



# USER MANUAL

---

Last Update: Oct 31/2021

© 2018-2021 NewAE Technology Inc. All rights reserved. Specifications are subject to change without notice. All product names are trademarks of their respective companies. ChipSHOUTER is a registered trademark of NewAE Technology Inc.

NewAE Technology Inc. makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. NewAE Technology does not make any commitment to update the information contained herein. NewAE Technology products are not intended, authorized, or warranted for use as components in applications intended to support or sustain life. NewAE Technology products are designed solely for teaching purposes.

## LIMITED WARRANTY AND LIMITATION OF LIABILITY

Each NewAE Technology Inc product is warranted to be free from defects in material and workmanship under normal use and service. The warranty period is one year and begins on the date of shipment. This warranty extends only to the original buyer or end-user customer of a NewAE Technology Inc authorized reseller, and does not apply to probes (including EMFI injection tips), exposed circuit boards, fault injection targets, or to any product which, in NewAE Technology Inc's opinion, has been misused, altered, neglected, contaminated, or damaged by accident or abnormal conditions of operation or handling (including failing to observe required ESD handling procedures).

Authorized resellers shall extend this warranty on new and unused products to end-user customers only but have no authority to extend a greater or different warranty on behalf of NewAE Technology Inc. NewAE Technology Inc.'s warranty obligation is limited, at NewAE Technology Inc.'s option, to refund of the purchase price, free of charge repair, or replacement of a defective product which is returned to a NewAE Technology Inc. within the warranty period. To obtain warranty service, contact NewAE Technology Inc.

If NewAE Technology Inc. determines that failure was caused by neglect, misuse, contamination, alteration, accident, or abnormal condition of operation or handling, including failures caused by use outside the product's specified rating, or normal wear and tear of mechanical components, NewAE Technology Inc will provide an estimate of repair costs and obtain authorization before commencing the work.

THIS WARRANTY IS BUYER'S SOLE AND EXCLUSIVE REMEDY AND IS IN LIEU OF ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. NEWAE TECHNOLOGY INC SHALL NOT BE LIABLE FOR ANY SPECIAL, INDIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OR LOSSES, INCLUDING LOSS OF DATA, ARISING FROM ANY CAUSE OR THEORY.

Since some countries or states do not allow limitation of the term of an implied warranty, or exclusion or limitation of incidental or consequential damages, the limitations and exclusions of this warranty may not apply to every buyer. If any provision of this Warranty is held invalid or unenforceable by a court or other decision-maker of competent jurisdiction, such holding will not affect the validity or enforceability of any other provision.

NewAE Technology Inc.  
127 Joseph Zatzman Dr  
Dartmouth, NS. Canada  
sales@newae.com 1-888-GLITCHY (US/CA) or +1 902 800 8880

## Table of Contents

Introduction.....	5
Safety Information .....	6
High Voltage Warnings .....	9
Packing Information .....	11
Background and Quick Start Guide .....	14
Device Architecture .....	15
Specifications.....	17
General Specifications .....	17
I/O Characteristics .....	18
High Voltage Characteristics .....	18
Pulse Source Characteristics .....	19
Inserted Pulse Characteristics .....	19
External Connections .....	22
SMA High Voltage Output .....	22
Attaching/Removing SMA Connectors .....	23
SMB Trigger Input .....	23
External Trigger & Self-Checks with triggersafe .....	25
External Trigger & ChipWhisperer Glitch Output .....	26
RJ12 Expansion Connector .....	27
DC Power Jack .....	27
Oscilloscope Probe Connectors .....	28
Pulse Generation .....	29
Generated Pulse vs. Inserted .....	29
Active-High vs. Active-Low Inputs .....	29
Basic Pulse Generator .....	30
Programmable Pulse Generator .....	30
Simple EMFI Target (CW322) .....	32
Ballistic Gel EMFI Target (CW522) .....	35
Injection Tip Usage .....	38
Avoiding Spark Discharge .....	40
Polyimide Tape Protective Layer .....	40
Oscilloscope Pulse Shape Monitoring .....	41
Adjusting for Oscilloscope Setting .....	43
Forced-Air Cooling .....	45
Fault Modes.....	48
Probe Disconnected Fault .....	50
Over-Temperature Fault .....	50
Triggered when Disarmed .....	51
Trigger Length Invalid .....	51
Internal Faults .....	52
Serial Interface .....	53

Command List .....	54
USB Interface .....	61
Saving Settings Between Power Cycles .....	63
Python API Interface .....	64
XY(Z) Table Connection .....	65
Troubleshooting .....	66
Schematic.....	68

## Table of Figures

Figure 1: Overview of ChipSHOUTER device architecture.....	15
Figure 2: 4mm tip pulse width.....	20
Figure 3: 1mm tip pulse width.....	21
Figure 4: External connectors on the ChipSHOUTER.....	22
Figure 5: RJ12 Connector on ChipSHOUTER Panel.....	27
Figure 6: CW322 Simple Target.....	32
Figure 7: CW522 Ballistic Gel.....	35
Figure 8: Detail of the 4MM/CCW tip construction.....	38
Figure 9: Detail of included probes.....	39
Figure 8: Inserted pulse viewed on oscilloscope screen.....	41
Figure 11: Tuning oscilloscope probe.....	43
Figure 10: Example calibration waveform.....	44
Figure 11: Removing blanking plug.....	46
Figure 14: Adding air inlet adapter.....	46
Figure 15: USB Interface for ChipSHOUTER.....	61

## Introduction

The CW520 (ChipSHOUTER) is a fully-featured Electromagnetic Fault Injection platform that can be used to discover and characterize vulnerabilities in embedded systems. ChipSHOUTER makes EMFI available to test labs, engineering development firms, educators, and embedded enthusiasts. With a flexible API and bundled practice targets the system is a platform for experimentation and education right out of the box. Paired with an X-Y table and some basic python scripting the ChipSHOUTER becomes a fully automatable EMFI platform capable of precision testing and fault characterization. This manual will give a basic background for the principles behind the device, using it safely, and example injections on included targets. Users will also be directed to further reading on advanced uses, where professionals and researchers can take advantage of the modular design to further fine tune their processes and experiments.



## Safety Information












CAREFULLY READ BOTH THE FOLLOWING GENERAL SAFETY INFORMATION, AND SAFETY INFORMATION IN THE SECTION ENTITLED “HIGH VOLTAGE WARNINGS”:

- This product generates strong electronic and magnetic fields:
  - DO NOT use around persons with implanted or attached medical devices such as pace-makers, implanted defibrillators, or medication pumps.
  - DO NOT use around safety-critical devices, or anything where interruption of device function would be undesirable.
- DO NOT touch the injection tip or high voltage connector when device is armed or discharging.
- DO NOT aim or position the injection tip onto a person or other living tissue.
- This product is capable of PERMANENTLY DESTROYING devices under test.
- This product is capable of PERMANENTLY DAMAGING devices under test. NEVER return a tested device to service, even if it appears operational, as the functionality of this device could be affected.
- DO NOT operate the product with covers removed or the case open. Hazardous voltage exposure is possible.
- IF you hear or notice electrical discharge, immediately discontinue operation and remove power

from the ChipSHOUTER by unplugging the power source. Check connections are secure and for damage to the probe. If probe is damaged destroy and discard it, and replace with an undamaged probe.

- IF you notice smoke or unusual odors emitted from the ChipSHOUTER, immediately discontinue operation and remove power from the ChipSHOUTER by unplugging the power source. Store the device where it cannot accidentally be used, and contact us for repair or replacement information.
- DO NOT operate the product with the air inlet cover removed without connecting an air hose. If an air hose is removed immediately replace the air inlet cover.
- Repairs must only be performed by an approved technician.
- DO NOT expose the ChipSHOUTER to water or other liquids, DO NOT submerge the ChipSHOUTER in water or any liquid, and do not use ChipSHOUTER as a bath toy under ANY circumstances.
- Keep ChipSHOUTER away from children and especially smart dogs.
- DO NOT use ChipSHOUTER or any accessories if they appear damaged in any way, paying careful attention to the insulation on the injection tips.
- CAREFULLY READ the high voltage warnings section.
- Familiarize yourself with the warning and label pictures from the table below.

## ChipSHOUTER Users Manual: Safety Information

Symbol	Description	Symbol	Description
	WARNING. RISK OF DANGER.		WARNING. HAZARDOUS VOLTAGE. Risk of electric shock.
	Consult user documentation.		DC (Direct Current)
	Conforms to European Union directives.		AC (Alternating Current)
	For indoor use only.		Do not disassemble unit.
	This product complies with the WEEE directive marking requirements. The affixed label indicates that you must not discard this electronic product in domestic household waste. Product Category: With reference to the equipment types in the WEEE Directive Annex I, this product is classed as category 9 “Monitoring and Control Instrumentation” product. Do not dispose of this product as unsorted municipal waste. Please contact us to dispose/recycle this product.		



## High Voltage Warnings



In addition to the safety warnings regarding the ChipSHOUTER operation, there are some specific additional warnings related to the high voltage circuitry. Please carefully read both the “Safety Information” in addition to these “High Voltage Warnings”. All users of the ChipSHOUTER must be aware of these warnings.



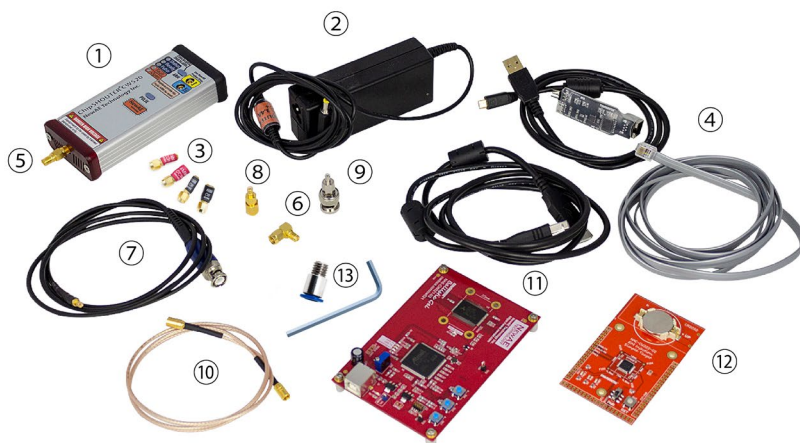
**ChipSHOUTER can generate hazardous voltages. It is very important that everyone who will be operating the ChipSHOUTER carefully reads and understands this manual and the warning instructions. If you have questions about these warnings please contact NewAE immediately.**

- ChipSHOUTER can generate strong magnetic and electrical fields. DO NOT use around safety-critical equipment. DO NOT allow a person with an implanted or on-body medical device near the ChipSHOUTER.
- The SMA center pin has hazardous voltage present. DO NOT touch or otherwise expose this connection.
- DO NOT touch the injection probe or high voltage connector when device is armed or discharging.
- DO NOT attempt to arm the ChipSHOUTER without a EMFI injection probe attached.
- DO NOT use the ChipSHOUTER to generate a spark-gap discharge. In addition to exposing hazardous

voltages, this may generate U.V. light and other dangerous radiation. ChipSHOUTER will also be severely damaged during the discharge process, as the spark-gap discharge exceeds allowed  $dv/dt$  ratings of the driver circuit.

- The insulation on the injection probes must be unbroken for your protection. Carefully inspect the probes for damage to the insulation, and destroy (to prevent accidental reuse) and discard any damaged probes.
- DO NOT position the injection probes in such a manner they will scrape conductive areas of the device under test.
- The SMA connector shell is NOT connected to the enclosure (chassis). Do not short the SMA connector shell to the enclosure or ground, as otherwise high voltages and currents could pass through this connection.
- Do not connect anything besides a EMFI injection probe or included accessory to the ChipSHOUTER output.

## Packing Information



- |   |   |
|---|---|
| ① ChipSHOUTER CW520 Main Unit                         | ⑧ SMB to SMA adapter                          |
| ② 19V / 3.4A Power Adapter                            | ⑨ SMB to BNC adapter                          |
| ③ Injection probe/tips (1mm, 4mm)                     | ⑩ SMB Cable                                   |
| ④ Isolated USB Adapter + RJ12 Cable + Micro USB Cable | ⑪ CW521 Ballistic Gel SRAM Target + USB Cable |
| ⑤ SMA Saver (Installed)                               | ⑫ CW322 Simple EMFI Target + CR2032 Battery   |
| ⑥ SMA Right angle adapter                             | ⑬ Cooling air adapter and 4mm wrench          |
| ⑦ Oscilloscope Probe Adapter (x2)                     |   |

1. The ChipSHOUTER CW520 main unit is the EMFI fault injection platform itself.
2. The 19V power supply provides DC power to the ChipSHOUTER.
3. The injection probe tips must be added onto the end of the ChipSHOUTER before using the device. Do not touch the probes during operation.
4. The Isolated USB adapter provides a computer interface to the ChipSHOUTER.
5. The SMA Saver is a sacrificial SMA male to female adapter. It is added onto the ChipSHOUTER to save wear and tear on the ChipSHOUTER SMA connector. The SMA Saver can easily be replaced in case it is damaged.
6. The SMA right angle adapter is used in combination with a horizontal mount XY table.
7. The oscilloscope probe adapter allows monitoring of the pulse inserted at the tip of the ChipSHOUTER itself.
8. The SMB to SMA adapter allows interfacing the external trigger input with the ChipWhisperer trigger outputs, or other equipment with logic-level SMA outputs.
9. The SMB to BNC adapter allows interfacing the external trigger input with regular lab equipment.
10. The SMB cable is used to connect the external trigger.
11. The Ballistic Gel SRAM target provides detailed information about the effectiveness of a fault injection pattern.
12. The Simple EMFI target allows quick validation that a fault injection probe is working.

13. The cooling air adapter allows you to insert dry high-pressure air into the ChipSHOUTER for cooling. The adapter may look different or be of different material than shown here.

We are continuously improving our products. Some of the accessories or the device may look different than the photos used for this manual, but this is part of our continuous refinement of the product. If you have questions about the parts received please contact us.

## Background and Quick Start Guide

Electromagnetic Fault Injection (EMFI) is a way of injecting transient faults into electronic systems without direct electrical contact. This is accomplished by generating a rapidly changing magnetic field directed at the Device under Test (DuT). Changing magnetic fields cause induced currents in the DuT, resulting in changing voltage levels on internal signals. These changing voltage levels can cause incorrect read (or write) operations, affecting results of latches, registers, and more. Corrupting memory, resetting lock bits, skipping instructions, and inserting faults into cryptographic operations are all applications of EMFI. This can be used for embedded security research, validating fault-tolerance of algorithms, and validating fault-tolerance of entire systems.

To use the ChipSHOUTER in its simplest configuration you need only three things: the ChipSHOUTER itself, the included 19V power adapter, and one of the included injection tips. Attach the tip to the high voltage output of the device, and the power adapter to the DC input. Holding the ARM button arms the device and pressing pulse generates a fault. Pressing ARM again will disarm the device. Application of the device and more detail on performing injections is included in the sections pertaining to the included targets.

# Device Architecture

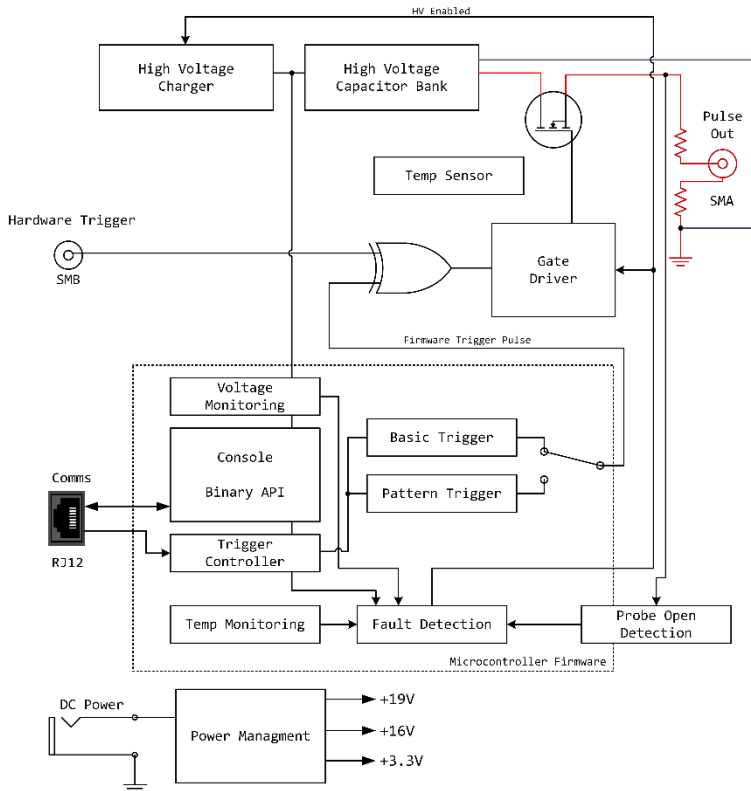


Figure 1: Overview of ChipSHOUTER device architecture.

Fundamentally, the ChipSHOUTER provides a high voltage charge that is discharged through an inductor (the “injection tip”). This injection tip generates a powerful magnetic field that can be used to induce faults in a target device.

To make using the device easier, the ChipSHOUTER includes a microcontroller that controls device operation. This includes detection of fault conditions such as over-temperature or invalid operational requests. Once a fault is

active, the device will prevent “arming” (turning on the high-voltage circuit) until the condition is cleared, and possibly acknowledged by the user.

This microcontroller can also generate pulse waveforms. These waveforms can either be basic pulses of a specified lengths, or more complicated patterns involving switching the high voltage on/off in 21nS time-steps.

To reduce the delay between a trigger event and the pulse injection, a special hardware trigger is also present that directly drives the high-voltage switch. This hardware trigger allows entirely arbitrary on/off pulses to be sent into the injection tip. This hardware trigger can be used with general-purpose test equipment or specific power analysis equipment such as the ChipWhisperer.

The output connector is a SMA connector jack. For safety reasons the device uses “high-side” switching, which means the high voltage is present ONLY during the pulse operation itself. The output includes two current-limiting resistors to prevent device destruction even when discharging into a direct short, and two catch diodes to absorb the reverse voltage spike generated by the collapsing magnetic field.

# Specifications

## General Specifications

Power supply (ChipSHOUTER DC Input)...	19V DC $\pm 10\%$ , 3.4A
Power consumption (standby).....	0.4W Typical
Power consumption (armed).....	5W Typical
Power consumption (charging/pulsing)..	5W to 50W Typical
Power supply (AC-DC adapter).....	100-240VAC, 50/60Hz, 1.5A
Size (ChipSHOUTER main unit).....	130 x 55 x 25 mm
Weight (ChipSHOUTER main unit).....	180 g
Altitude	
Operating.....	2000 m
Storage.....	12 000 m
Storage Temperature.....	-40°C to 60°C
Operating Temperature.....	5°C to 40°C
Relative Humidity.....	Noncondensing
	0 % to 80 % @ 5°C to 30°C
	Decreasing linearly to 50 % @ 40°C

### Safety

EN 61010-1:2010..... Pollution Degree 2

### Electromagnetic Compatibility

International..... EN 61326-1: Portable Electromagnetic

Environment; EN 61326-2-2 CISPR 11: Group 2, Class A

*Group 2: This equipment intentionally generates RF energy that is used in electromagnetic coupling, inductive coupling, and capacitive coupling for material analysis or inspection.*

*Class A: This equipment is suitable for use in all establishments other than domestic and those directly connected to the public low voltage power supply network that supplies buildings used for domestic purposes.*

*There may be potential difficulties in ensuring electromagnetic compatibility in other environments due to conducted and radiated disturbances.*

*Emissions that exceed the levels required by CISPR 11 can occur when the equipment is connected to a test object. The equipment may not meet the immunity re-*

quirements of this standard when test leads and/or test probes are connected.

USA (FCC)..... 47 CFR 15 subpart B. This product is  
considered and exempt device per clause  
15.103.

Operation is subject to the following two conditions: (1) this device may not cause harmful interference and (2) this device must accept any interference received, including interference that may cause undesired operations. You must discontinue use of this device if it causes interference to another user, and remedy the interference before continuing operation of this device.

## I/O Characteristics

Serial command interface..... 3.3V CMOS Serial, 115200 baud, 8N1

```
Protocol ..... (1) ASCII command prompt
                  (2) Binary
```

Serial connection..... RJ12 connector with GND, TX/RX,  
3.3V output, and switchable pulse/arm  
pin.

```
Hardware trigger connector type .....SMB connector, center-positive
```

```
Hardware trigger threshold ..... 2V
```

Hardware trigger absolute max ratings -0.5V to 6.5V

Hardware trigger impedance..... 50Ω / 1.8KΩ (Switchable)

Hardware trigger level.....Active-high / Active-Low (Switchable)

Injected waveform monitor..... BNC connector for mating with standard  
1M $\Omega$  || 10-25pF oscilloscope input. Ad-  
justable compensation trimmer for fine-  
tuning match.

Voltage monitor attenuation..... 20x attenuation

Voltage monitor output range.....  $\pm 25\text{V}$  into properly matched oscilloscope input

## High Voltage Characteristics

Characteristic	Min	Typ	Max	Units
Programmable voltage range	150		500	V
Charge rate	30		40	V/ms
Charge energy			625	mJ
Measured voltage accuracy via digital interface	$\pm(5\% + 10V)$			

## Pulse Source Characteristics

- Pulse generator source..... (1) Internal pulse generator, basic  
 (2) Internal pulse generator,  
       programmable pattern  
 (3) External hardware trigger

Characteristic	Min	Typ	Max	Units
Basic pulse generator				
Pulse width range	80		960	nS
Pulse width resolution		80		nS
Pulse width jitter		350		pS std-dev
Pulse dead-time (between repeats)	1		1000	mS
Pulse repetition count (per Trigger event)	1		10000	
Programmable pattern generator				
Pulse width resolution (time-steps)		20.83		nS
Time-steps per pulse	1		5000	Time-steps
Total pulse width	0.0208		100	uS
Pulse output state per time-steps		1/0		
Pulse width jitter <i>tested pulse width of 80nS</i>		350		pS std-dev
Hardware Input Trigger				
Delay <i>Tested high voltage 150V to 500V</i>		75		nS
Delay jitter <i>Tested high voltage 150V to 500V</i>		150		pS std-dev
Width jitter <i>Tested high voltage 150V to 300V</i>		800		pS std-dev
Width jitter <i>Tested high voltage 300V to 500V</i>		220		pS std-dev

## Inserted Pulse Characteristics

Characteristic	Min	Typ	Max	Units
Pulse width into 1mm injection tip	15		80	TYPICAL nS
Pulse width into 4mm injection tip	24		480	TYPICAL nS
Minimum consecutive pulse spacing <i>Tested with 4mm injection tip at   voltage setting of 500V</i>				
2 Pulses		100		ns
3 Pulses		175		ns

*For details of the pulse insertion see following text.*

While the pulse generator characteristics show that a wide variety of pulses can be applied to the injection tip, the actual resulting pulse characteristics will depend considerably on the tip properties itself. It is not possible to achieve every injection result on every tip.

Insertion of multiple pulses may require adjusting the driven pulses, which is achieved with the “monitor” port output. As an example, achieving (approximately) the same pulses multiple times is shown with the following pattern trigger waveform setting for 1, 2, and 3 pulses. Note the spacing between pulses varies as well:

```
s w 1110
s w 111011110
s w 111000011110000111110
```

The following figures (Figure 2 and Figure 3) show examples of possible range of pulses that can be achieved on one test of the provided 1mm and 4mm tips.

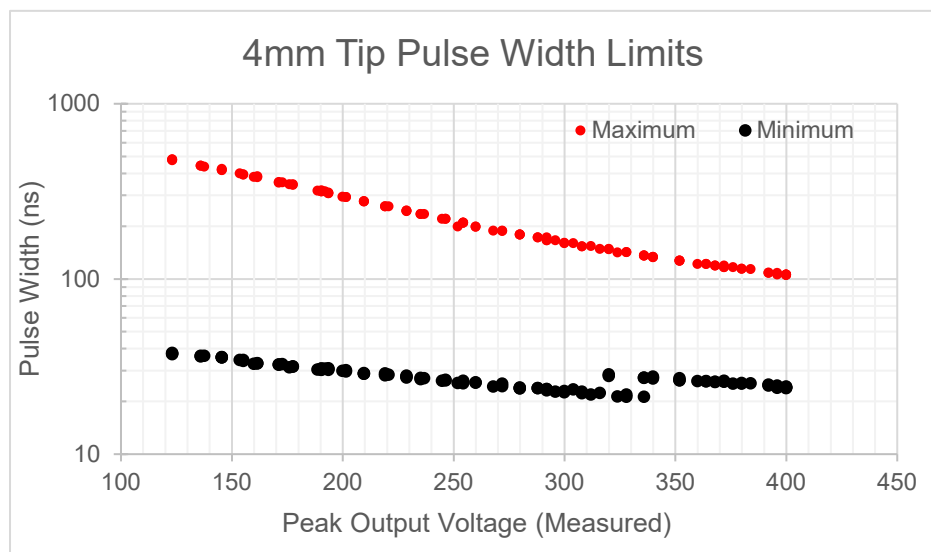


Figure 2: 4mm tip pulse width

These figures were generated by using the external hardware trigger to sweep a range of input pulse widths over a range of set capacitor bank voltages. They represent typical (not guaranteed) characteristics, taken at 25C.

The allowable range is between the minimum and maximum width values at a given voltage. Note the smaller (1mm) tip almost always generates a narrow pulse, regardless of input pulse duration. This is especially apparent at high charge voltage values.

The larger (4mm) tip allows a wider range of possible pulse widths, and more closely follows the commanded input width. It is extremely important to use the oscilloscope monitoring outputs to see the actual pulse injected into your target probe, or use an external H-Field probe to monitor it.

The actual number of consecutive pulses is limited almost entirely by probe characteristics, and not the ChipSHOUTER itself. This can be seen in that inserting more consecutive pulses often requires more delay between them.

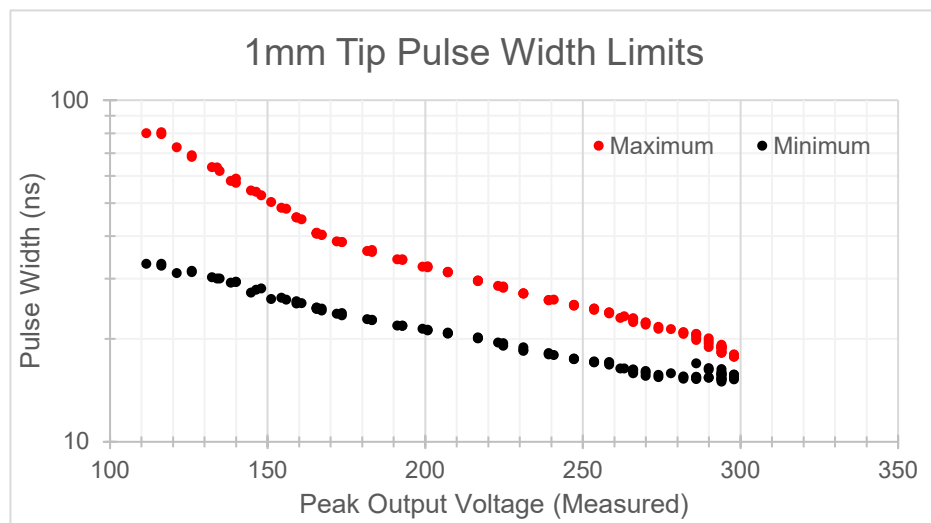


Figure 3: 1mm tip pulse width

## External Connections

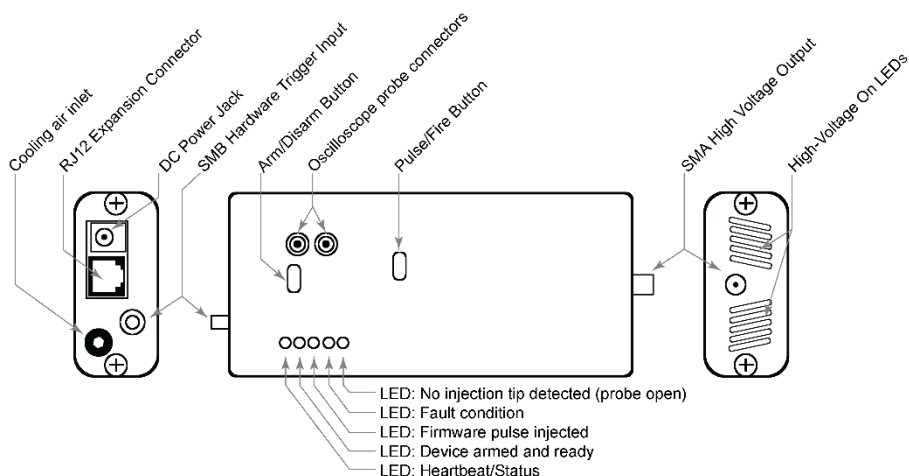


Figure 4: External connectors on the ChipSHOUTER.

### SMA High Voltage Output

The SMA high voltage output is where injection tips are attached. The outer shell does not directly connect to chassis ground, so you **MUST NOT** attach the outer shell via a metal clamp or similar to any electrical conductor during operation.

Note the SMA connector will wear over time, and a loosely attached injection tip can cause arcing which will permanently damage the connector, reducing performance. To avoid this, your ChipSHOUTER comes with a “SMA Saver” attached, which is a SMA male to female adapter. Do not remove the SMA saver under normal circumstances, and instead attach injection tips to the SMA saver output.

If the SMA saver becomes worn or damaged, remove the SMA saver and replace with a new one. These can be purchased from us, or you can use a high-quality SMA male to female adapter such as Amphenol 132171.

### Attaching/Removing SMA Connectors

To attach or remove a SMA connector (such as the probe tip), you should note that **ONLY** the outer connector nut is designed to rotate. The center pin of the SMA connector should not be rotated during the removal or attachment process, as rotating this pin can cause damage to both sides.

Instead, you should hold the body of the item being removed firmly, while spinning the connector nut (using a 8mm wrench if needed) to remove or attach. If you simply rotate the connector nut without holding the body stationary, it is easy to rotate the body of the SMA connector and thus also rotate the internal contact pin (which we want to avoid).

To achieve repeatable connections, a torque wrench is recommended. SMA connectors are typically tightened to 1 Nm / 8 lb-in.

### SMB Trigger Input

The SMB connector is a *hardware* trigger input. As explained in the device architecture, this trigger input is connected directly to the high-voltage switch without being routed through the controller.

This connection ensures the highest-speed and most direct control of the pulse shape is possible. The input is designed for 3.3V LVCMOS signal levels, but can accept up to a 6.5V input signal safely.

The SMB trigger input can be configured in one of three modes:

- Active-low pulse, high-impedance (approx. 2K $\Omega$ ).
- Active-high pulse, high-impedance (approx. 2K $\Omega$ ).
- Active-high pulse, 50 $\Omega$  impedance (DEFAULT).

A suitable pulse for this input can be generated by a laboratory pulse generator, a custom FPGA or other board, or the ChipWhisperer.

If interfacing with ChipWhisperer, the recommended method is to use the HS-OUT SMA connector on the CW506 advanced breakout board. This requires you to configure that the glitch out is routed to the HS-OUT pin, which means you cannot use the HS-OUT for clock generation. An easier method is to use the ChipWhisperer glitch crowbar output, described in the next section.

Note that internally the hardware trigger is also routed to the microcontroller. The microcontroller needs to know when a fault is being inserted, as this (a) resets the arm timeout count, and (b) tells the microcontroller to ignore invalid temperature readings that occurring during a discharge event due to noise on the temperature sensor. If your trigger source does not have a strong enough drive, it may be possible to trigger the actual fault injection without the microcontroller being aware.

This typically results in (1) the device automatically disarming during use, and (2) a “temperature sensor error” fault. Ensure you are driving to proper 3.3 LVCMOS levels, if you are using the 50 $\Omega$  termination mode you can disable this to increase drive levels as a test.

When using the hardware trigger, note there is currently *no indication that a trigger event is occurring*. That is,

there is no message printed to serial port as when you press the front-panel button or use the API. You must use the oscilloscope monitor probe with the hardware trigger to confirm the device is triggering.

### External Trigger & Self-Checks with `triggersafe`

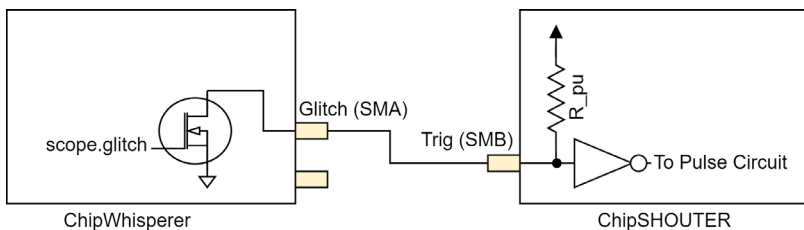
The ChipSHOUTER performs extensive self-checks during operation to confirm no hazardous conditions are present. Those self-checks will return invalid results during a pulse insertion event, and thus will be discarded. The ChipSHOUTER requires a successful self-check within a certain time period or a fault is raised, thus if you are using a rapid repeat time the ChipSHOUTER may never get to run a self-check and will raise a fault.

The ChipSHOUTER exposes a command called `“triggersafe”`, which tells the ChipSHOUTER that it is safe to run the self-checks (i.e., you are not about to insert a pulse). If using the external trigger input, you can use this `triggersafe` command to reduce the chance of arbitrary faults. Note that occasionally inserting a pulse during the safety checks can cause a fault condition too, so by taking control of the safety check timing you may have more reliable operation with the external trigger.

Once you run `“triggersafe”`, you must continue to call it frequently (programmable, but normally around every 5 - 30 seconds). Calling `“triggersafe”` effectively disables the automatic checks until the next disarm/arm cycle. See the documentation on serial commands for more details.

## External Trigger & ChipWhisperer Glitch Output

The ChipWhisperer “glitch” connector can be used to generate an active-low pulse by taking advantage of the pull-up inside the ChipSHOUTER. To use this, configure the ChipWhisperer “glitch” by enabling the low power (LP) glitch crowbar output. The ChipSHOUTER has an internal pull-up on the hardware trigger input when configured for active-low mode, allowing the LP-glitch crowbar output to serve as an open-drain output. A simplified schematic of this is shown below:



A SMA to SMB adapter is included in the ChipSHOUTER to make this easy to perform. This works with both the ChipWhisperer-Lite and ChipWhisperer-Pro. To use this setup requires first on the ChipWhisperer performing any glitch setup and trigger configuration, then performing:

```
--setup your scope.glitch settings--
scope.glitch.output = "enable_only"
scope.io.glitch_lp = True
```

Then you could run the following commands on the serial console to configure the external trigger as active-low and high-impedance:

```
set hwtrig_term 0
set hwtrig_mode 0
```

You can also accomplish the configuration using the Python API for the ChipSHOUTER instead of the serial console.

### RJ12 Expansion Connector

The ChipSHOUTER can be controlled using asynchronous serial through the RJ12 port on the device. DO NOT connect this cable to general use ports on other devices like ethernet or phone ports. Connection to a computer can be easily made by using the USB interface board and a micro-USB cable. The pinout found on that board can be used by more advanced users to interface with the ChipSHOUTER using other specialized equipment. The USB adapter board requires FTDI VCP drivers to be installed. The serial configuration of the ChipSHOUTER is 115200 baud 8N1.

The pinout of the RJ12 jack is shown in Figure 5. More information on the USB adapter board is provided on page 61.

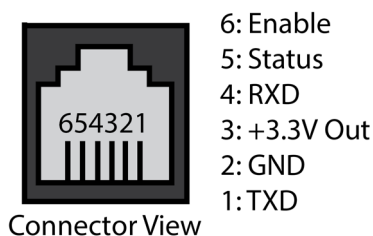


Figure 5: RJ12 Connector on ChipSHOUTER Panel.

### DC Power Jack

The ChipSHOUTER uses a 4.75mm x 1.7mm center-positive barrel connector (EIAJ-03), with a 19VDC  $\pm$  10% input voltage. During standby (not armed) the ChipSHOUTER draws approximately 20mA, during armed state it draws approximately 250mA, and during discharge draws between 0.3A-3.3A. Use only the pro-

vided DC power supply with the ChipSHOUTER, which has a rating of 19V/3.42A.

### Oscilloscope Probe Connectors

Both the voltage and current output of the ChipSHOUTER can be monitored via two probe connections on the top side of the device. Adapter cables are included for connecting to an oscilloscope, use only these matching NewAE cables for this purpose. The external portions of these two probes are identical and can be plugged in to either socket. These are described in more detail on page 41 in the section Oscilloscope Pulse Shape Monitoring.

## Pulse Generation

The ChipSHOUTER involves an advanced pulse trigger system. This can be used to build a pattern for injecting a fault into a target device, or working with existing laboratory equipment. This section describes some of the pulse generation architecture to help you understand the capabilities of the ChipSHOUTER.

### Generated Pulse vs. Inserted

One of the most critical points to understand is that the generated pulse will not be the same as the inserted pulse. This is for several reasons, primarily due to (1) saturation and fundamental physical limits of the injection tips, and (2) limitations of the ChipSHOUTER. The physical limitations of the injection tips are responsible for most limitations. Issues such as the core material saturation result in limits regarding how many pulses can be inserted in quick succession.

The ChipSHOUTER oscilloscope probe monitoring points can be used to monitor the actual inserted pulse. Typically you can use this to tune the generated pulse to more closely resemble the intended injection.

### Active-High vs. Active-Low Inputs

The ChipSHOUTER can internally switch between active-high and active-low trigger operation. This is done because the external input can be switched from active-high to active-low, which internally inverts the entire trigger system logic.

The basic pulse generator takes care of this for you, but the programmable trigger does not. When using the programmable trigger be sure to switch the external input to “active-high” mode.

Note you may see small differences between active-high and active-low mode. The ChipSHOUTER remains an electronic device and is sensitive to the very high-power fields being generated. Active-high and active-low modes show slightly different susceptibility to various noise sources.

### Basic Pulse Generator

The basic pulse generator can generate single or multiple pulses, with relatively large spacing between them.

### Programmable Pulse Generator

The programmable pulse generator can be used to generate complex patterns, including multiple pulses and delays. It also provides a much shorter time resolution than the basic pulse generator.

The pattern is recorded as a binary pattern, where each digit represents a time-step. For example to generate two 60nS pulses with a 80nS delay (approximately), you would write the pattern 011100001110 into the pulse generator memory.

You must end the pattern with an inactive-state. If the device is in active-high mode, this means you must end the pattern with a ‘0’. Failure to do this will result in a trigger error or other problems.

Note you will often find that the second (and later) pulses require a longer trigger pattern to generate the in-

tended injected pulse. Thus in reality you may find 0111000011110 is needed (an extra '1' on the second pulse).

**CAUTION:** When writing a pattern, ensure you end with an inactive state. It is suggested to also start the pattern with an inactive state for symmetry.

The programmable trigger still uses the repeat and dead-time parameters. You may wish to set repeat to '1' to avoid repeating the pattern unexpectedly. Some hints about using the pattern trigger:

1. You will need to experiment with the pattern to get the desired output. This pattern will change drastically with attached probe tips, and as you extend the pattern.
2. You may get different effects using 'active-high' and 'active-low' triggering. This is because certain internal cross-talk events may interfere with complicated patterns.

## Simple EMFI Target (CW322)

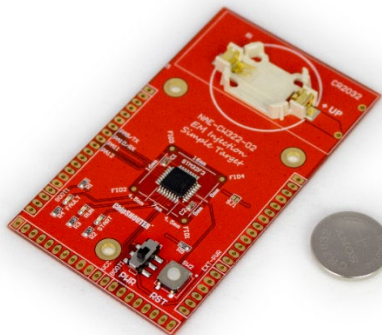


Figure 6: CW322 Simple Target

The CW322 (Simple Target) is an easy to use target with the ChipSHOUTER platform, and a good first introduction to EMFI. The board features an STM32F303K8T6 that is pre-programmed with very simple firmware, some of which is shown in Listing 1. The microcontroller simply uses two loops to multiply 300 by 300 and check the result. The board features 3 LEDs that indicate the state of the device.

The START LED shows when the device begins code execution and will light whenever the device is reset. The RUN LED blinks as the code is properly executed, if this light stops blinking the device has frozen. The FAULT LED blinks whenever the multiplication returns an incorrect value. This normally never happens, but this abnormal behaviour can be reliably induced by the ChipSHOUTER.

```
#define RUN_CNT 2000
#define OUTER_LOOP_CNT 300
#define INNER_LOOP_CNT 300

void glitch_loop(void)
{
    volatile uint32_t i, j;
    volatile uint32_t cnt;
    uint32_t blink_status = 1;

    uint32_t run_cnt = 0;
    uint32_t glitch_cnt = 0;
    for(run_cnt = 0; run_cnt < RUN_CNT; run_cnt++){

        //run led on
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_4, blink_status);
        blink_status ^= 1;
        cnt = 0;
        for(i = 0; i < OUTER_LOOP_CNT; i++) {
            for(j=0; j < INNER_LOOP_CNT; j++){
                cnt++;
            }
        }

        //look for glitch
        if (i != OUTER_LOOP_CNT || j != INNER_LOOP_CNT ||
            cnt != (OUTER_LOOP_CNT * INNER_LOOP_CNT) ) {
            //if glitched, reset the run count and blink the fault LED
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, SET);
            delay100ms(3);
            HAL_GPIO_WritePin(GPIOB, GPIO_PIN_3, RESET);
            run_cnt = 0;
        }
    }
}
```

Listing 1: EMFI Simple code example

For this experiment, you will need the ChipSHOUTER, the included 19v power adapter, one of the included 4mm injection tips, and the simple target board.

1. To start the simple target board, slide the PWR switch up. The START light should briefly flash, followed by a steady blink from the RUN light.
2. Place the simple target on a flat surface and plug the ChipSHOUTER into the included 19v power adapter.

The STATUS, FAULT, and OPEN LEDs on the ChipSHOUTER should light up.

3. Screw one of the 4mm injection tips onto the High voltage output connector of the ChipSHOUTER, this should cause the FAULT and OPEN lights to go off.
4. Hold the ARM button until the arming chime sounds and release the button. The system is now armed and ready on inject a glitch.
5. Hold the probe very close to the chip on the Simple Target and press the PULSE button on the ChipSHOUTER to inject a field pulse.
6. Move the probe across the chip while holding the PULSE button and observe the effect on the LEDs.

In some locations the chip will reset or stop working. In others the chip will blink the fault LED, indicating that the multiplication operation has been corrupted. This shows a successful fault injection where the intended output of the device has been changed without directly interfacing with the device in any way. This is the heart of EMFI.

**CAUTION:** Observe proper ESD handling requirements with the board.

**CAUTION:** Fault injection can permanently destroy the injection target. Always start at a further distance and move towards the target until you see fault interactions.

## Ballistic Gel EMFI Target (CW522)

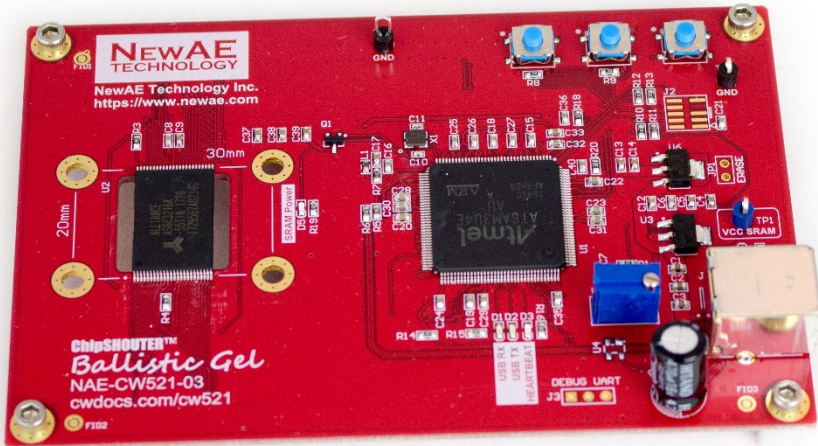


Figure 7: CW522 Ballistic Gel

The CW522 (Ballistic Gel Target) is an SRAM board with a microcontroller for control and connectivity. The target is called the Ballistic Gel because it records an imprint of the magnetic field injected into it, like a ballistic gel block leaves an imprint of a projectile. This acts as an example of memory corruption, and this process demonstrates some of the ChipSHOUTER pulse settings.

1. Attach one of the 1mm injection tips to the high voltage output of the ChipSHOUTER.
2. Plug the Ballistic Gel target into your computer using a USB cable and place the target on a flat surface.

The required drivers can be downloaded from ChipSHOUTER.com.

3. Connect the ChipSHOUTER to your computer by first using an RJ12 cable to connect the ChipSHOUTER to the USB interface board, and then connecting the interface board to your computer using a micro-USB cable.
4. If drivers for the interface are not installed, install the universal FTDI VCP driver from the FTDI website. The interface should connect as a virtual com port, which can be confirmed using your computers device manager.
5. Start a terminal session using your favorite terminal program. PuTTY works well for this. The serial configuration of the ChipSHOUTER is 115200 baud 8N1
6. Connect the 19V power adapter to the ChipSHOUTER. If your terminal was configured correctly a welcome message should be displayed as the device boots.
7. Test connectivity with the shouter by sending a question mark (?) to the device. This should return the serial command list.
8. In a separate command line or python interpreter, run `ballisticgel.py` to connect to the ballistic gel target.
9. Arm the ChipSHOUTER by sending the command `arm` over the serial link. You should hear the arm chime.
10. Hold the injection tip over the center of the SRAM chip on the ballistic gel target. Press the pulse button on the ChipSHOUTER or send the command `pulse` over the serial link.

11. Disarm the ChipSHOUTER by pressing the arm button or by sending the `disarm` command over the serial link.
12. Press enter in the Ballistic Gel script terminal to read the injected fault pattern.
13. Change the pulse settings on the ChipSHOUTER using serial commands. `set voltage 300` will set the capacitor bank voltage to 300V. `set pulse width 160` will set the output pulse width to 160ns. `set pulse repeat 10` will send 10 pulses on a single pulse command. `set pulse deadtime 10` will set the delay between pulses to 10ms. Use these settings for the next test.
14. Repeat steps 9-12 with the new pulse settings. You can adjust these settings more to see how each one affects the injected corruption. More data on these effects can be found on the ChipWhisperer wiki.

You should now have a basic grasp of the ChipSHOUTER configuration options. There are many more advanced options documented on the ChipWhisperer wiki and in ChipSHOUTER application notes.

**CAUTION:** Observe proper ESD handling requirements with the board.

**CAUTION:** Fault injection can permanently destroy the injection target. Always start at a further distance and move towards the target until you see fault interactions.

## Injection Tip Usage

There are four injection tips included with the ChipSHOUTER. Two 4mm tips and two 1mm tips, each with both “clockwise” (CW) and “counter-clockwise” (CCW) versions. The size of the tips refers to the diameter of the ferrite core inside the coil, and the CW/CCW refers to the wrap direction of the wire around the ferrite core. The wrapping direction can be used to specify a positive or negative voltage induced into a specific target (you may need to experiment to determine which wrap direction corresponds to positive/negative on your specific example).

Looking into the end of the tip, you will notice the top conductor always goes to the outside of the coil. This lead connects to the SMA shield (which is the lower or negative) voltage. This is done to keep the highest voltage on the inside of the probe. Figure 8 shows the detail of the probe construction underneath the opaque label.



**Top (Outer) Conductor connects  
to low-side (-) in both CW/CCW**

Figure 8: Detail of the 4MM/CCW tip construction.

An example of the four included tips is shown as a photo taken from the probe ends in Figure 9. Note the CW/CCW wrap direction is based on looking into this probe tip. Knowing the outer conductor is the lower voltage, and using the right-hand rule, you can infer that the CCW tip results in a magnetic field coming *out* of the tip, and CW results in a magnetic field coming *into* the tip.



Figure 9: Detail of included probes.

The larger 4mm tips are more powerful and better for manual use and insensitive targets. They generate a wide field that is good for discovering new vulnerabilities and they have the best chance to disrupt a circuit in some way. The smaller 1mm tips are better for precision work, as they generate a narrower field and can be positioned more precisely. These tips are good for characterising known faults where location is critical, and for dealing with sensitive targets.

The use of smaller diameter tips may require you to partially decapsulate the target. In many cases this can be accomplished with mechanical sanding/lapping/milling and avoids the use of chemical etching. This is required as the inserted field does not penetrate as deeply with smaller diameter tips.

The tip size will affect your actual pulse inserted, it is always suggested to use the pulse shape monitoring output to better understand the injected pulse. You can see additional documentation and examples of the pulse shapes from the app-notes on our website.

### Avoiding Spark Discharge

The ChipSHOUTER is designed as primarily a magnetic field generation device, and is not designed to generate spark discharge events. A spark discharge event causes a very high  $dV/dT$ , which can permanently destroy the output stage of the ChipSHOUTER.

When attaching tips, ensure they are tight. A loose tip may spark during discharge, which will (a) cause substantial pitting and mechanically damage the contacts, and (b) can electrically damage the output stage.

Likewise, ensure there is never damage to insulation of discharge tips. This is especially important if using 3<sup>rd</sup> party tips, as it is possible for closely wound coils to discharge between windings. This can also cause damage to the ChipSHOUTER.

### Polymide Tape Protective Layer

Later versions of the probes have a thin (1.5 mil) layer of protective polymide tape integrated into the probe construction. This appears as a translucent yellow or orange layer around the probe tips. This tape is designed to provide additional resistance against abrasion if you wear the probe tips. Do not remove this layer.

## Oscilloscope Pulse Shape Monitoring

To monitor the injected pulse, two oscilloscope adapter probes are included. These adapters are based on standard oscilloscope probes, but with the business end of the probe built into the ChipSHOUTER itself.

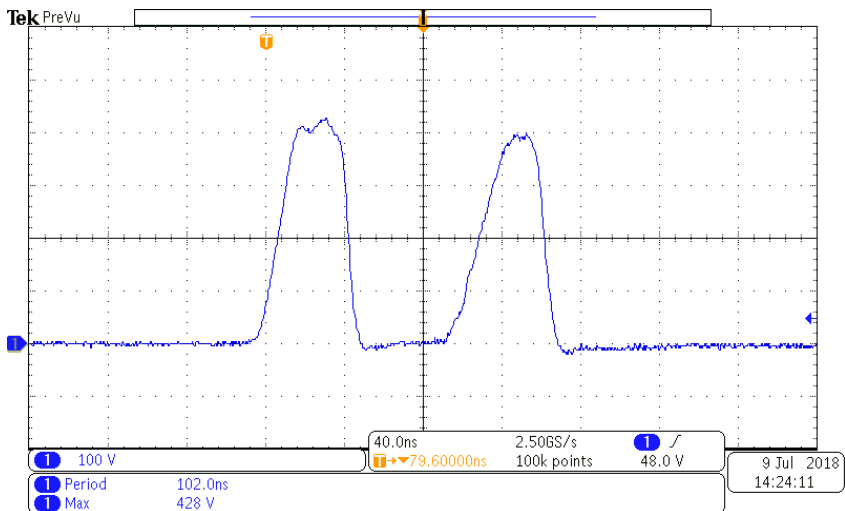


Figure 10: Inserted pulse viewed on oscilloscope screen.

This allows you to monitor the high-voltage output without risk of exposing yourself to high voltages. These probes are designed only for usage with a standard 1M $\Omega$ ||10-25pF oscilloscope input.

**CAUTION:** Usage with any other input type (including higher or lower impedance) can result in damage to your device and exposure to high voltages. Be sure to carefully review the voltage limits

discussed here and ensure any connected oscilloscopes will meet the voltage limits.

To use the probe, simply connect the MCX connector into the ChipSHOUTER front connection. There are two front connections: a “voltage” monitor, and a “current” monitor.

The voltage monitor provides a 20:1 attenuation, so using this means setting your oscilloscope up with a 20:1 attenuation rating. Note that at a peak 500V pulse voltage, the 20:1 attenuation means your oscilloscope front-end will see 25V at the 1M $\Omega$  input.

**CAUTION:** Confirm your oscilloscope 1M $\Omega$  maximum voltage rating is at least 25V. Due to ringing at the tip voltages may exceed 500V, so a  $\pm 30$ V rating is recommended. NewAE Technology Inc. cannot accept any liability for damage to your oscilloscope or other connected equipment, and you use this monitor at your own risk.

The current monitor provides a 10:1 attenuation, but this is not a calibrated current monitoring output. Instead it is used to provide general information on pulse shape.

The current is monitored across a pulse-tolerant thick-film resistor. This resistor is used as part of the current limiting and back e.m.f. absorption circuit.

## Adjusting for Oscilloscope Setting

You will need to adjust the probe for your specific oscilloscope. This can be done by adjusting the small compensation trimmer that is located on the BNC body (see Figure 11).



Figure 11: Tuning oscilloscope probe.

The most effective calibration would be performed by attaching a resistive load to the SMA connector output, and observing the voltage with an externally calibrated oscilloscope probe on this resistor. Doing so requires exposing high voltages and is not covered by this manual.

Instead we present a simple calibration that should work for *most* users. For calibration the 4mm injection tip should be connected to the ChipSHOUTER and the pulse output should be set to simple mode with a pulse width of 80 (ns), a dead-time of 10 (ms) and a voltage of 400v. This process is made easier by setting pulse repeats to a high number (100 works

well) and using the average mode on your oscilloscope with a low number of samples (8 works well). While pulsing the ChipSHOUTER adjust the small trimmer in the probe body until the maximum pulse amplitude reads 350 volts. Your probe is now calibrated. The measured voltage is not 400V as the oscilloscope measures the voltage at the probe output, and there is some drop across the internal protection resistors.

Due to oscilloscope variation, you may not achieve the 350V measurement. Instead we recommend selecting your “calibration” voltage and recording it here. This calibration will still help you recreate results on your oscilloscope if you need to switch ChipSHOUTERs or probes.

Oscilloscope Identifier	Voltage Measured	Date

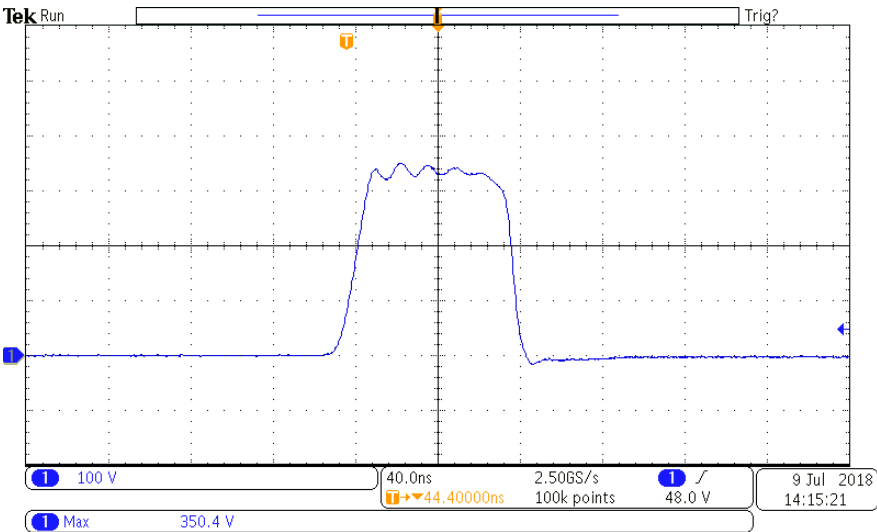


Figure 12: Example calibration waveform.

## Forced-Air Cooling

During regular operation, the ChipSHOUTER will heat up if using continuous discharge. When internal temperatures reach a set point, the device will go into a thermal shut-down and wait for natural cooling to take the device into safe operating range.

If using ChipSHOUTER in high duty cycle operation, or at elevated local temperature, you may wish to use forced-air cooling to improve performance.



Use **ONLY** dry filtered air or an inert gas such as nitrogen. Compressed air normally includes both oil and water vapour. If using compressed air an inline filter must be used to remove condensate, failure to do so may cause shock danger due to condensation inside unit, or cause permanent damage of the ChipSHOUTER. Never use a flammable or explosive gas.

ChipSHOUTER provides a M8x1.25 threaded hole, into which the tube adapter may be inserted. **Dry** room-temperature forced air may be inserted into the ChipSHOUTER from this port.

To use this port, you will need to use a 4 mm hex wrench (provided) to remove the blanking port. Once you have removed the blanking port, you can insert the 6mm tube adapter into this port and attach your air source. ChipSHOUTER does not require high pressure air as there is little back-pressure.



Figure 13: Removing blanking plug.

The blanking plug is a M8x1.25 x 16mm set screw, and if the blanking plug is lost a M8x1.25 bolt can be used until the proper replacement is procured. The air inlet must never be left open.



Figure 14: Adding air inlet adapter.

The air inlet adapter is designed to accept 6mm plastic hosing. This hosing is frequently used by air compressors and aquarium pumps, and high flexible silicon tubing can be used if mounting the ChipSHOUTER to an XY(Z) table.



**When a hose is not connected, connect either a blanking port to the hose connection OR remove the hose adapter and replace with the blanking port screw. Failure to do so leaves high voltage exposed through the cooling hole, and you must never operate the device without the blanking plug or hose present.**

While dried compressed air can be used, a normal air compressor is not suitable for use in an office or lab environment. Instead a small air pump can be found that operates from your local power supply.

Examples of such air pumps include aquarium air pumps (look for very high-flow) used for aerators. Various linear piston air pumps (sometimes called “electromagnetic air pumps” due to use of electromagnets to oscillator piston) are available which are reasonably quiet, small, and with sufficient flow to cool the ChipSHOUTER during long operations.

Even a small airflow (such as from an aquarium pump) will substantially improve the cooling capability of the ChipSHOUTER.

If using dried compressed air, ensure you are using a pressure regulator to limit to a maximum of 2 bar (30 PSI) and 10 CFM. You should start at a lower value by slowly increasing the pressure from the regulator output, and checking the device temperature during operation. Using the lowest pressure/flow possible will reduce the chance of condensation forming internally.

## Fault Modes



ChipSHOUTER faults indicate unexpected operating conditions. If faults occur, carefully read and understand this section of the user manual to take the proper corrective action. If it is not clear what fault has occurred, please discontinue use of the device and contact us immediately.

ChipSHOUTER has several possible faults. The specific faults can be determined via the serial interface (described in another section), however the most common faults are also described with special blink patterns and indicators on the LEDs.

<i><b>Fault</b></i>	<i><b>Fault Description</b></i>
Probe	Probe disconnected or damaged (open).
Overtemp	Internal temperatures too high.
Panel Open	Front panel removed or not secure.
High Voltage Error	Measured high-voltage is higher/lower than expected.
RAM CRC	RAM CRC failed.
EEPROM CRC	EEPROM CRC failed.
GPIO	GPIO state does not match expected.
Charge Error	Charge circuit error, likely DC input voltage out-of-spec or brown-out.
Trigger	Trigger too long or invalid.
Hardware Exc	Internal hardware failure detected.
Trigger Glitch	Device triggered while disarmed.
Over Voltage	Charge voltage higher than expected.

Sensor Fault	Temperature sensors not communicating, possibly trigger occurring too frequently without arm ready check.
--------------	---

Any active fault will prevent the ChipSHOUTER from arming (prevents the high voltage charge from becoming active), and the fault condition must be fixed before you attempt to arm the device.

When a fault is active, the “FAULT” LED will be on. Do not attempt to arm the device when a fault is active. Some faults will also trigger a fault tone to make the error condition clear.

If a device is already armed when certain critical faults occur, the fault will latch and the device will disarm. In this case it not enough to simply fix the condition. In addition you must clear the latched fault after fixing the error condition. This latch prevents the ChipSHOUTER from automatically re-arming when an error occurs.

The latched fault can be cleared in three ways:

1. Hold the “ARM” button down for 8 seconds, the ChipSHOUTER will either have 3 short beeps (*fault cleared OK*) or one long tone (*fault could not clear as condition has not been fixed*).
2. Using the serial port, the command `set fault none`.
3. Using the serial port, the command `A` (capital A) will automatically clear faults & arm the ChipSHOUTER.

## Probe Disconnected Fault

The ChipSHOUTER looks for a low-impedance connection on the SMA connector. This connection is used to detect that a fault injection probe is attached.

If the ChipSHOUTER is armed when a probe is removed, this immediately causes a latched fault. As the probe **SHOULD NEVER** be removed from the ChipSHOUTER when armed, this is a serious fault condition. When switching probe tips, note it is much quicker to disarm the ChipSHOUTER, switch tops, and re-arm it. The latched fault condition is by design slow to clear, as during probe changes you should always disarm the ChipSHOUTER first.

If using the external hardware trigger input, you may get this fault if inserting large batches of faults with minimal delay between them (e.g., using a simple pulse-wave from a signal generator). You should instead gate the output to only enable faults for short bursts, and prove the ChipSHOUTER time to perform safety checks in-between bursts.

## Over-Temperature Fault

The ChipSHOUTER contains three temperature sensors. These sensors are on the MOSFET (electronic switch), the e.m.f. catch diodes, and the transformer used to generate the high voltage.

If any of these devices are over-temperature, the ChipSHOUTER will shut down. This fault condition automatically clears once the device cools down.

The temperature sensors cannot be read during the discharge event. If using the external hardware trigger in quick succession, you may also get an error indicating a

temperature sensor fault. This occurs when the ChipSHOUTER is unable to check the device temperatures for a predetermined time.

If using the external trigger, it is recommended to also send the `triggersafe` command over the serial interface during times the trigger is known to be inactive. This command tells ChipSHOUTER that it can perform the required self-checks (including temperature checks), and will not be interrupted by the discharge event.

See the API documentation (online) for more details of this, or the serial interface documentation on page 54.

### Triggered when Disarmed

The external trigger input should not be triggered when the device is disarmed. If this occurs, a fault tone sounds in addition to the fault LED blinking. This fault mode often happens if ChipSHOUTER disarms due to a fault while your external circuitry is still attempting to trigger the ChipSHOUTER.

It is expected the external user is gating the trigger input, as otherwise triggers could occur during the arming process (resulting in malformed pulses).

The error tone will sound (without the fault LED blinking) if you attempt to use the PULSE button or pulse command over the serial interface while disarmed.

### Trigger Length Invalid

The external hardware trigger should be used only to insert short pulses, as the internal capacitor bank does not have sufficient energy storage for long pulses.

This error typically means the external interface has the wrong polarity setting. The external interface can be set for active-low or active-high operation to interface with a wide variety of standard lab equipment.

If the ChipSHOUTER is set for active-low operation, this error could occur when the attached equipment is turned off or disconnected while the ChipSHOUTER is still armed. Instead you must first disarm the ChipSHOUTER before turning off the trigger generation device.

### Internal Faults

The device has a variety of internal faults. If these faults become persistent it indicates a likely hardware failure that requires repair of the ChipSHOUTER. Internal faults include:

- RAM CRC error, FLASH CRC error, or firmware signature verification error.
- Measured capacitor bank voltage differs from set voltage.
- Permanent failure of ability to measure temperature (sensor failure).
- Input power supply (19V DC) is out of-spec, either too high or too low. This most often occurs if power supply browns-out during operation.
- High-voltage charge circuit error (over-voltage, over-temp, or input voltage out-of-spec).
- Enclosure has been opened (interlock switch activated).

## Serial Interface

The ChipSHOUTER has a simple 3.3V TTL serial interface, which you can connect to at 115200 baud, 8N1. The serial interface presents a console that includes the current state of the device. This is useful to watch for the device entering a fault state indicating device errors are occurring. The console format is shown below:

### # armed : get voltage

Note the ‘armed’ indicates a state, and ‘get voltage’ is a command to the device. The following screenshot shows a typical interaction with the ChipSHOUTER console:

```
# disarmed:a
# arming:# arm

# armed:set voltage 400          400v .....[capacitor bank voltage]
# voltage                      394v(measured) ...[capacitor bank voltage]
# voltage

# armed:s p w 160              160ns .....[pulse width (nS)]
# pulse width                  160ns(measured) ..[pulse width (nS)]
# pulse width

# armed:p

# com_triggered:# pulse          160ns(measured) ..[Pulse (follows either basic or pattern trigger)]
# pulse                        160ns .....[Pulse (follows either basic or pattern trigger)]
# pulse                        5msec (deadtime) ..[Pulse (follows either basic or pattern trigger)]
# pulse                        1 Repeat .....[Pulse (follows either basic or pattern trigger)]
Pulse [Comp]

# armed:d

# disarming:# disarm

# disarmed:[]
```

**NOTE:** Previous firmware (before 1.9.0) only work if sent an entire serial line at once after you hit enter, and not send character-by-character as you type. Terminal emulators such as Termite (by CompuPhase) will send an entire terminal line at once, but Putty typically sends character-by-character as you type. Contact NewAE for firmware updates, as each firmware is uniquely coded to the unit.

## Command List

The commands available are listed below. A similar list can be generated at any time by sending the word *help* to the ChipSHOUTER interface.

***help***

***?***

*Prints the help menu.*

***get id***

***gi***

*Print board ID (required for firmware updates).*

***get state***

***gs***

*Print arm of device (arm/disarmed/fault).*

***[get / set] voltage [150..500]***

***[g / s] v***

*Print current or set value for capacitor charge voltage. If the device is in the armed state, the (actual) measured voltage will also be reported. When device is disarmed the high-voltage is not turned on, so reported measure voltages are invalid.*

Example:

```
# disarmed: set voltage 150
# voltage    500v .....[capacitor bank voltage]
# voltage    21v(measured)[capacitor bank voltage]
# disarmed: s v 500
```

***[get / set] pulse width [80..1000]***

***[g / s] pw***

*Print current or set value for pulse width in nS. Using this method has a coarse pulse width of 80nS, so the value will be reported as being mapped to the nearest possible value in the 'meas-*

ured' result. If better resolution is needed see the programmable trigger option.

Example:

```
# disarmed: set pulse width 120
# pulse width 120ns .....[pulse width (nS)]
# pulse width 80ns(measured) ..[pulse width (nS)]
# disarmed: s p w 200
# pulse width 200ns .....[pulse width (nS)]
# pulse width 160ns(measured) .[pulse width (nS)]
# armed: g p w
# pulse width 200ns .....[pulse width (nS)]
# pulse width 160ns(measured) .[pulse width (nS)]
```

***[get / set] pulse repeat [1..10000] [g / s] p r***

Print or set value for number of pulses per trigger, the trigger being the pulse command, the front-panel button, or the RJ12 firmware pulse pin when enabled.

Example:

```
#: set pulse repeat 1
#: s p r 5
```

***[get / set] pulse deadtime [1..1000] [g / s] p d***

Print or set value for time between pulses in mS, the total pulse waveform will be repeat \* deadtime Long.

***[get / set] arm\_timeout [0...60] [g / s] a***

Print or set value for the automatic disarm timer in minutes. The disarm time automatically happens when no pulse has occurred in the arm\_timeout minutes, and is used to reduce temperature in the ChipSHOUTER along with improving safety.

***[get / set] hwtrig\_term [1/0]                      [g / s] hwt***

Configure hardware trigger (SMB connector) as high impedance [0] or 50Ω [1]. The 50Ω impedance option puts a 50Ω resistor to ground. If you are not using the hardware trigger it is suggested to set this ON, as it will reduce potential noise on the hardware trigger causing glitches.

***[get / set] hwtrig\_mode [1/0]                      [g / s] hwm***

Configure hardware trigger (SMB connector) as active high [1] or active low [0]. When configured as active low ensure the pin is externally driven high during operation to prevent false triggers.

This command switches the entire internal trigger logic. When switching `hwtrig_mode` and using the pattern trigger, you will need to invert the pattern trigger logic.

***[get / set] emode [1/0]                              [g / s] e***

Use pin 2 on RJ12 connector as either arm [0] or firmware trigger [1]. Note this pin is NOT the hardware trigger input.

***[get / set] mute [1/0]                              [g / s] m***

Mute the internal buzzer, good for automated testing and avoiding beeping driving you crazy.

***[get / set] bootbits                                  [g / s] bb***

Configure bootloader mode, only for firmware upgrades.

## *get fault*

*gf*

*Print the current state of all faults.*

## *get fault\_active*

*gfa*

*Print any active faults, for example the current state of the probe open detection.*

## *get fault\_latch*

*gfl*

*Print any latched faults, which may not be currently active but occurred once and must be cleared manually.*

## *get fault [latched / current] <type> gf[l/c] <t>*

*Get the state of a specific fault, current or latched. <type> is the fault type, and <t> is the associated shorthand. Table of <type> options below.*

Fault	Shorthand
probe	p
overtemp	ot
open	o
highv	hv
ramcrc	rc
eecrc	e
gpio	g
charge	cf
trigger	t
hw	h
trig_g	tg
overvoltage	ov
temp_sensor	ts

*Details on fault meanings and troubleshooting can be found in the faults section of this manual.*

***get temperature <sensor>***

***g temp <s>***

Print temperature reading from one of the sensors. Sensors are mosfet, xformer, and diode with associated shorthand versions m, x, and d.

***get triggersafe***

***g ts***

Print confirmation that device is ready to be triggered. When triggering externally using the hardware trigger input at fast repeat counts, it is recommended to run this command in-between trigger attempts. Running the command allows the ChipSHOUTER to perform needed safety self-checks that cannot be performed during the trigger event.

If the needed safety checks cannot be performed for a certain length of time, the device will enter fault mode.

***[get / set] absent\_temp [1..60]***

***[g / s] at***

Configure maximum time the temperature sensors can be skipped for. The temperature sensors cannot be read during pulse events, and the ChipSHOUTER keeps a timer of how old the last temperature reading is.

The timer is reset during routine self-checks (if triggers are not coming in quickly), or in response to the triggersafe command.

***[get / set] pat\_enable [1/0]***

***[g / s] pat***

Configure whether trigger caused by pulse command, front-panel button, or firmware trigger input is simple [0] or pattern [1].

***[get / set] pat\_wave [011100...0] [g / s] w***

Configure pulse pattern, takes binary string as input. There is a maximum length of 67 characters due to internal buffers, you can extend the wave further using the `pat_append` command. If using long pattern triggers the API allows easier down-loading of complex waveforms.

Note the pattern trigger “active” value depends on the setting of `hwtrig_mode`. If the external hardware trigger is set to active-low, the pattern trigger will follow this (a ‘0’ causes a pulse).

The pattern trigger **MUST END WITH AN INACTIVE VALUE** to prevent a trigger error, for example ending with a ‘0’ when the ChipSHOUTER is in active-high trigger mode (the default).

Examples:

```
#: set pat_wave 0111000
#: set pat_wave 0111100000000000011111000
```

***set pat\_append [111000...0] s wa***

Adds input string of binary values to trigger pattern. Useful to extend waveform past allowed length that can be sent in one message.

Examples:

```
#: set pat_wave 0111
#: set pat_append 00000011100
# armed: g w
# pat_wave 011100000011100
```

***set fault none***

***sfn***

Clear latched faults, if an active fault is present the fault will still prevent arming.

***arm***

***a***

Arms device (charges high voltage capacitor bank). If no trigger occurs the device will automatically disarm after `arm_timeout` seconds.

If arming fails, the device may have an active fault. Check active and latched faults with the `get fault` command.

***Arm***

***A***

Clears latched faults and arms device, equivalent to running `set fault none` followed by `arm`. This command is useful when using the external trigger, as you may need to quickly clear a latched fault and arm the device.

***disarm***

***d***

Disarms device (turns off high voltage and discharges capacitor bank internally).

***safe***

***S***

See `disarm`.

***pulse***

***p***

Triggers simple or pattern pulse according to settings.

### ***Reset\_config***

### ***Reset***

*Resets configuration to product default, will cause EEPROM CRC error on next boot.*

### ***reset***

*Reboots the board, maintains most settings.*

### ***write\_nvram***

### ***eeeprom***

Added in Firmware 1.9.0

*Saves current settings to EEPROM (non-volatile memory). Settings include the voltage, pulse, external trigger configuration, etc. You must call this command if you want settings to persist between power-cycles (earlier firmware versions always wrote EEPROM). Do not call this excessively (i.e., in a loop with every change) as EEPROM has a theoretical limit on lifetime of 70K changes.*

## USB Interface

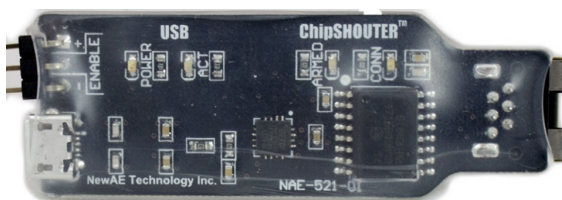


Figure 15: USB Interface for ChipSHOUTER

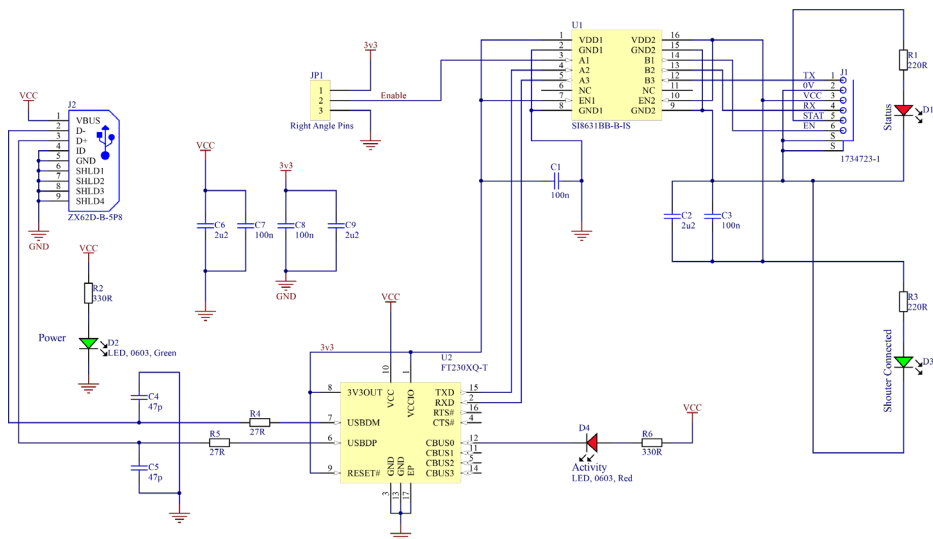
The provided USB to serial interface provides a simple method of using the ChipSHOUTER, the USB interface is shown in Figure 15. This USB interface features:

1. Isolation provides protection both from ground loops and potential voltage spikes due to ChipSHOUTER malfunction.
2. LED shows when ChipSHOUTER is connected and powered.
3. LED shows when the ChipSHOUTER is armed.
4. LED shows when the USB cable is present and power is being supplied to the USB interface from the computer.
5. LED shows when data is being transmitted (TX/RX).
6. Enable/Pulse pin allows you to arm/disarm the ChipSHOUTER via GPIO, or send the 'pulse' command. This requires software setup on the ChipSHOUTER to configure this pin.

The 3-pin header has the 'ENABLE' pin as the center pin. You can mount a jumper to arm the ChipSHOUTER without the serial command interface, where you can pull the pin high (+ mark) or low (- mark). **NOTE:** You must still have the micro-USB cable connected to provide the power for the isolator to translate this signal, even if only using the jumper.

The USB interface uses a FTDI FT230X chip. To ensure maximum cross-platform compatibility, the default FTDI VID/PID has been maintained. Drivers for almost any system can be found on the FTDI driver website, being sure to specify the "Virtual Com Port" (VCP) option which is currently available at <http://www.ftdichip.com/Drivers/VCP.htm> .

A Full-size schematic of the USB interface can be found on the ChipSHOUTER github page. It is recreated here (in small scale):



## Saving Settings Between Power Cycles

By default settings such as the voltage, pulse type, and external trigger configuration are not saved between power cycles.

If you wish to save settings, you must call the `write_nvm` command to persist the settings in the internal EEPROM. You can always return the product to default state with the `Reset config` command.

## Python API Interface

The ChipSHOUTER can be manipulated via python which allows the device to be incorporated into more complex test setups. By writing custom python scripts the ChipSHOUTER can be used in conjunction with the chipwhisperer platform, oscilloscopes, and anything else that can be hooked into python. Below is a usage example for the Python API. For further examples and full documentation visit:

<https://github.com/newaetech/ChipSHOUTER>

and see the Python API. Documentation is also held for the API at <https://chipshouter.readthedocs.io/>

```
from chipshouter import ChipSHOUTER

#Configure ChipSHOUTER connection
cs = ChipSHOUTER("com3")

#Configure ChipSHOUTER pulse settings
cs.pulse.width = 80
cs.pulse.repeat = 1
cs.pulse.deadtime = 10
cs.voltage = 500

#arm and pulse
cs.armed = 1
cs.pulse = 1

#disarm
cs.armed = 0

#save our settings to EEPROM
cs.eeprom_write()
```

## XY(Z) Table Connection

The ChipSHOUTER requires steady positioning over the target, and some form of support is needed. A number of possible solutions can be used here depending on budget and required accuracy. It is also recommended to include positioning ability, to more precisely adjust the location of the ChipSHOUTER on the chip surface.

NewAE Technology Inc provides the ChipShover™ which has included mounting brackets and easy integration with the ChipSHOUTER environment.

Alternatively, you can use a variety of 3<sup>rd</sup> party solutions. These include anything from using a maneuverable vise (such as the Panavise®), a 3D printer gantry, or a commercial microscope positioning stage. See the ChipSHOUTER website for more information and examples of these positioning solutions. 3D STL files are available for printing various interconnect solutions.

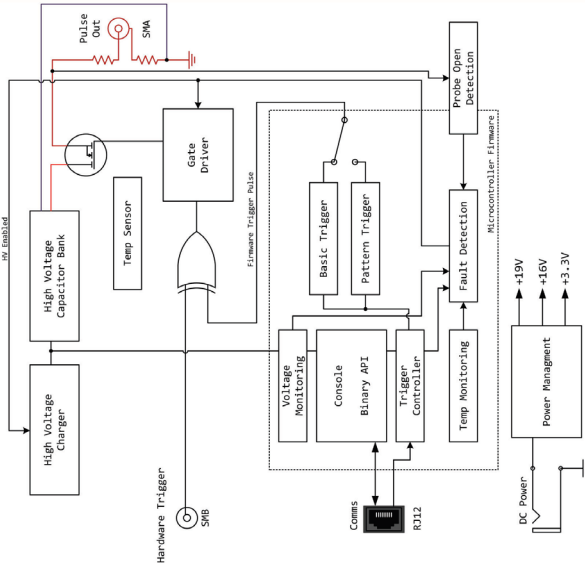
## Troubleshooting

Symptom	Possible Cause	Solution
Arming fails.	<ul style="list-style-type: none"> <li>Active fault condition.</li> </ul>	<ul style="list-style-type: none"> <li>Check faults via serial port or API.</li> <li>Check temperature of unit.</li> <li>Check for toggling signal on external inputs.</li> </ul>
Device resets during use.	<ul style="list-style-type: none"> <li>During high-current discharge, sufficient noise can cause self-reset of the device.</li> </ul>	<ul style="list-style-type: none"> <li>Change voltage settings and/or increase pulse width.</li> <li>Using API to detect device reset, recover from fault.</li> </ul>
Excessive “sensor faults” when using external trigger.	<ul style="list-style-type: none"> <li>Insufficient time for self-checks to occur between triggers.</li> </ul>	<ul style="list-style-type: none"> <li>Send “triggersafe” command before each external trigger event.</li> <li>Slow down external triggers.</li> </ul>
Using external hardware trigger causes probe open fault.	<ul style="list-style-type: none"> <li>External trigger is rapidly repeating many times (for example, being driven with an ungated signal generator), and the safety checks cannot occur.</li> </ul>	<ul style="list-style-type: none"> <li>Limit the fault insertion to occur in bursts at a time, providing a space for the device to perform safety checks.</li> <li>Send “triggersafe” command before each burst.</li> </ul>
Excessive “sensor faults” when using external trigger. --or-- Device disarms during use, even though external trigger used to pulse device.	<ul style="list-style-type: none"> <li>External trigger level is insufficient, causing triggering of MOSFET but the system monitor is unaware.</li> </ul>	<ul style="list-style-type: none"> <li>Confirm level of trigger input. If using 50-ohm termination temporarily turn this off to increase drive level.</li> </ul>


## ChipSHOUTER Users Manual: Troubleshooting

Symptom	Possible Cause	Solution
Charge fault occurs.	<ul style="list-style-type: none"><li>• Power supply is insufficient.</li></ul>	<ul style="list-style-type: none"><li>• Use different outlet.</li><li>• Replace 19V AC-DC power supply.</li></ul>
Device does not boot (check serial output).	<ul style="list-style-type: none"><li>• Internal FLASH corruption.</li></ul>	<ul style="list-style-type: none"><li>• Perform firmware update/recovery.</li></ul>
Odd smells or sounds from ChipSHOUTER.	<ul style="list-style-type: none"><li>• Internal damage</li></ul>	<ul style="list-style-type: none"><li>• DISCONTINUE USE OF DEVICE IMMEDIATELY.</li></ul>
Arcing sound from injection tip.	<ul style="list-style-type: none"><li>• Injection tip not tight, or injection tip damaged.</li></ul>	<ul style="list-style-type: none"><li>• Check injection tip connections with device disarmed.</li><li>• Replace injection tip.</li></ul>
USB interface drivers do not load.	<ul style="list-style-type: none"><li>• Drivers are not being loaded.</li></ul>	<ul style="list-style-type: none"><li>• Check FTDI website for latest VCP drivers.</li><li>• Use different USB port.</li></ul>
Continuous trigger faults.	<ul style="list-style-type: none"><li>• External trigger pin is being pulled to active state.</li></ul>	<ul style="list-style-type: none"><li>• Check if hardware trigger is set to active high or active low.</li><li>• Enable 50-ohm termination with active-high hardware trigger mode.</li></ul>
ChipSHOUTER goes into thermal shutdown.	<ul style="list-style-type: none"><li>• Excessive heat due to continuous operation.</li></ul>	<ul style="list-style-type: none"><li>• Reduce trigger rate.</li><li>• Disarm ChipSHOUTER in-between trigger events.</li><li>• Use external cooling.</li></ul>
Pattern trigger not working as expected.	<ul style="list-style-type: none"><li>• Pattern trigger does not match hardware trigger polarity.</li><li>• Probe characteristics mean injected output does not match the programmed pattern.</li></ul>	<ul style="list-style-type: none"><li>• Check hwtrig_mode is set active-high.</li><li>• Modify pattern trigger to achieve desired output.</li></ul>

## Schematic



**WARNING:** This ChipSHOUTER schematic is provided for reference and informational purposes only, and is copyright NewAE Technology Inc, 2016-2020. It is provided without warranty of any kind, express, implied or otherwise, including without limitation, any warranty of fitness for a particular purpose.

Title: <b>Title Page</b>		Lead Designer: Colin O'Flynn	
Rev: 07	Project: <b>ClipSHOOTER</b>	Approved: Yes	 NewAE.com
Date: 2024-01-12	Time: 1:38:01 PM	License: <b>NewAE</b>	
File: ew320_titpage.SchDoc		Copyright © NewAE Technology, Inc.	

**SUBS:**  
 PDZ10B,115 <=> PDZ12B,115  
 For 750k/820k HV resistors (sub ALL):  
 R1 1k4 1k4  
 R2 820k 750k  
 R24 820k 750k  
 R27 82k 30k  
 R41 820k 750k  
 R42 820k 750k

- Changes -04to-01:
  - \* Clamp diodes on SP-wrap input.
  - \* Add cap on PULSE-IN pin
  - \* Add buffer on PULSE-IN pin
  - \* Check UV pin routing
  - \* Check UV pins, test if any
- \* Fix VISA connect to MOSFET Drain
- \* Add LED on front panel driven by "charge" signal
- \* Fix op-amp feedback connections on current monitor
- \* Add LED on front panel driven by "charge" signal
- \* Reduce adjustable voltage range
- \* Fix power jack footprint
- \* Add 10k pull-up resistor on PN1/SC74A11/LFS
- \* Change current limiting resistor values (measure = 0.02, limit = 0.1)
- \* Add 100nF bypass capacitor
- \* Add fan controller
- \* Increase indicator current handling capability

Changes -01 to -02:

- \* Internal release/ Version Numbering

Changes -02to-03:

- \* Dump clamp diode P-N change for supply diode
- \* Remove MOSFET gate driver, change measurement circuitry to be like I/O probe tip
- \* Add D101 SA R211
- \* Add SVA for pulse-in, with 50 ohm option, use LVDS for routing closer to SCX6 gate
- \* Add termination on internal digital signals
- \* Add LED on front panel
- \* Change blood sensor values again (200K --> 300K for flow bleed, 33K --> 120K for fast bleed)
- \* Change blood sensor to MOSFET, due to potentially exceeding Vce rating of NPN

Changes -03 to -04:

- \* Fix LVDS P-N pin swap
- \* Add 100nF bypass capacitor
- \* Fix missing PULSIN connection to XN1B0A
- \* Add LDO to generate MOSFET bootstrap as regulated LV supply

Changes -04 to -05:

- \* Charge input to 19V
- \* Change MOSFET gate
- \* Change LDO type to low-dropout, smaller footprint
- \* Change input re-pd MOSFET to T/H for better layout

Changes -05 to -06:

- \* Charge MOSFET driver to include PNP switch-off transistor
- \* Add 100nF bypass capacitor
- \* Addition of ESD blocking diodes on dry signals

Changes -06 to 07:

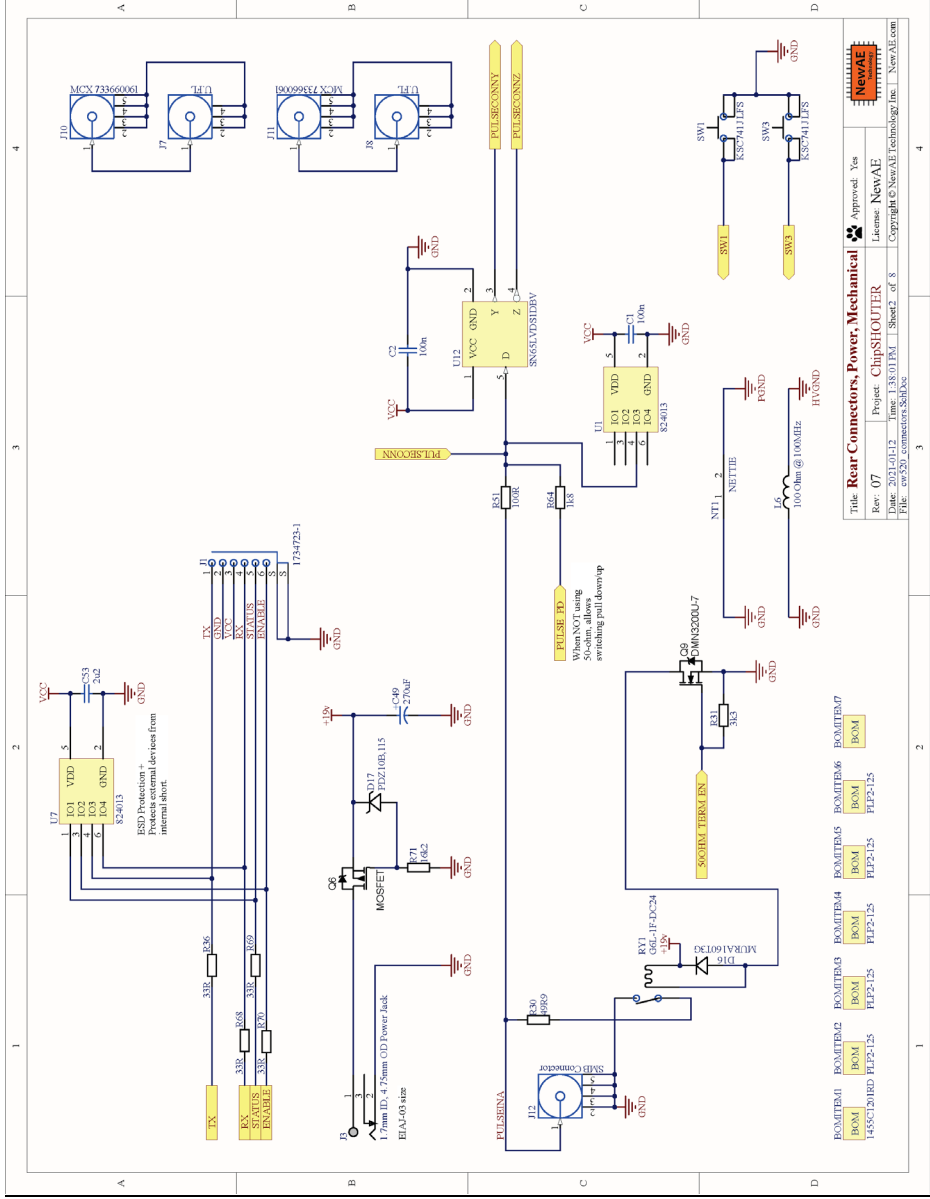
- \* Factor of power measurement capacitors (2x caps in series)
- \* Fix resistor values around detection of charge finished
- \* Change MOSFET P/N for torrey (and other switches) which has 30V rating instead of 20V
- \* Change SMA on PULSIN to Sx/B (avoid swa connector as HV)

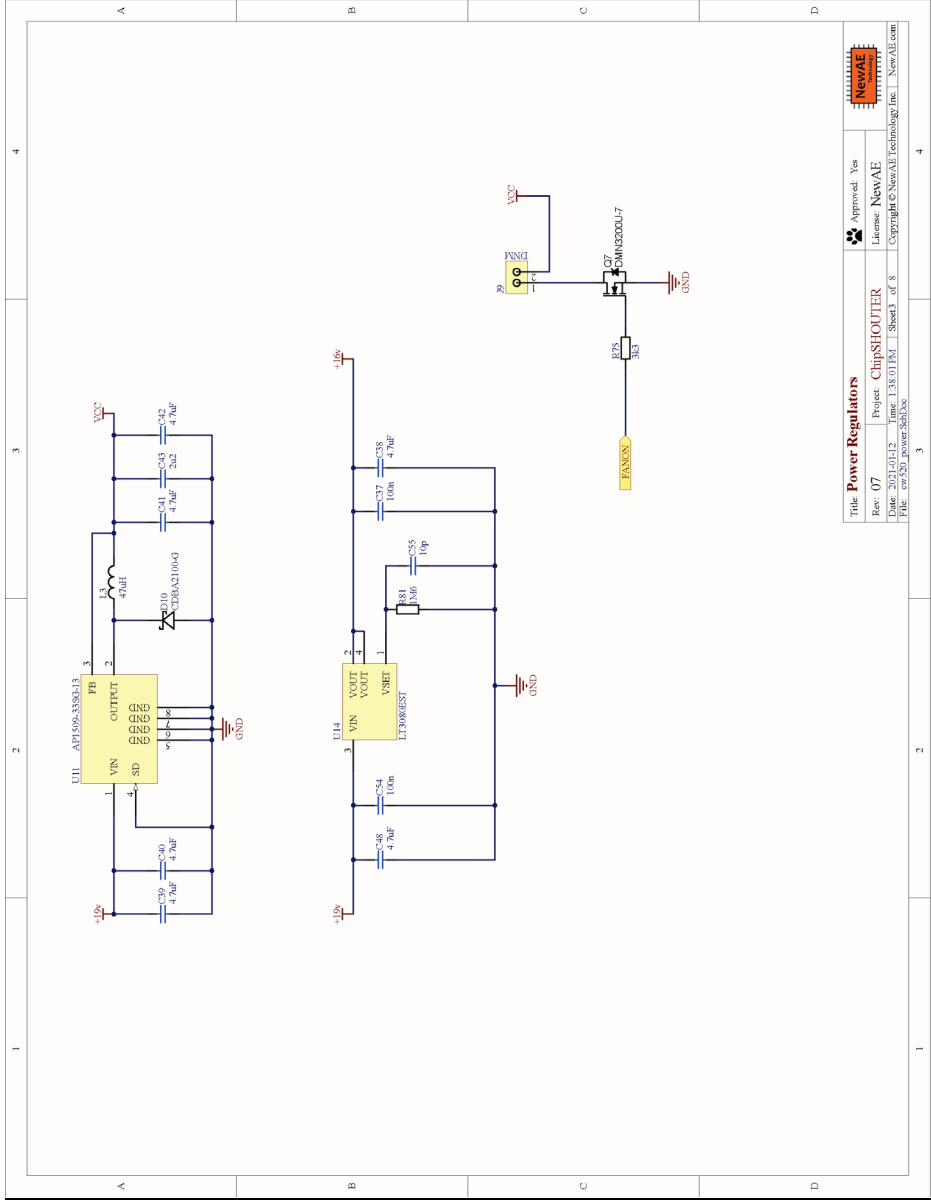
Changes 07A (Production, no PCB changes):

- \* Change HV feedback resistor to 82K due to supply issues (also change HV feedback as needed)
- \* Change PEZT13B to PEZT09 due to supply issues
- \* Change 100nF bypass capacitor to 100nF
- \* Change R12, 36K --> 33K
- \* Change R13, 270K --> 33K (prevent UV lockout during intense pulse firing)

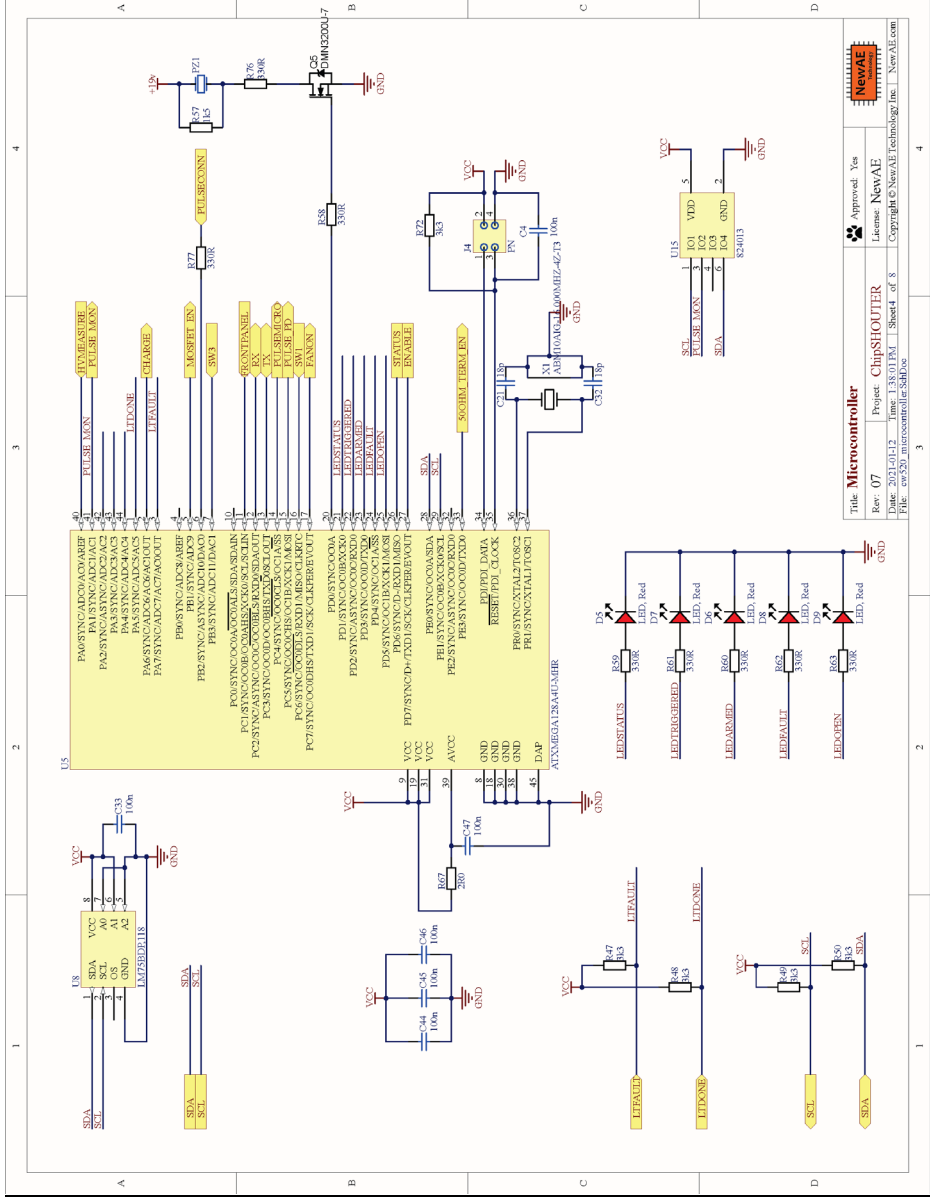
Changes 07B (Production):

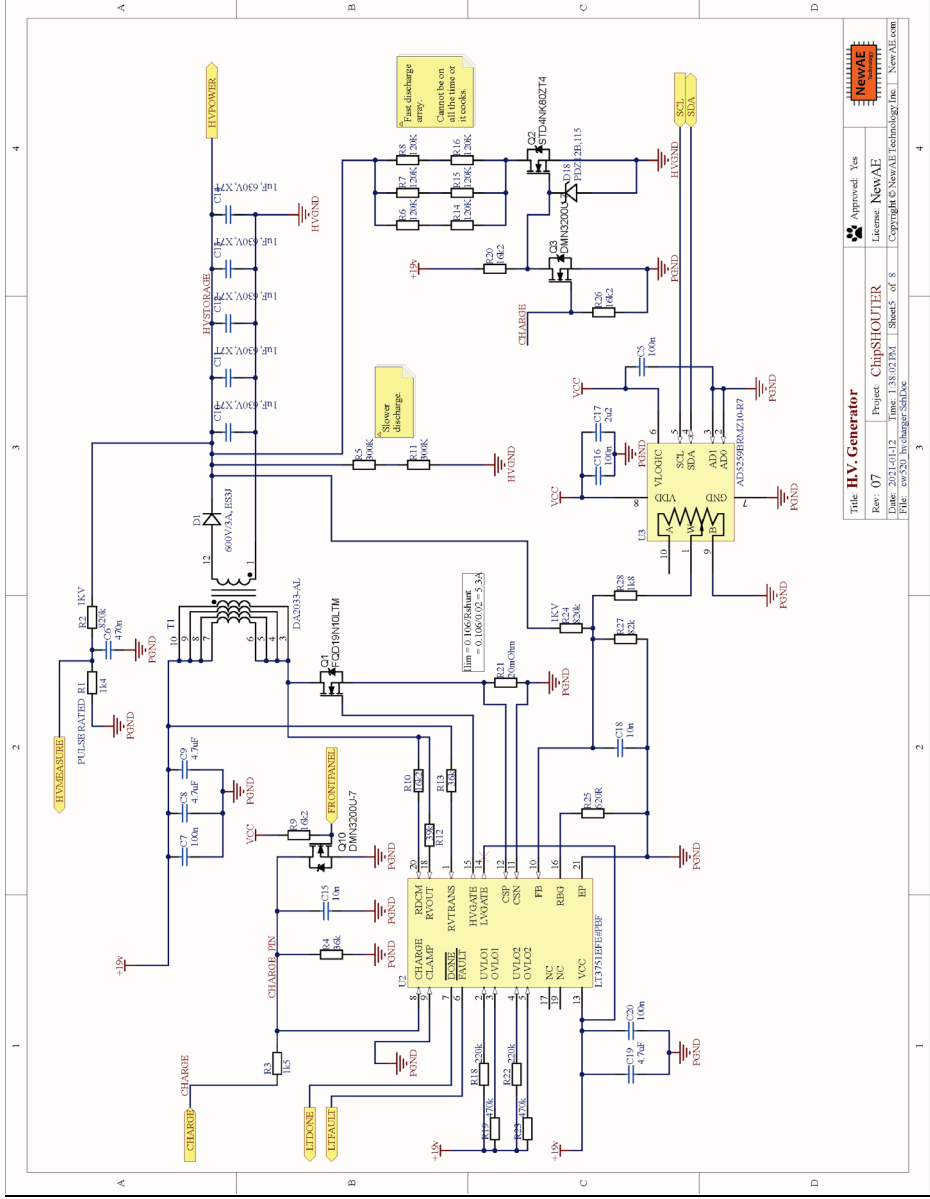
- \* Change R1 to R4 (firmware was built assuming R4, incorrectly specified as R13 in latest 07A release)





Title: <b>Power Regulators</b>		Approved: Yes	
Rev: 07	Project: ChipSHOUTER	License: NewAE	
Date: 2023-01-12	Time: 1:38:01 PM	Sheet 3 of 8	
File: ew520_power_sch.doc		Copyright © NewAE Technology Inc.	





Title: <b>H.V. Generator</b>		Approved: Yes	
Rev: 07	Project: ChipSHOUTER	License: NewAE	
Date: 2023-01-12	Time: 1:38:02 PM	Sheet 5 of 8	
File: ew520 hv_generator_sch.vcx		Copyright © NewAE Technology Inc.	







## Additional Links

If you have trouble with the ChipSHOUTER, some handy places to turn to:

- We post errata as part of the **GITHub repository** ([github.com/newaetech/ChipSHOUTER](https://github.com/newaetech/ChipSHOUTER))
- Our support forum (<https://forum.newae.com>) is actively monitored.
- Our support portal (<https://support.newae.com/>) has additional resources and you can open a support case.

## Manual Revision History

Revision	Release Date	Changes
1.4	31-OCT-2021	<ul style="list-style-type: none"><li>• Move this table into what was a blank page.</li><li>• Get spooked.</li><li>• Add links to errata + support.</li></ul>
1.3	25-FEB-2021	<ul style="list-style-type: none"><li>• Change cooling input, add schematic</li></ul>
1.2	7-DEC-2020	<ul style="list-style-type: none"><li>• Mention EEPROM write required in new firmware.</li><li>• Update API details</li></ul>
1.1	9-SEPT-2019	<ul style="list-style-type: none"><li>• Fix RJ12 pinout, small clarifications throughout, add more detail of external hardware trigger, add more detail of probe construction.</li></ul>
1.0	20-JULY-2018	<ul style="list-style-type: none"><li>• Initial release, minor fixes.</li></ul>

<u>Serial Command</u>	<u>Short Form</u>
<i>help</i>	<i>?</i>
<i>get id</i>	<i>g i</i>
<i>get state</i>	<i>g s</i>
<i>[get / set] voltage [150..500]</i>	<i>[g / s] v</i>
<i>[get / set] pulse width [80..1000]</i>	<i>[g / s] p w</i>
<i>[get / set] pulse repeat [1..10000]</i>	<i>[g / s] p r</i>
<i>[get / set] pulse deadtime [1..1000]</i>	<i>[g / s] p d</i>
<i>[get / set] arm_timeout [0...60]</i>	<i>[g / s] a</i>
<i>[get / set] hwtrig_term [1/0]</i>	<i>[g / s] hwt</i>
<i>[get / set] hwtrig_mode [1/0]</i>	<i>[g / s] hwm</i>
<i>[get / set] emode [1/0]</i>	<i>[g / s] e</i>
<i>[get / set] mute [1/0]</i>	<i>[g / s] m</i>
<i>[get / set] absent_temp [1..0]</i>	<i>[g / s] at</i>
<i>get fault</i>	<i>g f</i>
<i>get fault_active</i>	<i>g fa</i>
<i>get fault_latch</i>	<i>g fl</i>
<i>get fault [latched / current] &lt;type&gt;</i>	<i>g f [l / c] &lt;t&gt;</i>
<i>get temperature &lt;sensor&gt;</i>	<i>g temp &lt;s&gt;</i>
<i>get triggersafe</i>	<i>g ts</i>
<i>[get / set] pat_enable [1/0]</i>	<i>[g / s] pat</i>
<i>[get / set] pat_wave [011100...0]</i>	<i>[g / s] w</i>
<i>set pat_append [111000...0]</i>	<i>s wa</i>
<i>set fault none</i>	<i>s f n</i>
<i>arm</i>	<i>a</i>
<i>Arm</i>	<i>A</i>
<i>disarm</i>	<i>d</i>
<i>safe</i>	<i>S</i>
<i>pulse</i>	<i>p</i>
<i>Reset_config</i>	<i>Reset</i>
<i>reset</i>	<i>r</i>
<i>write_nvm</i>	<i>eeeprom</i>

# *CW520 ChipSHOUTER® User Manual*

Do you want to know how to use your new CW520 Electromagnetic Fault Injection Tool? While you have certainly had a nice stroke of good luck, because the CW520 ChipSHOUTER user manual is the *actual user manual* for this tool. Trying to figure out if ChipSHOUTER supports ECL logic level inputs (it does not, please don't try)? Trying to find out if ChipSHOUTER is powered by blockchain, and if so how you can invest? Answers to these questions and more can be found inside this user manual. As a special bonus, it is printed on a special combustible material that could *save your life* when hiking and lost in the woods.

"We do not review this type of material"

-*New York Times*

"Please stop contacting us"

-*Washington Post*

## About the Author

Luna is QA Manager for NewAE Technology Inc. When not performing important electro-static discharge testing duties she enjoys chasing squirrels and upcycling found organic material into new perfume lines. In her spare time she leads an initiative to reduce food waste in her local community. She lives in Halifax, NS, Canada.



Printed in Canada.

NewAE Technology Inc.

Self-Publishing Division

127 Joseph Zatzman Dr.

Dartmouth, NS. B3B 1W1. Canada.

Phone: 1-888-GLITCHY

\$19.99 CAD (\$99.99 USD)

ISBN: 978-1-9994176-0-4

