

# 南台科技大學

電子工程研究所

碩士學位論文

## 基於雲端之智慧家庭自動化 Cloud-based Smart Home Automation

研 究 生：游又嘉

指導教授：楊榮林

中華民國一〇二年七月



# 南台科技大學 碩士論文

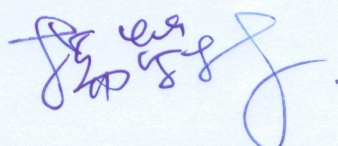
## 基於雲端之智慧家庭自動化

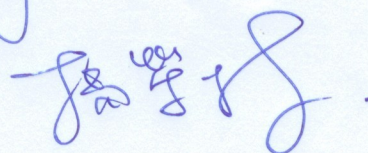
研究生：游又嘉

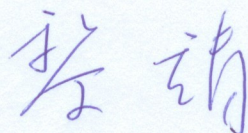
本論文業經審查及口試合格特此證明

論文考試委員

郭景明  姚嘉瑜 

楊榮林 

指導教授 楊榮林 

所 長 黎靖 

中 華 民 國 一 〇 二 年 七 月 二 十 九 日



# 摘要

近年來隨著行動作業系統與雲端科技的進步，同時改變了人們的生活型態，為了提升人類的居家生活品質，居家智慧系統已逐漸發展成型。現今多數的新建房屋大多包含智慧家庭系統，智慧家庭系統不僅擁有防盜功能，使用者更可透過觸控介面或行動裝置，觀看家中狀況或遠端遙控各種電器。

現今擁有居家智慧系統的房屋大多為建造前已規劃在內，對於老舊房屋事想加裝居家智慧系統，不僅金額昂貴且須破壞裝潢，為此我們以系統靈活性與擴充性為優先考量，開發一基於雲端之智慧家庭自動化系統，讓使用者可擴充舊有保全系統，同時增加居家智慧系統的功能，提供濟實惠的解決方案給有需要的家庭。

為降低硬體需求，採用 Arduino MEGA 作為居家防護的核心，我們將其命名為「HSG」。並整合多項 Google 所提供的免費服務，開發一低成本且具擴展性的雲端服務平台，使 HSG 透過此平台相互溝通。

關鍵詞: 智慧家庭系統、感測器、雲端服務、微控制器

# Abstract

In recent years, Mobile Operating System and Cloud Computing Technologies are grows rapidly and changed the behavior of human being. To increase the quality of human life, the Smart Home System has been developed gradually. Nowadays, most of the new construction project contains smart home system. Smart Home System not only has security function but also can provide user remote control the household electric appliance.

Most of the smart home system has been planning before the construction. If people want to retrofitting smart home system, not only take high cost but also have to refit the house. Therefore, we propose a Cloud-based Smart Home Automation to provide users an affordable solution.

In order to cost down the information technology hardware requirements, a low-end Arduino MEGA board is used to implement the critical hardware module for the proposed idea. We named it as home safety gateway (HSG) which module utilizes the public network, internet, to communicate to the other alliance HSG modules. We not only integration multiple Google service but also bring most management and hairy tasks to the nearly free cloud-based service platform, Google app engine, to develop a cheap and extensible web service.

Key words: smart home system, sensor, cloud computing, microprocessor



## 誌 謝

在就讀碩士與專題生的三年半時間內，非常感謝我的指導教授楊榮林 博士。在這段求學期間，楊老師以自身的專業能力，教導我電子領域與程式設計相關專業知識，培養我擁有獨立研究的能力，並在我的研究生涯提供豐富的研究資源和良好的環境。楊老師不僅在學業上給予幫助，同時在做人處事與未來人生規劃上，不時給予關懷和建言，使我在學業、研究或是做人處事上都獲益良多。

同時對於 P306 實驗室，首先我要感謝所有的學長姐小妞、佩萱、佑家、宏煜、胖胖、尤尤、盧哥哥等各位學長們，在專題及碩班期間對我的提攜與照顧。也感謝垣守、佑瑄、維哲、岳修、辰蔚與基服社夥伴們的陪伴，讓我在大學期間充滿笑聲與回憶。再來我要感謝哲齊、祐成、明祐在專題和碩班期間，不管是任何難題，大家一同努力一同患難的過程，感謝你們的支持與陪伴。也感謝六年的同學思賢、育豪、昭毅、世瑋、常仁、必勝以及學弟思摩、俊興、宥佑與育綸的陪伴與合作，讓實驗室總是充滿歡笑與快樂。

最後，我要感謝我最重要的家人和女友思佑，在求學階段對我的支持與付出。首先感謝父母與祖父母二十六年以來的養育之恩，在你們的精心照顧下，不曾給我任何時間、生活與經濟壓力，讓我可以更加的專注於學業。也謝謝我的姐姐乃昕，在生活與做人處事上給予我多次協助與叮嚀。也感謝我的女友思佑，體諒我許多時刻無法陪伴她，並在我低落時給予關懷與鼓勵。最後，我要對以上所有人，致上我最深的感謝與致意。

游又嘉 謹誌

南台科技大學電子工程研究所

中華民國 壹佰 年 七 月



# 目 錄

第一章	緒論 .....	1
1.1	研究動機與目的 .....	1
1.2	系統概述 .....	2
1.3	論文架構 .....	4
第二章	背景研究 .....	5
2.1	居家智慧系統 .....	5
2.1.1	市場現況 .....	5
2.1.2	產品分析 .....	6
2.1.3	分析結果 .....	8
2.2	雲端科技 .....	8
2.2.1	雲端優勢 .....	8
2.3	行動作業系統 .....	9
第三章	居家安全對外匪道 .....	10
3.1	開發語言 .....	10
3.1.1	C 語言 .....	10
3.2	Arduinod 開發板 .....	10
3.2.1	Arduino MEGA .....	10
3.2.2	Arduino Ethernet shield .....	11
3.2.3	Arduino GSM shield .....	12
3.3	感測器 .....	13
3.3.1	溫度感測器 .....	13
3.3.2	磁簧感測器 .....	13
3.3.3	煙霧感測器 .....	14
3.3.4	拉力感測器 .....	15
3.4	遠端遙控與感測器佈防 .....	15
3.4.1	Web Server DNS .....	16
3.4.2	遠端遙控 .....	16
3.4.3	狀態回傳 .....	16
3.4.4	感測器佈防 .....	17
3.4.5	感測器例外排除 .....	17
3.5	簡訊派送 .....	18
第四章	資料庫與雲端服務 .....	19
4.1	開發平台與服務簡介 .....	19
4.1.1	Python .....	19
4.1.2	Google App Engine .....	20
4.1.3	Fusion Tables .....	20

4.1.4	SQL.....	21
4.1.5	Google maps API.....	22
4.1.6	Google APIs Console.....	23
4.1.7	OAuth.....	23
4.2	資料庫結構 .....	24
4.2.1	帳戶資料 .....	24
4.2.2	連絡人資料 .....	25
4.2.3	電器資料 .....	25
4.2.4	排程資料 .....	25
4.2.5	防區資料 .....	26
4.2.6	事件紀錄 .....	26
4.3	GAE 實作 .....	26
4.3.1	Webapp 架構 .....	26
4.3.2	雲端與資料庫連結 .....	27
4.3.3	Service Account 認證 .....	27
4.3.4	應用程式資料存取 .....	29
4.3.5	警報派送 .....	32
第五章	APP 實作 .....	34
5.1	開發平台 .....	34
5.1.1	MIT App Inventor .....	34
5.2	實作方法 .....	35
5.2.1	以 FusiontablesControl 套件實現.....	35
5.2.2	以 GAE 實現 .....	39
第六章	結論與未來展望 .....	41
參考文獻	.....	42



## 圖目錄

圖 1-1 系統架構圖 .....	2
圖 1-2 HSG 群組示意圖 .....	3
圖 2-1 智能家庭整合系統 .....	6
圖 2-2 智能銀髮照護系統 .....	7
圖 2-3 智能家庭控制系統 .....	7
圖 3-1 Arduino MEGA .....	11
圖 3-2 Arduino Ethernet shield .....	12
圖 3-3 Arduino GSM shield .....	12
圖 3-4 磁簧開關 .....	14
圖 3-5 光學煙霧感測器 .....	15
圖 3-6 拉力感測器 .....	15
圖 3-7 HSG 接收電器控制訊息 .....	16
圖 3-8 HSG 回傳狀態格式 .....	17
圖 3-9 HSG 接收例外清單訊息 .....	18
圖 4-1 Google Fusion Tables .....	21
圖 4-2 帳戶資料表單 .....	25
圖 4-3 webapp 配置方式 .....	27
圖 4-4 Service Account 申請步驟 .....	28
圖 4-5 Service Account 取得認證方法 .....	29
圖 4-6 新增聯絡人程式碼 .....	29
圖 4-7 家電控制程式碼 .....	31
圖 4-8 cron.yaml 程式碼 .....	32
圖 4-9 警報郵件內容 .....	33
圖 5-1 MIT App Inventor Layout .....	35
圖 5-2 APP 使用者介面圖 .....	36
圖 5-3 API Key 產生流程 .....	36
圖 5-4 Fusion Tables 傳送資料 .....	37
圖 5-5 Fusion Tables 資料回傳 .....	37
圖 5-6 登入驗證區塊 .....	38
圖 5-7 APP 使用者介面 .....	39
圖 5-8 範例網址 .....	40
圖 5-9 GAE webapp 程式碼 .....	40





## 表目錄

表 3 -1 Arduino MEGA 規格表 .....	11
表 4-1 資料庫分類 .....	24

# 第一章 緒論

## 1.1 研究動機與目的

本論文將開發一基於雲端之智慧家庭自動化系統，提升居家生活便利性與安全性，提供使用者一個便利與安全的生活環境。現今社會家庭結構相較以往不大相同[1]，過去家庭結構大多以三代同堂為主，由退休老人、工作階級加上孩童所組成，形成孩童由退休老人照顧，而孩童也可注意老人的突發狀況，而工作階級可放心外出工作的互補狀態，但現今社會勞動市場的就業年齡與退休年齡逐年拉高，許多家庭中的第一代與第二代同時皆為工作階級，使居家長時間呈現空屋狀態，形成居家安全的一大漏洞。而隨著時間轉移，上述情況家庭中的高齡就業人口退休，通常以達相當高齡的狀態，甚至喪失生活機能，連一般生活都無法自理，許多家庭會將退休老人送至安養院，孩童則交由保母或安親班照護，亦或者少部分家庭在經濟狀況允許之下，額外聘請外籍看護照護家中老人，但不見得所有家庭皆負擔的起這筆額外開銷，加上台灣人口老化問題日趨嚴重，使得居家安全與銀髮族問題逐漸變成許多家庭的一份隱憂，也是近年來許多家庭重要的課題之一。

自進階積體電路輔助設計實驗室(Advanced Integrated Circuit Design & CADLab.)創立今，持續且致力於數位電路設計與嵌入式系統相關技術與研究，而實驗室的研究領域和發展，也隨著雲端科技與行動作業系統的時代來臨，進而更加廣闊與多元，圖 1-1 為系統架構圖。我們將嵌入式系統及微控制器實作經驗，結合所學之程式語言開發專長，設計此「基於雲端之智慧家庭自動化」系統，我們使用 Arduino 開發板與各種擴充模組，整合多種免費 Google API 服務，並搭配自行設計之 Android app，提供一般民眾低價且實惠的居家智慧系統的選擇，為社會付出貢獻盡一份心力，改善他人生活品質與安全。

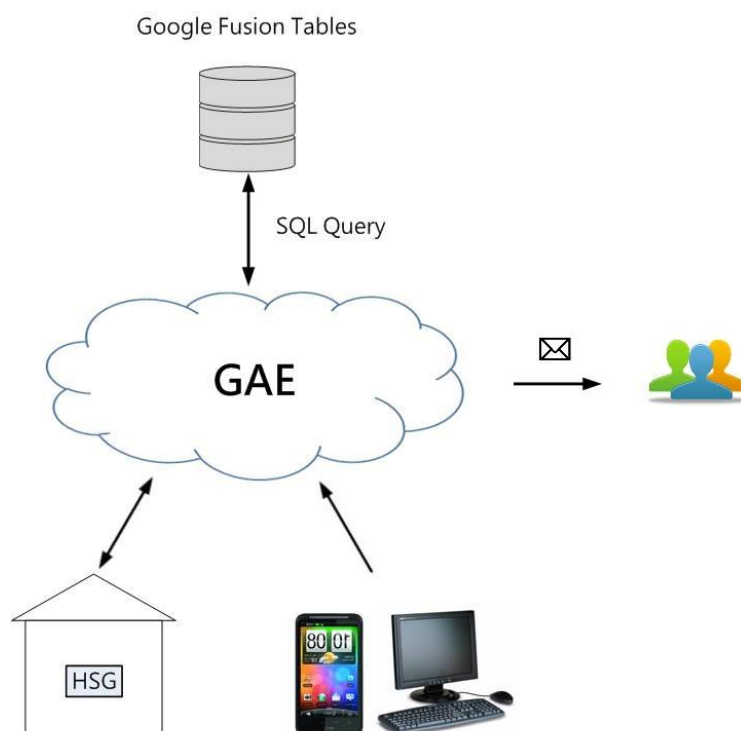


圖 1-1 系統架構圖

## 1.2 系統概述

我們將整體系統分為 HSG(Home Safety Gateway)、雲端服務與使用者介面三大部份，HSG 部分由 Arduino 開發板[2][3]作為中心，接收家中所安裝的各種感測器訊號，加上實驗室自行設計的用電安全偵測與開關例外遙控裝置，並於發生警報時與 GAE(Google App Engine)溝通並進行警報派送服務。在雲端服務方面，是由我們自行開發並放置於 GAE 的網路應用程式，結合 Google Fusion Tables 所組成，在 GAE 上的網路應用程式負責使用者資料管理、遠端設定開關例外、遠端遙控家電、家電定時設定、同盟關係申請/確認/刪除與發送警報訊息，而 Google Fusion Tables 則作為資料存放區，負責存放使用者基本資料、使用者好友名單、同盟紀錄與事件歷史記錄，在用電量或居家環境發生異狀時，HSG 透過網路發送訊息給 GAE 上的網路應用程式，讓應用程式透過存取 Google Fusion Tables 中的使用者資料，

發送警報郵件至好友名單，並將事件紀錄於事件歷史記錄，提供使用者隨時查詢。在一般狀況下，應用程式則提供使用者查看居家安全狀況、遠端遙控電器、變更開關例外裝置、家電定時設定。在本系統中提供給使用者兩種使用介面，分別為 Android App 和 GAE 上的網路應用程式。Android App 我們使用 MIT (Massachusetts Institute of Technology) 所開發的 App Inventor 開發平台，設計 Android 手機 App，提供使用者快速查詢居家用電狀況與遠端遙控開關功能，使用者也可透過 App 快速新增與編輯個人好友名單與查詢事件紀錄，而網路應用程式提供網頁介面並擁有 APP 上所有服務功能，與 APP 不同之處在於擁有申請與編輯同盟功能，使用者可在網頁介面中提出、確認同盟申請，待對方確認過後即可共用好友名單形成同盟，亦或者是刪除同盟關係。

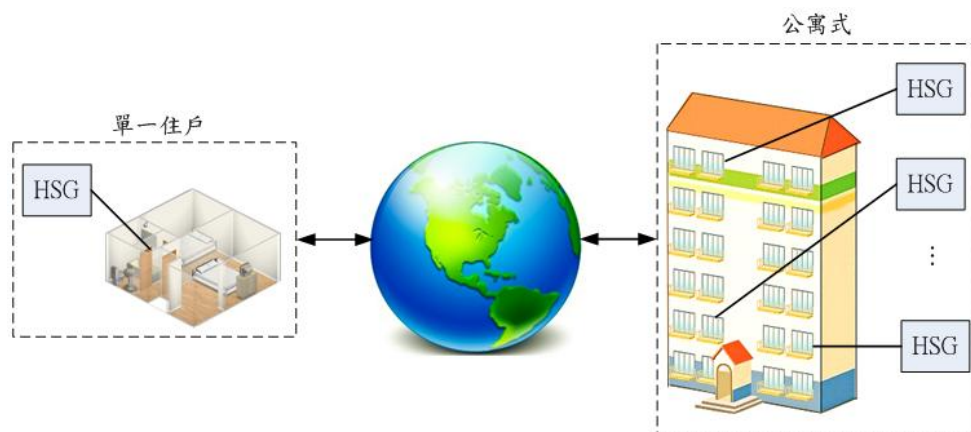


圖 1-2 HSG 群組示意圖

現今的市售智慧家庭與保全系統，大多包含於購屋成本內，在房屋設計與建置時，同時進行配線與系統配置，讓使用者沒有多餘選擇，若遇到使用上的問題或消費糾紛，想增加系統功能或更換保全公司都非常困難。而一般在建置時沒有智慧系統的房屋，若想事後額外加裝系統，則會有需破壞裝潢加上花費昂貴等問題。為避免上述狀況，本系統以系統靈活性與擴充性為優先考量，在設計理念中加入群組化的概念，讓系統不但可獨立運作自成系統，更可當作端點，將自家或

社區分為不同區域，以星狀拓樸或環狀拓樸方式連接，如圖 1-2。使用者也可用於擴充原有智慧家庭系統，解決原有智慧家庭系統的功能擴充問題。對於一開始沒有安裝智慧家庭系統的房屋，為解決加裝系統上的配線問題，一般智慧家庭系統，大多以單一線路串連固定區域中所有感測器。在本系統中我們應用常年來累積的專業領域知識，依照個人需求增加所需之功能，進一步達到客制化功能，讓本系統擁有高擴充性，在使用上更加靈活多變。

### 1.3 論文架構

本論文主要分為六大章節，第二章主要介紹本論文中所用到之硬體與軟體，包含開發版、感測器、程式語言、Google 服務與開發平台。第三章首先介紹 HSG 所使用的規格與硬體周邊，接著描述 HSG 在本研究中所扮演的角色與其所具備之功能。第四章一開始先簡短介紹系統中資料庫的結構，接著介紹雲端的服務與作用。第五章為研究過程中兩種 APP 實作與問題解決。最後第六章進行整體的總結與討論，並提出對本研究未來的展望與期許。

## 第二章 背景研究

### 2.1 居家智慧系統

本章節主要針對一般市售居家智慧系統，進行相關產品與市場分析。包含市場現況與技術分析，作為研究的參考方向與功能目標。

#### 2.1.1 市場現況

近年來台灣隨著科技進步與房市交易熱絡，新興的建案大多擁有居家智慧系統，同時也帶動智慧家庭這個新興產業，全球家電大廠紛紛搶進市場，包含三星電子 (Samsung Electronics)、樂金 (LG)、索尼(Sony)與國際牌(Panasonic)等科技大廠，近年來積極推出相關家電產品。而 Google 則在 2011 年時推出 Android@Home 概念，希望 Android 系統的手持式裝置加入家庭自動化控制的領域，讓手機與平板成為居家中各種家電的控制器。美國的電信商 AT&T 與日本的 NTT DoCoMo，則擁有整合家電的網路平台，並與醫療照護、娛樂與購物等業者結合，形成多元的智慧居家系統。

台灣在智能建築發展上也具有一定的成果，但仍有相當多問題需要克服。目前台灣仍未對智能建築制定相關標準規範與技術法規，各家廠商間沒有一個有效規格，造成管理上無法統一的困擾。台灣必須透過製造、服務、電子、建築與電信業者投入整合，加入健康照護、安全監控和娛樂服務並制定相關規範，才能趕上國外相關廠商的發展速度。





### 2.1.2 產品分析

居家智慧系統為建築與科技兩個產業的結合，涵蓋保全系統、居家照護、自動化控制與娛樂服務四大部分。我們分析尚茂公司[4]所推出的產品，並以其系統作為本研究的功能目標。

## ● 保全系統

圖 2-1 為尚茂公司所推出的智能家庭整合系統，使用者可以透過網路得知家中的狀況。居家感測器以無線方式將狀態回傳到家中主機，主機會透過網路將狀態上傳至伺服器，使用者可以利用手機或電腦連線到伺服器查看狀態，當家中發生異狀時伺服器會傳送訊息通知使用者。

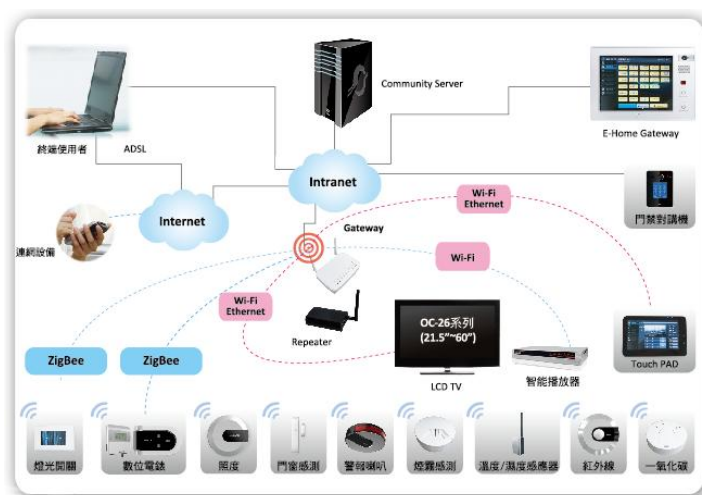


圖 2-1 智能家庭整合系統

## ● 居家照護

圖 2-2 為智能銀髮照護系統，連結使用者家中體重、血壓、血糖計等醫療設備定期量測生理資訊，並上傳數據進行分析統計，並結合視訊與網路提供異常事件警告、緊急求救與視訊諮詢等服務。在娛樂方面透過智慧型顯示器與電視，提供

**健康照護(Health care)**

整合家中生理監測設備，連結遠距健康照護平台，提供有人健康照護服務。

The diagram illustrates a smart home health care system. At the top, a hospital (醫院) is connected via a cloud (雲端) to a smart home. The smart home is equipped with various sensors and devices for monitoring health and safety. The components include:

- 遠距醫療問診服務 (Remote medical consultation service):** Represented by a cloud icon and a video call interface.
- 生理訊號採收設備 (Physiological signal collection device):** A device that collects physiological data from the home.
- 生理監測資訊 (Physiological monitoring information):** A screen displaying monitoring data.
- 生理狀態緊急提醒設備 (Physiological status emergency reminder device):** A device that provides emergency alerts for physiological status.
- 體溫異常警示 (Body temperature abnormality warning):** A device that provides warnings for abnormal body temperature.
- 跌倒偵測 (Fall detection):** A device that detects falls.

- 自動化控制

The diagram illustrates a smart home system architecture. A central **Home Information Gateway** is connected to various smart devices across a three-story house:

- 3F:** A **警報器** (Alarm) is connected to the gateway.
- 2F:** A **煙霧偵測器** (Smoke Detector) is connected to the gateway.
- 1F:** A **智慧型門鎖對講機** (Smart Door Lock Intercom) is connected to the gateway.
- 1F:** A **瓦斯偵測器** (Gas Detector) is connected to the gateway.
- 1F:** A **瓦斯控制閥** (Gas Control Valve) is connected to the gateway.

7

### 2.1.3 分析結果

透過分析以上各項市售產品，我們以保全系統與自動化控制作為本研究之重點。以系統靈活性與擴充性為優先考量，讓使用者獨立使用也可結合舊有保全系統，並可遠端控制家電達到自動化控制的功能。

## 2.2 雲端科技

雲端科技不僅改變資訊科技的發展趨勢，同時也改變了商業的運作模式。對資訊科技來說，雲端不是創新技術而是創新概念，但對於商業經營來說，卻大不相同，因此我們針對雲端服務與以往自架伺服器方式進行分析，了解雲端的優勢與。

### 2.2.1 雲端優勢

企業與大型機構的營運方式隨著雲端服務問世正逐漸在改變當中，企業為因應資料儲存與運算，須設立機房架設 IT 設備，並設立專屬 IT 部門，其後續維護與修繕成本更佔去大半 IT 預算，而雲端運算與資料存取皆進行於供應商伺服器，使用者依照需求租用所需等級的雲端服務，低需求用戶甚至使用免付費等級即可符合所需。隨著資料及用戶增加，只需升級雲端伺服器等級，不管在提升硬體效能(垂直擴展)或伺服器分散負載(橫向擴充)方面，相較於自架伺服器，具備更佳的擴充性，加上租用伺服器成本遠低於購買伺服器設備，更省去日後維修所帶來的龐大成本。

## 2.3 行動作業系統

自從 Apple iOS 與 Google Android 的問世，大幅改變了行動作業系統的現況，讓智慧型手機與平板在 2010 年開始出現暴量的增加，預計 2014 年智慧型裝置出貨量將達到 14 億之多，因此產品與服務結合智慧型裝置，將是未來不可避免的趨勢。

## 第三章 居家安全對外匯道

本系統中 HSG 負責監控家中感測器與電器狀態，當警報發生時傳送資訊給雲端服務進行警報通知，同時也負責接收雲端所傳來的訊息，加以分析後執行任務動作。HSG 在一般狀況時皆為 web server[5]狀態，當需要對外傳送訊息時則轉為 web client[6]狀態。

### 3.1 開發語言

#### 3.1.1 C 語言

C 語言是一種用於系統與應用程式開發的程式語言。具有豐富功能、高靈活性、較強表達力、高效率和高移植性等優點。近 25 年來 C 語言備受程式設計師青睞，是近年來最為廣泛的程式語言[7]。C 語言存在於各種作業系統中，如 UNIX、Microsoft Windows 及 Linux 等，並影響許多後來出現的程式語言，如 C++、C#、Java。

### 3.2 Arduino 開發板

本章節將介紹整體系統中所使用的硬體，首先介紹 Arduino 開發版與 Arduino 擴充模組，接著介紹感測器功能與整合過程。

#### 3.2.1 Arduino MEGA

表 3 -1 為 Arduino MEGA 規格表，Arduino MEGA 是 Arduino 針對高端發者所研發出的作品。Arduino MEGA[8]以具有 Bootloader 的 ATmega1280 作為核心，並擁有 54 個數位 I/O 介面，包含 14 個提供 PWM 的輸出介面、16 個類比 I/O 介面與 4 組 UART(硬體串列埠)、一個 16MH 的石英震盪器，以及一個 USB 接口可以連結 Android 手機或裝置，圖 3-1 為 Arduino MEGA 實體圖。提供開發者選擇 USB 介面供電或 AC-DC 適配器、電池做為外部供電電源。由於 ATmega1280 內建 Bootloader，因此能透過 USB 直接下載程式，不需使用其他外部燒入器，同時降低開發成本。在本研究中做為 HSG 的主要核心，負責接收感測器訊號與控制電器開關。

表 3 -1 Arduino MEGA 規格表

Arduino MEGA 規格表	
控制器核心	ATmega2560
控制電壓	5V
建議輸入電	7-12 V
最大輸入電壓	6-20 V
數位 I/O Pins	54
類比輸入 Pins	6 組
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz



圖 3-1 Arduino MEGA

### 3.2.2 Arduino Ethernet shield

圖 3-2 為 Arduino Ethernet shield[9]，專門用於擴充網路功能的 Arduino 擴充模



組，透過內建的 Wiznet W5100 乙太網路晶片，讓 Arduino 具有可以傳遞 TCP 及 UDP 網路叢集的功能，透過編碼之後讓 Arduino 成為簡易的 Web Sever，即可透過網路遠端控制/存取 Arduino 的數位與類比介面狀態等網路應用。同時也保留原本 PIN 腳規劃，其他擴充板可疊接在板上做應用，並支援 micro-SD 插槽將通過網路服務取得的資料儲存至 SD 卡內。

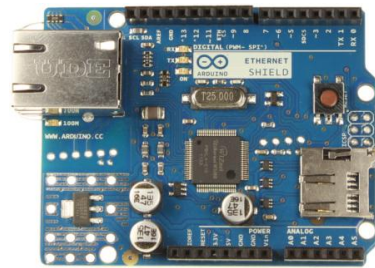


圖 3-2 Arduino Ethernet shield

### 3.2.3 Arduino GSM shield

圖 3-3 為 Arduino GSM shield[10]，專門用於擴充 GSM 功能的 Arduino 擴充模組。具有 SIM 卡插槽，支援 GSM850MHz、GSM900MHz、DCS1800MH 與 PCS1900MHz 多種標準。



圖 3-3 Arduino GSM shield

### 3.3 感測器

感測器為接收信號或激發時產生反應的器具，以取代人類五大感官記錄為主，分別為視覺、聽覺、觸覺、嗅覺與味覺，另外包含探測、接收外界訊號、物理條件與人類無法偵測的化學變化，並限於指定條件且不受其餘外在事物影響，條件觸發後經由轉換變成對應輸出的裝置。一般以物理、生物、化學原理分為三大類，常用於安全、自動化控制、監控與防盜。以下介紹居家安全中最常使用與本研究相關感測器。

#### 3.3.1 溫度感測器

偵測環境溫度變化，將溫度傳回接收處，以便分析是否發生異狀，用於高用電量電器插座旁。

#### 3.3.2 磁簧感測器

磁簧開關為一般最常見的感測器，裡面擁有一組磁簧開關做為磁感應元件，可做為設備定位、計次、設備現況偵測。內部構造由磁鐵裝置加上磁簧感應器，由二片鐵質簧片，平行或垂直重疊放置中間形成一道間隙，一般兩片扁平簧片會鍍上銻或金確保其最佳功能，再將簧片完全密封於充入惰性氣體的玻璃管中，當磁場接近時磁性足夠時簧片相吸，並觸發訊號由 OFF 轉至 ON。由於構造上沒有機械式零件與電子零件，因此可以避免卡住與電路損壞等故障發生，加上接點壽命可達數百萬次，一般最常用於門窗檢測。圖 3-4 為磁簧開關。



圖 3-4 磁簧開關

### 3.3.3 煙霧感測器

煙霧感測器一般分為光學煙霧感測器和離子煙霧感測器兩種。煙霧感測器內部分為兩部分，一部分負責發射光束一部分負責感測光束，當煙霧通過光束時，煙霧中的微粒會阻礙並反射一部分的光源，感測器偵測到異常現象送出訊號。離子煙霧偵測器原理與光學煙霧偵測器不同，採用電感器偵測比光學煙霧感測到更小的煙霧微粒。感測器內部中擁有一個空腔，空腔中的空氣被電離可以通過微弱的電流，當煙霧微粒進入空腔，造成空氣中的電阻值上升進而造成電流變小，此時內部的微晶片偵測到電流變小，隨即發送訊號或警報。美國國家標準與技術研究所(National Institute of Standards and Technology)曾對兩種煙霧感測器測試，在擁有火焰的火災中，離子式比起光電式偵測器快約 50 秒偵測到訊號，但悶燒型沒有火焰的火災中，離子式煙霧比起光電式偵測器卻慢了 30 分鐘，為達到最好的安全效果，同時安裝兩種是使用者最好的選擇。一般裝置於室內天花板用於火災檢測。

圖 3-5 為光學煙霧感測器。



圖 3-5 光學煙霧感測器

### 3.3.4 拉力感測器

拉力感測器是利用一組強力彈簧產生反向力，讓使用者使用鋼索或強力線材，將彈簧拉長至安全範圍內固定，一旦彈簧拉力改變離開安全範圍即發出訊號通知。本研究中結合以往專題概念，以拉力感測器作為感應端，將拉力感測器裝設在窗外，使用鋼絲及其他不易斷裂的線材進行延伸，取代一般傳統家用鋁窗，類似隱形鐵窗功能，兼具美觀、安全，且改善鐵窗逃生困難造成的危險。圖 3-6 為拉力感測器。

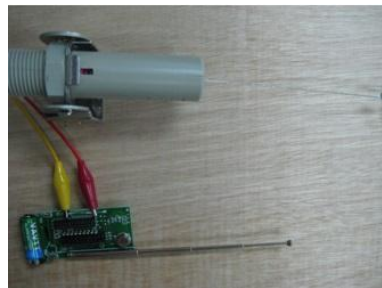


圖 3-6 拉力感測器

## 3.4 遠端遙控與感測器佈防

本研究中所使用的 Arduino MEGA 透過擴充 Arduino Ethernet shield 擁有對外網路功能，透過引數方式將參數包裹於 URL，即可與雲端服務上的應用程式進行溝通，達到遠端遙控與感測器佈防功能。Arduino MEGA 具有數位與類比 I/O 接腳，數位接腳負責遠端控制服務，而類比輸入則負責感測器佈防。

### 3.4.1 Web Server DNS

一般家庭通常在家中會裝設網路路由器，因此 HSG 的 Arduino 在架設 web server 時，需透過第三方 DDNS 服務取得專屬網址，才能接收外部網路以及雲端所傳送的訊息。而本研究中採用的是免費的第三方服務 No-IP，透過申請帳號使用者即可，

### 3.4.2 遠端遙控

使用者可將電器或裝置的電源接上簡易開關電路，並將開關電路控制接腳連結至 Arduino MEGA 上的數位 I/O，當使用者透過雲端服務頁面與我們自行設計的 Android App 遠端遙控時，雲端服務會先將電器狀態寫入電器資料中，再將參數包裹於使用者的 ARDUINO\_URL 內，傳送至使用者家中 HSG 的 Arduino，圖 3-7 為 HSG 接收到的 URL 範例，HSG 經由分析後對腳位進行控制。

```
GET /?D4=true&D5=true&D1=true&D2=false&D3=false HTTP/1.1
```

圖 3-7 HSG 接收電器控制訊息

### 3.4.3 狀態回傳

本系統中所有使用者資訊會存放於 Fusion Tables，而雲端服務會不定時向 Arduino 要求電器與感測器狀態，此時 Arduino 會以 web client 身分使用 GET 方式傳送訊息給雲端服務，如圖 3-8 中所示 Arduino 會依照雲端要求，回傳家電或感測器的狀態值，以 URL 方式將參數包裹於其中，讓雲端服務加以分析取得其中的參數。

```
改變家電的狀態值
req_client.println("GET /ElectCtrl/arduinochange/elect?user=xilan&SPcode=qaz123&cmd=D1:true,D2:false HTTP/1.1");

改變感測器的狀態值
req_client.println("GET /ElectCtrl/arduinochange/sensor?user=xilan&SPcode=qaz123&1F=27 HTTP/1.1");
```

圖 3-8 HSG 回傳狀態格式

### 3.4.4 感測器佈防

類比輸入腳負責感測器佈防，我們使用研究室自行設計的 ID 識別裝置，經由此裝置將防盜感測器與舊有感測器串接，形成一個防區，而此防區可拆分為數個區塊，每一個區塊擁有獨特且唯一的辨識碼，使用此辨識碼做為感測器的觸發位置判斷，當防區內感測器發生異狀，Arduino 辨別完位置後立即透過網際網路與雲端服務連結，並將資訊參數以引數方式包裹於 URL 內，讓雲端服務派送警報郵件給私人聯絡人與同盟好友。

### 3.4.5 感測器例外排除

在本研究中感測器是以防區方式區分位置，每一個防區內所有經感測器串連後接上 ID 識別裝置，裝置會依照電阻值產生一組編碼，而在 Arduino 內具有所有防區個別比對碼，當比對碼與編碼進行比對不符後，Arduino 便通知雲端服務啟動警報。由於一般狀況下大多數感測器普遍裝置於門窗上與出入口，一旦遇到特殊事件，需要暫時解除某防區中單一或多個感測器的警報感測，如：某扇窗戶需暫時打開通風、家庭聚會避免誤觸關閉所有感測器。

當使用者在雲端上設定例外狀態時，雲端會將使用者設定轉為比對碼，並更新於 Fusion Tables 中的防區資料，同時傳送給 HSG 的 Arduino 更新防區比對碼，暫時解除防區中的部分感測器警報，圖 3-9 為 HSG 接收到的例外清單格式範例，將收到的訊息經過字串處理後，會得到 1F 防區比對碼 94、2F 防區比對碼 2、3F



防區比對碼 96 的參數。

```
GET /?1F=94&2F=2&3F=96 HTTP/1.1
```

圖 3-9 HSG 接收例外清單訊息

### 3.5 簡訊派送

在本研究中主要以電子郵件方式派送警報，警報內附有警報發生位址與聯絡方式，為避免突發性網路問題，間接造成郵件警報派送失誤，為求安全起見可擴充 GSM 模組派送手機簡訊，確保其警報派送強健性。

## 第四章 資料庫與雲端服務

本系統中以雲端服務為主要運作核心，負責所有對資料庫存取、提供使用者網頁服務介面、遠端遙控電器、電器定時排程、警報派送和 APP 訊息接收。而在研究初期規劃中 Android APP 是以不經由雲端服務，直接透過套件對資料庫進行存取，但開發過程中遇到些許無法解決之問題，決定改由透過 webapp 方式讓雲端服務接收參數，分析之後為 APP 進行資料存取動作。

### 4.1 開發平台與服務簡介

本章節主要介紹本研究中所使用的開發平台與相關 Google 服務，首先介紹雲端開發的平台 GAE 與開發語言 Python，接著是資料庫所使用的 Fusion Tables 與 SQL，最後為雲端與資料庫連結所使用的認證方式。

#### 4.1.1 Python

Python 是一種成熟且穩定的物件導向、直譯式程式語言[11]，支援命令式程式設計、物件導向程式設計、函數式編程、面向側面程式設計、泛型編程等多種編程範式，直譯器為 C 語言編寫的 CPython。Python 具有清晰簡潔的語法和完善的標準庫。大部分使用無重複意義的英語單字，避免文字衝突並達到語法清晰簡潔的效果，而最大特色在於定義程式區塊和層級方式，多數程式語言大多採用括號及分號定義程式區塊與層級，而 Python 與其他程式語言不同，採用縮排對齊換行做為區分方式。Python 與大多數動態語言一樣，具有 GC(garbage collection)功能，自動管理記憶體使用狀態。

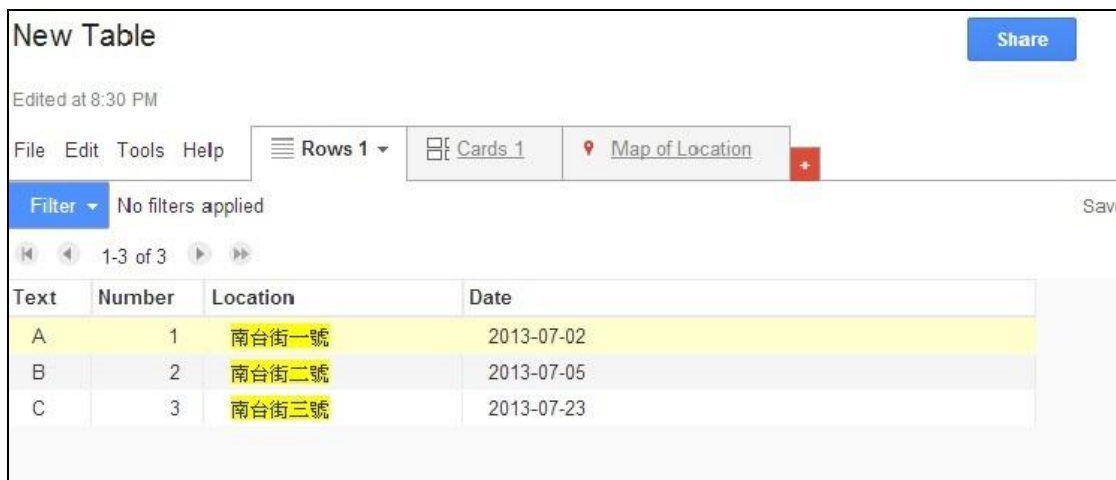


## 4.1.2 Google App Engine

GAE 是一個使用 Google 資料管理中心，提供開發者開發、託管網路應用程式的開發平台，2008 年 4 月由 Google 發行第一個 beta 版本。GAE 以雲端運算技術為基礎，跨越多個伺服器與資料中心來虛擬化應用程式，支援多種程式設計語言編寫的網路應用程式。GAE 提供 Python 專屬執行階段環境[12]，包含 Python 直譯器與標準函式庫，同時也提供 Java 執行環境，使用 Java (JVM、Java servlet、Java) 或使用 JVM 直譯器或編譯器的語言 (JavaScript、Ruby) 建置應用程式。Python 跟 Java 執行環境可確保應用程式快速、安全地執行，且不受系統其他應用程式阻礙，GAE 提供使用者固定額度的資源，若需更多的資源支援，僅需支付額外的費用即可獲得更多的儲存空間、頻寬與 CPU 負載。

## 4.1.3 Fusion Tables

Fusion Tables 為 2009 年 Google 所推出類資料庫/試算表服務，如圖 4-1，但目前仍處於試用階段。Google Fusion Tables 提供使用者將資料以類似 SQL 方式儲存於表單中，加上支援 csv、tsv、xls 等檔案格式，讓使用者可以輕鬆轉換平台。對於應用程式開發者，Google 提供 Fusion Tables 對應 API[13]，可對資料進行存取與更新，方便開發者設計各種應用程式。同時，Fusion Tables 也提供專業的資料統計與地圖化功能，將資料顯示在地圖或強度圖，如：線狀、條狀、餅狀圖或散點圖，也可以使用運動表格或者顯示於時間線，讓以往使用者需要專業 GIS 軟體才能達到的效果，在 Fusion Tables 上輕易的就可以達成，更可透過權限設定，可與他人共同享有編輯或檢視功能。Google 對於 Fusion Tables 使用者提供分享平台，使用者與其使用者之間，可輕易地相互交換分享各種資料與訊息，如：各地旅遊資訊、地區人口統計表等。



**New Table** Share

Edited at 8:30 PM

File Edit Tools Help Rows 1 Cards 1 Map of Location

**Filter** No filters applied Sav

1-3 of 3

Text	Number	Location	Date
A	1	南台街一號	2013-07-02
B	2	南台街二號	2013-07-05
C	3	南台街三號	2013-07-23

圖 4-1 Google Fusion Tables

#### 4.1.4 SQL

SQL 是一種數據資料庫的標準查詢語言，主要可分為資料定義語言、查詢語言、操作語言、控制語言，屬於高階非結構化程式設計語言[14]。最早使用於 IBM 公司的資料庫系統開發。在 1986 年時，得到了美國國家標準局認證成為國際標準，經過多次更新後，現今最流行的為 1992 年版本。SQL 主要作用在於提供使用者簡便的查詢語言，讓使用者快速查詢資料庫中的資料，以及對資料進行各種操作。SQL 雖然大多沒有圖形介面，但 SQL 語法具有巢狀特性，使其擁有高靈活性與強大的功能，並可搭配各種程式進行各種大量的資料運算。在程式語言中需要許多步驟與程式碼才可達到的功能，在 SQL 上只需要一個語法即可完成(如:SELECT)。但各種資料庫系統通常會對 SQL 規範作些許修改和擴充，所以在不同資料庫系統之間，各自的 SQL 並不能完全互相通用。

- 資料定義語言

用於新增/刪除資料表、新增/刪除/編輯欄位與管理資料結構，如:Create、Drop。

- 資料查詢語言

用於向資料庫索取符合條件資料的相關資訊，如:Select。

- 資料操作語言

用於新增、刪除、更新一筆或多筆資料，如:Insert、Delete、Update。

- 資料控制語言

用於處理與資料庫安全和權限相關的設定，如:Grant、Revoke、Alter。

### 4.1.5 Google maps API

由 Google 所提供之一系列對應 Google Maps API[15]，讓使用者與應用程式開發者使用，包含 Maps JavaScript API、Maps Image API、網路服務、Google Maps API for Business、Places API、Google Earth API，其中僅有 Google Maps API for Business 為付費服務，其餘皆為免費服務。

- Web Services

提供應用程式 Google 地圖服務的 HTTP 介面，以規範的方式將參數包裹於網址內傳送至服務，服務以 JSON 或 XML 資料形式，回傳網址參數中所要求的資料，如:座標、經緯度、地址等相關地理資料，讓應用程式加以分析使用。

- Google Maps API for Business

使用與 Web Service 相同的程式碼基礎，但提供企業級使用者更加合適的服務，包括擁有更大的處理服務要求容量、適合商業使用的條款與細則、完善的服務等級保障合約、提供支援與服務、企業內部網路應用程式支援、地圖中的廣告控制權。使企業級使用者輕易的將互動式地圖嵌入於網站上，經過組織後的地圖以便客戶找出所需的資料，進而提升客戶的使用經驗。

- **Places API**

使用與 Web Services 同樣方式，採取網址包裹參數方式向伺服器要求資料，選擇 JSON 或 XML 資料格式回傳。而不同於 Web Services 之處，在於傳送的地點定義成建築物或名勝景點，回傳的資料則是相關周邊資訊與詳細資料。

- **Google Earth API**

Google Earth 是一款由 Google 開發的地球模擬軟體，透過空拍與衛星照片，整合 GIS 資料模擬出地球的 3D 模型。開發人員可透過 Google Earth API 將 Google Earth 所有工具及 3D 成像功能嵌入網頁。同時也可運用 API 進行繪製標記與線條、用圖像掩蓋地表、加入 3D 模型，也可透過載入 KML 檔案等方式，製作出細膩的 3D 地圖應用程式。

#### **4.1.6 Google APIs Console**

Google 開發並提供各種領域基於網際網路的產品與服務，同時也開放給開發者各種 API 呼叫，將開發的應用程式和各種 Google 服務結合。開發者必須透過申請取得金鑰，讓應用程式獲得存取 Google 服務的權限，以往各個服務皆擁有獨立的申請頁面。而隨著服務的多元化，單一應用程式包含數種 Google 服務，此情形下，單一應用程式中必須申請多個金鑰，對於開發者造成一定的困擾，為此 Google 推出 Google APIs Console [16][17]，整合多數 Google 服務的 API 申請，開發者創建專案後，並為其開通所需要使用的 API 後，產生一組金鑰即可通用於多種 API 服務。

#### **4.1.7 OAuth**

OAuth 是一種開放標準，使用者不需要向第三方應用提供帳號與密碼，第三方



應用即可使用使用者在網站上的個人資源。OAuth 允許使用者提供一個令牌(Token)，存取使用者訪放在特定服務上的資料。使用者可指定該令牌的存取權限，指定在特定時段或特定資料而非所有內容。

## 4.2 資料庫結構

如表 4-1 資料庫分類本研究將資料分為管理員資料、使用者私人資料、紀錄資料三類，以表單方式儲存於 Google Fusion Tables 中。所有表單皆擁有固定格式與屬性，因此為求安全與應用程式執行完整性，私人資料表單需待管理員手動創建。

表 4-1 資料庫分類

管理員資料	使用者私人資料	紀錄資料
帳戶表單	聯絡人資料	警報紀錄
	電器資料	同盟申請
	排程資料	註冊申請紀錄
	防區資料	

### 4.2.1 帳戶資料

帳戶資料存放於 Fusion Tables 中的 Account 表單內，如圖 4-2，負責存放使用者的基本資料與其他私人表單的連結參數。每份表單都擁有一個單一且獨特的辨識碼"Table ID"，每個使用者在系統中擁有多個私人表單，當應用程式進行不同服務時，所需要存取的表單也不同，而當新使用者註冊時，個人基本資料在註冊時會由應用程式主動新增加入，但私人表單必須由管理者手動創建，並將這些表單的 Table ID 手動填入 Account 表單內。

Account 存放認證資訊 Fidder at 4:32 PM													
File Edit Tools Help													
Filter No filters applied. Sorted by user													
1-3 of 3													
user	Tel	Date	ID	SPcode	tableid	tableurl	co_owmrtbid	address	ElectCtrl	SensorDetect	Record_table	Logurl	arduino_url
chiayi	0910	2013-06-06	chia	aaaa	1yex13Bz	https://www.viz=G...	1yex13BZn8ht5_sjj4T2b...	南台街1號	1i0JCWxaX9u	1OTFcZ9HX3ZnBfQVD5INNTeeP37	104DRtnLXhUIEQZ5CJgF9IDUvp	https://www.viz=G...	http://appliances.zaplo.c
test	2295	2013-06-27	test	a	1V5UBjul	https://www.viz=G...			1u1kMpvMmQ	1OTFcZ9HX3ZnBfQVD5INNTeeP37			
xilan	1234	2013-06-25	xilai	qaz123	1-jj4T2bfn5WDy1uL	https://www.viz=G...	1-jj4T2bfnMK9M4-5WDy1uUo6KcQfG	南台街1號	1AtvHm3qL8AJAYAbcdDVus	1GH10xivIL9wH6	1-sYF-XsLHXirGdGyDFNSHsgl		http://appliances.zaplo.c

圖 4-2 帳戶資料表單

## 4.2.2 連絡人資料

連絡人表單用於存放使用者的連絡人清單，包含姓名和信箱欄位，負責在使用者與使用者同盟在警報發生時，讓應用程式派送警報郵件使用。

## 4.2.3 電器資料

為每位使用者在 Fusion Tables 中都會擁有的一份表單，負責儲存使用者個腳位電器名稱與開關狀態。當使用者利用網頁或 APP 遠端遙控切換狀態後，會將當前電器狀態儲存於此，當使用者更換腳位或電器時，可以透過網頁中的家電編輯頁面進行名稱修改。

## 4.2.4 排程資料

電器排程表單做應用程式執行排程任務的依據，使用者可以到網頁介面的家電排成頁面，設定某個電器於指定時間改變為指定狀態，所有排程將會記錄於此表單，排程執行成功後也會記錄於事件表單。

## 4.2.5 防區資料

防區管理表單負責存放使用者家中所有防區資訊，包含位置、感測器排序以及比對碼。使用者可以將自家中的感測器區分成多個防區，透過修改比對碼讓使用者在特殊情況下設定例外清單。

## 4.2.6 事件紀錄

事件紀錄可分為註冊申請表單、警報紀錄與同盟申請。註冊申請表單為新使用者申請服務時，記錄申請者基本資料以便管理員開通服務。當使用者觸發警報時，系統會自動將事件、地點或感測器位置寫入警報紀錄表單，以此表單作為依據方便使用者事後查詢，而同盟申請表單負責記錄申請雙方的聯絡人 Table ID，當雙方皆同意時，系統會將雙方的 Table ID 相互加入帳戶資料內的同盟好友欄位。

## 4.3 GAE 實作

本系統中 GAE 上的應用程式使用 webapp 方式，將所有功能依照 URL 方式區分，使用者只須將要求的參數包裹於 URL 內，應用程式便會依照 URL 對應的應用程式執行動作。

### 4.3.1 Webapp 架構

本系統中應用程式使用 webapp 架構進行實作，應用程式會依照使用者所觸發的 URL 進行指定的動作，圖 4-3 為本系統中 URL 的配置，在圖 4-3 中我們可以看到，每一個 URL 後面皆有一個對應的 Class。

```
app = webapp2.WSGIApplication([
    ('/', LoginPage),
    ('/Logout', LogoutPage),
    ('/Contact', ContactPage),
    ('/EditElectCtrl', EditElectCtrlPage),
    ('/ElectCtrl', ElectricDeviceCtrlPage),
    ('/ElectCtrl/change', ChangePage),
    ('/ElectCtrl/change/(.*)', ChangePage),
    ('/ElectCtrl/arduinochange/(.*)', ArduinoPage),
    ('/SensorDetect', SensorDetectPage),
    ('/SensorDetect/(.*)', EditSensorNamePage),
    ('/FriendChoose', FriendPage),
    ('/FriendChoose/(.*)', FriendPage),
    ('/ShowTable/(.*)/(.*)', AndroidPage),
    ('/ShowLog/(.*)/(.*)', AndroidLogPage),
    ('/ShowMap/(.*)/(.*)', ShowMapPage),
    ('/AndroidQueryDelete/(.*)/(.*)/(.*)/(.*)', SQLDelPage),
    ('/AndroidQueryInsert/(.*)/(.*)/(.*)/(.*)', SQLInsertPage),
    ('/AndroidQueryUpdate/(.*)/(.*)/(.*)/(.*)/(.*)', SQLUpdatePage),
    ('/AlarmLog', AlarmLogPage),
    ('/Alarm/(.*)/(.*)/(.*)', AlarmPage),
    ('/AndroidEDCPage/(.*)/(.*)', AndroidEDCPage),
    ('/Schedule', editSchedulePage),
    ('/Schedule/(.*)', editSchedulePage),
], debug=True)
```

圖 4-3 webapp 配置方式

### 4.3.2 雲端與資料庫連結

GAE 與 Fusion Tables 是兩種完全不同的 Google 服務，而當 GAE 上的應用程式要對 Fusion Tables 進行存取資料的方式，是以 API 方式[13]發送 SQL 對資料進行存取。而應用程式存取資料前需取得權限，而 Google APIs Console 中提供 OAuth2.0 API key 與 Service Account 兩種認證方式，在最後實作兩種方式後選擇以 Service Account 認證機制，第五章實作與結果將詳細描述過程與原因。

### 4.3.3 Service Account 認證

本系統中所有使用者資料皆儲存於 Fusion Tables 中，當應用程式要與 Fusion

Tables 做資料存取動作，必須透過 Google Console 中的 Service Account[18]，如圖 4-4 步驟申請一組 Service Account 給應用程式作為認證使用。

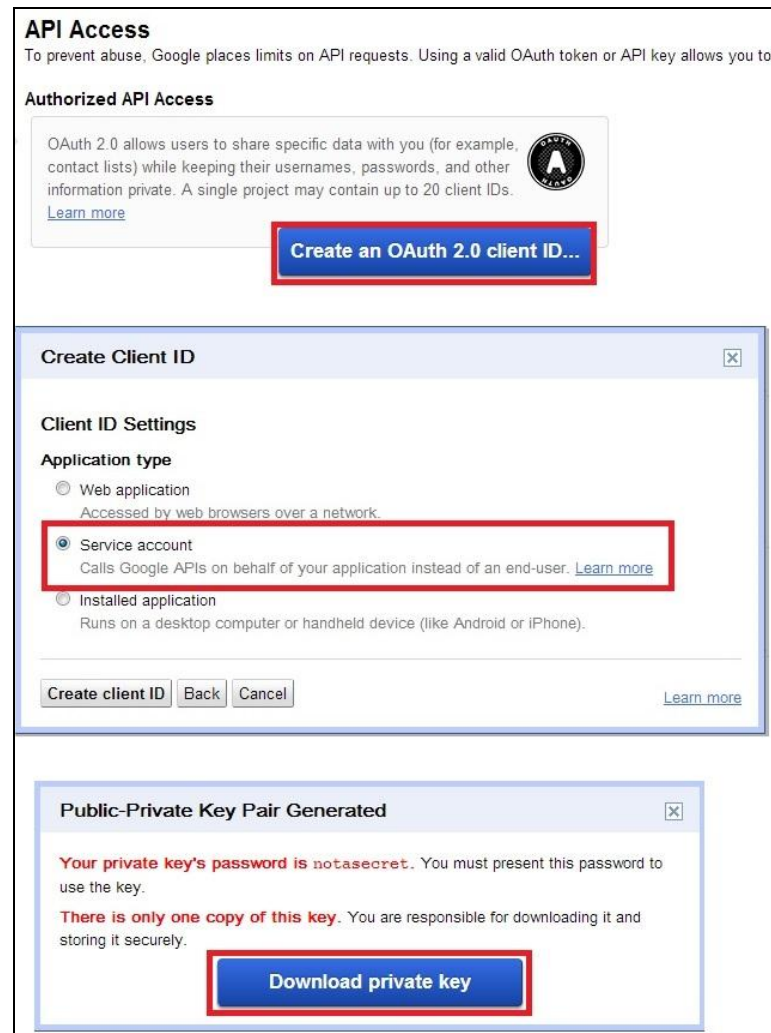


圖 4-4 Service Account 申請步驟

如上述申請完 Service Account 後，系統會給開發者 Client ID、Email Address 以及一個 pcks 12 格式的 Private Key，首先開發者必須先把 Client ID、Email Address 記錄下來，再將 Private Key 透過 openssl 轉成 pem 格式，開發者須將 privatekey.pem 與 PyCryptoSignedJWT.py 檔案放到 GAE 專案內，並在應用程式中加入程式碼及所需參數，其中 SERVICE\_ACCOUNT 和 SCOPES 分別為 Email Address 與 Fusion Tables api 認證網址，透過第三方的函式庫 PyCryptoSignedJWT 包裹參數傳送至認

證網址並回傳 Access Token，如圖 4-5 程式碼方式取得認證，當應用程式取得認證之後，即可使用 Google 所提供的 API，傳送 SQL 對 Fusion Tables 進行資料存取。。

```
from apiclient.discovery import build
from PyCryptoSignedJWT import PyCryptoSignedJwtAssertionCredentials

class ServiceAccount:
    def getCredentials(self):
        key = open(KEY).read()
        credentials = PyCryptoSignedJwtAssertionCredentials(
            SERVICE_ACCOUNT,
            key,
            scope=" ".join(SCOPES))
        http = httplib2.Http()
        httplib2.debuglevel = True
        return credentials.authorize(http)
```

圖 4-5 Service Account 取得認證方法

#### 4.3.4 應用程式資料存取

圖 4-6 新增連絡人程式碼範例，雲端服務中的應用程式透過 4.3.3 章節中的方法取得認證後，便可以使用 SQL 對資料庫進行存取動作，經由設計後實現雲端服務各項功能。

```
if action == 'add':
    if 'rows' not in query.keys():
        self.response.out.write('add')
        sql = 'INSERT INTO %s (%s,%s) VALUES (\'%s\', \'%s\')' \
            %(CONTACT_TABLE, query['columns'][0],
              query['columns'][1], name, email)
        #self.response.out.write(sql)
        add = service.query().sql(sql=sql).execute(http)
        self.response.out.write(add)
        if 'rows' in add.keys():
            info = "建立成功"
        else:
            info = '已經存在於清單中'
```

圖 4-6 新增聯絡人程式碼

## ● 使用者註冊

對於一般尚未註冊的使用者，必須到雲端服務的登入頁面進行註冊動作，填寫基本資料並提交之後，待管理員審核、創建私人表單與 HSG 安裝，即可進入使用者服務頁面。由於資料庫中大部份資料必需由管理員手動產生並填入，加上 GAE 與 Fusion Tables 具有一定存取次數限制與容量限制，為避免遭人惡意使用，註冊完成後使用者資料僅暫存於申請記錄中。

## ● 編輯聯絡人

本系統中的每位使用者，在 Fusion Tables 皆擁有一份獨立的聯絡人表單，使用者可以登入雲端服務的網頁介面，進行資料修改動作，應用程式會依照使用者的選擇不同，將參數包裹於 SQL 指令內對資料庫進行存取，對聯絡人清單進行新增/修改/刪除等動作。

## ● 互助同盟

本系統提供使用者與使用者之間建立同盟關係，當使用者申請同盟時系統會將申請者與對象記錄於同盟申請記錄，應用程式會透過 SQL 的 SELECT 方式，尋找使用者是否同盟申請，透過網頁的介面提供使用者勾選是否同意。每位使用者僅能看到自己的聯絡人清單，並無法看到同盟關係者的聯絡人清單，但在警報派送時，應用程式同時會對私人名單以及同盟關係者的聯絡人清單進行派送。家電管理

## ● 家電控制

每位使用者家中 HSG 的 Arduino 都具有 Web Server 功能，當使用者透過雲端服務設定家電狀態時，雲端服務會將狀態更新至家電管理表單，接著將參數包裹於 URL 內傳送給 HSG，HSG 經由分析後取得參數控制電器改變狀態，圖 4-7 為家

電控制程式碼，應用程式先從帳戶資料內取得使用者的 ARDUINO\_URL，以及網頁中使用者改變的電器狀態 form\_data，透過 fetch 函式中的 GET 方式給 HSG 的 Arduino。

```
else:
    cmds = self.request.arguments()
    ARDUINO_URL = self.request.cookies.get('ARDUINO_URL')
    #print('-----')
    form_fields={}
    for acmd in cmds:
        state = self.request.get(acmd).encode('utf-8')
        acmd = acmd.encode('utf-8')
        form_fields[acmd] = state.encode('utf-8')
        form_data = urllib.urlencode(form_fields)
    try:
        url = '%s?%s' % (ARDUINO_URL, form_data)
        print(url)
        result = urlfetch.fetch(url,
                                method=urlfetch.GET,
                                headers={'Content-Type': 'application/x-www-form-urlencoded'})

        self.redirect('/ElectCtrl')
    except DownloadError:
        self.response.out.write('error:')
        self.response.out.write(url)
        print(url)
    #tell arduino what i changed use url fetch
```

圖 4-7 家電控制程式碼

## ● 家電排程

本系統除了提供使用者遠端控制電器開關功能，也提供家電排程頁面，讓使用者設定預定工作任務排程。當使用者必須對特定電器或裝置，在指定時間內指定開關狀態，或者對電器與裝置使用達到某種循環規律，即可透過此項服務功能達到自動化的效果。目前服務提供每天、每週、每月三種週期，讓使用者快速預定排程，使用者選擇完週期和次數後，雲端會依照次數與規律週期自動產生排程記錄於資料庫中。

一般狀況下應用程式在未觸發時並不會動作，而為達到排程功能我們必須持



續觸發應用程式比對時間，為此我們使用 GAE 所提供的 Cron 排程服務，開發者可以設定時間點或頻率定時觸發指定的 URL，並將其寫入專案中的 cron.yaml 檔案內，GAE 便會依照所描述的時間定時觸發。圖 4-8 為 cron.yaml 檔，本系統使用此服務，指定每分鐘觸發排程掃描的應用程式 URL，透過掃描所有電器排程表單，時間符合便執行使用者所設定的任務。

```
cron:  
- description: monday morning mailout  
  url: /ElectCtrl/change/sche  
  schedule: every 1 minutes  
  timezone: Asia/Taipei
```

圖 4-8 cron.yaml 程式碼

## ● 防區管理

本系統提供使用者編輯自家防區與例外清單，使用者可以透過此頁面對防區內的感測器進行名稱定義與更改防區比對碼，透過名稱定義使用者可以清楚且明瞭的切換家中不同位置的例外狀態，而更改比對碼後應用程式會先將資料更新於防區資料表單，接者把比對碼以參數方式包裹於 URL 內，傳送給使用者 HSG 的 Arduino 版，更新其中的比對碼以達到例外功能。

### 4.3.5 警報派送

雲端服務上擁有一個隱藏的警報通知應用程式，當家中感測器發出異狀會觸發警報，Arduino 會將警報資訊與使用者帳密以參數方式包裹於 URL，應用程式將所取得的參數與資料庫比對正確後，會依照使用者連絡人清單派送郵件警報，再對同盟好友連絡人清單進行派送，圖 4-9 為警報郵件內容，郵件內容包含使用者聯絡方式、警報發生位址，並包含 Google Maps API[15]產生的兩個網址，為避免中

文編碼問題，一個為經緯度定位、一個為地址定位，應用程式會先取得資料庫中使用者的中文地址，經由 urllib 函式庫中 quote 轉碼後，將編碼包裹於 URL 透過 Google maps API 的服務，回傳地址相關資料的 JSON 檔分析後取得經緯度，將經緯度包裹 Google Maps API 經緯度定址功能的 URL 參數內。使用者只需點選即可連結至 Google Map 查看，而行動裝置則會打開 Google 地圖 App 定位，讓使用者快速取得相關位址資訊。

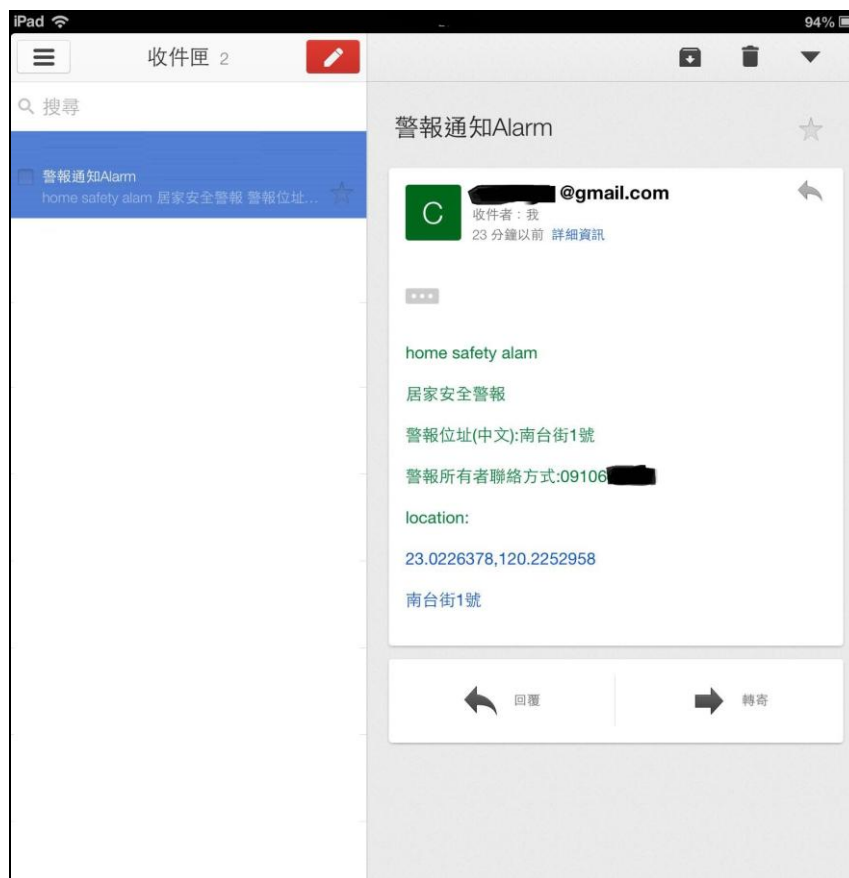


圖 4-9 警報郵件內容

## 第五章 APP 實作

本研究將系統分為三大部分，分別為 HSG、雲端服務、使用者介面，而 Android APP 在本系統中負責將參數包裹於 URL，傳送給雲端服務進行遠端電器控制、聯絡人清單編輯與事件紀錄等功能。

### 5.1 開發平台

#### 5.1.1 MIT App Inventor

App Inventor 是 Google 實驗室的一個子計畫，但在 2011 年 Google 宣布中止計畫並停止網頁版服務。2012 年 1 月 1 日正式轉移給 MIT 麻省理工學院行動學習中心，3 月 4 日以 MIT App Inventor[19]名稱發布並提供 Google 使用者使用。開發者首先在瀏覽器上設計好程式介面，如圖 5-1，接著是應用程式的行為描述，而不同於其他所有程式語言，MIT App Inventor 去除以往繁雜的程式碼描述方式，使用圖形介面的積木拼圖堆疊的程式描述方法，將各種語法描述分成各種積木拼圖，開發者拖曳拼圖相互結合達到功能需求。開發者可以使用 Android 模擬器或使用 Android 手機，透過掃描 QRCode 下載程式到手機即時測試。相較於一般程式編碼方式，MIT App Inventor 安裝環境簡易以及所有運算皆在雲端上執行，非常適合初學者與課程教學使用。

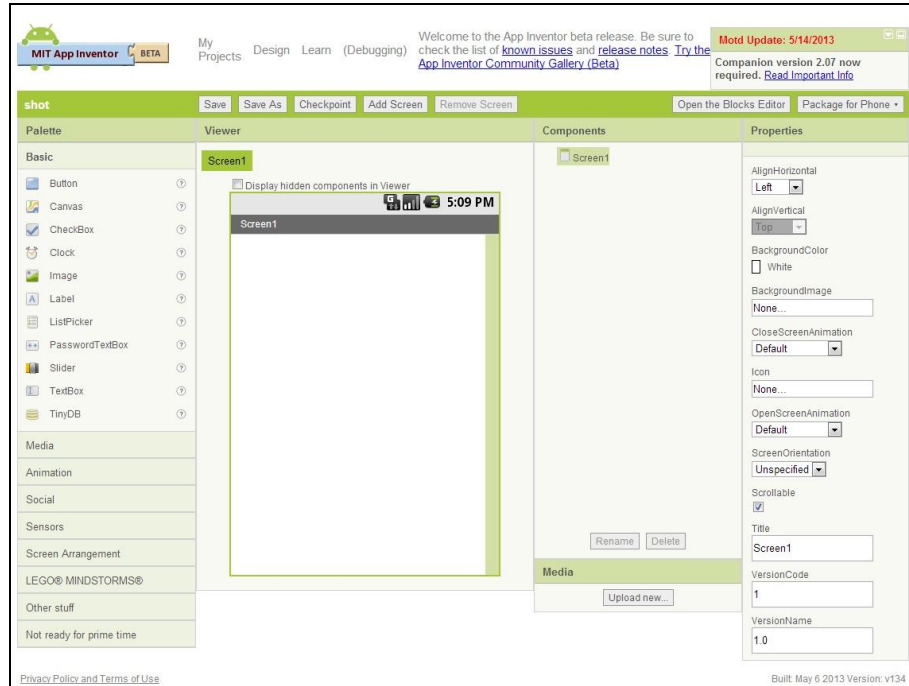


圖 5-1 MIT App Inventor Layout

## 5.2 實作方法

本研究進行期間共使用兩種方式實作 APP，在研究初期採用 APP Inventor 內建的 FusiontablesControl 元件對資料庫做存取，後期透過 GAE 使用 webapp 搭配 WSGIApplication，透過 URL 觸發 GAE 上特定的應用程式。兩項做法各有其優缺點，最後決定將採用後者將所有資料存取動作交於 GAE 執行。

### 5.2.1 以 FusiontablesControl 套件實現

#### ● APP 介面

圖 5-2 為 APP 登入後的使用介面圖，具有基本存取資料以及位置查詢功能，並顯示目前聯絡人清單。

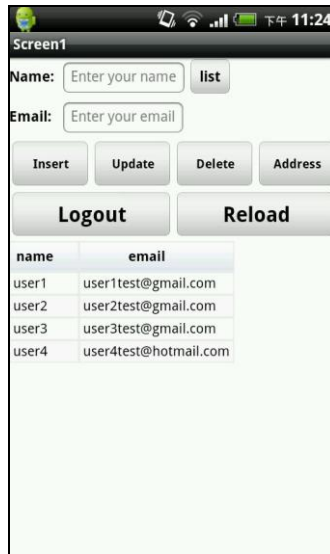


圖 5-2 APP 使用者介面圖

## ● APIKey 取得

在 MIT App Inventor 中提供 FusiontablesControl 套件，讓開發者可以透過功能區塊與 Google Fusion Tables 連結。如圖 5-3 中步驟，開發者需到 Google APIs Console 專案中開通 Fusion Tables 權限，接著申請一組專屬 Android APP 的 API Key，並將此組 API Key 輸入至 FusiontablesControl 內的 APIKey 欄位中。

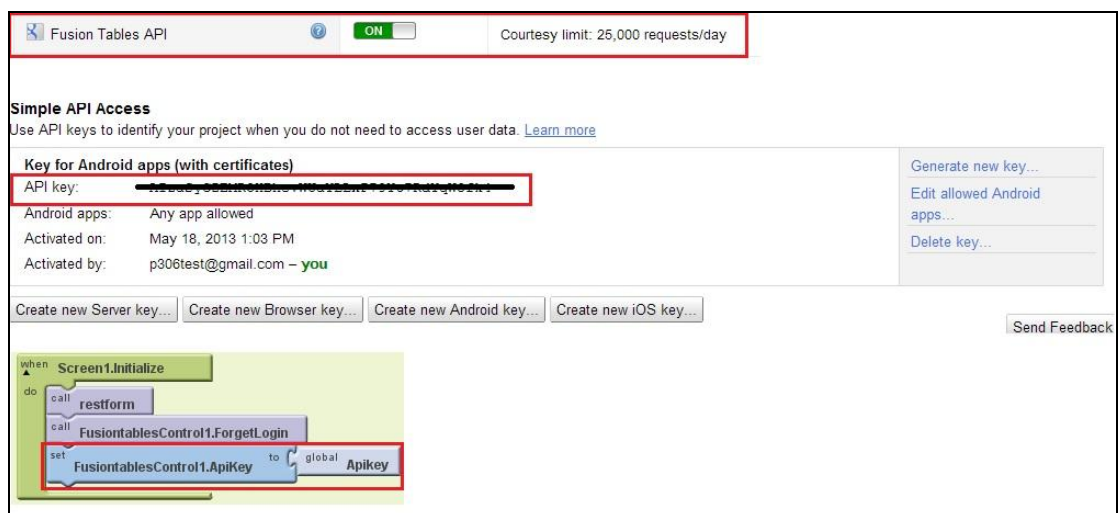


圖 5-3 API Key 產生流程

- Fusion Tables 資料存取方式

Google Fusion Tables 提供對應 API 讓程式開發者在應用程式中，以 SQL 向 Google Fusion Tables 表單存取資料，而在 MIT App Inventor 中同樣也提供此服務，開發者可以使用 FusiontablesControl 套件所提供的 Query 與 SendQuery 兩個功能區塊，如圖 5-4，開發者僅需將 SQL 語句字串放入 Query 並透過 SendQuery 送出，如圖 5-5 Google Fusion Tables 會回傳資料讓開發者進行分析取得所需資料。

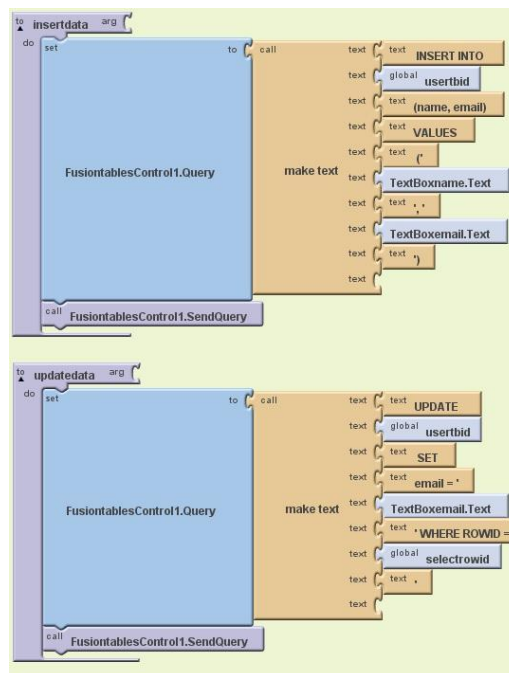


圖 5-4 Fusion Tables 傳送資料

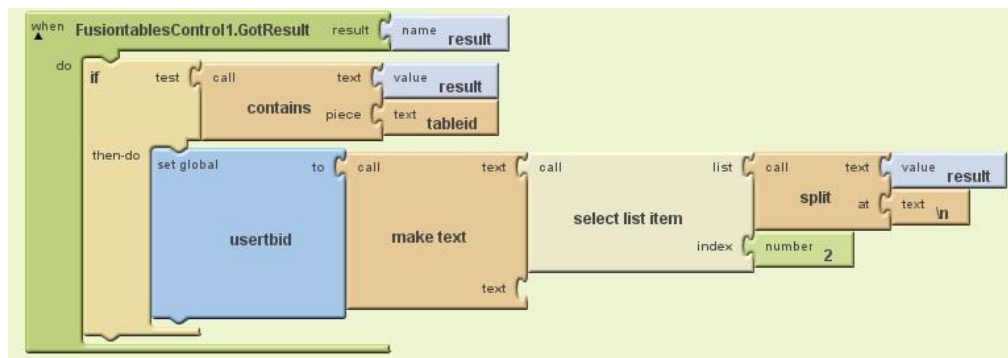


圖 5-5 Fusion Tables 資料回傳

## ● 問題與解決方式

在實作此 APP 的過程中遇到區塊優先權與 Fusion Tables 權限問題，在進行身分驗證時，判斷區塊與資料回傳區塊發生衝突，圖 5-6 為 ButtonNameLogin 按鍵判斷區塊與回傳資料區塊，當按鈕被觸發時呼叫 getSPcode 函式，APP 發送 SQL 給 Fusion Tables，而 Fusion Tables 回傳使用者於 Account 表單中的 SPcode，讓 APP 進行身分驗證動作。但當資料回傳時，必須等到按鍵判斷區塊結束才能進行動作，因此 SPcode 並沒有存進所設定的變數中，因此無法進行身份的驗證，自行加入時間延遲區塊依舊無法解決。

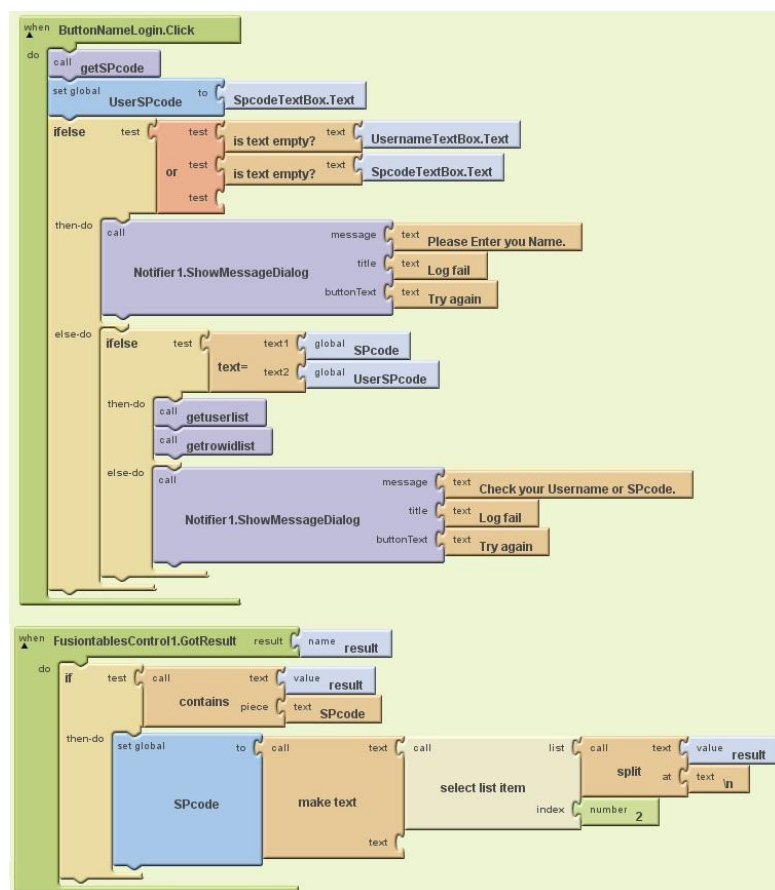


圖 5-6 登入驗證區塊

MIT APP Inventor 中的 FusiontablesControl 套件是以 API Key 存取 Fusion Tables 的資料，但我們發現存取的表單無法為“私人”，權限必須為“知道連結的使用者”

與“公開在網路上”這兩種權限，後者完全無安全性可言，而後者安全性則高於後者一點，但有心人士只需取得 Table ID，即可使用 SQL 取得裡面的資料，因此，我們決定將資料存取的動作放置到雲端服務，並透過 webapp 方式來實現。

## 5.2.2 以 GAE 實現

- APP 介面

圖 5-7 為 APP 登入後的使用介面圖，具有編輯聯絡人資料、遠端電器控制、警報紀錄與 HSG 位置查詢功能，並顯示目前聯絡人清單。

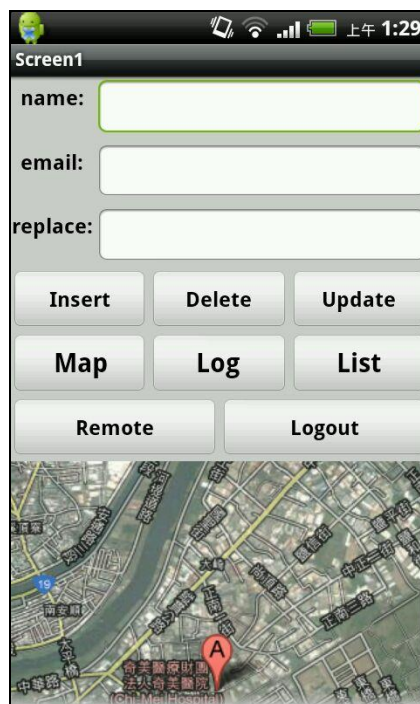


圖 5-7 APP 使用者介面

- 以 WebViewer 觸發 GAE

前一版 APP 在執行過程中並無經過 GAE，而是使用內建 FusionTablesControl



套件與 Fusion Tables 進行資料存取動作，而考量前一版遇到的兩個問題與整體系統流暢性，最後決定將所有對資料存取的動作放置於 GAE，將動作與參數包裹於 URL，經由 WebViewer 觸發後讓 GAE 分析並進行動作。圖 5-8 紅框中為 GAE 接收 APP 訊息參數並分析的頁面。包含查看聯絡人清單(ShowTable)、警報紀錄(ShowLog)、查看位址(ShowMap)、新增聯絡人(AndroidQueryInsert)、刪除聯絡人(AndroidQueryDelete)、更新聯絡人(AndroidQueryUpdate)與電器控制(AndroidEDCPage)。例:當使用者新增聯絡人 alex、信箱為 alex@gmail.com，當 Insert 按鈕按下時，隱藏的 WebViewer 會觸發圖 5-9 中的 URL。

<http://APPURL/AndroidQueryInsert/username/SPcode/alex/alex@gmail.com>

圖 5-8 範例網址

GAE 會依照 username 與 SPcode 去尋找該新增的聯絡人表單，並新增 alex 與 alex@gmail.com。

```
(/FriendChoose', FriendPage),  
(/FriendChoose/(.*)/(.*)', FriendPage),  
(/ShowTable/(.*)/(.*)', AndroidPage),  
(/ShowLog/(.*)/(.*)', AndroidLogPage),  
(/ShowMap/(.*)/(.*)', ShowMapPage),  
(/AndroidQueryDelete/(.*)/(.*)/(.*)/(.*)', SQLDelPage),  
(/AndroidQueryInsert/(.*)/(.*)/(.*)/(.*)', SQLInsertPage),  
(/AndroidQueryUpdate/(.*)/(.*)/(.*)/(.*)', SQLUpdatePage),  
(/AlarmLog', AlarmLogPage),  
(/Alarm/(.*)/(.*)/(.*)', AlarmPage),  
(/AndroidEDCPage/(.*)/(.*)', AndroidEDCPage),  
#('/sche', sche),  
(/Schedule', editSchedulePage),  
(/Schedule/(.*)', editSchedulePage),  
, debug=True)
```

圖 5-9 GAE webapp 程式碼

## 第六章 結論與未來展望

本研究對於想整合雲端服務與硬體的開發者，在入門時具有不錯的參考價值。本論文包含開發環境設定、應用程式認證過程與開發環境配置詳細呈現，讓入門者可以參考本研究的系統架構，自行設計雲端服務並整合硬體開發版。

我們用自身的專業知識與語言撰寫能力開發出此系統，本研究中除了硬體成本外，其餘服務及開發平台皆為免費，但也因此在功能上難免受到限制，若使用者過多則需要付出些許費用提升限制。而本研究在裝設硬體部分仍需相當繁瑣的過程，並且在使用者註冊後的表單創建仍無法自動化產生，須由開發者手動操作，對整體效率而言具有相當的影響。在 APP 功能部分太過簡易，對於整體使用的即時性上仍顯不足。

## 參考文獻

- [1]內政部統計處, 2013, 101 年底人口結構分析, 內政部, Available:  
[http://www.moi.gov.tw/stat/news\\_content.aspx?sn=7121](http://www.moi.gov.tw/stat/news_content.aspx?sn=7121)
- [2]Michael Margolis , “Arduino Cookbook”, 2nd Edition, O'Reilly Media, 2011
- [3]Massimo Banzi , “Getting Started with Arduino”, O'Reilly Media, 2008
- [4] 尚茂智能科技股份有限公司, [Online], 2013, Available:  
<http://www.sunmoretek.com/index.php>
- [5]Arduino Ethernet Library Web Server, [Online], Feb 12, 2013, Available:  
<http://arduino.cc/en/Tutorial/WebServer>
- [6]Arduino Ethernet Library Web Client, [Online], April 23, 2012, Available:  
<http://arduino.cc/en/Tutorial/WebClient>
- [7]TIOBE Software BV. Jul.2013. TIOBE Programming Community Index for July 2013. TIOBE Software BV. Available:  
<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>
- [8]Arduino Mega, [Online], April 26 2012, Available:  
<http://arduino.cc/en/Main/arduinoBoardMega>
- [9]Arduino Ethernet Shield, [Online], May 16, 2013, Available:  
<http://arduino.cc/en/reference/ethernet>
- [10]Arduino GSM Shield, [Online], Mar 11, 2013, Available:  
<http://arduino.cc/en/Main/ArduinoGSMShield>
- [11]David Beazley, Brian K. Jones , “Python Cookbook”, 3rd Edition, O'Reilly Media, 2013
- [12]Google App Engine for Python, [Online], Jun 20, 2013, Available:  
<https://developers.google.com/appengine/docs/python/overview?hl=zh-tw>
- [13]Row and Query SQL Reference, [Online], May 8, 2013, Available:  
<https://developers.google.com/fusionaltables/docs/v1/sql-reference>
- [14] SQL 語法教學, [Online], 2013, Available: [www.1keydata.com/tw/sql/sql.html](http://www.1keydata.com/tw/sql/sql.html)
- [15]Google Maps API, [Online], Feb 1, 2013, Available:  
<https://developers.google.com/maps/>
- [16]Google APIs Console Help, [Online], Mar 13, 2013, Available:  
<https://developers.google.com/console/help/?hl=zh-TW>
- [17]Google Accounts Authentication and Authorization, [Online], Feb 16, 2013.  
Available: <https://developers.google.com/accounts/docs/OAuth2>
- [18]Use Application-owned Accounts, [Online], July 17, 2013, Available:  
[https://developers.google.com/drive/service-accounts#google\\_apis\\_console\\_project\\_service\\_accounts](https://developers.google.com/drive/service-accounts#google_apis_console_project_service_accounts)



[19]Support for Using MIT App Inventor, [Online], 2013, Available:  
<http://appinventor.mit.edu/explore/support.html>