Are-Inc CNC board

Introduction

The Are-Inc CNC board are a clone of the work by Bart Dring.
He has made a fantastic work developing the software “Grbl_ESP32” and “FluidNC”.

His work is shared as “Open Source” and this is the Are-Inc version of his work.

To get more information regarding his software and his boards you can find him at GitHub under the name “bdring”.

License

Creative Commons Attribution - ShareAlike 4.0 International
There are no commercial restrictions on using this design, but you must attribute the design to Bart Dring or @buildlog. You must also publish any changes or improvements to the design with the same open source license.

Features

ESP32 Microcontroller:

- Very Fast: Dual core 32 bit, 240 MHz
- Large Flash Memory
- Wifi:
- Bluetooth:

Grbl_ESP32 Firmware:
- Compatible with most existing Grbl gcode sender software.
- Web User Interface
- Multiple Interfaces
  USB Serial:
  Web User Interface:
  Telnet:
  Bluetooth:
  SD Card:

- (4) Stepper Motor Drivers (possible to add extra external driver for a 5th Stepepr motor)
  Possible for Motors ganging in firmware and have an auto squaring feature executed during homing.
- Limit Switch Connectors:
- Control Pins:
- Spindle Control:

User manual and help files.

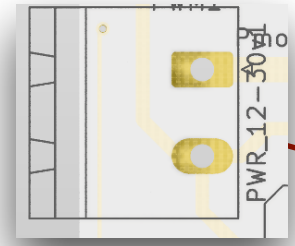
All the information to use the software is found in the FluidNC WIKI page.
(at the time of writing this the WIKI is found at this address: http://wiki.fluidnc.com/)

Power, Serial and SPI connectors

Power to the card is depending on the Electrolyte capacitors used on the board and the power converter board used. Always match the power converter and electrolyts to the power you need for your setup. The converter needs to deliver 5 volt and at least 1Amp.

The Card is delivered with a converter board supporting max 30Volt. And electrolyte capacitors rated at 25volt.

PWR_12-30v

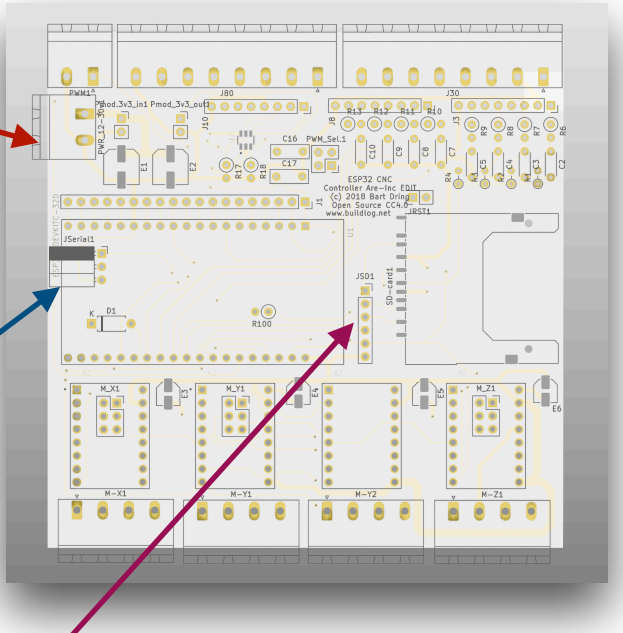


Main power connector.
1. (Square) Positive (+)
2. (Oval) Negative (-)

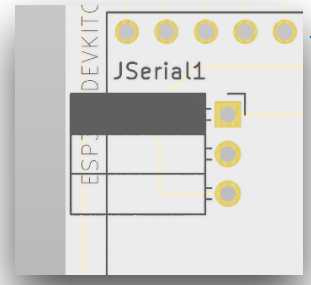
Max voltage is depends on the Electrolyte capacitors used.

MAX 30volt!

Overview controller card.



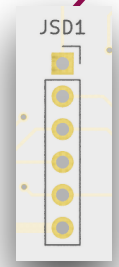
JSerial1 connector



Serial connector

1. (Black) TX
2. RX
3. Ground

JSD1 connector



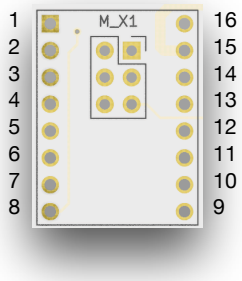
SPI connector and SD card breakout connector.

1. MISO (Square pad)
2. MOSI
3. CS “Chip select”
4. SCK “Clock”
5. 3,3volt
6. Ground

Driver chip and Stepper connections

The card uses common driver chip-modules like DRV8825, A4988

1. Enable
2. M 0
3. M 1
4. M 2
5. Reset
6. Sleep
7. Step
8. Dir



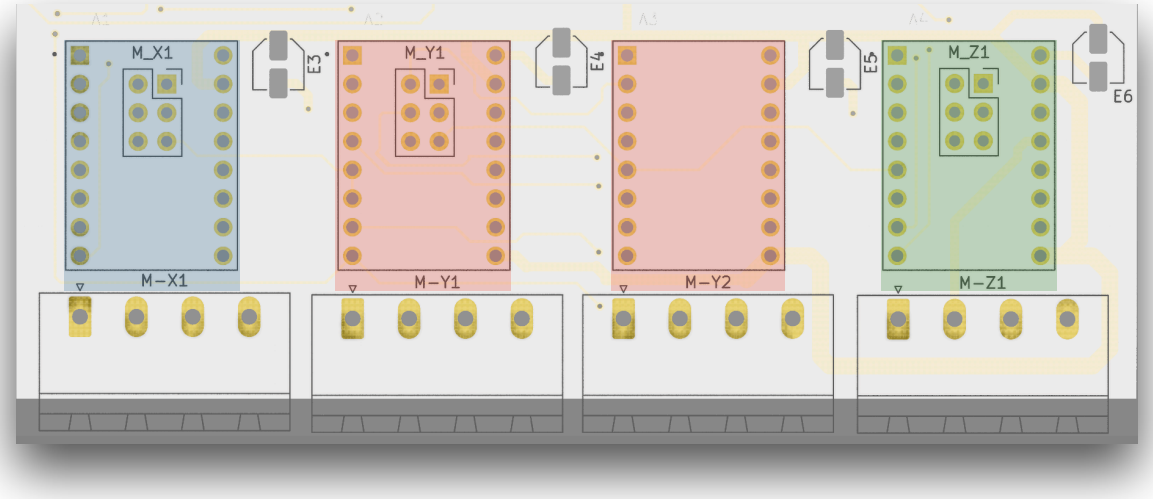
16. Voltage Motor
15. Ground Motor
14. B2
13. B1
12. A1
11. A2
10. Fault
9. Ground

Driver for X-axle stepper.

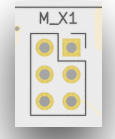
Driver for Z-axle stepper.

Driver(s) for Y-axle stepper(s).

The two driver locations on the card are for making a setup with two stepper motors driving the Y-axle. (Commonly used for router table setups)



Microstepping setup:



From the top..
M0
M1
M2

M0	M1	M2	Microstep Resolution
Low	Low	Low	Full step
High	Low	Low	Half step
Low	High	Low	1/4 step
High	High	Low	1/8 step
Low	Low	High	1/16 step
High	Low	High	1/32 step
Low	High	High	1/32 step
High	High	High	1/32 step

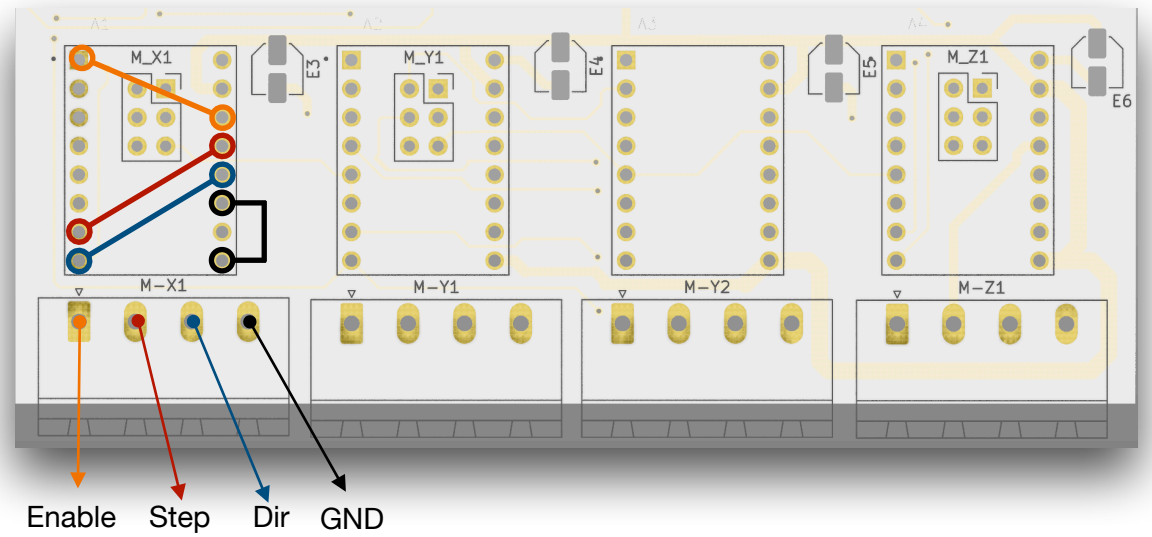
Settings for the DRV8825 driver module.

Stepper motor connector's

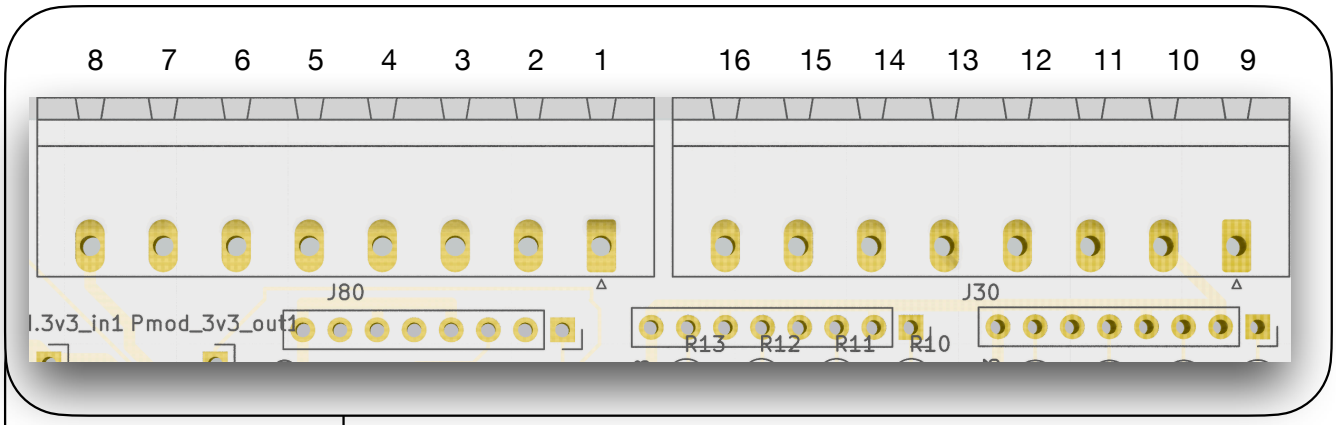


Use the CNC-card with external stepper drivers.

To use the CNC-card with external stepper drivers, there is a way to utilise the screw-terminals for the stepper motors as connectors for the external drivers. By connecting the pin holes as shown below you transfer the En, Step & Dir to the screw-terminals.

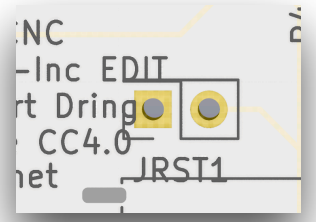


Board header pinouts.



1. X-LIMIT
2. *X-limit Ground*
3. Y-LIMIT
4. *Y-limit Ground*
5. Z-LIMIT
6. *Z-limit Ground*
7. PROBE
8. *Probe Ground*

9. DOOR
10. *Door Ground*
11. (GRBL) RESET
12. *Reset Ground*
13. START
14. *Start Ground*
15. HOLD
16. *Hold Ground*

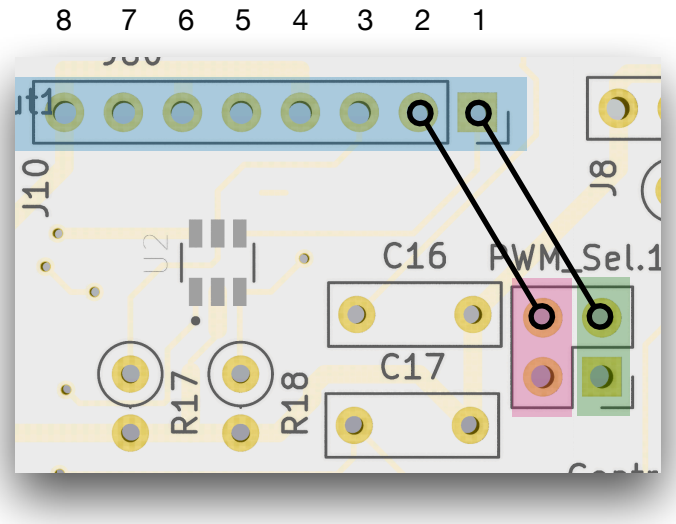


JRST1 connector:
This is the ESP32 “Enable (en)” pin that is lifted out to a pin terminal for making easy access to ESP32 reset when doing upgrades or other work that need a “Reset” of the MCU.

1. Spindle PWM 5-volt
2. Spindle PWM 3,3-volt
3. Spindle Enabled
4. Spindle ground
5. MIST
6. Mist ground
7. FLOOD
8. Flood ground

Join these pins to set PWM output to 3,3volt terminal pin.

Join these pins to set PWM output to 5volt terminal pin.



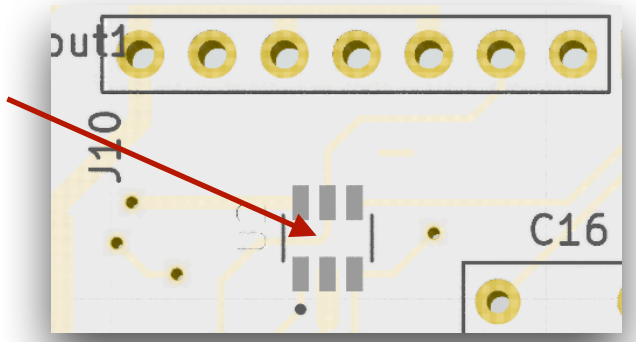
As the connecting lines in picture illustrate, one can use these pins to connect other things to the two “screw terminal” outlets. **But make sure that you have no connection to the lower tabs at the same time!!!!**

Spindle controller TXS0101

The empty slot on the board, marked “U2” is for a Texas Instrument “TXS0101”
TXS0101 is a “*1-Bit Bidirectional Level-Shifting, Voltage-Level Translator With Auto-Direction-Sensing for Open-DrainandPush-Pull Applications”*.

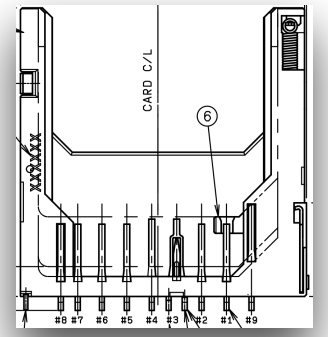
The use on this board is to translate the PWM Spindle signal to 0-10volt for use with external spindle controllers.

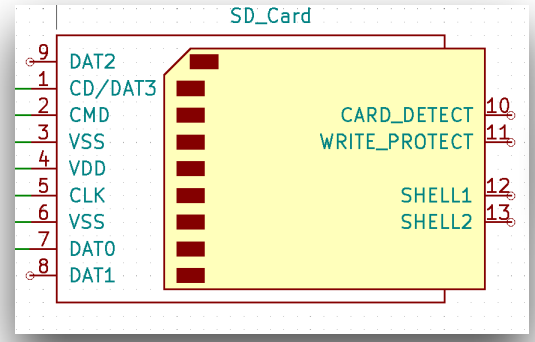
TXS0101

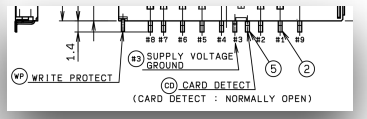


SD-Card slot on the board

There is a slot for a SD-card holder on the board. The holder is a Kyocera 145638109511859+.



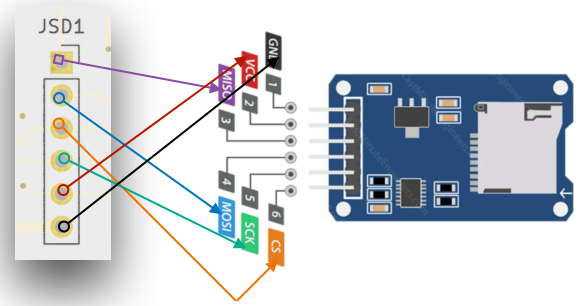




External SD-card module
There are several module boards for SD-cards, the pin-header JSD1 can be used for connecting to those boards.

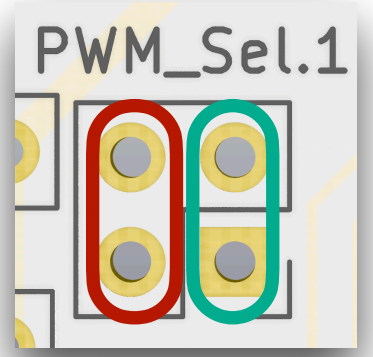
The JSD1 pin.header only support 3,3volt modules. If you have a 5volt module you need to use 5volts elsewhere from the main board, and use “Level shifter” for the controll signals.

(Easiest solution is of course to use a 3,3volt module, where the connections will be straight forward.)



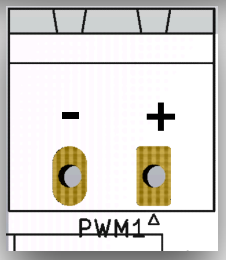
Connecting Spindle or Laser

There are a few options here and we quickly take some examples here.



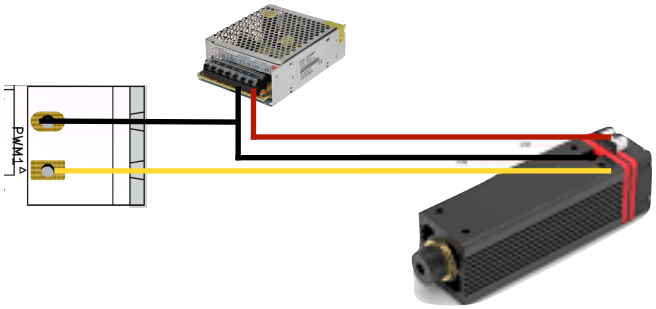
Using a “bridge” between the pads marked with **RED** gives you A 3,3 volt PWM signal.

And a “bridge” between the pads maeked with **GREEN** activete The 5 volt PWM signal.
(To get the 5 volt output you also need to Have the TXS0101 ic mounted with the resistors and capacitors That belongs to that circuit.)

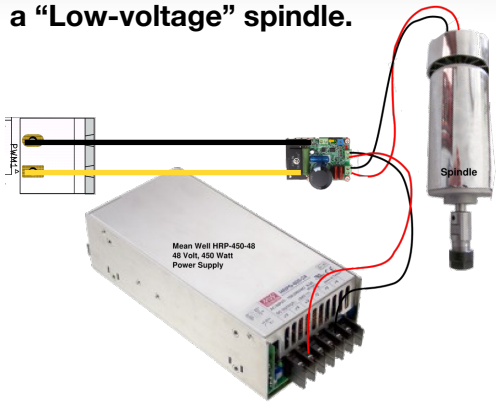


The final output is routed to the PWM screw terminal marked with “PWM1” in the corner of the board.

Connecting a Laser module.



Connecting a “Low-voltage” spindle.



Connect a VFD-spindle.

See the Online WIKI page to read more regarding the different VFD-spindles supported directly by this card/software.

0-10V
BESC
VFD (RS485)
H2A RS485
Huanyang RS485
YL620 RS485
H100 RS485
P2 Series RS485
NowForever RS485

Config.yaml entries

Here is a list with the cards
Hardwired GPIO pins:

```
axes:
  shared_stepper_disable_pin: gpio.13:low

  x:
     limit_neg_pin: gpio.17
      step_pin: gpio.12
      direction_pin: gpio.14

  y:

     limit_neg_pin: gpio.4
      step_pin: gpio.26
      direction_pin: gpio.15

  z:

     limit_neg_pin: gpio.16
      step_pin: gpio.27
       direction_pin: gpio.33

spi:
  miso_pin: gpio.19
  mosi_pin: gpio.23
  sck_pin: gpio.18

sdcard:
  cs_pin: gpio.5
  card_detect_pin: NO_PIN
  frequency_hz: 0
```

(default of 20000000 (20MHz) try 10000000 if you have problem connecting to the SD-card)

Both driver for the Y axes on the board share the same DIR & STEP pins. Same with the LIMIT switch pin.

There are several unused pins on the bord and there is room to add both DIR & STEP separately for the drivers and also for extra LIMIT switch , to make use of "auto squaring" of the Y axle.

```
coolant:
  flood_pin: gpio.25:low
  mist_pin:  gpio.21:low

control:
  safety_door_pin: gpio.35:low
  reset_pin: gpio.34:low
  feed_hold_pin: gpio.36:low
  cycle_start_pin: gpio.39:low

probe:
  pin: gpio.32:low

Laser, Spindle, Pen (or other):
  output_pin: gpio.2          (PWM or On/Off according to setup.)
  enable_pin: gpio.22         (Set as high or low according to setup)
```

Spindle 0-10v setup:

Use of an external VFD with a 0-10v control signal.
The YAML entry “speed_map” is used to map a certain output voltage to a specific RPM range.

Ex:
A spindle that can produce 24000rpm at maximum, but struggle to perform usable power below 1000rpm.

speed_map: 0=0.000% 1000=0.000% 24000=100.000%

The spindle speed or power is 0% until S=1000, then ramps up linearly to full speed when S=24000, remaining at full speed if S>24000.

```
10V:
  pwm_hz: 5000
  output_pin: gpio.2
  enable_pin: NO_PIN
  direction_pin: gpio.22
  disable_with_s0: false
  s0_with_disable: true
  spinup_ms: 0
  spindown_ms: 0
  tool_num: 0
  speed_map: 0=0.000% 1000=0.000% 24000=100.000%
  off_on_alarm: false
```

Relay or Off/On Spindle/Laser:

You can specify that any non-zero S value turns on the spindle with
speed_map: 0=0% 0=100% 1=100%
The transition from off (0%) to full-on (100%) happens just after 0.

Fully Linear Spindle:

speed_map: 0=0% *10000*=100%
(Where “*10000*” is the maximum rpm for the spindle…)

Laser:

```
Laser:
  pwm_hz: 5000
  output_pin: gpio.2
  enable_pin: NO_PIN
  disable_with_s0: false
  s0_with_disable: true
  tool_num: 0
  speed_map: 0=0.000% 255=100.000%      (speed_map is used to set power on laser)
  off_on_alarm: true
```

More information how to setup the different spindle’s/VFD/Laser/Pens a.s.o. check the “**FluentNC**” wiki page. Here you find an extensive manual covering everything in the software.

Setup SD-card

Supported SD Cards

MMC cards are not supported.

2 GB SD cards are not supported, because they use a different internal block size compared to all other sizes.

SD cards 64 GB and larger are supported only if they are reformatted into FAT-32 format. The factory format on large cards is ExFAT, which the libraries that we use do not handle.

Card Formatting
FluidNC uses third party libraries to access SD cards. In general, the smallest, oldest and slowest cards tend to work best. As cards get larger, more and more RAM memory is needed to keep track of where files are located. That can be a problem on an ESP32, whose RAM is limited compared to, say, PCs.
Some people have trouble when SD cards have been formatted by Windows, but were able to solve the problem by formatting with the software “SD Card Formater”.
The filesystem format must be FAT-32. ***(ExFAT is not supported)***.

Modern cards larger than 32GB usually come with ExFAT. To use such a card, you must use a partitioning tool to create a partition no larger than 32GB and format that as FAT32.

File Names
The maximum length of a filename or subdirectory name is 30 characters. That limit applies to each component of the path. The overall path can be longer, but it is best to keep the overall length down to 100 characters or so, lest it exceed the length of a command line or some other internal buffer.

File Sizes
FAT-32 has a limitation of a maximum individual file size of 4GB. That should not be a problem, as an individual GCode files rarely exeeceds a few megabytes.

Help & Troubleshooting
Getting responses with "*No SD Card*" Check your startup messages with $SS to see if the card has any configuration errors or other information about the SD Card setup.

If it does not work with your DIY setup or adapter..
SD cards are very layout (connections) and timing sensitive. Many people have reported that their hand made circuits do not work. Some of these will work with example programs but not FluidNC.

If you have problem initiate the SD card, you can try to lower the frequency_hz: setting to 10000000 or lower.

SD-card Config File settings.
We need to setup both “SPI” and the actual SD-card in the YAML config file.

spi Section
Here is the config file section for spi:

spi:
miso_pin: gpio.19
mosi_pin: gpio.23
sck_pin: gpio.18

It is recommended that you use the default pin numbers, because these have been thoroughly tested, but other pins should work.

Sd-card Section
You need an SD-card section with at least the cs_pin defined. The card_detect_pin is supported, but there are no features associated with it other than showing it in the startup messages.
So there is no real reason to use a separate pin on the MCU just to send that text message.

Here is the config file section for SD-card:

sdcard:
cs_pin: gpio.5
card_detect_pin: NO_PIN
frequency_hz: 20000000

Analog Pins

analog0_pin:
Type: Pin
Range: gpio
Default: NO_PIN

Details: A PWM signal is output on this pin. It is controlled by the M67 command. M67 E0 Q23.87 would turn on analog0 with a 23.87% percent duty cycle. M67 E0 Q0 would turn off analog0.

analog0_hz:
Type: Integer
Range: 1 to 20000000
Default:

Details: The frequency of the PWM signal.
The analog signal can be used to control RC Servos. Consider a 50Hz (typical) servo with pulse range of 1ms to 2ms. 50Hz has a period of 20ms, so 5% is 1ms and 10% is 2ms.

Digital Pins

digital0_pin:
Type: Pin
Range: gpio or i2so
Default: NO_PIN

Details: The output is on this pin. It is controlled via M62, M63, M64 and M65 commands. M62 P0 Would turn digital0 pin on. M63 P0 Would turn digital0 pin off. Like all output pins, you can set the active state with the :high or :low attribute.

Config Examples:
(set the number for your pin instead of the "xx" in gpio.xx)

```
user_outputs:
  analog0_pin: gpio.xx
  analog1_pin: gpio.xx:low          (Here we have set the pin active state to "low")
  analog2_pin: NO_PIN
  analog3_pin: NO_PIN
  analog0_hz: 5000
  analog1_hz: 5000
  analog2_hz: 5000
  analog3_hz: 5000
  digital0_pin: gpio.xx
  digital1_pin: gpio.xx
  digital2_pin: is2o.xx
  digital3_pin: is2o.xx:low         (Here we have set the pin active state to "low")
```

Control Inputs

This section is used for control inputs. These are typically used with switches.

Initial State
All control inputs must be in the non active state at turn on. This is to prevent you from using a machine with a stuck switch. The active state can be changed using the high/low attributes.

Config example:
(Add your "Pin number" where it now state "NO_PIN")

```
control:
  safety_door_pin: NO_PIN
  reset_pin: NO_PIN
  feed_hold_pin: NO_PIN
  cycle_start_pin: NO_PIN
  macro0_pin: NO_PIN
  macro1_pin: NO_PIN
  macro2_pin: NO_PIN
  macro3_pin: NO_PIN
```

Probe

Config example:

probe:
pin: gpio.12
check_mode_start: true

Setup
Make sure the pin has the right attribute for active high or low.

Using the probe
Basic probing uses a ***G38.2*** command to probe to a switch or electrical contact. You can adjust the active state of the signal with the high/low attribute.

If you set check_mode_start: true it will verify the probe is not touched before the move.
If it is you will get an alarm 4

If the probe move completes without activating a probe input it will fail with an alarm 5.

If the probe is successful it will issue a message like this [PRB:151.000,149.000,-137.505:1] with the machine position at the time of the touch specified.
The 1 at the end specifies a successful probe.

After touching, the machine will decelerate to a stop. This means the machine location will differ from the location in the message after the probe. It is usually only a tiny amount, but you should account for it.
If the probe fails, it will issue a message like this [PRB:0.000,0.000,0.000:0]. The 0 at the end indicates a failed probe.

You need to specify move axis parameters and a feedrate.
G38.2 Z-5 F200
This would probe towards the work position Z-5 with a feedrate of 200.

G38.2 G91 Z-5 F200
The G91 parameter will cause it to move down in Z by 5 with a feedrate of 200.
Note: Be aware that G91 will persist and affect subsequent gcodes. You should reset it to G90 if that was the previous mode.

You can probe towards any point in any or multiple axes at once. Soft limts will be respected. If the command requests a maximum travel that exceeds the range and soft limits are true for the axis, you will get an alarm.

Setting Work Zero
The value in the PRB message is the exact point where the probe activates. The machine will then decelerate to a stop. The machine will no longer be at the probe point even if the probe is going slowly. The best way to set a work zero is to use the ***G10 L2 Px*** command with **x** being the coordinate system.
P0 would be the current system. P1 through P6 would be used to specify G54-G59.

To set the current work Z to 0 after getting [PRB:151.000,149.000,-137.505:1] send ***G10 L2 P0 Z-137.505***, using the z value from the PRB message. If you have a plate or other offset, just add that to the PRB value.

Optional P parameter

You can use an optional "P" parameter that specifies the thickness (or offset from 0) of the probe device, like a plate or puck.

Upon successful probe the offset of the current system is zeroed with the P value applied. This will makes it much easier for displays and senders to deal with probing.

Example: ***G38.2 G91 F80 Z-20 P8.00***

This will probe an incremental (G91) amount of -20 in Z. It will set the probe location to 8.00 on the Z axis in the current work coordinate system.

Notes:
After the probe touch, the status will show the axis location to be a little off from the probe location. This is due to the deceleration after the probe touch. The probe location is accurate to the actual touch point. If you want to minimize the the over travel use a lower speed (best) or faster acceleration (in your config file).
You can only specify movement in one axis when you use the P parameter.







Index:

Are-Inc CNC board 1

Introduction 1

License 1

Features 1

User manual and help files. 1

Power, Serial and SPI connectors 2

Driver chip and Stepper connections 3

Use the CNC-card with external stepper drivers. 4

Board header pinouts. 5

Spindle controller TXS0101 6

SD-Card slot on the board 6

External SD-card module 6

Connecting Spindle or Laser 7

Connecting a Laser module. 7

Connect a VFD-spindle. 7

Connecting a “Low-voltage” spindle. 7

Config.yaml entries 8

Hardwired GPIO pins: 8

Spindle 0-10v setup: 9

Relay or Off/On Spindle/Laser: 9

Fully Linear Spindle: 9

Laser: 9

Setup SD-card 10

Supported SD Cards 10

Card Formatting 10

File Names 10

File Sizes 10

Help & Troubleshooting 10

SD-card Config File settings. 11

spi Section 11

Sd-card Section 11

Analog Pins 11

Digital Pins 12

Control Inputs 12

Probe 13

Setup 13

Using the probe 13

Setting Work Zero 13

Optional P parameter 14