# User Interface Block Validation

Block Champion: Savannah Tanner

Date: Jan-24-2024

## 1. Description

The user interface serves as a bridge between the user and the technology embedded in the fish tank monitor. It contains a number of pages on a rectangular e-ink display which is controlled with a scrolling system. These pages include the home, settings, parameter history, inhabitants list, and notification pages. Each of these pages serves a different but interconnected purpose.

The home page provides essential information to the user including the date, time, and temperature of the tank. Additionally, it offers two options for user interaction, enabling the user to indicate whether the fish tank has been cleaned or the fish have been fed. It also includes a notification symbol to indicate that there are new notifications to be checked by the user. This page serves as a quick and crucial start for the user to access information on their tank.

The settings page includes the option for the user to alter the date, time, parameter ranges, and integrated routines. The monitor's routines allow the user to receive notifications to clean their tank or feed their fish, and due to the wide variety of requirements across species, these conditions must be updateable. The automatic testing routine also must be controlled depending on the frequency of testing desired by the user. As the parameter testing is done automatically, this routine controls the machine's testing unit. The settings page also offers the user the opportunity to export their testing data onto an external device.

The parameter history page is designed for users to comprehensively review their test results. The data is presented in both graph and list forms for each parameter, providing a long-term overview of the tank's conditions. This feature enhances the user's ability to track changes in their fish tank over time, contributing to proper fish care and informed decision-making regarding the tank environment.

The inhabitants list page offers a centralized location for users to keep track of the various creatures living in their fish tank. This list contributes to the user's awareness of the tank's population, facilitating better care and management.

The notifications page serves as a critical communication hub to display important notices for the user. It displays alerts for potentially dangerous test results, errors within the system, and information on when the fish tank monitor is running low on test strips. By providing this information in a timely manner, users can stay informed about potential issues within their tank.

The user interface is designed for a rectangular, e-ink display which only updates upon request. Because of this, the layout is simple and easy to understand to increase accessibility. Because of the e-ink screen, the display does not use any color and instead displays black text and images on a white background.

The fish tank monitor's user interface is thoughtfully designed to improve and enhance the user experience and facilitate effective management of the fish tank environment. Its user-friendly pages, customization options, data representation, and notification system combine to ensure the well-being of the aquatic ecosystem of the fish tank.

## 2. Design

In the design portion for the user interface block, I have included a black box diagram, a flowchart displaying the flow of the pages, and a wiring description. The black box diagram details the connections between the user and the user interface. This connection entails selections from the user, moving between pages, and commands that the user might use within the system. The next connection is between the user interface and the user output. This represents the screen output to the user allowing them to operate the system. The final connection is between the user interface and the screen data. This connection represents the user interface's control over the output of the screen. The flowchart represents how the user can move about the interface. It shows all five pages, buttons on each page, sub-pages such as the graph and list views of the test results, as well as certain constants displayed on each screen. The wiring description shows how the color sensor and temperature sensor connect to the screen, though this does not yet account for the printed circuit board designed by our team.

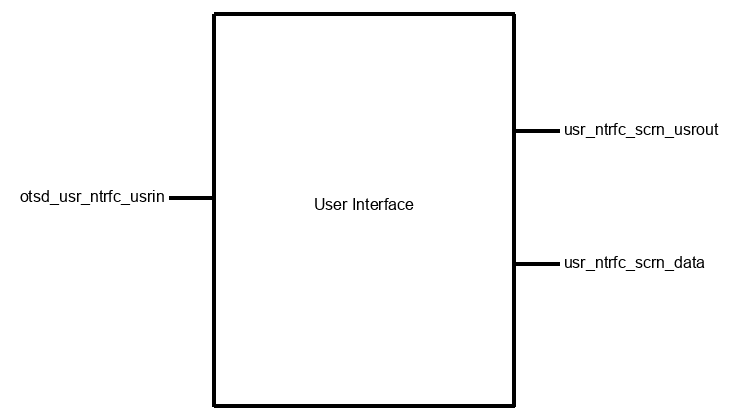
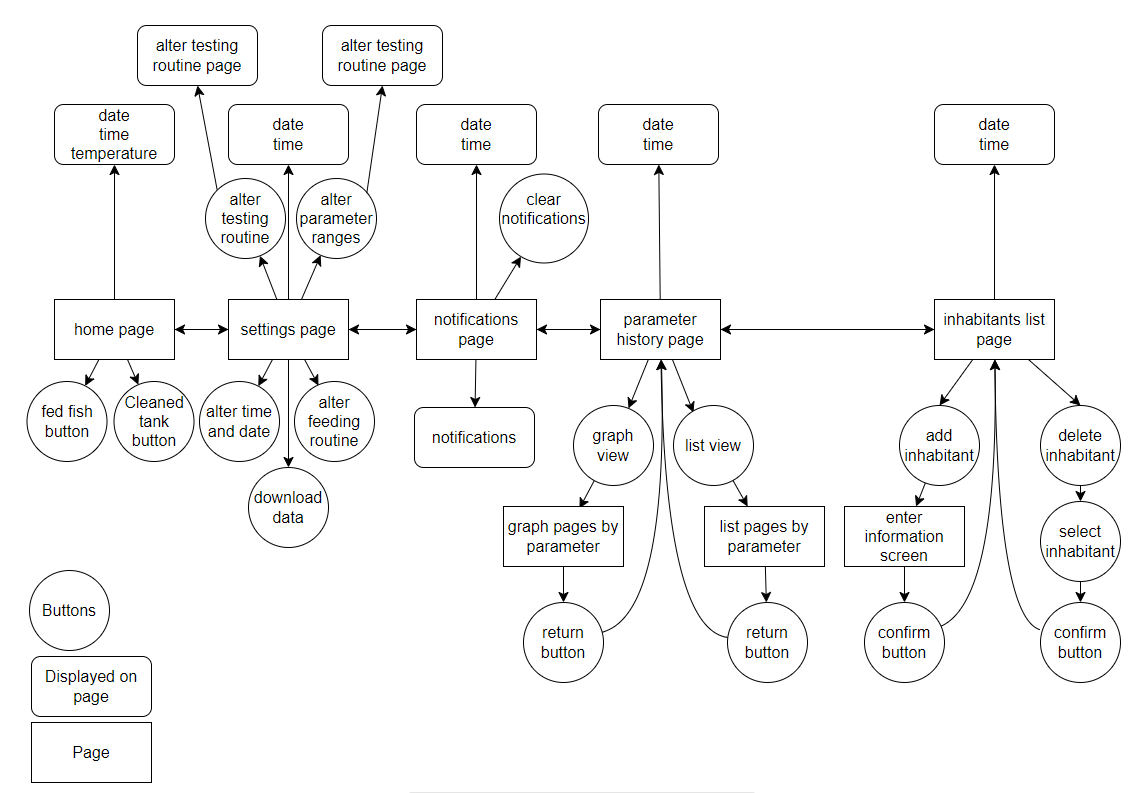


Figure 1: User Interface Block Diagram; Inputs: otsd\_usr\_ntrfc\_usrin; Outputs: usr\_ntrfc\_scrn\_usrout, usr\_ntrfc\_scrn\_data

Figure 2: User Interface Flow Chart

Figure 2: User Interface Flow Chart

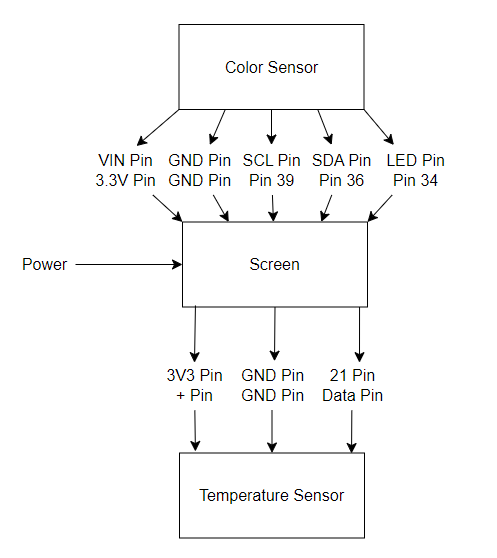


Figure 3: Wiring Diagram between Screen, Color Sensor, and Temperature Sensor

## 3. General Validation

The design of the user interface block for the fish tank monitor has been chosen to align with the specific needs and objectives of the system. Several factors were taken into consideration including cost-effectiveness, part availability, user friendliness, performance, interaction between blocks, size constraints, and the needs of potential users.

Firstly, the choice of a rectangular e-ink display was made based on a combination of its cost-effectiveness and energy efficiency. E-ink displays only update upon request, reducing power consumption and ensuring a longer lifespan for the monitor. As an alternate solution, a different display technology, such as an LCD screen, could have been considered, however the choice of e-ink aligns with the project's focus on energy efficiency and simplicity, catering to the needs of both users and the overall system.

The simplicity and easy understanding of the layout were prioritized to increase accessibility for all users. The absence of color in the design, while in part due to limitations of the e-ink screen, contributes to a clean and straightforward visual representation. This design decision not only enhances user experience but also reduces potential distractions, keeping the device's sole focus on aquatic care.

Due to the interconnected nature of the system, the user interface is designed to provide a seamless experience across pages. The cohesive design ensures that users can easily navigate between functionalities, contributing to user-friendly interactions.

The inclusion of essential features on the home page, such as date, time, and temperature display, along with options for indicating tank cleaning and fish feeding, addresses the immediate and frequent needs of users. This quick access to crucial information aligns with the objective of providing users with a comprehensive overview of their fish tank's status.

The settings page offers customization options, acknowledging the diverse requirements of different aquatic environments. Users can alter the date, time, parameter ranges, and integrated routines, allowing for adaptability to specific conditions. The export function for testing data to an external device adds an extra layer of flexibility for users who may want to analyze data beyond the monitor's interface.

The parameter history and inhabitants list pages contribute to long-term monitoring and management. Users can track changes in tank conditions over time and maintain awareness of the tank's population. This aligns with the goal of promoting proper fish care and informed decision-making regarding the tank environment.

The notifications page serves as a critical communication hub, providing timely alerts for potential issues within the tank. This feature enhances user awareness and allows for prompt responses to critical situations to lower the risk of harm to species within the tank.

In conclusion, the thoughtful design of the user interface block takes into account various considerations to meet the specific requirements of the fish tank monitor system. By balancing factors like cost, accessibility, customization, and long-term monitoring, the user interface contributes significantly to the overall success of the fish tank monitoring system.

## 4. Interface Validation

User Interface Block Interface Validation

| **Interface** | **Interface Property** | **Why is this interface this value?** | **Why do you know that your design details for this block**  **above meet or exceed each property?** |
| --- | --- | --- | --- |
| otsd\_usr\_ntrfc\_usrin : Input | Other: interrupt based | The user interface will be interrupt based so that the system is not constantly checking for a variety of conditions. This will be simpler, more efficient, and faster. | The interrupt based design ensures that the system is responsive, avoiding the constant checking and conserving the system's resources. This approach exceeds property requirements by providing efficiency and speed to user interaction. |
| otsd\_usr\_ntrfc\_usrin : Input | Type: multi-directional | The user interface will be multi-directional to enhance the user's interaction. This will allow the user to navigate the various functionalities of the system by pressing buttons in various directions. This will help the user interface to be more user friendly and usable. | The multi-directional input enhances the user's interaction. It offers flexibility so that the user can navigate the system as they choose. This design meets the property by creating a user-friendly interface and accommodating diverse user preferences. |
| otsd\_usr\_ntrfc\_usrin : Input | Type: button pushes | The user interface will be navigated through the use of buttons that are connected to the system. The presses of each button will allow the user to move between pages to monitor their fish tank effectively. | Button based navigation ensures an intuitive and straightforward user experience. This design meets the property requirements by offering a practical and easily understood input mechanism. |
| usr\_ntrfc\_scrn\_usrout : Output | Other: paper-like | The screen we have selected is a "paper-like" screen. Specifically, we have selected an Electronic Ink display which mimics the appearance of ink on paper and only uses power when the content of the screen changes. This makes the screen more energy efficient since it is not wasting energy on updating the screen constantly. | The choice of a paper-like display aligns with the goal of our project being energy efficient, updated only when necessary. This detail meets our property requirements for energy efficiency. |
| usr\_ntrfc\_scrn\_usrout : Output | Type: Screen output consists of graphs, lists, numbers, text, and illustrations (all of which are depicting some information about the aquarium's water parameters) | The screen has a variety of necessary outputs. These all allow the user to properly monitor their aquarium. The user will be capable of tracking their tank parameters over time with graphs that are displayed by the system as well as in a list format so that they can see more exact information by date. The system must display the numerical values for each parameter as well as text defining what the user is seeing. The illustrations that will be displayed by the system are meant to be used as labels for various pages in the system's user interface. | The diverse screen outputs cater to a variety of user preferences and needs, meeting the property requirements for a comprehensive and informative display. The design ensures that users can track their tank parameters measured by the system, meeting the usability standards for beginners and experienced hobbyists. |
| usr\_ntrfc\_scrn\_usrout : Output | Usability: Output must be useful and usable, must be understandable by 9/10 users | The system must be complex enough to keep track of complicated information necessary for the health of aquariums, but it must be displayed in a simple enough manner for all fish tank hobbyist levels. Beginner hobbyists should be as capable of using the system as those who have been involved in the hobby for a long time. | The system is designed to balance complexity and simplicity, meeting the usability requirements for both experienced and beginner hobbyists. The graphical and list representations enhance data comprehension and clear-labeling ensures a user-friendly experience, meeting the property requirements. |
| usr\_ntrfc\_scrn\_data : Output | Datarate: 400 kbit/s | The selected datarate is meant to suit the slower refresh rate of the e-ink display and suit the processing speed of our selected chip, the ESP32. | The datarate was selected to match the characteristics of our display and the processing speed of the ESP32. The design detail ensures optimal communication, prevents data transfer bottlenecks, and aligns with the system's hardware capabilities. |
| usr\_ntrfc\_scrn\_data : Output | Messages: user interface tells screen what to display depending on user input through buttons. Displays screens with graphs, illustrations, lists, text, and numbers. | The messages sent from the user interface to the screen in the form of data represent the need for the screen to update from one page to another. Depending on the user's selection with the use of buttons, the screen will need to display graphs and lists of parameter data over time, illustrations of different pages, and text representing parameter values. | The structured messages ensure effective communication, instructing the screen on updates based on user inputs. This design meets property requirements for dynamic and context-aware screen updates. |
| usr\_ntrfc\_scrn\_data : Output | Protocol: I2C | The I2C protocol was selected to allow for bi-directional communication and simplicity as well as library support. It was also selected because we are using a variety of devices and I2C will allow for each device to be used with the same communication bus. Lastly, I2C is a standard used widely in electronics. | The selection of I2C protocol supports bi-directional communication, simplicity, and library support. It ensures compatibility with various devices on the same communication bus, meeting the property requirements for reliable and standardized communication within the system. |

## 

## 5. Verification Plan

1. Testing Interrupt-Based Input (otsd\_usr\_ntrfc\_usrin)
   1. Simulate user interactions triggering interrupts (button presses)
   2. Verify that the interrupt-based design responds promptly and efficiently
   3. Ensure that the system does not constantly check for conditions, validating the interrupt-based approach
2. Testing Multi-Directional Input (otsd\_usr\_ntrfc\_usrin)
   1. Use buttons in various directions to navigate through the system
   2. Confirm that the multi-directional input enhances the user's interaction
   3. Ensure flexibility in navigating different functionalities
3. Testing Button-Based Navigation (otsd\_usr\_ntrfc\_usrin)
   1. Simulate button presses to navigate between pages and functionalities
   2. Validate that the button-based navigation is intuitive and easily understood
   3. Confirm practicality and effectiveness of button-based input
4. Testing "Paper-Like" Screen Output (usr\_ntrfc\_scrn\_usrout)
   1. Display various types of content on the e-ink screen (graphs, lists, numbers, text, and illustrations)
   2. Verify that the screen mimics the appearance of ink on paper
   3. Confirm that the screen only uses power when the content changes
5. Testing Screen Output Usability (usr\_ntrfc\_scrn\_usrout)
   1. Observe and collect feedback from users with varying levels of experience
   2. Ensure that the output is useful and understandable by at least 9 out of 10 users
   3. Validate that the graphical and list representations meet usability standards
6. Testing Datarate Compatibility (usr\_ntrfc\_scrn\_data)
   1. Send data at the specified rate (400 kbit/s) to match the e-ink display's refresh rate
   2. Verify that the datarate suits the processing speed of the ESP32
   3. Ensure optimal communication without data transfer bottlenecks
7. Testing User Interface Messages (usr\_ntrfc\_scrn\_data)
   1. Send messages from the user interface to the screen to update pages
   2. Confirm that the screen responds correctly to user inputs with the appropriate updates
   3. Validate that messages instruct the screen to display graphs, lists, illustrations, text, and numbers
8. Testing I2C Protocol (usr\_ntrfc\_scrn\_data)
   1. Implement bi-directional communication using the I2C protocol
   2. Verify that the communication is reliable and consistent
   3. Ensure that I2C protocol supports the standardized communication required for the system
9. Comprehensive Integration Testing
   1. Integrate the user interface block into the overall fish tank monitor system
   2. Test the block's interaction with other blocks and components
   3. Validate that the user interface seamlessly contributes to the effective management of the fish tank environment
10. User Feedback and Iterative Testing
    1. Gather feedback from users during real-world usage
    2. Iterate on any identified issues or areas for improvement
    3. Conduct additional testing if changes are made to the user interface block
11. Validation against System RequirementS
    1. Verify that the user interface block, when integrated into the system, meets all specified system requirements
    2. Ensure that the block contributes significantly to the overall success of the fish tank monitoring system

## 6. References and File Links

### 6.1 References

### 6.2 File Links

[User Interface Paper Prototype](https://docs.google.com/document/d/1qcesZ8mIA9sEpSjqNkljarwgtFWUU8wRginwR6MpLrY/edit?usp=sharing)

[Interactive UI (slides)](https://docs.google.com/presentation/d/1fcd7UCvtgRXrfySkEQBxXgiFbmlICC6Yp_s5Ec1n-Jc/edit?usp=sharing)

## 7. Revision Table

| **Date** | **Action** |
| --- | --- |
| 1/24/2024 | Document creation; sections 1-7 filled in; section completed; |

# Sensor Control Block Validation

Block Champion: Savannah Tanner

Date: Jan-24-2024

## 1. Description

The sensor control block is responsible for the control of the temperature sensor and the color sensor. It is a pivotal element within our fish tank monitoring system, orchestrating the primary functionality of the device by utilizing the sensors and gathering data for the user. This module is tasked with the control, interpretation, and management of all sensor operations ensuring accurate data collection and communication with the primary board of the device for comprehensive parameter monitoring. The block serves as the system's primary nerve center for sensor-related operations, offering a systematic approach to the color and temperature data collection. Its primary objectives are to initialize, control, and interpret data which provide essential information to the user through the screen and user interface.

The block first initializes the color and temperature sensors. Within a loop, the system initializes the color sensor only when required for testing to save power. The temperature sensor runs constantly as the data that it collects is more frequently needed for fish tank hobbyists. The data from the temperature sensor are sent to the main board to be converted into readable temperatures and displayed for the user. This value is constantly displayed on the home screen of the user interface, so each time that the loop repeats the temperature sensor re-reads the temperature of the water.

When the color sensor is initialized, it reads the values across five squares on a test strip. This data is sent to the main board for interpretation where the data is converted into color values in hex code. Once these color values are obtained, they can be mapped to the proper parameter values based on the color key provided with the test strips. This step ensures that any nuances in aquatic conditions are properly reflected.

The sensor control block is controlled via Arduino code which is connected to the user interface block. Both blocks are interconnected and run simultaneously to ensure that the sensor results are properly displayed and saved in the user interface. The color sensor portion of the programming is designed to take the color value and compare it to hard-coded color values which represent the values provided with the test strips. The results will be compared and mapped to the closest color match to return the matching parameter values to the user. This code is the same across all parameters with the only variation being in the color values used for mapping and the values associated with those colors.

In order to avoid errors, the sensor control block incorporates error-handling mechanisms. Should it lose connection to any of the sensors it will indicate an error and work to re-establish the connection. This helps the system to remain reliable for the user. Additionally, the color sensor enters a low-power state when not in use to save power and align with the goal of energy efficiency within our system.

## 2. Design

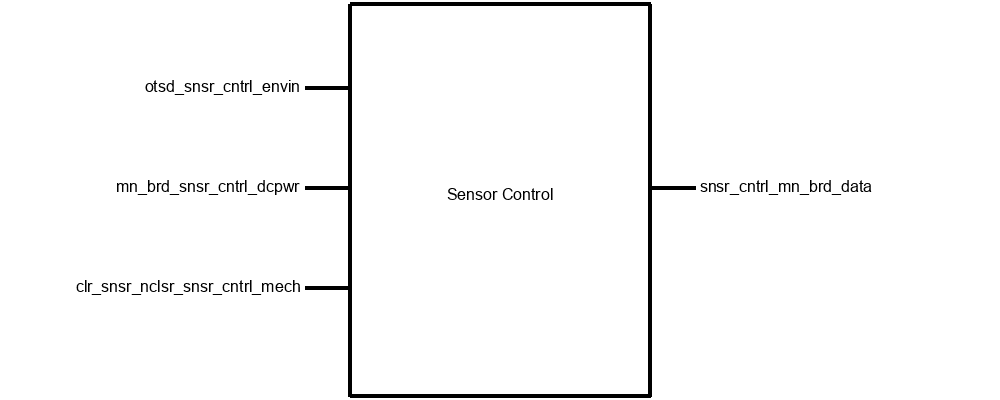


Figure 4: Sensor Control Block Diagram; Inputs: otsd\_snsr\_cntrl\_envin, mn\_brd\_snsr\_cntrl\_dcpwr, clr\_snsr\_nclsr\_snsr\_cntrl\_mech; Outputs: snsr\_cntrl\_mn\_brd\_data

General Code Flow

1. Initialize Color Sensor
   1. Set up the necessary configurations for the color sensor
   2. Calibrate the sensor if required for accurate color readings
2. Initialize Temperature Sensor
   1. Set up configurations for the temperature sensor
   2. Ensure continuous temperature sensing is enabled
3. Main Loop
   1. Enter the main loop for continuous operation
   2. Read Temperature continuously
      1. Read the temperature from the temperature sensor
4. Read Color Data
   1. Trigger the color sensor to read the colors from the test strip
   2. Capture the color values from the five squares on the test strip
5. Interpret Color Data
   1. Process the color data to determine the parameter values
   2. Utilize a color key or mapping to convert color information into specific parameter values
6. Send Data to Main Board
   1. Package the interpreted data, including color parameter values and temperature
   2. Send the data to the main board for further processing
7. Delay
   1. Introduce a delay to control the frequency of readings
   2. Adjust the delay to balance real-time responsiveness and system efficiency
8. Repeat
   1. Go back to the main loop to continue the sensor readings
9. Error Handling
   1. Implement error-checking mechanisms to handle unexpected issues
   2. Log or notify the system of any errors encountered during operation
10. Power Management
    1. Implement power-saving features if applicable
    2. Consider putting the color sensor or other components into a low-power state during idle periods
11. End
    1. Include procedures for proper shutdown or termination when needed

## 3. General Validation

The sensor control block's design has been crafted to meet the specific needs of our system. It has been designed to ensure efficient and accurate operation in monitoring the aquarium parameters. The integration of its interfaces contributes to the comprehensive functionality of the color and temperature sensors, facilitates seamless communication with the main board, and provides structural integrity to the sensor control block.

The initialization of the color sensor is vital to its accurate functioning. By setting up the necessary configurations and calibration, the design ensures that the sensor is ready to capture precise color information from the test strips. Alternatively, we considered purchasing individual sensors such as a sensor that is only designed to collect pH data. We decided against this due to the high price of these as well as the complexity of trying to use several sensors in a single system. The initiation of the temperature sensor follows a similar approach, configuring it for continuous temperature sensing to provide real-time water temperature data.

The main loop controls the continuous operation of the sensor control block. Its systematic reading of color data triggers the color sensor to capture test strip information. The temperature sensor reads the current water temperature in this loop as well. Data from both of these sensors is interpreted based on predefined keys set up in the sensor control block so that the parameters can be properly observed.

Communication with the main board allows for interpreted data to be packaged and transmitted. The versatility of the design is evident in the implementation of error-handling mechanisms. These mechanisms protect against unexpected issues. Error logs or notifications contribute to the system's reliability and enable prompt responses to anomalies during operation.

Power management is crucial and our design includes provisions for power-saving features, such as allowing the color sensor to enter a low-power state when not in use. This contributes to our goal of being energy-efficient in our design. Upon shutdown, all resources are properly cleaned up and the system maintains integrity for further use.

## 

## 4. Interface Validation

User Interface Block Interface Validation

| **Interface** | **Interface Property** | **Why is this interface this value?** | **Why do you know that your design details for this block**  **above meet or exceed each property?** |
| --- | --- | --- | --- |
| otsd\_snsr\_cntrl\_envin : Input | Other: Color sensor reading between #000000 and #FFFFFF | The color sensor must be able to read all possible color values from black (#000000) to white (#FFFFFF) so that any and all parameter values can be outputted to the user based on the test strip color results | Our color sensor is calibrated to accurately capture color information across the whole range of colors, validated through testing for precision and reliability |
| otsd\_snsr\_cntrl\_envin : Input | Temperature (Absolute): temperature of water measured (between 50-100 degrees Fahrenheit) | Different aquatic environments require different water temperatures, so the values were set to accommodate both cold water environments and warm water environments | Our temperature sensor is calibrated to accurately capture temperature readings within the range of 5-100 degrees Fahrenheit, validated through testing for precision and reliability |
| otsd\_snsr\_cntrl\_envin : Input | Water: total immersion of temperature sensor in water | The temperature sensor is required to be submerged in water to properly read the temperature so it must be totally immersed in water | Our design specifies total immersion to prevent potential inaccuracies from partial exposure or water level fluctuations; the temperature sensor's waterproof design has undergone testing to ensure functionality and accuracy when fully immersed in water |
| mn\_brd\_snsr\_cntrl\_dcpwr : Input | Inominal: 210 uA | The nominal current input is set to 210 uA to ensure proper functioning of the sensor control block without overloading it | The design specifies the nominal current input as 210 uA, ensuring that the sensor control block operates within the specified limits, meeting the property requirements |
| mn\_brd\_snsr\_cntrl\_dcpwr : Input | Ipeak: 300 uA | The peak current input is set to 300 uA to handle temporary increases in power demand without causing issues within the system. | The design accounts for peak current input of 300 uA, allowing the sensor control block to handle temporary increases in power demand, meeting the property requirements |
| mn\_brd\_snsr\_cntrl\_dcpwr : Input | Vmax: 2.0v | The maximum voltage is set to 2.0 volts to prevent damage to the sensor control block from excessive voltage. | The design ensures that the maximum voltage input does not exceed 2.0V, preventing damage to the sensor control block and meeting the property requirements |
| mn\_brd\_snsr\_cntrl\_dcpwr : Input | Vmin: 1.7v | The minimum voltage is set to 1.7 volts to ensure the sensor control block receives enough voltage for proper functionality. | The design specifies the minimum voltage input as 1.7V, ensuring that the sensor control block receives adequate voltage, meeting the property requirements |
| snsr\_cntrl\_mn\_brd\_data : Output | Datarate: 400 kbit/s | The datarate of 400kbit/s ensures efficient communication between the sensor control block and the main board. | The design aligns with the specified datarate of 400 kbit/s, ensuring optimal communication without introducing delays, meeting the property requirements for efficient data transfer |
| snsr\_cntrl\_mn\_brd\_data : Output | Messages: sensor sends color data (in hex form) to board to be interpreted into water parameter values based on the test strip color key | The messages ensure effective communication and interpretation of color data by the main board for deriving water parameter values. | The design facilitates the transmission of color data in hex form, allowing the main board to interpret the data into water parameter values based on the test strip color key, meeting the property requirements |
| snsr\_cntrl\_mn\_brd\_data : Output | Protocol: I2C | The I2C protocol supports bi-directional communication, providing a reliable and standardized communication method for the sensor control block and the main board | The selection of the I2C protocol ensures reliable and standardized communication between the sensor control block and the main board, meeting the property requirements for effective communication |
| clr\_snsr\_nclsr\_snsr\_cntrl\_mech : Input | Fasteners: clips | Fasteners in the form of clips are used in the mechanical design to secure components in place | The design incorporates clips as fasteners in the mechanical structure, ensuring secure placement of components, meeting the property requirements for effective fastening |
| clr\_snsr\_nclsr\_snsr\_cntrl\_mech : Input | Other: MAT PETG | The material PETG (polyethylene terephthalate glycol) with a maximum temperature of 230°C is used for specific components in the mechanical design | The use of MAT PETG, with a specified maximum temperature, meets the property requirements for material selection, ensuring the mechanical components can withstand the intended operating conditions |
| clr\_snsr\_nclsr\_snsr\_cntrl\_mech : Input | Other: temp < 230 C | The temperature limit for components made of MAT PETG is set to be below 230°C to prevent degradation or damage to the material | The design specifies a temperature limit below 230°C for components made of MAT PETG, ensuring the material's integrity, meeting the property requirements for effective thermal management |

## 

## 5. Verification Plan

1. Testing Temperature Sensor Initialization (otsd\_snsr\_cntrl\_envin):
   1. Power up the sensor control block (mn\_brd\_snsr\_cntrl\_dcpwr).
   2. Verify that the temperature sensor (otsd\_snsr\_cntrl\_envin) initializes successfully.
   3. Confirm that the necessary configurations for the temperature sensor are set.
   4. Check if continuous temperature sensing is enabled.
2. Testing Color Sensor Initialization (otsd\_snsr\_cntrl\_envin):
   1. Power up the sensor control block (mn\_brd\_snsr\_cntrl\_dcpwr).
   2. Ensure the color sensor (otsd\_snsr\_cntrl\_envin) initializes correctly.
   3. Confirm that the necessary configurations for the color sensor are set.
   4. Check calibration if required for accurate color readings.
3. Main Loop - Temperature Reading (otsd\_snsr\_cntrl\_envin, snsr\_cntrl\_mn\_brd\_data):
   1. Enter the main loop for continuous operation.
   2. Monitor continuous temperature readings from the temperature sensor (otsd\_snsr\_cntrl\_envin).
   3. Confirm that the temperature values are being read accurately.
   4. Check if the temperature values are constantly displayed on the user interface (snsr\_cntrl\_mn\_brd\_data).
4. Main Loop - Color Data Reading (otsd\_snsr\_cntrl\_envin, snsr\_cntrl\_mn\_brd\_data):
   1. Trigger the color sensor (otsd\_snsr\_cntrl\_envin) to read colors from the test strip.
   2. Capture color values from the five squares on the test strip.
   3. Confirm that the color sensor data is sent to the main board (snsr\_cntrl\_mn\_brd\_data).
   4. Ensure the data is converted into color values in hex code on the main board.
5. Interpret Color Data (otsd\_snsr\_cntrl\_envin, snsr\_cntrl\_mn\_brd\_data):
   1. Process the color data to determine parameter values.
   2. Utilize a color key or mapping to convert color information into specific parameter values.
   3. Verify that the interpreted color parameter values align with the expected results.
6. Send Data to Main Board (otsd\_snsr\_cntrl\_envin, snsr\_cntrl\_mn\_brd\_data):
   1. Package the interpreted data, including color parameter values and temperature.
   2. Confirm that the data is sent to the main board (snsr\_cntrl\_mn\_brd\_data) for further processing.
7. Delay and Repeat (otsd\_snsr\_cntrl\_envin):
   1. Introduce a delay to control the frequency of readings.
   2. Adjust the delay to balance real-time responsiveness and system efficiency.
   3. Confirm that the system repeats the main loop for continuous sensor readings.
8. Error Handling (otsd\_snsr\_cntrl\_envin):
   1. Simulate a loss of connection to any of the sensors.
   2. Verify that an error is indicated by the sensor control block.
   3. Confirm that the block works to re-establish the connection.
9. Power Management (otsd\_snsr\_cntrl\_envin):
   1. Test the low-power state functionality of the color sensor (otsd\_snsr\_cntrl\_envin) when not in use.
   2. Confirm that the color sensor enters a low-power state to save power.
   3. Validate that this aligns with the goal of energy efficiency within the system.
10. Shutdown Procedure (otsd\_snsr\_cntrl\_envin):
    1. Initiate a shutdown or termination process.
    2. Ensure all resources are properly cleaned up.
    3. Confirm that the system maintains integrity for further use after shutdown.
11. Interface Validation (otsd\_snsr\_cntrl\_envin):
    1. Validate each input interface condition individually.
    2. Confirm that each input condition produces the specified output interfaces and associated properties.
    3. Ensure that color sensor readings cover the entire range (#000000 to #FFFFFF).
    4. Verify that temperature sensor readings fall within the specified range (50-100 degrees Fahrenheit).
    5. Confirm total immersion of the temperature sensor in water for accurate readings.
12. Mechanical Design Validation (clr\_snsr\_nclsr\_snsr\_cntrl\_mech):
    1. Test the mechanical design with clips as fasteners (clr\_snsr\_nclsr\_snsr\_cntrl\_mech).
    2. Verify the use of MAT PETG as specified for specific components (clr\_snsr\_nclsr\_snsr\_cntrl\_mech).
    3. Ensure the temperature limit for MAT PETG components is below 230°C.

## 6. References and File Links

### 6.1 References

### 6.2 File Links

[Code Folder](https://drive.google.com/drive/folders/1PNjqr1ornI5Qnpzj_nd4JuXxndOOEzkr?usp=drive_link)

## 7. Revision Table

| **Date** | **Action** |
| --- | --- |
| 1/24/2024 | Document creation |
| 1/25/2024 | Sensor Control section created; sections 1-7 completed  Document completed |