Low-cost Magnetic Resonance Console Architecture using an Open Source for Laboratory Scale Systems

Chennagiri Rajarao Padma, Ravikumar K. M.

***Abstract:** MRI systems with proprietary hardware must use pulse programming, which is less expensive. Pulse programming consoles utilise Digital Signal Processors, Complex Programming Logic Devices, and microcontrollers, which are typically restricted to specific architectures. General–purpose, extremely affordable electronics board featuring these architectures are now capable enough to be directly implemented in MRI consoles. Here we present the architectural details of various consoles with novel designs and their limitations. Finally, we propose a console design which was created utilising widely accessible Arduino Boards to connect to Pulseq-GPI implementations at a reduced cost of $225.*

***Keywords:** Direct Digital Synthesizer, Microcontrollers, MR Console Architectures, Pulseq-GPI*

I. INTRODUCTION

Magnetic Resonance Imaging (MRI) system consoles have different architectures depending on the type of data and logic processing device, its implementation and the software to run the system. MRI pulse sequences require an intelligent console that coordinates real-time operation of RF and gradient systems, data gathering, and processing. The use of proprietary software, hardware, and interfaces hinders growth and prevents adjustments to novel methods and experiments. A significant design challenge is posed by meeting contemporary requirements for a large number of channels, real-time imaging capability and customizable interfaces. While radio frequency sampling and synthesis are easily handled by Software-Defined Radio (SDR), effective data management and real-time control still pose significant challenges. Scalable modular system architecture is necessary for the practical and versatile implementation of such systems. Peripheral Component Interconnect (PCI) cards, which offer high-throughput, low-latency connectivity at the expense of scalability, cost efficiency, and design time.

Although performing tasks in real-time with software or a microcontroller may offer considerable flexibility, it places severe limits on programme design. It makes wasteful use of the available processing power and hardware resources.

II. ARCHITECTURES

Console architectures are classified based on the processing device and its hardware and software capabilities. Microcontrollers (µCs) are self-contained systems comprising a processor, a fixed amount of memory, and peripherals all integrated on a single chip. They have the capability for mixed-signal processing at reduced costs and programming complexity.

S. Handa et al., have demonstrated a single chip pulse programmer (PPG) using ADµC7026 microcontroller board,[1] which consists of a 32-bit reduced instruction set computer central processing unit (RISC CPU) core(ARM7TDMI) with a clock frequency of 41.78 MHz, 62 kilo bytes (kB) of programming (flash) memory and eight kB of static random access memory (SRAM), a 16-channel analog to digital converter (ADC)and four digital to analog converters (DACs), two timers, and a 40-bit digital input/output (I/O) port. The board was integrated with a personal computer (PC) via a universal serial bus (USB) to RS-232C converter circuit, and on the other end, with an MRI transceiver and gradient driver. The programs for the microcontroller and host PC were developed using the KEIL development suite and the GNU C compiler on a Linux emulation platform, respectively. The cost of the board is approximately U.S. $10, with limitations of relatively long delay times between events and event data memory. The work of M. Tsuda et al. interfaced a PC with three 32-bit ARM-based RISC µCs and digital oscilloscopes, featuring both inbuilt and external arbitrary waveform generators (AWGs).[2]. The µCs used were an Arduino DUE with a clock frequency of 84 MHz, two 12-bit DACs, 96 kB SRAM, and 54 I/O pins. The Pico Scope 3205B (250 MHz sampling speed) was used in one design, along with an external AWG-100 (100 MHz sampling speed). In another design, an oscilloscope with a built-in AWG (Pico Scope 5242B) (1 GHz sampling speed) was employed. The programs were developed using Qt 5.3.0 and C++ on the host PC, which downloads the target programs to the µCs using the Arduino Integrated Development Environment. The total cost of either console was less than $1,500. Although under-sampling and phase correction were required, the system provided flexibility in pulse sequence design.

Manuscript received on 31 December 2022 | Revised Manuscript received on 07 January 2023 | Manuscript Accepted on 15 January 2023 | Manuscript published on 30 January 2023.
*Correspondence Author(s)
Chennagiri Rajarao Padma*, Assistant Professor, Department of Medical Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru (Karnataka), India. E-mail: padma-mirc@dayanandasagar.edu, ORCID ID: https://orcid.org/0000-0001-5874-6823
Dr. K. M. Ravikumar, Director, The Oxford Education Institutions, Bengaluru (Karnataka), India. Email:kmravikumar75@gmail.com, directortoce@theoxford.edu, ORCID ID: https://orcid.org/0000-0002-3961-9340

An inexpensive, widely available Arduino Duemilanove board, based on the ATmega 328 µC (16 MHz clock frequency) and utilising the Arduino open-source programming platform, was used for MR spectroscopic measurements in Earth's magnetic field by Carl A. Michal.[3]The µC has 14 digital I/O pins, six analog input pins that acted as a pulse programmer, audio frequency synthesizer, ADC and could produce trains of phase coherent radio frequency pulses with reproducible timing allowing signal averaging and phase cycling. The µC was used as a state machine, and pre-calculated states and events were downloaded from the host PC, as the available memory was limited to run sequences in real time. The cost of the system is within $200 and is restricted to spectroscopic sequences.

M. Twieg et al. have demonstrated an open-source, fully packed and functional NMR relaxometry platform based on the AT32UC3C1512 µC from Atmel.[4] The 32-bit CPU operating at 64 MHz with 512 kB flash, 64kB SRAM, 16 ADC channels and 2 DACs allowed for more sophisticated and configurable pulse sequences, and faster handling of sequence events. The board was integrated with a PC running MATLAB and an AD9958 DDS module for RF signals, to define experiments and analyse sampled data. The cost of the chip is less than $ 750.

Meghan E. Halse's work on "Terranova" featured a Digital Signal Processor board at its core, providing pulse programming and signal acquisition functionality. This board utilised a USB Interface to communicate with the PC.[5] The interfacing software used was Prospa.

Field-Programmable Gate Arrays (FPGAs) are integrated circuits that contain an array of programmable logic blocks, which can be configured using a hardware description language to perform complex combinational functions. They are generally higher in cost and programming complexities compared to microcontrollers, but are more versatile.

A home-built MRI system with a digital spectrometer was demonstrated by S. Jie et al.[6] The console design consisted of XC2S200 (Xilinx) RF waveform generator, digital receiver, gradient waveform generator, and pulse generator boards are all logically controlled by an FPGA processor. The boards were interfaced with a PC using PLX9052 chips, serving as a bridge between the local bus and the PCI bus. Additionally, one or two 16-bit SRAMs of 256 kB were attached to each board for data storage. The AD9854 DDS chip, operating at a clock frequency of 50 MHz, was used for generating RF pulses. The receiver was based on the AD9874 chip for software radio technology, and the gradient board utilises the AD5542 DAC chip for each channel. Delphi 7.0, running on the host PC, was used to create the software winMRI, which controls hardware and displays gathered data in real-time. The system was implemented for $ 12000.

K. Takeda demonstrated that a single FPGA chip was sufficient to function as a PPG, DDS, digital receiver, and PC interface for command and data transfer.[7]. The system consists of an Altera EP2C70F672C8 (Cyclone II) FPGA chip, controlled by a 30 MHz crystal, and several auxiliary boards for USB connectivity, direct digital synthesis (DDS), RF transmission, signal acquisition, and other purposes. Each of the three RF channels in this spectrometer can modulate the amplitude, phase, and frequency of RF irradiation at frequencies up to 400 MHz. The other modules surround the motherboard. The source codes for the VHDL-written programmes for the core modules that the FPGA uses to execute them in parallel are accessible on the author's website.

Preeti Hemnani et al. have demonstrated in detail the setup of an FPGA as a Pulse programmer and a direct digital synthesiser.[8] Their architecture requires a DDS integrated with an external DAC to generate coherent and accurate Phase and frequency modulated pulses for very short intervals. The pulse programmer was designed as a finite-state machine to develop precise timing pulses with durations as short as one microsecond. LabVIEW was the application software for the FPGA, and communication with the PC was established through the PCI bus.

H-Y Chen et al. have successfully demonstrated the improved capabilities of the system using Cyclone II, Altera FPGA, operating at 50 MHz.[9] To do away with the requirement for an external DAC, they have integrated the digital frequency generation with an amplifier (MAX4384EUD+; Maxim Integrated, CA) and a bandpass filter. The custom logic blocks are used to run the NMR pulse programme. An FPGA-programmed direct memory access (DMA) controller is used to play back the pulse sequence. By adjusting the clock speed used to time the events of the pulse sequence, even the device's time resolution can be altered in this design. The onboard digitiser provides the data acquisition capabilities. The data is read from the memory and sent to a host computer over a USB port using the same processor.

A digital I/O board was used as a pulse programmer by Hashimoto et al., which features 32-bit input/output lines with a maximum transmit rate of 20 MHz, 32 MB on-board memory, 30 DAC outputs, and a mixer with a filter.[10] Cyclone III, ALTERA FPGA was configured as the I/O board, running at a clock frequency of 60 MHz. It communicates with the PC via USB. The interfacing software was developed using C/C++ and the .NET Framework 2.0 of Visual Studio 2008, running on Windows 7.

In a radio communication system known as Software Defined Radio (SDR), software on an embedded system is used to implement components that were previously implemented in hardware. ADCs, DACs, an FPGA for basic filtering and signal down- and up-conversion, and a USB interface are the typical components of SDRs. SDRs are inexpensive, easy to program and operate at relatively high frequencies. Two architectures based on SDR are discussed.

C. J. Hasselwander et al. assembled a console comprising two GNU Radio-compatible SDRs: one for RF excitation and reception, and the other for gradient pulse generation.[11] The two SDRs are synchronised through pulses generated by the master in the master-slave SDR configuration. The software systems were built using the Python programming language in GNU Radio, and the authors have also released a software package implemented for four different sequences on the same platform. The total SDR hardware cost was $2000.

A. Asfour et al. worked on a fully digital RF electronics design for NMR systems at low field.[12] The system consisted of a DDS AD9852, for pulse generation, a SDR based on an evaluation board of CLC5902 (National semiconductor) chip, for digital reception of NMR signals and a Digital Signal Processor (DSP) ADSP-2106x SHARC, for system control and the generation of the gradient signals (pulse programmer). The DDS is used for pulse generation, and the SDR serves as a receiver. The DSP controls the DDS and SDR via its parallel external bus and is also interfaced with a PC. The SDR was integrated as close to the receiving coil as possible to minimise noise and distortion associated with the analogue mixing stage. The software for the system was developed using the LabWindows/CVI environment and DSP assembly language.

Another architecture was designed using Complex Programmable Logic Devices (CPLDs), which have a complexity between that of programmable logic arrays and FPGAs, with architectural features of both.

Pascal P. Stang et al. demonstrated a scalable console called "MEDUSA" that utilises programmable logic for frequency synthesising, sampling, and synchronisation.[13] It consisted of 16 modules over a 16-bit parallel data link with a dedicated logic core and system controller. The hardware included an Altera MAX-II EPM1270 CPLD logic core with identification registers, a Cypress CY7C1061 2Mbyte high-speed SRAM, and a DMA. The logic core was responsible for timing, sampling, and data flow. The LPC2214 60 MHz 32-bit ARM7 system controller was responsible for coordinating different modules. Matlab was used for system programming. The cores of the CPLD were developed using Verilog HDL. USB High-Speed (480Mbit/sec) support is implemented using a Cypress Semiconductor CY7C68013A FX2 USB peripheral interface.

Rapid sequence prototyping is supported by the open-source and hardware-independent programming environment Pulseq.

Layton, Kelvin J et al., propose a new file format that may be used to store hardware events and timing data for MR pulse sequences in an effective manner [14]. The file is converted to the proper instructions for running the sequence on MR hardware using platform-specific interpreter modules. Sequences can be created using a graphical interface or high-level languages like MATLAB.

Keerthi Sravan R. et al., To access Pulseq, the author demonstrated an open-source implementation of Pulseq in the GPI Lab [15,16]. Additionally, it enables the integration of pulse sequence design with other components of the MR research pipeline—simulation, reconstruction, image analysis, and visualisation—on a single platform using GPI. Through a gradient-echo remembered sequence, the Pulseq-GPI implementation is demonstrated to be capable of generating all the sequences that Pulseq currently offers.

Commercially available consoles, such as Portable Lab, are bench-top MRI scanners with hands-on examples optimised for both scientific and educational use. It is a tool for developing and testing of MR hardware and MR sequences on a desktop with six Digital I/Os3MTM Mini Delta Ribbon (MDR); 20 pins (10220-55G3PC) and open MATLAB interface [17] whereas **MEDUSA,** an open system, combines distributed processing and buffering with scalability provided by the Universal Serial Bus. Fast programmable logic is used in Medusa's modular design for sampling and synchronization.[13]. Another popular MRI system is **Terranova- MRI: Earth's field MRI teaching system [18]**, which is straightforward, portable, and user-friendly. It provides a fantastic starting point for education for new researchers in the field of NMR and was created specifically to study a variety of contemporary pulsed FT NMR and MRI experiments and concepts.

III. IMPLEMENTATION

The reported pulse programming console implementations utilise DSPs, complex programmable logic devices (CPLDs), and microcontrollers, which are often limited to specific architectures and require more time and resources to develop. As a result, at a reduced cost of $225, MR consoles for laboratory-scale systems on an open-source platform were constructed for 9.5mT using accessible Arduino Boards to connect with Pulseq-GPI implementations. Data extracted from the Pulseq-GPI, stored, and uploaded as a text file, will support the microcontroller in playing a 5-second dwell period RF pulse and gradient waveforms (GX and Gy). Currently, efforts are underway to integrate an analogue-to-digital converter for the Gradient Recalled Echo (GRE) sequence and accelerate the uploading of the entire sequence's waveforms.

A. Methods:

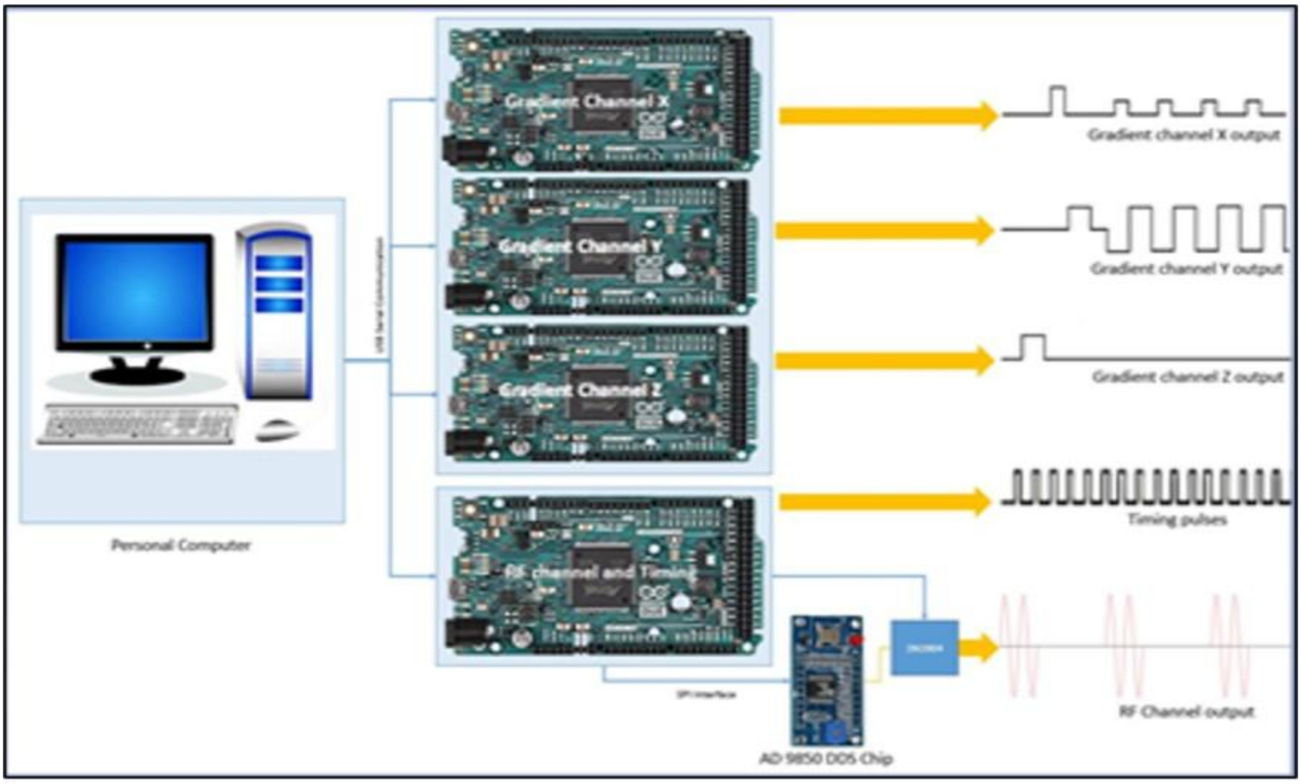
1. Hardware description: As illustrated in figure 1a, the console is made up of 32-bit general-purpose Arduino DUE boards, for each gradient channel Gx and Gy, for timing and regulating radio frequency (RF) pulses, and one board that can be readily expanded to include the third gradient channel (Gz). The AD9850 Direct Digital Synthesiser chip was connected to the RF-timer board and programmed to produce a 405 kHz RF sine wave, the output of which was sent to the collector of a 2N3904 transistor. Rectangular RF pulses were produced by switching the transistor's output with RF-timer pulses. The gradient waveforms were played out on the internal Digital-to-Analogue Converter outputs of the gradient boards, Gx and Gy. To upload waveform data, each board is connected to a personal computer via the USB serial communication protocol.

2. Software implementation:

As seen in Figure 1b, the Pulseq-GPI-based implementation creates a .seq file that has been decoded into three distinct gradient files and one RF pulse file. These files were then uploaded to each of the corresponding boards using Python, which was running on a PC.

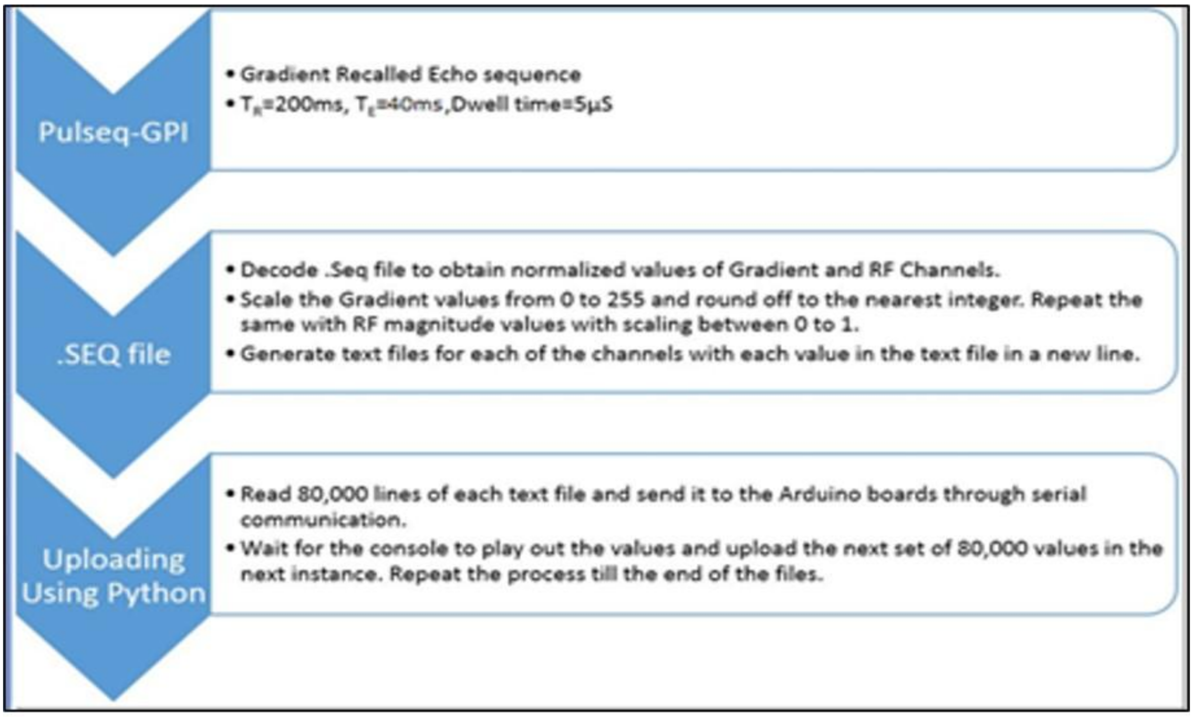
Retrieval Number: 100.1/ijitee.B94130112223
DOI: 10.35940/ijitee.B9413.0112223
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)



(a)

Figure 1 (a) This shows X, Y, Z gradient channel boards with outputs of each channel and RF channel. A serial peripheral interface (SPI) is connected to the AD 9850 DDS chip. This chip is controlled by the RF and timing Arduino board.



(b)

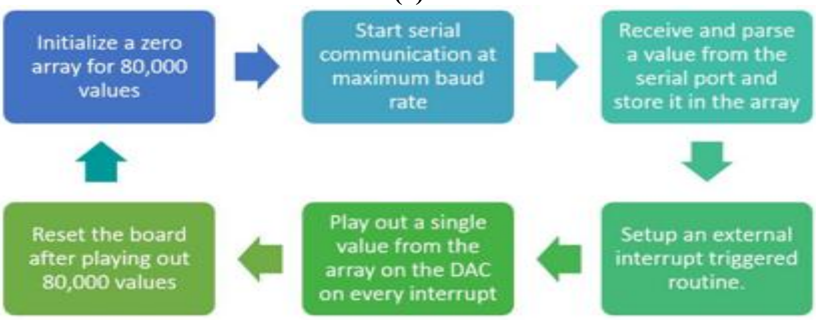
Figure 1 (b): Flow chart shows the events from Pulseq- GPI to generation of RF and Gradient waveforms.

Retrieval Number: 100.1/ijitee.B94130112223
DOI: 10.35940/ijitee.B9413.0112223
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

On one occasion, 80,000 values, or data for two TR, were transferred to the boards and played out, taking around 11.2 seconds. Using scheduled interrupts from the DUE board, the timer produced precise synchronization pulses every 5 s for each instance. The transistor switch is controlled by the RF timer board based on the stored magnitudes of the RF pulse, as illustrated in Figure 2a. The gradient boards are interrupt-driven and generate voltage values on their DACs in response to externally triggered interrupts, as shown in Figure 2b.

(a)



(b)

Figure 2: (a) shows the program flow diagrams for the Timer-RF of the channel from the Arduino DUE board, and (b) shows the program flow diagram of the Gradient Channel Board.

The system consists of three boards and can play any two gradient channels together with an RF channel. This may be extended to include a fourth board for a third gradient channel. Figure lists the price of the console being created.

3. Cost of components:

Components	Cost	Source
Arduino Due	$38*5= $190	Arduino Store
AD 9850 DDS Chip	$15	EBay
2N3904 Transistor & wires	$5	Regional vendors
Total= $210 + (approx. Shipping costs) $15 = $225		

Figure 3: Arduino Due boards for three gradient channels, an ADC channel board, and a timer board for the RF channel are listed. They are sourced from regional vendors for $225.

B. Results:

The designed hardware is a cost-effective solution to generate three Gradient and RF Channel waveforms. Hardware implementation using Arduino reduces the complexity of the MR pipeline design and development processes. Figure 4a shows the plot of uploaded data, and Figures 4b and 4c show two TRs with a dwell time of 5 µs for 400 ms of RF channel and Gradient channel waveforms. The RF channel and Gz channel outputs are illustrated separately in Figures 4d and 4e. Similarly, Gx and Gy values are displayed in Figure 4f.

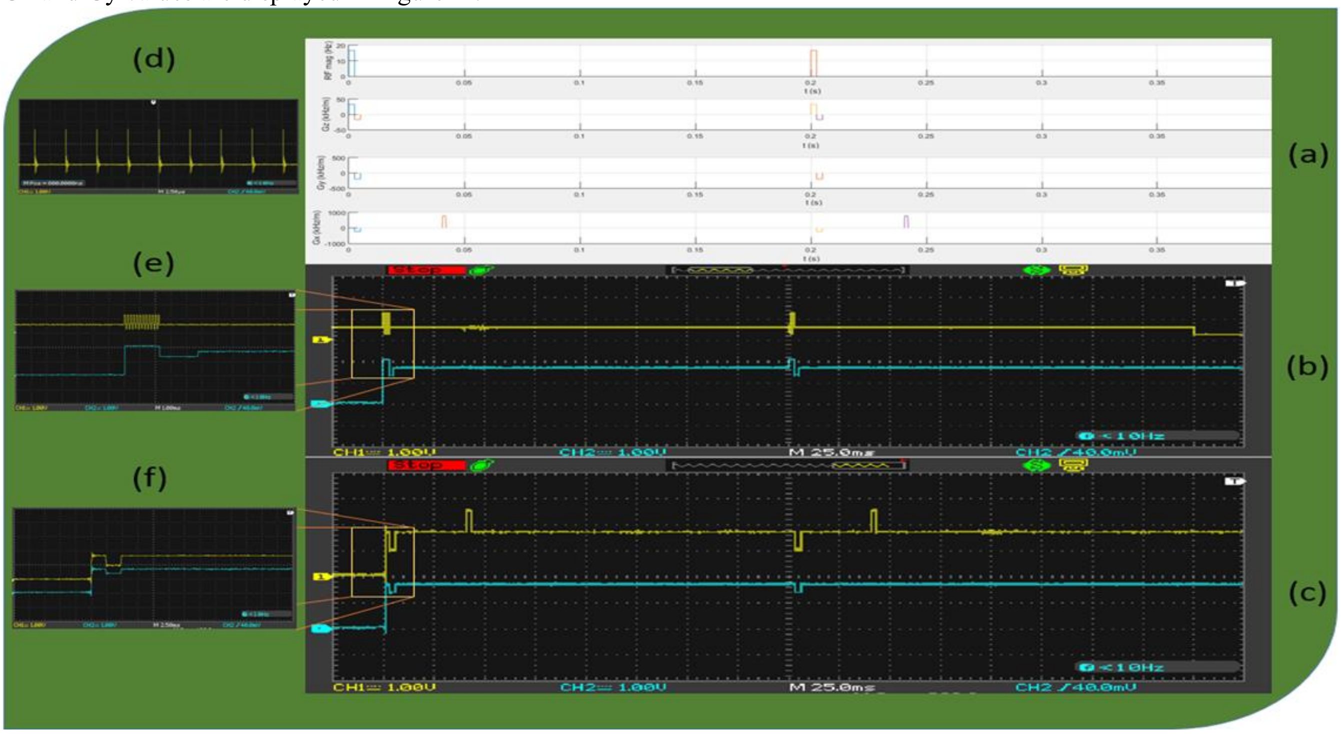


Figure 4: (a) shows the plot for the repetition time (TR) of 400ms for three gradient waveforms with the RF channel shown separately. (b) Timing diagram of Gz channel (blue) and RF channel (yellow) (c) X Gradient (yellow) and Z gradient (blue) are shown. (d) Synchronization pulses of 5µs from RF channel (e) Timing diagram of X- Gradient (blue) and RF channel (yellow) (f) Timing diagram of X- Gradient (yellow) and Y Gradient (blue).

IV. DISCUSSION AND CONCLUSION

The system provides an affordable way to produce waveforms created using Pulseq-GPI. The system is being integrated with an ADC (for reception) to reduce the time required to upload data for one instance (400ms) and/or interface the system with external memory cards to store larger sequence values. Future work involves interfacing the console with coil driver apparatus and integrating it with low-field lab-scale MRI systems.

ACKNOWLEDGEMENTS

We would like to thank Dr. Sairam Geethanath, Director, Medical Imaging Research Centre, DSCE, for the support and guidance extended during the work.

DECLARATION

Funding/ Grants/ Financial Support	No, I did not receive.
Conflicts of Interest/ Competing Interests	No conflicts of interest to the best of our knowledge.
Ethical Approval and Consent to Participate	No, the article does not require ethical approval or consent to participate, as it presents evidence that is not subject to interpretation.
Availability of Data and Material/ Data Access Statement	Not relevant.
Authors Contributions	Chennagiri Rajarao Padma, Methodology, Software, investigation, Visualization, Validation, Writing the draft, K.M. Ravikumar, Conceptualization, Validation, review and editing, and Supervision

REFERENCES

1. Handa, S., Domalain, T. and Kose, K., 2007. “Single-chip pulse programmer for magnetic resonance imaging using a 32-bit microcontroller.” Review of scientific instruments, 78(8), p.084705. [CrossRef]
2. Tsuda, M., Tamada, D., Terada, Y. and Kose, K., 2016. “Development of Digital MRI Consoles Using General-Purpose Digital Instruments and Microcontroller Boards.” Applied Magnetic Resonance, 47(8), pp.847-858. [CrossRef]
3. Michal, C.A., 2010. “A low-cost spectrometer for NMR measurements in the Earth's magnetic field.” Measurement Science and Technology, 21(10), p.105902. [CrossRef]
4. Michael Twieg, Matthew J Riffe, Natalia Gudino, and Mark A Griswold“An Open Source Mobile NMR Relaxometry Platform,” ISMRM 2013
5. Halse, Meghan Eileen. "Multi-Acquisition and Multi-Dimensional Earth's Field Nuclear Magnetic Resonance Spectroscopy." (2009).

6. Jie, S., Qin, X., Ying, L. and Gengying, L., 2005. "Home-built magnetic resonance imaging system (0.3 T) with a complete digital spectrometer." Review of scientific instruments, 76(10), p.105101. [CrossRef]
7. Takeda, Kazuyuki. "OPENCORE NMR: Open-source core modules for implementing an integrated FPGA-based NMR spectrometer." Journal of Magnetic Resonance 192.2 (2008): 218-229. [CrossRef]
8. Hemnani, Preeti, A. K. Rajarajan, and S. V. G. Ravindranath. "FPGA-based RF pulse generator for NQR/NMR spectrometer." Procedia Computer Science 93 (2016): 161-168. [CrossRef]
9. Chen, Hsueh-Ying, et al. "An ultra-low-cost NMR device with arbitrary pulse programming." Journal of Magnetic Resonance 255 (2015): 100-105. [CrossRef]
10. Hashimoto, Seitaro, Katsumi Kose, and Tomoyuki Haishi. "Development of a pulse programmer for magnetic resonance imaging using a personal computer and a high-speed digital input–output board." Review of scientific instruments 83.5 (2012): 053702. [CrossRef]
11. Hasselwander, Christopher J., Zhipeng Cao, and William A. Grissom. "gr-MRI: A software package for magnetic resonance imaging using software-defined radios." Journal of Magnetic Resonance 270 (2016): 47-55. [CrossRef]
12. Asfour, Aktham, Kosai Raoof, and Jean-Paul Yonnet. "Software defined radio (SDR) and direct digital synthesizer (DDS) for NMR/MRI instruments at low-field." Sensors 13.12 (2013): 16245-16262. [CrossRef]
13. Stang, P. P., Conolly, S. M., Santos, J. M., Pauly, J. M., & Scott, G. C. (2012). "Medusa: a scalable MR console using USB". IEEE Transactions on Medical Imaging, 31(2), 370-379. [CrossRef]
14. Layton, Kelvin J., Stefan Kroboth, FengJia, Sebastian Littin, Huijun Yu, Jochen Leupold, Jon-Fredrik Nielsen, Tony Stöcker, and Maxim Zaitsev. 2017. "Pulseq: A rapid and hardware-independent pulse sequence prototyping framework." Magnetic resonance in medicine, 77(4), p. 1544-1552 [CrossRef]
15. Keerthi Sravan R, Imam Ahmed Shaik, Stefan Kroboth, Maxim Zaitsev, and Sairam Geethanath."Implementation of Pulseq in GPI Lab" ISMRM 2017.
16. Ravi, Keerthi Sravan, Sneha Potdar, Pavan Poojar, Ashok Kumar Reddy, Stefan Kroboth, Jon-Fredrik Nielsen, Maxim Zaitsev, Ramesh Venkatesan, and Sairam Geethanath. "Pulseq-Graphical Programming Interface: Open-source visual environment for prototyping pulse sequences and integrated magnetic resonance imaging algorithm development." Magnetic resonance imaging 52 (2018): 9-15. [CrossRef]
17. http://pure-devices.com/images/Flyer/Brochure_Portable_Lab_web.pdf
18. http://www.magritek.com/applications/nmr-and-mri-education

AUTHORS PROFILE

Prof. Padma C.R. has over two decades of experience in teaching at the Department of Medical Electronics Engineering, Dayananda Sagar College of Engineering, Bengaluru. She has been engaged in research for the past ten years. She has worked on the project "National Mission on Indigenous MRI," funded by the Ministry of Electronics and Information Technology, Government of India, as well as student projects supported by KSCST. She has been awarded the E. K. Zavoisky stipend from the committee of the International Society of Magnetic Resonance in Medicine (ISMRM) Annual Meeting, held from 16 to 21 June 2018, in Paris, and a Stipend from the committee of the ISMRM workshop, held from 29 to 31 March 2019, at the India International Centre, New Delhi. She has publications in several national and international conferences and journals. She is a life member of the National Magnetic Resonance Society (NMRS), the International Society of Magnetic Resonance in Medicine (ISMRM) and the Indian chapter of the ISMRM. Indian chapter. She holds certifications from NPTEL and Coursera to her credit, and has participated in training sessions for Siemens employees on Magnetic Resonance Imaging.

Dr. K. M. Ravikumar, a Senior IEEE member, is presently working as Director at the Oxford Educational Institution in Bengaluru. He has a total of 23 years of teaching experience and has published around 55 papers in international and national journals and Conferences. He has also worked as a special officer at the VTU regional office, Mysore, and he was also a "Sub Committee -2 member of Task Force on implementation of National Education Policy-2020" (Governance & Regulation in Higher Education)- Karnataka Higher Education Parishat. He has guided eight research scholars in their PhD work and is currently guiding 5. He is also a member of the NBA and NAAC, serving as an assessor. He is a member of several professional bodies, including FIE, LMISTE, and SMIEEE. He is an "Executive Committee Member" of Sir M Visvesvaraya Memorial Foundation, Bangalore and Exe-Com member of IEEE-RAS. He is also a Member of the Local Project Committee (LPC) of Seva-in-Action (SiA), National Trust, Ministry of Social Justice & Empowerment, GOI.