PRODUCT OUTGOING QUALITY INSPECTION REPORT

SKY15-P20A

S/N: SR2025012002001

Product Manager	R&D Leader	Mechanical Engineer	System Engineer	Test Engineer
Honwen Hung	James Chan	Fulin Chuang	William Cheng	Mike Chen

Date: May 21, 2025

Outgoing Quality Inspection

SKY15-P20A

INDEX

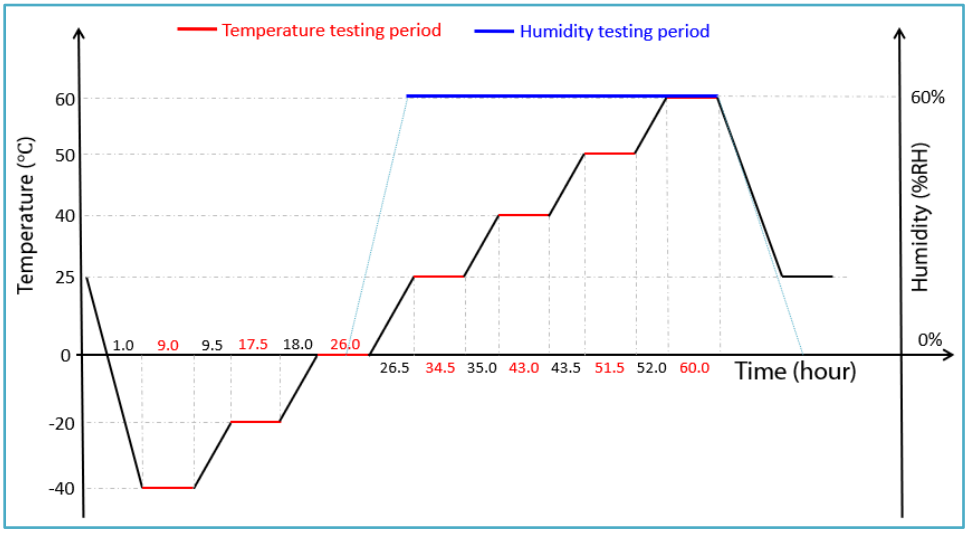
1. SPECIFICATION

1-1. SYSTEM CONFIGURATION

Display	4:3 panel, 1024 × 768 resolution, 1000 cd/m^2 typical brightness, ≥900:1 contrast ratio, LED backlight.
A/D Board	HDMI input.
Function Key	F1~F20 Function Key.
Touch Screen	No touch panel is integrated in this version.

2. TEST PLAN

2-1. THERMAL MEASUREMENT PROCESS

Test Purpose	The purpose of conducting thermal profile testing is to identify potential thermal issues in the Equipment Under Test (EUT). Given that semiconductor failure rates increase significantly with rising junction temperatures, this testing contributes to the overall assessment of product reliability. As the system undergoes a cooling phase, operational modes may shift depending on stack configuration, temperature, and heat dissipation characteristics. Thermal mapping provides critical insight for optimizing thermal management strategies and determining the most effective component layout and monitoring arrangements.
Test Equipment	1. KSON THS-B4T-150 Chamber.
Quantity Tested	Minimum 1 Set
Test Software	N/A
Test Procedure	1. Thermal Pre-Scan Measurement: Temperature Range: **-40°C to 60°C** Humidity Condition: **60% RH** (when temperature exceeds 25°C) 2. Actual Thermal Measurement Procedure: 2.1. Identify the test points using the infrared thermal image and attach thermocouples to the identified hot spots. 2.2. Place the Equipment Under Test (EUT) in the thermal chamber and configure the test temperature profile according to the specified requirements. 2.3. After running the test continuously for 8 hours, record the peak temperature observed at each thermocouple measurement point. 2.4. Power off both the thermal chamber and the EUT. 2.5. Verify that the recorded temperature data for each component remains within its specified operating temperature range, as defined in the component specification or approval documents.
Test Diagram of Curves	Environment defines for 60 hours. 

2-2. Test Result

2-2-1. Temperature Cycle

Aging tests were performed on individual components across a range of temperature settings, under both maximum load and full load conditions, to evaluate thermal endurance and operational stability over time.

Test Temperature	Test Result
-40˚C	PASS
-20˚C	PASS
0˚C	PASS
25˚C / 60%RH	PASS
40˚C / 60%RH	PASS
50˚C / 60%RH	PASS
60˚C / 60%RH	PASS

2-2-2. I/O Function

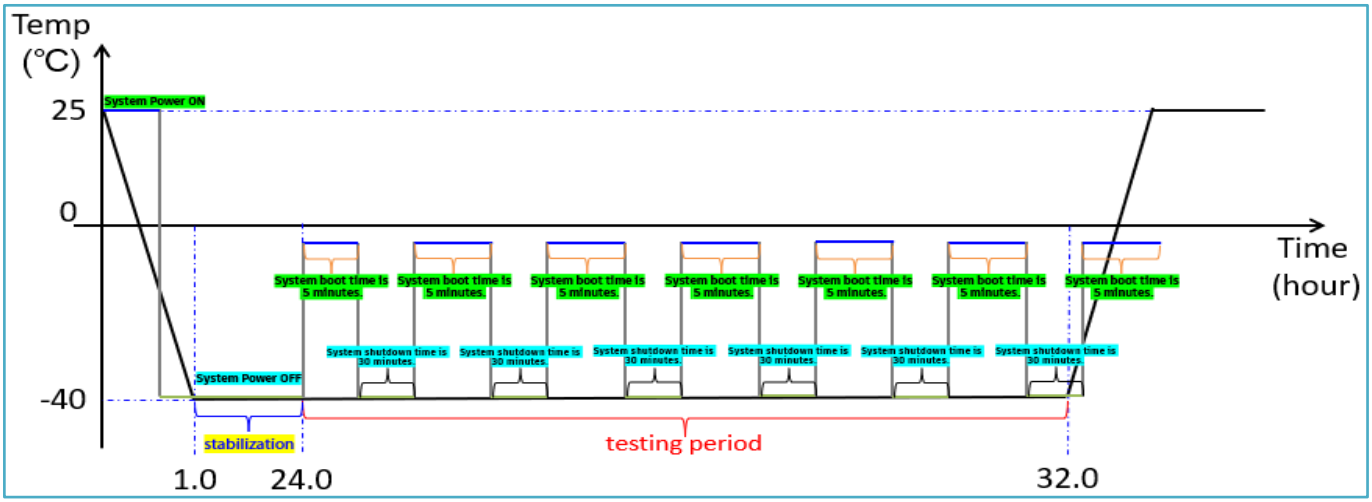
Confirm that the system specifications and all input/output (I/O) interfaces are correctly configured and functioning as intended, in accordance with the defined technical standards.

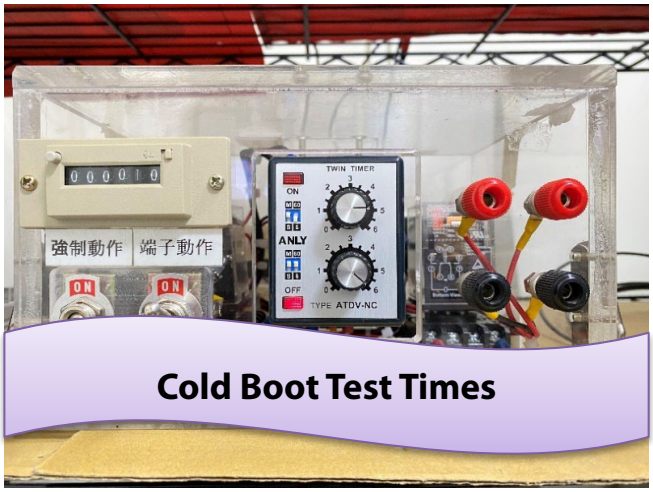
Item	Test Criteria	Result
USB	This component connects to the host and allows configuration of function keys.	PASS
HDMI	HDMI input passed functional verification at 1024 × 768 resolution.	PASS

2-2-3. Low-temperature & Boot-up

Apply power to the system under a -40°C ambient condition and confirm successful system boot-up, ensuring stable initialization and operation at low temperatures.

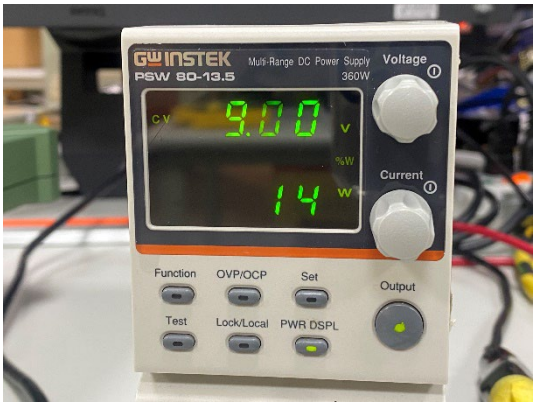
Ambient Temp.	Cold Boot Test Times	Test Result
-40°C	10 times	PASS

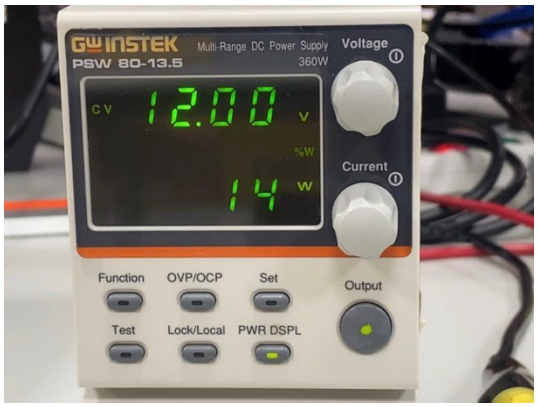


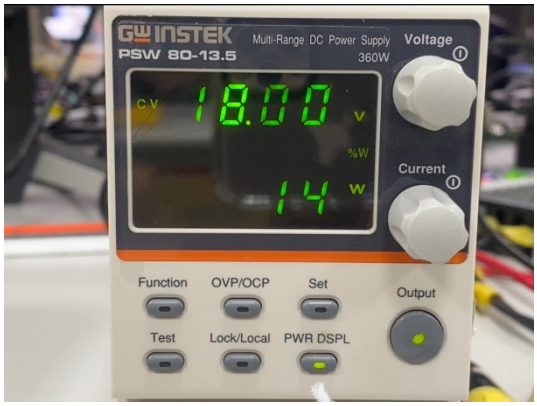


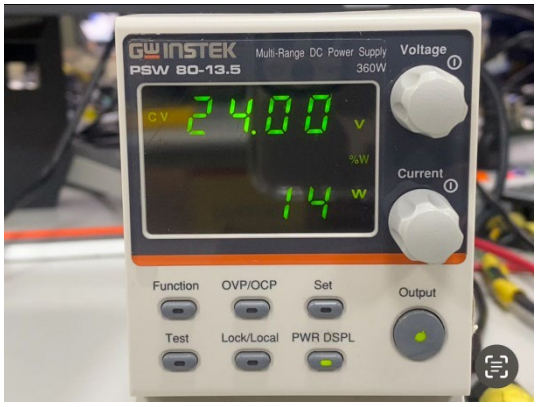
2-3. Power Consumption

Voltage	Current	Wattage
9.0V	1.55A	14W
12.0V	1.17A	14W
18.0V	0.78A	14W
24.0V	0.58A	14W
36.0V	0.39A	14W









3. TEST PHOTO IN LAB

- Chamber in -40°C

- Chamber in -20°C

- Chamber in 0°C

- Chamber in 25°C / 60%RH

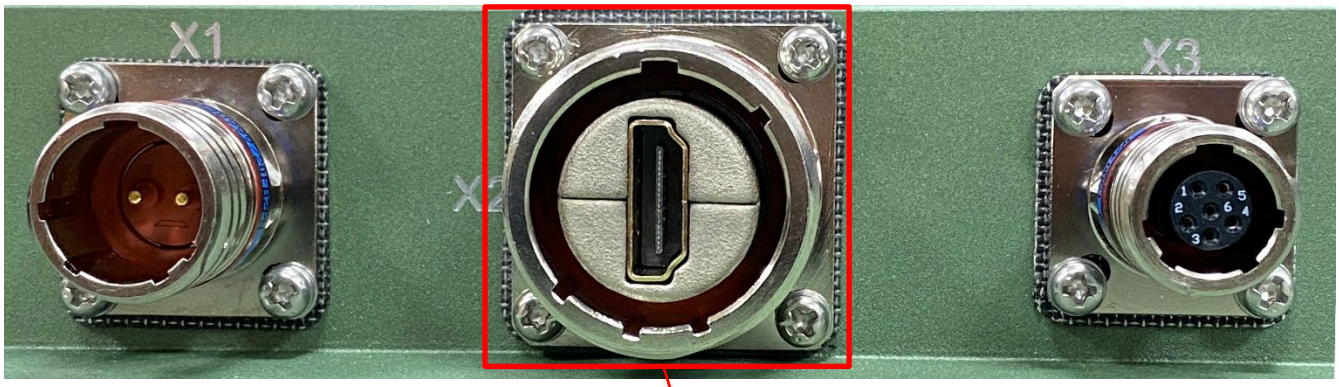
- Chamber in 40°C / 60%RH

- Chamber in 50°C / 60%RH

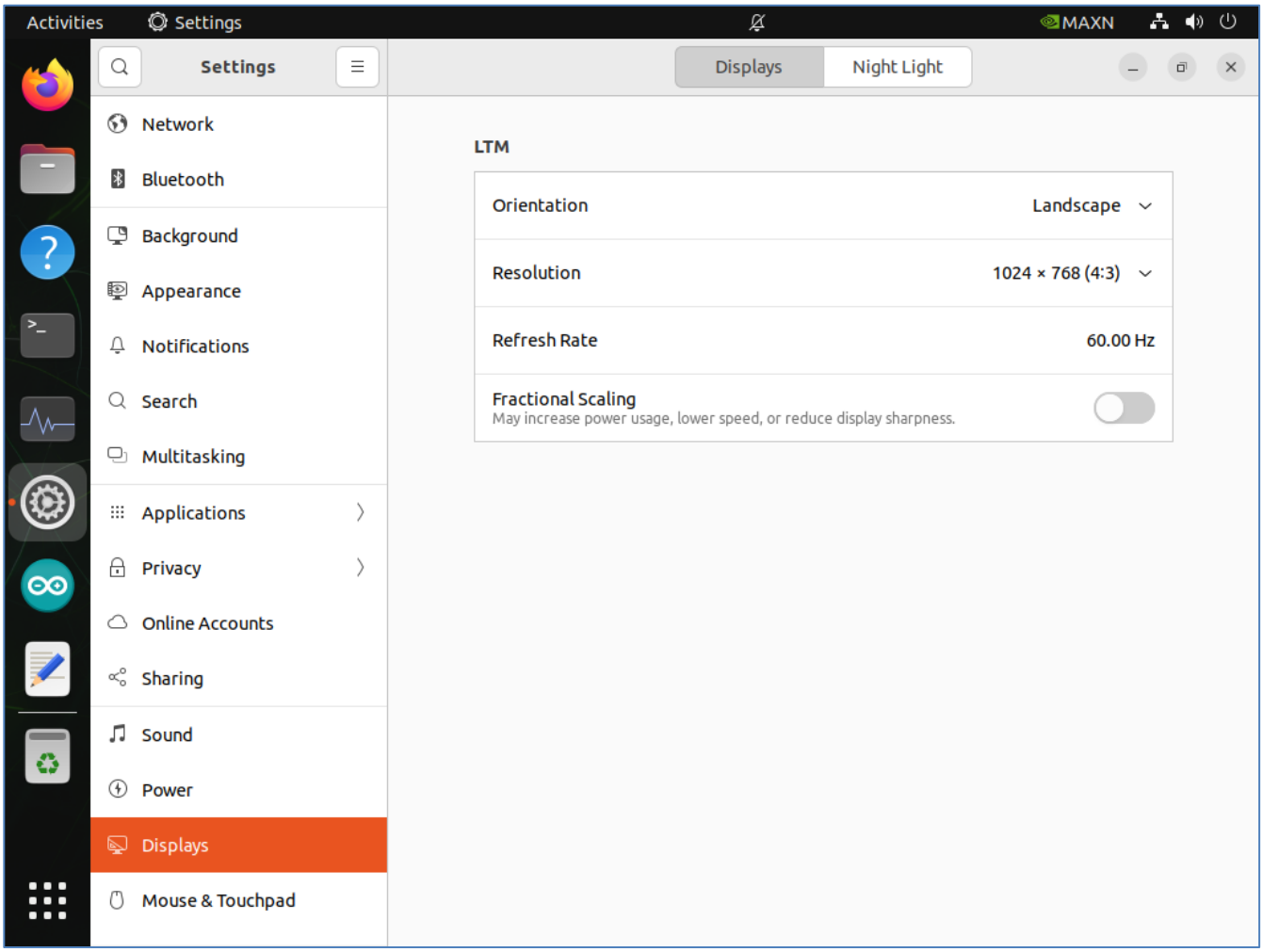
- Chamber in 60°C / 60%RH

4. I/O FUNCTION

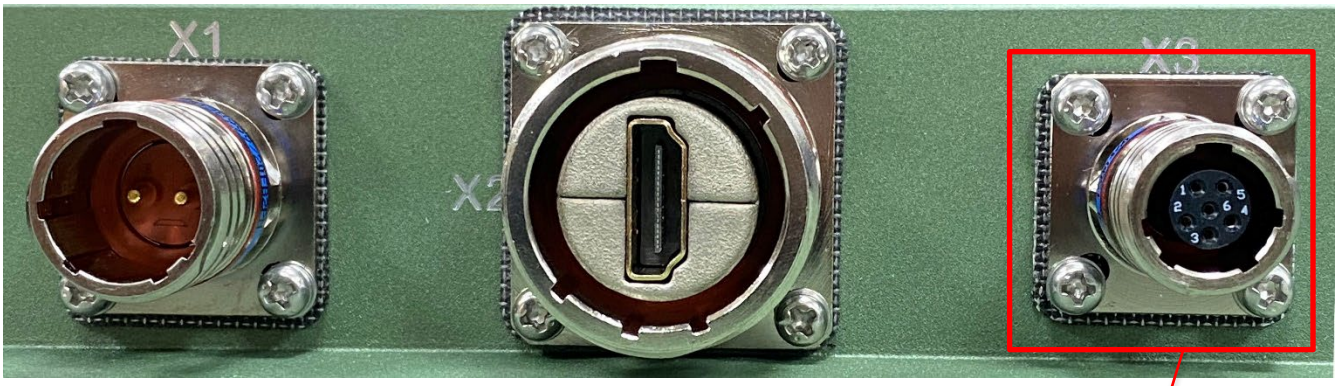
4-1. HDMI (Resulosion: 1024 x 768)

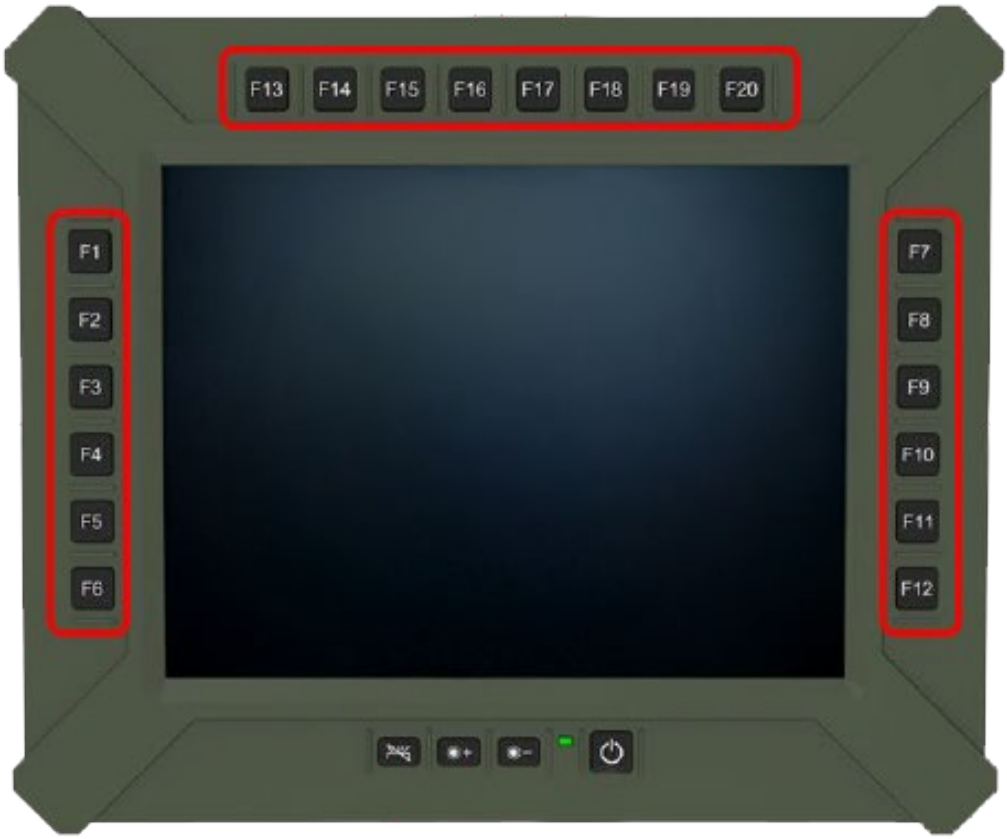


HDMI



4-2. USB (Intended for function key use only)





5. FUNCTION KEY

Key	Test after setting up the command	Result

	if (F1.fallingEdge()) {
	//Keyboard.println("F1 press"); //This displays F1 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_A);
	Keyboard.release(KEY_A);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F2.fallingEdge()) {
	//Keyboard.println("F2 press"); //This displays F2 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_B);
	Keyboard.release(KEY_B);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F3.fallingEdge()) {
	//Keyboard.println("F3 press"); //This displays F3 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_C);
	Keyboard.release(KEY_C);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F4.fallingEdge()) {
	//Keyboard.println("F4 press"); //This displays F4 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_D);
	Keyboard.release(KEY_D);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F5.fallingEdge()) {
	//Keyboard.println("F5 press"); //This displays F5 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_E);
	Keyboard.release(KEY_E);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F6.fallingEdge()) {
	//Keyboard.println("F6 press"); //This displays F6 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_F);
	Keyboard.release(KEY_F);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F7.fallingEdge()) {
	//Keyboard.println("F7 press"); //This displays F7 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_G);
	Keyboard.release(KEY_G);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F8.fallingEdge()) {
	//Keyboard.println("F8 press"); //This displays F8 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_H);
	Keyboard.release(KEY_H);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F9.fallingEdge()) {
	//Keyboard.println("F9 press"); //This displays F9 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_I);
	Keyboard.release(KEY_I);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F10.fallingEdge()) {
	//Keyboard.println("F10 press"); //This displays F10 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_J);
	Keyboard.release(KEY_J);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

SKY15-P20A

Key	Test Command	Result

	if (F11.fallingEdge()) {
	//Keyboard.println("F11 press"); //This displays F11 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_K);
	Keyboard.release(KEY_K);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F12.fallingEdge()) {
	//Keyboard.println("F12 press"); //This displays F12 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_L);
	Keyboard.release(KEY_L);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F13.fallingEdge()) {
	//Keyboard.println("F13 press"); //This displays F13 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_M);
	Keyboard.release(KEY_M);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F14.fallingEdge()) {
	//Keyboard.println("F14 press"); //This displays F14 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_N);
	Keyboard.release(KEY_N);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F15.fallingEdge()) {
	//Keyboard.println("F15 press"); //This displays F15 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_O);
	Keyboard.release(KEY_O);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F16.fallingEdge()) {
	//Keyboard.println("F16 press"); //This displays F16 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_P);
	Keyboard.release(KEY_P);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F17.fallingEdge()) {
	//Keyboard.println("F17 press"); //This displays F17 key is pressed
	Keyboard.press(KEY_CAPS_LOCK);
	Keyboard.press(KEY_Q);
	Keyboard.release(KEY_Q);
	Keyboard.release(KEY_CAPS_LOCK);
	Keyboard.press(KEY_ENTER);
	Keyboard.release(KEY_ENTER);
		PASS

	if (F18.fallingEdge()) {
	//Keyboard.println("F18 press"); //This displays F18 key is pressed
	Keyboard.press(KEY_PRINTSCREEN);
	Keyboard.release(KEY_PRINTSCREEN);
		PASS

	if (F19.fallingEdge()) {
	//Keyboard.println("F19 press"); //This displays F19 key is pressed
	Keyboard.press(MODIFIERKEY_LEFT_CTRL);
	Keyboard.press(KEY_A);
	Keyboard.release(KEY_A);
	Keyboard.release(MODIFIERKEY_LEFT_CTRL);
		PASS

	if (F20.fallingEdge()) {
	//Keyboard.println("F20 press"); //This displays F20 key is pressed
	Keyboard.press(KEY_BACKSPACE);
	Keyboard.release(KEY_BACKSPACE);
		PASS

-----END-----