HIWIN®

XEG Dynamic-Link Library Command

User Manual

Original Instruction

www.hiwin.tw

Table of Contents

1. Introduction

HIWIN Electric Gripper Control APIs (HEGCA) are derived from HIWIN Electric Gripper Software (HEGS) and HEGCA is XEG series electric gripper command list. The command lists allow the user to program the designed part. The dynamic link library of HIWIN electric gripper can perform operations up to 16 (include) at the same time. For example, communication connection, disconnection, initialize electric gripper, control gripper movement, gripping, identification of gripping object, monitoring of gripper status. During communication, API used to return the gripper label value. During the operation, the user can operate the connected circuit gripper of any group according to the value of gripper label.

2. Safety Declaration

Before using the HIWIN XEG series Electric Gripper, be sure you have read this entire manual in detail and strictly follow all rules to ensure your safety during operation. The purpose of this safety declaration is to inform users to be cautious when operating the HIWIN XEG series Electric Gripper. Please use standard safety practices throughout all stages of operation. As an extra safety precaution, users should wear personal protective equipment to prevent against injury and unexpected equipment failure.

3. Description of Product

3.1. Hardware Outline

- Hardware is composed of:

Item	Name	Description
1	Electric Gripper	XEG-Series
2	Electric Gripper Controller	XEG-C1
3	Power Supply	DC24V, 0.5A
4	Actuator Cable	Connector between Controller (CN1) and Electric Gripper
5	Power Cable	Connector between Controller and power supply
6	USB Cable	Connector between Controller (CN2) and PC with USB
7	PC	With USB, need to check Serial comport

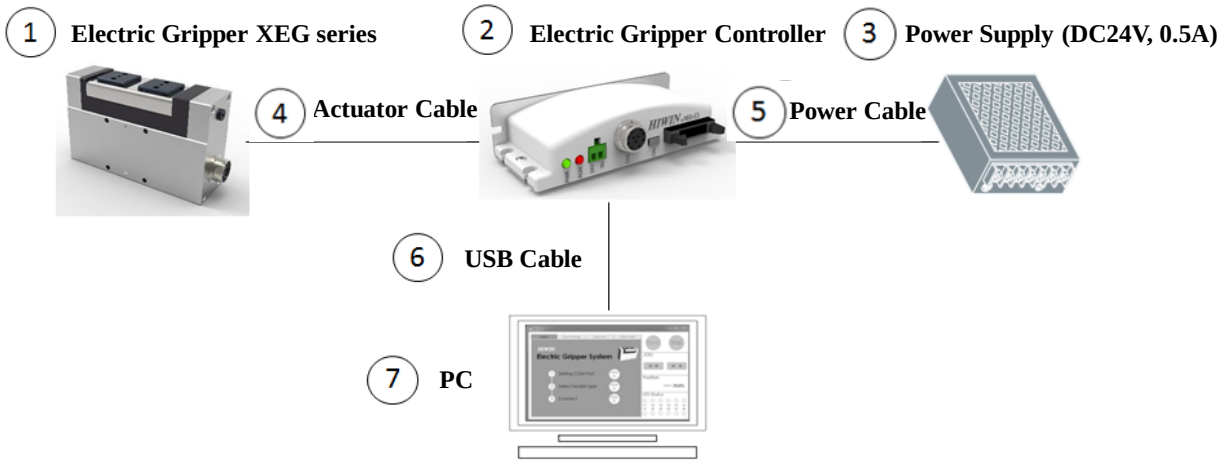
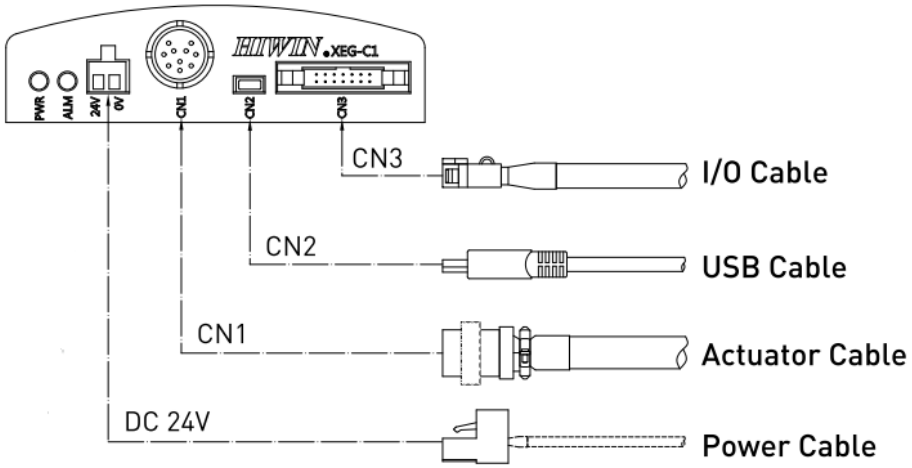


Fig.1 Hardware architecture



Name	Illustration
PWR	Power supply lamp (Green)
ALM	Error status lamp (Red)
24V/0V	Power supply terminal
CN1	Actuator terminal
CN2	Communication terminal
CN3	I/O connector

Fig.2 Electric Gripper controller connectors and functions

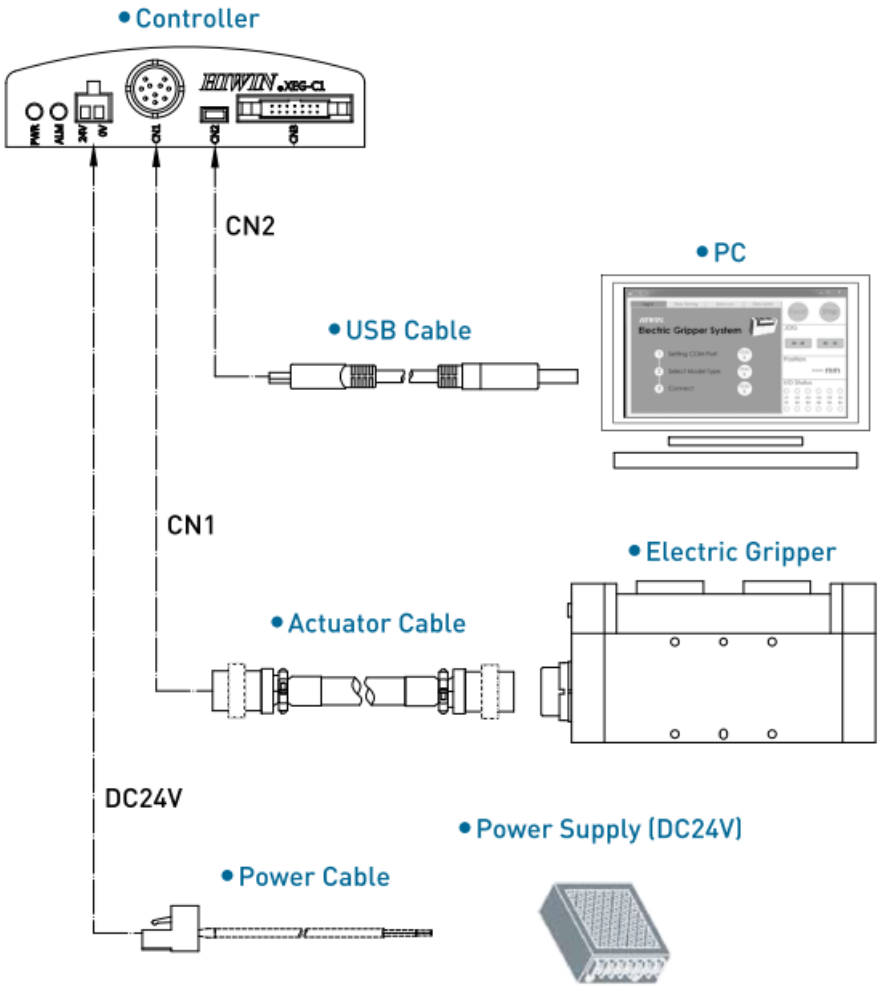


Fig.3 Electric Gripper system diagram

3.2. Software Outline

- Software is composed of:
  - Windows OS, Visual Studio, C++
  - After Electric Gripper controller firmware version 2.0.13.

3.3. HIWIN Electric Gripper Control APIs outline

- The HEGS, which control the Electric Gripper, includes:
  - Connection method
  - Connection fail detect
  - Connection close
  - Firmware version verification
  - Monitor IO Status; Work State; Busy state; Alarm state
  - Monitor Current Position
  - Stop Gripper
  - Reset Gripper
  - MOVE mode
  - GRIP mode
  - EXPERT mode

4. Program Command Introduction

4.1. Command Introduction

The HEGS, which offers 14 command, can directly control and monitor XEG series Electric Gripper for users.

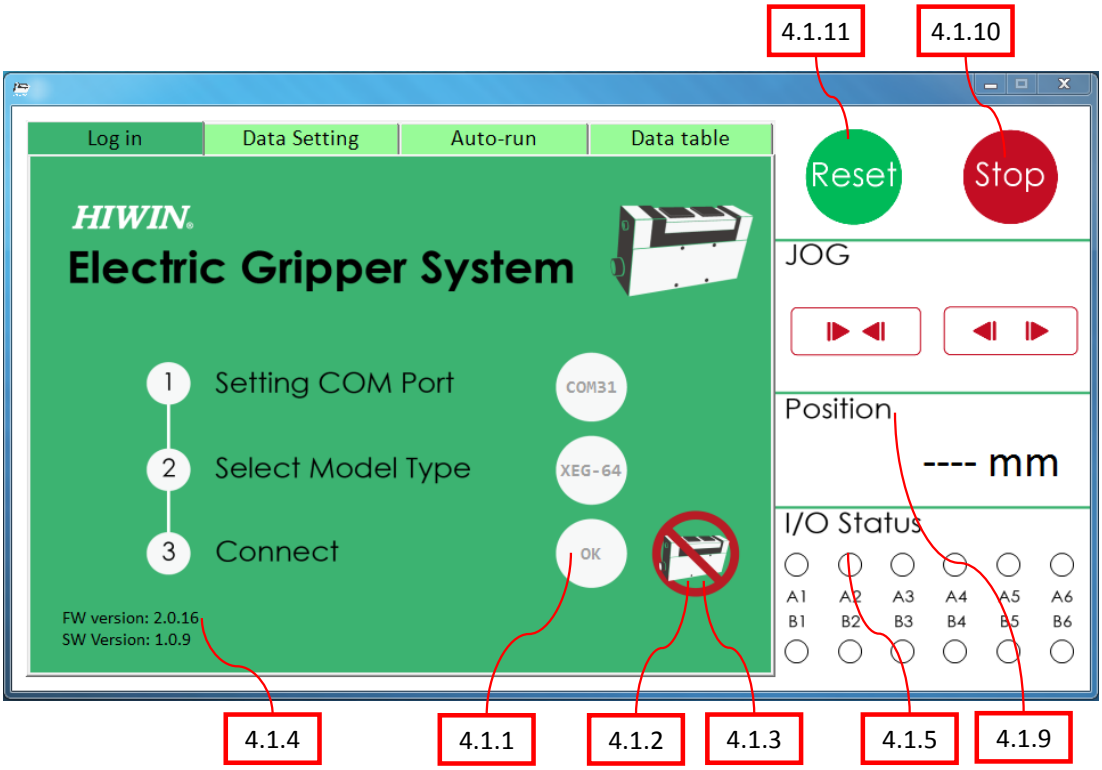


Fig 4 Electric Gripper Operation Software Interface (Main Page)

4.1.1 HEG StartConnect(int SettingComPort, int SelectModelType)

- Description:
  Set Comport and gripper model to which USB is connected
- Parameter:
  SettingComPort is the input value, for example, connected to COM1, the value is entered as 1 and the rest so on. Currently it supports from COM1~COM99;
  SelectModelType is the gripper model. If the model is XEG16, enter value 16, if the model is XEG32, enter value 32, whereas model XEG64, the value will be 64.

- Return:

HEG is an integer value (Integer)；

<100: Return id gripper label；

>1000: Abnormal, reference 4.2 error code.

- Example: ©

```
int id;
id = StartConnect(10, 32);
```

4.1.2 int DetectConnect(HEG id)

- Description:

  Detect current disconnection or not.

- Parameter:

  id is the model label.

- Return:

  0: This setting is connected to ComPort;

  >1000: Abnormal, reference 4.2 error code.

- Example: ©

```
int ErrorCode;
ErrorCode = DetectConnect(id);
```

4.1.3 int CloseConnect(HEG id)

- Description:

  Close the Comport to which the USB is connected.

- Parameter:

  id is the model label.

- Return:

  0: Successfully closed；

  >1000: Abnormal, reference 4.2 error code.

- Example: ©

```
int ErrorCode;
ErrorCode = CloseConnect (id);
```

4.1.4 int CurFirmwareVersion(HEG id, int &Ver1, int &Ver2, int &Ver3)

- Description:

  Return to current firmware version.

- Parameter:

id is the model label；

If the current version is 2.0.16, Ver1 is 2, Ver2 is 0, and Ver3 is 16.

- Return:

  0: Obtain version value successful；

  >1000: Abnormal, reference 4.2 error code.

- Example: ©

  int ErrorCode, Ver1, Ver2, Ver3;

  ErrorCode = CurFirmwareVersion (id, Ver1, Ver2, Ver3);

4.1.5 int IOStatus(HEG id, unsigned int &InputData, unsigned int &OutputData)

- Description:

  According to the description of the gripper software, it can be divided into BACK, BUSY and ERROR backhaul applications.

- Parameter:

  id is the model label；

  InputData represents in binary (A8)(A7)(A6)(A5)(A4)(A3)(A2)(A1) ；

  OutputData represents in binary (B8)(B7)(B6)(B5)(B4)(B3)(B2)(B1)。

- Return:

  0: Obtain status value successful；

  >1000: Abnormal, reference 4.2 error code.

- Remark description:

No.	I/O	Symbol	Function
A1	Input	IN0	Position data bit0
A2		IN1	Position data bit1
A3		IN2	Position data bit2
A4		IN3	Position data bit3
A5		IN4	Position data bit4
A6		START	Command input

A7,A8 no definition

No.	I/O	Symbol	Function
B1	output	BUSY	Executing command
B2		HOLD	Gripping range check
B3		ALM-CODE1	Error status number
B4		ALM-CODE2	

B5,B6,B7,B8 no definition

CODE1	CODE2	Error status
0	0	-
1	0	Position fault
0	1	Over travel
1	1	Original point fault

- Example: ©

```
int ErrorCode, InputData, OutputData;
ErrorCode = IOStatus (id, InputData, OutputData);
int sigBusy, sigHold, sigAlarm;
sigBusy = OutputData & 0x01;
sigHold = OutputData & 0x02;
sigAlarm = OutputData & 0x0C;
```

4.1.6 bool WorkState(HEG id, int &ErrorCode)

- Description:

  For the use of ResetMotion (...), RunMove (...), RunGrip (...), RunExpert (...) programs, it is strongly recommended to use this function to decide the completion of the work, and then proceed to the next instruction.

- Parameter:

  id is the model label；

  ErrorCode return：

  0: Obtain status successful；

  >1000: Abnormal, reference 4.2 error code

- Return:

  true: data setting and operation in progress；

  false: data setting and operation complete.

- Example: ©

```
int ErrorCode;
if( WorkState(id, ErrorCode) ); // operation in progess
else; //Idle state
```

4.1.7 bool HoldState(HEG id, int &ErrorCode)

- Description:

  Use this function to determine whether the object is gripped or not. Note that WorkState() must be used before this function to confirm that the current operating state has stopped.

- Parameter:

  id is the model label；

  ErrorCode return：

  0: Obtain status successful；

  >1000: Abnormal, reference 4.2 error code

- Return:

  true: Object gripped；

  false: No object gripped

- Example: ©

```
int ErrorCode;
if( HoldState (id, ErrorCode) ); // object gripped
else; //no object gripped
```

4.1.8 int AlarmState(HEG id)

- Description:
- Use this function to determine whether the gripper is alarmed or not. Note that WorkState() must be used before this function to confirm whether the current operating state has stopped. It is recommended to run the RunMove(...), RunGrip(...), RunExpert(...), and ResetMotion(...) commands. After that, check if the gripper has an abnormal state. If it is abnormal, stop the user program immediately.
- Parameter:
  id is the model label
- Return:
  0: Obtain status successful;
  >1000: Abnormal or alarmed, reference 4.2 error code
- Example: ©

```
int ErrorCode;
ErrorCode = AlarmState(id);
if( ErrorCode > 1000); // abnormal or alarmed occur
else; //no abnormal or alarmed
```

4.1.9 double CurrentPos(HEG id, int &ErrorCode)

- Description:
  When status is not in BUSY, the current position will be returned.
- Parameter:
  id is the model label
  ErrorCode return：
  0: Gripper position obtained successful
  >1000: Abnormal, reference 4.2 error code
- Return:
  Data type is double, the current gripper position (unit: mm).
- Example: ©

```
int ErrorCode;
double Position;
Position = CurrentPos (id, ErrorCode)
```

4.1.10 int StopMotion(HEG id)

- Description:
  Emergency top the current motion of the gripper.
- Parameter:
  id is the model label.
- Return:
  0: Command operation successful.
  >1000: Abnormal, reference 4.2 error code.
- Example: ©
  int ErrorCode;
  ErrorCode = StopMotion (id);

4.1.11 int ResetMotion(HEG id)

- Description:
  After the power is re-powered, the gripper is required to use RESET to find the origin position, the Run-Home return mode must be executed first when RunMove(...), RunGrip(...), and RunExpert(...) are executed.
- Parameter:
  id is the model label
- Return:
  0: Command operation successful；
  >1000: Abnormal, reference 4.2 error code.
- Remark:
  WorkState(…) can be used to confirm whether action is completed, if it is required.
- Example: ©
  int ErrorCode;
  ErrorCode = ResetMotion(id);

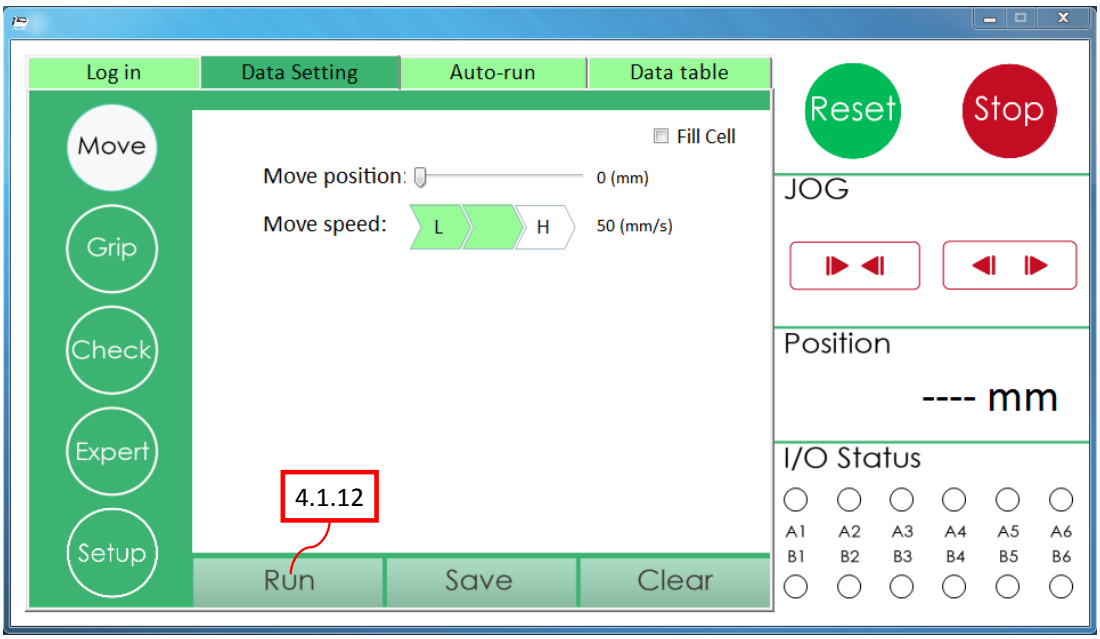


Fig. 5 Electric Gripper Operation Software Interface (Move Mode)

4.1.12 int RunMove(HEG id, double MovPosition, int MovSpeed)

- Description:
  Operate move mode.
- Parameter:
  id is the model label.
  MovPosition is the absolute position of the finger movement (unit: mm);
  MovSpeed is the speed at which the finger moves (unit: mm).
- Return:
  0: Command operation successful;
  >1000: Abnormal, reference 4.2 error code.
- Remark:
  (1) WorkState(…) can be used to confirm whether action is completed, if it is required.
  (2) Movement speed reference comparison table:

Electric Gripper Type	Move Speed (mm/s)	
	Min	Max
XEG-16	1	60
XEG-32	1	80
XEG-64	1	100

- Example: ©

  int ErrorCode;

  ErrorCode = RunMove (id, 16, 60);

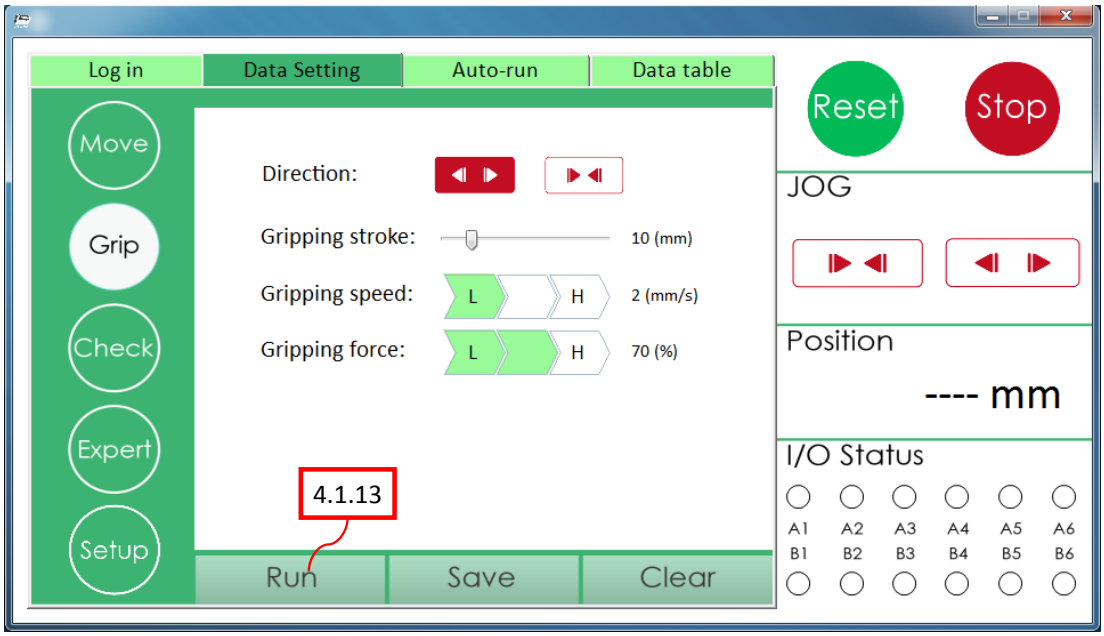


Fig. 6 Electric Gripper Operation Software Interface (Grip Mode)

4.1.13 int RunGrip(HEG id, char Dir, int Str, char GriSpeed, char GriForce)

- Description:

  Operate grip mode with directionality,

- Parameter:

  id is the model label;

  Dir, inward and outward support, set inward to the letter C or c, outward set to the letter O or o;

  Str, relative stroke (unit: mm)

  GriSpeed, gripper feed rate (ex: L: 2, M: 5, H: 10 mm / s), L input value is low, M input value is medium, H input value is high;

  GriForce, the force of gripper (ex: L: 40%, M: 70%, H: 100%), the L input value is low, the M input value is medium, and the H input value is high. Detailed reference to the following notes.

- Return:

  0: Command operation successful；

  >1000: Abnormal, reference 4.2 error code

- Remark:

(1) WorkState(…) can be used to confirm whether action is completed, if it is required.

(2) Comparison table of grip speed and force:

Electric Gripper Type	Grip Speed (mm/s)			Grip Force(%)		
	L	M	H	L	M	H
XEG-16	2	5	10	50	75	100
XEG-32	2	5	10	40	70	100
XEG-64	2	8	15	40	70	100

(3) For example, XEG-16 GripSpeed is L、GripForce is M, the actual speed is 2mm/s and actual force is 37.5N.

- Example: ©

int ErrorCode;

ErrorCode = RunGrip(id, ‘c’, 5, ‘H’, ‘M’);

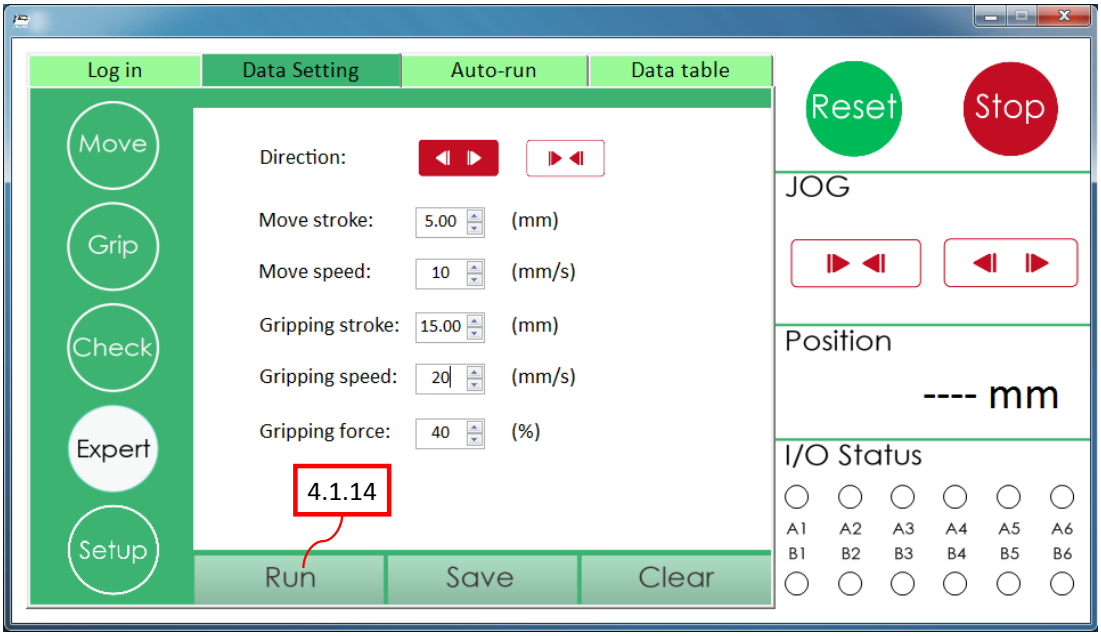


Fig.7 Electric Gripper Operation Software Interface (Expert Mode)

4.1.14 int RunExpert(HEG id, char Dir, double MovStr, int MovSpeed, double GriStr, int GriSpeed, int GriForce)

- Description:
  Operate expert mode with directionality.
- Parameter:
  id is the model label;
  Dir, inward and outward support, set inward to the letter C or c, outward set to the letter O or o;
  MovStr relative position (only the position reaches the orientation), the unit is mm;
  MovSpeed moving speed (unit: mm/s);
  GriStr grip mode displacement (unit: mm);
  GriSpeed grip mode speed (unit: mm / s);
  GriForce force of the grip mode (%) and refers to the 4.2 grip set specification table.
- Return:
  0: Command operation successful；
  >1000: Abnormal, reference 4.2 error code.
- Remark:
  (1) WorkState(…) can be used to confirm whether action is completed, if it is required.
  (2) Speed and force reference table:

Electric Gripper Type	Move Speed (mm/s)		Grip Speed (mm/s)		Grip Force(%)	
	Min	Max	Min	Max	Min	Max
XEG-16	1	60	1	10	50	100
XEG-32	1	80	1	20	40	100
XEG-64	1	100	1	20	40	100

- Example: ©
  int ErrorCode;
  ErrorCode = RunExpert (id, ‘c’, 10, 60, 10, 10, 70);

4.2. Error Code and Specification Setting

The following is the process of using the function, if the return value or ErrorCode return value is not 0, it means that an error has occurred. The following shows an error occurred when the electric gripper goes wrong.

4.2.1 Hardware abnormal

Alarm Code	Description	Reason	Solution
1001	Unable to initialize connection	Connection failed, no data returned.	Refer to the manual to install the gripper driver.
			Check that the 24V power supply is properly connected. Check that the USB cable is properly connected. Check that the serial port is set correctly.
		Enter firmware update mode by mistake.	Confirm that START is OFF, re-power 24V and USB power, you can enter the general mode.
1002		Exceed the connection name limit.	Modify and set connection port be to less than or equal to COM99.
1003		Connection discontinued.	Unplug and reconnect the USB cable.
1004		Gripper serial port unopened.	Close the serial port and reconnect.

4.2.2 Electric Gripper Setting abnormal

Alarm Code	Description	Reason	Solution
2001	Connection abnormal	Gripper type setting error	Check that the gripper type setting is correct
2002	Motion command abnormal	Id is not a DLL return	Check if user program id is returned by the function StartConnect()
2003		Repeat the command in succession	Wait for gripper Busy to end, a new command is issued

2011		Gripper stroke setting is greater than the total stroke.	Check that gripper stroke input is correct.
2012		Position setting is less than zero.	Check that the gripper movement position input is correct.
2013		Movement speed setting is greater than the preset range.	Check that the gripper movement speed input is correct.
2014		Movement speed setting is less than the preset range.	
2021		Gripper movement direction setting.	Check that the gripper movement direction input is correct.
2022		The gripping displacement setting is greater than the range of motion.	Check that the gripping displacement input is correct.
2023		The less displacement setting is greater than the range of motion,.	
2024		Gripping speed is greater than the preset range.	Check that gripping speed input is correct
2025		Gripping speed is less than the preset range.	
2026		Gripping force is great than the preset range.	Check that the gripping force input is correct.
2027		Gripping force is less than the preset range.	

4.2.3 Abnormal Motion

Alarm Code	Description	Reason	Solution
3001	Position error	There are obstacles in the moving stroke.	Eliminate obstacles during the trip.
3003	Reset abnormal	Workpiece have not been removed during the trip.	Check that there are no foreign objects in the itinerary.

		The finger design interferes with the stroke	Modify the finger design.

4.2.4 Electric Gripper Specification Table

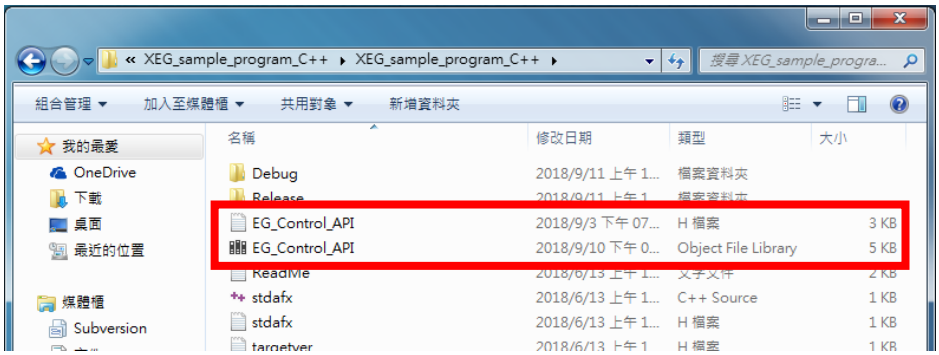
Specification	XEG-16	XEG-32	XEG-64
Stroke	16 mm	32 mm	64 mm
Grip Force	25-50 N	60-150 N	180-450 N
Move Speed	1-60 mm/s	1-80 mm/s	1-100 mm/s
Grip Speed	1-10 mm/s	1-20 mm/s	1-20 mm/s

Appendix

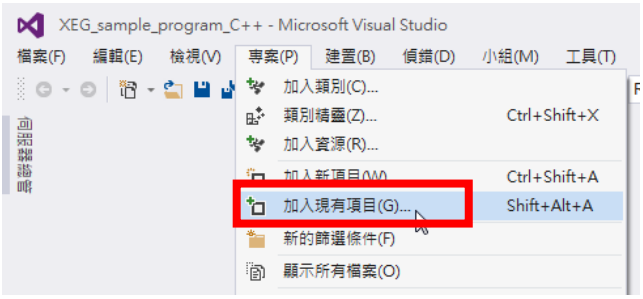
The purpose of this catalog is to provide users of the Visual Studio environment to quickly import this dynamic command library into your project, eliminating the hassle of environment settings, including Visual C++, C# and Basic projects:

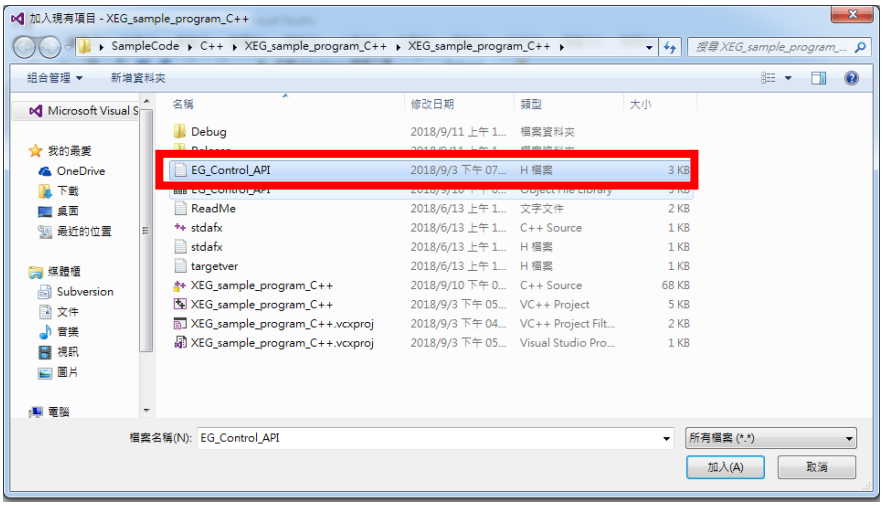
A1. Visual C++ Environment Setting Process

Step 1. Place the file EG_Control_API.h and EG_Control_API.lib in the project.

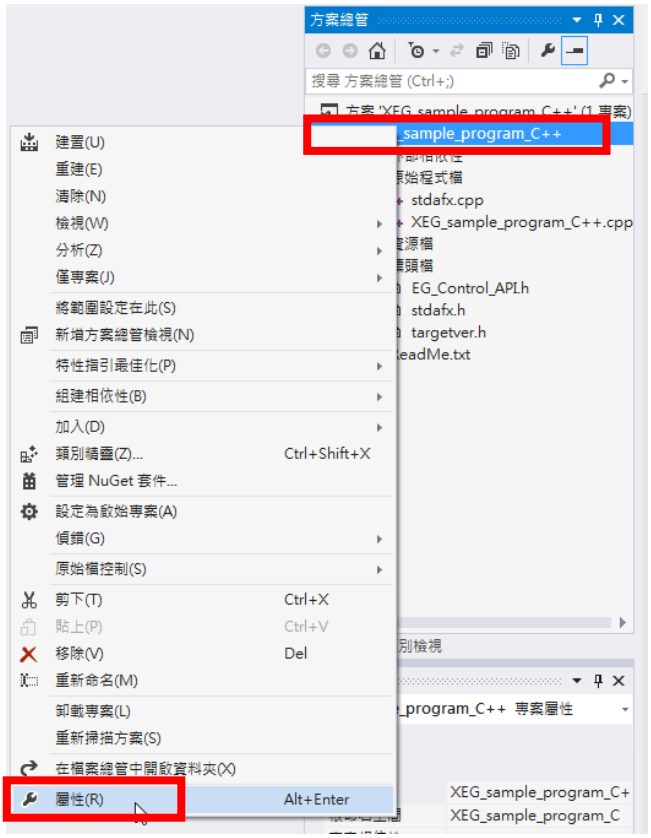


Step 2. Click on Project→Add Existing Project→Select File(*.h)→Add

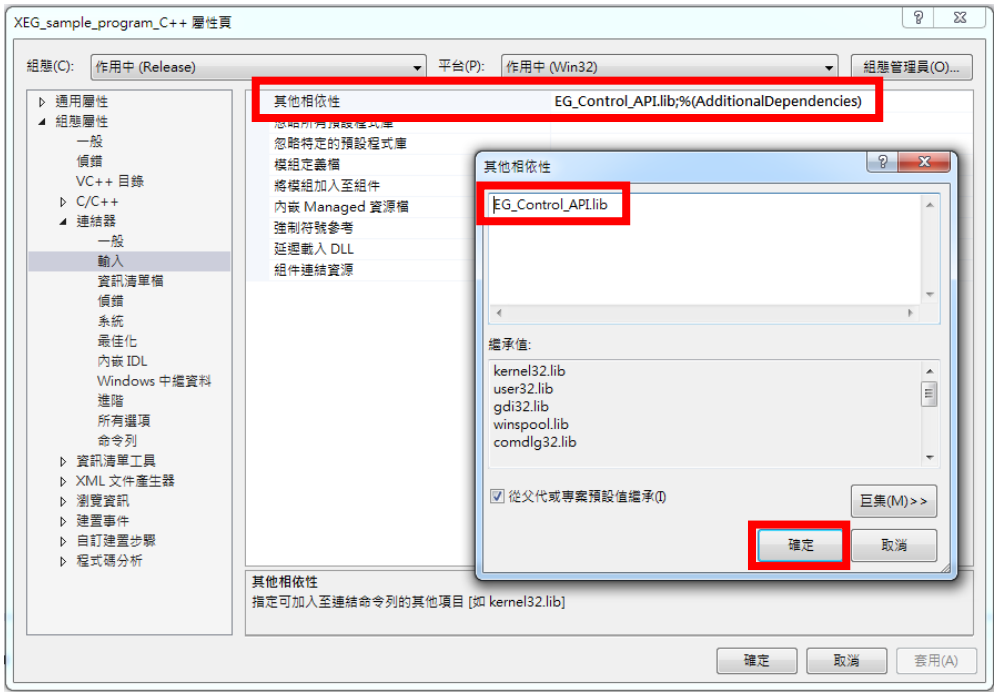




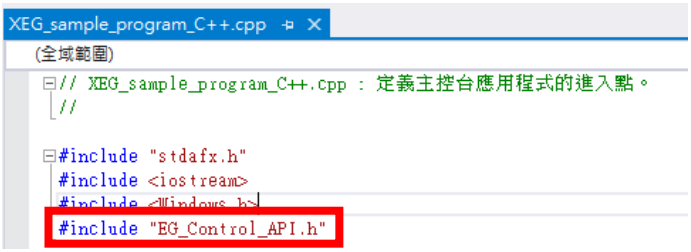
Step 3. Right click on the property and select “Properties”.



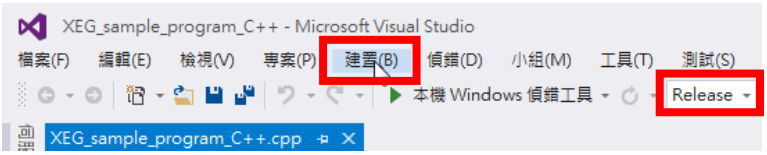
Step 4. Select Linker→Input→Additional Dependencies→Enter EG_Control_API.lib→OK



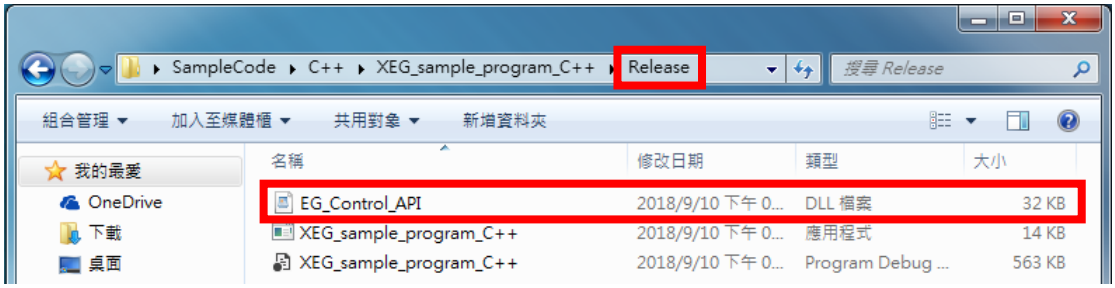
Step 5. Include「EG_Control_API.h」→Start using DLL



Step 6. Create a project, pay attention to the configuration and the corresponding file of the configuration.

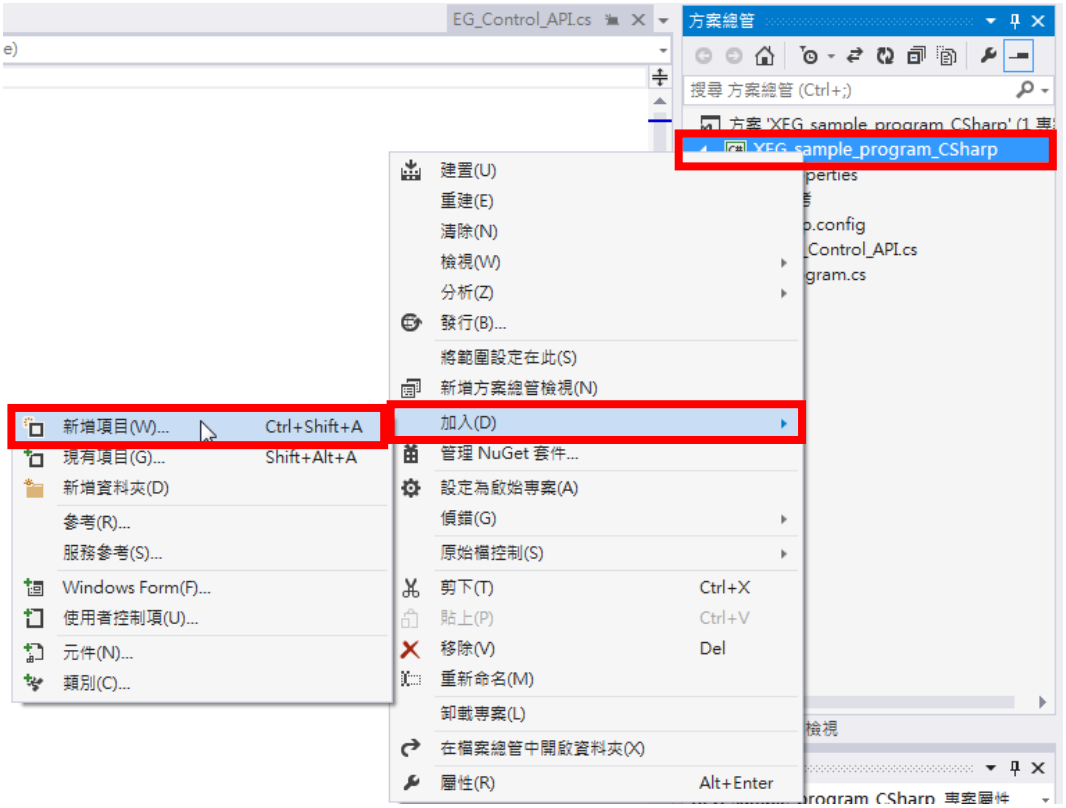


Step 7. Put EG_Control_API.dll into 專案/Release(Debug) path folder→ Direct execute the application

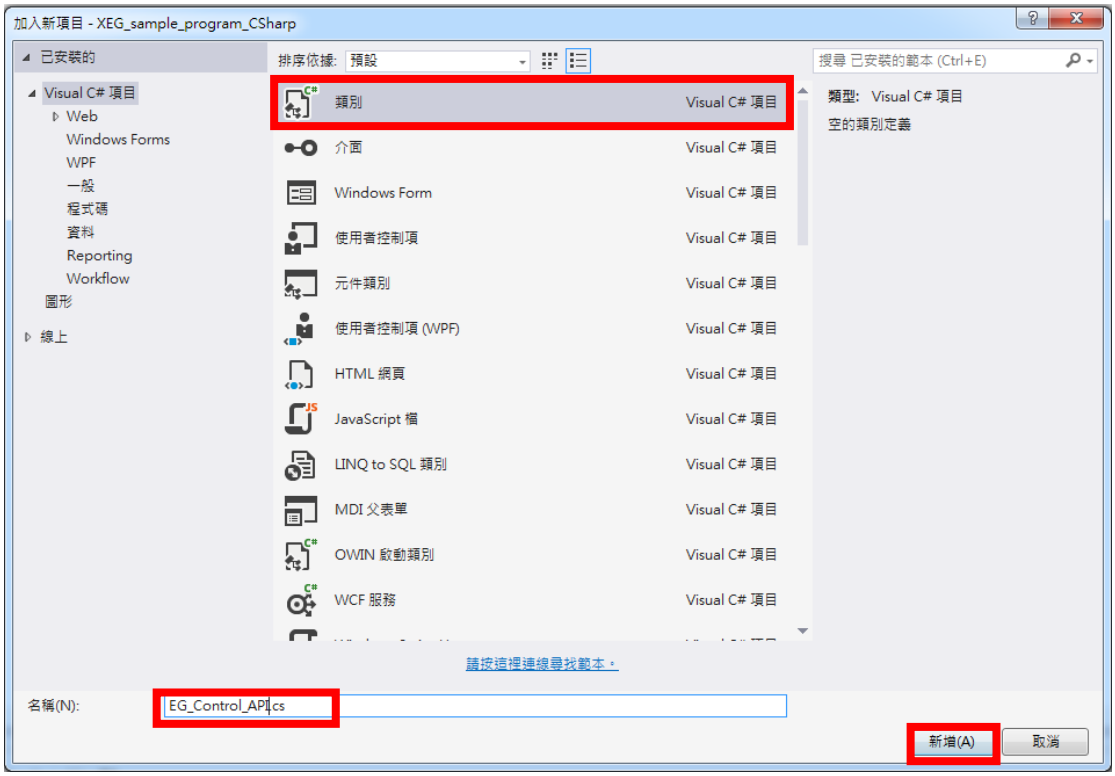


A2. Visual C# Environment Setting Process

Step 1. Right click on project→ Add → New Item



Step 2. Select Category→ Modify Class Name→ Add



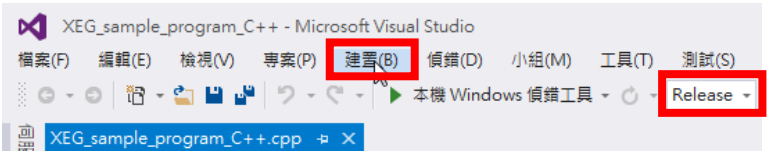
Step 3. Enter the required function.

EG_Control_API.cs | XEG_sample_program_CSharp

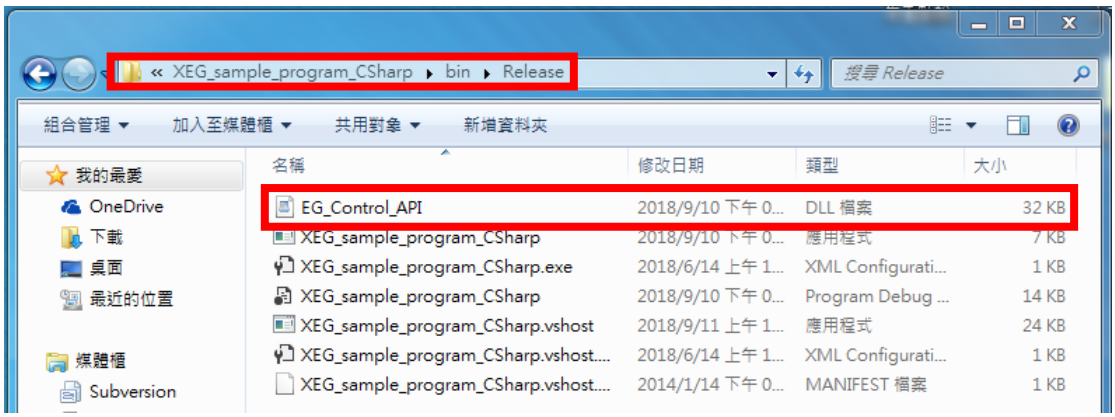
XEG_sample_program_CSharp.EG_Control_API

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using System.Runtime.InteropServices;

namespace XEG_sample_program_CSharp
{
    class EG_Control_API
    {
        #region EG_Control_API
        [DllImport("EG_Control_API.dll")]
        public static extern int StartConnect(int SettingComPort, int SelectModelType);

        [DllImport("EG_Control_API.dll")]
        public static extern int DetectConnect(int id);

        [DllImport("EG_Control_API.dll")]
        public static extern int CloseConnect(int id);

        [DllImport("EG_Control_API.dll")]
        public static extern int CurFirmwareVersion(int id, ref int Ver1, ref int Ver2, ref int Ver3);

        [DllImport("EG_Control_API.dll")]
        public static extern double CurrentPos(int id, ref int ErrorCode);

        [DllImport("EG_Control_API.dll")]
        public static extern int IOStatus(int id, ref uint InputData, ref uint OutputData);
```

Step 4. Create a project, pay attention to the configuration and the corresponding file of the configuration.



Step 5. Put EG_Control_API.dll into Project/Release(Debug) path folder

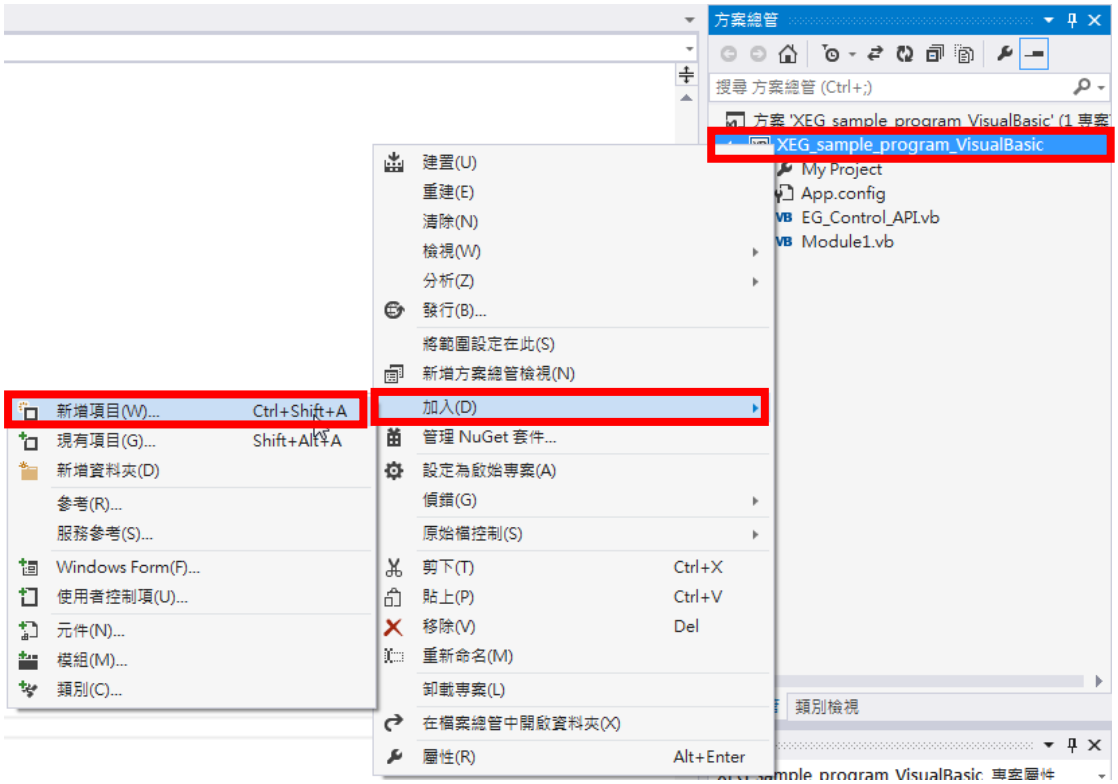


Step 6. Start using DLL

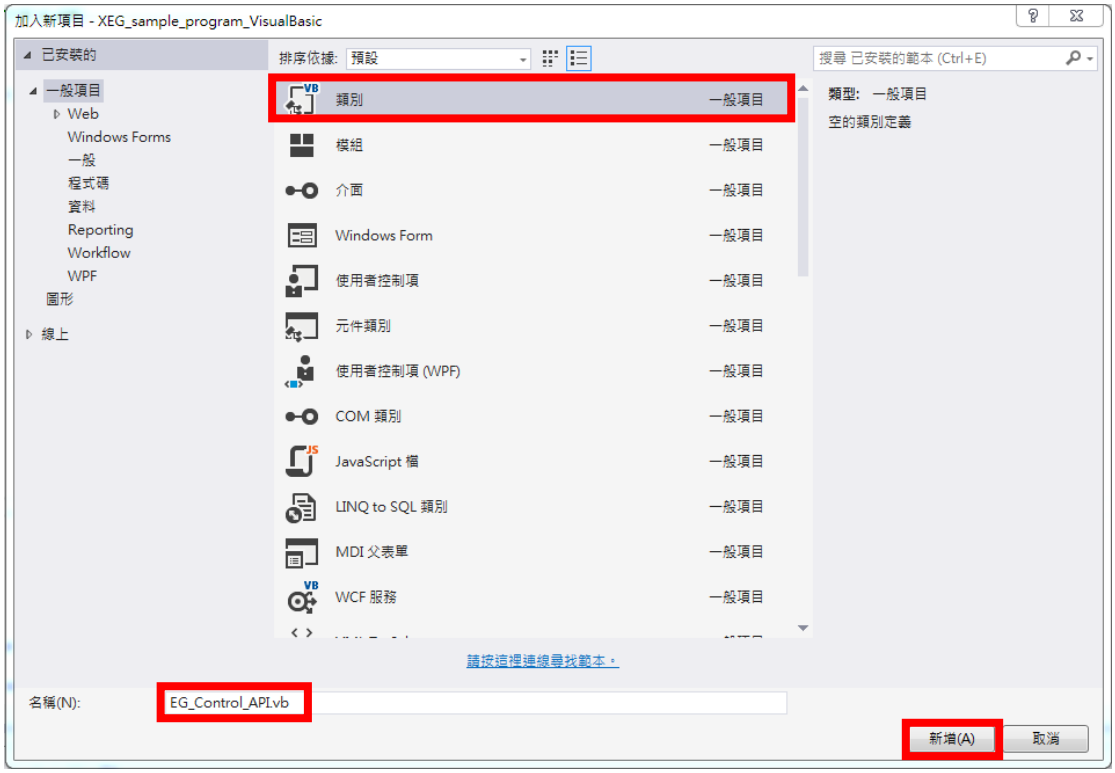
```
namespace XEG_sample_program_CSharp
{
    class Program
    {
        static void Main(string[] args)
        {
            int ErrorCode;
            double Position;
            int Alarm;
            bool Hold;
            int EGErrorINT = 1000; //All Error Code exceed 1000

            //---------------OPEN SERIAL PORT---------------
            int id = EG_Control_API.StartConnect(25, 64);
            int id2 = EG_Control_API.StartConnect(19, 32);

            if (id >= EGErrorINT ||
                id2 >= EGErrorINT)
            {
                Console.WriteLine("Open fail...");
                    Console.ReadKey();
                    return;
            }
            else
            {
                Console.WriteLine("Open success...");
            }
```

A3. Visual Basic Environment Setting Process

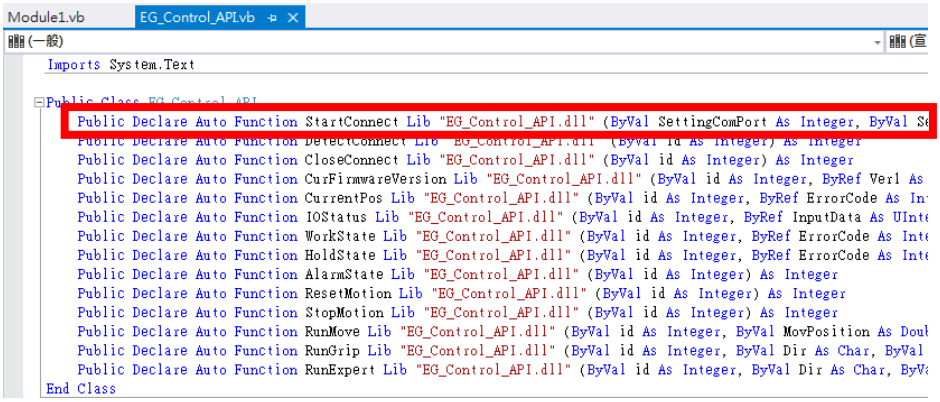
Step 1. Right click on project→ Add → New Item



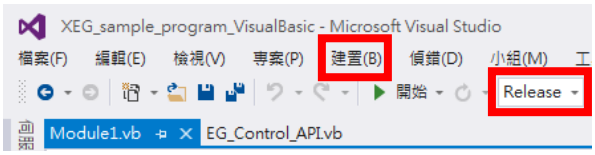
Step 2. Select Category→ Modify Class Name→ Add



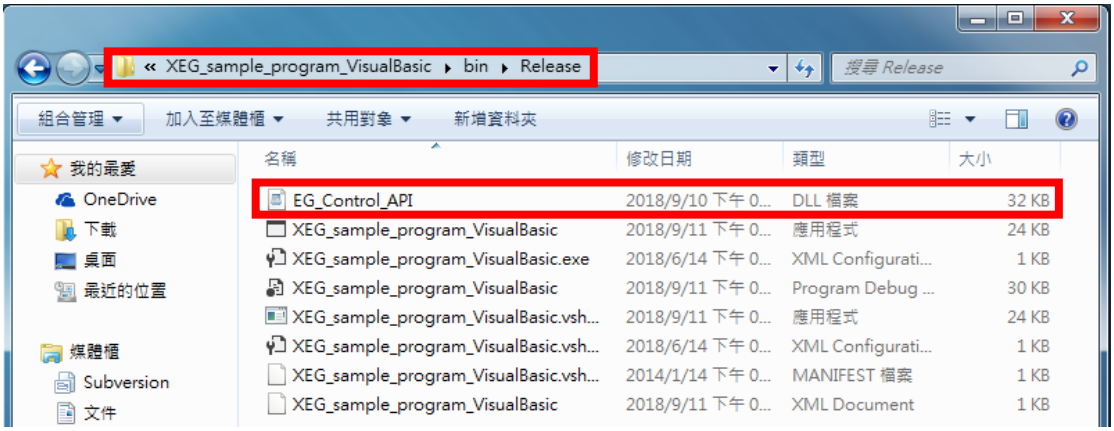
Step 3. Enter the required function



Step 4. Create a project, pay attention to the configuration and the corresponding file of the configuration.



Step 5. Put EG_Control_API.dll into 專案/Release(Debug) path folder



Step 6. Start using DLL

```
Module Module1

    Sub Main()
        Dim ErrorCode As Integer
        Dim Position As Double
        Dim Alarm As Integer
        Dim Hold As Boolean
        Dim EGErrorINT As Integer = 1000
        'All Error Code exceed 1000

        '---------------OPEN SERIAL PORT---------------
        Dim id As Integer = EG_Control_API.StartConnect(25, 64)
        Dim id2 As Integer = EG_Control_API.StartConnect(19, 32)

        If id >= EGErrorINT Or id2 >= EGErrorINT Then
            Console.WriteLine("Open fail...")
            Console.ReadKey()
            Return
        Else
            Console.WriteLine("Open success...")
        End If
```

HIWIN®

Subsidiaries / Research Center

HIWIN GmbH
OFFENBURG, GERMANY
www.hiwin.de
www.hiwin.eu
info@hiwin.de

HIWIN Schweiz GmbH
JONA, SWITZERLAND
www.hiwin.ch
info@hiwin.ch

HIWIN KOREA
SUWON · MASAN, KOREA
www.hiwin.kr
info@hiwin.kr

HIWIN JAPAN
KOBE · TOKYO · NAGOYA · NAGANO · TOHOKU · SHIZUOKA · HOKURIKU · HIROSHIMA · FUKUOKA · KUMAMOTO, JAPAN
www.hiwin.co.jp
info@hiwin.co.jp

HIWIN s.r.o.
BRNO, CZECH REPUBLIC
www.hiwin.cz
info@hiwin.cz

HIWIN CHINA
SUZHOU, CHINA
www.hiwin.cn
info@hiwin.cn

HIWIN USA
CHICAGO, U.S.A.
www.hiwin.com
info@hiwin.com

HIWIN SINGAPORE
SINGAPORE
www.hiwin.sg
info@hiwin.sg

Mega-Fabs Motion System, Ltd.
HAIFA, ISRAEL
www.mega-fabs.com
info@mega-fabs.com

HIWIN Srl
BRUGHERIO, ITALY
www.hiwin.it
info@hiwin.it

HIWIN TECHNOLOGIES CORP.
No. 7, Jingke Road,
Taichung Precision Machinery Park,
Taichung 40852, Taiwan
Tel: +886-4-23594510
Fax: +886-4-23594420
www.hiwin.tw
business@hiwin.tw