



室内空気品質監視器:Atmel STARTとMPLAB®コード構成部(MCC) を使うファームウェア作成

序説

著者: Babashree Ingale, Ankit Phougat, Microchip Technology Inc.

室内空気品質監視器(AQM:Air Quality Monitor)はMicrochipのAVR-IoT WG開発基板とMikroElektronikaの感知器 といくつかのClick基板を使って実装されます。AVR-IoT WG開発基板はMicrochipのATmega4808マイクロコントローラ (AVR[®] MCU)、ATECC608A CryptoAuthentication™安全素子、ATWINC1510 Wi-Fi[®]単位部が装備されます。

AQMは粒子状物質(PM2.5)、二酸化炭素(CO2)、総揮発性有機化合物(TVOC:Total Volatile Organic Components) のような主な気中浮遊汚染物質と共に湿度と温度を監視します。マイクロコントローラは採取されたこれらの読み取りを処理してPM2.5感知器の読み取りから空気質指数(AQI:Air Quality Index)を計算します。AQIと採取した他の空気品質要素は外部EEPROMに格納され、OLEDで表示され、Googleクラウトに対して更新されます。

AQM文書概要

以下の文書がAQMシステム資料を網羅します。

- 1. この応用記述はAQM応用に対するAVR-IoT階層(AVR-IoT WGソースコート)の使い方を網羅します。更に、Atmel STARTとMCCを使ってマイクロコントローラ周辺機能とClick基板の構成設定詳細を網羅します。室内AQM応用は AVR-IoT WGソースコートの先頭で作成されます。
- 2. 「AN3403 室内空気品質監視器:概念と実装」応用記述はAQMシステムのハートウェアとファームウェアの概要とそれの電力考察を記述します。
- 3.「空気品質監視器:使用者の手引き」はハートウェア準備、ハートウェア接続、動作手順、LED表示、AVR-IoT WG開発 基板のWi-Fi単位部の準備、クラウト上でのAQMデータの可視化を網羅します。

このファームウェア作成の手引きを読む前に「AN3403 - 室内空気品質監視器:概念と実装」応用記述を読むことが推奨さます。

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

序詞	兑 •••		1
1.	Atmel	STARTを使う室内空気品質監視器ファームウェア作成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	1.1.	AVR-IoT WG感知器節点例プロジェクト生成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	1.2.	追加周辺機能構成設定 ····································	4
	1.3.	START コート [*] ファイル作成 ・・・・・・ 1	0
	1.4.	応用コードと感知器ドライバを追加 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	1
	1.5.	Atmel START生成コートを変更・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
	1.6.	יארר ¹ 722 · · · · · · · · · · · · · · · · · ·	2
2.	MCC	を使う室内空気品質監視器ファームウェア作成 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
	2.1.	新規MPLAB [®] Xプロジェクト作成 ····································	3
	2.2.	AVR-IoT WG感知器節点例プロジェクト生成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
	2.3.	追加周辺機能構成設定 · · · · · · · · · · · · · · · · · · ·	4
	2.4.	MCCコート [*] ファイル作成・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
	2.5.	応用コートと感知器トライバを追加 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	2
	2.6.	MCC生成コートを変更 ····································	2
	2.7.	コンハイラ設定・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	3
3.	応用	誓き込み ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	4
	3.1.	Atmel Studioで書き込み ····································	4
	3.2.	MPLAB® X IDEで書き込み・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	5
4.	結び	2	5
5.	改訂和	复歴······2	5
Mic	rochip	ካ፲7[°] ቻ ለት · · · · · · · · · · · · · · · · · · ·	6
製品	習変更	通知サービス・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	6
お	F 様支	援······2	6
Mic	rochip	デバイス コード保護機能 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	6
法的	勺通知	2	6
商权	票 • • • ∙	2	7
品質	〔 管理	<u> </u>	7
世界	早的な!	販売とサービス ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	8

1. Atmel STARTを使う室内空気品質監視器ファームウェア作成

本章はAtmel STARTでAVR-IoT WGソースコートを生成する手順と室内AQM応用ファームウェアを 作成ための手順の一覧を網羅します。

表1-1.はAtmel STARTを使ってファームウェアの作成に使われるソフトウェア ツールと版番号を一覧にします。言及された、またはそれ以降の版のツールを使うことが推奨されます。

表1-1. ソフトウェア ツールと版番号				
ソフトウェア ツール	版番号			
Atmel Studio 7	7.0.2397版			
Atmel START	1.7.279版			
ATmega_DFP	1.4.331版			
AVR GCC	5.4.0版			
avr8_gnu	3.6.2.1778版			

1.1. AVR-IoT WG感知器節点例プロジェクト生成

・ Atmel Studioを開いてください。File(ファイル)→New(新規)→Atmel START Example Project(Atmel START例プロジェクト)へ行ってください。

・BROWSE EXISTING EXAMPLE(既存例閲覧)頁が読み込まれるまで待ってください。

助言:より良い感じにするために頁を最大化してください。

・検索してAVR-IoT WG Sensor Node(AVR-IoT WG感知器節点)例を選んでください。その後にOPEN SELECTED EXAMPLE(選んだ例を開く)をクリックしてください。

			1 11 - 11 6 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 -	
tmei START examples are a p	for each compatible board a	aded programmers, example projects will work "out of the box" but	are also easily to modify using Atmel STA	KI.
formation about each exam	iple.	and device, resulting in high quality production ready code with a sink	an memory rootprint. Read the available t	isel guides to get mor
ooking for older examples? <u>/</u> xamples	Atmel Software Framework 3	contains previous releases of non-configurable examples.Click File -	New - Example Project menu in Atmel Stu	udio to see the ASF3
ampies.				
earch: AVR-IoT Wo	G Sensor Node	Category: All ~	Board: All	
Name	Categories	Description	Board(s) supported	User guide
AVR IoT Alibaba Cloud Sensor Node	<u> </u>	A sensor node for the Alibaba IoT Cloud based on the ATmega4808, WINC1510 and ECC608A.	ATmega4808 AVR IoT WG	User guid
AVR IoT WG Sensor Node	<u> </u>	A sensor node for the Google IoT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.	ATmega4808 AVR IoT WG	User guid
AVR loT WG Sensor Node with AirQuality Click	<u> </u>	A sensor node for the Google IoT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.	ATmega4808 AVR IoT WG	User guid
AVR IoT WG Sensor Node with Weather Click	<u> </u>	A sensor node for the Google IoT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.	ATmega4808 AVR IoT WG	User guid
with weather Click	2 (1)	ATmega4808, WINCISTO and ECC608A.		

・マイクロコントローラ周辺機能のDASHBOARD(計器盤)表示とそのドライハの構成設定ウィントウが開きます。

図1−2. 計器盤表示	
Atmel START ATmega4808	What's New Help
MY SOFTWARE COMPONENTS	(?
Add software component Add software component Add software component System driver	Show system drivers C
	CRYPTO AUTHENTICATIONO
BOD 🔯 CPUINT 🔅 CLKCTRL 🔅 SLPCTRL 🔅	
AVR IOT WG SENSOR NODE A sensor node for the Google IOT Core Cloud based on the ATmega4808, WINC1510 and ECC608A.	\otimes
GENERAL	
1 User guide	
S Link	
Rename component	
GENERATE PROJECT	• •

1.2. 追加周辺機能構成設定

下の小項目は室内空気品質監視器応用に必要とされる追加の単位部/周辺機能の構成設定を説明します。

1.2.1. ADC

- ・SHT31アナログ感知器は湿度と温度の測定に使われます。ADC周辺機能はこの感知器出力を読むように構成設定されます。ADC 周辺機能トライハはAVR-IoTWG感知器節点例プロジェクトの一部としてプロジェクトに追加されています。故に、もう一度プロジェクトに ADC周辺機能トライハを追加する必要はありません。
- ・ADCを構成設定するため、ADC_0をクリックしてください。ADC構成設定ウィントウが現れます。
- SHT31感知器の温度と湿度の出力ピンは各々、PD6とPD7のポートピンに接続されます。それらのポートピンをADCチャネルとして構成設定するため、COMPONENT SIGNALS(構成部品信号)部分へ行き、PD6とPD7をADCピンとして選んでください。AVR-IoT WG基板でPD5に接続された光感知器は室内AQM応用で使われないのでPD5ピンを非選択にしてください。

Atmel START ATmega4808 What's New Help MY SOFTWARE COMPONENTS Image: Component strain initialization + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case ADC: Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case ADC: Image: Component strain + API to support simple ADC use-case Image: Component strain + API to support simple ADC use-case ADC use-
MY SOFTWARE COMPONENTS Image: Component component ADC_0 Image: Component component Image: Component component Component component Component component Component component Component component AIN/0: PD0 (Postriprio) Image: Component Cocks AIN/1: PD1 (Postriprio) AIN/2: PD2 (Postriprio) Image: Component Cocks AIN/2: PD2 (Postriprio) AIN/2: PD2 (Postriprio)
ADC_0 MCode basic driver: Peripheral initialization + API to support simple ADC use-case GENERAL COMPONENT SETTINGS COMPONENT SIGNALS I User guide Driver: Driver:sADC:Basic AIN/0: PD0 (PORT/P/10) Rename component CLOCKS AIN/1: PD1 (PORT/P/11) Remove component ADC: Main Clock (CLK_MAIN) (10 MHz) AIN/2: PD2 (PORT/P/12)
Component Component Component Image: Component Clock (CLK_MAIN) (10 MHz) AIN/2: PD2 (PORT/P12) Image: Component ADC: Main Clock (CLK_MAIN) (10 MHz) AIN/2: PD2 (PORT/P12)
Image: Second point control of the
Image: CLOCKS AIN/1: PD1 (PORT/P/11) Image: CLOCKS AIN/1: PD2 (PORT/P/12) Image: CLOCKS AIN/2: PD2 (PORT/P/12) Image: CLOCKS AIN/2: PD2 (PORT/P/12) Image: CLOCKS AIN/2: PD2 (PORT/P/12)
Image: Construction of the second s
Remove component
AIN/4: PD4
AIN/5: PD5
AIN/6: VD6
AIN/7: VD7
AIN/12: PF2 (PORT/P/22)
AIN/13: PF3 (PORT/P/23)
AIN/14: PF4 (PORT/P/24)
AIN/15: PF5 (PORT/P/25)
GENERATE PROJECT

注: AQMハートウェアは2つの温度感知器を持ちます。1つはアナログ感知器(SHT31-ARP)で、別の1つは(AVR-IoT WG基板)の基板上 MCP9808デジタル感知器です。AQMは温度監視にMCP9808デジタル感知器を使います。

・ ADCの採取累積機能が室内AQM応用に使われま た ADC 携 式 ついじゅつ CONFICU	図1-4. ADC採取構成設定
9。ADC構成設定リイアリでSAMPLING CONFIGU RATION(採取構成設定)をEnable(許可)してくださ	SAMPLING CONFIGURATION
い。 SAMPLING CONFIGURATION(採取構成設定)部でSAMPNUM: Accumulation Samples(SAMPN	ASDV: Automatic Sampling Delay 🛛 🖓
UM:採取累積)をAccumulate 8 samples(8採取累積)	SAMPDLY: Sampling Delay Selection: 📀 0x0
に、DUTYCYC: Duty Cycle(DUTYCYC:アューアイサ イクル)を25% Duty cycle(25%デューティサイクル)に構成	INITDLY: Initial Delay Selection: 📀 Del
設定してください。	SAMPNUM: Accumulation Samples: 📀 Acc
	DUTYCYC: Duty Cycle: 259
	SAMPLEN: Sample length: 0x0

X]1-4. ADC採取構成設定				
	SAMPLING CONFIGURATION			Enable:	\checkmark
	ASDV: Automatic Sampling Delay Variation:	0			
	SAMPDLY: Sampling Delay Selection:	0	0x0	he	×v
	INITDLY: Initial Delay Selection:	0	Delay 0 CLK_ADC cycles		~
	SAMPNUM: Accumulation Samples:	0	Accumulate 8 samples		~
	DUTYCYC: Duty Cycle:	0	25% Duty cycle		~
	SAMPLEN: Sample length:	0	0x0	he	×v
	SAMPCAP: Sample Capacitance Selection:	0			
1					

1.2.2. UART

- ・SPS30 PM2.5感知器はUARTインターフェースを通してマイクロコントローラと通信します。
- ・UART1を構成設定するため、ADD Software Component(ソフトウェア構成部品追加)をクリックすることによってUSART周辺機能ドライバを 追加してください。

図1-	-5. ソウトウェア構成部品追加	
	Atmel START ATMEGR4808	What's New Help
	MY SOFTWARE COMPONENTS	?
👰 DASHBOARD	Add software component Com	Show system drivers 👁 🔶 Show hardware 👁
•		CRYPTO AUTHENTICATION C

Drivers(ドライバ)タブを展開してUSART周辺機能ドライバを追加してください。SELECTED COMPONENTS(選んだ構成部品)枠にUSARTが現れます。

図1-6. USART単位部ドライバ追加

ADD SOFTWARE COM	IPONENTS			3		
Filter	Filter					
Name	Description	1		Add		
O SPI	SPI Serial Peripheral Interface (SPI), synchronous serial communication interface.					
Ö Timer	Timer Task timer functionality using a timer counter peripheral.					
USART Universal Synchronous Asynchronous Receiver Transmitter (USART) to transfer data from one device to the other.						
Ö VREF	VREF Voltage Reference for ADC, DAC, and AC peripherals, providing programmable voltages for each reference source.					
Ö WDT	WDT Watchdog Timer (WDT) for monitoring correct operation and make it possible to recover from error situations such as runaway or deadlock code.					
SELECTED COMPON	IENTS					
Name		Count	Add/remove	Remove all		
C USART		1	$\ominus \oplus$	面		
Add component(s) Canc						

・UART周辺機能を構成設定するためにDASHBOARD(計器盤)表示でUSART_1をクリックしてください。COMPONENT SETTINGS(構成部品設定)部でMode(動作形態)をAsync IRQ Mode(非同期割り込み動作)として、Instance(実体)をUSART1として構成設定してください。COMPONENT SIGNALS(構成部品信号)部でRXDをPC1に、TXDをPC0に構成設定してください。BASIC CONFIGURATI ON(基本構成設定)部でBaud Rate(ボーレート)を115200に設定してください。

図1−7. USART単位部構成設定								
GENERAL	COMPONE	NT SETTINGS		COMPONE	NT SIGNALS			
User guide	Driver:	Drivers:USART:Basic	\sim	RXD:	PC1	\sim		
	Mode:	Async IRQ Mode	\sim	TXD:	PC0	\sim		
Kename component	Instance:	USART1	\sim	XCK:		~		
Remove component	CLOCKS			XDIR:		~		
	USART:	Main Clock (CLK_MAIN) (10 M	1Hz) 🗸					
DRIVERS:USART:BASIC (ASYNC IR	Q MODE) CONFIGURATIC	IN ON USART1						
DRIVER CONFIGURATION		ADVANCED	CONFIGURATION	N				
Printf support:		LBME: Loop	-back Mode Enab	le: 🕜				
RX Buffer Size:	8	V DBGRUN: D	ebug Run:	0				
TX Buffer Size:	8	~						
BASIC CONFIGURATION								
RXEN: Reciever enable:								
TXEN: Transmitter Enable:								
PMODE: Parity Mode: 🥳	No Parity	~						
SBMODE: Stop Bit Mode:	1 stop bit	~						
CHSIZE: Character Size:	Character size: 8 bit	~						
Baud Rate: 🛛 🔞	115200	dec v						

1.2.3. NVM

- ・内部EEPROMは既定要素と外部EEPROMの送出と書き込みの指標の格納に使われます。内部EEPROMはNVM制御周辺機能を 使ってアクセスされます。
- ・NVM制御を構成設定するため、USART部で記述された手順に従うことによってFlash周辺機能ドライバを追加してください。
- ・DASHBOARD(計器盤)表示でFLASH_0を選んでください。
- ・構成部品をNVMに改名してください。

	MY SOFTWARE COMPONENTS		?
			¢
)		800 O	
9		RENAME COMPONENT ×	\otimes
	GENERAL	Specify new name: NVM Rename COMPONENT SIGNALS	
	User guide		
	🔊 Rename component		
	Remove component		

1.2.4. 事象システム

- ・AVR-IoT WG基板上の基板上切替器はOLED表示器をONに切り替えるのとOLEDで表示される要素を変更するのに使われま す。AVR-IoT WG基板上のSW0基板上切替器はマイクロコントローラのPORT1_PIN6に接続されます。マイクロコントローラの事象システムと CCL周辺機能を使い、どんなソフトウェア介在もなしに切替器押下事象を検出するように切替器跳ね返り制御機構が実装されます。
- ・USART部で記述されたような手順に従うことによってEvent System(事象システム)とDigital Glue Logic (CCL)(デジタル接続論理回路 (CCL))の両トライハを追加してください。

- ・図1-9.で示されるように、事象システム周辺機能を構成設定するため、EVENT(事象)シンボルをクリックしてください。EVENT SYSTEM CONFIGURATOR(事象システム構成設定部)ウィントウが開きます。
- Periodic Interrupt Timer output 3(周期割り込み計時器3)をConnect user to event cvhannel 1(使用部を事象チャネル1に接続)に、 Port 1 Pin 6(ホート1 ビン6)をConnect user to event cvhannel 4(使用部を事象チャネル4に接続)に接続してください。接続するにはクリッ クして保持し、そしてトラックして(引き摺って)ください。従って、PIT3とPort1 Pin6は各々、チャネル1とチャネル4に対する事象生成部として 働きます。



・図1-9.で示されるように、同じ方法でConnect user to event cvhannel 1(使用部を事象チャネル1に接続)をLUT1IN_Aに、Connect user to event cvhannel 4(使用部を事象チャネル4に接続)をLUT1IN_Bに接続してください。従って、LUT1IN_AとLUT1IN_Bは各々、チャネル1とチャネル4に対する事象使用部として働きます。

1.2.5. CCL

- ・どんなソフトウェア介在もなしに切替器跳ね返り制御機構を実装するため、濾波器任意選択と共にCCL周辺機能の参照表1(LUT1)が 使われます。CCL周辺機能は各切替器押下事象に対して割り込みを生成します。
- ・図1-10.で示されるように、CCLシンボルをクリックしてください。CCL CONFIGURATOR(CCL構成設定部)ウィンドウが開きます。
- ・構成設定のためにLUT1を選ぶのにLUT1をクリックしてください。LUT1チェック枠をチェックすることによってLUT1を許可してください。



- ・図1-10.で示されるように、LUT1 Event A(LUT1事 象A)をLUT1のC入力へ、LUT1 Event B(LUT1事 象B)をLUT1のB入力へ接続してください。
- ・図1-10.で示されるように、LUT1内側のSettings(設定)アイコンをクリックしてください。
- CLKSRC: Clock Source Selection(CLKSRC: クロック 元選択)をIN[2] is clocking the LUT(IN2がLUTをク ロック駆動)に構成設定し、Settings(設定)ウィンドウを 閉じてください。

図1-11. LUT1クロック元構成設定

CCL SETTINGS		×
LUT1 (SETTINGS)		C
CLKSRC: Clock Source Selection: 🛛 🔞	IN[2] is clocking the LUT	\sim
INSEL0: LUT Input 0 Source Selection: ()	Masked input	\sim
INSEL1: LUT Input 1 Source Selection: ()	Event input source B	\sim
INSEL2: LUT Input 2 Source Selection: ()	Event input source A	~
TRUTH1: Truth 1: 0	0	dec v
OUTEN: Output Enable: 0		
		Close

 LUT1 CONFUGIRATION(LUT1構成設定)部へ行ってください。Truth Logic(真理論理回路)でVaule(hex)(値(16進数))を1に設定 してください。Filter Options(濾波器任意選択)をEnable Filer(濾波器許可)に構成設定し、Clock Selection(クロック選択)をIN2 Clock (IN2クロック)に構成設定してください。



・最後に、LUT1は下図で示されるように見えます。



- ・マイクロコントローラは周期的にスタンハイ休止動作へ行きます。マイクロコントローラがスタンハイ休止動作の間に切替器押下事象を検出するため、CCLはスタンハイ休止動作で動かなければなりません。
- ・DASHBOARD(計器盤)表示へ行き、Digital Glue Logic(デジタル接続論理回路)を選んでください。
- ・BASIC CONFIGURATION(基本構成設定)部でRUNSTDBY: Run in Standby(スタンハイで走行)任意選択をチェックしてください。

図1-1	4. スタンバイで走行ビット構成設定	
	MY SOFTWARE COMPONENTS	
ARD	DRIVERS:CCL:INIT (CCL) CONFIGURATION ON CCL	
SHBO	BASIC CONFIGURATION	LOOKUP TABLE 0 CONFIGURA
B	ENABLE: Enable:	LUTEN: LUT Enable:
⊳ ğ•(RUNSTDBY: Run in Standby: 🛛 🗸	OUTEN: Output Enable:
<u>ш</u> ,	SEQSEL0: Sequential Selection 0: Sequential logic disabled	CCL Configurator Clock: CCL cl
	SEQSEL1: Sequential Selection 1: 🔹 Sequential logic disabled	CIOCK:
\odot	LOOKUP TABLE 1 CONFIGURATION	INSEL0: LUT Input 0 Source Selection
Ð	LUTEN: LUT Enable:	INSEL1: LUT Input 1 Source Sel

 INTERRUPT CONTROL CONFUGIRATION(割り込み制御構成設定)部でINTMODE1: Interrupt Mode 1 Selection(INTMODE1: 割り込み動作1選択)をSense rising edge(上昇端感知)に、他のMode n Selection(動作n選択)をInterrupt disabled(割り込み禁止)に 構成設定してください。

図1-15.割り込み制御構成設定					
INTERRUPT CONTROL CONFIGU	IRATION	I			
INTMODE0: Interrupt Mode 0 Selection:	0	Interrupt disabled	~		
INTMODE1: Interrupt Mode 1 Selection:	0	Sense rising edge	~		
INTMODE2: Interrupt Mode 2 Selection:	0	Interrupt disabled	~		
INTMODE3: Interrupt Mode 3 Selection:	0	Interrupt disabled	~		

1.2.6. 休止動作

- ・AQM応用はスタンバイ休止動作を使います。マイクロコントローラは実行すべき作業がない時に休止動作へ行きます。
- 休止動作を構成設定するため、DASHBOAR(計器盤)表示でSLPCTRL(休止制御器)をクリックしてください。SEN: Sleep enable(SEN: 休止許可)を許可してSMODE: Sleep mode(SMODE:休止動作形態)をStandby Mode(スタンハイ動作)に構成設定してください。

図1-16. 休止動作構成設定			
	SLF	CTRL	\otimes
	SLP	CTRL	
GENERAL	COMPONENT SETTINGS		COMPONENT SIGNALS
1 User guide	Driver: Drivers:SLPCTR	L:Init	
DRIVERS:SLPCTRL:INIT (SLPCTRL) CONFIG	SURATION ON SLPCTRL		C
ENABLE CONFIGURATION		SLEEP MODE CONFIGURAT	ION
SEN: Sleep enable: 0		SMODE: Sleep mode:	Standby Mode V

1.2.7. EEPROM 3 Click

空気品質感知器読み取り値はEEPROM 3 Click基板上に存 在する外部EEPROMに格納されます。Add Software Compon ent(ソフトウェア構成部品追加)をクリックすることによってEEPROM 3を含めてください。Middleware(中間ソフトウェア)その後にStorag e Click(Click記憶装置)を展開し、EEPROM3 Clickを追加して ください。

図1-17. EEPROM 3 Click追加

ADD SOFTWARE COMPON	IENTS				
Filter					
Name + Power Managem	Description			Add	
± 🗅 Sensors Click					
– 🗁 Storage Click					
 EEPROM Click It features 24C08WP - a highly reliable, high performance CMOS technology serial 8K EEPROM in DIP packaging. The 24C08WP EEPROM uses the I2C addressing protocol. 					
SEEPROM2 Cli	EEPROM 2 click carries ST's M95M02-DR DIP-8 socket EEPROM chip with 256 KB of memory. The chip has byte and page write speeds equal or less than 10 ms. The board communicates with the target board MCU through the mikroBUS SPI interface				
SELECTED COMPONENT	S	Count	Add/romovo	Pomovo all	
		1		Remove an	
Seer NOWIS CITCK		1	00	ш	
Add component(s) Cancel					

1.3. STARTコート ファイル作成

コードを生成するため、下部に存在するGENERATE PROJECT(プロジェクト生成)をクリックしてください。プロジェクト要素を設定するためにNew Atmel Start Project(新しいAtmel STARTプロジェクト)が現れます。図1-18.で示されるように、Project Name(プロジェクト名)とLocation(場所)を入力し、その後にOKをクリックしてください。

図1-1	8. プロジェクト生原	炗			
	MY SOFTWARE CO	OMPONENTS			?
DASHBOARD	 Application Middleware Driver System driver 		Add software component		Show dependencies C Show system drivers C Show hardware D
စန္ဒီ။		Atmel Start Importer		×	
	SCHEDULER ADC.0	New Atmel Start I	Project		
\odot		Project Name:	AQM_mega4808		
Ð		Location:		Browse	(×)
6		Solution:	Create New Solution	×	Ŭ
7	GENERAL	Solution Name:	AQM_mega4808		
	🖠 User gui	view Project Summary		Cancel	~
	💉 Rename com	ponent		Currect	~
	Remove com	ponent		<i>If you can't find your comp you have selected the corn component, and also corre the component you wish to</i>	<i>onent in the list make sure ect variant for this ct driver and/or mode for o select.</i>
	EEPROM3 CLICKBOAR	RD CONFIGURATION			C
	EXAMPLE CONFIGURATIO	<u>NN</u>			•
			GENERATE PROJECT		

・構成設定とコート
生成が完了されたのでAtmel STARTウィント
ウを閉じてください。

1.4. 応用コートと感知器トライバを追加

・ファームウェアは下のようにGitHubでダウンロード用に利用可能です。



GitHubで(Atmel Studio)コート 例を見てください。 貯蔵庫を閲覧するにはクリックしてください。

- ・ダウンロードしたファームウェアからapplicationとsensorのフォルダを複写してそれらをプロジェクトフォルダ内に配置してください。
- 両フォルダをプロジェクトに追加するため、Atmel StuidoでSolution Explorer(解決策エクスプローラ)へ行き、図1-19.で示されるように、Show All Files(全ファイルを表示)をクリックしてください。今や、前の段階で追加したapplicationとsensorのフォルダを見ることができます。図
 1-19.で示されるように、applicationフォルダを右クリックしてInclude in project(プロジェクトに含める)を選んでください。
- ・sensorフォルタを追加するためにこれらの手順を繰り返してください。

図1-1	19. 応月	月ファイル追加				
			Solu	tio	n Explorer 🛛 🝷 👎	х
			G	C	🖸 🐻 - 🗗 🔞 🔎 🚳 D	
			Sear	ch	Solution Explorer (Ctrl+;)	ρ
			4		AQM_mega4808	
					📴 Dependencies	11
					📴 Output Files	
				⊳	Libraries	
				⊳	.atmelstart	
		Open Folder			application	
	Ba	Copy Full Path			aqm_tasks	
		c u			display.c	
	1	Collapse		_	display.n	
		Add	•	·	i read.c	
		Scope to This		_	ead.n	
	Ē	New Solution Explorer View			send h	
				-	sleen.c	
		Include In Project		┛	sleep.h	
	¥	Cut	Ctrl+X		store.c	
	ŋ	Сору	Ctrl+C		store.h	
	×	Delete	Del		aqi_calculation.c	
	χ	Rename	F2		aqi_calculation.h	
	بو	Properties		-1) parameters.c	
	~	riopenies			j parameters.n	
				к Ь		
				Þ		
				Þ	credentials storage	
				Þ	cryptoauthlib	
				-	Debug	
				ь	n doc	Ŧ

1.5. Atmel START生成コートを変更

Atmel STARTが生成したAVR-IoT WGソースコートと周辺機能のAPIのいくつかは変更が必要です。必要とされる変更についてより多くの詳細を見つけるため、下のように、Atmel STARTによって生成されたAVR IoT WGソースコートとGitHubでダウンロート、用に利用可能なファームウェアの両方を比較することが必要です。表1-2.は変更の説明と共に変更が必要とされるファイルと関数を一覧にします。



GitHubで(Atmel Studio)コート 例を見てください。

貯蔵庫を閲覧するにはクリックしてください。

表1-2. Atmel START生成コートに対する変更の一覧

ソース ファイル	関数	変更説明
main.c	SendToCloud()	クラウト、ヘデータを送るのに専用作業が作成されます。従って、この関数はも はや不要です。
	main()	要素初期化とクラウド構成設定
		PIT初期化
	application_init()	表示器ON切り替え
application_manager.c		作成された応用作業
	MAIN_dataTask()	SendtoCloud()関数呼び出しなし
cryptoauthlib¥lib¥jwt¥atca_jwt.c	atca_jwt_finalize()	JWT作成中にハードウェアSHA機能を使うように切り替え(メモリ節約のため)

1.6. コンパイラ設定

本項はコード最適化のために必要とされるコンパイラ設定について説明します。

1.6.1. "Use subroutines for function prologue/epilogue"有効化

- ・プロジェクト名上で右クリックしてProperties(特性)を選んでください。
- Toolchain(ツールチェーン)タブでAVR/GNU C Compiler(AVR/GNU Cコンハ[°]イラ)下のGeneral(全般)をクリックしてください。図1-20.で示されるように、Use subroutines for function prologue/epilogue(プロローク)/エピローク)用関数にサブルーチンを使用)を許可してください。

図1-20. 7 [°] 미-	ーグ/エピローグ用関数	にサブルーチンを	使用の構成設定				
AQM_mega4808* 👳	× i2c_simple_master.h i	2c_simple_master.c*	application_manager.c	timeout.c	ccl.c	main.c	
Build Build Events	Configuration: Active (Debu	ıg) 🗸 Pla	tform: Active (AVR)	~			
Toolchain*	Configuration Manager						
Device							
Tool	AVR/GNU Common	AVR/GNU	C Compiler 🔿 General				
Packs	General	✓ Use su	ubroutines for function prologu	es and epilogues (-	mcall-prologu	es)	
Advanced	AVR/GNU C Compiler	Chang	ge the stack pointer without disa	abling interrupts (-r	nno-interrupts	s)	
	Preprocessor	🗹 Defau	lt char type is unsigned (-funsig	ned-char)			
	Symbols	🗹 Defau	lt bitfield type is unsigned (-fun	signed-bitfields)			

注:これはコードの大きさを最適化するのに必要とされます。

1.6.2. コンパイラ最適化設定

・プロジェクト名上で右クリックしてProperties(特性)を選んでください。

Toolchain(ツールチェーン)タブでAVR/GNU C Compiler(AVR/GNU Cコンハペイラ)下のOptimization(最適化)をクリックしてください。Optimization Level(最適化水準)の引き落としメニューからOptimize for size(大きさ最適化)を選んでください。

図1-21.コンハイ	7最適化設定							
Build Build Events	Configuration: Active (Debug)	V Platform: Active (AV	₹) ✓					
Toolchain*	Configuration Manager							
Device								
Tool	▲ Image: AVR/GNU Common AVR/GNU C Compiler ⇒ Optimization							
Packs	Output Files	Optimization Level:	Optimize for size (-Os) v					
Advanced	 AVR/GNU C Compiler General Preprocessor 	Other optimization flags:						
	Symbols	Prepare functions for gard	page collection (-ffunction-sections)					
	Optimization	✓ Prepare data for garbage	collection (-fdata-sections)					
	Debugging	Pack Structure members t	ogether (-fpack-struct)					
	Warnings Miscellaneous	Allocate only as many byt	es needed by enum types (-fshort-enums)					

2. MCCを使う室内空気品質監視器ファームウェア作成

本章はMCCでAVR-IoT WGソースコードを生成する手順と室内AQM応用ファーム 表2-1. ソフトウェア ツールと版番号 ウェアを作成ための手順の一覧を網羅します。

表2-1.はMCCを使ってファームウェアの作成に使われるソフトウェア ツールと版番号を一 覧にします。言及された、またはそれ以降の版のツールを使うことが推奨されま す。

ソフトウェア ツール	版番号
MPLAB [®] X	5.35版
AVR GCC	5.4.0版
avr8_gnu	3.6.2.1778版
MCC	3.95.0版
8ビットAVR MCUライフ・ラリ	2.1.0版
AVR-IoT WG感知器節点	1.2.0版
基礎サービス	0.1.34版
mikroElectronika Clickライブラリ	1.1.1版
ATmega_DFP	2.1.87版

2.1. 新規MPLAB[®] X7[°]ロシ^{*}ェクト作成

File(ファイル)⇒New Project(新規プロジェクト)へ行ってStandalone project(独立型プロジェクト)を選んでください。Device(デベイス)としてATm ega4808を、Compiler(コンパイラ)としてAVR (GCC Compiler)(AVR (GCCコンパイラ)を選んでください。プロジェクト名と場所詳細を入力して プロジェクトを作成してください。図2-1.で示されるように、新しいMPLAB Xプロジェクトの開始頁が現れます。



2.2. AVR-IoT WG感知器節点例プロジェクト生成

・ MPLAB XでMCCアイコンをクリックするか、またはTools(ツール)⇒Embedded(組み込み)⇒MPLAB X Code Configurator v3 Open/Close (MPLAB Xコード構成部第3版を開く/閉じる)をクリックすることによってMCCを開いてください。

図2-2. MCCを開く				
MPLAB X IDE v5.35 - AQM_mega4808 : default			•	– 8 ×
<u>File Edit View Navigate Source Refactor Production Debug Team</u>	<u>Tools</u> <u>W</u> indow <u>H</u> elp			Q Search (Ctrl+I)
👚 🛅 🔡 🍓 🔀 🐂 📋 🍤 🥙 default	Embedded >	MPLAB® Code Configurator v3: Open/Close) 🛍 🖬 🖄 🛍 🖓	How do I? Keyword(s)
Projects × Files Classes	Packs	Store x		
The set of the se	_emplates DTDs and XML Schemas Plugins Plugins Download		S	Міскоснір
	Options	F	LEARN & DISCOVER MY MPLAB®	XIDE WHAT'S NEW
	 Oreate New Import Legacy Import Prebuilt 	Recent Projects	Microchip Login	^
AQM_mega4808 - Dashboard × Navigator		AGM_v3_sps30 AGM_v3_sps30 AGM_w43_sps30 AGM_megu4808 amegu4808_air-quality-monitor-mplab	Aiready Registered? E-mail Address: Password:	New to Picrochip? Register now to enjoy the benefits of software update alerts and easy software purchasing, licensing and account viewing.
Construction Configuration: default Configuration: default Configuration: default	ور <u> </u>	S AQM_mega4808	LOGIN Forgot Password?	REGISTER NOW
CRC32: Hex file unavailable Packs	<			
ATmega_DFP (2.0.12) Gompler Toolchain Art R(v5.4.0) [C:Program Files (x86) Atmel Studio\7.0 toolch Production Image: Optimization:	Search Results × Output	at		
<	·			

- ・MCC構成設定ウィントウが開くまで待ってください。新しいプロジェクトのために一旦構成設定ウィントウが現れたなら、Save(保存)をックリックしてください。
- Device Resources(デバイス資源)部でInternet of Things(物のインターネット)タブを展開して ください。AVR-IoT WG Sensor Node(AVR-IoT WG感知器節点)例プロジェクトを追加 してください。
- 上の段階でVR-IoT WG Sensor Node(AVR-IoT WG感知器節点)がプロジェクトに追加 されます。

図2-3. AVR-IoT WG感知器節点		
Device Resources	۷	-
VREF		^
▼ Libraries		
 Š Foundation Services 		
► 🔵 LIN		
► 🛃 LED		
► 🗲 SWITCH		
 Mikro-E Clicks 		n
 Internet Of Things 		
 Examples 		
🛞 🛨 🔿 AVR-IoT WG Sensor Node		
 Libraries 		
 Examples 		Ų

2.3. 追加周辺機能構成設定

下の小項目は室内空気品質監視器応用に必要とされる他のMCC単位部/周辺機能の構成設定を説明します。

2.3.1. ADC

- ・SHT317ナロケ感知器は湿度と温度の測定に使われます。ADC周辺機能はこの感知器出力を読むように構成設定されます。ADC0周辺機能ト・ライハ、はAVR-IoT WG感知器節点例プロジェクトの一部としてプロジェクトに追加されています。もう一度プロジェクトにADC周辺機能ト・ライハを追加する必要はありません。
- ADCを構成設定するため、Project Resources(プロジェクト資源)タブ下のADC0周辺機 能をクリックしてください。

Projects	Files	Services	Resource M	anage ×				
Tree View	Flat View							
Project	Resource	Generat	e Import	Export				
 Internet 	t Of Things				^			
Libra	ries							
▼ Exam	ples							
0	🗙 🔿 AV	/R-IoT WG Se	nsor Node		6			
 Periphe 	rals							
🛞 🔀 📇 TWI0								
	🛞 🗙 🕀 ADC0							
	🖌 🕀 ADC	0			1			
	🔇 🕀 ADC	0			1			
9 19 19 19 19	🖌 🕀 ADC 🕻 📸 RTC 🄇 🚍 USAI	0 RT2						

- ・ADC構成設定ウィントウが開きます。
- ・ADCの採取累積機能が室内AQM応用に使われます。Hardware Settings(ハードウェア設定)部でSample Accumulation Number(採取 累積数)をAccumulate 8 samples(8採取累積)に構成設定してください。

図2-5. ADC0構成設定	
ADC0	
않 Easy Setup 📃 Registers	
Result Selection :	10-bit mode
 Hardware Settings 	
Enable ADC:	\checkmark
Sampling Frequency(Hz):	14204 ≤ 48076 ≤ 48076
ADC Clock(Hz):	625000
Sample Accumulation Number:	Accumulate 8 samples
Sample Length (# of ADC Clock) :	0 ≤ 0 ≤ 31
Voltage Reference :	VDD -

注: AQMハートウェアは2つの温度感知器を持ます。1つはアナログ感知器(SHT31-ARP)で、別の1つは(AVR-IoT WG基板)の基板上 MCP9808デジタル感知器です。AQM応用は温度監視にMCP9808デジタル感知器を使います。

2.3.2. ピン管理部

・SHT31感知器の温度と湿度の出力ピンは各々、PD6とPD7のポートピンに接続されます。

・ADCチャネルとして、PD6とPD7のポートピンを構成設定するため、Pin Manager Grid View(ピン管理部格子表示)へ行ってください。PD6 とPD7をADC0用に構成設定してください。光感知器出力に接続されるPD5ピンはAVR-IoT WG感知器節点例プロジェクトによって予 めADCチャネルとして構成設定されています。光感知器が室内AQM応用で使われないのでADCチャネルとしてのPIn PD5 configuratio n(PD5ピン構成設定)を非選択にしてください。

Search Resul	ts (Output	Notification	ns (M	ICC]		Pin M	lana	ger:	Grid	View	×																		
Package:	QFP32	-	Pin No:	30	31	32	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	20	21	22	23	24	25	26
							Port	A 🔻					Port	c 🔻					Port	D 🔻						Po	ort F	•		
Module	Fun	ction	Direction	0	1	2	3	4	5	6	7	0	1	2	3	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6
ADC0	AINx		input													î	î	î.	î.	î.	ĵ.	ô	ô			î	î.	î	î	
	CLKI		input	î																										
	CLKO		output								î																			
	TOSC1		input																					ĵ.						
	TOSC2		input																						ĵ.					
	GPIO		input	î	î.	î.	î	î	î	î.	ì	î.	î.	ĵ,	ĵ,	î	î	î.	î.	î.	î.	î.	î.	î.	î	ô	â	â	â	Ê
Pin Module 🔻	GPIO		output	î	â	î.	î	î	î	î.	ì	î.	î.	ĵ,	î.	â	ô	â	â	î	î	î	î	î.	î	î	î.	î	î	î.
RSTCTRL	RESET		input																											î
	MISO		input						A				2																	_

© 2020 Microchip Technology Inc.

2.3.3. UART1

- ・SPS30 PM2.5感知器はUARTインターフェースを通してマイクロコントローラと通信します。
- ・UARTを構成設定するため、Device Resources(デバイス資源)部からUSART1トライバを 追加してください。Device Resources(デバイス資源)⇒Peripherals(周辺機能)⇒USAR
 T⇒USART1を通して誘導してAdd(追加)シンボルをクリックしてください。

図2-7. USART1周辺機能選択



- ・USART1構成設定ウィントウが現れます。
- Software Settings(ソフトウェア設定)部でInterrupt Driven(割り込み駆動)任意選択を許可してください。SPS30感知器が115200のボーレートを支援するためBaud Rate(ボーレート)を115200に設定してください。

図2-8. USART1構成設定	
USART1	
🔯 Easy Setup 📄 Registers	
Software Settings	
API Prefix:	USART1
Interrupt Driven:	
RX Buffer Size (Bytes):	8 ~
TX Buffer Size (Bytes):	8 💌
Printf support:	
 Hardware Settings 	
Mode:	Async Mode 👻
Baud Rate:	1 ≤ 115200 ≤ 1000000
Error Percent:	-0.064%
Enable USART Receiver:	\checkmark
Enable USART Transmitter:	\checkmark
Parity Mode:	No Parity -
Stop Bit Mode:	1 stop bit 🔹
Character Size:	Character size: 8 bit

2.3.4. NVM

- ・内部EEPROMは既定要素と外部EEPROMの指標の格納に使われます。内部EEPR OMはNVM制御周辺機能を使ってアクセスされます。
- NVM制御を構成設定するため、Device Resources(デバイス資源)部からNVM control (NVM制御)トライバを追加してください。Device Resources(デバイス資源)⇒Peripherals (周辺機能)⇒NVMCTRL(NVM制御器)を通して誘導してAdd(追加)シンボルをクリックし てください。

図2-9. NVM制御周辺機能選択



- ・NVMCTRL構成設定ウィンドウが開きます。
- ・Software Settings(ソフトウェア設定)部でAPI Prefix(API接頭辞)をNVMに構成設定してください。

図2-10. NVM制御構成設定

NVMCTRL	
🔯 Easy Setup 📃 Registers	
 Software Settings 	
API Prefix:	NVM
Place Flash Functions in Separate Segment:	
 Hardware Settings 	
Enable Application Code Write Protect: Enable Boot Lock:	
► Interrupt Settings	

2.3.5. 事象システム

- ・AVR-IoT WG基板上の基板上切替器はOLED表示器をONに切り替えるのとOLEDで表示される要素を変更するのに使われま す。AVR-IoT WG基板上のSW0基板上切替器はマイクロコントローラのPORT1_PIN6に接続されます。マイクロコントローラの事象システムと CCL周辺機能を使い、どんなソフトウェア介在もなしに切替器跳ね返り制御機構が実装されます。
- ・事象システムを構成設定するため、Device Resources(デハイス資源)部からEvent System(事象システム)トライハを追加してください。Device Resources(デハイス資源)⇒Peripherals(周辺機能)⇒EVSYS(事象システム)を通して誘導してAdd(追加)シンボルをクリックしてください。
 EVSYSウイントウが開きます。
- ・事象チャネル1は周期割り込み計時器出力をCCLのLUT1に接続するの使われ、事象チャネル4は切替器ポートピンをCCLのLUT1に接続するのに使われます。
- ・図2-11.で示されるように、チャネル1に対してEvent Generator(事象生成部)をRTC_PIT3として、Event User(事象使用部)をCCLLUT1 Aとして構成設定してください。
- ・図2-11.で示されるように、チャネル4に対してEvent Generator(事象生成部)をPORT1_PIN6として、Event User(事象使用部)をCCLLU T1Bとして構成設定してください。

図2-11. 事象システム構成設定

E	VSYS															
4	🕃 Easy Setup 📃 Registers															
	 Software Settings 															
	API Prefix:	EVSYS														
	 Event System Settings 															
Н	Event Generator	Channels												Event	Users	
			ADC0	CCLLUT0/	A CCLLUTOE	B CCLLUT1A	CCLLUT1	B CCLLUT2/	A CCLLUT2E	CCLLUT3/	A CCLLUT3E	EVOUTA	EVOUTB	EVOUTC	EVOUTD	EVOUTE
	OFF *	CHANNEL0														
	RTC_PIT3 -	CHANNEL1				\checkmark										
	OFF •	CHANNEL2														
	OFF -	CHANNEL3														
	PORT1_PIN6 -	CHANNEL4					\checkmark									
	OFF -	CHANNEL5														
l																

注: CCL周辺機能の読み込みためにLoad peripheral CCL(CCL周辺機能読み込み)警告が現れます。CCL周辺機能読み込みに ついては次の小項目が説明するので、この警告を無視してください。

2.3.6. CCL

- ・切替器跳ね返り制御を実装するため、濾波器任意選択と共にCCL周辺機能のLUT1が使われます。
- ・CCL周辺機能は各切替器押下事象に対して割り込みを生成します。
- ・ CCL周辺機能を構成設定するため、Device Resources(デバイス資源)部からCCL(構成設定可能な注文論理回路)ドライバを追加して ください。Device Resources(デバイス資源)⇒Peripherals(周辺機能)⇒CCLを通して誘導してAdd(追加)シンボルをクリックしてください。
- ・CCL構成設定ウィンドウが開きます。
- Interrupt Settings(割り込み設定)ウィンドウでLUT1-OUT Interrupt(LUT1-OUT割り込み)任意選択の引き落としメニューからSense rising edge(上昇端感知)を選んでください。

図2	!-12. LUT1割り込み構成設	定	
СС	L		
6	Easy Setup 🗮 Registers		
	 Hardware Settings 		
E	Enable CCL:	\checkmark	
	 Interrupt Settings 		
l	UT0-OUT Interrupt:	Interrupt disabled	
l	UT1-OUT Interrupt:	Sense rising edge	
l	UT2-OUT Interrupt:	Interrupt disabled	
ι	UT3-OUT Interrupt:	Interrupt disabled -	
	 CCL Settings 		
			\uparrow

・CCL Setting(CCL設定)部でLUT1をクリックすることによってLUT1を選んでください。

図2-13. LUT1選択	5		
CCL			
😳 Easy Setup 📃 Registers			
LUT3-OUT Interrupt:	Interrupt disabled		• ^
▼ CCL Settings			
LUT1-IN0 LUT1-IN1 LUT1-IN2	LUT1	LUT1-OUT	

- ・図2-14.で示されるように、LUT1を許可するため、LUT1 Configuration(LUT1構成設定)部でEnable LUT(LUT1許可)任意選択を チェックしてください。
- ・図2-14.で示されるように、LUT1-IN1任意選択の引き落としメニューからEVENTB(事象B)を選んでください。PORT1_PIN6が事象チャネ ル4を通してEVENTBに接続されるため、この切替器信号はLUT1への最初の入力として働きます。
- ・図2-14.で示されるように、LUT1-IN2任意選択の引き落としメニューからEVENTA(事象B)を選んでください。RTC_PIT3が事象チャネル1 を通してEVENTAに接続されるため、これはCCL濾波器へクロック信号を提供します。

义]2-14. LUT1構成設	定			
4	CCL			•	
ſ	🔅 Easy Setup 📄 Registers				
	< (>	Î Î
			LUT 1 Configuration		
			LUT Enable		
	Enable LUT: 😲	\checkmark			
			Inputs and Outputs		
	LUT-IN0: 😲	MASK		•	h
	LUT-IN1: 😯	EVENTB		•	
	LUT-IN2: 😮	EVENTA		-	U,
Ľ	Enable LUT-OUT: 😮				
			Additional Configuration		
L	Filter Options: 🕐	DISABLE		-	~

- ・図2-15.で示されるように、CCL構成設定ウィンドウのAdditional Configuration(追加構成設定)部で濾波器を許可するためにFilter Options(濾波器任意選択)をFILTER(濾波器)に構成設定してください。
- ・ Clock Selection(クロック任意選択)をIN2に構成設定してください。

义	2-15. CCL LUTTC	の濾波器動作	F許可とクロック選択	
C	CL			?
Ę	🕃 Easy Setup 📄 Registers			
	LUT-IN1: 😮	EVENTB		•
	LUT-IN2: 😲	MASK		-
	Enable LUT-OUT: 😮			
			Additional Configuration	
I	Filter Options: 😮	FILTER		-
I	Enable Edge Detector:			
	Clock Selection: 😗	IN2		-
			Truth Table	
	Custom			-
	IN2 IN1	INO	TUO	
				v

・マイクロコントローラは周期的にスタンハイ休止動作へ行きます。マイクロコントローラが スタンハイ休止動作の間に切替器押下事象を検出するため、CCLはスタンハイ 休止動作で動かなければなりません。CCL構成設定ウィントウのRegisters(レ ジスタ)タブでCTRLAレジスタ下のRUNSTDBY(スタンハイ動作で走行)任意選択 をenabled(許可)に構成設定してください。

図2-16. スタンバイで走行ビット構成設定

CCL
හි Easy Setup 📃 Registers
▼ CCL
Interrupt Enables
V INT1
INT2
INTO INTO
INT3
▼ Register: CTRLA 0x41
ENABLE enabled
RUNSTDBY enabled
▼ Register: INTCTRL0 0x4

 LUT1CTRLCレジスタでINSEL2をEvent input source A(事象 入力元A)として構成設定してください。INSEL2はクロック選択後 にこれの構成設定が遮蔽入力に変更されるため、INSEL2が 再び構成設定されます。

図2-17. LUT1入力2構成設定

CCL							
🔅 Easy Setup 📃 Registers							
OUTEN disabled							
INSEL0 Masked input							
INSEL1 Event input source B							
▼ Register: LUT1CTRLC 0x3							
INSEL2 Event input source A							
▼ Register: LUT2CTRLA 0x0							
CLKSRC CLK_PER is clocking the LUT							
EDGEDET Edge detector is disabled							

・CCL構成設定ウィントウのEASY Setup(簡単設定)タブでTruth Table(真理値表)を開き、それをNORに構成設定してください。

义;	図2-18. 真理値表構成設定						
c	CCL						
5	🐉 Easy Setup 📄	Registers					
	Enable Edge D	etector:			^		
	Clock Selection	: 😗 🛛 🛛 IN	2	·			
				Truth Table			
	NOR			*			
	IN2	IN1	INU	OUI			
	0 0	IN1 0	0 0	001			
	0 0	0 0	0 1	001 			
	0 0 0	0 0 1	0 1 0	OUI			
	0 0 0 0 0	0 0 1 1	0 1 0 1	OUT			
	0 0 0 0 0 1	0 0 1 1 0	0 1 0 1 0 1 0	OUT			
	IN2 0 0 0 0 1	IN1 0 1 0 0	INU 0 1 0 1 0 1 0 1 0 1	OUI 1 0 0 0 0 0			
	IN2 0 0 0 0 1 1	IN1 0 1 0 0 1 0 0 1 0 1 0 1 1 0 1	INU 0 1 0 1 0 1 0 1 0 1 0	OUT Image: Im			
	IN2 0 0 0 1 1 1	IN1 0 1 1 0 0 1 1 1 1	INU 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1	OUI 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0			

・前の全ての構成設定完了後、LUT1は図2-19.で示されるように見えます。



2.3.7. 休止動作

- ・マイクロコントローラは全ての作業の完了後にスタンバイ休止動作へ行きます。
- 休止動作を構成設定するため、Project Resources(プロジェクト資源)部でSystem Module(システム単位部)をクリックしてください。System Module(システム単位部)ウィン ドウが開きます。
- ・図2-20.で示されるように、Registers(レジスタ)タブをクリックしてください。 注: Registers(レジスタ)タブは開くのにしばらくかかるかもしれません。
- SLPCTRL.CTRLAレジスタへ行き、SEN(休止許可)をenabled(許可)に、SMODE (休止動作形態)をStandby Mode(スタンハイ動作)に構成設定することによって休 止動作を許可してください。

図2-20. スタンバイ休止動作構成設定 System Module				
Easy Setup 📃 Registers				
SWRE disabled +				
▼ SLPCTRL				
▼ Register: SLPCTRL.CTRLA 0x3				
SEN enabled				
SMODE Standby Mode				
▼ WDT				
▼ Register: WDT.CTRLA 0x0				

2.3.8. PIT

- マイクロコントローラは周期割り込み計時器(PIT)割り込みを得る時に必ず周期的に休止動作から起き上がります。PITはRTC単位部の 一部です。PITを構成設定するため、Project Resources(プロジェクト資源)⇒Peripheral(周辺機能)⇒RTCを通して誘導することによっ てProject Resources(プロジェクト資源)からRTCを選んでください。
- Periodic Interrupt Timer(周期割り込み計時器)部でPIT Enable(PIT許可)任意選択を許可してください。Period Selection(周期選択)をRTC Clock cycle 2048(2048 RTCクロック周期)に構成設定してください。これはRTCクロックを2048分周します。
- ・Periodic Interrupt Enable(周期割り込み許可)任意選択を許可してください。

図2−21. PIT構成設定				
RTC		()		
😳 Easy Setup 📃 Registers				
RTC Clock(Hz):	32000	^		
RTC Clock Source Selection:	Internal 1kHz OSC 🗸]		
External Clock(Hz):	1 ≤ 32000 ≤ 32000			
Prescaling Factor:	RTC Clock / 1 -]		
Compare:	1 s ≤ 0 s ≤ 3 s			
Actual Compare:	0 s			
Period:	1 s ≤ 2.048 s ≤ 3 s			
Actual Period:	2.048 s			
▼ Periodic Interrupt Timer				
PIT Enable:	\checkmark			
Period Selection:	RTC Clock Cycles 2048 🗸			
Periodic interrupt Enable:	\checkmark			
► Interrupt Settings				

2.3.9. EEPROM 3 Click

空気品質感知器読み取り値はEEPROM 3 Click基板上に存在する外部EEPROMに格納 されます。EEPROM 3 Click基板はMCC枠組みによって支援されます。Device Resource s(デバイス資源)部からEEPROM3 Clickドライバをプロジェクトに含めてください。Device Resou rces(デバイス資源)→Mikro-E Clicks(MikroElektronika Click)→Storage(記憶装置)→EEP ROM3を通して誘導してAdd(追加)シンボルをクリックしてください。

・ EEPROM 3 Click構成設定ウィンドウが開きます。 EEPROM 3 Clickに必要とされる追加の 構成設定はありません。

図2-22. EEPROM 3 Click追加



2.4. MCCコート ファイル作成

- ・ Resouce Manager(資源管理部)⇒Project Resources(プロジェクト資源)ウィントウの左手側にあるGenerate(生成)釦をクリックし、コート、生成が完了するのを待ってください。
- 注: この構成設定が警告を持つことを示すポップアップが現れるかもしれません。全ての警告が処置済みなので続けるためにYESをク リックしてください。

2.5. 応用コート、と感知器ト、ライハを追加

・ファームウェアは下のようにGitHubでダウンロード用に利用可能です。



GitHubで(MPLAB[®] X)コード例を見てください。

貯蔵庫を閲覧するにはクリックしてください。

- ・ダウンロードしたファームウェアからapplicationとsensorのフォルダを複写してそれらをプロジェクトフォルダ内に配置してください。
- ・両フォルダをプロジェクトに追加するため、Project(プロジェクト)ウィントウへ行き、Source Files(ソースファイル)フォルダを選択してください。適応メ ニューを開くために右クリックしてAdd Existing Items frrom folder(フォルダから既存項目を追加)を選んでください。Add folder(フォルダ追 加)をクリックしてください。応用コートを含む複写したばかりのapplicationフォルダを選び、その後にselect(選択)任意選択をクリックしてくだ さい。感知器トライバを含むsensorフォルダを追加するためにこれらの手順を繰り返してください。両フォルダ選択後、Add(追加)釦をクリッ クしてください。
- ・プロジェクトへのフォルダ追加のより多くの詳細な手順についてはMicrochip開発者手助けウェブサイトの頁を参照してください。

2.6. MCC生成コートを変更

MCCが生成したAVR-IoT WGソースコートと周辺機能のAPIのいくつかは変更が必要です。変更についてより多くの詳細を見つけるため、以降のように、MCCによって生成されたAVR IoT WGソースコートとGitHubでダウンロート、用に利用可能なファームウェアの両方を比較することが必要です。表2-2.は変更の説明と共に変更が必要とされるファイルと関数を一覧にします。



GitHubで(MPLAB[®] X)コート、例を見てください。 貯蔵庫を閲覧するにはクリックしてください。

長2-2. MCC生成コートに対する変更の一覧				
ソース ファイル	関数	変更説明		
main.c	SendToCloud()	クラウト、ヘデータを送るのに専用作業が作成されます。従って、この 関数はもはや不要です。		
	main()	要素初期化とクラウド構成設定		
mcc_generated_files/src/ccl.c	ISR(CCL_CCL_vect)	この割り込みは切替器押下で起きます。		
		PIT ISR呼び戻し関数設定		
mcc_generated_files/src/rtc.c	RTC_Initialize()	全レジスタ同期待機		
	ISR(RTC_PIT_vect)	最後でPIT7ラゲ解消		
	application init()	表示器をONに切り替え		
mcc_generated_files/	application_init()	作成された応用作業		
application_manager.c	MAIN_dataTask()	SendtoCloud()関数呼び出しなし		
mcc_generated_files/ cryptoauthlib/lib/jwt/atca_jwt.c	atca_jwt_finalize()	更新されたatcab_hw_sha2_256()関数はJWT作成中にハートウェア SHA機能を使います(メモリ節約のため)。		
mcc_generated_files/cloud/ crypto_client/crypto_client.c		ATCA_PRINTF定義を注釈化(メモリ節約のため)		
mcc_generated_file/ config/cryptoauthlib_config.h		ATCA_PRINTF定義を注釈化(メモリ節約のため)		
mcc_generated_file/EEPROM3_driver.c		MCCが生成したI2Cドライバによって変更されたI2C関数名		
	I2C_0_wake_up()			
mcc_generated_files/	hal_i2c_writeNBytes()	MCCが生成したI ² Cドライハ゛によって変更されたI2C_BUSY関数名		
cryptoautino/ no/ nai/ nai_ize.c	hal_i2c_readNBytes()			

2.7. コンパ[°]イラ設定

本項はコード最適化のために必要とされるコンパイラ設定を説明します。

2.7.1. "Use subroutines for function prologue/epilogue"有効化

プロジェクト名上で右クリックしてProperties(特性)を選んでください。AVR GCC (Global Oprions)(AVR GCC(全般任意選択))下のavr-g ccをクリックしてください。Use subroutines for function prologue/epilogue(プロローグ/エピローグ)用関数にサブルーチンを使用)を許可してください。

図2-23. プロローグ/エピローグ用関数にサブルーチンを使用の構成設定					
😵 Project Properties - AQM_mega4808	>				
Categories: General File Indusion/Exclusion Conf: [default]	Options for avr-gcc (v5.4.0) Option categories: General				
·····	Use subroutines for function prologues and epilogues				
····· • Libraries	Change the stack pointer without disabling interrupts				
Building Avr GCC (Global Options)	Default char type is unsigned				
avr-as	Default bitfield type is unsigned				
····· ○ avr-as-pre ····· ○ avr-gcc ····· ○ avr-g++ ····· ○ avr-ld ····· ○ avr-ar					

・Apply(適用)そしてその後にOKをクリックしてください。

注:これはコードの大きさを最適化するのに必要とされます。

2.7.2. コンパイラ最適化設定

・応用ファームウュアは効率的にコードメモリを使うため、大きさに対して最適化されます。プロジェクト名上で右クリックしてProperties(特性)を選んでください。AVR GCC (Global Oprions)(AVR GCC(全般任意選択))下のavr-gccをクリックしてください。Option categories(任意選択区分)の引き落としメニューからOptimization(最適化)を選んでください。optimization-level(最適化水準)を"s"に設定してください。

図2-24. コンパイラ最適化設定					
😵 Project Properties - AQM_mega4808 X					
Categories: General File Indusion/Exclusion Conf. [default]	Options for avr-gcc (v5.4.0) Option categories: Optimization ~	Reset			
······ · Loading	optimization-level	s ~			
···· • Libraries	Prepare functions for garbage collections				
O Building Avr GCC (Global Options)	Prepare data for garbage collections				
····· • avr-as	Pack Structure members together				
····· O avr-as-pre	Allocate only as many bytes needed by enum types				
····· • avr-gcc	Use rjmp/rcall (limited range) on >8K devices				
···· • avr-ld					
avr-ar					

・Apply(適用)そしてその後にOKをクリックしてください。

3. 応用書き込み

3.1. Atmel Studioで書き込み

- ・デバイスに書き込むため、マイクロUSBケーブルを使ってAVR-IoT WG基板をPCに接続してください。
- ・Start without Debugging(デバッグなしで開始)(Ctrl+Alt+F5)アイコンをクリックしてください。

図3-1.	デバッグなしで開始					
AQM	mega4808 - AtmelStudio	Standard Mode 🔻	Quick Launch (Ctrl+Q)	. 8 ×		
File Edit	The File View Water And Action Particle United States And Action Participation (1997)					
inc cure				- 1		
:	🛛 📅 📲 🎽 📲 📲 📲 🕷 🗗 🔠 🛛 🖓 🐨 🦿 📓 🕎 🐨 🖓 Debug 🔹 Debug Browser * 💦 👘 AqmPrinter	•	M N C 🖬 🐝 M 🗆 📲 📲	± 1 3 3 ÷		
) 🕅 🗗	= -> = -> -> -> 🔹 -> 🔹 -> -= -= Armega4808 🦹 UPDI on nEDBG (ATML320308	31800012574) 🖕				
main.c 👳	×	+	Solution Explorer	- ₽ ×		
→ main.c	 C:\Users\i20934\Desktop\aqm_studio_mega4808\AQM_mega4808\AQM_mega4808\AQM_mega4808\main.c 	- ኛ Go	0 0 A 10 - 7 B 🔑 🖊 🗕	🚳 D		
1	i/*	÷	Search Solution Explorer (Ctrl+1)	0		
2	\file main.c		Search Solution Explorer (Cur+,)			
3			Application	^		
4	\brief Main source file.		Cli			
5	(c) 1918 Microschip Tashpalamu Tas, and its subsidianias		Cloud			
7	(c) zers microchip recimology file, and its substationes.		Config			
8	Subject to your compliance with these terms, you may use Microchip software and any		Credentials_storage			
9	derivatives exclusively with Microchip products. It is your responsibility to comply with third party		Cryptoauthlib			
10	license terms applicable to your use of third party software (including open source software) that		doxygen			
11	may accompany Microchip software.		examples			
12			Include			
13	THIS SOFTWARE IS SUPPLIED BY MICROCHIP "AS IS". NO WARRANTIES, WHETHER		Image:			
14	EXPRESS, IMPLIED OR STATUTORY, APPLY TO THIS SOFTWARE, INCLUDING ANY		Image: Sensors			
15	IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, AND FITNESS		Image: Second			
16	FOR A PARTICULAR PURPOSE.		I utils			
17			b inc			
10	IN NO EVENT WILL MILLOUGHTP BE LIABLE FOR ANY INDIRECT, SPECIAL PONITIVE,		application manager.c			
20	INCLUENTAL OR CONSEQUENTIAL COSS, DARAGE, COST OR EARENSE OF ANT ALTO		application manager.h			
21	HAS BEEN ADVISED OF THE POSSIBILITY OR THE DAMAGES ARE FORESEABLE. TO		atmel start.c			
22	THE FULLEST EXTENT ALLOWED BY LAW, MICROCHIP'S TOTAL LIABILITY ON ALL		atmel start b			
23	CLAIMS IN ANY WAY RELATED TO THIS SOFTWARE WILL NOT EXCEED THE AMOUNT		b hanner h			
24	OF FEES, IF ANY, THAT YOU HAVE PAID DIRECTLY TO MICROCHIP FOR THIS		G debug print c			
25	25 SOFTWARE.					
26	L*/		debug_printin			
27			lad a			
28	#include <stdio.h></stdio.h>		C led.c			
29	#include <string.h></string.h>		in lea.n			
30	#Include Statio.n>		c main.c	_		
32	#include "lad b"	-	sensors_handling.c			
100 % -		•	h sensors_handling.h			
Immediate \	Vindow Find Results 1 Error List Output					
Ready			Col 1 Ch 1	INS		

3.2. MPLAB[®] X IDEで書き込み

- ・デバイスに書き込むため、マイクロUSBケーブルを使ってAVR-IoT WG基板をPCに接続してください。
- ・Make and Program Device(作成してデバイスに書き込み)アイコンをクリックしてください。

図3-2. 作成してデバイスに書き込み				
MPLAB X IDE v5.35 - AQM_mega4808 : default				– 0 X
<u>File Edit View Navigate Source Refactor Production Debug Team Too</u>	ls <u>W</u> indow <u>H</u> elp			Q-
🗄 🎦 🔒 🍓 🔀 🐂 💼 🦻 🥐 🖉 default	🔍 🕆 🖓 • 🕅 • 🛣	🔁 • 🗟 🚯 • 🔳 🔘 🥥	💿 💼 🖬 🏚 🗐 📲 📲 🚥 🍬	+ How do IP Keyword(s)
Projects × Files Classes	Start Page 🗙 🛒 MPLAB X Store	×		
AQW_mega\$808 Age = Ag	MPLAB X IDE		LEARN & DISCOVER MY MPLAB®)	AICROCHIP
	Oreate new Import Legacy Import Prebuilt	Recent Projects	Microchip Login Already Registered? E-mail Address:	A New to Microchip? Redister now to enjoy the benefits
AQM_mega4808 - Dashboard × Navigator % Yg AQM_mega4808 % Yg AQM_mega4808 % Yroject Type: Application - Configuration: default % Yo %	DATA SHEETS	ACM_v2_sps30 ACM_mega4808_air-quality-monitor-mplab processing_test atmega4808-air-quality-monitor-mplab ACM_mega4808	Password: LOGIN Forgot Password?	region new solvy of a contract of software update alerts and easy software purchasing, leensing and account viewing.
Packs	٢			>
ATmega_DFP (2.0.12)	Search Results × Output			
AVR. (v5.4.0) [C:\Program Files (x86)\Atmel\Studio\7.0\toold Production Image: Optimization: Data 6,144 (0x1800) bytes Data Used: 0 (0x0) Free: 6,144 (0x1800) Program 49,152 (0xC000) bytes C				

<mark>4</mark>. 結び

この文書はAVR-IoT WGソースコートの先頭での室内AQM応用ファームウェア作成の詳細を網羅します。この文書はAtmel STARTとMCC でAVR-IoT WGソースコートをどう生成するかの手順も説明します。

5. 改訂履歴

資料改訂	日付	注釈
А	2020年3月	初版文書公開

Microchipウェフ゛サイト

Microchipはhttp://www.microchip.com/で当社のウェブサ小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のに使われます。利用可能な情報のいくつかは以下を含みます。

- ・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設計協力課程会員一覧
- ・Microshipの事業 製品選択器と注文の手引き、最新Microchip報道発表、セミナーとイベントの一覧、Microchip営業所の一覧、代理 店と代表する工場

製品変更通知サービス

Microchipの製品変更通知サービスは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/pcnへ行っ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
- ・組み込み解決技術者(ESE:Embedded Solutions 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ESE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トの安全を保証することはできません。コート、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、Adaptec、AnyRate、AVR、AVRロゴ、AVR Freaks、BesTime、BitCloud、chipKIT、chipKITロ ゴ、CryptoMemory、CryptoRF、dsPIC、FlashFlex、flexPWR、HELDO、IGLOO、JukeBlox、KeeLoq、Kleer、LANCheck、LinkMD、 maXStylus、maXTouch、MediaLB、megaAVR、Microsemi、Microsemiロゴ、MOST、MOSTロゴ、MPLAB、OptoLyzer、PackeTime、PI C、picoPower、PICSTART、PIC32ロゴ、PolarFire、Prochip Designer、QTouch、SAM-BA、SenGenuity、SpyNIC、SST、SSTロゴ、Super Flash、Symmetricom、SyncServer、Tachyon、TempTracker、TimeSource、tinyAVR、UNI/O、Vectron、XMEGAは米国と他の国に於 けるMicrochip Technology Incor poratedの登録商標です。

APT、ClockWorks、The Embedded Control Solutions Company、EtherSynch、FlashTec、Hyper Speed Control、HyperLight Load、 IntelliMOS、Libero、motorBench、mTouch、Powermite 3、Precision Edge、ProASIC、ProASIC Plus、ProASIC Plusロゴ、Quiet-Wire、 SmartFusion、SyncWorld、Temux、TimeCesium、TimeHub、TimePictra、TimeProvider、Vite、WinPath、ZL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、BlueSky、BodyCom、CodeGuard、 CryptoAuthentication、CryptoCompanion、CryptoController、dsPICDEM、dsPICDEM.net、Dynamic Average Matching、DAM、EC AN、EtherGREEN、In-Circuit Serial Programming、ICSP、INICnet、Inter-Chip Connectivity、JitterBlocker、KleerNet、KleerNet¤ ゴ、memBrain、Mindi、MiWi、MPASM、MPF、MPLAB Certified¤ゴ、MPLAB、MPLINK、MultiTRAK、NetDetach、Omniscient Code Generation、PICDEM、PICDEM.net、PICkit、PICtail、PowerSmart、PureSilicon、QMatrix、REALICE、Ripple Blocker、SAM-ICE、Se rial Quad I/O、SMART-I.S.、SQI、SuperSwitcher、SuperSwitcher II、Total Endurance、TSHARC、USBCheck、VariSense、View Sens e、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。

Adaptec^{ロコ}、Frequency on Demand、Silicon Storage Technology、Symmcom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。

© 2020年、Microchip Technology Incorporated、米国印刷、不許複製

品質管理システム

Microchipの品質管理システムに関する情報についてはhttp://www.microchip.com/qualityを訪ねてください。

日本語© HERO 2020.

本応用記述はMicrochipのAN3417応用記述(DS00003417A-2020年3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。



米国

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: http://www.microchip.com 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

オーストラリア - シト・ニー Tel: 61-2-9868-6733 中国 - 北京 Tel: 86-10-8569-7000 中国 - 成都 Tel: 86-28-8665-5511 中国 - 重慶 Tel: 86-23-8980-9588 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 中国 - 香港特別行政区 Tel: 852–2943–5100 中国 - 南京 Tel: 86-25-8473-2460 中国 - 青島 Tel: 86-532-8502-7355 中国 - 上海 Tel: 86-21-3326-8000 中国 - 瀋陽 Tel: 86-24-2334-2829 中国 - 深圳 Tel: 86-755-8864-2200 中国 – 蘇州 Tel: 86-186-6233-1526 中国 - 武漢 Tel: 86-27-5980-5300 中国 - 西安 Tel: 86-29-8833-7252 中国 - 廈門 Tel: 86-592-2388138 中国 - 珠海 Tel: 86-756-3210040

亜細亜/太平洋

亜細亜/太平洋 イント・- ハンカ・ロール

Tel: 91-80-3090-4444 イント - ニューデリー Tel: 91-11-4160-8631 イント・フネー Tel: 91-20-4121-0141 日本 - 大阪 Tel: 81-6-6152-7160 日本 - 東京 Tel: 81-3-6880-3770 韓国 - 大邱 Tel: 82-53-744-4301 韓国 - ソウル Tel: 82-2-554-7200 マレーシア – クアラルンプール Tel: 60-3-7651-7906 マレーシア ー ヘ・ナン Tel: 60-4-227-8870 フィリピン ー マニラ Tel: 63-2-634-9065 シンガポール Tel: 65-6334-8870 台湾 - 新竹 Tel: 886-3-577-8366 台湾 - 高雄 Tel: 886-7-213-7830 台湾 - 台北 Tel: 886-2-2508-8600 タイ ー バンコク Tel: 66-2-694-1351 ベトナム ー ホーチミン Tel: 84-28-5448-2100

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 トイツ – ガルヒング Tel: 49-8931-9700 ドイツ – ハーン Tel: 49-2129-3766400 トイツ - ハイルブロン Tel: 49-7131-72400 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 オランダーデルーネン Tel: 31-416-690399 Fax: 31-416-690340 ノルウェー - トロンハイム Tel: 47-72884388 ホ[°]ーラント[゛]ー 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 - マドリート 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 - ウォーキンガム Tel: 44-118-921-5800 Fax: 44-118-921-5820

Fax: 905-695-2078