

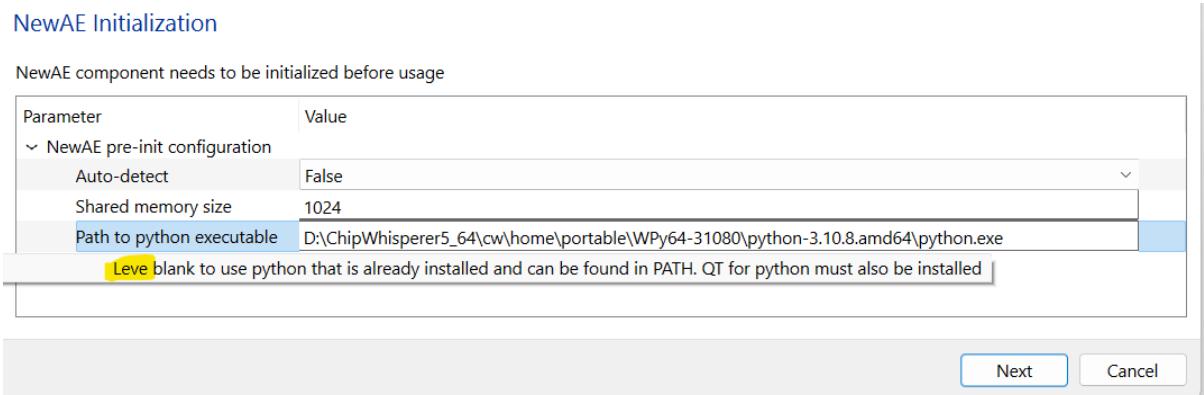
# Testování NewAE pluginu + TScopeWidget

Datum testování: 11.3.2024

Testovaný commit: 02593c40c89418cf66214ca0b76210c723cdf13c

Prostředí: Windows 10 Pro, Qt Creator 10.0.1, Qt 6.5.0 MinGW 64-bit, Python 3.10.8

- Vypadlo písmenko:



- Automaticky nainstalovaný python při instalaci CW na Windows je python 3.10.8. Plugin vyžaduje alespoň 3.11 (dle skryté nápovědy v path to python executable). V testování jsem pokračoval s python 3.10.8.
- Mít executable.py v buildu ve složce testing mi nepřijde vhodné. Chvíli mi zabralo, než jsme ho našel.
- Při první inicializaci pluginu, TraceXpert zastavil python kvůli timeout (tuším že to bylo tak 5 sekund – dost krátké). Při druhém pokusu to prošlo. Doporučil bych prodloužit timeout. Ale znova jsem tento problém nikdy nepozoroval.
- GUI: když dám apply parametrů u osciloskopu, vypíše se chyba „Download stopped! Trying to display trace with invalid index!“. Proč se stahuje/vypisuje záznamy, když vůbec neměřím? „Download stopped!“ chápu, že zastaví měření, pokud čekám na záznam o spotřebě, ale ani to nefunguje (v takovém případě program spadne). V každém případě se graf nemá překreslovat.

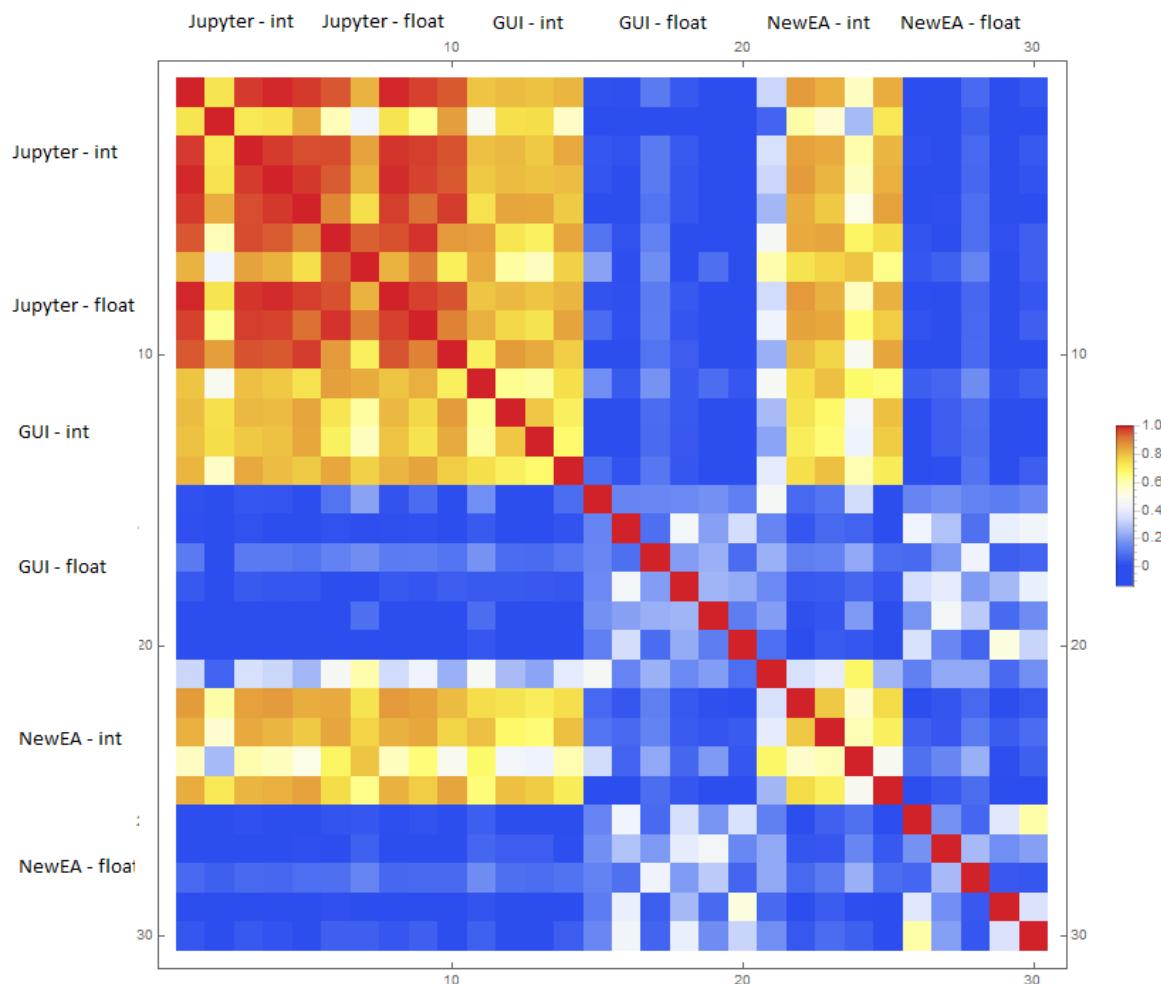
## Testování validnosti získaných záznamů o spotřebě

Naměřil jsem 5 záznamů spotřeby od každého z následujících setupů:

1. ChipWhisperer a Jupyter s `as int = true` (referenční měření)
2. ChipWhisperer a Jupyter s `as int = false` (referenční měření)
3. Vykreslená data pomocí GUI s `Get traces as int = true`
4. Vykreslená data pomocí GUI s `Get traces as int = false`
5. Vrácená data pluginem NewAE (buffer v `downloadTraces`), `as int = true`
6. Vrácená data pluginem NewAE (buffer v `downloadTraces`), `as int = false`

Z vrácených záznamů jsem vytvořil korelační matici. Ta by měla odhadnout nakolik si jsou záznamy podobné a v ideálním případě by korelace měla kopírovat korelaci z referenčního měření. Použitá datová struktura by neměla mít vliv na korelaci, protože nezáleží na absolutních hodnotách, ale na relativním vztahu mezi samplly (jinými slovy škálování dat nemá vliv). Všechna měření měla nastavené stejné parametry (default nastavení oscila, `timeout=0`, `number of samples=1000` a stejný kód na targetu).

Výsledná korelační matici:



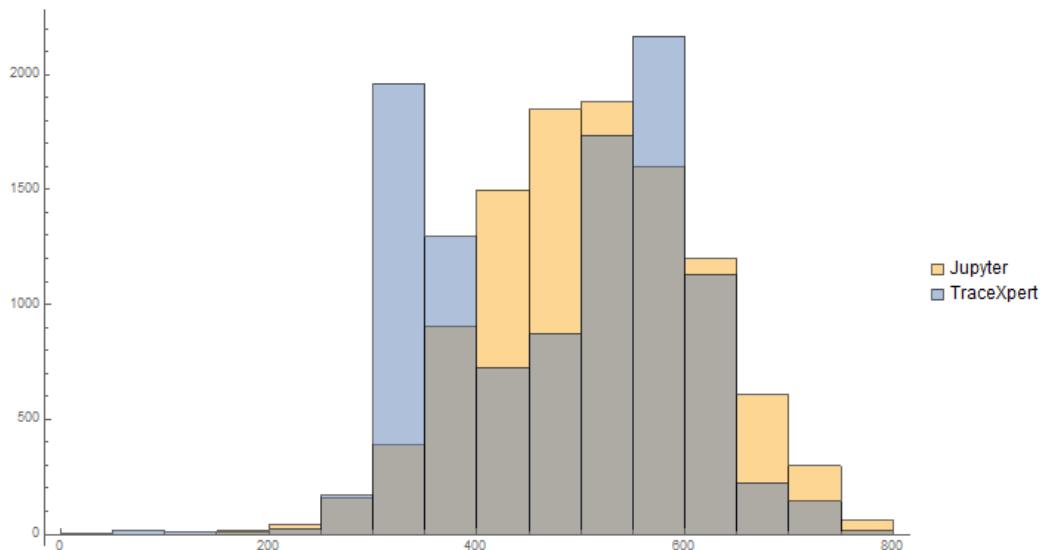
Závěr:

1. Plugin NewAE s `as int=false` vrací špatná data.
2. Plugin NewAE s `as int=true` funguje, ale data nekorelují takřka jak v referenčním měření
3. GUI tímto testem prošel (má stejnou korelací jako má plugin, který mu dodává data)

Pozorování samotných hodnot:

- GUI s `int=true` vypisuje racionální čísla (ne celočíselné). Dochází k přepočtu a to by nemělo.
- NewAE s `int=true` škáluje naměřené hodnoty konstantou 64. Jediný důsledek (co mě napadá) je, že místo nevyužitých most significant bitů jsou nevyužity lowest significant bits. Nevidím v tom benefit, akrát zpomalení. Pro zbytek testování jsem toto škálování zakomentoval, abych pracoval se stejnými hodnotami jako vrací Jupyter.
- Možné zmatení uživatele: V GUI a parametrech se nastavuje, že chceme strukturu `int`, ale v kódu se použije `uint` (chápu, že to je název převzatý z názvu parametru CW). Normálně by to nevadilo, protože vrácená čísla (mluvím o `as int=true`, ne o `floatech`) jsou maximálně 10bitové. Po přeskálování hodnotou 64 se ale může most significant bit nastavit na jedna a pak by byl rozdíl mezi `int` a `uint` (traceXpert správně hodnoty převede/použije, ale externí program by nemusel).
- Pro `as int=false`: hodnoty z referenčního měření jsou i záporné, ale plugin NewAE vrací pouze kladné hodnoty. Tohle má vliv na korelací.

Zkusil jsem vykreslit histogram hodnot naměřených v Jupyteru vs TraceXpertu. Set-up byl čistě defaultní (s výjimkou počtu samplů). Použito bylo 6 záznamů, kde každý záznam měl 1750 samplů:



Je zde vidět rozdíl. Můj předpoklad je, že když používáme zcela stejný set-up (jak HW tak nastavení oscila), pak by měly být samplify ze stejného rozdělení.

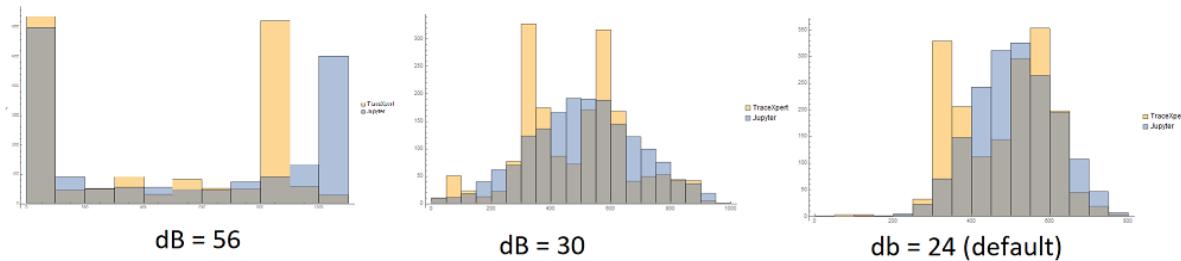
Závěrem bych dodal, že měření v Jupyteru vs v TraceXpertu se liší (viz histogram). Dle korelace soudím, že Jupyter vrací přesnější záznamy o spotřebě. Důvod rozdílu neznám.

## Testování parametrů osciloskopu NewAE

Testoval jsem pouze parametry, které jsem někdy reálně využíval. Jsou to parametry pro práci s triggerem, s ADC převodníkem, s hodinami a subparametry pod „gain“ parametrem.

Následující seznam obsahuje úplný výpis parametrů, které jsem testoval.

- Následující parametry fungovaly dobře:
  - scope.adc: offset, presamples
  - scope.clock: adc\_src, freq\_ctr\_src
- Parametry, které vyžadují přečíst aktuální hodnoty po provedení samotného nastavení parametrů. Pro tyto parametry TraceXpert nevrací aktuální hodnotu, ale hodnotu během nastavení. Pro správnou informaci tak musím dvakrát kliknout na `Apply` parametrů.
  - scope.clock.adc\_src (adc\_freq)
    - Problém je s `adc_freq`, který měří reálnou frekvenci snímkování. Po změně nastavení by se mělo chvíli počkat (doc: „May take up to 0.5s to stabilize after `adc_locked` is True.“) a pak přečíst naměřenou hodnotu. Jiné omezení dokumentace neklade, ale minimálně by se měl python znova dotázat na aktuální hodnotu.
  - Scope.clock.freq\_ctr (a freq\_ctr\_src): Tyto parametry slouží k měření/kontrole frekvence z různých zdrojů (externí vs interní hodiny). Ve výchozím nastavení se měří externí hodiny, které, když nejsou připojeny, mají hodnotu 0. Pokud chci měřit interní hodiny, pak musím nastavit `freq_ctr_src= "clkgen"`, dát `apply` a pro aktualizaci `freq_ctr` znova kliknout na `apply`.
- Scope.adc.decimate: pokud nastavím nevalidní hodnotu (nula nebo příliš vysoké číslo), pak spadne python a po chvíli i traceXpert. Pro validní hodnoty funguje dobře.
- GUI: když nastavím nesmyslně scope.gain.db (>56), GUI mě nijak neupozorní (ale vidím qWarning od pluginu) – patrně se nenastavuje state u daného parametru. Pozorováno i u jiných parametrů.
- scope.gain.db: Nastavil jsem maximální hodnotu (56 dB) a dostávám rozdílné výsledky pro měření v Jupyteru a TraceXpertu. V Jupyteru silně převládala hodnota 1023 a u traceXpertu 889. Viz histogramy pro jiné hodnoty (měřeno opakováně s podobnými výsledky):



Soudím, že tento parametr funguje dobře, ale rozdílné výsledky jsou důsledkem jiné chyby (bude souviset s jinými výsledky s default nastavením).

- scope.adc.trig\_count: Hodnota se aktualizuje při `apply` parametrů. Ale obecně s použitím by mohl být problém. `Trig_count` funguje tak, že počítá kolik samplů by se naměřilo během `trigger=high` a zároveň jeho hodnotu resetuje `scope.arm()`. Má tedy smysl aktualizovat hodnotu po měření a ne po změně parametrů. Samozřejmě mohu po měření kliknout na `apply` parametrů a pak se hodnota aktualizuje.

- Ještě jednu chybu jsem dokázal dostat, když samotný parametr byl prázdný (po reinicializaci zařízení), pak mi qWarning hlásil „Error reading subparam ADC->trig\_count“ (patrně kvůli prázdné hodnotě). Já ovšem nemohu změnit read-only hodnotu. Zajímavé je, že restart programu (celého traceXpertu) nepomohl (hodnota byla stále prázdná). Musel jsem odpojit a připojit zařízení. Tuto chybu jsem pozoroval dvakrát, ale nedokázal jsem ji replikovat. Šlo by vypnout kontrolu hodnoty?
- Scope.clock.clkgen (mul|div|freq) : Tato trojice parametrů spolu souvisí. Změnou mul/div se změní frekvence a naopak změnou frekvence se změní mul/div (plus změna mul může ovlivnit div a naopak). Problém je (předpokládám), že se traceXpert vždy pokusí změnit **všechny tři** parametry najednou/postupně. Výsledkem je, že clkgen\_div se zcela ignoruje, protože se nastavuje jako první a při nastavení frekvence se hodnota přepíše. Výsledkem je (pro uživatele) nepředvídané nastavení. Způsob řešení by mohlo být nastavovat jen změněné parametry, nebo povolit úpravu pouze frekvence (tu lze nyní správně nastavovat). Opět při nastavení nevalidních hodnot nevidím hlášení o chybě (qWarning se vypíše).
- Parametry s Run?: dost bych ocenil poznámku u parametru, který vysvětlí použití. Zda se script spustí předtím, než se aplikují parametry níže, nebo zda se aplikuje pouze default, nebo zda má přednost defaultní hodnota před námi definovanými.
  - Jak resetuji parametry na výchozí hodnoty? Dejme tomu, že nastavím špatně parametry. Jak je vyresetuji pomocí default\_setup? Z toho co pozorují default\_setup nemá nejmenší vliv na cokoliv, protože při inicializaci vidím rovnou defaultní hodnoty a parametry níže mají vždy přednost (a vždy se nastaví). To stejné platí např. pro cglitch\_setup. Ten má nastavit například scope.glitch.clk\_src=clkgen. Já ovšem v parametrech mám původní hodnotu „target“, která má přednost.