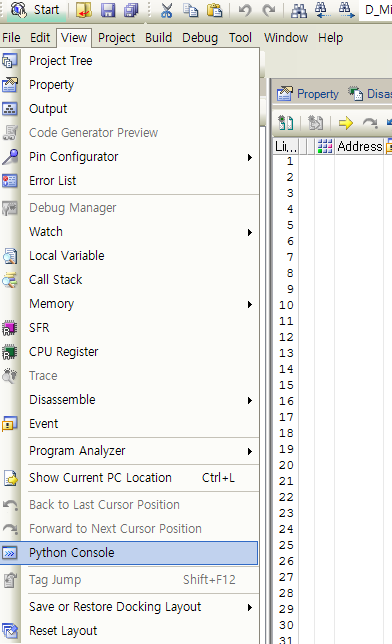
**CS+의 Python Console 기능 활용 방법**

* CS+ Python Console 도구의 Python 기본 라이브러리를 활용하여 Debug Tool(E1, E2 lite)을 제어하여 MCU의 RAM값을 참조하거나 설정이 가능하며, 프로그램 테스트 시 활용할 수 있다.
* CS+의 Watch 도구에서 읽고 쓸 수 있는 변수 값을 활용하여, Python Code 내에서 조건 및 출력으로 사용가능하며, Python 기본 라이브러리를 이용하여 파일출력(Log 출력)도 가능하다.
* MCU 프로그램의 변경 없이 원하는 대로 변수 값을 읽고, 설정하는 등의 반복적인 테스트가 가능하다.
* CS+ Python function (for debug tool) (URL 참고)

<http://tool-support.renesas.com/autoupdate/support/onlinehelp/csp/V6.00.00/CS+.chm/PythonConsole.chm/Output/py_Python_function32.html>



**1) 주요 활용 가능한 기능(function) 소개**

(1) debugger.Watch.GetValue

- 변수 값을 읽음. (Watch Tool에서 읽는 변수 값과 동일)

- Debug Tool이 지원하는 속도로 변수 값을 읽을 수 있음. (최소 100ms)

- Format :

|  |
| --- |
| debugger.Watch.GetValue(*variableName*, *encode* = Encoding.Default, *watchOption* = WatchOption.Auto) |

|  |  |  |
| --- | --- | --- |
| Argument | Description | |
| *variableName* | Specify the variable name, register name, or I/O register name/SFR register name to reference. | |
| *encode* | Examples: Encoding.utf-8, Encoding.euc-jp | |
| *watchOption* | Specify an option. | |
| Type | Description |
| WatchOption.Auto | Automatically detect when displaying (default). |
| WatchOption.Binary | Display in binary format. |
| WatchOption.Octal | Display in octal format. |
| WatchOption.Decimal | Display in decimal format. |
| WatchOption.SignedDecimal | Display in signed decimal format. |
| WatchOption.UnsignedDecimal | Display in unsigned decimal format. |
| WatchOption.Hexdecimal | Display in hexadecimal format. |
| WatchOption.String | Display as a string. |
| WatchOption.Sizeof | Display the variable size in decimal format. |
| WatchOption.Float | Display in float type. |
| WatchOption.Double | Display in double type. |

- Function 예제

(2) debugger.Watch.SetValue

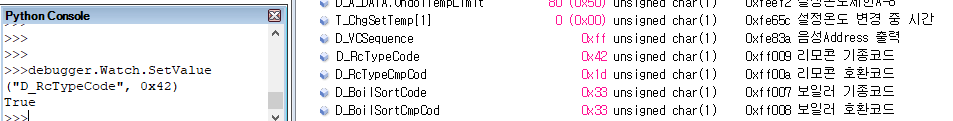
- 변수 값을 설정. (Watch Tool에서 설정 가능한 변수활용)

- Format :

|  |
| --- |
| debugger.Watch.SetValue(*variableName*, *value*) |

|  |  |
| --- | --- |
| argument | Description |
| *variableName* | Specify the variable name, register name, and I/O register name or SFR register name to set. |
| *value* | Specify the value to set. |

- Function 예제



**2) 프로그램 테스트 시 활용 예제**

(EX\_1) Wi-Fi 온도조절기의 시계 테스트 Log 출력

- 온도조절기의 시계 DATA가 변경된 경우에만 Log 기록(txt파일) 하도록 Python Code 작성

(시계변수 값("D\_Hour", "D\_Minute", "D\_RTC.Hour", "D\_RTC.Minute", "D\_RTC.Second”)중 값 변경 시 Log기록)

import time, datetime

def convertBase(number, base): # convert format

T="0123456789ABCDEF"

i,j=divmod(number,base)

if i==0:

return T[j]

else:

return convertBase(i,base)+T[j]

hexdecimal = WatchOption.Hexdecimal

**variable = ["D\_Hour", "D\_Minute", "D\_RTC.Hour", "D\_RTC.Minute", "D\_RTC.Second"]**

variable\_val = [0,0,0,0,0]

today = datetime.datetime.now()

filename = today.strftime("%Y%m%d")+'.txt'

path = 'C:\Users\jadu900142\Documents\csp\_py/'+filename

delay = 0.5

f = open(path, 'a')

i=0

data = str(datetime.datetime.now())+', '

for i in variable :

data += i+', '

f.write(data+'\n')

f.close()

while(True) :

time.sleep(delay)

i=0

j=0

flag = 0

data = str(datetime.datetime.now())+', '

for i in variable :

debugger\_watch\_get = int(**debugger.Watch.GetValue**(i, hexdecimal))

if i == variable[j] :

if j >= 2 :

debugger\_watch\_get = convertBase(debugger\_watch\_get, 16)

else :

pass

if variable\_val[j] != debugger\_watch\_get :

variable\_val[j] = debugger\_watch\_get

flag = 1

j+=1

i=0

if flag == 1 :

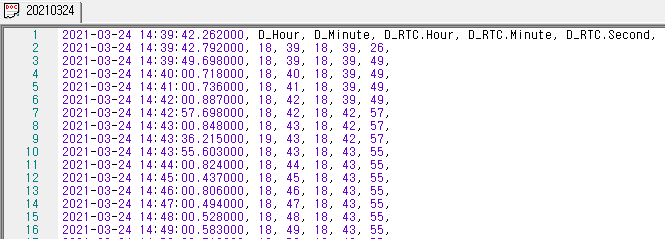
for i in variable\_val :

data += str(i)+', '

f = open(path, 'a')

f.write(data+'\n')

f.close()

* 해당 코드를 “time.py” 파일로 작성 후 CS+ 해당 프로젝트의 Python Console 에서 실행
* Log 파일(txt파일) 출력 확인 (뉴질랜드 수출용 온도조절기 시계DATA)

(EX\_2) 보일러 온도조절기 및 보일러 본체의 온도 범위 확인

- 온돌모드로 변경 후 해당 온돌설정범위를 한 단계씩 Up하여 확인하고, 실온모드로 변경 후 해당 실온설정온도 범위를 한 단계씩 Up하여 확인하는 Python code 작성

import time, datetime

path = 'C:\Users\jadu900142\Documents\csp\_py/teet\_temp.txt'

**Set\_HTemp = [35, 40, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 65, 70, 75, 80]**

**Set\_RTemp = [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40]**

indoor = 1

flow = 0

**Wth = "D\_BoilHTset"**

D\_Set\_Htemp = "C\_SetPtr[1]"

D\_Set\_Rtemp = "C\_SetPtr[2]"

**HMode = "F\_BLsts1.Bit.b4"** # flow(0) or indoor(1)

def Temp\_test(SetTemp, DSet, Watch, txtpath) :

hexdecimal = WatchOption.Hexdecimal

debugger\_watch\_get = 0

delay = 1

i=0

for i in range(len(SetTemp)) :

f = open(txtpath, 'a')

f.write(str(datetime.datetime.now())+', '+str(DSet)+' SET: , '+str(i)+'\n')

#debugger.Watch.SetValue(str(DSet), i)

if DSet == "C\_SetPtr[1]" :

debugger.Watch.SetValue("T\_ChgSetTemp[1]", 100)

else :

if i == 0 :

debugger.Watch.SetValue("T\_ChgSetTemp[2]", 250)

else :

debugger.Watch.SetValue("T\_ChgSetTemp[2]", 100)

debugger.Watch.SetValue("T\_WaitLight", 200)

debugger.Watch.SetValue(DSet, i)

while debugger\_watch\_get != SetTemp[i] :

debugger\_watch\_get = int(debugger.Watch.GetValue(str(Watch), hexdecimal))

f.write(str(datetime.datetime.now())+', '+str(Watch)+' Watch: , '+str(debugger\_watch\_get)+'\n')

if SetTemp[i] == SetTemp[len(SetTemp)-1] :

f.write(str(datetime.datetime.now())+', '+str(Watch)+' Watch: , ''TEST OK'+'\n')

f.close()

time.sleep(delay)

def mode\_test(DSet, Mode, txtpath) :

hexdecimal = WatchOption.Hexdecimal

while int(debugger.Watch.GetValue(str(DSet), hexdecimal)) != Mode :

debugger.Watch.SetValue("T\_WaitLight", 200)

debugger.Watch.SetValue("D\_SwCom", 0x09)

time.sleep(0.5)

if Mode == 0 :

s = "flow mode"

else :

s = "indoor mode"

f = open(txtpath, 'a')

f.write(str(datetime.datetime.now())+', '+str(DSet)+' Watch: , '+str(s)+'\n')

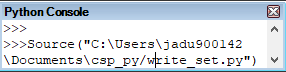
f.close()

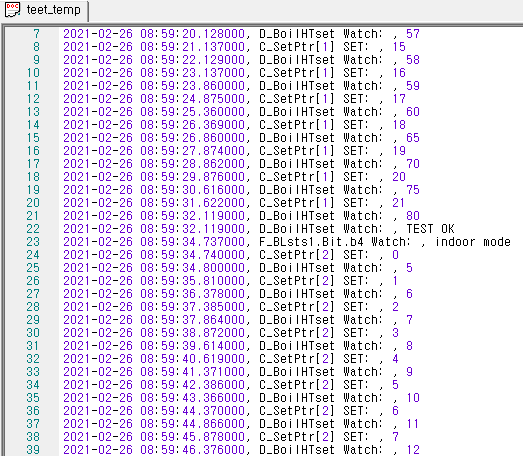
mode\_test(HMode, 0, path) # flow mode

Temp\_test(Set\_HTemp, D\_Set\_Htemp, Wth, path) # flow temp

mode\_test(HMode, 1, path) # indoor mode

Temp\_test(Set\_RTemp, D\_Set\_Rtemp, Wth, path) # indoor temp

* 해당 코드를 “write\_set” 파일로 작성 후 CS+ 해당 프로젝트의 Python Console 에서 실행
* Log 파일(txt파일) 출력 확인.

(Python Code에서 설정한 온도Pointer와 보일러 본체로부터 수신받은 설정온도 확인)

**3. 주의사항**

- Python Code 내에서 한글 주석 사용 불가 (CS+에서 실행 시 Error 발생)

- 한번에 실행하는 Python Cdoe내에서 호출하는 함수가 많거나 코드 길이가 길어지면 CS+에서 Error 발생.

- CS+에서는 .NET 기반의 Ironpython을 사용하기 때문에 python2 문법을 사용.

참고(예제코드) : \\rdsvr\_se\SW\_Team\공통문서\10\_개인자료\12\_손선국\python