

FCAN-View ソフト 設計書

0.09 版 : 2011年09月21日

版数管理表

版数	日 付	内 容	担 当
0.00	2011/02/18	暫定版作成	
0.01	2011/03/31	GUI 変更	
0.02	2011/04/12	送受信変更。Graph 変更	
0.03	2011/05/20	PcanGui->FCAN_View 変更	
0.04	2011/06/06	TxData,RxDataFormat 変更	
0.05	2011/08/05	前面修正	
0.06	2011/08/17	誤植修正。Format 画面変更。Read 画面変更。	
0.07	2011/09/12	Read 画面変更。Write 画面変更。	
0.08	2011/09/19	誤植修正。構成変更。	
0.09	2011/09/21	軽微な誤植修正	

目次

1. 適用	1
2. 関連仕様書・関連規格	1
3. 概要	1
4. 開発環境	1
5. 実行環境セットアップ	2
5.1. 再配布可能パッケージ	2
5.2. PCAN-USBハードウェアセットアップ	2
6. 取扱説明	3
6.1. 全設定の保存と読み出し	3
6.2. ハードウェアの設定動作	4
6.3. ライトメッセージ動作	7
6.4. リードメッセージ動作	11
6.5. グラフ表示動作	15
7. 構造(クラス)構成	21
8. FCAN-Viewi-Qwidget	22
8.1. Qwidget(ウィンドウ)	22
8.2. Qwidget(ウィンドウ)メニュー	22
8.3. メンバー関数	23
8.3.1. void FcanViewSet(FCAN_View * fcanview)	23
8.3.2. void SysConfigSet(SysConfig * sysconfig)	24
8.3.3. void InitFileSet(InitFile * initfile)	24
8.3.4. void HardwareSet(Hardware * hardware)	24
8.3.5. void RxDataSet(RxData * rxdata)	24
8.3.6. void TxDataSet(TxData * txdata)	24
8.3.7. void RxDataFormatSet(RxDataFormat * rxdataformat)	24
8.3.8. void RxDataFilterSet(RxDataFilter * rxdatafilter)	25
8.3.9. void GraphSet(Graph * graph)	25
8.3.10. void PlotterSet(Plotter * plotter)	25
8.3.11. void PlotSettingsSet(PlotSettings * plotsettings)	25
8.3.12. void InformationSet(Information * information)	25
8.3.13. void CanDataSet(CanData * candata)	25
8.3.14. void CanDataMngSet(CanDataMng * candatamng)	26
8.3.15. void CanFrameSet(CanFrame * canframe)	26
8.3.16. void Start()	26
8.3.17. void HardwareSetting(QString qstr)	26
8.3.18. void statusBar_message(QString qstr)	26
8.3.19. void statusBar_clear()	26
8.3.20. void HardwareView(int mode)	27
8.3.21. void RxDataFilterView(int mode)	27
8.3.22. void RxDataFormatView(int mode)	27
8.3.23. void RxDataView(int mode)	27
8.3.24. void TxDataView(int mode)	27
8.3.25. void GraphView(int mode)	27
8.3.26. void GraphViewShow(int no)	28
8.3.27. void GraphViewClose(int no)	28
8.3.28. QString GraphInit(int no, int mode, QString line)	28
8.3.29. void PlotterView(int mode)	28
8.3.30. void createMenus()	28

8.3.31.	void ViewAdd()	28
8.4.	スロット宣言	29
8.4.1.	void menu_initfile()	29
8.4.2.	void menu_hardware()	29
8.4.3.	void menu_rxdata()	30
8.4.4.	void menu_txdata()	30
8.4.5.	void menu_rxdataformat()	30
8.4.6.	void menu_rxdatafilter()	30
8.4.7.	void menu_newgraph()	30
8.4.8.	void menu_graph0()	30
8.4.9.	void menu_graph1()	31
8.4.10.	void menu_graph2()	31
8.4.11.	void menu_graph3()	31
8.4.12.	void menu_graph4()	31
8.4.13.	void menu_graph5()	31
8.4.14.	void menu_graph6()	31
8.4.15.	void menu_graph7()	31
8.4.16.	void menu_graph8()	32
8.4.17.	void menu_graph9()	32
8.4.18.	void menu_information()	32
8.4.19.	void menu_close()	32
8.4.20.	void menu_triggered()	32
8.4.21.	void closeEvent(QCloseEvent * pEvent)	32
8.5.	データメンバー宣言	33
9.	InitFile-Qwidget	34
9.1.	Qwidget(ウィンドウ)	34
9.2.	メンバー関数	35
9.2.1.	void FcanViewSet(FCAN_View * fcanview)	35
9.2.2.	void HardwareSet(Hardware * hardware)	35
9.2.3.	void RxDataFormatSet(RxDataFormat * rxdataformat)	35
9.2.4.	void RxDataFilterSet(RxDataFilter * rxdatafilter)	35
9.2.5.	void TxDataSet(TxData * txdata)	36
9.2.6.	void InformationSet(Information * information)	36
9.3.	スロット宣言	37
9.3.1.	void closeEvent(QCloseEvent * pEvent)	37
9.3.2.	void btnInitLoadFileFind_Click()	37
9.3.3.	void btnInitLoad_Click()	38
9.3.4.	void btnInitSaveFileFind_Click()	38
9.3.5.	void btnInitSave_Click()	39
9.4.	データメンバー宣言	39
10.	Hardware-Qwidget	40
10.1.	Qwidget(ウィンドウ)	40
10.2.	メンバー関数	41
10.2.1.	void FcanViewSet(FCAN_View * fcanview)	41
10.2.2.	void InformationSet(Information * information)	41
10.2.3.	TPCANStatus timRead(TPCANMsg* MessageBuffer, TPCANTimestamp* TimestampBuffer)	42
10.2.4.	TPCANStatus btnWrite_Click(TPCANMsg* MessageBuffer)	42
10.2.5.	TPCANStatus btnSetFilter_Click(DWORD FromID, DWORD ToID, TPCANMode Mode)	42
10.2.6.	TPCANStatus btnResetFilter_Click(DWORD FromID, DWORD ToID, TPCANMode Mode)	43
10.2.7.	QString HardWareInit(int mode, QString line)	43
10.2.8.	void ListenSet(int mode)	44
10.2.9.	void ResetSet()	44

10.2.10.	void createMenus()	44
10.2.11.	void createMenuActions()	44
10.2.12.	int ValueCheck(QString ValueString)	44
10.3.	スロット宣言	45
10.3.1.	void closeEvent(QCloseEvent * pEvent)	45
10.3.2.	void btnDllInfo_Click()	45
10.3.3.	void btnGetDevNumber_Click()	46
10.3.4.	void btnGetDevNumber_Click()	46
10.3.5.	void btnRelease_Click()	47
10.3.6.	void btnInit_Click()	47
10.3.7.	void btnInfo_Click()	48
10.3.8.	void menu_triggered()	48
10.3.9.	void menu_loadfile()	49
10.3.10.	void menu_savefile()	49
10.3.11.	void cbbHws_Changed(const QString &s)	50
10.4.	データメンバー宣言	50
11.	RxDataFilter-Qwidget	51
11.1.	Qwidget(ウインドウ)	51
11.2.	メンバー関数	52
11.2.1.	void FcanViewSet(FcanView * fcanview)	52
11.2.2.	void InformationSet(Information * information)	52
11.2.3.	void HardWareSet(HardWare * hardware)	52
11.2.4.	QString RxDataFilterInit(int mode, QString line)	53
11.2.5.	void createMenus()	53
11.2.6.	void createMenuActions()	53
11.2.7.	int ValueCheck(QString ValueString)	53
11.3.	スロット宣言	54
11.3.1.	void closeEvent(QCloseEvent * pEvent)	54
11.3.2.	void btnHWFilterSetReset_Click()	55
11.3.3.	void btnSWFilterSetReset_1_Click()	56
11.3.4.	void menu_loadfile()	57
11.3.5.	void menu_savefile()	57
11.4.	データメンバー宣言	58
12.	RxDataFormat-Qwidget	59
12.1.	Qwidget(ウインドウ)	59
12.2.	メンバー関数	60
12.2.1.	void FcanViewSet(FCAN_View * fcanview)	60
12.2.2.	void InformationSet(Information * information)	60
12.2.3.	void CanFrameSet(CanFrame * canframe)	60
12.2.4.	QString RxDataFormatInit(int mode, QString line)	61
12.2.5.	void ViewFormat()	61
12.2.6.	void createMenus()	62
12.2.7.	void createMenuActions()	62
12.2.8.	void DelFormat()	62
12.2.9.	void DelFormatAll()	63
12.2.10.	void ViewFormatWrite()	63
12.2.11.	int ValueCheck(QString ValueString)	63
12.3.	スロット宣言	64
12.3.1.	void closeEvent(QCloseEvent * pEvent)	64
12.3.2.	void btnSetFormat_Click()	64
12.3.3.	void btnOverWriteFormat_Click()	65
12.3.4.	void btnDeleteFormat_Click()	65

12.3.5.	void btnDeleteAllFormat_Click()	65
12.3.6.	void menu_triggered()	65
12.3.7.	void menu_loadfile()	66
12.3.8.	void menu_savefile()	66
12.3.9.	void tblView_CellClicked(int row, int column)	67
12.3.10.	void tblView_CellDoubleClicked(int row, int column)	67
12.3.11.	void resizeEvent(QResizeEvent *event)	67
12.3.12.	void keyPressEvent(QKeyEvent *event)	67
12.4.	データメンバー宣言	67
13.	TxDat-Qwidget	68
13.1.	Qwidget(ウインドウ)	68
13.2.	メンバー関数	69
13.2.1.	void FcanViewSet(FCAN_View * fcanview)	69
13.2.2.	void InformationSet(Information * information)	69
13.2.3.	void HardWareSet(HardWare * hardware)	69
13.2.4.	TPCANStatus Write_Message(TPCANMsg* MessageBuffer)	69
13.2.5.	QString TxDataInit(int mode, QString line)	70
13.2.6.	void SetMessage()	70
13.2.7.	void createMenus()	70
13.2.8.	void createMenuActions()	70
13.2.9.	int ValueCheck(QString ValueString)	70
13.3.	スロット宣言	71
13.3.1.	void closeEvent(QCloseEvent * pEvent)	71
13.3.2.	void btnWrite_Click()	71
13.3.3.	void btnWriteFileFind_Click()	72
13.3.4.	void btnWriteFile_Click()	72
13.3.5.	void btnSetMessage_Click()	72
13.3.6.	void btnOverWriteMessage_Click()	73
13.3.7.	void btnDelMessage_Click()	73
13.3.8.	void chbTxTimer_Toggled()	73
13.3.9.	void cbbTxLength_Changed(const QString &s)	73
13.3.10.	void chbRemoteRequest_Toggled()	73
13.3.11.	void chbDisplay_Toggled()	73
13.3.12.	void tblTxData_CellClicked(int row, int column)	74
13.3.13.	void tblTxData_CellDoubleClicked(int row, int column)	74
13.3.14.	void timWrite()	74
13.3.15.	void menu_triggered()	75
13.3.16.	void menu_loadfile()	75
13.3.17.	void menu_savefile()	75
13.3.18.	void resizeEvent(QResizeEvent *event)	76
13.3.19.	void keyPressEvent(QKeyEvent *event)	76
13.4.	データメンバー宣言	76
14.	RxDat-Qwidget	77
14.1.	Qwidget(ウインドウ)	77
14.2.	メンバー関数	78
14.2.1.	void FcanViewSet(FCAN_View * fcanview)	78
14.2.2.	void InformationSet(Information * information)	78
14.2.3.	void HardWareSet(HardWare * hardware)	78
14.2.4.	void SysConfigSet(SysConfig * sysconfig)	79
14.2.5.	void RxDataFilterSet(RxDataFilter * rxdatafilter)	79
14.2.6.	void CanDataSet(CanData * candata)	79
14.2.7.	void CanDataMngSet(CanDataMng * candatamng)	79

14. 2. 8.	TPCANStatus Read_Message(TPCANMsg* MessageBuffer, TPCANTimestamp* TimestampBuffer)	80
14. 2. 9.	void RxDataFileWrite(DWORD Count)	80
14. 2. 10.	void ProcessMessage(TPCANMsg MyMsg, TPCANTimestamp MyTimeStamp)	81
14. 2. 11.	void RxDataTime(TPCANTimestamp MyTimeStamp)	81
14. 3.	スロット宣言	82
14. 3. 1.	void closeEvent(QCloseEvent * pEvent)	82
14. 3. 2.	void btnRxStart_Click()	82
14. 3. 3.	void btnRxStop_Click()	83
14. 3. 4.	void btnRxDataFind_Click()	83
14. 3. 5.	void chbViewReport_Toggled()	83
14. 3. 6.	void timRead()	84
14. 3. 7.	void timDisplay()	84
14. 4.	データメンバー宣言	85
15.	Graph-Qwidget	86
15. 1.	Qwidget(ウインドウ)	86
15. 2.	メンバー関数	87
15. 2. 1.	void FcanViewSet(FCAN_View * fcanview)	88
15. 2. 2.	void InformationSet(Information * information)	88
15. 2. 3.	void SysConfigSet(SysConfig * sysconfig)	88
15. 2. 4.	void RxDataSet(RxData * rxdata)	88
15. 2. 5.	void RxDataFormatSet(RxDataFormat * rxdataformat)	88
15. 2. 6.	void PlotterSet(Plotter * plotter)	88
15. 2. 7.	void PlotSettingsSet(PlotSettings * plotsettings)	89
15. 2. 8.	void CanDataSet(CanData * candata)	89
15. 2. 9.	void CanDataMngSet(CanDataMng * candatamng)	89
15. 2. 10.	void CanFrameSet(CanFrame * canframe)	89
15. 2. 11.	void GraphNumberSet(int no)	89
15. 2. 12.	QString GraphInit(int mode, QString line)	90
15. 2. 13.	double EndScroll()	90
15. 2. 14.	int GetId(int Select)	90
15. 2. 15.	int GetFunc(int Select)	90
15. 2. 16.	int GetItem(int Select)	90
15. 2. 17.	void chbGraph1_Load()	91
15. 2. 18.	void chbGraph2_Load()	91
15. 2. 19.	void chbGraph3_Load()	91
15. 2. 20.	void chbGraph4_Load()	91
15. 2. 21.	void chbGraph5_Load()	91
15. 2. 22.	void cbbGraphId1_Load()	92
15. 2. 23.	void cbbGraphId2_Load()	92
15. 2. 24.	void cbbGraphId3_Load()	92
15. 2. 25.	void cbbGraphId4_Load()	92
15. 2. 26.	void cbbGraphId5_Load()	92
15. 2. 27.	void cbbGraphFunc1_Load()	93
15. 2. 28.	void cbbGraphFunc2_Load()	93
15. 2. 29.	void cbbGraphFunc3_Load()	93
15. 2. 30.	void cbbGraphFunc4_Load()	93
15. 2. 31.	void cbbGraphFunc5_Load()	93
15. 2. 32.	void PlotterSetting()	93
15. 2. 33.	void GraphReSize()	94
15. 2. 34.	void createMenus()	94
15. 2. 35.	void createMenuActions()	94
15. 2. 36.	int ValueCheck(QString ValueString)	94

15.3. スロット宣言	95
15.3.1. void closeEvent(QCloseEvent * pEvent)	95
15.3.2. void btnGraphDraw_Click()	96
15.3.3. void btnGraphClose_Click()	97
15.3.4. void btnGraphStartStop_Click()	97
15.3.5. void btnGraphFileFind_Click()	98
15.3.6. void btnGraphFileLoad_Click()	98
15.3.7. void rdbGraphRecive_Toggled()	98
15.3.8. void chbGraph1_Toggled()	99
15.3.9. void cbbGraphId1_Changed(const QString &s)	100
15.3.10. void cbbGraphFunc1_Changed(const QString &s)	101
15.3.11. void chbYaxis2_Toggled()	102
15.3.12. void chbYaxis3_Toggled()	102
15.3.13. void chbYaxis4_Toggled()	102
15.3.14. void chbYaxis5_Toggled()	102
15.3.15. void timGraph()	103
15.3.16. void menu_triggered()	103
15.3.17. void menu_loadfile()	104
15.3.18. void menu_savefile()	104
15.4. データメンバー宣言	105
16. CanData-Class	106
16.1. メンバー関数	106
16.2. スロット宣言	106
16.3. データメンバー宣言	106
17. CanDataMng-Class	107
17.1. メンバー関数	107
17.1.1. int num(void)	107
17.1.2. void set(CanData &canData)	107
17.1.3. int get(int no, CanData &canData)	107
17.1.4. bool isFull(void)	107
17.1.5. void clear(void)	107
17.2. スロット宣言	108
17.3. データメンバー宣言	108
18. CanFrame-Class	109
18.1. メンバー関数	109
18.1.1. CanFrameNode * nodeAdd(int nodeID, QString name)	109
18.1.2. CanFrameNode * nodeSearch(int nodeID)	109
18.1.3. CanFrameFunction * functionSearch(int nodeID, int functionCode)	109
18.1.4. int nodeNum(void)	109
18.1.5. int nodeID(int index)	110
18.1.6. void nodeName(int index, QString &)	110
18.1.7. void nodeDelete(int index)	110
18.1.8. void nodeDeleteId(int id)	110
18.2. スロット宣言	110
18.3. データメンバー宣言	110
19. CanFrameNode-Class	111
19.1. メンバー関数	111
19.1.1. CanFrameFunction * functionAdd(int functionCode, QString functionName)	111
19.1.2. CanFrameFunction * functionSearch(int functionCode)	111
19.1.3. int functionNum(void)	111
19.1.4. int functionCode(int index)	111
19.1.5. void functionName(int index, QString &)	112

19.1.6.	void functionDelete(int index)	112
19.1.7.	void functionDeleteFunc(int code)	112
19.2.	スロット宣言	112
19.3.	データメンバー宣言	112
20.	CanFrameFunction-Class	113
20.1.	メンバー関数	113
20.1.1.	int formatAdd(CanFrameFormat * pFormat)	113
20.1.2.	int formatNum(void)	113
20.1.3.	CanFrameFormat * formatGet(int no)	113
20.1.4.	void formatName(int no, QString &)	113
20.1.5.	int formatNo(int index)	114
20.1.6.	void formatDeleteIndex(int index)	114
20.1.7.	void formatDeleteItem(int no)	114
20.1.8.	int code(void)	114
20.2.	スロット宣言	114
20.3.	データメンバー宣言	114
21.	CanFrameFormat-Class	115
21.1.	メンバー関数	115
21.1.1.	CanFrameFormatType typeGet(void)	115
21.1.2.	int nameGet(QString & value)	115
21.1.3.	int denominationGet(QString & value)	115
21.2.	スロット宣言	116
21.3.	データメンバー宣言	116

1. 適用

本書は、『FCAN-View ソフト Qt 版』（以下、本ソフト）に適用します。

2. 関連仕様書・関連規格

PCAN-Basic V1.04.12	PEAK-System
qt-win-opensource-4.7.2-vs2008.exe	Qt
qt-vs-addin-1.1.9.exe	Qt
QtにてPCan-Basic機能ソフト構成書	

3. 概要

本ソフトは Qt の GUI 上にて PEEK-SYSTEM 社が提供する PCAN-BASIC のライブラリを正常に利用しメッセージの送受信を行い、受信メッセージを指定されたフォーマットにもとづきグラフ表示することが可能です。

4. 開発環境

Qt 4.7.2
qt-sdk-win-opensource-2010.05 ライブラリ
MicroSoft VisualStudio 2010 Professional
PEAK-System PCAN-USB
PCAN-Basic V1.04.12

本ソフトでは、PEAK-SYSTEM 社 PCAN-USB のハードウェアに無償添付されている PCAN-Basic の API を使用しています。PCAN-Basic については、下記 URL を参照してください。

<http://gaillogic.co.jp/product/ps/pdf/PCAN-Basic.pdf>

また、詳細な使用方法についてはハードウェアに添付されている下記ファイルを参照してください。

Downgrade of Visual Studio Projects.pdf

本ソフトでは、WIN32 用 DLL を使用して開発されていますので X64 対応未になります。

5. 実行環境セットアップ

5.1. 再配布可能パッケージ

本ソフトで提供されるファイルは下記のとおりです。

PCAN-View.exe	→	本ソフト実行ファイル
PCANBasic.dll	→	PCANBasicダイナミックリンクライブラリ
QtCore4.dll	→	Qtダイナミックリンクライブラリ
QtGui4.dll	→	Qtダイナミックリンクライブラリ
Images	→	本ソフトで使用するイメージファイル格納ディレクトリ

本ソフトはランタイムライブラリを使用していますので下記URLよりダウンロードしてインストールを実施してください。

<http://www.microsoft.com/downloads/ja-jp/details.aspx?FamilyID=a7b7a05e-6de6-4d3a-a423-37bf0912db84>

5.2. PCAN-USB ハードウェアセットアップ

PEAK-SYSTEM 社 PCAN-USB のハードウェアセットアップについてはPCAN-USB に添付されている ProductCD-ROM 内の下記ファイルを参照してください。

Pdf/PCAN-USB_UserMan_eng.pdf

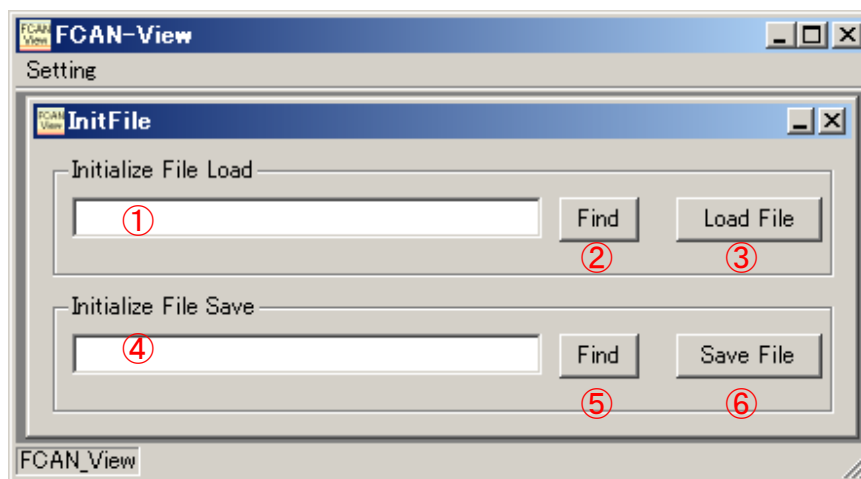
また、ProductCD-ROM の詳細及び使用制限については下記 URL を参照してください。

www.peak-system.com

6. 取扱説明

6.1. 全設定の保存と読み出し

「Setting」メニューから「File」を選択して起動する InitFile ウィンドウにて、各機能で利用されている設定を一括して保存、読み出すことが可能です。
Initialize File Save で、指定したファイルに現在使用されているすべての機能の設定が保存されます。
Initialize File Load で、保存した初期化ファイルをロードすることで、設定が反映されます。

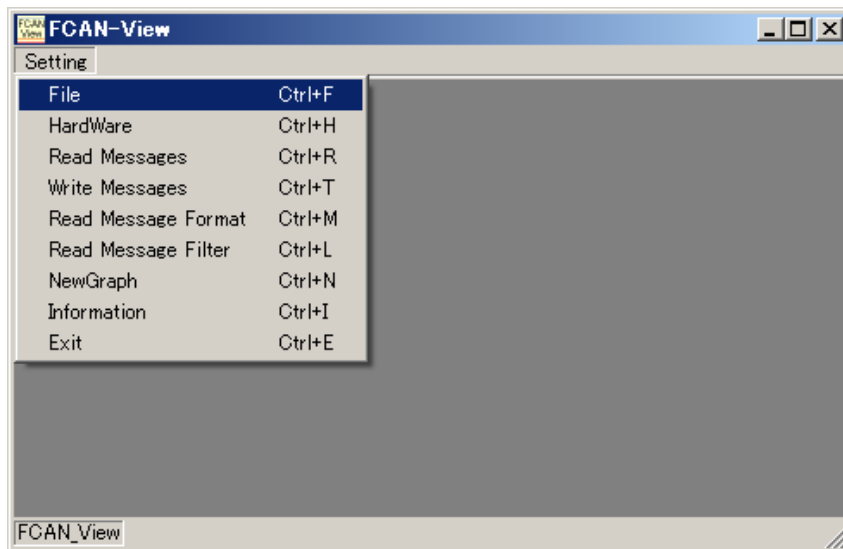


- ① : 設定ファイルを指定します。
- ② : 設定ファイルの検索を行いファイルを指定します。
- ③ : ①にて指定された設定ファイルの記述に従い各設定を実施します。
- ④ : 設定保存ファイルを指定します。
- ⑤ : 設定保存ファイルの検索を行いファイルを指定します。
- ⑥ : ④にて指定された設定ファイルに各設定を保存します。

6. 2. ハードウェアの設定動作

ハードウェアの設定を行う場合、下記の手順になります。

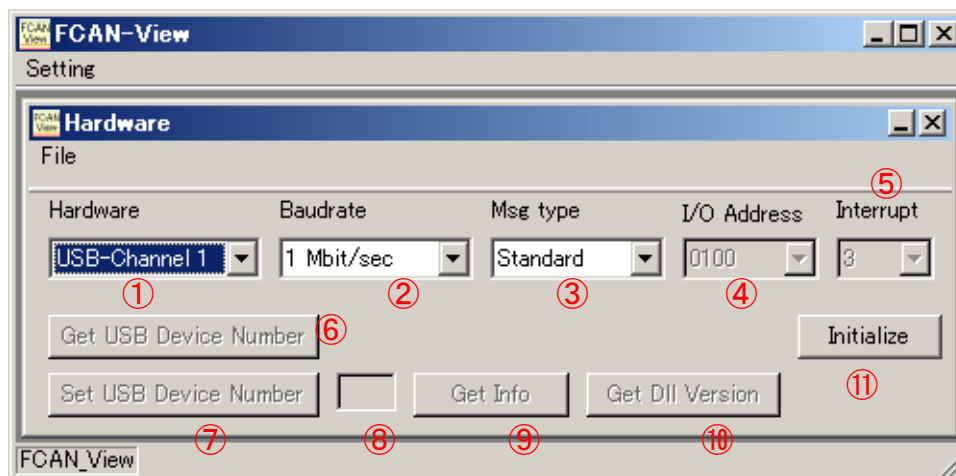
(1) ハードウェアを設定。



「Setting」メニューから「HardWare」を選択しクリックします。その後ハードウェア設定画面が現われます。「Setting」メニューのショートカットは、

No	項目	ショートカットキー
1	File	Ctrl + F
2	HardWare	Ctrl + H
3	Read Message	Ctrl + R
4	Write Message	Ctrl + T
5	Read Message Format	Ctrl + M
6	Read Message Filter	Ctrl + L
7	NewGraph	Ctrl + N
8	Information	Ctrl + I
9	Exit	Ctrl + E

になります。

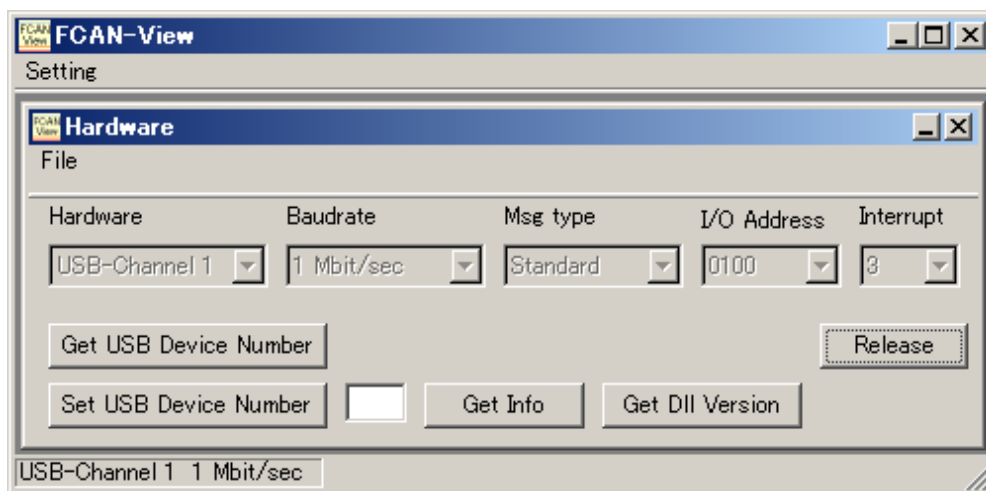


- ① : 使用するハードウェア種別を指定します。
- ② : 転送ボーレートを指定します。
- ③ : メッセージタイプを指定します。
- ④ : 使用するハードウェアが ISA バス版の場合 I/O アドレスを指定します。
- ⑤ : 使用するハードウェアが ISA バス版の場合割り込み番号を指定します。
- ⑥ : 使用するハードウェアが USB 版の場合、接続されたデバイス番号を表示します。
- ⑦ : 使用するハードウェアが USB 版の場合、⑧の指定番号をデバイス番号に設定します。
- ⑧ : 使用するハードウェアが USB 版の場合、デバイス番号を入力します
- ⑨ : 現在のハードウェア状態を表示します。
- ⑩ : 現在使用している『PCAN-BASIC』の DLL 情報を表示します。
- ⑪ : ①～⑤の設定にもとづきドライバをイニシャライズ/開放します。

「HardWare (ハードウェア種別)」、「Baudrate (ボーレート)」、「Msg type」を接続しているハードウェアに合わせて選択します。

「I/O Address」及び「Interrupt」はハードウェア種別がISA版及びDNG版時設定可能になります。選択が終了したら「Initialize」ボタンをクリックしハードウェアの設定を行います。

ハードウェアをリリースする場合は、



「Release」ボタンをクリックしハードウェアを開放します。

ハードウェアの設定を「File」メニューでファイルを選択することで保存可能 (Save Setting File) することが可能です。保存したハードウェア設定を呼び出す (Open Setting File) ことも可能です。「File」メニューのショートカットは、

No	項目	ショートカットキー
1	Open Setting File	Ctrl + O
2	Save Setting File	Ctrl + S

になります。

6. 3. ライトメッセージ動作

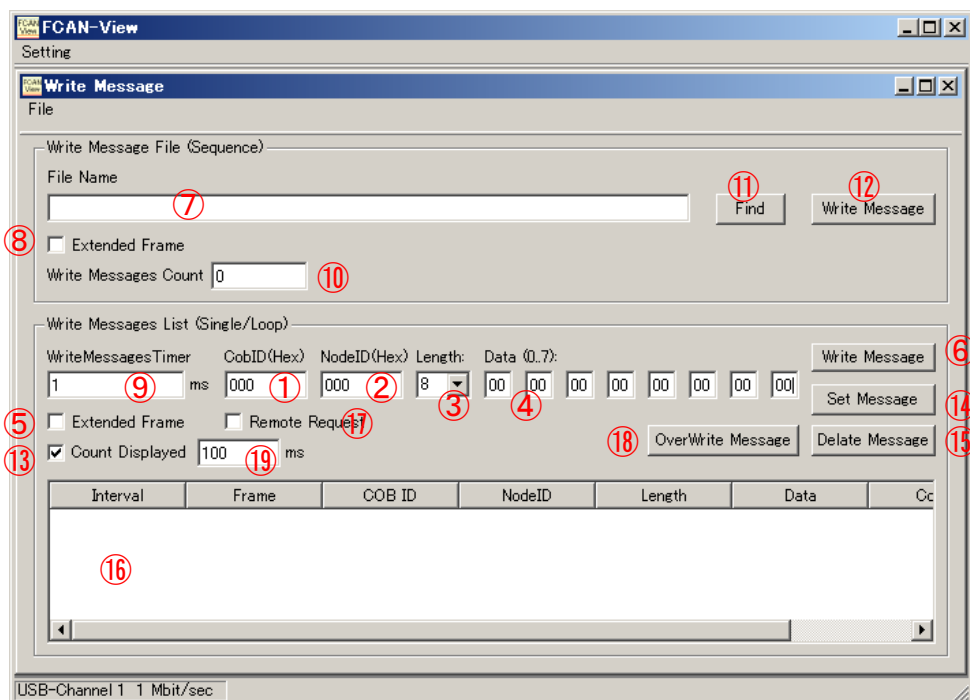
ライトメッセージ動作を行う場合、下記の手順になります。

(1) ハードウェアを設定。

「ハードウェアの設定動作」と同様に設定行います。詳細は「ハードウェアの設定動作 6. 2」を参照して下さい。

(2) ライトメッセージの生成及び送出。

「Setting」メニューから「Write Message」を選択しクリックします。その後ライトメッセージ設定画面が現われます。



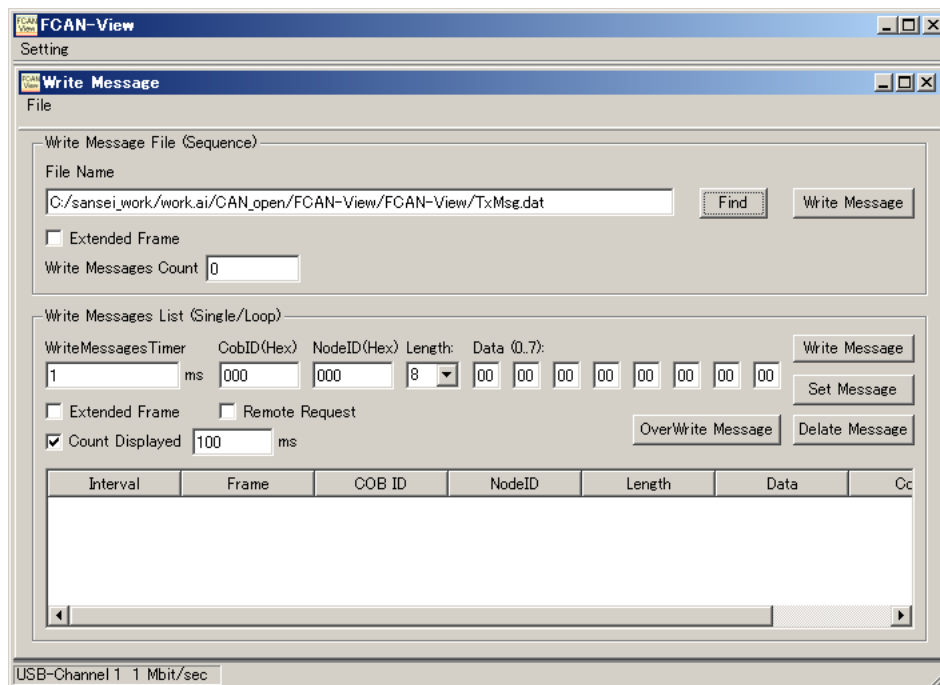
- ① : 送信メッセージ COB-ID を設定します (Hex 値)。
- ② : 送信メッセージ ID を設定します (Hex 値)。
- ③ : 送信メッセージのデータバイト数を設定します。
- ④ : 送信メッセージのデータを設定します。左側から 0, 1, 2, 3, 4, 5, 6, 7 の順に出力されます (Hex 値)。
- ⑤ : 送信メッセージタイプの EXTENDED タイプ設定を選択します。
- ⑥ : ①～⑤に設定されたメッセージを送信します。
- ⑦ : 送信メッセージ用ファイル名を指定します。
- ⑧ : 送信メッセージタイプの EXTENDED タイプ設定を選択します。
- ⑨ : 送信メッセージ間隔を設定します (Dec 値)。
- ⑩ : 送信メッセージ数を表示します (Dec 値)。
- ⑪ : 送信メッセージ用ファイルを検索します。
- ⑫ : 送信メッセージ用ファイルに記載された送信メッセージを送信します。
- ⑬ : 送信メッセージカウント表示間隔を選択します。
- ⑭ : ①～⑤に設定された送信メッセージを⑬に格納します。
- ⑮ : ⑬内で選択された送信メッセージを削除します。
- ⑯ : 送信メッセージを表示します。(送信メッセージリスト)
- ⑰ : 送信メッセージを RemoteRequest に設定します。
- ⑱ : ①～⑤に設定された送信メッセージを⑯の選択されたメッセージに上書きします。
- ⑲ : 送信メッセージカウント表示間隔を設定します。

[Write Message File (Sequence)]

ライトメッセージの送出にはあらかじめ作成しておいたメッセージをファイルに記述し送信することが可能です。

「Write Message File」グループ内「Find」ボタンにて送出するファイルを選択します。

選択後、「Write Message File」グループ内「Write Message」をクリックしメッセージを送出します。



ライトメッセージファイルの構成を下記に示します。

Write Message File 設定ファイルは左側から下記の構成になります。左端に「#」がある場合コメント行として判断されます。

```
#[0]送信モード
#[1]送信間隔(ms) (Dec)
#[2]CobID (Hex)
#[3] ID (Hex)
#[4]送信データ長 (Dec)
#[5]送信データ 0 (Hex)
#[6]送信データ 1 (Hex)
#[7]送信データ 2 (Hex)
#[8]送信データ 3 (Hex)
#[9]送信データ 4 (Hex)
#[10]送信データ 5 (Hex)
#[11]送信データ 6 (Hex)
#[12]送信データ 7 (Hex)
T, 100, 0, 0A, 8, 01, 02, 03, 04, 05, 06, 07, 08
```

[Write Message List (Single/Loop)]

ライトメッセージの生成は、「Write Message List」グループ内「CobID」、「ID」、「Length」、「Data」、「Extended Frame」の各設定を行った後「Set Message」ボタンをクリックすることで可能です。

設定されたライトメッセージはメッセージリストに追加されます。

メッセージリストのメッセージは、メッセージリスト内データをダブルクリックによる選択にて「CobID」、「ID」、「Length」、「Data」、「Extended Frame」に呼び込まれます。

「Over Write Message」ボタンをクリックするとメッセージリストの選択されたメッセージを上書きします。

「Set Message」ボタンをクリックすると新規メッセージが作成されます。

また、選択状態で「Delete Message」をクリックすることにより削除することが可能です。

選択状態で「Del」キー入力でも削除可能です。

作成されたメッセージは「Write Message」をクリックすることで送出します。

送出されるメッセージは「CobID」、「ID」、「Length」、「Data」、「Extended Frame」に表示されている内容になります。

メッセージリスト内の「CobID」、「ID」をダブルクリックによる選択で「Write Message」をクリックする動作と同様な動作を実行します。送出されるメッセージは選択されたメッセージリスト内のメッセージになります。

メッセージリスト内の「Interval」にレ点を入れて送信した場合「Write Message Timer」に指定されたタイマ間隔にて連続で同じメッセージを出力することが可能です。

送出されたメッセージ数はメッセージリスト内の「Count」に表示されます。

メッセージリスト内の「Count」の表示間隔は「Count Displayed」の設定で行います。

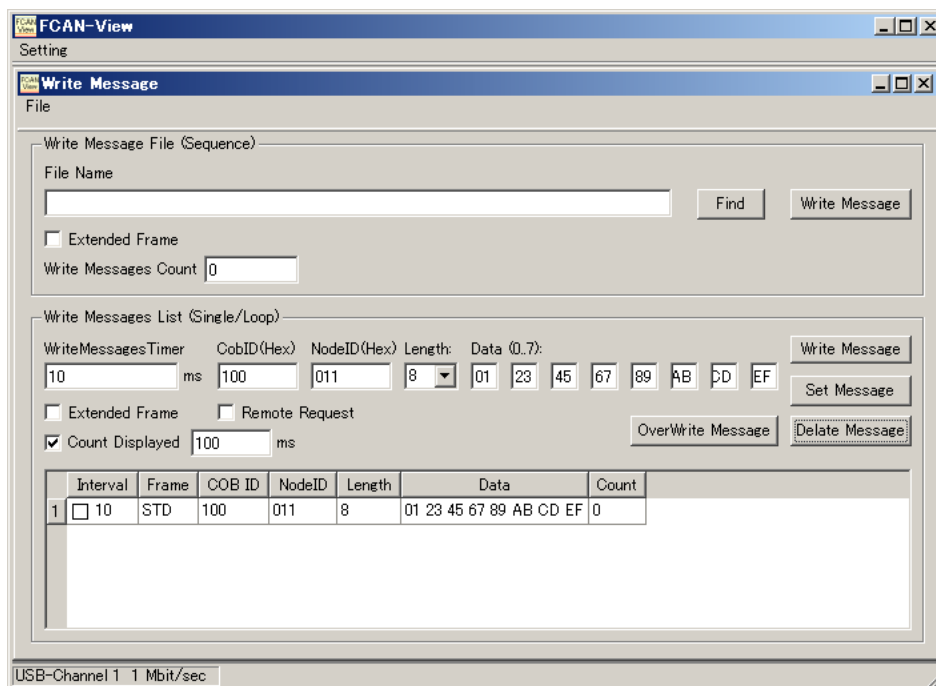
「Count Displayed」のチェックを外すと「Count」表示は更新されません。

「Count Displayed」のチェックした状態で指定したタイマ間隔で「Count」表示は更新されます。

タイマ間隔にゼロを指定した場合、「Count」表示は「Write Message」を出力したタイミングで更新されます。

Interval	Frame	COB ID	NodeID	Length	Data	Count
1	<input type="checkbox"/> 0	STD	100	011	8	01 23 45 67 89 AB CD EF
						0

メッセージリスト内の「Interval」にレ点が入っていない場合は1回のみメッセージを出力します。



「File」メニューのショートカットは、

No	項目	ショートカットキー
1	Open Write Message	Ctrl + O
2	Save Write Message	Ctrl + S

になります。メッセージリストの読み込み/保存に使用可能です。

Write Message Single/Loop 設定ファイルは左側から下記の構成になります。左端に「#」がある場合コメント行として判断されます。

```
#[0] 送信間隔 (ms) (Dec)
#[1] 送信タイプ (STD/EXT)
#[2] CobID (Hex)
#[3] ID (Hex)
#[4] 送信データ長 (Dec)
#[5] 送信データ 0 (Hex)
#[6] 送信データ 1 (Hex)
#[7] 送信データ 2 (Hex)
#[8] 送信データ 3 (Hex)
#[9] 送信データ 4 (Hex)
#[10] 送信データ 5 (Hex)
#[11] 送信データ 6 (Hex)
#[12] 送信データ 7 (Hex)
#[13] 送信数
1, STD, 000, 001, 8, 11 12 13 14 15 16 17 18, 0
```

ライトメッセージの Single/Loop の判別は送信間隔 (ms) の設定内容により判断されます。送信間隔に「0」設定の場合 Single ライトメッセージ、それ以外の場合 Loop ライトメッセージになります。

6. 4. リードメッセージ動作

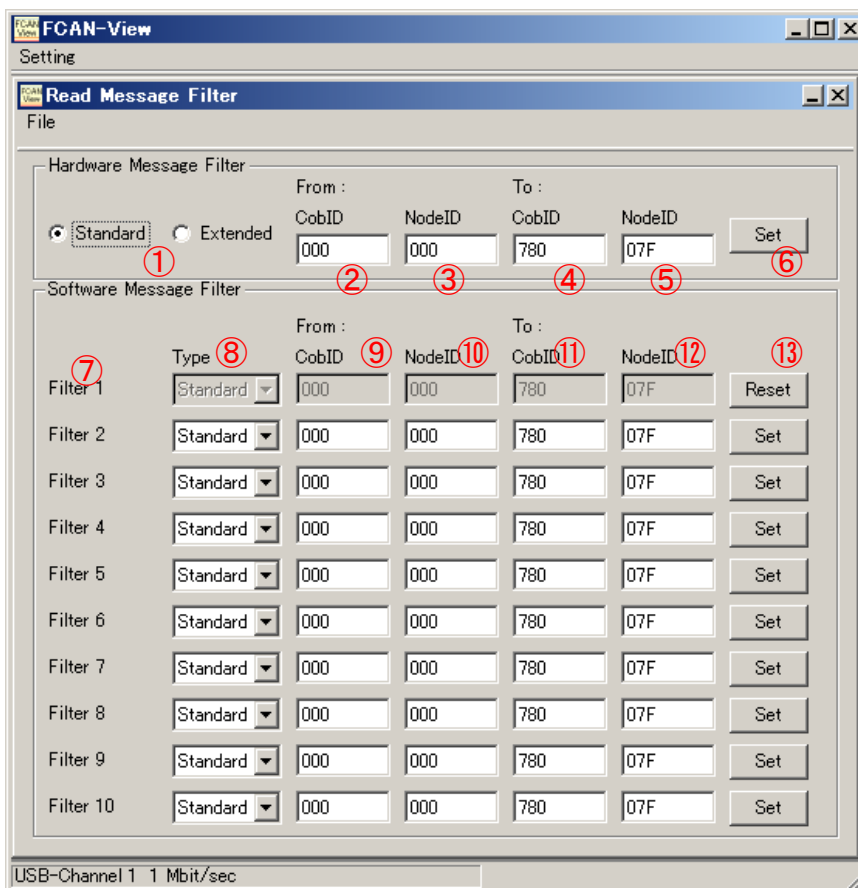
リードメッセージ動作を行う場合、下記の手順になります。

(1) ハードウェアを設定。

「ハードウェアの設定動作」と同様に設定行います。詳細は「ハードウェアの設定動作 6. 2」を参照して下さい。

(2) リードメッセージのフィルタを設定。

「Setting」メニューから「Read Message Filter」を選択しクリックします。その後フィルタ設定画面が現われます。



① : ハードウェアフィルタのメッセージタイプを指定します。

② : ハードウェアフィルタの開始 COB-ID 指定します。

③ : ハードウェアフィルタの開始 ID 指定します。

④ : ハードウェアフィルタの終了 COB-ID 指定します。

⑤ : ハードウェアフィルタの終了 ID 指定します。

⑥ : ハードウェアフィルタを設定/解除します。

⑦ : ソフトウェアフィルタの設定番号を指定します。

⑧ : ソフトウェアフィルタのメッセージタイプを指定します。

⑨ : ソフトウェアフィルタの開始 COB-ID 指定します。

⑩ : ソフトウェアフィルタの開始 ID 指定します。

⑪ : ソフトウェアフィルタの終了 COB-ID 指定します。

⑫ : ソフトウェアフィルタの終了 ID 指定します。

⑬ : ソフトウェアフィルタを設定/解除します。

* Filter2~Filter10 までも同様に設定します。

「HardWare Message Filter」グループではハードウェアにて受信可能とするメッセージタイプ及び「CobID」,「ID」を設定し「Set」ボタンをクリックします。解除する場合は「Reset」ボタンが選択可能になりますので「Reset」ボタンをクリックして下さい。受信されるリードメッセージは

開始CpbID+ID <= リードメッセージCpbID+ID <= 終了CpbID+ID
の範囲になります。

「SoftWare Message Filter」グループでは「HardWare Message Filter」と同様にソフトウェアにてフィルタ処理が可能です。設定方法は「HardWare Message Filter」グループと同じです。

受信されるリードメッセージは

開始CpbID+ID <= リードメッセージCpbID+ID <= 終了CpbID+ID
の範囲になります。

受信されるメッセージとハードウェアフィルタ/ソフトウェアフィルタとの関係は、
ハードウェアフィルタ and (ソフトウェアフィルタ1 or ソフトウェアフィルタ2 or …… or ソフトウェアフィルタ10)
の条件になります。

全メッセージを受信する場合、ハードウェアフィルタをリセットし、ソフトウェアフィルタに全範囲を設定設定して下さい。

「File」メニューのショートカットは、

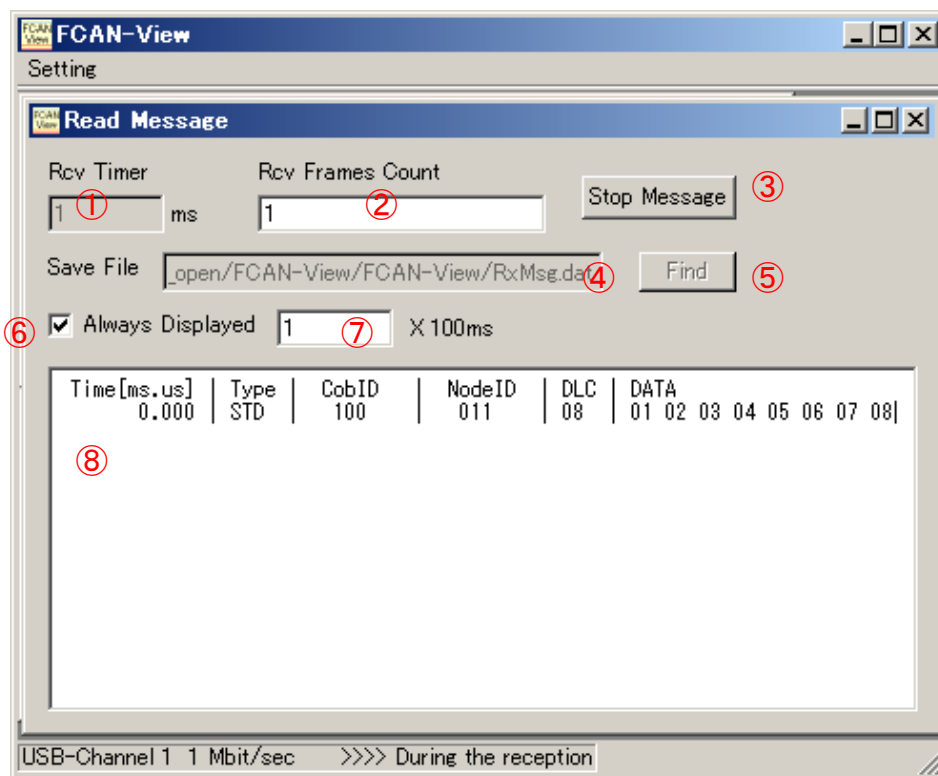
No	項目	ショートカットキー
1	Open File	Ctrl + O
2	Save File	Ctrl + S

になります。

設定したフィルタリストの読み込み/保存に使用可能です。

(3) リードメッセージの受信。

「Setting」メニューから「Read Message」を選択しクリックします。その後リードメッセージ設定画面が現われます。



- ① : メッセージを受信するタイマ設定間隔を設定します (Dec 値)。
- ② : 受信メッセージ数を設定します (Dec 値)。
- ③ : 受信動作を開始/停止します。
- ④ : 受信したメッセージを保存するファイル名を指定します。
- ⑤ : 受信したメッセージを保存するファイルを検索します。
- ⑥ : 受信メッセージの表示/非表示を切り替えます。
- ⑦ : 受信メッセージの表示間隔を設定します (Dec 値)。
- ⑧ : 受信メッセージを表示します。

「Rcv Timer」にて受信処理間隔時間を設定します。設定した間隔時間内に受信したデータを一括して指定したタイミングで処理します。

受信処理負荷を軽減したい場合は、100msなどある程度大きな時間を設定してください。ただし、設定した時間間隔毎に処理メモリに取り込むため、このメモリを利用してグラフ表示を行う機能なので表示の遅延が起こる場合があります。

「Rcv Frame Count」にて受信するメッセージ数を設定します。

「Find」ボタンによりリードメッセージを保存するファイルを選択します。

設定が終了したら「Start」ボタンをクリックし受信動作を開始します。受信されたメッセージ数は「Rcv Frame Count」にて表示されます。

受信開始後、強制的に停止する場合「Stop」ボタンをクリックします。「Start」ボタンクリック後「Stop」ボタンはクリック可能になります。

「Always Displayed」にチェックをいれることで常に受信メッセージを設定した表示間隔にて表示します。チェックを外すと受信メッセージは表示しません。

受信開始後、「Always Displayed」の設定は禁止されます。表示を行う場合かならず受信開始前に設定して下さい。

受信メッセージ格納ファイルは左側から下記の構成になります。

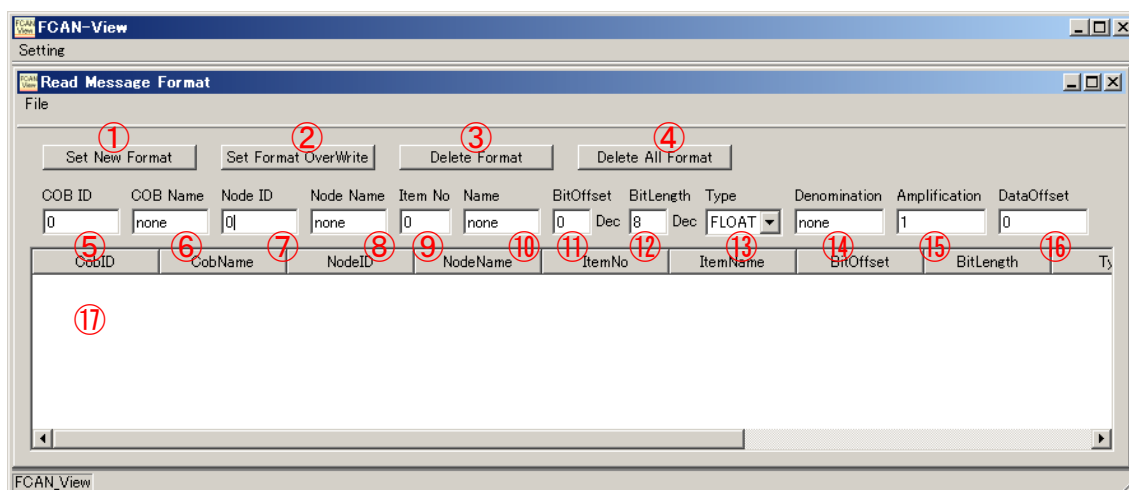
```
#[0] 受信モード
#[1] 時間 ms. us (Dec)
#[2] CobID (Hex)
#[3] ID (Hex)
#[4] 送信データ長 (Dec)
#[5] 送信データ 0 (Hex)
#[6] 送信データ 1 (Hex)
#[7] 送信データ 2 (Hex)
#[8] 送信データ 3 (Hex)
#[9] 送信データ 4 (Hex)
#[10] 送信データ 5 (Hex)
#[11] 送信データ 6 (Hex)
#[12] 送信データ 7 (Hex)
R, 0, 000, 00000500, 00000029, 08, 00, 00, 00, 00, 00, 00, 00, 00
```

6. 5. グラフ表示動作

グラフ表示動作を行う場合、下記の手順になります。

(1) フォーマットデータを設定。

「Setting」メニューから「Read Message Format」を選択しクリックします。その後フォーマット設定画面が現われます。



- ① : 新規フォーマットを追加します。
- ② : 指定されたフォーマットに上書きします。
- ③ : 指定されたフォーマットを削除します。
- ④ : 現在設定されているフォーマットを全て削除します。
- ⑤ : COB-ID を設定します (Hex 値)。
- ⑥ : COB-ID 名称を設定します。
- ⑦ : ノード ID を設定します (Hex 値)。
- ⑧ : ノード ID 名称を設定します。
- ⑨ : アイテム No を設定します (Dec 値)。
- ⑩ : アイテム名称を設定します。
- ⑪ : ビット位置オフセットを指定します (Dec 値)。
- ⑫ : ビット長を指定します (Dec 値)。
- ⑬ : アイテム値の型を指定します。
- ⑭ : アイテム値の単位を設定します。
- ⑮ : アイテム値の倍率を設定します (Float 値)。
- ⑯ : アイテム値のオフセット値を設定します。
- ⑰ : 現在設定されているフォーマット状況を表示します。

内部データ値は下記の値になります。

内部データ値 = (受信アイテム値 × アイテム値の倍率) + アイテム値のオフセット値

「CobID」から「DataOffset」までの項目を設定します。

設定後、「Set New Format」ボタンをクリックしフォーマットデータを設定します。設定されると下の「Format View List」に表示されます。

「Format View List」に表示された行をダブルクリックすることにより上の編集可能エリアにコピーされます。

「Format View List」を選択後、「Set Format OverWrite」ボタンをクリックしフォーマットデータを上書きすることが可能です。

「Format View List」を選択後「Del」キー入力により選択されたフォーマットを削除することが可能です。また「Delete Format」ボタンによっても削除することが可能です。「Delete All Format」ボタンは登録されている全てのフォーマットを削除します。

「File」メニュー内「Open RxDataFormat」を選択しフォーマットデータファイルを読み込むことが可能です。「File」メニュー内「Save RxDataFormat」を選択することによりフォーマットバッファ内のすべてのフォーマットデータをファイルに出力します。

「File」メニューのショートカットは、

No	項目	ショートカットキー
1	Open RxDataFormat	Ctrl + O
2	Save RxDataFormat	Ctrl + S

になります。

設定したフォーマットの読み込み/保存に使用可能です。

(2) フォーマット設定ファイルの構成。

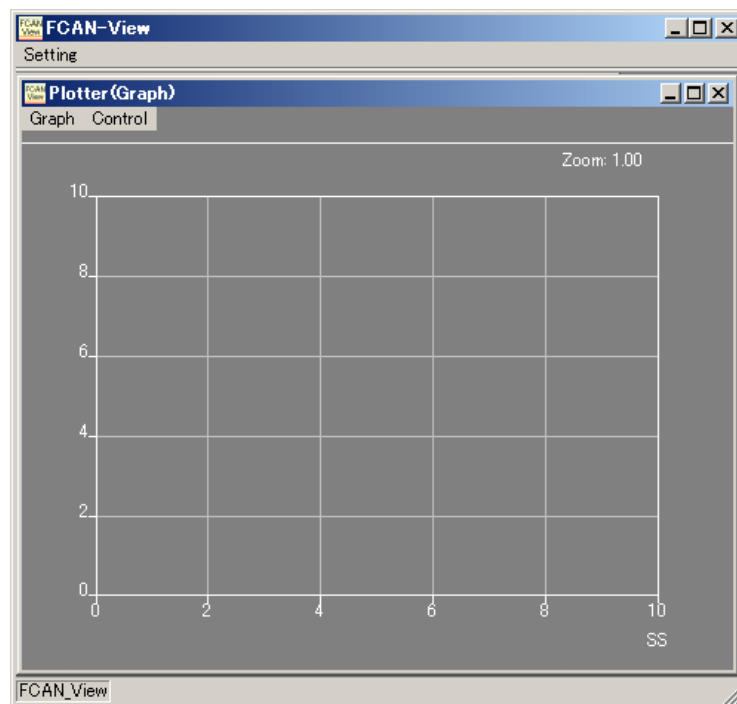
Read Message Format 設定ファイルは左側から下記の構成になります。左端に「#」がある場合コメント行として判断されます。

#[0]CobID (Hex)	→	COB-ID を設定します (Hex 値)
#[1]CobName	→	COB-ID 名称を設定します
#[2]NodeID (Hex)	→	ノード ID を設定します (Hex 値)
#[3]NodeName	→	ノード ID 名称を設定します
#[4]ItemNo (Dec)	→	アイテム No を設定します (Dec 値)
#[5]ItemName	→	アイテム名称を設定します
#[6]BitOffset (Dec)	→	ビットオフセットを指定します (Dec 値)
#[7]BitLength (Dec)	→	ビット長を指定します (Dec 値)
#[8]Type (0:Float, 1:UInt, 2:Int)	→	アイテム値の型を指定します
#[9]Denomination	→	アイテム値の単位を設定します
#[10]Amplification (Dec)	→	アイテム値の倍率を設定します (FLOAT)
#[11]DataOffset (Dec)	→	アイテム値のオフセット値を設定します

0, COBID0, 00, ID00, 0, ITEM0000, 00, 08, 0, 0000, 1, 0

② グラフ表示を設定。

「Setting」メニューから「NewGraph」を選択しクリックします。その後プロッタ画面が現れますので「Graph」メニューから「Setting」を選択することによりグラフ設定画面が現れます。



「Graph」メニューのショートカットは、

No	項目	ショートカットキー
1	Setting	Ctrl + S
2	Print	Ctrl + P

になります。

Printにより画面を印刷することができます。

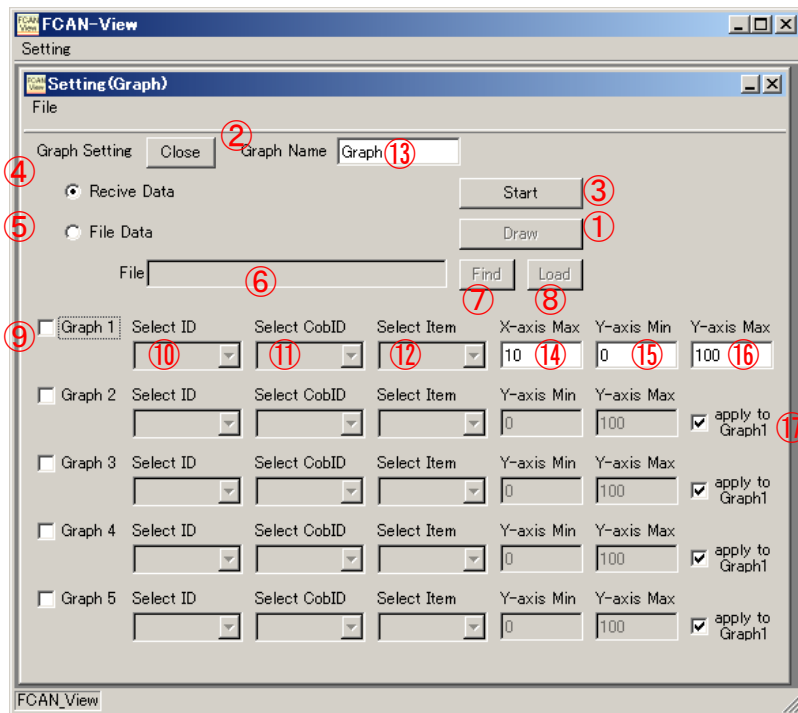
「Control」メニューのショートカットは、

No	項目	ショートカットキー
1	StartStop	Ctrl + B
2	ViewEndPoint	Ctrl + E

になります。

「StartStop」はGraph-SettingのRecvDataの「Start」又は「Stop」ボタンの押下動作と同意です。

「ViewEndPoint」はグラフのX軸最終点に表示を移動します。



- ① : 設定値に従いグラフを表示します。
- ② : グラフ設定画面表示を閉じます。
- ③ : 受信と同時に表示を選択時、受信メッセージ動作を開始します。
- ④ : 受信メッセージ動作と同時にグラフ表示を行います。
- ⑤ : 受信したデータファイルのデータをグラフ表示を行います。
- ⑥ : 受信メッセージファイル名を指定します。
- ⑦ : 受信メッセージファイルを検索します。
- ⑧ : 受信目メッセージファイルをデータバッファに読み込みます。
- ⑨ : グラフ 1 項目の表示をします。
- ⑩ : グラフ 1 の ID を選択します。
- ⑪ : グラフ 1 の COB-ID を選択します。
- ⑫ : グラフ 1 の Item を選択します。
- ⑬ : グラフ名を設定します。
- ⑭ : グラフの X 軸最大値を入力します (Dec 値)。
- ⑮ : グラフの Y 軸最小値を入力します (Dec 値)。
- ⑯ : グラフの Y 軸最大値を入力します (Dec 値)。
- ⑰ : グラフ 1 の Y 軸最小値及び Y 軸最大値をグラフ 2~5 で共用します。
- * ⑨~⑫、⑮~⑰は各グラフ 2~5 にも同様に選択されます。

「File」メニューのショートカットは、

No	項目	ショートカットキー
1	Open Graph Setting	Ctrl + O
2	Save Graph Setting	Ctrl + S

になります。

グラフ表示設定の読み込み/保存に使用可能です。

「Setting」メニューから「NewGraph」を選択しクリックします。その後プロッタ画面が現れますので「Graph」メニューから「Setting」を選択することによりグラフ設定画面が現れます。

「Recive Data」又は「File Data」を選択します。

「File Data」を選択の場合は、「Find」ボタンクリックによりリードメッセージファイルを選択し、「Load」ボタンクリックによりファイルを内部データバッファに読み込みます。

描画するグラフは「Graph1」～「Graph5」までのチェックボタンにより選択します。詳細なアイテムは「Select ID」, 「Select CobID」, 「Select Item」の順番で選択を行います。途中、次ぎの選択項目が表示されない場合は、再度前項目の選択をやり直し下さい。


「X-axis Max」にてX軸最大値を入力します。これは「Recive Data」選択時のみ有効になります。

「Y-axis Min」にてY軸最小値を、「Y-axis Min」にてY軸最大値を入力します。「apply to Graph1」をチェックした場合、Graph1のY-axis Min, Y-axis MaxがGraph1～Graph5まで共通して使用されます。以上の設定が終了しましたら「Draw」ボタンをクリックしグラフを描画します。

「Recive Data」を選択の場合は、「File Data」と同様に表示するグラフの設定を行った後、「Start」ボタンをクリックし受信メッセージがデータバッファにライトされるのを待ちます。

「Recive Data」を選択の場合は、全てのグラフ設定を終了後「start」ボタンをクリックした後、リードメッセージの「Start」ボタンをクリックして下さい。

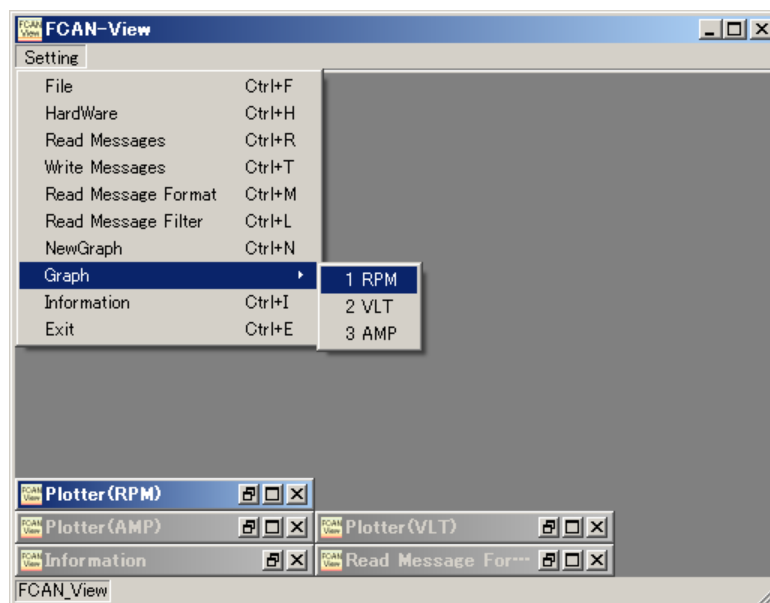
停止する場合は、リードメッセージの「Stop」ボタンクリック後、グラフの「Stop」ボタンをクリックして下さい。

「Graphウインドウ」及び「Plotterウインドウ」の右上にある  マークにてウインドウを削除しないで下さい。

どちらかを削除した場合、「Graphウインドウ」及び「Plotterウインドウ」の両方が削除されます。設定画面削除は「Close」ボタンをクリックして行って下さい。

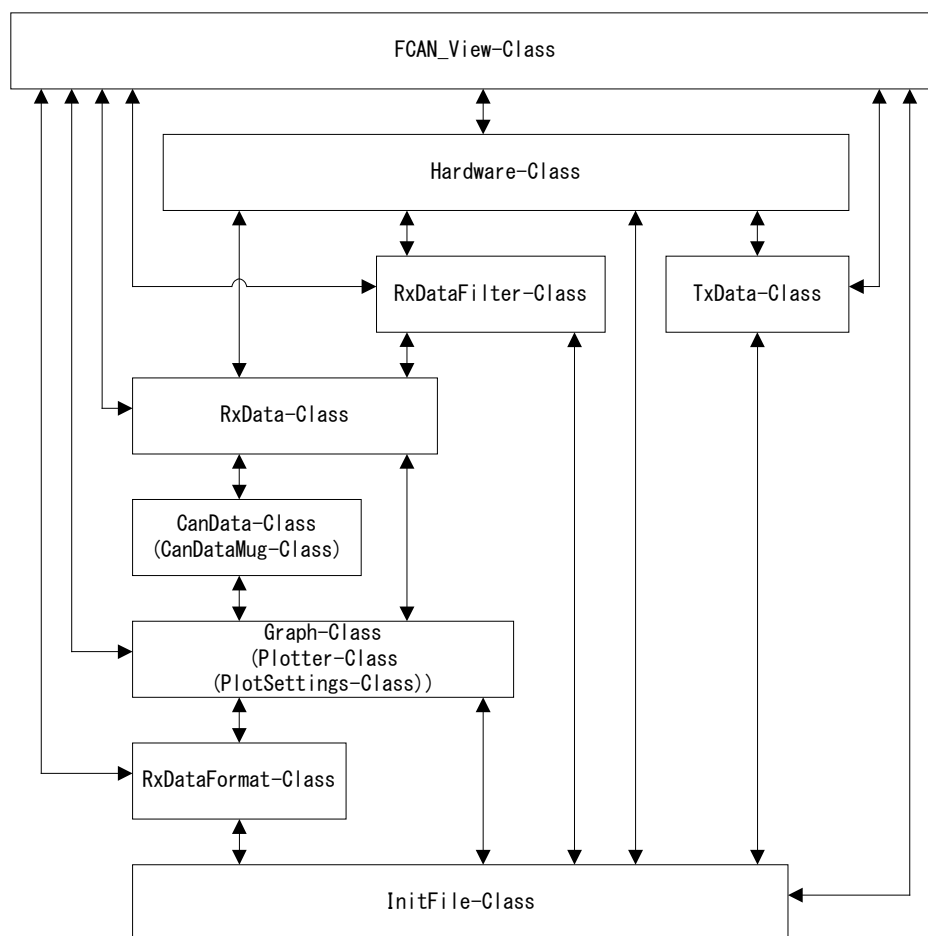
各グラフには名称がつけられます。「Graph Name」にて設定して下さい。設定後設定画面を「Close」ボタンにて削除することにより名称はグラフに反映されます。

また、「Setteing」メニューから表示しているグラフを探す場合、「Setteing」メニュー内「Graph」メニューにて選択が可能です。「Graph」メニューに表示される名称はグラフ設定にて設定した名称になります。



フォーマットデータを変更した場合、「Graph1」～「Graph5」までのチェックボタンを有効/無効と変化させフォーマット情報を更新してください。

7. 構造(クラス)構成

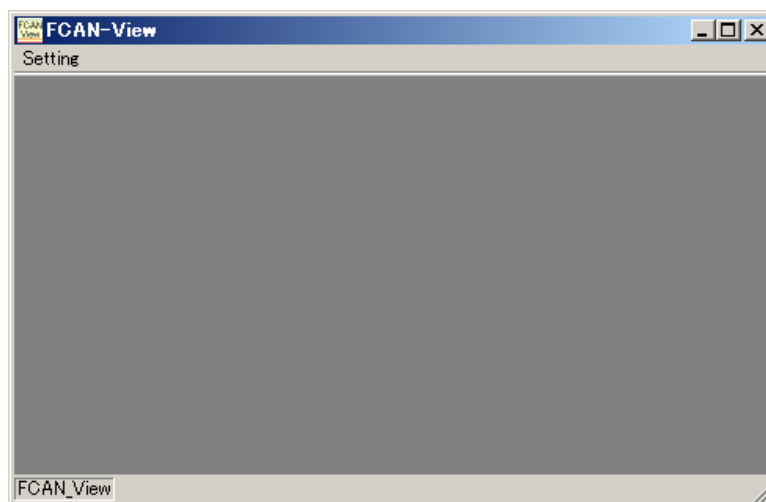


* Information-Class は各クラスから呼出され表示するのみです。

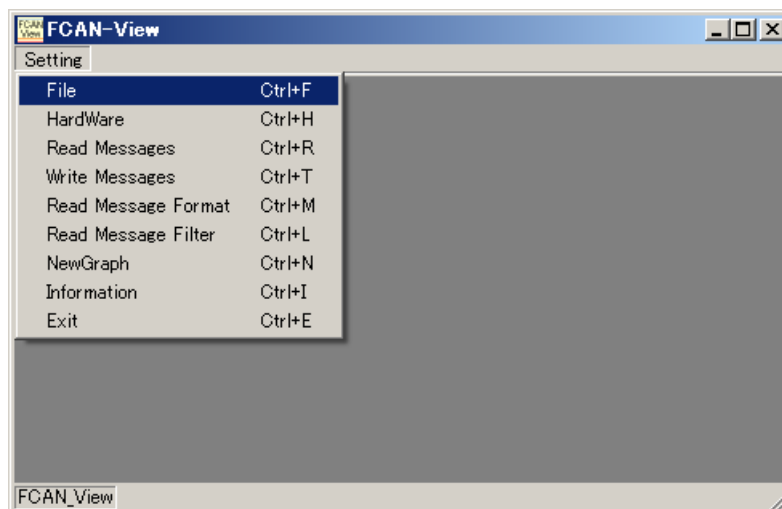
No	Class	用途
1	FCAN_View	メインクラス
2	InitFile	各設定の初期化/保存を行います。
3	Hardware	PCAN-USB 等のハードウェアを制御します。
4	TxData	ライトメッセージ制御を行います。
5	RxData	リードメッセージ制御を行います。
6	RxDataFilter	リードメッセージのフィルタリング処理の設定を行います。
7	RxDataFormat	リードメッセージのデータフォーマットの設定を行います。
8	Graph	リードメッセージを指定された設定に従いデータを抽出します。
9	Information	各クラスからの情報を表示します。
10	Plotter	グラフクラスにて抽出されたデータをグラフに描画します。
11	Plotsettings	グラフ描画に必要な設定を行います。
12	CanData	メッセージバッファの制御を行います。
13	CanDataMng	メッセージバッファ本体です。
14	CanFrame	フォーマットデータの制御を行います。
15	CanFrameNode	ID フォーマットバッファ本体及び制御を行います。
16	CanFrameFunction	Cob-ID フォーマットバッファ本体及び制御を行います。
17	CanFrameFormat	Item フォーマットバッファ本体及び制御を行います。

8. FCAN-Viewi-Qwidget

8.1. Qwidget(ウィンドウ)



8.2. Qwidget(ウィンドウ)メニュー



8.3. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	SysConfigSet	SysConfig-Class ポインタコピー
3	InitFileSet	InitFile-Class ポインタコピー
4	HardwareSet	HardWare-Class ポインタコピー
5	RxDataSet	RxData-Class ポインタコピー
6	TxDataSet	TxData-Class ポインタコピー
7	RxDataFormatSet	RxDataFormat-Class ポインタコピー
8	RxDataFilterSet	RxDataFilter-Class ポインタコピー
9	GraphSet	Graph-Class ポインタコピー
10	PlotterSet	Plotter-Class ポインタコピー
11	PlotSettingsSet	PlotSettings-Class ポインタコピー
12	InformationSet	Information-Class ポインタコピー
13	CanDataSet	CanData-Class ポインタコピー
14	CanDataMngSet	CanDataMng-Class ポインタコピー
15	CanFrameSet	CanFrame-Class ポインタコピー
16	Start	FCAN_View 初期化
17	HardWareSetting	ハードウェア設定内容保存
18	statusBar_message	ステータスバーメッセージ表示
19	statusBar_clear	ステータスバーメッセージクリア
20	HardwareView	HardWare-Qwidget 表示
21	RxDataFilterView	RxDataFilter-Qwidget 表示
22	RxDataFormatView	RxDataFormat-Qwidget 表示
23	RxDataView	RxData-Qwidget 表示
24	TxDataView	TxData-Qwidget 表示
25	GraphView	Graph-Qwidget 表示
26	GraphViewShow	Graph-Qwidget 再表示
27	GraphViewClose	Graph-Qwidget クローズ
28	GraphInit	Graph-Qwidget 初期化
23	PlotterView	Plotter-Qwidget 表示

private 関数

No	関数名	用途
1	createMenus	メニューバーの生成
2	ViewAdd	各画面生成時のポインタ更新

8.3.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.2. void SysConfigSet(SysConfig * sysconfig)

項	変数	説明
引数	SysConfig * sysconfig	SysConfig クラスポインタ
戻値	なし	—

SysConfigSet 関数は SysConfig-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.3. void InitFileSet(InitFile * initfile)

項	変数	説明
引数	InitFile * initfile	InitFile クラスポインタ
戻値	なし	—

InitFileSet 関数は InitFile-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.4. void HardwareSet(Hardware * hardware)

項	変数	説明
引数	Hardware * hardware	Hardware クラスポインタ
戻値	なし	—

HardwareSet 関数は Hardware-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.5. void RxDataSet(RxData * rxdata)

項	変数	説明
引数	RxData * rxdata	RxData クラスポインタ
戻値	なし	—

RxDataSet 関数は RxData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.6. void TxDataSet(TxData * txdata)

項	変数	説明
引数	TxData * txdata	TxData クラスポインタ
戻値	なし	—

TxDataSet 関数は TxData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.7. void RxDataFormatSet(RxDataFormat * rxdataformat)

項	変数	説明
引数	RxDataFormat * rxdataformat	RxDataFormat クラスポインタ
戻値	なし	—

RxDataFormatSet 関数は RxDataFormat-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.8. void RxDataFilterSet(RxDataFilter * rxdatafilter)

項	変数	説明
引数	RxDataFilter * rxdatafilter	RxDataFilter クラスポインタ
戻値	なし	—

RxDataFilterSet 関数は RxDataFilter-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.9. void GraphSet(Graph * graph)

項	変数	説明
引数	Graph * graph	Graph クラスポインタ
戻値	なし	—

GraphSet 関数は Graph-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.10. void PlotterSet(Plotter * plotter)

項	変数	説明
引数	Plotter * plotter	Plotter クラスポインタ
戻値	なし	—

PlotterSet 関数は Plotter-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.11. void PlotSettingsSet(PlotSettings * plotsettings)

項	変数	説明
引数	PlotSettings * plotsettings	PlotSettings クラスポインタ
戻値	なし	—

PlotSettingsSet 関数は PlotSettings-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.12. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数は Information-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.13. void CanDataSet(CanData * candata)

項	変数	説明
引数	CanData * candata	CanData クラスポインタ
戻値	なし	—

CanDataSet 関数は CanData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.14. void CanDataMngSet(CanDataMng * candatamng)

項	変数	説明
引数	CanDataMng * candatamng	CanDataMng クラスポインタ
戻値	なし	—

CanDataMngSet 関数は CanDataMng-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.15. void CanFrameSet(CanFrame * canframe)

項	変数	説明
引数	CanFrame * canframe	CanFrame クラスポインタ
戻値	なし	—

CanFrame 関数は CanFrame-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

8.3.16. void Start()

項	変数	説明
引数	なし	—
戻値	なし	—

Start 関数は FCAN_View 初期化を行います。Workspace の作成及び Information-Widget の表示を行います。

8.3.17. void HardwareSetting(QString qstr)

項	変数	説明
引数	QString qstr	ハードウェア設定内容
戻値	なし	—

HardwareSetting 関数はハードウェア設定内容を保存しステータスバーに表示する場合に利用します。

8.3.18. void statusBar_message(QString qstr)

項	変数	説明
引数	QString qstr	ステータスバー表示内容
戻値	なし	—

statusBar_message 関数は指定された内容をステータスバーに表示します。

8.3.19. void statusBar_clear()

項	変数	説明
引数	なし	—
戻値	なし	—

statusBar_clear 関数はステータスバーに表示された内容をクリアします。

8.3.20. void HardwareView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

HardwareView 関数は Hardware-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.21. void RxDataFilterView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

RxDataFilterView 関数は RxDataFilter-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.22. void RxDataFormatView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

RxDataFormatView 関数は RxDataFormat-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.23. void RxDataView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

RxDataView 関数は RxData-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.24. void TxDataView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

TxDataView 関数は TxData-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.25. void GraphView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

GraphView 関数は Graph-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.26. void GraphViewShow(int no)

項	変数	説明
引数	int no	Graph-Widget の画面番号
戻値	なし	—

GraphViewShow 関数は該当する Graph-Widget の再表示を行います。

8.3.27. void GraphViewClose(int no)

項	変数	説明
引数	int no	Graph-Widget の画面番号
戻値	なし	—

GraphViewClose 関数は該当する Graph-Widget を閉じます。

8.3.28. QString GraphInit(int no, int mode, QString line)

項	変数	説明
引数1	int no	Graph-Widget の画面番号
引数2	int mode	画面表示モード(0:通常 1:表示後最小化)
引数3	QString line	Graph 初期化内容
戻値	QString	Graph 初期化結果又は Graph 設定内容

GraphInit 関数は該当する Graph-Widget の初期化又は設定内容の読み出しを行います。

「Initialize File Load」又は「Initialize File Save」動作時に呼び出されます。

8.3.29. void PlotterView(int mode)

項	変数	説明
引数	int mode	画面表示モード(0:通常 1:表示後最小化)
戻値	なし	—

PlotterView 関数はPlotter-Widget の表示を行います。Mode の設定により画面の最小化を行います。

8.3.30. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューバーの生成を行います。

8.3.31. void ViewAdd()

項	変数	説明
引数	なし	—
戻値	なし	—

ViewAdd 関数は各画面表示時の各クラスポインタの更新を行います。

8.4. スロット宣言

No	関数名	用途
1	menu_initfile	メニューバーより「File」が選択された場合に呼出される関数
2	menu_hardware	メニューバーより「HardWare」が選択された場合に呼出される関数
3	menu_rxdata	メニューバーより「Read Message」が選択された場合に呼出される関数
4	menu_txdata	メニューバーより「Write Message」が選択された場合に呼出される関数
5	menu_rxdataformat	メニューバーより「Read Message Format」が選択された場合に呼出される関数
6	menu_rxdatafilter	メニューバーより「Read Message Filter」が選択された場合に呼出される関数
7	menu_newgraph	メニューバーより「NewGraph」が選択された場合に呼出される関数
8	menu_graph0	メニューバーより「Graph」が選択された場合に呼出される関数 (No-0)
9	menu_graph1	メニューバーより「Graph」が選択された場合に呼出される関数 (No-1)
10	menu_graph2	メニューバーより「Graph」が選択された場合に呼出される関数 (No-2)
11	menu_graph3	メニューバーより「Graph」が選択された場合に呼出される関数 (No-3)
12	menu_graph4	メニューバーより「Graph」が選択された場合に呼出される関数 (No-4)
13	menu_graph5	メニューバーより「Graph」が選択された場合に呼出される関数 (No-5)
14	menu_graph6	メニューバーより「Graph」が選択された場合に呼出される関数 (No-6)
15	menu_graph7	メニューバーより「Graph」が選択された場合に呼出される関数 (No-7)
16	menu_graph8	メニューバーより「Graph」が選択された場合に呼出される関数 (No-8)
17	menu_graph9	メニューバーより「Graph」が選択された場合に呼出される関数 (No-9)
18	menu_information	メニューバーより「Information」が選択された場合に呼出される関数
19	menu_close	メニューバーより「Exit」が選択された場合に呼出される関数
20	menu_triggered	メニューが選択されたトリガ関数
21	closeEvent	FCAN_View の終了処理関数

8.4.1. void menu_initfile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「File」が選択された場合、FCAN_View の workspace 内に InitFile-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.2. void menu_hardware()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「HarWare」が選択された場合、FCAN_View の workspace 内に HarWare-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.3. void menu_rxdata()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Read Message」が選択された場合、FCAN_View の workspace 内に RxData-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.4. void menu_txdata()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Write Message」が選択された場合、FCAN_View の workspace 内に TxData-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.5. void menu_rxdataformat()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Read Message Format」が選択された場合、FCAN_View の workspace 内に RxDataFormat-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.6. void menu_rxdatafilter()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Read Message Filter」が選択された場合、FCAN_View の workspace 内に RxDataFilter-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.7. void menu_newgraph()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「NewGraph」が選択された場合、FCAN_View の workspace 内に Graph-Widget の表示を行います。表示画面フラグの更新を実施します。

8.4.8. void menu_graph0()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph0-Widget の再表示を行います。

8.4.9. void menu_graph1()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph1-Widget の再表示を行います。

8.4.10. void menu_graph2()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph2-Widget の再表示を行います。

8.4.11. void menu_graph3()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph3-Widget の再表示を行います。

8.4.12. void menu_graph4()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph4-Widget の再表示を行います。

8.4.13. void menu_graph5()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph5-Widget の再表示を行います。

8.4.14. void menu_graph6()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph6-Widget の再表示を行います。

8.4.15. void menu_graph7()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph7-Widget の再表示を行います。

8. 4. 16. void menu_graph8()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph8-Widget の再表示を行います。

8. 4. 17. void menu_graph9()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Graph」が選択された場合、Graph9-Widget の再表示を行います。

8. 4. 18. void menu_information()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Information」が選択された場合、FCAN_View の workspace 内に Information-Widget の表示を行います。表示画面フラグの更新を実施します。

8. 4. 19. void menu_close()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューバーより「Exit」が選択された場合、FCAN_View の workspace 内の全てのウインドウを閉じます。表示画面フラグの更新を実施します。

8. 4. 20. void menu_triggered()

項	変数	説明
引数	なし	—
戻値	なし	—

メニューが選択されたトリガにて、メニュー内の内容を再構築します。

8. 4. 21. void closeEvent(QCloseEvent * pEvent)

項	変数	説明
引数	QCloseEvent * pEvent	FCAN_View-Widget のクローズイベント
戻値	なし	—

FCAN_View-Widget のクローズイベントが発生した場合、FCAN_View の workspace 内の全てのウインドウを閉じます。表示画面フラグの更新を実施します。

8.5. データメンバー宣言

public 変数

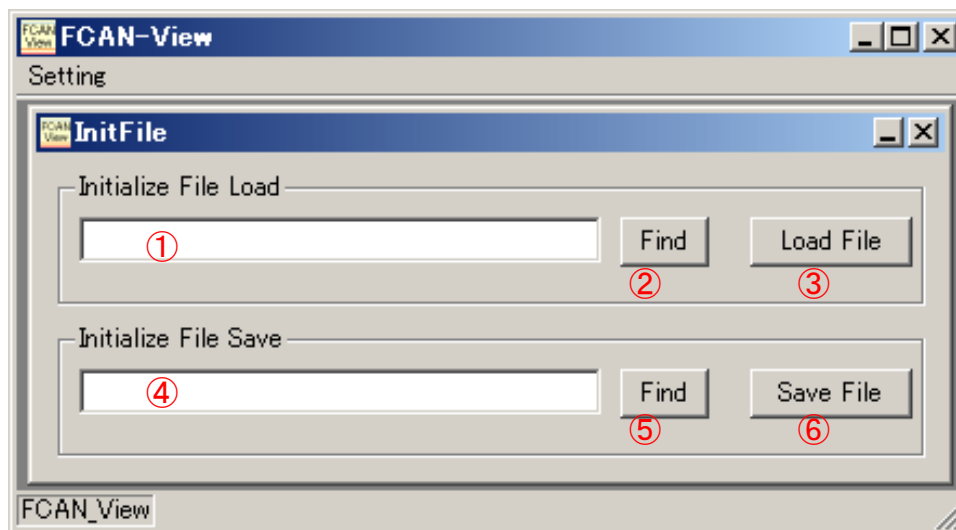
No	変数名	型	用途
1	ViewWindow	bool [10]	各画面表示状況用フラグ
2	ViewGraph	bool [10]	Graph 画面表示状況用フラグ
3	ViewGraphCount	int	Graph 画面表示数
4	GraphNumber	int	Graph 画面番号

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pSysConfig	SysConfig	SysConfig-Class ポインタ
3	* pInitFile	InitFile	InitFile-Class ポインタ
4	* pHardWare	HardWare	HardWare-Class ポインタ
5	* pRxData	RxData	RxData-Class ポインタ
6	* pTxData	TxData	TxData-Class ポインタ
7	* pRxDataFormat	RxDataFormat	RxDataFormat-Class ポインタ
8	* pRxDataFilter	RxDataFilter	RxDataFilter-Class ポインタ
9	* pGraph	Graph	Graph-Class ポインタ
10	* pPlotter	Plotter	Plotter-Class ポインタ
11	* pPlotSettings	PlotSettings	PlotSettings-Class ポインタ
12	* pInformation	Information	Information-Class ポインタ
13	* pCanData	CanData	CanData-Class ポインタ
14	* pCanDataMng	CanDataMng	CanDataMng-Class ポインタ
15	* pCanFrame	CanFrame	CanFrame-Class ポインタ
16	* workspace	QWorkspace	FCAN_View 用 Workspace ポインタ
17	* statusBarLabel	QLabel	ステータスバー用ラベル
18	HardWareInfo	QString	ハードウェア設定内容

9. InitFile-Qwidget

9.1. Qwidget(ウィンドウ)



「InitFile」では、HardWare, RxDataFilter, RxDataFormat, TxData 及び Graph の各設定及び保存を行います。

- ① : 設定ファイルを指定します。
- ② : 設定ファイルの検索を行いファイルを指定します。
- ③ : ①にて指定された設定ファイルの記述に従い各設定を実施します。
- ④ : 設定保存ファイルを指定します。
- ⑤ : 設定保存ファイルの検索を行いファイルを指定します。
- ⑥ : ④にて指定された設定ファイルに各設定を保存します。

9.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	HardwareSet	Hardware-Class ポインタコピー
3	RxDataFormatSet	RxDataFormat-Class ポインタコピー
4	RxDataFilterSet	RxDataFilter-Class ポインタコピー
5	TxDataset	TxDat-Class ポインタコピー
6	InformationSet	Information-Class ポインタコピー

9.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.2.2. void HardwareSet(Hardware * hardware)

項	変数	説明
引数	Hardware * hardware	Hardware クラスポインタ
戻値	なし	—

HardwareSet 関数はHardware-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.2.3. void RxDataFormatSet(RxDataFormat * rxdataformat)

項	変数	説明
引数	RxDataFormat * rxdataformat	RxDataFormat クラスポインタ
戻値	なし	—

RxDataFormatSet 関数はRxDataFormat-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.2.4. void RxDataFilterSet(RxDataFilter * rxdatafilter)

項	変数	説明
引数	RxDataFilter * rxdatafilter	RxDataFilter クラスポインタ
戻値	なし	—

RxDataFilterSet 関数はRxDataFilter-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.2.5. void TxDataSet(TxData * txdata)

項	変数	説明
引数	TxData * txdata	TxData クラスポインタ
戻値	なし	—

TxDataSet 関数は TxData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.2.6. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数は Information-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

9.3. スロット宣言

No	関数名	用途
1	closeEvent	InitFile の終了処理関数
2	btnInitLoadFileFind_Click	初期化ファイル検索関数
3	btnInitLoad_Click	初期化ファイル呼び込み関数
4	btnInitSaveFileFind_Click	初期化保存ファイル検索関数
5	btnInitSaveFile_Click	初期化保存ファイル生成関数

9.3.1. void closeEvent(QCloseEvent * pEvent)

項	変数	説明
引数	QCloseEvent * pEvent	FCAN_View-Widget のクローズイベント
戻値	なし	—

InitFile-Widget のクローズイベントが発生した場合、InitFile のウインドウを閉じます。
表示画面フラグの更新を実施します。

9.3.2. void btnInitLoadFileFind_Click()

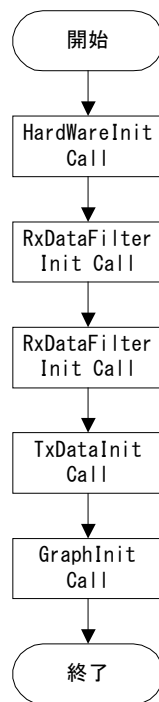
項	変数	説明
引数	なし	—
戻値	なし	—

「Initialize File Load」グループ内の Find ボタンが押された場合の処理を行います。
QFileDialog を開きファイルの設定可能としファイルを指定します。

9.3.3. void btnInitLoad_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Initialize File Load」グループ内の Load File ボタンが押された場合の処理を行います。
指定されたファイルを元に HardWare, RxDataFilter, RxDataFormat, TxData 及び Graph の各指定を実施します。



9.3.4. void btnInitSaveFileFind_Click()

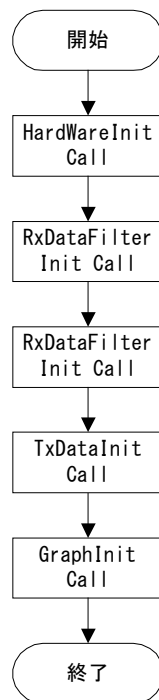
項	変数	説明
引数	なし	—
戻値	なし	—

「Initialize File Save」グループ内の Find ボタンが押された場合の処理を行います。
QFileDialog を開きファイルの設定可能としファイルを指定します。

9.3.5. void btnInitSave_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Initialize File Save」グループ内の Save File ボタンが押された場合の処理を行います。
指定されたファイルに HardWare, RxDataFilter, RxDataFormat, TxData 及び Graph の各設定を保存します。



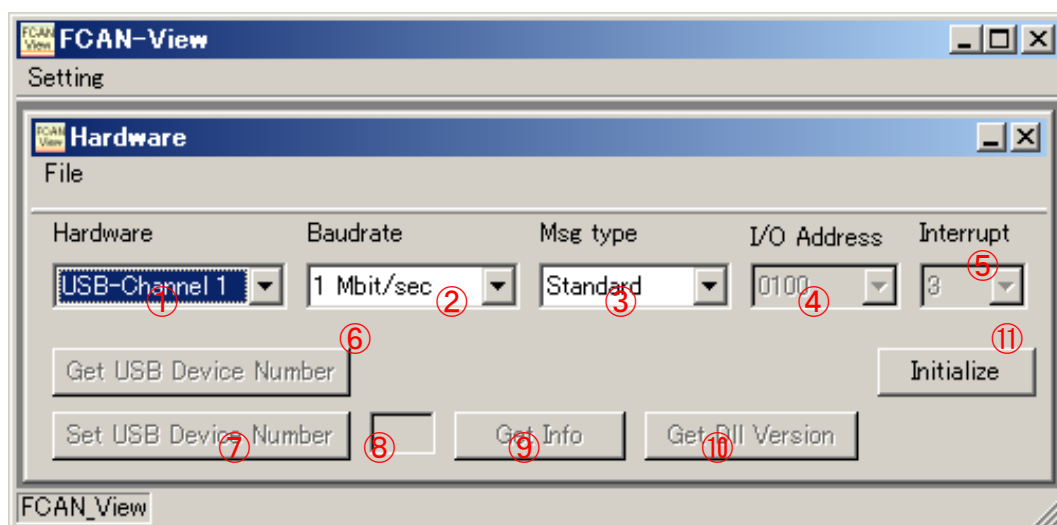
9.4. データメンバー宣言

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pHardWare	HardWare	HardWare-Class ポインタ
3	* pRxDataFormat	RxDataFormat	RxDataFormat-Class ポインタ
4	* pRxDataFilter	RxDataFilter	RxDataFilter-Class ポインタ
5	* pTxData	TxData	TxData-Class ポインタ
6	* pInformation	Information	Information-Class ポインタ

10. Hardware-Qwidget

10.1. Qwidget(ウィンドウ)



- ① : 使用するハードウェア種別を指定します。
- ② : 転送ボーレートを指定します。
- ③ : メッセージタイプを指定します。
- ④ : 使用するハードウェアが ISA バス版の場合 I/O アドレスを指定します。
- ⑤ : 使用するハードウェアが ISA バス版の場合割り込み番号を指定します。
- ⑥ : 使用するハードウェアが USB 版の場合、接続されたデバイス番号を表示します。
- ⑦ : 使用するハードウェアが USB 版の場合、⑧の指定番号をデバイス番号に設定します。
- ⑧ : 使用するハードウェアが USB 版の場合、デバイス番号を入力します
- ⑨ : 現在のハードウェア状態を表示します。
- ⑩ : 現在使用してる『PCAN-BASIC』の DLL 情報を表示します。
- ⑪ : ①～⑤の設定にもとずきドライバをイニシャライズ/開放します。

10.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	InformationSet	Information-Class ポインタコピー
3	timRead	RxData からのメッセージ受信関数
4	HardWareInit	InitFile からの HardWare 設定用関数
5	btnWrite_Click	TxData からのメッセージ送信関数
6	btnResetFilter_Click	RxDataFilter からのフィルター設定関数
7	btnSetFilter_Click	RxDataFilter からのフィルターリセット関数
8	ListenSet	Listen モード設定関数
9	ResetSet	ハードウェアリセット関数

private 関数

No	関数名	用途
1	createMenus	メニュー生成関数
2	createMenuAction	メニューアクション生成関数
3	ReadMessage	メッセージ受信関数
4	ValueCheck	入力文字チェック関数

10.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

10.2.2. void InformationSet(Information * information)

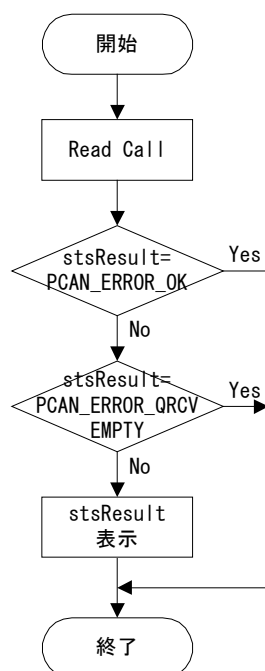
項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数はInformation-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

10.2.3. TPCANStatus timRead(TPCANMsg* MessageBuffer, TPCANTimestamp* TimestampBuffer)

項	変数	説明
引数 1	TPCANMsg* MessageBuf	メッセージバッファポインタ
引数 2	TPCANTimestamp* TimestampBuffer	タイムスタンプバッファポインタ
戻値	TPCANStatus	ステータス

timRead関数はRxData-Classより呼びだされ指定されたメッセージバッファポインタ及びタイムスタンプバッファポインタにデータをライトします。



10.2.4. TPCANStatus btnWrite_Click(TPCANMsg* MessageBuffer)

項	変数	説明
引数	TPCANMsg* MessageBuffer	メッセージバッファポインタ
戻値	TPCANStatus	ステータス

btnWrite_Click関数はTxData-Classより呼びだされ指定されたメッセージバッファポインタのデータを出力します。

10.2.5. TPCANStatus btnSetFilter_Click(DWORD FromID, DWORD ToID, TPCANMode Mode)

項	変数	説明
引数 1	DWORD FromID	フィルタ開始 ID
引数 2	DWORD ToID	フィルタ終了 ID
引数 3	TPCANMode Mode	メッセージタイプ
戻値	TPCANStatus	ステータス

btnSetFilter_Click関数はRxDataFilter-Classより呼びだされ指定された開始IDから終了IDの設定を行います。受信されるIDは、

フィルタ開始ID ≤ 受信ID ≤ フィルタ終了ID
の範囲になります。

10.2.6. TPCANStatus btnResetFilter_Click(DWORD FromID, DWORD ToID, TPCANMode Mode)

項	変数	説明
引数 1	DWORD FromID	フィルタ開始 ID
引数 2	DWORD ToID	フィルタ終了 ID
引数 3	TPCANMode Mode	メッセージタイプ
戻値	TPCANStatus	ステータス

btnResetFilter_Click 関数は RxDataFilter-Class より呼びだされ指定された開始 ID から終了 ID の設定を行います。受信される ID は、

フィルタ開始 ID <= 受信 ID <= フィルタ終了 ID

の範囲になります。btnResetFilter_Click 関数はメッセージタイプにより下記値に固定されています。

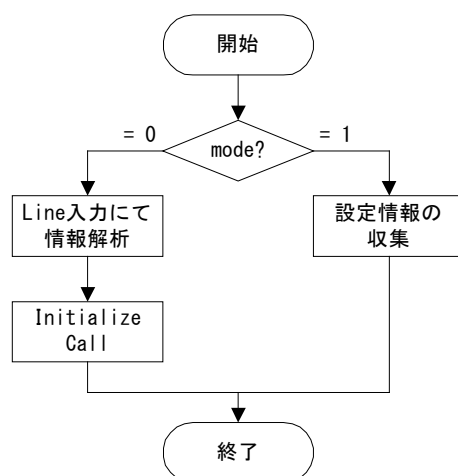
Standard = フィルタ開始 ID (0x0) フィルタ終了 ID (0x7FF)

Extended = フィルタ開始 ID (0x0) フィルタ終了 ID (0x1FFFFFFF)

10.2.7. QString HardWareInit(int mode, QString line)

項	変数	説明
引数 1	int mode	初期化モード (0: Load 1:Save)
引数 2	QString line	InitFile がら読み込んだラインデータ
戻値	QString	HardWare 指定情報出力

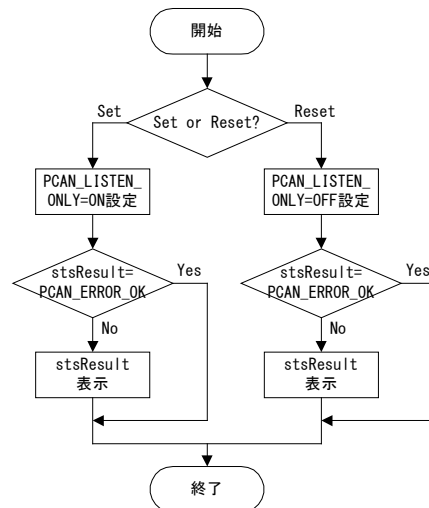
HardWareInit 関数は InitFile-Class より呼びだされた Mode により HardWare の設定及び設定情報の出力を行います。



10.2.8. void ListenSet(int mode)

項	変数	説明
引数	int mode	設定モード(0: OFF 1:ON)
戻値	なし	—

「Set Listen Only」ボタンが押された場合の処理を行います。Listen モードに設定します。



10.2.9. void ResetSet()

項	変数	説明
引数	なし	—
戻値	なし	—

ResetSet 関数はハードウェアをリセットします。

10.2.10. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューの生成を行います。

10.2.11. void createMenuActions()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenuActions 関数はメニューの各アクション生成を行います。

10.2.12. int ValueCheck(QString ValueString)

項	変数	説明
引数	QString ValueString	入力文字列
戻値	int	エラーフラグ

ValueChec 関数は入力文字列が16進値であるかを判断します。

10.3. スロット宣言

No	関数名	用途
1	closeEvent	HardWare の終了処理関数
2	btnDllInfo_Click	DLL 情報取得関数
3	btnGetDevNumber_Click	USB デバイス番号取得関数
4	btnSetDevNumber_Click	USB デバイス番号設定関数
5	btnRelease_Click	デバイスドライバ開放関数
5	btnInit_Click	デバイスドライバ設定関数
6	btnInfo_Click	ハードウェア情報表示関数
8	cbbHws_Changed	ハードウェア種別選択トリガ関数
9	menu_triggered	メニュー選択関数
10	menu_loadfile	初期化ファイルロード関数
11	menu_savefile	初期化ファイルセーブ関数

10.3.1. void closeEvent(QCloseEvent * pEvent)

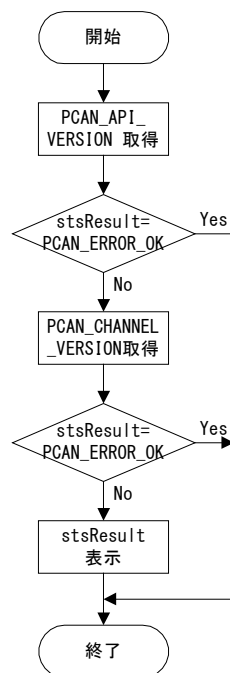
項	変数	説明
引数	QCloseEvent * pEvent	HardWare-Widget のクローズイベント
戻値	なし	—

HardWare-Widget のクローズイベントが発生した場合、HardWare のウィンドウを閉じます。
表示画面フラグの更新を実施します。

10.3.2. void btnDllInfo_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

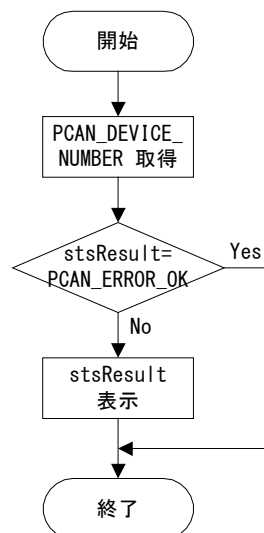
「Get Info」ボタンが押された場合の処理を行います。
PCAN-BASIC 内 GetValue 関数を利用して情報を取得します。



10.3.3. void btnGetDevNumber_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

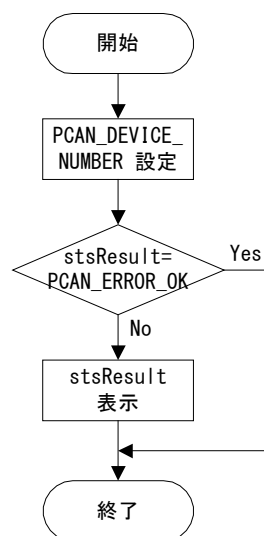
「Get USB Device Number」ボタンが押された場合の処理を行います。
PCAN-BASIC 内 GetValue 関数を利用して情報を取得します。



10.3.4. void btnSetDevNumber_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Set USB Device Number」ボタンが押された場合の処理を行います。
PCAN-BASIC 内 SetValue 関数を利用して設定します。



10.3.5. void btnRelease_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Release」ボタンが押された場合の処理を行います。

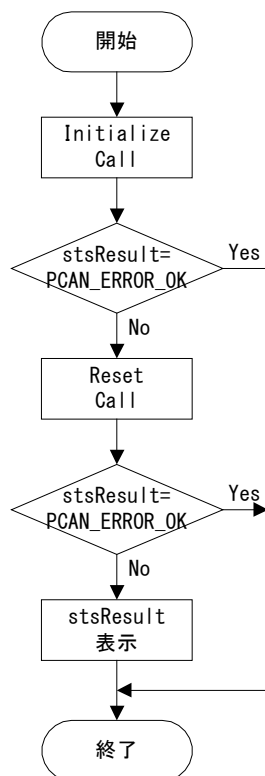
PCAN-BASIC 内 Uninitialize 関数を利用して設定します。

10.3.6. void btnInit_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Initialize」ボタンが押された場合の処理を行います。

PCAN-BASIC 内 Uninitialize 関数を利用して設定し、PCAN-BASIC 内 Reset 関数にて初期化します。

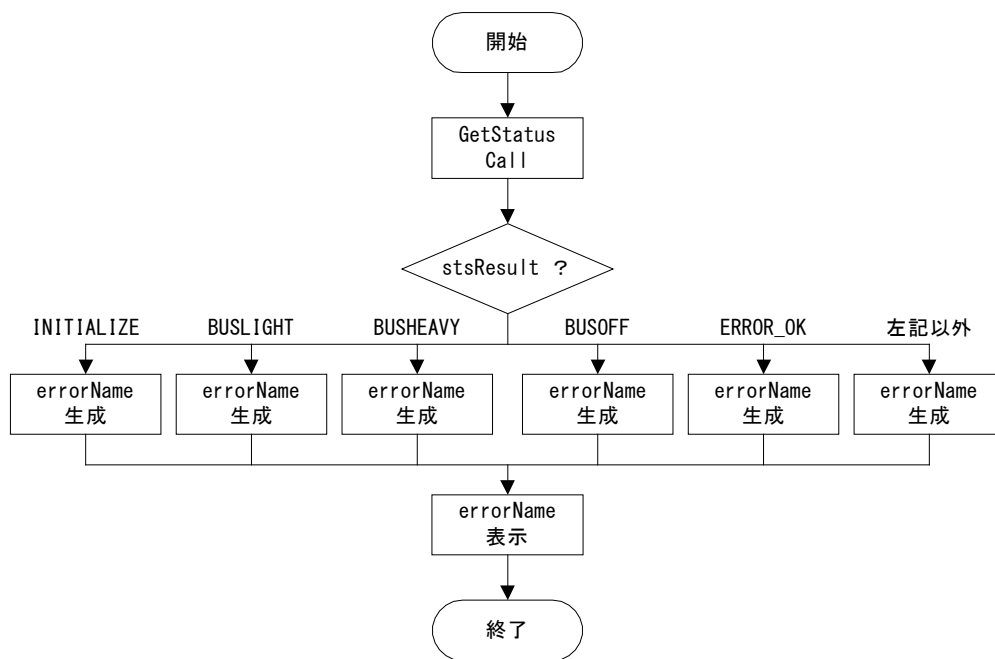


10.3.7. void btnInfo_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Get Info」ボタンが押された場合の処理を行います。

PCAN-BASIC 内 GetStatus 関数を利用して情報を取得します。



10.3.8. void menu_triggered()

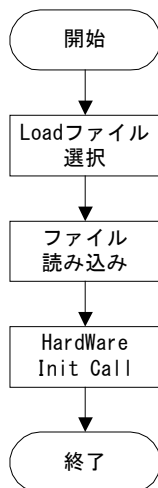
項	変数	説明
引数	なし	—
戻値	なし	—

メニューが選択された時、再メニューアクションを生成します。

10.3.9. void menu_loadfile()

項	変数	説明
引数	なし	—
戻値	なし	—

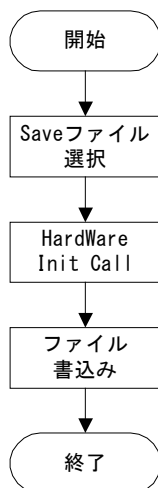
メニュー内「Load File」が選択された時、指定されたファイルデータをロードし設定します。



10.3.10. void menu_savefile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニュー内「Save File」が選択された時、設定情報を指定されたファイルにセーブします。



10.3.11. void cbbHws_Changed(const QString &s)

項	変数	説明
引数	const QString &s	選択アイテム
戻値	なし	—

「HardWare」にてハードウェア種別が選択された場合の処理を行います。
必要のない項目を非表示にします。

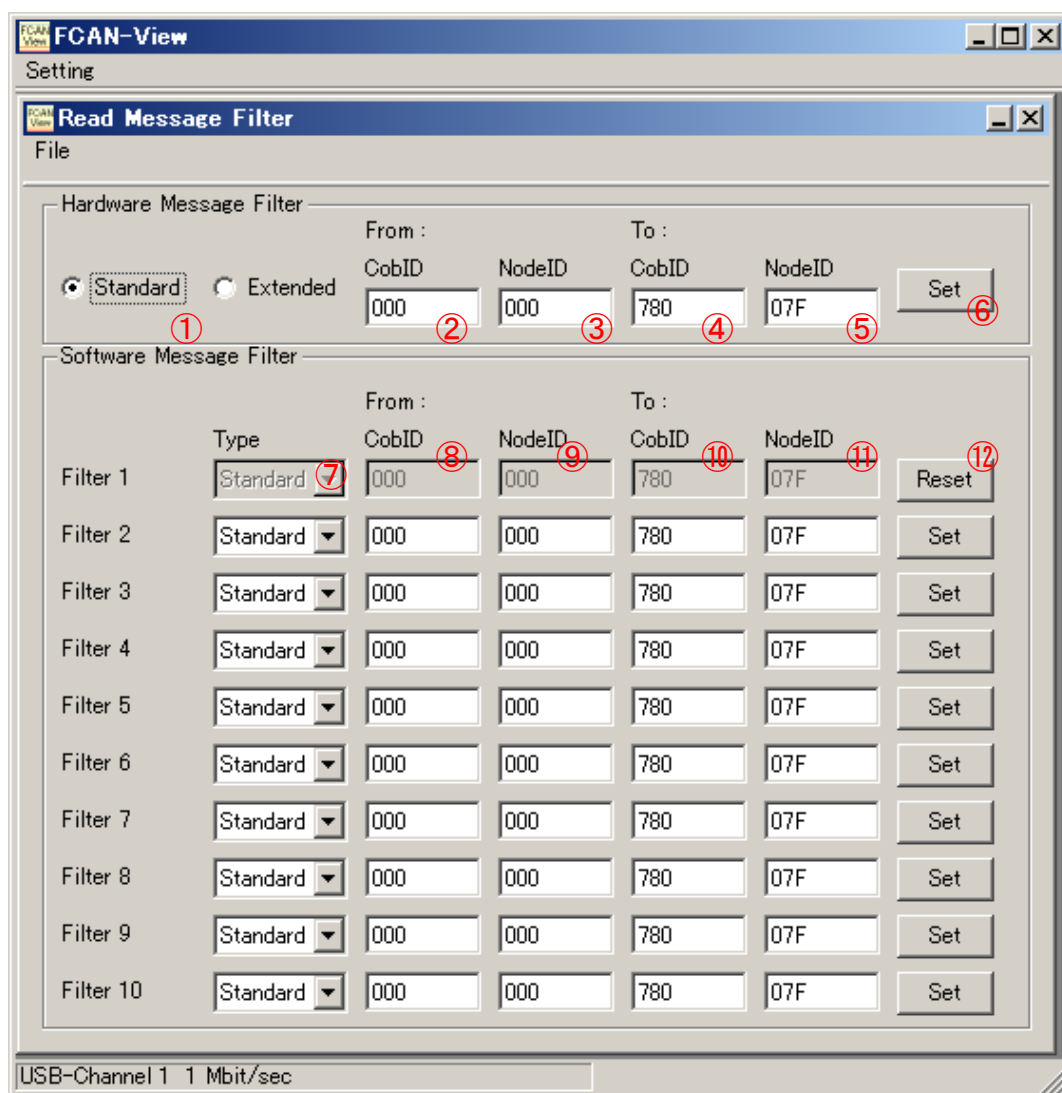
10.4. データメンバー宣言

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pInformation	Information	Information-Class ポインタ
3	HardWareIoInterruptSelect	int	IO/Interrupt 選択フラグ

11. RxDataFilter-Qwidget

11.1. Qwidget(ウインドウ)



- ① : ハードウェアフィルタのメッセージタイプを指定します。
- ② : ハードウェアフィルタの開始 COB-ID 指定します。
- ③ : ハードウェアフィルタの開始 ID 指定します。
- ④ : ハードウェアフィルタの終了 COB-ID 指定します。
- ⑤ : ハードウェアフィルタの終了 ID 指定します。
- ⑥ : ハードウェアフィルタを設定/解除します。
- ⑦ : ソフトウェアフィルタの設定番号を指定します。
- ⑧ : ソフトウェアフィルタのメッセージタイプを指定します。
- ⑨ : ソフトウェアフィルタの開始 COB-ID 指定します。
- ⑩ : ソフトウェアフィルタの開始 ID 指定します。
- ⑪ : ソフトウェアフィルタの終了 COB-ID 指定します。
- ⑫ : ソフトウェアフィルタを設定/解除します。

* Filter2~Filter10 までも同様に設定します。

11.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	InformationSet	Information-Class ポインタコピー
3	HardWareSet	HardWare-Class ポインタコピー
4	RxDataFilterInit	InitFile からの RxDataFilter 設定用関数

private 関数

No	関数名	用途
1	createMenus	メニュー生成関数
2	createMenuAction	メニューアクション生成関数
3	ValueCheck	入力文字列チェック関数

11.2.1. void FcanViewSet(FcanView * fcanview)

項	変数	説明
引数	FcanView * fcanview	FcanView クラスポインタ
戻値	なし	—

FcanViewSet 関数は FCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

11.2.2. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数は Information-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

11.2.3. void HardWareSet(HardWare * hardware)

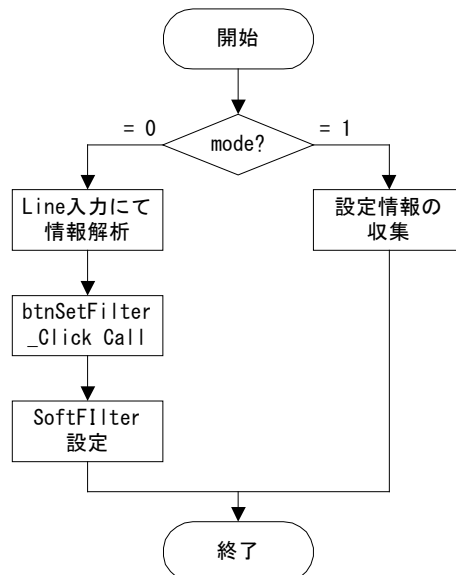
項	変数	説明
引数	HardWare * hardware	HardWare クラスポインタ
戻値	なし	—

HardWareSet 関数は HardWare-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

11.2.4. QString RxDataFilterInit(int mode, QString line)

項	変数	説明
引数 1	int mode	初期化モード (0: Load 1:Save)
引数 2	QString line	InitFile がら読み込んだラインデータ
戻値	QString	RxDataFilter 指定情報出力

RxDataFilterInit 関数は InitFile-Class より呼びだされた Mode により RxDataFilter の設定及び設定情報の出力を行います。



11.2.5. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューの生成を行います。

11.2.6. void createMenuActions()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenuActions 関数はメニューの各アクション生成を行います。

11.2.7. int ValueCheck(QString ValueString)

項	変数	説明
引数	QString ValueString	入力文字列
戻値	int	エラーフラグ

ValueChec 関数は入力文字列が 16 進値であるかを判断します。

11.3. スロット宣言

No	関数名	用途
1	closeEvent	RxDataFilter の終了処理関数
2	BtnHWfilterSetReset_Click	HW フィルタ設定/解除関数
3	BtnSWfilterSetReset_1_Click	SW フィルタ設定/解除関数 1
4	BtnSWfilterSetReset_2_Click	SW フィルタ設定/解除関数 2
5	BtnSWfilterSetReset_3_Click	SW フィルタ設定/解除関数 3
6	BtnSWfilterSetReset_4_Click	SW フィルタ設定/解除関数 4
7	BtnSWfilterSetReset_5_Click	SW フィルタ設定/解除関数 5
8	BtnSWfilterSetReset_6_Click	SW フィルタ設定/解除関数 6
9	BtnSWfilterSetReset_7_Click	SW フィルタ設定/解除関数 7
10	BtnSWfilterSetReset_8_Click	SW フィルタ設定/解除関数 8
11	BtnSWfilterSetReset_9_Click	SW フィルタ設定/解除関数 9
12	BtnSWfilterSetReset_10_Click	SW フィルタ設定/解除関数 10
13	menu_triggered	メニュー選択関数
14	menu_loadfile	初期化ファイルロード関数
15	menu_savefile	初期化ファイルセーブ関数

11.3.1. void closeEvent(QCloseEvent * pEvent)

項	変数	説明
引数	QCloseEvent * pEvent	RxDataFilter-Widget のクローズイベント
戻値	なし	—

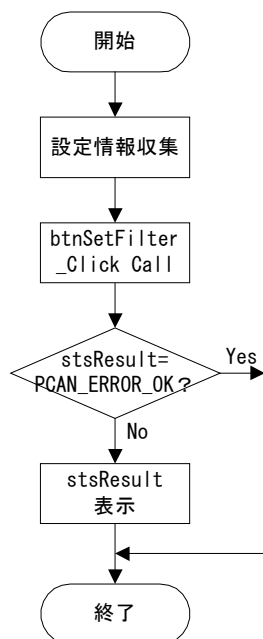
RxDataFilter-Widget のクローズイベントが発生した場合、RxDataFilter のウインドウを閉じます。
表示画面フラグの更新を実施します。

11.3.2. void btnHWfilterSetReset_Click()

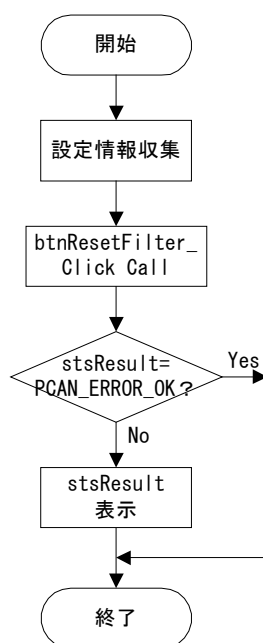
項	変数	説明
引数	なし	—
戻値	なし	—

「HardWare Message Filter」内の「Set」/「Reset」ボタンが押された場合の処理を行います。
HardWare-Class 内 btnSetFilter_Click 関数を利用して情報を取得します。

「Set」時の処理



「Reset」時の処理

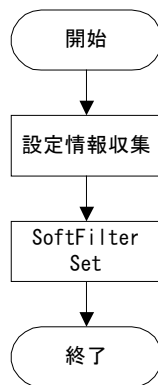


11.3.3. void btnSWfilterSetReset_1_Click()

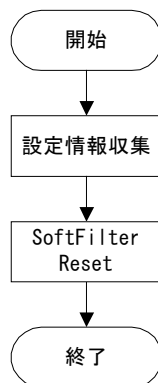
項	変数	説明
引数	なし	—
戻値	なし	—

「SoftWare Message Filter」内の「Set」/「Reset」ボタンが押された場合の処理を行います。

「Set」時の処理



「Reset」時の処理

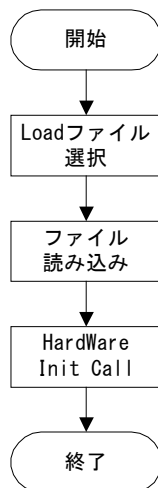


* void btnSWfilterSetReset_2_Click()～void btnSWfilterSetReset_10_Click() も同様な処理を実施しています。

11.3.4. void menu_loadfile()

項	変数	説明
引数	なし	—
戻値	なし	—

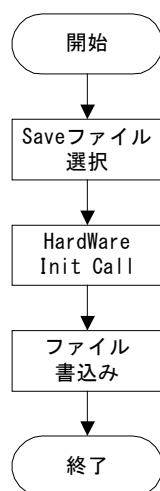
メニュー内「Load File」が選択された時、指定されたファイルデータをロードし設定します。



11.3.5. void menu_savefile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニュー内「Save File」が選択された時、設定情報を指定されたファイルにセーブします。



11.4. データメンバー宣言

public 変数

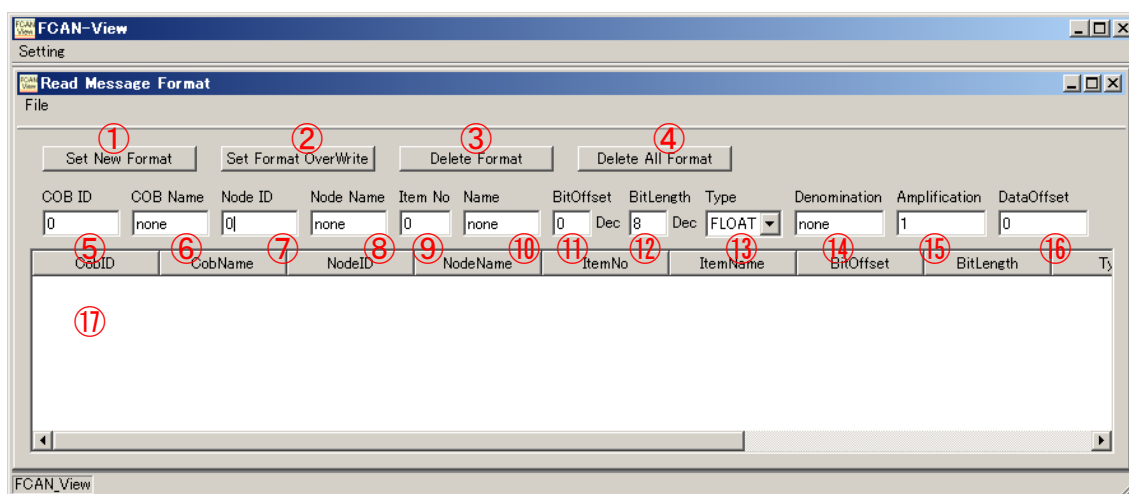
No	変数名	型	用途
1	SoftMinValue	Unsigned long int [10]	ソフトウェアフィルタ開始 CobID+ID
2	SoftMaxValue	Unsigned long int [10]	ソフトウェアフィルタ終了 CobID+ID
3	SoftSet	Unsigned long int [10]	ソフトウェアフィルタ設定状況

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pInformation	Information	Information-Class ポインタ
3	* pHardWare	HardWare	HardWare-Class ポインタ
4	SoftStandard	Unsigned long int [10]	ソフトウェアフィルタメッセージタイプ

12. RxDataFormat-Qwidget

12.1. Qwidget(ウィンドウ)



- ① : 新規フォーマットを追加します。
- ② : 指定されたフォーマットに上書きします。
- ③ : 指定されたフォーマットを削除します。
- ④ : 現在設定されているフォーマットを全て削除します。
- ⑤ : COB-ID を設定します (Hex 値)。
- ⑥ : COB-ID 名称を設定します。
- ⑦ : ノード ID を設定します (Hex 値)。
- ⑧ : ノード ID 名称を設定します。
- ⑨ : アイテム No を設定します (Dec 値)。
- ⑩ : アイテム名称を設定します。
- ⑪ : ビット位置オフセットを指定します (Dec 値)。
- ⑫ : ビット長を指定します (Dec 値)。
- ⑬ : アイテム値の型を指定します。
- ⑭ : アイテム値の単位を設定します。
- ⑮ : アイテム値の倍率を設定します (Float 値)。
- ⑯ : アイテム値のオフセット値を設定します。
- ⑰ : 現在設定されているフォーマット状況を表示します。

内部データ値は下記の値になります。

内部データ値 = (受信アイテム値 × アイテム値の倍率) + アイテム値のオフセット値

12.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	InformationSet	Information-Class ポインタコピー
3	CanFrameSet	CanFormat-Class ポインタコピー
4	RxDataFormatInit	InitFile からの RxDataFormat 設定用関数
5	ViewFormat	既存フォーマット表示関数

private 関数

No	関数名	用途
1	createMenus	メニュー生成関数
2	createMenuAction	メニューアクション生成関数
3	DelFormat	フォーマット行削除関数
4	DelFormatAll	フォーマットバッファクリア関数
5	ViewFormatWrite	表示フォーマット格納関数
6	ValueCheck	入力文字列チェック関数

12.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数は FCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

12.2.2. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数は Information-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

12.2.3. void CanFrameSet(CanFrame * canframe)

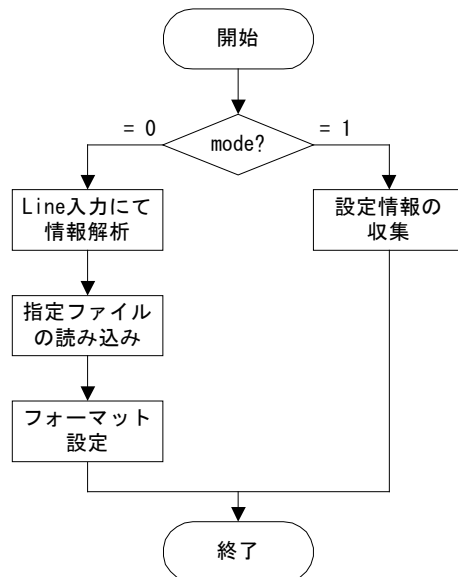
項	変数	説明
引数	CanFrame * canframe	CanFrame クラスポインタ
戻値	なし	—

CanFrameSet 関数は CanFrame-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

12.2.4. QString RxDataFormatInit(int mode, QString line)

項	変数	説明
引数 1	int mode	初期化モード (0: Load 1:Save)
引数 2	QString line	InitFile がら読み込んだラインデータ
戻値	QString	RxDataFormat 指定情報出力

RxDataFormatInit 関数は InitFile-Class より呼びだされた Mode により RxDataFormat の設定及び設定情報の出力を行います。



RxDataFormat の初期化は、事前にファイルロードにて設定した場合のみ有効です。

12.2.5. void ViewFormat()

項	変数	説明
引数	なし	—
戻値	なし	—

ViewFormat 関数は現在登録されているフォーマットを全て表示します。

12.2.6. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューの生成を行います。

12.2.7. void createMenuActions()

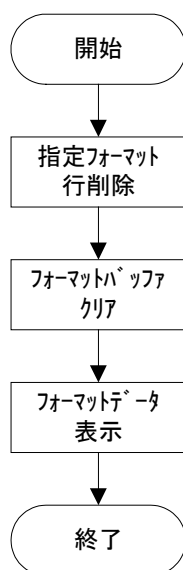
項	変数	説明
引数	なし	—
戻値	なし	—

createMenuActions 関数はメニューの各アクション生成を行います。

12.2.8. void DelFormat()

項	変数	説明
引数	なし	—
戻値	なし	—

指定されたフォーマットを削除します。



12.2.9. void DelFormatAll()

項	変数	説明
引数	なし	—
戻値	なし	—

フォーマットバッファ内のすべてのフォーマットデータを削除します。

12.2.10. void ViewFormatWrite()

項	変数	説明
引数	なし	—
戻値	なし	—

表示されているすべてのフォーマットデータをフォーマットバッファに格納します。

12.2.11. int ValueCheck(QString ValueString)

項	変数	説明
引数	QString ValueString	入力文字列
戻値	int	エラーフラグ

ValueChec 関数は入力文字列が 16 進値であるかを判断します。

12.3. スロット宣言

No	関数名	用途
1	closeEvent	RxDataFormat の終了処理関数
2	btnSetFormat_Click	個別フォーマット設定関数
3	btnOverWriteFormat_Click	個別フォーマット上書き設定関数
4	btnDeleteFormat_Click	フォーマット削除関数
5	btnDeleteAllFormat_Click	全フォーマット削除関数
6	menu_triggered	メニュー表示切替え関数
7	menu_loadfile	フォーマットファイル読み込み関数
8	menu_savefile	フォーマットファイル書き込み関数
9	tblView_CellClicked	フォーマット表示シングルクリック関数
10	tblView_CellDoubleClicked	フォーマット表示ダブルクリック関数
11	resizeEvent	QWidget サイズ変更関数
12	keyPressEvent	キー入力関数

12.3.1. void closeEvent(QCloseEvent * pEvent)

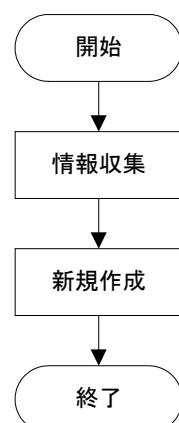
項	変数	説明
引数	QCloseEvent * pEvent	RxDataFormat-Widget のクローズイベント
戻値	なし	—

RxDataFormat-Widget のクローズイベントが発生した場合、RxDataFormat のウィンドウを閉じます。
表示画面フラグの更新を実施します。

12.3.2. void btnSetFormat_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Set New Format」ボタンが押された場合の処理を行います。
設定されたフォーマットデータを新たに作成します。

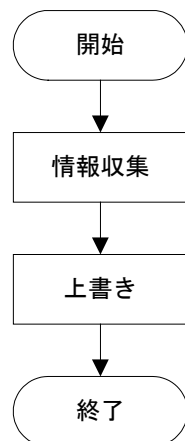


12.3.3. void btnOverWriteFormat_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Set Format OverWrite」ボタンが押された場合の処理を行います。

選択されたフォーマットを削除し、設定されたフォーマットデータに上書きし作成します。



12.3.4. void btnDeleteFormat_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

選択されたフォーマットを削除します。

12.3.5. void btnDeleteAllFormat_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

すべてのフォーマットを削除します。

12.3.6. void menu_triggered()

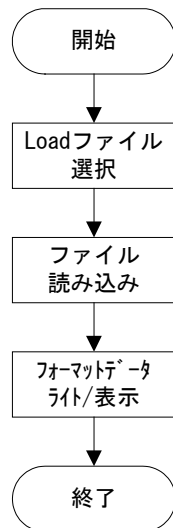
項	変数	説明
引数	なし	—
戻値	なし	—

メニューが選択された時、再メニューアクションを生成します。

12.3.7. void menu_loadfile()

項	変数	説明
引数	なし	—
戻値	なし	—

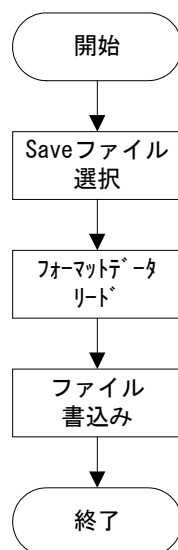
メニュー内「Open RxDataFormat」が選択された時、指定されたファイルデータをロードし設定します。



12.3.8. void menu_savefile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニュー内「Save RxDataFormat」が選択された時、設定情報を指定されたファイルにセーブします。



12.3.9. void tblView_CellClicked(int row, int column)

項	変数	説明
引数 1	int row	行位置
引数 2	int column	列位置
戻値	なし	—

フォーマットデータ表示部にてシングルクリックされた場合、指定された行を選択表示にします。

12.3.10. void tblView_CellDoubleClicked(int row, int column)

項	変数	説明
引数 1	int row	行位置
引数 2	int column	列位置
戻値	なし	—

フォーマットデータ表示部にてダブルクリックされた場合、指定された行を選択表示にします。
選択されたフォーマットデータを上部編集可能エリアにコピーします。

12.3.11. void resizeEvent(QResizeEvent *event)

項	変数	説明
引数	QResizeEvent *event	リサイズイベント
戻値	なし	—

RxDataFormat-Widget サイズが変更された場合、フォーマット表示エリアを拡大/縮小します。

12.3.12. void keyPressEvent(QKeyEvent *event)

項	変数	説明
引数	QKeyEvent *event	キー入力イベント
戻値	なし	—

キー入力情報を判断し「DEL」キー入力を判断します。

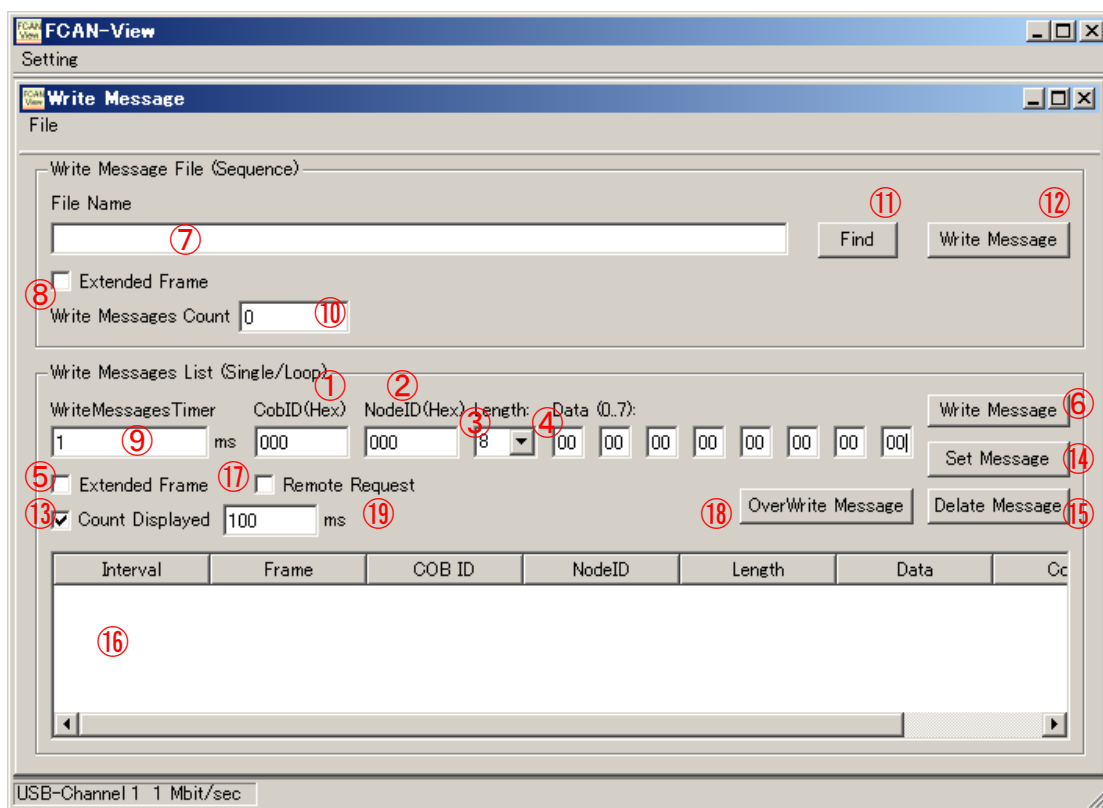
12.4. データメンバー宣言

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pInformation	Information	Information-Class ポインタ
3	* pCanFrame	CanFrame	CanFormat-Class ポインタ
4	FormatLoadSettingMode	QString	フォーマットファイル状況保持
5	FormatLoadSettingName	QString	フォーマットファイル名保持

13. TxData-Qwidget

13.1. Qwidget(ウインドウ)



- ① : 送信メッセージ COB-ID を設定します (Hex 値)。
- ② : 送信メッセージ ID を設定します (Hex 値)。
- ③ : 送信メッセージのデータバイト数を設定します。
- ④ : 送信メッセージのデータを設定します。左側から 0, 1, 2, 3, 4, 5, 6, 7 の順に出力されます (Hex 値)。
- ⑤ : 送信メッセージタイプの EXTENDED タイプ設定を選択します。
- ⑥ : ①～⑤に設定されたメッセージを送信します。
- ⑦ : 送信メッセージ用ファイル名を指定します。
- ⑧ : 送信メッセージタイプの EXTENDED タイプ設定を選択します。
- ⑨ : 送信メッセージ間隔を設定します (Dec 値)。
- ⑩ : 送信メッセージ数を表示します (Dec 値)。
- ⑪ : 送信メッセージ用ファイルを検索します。
- ⑫ : 送信メッセージ用ファイルに記載された送信メッセージを送信します。
- ⑬ : 送信メッセージカウント表示間隔を選択します。
- ⑭ : ①～⑤に設定された送信メッセージを⑯に格納します。
- ⑮ : ⑯内で選択された送信メッセージを削除します。
- ⑯ : 送信メッセージを表示します。(送信メッセージリスト)
- ⑰ : 送信メッセージを RemoteRequest に設定します。
- ⑱ : ①～⑤に設定された送信メッセージを⑯の選択されたメッセージに上書きします。
- ⑲ : 送信メッセージカウント表示間隔を設定します。

13.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	InformationSet	Information-Class ポインタコピー
3	HardWareSet	HardWare-Class ポインタコピー
4	TxDatInit	送信メッセージ初期化

private 関数

No	関数名	用途
1	Write_Message	送信メッセージ関数呼び出し関数
2	SetMessage	送信メッセージ格納関数
3	createMenus	メニュー生成関数
4	createMenuAction	メニューアクション生成関数
5	ValueCheck	入力文字列チェック関数

13.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

13.2.2. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数はInformation-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

13.2.3. void HardWareSet(HardWare * hardware)

項	変数	説明
引数	HardWare * hardware	HardWare クラスポインタ
戻値	なし	—

HardWareSet 関数はHardWare-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

13.2.4. TPCANStatus Write_Message(TPCANMsg* MessageBuffer)

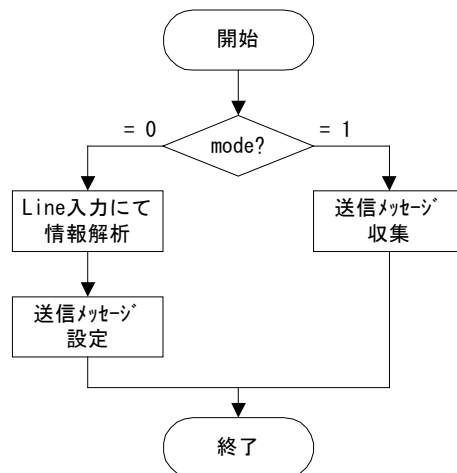
項	変数	説明
引数	TPCANMsg* MessageBuffer	メッセージバッファポインタ
戻値	TPCANStatus	ステータス

HardWare-Class 内の「btnWrite_Click」関数を呼出します。

13.2.5. QString TxDataInit(int mode, QString line)

項	変数	説明
引数1	int mode	初期化モード(0: Load 1:Save)
引数2	QString line	InitFile がら読み込んだラインデータ
戻値	QString	TxData 指定情報出力

TxDataInit 関数は InitFile-Class より呼びだされた Mode により TxData の設定及び設定情報の出力を行います。主に「送信メッセージリスト」の設定及び保存を行います。



13.2.6. void SetMessage()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Set Message」ボタンが押された場合の処理を行います。設定された送信メッセージを格納します。

13.2.7. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューの生成を行います。

13.2.8. void createMenuActions()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenuActions 関数はメニューの各アクション生成を行います。

13.2.9. int ValueCheck(QString ValueString)

項	変数	説明
引数	QString ValueString	入力文字列
戻値	int	エラーフラグ

ValueChec 関数は入力文字列が 16 進値であるかを判断します。

13.3. スロット宣言

No	関数名	用途
1	closeEvent	TxData の終了処理関数
2	btnWrite_Click	送信メッセージ開始関数
3	btnWriteFile_Click	送信メッセージファイル開始関数
4	btnWriteFileFind_Click	送信メッセージファイル検索関数
5	btnSetMessage_Click	送信メッセージ格納関数
6	btnDelMessage_Click	送信メッセージ削除関数
7	btnOverWriteMessage_Click	送信メッセージ上書き関数
8	chbTxTimer_Toggled	送信メッセージタイマ選択関数
9	cbbTxLength_Changed	送信メッセージデータ長選択関数
10	chbRemoteRequest_Toggled	送信 RTR メッセージデータ選択関数
11	chbDisplay_Toggled	送信数表示選択関数
12	tblTxData_CellClicked	送信メッセージ選択関数
13	tblTxData_CellDoubleClicked	送信メッセージダブルクリック関数
14	timWrite	送信タイマ関数
15	menu_triggered	メニュー選択関数
16	menu_loadfile	初期化ファイルロード関数
17	menu_savefile	初期化ファイルセーブ関数
18	resizeEvent	QWidget サイズ変更関数
19	keyPressEvent	キー入力関数

13.3.1. void closeEvent(QCloseEvent * pEvent)

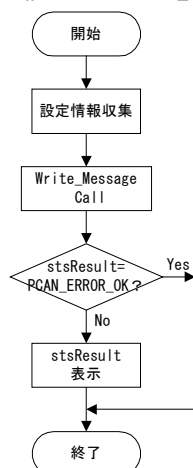
項	変数	説明
引数	QCloseEvent * pEvent	TxData-Widget のクローズイベント
戻値	なし	—

TxData-Widget のクローズイベントが発生した場合、TxData のウィンドウを閉じます。
表示画面フラグの更新を実施します。

13.3.2. void btnWrite_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Write Message」ボタンが押された場合の処理を行います。
設定された送信メッセージを出力します。



13.3.3. void btnWriteFileFind_Click()

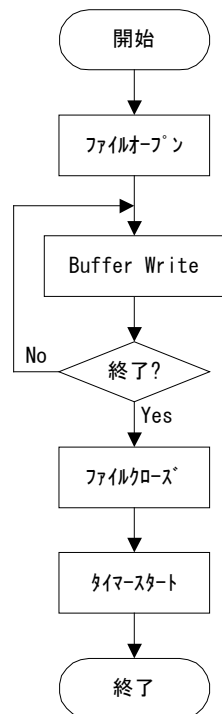
項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages File」グループ内「Find」ボタンが押された場合の処理を行います。
QFileDialog を開きファイルの設定可能としファイルを指定します。

13.3.4. void btnWriteFile_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages File」グループ内「Write Message」ボタンが押された場合の処理を行います。
指定されたファイルを開き、順番に送信メッセージを出力します。



13.3.5. void btnSetMessage_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Set Message」ボタンが押された場合の処理を行います。
設定された送信メッセージを格納します。

13.3.6. void btnOverWriteMessage_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Over Write Message」ボタンが押された場合の処理を行います。
設定された送信メッセージを上書きします。

13.3.7. void btnDelMessage_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Delete Message」ボタンが押された場合の処理を行います。
設定された送信メッセージを削除します。

13.3.8. void chbTxTimer_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Write Message Timer」チェックボタンが変化した場合の処理を行います。
設定によりタイマ設定を非表示にします。

13.3.9. void cbbTxLength_Changed(const QString &s)

項	変数	説明
引数	const QString &s	選択アイテム
戻値	なし	—

「Write Messages」グループ内「Length」が変化した場合の処理を行います。
設定によりデータ設定を変化させます。

13.3.10. void chbRemoteRequest_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「RemoteRequest」が変化した場合の処理を行います。
設定により RTR 送信を設定します。

13.3.11. void chbDisplay_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Write Messages」グループ内「Count Displayed」が変化した場合の処理を行います。
設定により連続送信時の送信カウント表示期間を有効にします。

13.3.12. void tbITxData_CellClicked(int row, int column)

項	変数	説明
引数1	int row	ROW 値
引数2	int column	COLUMN 値
戻値	なし	—

「Write Messages」グループ内送信データ格納ボックスのある場所が選択された場合の処理を行います。
選択された ROW 値により ROW 一行を選択色に設定します。

13.3.13. void tbITxData_CellDoubleClicked(int row, int column)

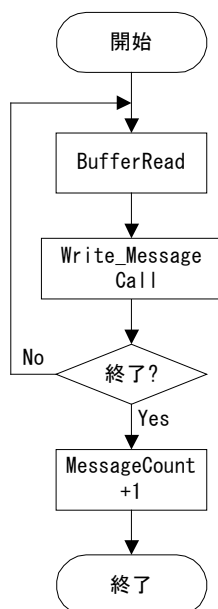
項	変数	説明
引数1	int row	ROW 値
引数2	int column	COLUMN 値
戻値	なし	—

「Write Messages」グループ内送信データ格納ボックスのある場所が選択された場合の処理を行います。
選択された ROW 値により ROW 一行を選択色に設定します。選択された列が「CobID」、「NodeD」、「Data」
の場合、選択された送信データを送信メッセージ作成場所にコピーします。
選択された列が「CobID」、「NodeD」の場合データ送信処理を実施します。

13.3.14. void timWrite()

項	変数	説明
引数	なし	—
戻値	なし	—

btnWriteFile_Click 関数にて起動されたタイマが設定時間毎に呼び出されます。
バッファに待避された送信メッセージを送信します。



13.3.15. void menu_triggered()

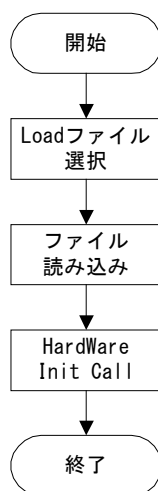
項	変数	説明
引数	なし	—
戻値	なし	—

メニューが選択された時、再メニューアクションを生成します。

13.3.16. void menu_loadfile()

項	変数	説明
引数	なし	—
戻値	なし	—

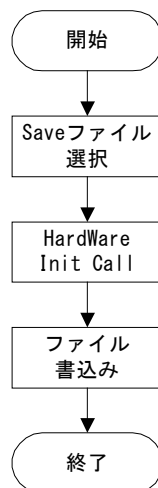
メニュー内「Load File」が選択された時、指定されたファイルデータをロードし設定します。



13.3.17. void menu_savefile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニュー内「Save File」が選択された時、設定情報を指定されたファイルにセーブします。



13.3.18. void resizeEvent (QResizeEvent *event)

項	変数	説明
引数	QResizeEvent *event	リサイズイベント
戻値	なし	—

TxData-Widget サイズが変更された場合、フォーマット表示エリアを拡大/縮小します。

13.3.19. void keyPressEvent (QKeyEvent *event)

項	変数	説明
引数	QKeyEvent *event	キー入力イベント
戻値	なし	—

キー入力情報を判断し「DEL」キー入力を判断します。

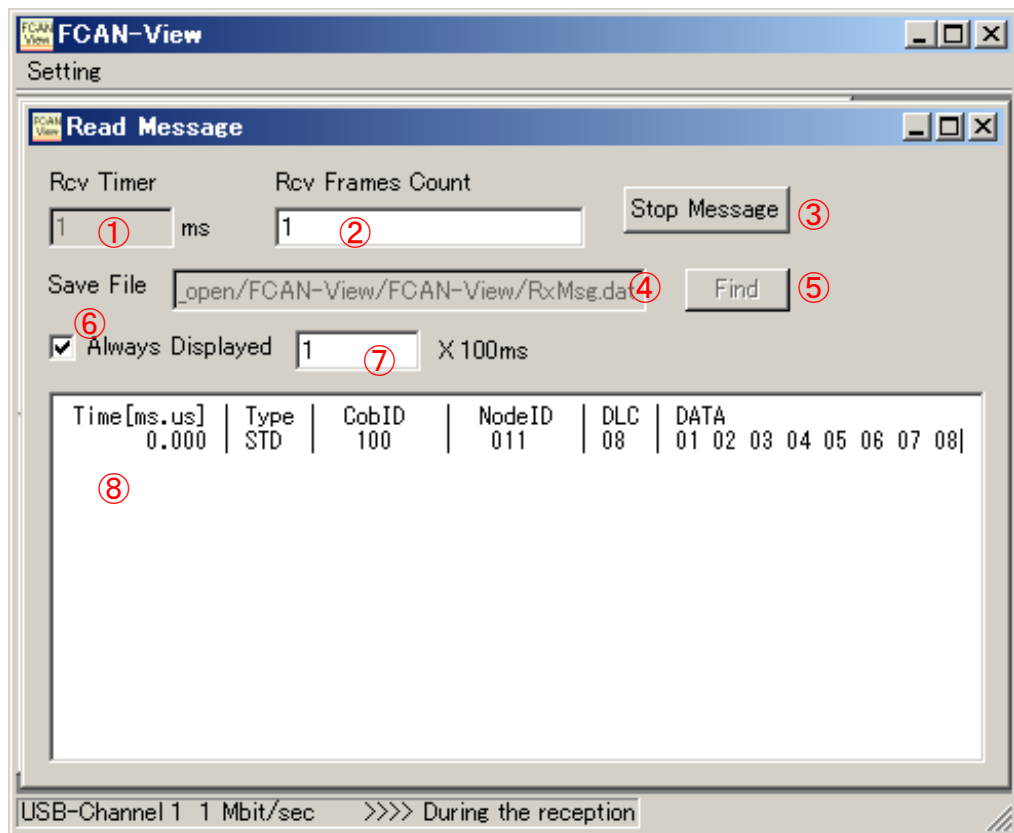
13.4. データメンバー宣言

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pInformation	Information	Information-Class ポインタ
3	* pHardWare	HardWare	HardWare-Class ポインタ
4	*WriteTimer	QTimer	WriteTimer ポインタ
5	WriteFileLine	int	ファイル呼び込み現在位置
6	WriteFileLineMax	int	ファイル呼び込み最大値
7	MessageCount	int	送信メッセージ数
8	TxMode	int	送信モード
9	TxRow	int	送信メッセージ ROW 値
10	TxRowSelect	int	送信メッセージ選択 ROW 値
11	IntervalCount	int	送信メッセージ送信時間
12	RemoteRequestMode	int	送信 RTR モード
13	DoubleClickFlag	int	送信メッセージリストダブルクリックフラグ
14	OverWriteFlag	int	送信メッセージオーバーライトフラグ
15	TimerCountCycle	int	送信タイマカウント
16	DisplayTime	int	送信メッセージ表示時間
17	TxDataLine	QStringList	送信メッセージリスト

14. RxData-Qwidget

14.1. Qwidget(ウインドウ)



- ① : メッセージを受信するタイマ設定間隔を設定します (Dec 値)。
- ② : 受信メッセージ数を設定します (Dec 値)。
- ③ : 受信動作を開始/停止します。
- ④ : 受信したメッセージを保存するファイル名を指定します。
- ⑤ : 受信したメッセージを保存するファイルを検索します。
- ⑥ : 受信メッセージの表示/非表示を切り替えます。
- ⑦ : 受信メッセージの表示間隔を設定します (Dec 値)。
- ⑧ : 受信メッセージを表示します。

14.2. メンバー関数

public 関数

No	関数名	用途
1	FcanViewSet	FCAN_View-Class ポインタコピー
2	InformationSet	Information-Class ポインタコピー
3	HardWareSet	HardWare-Class ポインタコピー
4	SysConfigSet	SysConfig-Class ポインタコピー
5	RxDataFilterSet	RxDataFilter-Class ポインタコピー
6	CanDataSet	CanDataSet-Class ポインタコピー
7	CanDataMngSet	CanDataMngSet-Class ポインタコピー

private 関数

No	関数名	用途
1	Read_Message	受信メッセージ関数呼び出し関数
2	RxDataFileWrite	受信メッセージファイルライト関数
3	ProcessMessage	受信メッセージ表示/保存関数
4	RxDataTime	受信メッセージ時間変換関数

14.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.2. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数はInformation-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.3. void HardWareSet(HardWare * hardware)

項	変数	説明
引数	HardWare * hardware	HardWare クラスポインタ
戻値	なし	—

HardWareSet 関数はHardWare-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.4. void SysConfigSet(SysConfig * sysconfig)

項	変数	説明
引数	SysConfig * sysconfig	SysConfig クラスポインタ
戻値	なし	—

SysConfigSet 関数は SysConfig-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.5. void RxDataFilterSet(RxDataFilter * rxdatafilter)

項	変数	説明
引数	RxDataFilter * rxdatafilter	RxDataFilter クラスポインタ
戻値	なし	—

RxDataFilterSet 関数は RxDataFilter-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.6. void CanDataSet(CanData * candata)

項	変数	説明
引数	CanData * candata	CanData クラスポインタ
戻値	なし	—

CanDataSet 関数は CanData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.7. void CanDataMngSet(CanDataMng * candatamng)

項	変数	説明
引数	CanDataMng * candatamng	CanDataMng クラスポインタ
戻値	なし	—

CanDataMngSet 関数は CanDataMng-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

14.2.8. TPCANStatus Read_Message (TPCANMsg* MessageBuffer, TPCANTimestamp* TimestampBuffer)

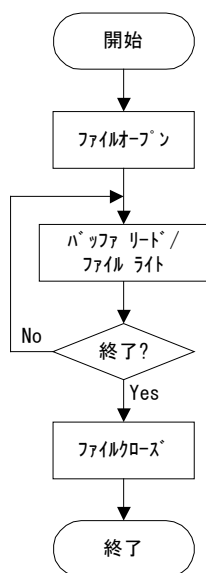
項	変数	説明
戻値 1	TPCANMsg* MessageBuffer	受信メッセージバッファポインタ
戻値 2	TPCANTimestamp* TimestampBuffer	受信タイムスタンプバッファポインタ
戻値	TPCANStatus	ステータス

HardWare-Class 内の「timRead」関数を呼び出します。

14.2.9. void RxDataFileWrite (DWORD Count)

項	変数	説明
引数	DWORD Count	受信メッセージ数
戻値	なし	—

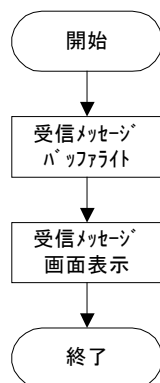
RxDataFileWrite 関数は引数 Count 数分受信メッセージバッファのデータをファイルに出力します。



14.2.10. void ProcessMessage (TPCANMsg MyMsg, TPCANTimestamp MyTimeStamp)

項	変数	説明
戻値 1	TPCANMsg* MessageBuffer	受信メッセージバッファポインタ
戻値 2	TPCANTimestamp* TimestampBuffer	受信タイムスタンプバッファポインタ
戻値	なし	—

HardWare-Class 内の「timRead」関数を呼び出します。



14.2.11. void RxDataTime (TPCANTimestamp MyTimeStamp)

項	変数	説明
引数	TPCANTimestamp MyTimeStamp	受信時間バッファ
戻値	なし	—

RxDataTime 関数は受信時間より前メッセージとのメッセージ間隔を計算します。

14.3. スロット宣言

No	関数名	用途
1	closeEvent	TxData の終了処理関数
2	btnRxStart_Click	受信メッセージ開始関数
3	btnRxStop_Click	受信メッセージ停止関数
4	btnRxDataFind_Click	受信メッセージファイル検索関数
5	chbViewReport_Toggled	受信メッセージ表示関数
6	timRead	受信タイマ関数
7	timDisplay	受信メッセージ表示関数

14.3.1. void closeEvent(QCloseEvent * pEvent)

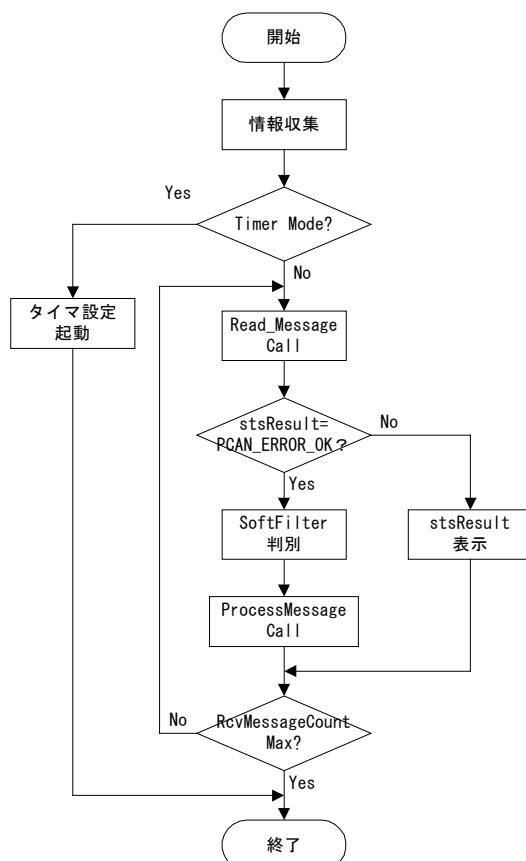
項	変数	説明
引数	QCloseEvent * pEvent	RxData-Widget のクローズイベント
戻値	なし	—

RxData-Widget のクローズイベントが発生した場合、RxData のウィンドウを閉じます。
表示画面フラグの更新を実施します。

14.3.2. void btnRxStart_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

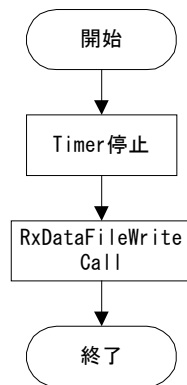
「Start」ボタンが押された場合の処理を行います。
GraphStart 関数と同じ動作になります。



14.3.3. void btnRxStop_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Stop」ボタンが押された場合の処理を行います。
GraphStop 関数と同じ動作になります。



14.3.4. void btnRxDataFind_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Find」ボタンが押された場合の処理を行います。
QFileDialog を開きファイルの設定可能としファイルを指定します。

14.3.5. void chbViewReport_Toggled()

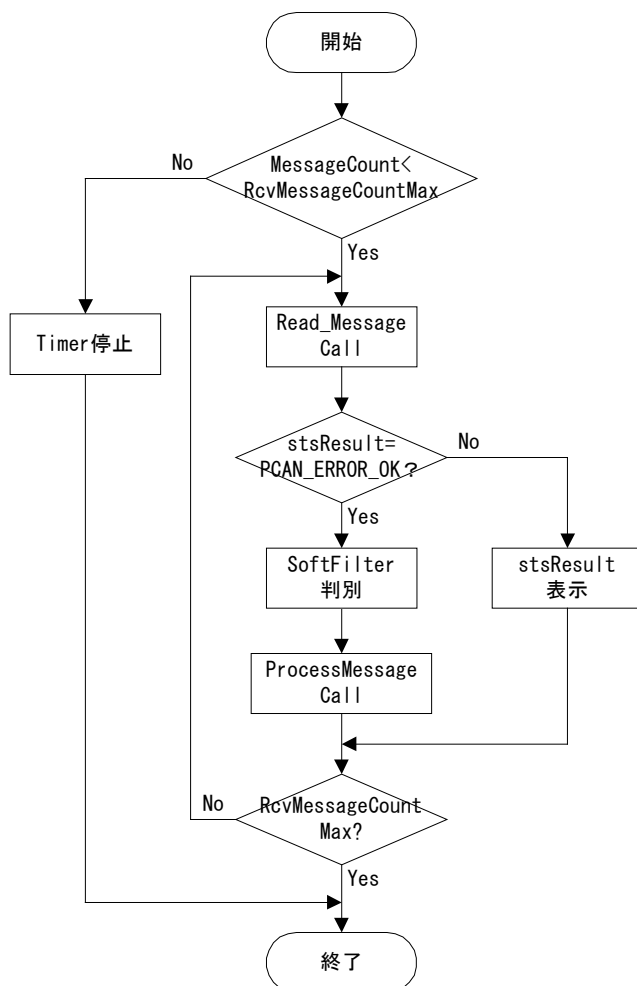
項	変数	説明
引数	なし	—
戻値	なし	—

「Always Displayed」にチェックが入れられた場合の処理を行います。
受信メッセージの表示/非表示の切り替えを行います。

14.3.6. void timRead()

項	変数	説明
引数	なし	—
戻値	なし	—

btnRxStart_Click 関数にて起動されたタイマが設定時間毎に呼び出されます。
受信メッセージをバッファに待避します。



14.3.7. void timDisplay()

項	変数	説明
引数	なし	—
戻値	なし	—

btnRxStart_Click 関数にて起動されたタイマが設定時間毎に呼び出されます。
受信メッセージを表示します。

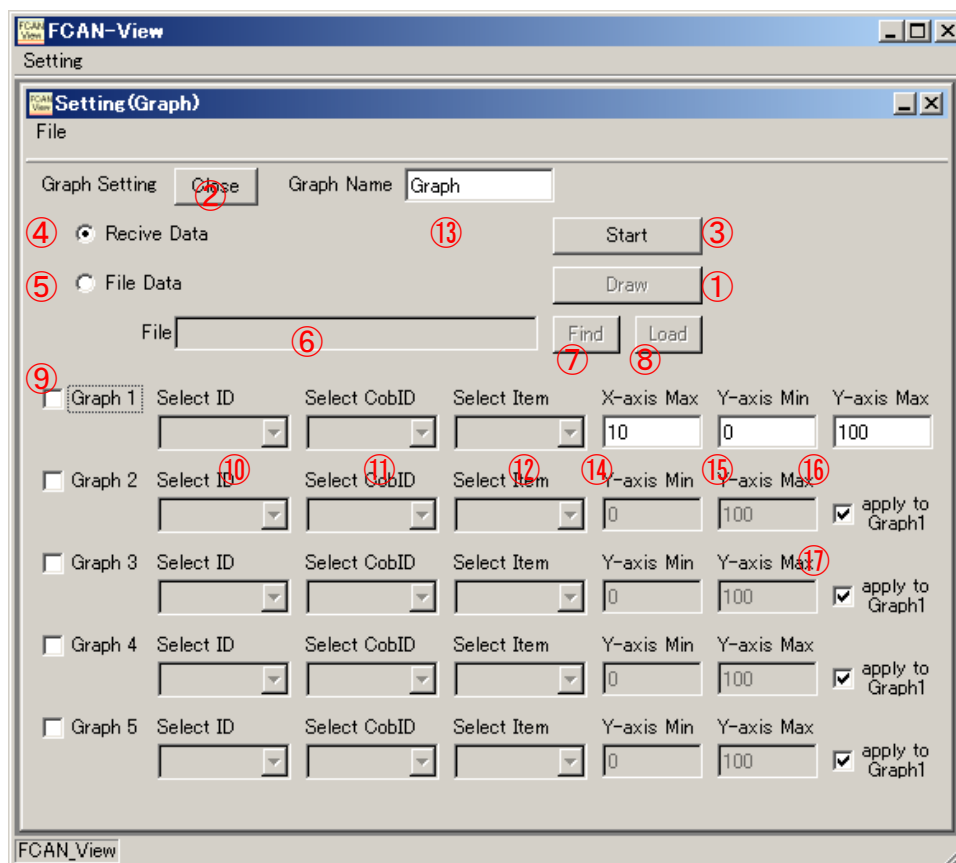
14.4. データメンバー宣言

private 変数

No	変数名	型	用途
1	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
2	* pInformation	Information	Information-Class ポインタ
3	* pHardWare	HardWare	HardWare-Class ポインタ
4	* pSysConfig	SysConfig	SysConfig-Class ポインタ
5	* pRxDataFilter	RxDataFilter	RxDataFilter-Class ポインタ
6	* pCanDataMng	CanDataMng	CanDataMng-Class ポインタ
7	* pCanData	CanData	CanData-Class ポインタ
8	*ReadTimer	QTimer	ReadTimer ポインタ
9	*DisplayTimer	QTimer	DisplayTimer ポインタ
10	MessageCountNow	unsigned long int	バッファ呼び出し現在位置
11	RcvMessageCountMax	unsigned long int	バッファ呼び出し最大値
12	DisplayMessageCountNow	unsigned long int	受信メッセージ表示済み値
13	DefaultTimeMillis	unsigned long int	基本ミリ時間
14	DefaultTimeMicros	unsigned short int	基本マイクロ時間
15	RxDataTimeMillis	unsigned long int	受信ミリ時間
16	RxDataTimeMicros	unsigned short int	受信マイクロ時間
17	TimeGetFlag	int	受信開始フラグ
18	RcvCountMax_str	Qstring	受信カウント終了値

15. Graph-Qwidget

15.1. Qwidget(ウィンドウ)



- ① : 設定値に従いグラフを表示します。
 - ② : グラフ設定画面表示を閉じます。
 - ③ : 受信と同時に表示を選択時、受信メッセージ動作を開始します。
 - ④ : 受信メッセージ動作と同時にグラフ表示を行います。
 - ⑤ : 受信したデータファイルのデータをグラフ表示を行います。
 - ⑥ : 受信メッセージファイル名を指定します。
 - ⑦ : 受信メッセージファイルを検索します。
 - ⑧ : 受信目メッセージファイルをデータバッファに読み込みます。
 - ⑨ : グラフ 1 項目の表示をします。
 - ⑩ : グラフ 1 の ID を選択します。
 - ⑪ : グラフ 1 の COB-ID を選択します。
 - ⑫ : グラフ 1 の Item を選択します。
 - ⑬ : グラフ名を設定します。
 - ⑭ : グラフの X 軸最大値を入力します (Dec 値)。
 - ⑮ : グラフの Y 軸最小値を入力します (Dec 値)。
 - ⑯ : グラフの Y 軸最大値を入力します (Dec 値)。
 - ⑰ : グラフ 1 の Y 軸最小値及び Y 軸最大値をグラフ 2~5 で共用します。
- * ⑨~⑫、⑮~⑰は各グラフ 2~5 にも同様に選択されます。

15.2. メンバー関数

public 関数

No	関数名	用途
1	SysConfigSet	SysConfig-Class ポインタコピー
2	RxDataSet	RxData-Class ポインタコピー
3	RxDataFormatSet	RxDataFormat-Class ポインタコピー
4	PlotterSet	Plotter-Class ポインタコピー
5	PlotSettingsSet	PlotSettings-Class ポインタコピー
6	CanDataSet	CanData-Class ポインタコピー
7	CanDataMngSet	CanDataMng-Class ポインタコピー
8	CanFrameSet	CanFrame-Class ポインタコピー
9	InformationSet	Information-Class ポインタコピー
10	FcanViewSet	FCAN_View-Class ポインタコピー
11	GraphNumberSet	グラフ番号設定関数
12	GraphInit	グラフ設定初期化関数
13	EndScroll	グラフ終了ポイント検出関数

private 関数

No	関数名	用途
1	GetId	グラフ表示 ID 取得関数
2	GetFunc	グラフ表示 Function 取得関数
3	GetItem	グラフ表示 Item 取得関数
4	chbGraph1_Load	フォーマット ID 表示関数
5	ChbGraph2_Load	フォーマット ID 表示関数
6	ChbGraph3_Load	フォーマット ID 表示関数
7	ChbGraph4_Load	フォーマット ID 表示関数
8	ChbGraph5_Load	フォーマット ID 表示関数
9	cbbGraphId1_Load	フォーマット COB-ID 表示関数
10	CbbGraphId2_Load	フォーマット COB-ID 表示関数
11	CbbGraphId3_Load	フォーマット COB-ID 表示関数
12	CbbGraphId4_Load	フォーマット COB-ID 表示関数
13	CbbGraphId5_Load	フォーマット COB-ID 表示関数
14	cbbGraphFunc1_Load	フォーマット ITEM 表示関数
15	CbbGraphFunc2_Load	フォーマット ITEM 表示関数
16	CbbGraphFunc3_Load	フォーマット ITEM 表示関数
17	CbbGraphFunc4_Load	フォーマット ITEM 表示関数
18	CbbGraphFunc5_Load	フォーマット ITEM 表示関数
19	PlotterSetting	グラフ初期設定関数
20	GraphReSize	Graph-Widget 再表示関数
21	createMenus	メニュー生成関数
22	createMenuAction	メニューアクション生成関数
23	ValueCheck	入力文字チェック関数

15.2.1. void FcanViewSet(FCAN_View * fcanview)

項	変数	説明
引数	FCAN_View * fcanview	FCAN_View クラスポインタ
戻値	なし	—

FcanViewSet 関数はFCAN_View-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.2. void InformationSet(Information * information)

項	変数	説明
引数	Information * information	Information クラスポインタ
戻値	なし	—

InformationSet 関数は Information-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.3. void SysConfigSet(SysConfig * sysconfig)

項	変数	説明
引数	SysConfig * sysconfig	SysConfig クラスポインタ
戻値	なし	—

SysConfigSet 関数は SysConfig-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.4. void RxDataSet(RxData * rxdata)

項	変数	説明
引数	RxData * rxdata	RxData クラスポインタ
戻値	なし	—

RxDataSet 関数は RxData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.5. void RxDataFormatSet(RxDataFormat * rxdataformat)

項	変数	説明
引数	RxDataFormat * rxdataformat	RxDataFormat クラスポインタ
戻値	なし	—

RxDataFormatSet 関数は RxDataFormat-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.6. void PlotterSet(Plotter * plotter)

項	変数	説明
引数	Plotter * plotter	Plotter クラスポインタ
戻値	なし	—

PlotterSet 関数は Plotter-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.7. void PlotSettingsSet(PlotSettings * plotsettings)

項	変数	説明
引数	PlotSettings * plotsettings	PlotSettings クラスポインタ
戻値	なし	—

PlotSettingsSet 関数はPlotSettings-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.8. void CanDataSet(CanData * candata)

項	変数	説明
引数	CanData * candata	CanData クラスポインタ
戻値	なし	—

CanDataSet 関数はCanData-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.9. void CanDataMngSet(CanDataMng * candatamng)

項	変数	説明
引数	CanDataMng * candatamng	CanDataMng クラスポインタ
戻値	なし	—

CanDataMngSet 関数はCanDataMng-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.10. void CanFrameSet(CanFrame * canframe)

項	変数	説明
引数	CanFrame * canframe	CanFrame クラスポインタ
戻値	なし	—

CanFrameSet 関数はCanFrame-Class のポインタをコピーしています。
このポインタを利用し、各関数を呼び出します。

15.2.11. void GraphNumberSet(int no)

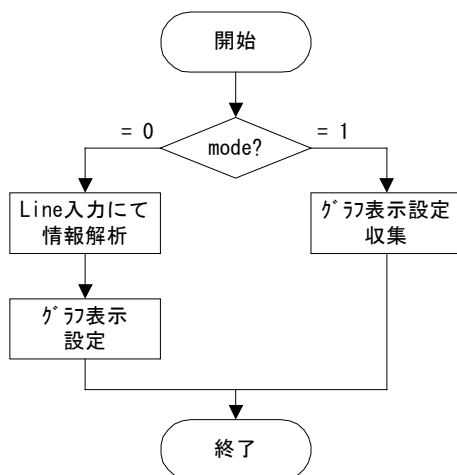
項	変数	説明
引数	int no	グラフ画面番号
戻値	なし	—

GraphNumberSet 関数は指定されたグラフ画面番号の設定を行います。

15.2.12. QString GraphInit(int mode, QString line)

項	変数	説明
引数1	int mode	初期化モード(0: Load 1:Save)
引数2	QString line	InitFile がら読み込んだラインデータ
戻値	QString	Graph 指定情報出力

GraphInit 関数は InitFile-Class より呼びだされた Mode により Graph の設定及び設定情報の出力を行います。



15.2.13. double EndScroll()

項	変数	説明
引数	なし	—
戻値	Doible	グラフ終了位置

現在表示されているグラフのX 軸最終時間を返します。

15.2.14. int GetId(int Select)

項	変数	説明
引数	int Select	選択グラフ番号
戻値	Int	GraphID

現在設定されている「Select ID」の Index 番号を返します。

15.2.15. int GetFunc(int Select)

項	変数	説明
引数	int Select	選択グラフ番号
戻値	Int	GetFunc

現在設定されている「Select CobID」の Index 番号を返します。

15.2.16. int GetItem(int Select)

項	変数	説明
引数	int Select	選択グラフ番号
戻値	Int	GetItem

現在設定されている「Select Item」の Index 番号を返します。

15.2.17. void chbGraph1_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph1 の「Select ID」の内容をフォーマットのデータを収集し更新します。
「Graph1」チェックボタンの変化にて呼び出されます。

15.2.18. void chbGraph2_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph2 の「Select ID」の内容をフォーマットのデータを収集し更新します。
「Graph2」チェックボタンの変化にて呼び出されます。

15.2.19. void chbGraph3_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph3 の「Select ID」の内容をフォーマットのデータを収集し更新します。
「Graph3」チェックボタンの変化にて呼び出されます。

15.2.20. void chbGraph4_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph4 の「Select ID」の内容をフォーマットのデータを収集し更新します。
「Graph4」チェックボタンの変化にて呼び出されます。

15.2.21. void chbGraph5_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph5 の「Select ID」の内容をフォーマットのデータを収集し更新します。
「Graph5」チェックボタンの変化にて呼び出されます。

15.2.22. void cbbGraphId1_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph1 の「Select CobID」の内容をフォーマットのデータを収集し更新します。

Graph1 の「Select ID」選択の変化にて呼び出されます。

15.2.23. void cbbGraphId2_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph2 の「Select CobID」の内容をフォーマットのデータを収集し更新します。

Graph2 の「Select ID」選択の変化にて呼び出されます。

15.2.24. void cbbGraphId3_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph3 の「Select CobID」の内容をフォーマットのデータを収集し更新します。

Graph3 の「Select ID」選択の変化にて呼び出されます。

15.2.25. void cbbGraphId4_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph4 の「Select CobID」の内容をフォーマットのデータを収集し更新します。

Graph4 の「Select ID」選択の変化にて呼び出されます。

15.2.26. void cbbGraphId5_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph5 の「Select CobID」の内容をフォーマットのデータを収集し更新します。

Graph5 の「Select ID」選択の変化にて呼び出されます。

15.2.27. void cbbGraphFunc1_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph1 の「Select Item」の内容をフォーマットのデータを収集し更新します。

Graph1 の「Select CobID」選択の変化にて呼び出されます。

15.2.28. void cbbGraphFunc2_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph2 の「Select Item」の内容をフォーマットのデータを収集し更新します。

Graph2 の「Select CobID」選択の変化にて呼び出されます。

15.2.29. void cbbGraphFunc3_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph3 の「Select Item」の内容をフォーマットのデータを収集し更新します。

Graph3 の「Select CobID」選択の変化にて呼び出されます。

15.2.30. void cbbGraphFunc4_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph4 の「Select Item」の内容をフォーマットのデータを収集し更新します。

Graph4 の「Select CobID」選択の変化にて呼び出されます。

15.2.31. void cbbGraphFunc5_Load()

項	変数	説明
引数	なし	—
戻値	なし	—

Graph5 の「Select Item」の内容をフォーマットのデータを収集し更新します。

Graph5 の「Select CobID」選択の変化にて呼び出されます。

15.2.32. void PlotterSetting()

項	変数	説明
引数	なし	—
戻値	なし	—

グラフの初期化を行います。

15.2.33. void GraphReSize()

項	変数	説明
引数	なし	—
戻値	なし	—

グラフ Widget の再表示を行います。

15.2.34. void createMenus()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenus 関数はメニューの生成を行います。

15.2.35. void createMenuActions()

項	変数	説明
引数	なし	—
戻値	なし	—

createMenuActions 関数はメニューの各アクション生成を行います。

15.2.36. int ValueCheck(QString ValueString)

項	変数	説明
引数	QString ValueString	入力文字列
戻値	int	エラーフラグ

ValueChec 関数は入力文字列が 16 進値であるかを判断します。

15.3. スロット宣言

No	関数名	用途
1	closeEvent	Graph の終了処理関数
2	btnGraphDraw_Click	グラフ描画関数
3	btnGraphClose_Click	グラフ設定画面消去関数
4	btnGraphStartStop_Click	受信メッセージ開始/停止関数
5	btnGraphFileFind_Click	グラフ表示用受信メッセージファイル検索関数
6	btnGraphFileLoad_Click	グラフ表示用受信メッセージファイルロード関数
7	rdbGraphRecive_Toggled	グラフ表示データ選択関数
8	chbGraph1_Toggled	グラフ 1 表示/非表示切替え関数
9	chbGraph2_Toggled	グラフ 2 表示/非表示切替え関数
10	chbGraph3_Toggled	グラフ 3 表示/非表示切替え関数
11	chbGraph4_Toggled	グラフ 4 表示/非表示切替え関数
12	chbGraph5_Toggled	グラフ 5 表示/非表示切替え関数
13	cbbGraphId1_Changed	グラフ 1 ID 切替え関数
14	cbbGraphId2_Changed	グラフ 2 ID 切替え関数
15	cbbGraphId3_Changed	グラフ 3 ID 切替え関数
16	cbbGraphId4_Changed	グラフ 4 ID 切替え関数
17	cbbGraphId5_Changed	グラフ 5 ID 切替え関数
18	cbbGraphFunc1_Changed	グラフ 1 CobID 切替え関数
19	cbbGraphFunc2_Changed	グラフ 2 CobID 切替え関数
20	cbbGraphFunc3_Changed	グラフ 3 CobID 切替え関数
21	cbbGraphFunc4_Changed	グラフ 4 CobID 切替え関数
22	cbbGraphFunc5_Changed	グラフ 5 CobID 切替え関数
23	chbYaxis2_Toggled	グラフ 2 Y 軸切替え関数
24	ChbYaxis3_Toggled	グラフ 3 Y 軸切替え関数
25	ChbYaxis4_Toggled	グラフ 4 Y 軸切替え関数
26	ChbYaxis5_Toggled	グラフ 5 Y 軸切替え関数
27	timGraph	グラフ表示間隔タイマ関数
28	menu_triggered	メニュー選択関数
29	menu_loadfile	初期化ファイルロード関数
30	menu_savefile	初期化ファイルセーブ関数

15.3.1. void closeEvent(QCloseEvent * pEvent)

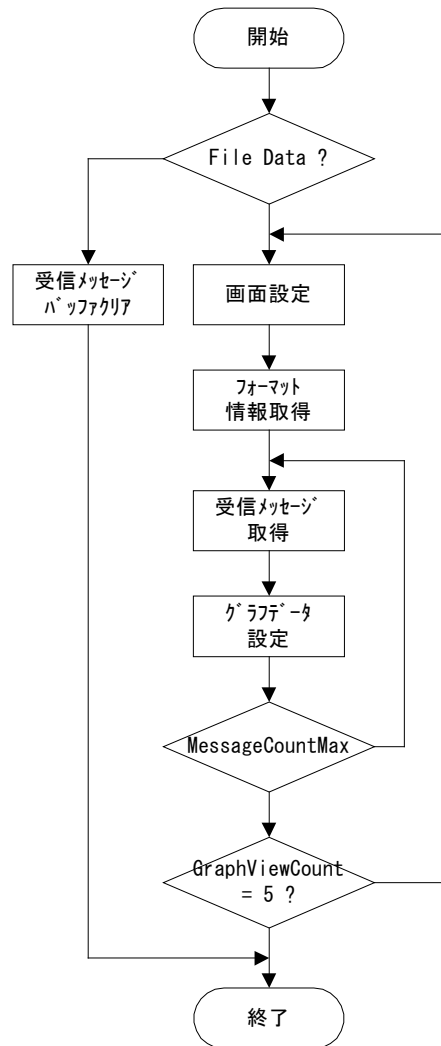
項	変数	説明
引数	QCloseEvent * pEvent	Graph-Widget のクローズイベント
戻値	なし	—

Graph-Widget のクローズイベントが発生した場合、Graph のウィンドウを閉じます。
表示画面フラグの更新を実施します。

15.3.2. void btnGraphDraw_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

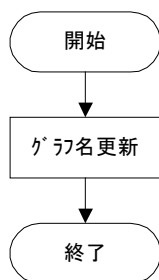
「Draw」ボタンが押された場合の処理を行います。
グラフ描画を行います。



15.3.3. void btnGraphClose_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

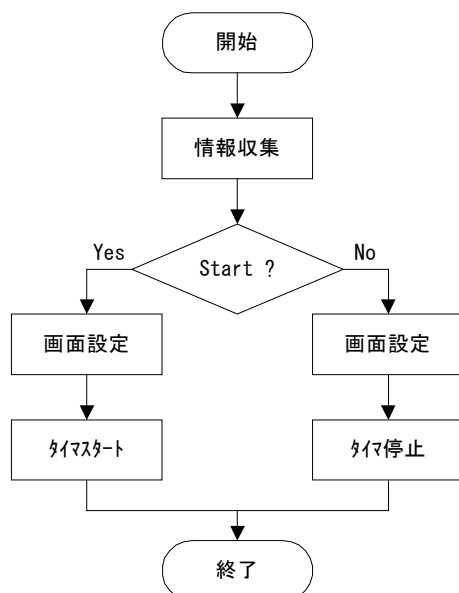
「Close」ボタンが押された場合の処理を行います。
表示されたグラフ設定画面を閉じます。



15.3.4. void btnGraphStartStop_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Start」又は「Stop」ボタンが押された場合の処理を行います。
「Start」時は、タイマをスタートし1秒毎に受信メッセージを確認します。
「Stop」時は、タイマをストップし停止処理を実施します。



15.3.5. void btnGraphFileFind_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

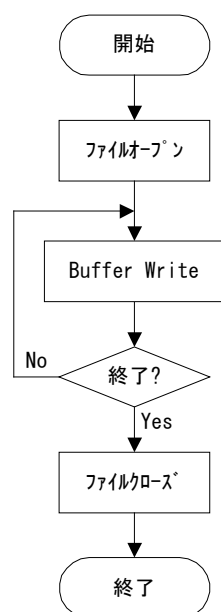
「Find」ボタンが押された場合の処理を行います。

QFileDialog を開きファイルの設定可能としファイルを指定します。

15.3.6. void btnGraphFileLoad_Click()

項	変数	説明
引数	なし	—
戻値	なし	—

「Load」ボタンが押された場合の処理を行います。



15.3.7. void rdbGraphRecive_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Recive Data」又は「File Data」の切替えが行われた場合の画面表示を変更します。

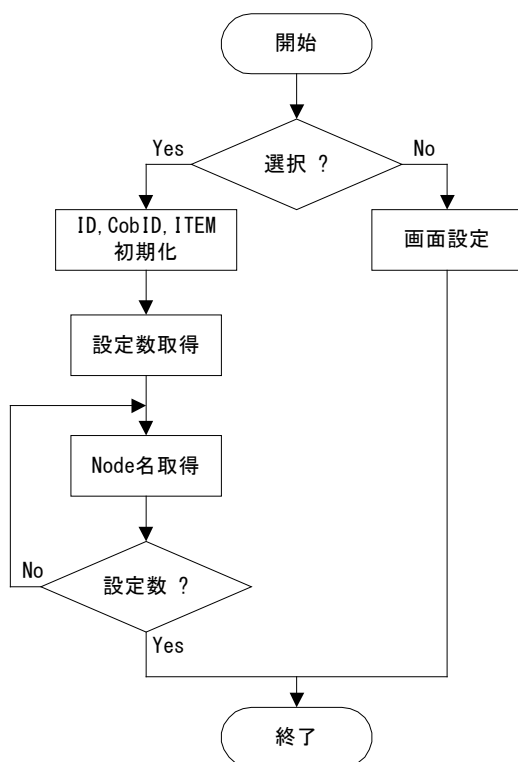
15.3.8. void chbGraph1_Toggled()

項	変数	説明
引数	const QString &s	表示 ID
戻値	なし	—

「Graph1」が選択/解除された場合、画面表示を変更します。

「Graph1」が選択 Select ID の選択が可能になるように現在フォーマットバッファに格納されているデータより全ての ID を検索し表示可能にします。

「Graph2」, 「Graph3」, 「Graph4」, 「Graph5」の動作も同様です。



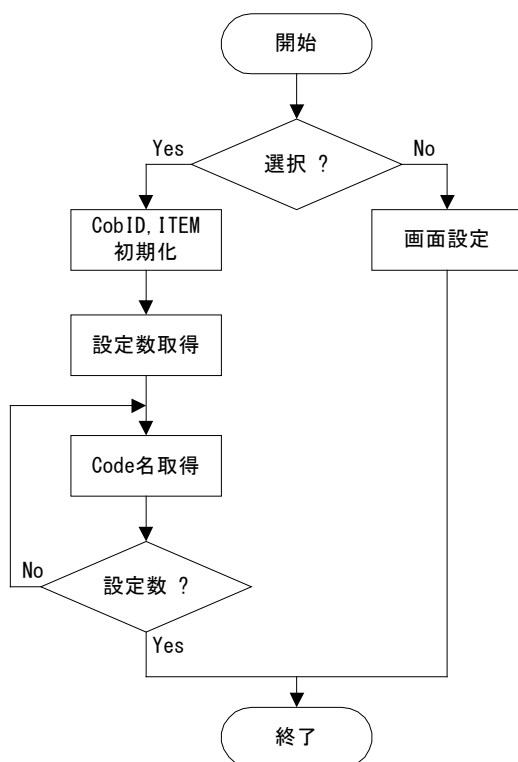
15.3.9. void cbbGraphId1_Changed(const QString &s)

項	変数	説明
引数	const QString &s	表示 Id
戻値	なし	—

「Graph1」の「Select ID」が変更/選択された場合の処理を行います。

「Select ConID」が選択が可能になるように現在フォーマットバッファに格納されているデータより該当する ID の全ての CobID を検索し表示可能にします。

「Graph2」, 「Graph3」, 「Graph4」, 「Graph5」の動作も同様です。



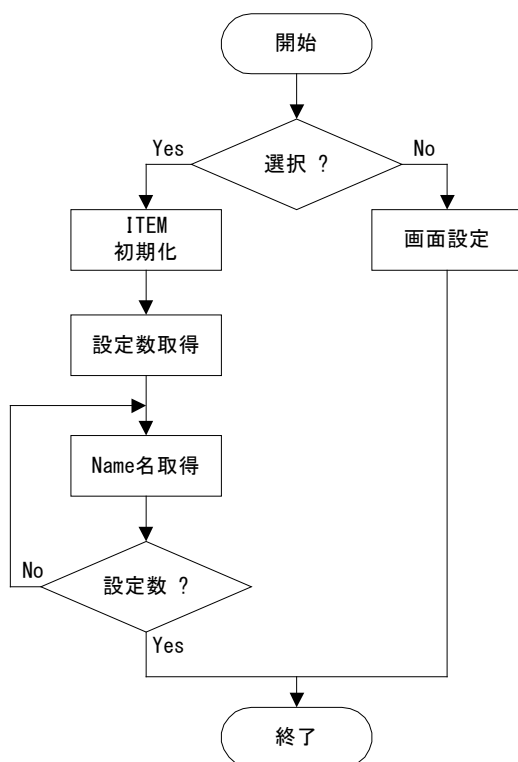
15.3.10. void cbbGraphFunc1_Changed(const QString &s)

項	変数	説明
引数	const QString &s	表示 Func
戻値	なし	—

「Graph1」の「Select CobID」が変更/選択された場合の処理を行います。

「Select Item」が選択が可能になるように現在フォーマットバッファに格納されているデータより該当する ID・Func の全ての Item を検索し表示可能にします。

「Graph2」, 「Graph3」, 「Graph4」, 「Graph5」の動作も同様です。



15.3.11. void chbYaxis2_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Graph2」の Y 軸最大値の有効/無効の切り替え処理を行います。

15.3.12. void chbYaxis3_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Graph3」の Y 軸最大値の有効/無効の切り替え処理を行います。

15.3.13. void chbYaxis4_Toggled()

項	変数	説明
引数	なし	—
戻値	なし	—

「Graph4」の Y 軸最大値の有効/無効の切り替え処理を行います。

15.3.14. void chbYaxis5_Toggled()

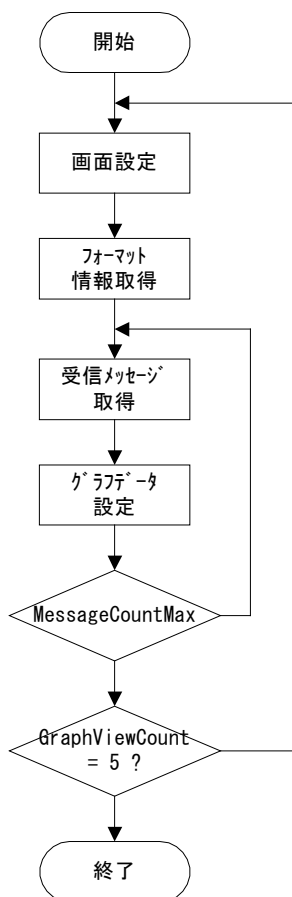
項	変数	説明
引数	なし	—
戻値	なし	—

「Graph5」の Y 軸最大値の有効/無効の切り替え処理を行います。

15.3.15. void timGraph()

項	変数	説明
引数	なし	—
戻値	なし	—

btnGraphStartStop_Click 関数にて起動されたタイマが設定時間毎に呼び出されます。
受信メッセージをバッファから取得しグラフに表示します。



15.3.16. void menu_triggered()

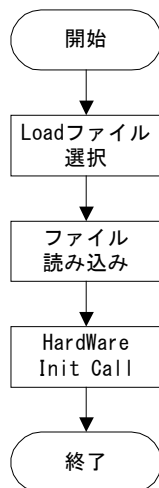
項	変数	説明
引数	なし	—
戻値	なし	—

メニューが選択された時、再メニューアクションを生成します。

15.3.17. void menu_loadfile()

項	変数	説明
引数	なし	—
戻値	なし	—

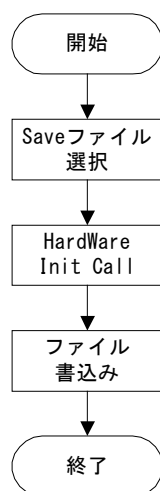
メニュー内「Load File」が選択された時、指定されたファイルデータをロードし設定します。



15.3.18. void menu_savefile()

項	変数	説明
引数	なし	—
戻値	なし	—

メニュー内「Save File」が選択された時、設定情報を指定されたファイルにセーブします。



15.4. データメンバー宣言

public 変数

No	変数名	型	用途
1	GraphNumber	int	Graph 番号
2	GraphWindowOpen	int	Graph ウィンドウオープンフラグ
3	GraphYaxisSelect	int [5]	Graph-Yaxis 選択
4	GraphName	QString	Graph 名称
5	GraphWindow	bool [5]	Graph 表示数
6	pGraphItem	CanFrameFormat	Graph-Item ポインタ

private 変数

No	変数名	型	用途
1	* pSysConfig	SysConfig	SysConfig-Class ポインタ
2	* pRxDataFormat	RxDataFormat	RxDataFormat-Class ポインタ
3	* pPlotter	Plotter	Plotter-Class ポインタ
4	* pPlotSettings	PlotSettings	PlotSettings-Class ポインタ
5	* pCanData	CanData	CanData-Class ポインタ
6	* pCanDataMng	CanDataMng	CanDataMng-Class ポインタ
7	* pCanFrame	CanFrame	CanFrame-Class ポインタ
8	* pInformation	Information	Information-Class ポインタ
9	* pFcanView	FCAN_View	FCAN_View-Class ポインタ
10	*GraphTimer	QTimer	GraphTimer ポインタ
11	MessageCount	int	受信メッセージカウンタ
12	MessageTimeGen	long long	グラフ表示基本時間軸値
13	TimeGetFlag	int	グラフ表示基本時間取得フラグ
14	GraphRighViewCount	int	グラフ表示開始時間
15	LoardFileError	int	受信メッセージ読み込みエラー
16	IntAndFloat	int ival unsigned int uival float fval	データ変換
17	LongIntAndFloat	Long long ival unsigned long long uival double fval	データ変換

16. CanData-Class

16.1. メンバー関数

public 関数

No	関数名	用途
1	なし	—

16.2. スロット宣言

No	関数名	用途
1	なし	—

16.3. データメンバー宣言

private 変数

No	変数名	型	用途
1	type	int	データタイプ
2	id	int	ID 番号バッファ
3	func	int	FUNC 番号バッファ
4	len	int	LEN バッファ
5	data	unsigned char [CAN_FRAME_LEN_MAX]	Data バッファ
6	mtime	unsigned long int	ms 時間バッファ
7	utime	unsigned short int	us 時間バッファ

CAN_FRAME_LEN_MAX は現在 8 に設定されています。

17. CanDataMng-Class

17.1. メンバー関数

public 関数

No	関数名	用途
1	num	CanData-Class 数関数
2	set	CanData-Class 設定関数
3	get	CanData-Class 取得関数
4	isFull	CanData-ClassFULL 関数
5	clear	CanData-Class 削除関数

17.1.1. int num(void)

項	変数	説明
引数	なし	—
戻値	int	設定数

num 関数は CanData-Class の設定数を返します。

17.1.2. void set(CanData &canData)

項	変数	説明
引数	CanData &canData	CanData-Class ポインタ
戻値	なし	—

set 関数は CanData-Class を追加します。

17.1.3. int get(int no, CanData &canData)

項	変数	説明
引数 1	int no	CanData-Class 番号
引数 2	CanData &canData	CanData-Class ポインタ
戻値	int	ステータス

get 関数は指定された CanData-Class 番号のポインタを設定します。

17.1.4. bool isFull(void)

項	変数	説明
引数	なし	—
戻値	Bool	ステータス

isFull 関数は CanData-Class の設定容量が指定されたバッファ容量と比較し状態を返します。

17.1.5. void clear(void)

項	変数	説明
引数	なし	—
戻値	なし	—

clear 関数は全ての CanData-Class 設定を削除します。

17.2. スロット宣言

No	関数名	用途
1	なし	—

17.3. データメンバー宣言

private 変数

No	変数名	型	用途
1	lock	QReadWriteLock	ロック
2	recodeNum	volatile int	CanData-Class 数
3	dataVec	QVector<CanData *>	CanData-Class バッファ

現在「dataVec」の最大値 Vector 数は 1,000,000 に設定されています。

18. CanFrame-Class

18.1. メンバー関数

public 関数

No	関数名	用途
1	nodeAdd	ノード追加関数
2	nodeSearch	ノード検索関数
3	functionSearch	ファンクション検索追加関数
4	nodeNum	ノード数関数
5	nodeID	ノード ID 関数
6	nodeName	ノード名称関数
7	nodeDelete	ノード削除関数 (Index 基準)
8	nodeDeleteId	ノード削除関数 (ID 基準)

18.1.1. CanFrameNode * nodeAdd(int nodeID, QString name)

項	変数	説明
引数 1	int nodeID	ノード ID
引数 2	QString name	ノード名称
戻値	CanFrameNode *	CanFrameNode ポインタ

nodeAdd 関数は指定されたノード ID , ノード名称のノードを作成しそのポインタを返します。

18.1.2. CanFrameNode * nodeSearch(int nodeID)

項	変数	説明
引数	int nodeID	ノード ID
戻値	CanFrameNode *	CanFrameNode ポインタ

nodeSearch 関数は指定されたノード ID を検索し、そのポインタを返します。

該当 ID が無い場合、NULL を返します。

18.1.3. CanFrameFunction * functionSearch(int nodeID, int functionCode)

項	変数	説明
引数 1	int nodeID	ノード ID
引数 2	int functionCode	ファンクション CODE
戻値	CanFrameFunction *	CanFrameFunction ポインタ

functionSearch 関数は指定されたノード ID , ファンクション CODE を検索し、そのポインタを返します。

該当 ID, CODE が無い場合、NULL を返します

18.1.4. int nodeNum(void)

項	変数	説明
引数	なし	—
戻値	int	CanFrameNode 数

nodeNum 関数は設定されている CanFrameNode 数を返します。

18.1.5. int nodeID(int index)

項	変数	説明
引数	int index	インデックス
戻値	int	ノード ID

nodeID 関数はベクタのインデックスからノード ID を返します。

18.1.6. void nodeName(int index, QString &)

項	変数	説明
引数 1	int index	インデックス
引数 2	QString &	ノード名称ポインタ
戻値	なし	—

nodeName 関数はベクタのインデックスからノード名称を返します。

18.1.7. void nodeDelete(int index)

項	変数	説明
引数 1	int index	インデックス
戻値	なし	—

nodeDelete 関数はベクタのインデックスからノードを削除します。

18.1.8. void nodeDeleteId(int id)

項	変数	説明
引数 1	int index	インデックス
戻値	なし	—

nodeDelete 関数はベクタの ID からノードを削除します。

18.2. スロット宣言

No	関数名	用途
1	なし	—

18.3. データメンバー宣言

private 変数

No	変数名	型	用途
1	nodeVec	QVector<CanFrameNode *>	CanFrameNode バッファ

19. CanFrameNode-Class

19.1. メンバー関数

public 関数

No	関数名	用途
1	functionAdd	ファンクション追加関数
2	functionSearch	ファンクション検索関数
3	functionNum	ファンクション数関数
4	functionCode	ファンクション Code 取得関数
5	functionName	ファンクション名称取得関数
6	functionDelete	ファンクション削除関数 (index 基準)
7	functionDeleteFunc	ファンクション削除関数 (code 基準)

19.1.1. CanFrameFunction * functionAdd(int functionCode, QString functionName)

項	変数	説明
引数 1	int functionCode	ファンクション CODE
引数 2	QString functionName	ファンクション名称
戻値	CanFrameFunction *	CanFrameFunction ポインタ

functionAdd 関数は指定されたファンクション CODE , ファンクション名称のファンクションを作成しそのポインタを返します。

19.1.2. CanFrameFunction * functionSearch(int functionCode)

項	変数	説明
引数	int functionCode	ファンクション CODE
戻値	CanFrameFunction *	CanFrameFunction ポインタ

functionSearch 関数は指定されたファンクション CODE を検索し、そのポインタを返します。
該当 CODE が無い場合、NULL を返します。

19.1.3. int functionNum(void)

項	変数	説明
引数	なし	—
戻値	int	CanFrameFunction 数

nfunctionNum 関数は設定されている CanFrameFunction 数を返します。

19.1.4. int functionCode(int index)

項	変数	説明
引数	int index	インデックス
戻値	int	ファンクション CODE

functionCode 関数はベクタのインデックスからファンクション CODE を返します。

19.1.5. void functionName(int index, QString &)

項	変数	説明
引数1	int index	インデックス
引数2	QString &	ファンクション名称ポインタ
戻値	なし	—

functionName 関数はベクタのインデックスからファンクション名称を返します。

19.1.6. void functionDelete(int index)

項	変数	説明
引数1	int index	インデックス
戻値	なし	—

functionDelete 関数はベクタのインデックスからファンクションを削除します。

19.1.7. void functionDeleteFunc(int code)

項	変数	説明
引数1	int index	インデックス
戻値	なし	—

functionDelete 関数はベクタの CODE からファンクションを削除します。

19.2. スロット宣言

No	関数名	用途
1	なし	—

19.3. データメンバー宣言

public 変数

No	変数名	型	用途
1	id	int	CanFrameNodeID
2	name	QString	CanFrameNode 名称

private 変数

No	変数名	型	用途
1	functionVec	QVector<CanFrameFunction *>	CanFrameFunction バッファ

20. CanFrameFunction-Class

20.1. メンバー関数

public 関数

No	関数名	用途
1	formatAdd	フォーマット追加関数
2	formatNum	フォーマット数関数
3	formatGet	フォーマット取得関数
4	formatName	フォーマット名称取得関数
5	formatNo	フォーマット番号取得関数
6	FormatDeleteIndex	フォーマット削除関数(index 基準)
7	formatDeleteItem	フォーマット削除関数(no 基準)
8	code	フォーマットコード取得関数

20.1.1. int formatAdd(CanFrameFormat * pFormat)

項	変数	説明
引数	CanFrameFormat * pFormat	CanFrameFormat ポインタ
戻値	int	ステータス

formatAdd 関数は指定された CanFrameFormat を追加しそのステータスを返します。

20.1.2. int formatNum(void)

項	変数	説明
引数	なし	—
戻値	int	CanFrameFormat 数

formatNum 関数は設定されている CanFrameFormat 数を返します。

20.1.3. CanFrameFormat * formatGet(int no)

項	変数	説明
引数	int no	フォーマット番号
戻値	CanFrameFormat *	CanFrameFormat ポインタ

formatGet 関数は設定されている番号から CanFrameFormat を検索し該当する CanFrameFormat ポインタを返します。該当番号が無い場合 NULL を返します。

20.1.4. void formatName(int no, QString &)

項	変数	説明
引数1	int no	フォーマット番号
引数2	QString &	フォーマット名称ポインタ
戻値	なし	—

formatName 関数は CanFrameFormat 番号からフォーマット名称を返します。

20.1.5. int formatNo(int index)

項	変数	説明
引数	int index	インデックス
戻値	int	フォーマット番号

formatNo 関数はベクタのインデックスからフォーマット番号を返します。

20.1.6. void formatDeleteIndex(int index)

項	変数	説明
引数	int index	インデックス
戻値	なし	—

formatDelete 関数はベクタのインデックスからフォーマットを削除返します。

20.1.7. void formatDeleteItem(int no)

項	変数	説明
引数	int index	インデックス
戻値	なし	—

formatDelete 関数はベクタの NO からフォーマットを削除返します。

20.1.8. int code(void)

項	変数	説明
引数	なし	—
戻値	int	フォーマットコード

code 関数はフォーマットコードを返します。

20.2. スロット宣言

No	関数名	用途
1	なし	—

20.3. データメンバー宣言

public 変数

No	変数名	型	用途
1	m_name	QString	CanFrameFormat 名称

private 変数

No	変数名	型	用途
1	formatVec	QVector<CanFrameFormat *>	CanFrameFormat バッファ

21. CanFrameFormat-Class

21.1. メンバー関数

public 関数

No	関数名	用途
1	typeGet	フォーマットタイプ取得関数
2	denominationGet	フォーマット単位取得関数

private 関数

No	関数名	用途
1	CanFrameFormatSub	フォーマット設定サブ関数

21.1.1. CanFrameFormatType typeGet(void)

項	変数	説明
引数	なし	—
戻値	CanFrameFormatType	フォーマットタイプ

typeGet 関数は CanFrameFormat タイプを返します。

21.1.2. int nameGet(QString & value)

項	変数	説明
引数	QString & value	フォーマット名称ポインタ
戻値	int	ステータス

nameGet 関数はフォーマット名称ポインタを設定します。

21.1.3. int denominationGet(QString & value)

項	変数	説明
引数	QString & value	フォーマット名称ポインタ
戻値	int	ステータス

denominationGet 関数はフォーマット単位ポインタを設定します。

21.2. スロット宣言

No	関数名	用途
1	なし	—

21.3. データメンバー宣言

public 変数

No	変数名	型	用途
1	type	CanFrameFormatType	フォーマットタイプ
2	no	int	フォーマット番号
3	name	QString	フォーマット名称
4	bitOffset	int	フォーマットビットオフセット
5	bitLen	int	フォーマットビット長
6	denomination	QString	フォーマット単位
7	f	float offset float amplification	フォーマットデータタイプ
8	I	long long offset float amplification	フォーマットデータタイプ
9	u	long long offset float amplification	フォーマットデータタイプ