User's Manual

Model MX190

API for the MX100/DARWIN First Step Guide

Using the API to create programs for the MX100

Please read this manual before creating any MX100 programs with the API.

This manual provides some basic information about the MX100 for creating MX100 programs. For detailed information about the API and MX100, refer to the manuals in the table below. Each manual is included on the CD-ROM that came with its respective product.

Manual Name	Manual No.
API for the MX100/DARWIN User's Manual	IM MX190-01E
MX100 Data Acquisition Unit User's Manual	IM MX100-01E

This manual also describes the differences between the API and the extended API. Make sure you understand the features of the extended API before choosing which API best fits your application.

The Yokogawa Web site contains program examples that are not included in the *API for the MX100/DARWIN user's manual* (IM MX190-01E). See the appendix.

Table of Contents

YOKOGAWA
Yokogawa Electric Corporation

1st Edition: June 2009 (YK)

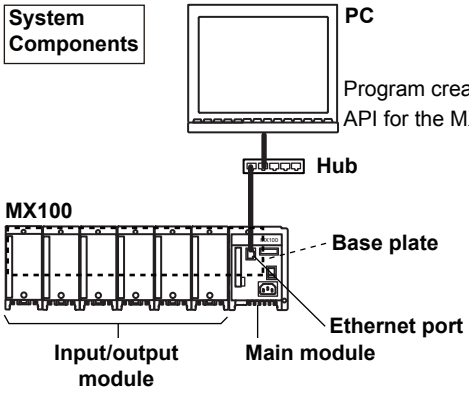
IM MX190-02E
1st Edition

Data Acquisition System Using the MX100

System Components, Module Numbers, and Channel Numbers

System Components

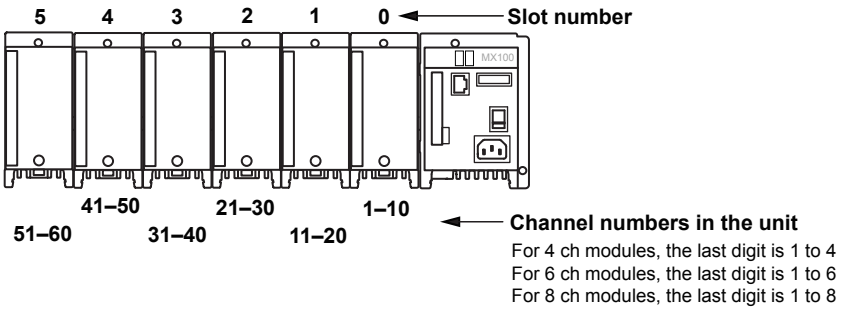
A data acquisition system using the MX100 consists of the data acquisition unit (hereinafter, the MX100), a PC, and a device for connecting to a network. A single MX100 comprises a "main module" with an Ethernet port, "input/output modules" for signal I/O, and a "base plate" for mounting and connecting these components.



Module Numbers

The module numbers assigned to the input/output modules are those of the slot number in which the modules are installed. The slot numbers are 0, 1, 2, 3, 4, and 5. If a module occupies more than one slot, its module number is the lowest slot number that it occupies.

Example: The module number of a 30-CH, Medium-Speed DVC/TC/DI Input Module occupying slots 2, 3, and 4 is "2."



Channel Numbers

Channels are specified by the channel numbers in the unit. The channel numbers in the unit are determined by the slot number of the module (slot position in the unit) and the channel of that module (the terminal position in the module).

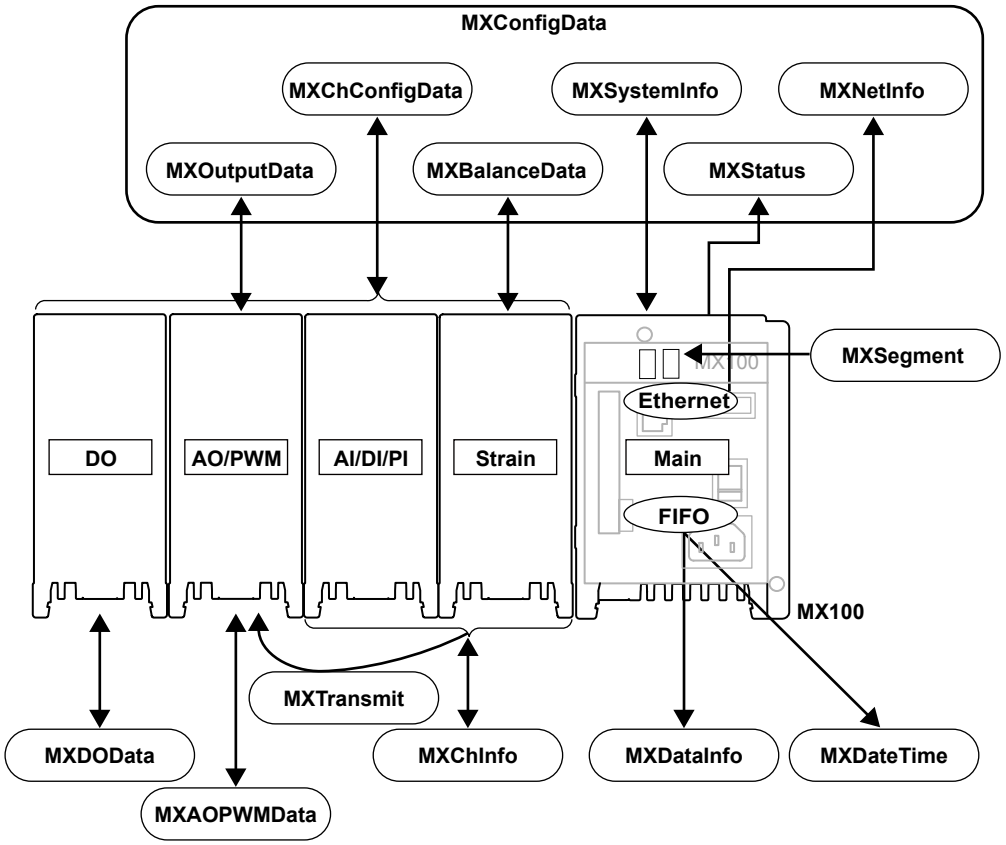
Example: For a module installed in slot 3, its number 2 channel is assigned channel number "32" in the unit.

Note

On the MX100 Standard and MXLogger software programs by Yokogawa, the channel numbers are expressed as 5-digit numbers consisting of the unit number and the channel number in the unit. However with the API, communication takes place by specifying the IP address, therefore only the channel number in the unit is used as the channel number.

Correspondance between MX100 Data and API Structures

The following shows the correspondance between storage structures in the API and their relevant MX100 data.



MXSegment, etc. : Name of structure
← : Arrows indicate direction of data flow
DO : Digital output module
AO : Analog output module
PWM : Pulse width output module
AI : Analog input module
DI : Digital input module
PI : Pulse input module
Strain : Strain input module
Main : Main module
Ethernet : Ethernet communication function
FIFO : Data acquisition function (for details, see page 5 of this manual)

For details on each structure, see the relevant pages in the API for the MX100/DARWIN user's manual (IMMX190-01E).

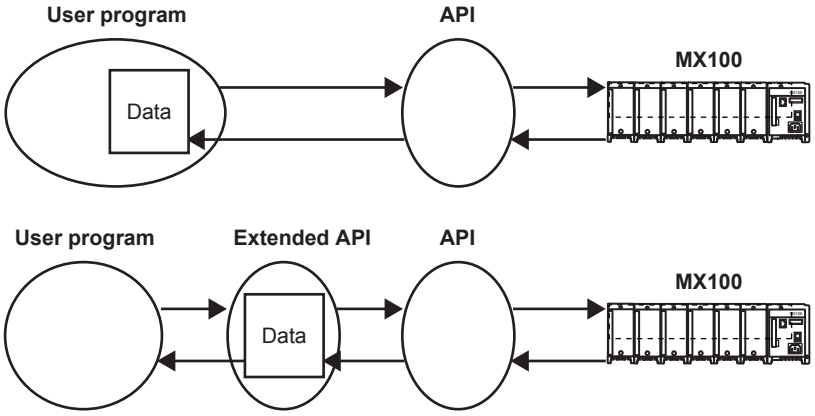
Structure	Description	Reference
MXAOPWMData	Output data from output modules (AO/PWM).	Page 6-37
MXBalanceData	Initial balancing data of the strain input module.	Page 6-35
MXChConfigData	Channel settings data.	Page 6-30
MXChInfo	Measurement channel display data (input modules).	Page 6-30
MXConfigData	All settings data.	Page 6-36
MXDataInfo	Measured data values.	Page 6-27
MXDateTime	Measured data time information.	Page 6-26
MXDOData	Output data from digital output modules (DO).	Page 6-37
MXNetInfo	Communication settings data.	Page 6-35
MXOutputData	Output module settings data (AO/PWM).	Page 6-36
MXSegment	7-segment LED display data.	Page 6-37
MXStatus	System status data.	Page 6-34
MXSystemInfo	System settings data, including for the main module.	Page 6-33
MXTransmit	Transmission output data from input module to output module.	Page 6-37

The API and Extended API

Features of the Extended API

This software contains two components: an API and an extended API. The extended API is designed to make it easier to create programs. It combines several API functions to give it more user-friendly functionality. The extended API is higher in level, and works by calling the "standard" API. The figure below describes the differences between the two APIs. For more details on the API and extended API, see section 1.2, "Software Components and Features" in the *API for the MX100/DARWIN user's manual* (IM MX190-01E).

API and extended API for the MX100



Simplified Programs

With the API, the user sets up the saving area for retrieved data. Therefore performance, which is affected by memory allocation and utilization, can be controlled by the user program. By contrast, the extended API stores and manages the data itself. This creates overhead, but simplifies the program because the user does not need to be concerned with the saving area.

Simplified Data Operations

With the extended API, structures are not used for a function's return values or parameters.

Simplified Functions

Functions of the extended API are simplified into the following two types.

- Status transition functions: Functions that perform communication with measuring instruments, change the status of measuring instruments, and change the data held by the extended API
- Retrieval functions: Functions that reference the data held by the extended API

Additional Programming Language Support

API	Extended API
Visual C++	Visual C++
Visual C	Visual C
Visual Basic	Visual Basic
	Visual Basic .NET
	C#

Notes

Operation is not guaranteed if functions of both the API and extended API are combined. For more notes on the extended API, see page 12-2, "Notes" in the *API for the MX100/DARWIN user's manual* (IM MX190-01E).

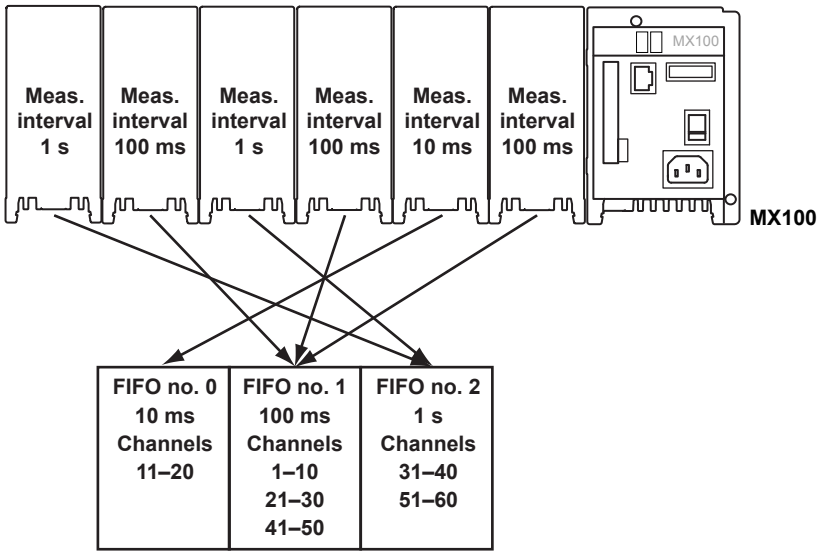
Data Acquisition Function (FIFO)

MX100 Data Acquisition Function (FIFO)

Measurement Interval and FIFO

With the MX100, you can set up to three measurement intervals per module. Buffers are set up in the MX100 to store acquired data for each measurement interval. These buffers operate on a FIFO (first in first out) basis. FIFO means that data is written to the buffer from top to bottom, and when full, new data overwrites old data starting from the top. Each FIFO buffer is assigned the number 0, 1, or 2 in the order of shortest measurement interval.

Measured data from channels on moduels set to the same measurement intervals are stored in the FIFO buffer of the same number. Thus, if you specify a FIFO number to retrieve data from the MX100, you can retrieve measured data from all channels having the same measurement interval.



Channels not set up for measurement are not included.

The table below shows the measurement intervals that can be set on modules.

Module	Model	Selectable Measurement Intervals											
		10	50	100	200	500	1	2	5	10	20	30	60
		ms					s						
4-CH, High-Speed Universal Input Module	MX110-UNV-H04	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
10-CH, Medium-Speed Universal Input Module	MX110-UNV-M10	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
30-CH Medium Speed DCV/TC/DI Input Module	MX110-VTD-L30	N	N	N	N	Y	Y	Y	Y	Y	Y	Y	Y
6-CH, Medium-Speed Four-Wire RTD Resistance Input Module	MX110-V4R-M06	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
4-CH, Medium-Speed Strain Input Module	MX112-B12-M04 MX112-B35-M04 MX112-NDI-M04	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
10-CH, High-Speed Digital Input Module	MX115-D05-H10 MX115-D24-H10	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
10-CH Pulse Input Module	MX114-PLS-M10	N	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Y: selectable, N: not selectable

Retrieving Data with the API or Extended API

The following describes retrieving measured data from a FIFO buffer using the API or extended API.

Program Flow

The flow of programs that retrieve the MX100's measured data are shown respectively for the API and extended API in the tables below. Please refer to the program examples. See the appendix for the locations of the program examples.

API	Function	Program Flow	
	Retrieval of measured data	1	Start FIFO
		2	Get the retrievable data range from the FIFO buffer
		3	Get the measured data time information from the FIFO buffer
		4	Get the measured data values from the FIFO buffer
		5	Stop FIFO

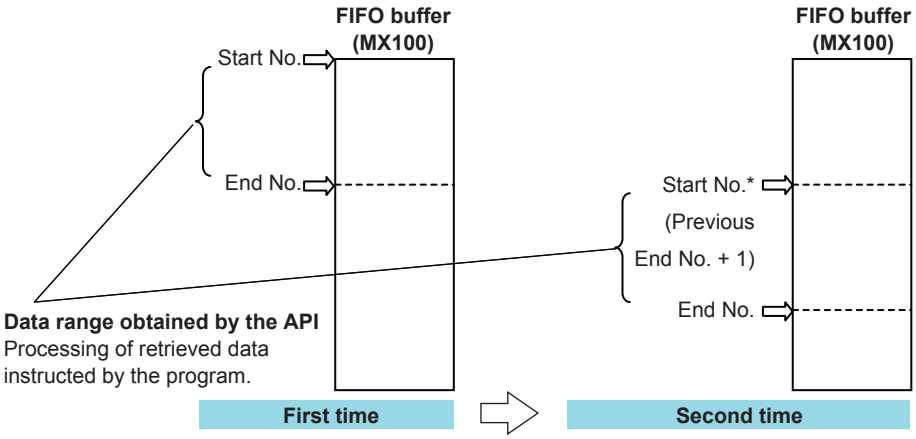
You can retrieve measured data from a FIFO buffer continually by repeating 2, 3, and 4.

Extended API	Function	Program Flow	
	Retrieval of measured data	1	Start FIFO
		2	Advance one measurement point
		3	Retrieve the measured data at the measurement point
		4	Stop FIFO

The program advances by one measurement point each time 2 is executed. Thus, you can retrieve measured data in order by repeating 2 and 3.

With the API

The figure below shows the conditions when the API retrieves measured data from a FIFO buffer (2, 3, and 4 in the "Program Flow" table). The API specifies the range (Start No. and End No.), and retrieves the measured data time information (MXDateTime) and data values (MXDataInfo) from the FIFO buffer. The range that can be specified is obtained by the data range retrieval function.



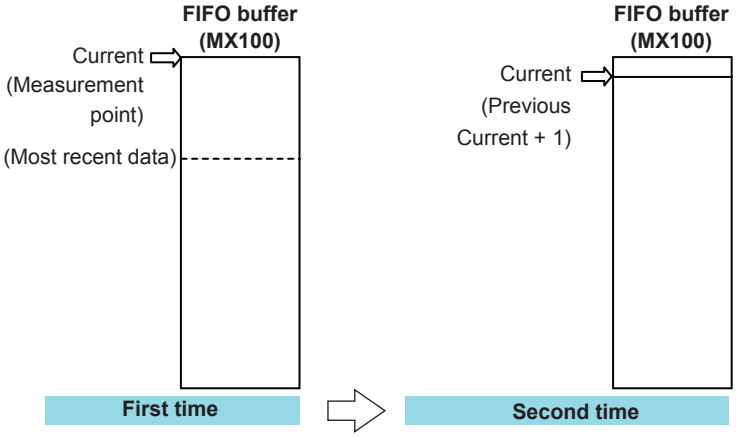
* Data up to the end of the specified range may not be able to be retrieved in one data retrieval process. In this case, the number following the last data able to be retrieved becomes the start number for the next retrieval process.

For information about the functions used, see the referring pages in the table below from the API for the MX100/DARWIN user's manual (IM MX190-01E). See the appendix for the locations of the program examples.

Programming Language	Reference
	Get retrievable data range, get measured data time information, and get measured data
Visual C++	Page 2-8, "Retrieval of Measured Data (FIFO Designation)"
Visual C	Page 3-4, "Retrieval of Measured Data (FIFO Designation)"
Visual Basic	Page 4-4, "Retrieval of Measured Data (FIFO Designation)"

With the Extended API

The figure below shows the conditions when the extended API retrieves measured data (2 and 3 in the "Program Flow" table). When specifying instantaneous values, the most recent data becomes "Current."



To advance to a measurement point, use the measured data retrieval functions (status transition functions). To read Current data, use the appropriate retrieval functions.

For information about the measured data retrieval functions (status transition functions) and the retrieval functions, see the referring pages in the table below from the *API for the MX100/DARWIN user's manual* (IM MX190-01E). See the appendix for the locations of the program examples.

Programming Language	Reference	
	Retrieval functions of the status transition functions	Retrieval functions
Visual C++	Page 12-8, "Retrieval Functions"	Page 12-9
Visual C	Page 13-6, "Retrieval Functions"	Page 13-7
Visual Basic	Page 14-6, "Retrieval Functions"	Page 14-7
Visual Basic .NET	Page 15-6, "Retrieval Functions"	Page 15-7
C#	Page 16-6, "Retrieval Functions"	Page 16-7

Data Acquisition with Instantaneous Values

Data acquisition with instantaneous values involves retrieving the most recent data from the FIFO buffer. The API can access the MX100's measured data in as fast as 100 ms. Thus, instantaneous values are retrieved every 100 ms at best. To retrieve data at faster measurement intervals, the FIFO number must be specified.

Starting and Stopping FIFO

FIFO must be started before retrieving data from the MX100. When FIFO is started by the API, all FIFOs are started, and when stopped, all FIFOs are stopped.

Notes When Retrieving Data

The PC must access the MX100 and read out the data before the buffer becomes full. The size of the three FIFO buffers combined is 2 MB. For example, when acquiring 60 channels of data at a measurement interval of 10 ms, the buffer becomes full in approximately 60 seconds.* When the buffer becomes full, the MX100 overwrites old data with new starting from the top. Therefore, the PC must retrieve the data from the MX100 at intervals of less than 60 seconds. Determine the actual retrieval interval based on your application.

* For instructions on how to calculate the time required to fill the buffer, see appendix 3, "Calculation of the MX100 Timeout Value" in the API for the MX100/DARWIN user's manual (IM MX190-01E).

If access is not gained in approximately 3 minutes, the MX100 breaks off communications.

Appendix

Locations of Program Examples

User's Manual

The program examples in the table below are located in the *API for the MX100/DARWIN user's manual* (IM MX190-01E) included on the CD-ROM.

Program Examples for the API

Programming Language	Description	Reference
Visual C++	Retrieval of the Measured Data	Page 2-10
	Retrieval of Setup Data and Configuration	Page 2-13
Visual C	Retrieval of the Measured Data	Page 3-7
	Retrieval of Setup Data and Configuration	Page 3-11
Visual Basic	Retrieval of the Measured Data	Page 4-7
	Retrieval of Setup Data and Configuration	Page 4-10

Program Examples for the Extended API

Programming Language	Description	Reference
Visual C++	Retrieval of the Measured Data	Page 12-16
	Retrieval of Setup Data and Configuration	Page 12-18
Visual C	Retrieval of the Measured Data	Page 13-13
	Retrieval of Setup Data and Configuration	Page 13-15
Visual Basic	Retrieval of the Measured Data	Page 14-13
	Retrieval of Setup Data and Configuration	Page 14-15
Visual Basic .NET	Retrieval of the Measured Data	Page 15-13
	Retrieval of Setup Data and Configuration	Page 15-15
C#	Retrieval of the Measured Data	Page 16-13
	Retrieval of Setup Data and Configuration	Page 16-15

Web Page

The program examples can be found at the following URL.
http://www.yokogawa.com/ns/mx100/download/

Description	Programming Language	API	Extended API
Retrieving measured data and displaying waveforms.	Visual C++	Y	Y
	Visual C	Y	Y
	Visual Basic	Y	Y
	Visual Basic .NET	N	Y
	C#	N	Y

Y: Program example available.
N: Program example not available (not supported).

Location of the FAQ (Frequently Asked Questions)

Freuquently asked questions and their answers are given at the following URL.
http://www.yokogawa.com/ns/support/faq/ns-faq-mx190_01.htm