# **Color Sensor Module Block Validation**

Block Champ: Caleb Neill

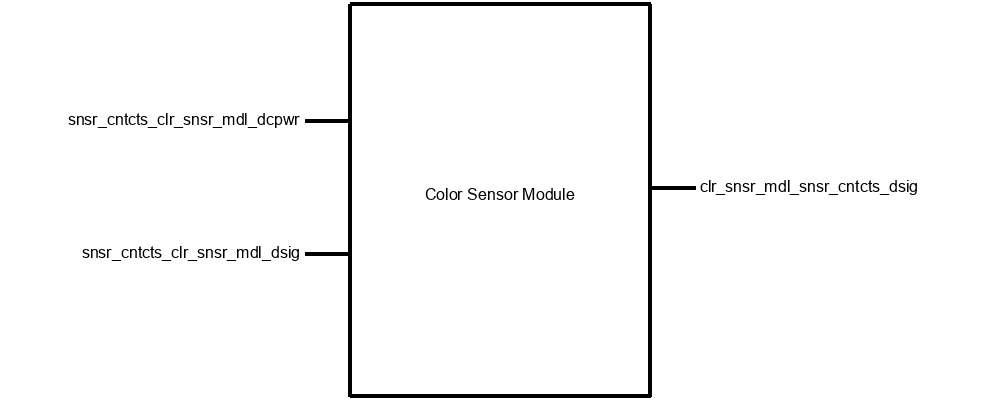
### **1. Description**

The Color Sensor module is responsible for moving the color sensor up and down to scan the activated test strips. It must move in a controlled manner so that we can get accurate color readings of the test strips and therefore accurate measurements of each chemical test of the tank water. To achieve this goal, much thought has been put into the mode of translation. Originally when the plan was to maneuver the test strips using a conveyor belt the strips simply would have passed underneath the sensor. But as mentioned in the previous block, the conveyor belt comes with an array of problems. So the change to shifting the color sensor occurred. The choice then needed to be made of how to have consistent vertical translation. The first idea was using a spirally stainless steel rod for a screw drive, but fitting the screw and guiding rods into the tight working area proved to be more difficult than originally thought.. The next design Idea was a chain system, but a chain system poses the problem of needing tension on the chain, and chain breaks or jamming is common. The final idea is to use a rack and pinion system, which while requiring pressure to keep the teeth aligned and meshing.

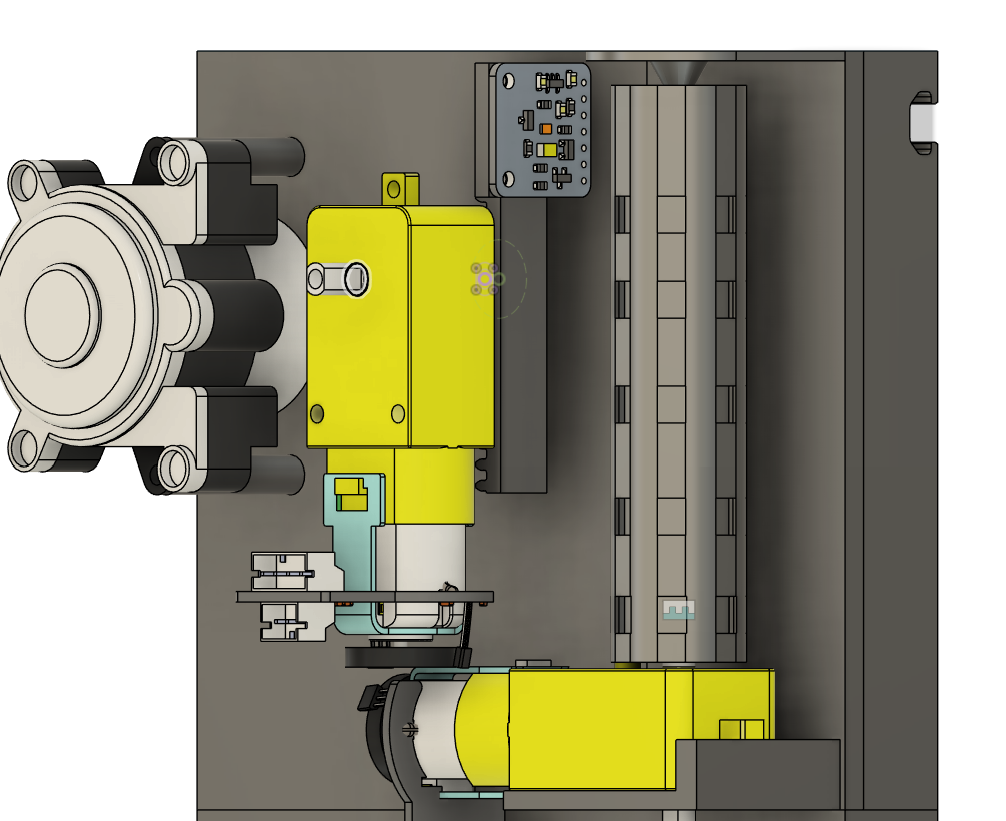
Another important consideration is that the color sensor module is located underneath the test strip activation module, and it might be exposed to water. To limit the exposure, the module's resting position will be at the top of the fixture, which is confirmed with a limit switch mounted at the highest position. The color sensor then is above the first hole for the test strip to be viewed from, and hopefully out of splash range. However, as an extra precaution, the board will also be coated in an epoxy resin other than the photoresistor or led to not affect data collection. Another important design choice was to add connectors for the color sensor directly in the module so that if the color sensor fails, it can easily be removed by removing two screws and pulling it away from the gantry terminals. This whole system will be tied closely with both the test strip cylinder and the test strip activation module, with the activation module being mounted above the test strip cylinder and color sensor module, able to disperse water in the test strip cylinder, which is parallel with the color sensor module, which can then move up and down to read the activated test strip.

### 2. Design

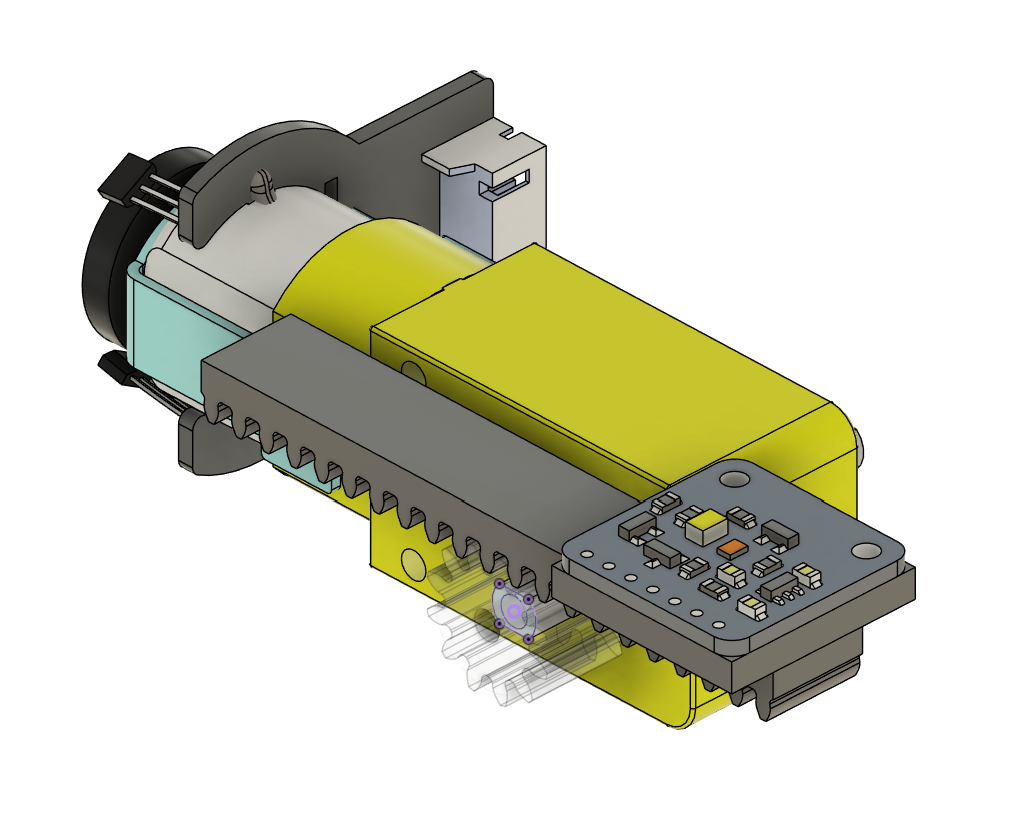
There are three interfaces for the Color Sensor Module, which will be addressed from top to bottom, starting with the inputs. Sensor contacts color sensor module dc power denotes the power supplied to the motor and encoder, along with the limit switch. Sensor contacts color sensor module digital signal is the pwm signal sent from the main board motor controller to the dc motor to translate the color sensor gantry up and down to read all test strip positions. Finally, the color sensor module sensor contacts a digital signal transmits data from the motor encoder, and limit switch back to the main board so the position of the color sensor can be accurately tracked.

****

**Block Diagram**



**General Testing Assembling**



**Rack and Pinion Assembly**

### 3. General Validation

A second major consideration was what materials could safely interact with the tank water without adding any debris or toxic chemicals, yet still hold up to wear and tear. It was eventually decided that PETG would be used, as it isn’t biodegradable and is food-safe. For these reasons, it was chosen. A second important decision that I have previously discussed was choosing what mechanical method to use to move the color sensor up and down to read the test strips. The most mechanically simple design was the screw drive method, which is why I have chosen it. The general procedure is when the color sensor is not in use, it is held at max height to avoid splashing. Once a new test strip has been activated, it is turned on and moved from the top to the bottom, scanning the tabs along the way. Once it has reached the bottom it turns off and is returned to the max position at the top.

### 4. Interface Validation

| **Interface Property** | **Why is this interface property this value?** | **Why do you know that your design details for this block**  **above meet or exceed each property?** |
| --- | --- | --- |
| **clr\_snsr\_mdl\_snsr\_cntcts\_dsig: Output** | | |
| Logic-Level: Active High | This logic level was chosen as it has returned to the based position when the logic level is high. | It is known that 3pin limit switches can be set to active low or active high based on how they are wired up, and having them set to active high is more convenient in this case for interrupts. |
| Vmax: 3.3V | This is simply a data line, so a standard 3.3V for logic high was chosen. | For the chosen motor, this is to specification as shown by its datasheet. [1] |
| Vmin: 0V | This is simply a data line, so a standard 0V for logic low was chosen | For the chosen motor, this is to specification as shown by its datasheet. [1] |

| **snsr\_cntcts\_clr\_snsr\_mdl\_dsig: Input** | | |
| --- | --- | --- |
| Max Frequency: 115200bps | This value is pulled directly from the esp32 manual and is the typical operating baud rate. | As the motor is being run with a PWM signal, 115200 should exceed the amount of granularity necessary to operate the motor at a reasonable speed. |
| Vmax: 3.3V | This is simply a data line, so a standard 3.3V for logic high was chosen. | For the chosen motor, this is to specification as shown by its datasheet. [1] |
| Vmin: 0V | This is simply a data line, so a standard 0V for logic low was chosen | For the chosen motor, this is to specification as shown by its datasheet. [1] |

| **snsr\_cntcts\_clr\_snsr\_mdl\_dcpower: Input** | | |
| --- | --- | --- |
| Inominal: 500 mA | This value was chosen based on the current power draw of the motor | As the motor is being run somewhat intermittently, as well as little power will be required as the screw drive gives a torque advantage and the color sensor module will be quite light as it is two bearings a bit of PETG and the color sensor pcb. |
| Ipeak:  2.8 A | This value was specified by the motor datasheet | We should have no problem keeping under this amount as the color sensor gantry is very light. |
| Vmax: 5.1V | This is the operating voltage given by the dc power convertor. | For the chosen motor, this is inside the specified limits. [1] (7.5 V ~ 4.5 V) |
| Vmin: 4.9V | This is the operating voltage given by the dc power convertor. | For the chosen motor, this is inside the specified limits. [1] (7.5 V ~ 4.5 V) |

### 5. Verification Plan

For Interface **clr\_snsr\_mdl\_snsr\_cntcts\_dsig**

1. Connect the limit switch output to an oscilloscope.
2. Supply voltage 3.3v.
3. Press and release the switch, causing a logic high and low, respectively.
4. Connect the encoder outputs to an oscilloscope.
5. Supply voltage 3.3v.
6. Rotate the encoder, noticing that one way A follows B in oscillation, and rotating the other way B follows A.

For Interface **snsr\_cntcts\_clr\_snsr\_mdl\_dcpower**

1. Connect an oscilloscope to the power lines run to the motor
2. Send a PWM signal over the data lines
3. Confirm that power draw, current, and voltage are all within spec.

For Interface **snsr\_cntcts\_clr\_snsr\_mdl\_dsig**

1. Connect an oscilloscope to the data lines run to the motor
2. Send a PWM signal over the data lines
3. Confirm that signals are being sent using the oscilloscope and confirm the rotation of the motor.

### 6. References and File Links

#### 6.1 References

**[1] “Micro\_DC\_Motor\_with\_Encoder-SJ01\_SKU\_\_FIT0450,” DFRobot, https://wiki.dfrobot.com/Micro\_DC\_Motor\_with\_Encoder-SJ01\_SKU\_\_FIT0450 (accessed Jan. 28, 2024).**

### 7. Revision Table

| Date | Action |
| --- | --- |
| 3/10/2024 | Added CAD Designs and general revisions |
| 12/6/2024 | Initial Drafting |
| 1/28/2024 | Updated Draft |

# Self Reflection:

I would give myself an 8/10. I have put a significant amount of time and effort into completing this documentation set but I haven’t had much time to do actual design work to get things working. Ultimately I think this format in my particular situation where prototyping is relatively quick and easy is silly to plan out all these ideas without physical proof that the design works. Design should be iterative, where you build on top of working ideas, not spend so much time trying to verify something works in a simulation just to break in real-world testing. This might be different with other projects, but I think general testing, then building a working design, and documenting the ideas as you go along would make much more sense for all groups who don’t have a single shot to test something, which from talking to other students seems like a majority of the class. This assignment was overall overwhelming and did very little to move the design forward. Hopefully in the future OSU will reformat this class and not require a WIC on top of technical writing, which has multiple assignments that should fill the same requirement.