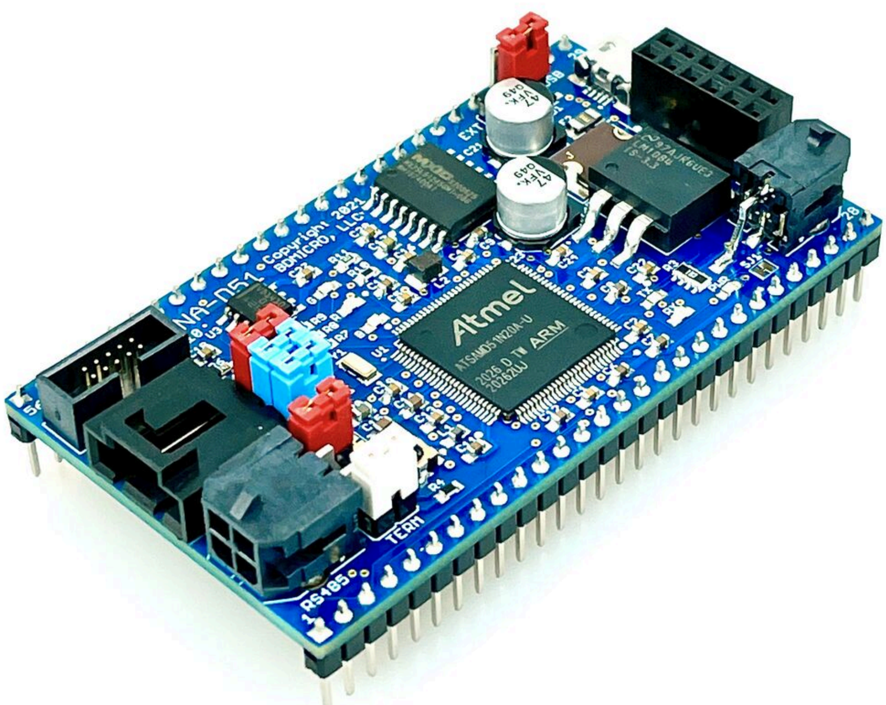
BDMICRO VINA-D51

ARM Cortex M4 Core Module

Technical Manual

Conceive. Design. Do.

Control your world.



BDMICRO, LLC
https://www.bdmicro.com/
January 4, 2022

Contents

List of Figures

List of Tables

1 DISCLAIMER

BDMICRO, LLC products are designed and assembled with care and all assembled products must pass functional testing to ensure quality before they are shipped. However, our products are not rated for use in medical, life support, or systems where failure could result in the loss of life or have serious life threatening or economic impact.

BDMICRO, LLC does not warrant merchantability for any purpose and shall not be liable for any direct, indirect, incidental, special, exemplary, or consequential damages of any kind, however incurred, through the use of our products.

2 Quick Start

Getting started with the VINA-D51 is quick and easy. It comes fully loaded with the UF2 Bootloader and CircuitPython. A sample program is pre-installed that flashes the on-board red, green, and blue LEDs (`LED_R`, `LED_G`, and `LED_B`), in a sequential pattern followed by a short simultaneous flash of all three to let you know everything is working as expected. You'll also note the green `PWR` LED to indicate the board is powered.

Simply:

1. Set the `PWR_SEL` jumper to `USB`.
2. Plug a USB micro-B cable into the USB port of VINA-D51.
3. Plug the other end of the USB cable into your computer or USB hub.

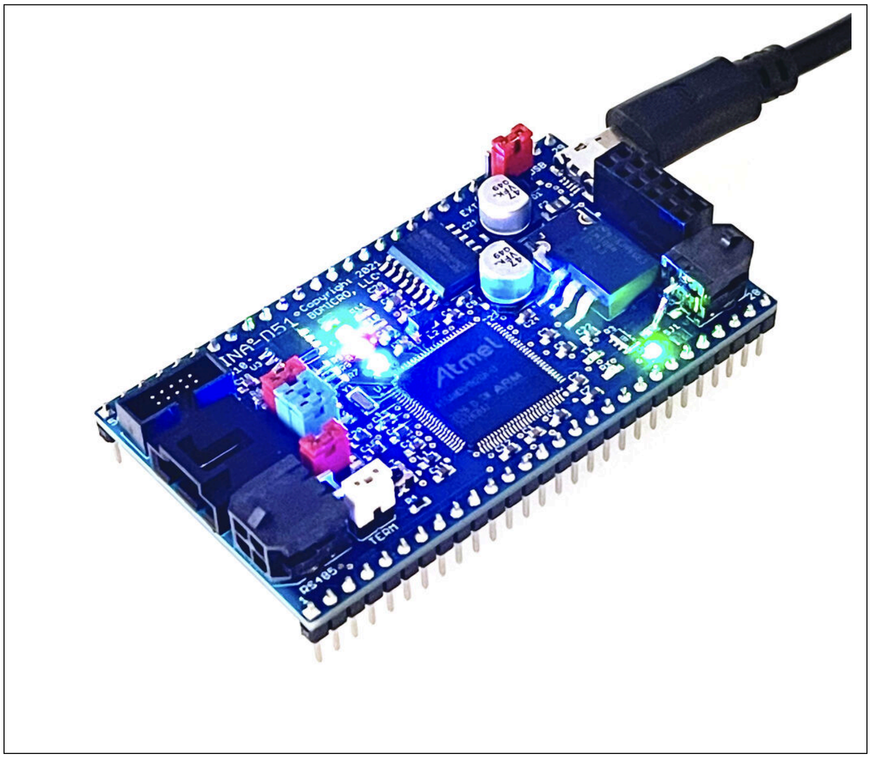


Figure 1: `code.py` simple LED pattern generator executing

The expected results are as follows:

1. Power is supplied from the USB port of your computer or hub, the `PWR_SEL` jumper set to `USB` routes power from the USB port to power VINA-D51.
2. The `PWR` LED will light, indicating power has been supplied.

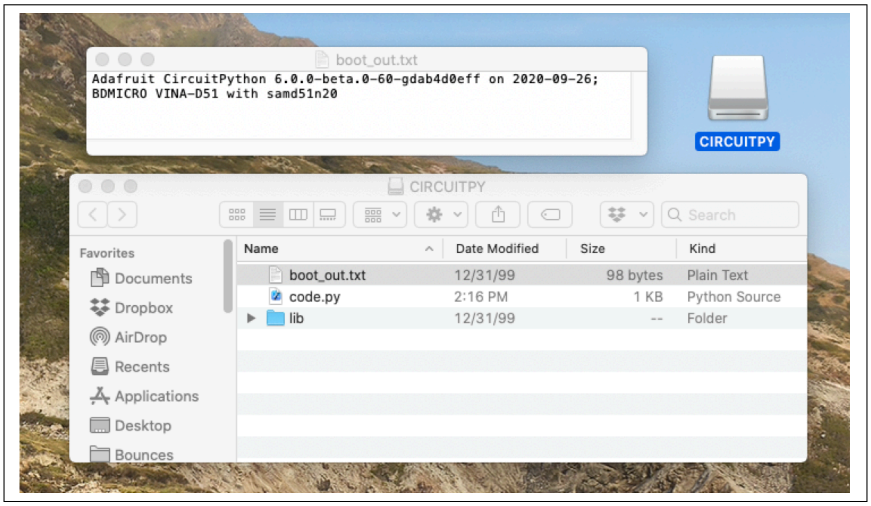


Figure 2: MacOS Desktop with `CIRCUITPY` flash storage drive

3. VINA-D51 will power-up and negotiate with your computer as a mass storage device. When this is successful, you will see a new disk drive appear on your desktop named `CIRCUITPY`. See **Figure 2** showing the drive and sample content of the `boot_out.txt` file indicating the CircuitPython version and the board it is running on, in this case, the `BDMICRO VINA-D51 with samd51n20`.

4. CircuitPython will automatically load and run the BDMICRO pre-installed `code.py` program, located at the root level of the `CIRCUITPY` drive.

5. The sample `code.py` program initializes the LEDs, and begins flashing the on-board red, green, and blue LEDs in a sequential pattern to indicate it is working as expected. **Figure 1** shows the code running on the board with the USB cable connected, `PWR` LED on, and all three `LED_R`, `LED_G`, and `LED_B` LEDs lit. See **Figure 3** for the full source of the pre-installed `code.py` program.

```python
# Simple LED blink using on-board leds

import os
import board
import digitalio
import time

LEDON  = False
LEDOFF = True

class leds:
    def __init__(self, leds):
        self.leds = leds
        return None

    def blink(self, ms):
        for led in self.leds:
            led.value = LEDON
        time.sleep(ms/1000)
        for led in self.leds:
            led.value = LEDOFF
        time.sleep(ms/1000)
        return None

    def on(self):
        for led in self.leds:
            led.value = LEDON
        return None

    def off(self):
        for led in self.leds:
            led.value = LEDOFF
        return None

# initialize LEDs
r_led           = digitalio.DigitalInOut(board.LED_R)
r_led.direction = digitalio.Direction.OUTPUT
r_led.value     = LEDOFF

g_led           = digitalio.DigitalInOut(board.LED_G)
g_led.direction = digitalio.Direction.OUTPUT
g_led.value     = LEDOFF

b_led           = digitalio.DigitalInOut(board.LED_B)
b_led.direction = digitalio.Direction.OUTPUT
b_led.value     = LEDOFF

all_leds = leds([r_led, g_led, b_led])
rled     = leds([r_led])
gled     = leds([g_led])
bled     = leds([b_led])

all_leds.off()

t1 = .2
t2 = .5

while True:
    rled.blink(100)
    time.sleep(t1)

    gled.blink(100)
    time.sleep(t1)

    bled.blink(100)
    time.sleep(t1)

    time.sleep(t2)
    all_leds.blink(100)
    time.sleep(t2)

```

Figure 3: `code.py` simple LED pattern generator source code

There is much more to VINA-D51. This section was intended to get you up and running quickly. Namely:

1. Initial power-up and verification using a simple USB connection to a PC or laptop, extra cables or power supply are not required
2. Brief intro to the default configuration
3. Demo of the sample, pre-installed `code.py` program for reference
4. Demonstrate that you do not need a hardware programmer or complex development environment setup to get started quickly and develop code for VINA-D51; all that is needed is a PC or laptop, and a text editor to begin developing applications with VINA-D51
5. Introduce the `PWR_SEL` jumper to set the power source
6. Provide a visual indication that your VINA-D51 is operational and ready to go

NOTE : While VINA-D51 ships with CircuitPython to help you get started quickly and easily, you can develop in other languages, compilers, and environments if you choose.

For example, many developers choose GCC's `gcc-arm-none-eabi`, a popular and capable cross-compiler available from `https://developer.arm.com`. Pre-built downloads are available for Windows, MacOS, Linux, and others. Or you can build the cross development tools yourself from source. With C, you have full hardware control over the MCU including interrupts and timers for event driven applications.

The Keil compiler is a stellar professional development environment and compiler as well. `https://www.keil.com`

In all cases, code examples and bootstrap to get started quickly are usually included for rapidly configuring a project.

NOTE : Because we ship the VINA-D51 with the UF2 Bootloader installed, a hardware programmer is not needed to flash new firmware to the board, independent of using CircuitPython. The UF2 Bootloader will flash your C compiled programs as well, once converted to the UF2 format. The UF2 Bootloader makes reflashing the on-chip firmware easy using Mac or Windows drag and drop file operations, replacing CircuitPython or other application.

3 Applications

The VINA-D51 is a powerful, robust, pluggable control module designed for modular control applications. Built with all-industrial grade components and features, it is well-suited for a wide variety of automation and control requirements, including harsh environments.

VINA-D51 saves time and money by providing a proven, modular design, that enables custom application-specific solutions to move from the drawing board to finished product quickly.

NOTE : VINA-D51 enables you to develop your application-specific board independently of the control section. This simplifies your design, saves CAD time during the design phase, reduces errors that can cost weeks of PCB re-manufacture to correct, and saves debug time during the prototype and testing phases. By starting with VINA-D51, you start from a known-good foundation, reducing complexity and risk.

Incorporate the VINA-D51 in your designs over and over again and multiply your savings. Useful for custom applications, one-off projects, breadboarded prototypes, and even standalone projects. VINA-D51 has all of the common essentials built-in, on-board.

VINA-D51 is designed to include the most common components and facilities you need for the main control module:

- **POWER** : Powerful 32-bit ARM Cortex M4 microcontroller — based on the Microchip Atmel ATSAMD51N20w/1MB of program flash and 256K of RAM at 120 MHz Core clock
- **INDUSTRIAL GRADE** : All industrual spec parts for reliable operation in demanding applications
- **CONNECTIVITY** : USB interface for connecting to your development PC or Laptop computer for programming; provides virtual com port for CircuitPython console and UART, and mass storage filesystem interface to the 16 MB on-board flash for programs and data storage; can additionally be used in your application
- **CONTROL & INTERFACING** : RS485, I2C, WIFI : Latching I2C interface connector on-board for sensors, controllers, displays, keypads, etc, on this common bus; latching RS485 interface connector on-board, a UART-style differential bus with high noise immunity common in manufacturing and control automation applications; 12-pin AUX expansion socket on-board and WIFI-enabled using the ubiquitous and inexpensive ESP-01S module and general expansion capability; additionally, VINA-D51 brings out Digital I/O, UARTs, I2C, SPI, A/D, DAC, USB, and many other useful control and interfacing capabilities to the edge
- **DEVELOPMENT and DEBUG** : CircuitPython and UF2 Bootloader pre-installed to get started developing right away with just a laptop or PC and a text editor; no extra software or IDEs to install and no extra hardware to purchase; and with a standard ARM Cortex SWD debug header on-board, low level programming in C/C++ with source-level, single-step debugging utilizing a hardware programmer like the Segger J-Link
- **DATA STORAGE** : Large 16 MB of on-board flash filesystem storage space for your programs and data presented as an external drive on your development workstation; soldered directly to the board, this flash storage won't suffer from corrosion or vibration in harsh environments like SD-Card type storage are prone
- **FLEX VOLTAGE** : Robust power section with wide voltage input range for flexible and reliable operation, and enough headroom to supply power to other devices - perhaps even your whole

application board - up to about 3 amps; incorporates industrial grade Molex latch-lock connector that will not vibrate loose even in the most demanding applications

To get you started even more quickly, BDMICRO provides the Autodesk Eagle CAD part for VINA-D51. Getting started is as easy as downloading the part and adding it to your application-specific design, allowing you to focus on your custom application's needs, with all the right connectors, headers, and other circuitry needed for your specific project solution.

NOTE : Download the Eagle Part Library for VINA-D51 from our website here:
`https://www.bdmicro.com/products/vina-d51`

When it is time to test your custom application board, you know you've started from a solid foundation with the VINA-D51 module. VINA-D51 saves you from having to build and design the control section and other commonly needed facilities of your application over and over. Even when cut and pasting from one design to another, it is easy to make mistakes with dangling or conflicting signal names and other common errors. So it saves you money by saving you time - both in design work, as well as testing and debugging.

And when it is time to design and build another custom application solution, you are already familiar with the VINA-D51 and its capabilities. You know you can get a brand new custom application up and running with it quickly and with confidence. Saving you time and money, over and over again.

4 Autodesk Eagle CAD Part for VINA-D51

The Autodesk Eagle CAD Part for VINA-D51 is available for download here:
`https://www.bdmicro.com/products/vina-d51`. See **Figure 4** for the part's board outline, and **Figure 5** for the schematic symbol.

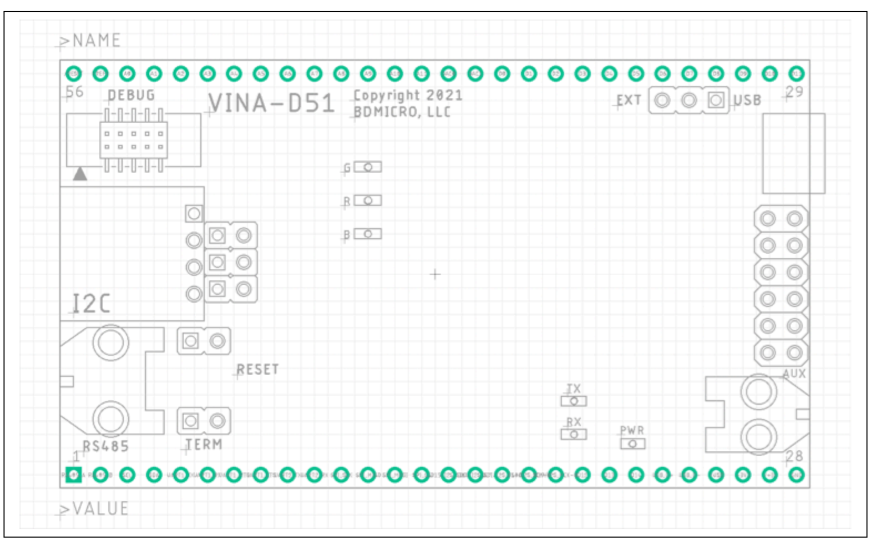


Figure 4: Autodesk Eagle CAD Part for VINA-D51 (board)

> **NOTE :** You can turn off the $tDocu$ layer in the Eagle layout editor to eliminate the part reference locations on the VINA-D51 part. You might want to do this if you are laying out parts underneath the VINA-D51 part for placement clarity.

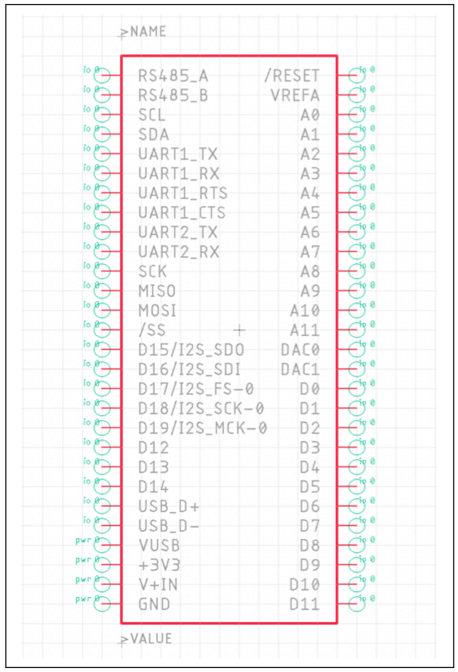


Figure 5: Autodesk Eagle CAD Part for VINA-D51 (schematic symbol)

5 Notable On-Board Features

The VINA-D51 has several notable on-board features. These include:

1. **32-bit ARM Cortex M4 MCU** — with 1MB Firmware Flash and 256KB RAM clocked at 120MHz (Microchip Atmel ATSAMD51N20), with A/D, DAC, Digital I/O, UARTs, I2C, SPI, PWM, Timers, Capacitive Touch, and many other capabilities
2. **1MB Firmware Flash** — large firmware space
3. **256KB Static RAM** — large, fast, internal RAM
4. **Hardware Peripheral Control** — A/D, DAC, Digital I/O, UARTs, I2C, SPI, I2S, PWM, Timers, Capacitive Touch, and more
5. **Highly Configurable** — to suit application needs
6. **CircuitPython** — pre-installed (MicroPython derivative)
7. **Easy to get started** developing immediately using only a PC or Laptop and a text editor
8. **UF2 Bootloader** — pre-installed eliminating the need for additional hardware or dedicated programmer to flash the firmware

9. **No IDEs needed** or complicated development environments to set up or install and no additional hardware needed for programming
10. **USB Port** for programming and interfacing
11. **Virtual COM Port** for connecting to the console using standard terminal emulator from a PC or Laptop - no additional hardware required for programming
12. **Large 16 MB Flash Filesystem** for data and programs; used by CircuitPython filesystem or other custom application; can be used for program and data storage under application control; soldered directly on the board, more reliable than SD-card type filesystem storage especially for harsh environments
13. **Mass Storage Device Interface to PC or Mac** for downloading programs
14. **WIFI** using the AUX module socket with an ESP-01S or BDMICRO ATW-01 WiFi Module with the WINC-1500
15. **Wide voltage input from 3.7 to 24V** with robust Latch-Lock connector that won't vibrate loose even in harsh environments; mating connector is the Molex 0430250208 Microfit 3.0 Series Latch Lock connector
16. **USB Port Power**
17. **RS485 Port** — with optional termination on-board, for interfacing with this common, differential, high noise immunity industrial control bus — mating part is the Molex 0430250408 3mm MicroFit 3.0 Series; optional power can be supplied to or from the RS485 bus or I2C bus through jumper selection; one can even supply power to VINA-D51 using regulated 3.3V through this connector (jumper selectable)
18. **I2C Connector** — on-board, 4-pins for SCL, SDA, GND, and optionally 3.3V; with on-board pull-ups, for easily connecting I2C sensors, displays, etc — mating part is the Molex 0050579404 SL 70066 Series with robust latch lock that won't vibrate loose even in harsh environments; optional power can be supplied to the I2C connector through jumper selection to power I2C devices or to even power the VINA-D51 from regulated 3.3V through this connector (jumper selectable)
19. **On-Board I2C pull-ups** — jumper selectable
20. **32kHz RTC Crystal** — for accurate time keeping
21. **/RESET Switch** — single-tap to reset the board, double-tap to enter the bootloader
22. **ARM Cortex SWD Debug Port** — for low-level firmware flashing and source level single-step debugging with hardware programmer support such as the Segger J-Link
23. **Robust, Latch-Lock Power Connectors** — for Power, RS485, and I2C — will not vibrate loose even in harsh environments
24. **Jumper Selectable Power** — either USB or External
25. **Jumper-Enabled RS485 Bus Termination**
26. **Jumper-Enabled I2C Bus Pull-Ups**
27. **Jumper-Enabled 3.3V Power to or from the RS485 Bus**
28. **Jumper-Enabled 3.3V Power to or from the I2C Bus**

29. **Fused Power Inputs** — both USB and External Power are fused appropriately for their voltage input and power draw
30. **Power Indicator LED** — with cuttable solder-jumper to disable for very lower power modes
31. **Three Programmable Status LEDs** — Red, Green, and Blue - connected to digital I/O and fully programmable under application control
32. **RX and TX LEDs** — for use by CircuitPython for indicating transmission and reception of data over USB, or fully programmable under application control
33. **Small Size** — 1.6 x 2.8 inches
34. **Low Power** — draws about 30mA, lower power modes are available through programming
35. **All Industrial Grade Parts** — ready for demanding environments and control applications

6 Pinout Diagram

See **Figure 6** for the VINA-D51 pinout assignments.

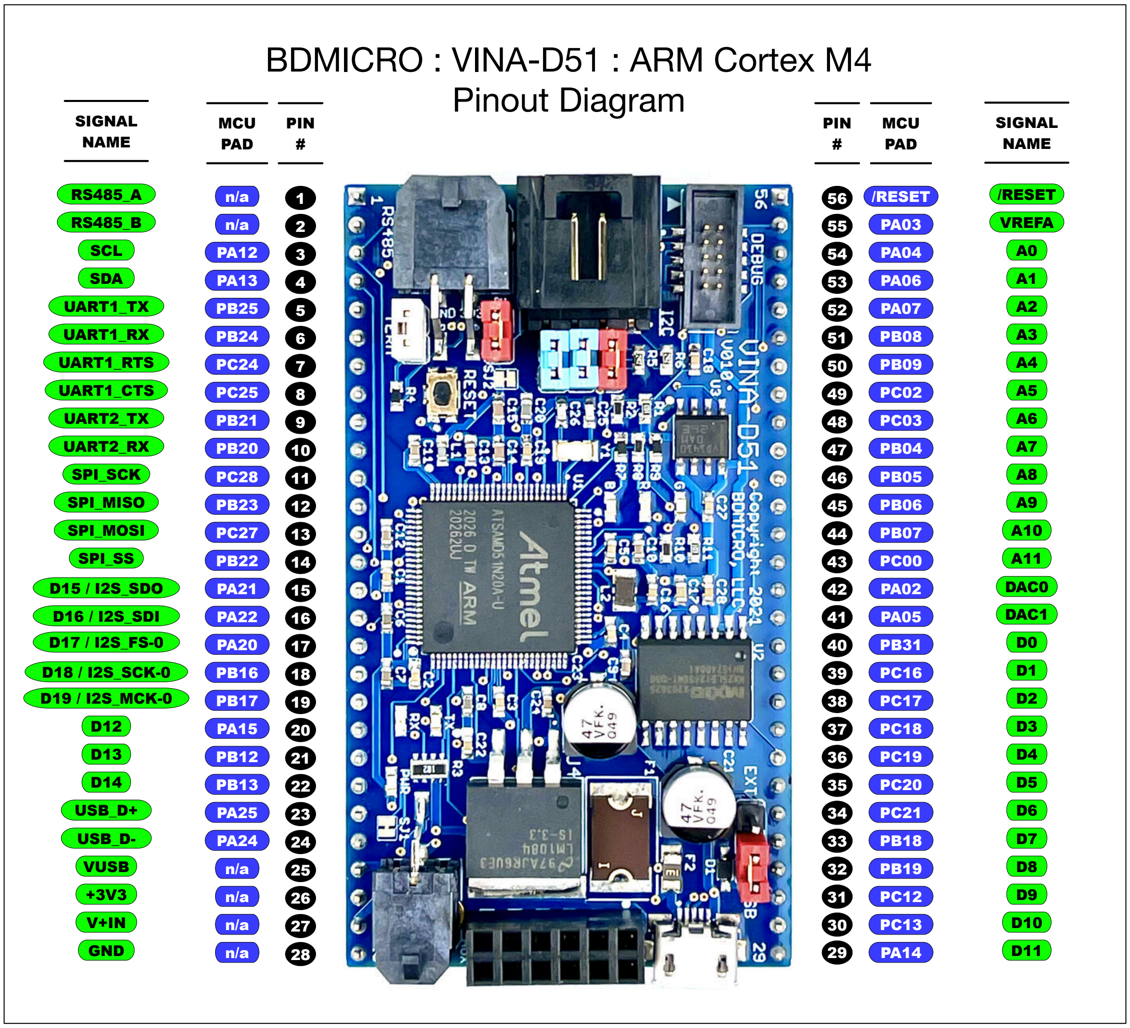


Figure 6: VINA-D51 Pinout Assignments

NOTE : Signal names are derived from the convenient function or peripheral they connect to. The MCU Pad name is also provided for ATSAMD51N20 datasheet reference. See **Table 1** and **Table 2** table detail in this section for the extensive list of alternate peripheral functions available for each pin.

VINA-D51 Pinout

Pin #	Signal Name(s)	MCU Pad	Primary Peripheral	Alternate Peripherals Available
1	RS485_A	NONE	NONE	NONE
2	RS485_B	NONE	NONE	NONE
3	I2C1_SCL, SCL	PA12	SERCOM4,PAD-1/3	AC,CMP-0/1 EIC,EXTINT-12 PCC,DEN1 PORT,P-12 SDHC0,SDCD SERCOM2,PAD-0/1 SERCOM4,PAD-1/3 TC2,WO-0/2 TCC0,WO-6/1 TCC1,WO-2/4
4	I2C1_SDA, SDA	PA13	SERCOM4,PAD-0/3	AC,CMP-1/1 EIC,EXTINT-13 PCC,DEN2 PORT,P-13 SDHC0,SDWP SERCOM2,PAD-1/1 SERCOM4,PAD-0/3 TC2,WO-1/2 TCC0,WO-7/1 TCC1,WO-3/4
5	UART1_TX	PB25	SERCOM2,PAD-0/4	AC,CMP-1/3 EIC,EXTINT-9 PDEC,QDI-2/4 PORT,P-57 SERCOM0,PAD-1/2 SERCOM2,PAD-0/4
6	UART1_RX	PB24	SERCOM2,PAD-1/4	AC,CMP-0/3 EIC,EXTINT-8 PDEC,QDI-1/4 PORT,P-56 SERCOM0,PAD-0/2 SERCOM2,PAD-1/4
7	UART1_RTS	PC24	SERCOM2,PAD-2/4	EIC,EXTINT-8 PORT,P-88 SERCOM0,PAD-2/2 SERCOM2,PAD-2/4
8	UART1_CTS	PC25	SERCOM2,PAD-3/4	EIC,EXTINT-9 PORT,P-89 SERCOM0,PAD-3/2 SERCOM2,PAD-3/4
9	UART2_TX, TX, I2C2_SDA	PB21	SERCOM7,PAD-0/4	EIC,EXTINT-5 GCLK,IO-7 PORT,P-53 SDHC1,SDDAT-3 SERCOM3,PAD-1/2 SERCOM7,PAD-0/4 TCC1,WO-3/2
10	UART2_RX, RX, I2C2_SCL	PB20	SERCOM7,PAD-1/4	EIC,EXTINT-4 GCLK,IO-6 PDEC,QDI-2/2 PORT,P-52 SDHC1,SDDAT-2 SERCOM3,PAD-0/2 SERCOM7,PAD-1/4 TCC1,WO-2/2
11	SPI1_SCK, SCK	PC28	SERCOM1,PAD-1/3	CCL,IN-5 EIC,EXTINT-12 PORT,P-92 SERCOM1,PAD-1/3
12	SPI1_MISO, MISO	PB23	SERCOM1,PAD-3/3	CCL,OUT-0 EIC,EXTINT-7 GCLK,IO-1 OSCCTRL,XOUT-1 PDEC,QDI-0/4 PORT,P-55 SERCOM1,PAD-3/3 SERCOM5,PAD-3/4 TC7,WO-1/2
13	SPI1_MOSI, MOSI	PC27	SERCOM1,PAD-0/3	CCL,IN-4 EIC,EXTINT-11 PORT,P-91 SERCOM1,PAD-0/3
14	SPI1_SS, SS	PB22	SERCOM1,PAD-2/3	CCL,IN-0 EIC,EXTINT-6 GCLK,IO-0 OSCCTRL,XIN-1 PDEC,QDI-2/3 PORT,P-54 SERCOM1,PAD-2/3 SERCOM5,PAD-2/4 TC7,WO-0/2 USB,SOF_1KHZ/2
15	D15, I2S_SDO	PA21	I2S,SDO/2	ADC0,X-15 ADC0,Y-15 EIC,EXTINT-5 I2S,SDO/2 PCC,DATA-5 PORT,P-21 SDHC1,SDCK SERCOM3,PAD-3/2 SERCOM5,PAD-3/2 TC7,WO-1/1 TCC0,WO-1/6 TCC1,WO-5/1
16	D16, I2S_SDI	PA22	I2S,SDI/2	ADC0,X-16 ADC0,Y-16 CCL,IN-6 EIC,EXTINT-6 I2S,SDI/2 PCC,DATA-6 PORT,P-22 SERCOM3,PAD-0/1 SERCOM5,PAD-1/2:3:4 TC4,WO-0/3 TCC0,WO-2/6 TCC1,WO-6/1
17	D17, I2S_FS_0	PA20	I2S,FS-0/2	ADC0,X-14 ADC0,Y-14 EIC,EXTINT-4 I2S,FS-0/2 PCC,DATA-4 PORT,P-20 SDHC1,SDCMD SERCOM3,PAD-2/2 SERCOM5,PAD-2/2 TC7,WO-0/1 TCC0,WO-0/6 TCC1,WO-4/1
18	D18, I2S_SCK_0	PB16	I2S,SCK-0/2	CCL,IN-11 EIC,EXTINT-0 GCLK,IO-2 I2S,SCK-0/2 PORT,P-48 SDHC1,SDCD SERCOM5,PAD-0/1 TC6,WO-0/1 TCC0,WO-4/6 TCC3,WO-0/2
19	D19, I2S_MCK_0	PB17	I2S,MCK-0/2	CCL,OUT-3 EIC,EXTINT-1 GCLK,IO-3 I2S,MCK-0/2 PORT,P-49 SDHC1,SDWP SERCOM5,PAD-1/1 TC6,WO-1/2 TCC0,WO-5/6 TCC3,WO-1/2
20	D12	PA15	TCC2,WO-1/1	EIC,EXTINT-15 GCLK,IO-1 OSCCTRL,XOUT-0 PORT,P-15 SERCOM2,PAD-3/1 SERCOM4,PAD-3/3 TC3,WO-1/1 TCC1,WO-3/5 TCC2,WO-1/1
21	D13	PB12	TCC3,WO-0/1	ADC0,X-26 ADC0,Y-26 EIC,EXTINT-12 GCLK,IO-6 I2S,SCK-1/1 PORT,P-44 SDHC0,SDCD SERCOM4,PAD-0/1 TC4,WO-0/2 TCC0,WO-0/5 TCC3,WO-0/1
22	D14	PB13	TCC3,WO-1/1	ADC0,X-27 ADC0,Y-27 EIC,EXTINT-13 GCLK,IO-7 I2S,MCK-1/1 PORT,P-45 SDHC0,SDWP SERCOM4,PAD-1/1 TC4,WO-1/2 TCC0,WO-1/5 TCC3,WO-1/1
23	USB_DP	PA25	USB,DP	CCL,OUT-2 EIC,EXTINT-9 PDEC,QDI-1/3 PORT,P-25 SERCOM3,PAD-3/1 SERCOM5,PAD-3/3 TC5,WO-1/3 USB,DP
24	USB_DM	PA24	USB,DM	CCL,IN-8 EIC,EXTINT-8 PDEC,QDI-0/3 PORT,P-24 SERCOM3,PAD-2/1 SERCOM5,PAD-2/3 TC5,WO-0/3 TCC2,WO-2/1 USB,DM
25	VUSB			
26	+3V3	VCC	VCC	VCC
27	V+IN			
28	GND	GND	GND	GND

Table 1: VINA-D51 Pinout Table

VINA-D51 Pinout (Continued)				
Pin #	**Signal Name(s)**	**MCU Pad**	**Primary Peripheral**	**Alternate Peripherals Available**
29	D11	PA14	TCC2,WO-0/1	EIC,EXTINT-14 GCLK,IO-0 OSCCTRL,XIN-0 PCC,CLK PORT,P-14 SERCOM2,PAD-2/1 SERCOM4,PAD-2/3 TC3,WO-0/1 TCC1,WO-2/5 TCC2,WO-0/1
30	D10	PC13	TCC1,WO-7/3	EIC,EXTINT-13 PCC,DATA-11 PORT,P-77 SERCOM6,PAD-0/4:5 SERCOM7,PAD-1/1:3 TCC0,WO-3/3 TCC1,WO-7/3
31	D9	PC12	TCC1,WO-6/3	EIC,EXTINT-12 PCC,DATA-10 PORT,P-76 SERCOM6,PAD-1/4:5 SERCOM7,PAD-0/1:3 TCC0,WO-2/3 TCC1,WO-6/3
32	D8	PB19	TCC1,WO-1/3	EIC,EXTINT-3 GCLK,IO-5 PDEC,QDI-1/2 PORT,P-51 SDHC1,SDDAT-1 SERCOM5,PAD-3/1 SERCOM7,PAD-3/4 TCC1,WO-1/3
33	D7	PB18	TCC1,WO-0/3	EIC,EXTINT-2 GCLK,IO-4 PDEC,QDI-0/2 PORT,P-50 SDHC1,SDDAT-0 SERCOM5,PAD-2/1 SERCOM7,PAD-2/4 TCC1,WO-0/3
34	D6	PC21	TCC0,WO-5/4	CCL,IN-10 EIC,EXTINT-5 PORT,P-85 SDHC1,SDWP TCC0,WO-5/4
35	D5	PC20	TCC0,WO-4/4	CCL,IN-9 EIC,EXTINT-4 PORT,P-84 SDHC1,SDCD TCC0,WO-4/4
36	D4	PC19	TCC0,WO-3/4	EIC,EXTINT-3 PORT,P-83 SERCOM0,PAD-3/4 SERCOM6,PAD-3/1 TCC0,WO-3/4
37	D3	PC18	TCC0,WO-2/4	EIC,EXTINT-2 PDEC,QDI-2/1 PORT,P-82 SERCOM0,PAD-2/4 SERCOM6,PAD-2/1 TCC0,WO-2/4
38	D2	PC17	TCC0,WO-1/4	EIC,EXTINT-1 PDEC,QDI-1/1 PORT,P-81 SERCOM0,PAD-0/4 SERCOM6,PAD-1/1 TCC0,WO-1/4
39	D1	PC16	TCC0,WO-0/4	EIC,EXTINT-0 PDEC,QDI-0/1 PORT,P-80 SERCOM0,PAD-1/4 SERCOM6,PAD-0/1 TCC0,WO-0/4
40	D0	PB31	TCC0,WO-7/4	EIC,EXTINT-15 PORT,P-63 SERCOM5,PAD-0/5 SERCOM7,PAD-1/5 TC0,WO-1/3 TCC0,WO-7/4 TCC4,WO-1/2
41	DAC1	PA05	DAC,VOUT-1	AC,AIN-1 ADC0,AIN-5 CCL,IN-1 DAC,VOUT-1 EIC,EXTINT-5 PORT,P-5 SERCOM0,PAD-1/3 TC0,WO-1/1
42	DAC0	PA02	DAC,VOUT-0	ADC0,AIN-0 DAC,VOUT-0 EIC,EXTINT-2 PORT,P-2 RTC,IN-2
43	A11	PC00	ADC1,AIN-10	ADC1,AIN-10 EIC,EXTINT-0 PORT,P-64 RTC,IN-3
44	A10	PB07	ADC1,AIN-9	ADC0,X-25 ADC0,Y-25 ADC1,AIN-9 CCL,IN-7 EIC,EXTINT-7 PORT,P-39
45	A9	PB06	ADC1,AIN-8	ADC0,X-24 ADC0,Y-24 ADC1,AIN-8 CCL,IN-6 EIC,EXTINT-6 PORT,P-38
46	A8	PB05	ADC1,AIN-7	ADC0,X-23 ADC0,Y-23 ADC1,AIN-7 EIC,EXTINT-5 PORT,P-37
47	A7	PB04	ADC1,AIN-6	ADC0,X-22 ADC0,Y-22 ADC1,AIN-6 EIC,EXTINT-4 PORT,P-36
48	A6	PC03	ADC1,AIN-5	ADC1,AIN-5 EIC,EXTINT-3 PORT,P-67
49	A5	PC02	ADC1,AIN-4	ADC1,AIN-4 EIC,EXTINT-2 PORT,P-66
50	A4	PB09	ADC1,AIN-1	ADC0,AIN-3 ADC0,X-2 ADC0,Y-2 ADC1,AIN-1 CCL,OUT-2 EIC,EXTINT-9 PORT,P-41 SERCOM4,PAD-1/2 TC4,WO-1/1
51	A3	PB08	ADC1,AIN-0	ADC0,AIN-2 ADC0,X-1 ADC0,Y-1 ADC1,AIN-0 CCL,IN-8 EIC,EXTINT-8 PORT,P-40 SERCOM4,PAD-0/2 TC4,WO-0/1
52	A2	PA07	ADC0,AIN-7	AC,AIN-3 ADC0,AIN-7 ADC0,X-5 ADC0,Y-5 CCL,OUT-0 EIC,EXTINT-7 PORT,P-7 SDHC0,SDWP SERCOM0,PAD-3/3 TC1,WO-1/1
53	A1	PA06	ADC0,AIN-6	AC,AIN-2 ADC0,AIN-6 ADC0,X-4 ADC0,Y-4 CCL,IN-2 EIC,EXTINT-6 PORT,P-6 SDHC0,SDCD SERCOM0,PAD-2/3 TC1,WO-0/1
54	A0	PA04	ADC0,AIN-4	AC,AIN-0 ADC0,AIN-4 ADC0,X-3 ADC0,Y-3 CCL,IN-0 EIC,EXTINT-4 PORT,P-4 SERCOM0,PAD-0/3 TC0,WO-0/1
55	VREFA	PA03	ADC0,AIN-1	ADC0,AIN-1 ADC0,X-0 ADC0,Y-0 EIC,EXTINT-3 PORT,P-3
56	RESET	RESET_N	PM,RESET_N	PM,RESET_N

Table 2: VINA-D51 Pinout Table

IMPORTANT : It is important to note that any pin can be configured for any of the alternate peripheral functions. You are not limited to the primary peripheral. The primary peripheral was chosen so that the VINA-D51 presented all of the most important features needed for a modular controller targeted at a variety of applications, to ensure that it is as flexible as possible with regard to the pins brought out to the edge. If any of these primary peripherals are not suitable for your application, VINA-D51 has the flexibility to redefine them through programming to any of the alternate peripheral functions to suit your needs.

7 Mechanical Diagram

See **Figure 7** for the VINA-D51 mechanical layout with dimensions.

NOTE : Units are in inches.

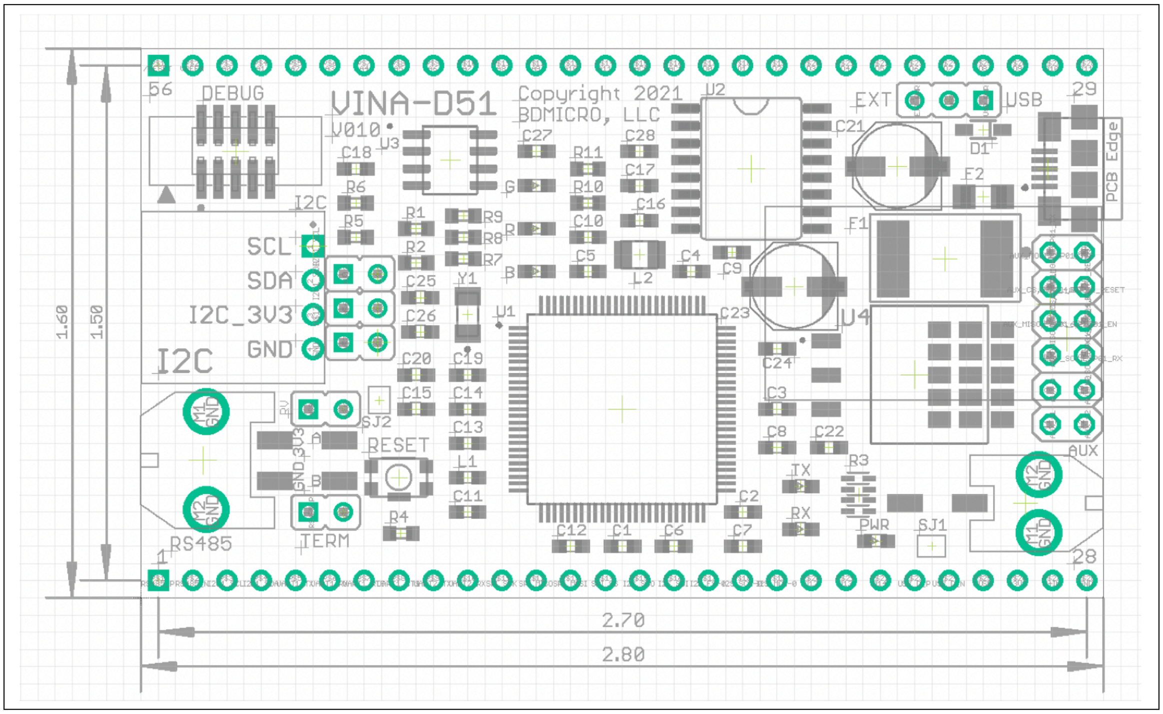


Figure 7: VINA-D51 Mechanical Layout

8 Power

See **Figure 8** for the a diagram of the external power connection. The on-board regulator and fusing accepts a wide voltage range for flexibility and to minimize the chance of over-voltage using common power supplies. Accepted input range is 3.7V to 24V unregulated DC.

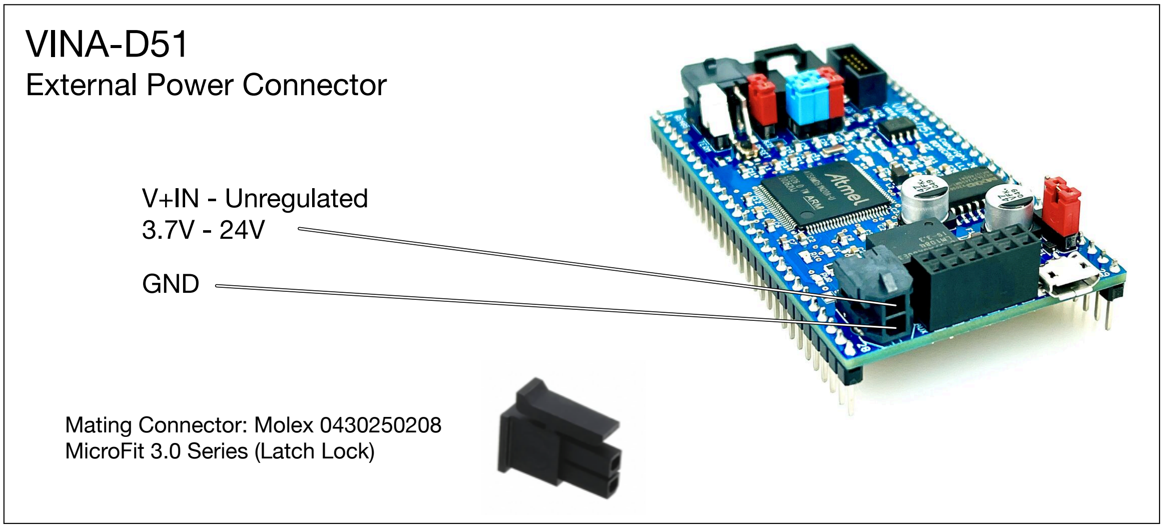


Figure 8: VINA-D51 External Power Connector

NOTE : The on-board regulator can deliver up the 3A of current. The VINA-D51 itself only uses a fraction of that - about 20mA. So substantial headroom is left over for powering other circuits like sensors and displays. As with any circuit, be sure and power it with a suitable, high quality power supply that meets the power requirements of all of your components.

NOTE : Both the External Power and USB power inputs are fused using resettable PTC fuses sized appropriately for the power from each source.

While not depicted in the diagrams, VINA-D51 can also be powered by the 3V3 pin of the I2C connector or that of the RS485 connector. See **Section 11** for a description of the 3.3V pin of the RS485 connector. See **Section 12** for a description of the 3.3V pin of the I2C connector. These sections identify the 3.3V locations within their respective connectors and the jumpers that enable them. The setup for this configuration is:

1. ***Remove*** the `PWR_SEL` jumper
2. Connect one of the VINA-D51 I2C or RS485 connectors to a connection that supplies **regulated 3.3V** to the 3V3 pin
3. Install either the `IV` or the `RV` jumpers depending on whether power is comming from the I2C or the RS485 connector, respectively.

This configuration will power VINA-D51 using regulated 3.3V from an external source through the I2C or RS485 connector. The I2C connector can be used with the `IV` jumper installed, or the RS485 connector can be used with the RV jumper installed. Note that input power filtering from the External Power connection and USB power connection are bypassed. The advantage is that it allows VINA-D51 power to piggy back off of another device's power, eliminating the need for multiple power supplies, assuming the origin power supply is sufficiently capable and clean to power VINA-D51.

See **Figure 9** for the diagram and location of the PWR_SEL jumper. This jumper is used to select between the external latch lock Molex power connector near pin 28, or powering the board from the USB power of your computer or hub. Or it can be removed to supply power to VINA-D51 using one of the 3V3 pins available: Pin 26, I2C-1 3V3, I2C-2 3V3, or RS485 3V3 from an external source.

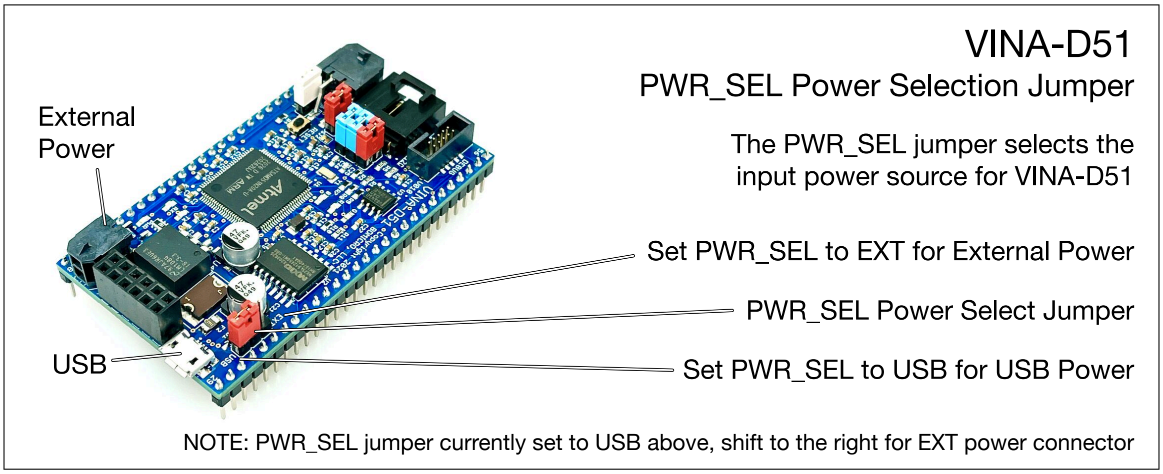


Figure 9: VINA-D51 PWR_SEL Jumper for USB or External Power Selection

See **Table 3** for a description of the power inputs and outputs available at the edge pins of the VINA-D51 controller module.

VINA-D51 Main Power Connections Pins at Module Edge				
Pin #	**Signal**	**Purpose**	**Minimum**	**Maximum**
25	VUSB	USB Connector Output Voltage — same as from the USB Connector's voltage input (normally 5V) regardless of the PWR_SEL jumper position; use the PWR_SEL jumper set to the USB position to select USB as the source of power	3.7V	5V
26	+3V3	VCC — Input or Output — if using +3V3 as input power to VINA-D51, disconnect the PWR_SEL jumper; if using +3V3 as an output to power external devices, do not overload the on-board regulator or USB bus depending on the setting of the PWR_SEL jumper	3.3V	3.3V
27	V+IN	Unregulated input voltage - same connection as the External Power connector; use the PWR_SEL jumper set to the EXT position to use this input as the power source	3.7V	24V
28	GND	Board Ground	0V	0V

Table 3: VINA-D51 Power and Ground Pin Connection

9 RESET Switch

See **Figure 10** for the location of the reset switch. The VINA-D51 reset switch is a momentary, normally-open switch. A single-tap will reset VINA-D51. Double-tap to enter the pre-installed UF2 bootloader.

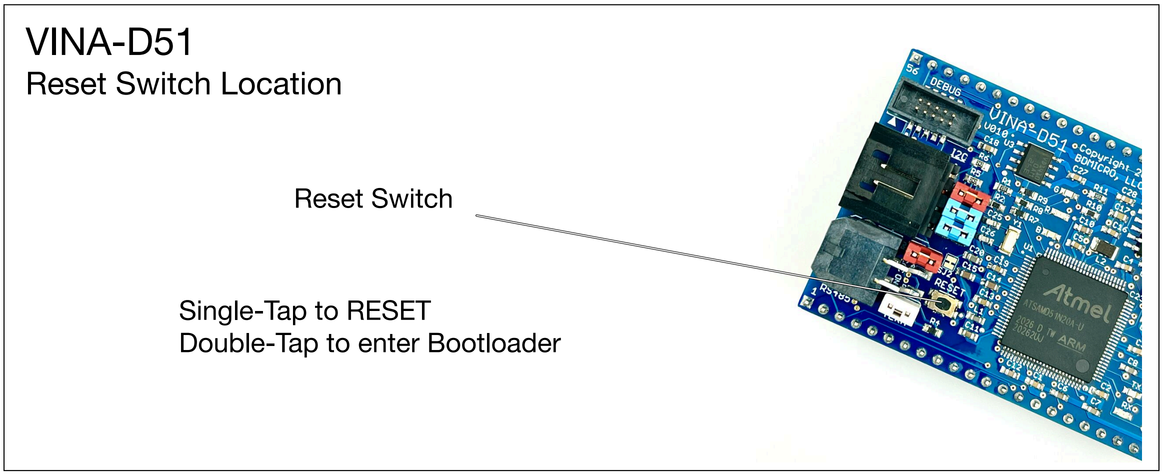


Figure 10: VINA-D51 Reset Switch Location

10 USB

See **Figure 11** for the location of the USB connection.

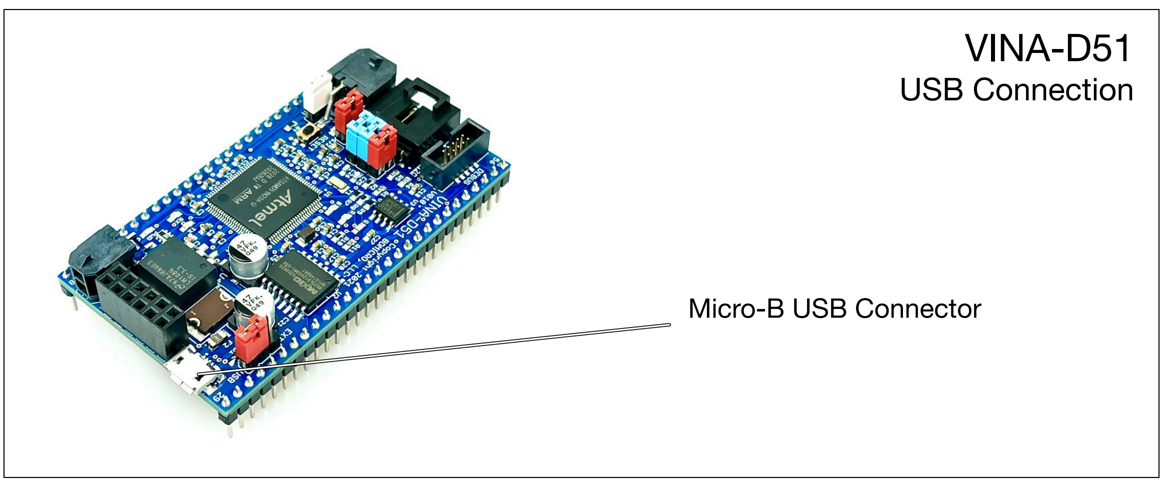


Figure 11: VINA-D51 USB Connector

11 RS485

See **Figure 12** for the a diagram of of the RS485 connector and bus termination jumper.

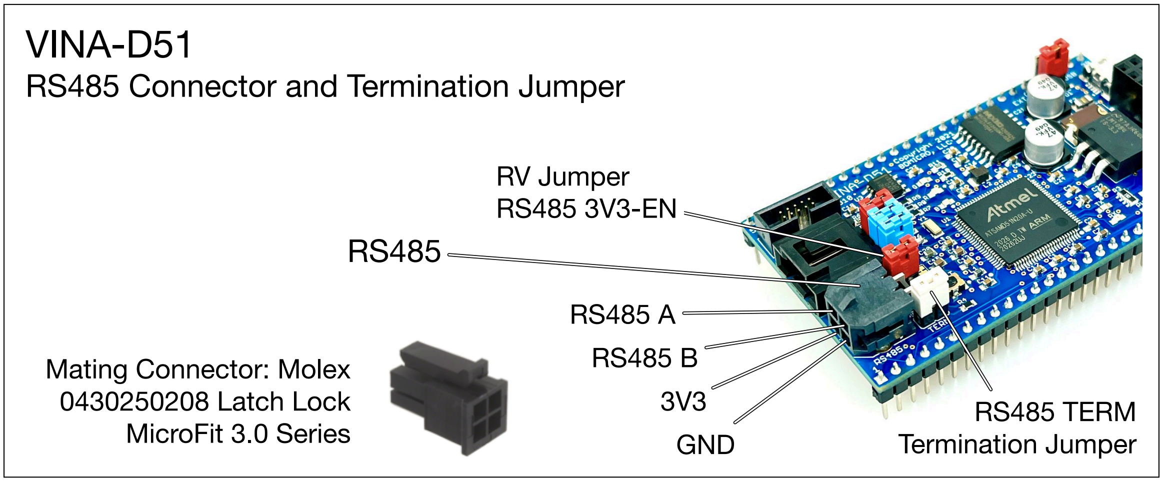


Figure 12: VINA-D51 RS485 Connector and Bus Termination Jumper

VINA-D51 includes one dedicated RS485 interface. The RS485 interface is accessible through Pin 1 and Pin 2 on the board edge — RS485 A and B signals respectively, see **Figure 6** and **Table 1**. RS485 is also available by cable using the on-board 4-pin Molex connector supplying RS485-A, RS485-B, GND, and optionally 3.3V (see **Figure 12**). Additionally, an RS485 termination resistor is available on-board with a jumper to optionally enable or disable termination at the VINA-D51. See **Figure 12** for the location of the connector and associated jumpers for 3.3V EN, and TERM.

The RS485 interface is internally connected to SERCOM5 for full RS485 hardware support of RX, TX, TE (transmitter enable), and RE (receiver enable).

The 3.3V pin connections allow you to power external RS485 devices from the RS485 bus connector as long as their power requirements do not exceed that of the on-board power supply.

The mating connector for the on-board RS485 connector is the Molex 0430250408 Latch Lock MicroFit 3.0 Series connector, that is robust and will not vibrate loose even in harsh environments.

NOTE : The on-board regulator is rated for a maximum of 3A of current. The VINA-D51 itself only uses a fraction of that - about 20mA. The additional headroom can be used to power external devices like other RS485 devices if their power requirements do not exceed that of the on-board power regulation section. As with any circuit, be sure and power it with a suitable, high quality power supply that meets the power requirements of all of your components.

11.1 RS485 Bus Termination

The RS485 bus requires a terminating resistor for proper opertion. See **Figure 12** for the location of the TERM jumper that engages the on-board 120 Ohm termination resistor. If your RS485 bus is terminated elsewhere on the bus, remove the TERM jumper to avoid double-termination.

11.2 RS485 Bus Power

See **Figure 12** for the location of the `RV` jumper that enables 3.3V to the 3V3 pin of the 4-pin RS485 connector. Disconnect the `RV` jumper to disable 3.3V at the connector.

NOTE : If your RS485 device utilizes power from the RS485 bus, and the cable run is short, install the `RV` jumper to enage power. If your RS485 device draws more power than the VINA-D51 board can supply, or the cable run is long, remove the jumper and power your RS485 device from another power supply.

12 I2C

See **Figure 13** for the a diagram of of the I2C connector.

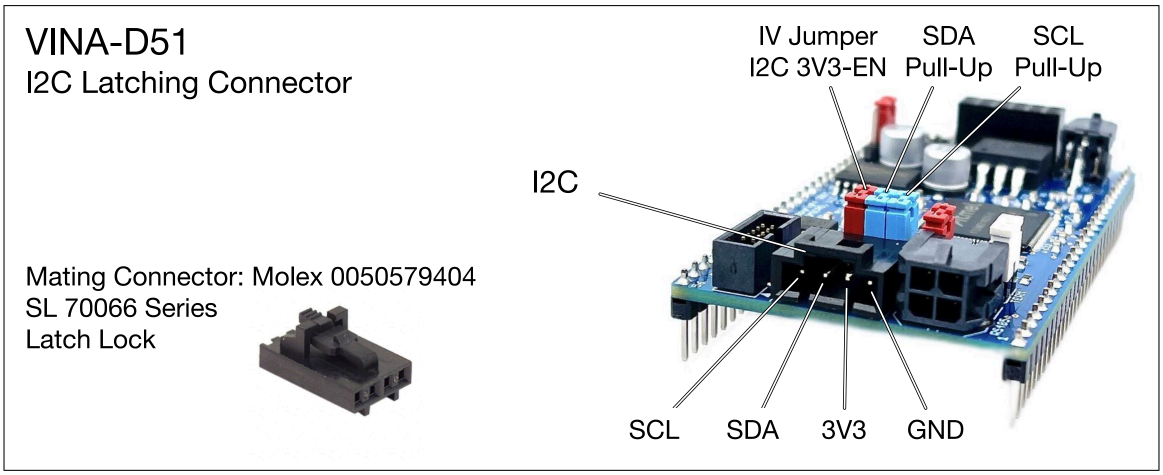


Figure 13: VINA-D51 I2C Connector

VINA-D51 includes one, 4-pin I2C connector supplying connections for SCL, SDA, GND, and optionally 3.3V. The SCL and SDA pins can be connected to VINA-D51's on-board I2C pull-ups using the indicated jumpers in **Figure 13**.

The I2C connector is connected to the same I2C bus as the edge pins labeled SCL and SDA internally. This allows you to easily connect an I2C peripheral using a cable and daisy chaining the bus for additional devices. Use the on-board pull-ups if needed - only one I2C device on the bus needs to have the pull-ups engaged, and VINA-D51 provides these if no other devices on the bus have pull-ups available.

The 3.3V pin connections allow you to power external I2C devices from this connector as long as their power requirements do not exceed that of the on-board power supply.

The mating connector for the on-board I2C connectors is the Molex 0050579404 SL 70066 Series Latch Lock connector for a secure connection that is resistant to vibration and suitable for harsh environments.

NOTE : The on-board regulator is rated for a maximum of 3A of current. The VINA-D51 itself only uses a fraction of that. The additional headroom can be used to power external devices like I2C sensors and displays. As with any circuit, be sure to power it with a suitable, high quality power supply that meets the power requirements of all of your components.

12.1 I2C Pull Ups

See **Figure 13** for the location of the SCL and SDA pull-up jumpers. These jumpers enable the on-board pull-up resistors for the I2C bus. If you wish to use the SCL and SDA pins for a purpose other than I2C, or another I2C device on the bus is supplying the pull-ups, disconnect these jumpers.

12.2 I2C Bus Power

See **Figure 13** for the location of the IV jumper that enables 3.3V to the 3V3 pin of the 4-pin I2C connector.

NOTE : The I2C connector on the VINA-D51 board connects internally to the same I2C pins on the board edge, pins 3 & 4. The I2C connector allows for conveniently connecting I2C peripherals using a cable connection and can optionally power the connected I2C peripheral(s).

If your I2C device utilizes power from the I2C bus, install the IV jumper (**Figure 13**) for the connector. If your I2C device draws more power than the VINA-D51 board can supply, remove the jumper and power the I2C device from another power supply.

13 AUX Socket and Expandable Versatility

VINA-D51 provides an versatile auxiliary socket named 'AUX'. This socket is intended for easy expansion using plug-in modules. For example, pins 1-8 of the 12-pin AUX socket match up the pins of the ESP-01S for providing easy WiFi connectivity. And BDMICRO offers the ATW-01 module using the WINC-1500 module that plugs into the AUX socket for more advanced WiFi capability.

The AUX socket is not limited to those modules. One can easily design a useful module to work with the VINA-D51 in order to expand its capabilitys. The AUX port brings out a full ATSAMD51N20 SERCOM perhipheral hardware in order to communicate with plug-in modules using I2C, RS232 TX/RX w/optional RTS/CTS for flow control, or SPI bus. Additionally, four GPIO pins are also brought out for digial I/O control and sensing, PWM, A/D, and IRQ inputs, greatly expanding the usefulness of the socket and VINA-D51 when the on-board features may not meet all your needs. Thus, the AUX connector is suitable for any application that can be implemented with any combination of those pins and peripherals.

See **Figure 14** for the location of the AUX socket and it's pinout. See **Figure 15** for examples of several different modules plugged into the AUX connector. Specifically note the location and orientation of the ESP-01S module since it doesn't completely fill the socket and has four empty positions. The ESP-01S, if used, should fill pins 1-8, which are located closest to the USB connector.

Other applications for the AUX socket are many, from LoRa radio and other communications modules, servo controllers, special purpose sensors, and so on. It is designed for expandability, and future-proofing VINA-D51 vs having to switch to another controller or placing additional circuitry on a VINA-D51 application carrier board in order to meet the needs of the application.

See **Table 4** for the detailed pinout and MCU pads that brought out to the AUX socket.

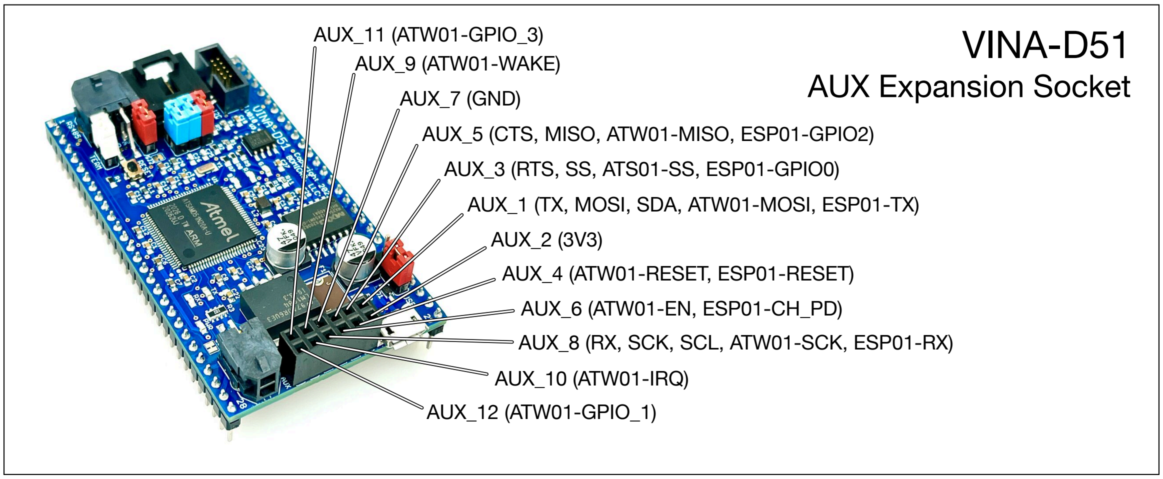


Figure 14: VINA-D51 AUX Socket Header

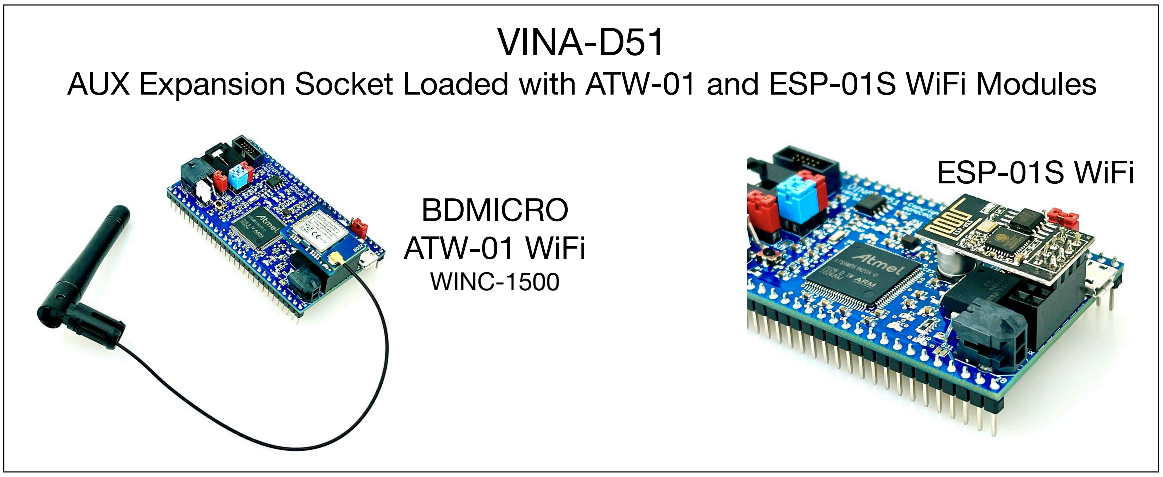


Figure 15: VINA-D51 AUX Socket Header With ATW-01 and ESP-01S WiFi Modules Installed

NOTE : Signals in **Table 4** are not brought out to the VINA-D51 edge pins. They are only available on the 12-pin AUX Socket. Pin numbers match the standard pin assignments of this connector style with Pin 1 being in the top left corner, with the orientation of the top the side closest to the USB connector. The AUX Socket with pin labels is shown in **Figure 14**.

NOTE : **Versatile Expansion Socket** — While the 12-pin AUX socket is designed to interface with expansion modules, its pin layout accommodates the common ESP-01S WiFi module and the BDMICRO ATW-01 WINC-1500 WiFi module. Endless other devices and modules can be designed with the same compatible pinout and utilized as well. The possibilities are abundant, utilizing I2C, UARTs, SPI, PWMs, Input Capture and Counters, Analog Inputs, external interrupts, and general purpose digital I/O as accommodated by the MCU peripherals available on this versatile socket.

This capability opens many possibilities for expansion and utility.

VINA-D51 AUX Socket Pin-Out Detail				
AUX Socket Pin #	Signal Name(s) / Purpose	MCU PAD	MCU Primary Peripheral	MCU Peripheral Alternates
1	AUX_1, AUX_UART_TX, AUX_SPI_MOSI, AUX_I2C_SDA, ATW01_MOSI, ESP01_TX	PA17	SERCOM3,PAD-0/3	ADC0,X-11 ADC0,Y-11 CCL,IN-1 EIC,EXTINT-1 GCLK,IO-3 PCC,DATA-1 PORT,P-17 SERCOM1,PAD-1/1 SERCOM3,PAD-0/3 TC2,WO-1/3 TCC0,WO-5/5 TCC1,WO-1/1
2	AUX_2, 3V3	VCC	VCC	VCC
3	AUX_3, AUX_UART_RTS, AUX_SPI_SS, ATW01_SS, ESP01_GPIO0	PA18	SERCOM3,PAD-2/3	AC,CMP-0/2 ADC0,X-12 ADC0,Y-12 CCL,IN-2 EIC,EXTINT-2 PCC,DATA-2 PORT,P-18 SERCOM1,PAD-2/1 SERCOM3,PAD-2/3 TC3,WO-0/2 TCC0,WO-6/3 TCC1,WO-2/1
4	AUX_4, ATW01_RESET, ESP01_RESET	PC14	PORT,P-78	EIC,EXTINT-14 PCC,DATA-12 PORT,P-78 SERCOM6,PAD-2/4 SERCOM7,PAD-2/1 TCC0,WO-4/3 TCC1,WO-0/5
5	AUX_5, AUX_UART_CTS, AUX_SPI_MISO, ATW01_MISO, ESP01_GPIO2	PA19	SERCOM3,PAD-3/3	AC,CMP-1/2 ADC0,X-13 ADC0,Y-13 CCL,OUT-0 EIC,EXTINT-3 PCC,DATA-3 PORT,P-19 SERCOM1,PAD-3/1 SERCOM3,PAD-3/3 TC3,WO-1/2 TCC0,WO-7/3 TCC1,WO-3/1
6	AUX_6, ATW01_EN, ESP01_CH_PD	PC15	PORT,P-79	EIC,EXTINT-15 PCC,DATA-13 PORT,P-79 SERCOM6,PAD-3/4 SERCOM7,PAD-3/1 TCC0,WO-5/3 TCC1,WO-1/5
7	AUX_7, GND	GND	GND	GND
8	AUX_8, AUX_UART_RX, AUX_SPI_SCK, AUX_I2C_SCL, ATW01_SCK, ESP01_RX	PA16	SERCOM3,PAD-1/3	ADC0,X-10 ADC0,Y-10 CCL,IN-0 EIC,EXTINT-0 GCLK,IO-2 PCC,DATA-0 PORT,P-16 SERCOM1,PAD-0/1 SERCOM3,PAD-1/3 TC2,WO-0/3 TCC0,WO-4/5 TCC1,WO-0/1
9	AUX_9, ATW01_WAKE	PA27	PORT,P-27	ADC0,X-18 ADC0,Y-18 EIC,EXTINT-11 GCLK,IO-1 PORT,P-27
10	AUX_10, ATW01_IRQ	PC01	PORT,P-65	ADC1,AIN-11 EIC,EXTINT-1 PORT,P-65 RTC,IN-4
11	AUX_11, ATW01_GPIO_3	PC10	PORT,P-74	EIC,EXTINT-10 PORT,P-74 SERCOM6,PAD-2/5 SERCOM7,PAD-2/3 TCC0,WO-0/3 TCC1,WO-4/4
12	AUX_12, ATW01_GPIO_1	PC11	PORT,P-75	EIC,EXTINT-11 PORT,P-75 SERCOM6,PAD-3/5 SERCOM7,PAD-3/3 TCC0,WO-1/3 TCC1,WO-5/4

Table 4: VINA-D51 AUX Socket Pinout Detail

14 ARM Cortex SWD Debug Header

The ARM Cortex debug header is a standard 10-pin header for use with hardware programmers and debuggers like the Segger J-Link. If using the Segger J-Link, use it with Segger J-Link 9-Pin Cortex-M Adapter (8.06.02).

14.1 Segger J-Link

BDMICRO uses the Segger J-Link Ultra+ for flashing the initial UF2 bootloader firmware and for source level debugging. The J-Link is available in several models and form factors, including an economical EDU model for non-commercial purposes. The Segger J-Link has several attractive features, notably, it ships

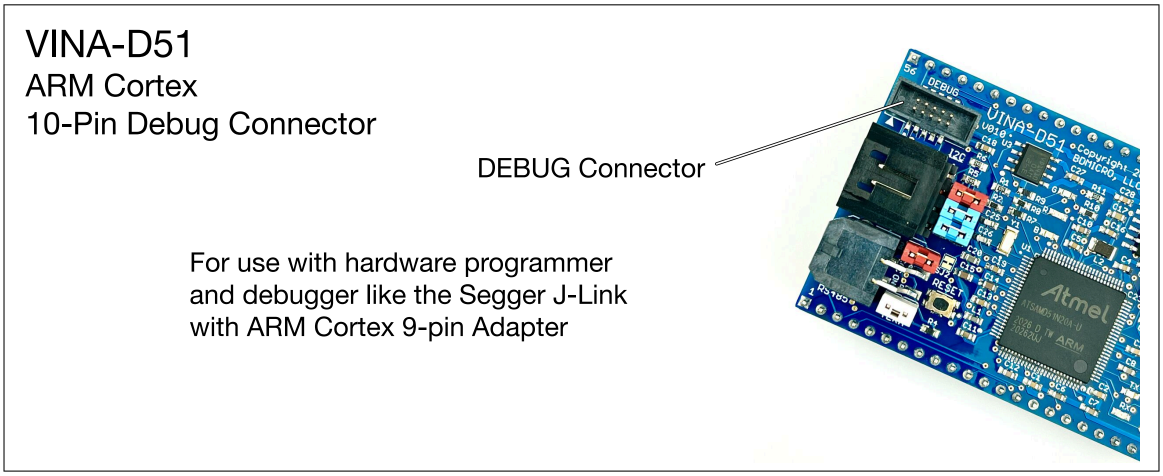


Figure 16: VINA-D51 Debug Header

with the GNU GDB server specifically for the J-Link, includes the J-Link command line utility, and multi-platform support including Windows, MacOS, and Linux.

The J-Link (or similar hardware programmer) enables you to program the flash on the VINA-D51, independently of the pre-installed UF2 bootloader. Under normal circumstances, this is not required since VINA-D51 ships with the UF2 bootloader pre-installed, enabling easy flash programming using MacOS or Windows drag and drop file operations without having to purchase or set up additional hardware and software.

If you wish to replace the UF2 bootloader with another bootloader, or load your application directly without bootloader support, you will need a hardware programmer similar to the Segger J-Link.

> **NOTE** : The UF2 bootloader is capable of self-update to overwrite itself with a newer version of the UF2 bootloader. So you can update the bootloader to a newer version without having to purchase and use a hardware programmer like the Segger J-Link.

14.2 Using the Segger J-Link Command-Line and VINA-D51

The Segger J-Link comes with a command-line tool to, among other things, flash your firmware onto your board, in this case the VINA-D51 with the atsamd51n20 chip. To utilize the command-line tool, connect your J-Link to USB of your development computer, connect the J-Link *Target* side to the 10-pin ARM Cortex Debug Header on the VINA-D51, and supply power to the VINA-D51, either using the external power connector or the USB connector.

We use MacOS for development and by default, the Segger applications are installed in:

```
/Applications/SEGGER/JLink
```

With the command-line utility at:

```
/Applications/SEGGER/JLink/JLinkExe
```

You can write a simple shell script to shorten command-line invokation as follows:

```
#!/bin/ksh
/Applications/SEGGER/JLink/JLinkExe $*
```

Name it something like `jlink` and put it in `$HOME/bin` and make sure `$HOME/bin` is in your `PATH` environment variable.

At this point, you can invoke the Segger J-Link command-line tool by executing your `jlink` short-name script. Once the tool is running, connect to VINA-D51 by typing `connect` at the `J-Link>` prompt. Enter `atsamd51n20` for the device. Select `s` and press return to connect using the SWD protocol, and accept the default target interface speed. You should see something similar to **Figure 17** when connected.

```
bsd — bsd@yyz: ~ — JLinkExe ◂ jlink — 80×59
Last login: Fri Sep 25 10:14:04 on console
[bsd@yyz]:/bsd- jlink
SEGGER J-Link Commander V6.84a (Compiled Sep  7 2020 17:28:57)
DLL version V6.84a, compiled Sep  7 2020 17:28:41

Connecting to J-Link via USB...O.K.
Firmware: J-Link Ultra V4 compiled Jul 17 2020 16:24:36
Hardware version: V4.00
S/N: [redacted]
License(s): RDI, FlashBP, FlashDL, JFlash, GDB
VTref=3.316V


Type "connect" to establish a target connection, '?' for help
J-Link>connect
Please specify device / core. <Default>: ATSAMD51N20
Type '?' for selection dialog
Device>atsamd51n20
Please specify target interface:
  J) JTAG (Default)
  S) SWD
  T) cJTAG
TIF>s
Specify target interface speed [kHz]. <Default>: 4000 kHz
Speed>
Device "ATSAMD51N20" selected.


Connecting to target via SWD
InitTarget() start
InitTarget()
InitTarget() end
InitTarget() end
Found SW-DP with ID 0x2BA01477
DPIDR: 0x2BA01477
Scanning AP map to find all available APs
AP[2]: Stopped AP scan as end of AP map seems to be reached
AP[0]: AHB-AP (IDR: 0x24770011)
AP[1]: AHB-AP (IDR: 0x74770001)
Iterating through AP map to find AHB-AP to use
AP[0]: Core found
AP[0]: AHB-AP ROM base: 0x41003000
CPUID register: 0x410FC241. Implementer code: 0x41 (ARM)
Found Cortex-M4 r0p1, Little endian.
FPUnit: 6 code (BP) slots and 2 literal slots
CoreSight components:
ROMTbl[0] @ 41003000
ROMTbl[0][0]: E00FF000, CID: B105100D, PID: 000BB4C4 ROM Table
ROMTbl[1] @ E00FF000
ROMTbl[1][0]: E000E000, CID: B105E00D, PID: 000BB00C SCS-M7
ROMTbl[1][1]: E0001000, CID: B105E00D, PID: 003BB002 DWT
ROMTbl[1][2]: E0002000, CID: B105E00D, PID: 002BB003 FPB
ROMTbl[1][3]: E0000000, CID: B105E00D, PID: 003BB001 ITM
ROMTbl[1][4]: E0040000, CID: B105900D, PID: 000BB9A1 TPIU
ROMTbl[1][5]: E0041000, CID: B105900D, PID: 000BB925 ETM
ROMTbl[1][6]: E0042000, CID: B105900D, PID: 003BB907 ETB
Cortex-M4 identified.
J-Link>
```

Figure 17: Segger J-Link Command-Line Tool with VINA-D51

From the `J-Link>` prompt, you can load your flash using the `loadbin` command, examine memory and registers, set memory and registers, start and halt the MCU, and much more. Use the `?` command for a full listing of the commands available and their syntax.

14.3 Source Level Debugging using the Segger J-Link and VINA-D51

To utilize source-level, single-step debugging with the Segger J-Link, do the following:

1. Install the `gcc-arm-none-eabi` tools for your platform, and ensure the `bin` directory for the tools is in your PATH.
2. Install the Segger J-Link full software and documentation pack available from Segger for your platform
3. Plug the Segger J-Link into your computer using a USB cable
4. Plug the J-Link 10-pin adapter connector into the debug header of the VINA-D51
5. Ensure your board is powered either through the USB connector or External Power connector
6. Start the Segger JLinkGDBServer for your platform
7. Within the JLinkGDBServer, connect to your Segger J-Link using USB (see **Figure 18**)

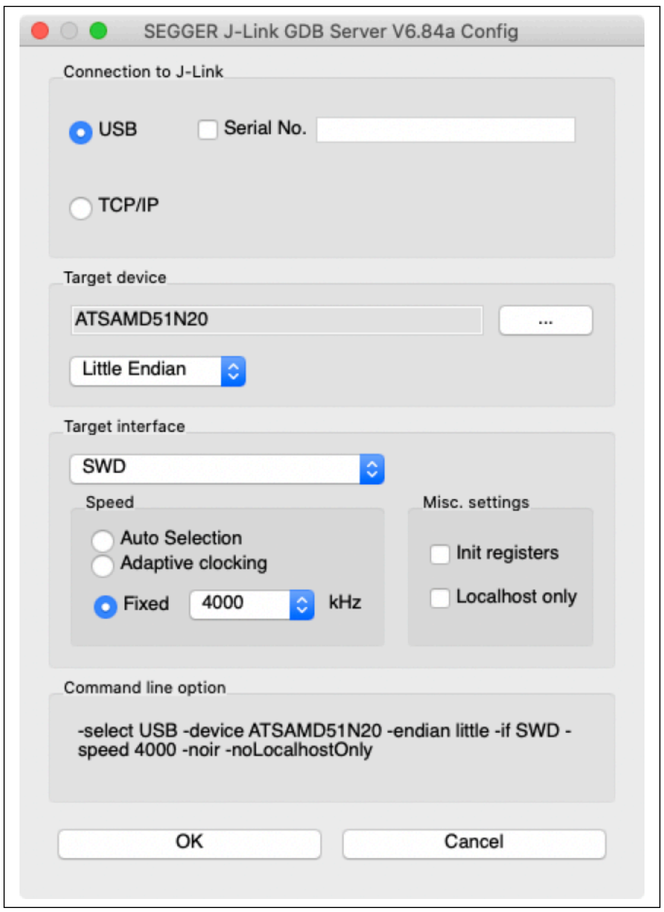


Figure 18: Segger J-Link GDB Server Settings for VINA-D51

   (a) Select the Target device *atsamd51n20*
   (b) Select the Target interface *SWD*
   (c) Leave everything else default
   (d) Click “OK”

8. The JLinkGDBServer will connect to your J-Link and wait for a remote GDB connection (see **Figure 19**)

Figure 19: Segger J-Link GDB Server Waiting for Connection from GDB

9. Bring up a command-line window and start `GDB` using the `arm-none-eabi-gdb` command

10. At the `(gdb)` prompt, connect to the remote JLinkGDBServer using the port that it advertises - note, you can find port as the highlighted line in **Figure 19**:

```
(gdb) target remote localhost:2331
```

11. You should now see `GDB` and JLinkGDBServer connect and negotiate and be returned to the `(gdb)` prompt

12. At this point, you are now debugging the remote target using `GDB` and all of its commands are available such as setting breakpoints, single-stepping, examining variables, back-trace the stack, and all of the other powerful debug capabilities of GDB (see **Figure 20**)

Figure 20: GNU GDB Source Level Debugging with the Segger J-Link Command-Line Tool and VINA-D51

NOTE : To utilize source level debugging you need to compile your application with the -g option to include debug symbols.