Preventing Electrical Damage in ThingMagic Nano Modules

Dan Ratner

7/21/2016

Since the release of the Nano module, we have had a few reports of Nano modules ceasing to work properly after weeks or months of successful operation in customer-designed readers. The two must common symptoms are:

- The module fails to initialize and will not communicate with the host
- The module will initialize, but presents a "firmware CRC error" (Error number 0200h) to the host indicating that the application firmware is corrupted.

We believe the cause of these failures is memory corruption resulting from the circuitry around the Nano not following the interface guidelines we give in our User Guide, which are based on requirements from the supplier of the processor IC we use. There are two areas are of particular concern:

1. Input lines remaining high when the Nano module is turned off
2. Programming pin SWCLK allowed to float if not used

Input Line Requirements

In the Nano Design Guide, we give the following requirements for input lines. (The relevant clause is underlined for emphasis.)

Input High-level Voltage	1.9 V min to indicate high state; 3.7 V max when module is powered up, no more than 0.3 V higher than Vout when module is turned off to prevent damage.

"Vout" is the DC voltage that the Nano sources on pin 10 and is nominally 3.3V as long as the DC input voltage (pins 16 and 17) is greater than 1.8V.

This input voltage restriction applies to both the serial interface input line ("receive" side) and the GPIO lines. Our recommended carrier board schematic diagram shows a buffer circuit that would enforce this requirement for the RX input line. We did not take steps to protect the GPIO lines on the carrier board because they can be used as either inputs or outputs depending on how the Nano is programmed.

We have discovered that the most common "damage" that is caused by violating this requirement is corruption of the bootloader or application memory. Application memory corruption causes the module's bootloader to report a firmware CRC error, indicating that the firmware is not operational. If Bootloader memory is corrupted, the module may not respond at all. The altered memory cannot be repaired through user-accessible module interfaces, so the module cannot be repaired in the field.

Options for the way Nanos are shut down (by DC power or Shutdown line) and by the way the Nano and input drivers are powered (by an independent supply or the same supply) leads to three solutions, depending on the use case:

Case 1: If the Nano is powered by the same source that powers the circuitry that drives the input lines to the Nano (so that everything is powered up or down together), and the Shutdown line is not used to power off the Nano, there should be no potential for the input lines to remain high when the Nano is powered off and the Nano should never experience this memory corruption issue. See Figure 1.

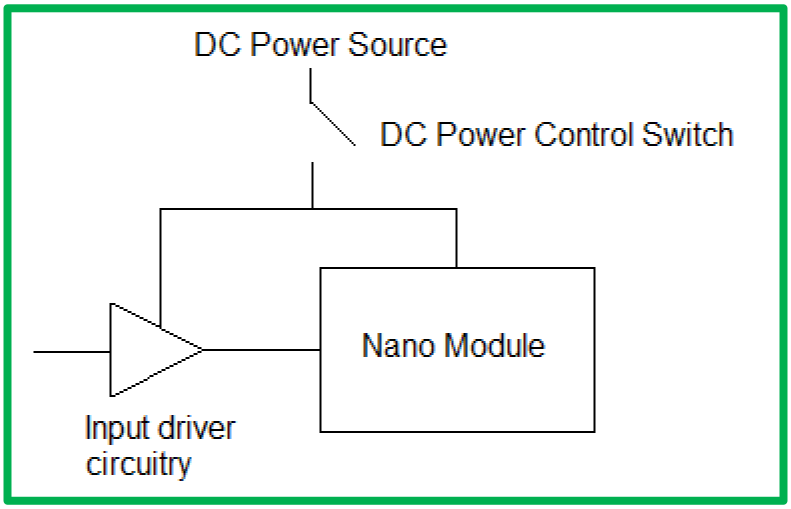


Figure 1

Case 2: If the Nano is powered by a different circuit than the logic that drives the RX and GPI lines, so that the Nano's power could go down while the voltage to the input lines remains high, the memory corruption issue could occur (regardless of whether the Nano is powered down through dropping the DC supply voltage or using the Shutdown line.) See Figure 2.

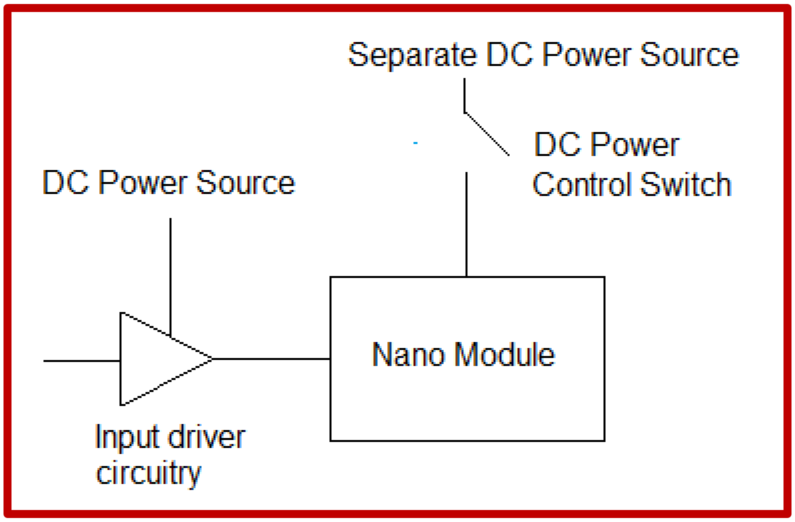


Figure 2

Case 3: If the Nano is powered by the same source that powers the circuitry that drives the input lines to the Nano (so that everything is powered up or down together), but the Shutdown line is used to power off the Nano while all other circuitry remains active, there is a potential for the memory corruption issue to occur. See Figure 3.

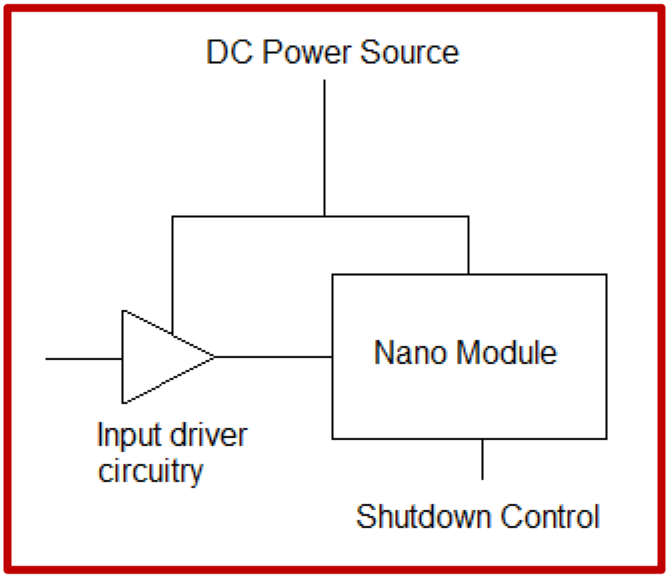


Figure 3

In the second and third cases, there are two ways to prevent the issue from occurring:

(1) Ensure, though software, that all input lines are low before shutting off the Nano module and remain low until power is applied to the Nano to turn it on again. See Figure 4 and Figure 5.

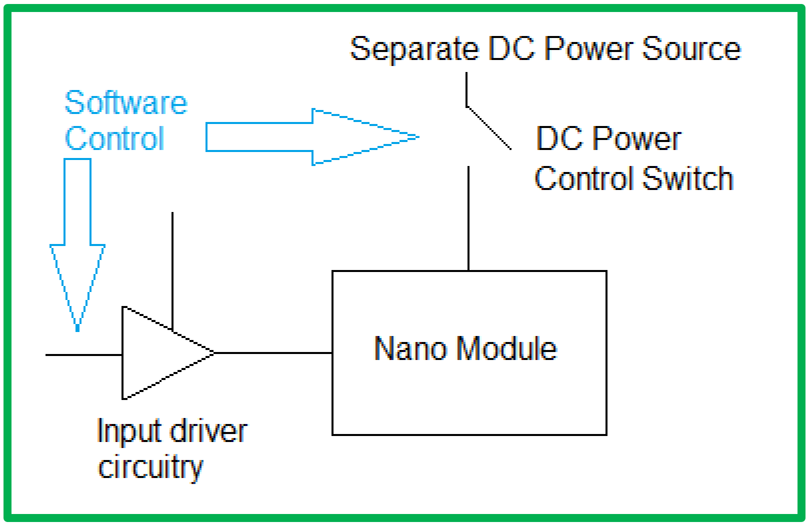


Figure 4

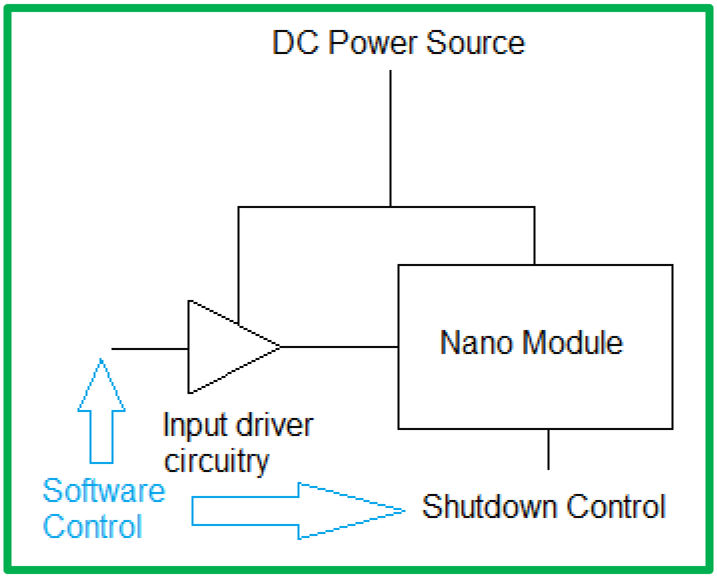


Figure 5

(2) Add buffer circuitry to drive the input lines and power this circuitry from a voltage source originating from the Nano. That way, dropping power to the Nano drops the power to the driver circuit as well, so the input voltage can never remain high when the Nano is off. See Figure 6 and Figure 7.

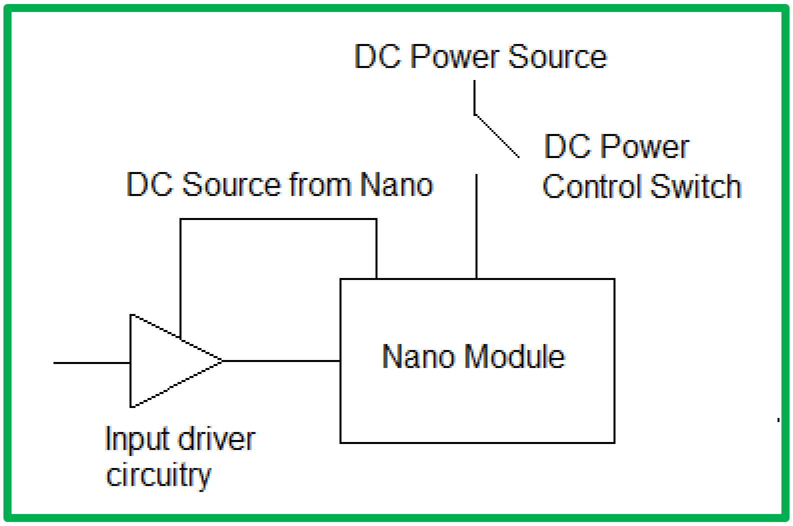


Figure 6

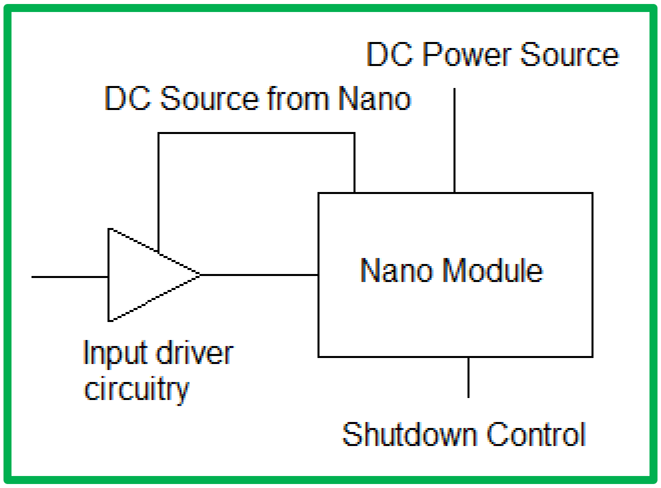


Figure 7

When we designed the Nano Carrier Board, we added a Nano-powered buffer to the RX input line (NXP 74LVC2G17). We did not buffer the GPIO lines, not knowing whether they would be used as inputs or outputs in customer applications.

Figure 8 is a schematic diagram of the buffers on the Nano Carrier Board:

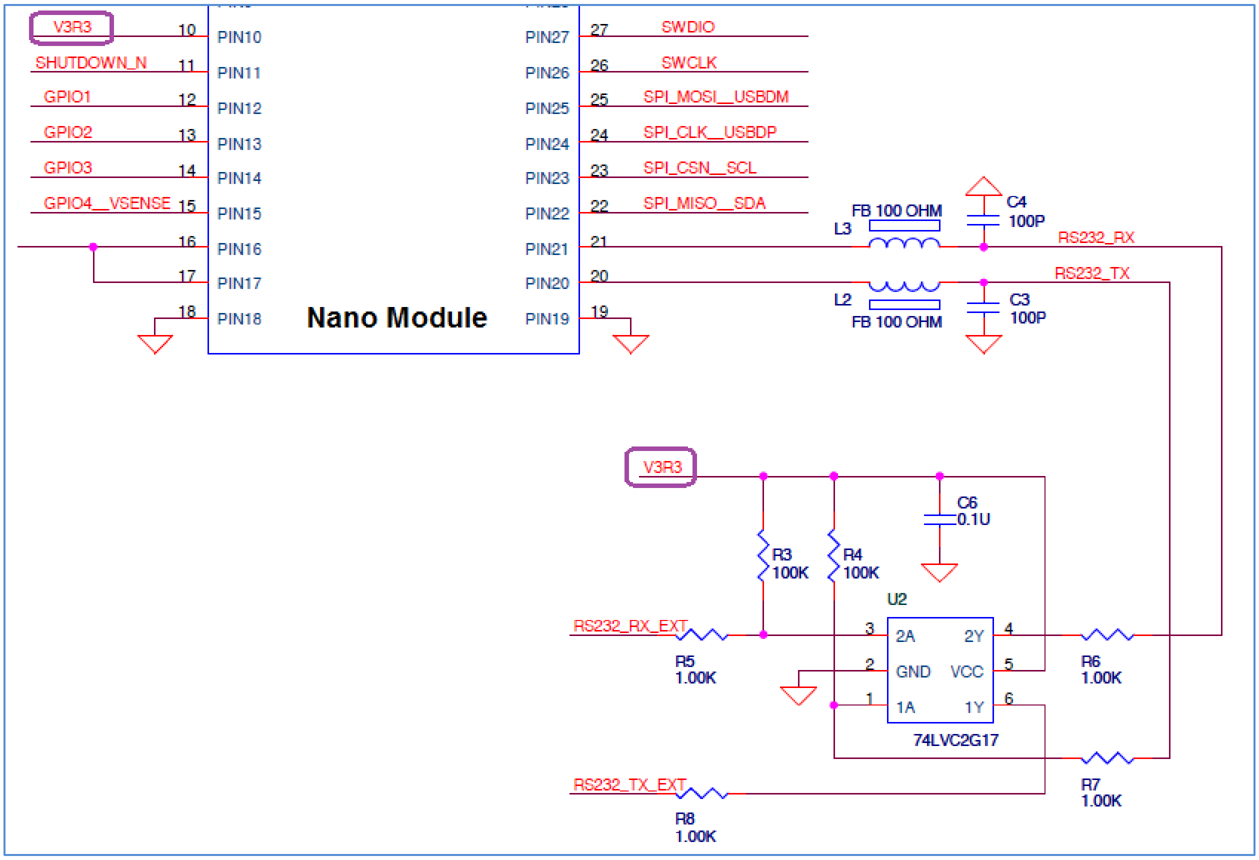
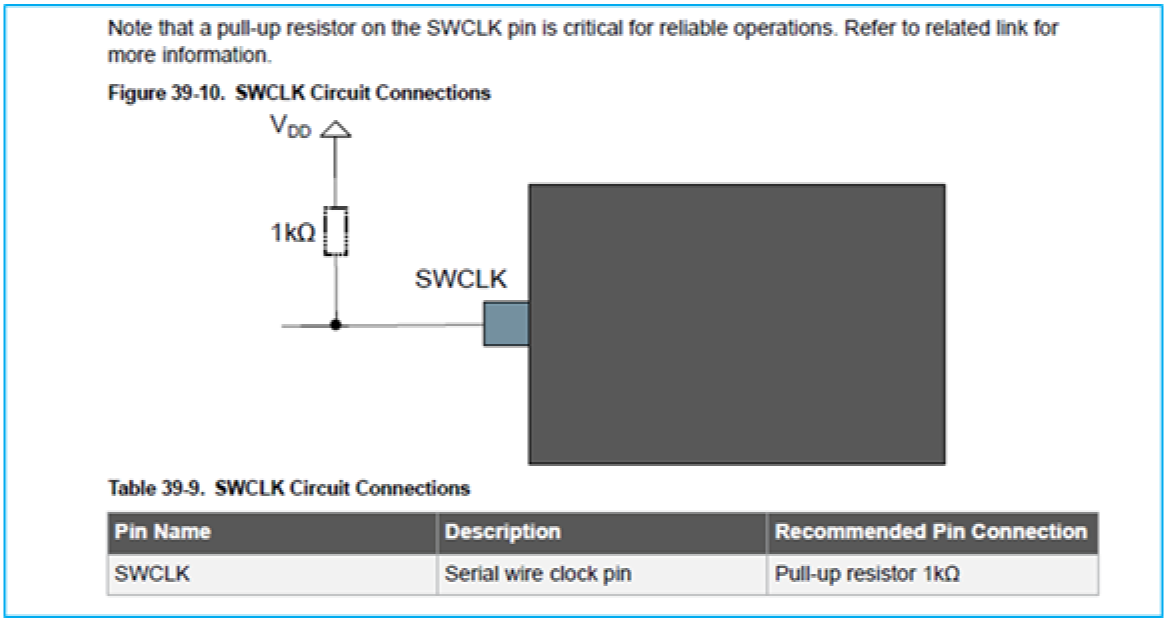


Figure 8

V3R3 in this schematic (pin 10) provides DC power for the buffer, and is the same voltage used to power the internal circuitry of the Nano and is switched by the Shutdown line. That ensures that the output of the buffer circuit can never be higher than the power that drives the Nano processor, and compliance with input Voltage requirements is assured.

SWCLK Pin Requirements

There is an additional design recommendation which impacts the Nano that was recently published by the processor’s manufacturer. It was revealed so recently that our current carrier board design does not reflect it. The warning is reproduced here in Figure 9:



Note that a pull-up resistor on the SWCLK pin is critical for reliable operations. Refer to related link for more information.

Figure 39-10. SWCLK Circuit Connections

Table 39-9. SWCLK Circuit Connections

Pin Name	Description	Recommended Pin Connection
SWCLK	Serial wire clock pin	Pull-up resistor 1kΩ

Figure 9

In order to comply with this requirement, you must add a 1K resistor between pin 10 and pin 26 of the Nano module, as shown here in Figure 10:

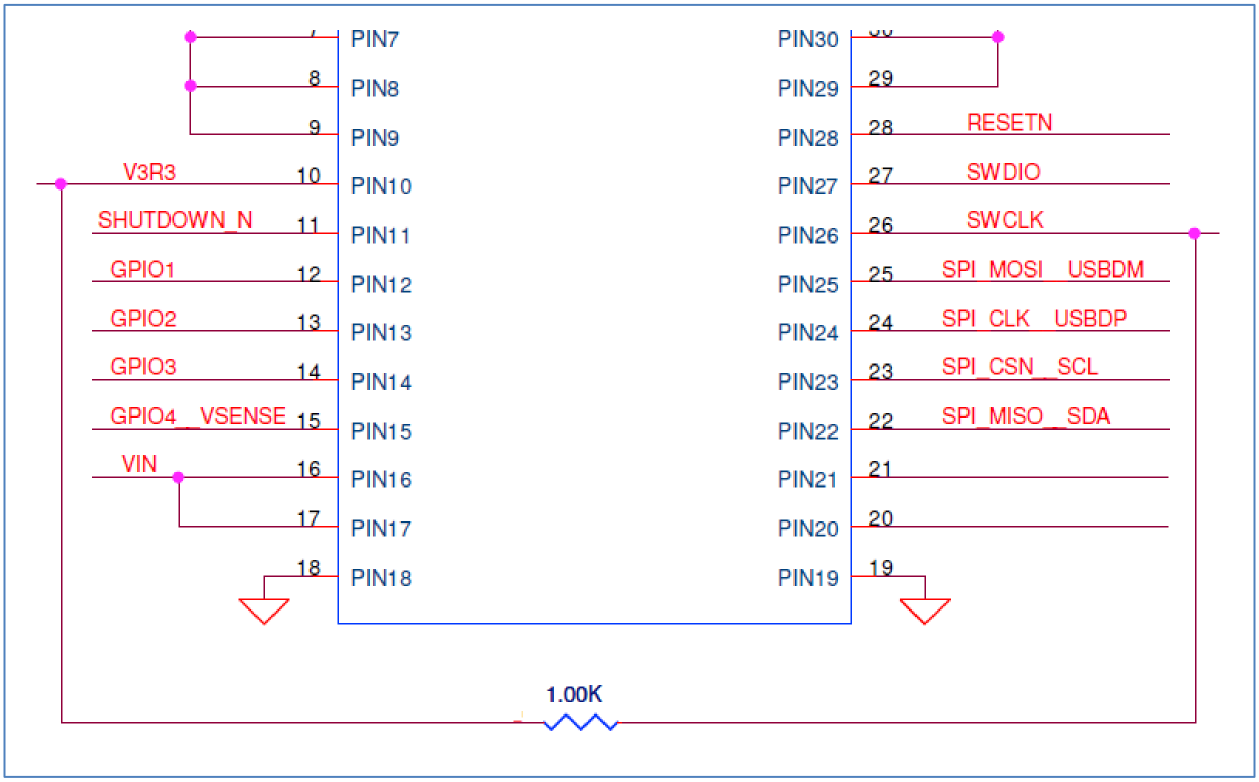


Figure 10

If you have further questions about these issues and our recomendations, please contact support@thingmagic.com.