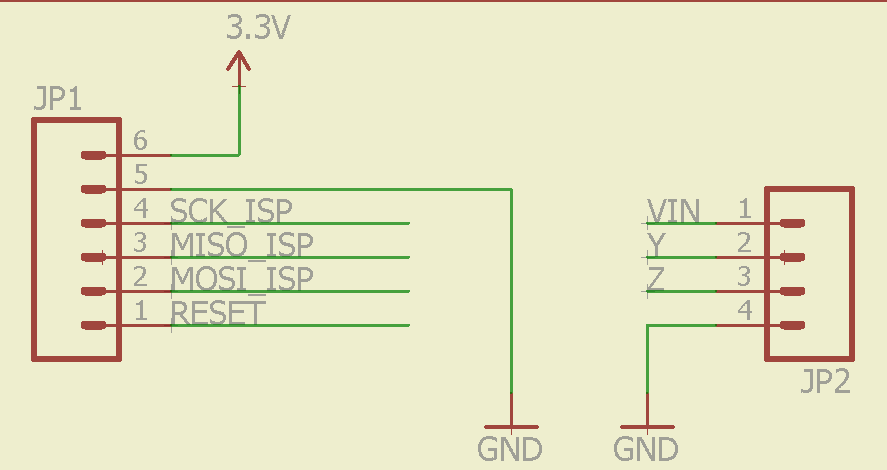
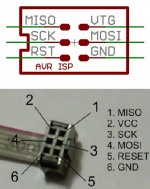
Programming the Water Pressure Sensor

We have two different parts that need different connections: the Progamming Mode and the Serial Mode.

# Programming Mode

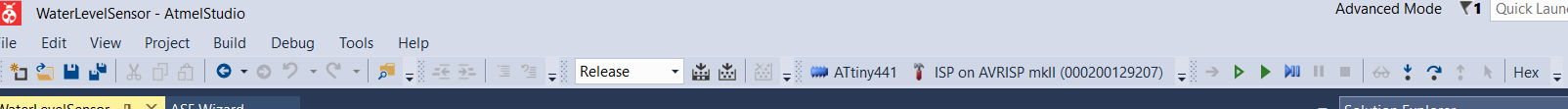
Connect the AVRMYISPMKII with the WAP-board with the pinout as below. Mind the profile on the 6-pole ISP-connector.



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 RS485 and the WAP are anyway connected!

Open the project in Atmel studio and compile it with F7 on “release” dropdown.

Select ATtiny441 in the program and the AVR MKII programmer as below:



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):

EXTENDED FUSE: 0xED

Set: HIGH FUSE: 0xD5

LOW FUSE: 0xC2

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

# Serial Mode

Connect the Uartsbee tot he RS485 breakout board from SparkFun, GND-GND, VCC-VCC, TX-RX-I, RX-TX-O. **Do not connect the CTS tot he RTS!!**

Y-pin of WAP: ±16.5kV HBM, ±7kV IEC61000 (contact method) ESD Protected inverting differential transmitter output. **Connect this pin to the A-pin of the RS485 breakout.**

Z-pin of WAP: ±16.5kV HBM, ±7kV IEC61000 (contact method) ESD Protected noninverting differential transmitter output. **Connect this pin to the B-pin of the RS485 pinout.**

Plug it in a usb-port, open Arduino IDE, select the right COM and open the serial monitor. Set the aud-rate to 4800. The Wap board outputs the pressure in millibars, the temperature in a 9 digit value and a checksum.