| Intrepid Control Systems, Inc.<br>アプリケーションノート | neoECU<br>サンプルプログラム<br>No1 |
| --- | --- |

（株）日本イントリピッド・コントロール・システムズ

# 目次

1. 概要 3
2. neoECU-10 3
2.1. サンプルプログラム 3
2.1.1. NE10_LEDBlinking.vs3 3
2.1.2. NE10_ButtonLEDExample.vs3 4
2.1.3. NE10_AnalogInputs.vs3 6
3. neoECU-20 8
3.1. サンプルプログラム 8
3.1.1. HS to MS Gateway.vs3 8
4. VSpy3 ヘルプ 11
5. その他 11
6. 変更履歴 11
7. 用語一覧 11

# 1.　概要

本資料は、neoECU ディバイス（neoECU-10/neoECU-20）を使用したアプリケーションプログラムの作成方法についてサンプルプログラムを使用して解説することを目的とします。　尚、本資料を読まれる前に“アプリケーションノート neoECU　基礎編“に目を通しておく必要があります。

# 2.　neoECU-10

## 2.1.　サンプルプログラム

neoECU-10 では 8 つのサンプルプログラムを準備しています。

| ファイル名 | 内容 |
|---|---|
| NE10_LEDBlinking.vs3 | neoECU-10 の LED を点滅するサンプルプログラム |
| SW NeoECU-10 HVWU.vs3 | SW CAN を設定し、neoECU-10 HLSL+向けの高電圧ウェイクアップ |
| NE10_LSFT.vs3 | neoECU-10 HLSL+での LSFT CAN を設定するプログラム |
| NE10_ButtonLedExample.vs3 | 押したボタンに応じる LED を点滅するプログラム |
| NE10_MiscIO_Input.vs3 | MISC IO の LED を設定するプログラム |
| NE10_MiscIO_Output.vs3 | MISC IO を出力にする、ラインをトーグルするプログラム |
| NE10_AnalogInputs.vs3 | アナログ入力を読み込み、CAN フレームで送信するプログラム |
| HS To MS Gateway.vs3 | HS CAN の通信を全部 MS CAN に変換するサンプルプログラム |

表 2.1.1

今回は、NE10_LEDBlinking.vs3、NE10_ButtonLedExample.vs3 と NE10_AnalogInputs.vs3 のサンプルプログラムについて解説します。

### 2.1.1.　NE10_LEDBlinking.vs3

本プログラムは 5 つの LED を順番に点灯/消灯するプログラムです。

1. VSpy で上記ファイルを開いて下さい。
2. CoreMini Console を使用し、プログラムを neoECU-10 に書き込んで下さい。
3. Scripting and Automation のメニューで、Function Blocks を開いて下さい。
4. このサンプルプログラムでは一つのファンクションブロックを持っています。
5. ファンクションブロックのスクリプトは以下のようになっています。

(1)　LED3、4、5 の MISC IO を出力に設定

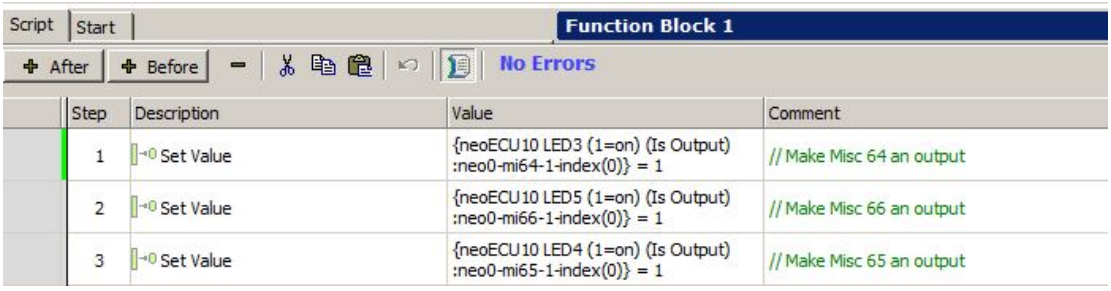


図 2.1.1.1

(2)　5~14 行目は、LED5 から LED1 までを順番に点灯するコードです

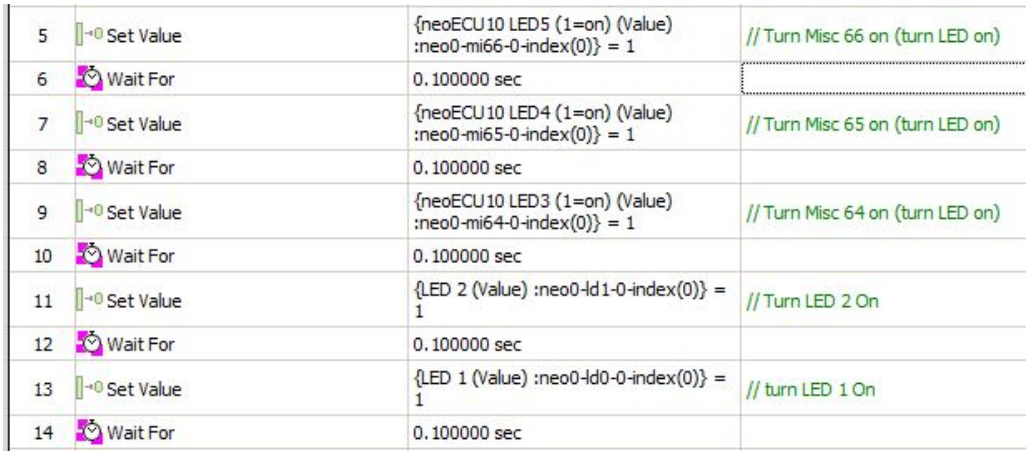


図 2.1.1.2

(3) 15~24 行目までは、LED5 から LED1 までを順番に消灯するコードです。

| | | | |
|---|---|---|---|
| 15 | Set Value | {neoECU10 LED5 (1=on) (Value) :neo0-mi66-0-index(0)} = 0 | // Turn Misc 66 on (turn LED off) |
| 16 | Wait For | 0.100000 sec | |
| 17 | Set Value | {neoECU10 LED4 (1=on) (Value) :neo0-mi65-0-index(0)} = 0 | // Turn Misc 65 on (turn LED off) |
| 18 | Wait For | 0.100000 sec | |
| 19 | Set Value | {neoECU10 LED3 (1=on) (Value) :neo0-mi64-0-index(0)} = 0 | // Turn Misc 64 on (turn LED off) |
| 20 | Wait For | 0.100000 sec | |
| 21 | Set Value | {LED 2 (Value) :neo0-ld1-0-index(0)} = 0 | // Turn LED 2 OFF |
| 22 | Wait For | 0.100000 sec | |
| 23 | Set Value | {LED 1 (Value) :neo0-ld0-0-index(0)} = 0 | // turn LED 1 Off |
| 24 | Wait For | 0.100000 sec | |
| 25 | Jump To | Step 5 | |

図 2.1.1.3

(4) 以下すべての LED が点灯した状態です。

図 2.1.1.4

### 2.1.2. NE10_ButtonLEDExample.vs3

本プログラムは neoECU-10 の内部プッシュスイッチを押すことにより LED を点灯させる為のプログラムです。

1. VSpy でファイルを開いて下さい。
2. CoreMini Console を使用し、プログラムを neoECU-10 に書き込んで下さい。
3. Scripting and Automation のメニューで、Function Blocks を開いて下さい。

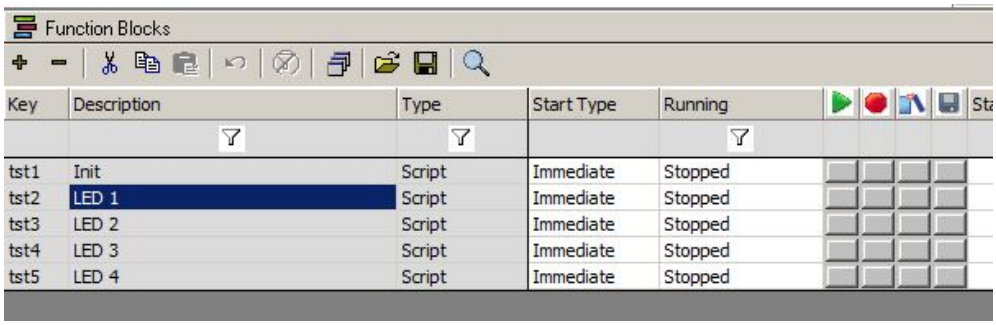


図 2.1.2.1

4. Init のクリプトでは LED3、4、5 を出力に設定しています。

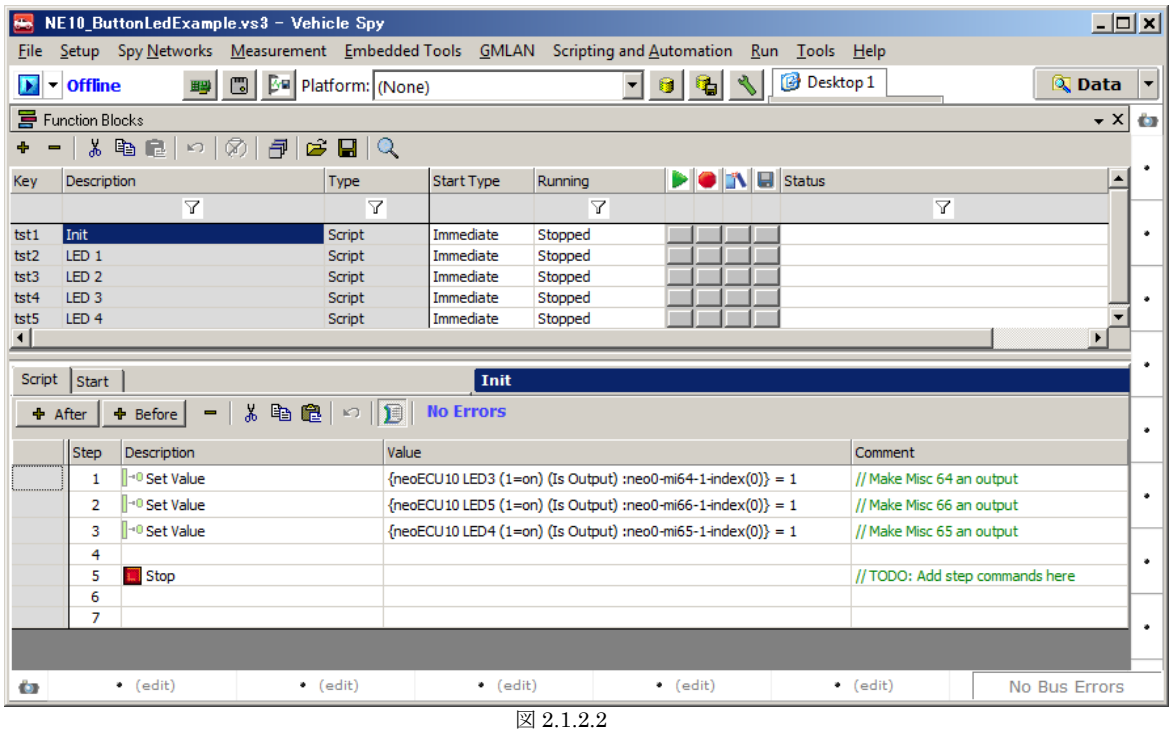

図 2.1.2.2

5. このスクリプトの 1 行目で、LED の Value に対して内部プッシュスイッチの値を設定しています。　つまり、ボタンを押した時に LED を ON するように設定しています。

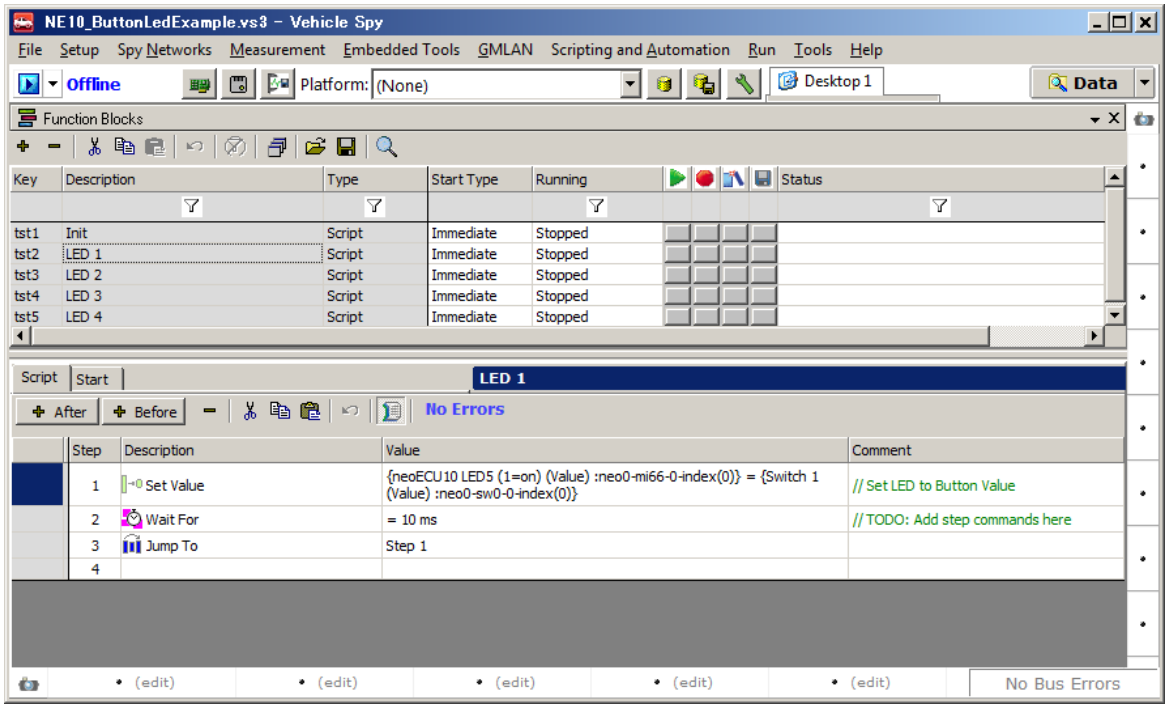

図 2.1.2.3

6. LED 1 から LED 4 までのファンクションブロックは LED の指定番号が違う以外は同じスクリプトとなります。

### 2.1.3. NE10_AnalogInputs.vs3

本プログラムは neoECU-10 を使用してアナログ入力を取り込み、CAN データとしてその値を送信します。

1. VSpy でファイルを開いて下さい。
2. CoreMini Console を使用し、プログラムを neoECU-10 に書き込んで下さい。
3. Scripting and Automation のメニューで、Function Blocks を開いて下さい。
4. このサンプルプログラムには一つのファンクションブロックがあります。
5. ファンクションブロックのスクリプトは以下のようになっています。

(1) 1~4 行目まではアナログ入力を有効にするスクリプトです。

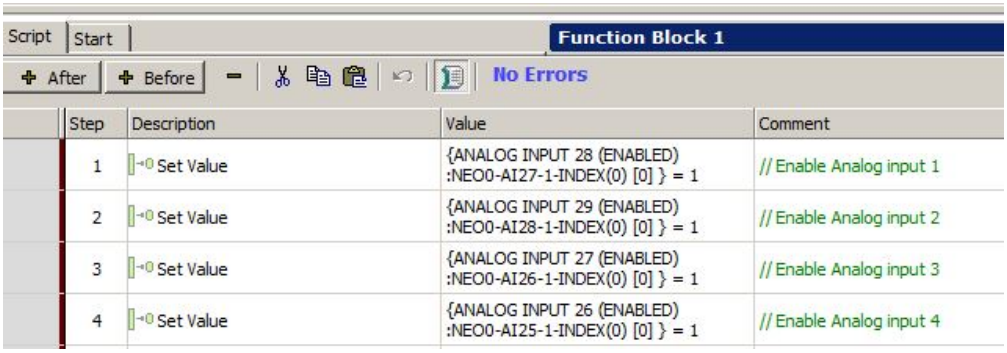


図 2.1.3.1

(2) 6~10 行目まではアナログ入力の値をシグナルデータに保存するスクリプトです。10 行目で定義したメッセージを送信します

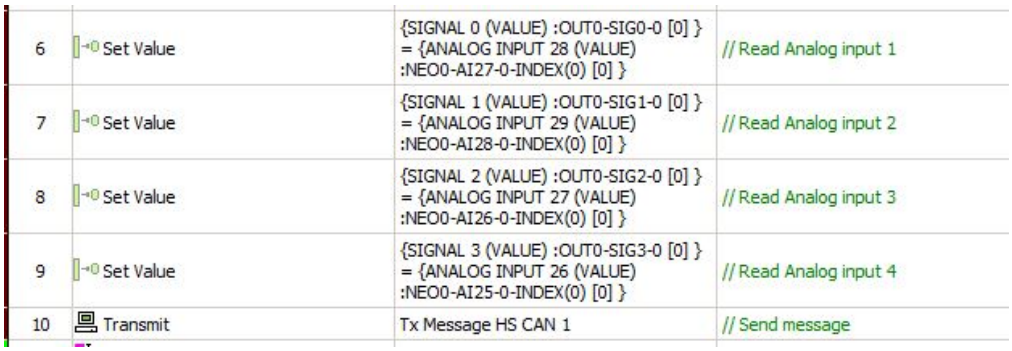


図 2.1.3.2

(3) 「Run with Monitor」をクリックしてスクリプトを実行します。　「DataBytes」にはアナログデータが出力されます。

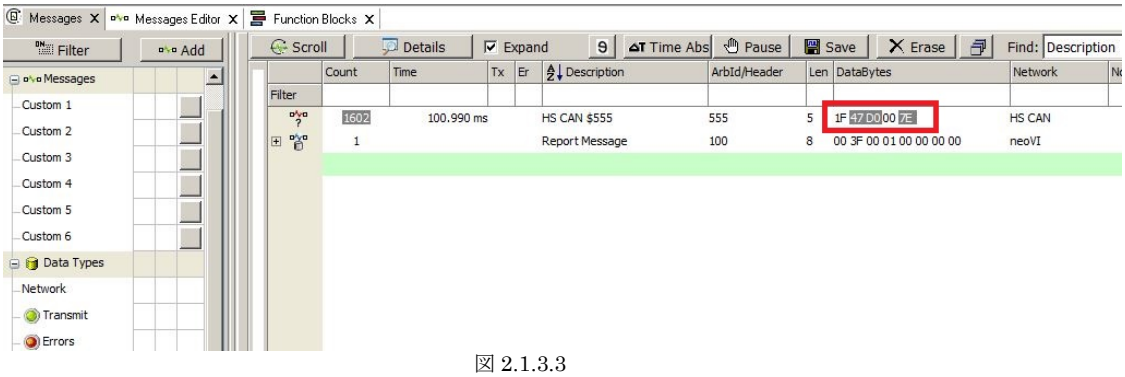


図 2.1.3.3

6. 「neoECU-10 から ID555 の HS CAN メッセージを送信しています。　「DataBytes」の 1-5 バイトまでの値がアナログデータとなります。

7. ID555 の HS CAN メッセージのシグナル値として以下が定義されています。　1 つのアナログ値は 10 ビットデータとなります。

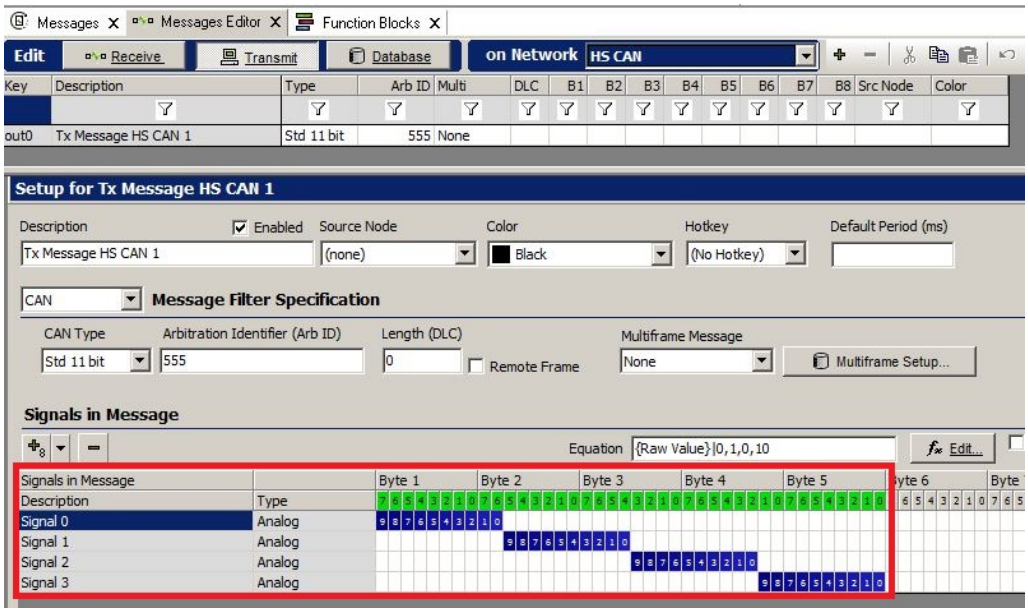

図 2.1.3.4

# 3.　neoECU-20

## 3.1.　サンプルプログラム

neoECU-20 では２つのサンプルプログラムを準備しています。

| ファイル名 | 内容 |
|---|---|
| HS To MS Gateway.vs3 | HS CAN メッセージを MS CAN に送信します。 |
| CoreMini Gateway Example.vs3 | 上記に加え、MS CAN メッセージを HS CAN に送信します。 |

表 3.1.1

今回は、HS To MS Gateway vs3 のサンプルプログラムを解説します。

### 3.1.1.　HS to MS Gateway.vs3

本サンプルプログラムは HS CAN と MS CAN 間のゲートウェイとして動作します。　HS CAN から入力されたデータを MS CAN へ送信します。　今回は以下の接続状態で試験を行うものとします。

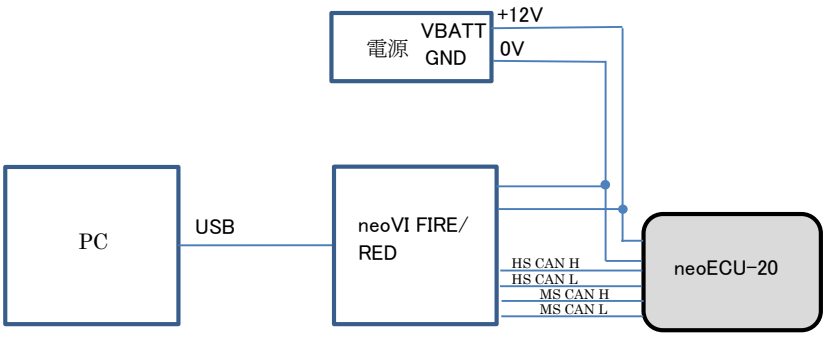


図 3.1.1.1

neoECU-20 には、既に以下のスクリプト（HS To MS Gateway.vs3）が書き込まれているものとします。　以下にファンクションブロックを示します。HS CAN へのメッセージが入力された場合、そのデータを MS CAN へ送信しているのがわかると思います。

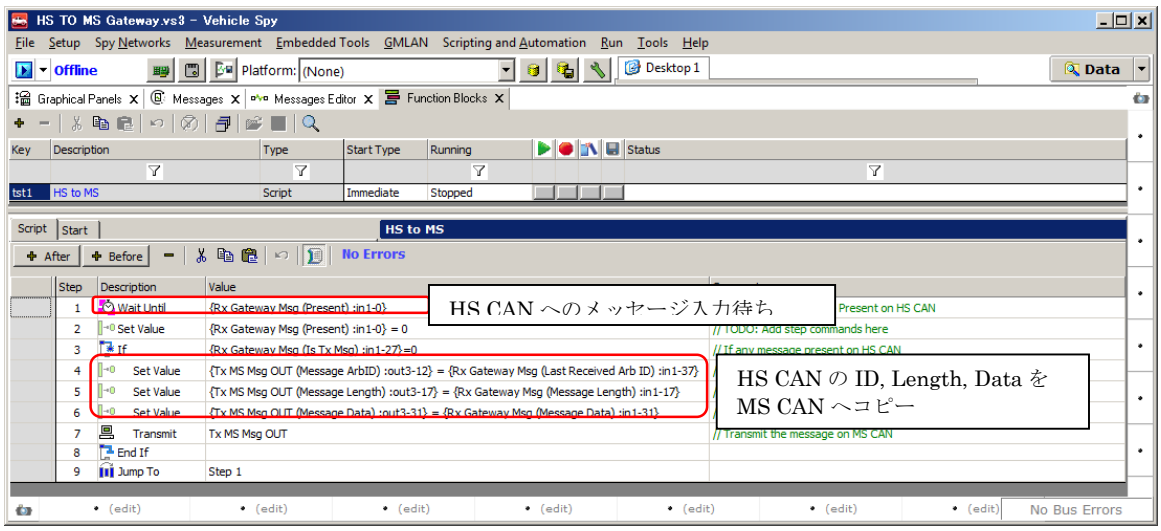


図 3.1.1.2

1. neoVI FIRE/RED より HS CAN 信号を送信する為に送信メッセージの作成が必要です。今回は、上記 neoECU-20 で使用した HS To MS Gateway.vs3 が既に各送受信メッセージを持っている為、それをそのまま使用することにします。　よって VSpy から HS To MS Gateway.vs3 ファイルを読み込みます。
2. 読み込み終了後、Spy Networks → Tx Panel をクリックして下さい。

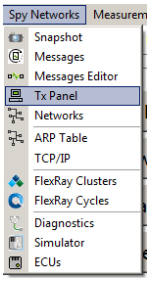


図 3.1.1.3

3. ウィンドウサイズを調整し Messages ウィンドウと Tx Panel のウィンドウのどちらも見えるようにして下さい。

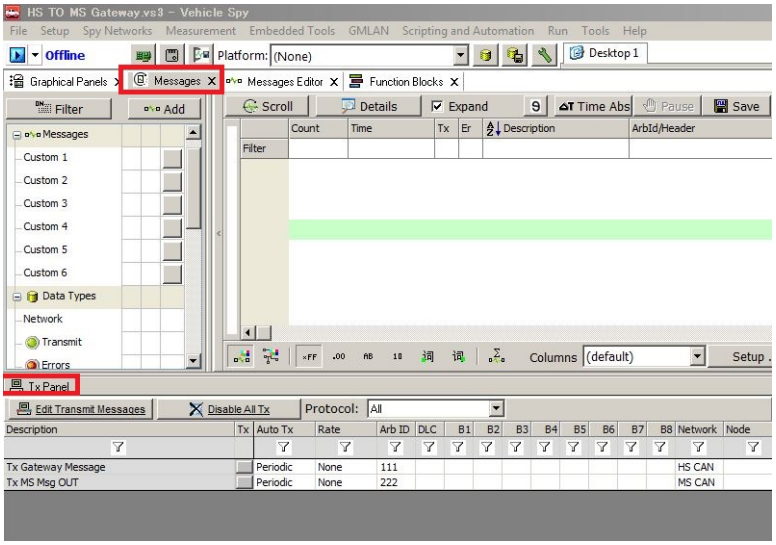


図 3.1.1.4

4. VSpy の画面の左上の部分の青い矢印のところに、「Run with Transmit」を選択して下さい。

図 3.1.1.5

5. Online 後最初は何も出力されていませんが、Tx Panel の Tx Gateway Message ボタンを押すと送受信メッセージが出力されます。

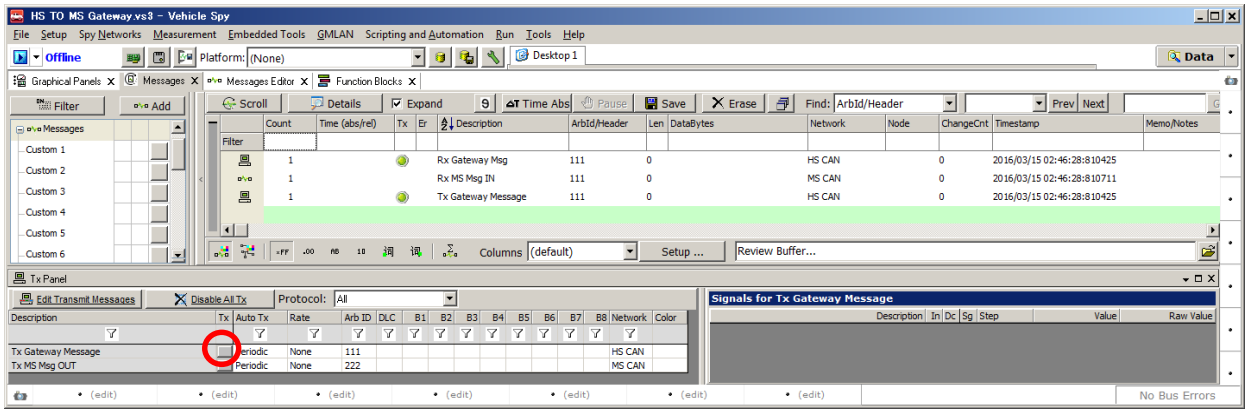


図 3.1.1.6

6. Messages 画面上、HS CAN の送信メッセージとしてメッセージ名 Tx Gateway Message、ID が 111 として表示されています①。それに対して、MS CAN から受信メッセージとしてメッセージ名 Rx MS Msg IN、ID が 111 として受信されています②。これにより neoECU-20 に書き込んだスクリプト（HS CAN メッセージを MS CAN メッセージへ送信）は正常に動作している事がわかります。　又、他のメッセージ③は neoVI FIRE/RED 上で HS CAN の送信 ID と同じ ID で HS CAN の受信メッセージとして定義してある為、HS CAN メッセージとして送信したメッセージをそのまま受信メッセージとして表示しています（neoECU-20 のゲートウェイウェイ動作には関係がありませんので無視しても構いません）。

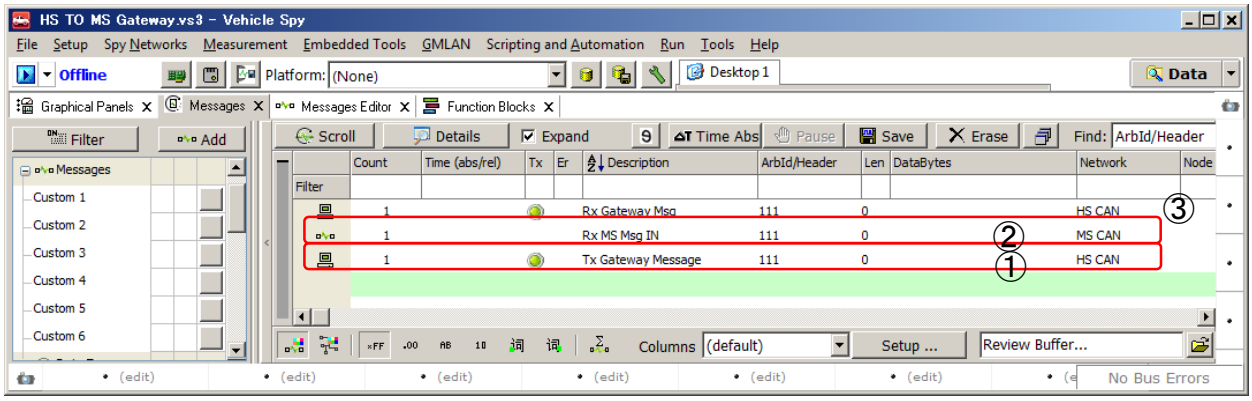


図 3.1.1.7

# 4.　VSpy3ヘルプ

VSpy3 上から neoECU に関するヘルプが参照できます。　VSpy3 メニューの Help ウィンドウを開き、検索メニューから neoECU と入力します。以下のようにヘルプが表示されます。

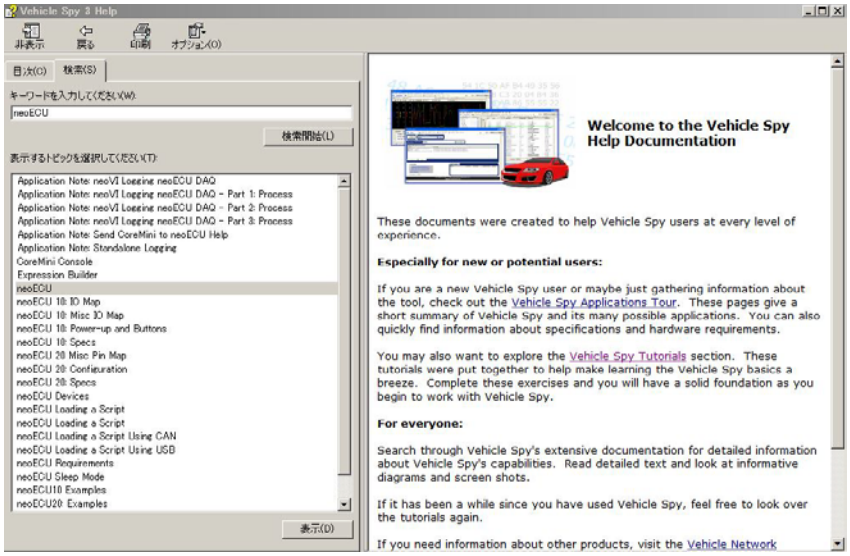

図 4.1 VSpy3 ヘルプ

# 5.　その他

ご質問は icsjapan@intrepidcs.com　までメールでご連絡お願いします。

# 6.　変更履歴

| 日付 | ヴァージョン | 変更内容 | 作成者 |
|---|---|---|---|
| 2016/02/26 | 1.0 | 初版。 | ホアン |
| 2016/03/16 | 1.1 | 全般的見直し | 春川 |

表 9.1 変更履歴

# 7.　用語一覧

| | |
|---|---|
| VSpy3 | 当社ソフトウエア製品 Vehicle Spy 3 の略称。3 つのヴァージョンが存在します。　詳細は以下を参照下さい。<br>http://www.intrepidcs.com/knowyourvspy.html |
| Function Blocks | VSpy3 のプログラミングに使用します。 |
| CoreMini（コアミニ） | neoVI のスタンドアロン動作に必要な設定ファイル。用例：CoreMini をコンパイルする。CoreMini を neoVI へダウンロードする。 |

図 7.1 用語一覧