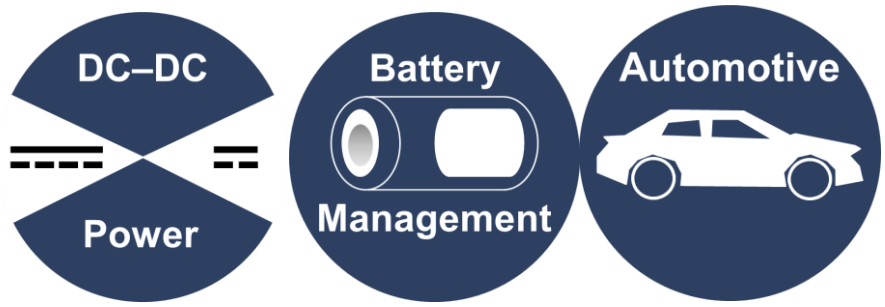
ON Semiconductor

Is Now

To learn more about onsemi™, please visit our website at
www.onsemi.com

AUTOMOTIVE ADAS PREREGULATOR STR-ADAS-PREREGULATOR-GEVK Application Note: JSON commands



Introduction

The Strata Assisted STR-ADAS-PREREGULATOR-GEVK provides an evaluation kit for the NCV881930 Synch Buck Controller and for the NCV6357 configurable 5.0 A Adaptive-On-Time (AOT) Step Down Converter a with I2C programmable output voltage from 0.6 V to 3.3 V in 12.5 mV steps. The NCV6357 also offers multiple operating modes: PFM for low load, Pseudo-PWM (PWM) for medium to high load, and forced PPWM which can be set by the operator. The evaluation kit is rated for automotive applications. This evaluation kit can be used in tandem with the Strata environment GUI to enable/disable the DC-DC converter, change the operating mode, and monitor other telemetry including interrupts, input/output voltage, power dissipated, and temperature.

Features

- **Vin range from 6.0 .. 16.0 V, 40.0 V peak**
  - **Output 1 5.0 V @ 10.0 A peak**
    - **NCV881930 Synchronous Buck Controller + NVMFS5C460NL 40 V Dual N-FET**
  - **Output 2 3.3 V @ 5.0 A peak**
    - **NCV6357MTWDTXG Step Down Converter, AOT, Configurable 5.0 A**
- **NCV6357MTWDTXG Programmable Vout from 0.6V to 3.3V in 12.5mV steps**
- **NCV6357MTWDTXG Adaptive-On-Time (AOT)**
- **NCV6357MTWDTXG Operation at up to 2.4MHz switching frequency**
- **NCV6357MTWDTXG Both PFM and PPWM operation with automatic transition for Optimum Efficiency**
- **NCV6357MTWDTXG 3.0 x 4.0 mm DFN-14 package**
- **Automotive and industrial rated AEC−Q100 Qualified and PPAP Capable**

Applications

- **DC-DC Power**
- **Advanced driver-assistance systems (ADAS).**
- **Automotive POL**
- **Instrumentation**

JSON commands

JSON commands generate and parse JSON-format language objects. "JSON or JavaScript Object Notation is a generic text-based data format derived from object notation of the ECMAScript language." JSON is independent from any other language, but uses conventions that are familiar to programmers using C++ or JavaScript, Perl, Python and so on. It is a format that is particularly suitable for data exchange.

JSON syntax :

JSON syntax is based on the following principles:

- data consists of name/value pairs,
- data is separated by commas,
- objects are defined by braces {},
- arrays are defined by brackets [ ].

JSON objects

JSON objects are defined by braces and can contain an undefined number of name/value pairs, for example:

```
{"cmd":"request_platform_id"}
```

You can copy and paste them into CoolTerm - use text entry line in the bottom, do not write into the window main text area.

Built-In Commands

- Platform Identification Request
- Set Enable 5V
- Set Enable 3V3
- Set VSEL pin
- Program VSelect1 Output value
- Program Enable for Vselect 1 Output Voltage
- Program VSelect0 Output value
- Program Enable for Vselect 0 Output Voltage
- Enable Power Good
- Power Good Active on DVS
- Set Reset Timeout for Power Good
- Enable Active Discharge
- Select Debounce Time
- Select DVS Speed
- Select Delay upon Enabling
- Set VSEL pin Gating
- Select Sleep mode
- Set Dynamic Voltage Scaling (DVS) Mode
- Set Operating Mode for VSEL1
- Set Operating Mode for VSEL0
- Re-Arm the device after TSD/ISHORT
- Reset Indicator
- Force Reset
- Select pre-Warning Thermal Threshold Setting
- Select Inductor Peak Current Settings
- Read Interrupt Sense Register
- Read Initial Status of the pin and register settings

Asynchronous Notifications

- Pause Periodic
- Notify Interrupt
- Reset detected on NCV6357
- Reset Occurred Notification
- Read Voltage and Current
- Read Temperature Sensor
- Read Interrupt Errors

Built-In Commands

Built-in commands are used to reset, configure and control the Platform Controller (PC) itself. These commands are available at any time, whether there is an associated device under test (DUT) identified or not. The built-in commands include:

Causes the PC to return the PC board identifier and revision, and firmware revision (or build timestamp.)

Platform Identification Request
Causes the PC to return the PC board identifier and revision, and firmware revision (or build timestamp.)

```
// HC -> PC
{"cmd":"request_platform_id"}

// After command completes, the platform ID is merged with the firmware revision:
Format: "BU.Year.Serial.Major#,Minor#,FW_Major, FW_Minor, FW Patch, UUID"
```

Set Enable 5V
Set enable pin of the switcher high or low.

```
// HC -> PC
{"cmd":"set_enable_5v","payload":{"enable_5v": "on"}}
// on or off (string) (slide switch)
```

Set Enable 3V3
Set enable pin of the switcher high or low.

```
// HC -> PC
{"cmd":"set_enable_3v3","payload":{"enable_3v3": "on"}}
// on or off (string) (slide switch)
```

Set VSEL pin
Set VSEL pin of the switcher high or low to select between two programmed output voltages.

```
// HC -> PC
{"cmd":"set_vselect","payload":{"vsel": "on"}}
// on or off (string) (slide switch)
```

Program VSelect1 Output value
Program Vselect1 output value from 0 to 216 which represents 600mV to 3300mV.

```
// HC -> PC
{"cmd":"set_prog_vsel1","payload":{"prog_vsel1":0}}
// (int) Value varies from 0 to 216.
// Dropdown value represented is 600mV to 3300mV in steps of 12.5mV
// Use following equation: o/p voltage = (("prog_vsel1" * 12.5) + 600)
```

Program Enable for Vselect 1 Output Voltage
Enable/disable Vselect1 output voltage.

```
// HC -> PC
{"cmd":"set_enable_vsel1","payload":{"enable_vsel1":"on"}}
// (string) Slide switch "on" or "off"
```

Program VSelect0 Output value
Program Vselect0 output value from 0 to 216 which represents 600mV to 3300mV.

```
// HC -> PC
{"cmd":"set_prog_vsel0","payload":{"prog_vsel0":0}}
// (int) Value varies from 0 to 216.
// Dropdown value represented is 600mV to 3300mV in steps of 12.5mV
// Use following equation: o/p voltage = (("prog_vsel0" * 12.5) + 600)
```

Program Enable for Vselect 0 Output Voltage
Enable/disable Vselect0 output voltage.

```
// HC -> PC
{"cmd":"set_enable_vsel0","payload":{"enable_vsel1":"on"}}
// (string) Slide switch "on" or "off"
```

Enable Power Good
Enable/disable pgood signal.

```
// HC -> PC
{"cmd":"set_pgood_enable","payload":{"pgood":"on"}}
// (string) Slide switch "on" or "off"
```

Power Good Active on DVS
Enable/disable pgood signal during dynamic voltage scaling.

```
// HC -> PC
{"cmd":"set_pgood_on_dvs","payload":{"pgood":"on"}}
// (string) Slide switch "on" or "off"
```

Set Reset Timeout for Power Good
Timeout reset setting for pgood.

```
// HC -> PC
{"cmd":"enable_active_discharge","payload":{"active_discharge":"on"}}
// (int) Combobox values 0-3 representing 0ms, 8ms, 32ms and 64ms
```

Enable Active Discharge
Enable/disable active discharge on the output.

```
// HC -> PC
{"cmd":"enable_active_discharge","payload":{"active_discharge":"on"}}
// (string) Slide switch "on" or "off"
```

Select Debounce Time
Set debounce time for enable pin from 0 to 4 representing the values below.

```
// HC -> PC
{"cmd":"set_debounce_time","payload":{"debounce_time":0}}
// (int) index varies from 0 to 3 drop down
//  0: "No Debounce",          1: "1-2 uS"
//  2: "2-3 uS",               3: "3-4 uS"
```

Select DVS Speed
Set dynamic voltage scaling steps from 0 to 3 which represent values below.

```
// HC -> PC
{"cmd":"set_dvs_speed","payload":{"dvs_speed":0}}
// (int) index varies from 0 to 3 drop down
//  0: "6.25mV step / 0.333uS",          1: "6.25mV step / 0.666uS"
//  1: "6.25mV step / 1.333uS",          3: "6.25mV step / 2.666uS"
```

Select Delay upon Enabling
Set delay upon enabling the switcher from values 0 to 7 which represent 0ms to 14ms of delay.

```
// HC -> PC
{"cmd":"set_delay_on_enable","payload":{"delay":0}}
// (int) index varies from 0 to 7 drop down
// 0 mS to 14 mS in steps of 2 mS
```

Set VSEL pin Gating
Select programmable out voltage from Vselect0 and Vselect1. This is same setting as the VSEL pin just the setting is done through i2c.

```
// HC -> PC
{"cmd":"set_vsel_gating","payload":{"vsel_gating":"on"}}
// (string) Slide switch "on" or "off"
```

Select Sleep mode
Enable/disable sleep mode.

```
// HC -> PC
{"cmd":"set_sleep_mode","payload":{"sleep_mode":"on"}}
// (string) Slide switch "on" or "off"
```

Set Dynamic Voltage Scaling (DVS) Mode
Set operation mode between Auto and forced PPWM during dynamic voltage scaling.

```
// HC -> PC
{cmd":"set_dvs_mode","payload":{"dvs_mode":"forced_ppwm"}}
// or "auto" slide switch (string)
```

Set Operating Mode for VSEL1

Set operation mode for Vselect1.

```
// HC -> PC
{"cmd":"set_ppwm_vsel1_mode","payload":{"ppwm_mode":"forced_ppwm"}}
// or "auto" slide switch (string)
```

Set Operating Mode for VSEL0
Set operation mode for Vselect0.

```
// HC -> PC
{"cmd":"set_ppwm_vsel0_mode","payload":{"ppwm_mode":"forced_ppwm"}}
// or "auto" slide switch (string)
```

Re-Arm the device after TSD/ISHORT
Re-arm the device after thermal shut-down fault or short circuit.

```
// HC -> PC
{"cmd":"rearm_device_setting","payload":{ "rearm_device":"on"}}
// or "off" slide switch (string)
```

Reset Indicator
This command changes state of reset status indicator. On power reset it is set default.

```
// HC -> PC
{"cmd":"reset_status_indicator","payload":{"reset_status":"default"}}
// "reset" Red led indicator (string)
```

Force Reset
This setting will reset all the registers of the switcher to default value.

```
// HC -> PC
{"cmd":"force_reset_registers","payload":{ "reset_status":"reset"}}
// "default" button (string)
```

Select pre-Warning Thermal Threshold Setting
Set pre-warning thermal threshold from 0 to 3 representing values below.

```
// HC -> PC
{"cmd":"set_thermal_threshold","payload":{"thermal_threshold":0}}
// (int) index varies from 0 to 3 combobox
// 83°C, 94°C, 105°C, 116°C
```

Select Inductor Peak Current Settings
Set inductor peak current which is a setting for over-current protection threshold.

```
// HC -> PC
{"cmd":"set_ipeak_current","payload":{"ipeak_current":0}}
// (int) index varies from 0 to 3 combobox
// 5.2A(Iout = 3.5A), 5.8A(Iout = 4.0A), 6.2A(Iout = 4.5A), 6.8A(Iout = 5.0A)
```

Read Interrupt Sense Register
Read Interrupt sense register 8 bit value. This is a real time.

```
// HC -> PC
{"cmd":"read_sense_register"}
```

```
// PC -> HC
{"notification":{"value":"status_sense_register","payload":{
"sense_reg_value": 70 }}
// 8 bit value
```

Read Initial Status of the pin and register settings
Read Initial Status of all the pins and notify the host. This command is used to update the UI widgets initial settings. The notification is broken in to 2 parts as the buffer being used is 512 bytes only.

```
// HC -> PC
```

```
{"cmd":"read_initial_status"}
```

Asynchronous Notifications

These notifications can occur at any time and are not associated with any particular command.

Pause Periodic
Set the the pause_flag to true to pause.

```
// HC -> PC
{"cmd":"pause_periodic","payload":{"pause_flag": true}}
```

Notify Interrupt
This notification occurs when pgood pin state changes.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"status_interrupt","payload":{"pgood":"bad" }}
// interrupt indicator LED ("bad" => red otherwise "green")
```

Reset detected on NCV6357
Notify the host if NCV6357 goes under power reset. Reset indicator bit changes.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"power_cycle_status","payload":{"reset":"occurred" }}
// Reset LED indicator turns red otherwise grey
```

Reset Occurred Notification
This notification occurs when MCU has restarted. It is sent out at the start of the program.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"status_mcu_reset","payload":{"mcu_reset": "occurred"}}
```

Read Voltage and Current
This will notification will notify the host with Vin, Vout, Iin, Iout efficiency, output power and power dissipated.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"status_voltage_current","payload":{"vin":12.00,"vout_5v":5.017,"
vout_3v3":3.309,"iin":6.80,"iout_5v":0.50,"iout_3v3":0.73,"efficiency":6,"power_dissipated
":76.60,"output_power":5.00}}}
```

Read Temperature Sensor
Notify board temperature to the host. The temperature is read from temp sensor.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"status_temperature_sensor","payload":{"temperature":25}}
// in Celsius  (-55 to 125 degrees)
```

Read Interrupt Errors
Notify the host with all the errors detected when pgood pin goes low i.e. interrupt occurs.

```
// PC → HC   (Periodic Notification)
{"notification":{"value":"status_ack_register","payload":{events_detected":[
"power_good_event_detected","dcdc_over_current_event_detected","under_voltage
_event_detected","thermal_prewarning_event_detected","thermal_shutdown_event_
detected"]}}}
```

One-shot Build-in Commands

These commands are common to all platforms and should not be referenced in children pages.

// PC → HC (request_platform_id and set_platform_id)

```
{ "cmd" : "request_platform_id" }
{ "cmd" : "set_platform_id", "payload" : { "verbose_name":"<string>",
"platform_id": "<string>" }}
```


PUBLICATION ORDERING INFORMATION

LITERATURE FULLFILLMENT:
Literature Distribution Center for ON Semiconductor 19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA

Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada

N. American Technical Support:
800−282−9855 Toll Free USA/Canada

Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910

ON Semiconductor Website:
https://www.onsemi.com/

Order Literature:
https://www.onsemi.com/orderlit

Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

For additional information, please contact your local Sales Representative

PUBLICATION ORDERING INFORMATION

AUTOMOTIVE ADAS PREREGULATOR STR-ADAS-PREREGULATOR-GEVK

LITERATURE FULLFILLMENT:
Literature Distribution Center for ON Semiconductor 19521 E. 32nd Pkwy, Aurora, Colorado 80011 USA

Phone: 303−675−2175 or 800−344−3860 Toll Free USA/Canada
Fax: 303−675−2176 or 800−344−3867 Toll Free USA/Canada
Email: orderlit@onsemi.com

N. American Technical Support:
800−282−9855 Toll Free USA/Canada

Europe, Middle East and Africa Technical Support:
Phone: 421 33 790 2910

ON Semiconductor Website:
https://www.onsemi.com/

Order Literature:
https://www.onsemi.com/orderlit

For additional information, please contact your local Sales Representative