



# CANモニタ—FLCM1 説明書

2025年2月11日 Rev.1.0

## はじめに

この度はFundyLabのCANモニターをご購入いただき、誠にありがとうございます。

本製品は、CAN(Controllor Area Network)通信を監視・解析するためにCAN 2.0Bに対応して設計されており、ロボットコンテスト等のロボットや車載ネットワーク及び産業用ネットワーク等、さまざまな用途での利用が可能です。本マニュアルは、製品の機能や操作方法、接続方法等を分かりやすく解説することを目的としています。

## 主な機能

1. CAN通信のリアルタイム監視  
CANバス上の通信データをリアルタイムでモニタリングし、データフレームを解析・表示します。モニタ機能のみのため、CANバス上の通信データに影響を与えません。またCAN Vbus電圧の簡易表示機能が付いています。
2. フィルタリングおよび絞り込み  
特定のCANメッセージIDやデータパターンに基づいたフィルタリング機能を搭載。必要なデータのみを効率的に抽出します。
3. エラー検出  
特定のエラーを検出しエラーコードとLEDで通知します。
4. データ出力機能  
フィルタリング後のデータを、USB-シリアル経由のPC出力又はAUXコネクタのSPI出力で送信します。  
コンペアアウト機能でデータフレームのデータを閾値と比較した結果を、AUXコネクタより出力します。
5. 設定の保存と復元  
ユーザー設定を内部メモリに保存し、再起動後も設定が保持されます。フィルタリング設定を複数保存できます。
6. ユーザーインターフェース  
2.8inchタッチパネル付きLCD及び操作ボタンにより、直感的な操作が可能です。DSUB-9オスコネクタに接続された電源電圧の簡易表示機能もついています。
7. 各種入出力端子  
TypeC USBコネクタ、DSUB-9オスコネクタ、CANのピン配置切替ヘッダ、AUXコネクタ、電源切り替えスイッチ
8. ファームウェア  
公開されている内蔵ArduinoスケッチでCANリモコンやゲーム機等お好み開発が可能

## 安全上の注意事項

1. 本機は実験研究用の開発キットです。本機を使用したことにより生じたいかなる事象も保証いたしません。
2. 電源及び接続  
指定された電源電圧範囲内及び正しい接続でご使用ください。意図した／しないにかかわらず誤った状態で使用した場合、本機又は接続された機器が誤動作・故障・発煙・発火の恐れがあります。万が一そのような状態が生じた場合は、速やかに電源やコネクタを外してください。  
接続時は必ず電源を切った状態で行ってください。
3. 取り扱い  
落下、衝撃、静電気などに十分ご注意ください。水分や埃の多い場所での使用は避けてください。
4. 周辺機器との接続  
他の電子機器との接続時に誤配線が生じないように、配線図を必ずご確認ください。USBやシリアルケーブル等の取り扱いに注意し、無理な力を加えないでください。
5. プログラムの更新  
ソースコードに手を加える場合、初期不良による保証を含めたあらゆる保証の対象外となります。動作確認を十分に行った上で利用してください。不明な点がある場合は、必ずサポート窓口までお問い合わせください。



## 各部の名称



## 接続方法

1. DSUB-9ピンオスコネクタに接続するケーブルの種類を確認します。  
CAN標準ケーブル・OBD-IIケーブルのどちらでも接続可能です。コネクタを接続する前に、ピン配置切替ヘッダを手で正しい側に差し替えます。上側に挿すとOBD-II用ピン配置、下側に挿すとCAN標準ピン配置となります。
2. Terminate(終端)ON/OFFヘッダを切り替えます。  
ヘッダを上側に挿すとOFF、下側に挿すとONになります。終端抵抗がCANバスの両端機器

で既に接続されている場合は、基本的に本機の終端は不要のため、ヘッダを手でOFF側に差し替えます。終端抵抗が必要な場合のみON側に差し替えてください。

### 3. 電源を接続します。

DSUB-9pinオスコネクタの9pinから7V以上を給電する方法と、USB Type-Cコネクタから5Vを給電する方法があります。

DSUB-9pinオスコネクタから給電する場合、電源の一極(GND)のピン配置はピン配置切り替えヘッダで決定されます。ヘッダを切り替えた後、“仕様—CANバス接続”項の表における「Signal GND」ピンが電源の一極(GND)となります。なお電源の+極はヘッダの切り替えにかかわらず9ピンとなります。必ず±の極性を確かめてください。

## 基本操作方法

1. 電源SWはDSUB-9pin又はUSB-Cのどちらから給電するかを切り替える切り替えスイッチとなっていますので、スライドしてONします。
2. 電源をONにすると起動画面及びメロディーが流れ、その後画面の上部に水色のステータスラインが表示されたモニタモードになります。
3. Cボタンを2回押すと設定モードに入ります。U、D、E、Cボタンで操作していきます。Uボタンは上に、Dボタンは下にカーソルを移動できます。Eボタンで決定、Cボタンでキャンセルします。設定モード時は画面のタッチでも選択できます。
4. TOP MENUでCAN speed画面に入り、CANバスの通信速度(ボーレート)を選択します。ボーレートの設定をCAN通信速度と一致させないと、モニタエリアに戻ってもCANメッセージは表示されません。
5. TOP MENUでHardWareFilter画面に入り、CANメッセージID(以下CAN ID又はIDと省略)フィルタリングの設定を必要に応じて行います。CANバスに流れているメッセージの量が多い場合はメッセージが流れてしまい読めないため、本機能で必要なCAN IDだけ表示されるように設定します。
6. さらに高度なフィルタリング、例えばCANメッセージ中のデータbytesにおける切り出しが必要な場合は、必要に応じてSoftWareFilterの設定を行います。また閾値と比較してHigh/Lowのハードウェア出力ができるCompareOutの設定を必要に応じて行います。
7. 設定後Cボタンか画面のBACKをタップしていくと、設定がマイコンのFlash領域に保存されると同時にモニタモードに戻り、設定に応じてステータスラインにフィルタ設定が表示され、CANメッセージがスクロール表示されます。
8. モニタモード中にEボタンを押すとスクロールが一時停止できます。もう一度押すと解除されます。
9. USB-Cからはフィルタリング後のCANメッセージやエラーコードが出力されています。PCIに繋いでArduino等のシリアルモニターで確認できます。またAUXコネクタのSPI(ホスト/マスタ)からも同様の短縮メッセージが出力されています。

## CANについて及び用語の定義

- CAN: Controller Area Networkの略。Robert Bosch社によって発表され、国際規格ISO11898となり、車載ネットワークプロトコル(通信規格)や産業機器向けに普及してきました。
- CANメッセージ: 本機ではCANバスに流れる情報を指します。CANメッセージには、データフレーム、リモートフレーム、オーバーロードフレーム、エラーフレームの4つのフレームがあります。本機ではデータフレームのみ扱います。特にCANのデータフレーム自体、又はデータフレーム中のIDとデータフィールドのみを合わせたものを指します。
- マルチ・マスター方式: CANでは、2本+GNDのCANバスに並列に機器を接続でき、複数の機器から別々にバスに情報(メッセージ)を送信できます。そのため、データの衝突(Collision)が発生します。その場合、どの情報を優先するかはIDによって決定されます。
- ID: Identifier(識別子)の略。CANのデータフレーム中にあり、重要なデータに高い優先順位を設定するために、IDの値が小さいメッセージがバス中で優先されます。
- データフィールド: CANメッセージで実際に送信したい値の部分。本機はCAN 2.0Bのためデータフレーム長は0~8バイトまでとなります。一方CAN FD規格ではデータ長が最大64バイトまでですが、本機では対応しておりません。

## 機能詳細(モニタモード)

モニタモード時は以下の表示となり、画面上部の青い帯がステータスライン、その下がモニタエリアです。ステータスラインには固定表示、Vbus電圧表示、各種ステータスアイコンが表示されます。モニタエリアには、設定でフィルタリングされたCANメッセージがスクロール表示されます。

```

CANID: Message0-7 0.00 Hz
00000214 Msg: 3e ea d4 9
00000206 Msg: fd
00000206 Msg: 1e 10 5d
00000201 Msg:
00000251 Msg: 8d 75 23 ba 90
0000019c Msg: 9d
0000033d Msg:
0000010e Msg: f9
00000105 Msg: f9
  
```

スクロール表示中のボタンの機能を以下に示します。

Eボタン	スクロール一時停止・解除
Cボタン	スクロール一時停止・設定モード移行
Uボタン	フィルタリング表示モード変更 押すたびにハードウェアフィルタリング表示HD・ソフトウェアフィルタリング表示SDをON/OFFします

## ステータスライン

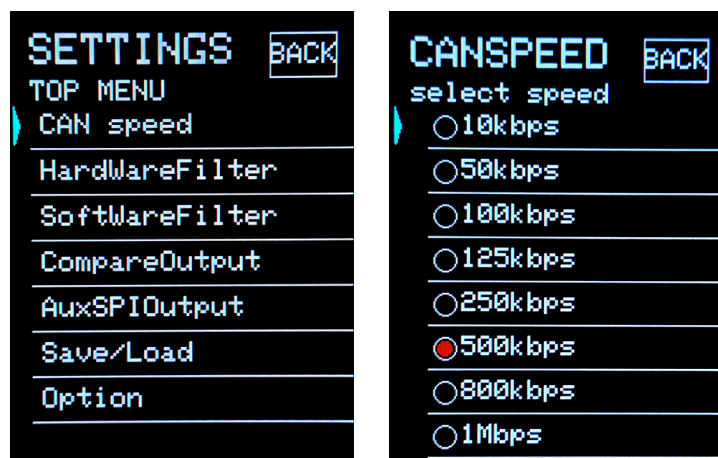
ステータスラインには"CANID: ~"等の固定表示、DSUB-9pinオスコネクタに印加されたCANのVbus電圧表示、各種ステータスアイコンが表示されます。

HF	Hardware Filter	CAN IDのハードウェアフィルタが有効の時に点灯します。設定でHWFのMask0/Mask1がどちらも0ではない状態で有効となります
HD	Hwf value Displayed	CAN IDがハードウェアフィルタリングされたメッセージをモニタエリアにスクロール表示します
HA	Hardware Aux output	CAN IDがハードウェアフィルタリングされたメッセージのAUXコネクタSPI出力有効時に点灯します
SF	Software Filter	CAN IDのソフトウェアフィルタが有効の時に点灯します。設定でSoftwareFilter0~7のいずれかONで有効な状態となります
SD	Swf value Displayed	CAN IDがソフトウェアフィルタリングされたメッセージのAUXコネクタSPI出力有効時に点灯します
SA	Software Aux output	CAN IDがソフトウェアフィルタリングされたメッセージを表示します
CP	ComPare out	コンペアアウトが有効の時に点灯します。設定でCompareOut0~5のいずれかONで且つ指定されたSoftwareFilterがONの状態では有効となります

## 機能詳細(設定モード)

### TOP MENU

モニタモードでCボタンを2回押すと、設定画面に入ります。



## CAN speed

CAN通信の速度を設定します。CANバスの通信速度と一致していないとモニタ出来ません。リストに無い1Mbps以下の通信速度が必要な場合、ファームウェアを変更することで対応可能です。

## HardWareFilter

CAN IDを内蔵のCANコントロールIC(ハードウェア)でフィルタリングする機能です。多くのCANメッセージが流れてモニタ出来ない時や、本機の処理速度制限等によりエラーの赤色LEDが点滅している時に、本機能でCAN IDを絞り込みます。

CAN IDのID長さは、CAN 2.0A(Standard, 11bits)、CAN 2.0B(Extended, 29bits)から選択できます。選択は、Filter0~5の設定項目内で行います。



## Mask / Filterについて

Hardware FilterはMask0とMask1の2グループに分かれています。CANバスに流れてきたCANメッセージをCANコントロールICで認識してコピー(これを受信と呼びます)したのち、そのCANメッセージ



のIDがMask0グループで処理され、一致判定しない場合はMask1グループで処理され、それでも一致判定しない場合はCANメッセージがCANコントロールIC内で破棄されます。

一致判定されたメッセージは全て本機のマイコンで取得されハードウェアフィルタリングされたメッセージとしてモニタ表示、後段のソフトウェアフィルタリング機能へ転送、USB-C出力、AUX SPI出力処理します。

### Mask / FilterのCAN ID一致判定のしくみ

Mask bit val設定画面にて、CAN IDを後段のFilterで比較させるビットを本Maskにて指定します。Maskビット値が0の場合、そのビットは後段のFilterとCAN IDとのビット比較がされずパスします。例えばMask0を16進数で000(Standard, 11bits)、00000000(Extended, 29bits)に設定した場合は、CAN IDの全ビットがFilterとビット比較処理されずパスされるため、CAN IDが一致判定となり、ハードウェアフィルタリングされた値として出力されます。逆に16進数で7FF(Standard, 11bits)、1FFFFFFF(Extended, 29bits)に設定した場合は、CAN IDの全ビットがFilterとビット比較処理され、その結果全ビットが一致した場合はそのCAN IDが一致判定となります。

上記の通りビット判定(2進数)によるCAN IDフィルタリングのため、IDの個別指定(完全一致)、又は2の累乗の値によるIDの範囲指定がおすすめです。以下それぞれの設定例を示します。

### Mask / Filterの設定(CAN ID個別指定)

CAN ID(Standard: 11bits / Extended: 29bits)を個別指定、つまり特定の1つのIDを指定する場合、以下の手順で設定します。IDは16進数で設定します。10進→16進変換はPCのプログラマー電卓や各種Webサイトをご確認ください。Filter0~5の設定により、同時に最大6つまで個別ID指定ができます。

以下設定手順を示します。

1. Filter0~5のいずれかに入り、先にID lengthのメニューに入り、CAN ID長さをStandard、Extendedのどちらかに設定します。
2. 戻ってFilter valueメニューに入り、個別に指定したいCAN IDを16進数で指定します。
3. value設定後は、Eボタンを押すかENTERをタッチすると確定し、前の画面に戻ります。Cボタンを押すかBACKをタッチすると、変更した値は破棄されるため注意。
4. Mask0とMask1を両方とも最大値に設定します。最大値の設定は、7FF(Standard, 11bits)又は1FFFFFFF(Extended, 29bits)になるよう、カーソルをLボタンで一番左の桁に移動し、Uボタンを繰り返し押して上記最大値にして最後Eボタン/ENTERで確定します。
5. 設定後モニタモードに戻り、ステータスバーに HF HD アイコンが表示されていれば、CAN IDがハードウェアフィルタリングされた値がスクロール表示されます。アイコンが表示されていない場合、Uボタンを押して表示させます。

### Mask / Filterの設定(CAN ID個別指定)

CAN ID(Standard: 11bits / Extended: 29bits)を範囲指定する場合、以下の手順で設定します。範囲の指定は2の累乗(2,4,8,16,32,...)に基づいて指定すると設定が容易です。IDは16進数で設定します。10進→16進変換はPCのプログラマー電卓や各種Webサイトをご確認ください。Mask0が

グループとMask1グループの2グループで別々に設定できます。Mask0グループのフィルタリング範囲の開始位置はFilter0~1で2つ設定でき、Mask1グループはFilter2~5で4つ設定できます。

以下設定手順を示します。

1. Mask0グループを設定する場合はFilter0~1、Mask1グループを設定する場合はFilter2~5のいずれかに入り、先にID lengthのメニューに入り、CAN ID長さをStandard、Extendedのどちらかに設定します。
2. 戻って設定したいMask0、又はMask1で、フィルタリング範囲の幅を以下の表のとおり16進数で指定します。Mask0、Mask1のどちらか片方のみ範囲を設定したい場合、使用しない方のMask値は最大値を設定しておきます。

ID範囲の幅(10進)	Standard: 11bits	Extended: 29bits
2	7FE	1FFFFFFE
4	7FC	1FFFFFFC
8	7F8	1FFFFFF8
16	7F0	1FFFFFF0
32	7E0	1FFFFFE0
64	7C0	1FFFFFC0
128	780	1FFFFF80
256	700	1FFFFF00

3. Filter valueメニューに入り、CAN IDのフィルタリング範囲の開始位置を以下の対応する表のとおり16進数で指定します。フィルタリング範囲の開始位置を複数設定しない場合は、Mask0、Mask1グループ内におけるFilterの値を全て同じに設定します。

ID範囲の幅(10進): 2に設定した場合

ID開始位置(10進)	Standard: 11bits	Extended: 29bits
0	000	00000000
2	002	00000002
4	004	00000004
6	006	00000006
8	008	00000008

10	00A	0000000A
12	00C	0000000C
14	00E	0000000E
16	010	00000010
:	:	:

## ID範囲の幅(10進):4に設定した場合

ID開始位置(10進)	Standard: 11bits	Extended: 29bits
0	000	00000000
4	004	00000004
8	008	00000008
12	00C	0000000C
16	010	00000010
:	:	:

## ID範囲の幅(10進):8に設定した場合

ID開始位置(10進)	Standard: 11bits	Extended: 29bits
0	000	00000000
8	008	00000008
16	010	00000010
24	018	00000018
:	:	:

## ID範囲の幅(10進):16に設定した場合

ID開始位置(10進)	Standard: 11bits	Extended: 29bits
0	000	00000000

16	010	00000010
32	020	00000020
48	030	00000030
:	:	:

ID範囲の幅(10進):32に設定した場合

ID開始位置(10進)	Standard: 11bits	Extended: 29bits
0	000	00000000
32	020	00000020
64	040	00000040
96	060	00000060
:	:	:

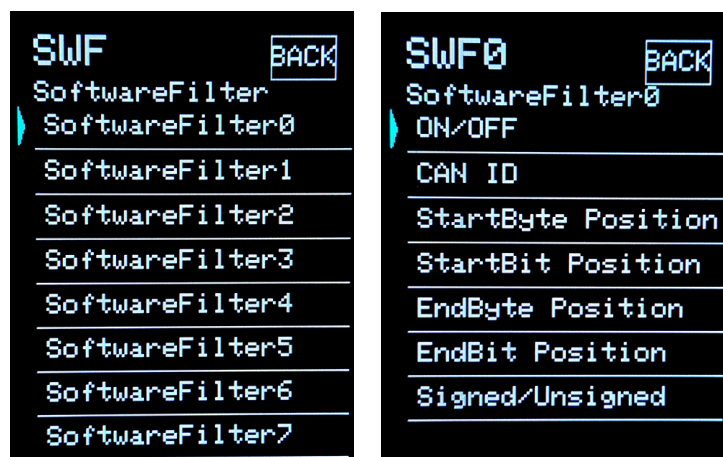
- 設定後モニタモードに戻り、ステータスバーに HF HD アイコンが表示されていれば、CAN ID がハードウェアフィルタリングされた値がスクロール表示されます。アイコンが表示されていない場合、Uボタンを押して表示させます。

## SoftWareFilter

CAN IDでハードウェアフィルタリングされた後のCANメッセージを、さらにマイコンのソフトウェアによってCAN IDでフィルタリングする機能です。さらに高度なフィルタリング、例えばCANデータフレーム中のデータbytesにおける切り出しが可能です。切り出した結果を表示したり出力したりできます。また閾値と比較してHigh/Lowのハードウェア出力ができるCompareOutへ値を出力します。

ソフトウェアフィルタはSoftwareFilter0~7の8通り指定でき、それぞれON/OFFやCAN ID個別指定、切り出し開始位置、切り出し終了位置、切り出した値の符号の有無を指定できます。

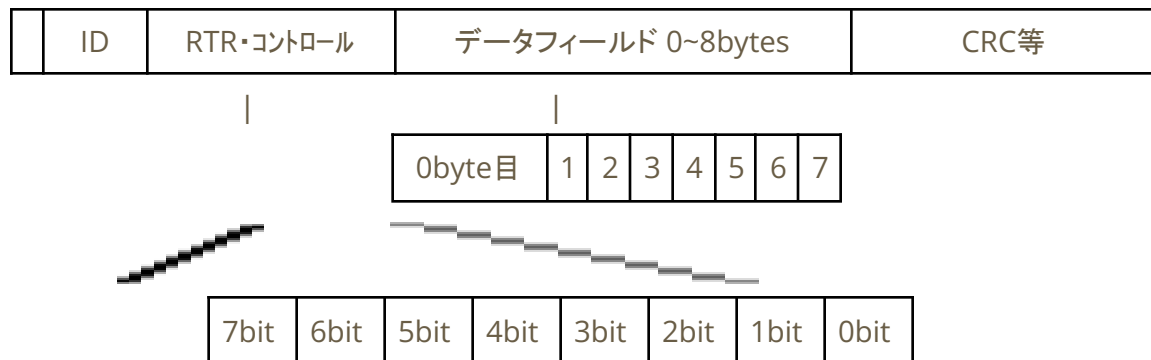




## SoftwareFilterの設定

1. SoftwareFilter0~7のいずれかのメニューに入り、ON/OFFにてONを選択します。
2. CAN IDにて、個別の(1つの)CAN IDを設定します。
3. StartByte Position、StartBit Positionにて、切り出しの開始位置を指定します。EndByte Position、EndBit Positionにて、切り出しの終了位置を指定します。位置の指定は以下の図と表を参照してください。

簡略化したCANデータフレーム



切り出し位置設定例

データフィールド例(16進)	切り出した値(16進)	StartByte Position	StartBit Position	EndByte Position	EndBit Position
e3 56 8f 1d 20 (5bytesの例)	e3	0	7	0	0
	2	4	7	4	4
	568f	1	7	2	0
6f 86 a1 4d cf 49 4a 48	a14dcf49	2	7	5	0

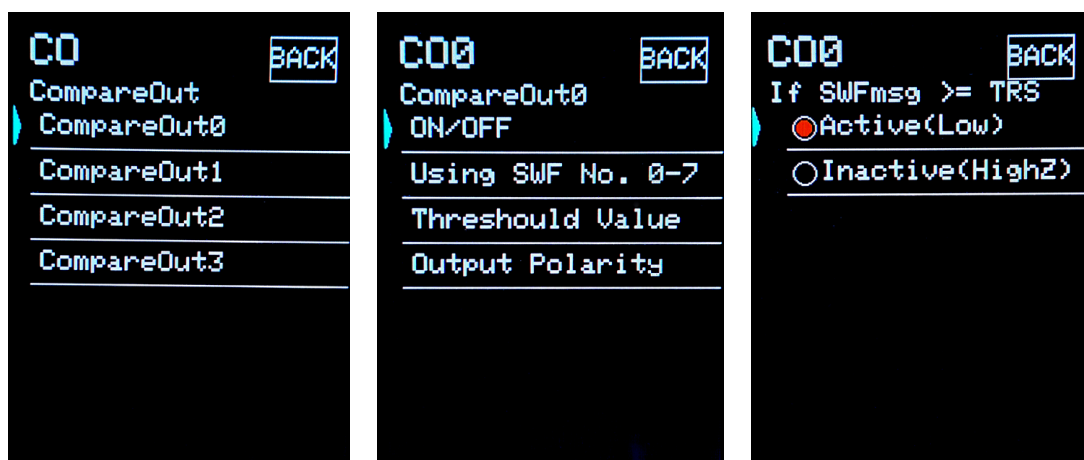
- Signed/Unsignedにて、切り出した値を符号ありか符号なしか選択します。Signedを選択すると符号ありとして、正負の値になります。切り出したビット列の最上位(もっとも左のbit)が"1"だと負、"0"だと正を表し、2の補数表現で扱われます。Unsignedを選ぶと符号なしとして、0~正の値になります。

例えば上記切り出し位置設定例の切り出したい値がe3の場合、Signedを選択した場合は出力値は-29となり、Unsignedを選択した場合は出力値は227となります。また切り出したビット値の幅が1byte以内の場合は1byteで処理され、それを超えると2bytesで処理され、それを超えると4bytesで処理されます。

- 設定後モニタモードに戻り、ステータスバーに SF SD アイコンが表示されていれば、CAN ID がソフトウェアフィルタリングされた値がスクロール表示されます。アイコンが表示されていない場合、Uボタンを押して表示させます。

## CompareOutput

ソフトウェアフィルタリングによりデータフィールドから切り出された値を、設定した閾値と比較して大きいか小さいか判定した結果をデジタル値として出力し、ディスプレイやLEDで表示したりAUXコネクタからON/OFF出力したりする機能です。そのため各種電子機器の特定の状態を検知して異常値を超えたらアラームを点灯させたり別の機器に通知したりすることができます。CompareOut0~3の4チャンネル使用できます。



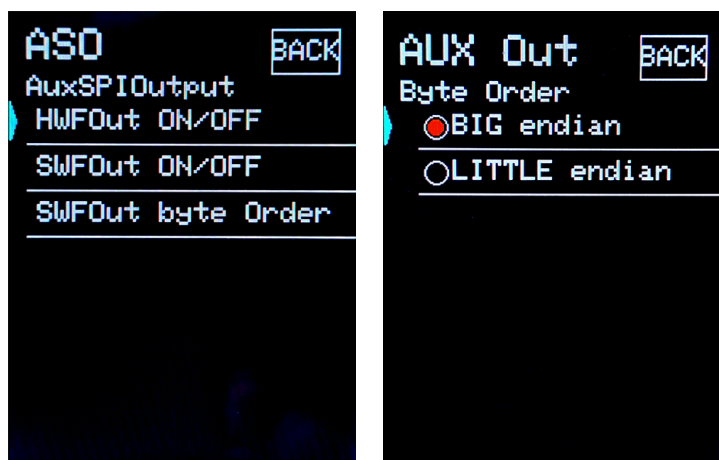
## CompareOutの設定

- CompareOut0~3のいずれかのメニューに入り、ON/OFFにてONを選択します。ONIになると、モニタモード時にステータスラインに CQ アイコンが表示されます。
- Using SWF No. 0-7で、切り出しデータ入力元のソフトウェアフィルタ番号を選択します。選択した番号のソフトウェアフィルタは予め設定しておいてください。
- Threshold Valueで、判定の閾値を16進数で入力します。

4. Output Polarityで、CO判定結果の極性を指定します。上記で指定したソフトウェアフィルタにより切り出された値(SWFmsg)が、上記で指定した閾値を超えた場合に、出力をActiveにするかInactiveにするか指定します。AUXコネクタ出力のみトランジスタにより極性が反転されるため、Activeの時はトランジスタがONしてLowとなります。
5. 設定後モニタモードに戻り、ステータスバーに CP アイコンが表示されていれば、上記判定結果が変化したときに画面に表示され、対応のLEDが点灯/消灯し、AUXコネクタから出力されます。

## AuxSPIOutput

AUXコネクタのSPI通信出力より、ハードウェアフィルタ後、ソフトウェアフィルタ後のCANメッセージの出力をそれぞれON/OFFできます。

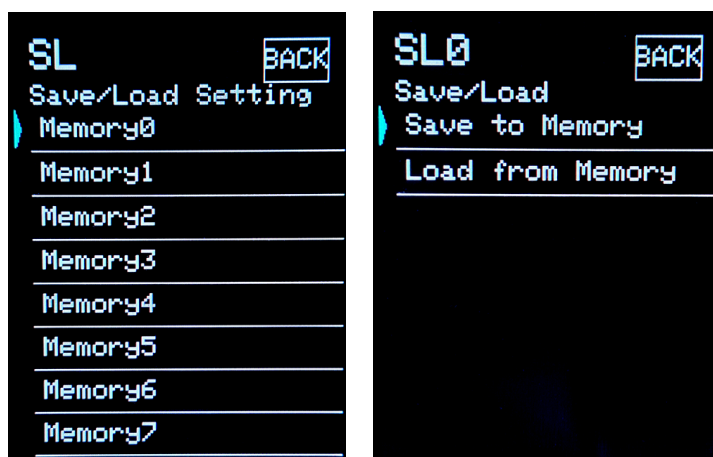


### AuxSPIOutputの設定

1. HWFOut ON/OFF、SWFOut ON/OFFメニューに入り、それぞれON/OFFを設定します。
2. SWFOutをONした場合、SWFOut byte Orderメニューで、バイトオーダー(出力バイト列の順番)を指定できます。CANのデータフィールドと同じバイトオーダーはBIG endianです。

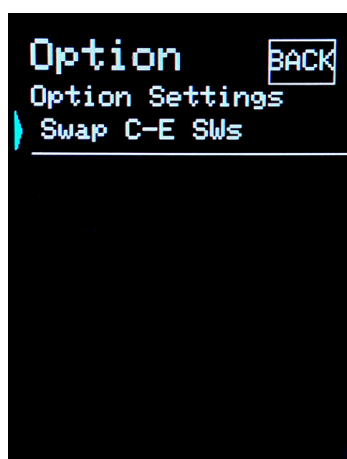
## Save/Load

設定値をまとめて保存できます。Memory0~7の8通り内蔵マイコンのFlash領域にSave/Loadできます。また本Memory0~7に保存しなくても、仮の設定値がFlash領域に保存され、電源起動時に読み出されます。



## Option

その他のオプション設定メニューです。Swap C-E SWsは、EボタンとCボタンの機能を入れ替えます。デフォルトでEボタンはEnter(確定)、CボタンはCancel(キャンセル)です。



## 機能詳細(CAN接続部)

### ピン配置切替ヘッダ

CANケーブルを本機のDSUB-9ピンオスコネクタに接続する前に、接続するケーブルの種類に合わせてヘッダの位置を変更します。CAN標準ケーブル・OBD-IIケーブルのどちらでも接続可能です。

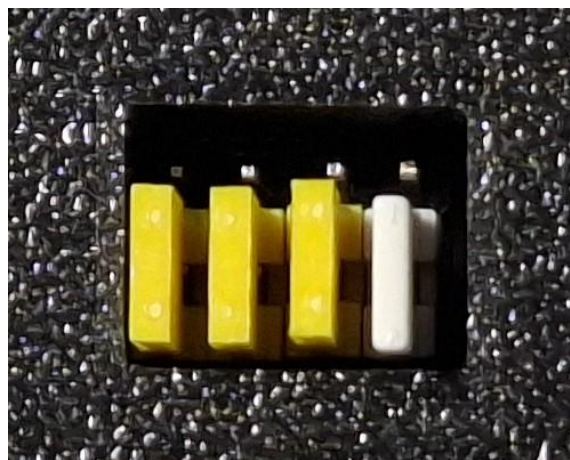
下図の通り左3つのヘッダを指やラジオペンチ等で引き抜き、正しい側に差し替えます。上側に挿すとOBD-II用ピン配置、下側に挿すとCAN標準ピン配置となります。(ヘッダの色は変更になる場合があります)



OBD2用ピン配置(左3つが上側)



CAN標準ピン配置(左3つが下側)



## DSUB-9pinオスコネクタピン配置

DSUB-9pinオスコネクタ搭載。上記ヘッダの切替で2種類のピン配置(ケーブル)に対応。

- CAN標準ピン配置(CiA DS102推奨)
- OBD2用ピン配置(市販のOBD2-DB9変換ケーブル用)

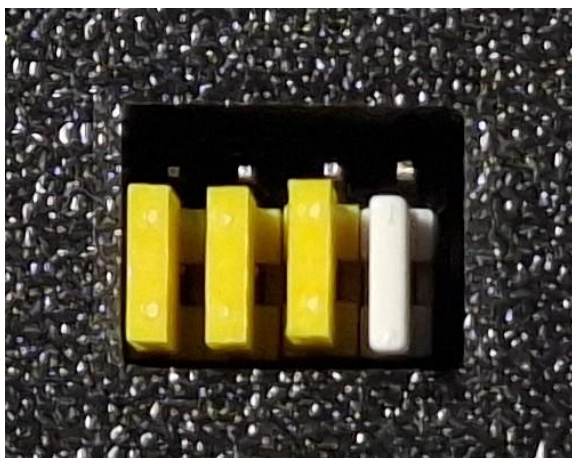
DSUB-9ピンNo.	CAN標準ピン配置	OBD2用ピン配置
1	-	Signal GND
2	CAN-L	Chassis GND
3	V- (Signal GND)	CAN-H
4	-	-
5	-	CAN-L
6	-	-
7	CAN-H	-
8	-	-
9	V+ (Power In)	Battery Power In
シールド	GND	GND



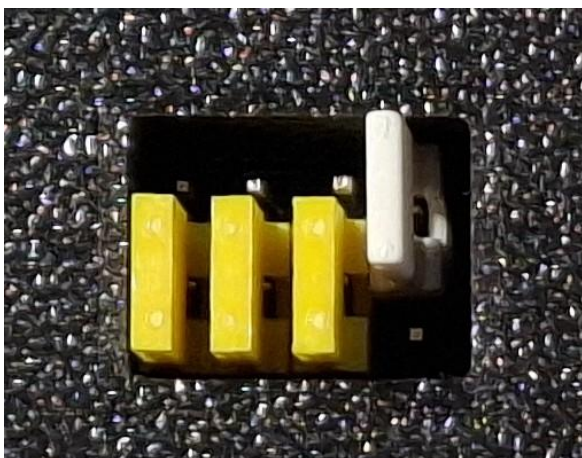
### TerminateON/OFFヘッダ

CANバスにおける終端抵抗のON/OFFをヘッダの位置で切替できます。下図の通り右1つのヘッダを指やラジオペンチ等で引き抜き、正しい側に差し替えます。下側に挿すとON、つまり終端抵抗が接続され、上側に挿すとOFFとなります。(ヘッダの色は変更になる場合があります)

Terminate ON(右1つが下側)



Terminate OFF(右1つが上側)




## 機能詳細(AUXコネクタ)

### ピン配置

AUX(auxiliary)コネクタのピン配置を示します。表のPin1,2側がLボタン側、表のPin19,20側がDSUB-9pinオスコネクタ側です。

表のnRESET及びSPI nSSの“n”は論理反転の意味です。

機能	Pin No.		Pin No.	機能
GND	20		19	GND
CAN L	18	17	COCOM	
CAN H	16	15	CO3	
+5V	14	13	CO2	
+3.3V	12	11	CO1	
ERROR	10	9	CO0	
nRESET	8	7	SPI nSS	
GND	6	5	NC	
SPI MOSI	4	3	SPI SCK	
NC	2	1	SPI MISO	

## 機能

### SPI出力 Pin1~7

SPI通信出力端子です。本機がSPI通信におけるホスト(マスタ)固定です。MOSI、SCK、nSS端子は出力端子です。

### nRESET

本機内部の各ICのRESET端子に接続されており、3.3Vに抵抗でプルアップされています。本機のRESETボタンが押されるとLowになり、本機の動作がリセットされます。

### CO0~3, COCOM

CompareOut機能の出力です。CO0~3はそれぞれトランジスタのコレクタ端子に接続されているため電流はシンク(引き込み)方向のみとなります。+5V端子から外付け抵抗を介してLEDを点灯させたり、別電源を用意してリレーやソレノイド等の誘導性負荷を接続できます。誘導性負荷を接続する場合は逆起電力によるトランジスタ故障を防ぐためCOCOM端子を誘導性負荷の電源に接続してください。

### ERROR

マイコンのエラー信号出力端子に接続されています。マイコンがエラーを出すと、ステータスLEDのERROR(赤色)を点灯させると共に本端子がHigh(3.3V)になります。

## +3.3V, +5V, GND

外部回路用電源出力です。約50mA程度出力可能です。GNDは、DSUB-9pinオスコネクタ経由でCANバスのSignal GNDと電氣的に接続されます。

## CAN H, CAN L

DSUB-9pinオスコネクタ経由でCANバスラインに直接接続されます。

## 仕様

CANトランシーバ/コントローラはMCP2551/MCP2515互換ICを使用。ディスプレイはILI9341搭載2.8" TFT-LCDを使用。マイコンはATSAMD21G18 32bit MCUを使用。

## 表示・フィルタリング機能

40行×40桁、縦長、最上位行は凡例/ステータス表示。

- 2.8"320x240dot TFT-LCD、40行×40桁、縦長、バックライト & タッチパネル搭載
- スクロール速度： 30ms/Line max.
- 最大6種類のCANデータフレームIDフィルタリング設定が可能
- CAN ID： 16進4桁、CAN DATA: 16進2桁×8bytesを1行に表示
- Vbus電圧表示： DSUB-9pinオスコネクタの9pin電圧を約100ms毎に更新表示
- 何も設定しない初期状態の場合は、Mask0のmaskbit valとFilter0~5のfilter valが全て0で、全てのCANメッセージが表示されます。その場合、CANバスに流れているメッセージの量が多い場合はメッセージが流れてしまい全く読めません。また
- CAN 2.0A (ID長: 11bits, Standard)、CAN 2.0B (ID長: 29bits, Extended) の選択は、Filter0~5内で行います。StandardとExtendedの両方のメッセージをすべて表示させる場合は、Filter0~5のいずれか1つ以上のID Length設定をExtendedに変更して、Mask0とMask1のmaskbit valを両方0にします。

## 出力機能

フィルタリング機能によってフィルタされたCANメッセージをUSBコネクタ経由でPCに送信できるUSB通信機能搭載。PC側はCOMポートとして認識でき、ArduinoIDEや通信ソフト等でCANメッセージを受信できます。但し通信遅延や内部遅延の都合上、フィルタリングの内容によってはCANメッセージが欠落して受信される可能性があります。

- USBシリアル通信: 115200bps、データ8bit、ストップ1bit

またAUXコネクタにおけるSPI通信端子にて上記と同じメッセージを出力するSPI出力機能搭載。

- AUXコネクタ: 2.54mmピッチ、20pinのピンソケット
- SPI通信: 4MHz、本機がホスト(マスタ)固定、SCK、MOSI、SS



- SPI電圧: 3.3V

コンペア出力機能によって設定した閾値との比較結果をAUXコネクタから出力できます。CO0~CO3に対応したピンがあります。リレーなどの誘導性負荷を動作させる場合は、COCOMをリレーなどの電源電圧に接続してください。

- 最大シンク電流: 1A peak
- 耐電圧: DC 50Vmax(CO0~3、COCOM共)

## CANTランシーバ/コントローラ

- CANTランシーバ/コントローラ: MCP25625(MCP2551/MCP2515互換IC)
- フォーマット: CAN V2.0A(標準11bit ID)、2.0B(拡張29bit ID)
- 通信速度: 10k、50k、100k、125k、250k、500k、800k、1Mbps
- 終端抵抗120Ω、ヘッダの切替でON/OFF可能

## 入力電源

DSUB-9コネクタの9pinから直接又はOBD-IIケーブルにてDC12Vを入力する方法と、USB-typeCコネクタによるDC5Vを入力する方法が可能です。

- DSUB-9ピンコネクタ9pin入力電圧: DC7~18V
- USB-Cコネクタ入力電圧: DC5V±5%

入力電圧が上記範囲以下の場合、正常に起動しなかったり異常動作の可能性があります。また上記電圧範囲を超えた場合、本機や本機に接続された機器の異常動作・故障・発煙・発火等の恐れがあります。

## ファームウェア

Arduinoで開発可能です。最新ファームウェアはこちら。ファームウェアのアップデートご要求やバグレポート等歓迎します。下記お問合せ先までご連絡ください。

## 消費電力・寸法・重量・動作温度

消費電力: 5V 200 mA typ.

寸法: WDH = 115 x 100 x 22mm(突起物除く)

重量: 150 g typ.

動作温度/保存温度: 室温 (車内など50°C以上の場所に放置しない、結露させない)

## 制限事項

- 本製品は実験研究用の開発キットです。法令に対応したスキャンツール類ではありません。また暗号化されたOBD2メッセージの表示や診断機能等はありません。

- 通信遅延や内部遅延の仕様上、フィルタリング設定内容によっては表示及びシリアル通信時にCANメッセージが欠落する可能性があります。
- 初期不良交換以外の本機を使用したことにより生じたいかなる事象も保証いたしません。

## 技術お問い合わせ先

ファームウェア等の機能追加・修正依頼はこちら

[fundylab@gmail.com](mailto:fundylab@gmail.com)