Low-Cost Chlorine Monitoring System Rev1.4

User Manual Rev1.0

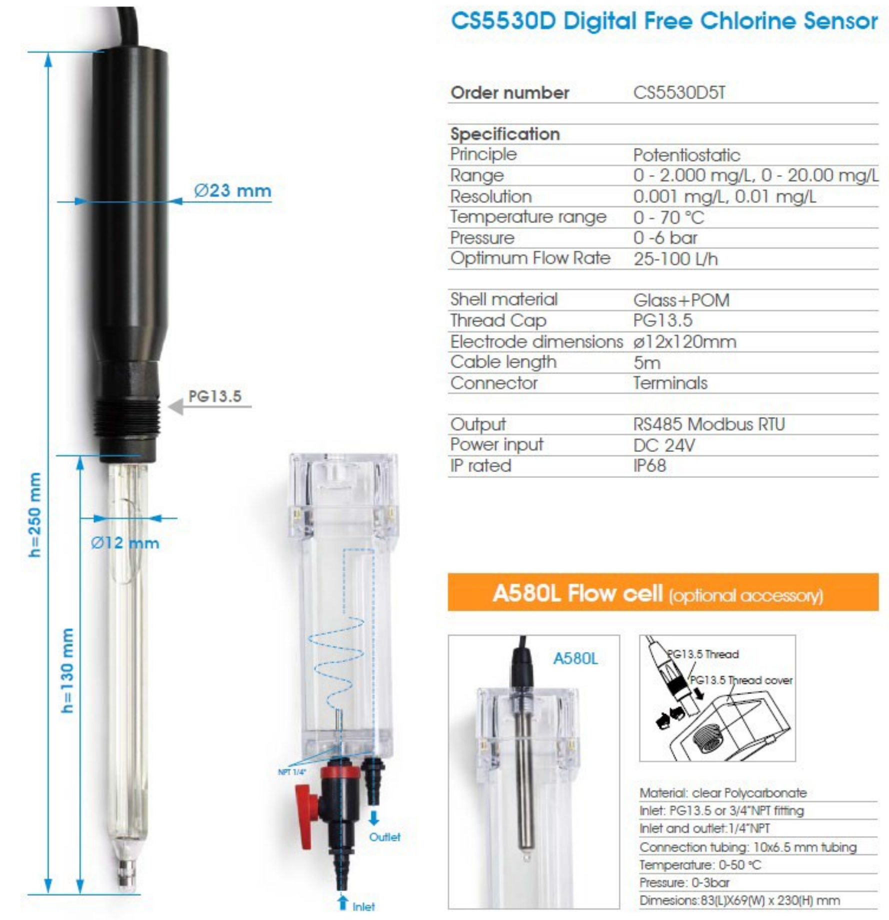
Summary

A fully automated system, controlled by a custom-designed Arduino-compatible PCB is used for free residual chlorine monitoring using a certified, low-cost commercially available, amperometric FCL sensor in combination with a dedicated flow cell, local data storage on a microSD card and a Bluetooth Module for the data damp of the logged measurements (data) from the SD card. The flow cell provides an adjustable constant flow rate of 0.25 L/min to the sensor with a drain to "waste", thus making the system independent of pipe network flow or pressure. The water "loss" is 0.5L per measurement session with a duration of 130 seconds and can be collected in a water canister with a tap and used even as drinking water since the measurement does not affect its quality. So, for the default 4 measurements daily (every 6 hours: 6AM, 12AM, 6PM, 12PM) 2 Liters will be collected in the water canister per 24hours.

The selected digital free residual chlorine sensor (CS5530D) provides reliable operation and infrequent, easy to perform maintenance (no buffer solutions, no membrane, etc.) thus allowing long unattended operation at low operating cost. The fully automated system is powered from a reliable Li-Ion Battery (NCR18650, 3.7V, 2.9Ah), which has to be replaced with a newly charged one when depleted (every 1 or 2 months according to the condition of the battery). The system mainly stays in Sleep (low-power) Mode with a current consumption of 0.4mA and wakes up only for a short period of time (130sec) to perform the measurements.

During the measurement session the system samples the water every 5 seconds (user-defined) and an average value of 21 measurements (105sec) is locally stored on an SD card with a timestamp. The complete system (Main PCB, Flow Cell, and Sensor) is housed in a waterproof IP65 plastic enclosure. All the components (hardware and software) are readily available off the shelf and are of economic cost. Additionally, the system allows for simple in-field calibration (ZERO and SLOPE) even without the use of a laptop, where only water samples with free residual chlorine content of 1mg/L and 0mg/L are required.

The FCL electrode



CS5530D Digital Free Chlorine Sensor

Order number	CS5530D5T
Specification	
Principle	Potentiostatic
Range	0 - 2.000 mg/L, 0 - 20.00 mg/L
Resolution	0.001 mg/L, 0.01 mg/L
Temperature range	0 - 70 °C
Pressure	0 -6 bar
Optimum Flow Rate	25-100 L/h
Shell material	Glass+POM
Thread Cap	PG13.5
Electrode dimensions	ø12x120mm
Cable length	5m
Connector	Terminals
Output	RS485 Modbus RTU
Power input	DC 24V
IP rated	IP68

A580L Flow cell (optional accessory)

Material: clear Polycarbonate
Inlet: PG13.5 or 3/4"NPT fitting
Inlet and outlet:1/4"NPT
Connection tubing: 10x6.5 mm tubing
Temperature: 0-50 °C
Pressure: 0-3bar
Dimesions:83(L)X69(W) x 230(H) mm

*Figure 1: Main specifications and dimensions of the selected sensor (*offered only with range 0-20mg/L and opt. flow rate 10-30L/h).*

- Isolated power and output signal to ensure electrical safety
- Built-in protection circuit of power supply & communication chip
- Good environmental resistance and easier installation and operation
- Platinum sensor, three-electrode method
- Body Material: Glass
- Working pressure: 0-6 bar
- Dimensions: length 110mm, 12mm diameter
- Suitable for pipe installation (PG13.5 thread)
- 4 line electrical interface (2 power lines, 2 RS-485 signal lines)
- Power Supply: 9-24 VDC
- Current consumption: <30mA
- Output: isolated current output RS485 (RS485 Modbus RTU)
- Chlorine Measurement range: 0 – 20 mg/L
- Resolution: 0.01 mg/L
- Accuracy: ±1%
- Stability: ±0.01 mg/L/24h

- Automatic real-time temperature compensation: 0 - 70 °C, resolution 0.1 °C
- Optimum Flow Rate: 10-30L/h (0.16-0.5 L/min)
- Low maintenance cost
- IP68 Protection

Main System Board

The Arduino-compatible custom-designed PCB is based on a fully optimized design according to the specific project aims, as shown in Fig. 2. All system modules and components are well denoted in the two schematic sheets, the PCB design.

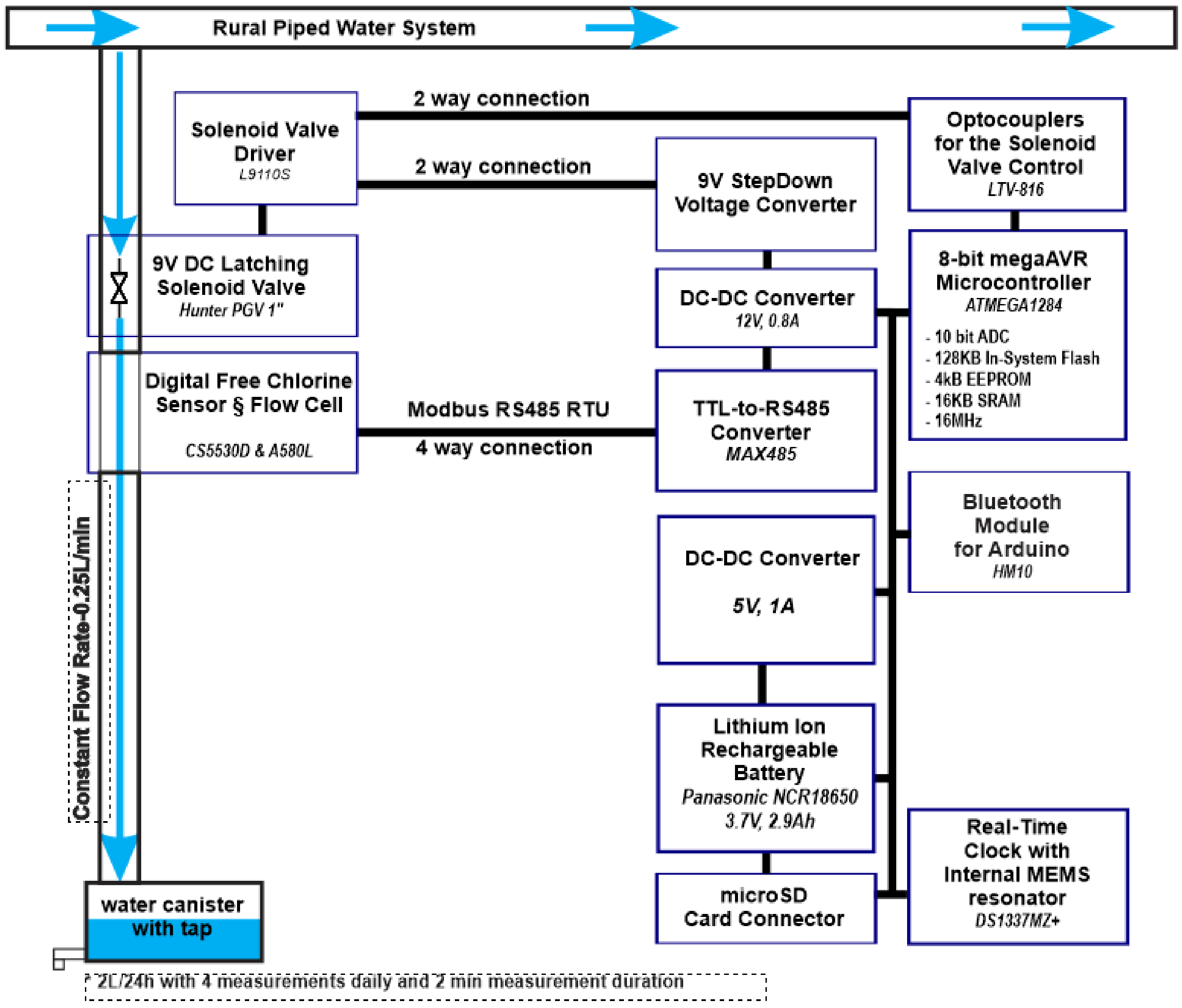
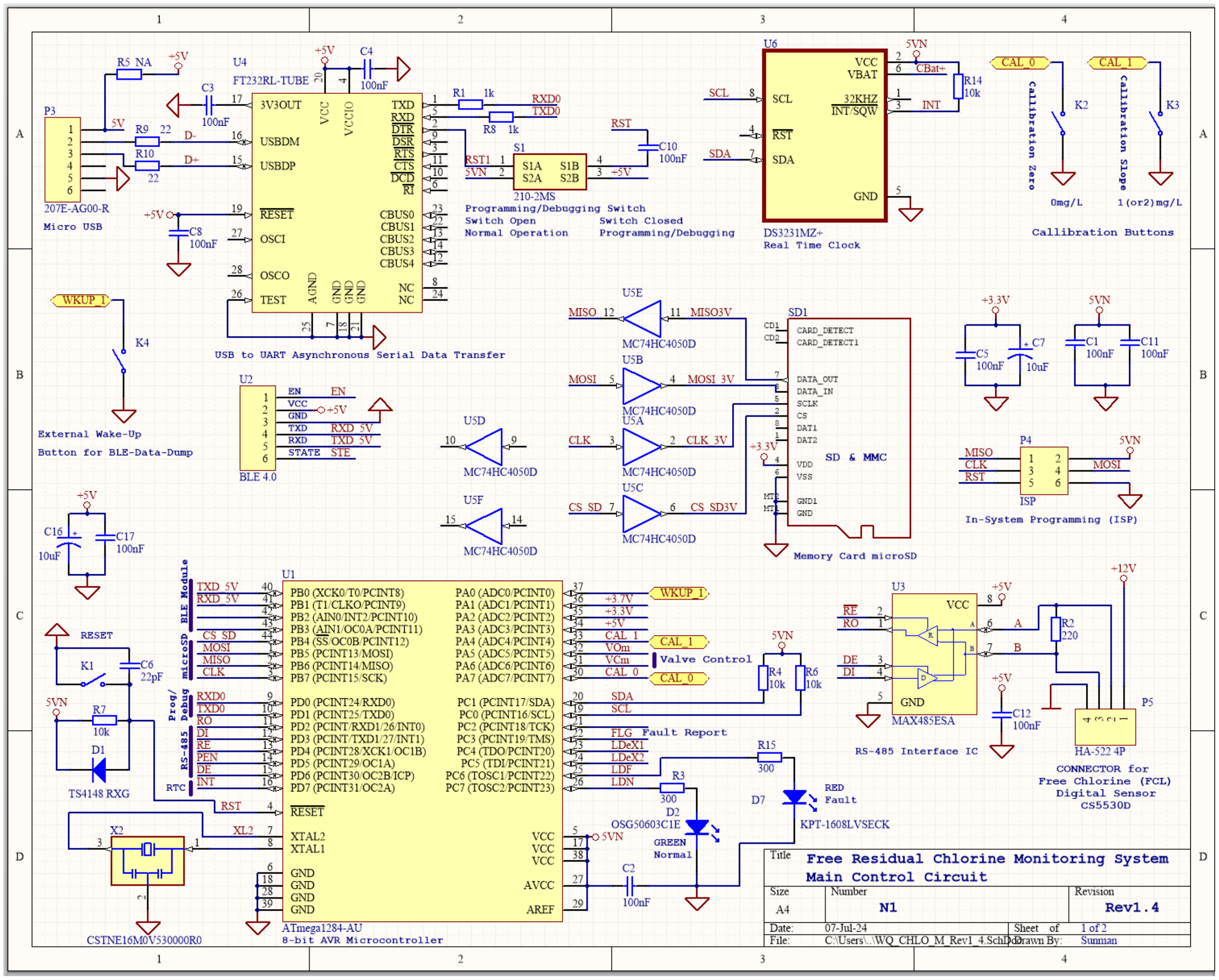


Figure 2: Block diagram of the free chlorine monitoring system

Hardware

The two main schematic sheets depicting the complete circuit of the system are shown in Fig. 3 & 4, where the PCB design is shown in Fig. 5.



USB to UART Asynchronous Serial Data Transfer
Programming/Debugging Switch
DS3231MZ+ Real Time Clock
Callibration Buttons
External Wake-Up Button for BLE-Data-Dump
Memory Card microSD
In-System Programming (ISP)
ATmega1284-AU 8-bit AVR Microcontroller
RS-485 Interface IC
CONNECTOR for Free Chlorine (FCL) Digital Sensor CS5530D
Free Residual Chlorine Monitoring System
Main Control Circuit
Rev1.4

Figure 3: Main schematic of the free chlorine monitoring system.

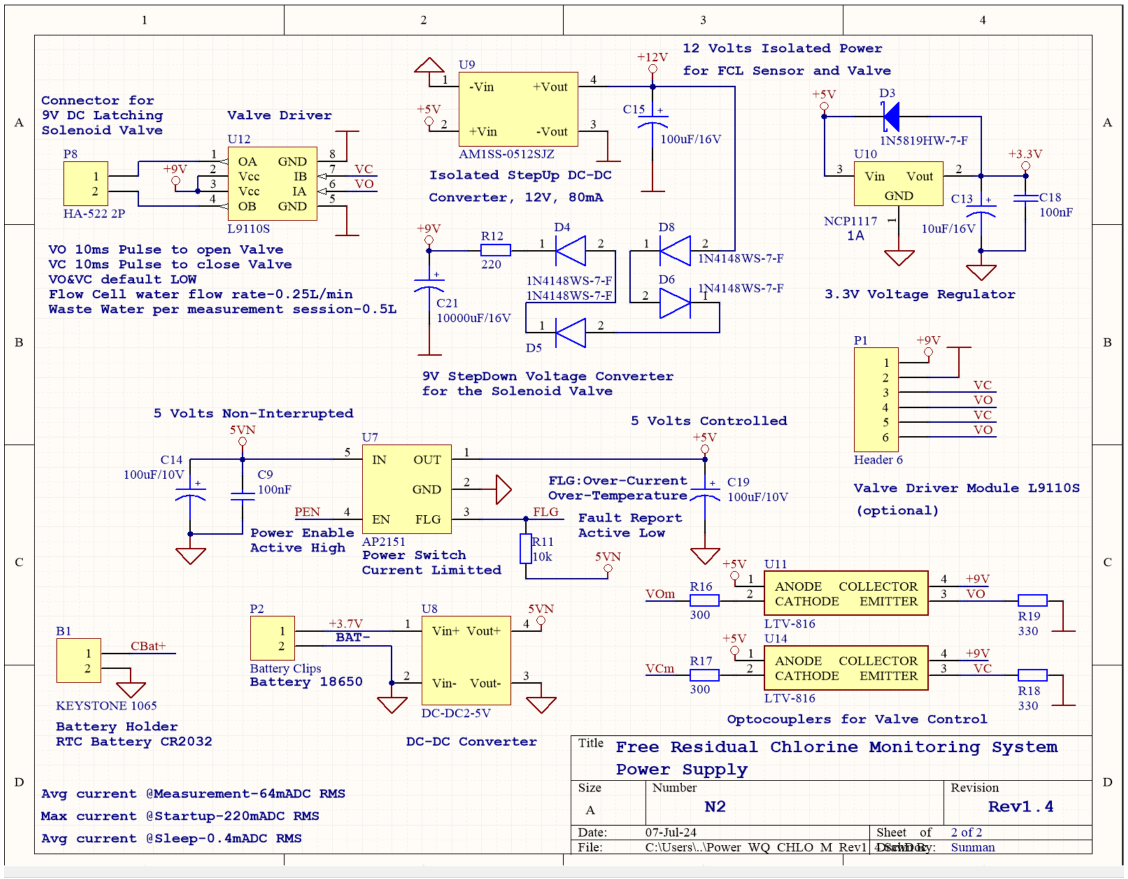

Connector for 9V DC Latching Solenoid Valve
Valve Driver
P8
HA-522 2P
U12
L9110S
OA
OB
Vcc
GND
IA
IB
VC
VO
+9V
VO 10ms Pulse to open Valve
VC 10ms Pulse to close Valve
VO&VC default LOW
Flow Cell water flow rate-0.25L/min
Waste Water per measurement session-0.5L
U9
AM1SS-0512SJZ
-Vin
+Vin
+Vout
-Vout
+5V
+12V
C15
100uF/16V
Isolated StepUp DC-DC Converter, 12V, 80mA
12 Volts Isolated Power for FCL Sensor and Valve
D3
1N5819HW-7-F
U10
NCP1117 1A
Vin
Vout
GND
+3.3V
C13
10uF/16V
C18
100nF
3.3V Voltage Regulator
R12
220
C21
10000uF/16V
D4
D5
D6
D8
1N4148WS-7-F
9V StepDown Voltage Converter for the Solenoid Valve
P1
Header 6
Valve Driver Module L9110S (optional)
5 Volts Non-Interrupted
5VN
C14
100uF/10V
C9
100nF
U7
AP2151
IN
OUT
GND
EN
FLG
PEN
Power Enable Active High
Power Switch Current Limitted
5 Volts Controlled
FLG:Over-Current Over-Temperature
Fault Report Active Low
C19
100uF/10V
R11
10k
U11
U14
LTV-816
ANODE
CATHODE
COLLECTOR
EMITTER
VOm
VCm
R16
R17
300
R18
R19
330
Optocouplers for Valve Control
B1
CBat+
KEYSTONE 1065
Battery Holder RTC Battery CR2032
P2
+3.7V
BAT-
Battery Clips
Battery 18650
U8
DC-DC2-5V
Vin+
Vout+
Vin-
Vout-
DC-DC Converter
Avg current @Measurement-64mADC RMS
Max current @Startup-220mADC RMS
Avg current @Sleep-0.4mADC RMS
Title Free Residual Chlorine Monitoring System Power Supply
Size A
Number N2
Revision Rev1.4
Date: 07-Jul-24
Sheet 2 of 2
File: C:\Users\..\Power_WQ_CHLO_M_Rev1
Drawn By: Sunman

Figure 4: Schematic of the power supply of the chlorine monitoring system.

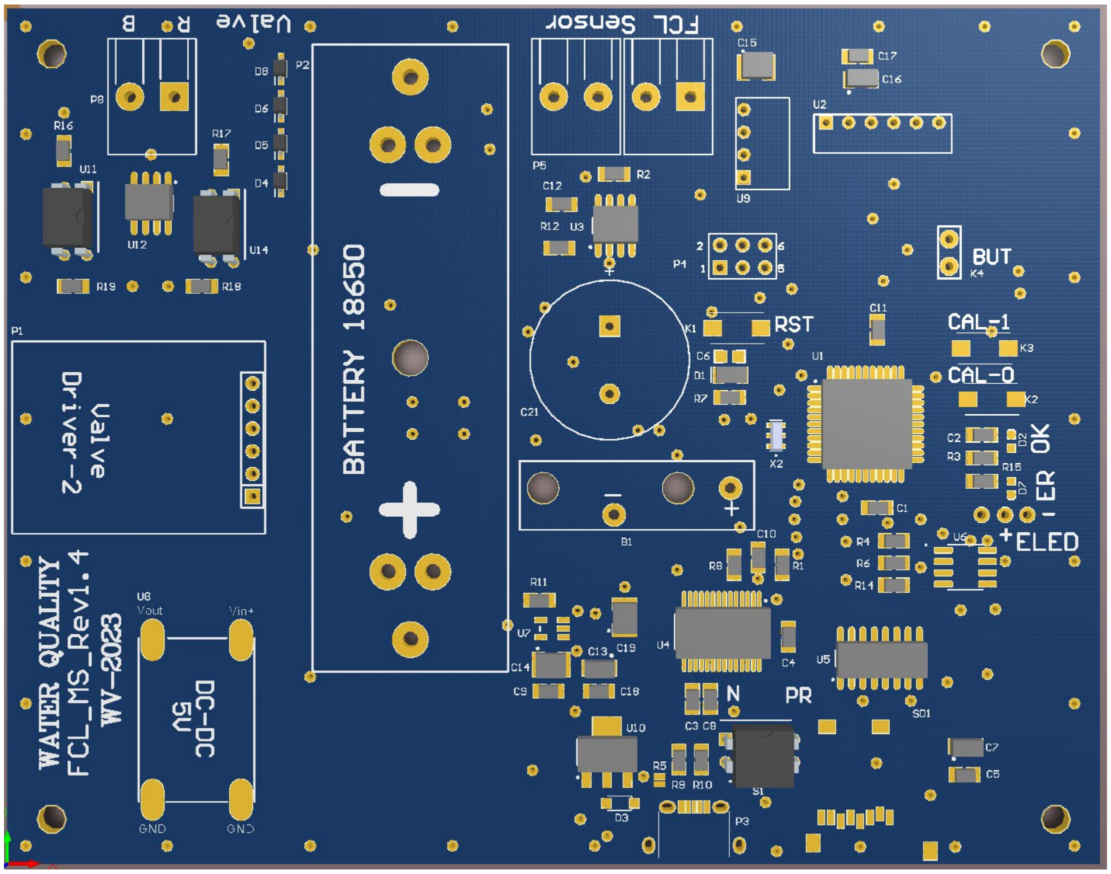


Figure 5: PCB design of the free chlorine monitoring system.

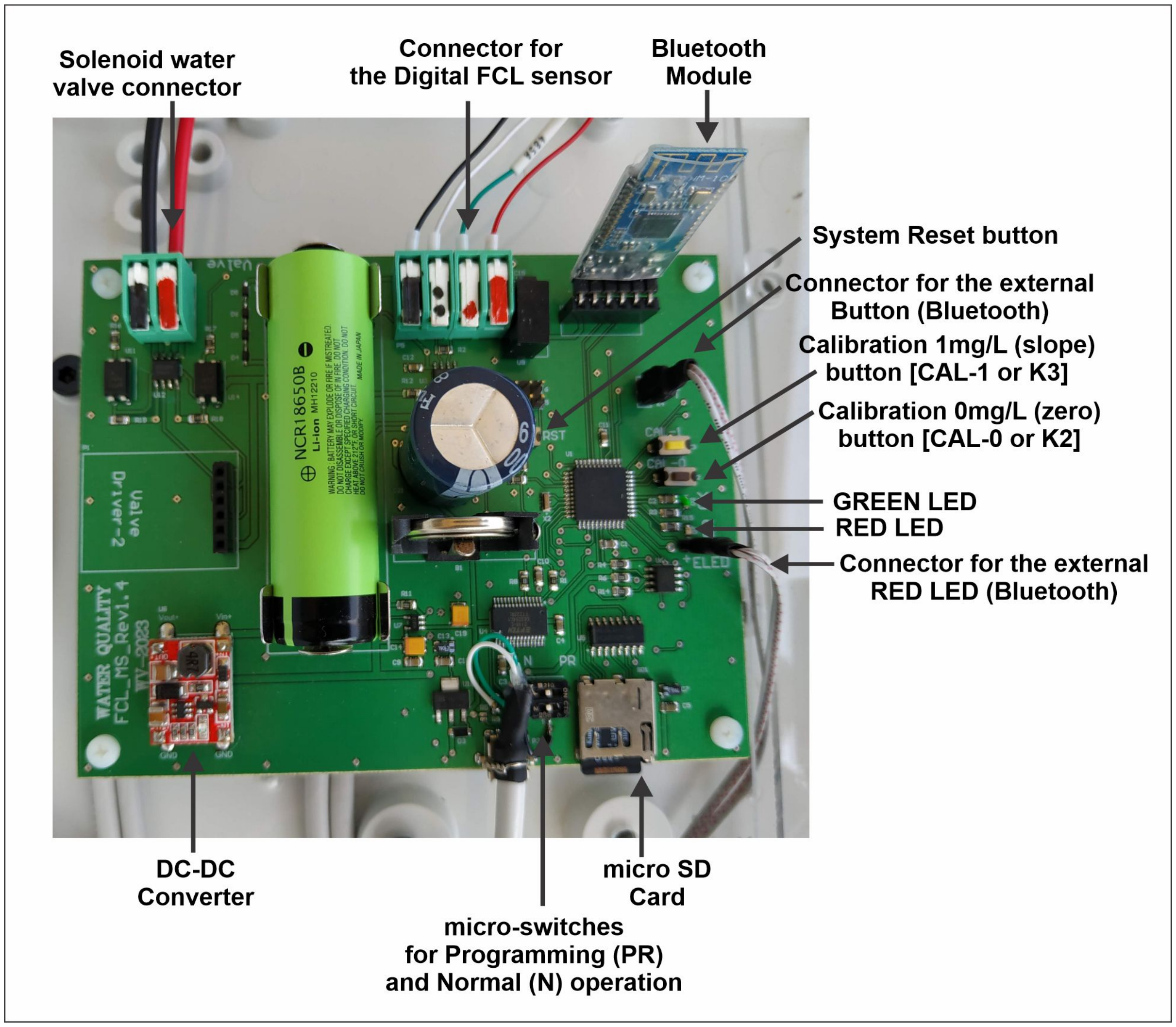

Solenoid water valve connector
Connector for the Digital FCL sensor
Bluetooth Module
System Reset button
Connector for the external Button (Bluetooth)
Calibration 1mg/L (slope) button [CAL-1 or K3]
Calibration 0mg/L (zero) button [CAL-0 or K2]
GREEN LED
RED LED
Connector for the external RED LED (Bluetooth)
DC-DC Converter
micro SD Card
micro-switches for Programming (PR) and Normal (N) operation

Figure 6: Photo of the final PCB prototype

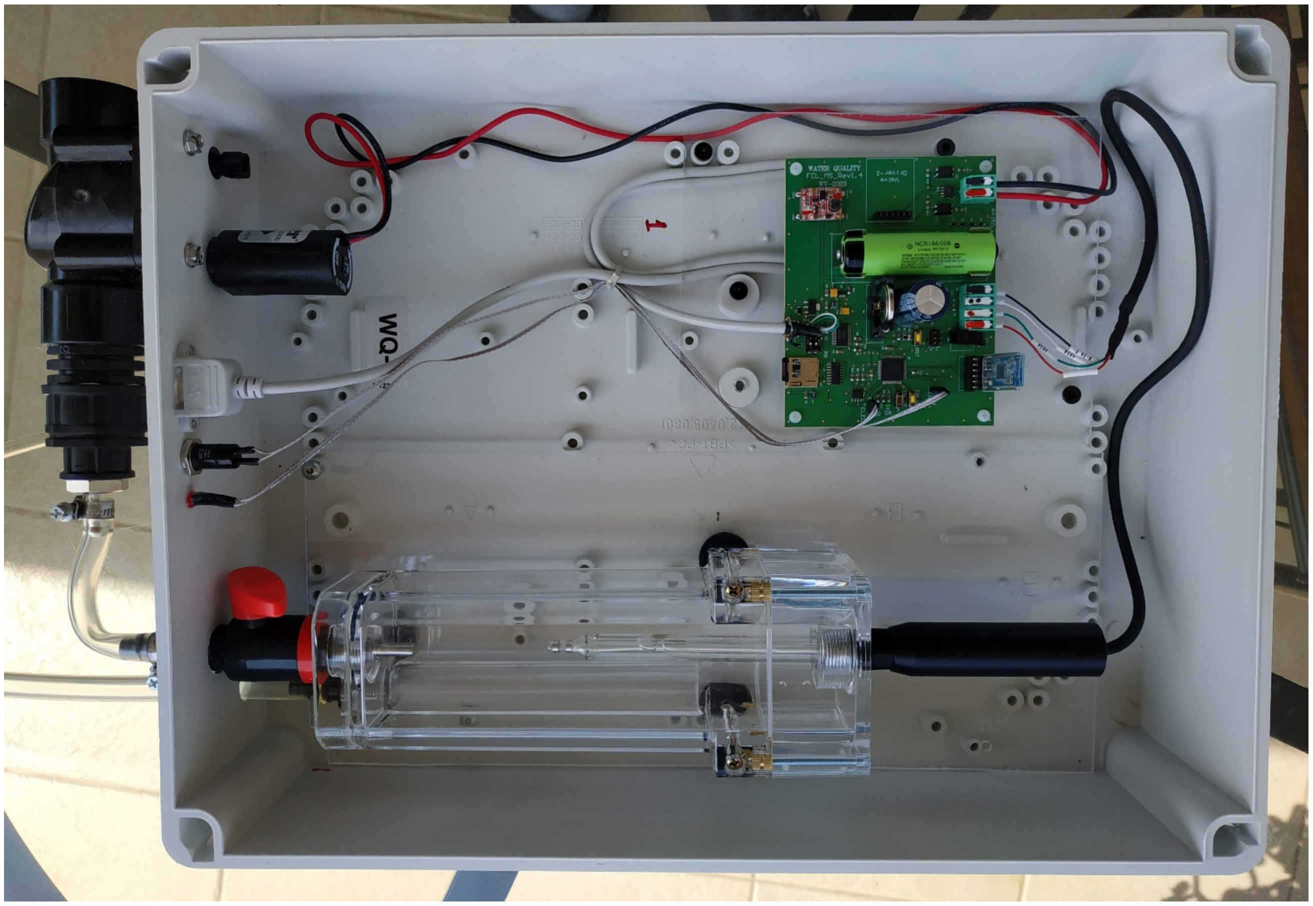


Figure 7: Photo of the final working system prototype (without front/top cover)

Application Firmware

The Application Firmware is written as Arduino “Sketch”, using standard Arduino syntax and within the Arduino GUI. The general term "sketch" is a synonym for Arduino program regardless of the host compile environment. First, a bootloader is programmed/burned through ICSP using Arduino GUI and a compatible programmer (AVRISP mkII in our case). After the programming is done via USART (serial programming) and the programmer is not needed.

In order to (re)program the main board with an updated firmware:

- Set the two micro-switches near the USB port on the main PCB board to the "PR" (**PRogramming Mode**) position; this automates the required reset toggle during programming.
- Compile and upload the firmware to the device via USB (using the Arduino IDE; tested with version 2.2.1)
- After the device has been (re)programmed, set the two micro-switches back to the "N" (**Normal Mode**) position for normal power-saving operation.

After Power On/Reset or WakeUp, the system configures I/O ports, closes preventively the Solenoid Valve and starts initializing the peripherals (USB, SD card, Bluetooth module, FCL sensor, RTC, etc.), and sends the related debug messages. After System Reset (or Power On) and if the initialization is successful, the system enables the RTC alarm wakeup interrupts and sets the RTC alarm according to the selected time interval between the FCL measurements, based on the selected number of automated readings per 24 hours. After the SD card is stopped, to prevent eventual file corruption and goes into Sleep (low-power) Mode until woken Up by RTC Alarm or Manually by short or long-pressing pushbuttons CAL-1 or CAL-0 as printed on the PCB (or K3, K2 as denoted in the schematic design). A short press of either CAL-1 or CAL-0 will wake up the system and start an FCL measurement session. After WakeUp, the system is initializing again all the peripherals, and if all is OK and the battery level is high enough, the system opens the solenoid valve, waits 15 seconds for Flow Cell conditioning, and starts reading the measured FCL values. The flow cell provides a controlled constant flow rate of 0.25 L/min to the sensor with a drain to "waste", thus making the system almost independent of pipe network flow or pressure. The water "loss" is 0.5L per measurement session with a duration of about 2 minutes and can be collected in a water canister with a tap and used even as drinking water since the measurement does not affect its quality. So, if 4 measurements are needed daily, 2 Liters of water will be collected in the canister for 24 hours. When 21 measurements (user-defined) are obtained, an average value is calculated, and written to the SD card. Subsequently, the system goes again into Sleep (low-power) Mode. In this mode, it consumes about 0.43mA and if up to 4 measurements per day are selected, it may work on battery up to 2 months before recharging is needed.

Calibration

The main PCB employs three pushbuttons: RST (or K1) for system Reset, and CAL-0 (or K2) and CAL-1 (or K3) for the Calibration of the Digital Free Chlorine Sensor (CS5530D) when long-pressed and for starting a measurement session if short pressed. Before the Calibration it is recommended to connect to a Laptop/PC to the system using an USB Cable, to serve as terminal monitor console in order to be able to follow all the debug messages during the calibration procedure.

The Calibration Procedure can be started only after a System Reset (or PowerOn) which can be triggered by pressing the RST pushbutton or inserting a new Battery. If the following initialization of the peripherals is successful, the system enters into a Calibration Waiting Mode and waits for 30 seconds for an eventual Calibration process to be started by long pressing any of the CAL-0 or CAL-1 buttons. The Calibration Waiting Mode is indicated by a specific LED pattern: GREEN LED blinks twice, waits for 1 second and repeats (see LEDs Patterns). After the Calibration Waiting Mode (30sec) has elapsed and none of the Calibration buttons has been pressed, the system goes into Sleep (low-power) Mode until woken Up by RTC Alarm or Manually by short or long-pressing any of the pushbuttons.

Calibration of the Slope point (1mg/L FCL)

Inserting the sensor in a water sample with known free chlorine concentration ("S") and pressing CAL-1 button for more than 3 seconds will initiate a Calibration Procedure for calibrating the so-called Slope point (Calibration Slope). The free chlorine concentration ("S") of the calibration water sample should be 1mg/L FCL (the current default value in the Firmware) as recommended by the manufacturer of the Sensor (CS5530D). The status (successful or not) of the calibration can be confirmed by observing the debug messages in the terminal monitor console. If a water sample with different FCL value "S" is selected, then this value must be declared in the Arduino sketch, compiled, and programmed before the actual calibration, where a water solution with an "S" mg/L free chlorine should be prepared and used for calibrating the slope.

Calibration of the Zero (offset) point (0mg/L FCL)

Similarly, inserting the sensor tip in a chlorine-free water sample and pressing CAL-0 button for more than 3 seconds will initiate a dedicated Calibration Procedure for calibrating the Zero point (Calibration Zero), where the measured value will be referred to as ZERO (0mg/L).

For uncalibrated FCL sensor, **the slope point (for 1mg/L) must be calibrated first**. After it has been successfully calibrated, the zero (offset) point can be calibrated next. A potential error message in the terminal monitor console "**No standard solution info or no standard solution**" during the calibration for the zero (offset) point means that the slope point has not been calibrated successfully. During the initialization process the device checks the FCL sensor for the calibration status of the slope and zero (offset) points. If they have been calibrated the following messages should appear on a connected terminal monitor console:

```
Reading Reg 0x19 (Calibration Done Bit Flags)...
Reg 0x19: FCL Zero point Cal Done
Reg 0x19: FCL Slope point Cal Done
```

All the operational modes are properly indicated by specific LED lighting patterns as shown below.

LEDs Patterns:

GREEN & RED LEDs are OFF	Sleep (low-power) Mode
GREEN LED flashing SLOW	Initializing after PowerOn/Reset/WakeUp or is refreshing the water in the FLOW CELL.
GREEN LED blinks twice, waits for 1 second and repeats	The board waits for CAL-0 or CAL-1 button long presses (3 seconds) to start the calibration process for Zero Point (offset) or Slope Point respectively.
GREEN LED stable ON	Reading Free Chlorine (FCL) values (Measurement Session).
GREEN LED flashes 3 times, relatively fast	The average Free Chlorine (FCL) value is written to the SD card.
GREEN LED flashing FAST	Calibration Mode - Setting new Slope Point of the Digital Free Chlorine Sensor.
RED LED flashing FAST	Calibration Mode - Setting new Zero Point of the Digital Free Chlorine Sensor.
RED LED flashing SLOW	A fault condition is found during initialization
RED LED blinks twice before the device goes to sleep	Battery level is LOW. Replace the Battery.
External RED LED blinks twice, waits for 1 second and repeats	Woke up in Bluetooth Mode, and initializing.
External RED LED stable ON	Device is in Bluetooth Mode and is ready for connection and receiving commands from remote Android device.
External RED LED flashing FAST	Device is in Bluetooth Mode and sending data logs to the remote Android device.

The FCL sensor is designed to work within constant water flow from 10 to 30L/hour (0.16 – 0.5L/min) provided by the water pressure within the piped network and the Flow Cell. However, for the calibration procedure, it is a bit of a challenge to provide a constant water flow with a constant known FCL concentration for a certain time duration needed for the sensor's calibration procedure. This requires dedicated calibration setup has to be built for the purpose as shown in Fig. 8. The setup features a high-pressure water pump (0142YA-24-80 130PSI DC24V 80W) which provides almost the same water pressure as the local water piped system and allows water with altered and known FCL content to circulate in a closed loop through the Flow Cell, thus providing almost ideal conditions for the calibration procedure.

This calibration setup works perfectly fine but is a bit troublesome to implement outdoor in the real field of application of the system. So, a simplified method can be used during the calibration procedure, where the "constant" flow is simulated/created simply by moving fast the sensor probe within the calibration solution in a circular manner. An alternative approach is to use a magnetic or hand stirrer, but this was not tested. The main advantage of the method is the ease of use and the possibility of the system being calibrated in the place where it is installed or in the real field of application.

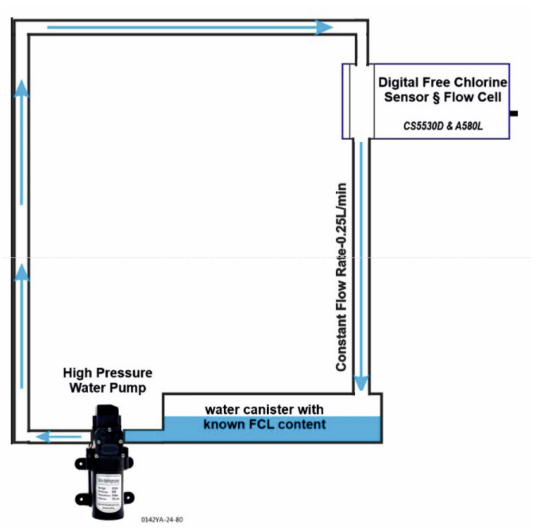

Digital Free Chlorine Sensor § Flow Cell
CS5530D & A580L
Constant Flow Rate-0.25L/min
High Pressure Water Pump
water canister with known FCL content

Figure 8: Calibration Setup with high-pressure water pump

Arduino Firmware Notes:

1. Please, do not remove the SD card from the Water Quality (WQ) device during the data synch operation with a connected Android device.
2. Please, do not remove the SD card during a measuring session.
3. You may **set Water Quality device ID,** by setting a positive integer value to the `WQP_DEVICE_SRV_ID` macro in the **wq_config.h** file from the Arduino sketch. The device has to be re-programmed with the new firmware afterwards to be assigned the new **ID**. The device **ID** is used to differentiate among Water Quality devices advertising themselves over Bluetooth, and to properly identify the source device of the record data when downloading them over Bluetooth with the Android app. Allowed **ID** values are 0 to 999.
4. You may set a valid time to the WQ device's RTC (Real Time Clock) with the following process (requires the WQ device to be programmed twice):
   - In the **wq_config.h** file of the Arduino sketch, set the `SETUP_RTC_TIME` macro to `(true)`, and the `INIT_RTC_TIME_DEF` macro to a long integer value, as returned from a unix timestamp website (eg. https://unixtimestamp.com/). Please keep the "L" suffix after the timestamp number (eg. `1706708374L`). You may also change the `GMT_HOURS_OFFSET_DEF` to set the hours difference from the UTC time tha the unix timestamp website returns. And you may set the `RTC_SECONDS_OFFSET` to compensate for the board programming delay, so that when programmed the board will set the RTC to a time as close as possible to the real one.
   - Compile the Arduino sketch and upload it to the board. After the upload has finished, wait for the board to be initialized until it eventually goes to sleep. The RTC will be set to the specified value (adjusted according to the auxiliary macros mentioned in the previous step)
   - In the **wq_config.h** file of the Arduino sketch, set the `SETUP_RTC_TIME` macro to `(false)`
   - Compile the Arduino sketch and upload it to the board. After the upload has finished, wait for the board to be initialized until it eventually goes to sleep.
5. If the WQ device has gone to sleep due to LOW POWER detected from the battery, it will no longer wake-up at the next alarm time to take measurements, until the battery has been replaced with a (re)charged one.

Bluetooth Mode, SD Card and downloading log files

1. In order for the device to be available for downloading the logs via Bluetooth, it needs to be woken up from the Bluetooth button in blue color situated on the bottom of the enclosure (**Bluetooth mode**).
2. When the device is woken up in Bluetooth mode, the external RED LED (the LED situated next to the Bluetooth button) will start flashing according to the following patterns:

Device is being initialized	Two short blinks followed by 1 second off, repeating
Device is available for connection and remote commands	Stable ON
Device is sending log data to remote Android device	Fast Blinking
Device is unavailable for connection or download process has finished or was canceled	Stable OFF

3. When the device is woken up in Bluetooth mode, and after it reaches the "**available for connection and remote commands**" state, it will wait for 1 minute for a valid request to download the log files to remote Android device. If no such request is made, the WQ device will go to sleep.
4. Please, **only attempt to connect to the WQ device from an Android device running the Android app, when the WQ LED is indicating that the device is available for connection and remote commands (LED is stable ON**). The device will show up in the Android app, before it is ready for proper connection and remote commands, so please wait until the WQ LED indicates availability (LED is stable ON).
5. The WQ LED ***does not*** indicate whether the WQ device is connected to or disconnected from a remote device.
6. During the process of downloading the log files, only a predefined number of recent log files will be downloaded at most. This number is indicated in the **wq_config.h** file of the Arduino sketch as the value for macro **`WQP_HISTORY_OF_FILES_TO_DOWNLOAD`** and it's value, currently 6, corresponds to up to **6 months**' of data records, provided no changes were made to the preset number of logs per log file. The number includes the current active log file (even if it's not full). Also, the log files that will be downloaded need to have consecutive numbering, eg D00003.txt, D00004.txt, D00005.txt, D00006.txt, D00007.txt, D00008.txt. If any are missing from this sequence, then less than the preset number of files will be downloaded, eg if two files, D00006.txt and D00004.txt, are missing, then only 4 files will be downloaded.
7. If the process of downloading the log files to a remote Android device has completed or was canceled, the WQ device will go to sleep. If the process needs to be repeated, the WQ device has to be woken up again.
8. If the process of waiting for a remote Bluetooth connection or the process of downloading the log files over Bluetooth to a remote Android device overlap with the time when the WQ would wake up to make measurements, then the measurement process will commence ***right after*** the downloading process has ended (or has been canceled), and then the WQ will go to sleep.
9. The process of downloading log files can conclude successfully or be canceled by the WQ device or the remote Android device. The following are possible cases for ending the download process:
    - The preset history of log files has been downloaded successfully.
    - The WQ device encountered an irrecoverable error (canceled by WQ)

- The WQ device has very low power and has to go to sleep (canceled by WQ)
- The WQ device did not receive acknowledgement for sending data to the remote Android device for more than 10 seconds (canceled by WQ).
- The user of the remote Android device canceled the download process (canceled by Android app)

10. The preset number of **logs *per log file*** is set to correspond to 30 days of data records (macro **DAYS_PER_LOG_DATA_FILE**). When a log file reaches the number of data records for the preset days number, it is rotated, and a new log file is created (with incremental suffix in its filename) where the future data records will be stored. Please, note that a log file will contain records for the preset number of days, regardless of the date month, and thus it can span across consecutive months, eg. a log file may contain data records from 10 days of January and 20 days of February.
11. It is recommended to try and keep the number of files on the SD card to a low number, by cleaning up any very old "**d#####.txt**" files. Do not delete the **countf.txt** file, as this file keeps track of the log files sequence and the number of records in the currently active log file.
12. If you have installed a cleanly formatted SD card, a proper countf.txt file will be created automatically, when the WQ device starts with the new SD card.
13. Please do not change the values inside the **countf.txt** file manually, but if absolutely required, write the number "2" in the first row, followed by a new line, and nothing else. The maximum allowed value for this number is "32767"**.**

Android App Notes:

1. WQ device is advertised with a name prefix of "**WQProj**", e.g. "WQProj001". The number suffix (here "001") indicates the server **ID** of the WQ device.
2. Please, don't connect to the WQ device too soon, even if its name appears on the list of results from the Bluetooth scan view of the Android app. **Wait** for the external LED on the WQ device to indicate availability for connection.
3. The downloaded file logs are saved in the "**Internal Shared Storage > Download**" folder (and not to external SD card).
4. The downloaded file logs are prefixed by the server **ID** of the WQ device, which they come from. E.g. "1-d00001.txt", "1-d00002.txt" are log files received from the WQ device with server ID "1".
5. The downloaded file logs will not overwrite any log files which were previously saved in the Download folder, if they are identically named. Instead, new files will be created alongside the existing ones with a (x) prefix. E.g. "1-d00001.txt", "1-d00001.txt (1)", "1-d00001.txt (2)"
6. If the download session is cancelled from the Android app, then the WQP device will go to sleep (as soon as it receives the command or detects disconnection), unless the WQP device has a measurement session queued for execution, in which case, the measurement session will commence and then the WQP device will go to sleep.
7. During an ongoing Data Download process, the Android app sends pings to the WQP device every 2.5 seconds, to inform it that it's still connected and listening. If the WQP device does not receive such a ping message within 10 seconds, then it decides that the remote Android device has been disconnected and ends (cancels) the download session. The WQP device waits for a "ping" for a longer period than the period that the Android app sends the "ping" messages, because it is possible for some of these messages to get lost, due to how the serial communication with the BLE module works on the WQP device.