ISBN: 978-3-95450-212-7 ISSN: 2673-5482 doi:10.18429/JACoW-CYCLOTRONS2022-MOP0015

# TRIUMF LLRF CONTROL SYSTEM UPGRADE

Xiaoliang Fu*, Ken Fong, Qiwen Zheng, Thomas Au, TRIUMF, Vancouver, Canada

*Abstract*

The LLRF system for Transfer line from Drift Tube LINAC(DTL) to Superconducting Linac (SCB) (DSB) was an analog-digital hybrid system running at 35.36MHz. The system controls the amplitude/phase and tuning for a buncher cavity on the beamline. During the 2022 October shutdown, the system is upgraded to a new fully-digital LLRF system. The new digital LLRF system is based on TRIUMF's universal LLRF hardware with a new firmware. Instead of using a VXI mainframe, the new system adopts a NIM bin and uses USB communication with the local control PC. The amplitude/phase regulation is implemented in the FPGA firmware, and the tuning loop is implemented in the PC software, but driven by the FPGA. The Debian 11 linux OS is running on ARM CPU, and the new digital LLRF system works as a standard window HID device. The linux OS allows the firmware be updated in-situ using Ethernet communication. The detailed design is described in this paper.

## INTRODUCTION

The previous LLRF system for DSB was an analog-digital hybrid system based on VXI mainframe that controls the amplitude/phase and tuning of the cavity. The LLRF system works at 35.36MHz, and this frequency is generated from the harmonics of the 5.89MHz main reference source. The tuning loop was implemented with phase comparison method, then upgraded with the sliding mode control. The RF part of the system works great for years. However, the motor control function of the tuning loop has minor problem related to the limit switch. Meanwhile, the VXI mainframe is obsolete and hard to buy for the future system. To fix the motor problem, a new digital LLRF system is designed to replace the existing system. The new system is based on ZYNQ FPGA and installed in the NIM bin crate instead of VXI mainframe. The daughter board is TRIUMF's universal LLRF hardware [1], and the mother board is equipped with RF signal conditioning circuit. The local control PC communicates with the LLRF system through USB and Ethernet. The new system adopts 35.36MHz reference signal and the output of the system is phase-locked to it. There are two outputs provided by the new system: one is for driving the amplifier chain and the other one is used as the reference for Isotope Separator and Accelerator(ISAC) II. The hardware of the new digital LLRF system is shown in Fig. 1.

## SYSTEM DESIGN

The system requires amplitude/phase control, tuning(step motor) control, and the global phase control. The 35.36MHz signal is used as the reference input. There are one cavity pickup signal and two output signals in the system. The system works in generator driven mode. From the previous experience of TRIUMF's digital LLRF system [1, 2], the new system for DSB is designed as below in Fig. 2.

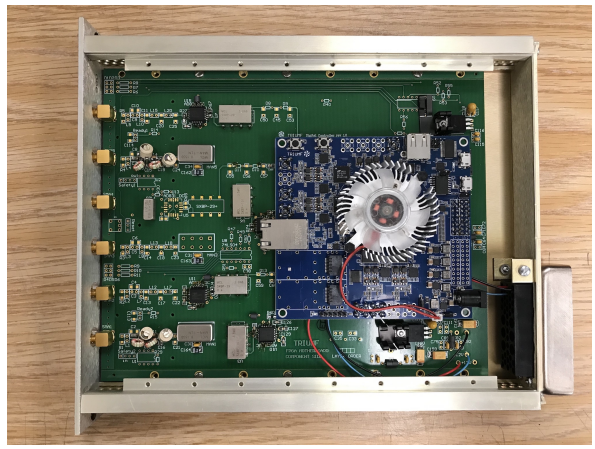

Figure 1: Picture of Digital LLRF system for DSB.

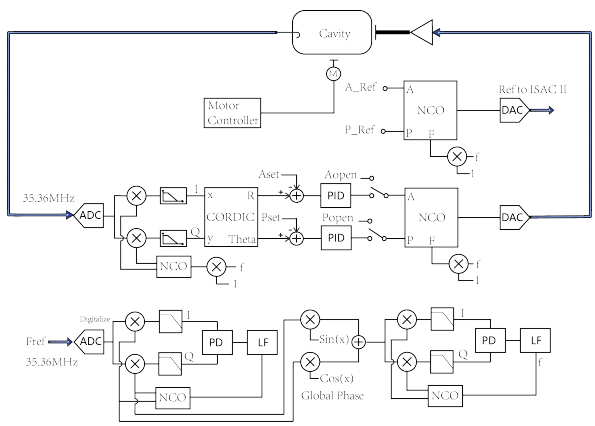


Figure 2: Digital LLRF system design for DSB.

### *Phase-locked Loop and Global Phase Shifter*

The new digital LLRF system works in driven mode and the phase of the output signal is not phase-locked to the reference signal. To resolve this issue, a digital phase-locked loop is introduced to the system. The digital phase-locked loop is based on a Costas loop which is widely used in communication systems. The first Costas loop is used to lock the frequency of the NCO to the 35.36MHz reference signal. After the first digital phase-locked loop, a global phase shifter is used to shift the phase of the reference signal, then the phase-shifted signal is used as the reference of the second phase-locked loop. The output of the second phase-locked loop is used as the basic frequency correction value for the system. Then, the two outputs of the system can be locked to the phase-shifted reference signal by adjusting

* xfu@triumf.ca

ISBN: 978-3-95450-212-7 ISSN: 2673-5482 doi:10.18429/JACoW-CYCLOTRONS2022-MOP0015



the frequency tuning words of the NCOs. Assume that the reference signal is [3]:

$$x(t) = A\cos(\omega_c t) \tag{1}$$

And the output of the local NCO is:

$$\begin{cases} I_0 &= \cos(\omega_0 t + \phi(t)) \\ Q_0 &= \sin(\omega_0 t + \phi(t)) \end{cases} \tag{2}$$

The results of the quadrature multiplier after the low-pass filter is:

$$\begin{cases} I_o = A/2 \cdot \cos[(\omega_c - \omega_0)t - \phi(t)] \\ Q_o = A/2 \cdot \sin[(\omega_c - \omega_0)t - \phi(t)] \end{cases} \tag{3}$$

After the phase detector, which is also a multiplier, the result is:

$$P_e(t) = A/8\sin(2(\omega_c - \omega_0)t - 2\phi(t)) \tag{4}$$

If we define:

$$\Delta\theta(t) = (\omega_c - \omega_0)t - \phi(t) \tag{5}$$

Then Eq. (4) can be written as:

$$P_e(t) = A/8\sin(2\Delta\theta(t)) \tag{6}$$

Based on Eq. (6), $P_e(t)$ is a function of $2\Delta\theta(t)$. Therefore, the frequency of the local NCO is controlled by the frequency and phase error between the reference signal and the local NCO. While in phase lock mode, $\omega_c = \omega_0$, the phase error and frequency error between the two signals are zero.

After the first phase-locked loop, an I/Q phase shifter is used to phase-shift the output of the NCO of the first phase-locked loop. The phase-shifted signal is used as the input of the second phase-locked loop. Thus, the global phase can be changed easily by the ARM CPU. Further more, the frequency of the output signal may be an integral multiple of the frequency of the reference signal, and the ratio is also controlled by the ARM CPU.

### *Amplitude and Phase Control*

For a single frequency system, assume the the cavity pick up signal is [1, 2, 4, 5]:

$$u_0(t) = A_0[1 + f(t)]\cos[\omega t + \phi'(t)] \tag{7}$$

where $f(t)$ is the amplitude modulation signal of the cavity. The output signal of tuning NCO is:

$$\begin{cases} I = \cos(\omega_1 t) \\ Q = \sin(\omega_1 t) \end{cases} \tag{8}$$

After the low pass filters, the mixing results of cavity signal and NCO signal are:

$$\begin{cases} I_2 = A_0[1 + f(t)]/2 \cdot \cos[(\omega - \omega_1)t + \phi'(t)] \\ Q_2 = A_0[1 + f(t)]/2 \cdot \sin[(\omega - \omega_1)t + \phi'(t)] \end{cases} \tag{9}$$

The amplitude can be calculated by:

$$U_2(t) = \sqrt{I_2^2(t) + Q_2^2(t)} = \frac{A_{0[1+f(t)]}}{2} \tag{10}$$

The phase of the cavity pickup signal is:

$$\Delta\theta = \phi'(t) = \arctan\frac{Q_2}{I_2} \tag{11}$$

Eq. (11) indicates that the phase error is independent from amplitude, frequency, and their modulation. Therefore, the phase control won't be effect by the phase-locked loop. After the amplitude error and phase error is obtained from the demodulator, a close loop amplitude and phase control is achieved by the PID controller. Since the NCO IP core doesn't have the amplitude modulation, the amplitude modulation is done by the multiplier whose inputs are the output of the amplitude PID controller and the NCO.

### *Tuning Loop*

The step motor controller is implemented in the FPGA and controlled by the ARM CPU. The ARM CPU can control the frequency, duty factor, and the operation mode of the motor controller. To work with different kinds of motor drivers, the polarities of the limit pins and enable pins can be programmed as well. An up/down counter is also included in the design to count the pulse number of each direction and the summation can be read by the CPU, as shown in Fig. 3. This summation can be used as the backup value of the motor position if the potentiometer is not available. There are three kinds of operation modes in the motor controller: manual control mode, single side hold mode, and auto reverse mode. In the manual mode, the controller ignore the limit signal and do what the users want. In the single side hold mode, if the motor hits the limit, the motor can't move towards this direction anymore, but it can still move towards the opposite direction. In the auto reverse mode, if the motor hits the limit switch, the motor will automatically switch to the other direction and keep moving. The LLRF system adopts the single side hold mode. Since the system has only one pickup signal from the cavity, the traditional phase comparison tuning method is not a good option for this system. The local control PC reads the phase of the LLRF output and the phase of the cavity pickup signal and then controls the motor through the motor controller. To achieve the best tuning condition, an open loop calibration has to be done before the system runs in closed loop mode. The user must run the system in open loop mode, and then manually tune the system. When the system is in perfect tuning condition, the user has to drag the sliding bar on the GUI to set the phase compensation value, shown in Fig. 4. This phase compensation value is the phase delay of the whole amplifier chain. Then the PC software can calculate the detuning angle correctly and maintain the tuning condition. When the phase delay of the amplifier is changed due to temperature changing or power level changing, the phase compensation value has to be changed as well.

ISBN: 978-3-95450-212-7 ISSN: 2673-5482 doi:10.18429/JACoW-CYCLOTRONS2022-MOP0015

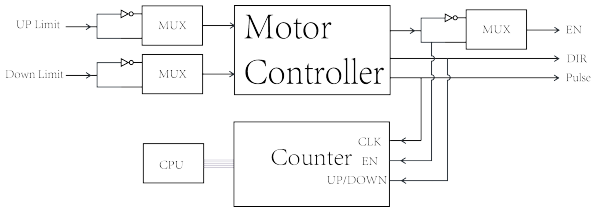


Figure 3: Motor controller for DSB LLRF system.

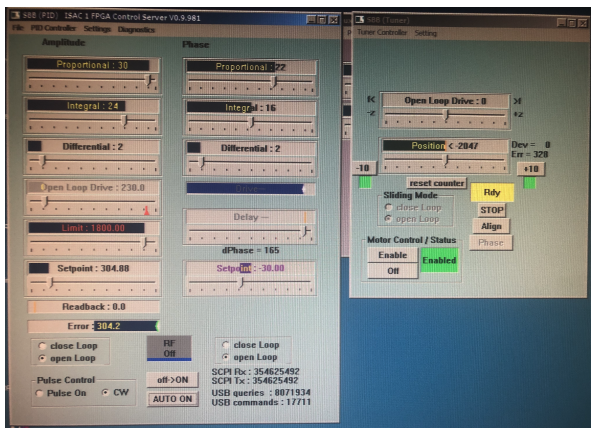

Figure 4: Local PC GUI for DSB LLRF system.

### *Linux OS and LLRF Apps*

The hardware system is equipped with large DDR3 RAM, Ethernet, and USB. In the previous TRIUMF LLRF system, the system is running with standalone code. Although the standalone code works well, it's hard to maintain and it's not user friendly since the programmer has to know all the details about the hardware system. A better solution is to use embedded Linux OS to manage all the hardware resources. The digital LLRF system adopts the Debian 11 as the Linux file system. The source code of Debian 11 is compiled and customized for the LLRF system. Python3 is chosen as the default Python environment. GCC is also available for users to compile source code locally. By default the user for the system is llrf and the root user is not enabled for ssh connections. Every operation is able to be done by the llrf account. The ARM CPU controls the FPGA hardware by reading and writing through the AXI-GPIOs. The Linux char device drivers are developed for the AXI-GPIOs. The device drivers are compiled into the Linux kernel and loaded when the OS is booting. The traditional read and write functions are not working well with Python code because of the buffer mechanism in Python. The ioctl driver is used for AXI-GPIOs writing and reading. Since the device driver is also very closed to the hardware level, it is not very convenient to call the device driver directly. A Python library is used to wrap the details of the hardware to make the system more user friendly. This Python library is called `pyllrf`, and by default it is installed in the system. Users can control the LLRF system without knowing anything about the FPGA firmware with the help from `pyllrf`. The hardware system can be initialized with four lines of Python code.

With the help from Python, the Jupyter notebook is also available. The new digital LLRF system runs a Jupyter notebook server by default and users can access to it through the internet. The port of jupyter notebook is set to 8888 by default. To access the jupyter notebook, users need to know the ip address of the LLRF system. With any web broswer, users can control LLRF system with python code. The popular Python libraries such as Numpy, scipy, pandas, matplotlib, and skit-learn are also avalible for the users. Jupyter bootebook also makes it possible to upload files to the filesystem and copy them to the FAT partition and upgrade the FPGA hardware design online without power off the system. The Linux HID device driver is implemented with the HID gadget feature of Linux kernel. Then in the Python code, the HID device information can be decided dynamically when the system is booting. The digital LLRF system is configured as an USB compliant device. When the system is plugged into a windows PC, the PC will automatically load the HID device driver and recognize the system as an USB HID device and an USB serial port. The system adopts 48 bytes length customized HID commands. The first two bytes of the commands determine the channel of the system to access, and the third byte determines if this a write or read operation. The rest bytes are for the parameters. The read/write command can read/write multiple parameters with one command to speed up the reading/writing. The local PC can control everything of the system through the HID commands.

### *Startup Procedure*

The system startup procedure is controlled by the local PC. The local PC accepts remote control commands from the control room through EPICS. After receiving the auto-on command, the system will start to work in pulse mode. The tuner will be moved to the open loop set-point. If the amplitude read back is bigger than the threshold, the local PC will switch the system into CW mode and close the tuning loop. Then the system enters the power ramping state to raise the power level. In the end, the amplitude and phase control loop will be closed and the startup procedure is done.

## TEST AND TUNE

The new digital system for DSB is installed on the control station and tested online. First of all, the motor controller is tested with the tuner. With the new system, the tuner will stop when hitting the limit switch, and then the limit indicator will be turned on on the local control GUI. The tuner can be moved out of the limit position by the GUI to the other direction. After the motor function test, the RF part is tested. The auto-on function works well and the system can finish the startup procedure in less than 2 minutes. The first test achieved 5kW power on the amplifier and the system is stable for one night. The next day, 10kW RF power was achieved on the amplifier and the system is still stable. However, the ISAC II LLRF system couldn't work well with the new reference signal from the new digital

ISBN: 978-3-95450-212-7 ISSN: 2673-5482 doi:10.18429/JACoW-CYCLOTRONS2022-MOP0015

LLRf system. The phase noise of the reference signal is measured and the result shows that there is a phase noise peak at 1.2kHz. Although this phase noise is out of the cavity bandwidth and it's under -70dBm, somehow it causes the ISAC II LLRF system to be unstable and the self-excited loop can't be locked.

The phase noise of the previous reference signal is quite clean and the new reference signal has a noise peak at 1.2kHz. This is the only difference between the two reference signals. In order to offer a better performance reference signal, the parameters of the digital PLL is tuned to reduce the phase noise in the low frequency range. After the proper adjustment, the phase noise is measured again, and the result is shown in Fig. 5. The green line in Fig. 5 is the phase noise of the reference signal and the blue line is the phase noise of the LLRF output after the parameter tuning. The test result shows that the phase noise peak at 1.2kHz is removed. With the tuned reference signal, ISAC II LLRF system can work properly, and the self-excited loop can be locked without any problem.

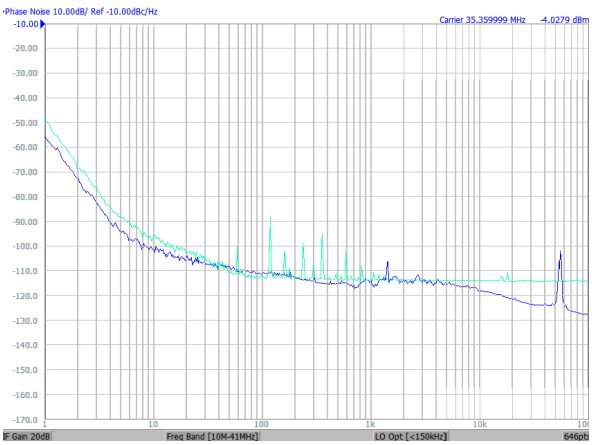


Figure 5: Phase noise of LLRF after tuning.

## CONCLUSION

An NIM bin based digital LLRF system is developed for TRIUMF DSB project. The hardware is based on TRIUMF's universal LLRF hardware and the software adopts Debian 11 as the Linux operating system. The communication between LLRF system and the local control PC is based on USB. A customized USB HID command set is designed for TRIUMF's new digital LLRF systems. A python library is written to control all the hardware of the system and communicate with the PC through USB HID commands. The online test shows that the requirement of the low-level RF control system has been satisfied.The new digital system is installed on the DSB system and the previous LLRf system is replaced cucessfuly.

## REFERENCES

[1] Xiaoliang Fu, Ken Fong, Zhiguo Yin, Qiwen Zheng, Thomas Au, "A digital phase-locked loop based LLRF system", *Nucl. Instrum. Methods Sect. A,* vol. 962, p. 163688, 2020. 10.1016/j.nima.2020.163688

[2] Xiaoliang Fu, Thomas Au, Ken Fong, Qiwen Zheng, "Sawtooth generation and regulation with a simple FPGA for TRIUMF'S ARIEL prebuncher", in *Proc. 10th Int. Particle Accelerator Conf. (IPAC'19)*, Melbourne, Australia, May 2019, pp. 2801–2803. doi:10.18429/JACoW-IPAC2019-WEPRB004

[3] Yong Du, "Digital filters based on MATLAB and FPGA", Publishing house of Electronics Industry, Beijing, 2014, pp. 320–334.

[4] Xiaoliang Fu, Zhiguo Yin, Ken Fong, *et al.*, "LLRF controller for high current cyclotron based BNCT system", *IEEE Trans. Nucl. Sci.*, vol. 68, no.10, pp. 2452–2458, 2021. doi:10.1109/TNS.2021.3097702

[5] Xiaoliang Fu, Zhiguo Yin, Ken Fong, *et al.*. "Design and implementation of general hardware and firmware for cyclotron low level radio frequency system", *At. Energy Sci. Tech.*, vol. 55, pp. 1886–1892, 2021.