INDEX

My Apology for the rough draft of this PDF. Hopefully it will get better over time (as I get time).

At this stage only one or two builders are using the Code, apart from my own Dual stage Inverter, so there may be little interest, however I found myself going back searching for simple Information that so easily gets forgotten, so this PDF is mainly for myself.

What is the differance betwen the **Single** (standard Inverter) and the **Dual** stage Inverter **Code?**

Dual version requires modifying pin A7 (see link above) which is grounded on the Controller PCB, ideally it should be done before soldering the Nano socket to the Controller PCB.

If the Controller board has already built and you wish to try this Dual code, you can still modify the small Nano PCB.

Dual code displays each Power Board Capacitor bank voltage seperatly on the LCD, it monitors both capacitor banks for precharge errors and charge voltage to correctly enable the Solenoid in a Dual Power Stage Inverter.

*Nano Inverter controller Code by myself - **KeepIS**: Using "Poidas" brilliant SPWM generator code module and incorporating symmetrical SPWM code generation scheme by "Wiseguy"*

*Circuits and boards were designed by **Wiseguy**.*

*SPWM Generator code module was written and developed by **Poida** over a number of years.*

This project was designed as a DIY Inverter with an Arduino Nano based controller, as usual in this DIY world, use all information, circuits and code at your own risk!

KeepIS, poida and wiseguy are forum user-names on thebackshed.com forum.

https://www.thebackshed.com/forum/ViewTopic.php?FID=4&TID=16761 ***LINK to the backshed Topic***

Most of the Menu settings are obvious, the few that are not will be covered fully.

1: Battery DC Calibrate: Measure the battery DC input voltage at connector J1 pin 2.

2: Capacitor Calibrate: There are 6 options for setting the Capacitor Bank voltage display:

1 Enter the Capacitor bank voltage measured at Power board +48v input terminal.
2 Restore this calibration value to default.
3 Increment the Calibration value.
4 Decrement the value.
5 Use the Battery DC calibration value.
6 Enter the Calibration value directly. (Only if you have previously written the calibration value down)

Note option 5. Using battery calibration value allows the same Battery ADC calibration value to be used for the Cap ADC, if the two voltages are not within 200mv you should calibrate the Cap bank separately.

When measuring the Battery and Capacitor voltages, the Inverter should be powered at Idle with the Kilovac relay closed or the the main breaker closed in manual wiring.

Why? The controller code is trying to get accurate running state voltages to monitor for a fault or sudden voltage drop between the battery input and the Power Board input voltage (cap bank).

5: AC relay on voltage: There are 2 options:

1 Enter the AC output voltage to control when an AC relay, if one is used, enables Inverter AC output to the load.

2 Auto activate AC relay:

Auto activate the AC enable pin, AC relay switching is enabled when Inverter SPWM reaches PID control state, this occurs at the end of the Inverter soft-start ramp up.

It does not mean that AC has reached its fully regulated voltage, it will be close, but if you are powering a load and not switching between mains and inverter, this will be fine.

However a sudden spike in Inverter Current occures when switching between Mains AC and Inverter AC if there is a differance between Mains and Inverter voltage at AC transfer under heavy loads.

Entering the voltage allows you to instruct the controller to wait until the Inverter AC has reached the desired AC voltage to match the Mains AC voltage before switching.

If your mains voltage is all over the shop then just set Auto.

8: Battery <-> Cap delta:

The Nano Controller monitors the voltage difference between the Inverter Battery connection and the Capacitor-bank voltage for any fault condition.

Simply put, it monitors the voltage drop across the DC Solenoid, it sets the maximum voltage difference for the Inverter in its running state, the Nano also monitors the “Soft Start” to “RUN” transition delta, and uses 25% of the normal cap delta value as the maximum allowable difference prior to enabling the Solenoid.

IE: If Normal Cap Delta is set to 1 volt, then maximun Delta at the end of soft start must be under 250mv.

9: Battery low-V disable: Low battery cutoff voltage, the Inverter stops until the battery increases above Low Voltage restart.

A: Battery low-V timeout: Low voltage timeout before the Inverter stops: **NOTE:** "Inverter STOPS" IF:

1: Battery voltage is below the low voltage disable voltage for Timeout seconds [adjustable 5 to 9000].
2: AC current is less than 5A and Battery voltage is below the Low voltage disable setting.
3: Battery voltage is more than 10% below Battery low Voltage disable.

B: Battery Low-V Restart:

If the battery is going flat, and voltage has dropped below low voltage cut off (menu 9) for more than xx seconds, the inverter will stop and wait for the voltage to reach a level that might indicate the battery has a higher state of charge, this voltage is the Restart voltage.

C: Battery restart delay: Time in minutes from 1 to 300 minutes (1min to 5hrs)

After low voltage disable, the time in Minutes before the Inverter will try to restart.

A message is displayed on the LCD along with the timer countdown value to inform the user.

The Countdown display appears once the battery voltage is above the restart voltage, this allows time based wait for Battery charge before attempting to ramp up the Inverter. This can be used to prevent a possible cycling condition in some battery and charging setups.

D: Battery HighV disable:

Inverter DC input over voltage shutdown, set a few volts above the maximum battery charge voltage.

K: Zero AC current offset:

This is to help with interference from Solar controllers causing the AC current to register above zero at idle, you can enter the above Zero value, and LCD AC current display shows zero until the voltage raises above the offset value.

NOTE: This option should not be used until the Zero offset adjustments in the new R7 Nano controller have been adjusted.

L: Force OCT power cycle:

This is mainly for automated power backup transfer switching between Mains and Inverter.

ON: Forces user intervention after an Over Current Trip that also causes a full Inverter power down.

1. The Inverter will respond to the OC Reset switch and restart as normal, this is considered user intervention.

2. If the inverter is powered down completely, so that the Nano has no power, on power up, if the inverter controller records that the last error was an "Over Current Trip" it will not restart until a power cycle, which can be is a quick Off-On of the inverter power switch (Auto wiring).

The LCD Message will be "Last ERR OC Restart!", which forces user intervention to press reset.

M: Fine Tune AC Frequency:

This allows the selection of available AC frequencies for mains synced clocks that run slow or fast, this may help with correct time keeping.

Press 1 to 6 in the Sub Menu to fine tune Inverter AC frequency.

```
Inverter AC Frequency adjust to
correct time sync for AC clocks

Any change requires power cycle
before Inverter will enable it

1: 50--     2: 50hz      3: 50++

 Countries that use 60hz mains

4: 60--    5: 60hz      6: 60++
```

Note: I have coded and included an "LCD" file that auto selects the baud rate when changing between 50Hz and 60Hz mains systems, simply power down the inverter and LCD. The Inverter must be fully powered down, no +5v on the LCD or Controller, any Frequency change will NOT be reflected in the Menu nor changed in the Nano controller or LCD baud rate detect code until a full power cycle has been preformed.

N: Display Calibrate values:

Displays the "Current" calibration values. These are the values that can be entered directly into the Menu setting when calibrating options are selected.

P: Restore INV E2PROM defaults: Restores all basic Voltage and Timer settings to default.

Q: Restore CAL E2PROM defaults: Restores all Calibration values to default.

R: Send LCD data over USB: Stream **s**ends the LCD buffer to the Terminal program.

S: Stream status over USB: Streams Test Code Flags to the Terminal program.

Modifying the grounded ADC pin A7 to read a voltage.

I decided to add an input for keeping an eye on the **second power board Capacitor bank** charge voltage, just in case I need the controller to hold-off closing the DC input Kilovac until both capacitor banks are at battery voltage.

Analogue input A7 is currently grounded on the Inverter Controller Board, that A7 pin needs to be isolated from ground. A 100k + 4.7k divider is fitted, just copy the same arrangement as the Vcap input divider R23 and R25 on the Controller board.

It actually works out very neatly as the Dual Inverter can utilize the spare pin 6 on J5-J6 and is coded to use the same calibration value as the existing VCap ADC in Setup. This is used in the Dual Stage Code for monitoring the second CAP board voltage for error conditions.

The results are perfect, there is absolutely no ADC cross-channel coupling with AC input current ADC on A6 or any other ADC channel.

On a single stage (standard) Inverter, if A7 is made available, it's use is completely automatic, if a voltage > 10v is applied to the resistive divider on ADC 7, the LCD displays that voltage instead of the HEX Code Version number:

IE on the Single stage standard Code Version: LCD displays "Ca2 53.2v", and once A7 is below 10v the LCD reverts to "Ver 7.5ks" so it can be used for any purpose required or just left at ground.

Inverter low battery voltage Ramp down and restart thoughts:

Obviously some of us have various ways to handle this outside of the Inverter - but not all external monitoring allows for control of a large DIY Inverter, having this setting in the inverter is common in almost every commercial Inverter I have seen, it's up to you if you want to employ it.

If an installation has not been correctly built, you will have more trouble than just the battery cut-off settings, knowing your setup and monitoring various state of charge and voltages under system loading is something we need to do and record if we plan to set it up correctly. Low voltage cut-off will be problematic if the battery to inverter DC path is not wired with appropriate gauge cable and correctly terminated at every point - no weak links allowed in any large inverter.

Low voltage ramp down and restart voltage points and the implementation of Low voltage Cut-off logic is not as simple as it appears, different loads, battery capacity and technology employed in out individual DIY off-grid Solar and Inverter builds influence the operation of this function.

Large LFePO4 bank voltage differences from charged and running under "average loads", to running at a lower SOC, is often measured in millivolts down to 40% SOC. In battery types with slightly higher internal R, the delta is often measured in volts from a running charged state down to around 40% SOC.

There is voltage sag under heavy loads, and there can be high transient startup loads or short duration increased loads, such as a fridge, freezer or an electric jug, all voltages are impacted by the current state of charge and system build type and quality.

The transients are easy to handle with a simple short timeout period for Low voltage cut-off setting, but you may want to extend the timeout period for that fridge or Jug, especially when batteries are getting down and operating near the cut-off voltage, this may not be to much of a problem with a good large LFP battery bank, but could pose a problem depending on size and SOC with older battery technology's.

I've come up with a workable solution for my Inverter and have implemented the following, these additions do not complicate the code or the operation of the controller in any way.

First, an adjustable timeout setting "In Setup" for low battery voltage cut-off, if the voltage drop is more than 10% below the normal cut-off voltage setting the timeout period is automatically reduced, if 10% below cut-off, the inverter will stop almost immediately and wait for the restart voltage set point, an adjustable restart timer "In Setup", is then implemented to allow the batteries to attain some SOC.

The restart point could be tricky if set low, and even then, with LFP and lower charge rates, or high charge rates with high current chargers, the inverter could restart with a low OR no change in SOC.

1: Battery voltage is below the low voltage disable voltage for X seconds [adjustable 5 to 9000], stop inverter.

2: Implemented "low AC current" logic for Low battery voltage under 3 amps AC, if less than 3 AAC and below cut-off, stop inverter.

3: If AC current is above 3A, and if the voltage drop is more than 10% below the low voltage cut-off, stop the inverter.

4: Once the Voltage is above the restart voltage, (charging) an adjustable Delay period "In Setup" starts a countdown before the inverter is allowed to restart and ramp up.

The delay period can be set from 1 minute to many hours.

This at least gives the batteries some time to charge, or it can simply be be used to stop a possible charge to load, low voltage trip cycle condition.

EXAMPLE of Using the Menu with a Terminal program:

To select item 1 - Battery DC calibrate

Type 1 into the Terminal Window, don't press the Enter key.

The terminal window will scroll to show the following:

1

1: Enter Battery voltage

2: Restore default Cal

3: Increment Cal value

4: Decrement Cal value

5: Enter the Cal value

Note: Pressing Enter now without entering a value will exit with no change and reshow the Main Menu.

Pressing only the option Number, each option does the following.

1. Allows entering the measured battery voltage followed by the Enter key.

2. Will immediately restore the Calibration Default value.

3. Will Increment the calibration value.

4. Will Decrement the calibration value.

5. Allows directly entering a calibration value IE 0.10119 followed by Enter.

NOTE: Entering the measured voltage [option 1] automatically calculates the calibration value and saves it, however if there is noise on the ADC input to the Nano, the calibration value could be slightly off, that is the reason for [options 3 and 4] changing the calibrating value slightly allows you to make the LCD reading for Battery voltage match the measured value.

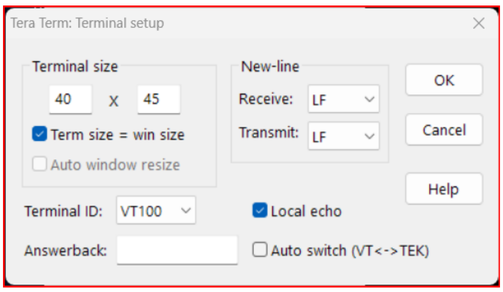
After making changes, the LCD display will immediately be updated, however the Terminal window via USB is a STATIC Menu, it's updated (redrawn) after each change, or by pressing the enter key. The Menu it a snapshot, noise could affect the readings, verify Menu values with the LCD display as this updates in real time, pressing the enter key in Terminal will also update the values with each press, but not displayed as accurately as the LCD display.

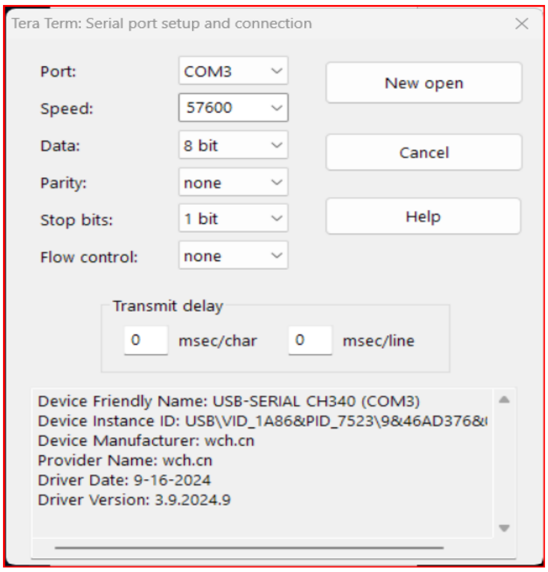
Tera Term Terminal program

Setup TeraTerm:

In Setup > Terminal > New Line for Receive and Transmit = LF

Terminal ID is VT100, Local echo can be on or off.





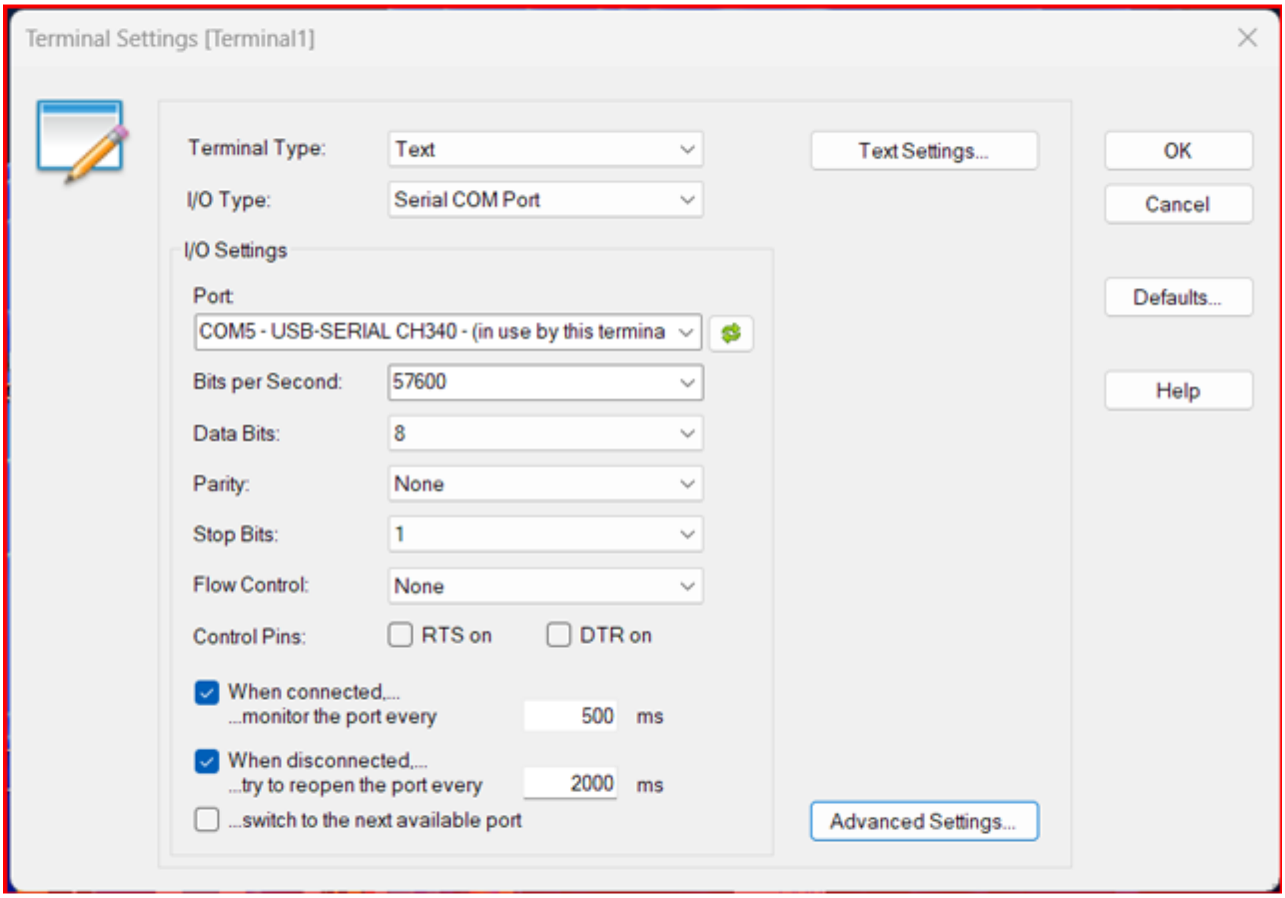
Setup > Serial Port >
Speed 57600, 8 bit, no Parity, 1 stop bit, no flow control.

YAT Terminal program Setup

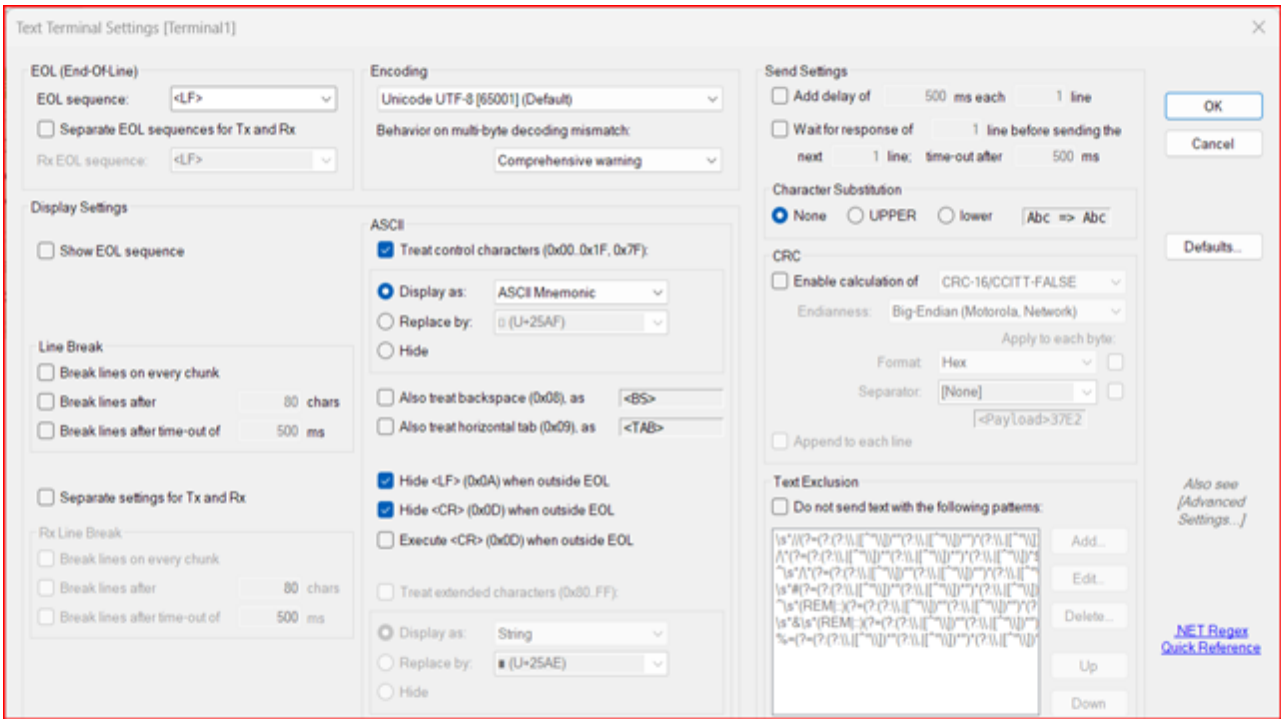
Note: I spent all of two minuted on this Terminal program, I have not used it before, just checking it worked correctly.

After this very brief look at YAT, I do like the program and will look at it in more detail when I get time.

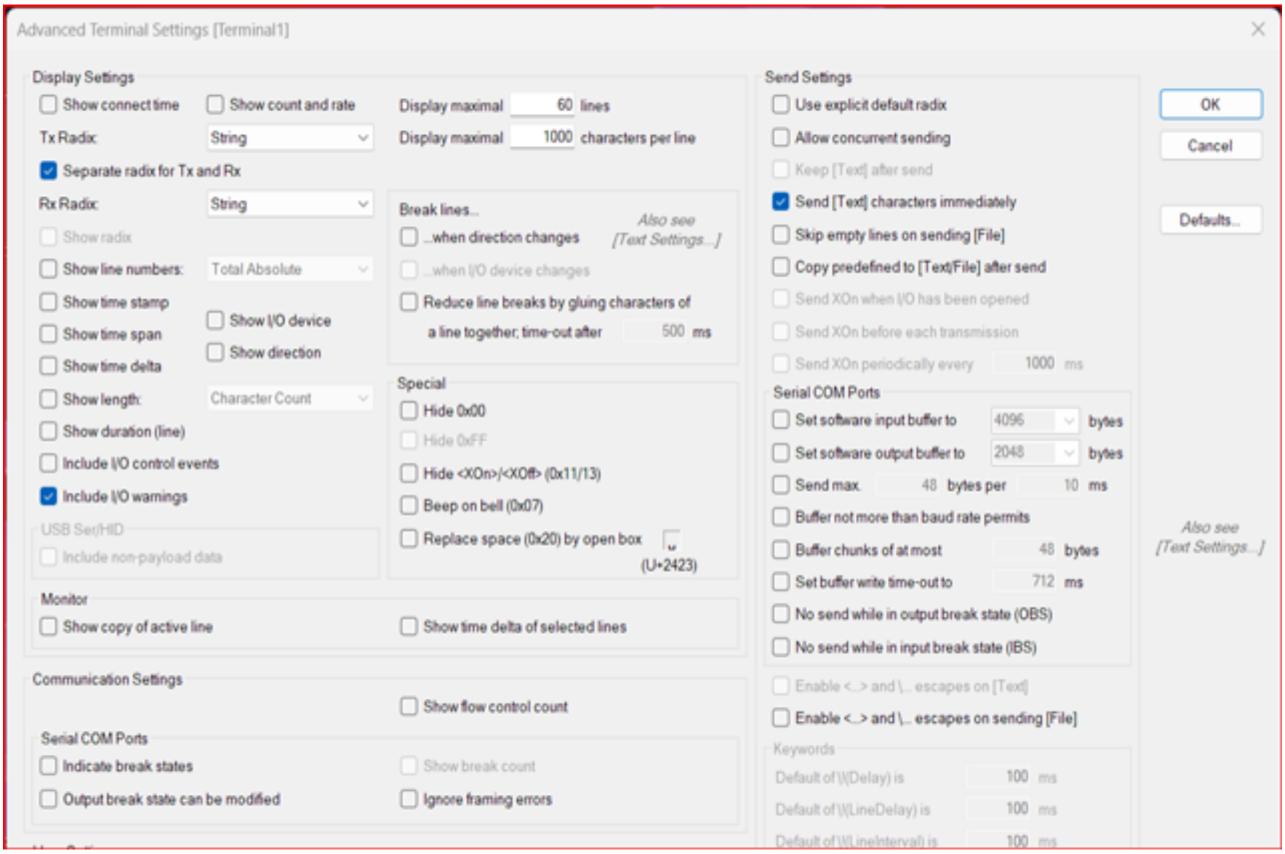
Terminal.



Text Settings



Advanced Settings.



Programmer for uploading the Inverter Hex file to the Nano:

The Arduino IDE supports these small programmer boards and the IDE can upload Hex Files, but using the Arduino IDE overwrites EEPROM data each time - yes I know it can be stopped, but it's not worth the bother.

The AVRDudess program supplied in the **Nano Setup.zip** file is a Standalone version that only needs to be unzipped into a folder to run, so you don't need the hassle of installing the full Arduino IDE application if using AvrDudess to program HEX files into the Nano.

Below is the programmer. I tested it against much more expensive boards that promised fast speed - but alas NO - This $10 unit is twice as fast.

6 to 10 pin Reference.   NANO

Programmer

The only issue is the supplied lead has to modified to plug into the Nano header, the Header on the Nano board is 6 pin and the programmer board has a 10 pin Header.

I made one up, but I found the following adapter from Jaycar, this might be an option to make it easier for some of us.

This programmer can be had for as little as $5.00

JayCar   Duinotech ISP Programmer for Arduino and AVR

Core   Core Electronics USBasp USBISP 3.3V/5V AVR Programmer

JayCar   Jaycar 10 to 6 pin Adaptor if needed.

Zadig Driver installation steps for the Programmer board:

To use the programmer board on Windows 7 or 10, 11 you need to install the libusbK driver using Zadig. **Download**

Download Zadig from the Download Link or use version 2.9 in the "Nano SetupInfo.zip" file.

1. Copy the file into a Directory and run it. **Note:** The file may need to run in Administrator mode to Install the driver.

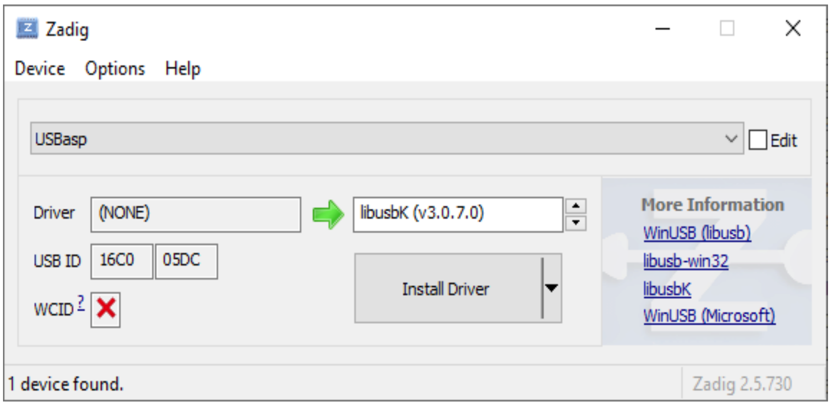
2. Connect the USBasp Programmer.

3. Select Options > Show all devices, change the main drop-down box to be USBasp.

4. You then want to change the driver by scrolling through the options until you reach the libusbK driver and click Install Driver, once installed correctly, the Programmer board will display as a **USBasp** device under **Universal Serial Bus devices** in **Windows Device Manager**.

In **USBasp > Properties > Driver** the following is shown on my system:

Driver provider. ibwdi **Driver Version**. 6.1.7600.16385 **Digital Signer.** USB\VID_16CO&PID_05DC (libwdi autogenerated)



Using AVRDudess to upload HEX files to the Nano.

1: Program uploading - Hex file.
2: Removing the Boot loader.
3: Reading and Setting Bit Fuses.

The following steps 4 and 5 are Optional.

4: Backing up your Inverter and Calibration settings.
5: Restoring EEprom setting.

The most common state for new Nano boards is with the boot loader pre-installed.

We are going to remove the bootloader, this also unlocks the protect fuses which the bootloader keeps resetting until it's removed.

The simple way has less steps and also gets the HEX Code file programmed in one go. Simply upload the Inverter code HEX file into the Nano, this also erases the flash memory at the same time.

Below is the AveDudess Screen grab, this is a stand-alone version with no install needed, just place in a directory and Run.

There are drop down lists of 100 or more Micros and Programmers: I have edited Setup and selected just the few we need:

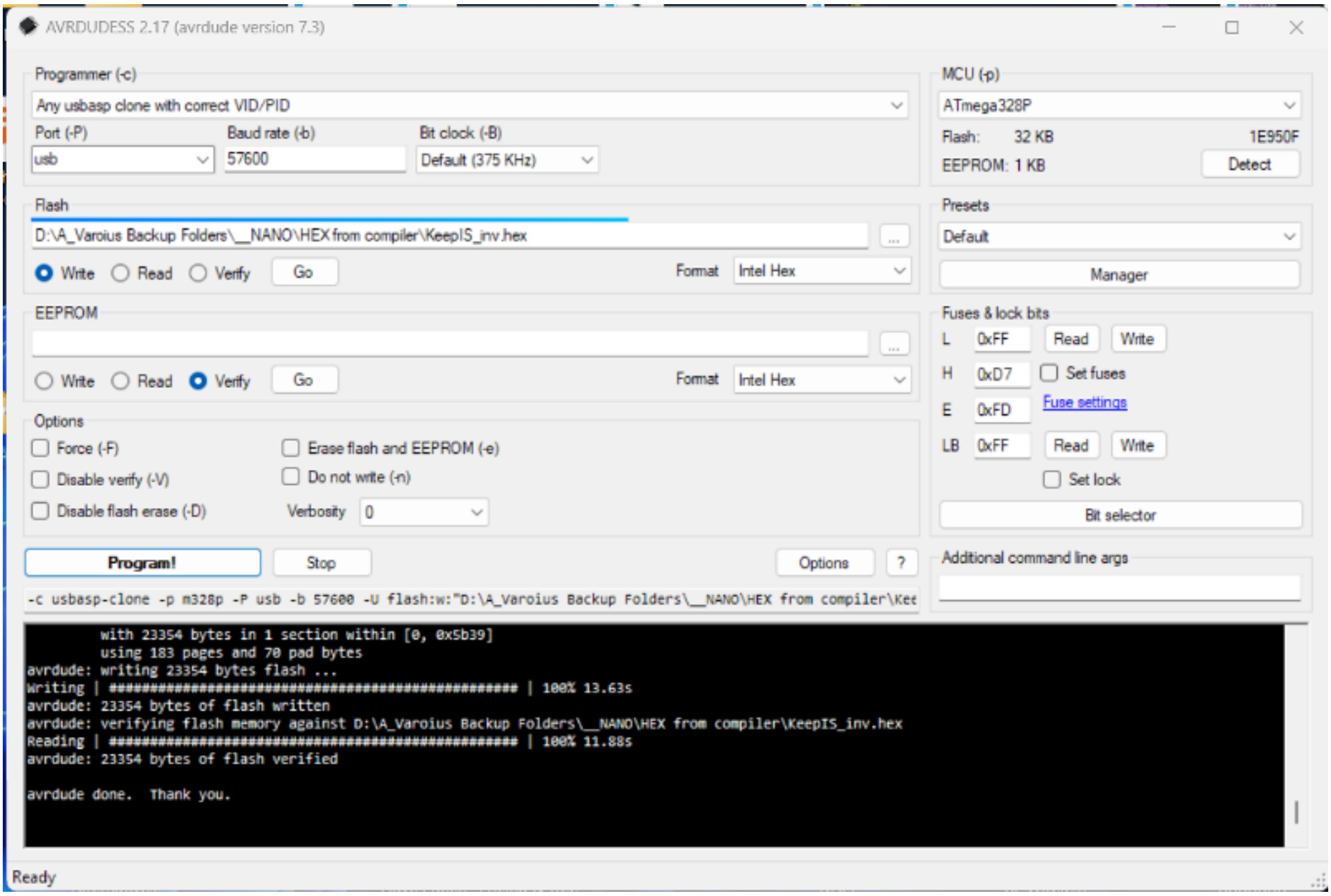
NOTE: You can click the Setup button and add or remove items from this Lists as you please.

Press the little button at the end of the **Flash** code file line [...] and select the HEX file to upload to the Nano.

The image below is how the program settings should look.

When you are ready to program the Nano, press **Program!**

The output should look like the bottom black output window in the image below.



After uploading the Hex file:

1: Press the top “Read” button under Fuses & Lock bits”

2: Press the long button “Bit selector”

The Top line in the pop up window “Lock Bits” should not have any Blue tick boxes, especially BLB12, BLB11, these were the Boot block enable bits.

Setting and Reading the Nano configuration Fuses:

Nano Fuses are “programmable nonvolatile registers”, one is used to stop EEprom data “Inverter settings”, from being overwritten, “lost” when Nano Flash is erased or when uploading New inverter HEX code to the Nano.

1: Press the lower Read button on the “Fuse & lock bits” frame.
2: Press the long “Bit selector” button. See Last post for the location:

The following image is likely what you will see for a New Nano board. The two blue boxes BLB12, BLB11, are the boot loader protect bits.

They stop that area of Flash memory from being overwritten when uploading standard Arduino code files via the boot loader and USB port in the Arduino IDE.

New Nano with boot-loader Config bits.

Below is what we want the bits to look like.

Sometimes there is something strange going on when reading and setting these Bits, so do the following to confirm it worked.

1: Read the Top "L" block
2: Set the Bit boxes to look like the last image.
3: Press "OK" to close the pop-up window.
4: Press the Top write button.
5: Press the Lower "LB" write button.

Repeat with the lower block

1: Read the Lower "LB" block.

If EESAVE is not equal to 0, do the following:

2: Set the Bit boxes to look like the last image.
3: Press "OK" to close the pop-up window.
4: Press the Lower write button.

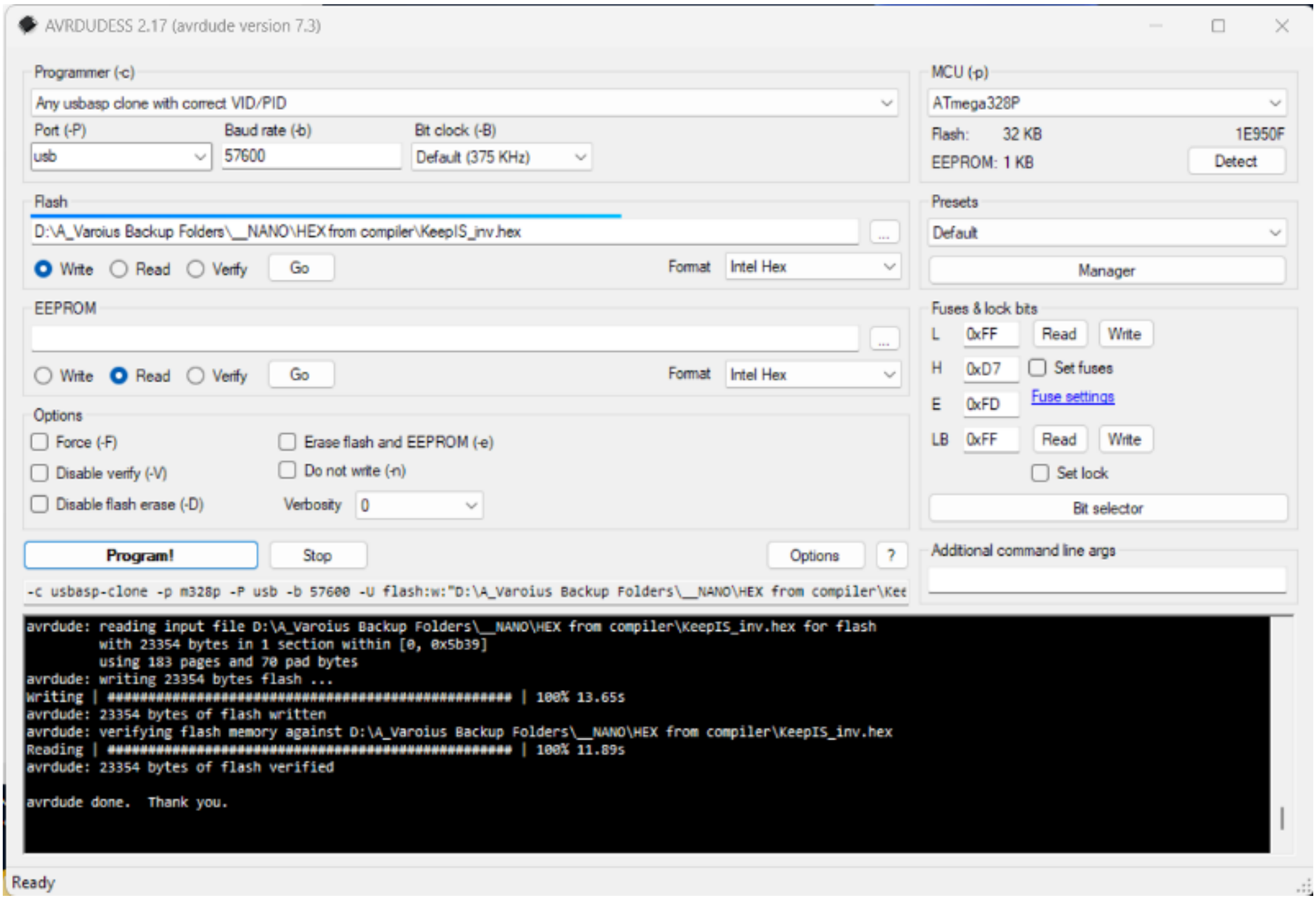
Now confirm both read the same, the Main setting we really need correct is EESAVE "it must be 0"

Backing up the Nano inverter settings:

Once you have setup the Inverter and calibrated inputs, It is a good idea to backup your settings just in case, a lot easier then having to re-calibrate and set everything up again.

NOTE: With the new NanoMmenu, there is really no need to backup EEprom settings, once the Inverter is calibrated, the calibrate values can be written down and entered into Setup with ease. So the following can be skipped.

Below is the setup screen we used to upload the Inverter HEX file, select the round "Read" option in the "EEPROM" frame.

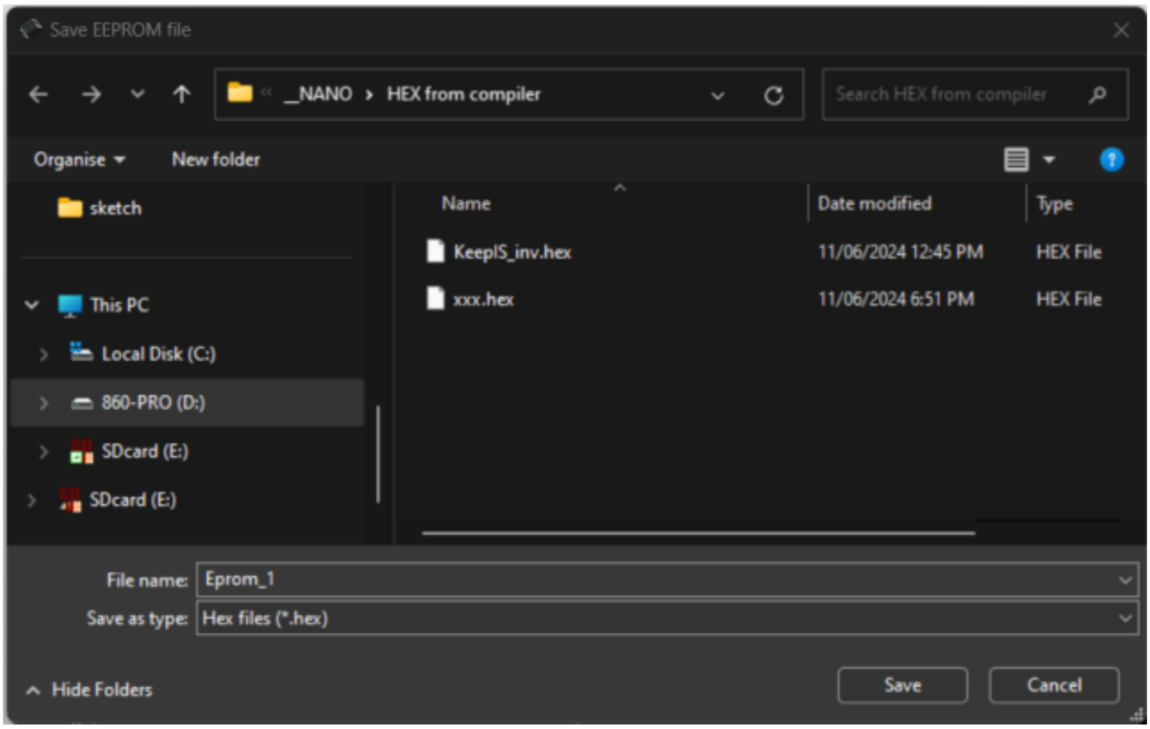


When you Click on the [...] button at the end of the EEPROM file line, you will open a window something like the following:

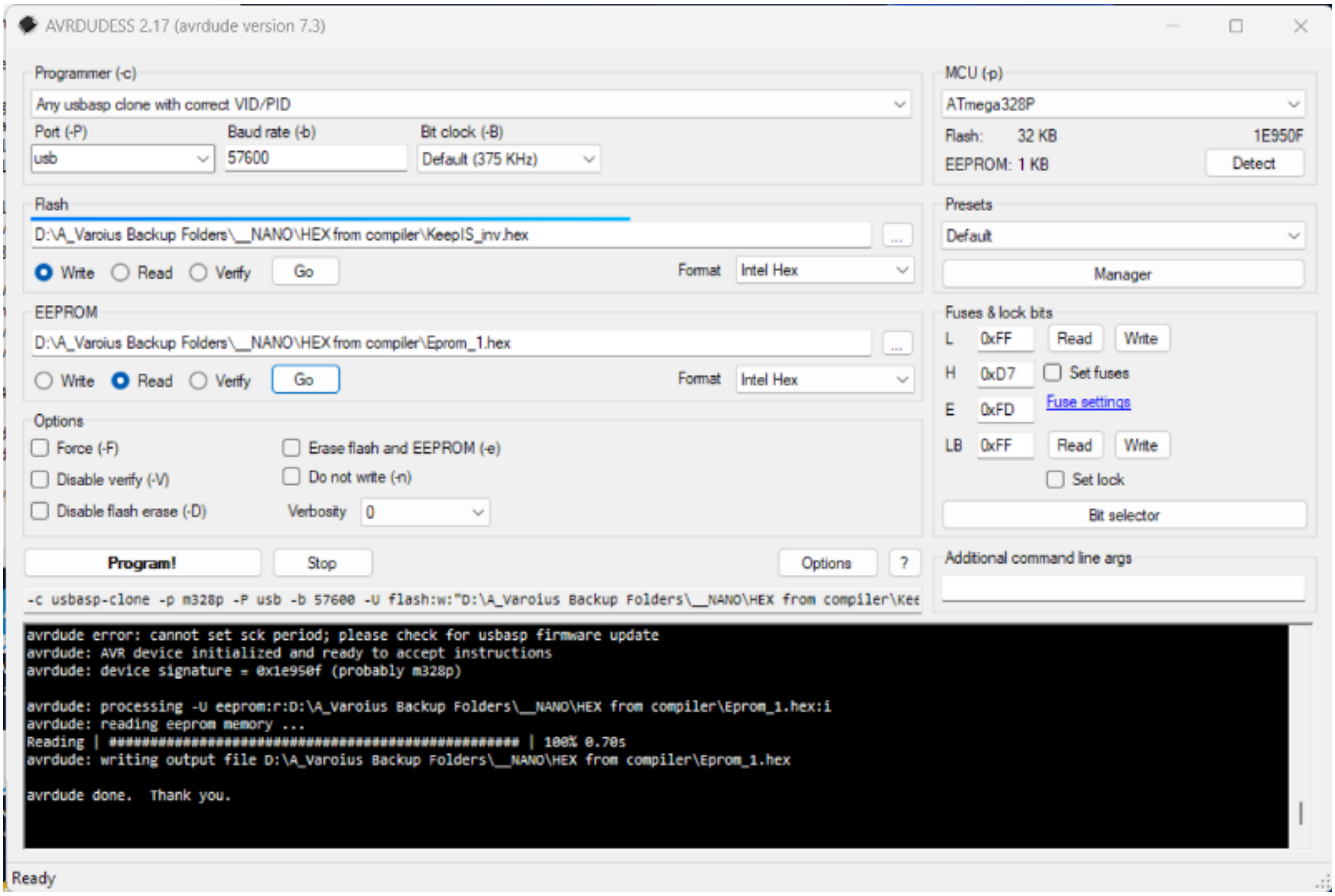
If a file exists it will be overwritten, type a new file name and it will be created when the EEprom from the Nano is read.

NOTE: You may have to use your chosen Directory viewer and create a Directory to keep the EEprom backup files in, then create an empty text file with a .HEX extension IE: "EEbackup.hex" in that directory. Load that file into ARVdudess for you first EEprom backup.

That file will be overwritten when you "Read" the EEprom. From then on the ARVdudess program will remember that directory path and file name. To makes another backup file, just select the same file and change the file name in the EEprom file name line in Avrdudess.

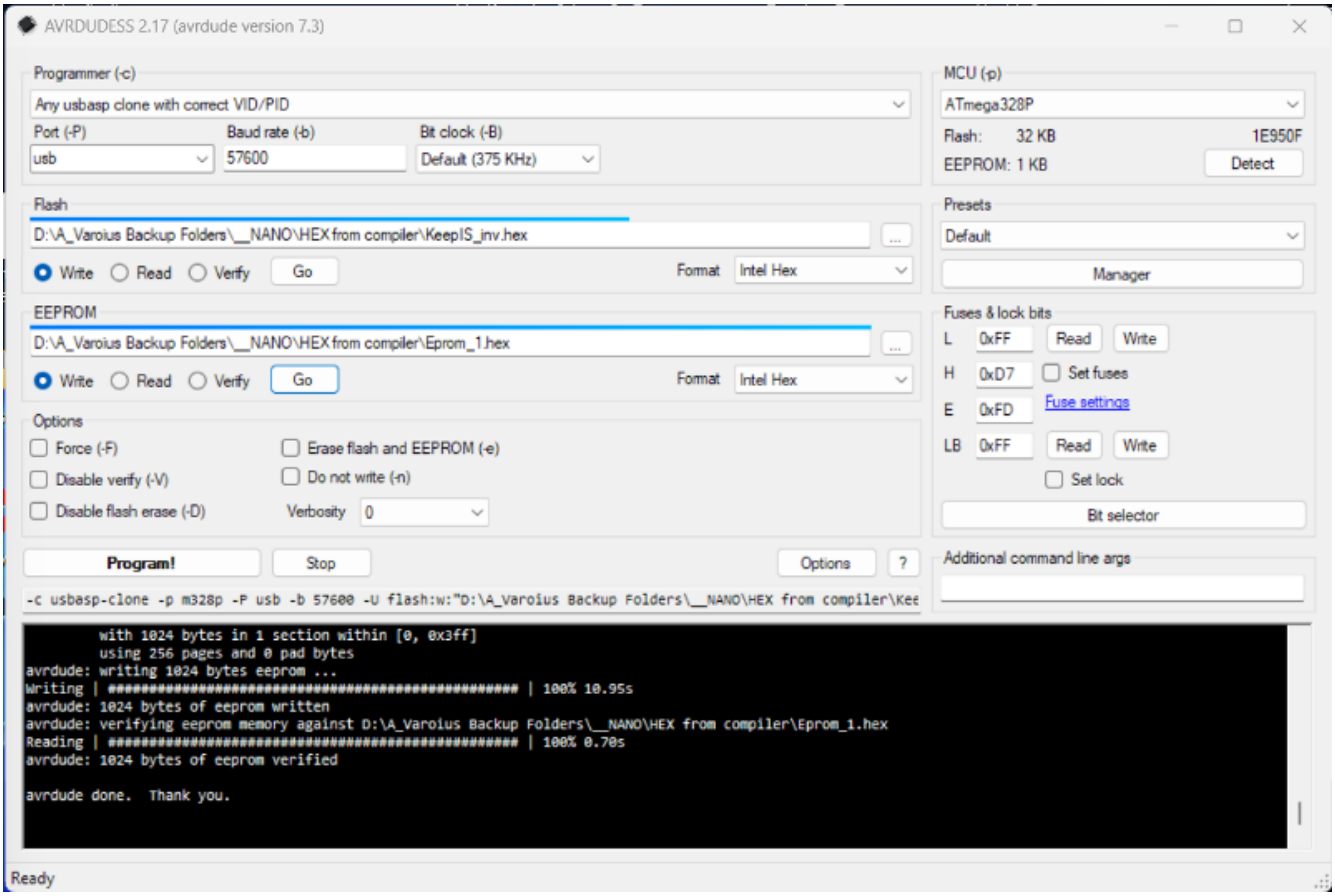


To Backup: Press the "Go" Button below the EEPROM Backup File entry line on AVRDUDESS. It's surrounded in a blue border in the Image below. This is what you should see after an EEprom read / backup, check the Black output window at the bottom.



To restore The Inverter backup EEprom file:

Same as above but Select the File to restore from, select the "Write" option below the File name line, press the "Go" button.



Restoring EEprom configuration Data does not affect the Main Code in the Nano.

Once restored you should delete the File path in from File line, this stops any accidental writing or reading to the EEprom or your

Backup file. Note: The format selection below the end of each File entry Line must always be INTEL HEX.

Stopping the USB port from Restarting the Inverter when plugging a PC into the USB port.

This is only my personal opinion and experience, but I would not put a Nano into an Inverter without USB +5v and Reset input disabled. I can safely plug and unplug my Laptop into the USB socket with the Inverter running a load, so menu changes can be done on a running Inverter without resetting the Nano and Inverter, I have a USB lead and socket from the Nano to the front of my Inverter cabinet for setup changes.

With the Reset and +5V isolated, there is no way a USB connection can interfere with the Nano +5v supply, or cause the Nano to Reset while the inverter is running.

The ease of setting up and configuring the Inverter via a PC is worth it IMHO, without removal of the Reset pathway, you will reset the Inverter every time you use the USB port or even access the port when launching a terminal program on the PC.

If you think you can set the inverter up without the Inverter running a load, then that's fine, resetting the Nano should not cause any issues with no or little load, however, I found that to accurately setup and calibrate the inverter, the final adjustments needed to be made with the Inverter running and fully warmed up.

Knowing that you can use the USB port at any time without accidentally resetting the Inverter is reassuring. Without the reset mod, if you forget and plug a PC into the USB port while the Inverter is under a high load, and you are happy to have it reset, then you just might void your warranty.

Perhaps I'm just being overly cautious.

Freeing the Nano Micro from the onboard USB controllers Power and Reset lines removes the danger of having a running Inverter suddenly reset when a USB cable and PC are connected to the Nano Board.

The Diode removed did the following:

Isolated +5v from the USB port on the Nano pcb from feeding the Nano Micro + 5V, as it normally feeds +5 to the Nano and Controller.

Removing the Diode is not a problem, we are using a small programmer board that powers the Nano during programming via the 6 pin header, when using the USB port, the Nano is powered by the inverter +5 volt line.

Removing the Reset Cap between the Nano Reset pin and the USB micro / USB port, disables the USB micro and anything on the USB port from resetting the Nano controller (Inverter). The USB controller can reset and corrupt the Nano at low voltage <3v, another reason to remove the Reset line.

Again, this USB Reset control is not needed when we program the Nano board using the small programmer board.

With these removed and with the automatic removal of the Boot loader when you program the Nano with a programmer board and Hex file makes for a really robust little board.

The USB port, when connected to a PC or Laptop, is used for setting up the operation parameters of the Inverter via the Setup Menu, it still functions perfectly and can be connected or disconnected to the Inverter while powered.

When writing code in the Arduino IDE and uploading code to test, it requires any other program using the Nano Com port, like a Console Terminal program, to be disconnected, after the upload has finished it has to be reconnected again.

It does not sound like much, but after a while it grinds on me.

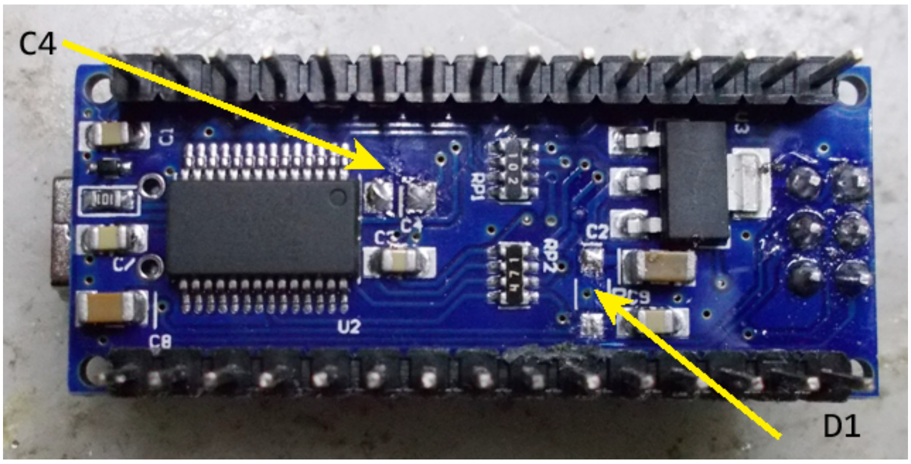
Using the small Programmer board with AvrDudess, allows you to point AvrDudess File input to the IDE compiler HEX output directory, so I build-verify the code in the IDE as normal, then Press Upload in AVRDudess, and the Hex file is simply ulpoaded.

Yes you can do this with the IDE, but there are options available to speed the upload up which are not available in the IDE - you can go behind the scenes in the IDE and make edits, but that is a pain In the A.

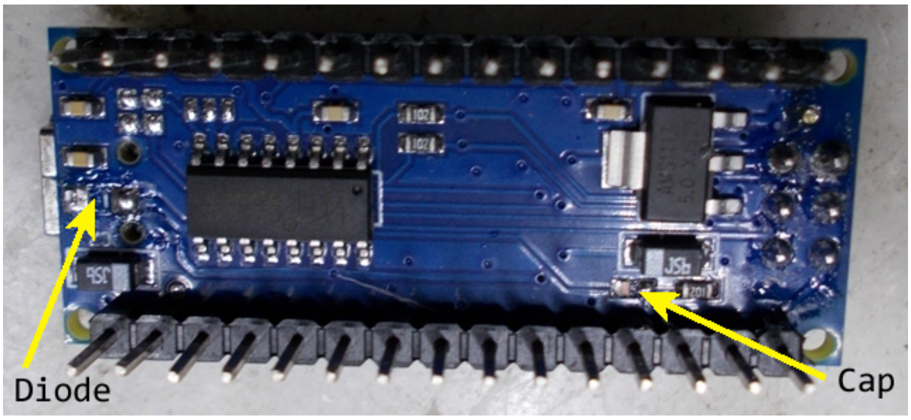
The big advantage is that the USB port on the Nano is not used to program, it remains connected the the Terminal COM program, and debug or other output streams to the Terminal program when the upload is finished.

Below: The position of the Capacitor and Diode on various Nano Boards that needs to be removed .

Board 1 with an FT232RL USB controller.

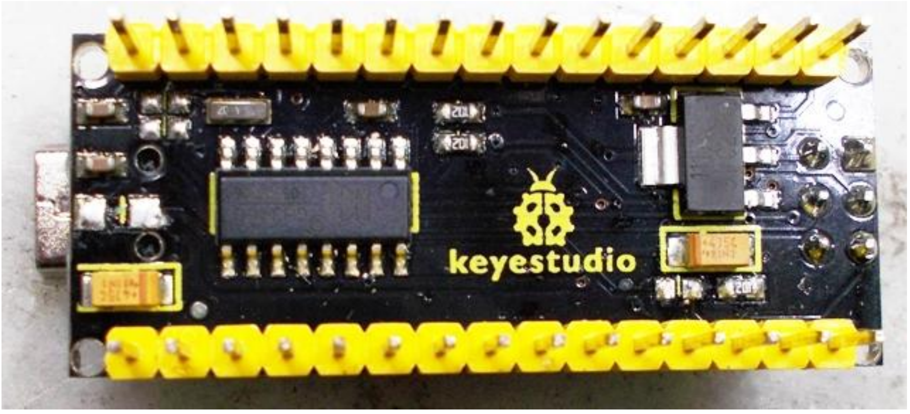


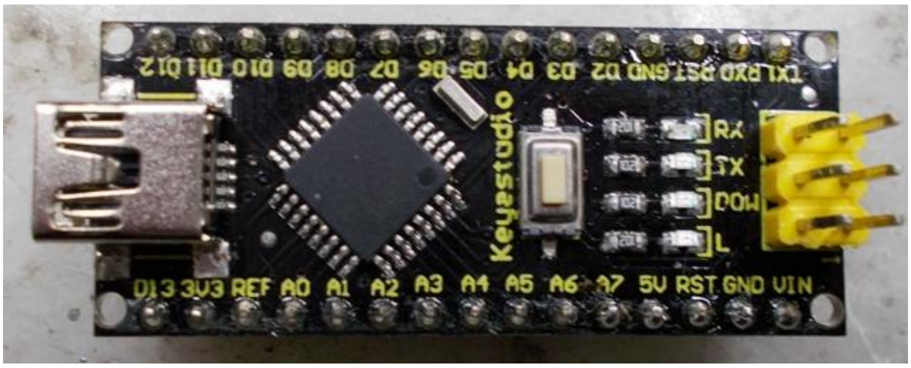
Board 2 with a CH340 USB controller.



A Nano Keyestudio, a bit expensive though. The first one I've seen with a crystal next to a CH340 USB controller.

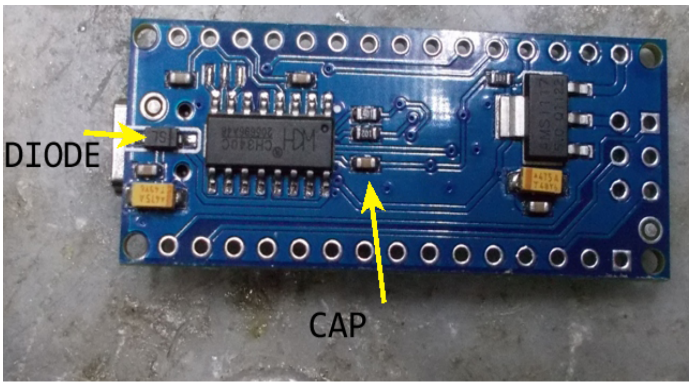
The mod is similar to the previous posted Board 2.

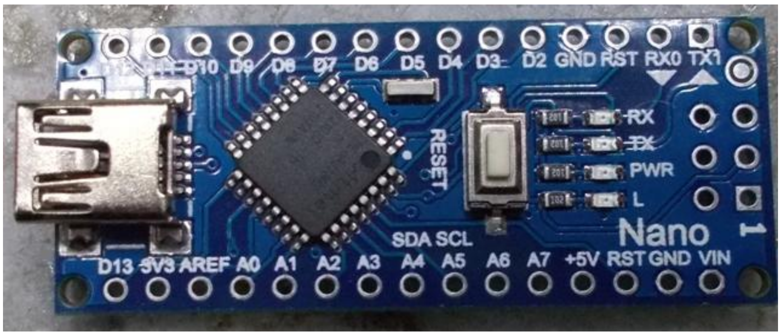




This one is a 328PB and reports a different ID number even with the PB setting, so AVRdudess reports it as unidentified and won't burn it.

You only need select the "Force(-F)" tick box to upload the Code, set fuses etc. This one came with no boot loader.

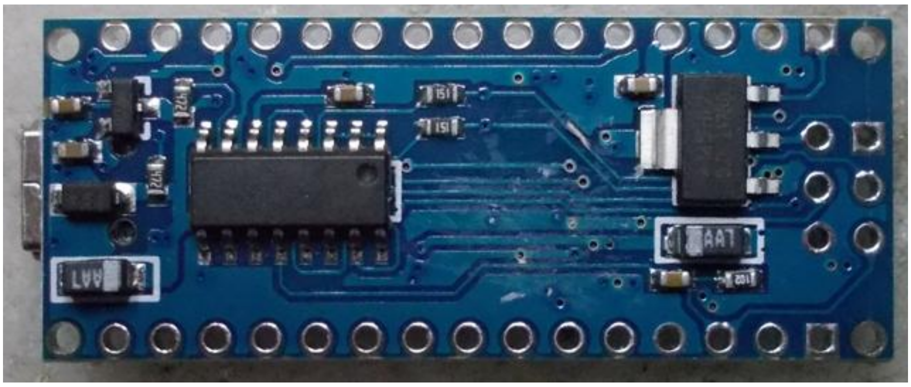


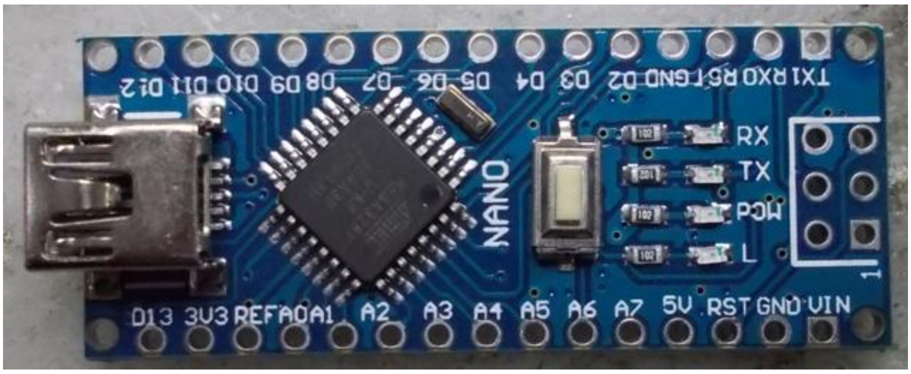


To end the Nano board roundup, I found this Nano board is well made and works, but has the older (slower) boot loader, it works fine but departs from the standard Nano with the addition of a device in the top LH corner, it may be a voltage reference for the Nano.

There appears to be no way to isolate USB +5V input from Nano +5V, if you wanted to isolate this along with the USB reset, this is not the board to use.

The link to the circuit does not show the extra components, and incorrectly shows a diode linking +5V USB to +5v Nano, this is not the case.





A note on the Nano boards:

Some clones use the MEGA328PB version of the MEGA328P, the PB version has 2 pins 3 and 6 for an extra I2C bus, pins 3 = GND and 6 = VCC in the 328P, apparently some clones with the 328PB have these wired to GND and VCC, so can't be used, as were are not setting these pins as outputs there appears to be no problem with using these boards.

I have some of the clone boards with the MEGA328PB, they all work fine:

Some Nano boards use the Old boot-loader, some use the new boot-loader, the main difference is the port speed, 115200 baud for the New boot-loader and 57600 baud for the old boot-loader.

If you can't program the HEX code into a Nano and it gives a message about not accessing the port or timing out after X retries, it may be a Nano board with the old boot loader, so change the port speed