

以 Sketchup 為基礎之家庭能源 管理系統人機介面設計

陳佑霖

國立台北科技大學
自動化科技研究所
碩士生

e-mail :

iiiiiryan@hotmail.com

蔡孟伸

國立台北科技大學
自動化科技研究所
教授

e-mail :

mstsai@ntut.edu.tw

摘要

將資訊與 3D 圖形介面整合，應用於能源管理系統，達到智慧家庭的概念。本研究電力資訊量測使用 MSP430 開發智慧型單相電表，並加入 Modbus 協定於電表中，使設備間可互相溝通；能源管理系統人機介面則利用 Google Sketchup 將周圍環境繪製出來，並使用內部支援的 Ruby 語法與網頁語法溝通，配合外部執行檔，利用 Modbus 協定與 SAANET 協定讀取設備資訊與控制設備，並顯示於網頁介面，網頁介面可呈現家電設備資訊，並可控制智慧家電狀態，來達到電能監控與節能的效果。

關鍵詞：Sketchup、Modbus、MSP430、智慧電表、家庭能源管理。

Abstract

In this study, a MSP430-chip-based single-phase smart meter is developed to retrieve related device information, for instance real power consumption on devices. Furthermore, homeowners are able to monitor the information, which is retrieved using the developed smart meter, and control devices via a Google Sketchup-based human-machine interface designed in this study as well such that the electricity energy and bills can be saved. Devices can communicate with each other via Modbus protocol and SAANET protocol.

Keywords: Sketchup, Modbus, MSP430, smart meters, home energy management.

1. 前言

由於地球暖化和環保意識之相關議題因應產生，各國皆思考應如何降低溫室效應以及二氧化碳之排放量，像是加裝太陽能發電、風力發電機，或是提升用電設備的效能，主要的目

的是為了達成節能減碳。由於社會變遷以及資訊化的發展，人們對於居家的安全性及生活的便利與舒適性逐漸有所要求，家庭中的家電設備能夠被智慧化，則居家的生活模式將會會大大的不同。

本研究以 Sketchup 3D 繪圖軟體建置實體模型，讓使用者能夠更清楚了解自家設備用電資訊，以及配合智慧家電，造成生活上更為便利性。美國加州一份報告指出：『讓使用者了解自己花費過多的地方，會間接的影響使用行為有效的降低使用量，經過統計約可以節省 25% 至 30% 的費用』。除此之外，Google 也引用牛津大學氣候變遷協會研究指出：『若用戶能取得自身之電力使用的詳細資訊，則其平均可使自身每個月節省 5% 至 15% 的用電費』[1]。住戶用電標準的反饋機制就是每月的帳單，雖然提供了總結的千瓦小時，不提供即時的訊息，使消費者無法明智的決定何時以及如何減少他們的消費[2]，本研究期望以清晰易懂的人機介面，讓使用者清楚了解自家用電資訊以及操作智慧家電的便利性，以達成節能減碳為目標。

本研究以 Sketchup 3D 繪圖軟體及內部支援 Ruby 語法，配合 HTML 網頁視窗來做為人機介面，控制與顯示各設備資訊；在文獻中之能源管理系統以 Visual Studio 設計圖控介面及軟體，結合 Zigbee 無線傳輸設備、SAANET 通訊協定、插座式電表來讀取記錄用電狀態；也有使用 Visual Basic 來建置虛擬系統，提供用戶一個簡單的電腦介面，對圖表進行點選，將資料庫及設備資訊顯示於虛擬環境中[3][4]。

本論文章節組織如下。本論文第二章為相關通訊技術介紹；第三章為介紹本論文所提出之家庭能源管理系統人機介面的系統架構設計；第四章說明本論文所提出之家庭能源管理系統人機介面的建置及實測結果分析；第五章為

本論文的結論。

2. 相關通訊技術介紹

2.1 Modbus 通訊協定

Modbus 是一種串列通信協議，由 Modicon 公司於 1979 年，為使用可程式邏輯控制器而發表出來的，為工業領域通信協議的標準，且為現在工業電子設備之間相當常用的連接方式，Modbus 協議是一個 master/slave 架構的協議，有一個節點是 master，其他使用 Modbus 協議參與通信的節點是 slave，每個 slave 設備都有一個唯一的地址。本研究使用 Modbus RTU 格式如圖 1 所示，所使用的功能為 0x03 讀取資料、0x06 寫入資料，開始與結束判斷是以間隔 3.5 字元時間以上作為判斷，由 Master 傳送資料到 Slave，Slave 回應資料至 Master，並配合錯誤檢查碼 CRC 檢查封包是否傳送完整[5]。

開始	站號	功能碼	資料	錯誤檢查	結束
3.5個字元時間或以上	1 byte	1 byte	0~252 bytes	2 bytes	3.5個字元時間或以上

← 錯誤檢查的計算範圍 →

圖 1 Modbus 封包格式

2.2 TCP-Socket 網路協定

TCP-Socket 流程如圖 2 所示。步驟如下：

- Step 1) 開啟 Socket 介面。Socket 就如同傳輸管道的出入口，建立傳輸應用的兩端端點須各自開啟一 Socket，往後的資料傳送，均依賴 read (送入此 Socket) 或 write (由 Socket 讀出) 資料。
- Step 2) 綁定 Binding Socket 地址。Socket 出入口需 Binding 到 TCP address，網路上的電腦主機才可以依據使用者所指定的 TCP address，建立資料傳送的通道。
- Step 3) 建立 Connection。電腦主機可以使用 Binding TCP address 後的 Socket，建立資料傳送通道。Server/Client 的傳輸模式，Server 端需呼叫 accept() 功能，等待 Client 的连接建立要求。Client 端則呼叫 connect() 功能，要求與 Server 主機建立连接通道。
- Step 4) 傳輸網路資料。Client 與 Server 主機間成功地建立连接通道後。兩端均可藉由各自的 Socket 出入口，read 或 write 資料到對方。
- Step 5) 關閉 Socket 介面。當應用程式不再使用 Socket 出入口做資料傳送時，需關

閉 Socket。

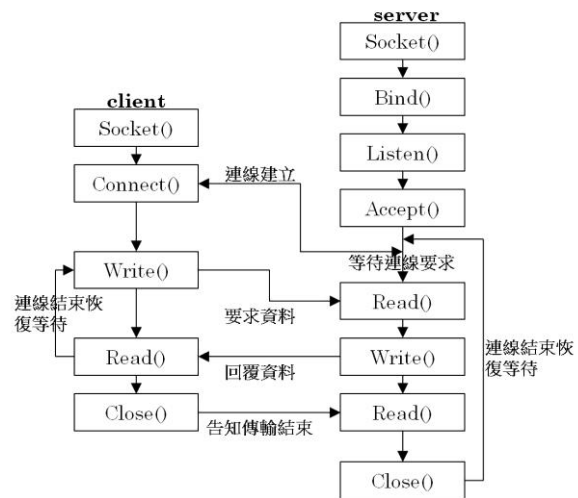


圖 2 TCP-Socket 流程圖

2.3 SAANET 家電協定

SAANET 家電控制通訊協定是智慧家電通訊模組與傳統家電微控制器間的一項雙向通訊協定，其通訊介面為 UART，如圖 3。其中智慧家電通訊模組處理有關 OSI 網路通訊之第一至第四層之協定，而有關第五至第七層之通訊協定則由 SAANET 家電控制通訊協定處理之。有關第一至第四層之通訊協定將包括 SCP、ECHONET、LONTALK 等國際性之智慧家電控制網路通訊協定。

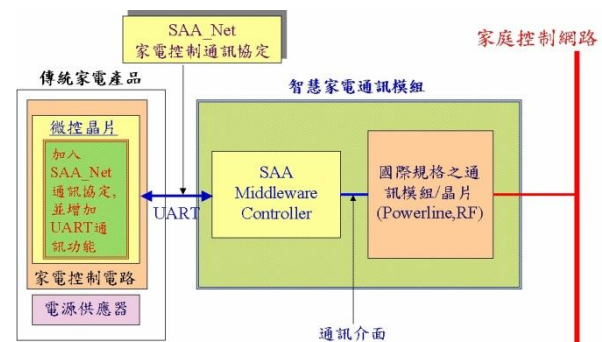


圖 3 智慧家電與傳統家電間通訊架構示意圖

一個 SAANET 家電控制通訊協定封包內含 6 個位元組，其傳送順序為 Byte0 先送，其後依序為 Byte1, Byte2, Byte3, Byte4, Byte5, 共 6 個位元組，每個位元組之內容定義如圖 4 所示 [6]。

		Bit								
		MSB	7	6	5	4	3	2	1	LSB
P A C K E T	Byte0	Packet Length(m Bytes)								
	Byte1	Type ID(家電類別)								
	Byte2	Service ID(功能類別)								
	Byte3	Action ID(R/W) (0:正數, 1:負數)	Data(High Byte)							
	Byte4	Data(Low Byte)								
Byte5	Check Sum(XOR 方法處理)									

圖 4 SAANET 封包格式

其定義如下：

- (1) Byte0 代表封包長度，為 06。
- (2) Byte1 為家電類別，如：冷氣 Type ID 為 01、冰箱為 02、洗衣機為 03、除濕機為 04、電視為 05 等等。
- (3) Byte2 為設定該家電動作之功能碼，如：要設定冷氣的溫度時，將 Byte2 Service ID 設定為 03。
- (4) Byte3 代表 Data High Byte。
- (5) Byte4 代表 Data Low Byte。
- (6) Byte5 為 Checksum Byte，其值計算方式為封包之第一個 Byte 至最後一個 Byte 為止，依序做 XOR 運算，並將其結果存入 Checksum Byte。

3. 系統架構及流程

本研究使用 Google 所提供的 Sketchup 3D 繪圖軟體建置實驗室模型，並使用網頁介面作為顯示設備資訊及控制設備狀態，如圖 5 所示，Client 端為 Window 系統上的 C 執行檔，Server 端為 Linux Ubuntu 系統上的 C 執行檔，Server 端與 Coordinator 連接，發送 Zigbee 封包與各設備 Router 進行通訊，Router 分別連接於兩台冷氣與單相智慧型電表。

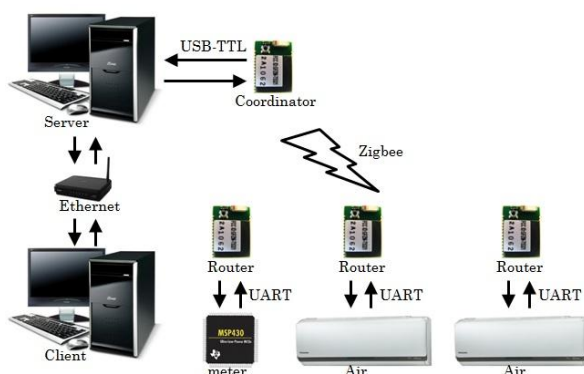


圖 5 系統架構圖

3.1 Google Sketchup 介紹

Sketchup 是一套相當容易學習，且可以直覺方式來操作的一套建模軟體，任何人都能夠迅

速準確的進行 3D 建模。使用 3D 模型，設計人員可做出更明智的決定、傳達專案細節，並與同事和客戶分享想法以達成共同目標。

在 1999 年，總部設在科羅拉多州波爾的 @Last Software 公司由一小群 AEC 軟體業的退伍軍人，為了讓人人在 3D 上設計無障礙而研發 Sketchup 軟體。而研發構想的主要目的就是令軟體有一個簡單而優雅的介面，讓使用者操作起來就像手繪素描一樣的自由。由於 Sketchup 在研發上特別重視設計者對作品的構思草圖，規劃設計、詳細設計乃至定稿一路的編修過程，所以令這套軟體不但具有現代數位媒介的速度，同時也融合了鉛筆素描的自然筆觸與手繪製圖靈活的自由度。並且還添加了材質庫的附貼、光線陰影介面等功能，更擁有動態 3D 模型展示功能取代了立體模型製作這繁複費時費工的事項。

Sketchup 內建 Ruby API，提供 Sketchup Ruby 開發者能擴展功能，以滿足他們的需求，通過創建 Ruby 文件於插件目錄下(Plugins)，能讓 Sketchup 做到下列事情：

- (1) 創建自行定義的繪圖工具，如創建視窗的工具。
- (2) 附加屬性到圖形元素，如成本、供應商等。
- (3) 自動執行常見任務，如產生一組旋轉的場景。
- (4) 分享資料於 Javascript 與 Ruby 之間。

本研究利用 Ruby 與 Javascript 之間資料分享，來傳送命令與接收命令，語法如下：

- (1) Executing JavaScript from a Sketchup script(*execute_script*).
- (2) Accessing Sketchup script from JavaScript(
HTML: `window.location="skp:callback_name@callback_data";` Ruby: `add_action_callback`).

並使用 Sketchup 提供的 Ruby API 來達成外掛選單、元件點選偵測、訊息視窗.. 等，語法如下：

- (1) Sketchup Ruby API(`Tool.onLButtonDown PickHelper.best_picked Tool.activate ComponentDefinition.name UI.menu UI.messagebox`).

本研究 Google Sketchup 安裝於 PC 上，欲

使用 TCP-Socket 連接方式與 Ubuntu 上 Server 進行連接，而 Sketchup 上 Ruby 語言並未支援 TCP-Socket 協定的 library，則本研究利用外部 C 執行檔取代 TCP-Socket 的 Client 端，利用 Ruby 的系統函數執行外部 C 執行檔，來達到與 Server 進行溝通，進而控制各設備狀態與讀取資訊。

3.2 系統流程

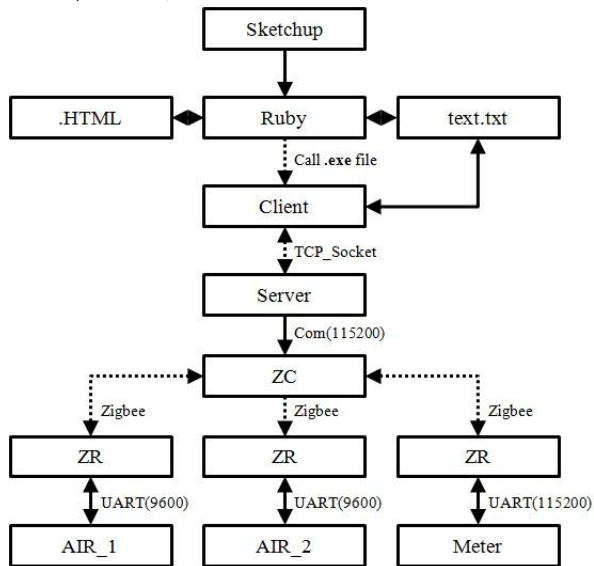


圖 6 系統流程圖

本文所發展之系統流程如圖 6 所示。在系統執行開始時，先開啟 Sketchup，並將 Ruby 所撰寫的外掛程式打開後，對元件做點選，則 Ruby 會偵測到被點選之元件，並開啟對應之網頁介面，網頁介面上設置功能按鈕及顯示文字視窗，當點選功能按鈕時，網頁將所點選的功能傳給 Ruby，Ruby 收到欲執行的功能，將命令寫入文字檔後，啟動外部執行檔 Client 端將文字檔內命令讀出，利用 TCP-Socket 將命令傳給 Server 端，Server 端收到命令後，對設備傳遞對應的命令經由 ZC(Zigbee coordinator)分別傳至對應設備的 ZR(Zigbee Router)，並等待設備回傳資訊，再將資訊從 Server 端回傳回 Client 端後，Client 端將資訊寫入文字檔，Ruby 通知網頁檔案已寫入，則網頁將文字檔讀出顯示於文字視窗內。本研究可讀取資訊的設備包含智慧型單相電表、智慧家電等。

3.3 智慧型單相電表



圖 7 智慧電表

圖 7 為本研究所使用的智慧型單相電表，利用 TI 公司的產品 MSP430 微控制器來開發智慧型單相電表，並加入 Modbus 協定於微控制器中，讓不同設備之間可以互相溝通。所使用的晶片為 MSP430F5529，目前使用兩組 12bits A/D 轉換器，分別量測電壓、電流訊號，並計算電壓 RMS 值、電流 RMS 值、視在功率及頻率；1 組 UART 與 TI Zigbee CC2530 模組進行溝通；2 組 Timer 分別計時 A/D 轉換器取樣頻率與 Modbus RTU 接收 3.5 字元時間計算。

3.3.1 前端電路設計

電壓、電流訊號來源為市電大約 110V、60Hz，由於交流電之弦波有正、負半週之分，而晶片可接受的訊號來源為 0~2.5V，所以需要加入變壓器來降壓，且需加入直流偏壓來消除負半週訊號，以達到晶片可接受電壓訊號範圍，本研究加入前端電路如圖 8 及圖 9 所示，

比壓器(PT)、比流器(CT)為將訊號縮小至晶片能接受的範圍，並加入直流偏壓 VCC，使訊號均為正值，以符合晶片可接受的訊號範圍。

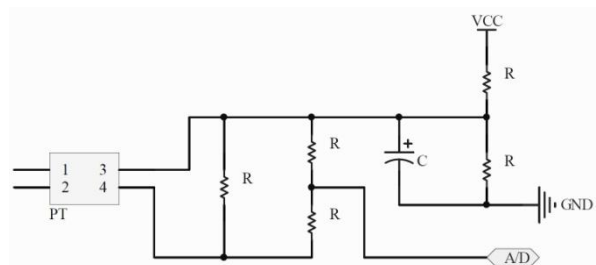


圖 8 PT 前端電路

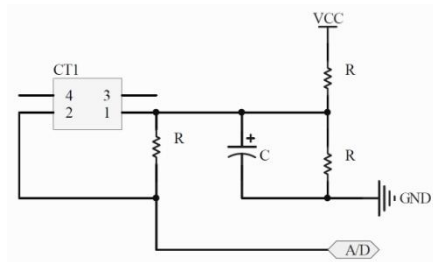


圖 9 CT 前端電路

3.3.2 電力參數計算

本研究智慧型單相電表，分別採用以下(1)~(4)之公式來計算電力參數於 MSP430 中，取樣頻率為 4096 Hz，則 1 秒取樣數則為 4096 筆，分別計算電壓、電流、功率、頻率等電力參數[7]。

- (1) **電壓有效值**： V_n 為第 n 個電壓取樣值； K_V 為電壓校正值；Sample count 為一秒取樣總數目。

$$V_{RMS} = K_V \times \sqrt{\frac{\sum_{n=1}^{\text{Sample count}} V_n^2}{\text{Sample count}}} \quad (a)$$

- (2) **電流有效值**： I_n 為第 n 個電流取樣值； K_I 為電流校正值；Sample count 為一秒取樣總數目。

$$I_{RMS} = K_I \times \sqrt{\frac{\sum_{n=1}^{\text{Sample count}} I_n^2}{\text{Sample count}}} \quad (b)$$

- (3) **視在功率**：在交流電路中，正弦交流電電路電壓有效值與電流有效值的乘積。

$$S = V_{RMS} \times I_{RMS} \quad (c)$$

- (4) **頻率計算**：採用過零點法判斷一週期，並以差分法計算頻率。

過零點法：圖 10 所示，取值會有三種可能，只取 1、3 做為通過零點判斷，通過零點兩次，即為一週期[8]。

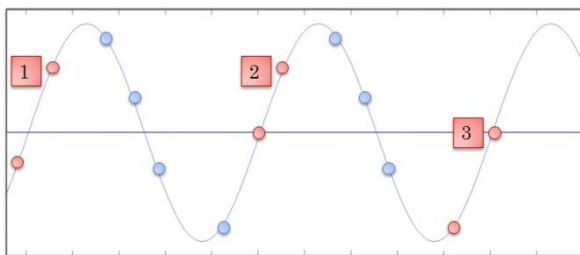


圖 10 過零點法

差分法：圖 11 所示，由(d)式與(e)式求得比例，再以(f)式計算總時間，總時間的倒數即為頻

率。

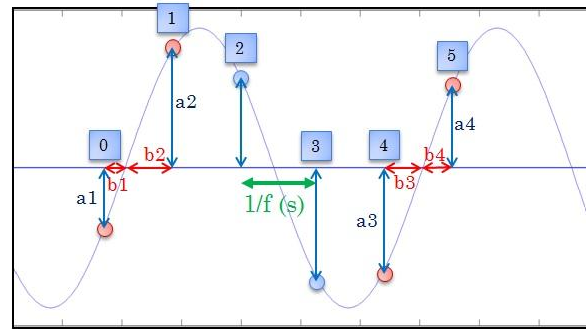


圖 11 差分法

$$a1:a2 = b1:b2 \quad (d)$$

$$a3:a4 = b3:b4 \quad (e)$$

$$\text{Total time} = \left[\frac{a2}{a1+a2} \times (4-1) \times \frac{a3}{a3+a4} \right] \quad (f)$$

3.3.3 傳輸方式

利用智慧電表上 MSP430 晶片與 TI CC2530 Zigbee 模組將 UART 連接為 Router 端，再經由 ZC 將 Zigbee 封包(夾帶 Modbus 命令)傳至 Router 端，Router 將收到 Zigbee 封包傳至 MSP430 微控器中解讀封包，MSP430 收到 Modbus 封包後將電力資訊經由 UART 傳至 Router 端，再經由 Router 回傳至 ZC。

3.4 智慧家電



圖 12 冷氣

本研究使用的智慧家電為 panasonic 所提供的冷氣，如圖 12 所示，冷氣與 PC 端連接，可由 PC 端 Server 傳送 SAANET 封包讀取命令或寫入命令至冷氣之中，獲得其狀態資訊及操控冷氣之功能。

3.4.1 傳輸方式



圖 13 Router

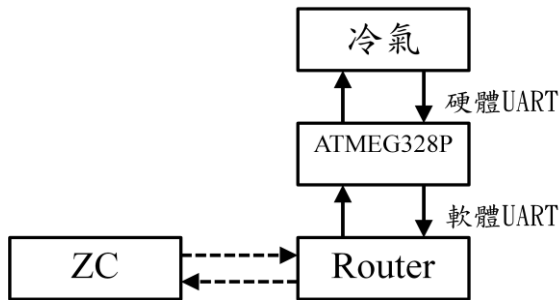


圖 14 硬體架構

圖 13 為安裝於冷氣上的 Zigbee Router，本研究採用 TI CC2530 Zigbee 模組，並加入 ATMEGA328P 微控制器，做為 Zigbee 封包截取，將 SAANET 命令傳送至冷氣。圖 14 為硬體架構，ZC 接於 PC Server 端，ZC 將 Zigbee 封包(夾帶 SAANET 命令)傳至 Router 端，Router 將收到 Zigbee 封包傳至 ATMEGA328P 微控器中解讀封包，再將 SAANET 命令傳至冷氣，冷氣回傳命令至 ATMEGA328P 微控器，再經由 Router 回傳至 ZC[9]。

4. 實測結果分析



圖 15 實驗室模型

圖 15 為本研究的 3D 模型建構，以實驗室為例，目前有點選功能的元件包含兩台冷氣，並於人機介面上讀取其狀態與控制其功能，並加入智慧型電表量測其電力資訊。

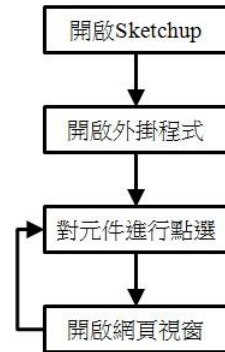


圖 16 操作流程圖

圖 16 為操作流程圖，首先先開啟 Sketchup，如圖 15 所示，



圖 17 外掛程式選單

接著開啟外掛程式如圖 17 所示，

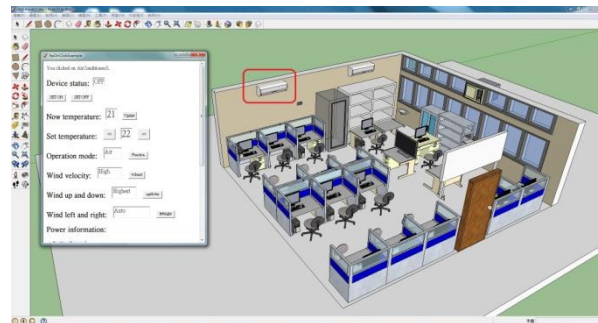


圖 18 冷氣元件

並對冷氣元件做點選如圖 18 所示，

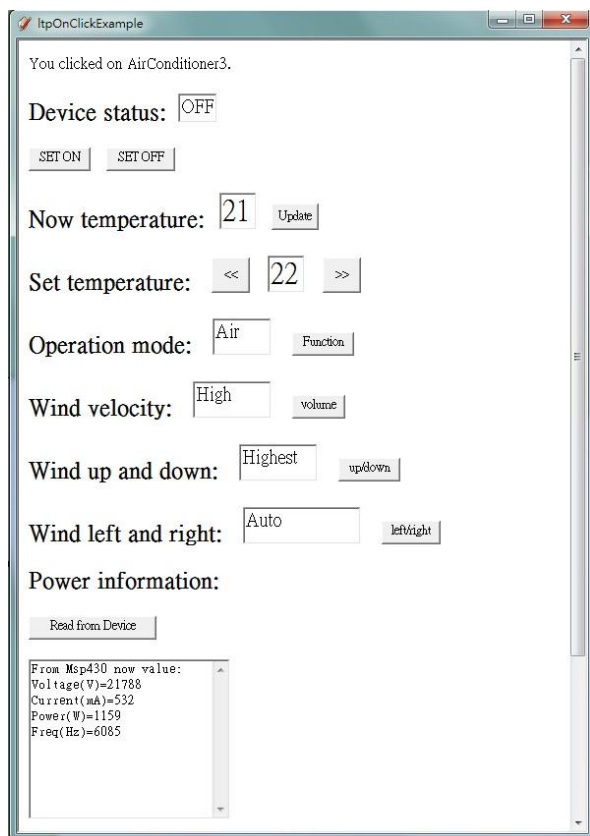


圖 19 網頁視窗

則開啟網頁視窗如圖 19 所示，為冷氣功能操作介面，點選 SET ON 或 SET OFF 時可開啟電源及關閉電源，並將狀態顯示於文字視窗中；點選 Update 時，將設備所有資訊讀回來，包括設備狀態、室內溫度、目前設定之溫度、目前運轉模式、目前風速大小、目前葉片擺動狀態電力資訊等等。

按鍵功能包含以下：

- (1) 設備狀態：ON/OFF
- (2) 室內溫度：XX 度
- (3) 溫度設定：XX 度
- (4) 運轉模式：Air/Dry/Auto
- (5) 風速大小：Auto/Low/Mid/High
- (6) 出風口葉片的上下功能：Auto/Highest/High/Mid/Low/Lowest
- (7) 左右功能：Auto/ Mid-Mid/ Right-Left/ Left-Right/ Left-Left/ Left-Mid/ Mid-Right/ Right-Right
- (8) 電力資訊：包含電壓有效值(217.88V)、電流有效值(5.32A)、視在功率(1159W)與頻率(60.85Hz)

5. 結論

本研究以 Sketchup 為基礎，並使用網頁介面作為監控的人機介面，讓使用者輕易就能知道

目前家電使用狀態，也可方便的使用此人機介面來操作家電功能，以達到智慧家庭的概念。

Sketchup 與人機介面可用於一般有網路連接的電腦，只要透過與 Server 端連接即可了解家電設備狀態與控制。

本研究目前能操控與讀取資訊的設備包含兩台冷氣，連接方式都是無線 Zigbee 傳輸的方式，在未來會陸續加入更多設備進入，使系統更為理想。

致謝

本論文承蒙國科會計畫「智慧電網主軸專案計畫-智慧家庭(建築)電能管理先導型計畫 NSC 101-3113-P-006-020-」之經費支援，特此致謝。

參考文獻

- [1] 蔡明韋、林郁修、蔡孟伸(2011)，「基於 ZigBee 之家庭能源管理系統研發」，2011 年民生電子研討會，台灣台中，11 月。
- [2] Nader, Wisam, “REAL-TIME POWER MONITORING, HOME AUTOMATION AND SUSTAINABILITY” (2011).Architectural Engineering-Dissertations and Student Research, Paper 9, University of Nebraska-Lincoln.
- [3] 游天龍，王安平，林政廷，「網路通訊系統應用於家庭能源之管理與控制」，第四屆智慧生活科技研討會論文集，1097-1102，工業研究院，2009。
- [4] 陳東弘、林政廷(2008)，「智慧型家庭能源管理系統之建置」，2008 年離島資訊技術與應用研討會，澎湖，5 月。
- [5] MODICON, Inc. (1996) Modicon Modbus Protocol Reference Guide, <http://www.modbus.org/specs.php>, U.S.A.
- [6] 凌拯民、王文義、陳品濂(2011)，「智慧家庭無線電能監控管理系統之設計」，2011 年電力工程研討會，台灣新北，12 月。
- [7] 李志宏，2011，多功能智慧型電度表開發，國立臺北科技大學碩士論文，臺北。
- [8] 何學農，2011，現代電能質量測量技術，美國福祿克公司，美國。
- [9] 趙奕昕、林郁修、蔡孟伸(2012)，「無線 Zigbee 智慧型插座開發」，2012 年電力工程研討會，台灣台北，12 月。