Test Results:

The test plan is outlined in the Test Plan document. The tests are executed as mentioned in the document. The outcomes of the all the tests are documented below.

1. **CBFIFO Testing:**
   Test Type: Automatic test suite

Test	Expected Outcome	Actual Outcome
Automated Test Suit	All test cases passed	All test cases passed

Conclusion: All test cases have passed and the CBFIFO is up and running!

2. **LED Testing:**
   Test Type: Semi – Automatic test
   The LEDs are lit in a sequence using delays after which PWM transitions are tested.

Test	Expected Outcome	Actual Outcome
White LED on	White LED on for 1 second	White LED on for 1 second
Red LED on	Red LED on for 1 second	Red LED on for 1 second
Green LED on	Green LED on for 1 second	Green LED on for 1 second
Blue LED on	Blue LED on for 1 second	Blue LED on for 1 second
PWM Testing starts		
White -> Red	Two second PWM transition	Two second PWM transition
Red -> Green	Two second PWM transition	Two second PWM transition
Green -> Blue	Two second PWM transition	Two second PWM transition
Blue -> White	Two second PWM transition	Two second PWM transition
Led off	LED turned Off	Led turned Off

Conclusion: All test cases have passed, and the LED works as expected. The PWM functionality also works as expected and all the color transitions are smooth.

3. **Interrupt Testing:**
   Test Type: Semi – Automatic test
   The user is asked to press the switch and touch sensor to check if interrupts are working.

Test	Expected Outcome	Actual Outcome
Switch Interrupt	Switch press detected	Switch press detected
Touch Sensor – touch left side	Touch sensor touch detected	Touch sensor touch detected
Touch Sensor – touch middle	Touch sensor touch detected	Touch sensor touch detected
Touch Sensor – touch right side	Touch sensor touch detected	Touch sensor touch detected

Conclusion: All the test cases have passed, and the switch interrupt works as expected. The touch sensor interrupt also works as touches were detected in left, right, and middle of the touch sensor.

4. **Accelerometer and I2C testing:**
Test Type: Semi – Automatic test
The user is asked to move the DAG to the required angles shown below to see if the accelerometer and I2C is working as expected.

Tests	Expected Outcome	Actual Outcome
Move DAG to 0 degrees	0 degrees detected	0 degrees detected
Move DAG to 30 degrees	30 degrees detected	30 degrees detected
Move DAG to 60 degrees	60 degrees detected	60 degrees detected
Move DAG to 90 degrees	90 degrees detected	90 degrees detected
Move DAG to 120 degrees	120 degrees detected	120 degrees detected
Move DAG to 150 degrees	150 degrees detected	150 degrees detected
Move DAG to 180 degrees	180 degrees detected	180 degrees detected
Move DAG to 0 degrees	0 degrees detected	0 degrees detected

Conclusion: All the test cases have passed, and the DAG is capable of measuring a range of angles. Thus, the accelerometer sensor and I2C is configured correctly and working as expected.

5. **Manual Testing:**

Command Processor:
Test Type: Manual Testing
A variety of commands are entered to check the error handling capability of the command processor.

Command	Expected Outcome	Actual Outcome
Author	Name of Author	Name of author
Info	Various insights	Various Insights
Calibrate	Successful calibration	Successful calibration
Set <angle>	Set the angle by moving DAG	Set the angle by moving DAG
Help	Information on all commands	Information on all commands
Autor	Unknown Command: Autor	Unknown Command: Autor
Author 26	Invalid input for author -- Check help	Invalid input for author -- Check help
“ info”	Works as expected – no change	Works as expected – no change
cAliBrAtE	Works as expected – no change	Works as expected – no change
Set -45	Enter a valid set angle!!	Enter a valid set angle!!
Set 181	Enter a valid set angle!!	Enter a valid set angle!!
Calibrate 36	Invalid Command for calibrate -- look at help	Invalid Command for calibrate -- look at help
Set abcf	Enter a number as an input	Enter a number as an input
SeT !!@#	Enter a number as an input	Enter a number as an input
SeT	Works as expected – no change	Works as expected – no change

Conclusion: The command processor is robust when it comes to handling erroneous inputs from the user.