WARNING

THE INSTALLATION SHALL BE MADE BY QUALIFIED INSTALLATION PERSONNEL AND SHOULD CONFORM TO ALL NATIONAL AND LOCAL CODES

Hi5a / Hi5 Controller Function Manual

HRRpcProxy v2.0.0

HD HYUNDAI ROBOTICS

Printed in Korea – Jun. 2023. 2nd Edition

List of Figures

1

About HRRpcProxy

1.1. Concept

Hi5/Hi5a controller can be remotely monitored or controlled via HR-RPC (Hyundai Robot – Remote Procedure Call) protocol. Hi5/Hi5a controller provide with HR-RPC on Ethernet UDP/IP layer on almost every feature of it.

But implementing HR-RPC protocol is difficult and bug-prone, Hyundai Robot provide with a library HRRpcProxy doing the HR-RPC communication for a host application. Using this library you can make your application remotely monitor or control Hi5/Hi5a robot controller, easily and rapidly.

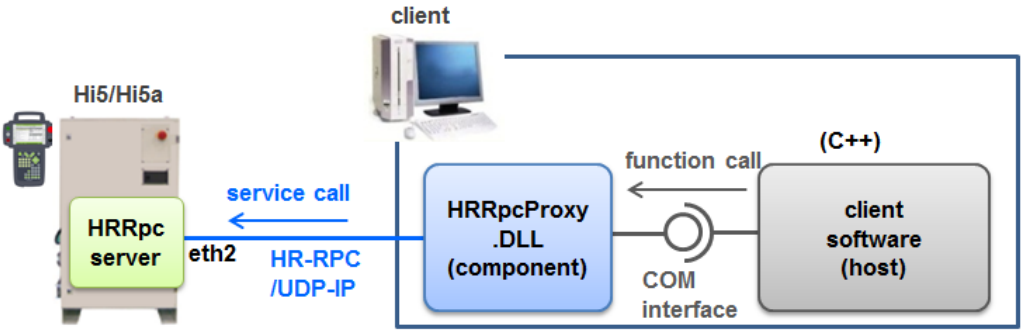


Figure 1.1 Concept of HRRpcProxy

1.2. Environment for development & execution

- Library type: Microsoft COM DLL/Win32
- Environment for execution: Windows XP/Vista/7/10
- Sample program 1: Visual Studio 2008 project, C++/MFC dialog-based application
- Sample program 2: Visual Studio 2008 project, C++/Win32 console application
- Required robot controller version: Hi5 V32.16-00 ~, Hi5a all versions

1.3. Files

folder	Contents
inc	**[interface header files]** • IBase.h ; interface for HR-RPC service-call • IState.h ; State Control and Monitoring interface • IMngFile.h ; File Manager interface (remote copy of controller files) • IProgress.h ; Progress interface. (for transferring of progress of file copy) • IPose.h ; interface for POSE data transfer • ErrCodeRef.h ; error code macro definitions
bin	**[library file]** • HRRpcDll.dll ; dynamic linking library (COM component)
doc	• Hi5RPCPX~.pdf ; basic manual • HRRpcProxy.chm ; reference manual (compiled HTML help file)
HRRpcDllCli	**[sample 1 – MFC dialog-based application]** • HRRpcCliDlg.h, HRRpcCliDlg.cpp ; main dialog class. Setting IP address, initialize, and end the connection. • DlgJoyJog.h, DlgJoyJog.cpp ; remote jog related example code. • DlgMngFile.h, DlgMngFile.cpp ; file-transfer related example code. • DlgState.h, DlgState.cpp ; getting/setting state example code. • HResMsg.h, HResMsg.cpp ; error code – message string map • HRRpcCli.h, HRRpcCli.cpp ; application class • JoyJog.h ; structure to easily access to jog-bits • PushButton.h, PushButton.cpp ; button class for jog • ProgressShow.h, ProgressShow.cpp ; example of implementation of IProgressShow for displaying remote file copy • HRRpcCli_Gb.cpp ; global functions • COM/Loose_COM.h, Loose_COM.cpp ; helper function to use COM component without registering.
HRRpcMonAx	**[sample 2 – Win32 console application]** • MonAx.h, MonAx.cpp ; joint, TCP monitoring example • main.cpp ; main entry • COM/Loose_COM.h, Loose_COM.cpp ; helper function to use COM component without registering.

1.4. Example execution screens

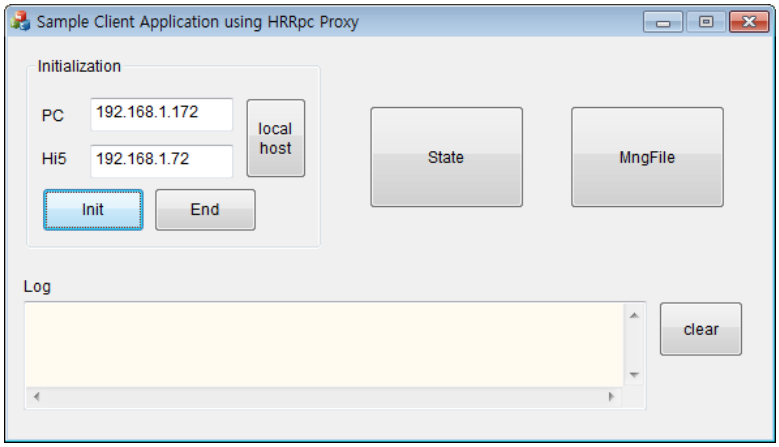


Figure 1.2 Main dialog of sample program

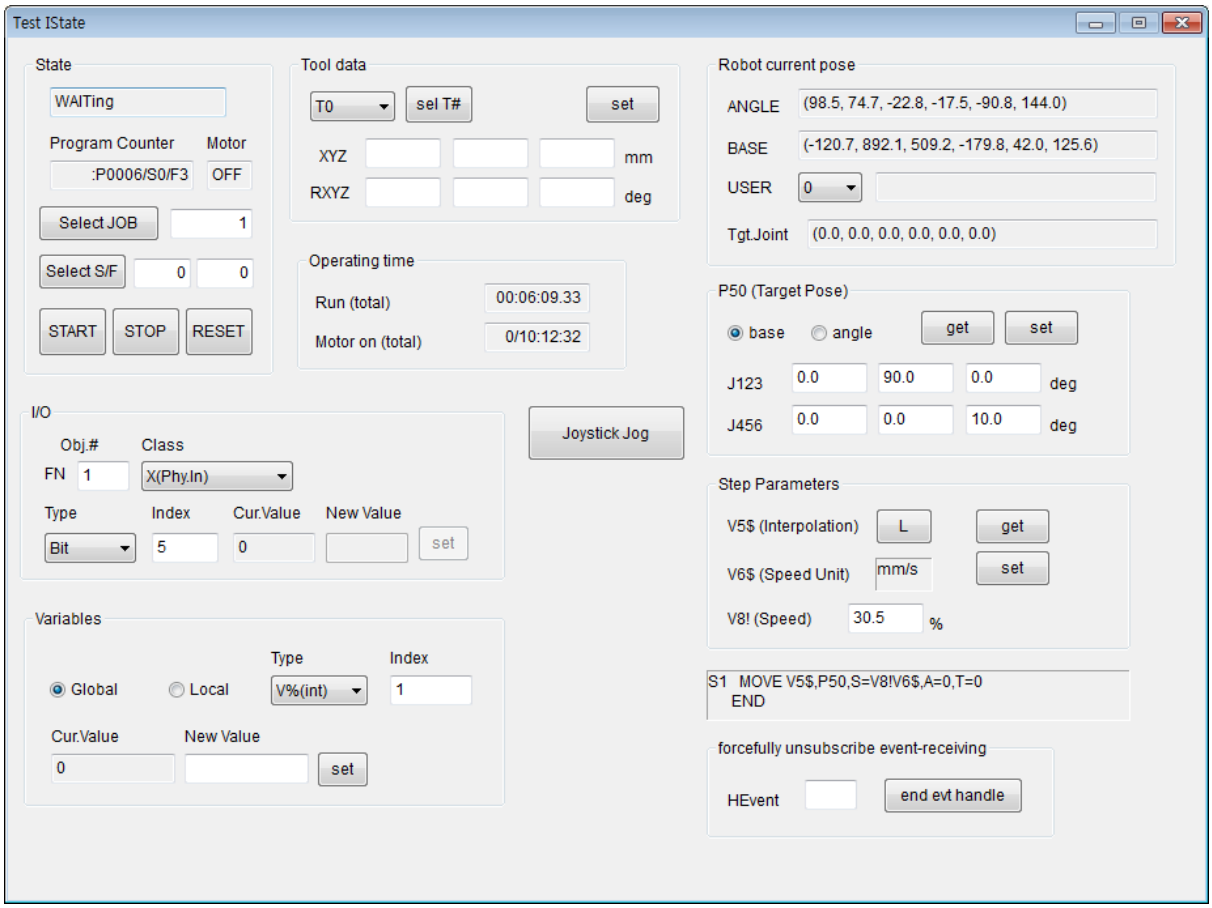


Figure 1.3 Dialog-box testing IState

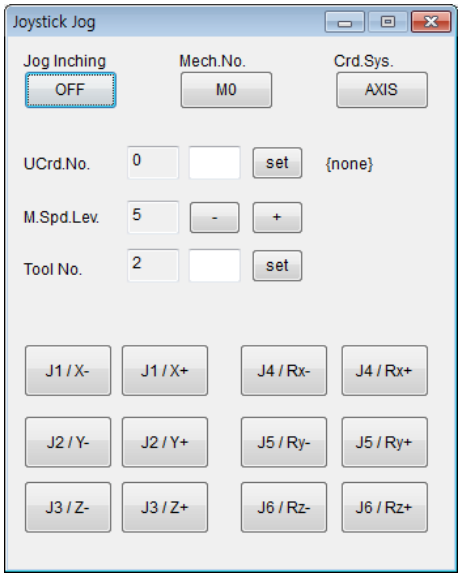


Figure 1.4 Dialog-box testing IState::JoystickJog()

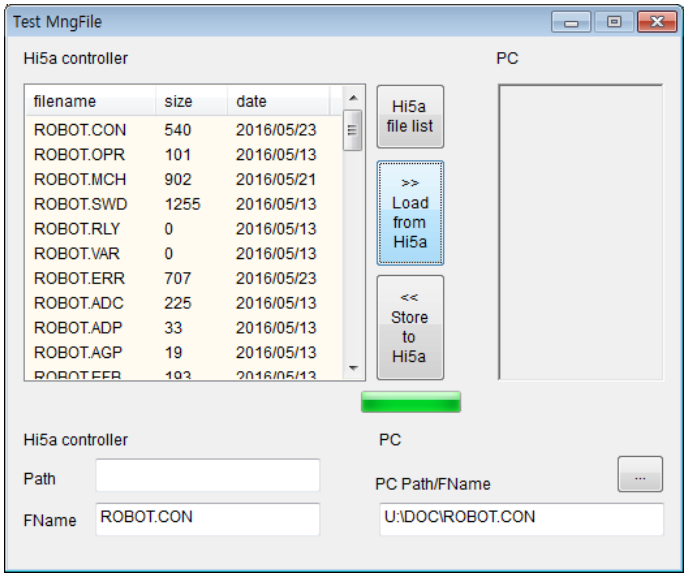
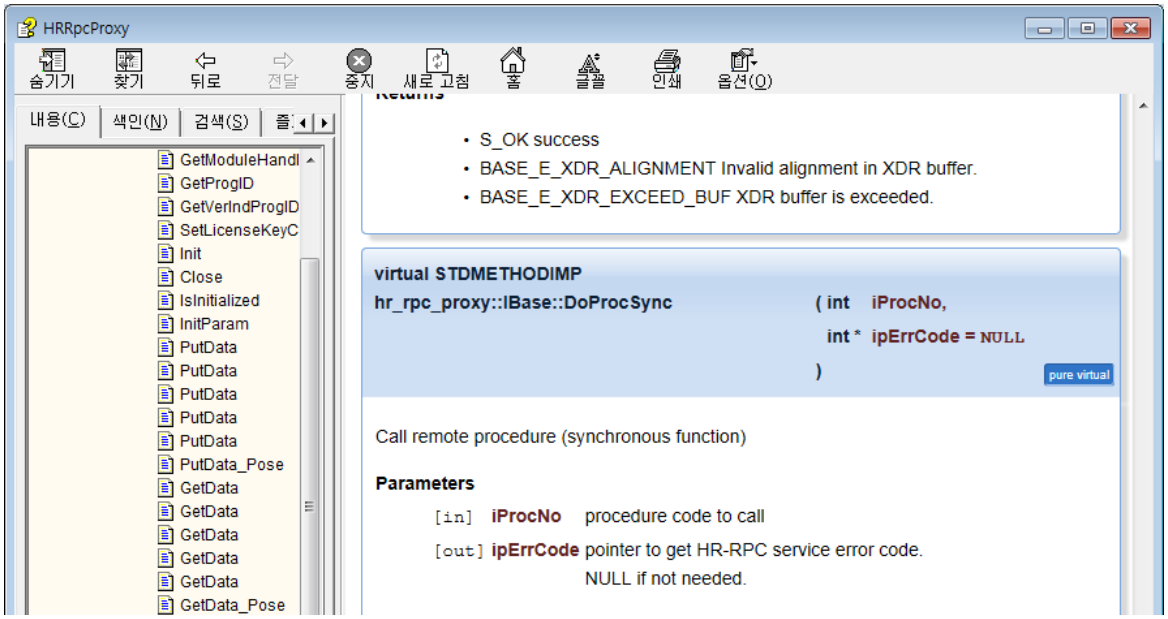


Figure 1.5 Dialog-box testing IMngFile


HRRpcProxy
숨기기
찾기
뒤로
전달
중지
새로 고침
홈
글꼴
인쇄
옵션(O)
내용(C)
색인(N)
검색(S)
GetModuleHandl
GetProgID
GetVerIndProgID
SetLicenseKeyC
Init
Close
IsInitialized
InitParam
PutData
PutData
PutData
PutData
PutData
PutData_Pose
GetData
GetData
GetData
GetData
GetData
GetData_Pose
S_OK success
BASE_E_XDR_ALIGNMENT Invalid alignment in XDR buffer.
BASE_E_XDR_EXCEED_BUF XDR buffer is exceeded.
virtual STDMETHODIMP
hr_rpc_proxy::IBase::DoProcSync (int iProcNo,
int * ipErrCode = NULL
)
pure virtual
Call remote procedure (synchronous function)
Parameters
[in] iProcNo procedure code to call
[out] ipErrCode pointer to get HR-RPC service error code.
NULL if not needed.

2

How to use

2. How to use

2.1. Class diagram

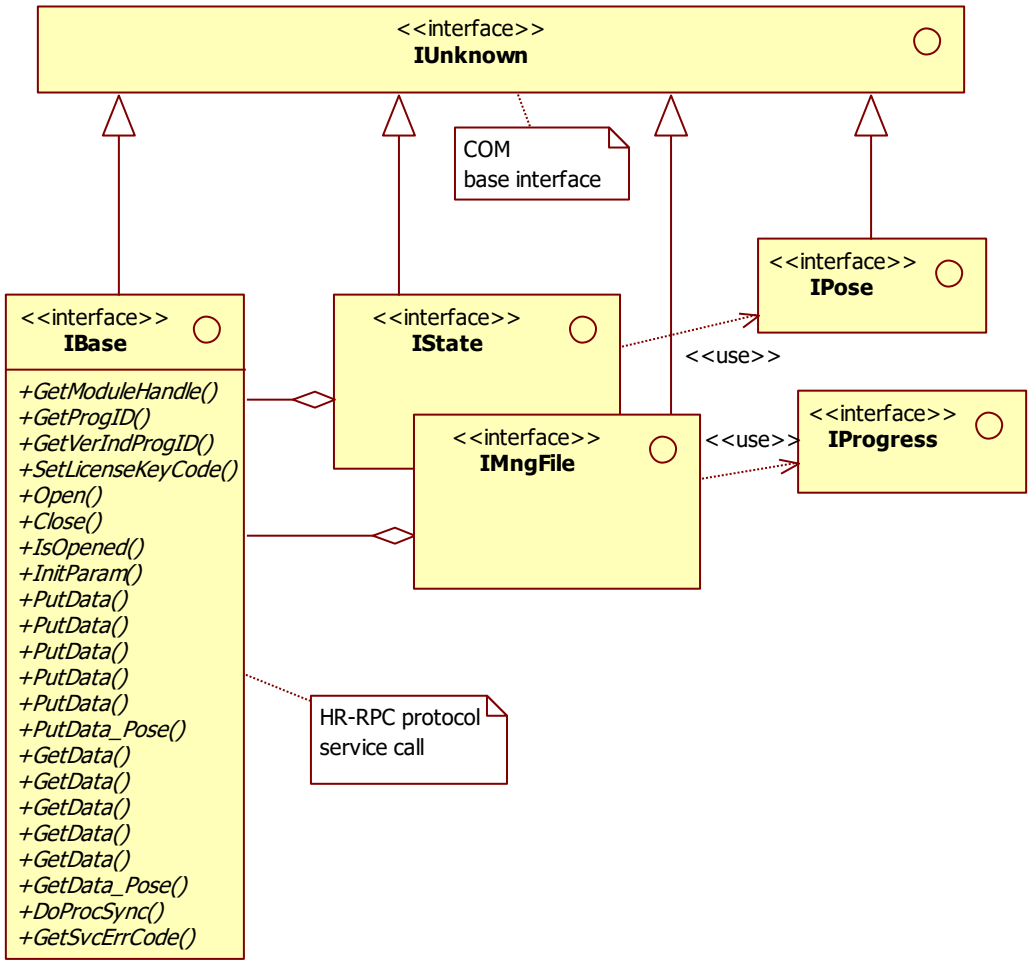


Figure 2.1 Class diagram of HRRpcProxy interfaces

2.2. Rough procedure

a) Startup WinSock.
b) Create Base COM component, and query IBase interface.
c) Input license key code with IBase::SetLicenseKeyCode()
d) IBase::Open( ) with local and remote IP address.
e) Call HR-RPC services.
f) (optional) Create helper components and query their interfaces (IState or IMngFile).
g) (optional) Set IBase pointer into helper components; (SetProxyBase())
h) (optional) Call helper component's member functions.
i) (optional) Release helper components.
j) IBase::Close( );
k) Release Base COM component.
l) Cleanup WinSock.

2.3. Startup / Cleanup WinSock

While initialize you host application startup WinSock. And before exit the application cleanup the WinSock. (If you use MFC library, you may call AfxSocketInit() instead of these.

```
// WINSOCK initialization
WORD wVersionRequested = MAKEWORD(2, 2);          // request up to WinSock version v2.2
WSADATA wsaData;
int iret = WSAStartup(wVersionRequested, &wsaData);
if(iret != 0)
{
        AfxMessageBox(IDP_SOCKETS_INIT_FAILED);
        return FALSE;
}
```

```
// Deregister itself from a Windows Sockets implementation
WSACleanup();
```

2.4. Creating / Releasing COM component

Generally, when you use COM component, beforehand you have to register the component to the system. You can do this with below command in command-line window.

registering	regsvr32 HRRpcProxy.dll
unregistering	regsvr32 –u HRRpcProxy.dll

But, the sample program provide a function named CreateInstanceFromInst(). With this function, you can create unregistered COM component and query the interface you want. If you use this way, you need to do LoadLibrary(HRRpcProxy.dll) on you own. Refer to the source code related to this function.

(This sample shows loading unregistered COM component. And of course, you can register the component, and use COM API function ::CoCreateInstance( ) to get the interface you want.)

```
// ----------------------------------------
// DLL dynamic linking
strPathFName = "HRRpcProxy.dll";
if((h_module_ = LoadLibrary(strPathFName)) == NULL) {
        // error handling..
        return;
}

// ----------------------------------------
// create COM component instance & get interface

// IBase
HRESULT hr = CreateInstanceFromInst(h_module_, hr_rpc_proxy::CLSID_Base
                        , hr_rpc_proxy::IID_IBase, (void**)&p_base_);
if(FAILED(hr)) {
        // error handling..
        return;
}
```

```
p_base_->Release();

FreeLibrary(h_module_);
```

2.5. License

To call IBase::Connect() without license error, you need to input license key code. For a legal license code, please contact to Hyundai Robot with the target system's MAC address (Physical Address).
You can check the MAC address with below command in command-line window.

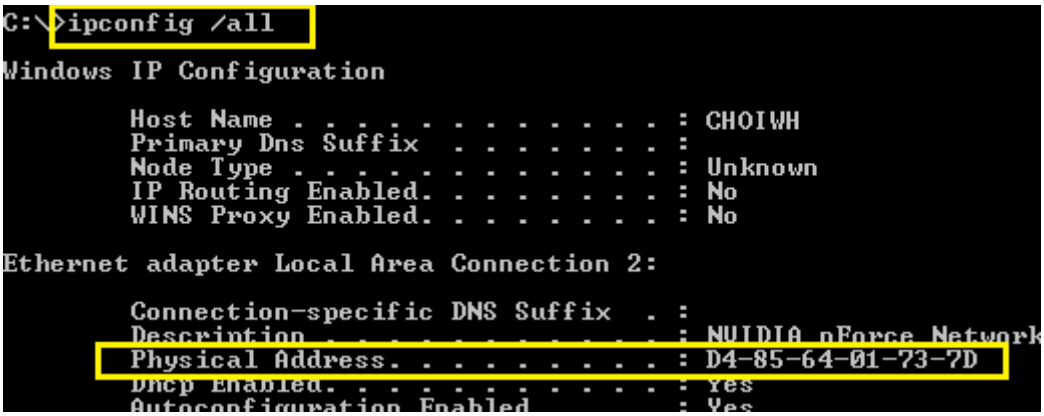


Figure 2.2 Getting MAC address

Then call IBase::SetLicenseKeyCode() with the provided key code and expiry date.

```
p_base_->SetLicenseKeyCode(12606732, 20180531);
```

2.6. Opening / Closing Base component

```
// Open
int         iErrCode;
hr = p_base_->Open(_bstr_t((LPCTSTR)strIpPc), _bstr_t((LPCTSTR)strIpRc), &iErrCode);
if(FAILED(hr)) {                                    // failure
            // error handling..
            return;
}
```

```
// Close
if(p_base_) {
            HRESULT hr = p_base_->Close();
}
```

2.7. Calling HR-RPC service with IBase

IBase has basic functions which call HR-RPC services. The procedure to call HR-RPC service is as below. You have to know exact list of input and output parameter and service number.

```
p_base->InitParam( );                              // get ready to insert input parameters
p_base->PutData( input param. 1 );
p_base->PutData( input param. 2 );
...
HRESULT hr = p_base->DoProcSysc( service number );  // call the service (blocked)
if(FAILED(hr)) {                                    // failure
        // error handling..
        return;
}
p_base->GetData( output param. 1 );            // extract output parameters
p_base->GetData( output param. 2 );
```

For error handling, you have to check the returned HRESULT value. HResMsg.cpp of sample source code contains some helper functions to do this. If hr==BASE_E_SERVICE, it means the Hi5/Hi5a robot controller has returned some service error code. You can check the code using IBase:: GetSvcErrCode() function. For some service error, you can make use of IBase::errcode2msg(), and the complete list of service errors are in ErrCodeRef.h

2.8. Helper components IState and IMngFile

For convenient and precise service-calls, IState and IMngFile provide member functions of frequently used service-call. Before use them, you have to bind IBase component into these component.

```
// ----------------------------------------
// Create COM component instances & get interface - IState
HRESULT hr = CreateInstanceFromInst(h_module_, hr_rpc_proxy::CLSID_State
                        , hr_rpc_proxy::IID_IState, (void**)&p_state_);
if(FAILED(hr)) {
        MessageBox("Error in getting IState.", 0, MB_OK | MB_ICONSTOP);
        return -1;
}

// bind the proxy base interface
p_state_->SetProxyBase(p_base_);
```

2.9. IState::JoystickJog

Prerequisite for using IState::JoystickJog( );

- In case of Hi5 robot controller, the main software should be V32.16-00 or later.
  In case of Hi5a robot controller, the main software should be V40.07-00 or later.
- [F2: System] – 4: Application parameter – 16: Joystick mode – Joystick mode : HR-RPC
- [F2: System] – 2: Control parameter – 2: Input/Output signal setting – 3: Input signal assign – Joystick mode : input signal #
- While jogging, the assigned input signal must be ON. When you don't jog the robot, turn off this signal for safety.
  (If it is hard to connect input signal for jog on/off, you can enable jog by inverting the signal value. ; [F2: System] – 2: Control parameter – 2: Input/Output signal setting – 1: Input signal attribute – register the signal # and check the 'Negative')
- The mode switch of teach pendant has to be [Automatic], not [MANUAL].

In remote jog example, Mech.No. is used only when additional axes mechanism is set-up, UCrd.No. is used only when more than 1 user coordinate system is defined. When you use additional axes or user coordinate system, please refer to Hi5/Hi5a operation manual.

● **Daegu Office (Head Office)**

50, Techno sunhwan-ro 3-gil, yuga, Dalseong-gun, Daegu, 43022, Korea

● **GRC**

477, Bundangsuseo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, Korea

● **대구 사무소**

(43022) 대구광역시 달성군 유가읍 테크노순환로 3 길 50

● **GRC**

(13553) 경기도 성남시 분당구 분당수서로 477

● **ARS : +82-1588-9997 (A/S center)**

● **E-mail : robotics@hyundai-robotics.com**