

Development of a pulse programmer for magnetic resonance imaging using a personal computer and a high-speed digital input–output board

Seitaro Hashimoto, Katsumi Kose, and Tomoyuki Haishi

Citation: *Rev. Sci. Instrum.* **83**, 053702 (2012); doi: 10.1063/1.4711132

View online: <http://dx.doi.org/10.1063/1.4711132>

View Table of Contents: <http://rsi.aip.org/resource/1/RSINAK/v83/i5>

Published by the [American Institute of Physics](#).

Related Articles

Modification of ordinary-mode reflectometry system to detect lower-hybrid waves in Alcator C-Mod
Rev. Sci. Instrum. **83**, 10E325 (2012)

Quantitative coherent scattering spectra in apertureless terahertz pulse near-field microscopes
Appl. Phys. Lett. **101**, 011109 (2012)

Gain measurements of scattering-assisted terahertz quantum cascade lasers
Appl. Phys. Lett. **100**, 261111 (2012)

On the sensitivity of terahertz gyrotron based systems for remote detection of concealed radioactive materials
J. Appl. Phys. **111**, 124912 (2012)

Helicity sensitive terahertz radiation detection by field effect transistors
J. Appl. Phys. **111**, 124504 (2012)

Additional information on *Rev. Sci. Instrum.*

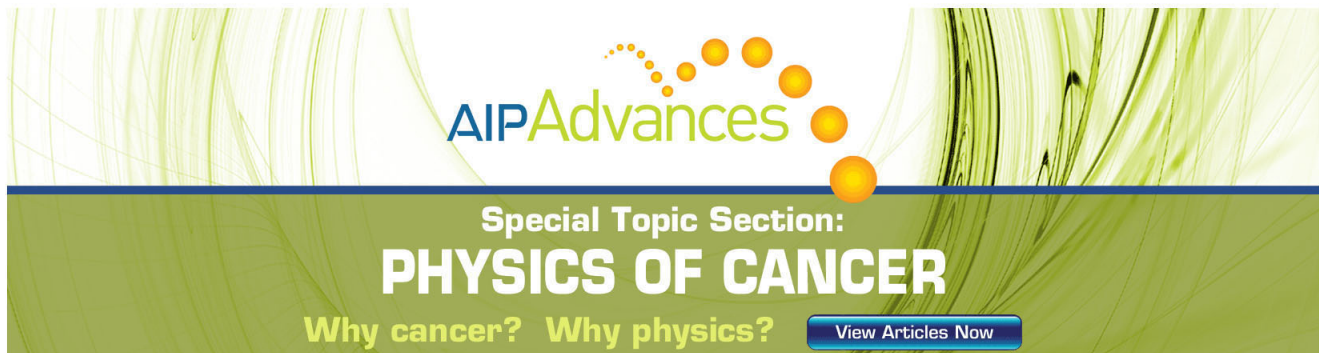
Journal Homepage: <http://rsi.aip.org>

Journal Information: http://rsi.aip.org/about/about_the_journal

Top downloads: http://rsi.aip.org/features/most_downloaded

Information for Authors: <http://rsi.aip.org/authors>

ADVERTISEMENT



AIP Advances

Special Topic Section:
PHYSICS OF CANCER

Why cancer? Why physics? [View Articles Now](#)

Development of a pulse programmer for magnetic resonance imaging using a personal computer and a high-speed digital input–output board

Seitaro Hashimoto,¹ Katsumi Kose,^{1,a)} and Tomoyuki Haishi²

¹*Institute of Applied Physics, University of Tsukuba, 1-1-1 Tennodai, Tsukuba 3058573, Japan*

²*MRTechnology Inc, 2-1-6 B5 Sengen, Tsukuba 3050047, Japan*

(Received 18 December 2011; accepted 17 April 2012; published online 7 May 2012)

We have developed a pulse programmer for magnetic resonance imaging (MRI) using a personal computer and a commercially available high-speed digital input–output board. The software for the pulse programmer was developed using C/C++ and .NET Framework 2.0 running under the Windows 7 operating system. The pulse programmer was connected to a digital MRI transceiver using a 32-bit parallel interface, and 128-bit data (16 bits \times 8 words) for the pulse sequence and the digitally detected MRI signal were transferred bi-directionally every 1 μ s. The performance of the pulse programmer was evaluated using a 1.0 T permanent magnet MRI system. The acquired MR images demonstrated the usefulness of the pulse programmer. Although our pulse programmer was developed for a specially designed digital MRI transceiver, our approach can be used for any MRI system if the interface for the transceiver is properly designed. Therefore, we have concluded that our approach is promising for MRI pulse programmers. © 2012 American Institute of Physics. [<http://dx.doi.org/10.1063/1.4711132>]

I. INTRODUCTION

A pulse programmer for magnetic resonance imaging (MRI), which generates waveforms for the radio-frequency (RF) pulses, magnetic field gradients, and data acquisition control, is a key unit that dominates the MRI system's performance. Most clinical MRI manufacturers have developed their own MRI pulse programmers optimized for their custom designed hardware units. Meanwhile, several specialized companies are supplying various types of MRI pulse programmers as well as other MRI units.^{1–4} The common feature of their MRI pulse programmers is that they are using specialized timing-generator hardware units developed in their companies.

On the other hand, various pulse programmers for nuclear magnetic resonance (NMR) and MRI have been reported for research purposes or laboratory uses.^{5–10} They use a digital signal processor (DSP),⁵ field programmable gate arrays (FPGAs),^{6–9} and a microcontroller.¹⁰ The MRI pulse programmers utilizing such hardware units described above have sufficient specifications, but the hardware developments require a long time and a significant cost. To overcome these difficulties, we propose a new approach to the MRI pulse programmer that uses a personal computer (PC) and a commercially available digital input–output (I/O) board, which requires no cost for the hardware development.

Intel (or compatible) CPU-based personal computers are very widely used for scientific instruments as well as personal use. Although the PC may be most cost effective and sufficiently powerful for most scientific applications, there are no reports of using a PC itself for a pulse generator for MRI. This is because the PC has many components requiring hardware interrupts such as the dynamic random access memory, image

display, keyboard, mouse, and other peripheral components. In addition, if multitasking operating systems such as Windows and UNIX/LINUX are used, it is impossible to generate jitter-free pulses for the MRI pulse programmer.

In this study, we have overcome this problem by using a large capacity (32 MB) buffer memory for the high-speed digital I/O board and our own software. We have constructed an MRI pulse programmer using a PC and the I/O board, and demonstrated its usefulness by MRI experiments.

II. HARDWARE

A. Pulse programmer

Figure 1 shows a block diagram of the pulse programmer developed in this study. It consists of a PC (Endeavor MR6500, EPSON, Suwa, Japan) and a high-speed digital I/O board (NI-6534, National Instruments, Austin, TX) connected to the PCI bus of the PC. The PC consists of a 64-bit multi-core CPU (Core i7-870, Intel, Santa Clara, CA) running at a clock frequency of 2.93 GHz, 12 GB memory (main memory: 4 GB, RAM disk drive: 8 GB), and a graphics board (ATI FirePro V3750, AMD, Sunnyvale, CA), running under the Microsoft Windows 7 operating system (32-bit version, Microsoft, Redmond, WA).

The high-speed digital I/O board has 32-bit I/O lines with a maximum transfer rate of 20 MHz (80 Mbytes/s for the 32-bit transfer). The 32-bit I/O lines are divided into 16-bit input and output lines. The 16-bit lines for the output are used for pulse sequence data transfer and those for the input are used for MRI signal data acquisition. The I/O board has 32 MB onboard memory per data path (input and output) for data buffering. Data transfer and data acquisition are performed synchronously with the external clock (8 MHz) supplied from the digital transceiver.

^{a)}E-mail: kose@bk.tsukuba.ac.jp. Tel.: +81-29-853-5335. FAX: +81-29-853-5205.

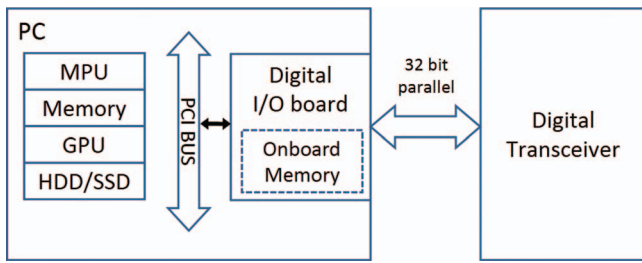


FIG. 1. Block diagram of the pulse programmer and the digital transceiver.

B. Digital transceiver

Figure 2 shows a block diagram of the digital transceiver (DTRX-7, MRTechnology, Tsukuba, Japan) used in this study. The transceiver was developed for low field (< 0.3 T, resonance frequency: < 13 MHz for protons) MRI applications. It includes a parallel I/O port for communication with the PC, a universal serial bus (USB) interface, three 16-bit digital to analog (DA) converters (1 megasample per second (MSPS)) for gradient fields, a 14-bit DA converter (60 MSPS) for RF pulses, a 16-bit analog to digital (AD) converter (60 MSPS) for MRI signal sampling, and an RF oscillator for the Larmor frequency (< 14 MHz) source.

The sampled MRI signal data (16 bits) are digitally detected using two Larmor frequency digital signals having 0° and 90° phases (digital quadrature detection¹¹), filtered using CIC (cascaded integrator-comb) filters, and down-sampled to 1 MSPS 16-bit data. Because the in-phase and quadrature-phase digital data are obtained at 1 MSPS, the down-sampled data have 1 MHz signal bandwidth, which is sufficient for most MRI applications.

Most of the digital circuits described above are configured on an FPGA (Cyclone III, ALTERA, San Jose, CA) running at the clock frequency of 60 MHz, which is common to the sampling clock of the NMR signal.

Figure 3 shows the data formats used for pulse sequence data transfer and MRI signal data acquisition. The eight sequential 16-bit words used for the RF pulse shape, the gradi-

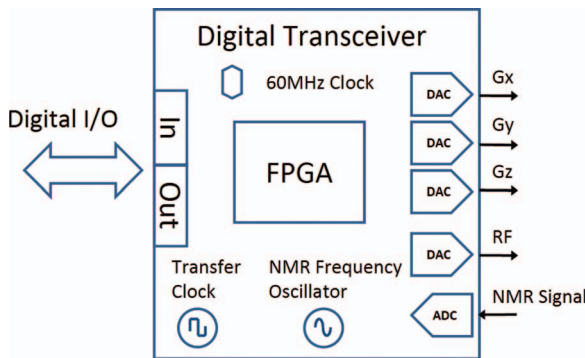


FIG. 2. Block diagram of the MRI digital transceiver. Three 16-bit DA converters are used for the waveforms of the three-channel magnetic field gradients. A 14-bit DA converter is used for the waveform of the RF pulse. A 16-bit AD converter is used for NMR signal sampling. All of the components are controlled by the FPGA (Cyclone III). The transceiver also includes an RF oscillator for NMR signal detection (up to 14 MHz) and a clock generator (8 MHz) for data transfer between the pulse programmer and the digital I/O board.

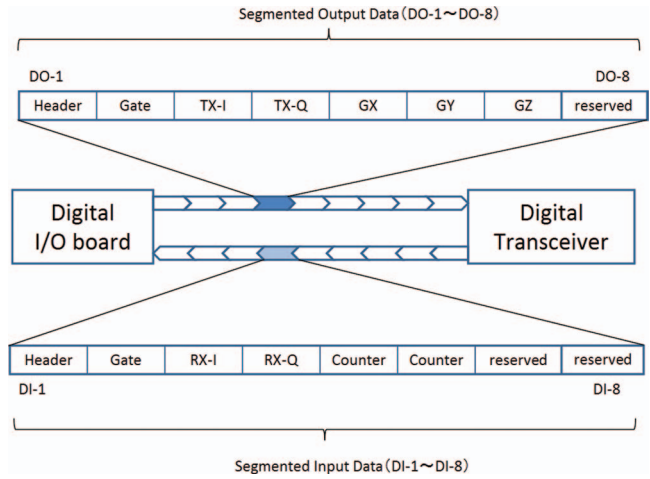


FIG. 3. Data transfer format used for the pulse sequence data transfer and NMR signal data acquisition. Eight 16-bit words used for the RF pulse shape, gradient field amplitudes, and gate masks are transferred from the PC to the digital transceiver every $1 \mu s$. Eight 16-bit words used for the NMR signal for in-phase (RX-I) and quadrature-phase (RX-Q) and their time stamps are transferred from the digital transceiver to the PC every $1 \mu s$.

ent field waveform, and the gate masks are transferred from the PC to the digital transceiver every $1 \mu s$. Therefore, the output transfer rate from the PC to the transceiver is 16 Mbytes/s. The eight sequential 16-bit words used for the MRI signal data of in-phase (RX-I) and quadrature-phase (RX-Q) and their time stamps are transferred from the digital transceiver to the PC every $1 \mu s$. Therefore, the input transfer rate from the transceiver to the PC is also 16 Mbytes/s.

C. Permanent magnet MRI system

Figure 4 shows a block diagram of the permanent magnet MRI system used for the imaging experiments. This system consists of a permanent magnet, a gradient coil set, an RF coil, and MRI electronics including the pulse programmer and the digital transceiver described above.

The permanent magnet (NEOMAX Engineering, Takasaki, Japan) is a yokeless magnet made of NdFeB

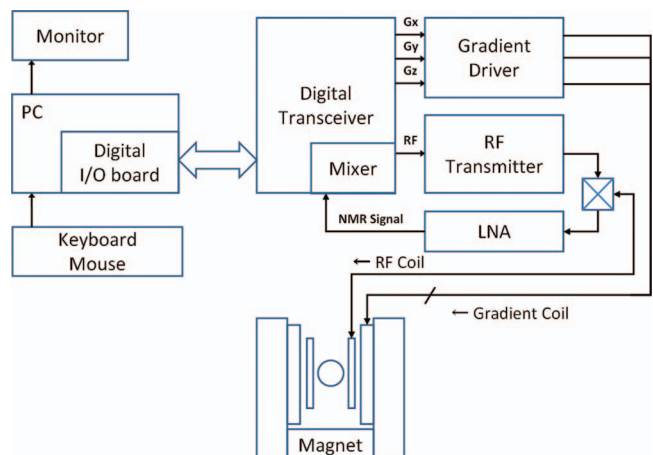


FIG. 4. Block diagram of the permanent magnet MRI system using the pulse programmer and the digital transceiver.

magnetic material blocks. The specifications of the magnet are: field strength = 1.03 T, gap width = 90 mm, homogeneity = 10 ppm over the 35 mm diameter sphere, and total weight = 980 kg.¹² The gradient coils were designed using the target field approach for the transverse gradient coil (Gx and Gy) and using a genetic algorithm for the axial (Gz) coil. The efficiencies of the gradient coils were 12.8, 12.6, and 12.2 mT/m for the Gx, Gy, and Gz coils, respectively. The RF coil was a solenoid (diameter = 64 mm, length = 80 mm) made of a Cu stripe (0.1 mm thick, 6 mm wide) divided into four elements using three chip capacitors (78 pF each).

Because the Larmor frequency (~ 43.85 MHz) of our system was too high for the digital transceiver developed for low field MRI applications (< 13 MHz), frequency converters using image rejection mixers were used to change the RF frequency from 12.8/43.85 MHz to 43.85/12.8 MHz, as shown in Fig. 4.

III. SOFTWARE

We developed software for the pulse programmer and the data acquisition system. All of the software was developed using C/C++ and .NET Framework 2.0 of Visual Studio 2008 (Microsoft, Redmond, WA) running under the Windows 7 operating system. To access the digital I/O board, we used the driver software (NI-DAQmx ver. 9.0.2, National Instruments, Austin, TX) supplied by the manufacturer.

A. Pulse programmer software

The function of the pulse programmer software (pulser program) is to convert the pulse sequence text file to binary data for the digital transceiver and to store the binary data in the onboard memory of the I/O board. We will describe the detail of the software using Figures 5 and 6.

Figure 5 shows the timetable and the corresponding timing chart for a 3D gradient echo sequence. The timetable consists of three columns. The first shows the time when the pulse sequence events take place in terms of 100 ns steps. Although the time resolution is 1 μ s in our system, we used 100 ns time steps to ensure compatibility with the pulse sequence files that have been used in our laboratory.^{5,13} The second column shows the events, represented by two capital letters, including, RF pulse (RF), field gradients (GX, GY, GZ), and data acquisition control (AD). The third column shows four-digit hexadecimal numbers representing the amplitude of the field

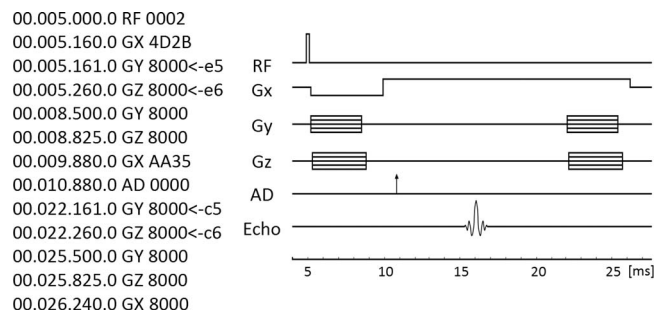


FIG. 5. (a) Typical timetable for a 3D gradient echo imaging sequence. (b) Timing chart for the 3D gradient echo sequence.

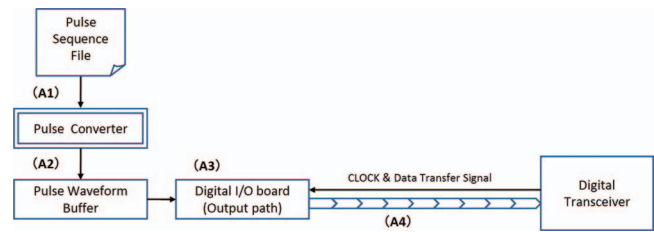


FIG. 6. Control flow of the pulse programmer software. The pulse sequence text file is converted to binary data (A1) and stored in one of two pulse waveform buffers (A2). The binary data are transferred from the pulse waveform buffer to the onboard memory of the digital I/O board (A3) and transferred to the digital transceiver (A4).

gradients for the GX, GY, or GZ, the shape of the RF pulses for RF, and the acquisition phase of the signal for AD. The additional terms, <-e5, <-e6, <-c5, and <-c6, show phase-encoding gradients, which are automatically incremented or decremented in the imaging sequence according to the gradient amplitude tables defined beforehand.

Figure 6 shows the control flow of the pulser program. First, the pulser program reads the pulse sequence text file as shown in Fig. 5, converts it to binary data to be directly output to the digital transceiver (A1), and stores the data in one of two pulse waveform buffers in the RAM disk (A2). Each data conversion is performed for one complete repetition time (TR) of the pulse sequence. For example, when the TR is 50 ms, the binary data amount to 800 000 bytes (16 bit \times 8 words \times 50 000 events including blank events), because the data transfer rate is 1 μ s.

After one of the pulse waveform buffers is filled, the stored binary data are transferred from the waveform buffer to the onboard memory of the digital I/O board (A3), and automatically output to the digital transceiver every 1 μ s (A4). Because we used two pulse waveform buffers in the RAM disk, and the data conversion program and the data transfer program are assigned to two separate thread programs, the output of the pulse sequence data is performed continuously.

B. Data acquisition software

The data acquisition software (acquisition program) transfers the continuously sampled NMR signal data from the digital transceiver to the onboard memory of the I/O board, transfers them from the onboard memory to the acquisition data buffer in the main memory, samples the k-space data from the NMR signal data, and processes them for data display and saving. We describe the details of the acquisition program using Figure 7.

Before the start of the MRI measurements, the acquisition program reads the pulse sequence text file and prepares the sampling table that describes how to sample the k-space data from the NMR signal data continuously sampled every 1 μ s in the digital transceiver (B1). The data flow of the sampled data is as follows.

The acquisition program transfers the continuously sampled NMR signal data from the transceiver to the onboard buffer memory of the I/O board (B2), and transfers them from the onboard memory to the acquisition data buffer in the main memory (B3). The acquisition program samples the

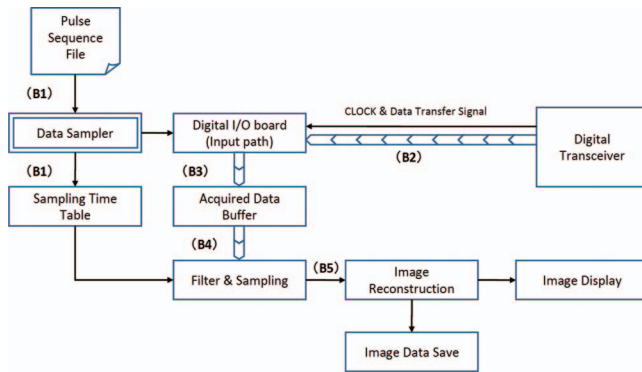


FIG. 7. Control flow of the data acquisition software. The k-space data for MR images are sampled from the continuously acquired NMR signal data and used for image reconstruction.

k-space data from the continuously sampled data according to the sampling table noted above (B4). Typically, the data sampling is performed every $10 \mu\text{s}$, because low-pass filtering with about 100 kHz cutoff frequency is already applied in the digital transceiver. The acquisition program saves the k-space data to the RAM disk drive and displays them on the display of the PC for monitoring the NMR signal (B5).

IV. EXPERIMENTS

We evaluated the performance of the pulse programmer, which was as follows.

The first experiment demonstrated the time resolution ($1 \mu\text{s}$) of the pulse programmer using a very fast ($\text{TR} = 1 \text{ ms}$) MRI pulse sequence. The operation of the pulse programmer including the PC operating system was also examined when the TR was decreased to 0.5 ms.

The second experiment was fast imaging of a chicken egg using 3D gradient echo sequences with and without rewinding phase-encoding gradients. This experiment was performed to confirm the steady state of the nuclear magnetization for the fast sequences. The pulse sequence parameters were: $\text{TR} = 20 \text{ ms}$, TE (echo time) = 3 ms, image matrix = $256 \times 256 \times 64$, field of view (FOV) = $(64 \text{ mm})^3$, NEX (number of excitations) = 4, total data acquisition time = 21.8 min.

The third experiment is high resolution imaging using a 3D spin echo sequence with an NMR lock sequence.¹³ The

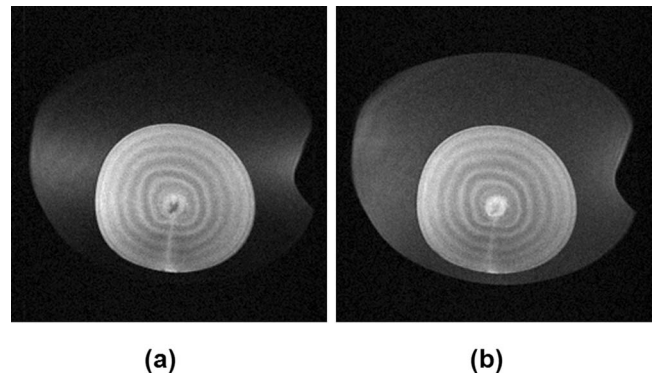


FIG. 9. 2D cross-sectional images of a chicken egg selected from 3D image datasets acquired with 3D gradient echo sequences. The left image was acquired without rewinding phase encoding gradients. A "FLASH band" is observed in the left image. The right image was acquired using rewinding phase-encoding gradients. The laminar structure of the yolk is visualized by T_2^* contrast. Data acquisition parameters are: $\text{TR} = 20 \text{ ms}$, $\text{TE} = 3 \text{ ms}$, $\text{FOV} = (64 \text{ mm})^3$. Image matrix = $256 \times 256 \times 64$. NEX = 4. Total data acquisition time = 21.8 min.

pulse sequence parameters are: $\text{TR} = 200 \text{ ms}$, $\text{TE} = 16 \text{ ms}$, image matrix = $512 \times 512 \times 64$, $\text{FOV} = (64 \text{ mm})^3$, NEX = 4, total data acquisition time = 7.3 h. This experiment was performed to confirm the long-term stability of our pulse programmer.

V. RESULTS

Figure 8 shows screenshots of the waveforms of the RF pulse and the field gradients (G_x , G_y , and G_z) displayed on a digital oscilloscope (TDS2014, Tektronix, Portland, OR). Figure 8(a) shows waveforms for the field gradients. Because the horizontal axis is $1 \mu\text{s}/\text{div}$, this picture clearly demonstrates that the $1 \mu\text{s}$ time-resolution is achieved for our pulse programmer. Figure 8(b) shows waveforms of the RF pulse and the field gradients for a 3D gradient echo sequence with $\text{TR} = 1 \text{ ms}$. This TR was the shortest one in which the MRI sequence worked without problems. However, when the TR was decreased to 0.5 ms, the Windows operating system became unstable. We show a solution to this problem in the next section.

Figure 9 shows 2D cross-sectional images selected from 3D image datasets of a chicken egg acquired with the fast 3D gradient echo sequences described above. The left image was

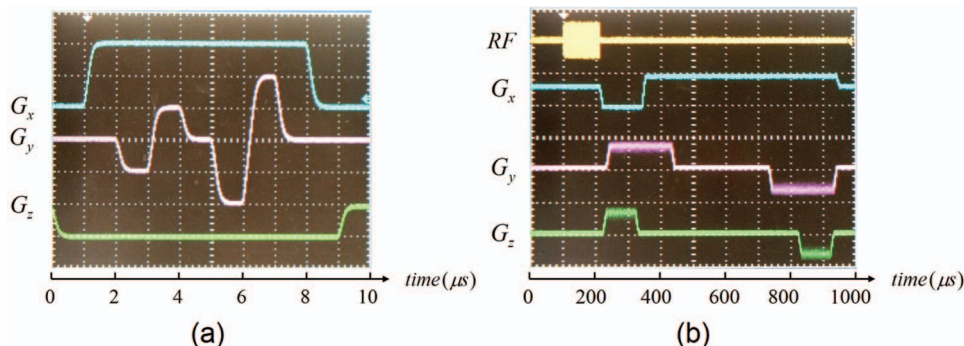


FIG. 8. (a) Waveforms for the field gradients (G_x , G_y , and G_z) generated by the pulse programmer. The horizontal axis is $1 \mu\text{s}/\text{div}$. (b) Waveforms of the RF pulse and the field gradients for a 3D gradient echo sequence. $\text{TR} = 1 \text{ ms}$. The horizontal axis is $100 \mu\text{s}/\text{div}$.

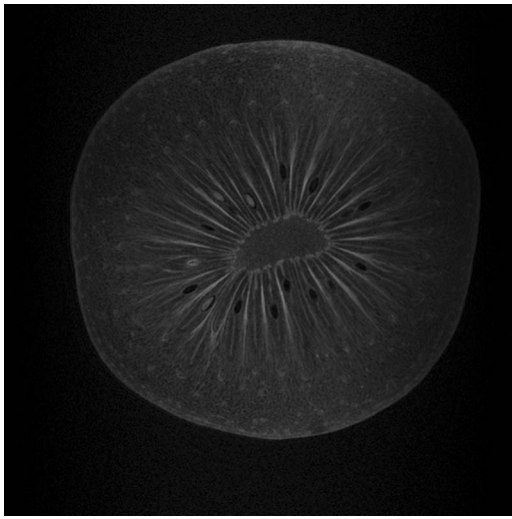


FIG. 10. 2D cross-sectional image of a kiwi fruit selected from a 3D image dataset acquired with a 3D spin echo sequence. Data acquisition parameters are: TR = 200 ms, TE = 16 ms, FOV = (64 mm)³. Image matrix = 512 × 512 × 64. NEX = 4. Total data acquisition time = 7.3 h.

acquired without rewinding gradients.¹⁴ A “FLASH band” is observed in the left image.¹⁵ The right image was acquired with a gradient echo sequence with rewinding gradients. The FLASH band disappeared in the right image because a steady state was established even for protons in the egg white having relaxation times (T_1 and T_2) much longer than TR (20 ms).

Figure 10 shows a cross-sectional image of a kiwi fruit selected from a 3D image dataset acquired with the 3D spin echo sequence described in the preceding section. The NMR lock operation was applied before every 2D data acquisition in k -space.¹³ The high-resolution image in Fig. 10 clearly shows that our pulse programmer worked well for long-term data acquisition.

VI. DISCUSSION

In this study, we have developed a flexible pulse programmer for MRI using a PC and a high-speed digital I/O board and demonstrated its usefulness by MRI experiments. In this section, we discuss on the advantages and disadvantages of our pulse programmer.

The first advantage of our pulse programmer is that we used only commercially available and general-purpose units, so that no hardware development was required. This means low hardware cost. From the viewpoint of the hardware cost, the pulse programmer using a one-chip 32-bit microcontroller reported by our group is much more inexpensive (~\$10).¹⁰ However, the capacity of the memory for the pulse sequence in the microcontroller was limited to 2 Kwords. Although some of the pulse programmers supplied by the companies¹⁻⁴ and reported by the research groups⁶⁻⁹ have sufficient performance for MR image acquisition; however, substantial time and costs for the hardware development are indispensable.

The second advantage of our pulse programmer is that software development is required only for the host PC. Comparing with pulse programmers using a DSP and a microcontroller that require software developments both the host and

target CPU, the software development for our system is much simpler. Comparing with pulse programmers using FPGAs, circuit design using a hardware description language is not required.

The third advantage of our pulse programmer is that the memory capacity for the waveforms of field gradients and RF pulses is very large or practically unlimited. This is because we used a RAM disk drive (8 GB) for the wave memory. Practically speaking, in MRI pulse programmers using a DSP, a microcontroller, and FPGAs, the capacity for the wave memory cannot be extended to several Gbytes, because they do not use commercially available DRAM memory modules, but use memory chips for their specially designed hardware units.

On the other hand, the first disadvantage of our pulse programmer is that unexpected interruptions to the operating system (Windows 7) may interrupt or disturb the MRI measurements. Because the capacity of the onboard memory of the digital I/O board is 32 Mbytes for each input and output path and 16 bytes (16 bits × 8 words) data are transferred every 1 μ s, interruptions of more than two seconds by the operating system change the TR of the pulse sequence.

While using the pulse programmer, we have never observed such interruptions or disturbances while the TR was larger than 1 ms. However, as described in Sec. V, the Windows operating system became unstable when the TR was decreased to 0.5 ms. We think that this instability was mainly caused by the multithreaded program used to transfer the binary data from the double buffer in the RAM disk to the buffer memory of the I/O board. However, if we write down the whole 2D imaging sequence within the TR (> 1 ms), a fast 2D or 3D imaging sequence with TR shorter than 1 ms can be installed without problems.

The second disadvantage of our pulse programmer is that its time resolution is limited to 1 μ s, relatively long compared with those reported earlier (20 ~ 100 ns) using a DSP,⁵ a microcontroller,¹⁰ and FPGAs.⁶⁻⁹ The 1 μ s time resolution was determined as a compromise between the transfer rate of the I/O board (80 Mbytes/s) and the capacity of the buffer memory (e.g., 8 000 000 000 bytes for TR = 500 s). However, because the signal bandwidth of the usual MRI signal is limited to several hundreds kHz, 1 μ s time resolution is sufficient for usual MRI sequences including echo-planar imaging.

VII. CONCLUSION

In this study, an MRI pulse programmer was developed using a commercially available PC and a high-speed digital I/O board with a large buffer memory. Although our pulse programmer was developed for a specially designed digital transceiver, our approach can be used for any MRI system if the interface for the MRI transceiver is properly designed. Therefore, we have concluded that our approach can produce economical and high-quality MRI pulse programmers with no hardware development cost.

ACKNOWLEDGMENTS

We thank Mr. Masaru Aoki at Digital Signal Technology, Inc. for technical advice on the digital transceiver.

- ¹See <http://www.tecmag.com/index.html> for Tecmag.
- ²See <http://www.magritek.com/> for Magritek.
- ³See <http://spincore.com/> for Spin Core.
- ⁴See <http://www.mrsolutions.co.uk/> for MR Solutions.
- ⁵K. Kose and T. Haishi, in *Spatially Resolved Magnetic Resonance*, edited by P. Blumler, B. Blumich, R. Botto, and E. Fukushima (Wiley-VCH, 1998), p. 703.
- ⁶S. B. Belmonte, R. S. Sarthour, I. S. Oliveira, and A. P. Guimaraes, *Meas. Sci. Technol.* **14**, N1 (2003).
- ⁷S. Jie, X. Qin, L. Yang, and L. Gengying, *Rev. Sci. Instrum.* **76**, 105101 (2005).
- ⁸K. Takeda, *Rev. Sci. Instrum.* **78**, 033103 (2007).
- ⁹W. Mao, Q. Bao, L. Yang, Y. Chen, C. Liu, J. Qiu, and C. Ye, *Meas. Sci. Technol.* **22**, 025901 (2011).
- ¹⁰S. Handa, T. Domalain, and K. Kose, *Rev. Sci. Instrum.* **78**, 084705 (2007).
- ¹¹M. Villa, F. Tian, P. Cofrancesco, J. Halánek, and M. Kasal, *Rev. Sci. Instrum.* **67**, 2123 (1996).
- ¹²T. Shirai, T. Haishi, S. Utsuzawa, Y. Matsuda, and K. Kose, *Magn. Reson. Med. Sci.* **4**, 137 (2005).
- ¹³T. Haishi, T. Uematsu, Y. Matsuda, and K. Kose, *Magn. Reson. Imaging* **19**, 875 (2001).
- ¹⁴A. Haase, J. Frahm, D. Matthaei, W. Hänicke, and K. D. Merboldt, *J. Magn. Reson.* **67**, 258 (1986).
- ¹⁵J. Frahm, W. Hänicke, and K. D. Merboldt, *J. Magn. Reson.* **72**, 307 (1987).