

Atmel Studio の環境立ち上げと 使用方法

Ver1.0.0
2015/5/12



本資料は、Atmel Studio で Studuino(スタディーノ)基板上で動作するプログラムを作成する際の手順書になります。Atmel Studio の変更に伴い、加筆・修正が加えられる可能性があります。また、本取扱説明書は、下記の資料を参照します。

更新履歴

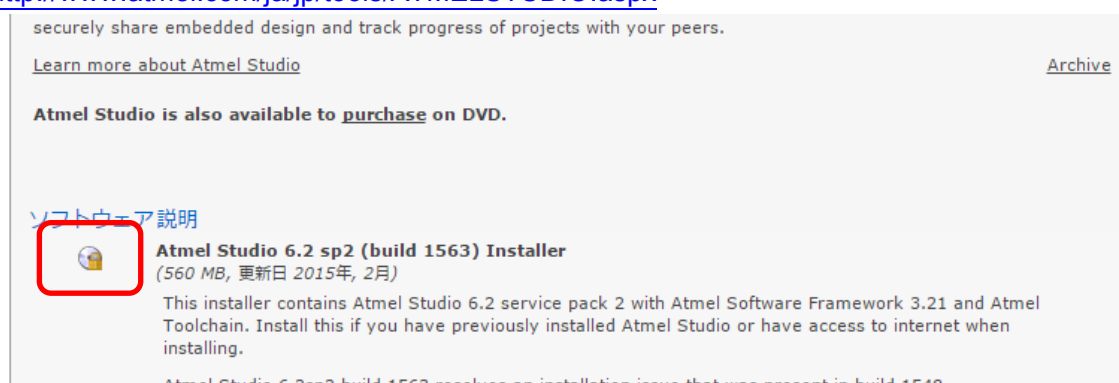
バージョン	前バージョンからの変更内容
ver.1.0.0	新規作成

本資料では、Atmel Studio を使用し、Studuino 基板上で動作するプログラムを作成する手順を記します。最初に Atmel Studio のダウンロードとインストールの手順を記し、次に Studuino 基板上で動作するプログラムを作成するための Atmel Studio の環境設定、最後にプログラミング例を記します。

1. Atmel Studio のダウンロードとインストール

1. 下記のウェブページにアクセスし、ソフトウェアをダウンロードします。

<http://www.atmel.com/ja/jp/tools/ATMELSTUDIO.aspx>

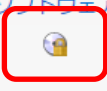


securely share embedded design and track progress of projects with your peers.

[Learn more about Atmel Studio](#) [Archive](#)

Atmel Studio is also available to [purchase](#) on DVD.

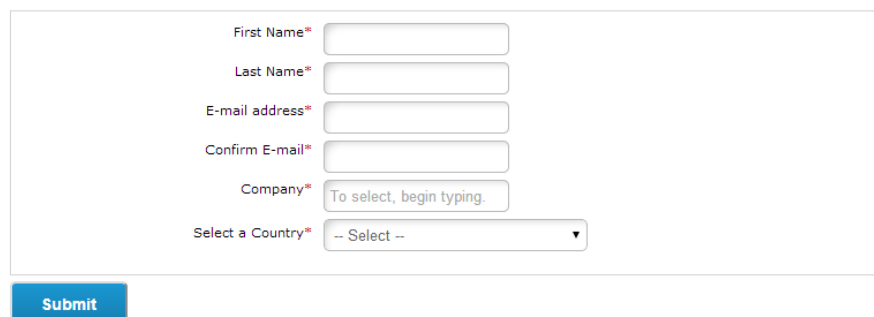
ソフトウェア説明

 **Atmel Studio 6.2 sp2 (build 1563) Installer**
(560 MB, 更新日 2015年, 2月)

This installer contains Atmel Studio 6.2 service pack 2 with Atmel Software Framework 3.21 and Atmel Toolchain. Install this if you have previously installed Atmel Studio or have access to internet when installing.

Atmel Studio 6.2sp2 build 1563 resolves an installation issue that was present in build 1548.

2. 必須項目を入力して Submit ボタンをクリックします。



First Name*

Last Name*

E-mail address*

Confirm E-mail*

Company*

Select a Country*

3. 2 で記入した E-mail address にメールが届くので、その本文の「download the file here」をクリックします。



Dear none,

Thank you for completing the form to download a file from our site.

You can [download the file here](#), or click the link below. If the link is not clickable, simply copy and paste the URL into your web browser.

4. リンク先には下図のように「Atmel Studio ... Installer」というリンクがあります。クリックしてインストーラをダウンロードします(バージョンはその時々で変わります)。

Home > Forms

Software Download

Your Download is Waiting for You

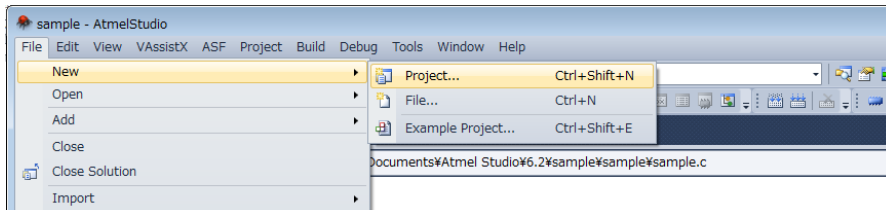
Please use the link below to download the file you requested.

[Atmel Studio 6.2 sp2 \(build 1563\) Installer](#)

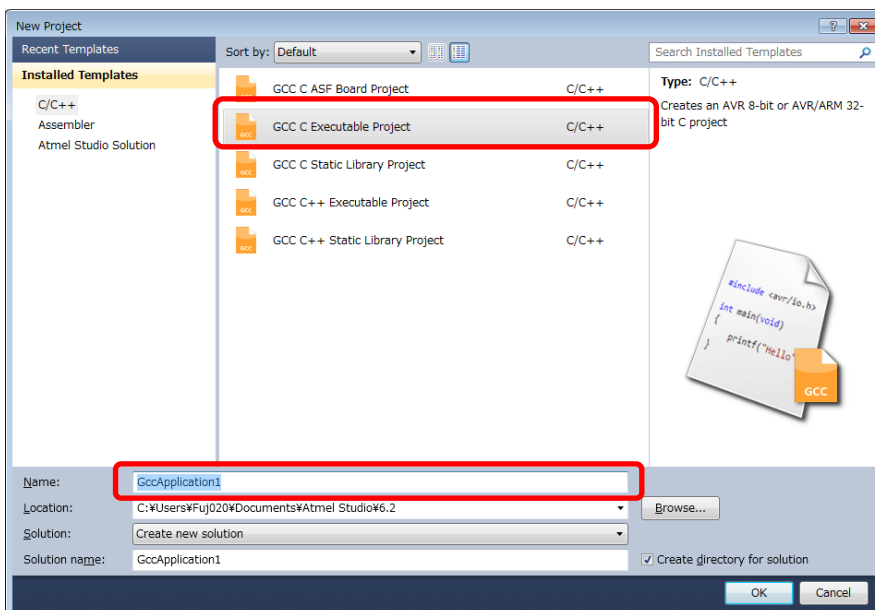
5. ダウンロードしたファイルを起動して、手順に従ってインストールします。
6. インストール完了後、自動的にセットアップが始まるので手順に従ってセットアップします。

2. Atmel Studio の環境設定

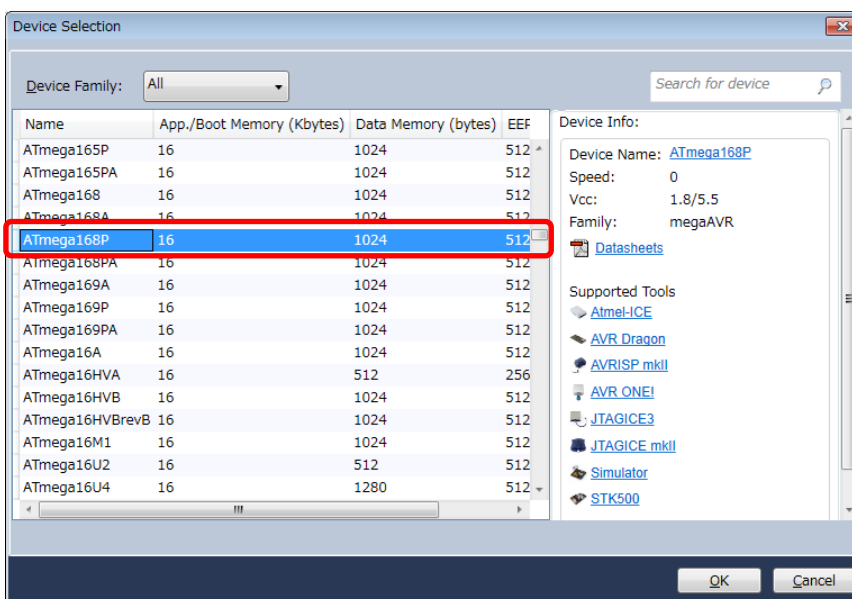
1. Atmel Studio を起動し、File メニュー→New→Project を選択します。



2. 「GCC C Executable Project」を選択し、Name:欄にプロジェクト名を設定します。



3. 表示される Device Selection ダイアログで、「ATmega168p」を選択して、OK ボタンをクリックします。

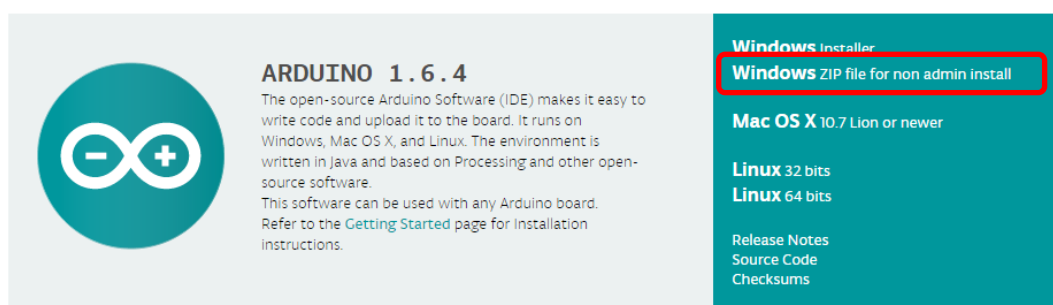


4. 転送プログラムを登録します。

本設定では、転送プログラムの登録に Arduino IDE の avrdude というプログラムを使用します。Arduino IDE をインストールしていない場合は、下記 URL のページにある下図の「Windows ZIP file...install」をクリックし、Arduino IDE をダウンロードして適当なフォルダに展開してください。

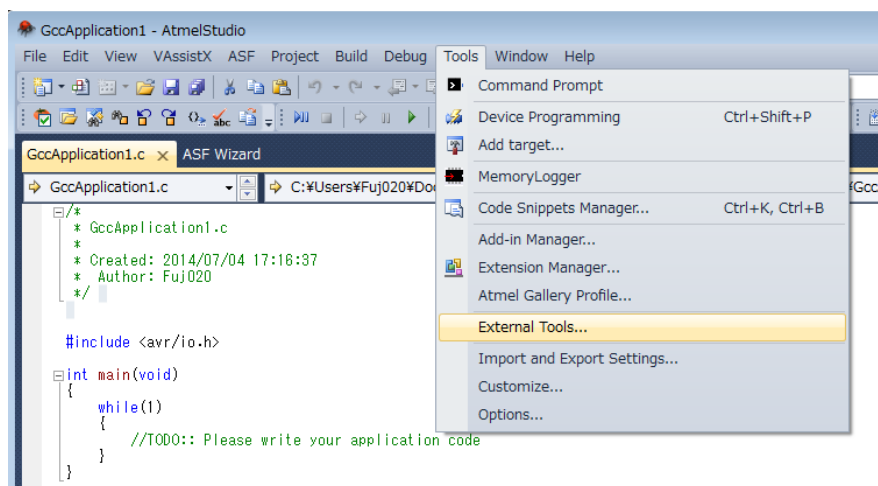
<http://www.arduino.cc/en/Main/Software>

Download the Arduino Software



以下、Arduino IDE を C ドライブに展開したとして、手順を進めます。

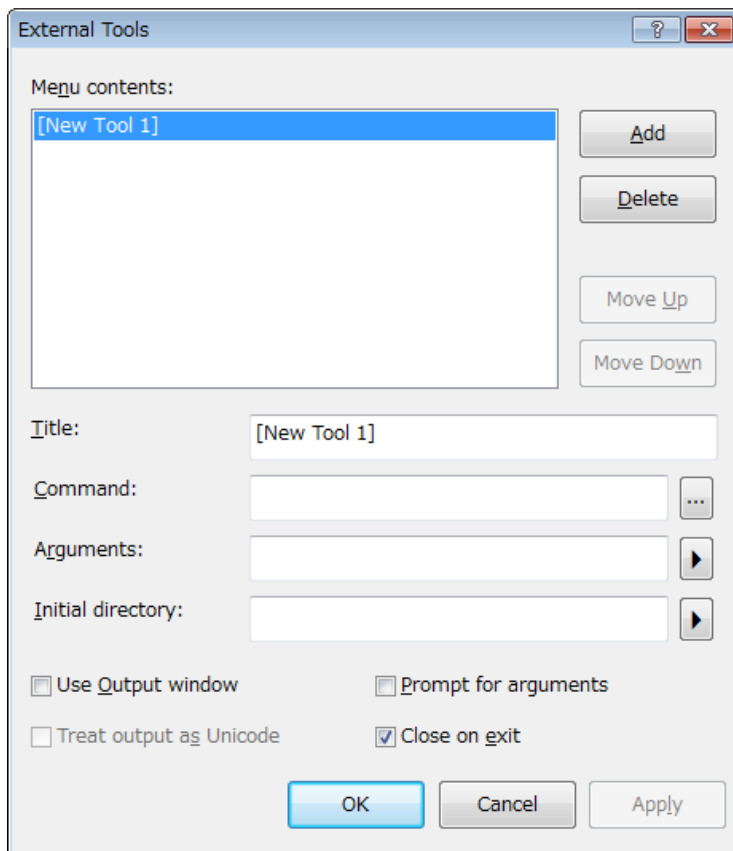
5. Tools メニュー→External Tools...を選択します。



※：本手順 5～7 は下記、URL の記事を参考にしております。

<https://asensar.com/blog/2013/how-to-integrate-avrdude-with-atmel-studio/>

6. 表示されるウィンドウに下記のように各項目を設定し、OK ボタンをクリックします。



- Title: &+適切な名前を設定します。

例) &Deploy

- Command:avrdude.exe をフルパスで設定します。

例) C:\arduino-1.6.4\hardware\tools\avr\bin

- Arguments:下記を設定します(例の赤字のところはユーザによって変わります)

```
-F -v -patmega168 -carduino -P\\.\COM3 -b115200 -D  
-Uflash:w:"$(ProjectDir)Debug\$(ItemFileName).hex":i -C
```

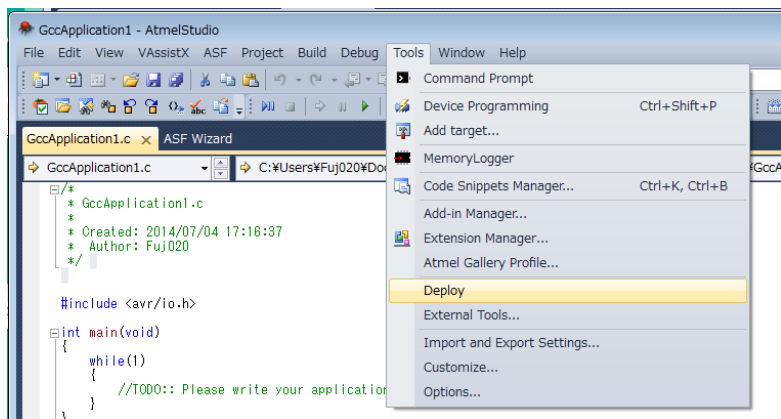
C:\arduino-1.6.4\hardware\tools\avr\etc\avrdude.conf

(※) COM3 のところは、Studuino 基板と PC を接続した際に使用される COM ポートになります。Studuino 基板と PC を接続して、Arduino IDE やデバイスマネージャなどで確認してください。

(※) avrdude.conf ファイルは、Arduino IDE を展開したフォルダにあります。

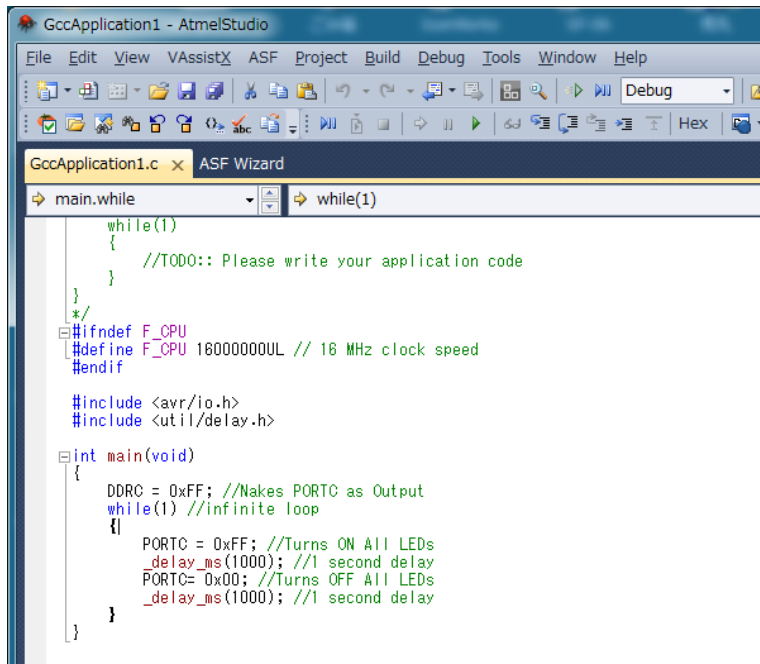
- Use Output window にチェックを入れます。

7. Tools メニューに 6 の Title: に設定した名前がメニューに表示されます。



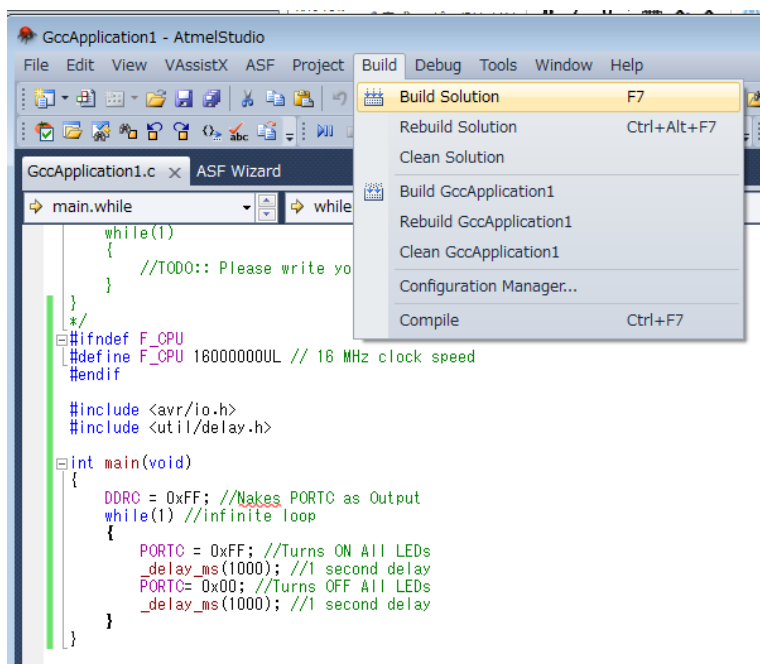
3. プログラミング例

1. プロジェクトを作成した際に作られるソースファイル(本例では、GccApplication1.c)に下記のようなプログラム(LED を1秒間隔で点滅するプログラム)を記述します(テキストが末尾にあります)。



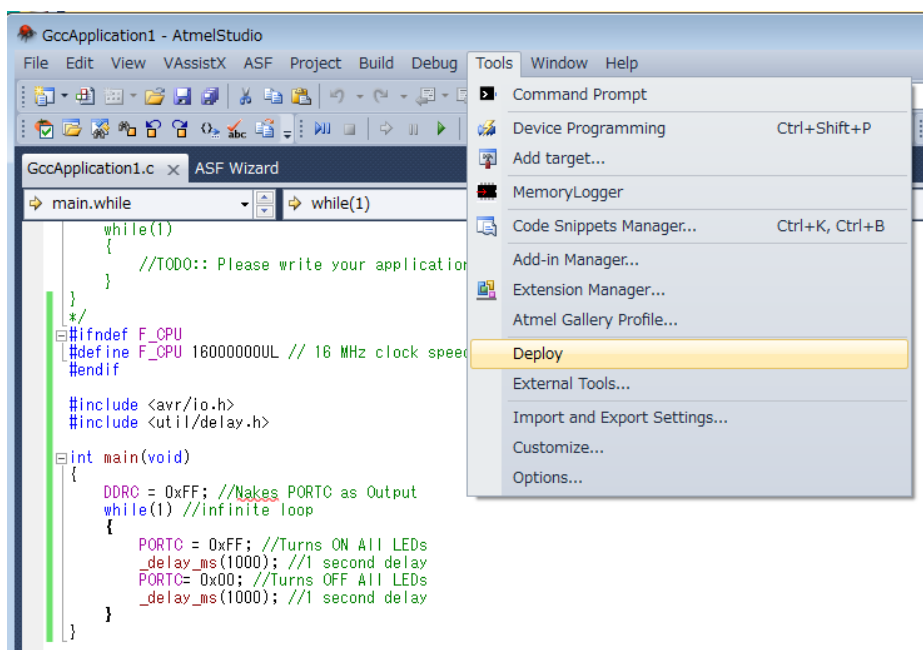
```
GccApplication1 - AtmelStudio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
GccApplication1.c x ASF Wizard
main.whi while(1)
while(1)
{
//TODO:: Please write your application code
}
*/
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
DDRC = 0xFF; //Nakes PORTC as Output
while(1) //infinite loop
{
PORTC = 0xFF; //Turns ON All LEDs
_delay_ms(1000); //1 second delay
PORTC= 0x00; //Turns OFF All LEDs
_delay_ms(1000); //1 second delay
}
}
```

2. Build メニュー→Build Solution を選択してプログラムをビルドします。



```
GccApplication1 - AtmelStudio
File Edit View VAssistX ASF Project Build Debug Tools Window Help
Build Solution F7
Rebuild Solution Ctrl+Alt+F7
Clean Solution
Build GccApplication1
Rebuild GccApplication1
Clean GccApplication1
Configuration Manager...
Compile Ctrl+F7
GccApplication1.c x ASF Wizard
main.whi while(1)
while(1)
{
//TODO:: Please write yo
}
*/
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif
#include <avr/io.h>
#include <util/delay.h>
int main(void)
{
DDRC = 0xFF; //Nakes PORTC as Output
while(1) //infinite loop
{
PORTC = 0xFF; //Turns ON All LEDs
_delay_ms(1000); //1 second delay
PORTC= 0x00; //Turns OFF All LEDs
_delay_ms(1000); //1 second delay
}
}
```

3. Studuino 基板の A0~A7 のいずれかに LED パーツを接続し、Studuino 基板と PC を USB ケーブルで接続します。
4. Tools メニュー→Deploy を選択してプログラムを Studuino 基板に送信します。



5. センサーコネクタにつなげた LED が 1 秒間隔で点滅します。

3.1. サンプルプログラム

サンプルプログラムを下記に記します。

```
#ifndef F_CPU
#define F_CPU 16000000UL // 16 MHz clock speed
#endif

#include <avr/io.h>
#include <util/delay.h>

int main(void)
{
    DDRC = 0xFF; //Nakes PORTC as Output
    while(1) //infinite loop
    {
        PORTC = 0xFF; //Turns ON All LEDs
        _delay_ms(1000); //1 second delay
        PORTC= 0x00; //Turns OFF All LEDs
        _delay_ms(1000); //1 second delay
    }
}
```