

**GOVERNMENT COLLEGE OF TECHNOLOGY,  
COIMBATORE**

Department of Electronics and Communication Engineering

**FPGA-Based Smart Water Safety Alert System  
using VSD Squadron FPGA Mini and MYOSA Sensor Modules**

Project Documentation Report

**Submitted by**

Kamalesh E

Undergraduate Student, B.E. Electronics and Communication Engineering

IEEE Student Member

**Platform Used**

VSD Squadron FPGA Mini Board with MYOSA Sensor Modules

**Project Theme**

Low-Cost FPGA-Based Water Safety Monitoring

Academic Year: 2025–2026

## Abstract

This project presents an FPGA-based smart water safety alert system using the VSD Squadron FPGA Mini board and MYOSA sensor modules. The system is designed as a low-cost digital prototype for classifying water quality conditions into three simple states: Safe, Caution, and Unsafe. The VSD Squadron FPGA Mini board acts as the main processing unit, while MYOSA sensor modules are used as the sensing front-end. The MYOSA MPU or motherboard is not used as the main controller in this implementation.

The project receives sensor warning inputs related to water clarity, temperature, and other water-quality indicators. For the initial prototype, the sensor outputs may be represented as digital warning signals or simulated using jumper inputs. The FPGA processes these inputs using Verilog-based decision logic and generates alert outputs through green, yellow, and red LEDs along with a buzzer.

The creative feature of this project is a bio-inspired risk accumulator. Instead of reacting immediately to a single abnormal input, the FPGA accumulates risk when abnormal conditions continue over time and slowly reduces the risk when the inputs return to normal. This approach reduces false alarms and gives the system a neuromorphic-inspired decision-making behavior.

The project is beginner-friendly, low-cost, and suitable for demonstration using available FPGA hardware and MYOSA sensor modules. The total bill of materials is planned within approximately 150 USD. In future, the prototype can be extended with calibrated pH, turbidity, TDS, conductivity, and temperature sensors for complete field-level water monitoring.

## 1. Introduction

Access to clean and safe drinking water is an important requirement for public health. In many rural and low-resource regions, people may depend on water from wells, tanks, canals, or local storage containers. These sources can become contaminated due to mud, suspended particles, chemical impurities, or temperature-related changes.

Conventional water testing often requires laboratory facilities, trained personnel, and time. A simple local monitoring system can help users quickly understand whether water is safe, doubtful, or unsafe. This project demonstrates an FPGA-based smart water safety alert system using the VSD Squadron FPGA Mini board.

In this implementation, the VSD Squadron FPGA Mini board is used as the main decision-making platform. MYOSA sensor modules are used only as sensor front-end modules. The MYOSA MPU board or motherboard is not used as the main controller. This keeps the project focused on FPGA-based digital processing while still making use of available MYOSA sensors.

The system provides universal alerts using green, yellow, and red LEDs along with a buzzer. This makes the result easy to understand without requiring a mobile phone, internet connection, or technical knowledge.

## 2. Problem Statement

Many communities do not have access to continuous water-quality monitoring. Laboratory-based testing is accurate but slow and may not be available for frequent checking. A low-cost system is needed to provide quick indication of water safety conditions.

The main problem addressed in this project is:

*How can an FPGA board and available MYOSA sensor modules be used to create a low-cost water safety alert system that classifies water condition into Safe, Caution, and Unsafe states?*

## 3. Objectives

The main objectives of this project are:

1. To design an FPGA-based water safety alert system.
2. To use the VSD Squadron FPGA Mini board as the main processing unit.
3. To use MYOSA sensor modules only as sensing front-end modules.
4. To classify water condition into Safe, Caution, and Unsafe states.
5. To provide simple LED and buzzer-based alerts.
6. To implement a bio-inspired risk accumulator in Verilog HDL.
7. To keep the total project cost within approximately 150 USD.
8. To prepare beginner-friendly documentation for project recreation.

## 4. Proposed Solution

The proposed system uses the VSD Squadron FPGA Mini board as the main processing platform. MYOSA sensor modules are used to provide input information related to water-quality conditions. The system may use sensor modules such as light/proximity sensor, temperature/environmental sensor, and optional analog sensor modules.

For the initial prototype, four warning input signals are considered:

1. Abnormal pH condition
2. High turbidity condition
3. High TDS or conductivity condition
4. Abnormal temperature condition

Each input is represented as a digital warning signal. A logic HIGH represents an abnormal condition, and a logic LOW represents a normal condition. The FPGA counts the number of abnormal inputs and updates a risk accumulator.

Based on the risk value, the system gives one of three outputs:

- Green LED: Safe condition
- Yellow LED: Caution condition
- Red LED and buzzer: Unsafe condition

## 5. System Architecture

The system architecture is shown using a text-based block diagram below.

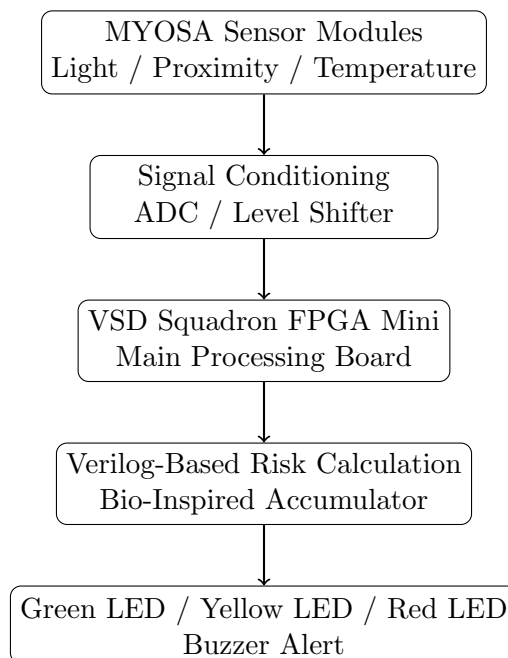


Figure 1: System architecture of the FPGA-based water safety alert system

## 6. Working Principle

The working principle is based on sensor-based risk detection. The MYOSA sensor modules provide information related to the water sample or environmental condition. For example, a light or proximity sensor can be used to estimate turbidity by measuring the amount of light passing through a water sample. Clear water allows more light to pass, while muddy or turbid water blocks or scatters light.

The sensor readings are converted into warning signals. These warning signals are given to the FPGA. The FPGA then performs the following operations:

1. Receives warning inputs from sensor modules or test jumper inputs.

2. Counts how many water-quality parameters are abnormal.
3. Updates a bio-inspired risk accumulator.
4. Classifies the condition into Safe, Caution, or Unsafe.
5. Activates the respective LED and buzzer output.

The system is designed so that one short abnormal reading does not immediately trigger a danger alarm. Instead, repeated abnormal readings increase the internal risk value. This improves stability and reduces false alarms.

## 7. Creativity and Innovation

The creative feature of this project is the bio-inspired risk accumulator implemented in FPGA hardware. A simple threshold-based system may switch immediately from Safe to Unsafe due to noise or a temporary input change. This project avoids that behavior.

The FPGA risk accumulator behaves like a simple neuron. It integrates abnormal inputs over time and slowly leaks the risk value when normal conditions return. This idea is inspired by leaky integrate-and-fire neuron behavior. If abnormal water conditions continue, the risk value rises and the system moves from Safe to Caution and then to Unsafe. If the water condition becomes normal again, the risk value slowly decreases.

This makes the system more reliable and gives a fresh FPGA-based approach to a simple water safety alert problem.

## 8. Complete Bill of Materials

### 8.1. Hardware Requirements

The project uses the VSD Squadron FPGA Mini board as the main processing platform. MYOSA sensor modules are used only as the sensing front-end. The MYOSA motherboard or MPU board is not used as the main controller in this implementation. The total planned hardware cost is kept within approximately USD 150.

Table 1: Hardware Bill of Materials

S.No.	Component	Qty.	Approx. Cost (USD)	Purpose
1	VSD Squadron FPGA Mini Board	1	37	Main FPGA processing board for implementing Verilog-based decision logic

S.No.	Component	Qty.	Approx. Cost (USD)	Purpose
2	MYOSA Light / Proximity Sensor Module	1	15	Used for optical turbidity sensing by measuring light passing through water
3	MYOSA Temperature / Environmental Sensor Module	1	15	Used to represent temperature or environmental abnormality input
4	MYOSA OLED / Display Module	1	15	Optional display module for showing water status or sensor value
5	MYOSA Buzzer / Actuator Module	1	10	Provides audible alert during unsafe condition
6	ADS1115 16-bit ADC Module	1	5	Converts analog sensor outputs into digital values for FPGA interface
7	Logic Level Shifter / Protection Resistor Network	1	5	Protects FPGA GPIO pins and matches signal levels if required
8	Green LED	1	1	Safe water indication
9	Yellow LED	1	1	Caution indication
10	Red LED	1	1	Unsafe water indication
11	220 Ohm Resistors	3	1	Current limiting for LEDs
12	Breadboard and Jumper Wires	1 set	5	Circuit prototyping and connections
13	Transparent Water Container / Test Tube	1	3	Holds the water sample during optical turbidity test
14	Phone Torch / White LED Light Source	1	2	Provides light for optical turbidity sensing
15	USB Cable and Miscellaneous Consumables	1 set	5	Programming, power, and mechanical support
<b>Estimated Total Cost</b>			<b>121</b>	Within USD 150 budget

**Note:** The cost values are approximate and may vary depending on supplier, shipping, and availability. Since the VSD Squadron FPGA Mini board and MYOSA sensor modules are already available, the actual additional expenditure for the prototype can be much lower than the estimated total.

## 8.2. Software Requirements

Table 2: Software and Tools Used

S.No.	Software/Tool	Purpose
1	Verilog HDL	Hardware description language used to design FPGA logic
2	Yosys	Synthesizes Verilog code for the iCE40 FPGA family
3	nextpnr-ice40	Performs place and route for the iCE40 FPGA
4	Project IceStorm	Generates and programs the FPGA bitstream
5	Arduino IDE / MYOSA Example Tools	Used only for checking MYOSA sensor module behavior before FPGA interfacing
6	diagrams.net / Fritzing	Used for drawing block diagrams and circuit diagrams
7	Camera / Mobile Phone	Used for capturing photos and demonstration video

## 9. Circuit Diagram and Connections

The project uses MYOSA sensor modules as the sensing front-end and the VSD Squadron FPGA Mini board as the main processing board. For the initial prototype, sensor warning inputs can be tested using jumper wires or switch inputs before connecting real sensor modules through ADC or digital output circuits.

### 9.1. Text-Based Circuit Schematic

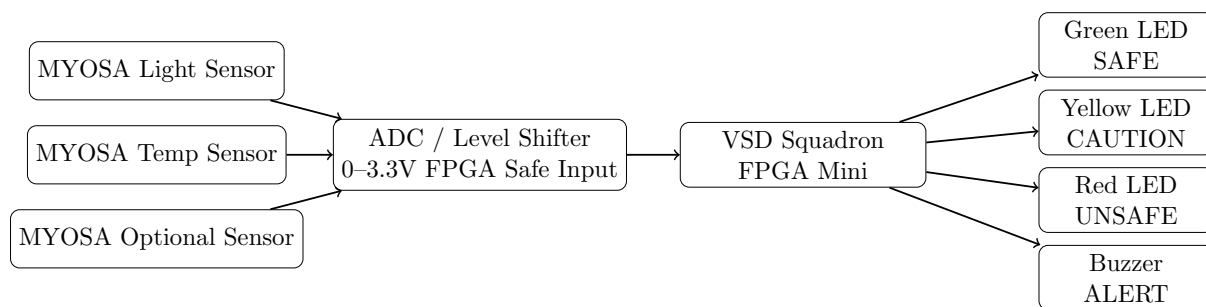


Figure 2: Text-based circuit schematic of the proposed system

## 9.2. Connection Table

Table 3: FPGA Connection Table

Signal Name	Connection	Purpose
ph_bad	FPGA GPIO input	Simulates abnormal pH condition
turbidity_bad	FPGA GPIO input	Simulates high turbidity condition
tds_bad	FPGA GPIO input	Simulates high TDS or conductivity condition
temp_bad	FPGA GPIO input	Simulates abnormal temperature condition
green_led	FPGA GPIO output	Safe water indication
yellow_led	FPGA GPIO output	Caution indication
red_led	FPGA GPIO output	Unsafe water indication
buzzer	FPGA GPIO output	Audible danger alert

## 9.3. Initial Prototype Wiring Description

For the first working prototype, the four sensor warning inputs can be simulated using jumper wires or switches. A logic HIGH indicates that the particular water-quality parameter is abnormal. A logic LOW indicates normal condition.

The LED outputs are connected through 220 Ohm current-limiting resistors. The buzzer output should be connected to an active buzzer module. If a high-current buzzer is used, a transistor driver must be added between the FPGA pin and the buzzer.

## 10. Step-by-Step Implementation Procedure

### 10.1. Step 1: Prepare the FPGA Toolchain

Install the FPGA toolchain required for the VSD Squadron FPGA Mini board. The tools used are Yosys, nextpnr-ice40, and Project IceStorm.

### 10.2. Step 2: Create Project Folder

Create a folder named `fpga_water_guardian`. Inside this folder, create the following files:

- `top.v`
- `water_guardian.pcf`
- `Makefile`
- `README.md`

### 10.3. Step 3: Write the Verilog Code

Create the main Verilog file named `top.v`. The Verilog code should receive four warning inputs and control three LEDs and one buzzer.

#### **10.4. Step 4: Create Pin Constraint File**

Create a physical constraint file for mapping Verilog signal names to FPGA GPIO pins. The exact pin numbers must be selected based on the VSD Squadron FPGA Mini board pinout.

#### **10.5. Step 5: Build the FPGA Bitstream**

Run synthesis, place-and-route, and bitstream generation using the FPGA toolchain.

#### **10.6. Step 6: Program the FPGA**

Connect the FPGA board to the laptop through USB and program the generated bitstream into the board.

#### **10.7. Step 7: Test Using Jumper Inputs**

Before connecting real sensors, test the system using jumper wires or switches. Apply logic HIGH and LOW values to the four input pins and observe LED and buzzer behavior.

#### **10.8. Step 8: Connect MYOSA Sensor Modules**

After verifying the FPGA logic, connect MYOSA sensor module outputs to the FPGA through suitable signal conditioning, ADC, or level shifting circuit. Ensure that FPGA input voltage limits are not exceeded.

#### **10.9. Step 9: Capture Project Evidence**

Capture the following:

1. Photo of FPGA board setup.
2. Photo of MYOSA sensor module placement.
3. Screenshot of Verilog code.
4. Screenshot of successful synthesis/build.
5. Photo or video showing Safe, Caution, and Unsafe output states.

### **11. Code and Contribution**

The project is implemented using Verilog HDL. The FPGA receives four digital input signals representing water-quality warning conditions. The code counts the number of abnormal inputs, updates a bio-inspired risk accumulator, and controls LED and buzzer outputs based on the risk level.

Listing 1: FPGA-Based Smart Water Safety Alert System Verilog Code

```

1 module top (
2     input wire ph_bad,
3     input wire turbidity_bad,
4     input wire tds_bad,
5     input wire temp_bad,
6
7     output reg green_led,
8     output reg yellow_led,
9     output reg red_led,
10    output reg buzzer
11 );
12
13 // Internal oscillator for iCE40UP5K FPGA
14 wire clk;
15
16 SB_HFOSC #(
17     .CLKHF_DIV("0b10")
18 ) internal_oscillator (
19     .CLKHFPU(1'b1),
20     .CLKHFEN(1'b1),
21     .CLKHF(clk)
22 );
23
24 // Slow tick generator
25 reg [23:0] slow_counter = 24'd0;
26 reg slow_tick = 1'b0;
27
28 always @(posedge clk) begin
29     slow_counter <= slow_counter + 1'b1;
30
31     if (slow_counter == 24'd6_000_000) begin
32         slow_counter <= 24'd0;
33         slow_tick <= 1'b1;
34     end else begin
35         slow_tick <= 1'b0;
36     end
37 end
38
39 // Count abnormal sensor conditions
40 wire [2:0] bad_count;
41 assign bad_count = ph_bad + turbidity_bad + tds_bad + temp_bad;
42
43 // Bio-inspired risk accumulator
44 reg [3:0] risk_accumulator = 4'd0;
45
46 always @(posedge clk) begin
47     if (slow_tick) begin
48         if (bad_count == 3'd0) begin
49             if (risk_accumulator > 4'd0)

```

```

50         risk_accumulator <= risk_accumulator - 1'b1;
51     else
52         risk_accumulator <= 4'd0;
53     end
54     else if (bad_count == 3'd1) begin
55         if (risk_accumulator < 4'd15)
56             risk_accumulator <= risk_accumulator + 1'b1;
57         else
58             risk_accumulator <= 4'd15;
59     end
60     else if (bad_count == 3'd2) begin
61         if (risk_accumulator < 4'd14)
62             risk_accumulator <= risk_accumulator + 2'd2;
63         else
64             risk_accumulator <= 4'd15;
65     end
66     else begin
67         if (risk_accumulator < 4'd12)
68             risk_accumulator <= risk_accumulator + 3'd3;
69         else
70             risk_accumulator <= 4'd15;
71     end
72 end
73 end
74
75 // State decision
76 reg [1:0] state;
77
78 localparam SAFE      = 2'b00;
79 localparam CAUTION  = 2'b01;
80 localparam DANGER    = 2'b10;
81
82 always @(*) begin
83     if (risk_accumulator <= 4'd3)
84         state = SAFE;
85     else if (risk_accumulator <= 4'd8)
86         state = CAUTION;
87     else
88         state = DANGER;
89 end
90
91 // Buzzer pattern
92 reg [22:0] beep_counter = 23'd0;
93
94 always @(posedge clk) begin
95     beep_counter <= beep_counter + 1'b1;
96 end
97
98 wire slow_beep;
99 wire fast_beep;

```

```

100
101 assign slow_beep = beep_counter[22];
102 assign fast_beep = beep_counter[20];
103
104 // Output control
105 always @(*) begin
106     green_led = 1'b0;
107     yellow_led = 1'b0;
108     red_led = 1'b0;
109     buzzer = 1'b0;
110
111     case (state)
112         SAFE: begin
113             green_led = 1'b1;
114             yellow_led = 1'b0;
115             red_led = 1'b0;
116             buzzer = 1'b0;
117         end
118
119         CAUTION: begin
120             green_led = 1'b0;
121             yellow_led = 1'b1;
122             red_led = 1'b0;
123             buzzer = slow_beep;
124         end
125
126         DANGER: begin
127             green_led = 1'b0;
128             yellow_led = 1'b0;
129             red_led = 1'b1;
130             buzzer = fast_beep;
131         end
132
133         default: begin
134             green_led = 1'b0;
135             yellow_led = 1'b0;
136             red_led = 1'b0;
137             buzzer = 1'b0;
138         end
139     endcase
140 end
141
142 endmodule

```

## 12. Example Constraint File

The following is an example constraint file. The exact GPIO pin numbers must be verified from the VSD Squadron FPGA Mini board pinout before final hardware implementation.

Listing 2: Example PCF Constraint File

```
1 # Input pins
2 set_io ph_bad          9
3 set_io turbidity_bad  10
4 set_io tds_bad        11
5 set_io temp_bad       12
6
7 # Output pins
8 set_io green_led      13
9 set_io yellow_led     18
10 set_io red_led       19
11 set_io buzzer        20
```

### 13. Makefile

The following Makefile can be used for building and programming the FPGA bitstream.

Listing 3: Makefile for FPGA Build

```
1 PROJECT = water_guardian
2 TOP = top
3 DEVICE = up5k
4 PACKAGE = sg48
5
6 all: $(PROJECT).bin
7
8 $(PROJECT).json: top.v
9     yosys -p "synth_ice40 -top $(TOP) -json $(PROJECT).json" top.v
10
11 $(PROJECT).asc: $(PROJECT).json water_guardian.pcf
12     nextpnr-ice40 --$(DEVICE) --package $(PACKAGE) --json $(PROJECT).json
13     --pcf water_guardian.pcf --asc $(PROJECT).asc
14
15 $(PROJECT).bin: $(PROJECT).asc
16     icепack $(PROJECT).asc $(PROJECT).bin
17
18 prog: $(PROJECT).bin
19     iceprog $(PROJECT).bin
20
21 clean:
22     rm -f *.json *.asc *.bin
```

## 14. Testing and Results

Table 4: Testing Table

Test Case	Input Condition	Expected Output	Observed Out
1	All inputs LOW	Green LED ON, buzzer OFF	To be filled after t
2	One input HIGH	Yellow LED ON, slow buzzer beep	To be filled after t
3	Three or four inputs HIGH	Red LED ON, fast buzzer beep	To be filled after t
4	Inputs changed from HIGH to LOW	Risk slowly decreases to Safe	To be filled after t

### 14.1. Result Images and Demonstration Evidence

The following evidence will be added after completing the hardware implementation:

1. Photograph of the VSD Squadron FPGA Mini board setup.
2. Photograph of MYOSA sensor module arrangement.
3. Screenshot of Verilog HDL code.
4. Screenshot of FPGA synthesis and programming.
5. Photo or video frame showing Safe condition.
6. Photo or video frame showing Caution condition.
7. Photo or video frame showing Unsafe condition.

**Note:** Since the project is currently in the planning and implementation stage, final output images will be inserted after testing the prototype.

## 15. Beginner Recreation Guide

A beginner can recreate this project by following these steps:

1. Install Yosys, nextpnr-ice40, and Project IceStorm.
2. Connect the VSD Squadron FPGA Mini board to the laptop.
3. Write the Verilog file `top.v`.
4. Create the FPGA constraint file.
5. Build the FPGA bitstream.
6. Program the FPGA board.
7. First test the design using jumper inputs.
8. Connect LEDs and buzzer to FPGA output pins.

9. Verify Safe, Caution, and Unsafe outputs.
10. Connect MYOSA sensor modules through ADC or level shifting circuit.
11. Repeat testing using actual sensor module inputs.
12. Capture photos, screenshots, and video demonstration.

## **16. Video Demonstration Plan**

The video demonstration should include the following:

1. Introduction of project title.
2. Display of VSD Squadron FPGA Mini board.
3. Display of MYOSA sensor modules used.
4. Explanation of sensor input signals.
5. Demonstration of Safe condition.
6. Demonstration of Caution condition.
7. Demonstration of Unsafe condition with buzzer.
8. Explanation of the bio-inspired risk accumulator.

## **17. Limitations**

This project is an educational prototype and has some limitations:

1. Initial testing may use simulated sensor warning inputs.
2. Real analog sensors require ADC or signal conditioning before FPGA input.
3. The system does not directly perform laboratory-level chemical analysis.
4. The MYOSA modules may require voltage-level compatibility checking.
5. Calibration is required before using real water-quality sensors.

## **18. Future Scope**

The project can be improved in the following ways:

1. Add real pH, turbidity, TDS, and temperature sensors.
2. Use an ADC module to convert analog sensor values into digital values for the FPGA.

3. Interface the MYOSA kit as a sensor front-end for the FPGA.
4. Add OLED or LCD display for showing the risk score.
5. Add data logging using external memory.
6. Add solar power for off-grid operation.
7. Add LoRa communication for remote village water monitoring.
8. Extend the logic into a complete neuromorphic water-quality monitoring system.

## 19. Rubric Mapping

Table 5: Project Rubric Mapping

<b>Rubric Category</b>	<b>Marks</b>	<b>How This Project Satisfies It</b>
Project Documentation	20	The report includes the project story, architecture, step-by-step instructions, testing plan, and demonstration evidence plan.
Complete BOM	20	The hardware and software items are listed clearly with approximate cost and total estimated budget within USD 150.
Schematics	10	Text-based block diagram, circuit schematic, physical arrangement, and FPGA connection table are provided. Real photographs or Fritzing diagrams can be added after implementation.
Code and Contribution	20	Complete Verilog HDL code with comments is included. The code implements input processing, risk calculation, and alert output.
Creativity	30	The project adds a bio-inspired risk accumulator instead of using only simple threshold detection, making the design more stable and creative.

## 20. Conclusion

This project demonstrates an FPGA-based smart water safety alert system using the VSD Squadron FPGA Mini board and MYOSA sensor modules. The FPGA is used as the main decision-making hardware, while MYOSA sensor modules are used as the sensing front-end. The MYOSA MPU or motherboard is not used as the main controller.

The system receives water-quality warning inputs, processes them using Verilog-based logic, and generates Safe, Caution, and Unsafe outputs using LEDs and a buzzer. The major creative feature is the bio-inspired risk accumulator, which improves stability by accumulating repeated abnormal conditions instead of reacting to a single sudden input.

The current version is a beginner-friendly FPGA prototype that can be demonstrated using jumper inputs and later extended with real MYOSA sensor module outputs through ADC or level shifting circuits. The design stays within the planned USD 150 budget and provides a clear path toward a complete field-level water monitoring system.

## References

1. VSD Squadron FPGA Mini Board Documentation.
2. MYOSA Sensor Module Documentation.
3. Yosys Open Synthesis Suite Documentation.
4. nextpnr-ice40 Documentation.
5. Project IceStorm Documentation.
6. Basic principles of water quality monitoring and optical turbidity sensing.