# Voltage Sequencer Datasheet VoltageSequencer V 1.00



| MUM | PSoC® Blocks | | | API Memory (Bytes) | | Pins |
|---|---|---|---|---|---|---|
| | Digital | Analog CT | Analog SC | Flash | RAM | |
| **Supported devices:** CY8C21x23, CY8C21x34, CY8C21x45, CY8C22x45, CY8C23x33, CY8C24x33, CY8C24x23A, CY8C27x43, CY8C24x94, CY8C29x66, CY8C28xxx | | | | | | |
| Configuration | 2 | 0 | 0 | 1100 | 20 | 1-2/rail |

## Features and Overview

- Directly interfaces with regulators that have a logic-level enable pin and integrates a voltage output monitoring circuitry that provides a "power good (PGOOD)" or similar logic-level status output pin
- Four interrupts for Power-up Complete, Power-down Complete, Rail Fail, and Power- up Fail
- The maximum number of rails supported depends on the number of pins available and whether or not PGOOD monitoring is required
- Provides auto mode enable option
- Comprises a wizard to configure voltage sequencing-related parameters
- The Sequencer tab in the wizard provides a graphical interface for better understanding of the sequencing logic

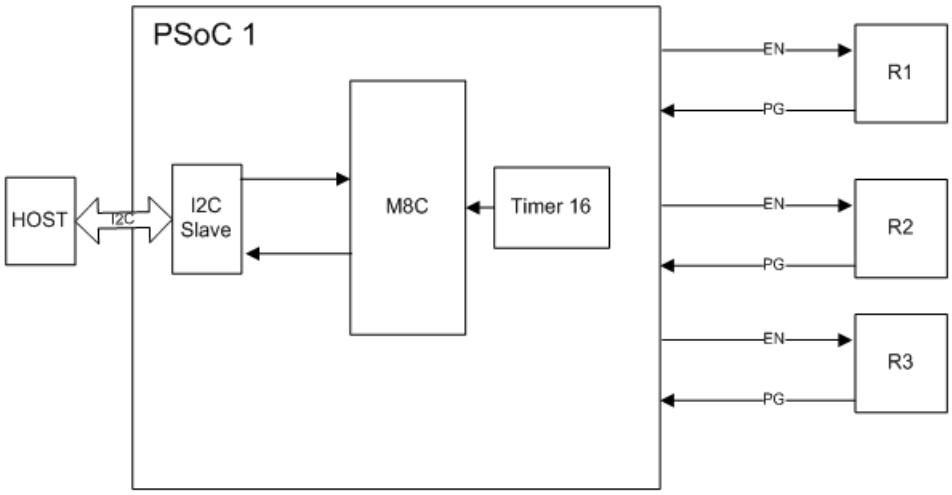
The Voltage Sequencer User Module is a one-stop solution to implement a voltage sequencing solution using any of the PSoC device families. The number of rails needed to support sequencing and features may vary from one family to another. Voltage sequencing takes care of sequencing of voltage rails in a system to ensure proper power up and power down sequences and flag when a fault is detected.

## Quick Start

1. Select and place the Voltage Sequencer User Module from the Power Management category of the user module catalog.
2. The user module is accompanied by a wizard that helps to configure parameters related to voltage sequencing (the wizard is explained later in this user module datasheet).
3. Enter all the appropriate values in the wizard to get the required voltage sequencing configuration.
4. Pin assignment is done with the help of the wizard that has a dynamic pin assignment wizard that displays the available graphics of the part with available pins and allowed pins for placement.
5. Generate the application and switch to the Application Editor.
6. Adapt the sample code, as required.
7. Program the PSoC on the target board with the hex generated by PSoC Designer™.

## Functional Description

Figure 1. VoltageSequencer Block Diagram




This IP Block sequences the regulators by providing the EN signals and reading back the PGOOD signal from the regulator, if the regulators give a signal. For example, if the PGOOD signal is received for regulator 1, then it proceeds to regulator 2 after the given 'Up delay'. If the device does not receive the PGOOD signal in a specific 'RAMP' time, then the device asserts a power-up failure interrupt and continues with the power-down sequence as configured, if the auto power-down mode is enabled.

## DC and AC Electrical Characteristics

Table 1. Voltage Sequencer AC Electrical Characteristics

| Parameter | Min | Typical | Max | Units | Conditions and Notes |
|---|---|---|---|---|---|
| EN Output Pin $V_{OH}/V_{OL}$ | | | Based on device datasheet | | At different $V_{DD}$ |
| PGOOD Input Pin $V_{IL}/V_{IH}$ | | | Based on device datasheet | | At different $V_{DD}$ |
| Timer Accuracy (1 ms) | | | ±4/5% | | Based on the device IMO variation (2.5%/4%) |

## Placement

Voltage Sequencer is a Multi User Module. The blocks for the user module are automatically placed when the user module is instantiated and user selections are entered; alternate placements are available only in certain configurations. Only one instance of the user module can be placed in the project.

Follow this step to place and configure this user module.

1. Select and place the Voltage Sequencer User Module; the Voltage Sequencer wizard pops up. The Wizard helps you to choose the UM configuration (there are many configurations of the Voltage Sequencer UM, depending on the selected device).

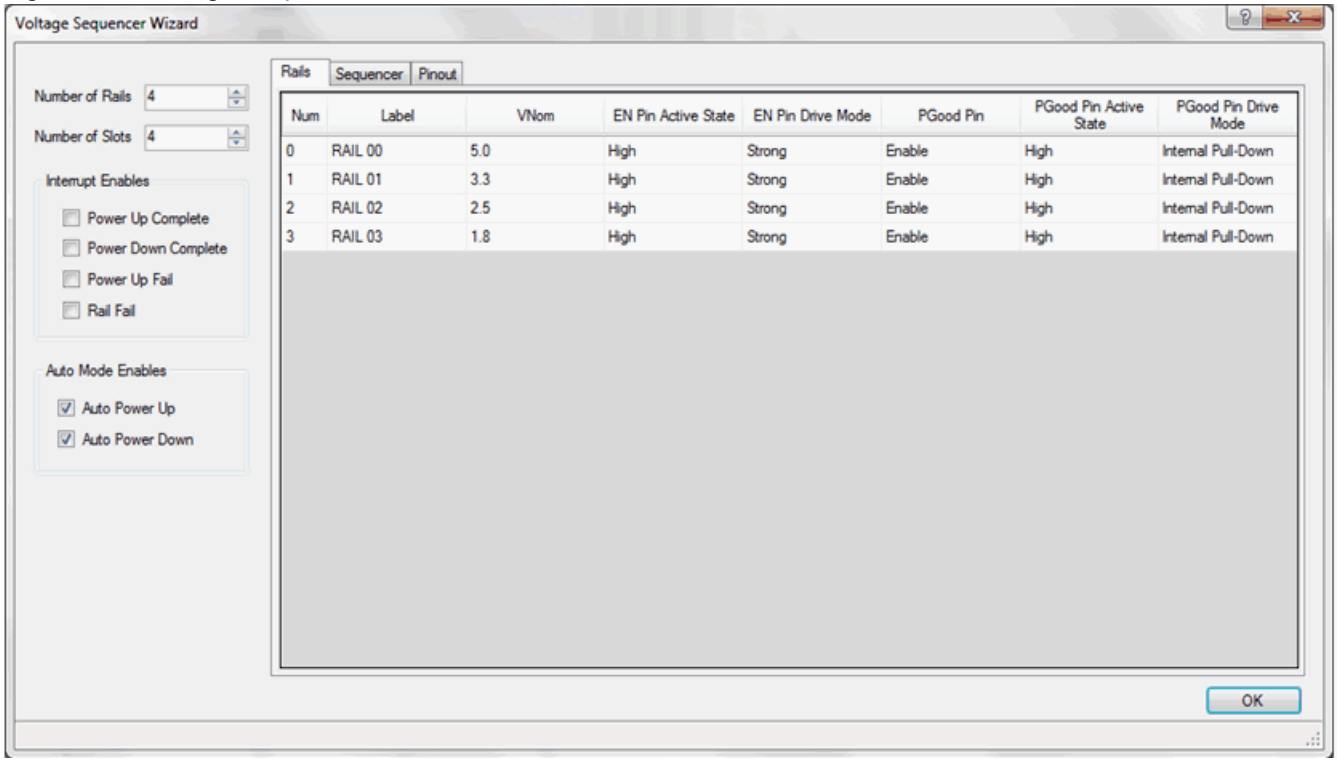
## Voltage Sequencer Wizard

The wizard has three tabs (Rails, Sequencer, and Pins). None of the resources are highlighted because the user module does not know the resource requirement until the Wizard parameters are configured. (Default – Rails tab selected).

There are four general settings on the left side of Wizard:

- Number of rails
- Number of slots
- Interrupt Enables
- Auto Mode Enables

Figure 2. Voltage Sequencer Wizard




**Number of Rails**

This parameter is used to specify the number of regulators that this user module will sequence. The valid range for this parameter is based on the part number selected. The default setting is 4. The max is the count of available GPIO pins divided by 2 if you choose PGOOD in MonType in the Monitor tab. The max value is the count of available GPIO pins divided by 1 if you choose None in MonType in Monitor Tab. The min value is 2.

**Number of Slots**

This parameter is used to specify the number of distinct sequencing time slots required to power-on all the regulators in the system. One or more voltage regulators can be assigned to each sequencer timing slot. Therefore, the Number of Slots setting cannot exceed the Number of Rails setting. Doing so results in a sequencing timing slot that has no voltage regulator Enables associated with it. The valid range for this parameter is based on the part number selected. The default setting is 4. The assignment of rails to the slot is done in the Sequencer tab. The max value is Number of Rails, while the min value is 2.

### Interrupt Enables

This parameter lists the available interrupts and you can select the interrupts required for your design.

- Power up Complete – This interrupt is flagged when the power-up sequence is completed successfully.
- Power Down Complete – This interrupt is flagged when the power-down sequence is completed successfully.
- Power Up Fail – This interrupt is flagged when the power-up sequence fails.
- Rail Fail – This interrupt is flagged when a particular rail fails after the power-up sequence is complete.

### Auto Mode Enables

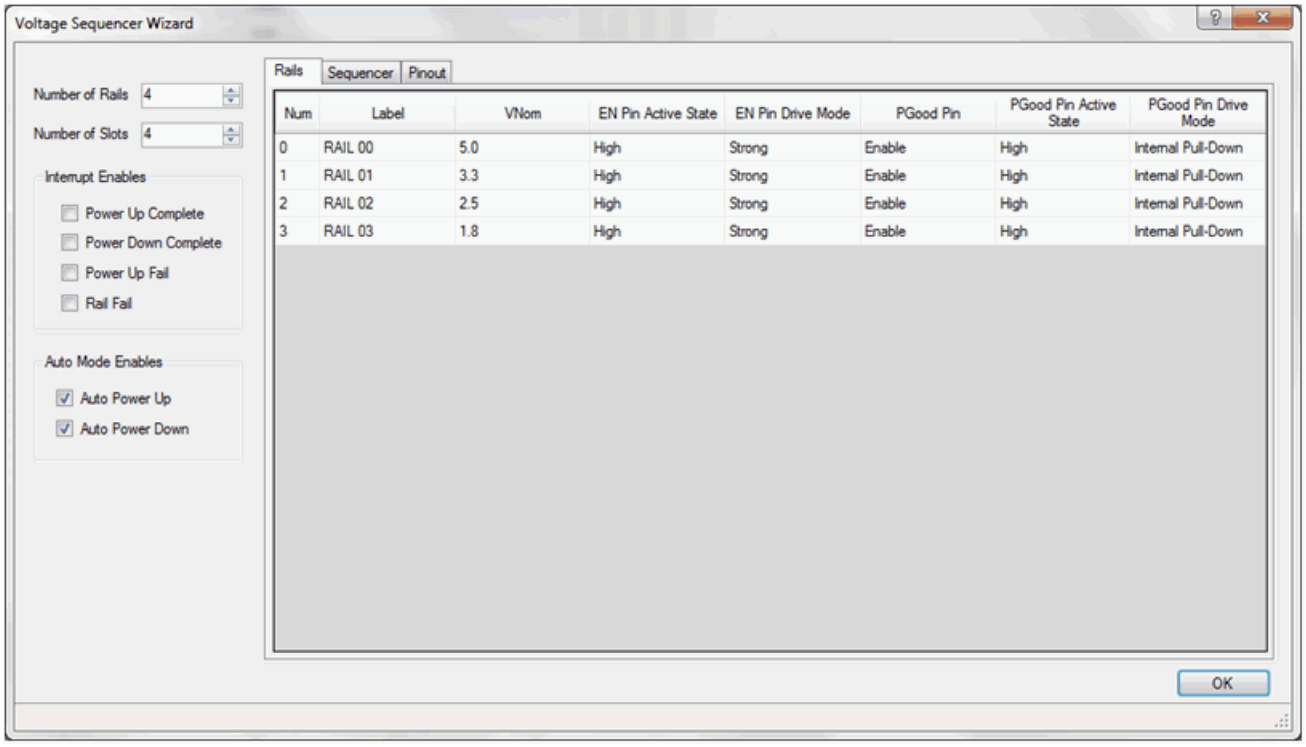
This parameter lists the auto power-up and auto power-down options.

- Auto Power Up – When it is enabled, the voltage rails are sequenced as configured when the start API is called. If it is not checked, then the sequencing is manually handled by the host using the APIs.
- Auto Power Down – When it is enabled, the voltage rails are powered down if there is an error condition as configured in the power-down tab. If it is not checked, then the power-down is manually handled by the host using the APIs.

**Note** When "Auto Power Down" is disabled, any error condition that occurs during powering up the rails results in the execution of Power Up Fail ISR (If enabled) and the rails will not be powered down automatically. Even though the user module detects the error conditions, it will continue to power up the rails because the "Auto Power Down" is disabled. Therefore, enabling the "Auto Power Down" setting is always recommended. This issue will be fixed in the next release of the user module.

## Rails Tab

Figure 3. Voltage Sequencer Wizard - Rails Tab



The number of rows created is based on the number of rails, with each row for a rail having the following parameters:

**Vnom**

These Up/Down boxes with edit capability and two decimal points have no impact on the user module operation. They are simply made available for annotation purposes to allow you to enter the nominal voltages (in VDC) of each regulator rail. These annotation names are carried forward into other tabs of the user module customizer for display and cross-referencing purposes. Default entries are: 5.0 V, 3.3 V, 2.5 V, and 1.8 V for the first four voltage rails and 3.3 V for all subsequent rails. The range is 00.00 to 99.99.

**EN Pin Active State**

This parameter sets the active logic level of the selected regulator enable (EN) signals. The options are High and Low.

**EN Pin Drive Mode**

This parameter sets the drive mode of the regulator enable (EN) pins for each regulator. Valid options are:

- Strong
- Pull Up
- Pull Down
- Open Drain Low
- Open Drain High

The default setting is Strong.

**Monitor Type**

There are two monitoring options:

- PGood
- None

Selecting the PGood option enables monitoring of the selected PGood pin through the power-up sequencing operation and also enables continuous real time monitoring of that regulator after power-up sequencing completes.

However, selecting None as the option disables monitoring of the selected PGood signal throughout the power-up sequencing operation and also disables continuous real time monitoring of that regulator after power-up sequencing completes.

**PGood Pin Drive Mode**

This parameter sets the drive mode of the regulator power good (PGood) status signals for each regulator. The valid options are:

- High Z
- Pull Up
- Pull Down

**PGood Pin Active State**

This parameter sets the logic level polarity of the regulator power good (PGood) status signals. The valid options are High or Low for PGood Pin Drive Mode. Default setting is "High Z". For the Pull Up Drive mode, the Active State is only "Low", and for "Pull Down", it is "High".

**Note** If the PGood Pin setting is "Disable", the PGood Pin Active State and the PGood Pin Drive Mode controls are inactive and grayed.

## Sequencer Tab

This tab is used to define the sequencing order and timing of the voltage sequencer. The Sequencer tab provides a graphical interface for better understanding of the sequencing logic.

**Power Down**

This parameter controls the regulator power-down sequence and can be configured in one of the following three ways using the drop-down menu:

- Forward: same order as the power-up sequence
- Reverse: reverse order of the power-up sequence
- All off: turns off all regulators simultaneously and immediately (without any power down delays)

**Note** If the Power Down setting is "All Off", the general power down delay time is zero and the DnDelay controls are inactive and grayed.

### Sequencer Timing Chart

To further assist with the process of setting up the sequencer, a graphical sequencer timing chart is displayed above the assignment matrix. This chart dynamically refreshes based on the parameter settings to show the currently defined sequencing order and timing.

Figure 4. VoltageSequencer Wizard – Sequencer Tab in Overlay Mode

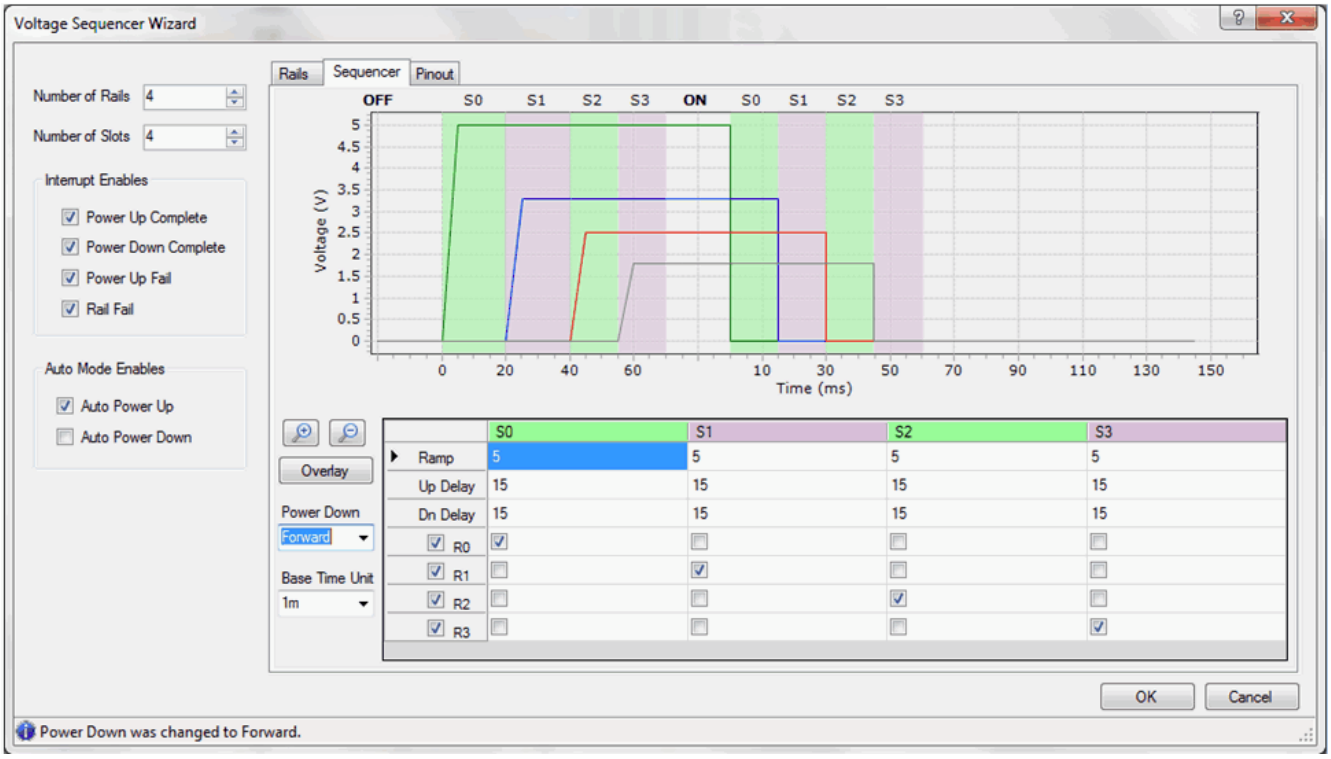
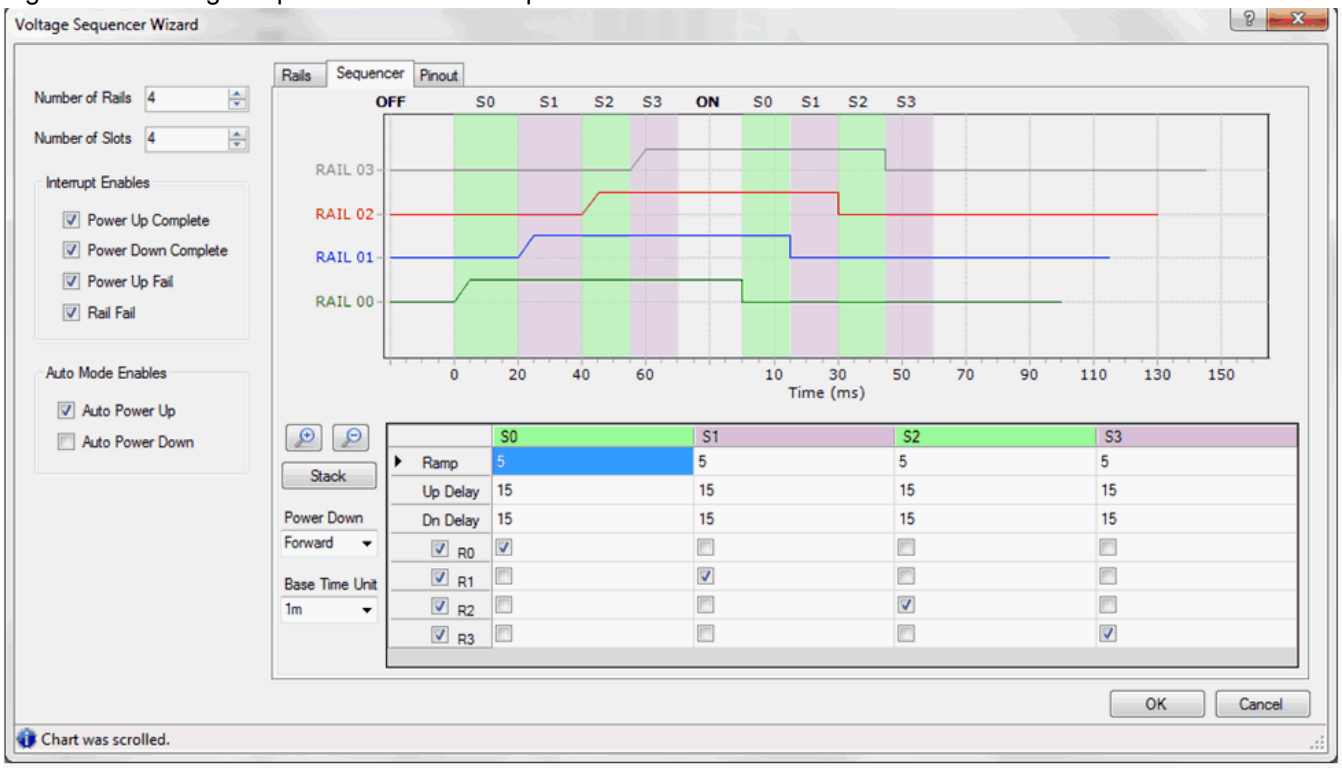



Figure 4 shows the chart configured in the default Overlay mode. This mode is designed to resemble an analog oscilloscope view of the sequenced voltage regulators. The vertical axis represents DC voltage. The horizontal axis shows the time base of the sequencer and indicates each of the sequencing slots in alternate blue and green colors. The left half of the chart shows the power-up sequence configuration, while the right half shows the power-down sequence configuration.

The "+" and "-" buttons enable time zooming of the waveform. Horizontal scrolling is also possible. The min is 1 ms time scale resolution because that is the base time scale. You cannot zoom in beyond that. The max scale is in 1 second. Every zoom step resizes the image approximately twice. For example,

when the time scale is 1 ms, if '+' is pressed, then scale changes first to 2 ms, then 5 ms, and then 10 ms, and so on.

Figure 5. VoltageSequencer Wizard – Sequencer Tab in Stack Mode




The stack mode is designed to resemble a logic analyzer view of the sequenced voltage regulators. The vertical axis represents when each rail is on (high) or off (low). The horizontal axis is the same as in the Overlay mode.

You can choose which regulator rails to display in the Timing Chart Window using the check boxes that appear directly near Rail number in the Rail Assign check box matrix.

**Time Base Unit**

This setting defines a specific time base unit. All sequencer timings are multiples of this value.The Time Base Unit value range is from 200 µs to 2 ms (200 µs, 500 µs, 1 ms, and 2 ms).

**Ramp**

This parameter enables you to enter the worst case expected ramp time of all the regulators assigned to that sequencing time slot, defined as the time between the regulator enable control pin (EN) being asserted and the regulator status pin (PGOOD) becoming valid.

Operationally, at the start of each new sequencing time slot, the sequencer enables the selected regulators, wait for the low-to-high or high-to-low transition (based on the PGOOD Pin Active State) in the PGOOD signals of the selected regulators until the time period specified by the Ramp parameter for that slot. If the PGOOD is received for the regulators within the Ramp time, then it goes to the Up delay explained in the next section and does not wait until the ramp time to move to the next slot. You should, therefore, enter a Ramp parameter that exceeds the actual expected ramp time of the regulators with some margin to guarantee that at the time of sampling, the "PGOOD" status pins have all stabilized and are valid. If "PGOOD" sampling returns a failure at the end of the ramp time, power-up sequencing halts. If the Auto Power Down parameter is checked, a power-down sequence is automatically initiated. If that parameter is not checked, application firmware is responsible for taking

appropriate corrective action. If Power Up Fail or Rail Fail interrupt enables are checked, an interrupt is generated at this time.

The actual ramp time is determined by the Ramp parameter multiplied by the Time Base Unit value. The ramp value is 0…255.

**Up Delay**

This parameter enables you to configure the duration of a functional sequencing time slot during power-up sequencing. This duration begins after successful PGOOD sampling. The sequencer remains in the current sequencing time slot until the Up Delay time expires. When the Up Delay time expires, the sequencer can move to the next time slot in the sequence.

The actual delay time is determined by the Up Delay parameter multiplied by the Time Base Unit value. The Up Delay value is 0…255.

**Dn Delay**

This parameter enables you to configure the duration of a functional sequencing time slot during power-down sequencing. This duration begins after the regulators assigned to the current time slot are disabled. The sequencer remains in the current sequencing time slot until the Dn Delay time expires. When it does, the sequencer can move to the next time slot in the sequence. This is not applicable for the "All-off" power-down sequence. The actual delay time is determined by the Dn Delay parameter multiplied by the time base unit value. The Dn Delay value is 0...255.

**Rail Assign**

Assigning voltage regulators to sequencing time slots is accomplished through a matrix of check boxes to simplify configuration and visualization of the sequence. The rows in the matrix represent voltage regulators. Regulator 0 is the uppermost row. The columns in the matrix represent sequencing time slots. Slot 0 is the extreme left column.

The rules for the assignment are as follows:

- Each regulator can be assigned to only one slot
- A slot must have at least one regulator assigned to it but may have many (up to n, based on the part number selected) assigned to it

After assigning a regulator to a slot, other check boxes in the column should be disabled. This ensures that you are not able to assign the same regulator to different slots.

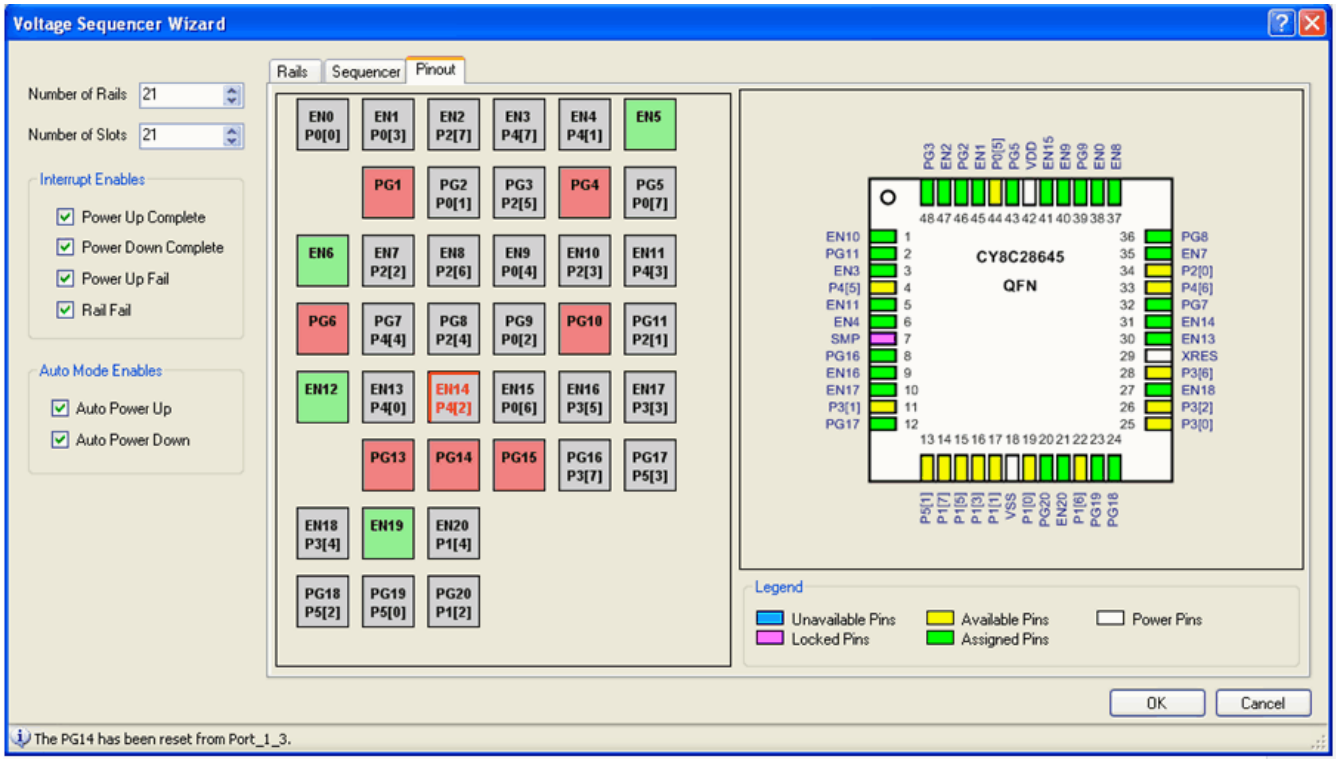
The customizer provides visual aids to indicate whether or not the user's assignment meets these rules. If a slot label (S0…Sn) at the top of each column is highlighted by a yellow background, it indicates that the slot does not have a regulator rail assigned to it. If a rail label (0…m) at the left of each row is highlighted by a yellow background, it indicates that the regulator is not assigned to a slot. The default setting is 1 rail assigned for each slot.

## Pins Tab

The user module parameters in Pins Tab control the pin assignment of the Voltage Sequencer User Module.

For the selected part number, the Pin assignment wizard displays the available graphics of the part with available pins and allowed pins for a particular function. There is an easy drag-and-drop mechanism to configure the EN and PG pins. Follow these steps to configure this wizard:

1. Left-click an EN or PG pin and drag it to any available pin. The port pin is green after selection and is no longer available. Change the Pin assignments by dragging the sensor of the port pin.
2. Repeat this procedure for the remaining EN and PG pins.
3. Click **OK** to complete




## Application Programming Interface

The Application Programming Interface (API) routines are provided as part of the user module to allow the designer to deal with the module at a higher level. This section specifies the interface to each function together with related constants provided by the “include" files.

Each time a user module is placed, it is assigned an instance name. By default, PSoC Designer assigns the VoltageSequencer_1 to the first instance of this user module in a given project. This name can be changed to any unique value that follows the syntactic rules for identifiers. The assigned instance name becomes the prefix of every global function name, variable and constant symbol. In the following descriptions the instance name has been shortened to VoltageSequencer for simplicity.

**Note**

In this, as in all user module APIs, the values of the A and X register may be altered by calling an API function. It is the responsibility of the calling function to preserve the values of A and X prior to the call if those values are required after the call. This “registers are volatile" policy was selected for efficiency reasons and has been in force since version 1.0 of PSoC Designer. The C compiler automatically takes care of this requirement. Assembly language programmers must ensure their code observes the policy, too. Though some user module API function may leave A and X unchanged, there is no guarantee they will do so in the future.

API routines are provided as part of the user module to allow the designer to deal with the module at a higher level. The following are the API programming routines provided for Voltage Sequencer.

## Global Variables

- **VoltageSequencer_baRailState** – The size depends on the number of rails (1 byte for 2..8 rails, 2 bytes for 9..16 rails, and so on up to 8 bytes for 64 rails). Each bit indicates the state of the certain rail (1 – enabled, 0 – disabled).
- **VoltageSequencer_baRailStatus** – The size depends on the number of rails (1 byte for 2..8 rails, 2 bytes for 9..16 rails, and so on up to 8 bytes for 64 rails). Each bit indicates the Fail status of the certain rail (1 – failed, 0 – good).
- **BYTE VoltageSequencer bSlot** – Pointer to the current slot during Power Up/Power Down sequences. It is also used as a Slot/Rail pointer for bSeqSlotUp/SeqSlotDown and EnableRail/DisableRail API functions.
- **BYTE VoltageSequencer bState** – Internal current sequencer state and also the interrupt enables.
- **VoltageSequencer _bTime** – Internal counter for sequencer timing maintenance.
- **VoltageSequencer _bFlag** – Internal flags for the sequencer state machine.

## VoltageSequencer_Start

**Description:**

Starts the user module and initiates a power-up sequence if the Auto Power Up parameter is enabled.

**C Prototype:**

```
void VoltageSequencer_Start (void);
```

**Assembly:**

```
lcall VoltageSequencer_Start
```

**Parameters:**

None

**Return Value:**

None

## VoltageSequencer_Stop

**Description:**

Stops the user module preventing real-time PGOOD monitoring and sequencing operations.

**C Prototype:**

```
void VoltageSequencer_Stop(void)
```

**Assembly:**

```
lcall VoltageSequencer_Stop
```

**Parameters:**

None

**Return Value:**

None

## VoltageSequencer _SetIntMode

**Description:**

Used to configure interrupt sources from the user module as specified by the IntMode parameter.

**C Prototype:**

```
void  VoltageSequencer_SetIntMode (BYTE bIntMode);
```

**Assembly:**

```
mov A, bIntMode
lcall VoltageSequencer_SetIntMode
```

**Parameters:**

bIntMode: bit map to enable Interrupt sources

| Flag Name Definition | Mask | Description |
|---|---|---|
| VoltageSequencer_POWER_UP_COMPLETE | 0x04 | All Rails/Slots Powered Up |
| VoltageSequencer_POWER_DOWN_COMPLETE | 0x08 | All Rails/Slots Powered Down |
| VoltageSequencer_POWER_UP_FAIL | 0x10 | Power Up Sequence in Progress |
| VoltageSequencer_RAIL_FAIL | 0x20 | Power Down Sequence in Progress |
| – | 0xC3 | Not significant |

**Return Value**

None

## VoltageSequencer _bGetIntMode

**Description:**

Configures interrupt sources from the user module as specified by the IntMode parameter.

**C Prototype:**

```
BYTE VoltageSequencer_bGetIntMode (void);
```

**Assembly:**

```
lcall VoltageSequencer_bGetIntMode
```

**Parameters:**

None

**Return Value:**

A bit map with flagged Interrupt sources:

| Flag Name Definition | Mask | Description |
|---|---|---|
| VoltageSequencer_POWER_UP_COMPLETE | 0x04 | All Rails/Slots Powered Up |
| VoltageSequencer_POWER_DOWN_COMPLETE | 0x08 | All Rails/Slots Powered Down |
| VoltageSequencer_POWER_UP_FAIL | 0x10 | Power Up Sequence in Progress |
| VoltageSequencer_RAIL_FAIL | 0x20 | Power Down Sequence in Progress |
| – | 0xC3 | Not significant |

## VoltageSequencer _EnableRail

**Description:**

Manually enables one selected regulator.

**C Prototype:**

```
void VoltageSequencer_ EnableRail (BYTE bRailNum);
```

**Assembly:**

```
mov A, bRailNum
lcall VoltageSequencer_ EnableRail
```

**Parameters:**

bRailNum: Number of rails desired to enable. Valid range is from 0 to the "Number of Rails – 1".

**Return Value:**

None

## VoltageSequencer _DisableRail

**Description:**

Manually disables one selected regulator.

**C Prototype:**

```
void VoltageSequencer_ DisableRail (BYTE bRailNum);
```

**Assembly:**

```
mov A, bRailNum
lcall VoltageSequencer_ DisableRail
```

**Parameters:**

bRailNum: Number of rails desired to disable. Valid range is from 0 to the "Number of Rails – 1".

**Return Value:**

None

## VoltageSequencer _ InitiatePowerDnSeq

**Description:**

Manually initiates a power-down voltage sequence. The power-down sequence is controlled by the PowerDnMode parameter and can be configured in one of these three methods: 1) “Forward", the

same order as the power-up sequence, 2) "Reverse" = reverse order to the power-up sequence, or 3) "All Off" = turn off all regulators simultaneously.

**C Prototype:**

```
void VoltageSequencer_ InitiatePowerDnSeq (void);
```

**Assembly:**

```
lcall VoltageSequencer_ InitiatePowerDnSeq
```

**Parameters:**

None

**Return Value:**

None

## VoltageSequencer _ InitiatePowerUpSeq

**Description:**

Manually initiates a power-up voltage sequence. If the Auto Power Down parameter is checked, then any PGOOD monitoring failures during sequencing results in an automatic power-down sequence.

**C Prototype:**

```
void VoltageSequencer_ InitiatePowerUpSeq (void);
```

**Assembly:**

```
lcall VoltageSequencer_ InitiatePowerUpSeq
```

**Parameters:**

None

**Return Value:**

None

## VoltageSequencer _ bSeqSlotUp

**Description:**

Manually powers up the selected sequencer slot. If the slot power-up sequence is successful, this API returns after the Ramp time and Up Delay time for this slot are completed. If the power-up sequence fails, this API returns after the Ramp time completes. If the Auto Power Down parameter is checked, then any PGOOD monitoring failures during sequencing result in an automatic power-down sequence.

**C Prototype:**

```
BYTE VoltageSequencer_bSeqSlotUp (BYTE bSlotNumber);
```

**Assembly:**

```
mov   A, bSlotNumber
lcall VoltageSequencer_bSeqSlotUp
```

**Parameters:**

BYTE bSlotNumber: Number of Slots desired to power up. Valid range is from 0 to the "Number of Slots – 1".

**Return Value:**

Status Value:

| Value | Description |
|---|---|
| 0x00 | Fail |
| 0x01 | Pass |

## VoltageSequencer _ SeqSlotDown

**Description:**

Manually powers down the selected sequencer slot. This API returns after the Dn Delay time for this slot completes.

Notes:

1. If the Power Down parameter is set to "All Off", then the there is no DnDelay time associated with powering down this slot.
2. There is no Delay time associated with powering down the final slot in the sequence independent of the Power Down parameter setting.

**C Prototype:**

```
VoltageSequencer_SeqSlotDown(BYTE bslotNumber);
```

**Assembly:**

```
mov   A, bSlotNumber
lcall VoltageSequencer_ SeqSlotDown
```

**Parameters:**

BYTE bSlotNumber: Number of Slots desired to power down. Valid range is from 0 to the "Number of Slots – 1".

**Return Value:**

None

## VoltageSequencer _bGetSequencerState

**Description:**

Returns the current state of the Sequencer

**C Prototype:**

```
BYTE VoltageSequencer_bGetSequencerState (void);
```

**Assembly:**

```
lcall VoltageSequencer_bGetSequencerState
```

**Parameters:**

None

**Return Value:**

State value

| Value Name Definition | Value | Description |
|---|---|---|
| VoltageSequencer_ALL_POWERED_DOWN | 0x00 | All Rails/Slots Powered Down |
| VoltageSequencer_ALL_POWERED_UP | 0x01 | All Rails/Slots Powered Up |
| VoltageSequencer_POWER_UP_PROGRESS | 0x02 | Power Up Sequence in Progress |
| VoltageSequencer_POWER_DOWN_PROGRESS | 0x03 | Power Down Sequence in Progress |

## VoltageSequencer _bGetRailState

**Description:**

Returns the state and fail status of the Rail.

**C Prototype:**

```
BYTE VoltageSequencer_bGetRailState (BYTE bRailNum);
```

**Assembly:**

```
mov   A, bRailNum
lcall VoltageSequencer_bGetRailState
```

**Parameters:**

bRailNum: Number of Rails desired to get its state. The valid range is from 0 to the "Number of Rails – 1".

**Return Value:**

State value:

| Value Name Definition | Value | Description |
|---|---|---|
| VoltageSequencer_RAIL_OFF | 0x00 | Rail is turned off |
| VoltageSequencer_RAIL_ON | 0x01 | Rail is turned on, Status is OK (PGOOD pin is active)Up |
| VoltageSequencer_RAIL_FAIL | 0x20 | Rail is failed (tried to turn on but PGOOD pin is inactive) |

## Sample Firmware Source Code

The following is the C Sample Code.

```
//-------------------------------------------------------------------------
// VoltageSequencer Sample Code
//-------------------------------------------------------------------------

#include <m8c.h>
#include "PSoCAPI.h"

BYTE bState, bSlotState, bRailState;

void main(void)
{

    VoltageSequencer_Start();

    M8C_EnableGInt;

    while(1)
    {
        bState = VoltageSequencer_bGetSequencerState();
        if(bState == VoltageSequencer_ALL_POWERED_UP)
        {

            VoltageSequencer_SeqSlotDown(1);
            bSlotState = VoltageSequencer_bSeqSlotUp(1);

            bRailState = VoltageSequencer_bGetRailState(1);
            VoltageSequencer_DisableRail(1);
            bRailState = VoltageSequencer_bGetRailState(1);
            VoltageSequencer_EnableRail(1);

            VoltageSequencer_DisableRail(2);
            VoltageSequencer_EnableRail(2);

            VoltageSequencer_InitiatePowerDnSeq();
        }

        if(bState == VoltageSequencer_ALL_POWERED_DOWN)
        {

            VoltageSequencer_InitiatePowerUpSeq();
        }
    }
}
```

The following is the ASM sample code:

```
;-----------------------------------------------------------------
; VoltageSequencer Assembly Sample Code
;-----------------------------------------------------------------

include "m8c.inc"
include "memory.inc"
include "PSoCAPI.inc"
```

```
export _main
export bState
export bRailState
export bSlotState

area globalvariables (rel,con,ram)

bState:     BLK 1
bRailState: BLK 1
bSlotState: BLK 1

area text (rel,con,rom,code)

_main:
    lcall VoltageSequencer_Start
    M8C_EnableGInt

.mainloop:
    lcall VoltageSequencer_bGetSequencerState
    RAM_SETPAGE_CUR >bState
    mov   [bState], A
    cmp   A, VoltageSequencer_ALL_POWERED_UP
    jnz   .checkPD
    lcall VoltageSequencer_SeqSlotDown
    mov   [bSlotState], A
    mov   A, 1
    lcall VoltageSequencer_bSeqSlotUp
    mov   [bSlotState], A
    mov   A, 1
    lcall VoltageSequencer_bGetRailState
    mov   [bRailState], A
    mov   A, 1
    lcall VoltageSequencer_DisableRail
    mov   A, 1
    lcall VoltageSequencer_bGetRailState
    mov   [bRailState], A
    mov   A, 1
    lcall VoltageSequencer_EnableRail
    mov   A, 2
    lcall VoltageSequencer_DisableRail
    mov   A, 2
    lcall VoltageSequencer_EnableRail
    lcall VoltageSequencer_InitiatePowerDnSeq

.checkPD:
    cmp   [bState], VoltageSequencer_ALL_POWERED_DOWN
    jnz   .mainloop
    lcall VoltageSequencer_InitiatePowerUpSeq
    jmp   .mainloop
```

# Version History

| Version | Originator | Description |
|---|---|---|
| 1.00 | KUK | Initial version. |

**Note** PSoC Designer 5.1 introduces a Version History in all user module datasheets. This section documents high level descriptions of the differences between the current and previous user module versions.