

# Automated Management of SPARK Module IOCs in SPEAR3

**Filippos Toufexis, PhD**

Electronics Engineer, SSRL, SLAC

[ftouf@slac.stanford.edu](mailto:ftouf@slac.stanford.edu)

Libera Workshop, June 17<sup>th</sup>, 2020





**SSRL/SPEAR3**

**3<sup>rd</sup> Generation Synchrotron**

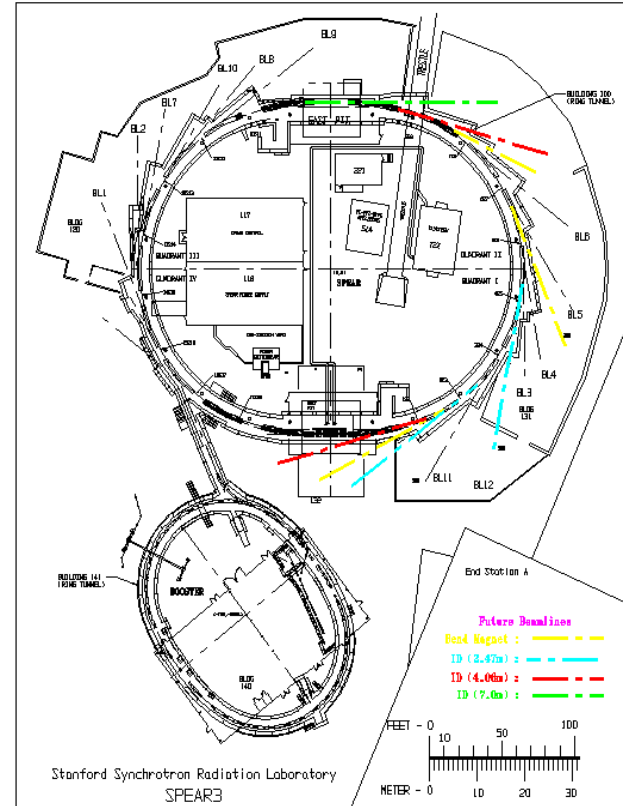
**3 GeV / 500 mA**

**234 m circumference**



# SPEAR3 Accelerator Complex

- Linac:
  - 120 MeV
  - Thermionic RF gun
  - 2.856 GHz
  - Commissioned in 1990
- Booster:
  - 3 GeV
  - 10 Hz resonant circuit
  - 358 MHz
  - Commissioned in 1990
  - Top-Off injection every 5 minutes
- SPEAR3 Storage Ring
  - 3 GeV
  - 500 mA
  - 476 MHz
  - 10 nm emittance (6 nm with new septum)
  - Commissioned in 2004



# BPM Systems in SPEAR3 Accelerator Complex

- Transport Lines:
  - 1990's-era Bergoz BPM processors with  $\sim 300$   $\mu\text{m}$  resolution
  - Last two BTS BPMs upgraded to SLAC uTCA processor with  $\sim 50$   $\mu\text{m}$  resolution
  - Last two BTS BPMs physically different than the rest (longer, smaller ID)
  - Recently purchased 2 + spare SPARK-EL to replace uTCA due to maintenance issues
- Booster:
  - Legacy system comprised single BPM processor and multiplexers to switch BPMs
  - Currently have 2 + spare SPARK-ERXR in the booster
  - Designing a multiplexer system to replace legacy multiplexers
- SPEAR:
  - Bergoz processors for operations and FOFB (no intention of upgrading)
  - In-house Echoteck TbT processors for Accelerator Physics are reaching end of life
  - 1 SPARK-ERXR for Accelerator Physics

# Considerations for SPARK deployment

- SPEAR3 operates with a very small team
- Most Operators do not have the skills to log in and configure modules
- **We need an Operator to be able to swap a failed module in the middle of the night with only guidance over the phone**

## uSD-based Deployment:

- Easy to swap modules
- uSD spares hard to maintain
- Need 2 uSD per BPM
- Extremely hard to upgrade firmware

## Network boot-based Deployment:

- Easy to maintain and upgrade
- Need to involve the network group to swap modules

# Network boot with Hostname/PV prefix DB

- Perform Network boot
- Hostnames based on BPM Processor model + Number, e.g. SPARK\_EL\_5
  - No connection with a physical BPM
- BPM Processor to Physical BPM mapping:
  - Handled with a Soft IOC
  - Startup script on the BPM Processor queries the Soft IOC **which BPM am I assigned?**
- **If module fails in the middle of the night:**
  - Operator assigns a spare processor to the BPM the previous module fails in EDM panel
  - Physically swaps the modules & powers up the new module
  - Network group updates a repaired module in DHCP later during normal business hours
- We can bring spare modules online to check them without affecting ops

# Network boot with Hostname/PV prefix DB

## Network Infrastructure



TFTP server supplies the boot image



DHCP server holds mapping MAC address -> hostname  
00:A0:C9:14:C8:29 -> SPARK\_EL\_4



Stores IOC files & tmp files  
Mount point for each hostname:  
    /.../SPARK\_EL\_3  
Mount point for each physical BPM:  
    /.../BTS\_BPM8

## Soft IOC Database



PV	Value
SPARK_EL_0	→ BTS-BPM09
SPARK_EL_1	→ BTS-BPM04
.....	
SPARK_EL_N	→ TEST
.....	
SPARK_EL_M	→ UNASIGNED

## EDM Panel

Transport Lines

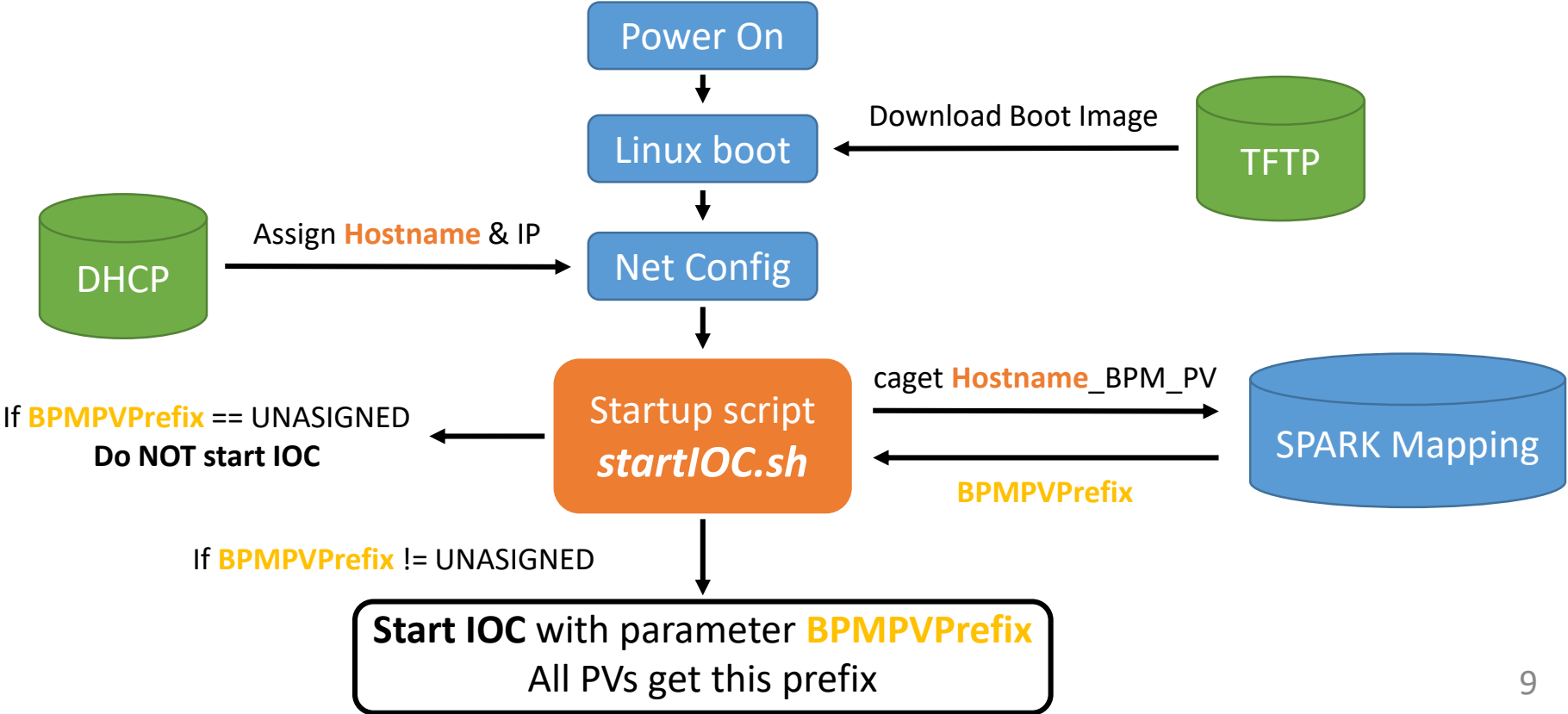
SPARK_EL_0	BTS-BPM09	▼
SPARK_EL_1	LTB-BPM02	▼
.....		

Booster

SPEAR



# SPARK Boot Sequence (simplified)



# Looking under the hood – Transport Line Assignment

Get the hostname

```
UNITNAME=`hostname`
```

spark-el-01

Get the Transport line assignment, write to tmp file

```
exec $CAGET -t SPARKMgr:$UNITNAME:BPM:TL |& tee $IOC_CONFIG/transportLineName
```

SPARKMgr:spark-el-01:BPM:TL

Check that caget did not return junk by measuring the number of words in the tmp file

```
cat $IOC_CONFIG/transportLineName | wc -w > $IOC_CONFIG/transportLineNameWC
```

```
if [ ! "$(head -n 1 $IOC_CONFIG/transportLineNameWC)" -eq 1 ]  
then
```

```
    echo "SPARKManager soft IOC not accessible" |& tee -a $STARTUP_ERROR  
    exit 1
```

```
fi
```

We have the transport line assignment

```
TRANSPORTLINE="$(head -n 1 $IOC_CONFIG/transportLineName)"
```

BTS

# Looking under the hood – PV Prefix

Similarly get BPM #

```
BPMNUMBER="$(head -n 1 $IOC_CONFIG/bpmName)"
```

BPM08

Calculate PV Prefix

```
BPM_PREFIX="SPARK-$TRANSPORTLINE-$BPMNUMBER"
```

Series of checks on PV Prefix...

```
if [ "$BPM_PREFIX" == "SPARK-LTB-BPM08" ]  
then  
  CAPUT_EXIT_STATUS=$(exec $CAPUT -t SPARKMgr:$UNITNAME:STATUS.VAL  
    "Error: SPARK-LTB-BPM08 doesn't exist")  
  exit 1  
fi
```

SPARK-BTS-BPM08

Check that IOC with this BPM Prefix has not already started.

```
exec $CAGET -t $BPM_PREFIX:input:max_adc |& tee -a $IOC_CONFIG/iocAlreadyRunning
```

SPARK-BTS-BPM08:input:max\_adc

If caget returns more than one word, then IOC was not accessible, then not running.

# Looking under the hood – Setting PVs

Similarly get BPM Settings  
Don't forget to check caget  
did not return junk!

```
exec $CAGET -t SPARKMgr:SPARK-$TRANSPORTLINE-GLOBAL:PickupOrientation | &  
tee $BPM_CONFIG/pickupOrientation
```

Write a number of EPICS  
commands that will be  
loaded in st.cmd to set the  
PVs

```
echo "dbpf(\"$BPM_PREFIX:dsp:pickup_orientation_sp\", \"$PICKUPORIENTATION\")" >  
/tmp/BPMSettings.env
```

**st.cmd:**

```
ioclnit()  
  
## Load Settings  
< /tmp/BPMSettings.env
```

# Looking under the hood – Start the IOC

Write the PV Prefix in env  
file

```
echo "epicsEnvSet(\"IOCNAME\", \"$BPM_PREFIX\")" > /tmp/IOCname.env  
echo "-> IOC name $BPM_PREFIX"
```

st.cmd:

```
< /tmp/IOCname.env
```

Start the IOC

```
exec $PROCSERV $PROCARGS $PORT $TOP/st.cmd
```

**NOTE: This is a very simplified set of commands  
Current startIOC.sh is 760 lines long.  
Includes several checks and logging commands**

# EDM Panels

Libera BTS BPM Manager

	Steering	Trig. Delay	Att. (dB)	Sensitivity	Pre-trigger	Post-trigger	Threshold	Kx (nm)	Ky (nm)	Xoff (nm)	Yoff (nm)	Global RB
BTS-BPM01												
BTS-BPM02												
BTS-BPM03												
BTS-BPM04												
BTS-BPM05												
BTS-BPM06												
BTS-BPM07												
BTS-BPM08	Excluded	0 0	0 0	High High	10 10	50 50	500 500	-5714800 -5714800	7682200 7682200	-1934900 -1934900	-1851100 -1851100	diagonal 1 linear 1
BTS-BPM09	Excluded	0 0	0 0	High High	10 10	50 50	500 500	-8658200 -8658200	8285400 8285400	-1197500 -1197500	-167190 -167190	diagonal 1 linear 1

Libera Transport Line BPM Processor Manager

Hostname	Transport Line	BPM #	IOC Start/Stop/Restart			Status
b140-tl-spark-el-01	BTS <input type="checkbox"/>	BPM09 <input type="checkbox"/>	Start	Stop	Restart	OK
b140-tl-spark-el-02	BTS <input type="checkbox"/>	BPM08 <input type="checkbox"/>	Start	Stop	Restart	OK
b140-tl-spark-el-03	UNASSIGNED <input type="checkbox"/>	BPM01 <input type="checkbox"/>	Start	Stop	Restart	OK

**Non-default Global BTS BPM Settings**

Pickup orientation:  diagonal  linear    Number of bunches:

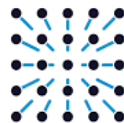
Calculation equation:  linear  diagonal    Data averaging:

# Status

- 2 SPARK-EL installed in parallel with the uTCA with 10 dB couplers
- **We have demo'ed the boot process**
- Next steps before transition to Ops:
  - Implement network authentication, time...
  - Port steering codes to use SPARK-EL PVs
  - Integrate in the control system



# Acknowledgements



INSTRUMENTATION  
TECHNOLOGIES

- Jeff Corbett
- Daniel Morataya
- Chris Ramirez
- Clemens Wermelskirchen
- Shantha Condamoor
- Peter Leban
- Matjaž Žnidarčič



