

車輛增強型診斷協定之自動化測試程序開發

張慶偉¹、郭俐君²、陳旻權³

¹財團法人車輛研究測試中心

¹E-mail: chingwei@artc.org.tw

摘要

增強型診斷介面協定，為目前固有燃油車與新興電動車共同使用的診斷介面協定之一，然而診斷介面的開發與測試流程複雜繁瑣，若由開發人員自行開發、測試將曠日費時。因此，本文使用以 Vector CANoe 為主要的開發測試輔助工具，配合 Vector Diva 診斷介面測試函式庫等開發軟體，由工程人員將測試邏輯透過程式化的系統開發，實現自動化測試程序，以加速開發與驗證流程。本文中將以車輛中心自組之電動車診斷系統開發為實際案例，從初期診斷規格擬定到診斷介面測試項目的自動化程式開發流程逐一介紹，提供工程人員作為診斷功能開發與測試驗證設計之參考。

關鍵詞：增強型診斷(Enhanced Diagnostics)、開發工具(Development tool)、CANoe、Diva。

1. 前言

隨著電子產業蓬勃發展，現行車輛為達到便利性及舒適性，車用電子的使用頻率隨之亦增。為保障整車的安全性，這些附加的車用電子系統的自我診斷能力將顯得更為重要，因此隨時監控系統是否異常，降低失效情形的發生，抑或是在失效情形發生的同時適時地告知駕駛人，減少意外的發生率，維持系統的穩定性，為系統在開發後期十分重要的考量。

診斷系統最早於 1984 年加州政府首次提出針對電子化引擎隨車診斷系統之提議案，並決議自 1988 年起所有在加州販售之小型車輛及輕型卡車須加裝車輛隨車診斷系統 OBD(On-Board Diagnostic)，發展初期主要是針對車輛的排污系統，負責監控汽車引擎的運轉效能，降低因車輛引擎運轉不良或是防污染設備失效造成的車輛廢氣排放污染。然而，隨著車輛電子化的程度提高，僅針對排污控制元件為主的診斷協定逐漸不敷使用，尤其是對於近年來興起的電動車系統而言，排污系統的監控已非主流，取而代之的是監控各系統間異常狀況。

因上述之需求，增強型診斷系統(Enhanced Diagnostics)隨之應運而生，該診斷系統沿襲 OBD 的概念，而不同於遵守 ISO 15031 的排污診斷，增強型診斷系統加強對於車輛上各系統進行監控的能力，不過此部份目前仍由各車廠自行發展，不在法規強制要求範圍之內，且應用對象並不受限，從變速箱、行車電腦至倒車雷達皆可使用，因此增強型診斷介面目前尚未有統一的標準。目前增強型診斷的介面相關標準，以 ISO 11898[1]、ISO 15765[2]、ISO 14229-1[3]等相關標準架

構為主，排污診斷與增強型診斷相關標準如表 1 所示。為此，中心在自行研發的電動車同時也具備自我診斷功能，以下本文將藉由車輛中心自組之電動車增強型診斷系統開發為例，從初期診斷規格擬定到診斷介面測試項目的開發流程逐一介紹。

表 1 排污診斷與增強型診斷

OSI layers	排污診斷	增強型診斷
Diagnostic application	ISO 15031-5	User defined ISO 14229
Application layer	ISO 15031-5	ISO 15765-3
Presentation layer	ISO 15031-5	NA
Session layer	NA	ISO 15765-3
Transport layer	NA	NA
Network layer	ISO 15765-4	ISO 15765-2
Data link layer	ISO 15765-4	ISO 11898-1
Physical layer	ISO 15765-4	User defined

2. 增強型 CAN 診斷介面設計流程

為達成車輛自我診斷的功能，以便於監控車上各系統，例如：電能系統、動力系統等核心系統的即時狀態，因此各系統在運作的同時，能自行根據數據判定異常，產生相對應的診斷碼，並根據異常的嚴重程度提醒駕駛者進行異常情形的排除或停駛檢修，避免異常情形被忽略後所造成不可復原的故障或危險。因此，在電動車的架構上，診斷系統為不可或缺的一環。

一個功能完整的自我診斷系統，除系統本身的軟、硬體偵錯設計外，尚需一個標準化的介面作為診斷資訊傳輸窗口。本中心在診斷介面設計項目中從整車網路的設計端到驗證端流程，均擁有深厚的技術與設備資源，其設備皆採用德國 Vector 開發測試工具鏈，如圖 1 所示。

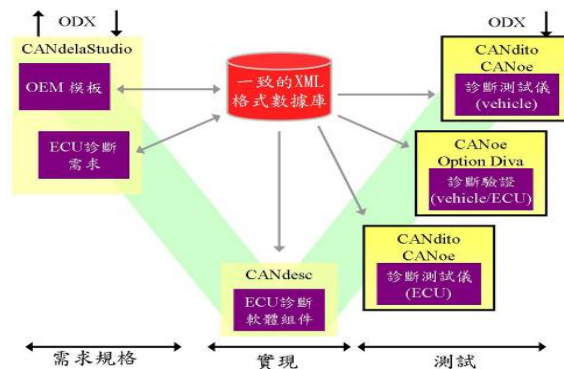


圖 1 診斷開發流程與 Vector 對應設備

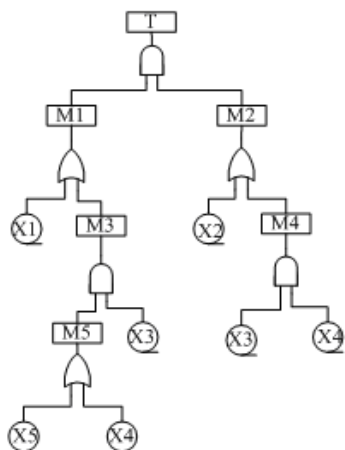


圖 2 失效樹分析

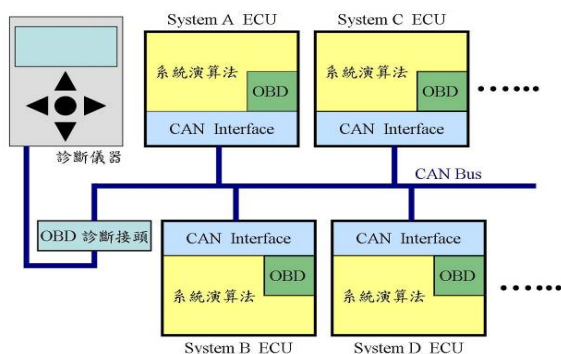


圖 3 分散式的診斷架構

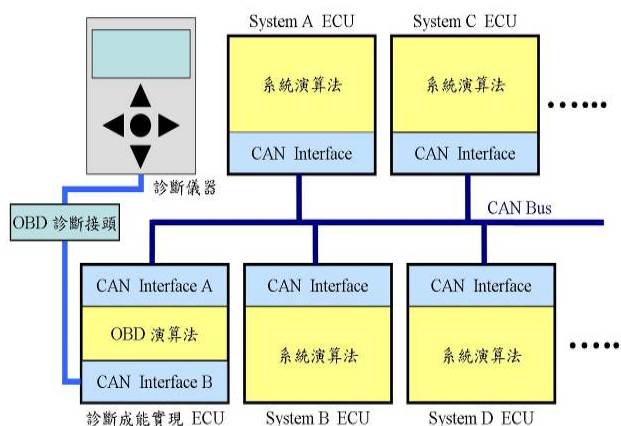


圖 4 集中式的診斷架構

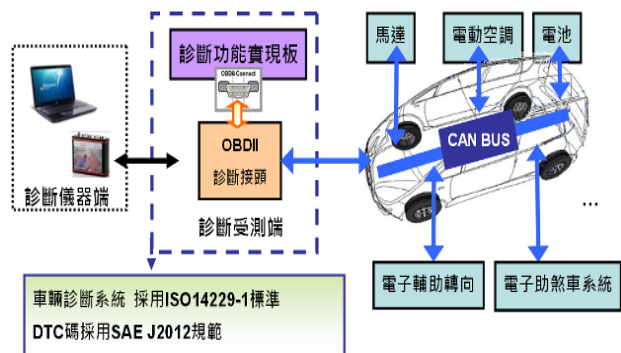


圖 5 i-EV 網路、診斷功能示意圖

首先，於制定診斷需求規格的階段中，必須針對欲設計的系統進行失效樹分析(Fault Tree Analysis)，如圖二所示，根據該系統可能出現的失效內容以及可能造成失效的原因，歸納出失效對照表，根據失效的類別給予適當的診斷服務；此時可以透過 CANdelaStudio，根據失效對照表定義 ECU 的診斷功能，並且匯出診斷數據庫，提供後續診斷開發工具使用。

為更清楚地說明診斷需求規格設計流程，以下以電能管理系統中的低壓 12V 電池模組為例。第一步先針對電能系統可能產生的失效或異常情形展開失效樹分析，根據失效或異常項目進行歸類，並推估可能的失效原因，低壓 12V 電池可能出現電壓部份的異常，如：電池訊號異常、偵測到迴路異常、電壓值異常(過高、過低)、充電異常(充電電壓過高、過低)、過度放電等可能的異常情形；在溫度部份可能出現的異常，如：無溫度訊號、溫度訊號異常(過高、過低)等，以及列出控制面板異常之情形。依據上述之異常情形，透過 CANdelaStudio 定義其診斷服務、通信參數、定時處理邏輯、DTC (Diagnostic Trouble Code)、DID(Data Identifier)等。

在實現系統的階段中，增強型診斷架構的實現方法主要可分為兩種架構：一種為分散式的診斷架構，主要應用於各系統於設計初期已將診斷模組加入考量的架構上，由於此架構能將診斷系統與介面實現於各系統上，因此分散式的架構較接近於整車廠的運作模式，各

系統分散式地獨立運作，發生異常時互不干擾，但礙於此種方法須於開發初期便須植入診斷架構，因此對於開發後期才加入診斷考量的產品來說，變更原始設計會有其困難度存在，其基本架構示意如圖 3 所示；另一種架構為集中式的增強型診斷，主要應用於研發後期避免影響其他系統的設計，所採用折衷的方法，各系統透過相對應的失效訊息告知異常的原因，並利用外接一個具備診斷介面的硬體，找出相對應的失效診斷碼，以達成診斷功能的實現，此種方式能快速整合各系統，但集中式的診斷架構將受限於轉接的中央節點，因此若中央節點發生異常將造成整車診斷異常之情形，因此採用集中式診斷架構可能會有瓶頸點的潛在問題，其基本架構示意如圖 4 所示。

中心目前研發的電動車，屬於技術整合性質，對於部分組件並不合適於後期加入自我診斷設計，故採取集中式的增強型診斷，其架構如圖 5 所示：電動車上的各系統(如：馬達、電動空調、電池…等系統)透過 CAN BUS 傳遞目前的資訊，以及新增一個診斷介面模組的節點，其採用 ISO 14229-1 的標準，且 DTC 碼的編碼方式採用 SAE J2012 的規範，該模組的目的為接收各系統送出的錯誤訊息，並根據事先訂定相對應的 DTC 碼進行判讀與比對，如此可減少各系統在開發後期變更設計所造成時間上的浪費。

3. 增強型診斷介面協定測試

一般整車廠在增強型診斷系統的開發流程中，如同上述制定診斷需求規格階段，整車廠將定義出診斷服務、通信參數、定時處理邏輯、DTC、DID等，並將其擬定的規格統整成為正式規格文件，轉交於供應商進行系統元件的軟硬體開發，待完成後再由整車廠依據先前定義的規格文件，擬定一套測試驗證的測試規範，內容包含測試流程與標準，驗證供應商提供之系統元件是否有達到原先規定的標準。

3.1 診斷協定測試項目

就系統應用端來看，目前CAN Bus尚未擁有測試驗證的相關國際規範，各大整車廠大多依據底層的網路通訊標準，自行發展完整的上層協定與後端的測試驗證規範。依據ISO定義的分層概念區分為數個測試的分項，一般網路測試項目可分為實體層測試(Physical)、資料連結層測試(Data Link)、應用層(Application)，而應用層中又可分為網路管理測試(Network Management)、交互層測試(Interaction Layer)、診斷介面測試(Diagnostic Software Modules)、應用程式模組測試(Application Software Modules)，同時CAN網路相關通訊容錯機制也常被整車廠納入測試項目之中，如圖6所示[4]。其中診斷介面測試主要為依據ISO15765、ISO14229的國際標準規範，針對網路層、應用層、診斷服務應用進行測試，如圖7所示。

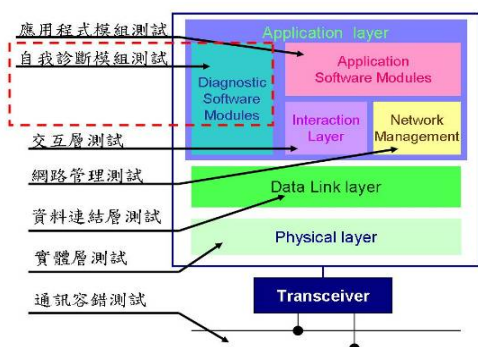


圖6 CAN節點構成要素與相對應的測試項目

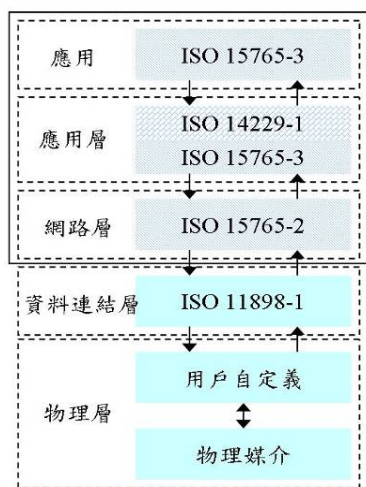


圖7 診斷體系結構

網路層的測試主要在於驗證ECU能否正常收發連續CAN Message，主要的項目有網路層協定控制資訊、定時參數、多筆Message發送；如表2中規定控制資訊結構，圖8定時參數時間位置。

應用層的測試項目則包含診斷請求Message格式、資料內容、應用測試、會話模式、安全模式、功能定址...等；診斷請求Message格式是針對請求Message長度、請求Message格式、組合DID測試與禁止肯定回應的位元測試；資料內容則是對於發送無效資料的請求Message，驗證ECU是否正確處置；而應用測試則是發送有效或無效的請求Message，驗證ECU收到有效請求時是否發送正確的肯定回應，並按照接收到的請求做出正確的回應動作，反之收到無效請求時，驗證ECU是否發送相應的否定回應；而會話模式是驗證會話模式下的模式切換與訪問被拒絕的狀態處置；安全模式則是驗證安全訪問、請求順序錯誤、安全狀態不正確、ECU復位與無效密碼下的狀態處置；功能定址則有簡單請求、禁止肯定回應位元、模式或安全狀態不正確、無效診斷服務、無效子功能與無效參數的測試子項目；最後基本測試則是針對基本服務測試、無效子功能、無效診斷服務與無效DID進行驗證測試。

表2 網路層協定控制資訊(N_PCI)結構

N_PCI 類型	N_PCI			
	Byte 1		Byte 2	Byte 3
	Bit7~4	Bit3~0		
單筆(SF)	0	DL<=7	N_Data	
第一筆(FF)	1	7<DL<=4095		N_Data
連續封包(CF)	2	SN	N_Data	
控制封包(FC)	3	FS	BS	STmin

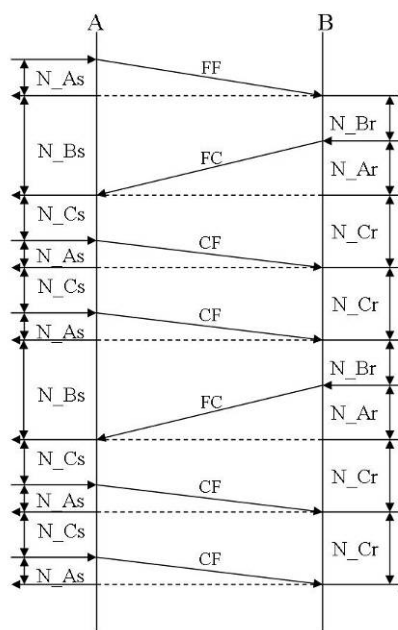


圖8 定時參數時間

上述的測試項目內容除了依據 ISO 標準的基本診斷功能的正向測試驗證外，還需進行反向測試驗證，如停止、重複、延遲發送某筆連續封包的錯誤狀態測試，因此網路層與應用層須進行的測試項目高達四百多項。以網路層的測試為例，若測試工具發送一筆 ECU 支援的連續封包請求 Message 給 ECU 端，所需進行的測試分項如表 3 中所示。

有鑑於上述眾多的測試項目，若全程由工程人員逐項進行測試驗證，將需花費眾多人力與時間，因此將測試流程導入自動化測試有其必要性，一般 CAN 的自動化測試程式是利用在 Vector 設備中提供類似 C 語言的 CAPL(CAN Access Programming Language)程式，透過程式的撰寫可將測試規範中的測試項目撰寫成自動化測試程式，並且同時紀錄一些與測試項目相關的描述內容，以增加該測試項目的可讀性。由於診斷測試項目為數眾多，如全由工程人員透過 CAPL 程式進行撰寫，除了需花費眾多人力與時間外，後續程式的維護也較不易進行，因此 Vector 提供 Diva 診斷功能測試函式庫產生自動化測試程式，縮短工程人員在診斷協定之自動化測試程序開發時程，以下將透過實際案例，介紹透過 Diva 實現自動化測試程序開發。

3.2 應用 CANDiva 自動生成自動化測試程序

CANDiva 全名為 CANoe Option Diagnostic Integration and Validation Assistant(以下簡稱 Diva) [5]，可根據 CANdelaStudio 輸出的診斷數據庫，自動產生 CANoe 診斷測試程式，執行測試項目並彙整出測試報告，此外，還可通過開發人員自定義的測試模式進行擴展，使測試範圍配置能輕易達成。

表 3 測試分項

正向測試
ECU 發送 FF 回應 Message 的 DL
功能定址請求 Message 的 FF
控制封包的狀態為 OVFLW 處理機制
控制封包的狀態為等待的處理機制
控制封包 BS 滿足規範要求
控制封包的 STmin 滿足規範要求
N_As、N_Bs、N-Cs 的時間參數
N_Data 資料內容
反向測試
不正確 DLC 與 SF_DL 處理機制
無效狀態下控制封包處理
長度不正確下控制封包處理
停止發送後連續封包
重複發送某筆連續封包
延遲發送某筆連續封包
不發送控制封包
重複發送控制封包
延遲發送控制封包

以中心系統為例，在開發的先期已透過 CANdelaStudio 將測試規範轉換成診斷介面數據庫，此時只需透過 Diva 匯入數據庫後，並根據需求進行測試配置，視其測試規格設定主要可分為 General、Intensity、Aspects、Services 等四項，如圖 9 為 Services 中 Data customization 的設定分頁，此分頁主要為針對需要測試的診斷服務進行勾選。待配置後即可由 Diva 自動編譯匯出自動測試程式，內容包含*.html 的診斷測試規範、CAPL 測試程式與 CANoe 測試環境檔，其診斷測試規範大致分為 Format、DataContent、Application、Session、General、Functional Addressing、Transport Layer 七項；其中，CAPL 測試程式可供測試人員針對特別的測試項目進行修改撰寫，CAPL 測試程式將依照測試的項目分別於不同的 Test_case 中呈現。

將 Diva 自動產生測試程式匯入 CANoe，此時透過生成的虛擬測試節點開啟欲測試的項目列表，項目內容對應診斷測試規範中的七項內容，如圖 10 所示，測試人員可於列表中勾選欲測試的項目，並於點擊 Test 視窗中的 Start 鍵後，CANoe 依序列執行勾選的測試項目，並於測試完後自動生成測試報告，如圖 11 所示。

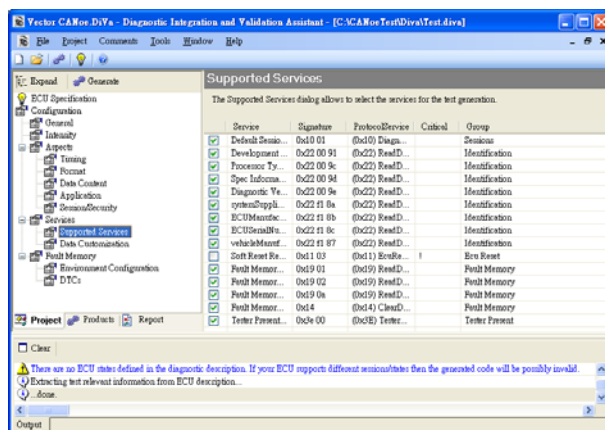


圖 9 Services 中 Data customization

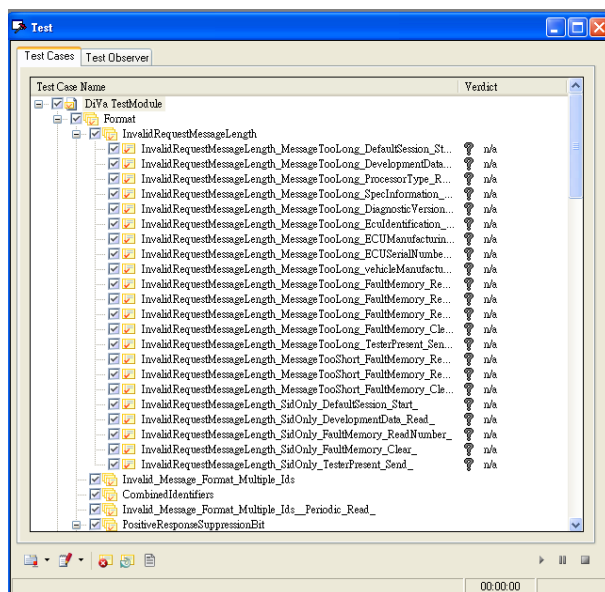


圖 10 欲測試的項目列表

4. 結論

因車輛電子的蓬勃發展，眾多的ECU等控制元件為保障整車的安全性，車用電子系統自我診斷能力的重要性可想而知，因此每個控制元件在安裝於實車之前，必須依循固定、嚴謹的測試流程進行驗證。其中診斷介面的測試攸關診斷功能的可靠度，然而這些測試分項約有數百項，而一台車上又有數十個系統，龐大的測試分項，若以人工進行測試將需花費漫長的時間。因此建立一套自動化的車輛增強型診斷驗證流程是有必要性的。

而本文依據車輛增強型診斷介面的開發流程，由最初的需求規格、診斷功能的實現，到後期的測試驗證逐一介紹目前車輛中心發展近況。在測試驗證中敘述診斷協定的測試項目數量之龐大，因此建議應用 Vector 診斷介面測試函式庫 Diva，可大幅減低開發人員在CAPL自動化測試程序的開發負擔，如此將可快速並準確地完成自動化測試程序的開發。後續將可透過自動化測試程序應用，迅速完成診斷協定的測試驗證，並在測試完成後自行產生一份完整的測試報告，如此將可降低測試人員所需花費的龐大時間與人力。

5. 致謝

研究承蒙經濟部技術處科專計畫支持(計畫編號：100-EC-17-A-04-02-0889)，才足以順利完成，在此至上誠摯的謝意。

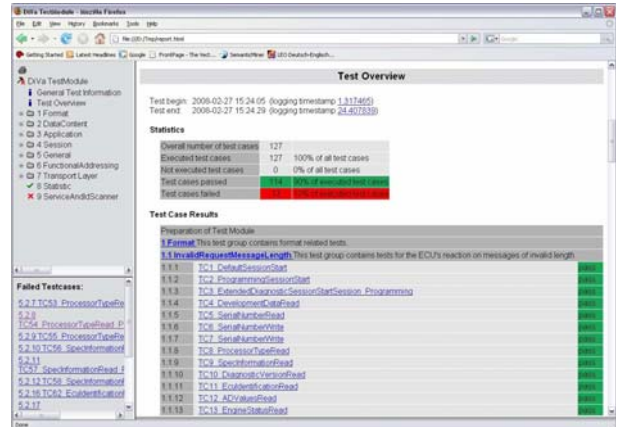


圖 11 測試報告

6. 參考文獻

期刊論文：

- [1] ISO 11898, "Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication", 1993.
- [2] ISO 15765, "Road vehicles -- Diagnostics on Controller Area Networks (CAN)", 2004.
- [3] ISO 14229, "Road vehicles -- Unified diagnostic services (UDS)", 2006.
- [4] 張慶偉、陳旻謹，CAN網路之CAPL自動化驗證程序設計技術，第十五屆車輛工程學術研討會，2010。
- [5] https://www.vector.com/vi_canoediva_en.html, Vector CANDiva User Manual.