nPZero power-saving IC

nPZero G1SY

Errata

Document ID: NPZ-G1SY-ERRATA-0001

Revision 2.00 – 2025.06.25

European Innovation Council

Co-funded by the European Union

nanopowersemi.com

Revision History

Revision	Date	Description
1.00	2025.04.22	First release
2.00	2025.06.25	Update for Y revision: Item #1 removed

European Innovation Council

Co-funded by the European Union

Funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EISMEA. Neither the European Union nor the granting authority can be held responsible for them.

Contents

1 Active issues

The following issues are present in the nPZero G1Y.

ID	Criticality	Description	Inherited from earlier revision	Workaround available	Fix planned
2	Medium	Incremental SRAM writing can result in corrupted data	Yes	X	
3	Medium	Short interrupt pulses cause wake-up but flag is 0	Yes		
4	Medium	VALP* might retain previously read value	Yes		
5	Low	Reset source invalid	Yes	X	
6	Low	SPI shifts out 2 bytes for 8-bit data type in modes 1 and 3	Yes		
7	Low	Device reads out 2 bytes over I2C when using 8-bit data type	Yes		
8	Medium	Power switch output voltage detection can cause lock-up	Yes		
9	Low	False ADC wake-up trigger when ADC_IN is below 600mV	Yes		

1.1 #2: Incremental SRAM writing can result in corrupted data

Symptoms	There is an issue when writing to sequential SRAM addresses, which can lead to corrupted SRAM data.
Conditions	Writing to multiple SRAM locations sequentially.
Consequences	Data on SRAM can be corrupted after writing (unless mitigated using the below write algorithm), partial SRAM writes and I2C burst writes can also trigger this issue.

Workaround	It is recommended to only write the SRAM through the official nPZero API, as the API will implement the necessary workarounds to avoid this issue.
	Alternatively, it is possible to work around the issue by changing the order at which the SRAM addresses are written.
	The below code snippet shows a way to implement this, when writing the entire content of the SRAM:
	`for (i = 0; i < 128 ; i++) {`
	`  k = (((~i)&0x07)<<3) + ((i&0x38) >> 3) + (i & 0x40);`
	`  np0_i2c_write(SRAM_ADDR + k, myData[k]);`
	`}`

1.2 #3: Short interrupt pulses cause wake-up but flag is 0

Symptoms	A wake-up pulse from a connected peripheral will cause the nPZero to wake up, but the corresponding peripheral trigger status bit will not be set.
Conditions	A peripheral uses interrupt pulses shorter than 2.5us.
Consequences	The nPZero will wake up as expected, but the FPx bit in the STA2 register will not be set.
Workaround	a) If possible, have the peripheral use a longer wake-up pulse b) If only one peripheral uses short interrupt pulses it is possible to surmise that a wake-up without any wake-up trigger listed in STA1 or STA2 must be caused by that specific peripheral

1.3 #4: VALP* might retain previously read value

Symptoms	The VALP* registers might contain incorrect data in some situations, potentially leading to false wake-up events.
Conditions	May happen when *all* of the below conditions are met: 1) More than one peripheral enabled 2) One or more peripherals used in polling mode 0, 1 or 2 3) A NAK occurring when attempting to read the sensor (mode 0 or 1), or a timeout occurring when waiting for an interrupt (mode 1 or 2)
Consequences	If a NAK happens on the I2C bus when attempting to read a sensor, or a timeout happens when waiting for an interrupt, the peripheral value registers will retain the value from the previously read sensor. Potentially this could lead to a false wake-up, if the value hits the trigger area of the affected peripheral. For peripherals using polling mode 2 the issue will have less impact, as the VALP* value is not used for triggering a wake-up.
Workaround	There is no known workaround that will fully correct the issue, but some guidelines exist: When a NAK occurs, it will set one of the PNAK_P* fields in the STA2 register, and this can be used to detect the occurrence of this issue. At this point, the value in the VALP* registers for the affected peripheral should be disregarded. In the case of an interrupt timeout there is no dedicated status bit, but if the VALP* registers of the peripheral matches the VALP* registers of another peripheral it could indicate the presence of this issue.

1.4 #5: Reset source invalid

Symptoms	The RST_SRC field in the STA1 register may report incorrect values, and should not be trusted, with the exception of bit 3 (Soft reset).
Conditions	Always.
Consequences	Incorrect field values.
Workaround	To detect a reset, it is possible to read one of the registers that was previously written to and see if the register has returned to the default (reset) value. If this is the case a reset of the nPZero has occurred.

1.5 #6: SPI shifts out 2 bytes for 8-bit data type in modes 1 and 3

Symptoms	SPI shifts out 2 bytes for 8-bit data type in modes 1 and 3
Conditions	Happens when *all* of the below conditions are met: 1) Peripheral used in SPI mode 2) SPI mode 1 or 3 used 3) 8-bit data type used
Consequences	Assuming the peripheral doesn't react negatively to the longer read there are no significant consequences, as the second byte read will be ignored by the nPZero.
Workaround	None

1.6 #7: Device reads out 2 bytes over I2C when using 8-bit data type

Symptoms	Similar to issue #6, the nPZero will read two data bytes when reading the value over I2C, even when 8-bit data mode is used.
Conditions	Happens when *all* of the below conditions are met: 1) Peripheral used in I2C mode 2) 8-bit data type used
Consequences	Assuming the peripheral doesn't react negatively to the longer read there are no significant consequences, as the second byte read will be ignored by the nPZero.
Workaround	None

1.7 #8: Power switch output voltage detection can cause lock-up

Symptoms	The device will hang after a peripheral is enabled when using the power switch with output voltage rise detection mode for the peripheral in question.
Conditions	Power switch output voltage rise detection feature used (PSWMOD_P*=0) for one or more peripherals.
Consequences	This mode should not be used, as it can cause lock-up of the device.
Workaround	Set PSWMOD_P* to 1 rather than 0, and add additional initialization time delay to account for the missing voltage rise detection. Typically the switch rise time is around 1ms, but this will depend on various factors, such as supply voltage and output load, and this will have to be adjusted for every design.

1.8 #9: False ADC wake-up trigger when ADC_IN is outside the operation range

Symptoms	False wake-ups if ADC_IN is outside the nominal range of 600mV to VBAT
Conditions	Happens when *all* of the below conditions are met: 1) External ADC channel wake-up enabled (WUP_A2) 2) ADC_IN pin floating, or set to a voltage below 600mV or above VBAT
Consequences	Unintentional wake-ups of the device.
Workaround	Either disable wake-up on the external ADC channel, or ensure that the voltage applied to the ADC_IN pin is always within the 600mV to VBAT range.

Tollbodgata 2 | 4611 Kristiansand | Norway
info@nanopowersemi.com | Telephone +47 989 02 050

European
Innovation
Council

Co-funded by
the European Union