

A Single-chip Event Sequencer and Related Microcontroller Instrumentation for Atomic Physics Research

E. E. Eyer¹

Physics Department, University of Connecticut, Storrs, CT 06269, USA

A 16-bit digital event sequencer with 50 ns resolution and trigger jitter is implemented by using an internal 32-bit timer on a dsPIC30F4013 microcontroller, controlled by an easily-modified program written in standard C. It can accommodate hundreds of output events, and adjacent events can be spaced as closely as 1.5 μ s. The microcontroller has robust 5 V inputs and outputs, allowing a direct interface to common laboratory equipment and other electronics. A USB computer interface and a pair of analog ramp outputs can be added with just two additional chips. An optional display/keypad unit allows direct interaction with the sequencer without requiring an external computer. Minor additions also allow simple realizations of other complex instruments, including a precision high-voltage ramp generator for driving spectrum analyzers or PZT positioners, and a low-cost PID controller and lock-in amplifier for laser frequency stabilization with about 100 kHz bandwidth.

I. INTRODUCTION

In many atomic and optical physics laboratories, particularly those involving ultracold atoms, there is a need for complex time sequencing of numerous devices on a microsecond time scale. For example, a typical trap loading sequence for cold atoms involves multiple changes to laser controllers, acousto-optic modulators, magnetic field coils, and CCD cameras.¹⁻³ This is difficult to accomplish with conventional pulse generators because numerous independent devices, often ten or more, must be controlled, and usually each device must be switched more than once during the sequence. One approach is to utilize a digital video pattern generator that loads a new 16-bit pattern at each clock increment, but this is very inefficient if the patterns change only occasionally, as is typical of experimental event sequencing. Another is to use a separate timer for each output device, but this quickly becomes expensive and unwieldy if numerous separate instruments are controlled. Also, most commercial pulse generators are unable to produce multiple pulses on any given output unless they form a regular periodic train.

A more logical approach is to use a dynamically programmable timer to set the interval from each event to the next, updating the full set of output data at the end of each interval. Commercial instruments are available that accomplish this with a fast FPGA gate array,⁴ yielding excellent performance and time resolution of up to 10 ns, but these are relatively complex systems typically costing thousands of dollars. This time resolution considerably exceeds the needs of most atomic physics laboratories, making a less capable but simpler solution highly desirable. While a homemade low-cost FPGA design is certainly possible, I have opted instead to use a general-purpose microcontroller. This requires a sacrifice in time resolution (limited to 50 ns in the present case), but it provides an extremely flexible and simple design that is programmed in standard C and can be easily modified by relatively inexperienced users. In its simplest form, a complete instrument can be realized using only a sin-

gle processor chip and a handful of passive supporting components.

More specifically, the designs below use the readily available dsPIC30F3013 or dsPIC30F4013 microcontrollers from Microchip Technology, which are nearly interchangeable except that the 40-pin dsPIC30F4013 has more digital input/output pins than the 28-pin dsPIC30F3013, and a slightly larger complement of internal modules. The event sequencer in Section II requires nearly all of the 40 pins of the larger controller, and uses an innovative timing scheme with one of its internal 32-bit timers to achieve efficient jitter-free event timing. I also briefly describe the design of a small keypad/display unit that can be connected for convenient user interaction with microcontroller-based instruments without requiring an external computer. Finally, I describe minor variations of the microcontroller circuit that can be used as an optical spectrum analyzer controller and as a proportional-integral-derivative (PID) controller for laser locking or other feedback control purposes. Additional designs using the same microcontrollers and display units for other physics-related purposes are described on my web page at the University of Connecticut,⁵ including a phase-locked rf frequency synthesizer and laser offset locking circuit.

II. EVENT SEQUENCER

A. Design

The choice of microcontroller is particularly important if a single-chip design is to be realized. To efficiently accommodate 16 digital output channels, a 16-bit processor is essential. To allow a wide range of timing intervals, at least one internal 32-bit timer is necessary, and it must be capable of generating an interrupt event when the timer register reaches a selected value.

Although inexpensive 16-bit and even 32-bit microcontrollers have now been available for some years, their use for custom applications in physics research is still rela-

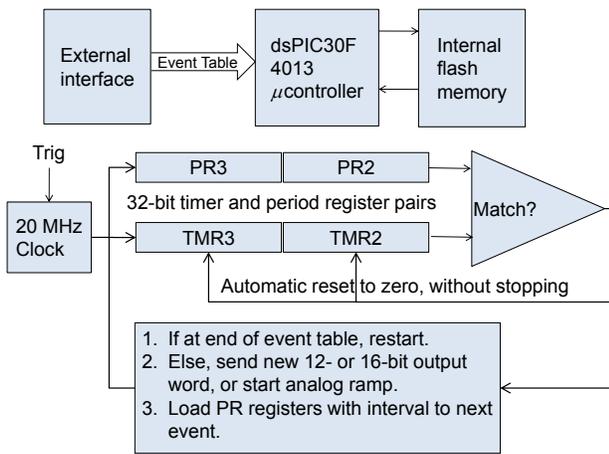


FIG. 1. Flow diagram for the sequence generator. The 32-bit timer is clocked continuously at 20 MHz, and is automatically reset, without stopping, when a matching event occurs.

tively uncommon, perhaps because 8-bit microcontrollers are more familiar and have a wider base of developed software. The dsPIC30F series from Microchip Technology is especially useful because of several very convenient features, a few of them unique:

- They have a full 16-bit instruction set optimized for use with a C compiler, eliminating the need for direct assembly-language programming. It includes a powerful set of fast Digital Signal Processing (DSP) instructions. An excellent compiler and software development environment is free for academic users.⁶
- Despite their low cost (typically \$4 to \$10 USD), these controllers have a full array of internal peripherals such as serial ports, timers, oscillators, and prioritized interrupt controllers. Once programmed, the peripherals operate autonomously from the main processor, so they require no programmed timing loops, and will never miss a clock cycle.
- When operated from a 5 V supply, they are fully compatible with traditional five-volt TTL logic levels, the most common digital signal convention in research laboratory equipment. Operation from 3.3 V or 3.6 V is also possible.
- Unlike many of the newest designs, several dsPIC30F processors are available with 28-pin or 40-pin dual-inline (DIP) packaging, making them convenient for breadboard use and for projects assembled by inexperienced students.

The basic logic of the event sequencer is shown in Fig. 1, and a full schematic diagram in Fig. 2. A 10.000 MHz quartz crystal (ECS 100-18-4X) is used with the built-in oscillator driver and phase-locked-loop (PLL) of the

dsPIC30F4013 to produce a 20 MHz time base. This clock sets the instruction cycle time to 50 ns, and is also used to clock an internal timer comprising registers TMR3:TMR2, used in a hardware-supported 32-bit mode. The sequencer program is initialized by loading a data table from the internal EEPROM flash memory of the dsPIC chip. After waiting for an optional interrupt-driven hardware trigger on the INT2 pin, the program sends out the initial digital pattern, loads the “period register” pair PR3:PR2 with the interval until the next event, and starts the timer. When the timer TMR3:TMR2 reaches the value set in PR3:PR2, it generates an interrupt and automatically resumes counting from a value of zero, without missing a clock cycle. At this point the program must send out the next digital output pattern, then load PR3:PR2 with the next time interval. The key design requirements are to accomplish this process with deterministic execution times, so that the interval between an interrupt and the corresponding digital output never varies, and to do it quickly, so that the period register is loaded before the next event is scheduled to occur. As long as this is done successfully, the timing remains entirely synchronous and independent of program execution. This process is repeated for each time-delay event until the entire output sequence is complete, after which the program can optionally loop back to restart the entire process.

The full C-language source code of the “PatternGen” program is available on the author’s web page,⁵ and occupies just 13% of the 48K program memory. To ensure deterministic program execution, the program is entirely self-contained and free of “housekeeping” tasks, and it assigns to the timer interrupt the highest processor interrupt priority, so it will be serviced immediately even if a lower-priority interrupt event is already in progress. The program is compiled on an external PC, and an inexpensive Microchip PICKit 2 programmer is used to transfer the compiled program to the microcontroller. The programmer uses a USB interface on the PC, and requires only a simple six-pin header on the microcontroller circuit card. The 1K EEPROM flash memory of the microcontroller can store up to 166 six-byte entries, each comprising a 16-bit digital output pattern and a 31-bit time delay count to the next event, specified in units of 50 ns. If more than 166 events are needed, the unused 86% of the reprogrammable internal program memory could be used to store a much larger data table, at the cost of slightly slower execution and some additional complexity in the PatternGen program.

The program includes support for a serial interface that can be used to transfer a new data table from an external computer, and also includes support for a custom keypad/display unit, described below, that can be used to monitor or change the operating status. Since the program is written entirely in standard C it can easily be modified to include other specialized capabilities and interfaces. For example, it would be very easy to include repetitive sub-cycles of arbitrary complexity, or to add

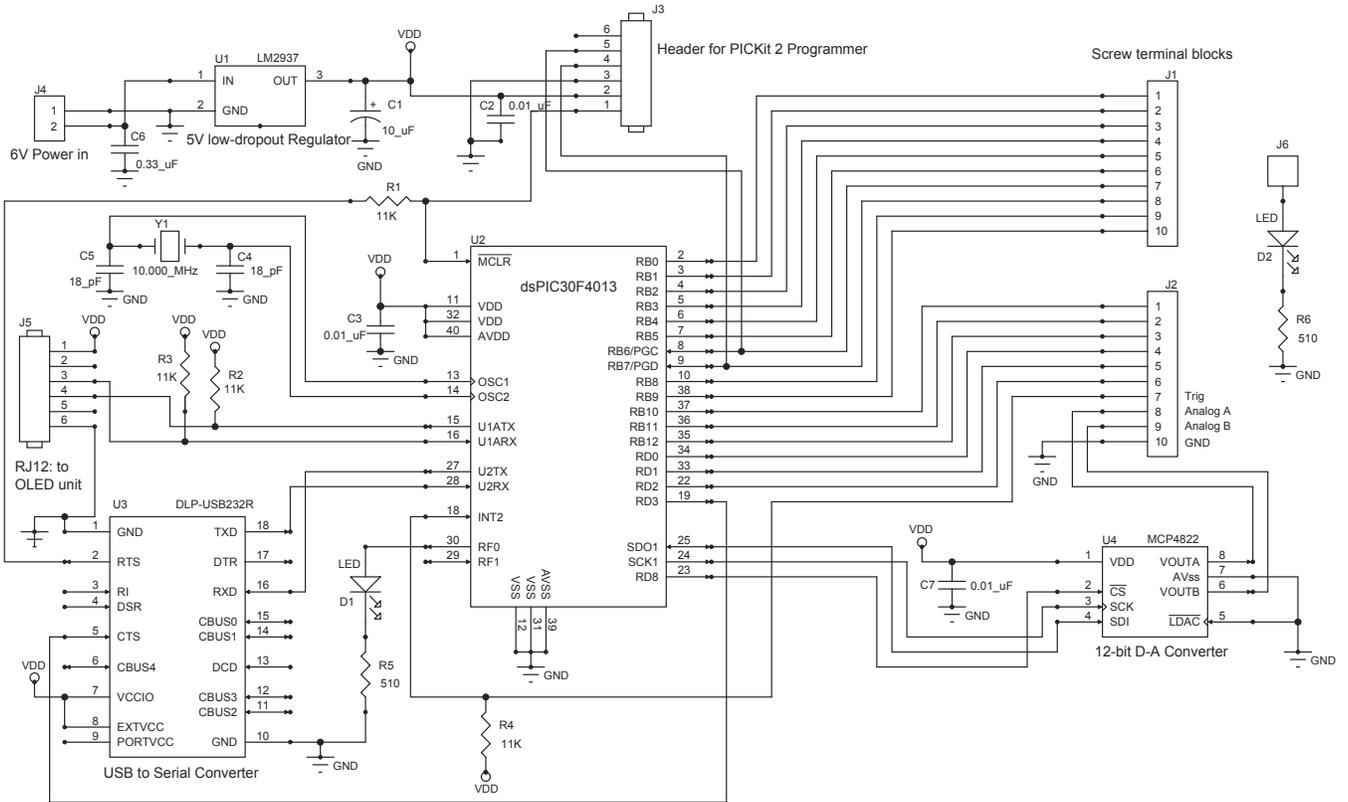


FIG. 2. Schematic diagram of the sequence generator. The USB-to-serial converter and D-A converter are optional. The six-pin header (J3) allows in-circuit reprogramming of the microcontroller using a Microchip PICKit2 programmer.

additional triggering events that will cause the sequence to pause until an external trigger occurs. Even greater flexibility is possible without modifying the circuit itself if the software is modified to alter the functions of the processor pins. For example, any of the first several pins normally used for Port B digital outputs (pins 2-10) can be reconfigured as analog inputs and used with the controller’s built-in 12-bit analog-to-digital converter.

The schematic diagram in Fig. 2 includes both the basic circuit and three optional support components that increase its utility. The basic device consists of the dsPIC microcontroller plus a quartz crystal, a six-pin programming header, two connectors for the I/O signals, and a few capacitors. The first supporting component is a low-dropout voltage regulator that allows operation from a semi-regulated 6 V or 9 V wall-plug power supply. The second is an inexpensive two-channel 12-bit digital-to-analog converter (DAC), Microchip MCP4822, which provides a pair of analog outputs. The DAC is supported in the PatternGen program by using a special flag bit in the data table entries, the high bit of the time interval word, to indicate that the entry is actually a special 12-byte event definition. One such “special event” allows direct analog outputs and another initiates an analog ramp, in which interrupts from timer TMR1 of the dsPIC trigger a periodic series of DAC updates. This ramp can proceed concurrently with new digital output

events, as can be seen in Fig. 3. Finally, the schematic diagram incorporates a small USB-to-serial converter module, DLP Design model DLP-USB232R, that allows USB communication with external computers. While the circuit can be successfully operated even on a breadboard, a convenient 6 × 9 cm printed circuit board has been designed that accommodates these components together with screw terminals blocks for the output pins, two indicator LEDs, and a connector for the optional keypad/display unit. Design information is available on the author’s web page.⁵

B. Sequencer Performance

Figure 3 shows an oscilloscope recording of two channels from a typical event sequence, one digital and the other analog. The analog outputs have a full scale of 4.096 V, with a typical temperature coefficient of 45 ppm/°C and a resolution of 1 mV determined by the 12-bit DAC range.

The time base accuracy of the sequencer is determined by the 10 MHz quartz crystal oscillator comprising Y1, C4, and C5 in Fig. 2, together with the internal PLL of the dsPIC. By slightly adjusting the value of capacitor C4 within the range of 15-30 pF, the clock frequency can

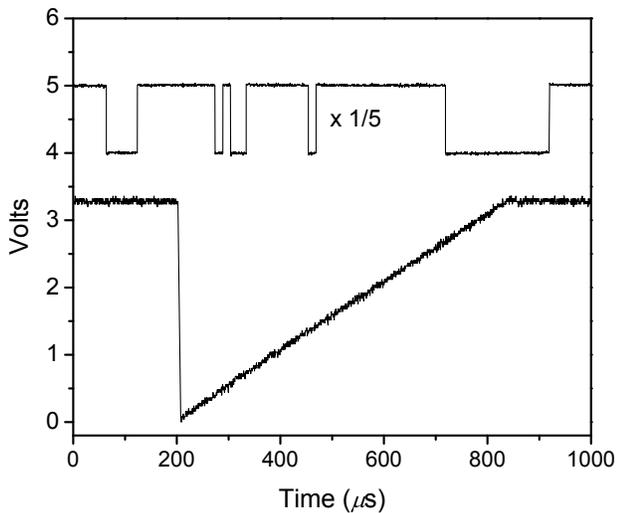


FIG. 3. Oscilloscope data, showing a complex digital pattern on one of the 16 digital output lines, together with a concurrent analog ramp from one of the two DAC outputs. The upper (digital) trace is reduced by a factor of five and shifted vertically for clarity.

be tuned to an accuracy of about 20 ppm, and it is typically stable to several ppm over a period of months. The clock jitter is negligible — the manufacturer’s specifications imply that the PLL jitter does not exceed 0.083 ns for the frequencies used in this design.⁷ The time resolution of the sequencer is equal to the instruction cycle time, which 50 ns for the present design, or 33.3 ns for the maximum dsPIC30F instruction clock speed of 30 MHz. The trigger jitter is one clock cycle, since an external trigger would usually be asynchronous with the instruction clock. However, subsequent time intervals accumulate no additional error from jitter. They also have no measurable jitter relative to one another, since the timer interrupt request is synchronous with the processor clock, and the interrupt latency for the dsPIC30F microcontrollers is exactly four instruction cycles except under unusual circumstances.⁸ The present version of the PatternGen program does not include any of the instructions that could cause this non-deterministic interrupt response to occur. Even if such instructions were added, the effect would be only to delay a single event by 50 ns, with no cumulative error in the overall sequence.

Another important consideration is the minimum possible interval between successive events. This is determined by the time needed to process the timer interrupts, which includes retrieving and writing the next data output, reloading the PR3:PR2 period register, and saving and restoring the registers needed to allow resumption of the interrupted program activity. To make sure that this time is minimized, I have closely examined the assembly code generated by the C compiler for the interrupt processor and fine-tuned the program code. Table I shows the resulting delay times for each type of event supported by the present program, based on direct measurements

TABLE I. Times required for interrupt processing for digital output patterns and various analog outputs.

Event type	Time (μs)
16-bit digital pattern	1.8
13-bit digital pattern (without Port D)	1.5
1-Channel DAC output	3.1
2-Channel DAC output	6.7
DAC ramp initiation	3.5

with an oscilloscope. For 16-bit digital outputs the minimum time between events is 1.8 μs , which is reduced to 1.5 μs if 13-bit outputs are used. This is because a 13-bit output can be achieved with a single write instruction to Port B of the microcontroller, while a 16-bit output requires separately writing the three highest bits to Port D. As shown in the table, analog outputs take considerably longer because of the time needed to write a 16-bit command word to the external DAC chip through an SPI serial interface. Surprisingly, the initiation of an analog ramp, rather than a static analog value, requires very little additional time. This is because the ramp startup requires only a series of write commands to internal registers, which start an additional internal timer that clocks subsequent ramp updates.

III. RELATED INSTRUMENTATION

A. Detachable display/keypad Unit

To provide a compact keypad/display unit that can be used interchangeably with many different microcontroller-based circuits, a unit was designed combining a 16-button keypad (All Electronics model KP-23) with a small color OLED display module (4D Systems $\mu\text{OLED-160-G1SGC}$).⁹ They are housed together in an 8 x 15 cm plastic enclosure, Bud HH-3600-B, intended for handheld use. The display can accommodate up to 16 lines of text or a mixture of text and graphics, the latter especially useful for displaying error signals in laser frequency locking applications. A dedicated microcontroller (Microchip PIC24FJ16GA002) is used to scan and debounce the keypad and to multiplex together the display and keypad signals into a single serial port. This is a 16-bit controller that is code-compatible with the dsPIC30F series, except that it is lacking a DSP unit. The schematic diagram and printed circuit board design, as well as photographs of the device, are available at the author’s website. The unit is interfaced to stand-alone microcontrollers via a 4-wire connection (power, ground, transmit, and receive) through a standard RJ-12 telephone jack. An example of this interface appears as connector J5 in Fig. 2.

A simple but flexible parameter editor for this unit has been written that runs as part of the main program on the host instrument’s microcontroller. It allows the user

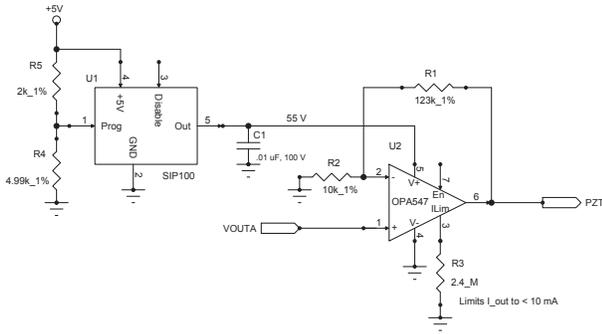


FIG. 4. Schematic diagram for the additions to Fig. 2 needed to realize a spectrum analyzer/PZT driver.

to display and modify control parameters appropriate to each application, such as voltage ranges, time constants, and triggering modes. Numerical values can be modified either by direct entry or by up-down arrow keys. An auto-scaling graphics function allows real-time display of data with a resolution of up to 128×160 points, determined by the OLED display resolution.

B. Controller for spectrum analyzer or PZT

An example of a major change in function with minimal hardware additions is shown in Figs. 4 and 5. This instrument is a flexible and accurate high-voltage ramp generator designed for use with an optical spectrum analyzer, although it can equally well drive other PZT-based devices such as a micropositioner or a tuning grating on an external-cavity diode laser. The only required hardware changes are the addition of a miniature high-voltage power supply and a high-voltage operational amplifier that can tolerate the capacitive loading of a PZT. The Texas Instruments OPA547 amplifier shown in Fig. 4 is reasonably inexpensive and can drive large currents for rapid PZT response (0.5 A), but is limited to a maximum supply voltage of 60 Volts. An alternative is the remarkable Apex PA240, which can drive up to 60 mA continuously with a maximum supply voltage of 350 Volts, yet is still fairly affordable. In either case the output of the MCP4822 DAC in Fig. 2 is used to provide the ramp, which is scaled up by a standard non-inverting operational amplifier configuration.

The software changes to produce an accurate repetitive ramp are more fundamental, but easily implemented. As shown in Fig. 5, counter TMR1 is used to set an period between regularly-spaced interrupts. The interrupt processor updates the DAC, using a variation of the Bresenham line-drawing algorithm¹⁰ to provide the smoothest possible ramp between the selected endpoints without requiring time-consuming floating-point calculations. Counter TMR2 is used to generate a fixed-width synchronization output pulse each time the ramp starts. All of the operating parameters (starting voltage, ending

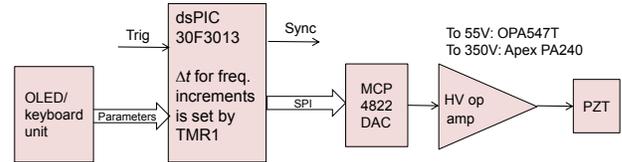


FIG. 5. Logic for the spectrum analyzer driver. Extensive use is made of the display/keypad unit for parameter display and entry, although the unit can operate autonomously once it has been configured.

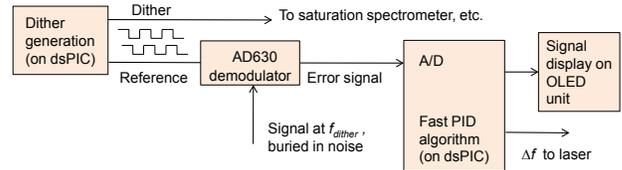


FIG. 6. Scheme for dither-based laser locking. The dsPIC timers are used to generate a dither signal and a phase-shifted reference synchronized to it, and the digital signal processing (DSP) features of the microcontroller are used for a fast PID control loop. The basic loop requires just 30 instruction cycles ($1.5 \mu\text{s}$ at 20 MHz).

voltage, scanning rate, etc.) can be easily changed using the detachable keypad/display unit. Once entered, they are stored in EEPROM memory, and will be recalled even if the keypad unit is disconnected and the instrument is powered down.

This design, which I have constructed using the smaller 28-pin dsPIC30F3013 variant of the microcontroller, has been used extensively with a Thorlabs SA200 spectrum analyzer. The overall performance is comparable to that of a specialized commercial spectrum analyzer controller that is also physically much larger. The 4096-step range of the 12-bit DAC is well-matched to the optical finesse of about 200. For applications requiring higher resolution, a 14-bit or 16-bit DAC could easily be substituted.

C. Diode laser frequency locking

Another useful variation is shown in block form in Fig. 6. This is a control system intended mainly for dither-based locking¹¹ of single-frequency diode lasers to an atomic resonance line observed in Doppler-free saturated absorption spectroscopy.¹² The microcontroller timers are now used to generate a square-wave dither waveform with a typical period of 10-100 μs , used to place a corresponding small frequency modulation or “dither” on the laser. A phase-shifted version of the same waveform is used as a reference signal for phase-sensitive lock-in detection of the change in resonant atomic absorption.

A simple lock-in detector can be implemented by using the reference waveform internally to the microcontroller to trigger successive analog-to-digital conversions

with its built in ADC, which are then subtracted pairwise to obtain the phase-discriminated signal. However, the dynamic range is severely limited by the 12-bit range of the ADC. A much better approach is to use a separate audio-frequency demodulator such as the Analog Devices AD630, which can recover signals buried in more than 100 dB of noise, comparable to a good commercial lock-in amplifier.

To provide frequency corrections based on the resulting error signal, a configurable proportional-integral-differential controller is implemented using the DSP features of the microcontroller. Using sample code available from Microchip Technology, a basic PID control loop can be implemented with a cycle time of just 30 instruction cycles. If the processor is operated at 20 MHz this corresponds to an update time of 1.5 μ s, which is actually faster than necessary to accommodate the operating range of the AD630 (up to about 100 kHz for good performance). Just as for the other instruments described here, the detachable keypad/display unit can be used to adjust the parameters and display the error signal and locking status. Alternatively, a serial or USB interface to an external computer can be used.

IV. CONCLUSIONS

Because modern microcontrollers have 16-bit or even 32-bit data paths and contain a wide array of self-contained peripheral devices, instruments of surprising complexity can be assembled with just a few chips. The design process is mainly a matter of producing the required software, and even this is made much easier by the availability of efficient C compilers. The timing sequencer described in this paper relies on an unusual application of an internal 32-bit timer and its “period register” to generate microsecond-scale timings on up to 16 digital outputs with negligible jitter. Minor additions to the sequencer circuit, together with changes to the software, also form the basis of two quite different instruments: a high-voltage PZT/spectrum analyzer driver and a flexible PID controller. These devices can be connected to

external computers for control, but they can also be operated in a stand-alone mode by connecting a portable keypad/display unit, which can be detached once new parameters values are set. Full design information and C-language source code for these instruments are available on the author’s web page, which is periodically updated as improvements become available.⁵ The simplicity and remarkable flexibility of these devices, together with their very low cost, makes them attractive alternatives to specialized commercial electronics.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation and the University of Connecticut Research Foundation.

- ¹W. Petrich, M. H. Anderson, J. R. Ensher, and E. A. Cornell, *J. Opt. Soc. Am. B* **11**, 1332 (1994).
- ²M. Prevedelli, F. S. Cataliotti, E. A. Cornell, J. R. Ensher, C. Fort, L. Ricci, G. M. Tino, and M. Inguscio, *Phys. Rev. A* **59**, 886 (1999).
- ³M. D. Barrett, J. A. Sauer, and M. S. Chapman, *Phys. Rev. Lett.* **87**, 010404 (2001).
- ⁴For example, the PulseBlasterUSB and related instruments are produced by SpinCore Technologies, <http://www.spincore.com>.
- ⁵E. E. Eyler, web page at <http://www.phys.uconn.edu/~eyler/microcontrollers/>, *Microcontroller Designs for Atomic, Molecular, and Optical Physics Laboratories*, University of Connecticut Physics Dept.
- ⁶MPLAB C Compiler for PIC24 MCUs and dsPIC DSCs, version for academic use, available as a free download from Microchip Technology, Inc., <http://www.microchip.com>. The version used for this work (3.12) includes full optimization features.
- ⁷Microchip Technology Inc., *dsPIC30F3014/4013 Data Sheet*, p. 180 (Document DS70138F, 2008, available at <http://www.microchip.com>).
- ⁸Microchip Technology Inc., *dsPIC30F Family Reference Manual*, Section 6.3.4 (2004 edition, available at <http://www.microchip.com>).
- ⁹Manufactured by 4D systems, <http://www.4dsystems.com.au/>.
- ¹⁰J. E. Bresenham, *IBM Systems Journal* **4**, 25 (1965).
- ¹¹M. F. Stone, M. Stone, M.S. Thesis, SUNY-Stony Brook (2001).
- ¹²M. D. Levenson and S. S. Kano, *Introduction to Nonlinear Laser Spectroscopy, Revised Ed.*, Academic Press, pp. 79-127 (1982).