Communications

The board speaks two types of protocols: text and binary. The *text protocol* is intended for direct human interaction such as for testing. The *binary protocol* is intended for machine to machine communications.

The text protocol is in the CLI style. There is a series of commands. Each command takes certain parameters. One can get a list of supported commands by typing ? Followed by the Enter key in a terminal that is connected to the board.

The binary protocol is a simple request/response model. The base component is a *message*. A controlling device sends a message to the board and the board responds in turn with another message. Specific functionality within the board is addressable via *call indexes*. A call index is a unique eight bit value that identifies a function. A call index is sent as part of the request message and also as part of the response message.

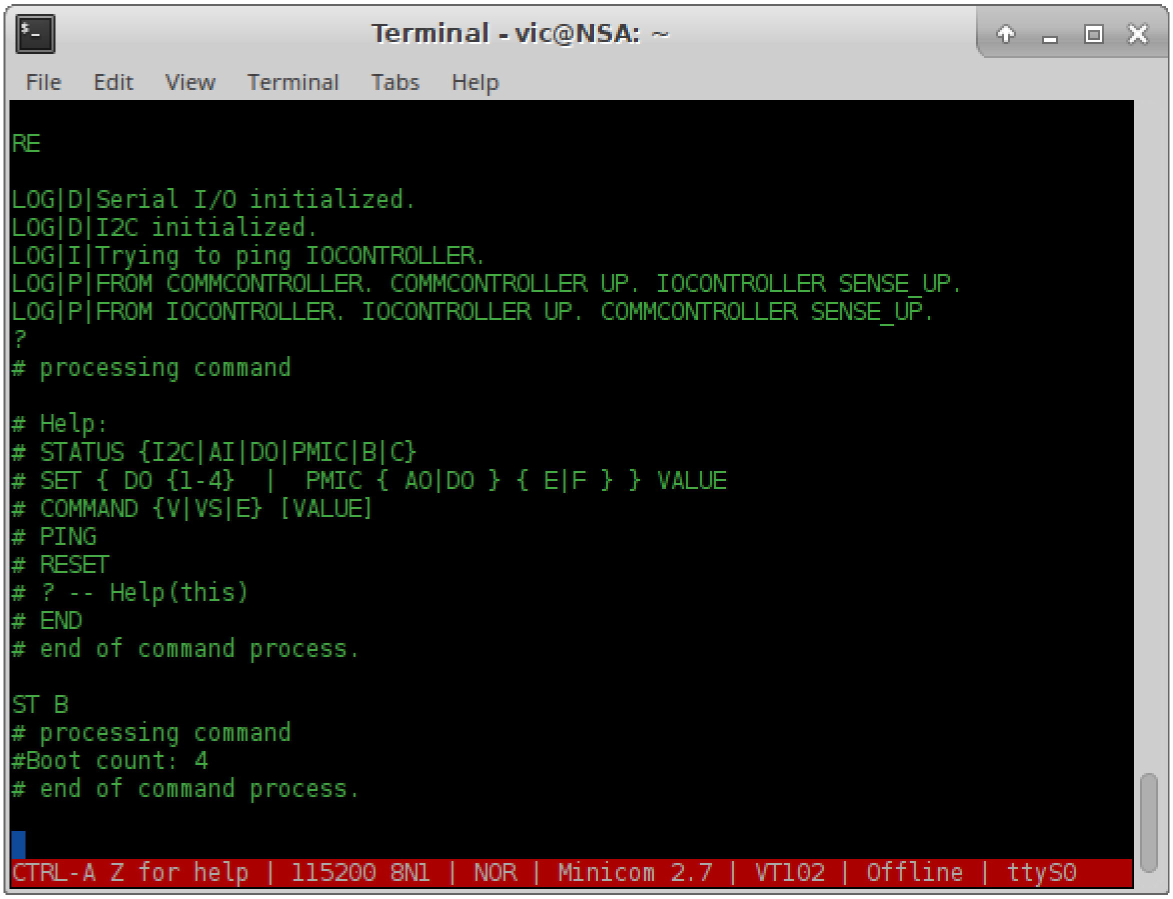
Physical Connection

Software

115200 8N1

Copyright © 2017 Vidas Simkus All Rights Reserved.

Terminal



```
RE

LOG|D|Serial I/O initialized.
LOG|D|I2C initialized.
LOG|I|Trying to ping IOCONTROLLER.
LOG|P|FROM COMMCONTROLLER. COMMCONTROLLER UP. IOCONTROLLER SENSE_UP.
LOG|P|FROM IOCONTROLLER. IOCONTROLLER UP. COMMCONTROLLER SENSE_UP.
?
# processing command

# Help:
# STATUS {I2C|AI|DO|PMIC|B|C}
# SET { DO {1-4}  |  PMIC { AO|DO } { E|F } } VALUE
# COMMAND {V|VS|E} [VALUE]
# PING
# RESET
# ? -- Help(this)
# END
# end of command process.

ST B
# processing command
#Boot count: 4
# end of command process.


CTRL-A Z for help | 115200 8N1 | NOR | Minicom 2.7 | VT102 | Offline | ttyS0
```

LOGIC_CORE

Binary Protocol

All multi-byte binary protocol types (for example integers) are sent down the wire with the MSB (Most Significant Byte) first. In other words, assuming a transmission of two words I_1 and I_2 , the receiving buffer will look like this:

Buffer index 0	Buffer index 1	Buffer index 2	Buffer index 3
I_1 MSB	I_1 LSB	I_2 MSB	I_2 LSB

> NOTE: This is in contrast to the I2C protocol which sends data down the wire LSB (Least Significant Byte) first. This is a design feature and definitely not an oopsie.

Copyright © 2017 Vidas Simkus All Rights Reserved.

The binary protocol is a simple request/response model. The controlling device sends a request and the board responds with a response.

Request

The request message has three fields:

1. Binary message identifier (one byte)
2. Request payload length (one word)
3. Request payload (function specific)

Binary message identifier

This is always 0x40 (ASCII @). Presence of this character indicates to the message parser on the board that the rest of the bytes are part of a binary message.

Payload length

This is a single word value that indicates the length of the payload in bytes. Maximum payload length is 65536 bytes.

Payload

Stream of bytes that are specific to the function.

Request Message

Binary message identifier								Request payload length MSB							
BYTE 0								BYTE 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x40															

Request payload length LSB								Request payload							
BYTE 2								BYTE 3 ...							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Request Message Payload

The only standard part of the payload is the first byte which is expected to be the call index. All remaining bytes are function specific.

Copyright © 2017 Vidas Simkus All Rights Reserved.

Continued on next page

Copyright © 2017 Vidas Simkus All Rights Reserved.

Response

The response message is very similar to the request message. The response message has the following fields:

1. Binary message identifier (one byte)
2. Call index (one byte)
3. Result code (one byte)
4. Response payload length (one word)
5. Response payload (function specific)

Binary message identifier

This is always character 0x10.

Call index

The call index that this message is in response to. This will be the same call index as in the request message.

Result code

Function specific result code. The only standard codes are one and zero; one is success and zero is failure.

Response payload length

Single word that indicates the length of the response payload.

Response payload

Unlike the request payload there is no structure. The response data is function specific.

Continued on next page

Copyright © 2017 Vidas Simkus All Rights Reserved.

Response Message

Binary message identifier								Call index							
BYTE 0								BYTE 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0
0x10								*Same as the call index in request*							

Result code. 1 == success								Response payload length MSB							
BYTE 2								BYTE 3							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

> NOTE: The binary message identifier is different between the request and response messages.

Binary commands

0x00 – Reset board

0x01 – Get analog input values

0x02 – Get digital output status

0x03 – Set digital output status

0x04 – Get PMIC status

Copyright © 2017 Vidas Simkus All Rights Reserved.

0x05 – Set PMIC status

0x06 – Get L1 calibration offsets

0x07 – Get L2 calibration offsets

0x08 – Set L1 calibration offsets

0x09 – Set L2 calibration offsets

0x0A – Get boot count

Copyright © 2017 Vidas Simkus All Rights Reserved.

BYTE 0								BYTE 1							
7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0

Copyright © 2017 Vidas Simkus All Rights Reserved.

Text Protocol

STATUS

I2C

AI

DO

PMIC

B

C

SET

DO

PMIC

COMMAND

V

VS

E

PING

RESET

?

Copyright © 2017 Vidas Simkus All Rights Reserved.