A single-chip pulse programmer for magnetic resonance imaging using a 32-bit microcontroller

Shinya Handa, Thierry Domalain, Katsumi Kose [a)]

Institute of Applied Physics, University of Tsukuba

Tsukuba, Ibaraki, 305-8573, Japan

a) To whom correspondence should be addressed.

e-mail: kose@bk.tsukuba.ac.jp

ABSTRACT

An MRI pulse programmer has been developed using a single-chip microcontroller (ADμC7026). The microcontroller includes all the components required for the MRI pulse programmer: a 32-bit RISC CPU core, 62 Kbytes of flash memory, 8 Kbytes of SRAM, two 32-bit timers, four 12-bit DA converters, and 40 bits of general purpose I/O. An evaluation board for the microcontroller was connected to a host PC, an MRI transceiver, and a gradient driver using interface circuitry. Target (embedded) and host PC programs were developed to enable MRI pulse sequence generation by the microcontroller. The pulse programmer achieved a (nominal) time resolution of approximately 100 ns and a minimum time delay between successive events of approximately 9 μs. Imaging experiments using the pulse programmer demonstrated the effectiveness of our approach.

I. INTRODUCTION

Several major companies now supply advanced and sophisticated Nuclear Magnetic Resonance (NMR) spectrometers and Magnetic Resonance Imaging (MRI) systems. However, when special purpose NMR or MRI systems are required, such as for portability, on-line monitoring, or educational purposes, custom-built NMR/MRI systems are desirable.[1-6] In developing a custom-built MRI system, the pulse programmer (PPG) is the key unit to be developed. This is because magnets, gradient probes, transmitters, preamplifiers, AD converter boards, and modulator/detector modules and chips are commercially available, whereas a PPG for MRI systems is very difficult to obtain commercially.

Although there are many papers describing PPGs for NMR spectrometers,[7-10] only a few paper report on PPGs for MRI systems.[5,11] This is because the PPG for MRI systems is required to generate three-channel magnetic field gradients and radio frequency (RF) pulse waveforms in complex imaging sequence loops and requires many bits (e.g., up to 128 bits) for their control. One solution to this problem is the use of a commercially available digital signal processor (DSP) board designed for real-time system control.[11] Because recent developments in LSI technology have enabled such performance to be achieved within a single chip, a PPG for MRI systems can be made using a single LSI device. In the work reported here, we have implemented the PPG function for MRI on a single 32-bit microcontroller (ADμC7026) and demonstrated its feasibility via MRI experiments.

II. HARDWARE

A single-chip 32-bit microcontroller (ADμC7026, Analog Devices, Inc., USA) was used for the PPG developed in this work. The chip includes a 32-bit RISC CPU core (ARM7TDMI, ARM Ltd., UK), 31 Kwords × 16-bit on-chip flash memory, two Kwords × 32-bit on-chip static RAM, a 16-channel AD converter, four 12-bit DA converters, two 32-bit timers, and a 40-bit general purpose digital I/O port.[12] We used an evaluation board (FRK-ADμC, CQ Publishing Co., Japan), in which an ADμC7026 chip is mounted on a 60 mm × 50 mm printed circuit board with a voltage regulator circuit,

RS-232C serial port, external I/O port, and other peripheral circuits.[13]

Figures 1 and 2 show the block diagram and an overview of the PPG developed in this work. As shown in the block diagram and the picture, the evaluation board was connected to the host PC's USB port via the USB to RS-232C converter circuit, and connected to the MRI transceiver (DTRX4, MRTechnology, Japan) and the gradient driver (±10V, ±5A) via the voltage level converter board. Although several chips and circuits were needed for interfaces, the ADμC7026 chip supplied all the MRI system's PPG functions.

Figure 3 shows a block diagram of the internal timer and interrupt circuit used for timing control of the PPG. The core clock frequency (41.78 MHz: the default frequency for the evaluation board) was used for Timer1, and an interrupt request (IRQ) from Timer1 was used for the event processing described later. Although the time resolution of the Timer1 was actually about 23.9 ns, we developed programs for the PPG on the assumption that the core clock frequency was 40 MHz and the time resolution was 25 ns.

III. SOFTWARE

Target (embedded) programs for the microcontroller were developed using the "Keil Development Suite for ARM" (ARM. Ltd., UK) supplied with the evaluation board. Console programs for the host PC were developed using the GNU C compiler on a Linux emulation console (Cygwin, Red Hat, USA). All the programs were developed on a host PC running the Windows XP operating system (Microsoft, USA).

Figure 4 shows an example of a pulse sequence timetable and corresponding time chart of a 3D gradient echo imaging sequence developed for our PPG. The timetable consists of three columns. The first column gives the time when the event takes place, in terms of 100 ns unit. Although 25ns time resolution can be achieved in this system, we used the 100 ns time resolution to keep compatibility with time sequence tables that have been used in our group.[11] The time units are exact when a 40 MHz core CPU clock is used. The second column gives the two letter event ID, namely RF

pulse, gradient field (GX, GY, or GZ), or AD converter trigger. The third column described by a four digit hexadecimal number gives amplitude of the gradient field for GX, GY, and GZ, ID number of the RF pulse shape for RF, and trigger pulse polarity and width (optional) of AD. Two additional terms, <-pe1 and <-pe2, describe phase encoding gradients: how the amplitude is automatically changed according to the phase encoding tables supplied in other disk files.

The timetable text file described above was converted into binary form for the microcontroller chip by a host PC program. The event times were converted to 32-bit (four-byte) time difference data between the times of two successive events because these data were to be loaded into the up/down counter of Timer1, which, at the end of its count, generated IRQ to the CPU to activate the event. The actual time difference data were quadrupled because the CPU core clock (41.78 MHz) was used for Timer1. The event ID (described in the 2nd column) and its content (described in the 3rd column) were converted to the relevant output port address (16-bit local address mapped in the 32-bit address space of the CPU) and the output data (amplitude or control word coded in 16 bits) used to activate the events. Usually, an event was converted to an eight-byte data item (time difference data:32 bits, local address:16 bits, and contents:16 bits), but some events (e.g., AD trigger) were converted to two eight-byte data items because the events required pulse (both set and reset) outputs.

Figure 5 shows a functional diagram of the programs and data files developed for the PPG. At the start of the PPG operation, text files (header file, event table file, and phase-encoding table file) are converted to binary and downloaded to the microcontroller memory, which also contains are the embedded program. After the downloading, the target system is reset to start the imaging pulse sequence.

IV. EXPERIMENTS

Before starting the imaging experiments, an imaging pulse sequence was generated at a constant repetition rate, and the RF pulse shapes and gradient waveforms were observed using a digital oscilloscope. No time jitter was observed within the sequence. We also confirmed the time resolution

of approximately 96 ns (23.9 ns × 4). The minimum time delay between two successive events was observed to be approximately 9 μs.

Two-dimensional (2D) imaging experiments were performed using a water phantom and a portable MRI system developed in our laboratory.[14] The water phantom was made of 37 glass capillaries (outer diameter = 1.6 mm, inner diameter = 0.8 mm) placed in an NMR sample tube (outer diameter = 15.0 mm, inner diameter = 13.5 mm) and filled with aqueous $CuSO_4$ solution. The magnet was a U-shaped permanent magnet (NEOMAX, Japan): field strength 0.3 T, gap width 80 mm, homogeneity approximately 50 ppm over a spherical volume of 30 mm diameter, and weight 60 kg.

Figure 6 shows a 2D image of the water phantom acquired with a 2D spin echo sequence (repetition time (TR) = 100 ms, echo time (TE) = 12 ms). The image matrix was 128 × 128 and the pixel size was 200 μm × 200 μm. Slice selection was not performed. This image clearly demonstrates the effectiveness of the PPG developed in this work.

V. DISCUSSION AND CONCLUSION

Our PPG achieved a (nominal) time resolution of 100 ns, and a minimum delay time between two successive events of approximately 9 μs. The time resolution could be improved to 25 ns (nominal) if the time difference data are not quadrupled. However, the 100 ns time resolution is sufficient for most MRI applications because the signal bandwidth is less than 100 kHz. The minimum delay time (9 μs) is determined by the overhead time in the interrupt sequence and other output sequences. Locating the embedded program in the SRAM area instead of flash memory could reduce this delay time, because the memory access time would be reduced from two clock times to one clock time. However, the present minimum delay time is sufficient for many MRI applications.

Recent reports on NMR and MRI PPGs describe using Field Programmable Gate Arrays (FPGAs) and achieving 10 ns to 50 ns time resolutions with 16 bit to 64 bit outputs.[5,9,10] Because our system has a 25-100 ns time resolution with 88 bit output (12 bit DAC × 4 and 40-bits GPIO), our system has achieved a performance similar to that of FPGA systems and a DSP system.[11] However,

our system has several advantages over FPGA systems. Firstly, less time is required for development because there is no need to design either logic circuitry for the PPG or output circuitry for the analog outputs. Secondly, there is cost. The microcontroller includes four 12-bit DA converters and the total cost is approximately only $US10. This cost cannot be achieved using an FPGA and external DA converters. On the other hand, there are limitations to our system: the relatively long minimum delay time (approximately 9 μs) between events, the limited capacity (two Kwords × 32-bits SRAM and 62 Kwords × 16-bits flash memory) for event data and phase-encoding tables, and a limited capacity for extensions such as quadrature modulation or RF phase control. However, we believe our system can be used in many conventional MRI applications.

In conclusion, we have developed an MRI PPG using a single-chip microcontroller and demonstrated its usefulness via imaging experiments.

REFERENCES

1. T. Haishi, T. Uematsu, Y. Matsuda, and K. Kose. Magn Reson Imag **19**, 875 (2001).

2. M. Rokitta and E. Rommel, Rev. Sci. Instrum. **71**, 4257 (2000).

3. K. Kose, Y. Matsuda, T. Kurimoto, S. Hashimoto, Y. Yamazaki, T. Haishi, S. Utsuzawa, H. Yoshioka, S. Okada, M. Aoki, and T. Tsuzaki. Magn Reson Med **52**, 440 (2004).

4. T. Shirai, T. Haishi, S. Utsuzawa, Y. Matsuda, and K. Kose. Magn Reson Med Sci **4**, 137 (2005).

5. S. Jie, X. Qin, L. Yang, and L. Gengying, Rev. Sci. Instrum. **76**, 105101 (2005).

6. N. Iita, S. Handa, S. Tomiha, and K. Kose, Magn Res Med **57**, 272 (2007).

7. E. A. Wachter, E. Y. Sidky, and T. C. Farrar, Rev. Sci. Instrum. **59**, 2285 (1988).

8. T. Toyoda, H. Yoshida, O. Oishi, and S. Miyajima, Rev. Sci. Instrum. **68**, 3140 (1997).

9. S. B. Belmonte, R. S. Sarthour, I. S. Oliveira, and A. P. Guimaraes, Meas. Sci. Technol. **14**, N1 (2003).

10. K. Takeda, Rev. Sci. Instrum. **78**, 033103 (2007).

11. K. Kose and T. Haishi, in Spatially Resolved Magnetic Resonance, Edited by P. Blumler, B. Blumich, R. Botto, and E. Fukushima, Wiley-VCH, 703 (1998).

12. Analog Devices, ADμC7026 Datasheet.

13. Design Wave Magazine, March 2006, CQ publishing Co., Tokyo.

14. K. Kose, in NMR Imaging in Chemical Engineering, Edited by S. Stapf and Song-I Han, Wiley-VCH, 77 (2006).

FIGURE CAPTIONS

FIG.1. The block diagram of the PPG developed in this work. The ADμC7026 microcontroller includes 31 Kwords × 16-bit on-chip flash memory, two Kwords × 32-bit SRAM, two 32-bit timers, four 12 bit DA converters, and 40 bits of general purpose I/O (GPIO). It was connected to the PC via a USB interface, and to the MRI transceiver and the gradient driver via voltage level converters (not shown).

FIG. 2. An overview of the PPG developed in this work. It comprises three printed circuit boards: the RS-232C to USB conversion board (on the left under the microcontroller evaluation board), the microcontroller evaluation board (on the left above the conversion board), and the voltage level converter board (on the right). The PPG is connected to the gradient driver using three SMA connectors shown on the voltage level converter board and connected to the MRI transceiver using an SMA connector and a flat cable.

FIG. 3. The block diagram of the timer and interrupt circuit used for timing control of the PPG. The core clock frequency (41.78 MHz) was used for Timer1, which generated an IRQ to the CPU at the end of its count to measure the delay time between two successive PPG events.

FIG. 4. A pulse sequence timetable (left) and corresponding time chart (right) for a 3D gradient echo imaging sequence developed for our PPG.

FIG. 5. A functional diagram of the software developed for the PPG. At the start of the PPG operation, text files (header file, event table file, and phase encoding table file) are converted to binary and downloaded to the microcontroller memory, which also contains the embedded program. After the downloading, the target system is reset to start the imaging pulse sequence. (The RF pulse shape file will be implemented in a future version.)

FIG. 6. A 2D MR image of a water phantom acquired via the PPG developed in this work. The image matrix was 128 × 128 and the pixel size was 200 μm × 200 μm.

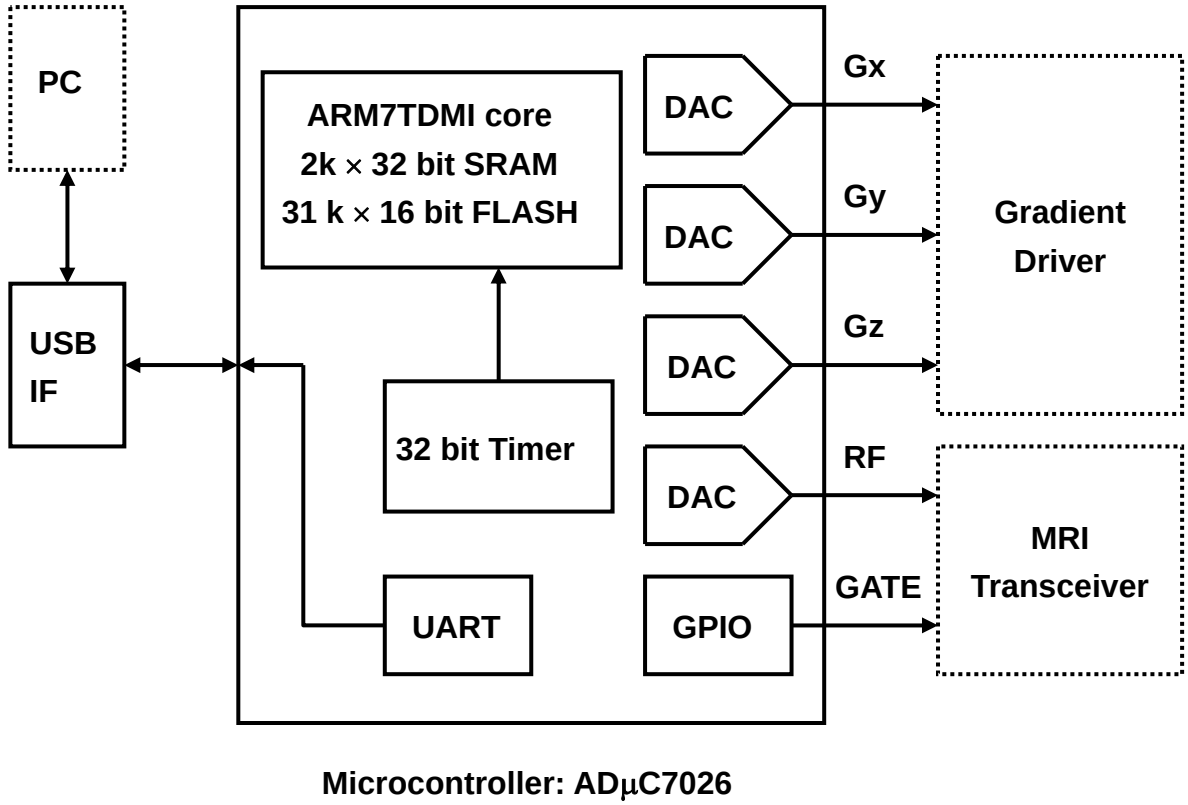

FIG.1. The block diagram of the PPG developed in this work. The ADμC7026 microcontroller includes 31 Kwords × 16-bit on-chip flash memory, two Kwords × 32-bit SRAM, two 32-bit timers, four 12 bit DA converters, and 40 bits of general purpose I/O (GPIO). It was connected to the PC via a USB interface, and to the MRI transceiver and the gradient driver via voltage level converters (not shown).

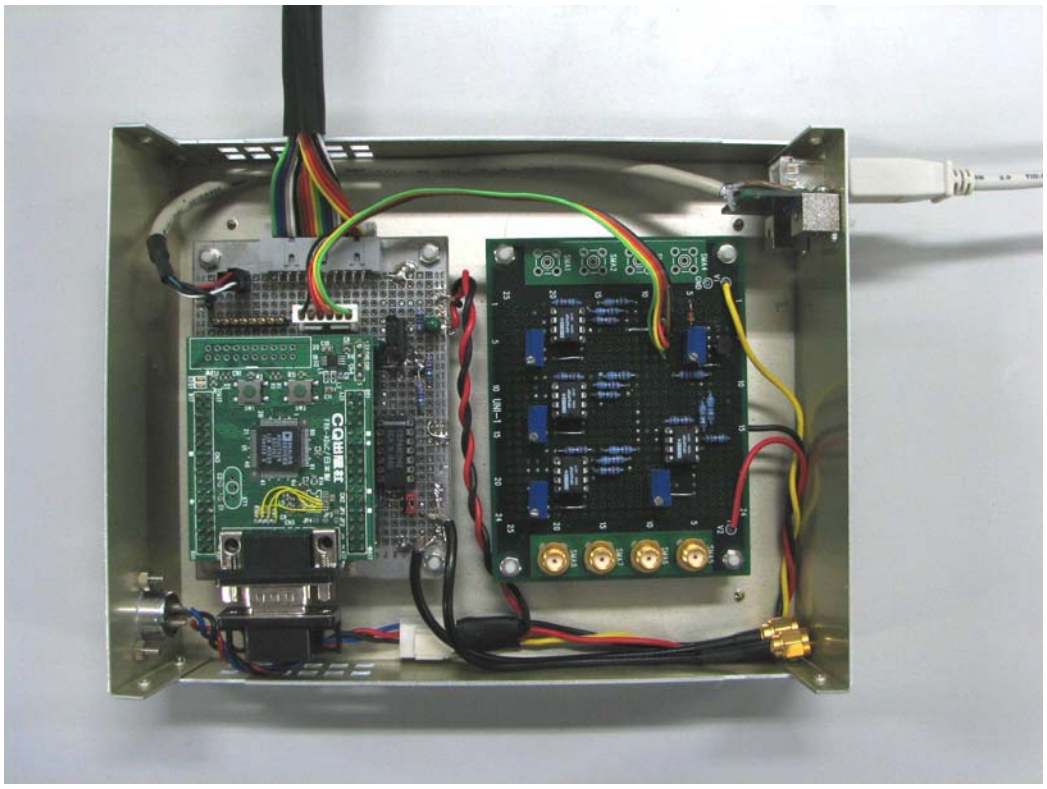

FIG. 2. An overview of the PPG developed in this work. It comprises three printed circuit boards: the RS-232C to USB conversion board (on the left under the microcontroller evaluation board), the microcontroller evaluation board (on the left above the conversion board), and the voltage level converter board (on the right). The PPG is connected to the gradient driver using three SMA connectors shown on the voltage level converter board and connected to the MRI transceiver using an SMA connector and a flat cable.

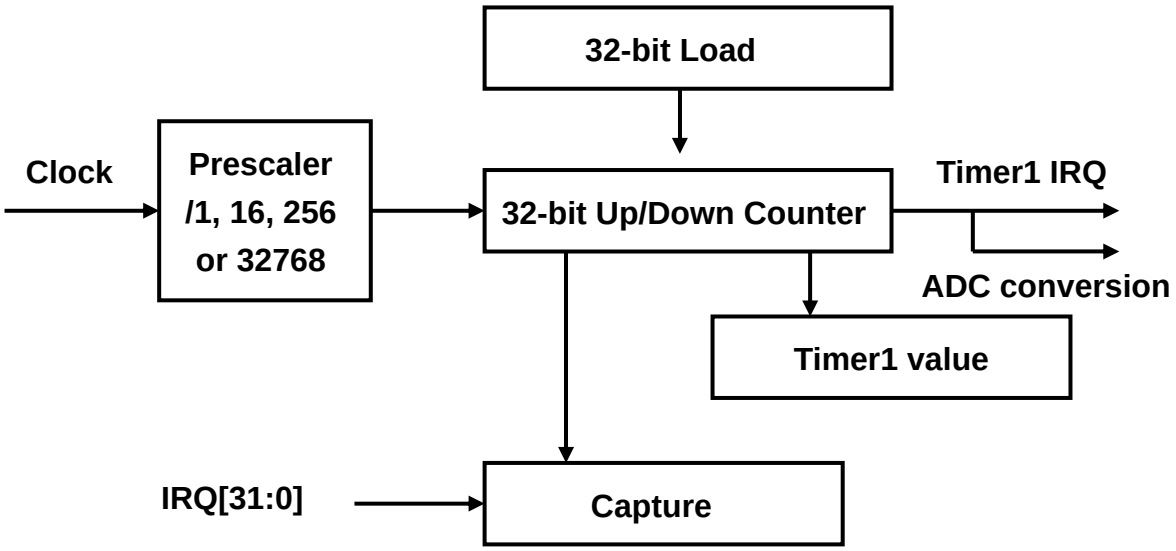

FIG. 3. The block diagram of the timer and interrupt circuit used for timing control of the PPG. The core clock frequency (41.78 MHz) was used for Timer1, which generated an IRQ to the CPU at the end of its count to measure the delay time between two successive PPG events.

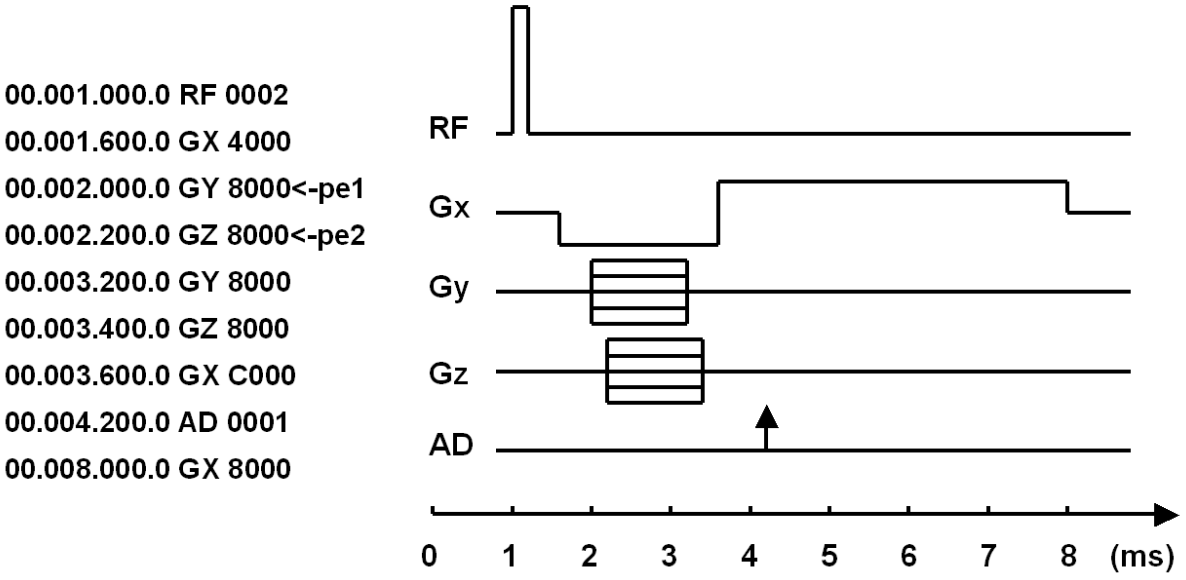

FIG. 4. A pulse sequence timetable (left) and corresponding time chart (right) for a 3D gradient echo imaging sequence developed for our PPG.

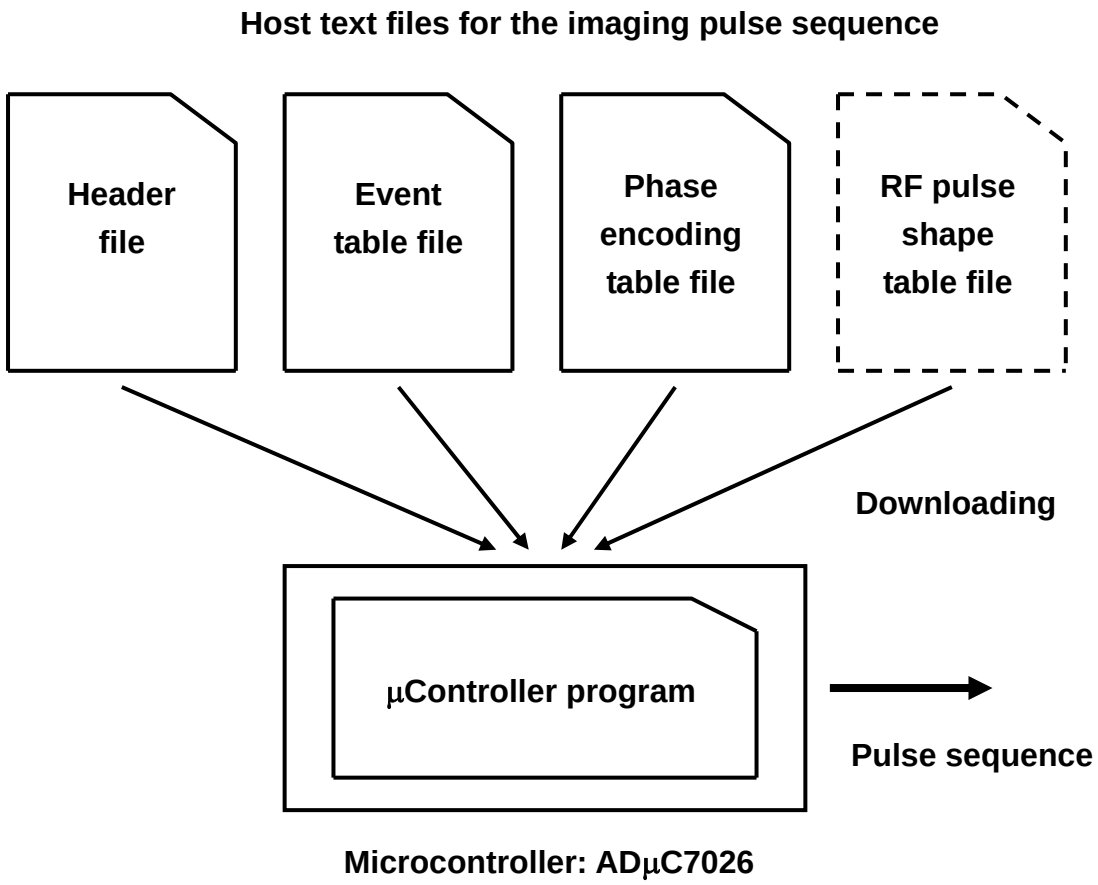

FIG. 5. A functional diagram of the software developed for the PPG. At the start of the PPG operation, text files (header file, event table file, and phase encoding table file) are converted to binary and downloaded to the microcontroller memory, which also contains the embedded program. After the downloading, the target system is reset to start the imaging pulse sequence. (The RF pulse shape file will be implemented in a future version.)

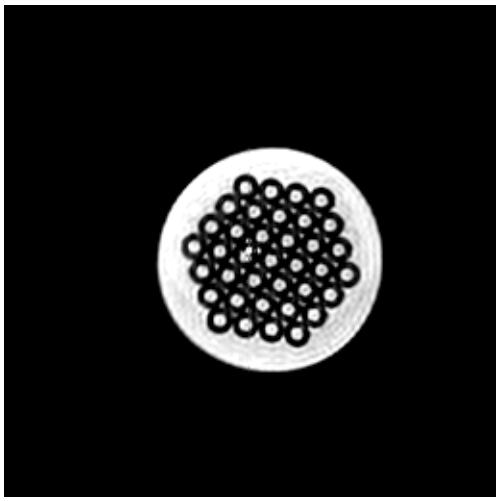

FIG. 6. A 2D MR image of a water phantom acquired via the PPG developed in this work. The image matrix was 128 × 128 and the pixel size was 200 μm × 200 μm.