

汽車開放式系統架構

大型資料通訊模組軟體的設計與實作

Design and Implementation of AUTOSAR Large Data

Com Module Software

指導教授：張大緯 教授

專題成員：林祐儀、羅偉安

開發工具：C、CS+ for CC、RH850/F1H

測試環境：Windows11、CS+ for CC

一、簡介：

所謂 AUTOSAR (Automotive Open System Architecture) 是一套車用軟體標準，其設立標準的主要目的即為提高軟體使用的可重複性。而在 AUTOSAR CP

(Classic Platform) 中可分為以下三個部分：應用層 (Application Layer)、運行環境層 (Runtime environment Layer) 以及基礎軟體層 (Basic Software Layer)。在各層中，不同模組的規格書則定義了該模組的行為及與其他模組間的互動。AUTOSAR 將功能切分成不同的模組，以利開發者更好的進行系統維護與更新。

而本專題即是基於 AUTOSAR 規範的 OS 上去實作 LdCom (Large Data Communication) 模組，最終目的是使其能按照 AUTOSAR 規範下來正常運行，內容則包含整個模組的軟體設計及實作，與 Com 模組不同的是 LdCom 只處理大型資料，也就是一個 IPDU (Interaction Layer Protocol Data Unit) 內部只會有一個 Signal，並且不需要額外對 Signal 進行處理。

以下為系統架構圖：

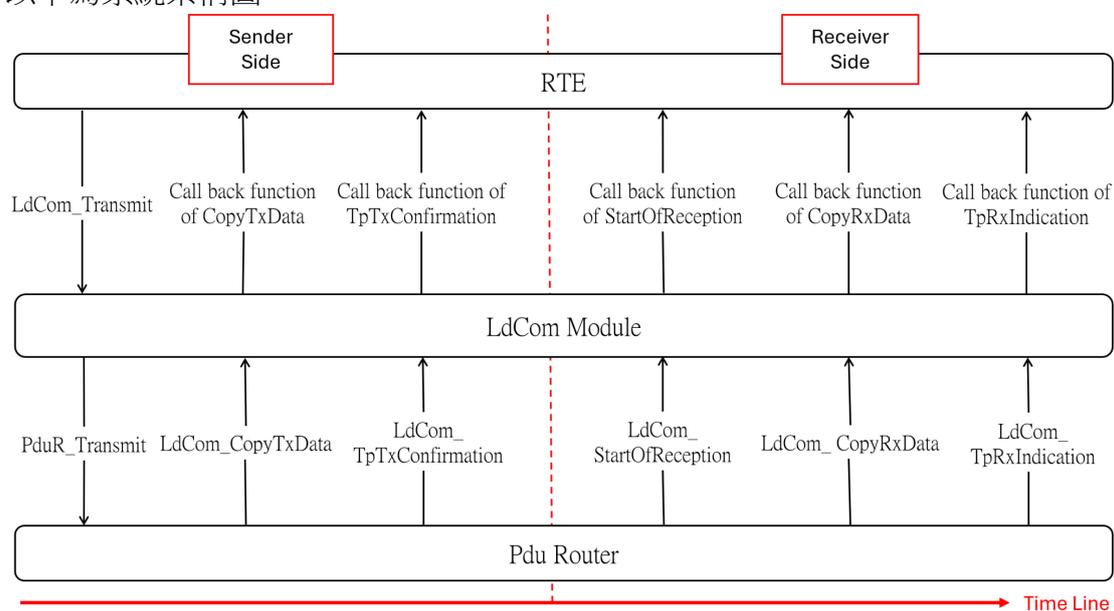


Figure 1 : LdCom 模組傳送與接收架構流程圖

由於 LdCom 在 Specification 中的定義為非自發性模組，這代表說如果要使用 LdCom 提供的 Api，需要由 RTE (LdCom 的上層) 或者 Pdu Router (LdCom 的下層) 來觸發。Sender Side 由 RTE 呼叫 LdCom_Transmit 進行傳輸，LdCom 在收到 IPDU 後會呼叫 PduR_Transmit，並繼續向下傳遞，後續便等待 Pdu Router 呼叫

LdCom 來完成傳往下層的工作。

而 Receiver Side 則是由 Pdu Router 呼叫 LdCom_StartOfReception 來開始往上傳遞 IPDU，並依序呼叫剩餘的 function 來完成傳往上層的傳遞工作。

二、測試結果：

a. 步驟 main_test.c / LdCom.c / up_Rte.c / LdCom_PbCfg.c / PduR_mock.c

(1) 撰寫 main_test.c / LdCom.c / up_Rte.c / PduR_mock.c 檔

(2) 設定好 LdCom_PbCfg.c 中的 IPDU 與 Callback Mapping 結構

(3) 將所有.c 檔 compile 並執行

b. 傳送執行結果和接收執行結果

```
{
    .LdComCopyTxDataCbk = up_Rte_LdComCbCopyTxData,
    .LdComTpTxConfirmationCbk = up_Rte_LdComCbTpTxConfirmation,
    .LdComTxConfirmationCbk = NULL,
    .LdComRxStartOfReceptionCbk = NULL,
    .LdComCopyRxDataCbk = NULL,
    .LdComTpRxIndicationCbk = NULL,
    .LdComRxIndicationCbk = NULL,
    .LdComTxTriggerTransmitCbk = NULL,
    .LdComPdurPduId = 0,
    .LdComHandleId = 0,
    .LdComLowerApi = LDCOM_TP,
    .LdComIPduDir = LDCOM_SEND,
    .LdComPduRef = &MyIPdu[0],
},
```

Figure 2：IPDU 與 Callback Mapping 設定

```
LdCom_Transmit(2, &PduInfo);
LdCom_Transmit(2, &PduInfo);
LdCom_Transmit(2, &PduInfo);
```

Figure 3：呼叫 LdCom_Transmit（3 次為一個 interrupt）

```
PduR_LdComTransmit called with ID: 0, Length: 8
This is Copy Tx Data CALL BACK!!
info->SduLength: 8
info->SduDataPtr:
12 34 56 78 9A BC DE F0
CopyTxData status: 0, available: 0
This is Tp_Tx_Confirmation CALL BACK!!
CanIfRxBuffer get: 12 34 56 78 9a bc de f0
```

Figure 4：Tx Test result in Putty

```
This is Start_Of_Reception CALL BACK!!
This is Copy_Rx_Data CALL BACK!!
info->SduLength: 8
info->SduDataPtr:
F0 DE BC 9A 78 56 34 12
CopyRxData status: 0, bufferSizePtr: 0
This is Tp Rx Indication CALL BACK!!
```

Figure 5：Rx Test result in Putty