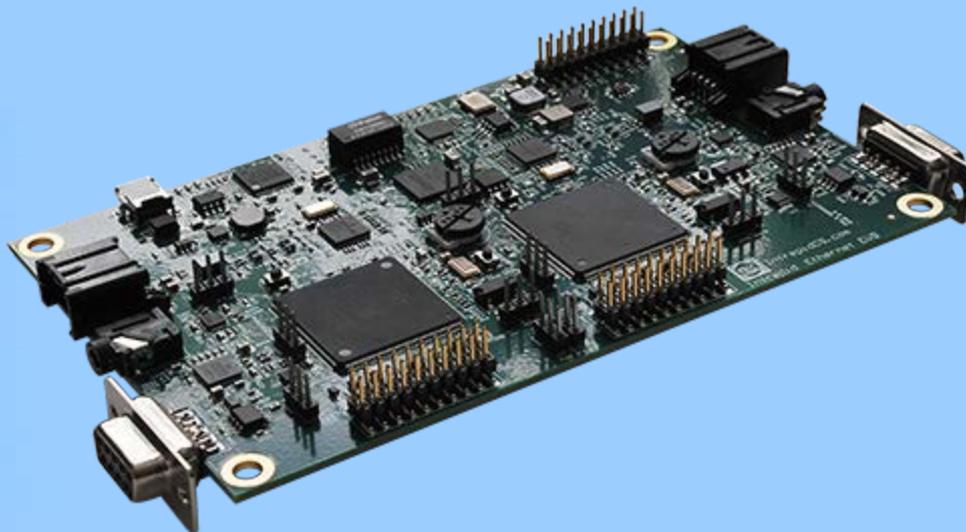


株式会社 日本イントリピッド・コントロール・システムズ



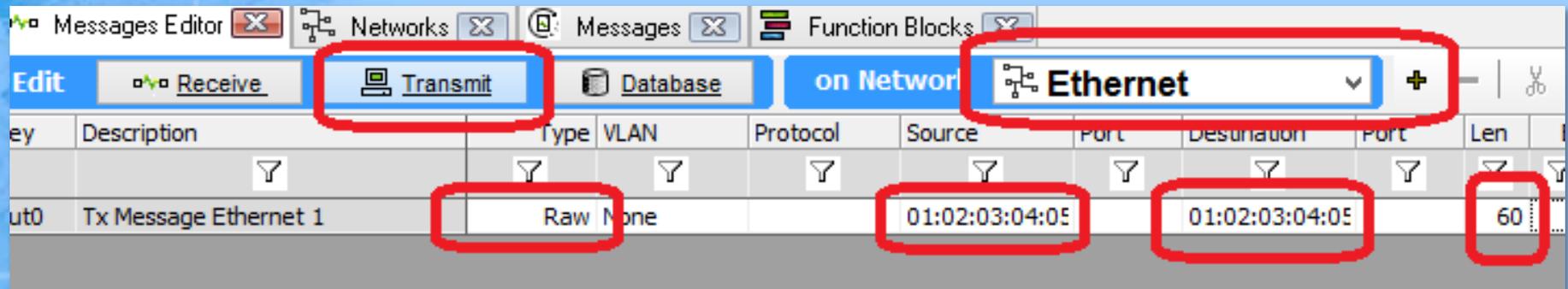
初めてのEthernet EVB



(株) 日本イントリピッド・コントロール・システムズ
231-0015 横浜市中区尾上町5-80 神奈川産業振興センター 7階3号室
TEL:045-263-9294 FAX:045-263-9296 URL:<http://intrepidcs.jp/>

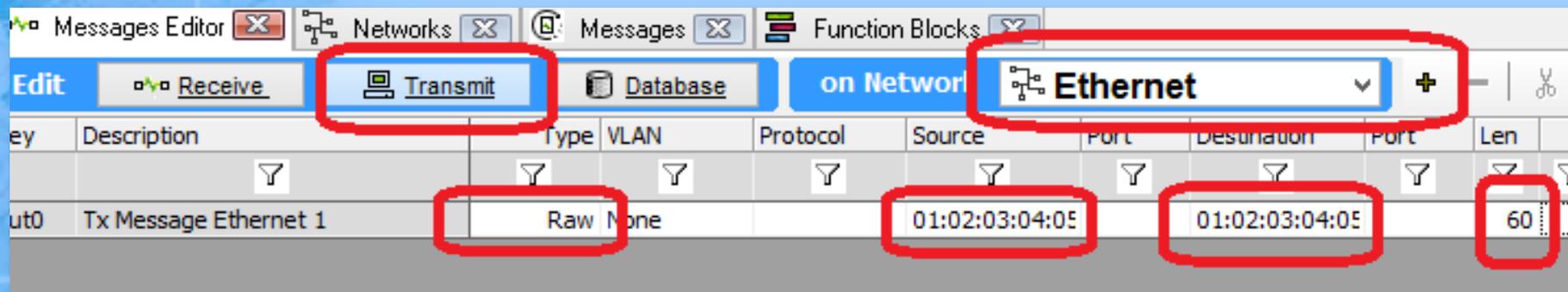
生値メッセージを送信

- まず静的MACアドレスに対して、生のEthernetフレームを送信してみます
- Spy Networks → Message Editor
 - “Transmit”ボタンをクリック



生値メッセージを送信- 2

- “on Network” ドロップダウンにて、Ethernetを選択 (デフォルトではHS CAN)
- “+” ボタンを押下し空のメッセージを作成
 - “Type” をRaw、 “Len” を60に設定(最小のEthernetフレーム)



生値メッセージを送信- 3

- 画面下部“Signals in message”横の“Ethernet payload”を選択：ここで定義したシグナルが、Ethernet payloadとして送信されます
- “+”を押下し、Ethernet payloadの最初のバイトを追加

The screenshot shows the 'Signals in Message' section of a software interface. At the top, there are three buttons: 'Deepest (Level 3 payload)', 'Level 2 payload', and 'Ethernet payload'. The 'Ethernet payload' button is highlighted with a red box. Below these buttons is a control panel with a '+' button (also highlighted with a red box), a dropdown arrow, and a '-' button. To the right, there is a 'Tx Event' dropdown set to 'None' and an 'Equation' field. Below this is a table with the following structure:

Signals in Message		Byte 1								Byte 2								Byte 3							
Description	Type	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5					
Signal 0	Analog	7	6	5	4	3	2	1	0																

生値メッセージを送信- 4

- ・ 次に、スタンドアロンにてEVBが先のメッセージを送信するよう、スクリプトを作成します
- ・ Scripting and Automation → Function Blocks
 - ・ 新規のスクリプト・ ファンクションブロックを追加
 - ・ 画面下部のStartタブにて、開始モードを“Immediate Embedded Only”に設定(PC上のVehicle Spyからは実行しない)



生値メッセージを送信- 5

Key	Description	Type	Start Type	Runn
tst0	Function Block 1	Script	Immediate Embedded Only	Stopp

Script Start Function

Start Immediately Embedded Only

生値メッセージを送信- 6

- Scriptタブにて、“Set Value step”を新規追加、そのValue欄をダブルクリック
- Tx Messages項目より、Tx Message Ethernet 1の下にあるSignal 0を選択
- Pick Set Value及びAdd To Expressionを押下
- Expressionに” + 1 ”を追加
 - Ethernetフレーム中のシグナル値をインクリメント



生値メッセージを送信- 7

Script Start Function

+ After + Before - [Icons]

The SetValue

Step	Description	Value
1	[-] Set Value	Click For Setup

Enter Expression for signal

Value To Set {Signal 0 (Value) :out0-sig2-0}

Expression {Signal 0 (Value) :out0-sig2-0} + 1

Clear Evaluate as text

fx Expression Builder

Rx Messages Database Tx Messages Signal Groups DAQ Jobs App Signals

Sort By: Networks Pick Set Value Add To Expression

Find Clear

- Ethernet
 - Tx Message Ethernet 1 (20)
 - Signal 0 (Analog)



生値メッセージを送信- 8

- Transmitステップを追加、先に定義したEthernetメッセージをドロップダウンより選択
- Wait Forステップを追加、値を0.5に設定

2	 Transmit	Tx Message Ethernet 1	
---	--	-----------------------	--

3	 Wait For	0.500000 sec	
---	---	--------------	--

生値メッセージを送信- 9

- ・ 次に、作成したスクリプトを、2つあるEVB上のノードのうち1つにダウンロード
- ・ Tools → Utilities → CoreMini Console
 - ・ 画面下部の“neoECU (CAN)”タブを選択
 - ・ “Poll for neoECUs”を押下
 - ・ “Ethernet EVB (50002)”を選択しSendを押下



生値メッセージを送信- 10

The screenshot displays the 'Device Configuration and Download' section of the IntrepidCS software. At the top, it shows 'Multi-frame Data Buffer Size: 60'. Below this, the 'Compile CoreMini' section indicates 'CoreMini Compiled with Warnings at UTC 2013/09/24 06:59:16.' and includes 'Compile' and 'Copy to Clipboard' buttons. The 'Device Configuration and Download' section features a dropdown menu for 'neoVI (USB) neoECU (CAN) RS232/UART/Bluetooth', with 'neoECU (CAN)' selected. Below this, the 'Select Download Tool to Communicate with neoECU' section shows 'VCAN3 123085' and 'HS CAN' selected, along with a 'Poll for neoECUs' button. A list of devices is shown, with 'Ethernet EVB (500002) App Version 1.4 - Running CoreMini' selected. To the right, there are buttons for 'Configure Device (Bit Rates, Initial Pin States, etc...)', 'Hardware Setup...', and 'Flash Firmware'. At the bottom, there are buttons for 'Send....', 'Extract...', and 'Clear', along with an 'Advanced Settings' checkbox.

生値メッセージを送信- 11

- Spy Networks → Messages
- 画面左上のスタートボタンを押下（青い矢印ボタン）
- 作成したメッセージが500ミリ秒毎に送信されていることを確認



生値メッセージを送信- 12

The screenshot shows a software interface for CAN bus analysis. The top status bar indicates the system is 'Online'. The main window displays a table of captured messages. A red box highlights a specific message with the following details:

Count	Time	Tx	Er	Description	ArbId/Header	Len	DataBytes	Network	Node	ChangeCnt	RTC Time
15	500.980 ms			Ethernet Woonsang_04:05:0...Woonsang_04:0...60	01 02 03 04 05 07 01 02 03 04 ...	Ethernet			0		2013/09/24 08:03:12:013777

The data bytes for this message are shown as:

```
Data:0000 : 5A 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0010 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  
0020 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Below the table, the details for the selected message are shown:

Details for "Ethernet Woonsang_04:05:06 to Woonsang_04:05:07"

- Message on Ethernet from Ethernet PCAP (icsenet.dll)
- Ethernet, Src: Woonsang (01:02:03:04:05:07), Dest: Woonsang (01:02:03:04:05:07)

Name	Value
01	02
02	03
03	04
04	05
05	07
06	01
07	02
08	03
09	04
10	...
11	...
12	...
13	...
14	...
15	...
16	...
17	...
18	...
19	...
20	...

