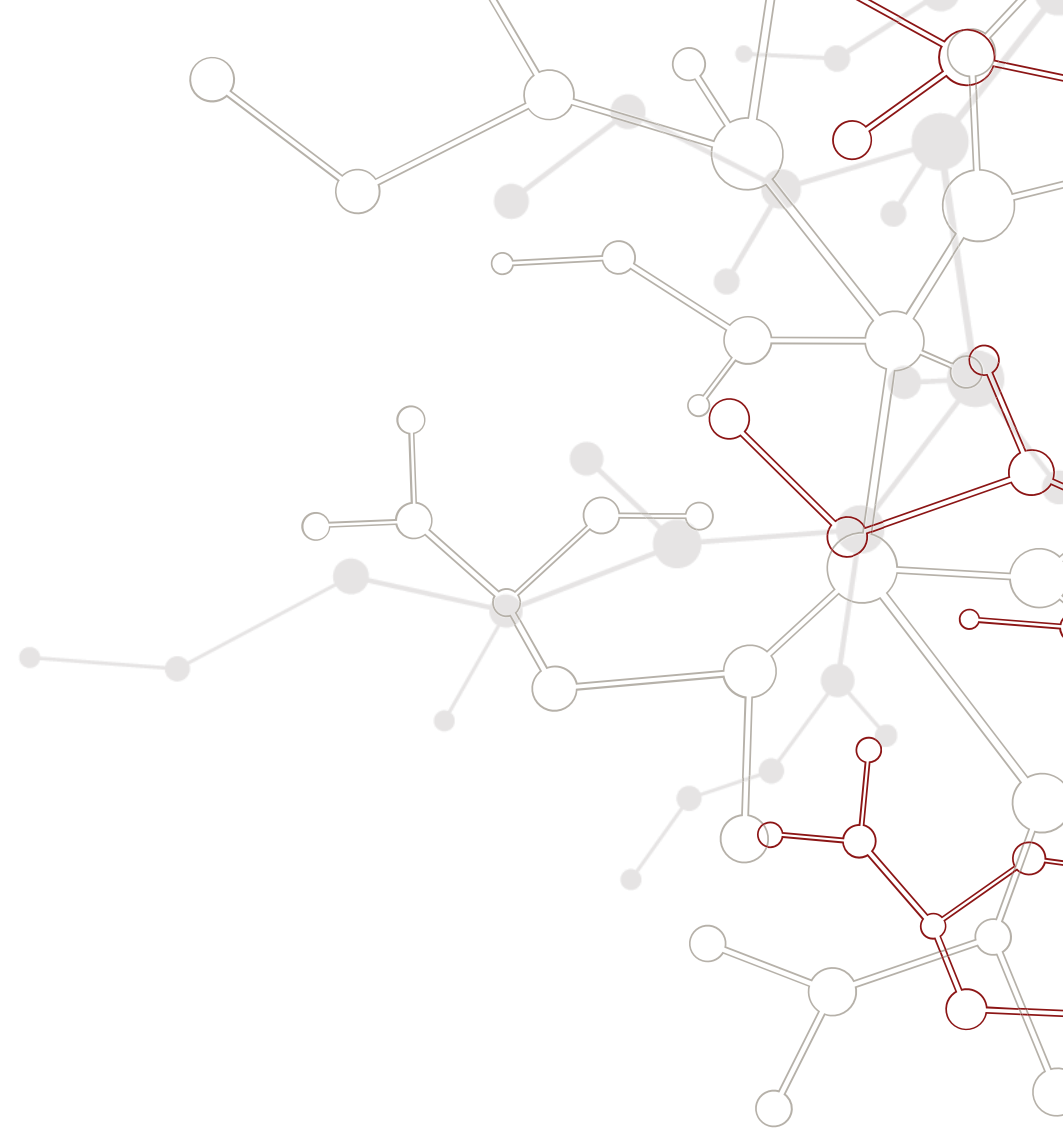


# Timing Pattern Generator GUI (TPGGUI) Development

## Status Update 2

Alexander Ng / Software Developer / EED/SWE-OPS

May 9, 2024



# Agenda

1. Recap

4. Pattern Considerations

2. Fixed Rate vs. AC Rate

5. Current State of TPGGUI

3. Pattern Composition

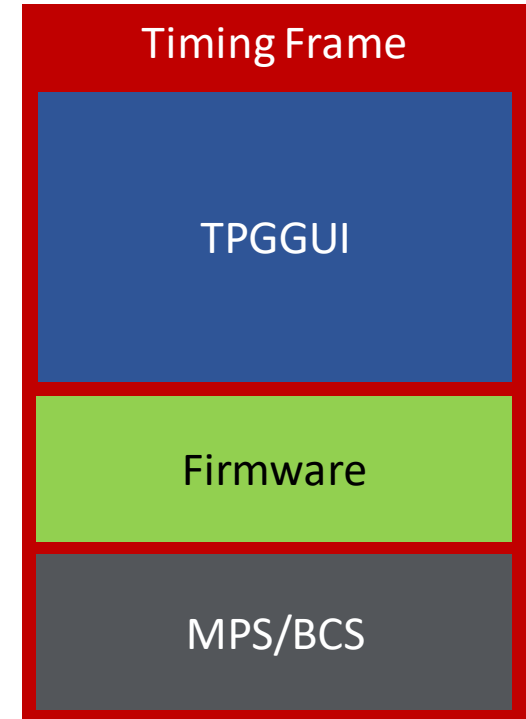
6. Roadmap

# Recap: What is a TPGGUI?

---

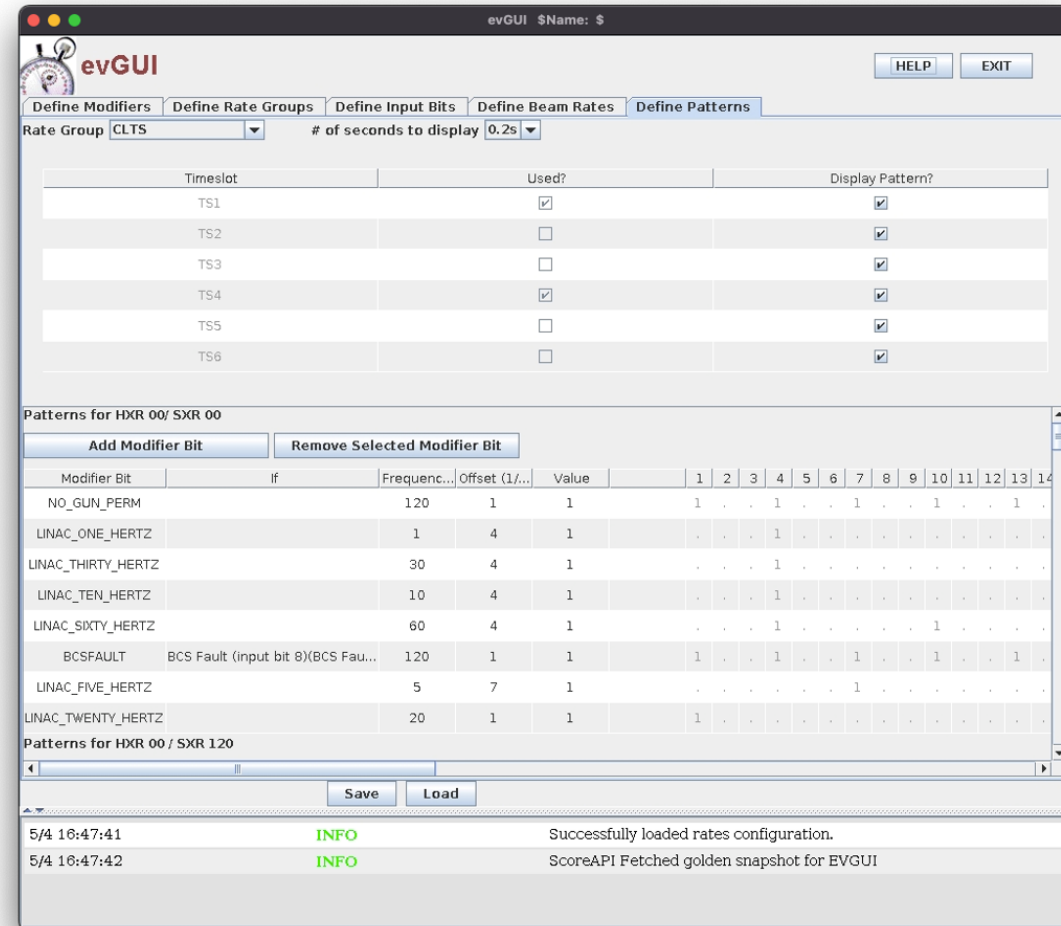
## Timing Pattern Generator (TPG) Graphical User Interface (GUI)

- The timing system sends out a timing pattern that contains relevant information to trigger the LASER system to generate beam and to the timing clients and devices along the SC LINAC.
- The timing pattern consists of a sequence of timing frames, which contains data set by firmware and software.
- The firmware defines information that is constantly sent out to identify frames, such as rate specific markers, pulse ID. Other bits in the timing frame are machine operation sensitive and need to be configured at a higher level.
- The TPGGUI will be the software tool used to program timing frames, updating information to the timing pattern to fulfill Operation requirements.
- The TPGGUI will allow operations to program more complex patterns and schedule beam going to multiple destinations while meeting machine and personnel safety requirements.



# Recap: Motivation for TPGGUI

- The Copper (Cu/NC) LINAC timing is driven by the EVG and manipulated with software tool: EVGUI.
- TPGGUI is analogous to EVGUI.
- Initially, the timing patterns were defined by python scripts in a GitHub repo. This pre-existing repo only handled single destination patterns without a visual/user friendly interface to manipulate the data or view possible conflicts on beam pulses assignments.
- Need to manipulate existing patterns and generate more complex patterns.
- Replaces manual construction/modification of requirements/parameters for existing patterns.
- Visually handle pulse assignment conflicts when scheduling beam to multiple destinations.



# Recap: What is a Timing Pattern?

---

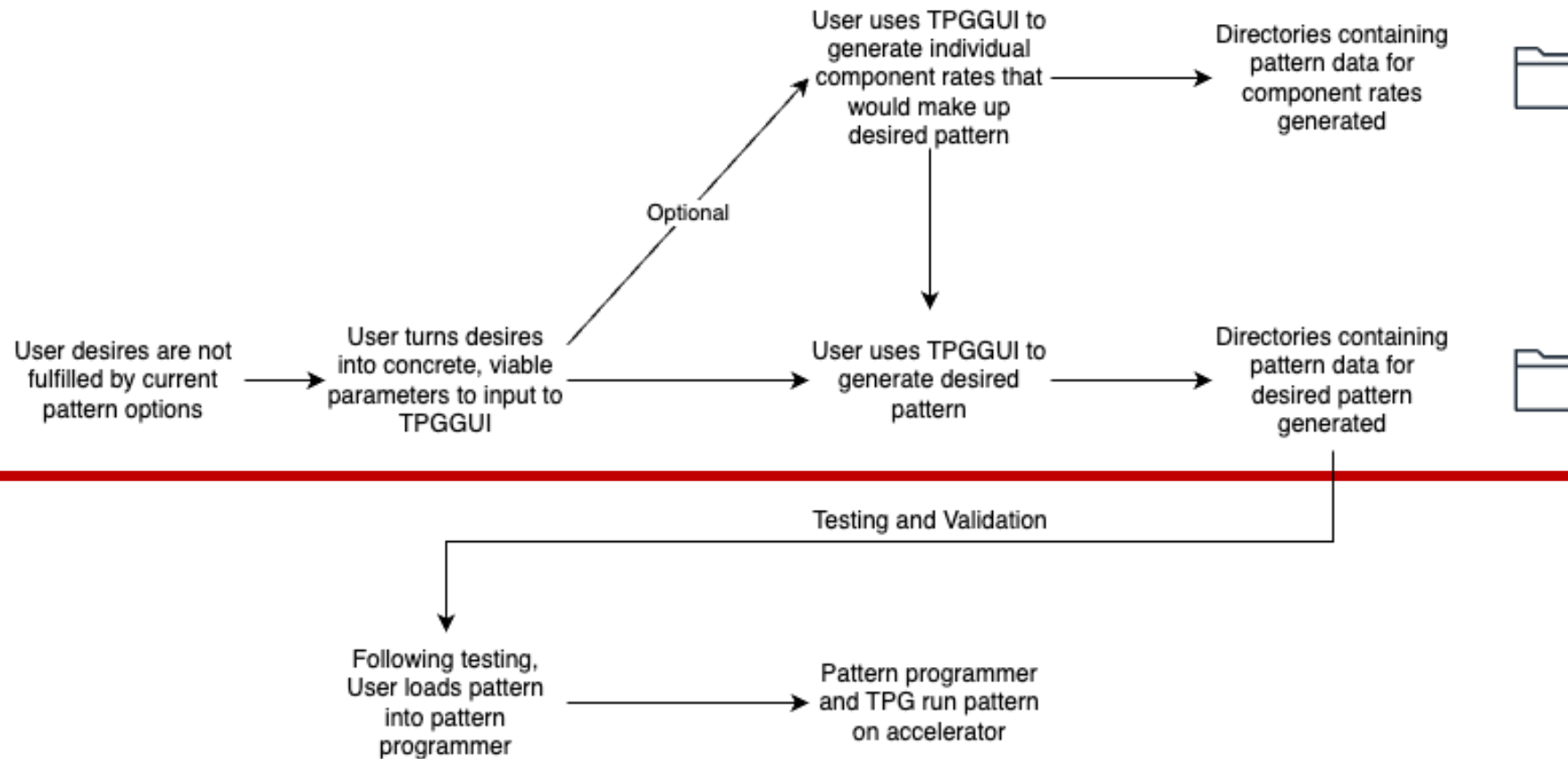
10 Hz on the mark

1. Send beam pulse
2. Wait for the 10 Hz marker
3. Go to instr. 1



- Set of instructions that tells the accelerator how, when and where to send a pulse.
- Each destination has a requested pattern of pulses.
- A timing pattern is the collection of patterns and rules that determine the behavior of each destination.

# Recap: How TPGGUI fits into operations



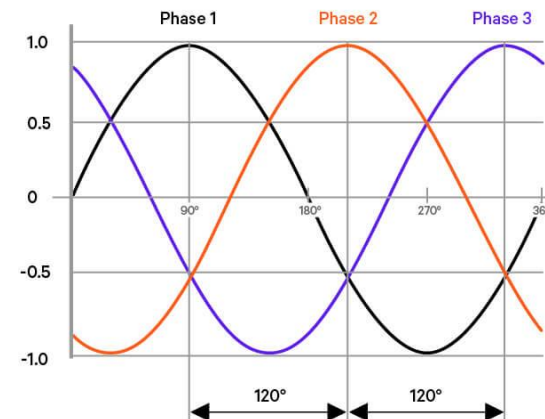
# Fixed Rate vs. AC Rate

## Fixed

- Synchronized to LLRF reference
- Timing interval: 0.98 second (1 TPGSEC)
- Frames per interval: 910000
  
- Base machine rate is 13/14 MHz, which does not fall evenly into 1 s interval. 0.98 s was selected as the repeating interval since it is exactly 910000 frames.

## AC

- Synchronized to AC power line
- Timing interval: 1 second
- Frames per interval: 360
  
- 60 Hz power \* 3 phases \* 2 zero crossings per cycle
- TS1 and TS4 are phase 1, TS2 and TS5 are phase 2, and TS3 and TS6 are phase 3



# Pattern Composition

---

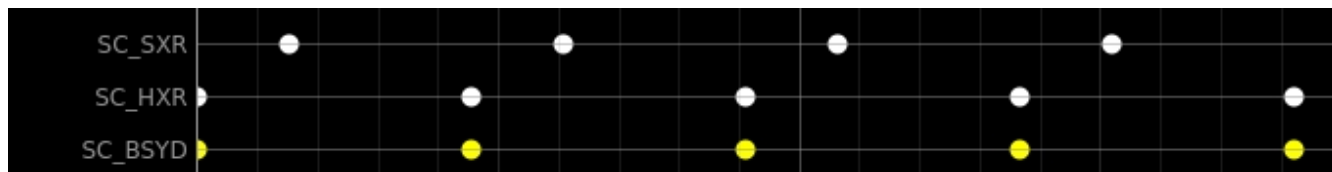
- When TPGGUI makes a pattern, it generates a set of JSON files and stores them in a directory. These files are consumed by TPG and firmware to drive the accelerator.
- Destination files
  - One per destination
  - Describes the requested rate for that destination
- Allow files
  - Multiple per destination, depending on the requested rate
  - Describes fallback rates (in case the requested rate is not allowed) and calculations based on beam class
- Control files
  - One per control engine (72 in total)
  - Describes auxiliary rules and control (non-beam producing) patterns
- Metadata file
  - One per pattern
  - Contains data used by TPGGUI and Pattern Selection GUI



# Pattern Considerations

## Default Offset

- If two destinations request beam on the same frame, beam will only go to the destination with higher priority (pulse stealing).
- To avoid unintentional conflicts, we defined 'default' offset values for each destination. Offset is defined as how many frames to delay the first pulse after the 1H marker (frame 0).
- Based on assumption of 71.4kHz max rate (13 frame spacing) to a single destination in multiple-destination pattern.



	Fixed Rate	AC
DIAGO	3	3 (TS4)
BSYD	0 (OTM)	0 (TS1)
HXR	5	3 (TS4)
SXR	7	0 (TS1)
LESA	0 (OTM)	3 (TS4)

# Pattern Considerations

---

## Lower Rates of Interest

- Lower allow rates: Depending on primary rate, 1/10/102/1020 Hz is set as fallback rates in case higher rates are not allowed. 0 Hz is always set.
- BPM calibration: If aggregate rate of all destinations is less than 200, set a 102 Hz with offset pattern to control engine 1
- Beam Synchronous Acquisition (BSA): Acquisition at lower rates. For each destination, attempts to find the closest rate to 1, 10, 102 Hz while still being under the primary rate, then generate an instruction set combining them. Pattern is assigned to control engine 3 + destination index.
- LINAC BSA: Same as above but calculated on the aggregate of downstream destinations. Important in instances of fallback pushing rate to BSYD, but BSYD primary rate is  $< 102$ . Currently being assigned by firmware.
- BSYD manual keepalive: If there is downstream rate, the BSYD must also request rate at the same timing in addition to its original request to allow for a fallback in case the downstream cannot receive.









# Current State of TPGGUI

---

# TPGGUI Project Goals




---

## Generate Complex patterns and upload them to TPG for Operation

- +  • Fixed Rate patterns interleaved within multiple destinations – offset one or more destinations from rate markers
- +  • Fixed Rate patterns pulse-stealing within multiple destinations – destination priority
-  • AC patterns pulse-stealing within multiple destinations – destination priority
-  • AC and Fixed Rate patterns pulse-stealing within multiple destinations – destination priority
-  • Burst patterns for multiple destinations
-  • Burst and Fixed Rate patterns within multiple destinations
-  • Burst, AC Rate and Fixed Rate patterns within multiple destinations – destination priority
-  • Upload a complex pattern, slightly modify behavior and save as a new complex pattern

### - Wishlist:

- Using the visualization tool to manipulate Burst patterns for multiple destinations
- Burst train of multiple burst patterns with different spacing

-  Completed + Enhancements
-  Newly Completed
-  In Progress

# Roadmap

---

- Continued maintenance and modification of existing patterns and pattern rules
- Bugfix burst patterns and refine requirements
- UI/UX improvements