

DataCatcher3

ユーザーズマニュアル



通信ラインモニタ

DataCatcher3

ハートランド・データ株式会社

〒326-0338 栃木県足利市福居町 361

TEL: 0284-22-8791 / FAX: 0284-22-8792

URL: <http://www.hlde.co.jp>

DataCatcher3 USER'S MANUAL について

本マニュアルは、『*DataCatcher3*（以下 *DC3*）』の主な仕様、基本操作方法 及びその他詳細な機能や設定について説明されております。

DC3 の操作方法はすべて DCNEXT の操作 방법에準じております。特に指定のない限り、本マニュアル内の「DCNEXT」に関する記述はすべて「DC3」と読み替えてください。

- 本プログラム及び説明書は著作権法で保護されており、弊社の文書による許可がない限り複製、転載、改変など一切できません。
- 本製品の内容及び仕様は予告なしに変更されることがあります。
- 本製品は、万全の注意を払って作成されていますが、ご利用になった結果については、弊社は一切の責任を負いかねますのでご了承下さい。
- Windows, Windows XP, Windows Vista, Windows 7, Windows8/8.1, Windows 10 は、米国マイクロソフト社の商標です。
その他、本文中に記載されている会社名・製品名は各社の商標または登録商標です。

Copyright(C) 2016 ハートランド・データ株式会社

目 次

1. 製品の機能・特徴.....	5
1-1: 汎用高速通信ラインモニター.....	5
2. ハードウェア仕様.....	6
2-1: キャッチングライン.....	6
2-2: A/D入力(アナログBOX使用).....	6
2-3: 外部トリガー入力.....	6
2-4: イベント出力ライン.....	6
3. 対応している通信プロトコル.....	7
3-1: 3線同期シリアル.....	7
3-2: I2CBUS.....	7
3-3: UART.....	7
3-4: 16ビットパラレル.....	7
3-5: データロガー.....	7
4. 仕様一覧.....	8
5. 動作環境.....	9
5-1: オペレーティングシステム.....	9
5-2: コンピュータ本体.....	9
5-3: CPU.....	9
5-4: メモリ.....	9
5-5: その他.....	9
6. インストール手順.....	9
6-1: PCアプリケーションのインストール方法.....	9
6-2: USBドライバのインストール方法(Windows XP).....	12
6-3: USBドライバのインストール方法(Windows7 32bit/64bit-OS).....	14
6-4: USBドライバのインストール方法(Windows 8/8.1, Windows10 : 32bit/64bit-OS).....	17
7. DC3 の構成.....	18
8. DC3 の概観.....	18
8-1: 背面.....	18
8-2: Data Catcher III I/F ボード.....	19
8-3: 説明.....	19
9. ハードウェアの取扱い.....	20
9-1: DC3 本体の立ち上げ手順.....	20
9-2: 注意事項.....	20
10. 基本操作.....	21
10-1: DC3 の起動.....	21
10-2: データログ画面.....	22
10-3: 各種ファイル操作を行う.....	23
10-4: 表示操作を行う.....	24
10-5: キャッチデータに条件を設定する.....	25
10-6: LOG表示フィルター設定ウィンドウ.....	26
10-7: キャッチフィルター設定ウィンドウ.....	27
10-8: イベントフィルター設定ウィンドウ.....	28
10-9: トリガーフィルター設定ウィンドウ.....	29
10-10: 共通マスクフィルター設定ウィンドウ.....	31
11. [3WIRE CATCH]Catch Program でキャッチできる波形.....	32
11-1: 3線シリアル通信の波形.....	32
12. [3WIRE CATCH]Catch Program の設定について.....	33

12-1: <キャッチタイプの設定>タグでの設定内容について	33
12-2: <チャンネル1><チャンネル2><チャンネル3>タグでの設定内容について	34
12-3: ポート設定	38
12-4: コネクター表	38
12-5: 制限事項	39
13. [I2CBUS_400K]Catch Program の設定について	40
13-1: <キャッチタイプの設定>タグでの設定内容について	40
13-2: <チャンネル1><チャンネル2>タグでの設定内容について	41
13-3: ポート設定	42
13-4: コネクター表	42
13-5: 制限事項	43
14. [I2CBUS_400K]Catch Program でキャッチできる波形	43
14-1: スタートコンディションと、ストップコンディションがペアになっているタイプ	43
14-2: スタートコンディションと、ストップコンディションの間にリスタートがあるタイプ	43
15. [UART]Catch Program でキャッチできる波形について	44
15-1: Low active の波形	44
16. [UART]Catch Program の設定について	45
16-1: <キャッチタイプの設定>タグでの設定内容について	45
16-2: <チャンネル1><チャンネル2>タグでの設定内容について	46
16-3: ポート設定	49
16-4: コネクター表	49
17. [PARA_CATCH]Catch Program でキャッチできる波形について	50
17-1: 16Bit までの DATA 幅の平行通信の波形	50
18. [PARA_CATCH]Catch Program の設定について	50
18-1: <キャッチタイプの設定>タグでの設定内容について	51
18-2: <チャンネル1><チャンネル2>タグでの設定内容について	52
18-3: ポート設定	56
18-4: コネクター表	56
18-5: 制限事項	57
19. [DATA_LOGGER]Catch Program について	58
19-1: デジタル信号取得	58
19-2: ADデータ取得	58
20. データロガー機能の設定について	59
20-1: <キャッチタイプの設定>タグでの設定内容について	59
20-2: <AD MASK/LEVEL DATA>タグの設定について	60
20-3: ポート設定	60
20-4: コネクター表	61
21. コマンド解析用定義ファイルの作成方法	62
21-1: 特 徴(こんなことができて便利)	62
21-2: 制限・制約	62
21-3: セクションについて	63
21-4: 定義体について	67
21-5: 書式について	68
21-6: 文字パターンの定義について	69
21-7: 特殊な定義体について	70
21-8: 具体的な記述例と画面イメージ	71

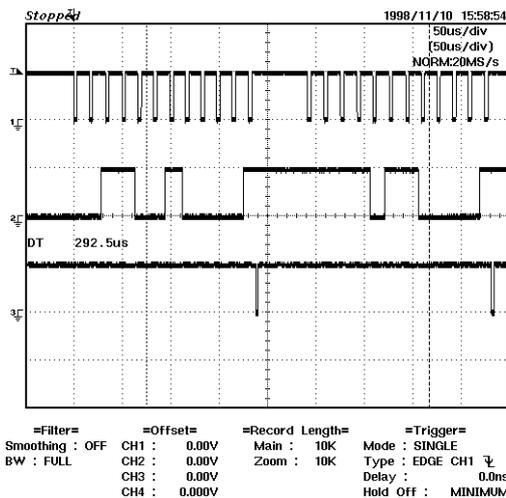
1. 製品の機能・特徴

1-1: 汎用高速通信ラインモニター

DC3 は組み込みマイコンの通信部分のデバック、評価を効率良く行う事を目的として開発された、**組み込みシステム開発用のラインモニター**です。

CPU とシリアルまたはパラレル I/F で接続される LSI を使用する場合や CPU 同士で通信を行うようなシステムを開発する際、通信データの内容を確認するためには、従来はデジタルオシロスコープ等で波形データを直接確認していました。

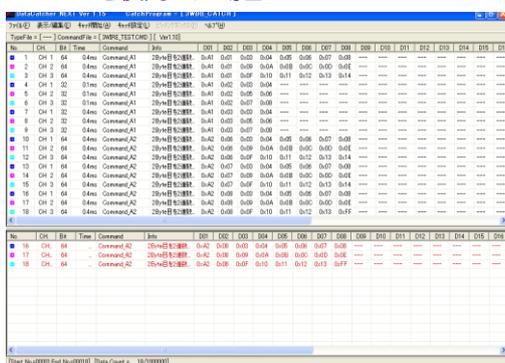
従来の方法（デジタルオシロを使用）



通信の内容を確認するためには 1/0 のビットパターンを数値に変換し、さらに LSI の仕様書を参照して数値に対応する処理内容を理解するという手順が必要でした。

また、デジタルオシロスコープではモニタできる時間も短く、処理全体の流れを把握する事は困難でした。

DC 3 を使用した場合



DC3 の特長

- ①通信データをパケットとして区切り、1 件ずつ表示します。
- ②通信データをユーザー定義の文字列として、分かりやすく表示します。
- ③リアルタイムでキャッチしたデータを表示します。
- ④パケットとパケットの間隔を 0.1/1.0msec の分解能で計測出来るのでタイミングの検証が簡単です

・DC3 を次のような方にお勧めします。

- 組み込みマイコンシステムの**ソフト開発担当者** ... デバッグ作業の軽減に
- 組み込みマイコンシステムの**評価担当者** ... マイコンソフトの評価・確認に
- 組み込みマイコンシステムの**ハード担当者** ... ソフト担当者との意思統一に

DC3 を使用する事により、周辺 LSI や CPU 間の通信における、デバッグ作業の負荷を大幅に軽減する事ができます。

2. ハードウェア仕様

2-1: キャッチングライン

- a) 最大CH構成 : 3CH
- b) 入力形式 : 470K Ω でGNDへPULLDOWN
- c) 入力電圧範囲 : 0V~5V
- d) 入力レベル : IN_A~IN_E(シュミットトリガ入力)
 VIH=2.31V VIL=0.99V
 IN_AX~IN_EX(3.3Vモード)
 VIH=2.00V VIL=0.8V
 IN_AX~IN_EX(1.8Vモード)
 VIH=1.17V VIL=0.63V
- e) 最大入力周波数 : 20MHz

2-2: A/D入力(アナログBOX使用)

- a) 最大CH構成 : 2CH
- b) 入力形式 : 1M Ω でGNDへPULLDOWN
- c) 入力電圧範囲 : 0V~24V(要入力レンジ切り替え)
- d) サンプルング周波数 : 10KHz/50KHz
- e) AD入力分解能 : 8BIT

2-3: 外部トリガー入力

- a) 入力ライン数 : 1CH
- b) 入力形式 : 470K Ω でGNDへPULLDOWN
- c) 入力電圧範囲 : 0V~5V
- d) 入力レベル : VIH=2.00V VIL=0.8V

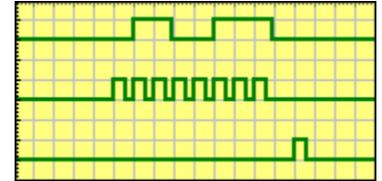
2-4: イベント出カライン

- a) 出カライン数 : 1CH
- b) 出力レベル : 0V~3.3V
- c) 最大出力電流 : 最大24mA

3. 対応している通信プロトコル

3-1:3線同期シリアル

- ・CLOCK、DATA、STBを有する3線同期シリアル通信
- ・20MHzまでのクロック周波数に対応
- ・3chまで同時使用が可能
- ・STBがないタイプもタイプファイルをカスタマイズすることで
パケットとして取り込み可能
- ・通信データのキャッチ(受信)/ピッチ(送信)が可能



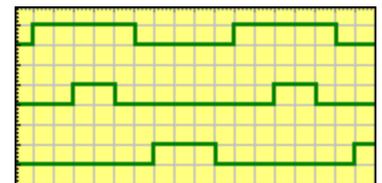
3-2:I2CBUS

- ・4MHzまでのクロック周波数に対応
- ・2Chまで同時使用可能



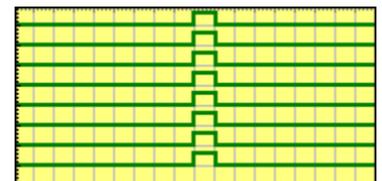
3-3:UART

- ・100BPS から 1000000BPS まで対応
- ただし固定値の場合は 110BPS から 921600BPS まで対応
- ・2Chまで同時使用が可能



3-4:16ビットパラレル

- ・16ビット幅までのデータをクロックに同期して取り込み
- ・連続データの場合、3.5MHzまでのクロック周波数に対応。
- ・非連続の場合は、10MHz程度まで可能。



3-5:データロガー

- ・CMOSレベルのデジタル入力.....8本
サンプリングレート 10KHz/50KHz
- ・0Vから24VのAD入力(8BIT分解能) ...2本
サンプリングレート 10KHz/50KHz



4. 仕様一覧

機能 / SPEC		DataCatcher3
		3線同期シリアル 通信
データキャッチ	対応通信速度	20 MHz MAX
	制限事項	なし
		UART
データキャッチ	対応通信速度	100 BPS ~ 1000000 BPS (※固定値の場合は 110 BPS ~ 921600 BPS)
	データ転送容量	57600 bps
	制限事項	921600 BPS でも、2CH 同時使用可能
		I2C BUS
データキャッチ	対応通信速度	4 MHz MAX
	制限事項	なし
		Pararell 通信
データキャッチ	対応通信速度	10 MHz MAX(連続の場合は、3.5MHz)
	入力 DataBit 数	16 Bit MAX
	制限事項	なし
		データロガー
サンプリング時間(周波数)		20usec(50KHz) / 100usec(10KHz)
CMOSレベルのデジタル入力		8ライン
0V~24VのAD入力		2ライン(AnalogBox 使用)
AD入力分解能		8 Bit
		その他
最大データログ件数		100 万件
PCとの接続		USB2.0
入力電圧範囲		<ul style="list-style-type: none"> ・ターゲットに合わせて、スレッシュホールドを 3.3V 電源タイプ/1.8V タイプに切替可能。(5V 入力可) ・スレッシュレベル <ul style="list-style-type: none"> IN_A~IN_E(シュミットトリガ入力) VIH=2.31V VIL=0.99V IN_AX~IN_EX(3.3V モード) VIH=2.00V VIL=0.8V IN_AX~IN_EX(1.8V モード) VIH=1.17V VIL=0.63V

< 補 足 >

- ・各種設定ファイル(ログファイル、コマンドファイル、タイプファイル、条件ファイル)は、DC2000/DC NEXT で使用していたものをそのまま使用できます。
- ・UART のボーレートは 100~1000000BPS までの任意の値が指定できますが、クロックの分周比によって、指定した値と、実際の値に誤差が生じます。この誤差は、以下の計算式によって求められます。

ユーザ指定ボーレート値を T、分周値 N、実際のボーレート R、誤差率 E としたとき、

$$N = \text{INT}(C / (T * 8))$$

$$R = \text{INT}(C / N / 8) [\text{bps}]$$

$$E = (1 - (T / R)) * 100 [\%]$$

となっています。仮に T = 115200[bps]とすると、

$$N = 56$$

$$R = 115327[\text{bps}]$$

$$E = 0.11[\%]$$

となります。

5. 動作環境

5-1:オペレーティングシステム

Microsoft Windows XP, Windows Vista, Windows 7, Windows 8/8.1, Windows10

5-2:コンピュータ本体

上記OSの動作するPC/AT互換機

5-3:CPU

Pentium 800MHz 以上を推奨

5-4:メモリ

256MB以上のRAMを推奨

5-5:その他

4MB以上の空き容量のあるハードディスク
USB2.0ポート

6. インストール手順

6-1:PCアプリケーションのインストール方法

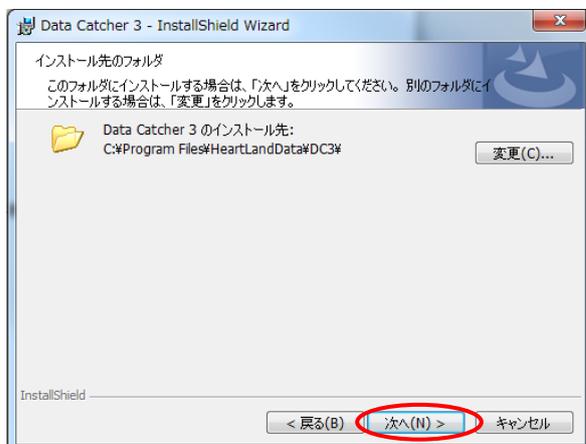
1. 付属のCD-ROM『DataCatcherⅢ Install Disc』をCD-ROMドライブに入れる。
アイコン [DC3Inst.msi]をダブルクリック。
2. 以下の画面が表示されたら、[次へ]をクリック。



3. 以下の画面が表示されたら、[次へ]をクリック。



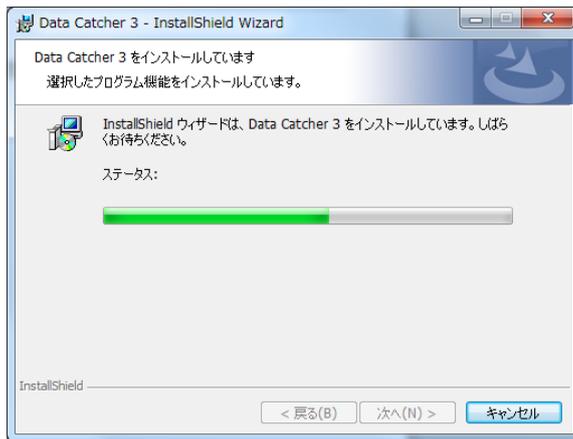
4. [インストール先のフォルダ]が表示されたら、[次へ]をクリック。
デフォルトの保存先で特に問題がなければ、このまま[次へ]をクリックして下さい。
別のフォルダにインストールしたいときは、[参照]をクリックしてインストール先を指定し直して下さい。



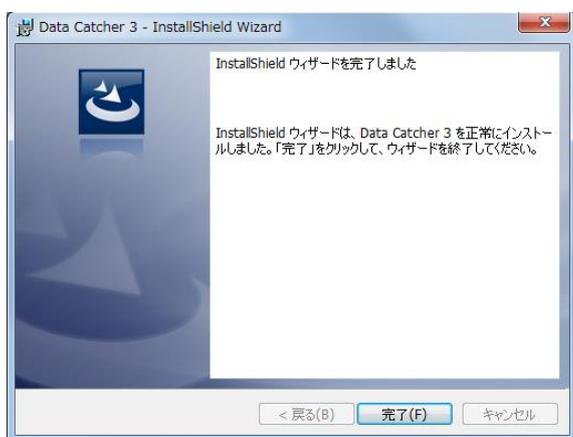
5. [プログラムをインストールする準備ができました]が表示されたら、[インストール]をクリック。



6. [DataCatcher III をインストールしています]と表示され、インストールが実行されます。



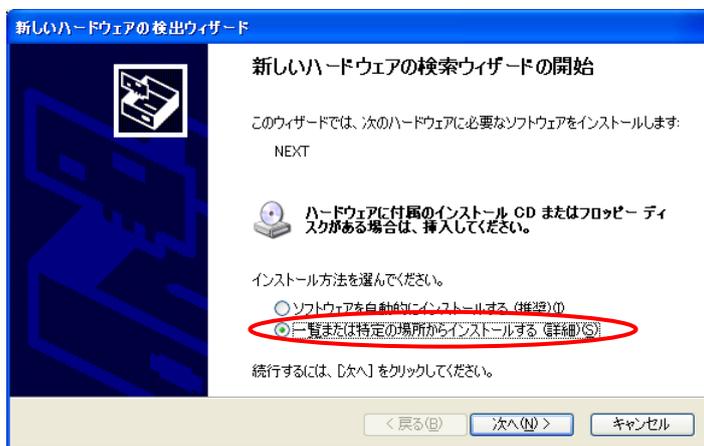
7. [InstallShield ウィザードを完了しました]と表示されたら、[完了]をクリック。
[完了]をクリックしたら『DataCatcherⅢ Install Disc』を取り出して下さい。



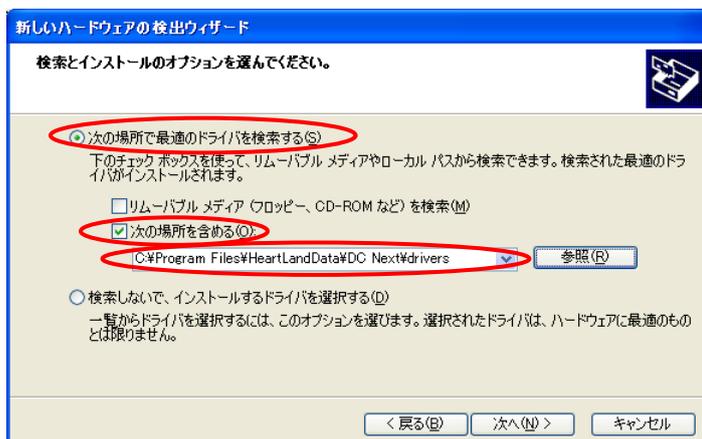
これでアプリケーションのインストールが完了しました。

6-2:USBドライバのインストール方法(Windows XP)

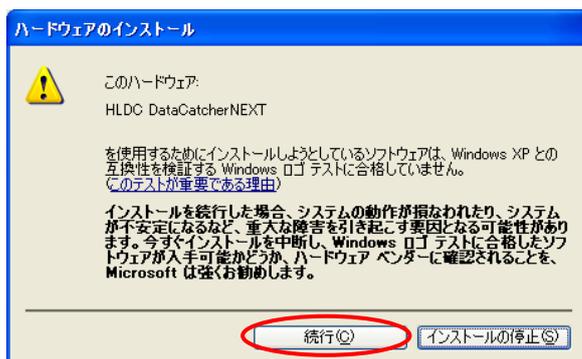
1. パソコンと DC3 本体を付属の USB ケーブルで接続する。
2. DC3 の電源を入れる。
DC3 本体に AC アダプタを差込み、DC3 本体の[POWER]スイッチを ON にして電源を入れて下さい。
3. ランプの点灯を確認します。
電源が入ると[POWER]ランプが緑色に点灯します。
[POWER]ランプの点灯後、[新しいハードウェアの検出ウィザード]が自動的に表示されます。
【注意】プローブをはずした状態で電源を投入して下さい。
4. [新しいハードウェアの検索ウィザードの開始]が表示されます。
[一覧または特定の場所からインストールする]を選んで[次へ]をクリックします。



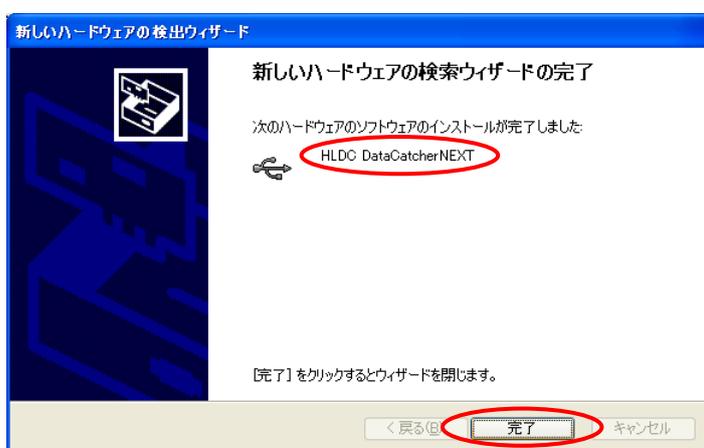
5. [検索とインストールのオプションを選んでください。]が表示されます。
[次の場所を含める]にチェックをつけ、[参照]ボタンをクリックします。
フォルダの参照画面で DC 3 のドライバが保存されているドライブ名→フォルダ名(HeartLandData) →フォルダ名(Dc3)→[driver]フォルダを選び、[OK]ボタンをクリックします。
テキストボックスに[フォルダの参照]で選択したドライバの保存先が表示されているのを確認して[次へ]ボタンをクリックします。



6. [このハードウェア～を使用するためにインストール使用としているソフトウェアは、Windows XP との互換性を検証する Windows ログテストに～]と表示されましたら、そのまま[続行]を選びます。

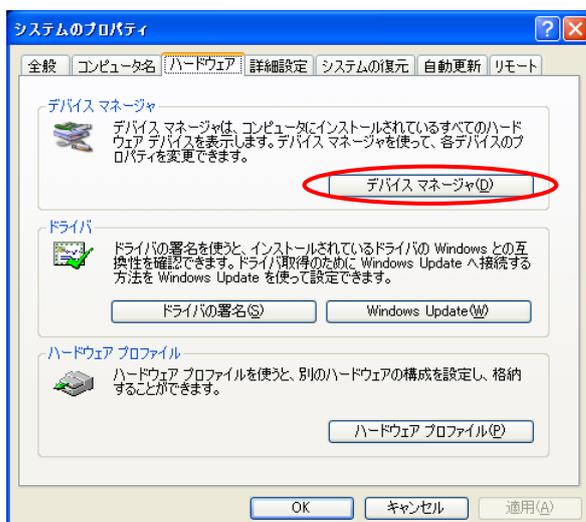


7. [新しいハードウェアの検索ウィザードの完了]が表示されましたら、[完了]をクリックします。

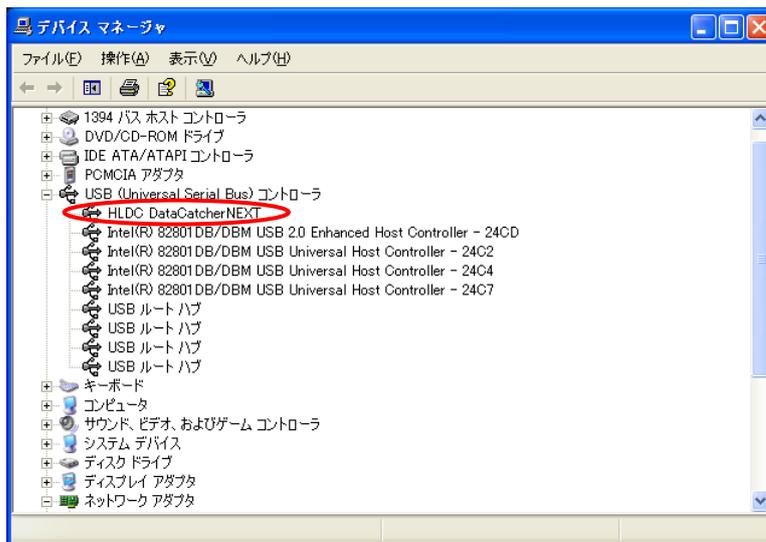


以上でインストールは完了です。

8. 正常にインストールされたかを確認します。
コントロール パネル→デバイスマネージャの「システムのプロパティ」内、「ハードウェア」タグを選択し、[デバイスマネージャ]をクリックします。



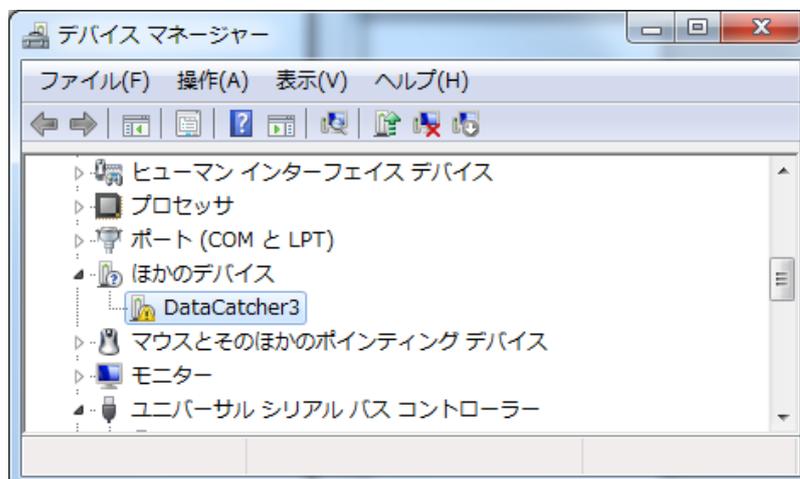
9. デバイスマネージャの「USB(Universal Serial Bus)コントローラ」の中に「HLDC DataCatcher3」の表示があれば正常にインストールされています。



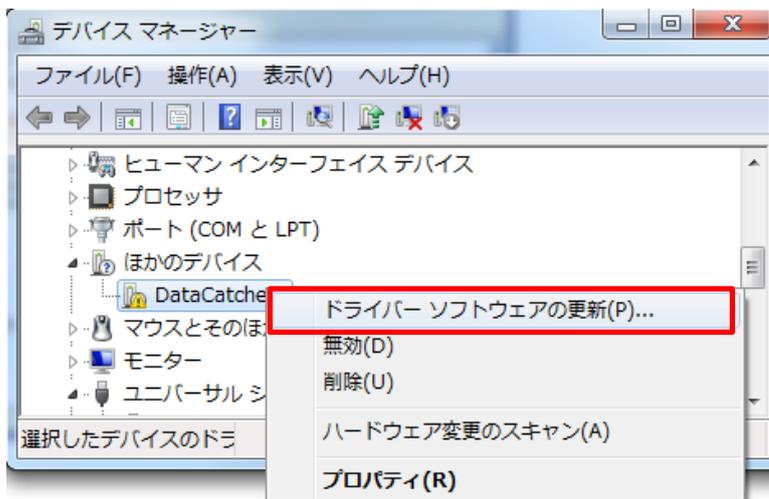
6-3:USBドライバのインストール方法(Windows7 32bit/64bit-OS)

Windows7 の場合、DC3 を接続し電源を入れても、USB デバイスドライバは自動でインストールされません。手動でインストールする必要があります。手動インストール方法は以下の通りです。

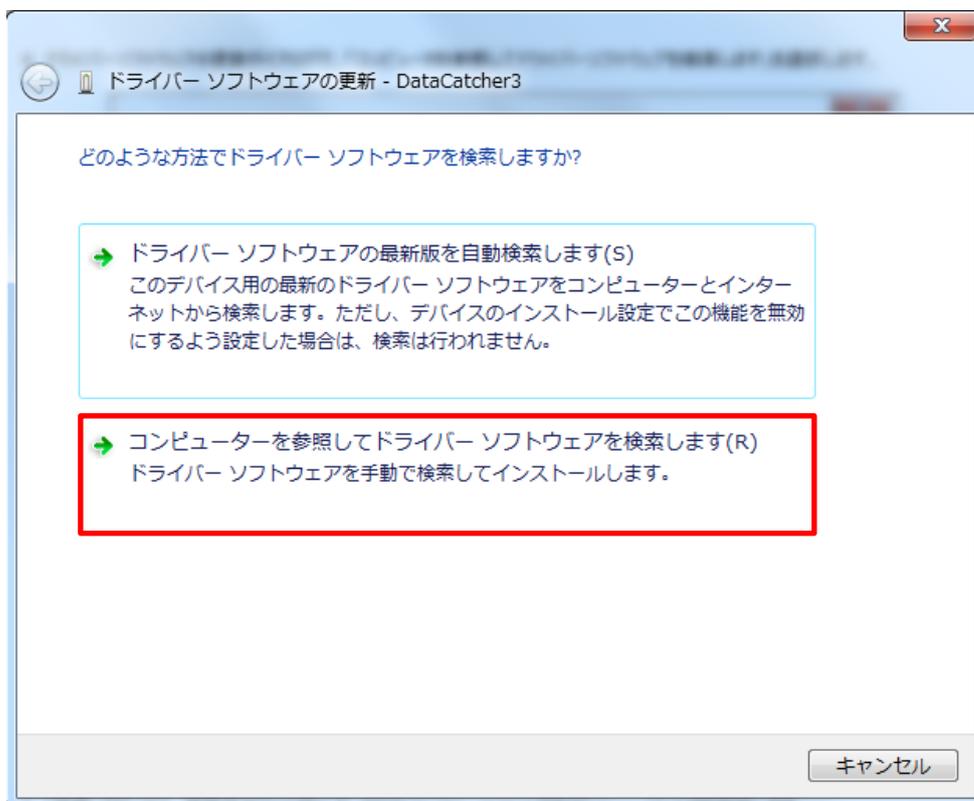
1. Windows のコントロールパネルから[デバイスマネージャ]を開きます。
2. デバイスマネージャで、「DataCatcher3」のデバイスを見つけます。



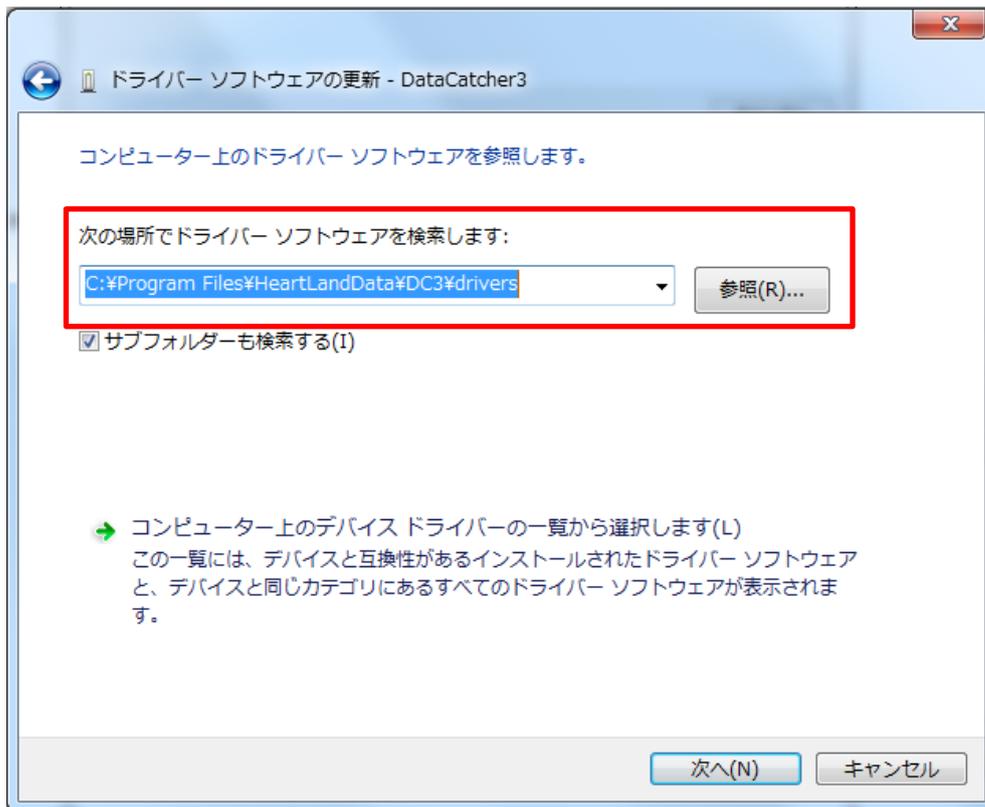
3. 「DataCatcher3」のデバイスを右クリックし、表示されたメニューから「ドライバーソフトウェアの更新」をクリックします。



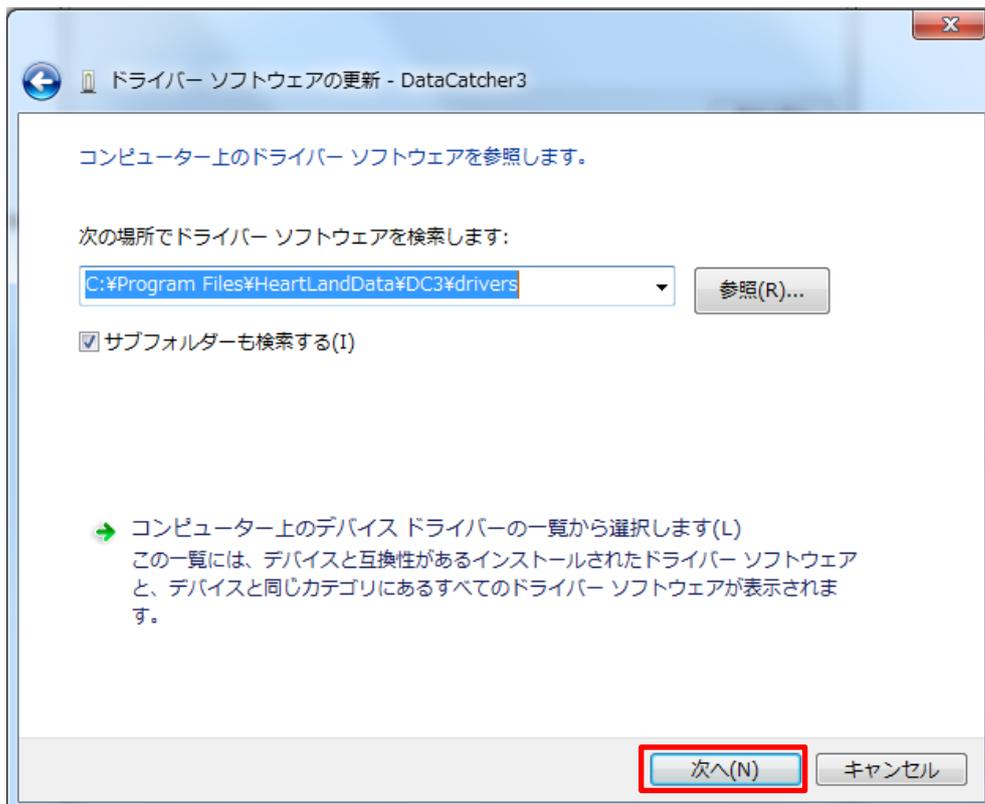
4. ドライバーソフトウェアの更新ダイアログで、「コンピュータを参照してドライバーソフトウェアを検索します」を選択します。



5. 「参照」ボタンから、検索先のフォルダとして、DC3 インストールフォルダ内の「drivers」フォルダを指定します。
32bit-OS の場合、DC3 インストールフォルダ内の「drivers¥x86」というフォルダを指定してください。
64bit-OS の場合、DC3 インストールフォルダ内の「drivers¥x64」というフォルダを指定してください。

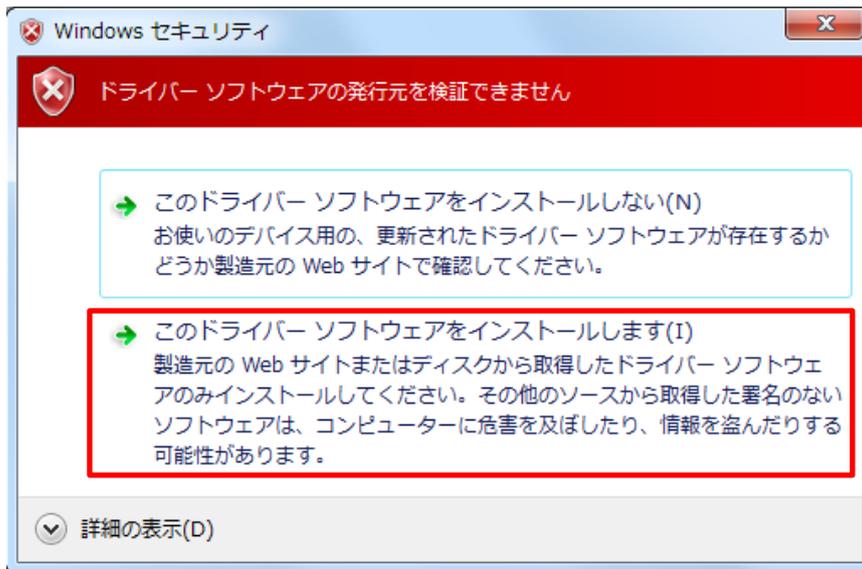


6. 「次へ」をクリックします。

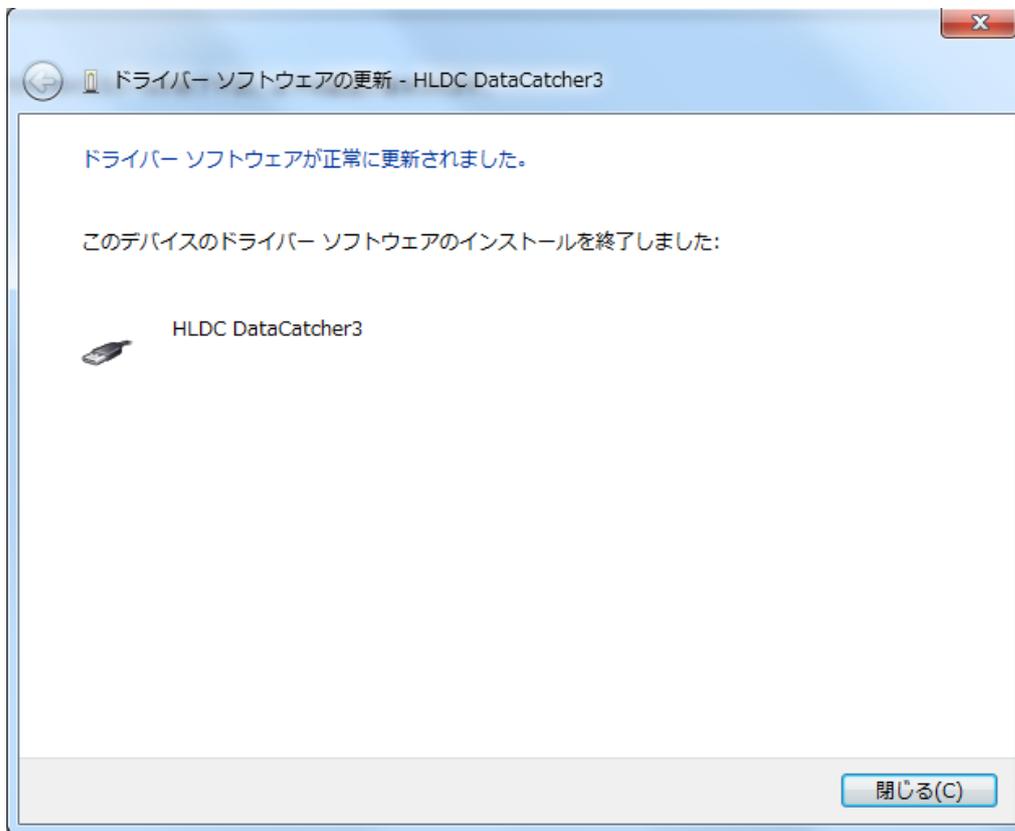


7. USBドライバのインストールが開始されます。

8. 以下警告が表示されたら、「このドライバーソフトウェアをインストールします」を選択します。



9. ドライバのインストールが終了すると、以下の画面が表示されます。



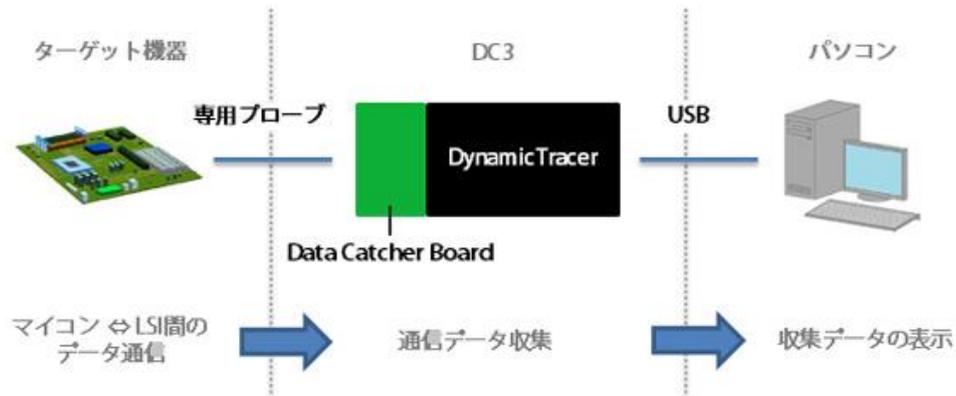
10. 「閉じる」をクリックし、終了します。

6-4:USBドライバのインストール方法(Windows 8/8.1, Windows10 : 32bit/64bit-OS)

Windows8・8.1/Windows10 の場合、Windows Update を経由して USB デバイスドライバが自動でインストールされます。

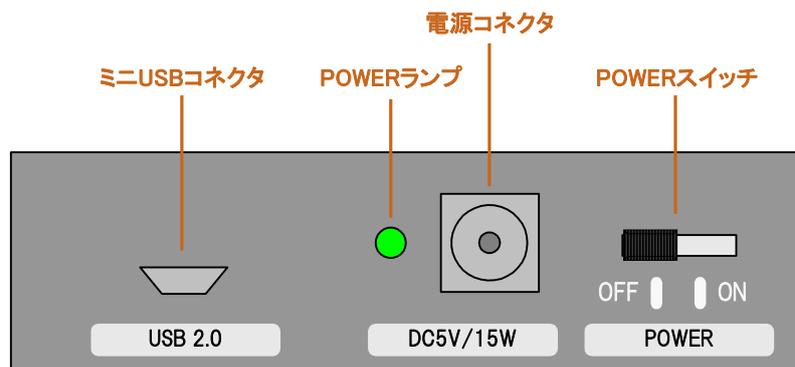
7. DC3 の構成

DC3 の本体側では、ターゲットの通信データの収集とパソコンへのアップロードを行います。パソコン側ではアップロードされたデータの加工、表示処理を行うというシステムになっています。



8. DC3 の概観

8-1: 背面

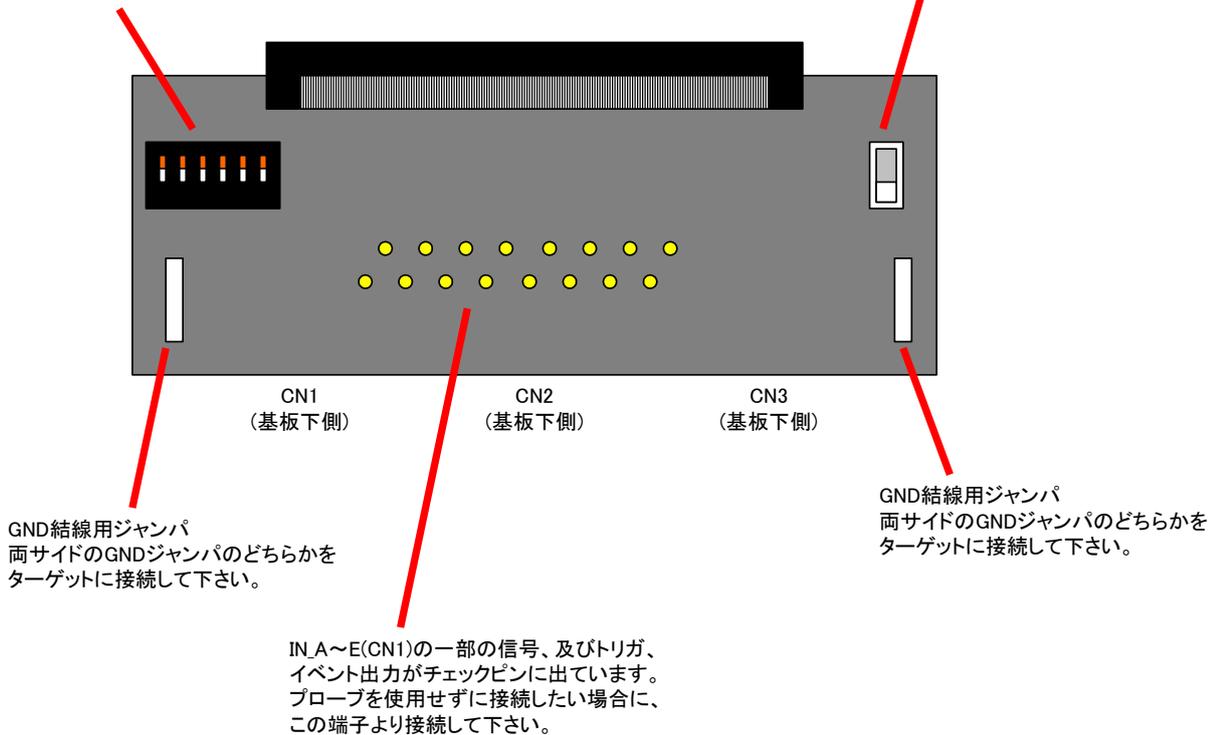


8-2: Data Catcher III I/F ボード

入力ポート選択用DIP SW9

IN_A~E(CN1)又はIN_AX~EX(CN2)のどちらのポートを使用するかを切り替えます。
※PCアプリの設定変更も必要

IN_AX~EX(CN2)を使用する場合の入力レベルを選択します。



8-3: 説明

GND 結線用ジャンパ	DC3 本体の GND 端子です。 ターゲットの GND と接続して下さい。
POWER ランプ	DC3 本体への電源投入により点灯します。
プローブコネクタ (CN1/2/3)	別売のプローブを使いターゲットの通信ラインと接続します。 接続のピンは通信によって異なります。 各通信のコネクター表を参照して下さい。
POWER スイッチ	DC3 本体の電源スイッチです。
ミニ USB コネクタ	付属の USB ケーブルを使い、パソコンと接続します。
電源コネクタ	付属の AC アダプタを接続します。

9. ハードウェアの取扱い

9-1:DC3 本体の立ち上げ手順

1. パソコンと接続

パソコンと DC3 本体を付属の USB ケーブルで接続します。

2. 電源投入

AC アダプタを差込み、DC3 本体_前面の POWER スイッチを ON にして電源を入れて下さい。

電源が入ると[POWER]ランプが緑色に点灯します。

これにより DC3 本体とパソコンとの接続準備が OK となります。

3. アプリケーション起動

PC アプリケーションを起動します。

起動方法については、後記の「10-1:DC3 の起動」を参照して下さい。

4. GND 端子を接続

別途用意頂いた GND 線を使い、DC3 本体の GND とターゲットの GND を接続します。

5. 入力ポートの設定

DIPスイッチを操作し、IN_A～IN_E 側を使用するか？IN_AX～IN_EX 側を使用するか？

を選択して切り替えてください。(PC アプリ側のポート設定も切り替える必要あり)

(通常は IN_A～IN_E を使用下さい。ノイズに強いポートです)

IN_AX～IN_EX ポートを選択した場合は、更にスレッショルドレベルを SW2 で選択して

下さい。電源電圧が 3.3V～5V 系については、3.3V を選択し、電源電圧が 1.8V の信号

については 1.8V 側を選択して下さい。

(どちらの設定でも入力電圧は 5V まで入力可能です)

6. 通信ラインを接続

別売りのプローブ等を使い、DC3 本体のプローブコネクタとターゲットの通信ラインを

接続して下さい。(基板上部のチェック端子からの接続も可能)

接続するピンは通信によって異なりますので、各通信のコネクタ表を参照して下さい。

9-2:注意事項

ハードウェアの取扱い上、次の事項に注意して下さい。

- ◎ **プローブを外した状態で電源を投入して下さい。**
- ◎ **DC3 本体の GND 端子とターゲットの GND ラインを必ず接続して下さい。**
- ◎ **長時間ロギングする場合などは、キャッチ動作中にハードディスクの電源が切れたり、システムスタンバイにならないように、パソコンの電源オプションを設定にしてください。**



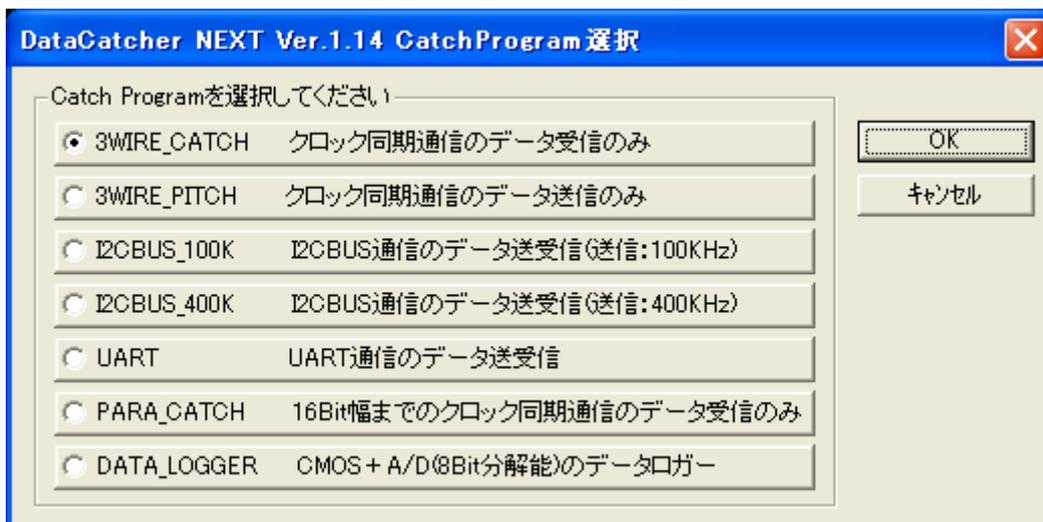
10. 基本操作

10-1:DC3 の起動

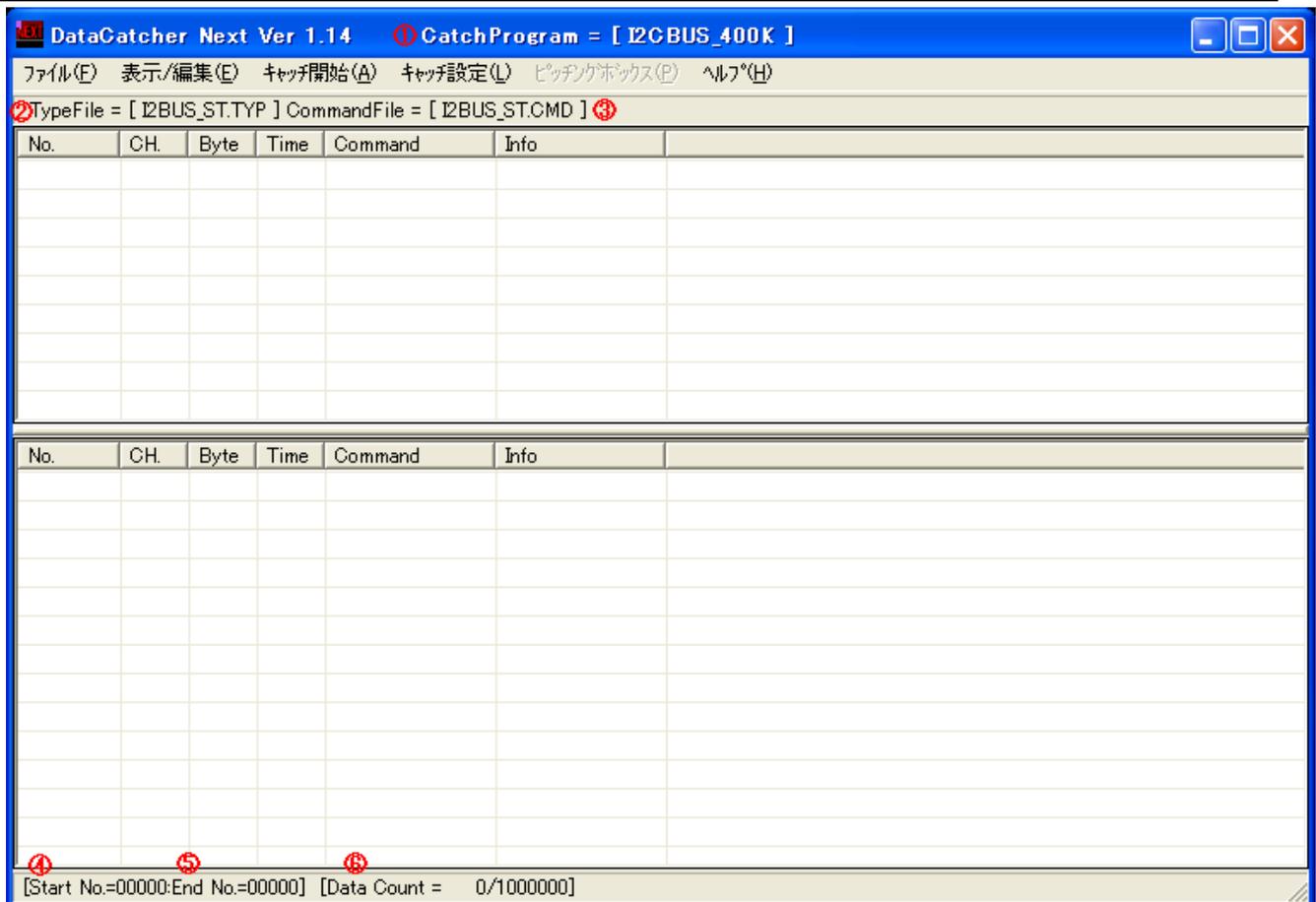
1. 「DataCatcherⅢ」のアイコンをダブルクリックするか、タスクバーの **<スタート>** メニューの **<プログラム(R)>** から **<DataCatcher3>** を選択してプログラムを起動して下さい。



2. [DataCathser3 CatchProgram 選択]が表示されたら、キャッチプログラムを選択します。CatchProgram のラジオボタンをクリックして選択し、[OK]を押すことでデータログ画面に切り替わります。
[キャンセル]を押すと CatchProgram:「3WIRE_CATCH」として、データログ画面に切り替わります。



10-2: データログ画面



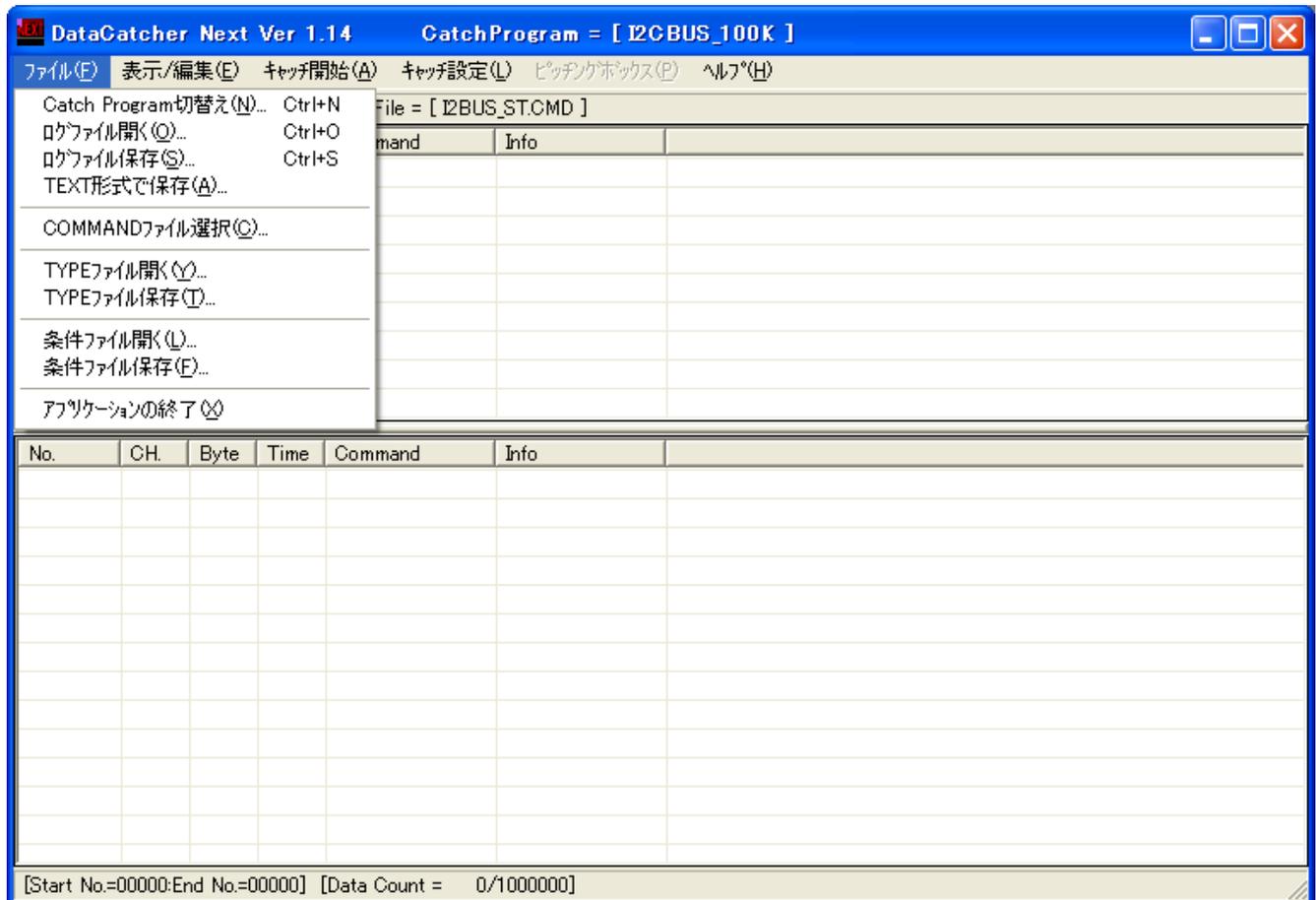
1. データログ画面が表示されます。

①	選択中のキャッチプログラム名を示します。
②	選択中のキャッチタイプのファイル名を示します。
③	選択中のコマンド解析用定義ファイル名を示します。
④	キャッチデータの開始 No.を示します。
⑤	キャッチデータの終了 No.を示します。
⑥	キャッチしたデータ数を示します。

※999999 件のデータをキャッチした後の画面表示についてはリング形式になっておりますので、Start No.が 000001 で End No.が 999999 の次は、Start No.が 000002, End No.が 000001 となり、その次は、Start No.が 000003, End No.が 000002 となります。

10-3:各種ファイル操作を行う

タイプファイル・コマンドファイルの読み出しや、ログ・タイプファイルの保存等を行います。



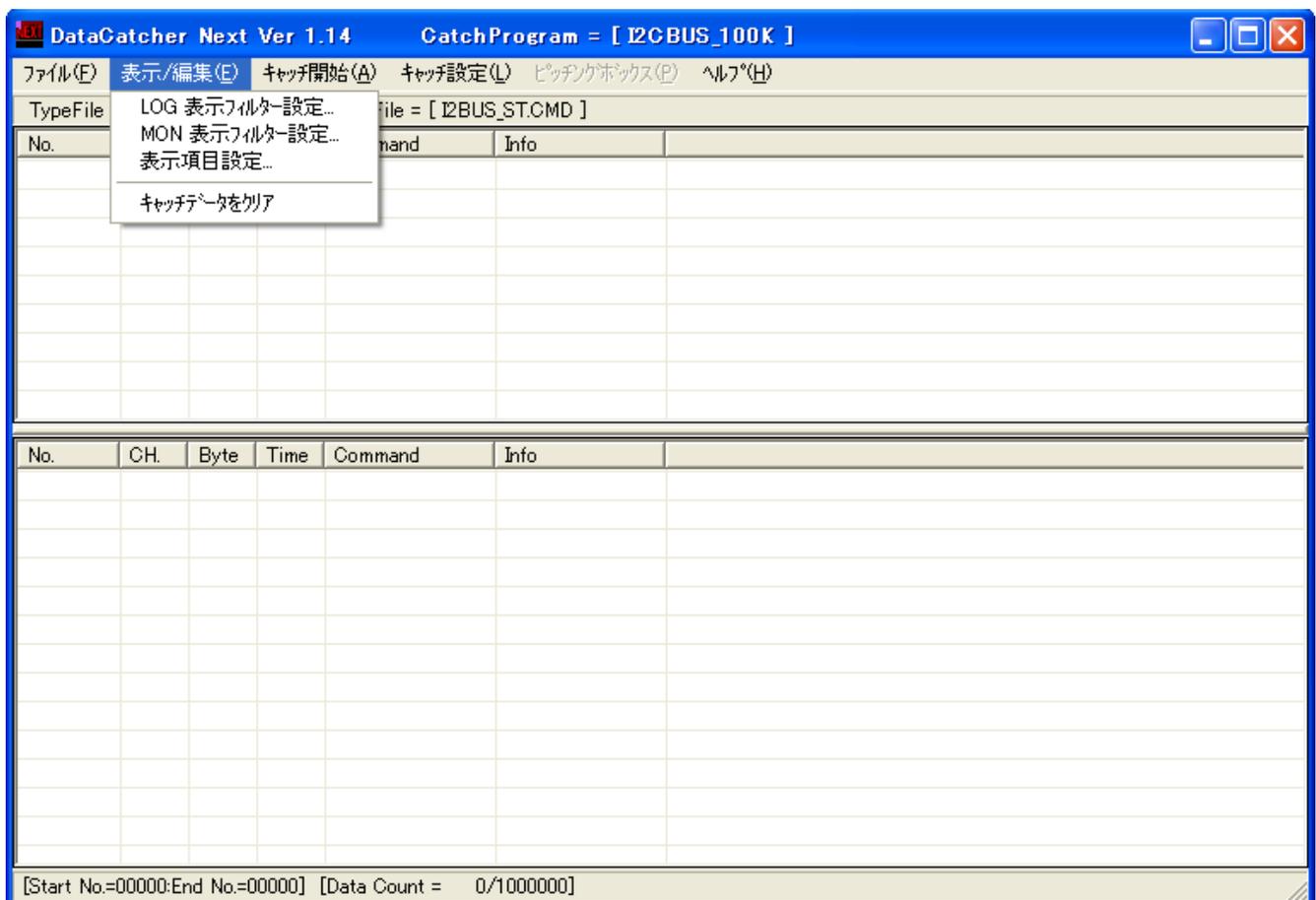
- ・Catch Program 切替え ... 現在ロードしているキャッチプログラムを、別の通信プロトコルの物に切替えます。
その際に、接続を解除し、キャッチデータもクリアされます。
- ・ログファイル開くlog 形式のログファイルを開きます。
開かれたデータは、保存した時点のイメージどおりに展開されます。
- ・ログファイル保存log 形式でキャッチデータをログファイルに保存します。

・TEXT 形式で保存txt 形式でファイルを保存します。
データとデータの間を”,”で区切った形式で保存しますので、
保存したデータは EXCEL などで編集可能です。
- ・COMMAND ファイル選択 ... 各 IC 用のコマンドファイルを読み出します。拡張子は.cmd です。
- ・TYPE ファイル開く ... キャッチタイプ設定が記述された、.typ 形式のファイルを読み出します。
- ・TYPE ファイル保存 ... キャッチタイプ設定ウィンドウにて編集された内容を、.typ 形式のファイルに書き出します。
キャッチタイプ設定ウィンドウについては、各キャッチプログラム用のマニュアルをご参照ください。

- ・条件ファイル開く ... 各種フィルター設定が記述された、.bit 形式のファイルを読み出します。
- ・条件ファイル保存 ... 表示フィルター・キャッチ条件・イベント出力条件・トリガー条件の各設定内容を.bit 形式のファイルに書き出します。
- ・アプリケーションの終了 ... DC3 を終了します。

10-4: 表示操作を行う

キャッチデータにフィルターを掛けたり、表示されているデータをクリアしたりします。

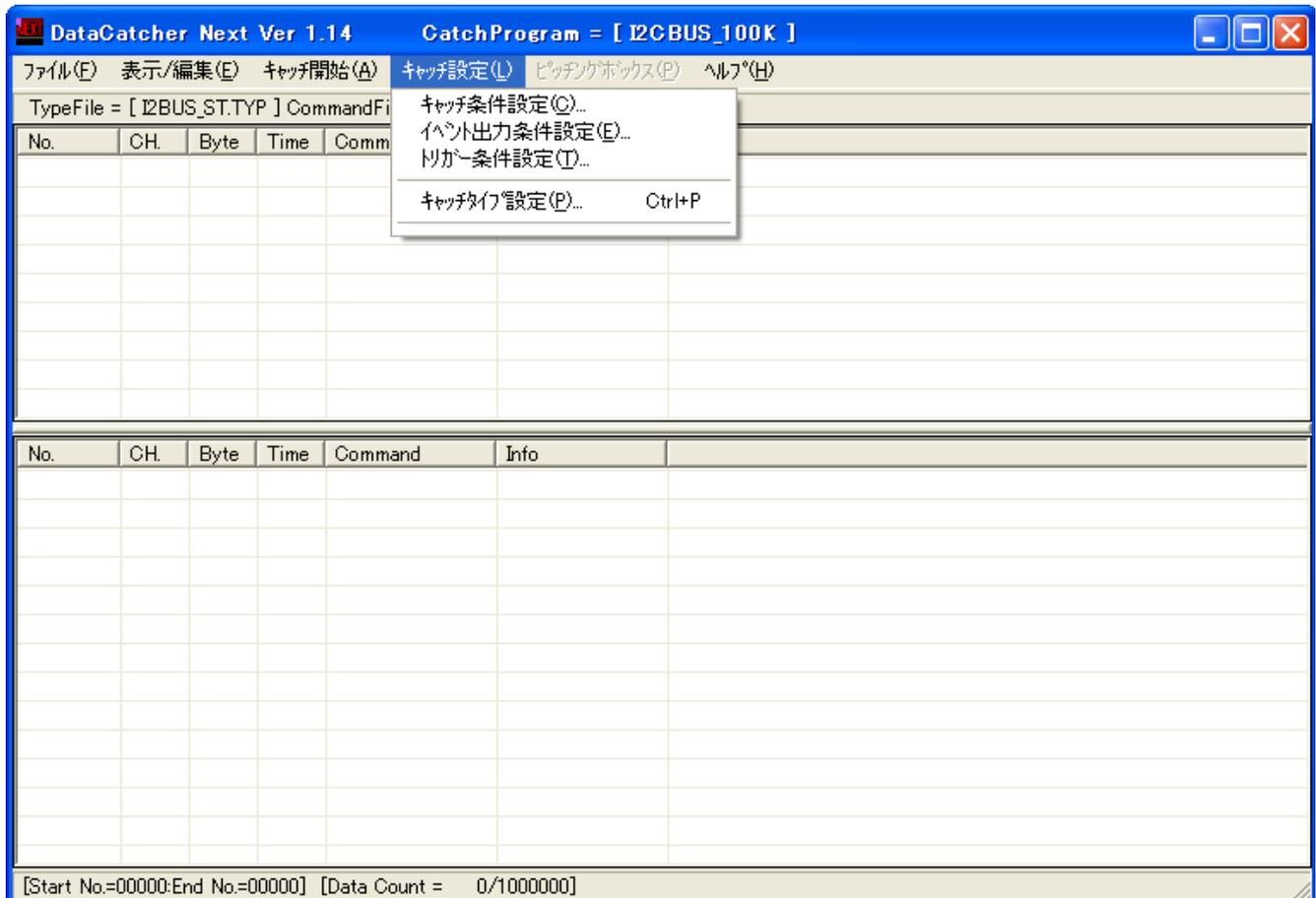


- ・LOG 表示フィルター設定 ... LOG 表示されるデータの内容を制限する、表示フィルターを設定します。
この項目を選択すると、LOG 表示フィルター設定ウィンドウが開きます。
- ・MON 表示フィルター設定 ... MONITOR 表示されるデータの内容を制限する、表示フィルターを設定します。
この項目を選択すると、MON 表示フィルター設定ウィンドウが開きます。
- ・表示項目設定 ... 表示されるデータの項目数や、形式を指定します。
この項目を選択すると、表示項目設定ウィンドウが開きます。

- ・キャッチデータをクリア ... 画面上に表示されているキャッチデータをクリアします。

10-5:キャッチデータに条件を設定する

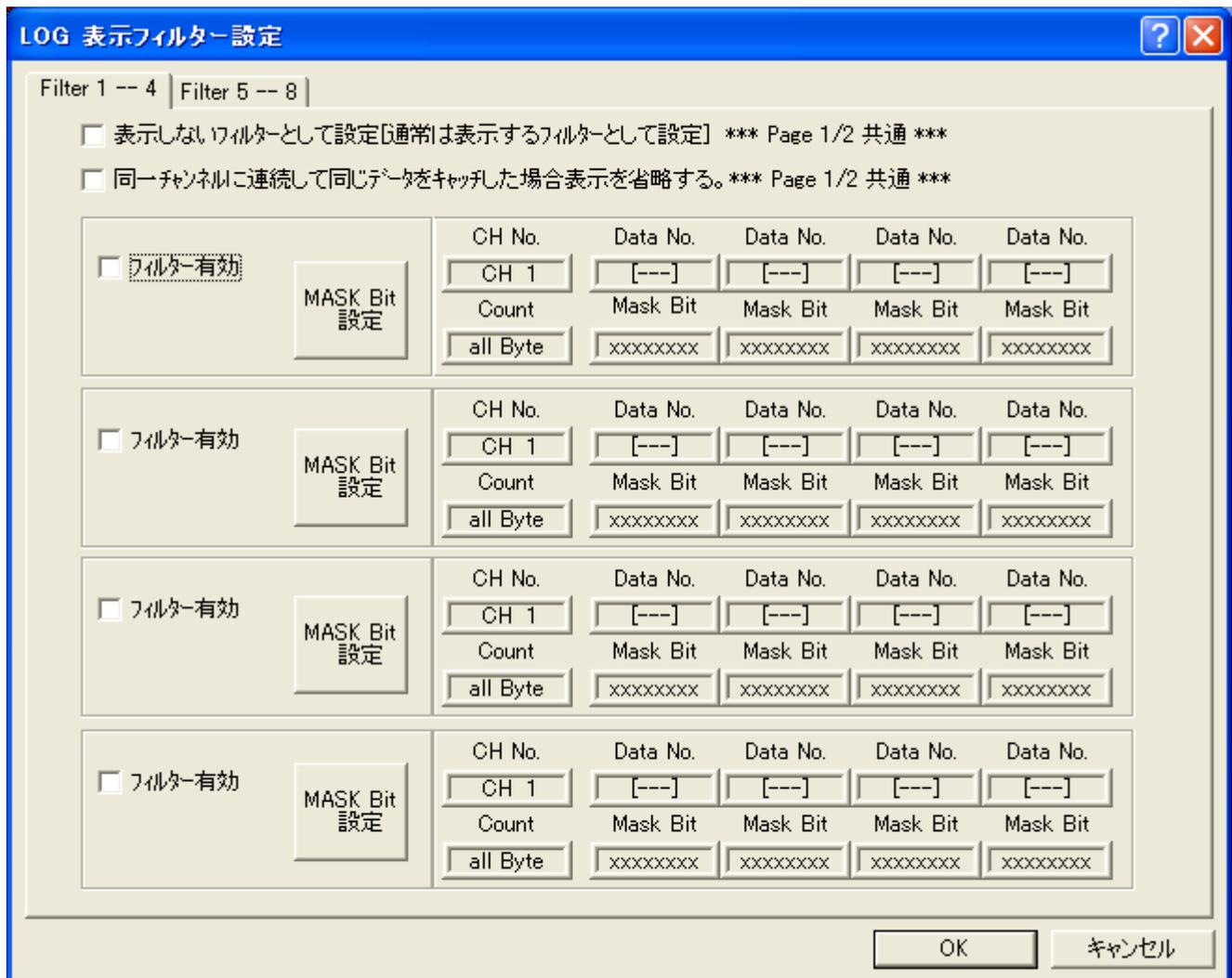
条件に一致したデータのみキャッチするようになり、キャッチしたデータの内容を判断したりしてイベント出力を行ったりします。



- ・キャッチ条件設定 ... キャッチするデータに条件を設定し、条件に一致したデータのみキャッチするようになり、また逆の動作をさせたりする事が出来ます。この項目を選択すると、キャッチフィルター設定ウィンドウが開きます。
- ・イベント出力条件設定 ... 条件に一致したデータをキャッチしたときに、イベント出力ラインからパルスを出力することができます。この項目を選択すると、イベントフィルター設定ウィンドウが開きます。
- ・トリガー条件設定 ... 条件に一致したデータをキャッチしたときに、データのキャッチ開始・キャッチ終了を行うトリガーを設定することができます。この項目を選択すると、トリガー条件設定ウィンドウが開きます。
- ・キャッチタイプ設定 ... 各キャッチプログラム用のマニュアルをご参照ください。

10-6: LOG表示フィルター設定ウィンドウ

LOG表示フィルター設定を選択した場合は、このウィンドウが開きます。



【各設定内容について】

- ・ 表示しないフィルターとして設定 [通常は表示するフィルターとして設定] チェックボックス
 - 表示フィルターを、設定した条件と一致したデータを表示させないフィルターとして使用する場合にチェックします。
OFFの場合は、条件と一致したデータのみ表示するフィルターとなります。
 - このチェックボックスは、全てのフィルター共通の設定となります。
- ・ 同一チャンネルに連続して同じデータをキャッチした場合表示を省略する。チェックボックス
 - ONすると2回以上同じデータをキャッチした場合、2回目以降のデータは表示なくなります。
OFFの場合は、同じデータが連続しても全てそのまま連続で表示されます。
 - このチェックボックスは、全てのフィルター共通の設定となります。
- ・ フィルター有効チェックボックス
 - チェックされたフィルター条件のみを有効とします。
チェックされていない条件については適用されません。
- ・ MASK Bit 設定ボタン
 - このボタンを押すと、共通マスクフィルター設定ウィンドウが開きます。

10-7: キャッチフィルター設定ウィンドウ

キャッチフィルター設定を選択した場合は、このウィンドウが開きます。

キャッチフィルター設定

Filter 1 -- 4 | Filter 5 -- 8

キャッチしないフィルターとして設定[通常はキャッチするフィルターとして設定] *** Page 1/2 共通 ***

<input type="checkbox"/> フィルター有効	MASK Bit 設定	CH No.	Data No.	Data No.	Data No.	Data No.
		CH 1	[---]	[---]	[---]	[---]
		Count	Mask Bit	Mask Bit	Mask Bit	Mask Bit
		all Byte	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

<input type="checkbox"/> フィルター有効	MASK Bit 設定	CH No.	Data No.	Data No.	Data No.	Data No.
		CH 1	[---]	[---]	[---]	[---]
		Count	Mask Bit	Mask Bit	Mask Bit	Mask Bit
		all Byte	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

<input type="checkbox"/> フィルター有効	MASK Bit 設定	CH No.	Data No.	Data No.	Data No.	Data No.
		CH 1	[---]	[---]	[---]	[---]
		Count	Mask Bit	Mask Bit	Mask Bit	Mask Bit
		all Byte	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

<input type="checkbox"/> フィルター有効	MASK Bit 設定	CH No.	Data No.	Data No.	Data No.	Data No.
		CH 1	[---]	[---]	[---]	[---]
		Count	Mask Bit	Mask Bit	Mask Bit	Mask Bit
		all Byte	xxxxxxxx	xxxxxxxx	xxxxxxxx	xxxxxxxx

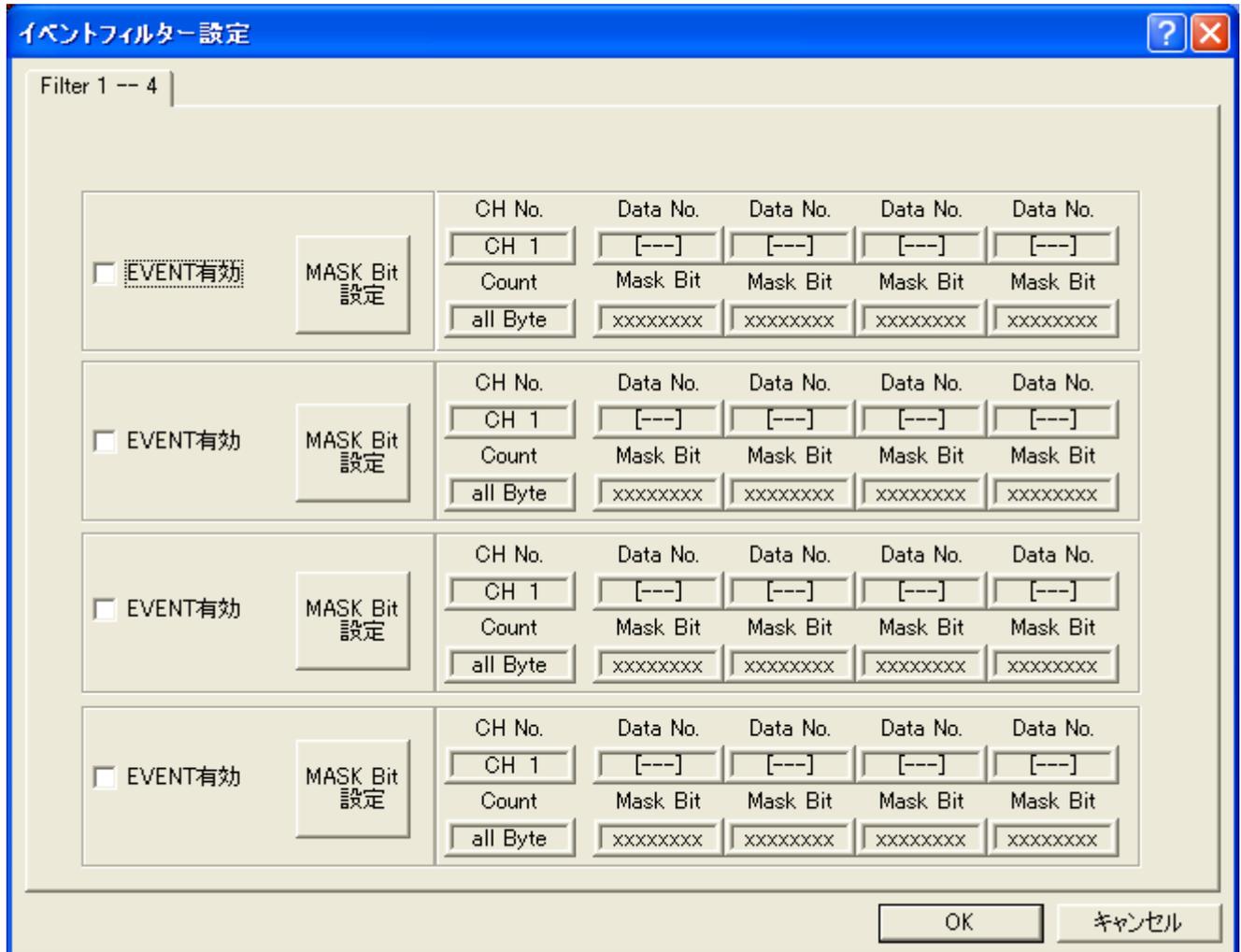
OK キャンセル

【各設定内容について】

- キャッチしないフィルターとして設定[通常はキャッチするフィルターとして設定]チェックボックス
 - キャッチフィルターを、設定した条件と一致したデータを表示させないフィルターとして使用する場合にチェックします。
OFFの場合は、条件と一致したデータのみキャッチするフィルターとなります。
 - このチェックボックスは、全てのフィルター共通の設定となります。
- フィルター有効チェックボックス
 - チェックされたフィルター条件のみを有効とします。
チェックされていない条件については適用されません。
- MASK Bit 設定ボタン
 - このボタンを押すと、共通マスクフィルター設定ウィンドウが開きます。

10-8: イベントフィルター設定ウィンドウ

イベント出力条件を選択すると、このウィンドウが開きます。

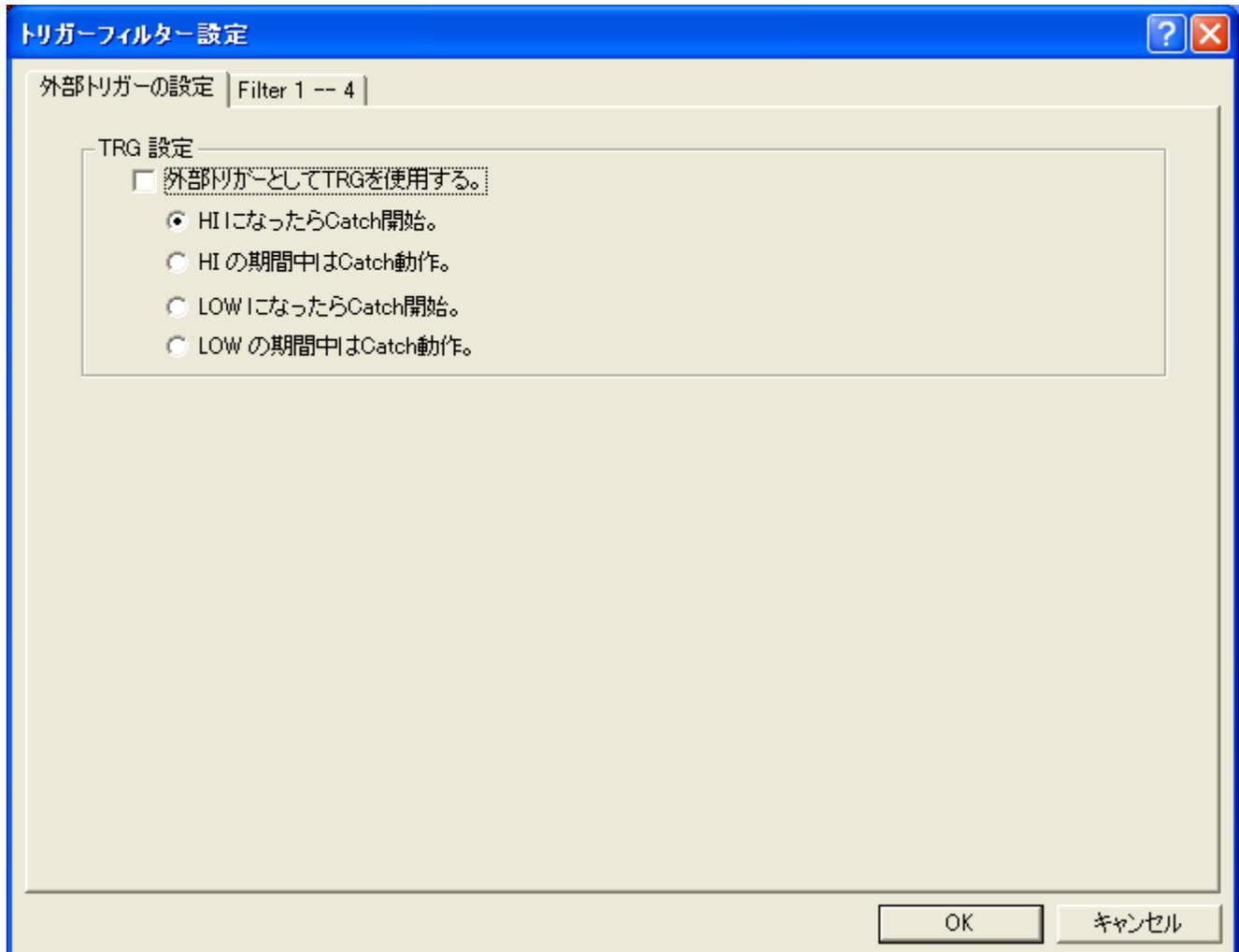


- EVENT 有効チェックボックス
 - チェックされたフィルター条件のみを有効とします。
チェックされていない条件については適用されません。
データをキャッチすると EVENT 端子が 0V になり、データを確認してフィルターと一致したデータをキャッチすると EVENT 端子が 0V から 3.3V に立ち上がります。
- MASK Bit 設定
 - このボタンを押すと、共通マスクフィルター設定ウィンドウが開きます。

10-9:トリガーフィルター設定ウィンドウ

トリガー条件設定を選択すると、以下のウィンドウが開きます。

- Page1 -



- ・ 「外部トリガーとして TRG を使用する」チェックボックス
 - 外部入力端子を使用するかを指定します。
 - ◇ HI になったら Catch 開始ラジオボタン
 - 選択された外部入力端子の状態が HI になったのを感知し、Catch 動作を開始します。
 - ◇ HI の期間中は Catch 動作ラジオボタン
 - 選択された外部入力端子の状態が HI になっている間のみ Catch 動作を行います。
 - ◇ LOW になったら Catch 開始ラジオボタン
 - 選択された外部入力端子の状態が LOW になったのを感知し、Catch 動作を開始します。
 - ◇ LOW の期間中は Catch 動作ラジオボタン
 - 選択された外部入力端子の状態が LOW になっている間のみ Catch 動作を行います。

トリガーフィルター設定

外部トリガーの設定 Filter 1 -- 4

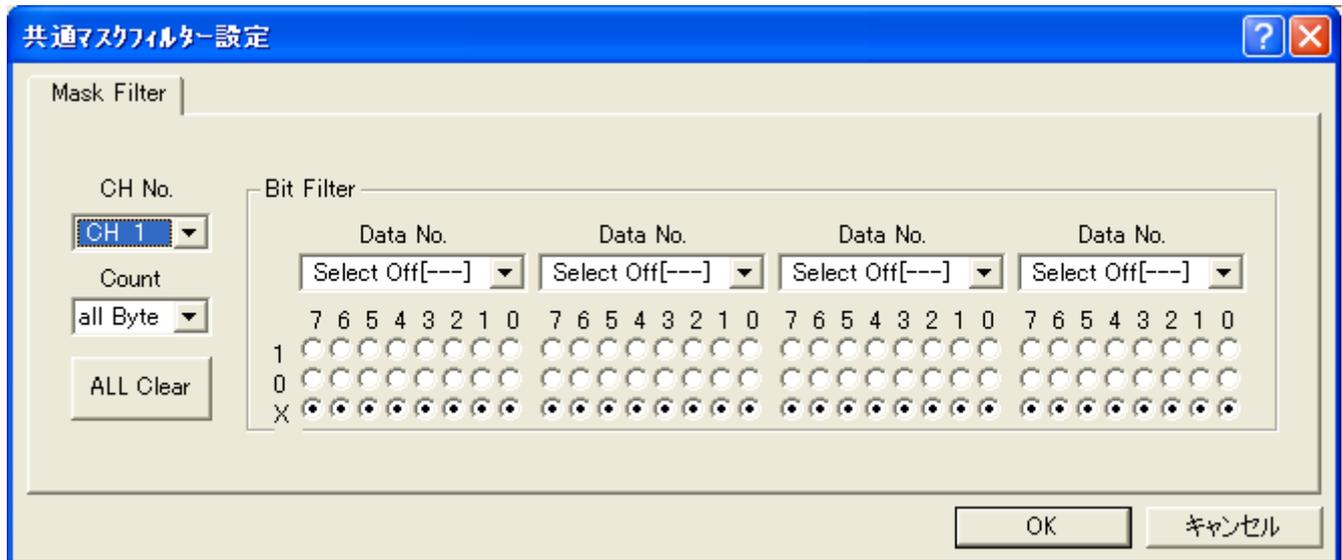
<input type="checkbox"/> Catch開始有効 <input type="checkbox"/> Catch終了有効 MASK Bit 設定	CH No. CH 1 Data No. [---] Data No. [---] Data No. [---] Data No. [---] Count Mask Bit Mask Bit Mask Bit Mask Bit all Byte xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
<input type="checkbox"/> Catch開始有効 <input type="checkbox"/> Catch終了有効 MASK Bit 設定	CH No. CH 1 Data No. [---] Data No. [---] Data No. [---] Data No. [---] Count Mask Bit Mask Bit Mask Bit Mask Bit all Byte xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
<input type="checkbox"/> Catch開始有効 <input type="checkbox"/> Catch終了有効 MASK Bit 設定	CH No. CH 1 Data No. [---] Data No. [---] Data No. [---] Data No. [---] Count Mask Bit Mask Bit Mask Bit Mask Bit all Byte xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx
<input type="checkbox"/> Catch開始有効 <input type="checkbox"/> Catch終了有効 MASK Bit 設定	CH No. CH 1 Data No. [---] Data No. [---] Data No. [---] Data No. [---] Count Mask Bit Mask Bit Mask Bit Mask Bit all Byte xxxxxxxx xxxxxxxx xxxxxxxx xxxxxxxx

OK キャンセル

- Catch 開始有効チェックボックス
 - このチェックボックスを ON にすると、Catch を開始してから設定した条件に一致するまでのデータは無効化され、一致するデータが現れた時に初めてデータが表示されます。
- Catch 終了有効チェックボックス
 - このチェックボックスを ON にすると、設定した条件に一致するデータが現れた時に、Catch 動作を終了します。
- MASK Bit 設定
 - このボタンを押すと、共通マスクフィルター設定ウィンドウが開きます。

10-10:共通マスクフィルター設定ウィンドウ

各種フィルター設定ウィンドウにある MASK Bit 設定ボタンをクリックすることで、このウィンドウが表示されます。

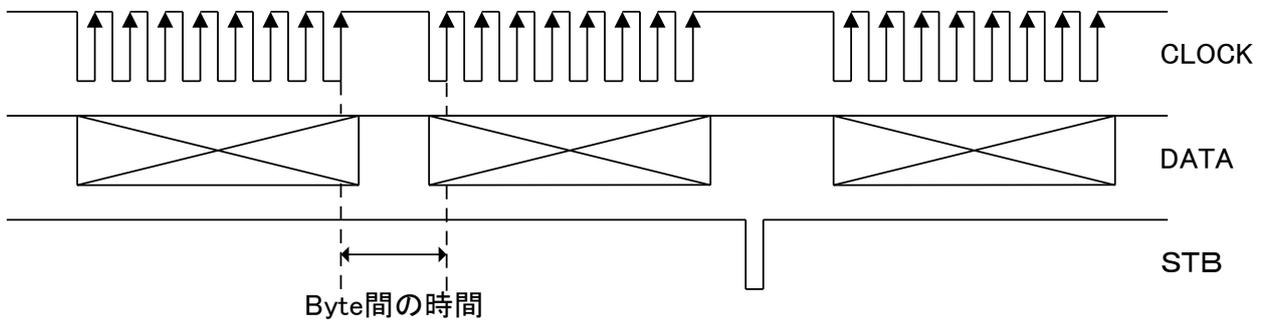


- ・ CH No.ドロップダウンリスト
 - マスクフィルターを適用するチャンネルを指定します。
- ・ Countドロップダウンリスト
 - 条件とするデータの長さを Byte または Bit で指定します。
単位は各キャッチプログラムによって異なります。
- ・ ALL Clear ボタン
 - 設定した条件を全て初期状態に戻します。
- ・ Bit Filter フィールド
 - フィルターの条件として 4 バイト分まで、ビット単位でデータを設定することができます。
 - ◇ Data No.ドロップダウンメニュー
 - フィルターの条件として指定するデータが、何バイト目かを指定します。
 - ◇ ビット指定用ラジオボタン
 - 条件となるデータを 1 ビットごとに指定できます。
X は 1/0 どちらにも当てはまらない位置にお使いください。

11. [3WIRE CATCH]Catch Program でキャッチできる波形

11-1:3線シリアル通信の波形

以下の「3線シリアル通信」に対応しています。

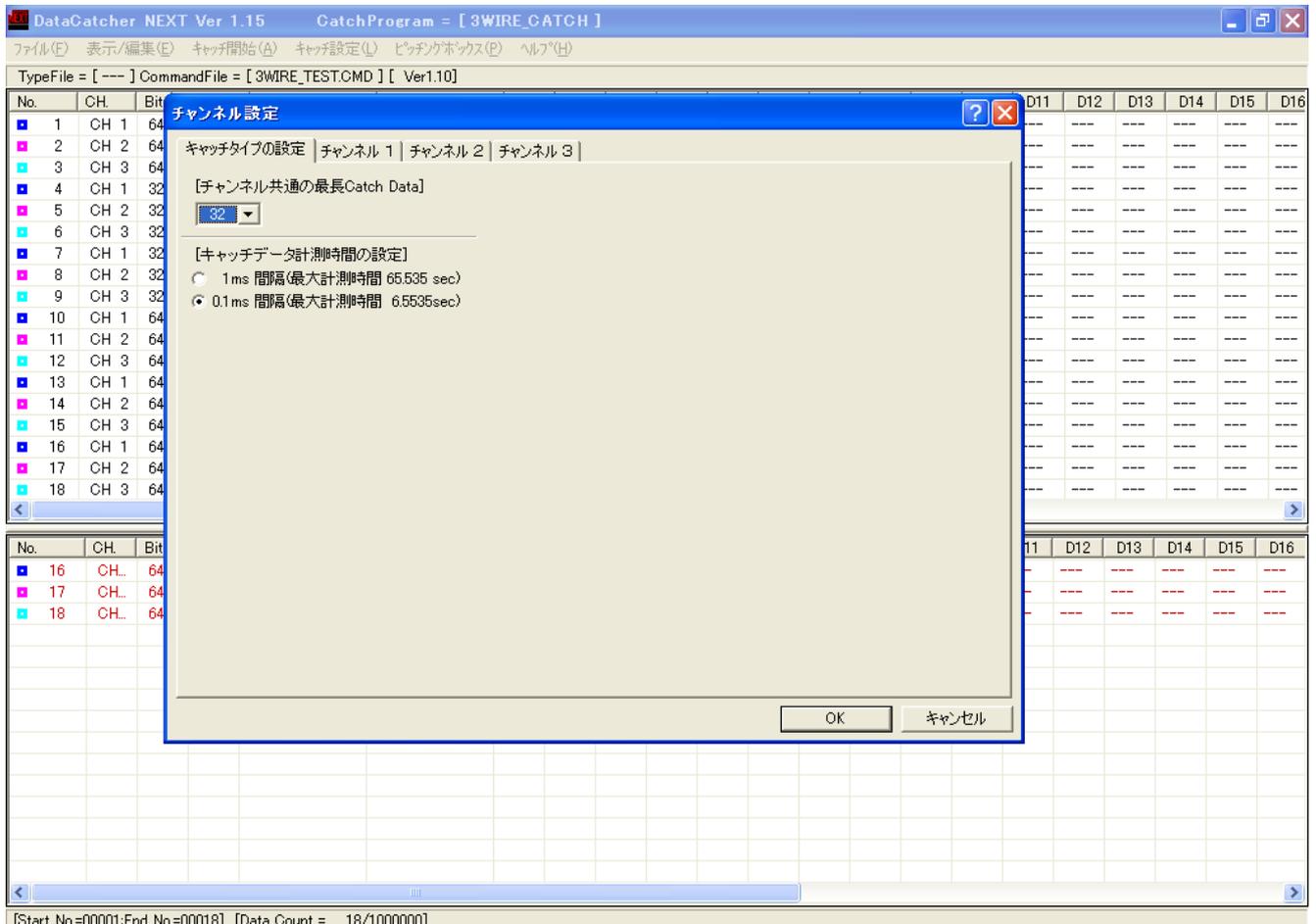


- 上記の様に一般的な「3線同期シリアル通信」には、STB信号がありますが、STB信号が無く、以下の様な形式にて通信パケットを確定するものについても対応しております。

- ◇ Byte 間の時間にて通信パケットを確定する形式
- ◇ 通信データの中に、通信パケットの長さを指定するデータがある形式
- ◇ 通信パケットの終了を指示するデータがある形式
- ◇ 通信パケットの長さを確定するデータがある形式

12. [3WIRE CATCH]Catch Program の設定について

[キャッチ設定(L)]-[キャッチタイプ設定(O)]をクリックすると、以下の〈チャンネル設定〉ダイアログボックスが開きますが、この設定を変更する事により、各種 3 線シリアル通信のキャッチが可能になります。以下にその設定内容について説明をします。



12-1:〈キャッチタイプの設定〉タグでの設定内容について

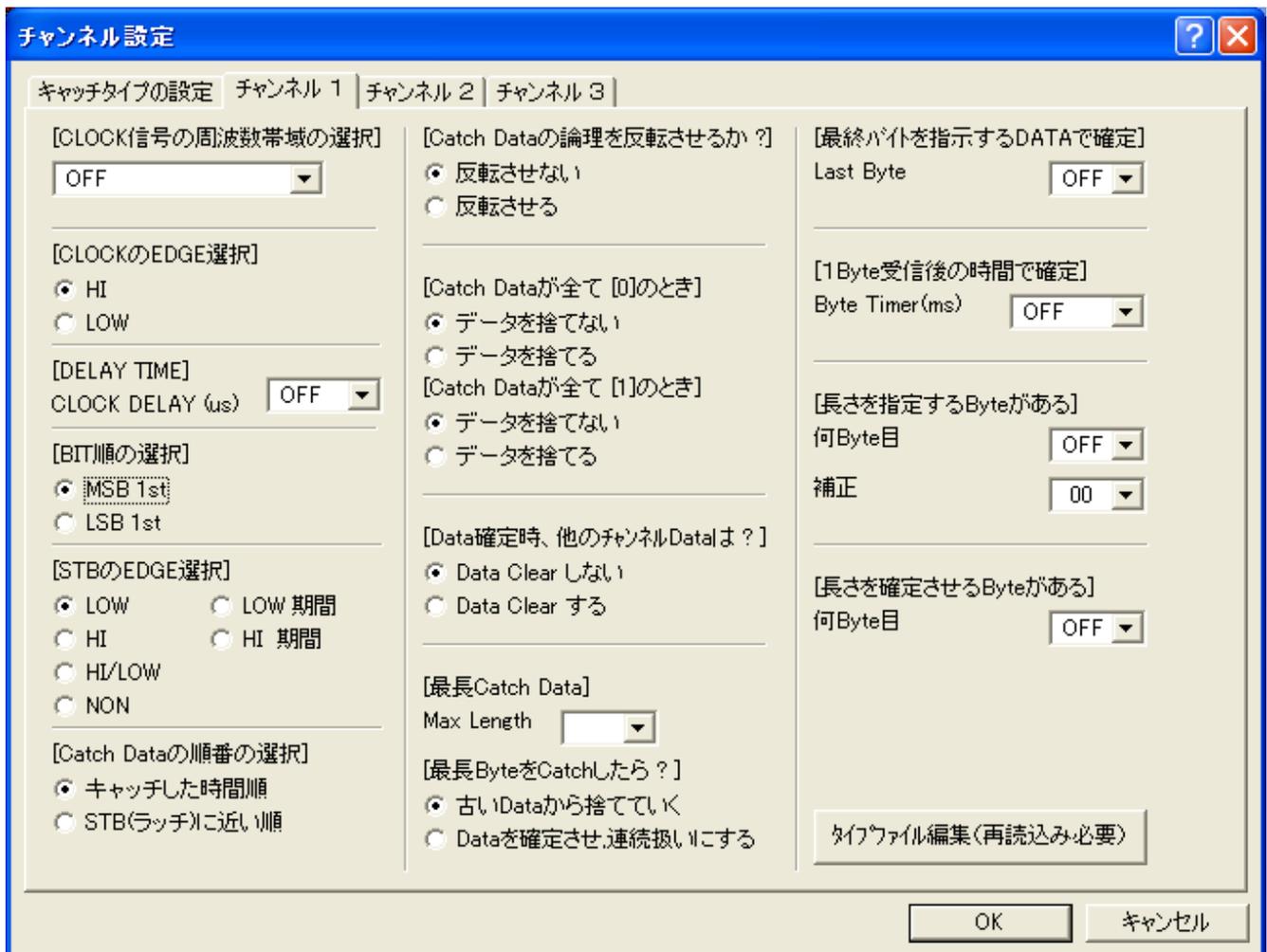
【チャンネル共通の最長 Catch Data】

<Max Length>

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。尚、MAX 値は 32 となっております。

【キャッチデータ計測時間の設定】

本システムでは、現在通信パケットのデータ確定から次のデータ確定迄の時間を測定しそれを画面に表示しますが、ここではその測定時間の分解能について設定します



12-2:〈チャンネル1〉〈チャンネル2〉〈チャンネル3〉タグでの設定内容について

【CLOCK の EDGE 選択】

キャッチする CLOCK 信号の DATA を取り込むエッジを選択して下さい。

【BIT 順の選択】

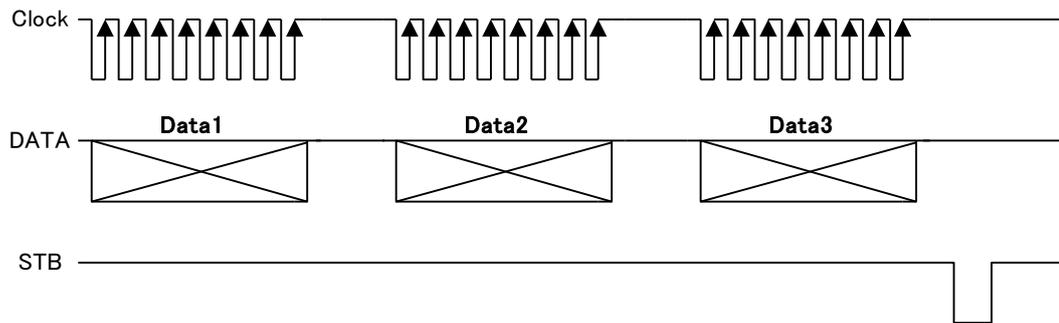
シリアル伝送方法を選択します。

【STB の EDGE 選択】

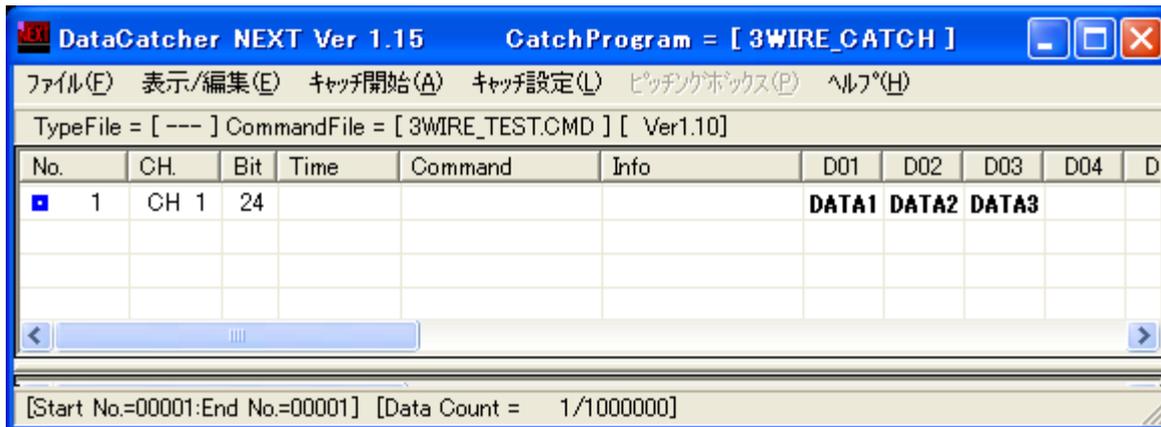
ラッチ信号の EDGE の選択をします。

【Catch Data 順番の選択】

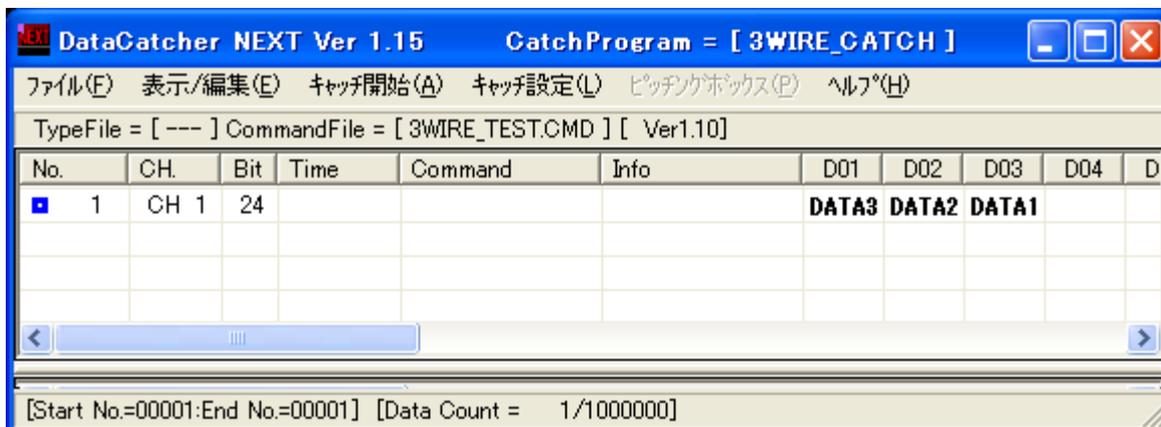
キャッチしたデータに対する表示順番を指定します。



[キャッチした時間順]のオプションボタンをオンすると以下のようになります。



[STB(ラッチ)に近い順]のオプションボタンをオンすると以下のようになります。



【Catch Data が全て[0]のとき】

STB(ラッチ)迄の全 bit が 0 であった場合の処理方法について設定します。

【Catch Data が全て[1]のとき】

STB(ラッチ)迄の全 bit が 1 であった場合の処理方法について設定します。

【Data 確定時、他のチャンネル Data は?】

各チャンネルが独立してデータの取り込みを行っている場合で、あるチャンネルが STB によりデータが確定されますが、その時の他チャンネルでのキャッチデータの扱いについて設定します。

※制限事項を確認の上、使用してください。

【最長 Catch Data】

<Max Length>

一度にキャッチ可能なデータ数を指定します。

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。

尚、MAX 値は 32 となっております。

※前頁の“チャンネル共通の最長 Catch Data”より短く設定して下さい。

【最長 Byte を Catch したら？】

STB が受信しないまま、上記で設定した最長キャッチデータ数に達した場合の処理方法について設定します。

○ [古い Data から捨てていく] オプションボタンをオンした場合

STB が受信される迄、古いデータから順に捨てていき、STB にてデータを確定する方法です。

○ [データを確定させ、連続扱いにする]オプションボタンをオンした場合

STB が受信されるまで、全てのデータを取り込み、ストップ・コンディションにてデータを確定する方法です。

尚、通信パケット全体のバイト数が、指定した最長バイトになったら一度パソコンにデータを表示し、その後は、連続チャンネルとして表示します。

以下の項目については、一つの通信パケットを確定する方法について設定するものです。

【最終バイトを指示する DATA で確定】

最終バイトを指示するデータで確定する場合、その内容を 0x00h～0xffh で指定し、特に指示するデータが無い場合は OFF として下さい。

【1Byte 受信後の時間で確定】

1通信パケットの確定を 1Byte 受信後の時間で決定します。設定値は 001～100 の間で選択が可能で、単位は msec です。特に指示するデータが無い場合は OFF として下さい。

【長さを指定する Byte がある】

通信パケットの長さを指定するデータが通信パケットにある時に設定します。

- ・ 何 Byte 目 : 通信パケット内の何 Byte 目にあるかを指定します。

01～32 で選択します。

特に指示するデータが無い場合は OFF として下さい。

- ・ 補 正 : 例えば STX や ETX が通信パケットに含まれている場合で、上記データがそれを含まない場合に、ここで含まない分について補正します。

01～32 で選択します。

特に指示するデータが無い場合は 00 として下さい。

【長さを確定させる Byte がある】

通信パケットの長さを確定するデータがある場合、ここでそのデータが何 Byte 目にあるかを指定します。01～32 で選択します。特に指示するデータが無い場合はOFFとして下さい。

尚、このデータに対応する通信パケットの長さについては、.typ ファイルにて次ページの様に対応させます。(以下の例はチャンネル1についての場合です。)

例えば、ここで指定した××Byte 目のデータが 0x12h で、通信パケットの Byte 数を 16Byte と設定する場合は以下の様になります。

— 実際のファイル内容 —

```

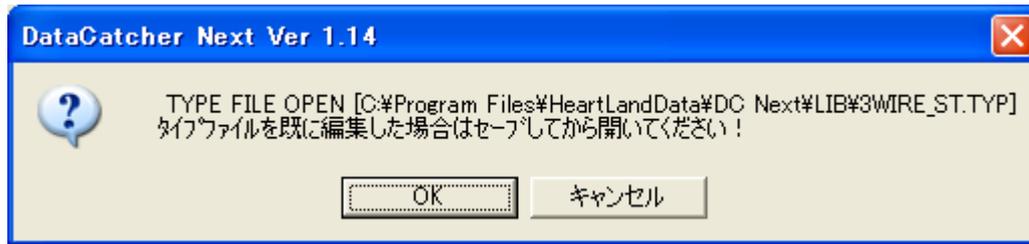
CH1_COMMAND_STB=00
; : コマンドによってパケットの長さを確定する。バイト数の設定値
; : 無効なコマンド 00
; : 有効なコマンド 01 -- ff (1 バイトから 255 バイトまで)
; :
; :                               CHI_LENGTH を超えたら一度データを上げます
; : 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
$00=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$10=00, 00, 10, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$20=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$30=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$40=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$50=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$60=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$70=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$80=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$90=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$a0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$b0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$c0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$d0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$e0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00
$f0=00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00, 00

```

【タイプファイル編集】

タイプファイルをメモ帳で開きます。タイプファイルを直接編集したい場合に利用してください。なお、チャンネル設定ウィンドウで変更した内容は、タイプファイルを保存するまでファイルに書き込まれませんので、既に変更をされた場合は、一度「OK」ボタンをクリックし、「ファイル」メニューより「TYPE ファイルを保存」を選んで保存した後、再度タイプファイル編集を行ってください。

このボタンを押すと、以下のインフォメーションウィンドウが開かれます。



12-3:ポート設定

[キャッチ設定(L)]-[ポート設定(M)]をクリックすると、DataCatcher III I/F Board のポート設定メニューが表示されます。画面の説明に従い、ボード本体の DIP SW と同様の設定を行ってください。

12-4:コネクター表

20pin コネクタピン配列(コネクタ挿入口側から見た図)

2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

CN1 / CN2 (コネクタ名は基板裏側に記載)

	端子名		信号名
1	IN_A[0]	IN_AX[0]	CH1 CLK 入力
2	IN_A[1]	IN_AX[1]	CH1 DATA 入力
3	IN_A[2]	IN_AX[2]	CH1 STB 入力
4	IN_A[3]	IN_AX[3]	
5	IN_B[0]	IN_BX[0]	CH2 CLK 入力
6	IN_B[0]	IN_BX[1]	CH2 DATA 入力
7	IN_B[2]	IN_BX[2]	CH2 STB 入力
8	IN_B[3]	IN_BX[3]	
9	IN_C[0]	IN_CX[0]	CH3 CLK 入力
10	IN_C[1]	IN_CX[1]	CH3 DATA 入力
11	IN_C[2]	IN_CX[2]	CH3 STB 入力

CN3

	端子名	信号名
1	TRIG[0]	外部トリガー入力
2	TRIG[1]	
3	EVENT[0]	イベントフィルターにて設定した EVENT 出力
4	EVENT[1]	

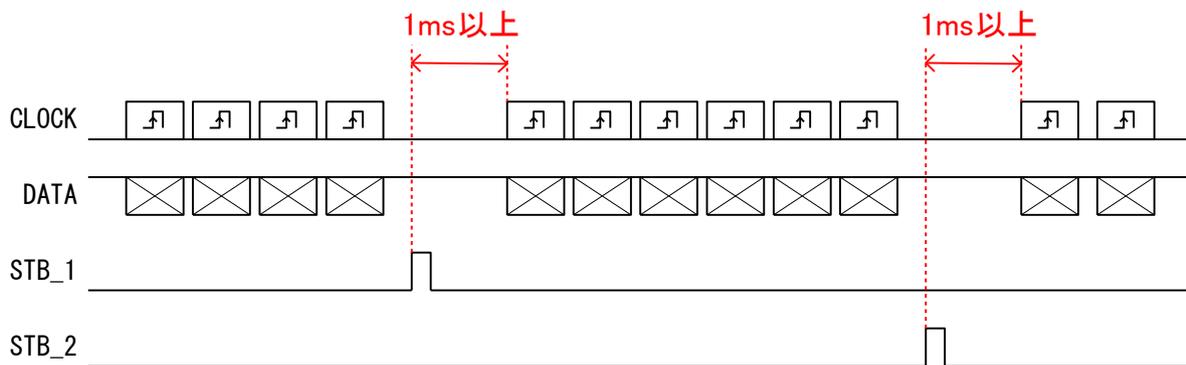
- 【注意】**・GND ターミナルとキャッチする機器の GND は必ず接続して下さい。
 尚、接続してもキャッチデータが不安定な場合はアースを取って下さい。
 ・必ず DC3 の電源を投入してからプローブを接続して下さい。
 故障の原因となります。
 ・複数チャンネルを同時にキャッチする際、CLOCK が共通でも各チャンネルに CLOCK の接続が必要です。

12-5:制限事項

受信可能なクロックの最高周波数

20MHz

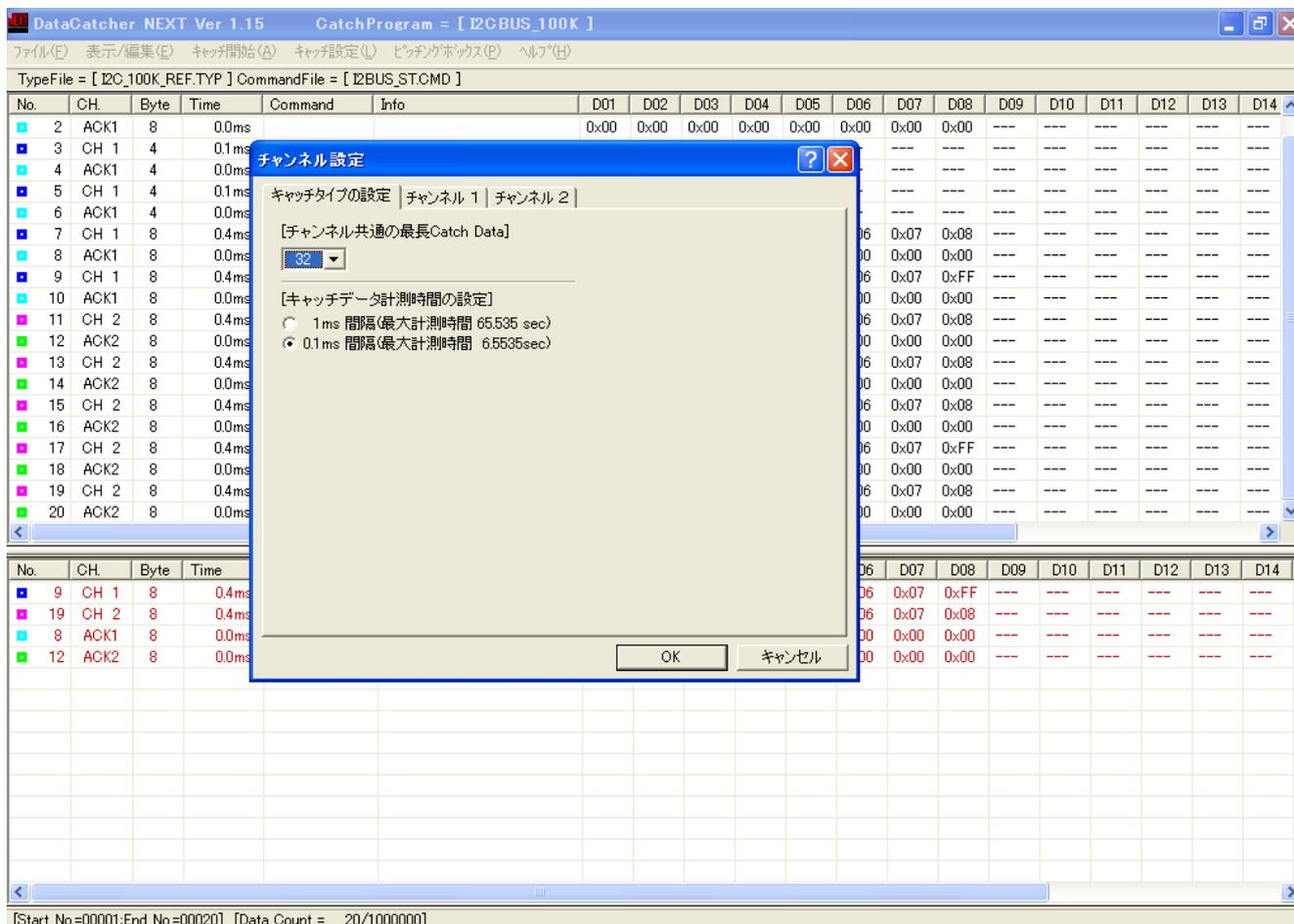
◆Data を確定した時に、他のチャンネルデータをクリアする機能について。



- ・上記の形式の通信において、この機能を使用するには、STB でパケット確定してから、次のクロックが発生するまでの時間が 1ms 以上 必要になります。

13. [I2CBUS_400K]Catch Program の設定について

[キャッチ設定(L)]-[キャッチタイプ設定(P)]をクリックすると、以下の<チャンネル設定>ダイアログボックスが開き、I2C通信における設定を行います。以下にその設定内容について説明をします。



13-1: <キャッチタイプの設定>タグでの設定内容について

【チャンネル共通の最長 Catch Data】

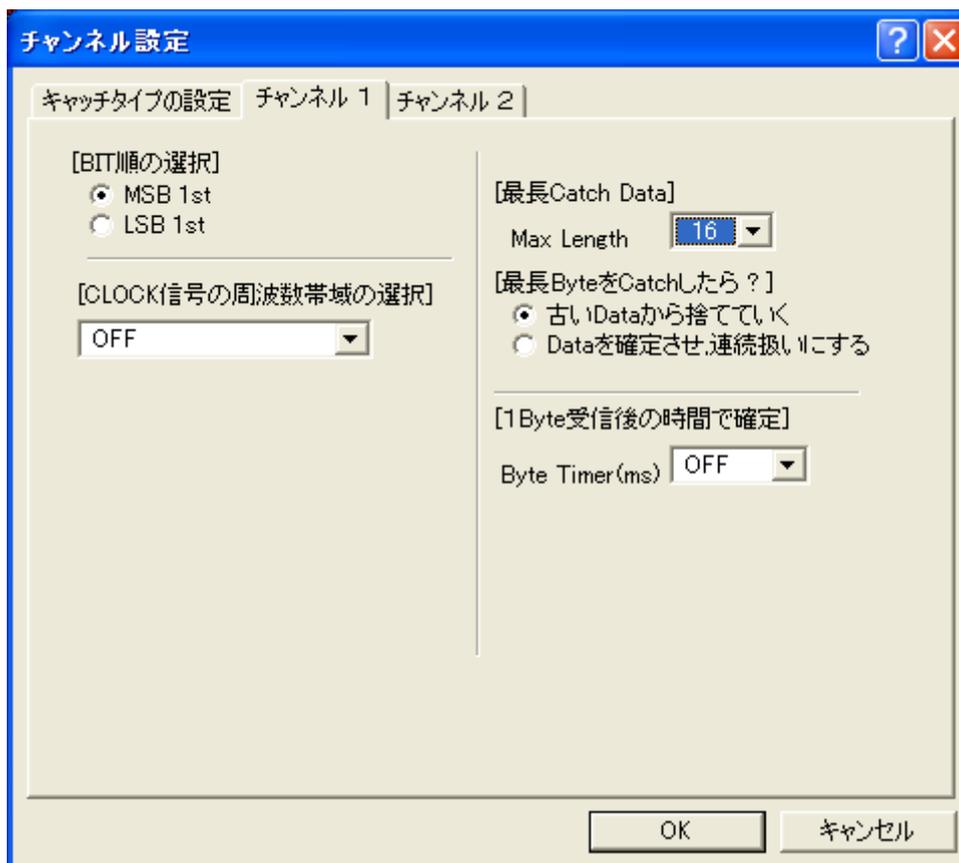
<Max Length>

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。尚、MAX 値は 32 となっております。

【キャッチデータ計測時間の設定】

本システムでは、現在通信パケットのデータ確定から次のデータ確定迄の時間を測定しそれを画面に表示しますが、ここでは、その測定時間の分解能について設定します。

13-2:〈チャンネル1〉〈チャンネル2〉タグでの設定内容について

**【BIT 順の選択】**

シリアル伝送方法を選択します。

【最長 Catch Data】

<Max Length>

一度にキャッチ可能なデータ数を指定します。

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。

尚、MAX 値は 32 となっております。

前頁の“チャンネル共通の最長 Catch Data”より短く設定して下さい。

【最長 Byte を Catch したら?】

ストップ・コンディションを受信しないまま、上記で設定した最長キャッチデータ数に達した場合の処理方法について設定します。

- [古い Data から捨てていく] オプションボタンをオンした場合
ストップ・コンディションが受信される迄、古いデータから順に捨てていき、ストップ・コンディションにてデータを確定する方法です。
- [Data を確定させ連続扱いにする] オプションボタンをオンした場合
ストップ・コンディションを受信される迄の全てのデータを取り込み、ストップ・コンディションにてデータを確定する方法です。
尚、通信パケット全体のバイト数が、指定した最長バイトになったら一度パソコンにデータを表示し、その後は、連続チャンネルとして表示します。

【1Byte 受信後の時間で確定】

1通信パケットの確定を 1Byte 受信後の時間で決定します。
設定値は 001～100 の間で選択が可能で、単位は msec です。

13-3:ポート設定

[キャッチ設定(L)]－[ポート設定(M)]をクリックすると、DataCatcherⅢ I/F Board のポート設定メニューが表示されます。画面の説明に従い、ボード本体の DIP SW と同様の設定を行ってください。

13-4:コネクタ表

20pin コネクタピン配列(コネクタ挿入口側から見た図)

2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

CN1 / CN2 (コネクタ名は基板裏側に記載)

	端子名		信号名
1	IN_A[0]	IN_AX[0]	CH1 CLK 入力
2	IN_A[1]	IN_AX[1]	CH1 DATA 入力
3	IN_A[2]	IN_AX[2]	
4	IN_A[3]	IN_AX[3]	
5	IN_B[0]	IN_BX[0]	CH2 CLK 入力
6	IN_B[1]	IN_BX[1]	CH2 DATA 入力
7	IN_B[2]	IN_BX[2]	
8	IN_B[3]	IN_BX[3]	

CN3

	端子名	信号名
1	TRIG[0]	外部トリガー入力
2	TRIG[1]	
3	EVENT[0]	イベントフィルターにて設定した EVENT 出力
4	EVENT[1]	

- 【注意】**・GND ターミナルとキャッチする機器の GND は必ず接続して下さい。
尚、接続してもキャッチデータが不安定な場合はアースを取って下さい。
・必ず DC3 の電源を投入してからプローブを接続して下さい。
故障の原因となります。

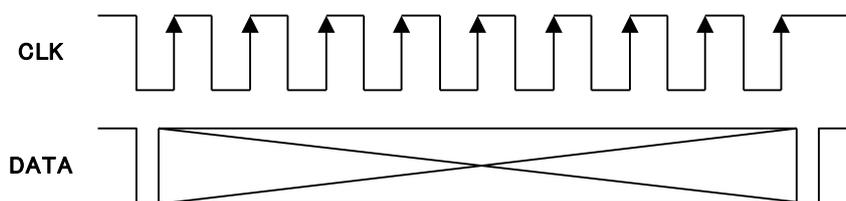
13-5:制限事項

受信可能なクロックの最高周波数

4MHz

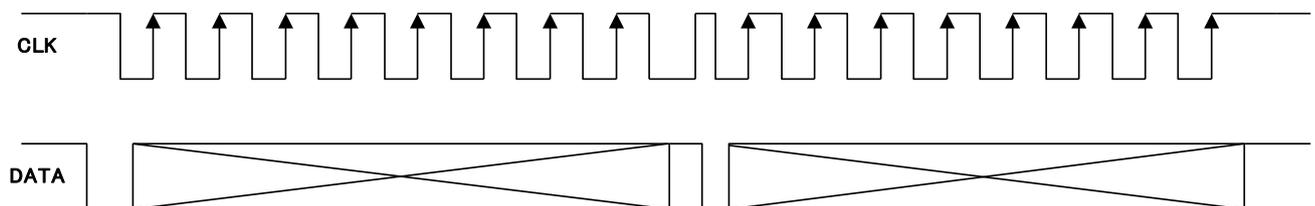
14. [I2CBUS_400K]Catch Program でキャッチできる波形

14-1:スタートコンディションと、ストップコンディションがペアになっているタイプ



2 チャンネルを使用し、2 系統までのバスをモニターすることができます。

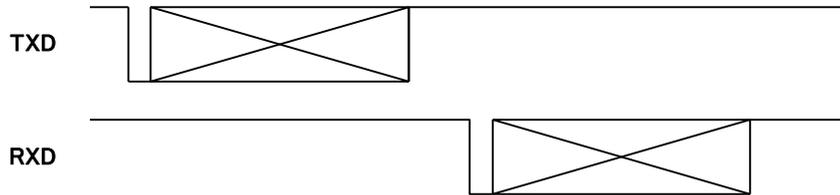
14-2:スタートコンディションと、ストップコンディションの間にリスタートがあるタイプ



2 チャンネルを使用し、2 系統までのバスをモニターすることができます。

15. [UART]Catch Program でキャッチできる波形について

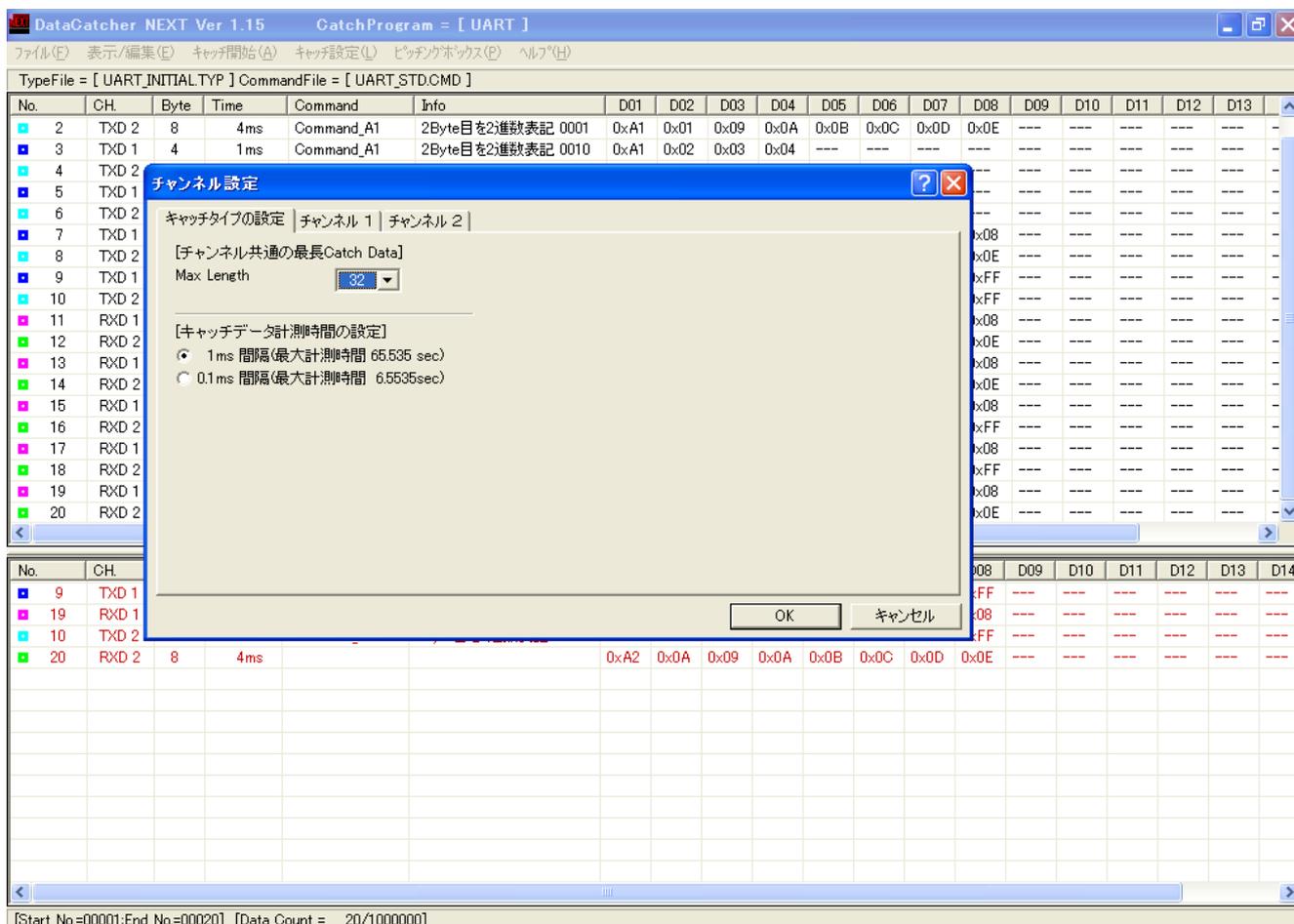
15-1:Low active の波形



主に CPU-デバイス間の通信を行う際の通信形態です。
2 チャンネルのポートを使用し、2 系統の通信ラインをキャッチできます。

16. [UART]Catch Program の設定について

[キャッチ設定(L)]-[キャッチタイプ設定(P)]をクリックすると、以下の〈チャンネル設定〉ダイアログボックスが開き、UART 通信における設定を行います。以下にその設定内容について説明をします。



16-1:〈キャッチタイプの設定〉タグでの設定内容について

【チャンネル共通の最長 Catch Data】

<Max Length>

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。尚、MAX 値は 32 となっております。

【キャッチデータ計測時間の設定】

本システムでは、現在通信パケットのデータ確定から次のデータ確定迄の時間を測定しそれを画面に表示しますが、ここでは、その測定時間の分解能について設定します。

16-2:〈チャンネル1〉〈チャンネル2〉タグでの設定内容について

【通信設定】

- <BPS> 通信速度を設定します。
▼をクリックし、リストを展開すると「110」～「921600」までの値から選択することが可能です。
また BPS の項目にキーボードから直接値を入力することも可能です。
その場合、「100」～「1000000」までの値を設定することが可能です。
- <BIT> データ・ビット数を設定します。(パリティビットは含んでおりません)
- <STOP BIT> STOP BIT の長さを設定します。
- <PARITY> パリティビットを設定します。

【STB の EDGE 選択】

- <TXD> TXD に対する通信パケット確定の為の EDGE を設定します。
- <RXD> RXD に対する通信パケット確定の為の EDGE を設定します。

【Start Bit 判定時 TXD (RXD)?】

片方のライン(TXD または RXD)のデータをキャッチしている最中に、もう片方のスタート・ビットをキャッチした時の、キャッチ中のデータについての処理方法について設定します。

【最長 Catch Data】

- <Max Length>
画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。
尚、MAX 値は 32 となっております。

※前頁の“チャンネル共通の最長 Catch Data”より短く設定して下さい。

【最長 Byte を Catch したら？】

通信パケットの確定条件が成立しないまま、上記で設定した最長キャッチデータ数に達した場合の処理方法について設定します。

- [古い Data から捨てていく] オプションボタンをオンした場合
通信パケットの確定条件が成立される迄、古いデータから順に捨てていき、通信パケットの確定条件の成立にてデータを確定する方法です。
- [Data を確定させ連続扱いにする] オプションボタンをオンした場合
通信パケットの確定条件が成立される迄の全てのデータを取り込み、通信パケットの確定条件の成立にてデータを確定する方法です。
尚、通信パケット全体のバイト数が、指定した最長バイトになったら一度パソコンにデータを表示し、その後は、連続チャンネルとして表示します。

【先頭バイトを指示する DATA あり】

通信パケットの先頭バイトを示すデータがある場合は、ここを設定します。

【最終バイトを指示する DATA で確定】

通信パケットの最終バイトを示すデータがあり、それで通信パケットを確定する場合はここを設定します。

最終バイトを指示するデータで確定する場合、その内容を 0x00h~0xffh で指定し、特に指示するデータが無い場合は OFF として下さい。

【1Byte 受信後の時間で確定】

1通信パケットの確定を 1Byte 受信後の時間で決定します。設定値は 001~100 の間で選択が可能で、単位は msec です。

【長さを指定する Byte がある】

通信パケットの長さを指定するデータが通信パケットにある時に設定します。

- ・ 何 Byte 目 : 通信パケット内の何 Byte 目にあるかを指定します。
01~32 で選択します。
- ・ 補 正 : 例えば STX や ETX が通信パケットに含まれている場合で、上記データがそれを含まない場合に、ここで含まない分について補正します。
01~32 で選択します。

【長さを確定させる Byte がある】

通信パケットの長さを確定するデータがある場合、ここでそのデータが何 byte 目にあるか、01～32 で選択し指定します。

尚、このデータに対応する通信パケットの長さについては .typ ファイルにて以下の様に対応させます。(以下の例はチャンネル1についての場合です。)

例えば、ここで指定した××Byte 目のデータが 0x12h で、通信パケットの Byte 数を 16Byte と設定する場合は以下の様になります。

— 実際のファイル内容 —

```

CH1_COMMAND_STB=00
; : コマンドによってパケットの長さを確定する。バイト数の設定値
; : 無効なコマンド 00
; : 有効なコマンド 01 -- ff (1 バイトから 255 バイトまで)
; :
; : CH1_LENGTH を超えたら一度データを上げます
; : 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
$00=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$10=00,00,00,10,00,00,00,00,00,00,00,00,00,00,00,00
$20=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$30=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$40=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$50=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$60=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$70=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$80=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$90=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$a0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$b0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$c0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$d0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$e0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$f0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

```

【タイプファイル編集】

タイプファイルをメモ帳で開きます。タイプファイルを直接編集したい場合に利用してください。

なお、チャンネル設定ウィンドウで変更した内容は、タイプファイルを保存するまでファイルに書き込まれませんので、既に変更をされた場合は、一度「OK」ボタンをクリックし、「ファイル」メニューより「TYPE ファイルを保存」を選んで保存した後、再度タイプファイル編集を行ってください。

このボタンを押すと、以下のインフォメーションウィンドウが開かれます。



16-3:ポート設定

[キャッチ設定(L)]-[ポート設定(M)]をクリックすると、DataCatcher III I/F Board のポート設定メニューが表示されます。画面の説明に従い、ボード本体の DIP SW と同様の設定を行ってください。

16-4:コネクタ表

20pin コネクタピン配列(コネクタ挿入口側から見た図)

2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

CN1 / CN2 (コネクタ名は基板裏側に記載)

	端子名		信号名
1	IN_A[0]	IN_AX[0]	CH1 TX 入力
2	IN_A[1]	IN_AX[1]	CH1 RX 入力
3	IN_A[2]	IN_AX[2]	CH1 TX-STB 入力
4	IN_A[3]	IN_AX[3]	CH1 RX-STB 入力
5	IN_B[0]	IN_BX[0]	CH2 TX 入力
6	IN_B[1]	IN_BX[1]	CH2 RX 入力
7	IN_B[2]	IN_BX[2]	CH2 TX-STB 入力
8	IN_B[3]	IN_BX[3]	CH2 RX-STB 入力

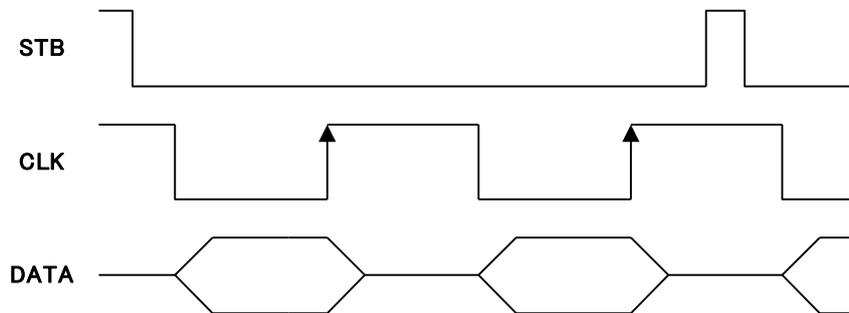
CN3

	端子名	信号名
1	TRIG[0]	外部トリガー入力
2	TRIG[1]	
3	EVENT[0]	イベントフィルターにて設定した EVENT 出力
4	EVENT[1]	

- 【注意】**・GND ターミナルとキャッチする機器の GND は必ず接続して下さい。
 尚、接続してもキャッチデータが不安定な場合はアースを取って下さい。
 ・必ず DC3 の電源を投入してからプローブを接続して下さい。
 故障の原因となります。

17. [PARA_CATCH]Catch Program でキャッチできる波形について

17-1:16Bit までの DATA 幅の平行通信の波形

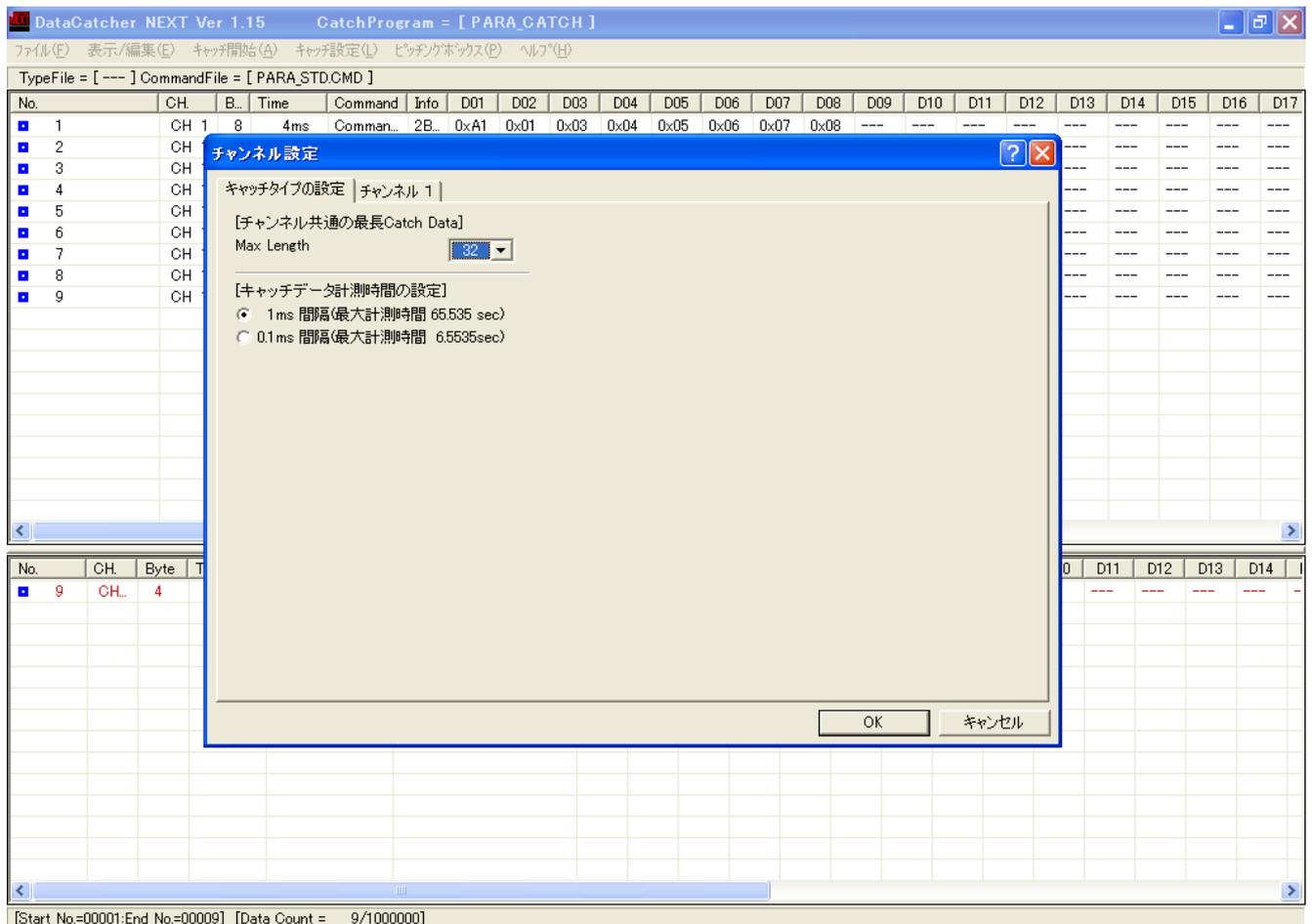


上記の様に一般的な「Parallel 通信」には STB 信号がありますが、STB 信号の無い以下の様な形式にて通信パケットを確定するものについても対応しております。

- ・Byte 間の時間にて通信パケットを確定する形式。
- ・通信データの中に、通信パケットの長さを指定するデータがある形式。
- ・通信パケットの終了を指示するデータがある形式。
- ・通信パケットの長さを確定するデータがある形式。

18. [PARA_CATCH]Catch Program の設定について

[キャッチ設定(L)]-[キャッチタイプ設定(P)]をクリックすると、以下の〈チャンネル設定〉ダイアログボックスが開き、平行通信における設定を行います。以下にその設定内容について説明をします。



18-1:〈キャッチタイプの設定〉タグでの設定内容について

【チャンネル共通の最長 Catch Data】

〈Max Length〉

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。尚、MAX 値は 32 となっております。

【キャッチデータ計測時間の設定】

本システムでは、現在通信パケットのデータ確定から次のデータ確定迄の時間を測定しそれを画面に表示しますが、ここでは、その測定時間の分解能について設定しま

18-2:〈チャンネル1〉〈チャンネル2〉タグでの設定内容について

チャンネル設定 (チャンネル 1)

キャッチタイプの設定

[CLOCK信号の周波数帯域の選択]
OFF

[CLOCK信号 データ取り込みエッジ選択]
 HI LOW

[DELAY TIME]
CLOCK DELAY (us) OFF

[DATA BIT数の指定]
 8Bit以下 9Bit以上

[BIT順の選択]
 MSB 1st LSB 1st

[STBのEDGE選択]
 LOW HI
 HI/LOW NON

[Catch Dataの順番の選択]
 キャッチした時間順
 STB(ラッチ)に近い順

[Catch Dataの論理を反転させるか?]
 反転させない
 反転させる

[Catch Dataが全て [0]のとき]
 データを捨てない
 データを捨てる

[Catch Dataが全て [1]のとき]
 データを捨てない
 データを捨てる

[最長Catch Data]
Max Length 16

[最長ByteをCatchしたら?]
 古いDataから捨てていく
 Dataを確定させ、連続扱いにする

[最終バイトを指示するDATAで確定]
Last Byte OFF

[1Byte受信後の時間で確定]
Byte Timer (ms) OFF

[長さを指定するByteがある]
何Byte目 OFF
補正 00

[長さを確定させるByteがある]
何Byte目 OFF

タイプファイル編集(再読み込み必要)

OK キャンセル

【CLOCK 信号データ取り込みエッジ選択】

DATA 確定のエッジを選択します。

【DATA BIT 数の指定】

DATAのBit幅を選択します。

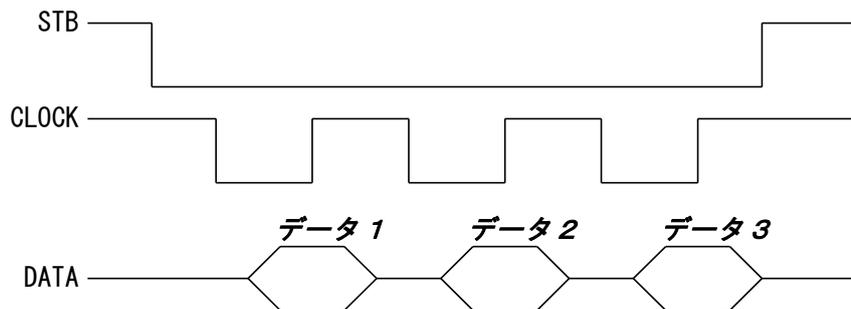
9Bit 以上を選択した時の最大 Bit 幅は 16Bit です。

【STB の EDGE 選択】

ラッチ信号の EDGE の選択をします。

【Catch Data 順番の選択】

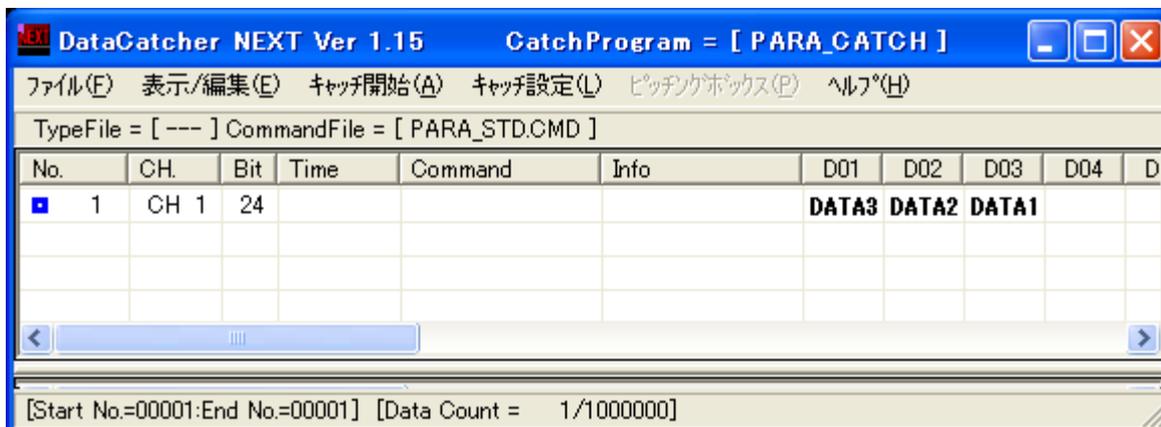
キャッチしたデータに対する表示順番を指定します。



【キャッチした時間順】のオプションボタンをオンすると以下のようになります。



【STB(ラッチ)に近い順】のオプションボタンをオンすると以下のようになります。



【Catch Data の論理を反転させるか？】

キャッチしたデータの論理反転/非反転を選択します。

【Catch Data が全て[0]のとき】

STB(ラッチ)迄の全 bit が 0 であった場合の処理方法について設定します。

【Catch Data が全て[1]のとき】

STB(ラッチ)迄の全 bit が 1 であった場合の処理方法について設定します。

【最長 Catch Data】

<Max Length>

一度にキャッチ可能なデータ数を指定します。

画面の DATA 列の列数を指定します。つまり、これを 10 と設定すると DATA 列として D01～D10 迄が表示され、16 と設定すると、D01～D16 迄が表示されます。尚、MAX 値は 32 となっております。

※前頁の“チャンネル共通の最長 Catch Data”より短く設定して下さい。

【最長 Byte を Catch したら？】

STB が受信しないまま、上記で設定した最長キャッチデータ数に達した場合の処理方法について設定します。

○ [古い Data から捨てていく] オプションボタンをオンした場合

STB が受信される迄、古いデータから順に捨てていき、STB にてデータを確定する方法です。

○ [データを確定させ、連続扱いにする]オプションボタンをオンした場合

STB が受信されるまで、全てのデータを取り込み、ストップ・コンディションにてデータを確定する方法です。

尚、通信パケット全体のバイト数が、指定した最長バイトになったら一度パソコンにデータを表示し、その後は、連続チャンネルとして表示します。

以下の項目については、一つの通信パケットを確定する方法について設定するものです。

【最終バイトを指示する DATA で確定】

最終バイトを指示するデータで確定する場合、その内容を 0x00h～0xffh で指定し、特に指示するデータが無い場合はOFFとして下さい。

【1Byte 受信後の時間で確定】

1通信パケットの確定を 1Byte 受信後の時間で決定します。設定値は 001～100 の間で選択が可能で、単位は<キャッチタイプの設定>タグ内の[キャッチデータ計測時間の設定]項目にて指定した単位となります。

特に指示するデータが無い場合はOFFとして下さい。

【長さを指定する Byte がある】

通信パケットの長さを指定するデータが通信パケットにある時に設定します。

- ・ 何 Byte 目 : 通信パケット内の何 Byte 目にあるかを 01～32 で選択し、指定します。
特に指示するデータが無い場合はOFFとして下さい。
- ・ 補 正 : 例えば STX や ETX が通信パケットに含まれている場合で、上記データがそれを含まない場合に、ここで含まない分について 01～32 で選択し、補正します。
特に指示するデータが無い場合は 00 として下さい。

※DATA の Bit 幅が 8BIT 以下でのみ使用可能です。

【長さを確定させる Byte がある】

通信パケットの長さを確定するデータがある場合、ここでそのデータが何 byte 目にあるかを 01～32 で選択し、指定します。

尚、このデータに対応する通信パケットの長さについては .typ ファイルにて以下の様に
対応させます。(以下の例はチャンネル1についての場合です。)

例えば、ここで指定した××Byte 目のデータが 0x12h で、通信パケットの Byte 数を
16Byte と設定する場合は以下の様になります。

※DATA の Bit 幅が 8BIT 以下でのみ使用可能です。

— 実際のファイル内容 —

```

CH1_COMMAND_STB=00
; : コマンドによってパケットの長さを確定する。バイト数の設定値
; : 無効なコマンド 00
; : 有効なコマンド 01 -- ff (1 バイトから 255 バイトまで)
; :                               CH1_LENGTH を超えたら一度データを上げます
; : 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f
$00=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$10=00,00,10,00,00,00,00,00,00,00,00,00,00,00,00,00
$20=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$30=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$40=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$50=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$60=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$70=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$80=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$90=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$a0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$b0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$c0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$d0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$e0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00
$f0=00,00,00,00,00,00,00,00,00,00,00,00,00,00,00,00

```

【タイプファイル編集】

タイプファイルをメモ帳で開きます。タイプファイルを直接編集したい場合に利用してください。

なお、チャンネル設定ウィンドウで変更した内容は、タイプファイルを保存するまでファイルに書き込まれませんので、既に変更をされた場合は、一度「OK」ボタンをクリックし、「ファイル」メニューより「TYPE ファイルを保存」を選んで保存した後、再度タイプファイル編集を行ってください。

このボタンを押すと、以下のインフォメーションウィンドウが開かれます。



18-3:ポート設定

[キャッチ設定(L)]-[ポート設定(M)]をクリックすると、DataCatcherⅢ I/F Board のポート設定メニューが表示されます。画面の説明に従い、ボード本体の DIP SW と同様の設定を行ってください。

18-4:コネクタ表

20pin コネクタピン配列(コネクタ挿入口側から見た図)

2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

CN1 / CN2 (コネクタ名は基板裏側に記載)

	端子名		信号名
1	IN_A[0]	IN_AX[0]	CLK 入力
2	IN_A[1]	IN_AX[1]	STB 入力
3	IN_A[2]	IN_AX[2]	
4	IN_A[3]	IN_AX[3]	
5	IN_B[0]	IN_BX[0]	DAT0 入力
6	IN_B[1]	IN_BX[1]	DAT1 入力
7	IN_B[2]	IN_BX[2]	DAT2 入力
8	IN_B[3]	IN_BX[3]	DAT3 入力
9	IN_C[0]	IN_CX[0]	DAT4 入力
10	IN_C[1]	IN_CX[1]	DAT5 入力
11	IN_C[2]	IN_CX[2]	DAT6 入力
12	IN_C[3]	IN_CX[3]	DAT7 入力
13	IN_D[0]	IN_DX[0]	DAT8 入力
14	IN_D[1]	IN_DX[1]	DAT9 入力
15	IN_D[2]	IN_DX[2]	DAT10 入力
16	IN_D[3]	IN_DX[3]	DAT11 入力
17	IN_E[0]	IN_EX[0]	DAT12 入力
18	IN_E[1]	IN_EX[1]	DAT13 入力
19	IN_E[2]	IN_EX[2]	DAT14 入力
20	IN_E[3]	IN_EX[3]	DAT15 入力

CN3

	端子名	信号名
1	TRIG[0]	外部トリガー入力
2	TRIG[1]	
3	EVENT[0]	イベントフィルタにて設定した EVENT 出力
4	EVENT[1]	

【注意】・GND ターミナルとキャッチする機器の GND は必ず接続して下さい。
 尚、接続してもキャッチデータが不安定な場合はアースを取って下さい。
 ・必ず DC3 の電源を投入してからプローブを接続して下さい。
 故障の原因となります。

18-5: 制限事項

受信可能なクロックの最高周波数

10 MHz MAX(連続の場合は、3.5MHz)

◆下記の packets を区切る機能は、DATA の Bit 幅が 8BIT 以下でのみ使用可能です。

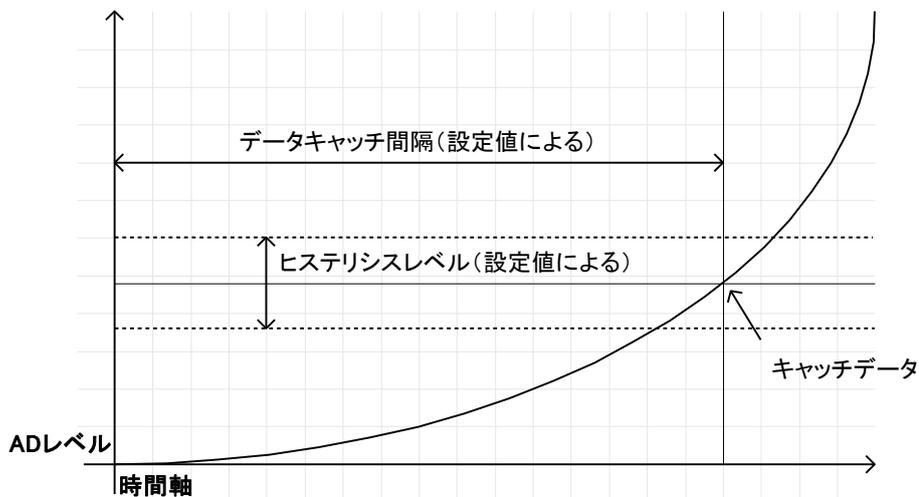
- ・長さを指定する Byte で packets を確定させる機能。
- ・長さを確定させる Byte で packets を確定させる機能。

19. [DATA_LOGGER]Catch Program について

19-1: デジタル信号取得

- ・使用ポートのスレッシュレベルにて、一定間隔で HIGH/LOW の判定をします。
- ・キャッチ間隔(RATE)は設定により 10KHz/50KHz が選択できます。
- ・8 系統の信号ラインがキャッチ可能です。

19-2: ADデータ取得

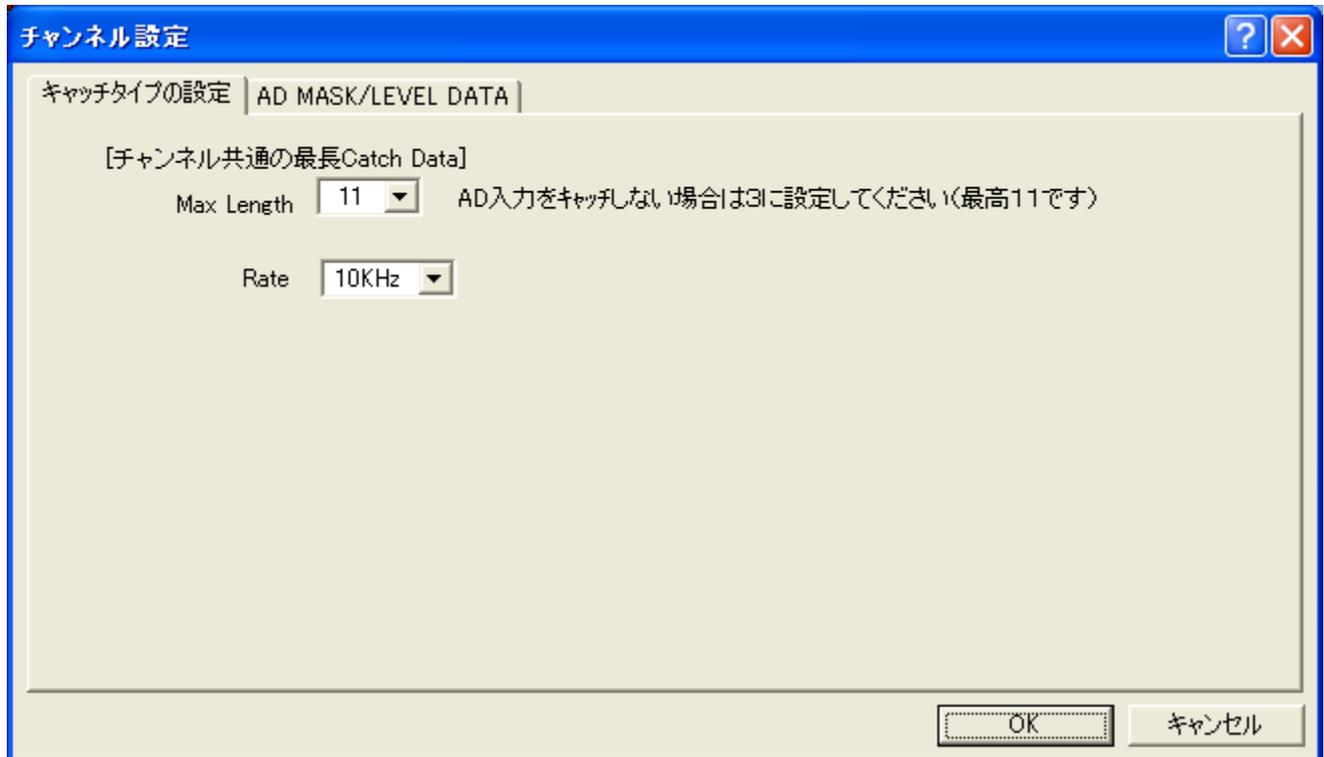


- ・ ADレベルを 8 ビット分解能でキャッチ可能です。
- ・ キャッチ間隔(RATE)は設定により 10KHz/50KHz が選択可能で、データに変化があった場合のみデータを画面上に表示します。
- ・ ヒステリシスレベルの設定により、データ変化量の調整が可能です。
- ・ 8 系統の通信ラインをキャッチ可能です。
- ・ アナログ入力については、別途 AnalogBox を購入頂く必要があります。

20. データロガー機能の設定について

20-1:〈キャッチタイプの設定〉タグでの設定内容について

[キャッチ設定(L)]-[キャッチタイプ設定(P)]をクリックすると、以下の〈チャンネル設定〉ダイアログボックスが開き、データロガー機能における設定を行います。
以下にその設定内容について説明をします。



【チャンネル共通の最長 Catch Data】

<Max Length>

画面の DATA 列の列数を指定します。つまり、これを 3 と設定すると DATA 列として D01～D3 迄が表示され、5 と設定すると、D01～D5 迄が表示されます。
なお、有効なのは 5 までです。

【Rate】

サンプリングレートを指定します。10KHz/50KHzの 2 通りの設定が切替できます。

20-2:〈AD MASK/LEVEL DATA〉タグの設定について

ADの取り込みについて変化の検出方法を設定できます。

チャンネル設定

キャッチタイプの設定 **AD MASK/LEVEL DATA**

AD Mask Bitの設定

AD1 設定値 MASK Bit	AD2 設定値 MASK Bit	AD3 設定値 MASK Bit	AD4 設定値 MASK Bit
0 Bit	0 Bit	0 Bit	0 Bit
AD5 設定値 MASK Bit	AD6 設定値 MASK Bit	AD7 設定値 MASK Bit	AD8 設定値 MASK Bit
0 Bit	0 Bit	0 Bit	0 Bit

ADヒステリシスLEVELの設定

AD1 設定値 LEVEL+/-	AD2 設定値 LEVEL+/-	AD3 設定値 LEVEL+/-	AD4 設定値 LEVEL+/-
004	004	004	004
AD5 設定値 LEVEL+/-	AD6 設定値 LEVEL+/-	AD7 設定値 LEVEL+/-	AD8 設定値 LEVEL+/-
004	004	004	004

初期値に戻す

OK キャンセル

- AD Mask Bit の設定
 - 8ビット分解能で取得するADデータをマスクする数をLSBで指定します。
例えば、0xFFのデータを1Bitマスクすると0xFE、4Bitマスクすれば0xF0となります。
特に指定しない場合は、0Bitを指定してください。
- ADヒステリシス Level の設定
 - ADレベル取得時のヒステリシスを設定します。設定されている範囲内の変化量は無視します。
例えば前回のデータが100でレベル設定が4のとき96から104までの値は無視します。
範囲は±1～99レベルです。特に指定しない場合は、OFFを設定してください。

20-3:ポート設定

[キャッチ設定(L)]-[ポート設定(M)]をクリックすると、DataCatcher III I/F Boardのポート設定メニューが表示されます。画面の説明に従い、ボード本体のDIP SWと同様の設定を行ってください。

20-4:コネクタ表

20pin コネクタピン配列(コネクタ挿入口側から見た図)

2	4	6	8	10	12	14	16	18	20
1	3	5	7	9	11	13	15	17	19

CN1 / CN2 (コネクタ名は基板裏側に記載)

	端子名		信号名
1	IN_A[0]	IN_AX[0]	DAT0 入力(デジタル)
2	IN_A[1]	IN_AX[1]	DAT1 入力(デジタル)
3	IN_A[2]	IN_AX[2]	DAT2 入力(デジタル)
4	IN_A[3]	IN_AX[3]	DAT3 入力(デジタル)
5	IN_B[0]	IN_BX[0]	DAT4 入力(デジタル)
6	IN_B[1]	IN_BX[1]	DAT5 入力(デジタル)
7	IN_B[2]	IN_BX[2]	DAT6 入力(デジタル)
8	IN_B[3]	IN_BX[3]	DAT7 入力(デジタル)

CN3

	端子名	信号名
1	TRIG[0]	外部トリガー入力
2	TRIG[1]	
3	EVENT[0]	イベントフィルターにて設定した EVENT 出力
4	EVENT[1]	

※2chのアナログ入力は、AnalogBox のコネクタより入力されます。

- 【注意】**
- ・GND ターミナルとキャッチする機器の GND は必ず接続して下さい。
 - 尚、接続してもキャッチデータが不安定な場合はアースを取って下さい。
 - ・必ず DC3 の電源を投入してからプローブを接続して下さい。
 - 故障の原因となります。

21. コマンド解析用定義ファイルの作成方法

コマンド解析用定義ファイルとは、キャッチしたデータ(以下の画面イメージ1の D01～D10)をもとに、画面イメージ1の“展開結果”で示す箇所への表示内容及びその展開方法について定義するファイルです。尚、このファイルは任意のディレクトリーに保存する事が可能ですが、拡張子は“.cmd“として下さい。また、インストールされた時点で、“¥HeartLandData¥DC3¥lib”ディレクトリーに“3WIRE_CATCH.CMD”などといった参考ファイルが出来ますので、それを参考に作成して下さい。

【画面イメージ1】

The screenshot shows a window titled "Dc2000 Ver2.0 CatchProgram = [3WIRE_ST]". Below the title bar are menu items: "ファイル(F)", "表示/編集(E)", "キャッチ開始(A)", "キャッチ設定(L)", and "ヘルプ(H)". Below the menus, it shows "TypeFile = [3wire_st.typ]" and "CommandFile = [HDC_LC78640.CMD]". The main area contains a table with columns: No., CH., Bit, Time, Command, Info, D01, and D02. The table has 7 rows of data. The third row (No. 3267) is highlighted in red, and a red arrow points to the "Info" column of this row with the label "展開結果". The status bar at the bottom reads "Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]".

No.	CH.	Bit	Time	Command	Info	D01	D02
1345	CH 1	8	7626ms	[MUTE -∞]	1 byte	0xEA	0x00
3256	CH 1	8	3664ms	[MUTE -∞]	1 byte	0xEA	0x00
3267	CH 1	8	408ms	[MUTE -∞]	1 byte	0xEA	0x00
4584	CH 1	8	1842ms	[MUTE -∞]	1 byte	0xEA	0x00
5899	CH 1	8	2445ms	[MUTE -∞]	1 byte	0xEA	0x00
7249	CH 1	8	2546ms	[MUTE -∞]	1 byte	0xEA	0x00
9171	CH 1	8	3784ms	[MUTE -∞]	1 byte	0xEA	0x00

21-1:特徴(こんなことができ便利)

- キャッチしたデータ(D01～)を command 列, Info 列に理解しやすい内容に展開出来ます。
- 展開内容については、汎用のテキスト・エディターを使用して、左辺に展開したいデータグループを指定して“=”の右辺に展開する書式を記述するだけです。
- 10進, 16進, ASCII, Bit 列, 文字列など多彩な展開ができます。
- 書式はC言語の printf()関数に似ている為、誰でも簡単にマスターできます。
- キャッチウィンドウから設定したコマンドファイル読み込むことにより変更内容がすぐに反映され、確認が簡単にできます。

21-2:制限・制約

- 文字列には“,”(カンマ)は使用できません。
- 文字列に“%”を使用する場合は、“%%”と記述して下さい。
- 1行の文字数は256文字迄です。
- 各行の先頭が“;”(セミコロン)で始まる行はコメント行となります。

21-3: セクションについて

【1】3WIRE_CATCH の場合

— 記述形式 —

[CH**ChNo.****Bit 数**]
 定義体=書式1,,書式2

ChNo.: 接続チャンネル No.を指定する。
Bit 数: 通信パケットの総 Bit 数を指定する。

— 解説 —

[CH1_008]と記述した場合、チャンネル1の通信パケットが8Bit で構成されているデータに対する定義である事を意味します。

尚、本システムでは1行に表示できるデータ数はD32(32Byte)迄となっており、例えば36Byteで通信パケットを構成する場合は、2行にまたがって表示し、CH 列の表示は”**CHCONTX**”(X は CHNo.)

となり、その列のD01(33Byte 目)~D04(36Byte 目)迄が通信データの対象となります。

この場合の記述は**[CHCONTChNo. Bit 数]**となり、**[CHCONT1_032]**と記述します。

[CH1_000] と記述した場合、通信パケットの Bit 数は無視され、チャンネル1でキャッチした全ての通信パケットが、展開の対象となります。

書式を“,,”(カンマカンマ)で区切った場合、その手前まで(書式1部分)が画面イメージ1のCommand 列の展開結果(この場合“**[MUTE -∞]**”)となり、以降(書式2部分)がInfo 列の展開結果(この場合“**1Byte**”)になります。

定義体及び**書式**については後述します。

— 注意 —

Bit 数は必ず3桁で指定して下さい。

例	記述例
Ch1 の通信パケットが 8Bit データの場合	[CH1_008]
Ch1 の通信パケットが 16Bit データの場合	[CH1_016]
Ch1 の通信パケットが 128Bit データの場合	[CH1_128]

アルファベットについては大文字、小文字の両方が記述可能ですが、混在はできません。

【2】I2CBUS_400K の場合

— 記述形式 —

[CHChNo. Byte 数]
定義体=書式1,,書式2

ChNo. : 接続チャンネル No.を指定する。
Byte 数 : 通信パケットの総 Byte 数を指定する。

— 解説 —

[CH1_008]と記述した場合、チャンネル 1 の通信パケットが 8Byte で構成されているデータに対する定義である事を意味します。

尚、本システムでは 1 行に表示できるデータ数は D32(32Byte)迄となっており、例えば 36Byte で通信パケットを構成する場合は、2 行にまたがって表示し、CH 列の表示は"CHCONTX"(X は CHNo.)

となり、その列の D01(33Byte 目)~D04(36Byte 目)迄が通信データの対象となります。

この場合の記述は**[CHCONTChNo. Byte 数]**となり、**[CHCONT1_004]**と記述します。

[CH1_000] と記述した場合、通信パケットの Byte 数は無視され、チャンネル 1 でキャッチした全ての通信パケットが展開の対象となります。

書式を“,”(カンマカンマ)で区切った場合、その手前まで(書式1部分)が画面イメージ 1 の Command 列の展開結果(この場合 "**[MUTE -∞]**")となり、以降(書式 2 部分)が Info 列の展開結果(この場合 "**1Byte**")になります。

定義体及び**書式**については後述します。

— 注意 —

Byte 数は必ず3桁で指定して下さい。

例	記述例
Ch1 の通信パケットが 8Byte データの場合	[CH1_008]
Ch1 の通信パケットが 16Byte データの場合	[CH1_016]

アルファベットについては大文字, 小文字の両方が記述可能ですが、混在はできません。

— 記述形式 —

[ACK ChNo. Byte 数]

定義体=書式1,,書式2

ChNo. : 接続チャンネル No.を指定する。
Byte 数 : 通信パケットの総 Byte 数を指定する。

— 解説 —

I2CBUS_400K では、2 行を 1 セットとして表示しており、1 行目には通信データを表示し、2 行目にはその通信データに対する ACK/NACK を表示します。

[ACK1_002]と記述した場合、チャンネル 1 の 2Byte の通信データについての ACK/NACK に対する定義である事を意味します。

【3】UART の場合

— 記述形式 —

[TX ChNo. Byte 数]

定義体=書式1,,書式2

ChNo. : 接続チャンネル No.を指定する。
Byte 数 : 通信パケットの総 Byte 数を指定する。

— 解説 —

[TX1_008]と記述した場合、チャンネル 1 の通信パケットが 8Byte で構成されているデータに対する定義である事を意味します。

尚、本システムでは 1 行に表示できるデータ数は D32(32Byte)迄となっており、例えば 36Byte で通信パケットを構成する場合は、2 行にまたがって表示し、CH 列の表示は"TXCONTX"(X は CHNo.)

となり、その列の D01(33Byte 目)~D04(36Byte 目)迄が通信データの対象となります。

この場合の記述は[TXCONTChNo. Byte 数]となり、[TXCONT1_004]と記述します。

[TX1_000] と記述した場合、通信パケットの Byte 数は無視され、チャンネル1でキャッチした全ての通信パケットが展開の対象となります。

書式を“,,”(カンマカンマ)で区切った場合、その手前まで(書式1部分)が画面イメージ1の Command 列の展開結果(この場合 “[MUTE -∞]”)となり、以降(書式2部分)が Info 列の展開結果(この場合 “1Byte”)になります。

定義体及び**書式**については後述します。

— 注意 —

Byte 数は必ず3桁で指定して下さい。

例	記述例
Ch1 の TXD の通信パケットが 8Byte データの場合	[TX1_008]
Ch1 の TXD の通信パケットが 16Byte データの場合	[TX1_016]

アルファベットについては大文字、小文字の両方が記述可能ですが、混在はできません。

— 記述形式 —

[RX ChNo. Byte 数]

定義体=書式1,,書式2

ChNo. : 接続チャンネル No.を指定する。

Byte 数 : 通信パケットの総 Byte 数を指定する。

— 解 説 —

[RX1_008]と記述した場合、チャンネル 1 の通信パケットが 8Byte で構成されているデータに対する定義である事を意味します。

尚、本システムでは 1 行に表示できるデータ数は D32(32Byte)迄となっており、例えば 36Byte で通信パケットを構成する場合は、2 行にまたがって表示し、CH 列の表示は"RXCONTX"(X は CHNo.)となり、その列の D01(33Byte 目)~D04(36Byte 目)迄が通信データの対象となります。

この場合の記述は**[RXCONTChNo. Byte 数]**となり、**[RXCONT1_004]**と記述します。

[RX1_000] と記述した場合、通信パケットの Byte 数は無視され、チャンネル1でキャッチした全ての通信パケットが展開の対象となります。

21-4: 定義体について

定義体には HEX 形式と Bit 形式の 2 種類があります。

【1】HEX 形式の定義体

— 記述形式 —

\$16進数h 又は \$16進数H

16進数 : 0~f 迄の 16 進数を指定する。

— 解説 —

半角の数字及び半角 a(A),b(B),c(C),d(D),e(E),f(F),x(X)以外は使用しないで下さい。

16進数が以下の時は定数扱いとなります。

半角の 0,1,2,3,4,5,6,7,8,9 半角 a(A),b(B),c(C),d(D),e(E),f(F)

16進数が x または X の場合は、そのデータを無視します。

— 記述例 —

\$10h=

\$0xh=

\$xxh=

\$12h34h=

\$12hxxhxxhxxh=

【2】Bit 形式の定義体

— 記述形式 —

\$2進数b 又は \$2進数B

2進数: 0,1 の 2 進数を指定する。

— 解説 —

半角の 0,1 及び半角 x(X)以外は使用しないで下さい。

2進数が半角の 0,1 の時は定数扱いとなります。

2進数が x または X の場合は、そのデータを無視します。

— 記述例 —

\$10000000b=

\$0000xxxxb=

\$xxxxxxxxb=

\$00010010B00110100B=

\$00010010bXXXXXXXXbXXXXXXXXbXXXXXXXXb=

【3】HEX 形式 Bit 形式の混在

Byte 単位で、HEX 形式と Bit 形式の混在が可能です。

— 記述形式 —



2進数 : 0,1 の 2 進数を指定する。
16進数 : 0~f 迄の 16 進数を指定する。

— 記述例 —

\$10000000b20h=
\$0000xxxxbXXh=
\$xxhxxxxxxxxxb=
\$00010010B00110100B=
\$xxh00010010b00110100Bxxh=

行の先頭が"\$="の場合は、無条件に全てのデータがこれに該当致します。
尚、データの展開処理は、記述(定義)された順に比較し、該当したものについて展開していきます。よって、この"\$="の定義が最初のほうにある場合は、全てそれに該当してしまいますので、注意して下さい。

21-5:書式について

通常の文字列については、Command 列または Info 列へそのまま表示され、以下の様に書式指定されたものについては、その書式内容に従い書式化され画面に表示されます。

— 記述形式 —



width : 表示フィールドの最小幅を指定する。
type : 表示タイプを指定する。
引数 : 展開の対象となるデータを指定する。

— 解説 —

【1】 width について

width は表示フィールドの最小幅をセットします。

- n** 少なくともn字が表示されます。実際の表示幅がn字より小さければ左に空白が埋められ表示されます。
- On** 少なくともn字が表示されます。実際の表示幅がn字より小さければ左に0が埋められ表示されます。

【2】 type について

type は表示フィールドへの表示方法をセットします。

- d** 整数 符号つき 10 進数で表示します。
- u** 整数 符号なし 10 進数で表示します。
- b** 整数 2 進数で表示します。
- x** 整数 16 進数(半角の a,b,c,d,e,f を使用)で表示します。
- X** 整数 16 進数(半角の A,B,C,D,E,F を使用)で表示します。
- c** 文字 引数で指定されたデータを ASCII 文字で変換します。

S?? 文字列 文字パターンテーブルの種類を指定します。

【3】 引数について

引数は上記 width と type の指示に従い表示する DATA 列の DATA No.を示します。

dmm-dnn

DATAm~DATAAn 迄のデータを示し、**mm** 及び **nn** には **01~32** 迄の値が入ります。

dnn が下位 Byte となり、**-dnn**を省略すると**dmm**の 1Byte のみが対象となります。

dmmbn-doobp

DATAm.BITn~DATAoo.BITp 迄のデータを示し、**mm** 及び **oo** には **01~32** 迄の値が入り、**n** 及び **p** には **1~8** 迄の値が入ります。

doobpが下位 Bit となり、**-doobp**を省略すると**dmmbn**の 1Bit のみが対象となります。

21-6:文字パターンの定義について

— 記述形式 —



?? : 文字列定義のセクション No.を指定する。

xx : 00~ff 迄の値を指定する。

文字列 : 展開結果となる文字列を指定する。

— 解説 —

S??には、**S01~S99迄(計 99種類)**の文字列パターンを定義するセクションが設定可能です。

\$xxには、**\$00~\$ff** 迄の 256 個のテーブル・データが設定可能です。

1文字列の最大文字数は半角で 63 文字です。

??には半角文字のみ記入可能です。

— 記述例 —

[S01]

\$00=文字列A : 引数での指定データが 00hex の時、文字列Aを表示します。

\$3F=文字列B: 引数での指定データが 3Fhex の時、文字列Bを表示します。

21-7:特殊な定義体について

表示行をまたがった定義が可能です。
これは、通信パケットの Byte 数が 32 を超える場合などに有効です。

— 記述例 —

```
[CHCONT1_000]
$-1txxh01h=ZERO=%(4b,d01b4-d01b1)
```

— 通信パケット内容 —

```
00h,01h,03h,02h,05h,08h,01h,04h,0fh,00h
      ↑                ↑
    CH1 行表示部分  CONT1 行表示部分
```

— 画面表示内容 —

No.	command	CH.	Byte	D01	D02	D03	D04	D05	D06	D07	D08
1		CH1	8	0x00	0x01	0x03	0x02	0x05	0x08	0x01	0x04
2	ZERO=1111	CONT1	2	0x0f	0x00						

— 解説 —

本システムでは、コマンド解析用の定義については表示行単位となっております。
つまり、上記画面表示内容では、No.1 行と No.2 行において、各々に定義が必要となります。

通信タイプの設定にて、通信パケットを折り返して 2 行に表示する事が可能
(詳細は各種キャッチプログラムのマニュアルを参照して下さい。)ですが、
(通信パケットの総 Byte 数が 32Byte を超える場合は、自動的に 33byte 目で折り返されます。)
この様な場合、この定義体が有効です。

上記の様に、実際に定義したいのは No.2 の行に対してですが、**\$-1t** を使用する事により
1 行手前のデータを参照する事が可能になります。
つまり、この場合は、No.1 行の D02 の **0x01h** が定義体の対象となります。
また、**\$+xt** の記述も可能であり、この場合は、x 行先が対象となり、x は **1~9** までの範囲で
記述可能です。

21-8:具体的な記述例と画面イメージ

- 【1】 以下の No.2 データにおいて、D01 が 0xF8 だったら Command 列に[Test Data]と表示し、Info 列に 32Bit と表示させたい場合の記述例とその結果について以下に説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data],,32Bit

— 画面イメージ —

No.	CH.	Bit	Time	Command	Info	D01	D02	D03	D04
35	CH 1	32	5584ms	[Test Data]	32Bit	0xF8	0x00	0x04	0xEB
1381	CH 1	32	2045ms	[Test Data]	32Bit	0xF8	0x00	0x04	0xEB
3303	CH 1	32	4072ms	[Test Data]	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data]	32Bit	0xF8	0x00	0x04	0xEB

Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]

- 【2】 上記の D01 に対し Bit 形式で定義した場合についての記述例について以下に説明します。尚、この場合も上記と同じ画面結果になります。

— 記述例 —

[CH1_032]

\$11111000bxxhxxhxxh=[Test Data],,32Bit

- 【3】 No.2 データの D03 について“[Test Data] : Data4=”の後に 5 桁で 10 進数表示 (実際の表示桁数が 5 桁より小さければ左に空白が埋められます)させたい場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data3=%(5d,d04),,32Bit

— 画面イメージ —

No.	CH.	Bit	Time	Command	Info	D01	D02	D03	D04
3303	CH 1	32	4072ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
5935	CH 1	32	2445ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
7285	CH 1	32	2546ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB

Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]

【4】 上記を 2 桁表示に変えた場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data3=%(2d,d03),,32Bit

— 画面イメージ —

No.	CH	Bit	Time	Command	Info	D01	D02	D03	D04
3303	CH 1	32	4072ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
5935	CH 1	32	2445ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB
7285	CH 1	32	2546ms	[Test Data] : Data3= 4	32Bit	0xF8	0x00	0x04	0xEB

Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]

【5】 【3】と同じ条件で、実際の表示桁数が指定桁数より小さければ左に 0 を埋める場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data3=%(5d,d03),,32Bit

— 画面イメージ —

No.	CH	Bit	Time	Command	Info	D01	D02	D03	D04
3303	CH 1	32	4072ms	[Test Data] : Data3=00004	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data] : Data3=00004	32Bit	0xF8	0x00	0x04	0xEB
5935	CH 1	32	2445ms	[Test Data] : Data3=00004	32Bit	0xF8	0x00	0x04	0xEB
7285	CH 1	32	2546ms	[Test Data] : Data3=00004	32Bit	0xF8	0x00	0x04	0xEB

Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]

- 【6】 No.2 データの D04 について“[Test Data]:Data4=”の後に 16 進数表示させたい場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data4=%(5x,d04),,32Bit

— 画面イメージ —

The screenshot shows the Dc2000 Ver2.0 software interface. The title bar reads "Dc2000 Ver2.0 CatchProgram = [3WIRE_ST]". The menu bar includes "ファイル(F)", "表示/編集(E)", "キャッチ開始(A)", "キャッチ設定(L)", and "ヘルプ(H)". Below the menu bar, it shows "TypeFile = [3wire_st.typ]" and "CommandFile = [test.CMD]". The main area contains a table with the following data:

No.	CH.	Bit	Time	Command	Info	D01	D02	D03	D04
35	CH 1	32	5584ms	[Test Data] : Data4= eb	32Bit	0xF8	0x00	0x04	0xEB
1381	CH 1	32	2045ms	[Test Data] : Data4= eb	32Bit	0xF8	0x00	0x04	0xEB
3303	CH 1	32	4072ms	[Test Data] : Data4= eb	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data] : Data4= eb	32Bit	0xF8	0x00	0x04	0xEB

At the bottom of the window, it says "Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]".

- 【7】 No.2 データの D09 について“[Test Data]:Data4=”の後に ASCII 文字で表示させたい場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data9=%(c,d09),,32Bit

— 画面イメージ —

The screenshot shows the Dc2000 Ver2.0 software interface. The title bar reads "Dc2000 Ver2.0 CatchProgram = [3WIRE_ST]". The menu bar includes "ファイル(F)", "表示/編集(E)", "キャッチ開始(A)", "キャッチ設定(L)", and "ヘルプ(H)". Below the menu bar, it shows "TypeFile = [3wire_st.typ]" and "CommandFile = [test.CMD]". The main area contains a table with the following data:

No.	CH.	Bit	Time	Command	Info	D01	D02	D03	D04	D09
35	CH 1	32	5584ms	[Test Data] : Data9=0	32Bit	0xF8	0x00	0x04	0xEB	0x30
1381	CH 1	32	2045ms	[Test Data] : Data9=0	32Bit	0xF8	0x00	0x04	0xEB	0x30
3303	CH 1	32	4072ms	[Test Data] : Data9=#	32Bit	0xF8	0x00	0x04	0xEB	0x23
4620	CH 1	32	1842ms	[Test Data] : Data9=#	32Bit	0xF8	0x00	0x04	0xEB	0x23
5935	CH 1	32	2445ms	[Test Data] : Data9=#	32Bit	0xF8	0x00	0x04	0xEB	0x23

At the bottom of the window, it says "Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]".

- 【8】 No.2 データの D01 についてデータが 0xF8 であった場合に“[Test Data]:Data4=”の後に“MUTE”という文字列表示をさせたい場合の記述例とその結果について説明します。

— 記述例 —

[CH1_032]

\$F8hxxhxxhxxh=[Test Data] : Data1=%(S01,d01),,32Bit

[S01]

\$F8=MUTE

— 画面イメージ —

No.	CH.	Bit	Time	Command	Info	D01	D02	D03	D04
35	CH 1	32	5584ms	[Test Data] : Data1=MUTE	32Bit	0xF8	0x00	0x04	0xEB
1381	CH 1	32	2045ms	[Test Data] : Data1=MUTE	32Bit	0xF8	0x00	0x04	0xEB
3303	CH 1	32	4072ms	[Test Data] : Data1=MUTE	32Bit	0xF8	0x00	0x04	0xEB
4620	CH 1	32	1842ms	[Test Data] : Data1=MUTE	32Bit	0xF8	0x00	0x04	0xEB
5935	CH 1	32	2445ms	[Test Data] : Data1=MUTE	32Bit	0xF8	0x00	0x04	0xEB

Log Display Mode [Start No.=00001:End No.=09191] [Data Count = 9191/100000]



ハートランド・データ株式会社

〒326-0338 栃木県足利市福居町 361

TEL: 0284-22-8791(代表) / FAX: 0284-22-8792

URL: <http://www.hldc.co.jp/>