Capturing Pulse Data on a Pulsing Reactor

General Atomics NPP-1000

- Controller: Netburner MOD 5441x module
- RTOS: µC/OS priority-based OS
- 12-bit A/D converter capable of 100 ksps
- Captures 50,000 12-bit samples in ½ second
- Communicates with GA digital console using Ethernet
- For pulsing, two main commands: acquire pulse and download pulse

General Atomics NPP-1000 (2)

Acquire pulse operation:

- NPP receives command via Ethernet to acquire pulse
- Firmware monitors ADC, waiting for power to rise above a 3% threshold (timeout: 15 seconds).
- Once the input achieves the 3% threshold, the firmware captures 50,000 readings, at approximately 10 µSec/reading.
- The firmware scans backwards (from the end) in the data to eliminate readings below 3%.
- After capturing data, control returns to other NPP operations.

General Atomics NPP-1000 (3)

Download data

- A separate Ethernet command instructs the NPP firmware to transmit the last pulse's data to the client (e.g., GA digital console).
- Typical download time: 5 to 15 seconds.
- The downloaded data is a text file containing two entries per line and one line for each reading.
- Data format: *time value*
- Time is the difference in µSec between this reading and the previous
- Value is the 12-bit ADC reading.

General Atomics NPP-1000 (4)

In theory, it all works great, except:

1. The NPP hardware has a watchdog timer that must be reset every two seconds. Should the watchdog time out, it puts the NPP-1000 into failsafe mode (all trips active).
2. While capturing data, the Netburner CPU is working strictly on reading the ADC. Resetting the watchdog timer doesn't always work. This sometimes leads to a locked up NPP-1000 (requiring a power cycle).
3. There seem to be some issues with downloading, too.

General Atomics NPP-1000 (5)

Two solutions:

1. Fix the firmware (tried this on several occasions)
2. Quit using the NPP-1000 built-in download capabilities and do something else.

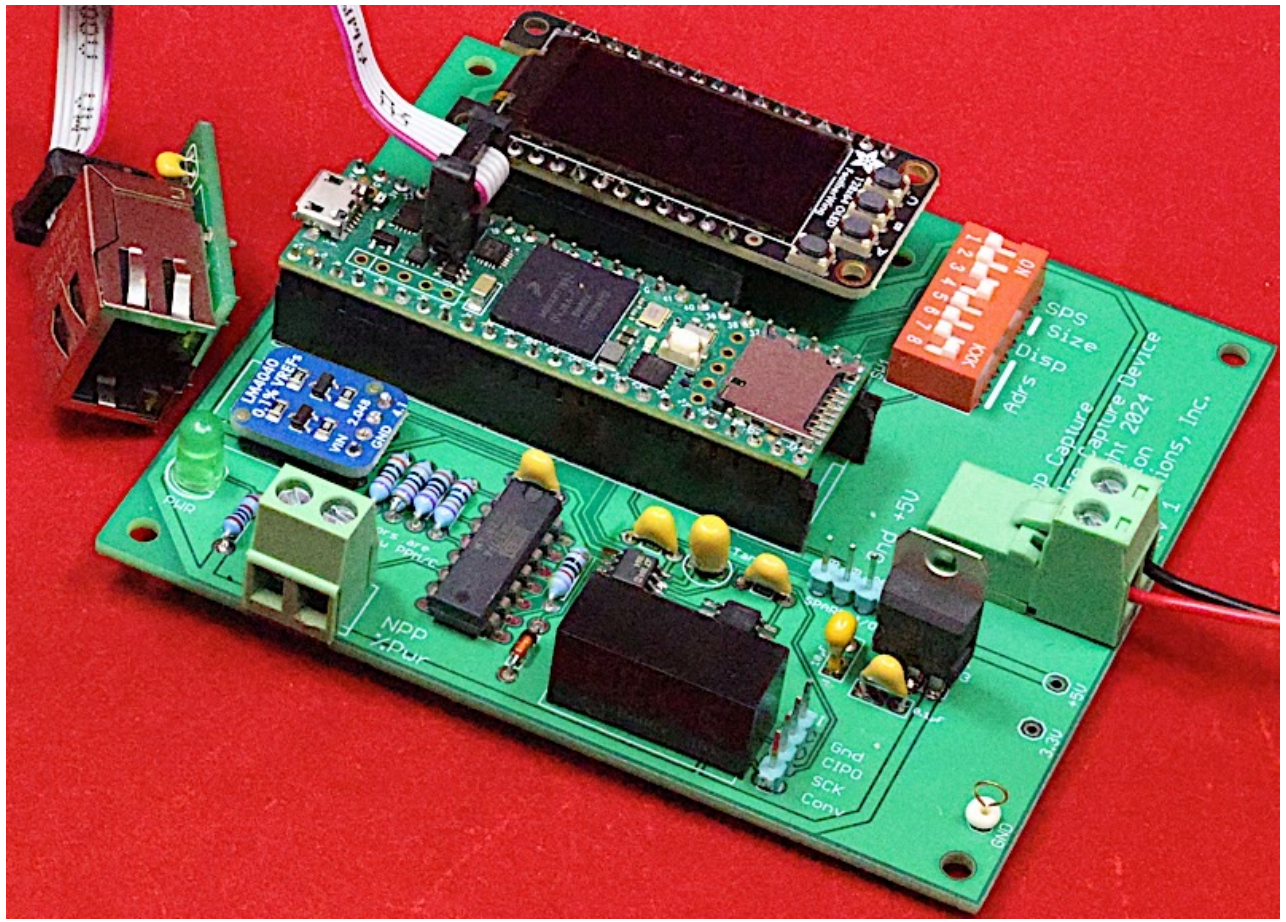
It’s likely that fixing the firmware may not be possible due to the performance limitations of the Netburner MOD5441x and the priority-based nature of µC/OS. So…

The NPP Pulse Capture Device

The Plantation Productions' NPP Pulse Capture device offloads the pulse capture and download operations to a separate embedded microcontroller, allowing the NPP-1000 to operate strictly in safety channel mode.

Pulse Capture Device (PCD) Hardware

- Teensy 4.1 (ARM Cortex M7F) microcontroller w/Ethernet interface
- LTC 1864 16-bit ADC (up to 250 ksps)
- 2.048V precision voltage source for ADC reference
- OPA4277 instrumentation opamp (voltage follower input and convert NPP 0-10V power signal to 0-2V for the LTC 1864)
- 128x64 OLED display


Sps
Size
Disp
Adrs
PWR
Gnd +5V
NPP
Gnd
CIPO
SCK
Conv
+5V
3.3V
GND

Pulse Capture Device (PCD) Signal Interface

- The PCD uses (isolated) 0-10V output from the NPP as its signal input
- A high-impedance “voltage follower) circuit prevents the PCD from affecting the 0-10V signal, if the site is using this signal for other purposes.
- For the extremely paranoid, you can use an optional ”PPISO4” isolation amplifier to isolate the PCD from other instrumentation on the 0-10V signal. This would protect that external equipment from the PCD and vice-versa. Also useful if using a different pulse data source. Under normal circumstance, the PPISO4 is unnecessary.

Pulse Capture Device Firmware

Teensy 4.1 firmware handles four commands:

- Capture pulse
- Download pulse
- Simulate pulse (two variations)
- Version number request

Capture Pulse Command

- Waits for the power level to rise above 2%
- Saves the last 1,024 ADC readings once it reaches the threshold
- Captures 50,000 readings upon hitting the threshold
- Prepends the 1,024 pre-threshold readings (or less, if threshold was found before the 1,024th reading) to the beginning of the pulse data
- Scans backwards to find the last reading greater than 2% (eliminating the data after the pulse)
- Leaves the data in memory for the download command

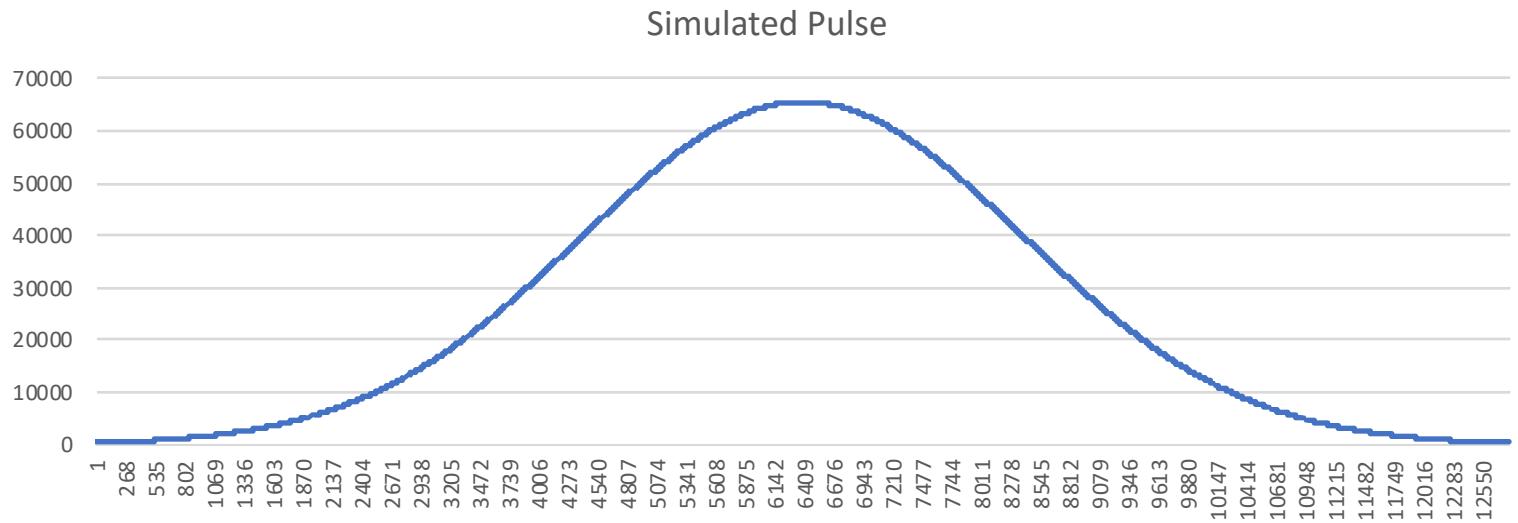
Download Command

- Transmits the last pulse data to the client via Ethernet
- Transmission format:
  time: *x1 x2 x3 x4 … x16*
- 16 ADC readings per line, each reading is a 16-bit hexadecimal value
- *time* is a 16-bit hexadecimal value specifying the time (in µSec) for the first reading on the line (each reading is 10 µSec)
- Download time is approximately 5-10 seconds for 50,000 readings (each line is delayed to allow the client to process the data).

Simulate Commands

Two simulate commands

- A very short pulse (approximately 32 data points) used to test the firmware from an Ethernet terminal program
- A larger pulse (around 13,000 readings) using a normal curve

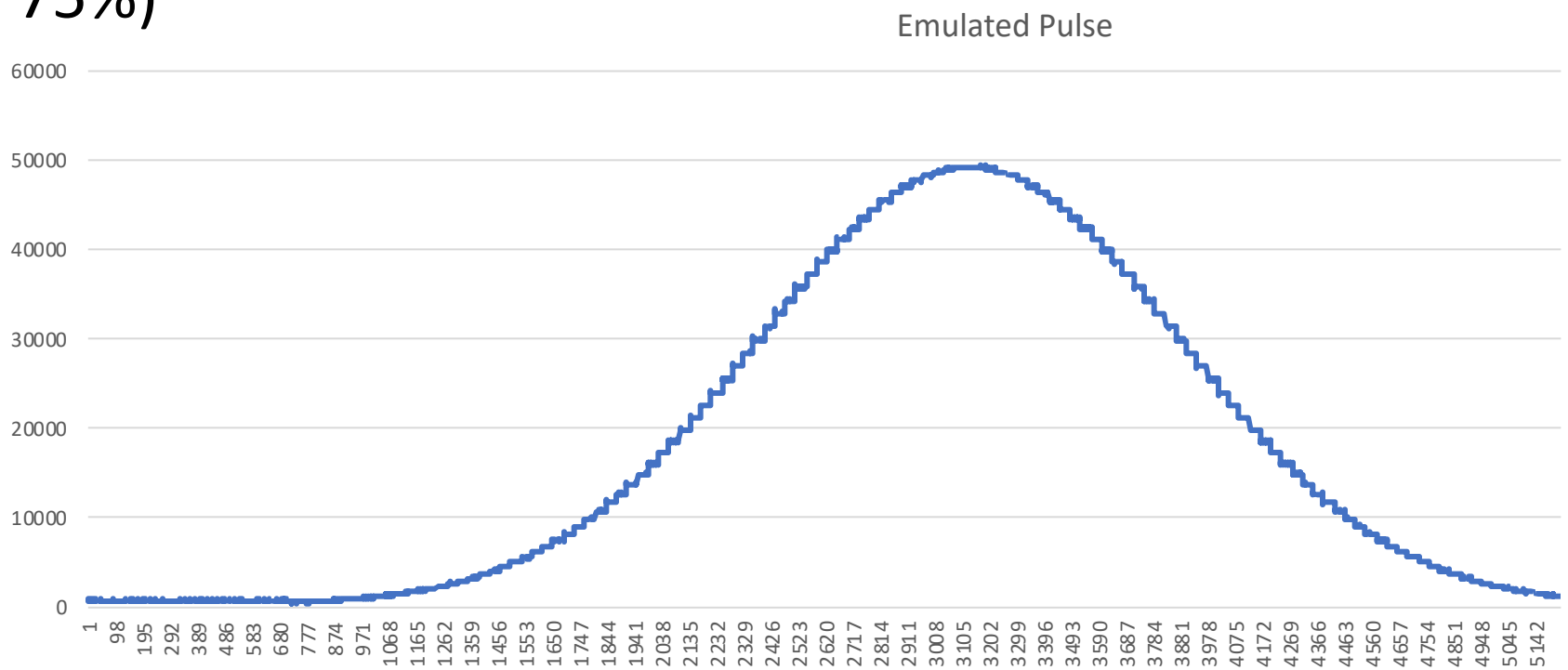


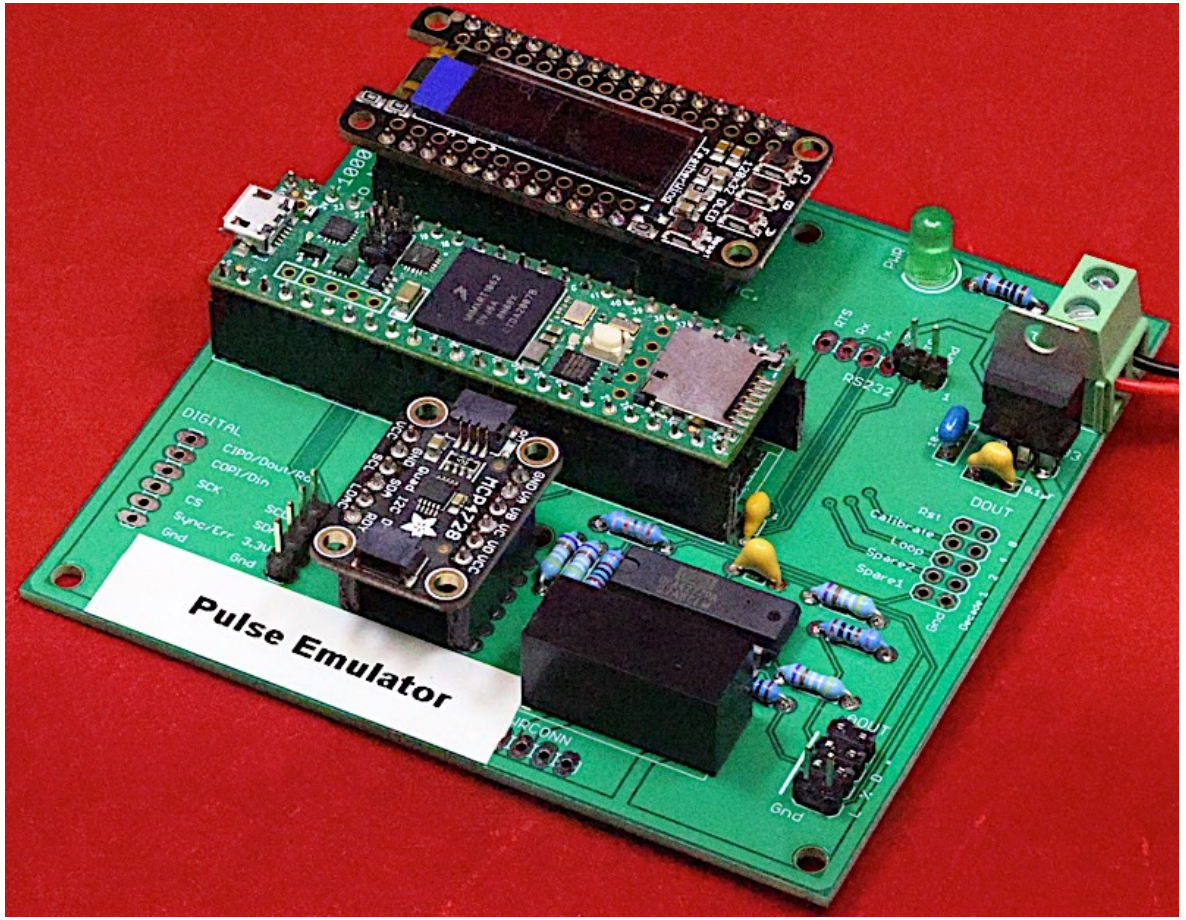
Version Command

- Returns the firmware version number (currently 01.00.00)
- Client software can use this to ensure the PCD is running the latest firmware

Pulse Emulator Device

- For testing the PCD.
- Teensy 4.1-based board with a D/A converter producing various simulated pulses (width = 12, 25, or 50 mSec, amplitude = 25%, 50%, 75%)




Pulse Emulator
DIGITAL
RS232
DOUT
Calibrate
Loop
Spare2
Spare1
Gnd
MCP4728

Questions