Processor:

- ATmega328P
- Using a low power processor to maximise time between battery replacement/maintenance
- Used to control the sensors, storing and sending data, calculating water levels based on sensor input.

Communication:

- Using LoRaWAN to connect to a WAN to send data from the sensor/device.
- This is received by a server, which will store the data in a database.
- Server will also be responsible for security.

Storage:

- On-device storage (SD card)
- Periodic measurements taken and first stored on the SD storage device as a permanent backup log (until cleared by an operator).
- Will also store a log of any errors that might occur.

Measurement Device:

- Ultrasonic sensor
    - https://www.maxbotix.com/Ultrasonic_Sensors/i2c_distance_sensors-2.htm
- Provides contactless installation to calculate water levels in specified time periods.
- Initial depth value is input by the installation operator with first setup to accurately calculate the water levels based on the total distance from the sensor to the bottom of the river (Total distance = initial distance from sensor to water + initial water level depth).
- Sensor sends pulse, and the processor calculates distance traveled based on the time taken to receive the reflected pulse back from the water top.
- Periodic measurements taken, stored locally on the SD card, then sent to the server to be stored in a secure database.

Case:

- Showerproof casing (no contact with the river itself)
- Used to attach to a secure base above a river (with the ultrasonic sensor faced down)
- Fairly easily disassembled with a hand clip/screw to remove the SD card or replace batteries

Power supply:

- 9V Lithium battery (10 year smoke detector)
- Performs well in colder weather
- Single use - low power usage

Data usage/analysis:

Pseudo code

Calculate water level (Ultrasonic sensor):

```
Const float initialWaterDepthInput;
//initialWaterDepthInput is populated by installation operator with first setup

Const float sensorToRiverBedDistance = sensorToRiverTopDistance() + initialWaterDepthInput;

float sensorToRiverTopDistance()
{
        sendUltrasonicPulse;
        long timePassed = receivePulse; //miliseconds
        return microsecondsToCentimeters(time);
}


float riverLevelCM()
{
        return sensorToRiverBedDistance - sensorToRiverTopDistance();
}

float microsecondsToCentimeters(long microseconds)
{
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the
  // object we take half of the distance travelled.
  return microseconds / 29 / 2;
}
```