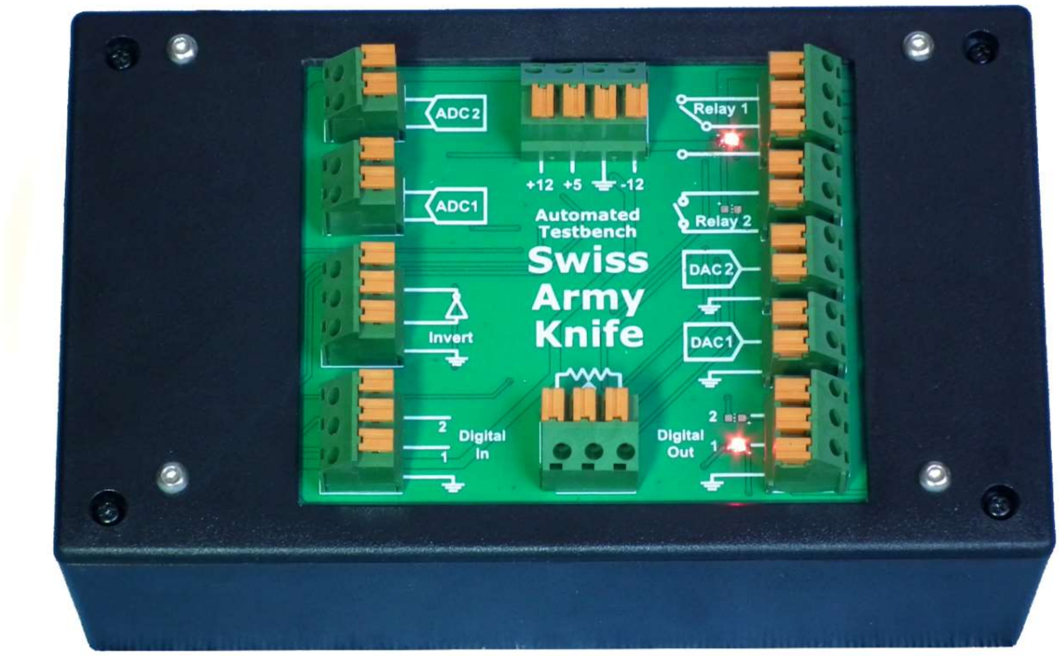
Test Bench Automation
Swiss Army Knife
V1.0 Feb 2022



This manual describes the features of the Swiss Army Knife, a general purpose, remote-controlled test bench aide.

Key Features

- One 0V - 10V DAC
- A 130-55,000 Hz sine generator with two output voltage ranges
- Two fully-differential -10V to +10V input ADCs (20V max difference)
- One analogue inverter with +/- 10V input and output range
- Two 3.3V digital outputs
- Two 5v tolerant digital inputs
- A 10k digital pot with +/-15V terminal range
- One 10A SPDT relay
- One 350mA reed relay
- +/- 15V and 5V power supply rails
- Remote control via serial terminal and WiFi telnet SCPI commands
- Web interface
- TestController device definition file
- 5V @ 1A plug-pack
- Open-source base code (excluding web interface) for roll-your-own additions

Repository: https://github.com/palmerr23/SwissArmyKnife

Contents

Safety

Isolated remote control and monitoring via WiFi is the recommended approach for this instrument.

1. **Once commissioned, do not make a non-isolated USB connection to the instrument.**
    - Differing ground potentials may cause damage unless a USB isolator is employed where a earth differential is present between the computer, this device and any equipment being tested.

      Both the computer and instrument may be damaged in either case.

Safe Operating Area and Protection

The device is optimised for use in controlling and testing equipment that has a voltage range of +/- 10V, and 3.3V or 5V logic.

Moderate protection is provided against voltages outside these ranges.

The device is isolated when run from a plug-pack and takes its ground potential from the device under test (DUT).

Overvoltage and Reverse Voltage Protection

- The analogue and digital inputs and outputs as well as the digital pot are protected against moderate over-voltage and reverse-voltage connections.
- +/- 15V safe operating range for analogue inputs and outputs
- Substantial voltages of either polarity may damage the device.
- 50V should be considered a maximum safe working voltage for any equipment attached to this device, despite some components having higher ratings.

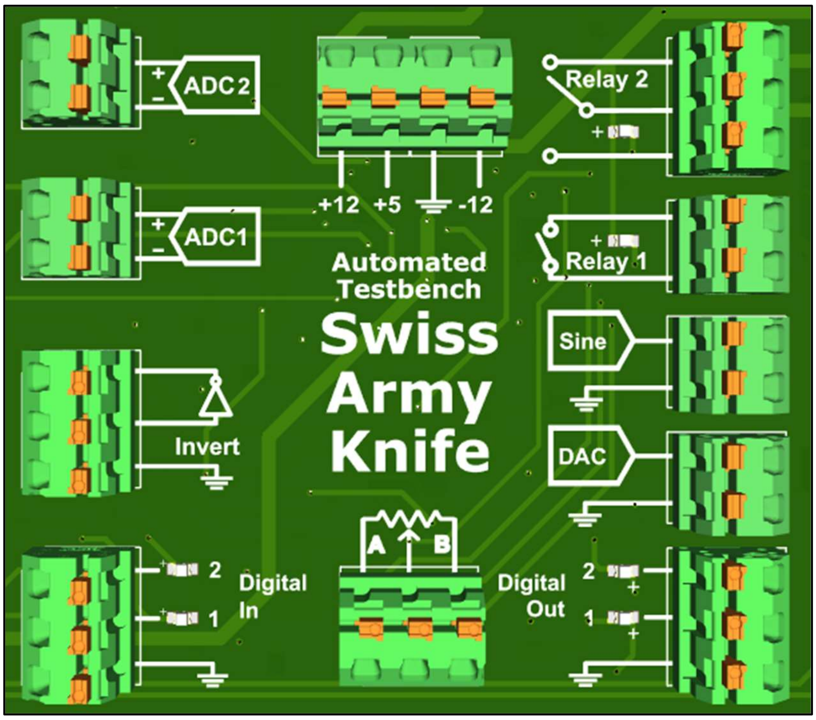
Supply Rails

- The supply rail terminals are not current limited.

  Short circuits may damage the device.

Further information can be found in the Specifications table at the end of this document.

Overview



The Swiss Army Knife offers nine fairly robust and flexible functions. It is intended to be operated via its web interface or using a SCPI-based test bench automation program such as TestController.

Functions

While the functional descriptions below use the web interface as examples, all operations are controllable via SCPI or TestController.

Analogue Inputs

There are two 16-bit fully differential analogue inputs, ADC1 & ADC2 that are able to measure +/- 10V DC to within a few millivolts accuracy after a simple calibration procedure (see below).

The two input terminals are floating and may be connected to any signal within a +/- 10V range (20V maximum differential voltage).

If higher voltages are to be measured, additional series resistance can be added to both inputs. An input resistance of 101K ohms should be used when calculating the voltage divider ratios. E.g., 404k resistances would raise the full-scale voltage to 50V. With re-calibration, 390k resistors could be used.

If ground-referenced measurements are made, the negative input (–) terminal should be connected to ground or erroneous measurements will result.

The inputs are protected against positive or negative over-voltages within the absolute maximum ratings.

DAC Output

The 0-10VDC analogue output has 256 step resolution.

Its value is set as a voltage, and the device will translate (round) this to the nearest step of approximately 40mV.

- If queried with the :SOUR:A1? command, the actual output voltage will be returned.
- The web interface will update each second with the actual output voltage.
- The TestController Setup popup interface will query and display the exact voltage set after the Set DAC button is clicked.
- The TestController Setup popup will not show changes made by other means such as scripting or the web interface until it is closed and re-opened.
- TestController's Current Values and Table views show the actual value.
- The voltage will need to be queried (:SOUR:A1?) in TestController scripts if the actual value is required for calculation.

If negative voltages are required, the analogue inverter may be used.

The buffered DAC output is protected against moderate misadventure by a 100Ω series resistance.

Sine Output

The sine output has a frequency range of 133 Hz to 55kHz and distortion of less than 1.3% across that range. The frequencies available are multiples of 133.33 Hz, and the software rounds settings down to the nearest available frequency.

- The :SOUR:SF? command will report the actual frequency set.
- The web interface will update each second with the actual frequency set.
- The TestController Setup popup interface will query and display the exact frequency set after the Set Freq button is clicked.
- The TestController Setup popup will not show changes made by other means such as scripting or the web interface until it is closed and re-opened.
- TestController's Current Values and Table views read the actual setting.
  The frequency will need to be queried (:SOUR:SF?) in TestController scripts if the actual frequency is required for calculation.

The generator has two selectable output voltages: 6V p-p (2.2V RMS) and 775mV p-p (270mV RMS).

These are settable in scripts using the :SOUR:SS 1 (low) and :SOUR:SS 8 (high = low * 8) commands.

Its output is DC coupled and the crossover is at 0V.

If finer control of the output voltage is required, teaming the sine wave generator with the digital pot can provide 256 discrete voltage steps.

As with the DAC, the sine buffer is protected against moderate misadventure by a 100Ω series resistance.

Analogue Inverter

A general-purpose op-amp inverter is included to provide additional flexibility in handling negative input or output voltages.

Other than series input and output resistors, it is unprotected against misconnection.

Digital Inputs and Outputs

Digital inputs and outputs use ESP32 3.3V GPIO pins with series resistances to limit current if they are misconnected.

Inputs

The digital inputs are weakly pulled down with approximately 50k resistances within the ESP32.

Logic transition points are 0.8V and 2.4V, and should and correctly read most types of 5V logic.

The inputs have LED indicators driven by separate GPIO pins to avoid loading the inputs.

The inputs have Zener diode protection, which also provides 5V tolerance. With 4.7k series input resistors, the current draw is 50nA in all states other than 5V logic high, where up to 0.3mA may be drawn.

Outputs

The outputs have LED indicators and Zener protection against misconnection.

While the outputs can drive up to 12 mA, drawing any significant output current will cause a voltage drop across the 220Ω current limiting resistor.

Logic transition points are < 0.33V and > 2.6V, and should correctly drive most types of 5V logic.

The outputs are tolerant to voltages of up to 50V before the Zener diode's power rating is exceeded.

Digital Potentiometer

All three terminals of the 256 step 10k digital pot are presented.

The digital pot is high-voltage type, which allows the pot terminals to be at any voltage within the +/- 15V analogue supply rails. There is no requirement for any of the terminals to be at higher or lower potential that the others.

If using the pot as a rheostat or in a low-impedance circuit, it should be noted that the wiper has an effective series resistance of 75-170 ohms.

Relays

Relay1 is a 350mA SPST reed relay.

Relay2 is a 10A SPDT relay.

The maximum operating rates is limited by the command processing software, to 16Hz. However, there can be a jitter of 200mS due to delays in transmission and decoding of commands.

Given these constraints, switching rates of less than 1Hz are recommended.

Both relays have LED indicators.

The maximum recommended voltage across both relays is limited to 50V by safety considerations for breadboard-style operation, rather than the characteristics of the relays themselves.

Power Supply

+/-15 V and +5V supply rails are available to power low-current breadboard hook-ups.

There is no short circuit or other protection on the supply rails.

At idle, the unit draws less than 100mA from the +5 supply, and around 200mA with both relays energised and all the LEDs lit.

With the specified boost converter and 1.5A plug pack, 500mA can be supplied at +15V and 200mA at -15V, if minimal breadboard 5V current is consumed.

Changing Communications Settings

WiFi credentials may be changed in two ways:

- Editing profile.json in the Data folder on the computer, and re-uploading it to the ESP32.
- Using the SCPI :SYSTem:SSID, SYSTem:PASSphrase and :SYSTem:HOSTname commands

To satisfy the requirements of IEEE 802.11, the SSID and PASSphrase values are case-sensitive.

- SSIDs may be up to 32 characters in length. While any valid 8-bit value (e.g. special characters) is permitted, non-alphabetic and non-numeric values should be used with care.
- Passphrases must be between 8 and 32 characters long. All 8-bit values are acceptable.

To change the WiFi credentials by editing the Profile.json file:

- The device will need to be re-calibrated after this process.
- Edit the Profile.json in the OTA-test\Data folder on your computer.
- Open the OTA-test program in the Arduino IDE. DO NOT compile the program.
- Using the tools > ESP32 Sketch Data Upload command, upload the files.
- Restart the ESP32 to use the new WiFi parameters.

Web Interface

A comprehensive web interface is provided for the device. All functions except communication parameters can be managed via this interface.

It is accessible from the web address indicated on the serial monitor after start-up, (on the standard Port 80) or via the hostname SwissArmy.local.

The URLs would be:

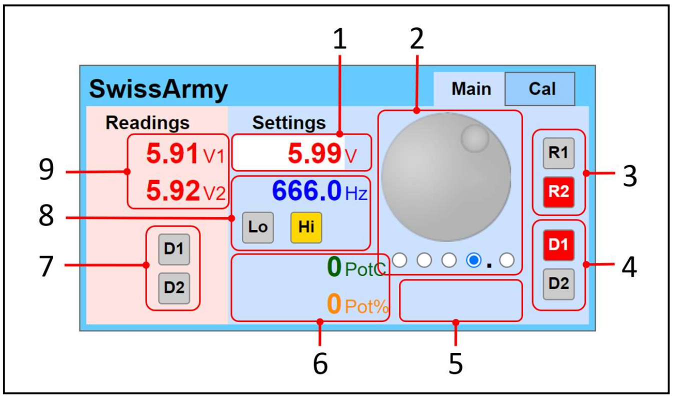
http://192.168.50.1
or
http://SwissArmy.local

Communication is not encrypted.

It is recommended that the hostname address format is used, as DHCP sourced IP addresses may change from time to time. On some WiFi LANs the hostname mode may not work due to router or operating system configurations.

It is recommended that settings are not changed on the web interface when TestController is being used. 'Remote' changes may not be updated in TestController's device Setup pop-up window, as settings are only read from the unit when the pop-up is loaded.

Main Tab



The Main tab has the input readings on the left (salmon background) and settings on the right (pale blue).

To set a numeric value, click on the setting to be changed (e.g. 1) and wind the knob (2).

The radio buttons (2) under the knob determine the size of the increment, from 0.1 to 1,000 units.

Under the sine generator frequency (8) setting are buttons for the low and high output ranges.

The digital pot (6) has two linked scales, one in counts (0 to 255) and the other in percent of rotation. Either may be used and the other will change synchronously.

Relay (3) and digital output (4) buttons are on the far right.

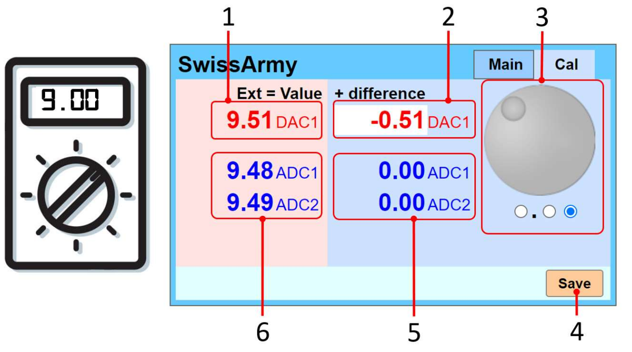
A message window (5) indicates when the web interface is disconnected from the unit, and when control limits have been reached.

Calibration Tab

The Calibration screen allows the DAC output voltage and input voltage readings to be trimmed.

The DAC set value (1) and ADC readings (6) and are on the left (salmon background), and the trimming values (2 & 5) on the right (pale blue).

The Save button (4) stores the offset values (2 & 5) set by the control knob and range radio buttons (3).



The external voltmeter is reading 9.00V, so the DAC will need to be adjusted downwards by 0.51V

Calibration Process

1. Connect the DAC output to
   ADC1 and ADC2 + input terminals.
2. Connect ADC1 and ADC2 – input terminals to ground.
3. On the Main web screen,
   set the DAC voltage close to 9.5V.
4. Connect an accurate DC voltmeter between the DAC output and ground.
5. On the Cal web screen,
   set the difference value for each of the two ADCs inputs and the DAC so that
   the sum of the reading on the left + the difference value = what's on the voltmeter.
6. Make sure that only values you want to change have non-zero *difference* values.
7. Click Save.
   The reading values on the left should update to a value closer to the voltmeter reading.
   The difference values will reset to zero.
8. Rinse and repeat as required.

Calibration attempts are ignored if the voltage reading being calibrated is less than 8V.

The *difference* values are limited to +/- 2 V to avoid mishaps. If the error is greater than 2V, repeat the process until the correct value is reached.

Each calibration setting may be saved separately (see step 6), or several calibrations made each time Save is clicked.

The new calibration values are saved permanently each time Save is clicked.

TestController

Most of the inbuilt functions are controllable from the device Setup pop-up in TestController.

Calibration and communications settings are not supported on the TestController interface.

It is best to avoid changing settings on the web interface while using TestController, as settings are not always updated in TestController if they are changed elsewhere.

Extensive TestController documentation is available at
https://lygte-info.dk/project/TestControllerCommands%20UK.html

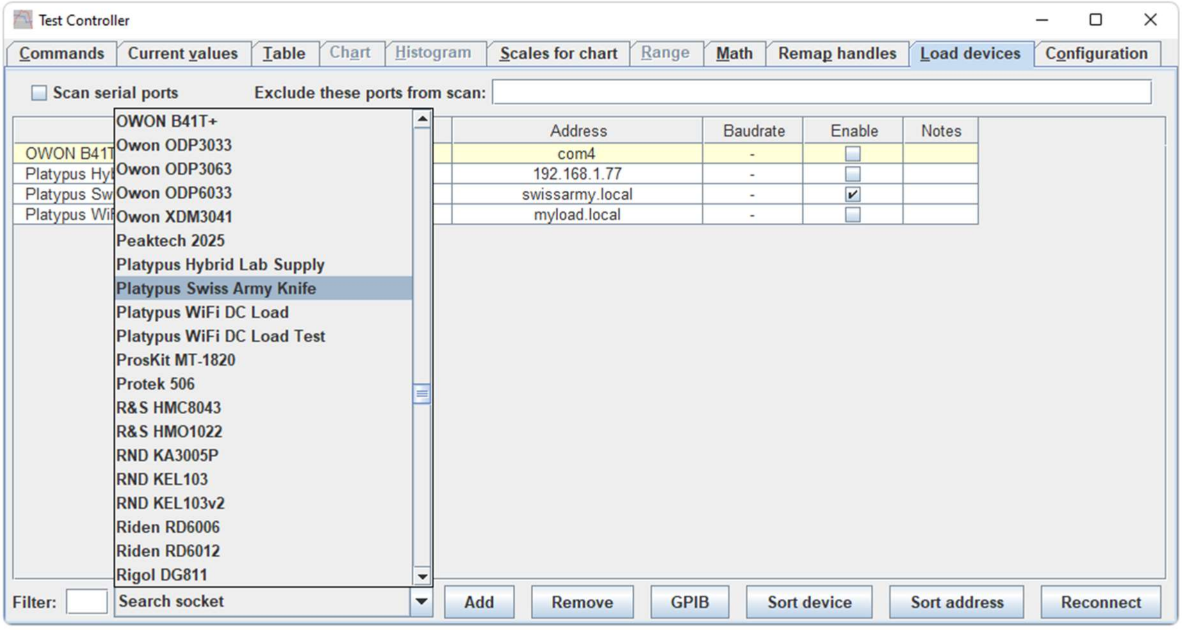
The TestController forum is at
https://www.eevblog.com/forum/testgear/program-that-can-log-from-many-multimeters/

Configuration

- Copy the SwissArmyKnife.txt device definition file downloaded from the repository into the Devices folder at the location you installed the TestController program. This may require Administrator privileges.
- Copy the SwissArmyKnifeHelp.txt file into your Documents > TestController > Settings folder. Once loaded, as you start to type commands, the Help window, below the command line, will display the remaining command options.

The device will be available next time TestController is started.

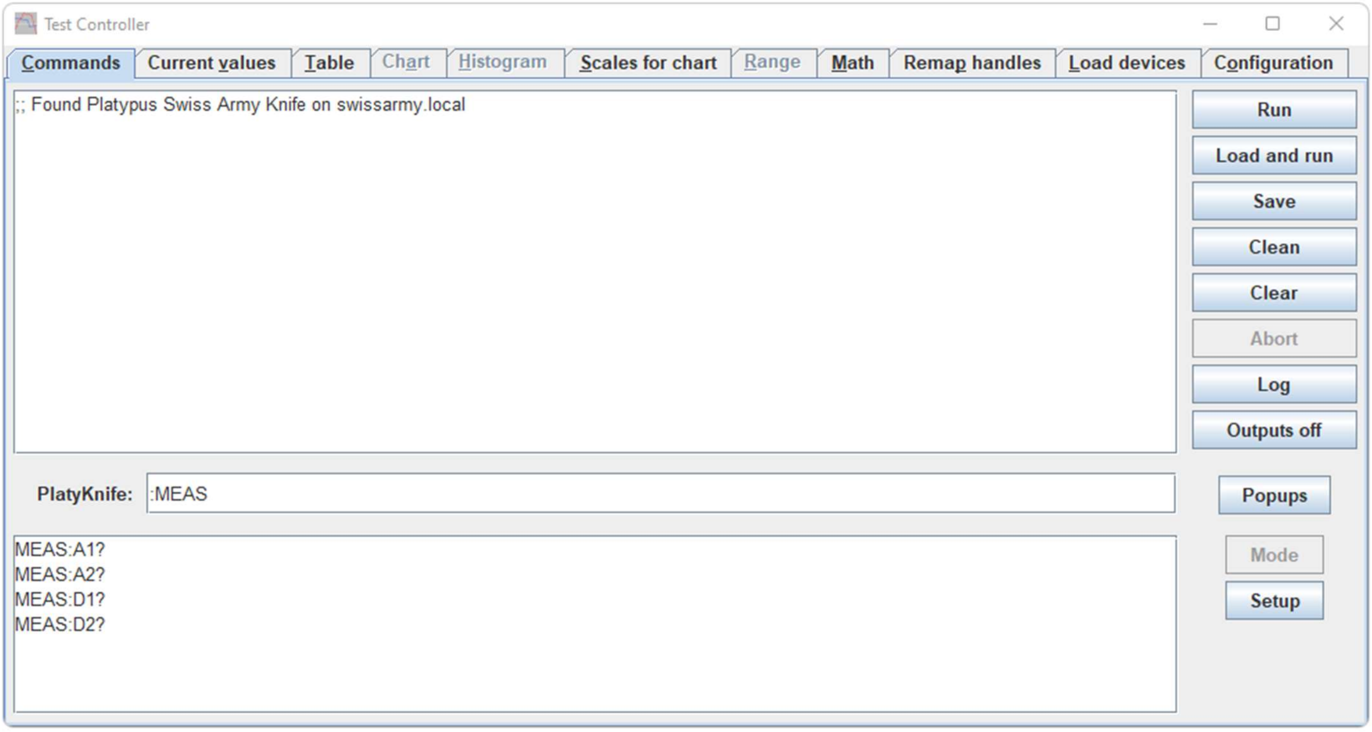
- Add the device on the Load Device tab. If DHCP is being used, the DNS name of the device is a better way to identify its Address, as a device's DHCP assigned IP address may change over time.



The Load Devices tab with several devices added. The DNS name for the device or its IP address can be used.

Operation

All of TestController's functions should operate once the device has been connected.



The main TestController window with the PlatyKnife device selected. There is a partially typed MEASure command in the command line, and the Help widow is displaying the remaining command options.

The device Setup pop-up window

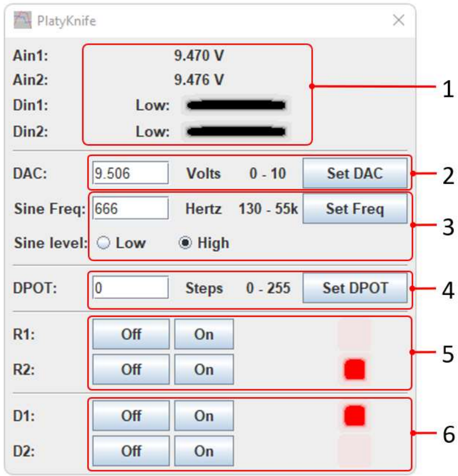
The Setup pop-up window is opened by clicking on the Setup button on the TestController Command tab, when the device is selected (Command Line). Most settings can be managed from this pop-up, excluding calibration and communications parameters.

The device's current *settings* (2, 3, 4, 5 & 6) are loaded into the pop-up when it is opened. They are not updated after that, unless a value is changed using the pop-up's controls.

Therefore, it is not advisable to change settings on the web interface while using TestController, as the current settings may not be displayed in the pop-up window if they are changed elsewhere.

The DAC output (2) is limited to 256 (40mV) steps. The software will set the voltage to the next available value below the setting, and the displayed value is updated.

The sine frequency (3) is limited to steps of 133.33Hz across the range by the ESP32 hardware. The software will set the frequency to the nearest available frequency less than the setting and update the displayed value.



The device pop-up window. Readings are displayed at the top of the screen. Control values are set in the bottom section.

Analogue and digital readings (1), are updated every 2 seconds.

Other values are only updated when a value is changed in the popup, or the popup window is closed and re-opened.

Scripting in TestController

The script sets up the initial values, and then the analogue output value is incremented by 0.25V until the limit is reached.

Each cycle waits for the log entry to be written before updating to the next value.

The loop instructions have been indented for clarity.

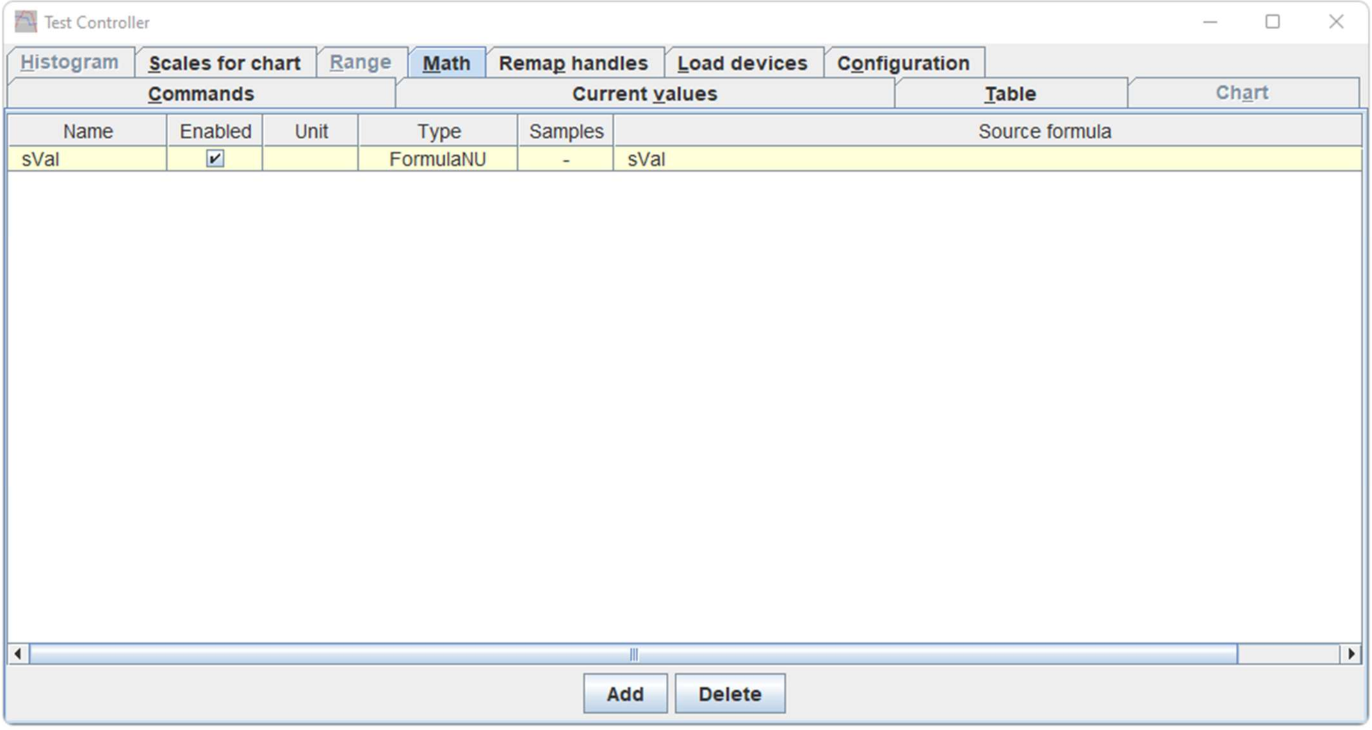
To capture the control variable, sVal, in the log and charts, it must be declared and set, at the top of the code, as a globalvar.

```
; ADC & DAC voltage tracking test
; create a control variable that can be logged
=globalvar sVal=0
; set initial value, let it settle and then wait until value has logged
=sVal=0.0
PlatyKnife::SOUR:A1 0.00
#delay 3
; don't log commands and log values every 3 seconds
#logcmds 0
#log 3
#hasLogged
; each iteration: update analogue output and wait for logging
#while (sVal<10.2)
    PlatyKnife::SOUR:A1 (sVal)
    #hasLogged
    =sVal=(sVal+0.25)
#endwhile
#hasLogged
#log 0
```

In the Math tab the variable is added, Enabled and given the Type FormulaNU.

The Type setting ensures that the correct value is not changed in the Table once it has been captured. See:

https://lygte-info.dk/project/TestControllerMath%20UK.html



The scripted sVal variable will not be available for logging or charting unless it has been added and Enabled in the Math tab.

SCPI Communication

Standard Commands for Programmable Instruments (SCPI) was developed in the early 1990's to provide a common syntax and command structure for programmable instruments from power supplies to oscilloscopes and beyond. It was designed as a master-slave protocol, with the controlling software always being master. While it was originally implemented on the GPIB bus (IEEE 488) other protocols, such as serial (including USB serial) and Telnet, are now commonly used.

The IVI Foundation, which is the successor to the non-profit SCPI Consortium, has a website with exhaustive documentation on SCPI and more recently developed, and more flexible instrument communication protocols such as VISA and VXI. https://www.ivifoundation.org/specifications/default.aspx

SCPI commands can be processed from several sources:

- A TELNET connection on Port 5025, for instance from TestController.
- A command packet from a member of the same *tracking group* as this instrument via UDP packet on Port 8888. (Messages for other tracking groups are read and discarded).

While USB Serial terminal, such as the Arduino Serial Monitor can also be used, significant care needs to be taken to avoid ground loops. (See the safety warning at the beginning of this document). It is highly recommenced that an *isolated* USB serial connection be used to avoid damage to the computer or instrument. USB Serial commands are equivalent to Telnet.

SCPI Command Format

Commands consist of case-sensitive keywords, separated by colons, and each keyword may have parameters associated with it, e.g. ":SOURce:VOLTage 3.6" or ":MEASure:VOLTage?"

Commands ending in a question mark are queries, and the instrument should return a value or set of values to any query.

Parameters may be integers, floating point numbers or strings, depending on the command.

Each command, such as "MEASure" can be issued using the full form or by using an abbreviation, which is always the part in upper case, and almost always four characters long. Thus ":MEAS:VOLT?" is equivalent to ":MEASure:VOLTage?"

Some SCPI rules for this instrument:

- Commands are case sensitive. "ABCX" is not equivalent to "aBCX"
- There *must* be one or more spaces between a command and its argument.
- There *may* be one or more spaces between an argument and its unit specifier
- String arguments are not enclosed in quotes.
- No other spaces are permitted within or between commands. For instance, ":INST:NAME Fred" (no extra spaces) is legal and ": INST : NAME Fred" (extra spaces) is illegal.
- All commands must start with a '*' or a ':', and colons must be inserted between each command in a command line.
- The natural unit (e.g. Volts, Amps) is assumed.
- Floating point number should always have a numeral before a decimal point (leading zero).
- Incorrectly formatted commands will be rejected, with no error message.
- Only one command may be issued on each command line.
- Floating point values are stored as single-precision. Thus, 7 significant figure accuracy is available, but not required.
  Value are generally returned with three decimal places, however accuracy may be limited to less decimal places. See the instrument Specifications table for value limits and accuracy.

In the detailed explanation of commands below

- Square brackets [ ] indicate the type of input required, e.g. [floating point] or [command]
- Angle brackets < > indicate the specific options available, e.g. <ON|OFF> or <CH1|CH2

SCPI Argument and Return Value Types

Data type	Description
Float	Floating point number string. A leading zero is not required for values < 1.0 Actual value set may be constrained by limits embedded in the instrument (noted in individual command descriptions).
Int	An integer value string in the 16-bit signed integer range of -32768 to 32767. May be further constrained by limits embedded in the instrument (noted in individual command descriptions).
String	Upper and lower case alpha numeric, plus keyboard symbols, no spaces (ASCII 0x21 .. 0x7E). No quote marks. Case sensitive.

SCPI Command Quick Reference

Command tree		Argument / Value	Function
*IDN?			Identify instrument
*TST?			Self-test (POST) results
:SOURCce			
	:A1	<float[V]>	Set analogue (DAC) output voltage
	:A1?	<float[V]>	Get analogue (DAC) output voltage setting
	:SF	<int[Hz]>	Set sine frequency
	:SF?	<int[Hz]>	Get sine frequency setting
	:SS	<int[1,8]>	Set sine voltage step
	:SS?	<int[1,8]>	Get set sine voltage step
	:D1 or :D2	<integer 0,1>	Set digital output value
	:D1? or D2?	<integer 0,1>	Get digital output setting
	:R1 or :R2	<Integer 0,1>	Set relay output state
	:R1? or R2?	<Integer 0,1>	Get relay output setting
	:DPOT	<integer 0..255>	Set digital pot value
	:DPOT?	<integer 0..255>	Get digital pot setting
		Digital pot settings in percent are not supported with SCPI commands.	
:MEASure			
	:A1? or A2?	<float[V] 0..10> or overflow	Read analogue input channel
	:D1? or D2?	<Integer 0,1>	Read digital input channel value
:SYSTEM			
	:PASSword	<text> [8-63 chars]	Set WiFi password
	:PASSword?		Not implemented (security)
	:SSID	<text> [2-32 chars]	Set WiFi SSID
	:SSID?		Get WiFi SSID
	:HOSTname	<text> [2-16 chars]	Set network Hostname
	:HOSTname?		Get network Hostname
	:RESTart		Restart Wifi after a COMMS change

SCPI Command List

*IDN?

Command format	*IDN?
Description	Identify the instrument
Return Info	Manufacturer, product type, instrument name/serial number, software version
Typical return	Platy,SwissArmy,00,v0.1

MEASure

Command format	:MEASure:[parameter]?
Description	Query the actual value (reading)

Ax?

Command format	:MEASure:A1? or :MEASure:A2?
Description	Measure analogue input voltage
Return Info	Floating point (Volts) -10.0V to +10.0V Overflow displays -999 or +999
Typical return	3.00 or -2.45

Dx?

Command format	: :MEASure:D1? or :MEASure:D2?
Description	Measure digital input value
Return Info	Integer <high = 1, low = 0>
Typical return	1

SOURce

Command format	:SOURce:[command]
Description	Set or query operational settings, such as voltage, frequency or output value.

A1

Command format	:SOURce:A1 [float]
Description	Set the analogue output voltage values are [0 to 10.0V]
Example	Set the target voltage to 2.50V :SOUR:VOLT 2.50

A1?

Command format	:SOURce:A1?
Description	Query the analogue output voltage setting
Return Info	Floating point, in Volts. Overflow displays -999 or +999
Typical return	8.43

SF

Command format	:SOURce:SF [int]
Description	Set the sine frequency values are [133 to 55,000] Hz The frequency is rounded down to the nearest 133.33 Hz step
Example	Set the target frequency to 1500 Hz :SOUR:SF 1500

SF?

Command format	:SOURce:SF?
Description	Query the sine frequency setting
Return Info	Integer in Hertz Actual frequency is reported.
Typical return	1500

SS

Command format	:SOURce:SS [int]
Description	Set the sine output voltage step values are [1,8] (The high voltage setting is 8 times the low value)
Example	Set the target frequency to the high output voltage :SOUR:SS 8

SS?

Command format	:SOURce:SS?
Description	Query the sine output voltage step
Return Info	Integer [1,8] (The high voltage setting is 8 times the low value)
Typical return	1

Dx

Command format	:SOURce:D1 [int] or :SOURce:D2 [int]
Description	Set the digital output value Integer <high = 1, low = 0>
Example	:SOUR:D1 1

Dx?

Command format	:SOURce:D1? or :SOURce:D2?
Description	Query the current target voltage setting
Return Info	Integer <high = 1, low = 0>
Typical return	1

Rx

Command format	:SOURce:R1 [int] or :SOURce:R2 [int]
Description	Set the relay state Integer <on = 1, off = 0>
Example	:SOUR:R1 1

Rx?

Command format	:SOURce:R1? or :SOURce:R2?
Description	Query the relay state
Return Info	Integer <on = 1, off = 0>
Typical return	1

DPOT

Command format	:SOURce:DPOT [int]
Description	Set the digital pot wiper position Integer <LH = 0, RH = 255>
Example	:SOURce:DPOT 128

DPOT?

Command format	:SOURce: DPOT?
Description	Query the digital pot wiper position
Return Info	Integer <LH = 0, RH = 255>
Typical return	127

Note: Digital pot settings in percent are not supported with SCPI commands.

SYSTem

Command format	:SYSTem:[command]
Description	Set or query system variables such as hostname and WiFi credentials

SSID

Command format	:SYSTem:SSID [string]
Description	Set the SSID of a preferred WiFi LAN The SSID must be 8 – 32 characters in length to conform with the 802.11 specification. The argument string is case sensitive and not enclosed in quotes. Restart, or follow this command with :SYST:RESTart to change the web interface URL.
Example	:SYST:SSID MyHomeWiFi

SSID?

Command format	:SYSTem:SSID?
Description	Returns the SSID of the connected WiFi LAN
Typical return	MyHomeWiFi

PASSphrase

Command format	:SYSTem:PASSphrase [string]
Description	Set the passphrase for the last entered SSID. The argument string is case sensitive and not enclosed in quotes. The passphrase must be 8 – 32 characters in length to conform with the 802.11 specification. Restart, or follow this command with :SYST:RESTart to change the web interface URL.
Example	:SYST:PASS MYpAssWord23

PASSphrase?

This command is not implemented, as it poses a security vulnerability for the WiFi network.

Command format	:SYSTem:PASSphrase?
Description	Returns a descriptive message.
Typical return	WiFi password can only be SET remotely.

HOSTname

Command format	:SYSTem:HOSTname [string]
Description	This command sets both the name displayed on the web interface and the hostname component of its URL. The argument string should not be enclosed in quotes. The URL generated is not case sensitive and will be of the form hostname.local The unit may be addressed name as well as its IP address. *The hostname will not change until the WiFi server is re-initialised.* Restart, or follow this command with :SYST:RESTart to change the web interface URL.
Example	:SYSTem:HOSTname myKnife

HOSTname?

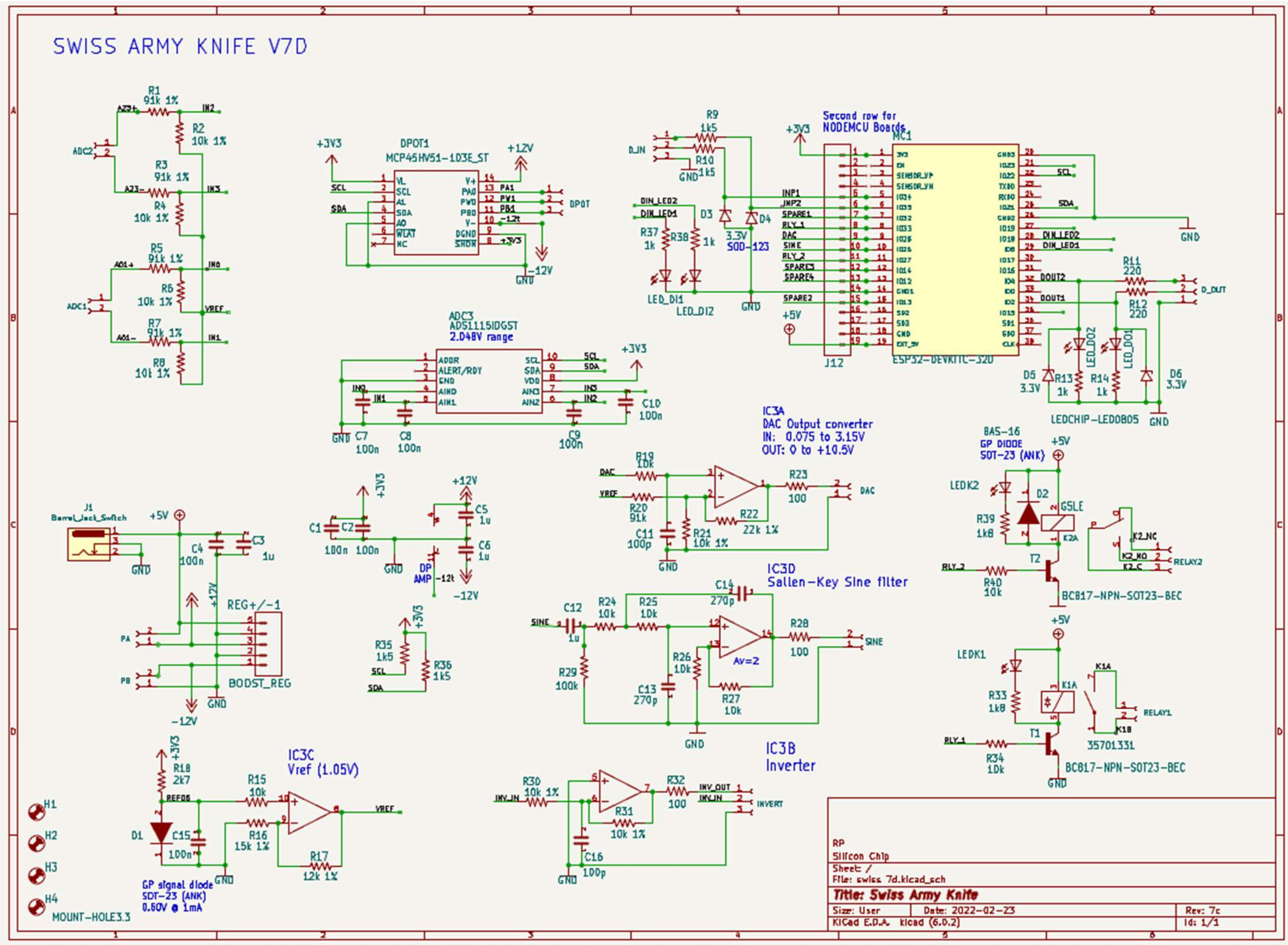
Command format	:SYSTem:HOSTname?
Description	Return the hostname (same as Instrument name) component of the URL
Typical return	myKnife

RESTart

Command format	:SYSTem:RESTart
Description	Restart the web interface. Used after changing HOSTname, SSID or PASSphrase
Example	:SYST:REST

Design Overview

The design and construction of this project is described in greater detail in the 2022 Silicon Chip article.



The ESP32 handles all communication. Its DAC and cosine generator are buffered for the analogue outputs.

Analogue inputs are divided by a factor of 10 and referenced to 1.0V, which is mid-range on the ADS1115 ADC's 2.048V scale. The ADS1115 has input protection specified for up to 10mA input currents.

The DAC output is amplified to 10V p-p by IC3A, with a 75mV zero offset corrected using the 1.0V reference voltage.

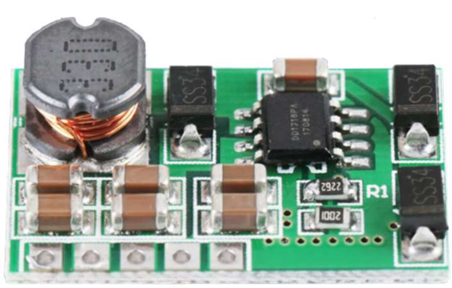
The sine generator output is smoothed and amplified by a Sallen-Key filter with a corner frequency of 75kHz.

Digital inputs and outputs are protected by Zener diodes which also enable 5V logic tolerance on the digital inputs.

LED indicators are used for the digital inputs and outputs and relays.

The reed relay, K1, has an inbuilt coil diode.

A 5V to +/-15V boost switching module powers the op-amp and the power terminals.



ESP32 pinout

ESP32 IO pins are not 5V tolerant. Only 0 – 3.3V signals should be connected directly to inputs.

GPIO outputs have a 12mA maximum current rating.

DAC and Cosine outputs are high impedance and shouldn't be loaded to less than 10kΩ.

GPIO	Function
2	Dout2
4	Dout1
21	SDA
22	SCL
25	Aout1
26	Sine
27	R1
33	R2
34	Din1
35	Din2
5	Din1 LED
18	Din2 LED

A Bill of Materials and construction details are included in the associated Silicon Chip article published in 2022.

Specifications

Item	Value
All inputs	Analogue working voltage: +/-15V Digital working voltage: 0 to 5V Absolute maximum: +/-50V
Analogue Input Voltage	Operating: -10 to +10V either input (+/- 20V differential reading within these limits) Resolution: 10mV Linearity: better than 1% (above 150mV)
Analogue Output Voltage	Operating: 0 to +10V Resolution: 40mV steps, rounded down by software Linearity < 2% (above 150mV)
Sine Output	Frequency: 133 to 55,000 Hz 133.33 Hz steps, rounded down by software. Voltage: High range 6.2V p-p (2.2V RMS) Low range: 775mV p-p (270mV RMS) Distortion: < 1.3% (high range)
Digital Inputs	3.3V logic, 5V tolerant 3.3V logic current: 50nA 5V logic current: < 0.3mA high, 50nA low
Digital Outputs	3.3V logic Maximum drive: 12mA High output: > 2.6V Low output: < 0.33V
Relays	Relay 1: 350mA, SPST, 50V (10mS operation) Relay 2: 10A, SPDT, 50V (100mS operation) Switching rate: 16Hz max > 1Hz recommended for acceptable jitter Jitter: 200mS (transmission and software decoding)
Digital Pot	Taper: Linear Steps: 255 Resistance: 10k Ohms Wiper resistance: < 170 ohms Terminal voltage range: +/-15V
DC input	5VDC 1A min (1.5A preferred) double insulated.
Dimensions	142mm X 83.5mm

Communication and remote control

Command processing time	
Average time for control change or response after a single command is issued	< 50mS Command execution jitter may be up to 200mS, for streams of commands due to network and processing delays.
SCPI commands	
Units accepted	No unit values or multipliers are accepted. • All analogue values are in volts [floating point or integer] • All digital values are integer [0 = Off or Low; 1 = On or High] • The sine generator output level is integer and only accepts 1 and 8. • Strings are not enclosed in quotes for SSID, PASSWORD or HOSTNAME
Remote control connections	
WiFi	802.11 b/g/n/e/i Auto-connect to the SSID/passphrase combinations set in the profile.json file or by SCPI :SYSTEM: commands. WPS is not supported. IP addresses are exclusively gained from WiFi LAN DHCP. The current IP address is displayed at startup on the Serial Monitor (115,600 baud). DNS name is SwissArmy.local (via an mDNS responder) unless the HOSTname has been changed via the SCPI command :SYST:HOSTname
Telnet	Port: 5025 SCPI commands accepted and results returned. Functionality has been tested with the open source software TestController Studio, and an instrument definition file for this software is included in the project's downloads.
Web (HTTP)	A web interface allows most settings can be altered and most actions initiated. Calibration and Communication settings are not available via the web interface. Port: 80 http://IPaddress **The web page is and not secured or encrypted.**
Isolated USB Serial	**Damage may occur if a non-isolated USB connection is used.** Baud rate: 115,200 SCPI commands accepted and results returned.

Troubleshooting

Symptom	Likely cause	Remediation
Voltage reading or output incorrect.	Calibration not completed.	See the *Calibration* sections.
Can’t access instrument (Web page or Telnet) remotely.	Wifi not connected. Network parameters incorrect	See *Changing Communication Settings* and *Specifications – WiFi* sections.
Can’t access web interface using hostname.	Duplicate hostname. On some WiFi LANs the hostname mode may not work due to router or operating system configurations.	Use the IP address found in the serial monitor startup messages.