

MX Foundation SDK offers you groups of function calls that interface with on-board MX core software and control all protocols and I/O supported by MAX Technologies. The on-board MX core handles the various protocol & I/O timing, data scheduling, buffering and filtering tasks

This combination offloads your host computer and makes available resources for your critical software applications. MX Foundation gives you high-level abstraction (channel class-oriented) of hardware and communication protocols.

In addition to the overall learning curve process and the time-to-market that can be tremendously reduced, MX Foundation also offers a full evolutionary capability by being hardware and operating system independent, thus reducing your overall system life-cycle support cost.

MX Foundation helps to reduce the difficulties and the engineering cost related to systems development that require the integration of different communication protocols, analog and discrete signals.

MX Foundation also helps to reduce the cost and time inconveniences of being trapped with a software application that runs under a specific operating system and hardware form factor, which might not be upgradeable to a new environment.



MX Foundation is based on a channel-class model and allows users to access all resources using a single API - minimizing learning-curves, maximizing code reusability and optimizing system maintainability and expandability

FEATURES

- ▶ "Common" set of ten groups of function calls that apply to all protocol and I/O supported by MAX Technologies (ASYNC RS-422/485, HDLC, A/D, D/A, DIO, MIL-STD-1553, ARINC429, ARINC629, ASCB, CSDB and more)
- ▶ MX-Foundation 32-bit for Windows and Linux works with MAX Technologies series 400 PCI, CPCI and PXI cards
- ▶ MX-Foundation 64-bit experimental support for Mac OS X Snow Leopard (10.6) works with MAX Technologies series 400 PCI cards
- ▶ "Embedded Service" Allows you to you develop user's specific C-code and download it on MAX Technologies 400 series intelligent boards
- ▶ Interfaces directly with the on-board MX-Core firmware of MAX Technologies' Intelligent IPack Carrier Boards
- ▶ Automatically detects all protocols, I/O and number of channels involved in a particular system
- ▶ Unique sophisticated on-board event scheduler that can transmit simple or complex scenarios with very high (1 microsecond) timing resolution, precision, synchronization and determinism over all channels and protocols

BENEFITS

- ▶ Reduces your overall time-to-market
- ▶ Reduces your overall system life-cycle support cost
- ▶ Reduces your overall learning curve process – always the same function calls that are used independently of the hardware platform, operating system, protocol and I/O involved in your development
- ▶ Reduces the load on your host CPU – unique sophisticated on-board "core" intelligence handles the various protocol and I/O communication tasks very precisely.



MX Foundation

Base Service <ul style="list-style-type: none"> ▶ "Channel class" approach ▶ Automatic channel detection ▶ "Intelligent" filter for accessing a specific channel on a specific protocol or all channels ▶ Easy to use and fast channel attribute configuration ▶ One function to address multiple protocols 	Queued Transmit Service <ul style="list-style-type: none"> ▶ Direct channel FIFO transmission ▶ Deferred transmission with 1-microsecond precision ▶ Choice of various priority queues per channel ▶ Programmable output delay ▶ Various buffer modes ▶ Precise data filtering ▶ Independent control over all transmitted messages
Scheduled Transmit Service <ul style="list-style-type: none"> ▶ Choice of various priority queues for all scheduled messages ▶ High priority and background queues can be used simultaneously on scheduled messages ▶ Schedules can be controlled by time or event ▶ Transmit up to 1024 schedules of linked messages ▶ Can be used for repetitive transmission or on time-tag ▶ All schedules can start simultaneously (Ex.: 64 ARINC429 transmitters) with 1-microsecond precision ▶ Port/channel oriented ▶ Real-time modification of the different words and events 	
Queued Receive Service <ul style="list-style-type: none"> ▶ Time correlation over all received channels (1-microsecond) ▶ One acquisition queue built within the card per receiver ▶ Store received data with precise time-tag and control word (error status) ▶ Precise filtering and triggering capability ▶ Resolution, buffer size, type and number of words are specifiable ▶ A function call is provided for the host application to read from a receiver acquisition queue directly 	Sampling Receive Service <ul style="list-style-type: none"> ▶ Receive incoming data with precise time-tag, message rate and control word (error status) ▶ Time correlation over all received channels (1-microsecond) ▶ Various buffer modes, sizes and types are specifiable ▶ Configurable kill-time ▶ Precise filtering and triggering capability ▶ One sampling record for each of the defined channels
Bridge Service <ul style="list-style-type: none"> ▶ Specify the source and destination ▶ Modify the received data at the bit level (force to zero, zero or invert) ▶ The retransmission deterministic delay is less than 2 ms ▶ Precise filtering 	Interrupt and Condition Service <ul style="list-style-type: none"> ▶ Allocate and manage a condition list ▶ Precise at the bit level ▶ Pre-trigger buffer size is configurable on reception
Embedded Service <ul style="list-style-type: none"> ▶ Free GNU tool-chains for Linux and Mac OS X ▶ On-board resulting code works with resident MX core and can access easily all on-board I/O resources ▶ Applications include: On-board time-critical message handling and routing, protocol bridging and gateway, analog and discrete signal processing, simulation models, feedback controls and more... 	

Ordering Information

Part Number	Description
MAX-MXF-200100	MX-FOUNDATION MULTI-PROTOCOL 32-bit API for Windows
MAX-MXF-200101	MX-FOUNDATION MULTI-PROTOCOL 32-bit API for Linux (please call for infos on binary distros and pricing)
MAX-MXF-200103	MX-FOUNDATION MULTI-PROTOCOL 64-bit API for Mac OS X 10.6 Snow Leopard (experimental)