APPLICATION NOTE

CANBus Prototyping Setup Guide

Using USB-CAN-A Adapter and NTN-5K-224CAN as example hardware.

Document No.	AN-CAN-NTN5K-TD
Revision	A
Date	August 2026
Applies To	MEAN WELL NTN-5K-CAN Series Specific Instructions for NTN-5K-224CAN
Prepared By	Thomas Denny | TRC Engineering Dept.

Safety Precautions

Beware of high voltage on NTN-5K-224CAN output terminals. ⚠️

DC Circuit breakers or fuses are recommended between batteries and inverter as an extra precaution.

Do **NOT** connect AC input to NTN-5K-224CAN until it has been properly programmed to charge batteries at the correct rate. Prematurely connecting AC can result in overcharging batteries or protection modes.

1. Overview

This guide will demonstrate how to initiate communication between windows based software and a MEAN WELL power supply via a USB to CANBus adapter. The documentation will include specific references for setting up operation of an NTN-5K-224CAN for testing with 2 x PSL-BT-121000 (100AH Lithium) batteries.

These set up steps are **critical** for this combination, as **failure to do so would result in improper battery charging**. The recommended CANBus adapter made by Waveshare is readily available and has free software. Many of the principles can be adapted to work with other power supplies, batteries, or CAN controllers, and some explanation will be provided on how to adapt to other scenarios.

IMPORTANT

Incorrect setup can cause improper battery charging, which may lead to battery damage or reduced lifespan.

2. What you'll need.

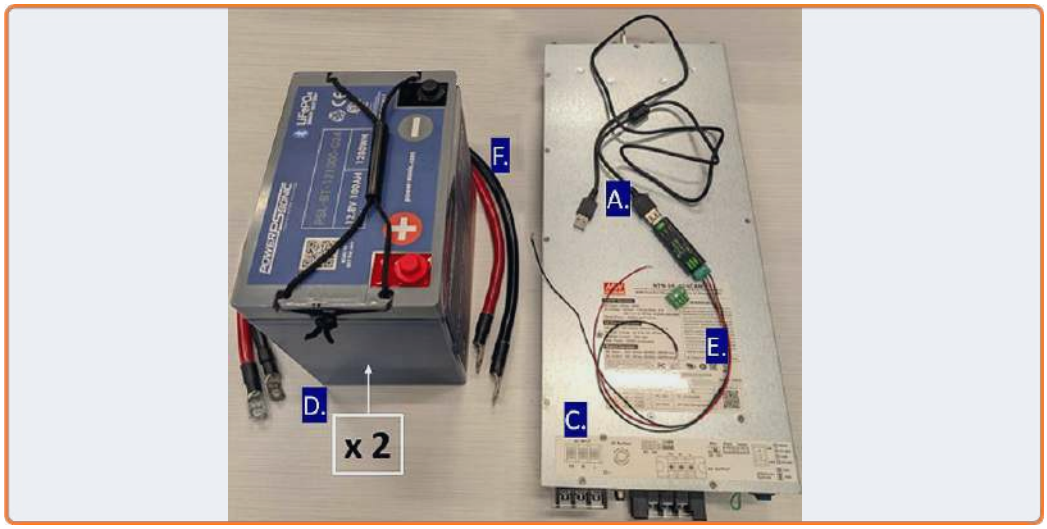
Hardware

- A Waveshare USB-CAN-A
- B Windows PC (not pictured)
- C NTN-5K-224CAN (Or other CANBus enabled PSU)
- D 2 x PSL-BT-121000 (If working with NTN-5K-224CAN)
- E 3 x Signal wires for CANH, CANL, GND (24AWG wire used in example)
- F Cables for power connections

Software / Tools

USB Driver:
CH341SER driver

CAN Configuration Software:
USB-CAN-A_TOOL_2.12



Full Kit (not including computer)

3. Hardware Setup

The below steps explain the setup of the hardware portion of communication via the USB-CAN adapter. These steps assume that your input power is already connected. For the NTN-5K-224CAN being used with PSL-BT-121000, this means that your 2 x 12V 100Ah batteries are already connected in series and connected to the DC input. **Do NOT** connect AC input in this example, as that would immediately start charging the batteries.

IMPORTANT
Do NOT connect AC input in this example, as that would immediately start charging the batteries.

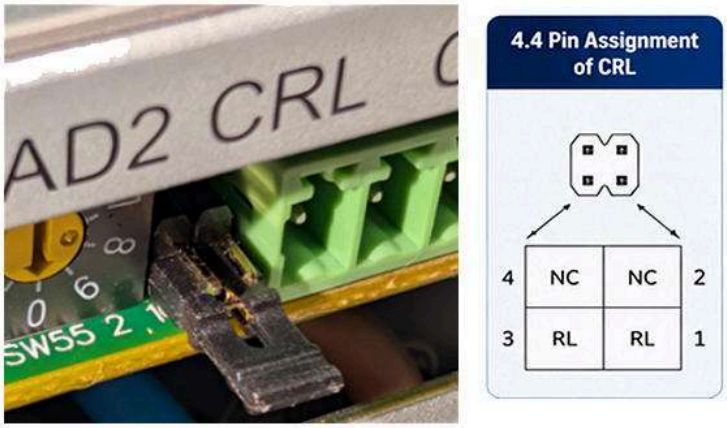
Step 1: Determine / Set address of power supply.

In this example, we will set the address to "00." This address will be used when communicating with the power supply. Different power supplies use different methods of setting the address: Some use jumpers, some use dip switches. For the NTN-5K, there are 2 rotary switches on the front of the unit. Set both arrows pointing down to set address to "00" as seen in the image to the right.

Hardware Step 1: NTN-5K-224CAN address set to 00.

Step 2: Set / Connect any required jumpers or termination resistors.

For proper communication, some power supplies require jumpers to be moved or termination resistors installed. For the NTN-5K-224CAN, you simply move the pre-installed CRL jumper from the top two pins to the bottom two pins (pins 1 & 3.) This connects a termination resistor across the data lines to prevent interference from noise. This is especially important if data transmission lines are longer.



Hardware Step 2: CRL jumper in correct position for communication (across pins 1 & 3.)

Step 3:

Wire WaveShare USB-CAN-A data lines to power supply connector.

For the NTN-5K-224CAN, this means using the included COMM connector and making the following connections:

1. Pin 1: GND
2. Pin 2: CANH
3. Pin 3: CANL

For other PSUs, check manual for data lines.
It's always recommended to connect signal ground.

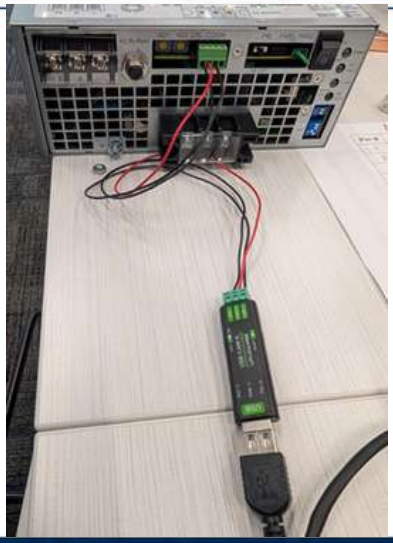
NOTE: When looking at these two connectors side by side as shown below, note how the pin order is REVERSED.

Correct wiring of USB-CAN-A to NTN-5K-224CAN COMM connector.

Step 4:

Connect Waveshare USB-CAN-A to power supply and computer.

The USB-CAN-A comes with a Male-Female USB extender cable to facilitate easy connection to computer.

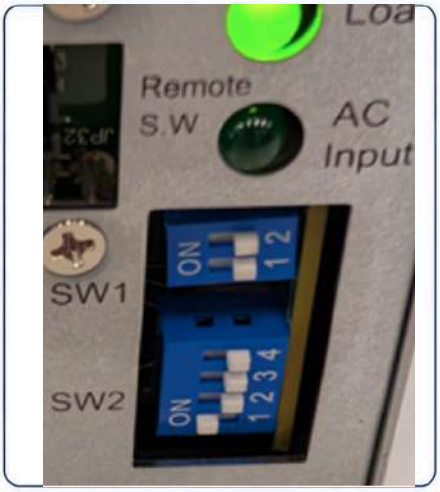


USB-CAN-A wired between NTN-5K-224CAN and computer.

Step 5: Make any other required hardware changes.

For this example, we set the AC output to 230Vac (Default setting was 220Vac output.)

- Dip SW2, S1: **ON**
- Dip SW2, S2: **OFF**



AC OUTPUT VOLTAGE, FREQUENCY, SAVING MODE SELECTABLE BY THE SW2				
S1	S2		S3	S4
OFF	OFF	100Vac or 200Vac	ON : 50Hz	ON : Saving Mode
OFF	ON	110Vac or 220Vac		
ON	OFF	115Vac or 230Vac	OFF: 60Hz	OFF: Non-Saving Mode
ON	ON	120Vac or 240Vac		

Hardware Step 5: Dip switch settings for 230Vac output from NTN-5K-224CAN

4. Software Setup

Make sure the hardware from Section 3 is powered on and connected before continuing.

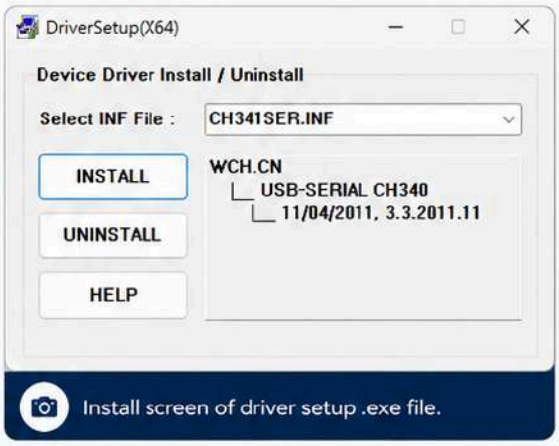
The instructions here are specifically written for the software made by Waveshare for the USB-CAN-A. There are other USB-CAN software tools available also. Full software setup instructions can be found here on Waveshare website: https://www.waveshare.com/wiki/USB-CAN-A. But important steps are all included below

STEP 1 Install the USB to SERIAL driver.

Download, unzip, and run the executable file. (CH341SER.EXE)

Download driver file:
https://files.waveshare.com/upload/c/c4/CH341SER.zip

.



Install screen of driver setup .exe file.

STEP 2 Download the Waveshare USB-CAN tool software.

Download the software found at the link below. Unzip to a folder of your choice. Note that there is no “Install” file for this software, therefore to open the software, simply run the executable file shown in the image below. (USBCANV2.12.exe) You may want to create a shortcut to this file and place it either on your desktop or in your start menu.

Download software file:
https://files.waveshare.com/wiki/USB-CAN-A/Tool/USBCANV2.12_English.zip

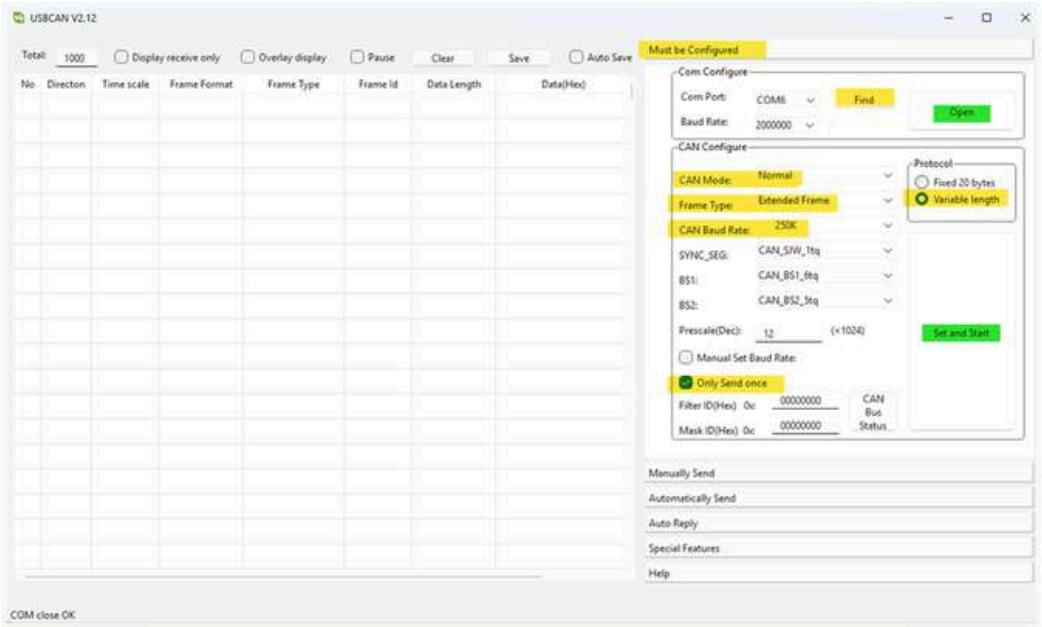
Software Step 2: USB-CAN tool software file.

STEP 3 Setup software configuration settings to match power supply.

In order to connect the software to the power supply, the below settings must be entered into the “Must be Configured” area. Extended frame and 250k baud rate are standards used by all MEAN WELL power supplies currently.

Below are the steps to follow for this portion of the setup. The image shows the important sections highlighted in yellow, and then the green highlighted areas show where to click after settings are entered.

1. Click Must be Configured section
2. Find COM port
3. Set CAN Mode to Normal
4. Frame type to Extended Frame
5. CAN Baud Rate to 250k (Note: This is different from “Baud Rate” under Com Configure section. That is USB rate, default 2M.)
6. Protocol to Variable length
7. Specify Only Send Once
8. CLICK “Open” in the Com Configure section
9. CLICK “Set and Start” in the CAN Configure section



Software Step 3: USB-CAN tool configuration settings

Step 4 Test Communication: "Manually Send" tab of the USBCAN tool.

This step requires entering a **"Message ID"** and then **"Data."**

Note: All values used are going to be in Hexadecimal, which is what the "0x" specifies. Whenever you see "0x" followed by an alphanumeric sequence from "zero" to "F", this is a Hexadecimal number.

1 Message ID:

This can generally be found in your power supply's manual, and will include a standard set of characters followed by the address of the power supply. As can be seen in the table below, when communicating from our controller to the NTN-5K-224CAN with an address of 00, we will use the following:

Example Message ID	0xC0500.

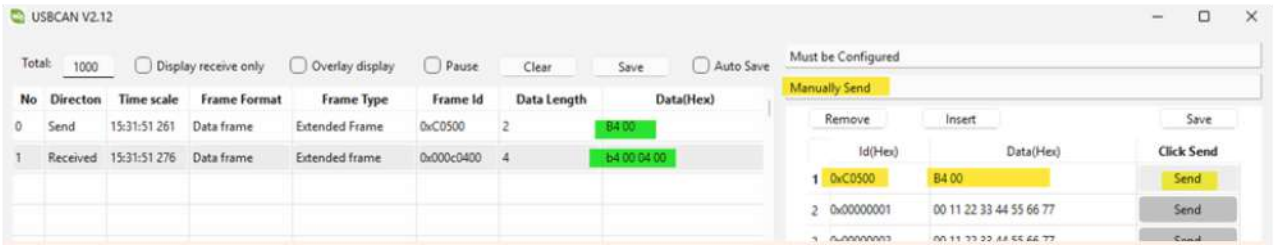
2 Data:

- What is entered in the "Data" column is going to include our "Commands" PLUS any values that we are writing. There are a few elements here that can be slightly confusing to those unfamiliar with this type of programming.
- First, commands can be either "read" or "write." When you "read," you send the command only, and the system will return the data in the appropriate memory location. When you "write," you send the command followed by the data values that you want to store in memory.
- Second, is that you have to reverse both commands and data bytes, but separately. This is so the "low byte" is read first. A space is also added between the two bytes as a delimiter (separation that the system recognizes between each byte.)
- In this example, we are going to do a read of "0x00B4" (Curve Config.) To do this, we are going to send the command (data) below:

Example Data	B4 00

6. 2.1 MESSAGE ID DEFINITION	
Message ID	**Description**
0x000C04XX	NTN-5K to Controller Message ID
0x000C05XX	Controller to NTN-5K Message ID
0x000C05FF	Controller broadcasts to NTN-5K Message ID

Note: XX means the address of the NTN-5K. Please refer to 4.7 Communication



Software Step 4: Message ID table for NTN-5K and communication test result with USBCAN software.

When we do this, and click “Send,” The NTN-5K-224CAN should send a response, which the USBCAN tool will show as a “Received” line in the history. This will include the command and the value associated. In this example, if we have communication, the NTN-5K-224CAN should send a response of:

Data: **b4 00 04 00**

Which, when you flip the two pairs of bytes, translates to **0x0004** as the data stored in the command **0x00B4** location. To see what this value means, see the resources section which includes the binary breakdown of a couple memory registers.

STEP 5 Let’s program! - Send all commands to make necessary changes and read values.

If you have received the response from the power supply as described in Software Step 4, then **CONGRATULATIONS! You are successfully communicating via CANBus!**

At this point, you can commence with communication with the power supply. For our example, we are going to send the critical values for properly charging and discharging 2 x PSL-BT-121000 batteries. We’ll want to enter the following data to set up the charge profile and discharge cutoff. **All IDs will be 0xC0500** as the address on the Inverter.

Step	Data (Command/Val)	Effect
1	B4 00 44 00	Changes to 2 stage charging (ideal for lithium)
2	B0 00 8C 0A	Changes charge current (CC) to 27.0 A
3	B1 00 54 0B	Changes charge voltage (CV) to 29.0 Vdc
4	B3 00 2C 01	Changes fully charged current condition to 3.0A
5	B9 00 B6 08	Changes battery alarm voltage to 22.30 Vdc
6	BA 00 98 08	Changes battery shutdown voltage to 22.00 Vdc

The data above is specific for our application. But the example below shows how you can determine hex values for ANY application!

EXAMPLE DETERMINATION OF DATA FOR STEP 5:

1. **Desired Outcome:**
   Change low battery alarm voltage to 22.30Vdc
2. **Find Relevant Command:**
   0x00B9

Command	Name	R/W	Size (Bytes)	Description
0x00B9	BAT_ALM_VOLT*	R/W	2	Battery low voltage alarm threshold (format: value, F = 0.01)

3. **Note:**
   (Format: value, F = 0.01) ⇒ Decimal values have a 0.01 factor between measured value and digital value. (e.g. To get 22.30 Vdc, we need a decimal value of 22.30 / 0.01 = 2230.
4. **Values:**
   (Convert Decimal to Hex - Online calculators available.)
   Battery Alarm Voltage Value: 22.30 ⇒ 2230 ⇒ **0x08B6**

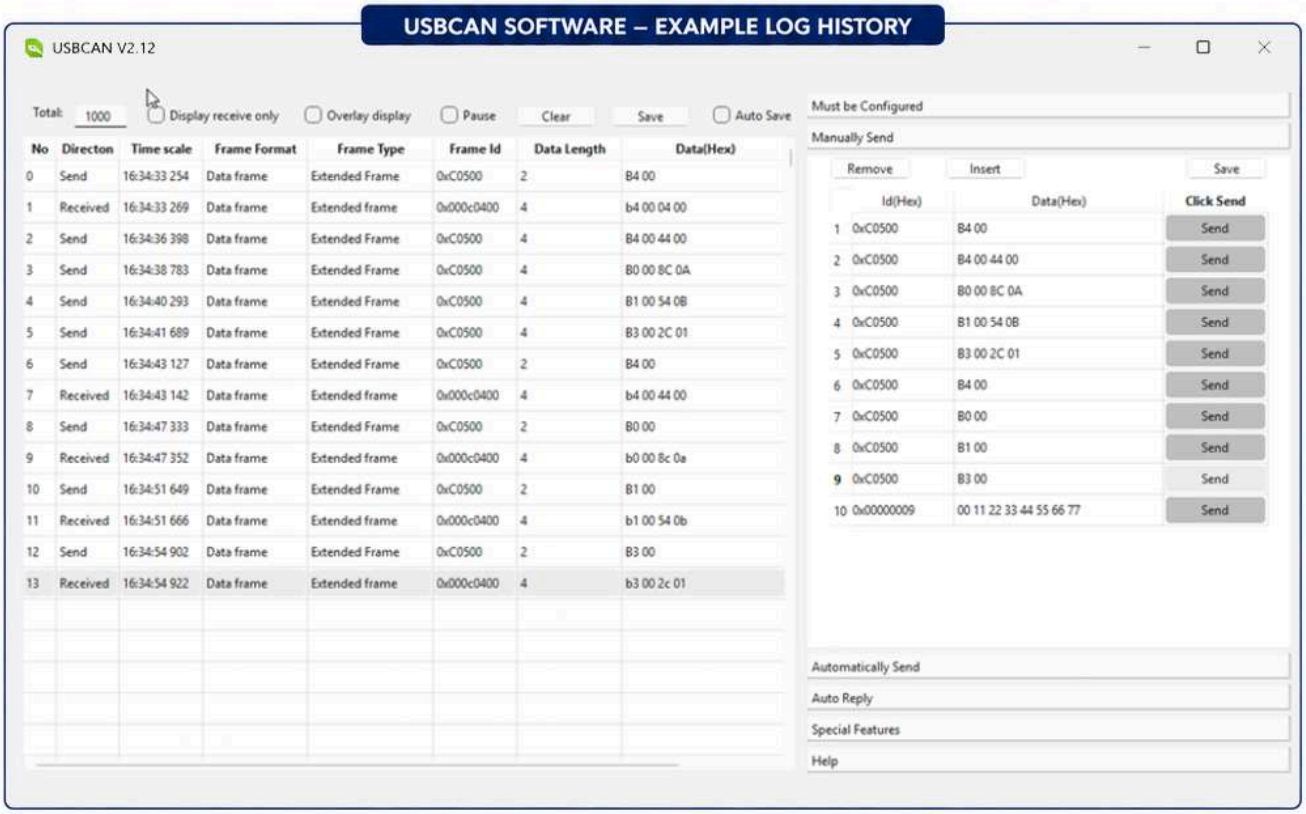
Reverse Command Bytes and Reverse Value Bytes to get Data: **B9 00 B6 08**

Once all values have been written to the NTN-5K, we can then enter all the same commands as “Reads” to ensure each value was changed correctly.

Step	Data (Command)	Expected Response
7	B4 00	b4 00 44 00
8	B0 00	b0 00 8c 0a
9	B1 00	b1 00 54 0b
10	B3 00	b3 00 2c 01
11	B9 00	b9 00 b6 08
12	BA 00	ba 00 98 08

When everything is complete, your send / receive history should look similar to the log in the image below. Note however, that this image shows a history of the charge setting changes only, and does not include the discharge cutoff changes (steps 5/6/11/12.)

USBCAN SOFTWARE – EXAMPLE LOG HISTORY



Software Step 5: USBCAN tool example log history showing the setting and reading of charge profile settings described.

5. Verification: Test rigorously!

That's it! You are now programming via CANBus! You can proceed to try various commands to glean information about your system as you proceed with prototyping. There are a variety of commands available that can help you understand the state of your electrical system. See below for some troubleshooting tips as well as resources that may be useful.

Give the Engineering Team at TRC Electronics a call to talk about your project requirements.
1-888-612-9514

A friendly reminder: It's always recommended to test your settings and design rigorously. Use a volt meter to verify voltages. Use a clamp-on ammeter to verify your charging currents. When it comes to safety, or when it comes to prototyping new hardware and software: Trust, but verify!

6. Troubleshooting

Symptom	Likely Cause	Fix
NTN-5K-224CAN Inverter won't turn on or won't charge batteries.	• Insufficient battery voltage / power available.	• Check battery voltage with a volt meter. If batteries are below Shutdown voltage of Inverter, they may need to be charged externally before reconnecting.
No communication response when sending data to PSU.	• Configuration settings or Message ID incorrect.	• Check settings – especially that "Extended Frame" and baud rate are correct. Make sure "Open" and "Set and Start" have both been clicked. Check ID matches manual.
Reading after writing shows a different result.	• Write value may have been out of range.	• Make sure the values you are sending are within the capabilities of the power supply. Also make sure bytes are being reversed: i.e. 0x08E6 => E6 08
No output or general unexpected behavior from power supply.	• Fault or Incorrect Setting	• The CANBus commands include a few which can be very helpful in troubleshooting. This includes INV_FAULT and INV_STATUS. These return bytes that when converted to binary notation can tell you what protections have been triggered or in what mode the inverter is.

7. Additional Resources

A. Partial list of CANBus commands for NTN-5K System, including all the ones used in the above example.

Command Code	Command Name	Transaction Type	# of Data Bytes	Description
0x0087	MFR_SERIAL_B0B5	R/W	6	Manufacture serial number
0x0088	MFR_SERIAL_B6B11	R/W	6	Manufacture serial number
0x00B0	CURVE_CC *	R/W	2	Constant current setting of charge curve (format: value, F=0.01)
0x00B1	CURVE_CV *	R/W	2	Constant voltage setting of charge curve (format: value, F=0.01)
0x00B2	CURVE_FV	R/W	2	Float voltage setting of charge curve (format: value, F=0.01)
0x00B3	CURVE_TC *	R/W	2	Taper current setting of charge curve (format: value, F=0.01)
0x00B4	CURVE_CONFIG *	R/W	2	Configuration setting of charge curve
0x00B5	CURVE_CC_TIMEOUT	R/W	2	CC charge timeout setting of charging curve (format: value, F=1)
0x00B6	CURVE_CV_TIMEOUT	R/W	2	CV charge timeout setting of charging curve (format: value, F=1)
0x00B7	CURVE_FV_TIMEOUT	R/W	2	FV charge timeout setting of charging curve (format: value, F=1)
0x00B8	CHG_STATUS	R	2	Charge status reporting
0x00B9	BAT_ALM_VOLT *	R/W	2	Battery low voltage alarm threshold (format: value, F=0.01)
0x00BA	BAT_SHDN_VOLT *	R/W	2	Battery low voltage shutdown threshold (format: value, F=0.01)
0x00BB	BAT_RCHG_VOLT	R/W	2	Battery recharge voltage threshold (format: value, F=0.01)
0x00BC	BAT_OV_ALM_VOLT	R/W	2	Battery over-voltage alarm threshold (format: value, F=0.01)
0x00C0	SCALING_FACTOR	R	6	Scaling ratio
0x00C2	SYSTEM_CONFIG	R/W	2	System configuration

NOTE: * Indicates commands that were used in the programming example in this document.

B. Breakdown of Curve_Config which we used to set the charger on the NTN-5K to 2-stage charging:

CURVE_CONFIG (0x00B4) :

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
High Byte	-	-	-	-	-	FVTOE	CVTOE	CCTOE
Low Byte	-	STGS	-	-	TCS		CUVS	

Low byte:
Bit 0:1 CUVS: Charge Curve Selection
00 = Customized charge Curve (default)
01 = Gel Battery
10 = Flooded Battery
11 = AGM Battery

Bit 2:3 TCS: Temperature Compensation Setting
00 = disable
01 = -3 mV/°C/cell (default)
10 = -4 mV/°C/cell
11 = -5 mV/°C/cell

Bit 6:STGS: 2/3 Stage Charge Setting
0 = 3 stage charge (default, CURVE_CV and CURVE_FV)
1 = 2 stage charge (only CURVE_CV)

STEP 4 BREAKDOWN

Received 0x0004 response from command 0x00B4.

Converting hexadecimal response to 8-bit binary:

High Byte: 0x00 = 0b00000000

Low Byte: 0x04 = 0b00000100

If you look at the above table for 0x00B4. That 1 in Bit 2 of Low Byte means TCS = 01, meaning the default -3 mV/°C cell temperature compensation is set.

Low Byte: 0x04 = 0b00000100

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	0	0	0	0	1	0	0

TCS = 01 (default)

STEP 5 BREAKDOWN

In Step 5, we then change 0x00B4 from 0x0004 to: 0x0044.

Converting hexadecimal response to 8-bit binary:

High Byte: 0x00 = 0b00000000

Low Byte: 0x44 = 0b01000100

Now Bit 6 of the low byte is also set to 1, which means the unit is set to 2-stage charging where it charges at constant voltage until cutoff current is reached, and then the charger shuts down.

Low Byte: 0x44 = 0b01000100

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
0	1	0	0	0	1	0	0

STGS = 1 (2-stage charge)

TCS = 01 (-3 mV/°C/cell default)

C. Inverter Status and Inverter Fault Command binary breakdowns. See NTN-5K Manual for more information.

INV_STATUS (0x011D)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
High byte	-	-	-	-	-	-	INV_PHASE	
Low byte	-	Bat_Low_ALM	SAVING	SOLAR_EN	CHG_ON	UTI_OK	BYP	INV

INV_FAULT (0x011E)

	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
High byte	Reserved	Reserved	Reserved	INV_Fault	Bat_OVP	Bat_UVP	FAN_FAIL	SHDN
Low byte	EEP_Err	SCP	INV_OVP	INV_UVP	OTP	OLP_150	OLP_115	OLP_100

D. Links - Datasheets, Manuals, and References

NTN-5K Datasheet:
https://files.trcelectronics.com/datasheets/ntn5k.pdf

NTN-5K Manual
https://www.meanwellusa.com/webapp/product/search.aspx?prod=NTN-5K&pdf=TIROLTVLLUUucGRm&a=4

PSL-BT-121000 Datasheet:
https://files.trcelectronics.com/datasheets/PSL-BT-121000%20G24%20M8.pdf

Waveshare USB-CAN-A:
https://www.waveshare.com/wiki/USB-CAN-A

Need Help?
Contact TRC Electronics Technical Support at 1-888-612-9514
or visit trcelectronics.com for additional resources.