Documentation: ART l ogics API-V0.1.doc

Version: V0.1

REVISION HISTORY

Ver sion	Date	Edit	Reviewed by	History
0.1	2024-05-10	Louis Yang	Nadir	1st version

catalogue

1. ART logics

1.1. company profile

ART logics Testing Equipment Technology (Shanghai) Co., Ltd. is committed to meet the production requirements of the automotive industry by designing and manufacturing high-quality electronic testing modules; the test modules cover test control unit, simulated volume acquisition, data conversion, communication conversion, various loads, and even delivery of test software modules; ART has delivered more than 100 equipment platforms and test bench integration solutions and large test systems, and all test modules are verified by the delivered projects.

01 Central European background

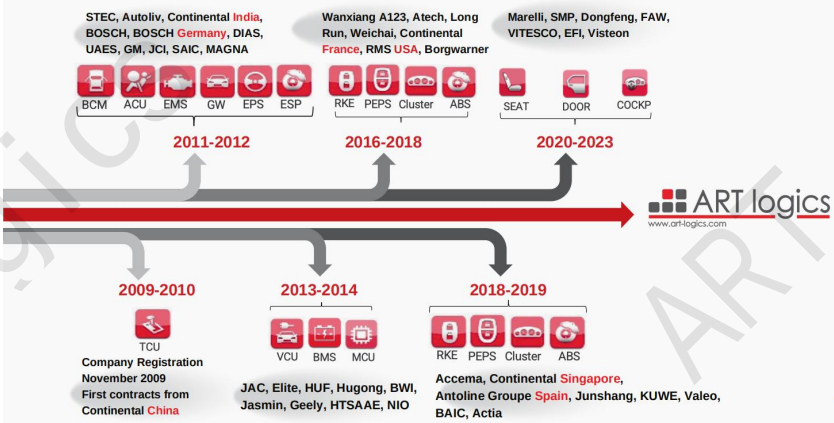
Since its establishment in 2009, ART logics Testing Equipment Technology (Shanghai) Co., Ltd. is an international group that has always provided high-quality testing equipment technology products and test system integration services in China and Europe. It is headquartered in Shanghai.

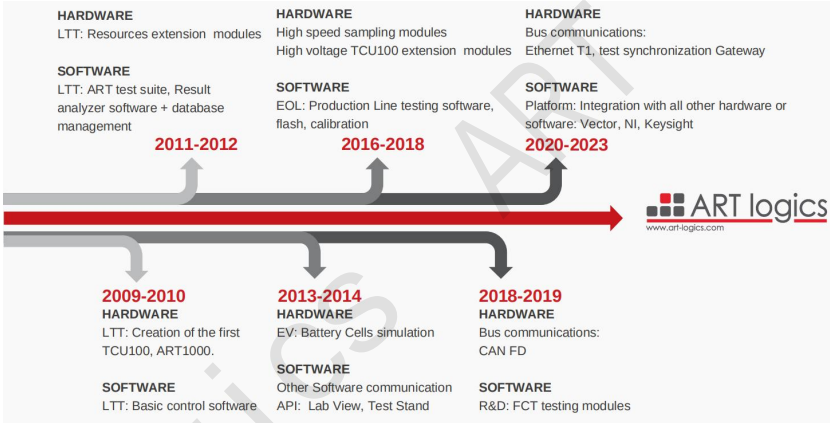
02 Professional team

The founder of the company is a senior manager and technical expert from the automotive electronics industry; research and development engineers from Europe and China focus on automotive electronics and professional mature project team responsible for system integration based on customer specific needs.

03 Efficient and win-win situation

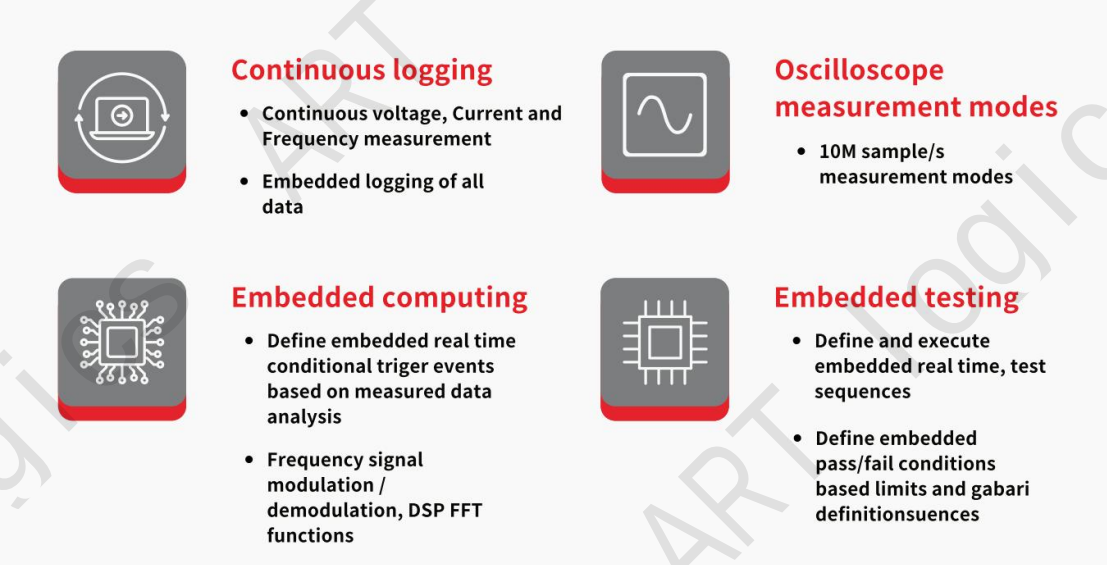
The company is actively committed to exploring various technology fields, including electric vehicles, and works closely with major original equipment manufacturers (OEMs)and tier 1 suppliers in the automotive industry.





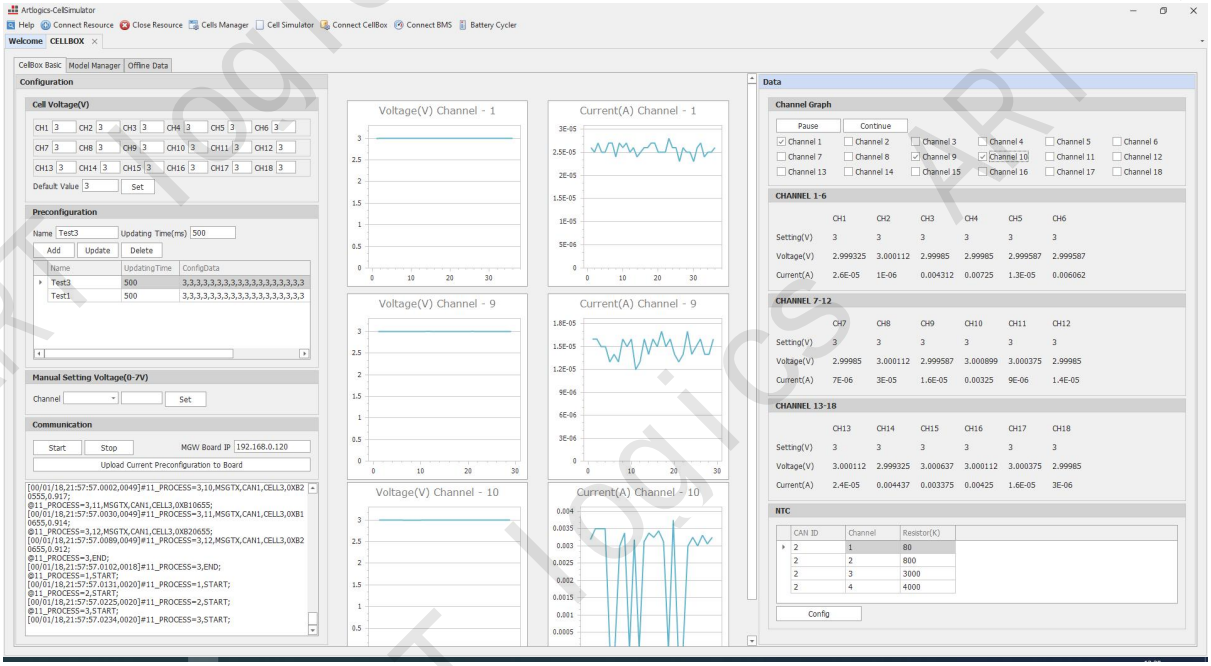
1.2. Product

1.3. Technology

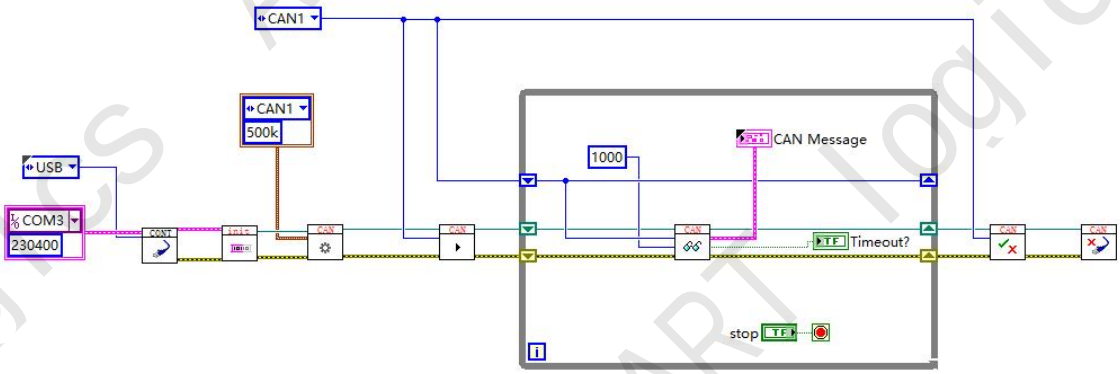


1.4. API application example

- The 18-channel battery simulator C # application software



- CAN Push The paradigm of in Labview



1.5. API document conventions

The library file name is the ARTlogicsBasic. The dll, based on the C #. NET Framework4.6 Compilation.
The namespace of the protocol command is the ArtLogicsBasic.

The namespace for the application operation is the ArtLogicsBasic.Operation。

2. ART Protocol

2.1. Summary description

ART products can be communicated by serial port or network port.

The command sent by the host computer to the ART product begins with "@" and ends with ";". The response message begin s with "#" and ends with ";".

Each ART product has an address code with a default of "11", attached after "@" and "#".

Examples of the command format are given below:

The upper computer is sent to the ART product: @ 11 _ SYSI D = RESOURCES;

Response message: # 11 _ SYSID = RESOURCES, R50, VI15, VO 8, AWG 8, CI5, DI19, DO 20, FI4, FO 8, CAN 2, LIN 2, KLI NE1;

2.2. communication API

2.2.1.Communication base class ARTCommunication

Class		
	Item	Description
	function	Provide communication interface
	NAMESPACE	ArtLogicsBasic
	statement	public class ARTCommunication

data		
	Name	Description
	DataDirectory	Communication log storage directory.
	Name	Communication port name.
	void HandleReceivedData	Processing function agent type for the response data.
	void SetStatusCallback	Function agent type that output the communication message.

function		
	Name	Description
	GetCommunicationInstance(string portName)	
	GetCommunicationInstance(string destip, int destport)	
	Dispose()	
	Dispose(string portName)	
	Dispose(string destip, int destport)	
	Start()	
	Stop()	
	IsReady()	
	Send(ARTPacket ap)	
	Send(byte[] data)	

	Send(string sdata)	
	GetDataQueue()	

2.2.2.Serial port communication, ARTSerialPortHelper

Class		
	Item	Description
	function	Provide the serial port communication interface
	N AMESPACE	ArtLogicsBasic
	statement	public class ARTSerialPortHelper: ARTCommunication

data		
	Name	Description

function		
	Name	Description
	CreateInstance(string portName, ARTBaudrate baudRate = ARTBaudrate.B57600, string dataDirectory = "")	Get the COM communication processor; create if not present;

2.3. The protocol command API

2.3.1.Command the base class ArtBaseCommand

Class		
	Item	Description
	function	Base class as a protocol command.
	N AMESPACE	ArtLogicsBasic
	statement	public class ArtBaseCommand

data		
	Name	Description
	void SetStatusCallback	State callback function agent type.
	void HandlerCallback	Data processing result callback function agent type.

function		
	Name	Description
	ArtBaseCommand(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	Create the command for processing the object
	Dispose()	Release the resource used when the command processes
	SetStatus(string status)	Call the state callback function

2.3.1.1. ArtBaseCommand(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)

statement	public ArtBaseCommand(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
logic	Construction commands process objects.	
Parameter	Type	Description
comHelper	ARTCommunication	The Communication Processor object.
deviceid	string	Device type identification of the product, two-digit number; can be empty for devices compatible with 4-bit identification.
unitid	string	Hardware identification of the product, two digits.
statusCallback	SetStatusCallback	Interface function used for the output information.

2.3.2.SYSID

Class		
	Item	Description
	function	Read the software version of MAIN CPU, PICs, BOARD ID, Extension and resources
	N AMESPACE	ArtLogicsBasic
	statement	public class SYSID: ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SYSID(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SYSID(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SYSID(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSYSID()	
	CommandSYSID_RESOURCES()	
	CommandSYSID_EXTENSIONS()	

2.3.3.GETDIG

Class		
	Item	Description
	function	Read the level of a digital input channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class GETDIG : ArtBaseCommand
	service logic	

data		
	Name	Description
	string channelStatus	The state of the c hannel

function		
	Name	Description
	GETDIG(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	GETDIG(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GETDIG(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GetDigital(string channel, ref string data)	
	CommandGETDIG(string channel)	
	CommandGETDIGByMask(string ChannelMask)	
	CommandGETDIG()	

2.3.4.SETDIG

Class		
	Item	Description
	function	Set one or more digital level output channels to a high level “1”.
	N AMESPACE	ArtLogicsBasic
	statement	public class SETDIG : ArtBaseCommand
	service logic	

data		
	Name	Description

	string channelStatus	

function		
	Name	Description
	SETDIG(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SETDIG(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SETDIG(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSETDIG(string channel)	
	CommandSETDIGByMask(string ChannelMask)	
	CommandSETDIG()	

2.3.5.CLRDIG

Class		
	Item	Description
	function	Set the digital level of an output channel to a low level “0”.
	N AMESPACE	ArtLogicsBasic
	statement	public class CLRDIG : ArtBaseCommand
	service logic	

data		
	Name	Description
	string channelStatus	

function		
	Name	Description
	CLRDIG(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CLRDIG(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CLRDIG(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandCLRDIG(string channel)	
	CommandCLRDIGByMask(string ChannelMask)	

2.3.6.RELAY

Class		
	Item	Description
	function	Control relays channels.
	N AMESPACE	ArtLogicsBasic
	statement	public class RELAY : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	RELAY(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	RELAY(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	RELAY(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	Open(string channelsOrMask)	
	CommandOPEN(string channelsOrMask)	
	Close(string channelsOrMask)	
	CommandCLOSE(string channelsOrMask)	
	Setrly(string channelMask)	
	CommandSETRLY(string channelMask)	

2.3.7.GETVOLT

Class		
	Item	Description
	function	Get the value of a voltage input channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class GETVOLT : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	GETVOLT(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	GETVOLT(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GETVOLT(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GetVoltage(string channel)	
	CommandGETVOLT(string channel)	
	ParseGETVOLT(ref string rspValue)	

2.3.8.SETVOLT

Class		
	Item	Description
	function	Set the voltage value of a voltage output channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class SETVOLT : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SETVOLT(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SETVOLT(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SETVOLT(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SetVoltage(string channel, string voltage)	
	CommandSETVOLT(string channel, string voltage)	

2.3.9.GETCURR

Class		
	Item	Description
	function	Get the value of a current sensor channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class GETCURR : ArtBaseCommand
	service logic	

data		
	Name	Description
	string currentValue	

function		
	Name	Description
	GETCURR(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	GETCURR(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GETCURR(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GetCurrent(string channel, ref string data)	
	CommandGETCURR(string channel)	

2.3.10. SETCURR

Class		
	Item	Description
	function	Set the resister value for each current sensor channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class SETCURR : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SETCURR(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SETCURR(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SETCURR(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSETCURR(string channel, string resisterValue)	

2.3.11. GETFRQ

Class		
	Item	Description
	function	Get the value from a frequency input.
	N AMESPACE	ArtLogicsBasic
	statement	public class GETFRQ : ArtBaseCommand
	service logic	

data		
	Name	Description
	string valueFrequency	
	string valueDutyCycle	

function		
	Name	Description
	GETFRQ(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	GETFRQ(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GETFRQ(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	GetFrequency(string channel, string FrequencyRange, string Window, ref string frequency, ref string dutycycle)	
	CommandGETFRQ(string channel, string FrequencyRange, string Window)	

2.3.12. SETFRQ

Class		
	Item	Description
	function	Set a frequency value to a frequency output channel.
	N AMESPACE	ArtLogicsBasic
	statement	public class SETFRQ : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SETFRQ(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SETFRQ(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SETFRQ(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSETFRQ(string channel, string Frequency, string DutyCycle, string VoltH, string VoltL)	

2.3.13. ETHIP

Class		
	Item	Description
	function	G et/Set ip, gateway, mask.
	N AMESPACE	ArtLogicsBasic
	statement	public class ETHIP : ArtBaseCommand
	service logic	

data		
	Name	Description
	string currentIp	
	string currentGatew ay	
	string currentMask	

function		
	Name	Description
	ETHIP(ARTCommunication comHelper, string devic eid, string unitid, SetStatusCallback statusCallback = null)	
	CommandSETIP(string iptype, string ipaddr)	
	CommandGETIP()	

2.3.14. MSGRX

Class

	Item	Description
	function	Read a message from the specified communication port.
	N AMESPACE	ArtLogicsBasic
	statement	public class MSGRX : ArtBaseCommand
	service logic	

data

	Name	Description

function

	Name	Description
	MSGRX(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	MSGRX(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	MSGRX(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandMSGRXByBufferName(string resource, string channel, string MessageName, string ByteNumber, string BufferName, string NOFEEDBACK)	
	MsgRX(string resource, string channel, string MessageName, string ByteNumber, string NEWESTMSG)	
	CommandMSGRX(string resource, string channel, string MessageName, string ByteNumber, string NEWESTMSG)	
	CommandCANClearMsg(string resource, string channel, string ClearMode)	

2.3.15. MSGTX

Class		
	Item	Description
	function	Write a message in the specified communication port.
	N AMESPACE	ArtLogicsBasic
	statement	public class MSGTX : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	MSGTX(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	MSGTX(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	MSGTX(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandCANTXByBufferName(string channel, string MessageName, string BufferName, string NOFEEDBACK)	
	CommandCANTXData(string channel, string MessageName, string Data, string LoopCmd, string LoopNb, string Interval)	
	CommandCANTXAbort(string channel, string AbortParam)	
	CommandLINTXData(string channel, string MessageName, string Data)	
	CommandKLINETXConfig(string symbol)	
	CommandKLINETXData(string MsgByteFlow)	

2.3.16. CALBRT

Class		
	Item	Description
	function	G et/Set calibration data.
	N AMESPACE	ArtLogicsBasic
	statement	public class CALBRT : ArtBaseCommand
	service logic	

data		
	Name	Description
	string currentScale	
	string currentOffset	

function		
	Name	Description
	CALBRT(ARTCommunication comHelper, string dev iceid, string unitid, SetStatusCallback statusCallbac k = null)	
	CommandCALBRT(string channel)	
	CommandCALBRTScale(string channel, string scal e)	
	CommandCALBRTOffset(string channel, string offse t)	

2.3.17. CONFIGAWG

Class		
	Item	Description
	function	Config the AW message related information.
	N AMESPACE	ArtLogicsBasic
	statement	public class CONFIGAWG : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CONFIGAWG(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CommandCONFIGAW(string channel, string waveformIndex, string partNb, string maxPart, string samples)	
	CommandCONFIGSIGNAL(string channel, string signalIdx, string partNb, string maxPart, string samples)	
	CommandSETAWG(string channel, string mode, string Frequency, string Amplitude, string Offset, string DutyCycle, string WaveformIdx, string SignalCmd, /*string SignalIdx,*/ string TimeBase, string Start Point, string NbPointToLoop)	
	CommandSTOP(string channel)	

2.3.18. CONFIGCAN

Class		
	Item	Description
	function	Config the CAN message related information.
	N AMESPACE	ArtLogicsBasic
	statement	public class CONFIGCAN : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CONFIGCAN(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CONFIGCAN(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CONFIGCAN(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandCONFIGNominal(string channel, string nominalValue)	
	CommandCONFIGWithData(string channel, string nominalValue, string nominalSP, string dataValue, string dataSP)	
	CommandCONFIGResistance(string channel, string resValue)	
	CommandCONFIGStandardTx(string channel, string messageName, string idType, string canId, string RMF, string FD, string BRS)	

	CommandCONFIGStandardRx(string channel, string messageName, string idType, string canId, string RMF, string FD, string BRS)	
	CommandCONFIGTpisoByChannel(string channel, string receptionMode, string maxWaitFC, string padding, string FCTimeout, string CFTimeout)	
	CommandCONFIGTpisoTxByChannel(string channel, string messageName, string CANmode, string canId, string msgExtensionType, string MsgAddExtension, string FCId, string FCExtensionType, string FCAddExtension)	
	CommandCONFIGTpisoRxByChannel(string channel, string messageName, string CANmode, string canId, string msgExtensionType, string MsgAddExtension, string FCId, string FCExtensionType, string FCAddExtension)	
	CommandCONFIGTpisoTxByMessage(string channel, string messageName, string CANmode, string canId, string msgExtensionType, string MsgAddExtension, string FCId, string FCExtensionType, string FCAddExtension, string receptionMode, string MaxWaitFC, string Padding, string FCTimeout, string CFTimeout)	
	CommandCONFIGTpisoRxByMessage(string channel, string messageName, string CANmode, string canId, string msgExtensionType, string MsgAddExtension, string FCId, string FCExtensionType, string FCAddExtension, string receptionMode, string MaxWaitFC, string Padding, string FCTimeout, string CFTimeout)	
	CommandCONFIGVirtualCar(string channel, string CanNode, string MsgID, string PaddingByte, string CFTime)	

2.3.19. CONFIGKLINE

Class		
	Item	Description
	function	Config the KLINE message related information.
	N AMESPACE	ArtLogicsBasic
	statement	public class CONFIGKLINE : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CONFIGKLINE(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CONFIGKLINE(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CONFIGKLINE(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandCONFIGBaudrate(string Baudrate)	
	CommandCONFIGRunningMode(string runningMode, string TIdle, string TWUP, string P1, string P2, string P3, string P4)	
	CommandCONFIGMessageMode(string messageMode, string Data, string TTPInterval)	

2.3.20. CONFIGLIN

Class		
	Item	Description
	function	Config the LIN message related information.
	N AMESPACE	ArtLogicsBasic
	statement	public class CONFIGLIN : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CONFIGLIN(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CONFIGLIN(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CONFIGLIN(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandCONFIG(string channel, string runningMode, string Version, string BaudRate, string SBFLength)	
	CommandCONFIGTX(string channel, string MessageName, string LinMode, string LinId)	
	CommandCONFIGRX(string channel, string MessageName, string LinMode, string LinId, string MsgLength)	

2.3.21. CONFIGSAMPLING

Class		
	Item	Description
	function	Configure Sampling Acquisitions
	N AMESPACE	ArtLogicsBasic
	statement	public class CONFIGSAMPLING : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CONFIGSAMPLING(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	CommandCHANNEL(CHANNEL channelParam)	
	CommandWINDOW(WINDOW winParam)	
	CommandSTART(string channel, string mode, SetDataCallback setData)	
	CommandSTOP( )	
	CommandSTOP(string channel)	

2.3.22. TSTRT

Class		
	Item	Description
	function	Notify the TCU that the configuration state is over and test will be launched.
	N AMESPACE	ArtLogicsBasic
	statement	public class TSTRT : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	TSTRT(ARTCommunication comHelper, string devic eid, string unitid, SetStatusCallback statusCallback = null)	
	TSTRT(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	TSTRT(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusC allback = null, string dataDirectory = "")	
	Tstrt()	
	CommandTSTRT()	

2.3.23. TSTOP

Class		
	Item	Description
	function	Ask the TCU to clear all previous configurations.
	N AMESPACE	ArtLogicsBasic
	statement	public class TSTOP : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	TSTOP(ARTCommunication comHelper, string devic eid, string unitid, SetStatusCallback statusCallback = null)	
	TSTOP(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	TSTOP(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusC allback = null, string dataDirectory = "")	
	Stop()	
	CommandTSTOP()	

2.3.24. SYNCHRO

Class		
	Item	Description
	function	Synchronization module
	N AMESPACE	ArtLogicsBasic
	statement	public class SYNCHRO : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SYNCHRO(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SYNCHRO(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SYNCHRO(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSYNCHROStart()	
	CommandSYNCHROStop()	
	CommandSYNCHROState()	

2.3.25. STORAGE

Class		
	Item	Description
	function	C ontrol SD CARD
	N AMESPACE	ArtLogicsBasic
	statement	public class STORAGE : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	STORAGE(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	STORAGE(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	STORAGE(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSTORAGEStart()	
	CommandSTORAGEStop()	
	CommandSTORAGESize()	
	CommandSTORAGEStopSaving()	

2.3.26. SYSERR

Class		
	Item	Description
	function	Diagnostic.
	N AMESPACE	ArtLogicsBasic
	statement	public class SYSERR : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SYSERR(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SYSERR(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SYSERR(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSYSERR()	

2.3.27. SYSTIME

Class		
	Item	Description
	function	System time
	N AMESPACE	ArtLogicsBasic
	statement	public class SYSTIME : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	SYSTIME(ARTCommunication comHelper, string deviceid, string unitid, SetStatusCallback statusCallback = null)	
	SYSTIME(string ipAddr, int ipPort, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	SYSTIME(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CommandSYSTIME()	

2.4. The TCU protocol handles the API

Class		
	Item	Description
	function	Unified available command objects for manage TCU
	N AMESPACE	ArtLogicsBasic
	statement	public class ArtTCU
	service logic	

data		
	Name	Description

function		
	Name	Description
	ArtTCU(ARTCommunication comHelper, string unitid = "11", ArtBaseCommand.SetStatusCallback statusCallback = null)	
	ArtTCU(string ipAddr, int ipPort, string unitid = "11", ArtBaseCommand.SetStatusCallback statusCallback = null, string dataDirectory = "")	
	ArtTCU(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, string unitid = "11", ArtBaseCommand.SetStatusCallback statusCallback = null, string dataDirectory = "")	
	CALBRT ObjCALBRT()	
	CLRDIG ObjCLRDIG()	
	CONFIGAWG ObjCONFIGAWG()	
	CONFIGCAN ObjCONFIGCAN()	
	CONFIGKLINE ObjCONFIGKLINE()	
	CONFIGLIN ObjCONFIGLIN()	
	ETHIP ObjETHIP()	
	GETCURR ObjGETCURR()	

	GETDIG ObjGETDIG()	
	GETFRQ ObjGETFRQ()	
	GETVOLT ObjGETVOLT()	
	HELLO ObjHELLO()	
	MSGRX ObjMSGRX()	
	MSGTX ObjMSGTX()	
	RELAY ObjRELAY()	
	SETCURR ObjSETCURR()	
	SETDIG ObjSETDIG()	
	SETFRQ ObjSETFRQ()	
	SETVOLT ObjSETVOLT()	
	STORAGE ObjSTORAGE()	
	SYNCHRO ObjSYNCHRO()	
	SYSERR ObjSYSERR()	
	SYSID ObjSYSID()	
	SYSTIME ObjSYSTIME()	
	TSTOP ObjTSTOP()	
	TSTRT ObjTSTRT()	

3. ART Process

3.1. Summary description

A mission profile is composed of several steps containing a n umber of commands.When running, those steps will trigger with a fixed time period, and all the commands associated to the step will execute one after the other.The process will go back to the f irst step after the last step is executed.

The TCU can define up to 32 mission profiles at the same tim e, and any number of commands can be defined in a mission pro file as long as the dedicated memory has enough room (the spac e taken in memory depends on the board and the type of comm and added).

3.2. P rocess Syntax

Declare a new process.The command PROCESS=<id>,DEFINE, <granularity>,<totalsteps> creates a new mission profile, associa tes an ID to it, sets a granularity (time between two steps) and the total number of steps.

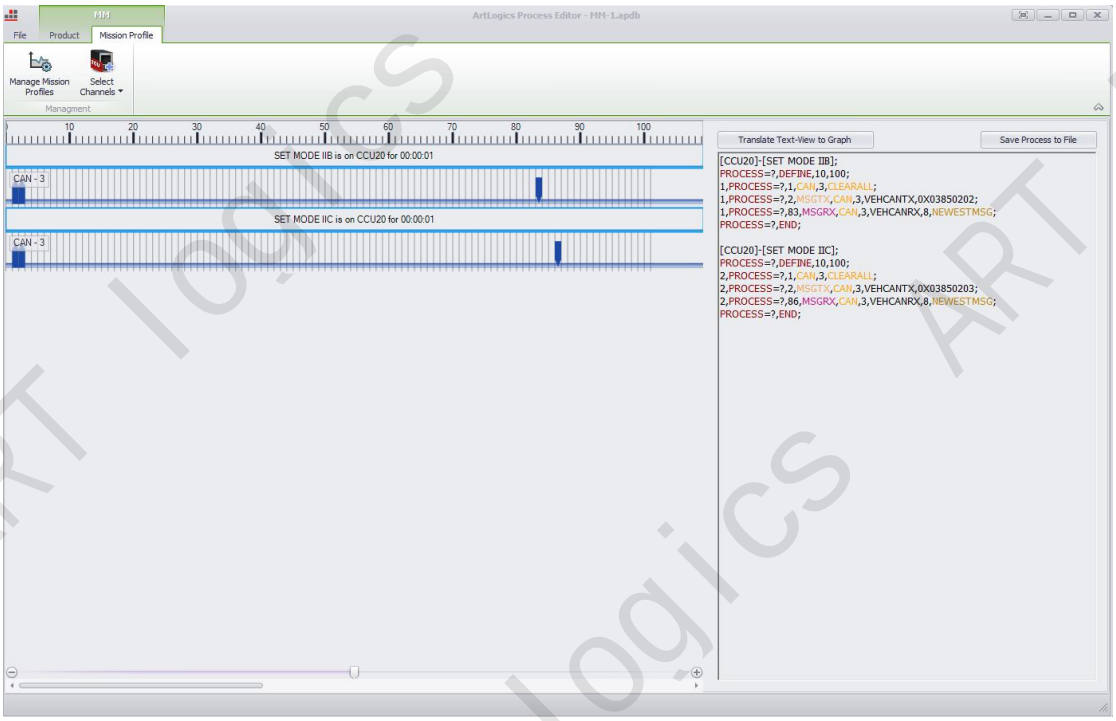
Add commands.PROCESS=<id>,<stepidx>,<cmdtype>,<cmdp arameters> add a new command in the mission profile at the ste p <stepidx>.

End the definition.Finally the command PROCESS=<id>,END command must be sent for finalizing the definition of the mission profile.This is necessary for allowing it to start running.

Running and fetching results.Defined mission profiles are runni ng using the commands PROCESS=START and PROCESS=STOP.W hen at least one loop has been completed it becomes possible to query the results of the commands with PROCESS=RESULT or P ROCESS=RESX.The response of those commands will be a seque nce of result values, in their order of definition.RESULT response will be standard ASCII text, and RESX response in an optimized binary format useful for heavy mission profiles.

3.3. The P rocess Editor

A rtLogics Process Editor Software can edit the p rocess and write the complete p rocess to export to the file.



3.4. ARTProcessPushOpr

Class		
	Item	Description
	function	Configure p rocess and obtain p rocess data
	N AMESPACE	ArtLogicsBasic.Operation
	statement	public class ARTProcessPushOpr
	service logic	

data		
	Name	Description

function		
	Name	Description
	ARTProcessPushOpr(ARTCommunication comm, str ing dataDirectory = "")	
	ARTProcessPushOpr(string ipAddr, int ipPort, string dataDirectory = "")	
	ARTProcessPushOpr(string portName, ARTBaudrat e baudRate = ARTBaudrate.B230400, string dataDi rectory = "")	
	Dispose()	
	SetProcess(ArtBaseCommand.HandlerCallback han dler, string unitid = "11", string processid = "1", str ing totalStep = "10", string granularity = "100")	
	SetProcess(string filePath, string processid = "", str ing unitid = "11", ArtBaseCommand.HandlerCallbac k handler = null, ArtBaseCommand.SetStatusCallba ck statusCallback = null)	
	SetProcess(List<string> procFileLines, string proces sid = "", string unitid = "11", ArtBaseCommand.Ha ndlerCallback handler = null, ArtBaseCommand.Set StatusCallback statusCallback = null)	

	AddCommand(string commandName, string commandParam, int stepNumber)	
	ProcessDefine(ArtBaseCommand.SetStatusCallback statusCallback = null)	
	ProcessStart(ArtBaseCommand.SetStatusCallback statusCallback = null)	
	ProcessStop(ArtBaseCommand.SetStatusCallback statusCallback = null)	
	GetPacketCount()	
	GetPacket()	
	GetPacket(ref string datas)	

4. A RT CAN Push

4.1. Summary description

Using the defined commands, the user can configure the bau drate of the bus, send and receive CAN messages.

The device automatically send the messages received to the h ost.

4.2. CANPUSH

Class		
	Item	Description
	function	Control of the C AN PUSH function
	N AMESPACE	ArtLogicsBasic
	concrete class	public class CANPUSH : ArtBaseCommand
	service logic	

data		
	Name	Description

function		
	Name	Description
	CANPUSH(ARTCommunication comHelper, string d eviceid, string unitid, SetStatusCallback statusCallb ack = null)	
	CommandCONFIG(string channel, string baudrate = "500K", string baseDirectory="", SetDataCallbac k dataHandler = null)	
	CommandSTART()	
	CommandSTOP()	
	StopToGetData()	

4.3. ARTCANPushOpr

Class		
	Item	Description
	function	Control C AN PUSH to obtain CAN port data
	N AMESPACE	ArtLogicsBasic.Operation
	statement	public class ARTCANPushOpr
	service logic	

data		
	Name	Description

function		
	Name	Description
	ARTCANPushOpr(ARTCommunication comm, string dataDirectory = "")	
	ARTCANPushOpr(string ipAddr, int ipPort, string da taDirectory = "")	
	ARTCANPushOpr(string portName, ARTBaudrate b audRate = ARTBaudrate.B230400, string dataDirect ory = "")	
	Config(string channel, string baudrate = "500K")	
	Start(string channel = "")	
	Stop(string channel = "")	
	GetPacketCount(string channel)	
	GetPacket(string channel)	
	GetPacket(string channel, ref int id, ref string form at)	

5. ART Fast Sampling

5.1. Summary description

The sampling mode of the acquisition is used to measure the voltage channels continuously or for a time window, with a sampling rate up to 100 kHz.To use this mode, the user needs to configure first the acquisition parameters and once the configuration is set and validated, the sampling operation can be started and stopped, with the voltage data values being pushed automatically to the host or saved in the SD Card in the stand alone mode.

5.2. ARTSamplingOpr

Class		
	Item	Description
	function	Configure the sampling to obtain the sampling data
	N AMESPACE	ArtLogicsBasic.Operation
	statement	public class ARTSamplingOpr
	service logic	

data		
	Name	Description

function		
	Name	Description
	ARTSamplingOpr(ARTCommunication comm, string dataDirectory = "")	

	ARTSamplingOpr(string ipAddr, int ipPort, string dataDirectory = "")	
	ARTSamplingOpr(string portName, ARTBaudrate baudRate = ARTBaudrate.B230400, string dataDirectory = "")	
	Dispose()	
	SetConfiguration(CONFIGSAMPLING.CHANNEL cfg channel, ArtBaseCommand.SetStatusCallback statusCallback, CONFIGSAMPLING.SetDataCallback setData)	
	SetConfiguration(string channel, string PeriodUnit, string AcqPeriod, string Speed, string TriggerSource = "", string TriggerType="", string Level = "")	
	StartToGetData(ArtBaseCommand.SetStatusCallback statusCallback = null)	
	Stop(ArtBaseCommand.SetStatusCallback statusCallback = null)	
	ReadyToGetData()	
	StopToGetData()	
	GetPacketCount(string channel)	
	GetPacket(string channel, ref string speed, ref int dataLen)	