SPUD general purpose machine controller

User Manual

RippleTech Ltd

2023-07-23

Revision History

Version	Date	Author(s)	Reviewed by
v0.3	23/7/23	RC	BB
v0.2	16/4/23	RC	BB
v0.1	01/10/22	RC	BB

Contact Details

RippleTech Ltd, 385 State Highway 33, Mourea, Rotorua 3074, New Zealand

Customer Service: info@rippletech.co.nz

Warranty

There are no warranties expressed or implied.

RippleTech Ltd shall not be liable for loss of profits, loss of use, consequential damages, or any other claim based on breach of warranty. Liability for damages of any kind shall be limited to one year after the purchase of the product and to the purchase price of the defective unit.

Notices

> **Notice To Operators** This entire manual should be read before using the SPUD controller.

> **Unsafe Operation** Using the SPUD controller to control a machine or vehicle without a complete and thorough understanding of the capabilities and limitations of the SPUD controller is unsafe and may cause harm. It is important that this manual be read and understood in its entirety before using the controller. Adherence to the statutory regulations regarding accident prevention, occupational safety and environmental protection is also important.

> **Warnings and Cautions Section** Please read the section on Warnings and Cautions carefully before operating the SPUD controller.

> **Use and Maintenance** Any questions regarding installing, operating, or maintaining the SPUD controller should be directed to RippleTech Ltd.

Contents

Introduction

This User Manual contains detailed information and instructions which when adhered to ensure the safe and effective set up, use and maintenance of the SPUD controller. The user should read the **entire** manual before using SPUD .

SPUD is a machine controller that can be used to control industrial machinery, electric vehicles and other machines. This manual describes the hardware that is makes up the SPUD controller, and shows which parts can be configured by the user. Other manuals describe versions of SPUD that have specific configurations and pre-loaded firmware.

The main hardware components of SPUD are as follows:

- 6 solid state relays for DC switching and/or PWM output
- 6 analog inputs with easily modified voltage ranges
- 4 digital input/output channels
- 2 CAN bus interfaces
- ESP32 processor (240 MHz processor, 4MB flash memory, Bluetooth, WiFi)

SPUD can be used to control various types of machine. The user can choose from several off-the-shelf firmware packages or program the onboard micro-controller using standard software.

WARNINGS

Untrained Personnel Only properly trained personnel should operate the controller.

Fire or Explosion Operation of the SPUD controller in the presence of flammable gases could cause a fire or explosion. Under no circumstances is the controller to be operated when explosive gases are present.

⚠ **Maximum ratings** Operation where SPUD is subjected to more than the maximum ratings in the SPUD datasheet may damage the controller and will void any warranty.

Cautions

ESD and EMI Electrostatic discharge (ESD) and electromagnetic interference (EMI) can negatively affect the performance of SPUD.

Overview

The SPUD controller comprises:

- analog inputs
- high power switches capable of high speed switching
- configurable low power digital outputs
- communication interfaces to connect with other computers
- a useful amount of hardware configuration
- a powerful well known microcontroller (ESP32)

Layout

The following image shows the SPUD circuit board. The jumpers — here in yellow — and the rectangular placeholders in the AI channels allow for hardware configuration by the user.

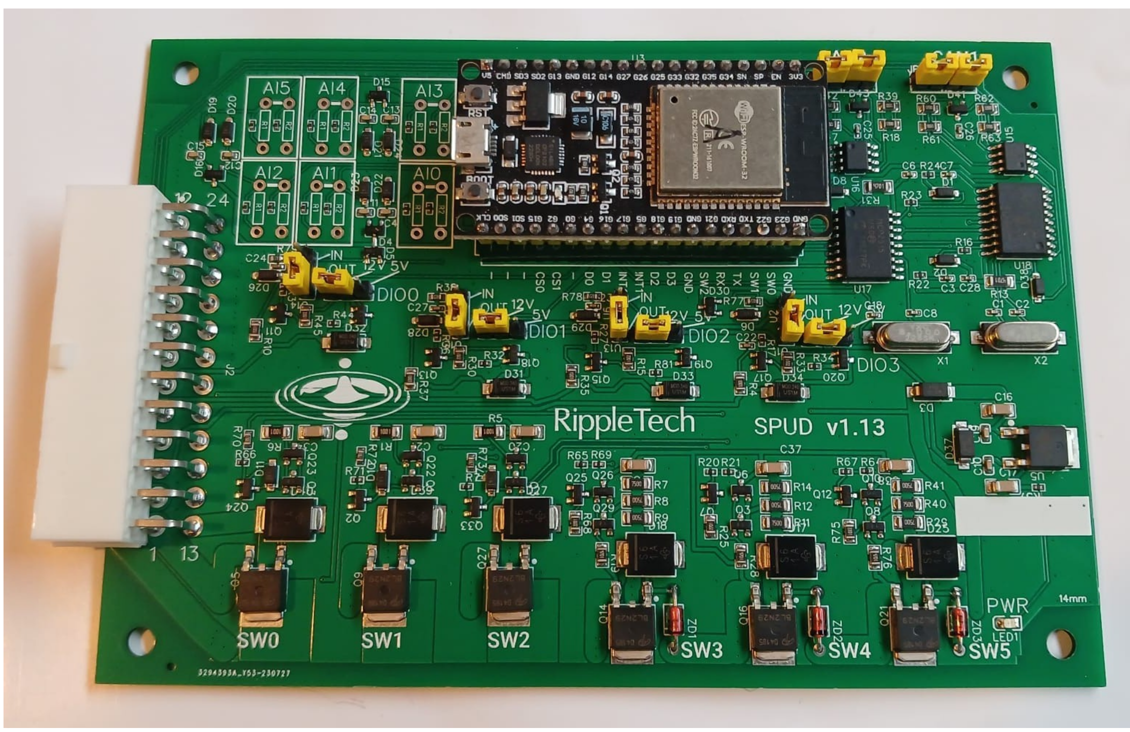


Figure 1: SPUD PCB board. The main connector is shown on the left, and the USB programming connector is on the ESP32 subboard. The AI channels are in the upper left area, the CAN bus interfaces in the upper right area. The high power switches are along the lower edge and the DIO channels in the middle of the board.

Connecting

The front panel of SPUD has a 24 pin connector with the following connections:

- Power (x2)
- GND
- 6 analog inputs
- 4 digital inputs/outputs
- 6 high power switches
- Two CAN bus interfaces

Figure 2: Connecting SPUD to the machine is done via a single connector..

Main connector

The main connector is a male 24-pin Minitek® Pwr 4.2 connector from Amphenol, and the pin out diagram is shown below. A matching harness can be purchased separately or bundled with SPUD .

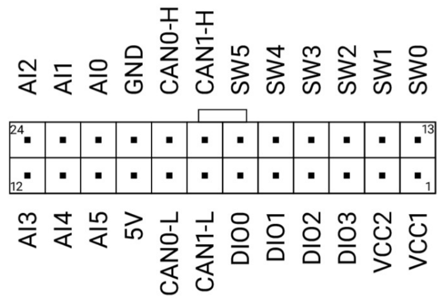
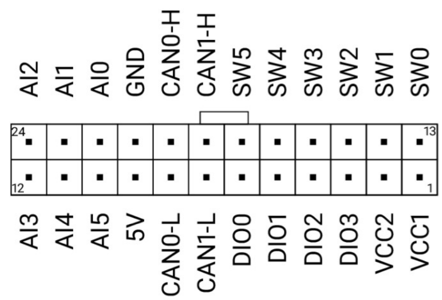


Figure 3: Pinout diagram, as seen when looking at the male connector on the front of the SPUD controller.



A harness can be purchased seperately or with SPUD . This includes the mating connector and fly leads that can be crimped or otherwise connected to the machine harness.

For users that want to build their own harness, the following table lists the required parts and tooling. Note that the Minitek series from Amphenol, which is detailed below, is compatible with the Minifit Jr series from Molex. Standard 24 pin or 20+4 pin ATX power cables can also be used with SPUD . Alternative crimp tools to that shown below are also available.

Item	Part number	Example
MINITEK FEMALE CONNECTOR HOUSING	10127815-24LF	
CONN SOCKET 16-22AWG CRIMP TIN	10127817-221LF	

Item	Part number	Example
CONN SOCKET 24-28AWG CRIMP TIN	10127817-222LF	
TOOL HAND CRIMPER 16-30AWG SIDE	10129483-001LF	

Connecting the USB cable for programming and monitoring

A standard USB (Micro Type-B) cable connects to the ESP32 daughter board that is connected to the main SPUD board. This cable is used for changing program settings, monitoring diagnostic data and uploading firmware to the device. Note that a 4 wire USB cable is needed for settings and firmware updates, and that good quality cables are recommended.

High power switches (SW)

The six high power switches on SPUD are supplied by VCC1 and VCC2 as shown in the figure below and the following table.

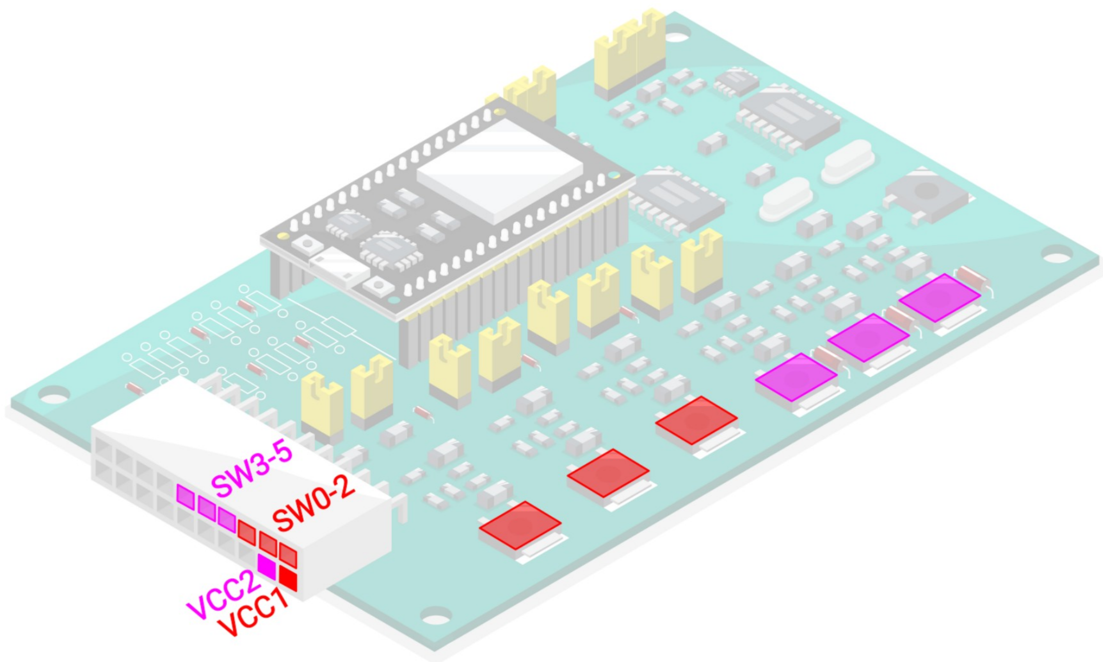


Figure 4: *High power switches and the two supply voltages.*

The two supply voltages can be different, allowing for switching of the following voltage ranges:

Supply	Switches
VCC_1 **7-16 V**	SW0, SW1, SW2
VCC_2 **5-30 V**	SW3, SW4, SW5

The subcircuits that drive the switches are high speed and efficient. This means they can be used as power relays and also as PWM outputs to drive DC motors and other loads. The maximum frequency depends on the type of load being driven, however practical experience demonstrates good square wave generation up to 25 kHz.

Note: the microprocessor and 5V rail is fed from VCC1, so VCC2 may be left disconnected if SW3-5 are not required.

Analog inputs (AI)

SPUD uses some of the Analog to Digital (ADC) channels available on the ESP32 microcontroller. Each analog input is equipped with first order input filters that allow different voltage ranges to be acquired. The as-shipped input voltage ranges can be modified by the user with the use of a soldering iron and a simple procedure.

A preconditioning subcircuit on each analog input provide a first order filter that attenuates each input voltage. The attenuation is constant up to approximately 200Hz, and rolls off above 200 Hz so that high frequencies are filtered out. In the “as-shipped” SPUD controllers the attenuation below 200 Hz gives the following input ranges:

AI Channel	Measurement Range
0	0-5 V
1	0-5 V
2	0-18 V
3	0-18 V
4	0-18 V
5	0-18 V

⚠ Values outside the measurement ranges can damage the ESP32 microcontroller (μC).

For different applications the user may want to change the input range on some or all of the AI channels. This is made possible on SPUD by replacing the as-shipped Surface Mount (SMD) resistors with standard through hole resistors.

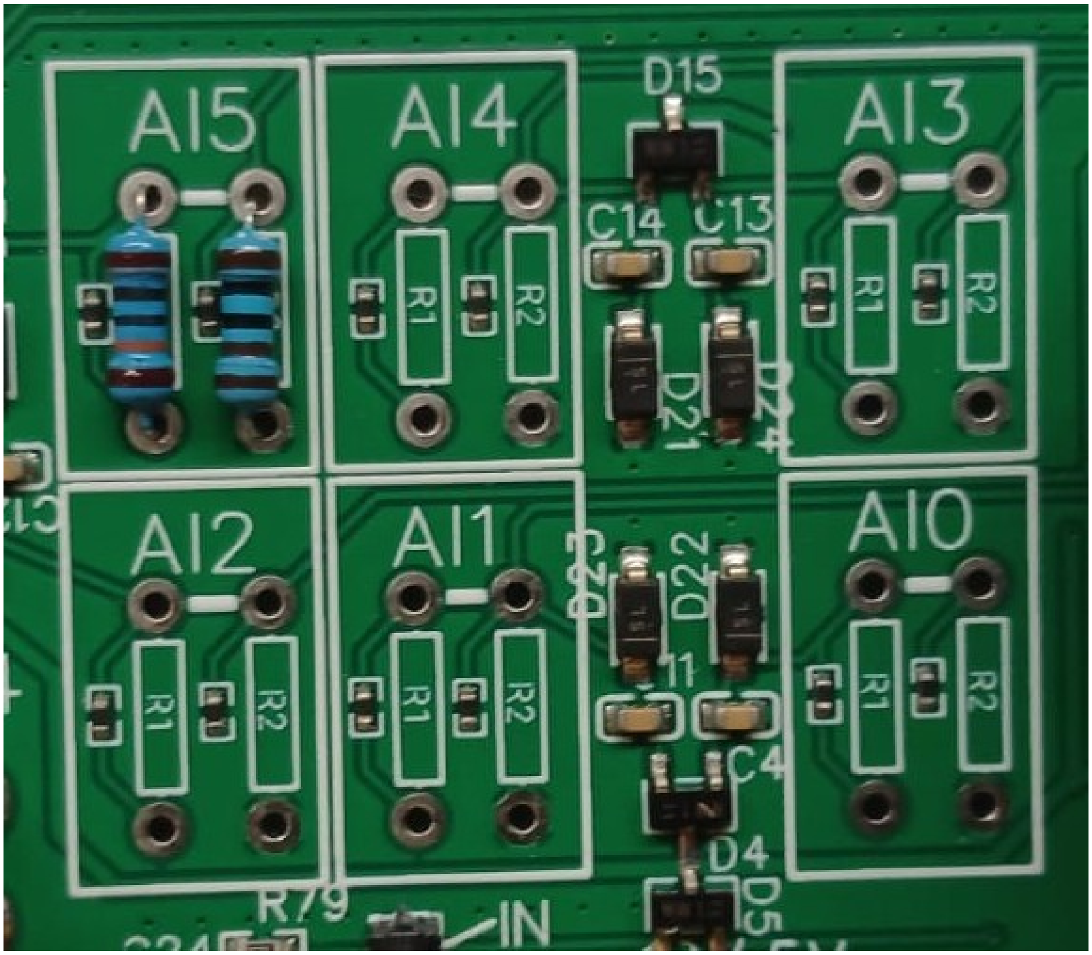


Figure 5: Replacing SMD resistors on AI5.

The as-shipped SMD resistors provide a voltage divider that maps the measurement range for each channel to the 0-3.3 V range that is safe for the microcontroller. The as-shipped resistor values are as follows:

AI Channel	Voltage divider (R1:R2)	AI input range
0	22k:33k	0-5 V
1	22k:33k	0-5 V
2	47k:10k	0-18 V
3	47k:10k	0-18 V
4	47k:10k	0-18 V

AI Channel	Voltage divider (R1:R2)	AI input range
5	47k:10k	0-18 V

The mapping is the standard voltage divider

$$v_{\mu C} = \frac{R_2}{R_1 + R_2} v$$

so for example the 22k:33k divider on AI0 reduces a 5.0 V input to 3.0 V at the ADC channel input:

$$v_{\mu C} = \frac{22000}{33000 + 22000} 5.0 \Rightarrow 3.0V$$

which is within the 3.3 V maximum of the ESP32 analog inputs.

If the user requires different voltage ranges the following method can be used:

1. Choose which resistors need to be replaced
2. Calculate the new resistor values using the voltage divider formula above
3. Remove the SMD resistors that need to be replaced using a soldering iron
4. Solder the new resistors in place
5. Check the voltage attenuation is as expected, starting with a low input voltage

⚠ The filtered signals are protected from voltage surges before being applied to the micrcontroller inputs, however sustained voltages outside the 0-3.3V range will damage the microcontroller and must be avoided.

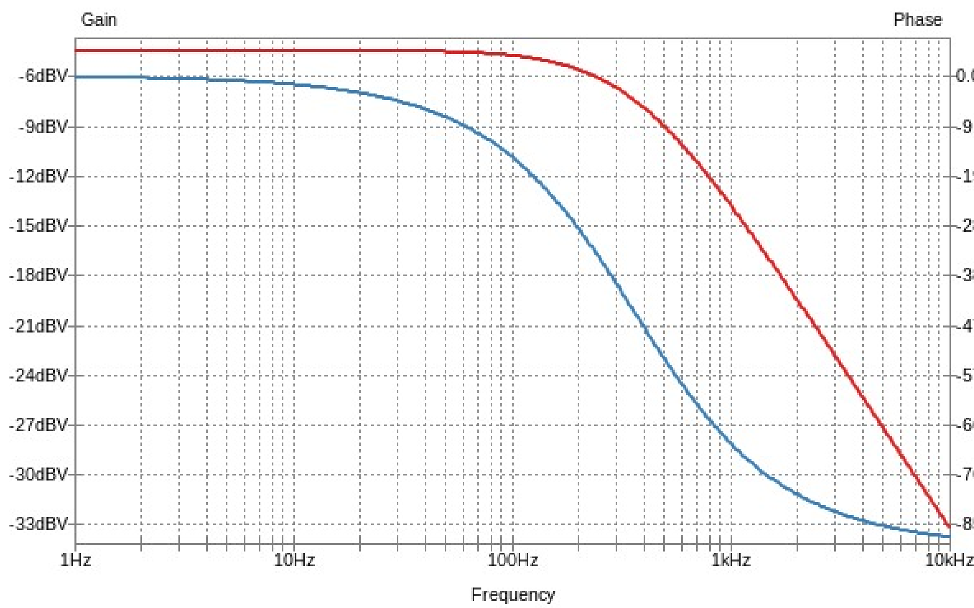


Figure 6: Low pass filter example (SPUD AI0). Magnitude is shown in red and phase in blue.

⚠ While the AI subcircuits have protection devices it is essential that the maximum ratings given in the datasheet are not exceeded.

Digital inputs/outputs (DIO)

SPUD is equipped with four DIO channels that can each be set to either input mode or output mode.

In input mode the voltage on the DIO pins are mapped to digital HIGH and digital LOW:

DIO_I		Microcontroller
0-4V	$\longrightarrow$	Digital LOW
8-18V	$\longrightarrow$	Digital HIGH

In output mode the sub-circuit delivers current on the DIO pins:

Microcontroller		DIO_I
Digital LOW	$\longrightarrow$	0V
Digital HIGH	$\longrightarrow$	5V/12V, max 200 mA

Each DIO subcircuit has two jumpers. One controls whether the pin is an input or an output. The other jumper should only be used if the pin is an output, and it controls whether the output provides 5V or VCC (normally 12V). The following three figures show DIO0 in the three different modes.

⚠ Always disconnect power from the unit before making changes, and ensure that if input mode is selected on a DIO channel that 5V/12V jumper is **removed**. The microcontroller will be damaged if backfed with either 5V or 12V.

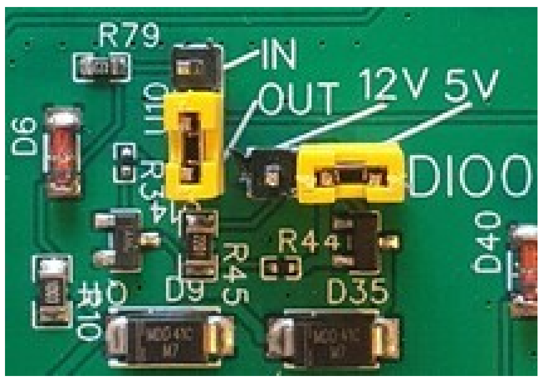


Figure 7: DIO0: 5V output mode.

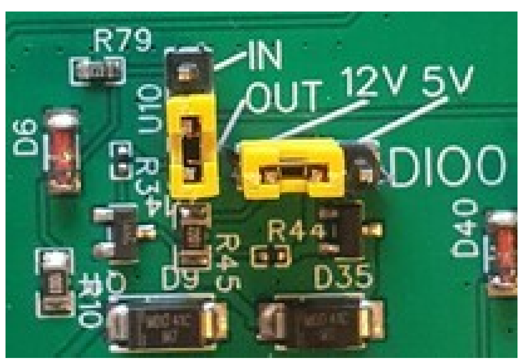


Figure 8: DIO0: 12V output.

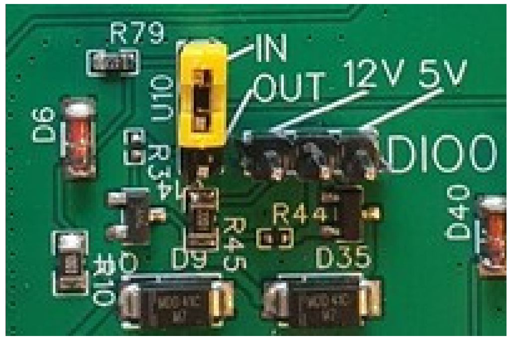


Figure 9: DIO0: input. Ensure the second jumper is removed from the 12V/5V pins.

⚠ While the subcircuits have protection devices it is essential that the maximum ratings given in the datasheet are not exceeded.

CAN bus interfaces

SPUD has two can bus interfaces (CAN0 and CAN1), and is shipped with two jumpers on each interface. When the two jumpers are in place the CAN bus is terminated at this interface with the standard 120Ω resistance. The interfaces should only be terminated if the interface in question is placed at the end of the CAN network. This figure below gives an example of CAN1 being used at the end of the CAN network and CAN0 being used as a “non-terminal” node.

Figure 10: CAN1 as the terminal node on one CAN bus, and CAN0 as a non-terminal node on (probably) another CAN bus.

Programming and loading code

SPUD has been designed around the popular ESP32 microprocessor. This allows many users to use their existing code and knowledge, and also means new users can utilise the ESP32 documentation and tutorials that are available online. The following links are good starting points:

- https://rippletech.co.nz/spud/Spud_Datasheet.pdf
- https://platformio.org/
- https://randomnerdtutorials.com/getting-started-with-esp32/
- https://docs.micropython.org/en/latest/esp32/tutorial/intro.html