1.	新規プログラムの作成

 Ardruinoの接続 USBで接続します。

(2) eclipseの起動 _eclipseをダブルクリック(よく使うようであれば、デスクトップにショートカットを作成してください。)

		(2)				_		
GO - 1961-9-	- • A0	cer (C:) • test • arc	iuino 🖡 pielades 🖡 ecil	ipse 🕨			_	• +• eclipse0) p
整理 ▼ 💼 開く 書き返	797	新しいフォルダー						i 🗸 🖬 🕡
☆ お気に入り	^	名前	^	更新日時		種類	サイズ	^ ^
▲ ダウンロード		.eclipseproduct		2014/08/15 2	21:31	ECLIPSEPRODU	1 KB	
■ デスクトップ		artifacts.xml		2014/08/15 2	21:45	XML ドキュメント	114 KB	
		eclipse.exe		2014/08/15 2	21:31	アプリケーション	312 KB	
一般近表示した場所		🚳 eclipse.exe -clea	n.cmd	2014/08/15 2	21:31	Windows コマン	1 KB	_
		eclipse.ini	種類: Windows コマンド	・スクリプト	1:45	構成設定	1 KB	E
📃 デスクトップ	-	eclipsec.exe	サイズ: 298 バイト 更新日時: 2014/08/15 2	1:31	21:31	アプリケーション	24 KB	

ワークスペース・ランチャー	×
フークスペースの選択	
Eclipse は、ワークスペースと呼ばれるフォルダーにプロジェクトを保管します。 このセッションに使用するワークスペース・フォルダーを選択してください。	
ワークスペース(<u>W</u>): C:¥test¥arduino¥workspace	◆ 参照(<u>β</u>)…
同 この選択をデフォルトとして使用し、会後この質問を表示しない((1))	
	<u> </u>



_ Arduino→New Arduino proiectを選択、次へをクリック

新規プロジェクト			100 Miles	
ウィザードを選択 Create a new Arduino sketch	and import arduino core code	if needed.		
ウィザード(<u>W</u>):				
フィルター入力				
▶ 🗁 一般				
🔺 🗁 Arduino				
New Arduino sketch	\triangleright			
▷ 🧀 C/C++				
D C SVN				
(?)	< 戻る(旦)	次へ(N) >>>	完了(E)	キャンセル
プロジェクトネームを適当に決	そめて(今回はtestにします))次へをクリック		
9	R. A. M. 1			
New Arduino sketch				
Create a new Arduino sketch	I.			
プロジェクト名(<u>P</u>): test				
☑ デフォルト・ロケーション	を使用(<u>D</u>)			
ロケーション(L): C:¥test¥ard	luino¥workspace¥test			参照(<u>R</u>)

	ファイル・システムを	選択(Y): デフォルト →			
?		< 戻る(<u>B</u>)		完了(<u>E</u>)	キャンセル

ボードとポートを選択し、完了オ	ドタンをクリックします。		
•			
Provide the Arduino inform	ntion.		
These settings can be ch	anged later.		
Arduino Environment Setti	ings		
Arduino Location	C:¥Program Files (x86)¥Arduino		
Your Arduino board specifi	cations		
Board:	Arduino Uno		\bigcirc
ポート:	СОМЗ		$\overline{\mathbf{O}}$
The used settings			
プロセッサー:	atmega328p	Processor Frequency (Hz):	1600000
Baud:	115200	Board Variant:	標準
UpLoadProtocol:	arduino	Disable Flushing:	
ライブラリー・フォルダー:	arduino	Cpp compile options:	
C compile options:		Link options:	
Build pid:		Build vid:	
?	< 戻る(<u>B</u>)	次へ(N) >	完了(E) キャンセル
-			

以下の画面が表示されます。 選択したボードの名前のブロジェクトとtestという名前のブロジェクトが生成されます。 書き込むソースコードはtestーtest.copです。

C/C++ - Eclipse	a- processes as - by p 10 3	
ファイル(E) 編集(E) ソース(<u>S</u>) リファク	ウタリング(I) ナビゲート(№) 検索(<u>A</u>) プロジェクト(<u>P</u>) AVR(⊻) 実行(<u>R</u>) ウィンドウ(Ψ) へ	ルプ(圧)
📫 🕶 🔚 🐚 📥 🗞 🕶 🗞 🕶 🖬 🕼	🌋 🔍 🗉 🖬 🖆 • 🎕 • 🗳 • 🔗 • 🔅 • 🕐 • 💽 • 🍕 • 🖉 🖉 🥔 🖉 • 🖄	· 월 후 뜻 수 후 수 후 물 · ·
	クイック・フ	アクセス 🖻 🗟 C/C++
🎦 プロジェクト・エクス 🛛 🖳 🗖		≝ 7 % [≫] 2 □ □
□ 🔄 😜 🍸		 €9 ▽
Arduino_Uno		表示するアウトラインはあり ません。
	◎ 問題 23 28 タスク 単 コンソール 目 プロバティー	9º
		עע–ג
	<	•
😤 test	93M / 248M	

Arduino Unoを右クリックし、一番下にあるプロパティを選択します。(以下、testについても同じ操作を行います。) ○ C/C++ - Eclipse

	A TANK I COSTO DO COMORTE TO S	B 1 0 /	
C/C++ - Eclipse		1 1 1	
ファイル(E) 編集(E) ソース(<u>S</u>) リファクタリング(I) ナビゲート	ヽ <u>(ℕ</u>) 検索(<u>A</u>) プロジェクト(<u>P</u>) AVR(<u>V</u>) 実行(<u>R</u>) ウィンドウ(<u>W</u>) ヘルプ(I	<u>H</u>)
📑 • 🗄 🕤 👻 • 🗞 • 🖬 🔮 뿣 🔌 🗉 🖬 🖄	- 🗈 - 🖻 - 🮯 - 🎋 - 🜔 - 🏊 - 🚑	😂 😂 🛷 🕶 🛓 👻 🖗	$\star \leftarrow \star \Rightarrow \star \mid \underline{\mathbf{z}}$
		クイック・アクセ	Z 🖹 🖻 🕞 C/C++
CISION · IDA ⊠ □ CISION · IDA ≅ □ CISION · IDA		8 * *	Eア 22 - ロ 家するアウトラインはあり せん。

	【 問題 33 四 タスク 目 コンソール □ プロパティー 0 項目		₽ ⁻ -
	說明 ^	リソース	<u>л</u>
	< III.		۱.
😂 test	93M / 248M		

<u>_C/C++ ビルド → 環境をクリックします。</u>

🍃 Arduino_Uno のプロパティー	states and the second second		Property and Room		
フィルター入力	環境				↔ • ↔ • •
 > リソース AnyEdit ツール > Arduino ▲ C/C++ ピルド ツール・チェーン・エ: 	構成: Release [アクティブ]				▼ 構成の管理
ディスカバリー・オフ:	設定する環境変数				追加
ビルト変数	変数	値	原点		
	ARDUINOBOARDNAME	Arduino_Uno	BUILD SYST		
操机	ARDUINOBOARDVARIANT	標準	BUILD SYST		編集
	ARDUINO_CORE_FOLDER	arduino	BUILD SYST		
WikiText	ARDUINO_CPP_COMPILE_OPTI		BUILD SYST		
▶ タスク・リポジトリー	ARDUINO_C_COMPILE_OPTIONS		BUILD SYST		定義解除
ビルダー	ARDUINO_IDE_VERSION	105	BUILD SYST		
プロジェクト参照	ARDUINO_LINK_OPTIONS		BUILD SYST		
実行/デバッグ設定	ARDUINO_PID_VALUE		BUILD SYST		
	ARDUINO_VID_VALUE		BUILD SYST		
	AVRDUDEACTIONOPTIONS		BUILD SYST		-
< <u> </u>	● ネイティブ環境へ変数を追加 ○ ネイティブ環境を指定された環境と	置換		デフォルトの復元(D)	適用(<u>A</u>)
?				ОК	キャンセル

ARDUINOBOADVARIANTを選択し、編集ボタンをクリックします。 🍃 Arduino_Uno のプロパティ フィルター入力 ↓ ↓ ↓ ↓ 環境 ▶ リソース AnyEdit ツール 構成: Release [アクティブ] ▼ 構成の管理… > Arduino ⊿ C/C++ ビルド ツール・チェーン・エ ディスカバリー・オプ 設定する環境変数 追加.... ビルド変数 変数 値 原点 選択... ロギング ARDUINOBOARDNAME Arduino_Uno BUILD SYST.. 環境 ARDUINOBOARDVARIANT 標準 BUILD SYST ... 編集... 設定 ARDUINO_CORE_FOLDER arduin BUILD SYST .. ▷ C/C++ 一般 削除 ARDUINO_CPP_COMPILE_OPTI ... BUILD SYST ... WikiText ARDUINO_C_COMPILE_OPTIONS 定義解除 BUILD SYST ... ▷ タスク・リポジトリー ARDUINO_IDE_VERSION 105 BUILD SYST ... ビルダー ARDUINO_LINK_OPTIONS BUILD SYST ... プロジェクト参照 実行/デバッグ設定 ARDUINO_PID_VALUE BUILD SYST ... ARDUINO_VID_VALUE BUILD SYST ... AVRDUDEACTIONOPTIONS BUILD SYST ... ◎ ネイティブ環境へ変数を追加 ◎ ネイティブ環境を指定された環境と置換 デフォルトの復元(<u>D</u>) 適用(<u>A</u>) • III • ? OK キャンセル

編集となっている箇所をstandardと変更してOKボタンをクリックします。

·前:			ARDUINOBOARDVARIAN	п	
ок		ャンヤル		変数	
	-				_
変更後)) 変数の編集					

<u>以下の画面は、適用ポタンをクリックし、OKポタンをクリックします。</u> → Arduino_Uno のプロパティー 「フィルター入力」 環境



			•
設定する環境変数			
変数	値	原点	
ARDUINOBOARDNAME	Arduino_Uno	BUILD SYST	
ARDUINOBOARDVARIANT	standard	USER: CONF	
ARDUINO_CORE_FOLDER	arduino	BUILD SYST	
ARDUINO_CPP_COMPILE_OPTI		BUILD SYST	
ARDUINO_C_COMPILE_OPTIONS		BUILD SYST	
ARDUINO_IDE_VERSION	105	BUILD SYST	
ARDUINO_LINK_OPTIONS		BUILD SYST	
ARDUINO_PID_VALUE		BUILD SYST	
ARDUINO_VID_VALUE		BUILD SYST	
AVRDUDEACTIONOPTIONS		BUILD SYST	
 ○ ネイティブ環境へ変数を追加 ◎ ネイティブ環境を指定された環境と 	置換		
			デフォルトの復元(D)
	設定する環境変数 変数 ARDUINOBOARDNAME ARDUINO_CORE_FOLDER ARDUINO_COPE_COMPILE_OPTI ARDUINO_COPE_COMPILE_OPTIONS ARDUINO_LIDE_VERSION ARDUINO_LIDK_OPTIONS ARDUINO_PID_VALUE ARDUINO_PID_VALUE ARDUINO_PID_VALUE AVRDUDEACTIONOPTIONS ● ネイティブ環境へ変数を追加 ● ネイティブ環境を指定された環境と	設定する環境変数 家数 値 ARDUINOBOARDNAME Arduino_Uno ARDUINO_CORE_FOLDER arduino ARDUINO_COPE_COMPILE_OPTI ARDUINO_COPE_COMPILE_OPTIONS ARDUINO_LINK_OPTIONS ARDUINO_HID_VALUE ARDUINO_PID_VALUE ARDUINO_PID_VALUE AVRDUDEACTIONOPTIONS ● ネイティブ環境へ強数を追加 ● ネイティブ環境を指定された環境と置換	設定する環境変数

tempの設定

追加をクリックします。 ⇒ test のプロパティー フィルター入力	環境	4.10				• <> • •
 > リソース AnyEdit ツール > Arduino a C/C++ ビルド ツール・チェーン・エ: ディスカ/(リー・オプ: 	構成: Release [アクティ: 設定する環境変数	J]			▼ 構	歳の管理
ビルド変数	変数	値	原点			
ロギング	ARDUINOBOARDNAME	Arduino_Uno	BUILD SYST			選択
環境	ARDUINOBOARDVA	standard	USER: CONF		E	編集
設定	ARDUINO_CORE_FOL	arduino	BUILD SYST			
▷ U/U++	ARDUINO_CPP_COMP		BUILD SYST			Elikt-
NATA Uポジトリー	ARDUINO_C_COMPIL		BUILD SYST			定義解除
ビルダー	ARDUINO_IDE_VERSI	105	BUILD SYST			
プロジェクト参昭	ARDUINO_LINK_OPT		BUILD SYST			
実行/デバッグ設定	ARDUINO_PID_VALUE		BUILD SYST		-	
< >	● ネイティブ環境へ変数を〕 ◎ ネイティブ環境を指定され	^{島加} れた環境と置換		デフォルトの復元(<u>D</u>)	適用(<u>A)</u>
?				ОК	キャンセノ	١

2回操作し、以下2つ入れます。すべての構成に追加もチェックします。

IMP C:¥temp			
● 新規変数	and part.		×
名前:	TMP		
値:	C:¥tem	P	変数
✓ すべての構成に追加			
ОК	キャンセル		
TEMP C:¥temp			
● 新規変数	manuferen.		×
名前:	TEMP		
値:	C:¥tem	P	変数
✓ すべての構成に追加			
ОК	キャンセル		

以下の画面で内容確認し、違っていれば、編集ボタンクリックで変更します。 TMP.TEMPがあっていれば、適用ボタンをクリックし、OKボタンをクリックします。

Arduino_Uno のプロパティー			1.101-0.048	COLUMN NUMBER	📄 🐼 A 般 🐸 👳 🥔 🖓	KANA -	x	
フィルター入力	環境	⇔ •						
 > リソース AnyEdit ツール > Arduino a C/C++ ビルド ツール・チェーン・エ: ディスカバ(リー・オプ: 	構成: Release [アクティ 設定する環境変数	7]				構成の智	官理	
ビルド変数	変数	値	原点			A 382	+	
ロギング	ARDUINO_IDE_VERSI	105	BUILD SYST			12	<i>g</i> (
現現	ARDUINO_LINK_OPT		BUILD SYST			編	集	
	ARDUINO_PID_VALUE		BUILD SYST			ă	ill e	
WikiText	ARDUINO_VID_VALUE		BUILD SYST				112	
▶ タスク・リポジトリー	AVRDUDEACTIONOP		BUILD SYST			定義	烏解除	
ビルダー	AVRDUDEOPTIONS	-patmega328p -card	BUILD SYST					
プロジェクト参照	AVRDUDEPATH	C:¥test¥arduino¥Ard	BUILD SYST					

実行/デバッグ設定	AVRTARGETFCPU	16000000	BUILD SYST	
	AVRTARGETMCU	atmega328p	BUILD SYST	
	BUILDARTIFACT	\${ProjName}.a	BUILD SYST	E
	CWD	C:¥test¥arduino¥wor	BUILD SYST	
	PATH	C:¥test¥arduino¥Ard	BUILD SYST	
	PWD	C:¥test¥arduino¥wor	BUILD SYST	
	TEMP	C:¥temp	USER: CONF	
	тмр	C:¥temp	USER: CONF	
				v
	🔘 ネイティブ環境へ変数	を追加		
	◎ ネイティブ環境を指定	こされた環境と置換		
۰ III + I				デフォルトの復元(<u>D</u>) 適用(<u>A</u>)
?				

ここまで、Arrduino_Unoと同じ操作をtestについても同じ操作をおこないます。

。設定が消えてしまわないよう、保管します。(ファイル→保管を選択します)

0	C/C++ - test/test.cpp - Eclipse	11000	
77	マイル(F) 編集(E) ソース(S)	リファクタリング(T)	ナビゲート(N) 検索(A) プロジェクト(P) AVR(V) 実行(R) ウィンドウ(W) ヘルプ(H)
	新規(N) ファイルを開く(.)	Alt+Shift+N ▶] マ 目 1 ピ + G + C + G + 体 + Q + Q + 品 色 G イ + クイック・アクセス - ☆ 団 / (C++)
	閉じる(C) すべて閉じる(L)	Ctrl+W Ctrl+Shift+W	
	保管(S) 別名保管(A) すべて保管(E) 前回保管した状態に戻す(T)	Ctrl+S Ctrl+Shift+S	I3 has an LED connected on most Arduino boards.↓ 13 has an LED connected on most Arduino boards.↓ I a 13.↓ I a 13.↓ <td< th=""></td<>
	移動(V) 名前変更(M) リフレッシュ(F) 遅択リソースのカウント タブィーンフパーフのが海	F2 F5	hillAliza the digital pin as an output.d ode(led, OUTPUT);d loop routine runs over and over again forever;d oov() {d kaWFTEk(led, HIGH); // turn the LED on (HIGH is the voltage level)d v(1500); // wait for a secondd

test.cppをダブルクリックします。 真ん中の上のウィンドウに、test.cppが表示されます。ここにプログラムを書きます。

⇒ C/C++ - test/test.cpp - Eclipse		10 5						
ファイル(E) 編集(E) ソース(S) リファクタリング(I) ナビゲート(<u>N</u>) 検索(<u>A</u>) プロジェクト(<u>P</u>) AVR(<u>V</u>) 実行(<u>R</u>) ウ	マインドウ(W) ヘルプ(日)					
** * 男 ⑤ △ ⑧ * ⑤、* 副 ❷ 🌋 🔍 河 ② ● 目 1 台 * 谷 * ピ * 〇 * 歩 * 〇 * ○ ▲ - 四 * ◇ *								
⊕ ▼ ∅ ▼ ⊕ + → + ≤		クイック・アクセス	λ [] [] [] [] [] C/C++					
🎦 プロジェクト・エクス 🛛 🖳 🗖 🚺 test.cpp 🕅			7 ⊠ [≫] 2 □ □					
Korresove for the setup function of th	the include below,) ion is called once at startup of the sketch,) alization code here,) ion is called in an endless loop,) ed code here,)	7	 E 1^a₂ R k² • # test.h setup() : void loop() : void 					
🔝 問題 🛛 🧔 タスク	🖳 コンソール 🔲 プロパティー							
0 頃目	×	1157	7 11					
a70-93		99	//					
	m		•					
書き込み可能 スマート挿入	117M / 248M							

中に、適当にBlinkでも書き込んでみてください。パンチミスがあると、色がつく場合があります。 ちなみに#include "testh"は必須です。消すと動きません。 書き込んだら、上の金槌のアイコンをクリックしてビルドします。

<u>_真ん中下のウィンドウのコンソールをクリックしま</u>	(す)		
C/C++ - test/test.cpp - Eclipse			
ファイル(E) 編集(E) ソース(S) リファクタリング	(I) ナビゲート(№) 検索(A) プロジェクト(P) AVR(⊻) 実行(R) ウィンドウ(Ψ) ヘルプ(出)		
🗈 🗝 🖩 🕒 👋 🔦 🖻 🚇 鑽 💱	x 📝 🌣 🗉 🖬 🔯 + 🚳 + 🗳 + 🞯 + 1 🎋 + 🖸 + 🎭 + 🔁 😕 😂 🔗 + 1 灯 +	§ - *> ↔ ↔ - → ·	1
		クイック	・アクセス 🖻 📴 C/C++
🎦 プロジェクト・エクスプローラー 🛙 🖓 🗖	€ test.cpp 🛛		ア X ⑧ M 目夕 □
S Arduino_Uno Set test Set test Set test Set test Set test. Set test.cpp U test.ch Set test.in Set led : int Sop() : void Setup() : void	<pre>13</pre>	E	 P ⁴/₂ ≥ × ⁵ • # I test.h Ied : int setup() : void loop() : void
⊳ ja test.h	🖹 問題 🛿 🖉 タスク 🗐 コンソール 💷 プロパティー		Ş [¬] − □
	0項目		
	説明	リソース	7.2

je j	書き し下のような英語がでれば、ビルド売 ● C/C++ - test/test.cp - Eclipse ファイル(E) 編集(E) ソース(S) リフ 同 ● 同 ● ● ※ ≪ ● ●	・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・	スマート挿入 27:1 (N) 検索(A) プロジェクト(E) AVR() 11 (2) × 63 × 6 ² × 6 ² × 6 ² × 6 ³ × 5 ⁴ ×	[™] <u>72M /</u> 248M 〕 248 (1) 248		,
	り * ③ * ゆ ゆ * * * ゴ つ プロジェクト・エクス ※ □ □ ● ● ▲ Arduino_Uno * ● ● Arduino_Uno * ● ● Arduino_Uno * ● ● Arduino_Uno * ● ● Release > ● @ Release > @ Release > @ test.h	test.cpp ⊠	ו LED connected on most Arduino boards e:J the runs once when you press reset: the digital pin as an output. J WTPUT); J ine runs over and over again forever: ed, HIGH): // turn the LED on (HIGH ed, LOW): // turn the LED off by // wait for a secondJ	ل با is the voltage level) پ پ	b b </td <td></td>	
		Mill 20529 CDT Build Console [7:] Data: Il byte: (.data + .bss + .noini Cinished building: siz sis:18:58 Build Finishe I	コンソール 20 プロパディー スト) (0.55 F0II): (t): (t): ad (took \$38ms): 144M / 248M	0 0 🕄 🖪 🖬 🖬		

ビルドできたらその左側にあるAVRと書かれたアイコンをクリックして、Arduinoにダウンロードします。

「レルトできたらての生態にのるAVRと音かれた	アイコンをクリックして、Arduirioにタリンロートします。							
C/C++ - test/test.cpp - Eclipse		10 To Th						
ファイル(F) 編集(E) ソース(S) リファクタリング	ウ(T) ナビゲート(N) 検索(A) プロジェクト(P) AVR(V) 実行(R) ウィンドウ(W) ヘルプ(H)							
	岱 ▼ 開 廠 △ ③ ▼ ≪ ▼ 局 (Q(**)) ↓ 彡 回 回 (岔 * 硲 * ピ * ♂ * ◎ * ○ * Q * ○ @ ● @ @ @ Ø * ½ * 秒 * や ◆ * → * ヱ							
クイック・アクセス 野 職 C/C++)								
🎦 プロジェクト・エクスプローラー 🛛 🖳 🗖	€ test.cpp 🔀	- 8	第78 ®M 目夕 □					
E	<pre>14 // the setup routine runs once when you press reset:] 15 void setup() [] 16 vii setup() [] 17 pindoc(led, UUTVUT);] 19]] 19]] 19]] 19]] 20 // the loop routine runs over and over again forever:] 21 Void loop() [] 22 digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)] 23 delsv(1000); // wait for a second] 24 digitalWrite(led, LOW); // turn the LED of by making the voltage LOW] 25 delsv(1000); // wait for a second] 26]] 27 / 4 </pre>		 P: □ I⁴_Z & x⁵ • ₩ I test.h Ied : int setup() : void Icop() : void 					
• setup() : void	🖳 問題 🛛 🔊 タスク 🗉 コンソール 🔲 プロパティー		\$ ▽					
⊳ <u>in</u> test.n	エラー: 0、 警告: 3、その他: 0							
		リソース	パス					
	▶ ▲ 智告 (3 項目)							
			•					
	89M / 248M							

もし、ダウンロードできなかっ また、testを右クリック→プロ き test のプロパティー	た場合、USBでarduino <u>パティを選択し、以下の</u>	に接続してあるか、確認しましょう。 D画面でBoardとポートが合っているか確	
	Arduino		$\Leftrightarrow \bullet \Rightarrow \Rightarrow \bullet \bullet$
▷ リソース AnyEdit ツール ▲ Arduino	Arduino Environment : Arduino Location	Settings C:¥test¥arduino¥Arduino	
Arduino	Your Arduino board sp	ecifications	
Target Hardware	Board:	Arduino Uno	•
▷ C/C++ ビルド ▷ C/C++ 一般	ポート:	СОМЗ	•

C/C++ ビルド	ポート:	COM3 •					
WikiText	The used settings						
タスク・リポジトリー	プロセッサー:	atmega328p	Processor Frequency (Hz):	1600000			
ビルダー プロジェクト参昭	Baud:	115200	Board Variant:	標準			
実行/デバッグ設定	UpLoadProtocol:	arduino	Disable Flushing:				
	ライブラリー・フォルダー:	arduino	Cpp compile options:				
	C compile options:		Link options:				
	Build pid:		Build vid:				
		デフォルトの	の復元(<u>D</u>) 適用	∃(<u>A</u>)			



とりあえず、動かすのは、おしまいです。

2. 既存プログラムをデバッグで動かす。

できないようです。(2011/12/03現在)

―― Arduinoで, ちょっと凝ったことをしようと思うと, デバッガが欲しいと思うことがあります. このあたり, 将来的に提供の計画はありますか?

ジャンルカ:これは、エンジニアの方々からよく聞かれます、デバッガがあったらいいとは思いますが、実現するのは簡単ではありません、AVRの内部のレ ンマンルカニスれば、エンシーアの方々からよく聞かれより、ナハッカかめつたらいいとは思いますか、美現するのは間単ではありません。AVRの内部のレ ジスタやメモリにアクセスするには、Atmel社のDebugWireの機能を利用する必要があります。JTAGインターフェースはありません。ところが、DebugWireは Atmel社の独自機能であり、その詳細は非公開です、内部のレジスタやメモリへのアクセスは、マイコンのアーキテクチャに非常に近い部分であり、企業秘 密を守るためには、DebugWireの詳細の公開は難しいでしょう。ただし、Atmel社はDebugWireの一部機能の公開を考えているようですので、将来はデバッ ガ機能のプラグインを実現できるかもしれません。

Arduinoは初学者やアーティストのためのツールであって、工業的なアプリケーションを目的とはしていません、デバッガが必要なくらいに複雑なアプリケーションを開発するのであれば、例えばAVRStudioなどの開発ツールを使うほうがいいでしょう。

引用 オープン・ハード「Arduino」の秘密をひも解く —— Arduino開発チーム代表 ジャンルカ・マティーノ氏に独占インタビュー ! http://www.kumikomi.net/archives/2011/12/in08gian.php