www.czasopisma.pan.pl PAN www.journals.pan.pl

jet INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2022, VOL. 68, NO. 1, PP. 123-128
Manuscript received June 16, 2021; revised January, 2022. DOI: 10.24425/ijet.2022.139859

Urukul – Open-source Frequency Synthesizer Module for Quantum Physics

Grzegorz Kasprowicz, Thomas Harty, Sébastien Bourdeauducq, Robert Jördens, David Allcock, David Nadlinger, Joseph W. Britton, Ana Sotirova, Dorota Nowicka

***Abstract*—We describe Urukul, a frequency synthesizer based on direct digital synthesis (DDS), optimized for wave generate control in atomic, molecular and optical (AMO) physics experiments. The Urukul module is a part of the Sinara family of modular, open-source hardware designed for the ARTIQ quantum operating system. The Urukul has 4-channel, sub-Hz frequency resolution, controlled phase steps and accurate output amplitude control. The module is available in two population variants. This paper presents Urukul module construction and obtained characteristics.**

***Keywords*—frequency synthesizer; FPGA; ion trap; Sinara; quantum instrumentation; cold atoms experiments**

I. Introduction

MODERN physical experiments require specialized, very precise electronic equipment. Scientists are often forced to use an improvised solution, self-made electronic, makeshift, impossible to recreate, which involves duplicating the workload each time. Moreover, existing systems are also unsuitable for some experiments, as exemplified by frequency generators in the microwave regime, which typically lacks an independent adjustment of the phase or suffers from slow update speeds in the millisecond range.

Sinara and ARTIQ[1] projects in response to this problem have created a hardware and software system for quantum science laboratories, which is both open-source and commercially available[2][3]. As part of the Sinara project, over 50 commercially available modules were created. Frequency synthesizer Urukul module[4] is, apart from the Kasli FPGA controller[5], one of the most used building blocks of many control systems built with the Sinara ecosystem. An example control system built with Sinara hardware is shown in Fig. **1**.

Fig. 1. Example Sinara hardware configuration

This project is supported by the Polish National Agency for Academic Exchange - NAWA.

Dorota Nowicka (corresp.auth., dorota.nowicka@cern.ch) and Grzegorz Kasprowicz are with Warsaw University of Technology; Thomas Harty, Ana Sotirova and David Nadlinger are with Oxford University; Sébastien Bourdeauducq is with M-Labs; Robert Jördens is with QUARTIQ; David Allcock is with the Univeristy of Oregon; Joseph W. Britton is with University of Maryland and Army Research Lab.

II. Urukul Module

The Urukul is a 4-channel, DDS-based frequency synthesizer for the Eurocard form factor. It provides sub-Hz frequency resolution, controlled phase steps and accurate output amplitude control.

The Urukul module is available in two population variants with either the AD9912 chip with a 48-bit FTW (frequency tuning word) or the AD9910 chip with a lower (32-bit) FTW, but with 14-bit digital amplitude (ASF) resolution to allow servicing of laser amplitude.

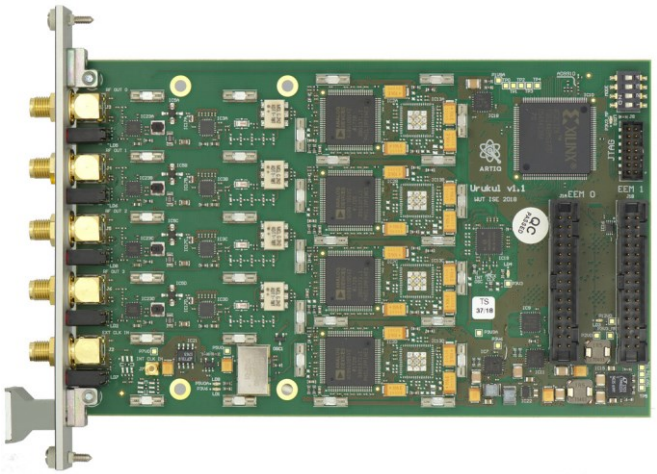

Fig. 2. Urukul module variant AD9910

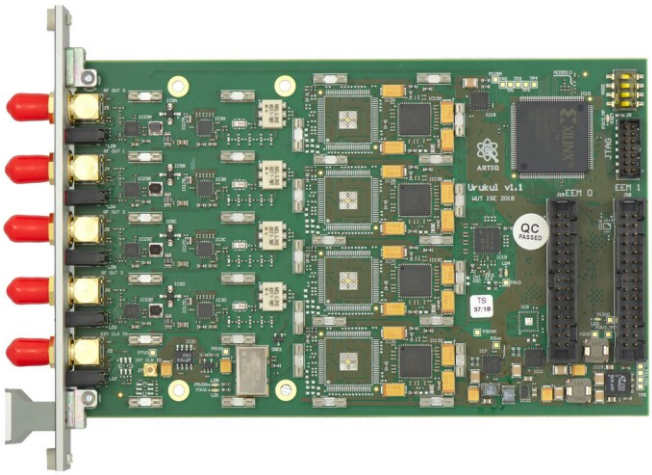

Fig. 3. Urukul module variant AD9912

Complete project documentation, including the source schematics, PCB layout, as well as production files is available from the Github repository[6]. Project sources are published under the CERN OHL license.

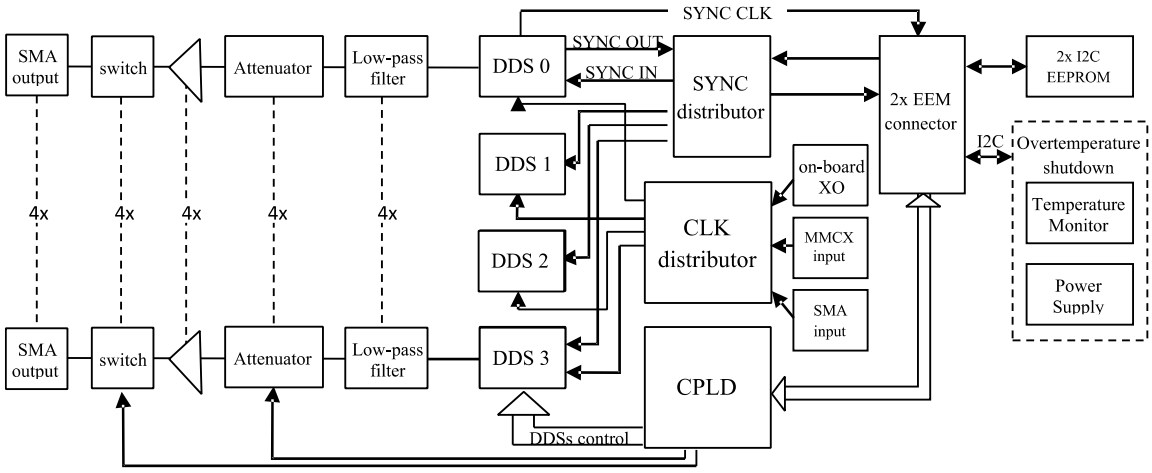

Fig. 4. Urukul block diagram

III. Block schematic and design considerations

The Urukul module block schematic is presented in Fig. 4. The main part of the Urukul module is an advanced, proprietary DDS technology digital-analog converter (DAC) which forms a digitally programmable, high frequency, analog output synthesizer capable of generating a frequency-agile sinusoidal waveform at frequencies up to 400 MHz. The user has access to the three signal control parameters that control the DDS: frequency, phase and amplitude. The AD9910 variant includes an integrated static RAM to support various combinations of frequency, phase, and/or amplitude modulation. The module's DDS specifications are listed in Table I.

TABLE I
DDS DETAILS AND SPECIFICATION

Parameter	AD9910	AD9912
Output frequency (-3 dB)	1 Hz – 400 MHz	
Frequency resolution	0.25 mHz (32 bit)	8 µHz (47 bit)
Phase offset resolution	16 bit	14 bit
Digital amplitude (ASF) resolution	14 bit	-
Power consumption	7 W	6.5 W

The AD9910 design variant suits the most common use cases as driving AOMs, which rarely come with more than 400MHz drive frequencies, without adding unnecessary cost and power consumption. The AD9912 variant is selected for applications requiring a larger FTW, such as optical clocks. These are modules with additional options, incl. signal filter. Then the origin of the output signal produced by the AD9910 and AD9912 is the combined DDS and DAC. The DAC output spectrum is shaped by the characteristic sin(x)/x (or sinc) envelope, due to the intrinsic zero-order hold effect associated with DAC generated signals. The sinc envelope can be compensated by the restoration function provided by the inverse sinc filter preceding the DAC. The inverse sinc filter is implemented as a digital FIR filter. It has a response characteristic that very nearly matches the inverse of the sinc envelope. The DAC response with the inverse sinc filter applied is shown in Fig. 5.[7]

For all four DDSs of the module to be synchronized with each other, a reference signal is distributed amongst them. It can come from the outside with the MMCX / SMA cable or from the first DDS of the module. Selection between MMCX and XO is made with onboard jumpers and between MMCX/XO and SMA - with software. The reference signal is multiplied by fanout IC channels and fed through aligned paths to the DDSs as an input signal of the phase-locked loop (PLL).

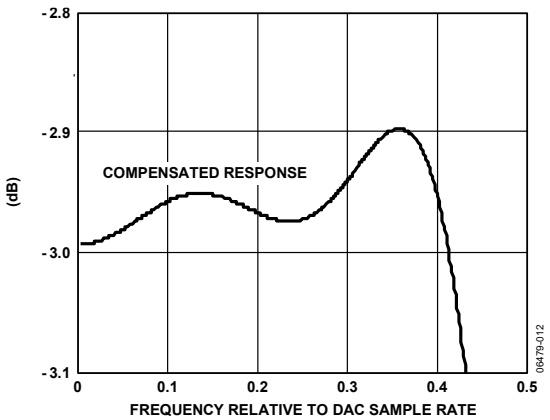

Fig. 5. DAC Response with Inverse Sinc Compensation

TABLE II
RF SIGNAL CHAIN

Parameter	Value
Digital step attenuator resolution	0.5 dB
Digital step attenuator range	0 to -31.5 dB
Digital step attenuator glitch duration	100 ns
Nominal max output power	10 dBm
RF switch speed	100 ns rise to 90 %
RF switch isolation	70 dB
Jitter RF switch	< 1 ns
Crosstalk (adjacent channels)	-84 dB with victim RF switch open -110 dB with victim switch closed
Cross-channel-intermodulation	-90 dB
Harmonics	at 6 dBm: -40 dB, -54 dB; at 10.5 dBm: -34 dB, -28 dB

TABLE III

CLOCK INPUT

Parameter	Value
Input frequency	10 MHz – 1 GHz
Input power	10 dBm
Sources	front panel SMA, internal MMCX (from Kasli),on-board XO
On-board XO frequency	
Selection between MMCX and XO	on-board jumpers
Selection between MMCX/XO and SMA	software

The power supply (Fig. 6.) includes overheating protection. The Urukul operates over an industrial temperature range, spanning −40 °C to + 85 °C. Its design requires airflow above 50 cm /s.

The front panel includes four SMA RF output, one SMA for the reference frequency input up to 1 GHz, and indicator LEDs.

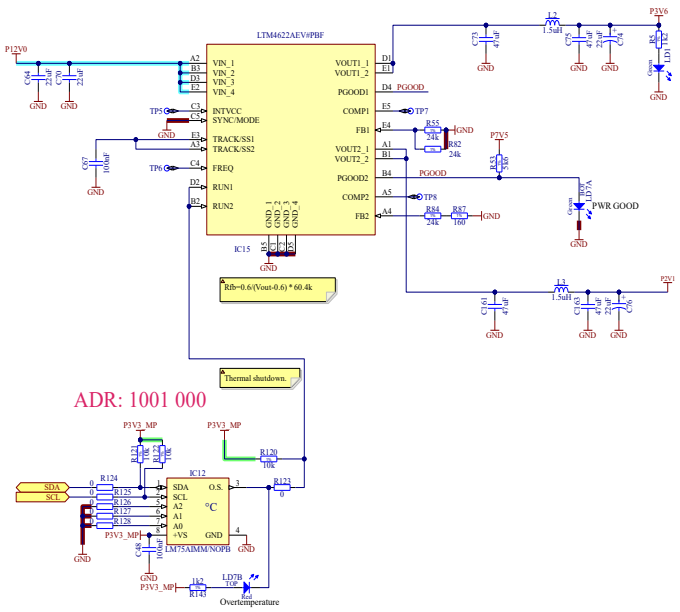

Fig. 6. Power supply with overheating protection

The complex programmable logic device (CPLD) on Urukul controls the SPI buses, multiplexes the different chips, and implements the shift register for control of miscellaneous functionality. The firmware of Urukul is available at Github.[8]

Urukul operates from one or two EEM connectors (Fig. 7). In standard SPI mode, the complete Urukul functionality can be accessed using only one. Then the connection of the second one allows for additional options, e.g. to interface with high-resolution RF switching and synchronization signals.

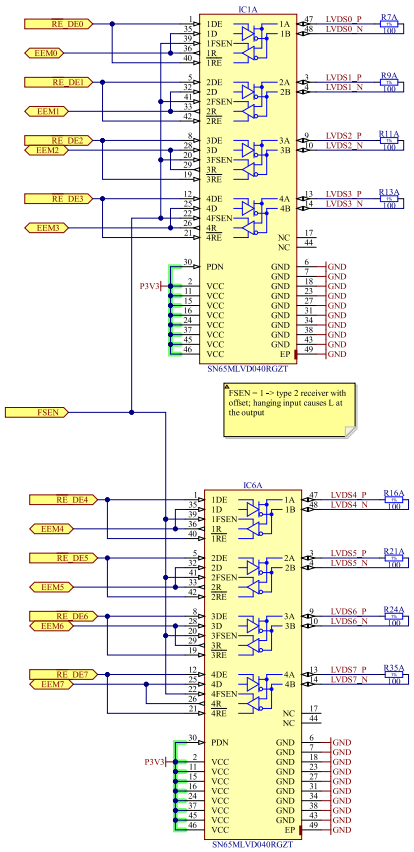

Fig. 7. EEM connector

The Urukul module is connected to an FPGA carrier using the 8 or 16 - channel LVDS interface. To properly identify the module version and interface configuration (single or double EEM), each EEM interface has a dedicated I2C interface connected to an EEPROM with a unique ID. The EEPROM holds a record with module identification string, HW version, manufacturer, date, etc. Such an approach eases firmware development since the software driver can check if it is compatible with the installed hardware module.

IV. SYNCHRONIZATION

Multiple devices are synchronized when their clock states match and they transition between states simultaneously. Clock synchronization allows the user to asynchronously program multiple devices but synchronously activate the programming by applying a coincident I/O update to all devices. The Urukul with AD9910 can be synchronized in three ways: DDS with each other on the same card, with different Urukul boards and with an external signal from FGPA. Clock distribution and generation scheme (version for AD9910 population) are shown in Fig. 8.

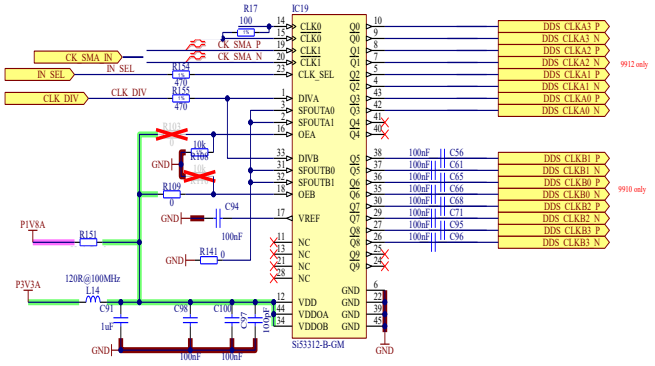

Fig. 8. Clock distribution and generation

Using the synchronization circuitry, a marker signal (SYNC) is distributed to all DDS that resets their synchronization signal dividers. The synchronization logic is divided into two independent blocks: a sync generator and a sync receiver, both of which use the local system-clock signal for internal timing. The sync generator block allows for one DDS in a group to function as a master timing source with the remaining devices slaved to the master.

Due to the sync generator, it is possible to synchronize Urukul modules connected with different FPGAs (e.g. Kasli carrier) as in the example circuit in Fig. 9.[7] The connection between FPGA controllers is then provided by DRTIO.

The control parameters change signal from the FGPA needs to be reproducibly adjusted so that the changes are applied at the right time moment. This is done using the DDS-internal delays seeded with a persistently stored initial value. Given this synchronization, the proper timing IO update can be inferred. These delay values are stored in EEPROM connected with I2C, so calibration of them should only be required once for a given hardware design.

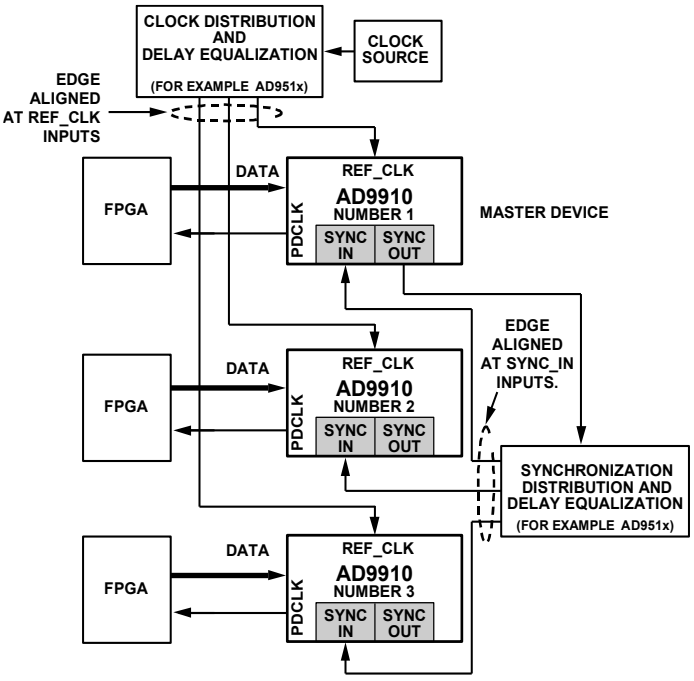

Fig. 9. Multiple devices synchronization

V. Design Verification

Due to the complexity and multi-layer structure, the critical blocks of the module was verified before drawing the schematics with SPICE simulations. Since one of the most important requirements for Urukul is precise signal shaping, the key is the output stage and clock distribution design.

After PCB layout and routing, the design was validated in terms of Signal Integrity and Power Integrity using the Mentor Graphics Hyperlynx tool. The modules were also thoroughly post-production tested, measuring bandwidth, phase noise, and channel cross-talk. The Urukul was also tested and is currently working in numerous experiments, mainly related to ion traps (see section X).

VI. Tests and Characterization

One of the most important tests of the Urukul module was phase noise measurement.[9] The setup consisted of a very clean 100 MHz reference input (10 dBm) connected with SMA. The Urukul was connected to a 12 V lab power supply (not to Kasli carrier as typically) to avoid noise from the FPGA board. For measurements channel 1 was used. The characteristic is presented in

Fig. 10, the measurements are presented in Table IV.

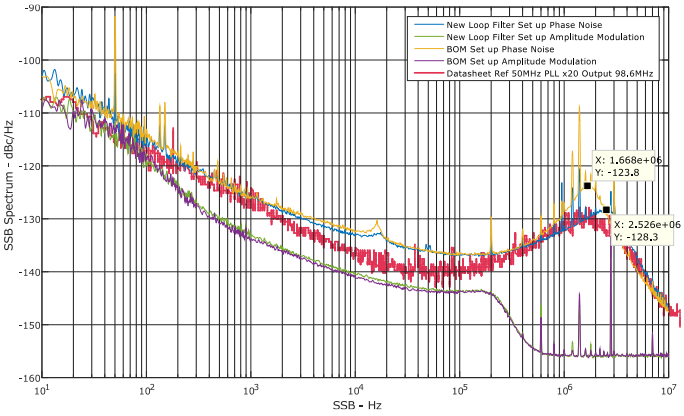

Fig. 10. Phase noise characteristic

Other parameters of the Urukul module were tested with a similar setup, but using also other channels. In particular, cross-channel intermodulation and channel cross-talk were measured. The results are presented in Table V and Table VI.

TABLE IV
PHASE NOISE MEASUREMENT RESULTS

Frequency	Phase noise
0.1 Hz	-85 dBc/Hz
1 Hz	-95 dBc/Hz
10 Hz	-107 dBc/Hz
100 Hz	-116 dBc/Hz
1 kHz	-126 dBc/Hz
10 kHz	-133 dBc/Hz
100 kHz	-135 dBc/Hz
1 MHz	-128 dBc/Hz
10 MHz	-149 dBc/Hz

TABLE V

CROSS-CHANNEL INTERMODULATION

Intermodulation	Aggressor	Victim	IMD product
-84 dB	1 MHz, 10 dBm	80.2 MHz	At 81.2 MHz
-90 dB	1.234 MHz, 10 dBm	80.123 MHz	At 81.357 MHz
< -100 dB	80.123 MHz, 10 dBm	81.234 MHz	At 82.345 MHz

TABLE VI
OTHER TESTS

Parameter	Value
Low frequency limit	100 kHz out: <-20dBm 1 MHz out: -10 dBm
Switch isolation	70 dB (attenuator at 63 digital)
Cross-talk	Victim switch open: -84 dB Victim switch closed: -110 dB

VII. SOFTWARE

The ARTIQ (Advanced Real-Time Infrastructure for Quantum physics) is a leading-edge control system for high-advanced physics experiments. It is compiled and executed on dedicated FPGA hardware (for example Kasli controller) with nanosecond timing resolution and sub-microsecond latency. It allows describing the entire experiment environment in a high-level, Python-based programming language. The ARTIQ also provides a graphical user interface, an experiment scheduling system and databases for experiments, devices, parameters and results.

It is dedicated to controlling Urukul and other Sinara modules, creating a coherent ecosystem. The integration modules together are very simple and can be implemented by just a few lines of code. Thanks to the highly portable FPGA design, the software can adapt to various laboratory setups and is resistant to hardware obsolescence.

ARTIQ is now used and supported by a growing number of research institutions worldwide. While ARTIQ is currently mostly used by atomic physics groups, its applicability reaches beyond ion trapping. Example Python code of the laser intensity servo can be found on Github[10].

VIII. ARTIQ DISTRIBUTED REAL-TIME INPUT/OUTPUT

Distributed Real-Time Input / Output (DRTIO)[11] is a time and data transfer system that allows ARTIQ RTIO channels to be distributed among several satellite devices synchronized and controlled by a central master device.

The link is a high-speed duplex serial line operating at 1 Gbps over copper or optical fiber. Time transfer and clock recovery may be done over the serial link alone or assisted by auxiliary signals.

The main source of DRTIO traffic is the remote control of RTIO output and input channels. The protocol is optimized to maximize throughput and minimize latency, and handle flow control and error conditions (underflows, overflows, etc.)

The DRTIO protocol also supports auxiliary, low-priority and non-realtime traffic. The auxiliary channel support overriding and monitoring TTL I/Os. Auxiliary traffic never interrupts or delay the main traffic, so that it cannot cause unexpected poor performance deterioration (e.g. RTIO underflows).

The lower layers of DRTIO are similar to White Rabbit[12], with the main differences such as: lower, deterministic latency, real-time/auxiliary channels, higher bandwidth, guaranteed cut-through switching, automatic enumeration, no Ethernet compatibility, only star or tree topologies are supported.

IX. EXPERIMENTAL APPLICATIONS

Urukul was used in the University of Wisconsin-Madison experiment in transporting cold atoms and Bose-Einstein condensate in an optical tape, studying their behaviour in cases where a harmonic potential and 1D geometry are not assumed[13]. In this experiment Urukul drove two AOMs (acousto-optic modulators) sharing a common clock i.a. all experiment was controlled by ARTIQ.

Using Urukul, a dynamic low-noise microwave source was created for experiments with cold atoms.[14] The ultra-low-noise oscillator at 7 GHz and Urukul for the dynamic parameter change were combined to yield frequencies in the range of 6.835GHz ± 25MHz for hyperfine transitions in 87Rb.

Another example application of the Urukul module (the AD9910 variant) in the quantum experiment is the Sampler-Urukul servo show in Fig. 11. The setup is the multi-channel, multi-profile laser intensity servo, where an 8-channel ADC module (Sampler [15]) monitors the plant - an AOM drove via a power amplifier. In the circuit is included a photodiode for monitoring. The Urukul drives the plant and the FPGA controller (Kasli [16]) performs computation.

CONCLUSIONS

The most important distinguishing features of the Urukul module are:

- sub-Hz frequency resolution, controlled phase steps,
- accurate output amplitude control of the AD9910 version,
- two available variants for fine amplitude control and more-bits tuning word,
- being part of the Sinara system, whose modular design allows the system to be scaled up to thousands of channels thanks to the DRTIO protocol,
- very low latency due to dedicated communication protocol,
- open-source design, including hardware and software that makes it attractive to the user because it mitigates product life cycle and support issues, and gives insight to design decisions and motivations.

The entire Sinara system fits perfectly for specific applications in experimental quantum physics. Contrary to popular COTS products intended for general industrial use, it takes into account the specific needs of customers running AMO labs. The open-source nature of the project allows the end-user to define requirements and make suggestions at all stages of research and design, as well as production and testing.

It is supported by the ARTIQ operating system, allowing building hard-real-time applications in a simple, short Python-based code.

A key factor in the success of the project is the involvement of two commercial vendors supporting the same Sinara products. This enables the end-user to conveniently change the vendor without having to make any configuration modifications.

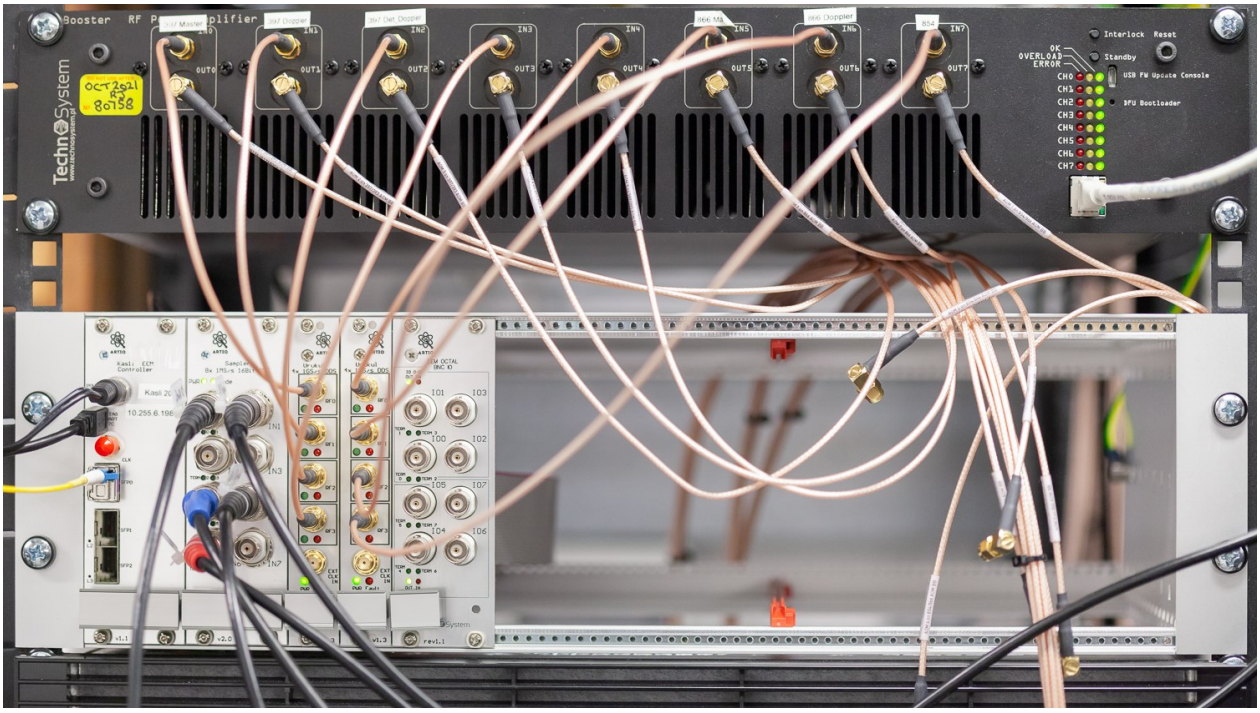

Fig. 11. Experimental setup with Urukul module as a part of multi-channel laser intensity servo (In crate from the left: Kasli[16], Sampler[15], two Urukuls, DIO BNC[17]. Above: Booster[18] - RF power amplifier)

REFERENCES

[1] "ARTIQ | M-Labs." https://m-labs.hk/experiment-control/artiq/ (accessed May 05, 2021).

[2] "Welcome to Sinara | ['Sinara']." https://sinara-hw.github.io/ (accessed May 05, 2021).

[3] S. Bourdeauducq, m-labs/artiq 3.1.

[4] "Home · sinara-hw/Urukul Wiki · GitHub." https://github.com/sinara-hw/Urukul/wiki (accessed May 05, 2021).

[5] P. Kulik, G. Kasprowicz, and M. Gąska, "Driver module for quantum computer experiments: Kasli," Photonics Appl. Astron., vol. 1080845, no. 1, 2018, https://doi.org/10.1117/12.2501709.

[6] "GitHub - sinara-hw/Urukul: 4 channel 1GS/s DDS (AD9910 or AD9912 variant)." https://github.com/sinara-hw/Urukul (accessed May 05, 2021).

[7] M. Kumm, "Urukul_AD9910 Manual," no. 2, pp. 1–2, 2008.

[8] "GitHub - quartiq/urukul: Urukul CPLD code." https://github.com/quartiq/urukul (accessed May 05, 2021).

[9] R. Jördens and G. Kasprowicz, "Urukul v1.0 detailed testing · Issue #3 · sinara-hw/Urukul." https://github.com/sinara-hw/Urukul/issues/3 (accessed May 05, 2021).

[10] "artiq/artiq/gateware/suservo at master · m-labs/artiq · GitHub." https://github.com/m-labs/artiq/tree/master/artiq/gateware/suservo (accessed May 05, 2021).

[11] S. Bourdeauducq, "DRTIO · m-labs/artiq Wiki." https://github.com/m-labs/artiq/wiki/DRTIO (accessed May 05, 2021).

[12] "Switch · Wiki · Projects / White Rabbit · Open Hardware Repository." https://www.ohwr.org/projects/white-rabbit/wiki/switch (accessed May 05, 2021).

[13] G. T. Hickman and M. Saffman, "Speed, retention loss, and motional heating of atoms in an optical conveyor belt," Phys. Rev. A, vol. 101, no. 6, 2020, https://doi.org/10.1103/PhysRevA.101.063411

[14] B. Meyer, A. Idel, F. Anders, J. Peise, and C. Klempt, "Dynamical low-noise microwave source for cold atom experiments," no. April, pp. 1–6, 2020, [Online]. Available: http://arxiv.org/abs/2003.10989 .

[15] "Home · sinara-hw/Sampler Wiki · GitHub." https://github.com/sinara-hw/Sampler/wiki (accessed May 05, 2021).

[16] "Home · sinara-hw/Kasli Wiki · GitHub." https://github.com/sinara-hw/Kasli/wiki (accessed May 05, 2021).

[17] "Home · sinara-hw/DIO_BNC Wiki · GitHub." https://github.com/sinara-hw/DIO_BNC/wiki (accessed May 05, 2021).

[18] "Home · sinara-hw/Booster Wiki · GitHub." https://github.com/sinara-hw/Booster/wiki (accessed May 05, 2021).