



## Test Procedure for the NCPXXGEVB Evaluation Board

### Initial setup:

1. Mount ONMCU\_V1 board into SODIMM socket
2. Connect jumpers OL1 – OL12
3. Connect jumpers JVS and JVDD
4. Connect strap NT1 and NT2

### Required Equipment:

- Bench power supply with A-meter
- Multimeter
- NCV7684I2CV1GEVB board

### Test procedure (current regulation)

5. Connect setup as shown above
6. Apply voltage and sweep  $V_{BAT}$  voltage from 0 to 14V
7. The current consumption should not exceed 250mA, the LEDs should be ON with 10mA current setting, the green status LEDs should indicate the voltage and DIAG red status LED should be turned OFF.
8. If something is not working, check the following:
9. Check  $V_{STRING}$  voltage sense using  $J_{UV}$  or  $J_{DCSHORT}$  jumper pin. If  $V_{STRING}$  is not 7.0V the DC/DC circuit is not mounted properly (please check Q1 transistor)
10. Check the VDD digital supply on the VDD jumper, the 5.0V should be present, if not, please check the LDO regulator.
11. Check the VCC output reference on the test point VCC. There should be 3.3V if not, the NCV7684 device is not mounted properly or NCV7684 is not supplied.
12. If the LEDs are not turned ON, check the voltage on the  $I_{SET}$  pin using  $J_{SHRT}$  jumper pin. There should be 1.0V if not, the device is not powered, or  $I_{SET}$  pin is shorted to ground.
13. If some of the LED strings are not turned on, the NCV7684 device is not mounted properly, or the LED connection is broke down.

### Test procedure (High current setting)

1. Apply 12V on the supply connector.
2. Connect JHiCur jumper.
3. The LED intensity should be increased from 10mA to 50mA per channel.
4. Check the current by any of LED Strings using A-meter connected instead of OLx jumper.

### Test procedure (Open Load detection)

1. Apply 12V on the supply connector.
2. Disconnect any of OLx jumpers.
3. The other LED strings should be turned ON, and DIAG led should flag an error.
4. Check the voltage on the DIAG pin. Should be  $<0.7V$ .

### Test procedure (Short on ISET detection)

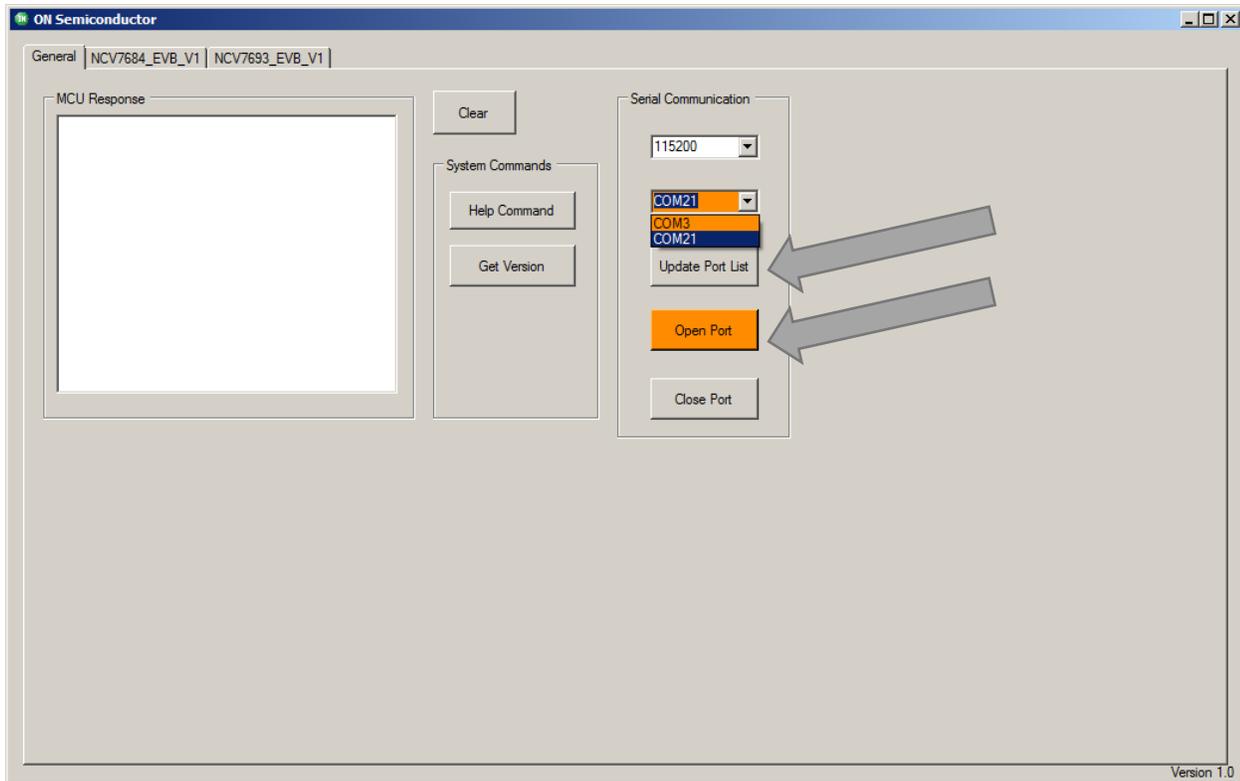
1. Apply 12V on the supply connector.
2. Connect the  $J_{SHRT}$  jumper.
3. The other LED strings should be turned ON, and DIAG led should flag an error.
4. Check the voltage on the DIAG pin. Should be  $<0.7V$ .

**ON Semiconductor**<sup>®</sup>

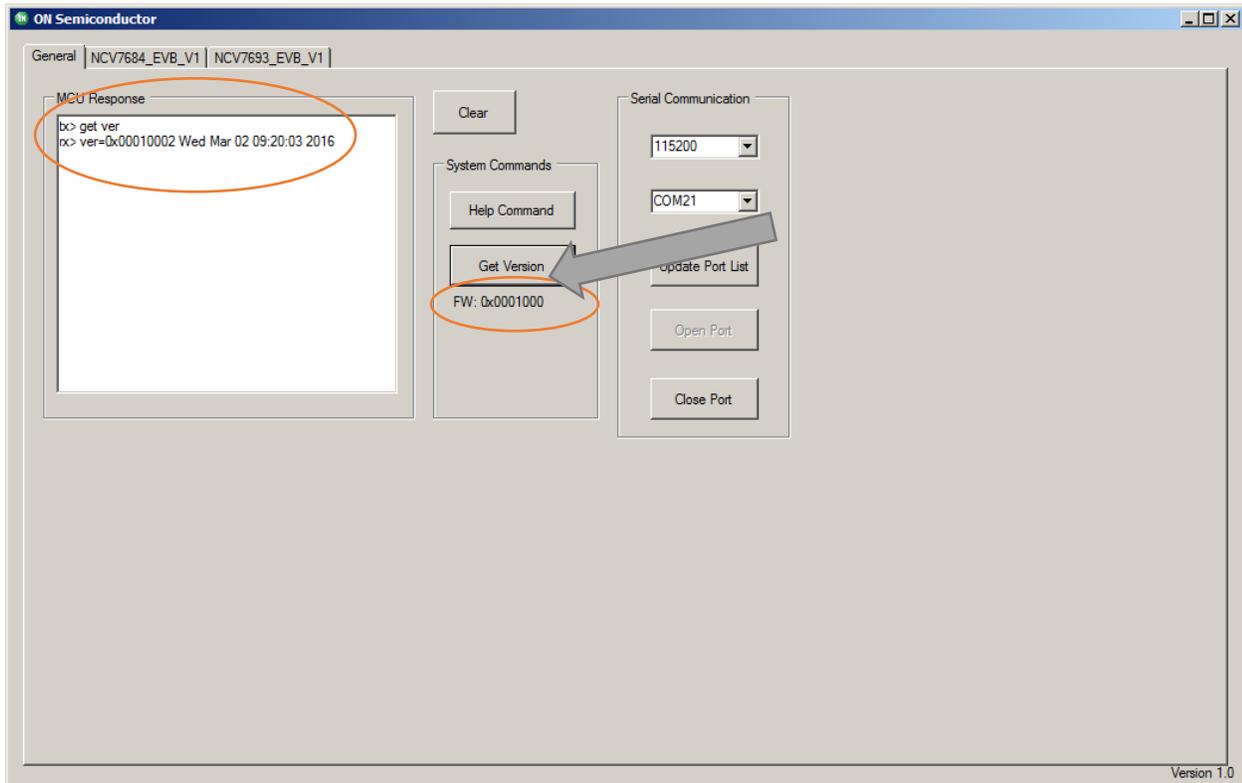


**Control Program description:**

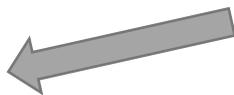
1. Apply 12V on the supply connector.
2. Connect mini USB cable between PC and J\_MUSB connector
3. Press SW\_RES button to reset the MCU
4. Open the Control Software
5. Press the Update Port Number button, select the COM Port and press Open port.



6. The NCV7684I2CV1GEVB board is properly connected if MCU receive the information using Get Version button



7. On the NCV7684\_EVB\_V1 tab click on the Write Config button to initialize the I2C communication at the 400 kHz, the Tx and Rx response from MCU should be visible in the msg box.





The screenshot shows the ON Semiconductor software interface for configuring an NCV7684 device. The 'General' tab is active, showing I2C Speed settings (100 kHz, 400 kHz, 1 MHz) and a 'Write Config' button. The 'I2C1CONF' register is highlighted with a red circle, showing its bit configuration: `0x00` for I2C1CONF, `0x01` for I2C1SPEED=1, `0x00` for I2C1CONF:ok, and `0x00` for I2C1SPEED:ok. The 'Volatile Registers' section shows the I2C\_CONF (W) / I2C\_CH\_STATUS (R) register with the FLAG bit set. The 'Hard Coding Register' section shows the ID\_VERS\_1 and ID\_VERS\_2 registers. The 'OTP registers' section shows the ADD\_SAM\_SET, SAM\_CONF\_1, and SAM\_CONF\_2 registers. The 'Identifiers' section shows the (W) - 0x00 - ID\_I2C\_CONF, (W) - 0x01 - ID\_PWM, (W) - 0x03 - ID\_PWM\_CONF, (R) - 0x08 - ID\_STATUS, (R) - 0x09 - ID\_FAULT, (W) - 0x02 - ID\_WRITEALL, and (R) - 0x0A - ID\_READALL registers. The 'LED setting indication' section shows a grid of red LEDs and a photo of the device with status LEDs.

8. To set NCV7684 device into I2C mode, set the FLAG bit and set some I2C conf bits to turn on the channels. And then click to write button.
9. Only activated outputs should be active. The communication between MCU and Software can be check in the message window.



10. PWM can be enabled by PWMEN bit and controlled by PWM\_GAIN and PWM\_GAIN\_EN register. After each change the appropriate write button has to be pressed. For PWM duty cycle setting, the slider can be used to quickly set the proposed value.



The screenshot displays the ON Semiconductor software interface for configuring registers. Key sections include:

- General:** I2C Speed (100 kHz, 400 kHz, 1 MHz), I2C1DATA, and I2C1DATA:ok.
- Volatile Registers:**
  - I2C\_CONF (W) / I2C\_CH\_STATUS (R):** Includes flags like FLAG, autoR, and PWMEN. A red circle highlights the 'Read' button.
  - PWM\_GAIN / PWM\_GAIN\_EN:** Includes PWMF1, PWMGAIN[8:0], and PWM\_GAIN\_EN[11:0]. A red circle highlights the 'Read' button.
  - PWM\_CONF:** Includes PWMLIN and PWMF2.
  - I2C\_STATUS:** Includes SC\_ISET, UV, TW, DiagErr, I2Cerr, diagRange, TSD, and OL.
  - FAULT STATUS:** Includes FAULT [11:0].
- Hard Coding Register:** ID\_VERS\_1, ID\_VERS\_2, D7, D6, D5, D4, D0.
- OTP registers:** ADD\_SAM\_SET, SAM\_CONF\_1, SAM\_CONF\_2.
- Identifiers:** (W) - 0x00 - ID\_I2C\_CONF, (W) - 0x01 - ID\_PWM, (W) - 0x03 - ID\_PWM\_CONF, (R) - 0x08 - ID\_STATUS, (R) - 0x09 - ID\_FAULT, (W) - 0x02 - ID\_WRITEALL, (R) - 0x0A - ID\_READALL, (W) - 0x20 - ID\_SET\_OTP, (R) - 0x28 - ID\_READ\_OTP, (W) - 0x21 - ID\_LOCK\_OTP.
- LED setting indication:** A section at the bottom right showing a circuit board with LEDs.

11. The registers can be read using Read Buttons. Only for reading OTP registers, the high voltage has to be brought to CSN pin to have access to the Hard coding and OTP registers