Watermeter Test Manual

Table of Contents

1 Overview

The Xerofill watermeter flow bench provides measured water flows through a watermeter under test. Sensors are provided for water pressure and temperature. A solenoid valve arrangement allows for the water flows to be started and stopped at selected times.

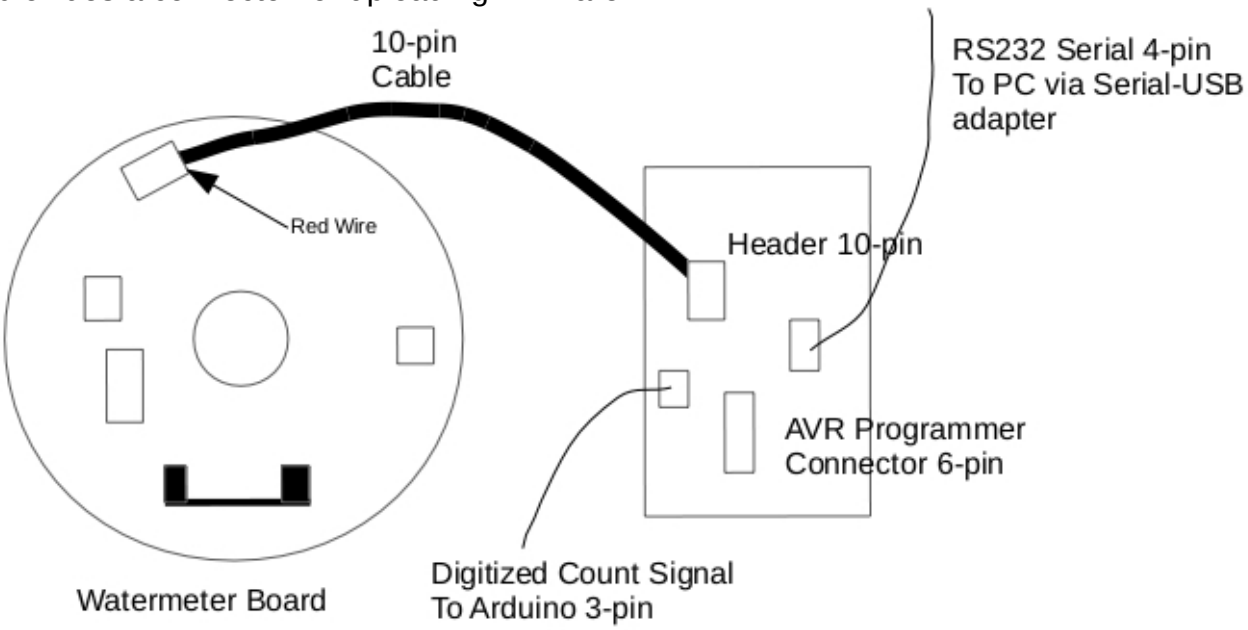
The purpose of the flow bench is to allow the watermeter to report its version of water flow and to compare this with measurements taken by an external processor. Measurements of temperature and water pressure are made at the same time.

This manual describes the components, test setup and measurement procedures for calibration of a watermeter.

2 Components

The test bench includes a water flow system incorporating a solenoid valve controlled by software.

The watermeter needs to have special test firmware loaded to count water flow and transmit the results serially via the test connector on board. An interface card converts the serially transmitted signals to RS232 for passing to an external PC. The card also gives access to the raw count signal and allows power to be passed to the watermeter. It provides a connector for uploading firmware.



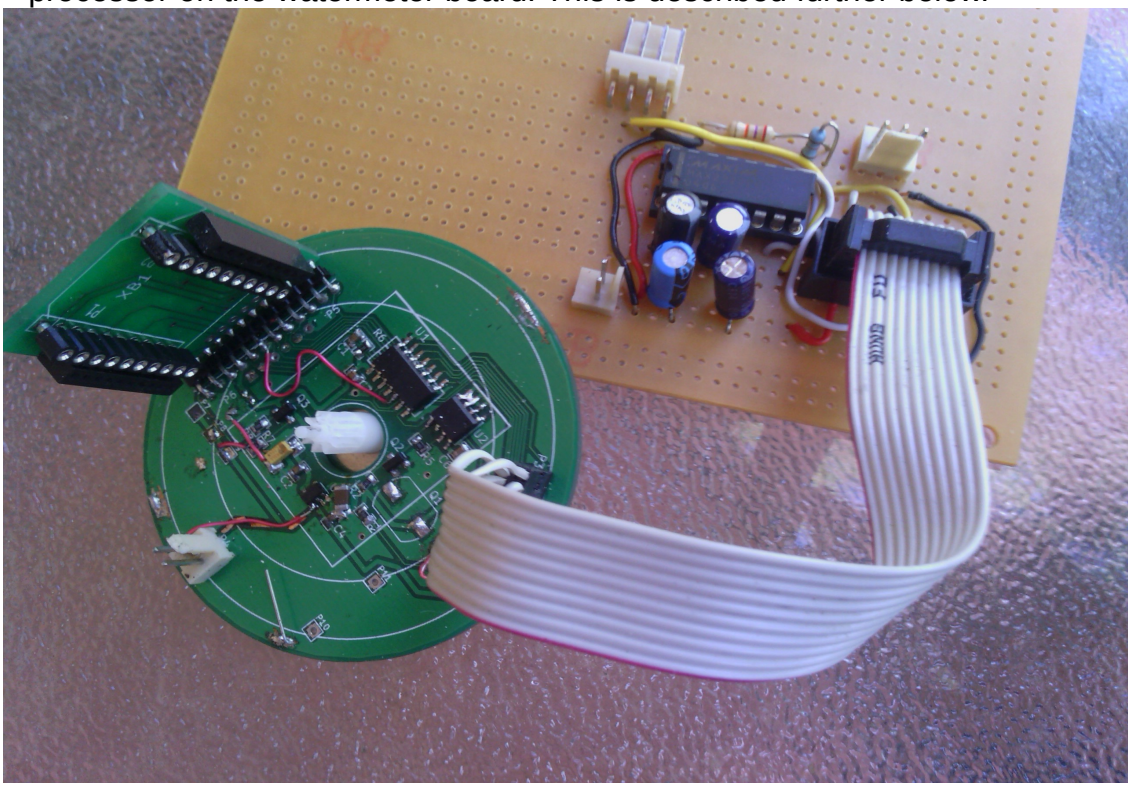
An Arduino board with a sensor shield performs a separate water flow measurement using the watermeter raw count signal. It also collects pressure and temperature measurements, the latter through an adapter board. The results are transmitted serially to an external PC

A program for the external PC is provided to collect, display and store the incoming data from the two serial sources, as well as control the water flow solenoid.

2.1 Interface Board

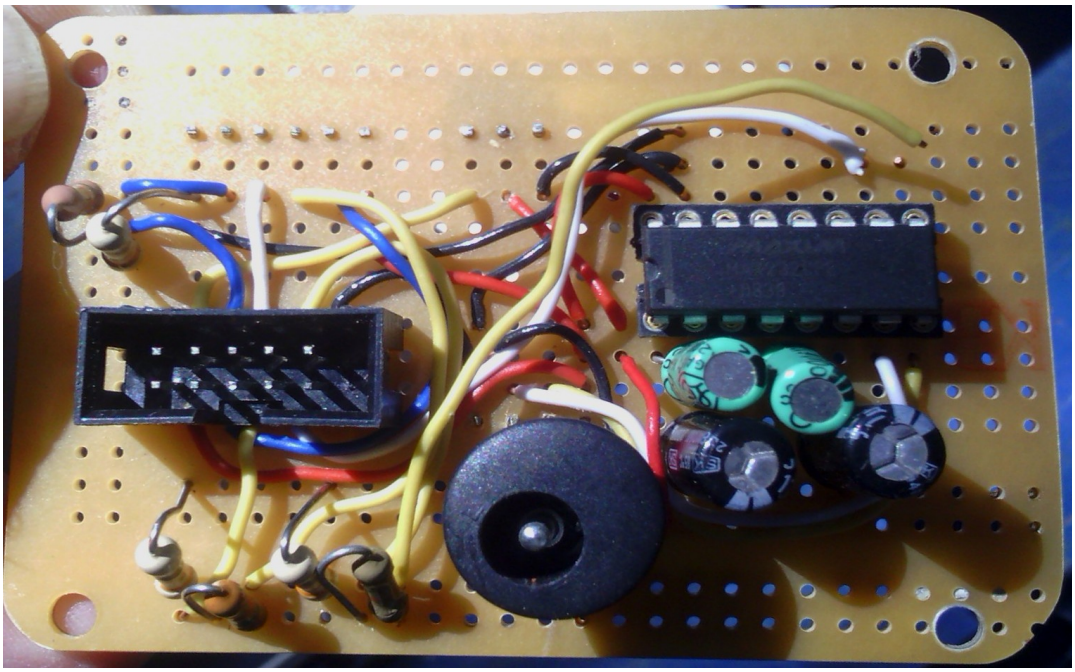
Two versions of the interface board are provided. One is free standing for lab testing, while the other is mounted on the testbench.

- The interface boards have a 10 pin header for connecting to the 10 pin header on the watermeter board. The interface board header is polarised and has the larger 0.1” spacing widely used in electronic equipment while the watermeter board header is unpolarised and has the smaller 2mm spacing. The cable connector has a red wire which must be oriented as shown above.
- The free standing interface board has a 2 pin polarised connector for connecting to the power, while the mounted interface board uses a 2.1” barrel socket. This powers both the interface board and the watermeter board.
- **The power to the boards must be 5V, with an absolute maximum of 6V, to avoid damage to the integrated circuit.**
- A three pin polarised connector is provided for connecting the digitised count signal from the watermeter board to the Arduino.
- A four pin polarised connector is provided for the serial signals sent from the AVR processor on the watermeter board. These will have been converted to RS232 levels by the integrated circuit on the interface board. A Serial-USB adaptor cable will normally required as most PCs now do not have serial interfaces. See below for interconnection information.
- A six pin polarised connector is also provided for uploading firmware to the AVR processor on the watermeter board. This is described further below.

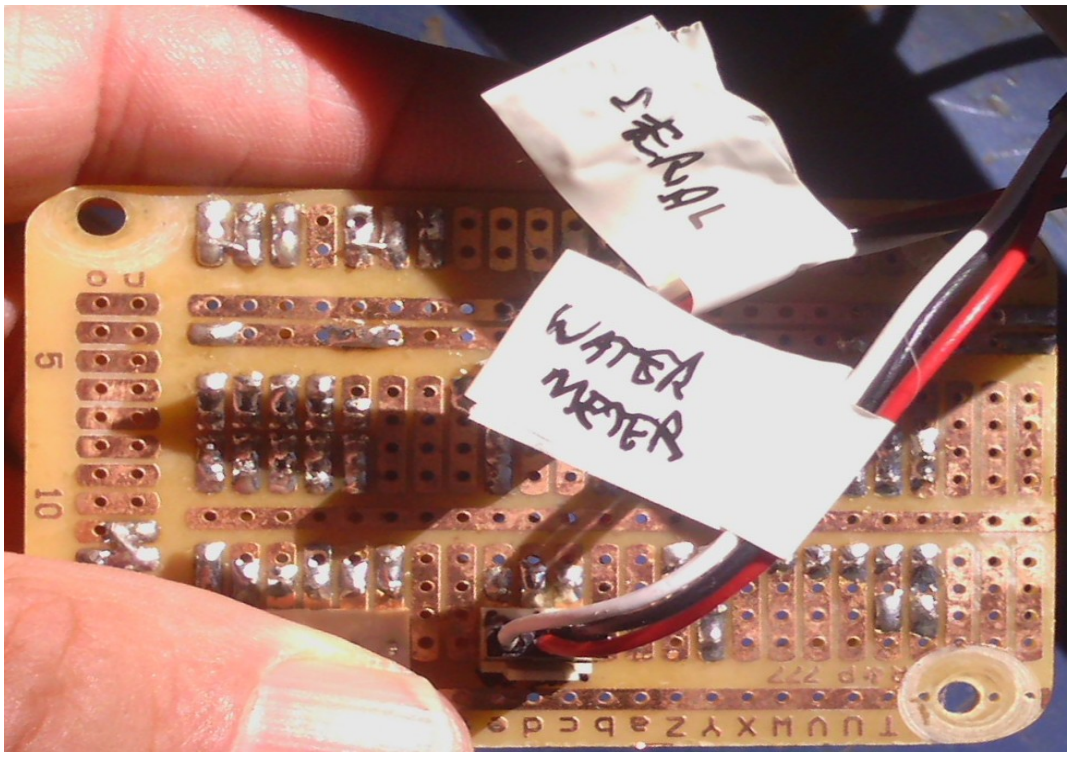


Free Standing Interface Card with 10-pin connector.

Note the orientation of the cable; the red wire appearing at the bottom.



Top View of the Mounted Interface Card



Bottom View of the Mounted Interface Card

2.2 Arduino Data Capture

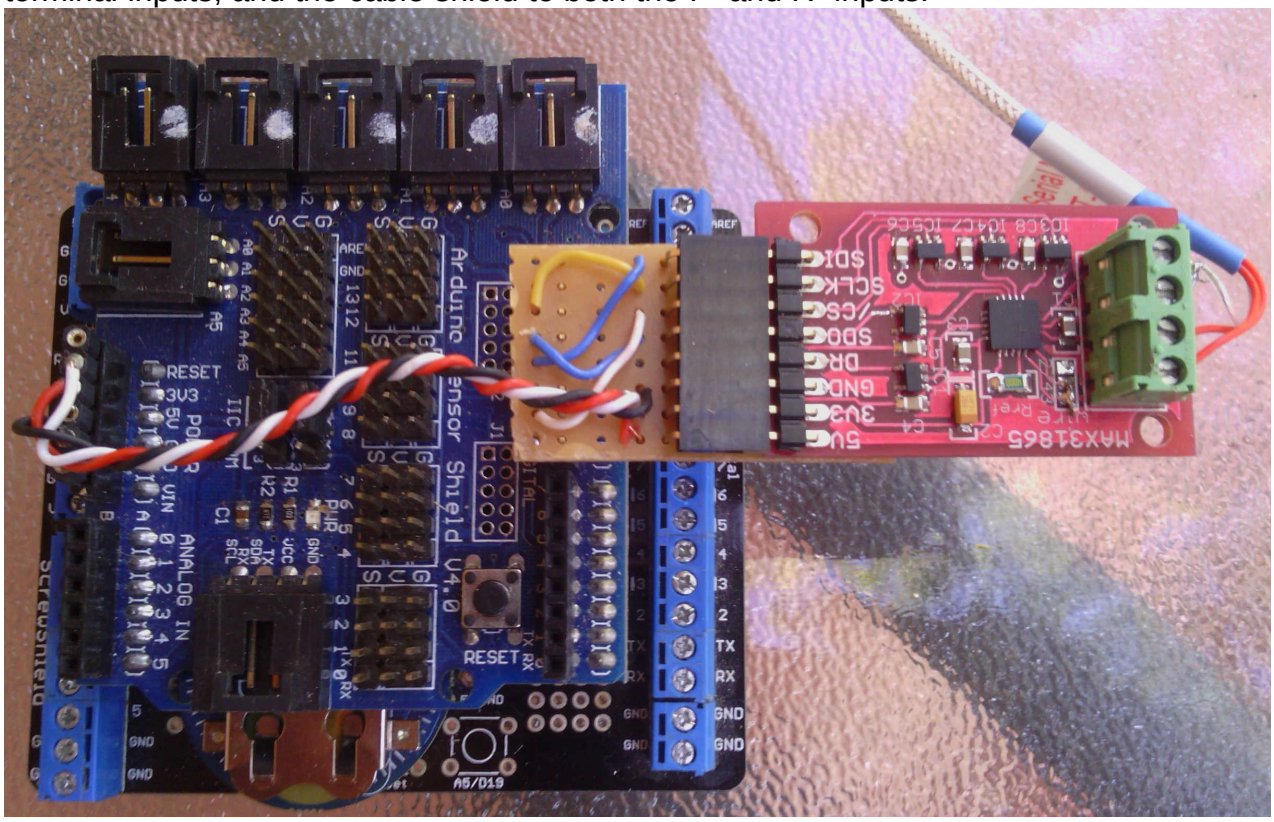
The Arduino Uno board with Sensor Shield V4 are used to interface the digitised counter signal, temperature and pressure, and communicate these to an external PC. The Arduino also carries an attached battery powered real time clock module Chronodot-RTC-DS3231 to provide time stamps for measurements.

2.3 Temperature Sensor Interface Board

The MAX31865 RTD-SPI breakout board from Maxim provides an SPI output that must be matched to the Arduino digital expansion connector. The prototyping board plus flying lead provides the following pin allocation:

MAX31865	Arduino
SDI	Digital 11
SCLK	Digital 13
CS	Digital 9
SDO	Digital 12
DR	-
GND	Power GND
3V3	Power 3V3
5V	Power 5V

The 3-wire sensor cable connects the two red wires in any order to the F+ and R+ screw terminal inputs, and the cable shield to both the F- and R- inputs.



Arduino UNO (hidden) with Screwshield and Chronodot-RTC-DS3231 with Sensor Shield and MAX3186 RTD-SPI board mounted on top.

2.4 Watermeter Firmware

A special test firmware for the watermeter AVR is provided that will transmit water flow counts serially at predefined intervals, nominally 0.5 seconds. The format is CSV with 6 blank fields followed by the count value in ASCII hexadecimal. Baud rate is 9600.

The firmware uses an interrupt on the digitised count signal to register rising transitions. A spike filter of 1ms duration is implemented. That is, if a rising edge transition is detected but lasts less than 1ms, it is discarded.

A watchdog timer is also implemented to reset the processor in the event of the program hanging. While not necessary in this test application, it is a reflection of code in the operational firmware that may allow possible faults to be detected.

3 Setup Procedure

1. Connect the 10-pin ribbon cable to the interface board ensuring that the slot on the plug matches the slot in the header. The 2mm header plug must be connected to the watermeter 10-pin header with the red wire in the correct position as shown above.

2. Connect the RS-232 serial output from the interface board to the PC via a serial-USB adaptor. These adaptors have a 9-pin Canon D male connector. There is a 9-pin Canon D female connector to 4-pin SIL male header adaptor to match the cable from the interface board. Ensure that the orientation is correct with the black wire connecting to the outer wired pin on the adaptor (see above).

3. Connect the Arduino to the PC via the USB cable. Under Linux this will normally appear as /dev/ttyACM0 as long as that device doesn't already exist. Otherwise it may appear as /dev/ttyACM1.

4. Connect the 3-wire count signal cable from the interface board to digital input 6 of the Arduino Sensor Shield. Ensure the black wire connect to the G (ground) pin and the white wire to the S (signal) pin.

5. Connect a 5V source to the barrel power connector on the interface board. Note that the voltage must not exceed 6V on this connector.

6. Plug the MAX31865 RTD-SPI board into the Arduino digital expansion connector (via the Sensor Shield), and its floating power connector to the Arduino expansion power connector according to the photograph shown above. The black wire attaches to ground, the red wire to 5V and the white wire to 3.3V.

7. Connect the temperature sensor to the MAX31865 RTD-SPI board. The two red wires can be any order to F+ and R+, with the cable shield connected to both F- and R-.

8. Connect the pressure sensor to analogue input A1 of the Arduino Sensor Shield. Ensure the black wire connect to the G (ground) pin and the white wire to the S (signal) pin.

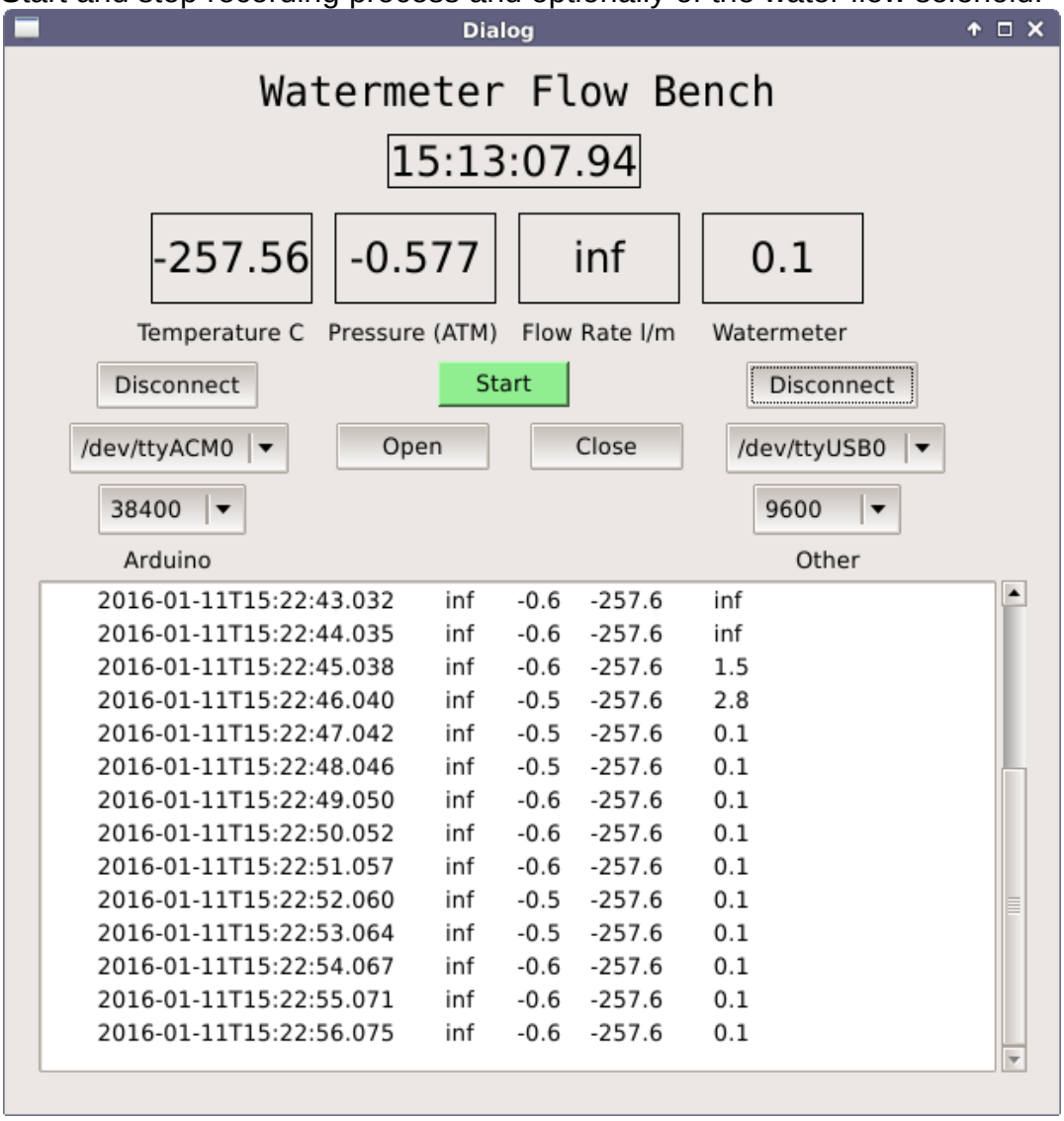
9. Connect the 3-wire solenoid cable to digital input 5 of the Arduino Sensor Shield.

10. A push button switch may be connected to digital input 4 of the Arduino Sensor Shield to allow a manual rather than a software start.

4 Graphical User Interface

The GUI software is provided to simplify collection of incoming data streams and control of the the measurement process.

- Opening of serial port connections to the Arduino and watermeter.
- Time display of temperature, pressure and water flow provided for monitoring purposes.
- Open and close of a save file. This save file will start recording when the green Start button is clicked and stop when the red Stop button is clicked. It will contain raw counts from the watermeter as well as timestamps, measured temperature and pressure.
- Start and stop recording process and optionally of the water flow solenoid.



In the image above:

- The time display is that of the PC, not the Arduino. The time stamp from the Arduino is stripped replaced with local time. The Arduino time can be incorrect if the RTC doesn't have a battery, and the records from the watermeter do not have any timestamp.
- Temperature and pressure are shown above (no sensors were connected, hence the odd values).
- Flow rate relates to a flowmeter that has now been removed. Both flow rate displays will show “inf” if no previous count values have been received.

- Watermeter flowrate is derived from the counts and a precomputed scaling factor.
- The left side controls relate to the Arduino serial interconnection. The Connect button must be clicked before the connection is made and results are shown.
- The right side controls relate to the watermeter, in this case on a serial-USB adapter. Note that a direct serial connection can also be selected if such exists on the PC in use.
- Open and Close relate to a file for storing the incoming results. Note that the Arduino records are stored but not displayed in this version.
- The green Start button will activate the solenoid and begin recording of results. This will then change to a red Stop button which will turn off the solenoid and suspend recording.

5 AVR Programming

The AVR processor on board the watermeter must be programmed using the supplied programmer board and the 6-pin connector on the interface board. This programmer is a custom build circuit that requires its own GUI. This requires a Linux installed PC. All relevant code is provided at https://github.com/ksarkies/AVR-Serial-Programmer. Detailed descriptions of the GUI, its installation and operation, are provided at http://www.jiggerjuice.info/electronics/projects/serialprogrammer/firmware-bootloader.html.

In the long term it is intended that the operational firmware in the watermeter shall incorporate a feature that can be used to provide a test mode for use with the flow bench.

For the moment the following procedure can be used to install and operate the programmer.

1. Download https://github.com/qextserialport/qextserialport/archive/master.zip
2. In a suitable location, e.g. a directory called “compile” create a subdirectory with the name “auxiliary”.
3. Unzip the above ZIP file into the subdirectory.
4. Download https://github.com/ksarkies/AVR-Serial-Programmer/archive/master.zip. This is the ZIP file containing the programmer code.
5. Unzip the subdirectory avr-serial-programmer-pc only into the selected directory, called “compile” above, and in a terminal change to the subdirectory “compile/avr-serial-programmer-pc“.
6. Ensure that gcc, qt4-developer and qt4-qmake are installed.
7. Execute the commands
    - $ qmake-qt4
    - $ make
8. As root copy the binary avrserialprog to a suitable place such as /usr/bin.

The GUI can now be invoked by typing avrserialprog from the command line. Refer to the above website page for instructions on its use.

Alternatively a compiled hex file, e.g. test.hex, can be programmed directly from the command line with

$ avrserialprog -n -b 19200 -P /dev/ttyUSB0 -w test.hex -v -d -x

The baud rate of 19200 is the default and is used by the programmer circuit.