

建構於智慧型行動裝置之複合式行車資訊儀表板的設計與實現

曾清標

崑山科技大學 電子工程系

一、研發目的：

車子的安全性是現代人購車的標準之一，除了車子所配備的安全設備外，如何降低車輛的機械故障，更是行車安全重要的一環，為了帶來更高等級的行車安全，除了平時的保養外，如果在行駛中能夠及時的發現汽車的異常，並適時的做出處置，如此一來才能使自己的生命財產安全都受到保護。

OBD-II(On-board Diagnosis)-車上診斷儀器界面，嵌入在目前的多數汽車和輕型貨車裡，在 70 年代和 80 年代初期間汽車製造商開始使用電子儀器控制引擎作用和診斷引擎問題，這主要將符合 EPA 排放標準，多年來車上診斷系統技術變得更加成熟。OBD-II 新標準在 90 年代中期，提供幾乎完全發動機控制並且監測汽車內部元件的狀況、車身和輔助附屬設備的零件，並且診斷控制汽車零件的狀況。

目前市售的智慧型手機幾乎都支援了，行動網路、相機鏡頭、加速度計、Bluetooth、GPS、擴充記憶體等功能，利用行動裝置內建的功能在搭配上 OBD-II 汽車診斷器，如此一來就可整合出一系列的整合式汽機車儀表系統。

二、階段目標：

為了達成上述之目的，擬以下列之研發主題分階段逐步地實現達到所要的理想目標。

階段一：

- 系統平台建立：

Android 是一個基於 Linux 核心(kernel) 的開放手機平台作業系統。與 Windows Mobile、Symbian 等手機作業系統處在同一級別。

Android 在 Linux 核心的基礎上，提供了各種合用的函式庫，和一個完整的應用程式框架。並採用較符合商用限制的 Apache 版權。在 Linux 核心的基礎上提供 Google 自製的應用程式運行環境（稱作 Dalvik，與 Sun 的 J2ME 不同），並提供基於 Eclipse 整合開發環境(IDE) 的免費、跨平台(Windows、Mac OS X、Linux)開發工具(SDK)，便於應用程式開發者學習、使用。

1. Android 開發軟體: Android 提供免費而且跨平台的整合開發環境，只要電腦能連接上網路，我們隨時都能下載相關工具下來，並開始開發 Android 應用程式。有了輕鬆易用的開發工具，我們可以把心力專注於如何將想法實現到應用程式上。
2. Android 虛擬機器: 由於在剛開始開發時，我們手邊並不一定已擁有 Android 設備。因此 Android 開發工具亦提供了相當強大的模擬器，能讓我們自由配置，模擬各種硬體規格的設備。在 Android 中一律把 Android 模擬器稱作「Android 虛擬機器」(Android Virtual Device)，簡寫為「AVD」。

為了使開發順利，必須建立起完整的系統平台，在 IDE 與 AVD 的開發與模擬都可展現出所撰寫的程式後，再進一步的匯入行動裝置進行一些硬體控制的開發，如此一來才能編寫出適用於行動裝置之 APP。

階段二：

- 外部信號來源：
 1. OBD-II 訊號:使用解碼 IC ELM327 搭配 Bluetooth 傳輸晶片設計出藍芽傳輸介面之汽車診斷器。
- UI 呈現：
 1. 美工圖片設計:此計畫為整合式汽機車儀表系統，在 UI 的版面上必須設計多種背景、錶框、儀錶圖、指針，在美工的構圖方面必須考慮到各種圖形在進行重疊呈現時會產生的透明度與解析度問題，如此才能完整的整合出可切換式儀表，來提供使用者切換。
 2. 類比指針設計:在儀表圖片的整合完成之後，行車狀態的指針變化就必須以程式來加以驅動，來達到使用者所要的視覺效果。
 3. 數位訊號設計:除了類比指針的展示外，及時數據也是不可或缺的資訊之一，所以提供使用者控制數字的開關與顏色大小，來整合出使用者的個人化介面。

註一: OBD 解碼 IC ELM327

1. 支援通訊標準 (OBD, OBDII, EOBD2):
 - ISO 15765-4 (CAN-B 125Kbps)
 - ISO 15765-4 (CAN-C 500Kbps, 11bit ID, 29bit ID)
 - ISO 14230-4 (KWP2000)

ISO 9141-2 (5 baud init. 10.4 kbaud)

SAE J1850 VPW (10.4 kbaud)

SAE J1850 PWM (41.6 kbaud)

SAE J1939

2. 可顯示目前車內感應器數據含：

引擎轉速 Engine RPM

負載 Load

水箱溫度 Coolant Temperature

油箱燃油程度身分 Fuel Status

車速 Vehicle speed

進氣差別器氣壓 Intake Manifold pressure (MAF/MAP)

點火正時 Timing Advance

空氣溫度 Intake Air Temperature

進氣速率 Air Flow Rate

節氣閥身分 Absolute Throttle Position

含氧感知器電壓 Oxygen sensor voltage /short term fuel trim

油壓 Fuel Pressure

階段三：

• 實機匯入與應用

1. Android 行動裝置:上述第二期開發系統是利用虛擬 Android 系統來模擬儀錶系統，第三期開發階段為了匯入現在最夯的 Android 系統的行動裝置平台，並將原本虛擬的 Android 系統換為真正的 Android 行動裝置。
2. GPS 定位:為了實現軌跡紀錄功能必須使用行動裝置中的 GPS 定位晶片，可由 GPS 晶片中取得目前使用者所在的經度、緯度、精確度、標高、時間、速度和方位，當紀錄了這些數據之後，未來可以由分析軟體在搭配上 OBD 的資訊來詳細的分析出所有的行車狀態。
3. 加速度計與陀螺儀:除了 OBD 資訊外，搭配了手機內部的感測器來使行車的數據更加的完整。
4. SQLite 資料庫儲存:當 APP 接收了大量的 OBD 資訊與 GPS 數據，應用程式必須及時儲存接收到的資料到資料庫中，在 Android 作業系統所

採用的方式為標準的 SQLite 資料庫，SQLite 是一種輕型資料庫可以供 APP 任意使用。

- 分析軟體

1. PC Analysis Software(LabVIEW)；
2. Mobile Analysis Software(Android):

當行動裝置於 SQLite 儲存了大量的行車資訊與行動軌跡，為了將大量的數據提供給使用者觀看，必須使用分析軟體來進行，分析軟體不但可以繪製出依照不同速度而有不同的色彩的軌跡外還可以及時的選取軌跡位置的行車資訊，或是將某項數據繪製成圖表，也可將不同時間所取得的數據來進行比對，若使用分析軟體來進行處理顯示，使用者就可以得到越精密的資訊。

三、工作項目：

階段一：Android 開發平台建立。

1. Android 開發軟體。
2. Android 虛擬機器。

階段二：外部硬體設計與 UI 呈現。

1. OBD-II 訊號分析。
2. OBD To Bluetooth 硬體整合。
3. UI 規劃。
4. 美工圖片的取得。
5. 畫面整合。
6. 程式流程與顯示。
7. 虛擬系統模擬 APP 測試。

階段三：實機匯入與應用。

1. 階段二成果移植匯入實機測試。
2. OBD-II 訊號處理。
3. 撰寫 GPS 應用。
4. 加速度計與陀螺儀應用。
5. SQLite 資料儲存。

分析軟體撰寫。

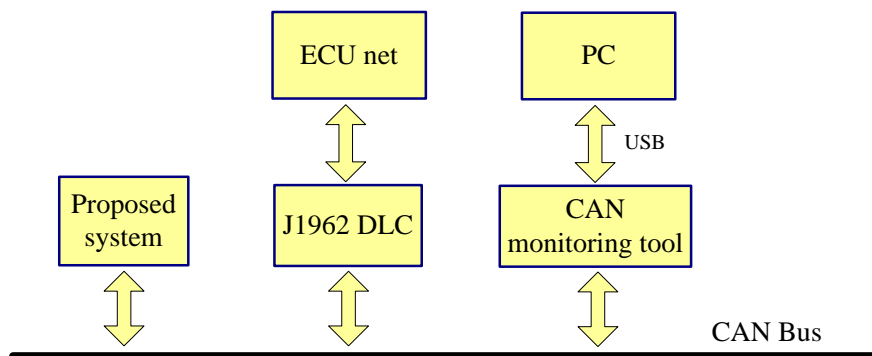
PC Analysis Software(LabVIEW)與 Mobile Analysis Software(Android)

四、發展平台的設計

4.1 系統架構

整體的系統架構包括可延伸的發展平台(Proposed system)、CAN匯流排訊息監測工具(本論文使用Kvaser Memorator硬體與X-Analyser軟體)和車用ECU網路，如圖一所示。透過汽車的J1962 DLC (Diagnostic Link Connector)診斷接頭將可延伸的發展平台連接到車用ECU網路(In-Car ECU net)，讓工程人員於發展平台上所發展的節點裝置也納入車用ECU網路的一部分，進而延伸車用ECU網路到發展平台上，以形成可延伸擴充功能的車用ECU網路，讓新節點或新系統在擴充後的ECU網路裡運行，以驗證新節點或新系統的可行性。

CAN匯流排訊息監測工具透過汽車的J1962 DLC診斷接頭可以將車用ECU網路的傳輸訊息導出，藉由X-Analyser軟體顯示在電腦螢幕上或是儲存在記憶體中。日後可以將儲存的訊息經由解析處理之後寫入可延伸的發展平台之微處理器記憶體中，如此，可延伸的發展平台也就具有模擬(emulating)車用ECU網路的功能，可獨立來發展相關的ECU節點裝置。



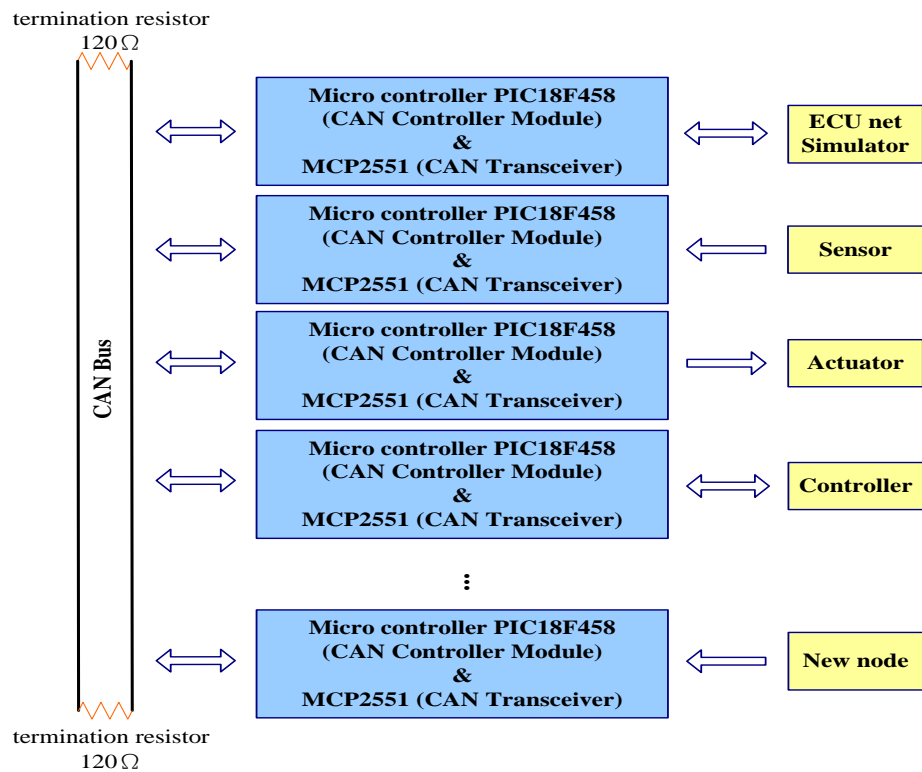
圖一、整體的系統架構

4.2 可延伸的發展平台(Proposed system)

可延伸的發展平台架構如圖二所示，主要是由CAN匯流排節點模組所組成，每個模組都內建有CAN Transceiver(MCP2551)，透過CAN Transceiver將CAN模組相互連接起來，使CAN節點可一直擴充，CAN Transceiver(MCP2551)可提供最大支援112個節點。

利用CAN匯流排節點模組可以開發出許多具有特定功能的節點，例如，如圖二所示的ECU模擬節點(emulator node)、監測節點(monitor node)、感測器節點(sensor node)、控制器節點(controller node)、致動器節點(actuator node)等，由這

些節點就可組成特定功能的應用系統。



圖二、可延伸的發展平台架構

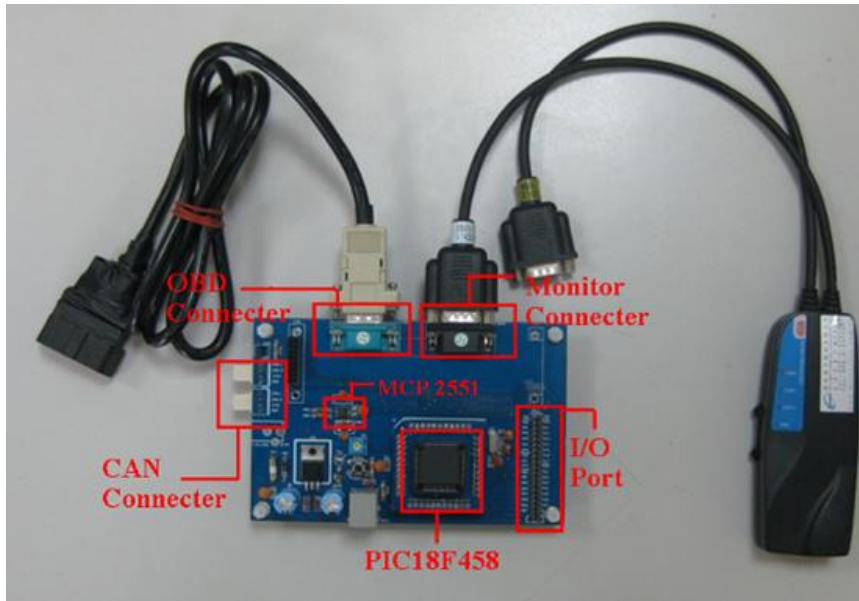
4.3發展平台的節點模組

發展平台之節點模組的外觀照片如圖三所示，建構有CAN匯流排介面、一般用途的數位 I/O Port、可連接汽車CAN OBD的J1962 DLC介面、CAN匯流排訊息監測工具連接端以及CAN節點擴充連接端。

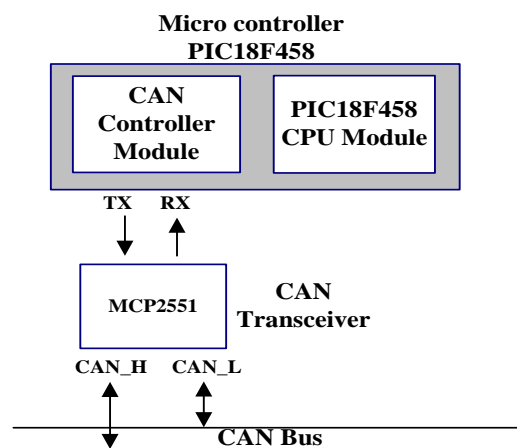
CAN匯流排介面如圖四所示，主要是由Microchip的微控制器PIC18F458和CAN Transceiver MCP2551所組成，是為每個CAN節點都必須具備的硬體介面。PIC18F458這顆MCU晶片所內建的許多資源，諸如Timer、A/D、UART、一般用途數位輸入輸出埠(I/O Port)，CAN匯流排介面模組可依照節點需求功能的不同，加以應用來實現出更多應用功能的節點。

透過OBD-II轉接端可以將CAN OBD與汽車ECU網路連結，讀取車身相關資訊，使節點可以連接至車用ECU網路，成為車用ECU網路的新節點。

Kvaser Memorator透過監測連接端可以連接到CAN匯流排，以監測CAN匯流排訊息傳輸的狀況，並且透過X-Analyzer軟體來加以解析。



圖三、發展平台之節點模組的外觀照片圖



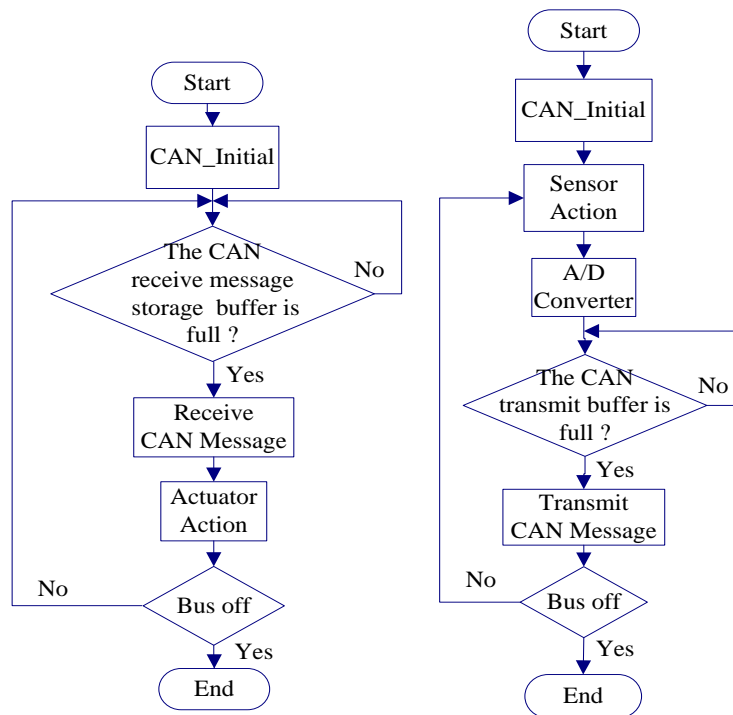
圖四、CAN匯流排介面架構

4.4 CAN節點模組之樣板韌體的設計

Microchip的參考文件提供了完整的CAN Library，參考表一所列，支援使用 Standard Data Frame以及Extended Data Frame，本研究中已經撰寫CAN傳輸樣板韌體(Template firmware)，使用者只需輸入特定的暫存器，即可完成傳送和接收(圖五)，讓應用程式可以快速開發建立。

表一、CAN Library

Function	Category
CANInitialize	Configuration
CANSetOperationMode	Configuration
CANSetOperationModeNoWait	Configuration
CANSetBaudRate	Configuration
CANSetMask	Configuration
CANSetFilter	Configuration
CANSendMessage	Operation
CANReceiveMessage	Operation
CANAbortAll	Operation
CANGetTxErrorCount	Status Check
CANGetRxErrorCount	Status Check
CANIsBusOff	Status Check
CANIsTxPassive	Status Check
CANIsRxPassive	Status Check
CANIsRxReady	Status Check
CANIsTxReady	Status Check



圖五、CAN傳輸(接收&傳送) 流程圖

4.5.1 CAN OBD-II 汽車標準資料格式

汽車OBD-II為診斷汽車零件是否故障或異常，OBD-II的模式1提供車內許多感測器的目前資料，藉由OBD-II PIDs(Parameter IDs)來讀取車身功能資料。

使用時將功能PID_code暫存器輸入數值 (圖六)，例如：車速為PID:0x0D、引擎轉速為PID:0x0C、水溫為PID:0x05等。接收時汽車會回應相對應的PID功能資

料，接收的資料有些需要公式運算才能得出正確的數值資訊 (圖七)，例如引擎轉速的運算式為 $((A*256)+B)/4$ ，運算結果才為正確的數值資料，其中A和B是CAN匯流排資料格式的位元組。

另外，詢問汽車功能的CAN匯流排節點識別碼0x7DF，接收汽車OBD資料要將Acceptance Filter設為0x7E8，都是依據ISO 15765-4文件所定義的規範。

```
If ( CANIsTxReady ( ) )
{
    TX_Data_Buf2[0] = 0x02 ;
    TX_Data_Buf2[1] = 0x01 ;
    TX_Data_Buf2[2] = PID_code;
    TX_Data_Buf2[3] = 0x00 ;
    TX_Data_Buf2[4] = 0x00 ;
    TX_Data_Buf2[5] = 0x00 ;
    TX_Data_Buf2[6] = 0x00 ;
    TX_Data_Buf2[7] = 0x00 ;
    CANSendMessage ( 0x7DF , TX_Data_Buf2 , 8 , CAN_TX_PRIORITY_1 &
        CAN_TX_STD_FRAME &
        CAN_TX_NO_RTR_FRAME ) ;
}
```

圖六、節點傳送PID_code給汽車ECU

```
if ( CANIsRxReady ( ) )
{
    CANReceiveMessage( &RX_ID1 , RX_Data_Buf1 , &RX_Data_Len1 , &RX1_Message_Flag ) ;
    if (RX_ID1==0x7E8)
    {
        Engine RPM = ((A*256)+B)/4 ;
    }
}
```

圖七、接收OBD資料，以RPM為例

4.5.2 CAN節點模組之客製化韌體的設計

由於CAN的應用相當廣泛，除了常見的車用內部控制網路之外，工業自動化與醫療設備也越來越多使用CAN Bus為控制網路，使用者可藉由輸入自訂格式的韌體規劃，可以快速的建立CAN節點。

節點用途不使用於汽車時，也就不侷限於CAN OBD-II的規範內，使用者可自己定義資料格式內容，但是必須符合CAN的規範來定義。樣板韌體已經將大部分的CAN設定規劃完成，使用者只需要在暫存器內輸入數值，就可完成CAN韌體設定。

韌體裡面TX_00~ TX_07暫存器分別可以輸入0 ~ 0xFF的範圍，識別碼暫存器TX_ID1在標準模式可使用0 ~ 0x7FF，使用者在TX_00 ~ TX_07和TX_ID輸入自訂的數字，即可完成傳輸規劃的動作(圖八)。接收的部分，RX_ID1為Acceptance Filter的暫存器，可設定指定接收的識別碼，即可接收CAN資料 (圖九)。

```

If ( CANIsTxReady () )
{
    TX_Data_Buf2[0] = TX_00 ;
    TX_Data_Buf2[1] = TX_01 ;
    TX_Data_Buf2[2] = TX_02 ;
    TX_Data_Buf2[3] = TX_03 ;
    TX_Data_Buf2[4] = TX_04 ;
    TX_Data_Buf2[5] = TX_05 ;
    TX_Data_Buf2[6] = TX_06 ;
    TX_Data_Buf2[7] = TX_07 ;
    CANSendMessage (TX_ID1 , TX_Data_Buf2 , 8 , CAN_TX_PRIORITY_1 &
                    CAN_TX_STD_FRAME &
                    CAN_TX_NO_RTR_FRAME ) ;
}

```

圖八、傳送時設定資料內容與識別碼

```

if ( CANIsRxReady () )
{
    CANReceiveMessage( &RX_ID1 , RX_Data_Buf1 , &RX_Data_Len1 , &RX1_Message_Flag ) ;
    if (RX_ID1== RX_ID1)
    {
        .....
    }
}

```

圖九、接收時設定資料內容與識別碼

五、實測結果與討論

在本論文中，提出兩個運用我們所發展出的可延伸的發展平台來進行產品開發的案例，以驗證系統的可行性，並且解說了整個發展過程。一個是將市售的機車儀表增設CAN功能以應用於大型重型機車(DUCATI)之儀表板的開發，另一個是透過CAN OBD介面來實現車輛監測功能以應用於汽車儀表板的開發。

5.1機車儀表板的開發案例

首先測試車款DUCATI_MONSTER_696實驗步驟如下：

- (1)監測ECU網路：拉開原廠儀表與ECU端(圖十)，由於大型重型機車不是CAN OBD-II標準協定，所以CAN的腳位位置沒有參考資料，我們依照CAN bus的CAN_H和CAN_L有電壓差特性，利用示波器從ECU端找到CAN_H、CAN_L (表二)，這是CAN Bus的通訊腳位。將CAN匯流排介面模組連接到CAN Bus上，並且使用Kvaser Memorator搭配X-Analyzer軟體來擷取CAN Bus的傳輸訊息並顯示在電腦螢幕上(圖十一)，經過分析之後，解析到CAN Bus傳輸的所有的ID有12個分別為010、020、028、300、029、080、081、100、211、280、290、201。

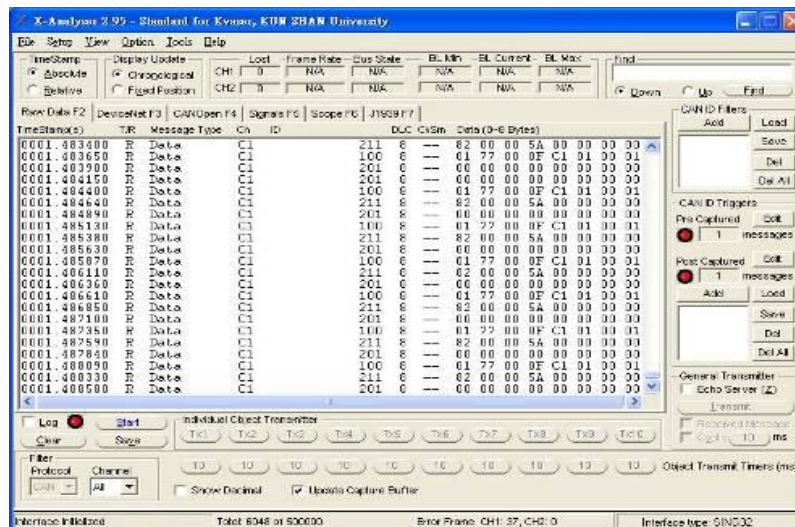


圖十、DUCATI_MONSTER_696機車儀表和ECU連接端

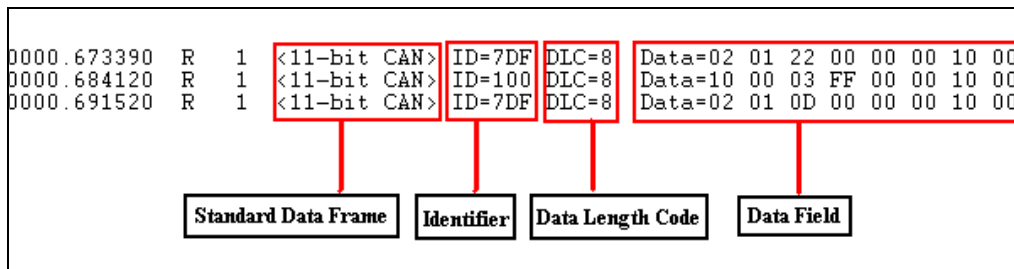
表二、DUCATI_MONSTER_696的ECU連接端腳位功能

11	12	13	14	15	16	17	18	19	20
X	右轉	左轉	前燈	MODE	引擎機油 壓力燈	左右轉 電力	遠燈	時速	防盜 IMMO
1	2	3	4	5	6	7	8	9	10
電瓶 +12V	X	電瓶 GND	時速	CAN_H	CAN_L	油量	x	Key	防盜 IMMO

(2)解析CAN資料之後，找到機車儀表相關的訊號包括了時速功能和引擎轉速功能以及空檔訊號(ID：0x100)、引擎進氣溫度和引擎機油溫度(CAN_ID：0x211)、故障燈訊息(CAN_ID：0x201)、引擎發動碼(CAN_ID：0x010、0x020、0x028、0x300、029)，而這款重型機車沒有引擎冷卻液溫度的資訊，所以將市售機車儀表的引擎冷卻液溫度位置改為顯示引擎進氣溫度的資料。



圖十一、DUCATI_MONSTER_696監測畫面



圖十二、解析CAN Bus資料

(3)在可延伸的發展平台上以樣本韌體為基礎來設計相對應的驅動程式，並且將機車CAN資料格式燒錄到單晶片微控制器中，使得發展平台具有模擬車用CAN ECU網路的節點，此模擬節點傳送著重型機車的資料格式以利後續機車儀表之CAN通訊功能的開發。



圖十三、機車儀表在開發平台中製作

(4)透過發展平台將機車儀表增加CAN通訊功能並製作完成，使機車儀表可以藉由CAN ECU傳輸資料，經過程式化運算與電路處理，顯示於機車儀表上。
 (5)將製作完成的機車儀表插入實際大型重型機車的ECU連接端，成功的顯示機車資訊。

5.2 汽車CAN OBD-II 傳輸協定測試

測試的車款為Mazda5，這台車的OBD-II資料傳輸使用CAN協定，而我們要測試的項目有2個，一項是在發展平台上透過CAN OBD-II介面來實現新增CAN匯流排節點，這個節點主要是做為監測用途，它能夠讀取並顯示指定ECU節點的資料；另一項是在汽車CAN bus中新增2個節點，一端傳輸A/D轉換訊號另一端節點接收並顯示數值範圍，測試在汽車現有的CAN bus中是否可以再增加CAN-ECU節點，其測試步驟如下：

(1)建立模擬節點與監測節點：SAE J1979定義了標準OBD-II PIDs，裡面已經說明

關於PID的相關內容與數值的範圍。我們根據標準的規範協助我們建立發展平台的模擬節點與監測節點，製作的監測節點的功能有車速(PID：0x0D)、引擎轉速(PID：0x0C)、進氣溫度(PID：0x0F)、引擎冷卻液溫度(PID：0x05)、油門位置(PID：0x11)、引擎發動時間(PID：0x1F)、引擎負載(PID：0x04)、空氣流通率(PID：0x10)、機油壓力(PID：0x0A)、進氣歧管壓力(PID：0x0B)、燃油管壓力(PID：0x22)、油量輸入(PID：0x2F)，利用LCD模組一次顯示兩種資訊並且按下按鍵後可以作翻頁的動作，所以只用一個LCD模組就可以顯示全部的功能資訊。

(2)監測節點與汽車連接測試：監測節點透過J1962診斷連接插座連接到汽車CAN OBD，但是Mazda5的OBD不支援機油壓力、燃油管壓力和油量輸入的功能，所以無法讀到該資料，其他在之前製作的功能可以成功顯示OBD資料，具備監測車身的功能。



圖十四、讀取OBD資訊並與汽車儀表板所顯示之訊息進行比對



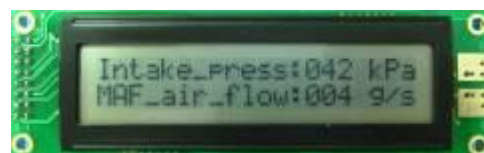
(a)讀取時速和引擎轉速



(b)讀取進氣和引擎冷卻液的溫度



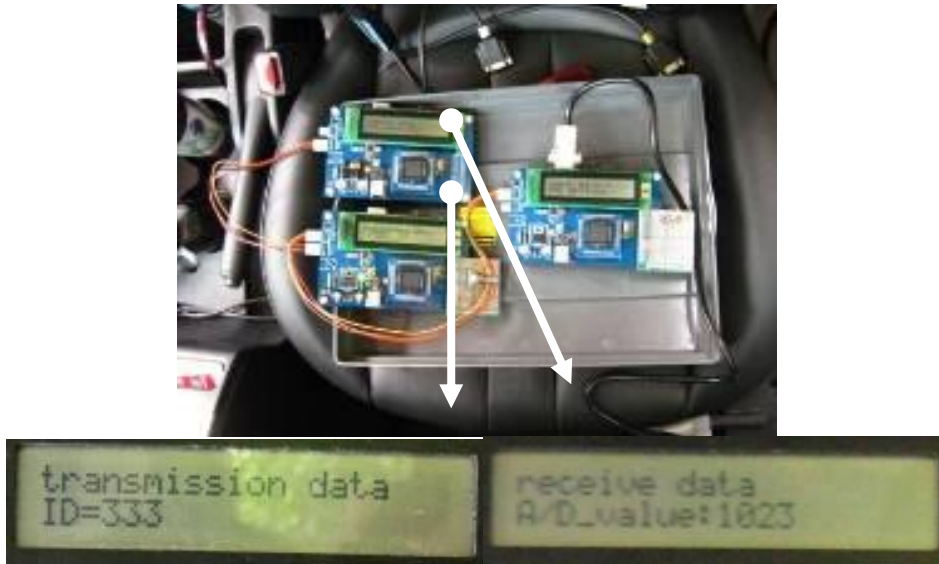
(c)油門位置和引擎啟動時間



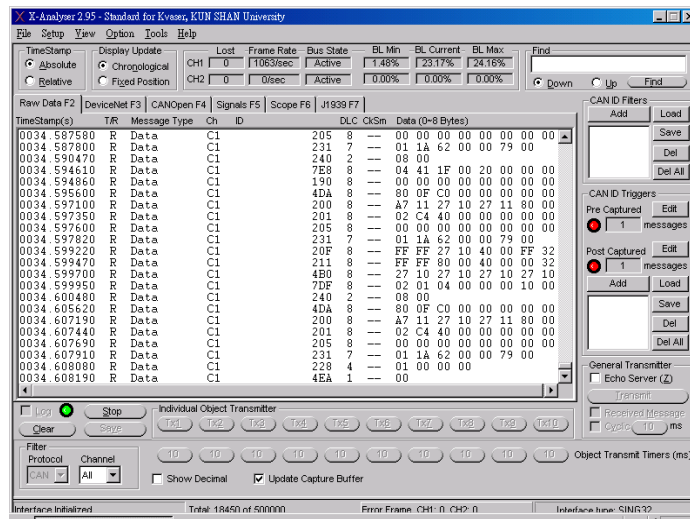
(d)進氣歧管壓力和空氣流通率

圖十五、其他讀取自OBD之訊息顯示

(3)新功能節點加入CAN bus連接測試：有2個新節點功能分別為一傳輸10 bit A/D轉換訊號與接收A/D訊號，為了避免傳輸的識別碼與原汽車的識別碼衝突，事先已利用監測工具找出汽車匯流排內流通的識別碼，我們選用識別碼0x333作為傳輸節點，而所製作的CAN-ECU節點可以在汽車裡成功傳輸與接收。



圖十六、新節點加入車用ECU網路



圖十七、監測工具顯示汽車 ECU net 的資料

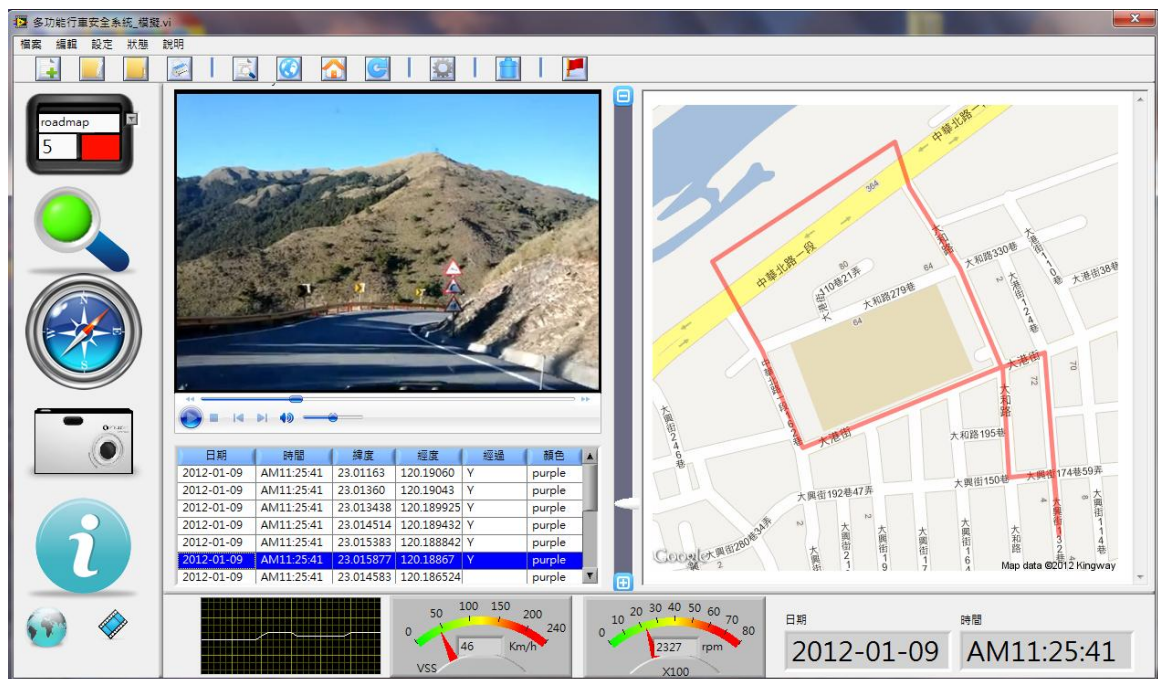
六、應用於行動裝置之多功能行車安全系統

車子的安全性是現代人購車的標準之一，除了車子所配備的安全設備外，如何降低車輛的機械故障，更是行車安全重要的一環，為了帶來更高等級的行車安全，除了平時的保養外，如果在行駛中能夠及時的發現汽車的異常，並適時的做出處置，如此一來才能使自己的生命財產安全都受到保護。

目前市售的智慧型手機幾乎都支援了，行動網路、相機鏡頭、加速度計、Bluetooth、GPS、擴充記憶體等功能，利用行動裝置內建的功能在搭配上 OBD-II 車上診斷系統，如此一來就可整合出一套多功能行車安全系統。

我們利用 Bluetooth 來發送 OBD-II 的行車數據，並在 Android 智慧型手機上設計一套 App，讓行動裝置模擬出數位錶、類比錶、三環錶、五環錶等等…，也可由使用者來自由的選擇各種錶的背景、外殼、錶圖、指針等等…，並將 GSP 資訊與以上數據一併儲存於手機中；在 PC 端軟體，可以由儲存下來的數據觀看看到使用者行車時位置與所有資訊。

我們利用 Bluetooth 來接收 OBD-II 的行車資訊，並驅動手機中的鏡頭來錄下行車影像，再搭配上加速度計與 GPS，並將以上數據與影像儲存於手機中，當意外發生時可由手機中取得當時的時間、影像、車速、位置、撞擊方向，使其成為可完整還原行車意外的行車紀錄器；在 PC 端軟體，可以由儲存下來的數據觀看到使用者行車時位置與所有資訊。



圖十八、多功能行車安全系統