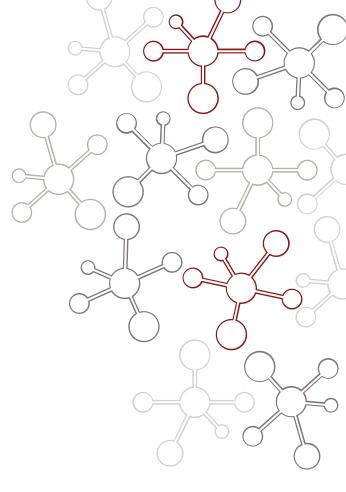
# Timing Pattern Generator (TPGGUI) Development

Status Update 1

Alexander Ng / Software Developer / EED/SWE-OPS

June 15, 2023



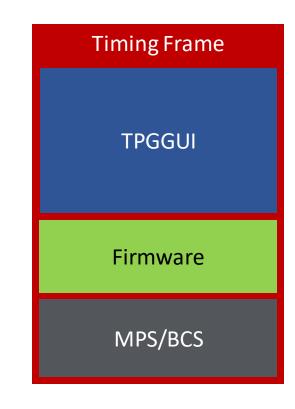




# What is a TPGGUI?

### 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.





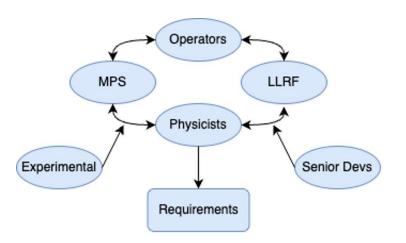
# Motivation for TPGGUI

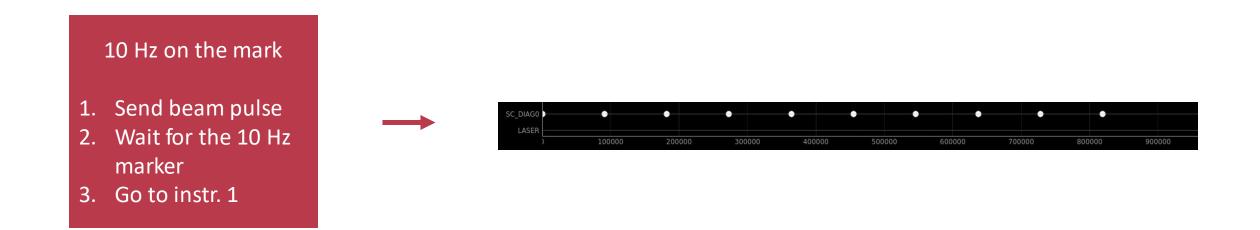
- Initially, the timing patterns were defined by python scripts in a GitHub repo. This pre-existing repo only handled single destination patterns without a visual/user friendly interface to manipulate the data or view possible conflicts on beam pulses assignments.
- The Copper (Cu/NC) LINAC timing is driven by the EVG and manipulated with software tool: EVGUI.
- TPGGUI is analogous to EVGUI.
- 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.

				60001	\$Name: \$															
<b>evGUI</b>															HEL	Р		EXI	т	]
Define Modifiers	Define Rate Gr		e Input Bits		Beam Rates	Define	Pati	terr	าร											
Rate Group CLTS	•	#ofseco	nds to disp	lay 0.2s	·															
	Timeslot			ι	Jsed?							Di	splay	y Pat	tterr	1?				
	TS1				V									V						
	TS2													V						
	TS3													V						
	TS4				<i>V</i>									V						
	TS5													V						
	TS6													2						
Patterns for HXR 0	0/ SXR 00																			-
Add Modif	ier Bit	Remove Sel	ected Modif	ier Bit																
Modifier Bit	(		Frequenc	Offset (1/	Value		1	2	3	4	5	6	7	8	9	10	11	12	13	1
NO_GUN_PERM			120	1	1															1
LINAC_ONE_HERTZ			1	4	1					1										
LINAC_THIRTY_HERTZ			30	4	1					1										
LINAC_TEN_HERTZ			10	4	1					1										
LINAC_SIXTY_HERTZ			60	4	1					1						1				
BCSFAULT	BCS Fault (input k	oit 8)(BCS Fau	120	1	1		1			1			1			1			1	
LINAC_FIVE_HERTZ			5	7	1								1							
LINAC_TWENTY_HERT2	2		20	1	1		1													
Patterns for HXR 0																				
	II																			
•		Save	Load																	
• • • • • • • • • • • • • • • • • • •		UNITO .			Successful	lly loaded	rat	es d	conf	igu	ratio	on.								
		INFO																		
• <del>•</del>		INFO			ScoreAPI I	Fetched g	gold	en	snaj	psh	ot fo	r E\	/GU	Ι						

- Code-specific Requirements:
  - Clean and maintainable code
  - UI to manipulate parameters
  - Reduce complexity for user/data visualization
- Machine Operation Requirements:
  - Keepalive triggers, BPM calibration triggers, BSA triggers, lower rate destination aware triggers
  - Maintaining Laser/Kicker thermal stability
- Coming soon: Formal Specification Document

### **Stakeholders**





• Set of instructions that tells the accelerator how, when and where to send a pulse

### **TPGGUI** Pattern Categories

### **Timing Patterns Definition**

#### **Standard Patterns**

• Fixed, AC Rate, and Burst (PRD) - commissioned

#### **Complex Patterns**

• Interleaved and pulse stealing

#### **Experimental Patterns**

• Special requirements for experiments such as: offset or odd frequencies.

#### Purpose for Categorization:

- "Protect" commissioned, established patterns: the Standard Patterns.
- Use the Standard Patterns as the base to re-define parameters and combine multiple of them to create additional complex patterns.

# **TPGGUI Project Goals**

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

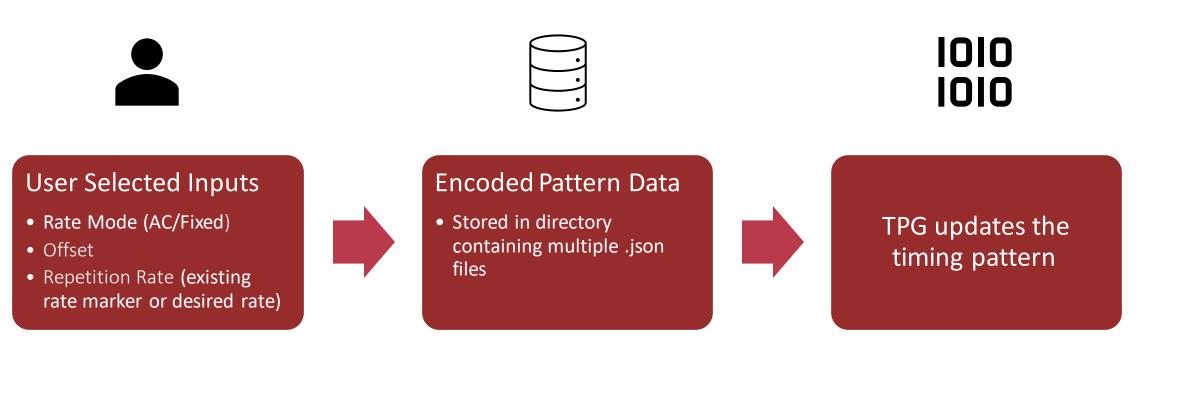
- Fixed Rate patterns interleaved within multiple destinations offset one or more destinations from rate markers
- Fixed Rate patterns pulse-stealing within multiple destinations destination priority
- AC patterns pulse-stealing within multiple destinations destination priority
- AC and Fixed Rate patterns 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

#### SLAC EED/SWE-OPS

### **TPGGUI** Data Flow



- TPGGUI code is a python-based repo on git (clone with eco TPGGUI)
- The patterns are in JSON files organized in folders by destination
- The interface is PYDM-hybrid (.ui and .py files)



### Current State of TPGGUI

	and Home										
	erator Fixed Rat	es Reference									
Pattern Brows	ser										
CON	IFLICT							<b></b>			
DUMPBSY	_KEEP										
SC_C	DASEL										
SC	c_sxr		-			• • •	•	• • • •	• • •		
sc	C_HXR — • - •	• • •	•	• • •	•	• • •					• • • • •
SC	_BSYD										
SC_0	DIAGO							<b>_</b>			
L	LASER										
	15800		1	5000		16200		16400		16600 1	6800
Pattern Selec	t									Pattern Generation	
										Description	Offset
Destination	Std. Fixed	SC_DIAG0	-	SC_BSYD Std. Fixed		SC_HXR Complex		SC_SXR Complex •	SC_DASEL Std. Fixed *		
	Name	Name		Name		Name		Name	Name	Fixed Rate  AC Rate	(1 MHz pulses)
Туре		* 500 Hz	*	0 Hz		FR 23.0 kHz	¥	FR 23.0 kHz off 20 *	0 Hz *		
	0 Hz	###		###		###		###	###	Rate (Hz) 928571 *	
Туре	0 Hz ###	###							View Selection	Loading selection to prowser: [" '500 Hz', '0 Hz', 'FR 23.0 kHz', 'I	UHZ', 🔺

#### Sections

- 1. Pattern Browser: Visual display of selected patterns
- 2. Pattern Select: Destination by destination selection of patterns
- 3. Pattern Generation: Creation of new patterns





- Testing generated patterns in operation
- "Complex" Pattern Generator: used to generate patterns with multiple destinations
- Support changes to AC rates, burst patterns
- Synchronizing patterns to any rate markers



Carolina Bianchini Mattison

Matt Weaver

Mike Zelazny

