



データ可視器(Data Visualizer)ソフトウェア使用者の手引き

説明



データ可視器(Data Visualizer)はデータを処理して可視化するためのプログラムです。データ可視器は組み込みデバッカの データ交換器インターフェース(GDI)とCOMポートのような様々な供給元からデータを受け取る能力があります。

本書は一般の方々の便宜のため有志により作成されたもので、Microchip社とは無関係であることを御承知ください。しおりの[はじめに]での内容にご注意ください。

目	次
目	次

説明 •••	
1. 概要	
1.1. 助	けを得る・・・・・・・・・・・・・・・・・・・・・・・・4
1.2. 鍵	となる概念
1.2.1.	作業空間 ••••••• 4
1.2.2.	接続概要 •••••• 5
1.2.3.	組み込みデバッガのデータ交換器インターフェース・・・・・5
1.2.4.	簡単な転送 ・・・・・6
1.2.5.	端点 ······ 6
1.3. 7 [*] -	-9可視器開始 ······ 6
2. 外部	妾続・・・・・・・・・・・・・・・・・・・・・・・ 7
2.1. 7 [*] -	-9交換器インターフェース (DGI) ·········7
2.1.1.	SPI129-71-7 ···· 9
2.1.2.	USART1/ンターフェース ······ 9
2.1.3.	TWIインターフェース 10
2.1.4.	GPIOインターフェース 10
2.1.5.	電力インターフェース ・・・・・ 11
2.1.6.	コート特性分析 12
2.1.7.	吸い込みテータ変換 ······ 24
2.1.8.	$DGI_{7} - 9 \pi - 929 \cdots 24$
2.2. 79	アル ホート · · · · · · · · · · · · · · · · · · ·
3. 可祝1	28
3.1. 端	末
3.1.1.	
3.1.2.	· · · · · · · · · · · · · · · · · · ·
3.2. 🔀	表 ····································
3.2.1.	図表単位部 30 図ま構成乳空例 37
3.2.2.	凶衣伸风改足例
3.3. 4 7	ロハコーノ 42 北口7コーフ[°]単位 部 42
332	1/1/1 / 単位的
34 =	$h = \frac{1}{2} \int \frac{1}{12} \int \frac{1}{1$
3.4.1.	雷力デバッグ単位部 ······ 50
3.4.2.	基本的な電流測定 ・・・・・・・・・・・・・・・・・・・・・・・54
3.4.3.	カーソルを使う電力分析 ・・・・・・・・・・・・・・・・56
3.4.4.	⊐-トの相互関係 ····· 56
3.5. 独	自計器盤 ······59
3.5.1.	計器盤単位部 •••••• 59
3.5.2.	計器盤構成設定例 ······ 69
4. ユー テ イ	ש ד ר •••••• 74
4.1. 採	取速度計数器 ······ 74
4.2. 7 7	イル記録部 ・・・・・・・・・・・・・・・・・・・・・ 74
4.2.1.	2進ファイルに記録・・・・・・・・・・・・・ 74
5. 規約	75
5.1. 7 [*] -	-9流れ規約 ・・・・・・・・・・・・・・・・ 75
5.1.1.	構成設定形式 ······75
5.1.2.	流れ形式 ······ 76
5.1.3.	基本的な使い方 ····································
5.1.4.	自動構成設定 ······ 77
5.1.5.	目動構成設定例 ····································
5.1.6.	
5.1./.	1言亏按枕ノア1ル形式 86
5.2. Ati	melT - 外規約 Al
5.2.1.	Atmeir ー % 規約 を 1 史 つ 転达 87
0.Z.Z. 502	NDF [7] 88 約12-ジの法れ 105
0.2.0.	

5.2.4. メッセージ 形式 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・
5.2.5. メッセージ型・・・・・・・・・・・・・・・・・・・・・・・ 105
6 . ⊐ -ト 例断片 ••••••••••••••••••••••••••••••••••••
6.1 . データ ホ [°] ーリンク [°] コート [*] 例 ・・・・・・・・・・・・・・・・・・・・・・・ 121
6.1.1. 計器盤制御を使う応用情報交換 ・・・・・・・・ 123
6.2 . 端末コー [、] 例・・・・・・・・・・・・・・・・・・・・・・・・123
6.3 . 図表コート [*] 例・・・・・・・・・・・・・・・・・・・・・・・・126
6.3.1 . 基本的な図表
6.3.2 . 文字列標識追加 ······ 128
6.3.3. 水平カーソルを使つコート・・・・・・・・・・・・・・・・・・・・・・・・・129
6.5. 計器盤⊐-ト例 ····································
6.6 . 目動構成設定□-ト例・・・・・・・・・・・・・・・・・・136
7. 既知の問題 ······ 139
8 . 資料改訂履歴 · · · · · · · · · · · · · · · · · · ·
Microchip ካェフ
お客様への変更通知サービス・・・・・・140
お客様支援 ・・・・・ 140
Microchipデバイス コート 保護機能 ・・・・・・・・・・・・・・・・・・・・・・・140
法的通知 ····· 140
商標 •••••••••••••••
DNVによって認証された品質管理システム・・・・・・ 1/1
世界的な販売レサービア 141

1. 概要

本章はデータ可視器(Data Visualizer)の主な単位部/機能の概要を与えます。各単位部は単位部の技術的な詳細と共に独立した節 で記述され、単位部をどう使うかを示す例や使用事例を含みます。各節は自己完結型で、使用者に対して素早く識別して興味ある 節/単位部を選ぶことが可能です。

データ交換器インターフェース (DGI:Data Gateway Interface)

データ交換器インターフェース(DGI)は汎用入出力(GPIO)監視、電力測定、コート、特性分析に加えて、SPI、I²C、USART上の双方向通信を許します。

Power Deb	agger Data Gate	way				Disconnes
15220000434						Start
ा झ 🏠 () झ	El USART 🟠	Птм 🛱 6 0	□ GP10 🛱 1 G== 0 1 G== 0 2 G== 0	Power 🔯 A Current 🍘 A Yokage 🍘 B Current 🍘	Code Profiling 🛱	

シリアル ホ[°]ート (Serial Port))

シリアルポートはシステム上の何れかのシリアルポートと通信します。

EDBG Virtual COM Port (COM127)		Connect
EDBG FIItual Com Fort (ComitEr)		DTR 🔲 RTS
Baud rate Par	ty Stop bits	Open Terminal Autodetect protoco

端末(Terminal)

端末は簡単な文や数値を表示して送ります。

timeout: 9000, (); //show manage profile link	successi	function	(data)	(uery(*#1iH	saderLinkMana	geProfile").	show
el").show(); //show myAtmel ifCustomerGUIDISRetriev if (c.which !== 8 85 uery("#hdnOrderingCode							
Char 🖉	del tel a III	Mauradania	al Maluer	III Chau Tim	IN A A	and a the face	I to De

図表(Graph)

図表はデータ元の値対時間を作図するのに使うことができます。

- ・応用タイミング(例えば、PWM周波数)を測定するためのカーソル(時間軸)
- ・応用の設定点や閾値を制御するための水平カーソル(データ値)
- ・帯域は独自設定可能な閾値以上の時間区間を強調表示します。
- ・文字列標識は図表にされた事象に説明文を追加するのに使うことができます。

オシロスコーフ[°] (Oscilloscope)

オシロスコープ。

- ・上昇端または下降端での端または閾値起動
- ・単発または継続起動のための走行停止制御
- ・応用タイミング(例えば、PWM周波数)を測定するためのカーソル(時間軸)





電力デバッグ

電力デバッグ

- ・コード実行と電力消費の相互関係
- Power Debuggerを使って測定された電流と電圧を表示(いくつかのキットでの組み込みデバッカ)



独自計器盤 独自計器盤

・図表、セグメント表示器、2値信号、表題、釦、直線状計器(定義された範囲内の値)、円図表(例えば、無線応用で送信されたもの対パケット消失用)を使って使用者応用を可視化して制御するための独自使用者インターフェースを構築

ユーティリティ

- ・MCU周波数(例えば、送られたADC採取の速度)を確認するための採取速度計数器
- ・ファイル記録部は選択可能な形式のファイルにやって来る全てのデータを記録します。



Statistics 0	^ >
Speed: 62540 Samples/sec	
Volume: 318150 Samples	

.og to File		~ >
File		
Type CSV BIN	ASCII HEX Timestamp	
•		

1.1. 助けを得る

F1押下によって何時でもヘルプを開くことができます。構成設定(Configuration)ウィントウで単位部を選んでF1を押すことにより、ヘルプが 関連する節で自動的に開かれます。

1.2. 鍵となる概念

本項はデータ可視器(Data Visualizer)で作業する時に理解するための鍵となる概念を記述します。

1.2.1. 作業空間

データ可視器(Data Visualizer)は図表、インターフェース、制御のようないくつかの要素から成ります。これら全ての要素が作業空間を形成します。

この要素は単位部と呼ばれ、それらのどれも作業空間に追加することができます。



1.2.2. 接続概要

データ可視器(Data Visualizer)は組み込みシステムのMCUで走行しているファームウェアと通信します。ファームウェアの変数を両方向で転送す ることができます。以下の例では温度値が可視器に送られて図表が作図されます。可視器で摺動子を引き摺ることによって濾波器強 度値が設定され、その後にMCUへ送られます。



1.2.3. 組み込みデバッガのデータ交換器インターフェース

Xplained Pro系統の基板は組み込みデバッガ チップを含みます。これはSPIやTWIのインターフェースまたは汎用入出力(GPIO)のどれかを 通してMCUに対して容易にデータ可視器(Data Visualizer)と通信させるデータ交換器インターフェース(GDI)を持ちます。



データ可視器に於いてDGI制御盤(DGI Control Panel)は組み込みデバッガのデータ交換器インターフェースと通信する単位部です。USBケー ブルで基板がコンピュータに接続されると、それを制御盤で選ぶことができます。利用可能なインターフェースの一覧が現れます。チェック枠を チェックすることによってそれらの1つ以上を許可してください。上図ではSPIインターフェースが許可されています。MCUは今やSPIホートで データ可視器と通信することができます。

1.2.4. 簡単な転送

MCUからデータ可視器(Data Visualizer)への単一値の送信はかなり簡単です。下図ではMCUがSPIインターフェース上に温度変数を送ります。可視器では組み込みデ・バッガ上のSPIインターフェースが許可されています。組み込みデ・バッガはDGI制御盤(DGI Control Panel)を通して可視器へSPIデータを送信します。



温度データを可視化するために図表(Graph)が追加されています。SPIデータはDGI制御盤(DGI Control Panel)でSPIインターフェースからプラ グアイコンを引き摺ってそれを作図(Plot)領域で落とすことによって作図するように配線されます。これは(Graph)単位部に新しい作図を 追加します。

1.2.5. 端点

データ可視器(Data Visualizer)に於けるデータは端点(Endpoint)から始まり端点で終わります。端点は吸い込み口(Sink)と供給源(Source)として参照されます。データ供給源は接続された1つ以上の吸い込み口にデータを送ります。

作業空間では端点が下で示される図形記号によって表されます。



1.3. データ可視器開始

データ可視器(Data Visualizer)はAtmel Studioインストーラの一部として含まれ、Atmel Studio拡張または独立動作のどちらかとして走らせることができます。

Atmel Studio内側の拡張としてデータ可視器を走らせるには右のよう にツール(Tools)メニューでそれを選んでください。

データ可視器機能を支援するキットはAtmel Studioに於いてそれらの開 始頁で拡張へのショートカットを含みます。

データ可視器の独立版がインストールされている場合、Windows®のスタート メニューでショートカットを探してください。独立版はgallery.atmel.comから のダウンロートで入手可能です。

Too	ls	Window	Help			
>	Co	Command Prompt				
1	Pack Manager					
1	Device Programming Ctrl+Shift+P					
2	Ad	dd target				
\sim	Da	ata Visualizer				
	Code Snippets Manager Ctrl+K, Ctrl+B					
¢	Extensions and Updates					
	Atmel Gallery Profile					
	External Tools					
	Import and Export Settings					
	C	ustomize				
Ф	0	ptions				

2. 外部接続

外界への全ての接続は外部接続(External Connection)項目下で見つかる単位部によって扱われます。

DGI制御盤(DGI Control Panel)はデータ交換器インターフェースを持つどのツールとも通信します。これは汎用入出力(GPIO)監視、電力測定、コート、特性分析に加えて、SPI、I²C、USART上での双方向通信の能力があります。

シリアル ポート制御盤(Serial Port Control Panel)はシステム上のどのシリアル ポートとも通信します。

2.1. データ交換器インターフェース (DGI)

データ交換器インターフェース(DGI:Data Gateway Interface)は組み込みディッカがを持つ殆どのキットで利用可能です。DGI制御盤はDGIディ イスと通信することができます。下図はDGI制御盤単位部を示します。

図2-1. データ交換器インターフェース制行	御盤				
DGI Control Panel					^ X
Power Debugger Data Gateway				-	Disconnect Start
			A	DP Logging 🗌 Autodetect protocol:	Reset MCU
Interfaces:					
🗆 SPI 🔯 📄 USART 🔯 🗐 TWI 🔯 🔲 G	GPIO 🔯 🔲	Power 🔯	🔲 Code Profiling 🙀		
€ ~ 0 € ~ 0 0	🦛 🗿 A	Current 💕	Code Location		
1.	😂 🖉 A	Voltage 🌑			
2 (🥵 🖉 в	Current 🌔			
3 (🥵 🗝 🕚 В	Voltage			



助言:新しいDGI制御盤はデータ可視器で構成設定(Configura tion)タブの単位部(Modules)項目内の外部接続(Externa l Connection)で開くことができます。

\bigcirc	- Configuration
8	Modules ?
nfiguration	 External Connection Data Gateway Interface (DGI) Serial Port Visualization Utilities Protocols

検出された全てのDGIデバイスがキット名と通番と共に引き落とし一覧で一覧にされます。接続(Connect)釦を使うと、選んだDGIデバイス に接続して利用可能なインターフェースについて問います。利用可能なインターフェースがインターフェース(Interfaces)下で一覧にされます。インター フェースを許可するには名前傍らのチェック枠をチェックしてください。インターフェースが許可されると、供給源と吸い込み口を他の端点に接続 することができます。歯車釦はインターフェースを構成設定するのに使われます。構成設定領域の説明についてはインターフェース詳細部分 をご覧ください。

インターフェースからのデータのポーリングを開始するには開始(Start)釦をクリックしてください。MCUをリセット(Reset MCU)チェック枠は開始中に MCUをリセットに保持させます。

データ可視器(Data Visualizer)は自動構成設定に関してAtmelデータ規約(ADP:Atmel Data Protocol)とデータ流れ規約の2つの異なる規約を支援します。ADP使用時、構成設定は目的対象応用コートに存在し、目的対象応用はデータ可視器からの要求で構成設定の設定を送ります。データ流れ規約使用時、構成設定はホストコンピュータに格納されたファイルに存在し、目的対象応用はデータ可視器によってどの構成設定ファイルが読み込まれるかを識別するIDだけを送ります。ADPのより多くの情報については「Atmelデータ規約」をご覧ください。データ流れ規約のより多くの情報については「データ流れ規約」をご覧ください。

自動構成設定を許可するには規約自動検出(Autodetect protocols)任意選択が許可されなければなりません。

Autodetect protocols



データ可視器が有効な何れかの構成設定ファイルを見つけることができない場合、正しい構成設定ファイルが存在するフォルダへのパスを問 う検索ダイアログウインドウが現れます。



フォルダ選択後、フォルダは自動構成設定検索パスに追加されます。

助言:検索パスをリセットして検索パスとして新しいフォルダを選ぶには規約自動検出(Autodetect protocols)任意選択文のリンクをクリックしてください。 データ可視器はその後に構成設定ファイルが存在するフォルダへのパスを問う閲覧ダイアログを 飛び出させます。元の検索パスは解消され、新たに選んだフォルダが検索パスとして設定されます。

重要: 3つ全ての構成設定ファイルは同じフォルダに存在しなければなりません。

2.1.1. SPIインターフェース

SPIインターフェース供給源はSPIインターフェースで受信した生の値を含みます。吸い込み口はSPIハスで受信した値を送り返します。SPIインターフェースの物理的な部分の更なる詳細についてはSPIデータ採取に使われるデバッグ、ツールの使用者の手引きをご覧ください。

▶ 重要: SPI吸い込み口が複数バ小型を持つ供給源に接続される場合、そのバ小順は予測不能かもしれません。

重要: SPIハードウェア単位部はLow活性チップ選択(CS)信号を使います。CSピンがHighの時に送られたどのデータも無視されます。

SPI構成設定(SPI Configuration)ダイアログはDGI制御盤(DG I Control Panel)内のSPIインターフェースから開かれます。



SPI Configuration	—
Transfer Mode	SCK normally low, Read data on rising edge 🔻
Force startup synchronization on	cs 🗹
Enable timestamping	
	OK Cancel

表2-2. 構成設定

領域名	値	使用法
Transfer Mode (転送動作形態)	 SCK normally low, Read data on rising edge (SCK通常Low、上昇端データ読み込み) SCK normally low, Read data on falling edge (SCK通常Low、下降端データ読み込み) SCK normally high, Read data on falling edge (SCK通常High、下降端データ読み込み) SCK normally high, Read data on rising edge (SCK通常High、上昇端データ読み込み) 	SPI動作形態、クロック位相と採取を制御
Force synchronization on CS (CSで強制同期)	ONまたはOFF	SPIインターフェースはチップ。選択線が2度の交互切り替え後にだけ許可されます。
Enable timestamping (時刻印許可)	ONまたはOFF	データはDGI時刻印インターフェースを通して時刻印付けされます(より遅い転送速度にされます)。

関連リンク 吸い込みデータ変換

2.1.2. USARTインターフェース

USARTインターフェース供給源はUSARTインターフェースで受信した生の値を含みます。吸い込み口はUSARTインターフェースで受信した値を送り返します。USARTインターフェースの物理的な部分の更なる詳細についてはUSARTデータ採取に使われるデバッグ ツールの使用者の手引きをご覧ください。



重要: USART吸い込み口が複数バイト型を持つ供給源に接続される場合、そのバイト順は予測不能かもしれません。

USART構成設定(USART Configuration)がイアログはDGI制御盤(DGI Control Panel)内の USARTインターフェースから開かれます。



San USART Configuration	on 🛛
Baud rate	115200
Char length	8 bits 🔻
Parity type	None 🔻
Stop bits	1 bit 🔻
Synchronous mode	
Enable timestamping	
ОК	Cancel

表2-3. 構成設定

領域名	值	使用法	
Baud rate (転送速度)	0~2000000	非同期動作でのUARTインターフェース用転送速度	
Char length (文字長)	5、6、7または8ビット	各転送でのデータビット数	
Parity type(パリティ形式)	None(なし)、Even(偶数)、Odd(奇数)、 Mark(マーク)、Space(スペース)	通信に使われるパリティ形式	
Stop bits (停止ビット)	1、1.5または2ビット	停止ビット数	
Synchronous mode (同期動作)	ONまたはOFF	同期または非同期の動作を選択	
Enable timestamping (時刻印許可)	ONまたはOFF	データはDGI時刻印インターフェースを通して時刻印を付けられます(より遅い転送速度にされます)。	

関連リンク 吸い込みデータ変換

2.1.3. TWIインターフェース

TWIインターフェース供給源はTWIインターフェースで受信した生の値を含みます。吸い込みロはTWIインターフェースで受信した値を送り返します。TWIインターフェースの物理的な部分の更なる詳細についてはTWIデータ採取に使われるデバッグ ツールの使用者の手引きをご覧ください。

▶ 重要: TWI吸い込み口が複数バイ型を持つ供給源に接続される場合、そのバイ順は予測不能かもしれません。

TWI構成設定(TWI Configuration)ダイアログはDGI制御盤(DGI Control Panel)内のTWIイン ターフェースから開かれます。





表2-4. 構成設定

領域名	値	使用法
Address (アドレス)	0~127	TWI従装置アドレス
Speed (速度)	100000または400000	TWI従装置用速度設定。タイミングに使用
Enable timestamping (時刻印許可)	ONまたはOFF	データはDGI時刻印インターフェースを通して時刻印を付けられます(より遅い転送速度にされます)。

関連リンク 吸い込みデータ変換

2.1.4. GPIOインターフェース

汎用入出力(GPIO)インターフェース供給源はuint8型で、許可されたGPIOピンのビット値を含みます。パケット はピン切り替わり毎に送信されます。吸い込み口はGPIOピンに受信した値を送り返します。GPIOイン ターフェースの物理的な部分の更なる詳細についてはGPIOデータ採取に使われるデバック、ツールの使用者 の手引きをご覧ください。

GPIO構成設定(GPIO Configuration)ダイアログはDGI制御盤(DGI Control Panel)内のGPIOインターフェー スから開かれます。



SPIO Configuration
GPIO 0 Monitor 📝
GPIO 1 Monitor 🗵
GPIO 2 Monitor 📝
GPIO 3 Monitor 📝
GPIO 0 Output
GPIO 1 Output
GPIO 2 Output
GPIO 3 Output
Mode Bus 🔻
OK Cancel

表2-5. 構成設定

領域名	値	使用法
GPIO 0 Monitor (GPIO 0監視)	ONまたはOFF	GPIOピン0を監視
GPIO 1 Monitor (GPIO 1監視)	ONまたはOFF	GPIOピン1を監視
GPIO 2 Monitor (GPIO 2監視)	ONまたはOFF	GPIOピン2を監視
GPIO 4 Monitor (GPIO 4監視)	ONまたはOFF	GPIOピン3を監視
GPIO 0 Output (GPIO 0出力)	ONまたはOFF	GPIOピンの出力を許可
GPIO 1 Output (GPIO 1出力)	ONまたはOFF	GPIOピン1出力を許可
GPIO 2 Output (GPIO 2出力)	ONまたはOFF	GPIOピン2出力を許可
GPIO 4 Output (GPIO 3出力)	ONまたはOFF	GPIOピン3出力を許可
Mode (動作形態)	Pin(ピン)、Bus(ハス)、 Latched Bus(ラッチしたハス)	GPIOピンを、独立したピン、4ビット バス、またはGPIO3の上 昇端でラッチされた3ビット バスとします。

>

重要: ハス動作(BusまたはLatched Bus)のどれかの使用時に全てのGPIOが採取されますが、監視を許可されたそれらのGPI Oだけが採取を起動します。例えば、GPIO 0~GPIO 2の全てが禁止されたGPIO監視を持ちますが、GPIO 3は許可さ れた監視を持つ場合、GPIO値はGPIO 3変化時にだけ採取されますが、4つ全てのGPIO値がGPIO 3変化時に読まれ ます。

関連リンク 吸い込みデータ変換

2.1.5. 電力インターフェース

電力(Power)インターフェースは接続された回路の電力消費を測定します。電力インターフェースの物理的な部分の更なる詳細については電力測定に使われるデバッグ ツールの使用者の手引きをご覧ください。

電力構成設定(Power Configuration)ダイアログはDGI制御盤(DGI Control Panel)内のPowerインターフェースから 開かれます。



電力構成設定(Power Configuration)ウィントウは接続されるデバックブ ツールの能力に依存して変わります。

Enable B Channel	1		
Trigger calibration			
Enable Range Source			
Lock ChA to High Rang	je 🔳		
Enable Voltage Output			
Voltage Output		1600	mV
Averaging	No		•

表2-6. 電力構成設定任意選	択	
領域名	値	使用法
Enable B Channel (Bチャネル許可)	ONまたはOFF	第2電力測定チャネルを許可。Aチャネルは常に許可されます。
Trigger calibration (校正を起動)	ONまたはOFF	電流測定回路の校正手順を起動します。更なる詳細については「電力測定校正」をご覧ください。
Enable Range Source (範囲供給元許可)	ONまたはOFF	基本電力測定チャネルに使う範囲を示す範囲供給元を提供。電力消費測定に使われる物理的なハードウェアは測定された瞬間電流に応じて異なる構成設定を持ちます。各構成設定は範囲として参照されます。
Lock ChA to High Range (Aチャネルを高範囲に固定)	ONまたはOFF	Power Debuggerに於いて、Aチャネルは低範囲への自動切り替えを避けるために高範囲に固定化することができます。これは範囲間切り替え時に重要な採取を失うことなく、電流消費での短い尖頭波形の検出を許します。
Enable Voltage Output (電圧出力許可)	ONまたはOFF	出力電圧(Voltage Output)摺動子によって与えられる値でPower Debuggerの電 圧出力を許可します。
Voltage Output (出力電圧)	0∼5500 mV	Power Debuggerは目的対象応用に給電するのに使うことができる調整可能な目的対象供給が特徴です。この設定はこの供給の出力電圧を制御します。この設定を有効にするために電圧出力許可(Enable Voltage Output)任意選択が許可されなければなりません。

助言: どの構成設定変更も電力構成設定(Power Configuration)ウィンドウでOKをクリックするまで無効です。例えば、電圧出力を 許可するために電圧出力許可(Enable Voltage Output)任意選択がチェックされなければならず、出力電圧(Voltage Outp ut)値設定とその後のOK押下で、電圧出力が実際に許可されて摺動子値に従って設定されます。

助言: Aチャネル範囲固定は既に低範囲で走行している場合に高電流範囲に戻ることをデバッガに強制しません。変更を強制す るために充分高い電流を待つか、またはデバッガを単純に停止して再開始するかのどちらです。

重要:電力(Power)インターフェースは電力(Power)単位部でだけ使うことができます。オシロスコープ(Oscilloscope)単位部と図表(Graph)単位部はどちらも電力(Power)インターフェースを使うことができません。

2.1.6. コート 特性分析

コート、特性分析(Code Profiling)インターフェースはプログラムカウンタとメモリ位置のような内部データをアクセスするのに目的対象デバイスのデバッグ インターフェースを使います。これはプログラムカウンタアトレスの時刻印付けされた採取を提供し、デバイスのプログラム実行での見識を許しま す。使用者はポーリングするために任意のメモリアトレスを選んでそれらの位置でデータ変数を制御することもできます。加えて、スタックの状態と目的対象の節電/休止動作を監視することが可能です。最後に、目的対象のチップ。上デバック(OCD)システムでのメッセージ、パイプを通 して目的対象から任意のデータを受け取ることが可能です。

上の機能の有効性は目的対象デバイス型式で変わり、より多くの詳細は以降の項で見つけることができます。

2.1.6.1. コート 特性分析インターフェース

コート、特性分析(Code Profiling)インターフェースをどう構成設定して使うかの2つの例については「データ ホーリング例」と「プログラム カウンタ ホーリング」をご覧ください。

重要: コート、特性分析(Code Profiling)インターフェースはデータ可視器(Data Visualizer)がAtmel Studio内の拡張として動く時にだけ 利用可能です。これはAtmel Studioデバッカ 後処理部を通して目的対象デバイス上のデバッグ システムをアクセスすることが必 要だからです。

コート^{*}特性分析構成設定(Code Profiling Configuration)ウィント^{*}ウはDGI制御盤(DGI Control Panel)内のCode Profilingインターフェースを許可した後に開くことができます。



Sode Profiling Configuration	×
Enable Code Location	V
Enable Stack Monitor	
AVR OCD messaging	
AVR sleep monitor	
Add Memory Location	
OK Cancel	

表2-7. 構成設定

領域名	値	使用法
Enable Code Location (コート [・] 位置許可)	ONまたはOFF	プログラム カウンタ採取の状態を制御
Enable Stack Monitor (スタック監視許可)	ONまたはOFF	スタックの使い方を監視するためにスタック ポインタのポーリングを許可(UPDIを持つAV R [®] MCUのみ)
AVR MCU OCD messaging (AVR MCU OCDメッセーシ゛)	ONまたはOFF	OCDメッセージの経路をAtmel Studioへよりもむしろデータ可視器(Data Visualizer)へ許可
AVR MCU Sleep monitor (AVR MCU休止監視)	ONまたはOFF	MCUの休止状態の監視を許可(UPDIを持つAVR MCUのみ)
Add Memory Location (メモリ位置追加)		ポーリングと制御のために新しいメモリ位置の登録を追加。(16進数)アドレス入力用文字枠、データ型選択、それと削除(Delete)釦が現れます。

各構成設定任意選択は以降の項で詳述されます。

2.1.6.2. コート 位置

コート位置(Code Location)機能はデータ可視器(Data Visualizer)に目的対象デバイスのプログラム カウンタの採取を許します。これは様々な 採取点に於いて目的対象で何が実行されつつあるかを見るのを可能にします。電力消費と共にコート、実行を相関付けするのに特に 電力測定と共に有用です。目的対象が命令を実行しているのと同じ位にPC値を読み出すのが殆どの場合で不可能なため、採取さ れたPC値はコート、の一部を示すだけです。採取された値は何れかの時点でどのコート、部分が実行されつつあるかを示すのに未だ有用 です。

コード位置(Code Location)機能をどう使うかの例については「プログラム カウンタ ポーリング」をご覧ください。

重要: コード位置(Code Location)機能はUPDIプログラム/デバッグインターフェースを特徴とするSAMデベイスとAVRデベイスでだけ利用 可能です。

2.1.6.3. スタック監視

スタック監視(Stack Monitor)は走行時にそれらのコート・のスタックの使い方の監視を開発者に許します。これはチップ上のデバック「単位部経 由でスタック ポインタ レシ スタを採取することによって行われます。コート、特性分析構成設定(Code Profiling Configuration)タ、イアログでスタック 監視(Stack Monitor)を許可し、その後にスタック監視(Stack Monitor)供給源を図表(Graph)の作図(Plot)の吸い込み口に接続してデバッ グ作業を開始してください。

重要: スタック監視(Stack Monitor)機能はUPDIプログラム/デバッグ インターフェースを特徴とするAVRデバイスでだけ利用可能です。

スタック監視(Stack Monitor)機能はポーリングを使って実装され、これは全てのスタックレベルが見られる訳ではないことを意味します。結果の図の粒度(細かさ)はデバイス クロックの速度、UPDIクロック速度、それと応用コートの特質の相関関係です。スタック監視使用時はUPDIクロックを最高に設定することが推奨されます。

ここで示される例はATtiny817で走っている応用から(線引きなしの)点としてのスタック追跡です。点はアドレス16372(\$3FF4)を示す主繰り返しに於いて"アイドル状態"のスタック ポインタと関数が呼ばれるための減少を持つ採取(試料)を示します。



注: データ可視器(Data Visualizer)はデバイス上のスタックの構成の知識を持たず、故にスタックホインタの生の採取(試料)を示すだけです。

2.1.6.4. AVR MCU OCDメッセーシ

AVR MCU OCDメッセージ システムはチップ上ディ、ッグ単位部の側方経路です。これはそれが停止された時にコアと通信するためにいくつ かのOCD変種で広範囲に使われますが、走行動作中はシステムによって使われません。走行時にディ、ッガ ヘメッセージを送るために最 終使用者コートによって使うことができます。走行動作では走行/停止状態に対してディ、ッガが定常的にOCDをポーリングし、同時にどの メッセージも受け取ります。AVR MCU OCDメッセージは(ディ、ッグ ピンを除き)どの専用ピンも使わないコート計装用チャネルです。メッセージは単 ー8ビット値で、データ可視器(Data Visualizer)に経路付けされない限り、既定によってAtmel Studioに送られて16進値として出力(Outpu t)ウィントウで表示されます。

AVR MCU OCDメッセージはいくつかの方法で使うことができます。下の例は様々な技法の3つ例を示します。

・ハントシェークなし、無保証配送

・ハントジェーク付き、遮断移送

・ハントジェーク付き、非遮断移送

OCDメッセージを使う標準的な方法はありません。これらの例で示される技法は各々、利点と欠点を持ち、目的対象デバイス上の異なる 資源を利用します。全てのAVRがOCDメッセージを支援する訳ではなく、全ての応用がOCDメッセージの使用に適応される訳ではありま せん。これは本質的にチップ上デバッグシステムの側方経路です。

コート、特性分析構成設定(Code Profiling Configuration)ダイロクでAVR MCU OCDメッセージを許可し、その後にAVR MCU OCDメッセージ供給源を図表(Graph)の作図(Plot)の吸い込み口に接続してディック作業を開始してください。メッセージはそれによってコートが計測 されない限り現れません。

AVR MCU OCDメッセージに対する殆どの代表的な使用事例は例で実演されるように、端末で表示されるASCII printf形式のデバック です。けれども、何れの8ビットデータ値や複合構造体さえ移送するのに使うことができます。メッセージは例えば、ADC採取完了割り込 みから送ることができ、ADC採取の8ビット値を直接OCDメッセージレジスタに書きます。これはその後にデータ可視器(Data Visualizer)で 図に直接作図することができます。

"既定の'Atmel StudioへのOCDメッセージ チャネルは50ms周期の固定採取速度で動きます。データ可視器から許可されると、ポーリング繰り返しはOCDメッセージを読んで移送するためにデバッガの'予備'周期を利用します。これはより高い単位処理量をもたらしますが、タジングでの決定性も減らします。

2.1.6.4.1. ハント・シェークなしのAVR MCU OCDメッセーシ

AVR MCU OCDメッセージを使う最も簡単な形式はどんな形式のハンドシェークもなしにレジスタへ直接書くことです。これは例えば、データ完 了性よりも実行速度がもっと重要な時に適切かもしれません。例えそれがデバッガによって未だ読まれていなくても、OCDメッセージレジ スタへの単一書き込みは前の値を上書きします。これはゆっくり変化するデータに使うことができます。

以下の例は様々なAVR MCU基本設計に於いてハンドシェークなしでAVR MCU OCDメッセージをどう出力するかを示します。

AVR UPDI目的対象デバイスでのOCDメッセージ

```
// AVR UPDI目的対象でのOCDメッセージの例
// ハントジェークなし、無保証
#define SYSCFG_OCDM SYSCFG.reserved_0x18
void ocd_putchar (char c)
{
    SYSCFG_OCDM = c;
}
```

AVR XMEGA[®]目的対象デバイスでのOCDメッセージ

注: DGIに基づくOCDメッセージは未だXMEGA目的対象で支援されません。ここで示されるコードはOCDメッセージをAtmel Studioへ押し 込みます。

```
// AVR XMEGA目的対象でのOCDメッセージの例
// ハンドシェークなし、無保証
void ocd_putchar (char c)
{
     OCD. OCDR0 = c;
}
```

AVR JTAG目的対象デバイスでのOCDメッセージ

```
// AVR JTAG目的対象でのOCDメッセージの例
// ハントジェークなし、無保証
void ocd_putchar (char c)
{
    OCDR = c;
}
```

```
2.1.6.4.2. ハント・シェークと遮断でのOCDメッセーシ
この例は新しい文字を受け入れる準備が整うまでOCDメッセーシ゛システム経由で送られた各文字を阻止します。テ゛ヾッガが切断された場
合にコードの完全な固着停止を防ぐために簡単な制限時間が使われます。この例はUPDIインターフェースを使ってAVR ATtiny817で走り
ますが、OCDメッセージを支援する他のAVR MCU基本設計でも同様の機構を使うことができます。
#include <avr/io.h>
#include <stdbool.h>
#define SYSCFG_OCDM SYSCFG.reserved_0x18
#define SYSCFG_OCDMS SYSCFG.reserved_0x19
bool ocd_print_ready (void)
{
   return ! (SYSCFG_OCDMS & (1 \ll 0));
                                            // 最後の文字が収集されたか?
}
bool ocd_print_char (char msg)
   // 簡単な制限時間機構
   uint8_t timeout = 0xFF;
   while (timeout-- && !ocd_print_ready());
   if (!timeout)
                                             // デバッガが収集に失敗なら、継続
        return false;
                                             // メッセーシを降ろす。
   SYSCFG OCDM = msg;
   return true;
void ocd_print (char* pmsg)
{ // メッセージ送出
   while (*pmsg) {
       if (!ocd_print_char(*pmsg++))
            return;
}
int main(void)
   ocd_print ("Hello World¥n");
                                            // OCDメッセージを送出
   while (1);
```

2.1.6.4.3. 割り込み駆動で緩衝したOCDメッセージ

AVR MCU OCDメッセージを使うもっと複雑な方法は、徐々にデバッガへ転送されるデータを注入することができるprintf関数に小さな入 出力緩衝部を伴います。printf緩衝部を周期的に処理するのに計時器割り込みが使われます。メッセージ、チャネルの準備が整っていて データが利用可能なら、各割り込みで文字が緩衝部から送られます。この例はJTAGインターフェースを持つmegaAVR®デバイスで走ります が、OCDメッセージを支援する他のAVRデバイス基本設計ではより小さな機構を使うことができます。

```
#include <avr/io.h>
#include <stdio.h>
#include <stdio.h>
#include <avr/interrupt.h>
// OCDメッセージに割り当てられる緩衝部
#define OCDR_BUFFER_SIZE 32
static uint8_t ocdr_buffer[OCDR_BUFFER_SIZE];
// 緩衝部位置指示子
static uint8_t head;
static uint8_t tail;
// テ`n`yガがメッセージを受信することを示すフラグ
```

```
static uint8_t debugger_attached = 1;
// 関数宣言
static int ocdr_putchar(char c, FILE *stream);
static FILE mystdout = FDEV_SETUP_STREAM(ocdr_putchar, NULL, _FDEV_SETUP_WRITE);
// 流れに文字を配置
static int ocdr_putchar(char c, FILE *stream)
{
   // デバッガが収集に失敗の場合、むしろ単に中止
   if (!debugger_attached)
        return 1;
   // 丸め付きで先頭位置指示子を進行
   uint8_t tmphead;
   tmphead = (head + 1);
   if (tmphead >= OCDR_BUFFER_SIZE)
        tmphead = 0;
   if (tmphead == tail)
    {
        // 溢れで中止
        debugger_attached = 0;
        return 0;
   }
   // データを追加
   ocdr_buffer[tmphead] = c;
   head = tmphead;
   return 1;
}
// 定期的にデータを送る計時器割り込み
ISR(TIMER0_OVF_vect)
{
   // データなしなら、継続
   if (head == tail)
        return;
   // 直前バイトが収集されていないなら、継続
   if (OCDR & 0x80)
        return;
   // 後尾位置指示子を進行
   uint8_t tmptail = (tail + 1);
   if (tmptail >= OCDR_BUFFER_SIZE)
        tmptail = 0x00;
   tail = tmptail;
   // デバッガヘデータ送出
   OCDR = ocdr_buffer[tmptail];
   // ホットプラグを許すために取り付けフラグをリセット
   debugger_attached = 1;
}
void ocdr_printf_init (void)
ł
   // 緩衝部位置指示子を0設定
   head = 0;
   tail = 0;
```

```
// TC構成設定、8MHzのDIV32は約1msでの溢れの刻時
    TIFR = (1 \ll TOVO);
   TIMSK = (1 \iff TOIEO);
    TCCR0 = (1 \iff CS01) | (1 \iff CS00);
    sei();
}
int main(void)
{
    // ポート初期化
    DDRB |= 0xFF;
    PORTB = 0x55;
    // 緩衝部初期化
    stdout = &mystdout;
    ocdr_printf_init();
    // 実演繰り返し
   uint8_t c = 0;
   while(1)
    {
         c++;
         PORTB = ^{\sim}c;
         printf("led %d¥n", c);
         // printf配給を保証するためにdelay>約8msでなければなりません。
         uint16_t delay = 0x3FFF;
         while (delay--);
```

2.1.6.5. AVR MCU休止監視

AVR MCU休止監視は走行時にAVR MCU CPUの休止動作状態を監視することを開発者に許します。休止動作は2値表現でどの低 電力動作(アイドル、パワーダウンなど)が活性かは示しません。AVR MCU休止監視は休止動作でCPUが費やす大まかな時間の量を判 断するのに有用で有り得ます。コート、特性分析構成設定(Code Profiling Configuration)ダイログでAVR MCU休止監視を許可し、その後 にAVR MCU休止監視供給源を図表(Graph)の作図(Plot)の吸い込み口に接続してディッグ作業を開始してください。

重要: AVR MCU休止監視(AVR MCU Sleep Monitor)機能はUPDIプログラム/デ・バック インターフェースを特徴とするAVRデ・バイスで だけ利用可能です。

AVR MCU休止監視機能はポーリングを使って実装され、これは全ての休止遷移が見られる訳ではないことを意味します。結果の図の 粒度(細かさ)はUPDIクロック速度と応用コードの特質の相関関係です。AVR MCU休止監視使用時はUPDIクロックを最高に設定すること が推奨されます。

下の図は使用中の休止監視の例を示します。'1'の値はMCUが休止動作であることを示し、'0'はそれが普通に走っていることを意味します。作図から、MCUが約2.2秒の期間で休止動作に入って抜け出し、各起き上がり周回で約275ms間'起きている'状態に留まることを(カーソルを使って)測定することができます。



2.1.6.6. データ ポーリングと制御

データボーリングと制御(Data Polling and Control)機能は目的対象デバイスで任意のメモリ位置の継続的な採取と変更を可能にします。この機能をどう使うかの例についてはデータボーリングと制御例をご覧ください。

重要: データ ホーリングと制御(Data Polling and Control)機能はSAMデバイスでだけ利用可能です。

ポーリングや制御をされるメモリ位置を追加するには以下を行ってください。

- すべきこと: ・ 追加する各メモリ位置に対してメモリ位置追加(Add Memory Location)釦を クリックしてください。
 - ・各位置のアドレスと形式を満たしてください。

各メモリ位置には1つの供給源と1つの吸い込み口があります。その位置の値を監視するには供給元をどれかの可視化単位部に接続し、メモリ位置の値を変えるにはどれかの供給源を吸い込み口に接続してください。

· v ' _o
Code Profiling
Code Location 🧲
0x200000EC 🌔 😐 💿
0x200000E8 🌑 👄

Enable Code Location	4			
0x200000ec	uint32	•	Write to MCU	Delete
0x200000e8	uint32		Write to MCU	Delete
Add Watch				
	Ok Can	cel		

重要: ポーリングに於いて関心のある変数をvolatileとして宣言することはそれらがSRAMに置かれコンパイラによってそれらの値が レジスタにキャッシュされないことを保証します。レジスタはポーリングできず、SRAM位置しかできません。

助言: データ ポーリングは絶対SRAM位置で動作します。それらが常にSRAM内の同じ位置で利用可能なようにこの目的に対して 全域変数を使うことが勧められます。スタック内のポーリンク位置はポーリング時のスタック前後関係に基づいて予測不能な結果 を生じ得ます。

2.1.6.6.1. データ ホ[°]ーリンク 例

電力消費分析のためにプログラム カウンタ採取をどう使うかの例は「プログラム カウンタ ホーリング」で見つけることができます。この項で使われ るのと同じ大容量記憶クラス例がデータ ホーリングとデータ変数の制御機能をどう使うのかの例としても適応されます。SAM L21 Xplained P ro基板はキット上の目的対象USBとデ・バッグ USBの両コネクタを通してホスト コンピュータに接続されます。ATSAML21目的対象デ・バイスはSAM L21 Xplained Proに対するASFからのUSB装置MSC例を走っています。

この例で使われるハードウェア構成設定と目的対象応用コードのより多くの詳細については「データホーリングコード例」をご覧ください。

この例が図表(Graph)と計器盤(Dashboard)の単位部を使うとは言え、原理はデータ可視器(Data Visualizer)で他の単位部と共にコート、特性分析(Code profiling)インターフェースを使うのと同じです。

最初に、図表(Graph)が目的対象応用内の変数を監視するように構成設定されます。

すること: ・ DGI制御盤(DGI Control Panel)内のコート、特性分析(Code Profiling)インターフェースに対する チェック枠をチェックすることによってコート、特性分析インターフェースを許可してください。



すること: ・ 歯車釦を押すことによってコード特性分析構成設定(Code Profiling Configuration)ウィンドウを開いてください。

Scode Profiling Configuration	×
Enable Code Location	
Enable Stack Monitor	
AVR OCD messaging	
AVR sleep monitor	
Add Memory Location	
OK Cancel	

Data Visualizer



すること:追加の軸を追加するために軸追加(Add axis)釦をクリックしてください。

A	dd axis S	croll by plots	. Q Q	Automatically f	it Y Scroll the m the left shift	ouse-wheel while pressing and ho key to zoom in the time axis.	alding	
•	Axis 0							
	New plot	New band	New string	Add Horiz, Cursor	Delete Axis			
•	Aois 1							
	New plot	New band	New string	Add Horiz, Cursor	Delete Axis			

これらは共に接続されるべき新しい供給源(変数)と吸い込み口(軸)です。





すること:実行を再開するためにAtmel Studioで継続(Continue) (F5)をクリックしてください。

助言: 停止状態のUSB装置はもはやWindows事象に応答せず、長すぎる間この状態を保たれた場合、パスから切断されるかも しれません。これを救済するには単にAtmel Studioで実行をリセットしてください。

データ可視器(Data Visualizer)で図表での出力を観察してください。ディスクをフォーマットして書き込み周回計数器がどう増えるかを監視し てください。両方の値が独立した軸で作図され、故にそれらはそれによって尺度調整することができます。出力はこのようなものに見 えるべきです。



この例の以降の部分は目的対象応用と情報を交換するために計器盤をどう使うかを示します。目的対象応用で必要とされるコード変 更のより多くの情報については「計器盤制御を使う応用情報交換」をご覧ください。

Data Visualizer



 すること: ・構成設定(Configuration)操作盤を開いてください。 ・入出力計器盤(I/O Dashboard)をダブルクリックすることによって新しい入出力計器盤部品を追加してください。 	Configuration Modules External Connection DGI Control Panel Serial Port Graph Terminal Protocols Utilities I/O Dashboard Logging
---	--

今や計器盤に摺動子(Slider)を追加することができます。

ひかん すること:・編集(Edit)チェック枠を選んでください。

- ・要素(Elements)タブを開いてください。
 - ・計器盤にSlider(摺動子)要素を引き摺ってください。

Aa Label
12 Numeric Input
ab Button
Crit Radio Group
Check Box
E Slider
Sig 1
Progress bar
Segment Display
 Graph 👻

摺動子(Slider)制御はいくつかの構成設定パラメータを持つことが必要です	かの構成設定パラメータを持つことが必要です。
--	------------------------

すること: 摺動子(Slider)要素を選択してそれの特性を設定してください。

- ・最大(<u>Maximun</u>) = 500
 - 最小(Minimum) = 100

Edit Show Gri Configuration Z-Index Left	ines Show Endpoints Load Save	
Тор	25	
Width	267	
	31	
Height		
Height Minimum	100	
Height Minimum Maximum	100	

今や計器盤にセグメント表示器(Segment Display)制御を追加することができます。

すること:・編集(Edit)チェック枠を選んでください。

- ・要素(Elements)タブを開いてください。
- ・計器盤にセグメント表示器(Segment Display)要素を引き摺ってください。

Dashboard I/O	* ×
	Aa Label
	Progress bar

セグメント表示器(Segment Display)制御はいくつかの構成設定パラメータを持つことが必要です。

すること: セグメント表示器(Segment Display)要素を選択してそれの特性を設定してください。

・桁数(Segment Count) = 3

0		•				
Edit C Show Gridling	es 🔲 Show Endpo	nts	Load Save			
Edit Show Gridlin Configuration Z-Index	es 📃 Show Endpo	nts	Load Save			
Edit Shew Gridlin Configuration Z-Index Left	es Show Endpe	nts	Load Save	 		
Edit Show Gridlin Configuration Z-Index Left Top	es Show Endpo 0 322 17	nts	Load Save			
Edit Show Gridlin onfiguration Z-Index Left Top Width	es Show Endpo 0 322 17 179	nts	Load Save	 		
Edit Show Gridlin ionfiguration Z-Index Left Top Width Height	es Show Endpo 0 322 17 179 100	nts	Load Save			
Edit Show Gridlin Configuration Z-Index Left Top Width Height Segment Count	es Show Endper 0 322 17 179 100 3	nts	Load Save			
Edit Show Gridlin Configuration Z-Index Left Top Width Height Segment Count Numeric Base	es Show Endpo 0 322 17 179 100 3 10	nts	Load Save			

摺動子(Slider)制御は今やデータ可視器(Data Visualizer)に於いて関連するどれかの吸い込み口に接続することができる供給源として 使うことができます。セグメント表示器(Segment Display)は同様に関連する供給源を接続するための吸い込み口として使うことができま す。

コート`特性分析(Code Profiling)データ ポーリンク インターフェースはデータの供給源とデータの吸い込み口の両方を提供します。 摺動子は今や 吸い込み口に、セクメント表示器は供給源に接続することができます。

すること:・編集(Edit)チェック枠のチェックを外してください。

- ・端点を表示(Show Endpoints)チェック枠を選んでください。
 - ・各供給源プラグを引き摺ってそれを吸い込み口で落とすことによって供給源と吸い込み口を接続してください。

nterfaces:		ADP Logging Autodetect ADP Reset M
E SPI 🙀 E USART 🙀 E TWI 🙀 E GPIO 🗱	Power Image: Code Profiling A Current Code Location A Voltage 0x20000E8 B Current Image: Code Profiling B Voltage Image: Code Profiling	
ishboard I/O	ורזרז	^
¢~		
* **	JUU	
<u></u>	JUU	

今やデータ可視器で接続が作られたので、システムを走行状態に置いてGUIを通して変数との情報交換を行うことができます。



- ・データ可視器(Data Visualizer)を開始してください。
- ・Atmel Studioで実行を再開(F5)してください。

摺動子は今や応用コート内のframe_comparator変数の制御装置です。 摺動子を引き摺ってLED点滅周波数の変化に気付いてください。 摺動子位置でのどの変化もディック インターフェースを通して目的対象ディイスへ送られ、新しい値が変数に格納されます。 同時に、 値は目的対象からも読み戻されてセグメント表示器で表示されます。

Dashboard I/O		^ X
[]	3 18	

2.1.7. 吸い込みデータ変換

DGIは元々8ビット値だけしか扱うことができないため、DGIによって受信された全ての値は以下の表での規則に従って再割り当てされます。

<u>表2−8. ⊤−9変換</u>	
データ型	
int8	uint8に割り当て。2の補数値は維持されます。
uint8	
int16	2の補数値は維持されます。2つのuint8値に分割。ビッグ エンディアン。
uint16	2つのuint8値に分割。ビッグェンディアン。
int32	2の補数値は維持されます。4つのuint8値に分割。ビック゛エンディアン。
uint32	4つのuint8値に分割。ビッグェンディアン。
Float	int32に割り当て。
Double	int32に割り当て。
XY8	X値が最初に送られ、その後にY軸。
XYu8	X値が最初に送られ、その後にY軸。
XY16	X値が最初に送られ、その後にY軸。
XYu16	X値が最初に送られ、その後にY軸。
XY32	X値が最初に送られ、その後にY軸。
XYu32	X値が最初に送られ、その後にY軸。
XYFloat	X値が最初に送られ、その後にY軸。
XYDouble	X値が最初に送られ、その後にY軸。
String	各文字のASCII値が送られます。Null終端が追加されます。
StringFloat	後続する文字列を持つint32として送出。
Boolean	偽は0として真は1として送られます。

2.1.8. DGIデータ ホーリング

データ交換器インターフェース(DGI)との通信は独立したC++ DLLを通して行われます。作業が開始されると、2ms毎にデータのためにDGIデ バイスをポーリングします。けれどもCPUが他の作業で多忙かもしれないので、このポーリングはより長い間隔で起こるかもしれません。

DGIデバイスが制限された緩衝部しか持たないため、DLLは溢れるのを避けるためにデバイスを定期的にポーリングすることが必要です。 従って、ポーリング作業中はCPUの使用を低く保つことが重要です。溢れ問題の場合、DGIインターフェースでの転送速度を減らすか、また は応用を中止することによってCPU負荷を減らすかのどちらかを行ってください。

2.2. シリアル ホ[°]ート

データ可視器(Data Visualizer)は標準PCシリアルボートに接続することができます。既定によってシリアルボート制御盤(Serial Port Control Panel)が開かれ、データ可視器を開始する時にDGI制御盤(DGI Control Panel)下に最小化されます。これを展開するには最小化した盤の右角の下向き矢印をクリックしてください。

DGI Control Panel	^ × .
	Connect Start
ADP Logging Autor	detect protocols 🔲 Show Config search path 📄 Reset MCU
Serial Port Control Panel	
助言 : データ可視器で構成設定(Configuration)タブの単位部(Modul es)項内の外部接続(External Connection)内で新しいシリアル ポート(Serial Port)制御盤を開くことができます。	Configuration Modules External Connection DGI Control Panel Serial Port Graph Terminal Protocols Utilities I/O Dashboard Logging

ボーレート(Baud rate)、停止ビット(Stop bits)、パリティ(Parity)は通信相手に対して必要とされる設定と一致するように設定されなければな りません。吸い込み口と供給源の端点はシリアルポートに対して出て行くデータとやって来るデータを表すために存在します。シリアルポート 制御盤の端点はuint8データ型で、DGI制御盤と同じ変換規則に従います。端末を開く(Open Terminal)チェック枠は自動的に開いて端 点に接続することを端末単位部にさせます。シリアルポートから切断すると、作成された端末単位部が閉じられます。

図2-2. シリアル ポート制御盤	
Serial Port Control Panel	^ X
EDBG Virtual COM Port (COM127)	Connect
Baud rate Parity Stop bits	Open Terminal Autodetect protocols
9600 None 🔻 1 bit 🔻 🌑 💿	

表2-9. 構成設定		
領域名	值	使用法
Baud rate (ボーレート)	600~2000000	シリアル インターフェースのボーレート
Parity (パリティ)	None(なし)、Even(偶数)、Odd(奇数)、 Mark(マーク)、Space(スペース)	通信に使われるパリティ形式
Stop bits (停止ビット)	1、1.5または2ビット	停止ビッ教
DTR (データ端末準備可)	ONまたはOFF	RS-232直列通信のデータ端末準備可(DTR)制御信号
RTS (送出要求)	ONまたはOFF	RS-232直列通信の送出要求(RTS)制御信号
Open Terminal (端末を開く)	ONまたはOFF	シリアル ホート制御盤(Serial Port Control Panel)のシリアル ホートと端 末間の自動接続で供給源と吸い込み口の接続で端末を開きま す。
Autodetect protocols (規約自動検出)	ONまたはOFF	Atmelデータ規約の自動検出またはデータ流れ規約自動構成設定。規約のより多くの情報については「Atmelデータ規約」と「データ流れ規約」をご覧ください。
Show Config search path (構成設定検索パス表示)	ONまたはOFF	規約自動検出(Autodetect protocols)が許可される時にだけ利用可能。データ流れ自動構成設定ファイル用検索パスを表示。

データ可視器(Data Visualizer)は自動構成設定に関して、Atmelデータ規約(ADP:Atmel Data Protocol)とデータ流れ規約の2つの異なる 規約を支援します。ADP使用時、構成設定は目的対象応用コートに存在し、目的対象応用はデータ可視器からの要求で構成設定を 送ります。データ流れ規約使用時、構成設定はホストコンピュータに格納されるファイルに存在し、目的対象応用はデータ可視器によって読み 込まれるべき構成設定ファイルを識別するIDを単に送ります。ADPのより多くの情報については「Atmelデータ規約」をご覧ください。データ 流れ規約のより多くの情報については「データ流れ規約」をご覧ください。

自動構成設定を許可するには規約自動検出(Autodetect protocols)が許可されなければなりません。

Autodetect protocols

接続(Connect)押下後、データ可視器は全てのインターフェースを許可すると同時にADPハント・シェークメッセーシ・またはデータ流れ構成設定ハ[°] ケットを探します。ADPハント・シェークメッセーシ・が受信されると、データ可視器は目的対象応用からの構成設定情報を要求します。データ流れ 構成設定ハ[°]ケットが見つかった場合、データ可視器は検出したIDに合う名前を持つ構成設定ファイルを探して自動構成設定検索ハ[°]スの フォルタ・を通して検索します。



重要: データ可視器がデータ流れ構成設定パケットを検出するのを確実にするため、秒毎に最低2度、目的対象によってそれが送られなければなりません。

重要: 非同期シリアル規約(例えば、DGI USARTとCDC仮想COMポート インターフェースで使われるUART規約)は自動検出のため以下のボーレートを使います。

表2-10. 規約の自動検出に関して非同期インターフェースで使われるボーレート

ホーレート 9600 19200 38400 57600 115200 230400 500000 1000000 2000000

この表にない何れかの速度を使うことは非同期インターフェース(DGI UARTとシリアル ポート/CDC仮想COMポート)上の規約の 自動検出に対して動きません。

助言:構成設定ファイルを探すためにデータ可視器によって使われる現在の検索パスを見るには構成設定検索パス表示(Show Config search path)任意選択をチェックしてください。

Show Config search path

検索パスはセミコロン(;)で分離されたパスの一覧です。データ可視器が自動構成設定IDを検出すると、正しいファイル名を持つ構成設定ファイルを探す一覧内のパスを通して検索します。

Config search path C:\Data Visualizer Config Files; C:\Data Stream Example Config Files;

データ可視器が有効な何れかの構成設定ファイルを見つけることができない場合、正しい構成設定ファイルが存在するフォルダへのパスを問う閲覧ダイアログウインドウが現れます。

Organize 🔻 🛛 New fo	lder		!≡ ▼ (
🔆 Favorites	Name	Date modified Type	Size
詞 Libraries		No items match your search.	
📜 Computer			
🛛 🏭 OSDisk (C:)			
🗣 Network			

フォルダ選択後、フォルダは自動構成設定検索パスに追加されます。

れます。

助言: 検索ハ[°]スをリセットして検索ハ[°]スとして新しいフォルダを選ぶには規約自動検出(Autodetect protocols)任意選択文のリンクをクリックしてください。 データ可視器はその後に構成設定ファイルが存在するフォルダ[°]へのハ[°]スを問う閲覧ダ[°]イアログを ポップアップします。元の検索ハ[°]スは解消され、新たに選んだフォルダ[°]が検索ハ[°]スとして設定さ



重要: 3つ全ての構成設定ファイルは同じフォルダに存在しなければなりません。

3. 可視化

やって来るデータは本章下に含まれる単位部を使って可視化することができます。

端末(Terminal)は生データまたはASCII符号化した文字のどちらかとして文字データとして表示します。文字に基づくデータを送る能力もあります。

図表(Graph)単位部は線描、帯域、文字列フラグとして時間に沿ってやって来るデータを可視化します。カーソル(Cursor)はデータを分析するのを手助けし、設定する閾値のための出力値を提供することができます。

オシロスコープ(Oscilloscope)単位部はデータの流れに於いて時間で繰り返される様式を分析するのに有用です。

電力分析(Power Analisys)単位部は特に時間に沿って電力消費を分析するために作られています。これはプログラム実行対電力消費の概要を得るためにプログラム カウンタ採取を可視化するコート、特性分析(Code Profiling)でも使うことができます。

独自計器盤(Custom Dashboard)単位部は応用に合わせて使用者インターフェースを作成するための独自設定可能な描画面です。図表などに加えて、釦、摺動子、チェック枠のような最も一般的な使用者入力が特徴です。

3.1. 端末

端末(Terminal)単位部は簡単な文字データや数値の表示と送出のためのありのままの端末です。

3.1.1. 端末単位部

端末(Terminal)単位部は簡単な文字データや数値の表示と送出に使われます。端末をどう構成設定するかの例については「端末構成設定例」をご覧ください。



3.1.1.1. 端末接続とデータ表示

データの流れは供給源と吸い込み口の端点を通して端末に接続されます。外部の供給源を端末の吸い込み口に落とすか、または端 末の供給源を外部の吸い込み口に引き摺って落としてください。端末の吸い込み口端点にやって来るデータは入力文字枠で表示さ れます。

3.1.1.2. データ送出

端末の供給源が吸い込みロ端点に接続されると、入力文字枠でデータを入力してEnterを押すことによってデータを送ることができます。文字枠で入力したものはどれも送信後に消去されます。文字枠は中断文字(例えば、\$55の生データ値送信に帰着する¥x55)の使用を支援します。

3.1.1.3. 16進数動作設定

データは通常、ASCII符号化されたデータの流れと仮定されます。データの16進値を表示するには16進数動作(Hexadecimal mode)チェック 枠を選んでください。

3.1.1.3. 入力文字枠の大きさ変更

入力文字枠は枠の下部をクリックして引き摺ることによって大きさ変更可能です。

3.1.2. 端末構成設定例

以下の例はSPIインターフェースを端末に接続する方法を示します。けれども、手続きは他の利用可能なデータ供給源の何れとも同じです。この例で使われる目的対象コートは「端末コート例」で見つけることができます。

すること: DGI制御盤(DGI Control Panel)で正しいツールを	選んでください。
DGI Control Panel	^ X
ATmega256RFR2 Xplained Pro ATML1784031800003140	Connect Start
	ADP Logging Autodetect protocols 🖉 Reset MCU
Interfaces:	

▶ すること: 選んだツールでDGIへの接続をするために接続(Connect)をクリックしてください。

ATmega256RFR2 Xplained Pro ATML1784031800003140	- Disconnect Start
nterfaces:	ADP Logging Autodetect protocols Reset MCU Code Profiling Code Location
3 🌓 🚳	

すること: · SPIチェック枠をクリックしてください。

・SPIチェック枠傍らの歯車釦をクリックすることによってSPI構成設定(SPI Configuration)ダイアログを開いてください。

ATmega256 ATML17840318000	RFR2 Xplained Pro	÷	Disconnect Start
	SPI Configuration	ß	Reset MCI
Interfaces:	Transfer Mode Force startup synchronization Enable timestamping	SCK normally low, Read data on rising edge on CS	

すること:・転送動作形態(Transfer Mode)をSCK normally low, Read data on rising edge(SCK通常Low、上昇端で読み込み) に設定してください。

・CSで始動同期を強制(Force start-up synchronization on CS)任意選択を許可してください。

すること:・構成設定(Configuration)操作盤を開いてください。

- ・データ可視器(Data Visualizer)に端末(Terminal)表示部を追加してください。
- ・DGI制御盤(DGI Control Panel)でインターフェースから供給源プラグを接続を作る端末用の吸い込み口に引き摺ってください。



すること:・作業を開始してください。

・Xplained Pro基板の釦(SW0)を押してください。 各釦押下で、基板上のLED0が交互切り替えしてメッセージが 端末に現れるでしょう。

時々、釦押下毎に複数のメッセージが現れます。これは釦採取ルーチンに何かの跳ね返り(チャタリング)抑制算法が必要とされることの兆候です。LED交互切り替えを監視するよりも端末出力を考察することによってこの問題を見つける方がずっと簡単です。

Terminal 0	
LED ON 1	
LED OFF 2	
LED ON 3	
LED OFF 4	
LED ON 5	
LED OFF 6	
LED ON 7	

3.2. 図表

図表(Graph)単位部は多用途の図表作図ツールです。

3.2.1. 図表単位部

図表(Graph)単位部は多用途の図表作図ツールです。大きな作図領域は1つの時間軸と1つ以上の値軸(Y軸)を持ちます。値軸はお互いに重なり合って積み重ねられます。図表をどう構成設定するかの例については「図表構成設定例」をご覧ください。



Y軸に追加することができる以下のような4つ形式の要素があります。

- ・作図(Plot)
- ・帯域(Band)
- ・文字列標識(String marker)
- ・水平カーソル(Horizontal cursor)

これらの各々の要素は以降の項で記述されます。

3.2.1.1. 図表構成設定操作盤

構成設定(Configuration)操作盤を通して、図表(Graph)単位部はシステムの残りに接続されます。ここではより多くの軸、作図、他の図表要素を追加することができます。ここでは供給源と吸い込み口を接続することによって図表要素も接続します。

図3-3. 図表制	川御部		
Add axis	Auto-scroll 2	Automatically fit Y 3	1. 軸追加(Add axis) 2. 自動スクロール(Auto-scroll)チェック枠 3. 自動Y適応(Automatically fit Y)チェック枠

3.2.1.1.1. 軸追加



図表にY軸を追加するために軸追加(Add axis)釦を押してください。それは作図領域に現れ、それの制御部が構成設定(Configuratio n)操作盤の釦に追加されます。

3.2.1.1.2. 軸削除

1. 削除を望む軸用の構成設定部分が隠されている場合、最初に矢印(V)アイコンをクリックすることによってそれを展開してください。

2. それの軸削除(Delete Axis)釦を押すことによって軸を削除してください。

3.2.1.1.3. 自動スクロールの許可と禁止

自動スクロールは最後に到着した採取(試料)を含むように作図領域を固定化します。自動スクロールが禁止される場合、マウスで時間軸を引き摺るか、またはスクロール輪によって手動でスクロールしてください。

自動スクロールは自動スクロール(Auto-scroll)チェック枠を選ぶことによって許可されます。

3.2.1.1.4. Y軸自動大きさ調整

自動Y適応(Automatically Fit Y)チェック枠がチェックされると、Y軸はその軸で作図の試料範囲全体の作図に適応するように自動的に拡大/縮小します。

3.2.1.2. 作図

作図(Plot)は変化している値を記述する曲線です。曲線はデータ供給元から受け取ったデータ採取(試料)間を描きます。試料は散発的 または固定間隔で到着し得ます。データ供給元が固定速度で採取することが既知なら、作図はこの採取速度に設定することができま す。このように、例え試料の送信で何らかの伸び縮みがあったとしても、曲線は正しく表示されます。試料が不規則な速度で来る場 合、採取速度を0に設定してください。これは試料の時刻印に従った時間軸に沿ってそれを試料の図表位置にします。図表内に複 数の作図がある場合、各作図はその作図に対して新しいデータが到着する時に更新します。

軸に作図を追加する時に、図表構成設定(Configuration)操作盤でその軸(Axis)下に新しい作図の作図(Plot)制御盤が配置されます。



3.2.1.2.1. 作図の追加と接続

作図(Plot)をデータ供給元に接続するにはデータ供給源プラグ記号を引き摺ってそれを新規作図(New plot)の吸い込み ロコネクタ記号で落としてください。



3.2.1.2.2. 作図禁止

図表の作図領域で作図の表示を停止するには作図の許可(Enable)チェック枠の選択を外してください。

3.2.1.2.3. 作図色変更

作図の線色は次のようにして変更することができます。

- 1. 作図の線色表示部をクリックしてください。
- 2. 開いたダイアログ枠で赤(Red)、緑(Green)、青(Blue)の摺動子を引き摺ることによって色を調節してください。
- 3. 作図の線と線色表示部が今や新しい色に変更されています。

3.2.1.2.4. 作図形式変更

作図形式は作図形式選択で選んだ形式を変更することによって線(Plot)と点(Points)に変更することができます。線(Plot)形式は継続 する線として作図した図表を表示する一方で、点(Points)形式は点としてだけ実際に描いた試料を表示します。

3.2.1.2.5. 固定採取速度でのデータ作図

データ供給元が固定速度で作図にデータを送る場合、作図の採取速度を設定することができます。採取速度(Sample rate)文字枠に数 値を入力して設定(Set)釦を押してください。

3.2.1.2.6. 時刻印付けされたデータの作図

データが不規則な間隔で到着する場合、採取(試料)が採取の時刻印を使って置かれたなら、図表はもっと正確な表示を提供します。 時刻印を使って作図するには採取速度(Sample rate)文字枠に0を入力して設定(Set)釦を押してください。

3.2.1.2.7. 作図削除

軸から作図を削除するには作図の制御操作盤で削除(Delete)釦を押してください。

3.2.1.2.8. カーソル

カーソル表示(Show Cursors)任意選択が許可される場合、作図領域に2つの垂直カーソルが現れます。このカーソルはマウスによって移動することができ、作図(Plot)制御盤がカーソルに関連するデータを表示します。

3.2.1.3. 帯域

帯域(Band)は帯域色で作図背景を強調表示する作図領域内の垂直記号です。例えば、温度読み取りの作図で、温度が或る値以上の場所の作図部分を強調表示する帯域を追加することができます。



帯域は最小と最大の限度を持ちます。帯域への入力がこれら2つの値の間なら、帯域が活性(ON)です。



3.2.1.3.1. 帯域の追加と接続

新しい帯域(Band)を追加してデータ供給元に接続するには、データ供給源プラグを引き摺ってそれを新規帯域(New band)の吸い込み口コネクタで落としてください。



3.2.1.3.2. 帯域色設定

帯域色表示部をクリックしてください。ダイアログ枠が開きます。RGB値を変更してOKを押してください。

注:帯域色変更時、その変更は既に図表内にある帯域領域には影響を及ぼしません。新しい帯域領域だけが新しい色を持ちます。

3.2.1.3.3. 反転帯域制限設定



最大限度が最小値よりも小さい値に設定される場合、帯域は反転した規則で動きます。今や、帯域は入力値が最大限度よりも小さい時、または入力値が最小限度よりも大きい場合に活性になります。

・最小と最大の値を入力し、最大値が最小値よりも小さいことを確実にしてください。設定(Set)釦を押してください。

3.2.1.3.4. 帯域色設定

帯域色表示部をクリックしてください。ダイアログ枠が開きます。RGB値を変更してOKを押してください。

注:帯域色変更時、その変更は既に図表内にある帯域領域には影響を及ぼしません。新しい帯域領域だけが新しい色を持ちます。

3.2.1.3.5. 帯域削除

帯域を取り去るには帯域の制御盤で削除(Delete)釦を押してください。

3.2.1.4. 文字列標識

供給元が文字列を送ると、文字列標識(String Marker)がそれらの短いメッセージを図表に付着させます。これらの標識は採取(試料)の時刻印に従って配置されます。



3.2.1.4.1. 文字列標識の追加と接続

新しい文字列標識(String Marker)を追加してデータ供給元に接続するには、データ<mark>供給源</mark>プラグを引き摺ってそれを 新規文字列(New string)の<mark>吸い込</mark>み口コネクタで落としてください。

N	ew string

3.2.1.4.2. 文字列標識色設定

文字列(String)制御盤で文字列色表示部をクリックしてください。RGB値を変更してOKを押してください。

注: 文字列標識色変更時、その変更は既に図表内にある文字列標識には影響を及ぼしません。新しい文字列標識だけが新しい色を持ちます。

3.2.1.4.3. 文字列標識の展開と折り畳み

大きな文字列が文字列標識に送られると、標識は作図領域でそれが 占有する空間を減らすために小さな箱に折り畳みます。 文字を見るにはそれが展開されなければなりません。

・標識をダブル クリックすることによって文字列標識を展開または折り畳 みを行ってください。 図3-10. 文字列標識の展開と折り畳み



3.2.1.4.4. 文字列標識削除

文字列標識を取り去るには文字列の制御盤で削除(Delete)釦を押してください。

3.2.1.5. 水平カーソル

水平カーソル(Horizontal cursor)は上下に引き摺る時に供給元(Source)として使うことができる値を出力する図表内の水平線です。



助言:応用の設定点や閾値を制御するのに水平カーソルを使ってください。



3.2.1.5.1. カーソル接続

カーソルをデータ吸い込みロに接続するにはカーソルのデータ供給源プラグを引き摺ってそれを目的対象のデータ吸い込みロコネクタで落としてください。

3.2.1.5.2. カーソル値変更

カーソル値を変更するにはカーソル線上にマウスを置いてください。マウス カーソルが取っ手に変わります。クリックしてカーソルを新しい位置に引き摺ってください。

代わりに、カーソル値は水平カーソル(Horizontal Cursor)構成設定の値(Value)領域で新しい値を入力することによって変えることができます。文字箱枠の外側でマウスをクリックすることによって文字箱が不活性にされるまで変更が有効にならないことに注意してください。
3.2.1.5.3. カーソル表題変更

カーソルの表題を変更するには水平カーソル(Horizontal Cursor)構成設定の表題(Label)領域で新しい表題を入力してください。文字枠の外側でマウスをクリックすることによって文字枠が不活性にされるまで変更が有効にならないことに注意してください。

3.2.1.5.4. カーソル色変更

カーソル(Horizontal Cursor)制御盤でカーソル色表示部をクリックしてください。ダイアログ枠が開きます。RGB値を変更してOKを押してください。

3.2.1.5.5. カーソル削除

カーソルを取り去るにはカーソル(Horizontal Cursor)制御盤で削除(Delete)釦を押してください。

3.2.1.6. 拡大縮小と平行移動

自動スクロール(Auto-scroll)と自動Y適応(Automatically fit Y)のチェック枠がチェックされると、最後の採取(試料)が表示されてY軸は全ての 値が見えるように拡大縮小されます。

手動で拡大、縮小、またはもっと詳細に区域を調べるために、これらの任意選択を禁止してマウスを使って拡大縮小と並行移動を行ってください。

3.2.1.6.1. X軸の拡大縮小

X軸は以下のように2つの異なる方法で拡大縮小することができます。

・マウスのスクロール輪を使用

1.1. 作図領域内の何処かをクリックしてください。

- 1.2. キーホートでShiftキーを押し続けてください。
- 1.3. マウスの輪をどちらかの方向へスクロールしてください。

X軸はマウス カーソルの辺りを中心にして(マウスの輪を回す方向に応じて)拡大または縮小します。

・X軸大きさ変更標識を引き摺り

2.1. マウス カーソルをX軸の大きさ変更標識の1つの上に置いてください。マウス カーソスが水平大きさ変更矢印に変わります。 2.2. クリックして水平方向に引き摺ってください。

3.2.1.6.2. Y軸の拡大縮小

Y軸は以下のように2つの異なる方法で拡大縮小することができます。

- ・マウスのスクロール輪を使用
- 1.1. 作図領域内の何処かをクリックしてください。
- 1.2. キーボートでCtrlキーを押し続けてください。
- 1.3. マウスの輪をどちらかの方向へスクロールしてください。

Y軸はマウスカーソルの辺りを中心にして(マウスの輪を回す方向に応じて)拡大または縮小します。

・X軸大きさ変更標識の引き摺り

2.1. マウス カーソルをX軸の大きさ変更標識の1つの上に置いてください。マウス カーソスが垂直大きさ変更矢印に変わります。 2.2. クリックして垂直方向に引き摺ってください。

3.2.1.6.3. 平行移動

図表周りの平行移動は以下のように2つの方法で行うことができます。

- ・作図領域の引き摺り
 - 1.1. マウス カーソルを作図領域の内側に置いてください。
 - 1.2. マウスの左釦を押し続けてください。
 - 1.3. マウスを引き摺ってください。
- ・軸の引き摺り

2.1. マウス カーソルを軸の1つ上に置いてください。マウス カーソスが指差し手に変わります。

2.2. クリックして軸を引き摺ってください。

3.2.2. 図表構成設定例

本節は光感知器を持つ夜間動作切り替え器を実装する目的対象応用で使う図表(Graph)をどう構成設定するかの例を与えます。この例がデータ可視器(Daya Visualizer)で利用可能なデータ供給元のいくつかだけを利用するとは言え、その手続きは全てのデータ供給元に対して同じです。この例で使われる目的対象コート、とハート、ウェア構成設定の説明は「図表コート、例」で見つけることができます。構成設定例の最初の部分は「図表コート、例」節の最初の小節(「基本的な図表」)で見つかるコート、を使います。例の進歩のために目的対象応用コート、に対する変更が必要とされる時は対応するコート、一覧へのリンクが提供されます。



(New plot)と記された吸い込みロへ引き摺ってください。

	DGI Control Panel	~ >
External Connection	ATmega256RFR2 Xplained Pro	Disconnect
Terminal Graph Oscilloscope Power Debugging Custom Dashboard Utilities Protocols	ADP Logging Autodetect protocols] Reset MCl
Messages Xplained Pro. 12:02:40:987: Available interfaces on ATmega256RFR2 Xplained Pro is SPI, TWI, GPIO, Code Profiling. 12:10:33:502: Disconnected from ATmega256RFR2 Xplained Pro. 12:10:41:547: EDBG Device disconnected; ATmega256RFR2 Xplained Pro. 12:10:52:202: Connected to Power Debugger Data Gateway. 12:10:52:202: Available interfaces on Power Debugger Data Gateway. 12:15:54:226: Disconnected from Power Debugger Data Gateway. 12:15:54:226: Connected from Power Debugger Data Gateway. 12:15:557: Connected to ATmega256RFR2 Xplained Pro. 12:16:10:827: Available interfaces on ATmega256RFR2 Xplained Pro is SPI, TWI, GPIO, Code Profiling. 12:15:15: Polling started on	Image: state of the state o	00x00:1
Timestamp Global scrolling 00:17:15.0792032	New plot New band Add Horiz. Cursor Delete Axis	

🔀 すること: ・ DGI制御盤(DGI Control Panel)で開始(Start)を押してください。

データが図表(Graph)単位部で作図されます。光感知器の上に手を翳す時に下図のような何かを見ることができます。



日中と夜間の動作を切り替えるのに光感知器データを使うことができます。夜間動作切り替え器を有用にするには動作間切り替え時の閾値が重要です。図表単位部は作図データが或る範囲内の時に記すための帯域(Band)と呼ばれる有用な機能を含みます。これは動作切り替え閾値の選択を簡単化するのに使うことができます。

💦 すること: ・ このインターフェースの供給源を新規帯域(New band)の吸い込み口に引き摺ってください。



夜間動作切り替え器が実際に動いて正しい閾値で切り替えることを見るには、図表(Graph)単位部の文字列標識機能が有用です。この例では動作形態が切り替えられる時毎に文字列を送るのに目的対象基板のCDC USART インターフェースが使われます。構成設定例のこの部分に対する目的対象応用コートは「文字列標識追加」で見つけることができます。

すること : ・データ可視器(Data Visualizer)で構成設定(Configuration)タブの単位部(Modul es)項で外部接続(External Connection)下で見つかるシリアル ポート(Serial Port) 制御盤を開いてください。	© Configuration	Configuration Modules External Connection DGI Control Panel Serial Port P Graph Terminal P Protocols Utilities I/O Dashboard Logging	
---	-----------------	--	--

すること:・接続されたキットに対応する正しいCOMポートを選んでください。

・応用コートに従ってシリアルポート パラメータを設定してください。

・端末を開く(Open Terminal)任意選択がチェックされていないことを確実にしてください。

erial Port Contr	ol Panel		^ ×
EDBG Vir	tual CO	M Port (COM127)	Connect
Baud rate	Parity	Stop bits	Open Terminal <u>Autodetect protocol</u>
9600	None •	1 bit 🔻 🌔 💿	



文字列標識は頂上に四角形を持つ垂直線として現れます。四角形をダブルクリックすることにより、文字列の文が表示されます。ADC データ値が閾値を横切ってから文字列メッセージがホストコンピュータに達するまでに当然幾許かの遅延があります。加えて、データの時刻印 はホストコンピュータで追加され、2つのシリアルインターフェースは同期されません。これはADC値と比較して文字列標識の整列誤りに帰着しま す。DGIはXplained Pro上のEDBGの時刻印機能を含み、これは性能の犠牲に於いてDGI制御盤(DGI Control Panel)で許可すること ができますが、CDCは時刻印機能を含みません。



助言: この例では文字列標識に対して独立したシリアル インターフェースが使われました。利用可能なシリアル インターフェース数が制限される場合、Atmelデータ規約(ADP)を使うことによってADCデータと文字列標識データの両方を流すのに同じインターフェースを 使うことができます。より多くの情報については「Atmelデータ規約」をご覧ください。

ここまでで、データ可視器の図表単位部は光感知器によって生成されたデータを表示するのと夜間動作切り替え器が2つの動作形態間 を交互切り替えする時を表示するのに使われています。図表単位部は目的対象応用が走行している間にそれと情報交換するのにも 使うことができます。この例では水平カーソルを使うことによって夜間動作閾値を動的に調整することができます。 構成設定例のこの部分に対する目的対象応用コードは「水平カーソルを使うコード」で見つけることができます。

| すること:・最初に、夜間動作閾値が動的な時に使われないため図表から帯域(Band)を取り去ってください。

- ・図表に水平カーソルを追加するために水平カーソル追加(Add Horiz. Cursor)をクリックしてください。
 - 水平カーソル(Horizontal Cursor)の供給源をシリアルホート制御盤(Serial Port Controlo Panel)の吸い込み口に引き 摺ってください。

Graph0
Cursor 0.0.3
<u> </u>
Add axis No scrolling Automatically fit Y Scroll the mouse-wheel while pressing and holding the left shift key to zoom in the time axis.
New plot New String Image: String in the s
Plot 0 Plot V Input Sample rate 0 Set Delete Graph Plot 0.0.0
Show Cursors
0.0005 Una 0.000Hz 0.000Hz
String 2 ABC Input Delete String Plot 0.0.2
Horizontal Cursor Input Delete Value 100 Label Cursor 0.0.3 Delete
Serial Port Control Panel
EDBG Virtual COM Port (COM127)
Baud rate Parity Stop bits Open Terminal
9600 None - 1 bit - 600

水平カーソルを移動するにはそれを引き摺るか、または形態設定(Configuration)の値(Value)に新しい値を入力するかのどちらかを行っ てください。その値は値(Value)文字枠がフォーカスを失う、即ち、値を入力した後に何処か別の場所をクリックするまで更新されません。 カーソルが移動される毎にデータ可視器はカーソルが接続されているシリアルポートに新しい浮動値を送ります。

助言: それが未だ走っている間にもっと詳しく調べるために自動スクロール(Auto-scroll)と自動Y適応(Automatically fit Y)をOFF にしてください。

3.3. オシロスコーフ゜

オシロスコープ(Oscilloscape)単位部は実時間でデータを可視化します。オシロスコープは繰り返し信号や稀な事象を捕獲するための起動補助単位部が特徴です。オシロスコープはデータの流れの様々な特性を測定するためのカーソルシステムも持ちます。

3.3.1. オシロスコーフ 単位部

オシロスコープ単位部は実時間でデータを可視化します。これは同時に4つの異なるデータの流れを監視するための4チャネルを持ちます。各 チャネルのデータの流れは各々違う色と共に作図領域で図表として可視化されます。垂直位置と各チャネルの振幅は変更することができま す。繰り返し信号または稀な事象の捕獲について、オシロスコープは起動補助単位部を持ちます。オシロスコープはデータの流れの様々な特 性を測定するためのカーソルシステムも持ちます。

オシロスコープをどう構成設定するかの例については「オシロスコープ構成設定例」をご覧ください。



3.3.1.1. オシロスコーフ 制御盤

オシロスコープ制御盤(Oscilloscope control panel)はオシロスコープが構成設定されてシステムの残りに接続される場所です。この制御盤は以降の項で詳細に記述される5つの区分を持ちます。

図3-13. オシロスコープ制御盤		
Run Control Run Stop Horizontal Range Offset Sample rate 100 kHz	Trigger Edge J L Level 81.62162 Source I Auto	Cursors Show Cursor 1 Cursor 2
Vertical Channel 1 Channel 2 Range 300 200 300 Offset 10 10 0	Channel 3 Channel 4 Range Range 300 300 ✓ Offset 0ffset 10 10	4 4 4

3.3.1.1.1. 垂直(Vertical)制御部

垂直制御部は4つのオシロスコープ。チャネルの各々に対して1つで、4つの補助部を持ちます。チャネル制御部はチャネルの吸い込み口に供給 源が接続されるまで禁止されます。

3.3.1.1.1.1. オシロスコープ 接続

信号またはデータの流れはチャネルの吸い込みロ端点を通してオシロスコープに接続されます。外部供給源を吸い込みロに落としてください。接続すると、チャネル制御部の残りが許可されます。

3.3.1.1.1.2. チャネル振幅調整

作図領域でチャネルが表示される時に信号の高さはチャネル振幅(Range)設定によって決められます。

振幅は以下のように3つの異なる方法で調整することができます。

- ・文字枠に振幅値を入力してください。新しい値を有効にさせるには文字枠を選択解除してください。
- ・文字枠の右に置かれた上/下矢印をクリックしてください。
- ・文字枠上に置かれたマウス カーソルで、振幅設定を増加/減少するためにマウスの輪をスクロールしてください。

3.3.1.1.1.3. チャネルの表示と非表示

チャネルが使用中の時に範囲(Range)チェック枠を解消することによって作図を作図領域から消すことができます。作図を再び表示するにはそれをクリックしてください。

3.3.1.1.1.4. チャネル変位調整

作図領域でチャネルの垂直位置は変位(Offset)設定で調整することができます。変位値は作図領域の底から上へのチャネルの0点の距離です。

変位を調整するには以下のように4つの方法があります。

- ・文字枠に変位値を入力してください。新しい値を有効にさせるには文字枠を選択解除してください。
- ・文字枠の右に置かれた上/下矢印をクリックしてください。
- ・文字枠上に置かれたマウスカーソルで、変位設定を増/減するためにマウスの輪をスクロールしてください。
- ・作図領域でチャネルの0基準線が許可されている場合、それを新しい位置に引き摺ってください。

3.3.1.1.1.5. 0基準線の表示と非表示

0基準線は既定によってチャネルが供給元に接続することによって許可される時に表示されます。0基準線はチャネルの色と同じ色で作図内に表示される水平線です。これは左端に0の表題も持ちます。

0基準線は変位(Offset)チェック枠をクリック/解消することによって表示/非表示にすることができます。

3.3.1.1.1.6. チャネル名の独自化

オシロスコープ単位部が作業空間に追加されると、4つのチャネルはChannel 1~Channel 4と呼ばれます。 何の信号がそのチャネルに接続さ れているかを憶えるためにこの表題を変更することができます。

・表題の内側をクリックして新しい名前を入力してください。

3.3.1.1.2. 走行制御(Run Control)部

走行/停止(Run/Stop)と単独(Single)の釦は走行制御です。これらの釦は作図が更新されるか否かを制御します。

以下のように3つの動作形態があります。

- ・停止(走行/停止(Run/Stop)釦が赤色)
- ・単独 (単独(Single) 釦が黄色)
- ・走行(走行/停止(Run/Stop)卸が緑色)

3.3.1.1.2.1. 走行動作移行

停止または単独の動作形態が活性(赤または黄の点灯)の時に走行/停止(Run/Stop)釦をクリックすることによって走行動作へ移行してください。釦が緑に切り替わり、作図は起動設定に従って継続的に更新します。

3.3.1.1.2.2. 単独動作移行

停止または走行の動作形態が活性(赤または緑の点灯)の時に単独(Single)釦をクリックすることによって単独動作へ移行してください。 釦が黄に切り替わり、作図は起動されて一度だけ更新します。

3.3.1.1.2.3. 停止動作移行

単独または停止の動作形態が活性(黄または緑の点灯)の時に走行/停止(Run/Stop)釦をクリックすることによって停止動作へ移行してください。釦が赤に切り替わり、作図は停止します。

3.3.1.1.3. 起動(Trigger)制御部

オシロスコープ起動補助単位部は入力信号の望む部分だけを識別して自動追跡するのを手助けします。走行制御部によって設定され る動作形態に応じて起動は以下のようにすることができます。

- ・周期的な信号に対する自動追跡と継続的な作図更新
- ・信号が或る水準を超える時にだけ作図更新

3.3.1.1.3.1. 端起動

端起動機構は起動水準を横切る信号を探します。正端起動については信号が起動水準の下から起動水準の上へ行かなければなりません。



3.3.1.1.3.2. 端起動形式設定

起動機構は正端起動と負端起動の2つの起動動作形態を持ちます。

- ・正端起動動作に設定するには 🛛 釦をクリックしてください。釦は活性の時に強調表示されます。
- ・負端起動動作に設定するには 🔃 釦をクリックしてください。釦は活性の時に強調表示されます。

3.3.1.1.3.3. 起動水準設定

起動水準は以下のように4つの異なる方法で調整することができます。

- ・文字枠に起動水準値を入力してください。新しい値を有効にさせるにはキーボードのTab釦を押すか、または文字枠外側でマウスをク リックしてください。
- ・文字枠の右に置かれた上/下矢印をクリックしてください。
- ・文字枠上に位置したマウス カーソルで起動水準を増/減するためにマウスの輪をスクロールしてください。
- ・作図領域で起動水準線を新しい位置に引き摺ってください。

3.3.1.1.3.4. 起動元選択

オシロスコープ。起動補助単位部は起動条件を探す時にチャネル信号の1つを使います。

・起動元として使うために選ばれるチャネルに対応する色付けされた起動元(Source)卸をクリックしてください。

3.3.1.1.3.5. 起動動作設定

オシロスオープ単位部は起動と自由走行の両動作を支援します。

- ・起動動作を許可するには標準(Normal)をクリックしてください。作図は起動条件が満たされた時にだけ更新されます。
- ・自由走行動作を許可するには自動(Auto)をクリックしてください。作図は継続的に更新され、起動条件は無視されます。

3.3.1.1.4. 水平(Horizontal)制御部

オシロスオープは一定の速度で作図線を描きます。X軸が時間軸です。軸の表題は起動点に相対する時間を示します。正しく表示され る表題のために、オシロスコープは供給元の採取速度を知ることが必要です。

3.3.1.1.4.1. 採取速度設定

採取速度(Sample rate)文字枠で供給元の採取速度を入力してください。

注: オシロスコープに接続された全ての供給元は同じ採取速度を持たなければなりません。そうでない場合、作図の線は時間軸に同期 しません。

3.3.1.1.4.2. 水平分解能設定

水平分解能は時間軸の範囲(Range)、または作図でどの時間帯が見えるのかを決めます。以下のような5つの異なる方法で調整することができます。

- ・ マウス カーソルを作図領域の内側に置いてください。拡大や縮小のためにマウスの輪を使ってください。
- ・文字枠に分機能値を入力してください。新しい値を有効にさせるために文字枠を非選択にしてください。
- ・文字枠の右に置かれた上/下矢印をクリックしてください。
- ・文字枠上に位置したマウスカーソルで、分解能設定を増/減するためにマウスの輪をスクロールしてください。
- ・分解能を変更するために時間軸拡大/縮小取っ手を引き摺ってください。

3.3.1.1.4.3. 水平変位設定

水平変位(Offset)は作図領域の中央に相対する起動点の位置です。一般的に、変位は起動点のどちらかの側の作図を調査するために変更されます。以下のように変位を変更する5つの異なる方法があります。

- ・マウス カーソルを作図領域の内側に置いてください。起動線、0基準線、カーソル線のどれにも接触しないことを確実にしてください。その後に変位を変更するためにマウスをクリックして水平に引き摺ってください。
- ・ マウス カーソルを時間軸に置いてください。その後に変位を変更するためにマウスをクリックして水平に引き摺ってください。
- ・文字枠に変位値を入力してください。新しい値を有効にさせるために文字枠を非選択にしてください。
- ・文字枠の右に置かれた上/下矢印をクリックしてください。
- ・文字枠上に位置したマウスカーソルで、変位設定を増/減するためにマウスの輪をスクロールしてください。

3.3.1.2. カーソル

オシロスコープは作図を調査するのに使うことができる2つのカーソルを持ちます。カーソルはパルス幅、振幅、周波数、等々のような測定を簡単化します。

各カーソルは垂直の1つと水平の1つで2つの線として作図領域で表示されます。垂直カーソル線が移動されると、水平線は作図線のよう に従い、垂直と水平のカーソル線は同じ点で交差します。

カーソルの各々に対してどのチャネルが供給元かを設定することができます。

作図領域の下部はデータ行です。カーソルの各々に対してXとYの値を表示します。加えて、△X、△Y、1/△X、が計算されて表示されます。



3.3.1.2.1. カーソルの表示と非表示

オシロスコープ制御盤(Oscilloscope contorol panel)で作図領域のカーソルとカーソルデータを表示または非表示にするために表示(Show)釦を 交互切り替えしてください。

3.3.1.2.2. カーソル供給元チャネル選択

オシロスコープの制御盤のカーソル(Cursor)群でそのチャネル用の供給元として使うチャネルを選ぶためにカーソル1(Cursor 1)とカーソル2(Cursor 2) の引き落とし一覧枠でクリックしてください。使うために選んだチャネルに合致する色を選んでください。 データ線でカーソルのXとYの表題は選んだチャネルの色に合致する色に変わります。

3.3.1.2.3. カーソル移動

カーソルの垂直線(X軸)だけを移動することができます。水平線(Y軸)はそれに従います。

・マウス カーソルを垂直カーソル線上に置いてください。マウス カーソルはカーソルを左/右に変えます。カーソルをクリックしてそれの新しい位置へ引き摺ってください。

カーソル再配置後、データ線の計測値が更新されます。

3.3.1.2.4. カーソルを視野内へ移動

何かの拡大縮小や平行移動後、カーソルが見える区域の遥か外側になり得ます。それを視野に連れ戻すのは以下のように簡単です。 ・データ線のX1とY1の表題上で右クリックしてください。飛び出たメニューから視野内に持って来る(Bring into view)を選んでください。

3.3.2. オシロスコーフ。構成設定例

本節は光感知器を持つ夜間動作切り替え器を実装する目的対象応用で使われるようにオシロスコープ(Oscilloscope)単位部をどう構成 設定するかの例を与えます。この例がデータ供給元としてSPIインターフェースを利用するだけとは言え、この手続きは全てのデータ供給元に 対して同じです。この例で使った目的対象コート、とハート・ウェア構成設定の説明は「オシロスコープ コート・例」で見つけることができます。



い込み口に引き摺ってください。

Configuration	DGI Control Panel	^
Modules	ATmega256RFR2 Xplained Pro	Disconnect
External Connection Graph Graph Oscilloscope Power Terminal Protocols Utilities UD Bashboard	And Placeboordoors's Interfaces: SPI SPI Code Profiling Code Location Code Location	ect ADP Reset MCI
Logging Message 13:19:04.080: EDBG Control Panel added. 13:19:45.114: Connected to ATmega256RFR2 Xplained Pro. 13:19:45.151: Available interfaces on ATmega256RFR2 Xplained Pro is SPI, TWI, GPIO, Code Profiling.		- Trigger level
	OCCOO Run Control Run Stop Horizontal Range Offset Sample rate 100 KHz Vertical Channel 1 Pange 300 m Offset 300 m Offset 00 ffset 00 ffset	
Fimestamp 00:01:20.5850000		

オシロスコープ(Oscilloscope)単位部は今やI/O1 Xplained Pro上で卓上灯のONとOFFを切り替えるための光感知器から採取したデータを 分析するのに使うことができます。



- すること:・採取速度(Sample rate)を100kHzに設定してください。
 - ・下降端(Edge)での起動を許可して動作形態(Mode)を標準(Normal)に設定してください。
 - ・DGI制御盤(DGI Control Panel)で開始(Start)を押してください。
 - ・オシロスコープ(Oscilloscope)単位部で走行/停止(Run/Stop)釦を押してください。

オシロスコープの作図領域でマウスと共に起動水準を引き摺ることによる起動水準の何等かの調整と、水平(Horizontal)と垂直(Vertical)の範囲(Range)を調整することによる作図での拡大後、事象での 電灯切り替えは右図のようなものを見ることができます。



カーソル(Cursors)をONに切り替えることによってON状態で安定させるため電灯に対してかかる時間を測定することが可能です。この場合、約300msかかりました(作図領域の⊿X)。 作図上で更に拡大すると、光点滅の周波数を測定するのにカーソルを使うことが可能です。作図領域の1/⊿X領域は周波数が約100Hzであることを示し、これは電灯の50Hz電力と良く一致します(電力は秒毎に極性を100回切り替えます)。



3.4. 電力デバッグ

電力ディック(Power Debugging)単位部はDGI制御盤(DGI Control Panel)の電力(Power)インターフェースによって生成される(一般的に電力測定として参照される)電流と電圧の測定を表示します。目的対象MCUでのコート、実行と目的対象応用の電力消費の相関のために、電力測定は同じ図表で汎用入出力(GPIO)とコート、特性分析(Code Profiling)のような様々な他のインターフェースと組み合わせることができます。

3.4.1. 電力ディック単位部

電力デバッグ(Power Debugging)単位部は接続されたキットの電流消費を表示します。基本的な電流測定を開始するには「基本的な電 流測定」節をご覧ください。カーソルをどう使うかの例については「カーソルを使う電力分析」をご覧ください。コート・実行と電流消費をどう互 いに関連付けるかの例については「コート・相互関係」をご覧ください。



重要: 電力(Power)単位部は電力(Power)インターフェースとだけ使うことができます。

3.4.1.1. 図表の尺度調整とスクロール



助言: それが未だ走っている間にもっと詳しく調べるために自動スクロール(Auto-scroll)と自動Y適応(Automatically fit Y)をOFF にしてください。

求められるように作図を移動するために尺度調整(Scale)と変位(Offset)を使ってください。この点に関して以下のようにマウスのスクロール輪が有用です。

- ・時間(X)軸で拡大/縮小するには作図上でShift-スクロールをしてください。
- ・Y軸で拡大/縮小するには作図上でCtrl-スクロールをしてください。
- ・時間(X)軸を水平移動してY軸(変位)を移動するには図表を引き摺ってください。
- ・Y軸で各々のチャネル(変位)を移動するには左または右の目盛りを引き摺ってください。
- ・チャネルを拡大/縮小するには各々の軸の目盛り上でCtrl-スクロールをしてください。
- ・拡大縮小する領域を選択するには右クリックして引き摺ってください。

3.4.1.2. 電力デバッグ単位部制御盤

電力ディッグ(Power Debugging)単位部の制御盤(Control Panel)は単位部の右上角に置かれます。

Power Analysis			×.
620nA -	այ, որ հանդիկայուրվել ու, վել ու, լե. այս այս գեղել է, կառու անգու է, ու տել ու, ու ու	- 6mA - 4.7mA	Channel A Channel B Channel B Code Location Code Location
440nA -	lan a na ana ana ana ana a ana a ana ana	-34mA	Cursors Auto-scroll Automatically fit Y Show zero
260nA -		- 2.1mA	
80nA - E		- 0.8mA	
-100nA - CH A Cur		0.5mA ස ප්	
-280nA -		1,8mA	
-460nA -		3,1mA	
-640nA -	in a second state of the second s	4,4mA	
-100064 -		5,7mA	
Ch A Window Ave 	1021 1022 1023 1026 1027 1028 1020 rgg balant		

▶ 通知:制御盤の全ての構成設定任意選択が全てのツールで利用可能な訳ではありません。例えば、Power DebuggerはAチャネルとBチャネルの両方を持ちます。全てのツールで全ての任意選択が見えますが、そのツールがそれらを支援しない限り無効です。

自動スクロール(Auto-scroll)任意選択はX軸方向(時間軸)でのスクロールを制御します。拡大して図表を詳細に調査するにはこの任意選択を禁止してください。

自動Y適応(Automatically fit Y)任意選択はデータ可視器(Data Visualizer)が図表内容に従ってY軸の範囲を自動的に調整するか否かを制御します。この任意選択が許可される場合、Y軸のどの手動調整も無効にされます。

0表示(Show zero)任意選択は自動Y適応(Automatically fit Y)が許可される時にY軸の0点が常に見えるかどうかを制御します。

3.4.1.2.1. チャネル構成設定

各電力測定チャネルに対しては電力分析(Power Analysis)単位部の制御盤(Control Panel)にチャネル構成設定 (Channel configuration)部があります。

チャネル(Chanel x)部は使用者に電力分析(Power Analysis)単位部で電流(Current)と電圧(Voltage)の図表の許可/禁止を許します。



通知: DGI制御盤(DGI Control Panel)(電力インターフェース参照)の電力構成設定(Power Configuration)でBチャネル許可(Enable B Channel)任意選択が選ばれない場合、例えツールがBチャネルを持っていても、Bチャネルは利用不能です。しかしBチャネル 構成設定は制御盤(Control Panel)で未だ目に見えます。

範囲(Range)設定は電流測定チャネルに対してデータの測定範囲を許します。電流測定チャネルによって支援される電流値の全範囲を網 羅するため、殆どのツールは各チャネルに対して2つ以上のハードウェア構成設定を持ちます。チャネルの範囲数は接続したツールで変わりま す。ハードウェア構成設定間の切り替えは測定された瞬間電流に基づいて自動的に行われます。



通知: 範囲図表はDGI制御盤(DGI Control Panel)の電力構成設定(Power Configuration)で範囲元許可(Enable Range Sour ce)任意選択が選ばれる場合にだけ目で見ることができます。

動作形態(Mode)任意選択は現在のツールに対してこれが許可された場合にデータの表示に使われる各種の平均化算法を許します。

3.4.1.2.2. コート 位置

コート[']位置(Code Location)部は任意選択なしで単に供給元接続を含みます。電力 分析(Power Analysis)図表でコート[']位置を許可するにはDGI制御盤(DGI Control Pan el)のコート[']特性分析(Code Profiling)インターフェースが許可されて、DGI制御盤のコート[']特 性分析構成設定(Code Profiling Configuration)でコート[']位置許可(Enable Code Loca tion)が許可されなければなりません。



3.4.1.2.3. GPIO

汎用入出力(GPIO)供給元は電力分析(Power Analysis)単位部の制御盤(Control Panel)の汎用入出力(GP IO)部でONまたはOFFに切り替えることができます。



電力分析(Power Analysis)単位部に対して利用可能なGPIOデータのために、DGI制御盤 (DGI Control Panel)で汎用入出力(GPIO)インターフェースが許可されなければなりません。



3.4.1.2.4. カーソル

電力分析(Power Analysis)単位部制御盤(Control Panel)のカーソル(Cursors)部は許可(Enabled)枠をチェック することによって図表で2つの垂直カーソルを許可することを使用者に許します。

カーソルはX軸に沿ってそれらを引き摺るためにマウス指示子を使うことによって移動することができ、また、それらは中央(Center)釦を押すことによって中央にすることができます。



カーソルが許可されると、図表のカーソル間の部分を様々な測定に使うことができます。測定は図表の下のカー ソル(Cursors)部で表示されます。



どの測定が表示されるべきかは電力分析(Power Analysis)単位部制御盤(Control Panel)でカーソル(Cursors)部の測定(Measurements) 補助部で選ぶことができます。

3.4.2. 基本的な電流測定

DGI Control Panel	^ X
Power Debugger Data Gateway	▼ Connect Start
	ADP Logging Autodetect protocols Reset MCU
Interfaces:	
	τ/τ
DGI Control Panel	A X
JS020000088	- Start
	ADP Logging Autodetect protocols Reset MCU
	🗐 GPIO 🔯 🗐 Power 🔯
€ ~ 0 € ~ 0	0 🐠 💿 A Current 🍘 Code Location
	1 Care A Voltage
	3 Come S B Current Come
電力(Power)インターフェースを許可して関注	連チャネルを監視するようにそれの設定を変更してください。
DGI Control Panel	^ X
Power Debugger Data Gateway	Disconnect
15020000088	Start
15020000088	ADP Logging Autodetect protocols Reset MCU
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO Power Code Profiling
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO Power Code Profiling A Current Code Location Code Location
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO Power Code Profiling 0 A Current Code Location A Voltage Power
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO Power Code Profiling A Current Code Location Code Location B Current B Voltage B Voltage
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO Power Code Profiling Code Location A Voltage B Current B Voltage B Voltage
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO GPIO COde Profiling Code Location COde Locatio
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO GPIO Code Profiling Code Location A Voltage Code Location B Voltage Voltage Code Location W
Interfaces:	Start ADP Logging Autodetect protocols Reset MCU GPIO GPIO A Current A Voltage Code Location B Voltage

すること: データ可視器(Data Visualizer)作業を開始してください。

3.4.2.1. 2つのチャネルの測定

2つのチャネルを持つハートウェア使用時、データ可視器(Data Visualizer)は(電力構成設定(Power Configuration)で禁止されない限り)両方 を同じ図表で表示します。

単位部の右の制御盤(Control panel)で使用者は利用可能な時に電流と電圧の作図だけでなく、範囲情報も 表示または非表示にすることができます。

既定により、両チャネルは電力分析(Power Analysis)図表で表示されますが、各作図はそれらを最適なように独立するために上下に移動することができます。



3.4.2.2. 図表の尺度調整とスクロール

助言: それが未だ走っている間にもっと詳しく調べるために自動スクロール(Auto-scroll)と自動Y適応(Automatically fit Y)をOFF にしてください。

求められるように作図を移動するには尺度調整(Scale)と変位(Offset)を使ってください。この点に関して以下のようにマウスのスクロール輪が有用です。

- ・時間(X)軸で拡大/縮小するには作図上でShift-スクロールをしてください。
- ・Y軸で拡大/縮小するには作図上でCtrl-スクロールをしてください。
- ・時間(X)軸を水平移動してY軸(変位)を移動するには図表を引き摺ってください。
- ・Y軸で各々のチャネル(変位)を移動するには左または右の目盛りを引き摺ってください。
- ・チャネルを拡大/縮小するには各々の軸の目盛り上でCtrl-スクロールをしてください。
- ・拡大縮小する領域を選択するには右クリックして引き摺ってください。

3.4.2.3. 電流平均化

電力分析(Power Analysis)単位部はチャネル毎に平均化された2つの値を表示します。1つは瞬間電流値を示し、一方で他方は図表視野内で見ることができる採取(試料)の平均を示します。

3.4.2.4. 電力測定校正

電流測定ハードウェアで完全な測定精度を達成するには測定を走らせる前に校正されるべきです。校正手順はDGI制御盤(DGI Contro l Panel)の電力(Power)インターフェースで電力構成設定(Power Configuration)を通して開始されます。

図3-18. 電力構成設定ウィンドウから電力測定校正の	·起動
DGI Control Panel	^ X.
Power Debugger Data Gateway	T Disconnect Start
	ADP Logging Autodetect protocols Reset MCU
Interfaces:	
🔲 SPI 🔯 🔲 USART 🙀 🗐 TWI 🙀 🗐 GPIO 🙀	🔽 Power 🙀 🔲 Code Profiling 🙀
	A Current Code Location
S Power Configuration	A Voltage 🥌
Enable B Channel	B Current
Trigger calibration	B Voltage
Enable Range Source	
Lock ChA to High Range 🔲	
Enable Voltage Output	
Voltage Output	v
Averaging No •	
OK Cancel	

校正手順を開始するには校正を起動(Trigger calibration)を選んでOKを押してください。その後、ツールに対する校正手順を完了する ため、指示に従ってください。

3.4.3. カーソルを使う電力分析

もっと詳しく電流を分析するためには電力分析(Power Analysis)のカーソル機能が有用です。

🎲 すること: ・ 電力分析(Power Analysis)単位部の右上隅で制御盤(Control Panel)を開いてください。

- ・カーソル(Cursors)項を展開してください。
- ・カーソルをONにするために許可(Enabled)枠をクリックしてください。



記憶:電流測定が未だ走っている場合、カーソルを許可する前に自動スクロール(Auto-scroll)を禁止することを確 実してください。そうでなければ図表視野はカーソルから遠く離れて急激にスクロールするでしょう。



上の例はONとOFFを定期的に交互切り替えするLEDを持つ目的対象基板の電流消費を示します。

単位部下部のカーソルデータはLEDがOFFの時の電流消費が約354µA、同時にLEDがONの時の電流消費が約6.5mAであることを示します。LED交互切り替えの1周回中の平均電流消費は約580µAです。電流測定チャネルは電圧も監視しているため、電力消費を直接測定することができます。電力分析(Power Analysis)単位部の制御盤(Control Panel)でカーソル(Cursors)部の測定(Measurements)部分で対応する任意選択を設定することによってこれを許可してください。

3.4.4. コート の相互関係

電流消費を最適化するには、電流測定が応用のコート、実行と互いに関連付けされなければなりません。データ可視器(Data Visualizer) はGPIO計装またはプログラム カウンタ採取読み出しの使用によってコート、相関を許します。これら両方法で重要なのは同じ時間基準を持 つコート、実行に関連する事象を電流消費として同じ図表で示すことができることです。

3.4.4.1. GPIO計装

応用コートに簡単なGPIO交互切り替えを挿入することにより、使用者は測定した電流とコート、実行間での共通参照点を生成することができます。データ可視器(Data Visualizer)は電流測定として同じ図表でGPIO事象を表示する能力があります。

GPIO計装の使用を実演するのに大容量記憶応用が使われます。

SAM L21 Xplained Pro基板の目的対象USBとデバックUSBの両方がホストコンピュータに接続されます。ATSAML21目的対象デバイスは SAM L21 Xplained Pro用ASFからのUSB装置MSC例(Atmel Studioでファイル(File)⇒新規(New)⇒例プロジェクト(Example Project)を選ん で"MSC"で検索)を走行しています。キット上の電流測定ジャンパはMCU電流を測定、I/O電流を迂回に設定されます。

大容量記憶装置のフォーマット走行後の電流図表は次のとおりです。



ディスクフォーマット操作は読みと書きの両操作から成りますが、 今現在その時に何が起こっているかを知らせるのは困難で す。読みと書きの操作を分け得るため、読み込み操作の開 始でGPIO0(ATSAML21のPB01)をHighに設定し、読み込み 操作の終わりでLowに設定するように応用コートが変更されま す。GPIO1(ATSAML21のPA16)は同様に書き込み操作に対 して交互切り替えされます。汎用入出力(GPIO)と電力(Powe r)の両インターフェースは右で示されるようにデータ可視器(Data Vi sualizer)のDGI制御盤(DGI Control Panel)で許可されなけれ ばなりません。



右で示されるように電力分析(Power Analysis)の制御盤(Control Panel)でGPIO2とGPIO3を禁止してください。

許可されたGPIO信号とで、使用者は読みと書きの操作の電流消費を区別することができます。黄色の信号は読み込み操作を合図するGPIO0で緑色の信号は書き込み操作を示すGPIO1です。



Channel A Current Voltage Range Mode HiRes Current Voltage Mode HiRes GPIO GPIO 0 1 2 3

3.4.4.2. フ[°]ロク^{*}ラム カウンタ ホ[°]ーリンク^{*}

目的対象からプログラムカウンタ(PC)が読み出される時毎に、現在実行されつつあるコート位置の正確なアトレス情報を得ます。データ可視器(Data Visualizer)は同じ図表で電流測定と共にPC値を表示することができます。これは電流消費図表の様々な採取点で目的対象 CPUによって何が実行されつつあるのかを見ることを使用者に許します。殆どの場合で目的対象が命令を実行するそばからPC値を 読み出すことが不可能なため、採取されたPC値はコート、実行の一部を示すだけです。採取された値は何れかの時点でどのコート、区域 が実行されつつあるかを示すのに未だ有用です。

大容量記憶クラス例を走行しているSAM L21 Xplained Pro基板がPCポーリンクを実演するのに使われます。

SAM L21 Xplained Pro基板の目的対象USBとデュックUSBの両コネクタはホストコンピュータに接続されます。ATSAML21目的対象デュイス はASFからのSAM L21 Xplained Pro用USB装置MSC例を走行しています(Atmel Studioでファイル(File)⇒新規(New)⇒例プロジェクト(Exa mple Project)を選んで"MSC"を検索してださい)。キット上の電流測定ジャンハプはMCU電流測定と入出力電流迂回に設定してください。 大容量記憶装置のフォーマット走行後の電流図表は次のとおりです。



ディスクフォーマット操作は読みと書きの両操作から成りますが、電流図表からその時に何が起こっているかを見るのは困難です。電流図 表に於いて様々な点で目的対象で何が起きているかのより多くの情報を得るには、プログラム カウンタ採取機能が有用です。 電流測定データと共にプログラム カウンタ採取(試料)を見るには電力(Power)とコート、特性分析(Code Profiling)の両インターフェースが許可され なければなりません。

すること: ・ DGI制御盤(DGI Control Panel)で電力(Power)インターフェースとコート、特性分析(Code profilin g)インターフェースの両方を許可してください。





フォーマット操作の間に許可されたプログラム カウンタ採取を持つ代表的な電流図表が下で示されます。

・コード位置許可(Enable Code Location)を選んでください。

ダイアログを開いてください。



図表で作図された黄色の点がポーリングされたプログラム カウンタ値を表します。Y軸でのこれらの位置は目的対象デベイスのコード空間に於 けるそれらの位置の視覚的表現です。採取(試料)の相対的な群化は違う関数の実行を示します。様式はこの技法を使って容易に見 ることができます。採取(試料)の1つの上でマウスを浮かせると、図表の下のコード位置詳細(Code location details)でその採取の位置だ けでなく、その点の電流採取の値も示します。



採取(試料)の1つでダブル クリックすると、エデ゙ィタが開いてコードの対応する行を強調表示します。

udi_msc.c + X_saml21_yplained_pro.h board_init.c ui.c usb_msc_example	main.c 🛎 X 👻
<pre></pre>	€G0 ±
udi_msc_csw.dCSMDataResidue -= block_size;	
	•
DGI Control Panel	~ ×
Prover Analysis 16.8mA 12.6mA 9 4.4mA 0 4.4mA 0 4.4mA 0 551.180 36.51.182 36.51.184 36.51.186 36.51.188 36.51.180 36.51.190 36.51.190 36.51.192 Code location details (Hover to resolve) Instant Usbmsc_example/src/ASF/common/services/usb/class/msc/device/udi_msc.c, Line 1113 (© 0x00008B0)	 X Channel A Channel B Code Location GPIO Cursors Auto-scroll Automatically fit Y Show zero
udimsc trans_block 15,53mA	

強調表示された採取(試料)はudi_msc_trans_blockと呼ばれる関数に位置しています。この関数はデータをRAMからUSBへ転送します。 図表から、目印での電流尖頭波形は、この尖頭波形中に全てのプログラム カウンタ採取(試料)が同じ位置からのため、この関数の実行 によって生成されたのを見ることができます。

3.5. 独自計器盤

計器盤(Dashboard)単位部は独自設定可能な図画的使用者インターフェース(GUI:Graphical User Interface)盤です。これは制御と目的 対象応用からのパラメータ表示に使うことができます。

3.5.1. 計器盤単位部

計器盤(Dashboard)単位部は独自設定可能な図画的使用者インターフェース(GUI:Graphical User Interface)盤です。これは制御と目的 対象応用ファームウェアからのハ[®]ラメータ表示に使うことができます。要素(Elements)(釦、表題、摺動子など)はGUIを形成する計器盤領域 に配置されます。各要素は値を送ったり受けたりするためにそれと関連する端点を持つことができます。例えば、摺動子は数値として 摺動子位置を出力する供給元を持ちます。端点は端点表示(Show Endpoints)任意選択が選ばれる時に表示されます。

計器盤をどう構成設定するかの例については「計器盤構成設定例」をご覧ください。

図3-19. 計器盤			
Dashboard I/O	Test	Dashboard	* ×
1	Score	Time	
	58	Position	
2 Edit Show	Endpoints		4
1. 計器盤領域	2. 編集(Edit)チュ	ック枠 3. 端点表示(Show Endpoints)チェック枠	4. 高さ調整タブ

3.5.1.1. 編集操作盤

(編集(Edit)チェック枠選択によって)編集動作へ行くと、 編集操作盤が見えるようになります。ここで使用者は 計器盤を独自設定することができます。 図3-20 編集般

🖉 Edit 🔲 Show Gridlines 🔲 Show Endpoints 📃 🛛 Load Save
Configuration

3.5.1.1.1. 計器盤での要素配置

既定により、計器盤領域は空です。要素は下の手順に従うこ とによって計器盤に配置することができます。

- ・編集(Edit)チェック枠をクリックしてください。
- ・計器盤領域の右上隅で要素(Elements)操作盤を開いてください。
- ・要素をクリックして保持してください。
- ・マウスを計器盤領域上に引き摺ってください。
- ・望む位置の計器盤領域で要素を落としてください。



3.5.1.1.2. 計器盤要素構成設定

全ての計器盤要素は編集動作の時に構成設定することができます。 パラメータは要素の形式に応じて変わりますが、それらを変更するための手順は同じです。

- ・編集(Edit)チェック枠をクリックしてください。
- ・それをクリックすることによって構成設定する要素を選択してください。構成設定(Configuration)ウィントウが選んだ要素に対して構成設定可能なパラメータを一覧にします。
- ・パラメータを望む値に変更してください。
- ・設定(Set)釦をクリックしてください。



3.5.1.1.3. 要素移動

位置と大きさに関係する全てのパラメータが要素構成設定で利用可能です。要素は編集動作に於いて計器盤領域でそれらをあちこち に引き摺ることによって移動することもできます。大きさ変更は要素を選択した後に角の黒いつまみを引き摺ることによって行うことが できます。

3.5.1.1.4. 要素削除

望まない要素を削除するには単純に左クリックすることによって要素を選択してその後にそれを削除するために右クリックしてください。

重要:この処置は恒久的で削除後に全ての構成設定が失われます。

3.5.1.1.5. 読み込みと保存

計器盤は編集動作で保存(Save)卸をクリックすることによって保存することができます。計器盤背景色に加えて、全ての要素と構成設定パラメータが保存されます。

計器盤を読み込むには読み込み(Load)釦をクリックして有効な計器盤保存ファイルを検索してください。

保存されたファイルは又草ファイルですかとのファイル拡張于も持つことができ、中括弧'{}'で囲まれ、セミコロン';'に	Dashboard I/O	×
よって分離された各計器盤要素に対する全ての構成設 定パラメータを含みます。各行は1つの構成設定パラメータ に対応し、各パラメータの形式はコンマ','によって分離され た10進バイ値の一覧です。各構成設定パラメータは最下 位パイト先行によって与えられます。構成設定パラメータの 順番は計器盤用の編集(Edit)任意選択が選ばれた時 の構成設定ウィントウでの構成設定パラメータの順番と同じ です。注釈は2重スラッシュ'//'によって記され、2重スラッ シュに出会うと解析部によって行の残りは無視されます。 保存された計器盤構成設定の簡単な例が右で与えら	TEST DASHBOARD Button	
れます。もっと複雑な例は「自動構成設定例」で見つけ ることができます。	Edit 🔲 Show Endpoints	

この計器盤用に保存された構成設定のファイル内容が下で与えられます。

0,
'¥0',
0, 255, 255, 255,
158, 0,
};
0, // Dashboard ID (計器盤ID)
0, // Element ID (要素ID)
DB_TYPE_LABEL, // Element Type (要素形式)
0, // Z-Index (GUI stack order) (Z指標(GUI階層順))
61, 0, // X-coordinate (X座標)
46, 0, // Y-coordinate (Y座標)
122, 0, // Width (幅)
17, 0, // Height (高さ)
12, // Font Size (フォントの大きさ)
1,
0, // Horizontal Alignment (水平整列)
0, // Vertical Alignment (垂直整列)
0, 255, 255, 255, // Background Color (背景色)
255, 0, 0, 0, // Foreground Color (前景色)
'T', 'E', 'S', 'T', ' ', 'D', 'A', 'S', 'H', 'B', 'O', 'A', 'R', 'D', '¥O', // Text (文章)
]};
0, // Dashboard ID (計器盤ID)
1, // Element ID (要素ID)
DB_TYPE_BUTTON, // Element Type (要素形式)

```
0, // Z-Index (GUI stack order) (Z指標(GUI階層順))
61, 0, // X-coordinate (X座標)
70, 0, // Y-coordinate (Y座標)
75, 0, // Width (幅)
25, 0, // Height (高さ)
12, // Font Size (フォントの大きさ)
'B', 'u', 't', 't', 'o', 'n', '¥0', // Text (文章)
0,
};
```

ファイル内の最初の要素は計器盤それ自身です。最初の行は計器盤ID (0)を定義します。その後に計器盤の表題(空文字列)、計器盤 の背景色(アルファ=\$00、赤=\$FF、緑=\$FF、青=\$FF)、計器盤の高さ(2バイト値、LSB先行、152,0→152→\$0098ピクセル)が続きます。

後続する要素は表題(Label)要素と釦(Button)要素です。

文字列がヌル終端(¥0)されることに注意してください。

通常、チェック枠は寄せ集められ、チェック枠の状態を示すためにチェック枠毎に1ビットだけが使われます。例えば、表題要素に対して編集 (Edit)ウィンドウには1つは太字体、1つは斜体と名付けられたお互いに後続する2つのチェック枠があります。これらは太字体チェック枠の状態を与えるビット0と斜体チェック枠の状態を与えるビット1を持つ1つの構成設定バイトに結合されます。上の例に於いて、これはフォントの大きさ(Font size)と水平整列(Horizontal alignment)間の1です。この1の値は表題の文字形式が斜体ではなく太字体であるべきことを示します。

引き落とし枠(Drop-down box)は選んだ任意選択に対応する番号を持つ単一ハイト値として与えられます。一覧で最上部の任意選択 が0の構成設定値に対応します。例えば、表題要素について水平整列(Horizontal alignment)は左(Left)=0、中央(Center)=1、右(Righ t)=2のどれかにすることができます。

3.5.1.1.6. 背景色設定

読み込み(Load)傍らの四角が背景色(Background color)選択部です。この選択をクリックすると、色選択部(Color selector)ダイアログを持ち出します。望む色を選ぶのに摺動子を使い、その後にOKを押してください。

3.5.1.2. 要素型式

様々な計器盤要素型がこの項で提示されます。全ての要素型はいくつかの共通パラメータを持ちます。これらは下表で一覧にされます。以降の項は各要素型に特有のパラメータだけを一覧にします。

パラメータ	型	使い方	
Z指標 (Z-index)	数值	順序指標、0は後ろに対して最も遠くに要素を配置します。	
左 (Left)	数値	水平配置、単位はピクセル	
上 (Top)	数值	垂直配置、単位はピクセル	
幅 (Width)	数值	ピクセルでの要素の幅	
高さ (Height)	数值	ピクセルでの要素の高さ	

表3-1. 共通要素パラメータ

3.5.1.2.1. 表題

表題(Label)要素は文章文字列を表示します。



3.5.1.2.1.1. 端点

表題(Label)要素は全ての形式の供給元を受け入れる吸い込み口端点を持ちます。表題(Label)に送られるどのデータも文字列に変換されて文章として表示されます。

3.5.1.2.1.2. 構成設定

表3-2. 表題特有パラメータ

パ゚ラメータ	型	使い方
フォントの大きさ (Font Size)	数値	フォントの大きさ調整
太字体 (Bold)	チェック枠	フォント形式を太字体に設定
斜体 (Italic)	チェック枠	フォント形式を斜体に設定
水平整列 (Horizontal Alignment)	引き落とし枠	指定した幅内で文章の整列を選択
垂直整列(Vertical Alignment)	引き落とし枠	指定した高さ内で文章の整列を選択
背景色(Background Color)	色	表題の背景色を設定
前景色(Foreground Color)	色	文章の色を設定
文章 (Text)	文字列	表題文章を設定

3.5.1.2.2. 数值入力

数値入力(Numeric Input)要素は計器盤に対して数値の入力を許します。



3.5.1.2.2.1. 端点

数値入力(Numeric Input)要素はint32型の供給源端点を持ちます。数値入力が変更される毎に値を持つパケットが送られます。

3.5.1.2.2.2. 構成設定

表3-3. 数値入力特有パラメータ

パ゚ラメータ	型	使い方
最小 (Minimum)	数值	入力の最小値
最大 (Maximum)	数值	入力の最大値
值 (Value)	数值	初期值

3.5.1.2.3. 釦

釦(Button)要素はそれが押される毎に事象を送ります。釦は標準押し釦または交互釦のどちらかとして構成設定することができます。釦はそれの機能を示すために静的な文章を持つことができます。交互釦として構成設定されると、文章は釦の状態に応じてONまたはOFFによって置き換えられます。何か他の物によってON/OFF文字を置き換えるには、ON状態文章である文章の最初の部分とOFF状態文章の2つ目の部分を持つ'/'で区切られた文章として、文章(Text)パラメータが与えられなければなりません。

図3−24. 釦
Button

3.5.1.2.3.1. 端点

釦(Button)要素はuint8型の供給源端点を持ちます。釦が押される毎にパケットが送られます。パケットの値は標準釦に対しては常に0で、OFF状態の交互釦に対しては0、ON状態の交互釦に対しては1です。

3.5.1.2.3.2. 構成設定

表3-4. 釦特有パラメータ

パ゚ラメータ	型	使い方
フォントの大きさ (Font Size)	数值	卸表札のフォントの大きさを設定
文章 (Text)	文字列	卸表札文章を設定。卸が交互釦として構成設定される場合、文章は'/'によって区切られるべきです。文章の最初の部分がON状態の文章、一方で2つ目の部分がOFF状態の文章です。
交互釦 (Toggle Button)	チェック枠	釦をON/OFF交互切り替え器として構成設定

3.5.1.2.4. ラジオ群

ラジオ群(Radio Group)要素は同時に1つの任意選択だけを選ぶことができるラジオ釦の群です。初めは 最初の任意選択が選ばれます。

Option 1 Option 2

3.5.1.2.4.1. 端点

ラジオ群(Radio Group)要素はuint16型の供給源端点を持ちます。要素の状態が変更される毎に現在有効な任意選択の指標と共に メッセージが送られます。

3.5.1.2.4.2. 構成設定

表3-5. ラジオ群特有パラメータ

パラメータ	型	使い方
フォントの大きさ (Font Size)	数値	卸文章のフォントの大きさ
ラジオ釦数 (Number of Radio Buttons)	数値	群内の釦数
方向 (Orientation)	数值	0=水平、1=垂直
文章 (Text)	文字列	指標0の釦用文章で始まり、各釦用文章を持つ'/'で区切られた文章

3.5.1.2.5. チェック枠

チェック枠(Check Box)要素はそれの状態が変更される毎に事象を送ります。

図3-26. チェック枠
Option

3.5.1.2.5.1. 端点

チェック枠(Check Box)要素はuint8型の供給源端点を持ちます。要素の状態が変更される毎にメッセージが送られます。枠がチェックされると1が送られ、未チェックの時に0が送られます。

3.5.1.2.5.2. 構成設定

表3-6. チェック枠特有バラメータ			
パ゚ラメータ	型	使い方	
フォントの大きさ (Font Size)	数值	文章のフォントの大きさ	
文章 (Text)	文字列	入力の最大値	
值 (Value)	数值	チェック枠表札文章を設定	

3.5.1.2.6. 摺動子

摺動子(Slider)要素は移動可能な目印を持つ直線状の棒です。目印は摺動子の値を調整するために移動することができます。

図3-27. 摺動子	
	,

3.5.1.2.6.1. 端点

摺動子(Slider)要素はdouble型の供給源端点を持ちます。 摺動子値が変更されると、値を持つパケットが送られます。

3.5.1.2.6.2. 構成設定

表3-7. 摺動子特有パラメータ				
パラメータ	型	使い方		
最小 (Minimum)	数值	摺動子の最小値を設定		
最大 (Maximum)	数値	摺動子の最大値を設定		
值 (Value)	数值	摺動子の値を設定		

3.5.1.2.7. 信号

信号(Signal)要素は単純な色に基づくON/OFF指示器です。



3.5.1.2.7.1. 端点

信号(Signal)要素は全てのデータ型を受け入れる吸い込みロ端点を持ちますが、文字列と多次元値を無視します。信号の色はブール 評価によって決められ、やって来る値が数値でそれが0よりも大きければ真(true)です。

3.5.1.2.7.2. 構成設定

表3-8.信号符有ハフメータ		
N [°] ラメータ	型	使い方
ON色 (Color ON)	色	信号がONの時に使われる色を選択
OFF色 (Color OFF)	色	信号がOFFの時に使われる色を選択

© 2017 Microchip Technology Inc.

3.5.1.2.8. 進捗バー

進捗バー(Progress Bar)要素は最小と最大の値間で値の位置を示す直線状の棒です。

図3-29. 進捗バー	

3.5.1.2.8.1. 端点

進捗ハー(Progress Bar)要素は数値データ型を受け入れる吸い込み口端点を持ちます。 値が受け取られると、最小と最大に応じて進 捗ハーの色付けされた領域の量を更新します。

3.5.1.2.8.2. 構成設定

表3-9. 進捗バー特有パラメータ 型 パラメータ 使い方 進捗バーの最小値を設定 最小 (Minimum) 数値 最大 (Maximum) 数值 進捗バーの最大値を設定 値 (Value) 数値 進捗バーの値を設定 色 (Color) 色 進捗バーの色を設定

3.5.1.2.9. セグメント表示器

セグメント表示器(Segment Display)要素は16セグメントLED表示器を擬態します。



3.5.1.2.9.1. 端点

セグメント表示器(Segment Display)要素は全数値データ型を受け入れる吸い込み口端点を持ちます。受け取った値が表示されます。

3.5.1.2.9.2. 構成設定

表3-10. セグメント表示器特有パラメータ

N [°] ラメータ	型	使い方
桁数 (Segment Count)		表示する16セグメントの桁数
基数 (Numeric Base)	数値	数を表示するのに使われる基数を設定
セグメント色 (Segment Color)	色	セグメント表示の色を設定

3.5.1.2.10. 図表

図表(Graph)要素はやって来るデータの流れを2次元図表で作図します。図 表はマウスの作用によって拡大縮小とスクロールを受け入れるか、またはどのマ ウスの作用も無視して静止にすべきかを構成設定することができます。



3.5.1.2.10.1. 端点

図表(Graph)要素は各作図に対して1つの吸い込み口端点を 持ちます。端点は全ての数値データ型を受け入れます。

図表(Graph)内の各作図は図表(Graph)要素の下部で作図に 対応する凡例(Legend)をクリックすることによって動的に表示ま たは非表示にすることができます。非表示にされた作図は凡 例で作図自身と同じ色を持つ見える作図と比べて灰色の凡 例を持ちます。

右の図表でSPI出力の作図は可視である一方でTWI出力の作図は隠されています。

図3-32. 可視のSPI出力作図と非表示のTWI出力作図を持つ図表 Graph 100 100 1386,92 1386,92 Auto-scroll M Automatically fit Y Legend SPI Output TWI Output

3.5.1.2.10.2.	構成設定
---------------	------

表3-11. 図表特有パラメータ		
パ゚ラメータ	型	使い方
表題色(Title Color)	色	表題文章の色を設定
背景色 (Background Color)	色	図表要素全体背景の色を設定
図表背景色 (Graph background Color)	色	図表作図領域背景の色を設定
表題 (Title)	文字列	図表の表題
作図数 (Number of plots)	数値	図表で表示する作図数。各作図は自身の吸い込み口端点を持ちます。
X最小 (X Minimum)	数値	X軸の最小値
X最大 (X Maximum)	数値	X軸の最大値
Y最小 (Y Minimum)	数値	Y軸の最小値
Y最大 (Y Maximum)	数値	Y軸の最大値
マウス作用 (Mouse interaction)	チェック枠	図表要素でのマウス作用を許可
右に適応 (Fit to right)	チェック枠	図表要素を計器盤右端まで拡大
自動尺度 (Autoscale)	チェック枠	作図データに従ってY軸を自動尺度調整
時間によってスクロール (Scroll by time)	チェック枠	時間によってX軸をスクロール。チェックなしの場合、X軸はやって来る作図採取(試料)によってスクロールします。
作図表示 (Show plot)	チェック枠	継続する図表作図表示(相互接続された採取(試料)点)
点表示 (Show points)	チェック枠	単一採取(試料)を点として表示

3.5.1.2.11. 円図表

円図表(Pie Chart)要素はやって来る流れの値を円図表の部分として表示します。



3.5.1.2.11.1. 端点

円図表(Pie Chart)要素は円図表内の各部分に対して1つの吸い込みロ端点を持ちます。吸い込みロ端点は全ての数値データ型を 受け入れます。

3.5.1.2.11.2. 構成設定

表3-12. 円図表特有パラメータ		
パラメータ	型	使い方
表題色 (Title Color)	色	表題文章の色を設定
背景色 (Background Color)	色	要素の背景色を設定
表題 (Title)	文字列	要素の表題
作図数 (Number of plots)	数值	円図表の部分数

3.5.1.2.12. 矩形

矩形(Rectangle)要素はマウスによってクリックされる毎にパケットを送ります。



3.5.1.2.12.1. 端点

矩形(Rectangle)要素はuint32型の供給源端点を持ちます。要素がマウス ポインタによってクリックされる毎に0の値を持つパケットが送られます。

3.5.1.2.12.2. 構成設定

表3-13. 矩形特有パラメータ

パ゚ラメータ	型	使い方
背景色 (Background Color)	色	矩形の中詰めの色を選択
前景色 (Foreground Color)	色	矩形の枠の色を選択

3.5.1.2.13. 表面図表

表面(Surface)図表要素は3D空間の表面として格子を 表示します。



3.5.1.2.13.1. 端点

表面(Surface)図表要素は格子(grid)型のどの供給元も受け入れる1つの端点を持ちます。

3.5.1.2.13.2. 構成設定

表3-14. 表面図表特有パラメータ

パラメータ	型	使い方
中詰め色 (Fill color)	色	表面中詰めの色を選択
網目色 (Mesh color)	色	表面網目の色を選択
背景色 (Background color)	色	背景の色を選択
背景階調色 (Background gradient color)	色	背景階調の色を選択
軸色 (Axis color)	色	軸の色を選択
刻時色 (Tick color)	色	刻時表題の色を選択
X回転 (X Rotation)	数値	X周囲視野の回転を設定
Y回転 (Y Rotation)	数値	Y周囲視野の回転を設定
Z回転 (Z Rotation)	数値	Z周囲視野の回転を設定
X軸表示 (Show X-axis)	チェック枠	X軸の可視性を設定
Y軸表示 (Show Y-axis)	チェック枠	Y軸の可視性を設定
Z軸表示 (Show Z-axis)	チェック枠	Z軸の可視性を設定
中詰め表示 (Show fill)	チェック枠	表面中詰めの可視性を設定
網目表示 (Show mesh)	チェック枠	表面網目の可視性を設定
調色板色付け使用 (Use palette coloring)	チェック枠	調色板色付けの使用設定 (赤-黄-緑-白)
尺度調整動作 (Scaling mode)	引き落とし枠	Y軸自動尺度の動作を選択
軸最小 (Axis minimum)	数值	Yに対する最小軸値を設定
軸最大 (Axis maximum)	数值	Yに対する最大軸値を設定

3.5.1.2.14. 表

表(Table)要素は表で1つ以上のデータ供給元を表示します。自動表題(Atuo Labels)と手動表題(Manual Labels)の2つの動作形態が 支援されます。自動表題動作では各セルが2つの領域に分けられ、左の領域はデータの流れの名前を持つ表題で、右の領域は実際の 流れデータです。手動表題動作では各セルが表題セルまたはデータセルのどちらかに手動で構成設定することができます。この動作形態 は構成設定で自動表題(Auto Labels)と名付けられたチェック枠によって選ばれます。「構成設定」をご覧ください。

3.5.1.2.14.1. 自動表題

自動表題動作使用時に各セルは1つのデータ供給元と関連して、データ供給元の名前がセルの左に、実際のデータが右に表示されます。 供給元名は吸い込みロ端点に接続される供給元から自動的に取得されます。(下図左側参照)

表(Table)要素はどのデータ供給元も受け入れる表のセル毎に1つの端点を持ちます。データは文字列に変換されて文として表示されます。(下図右側参照)

Delta1	0	Delta6	0
Delta2	0	Delta7	0
Delta3	0	Delta8	0
Delta4	0	Delta9	0
Delta5	0	Delta10	0

M ilta1	0	B lta6	0
lta2	0	lta7	0
ita 3	0	lta8	0
M lta4	0	lta9	1
lta5	0	ta10	0

端点は端点を表示(Show Endpoints)任意選択が選ばれる時に表示されます。



LABEL B

LABEL A

LABEL C

LABEL D

3.5.1.2.14.2. 手動表題

手動表題動作使用時に各セルは表題セルまたはデータセルのどちらかです。既定によって全てのセルはデータ セルです。表題セルは表題構成設定(Label Configuration)文字列の設定によって構成設定することができ ます。「構成設定」をご覧ください。表題構成設定(Label Configuration)文字列はセミコロン';'で分離された 表題の一覧を与えることによってどのセルが表題かを構成設定します。各表題は<列>:<行>:<表題>の形 式によって与えられます。左上のセルが0列、0行です。例が右で与えられます。

この表に対する表題構成設定(Label Configuration)は次のとおりです。

0:0:LABEL A;1:0:LABEL B;0:1:LABEL C;0:2:LABEL D

データセルだけが端点(Endopoints)を持ちます(右下の図)。

表題文章は表題構成設定(Label Configuration)解析部を混乱させることを避けるためコロン':'やセミコロン';'を含むべきではありません。

端点は端点を表示(Show Endpoints)任意選択が選ばれる時に表示されます。





۲

3.5.1.2.14.3. 構成設定

表3-15. 表特有パラメータ		
パ・ラメータ	型	使い方
データフォントの大きさ (Data Font Size)	数値	セルのデータ部のフォントの大きさを設定
表題フォントの大きさ(Label Font Size)	数値	セルの表題部のフォントの大きさを設定
データ列幅 (Data Column Width)	数値	各セルのデータ部の幅。この幅の変更が表全体の幅を変えることに注意してください。
表題列幅 (Label Column Width)	数値	各セルの表題部の幅。この幅の変更が表全体の幅を変えることに注意してください。
行高さ (Row Height)	数値	表内の各行の高さ。この高さの変更が表全体の高さを変えることに 注意してください。
行数 (Number of Rows)	数値	表内の行数
列数 (Number of Columns)	数值	表内の列数
自動表題(Auto Labels)	チェック枠	自動表題動作許可。禁止の場合、表題は手動で構成設定されなければなりません。
表題構成設定(Label Configuration)	文字列	手動表題動作(自動表題動作禁止)時に表題を構成設定する文字 列。形式は、〈列〉:〈行〉:〈表題〉;〈列〉:〈行〉:〈表題〉~です。
データ太字体 (Data Bold)	チェック枠	各セルのデータ部のフォントの形式を太字体に設定
データ斜体 (Data Italic)	チェック枠	各セルのデータ部のフォントの形式を斜体に設定
表題太字体 (Label Bold)	チェック枠	各セルの表題部のフォントの形式を太字体に設定
表題斜体(Label Italic)	チェック枠	各セルの表題部のフォントの形式を斜体に設定
背景色 (Background color)	色	表の背景色を設定
前景色 (Foreground color)	色	表の格子それとデータと表題の文の色を設定
表題水平整列 (Label Horizontal Alignment)	引き落とし枠	各セルの表題部で文の配置を選択(左(Left),中央(Center),右(Right))
テ [·] ータ水平整列 (Data Horizontal Alignment)	引き落とし枠	各セルのデータ部で文の配置を選択(左(Left),中央(Center),右(Right))

3.5.2. 計器盤構成設定例

本項は計器盤(Dashboard)単位部をどう構成設定するかの例を与えます。この例がデータ可視器(Data Visualizer)で利用可能なデータ 供給元と利用可能な計器盤要素の部分集合とは言え、基本的な原則は全ての要素とデータ供給元に適用できます。

この例は計器盤(Dashboard)単位部の手動構成設定を使いますが、計器盤を自動的に構成設定するためのAtmelデータ規約(ADP)を 使うことも可能です。ADPと自動構成設定された計器盤の例のより多くの情報については「Atmelデータ規約」をご覧ください。

この例で使った目的対象応用コート・とハート・ウェア構成設定の記述は「計器盤コート・例」で見つけることができます。



この例では計器盤用表題としての1つと摺動子用手助け文としての他の2つで、3つの表題(Label)要素が追加されます。1つの作図を 持つ図表(Graph)要素は光感知器データに使われるために追加されました。Y最小(Y Minimum)とY最大(Y Maximum)の値は各々、0と 255に設定されました。信号(Signal)要素はどの動作形態が活性かを見ることができるように追加されました。夜間動作が活性の時に 信号は暗い青(Color ON)に切り替わり、夜間動作が非活性の時に黄色(Color OFF)に切り替わります。最後に、摺動子は夜間動作 閾値を調整することを可能にするために追加されました。最小(Minimum)は0に設定され、最大(Maximum)は255に設定されました。摺 動子の左移動は閾値を低くし、夜間動作がより明るい光水準で活性になることに帰着します。

計器盤が構成設定されてしまうと、目的対象応用との通信を許可するために計器盤をシリアルインターフェースに接続する時間です。 計器盤の端点(endpoints)を結合することができるのに先立って、目的対象基板とホストコンピュータ間のインターフェースが構成設定されなけ ればなりません。この例はDGI SPIインターフェースとCDC USARTインターフェースを使います。CDCインターフェースはホストコンピュータ上で通常のシ リアルCOMポートとして現れます。

▶ すること: DGI制御盤(DGI Control Panel)で正しいツールを選んでください。

ATmega256RFR2 Xplained Pro	Connect
ATML1784031800003140	Start
	ADP Logging Autodetect protocols Reset MCU
	ADP Logging 🔲 Autodetect protocols 📝 Reset MCU

すること: 選んだツールでDGIへの接続をするために接続(Connect)をクリックしてください。

ATmega256RFR2 Xplained Pr ATML1784031800003140	0	- Disconnect Start
interfaces:	GPIO GPIO O O O O O O O O O O O O O O O O O O	ADP Logging Autodetect protocols Reset MCU Code Profiling

「、すること: ・ SPIチェック枠をクリックしてください。

・SPIチェック枠傍らの歯車釦をクリックすることによってSPI構成設定(SPI Configuration)ダイアログを開いてください。

ATmega2568 ATML178403180000	RFR2 Xplained Pro		Disconnect Start
Interfaces: SPI	SPI Configuration Transfer Mode Force startup synchronizatio Enable timestamping	SCK normally low, Read data on rising edge n on CS OK Cancel] Reset MCU





すること:・接続したキットに対応する正しいCOMポートを選んでください。

・応用コートーに従ってシリアルホートのハウメータを設定してください。

すること:・データ可視器(Data Visualizer)で構成設定(Configuration)タブの単位部(Modul

es)部で外部接続(External Connection)下で見つかるシリアルポート(Serial Por

・端末を開く(Open Terminal)任意選択がチェックされていないことを確実にしてください。

Baud rate Parity Stop bits DTR RTS Autodetect protocols	EDBG Virtual COM Port (COM127)		✓ Connect
Baud rate Parity Stop bits Autodetect protocols			🛛 🗹 DTR 🔲 RTS
Baud rate Parity Stop bits	Baud rate Parity	Stop bits	Open Terminal
			Autodetect protocols

すること:・編集(Edit)任意選択を選択解除にしてください。

t)制御盤を開いてください。

・端点を表示(Show Endpoints)任意選択をクリックしてください。

- ・SPIの供給源を図表の吸い込み口に引き摺ってください。
- ・シリアルポートの供給源を信号の吸い込み口に引き摺ってください。
- ・摺動子の供給源をシリアルポートの吸い込み口に引き摺ってください。

DGI Control Panel ATmega256RFR2 Xplained Pro ATMLI78409020000573 Interfaces: SPI 🔯 🛛 TWI 🔯 🔤 GPIO 🛱	Code Profiling	ADP Logging Autodetect ADP Reset MCU
Deshboard I/O Fight Mode Switch Light sensor data 250 150 150 0 0000 0005 Auto-scroll Automatically fit Y Legend SPI Output	BAGnter C	► ×
Edit 📝 Show Endpoints		
Serial Port Control Panel COM82 Baud rate Parity Stop bits 9600 None * 1 bit *	•	Disconnect Disconnect Open Terminal Autodetect ADP
今や計器盤は完全に構成設定され、夜間動作切り替え応用で情報交換するのに使うことができます。

- っ、すること:・端点を表示(Show Endpoints)任意選択を選択解除にしてください。
 - ・DGI制御盤(DGI Control Panel)で開始(Start)をクリックしてください。
 - ・シリアルポート制御盤(Serial Port Control Panel)で接続(Connect)をクリックしてください。



今やADCの生の値が図表で表示され、摺動子は夜間動作切り替えのために適切な閾値に調整することができます。信号(Signal)要素は切り替え器の状態を示します。

4. ユーティリティ

コーティリティは他の区部に合わない単位部ですが、未だデータを分析するのに有用です。

採取速度計数器(Samplerate Counter)はどれ位の程度のデータが転送されつつあるかを提供します。

ファイル記録(File Logger)単位部はやって来るデータを選択可能な形式のファイルに格納します。ファイル内容はその後に別の応用で分析 することができます。

4.1. 採取速度計数器

採取速度計数器単位部はやって来るデータの流れを取って流れに於けるやって来る採取(試料)の量を測定します。



採取速度計数器を使うには単に供給源を採取速度計数器単位部の吸い込み口に接続してデータの流れを開始してください。

4.2. ファイル記録部

ファイル記録(File Logger)単位部はやって来るデータを選択可能な形式のファイルに記録します。

図4-1.	7ァイル記録		
L	og to File	^ X	1. 出力ファイル(Output file)選択枠
1-(File		2. 出力形式(Output format)選択枠 3. 吸い込みロ入力
2	Type ◎ RAW ④ CSV ◎ BIN ◎ ASCII ◎ HEX 📝 Relative timestamps	\supset	4. 開始/停止(Start/Stop)釦
	Start		
	3 4		

4.2.1. 2進ファイルに記録

- ・出力形式(Type)選択枠で"BIN"出力形式を選んでください。
- ・出力ファイル(File)選択枠で"…"釦を押してパスと名前を選ぶことによって出力ファイルを設定してください。
- ・外部供給源を吸い込みロ入力に接続してください。
- ・記録を始めるために開始(Start)釦を押してください。釦は停止(Stop)釦で置き換えられます。
- ・記録されたデータを調査するのに先立ってファイルを閉じて記録を終えるために停止(Stop)釦を押してください。

<u>5</u>. 規約

殆どの通信インターフェースはデータを転送するのにバイトの流れを使います。これは8ビット精度の単一データ値に対して充分ですが、複数の値が同じインターフェース上で転送されることが必要とされる時にデータは規約で梱包されなければなりません。データ可視器(Data Visual izer)はそのような2つの規約を支援します。

データ流れ規約(Data Stream Protocol)は1つのインターフェース上でいくつかの数値を梱包するのに軽いフレーム形式を使います。これはやって来るデータの処理の能力だけで、同期の流れだけを支援します(即ち、全てのデータパケットは各データの流れからの1つの採取(試料)を含まなければなりません)。

Atmelデータ規約(ADP:Atmel Data Protocol)は双方向でのデータの流れを許します。ADPは一般的にデータ流れ規約よりももっと柔軟で、非同期の流れも支援します(即ち、各データの流れからのデータ採取(試料)は独立して送ることができます。)。

両方の規約は自動的に検出することができ、データの流れを定義し、計器盤を構成設定してデータの流れを計器盤の要素に接続する ことによってデータ可視器(Data Visualizer)を自動的に構成設定するのに使うことができます。主な違いはデータ流れ規約については 構成設定がホストコンピュータの構成設定ファイルに存在し、一方でADPについては構成設定が目的対象応用からホストコンピュータへ送られ ます。加えて、ADPは図表(Graph)と端末(Terminal)の表示部を構成設定することができます。計器盤表示部内の図表(Graph)要素は 両規約によって支援されます。

表5-1. 規約の比較

項目	デー タ流れ 規約	Atmelデータ 規約 (ADP)	注釈
複雑さ	簡単/軽い	複雑/柔軟	
個別の流れの開始/停止	×	0	
データ入力	0	0	目的対象応用からホストコンピュータへのデータ送信
データ出力	×	0	ホスト コンピュータから目的対象応用へのデータ送信
文字列データ型支援	×	0	様々な長さの文字列
ホスト コンピュータに格納された構成設定	0	×	データ可視器GUIの構成設定
目的対象応用に格納された構成設定	×	0	データ可視器GUIの構成設定
計器盤表示部構成設定	0	0	
図表表示部構成設定	×	0	
端末表示部構成設定	×	0	
データ集中	×	0	データの流れの複数採取(試料)を1つのパケットで送ることができます。
非同期のデータの流れ	×	0	違うデータの流れからの採取(試料)を独立して送ることができます。

○=支援、×=不支援

5.1. データ流れ規約

データ流れ単位部はやって来る生データの流れを取って複数のデータの流れに分割します。データ流れ形式は使用者によって提供される構成設定ファイルによって指定されます。

5.1.1. 構成設定形式

構成設定ファイルは行毎に1つのデータを指定するコンマ','で区切られ文章ファイルです。各行は次表で提示されるタグの1つによって変数のデータ形式を指定することによって始まります。出力格子内の変数の位置はその後に指標1で始まる2つの座標によって与えられます。最後のパラメータは変数に対して文章文字列を割り当てます。

表5-2. データ流れ型

型	大きさ	タク゛	例
符号なしバイト	1	В	B,1,1,Light
符号付きバイト	1	-B	-B,1,1,Encoder
符号なしショート	2	D	D,1,1,ADC
符号付きショート	2	-D	-D,1,1,ADC
符号なし語	4	W	W,1,1,Transfer rate
符号付き語	4	-W	-W,1,1,Status code
浮動小数点	4	F	F,1,1,Temperature
倍精度浮動小数点	8	DF	DF,1,1,Measurement
符号なしバイトの格子	$1 \times W \times D$	GB <wxd></wxd>	GB<10x10>,1,1,Surface
符号付きバイトの格子	$1 \times W \times D$	-GB <wxd></wxd>	-GB<10x10>,1,1,Surface
符号なしショートの格子	$2 \times W \times D$	GD <wxd></wxd>	GD<10x10>,1,1,Surface
符号付きショートの格子	$2 \times W \times D$	-GD <wxd></wxd>	-GD<10x10>,1,1,Surface
符号なし語の格子	$4 \times W \times D$	GW <wxd></wxd>	GW<10x10>,1,1,Surface
符号付き語の格子	$4 \times W \times D$	-GW <wxd></wxd>	-GW<10x10>,1,1,Surface
浮動小数点の格子	$4 \times W \times D$	GF <wxd></wxd>	GF<10x10>,1,1,Surface
倍精度浮動小数点の格子	$8 \times W \times D$	GDF <wxd></wxd>	GDF<10x10>,1,1,Surface

これは構成設定例です。

- D, 1, 1, ADCO
- D, 1, 2, ADC1
- D, 1, 3, ADC2 B, 2, 1, Prescaler

5.1.2. 流れ形式

データの流れは構成設定ファイルの指定と同じ順番で処理されます。全てのデータは最下位ハイトが最初に送られなければならないことを 意味するリトル ェンディアン値として与えられなければなりません。加えて、データの流れ変数の前の1ハイトと後ろの1ハイトから成る囲み部 が追加されなければなりません。この囲み部はデータの流れを同期するために解釈部によって使われます。開始ハイトは自動構成設 定用に予約されている\$5Fを除く任意の値にすることができますが、終了ハイトは開始ハイトの反転でなければなりません。自動構成設 定のより多くの情報については「自動構成設定」と「自動構成設定例」をご覧ください。構成設定ファイルは開始と終了のハイトを定義す べきではありません。

前項で与えられた構成設定例を考察してください。下図はADC0が185、ADC1が950、ADC2が0、Prescaler(前置分周器)が2の生デー タ送信例を与えます。

図5-1. デー	タの流れ		
開始	ADC0	ADC1	ADC2 Prescaler 終了
\sim \$03	\$B9 \$00	\$B6 \$03	\$00 \$00 \$02 \$FC ~

5.1.3. 基本的な使い方

圛5	-2. データ流れ音	fβ		
	Data Stream Cor	ntrol Panel		×
	Configuration	C:\adc.txt	Load 💭 Reset	6
	🚅 ADC			
1	7	2	3 4 5	

・ファイルを開く釦を押してください。(3)

- ・構成設定ファイルを選んでください。
- ・構成設定読み込み(Load)釦をクリックしてください。(4)
- ・吸い込み口入力を接続してください。(5)
- ・1つ以上の供給源出力を望む吸い込み口に接続してください。(1)

5.1.4. 自動構成設定

データ流れ形式はいくつかの予め定義された構成設定ファイルに基づいてデータ可視器(Data Visualizer)を自動的に構成設定するのに 使うことができます。これは構成設定が目的対象応用に格納されて要求でホストへ送られるAtmelデータ規約(ADP)自動構成設定と異な ります。データ流れ自動構成設定はデータの流れの構成設定、計器盤を構成設定してデータ流れ供給元出力を計器盤の要素へ接続す ることに制限されます。

5.1.5. 自動構成設定例

この例の目的は目的対象ハートウェア上の押し釦の状態を表示する信号(Signal)と目的対象応用での計数器の値を表示する図表(Grap h)を持つ計器盤を自動構成設定するために必要とされる構成設定ファイルをどう構成設定するかを示すことです。下の複写画像は最終的な計器盤を示します。



この例で使われる目的対象コードは「自動構成設定コート例」で見つけることができます。

自動構成設定を許すにはデータ可視器(Data Visualizer)によってデータの流れの内容を記述する.dsファイル、計器盤を記述する.dbファイル、データの流れの内容と計器盤の要素の間の接続を記述する.scファイルの3つのファイルが必要とされます。全てのファイルは応用コートによって送られる自動構成設定IDに合うように拡張子の前が"COFFEEC0FFEEC0FFEEC0FFEE"と名付けられるべきです。 例応用は符号なし16ビット値と符号なし8ビット値の2つの流れ構成部分を持つデータの流れを生成します。

すること: "COFFEECOFFEECOFFEE.ds"と呼ばれるファイルを作成して下の内容を追加してください。 D, 1, 1, count B, 2, 1, button

.dsファイルはデータの流れから2つの供給元を持つ表を作成します。列1行1は"count"と名付けられた符号なし16ビット値を持つ供給元 を含み、列2行1は"button"と名付けられた符号なし8ビット値を持つ供給元を含みます。.dsファイルのより多くの情報については「構成 設定形式」をご覧ください。

自動構成設定用に計器盤を構成設定する最も簡単な方法はデータ可視器(Data Visualizer)のGUIでそれを描いてその構成設定をファ イルに保存することです。



助言:構成設定(Configuration)部分でパラメータ変更は設定(Set)釦がクリックされるまで有効ではありません。

ー、すること:・表題(Label)要素を計器盤に引き摺ってください。

表題(Label)要素を選択してフォントの大きさ(Font Size)を16に設定、太字体(Bold)任意選択をチェック、文章(Text)領域を"Button State"(釦状態)に設定、幅(Width)を100に設定してください。

・設定(Set)を押してください。

Button Sta		Aa Label Image: Description Table I2 Numeric Input I2 Numeric Input I2 Button I2 On I2 Radio Group I Check Box I Slider I Signal
Edit Show Gridlines Configuration Z-Index	Show Endpoints Load Save	*
Left Top	63 48	
Width Height	100 22	
Font Size Bold	16	
Italic		
Horizontal Alignment	Top -	
Background Color Foreground Color		
Text	Button State	

「「」、すること:・信号(Signal)要素を計器盤に引き摺ってそれを釦状態(Button State)表題の下に配置してください。

- ・図表(Graph)要素を計器盤に引き摺ってそれを釦状態(Button State)表題と信号(Signal)の右に配置してください。
- ・図表(Graph)要素を選択して表題(Ttitle)を"count value"(計数値)、X最大(X Maximum)を5、Y最大(Y Maximum)を66000に設定してください。
- ・設定(Set)をクリックしてください。

Dashboard I/O		^)
	count value	lio Group 🕺 🔊
Button Stat	60000 - Che	ck Box
	Slid	er 37
	40000 - Sign	nal
	Pro	gress bar
	20000 - Seg	ment Display =
	Gra	ph
		Chart
	Auto-scroll Automatically fit Y	face Grid
	Legend Unconnected	*
🛛 Edit 🔲 Show Gridlines 🛽	Show Endpoints Load Save	
Configuration		
Z-Index	0	
Left	206	
Тор	12	
Width	320	
Height	240	
Title color		
Background color		
Graph background color		
Title	count value	
Number of plots	1	
X Minimum	0	
X Maximum	5	
Y Minimum	0	
Y Maximum	66000	
Mouse Interaction	V	
Fit to Right		
Autoscale		
Scroll by time		
Show plot	V	
Show points		
	Set	

📝 すること:・保存(Save)をクリックしてください。

	count va	alue
Button State	60000 -	
	40000	
	20000 -	
	o -	
	00:00:00 V Auto-scroll Mutomatically	00:00:0
	Legend Unconnected	

溄 すること: ・計器盤構成設定を"C0FFEEC0FFEEC0FFEEC0FFEE.db"として.dsファイルと同じフォルダに保存してください。

```
{
0,
'¥0',
0, 255, 255, 255,
44, 1,
};
{
0, // Dashboard ID (計器盤ID)
0, // Element ID (要素ID)
DB_TYPE_LABEL, // Element Type(要素形式)
0, // Z-Index (GUI stack order) (Z指標(GUI階層順))
63, 0, // X-coordinate (X座標)
48, 0, // Y-coordinate (Y座標)
100, 0, // Width (幅)
22, 0, // Height (高さ)
16, // Font Size (フォントの大きさ)
1,
0, // Horizontal Alignment (水平整列)
0, // Vertical Alignment (垂直整列)
0, 255, 255, 255, // Background Color (背景色)
255, 0, 0, 0, // Foreground Color (前景色)
'B', 'u', 't', 't', 'o', 'n', '', 'S', 't', 'a', 't', 'e', '¥0', // Text (文章)
};
0, // Dashboard ID (計器盤ID)
1, // Element ID (要素ID)
```

[.]dbファイル内容は以下と同様の内容を持つべきです。

```
DB_TYPE_SIGNAL, // Element Type (要素形式)
0, // Z-Index (GUI stack order) (Z指標(GUI階層順))
98, 0, // X-coordinate (X座標)
78, 0, // Y-coordinate (Y座標)
25, 0, // Width (幅)
25, 0, // Height (高さ)
255, 0, 255, 0, // Color On (ON色)
255, 255, 0, 0, // Color Off (OFF色)
};
0, // Dashboard ID (計器盤ID)
2, // Element ID (要素ID)
DB_TYPE_GRAPH, // Element Type (要素形式)
0, // Z-Index (GUI stack order) (Z指標(GUI階層順))
206, 0, // X-coordinate (X座標)
12, 0, // Y-coordinate (Y座標)
64, 1, // Width (幅)
240, 0, // Height (高さ)
255, 255, 255, // Title color (表題色)
0, 0, 0, // Background color (背景色)
20, 20, 20, // Graph background color (図表背景色)
'c', 'o', 'u', 'n', 't', ' ', 'v', 'a', 'l', 'u', 'e', '¥O', // Title (表題)
1, // Number of plots (作図数)
0,0,0,0, // X Minimum (X最小)
0,0,160,64, // X Maximum (X最大)
0,0,0,0, // Y Minimum (Y最小)
0,232,128,71, // Y Maximum (Y最大)
1,
1,
};
```

計器盤の要素のより多くの情報については「要素形式」をご覧ください。

最後に、.dsファイルで定義されたデータ流れ供給元を.dbファイルで定義された計器盤の要素に接続するための構成設定ファイルが作られなければなりません。

すること: ・他の構成設定ファイルと同じファルダに"COFFEEC0FFEEC0FFEEC0FFEE.sc"と呼ばれるファイルを作成して下の内容を追加してください。

button, 1 cont, 2

.sc7rイルは"button"と名付けられた流れ供給元をID 1を持つ計器盤要素(信号(Signal)要素)に、"count"と名付けられた流れ供給元 をID 2を持つ計器盤要素(図表(Graph)要素)に接続します。 これでは何ちまさせる時間です

それでは例を走らせる時間です。



	Configuration
6	Modules
nfiguration	 ✓ External Connection DGI Control Panel Serial Port > Graph Terminal > Protocols > Utilities I/O Dashboard Logging

一、すること:・接続したキットに対応する正しいCOMホートを選んでください。

・規約自動検出(Autodetect protocols)任意選択がチェックされ、ハ゜リティ(Parity)と停止ビット(Stop bits)の構成設定が目的対象応用に従って設定されていることを確実にしてください。ボーレート(Baud rate)は自動的に検出されます。

mEDBG	Virtual C	OM Port (COM135)	✓ Connect
			DTR 🔲 RTS
Paud rata	Davita	Stan hits	Open Terminal
baud rate	Panty		Autodetect protocols
38400	None 🔻	1 bit 🔻 💼	Show Config search path

か、
すること:
・
接続(Connect)をクリックしてください。

・データ可視器(Data Visualizer)が自動構成設定IDに対する構成設定ファイルを見つけることができない場合、構成設定ファイルへのパスを問うポップアップを表示します。構成設定が存在するフォルダを検索してOKをクリックしてください。

	puter 🖡 OSDISK (C:) 🖡 Data Stre	am Example Config Files	▼ * <u>†</u>	Search Data Stream E	ample
Organize 👻 🛛 New f	folder			1	• @
🔆 Favorites	Name	Date modified	Туре	Size	
詞 Libraries		No items match you	ir search.		
🖳 Computer					
🛛 🏭 OSDisk (C:)					
📬 Network					
Fr	older: Data Stream Example Cont	ig Files			

フォルダ選択後、フォルダは自動構成設定検索パスに追加されます。

助言:検索パスをリセットして検索パスとして新しいフォルダを選ぶには規約自動検出(Autodetect protocols)任意選択文のリンクをクリックしてください。 データ可視器はその後に構成設定ファイルが存在するフォルダへのパスを問う閲覧ダイアログを 飛び出させます。元の検索パスは解消され、新たに選んだフォルダが検索パスとして設定されます。



重要: 3つ全ての構成設定ファイルは同じフォルダに存在しなければなりません。

接続と自動構成設定ID検出後、データ可視器は次の画像のような何かを見るデータ流れ制御盤(Data Stream Control Panel)計器盤入出力(Dashboard I/O)を作成するでしょう。



図表(Graph)要素は丸められるまで継続的に上昇計数する計数を表現する鋸歯信号の走行を表示します。信号(Signal)要素はATtin y104 Xplained Nano基板上の押し釦の状態を示します。釦押下は信号(Signal)要素の色を赤から緑に変えます。

盤の右側の下向き矢印をクリックすることによってデータ流れ制御盤(Data Stream Control Panel)を展開すると、自動的に構成設定されたデータの流れの内容を表示します。

Data Stream Control Panel	Data Stream Control Panel	⊘ ×
---------------------------	---------------------------	------------

この流れは、計数器(counter)用の1つと釦状態(button)用の1つの、2つの供給元を持ちます。

Data Stream Cor	trol Panel		^ X
Configuration	C:\user\work\datavisualizer\user	Load Meset	
Count		Cee button	

シリアルポート制御盤(Serial Port Control Panel)を展開すると、データ可視器が38400のボーレートで検出したことを表示します。

Serial Port Cor	ntrol Panel		^ X
mEDBG Virtual COM Port (COM135)			Disconnect DTR RTS
Baud rate	Parity	Stop bits	 Open Terminal Autodetect protocols
38400	None 🔻	1 bit 🔻 🌍	Show Config search path

5.1.6. 自動構成設定形式

データ流れパケットの開始バイトが\$5Fの場合、このパケットは特別な構成設定パケットです。

表5−3. 構成設定パケット形式				
領域	大きさ	値	説明	
開始通票	1バイト	\$5F	構成設定パケット用に予約された開始通票	
チェックサム形式	4バイト	\$B4 \$00 \$86 \$4A	使われるチェックサム形式を指定。現在はLRC8だけを支援。	
構成設定識別子	12ハイト	何れかの値	構成設定のための固有識別子	
チェックサム	1시 イト	チェックサム形式に 従ったチェックサム	現在、LRC8チェックサム形式だけが支援されます。これは開始通票、チェックサム自身、終了通票を除くパケットのXOR総和です。	
終了通票	1/ㆍ/ㅏ	\$A0	データ流れ形式に従って終了通票は開始通票の逆数です。	

識別子はデータ可視器(Data Visualizer)を構成設定するのに使われる対応する構成設定ファイルを探すため、データ可視器によって使われる構成設定パケットで与えられます。以下のように3つの構成設定ファイルが必要とされます。

- ・データの流れを定義する.dsファイル。これは標準データ流れ形式ファイルで「構成設定形式」で与えらえれた形式に従います。
- ・計器盤を定義する.dbファイル。このファイルは計器盤をファイルに保存する時に生成されるファイルの形式に従います。「編集操作盤」をご 覧ください。
- ・.dsファイルで定義されるデータ流れ構成部分と.dbファイルで定義される計器盤の要素間の接続を定義する.scファイル。形式は「信号接続ファイル形式」で定義されます。

3つ全ての構成設定ファイルは各構成設定識別子バイトの16進値に等しい名前を持つべきです。例として、[\$12, \$34, \$56, \$78, \$9A, \$BC, \$DE, \$F0, \$12, \$34, \$56, \$78]の構成設定識別子は123456789ABCDEF012345678.sc、123456789ABCDEF012345678.db、 123456789ABCDEF012345678.dsと名付けられた構成設定ファイルに対応します。

データ流れ部自動構成設定機能はDGI直列インターフェース用DGI制御盤(DGI Control Panel)と、COMホートと仮想OCMホート(CDCインター フェース)用のシリアルホート制御盤(Serial Port Control Panel)の両方で利用可能です。

自動構成設定を許可するには規約自動検出(Autodetect protocols)任意選択が許可されなければなりません。

Autodetect protocols

接続(Connect)押下後、データ可視器は全てのインターフェースを許可すると同時にADPハント・シェークメッセージ・またはデータ流れ構成設定ハ[°] ケットを探します。ADPハント・シェークメッセージが受信された場合、データ可視器は目的対象応用からの構成設定情報を要求します。データ 流れ構成設定ハ[°]ケットが見つかった場合、データ可視器は検出したIDと一致する名前を持つ構成設定ファイルを探すために自動構成設 定検索ハ[°]スでフォルダを通して検索します。

重要: データ可視器がデータ流れ構成設定パケットを検出することを確実にするため、秒毎に最低2度目的対象によってそれが送られなければなりません。

•

重要: 非同期シリアル規約(例えば、DGI USARTとCDC仮想COMポート インターフェースで使われるUART規約)は自動検出のため以下のボーレートを使います。

表5-4. 規約の自動検出に関して非同期インターフェースで使われるボーレート

ホーレート 9600 19200 38400 57600 115200 230400 500000 1000000 2000000

この表にない何れかの速度を使うことは非同期インターフェース(DGI UARTとシリアル ポート/CDC仮想COMポート)上の規約の 自動検出に対して動きません。

助言: 構成設定ファイルを探すためにデータ可視器によって使われる現在の検索パスを見るには構成設定検索パス表示(Show Config search path)任意選択をチェックしてください。

Show Config search path

検索パスはセミコロン(;)で分離されたパスの一覧です。データ可視器が自動構成設定IDを検出すると、正しいファイル名を持つ構成設定ファイルを探す一覧内のパスを通して検索します。

Config search path C:\Data Visualizer Config Files; C:\Data Stream Example Config Files;

データ可視器が有効な何れかの構成設定ファイルを見つけることができない場合、正しい構成設定ファイルが存在するフォルダへのパスを問う閲覧ダイアログウインドウが現れます。

フォルダ選択後、フォルダは自動構成設定検索パスに追加されます。



助言: 検索パスをリセットして検索パスとして新しいフォルダを選ぶには規約自動検出(Autodetect protocols)任意選択文のリンクをクリックしてください。

データ可視器はその後に構成設定ファイルが存在するフォルダへのパスを問う閲覧ダイアロクを 飛び出させます。元の検索パスは解消され、新たに選んだフォルダが検索パスとして設定さ れます。



● 重要: 3つ

重要: 3つ全ての構成設定ファイルは同じフォルダに存在しなければなりません。

5.1.7. 信号接続ファイル形式

信号接続ファイルは.scファイル拡張子を持ち、.dsファイルで定義されたデータ流れ供給元と.dbファイルで定義された計器盤内のGUI要素間の 接続を指定します。.scファイル形式は行毎に1つの接続を指定するコンマ','で区切られた文章ファイルです。各行は<流れ名>,<要素ID> 形式に従います。流れ名は.dsファイルで定義され、各データ変数に割り当てられた文章文字列です。要素IDは計器盤の各要素に対し て.dbファイルで定義されます。

.scファイル内容の例は次のとおりです。

Delta1,2	
Delta2,2	

Planeと呼ばれる(データの)流れがID 0を持つ要素に接続され、Delta 1とDelta 2の両(データの)流れはID 2を持つ要素に接続されます。完全な自動構成設定例については「自動構成設定例」をご覧ください。

表(Table)要素(「表」参照)は(データの)流れ名と要素IDに加えていくつかの追加パラメータを持ちます。信号に接続するセルの列と行は.sc ファイルの行に対する付加(列(Column):<列番号>;行(Row):<行番号>)によって与え

られます。左上のセルは列0と行0によって指定されます。例として、右の表は以下の.sc7rイル内容によって供給元に接続されます。

Delta1,2(Column:1;Row:1)	F
Delta2,2(Column:1;Row:2)	
Delta3,2(Column:1;Row:3)	┡
Delta4,2(Column:1;Row:4)	
Delta5,2(Column:1;Row:5)	
Delta6,2(Column:1;Row:6)	
Delta7,2(Column:1;Row:7)	
Delta8,2(Column:1;Row:8)	L
Delta9,2(Column:1;Row:9)	
Delta10,2(Column:1;Row:10)	

Delta1	0	Delta6	0
Delta2	0	Delta7	0
Delta3	0	Delta8	0
Delta4	0	Delta9	0
Delta5	0	Delta10	0

この例では表要素が自動表題動作で、故に各セルは2つの領域、左に表題、右に実際のデータを持ちます。より多くの情報については 「自動表題」をご覧ください。

図表(Graph)要素(「図表」参照)は(データの)流れ名と要素IDに加えて追加パラメータも支援します。既定により、全ての作図は図表要素 で可視ですが、それらは作図に対応する図表表示部内の凡例(Legend)を使用者がクリックすることによって非表示または表示にするこ とができます。支援される追加パラメータは自動構成設定を行う時に作図を非表示にするように既定の動きを変更することを可能にしま す。これは.scファイルに(visible:0)追加することによって行われます。例として以下の.scファイルをご覧ください。

eltal,2(visible:0)	
elta2,2	
elta3,2(visible:0)	
elta4,2	
elta5,2	
elta6,2	
elta7,2(visible:0)	
elta8,2(visible:0)	
elta9,2	
elta10,2	

図表要素の結果はこのように見え ます。



図表に於いてDelta1、Delta3、Delta7、Delta8と名付けられた作図は全て非表示で、凡例(Legend)は灰色です。使用者は凡例をクリッ クすることによってそれら(の表示)を許可することができます。同じように使用者が対応する凡例をクリックすることによって見えている作 図を非表示にすることができます。

5.2. Atmelデータ規約

5.2.1. Atmelデータ規約を使う転送

Atmelデータ規約(ADP:Atmel Data Protocol)はデータ交換器インターフェース(DGI:Data Gateway Interface、「組み込みデ・バッカ」のデータ交換 器インターフェース」参照)を使うEDBGに基づくデ・バッカ(EDBG、Atmel-ICE、Power Debugger)を通して目的対象MCUからホストPCへ、また はシリアル ポートを使って直接ホスト コンピュータにデータを転送することを意図した、内容に依存しない規約です。ADPは内容に依存せず、 デ・バッカ を通す転送はディバッカ によって内容が解釈されないことを意味する透過です。

単一値の転送はかなり簡単です。しかし複数の値を転送するためにそれらは送信側と受信側の両方が理解する或る種の規約で包まれなければなりません。ADPはそのような規約です。MCUがそのデータ全てをADPパケットに包む場合、データ可視器(Data Visualizer)でそれを復号して独立したデータの流れに分離することができます。

右図で、MCUは温度と圧力の変数をADPパケット内に包 みます。データ可視器で、DGI制御盤(DGI Control Pan el)のSPI端点が今やADP制御盤(ADP Control Panel) のデータ(Data)端点に接続されます。ADP制御盤(ADP Control Panel)はハ[°]ケットを分離した温度と圧力のデータ の流れに復号します。これらはその後に図表(Graph)単 位部で2つの作図(Plot)線に接続することができます。

ADP規約は両方向でのデータ転送を支援します。加え て、MCUはデータ可視器で何の単位部が開かれてそれ らをどう接続すべきかを記述する構成設定パケットを送る ことができます。目的対象基板がホストコンピュータに接続 される時に全てのものが自動的に構成設定されます。



5.2.2. ADP例

ADP規約使用の例についてはSAM D21 Xplained Pro用ADP応用例を使うことができます。この例はAtmel StudioのAtmelソフトウェア枠 組み(ASF:Atmel Software Framework)で見つけることができます。これはI/O1 Xplained Pro基板と共にSAM D21 XPlained Proを使 います。

この例はデータ交換器インターフェース(GDI)を使いますが、どのシリアルポートでも十分です。組み込みデバッガ(EDBG)の「組み込みデバッガ」のデータ交換器インターフェース」をご覧ください。

5.2.2.1. 必要条件

- ・(データ可視器(Data Visualizerが含まれ))インストールされたAtmel Studio 7(またはそれ以降)を持つホスト コンピュータ
- ・ SAM D21 Xplained Proキット
- ・I/O1 Xplained Pro拡張

5.2.2.2. ハート・ウェア構成設定

例を走らせるには以下のハートウェア構成設定が必要とされます。

- ・SAM D21 Xplained ProのEXT1コネクタに接続されたI/O1 Xplaine d Pro拡張
- ホストPCからSAM D21 Xplained Pro上のDEBUG USBコネクタへ接続されたUSBケーブル

構成設定の写真が右で示されます。



5.2.2.3. 例走行

ADP例を走らせるには下の手順に従ってください。

すること: ・ Atmel Studioを開いてください。

- ・ファイル(File)⇒新規(New)⇒例プロジェクト(Example Project)を選んでください。
- ・ ADP example application SAM D21 Xplained Proを閲覧するか検索して、それを選択してください。
- ・望むフォルタを選んでプロジェクトに名前を与え、その後にプロジェクトを作成するためにOKをクリックしてください。

プロジェクトが生成され、その後は単にそれをコンパイルして目的対象基板に書くだけの問題です。

すること: ・ プロジェクトのプロハティを開いてください(解決策エクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロハティ(Properties)を選んでください)。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Build	Configuration: N/A Platform: N/A	
Build Events		_
Toolchain		
Device	Selected debugger/programmer	
Tool	EDBG • ATML2130021800012626 V Interface: SWD V	
Components		
Adversed		



🍌 すること: ・DGI制御盤(DGI Control Panel)でSAM D21 Xplained Proを選んでください。

- ・規約自動検出(Autodetect protocols)枠を選択してください。
 - ・接続(Connect)をクリックしてください。

SAM D21 Xplained Pro	Connec
ATML2130021800012626	Start
	ADP Logging 🕼 Autodetect protocols 🔲 Show Config search path 🔲 Reset N
nterfaces:	

データ可視器で下の複写画面のような何かを見るでしょう。

DGI Control Panel ADP Control Panel - Light Sensor Example for Xulained Pro	~ × ~ ×
Terminal - Status terminal	
It's bright again Entered day model	
	Clean Codd Vin C Hexaderimal Values C Show Timestame C Automatically Sciell to End
Light Sensor Example Dishlooard	
Light Soncer Evample	
Light Sensor Example 5	
Light Sensor Value	
Night Light	
LED Control 20.20 20.25	
Legend Light Stress	
🔚 Edit 🔜 Show Endpoints	
Control Dashboard	- X
rysteresis values. Also sample value.	
Low	
High	
Edit 📳 Show Endpoints	

5.2.2.4. 動かす方法

ADP例用コートがかなり広範囲なため、それを一覧にすることや詳細を全て記述することは意味を成しません。特に、ATSAMD21で必要とされる周辺機能をどう構成設定するかの詳細は省かれます。ADP例計器盤を作成するのに必要とされるADPメッセージは以降の節で詳述されます。コンピュータへ送った各メッセージ後、目的対象(SAM D21デバイス)は次のメッセージを送る前にMSG_CONF_ACKを待ちます。

助言: この例は計器盤の完全な自動構成設定を含みます。けれども、ADPは図表(Graph)、オシロスコープ(Oscilloscope)、または 端末(Terminal)のようなデータ可視器(Data Visualizer)内の様々な単位部へ手動で接続されるように(データ)流れの組を構 成設定するのに使うことができます。ADP制御盤(ADP Control Panel)は現在のADP実体に対して利用可能な吸い込み 口と供給源を示します。これらの吸い込み口と供給源はDGI制御盤(DGI Control Panel)とシリアルポート制御盤(Serial Por t Control Panel)の供給源と吸い込み口と同じように使うことができます。より多くの情報については「Atmelデータ規約」を ご覧ください。

5.2.2.4.1. シリアル インターフェース

ADP例はSAM D21からのADPデータをSAM D21 Xplained Pro基板上の組み込みデバッカ(EDBG)に流すためにSPIインターフェースを使います。EDBGはUSBを渡ってホストコンピュータへデータを流すのにデータ交換器インターフェース(DGI)を使います。目的対象基板がDGI能力を持つデバイスを含まなかった場合、ADPはシリアルインターフェースを渡ってコンピュータへ直接流してしまうことができます。これがその場合だったなら、DGI制御盤(DGI Control Panel)(「データ交換器インターフェース(DGI)」)の代わりにデータ可視器でシリアルポート制御盤(Serial Port Control Panel)(「シリアルポート」)が使われます。

5.2.2.4.2. 初期化

ハート・ウェア構成設定(例えば、シリアルインターフェース初期化、ADCと入出力ポート構成設定)後、応用はADPメッセージ送出開始の準備が整います。最初に送られるメッセージはMSG_REQ_HANDSHAKEです。

表5-5. MSG_REQ_HANDSHAKE				
領域	值	説明		
通票	\$FF	_		
メッセーシ [゙] ID	\$00	-		
データ長	10	-		
規約版番号	\$01	ADP例はADP規約第1版を使います。		
任意選択	\$01	GPIOはこの応用で使用中です。		
通票	{\$58, \$99, \$AB, \$C9, \$0F, \$E2, \$F7, \$AA}	ADP規約を確認するのに使われる通票		

このメッセージはホストがメッセージを受け取る準備が整ったことを示すMSG_RES_HANDSHAKEが受信されるまで繰り返されます。

5.2.2.4.3. ADP制御盤

ADP例は下の複写画面のような何かを見るようにADP制御盤(ADP Control Panel)を構成設定します。



通知: 複写画面の見かけはオペレーティング システムの版と構成設定で変わります。



ADP制御盤(ADP Control Panel)は下表で詳述されるメッセージによって構成設定されます。

表5-6. MSG_CONF_INFO

領域	值	説明
通票	\$FF	_
メッセーシ [゙] ID	\$28	_
データ長	177	_
表題	"Light Sensor Example for Xplained Pro¥0"(Xplained Pro用光感知器例)	_
説明	"This example demonstrates light intensity measurements through the ADC of a Xplained Pro board. You will need the I/O1 Xplained Pro (EXT1)¥0" (この例はXplained Pro基板を通した光量測定を実演します。I/O1 Xplained Pro (EXT1)が必要です。)	応用の短い説明

5.2.2.4.4. 光感知器計器盤

ADP例は下の複写画面のような何かを見るI/O1 Xplained Proの光感知器とLED用の計器盤を構成設定します。

▶ 通知: 複写画面の見かけはオペレーティングシステムの版と構成設定で変わります。



光感知器計器盤は下表で詳述されるメッセージによって構成設定されます。 最初に、計器盤が構成設定されなければなりません。

表5-7. MSG_CONF_DASHBOARD

領域	值。	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2A	-
データ長	38	-
ID	\$0000	計器盤ID
表題	"Light Sensor Example Dashboard¥0"(光感知器例計器盤)	計器盤表題
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFF FFFFFとして送信されます。「メッセージ形式」をご覧ください。)	計器盤の背景色
高さ	300	計器盤の(ピクセルでの)高さ

次に、計器盤に表題(Label)要素が追加されます。

表5-8. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL		
領域	值	説明
通票	\$FF	-
メッセーシ [*] ID	\$2B	-
データ長	47	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0000	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	5	要素のX座標位置。0は計器盤の最左端位置
Y−座標	5	要素のY座標位置。0は計器盤の最上部位置
幅	300	要素の幅 (ピクセル)
高さ	35	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	24	_
属性	\$00	太字体=OFF、斜体=OFF
水平整列	1	中央
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されな ければならないため、\$FFFFFFFFFFFFFとして送 信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255(各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"Light Sensor Example¥0"(光感知器例)	-

表5-9. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ [・] ID	\$2B	-
データ長	45	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0001	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	5	要素のX座標位置。0は計器盤の最左端位置
Y−座標	60	要素のY座標位置。0は計器盤の最上部位置
幅	129	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	-
属性	\$01	太字体=ON、斜体=OFF
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されな ければならないため、\$FFFFFFFFFFFFFとして送 信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255(各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"Light Sensor Value¥0"(光感知器値)	-

表5-10. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL		
領域	値	説明
通票	\$FF	-
メッセーシ [・] ID	\$2B	-
データ長	38	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0002	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	5	要素のX座標位置。0は計器盤の最左端位置
Y−座標	100	要素のY座標位置。0は計器盤の最上部位置
幅	82	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	-
属性	\$01	太字体=ON、斜体=OFF
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されな ければならないため、\$FFFFFFFFFFFFFとして送 信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255(各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"Night Light¥0"(常夜灯)	-

表5-11. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	38	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0003	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	5	要素のX座標位置。0は計器盤の最左端位置
Y−座標	230	要素のY座標位置。0は計器盤の最上部位置
幅	80	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	_
属性	\$01	太字体=ON、斜体=OFF
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFFFFとして送信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255 (各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"LED Control¥0" (LED制御)	-

(データの)流れは光感知器データを受け取るように構成設定することが必要です。

表5-12. MSG_CONF_STREAM

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$20	-
データ長	18	-
ID	\$0001	光感知器データの流れのID
形式	12	流れ形式flot
動作形態	2	目的対象から外へ
状態	0	流れ状態ON
表題	"Light sensor¥0"(光感知器)	データの流れの表題

それと光感知器データを示すための進捗バー(Progress bar)が追加されます。

表5-13. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_PROGRESS

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	29	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0004	この進捗バー要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	140	要素のX座標位置。0は計器盤の最左端位置
Y−座標	60	要素のY座標位置。0は計器盤の最上部位置
幅	145	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$03	ELEMENT_TYPE_PROGRESS
最小値	0	-
最大値	4	-
初期値	0	-
色	\$008000	進捗バーのRGB色

結局、光感知器データの流れが進捗バー(Progress bar)に接続されます。

表5-14. MSG_CONF_ADD_STREAM_TO_ELEMENT

領域	値	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0004	進捗バー要素のID
流れID	\$0001	光感知器データの流れのID

次に、図表(Graph)要素が計器盤に追加されます。

表5-15. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_GRAPH

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	53	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0007	この図表要素の固有ID
表題色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFFFFとして送信されます。「メッセージ形式」をご覧ください。)	表題のRGB色

表5-15 (続き). MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_GRAPH		
領域	値	説明
背景色	\$00000	図表枠のRGB色
図表背景色	\$00000	図表のRGB色
表題文	"Light level¥0" (光レヘ゛ル)	-
作図数	1	-
X最小	0	-
X最大	10	-
Y最小	0	-
Y最大	5	-
動作形態	\$00	マウス作用=OFF。描画部右端に図表を合わせる=OFF。

そして光感知器データの流れは図表(Graph)要素に接続されます。

表	表5-16. MSG_CONF_ADD_STREAM_TO_ELEMENT		
	領域	值	説明
	メッセーシ [゙] ID	\$2C	-
	データ長	6	-
	計器盤ID	\$0000	光感知器計器盤のID
	要素ID	\$0007	図表要素のID
	流れID	\$0001	光感知器データの流れのID

夜間動作を合図するために分離した(データの)流れが構成設定されます。

表5-17. MSG_CONF_STREAM

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$20	-
データ長	16	-
ID	\$0029	夜間動作データの流れのID
形式	2	流れ形式uint_8
動作形態	2	目的対象から外へ
状態	0	流れ状態ON
表題	"Night Mode¥0"(夜間動作)	データの流れの表題

夜間動作合図のために計器盤へ信号(Signal)要素が追加されます。

表5-18. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SIGNAL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	22	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0005	この信号要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	140	要素のX座標位置。0は計器盤の最左端位置
Y−座標	100	要素のY座標位置。0は計器盤の最上部位置
幅	25	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$04	ELEMENT_TYPE_SIGNAL
ON透過率	255(各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
ON色	\$008000	ON状態用RGB色
	\$00000	

表5-18 (続き). MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SIGNAL領域値説明OFF透過率255 (各\$FF文字は\$FFFFとして送信されなけれ
ばならないため、\$FFFFFとして送信されます。「メッ
セージ形式」をご覧ください。)-OFF色\$000000OFF状態用RGB色

そして光感知器データの流れは信号(Signal)要素に接続されます。

表5-19. MSG_CONF_ADD_STREAM_TO_ELEMENT		
領域	值	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0005	信号要素のID
流れID	\$0029	夜間動作データの流れのID

次に、目的対象へ釦(Button)状態を転送するために(目的対象へ)やって来る(データの)流れが構成設定されます。

表5-20. MSG_CONF_STREAM

領域	值	説明
通票	\$FF	-
メッセーシ゛ID	\$20	-
データ長	16	-
ID	\$0030	釦(データの)流れのID
形式	2	流れ形式uint_8
動作形態	0	目的対象へ
状態	0	流れ状態ON
表題	"LED Toggle¥0"(LED交互切り替え)	データの流れの表題

目的対象LEDを交互切り替えするために計器盤へ釦(Button)が追加されます。

表5-21. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_BUTTON

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	26	要素形式に依存
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0006	この釦要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	110	要素のX座標位置。0は計器盤の最左端位置
Y−座標	230	要素のY座標位置。0は計器盤の最上部位置
幅	75	要素の幅 (ピクセル)
高さ	50	要素の高さ(ピクセル)
要素形式	\$01	ELEMENT_TYPE_BUTTON
フォントの大きさ	10	-
釦の文章	"LED Toggle¥0"(LED交互切り替え)	-
交互切り替え釦	\$00=標準釦、\$01=交互切り替え釦	標準釦

そして釦の流れが釦(Botton)要素に接続されます。

表5-22. MSG_CONF_ADD_STREAM_TO_ELEMENT

領域	值	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0000	光感知器計器盤のID
要素ID	\$0006	釦要素のID
流れID	\$0030	釦(データの)流れのID

5.2.2.4.5. 制御計器盤

ADP例はADCを制御するように計器盤を構成設定します。制御計器盤(Control Dashboard)は下の複写画面のような何かを見るでしょう。



▶ 通知: 複写画面の見かけはオペレーティング システムの版と構成設定で変わります。



制御計器盤は下表で詳述されるメッセージによって構成設定されます。 最初に、次のように計器盤自身が構成設定されます。

表5-23. MSG_CONF_DASHBOARD		
領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2A	-
データ長	25	-
ID	\$0001	計器盤ID
表題	"Control Dashboard¥0"(制御計器盤)	計器盤表題
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFF FFFFFとして送信されます。「メッセージ形式」をご覧ください。)	計器盤の背景色
高さ	150	計器盤の(ピクセルでの)高さ

次に、計器盤に少数の表題(Label)要素が追加されます。

表5-24. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ゛ID	\$2B	-
データ長	45	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$0008	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	5	要素のX座標位置。0は計器盤の最左端位置
Y−座標	20	要素のY座標位置。0は計器盤の最上部位置
幅	128	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	-
属性	\$01	太字体=ON、斜体=OFF
水平整列	0	左
垂直整列	1	中央
次百へ続く		

表5-24 (続き). MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL		
領域	值	説明
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されな ければならないため、\$FFFFFFFFFFFFとして送 信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255 (各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	_
前景色	\$00000	前景のRGB色
表題文	"Hysteresis Values¥0"(ヒステリシス値)	-

表5-25. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	31	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000A	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	25	要素のX座標位置。0は計器盤の最左端位置
Y−座標	100	要素のY座標位置。0は計器盤の最上部位置
幅	30	要素の幅 (ピクセル)
高さ	25	要素の高さ (ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	-
属性	\$02	太字体=OFF、斜体=ON
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFFFFFとして送信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255 (各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"High¥0"(高)	-

表5-26. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	30	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$0009	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	25	要素のX座標位置。0は計器盤の最左端位置
Y−座標	60	要素のY座標位置。0は計器盤の最上部位置
幅	30	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL

表5-26 (続き). MSG_CONF_DASHBOARD_ELEMENT, ELEMENT_TYPE_LABEL

領域	値	説明
フォントの大きさ	14	-
属性	\$02	太字体=OFF、斜体=ON
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFFFとして送信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255(各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	_
前景色	\$00000	前景のRGB色
表題文	"Low¥0"(低)	-

表5-27. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_LABEL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	44	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000B	この表題要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	350	要素のX座標位置。0は計器盤の最左端位置
Y−座標	20	要素のY座標位置。0は計器盤の最上部位置
幅	130	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$00	ELEMENT_TYPE_LABEL
フォントの大きさ	14	-
属性	\$01	太字体=ON、斜体=OFF
水平整列	0	左
垂直整列	1	中央
背景透過率	0	-
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFFFFとして送信されます。「メッセージ形式」をご覧ください。)	背景のRGB色
前景色透過率	255 (各\$FF文字は\$FFFFとして送信されなけれ ばならないため、\$FFFFとして送信されます。「メッ セージ形式」をご覧ください。)	-
前景色	\$00000	前景のRGB色
表題文	"ADC Sample Value¥0" (ADC採取値)	-

次に、ヒステリシスの高値を設定するために(データの)流れが構成設定されます。

表5-28. MSG_CONF_STREAM 説明 値 領域 \$FF 通票 _ メッセーシ[゙]ID \$20 _ データ長 16 ヒステリシス高値データの流れのID ID \$0010 形式 4 流れ形式uint_16 目的対象へ 動作形態 0

表5-28 (続き). MSG_CONF_STREAM

領域	值	説明
状態	0	流れ状態ON
表題	"Hyst. High¥0"(ヒステリシス高)	データの流れの表題

計器盤からヒステリシスの高値を調整することができるように摺動子(Slider)が追加されます。

表5-29. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SLIDER

領域	値	説明
通票	\$FF	-
メッセーシ゛ID	\$2B	-
データ長	26	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000D	この摺動子要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	75	要素のX座標位置。0は計器盤の最左端位置
Y−座標	100	要素のY座標位置。0は計器盤の最上部位置
幅	156	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$02	ELEMENT_TYPE_SLIDER
最小值	2500	-
最大値	4000	-
初期値	3000	-

そしてヒステリシス高値が摺動子(Slider)要素に接続されます。

表5-30. MSG_CONF_ADD_STREAM_TO_ELEMENT		
領域	值	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000D	ヒステリシス高摺動子要素のID
流れID	\$0010	ヒステリシス高データの流れのID

ヒステリシス低値用の(データの)流れが作成され、上のヒステリシス高値用と同じ方法で摺動子(Slider)要素に追加されます。

表5-31. MSG_CONF_STREAM

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$20	-
データ長	15	-
ID	\$0011	ヒステリシス低値データの流れのID
形式	4	流れ形式uint_16
動作形態	0	目的対象へ
状態	0	流れ状態ON
表題	"Hyst. Low¥0" (ヒステリシス低)	データの流れの表題

表5-32. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SLIDER

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	26	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000C	この摺動子要素の固有ID
計 新盛 ID 要素ID	\$0001 \$000C	前御計希盤のID この摺動子要素の固有ID

表5-32 (続き). MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SLIDER

領域	値	説明
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	75	要素のX座標位置。0は計器盤の最左端位置
Y−座標	60	要素のY座標位置。0は計器盤の最上部位置
幅	156	要素の幅 (ピクセル)
高さ	25	要素の高さ(ピクセル)
要素形式	\$02	ELEMENT_TYPE_SLIDER
最小値	1000	-
最大値	2400	-
初期値	2000	-

表5-33. MSG_CONF_ADD_STREAM_TO_ELEMENT

領域	值	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000C	ヒステリシス低摺動子要素のID
流 れ ID	\$0011	ヒステリシス低データの流れのID

セグメント表示器(Segment display)へ供給される光感知器ADC値用の分離した(データの)流れが構成設定されます。

表5-34. MSG_CONF_STREAM

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$20	-
データ長	22	-
ID	\$0002	ADC値データの流れのID
形式	4	流れ形式uint_16
動作形態	2	目的対象から外へ
状態	0	流れ状態ON
表題	"Light Sensor ADC¥0"(光感知器ADC)	データの流れの表題

次に、4つの桁を持つセグメント表示器(Segment display)が計器盤に追加されます。

表5-35. MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SEGMENT

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$2B	-
データ長	20	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000D	このセグメント表示器要素の固有ID
Z−指標	0	順番指標、0は背面から最も遠くに配置
X−座標	500	要素のX座標位置。0は計器盤の最左端位置
Y−座標	20	要素のY座標位置。0は計器盤の最上部位置
幅	150	要素の幅 (ピクセル)
高さ	50	要素の高さ (ピクセル)
要素形式	\$05	ELEMENT_TYPE_SEGMENT
数	4	4桁
基数	10	普通の10進数
透過率	\$FF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFとして送信されます。「メッセージ形式」をご覧ください。)	0~255

表5-35 (続き). MSG_CONF_DASHBOARD_ELEMENT,ELEMENT_TYPE_SEGMENT領域値説明色\$FFF0000 (各\$FF文字は\$FFFFとして送信されな
ければならないため、\$FFFF0000として送信され
ます。「メッセージ形式」をご覧ください。)

そしてADC採取値の(データの)流れがセグメント表示器(Segment display)に接続されます。

表5-36. MSG_CONF_ADD_STREAM_TO_ELEMENT

領域	值	説明
メッセーシ [゙] ID	\$2C	-
データ長	6	-
計器盤ID	\$0001	制御計器盤のID
要素ID	\$000D	セグメント表示器要素のID
流れID	\$0002	ADC採取データの流れのID

5.2.2.4.6. 端末

ADP例は下の複写画面のような何かを見る端末単位部を構成設定します。



▶ 通知: 複写画面の見かけはオペレーティング システムの版と構成設定で変わります。

Terminal - Status terminal	**
It's bright again Entered day mode:	
6 M	Clear Add \r\n 🗐 Hexadecimal Values 📑 Show Timestamp 🐼 Automatically Scroll to End

端末単位部は下表で詳述されるメッセージによって構成設定されます。

最初に、(データの)流れが端末データをホストコンピュータへ送るように構成設定されます。

表5-37. MSG_CONF_STREAM		
領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$20	-
データ長	21	-
ID	\$0000	端末データの流れのID
形式	2	流れ形式uint_8
動作形態	2	出て行くデータの流れ(目的対象から外へ)
状態	0	流れ状態ON
表題	"Status messages¥0"(状態メッセージ)	データの流れの表題

次に、端末自身が構成設定されます。

表5-38. MSG_CONF_TERMINAL

領域	值	説明
通票	\$FF	-
メッセーシ [・] ID	\$26	-
データ長	26 –	
ID	\$0000 端末のID	
表題	"Status terminal¥0"(状態端末)	端末の表題
幅	80	文字数での幅
高さ	50	行数での高さ

表5-38 (続き). MSG_CONF_TERMINAL

領域	值	説明	
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されな ければならないため、\$FFFFFFFFFFFFFとして送 信されます。「メッセージ形式」をご覧ください。)	RGB背景色	
前景色	\$008000	RGB前景色	

最後に、データの流れが端末単位部に接続されます。

表5-39. MSG_CONF_ADD_TERMINAL

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$27	-
データ長	27	-
端末ID	\$0000	端末のID
流 れ ID	\$0000	(データの)流れID
動作形態	\$FF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFとして送信されます。「メッセージ形式」をご覧ください。)	やって来る文章で暗黙の改行=ON
文章色	\$000000 受信した文章の流れのRGB色	
標識文章	"Status messages¥0"(状態メッセージ)	-
標識文章色	\$00000	標識文章のRGB色

5.2.2.4.7. データ送信

端末単位部と2つの計器盤が前の部分で記述されるように構成設定されてしまうと、ADP例は構成設定した流れに従って継続的にホ ストコンピュータへデータを送り、ホストコンピュータからデータを受け取ります。下はATSAMD21目的対象からホストコンピュータへ送られるデータメッ セージの例です。

表5-40. 光感知器ADCデータの流れ 説明 領域 値 通票 \$FF メッセーシ[`]ID \$40 データ長 6 データ流れ数 1 流れID \$0002 ADC値データの流れのID バ仆数 2 データ流れからのバイト数 流れXデータ採取 634 (\$027A) データ流れのデータ (uint_16)

表5-41.夜間動作データの流れ

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$40	-
データ長	5	-
データ流れ数	1	-
流れID	\$0029	夜間動作データの流れのID
バ仆数	1	データ流れからのバイト数
流れXデータ採取	\$01 (明るい光、日中動作)	データ流れのデータ (uint_8)

夜間動作が変わると、例は別のMSG_CONG_TERMINALを送ることによって端末単位部の背景色も変えます。

表5-42. 端末背景色を白に更新するためのMSG_CONF_TERMINAL

領域	値	説明
通票	\$FF	-
メッセーシ゛ID	\$26	-
データ長	26	-
次百~ 娃/		

次貝へ続く

表5-42 (続き).端末背景色を白に更新するためのMSG_CONF_TERMINAL

A	н. Н	
領域	但	[
ID	\$0000	端末のID
表題	"Status terminal¥0"(状態端末)	端末の表題
幅	80	文字数での幅
高さ	50	行数での高さ
背景色	\$FFFFFF(各\$FF文字は\$FFFFとして送信されなければならないため、\$FFFFFFFFFFFとして送RGB背景色信されます。「メッセージ形式」をご覧ください。)	
前景色	\$008000	RGB前景色

表5-43. 状態メッセージの流れ

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$40	-
データ長	44	-
データ流れ数	1	-
流れID	\$0000	状態メッセージデータの流れのID
バ仆数	40	データ流れからのバイト数
流れXデータ採取	"It's bright again Entered day mode!¥r¥n"(再び明るくなり、日中動作移行) {\$49, \$74, \$27, \$73, \$20, \$62, \$72, \$69, \$67, \$68, \$74, \$20, \$61, \$67, \$61, \$69, \$6E, \$2E, \$2E, \$2E, \$20, \$45, \$6E, \$74, \$65, \$72, \$65, \$64, \$20, \$64, \$61, \$79, \$20, \$6D, \$6F, \$64, \$65, \$21, \$0D, \$0A}	データ流れのデータ (uint_8)

コンピュータから目的対象への様々なデータの流れ用データ メッセージの例は下表で見つけることができます。

表5-44. ヒステリシス低値流れデータ メッセージ			
領域	值	説明	
通票	\$FF	-	
メッセーシ [゙] ID	\$14	MSG_RES_DATA	
データ長	5	-	
流れID	\$0011	ヒステリシス低値データの流れのID	
送出バイト数	2	データ本体内のバ仆数	
データバイト	\$07C6	データ (uint_16)	

表5-45. ヒステリシス高値流れデータ メッセージ

領域	值	説明
通票	\$FF	-
メッセーシ [゙] ID	\$14	MSG_RES_DATA
データ長	5	-
流れID	\$0002 ヒステリシス高値データの流れのID	
送出バイ教	<u>2</u> データ本体内のバイト数	
データハイト	\$0BB7	データ (uint_16)

表5-46. ヒステリシス高値流れデータ メッセージ

領域	値	説明
通票	\$FF	-
メッセーシ [゙] ID	\$14	MSG_RES_DATA
データ長	4 –	
流れID	\$0030 LED交互切り替えデータの流れのID	
送出バイ教	1 データ本体内のバイト数	
データハイト	\$00	データ (uint_8)

5.2.3. メッセージの流れ

目的対象はシステムに於ける主装置、一方でホストコンピュータは従装置です。目的対象が通信を始め、コンピュータが応答します。けれども、コンピュータがコンピュータで生成されると直ぐにデータを目的対象MCUへ送信します。

目的対象とコンピュータ間で何かのデータが交換され得るのに先立ち、ハンドシェークメッセージを使って接続が確立されなければなりません。

5.2.4. メッセーシ 形式

ADP規約は通信の両方向と全てのメッセージ型に対して共通メッセージ形式を使います。

表5-47. ADPメッセージ 形式				
領域	大きさ	值	説明	
通票	1/ㆍイト	\$FF	ADPデータ用開始通票	
メッセーシ [゙] ID	1/ㆍイト	\$00~\$FE	送られるメッセージの型を識別	
データ長	2 ハ・イト	\$0000~\$FFFF	データパケットの長さ(バイト数)	
データ	Nハッイト	\sim	メッセージのデータ内容	

通票

\$FF以外の値(\$FFは有効なメッセージIDではありません。)が後続する\$FFがメッセージの開始を示すのに使われます。これは各メッセージ 間で\$FFが送られなければならないことを意味します。

データまたはデータ長の一部として\$FFの値が送信される場合、それの後に新しい\$FFが挿入されるべきです。メッセージ受信時、2つの \$FFは単一の\$FFとして復号されるべきです。この余分な\$FFハイトはデータ長領域に寄与しません。例えば、\$FFFFFFの値を持つ色 領域は\$FFFFFFFFFFFFとして送信されなければなりませんが、データ長には3ハイトが寄与するだけです。

\$FFの値はメッセージIDとして使うことが許されません。SPI上のデータに対するポーリング時、予期せぬ命令を起動しないように偽装文字として\$FF通票が使われなければなりません。

エンディアン

全てのメッセージデータはリトル エンディアンを使って順番化されます。

5.2.5. メッセージ型

要求メッセージ、構成設定メッセージ、データメッセージの3つの主なメッセージ型があります。

5.2.5.1. 要求パッセージ

要求メッセージはホスト PCからの情報/状態を要求するために目的対象によって使われます。これらのメッセージは初めにMSG_REQを付けられます。PCは常に初めにMSG_RESを付けた対応する応用メッセージで応答すべきです。

表5-48. 要求メッセージと応答

メッセージ"型	ID	説明
MSG_REQ_HANDSHAKE	\$00	ハント・シェーク要求
MSG_RES_HANDSHAKE	\$10	ハント・シェークに対する応答
MSG_REQ_STATUS	\$02	PCからの状態を要求
MSG_RES_STATUS	\$12	状態に対する応答
MSG_RES_DATA	\$14	PCから目的対象への生データ

5.2.5.2. 構成設定メッセージ

PCへ構成設定を送るために目的対象によって使われます。これらのメッセージは初めにMSG_CONFを付けられます。PCは応答メッセー ジ(MSG_ACK)でこれらのメッセージに応答すべきです。

表5-49.構成設定メッセージ、目的対象からPCへ

メッセージ型	ID	説明
MSG_CONF_STREAM	\$20	新しいデータの流れを作成
(予約)	\$21	(将来の使用のため予約)
MSG_CONF_GRAPH	\$22	新しい図表作成または既存のものを再構成設定
MSG_CONF_ADD_STREAM_TO_AXIS	\$23	データの流れを図表の軸に追加
MSG_CONF_CURSOR_TO_GRAPH	\$24	パラメータ カーソルを図表に追加
(予約)	\$25	(将来の使用のため予約)
MSG_CONF_TERMINAL	\$26	新しい端末作成または既存のものを再構成設定
MSG_CONF_ADD_TO_TERMINAL	\$27	データの流れを端末に追加
<u></u> 本百~ 結/		

伙貝へ続く

表5-49 (続き).構成設定メッセージ、目的対象からPCへ

メッセージ型	ID	説明
MSG_CONF_INFO	\$28	応用についての情報
MSG_CONF_AXIS	\$29	新しい軸を作成または既存のものを再構成設定
MSG_CONF_DASHBOARD	\$2A	計器盤容器を追加
MSG_CONF_DASHBOARD_ELEMENT	\$2B	計器盤容器に要素を追加
MSG_CONF_ADD_STREAM_TO_ELEMENT	\$2C	構成設定済みデータの流れを構成設定済み計器盤要素を結合

表5-50.構成設定メッセージ、PCから目的対象へ

メッセージ型	ID	説明
MSG_CONF_ACK	\$30	最後に受信した構成設定メッセージの状態

5.2.5.3. データ メッセーシ

PCへデータを送るために目的対象によって使われます。初めにMSG_DATAを付けられます。これらのメッセージもまた応答されるべきではありません。

表5-51. データ メッセーシ

メッセージ型	ID	説明
MSG_DATA_STREAM	\$40	1つ以上のデータの流れからデータを送出

5.2.5.4. 要求メッセージ詳細

5.2.5.4.1. MSG_REQ_HANDSHAKE

目的対象とPC間でどんなデータも交換し得るのに先立ってハンドシェーックメッセージを使って接続が確立されなければなりません。

● 重要:自動検出中にデータ可視器(Data Visualizer)によってハント・シェークメッセージが検出されるのを確実にするため、ハント・シェーク
 メッセージは秒毎に最低2度送られるべきです。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$00	-
データ長	2 バイト	10	-
規約版番号	1시간	\$01~\$FF	目的対象上の規約の版番号
任意選択	1시간	\$xx	(将来の使用のため予約)
通票	8ハ・イト	{\$58, \$99, \$AB, \$C9, \$0F, \$E2, \$F7, \$AA}	ADP規約を確認するのに使われる通票

5.2.5.4.2. MSG_RES_HANDSHAKE

PCは目的対象からのハント・シェーク要求にこのパケットで応答すべきです。PCは常に目的対象のハント・シェーク要求メッセーシ・で宣言された規約版を使って目的対象と通信すべきです。PCが或る理由のためにこれを行うことができない場合、ハント・シェーク要求は拒否されなければなりません。

領域	大きさ	值	説明
メッセーシ [゙] ID	1バイト	\$10	-
データ長	2 パイト	1	-
ハント・シェーク状態	1/차기	\$00: ハンドシェーク受け入れ \$01: ハンドシェーク拒否、無効規約版 \$02: ハンドシェーク拒否、他の理由	-

5.2.5.4.3. MSG_REQ_STATUS

PCからの状態を要求するために目的対象によって使われるメッセージ。これは新しい構成設定メッセージが送られる毎に状態を問うことは最良の慣行です。生データを送出すると同時に時折、状態を問うことは最良の慣行です。

領域	大きさ	値	説明
メッセーシ [・] ID	1.1パイト	\$02	-
データ長	2/ㆍイト	0	-

このメッセージはデータ領域を持ちません。

5.2.5.4.4. MSG_RES_STATUS

PCから目的対象への状態メッセーシ、一旦状態が目的対象から要求されてこのパケットが送られると、全ての状態フラグがPC上で解消されるべきです。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$12	-
データ長	2 パイト	2	-
状態符号	1バイト	\$0000	(将来の使用のため予約)

5.2.5.4.5. MSG_RES_DATA

PCから目的対象へのデータパケット。このメッセージはPCから非同期的に(即ち、目的対象がそれを要求していなくても)来得ます。

領域	大きさ	值	説明	
メッセーシ [゙] ID	1バイト	\$14	-	
データ長	2バイト	3+データ バイト数	-	
流れID	2 パイト	\$xxxx	(データの)流れのID	
送出バイト	1ハブイト	\$xx	データ本体のハイト数。目的対象が未知のデータの流れからの要求 されたデータしか持たない場合、またはデータの流れが送るデータを 持たない場合、この領域は0に設定されるべきで、適切な状態フラ グが設定されるべきです。	
データバイト	Nバイト	~	データ	

5.2.5.5.構成設定メッセージ詳細

5.2.5.5.1. MSG_CONF_STREAM

新しいデータの流れを作成するのに使われます。流れの形式は事象、文章、データのどれかで有り得ます。各流れは固有のIDを与えられなければなりません。

領域	大きさ	值	説明
メッセーシ [・] ID	1 バイト	\$20	-
データ長	2バイト	5+表題長(N)+パラメータ長(M)	-
ID	2バイト	$0000 \sim FFFF$	(データの)流れのID
型	1 バイト	STREAM_DATA_TYPE	流れ形式
動作形態	1.小子	0=やって来る(標準) 1=やって来る(単一値) 2=出て行く	流れ動作形態/方向、方向は目的対象から見て定義されます。
状態	1バイト	0=ON, 1=OFF	流れの状態
表題	Nバイト	ヌル終端文字列	データの流れの表題
パラメータ	Mバイト	バイト配列	流れ形式を指定するパラメータ

表5-52. STREAM_DATA_TYPE

形式符号	データ型	パラメータ	データの大きさ	形式符号	データ型	パラメータ	データの大きさ
0	EVENT	なし	0ハ*イト	8	XY_8	なし	2 ハ・イト
1	STRING	なし	Nバイト	9	XY_16	なし	4バイト
2	UINT_8	なし	1 ハ・イト	10	XY_32	なし	8시 イト
3	INT_8	なし	1 パイト	11	BOOL	なし	1시 사
4	UINT_16	なし	2 パイト	12	Float	なし	4ハ [*] イト
5	INT_16	なし	2 パイト	13	Double	なし	8시 イト
6	UINT_32	なし	4バイト			基準データ型(1バイト、例えば、	基準データ型
7	INT_32	なし	4ハ*イト	20	Grid(格子)	UINT_32に対しては 6) 換えの短(1 、(1)、換えの深さ(1 、(1))	×幅×深さ
						俗丁9/16(1/17)、俗丁9/休さ(1/17)	까지경역

XYデータ型はXとYの座標の組み合わせです。形式がXY_8なら、データは1バイトのX座標と1バイトのY座標を含みます。XY_16に対して データは2バイトのX座標と2バイトのY座標を含み、XY_32の各座標は4バイト長です。

5.2.5.5.2. MSG_CONF_GRAPH

新規作成、または既存の図表表示部の再構成設定に使われます。図表表示部は固有IDを必要とします。範囲、表題、背景色用の 値を設定することもできます。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$22	-
データ長	2 パイト	23+表題長(N)+X表題長(M)	-
ID	2/ㆍイト	\$0000~\$FFFF	(新しい)図表のID
表題	Nバイト	ヌル終端文字列	図表の表題
X最小	4ハ・イト	-	X範囲最小値
X最大	4ハ [*] イト	-	X範囲最大値
X表題	Mバイト	ヌル終端文字列	Xの表題
X尺度分子	4ハ [*] イト	-	X範囲尺度値。自動範囲を許可するには0に設定してください。
X尺度分母	4ハ [*] イト	-	X範囲尺度値。自動範囲を許可するには0に設定してください。
尺度調整動作	1バイト	0=尺度調整OFF、1=自動調整	垂直尺度調整
背景色	3 시 イト	\$RRGGBB	RGB背景色
スクロール動作	1시 イト	0=スクロールなし、1=段階 2=スクロール、3=円形/掃引	水平スクロール

5.2.5.5.3. MSG_CONF_ADD_STREAM_TO_AXIS

指定した図表表示部に(データの)流れを追加するのに使われます。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$23	-
データ長	2 パイト	32	-
図表ID	2 パイト	$0000 \sim FFFF$	図表のID
軸ID	2 パイト	\$0000~\$FFFF	軸のID
流れID	2 パイト	$0000 \sim FFFF$	(データの)流れのID
採取速度分子	4ハ [*] イト	-	流れの採取速度。不適用の場合は0に設定してください。
採取速度分母	4ハ [*] イト	-	流れの採取速度。不適用の場合は0に設定してください。
Y尺度分子	4ハ [*] イト	-	Y範囲尺度値。自動範囲を許可するには0に設定してください。
Y尺度分母	4ハ [*] イト	-	Y範囲尺度値。自動範囲を許可するには0に設定してください。
Y変位	4ハ [*] イト	-	値の変位(オフセット)
透過率	1.1パイト	0~255	透過率を調整
動作形態	1 小 イト	図表用: ビット0=線ON/OFF ビット1=点ON/OFF 文章用: 0=フラグ、1=文章	-
太さ	1バイト	0~255	線の太さ
色	3/ㆍイト	\$RRGGBB	線のRGB色

5.2.5.5.4. MSG_CONF_CURSOR_TO_GRAPH

図表にパラメータカーソルを追加するのに使われます。

領域	大きさ	値	説明	
メッセーシ [゙] ID	1バイト	\$24	-	
データ長	2バイト	35+表題長(N)	-	
流れID	2バイト	$0000 \sim FFFF$	(データの)流れのID	
図表ID	2 パイト	$0000 \sim FFFF$	図表のID	
軸ID	2 パイト	\$0000~\$FFFF	軸のID	
表題	Nバイト	ヌル終端文字列	カーソルの表題	
太さ	1バイト	0~255	線の太さ	
色	3 · ' イト	\$RRGGBB	カーソルのRGB色	
次百へ続く				
領域	大きさ	值	説明	
------	--------------------	------	-----------------	--
初期値	4ハ・イト	-	カーソルの開始点	
最小	4バイト	-	許容最小値	
最大	4ハ・イト	-	許容最大値	
尺度分子	4ハ [*] イト	-	尺度調整値の分子	
尺度分母	4バイト	-	尺度調整値の分母	
尺度変位	4バイト	-	尺度調整値の変位(オフセット)	
様式	1バイト	\$xx	(将来の使用のため予約)	

5.2.5.5.5. MSG_CONF_TERMINAL

新規作成、または既存の端末の再構成設定に使われます。端末は固有IDを必要とします。表題と背景色用の値も設定しなければなりません。

領域	大きさ	值	説明
メッセーシ [・] ID	1 バイト	\$26	-
データ長	2バイト	10+表題長(N)	-
ID	2 パイト	\$0000~\$FFFF	端末のID
表題	Nバイト	ヌル終端文字列	端末の表題
幅	1バイト	0~255	幅の文字数
高さ	1バイト	0~255	高さの行数
背景色	3ハ [*] イト	\$RRGGBB	RGB背景色
前景色	3ハ [*] イト	\$RRGGBB	RGB前景色

5.2.5.5.6. MSG_CONF_ADD_TO_TERMINAL

指定した端末に(データの)流れを追加するのに使われます。

領域	大きさ	値	説明		
メッセーシ [゙] ID	1バイト	\$27	-		
データ長	2 パイト	11+標識長(N)	-		
端末ID	2 パイト	\$0000~\$FFFF	端末のID		
流れID	2バイト	$0000 \sim FFFF$	(データの)流れのID		
動作形態	1バイト	ʻxxxxnxxxx'	n=やって来る文章で暗黙の改行		
文章色	3 バイト	\$RRGGBB	受信した流れの文章のRGB色		
標識文章	Nバイト	ヌル終端文字列	説明的な標識		
標識文章色	3バイト	\$RRGGBB	標識文章のRGB色		

5.2.5.5.7. MSG_CONF_INFO

応用についての情報を送るのに使われます。例えば、例応用を説明する文章文字列。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$28	-
データ長	2バイト	表題長(N)+説明長(M)	-
表題	Nバイト	ヌル終端文字列	応用の表題
説明	Mバイト	ヌル終端文字列	応用の短い説明

5.2.5.5.8. MSG_CONF_AXIS

新規作成、または図表表示部の既存の軸の再構成設定に使われます。軸は固有IDを必要とします。範囲、表題、軸色用の値も設定しなければなりません。

領域	大きさ	値	説明		
メッセーシ [・] ID	1.1パイト	\$29	-		
データ長	2 パイト	24+表題長(N)	-		

次貝へ続く

領域	大きさ	值	説明		
軸ID	2バイト	\$0000~\$FFFF	(新しい)軸のID		
図表ID	2 パイト	\$0000~\$FFFF	軸を含む図表のID		
表題	Nバイト	ヌル終端文字列	軸の表題		
Y最小	4ハ・イト	-	Y範囲最小値		
Y最大	4.小子ト	-	Y範囲最大値		
Y尺度分子	4.小子ト	-	Y範囲尺度値。自動範囲を許可するには0に設定してください。		
Y尺度分母	4.小子ト	- Y範囲尺度値。自動範囲を許可するには0に設定してくださ			
尺度動作	1/ㆍ/ト	0=尺度調整OFF、1=自動調整	至直尺度調整		
軸色	3/ㆍイト	\$RRGGBB	軸のRGB色		

5.2.5.5.9. MSG_CONF_DASHBOARD

計器盤の要素を配置することができる計器盤容器を追加します。

領域	大きさ	值	説明	
メッセーシ [゙] ID	1バイト	\$2A	-	
データ長	2バイト	7+表題長(N)	-	
ID	2 パイト	\$0000~\$FFFF	計器盤のID	
表題	Nバイト	ヌル終端文字列	計器盤の表題	
背景色	3ハ [*] イト	\$RRGGBB	計器盤の背景色	
高さ	2/ 기가	_	計器盤の(ピクセルでの)高さ	

5.2.5.5.10. MSG_CONF_DASHBOARD_ELEMENT

計器盤の要素を構成設定して指定した計器盤に追加します。下表は計器盤の全ての要素形式に対して共通する領域を示します。 要素形式に応じて追加領域が追加されなければなりません。以降をご覧ください。

領域	大きさ	値	説明		
メッセーシ [゙] ID	1バイト	\$2B	-		
データ長	2 パイト	14+要素特有長	要素形式に依存		
計器盤ID	2バイト	$0000 \sim FFFF$	要素を追加する計器盤のID		
要素ID	2 バイト	$0000 \sim FFFF$	要素の固有ID		
Z−指標	1バイト	_	順番指標、0は背面から最も遠くに配置		
X−座標	2 パイト	ピクセルでの座標値	要素のX座標位置。0は計器盤の最左端位置		
Y−座標	2 パイト	ピクセルでの座標値	要素のY座標位置。0は計器盤の最上部位置		
幅	2 パイト	ピクセルでの幅	要素の幅		
高さ	2バイト	ピクセルでの高さ	要素の高さ		
要素形式	1バイト	ELEMENT_TYPE	下の各要素形式をご覧ください。		

5.2.5.5.10.1. ELEMENT_TYPE_LABEL

下表はELMENT_TYPE_LABEL特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-53. ELEMENT_TYPE_LABEL特有ハラメータ				
パラメータ	値			
ELEMENT_TYPE	\$00			
ELEMENT_TYPE_LABEL使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	26+既定文章長(N)			

表5-54. MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	値	説明
フォントの大きさ	1.1パイト	1~255	-
属性	1.1パイト	\$0000~\$FFFF	-
水平整列	1.1パイト	0=左、1=中央、2=右	-
垂直整列	1.1パイト	0=上、1=中央、2=下	-
次百へ 続く			

次負へ続く

表5-54 (続き). MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域 領域 大きさ 説明 値 背景透過率 1バイト $0 \sim 255$ 背景色 3バイト \$RRGGBB 背景のRGB色 前景透過率 1バイト $0 \sim 255$ 前景色 3ハイト \$RRGGBB 前景のRGB色 表題文章 Nハイト ヌル終端文字列、最大100パイト

例

下の画像は画像の右の表で与えられた対応するパラメータを持つ2つの表題の例を示します。

• 通知: 要素の見かけはオペレーティングシステムの版と構成設定で変わります。

Dashboard I/O	A X.	表5-55. 表題1と表題	2	
		領域	表題1の値	表題2の値
	\odot	Z−指標	0	1
Label1		X−座標	30	90
	neg	Y−座標	30	70
Label2	র	幅	200	75
		高さ	100	25
		要素形式	\$00	\$00
		フォントの大きさ	16	12
		属性	\$01	\$00
		水平整列	0	1
		垂直整列	0	1
		背景透過率	255	100
		背景色	\$64FF64	\$646464
		前景透過率	255	255
		前景色	\$000000	\$7C0000
		表題文章	"Label1¥0"	"Label2¥0"
		(<mark>訳注</mark>)原書の表5-55.	と表5-56.は表5-	・55.に纏めました

5.2.5.5.10.2. ELEMENT_TYPE_BUTTON

下表はELMENT_TYPE_BUTTON特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-57. ELEMENT_TYPE_BUTTON特有パラメータ				
N [°] ラメータ	値			
ELEMENT_TYPE	\$01			
ELEMENT_TYPE_BUTTON使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	16+釦文章長(N)			

表5-58. ELEMENT_TYPE_BUTTON用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
フォントの大きさ	1バイト	0~255	-
釦文章	Nバイト	ヌル終端文字列、最大20バイト	-
交互切り替え釦	1/ㆍ/ト	0=標準釦、1=交互切り替え釦	-

例

次の画像は1つの標準釦と1つの交互切り替え釦の2つの釦の例を示します。例に対する要素データ領域は画像の右の表で示されます。



通知:要素の見かけはオペレーティングシステムの版と構成設定で変わります。

Dashboard I/O
Button
🗖 Edit 🔲 Show Endpoints

表5-59. 標準釦と交互切り替え釦

れている。除十五に入上が、日に出				
領域	標準釦の値	切替釦の値		
Z−指標	0	0		
X−座標	30	30		
Y−座標	30	70		
幅	75	120		
高さ	25	25		
要素形式	\$01	\$01		
フォントの大きさ	12	12		
釦文章	"Button¥0"	"ON/OFF¥0"		
交互切り替え釦	\$00	\$01		

(訳注)原書の表5-59.と表5-60.は表5-59.に纏めました。

5.2.5.5.10.3. ELEMENT_TYPE_SLIDER

下表はELMENT_TYPE_SLIDER特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-61. ELEMENT_TYPE_SLIDER特有パラメータ	
パラメータ	值
ELEMENT_TYPE	\$02
ELEMENT_TYPE_SLIDER使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	26

表5-62. ELEMENT_TYPE_SLIDER用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
最小値	4ハ [*] イト	-	-
最大値	4バイト	-	-
初期値	4ハイト	_	-

例

下の画像は0~50の範囲と20の初期値を持つ摺動子要素の例を示します。対応する要素データ領域は画像の右の表で与えられます。



通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。

Dashboard I/O	^ x
Edit Chow Endpoints	

表5-63. 摺動子				
領域	值			
Z−指標	0			
X−座標	20			
Y−座標	20			
幅	200			
高さ	30			
要素形式	\$02			
最小值	0			
最大値	50			
初期値	20			

5.2.5.5.10.4. ELEMENT_TYPE_PROGRESS

下表はELMENT_TYPE_PROGRESS特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-64. ELEMENT_TYPE_PROGRESS特有パラメータ	
N [°] ラメータ	値
ELEMENT_TYPE	\$03
ELEMENT_TYPE_PROGRESS使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	29

表5-65. ELEMENT_TYPE_PROGRESS用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	値	説明
最小値	4시 イト	-	-
最大値	4ハ [*] イト	-	-
初期値	4バイト	-	-
色	3バイト	\$RRGGBB	進捗バーのRGB色

例

下の画像は進捗バー要素の例を示します。対応する要素データ領域は画像の右の表で与えられます。



通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。



表5-66. 進捗バー				
領域	値			
Z−指標	0			
X−座標	10			
Y-座標	10			
幅	100			
高さ	30			
要素形式	\$03			
最小値	0			
最大値	100			
初期値	50			
色	\$01D328			

5.2.5.5.10.5. ELEMENT_TYPE_SIGNAL

下表はELMENT_TYPE_SIGNAL特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-67. ELEMENT_TYPE_SIGNAL特有パラメータ	
パ゚ラメータ	値
ELEMENT_TYPE	\$04
ELEMENT_TYPE_SIGNAL使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	22

表5-68. ELEMENT_TYPE_SIGNAL用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
ON透過率	1バイト	$0\sim\!255$	-
ON色	3バイト	\$RRGGBB	ON状態のRGB色
OFF透過率	1バイト	$0 \sim 255$	-
OFF色	3/ 기가	\$RRGGBB	OFF状態のRGB色

例

次の画像は信号要素の例を示します。対応する要素データ領域は画像の右の表で与えられます。



通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。

Dashboard I/O	^ X
🔲 Edit 🔲 Show Endpoints	

領域	値
Z−指標	0
X−座標	20
Y−座標	20
幅	50
高さ	50
要素形式	\$04
ON透過率	255
ON色	\$00FF00
OFF透過率	255
OFF色	\$FF0000

5.2.5.5.10.6. ELEMENT_TYPE_SEGMENT

下表はELMENT_TYPE_SEGMENT特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。

表5-70. ELEMENT_TYPE_SEGMENT特有パラメータ	
パ [°] ラメータ	値
ELEMENT_TYPE	
ELEMENT_TYPE_SEGMENT使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	20

表5-71. ELEMENT_TYPE_SEGMENT用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	値	説明
桁数	1시간	1~20	表示器数(桁数)
基数	1バイト	2~16	基数
透過率	1バイト	0~255	-
色	3バイト	\$RRGGBB	表示のRGB色

例

下の画像は2つの桁を持つセグメント表示器要素の例を示します。対応する要素データ領域は画像の右の表で与えられます。

通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。



表5-72. セグメント表示器		
領域	値	
Z−指標	0	
X−座標	30	
Y-座標	30	
幅	174	
高さ	150	
要素形式	\$05	
桁数	2	
基数	10	
透過率	255	
色	\$FD0000	

5.2.5.5.10.7. ELEMENT_TYPE_GRAPH

下表はELMENT_TYPE_GRAPH特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。この要素は目的対象からのデータを可視化する図表を生成します。各作図に対して1つの入力(データの)流れがあります。

表5-73. ELEMENT_TYPE_GRAPH特有バラメータ	
N [°] ラメータ	値
ELEMENT_TYPE	\$06
ELEMENT_TYPE_GRAPH使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	41+表題長(N)

表5-74. ELEMENT_TYPE_GRAPH用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
表題色	3パイト	\$RRGGBB	表題のRGB色
背景色	3ハ [*] イト	\$RRGGBB	図表枠のRGB色
図表背景色	3ハ [*] イト	\$RRGGBB	図表のRGB色
表題文章	Nバイト	ヌル終端文字列、最大20パイト	-
作図数	1バイト	1~10	作図の数
X最小	4バイト	浮動小数点、秒	<分>:<秒>形式に変換されます。
X最大	4ハ・イト	浮動小数点、秒	<分>:<秒>形式に変換されます。
Y最小	4ハ・イト	浮動小数点	-
Y最大	4バイト	浮動小数点	-
動作形態	1시 イト	ビット0: マウス作用ON/OFF ビット1: 下地右端に図表合致	-

例

下の画像は3つの作図入力を持つ図表要素の例を示します。対応する要素データ領域は画像の右の表で示されます。

shboard I/O				
9	G	iraph		
	4			
2,00			an a	
1,50			and a second second second second	
1.00				
0.50				
0.00				
00:00:00	00:00:30	00:01:00	00:01:30	
Legend Unconn	ected			

通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。

表5-75. 図表		
領域	値	
Z−指標	0	
X−座標	10	
Y−座標	10	
幅	500	
高さ	250	
要素形式	\$06	
表題色	\$FFFFFF	
背景色	\$000000	
図表背景色	\$646464	
表題文章	"Graph¥0"	
作図数	3	
X最小	0	
X最大	100	
Y最小	0	
Y最大	2.45	
動作形態	\$01	

5.2.5.5.10.8. ELEMENT_TYPE_NUMERICAL_INPUT

下表はELMENT_TYPE_NUMERICAL_INPUT特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。この要素型はPCから目的対象へ数値を出力するのに使われます。要素は計器盤で編集可能です。

表5-76. ELEMENT_TYPE_NUMERICAL_INPUT特有パラメータ	
パラメータ	値
ELEMENT_TYPE	\$07
ELEMENT_TYPE_NUMERICAL_INPUT使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	26

表5-77. ELEMENT_TYPE_NUMERICAL_INPUT用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
最小	4ハ・イト	符号付き32ビット整数	-
最大	4ハ [*] イト	符号付き32ビット整数	-
値	4バイト	符号付き32ビット整数	初期値

例

下の画像は-100~100の有効な値の範囲と30の初期値を持つ数値入力要素の例を示します。対応する要素データ領域は画像の右の表で示されます。



Dashboard I/O	^ X
30	
Edit 🔲 Show Endpoints	

表5-78. 数値入力				
領域	値			
Z−指標	0			
X−座標	20			
Y−座標	20			
幅	50			
高さ	25			
要素形式	\$07			
最小値	-100			
最大値	100			
值	30			

5.2.5.5.10.9. ELEMENT_TYPE_RADIO

下表はELMENT_TYPE_RADIO特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。この要素は1 つだけの任意選択を選ぶことができるラジオ釦の群を生成します。最初は先頭の任意選択が選ばれます。

表5-79. ELEMENT_TYPE_RADIO特有パラメータ	
N [°] ラメータ	值
ELEMENT_TYPE	\$08
ELEMENT_TYPE_RADIO使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	17+文章領域長(N)

表5-80. ELEMENT_TYPE_RADIO用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	値	説明
フォントの大きさ	1バイト	0~100	-
項目数	1시7ト	1~10	-
向き	1バイト	0=水平、1=垂直	-
文章領域	Nバイト	ヌル終端文字列、最大20バイト	'/'で分離された任意選択一覧

例

下の画像は3つの釦を持つラジオ釦群の例を示します。対応する要素データ領域は画像の右の表で与えられます。

👉 通知: 要素の見かけはオペレーティング システムの版と構成設定で変わります。

Dashboard I/O	表5-81. ラシ	がす釦群	
	領地	或	値
© Op 1	Z−指	標	0
© Op 2	X-座	E標	20
© On 3	Y-座	ឪ標	20
0000	巾畐	Î	100
	高	さ	100
	要素	形式	\$08
	フォントの	大きさ	16
	項目	数	3
	向	き	1
	文章行	湏域	"Op 1/Op 2/Op 3¥
Edit 🔲 Show Endpoints			

5.2.5.5.10.10. ELEMENT_TYPE_PIE

下表はELMENT_TYPE_PIE特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。この要素は円図 表を生成します。断片の大きさは目的対象からPCへのデータに基づいて更新されます。各断片に対して1つのデータの流れがありま す。

表5-82. ELEMENT_TYPE_PIE特有パラメータ	
N [°] ラメータ	値
ELEMENT_TYPE	\$09
ELEMENT_TYPE_PIE使用時MSG_CONF_DASHBOARD_ELEMENTの総データ長	21+表題長(N)

表5-83. ELEMENT_TYPE_RADIO用MSG_CONF_DASHBOARD_ELEMENTへの追加データ領域

領域	大きさ	值	説明
背景色	3バイト	\$RRGGBB	背景のRGB色
表題色	3バイト	\$RRGGBB	表題のRGB色
表題	Nバイト	ヌル終端文字列、最大20バイト	-
断片数	1バイト	1~10	-

例

下の画像は3つの断片を持つ円図表要素の例を示します。対応する要素データ領域は画像の右の表で与えられます。

・通知:要素の見かけはオペレーティングシステムの版と構成設定で変わります。



5.2.5.5.10.11. ELEMENT_TYPE_SURFACE

下表はELMENT_TYPE_SURFACE特有パラメータとMSG_CONF_DASHBOARD_ELEMENTメッセージ用追加領域を記述します。この要素は目的対象からの格子データ点を可視化する3D表面作図を生成します。格子(Grid)型データを受け入れる1つの入力(データの)流れがあります。

表5-85. ELEMENT_TYPE_SURFACE特有パラメータ					
	パ゚ラメータ				
ELEMENT_TYPE	\$0D				
ELEMENT_TYPE_SU	URFACE∤	吏用時MSG_CONF_DASHBOARI	D_ELEMENTの総データ長	47	
表5-86. ELEMENT_T	YPE_SUR	ACE用MSG_CONF_DASHBOARE	D_ELEMENTへの追加データ領域		
領域 大きさ 値 説明 説明					
中詰め色	中詰め色 3/小小 \$RRGGBB 表面中詰めのRGB色				
網目色	3 バイト	\$RRGGBB	表面網目のRGB色		
背景色 4小小 \$AARRGGBB 背景のRGB色					
背景階調色 4小小 \$AARRGGBB 背景階調のRGB色					
軸色 3 [×] 3 [×] 3 [×] 3 [×] 3 [×] 3 [×] 3 [×] 3 [×]					

軸色	3バイト	\$RRGGBB	軸ののRGB色	
刻時色	3ハ [*] イト	\$RRGGBB	刻時のRGB色	
X回転	1バイト	-90~90	角度(°)でのX周囲視野の回転	
Y回転	1バイト	-90~90	角度(°)でのY周囲視野の回転	
Z回転	1バイト	-90~90	角度(°)でのZ周囲視野の回転	
属性	1/차기ト	ビット0: X軸表示 ビット1: Y軸表示 ビット2: Z軸表示 ビット3: 中詰め表示 ビット4: 網目表示 ビット5: 調色板色付け使用	-	
尺度調整動作	1/ㆍイト	0=静的 1=天井と床を尺度調整 2=天井を尺度調整 3=床を尺度調整 4=天井と床を粘性尺度調整 5=天井を粘性尺度調整 6=床を粘性尺度調整	_	
軸最小	4バイト	符号小数点	Y軸の最小値(床)	
軸最大	4시 イト	符号小数点	Y軸の最大値値(天井)	

例

次の画像は表面図表要素の例を示します。例に対する要素データ領域は画像の右の表で示されます。

▶ 通知:要素の見かけはオペレーティング システムの版と構成設定で変わります。



5.2.5.5.11. MSG_CONF_ADD_STREAM_TO_ELEMENT

このメッセージは既に構成設定されている計器盤の要素に対して既に構成設定されている(データの)流れを結合するのに使われます。

領域	大きさ	値	説明
メッセーシ [・] ID	1バイト	\$2C	-
データ長	2 パイト	6	-
計器盤ID	2バイト	$0000 \sim FFFF$	与えらた要素の計器盤のID
要素ID	2バイト	$0000 \sim FFFF$	要素のID
流れID	2 パイト	\$0000~\$FFFF この要素用の(データの)流れのID	

5.2.5.5.12. MSG_CONF_ACK

最後に受信した構成設定メッセージが有効か否かを確認するためにPCによって目的対象へ送られます。

領域	大きさ	值	説明
メッセーシ [・] ID	1バイト	\$30	-
データ長	2 パイト	1	-
状態	1시 イト	0=非OK、1=OK	OK=最後の構成設定がOKで適用されました。 非OK=最後の構成設定が無効で破棄になりました。

5.2.5.6. データ メッセーシ 詳細

5.2.5.6.1. MSG_DATA_STREAM

このメッセージは許可された全ての(データの)流れからのデータをPCへ送るのに使われます。1つの単一メッセージで全ての(データの)流れから1つまたは複数のデータの採取(試料)を送ることが可能です。許可された(データの)流れからのデータだけが送られます。

領域	大きさ	値	説明
メッセーシ [゙] ID	1시7나	\$40	-
データ長	2/차기ト	1+流れ数(N)×(2+各流れのバ仆数(Xn) +流れでの各データ採取(試料)の長さ(Xd))	-
流れ数(N)	1시7사	1~255	-
流れID	2/ㆍイト	\$0000~\$FFFF	流れXのID
バイト数(Xn)	1시 사	1~255	流れXからのバ仆数
流れXデータ採取(M)	流れのデータ型依存(Xd)	-	流れXのデータ

最後の行は流れ内の各採取に対して繰り返し、最後の3行はメッセージ内の各流れに対して繰り返します。

5.2.5.6.2. MSG_DATA_STREAM_SINGLE

このメッセージは許可された単一の(データの)流れからのデータをPCへ送るのに使われます。許可された(データの)流れからのデータだけが送られます。

領域	大きさ	值	説明
メッセーシ [・] ID	1시~사	\$41	1
データ長	2/ㆍイト	2+流れの1データ採取(試料)長(Xd)	-
流れID	2/ ㆍ イト	$0000 \sim FFFF$	流れのID
流れデータ採取	流れのデータ型依存(Xd)	-	流れのデータ

6. コード例断片

本章はこの使用者の手引きに於いて例で使われるコード断片を含みます。

6.1. データ ホーリング コート 例

大容量記憶クラス例はデータ変数機能のデータ ポーリングと制御の使い方の例として使われます。SAM L21 Xplained Pro基板がキット上の 目的対象USBとデ・ハッグUSBの両コネクタを通してホストコンピュータに接続されます。ATSAML21目的対象デ・ハイスはASFからのSAM L21 Xplained Pro用USB装置MSC例を走行しています。

この例を通して動かすことができるには以下が必要とされます。

- ・ (データ可視器(Data Visualizer)が含まれた)Atmel Studio 7(またはそれ以降)を持つホスト コンピュータ
- ・ATSAML21 Xplained Proキット

💦 すること: ・ SAM L21 Xplained Pro基板上の目的対象USBとデバッグUSBの両コネクタを接続してください。

この例はASFからのSAM L21 Xplained Pro用USB装置MSC例(USB Device MSC Example)を使います。

「>>> すること:・Atmel Studioで新しい例プロジェクトを作成してください。

- ・新規例(New Example)ダイアログでSAM L21デバイス系統(または関連デバイス)を選び、"MSC"キーワートによって濾過してください。
- ・USB装置MSC例(USB Device MSC Example)を選んでください。
- ・プロジェクト/解決策を構築(F7)してください。



すること: ・ プロジェクト プロハティを開いていください(解決策ェクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロハティ(Properties)を選んでください)。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Build	Configuration: N/A Platform: N/A	
Build Events		
Toolchain		
Device	Selected debugger/programmer	
Device	EDBG • ATML2130021800012626 V Interface: SWD V	
Tool		
Components		

では、データ可視器(Data Visualizer)が目的対象から変数をどうポーリングしてそれらの値を図画的形式で表示し得るかを見てください。

重要: データ ポーリングはデータ可視器がAtmel Studio内の拡張として動く場合にだけ利用可能です。これはAtmel Studioデベッガ 後処理部を通してデベイス上のデベッグ システムをアクセスすることが必要だからです。



データ ポーリングはSRAM位置で動き、故に変数がSRAMの何処に配置されるかを探すのにAtmel Studioの監視(Watch)ウィントウを使うこ とが必要です。

- すること:・ui.cに追加された2つの全域変数の位置を突き止めてください。
 - ・各変数を右クリックして監視に追加(Add to Watch)を選んでください。
 - ・それの位置を見つけるために監視(Watch)ウィントウで各変数の型(type)領域を調査してください。

Watch 1		
Name	Value	Туре
read_count	0	volatile uint32_t(static storage at address 0x200000e8.)
write_count	0	volatile uint32_t(static storage at address 0x200000ec.)

コート、特性分析(Code Profiling)インターフェースを構成設定して2つの変数を図表に接続するためにデータ可視器(Data Visualizer)に切り替え戻してください。

6.1.1. 計器盤制御を使う応用情報交換

では、データ可視器(Data Visualizer)で計器盤に配置された構成部品が応用内の変数をどう接続することができ、従って計器盤が走 行時に応用とどう情報交換し得るかを見てください。

予め定義された1000 USB同期パルスの間隔(1秒)に代えて、元のコートに対して参照基準と比較する変数を追加してください。

🖉 **すること**: ここで示されるようにui_process()処理部にLED点滅部を含むようにui.cを変更してください。

```
volatile uint32_t frame_comparator = 100;
volatile uint32_t frames_received = 0;
void ui_process(uint16_t framenumber)
{
    frames_received++;
    if (frames_received >= frame_comparator) {
       LED_Toggle(LED_0_PIN);
       frames_received = 0;
    }
}
```

すること:・プロジェクト/解決策を構築(F7)してください。

- ・デバッグ開始と中断(Start Debugging and Break) (Alt+F5)を使ってデバッグ作業を開始してください。
- ・uint32_t frame_coparator変数の場所を見つけてください。

Watch 1			≁ ģ ×
Name	Value	Туре	-
frame_comparator	0	uint32_t(static storage at address 0x200000e8.)	

6.2. 端末コード例

端末(Terminal)単位部の代表的な使い方は印刷型デバッグです。目的対象デバイスから端末へデバッグメッセージを出力するのに直列インターフェースが使われます。以下の例ではSPIインターフェースが使われますが、手続きはどの直列インターフェースでも同じです。

目的対象デバイスはATmega256RFR2 Xplained Proキット上のATmega256RFR2です。基板上でSPIインターフェースが既に内部的に配線されているため、必要とされる接続はホストコンピュータとXplained Pro基板間のUSBケーブルだけです。

すること: ・ Atmel Studioで新しいプロジェクトを作ってください(ファイル(File)→新規(New)→プロジェクト(Project)→GCC C実行可能 プロジェクト(GCC C Executable Project))。

・自動生成されたmain.cの内容を下のコードで置き換えてください。

```
#include <avr/interrupt.h>
#include <avr/sleep.h>
volatile uint8_t led_on;
volatile uint8_t send_message;
volatile uint8_t ticker = 0;
const char* message_on = "LED ON ";
const char* message_off = "LED OFF ";
ISR (INT4_vect)
    // 簡単な跳ね返り対策
    if (PINE & (1 << 4))
    return;
    // LED更新
    if (led on)
        PORTB |= (1 << 4);
    else
        PORTB &= ^{\sim}(1 << 4);
    led_on = ~led_on;
                                               // led_on反転
```

```
// メッセージ送出フラグ設定
    send_message = 1;
    ticker++; // 刻時部進行(+1)
    // 刻時部範囲制限(リセット)
    if (ticker \geq 10)
    ticker = 0;
}
void spi_send (const char data)
{
   PORTB &= ^{\sim}(1 \ll \text{PINBO});
                                             // SPIに文字を送出
   SPDR = data;
    while (!(SPSR \& (1 << SPIF)));
                                              // 文字送出完了待機
    PORTB \mid = (1 \iff \text{PINB0});
int main(void)
ł
   DDRB = (1 \iff PINB4);
                                              // PORTB4出力設定
   PORTB \mid = (1 \iff \text{PINB4});
                                              // LED OFF
    led_on = 0;
                                              //
    PORTE |= (1<<PINE4);
                                              // 線の浮きを避けるために釦ピンのプルアップを許可
    EIMSK = (1 \iff INT4);
                                              // 釦ピンに対して下降端割り込みを許可
    EICRB = (1 \iff ISC41);
                                              11
    // SPI
   DDRB |= (1 << PINB2) | (1 << PINB1) | (1 << PINB0); // MOSI,SCK,/SSを出力として設定
    PORTB \mid = (1 \iff \text{PINB0});
                                              // /SSをhighに設定
   SPCR = (1 << SPE) | (1 << MSTR) | (1 << SPR0); // SPI許可、主装置、fck/16クロック速度に設定
    sei();
                                              // 全体割り込み許可
   while(1) {
        if(send_message) {
            // メッセージ送出
            const char* pmessage;
            if (led_on)
               pmessage = message_on;
            else
                pmessage = message_off;
            while (*pmessage)
                spi_send(*pmessage++);
            // 刻時部の値を送出
            spi_send(ticker + '0');
            spi_send('¥n');
            // 送出完了
            send_message = 0;
        }
   }
```

コート は Xplained Pro 基板上のスィッチ(SW0)が押された時に割り込み起動し、各押下でLED0を交互切り替えします。スィッチが押される度毎にSPIインターフェースでメッセージが出力されます。

🎽 すること: プロジェクト/解決策を構築(F7)してください。

Data Visualizer



すること: ・ プロジェクト プロペティを開いていください(解決策ェクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロペティ(Properties)を選んでください。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Build Build Events Toolchain Device Tool Components Advanced	Configuration: N/A Selected debugger/programm EDBG • ATML178403020000	er 0573 ·	Pla Interface	: JTA	G VA	
すること : デ'ベッグなしで開 ing) (Ctrl+Alt+F 的対象に応用を	始(Start Without Debugg う)を選ぶことによって目 書いてください。	Del NU	bug Tool Windows Start Debu Attach to Stop Debu Start With Disable de	s V Igging Target Igging out De BugW	Vindow Help and Break Alt+F5 t Ctrl+Shift ebugging Ctrl+Alt+F /IRE and Close	+F5 F5 t_1b\main.c
♥ すること: ツール(Tools)メニュー 内側の拡張として ださい。	ーでそれを選ぶことによってA Cデータ可視器(Data Visualize	Atme er)を同	l Studio 周いてく		Is Window Help Command Prompt Pack Manager Device Programming Add target Data Visualizer Code Snippets Manager Extensions and Updates Atmel Gallery Profile External Tools Import and Export Settings Customize Options	Ctrl+Shift+P Ctrl+K, Ctrl+B

6.3. 図表コート 例

図表(Graph)単位部をどう使うかを実演するには夜間動作切り替え器での例が使われます。目的対象ハードウェアとしてI/O1 Xplained P ro拡張を持つATmega256RFR2 Xplained Proが適合します。I/O1拡張基板は夜か昼間かを検出するのに使うことができる光感知器が特徴です。この情報は例えば、周辺が暗くなった時に電灯をONに切り替えるのに使うことができます。

この例は以下の装備とソフトウェアを必要とします。

- (データ可視器(Data Visualizer)が含まれた)Atmel Studio 7(またはそれ以降) を持つホスト コンピュータ
- ・ATmega256RFR2 Xplained Proキット
- ・I/O1 Xplained Pro拡張
- 例を走らせるには以下のハードウェア構成設定が必要とされます。
- ・ATmega256RFR2 Xplained ProのEXT1ヘッダに接続されたI/O1 Xplained Pr o拡張
- ・ホストコンピュータからATmega256RFR2 Xplained Proへ接続されたUSBケーブル 構成設定の写真が右で示されます。



6.3.1. 基本的な図表

始めるために、光感知器の採取を実装して図表での作図としてデータを見ることができるようにホストコンピュータへデータを流してください。

すること: ・Atmel Studioで新しいプロジェクトを作ってください(ファイル(File)⇒新規(New)⇒プロジェクト(Project)⇒GCC C実行可能 プロジェクト(GCC C Executable Project))。

・自動生成されたmain.cの内容を下のコードで置き換えてください。

<pre>#include <avr io.h=""></avr></pre>	
<pre>uint16_t adc_value = 0;</pre>	
<pre>void adc_init(void) {</pre>	
$ADMUX = (1 \iff REFS1);$	// 内部1.5V基準電圧、シングルェンド入力としてのADC0
ADCSRA \mid = (1< <aden);< td=""><td>// ADC許可</td></aden);<>	// ADC許可
while $(0x00 == (ADCSRB \& (1 << REFOK)));$	// 基準電圧安定待機
}	
uint16_t adc_sample(void){	
ADCSRA \mid = (1< <adsc);< td=""><td>// ADC変換起動</td></adsc);<>	// ADC変換起動
while $(0x00 == (ADCSRA \& (1 << ADIF)));$	// 変換完了待機
ADCSRA $\mid = (1 \iff ADIF);$	// 割り込み要求フラグ解除
return (ADC);	
}	
<pre>void spi_init(void) {</pre>	
$DDRB = (1 \le PINB0) (1 \le PINB2) (1 \le PINB2)$	31); // 従装置選択(PB0)、MOSI(PB2)、SCK(PB1)を出力に設定
PORTB $\mid = (1 << PINBO);$	// 従装置選択=High(非活性)
SPCR = (1< <spe) (1<<mstr);="" spi許可<="" td="" =""><td>J、主装置動作を設定、クロック極性と位相は既定を維持(上昇端で採取)</td></spe)>	J、主装置動作を設定、クロック極性と位相は既定を維持(上昇端で採取)
}	
<pre>void spi_send(uint8_t data) {</pre>	
PORTB &= \sim (1< <pinbo);< td=""><td>// 従装置選択=Low</td></pinbo);<>	// 従装置選択=Low
SPDR = data;	// SPIでデータ送信開始
while (0x00 == (SPSR & (1< <spif)));< td=""><td>// 送信完了待機</td></spif)));<>	// 送信完了待機
PORTB $\mid = (1 \leq PINB0);$	// 従装置選択=High
}	
int main(void) {	
adc init();	
spi init();	
while (1)	
adc value = adc sample();	
spi send(adc value & 0xFF); // ADC値	をSPI上でホストへ送出、下位8ビットだけが有用なデータを含みます。

コートは継続的にADCを採取してATmega256RFR2 Xplained Pro基板上のEDBG(組み込みデバッガ)へSPIインターフェースを渡ってデータを送ります。EDBGはその後にDGIを渡ってホストコンピュータへSPIデータを送ります。ATmega256RFR2のADCは10ビットですが、この例では下位側8ビットだけが有用なデータを含みます。

🎽 すること: プロジェクト/解決策を構築(F7)してください。



すること: ・ プロジェクト プロハティを開いていください(解決策ェクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロハティ(Properties)を選んでください)。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Toolchain Device	Selected debugger/programmer
Tool	EDBG • ATML1784030200000573 • Interface: JTAG •
Components	
Advanced	



Data Visualizer

いっ すること	: ツール(Tools)メニューでそれを選ぶことによってAtmel Studio 内側の拡張としてデータ可視器(Data Visualizer)を開いてく ださい。

Too	ls Window Help	
2	Command Prompt	
-	Pack Manager	
1	Device Programming	Ctrl+Shift+P
	Add target	
	Data Visualizer	
	Code Snippets Manager	Ctrl+K, Ctrl+B
¢	Extensions and Updates	
	Atmel Gallery Profile	
	External Tools	
	Import and Export Settings	
	Customize	
Ф	Options	

6.3.2. 文字列標識追加

動作切り替え器を実装する時です。切り替え器が実際に動いていて正しい閾値で切り替えていることを調べるには図表(Graph)単位 部の文字列標識機能が有用です。ATmega256RFR2 Xplained ProのCDC USARTインターフェースを構成設定することによって動作形態 が切り替えられる時毎に文字列を送ります。これらのメッセージはその後に文字列標識として図表内で表示することができます。最初 に、下で示されるようにコートを拡大してください。

重要: コート は目的対象CPUが8MHzで走行している場合にだけ動きます。正しくヒュースを設定するのにAtmel Studioのプログラミ

ング(Programming)ダイアログを使ってください(ツール(Tools)⇒デバイスプログラミング(Device Programming))。クロックは内部発振

 \rightarrow

一日本に設定されるべきで、UNDIV8L1-Aは設定され	しつべき じはめりません。
<pre>#include <avr h="" io.=""></avr></pre>	
#include <avr interrupt.h=""></avr>	
const char* message_on = "NIGHT MODE ON";	
const char* message_off = "NIGHT MODE OFF";	
uint16_t adc_value = 0;	
uint8_t nightmode_threshold = 40;	
<pre>uint8_t nightmode_active = 0;</pre>	
void adc_init(void){	
$ADMUX = (1 \ll REFS1);$	// 内部1.5V基準電圧、シングルエンド入力としてのADC0
ADCSRA \mid = (1< <aden);< td=""><td>// ADC許可</td></aden);<>	// ADC許可
while $(0x00 == (ADCSRB \& (1 << REFOK)));$	// 基準電圧安定待機
$\frac{1}{2}$	
uintib_t adc_sample(vold) { ADCCPA = (1/(ADCC))	// 100亦始却乱
$ADCSRA \mid = (I \land ADSC),$ $1 : 1 (0 - 00 = (ADCSDA + (I \land ADIE))):$	// ADC変換起動 // 本協安又往撤
while $(0x00 == (ADCSRA & (I << ADIF)))$,	// 发换元] 付饿 // 生心 7. 西卡·尼·納哈
$ADCSRA = (1 \langle ADIF \rangle,$	// 割り込み安水/// 脾际
return (ADC),	
$\frac{\text{vold spl_init(vold)}}{\text{pppp}} = \frac{1}{(1/(\text{plyp}))} = \frac{1}{(1/(\text{plyp}))} = \frac{1}{(1/(\text{plyp}))}$	
$DOD_{LD} = (1 < (DINDO) + (DINDO) + (1 < (DINDO) + (DINDO) $	51), // 仮装直選択(PB0)、MOSI(PB2)、SUK(PB1)を出力に設正 // 従け累認セルロ1/世活地)
$PORIB = (1 \langle (OPP) \rangle (1 \langle (NOTP) \rangle) \rangle (1 \langle (NOTP) \rangle)$	
SPCR = (1 < SPE) (1 < MSTR); // SPI = 1	」、土装直動作を設定、クロック/極性と位相は既正を維持(上昇端で採取)
void spi_send(uint8_t data) {	
PORTB &= (1 < PINB0),	// 從袋直迭状=LOW
SPDR = data;	// SPIでアータ达信開始
while $(0x00 == (SPSK \& (I < (SPIF)));$	// 达信元」符機
PORTB = (T << PTNB0);	// 促装直選択=High
vold cdc_init(vold) {	
$\bigcup BKKI = 51;$	// 8MHz UPU/ロックに基く9600のホーレート
UCSRIB = (I < < RXENI) (I < < TXENI) (I < < RXC)	JIE1), // 安信部計可、 法信部計可、 8ビット 乂子の大きさ、



、すること: プロジェクトを構築し、Atmel Studioのデバック(Debug)メニューで単に継続(Continue)(F5)を選ぶことによって応用を書い て走らせてください。

6.3.3. 水平カーソルを使うコート

ここまでは、光感知器によって生成された値を表示するのと夜間動作切り替え器が2つの動作間を交互切り替えする時を表示するの にデータ可視器(Data Visualizer)の図表(Graph)単位部が使われています。図表(Graph)単位部は目的対象応用が走行している間に それとの情報交換にも使うことができます。この例では水平カーソルを使うことによって夜間動作閾値を動的に調整することができます。 最初に、コート、はCDC USARTでやって来るデータを受け入れるように拡張されなければなりません。水平カーソルの出力4^{ハ・}小浮動値 で、CDCインターフェースを渡って目的対象応用へ送られます。この浮動値は夜間動作切り替え器用の閾値として使われます。

```
#include <avr/io.h>
#include <avr/interrupt.h>
const char* message_on = "NIGHT MODE ON";
const char* message_off = "NIGHT MODE OFF";
union u_float{
   float flt;
    char data[4];
};
uint16_t adc_value = 0;
uint8_t nightmode_threshold;
uint8_t nightmode_active = 0;
union u_float cdc_received_data;
uint8_t cdc_read_index=0;
ISR (USART1_RX_vect) {
    cdc_received_data.data[cdc_read_index] = UDR1 & 0xFF; // バ化はCDC UART、MSB先行で受信されます。
    if (3 == cdc_read_index) {
        nightmode_threshold = (uint8_t) cdc_received_data.flt; // 完全な浮動値が受信されます。
        cdc_read_index = 0;
```

```
else {
       cdc_read_index++;
void adc_init(void) {
   ADMUX = (1 \iff \text{REFS1});
                                          // 内部1.5V基準電圧、シングルェンド入力としてのADC0
   ADCSRA |= (1 \leq ADEN);
                                          // ADC許可
   while (0x00 == (ADCSRB & (1 << REFOK))); // 基準電圧安定待機
uint16_t adc_sample(void) {
   ADCSRA \mid = (1<<ADSC);
                                          // ADC変換起動
   while (0x00 == (ADCSRA \& (1 << ADIF)));
                                          // 変換完了待機
                                          // 割り込み要求フラグ解除
   ADCSRA \mid = (1 \iff ADIF);
   return (ADC);
}
void spi init(void) {
   DDRB |= (1<<PINB0) | (1<<PINB1); // 従装置選択(PB0)、MOSI(PB2)、SCK(PB1)を出力に設定
   PORTB |= (1<<PINBO);
                                          // 従装置選択=High(非活性)
   SPCR = (1 \leq SPE) | (1 \leq MSTR);
                                 // SPI許可、主装置動作を設定、クロック極性と位相は既定を維持(上昇端で採取)
void spi_send(uint8_t data) {
   PORTB &= ^{\sim}(1 \leq \text{PINB0});
                                          // 従装置選択=Low
   SPDR = data;
                                          // SPIでデータ送信開始
                                          // 送信完了待機
   while (0x00 == (SPSR \& (1 << SPIF)));
   PORTB |= (1<<PINB0);
                                          // 従装置選択=High
void cdc_init(void) {
                                           // 8MHz CPUクロックに基く9600のボーレート
   UBRR1 = 51;
   UCSR1B = (1<<RXEN1) | (1<<TXEN1) | (1<<RXCIE1); // 受信部許可、送信部許可、8ビット文字の大きさ、
                                                 // 受信割り込み許可
void cdc_send(const char data) {
   while (0x00 == (UCSR1A & (1<<UDRE1))); // もっとデータの準備が整うための送信部待ち
   UDR1 = data;
                                          // データを送出
void send_message(const char* message) {
   while (*message)
   cdc_send(*message++);
   cdc send(0);
                                         // 文字列標識はヌル終端を必要とします。
int main(void) {
   adc_init();
   spi_init();
   cdc init();
   sei();
                                          // 全体割り込み許可
   while (1) {
       adc_value = adc_sample();
       spi send(adc value & 0xFF); // ADC値をSPI上でホストへ送出、下位8ビットだけが有用なデータを含みます。
       // より大きなADC値は電灯なし
       if (adc_value > nightmode_threshold) {
           if (0x00 == nightmode_active) { // 夜間動作を不活性から活性に変更
               nightmode active = 0x01;
               send_message(message_on);
           }
       } else {
           if (0x01 == nightmode_active) { // 夜間動作を活性から不活性に変更
               nightmode_active = 0x00;
               send_message(message_off);
       }
   }
```

すること: ・ プロジェクトを構築し、Atmel Studioのデハック(Debug)メニューで単に継続(Continue) (F5)を選ぶことによって応用を書いて走らせてください。

6.4. オシロスコーフ[°] コート 例

オシロスコープ(Oscilloscope)単位部をどう使うかを実演するには光感知器目的対象応用が使われます。光感知器例は感知器を採取す るのにA/D変換器が使われるかなり一般的なデータ供給元で、この例は広範囲の他のデータ供給元に適用します。

この例は以下の装備とソフトウェアを必要とします。

- ・(データ可視器(Data Visualizer)が含まれた)Atmel Studio 7(またはそれ以降) を持つホストコンピュータ
- ・ATmega256RFR2 Xplained Proキット
- ・I/O1 Xplained Pro拡張

例を走らせるには以下のハードウェア構成設定が必要とされます。

- ・ATmega256RFR2 Xplained ProのEXT1ヘッダに接続されたI/O1 Xplained Pr o拡張
- ・ ホスト コンピュータからATmega256RFR2 Xplained Proへ接続されたUSBケーブル 構成設定の写真が右で示されます。

オシロスコープ(Oscilloscope)単位部で光感知器データを見ることができるには、光 感知器を採取して直列インターフェースを渡ってATmega256RFR2 Xplained Pro上 の組み込みデバッカ(EDBG)にデータを送るコートでATmega256RFR2目的対象が 書かれなければなりません。EDBGはその後にデータをホストコンピュータへ送るの にデータ交換器インターフェース(DGI:Data Gateway Interface)を使います。

最初に、Atmel Studioで目的対象応用コート用の新しいプロジェクトが構成設定されなければなりません。



すること: ・Atmel Studioで新しいプロジェクトを作ってください(ファイル(File)⇒新規(New)⇒プロジェクト(Project)⇒GCC C実行可能 プロジェクト(GCC C Executable Project))。

・自動生成されたmain.cの内容を下のコードで置き換えてください。

```
#include <avr/io.h>
#include <avr/interrupt.h>
uint16_t adc_value = 0;
volatile uint8_t send_data = 0;
void adc_init(void) {
                                         // 内部1.5V基準電圧、シングル エンド入力としてのADC0
   ADMUX = (1 \iff REFS1);
   ADCSRA |= (1<<ADATE) | (1<<ADATE) ; // ADC許可、自動起動動作、変換完了割り込み許可
                                        // ADC変換起動元としてタイマ/カウンタの比較A一致
   ADCSRB \mid = (1<<ADTS1) | (1<<ADTS0);
   while (0x00 == (ADCSRB & (1 << REFOK))); // 基準電圧安定待機
void spi init(void) {
   DDRB |= (1<<PINB0) | (1<<PINB1); // 従装置選択(PB0)、MOSI(PB2)、SCK(PB1)を出力に設定
   PORTB \mid = (1 < < PINB0);
                                          // 従装置選択=High(非活性)
                                          // 2倍速動作、CPUクロックが8MHzの時に4MHz SPIクロック
   SPSR = (1 \leq SPI2X);
   SPCR = (1<<SPE) | (1<<MSTR); // SPI許可、主装置動作を設定、クロック極性と位相は既定を維持(上昇端で採取)
void spi_send(uint8_t data) {
   PORTB &= ^{\sim}(1 << PINB0);
                                          // 従装置選択=Low
   SPDR = data;
                                          // SPIでデータ送信開始
   while (0x00 == (SPSR \& (1 \leq SPIF)));
                                         // 送信完了待機
   PORTB \mid = (1<<PINB0);
                                         // 従装置選択=High
void timer_init() {
                                         // 計時器用TOP值(比較A値)設定
   OCROA = 80;
                                         // 比較一致で計時器解消動作
   TCCROA = (1 \leq WGMO1);
   TCCROB = (1 \leq CSOO);
                                         // 前置分周なしCPUクロックによって計時器をクロック駆動
ISR (ADC_vect) {
   adc_value = (ADC);
                                          // 光感知器採取(試料)格納
   TIFRO \mid = (1 \leq OCFOA);
                                          // 次の採取を許可するために計時器割り込み要求フラグを解除
```

Data Visualizer

このコート、は100kHzの採取速度を与える10µs毎に新しい採取を取るようにADCを構成設定します。これはリセットするのに先立って80まで上昇計数する計時器を使うことによって達成されます。コート、は2分周(既定)のクロック前置分周器を持つ内部16MHzクロックとCKDIV8 ヒュース、設定なしで走行する目的対象CPUに基づきます。データ採取(試料)はDGI SPIインターフェースを渡ってEDBGへ送られます。SPIイン ターフェースは4MHzで走ります。ATmega256RFR2のADCは10ビットですが、この例では下位側8ビットだけが有用なデータを含みます。

✗ すること: プロジェクト/解決策を構築(F7)してください。

AtmelStudio					
ssistX ASF Project	Buil	d Debug Tools	Window	Help	
🗆 - 🛳 🖴 🗳 🕹 ć	*	Build Solution		F7	
↔ ५. ५ ৫ ৫ ⊾		Rebuild Solution Clean Solution		Ctrl+Alt+F7	9
	*	Build low_power_test_ Rebuild low_power_test Clean low power test	1b st_1b 1b		,lc
		Batch Build Configuration Manage	er		
		Compile		Ctrl+F7	

すること: ・ プロジェクト プロハディを開いてください(解決策ェクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロハディ(Prope rties)を選んでください)。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Build Build Events	Configuration: N/A Platform: N/A
Toolchain Device	Selected debugger/programmer
Tool	EDBG • ATML1784030200000573 • Interface: JTAG •
Components	
Advanced	

Data Visualizer



6.5. 計器盤コード例

計器盤(Dashboard)単位部をどう使うかを実演するには夜間動作切り替え器での例が使われます。目的対象ハードウェアとしてI/O1 Xpl ained Pro拡張を持つATmega256RFR2 Xplained Proが適合します。I/O1拡張基板は夜か昼間かを検出するのに使うことができる光感知器が特徴です。この情報は例えば、周辺が暗くなった時に電灯をONに切り替えるのに使うことができます。

この例は以下の装備とソフトウェアを必要とします。

- (データ可視器(Data Visualizer)が含まれた)Atmel Studio 7(またはそれ以降) を持つホスト コンピュータ
- ・ATmega256RFR2 Xplained Proキット
- ・I/O1 Xplained Pro拡張

例を走らせるには以下のハートウェア構成設定が必要とされます。

- ・ATmega256RFR2 Xplained ProのEXT1ヘッダに接続されたI/O1 Xplained Pr o拡張
- ・ホストコンピュータからATmega256RFR2 Xplained Proへ接続されたUSBケーブル 構成設定の写真が右で示されます。

Xplained Pro上のATmega256RFR2目的対象は夜間動作切り替え器を実行するコードを書かれなければなりません。



```
すること: ・Atmel Studioで新しいプロジェクトを作ってください(ファイル(File)⇒新規(New)⇒プロジェクト(Project)⇒GCC C実行可能
            フ<sup>°</sup>ロシ<sup>*</sup>ェクト(GCC C Executable Project))。
           ・自動生成されたmain.cの内容を下のコードで置き換えてください。
#include <avr/io.h>
#include <avr/interrupt.h>
union u_double{
   double dbl;
   char data[8];
};
uint16_t adc_value = 0;
uint8_t nightmode_threshold;
uint8_t nightmode_active = 0;
union u_double cdc_received_data;
uint8_t cdc_read_index=0;
ISR (USART1_RX_vect) {
   cdc_received_data.data[cdc_read_index] = UDR1 & 0xFF; // バ かはCDC UART、MSB先行で受信されます。
   if (7 == cdc_read_index) {
       nightmode_threshold = (uint8_t) cdc_received_data.db1; // 完全なダブル値が受信されます。
       cdc_read_index = 0;
   }
   else {
       cdc_read_index++;
void adc_init(void) {
   ADMUX = (1 \iff REFS1);
                                           // 内部1.5V基準電圧、シングルエンド入力としてのADC0
   ADCSRA \mid = (1<<ADEN);
                                           // ADC許可
   while (0x00 == (ADCSRB & (1 << REFOK))); // 基準電圧安定待機
uint16_t adc_sample(void) {
   ADCSRA = (1<<ADSC);
                                           // ADC変換起動
                                           // 変換完了待機
   while (0x00 == (ADCSRA \& (1 << ADIF)));
   ADCSRA \mid = (1 \iff ADIF);
                                           // 割り込み要求フラグ解除
   return (ADC);}
void spi init(void) {
   DDRB |= (1<<PINB0) | (1<<PINB2) | (1<<PINB1); // 従装置選択(PB0)、MOSI(PB2)、SCK(PB1)を出力に設定
   PORTB |= (1<<PINB0);
                                            // 従装置選択=High(非活性)
   SPCR = (1<<SPE) | (1<<MSTR); // SPI許可、主装置動作を設定、クロック極性と位相は既定を維持(上昇端で採取)
void spi_send(uint8_t data) {
   PORTB &= ^{\sim}(1 \leq PINB0);
                                           // 従装置選択=Low
   SPDR = data;
                                           // SPIでデータ送信開始
   while (0x00 == (SPSR \& (1 << SPIF)));
                                           // 送信完了待機
   PORTB |= (1<<PINBO);
                                           // 従装置選択=High
void cdc_init(void) {
                                           // 8MHz CPUクロックに基く9600のボーレート
   UBRR1 = 51;
   UCSR1B = (1<<RXEN1) | (1<<TXEN1) | (1<<RXCIE1); // 受信部許可、送信部許可、8ビット文字の大きさ、
                                                  // 受信割り込み許可
void cdc_send(const char data) {
   while (0x00 == (UCSR1A & (1<<UDRE1))); // もっとデータの準備が整うための送信部待ち
   UDR1 = data;
                                            // データを送出
int main(void) {
   adc init();
    spi_init();
```



このコート'は継続的にADCを採取してATmega256RFR2 Xplained Pro基板上のEDBG(組み込みデ`ハ`ッカ`)へSPIインターフェースを渡って デ`ータを送ります。EDBGはその後にDGIを渡ってデ`ータをホスト コンピュータへ送ります。ATmega256RFR2のADCは10ヒ`ットですが、この例 では下位側8ヒ`ットだけが有用なデ`ータを含みます。

加えて、コートはCDC USARTを構成設定して単一バイトとして夜間動作切り替え器の状態を送ります。CDC USARTで受信したデータは Double値として解析され夜間動作切り替え器用の閾値として使われます。

すること: プロジェクト/解決策を構築(F7)してください。

AtmelStudio						
ssistX ASF Project	Buil	d Debug	Tools	Window	Help	
🗆 - 😩 💾 🗳 👗 Ć	*	Build Solution			F7	
 ↔ G. G. C.\Users\ 	÷	Rebuild Soluti Clean Solution Build low_pow Rebuild low_poy Clean low_poy	on ver_test_ ower_te wer_test	1b st_1b 1b	Ctrl+Alt+F7	٤ اد
		Batch Build Configuration Compile	Manag	er	Ctrl+F7	

すること: ・ プロジェクト プロハプティを開いてください(解決策ェクスプローラ(Solution Explorer)でプロジェクトを右クリックしてプロハプティ(Prope rties)を選んでください)。

・ツール(Tool)タブで適切なツールとインターフェースを選んでください。

Build Build Events	Configuration: N/A	N/A 👻
Toolchain Device	Selected debugger/programmer	
Tool	EDBG • ATML1784030200000573 • Interface: JTAG	•
Components		
Advanced		

Data Visualizer

すること : 継続(Continue) (F5)を選ぶことによって目的対象に応 用を書いてデバッグを開始してください。		De	bug Tools Window Help Windows	•
) ()	ΨΨ ▲ ■ Δ • Σ	Start Debugging and Break Attach to Target Stop Debugging Start Without Debugging Disable debugWIRE and Close Continue Execute Stimulifile Set Stimulifile	Alt+F5 Ctrl+Shift+F5 Ctrl+Alt+F5 F5
すること : ツール(Tools)メニューでそれを選ぶことによってAtmel Stu 内側の拡張としてデータ可視器(Data Visualizer)を開いて ださい。	dio てく		Tools Window Help Command Prompt Pack Manager Device Programming Add target Data Visualizer Code Snippets Manager	Ctrl+Shift+P Ctrl+K, Ctrl+B
			Extensions and Updates Atmel Gallery Profile External Tools Import and Export Settings Customize Ontions	

6.6. 自動構成設定コード例

データ流れ規約の自動構成設定機能は何れかのDGI直列インターフェースまたはシリアルポート上で使うことができます。この例ではATtiny104 Xplained Nano基板の仮想COMポートが使われます。この基板は目的対象ATtiny104デバイスを書くのに使うことができる組み込みデバッガ(mEDBG)を持ち、mEDBGはATtiny104のUSARTビンに接続されるUSB上の仮想COMポートを提供します。

この例を走らせるのに必要とされるハートウェア接続はホストコンピュータとXplai ned Nano基板間にUSBケーブルを接続することだけです。

データ流れ規約はXplained Nano上の釦の状態と16ビット計数器の値をホストコンピュータへ送るのに使われます。



すること: ・Atmel Studioで新しいプロジェクトを作ってください(ファイル(File)⇒新規(New)⇒プロジェクト(Project)⇒GCC C実行可能 プロジェクト(GCC C Executable Project))。

・自動生成されたmain.cの内容を次のコードで置き換えてください。

#include <avr/io.h> uint8 t start token = 0xAB; uint8_t config_id_packet[] = { /* 通票 */ 0x5F, 0xB4, 0x00, 0x86, 0x4A, /* LRC8チェックサムを指定 */ 0xC0, 0xFF, 0xEE, 0xC0, 0xFF, 0xEE, 0xC0, 0xFF, 0xEE, 0xC0, 0xFF, 0xEE, /* 構成設定ID */ 0x78, /* 実際のチェックサム */ /* 通票の反転 */ 0xA0 }; uint16_t count = 0; uint8_t send_id = 0; void uart_send(uint8_t byte) { /* 送信緩衝部空待ち */ while (!(UCSRA & (1<<UDRE))); UDR = byte; /* 緩衝部にデータを配置、データ送出 */ } int main(void) { CCP = 0xD8;// 構成設定変更保護レジスタ CLKPSR = 0x00;/* 8MHz主クロックを得るためにクロック前置分周器を1に設定 */ /* ボーレートを38400に設定: UBR=(8000000/(16*38400))-1=12 */ UBRR = 12;UCSRB = $(1 \leq TXEN)$; /* USART送信部許可 */ PUEB |= (1<<PINB1); /* 釦ピンのプルアップを許可 */ while (1) { /* 構成設定ID送出 */ if $(send_id == 0)$ { for (int i = 0; i < sizeof(config_id_packet);i++)</pre> uart_send(config_id_packet[i]); $send_id = 100;$ /* データ送出 */ uart_send(start_token); uart send(count & 0xFF); $uart_send(count >> 8);$ uart_send (!(PINB & (1<<PINB1))); uart_send(~start_token); count +=100; send_id--; }

このコートは各反復に対して100ずつ増やされる16ビット計数器の値を継続的に送ります。加えて、ATtiny104 Xplained Nanoの押し釦 (PINB1)の状態が送られます。押し釦ピンは釦が押された時にLowであることに注意してください。100回の繰り返し反復毎に自動構成 設定IDハ[°]ケットが送られます。自動構成設定IDは\$C0,\$FF,\$EE,\$C0,\$FF,\$EE,\$C0,\$FF,\$EE,\$C0,\$FF,\$EEで、故に構成設定ファイル は**"COFFEEC0FFEEC0FFEEC0FFEE**"と名付けられなければなりません。自動構成設定IDハ[°]ケット形式は「自動構成設定」で記述さ れ、デ⁻タ ハ[°]ケットの形式は「流れ形式」で記述されます。目的対象ATtiny104デ⁻ハ⁻イスは8MHz内部RC発振器でクロック駆動され、仮想 OCMポ[°]ート用のUSARTは8ビット文字幅、1停止ビット、ハ[°]リティなしで38400のボ⁻レートで動きます。

すること : プロジェクト/解決策を構築(F7)してください。	AtmelStudio				
	ssistX ASF Project	Bui	ld Debug T	ools Window	Help
	u - 🖆 💾 🛣 🐰	6 🛎	Build Solution		F7
	6 4. C C .	-	Rebuild Solutio	n	Ctrl+Alt+F7
			Clean Solution		
	main.c [*] → X	"	Build low_powe	er_test_1b	
	C:\User	51	Rebuild low_po	ower_test_1b	LIC .
			Clean low_pow	er_test_1b	
			Batch Build		
			Configuration N	Manager	
			Compile		Ctrl+F7
	L				



7. 既知の問題

一般的に、データ可視器(Data Visualizer)は正しく機能するためにインストールされるべき全てのWindows更新を必要とします。特に、以下の任意選択知識基盤(KB:Knowledge Base)は図表に対して正しく表示されるのに必須です。

Windoqws版	重要なKB
Windows 8.1	 KB2975719, https://support.microsoft.com/en-us/help/2975719 KB2978092, https://support.microsoft.com/en-us/help/2978092
Windows 8	 · KB2975331, https://support.microsoft.com/en-us/help/2975331 · KB2978092, https://support.microsoft.com/en-us/help/2978092
Windows 7	 · KB2670838, https://support.microsoft.com/en-us/help/2670838 · KB2978092, https://support.microsoft.com/en-us/help/2978092

8. 資料改訂履歴

資料改訂	日付	注釈
42730A	2016年5月	初版資料公開
42730B	2017年1月	独自計器盤に表面図表と表の要素を追加 新しい資料雛型
40001903A	2017年6月	 Micrchip DS40001903A版でAtmel 42730B版を置き換え 図表要素の自動構成設定に自動非表示作図任意選択を追加 データ可視器が動くのに必要とされるKBに関する既知の問題を追加 データ流れ自動構成設定部分を追加 最新の自動検出規約機能でDGI制御盤とシリアルポート制御盤の記述を更新 Power DebuggerのVout制御に関する電力インターフェース構成設定を更新 コート、特性分析DGIインターフェースにスタック監視、AVR OCDメッセージとAVR休止監視の機能を追加 実装変更に従って計器盤表示部で表要素を更新 構成設定任意選択追加に従って計器盤で図表要素を更新 全般的に複写画像を更新
40001903B	2017年9月	図表単位部で水平カーソルを更新

Microchipウェフ゛サイト

Microchipはhttp://www.microchip.com/で当社のウェブ サイト経由でのオンライン支援を提供します。このウェブ サイトはお客様がファイルや情報を容易に利用可能にする手段として使われます。お気に入りのインターネット ブラウザ を用いてアクセスすることができ、ウェブ サイトは以下の情報を含みます。

- ・製品支援 データシートと障害情報、応用記述と試供プログラム、設計資源、使用者の手引きとハートウェア支援資料、最新ソフトウェア配布と 保管されたソフトウェア
- ・全般的な技術支援 良くある質問(FAQ)、技術支援要求、オンライン検討グループ、Microchip相談役プログラム員一覧
- ・Microchipの事業 製品選択器と注文の手引き、最新Microchip報道発表、セントとイベントの一覧、Microchip営業所の一覧、代理 店と代表する工場

お客様への変更通知サービス

Microchipのお客様通知サービスはMicrochip製品を最新に保つのに役立ちます。加入者は指定した製品系統や興味のある開発ツール に関連する変更、更新、改訂、障害情報がある場合に必ず電子メール通知を受け取ります。

登録するにはhttp://www.microchip.com/でMicrochipのウェブ サイトをアクセスしてください。"Support"下で"Customer Change Notificati on"をクリックして登録指示に従ってください。

お客様支援

Microchip製品の使用者は以下のいくつかのチャネルを通して支援を受け取ることができます。

- ・代理店または販売会社
- ・最寄りの営業所
- ・現場応用技術者(FAE:Field Aplication Engineer)
- ・技術支援

お客様は支援に関してこれらの代理店、販売会社、または現場応用技術者(FAE)に連絡を取るべきです。最寄りの営業所もお客様の手助けに利用できます。営業所と位置の一覧はこの資料の後ろに含まれます。

技術支援はhttp://www.microchip.com/supportでのウェブ サイトを通して利用できます。

Microchipデバイスコート、保護機能

Microchipデバイスでの以下のコード保護機能の詳細に注意してください。

- ・Microchip製品はそれら特定のMicrochipデータシートに含まれる仕様に合致します。
- ・Microchipは意図した方法と通常条件下で使われる時に、その製品系統が今日の市場でその種類の最も安全な系統の1つである と考えます。
- コート、保護機能を破るのに使われる不正でおそらく違法な方法があります。当社の知る限りこれらの方法の全てはMicrochipのデータシートに含まれた動作仕様外の方法でMicrochip製品を使うことが必要です。おそらく、それを行う人は知的財産の窃盗に関与しています。
- ・Microchipはそれらのコードの完全性について心配されているお客様と共に働きたいと思います。
- ・Microchipや他のどの半導体製造業者もそれらのコードの安全を保証することはできません。コード保護は当社が製品を"破ることができない"として保証すると言うことを意味しません。

コート、保護は常に進化しています。Microchipは当社製品のコート、保護機能を継続的に改善することを約束します。Microchipのコート、保護機能を破る試みはデジタルシニアム著作権法に違反するかもしれません。そのような行為があなたのソフトウェアや他の著作物に不正なアクセスを許す場合、その法律下の救済のために訴権を持つかもしれません。

法的通知

デバイス応用などに関してこの刊行物に含まれる情報は皆さまの便宜のためにだけ提供され、更新によって取り換えられるかもしれま せん。皆さまの応用が皆さまの仕様に合致するのを保証するのは皆さまの責任です。Microchipはその条件、品質、性能、商品性、 目的適合性を含め、明示的にも黙示的にもその情報に関連して書面または表記された書面または黙示の如何なる表明や保証もし ません。Microchipはこの情報とそれの使用から生じる全責任を否認します。生命維持や安全応用でのMicrochipデバイスの使用は完 全に購入者の危険性で、購入者はそのような使用に起因する全ての損害、請求、訴訟、費用からMicrochipを擁護し、補償し、免責 にすることに同意します。他に言及されない限り、Microchipのどの知的財産権下でも暗黙的または違う方法で許認可は譲渡されま せん。

商標

Microchipの名前とロゴ、Mcicrochipロゴ、AnyRate、AVR、AVRロゴ、AVR Freaks、BeaaconThings、BitCloud、CryptoMemory、CryptoR F、dsPIC、FlashFlex、flexPWR、Heldo、JukeBlox、KeeLoq、KeeLoqロゴ、Kleer、LANCheck、LINK MD、maXStylus、maXTouch、Med iaLB、megaAVR、MOST、MOSTロゴ、MPLAB、OptoLyzer、PIC、picoPower、PICSTART、PIC32ロゴ、Prochip Designer、QTouch、Rig htTouch、SAM-BA、SpyNIC、SST、SSTロゴ、SuperFlash、tinyAVR、UNI/O、XMEGAは米国と他の国に於けるMicrochip Technology Incorporatedの登録商標です。

ClockWorks、The Embedded Control Solutions Company、EtherSynch、Hyper Speed Control、HyperLight Load、IntelliMOS、mTou ch、Precision Edge、Quiet-Wireは米国に於けるMicrochip Technology Incorporatedの登録商標です。

Adjacent Key Suppression、AKS、Analog-for-the-Digital Age、Any Capacitor、AnyIn、AnyOut、BodyCom、chipKIT、chipKITロゴ、C odeGuard、CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、 DAM、ECAN、EtherGREEN、In-Circuit Serial Programming、ICSP、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet¤ ゴ、Mindi、MiWi、motorBench、MPASM、MPF、MPLAB Certifiedロゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PureSilicon、QMatrix、RightToucpロゴ、REAL ICE、Ripple Blocker、SAM-IC E、Serial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sense、WiperLock、Wireless DNA、ZENAは米国と他の国に於けるMicrochip Technology Incorporatedの商標です。

SQTPは米国に於けるMicrochip Technology Incorporatedの役務標章です。

Silicon Storage Technologyは他の国に於けるMicrochip Technology Inc.の登録商標です。

GestICは他の国に於けるMicrochip Technology Inc.の子会社であるMicrochip Technology Germany II GmbH & Co. KGの登録商 標です。

ここで言及した以外の全ての商標はそれら各々の会社の所有物です。

© 2017年、Microchip Technology Incorporated、米国印刷、不許複製

DNVによって認証された品質管理システム

ISO/TS 16949

Microchipはその世界的な本社、アリゾナ州のチャント・ラーとテンヘ。、オレゴン州グラシャムの設計とウェハー製造設備とカリフォルニアとイント・の設計センターに対してISO/TS-16949:2009認証を取得しました。当社の品質システムの処理と手続きはPIC® MCUとdsPIC® DSC、KEELOQ符号飛び回りデバイス、直列EEPROM、マイクロ周辺機能、不揮発性メモリ、アナログ製品用です。加えて、開発システムの設計と製造のためのMic rochipの品質システムはISO 9001:2000認証取得です。

日本語© HERO 2020.

本使用者の手引きはMicrochipのデータ可視器(Data Visualizer)使用者の手引き(DS40001903B-2017年9月)の翻訳日本語版です。 日本語では不自然となる重複する形容表現は省略されている場合があります。日本語では難解となる表現は大幅に意訳されている 部分もあります。必要に応じて一部加筆されています。頁割の変更により、原本より頁数が少なくなっています。

必要と思われる部分には()内に英語表記や略称などを残す形で表記しています。

青字の部分はリンクとなっています。一般的に赤字の0,1は論理0,1を表します。その他の赤字は重要な部分を表します。



世界的な販売とサービス

本社

2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 技術支援: http://www.microchip.com/ support ウェブ アトレス: www.microchip.com

米国

アトランタ Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455

オースチン TX Tel: 512-257-3370

ボストン Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088

シカゴ Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075

ダラス Addison, TX Tel: 972-818-7423 Fax: 972-818-2924

デトロイト Novi, MI Tel: 248-848-4000

ヒューストン TX Tel: 281-894-5983

インデアナポリス Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380

ロサンセ・ルス Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 D-J- NC

Tel: 919-844-7510 ニュ**ーヨーク** NY Tel: 631-435-6000

サンホセ CA Tel: 408-735-9110 Tel: 408-436-4270 カナダ - トロント Tel: 905-695-1980

Fax: 905-695-2078

亜細亜/太平洋

亜細亜太平洋支社 Suites 3707-14, 37th Floor Tower 6, The Gateway Harbour City, Kowloon 香港

Tel: 852-2943-5100 Fax: 852-2401-3431

オーストラリア - シトニー Tel: 61-2-9868-6733 Fax: 61-2-9868-6755

中国 - 北京 Tel: 86-10-8569-7000 Fax: 86-10-8528-2104

中国 - 成都 Tel: 86-28-8665-5511 Fax: 86-28-8665-7889

中国 - 重慶 Tel: 86-23-8980-9588 Fax: 86-23-8980-9500

中国 - 東莞 Tel: 86-769-8702-9880 中国 - 広州 Tel: 86-20-8755-8029

中国 – 杭州 Tel: 86-571-8792-8115 Fax: 86-571-8792-8116 中国 - 香港特別行政区

Tel: 852-2943-5100 Fax: 852-2401-3431 中国 - 南京

Tel: 86-25-8473-2460 Fax: 86-25-8473-2470 中国 - 青島

Tel: 86-532-8502-7355 Fax: 86-532-8502-7205

中国 - 上海 Tel: 86-21-3326-8000 Fax: 86-21-3326-8021

中国 - 瀋陽 Tel: 86-24-2334-2829 Fax: 86-24-2334-2393

中国 - 深圳 Tel: 86-755-8864-2200 Fax: 86-755-8203-1760

中国 - 武漢 Tel: 86-27-5980-5300 Fax: 86-27-5980-5118 中国 - 西安

Tel: 86-29-8833-7252 Fax: 86-29-8833-7256 亜細亜/太平洋

中国 - 廈門 Tel: 86-592-2388138 Fax: 86-592-2388130 中国 - 珠海 Tel: 86-756-3210040 Fax: 86-756-3210049 イント・ーハンガロール Tel: 91-80-3090-4444 Fax: 91-80-3090-4123 イント - ニューテリー Tel: 91-11-4160-8631 Fax: 91-11-4160-8632 イント・ファネー Tel: 91-20-3019-1500 日本 - 大阪 Tel: 81-6-6152-7160 Fax: 81-6-6152-9310 日本 - 東京 Tel: 81-3-6880-3770 Fax: 81-3-6880-3771 韓国 - 大邱 Tel: 82-53-744-4301 Fax: 82-53-744-4302 韓国 - ソウル Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934 マレーシア - クアラルンプール Tel: 60-3-6201-9857 Fax: 60-3-6201-9859 マレーシア ー ヘ・ナン Tel: 60-4-227-8870 Fax: 60-4-227-4068 フィリピン - マニラ Tel: 63-2-634-9065 Fax: 63-2-634-9069 シンガホール Tel: 65-6334-8870 Fax: 65-6334-8850 台湾 - 新竹 Tel: 886-3-5778-366 Fax: 886-3-5770-955 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600

Fax: 886-2-2508-0102 タイー バンコク Tel: 66-2-694-1351 Fax: 66-2-694-1350

欧州

オーストリア – ウェルス Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 テンマーク - コヘンハーケン Tel: 45-4450-2828 Fax: 45-4485-2829 フィンラント – エスホー Tel: 358-9-4520-820 フランス – パリ Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 フランス - サン クルー Tel: 33-1-30-60-70-00 トイツ – カ・ルトンク・ Tel: 49-8931-9700 ドイツ - ハーン Tel: 49-2129-3766400 トイツー ハイルブロン Tel: 49-7131-67-3636 ドイツ – カールスルーエ Tel: 49-721-625370 ドイツ - ミュンヘン Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 ドイツ – ローセンハイム Tel: 49-8031-354-560 イスラエル - ラーナナ Tel: 972-9-744-7705 イタリア ー ミラノ Tel: 39-0331-742611 Fax: 39-0331-466781 イタリア ー パドバ Tel: 39-049-7625286 オランダ – デルーネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-7289-7561 ホーラント – ワルシャワ Tel: 48-22-3325737 ルーマニア - ブカレスト Tel: 40-21-407-87-50 スペイン – マトリート Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 スウェーデン – イェーテホリ Tel: 46-31-704-60-40 スウェーデン – ストックホルム Tel: 46-8-5090-4654 イキ・リス – ウォーキンカ・ム Tel: 44-118-921-5800 Fax: 44-118-921-5820