

Intrepid Control Systems, Inc.

アプリケーションノート

ISO15765-2 メッセージ
送受信 基礎編



目次

1. 概要	3
2. ISO15765-2 プロトコル	3
2.1. Multiple frame transmission	3
2.2. Addressing Format	4
2.2.1. Normal addressing	4
2.2.2. Extended addressing	5
2.2.3. Mixed addressing	5
3. Vehicle Spy3 設定	6
3.1. テストシステム構成	6
3.2. Normal addressing	6
3.3. Extended addressing	13
4. Vehicle Spy3 サンプルプログラム	16
5. その他	17
6. 変更履歴	17
7. 用語一覧	17

1. 概要

本資料は、当社 Vehicle Spy3 を使用して ISO15765-2 に基づいた通信を行うための解説資料です。特に CAN のマルチフレーム通信を行う場合の Vehicle Spy3 の操作方法について解説します。

本資料は Vehicle Spy3 の基本操作及び ISO15765-2 のプロトコルが理解できている方を対象としており、事前に以下の解説ビデオに目を通されていることを前提と致します。

<http://www.intrepidcs.jp/vehicle-spy-3-3529935500124991248712458.html>
Sending and Receiving ISO15765-2 Messages (4:46) (日本語版)

2. ISO15765-2プロトコル

本章では CAN のマルチフレーム転送に必要な ISO15765-2 プロトコル概要を解説致します。

2.1. Multiple frame transmission

1. 一般に CAN 通信で長いメッセージを送る場合は、セグメント化を行い複数のメッセージに分けて通信が行われます。一般的な通信手順は以下のようになります。

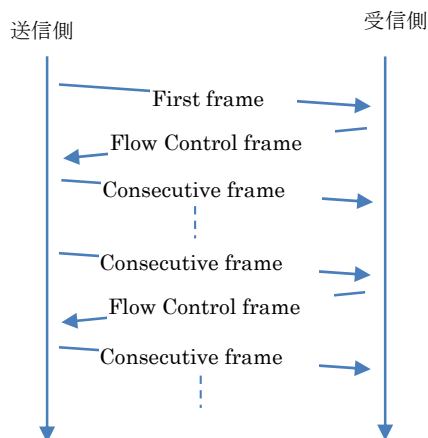


図 2.1.1

2. ここで、受信側ノードの処理能力により受信処理が間に合わない場合があります。このような場合、受信側では送信側に対して下図に示すような送信メッセージの送信間隔や送信メッセージ数等に制限をかけることができます。

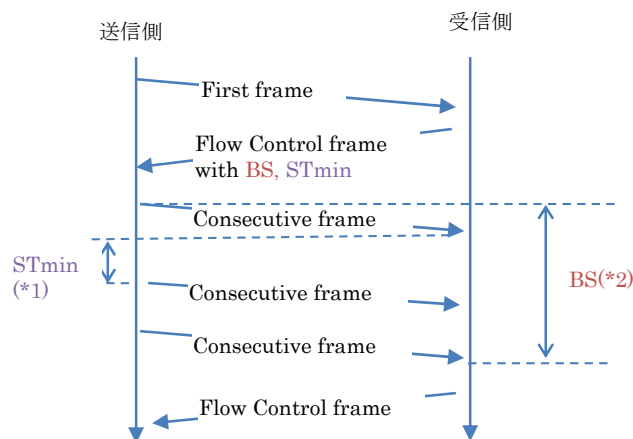


図 2.1.2

注記

(*1): Separation TimeMin (この間、送信側は送信を待たなければなりません)

(*2): Block Size(受信側で確認する必要が無い状態で、送信側から送られる最大数)

2.2. Addressing Format

2.2.1. Normal addressing

1. 以下の表はデータリンク層の Normal addressing モードにおけるデータ構成です。

N_PDU(*1) タイプ	CAN ID	CAN フレームデータフィールド							
		B1	B2	B3	B4	B5	B6	B7	B8
SF(*2)	(*6)	N_PCI(*7)	Data (実際の送信データ)						
FF(*3)	(*6)	N_PCI(*7)	N_PCI(*7)	Data (実際の送信データ)					
CF(*4)	(*6)	N_PCI(*7)	Data (実際の送信データ)						
FC(*5)	(*6)	N_PCI(*7)	N_PCI(*7)	N_PCI(*7)	-				

表 2.2.1.1

注記)

- (*1): Network Protocol Data Unit
- (*2): Single Flame
- (*3): First Frame
- (*4): Consecutive Frame
- (*5): Flow Control
- (*6): 実際の送受信 ID
- (*7): Network Protocol Control Information

以下に N_PCI の詳細を示します。

N_PDU Name	B1	B2	B3
SF	Bit7-4 Bit3-0 0000 SF_DL	-	-
FF	Bit7-4 Bit3-0 0001 FF_DL	Bit7-0 FF_DL	-
CF	Bit7-4 Bit3-0 0010 SN	-	-
FC	Bit7-4 Bit3-0 0011 FS	BS	STmin

表 2.2.1.2

SF_DL: Single Frame Data Length
 FF_DL: First Frame Data Length
 SN: Sequence Number
 FS: Flow Status

- 3. Normal fixed addressing
 29 ビット CAN ID 時に使用されます。本資料では説明を省きます。
- 4. Normal addressing での通信例
 以下のデータを設定して、送信側から 1 回データを送信した時のフローを示します。
 送信 ID: 0x54
 フロー制御用の ID: 0x765
 送信データ: 0x00,0x01,0x02, , , 0x25 (26 バイト分)

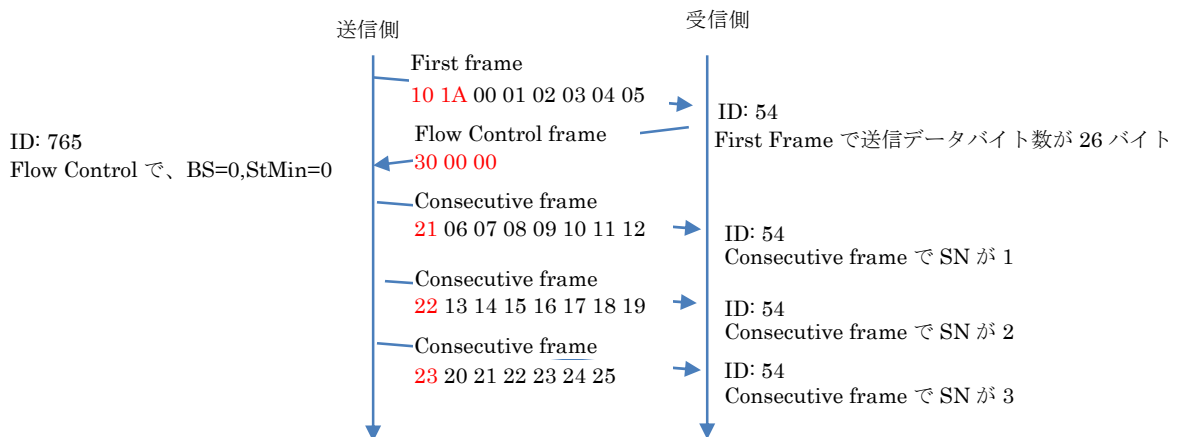


図 2.2.1.1

2.2.2. Extended addressing

- Extended addressing モードは“2.2.1 項 Normal addressing”に非常良く似ていますが、以下に示すように CAN フレームデータフィールドの Byte1 に N_TA が挿入されます。

N_PDU タイプ	CAN ID	CAN フレームデータフィールド								
		B1	B2	B3	B4	B5	B6	B7	B8	
SF		N_TA(*1)	N_PCI	Data (実際の送信データ)						
FF		N_TA(*1)	N_PCI	N_PCI	Data (実際の送信データ)					
CF		N_TA(*1)	N_PCI	Data (実際の送信データ)						
FC		N_TA(*1)	N_PCI	N_PCI	N_PCI	-				

表 2.2.2.1

注記)

(*1): Network Target Address

- Extended addressing での通信例
 以下のデータを設定して、送信側から 1 回データを送信した時のフローを示します。
 送信 ID: 0x54
 フロー制御用の ID: 0x765
 送信データ : 0x00,0x01,0x02, , , 0x25 (26 バイト分)

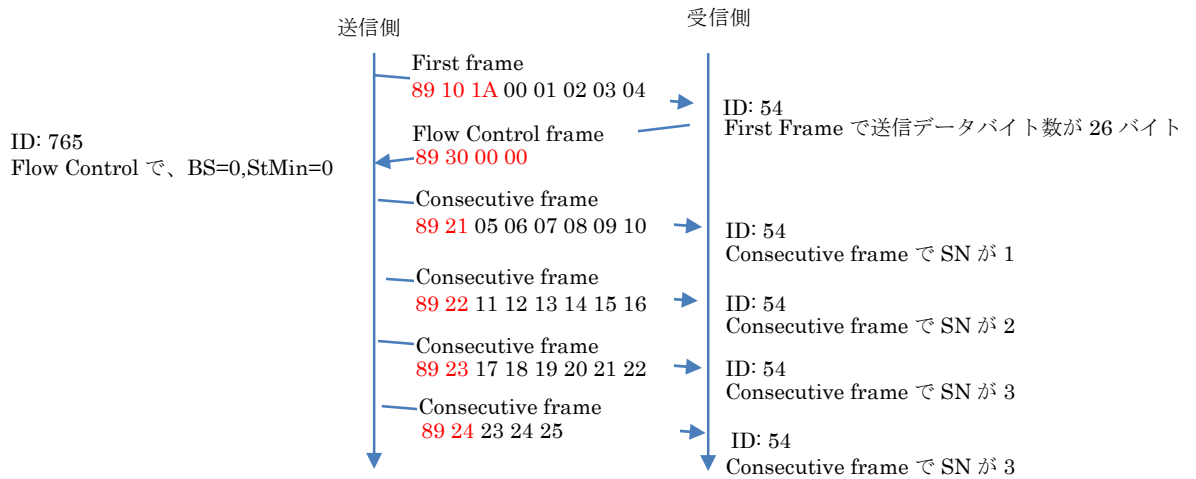


図 2.2.2.1

注記)

N_TA: 89 がすべてのフレームの 1 バイト目に入っている。

2.2.3. Mixed addressing

Mixed addressing モードはメッセージタイプがリモートダイアグノスティックスに設定された時にのみ使用されます。本資料では説明を省きます。

3. Vehicle Spy3設定

本章では、Vehicle Spy3 を使用して ISO15765-2 メッセージ通信を行う際の設定方法について解説致します。

3.1. テストシステム構成

以下に今回のテストで使用するシステム構成を示します。

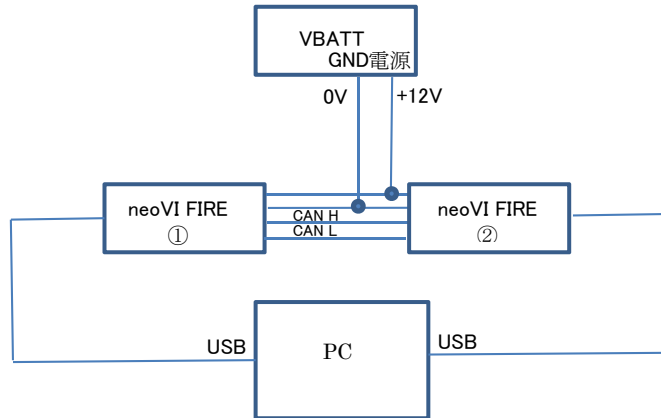


図 3.1.1

3.2. Normal addressing

- Vehicle Spy3 を起動して接続されている FIRE (neoVI FIRE①) を選択します。今回は neoVI FIRE①を neoVI FIRE 54483 とし、neoVI FIRE②を neoVI FIRE 52295 とします。

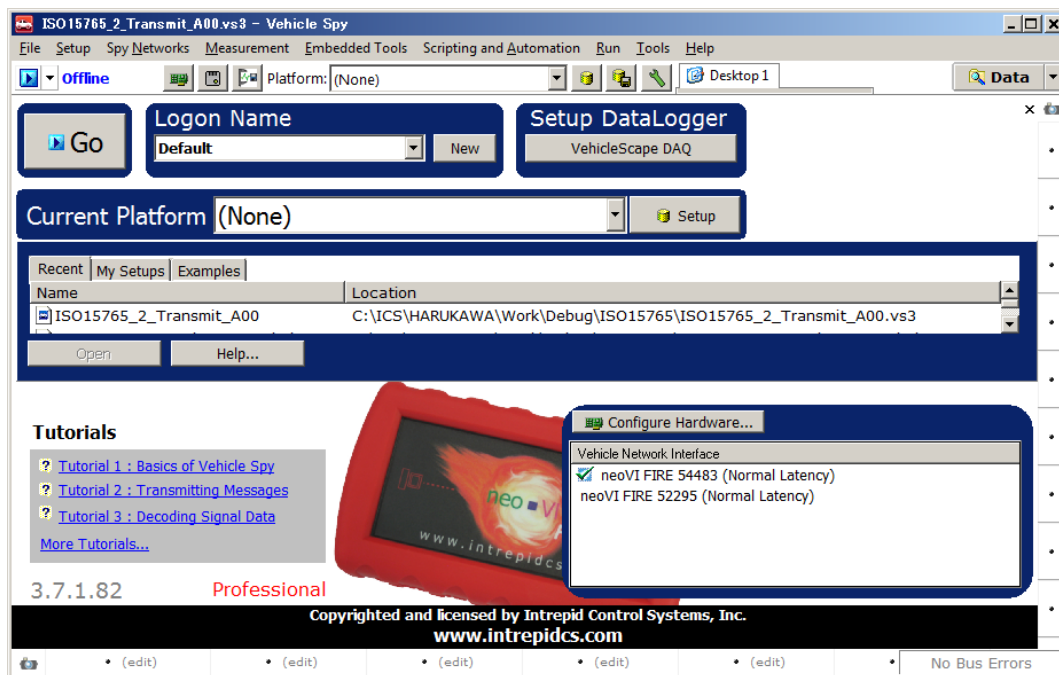


図 3.2.1

6. Message Editor を開き Transmit 画面で送信メッセージを 1 個作成します。この時、Multi の部分は ISO15765-2 を選択します。今回の例では以下に示すように、ID0x54 としています。

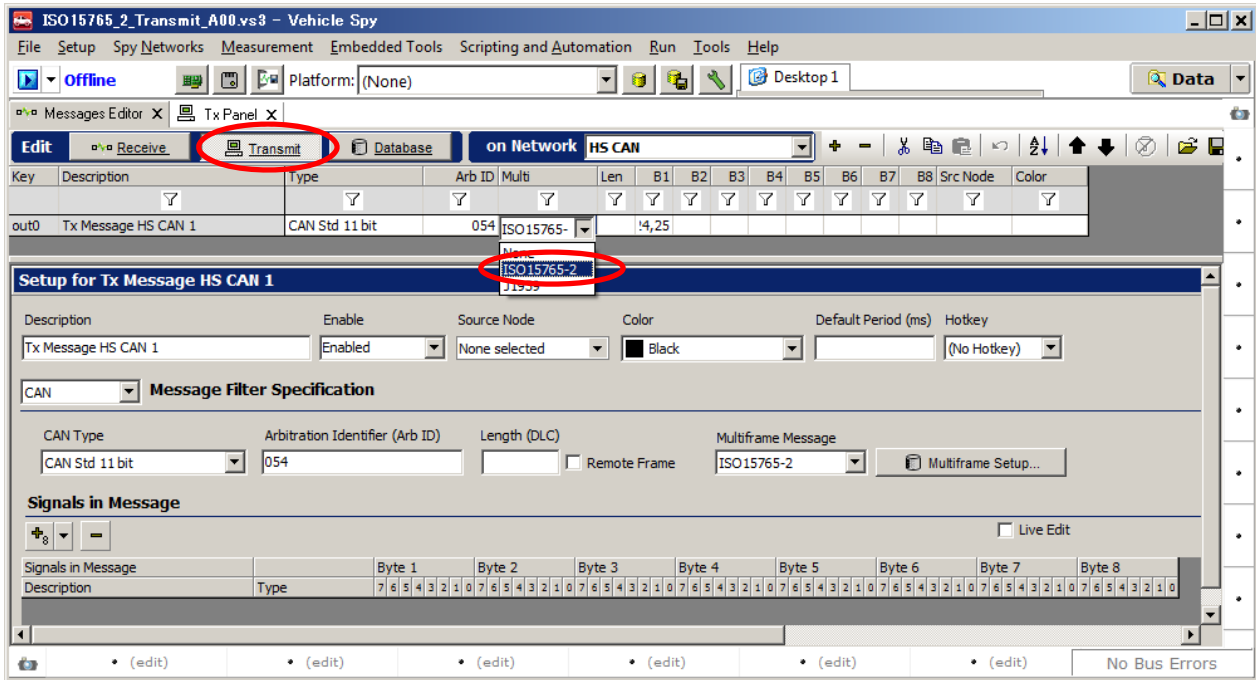


図 3.2.2

7. B1 部分をクリックして、データの入力を行います。今回は適当な値 00-0x25 まで 26 バイトのデータを入力します。

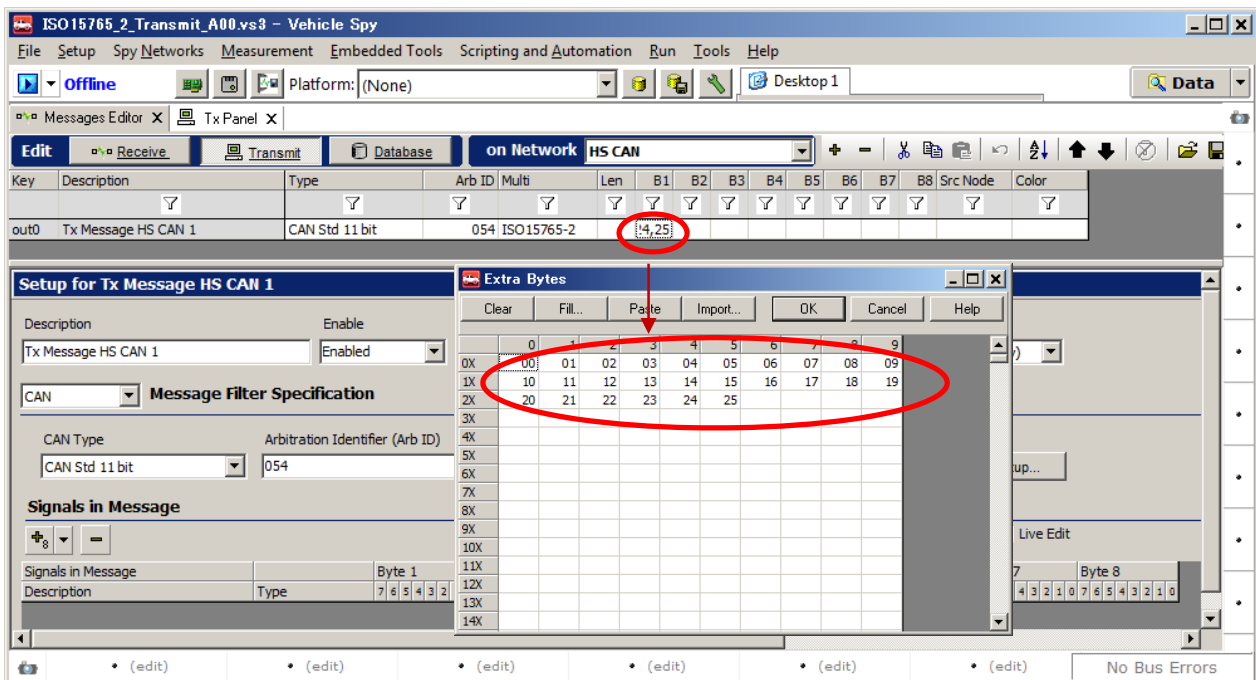


図 3.2.3

8. Multiframe Setup...を選択します。

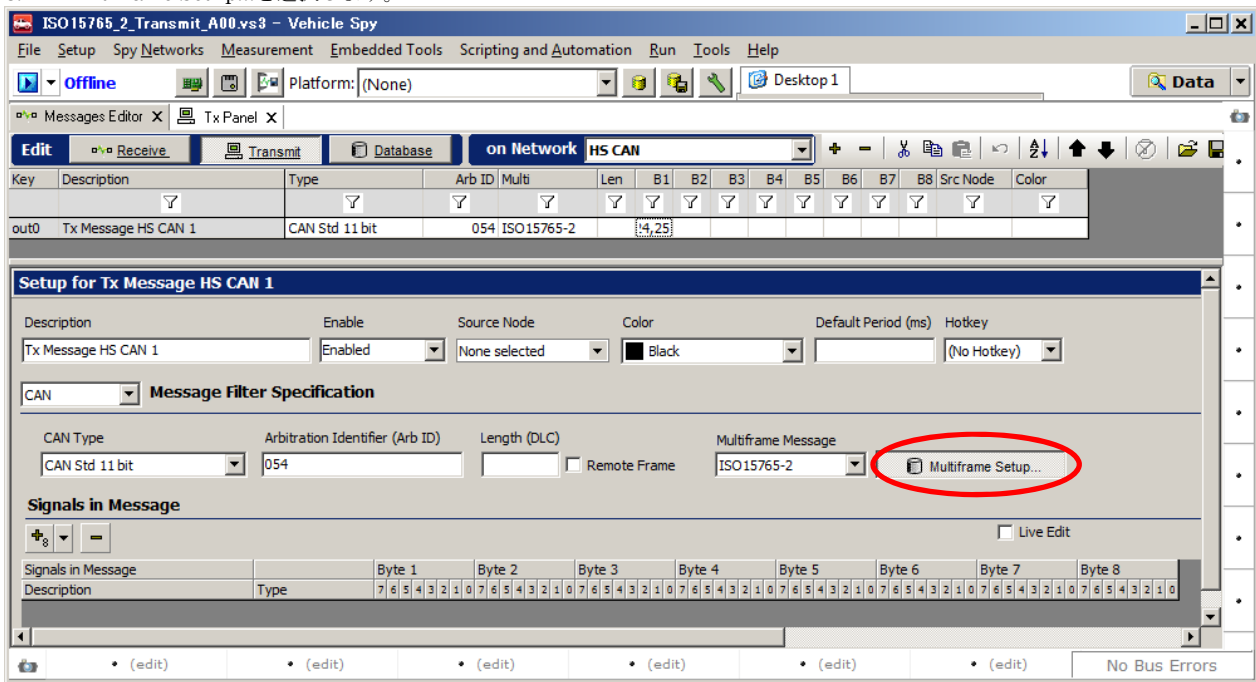


図 3.2.4

9. Transmitted Message, Flow Control Filter, Timeouts を設定します。今回は Normal addressing モードで使用する為、デフォルトのまま使用します。

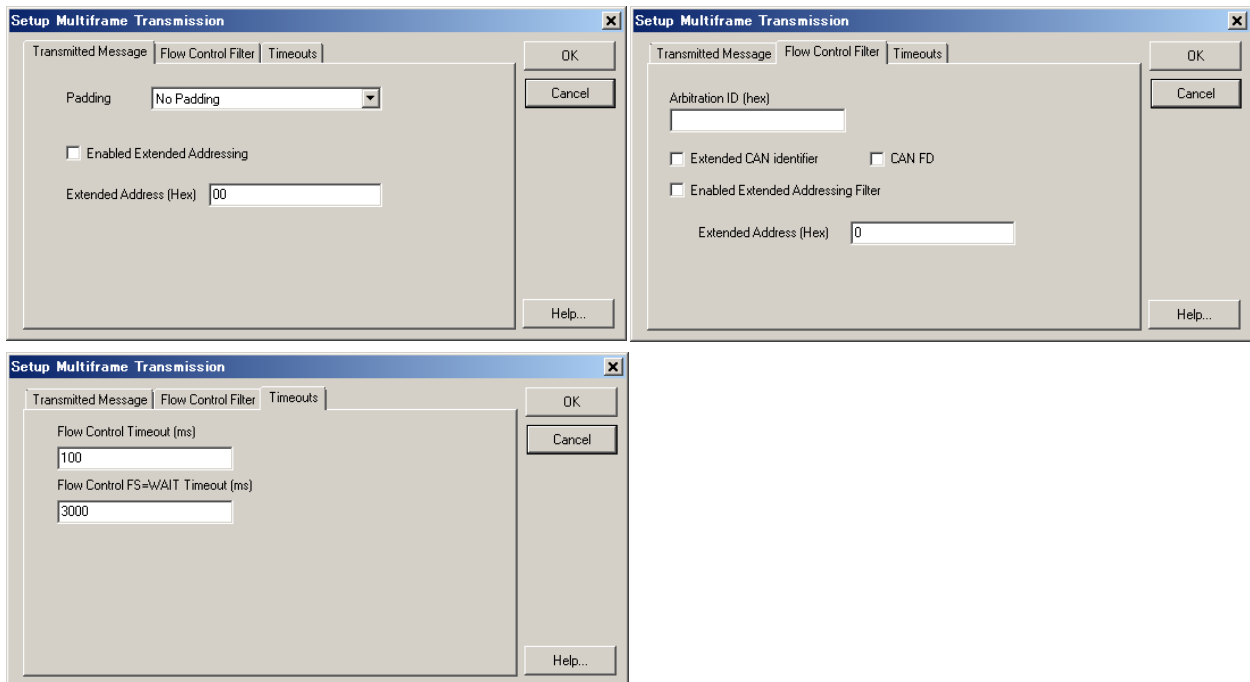


図 3.2.5

10. “Flow Control Filter”タブの”Arbitration ID (hex)”が空欄の場合は、受信側で設定された Arbitration ID が使用されます。 Arbitration ID を特定したい場合は”Arbitration ID (hex)”欄に設定します。

注記) neoVI FIRE/RED のデバイスをスタンドアロンで使用する場合は、”Arbitration ID (hex)”及び”CAN Type”の指定は必須となります。以下に参考例をしめします。

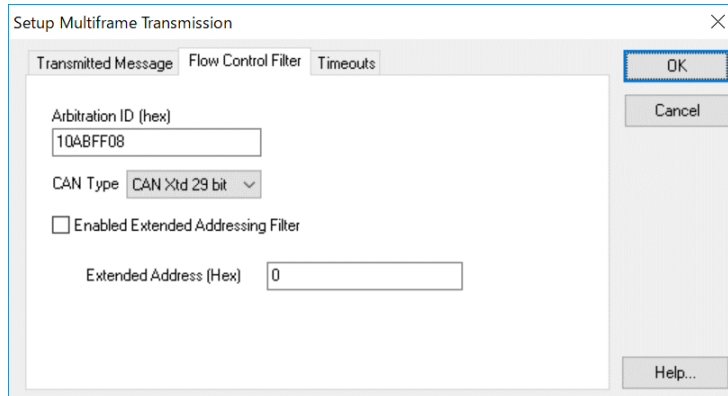


図 3.2.6

11. Messages と Tx Panel を以下のように表示させ、Online モードにします。これで、neoVI FIRE①から ISO15765-2 メッセージが送信できるようになります。

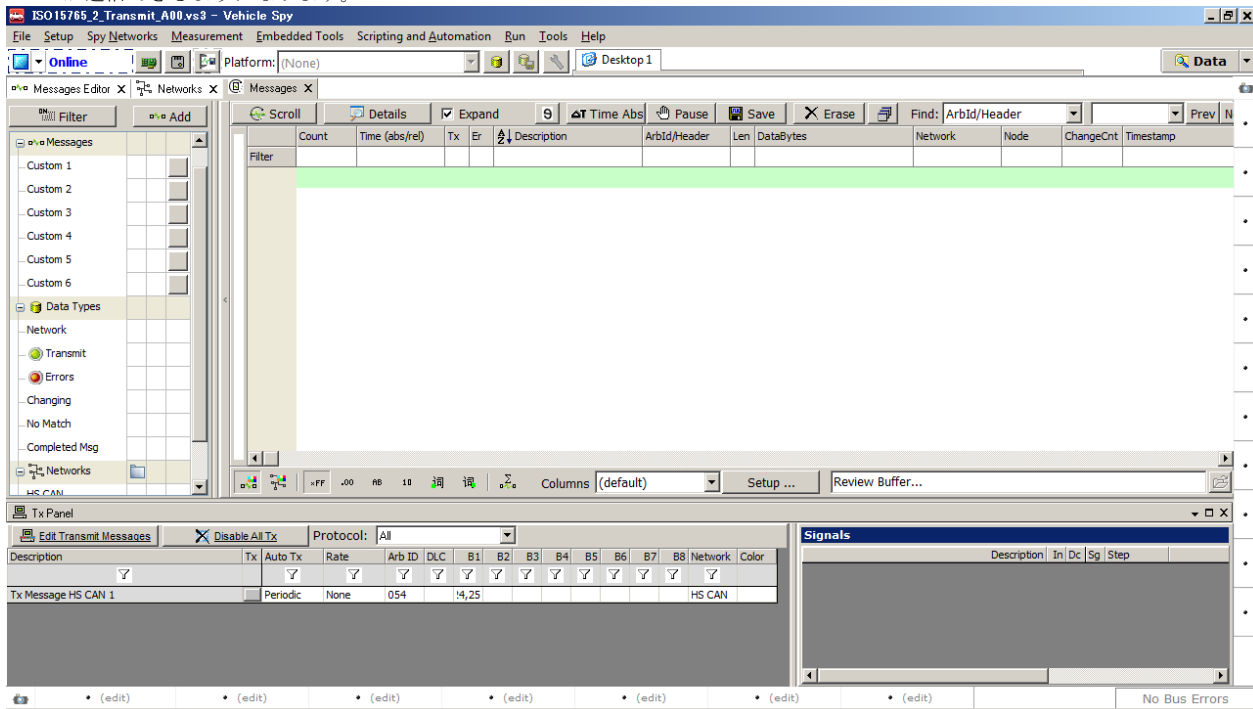


図 3.2.7

12. 今度は、受信側 neoVI FIRE②の設定を行います。Vehicle Spy3 をもう一つ起動させ以下の設定を行います。Message Editor を開き Receive 画面で受信メッセージを 1 個作成します。この時、Multi の部分は ISO15765-2 を選択します。今回の例では以下に示すように、先に設定した送信メッセージを受けるために ID0x54 とします。

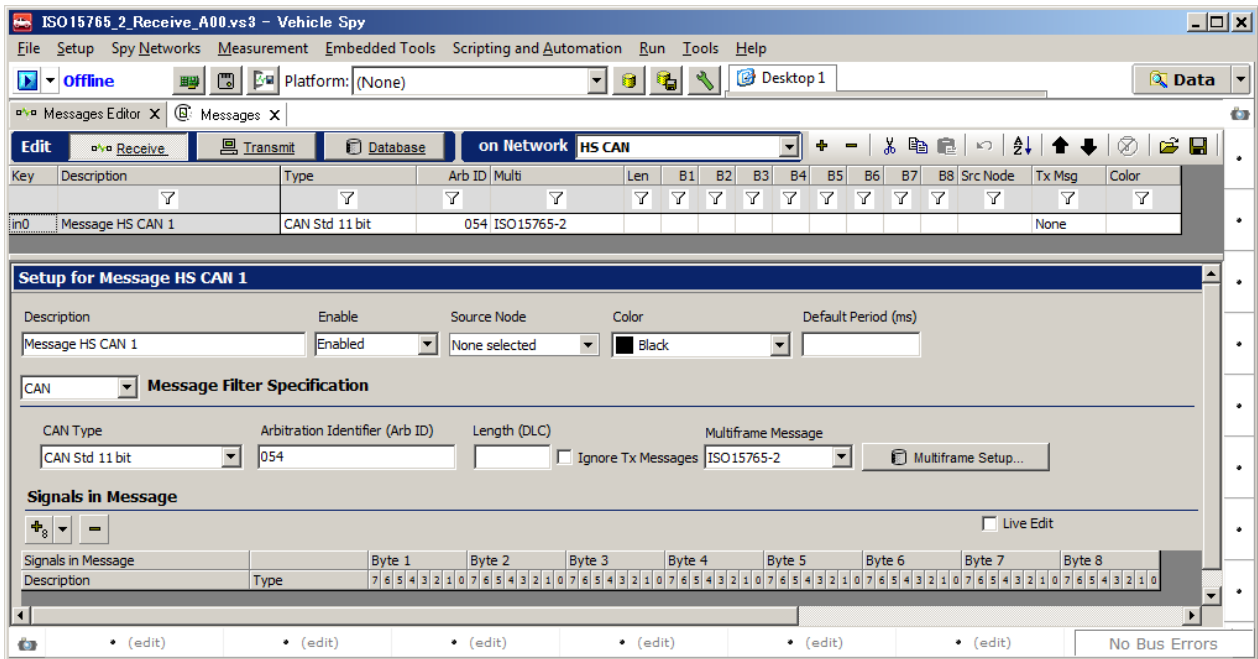


図 3.2.8

13. ここで、Multiframe Stup...を選択して、Rx Message Filter, Transmitted Flow Control, Timeouts を設定します。Flow Control frame としての ID を設定します。今回は、ID:0x765 とします。これはマルチフレーム通信では必須となります。

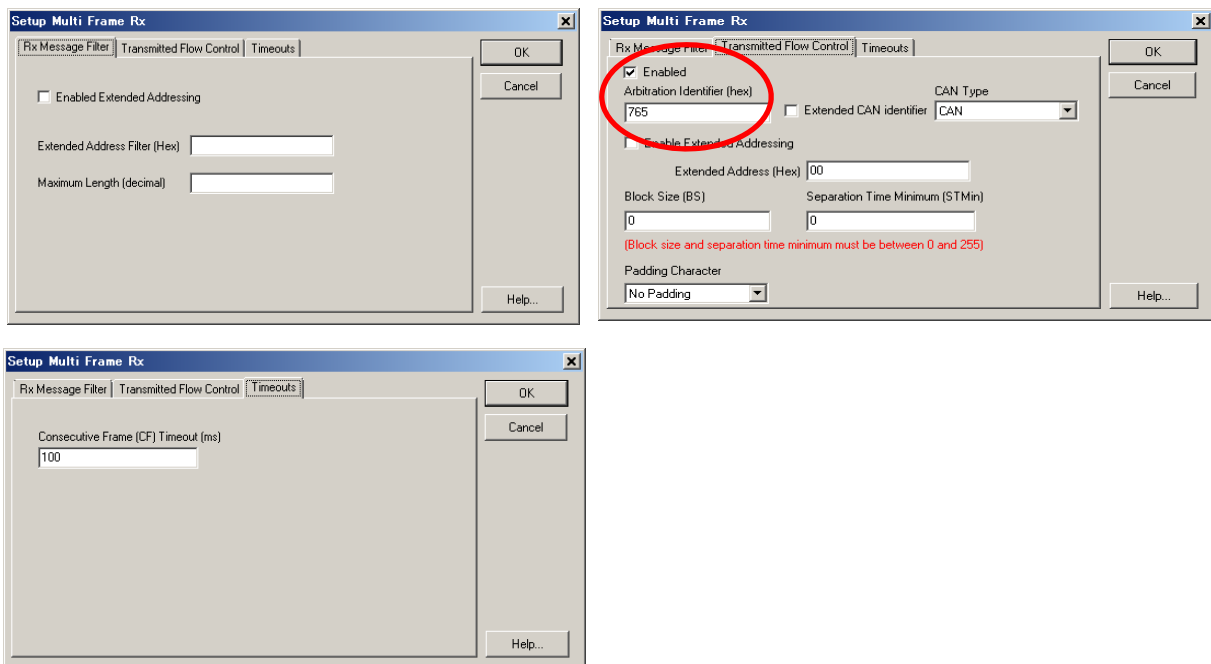


図 3.2.9

14. Arbitration Identifier(hex)に 29 ビット拡張アドレスを使用する場合は、”Extended CAN identifier”にチェックを入れます。以下参考例を示します。

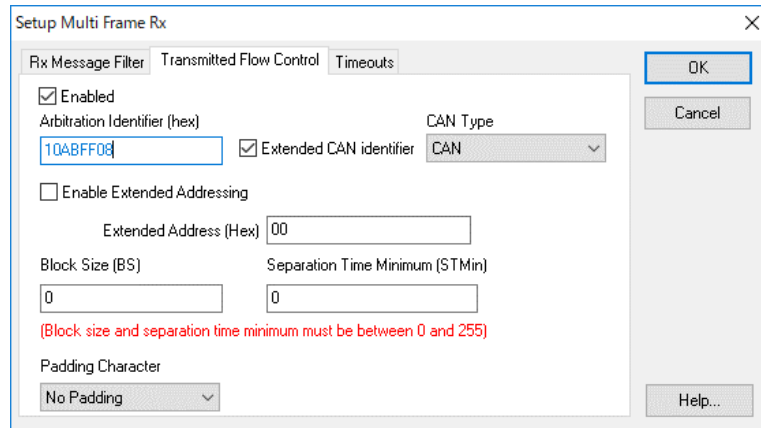


図 3.2.10

15. ここで、Messages 画面を表示させ、Online にします。

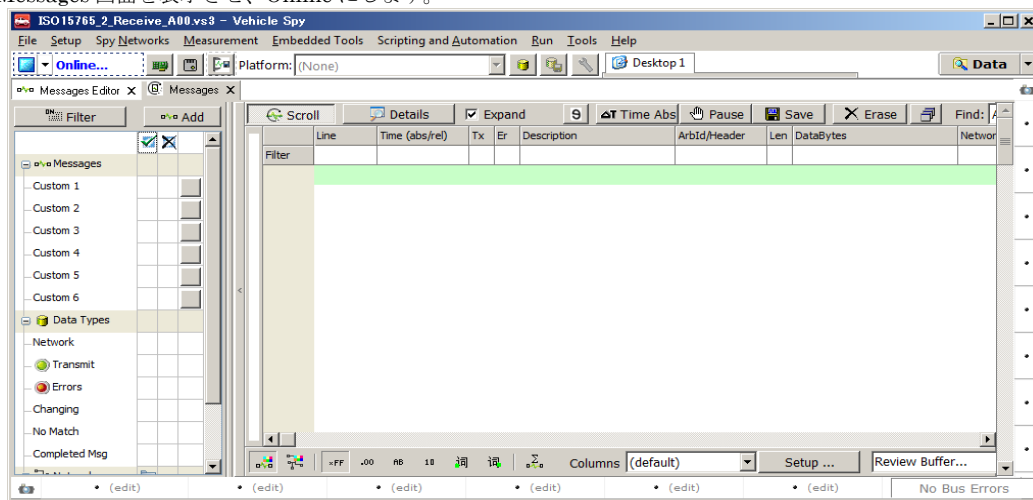


図 3.2.11

16. neoVI FIRE①からメッセージを送信してみます。送信メッセージ画面(図 3.2.6 参照)の Tx Panel から 1 メッセージを送信します。時系列に表示させるために Scroll ボタンを ON にしておきます。DataBytes 部を見ると、設定したデータが送信されていることがわかります。図 2.2.1.1 のデータフロー通りの内容となっています。

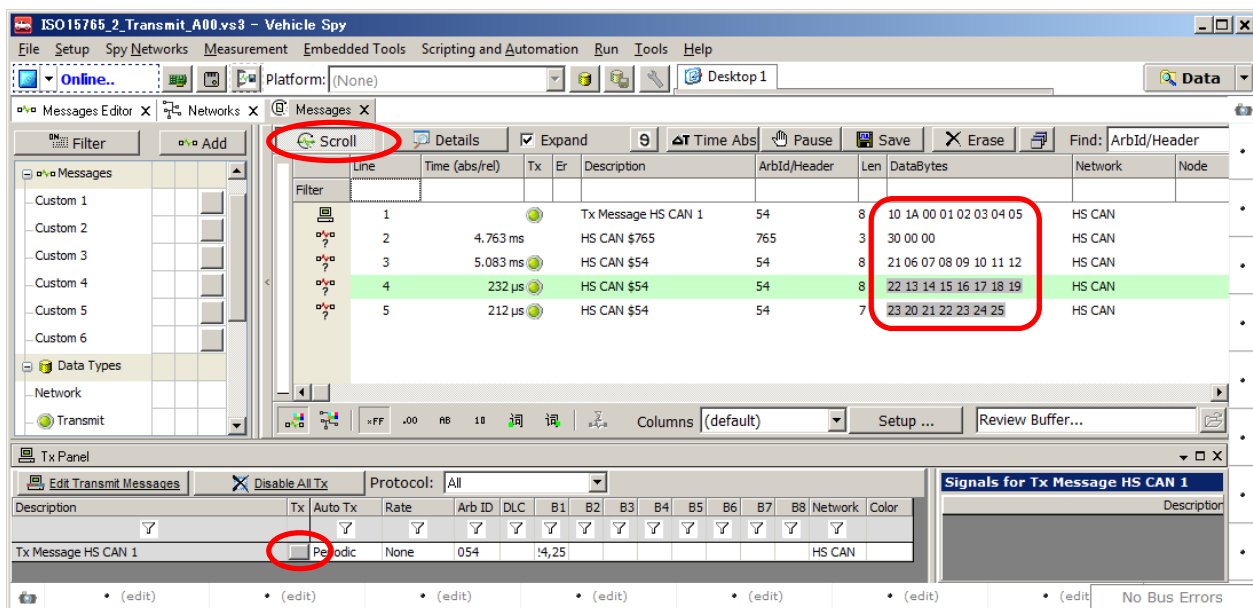


図 3.2.12

17. neoVI FIRE②の受信データをもう一つの Vehicle Spy3 から確認すると、以下のようにデータが 00-25 まで 26 バイト受信されていることがわかります。

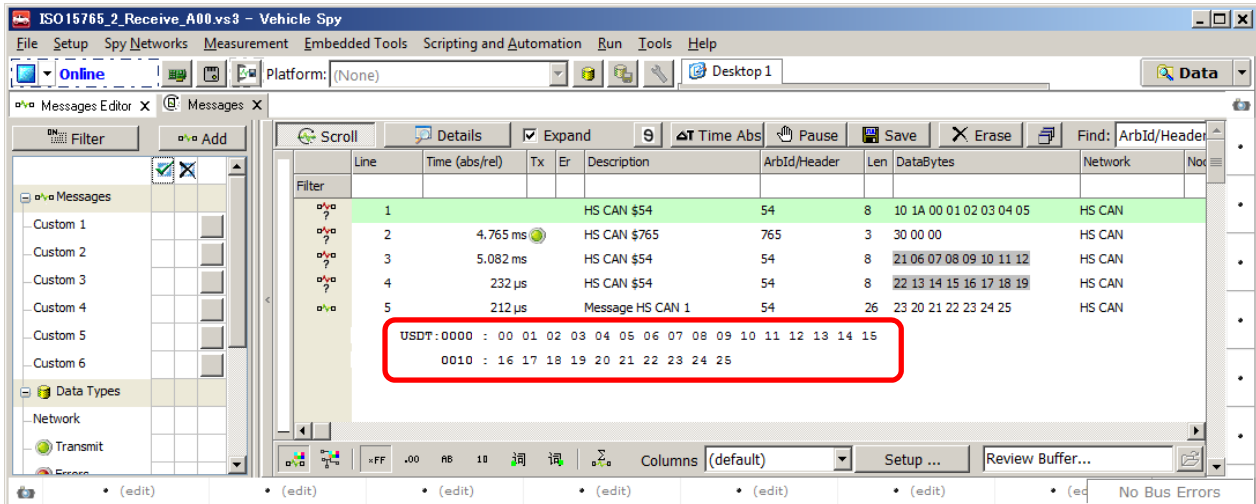


図 3.2.13

3.3. Extended addressing

送信側設定は、以下の設定部分を除き 3.2 項記載内容と同じです。

- 3.2.5 項の Transmitted Message, Flow Control Filter, Timeouts を設定します。
 - Transmitted Message 画面にする
 - Enabled Extended Addressing にチェックを入れる。
 - Extended Address[Hex]にアドレスを入力する（今回は 0x89）。
 - Flow Control Filter 画面にする。
 - Enabled Extended Addressing Filter にチェックを入れる。
 - Extended Address[Hex]にアドレスを入力する（今回は 0x89）。
 - その他デフォルトのまま。
 - 設定終了後、Messages 画面と Tx Panel 画面を表示させ、Online にする。 3.2.6 項と同じ画面となります。

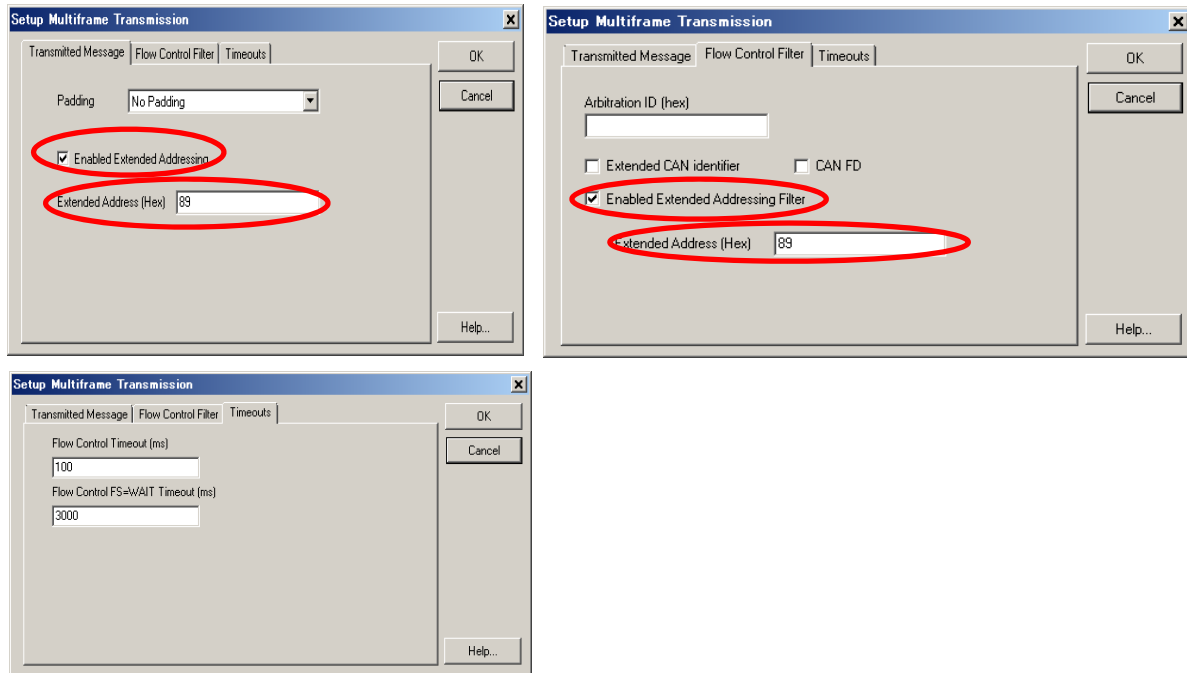


図 3.3.1

2. Rx Message Filter, Transmitted Flow Control, Timeouts を設定します。
- 1) Rx Message Filter 画面にする
 - 2) Enable Extended Addressing にチェックを入れる。
 - 3) Extended Address Filter[Hex]にアドレスを入力する (今回は 0x89)。
 - 4) Transmitted Flow Control 画面にする。
 - 5) Enabled Extended Addressing にチェックを入れる。
 - 6) Extended Address[Hex]にアドレスを入力する (今回は 0x89)。
 - 7) その他デフォルトのまま。
 - 8) 設定終了後、Messages 画面にして Online にする。 3.2.9 項と同じ画面となります。

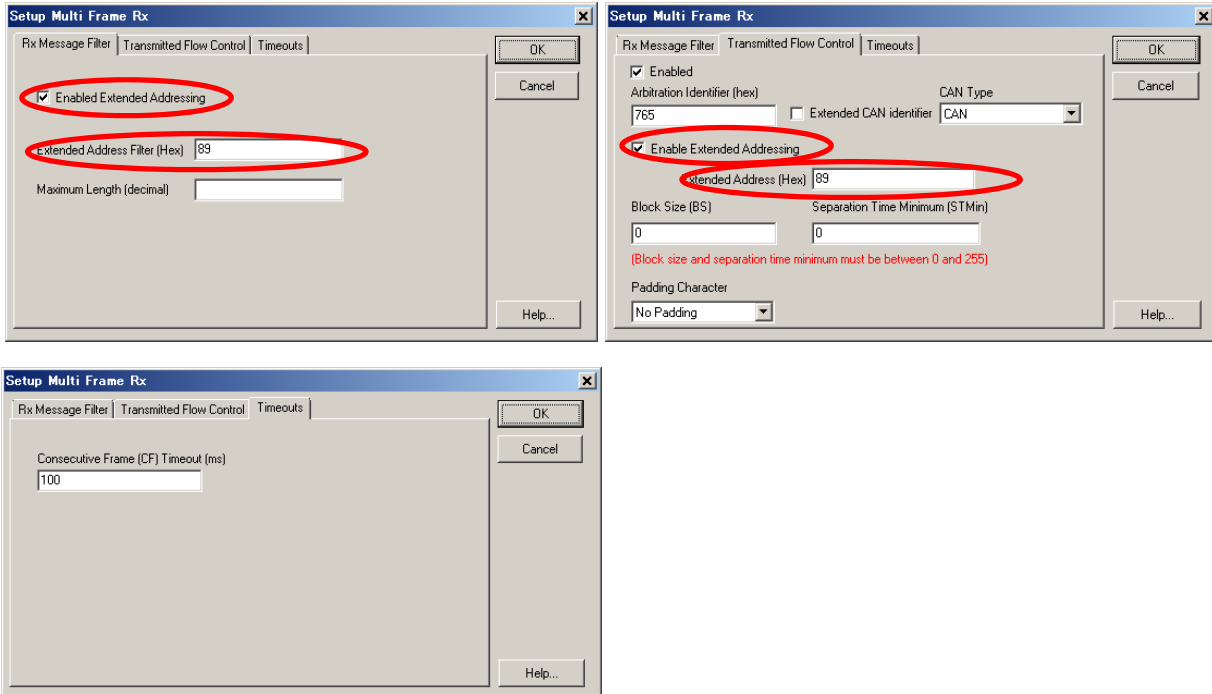


図 3.3.2

18. ここで、Tx Panelを使用して neoVI FIRE①からメッセージを送信してみます (3.2 項参照)。DataBytes 部を見ると、設定したデータが送信されていることがわかります。ただし、今回は1バイト目に 0x89 が挿入されていますが、これはまさに図 2.2.2.1 のデータフロー通りの内容となっています。

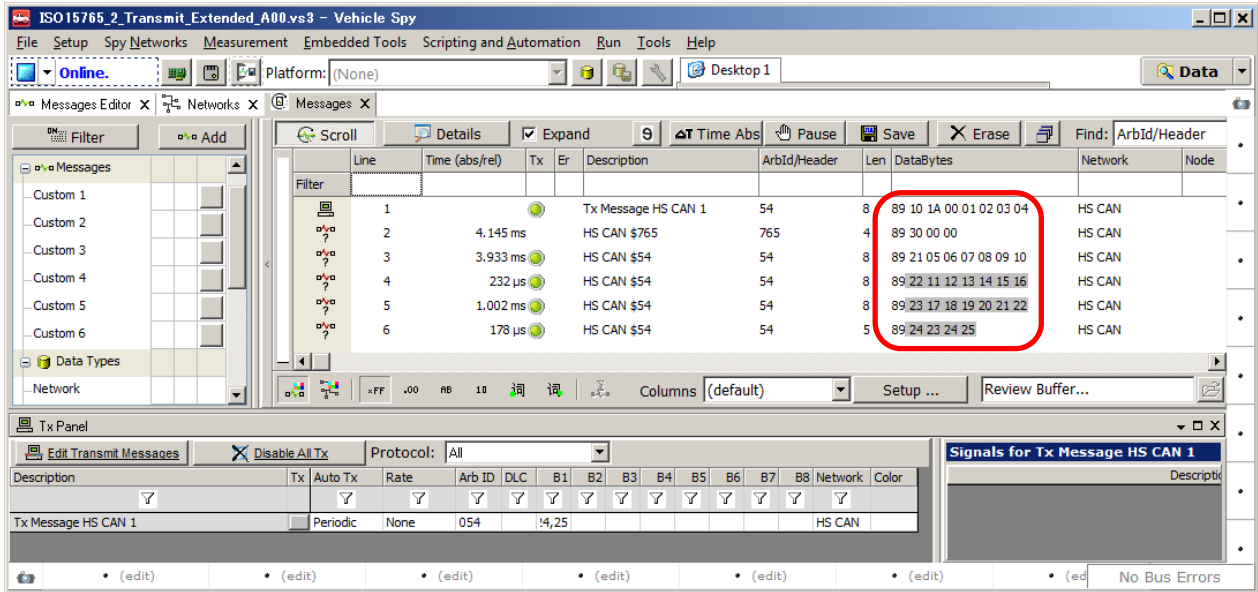


図 3.3.3

19. neoVI FIRE②の受信データをもう一つの Vehicle Spy3 から確認すると、以下のようにデータが 00-25 まで 26 バイト受信されていることがわかります。

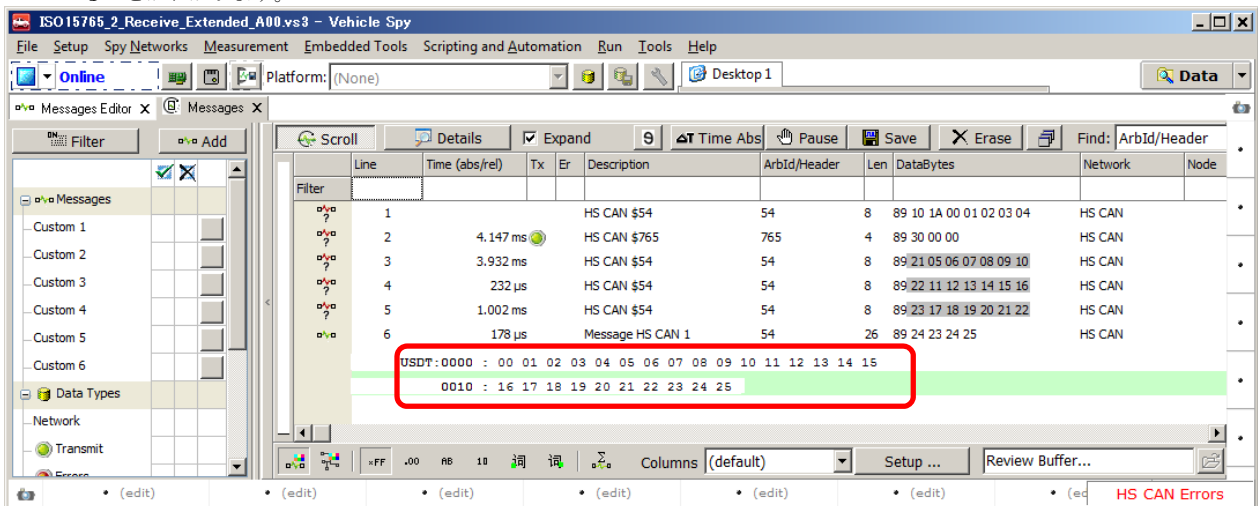


図 3.3.4

4. Vehicle Spy3サンプルプログラム

1. 以下今回のテストで使用したサンプルプログラムです。

ファイル名	内容
ISO15765_2_Transmit_A00.vs3	Normal addressing での送信用スクリプト。 (3.2 項 neoVI FIRE①用)
ISO15765_2_Receive_A00.vs3	Normal addressing での受信用スクリプト。 (3.2 項 neoVI FIRE②用)
ISO15765_2_Transmit_Extended_A00.vs3	Extended addressing での送信用スクリプト。 (3.3 項 neoVI FIRE①用)
ISO15765_2_Receive_Extended_A00.vs3	Extended addressing での受信用スクリプト。 (3.3 項 neoVI FIRE②用)

表 4.1

注記)

上記は当社 Web 上に ISO15765_2_RxTx_Sample_Basic_A00.zip として保存してあります。

<http://www.intrepidcs.jp/2896126009124801245412531125251254012489.html>

ISO15765_2_メッセージ送受信基礎編

2. ISO15765-2 を使用したアプリケーション例。

以下の例は neoVI FIRE を 2 つ使用して ECU 側と Tester 側をシミュレーションするものです。 Tester 側より特定のコマンドを送るとそれに応じて ECU 側から ISO15765-2 を使用してデータ返信するものです。

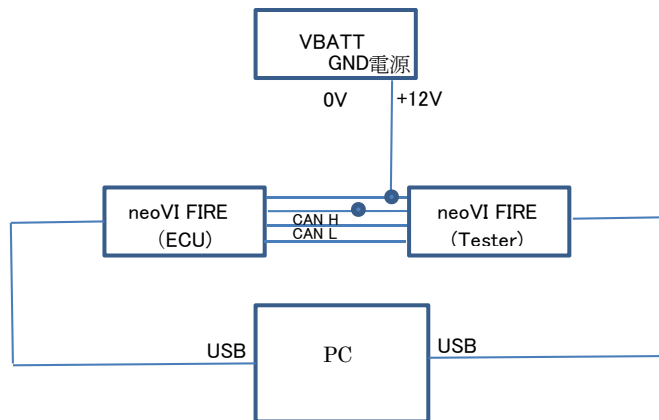


図 4.1

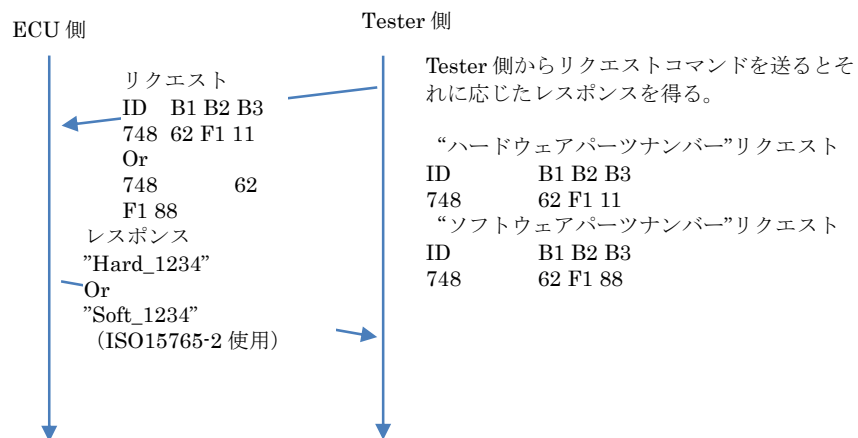


図 4.2

ファイル名	内容
ISO15765_2_ECU_Simulation_PC_Mode_A00.vs3	ECU 側スクリプト (PC モード)
ISO15765_2_Tester_Simulation_A00.vs3	Tester 側スクリプト
ISO15765_2_ECU_Simulation_StabdAlone_Mode_A00.vs3	ECU 側スクリプト(スタンドアロンモード)

表 4.2

注記)

上記は当社 Web 上に ISO15765_2_SampleApplication_Basic_A00.zip として保存してあります。

<http://www.intrepidcs.jp/2896126009124801245412531125251254012489.html>

ISO15765_2_アプリケーション例基礎編

5. その他

- 1) ご質問は icsjapan@intrepidcs.com までメールでご連絡をお願いします。

6. 変更履歴

日付	バージョン	変更内容	作成者
2016/04/28	1.0	初版。	春川
2016/05/11	1.1	4 項 2 番を追加。	春川
2017/07/11	1.2	- 3.2 項に neoVI FIRE/RED スダンドアロンでの使用時の注意点追加。 - 7 項 URL 変更。	春川

表 6.1 変更履歴

7. 用語一覧

Vehicle Spy (VSpy3)	当社ソフトウェア製品名。3 つのバージョンが存在します。 詳細は以下参照下さい。 https://www.intrepidcs.com/products/software/compare-vehicle-spy-versions/

表 7.1 用語一覧