MX: A Portable Data Acquisition and Control Toolkit

William M. Lavender

Biological, Chemical, and Physical Sciences Department
Illinois Institute of Technology
Chicago, Illinois 60616 USA

http://www.imca.aps.anl.gov/mx/
lavender@icarus.csrri.iit.edu

What is MX?

- MX is a portable beamline control system.
- It includes a set of several programs and libraries.

  MX server and MX update daemons manage the hardware.

  Supports both Tcl/Tk GUI applications and command line based applications.

- Device drivers and core functionality are encapsulated in the *libMx* library.
- Both servers and clients use the same *libMx* library and can use the same device drivers.
- Client-server operation is not required. This simplifies the debugging of device drivers.
- MX has its own TCP/IP protocol, but most of MX is independent of the protocols used.
- Clients can use foreign protocols exclusively, or a mix of MX and foreign protocols.

Current and Pending MX Users

- IMCA-CAT, APS *(protein crystallography)*
- MRCAT, APS *(materials science)*
- BioCAT, APS *(biophysics)*
- DND-CAT, APS *(materials science)*
- SBC-CAT, APS *(protein crystallography)*
- SER-CAT, APS *(protein crystallography)*
- MBC, ALS *(protein crystallography)*
- X3, NSLS *(small molecule crystallography)*
- IIT BCPS Dept. *(powder diffraction)*

Why I Wrote MX

- Beamline control software packages have tended to be environment specific and non-portable.

- Many are centered around a particular interface method such as CAMAC or VME.

- EPICS seemed to be a less than ideal choice, since it requires a rather high level of expertise to setup, maintain, and develop for.

- Users sometimes bring in their own equipment that they want to interface to the beamline.

MX Control System Goals

- Create a uniform and portable interface to motors and other devices.

- Encapsulate the core functionality in libraries like *libMx*.

- Make client-server functionality available.

- Require that *libMx* know very little about the application programs invoking it.

- Localize the knowledge *libMx* has of device hardware to driver functions with well defined interfaces.

MX Control System Goals (cont.)

- Emphasize operating system independence.
- Require only ANSI C and either BSD sockets or Winsock.
- Be able to interoperate with already existing beamline control systems.
- Try to minimize built-in assumptions about the structure of future data acquisition APIs.
- Do not depend on the presence of software that may not be freely distributed.
- Make the source code freely available.

MX Platforms

Linux 2.x (i386, m68k)	-	GCC
SGI Irix 6.3, 6.5	-	SGI C compiler
Solaris 2.5, 2.6	-	SPARCompiler C, GCC
HP/UX 10.20	-	HP/UX C compiler
SunOS 4.1.4	-	SPARCompiler C
Windows 95, 98, NT 4	-	Visual C++ 4, 5, or 6
MSDOS 6.2	-	DJGPP 2.0.2 & Watt32

MX Record Types

MX Interfaces

RS-232, GPIB, CAMAC, Port I/O, Generic

MX Devices

Motor, Scaler, Timer, Amplifier, Digital I/O, Analog I/O, MCA, MCS, Table

MX Variables

Local, MX, EPICS

MX Scans

Motor scan, Scaler scan, $\theta - 2\theta$ scan, Quick scan, XAFS scan, Slit scan, List scan

MX Servers

MX TCP-IP server

Current MX Motor Drivers

Am9513-based controller
APS undulator (via EPICS)
Bruker D8 (prerelease)
Compumotor 6K and 6000
DAC motor
Delta Tau PMAC
DSP Technology E500
IMS Panther HI/HE
Joerger SMC-24
MX network motor
National Instruments ValueMotion
Newport MM-3000
Newport MM-4000/4005
OMS VME-58 (via EPICS or m68k Linux)
Physik Instrumente E-662 piezo controller
SCIPE motor
Soft motor
Velmex VP9000
XIA HSC-1 slit controller

In addition, 12 different kinds of pseudomotor drivers have been written.

MX Pseudomotor Drivers

Average position
Elapsed time
Linear combination of motors
Monochromator
Position difference
Slit center
Slit width
Table rotation or translation
X-ray energy
X-ray wavelength
X-ray wavenumber
XAFS wavenumber (k)

Other MX Drivers

Scalers and Timers

Am9513, DSP Technology QS450,
Joerger VSC-8/16 (via EPICS),
Kinetic Systems 3610, Multichannel scaler,
MX network, EG&G Ortec 974, SCIPE, Soft

Amplifiers

Keithley 428, MX network, Soft

Digital I/O

Intel 8255 (mode 0), Kinetic Systems 3063,
Motorola MC6821, MX network,
National Instruments LabPC+, Soft

Analog I/O

Kinetic Systems 3112 & 3512, MX network,
National Instruments LabPC+

Multichannel Analyzers

EG&G Ortec Trump MCA

Multichannel Scalers

Struck or SIS 3801 (via EPICS)

Tables

APS Sector 17 ADC table

Example: Cu K-edge Scan

```
#include <stdio.h>
#include "mx_rec.h"
#include "mx_motor.h"
#include "mx_scalr.h"
#include "mx_timer.h"

int main( int argc, char *argv[] )
{
    MX_RECORD *energy_motor, *i_zero, *i_trans, *timer1;
    MX_RECORD *record_list;
    double energy, start, step_size, integration_time;
    int i, num_steps, busy;
    long i_zero_value, i_trans_value;

    mx_setup_database( &record_list, "example.dat" );
    energy_motor = mx_get_record( record_list, "energy" );
    i_zero       = mx_get_record( record_list, "I0" );
    i_trans      = mx_get_record( record_list, "It" );
    timer1       = mx_get_record( record_list, "timer1" );
    start = 8950; step_size = 1.0; num_steps = 101;
    integration_time = 1.0;
    for ( i = 0; i < num_steps ; i++ ) {
        mx_scaler_clear( i_zero );
        mx_scaler_clear( i_trans );
        energy = start + step_size * (double) i;
        mx_motor_move_absolute( energy_motor, energy, 0 );
        mx_timer_start( timer1, integration_time );
        busy = TRUE;
        while ( busy ) {
                mx_timer_is_busy( timer1, &busy );
                mx_msleep(100);
        }
        mx_scaler_read( i_zero, &i_zero_value );
        mx_scaler_read( i_trans, &i_trans_value );
        printf( "%10.3f  %10lu  %10lu\n",
                energy, i_zero_value, i_trans_value );
    }
    exit(0);
}
```

MX Driver API Examples

```
typedef struct {
        mx_status_type ( *motor_is_busy )( MX_MOTOR *motor );
        mx_status_type ( *move_absolute )( MX_MOTOR *motor );
        mx_status_type ( *get_position )( MX_MOTOR *motor );
        mx_status_type ( *set_position )( MX_MOTOR *motor );
        mx_status_type ( *soft_abort )( MX_MOTOR *motor );
        mx_status_type ( *immediate_abort )( MX_MOTOR *motor );
        mx_status_type ( *positive_limit_hit )( MX_MOTOR *motor );
        mx_status_type ( *negative_limit_hit )( MX_MOTOR *motor );
        mx_status_type ( *find_home_position )( MX_MOTOR *motor );
        mx_status_type ( *constant_velocity_move )( MX_MOTOR *motor );
        mx_status_type ( *get_parameter )( MX_MOTOR *motor );
        mx_status_type ( *set_parameter )( MX_MOTOR *motor );
} MX_MOTOR_FUNCTION_LIST;

typedef struct {
        mx_status_type ( *clear ) ( MX_SCALER *scaler );
        mx_status_type ( *overflow_set ) ( MX_SCALER *scaler );
        mx_status_type ( *read ) ( MX_SCALER *scaler );
        mx_status_type ( *read_and_clear ) ( MX_SCALER *scaler );
        mx_status_type ( *scaler_is_busy ) ( MX_SCALER *scaler );
        mx_status_type ( *start ) ( MX_SCALER *scaler );
        mx_status_type ( *stop ) ( MX_SCALER *scaler );
        mx_status_type ( *get_mode ) ( MX_SCALER *scaler );
        mx_status_type ( *set_mode ) ( MX_SCALER *scaler );
        mx_status_type ( *set_modes_of_associated_counters )
                                        ( MX_SCALER *scaler );
} MX_SCALER_FUNCTION_LIST;
```

Future Plans

- Asynchronous event handling.
- Performance and concurrency improvements.
- Development of more application specific GUIs.
- Create servers to export MX services to other control systems. *(EPICS, CORBA, ...)*
- Export MX APIs to more languages. *(Python, C++, Java, web servers, ...)*
- SER-CAT, MBC, and SBC-CAT are collaborating on the development of MX-based systems that use the Delta Tau PMAC controller.
- GUI database configuration tool.
- GUI server control tool.

Summary

- MX is now in use at 4 APS beamlines and in X-ray labs at IIT.

- Soon will be used at another 8 beamlines at APS, ALS, and NSLS.

- IMCA-CAT users have stated that the protein crystallography GUIs are among the best they have seen.

- We are interested in collaborating with other groups on the further development of MX.

Acknowledgements

A. Howard	-	IMCA-CAT, IIT
J. Chrzas	-	IMCA-CAT, IIT
L. Keefe	-	IMCA-CAT, IIT
C. Segre	-	MRCAT, IIT
B. Bunker	-	MRCAT, Notre Dame
N. Leyarovska	-	MRCAT, IIT
H. Tostmann	-	MRCAT, Univ. of Florida
G. Bunker	-	BioCAT, IIT
R. Fischetti	-	BioCAT, IIT
J. Fait	-	SER-CAT, Univ. of Georgia
G. Rosenbaum	-	SER-CAT, Univ. of Georgia
J. Quintana	-	DND-CAT, Northwestern
E. Westbrook	-	MBC