CANalyzer-Getting Started Tutorial

Introduction

This tutorial will focus on the software development of CAN tool. Including the database creation, control panel configuration. Note that this tutorial is based on you have certain knowledge of CAN bus, structure layer, and protocols.

What is CAN?

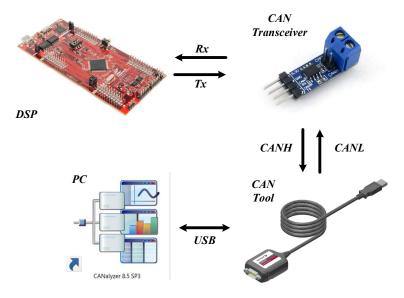
A Controller Area Network (CAN bus) is a robust vehicle bus standard designed to allow microcontrollers and devices to communicate with each other in applications without a host computer. It is a message-based protocol, designed originally for multiplex electrical wiring within automobiles to save on copper, but is also used in many other contexts.

What is CANalyzer?

CANalyzer is the comprehensive software tool with intuitive operation for analysis and stimulation of network communication. Use CANalyzer to check whether and what type of communication is occurring on the network. In addition to sending or recording data, interactive ECU diagnosis is also possible. For every application it offers powerful basic functions for beginners as well as extensive detailed functions for experienced users.

Before started

To realize the CAN control, certain hardware is needed. The typical system connection diagram is as shown below.



CANalyzer installation

The installation for CANalyzer is pretty easy, just pay attention that the driver of the CAN tool is needed. In this tutorial, the CAN tool is VN1610 from Vector. The CANalyzer software version is 8.5 SP3. After the installation of the driver and software, follow the steps below to check the device connection status:

- 1. connect the USB of the CAN tool to the PC;
- 2. Open CANalyzer;
- 3. On the top of the software, click Configuration->Network Hardware;
- 4. If you could see the figure of the CAN tool you connected, it's done.

∋- 💆 CAN 1	CAN channel 1		
Setup	Type:	VN1610	
Acceptance Filter Options	Controller:	Vector CAN/CAN FD IP Core	
S Hardware Sync	Cab/Piggy:	On board CAN 1051cap(Highspeed)	
	Exists:	Yes	
	CAN Contro	ller	
	Mode:	CAN. ~	
	Baud rate:	250.0 kBaud	
	Use dat	abase settings	
	Scan		

Create a new project

Click File->New Configuration. You can select the template of the new project. Here in this tutorial, let's select CAN_500kBaud_1ch.tcw. Therefore, a CAN project with 500kBaud is created. You can save the project by click File->Save Configuration. The initial interface is as shown below.

Template	X EVector CANalyzer /pro - Configuration1 *		- D	×
Select configuration template:	Ele View Start Configuration Iools Window He	🐉 Online- 💿 😠 💭 😿 💰 🗊 🗞 🖕	- 0 ×	
CAN_500kBaud_1ch.tcw CAN_500kBaud_2ch.tcw CAN_500kBaud_2ch.tcw CAN_83kBaud_1ch.tcw CAN_83kBaud_2ch.tcw CAN_83kBaud_2ch.tcw	습과 CAN 나듔 Channel 1	aud origina aud origina o o o o o o o o o o o o o	-co	
OK Cancel Help	Trace Configuration Analyss		0:00:00:00	↓ ↓

Using CANdb++ Editor to create the database

Click the CANdb++ Editor button $rac{W}{W}$ on the top bar to start the CANdb++ Editor. Then, click File->Create Database. Select CANTemplate.dbc. Name the dbc file and select the desired path to save the database.

Template X	🙀 Vector CANdb++ Editor - tutorial.dbc - [Overall View]		>	
Template for: CANdb Network (.dbc) 🗸	開き Edit View Options Window Help ゆ 日 中 名 信 回 え 日 回 回 同 総 名 日 000000000000000000000000000000000		- 8	×
RAttribute_template_ASR_NM.dbc Attribute_template_OSEK_NM.dbc CANceTemplate.dbc CAN_FD Template.dbc CAN_FD Template.dbc CAN_FD Template.dbc EmplyTemplate.dbc Vector_IL_Basic Template.dbc	₩ Networks Name Protocol Comment BusType ₩ tutorial ₩ tutorial CAN CAN* ₩ stages ~ Signals Signals Signals			
OK Cancel Help	1 Network(s) Ready	NUM		-4

Click the CANdb++ Editor button $rac{1}{2}$ on the top bar to start the CANdb++ Editor. Then, click File->Create Database. Select CANTemplate.dbc. Name the dbc file and select the desired path to save the database.

Right click Network nodes->New to define the network nodes. In this step, the only thing you need to do is to name the nodes. Here we will create 2 nodes. Simply named as Node_A and Node_B. After the nodes are created, you will see that two new nodes are listed under the Network nodes content.

Definition 5	Mapped Tx Sig.	S Manned	BxSin [54] T	x Messages	W Networks	Control units	Attributes	Comment	
	•	P					2		
Name:	Node_A								
Address:	0x0								
					_				
						OK	Cancel	Apply	Help
Norther C		And a start of the	(O	- 4					
	ANdb++ Editor -			v]				-	
Eile Edi	t <u>V</u> iew <u>O</u> ption	s <u>W</u> indow	Help					-	- 5
🛊 Eile 🛛 Edi 🗳 🔜 📭 I	t ⊻iew Option n©alioni na Arika Ins	is <u>W</u> indow . ∎ ⊞ ⊡ E	Help		Comme	ent		-	
Eile Edi Eile Edi Eile Edi Eile Edi Eile Edi Eile Edi Eile Edi Eile Edi Edi Eile Edi Edi Edi Edi Edi Edi Edi Edi Edi Edi	t View Option R C R R R R R R rks rial	s <u>W</u> indow	Help 3 ∰ ¶e ⊨		Comme	ent		-	
Eile Edi Eile Edi W Networ W Networ	t <u>V</u> iew Option R <u>3</u> R <u>2</u> <u>5</u> rks rial rk nodes	s Window	Help 🎊 🖷 🔳 Ime	Address	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto	t ⊻iew Option ■ I III III III III III rks rial rk nodes Ie_A	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod Messag	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod Messag	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod Messag	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod Messag	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	ent		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod Messag	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	nt		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	nt		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -
Eile Edi Eile Edi Networ E W tuto Networ E Nod E Nod	t View Option Compared to the second	s Window	Help I & I I IIIIIIIIIIIIIIIIIIIIIIIIIIIIII	Address 0x0	Comme	int		-	1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 - 1990 -

After the nodes are added, the next step is to define the CAN message. Right click Message->New. In the pop-up window, you need to define the message name, ID(mailbox ID), DLC(data length code), etc.

Message 'New	_Message_1 (0x0)'	×
Definition 🕾	Signals 👤 Transmitters 👤 Receivers Layout 🏹 Attributes Comment	
Name:	Message_A	
Type:	CAN Standard	\sim
ID:	DxAA DLC: 8	
Transmitter:	- No Transmitter -	\sim
Tx Method:	<n.a.></n.a.>	\sim
Cycle Time:	0	
	OK Cancel Apply Help	

Under the "Transmitters" tab, click "Add" button to add the Node_A as the transmission node. Which means, this message is send out form Node_A. Another way to do this is simply drag the message to the desired node to add this node as a transmission node.

efinition 🕤 Signals 💻 Transmitters 💻 Re	eceivers Layout Attributes Comment	
Name Address	Choose Objects	×
	Filter by: Name ~	
	Value: Node_A	<u>F</u> ilter
	Name Address	
	■ Node_A 0x0 ■ Node_B 0x0	
Add Remove	OK Cancel	Help

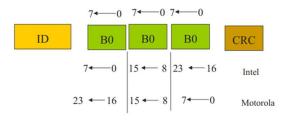
As the message is defined, it's time to include some signals in the message. To create a signal. Right click Signal->New, fill out the pop-up window as shown below. Here we will define a signal that represents the switching stauts, which only has two values, 0(OFF) or 1(ON). Thus, 1 bit data length is enough.

Signal 'New_Sig	gnal_2'						×
Definition 🖂 N	vlessages 👤 Receive	rs 🍸 Attribut	es Value Descriptions	Comment			
Name:	Switch_1A						
Length [Bit]:	1						
Byte Order:	Intel ~	Unit					
Value Type:	Unsigned V	Init. Value:)				
Factor:	1	Offset:)				
Minimum:	0	Maximum:)				
Value Table:	<none></none>		~				
🗸 Automatic n	nin-max calculation						
				ОК	Cancel	Apply He	lp

To put this signal to the message we just created, switch to the "Message" tab, add Message_A. Of course, drag the signal to Message A also works.

minaon	\boxtimes	Messages	L Receivers	Attributes Value Descrip	ptions Comment				
Name	ID	ID-Format	DLC [Byte]	Choose Ob	ojects				×
				Filter by:	Name			~	
				Value:	Message_A (C	IXAA)			<u>F</u> ilter
				Name	ID	ID-Format	DLC		
						ОК	Canc		Help

Note that in the signal definition window, the "Byte order" has two options, "Intel" and "Motorola", the difference between these two format can be identified as below:



You might notice that using 0 and 1 to represent OFF and ON is not very intuitive. Why not directly use "On" and "Off" to define the signal value? You can! On the top bar of CANdb++ Editor, click View->Value Tables, right click the blank area->New. Type the information as shown below, be sure to save the work.

	ew_Value_Table_1'					×
Definition Valu	e Descriptions					
Name:	Switch_Positions					
Comment		^				
		~				
			OK	Cancel	Apply	Help
lue Table 'Ne	aw Value Table 1'					
	ew_Value_Table_1'					>
efinition Valu	e Descriptions					>
efinition Valu Value	e Descriptions					>
efinition Valu	e Descriptions					, >
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					>
efinition Valu Value 0x0	e Descriptions Description Off					
efinition Valu Value 0x0	e Descriptions Description Off					
efinition Valu Value 0x0	e Descriptions Description Off					
efinition Valu Value 0x0	e Descriptions Description Off		∆dd	Bemove		
efinition Valu Value 0x0	e Descriptions Description Off		Add	Bemove		

After the value table is defined. It has to be associate to the signal. Right click the signal that need to be associated with(Switch_1A)->Edit Signal. Under the defination tab, unfold drop-downlist of Value Table. You will see the value table you just created, select that table. Then your signal defination is done.

gnal 'Switch_	1A'						
efinition 🖂 I	Messages 👤 Receive	rs 🍸 Attrib	utes Value Description	s Comment			
Name:	Switch_1A						
ength [Bit]:	1						
lyte Order:	Intel ~	Unit:					
'alue Type:	Unsigned \vee	Init. Value:	0				
actor:	1	Offset:	0				
4inimum:	0	Maximum:	0				
'alue Table:	<none></none>		~				
Automatic r	<mark>≺none></mark> mi Switch_Positions						
_							
				OK	Cancel	Apply	Help

Follow the steps we've done to create another signal. You can right click the signal that already crated under the root directory of "Signals", select "copy", and then paste it. Double click the pasted signal, rename it as "Switch_2A". Use the same method to associate this signal to Message_A as we did before, which is very efficient.

Let's continue to create the third signal, "Indicator_A", same as what we did before.

👯 Overall View		1				1.11		
Wetworks	Name	Message	Multiplexin	Star	Len	Byte Order	Value Type	Ini
	Switch_1A	Message_A	2	0	1	Intel	Unsigned	0
Network nodes	& Switch_2A	Message_A	<	1	1	Intel	Unsigned	0
	Indicator_A	Message_A	5	2	1	Intel	Unsigned	0
Messages								
Message_A (0xAA)								
Switch 1A								
- Switch_2A								
- [□] Indicator_A								
∼ Signals								
i → Indicator_A								
i → Switch_1A								
ia → Switch_2A								
dy	1						NUM	

Since we have multiple signals in Message_A now, how to arrange these signals in the message remains to be done. Message_A is defined as a 8-byte data. Assume we want the first bit of Message_A represents the signal "Switch_1A", and the second bit represents "Switch_2A", make the last bit represents "Indicator_A". There are two approach to make it.

First, right click the signals that has already been associated to Message_A, under the root directory of messages. Click edit mapped signal, in the pop-up window, fill the "Startbit" with the position you want the signal begins with. For example, for "Switch_1A", fill in 1. For "Switch_2A", fill in 2. For "Indicator_A", fill in 56.

Message Signal 'Me	ssage_A::Switch_1A'	×
Definition Message	Signal 👤 Receivers Value Descriptions 🗹 Attributes Comment	
Message Name:	Message_A	
Signal Name:	Switch_1A	
Startbit [Bit]:	1	
Multiplexortype	Signal	
	OK Cancel Apply	Help

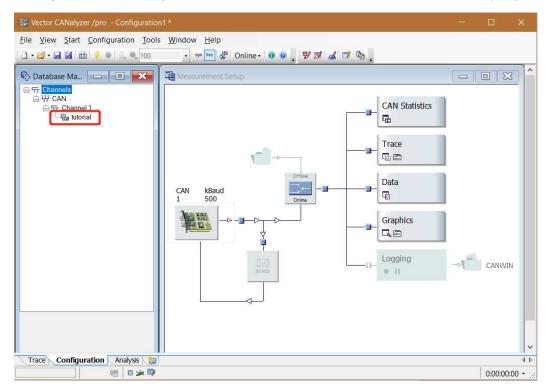
Second method is, right click Message_A-> Edit Message. Go to the "Layout" tab. You can drag the colored blocks, which represents different signals in this message, to relocate the position.

efinition 🔂 Signals	Transmitters	s 👤 Receive	rs Layout	🛃 Attributes	Comment				
/ultiplexor Signal:	– No Multiplexor	-						.05	~
		7	6	5	4	3	2	1	0
	0	witch_1A Sv	ritch_2A	5	4	3	2	1	0
	1	15	14	13	12	11	10	9	8
	2	23	22	21	20	19	18	17	16
Arrange	3	31	30	29	28	27	26	25	24
To <u>F</u> ront	4	39	38	37	36	35	34	33	32
To <u>B</u> ack	5	47	46	45	44	43	42	41	40
Add <u>R</u> emove	6	55	54	53	52	51	50	49	48
Bit index	7	63	62	61	60	59	58	57	ator_A 56 v

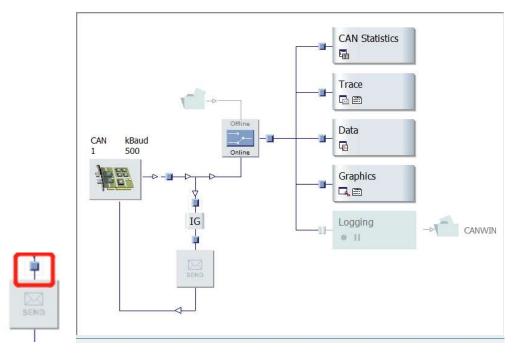
By now, you've succesfully created a complete message: Message_A. Don't forget to save the database.

Configure the CAN UI

With the database successfully created, let's go back to the CANalyzer main interface. On the left side, the database management window, right click Channel 1->Add Database, select the database you just saved.



On the right side, the Measurement Setup window, in the SEDN loop, right click the square node in the loop->Insert Interactive Generator Block. This node is help you to send command to the controller though CAN bus.



		Msg Par	rams		Triggerin	ig			1	Data F	eld		1				
	Message Name	Channel	DLC	Send	Cycle T	ime (ms)	0	1	2	3	4 5	6	7				
													-				
		17 I c															>
1	(orandard) ora	215															1
	New		lone		Special Fram	•	Del	oto		1 1		Out		Com	Pacto	Levout	
	New	•] (Clone		Special Fram	e •	Del	0.00				Cut		Сору	Paste 🗸	Layout	
	New Signal Name	Raw Va		11.	Special Fram Value	e 🔹		0.00	s Step	o Inc				Copy	Paste V	Layout	
	Sectors 10			11.				0.00	Step	o Inc				4.W	Paste	Layout	
2000 0 2000	Sectors 10			11.				0.00	s Step	o Inc				4.W	Paste -	Layout	
	Sectors 10			11.				0.00	s Step	o Inc				4.W	Paste 🗸	Layout	
	Sectors 10			11.				0.00	s Step	o Inc				4.W	Paste 🗸	Layout	
	Sectors 10			11.				0.00	s Step) Inc				4.W	Poste V	Layout	
	Sectors 10			11.				0.00	s Step	o Inc				4.W	Poste V	Layout	
	Sectors 10			11.				0.00	s Step	o Inc				4.W	Peste 🔻		
3	Sectors 10			11.				0.00	a Step					4.W	Peste	Layout	se

Double click the IG block, you will see interface below:

Double click the first row, select the Message_A we created in database. The included message and signals will appear on the IG interface. On the right side of Message_A, it gives the related setting. When sending this message, you can either click "now" to send it once, or check the box right after "now" button to make this message been send periodically. The cycle time could be adjusted also.

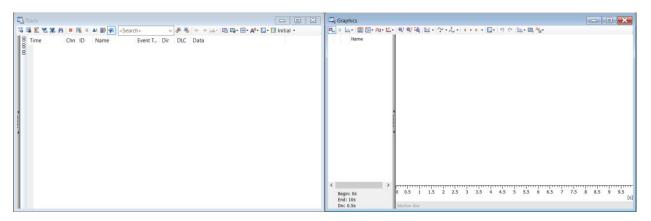
On the bottom part. Three signals that be involved in Message_A are also listed. You can choose the value or status of each signals, these settings will been send out simultaneously as the message.

ig IG														Selection of Messages	s and Bus Events			×
Message Name	Msg Pa			Trigge		- I			Data F									
	Channel		Send		e Time [ms]	0		2				7		Enter a search term				~ #8
> tutorial:Messad	CAN 1	8 -	now	10		0	0	0	0	0 0	0	0		Name Utorial Messages Nodes Nodes Node_B Node_B	Tx Node Node_A	Channel	Id Comment	
 Standard (CAN New SB Signal Name 	Raw\		Phy	Special Fra	ame 👻			s Ster	p Inc				Copy generation					
56 Indicator_A 6 Switch_2A 7 Switch_1A			Off Off	•		-	1		+	1	None None None		Define Define					
															tutorial 0xAA <u>O</u> K	Gancel	Δρρίγ	> Help
Default input mode: CAN																, ci		

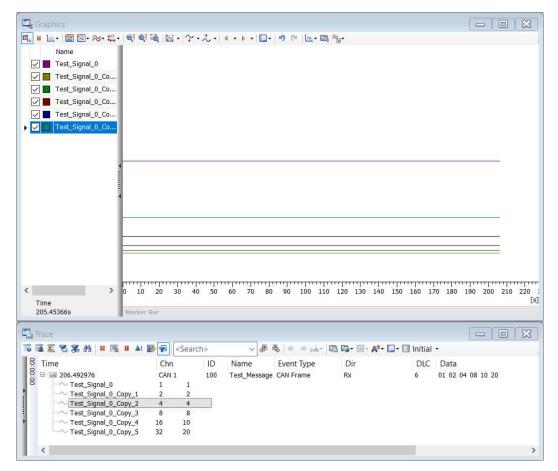
Signal Monitoring

We definitely want to see the signals that coming back from the controller. To monitor these data, two tools are often used.

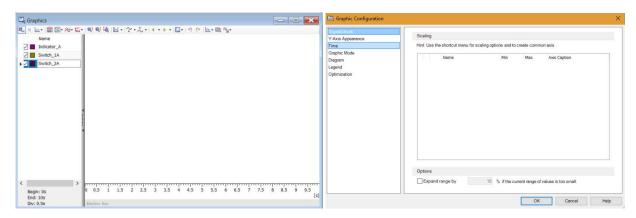
On the Measurement Setup window, double click "Tace" and "Graphics", you will find that two more windows will appear on your interface as below:



On the Trace window, you don't have to do anything, because once the connection is built, during the data communication between controller and PC, the defined messages and signals will automatically appear on this window and refresh with every communication cycle. An example during the communication is as shown below:



On the Graphic window, in order to see the variation trend of specific signal, right click the left side blank area-> Add signals to add the items you want to monitor. Then during the communication, you can observe the real-time plot here on the right. More plot settings, such as axis settings, colors could be found if you double click the right side blank area.



Till now, the configuration part for the CANalyzer side is finished.

Other Related Work

DSP coding:

Follow the document: "Enhanced Controller Area Network (eCAN) Reference Guide". The eCAN module should be configured, all the mailboxes that will be used should have unique message ID.

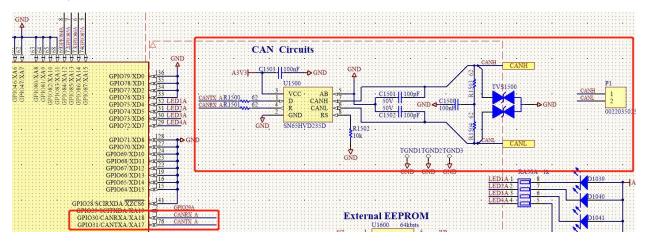
<pre>void CAN_Config(void) {</pre>	
ECanaRegs.CANME.all = 0x00000000;	<pre>//Ensure all mailboxes disabled; required before writing the MSGIDs</pre>
ECanaRegs.CANMD.all = 0xE0008000;	//We will use mailboxes 0-14 & 16-31 as general TRANSMIT mailboxes
ECanaMboxes.MBOX0.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX1.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX2.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX3.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX4.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX5.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX6.MSGID.all	= 0x80000000; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX7.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX8.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX9.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX10.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX11.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX12.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX13.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX14.MSGID.all	= 0; //Use 11-bit ID, No acc Mask, No auto-answer
ECanaMboxes.MBOX0.MSGID.bit.STDMSGID	= 0x00A0; //Set Std Msg ID value; Product Name
ECanaMboxes.MBOX1.MSGID.bit.STDMSGID	= 0x00A1; //Set Std Msg ID value; SW Revision
ECanaMboxes.MBOX2.MSGID.bit.STDMSGID	= 0x00A2; //Set Std Msg ID value; VIN# Higher byte
ECanaMboxes.MBOX3.MSGID.bit.STDMSGID	= 0x00A3; //Set Std Msg ID value; VIN# Lower Byte
ECanaMboxes.MBOX4.MSGID.bit.STDMSGID	= 0x00A4; //Set Std Msg ID value; Security Response
ECanaMboxes.MBOX5.MSGID.bit.STDMSGID	= 0x00A5; //Set Std Msg ID value; Vehicle Status
ECanaMboxes.MBOX6.MSGID.bit.EXTMSGID_L	
ECanaMboxes.MBOX7.MSGID.bit.STDMSGID	= 0x00A7; //Set Std Msg ID value; Vehicle Energy trip
ECanaMboxes.MBOX8.MSGID.bit.STDMSGID	= 0x00A8; //Set Std Msg ID value; Vehicle Energy Life
ECanaMboxes.MBOX9.MSGID.bit.STDMSGID	= 0x00A9; //Set Std Msg ID value; Motor Temps
ECanaMboxes.MBOX10.MSGID.bit.STDMSGID	= 0x00AA; //Set Std Msg ID value; Motor Current
ECanaMboxes.MBOX11.MSGID.bit.STDMSGID	= 0x00AB; //Set <u>Std Msg</u> ID value; INV_internal status
ECanaMboxes.MBOX12.MSGID.bit.STDMSGID	= 0x00AC; //Set <u>Std Msg</u> ID value; Performance Setting
ECanaMboxes.MBOX13.MSGID.bit.STDMSGID	= 0x0623; //Set Std Msg ID value; Battery Voltages
ECanaMboxes.MBOX14.MSGID.bit.STDMSGID	= 0x0624; //Set Std Msg ID value; Battery Currents

void SendDataCAN(void)

{

<pre>if (CpuTimer1.InterruptCount % 1 ==0)</pre>	
{ ECanaMboxes.MBOX3.MDL.word.HI_WORD = (int16)(FAULT_PFC);//	//FaultNo
ECanaMboxes.MBOX3.MDL.word.LOW_WORD = (int16)(IGA_FLT.Out);//wga	//Ia
ECanaMboxes.MBOX3.MDH.word.HI_WORD = (int16)(IGB_FLT.Out);//picd	//Ib
ECanaMboxes.MBOX3.MDH.word.LOW_WORD = (int16)(IGC_FLT.Out);//picg	//Ic
ECanaRegs.CANTRS.all = 0x0008;	
<pre>//ECanaMboxes.MBOX4.MDL.word.HI_WORD = (int16)(sys.RunStatus);//pidc.err*10</pre>	//Ua
ECanaMboxes.MBOX4.MDL.word.HI_WORD = (int16)(UGA_FLT.Out);	
ECanaMboxes.MBOX4.MDL.word.LOW_WORD = (int16)(UGB_FLT.Out);//pllangle	//Ub
ECanaMboxes.MBOX4.MDH.word.HI_WORD = (int16)(UGC_FLT.Out);//dqugD	//Uc
ECanaMboxes.MBOX4.MDH.word.LOW_WORD = (int16)(VDC_FLT.Out);//dqugQ	//PLL_Angle
ECanaRegs.CANTRS.all = 0x0010;	

Circuit layout:

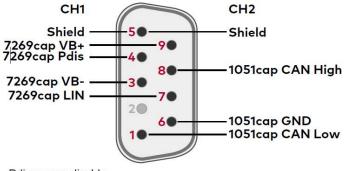


CAN interface:

D-SUB9 connector

The pin assignment of the D-SUB9 connector (CH1 and CH2) is as follows:

CH1/CH2



Pdis: power disable