

Motion Solutions by INtime® / EtherCAT® / Hivertec

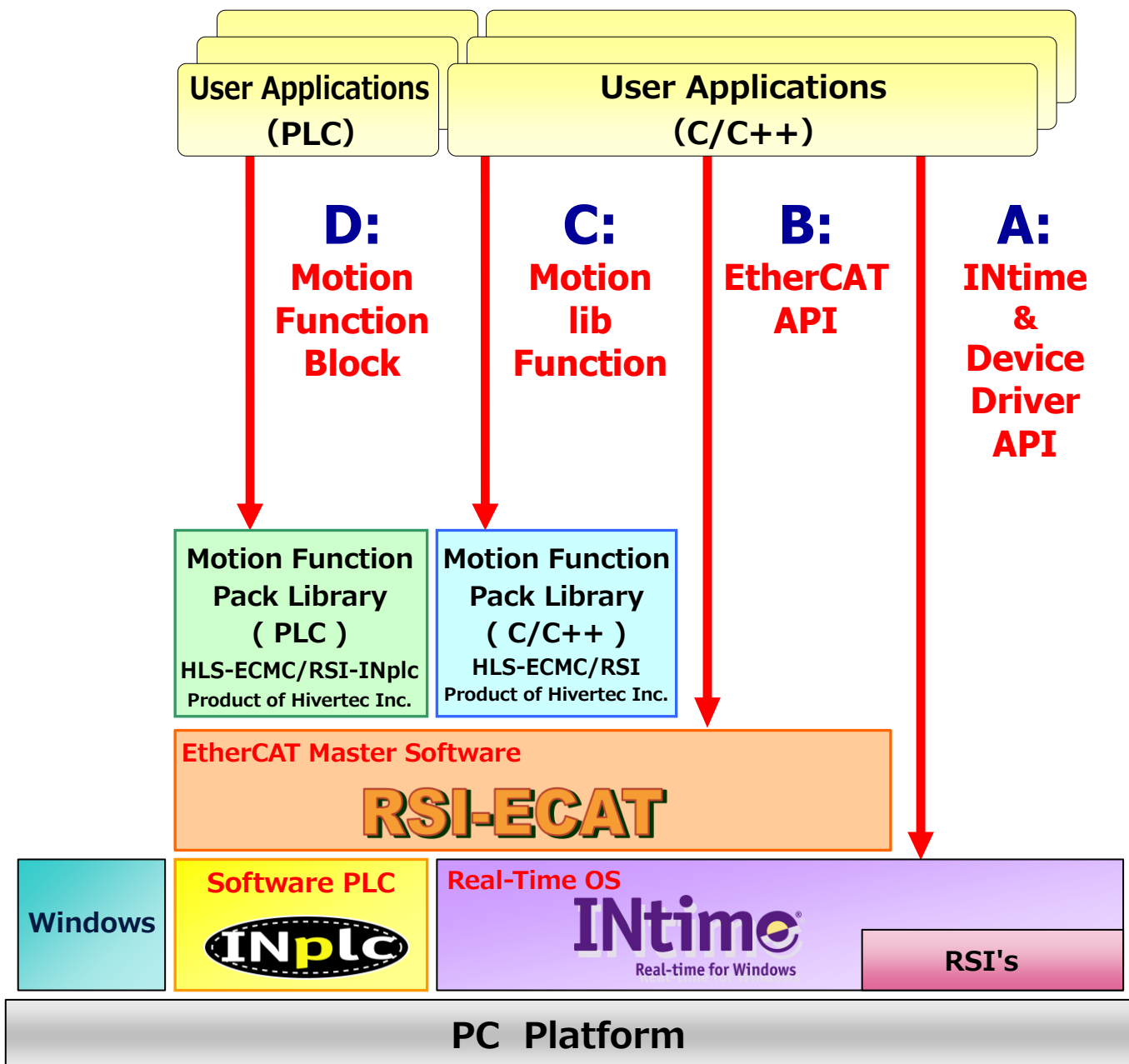


Figure 1 : Motion Solution by INtime®/EtherCAT®/Hivertec

*Please refer to another catalog or ask us individually on tools / environments in actual development.

A : INtime® & Driver API

INtime® is a real time OS which provides real-time performance / reliability so that PC can be used for industrial controls.

It can run coexistent with Window®. You can develop real time applications using Microsoft® Visual Studio. APIs for real time applications (※Table 2) and Drivers (※Table 3) already have been prepared.

Table 1 : INtime® Specifications

Priority Scheduling	256 levels
Kernel Tick	50us, 100us, 125us, 200us, 250us, 500us, 1ms, 2ms, 5ms, 10ms
Mailbox	FIFO Method/ Priority Method
Semaphore	FIFO Method/ Priority Method
Maximum Segment Size	4GB in system total
Interrupt	Handler / Thread Architecture, IRQ / MSI Available
Development Language	C/C++
Standard I/O	TCP/IP, TCP/IP, USB, RS232C
Maximum Object Quantity	8190 (including system)

Table 2 : INtime® API (excerpt)

Interrupt Management
SignalEndOfRtInter, SetRtInterruptHandler, ResetRtInterruptHandler, SignalRtInterruptThread, WaitForRtInterrupt, DisableRtInterrupt, EnableRtInterrupt
Mailbox Management
CreateRtMailbox, DeleteRtMailbox, SendRtHandle, ReceiveRtHandle, SendRtData, ReceiveRtData
Memory Management
AllocateRtMemory, FreeRtMemory, CreateRtMemoryHandle, DeleteRtMemoryHandle, MapRtShareMemory, MapRtPhysicalMemory, GetRtPhysicalAddress, GetRtSize
Object Management
CatalogRtHandle, LookupRtHandle, UncatalogRtHandle, GetRtHandleType, GetRtHandleTypeEx, InspectRtProcessDirectory
Process Management
CreateRtProcess, ExitRtProcess, RtNotifyEvent, RegisterRtDependency, UnregisterRtDependency, RegisterRtSponsor, UnregisterRtSponsor
Scheduler Management
knRtSleep, knStartRtScheduler, knStopRtScheduler
Semaphore Management
CreateRtSemaphore, DeleteRtSemaphore, WaitForReSemaphore, ReleaseRtSemaphore
Status Management
GetLastRtError, SetLastRtError, CopyRtSystemInfo, ReportRtEvent
Thread Management
CreateRtThread, DeleteRtThread, RtSleep, GetRtThreadPriority, GetRtThreadHandles, SetRtThreadPriority, SetRtProcessMaxPriority, GetRtThreadAccounting, SuspendRtThread, ResumeRtThread, GetRtThreadInfo
System Data Management
ntxGetLocationByName, ntxGetFirstLocation, ntxGetNextLocation, ntxGetNameOfLocation

Table 3 : INtime® Drivers Product

Serial Communication Bus
RSI-001, RSI-008, RSI-016, RSI-006, Standard Serial Control Driver
Digital Communication Bus
RSI-002, Standard USB Driver
A/D Conversion Board
AD12-16U(PCI)EH, AD16-16(LPCI)L Library
Image Recongniton
AWP-332-04, AUP-SEARCH, AZP-7101-02
Motion Control
RSI-612, RSI-534, RSI-578, RSI-5212, RSI-708, RSI-SMC4DF, RSI-SMC8DF, RSI-SMC4DL, RSI-SMC8DL
Real time network (FL-net)
RSI-040
Fieldbus
RSI-CCL, RSI-CCIE, RSI-060

B : EtherCAT® Master Software「RSI-ECAT」

We have RSI-ECAT-Master as an EtherCAT master software for INtime®. RSI-ECAT-Master has the functions described in Table 4. RSI-ECAT-Master can do basic I/O control such as Analog I/O and can do controls such as basic servo motors using API described in Table 5.

Table 4 : RSI -ECAT Master Function Specifications

Function	RSI-ECAT-Master		
	-	DC	RED
Basic spec			
Protocol	IEC61158-2/3/4/5/6-12		
Max Slaves	65,535		
Baudrate	100Mbps		
Data transport layer	100BASE-TX		
Duplex	Full		
Max cable length slave-slave	100m		
Master cycle			
Bus cycle time(max)	100us		
Bus cycle time(default)	500us		
Priority	Configurable		
Special feature			
DC support		○	○
Cable redundancy support			○
Hot connect support			○
64bit Windows support	○	○	○
INtime multi kernel support	○	○	○
Master class specification			
Master-Class	Class B	Class A	Class A + FP

Table 5 : RSI -ECAT Master API List

Category	API	Contents
API Conversation	EhGetEhNodeStatus	Confirm Running Status of EcatHandler
	EhOpen	Open API Conversation to make APIs available
	EhClose	Close API Conversation
Master API	EhRqState	Change Request for Master State
	EhGetState	Get Current State and Now Requesting State
	EhWaitForCyclic	Wait for Starting Master Cyclic Process
	EhGetSystemInfo	Get EcatHandler info
	EhRequestAsyncOutputs	Starts one cycle of asynchronous data output
	EhSetOverrunCheck	Set info about a cyclic overrun check function
	EhGetOverrunCheck	Acquire info on a cyclic overrun check function
	Slave API	EhGetSlaveCount
EhGetOnlineSlaveCount		Get number of connected slaves
EhFindSlave		Get detail info of slaves (VendorID, ProductCode, Instance)
EhFindSlaveByAlias		Get detail info of slave which has designated alias
EhFindSlaveBySlaveNo		Get detail info of slave which has designated slaves number
EhGetSlaveStatus		Get status of the target slave
EhGetALStatus		Gets AL status from the target slave
EhGetALStatusCode		Gets AL status code from the target slave
EhGetDLStatus		Gets DL status code from the target slave
EhReadRegister		Read data in ESC-register which the target slave has
EhWriteRegister		Write data in ESC-register which the target slave has
Event API	EhReadEEPROM	Read data in EEPROM which the target slave has
	EhWriteEEPROM	Write data in EEPROM which the target slave has
	EhRecalcChecksum	Update EEPROM checksum which the target slave has
	EhSetEventFilter	Set filter for diagnosing event
	EhGetEventFilter	Get filter info for set diagnosing event

Category	API	Contents
Event API	EhWaitForEvent	Wait for diagnosing event
DI Category API	EhDiRead	Read 16 bit data of designated DI channel
	EhDiBlock	Read optional size of data from optional offset position of DI category
	EhDiGetChNums	Get number of channels of DI category.
DO Category API	EhDoWrite	Write 16 bit data to designated DO channel
	EhDoRead	Read 16 bit data of designated DO Channel
DO Category API	EhDoBlock	Write optional size of data to optional offset position of DO category
	EhDoGetChNums	Get Number of channels of DO Category
AI Category API	EhAiRead	Read 32 bit data of designated AI channel
	EhAiBlock	Read Optional size of data from optional offset position of AI category
	EhAiSetRange	Set range for designated AI slave
	EhAiGetRange	Get range for designated AI slave
	EhAiGetChNums	Get number of channels of AI category
AO Category API	EhAoWrite	Write 32 bit data to designated AO channel
	EhAoRead	Read 32 bit data of designated AO channel
	EhAoBlock	Write optional size of data to optional offset position of AO category
	EhAoSetRange	Set range for designated AO slave
	EhAoGetRange	Get range for designated AO slave
	EhAoGetChNums	Get number of channels of AO category
OD Access API	EhReadOD	Read data from OD(Object Dictionary) of designated slave
	EhWriteOD	Write data to OD(Object Dictionary) of designated slave
	EhReadODByAlias	Read data from designated index of slave which has designated alias
	EhWriteODByAlias	Write data to designated index of slave which has designated alias
VIOS Access API	EhGetViosInOffset	Get offset value of VIOS Input area of slave which has designated alias
	EhGetViosOutOffset	Get offset value of VIOS Output area of slave which has designated alias
	EhReadByte	Read 8 bit data from VIOS Input area
	EhReadWord	Read 16 bit data from VIOS Input area
	EhReadDword	Read 32 bit data from VIOS Input area
	EhWriteByte	Write 8 bit data to VIOS Output area
	EhWriteWord	Write 16 bit data to VIOS Output area
	EhWriteDword	Write 32 bit data to VIOS Output area
	EhReadbackByte	Read 8 bit data from VIOS Output Area
	EhReadbackWord	Read 16 bit data from VIOS Output area
	EhReadbackDword	Read 32 bit data from VIOS Output area
	EhGetViosInAddress	Get the VIOS Input memory address
	EhGetViosOutAddress	Get the VIOS Output memory address

C : Motion Function Pack Library (C/C++) "HLS-ECMC/RSI" (presented by Hivertec Inc.)

Motion Function Pack Library "HLS-ECMC/RSI" is a software control module which does motion control on INtme.

You can achieve motion control with sophisticated performance using C language programs by INtime-SDK.

Since API functions have I/O variables based on international specifications, you can achieve motion control quite easily.

It controls servo drives with CiA402 driver profile.

Type is as follows.

Ex:HLS-ECMC0106/RSI+2

Function Basic Package axes number Option axes number

Function : 「01」 is Positioning ※Plan to release "Interpolation" etc...

Basic Package axes number : Basic Package as 6 axes,12 axes, 18 axes, 24 axes, 30 axes, 36 axes,42 axes, 48 axes, 54 axes, 60 axes

Option axes : 1~5 ※Up to 4 axes in option more than 60 axes

※We can correspond to axes customization adding to Basic Package.

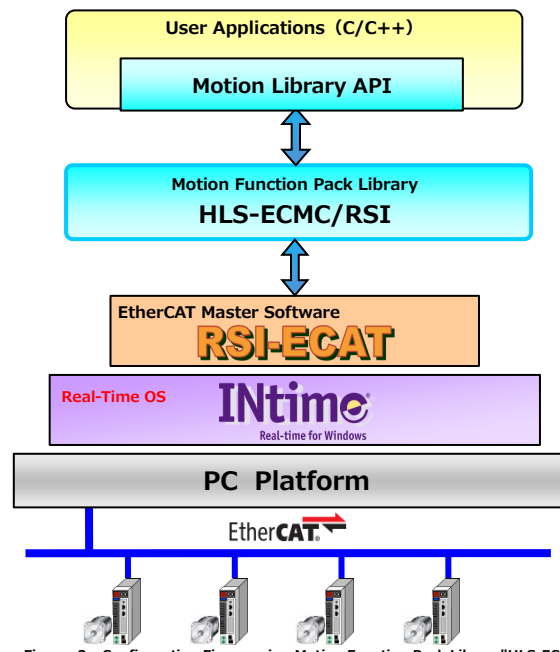


Figure 2 : Configuration Figure using Motion Function Pack Library "HLS-ECMC/RSI"

EtherCAT Motion Slave with Checking Operations :

- Hitachi Industrial Equipment Systems ADV Series
- LS Mecapion L7N Series
- Omron G5 Series
- Panasonic MINAS A5B
- SANYO DENKI SANMOTION R
- etc...

※ Any slaves must correspond to EtherCAT hm, CSP mode.

What is CiA402 Drive Profile? :

CiA402 Drive Profile is a profile for drives / motion control devices based on IEC61800-7-201 and IEC61800-7-301. As for EtherCAT series, it is defined in the definition part for controller function operation.

Notice) What you must create :

There are what you create in software configuration in PC.

Software for EtherCAT execution

You must create RTA file which is INtime execution file.

For instance, it carries out functions as follow.

- Reserve shared memory area (For it exchange I/O parameter of API functions for Motion Function Pack Library between Windows.)
- Read / Write function to shared memory
- Call API functions of Motion Function Pack Library by calling API functions of RSI-ECAT-Master

Software for user interface

You can create by VB / VC#.

Operating Environment :

Support INtime 4.2

Support RSI-ECAT-Master (/DC or /RED) Ver.2.2

※PC Spec depends on INtime, RSI-ECAT-Master. Please use NICs supported by RSI-ECAT-Master.

Table 6 : Motion Library API List

	Category	API	Contents
1	Administrative	MC_Power()	Power Control
2		MC_ReadStatus()	Read Status
3		MC_InitAxisSetting()	Initialize Axis Setting
4		MC_GetAxisSetting()	Get Axis Setting
5		MC_SetAxisSetting()	Set Axis Setting
6		MC_ReadAxisError()	Read Axis Error
7		MC_ReadActualPosition()	Read Current Position
8		MC_ReadActualVelocity()	Read Current Velocity
9		MC_ReadAxisInfo()	Read Axis Information
10		MC_ReadMotionState()	Read Motion State
11		MC_SetPosition()	Change Current Position
12		MC_SetOverride()	Set Override factor for Velocity, Adjustable-Velocity, Jerk
13		MC_TouchProbe()	Read Current Position by Trigger Event
14		MC_AbortTrigger()	Stop to read Current Position by Trigger Event
15		MC_Reset()	Reset Axis Error
16	Motion	MC_Home()	Homing
17		MC_Stop()	Stop Deceleration
18		MC_MoveAbsolute()	Absolute Value Positioning
19		MC_MoveRelative()	Relative Value Positioning
20		MC_MoveAdditive()	Add relative value in Positioning
21		MC_MoveVelocity()	Continuous Feed by designated velocity
22	EtherCAT®	EC_InitMaster()	Initialize Master
23		EC_GetMasterInfo()	Get Master Information
24		EC_OpenMaster()	Open Master
25		EC_CloseMaster()	Close Master
26		EC_GetSlaveInfo()	Get Slave Information
27		EC_InitSlave()	Initialize Slave
28		EC_GetDiagnosisMessage()	Get Slave Diagnosis Message
29		EC_SetDiagnosisMessage()	Set Slave Diagnosis Message
30		EC_ResetDiagnosisMessage()	Clear Slave Diagnosis Message
31		EC_ReadVersion()	Get Version of Motion Function Pack Library and Axes Number
32		EC_ReadSDO()	Read SDO Data
33		EC_WriteSDO()	Write SDO Data
34	IO	IO_inp	Read 8 bit s of PDO Data
35		IO_inpw	Read 16 bit s of PDO Data
36		IO_inpdw	Read 32 bit s of PDO Data
37		IO_outp()	Write 8 bit s of PDO Data
38		IO_outpw()	Write 16 bit s of PDO Data
39		IO_outpdw()	Write 32 bit s of PDO Data

D : Motion Function Pack Library (PLC) "HLS-ECMC/RSI-INplc" (presented by Hivertec Inc.)

Motion Function Pack Library "HLS-ECMC/RSI-INplc" is made according to Motion Function Block Specifications regulated by PLCopen who is the propeller of "International Standard PLC language Specification IEC61131-3".

Therefore, you can use not only C language but also PLC language. You can use functions for motion controls from PLC as if you used C language.

Regarding development by INplc, please ask us individually.

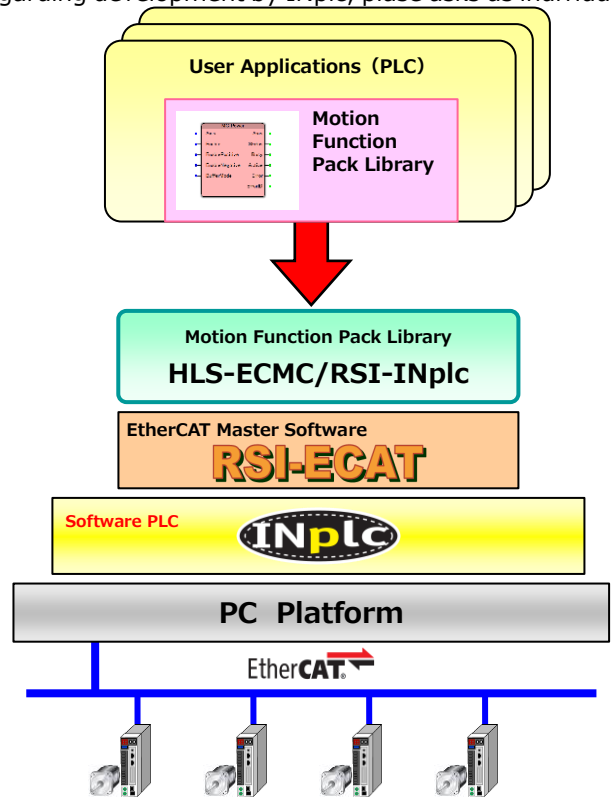


Figure 3 : Software Configuration by Motion Function Pack Library

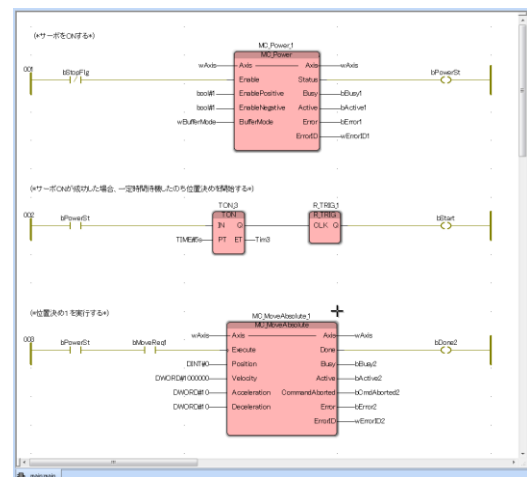


Figure 4 : Programming Sample (Ladder)

※ TenAsys®, INtime®, eVM® and iRMX® are registered trademarks in USA of the TenAsys Corporation.

※ EtherCAT® is a trademark of Beckhoff Automation GmbH, Germany.