Embedded based teaching pendant optimized for industrial robots.

DTP7H-W
Windows CE7
API User's Manual

(R1) Version

DAINCUBE Corp.
ARM Cortex-A9 Windows CE7 system

FORM 170703F – 2019.04.03

DTP7H-W Windows CE7
API User's Manual
DAINCUBE Corp.
Web: www.daincube.com
E-mail: support@daincube.com
Tel: 82-32-329-9783~4
Fax: 82-32-329-9785

#401-701, Bucheon TechnoPark 4-Danji,
655 Pyeongcheon-ro, Wonmi-gu, Bucheon-Si,
Gyeonggi-Do, Republic of Korea

Preface

Copyright notice

Important information

This documentation is intended for qualified audience only.
The product described herein is not an end user product.
It was developed and manufactured for further processing by trained personnel.
This product is protected by the "End-User License Agreement" (EULA).

Disclaimer

Although this document has been generated with the utmost care no warranty or liability for correctness or suitability for any particular purpose is implied. The information in this document is provided “as is” and is subject to change without notice.

Trademarks

All used product names, logos or trademarks are property of their respective owners.

Product support

DAINCUBE Corp.
Web: www.daincube.com
E - MAIL: support@daincube.com

Safety precautions

Be sure to observe all of the following safety precautions.

Strict observance of these warning and caution indications are a MUST for preventing accidents, which could result in bodily injury and substantial property damage. Make sure you fully understand all definitions of these terms and related symbols given below, before you proceed to the manual.

Symbols

The following symbols may be used in this specification:

⚠ Warning

Warnings indicate conditions that, if not observed, can cause personal injury.

⚠ Caution

Cautions warn the user about how to prevent damage to hardware or loss of data.

✎ Note

Notes call attention to important information that should be observed.

Revision history

Revision	Data	Comment
Version 1.0	2018.03.26	Initial version
Version 1.2.1	2019.04.04	Form modification, errata correction Version

Contents

1. Introduction

This document explains the DTP7H-W key, LED and Buzzer to make it easier for users to develop applications. Key, LED and Buzzer are controlled via serial communication, and DAINCUBE provides all device drivers and examples for application developers.

2. Serial communication function

2.1. COM port Open(), Close()

Open and close the serial communication port to enable the DTP7H-W's key, LED, and buzzer operation.

Serial Port : COM1
Baudrate : 115200
Parity : None
Data bit : 8 bit
Stop bit : 1 bit

```
m_comm= new CMycomm(_T("COM1"),_T("115200"),_T("None"),_T("8 Bit"),_T("1 Bit"));
if( m_comm->Create(GetSafeHwnd()) != 0 ) {
   comport_state=true;
} else {
   AfxMessageBox(_T("COM PORT OPEN ERROR!"));
}
```

(1) Parameters

Port
Used serial port name
Baudrate
Serial Port communication Baudrate.
Parity
Serial Port Parity.
Databit
Databit.
Stopbit
Stopbit.

(2) Return value

Return error value = 0

(3) Explanation

DTP7H-W Because ETC driver is controlled through serial port, COM port must be opened

(4) Requirements

Function	Header	Reference

CMycomm()
Create()
	Mycomm.h	Mycomm.cpp

(5) Exemple

```
void CserialDlg::OnBnClickedBtConnect()
{
      if(comport_state) {      // Close COM port
      if(m_comm) {
         m_comm->Close();
         m_comm = NULL;
         comport_state=false;
   } else {      // Initial COM port
      m_comm= new CMycomm(_T("COM1"),_T("115200"),_T("None"),_T("8 Bit"),_T("1 Bit"));
      if( m_comm->Create(GetSafeHwnd()) != 0 ) {
            comport_state=true;
        } else {
            AfxMessageBox(_T("COM PORT OPEN ERROR!"));
        }
    }
}
```

2.2. Send()

DTP7H-W LED and Buzzer are transmitted through Serial Packet.

```
BOOL CMycomm::Send(char *outbuf, DWORD *len);
```

(1) Parameters

outbuf

Transmit data buffer.

len

Transmit data length.

(2) Return value

Return error value = 0

Return success value = 0

(3) Explanation

In order to turn on / off the LED or Buzzer, control by sending serial packet.

(4) Requirements

Function	Header	Reference
Send()	Mycomm.h	Mycomm.cpp

(5) Exemple

```
void CserialDlg::OnBnClickedBtLed1()
{
    // TODO: Add your control notification handler code here
    char buf_printf[10] = {0, };
    unsigned int crc_buf;
    DWORD dwBytes = 0;

    buf_printf[0] = STX;               // STX
    buf_printf[1] = MOD_SET;           // MOD (get : 0x10, set : 0x11)
    buf_printf[2] = SEL_LED;           // SEL (LED : 0x3A)
    buf_printf[3] = LEFT_LED1;           // Data1
    buf_printf[4] = LED_BLUE; // Data2 (off : 0x30, blue : 0x31, red : 0x32, all : 0x33)
    buf_printf[5] = DATA_RESERVED;         // Data3 (Reserved : 0x20)
    crc_buf = crc16_append(buf_printf,6);
    buf_printf[6] = (char)(crc_buf>>8)&0xff;
    buf_printf[7] = (char)crc_buf&0xff;
    buf_printf[8] = ETX;               // ETX
    buf_printf[9] = '\0';

    dwBytes = strlen(buf_printf);
    m_comm->Send(buf_printf, &dwBytes);
}
```

2.3. Receive()

Receive Serial Packet to use DTP7H-W key.

```
int CMycomm::Receive(LPSTR inbuf, int len);
```

(1) Parameters

inbuf

Recive data buffer.

len

Recive data length.

(2) Return value

Return error value = 0 or -1
Return success value = length

(3) Explanation

Key receives and controls Serial Packet.

(4) Requirements

Function	Header	Reference
Receive()	Mycomm.h	Mycomm.cpp

(5) Exemple

```
LRESULT CserialDlg::OnReceive(WPARAM length, LPARAM lpara)
{
   if(m_comm && comport_state) {

      while(length--)
      {
         m_comm->Receive(&g_Receive_Buffer[g_Head_Pointer],1);

         if(g_Head_Pointer >= BUFF_MAX-1)
            g_Head_Pointer = 0;
         else
            g_Head_Pointer++;
      }
   }

   return 0;
}
```

2.4. Protocol CRC

The serial packet of the ETC control also contains the CRC information, so the CRC value must be calculated.
CRC calculation is performed with information from the first packet to the sixth packet among the 9-bit serial packets.

```
      unsigned int crc16_append(const void* data, int len);
```

(1) Parameters

data

The buffer of the Serial Packet for which to calculate the CRC value.

len

The length of the buffer in the Serial Packet for which to calculate the CRC value.

(2) Return value

16-bit CRC value.

(3) Explanation

Calculates the Send / Receive CRC value of the serial packet.

(4) Requirements

Function	Header	Reference
crc16_append()		

```
// CRC Table
static const unsigned int crc16tab[] = {
    0x0000, 0xC0C1, 0xC181, 0x0140, 0xC301, 0x03C0, 0x0280, 0xC241,
    0xC601, 0x06C0, 0x0780, 0xC741, 0x0500, 0xC5C1, 0xC481, 0x0440,
    0xCC01, 0x0CC0, 0x0D80, 0xCD41, 0x0F00, 0xCFC1, 0xCE81, 0x0E40,
    0x0A00, 0xCAC1, 0xCB81, 0x0B40, 0xC901, 0x09C0, 0x0880, 0xC841,
    0xD801, 0x18C0, 0x1980, 0xD941, 0x1B00, 0xDBC1, 0xDA81, 0x1A40,
    0x1E00, 0xDEC1, 0xDF81, 0x1F40, 0xDD01, 0x1DC0, 0x1C80, 0xDC41,
    0x1400, 0xD4C1, 0xD581, 0x1540, 0xD701, 0x17C0, 0x1680, 0xD641,
    0xD201, 0x12C0, 0x1380, 0xD341, 0x1100, 0xD1C1, 0xD081, 0x1040,
    0xF001, 0x30C0, 0x3180, 0xF141, 0x3300, 0xF3C1, 0xF281, 0x3240,
    0x3600, 0xF6C1, 0xF781, 0x3740, 0xF501, 0x35C0, 0x3480, 0xF441,
    0x3C00, 0xFCC1, 0xFD81, 0x3D40, 0xFF01, 0x3FC0, 0x3E80, 0xFE41,
    0xFA01, 0x3AC0, 0x3B80, 0xFB41, 0x3900, 0xF9C1, 0xF881, 0x3840,
    0x2800, 0xE8C1, 0xE981, 0x2940, 0xEB01, 0x2BC0, 0x2A80, 0xEA41,
    0xEE01, 0x2EC0, 0x2F80, 0xEF41, 0x2D00, 0xEDC1, 0xEC81, 0x2C40,
    0xE401, 0x24C0, 0x2580, 0xE541, 0x2700, 0xE7C1, 0xE681, 0x2640,
    0x2200, 0xE2C1, 0xE381, 0x2340, 0xE101, 0x21C0, 0x2080, 0xE041,
    0xA001, 0x60C0, 0x6180, 0xA141, 0x6300, 0xA3C1, 0xA281, 0x6240,
    0x6600, 0xA6C1, 0xA781, 0x6740, 0xA501, 0x65C0, 0x6480, 0xA441,
    0x6C00, 0xACC1, 0xAD81, 0x6D40, 0xAF01, 0x6FC0, 0x6E80, 0xAE41,
    0xAA01, 0x6AC0, 0x6B80, 0xAB41, 0x6900, 0xA9C1, 0xA881, 0x6840,
    0x7800, 0xB8C1, 0xB981, 0x7940, 0xBB01, 0x7BC0, 0x7A80, 0xBA41,
    0xBE01, 0x7EC0, 0x7F80, 0xBF41, 0x7D00, 0xBDC1, 0xBC81, 0x7C40,
    0xB401, 0x74C0, 0x7580, 0xB541, 0x7700, 0xB7C1, 0xB681, 0x7640,
    0x7200, 0xB2C1, 0xB381, 0x7340, 0xB101, 0x71C0, 0x7080, 0xB041,
    0x5000, 0x90C1, 0x9181, 0x5140, 0x9301, 0x53C0, 0x5280, 0x9241,
    0x9601, 0x56C0, 0x5780, 0x9741, 0x5500, 0x95C1, 0x9481, 0x5440,
    0x9C01, 0x5CC0, 0x5D80, 0x9D41, 0x5F00, 0x9FC1, 0x9E81, 0x5E40,
    0x5A00, 0x9AC1, 0x9B81, 0x5B40, 0x9901, 0x59C0, 0x5880, 0x9841,
    0x8801, 0x48C0, 0x4980, 0x8941, 0x4B00, 0x8BC1, 0x8A81, 0x4A40,
    0x4E00, 0x8EC1, 0x8F81, 0x4F40, 0x8D01, 0x4DC0, 0x4C80, 0x8C41,
    0x4400, 0x84C1, 0x8581, 0x4540, 0x8701, 0x47C0, 0x4680, 0x8641,
    0x8201, 0x42C0, 0x4380, 0x8341, 0x4100, 0x81C1, 0x8081, 0x4040
};

// calculation CRC16
unsigned int ::crc16_append(const void* data, int len)
{
    int i          = 0;
    unsigned int c16       = 0;
    unsigned char* buf = NULL;

    buf  = (unsigned char*) data;
    for(i = 0; i < len; i++)
        c16 = (c16 >> 8) ^ crc16tab[(c16 ^ buf[i]) & 0xff];

    return c16;
}
```

3. ETC Driver control protocol

The protocols for controlling the DTP7H-W key, LED, and buzzer are 9-bit and 1-packet, and the bit information is as follows.

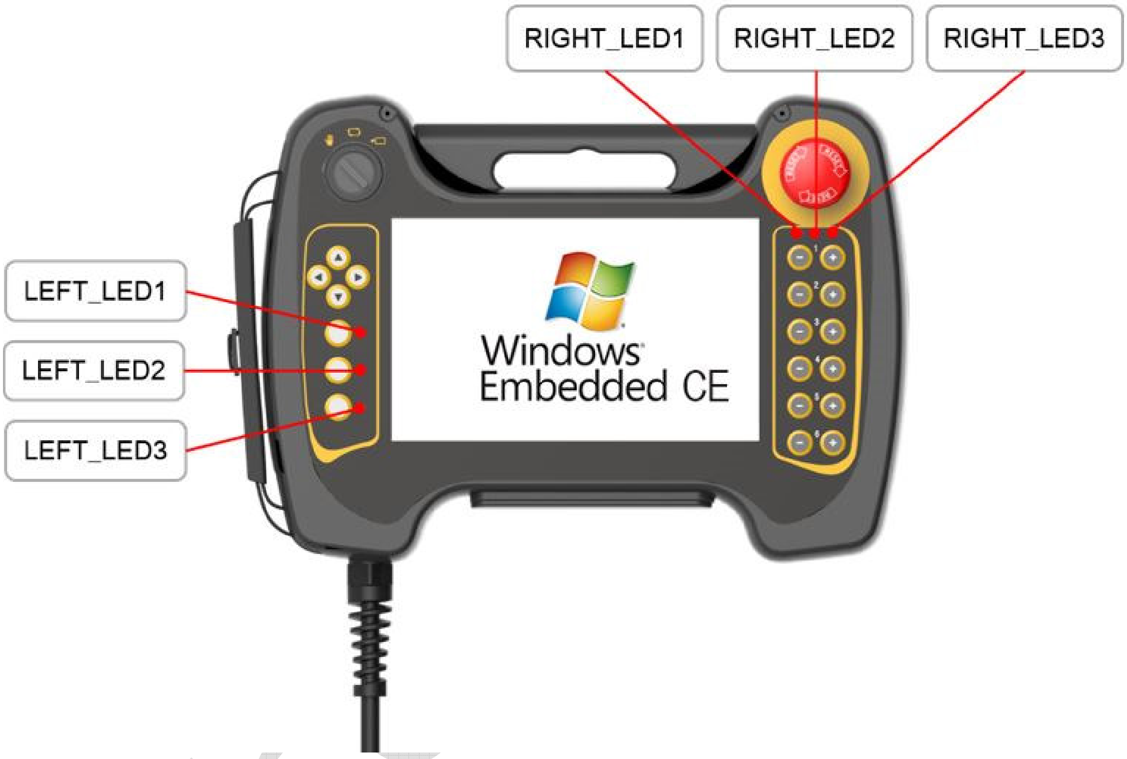
STX	Start Byte
MOD	Get/Set Select
SEL	Control Device Select (Key, LED, Buzzer)
DATA1	Control Device value
DATA2	Control Device value
DATA3	Control Device value
CRC_H	CRC High (STX ~ DATA3)
CRC_L	CRC Low (STX ~ DATA3)
ETX	End Byte

Values of define information for Key, LED and Buzzer control are as follows.

```
//KEYPAD
#define KEY_A    30
#define KEY_B    48
#define KEY_C    46
#define KEY_D    32
#define KEY_E    18
#define KEY_F    33
#define KEY_G    34
#define KEY_H    35
#define KEY_I    23
#define KEY_J    36
#define KEY_K    37
#define KEY_L    38
#define KEY_DOWN  108
#define KEY_UP     103
#define KEY_RIGHT  106
#define KEY_LEFT  105
#define KEY_F1     59
#define KEY_F2     60
#define KEY_F3     61
#define KEY_F6     64
#define KEY_F7     65
#define KEY_F8     66

// ETC
#define LEFT_LED1      0x41     // ASCII Value
#define LEFT_LED2      0x42
#define LEFT_LED3      0x43
#define RIGHT_LED1     0x61
#define RIGHT_LED2     0x62
#define RIGHT_LED3     0x63
#define LED_ALL      0x7F

//SERIAL
#define STX          0x02
#define MOD_GET        0x10
#define MOD_SET        0x11
#define SEL_LED      0x3A
#define SEL_BUZZ     0x3B
#define SEL_KEYPAD       0x3D
#define DATA_RESERVED  0x20
#define ETX          0x03
```

```
#define KEYPAD_UP     0x30
#define KEYPAD_DOWN     0x31
#define LED_OFF         0x30
#define LED_BLUE     0x31
#define LED_RED         0x32
#define LED_ALL_ON    0x33
#define BUZZ_OFF     0x30
#define BUZZ_ON         0x31
```

3.1. LED control protocol



LED Control Serial Packet

STX	MOD	SEL	DATA1	DATA2	DATA3	CRC_H	CRC_L	ETX
0x02	0x11	0x3A	0x41	0x33	0x20	0xXX	0xXX	0x03
1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE

- MOD : 0x11 = MOD_SET
- SEL : 0x3A = SEL_LED
- DATA1 : 0x41 = LEFT_LED1, 0x42 = LEFT_LED2, 0x43 = LEFT_LED3,
  0x61 = RIGHT_LED1, 0x62 = RIGHT_LED2, 0x63 = RIGHT_LED3
- DATA2 : 0x30 = OFF, 0x31 = BLUE, 0x32 = RED, 0x33 = ALL
- DATA3 : 0x20 = DATA_RESERVED

3.2. Buzzer control protocol

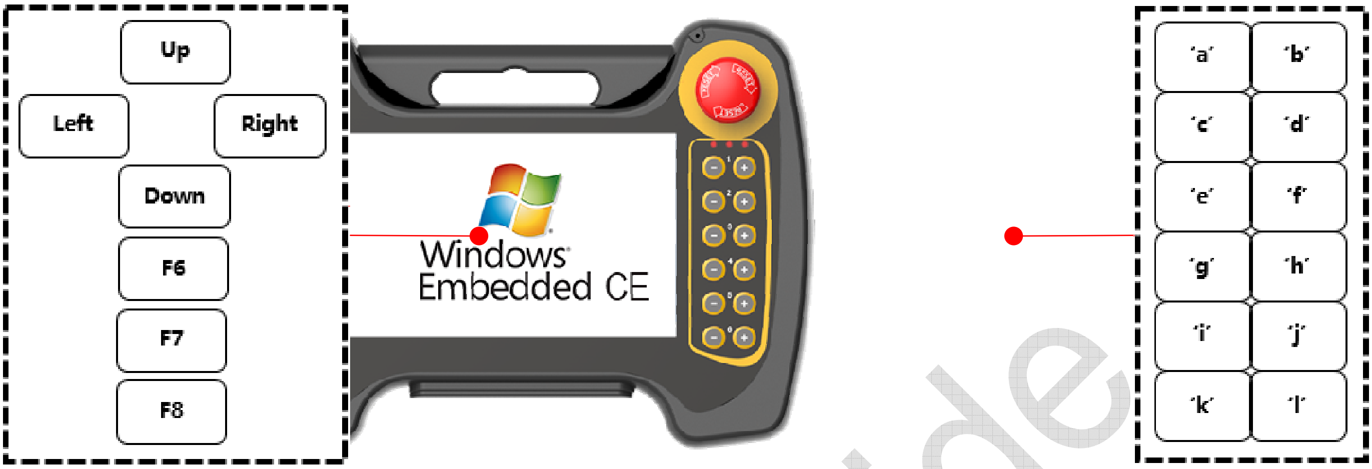
Buzzer Control Serial Packet

STX	MOD	SEL	DATA1	DATA2	DATA3	CRC_H	CRC_L	ETX
0x02	0x11	0x3B	0x31	0x20	0x20	0xXX	0xXX	0x03
1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE

- MOD : 0x11 = MOD_SET
- SEL : 0x3B = SEL_BUZZ

- DATA1 : 0x30 = OFF, 0x31 = ON
- DATA2 : 0x20 = DATA_RESERVED
- DATA3 : 0x20 = DATA_RESERVED

3.3. Keypad control protocol



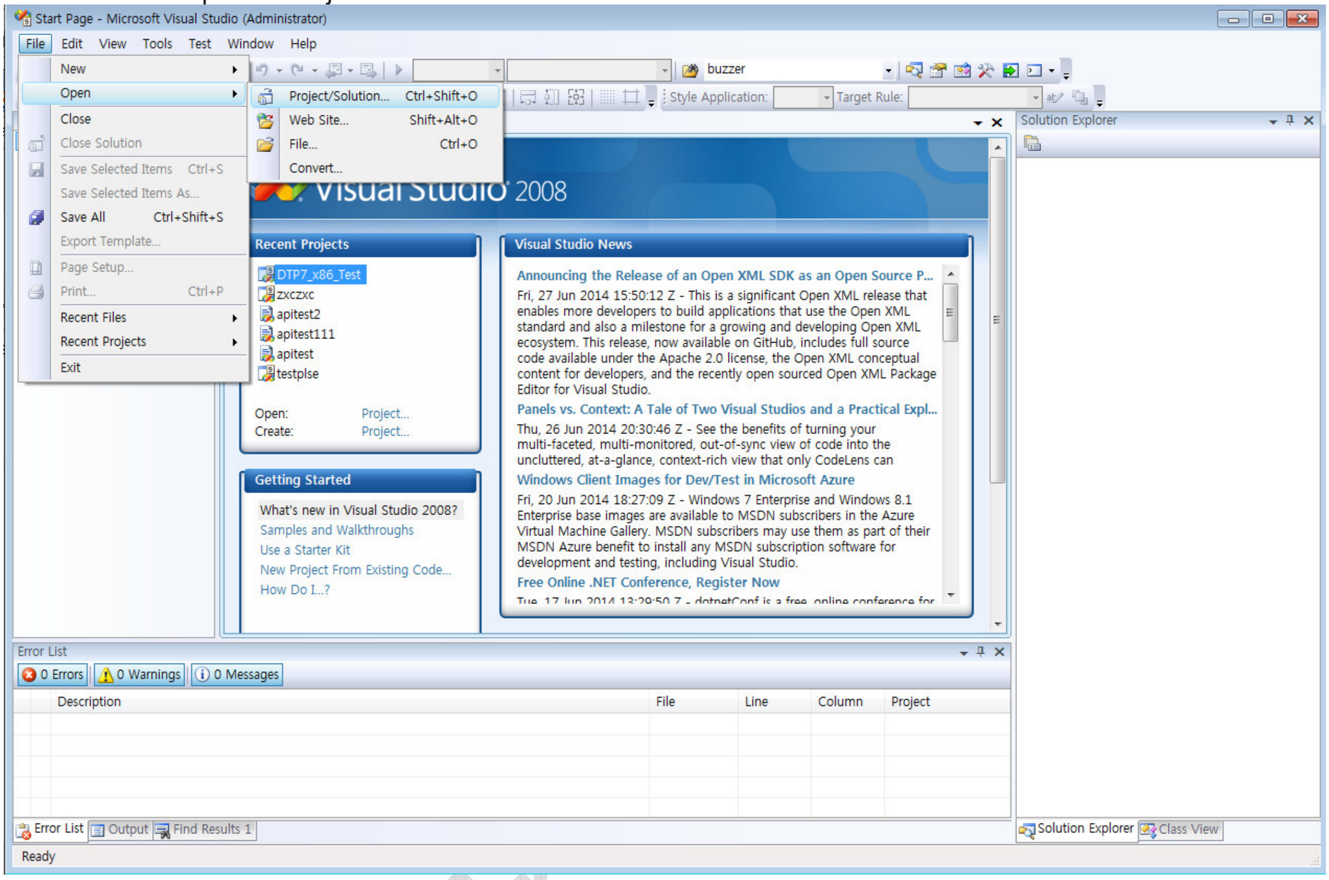
Keypad Control Serial Packet

STX	MOD	SEL	DATA1	DATA2	DATA3	CRC_H	CRC_L	ETX
0x02	0x10	0x3C	0x30	0x33	0x30	0xXX	0xXX	0x03
1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE	1BYTE

- MOD : 0x10 = MOD_GET
- SEL : 0x3D = SEL_KEYPAD
- DATA1 : 0x30 = KEYPAD_UP, 0x31 = KEYPAD_DOWN
- DATA2 : KEY_A = 30, KEY_B = 48, KEY_C = 46, KEY_D = 32, KEY_E = 18, KEY_F = 33, KEY_G = 34, KEY_H = 35, KEY_I = 23, KEY_J = 36, KEY_K = 37, KEY_L = 38, KEY_DOWN = 108, KEY_UP = 103, KEY_RIGHT = 106, KEY_LEFT = 105, KEY_F6 = 64, KEY_F7 = 65, KEY_F8 = 66, KEY_F9 = 67
- DATA3 : 0x20 = DATA_RESERVED

4. ETC Sample Usage

Open using Visual Basic to use the Sample Project provided by DAINCUBE.
Select File -> Open -> Project / Solution.



Announcing the Release of an Open XML SDK as an Open Source P...

Fri, 27 Jun 2014 15:50:12 Z - This is a significant Open XML release that enables more developers to build applications that use the Open XML standard and also a milestone for a growing and developing Open XML ecosystem. This release, now available on GitHub, includes full source code available under the Apache 2.0 license, the Open XML conceptual content for developers, and the recently open sourced Open XML Package Editor for Visual Studio.

Panels vs. Context: A Tale of Two Visual Studios and a Practical Expl...

Thu, 26 Jun 2014 20:30:46 Z - See the benefits of turning your multi-faceted, multi-monitored, out-of-sync view of code into the uncluttered, at-a-glance, context-rich view that only CodeLens can

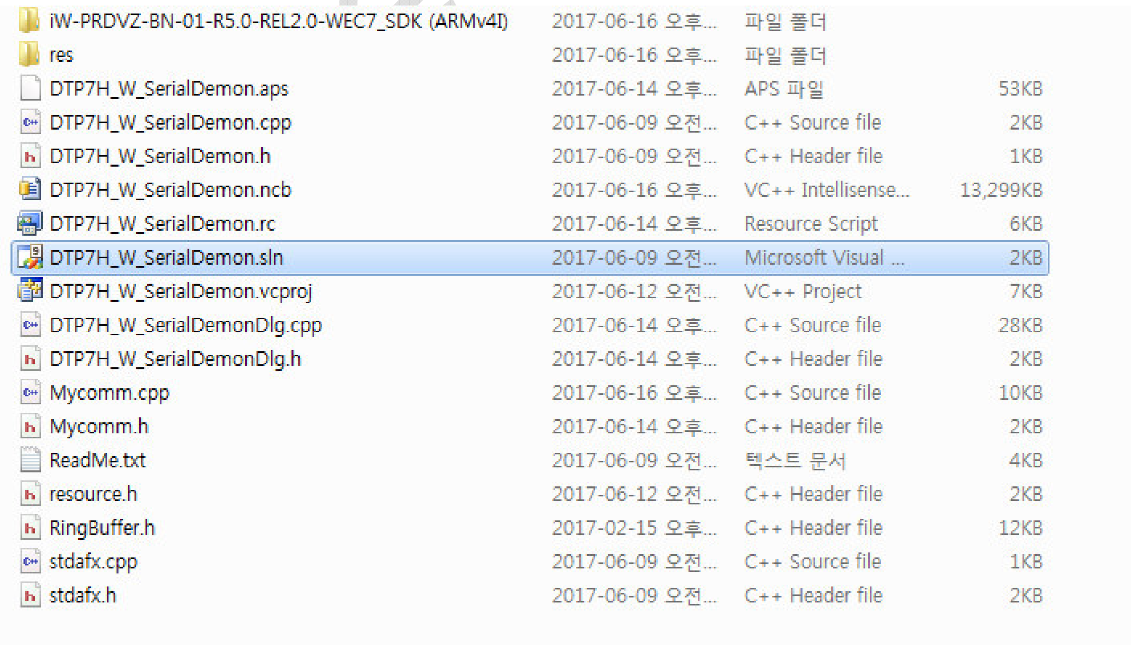
Windows Client Images for Dev/Test in Microsoft Azure

Fri, 20 Jun 2014 18:27:09 Z - Windows 7 Enterprise and Windows 8.1 Enterprise base images are available to MSDN subscribers in the Azure Virtual Machine Gallery. MSDN subscribers may use them as part of their MSDN Azure benefit to install any MSDN subscription software for development and testing, including Visual Studio.

Free Online .NET Conference, Register Now

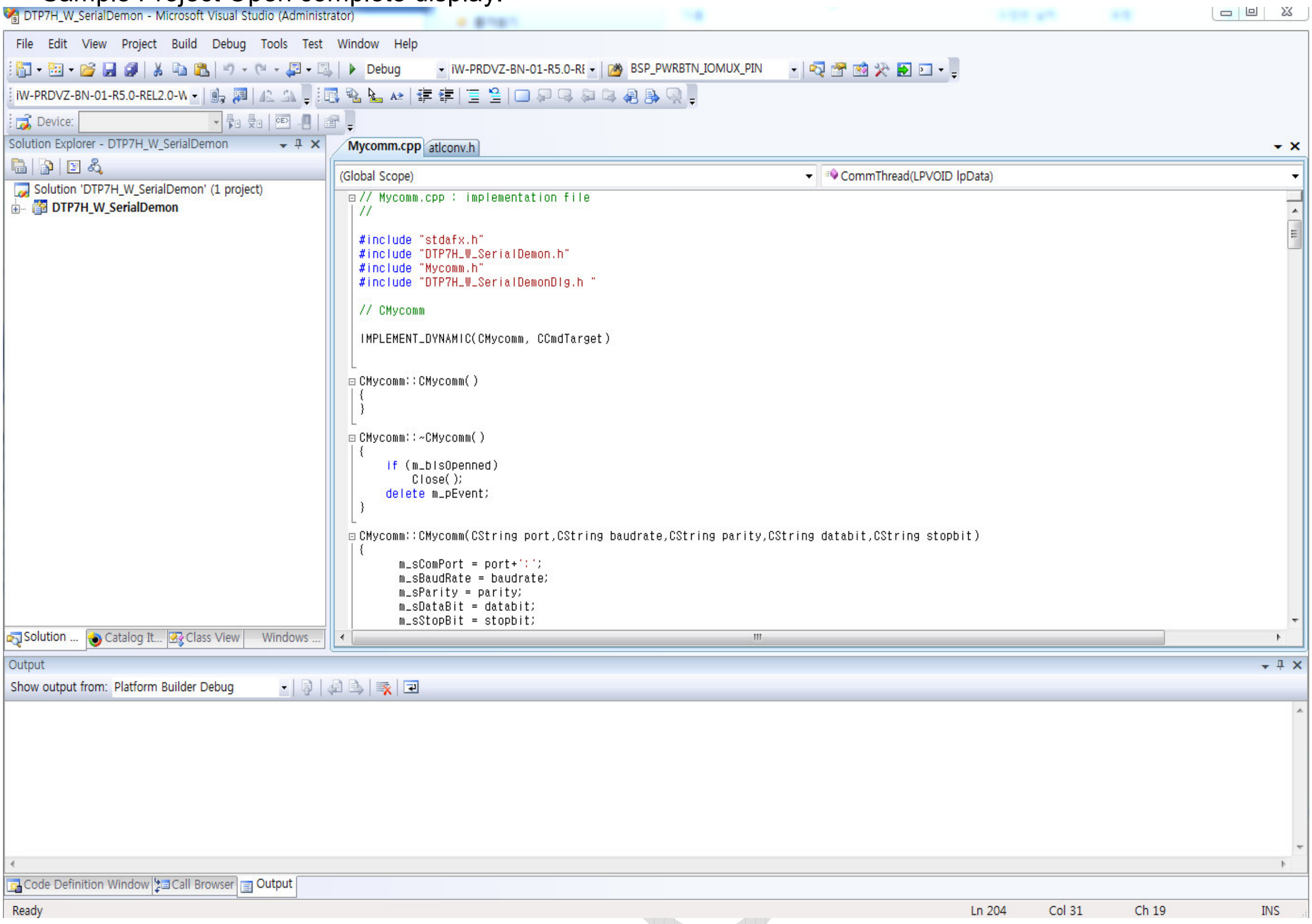
Tue, 17 Jun 2014 13:29:50 Z - dotnetConf is a free, online conference for

Select Project "02_Sample >> DTP7H_W_SerialDemon_ETC >> DTP7H_W_SerialDemon.sln"



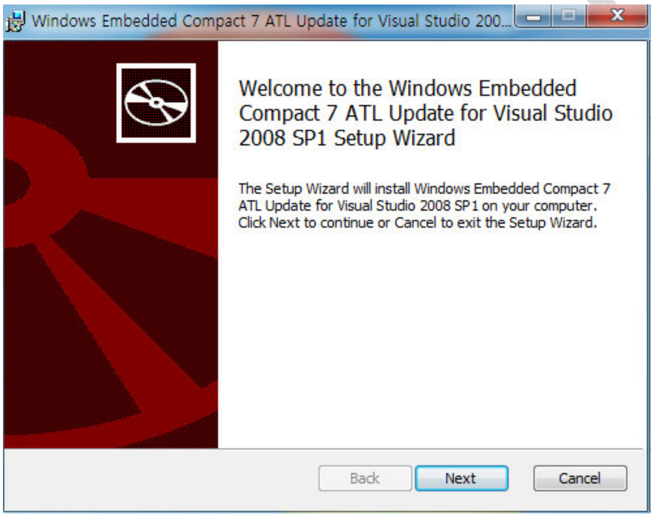
iW-PRDVZ-BN-01-R5.0-REL2.0-WEC7_SDK (ARMv4I)	2017-06-16 오후...	파일 폴더	
res	2017-06-16 오후...	파일 폴더	
DTP7H_W_SerialDemon.aps	2017-06-14 오후...	APS 파일	53KB
DTP7H_W_SerialDemon.cpp	2017-06-09 오전...	C++ Source file	2KB
DTP7H_W_SerialDemon.h	2017-06-09 오전...	C++ Header file	1KB
DTP7H_W_SerialDemon.ncb	2017-06-16 오후...	VC++ Intellisense...	13,299KB
DTP7H_W_SerialDemon.rc	2017-06-14 오후...	Resource Script	6KB
DTP7H_W_SerialDemon.sln	2017-06-09 오전...	Microsoft Visual ...	2KB
DTP7H_W_SerialDemon.vcproj	2017-06-12 오전...	VC++ Project	7KB
DTP7H_W_SerialDemonDlg.cpp	2017-06-14 오후...	C++ Source file	28KB
DTP7H_W_SerialDemonDlg.h	2017-06-14 오후...	C++ Header file	2KB
Mycomm.cpp	2017-06-16 오후...	C++ Source file	10KB
Mycomm.h	2017-06-14 오후...	C++ Header file	2KB
ReadMe.txt	2017-06-09 오전...	텍스트 문서	4KB
resource.h	2017-06-12 오전...	C++ Header file	2KB
RingBuffer.h	2017-02-15 오후...	C++ Header file	12KB
stdafx.cpp	2017-06-09 오전...	C++ Source file	1KB
stdafx.h	2017-06-09 오전...	C++ Header file	2KB

Sample Project Open complete display.


DTP7H_W_SerialDemon - Microsoft Visual Studio (Administrator)
File Edit View Project Build Debug Tools Test Window Help
Debug
iW-PRDVZ-BN-01-R5.0-RI
BSP_PWRBTN_IOMUX_PIN
iW-PRDVZ-BN-01-R5.0-REL2.0-W
Device:
Solution Explorer - DTP7H_W_SerialDemon
Solution 'DTP7H_W_SerialDemon' (1 project)
DTP7H_W_SerialDemon
Solution ...
Catalog It...
Class View
Windows ...
Mycomm.cpp
atlconv.h
(Global Scope)
CommThread(LPVOID lpData)
Output
Show output from: Platform Builder Debug
Code Definition Window
Call Browser
Output
Ready
Ln 204
Col 31
Ch 19
INS

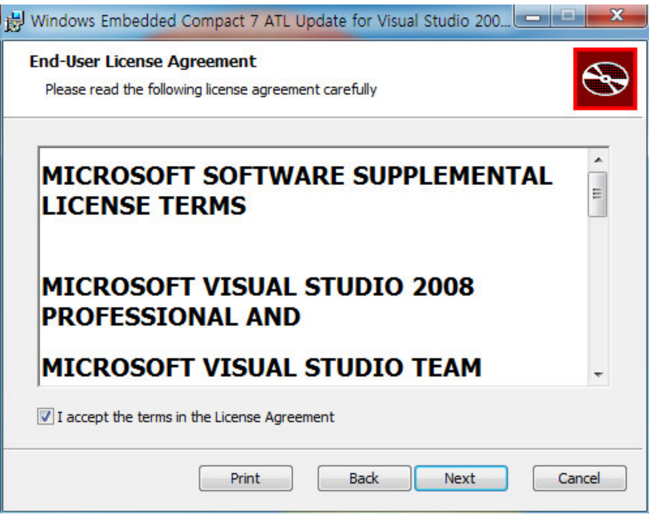
```
// Mycomm.cpp : implementation file
//

#include "stdafx.h"
#include "DTP7H_W_SerialDemon.h"
#include "Mycomm.h"
#include "DTP7H_W_SerialDemonDlg.h "

// CMycomm

IMPLEMENT_DYNAMIC(CMycomm, CCmdTarget)


CMycomm::CMycomm()
{
}

CMycomm::~CMycomm()
{
    if (m_bIsOpenned)
        Close();
    delete m_pEvent;
}

CMycomm::CMycomm(CString port,CString baudrate,CString parity,CString databit,CString stopbit)
{
      m_sComPort = port+':';
      m_sBaudRate = baudrate;
      m_sParity = parity;
      m_sDataBit = databit;
      m_sStopBit = stopbit;
```

If the build does not work normally, install the attachment VisualStudioDeviceWindowsEmbeddedCompact7.msi.


Windows Embedded Compact 7 ATL Update for Visual Studio 200...
Back
Next
Cancel

Welcome to the Windows Embedded Compact 7 ATL Update for Visual Studio 2008 SP1 Setup Wizard

The Setup Wizard will install Windows Embedded Compact 7 ATL Update for Visual Studio 2008 SP1 on your computer. Click Next to continue or Cancel to exit the Setup Wizard.


Windows Embedded Compact 7 ATL Update for Visual Studio 200...
Print
Back
Next
Cancel

End-User License Agreement

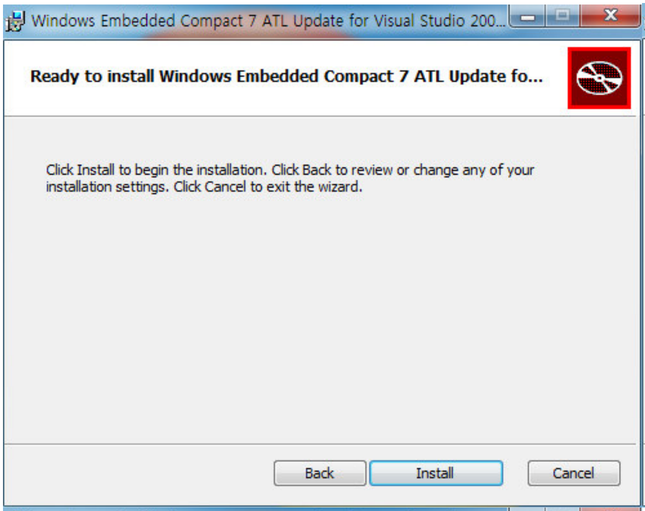
Please read the following license agreement carefully

MICROSOFT SOFTWARE SUPPLEMENTAL LICENSE TERMS

MICROSOFT VISUAL STUDIO 2008 PROFESSIONAL AND

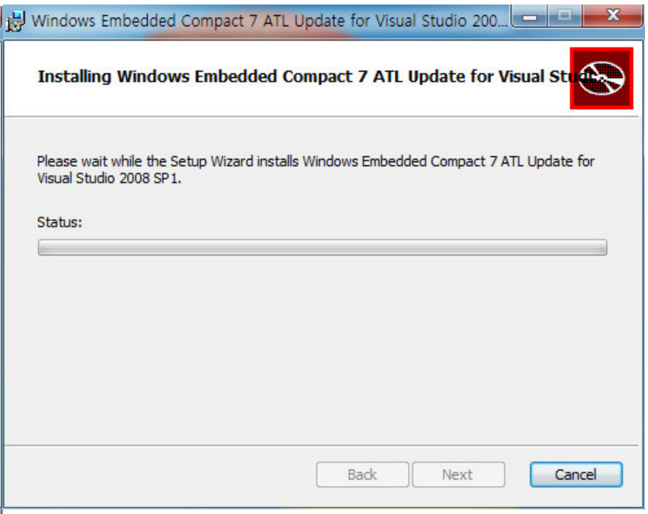
MICROSOFT VISUAL STUDIO TEAM

☑ I accept the terms in the License Agreement


Windows Embedded Compact 7 ATL Update for Visual Studio 200...
Back
Install
Cancel

Ready to install Windows Embedded Compact 7 ATL Update fo...

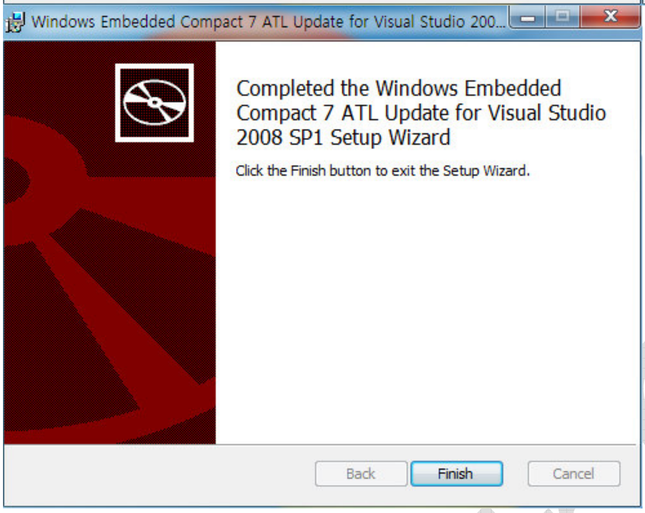
Click Install to begin the installation. Click Back to review or change any of your installation settings. Click Cancel to exit the wizard.


Windows Embedded Compact 7 ATL Update for Visual Studio 200...
Back
Next
Cancel

Installing Windows Embedded Compact 7 ATL Update for Visual Stu...

Please wait while the Setup Wizard installs Windows Embedded Compact 7 ATL Update for Visual Studio 2008 SP1.

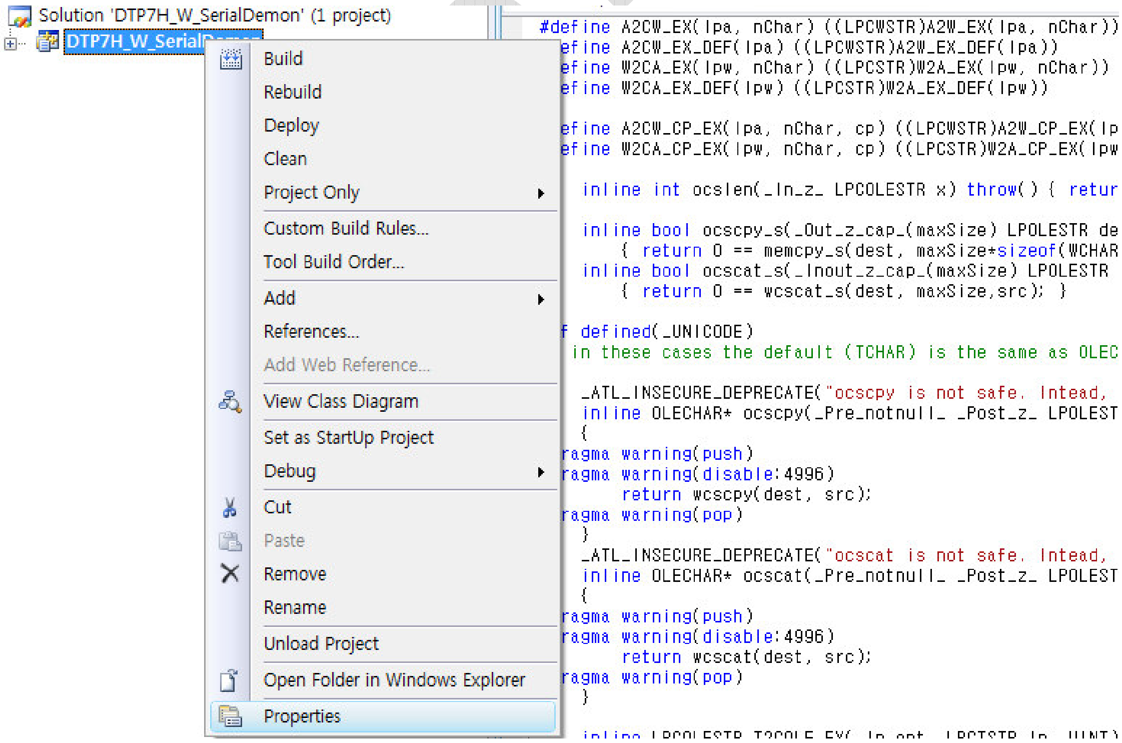
Status:


Windows Embedded Compact 7 ATL Update for Visual Studio 200...
Back
Finish
Cancel

Completed the Windows Embedded Compact 7 ATL Update for Visual Studio 2008 SP1 Setup Wizard

Click the Finish button to exit the Setup Wizard.

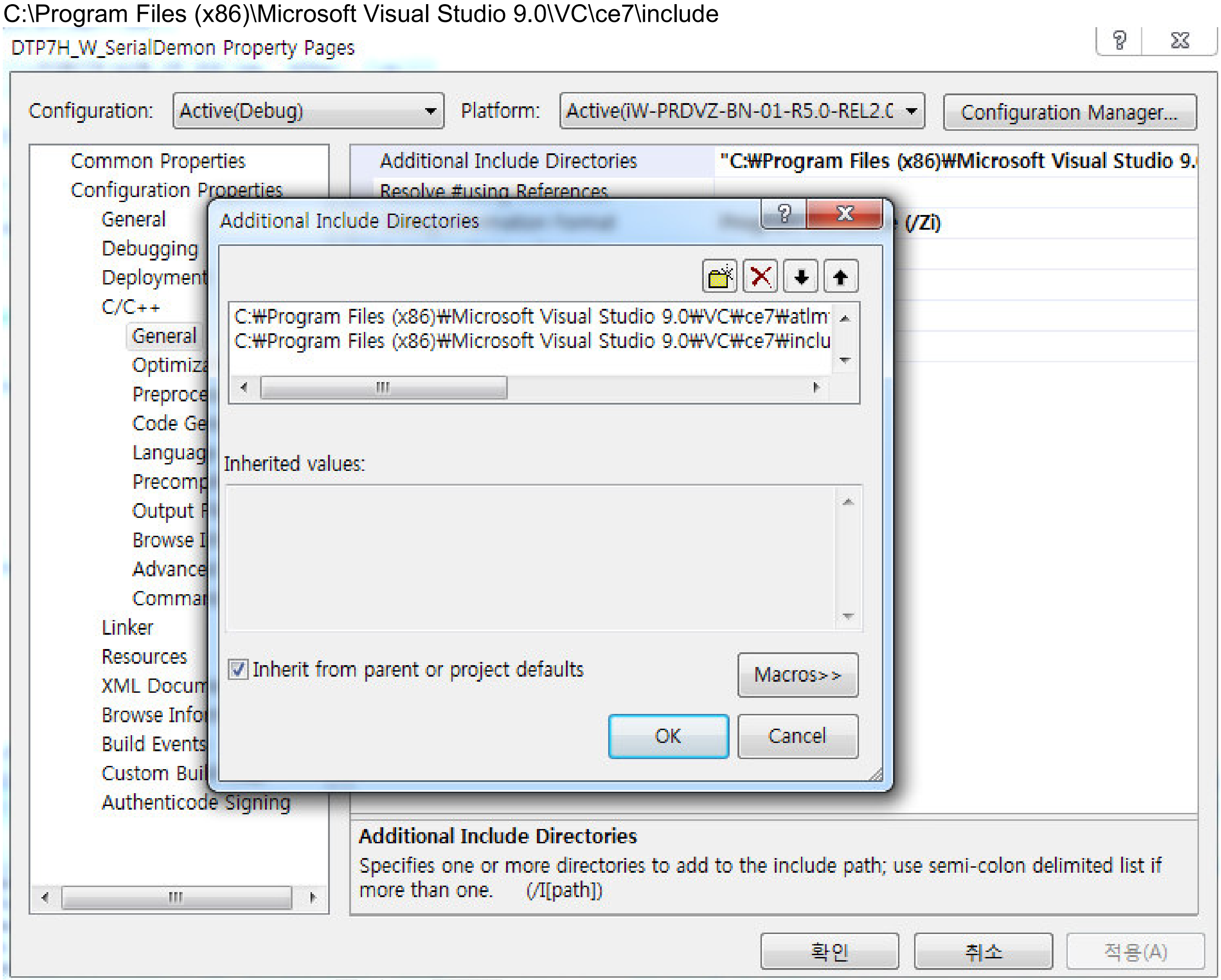
When the installation is complete, the project properties are entered.



```
#define A2CW_EX(lpa, nChar) ((LPCWSTR)A2W_EX(lpa, nChar))
efine A2CW_EX_DEF(lpa) ((LPCWSTR)A2W_EX_DEF(lpa))
efine W2CA_EX(lpw, nChar) ((LPCSTR)W2A_EX(lpw, nChar))
efine W2CA_EX_DEF(lpw) ((LPCSTR)W2A_EX_DEF(lpw))

efine A2CW_CP_EX(lpa, nChar, cp) ((LPCWSTR)A2W_CP_EX(lp
efine W2CA_CP_EX(lpw, nChar, cp) ((LPCSTR)W2A_CP_EX(lpw

 inline int ocslen(_In_z_ LPCOLESTR x) throw() { retur

 inline bool ocscpy_s(_Out_z_cap_(maxSize) LPOLESTR de
     { return 0 == memcpy_s(dest, maxSize*sizeof(WCHAR
 inline bool ocscat_s(_Inout_z_cap_(maxSize) LPOLESTR
     { return 0 == wcscat_s(dest, maxSize,src); }

f defined(_UNICODE)
 in these cases the default (TCHAR) is the same as OLEC

 _ATL_INSECURE_DEPRECATE("ocscpy is not safe. Intead,
 inline OLECHAR* ocscpy(_Pre_notnull_ _Post_z_ LPOLEST
 {
ragma warning(push)
ragma warning(disable:4996)
     return wcscpy(dest, src);
ragma warning(pop)
 }
 _ATL_INSECURE_DEPRECATE("ocscat is not safe. Intead,
 inline OLECHAR* ocscat(_Pre_notnull_ _Post_z_ LPOLEST
 {
ragma warning(push)
ragma warning(disable:4996)
     return wcscat(dest, src);
ragma warning(pop)
 }
```

Add that path to "Additional Include Directions" in " Configuration Properties -> C / C ++ -> General ".

Path:
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\ce7\atlmfc\include
C:\Program Files (x86)\Microsoft Visual Studio 9.0\VC\ce7\include



DTP7H_W_SerialDemon Property Pages

Configuration: Active(Debug)
Platform: Active(iW-PRDVZ-BN-01-R5.0-REL2.C

Additional Include Directories "C:₩Program Files (x86)₩Microsoft Visual Studio 9.

Additional Include Directories

C:₩Program Files (x86)₩Microsoft Visual Studio 9.0₩VC₩ce7₩atlm
C:₩Program Files (x86)₩Microsoft Visual Studio 9.0₩VC₩ce7₩inclu

Inherited values:

☑ Inherit from parent or project defaults

Additional Include Directories
Specifies one or more directories to add to the include path; use semi-colon delimited list if more than one. (/I[path])