

Microcontroller-based servo for two-crystal X-ray monochromators

D. P. Siddons

National Synchrotron Light Source, Brookhaven National Laboratory, Upton, NY 11973, USA. E-mail: siddons@bnl.gov

(Received 4 August 1997; accepted 18 November 1997)

Microcontrollers have become increasingly easy to incorporate into instruments as the architectures and support tools have developed. The PIC series is particularly easy to use, and this paper describes a controller used to stabilize the output of a two-crystal X-ray monochromator at a given offset from its peak intensity position, as such monochromators are generally used.

Keywords: microcontrollers; two-crystal X-ray monochromators; PIC series.

1. Introduction

X-ray monochromators often use two separate crystals in a mechanism which maintains their angular position with high precision (typically less than 1 arcsec). In general, these mechanisms lack long-term stability, particularly under the varying heat load encountered in synchrotron radiation operations. Usually some type of feedback system is necessary if the monochromator output is to be sufficiently stable. Many such systems are in existence (representative examples have been reported by Krolzig *et al.*, 1984; Bridges, 1987; Ramanathan *et al.*, 1987) ranging from simple analog circuits driving a piezoelectric transducer to complex systems involving computers and multiple nested loops. The problem which the present device attempts to solve arose from the need to control a system where the primary Bragg-angle control of the second monochromator crystal is by means of a stepping motor, rather than a piezo transducer. In this case a simple analog system is not appropriate, but the monochromator in question is intrinsically already quite stable, so a simple system would be adequate, and indeed desirable. The PIC16C74 (Microchip Technology Incorporated, Chandler, Arizona, USA) provides a range of built-in resources which make it ideal for such a project. In addition, there exist highly optimized C-language compilers targeted to this family of devices (PCB and PCM, PIC C compilers, Custom Computer Services, Inc., PO Box 2452, Brookfield, WI 53008, USA). Thus, a low-component-count system can be built readily.

2. System description

In order to make a useful self-contained device, the final system should contain the following.

- (i) A text display to provide diagnostics and system information to the user.
- (ii) Some input keys to control the basic functions of the system.
- (iii) Some method of communicating with the beamline control computer to allow coordination with the data-acquisition functions.

(iv) An analog input capability to sense the monochromator output.

(v) A way of compensating for storage-ring beam decay.

The PIC16C74 provides a wealth of input–output (IO) capabilities, including multi-channel analog–digital convertors (ADC), up to 32 bits of digital IO, and pulse-width modulated analog outputs. The built-in ADCs have only 8 bit resolution, so it was decided to add a separate 12 bit ADC chip to the system. A serial-output device was chosen (the LTC129; Linear Technology Corporation, 1630 McCarthy Boulevard, Milpitas, CA 95035, USA), since it requires only 4 bits of IO to interface to it. This device provides a two-input multiplexor, giving additional flexibility. The PIC has an on-chip UART, which can be configured to be an interrupt-driven communication channel. External level convertors are required to make a true RS232-compatible interface. The MAX232 device (obtainable from many suppliers, *e.g.* Digi-Key, 701 Brooks Avenue, South Thief River Falls, MN 56701-0677, USA) conveniently provides that function. Text display is provided by a 4 line \times 20 character liquid-crystal display (LCD) module which has an intelligent interface requiring only 6 IO bits (the Optrex DMC-20481NY-LY-B is also available from several sources, including Digi-Key above). User control is effected by five push buttons, requiring another 5 IO bits. The user can change the set point of the servo, and can move the control motor at one of two speeds. The user can also disable the servo loop to facilitate initial set-up.

The serial IO channel is used to provide a simple method of tuning the servo system parameters; parameters which will not require user adjustment and so do not merit front-panel controls. It could also be used to inform the beamline computer of any monochromator malfunction to prevent data collection at such times.

The value of the storage-ring beam current at NSLS is made available to users in the form of a pulse train, the repetition rate of which is proportional to the stored current. Since most data-acquisition systems include some method of counting pulses, this is a simple way for experimenters to acquire this parameter. The servo needs to scale the observed intensity from the monochromator with the beam current, so that the position on the rocking-curve to which the servo is locked does not change as the beam decays. This is conveniently achieved by this system, since the PIC16C74 has a period-measurement capability built-in. Thus, the scaling can be accomplished by simply multiplying the observed signal with the period of the beam-current monitor. This

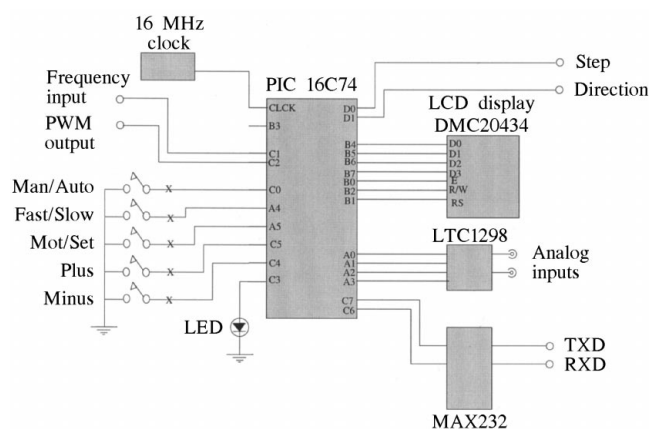


Figure 1
Schematic diagram of the servo system.

avoids the need to make a division, which is slower than multiplication. Although the current system does not do so, it would be straightforward to implement a non-linear relationship between beam current and output intensity. Fig. 1 shows a schematic of the system.

In the intended application, the Bragg-angle control is effected by a Burleigh Inchworm linear translator (Burleigh Instruments Inc., Burleigh Park, Fishers, NY 14453, USA) but it can be applied to any system driven in a stepwise manner.

The system is programmed in the C programming language, using a commercial C compiler specifically designed for use with the PIC family of processors. As such it performs substantial optimization of the code, and makes extremely efficient use of the memory resources available. Fig. 2 shows a flow-chart for the control system.

When the servo is disabled, the system loops through, reading the inputs, updating the display and checking for front-panel button activity. If such activity is detected, appropriate action is taken. No attempt is made to debounce the switch inputs. Instead, if a button is detected as being pushed, the system inserts a 100 ms

delay after completing the requested action. Thus, if the button is held down, the action is repeated roughly ten times per second.

When the servo is enabled, the system loops through, this time calculating the PID loop equation, making the demanded move and then entering the front-panel loop as described above. This sequence permits the user to adjust the setpoint and motor position in real time, even when the servo is active. The motion permitted per loop is limited to 100 steps, primarily to exercise some control over the loop timing. Since the maximum step rate for the motor we use is 1000 s^{-1} , long moves will significantly slow down the loop. This is damaging to the loop stability, so we need to control it.

Tuning the feedback loop is effected by changing the parameters $C1-C4$ from their default values *via* commands through the serial port. This port operates in an interrupt-driven mode *via* a pair of buffers. Received characters are stored in the input buffer until a carriage-return character is received. The receive interrupt service routine sets a flag which the main loop checks every time through. A simple-minded interpreter decodes the instructions and modifies system variables as required. The flag is then reset and normal looping resumes. Commands are either to set a variable, or to send its current value out over the serial port. Such responses are made by filling the output buffer and signalling the interrupt system to start sending. After transmission is complete, the transmit interrupt is disabled.

The C code to implement this algorithm comprises about 800 lines, and occupies about 75% of the on-chip resources (memory and IO). Code listings can be obtained from the author on request.

The speed of the update cycle is controlled by the maximum stepping rate of the control output. In the present case the motor was limited to a maximum of 1000 steps s^{-1} . Typical position corrections were around 10 steps, causing a 10 ms delay. Additional significant delay comes from the three-term algorithm arithmetic. Future processors in this series will include a hardware multiply capability, which will remove that from the limiting factors. Further delays arise from the display updating process, because the display is a CMOS device and rather slow. The net result is a cycle time of around 50 ms on average. Under these conditions, the step response time for a 20% change in setpoint is around 0.5 s. For our application, where the problem is mainly slow drifts, this is more than adequate. Fig. 3 shows a photograph of the prototype device.

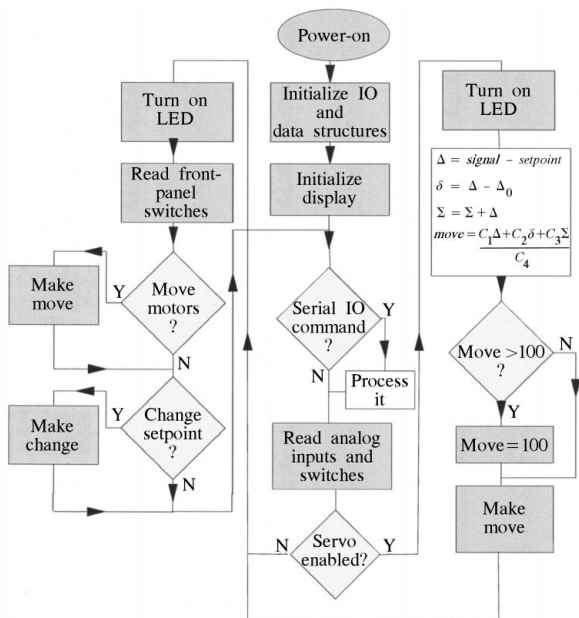


Figure 2
Flow diagram showing the logic implemented by the microcontroller.

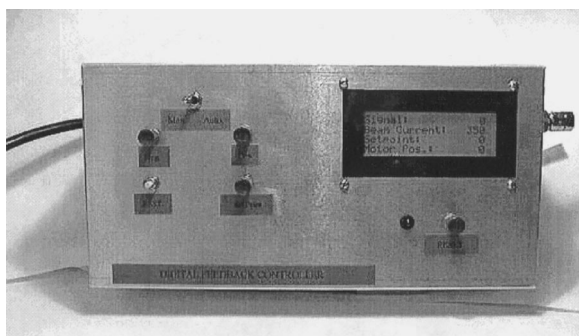


Figure 3
Photograph of the prototype unit.

3. Conclusions

A digital feedback system has been developed which satisfies the need for a simple solution for monochromators controlled by stepping motors.

The mention of trade names and proprietary devices in this article does not imply any endorsement of any product by the author. NSLS is funded by the Department of Energy under contract number DE-AC02-76CH00016.

References

- Bridges, F. (1987). *Nucl. Instrum. Methods Phys. Res. A*, **257**, 447–450.
- Krolzig, A., Materlik, G., Swars, M. & Zegenhagen, J. (1984). *Nucl. Instrum. Methods Phys. Res. A*, **219**, 430–434.
- Ramanathan, M., Engbretson, M. & Montano, P. A. (1987). *Nucl. Instrum. Methods Phys. Res. A*, **266**, 471–474.