

eLogger Web API DLL -register.dll 應用說明書-



Version 1.0.2 ,2012/10/29

Written by Amber Hsieh

Warranty

All products manufactured by ICP DAS are under warranty regarding defective materials for a period of one year, beginning from the date of delivery to the original purchaser.

Warning

ICP DAS assumes no liability for any damage resulting from the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, not for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Copyright © 2012 by ICP DAS Co., Ltd. All rights are reserved.

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Contact US

If you have any problem, please feel free to contact us.
You can count on us for quick response.

Email: service@icpdas.com

目錄

1. 介紹	5
2. register.dll的函數說明	6
2.1. 讀取記憶體資料函數	7
2.1.1 readAI=xxxxx.....	8
2.1.2 readAI_UInt=xxxxx.....	9
2.1.3 readAI_Long=xxxxx.....	9
2.1.4 readAI_ULong=xxxxx.....	9
2.1.5 readAI_Float=xxxxx.....	9
2.1.6 readAI=xxxxx&len=ooo	10
2.1.7 readAI_UInt=xxxxx&len=ooo	11
2.1.8 readAO=xxxxx.....	12
2.1.9 readAO_UInt=xxxxx.....	13
2.1.10 readAO_Long=xxxxx.....	13
2.1.11 readAO_ULong=xxxxx.....	13
2.1.12 readAO_Float=xxxxx.....	13
2.1.13 readAO=xxxxx&len=ooo	14
2.1.14 readAO_UInt=xxxxx&len=ooo	15
2.1.15 readDI=xxxxx.....	16
2.1.16 readDI=xxxxx&len=ooo	17
2.1.17 readDO=xxxxx.....	18
2.1.18 readDO=xxxxx&len=ooo	19
2.2. 寫入記憶體資料函數	20
2.1.1 writeAO=xxxxx&data=value	21
2.1.2 writeAO_UInt=xxxxx&data=value	22
2.1.3 writeAO_Long=xxxxx&data=value	22
2.1.4 writeAO_ULong=xxxxx&data=value	22
2.1.5 writeAO_Float=xxxxx&data=value	22
2.1.6 writeDO=xxxxx&data=value	23
2.3. 錯誤碼說明.....	24

3	如何設定讀取register.dll	25
3.1.	Windows CE5 based PACs	26
3.2.	Windows CE6 based PACs	27
3.3.	Windows Embedded Standard 2009 PACs	28
4	如何開發Android App程式	31
	範例 1：讀取eLogger Shared Memory AI第 1 個位址的資料	31
	範例 2：寫入資料到eLogger Shared Memory DO第 0 個位址	37
	附錄 A. 修訂紀錄	42

1. 介紹

eLogger 為泓格所開發的一套免費數據採集軟體，您可以使用 eLogger 來規劃泓格以下所列 PAC 控制器上的人機介面，去快速、簡單地架設 I/O 監視和控制系統：

- Windows CE .NET 5.0 based PACs(WinPAC、ViewPAC)、
- Windows CE .NET 6.0 based PACs(XP8000-CE6 系列)、
- Windows Embedded Standard 2009 PACs(XP8000 系列)

爲了方便使用者可以從遠端透過網路了解 I/O 模組的狀態，我們開發了一個 Web API DLL-“register.dll”，藉由 HTTP 協定來讀取和寫入 eLogger 的 Shared Memory 資料。使用者只需開啓瀏覽器存取這個 DLL 文件，並輸入符合格式的參數即得到欲讀取和寫入位址的數值資料，若輸入的 URL 錯誤則會回傳一串“錯誤訊息”，因此使用者可以利用 register.dll 來開發網頁或是手機應用程式去做遠端 I/O 的監控。

eLogger + register.dll



Ethernet

輸入URL讀寫資料



2. register.dll 的函數說明

eLogger 包含了一個記憶體位址的內存，register.dll 提供了 24 種參數方法讓使用者可以依照自己的需求去讀寫 eLogger 的 SharedMemory 資料，其 URL 的輸入格式為：**http://IP 位址/.../register.dll?參數定義**

範例 1:

如果 register.dll 檔案存放在網站根目錄中，URL 設定方式如下圖所示：

http://192.168.0.1/register.dll?readAI=00000

PAC 控制器的 IP 位址

在問號(?)之後填入參數讀寫資料

範例 2:

如果 register.dll 檔案存放在網站根目錄底下的其他目錄中，URL 設定方式如下圖所示：

http://192.168.0.1/eLogger/register.dll?readAI=00000

指定 DLL 在 PAC 控制器中的路徑位址

在問號(?)之後填入參數讀寫資料

2.1. 讀取記憶體資料函數

Read API 函數說明

使用者可以使用 Read APIs 直接讀取 eLogger Shared Memory 的資料。

函數功能列表

函數	敘述
readAI=xxxxx	讀取 AI 單一位址的"整數"型態 Analog Input 數值
readAI_UInt=xxxxx	讀取 AI 單一位址的"無符號整數"型態 Analog Input 數值
readAI_Long=xxxxx	讀取 AI 單一位址的"長整數"型態 Analog Input 數值
readAI_ULong=xxxxx	讀取 AI 單一位址的"無符號長整數"型態 Analog Input 數值
readAI_Float=xxxxx	讀取 AI 單一位址的"浮點數"型態 Analog Input 數值
readAI=xxxxx&len=ooo	讀取 AI 多個連續位址的"整數"型態 Analog Input 數值
readAI_UInt=xxxxx&len=ooo	讀取 AI 多個連續位址的"無符號整數"型態 Analog Input 數值
readAO=xxxxx	讀取 AO 單一位址的"整數"型態 Analog Output 數值
readAO_UInt=xxxxx	讀取 AO 單一位址的"無符號整數"型態 Analog Output 數值
readAO_Long=xxxxx	讀取 AO 單一位址的"長整數"型態 Analog Output 數值
readAO_ULong=xxxxx	讀取 AO 單一位址的"無符號長整數"型態 Analog Output 數值
readAO_Float=xxxxx	讀取 AO 單一位址的"浮點數"型態 Analog Output 數值
readAO=xxxxx&len=ooo	讀取 AO 多個連續位址的"整數"型態 Analog Output 數值
readAO_UInt=xxxxx&len=ooo	讀取 AO 多個連續位址的"無符號整數"型態 Analog Output 數值
readDI=xxxxx	讀取 DI 單一位址的 Digital Input 狀態
readDI=xxxxx&len=ooo	讀取 DI 多個連續位址的 Digital Input 狀態
readDO=xxxxx	讀取 DO 單一位址的 Digital Output 狀態
readDO=xxxxx&len=ooo	讀取 DO 多個連續位址的 Digital Output 狀態

2.1.1 readAI=xxxxx

使用此參數可以讀取 AI 單一位址的”整數”型態 Analog Input 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大寫的限制。

範例:

<http://10.0.0.183/register.dll?readAI=00010>

讀取 eLogger Shared Memory-AI 第 10 個位址的 Analog Input 數值。

The screenshot shows a software interface with a table of memory addresses and a browser window. The table lists memory addresses from InputRegister[0] to InputRegister[11], with names like Sin, Rnd, and Input Register1 through Input Register10. The browser window shows the URL <http://10.0.0.183/register.dll?readAI=00010> and the result 55. Red circles highlight the 'AI Tag' in the software interface and the URL in the browser. A red arrow points from the 'AI Tag' to the browser window.

Memory Address	Name	Location	Description
InputRegister[0]	Sin	Math Curve->MathCurvID1->Sin	The value of the Sin.
InputRegister[1]	Rnd	Math Curve->MathCurvID1->Rnd	Random value.
InputRegister[2]	Input Register1	ModbusSerial->COM1_ID1->Input Register1	COM1_ID1_Address:30001
InputRegister[3]	Input Register2	ModbusSerial->COM1_ID1->Input Register2	COM1_ID1_Address:30002
InputRegister[4]	Input Register3	ModbusSerial->COM1_ID1->Input Register3	COM1_ID1_Address:30003
InputRegister[5]	Input Register4	ModbusSerial->COM1_ID1->Input Register4	COM1_ID1_Address:30004
InputRegister[6]	Input Register5	ModbusSerial->COM1_ID1->Input Register5	COM1_ID1_Address:30005
InputRegister[7]	Input Register6	ModbusSerial->COM1_ID1->Input Register6	COM1_ID1_Address:30006
InputRegister[8]	Input Register7	ModbusSerial->COM1_ID1->Input Register7	COM1_ID1_Address:30007
InputRegister[9]	Input Register8	ModbusSerial->COM1_ID1->Input Register8	COM1_ID1_Address:30008
InputRegister[10]	Input Register9	ModbusSerial->COM1_ID1->Input Register9	COM1_ID1_Address:30009
InputRegister[11]	Input Register10	ModbusSerial->COM1_ID1->Input Register10	COM1_ID1_Address:30010

Internet Explorer window: <http://10.0.0.183/register.dll?readAI=00010> - Windows Internet Explorer

Result: 55

2.1.2 readAI_UInt=xxxxx

使用此參數可以讀取 AI 單一位址的”無符號整數”型態 Analog Input 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.3 readAI_Long=xxxxx

使用此參數可以讀取 AI 單一位址的”長整數”型態 Analog Input 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.4 readAI_ULong=xxxxx

使用此參數可以讀取 AI 單一位址的”無符號長整數”型態 Analog Input 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.5 readAI_Float=xxxxx

使用此參數可以讀取 AI 單一位址的”浮點數”型態 Analog Input 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.6 readAI=xxxxx&len=ooo

使用此參數可以讀取 AI 多個連續位址的”整數”型態 Analog Input 數值，”xxxxx”表示欲讀取的起始位址，其格式一率須填五位數(假若您設起始位址為 10，請填入 00010，其他以此類推)，可設定的起始區間為 00000~99999；”ooo”表示欲讀取的位址長度，其格式一率須填三位數(假若您要讀取 4 個位址，請填入 004，其他以此類推)，可讀取的最大長度為 256 個；要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開，而且每個數值為 6 位數的固定長度(如果數值小於 6 位數，則前面用空格補滿)，例如：
“ 32767, 12,-32768, -10”，這樣方便使用者在自己的應用程式處理回傳資料時，可以取長度或是取逗號來分割數值。

範例:

<http://10.0.0.183/register.dll?readAI=00000&len=005>

讀取 eLogger Shared Memory-AI 第 0~第 4 個位址的 Analog Input 數值。

Memory Address	Name	Location	Description
InputRegister[0]	Sin	Math Curve->MathCurvID1->Sin	The value of the Sin.
InputRegister[1]	Rnd	Math Curve->MathCurvID1->Rnd	Random value.
InputRegister[2]	Input Register1	ModbusSerial->COM2_ID1->Input Register1	COM2_ID1_Address:30001
InputRegister[3]	Input Register2	ModbusSerial->COM2_ID1->Input Register2	COM2_ID1_Address:30002
InputRegister[4]	Input Register3	ModbusSerial->COM2_ID1->Input Register3	COM2_ID1_Address:30003
InputRegister[5]	Input Register4	ModbusSerial->COM2_ID1->Input Register4	COM2_ID1_Address:30004
InputRegister[6]	Input Register5	ModbusSerial->COM2_ID1->Input Register5	COM2_ID1_Address:30005
InputRegister[7]	Input Register6	ModbusSerial->COM2_ID1->Input Register6	COM2_ID1_Address:30006
InputRegister[8]	Input Register7	ModbusSerial->COM2_ID1->Input Register7	COM2_ID1_Address:30007
InputRegister[9]	Input Register8	ModbusSerial->COM2_ID1->Input Register8	COM2_ID1_Address:30008
InputRegister[10]	Input Register9	ModbusSerial->COM2_ID1->Input Register9	COM2_ID1_Address:30009
InputRegister[11]	Input Register10	ModbusSerial->COM2_ID1->Input Register10	COM2_ID1_Address:30010

2.1.7 readAI_UInt=xxxxx&len=ooo

使用此參數可以讀取 AI 多個連續位址的“無符號整數”型態 Analog Input 數值，“xxxxx”表示欲讀取的起始位址，其格式一率須填五位數(假若您設起始位址為 10，請填入 00010，其他以此類推)，可設定的起始區間為 00000~99999；“ooo”表示欲讀取的位址長度，其格式一率須填三位數(假若您要讀取 4 個位址，請填入 004，其他以此類推)，可讀取的最大長度為 256 個；要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開，而且每個數值為 6 位數的固定長度(如果數值小於 6 位數，則前面用空格補滿)，例如：
“ 32767, 12, 65535, 888”，這樣方便使用者在自己的應用程式處理回傳資料時，可以取長度或是取逗號來分割數值。

2.1.8 readAO=xxxxx

使用此參數可以讀取 AO 單一位址的“整數”型態 Analog Output 數值，
“xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，
請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參
數有大小寫的限制。

範例:

<http://10.0.0.183/register.dll?readAO=00005>

讀取 eLogger Shared Memory-AO 第 5 個位址的 Analog Output 數值。

The image shows a software interface with a table of holding registers and a browser window. The table lists holding registers from 0 to 9, each with a name, location, and description. The register at index 5 is highlighted with a red circle. Below the table, a browser window shows the URL <http://10.0.0.183/register.dll?readAO=00005> and the result -985.

Memory Address	Name	Location	Description
HoldingRegister[0]	Holding Register1	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40001
HoldingRegister[1]	Holding Register2	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40002
HoldingRegister[2]	Holding Register3	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40003
HoldingRegister[3]	Holding Register4	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40004
HoldingRegister[4]	Holding Register5	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40005
HoldingRegister[5]	Holding Register6	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40006
HoldingRegister[6]	Holding Register7	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40007
HoldingRegister[7]	Holding Register8	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40008
HoldingRegister[8]	Holding Register9	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40009
HoldingRegister[9]	Holding Register...	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40010

Browser window: <http://10.0.0.183/register.dll?readAO=00005> - Windows Internet Explorer
Result: -985

2.1.9 readAO_UInt=xxxxx

使用此參數可以讀取 AO 單一位址的”無符號整數”型態 Analog Output 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.10 readAO_Long=xxxxx

使用此參數可以讀取 AO 單一位址的”長整數”型態 Analog Output 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.11 readAO_ULong=xxxxx

使用此參數可以讀取 AO 單一位址的”無符號長整數”型態 Analog Output 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.12 readAO_Float=xxxxx

使用此參數可以讀取 AO 單一位址的”浮點數”型態 Analog Output 數值，”xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

2.1.13 readAO=xxxxx&len=ooo

使用此參數可以讀取 AO 多個連續位址的”整數”型態 Analog Output 數值，”xxxxx”表示欲讀取的起始位址，其格式一率須填五位數(假若您設起始位址為 10，請填入 00010，其他以此類推)，可設定的起始區間為 00000~99999；”ooo”表示欲讀取的位址長度，其格式一率須填三位數(假若您要讀取 4 個位址，請填入 004，其他以此類推)，可讀取的最大長度為 256 個；要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開，而且每個數值為 6 位數的固定長度(如果數值小於 6 位數，則前面用空格補滿)，例如：
“ 32767, 12, 65535, -10”，這樣方便使用者在自己的應用程式處理回傳資料時，可以取長度或是取逗號來分割數值。

範例:

<http://10.0.0.183/register.dll?readAO=00001&len=009>

讀取 eLogger Shared Memory-AO 第 1~第 9 個位址的 Analog Output 數值。

The screenshot shows the eLogger software interface. On the left, a tree view shows 'AO Tag' selected. The main window displays a table of Holding Registers:

Memory Address	Name	Location	Description
HoldingRegister[1]	Holding Register2	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40001
HoldingRegister[2]	Holding Register3	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40002
HoldingRegister[3]	Holding Register4	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40003
HoldingRegister[4]	Holding Register5	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40004
HoldingRegister[5]	Holding Register6	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40005
HoldingRegister[6]	Holding Register7	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40006
HoldingRegister[7]	Holding Register8	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40007
HoldingRegister[8]	Holding Register9	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40008
HoldingRegister[9]	Holding Register...	ModbusSerial->COM2_ID1->Holding Regist...	COM2_ID1_Address:40009

Below the table, a Windows Internet Explorer window shows the URL <http://10.0.0.183/register.dll?readAO=00001&len=009> and the response: 1413, 3160, -30402, 2379, -47, 0, 0, 4262, 0.

2.1.14 readAO_UInt=xxxxx&len=ooo

使用此參數可以讀取 AO 多個連續位址的“無符號整數”型態 Analog Output 數值，“xxxxx”表示欲讀取的起始位址，其格式一率須填五位數(假若您設起始位址為 10，請填入 00010，其他以此類推)，可設定的起始區間為 00000~99999；“ooo”表示欲讀取的位址長度，其格式一率須填三位數(假若您要讀取 4 個位址，請填入 004，其他以此類推)，可讀取的最大長度為 256 個；要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開，而且每個數值為 6 位數的固定長度(如果數值小於 6 位數，則前面用空格補滿)，例如：“ 32767, 12, 65535, 888”，這樣方便使用者在自己的應用程式處理回傳資料時，可以取長度或是取逗號來分割數值。

2.1.15 readDI=xxxxx

使用此參數可以讀取 DI 單一位址的 Digital Input 狀態(1=On;0=Off)，“xxxxx”表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大寫的限制。

範例:

<http://10.0.0.183/register.dll?readDI=00003>

讀取 eLogger Shared Memory-DI 第 3 個位址的 Digital Input 狀態。

The screenshot shows a software interface with a tree view on the left and a table on the right. The tree view includes 'System', 'Driver(New)', 'Math Curve', 'MathC...', 'ModbusSe...', 'COM2_...', 'Tag Mapping', 'AI Tag', 'AO Tag', 'DI Tag', and 'DO Tag'. The 'DI Tag' is highlighted with a red circle. The table on the right has columns: 'Memory Ad...', 'Name', 'Location', 'Description', and 'Note'. The table contains 10 rows of data for Input Status 1 through 10. The row for 'Input Status3' is highlighted with a red circle. Below the table is a Windows Internet Explorer browser window showing the URL 'http://10.0.0.183/register.dll?readDI=00003' in the address bar, which is also highlighted with a red circle. The browser content area displays the number '1'.

Memory Ad...	Name	Location	Description	Note
InputStatus[0]	Input Status1	ModbusSerial->COM2_ID1->Input Status1	COM2_ID1_Address:10001	
InputStatus[1]	Input Status2	ModbusSerial->COM2_ID1->Input Status2	COM2_ID1_Address:10002	
InputStatus[2]	Input Status3	ModbusSerial->COM2_ID1->Input Status3	COM2_ID1_Address:10003	
InputStatus[3]	Input Status4	ModbusSerial->COM2_ID1->Input Status4	COM2_ID1_Address:10004	
InputStatus[4]	Input Status5	ModbusSerial->COM2_ID1->Input Status5	COM2_ID1_Address:10005	
InputStatus[5]	Input Status6	ModbusSerial->COM2_ID1->Input Status6	COM2_ID1_Address:10006	
InputStatus[6]	Input Status7	ModbusSerial->COM2_ID1->Input Status7	COM2_ID1_Address:10007	
InputStatus[7]	Input Status8	ModbusSerial->COM2_ID1->Input Status8	COM2_ID1_Address:10008	
InputStatus[8]	Input Status9	ModbusSerial->COM2_ID1->Input Status9	COM2_ID1_Address:10009	
InputStatus[9]	Input Status10	ModbusSerial->COM2_ID1->Input Status10	COM2_ID1_Address:10010	

2.1.16 readDI=xxxxx&len=ooo

使用此參數可以讀取 DI 多個連續位址的 Digital Input 狀態(1=On ; 0=Off) , "xxxxx"表示欲讀取的起始位址 , 其格式一率須填五位數(假若您設起始位址為 10 , 請填入 00010 , 其他以此類推) , 可設定的起始區間為 00000~99999 ; "ooo"表示欲讀取的位址長度 , 其格式一率須填三位數(假若您要讀取 4 個位址 , 請填入 004 , 其他以此類推) , 可讀取的最大長度為 256 個 ; 要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開 , 而且每個數值為 6 位數的固定長度(如果數值小於 6 位數 , 則前面用空格補滿) , 例如 : " 1, 1, 0, 1" , 這樣方便使用者在自己的應用程式處理回傳資料時 , 可以取長度或是取逗號來分割數值。

範例:

<http://10.0.0.183/register.dll?readDI=00000&len=005>

讀取 eLogger Shared Memory-DI 第 0~第 5 個位址的 Digital Input 狀態。

The screenshot shows a software interface with a table of Digital Input (DI) status and a web browser displaying the result of a readDI request.

Memory Ad.	Name	Location	Description
InputStatus[0]	Input Status1	ModbusSerial->COM2_ID1->Input Status1	COM2_ID1_Address:10001
InputStatus[1]	Input Status2	ModbusSerial->COM2_ID1->Input Status2	COM2_ID1_Address:10002
InputStatus[2]	Input Status3	ModbusSerial->COM2_ID1->Input Status3	COM2_ID1_Address:10003
InputStatus[3]	Input Status4	ModbusSerial->COM2_ID1->Input Status4	COM2_ID1_Address:10004
InputStatus[4]	Input Status5	ModbusSerial->COM2_ID1->Input Status5	COM2_ID1_Address:10005
InputStatus[5]	Input Status6	ModbusSerial->COM2_ID1->Input Status6	COM2_ID1_Address:10006
InputStatus[6]	Input Status7	ModbusSerial->COM2_ID1->Input Status7	COM2_ID1_Address:10007
InputStatus[7]	Input Status8	ModbusSerial->COM2_ID1->Input Status8	COM2_ID1_Address:10008
InputStatus[8]	Input Status9	ModbusSerial->COM2_ID1->Input Status9	COM2_ID1_Address:10009
InputStatus[9]	Input Status10	ModbusSerial->COM2_ID1->Input Status10	COM2_ID1_Address:10010

The web browser displays the result of the readDI request: "1, 0, 0, 1, 1".

2.1.17 readDO=xxxxx

使用此參數可以讀取 DO 單一位址的 Digital Output 狀態(1=On ; 0=Off) , "xxxxx"表示欲讀取的位址，其格式一率須填五位數(假若您要讀取第 10 個位址，請填入 00010，其他以此類推)，可讀取的位址區間為 00000~99999；要注意參數有大小寫的限制。

範例:

<http://10.0.0.183/register.dll?readDO=00011>

讀取 eLogger Shared Memory-DO 第 11 個位址的 Digital Output 狀態。

The screenshot shows a software interface with a table of coil statuses and a web browser window. The table lists coil statuses from 0 to 14, with Coil 11 highlighted in red. The web browser window shows the URL <http://10.0.0.183/register.dll?readDO=00011> and the result '0'.

Memory Ad...	Name	Location	Description	Note
CoilStatus[0]	Coil1	ModbusSerial->COM2_ID1->Coil1	COM2_ID1_Address:00001	
CoilStatus[1]	Coil2	ModbusSerial->COM2_ID1->Coil2	COM2_ID1_Address:00002	
CoilStatus[2]	Coil3	ModbusSerial->COM2_ID1->Coil3	COM2_ID1_Address:00003	
CoilStatus[3]	Coil4	ModbusSerial->COM2_ID1->Coil4	COM2_ID1_Address:00004	
CoilStatus[4]	Coil5	ModbusSerial->COM2_ID1->Coil5	COM2_ID1_Address:00005	
CoilStatus[5]	Coil6	ModbusSerial->COM2_ID1->Coil6	COM2_ID1_Address:00006	
CoilStatus[6]	Coil7	ModbusSerial->COM2_ID1->Coil7	COM2_ID1_Address:00007	
CoilStatus[7]	Coil8	ModbusSerial->COM2_ID1->Coil8	COM2_ID1_Address:00008	
CoilStatus[8]	Coil9	ModbusSerial->COM2_ID1->Coil9	COM2_ID1_Address:00009	
CoilStatus[9]	Coil10	ModbusSerial->COM2_ID1->Coil10	COM2_ID1_Address:00010	
CoilStatus[10]	Coil11	ModbusSerial->COM2_ID1->Coil11	COM2_ID1_Address:00011	
CoilStatus[11]	Coil12	ModbusSerial->COM2_ID1->Coil12	COM2_ID1_Address:00012	
CoilStatus[12]	Coil13	ModbusSerial->COM2_ID1->Coil13	COM2_ID1_Address:00013	
CoilStatus[13]	Coil14	ModbusSerial->COM2_ID1->Coil14	COM2_ID1_Address:00014	
CoilStatus[14]	Coil15	ModbusSerial->COM2_ID1->Coil15	COM2_ID1_Address:00015	

Windows Internet Explorer window showing the URL <http://10.0.0.183/register.dll?readDO=00011> and the result '0'.

2.1.18 readDO=xxxxx&len=ooo

使用此參數可以讀取 DO 多個連續位址的 Digital Output 狀態(1=On ; 0=Off)，“xxxxx”表示欲讀取的起始位址，其格式一率須填五位數(假若您設起始位址為 10，請填入 00010，其他以此類推)，可設定的起始區間為 00000~99999；“ooo”表示欲讀取的位址長度，其格式一率須填三位數(假若您要讀取 4 個位址，請填入 004，其他以此類推)，可讀取的最大長度為 256 個；要注意參數有大小寫的限制。使用此參數方法回傳的每一個數值中間是以逗點符號區隔開，而且每個數值為 6 位數的固定長度(如果數值小於 6 位數，則前面用空格補滿)，例如：“ 1, 1, 0, 1”，這樣方便使用者在自己的應用程式處理回傳資料時，可以取長度或是取逗號來分割數值。

範例:

<http://10.0.0.183/register.dll?readDO=00010&len=005>

讀取 eLogger Shared Memory-DO 第 10~第 14 個位址的全部 Digital Output 狀態。

The screenshot shows the eLogger software interface with a table of DO tags and a browser window displaying the result of a readDO request.

Memory Ad...	Name	Location	Description	Note
CoilStatus[0]	Coil1	ModbusSerial->COM2_ID1->Coil1	COM2_ID1_Address:00001	
CoilStatus[1]	Coil2	ModbusSerial->COM2_ID1->Coil2	COM2_ID1_Address:00002	
CoilStatus[2]	Coil3	ModbusSerial->COM2_ID1->Coil3	COM2_ID1_Address:00003	
CoilStatus[3]	Coil4	ModbusSerial->COM2_ID1->Coil4	COM2_ID1_Address:00004	
CoilStatus[4]	Coil5	ModbusSerial->COM2_ID1->Coil5	COM2_ID1_Address:00005	
CoilStatus[5]	Coil6	ModbusSerial->COM2_ID1->Coil6	COM2_ID1_Address:00006	
CoilStatus[6]	Coil7	ModbusSerial->COM2_ID1->Coil7	COM2_ID1_Address:00007	
CoilStatus[7]	Coil8	ModbusSerial->COM2_ID1->Coil8	COM2_ID1_Address:00008	
CoilStatus[8]	Coil9	ModbusSerial->COM2_ID1->Coil9	COM2_ID1_Address:00009	
CoilStatus[9]	Coil10	ModbusSerial->COM2_ID1->Coil10	COM2_ID1_Address:00010	
CoilStatus[10]	Coil11	ModbusSerial->COM2_ID1->Coil11	COM2_ID1_Address:00011	
CoilStatus[11]	Coil12	ModbusSerial->COM2_ID1->Coil12	COM2_ID1_Address:00012	
CoilStatus[12]	Coil13	ModbusSerial->COM2_ID1->Coil13	COM2_ID1_Address:00013	
CoilStatus[13]	Coil14	ModbusSerial->COM2_ID1->Coil14	COM2_ID1_Address:00014	
CoilStatus[14]	Coil15	ModbusSerial->COM2_ID1->Coil15	COM2_ID1_Address:00015	

The browser window shows the URL <http://10.0.0.183/register.dll?readDO=00010&len=005> and the response "0, 1, 1, 0, 0".

2.2. 寫入記憶體資料函數

Write API 函數說明

使用者可以使用 Write APIs 直接寫入指定的數值到 eLogger Shared Memory 中。

函數功能列表

函數功能	敘述
writeAO=xxxxx&data=value	寫入一個整數型態的 Analog Output 數值至使用者所指定的 AO 位址中
writeAO_UInt=xxxxx&data=value	寫入一個無符號整數型態的 Analog Output 數值至使用者所指定的 AO 位址中
writeAO_Long=xxxxx&data=value	寫入一個長整數型態的 Analog Output 數值至使用者所指定的 AO 位址中
writeAO_ULong=xxxxx&data=value	寫入一個無符號長整數型態的 Analog Output 數值至使用者所指定的 AO 位址中
writeAO_float=xxxxx&data=value	寫入一個浮點數型態的 Analog Output 數值至使用者所指定的 AO 位址中
writeDO=xxxxx&data=value	改寫指定 DO 位址的 Digital Output 狀態

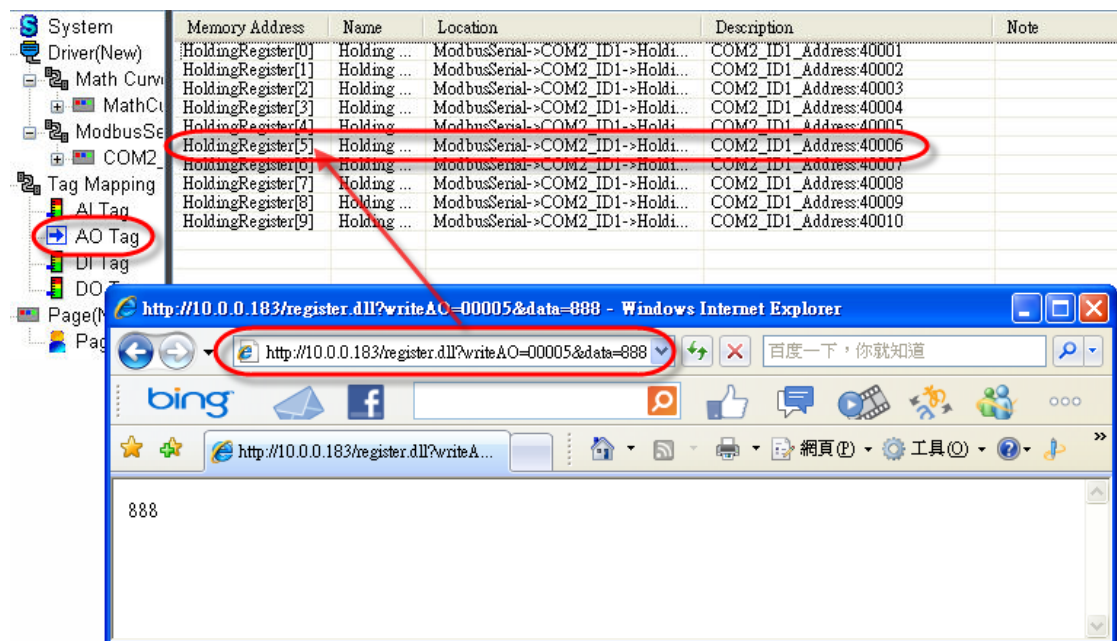
2.1.1 writeAO=xxxxx&data=value

使用此參數可以寫入一個”整數”型態的 Analog Output 數值至您指定的 AO 位址中, ”xxxxx”表示欲寫入的指定位址, 其格式一率須填五位數(假若您要改寫第 10 個位址, 請填入 00010, 其他以此類推), 可寫入的位址區間為 00000~99999; ”value”表示欲寫入的”整數”型態數值; 要注意參數有大小寫的限制。

範例:

<http://10.0.0.183/register.dll?writeAO=00005&data=888>

寫入數值 888 到 eLogger Shared Memory-AO 的第 5 個位址。



2.1.2 writeAO_UInt=xxxxx&data=value

使用此參數可以寫入一個“無符號整數”型態的 Analog Output 數值至您指定的 AO 位址中，“xxxxx”表示欲寫入的指定位址，其格式一率須填五位數(假若您要改寫第 10 個位址，請填入 00010，其他以此類推)，可寫入的位址區間為 00000~99999；“value”表示欲寫入的“無符號整數”型態數值；要注意參數有大小寫的限制。

2.1.3 writeAO_Long=xxxxx&data=value

使用此參數可以寫入一個“長整數”型態的 Analog Output 數值至您指定的 AO 位址中，“xxxxx”表示欲寫入的指定位址，其格式一率須填五位數(假若您要改寫第 10 個位址，請填入 00010，其他以此類推)，可寫入的位址區間為 00000~99999；“value”表示欲寫入的“長整數”型態數值；要注意參數有大小寫的限制。

2.1.4 writeAO_ULong=xxxxx&data=value

使用此參數可以寫入一個“無符號長整數”型態的 Analog Output 數值至您指定的 AO 位址中，“xxxxx”表示欲寫入的指定位址，其格式一率須填五位數(假若您要改寫第 10 個位址，請填入 00010，其他以此類推)，可寫入的位址區間為 00000~99999；“value”表示欲寫入的“無符號長整數”型態數值；要注意參數有大小寫的限制。

2.1.5 writeAO_Float=xxxxx&data=value

使用此參數可以寫入一個“浮點數”型態的 Analog Output 數值至您指定的 AO 位址中，“xxxxx”表示欲寫入的指定位址，其格式一率須填五位數(假若您要改寫第 10 個位址，填入 00010，其他以此類推)，可寫入位址區間為 00000~99999；“value”表示欲寫入的“浮點數”型態數值；要注意參數有大小寫的限制。

2.1.6 writeDO=xxxxx&data=value

使用此參數可以改寫您指定的 DO 位址的 Digital Output 狀態(1=On ; 0=Off) , "xxxxx"表示欲寫入的指定位址 , 其格式一率須填五位數(假若您要改寫第 10 個位址 , 請填入 00010 , 其他以此類推) , 可寫入的位址區間為 00000~99999 ; "value"表示欲改寫的狀態(1=On ; 0=Off) ; 要注意參數有大小寫的限制。

範例:

<http://10.0.0.183/register.dll?writeDO=00010&data=1>

寫入 1 到 eLogger Shared Memory-DO 的第 10 個位址 , 使其狀態為 On 。

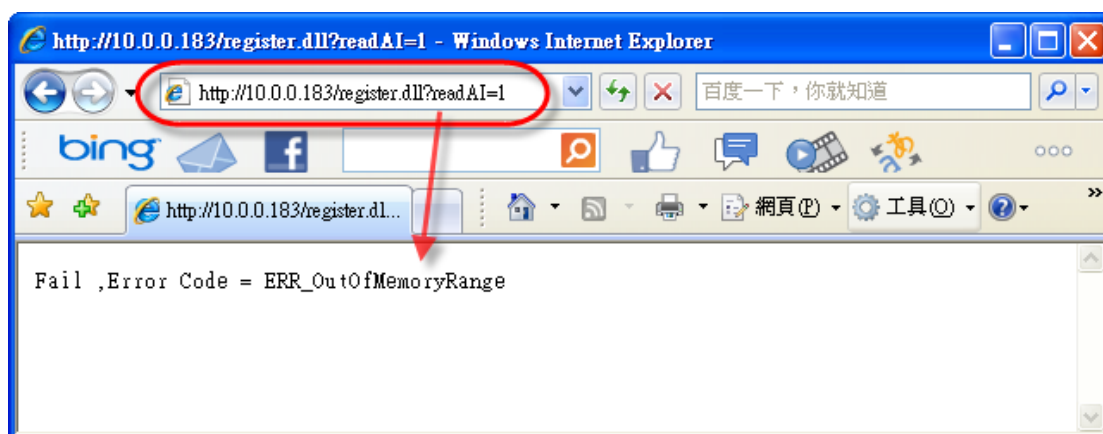
The image shows two screenshots. The top one is a screenshot of the eLogger software interface, specifically the 'Tag Mapping' section. It displays a table of tags with columns for Memory Address, Name, Location, Description, and Note. The tag 'CoilStatus[10]' is highlighted with a red circle, and its location is 'ModbusSerial->COM2_ID1->Coil11'. The bottom screenshot is a Windows Internet Explorer browser window showing the URL 'http://10.0.0.183/register.dll?writeDO=00010&data=1' in the address bar, which is also circled in red. The browser content area shows the number '1'.

Memory Address	Name	Location	Description	Note
CoilStatus[0]	Coil1	ModbusSerial->COM2_ID1->Coil1	COM2_ID1_Address:00001	
CoilStatus[1]	Coil2	ModbusSerial->COM2_ID1->Coil2	COM2_ID1_Address:00002	
CoilStatus[2]	Coil3	ModbusSerial->COM2_ID1->Coil3	COM2_ID1_Address:00003	
CoilStatus[3]	Coil4	ModbusSerial->COM2_ID1->Coil4	COM2_ID1_Address:00004	
CoilStatus[4]	Coil5	ModbusSerial->COM2_ID1->Coil5	COM2_ID1_Address:00005	
CoilStatus[5]	Coil6	ModbusSerial->COM2_ID1->Coil6	COM2_ID1_Address:00006	
CoilStatus[6]	Coil7	ModbusSerial->COM2_ID1->Coil7	COM2_ID1_Address:00007	
CoilStatus[7]	Coil8	ModbusSerial->COM2_ID1->Coil8	COM2_ID1_Address:00008	
CoilStatus[8]	Coil9	ModbusSerial->COM2_ID1->Coil9	COM2_ID1_Address:00009	
CoilStatus[9]	Coil10	ModbusSerial->COM2_ID1->Coil10	COM2_ID1_Address:00010	
CoilStatus[10]	Coil11	ModbusSerial->COM2_ID1->Coil11	COM2_ID1_Address:00011	
CoilStatus[11]	Coil12	ModbusSerial->COM2_ID1->Coil12	COM2_ID1_Address:00012	
CoilStatus[12]	Coil13	ModbusSerial->COM2_ID1->Coil13	COM2_ID1_Address:00013	
CoilStatus[13]	Coil14	ModbusSerial->COM2_ID1->Coil14	COM2_ID1_Address:00014	
CoilStatus[14]	Coil15	ModbusSerial->COM2_ID1->Coil15	COM2_ID1_Address:00015	

2.3. 錯誤碼說明

當使用者輸入錯誤或者是無效的 URL 格式時，register.dll 會回傳錯誤訊息的字串，錯誤訊息範例如下圖所示：

”Fail, Error Code: ERR_OutOfMemoryRange”






以下列一表格說明所有錯誤碼的代表意義

Error Code	Description
ERR_OutOfMemoryRange	輸入位址錯誤，不符格式
ERR_WrongCommand	輸入的參數錯誤(可能拼錯字或者是大小寫不符規定)
ERR_WriteDataNotNumber	寫入 Shared Memory 的數值不是數字
ERR_WriteDataNotInterger	寫入 Shared Memory 的數值不是整數
ERR_WriteDataNotUInterger	寫入 Shared Memory 的數值不是無符號整數
ERR_WriteDataNotLong	寫入 Shared Memory 的數值不是長整數。
ERR_WriteDataNotULong	寫入 Shared Memory 的數值不是無符號長整數
ERR_Unknown	不明的錯誤

3 如何設定讀取 register.dll

eLogger 的 Runtime 執行程式因應不同的作業系統平台而有所分別，其 Web API DLL 也是一樣因作業系統不同而有所區別，所以不同平台之間的 register.dll 不能混用，當您從泓格的官網下載 eLogger 的安裝檔解壓縮後，請複製對應的 Runtime 資料夾複製到您的控制器中，再按照以下的步驟說明來配置 register.dll。

以下表格說明不同 OS 的 PACs 必須搭配正確的 eLogger runtime 檔案與 register.dll 才能正常執行。

OS	eLogger runtime file	register.dll
CE5	For_WinPAC	 register.dll 1.0.2.0 eLogger Web API CE5
CE6	For_XP8000CE6	 register.dll 1.0.2.0 eLogger Web API CE6
XPE	For_XP8000WES	 register.dll 1.0.2.0 eLogger Web API XPE

3.1. Windows CE5 based PACs


Step 1: 請將”For_WinPAC”資料夾複製到 CE5 平台控制器的自己指定資料夾中。

例如：\Micro_SD\For_WinPAC

Step 2: 將\Micro_SD\For_WinPAC\system_disk\icpdas\system 資料夾裡的 SharedMemory.dll 複製到控制器的系統資料夾：

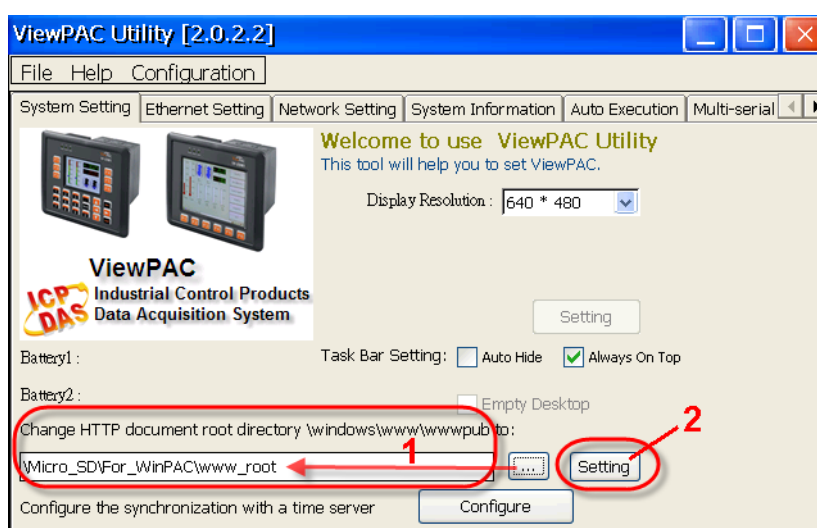
\System_Disk\icpdas\system，這樣才能讓多個程式可以同時去讀寫 eLogger 的 SharedMemory。

Step 3: 開啓控制器上桌面的 ViewPAC Utility，切換到 System Setting 頁籤，點

選  按鈕將控制器的網站根目錄改成

\Micro_SD\For_WinPAC\www_root，然後點選 Setting 按鈕完成設定。

讓使用者可以透過網路操作 register.dll。



3.2. Windows CE6 based PACs

Step 1: 請將”For_XP8000CE6”資料夾複製到 CE6 平台控制器的自己指定資料夾中。

例如：System_Disk\For_XP8000CE6

Step 2: 將\System_Disk\For_XP8000CE6\system_disk\icpdas\system 資料夾裡的”SharedMemory_CE6.dll”複製到控制器的系統資料夾：

\System_Disk\ICPDAS\SYSTEM，這樣才能讓多個程式可以同時去讀寫 eLogger 的 SharedMemory。

Step 3: 將\System_Disk\For_XP8000CE6\www_root 資料夾裡的”register.dll”複製到控制器的網站根目錄(註)中：\System_Disk\www，這樣才能讓使用者可以透過網路操作 register.dll。

註:請開啓桌面上的 XPAC Utility，切換到 Network 頁籤，確認網站根目錄是否為\System_Disk\www，若不是請點選...按鈕作變更，然後再按”Apply”按鈕完成設定。

3.3. Windows Embedded Standard 2009 PACs

Step 1: 請將”For_XP8000WES”資料夾複製到 XPE 平台控制器的自己指定資料夾中。

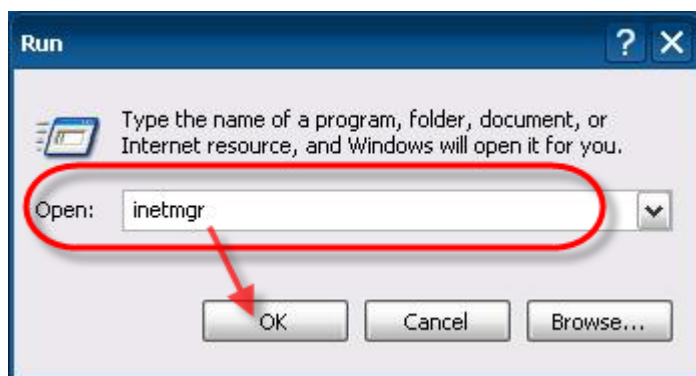
例如：T:\For_XP8000WES

Step 2: 將 T:\For_XP8000WES \windows 資料夾裡的”SharedMemory_XP.dll”複製到控制器的系統資料夾：C:\windows，這樣才能讓多個程式可以同時去讀寫 eLogger 的 SharedMemory。

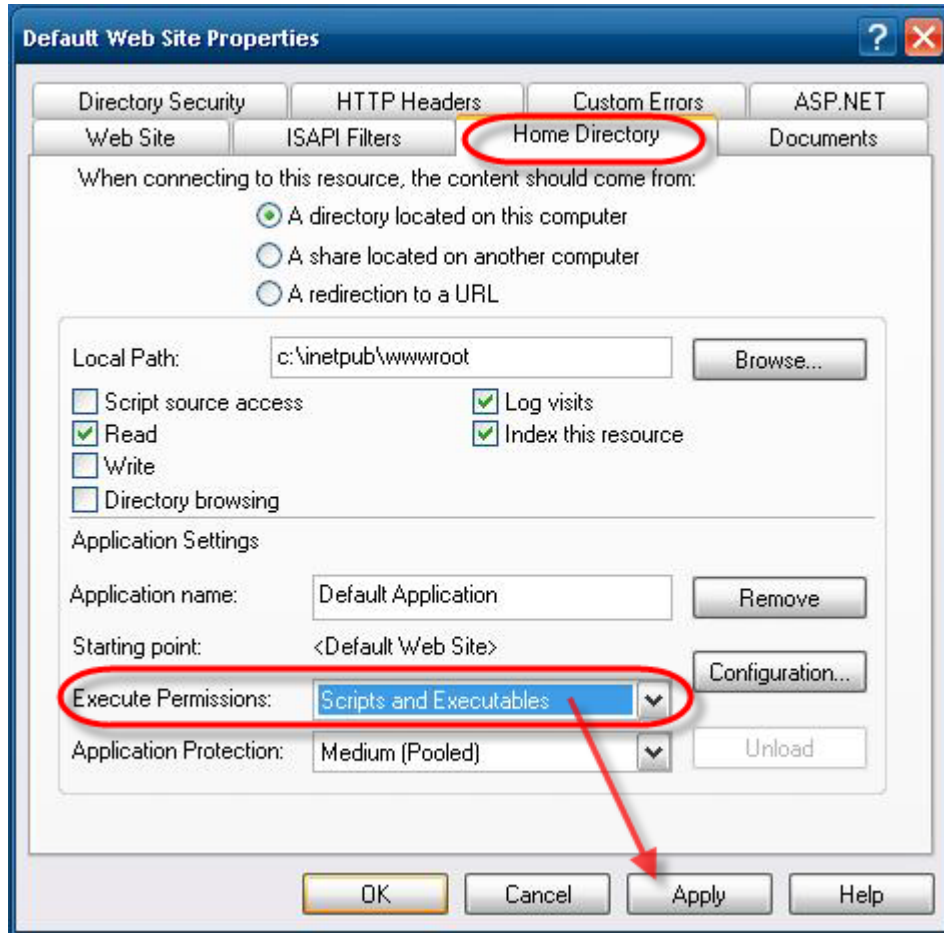
Step 3: 將 T:\For_XP8000WES \www_root 資料夾裡的”register.dll”複製到控制器的網站根目錄中：C:\inetpub\wwwroot。

Step 4: 啓用 Web 目錄的”Scripts and Executes”權限，讓使用者可以透過網路操作 register.dll。

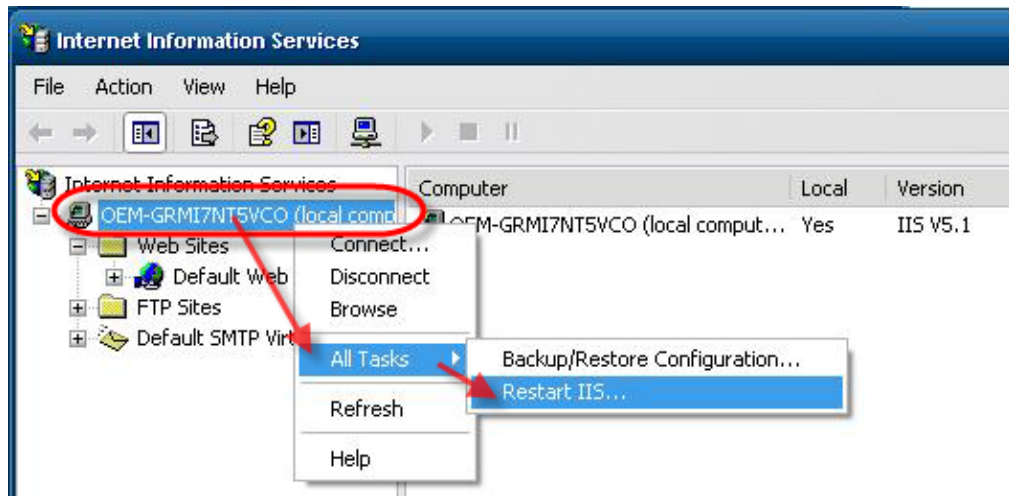
4.1. Start → Run → 在打開的系統執行對話框中輸入”inetmgr”開啓 IIS 管理員。



4.2. 在左邊視窗的”Default Web Site”上按右鍵選擇”Properties” → 切換到”Home Directory”，將 Execute Permissions 更改為”Scripts and Executables” → 點選 Apply 按鈕 → 點選 OK 按鈕。



4.3. 回到 IIS 設定視窗中，在主電腦名稱上按右鍵選擇重新啟動 IIS 完成設定。



4 如何開發 Android App 程式

<相關安裝步驟以及教學說明>

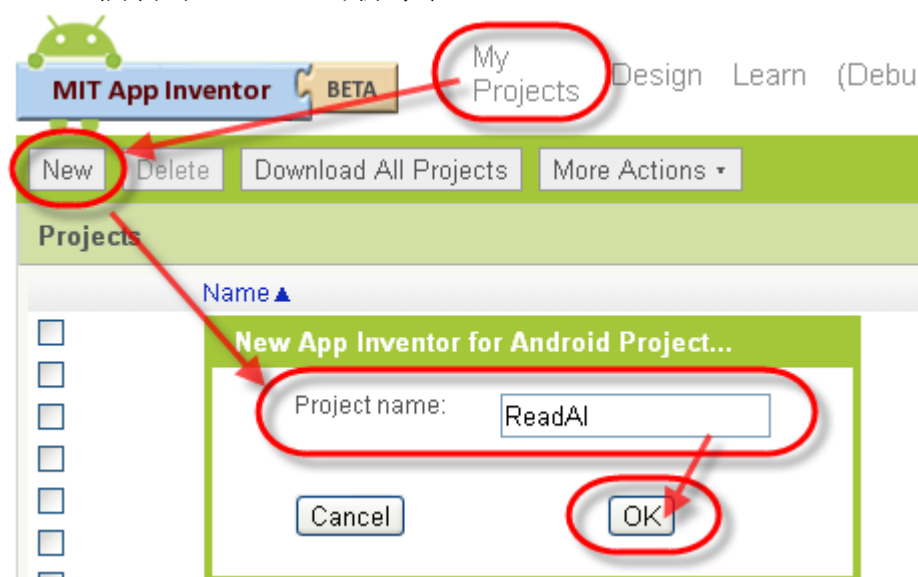
[MIT App Inventor官網](#)

[App Inventor中文學習網](#)

範例 1：讀取 eLogger Shared Memory AI 第 1 個位址的資料

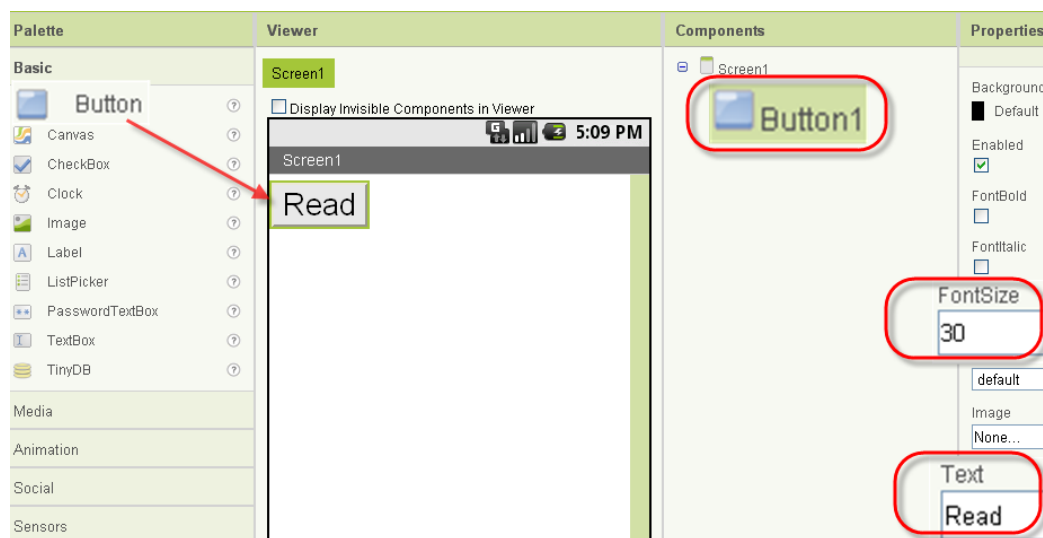
Step 1: 建立新專案

登入Gmail → 開啓[App Inventor設計網頁](#) → 到My Projects頁面建立一個名為"ReadAI"的新專案。

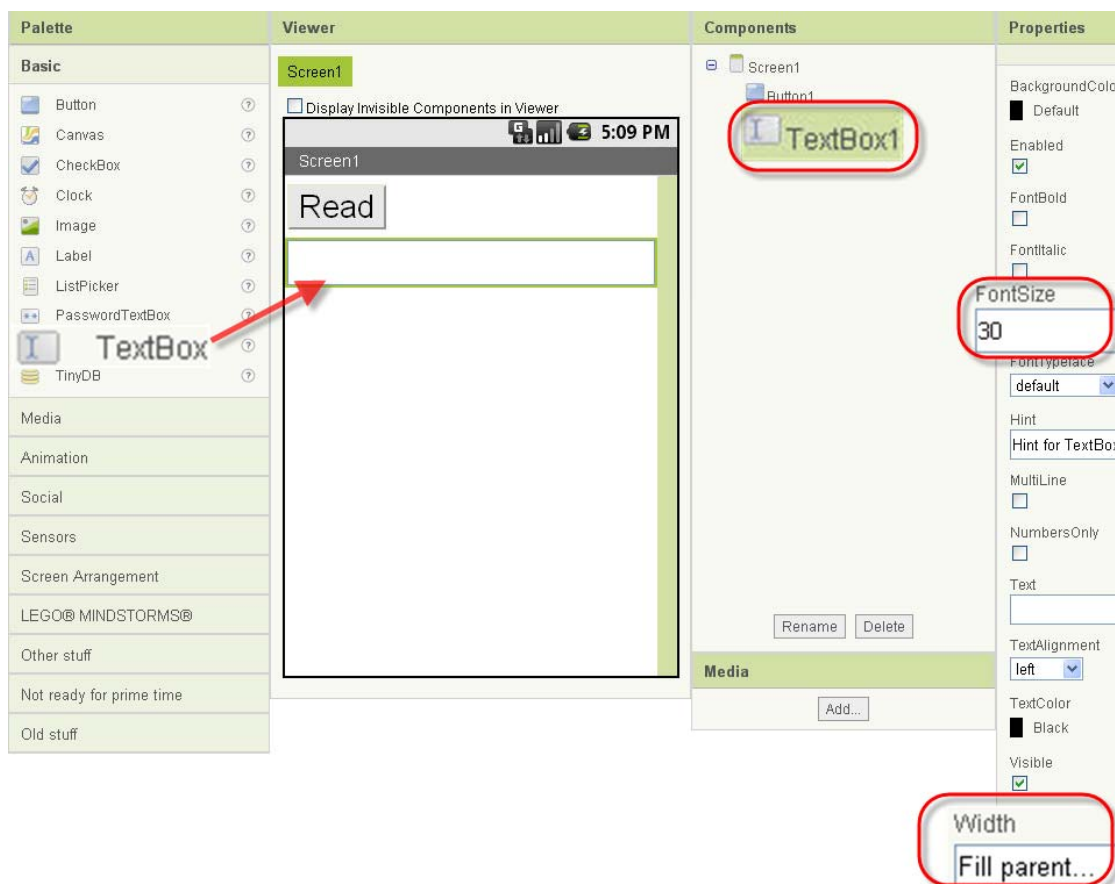


Step 2: App Inventor Designer 設計外觀

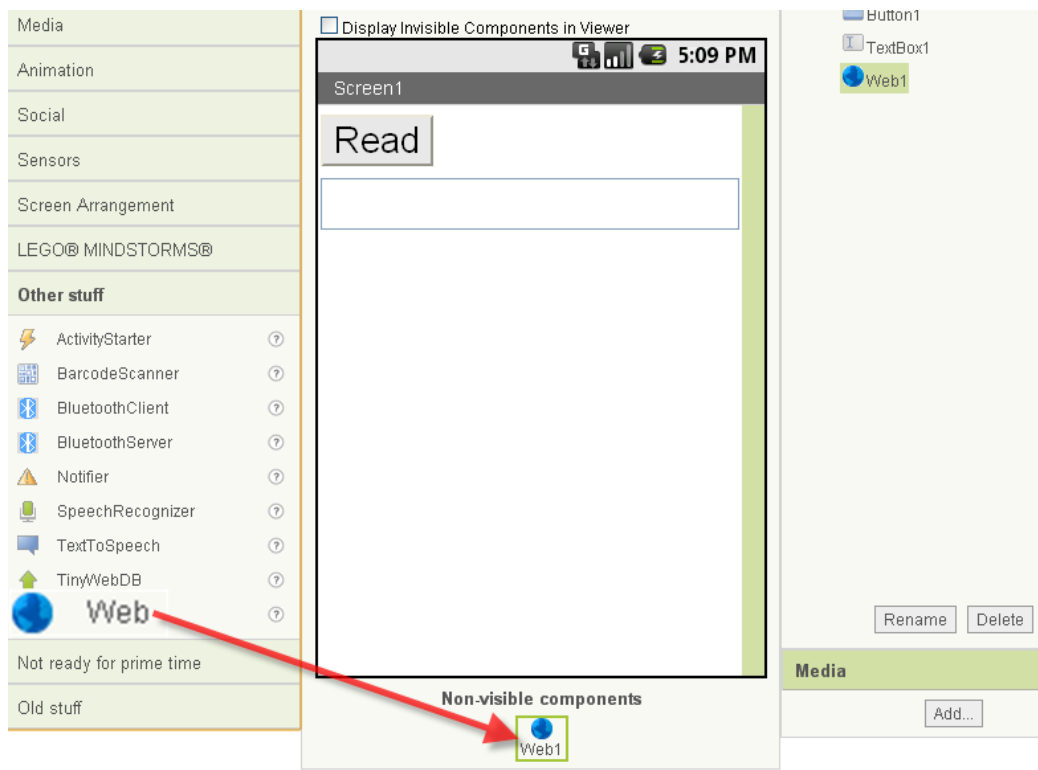
2.1 由左側的 Basic 元件區新增一個按鈕 Button，字體大小設為 30，文字設為“Read”，其餘不變。



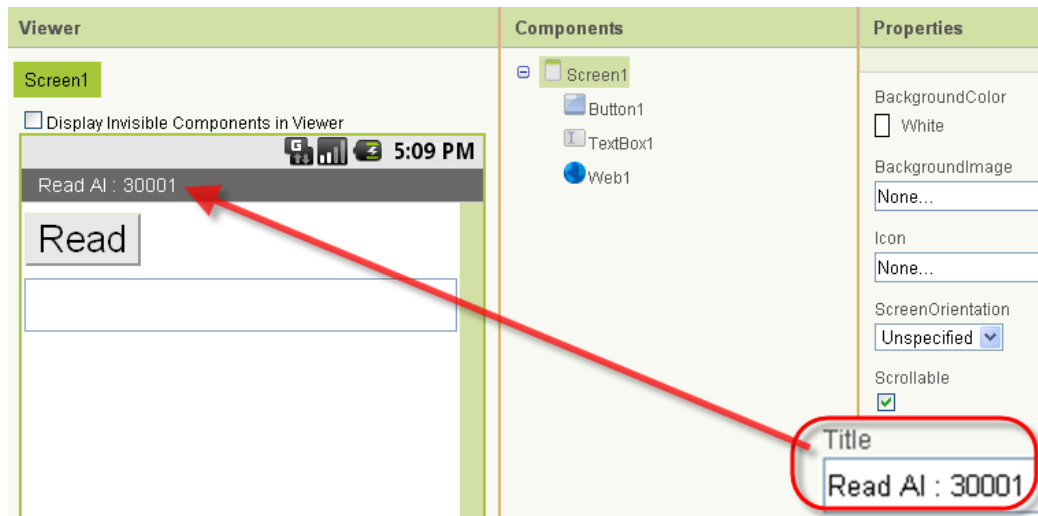
2.2 由左側的 Basic 元件區新增一個文字方塊 TextBox，字體大小設為 30，寬度設為“Fill Parent”，其餘不變。



2.3 由左側的 Other stuff 元件區新增一個 Web 組件。



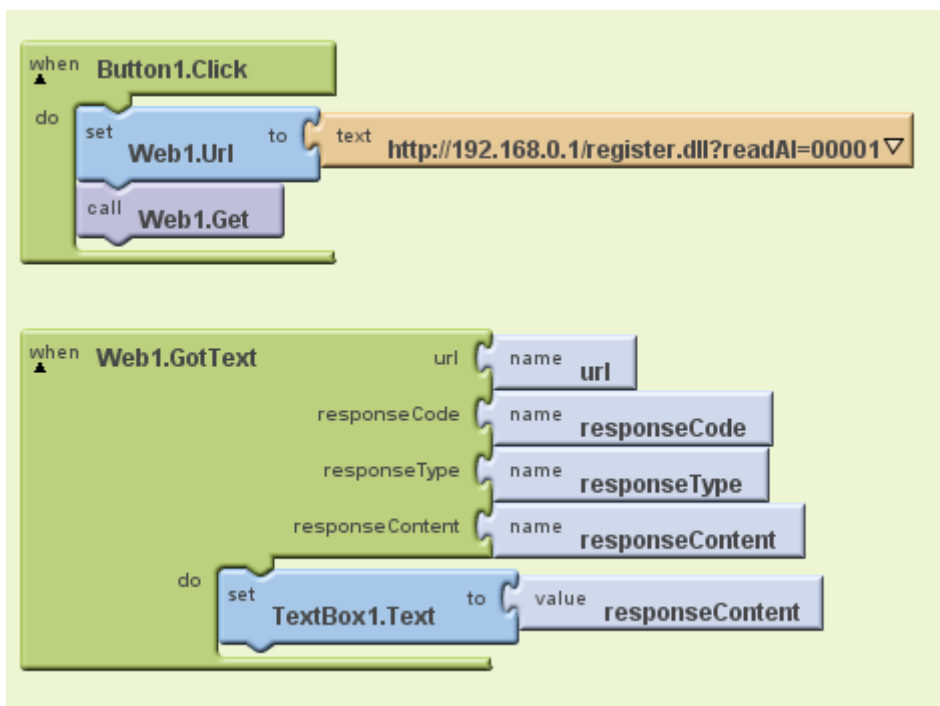
2.4 將 Screen1 的 Title 設為" Read AI :30001"，到本步驟頁面設計完成。



Step 3: App Inventor Blocks Editor 拼湊程式邏輯

3.1 點選"Open the Blocks Editor"進入 Blocks Editor，會先下載一個 ApplinventorForAndroidCodeblocks.jnlp 檔，點選後即可開啓 Java 並進入 Blocks Editor。

3.2 在 Blocks Editor 裡，依序新增指令，並組合如下圖。



3.2.1 指令說明：點擊 Button1 的按鈕時，送出讀取 eLogger Shared Memory AI 第 1 個位址資料的 web 請求。



1. My Blocks → Button1 → when Button1.Click 事件

代表當 Button1 被按下時，執行所包含的動作。

2. My Blocks → Web1 → set Web1.Url 事件

設定 Web1 元件的 URL 為本指令右方插槽中的參數。

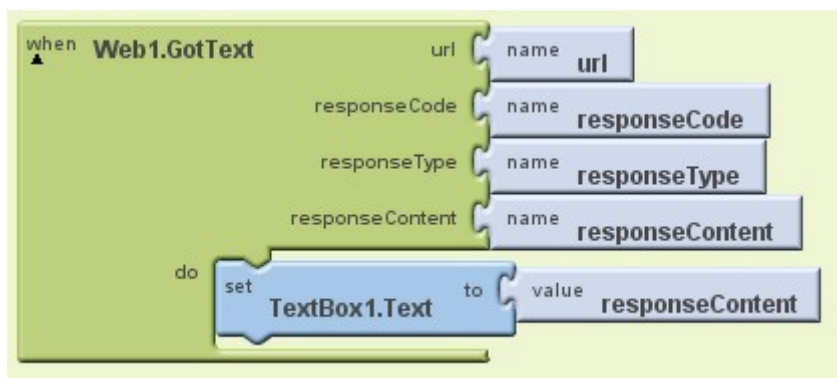
3. Built-In → Text → text 參數

將其內容"text"改為 http://192.168.0.1/register.dll?readAI=00001 (IP 位址需改為欲連接的 PAC 控制器的 IP 位址)。

4. My Blocks → Web1 → Call Web1.Get 指令

送出 web 的請求。

3.2.2 指令說明：當 web 傳送的請求回傳時會觸發"Web1.GotText"的事件，接收資料存放在 responseContent 的參數裡，並將 TextBox1 的文字內容改為 responseContent 參數的內容。



1. My Blocks → Web1 → when Web1.GotText 事件

代表當事件觸發時，執行所包含的動作。

2. My Blocks → TextBox1 → set TextBox1.Text to 指令

代表設定 TextBox1 的文字(Text)內容為本指令右方插槽中的參數。

3. My Blocks → My Definitions → responseContent 參數

內容為從 web 接收的資料。

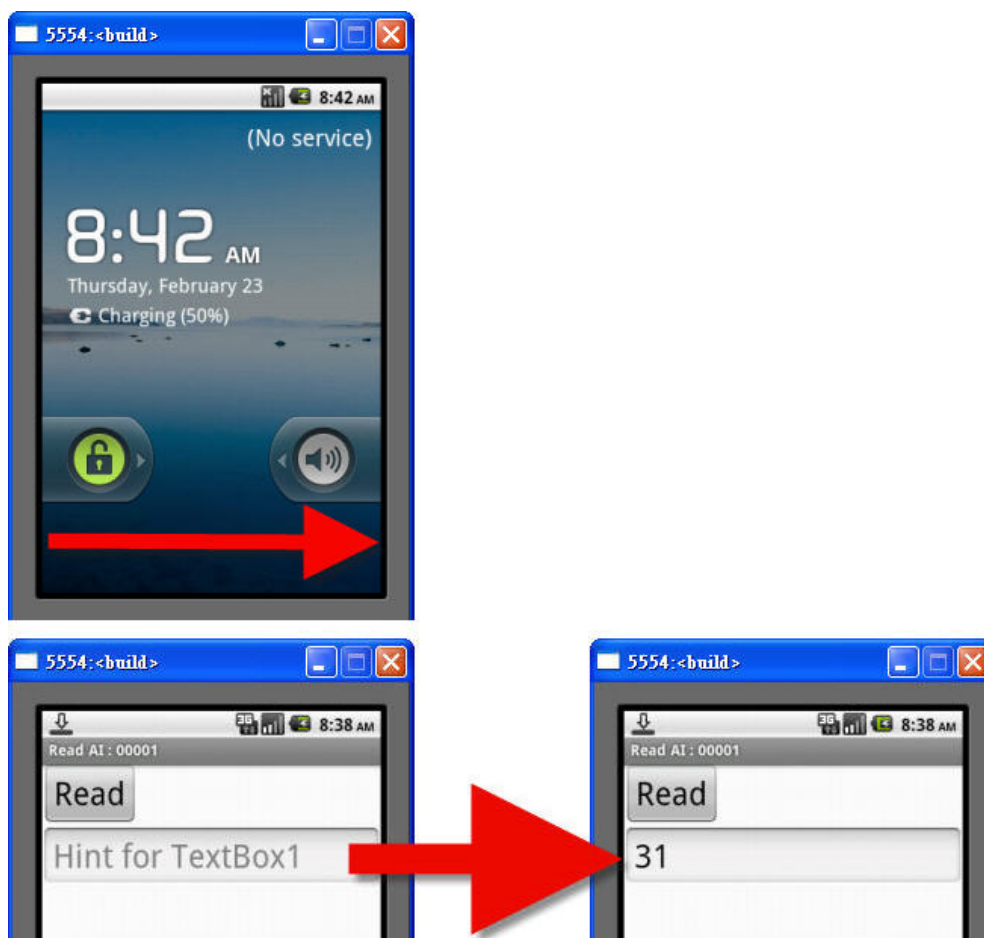
Step 4: 開始執行程式

4.1 點選 Blocks Editor 右上方的"New Emulator"來啟動模擬器(如您使用實體 Android 裝置，本段可跳過)。

4.2 點選 Blocks Editor 右上方的 "Connect to Device..."，接著點選 "Emulator 5554"或您的裝置將程式安裝到指定位置，安裝時電話圖樣會變成黃色並閃動。



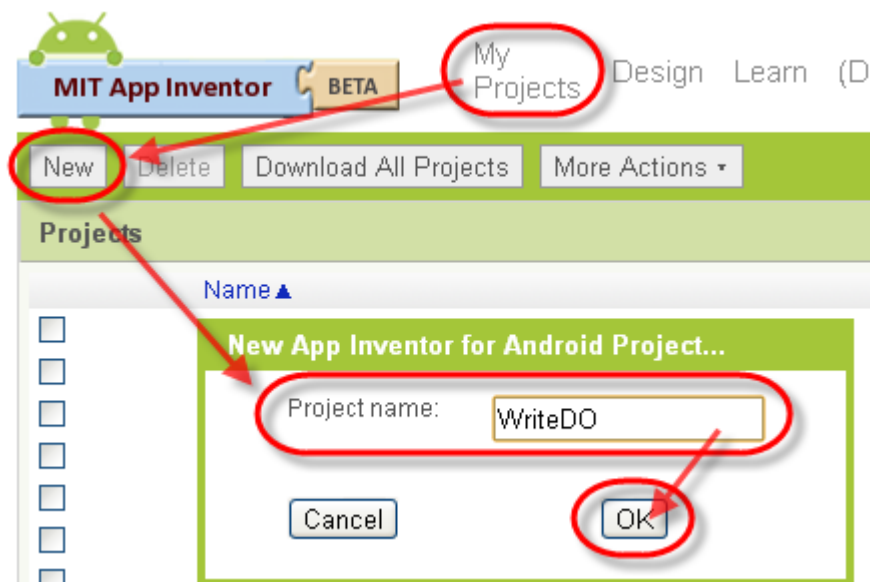
4.3 請將模擬器解鎖，靜待一會時間即可看到程式，點選"Read"按鈕即可讀取資料。



範例 2：寫入資料到 eLogger Shared Memory DO 第 0 個位址

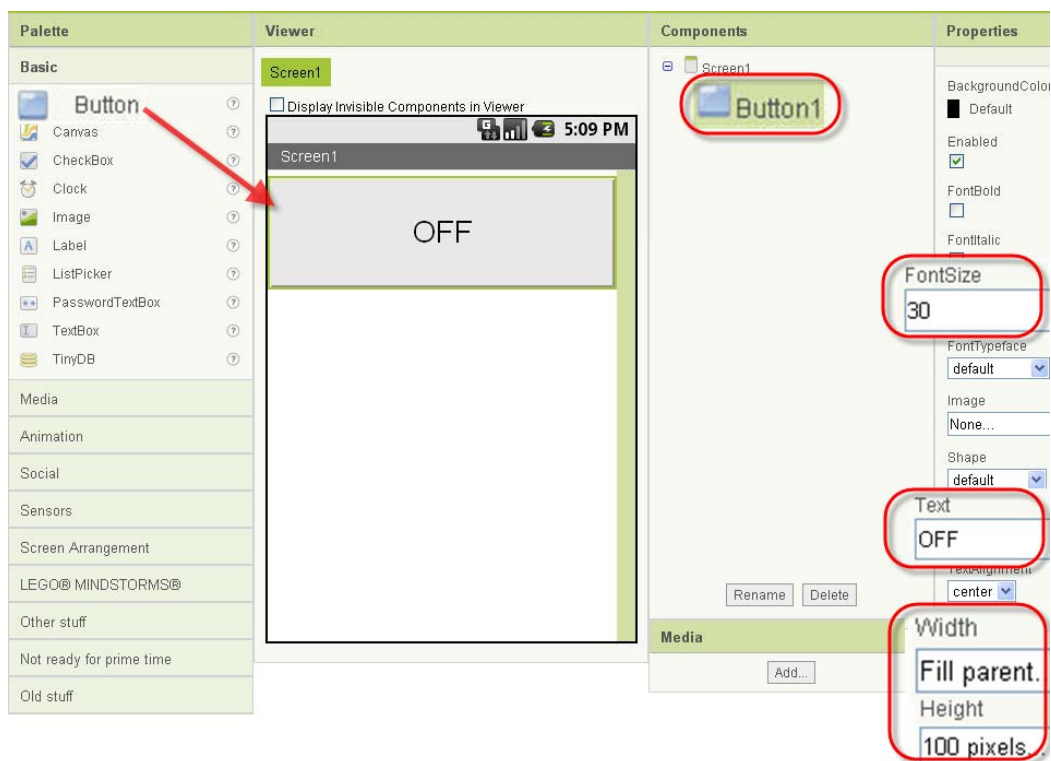
Step 1: 建立新專案

登入Gmail → 開啓[App Inventor設計網頁](#) → 到My Projects頁面建立一個名為"WriteDO"的新專案。

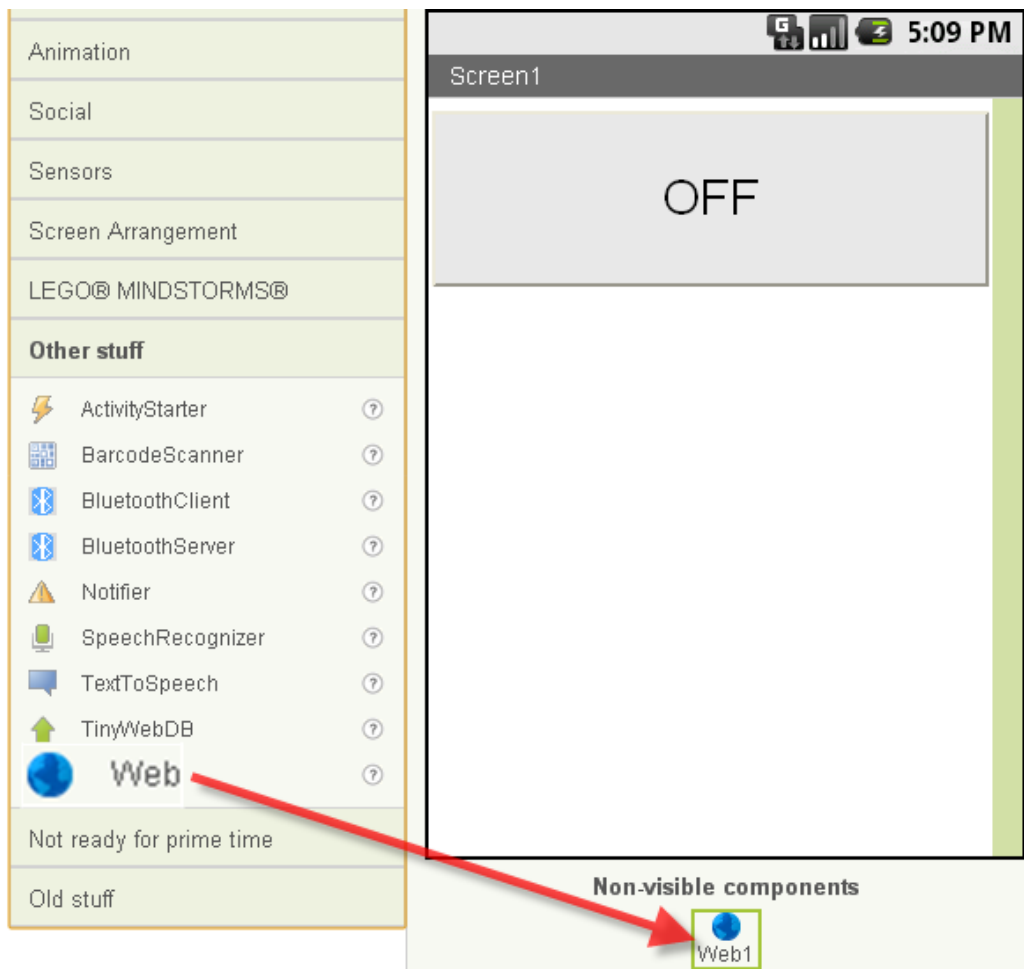


Step 2: App Inventor Designer 設計外觀

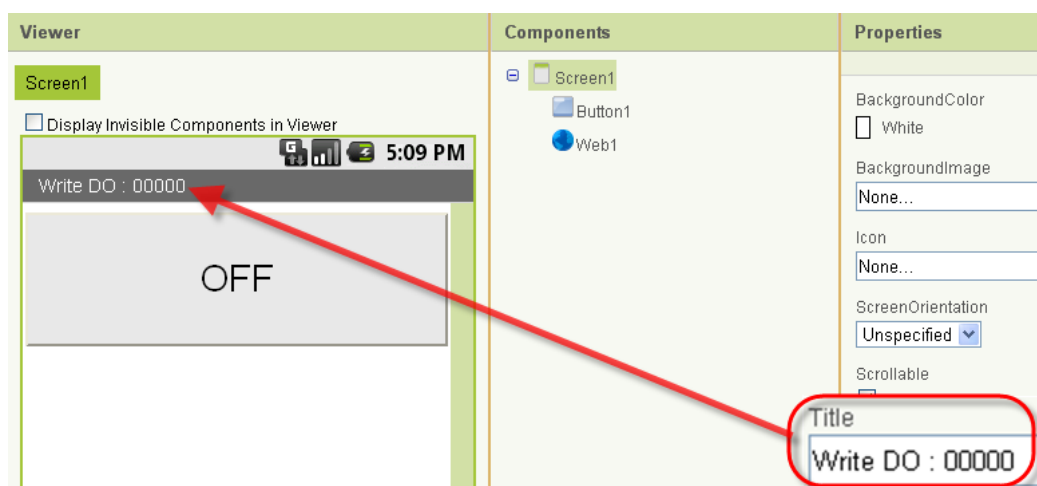
2.1 由左側的 Basic 元件區新增一個按鈕 Button，字體大小設為 30，文字設為"OFF"，寬度設為"Fill parent"，高度設為"100 pixels"，其餘不變。



2.2 由左側的 Other stuff 元件區新增一個 Web 組件。



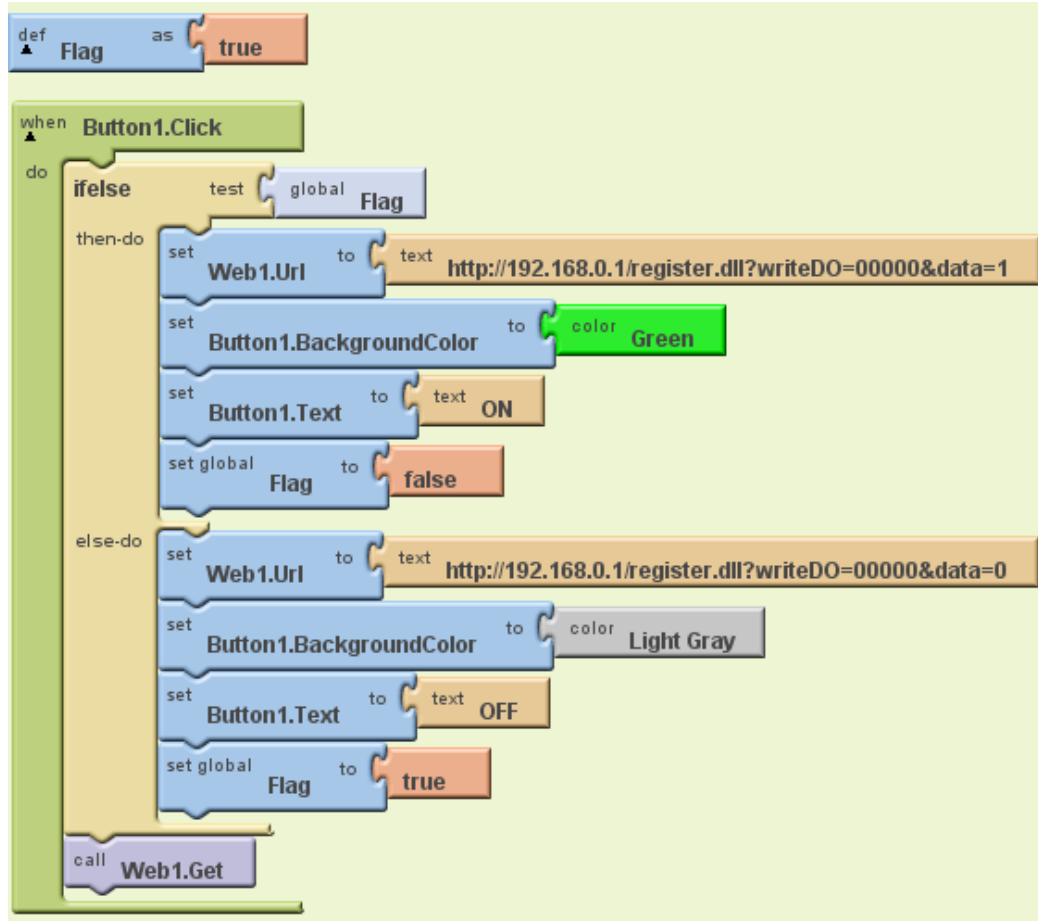
2.3 將 Screen1 的 Title 設為" Write DO : 00000"，到本步驟頁面設計完成。



Step 3: App Inventor Blocks Editor 拼湊程式邏輯

3.1 點選"Open the Blocks Editor"進入 Blocks Editor，會先下載一個 ApplinventorForAndroidCodeblocks.jnlp 檔，點選後即可開啓 Java 並進入 Blocks Editor。

3.2 在 Blocks Editor 裡，依序新增指令，並組合如下圖。



3.2.1 指令說明：自行定義一個名為"Flag"的全域變數，並預設為 true。



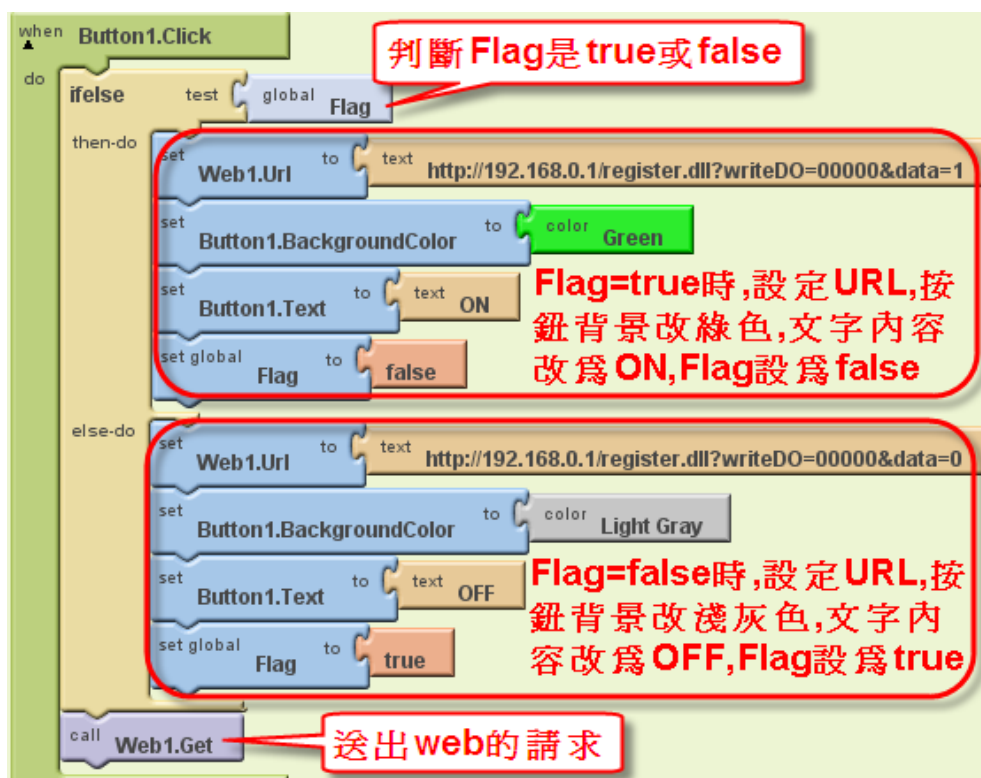
1. Built-In → Definition → def variable as 事件

代表定義一個全域變數，內容為右方插槽中的參數，將全域變數名稱"Variable"改為 Flag。

2. Built-In → Logic → true 參數

代表布林值"True"。

3.2.2 指令說明如下圖：



1. My Blocks → Button1 → when Button1.Click 事件

代表當 Button1 被按下時，執行所包含的動作。

2. Built-In → Control → ifelse 指令

當 test 右方插槽參數狀態為 true 時，執行 then-do 所包含的動作；否則當 test 右方插槽參數狀態為 false 時，執行 else-do 所包含的動作。

3. My Blocks → My Definitions → global Flag 參數

代表全域變數 Flag 的內容。

4. My Blocks → Web1 → set Web1.Url 事件

設定 Web1 元件的 URL 為本指令右方插槽中的參數。

5. Built-in → Text → text 參數

將"text"內容改爲<http://192.168.0.1/register.dll?writeDO=00000&data=1>，即寫入資料 1 到 eLogger Shared Memory DO 的第 0 個位址。

6. My Blocks → Button1 → set Button1.BackgroundColor to 指令

代表設定 Button1 的背景顏色為本指令右方插槽中的參數。

7. Built-In → Colors → color Green 參數

代表參數內容為綠色。

8. My Blocks → Button1 --> set Button1.Text to 指令

代表設定 Button1 的文字內容為本指令右方插槽中的參數。

9. My Blocks → Web1 → Call Web1.Get 指令

送出 web 的請求。

Step 4: 開始執行程式

4.1 點選 Blocks Editor 右上方的"New Emulator"來啟動模擬器(如您使用實體 Android 裝置，本段可跳過)。

4.2 點選 Blocks Editor 右上方的 "Connect to Device..."，接著點選 "Emulator 5554"或您的裝置將程式安裝到指定位置，安裝時電話圖樣會變成黃色並閃動。



4.3 請將模擬器解鎖，靜待一會時間即可看到程式，切換按鈕送出資料改寫 eLogger Shared Memory DO 的第 0 個位址。



附錄 A. 修訂紀錄

版本	修訂日期	修訂內容
1.0.1	2012/03/06	第一次 release。
1.0.2	2012/10/29	加入新的參數功能說明。