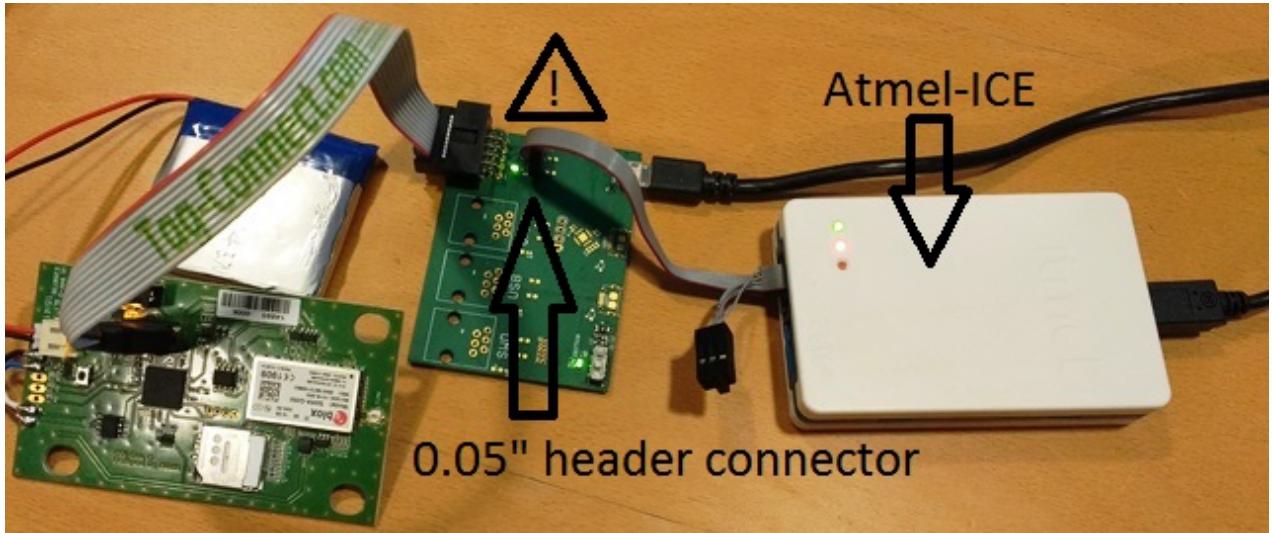


Figure A.4: Atmel-ICE

Steps for burning the bootloader

1. Connect the tag connector and the adapter board
2. Power the SSU using a battery or through the USB cable connected to the adapter board
3. Use the small flat cable that came with the Atmel-ICE. Put one end in the SAM port (of the Atmel-ICE) and connect the other end to the 2x5pins 0.05" pitch header on the adapter board. Please note the correct orientation of the IDC connector/flat cable.
4. Connect Atmel-ICE with USB to PC
5. Optionally install the driver for the Atmel-ICE by installing the latest Atmel Studio
6. Start Arduino IDE and select Tools→Board→Arduino/Genuino Zero (Programming port)
7. In Arduino IDE select Tools→Programmer→Atmel ICE
8. In Arduino IDE select Tools→Burn bootloader

If burning the bootloader is successful the Arduino IDE looks like the picture below. Sometimes the SSU needs a reset just before burning the bootloader. The SSU will now show up with a COM port in Windows Device manager and it is ready to load the embedded software.

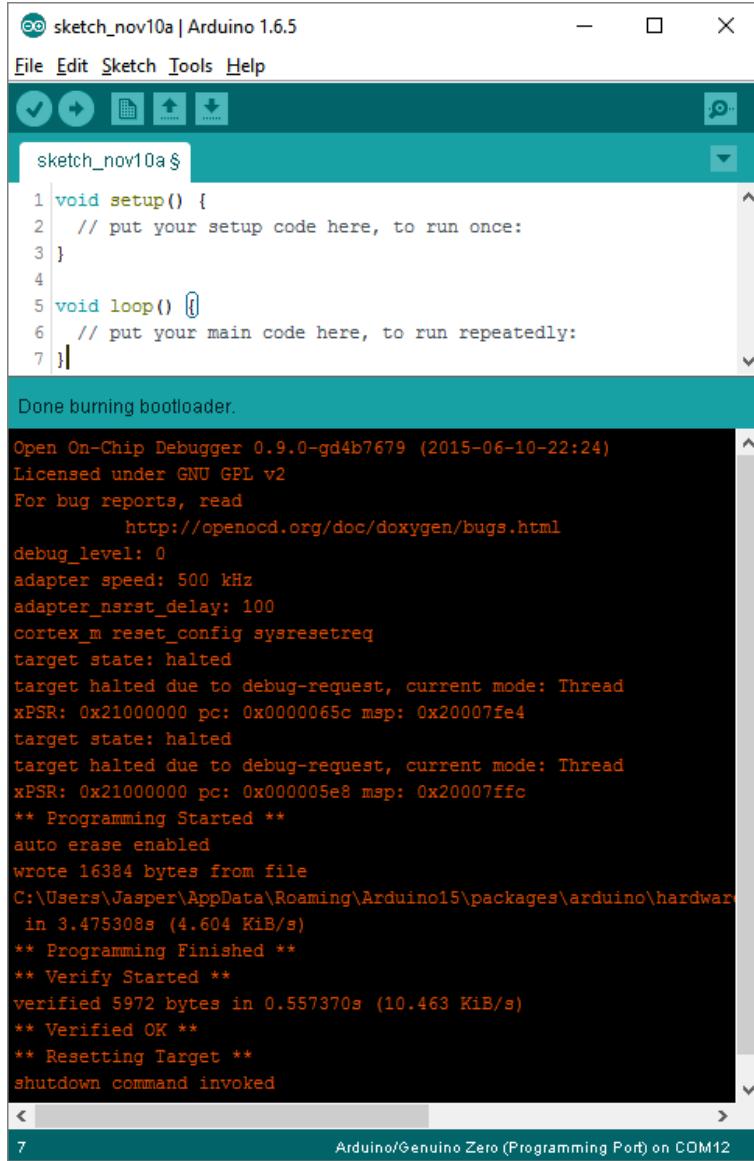


Figure A.5: Burning the SSU bootloader is successful

A.5.2 Flashing the SSU

After burning the bootloader the SSU will show up with a COM port in Windows Device Manager. Now the SSU microcontroller can be programmed using the Arduino IDE over the USB.

1. Download the latest firmware, at the time of writing: http://downloads.sodaq.net/SSU_WAP-v0.2.0-14-gc8846a2.zip
2. Unzip the firmware on your computer
3. Open SSU_WAP.ino in Arduino IDE
4. Make sure board SODAQ SSU is selected
5. Make sure the right COM port is selected
6. Press upload (ctrl+u)

A.6 Programming the WAP

Please note that the programming of the WAP has to be done before assembling the WAP into the casing.

A.6.1 Burning bootloader and set fuses to the WAP

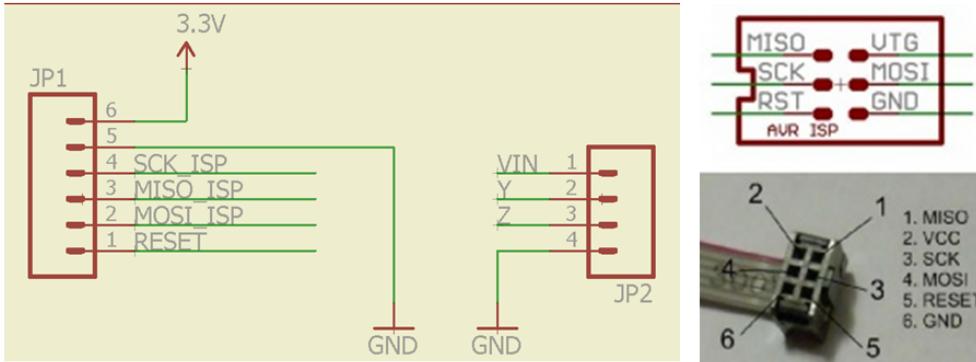


Figure A.6: Pinout of Atmel ISP programmer

Burning a bootloader can be done by using an Atmel ISP programmer (AVRMYISPMKII). See the steps below:

1. Connect the ISP programmer to the WAP-board with the pinout as below. Mind the profile on the 6-pole ISP-connector.
2. We also need power on the WAP board. For this we need an external power supply of max 6.5V connected to VIN on JP2. Make sure the GND of the RS45 and the WAP are anyway connected!
3. Open the project in Atmel studio and compile it with F7 on "release" dropdown.
4. Select ATtiny441 in the program and the AVR MKII programmer as below:



Figure A.7: Atmel Studio settings to burn bootloader

5. Build the program without debugging. It should show "ready" to the left below.

The fuses have to be burned too. In the WAP project folder, there is a README.md file with the right settings for the fuses. Go to Tools/device programming, select the right programmer and board (if not done) and click "Apply" and "Read". If everything is OK, it should show a Target voltage of around 3.4V. Switch to vertical tab "Fuses". Set the fuses as following (source: README.md):

1. EXTENDED FUSE: 0xED
2. HIGH FUSE: 0xD5
3. LOW FUSE: 0xC2

Click program. The right fuse values are written and everything is burned.

A.6.2 Flashing the WAP

Connect the Uartsbee to the RS45 breakout board from SparkFun, GND-GND, VCC-VCC, TX-RX-I, RX-TX-O. Do not connect the CTS to the RTS!!

Y-pin of WAP: $\pm 16.5\text{kV}$ HBM, $\pm 7\text{kV}$ IEC61000 (contact method) ESD Protected inverting differential transmitter output. Connect this pin to the A-pin of the RS485 breakout.

Z-pin of WAP: $\pm 16.5\text{kV}$ HBM, $\pm 7\text{kV}$ IEC61000 (contact method) ESD Protected noninverting differential transmitter output. Connect this pin to the B-pin of the RS485 pinout. Plug it in to a USB-port, open Arduino IDE, select the right COM and open the serial monitor. Set the baud-rate to 4800. The WAP outputs the pressure in millibars, the temperature in a 9 digit value and a checksum. This last sentences are used to test a separate WAP. The connections to the WAP are described in figure A.8

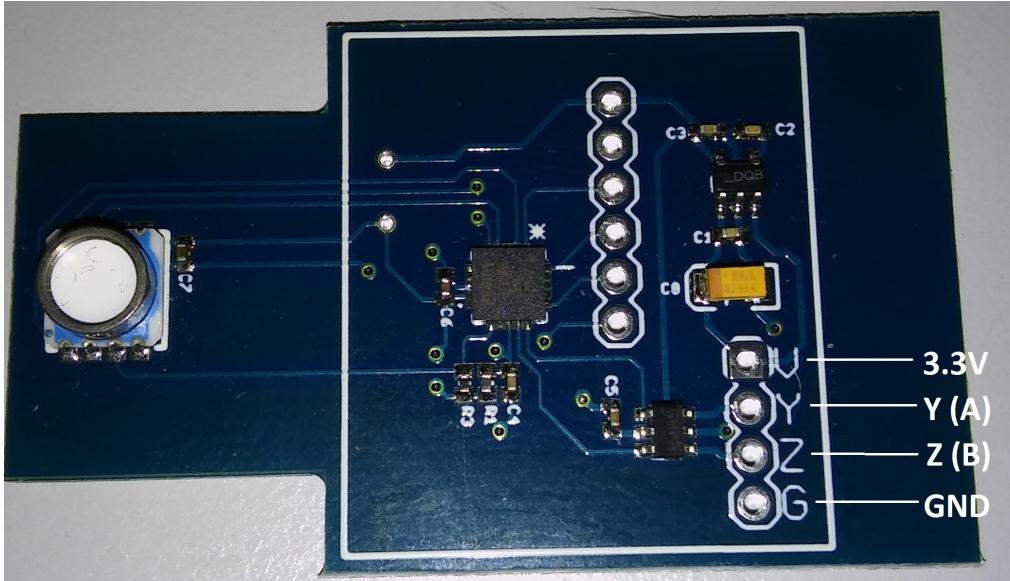


Figure A.8: Pinout description of the WAP

A.7 Wiring between SSU and WAP

The 25 meter cable must be soldered between SSU and WAP as shown in the picture below. Note the color coding of the wires. The used cable is order code 08.3084.000.000 from <https://www.binder-connector.nl>



Figure A.9: Solder connections of SSU and WAP

Appendix B

Validation

B.0.1 Validating SSU-WAP in 2m of water

On Nov 3rd 2016 the SSU-WAP was validated in a 2m long tube with tap water and the measured accuracy is 2%, which is in line with the MS5803 datasheet. It is required to zero-out the pressure difference between WAP and BME280 otherwise accuracy is 8%.

Test setup

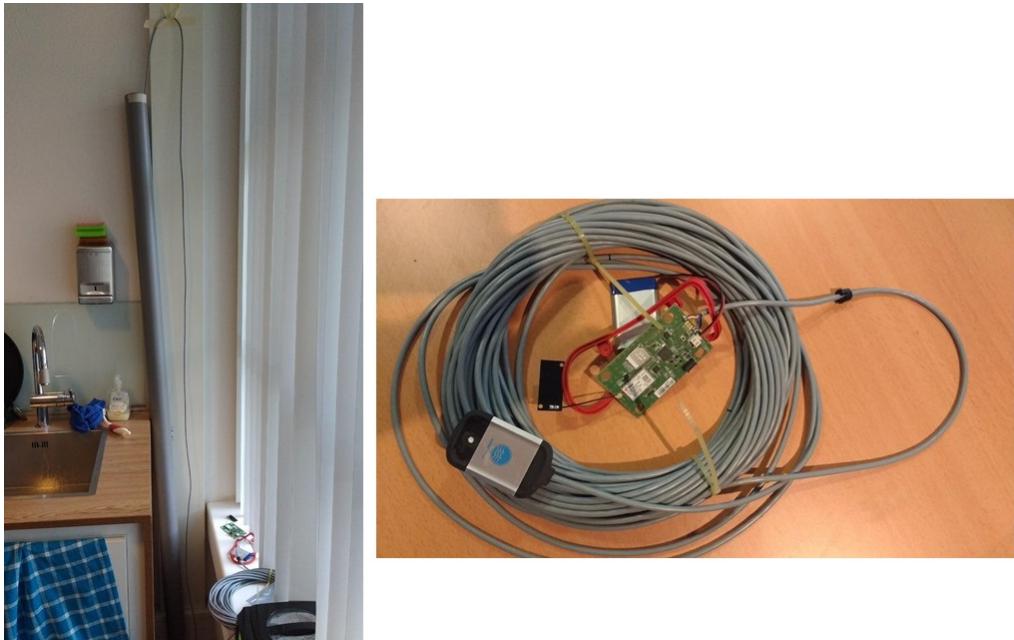


Figure B.1: 2 Meter long tube with water and SSU-WAP

A 2 meter long tube was filled with 1.98m of tap water. The WAP is on a 25m long cable connected to a SSU R5 board. SSU board was powered from a battery and transmits the measured values over 2G cellular to the MQTT broker.

The values were read by making an MQTT connection to the MQTT broker. Below is a picture of the MQTT client (MQTTfx) for one measurement. Format: device id, unix timestamp, bme280 temperature, bme280 pressure(hPa), WAP pressure(dPa), WAP temperature, batteryvoltage.

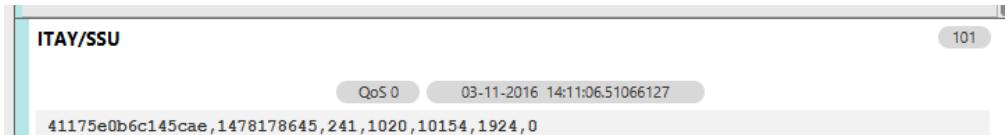


Figure B.2: MQTT message read from the MQTT client

As you can see there is an offset between WAP and BME280 pressure causing a measurement error, this must be zeroed out in the factory during electrical test. BME280 datasheet says accuracy is 0.1%, and stability is 0.1% per year. WAP (MS5803) datasheet says accuracy is 2% and stability is 2% per year.

Test methods

1. Depth is calculated from the pressure difference between BME280 and WAP
2. Same, but WAP is zeroed against the BME280

Formula: depth (m) = pressure/ (gravity * rho)

Pressure in Pa is given by the SSU-WAP

Gravity is 9.81m/s² on earth.

Rho is 1000kg/m³ for fresh water.

Test results

actual depth (m)	Method 1					Method 2		
	BME (Pa)	WAP (Pa)	delta (Pa)	depth (m)	rel	delta (Pa)	depth (m)	rel
0	102000	101690	-310	-0.03	0.00	0	0.00	0.00
0.5	102000	106510	4510	0.46	0.92	4820	0.49	0.98
1	102000	111330	9330	0.95	0.95	9640	0.98	0.98
1.5	102000	116240	14240	1.45	0.97	14550	1.48	0.99
1.98	102000	120910	18910	1.93	0.97	19220	1.96	0.99

Figure B.3: In yellow actual depth, in green the measured depth using method1 and method2.

When the offset between BME280 and WAP is zeroed out the measured accuracy is 2%, otherwise 8%.

Conclusion

The WAP pressure sensor is fit for its function to measure water pressure if the specification is 2%.

B.0.2 Validating SSU-WAP in 20m of water

See appendix C for the validation in 20m of (fresh) water.

B.0.3 Validating the WAP alone in 1m water

Gerard Hogenhout - Sept 2016

The WAP works with the MS5803 water pressure sensor. In this test it has been tested without SSU (datasheet https://cdn.sparkfun.com/datasheets/Sensors/Weather/ms5803_4ba.pdf)

It returns the water pressure over I2C in millibar, the actual value of the temperature and a checksum. It is known that 10 meter of water causes a pressure of 1 bar, so 10 cm of water causes a pressure of 10 millibar. This was verified with two WAPS(B.4). You can see that the sensors are not only the same, but also have linear curves. This means that the sensors do not have to be calibrated.

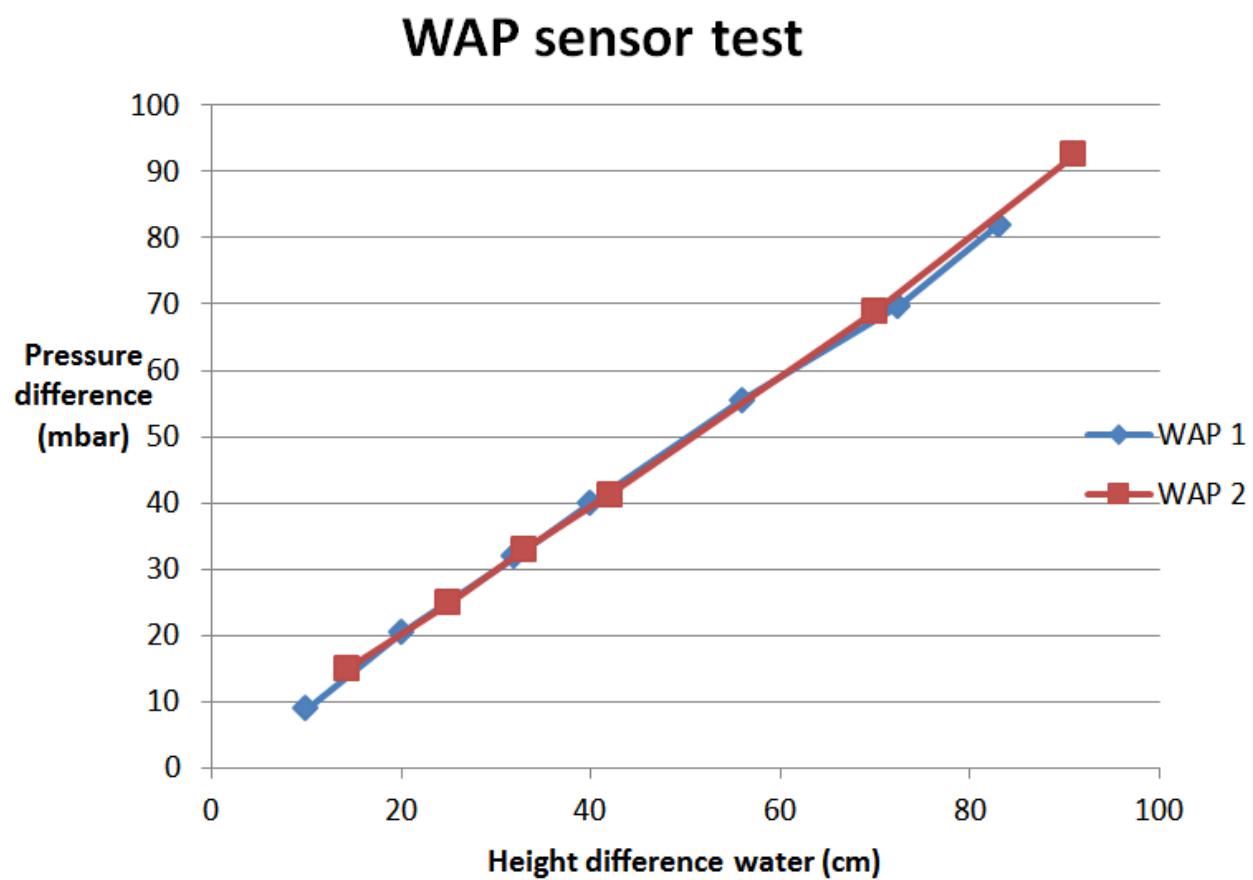


Figure B.4: Test report of two WAPs

Appendix C

Validation 20m

Validating SSU-WAP pressure sensor in 20m fresh water

19 Dec 2016. Jasper Sikken (SODAQ)

Summary

We WAP pressure sensor was tested at 5, 10, 15, and 20m deep from a kayak in the Maarsseveenscheplas with fresh water. Observed error is 0.4%, while the expected error as based on the WAP pressure sensor datasheet is 2%. At tested low temperature (7 degrees Celsius) the cable was a bit stiff; it did not go straight down and that introduced a measurement error. This was solved by adding a 1kg weight to the WAP.

Method

Formula: $\text{Depth(m)} = \text{delta pressure}/(\text{g} * \rho)$

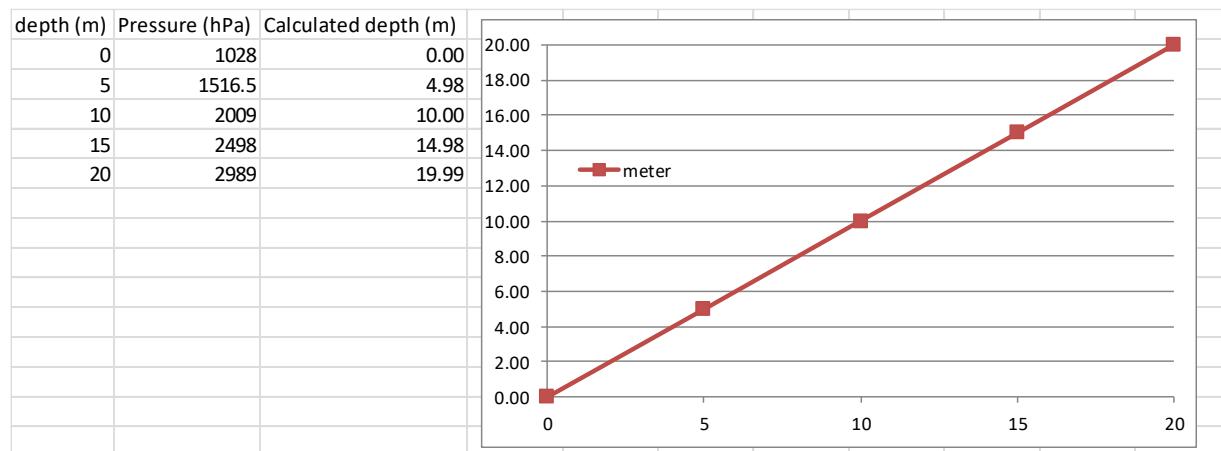
delta pressure = WAP pressure - BME280, pressure is in Pa, g = 9.81 m/s², rho = 1000kg/m³

Simplified formula: $\text{Depth(m)} = (\text{WAP pressure} - \text{BME280 pressure})/98.1$

Where WAP and BME280 pressure is in hPa

The measurements

Maarsseveenscheplas is 29m deep at its deepest point. At first I measured without heavy weight connected to the cable. I found that the cable is stiff and did not go straight down. Also the measurements were reading too low at 10, 15 and 20m depth (9.8m, 13.27m, 19.50m). Air and water temperature was about 7 degrees Celsius. In addition we believe the WAP must have rested on the bottom at 20m (12.31m). So we went further onto the lake and we added a heavy weight (about 1kg) to the WAP and then the measurements were correct. Suggestion is to use more flexible cable and/or add weight to the WAP. Every measurement we repeated to make sure the measured pressure was stable. Above water surface 1028, 1028 (BME280, WAP) as expected. Expected error is 2% and I measured 0.4% error.



Conclusion

Measurement results show that the WAP pressure accuracy is according to the specification.