uClinux 作業系統下纂寫 CGI 網頁程序(uClinux-7)

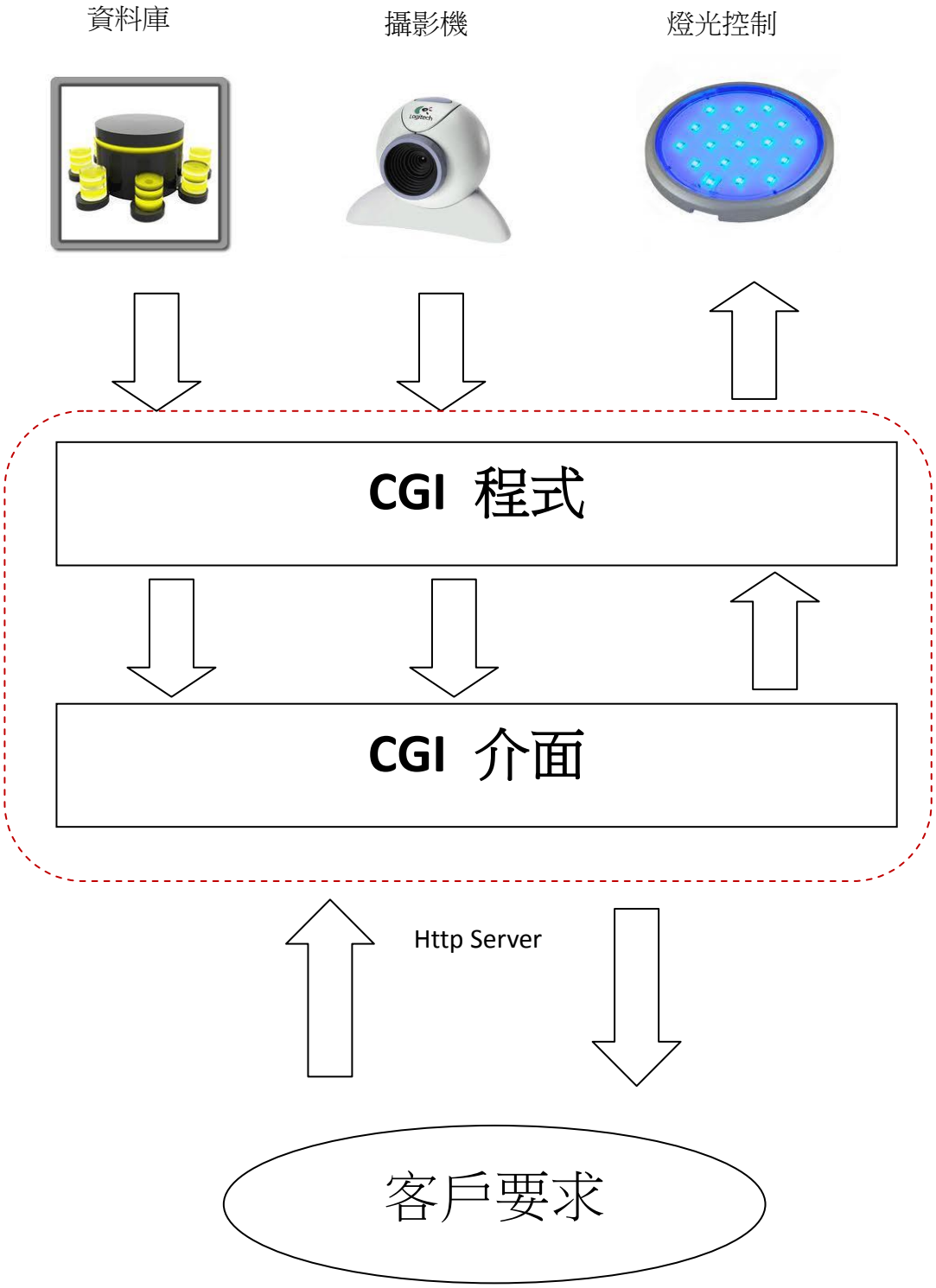
使用者透過瀏覽器觀賞網頁時，瀏覽器會和網路主機上的 Http server 建立一個連線，然後 Http server 會到事先做好的 HTML 檔案中，找出使用者要看的網頁， 回傳給瀏覽器，讓使用者觀賞；由於 HTML 檔案是靜態的，無法顯示即時資料，更由於 HTML 格式較為簡單，無法達成一般複雜程式的需求，因此就有人想到要**透過程式來產生 HTML 內容**，這就是 CGI 的來由。

CGI 是什麼呢? 它的全名是 Common Gateway Interface,，規定了 Http server 和 CGI 程式之間傳遞參數和結果的方法；在瀏覽 CGI 網頁時，同樣地瀏覽器會和網路主機上的 Http server 建立一個連線，但是接著 Http server 不是去抓檔案，而是去啟動一個 CGI 程式，然後將 CGI 程式所產生的內容當成 HTML 傳回給使用者瀏覽器。

舉證券交易的網路運作為例，使用者會上谷哥(Google)上查詢某一檔股票的即時或是歷史紀錄，網頁的顯示是在 Http server 下的 HTML 格式，但是資料的取得勢必是從資料庫讀取，Http server 和資料庫中間的界面即可稱之為 CGI，它可以取得 HTTP 使用者送來的查詢指令，轉換成現存資料庫可以了解的指令， 用以取得資料後，再將資料轉換為 HTML 格式，最後經由 HTTP server 傳回給使用者，顯示在使用者的瀏覽器上。

根據上述的需求，CGI 程式必須可以以任何語言來纂寫，因為它必須可讀取系統內的任意資料，包含了硬體的狀態或是控制，在 uClinux 作業系統下，最方便的語言就是 C 語言囉。

一、CGI 介面運作架構



上圖顯示當客戶透過網路要求 Http Server，顯示主機之資料庫內容或是攝影機即時資料；亦控制主機之各項硬體設備時，Http Server 將會透過 CGI 介面啟動所指定的 CGI 程式，來讀取或是控制所指定之硬體，並將結果透過 CGI 介面回傳。

二、CGI 與 Form 標籤

記得 HTML 語法中有個<FORM>標籤嗎? 這就是 CGI 程式主要應用的地方，舉例說明如下：

```
<FORM ACTION= ”/cgi-bin/ESDLED_cgi” METHOD=GET>
        <input name ="LED1" type="checkbox"/> LED1
        <input name="LED2" type="checkbox" /> LED2
        <input name="LED3" type="checkbox" /> LED3
        <input name="LED4" type="checkbox" /> LED4
        <input name="LED5" type="checkbox" /> LED5
        <input name="LED6" type="checkbox" /> LED6
        <input name="LED7" type="checkbox" /> LED7
        <input name="LED8" type="checkbox" /> LED8
        <input name="LEDCONTROL" type="submit" value="確定" /></p>
</FORM>
```

- FORM 標籤定義 checkbox 之名稱分別為 LED1 – LED8，當按下 submit 按鈕後，瀏覽器將會執行 ACTION 內所定義的 CGI 程式(上例中為 ESDLED_cgi)，並將選定的結果傳送到伺服器上，若 HTTP 發現這是一個 CGI 的要求， 就會藉由 CGI 去呼叫指定的程式，並建立起互相溝通的管道。
- FORM 標籤所定義的 METHOD=GET，為 CGI 介面傳遞的方式，常用的方法為 POST 及 GET。

三、CGI 提供的溝通管道

<FORM>標籤內的 method 屬性，有二種值分別為 get 和 post，別代表了 CGI 的二種溝通管道：

- **method=get** 藉由環境變數來傳遞資料，瀏覽器會將你填入<FORM>裡的資料附加在 action 屬性所指定的 CGI 程式名稱後面， 並以"?"隔開，當 HTTP 伺服器收到這個要求後會將"?"後面的字串存放在 QUERY_STRING 這個環境變數中， 於是 CGI 程式就可以透過這個環境變數取得<FORM>裡面的資料了，uClinux 作業系統下的 Boa 應用程式提供此種溝通管道；此種方法的最大缺點為，環境變數的大小是有一定的限制的，當需要傳送大量資料時，儲存環境變數的空間可能會不足，造成資料接收不完全，甚至無法執行 CGI 程式。
- **method=post** 是利用 I/O 重新導向的技巧，讓 CGI 程式可以藉由 STDIN 和 STDOUT 直接跟瀏覽器溝通。 當我們指定用這種方法傳遞<FORM>裡面的資料時，HTTP 伺服器收到資料後會先放在一塊輸入緩衝區中，並且將資料的

大小記錄在 CONTENT_LENGTH 這個環境變數， 然後呼叫 CGI 程式並將 CGI 程式的 STDIN 指向這塊緩衝區，於是我們就可以很順利的透過 STDIN 和環境變數 CONTENT_LENGTH 得到所有的資料，再沒有資料大小的限制了；這個方法在 uClinux 作業系統下筆者測試後發覺有些問題，無法正確使用。

以下舉筆者在 ESD44B0_B 目標板上所纂寫的 LED 網路控制 CGI 程式 Get 方法為例：

如上圖所示，當使用者勾選 LED1 名稱後按下確定鍵，瀏覽器執行 ESDLED_cgi 程式，並透過 CGI 介面傳遞 QUERY_STRING 環境變數內容為?’LED1=on&……’，當 CGI 應用程式 ESDLED_cgi 接收此訊息後，分析所接收的資訊，依照使用者所要求執行對應之程式碼。

四、CGI 介面的環境變數

在 uClinux 作業系統下的 CGI 程式設計皆透過 Get 方法利用環境變數來傳遞訊息，以下列出 Get 方法使用的環境變數：

環境變數	內容
AUTH_TYPE	存取認證型態。
CONTENT_LENGTH	經由標準輸入傳遞給 CGI 程式的資料長度，以 bytes 或字元數來計算。
CONTENT_TYPE	query 資料的 MIME 型態。
GATEWAY_INTERFACE	伺服器的 CGI 版本編號。
HTTP_(string)	client 端的檔頭資料，由各瀏覽器自訂。
PATH_INFO	傳遞給 cgi 程式的額外路徑資訊。
QUERY_STRING	傳遞給 CGI 程式的 query 資訊，也就是用"?"隔開，添加在 URL 後面的字串。
REMOTE_ADDR	client 端(發出 request 那一端)的主機名稱。

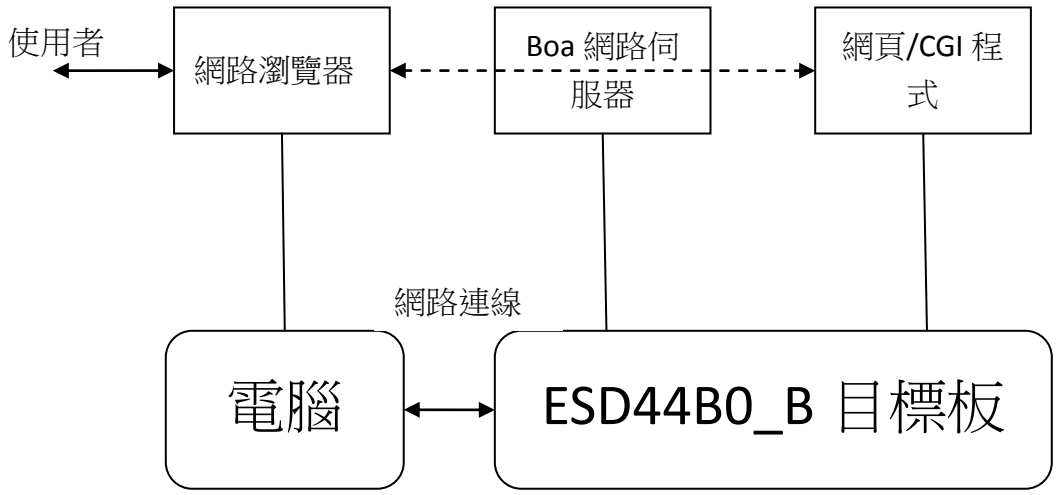
REMOTE_HOST	client 端的 IP 位址。
REMOTE_USER	client 端送出來的使用者名稱。
REMOTE_METHOD	client 端發出 request 的方法。
SCRIPT_NAME	CGI 程式所在的虛擬路徑。
SERVER_NAME	伺服器的名稱或 IP 位址。
SERVER_PORT	收到要求埠之編號。
SERVER_PROTOCOL	所使用的通訊協定和版本編號。
SERVER_SOFTWARE	伺服器程式的名稱和版本。

五、CGI 程式的輸出

CGI 程式在處理完資料後，再將要傳給使用者看的網頁內容往標準輸出 (就是螢幕啦)送。這些送往螢幕的資料會被 Http 伺服器所攔截，並將這些內容送往使用者的瀏覽器，接著使用者便看到畫面了，uClinux 作業系統下使用 C 纂寫的 CGI 程式，就是透過如 printf 函式的呼叫來完成，除此之外需注意 CGI 程式產生的結果，除了原本的 HTML 內容外，要先送出 HTTP Header Content-type:text/html 和一行空白，以 C 語言為例即為 printf("Content-type: text/html\n\n<HTML><HEAD><TITLE>%s</TITLE></HEAD>",title);

六、網路伺服器下執行 CGI 程式

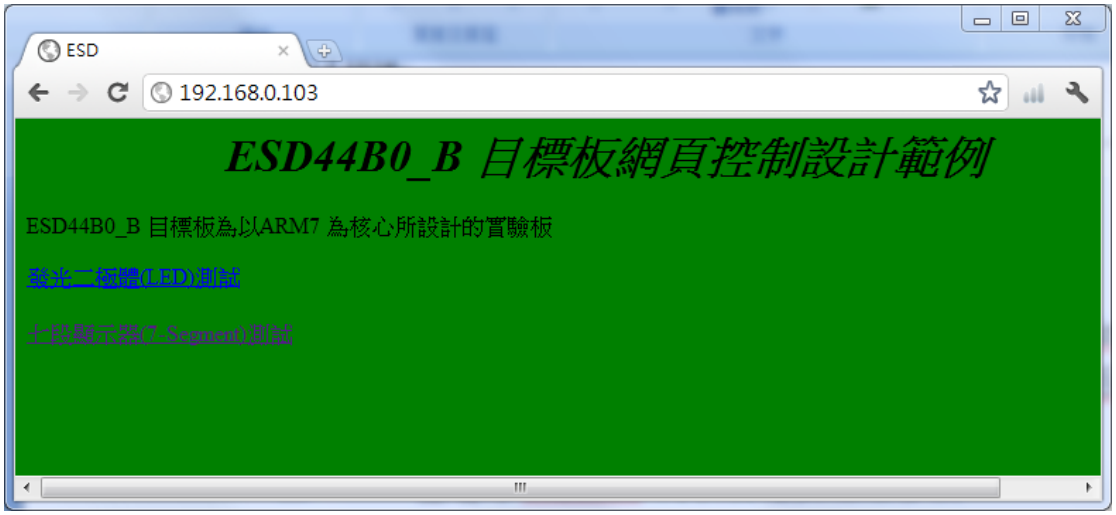
ESDLED_cgi 為使用 CGI 介面而纂寫，配合在 ESD44B0_B 目標板執行 Boa 伺服器服務，而達成驅動發光二極體的應用程式，其系統架構圖如下所示：



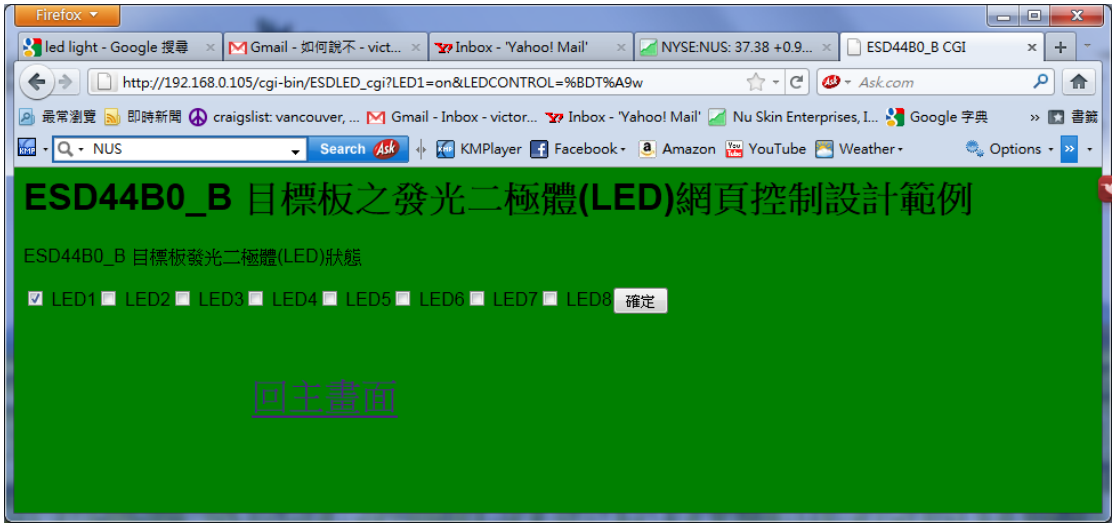
使用者透過網路瀏覽器及 Boa 網路伺服器，來執行已存在 ESD44B0_B 目標板的網頁或是 CGI 程式，ESDLED_cgi 程式被 index.html 呼叫，它可分別控制 ESD44B0_B

上的 8 組發光二極體。

- Boa 應用程式的設定：Boa 網路伺服器可在 uClinux 的’ make xconfig’ 內點選被啟動，詳情請參閱’ uClinux 下的網路伺服器’ 一篇；節錄啟動 Boa 應用程式後，所需檔案放置的位置下：
  - index.html 將被預設放置於’/home/httpd/’目錄下，編譯核心時被放置於 user/boa/src/目錄下，編譯完成後自動被放置於唯獨檔案系統之 /home/httpd/目錄下，當使用者透過瀏覽器使用”http://ESD44B0_B 所在的 IP 位址” ，訪問 Boa 網頁伺服器，index.html 將會顯示於使用者的瀏覽器上，如下圖所示：



  - CGI 應用程式將被放於’/home/httpd/cgi-bin’子目錄下，系統建置者須將所欲執行的 CGI 程式放置於此子目錄下，ESDLED_cgi 在系統建置時就被放入唯讀記憶體內，當點選首頁之’ 發光二極體(LED)測試’ 選項時，即呼叫 ESDLED_cgi 應用程式：



七、ESDLED_cgi 程式的編譯

在 user 子目錄下建立 ESDLED_cgi 次目錄，此專案目錄下的檔案及功能說明如下：

檔案名稱	功能說明
cgi.c	主程式
cgivars.c	處理 Get 或 Post 方法所傳遞之環境變數
htmllib.c	輸出 html 格式之檔頭等訊息
template.c	ESDLED_cgi 網頁之頁面輸出
Makefile	製作檔
index.html	首頁
ESDLED_cgi	CGI 應用程式

- 複製應用程式於指定目錄：Makefiles 內製作唯讀檔案系統的內容如下：

```
romfs:
    $(ROMFSINST) /home/httpd/cgi-bin/ESDLED_cgi
```

- 將 ESDLED_cgi 應用程式加入編譯行列：在下列檔案中分別新增內容

檔名及路徑	行號	內容
uClinux-44b0/user/Makefile	47	dir_$(CONFIG_USER_CGI_ESDLED) += ESDLED_cgi
uClinux-44b0/config/config.tk	4618	bool $w.config.f 10 2 "ESD LED cgi" CONFIG_USER_CGI_ESDLED
	5475	set CONFIG_USER_CGI_ESDLED 0
	6741	global CONFIG_USER_CGI_ESDLED
	6742	write_tristate $cfg $autocfg CONFIG_USER_CGI_ESDLED $CONFIG_USER_CGI_ESDLED [list $notmod] 2
uClinux-44b0/config/configure.help	57-59	CONFIG_USER_CGI_ESDLED A ESD LED CGI demo program Approx. binary size: 10k
uClinux-44b0/vendors/ESD/44b0/ config.vendor	378	CONFIG_USER_CGI_ESDLED = y

- 編譯映像檔：執行 make 以編譯映像檔，完成後將映像檔燒錄至 ESD44B0_B 目標板內。

八、設計 ESDLED_cgi 程式

CGI 介面為在使用者網路瀏覽器，及網路伺服器間資料傳遞的協定，ESDLED_cgi 程式為以 C 語言纂寫，透過網路瀏覽器，控制 ESD44B0_B 目標板的 CGI 程式，此專案由首頁呼叫 ESDLED_cgi 程式。

首頁：


```
<HTML>
<HEAD>
<TITLE>ESD</TITLE>
</HEAD>
<BODY bgcolor="HotTrack" style="background-color: #008000; width: auto; height: auto;
top: auto; right: auto; bottom: auto; left: auto;">
<H1 align="center"
style="font-size: xx-large; font-weight: bolder; font-style: italic; width:
820px;">ESD44B0_B 目標板網頁控制設計範例</H1>
    <p style="width: 820px"> ESD44B0_B 目標板為以ARM7 為核心所設計的實驗板
    </p>
    <a href="./cgi-bin/ESDLED_cgi" title="">發光二極體(LED)測試</a><br
/><br />
    <a href="" title="">七段顯示器(7-Segment)測試</a><br /><br />
</BODY>
</HTML>
```

選擇發光二極體(LED)測試選項後呼叫 ESDLED_cgi LED 控制程式。

ESDLED_cgi CGI 程式：進入 ESDLED_cgi 主程式利用 getRequestMethod 函式呼叫，判斷 FORM 格式利用 Get 或是 Post 方法來傳遞參數，再呼叫 template_page 函式顯示控制 Form，當 Submit 按下後，CGI 介面接受瀏覽器 Form 的要求，再將此要求傳給 ESDLED_cgi 程式，此程式接受所傳遞的要求後，控制 ESD44B0_B 目標版之 LED。

getvars = getGETvars();

getGETvars 函式將會解析 QUERY_STRING 環境變數，CGI 在 ESDLED_cgi 的 Form 的 Get 方法要求下，QUERY_STRING 環境變數將會如’LED1=on&……’此字串，getGETvars 解析此環境變數，後回傳 2 維陣列，分別為變數名稱，及動作，CGI 程式利用 getGETvars 來控制發光二極體。

template_page(getvars, form_method);

template_page 函式接收已解析過的 getvars 二維陣列，如果名稱 LED1 的相對陣列為 on 即將此發光二極體點亮，依此類推。

透過 CGI 控制發光二極體的專案，讀者應該可以將 ESD44B0_B 目標板變化成各種不同類型的產品，如利用 ESD44B0_B 來控制家中的各類電器，甚至可以當成一台防盜主機，在辦公室即可了解家中的一切狀況，甚至可以用智慧型手機來當成控制台等。

Victor 於加拿大