.NET Gadgeteer IoT キット 学習テキスト 準備と基本操作

2015年2月12日

株式会社デバイスドライバーズ

| 1. はじ | こめに |
|-------|------------------------------|
| 1.1. | 開発環境3 |
| 1.2. | ターゲット環境 |
| 1.3. | ライセンスと免責事項4 |
| 1.4. | 演習の概要 |
| 2. 開発 | ⁸ 環境の準備5 |
| 2.1. | Visual Studio 2013 Update 45 |
| 2.2. | SDK パッケージの入手7 |
| 2.3. | SDK パッケージのインストール10 |
| 3. ファ | アームウェアの確認と更新16 |
| 3.1. | ファームウェアの確認16 |
| 3.2. | TinyCLR だけの更新18 |
| 3.3. | TinyBooter の更新 |
| 4NE | TMF アプリケーション開発の基本27 |
| 4.1. | VC#利用前の準備 |
| 4.2. | 単純なアプリケーションの開発 |
| 4.3. | 単純なアプリケーションの動作確認 |
| 4.4. | メインボードのネットワーク設定 |
| 4.5. | ネットワーク・アプリケーションの開発40 |
| 4.6. | 参考文献 |
| 5. その | >他 |

目次

1. はじめに

.NET Micro Framework (NETMF) および.NET Gadgeteer は、Microsoft Research 社が開 発して来た組み込みシステム向けのオペレーティング・システム、開発環境とライブラリ、及び それらを開発するための規格です。現在は Apache Version 2 ライセンスに基づいたオープンソ ースとなり、Microsoft 社のほか FEZ Spider 等のハードウェアを開発している GHI Electronics 社も参画して開発をしています。

この学習テキストでは.NET Gadgeteer が動作する FEZ Spider メインボード、及びそれと組 み合わせて利用可能なモジュール群を使用して IoT (Internet of Things) を容易に体験するた めの準備と基本的な開発手順を示します。自己学習にも活用して下さい。

このテキストでは次の印と項目を使用しています。独習する際に参考にしてください。

| ●演習 | 自己学習を進める上で必須となる演習です。 |
|-------|---------------------------------|
| ○演習 | オプションの演習項目です。条件に合致する人を対象にしています。 |
| ★注意★ | 演習を進める上での重要な注意点を示します。 |
| ■ヒント■ | 演習を進める上での有効なヒントと参考情報を示します。 |
| □解説□ | 演習や手順に関する解説です。 |

■ヒント:インストール先のフォルダ名■

本文中の説明では、各種ソフトウェアを 64bit 版 Windows 環境にインストールして利用する 場合を想定します。32bit 版環境にインストールして使用する場合には、インストール先のフォ ルダ名を C:¥Program Files (x86) から C:¥Program Files に読み替えてご利用ください。

1.1. 開発環境

オペレーティング・システム: Windows Vista 以降を搭載した PC

開発環境 (Visual Studio): Visual Studio Express 2013 with Update 4 for Windows Desktop, Visual Studio Community 2013 Update 4,または Visual Studio Professional 2013 with Update 4 以上(上位版も可、他バージョンは利用不可)

NETMF SDK: Microsoft .NET Micro Framework SDK 4.3 (QFE2) (他バージョンは利用不可)

Gadgeteer Core: Microsoft Gadgeteer Core 2.43.1000 以降

GHI SDK: GHI Electronics NETMF SDK 2014.R5 以降

.NET Gadgeteer 修正版ライブラリ: Gadgeteer.Webserver43

(株式会社デバイスドライバーズにて作成・配布、本演習に添付)

必要な DISK 容量: 空きディスク領域 2GB

マシン仕様: Pentium 4 2GHz 以上、メモリ 2GB 以上を推奨。

■ヒント:複数種類の Visual Studio のインストール■

Visual Studio はインストールの順番に関係なく、Express 版でも別のバージョンでも同じ開 発環境(マシン)にインストール可能です。各 SDK は、Visual Studio のプラグインとなりま すので、後からインストールした Visual Studio で SDK を使用する場合には、SDK の再インス トールが必要です。

1.2. ターゲット環境

本テキストで動作対象としているメインボードは、現在 FEZ Spider だけです。 メインボード GHI electronics FEZ Spider (TinyBooter4.3.4.0 以降 TinyCLR 4.3.6.0 以降) モジュール: USB Client SP Module または USB Client DP Module

Multicolor LED Module, Ethernet J11 Module (オプションのプログラミング演習で使用)

★注意:ファームウェアのバージョン★

上記以外のバージョンでも動作する可能性はありますが、本テキスト執筆時点では動作確認していませんのでご注意下さい。

1.3. ライセンスと免責事項

- 本資料に基づいて演習、自己学習を実施した結果、付帯して配布するソフトウェアの運用に おいて生じた、いかなる損害について一切責任は負いません。
- ② 本資料で提供するソフトウェアは、Ms-PL(http://opensource.org/licenses/ms-pl)で提供 します。演習で使用するソフトウェアで別のライセンスが明示されている場合には、そのラ イセンスに従って使用してください。

1.4. 演習の概要

① 目的

本テキストでは.NET Gadgeteer を使用した組み込みアプリケーション開発方法の基礎を学びます。

- ② 必要知識
- · Visual Studioの基本的な使い方を知っていることが望ましい(必須ではない)
- ・C#に関する知識があることが望ましい(必須ではない)
- 3 開発言語
- · C#
- ④ 演習終了後に得られる知識
- ・.NET Gadgeteer を利用した簡単な組み込みシステムの開発方法
- .NET Gadgeteer の開発システムの使いこなしに必要なソフトウェアのインストールとファ
 ームウェアのメンテナンス手順

2. 開発環境の準備

.NET Gadgeteer / .NET Micro Framework の開発にはコンパイルとデプロイ、デバッグを行 うツールである Visual Studio とそれをサポートする SDK が必要です。開発を始める前に Windows マシンに開発環境をインストールして準備します。

2.1. Visual Studio 2013 Update 4

○演習1)

以下の手順で Visual Studio Express 2013 for Windows Desktop を入手し、インストールを 行います。

すでにインストール済の場合、Professional 以上をインストール済の場合は必要ありません。

■ヒント: Visual Studio Express 2013 with Update 4 for Windows Desktop の入手先■

- (1) http://www.visualstudio.com/downloads/download-visual-studio-vs
- 2 http://www.microsoft.com/ja-jp/download/details.aspx?id=44914
- ③ インターネットで「Visual Studio Express 2013 update 4」で検索する。
- ④ 利用権利を持っている人はMSDNのサブスクライバー・ダウンロードからも入手可能です。

| (← ⊙) thttp://www.microsoft.com/ja-jp/download/details.aspx?id=44914 | Microsoft Visu × 🗎 🕆 🌣 |
|---|---|
| プデリル(E) 補属(E) 表示(L) あえに入り(A) ツール(L) ハルノ(H) ■ Microsoft | ^ |
| Download Center | ٩ |
| 購入 🗸 製品 🗸 カテゴリ 🗸 サポート 🗸 セキュリテイ 🗸 | |
| Microsoft Visual Studio Express 2013 fo Update 4 | or Windows Desktop |
| 言語を選択: 日本語 | 無料の更新プログ ラム |
| Visual Studio Express 2013 for Windows Desktop では、C#、 Visual Basic、および C++ でデスクトップ アプリを作成できます。ま た、Windows Presentation Foundation (WPF)、Windows フォー ム、および Win32 がサポートされています。 | セキュリティ更新プログラム ソフトウェア更新プログラム Service Pack ハードウェアドライバー |
| ↔ 詳細 | @ Microsoft Update |
| (→) システム要件 | ()天1) |
| (土) インストール方法 | |
| | Microsoft のおすすめ |
| | Visual Studio |

http://www.microsoft.com/ja-jp/download/details.aspx?id=44914

から入手してインストールする場合には次の様に「ダウンロードするプログラムを選んでください」の画面で、wdexpress_full.exeを選択して実行すると、その場でインストールが始まります。

| | | | - 🗆 🗙 |
|---|--------------------|---------------------|----------------------|
| ← → Http://www.microsoft.com/ja-jp/download/details.aspx?id=44914 | ・ C 🚦 Download Mic | rosoft Visu × | ↑ ★ ☆ |
| ファイル(E) 編集(E) 表示(⊻) お気に入り(A) ツール(工) ヘルプ(出) | | | |
| Microsoft | | | |
| Download Center | | | ٩ |
| 勝口 測던 동금국내 부분 두 동구동내는 기 | | | |
| ダウンロードするプログラムを選んで | でください。 | | × |
| ファイル名 | サイズ | (N | |
| vs2013.4_dskexp_JPN.iso | 6.5 GB | タウンロードの概要: | |
| wdexpress_full.exe | 1.1 MB | 1. Wackpress_numexe | |
| | | | |
| | | | |
| | | | |
| | | 合計サイズ: 1.1 MB | |
| | | | |
| | | | _ 次へ |
| (+) インストール方法 | | | |
| (土) 関連リソース | | | すすめ |
| | | | tudio nity 2013 を |

★注意:古いバージョン★

Visual Studio 2010 以前の古いバージョンは、現在のところ、このテキストで解説している.NET Micro Framework / .NET Gadgeteer バージョン 4.3 の開発には利用できません。

○演習2)

前項のいずれかの方法で Visual Studio 2013 Update 4 を入手してインストールします。

□解説: Visual Studio 更新プログラム Update 4□

Visual Studioの更新プログラムは各バージョンの Visual Studio 公開後に発生した機能追加 や、問題修正のために配布されています。Visual Studio 2013 Update 4 は、インストール時に 最初から Update 4 の更新内容を含んでいるものです。

更新プログラムはバージョンが同じであれば Express 版を含む全ての Visual Studio に適合可 能です。本テキスト作成時点での最新版の更新プログラムは「更新プログラム 4」です。Visual Studio 2013 をすでに利用していて、Update 4 にバージョンアップしていない場合は、コンパ イラや開発ツール関連の機能拡張が行われているので、インストールしておくことをお勧めしま す。

2.2. SDK パッケージの入手

●演習3)

各サイトから SDK をダウンロードしてインストールします。

★注意:古いバージョンのアンインストール★

Microsoft .NET Micro Framework SDK 4.3 (RTM) 等、上記に指定したよりも古い各 SDK がインストールしてある場合には、SDK のインストールを始める前に、アンインストールして おく必要があります。必ずコントロールパネルの「プログラムと機能」で確認して下さい。

古い SDK をインストールしたままだと、新 SDK のインストール時にアンインストールや修 復インストールを求められます。各略号は、

QFE=Quick Fix Engineering

RTM=Release To Market

の略で、QFE は RTM の不具合を修正したバージョンです。インストール済バージョンの確認と アンインストールは「プログラムと機能」を起動して行います。

□解説:各 SDK パッケージの入手先□ 以下のサイトから各 SDK を入手します。

- Microsoft .NET Micro Framework SDK 4.3 (QFE2) http://netmf.codeplex.com/releases/view/611040
- ② Microsoft .NET Gadgeteer Core バージョン 2.43.1000 http://gadgeteer.codeplex.com/releases

③ GHI NETMF SDK 2014 R5

■ヒント:各ソフトウェアの入手■

以下の GHI Electronics 社のホームページには入手先のリンクがインストール順に掲載されて います。ただしここのリンクは現在のところ、英語版 Visual Studio 2012 と Microsoft .NET Micro Framework SDK 4.3 (QFE1)へのリンクとなっているので注意して下さい。

https://www.ghielectronics.com/support/netmf

次ページに示す様に、.NET Micro Framework(NETMF)の画面の **Step 4** の NETMF and Gadgeteer Package 2014 R5 の部分をクリックするとダウンロード画面に移動します。



NETMF and Gadgeteer Package 2014 R5 のダウンロード画面では、**NETMF and** Gadgeteer Package 2014 R5.exe (48.17 MB) のファイル名をクリックします。



ファイル名をクリックするとログイン画面に移動するので、ユーザー未登録の場合には 「Register」をクリックしてユーザー登録します。

| | | - • × |
|------------------------------------|---|-------------------|
| | er 🔎 🗸 🔒 🕈 👜 Register - GHI Electronics 🗙 | ĥ★ \$ |
| ファイル(E) 編集(E) 表示(V) お気に入り(A) ツール(1 |) ヘルプ(<u>H</u>) | |
| Catalog | Start a search Start a search O | ₹† f ⊵ ゐ ^ |
| Technologies Catalog Support Cor | munity Company | Log In |
| Home > Register | | |
| Register | | |
| | | |
| Log In | * Indicates required fields. | |
| Register | E-mail | |
| Password Reset * Pa | sword | |
| * <u>Us</u> | rname | |
| Tir | ezone Select: | |
| | Please help us verify you aren't a hot or snammer by answering | |
| | the electronics-related question in English below: | |
| | What color is dirt? | |
| | num r know, <u>snow me a new quesuon</u> . | |
| | liswei | |
| | I agree to the <u>Terms of Use</u> and <u>Privacy</u> policies. | |
| | Submit | |
| | | |
| | | |
| | | |
| | | |
| 0 2 | 103-2014 GHI Electronics LLC All rights reserved. Policies Contact Us | |
| | | ~ |

英語の質問に答えてユーザー登録をした後は、そのままダウンロード画面に戻ってダウンロー ド可能です。

2.3. SDK パッケージのインストール

●演習4)

入手した SDK パッケージのインストールを順番に行います。

① Microsoft .NET Micro Framework SDK 4.3 (QFE2)

netmf-v4.3.2-SDK-QFE2-RTM.zip を適当なフォルダに展開すると次の様に、 MicroFrameworkSDK.MSIファイルと3種のvsixファイルができます。vsixファイルは各 Visual Studio 用のアドインです。



Visual Studio を終了しておいて、最初に MicroFrameworkSDK.MSI をダブルクリックして 実行すると、Microsoft .NET Micro Framework SDK 4.3 (QFE2)のインストールが始まります。







ライセンス確認後、Setup Typeの画面では「Complete」を選択します。



インストール完了後、Visual Studio を終了したまま、netmfvs2013.vsix ファイルをダブルク リックして、Visual Studio 2013 用アドインのインストールを行います。

② Microsoft Gadgeteer Core 2.43.1000

GadgeteerCore.msi.をダブルクリックして、NET Gadgeteer Core のインストールを行います。



ライセンス確認後、インストール先ディレクトリはデフォルトのまま進みます。

| Installation Location |
|---|
| Microsoft .NET Gadgeteer Core will be installed at the following location on your computer. To install in a different location, click Browse and then select a location. |
| Location: |
| C:\Program Files (x86)\Microsoft .NET Gadgeteer\ Browse |
| |
| |
| |
| |
| |

.NET Gadgeteer Core のインストールが完了しました。

③ GHI NETMF SDK 2014 R5

入手した NETMF and Gadgeteer Package 2015 R2.exe を起動して、表示される画面の利 用条件にチェックを入れた後、install をクリックしてインストールを開始します。



少しするとAtmel社のFlash ROMメンテナンスツール(SAM-BA)のインストールダイアログ が表示されるので、「Next」をクリックして進み、インストールを完了します。



ライセンス確認、インストール先のディレクトリ確認、メニューフォルダー確認後、完了画面 が出ます。 ブラウザで SAM-BA の USB ドライバの互換性に関する注意ページが開くので、閉じます。



続いて STMicroelectronics 社の Flash ROM ツール(DfuSe)のインストールが自動的に始まります。



| InstallShield Wizard | × |
|--|---------------|
| Customer Information Please enter your information. | No. |
| User Name: | |
| Atomu Hidaka | |
| Company Name: | |
| DDLI | |
| Install this application for: | |
| Anyone who uses this computer | (all users) |
| ⊂ Only for <u>m</u> e (Atomu Hidaka) | |
| | |
| nstallShield | |
| < Back | Next > Cancel |

インストール先のディレクトリ確認後インストールが進み、デバイスドライバのインストール 確認ダイアログ画面が出るので、インストールをクリックします。

| • | Windows せ | キュリティ | ĸ |
|---|--|-----------------------------------|---|
| このデバイス ソフト | ウェアをインストールしますか? | | |
| 名前: GHI I 《 ^柔 発行元: GH | Electronics LLC ユニバーサル シリアル H Electronics LLC | パスデバイス | |
| □ "GHI Electronic る(<u>A</u>) | s LLC" からのソフトウェアを常に信頼す | インストール(I) インストールしない(N) | |
| ・ ・ | らのドライバー ソフトウェアのみをインストーノ | ししてください。 安全にインストールできるデバイス ソフトウェアを | 2 |

Close をクリックして GHI NETMF SDK のインストールを完了します。



顧客情報入力のダイアログにはユーザー名と会社名を入力します。

3. ファームウェアの確認と更新

GHI electronics 社の FEZ シリーズを購入した場合、製品に最新のファームウェアがインスト ールされているとは限りません。.NET Micro Framework では、デバイスの Flash ROM に格 納されているファームウェアを容易に更新できるのが特長のひとつでもあるので、常にファーム ウェアのバージョンを確認し、必要に応じて更新して行きます。

□解説: .NET Micro Framework のファームウェア構成□

.NET Micro Framework の基本ソフトウェアはブートローダである **TinyBooter** と、CLR イ ンタプリタを含むオペレーティング・システムの **TinyCLR** から構成されます。これらは各ライ ブラリと密接に関係しているため、開発環境、使用するライブラリを含む開発対象アプリケーシ ョンとデバイスのファームウェアのバージョンを合わせる必要があります。



図 .NET Micro Framework / .NET Gadgeteer のソフトウェア構成

3.1. ファームウェアの確認

●演習5)

次の手順で FEZ Spider の現在のファームウェアのバージョンを確認します。

手順1:PCとの接続

FEZ Spider メインボードの1番ソケット(ソケットD)に USB Client DP モジュールまたは、USB Client SP モジュールを接続後、USB ケーブルで PC の USB ソケットに接続します。



手順2:FEZ Spider の接続確認

デスクトップ上に作成された FEZ Config のショートカットから FEZ Config を起動して接続 を確認します。

| • | FEZ Config | - 🗆 | × |
|---------------------------------|---|------------|--------|
| <u>A</u> dvanced A <u>b</u> out | | | |
| Connection | USB EMX.Gadgeteer Check device for update Ping | Ŧ | |
| Firmware Updater | -Output Information- | | |
| Network Configuration | | | |
| LCD Configuration | | | |
| Deployment (Advanced) | <u>O</u> lear O | Dutput | |
| ii. | EMX_Gadget | eer is con | nected |

接続先が USB EMX_Gadgeteer となっていて EMX_Gadgeteer is connected と表示されてい れば OK です。問題がある場合にはケーブル接続とドライバのインストール状態を確認して下 さい。

手順3:バージョンとアップデート確認

FEZ Config の Connection 画面で「Check device for update」をクリックすると画面の Output Information 領域にバージョン情報が表示されます。赤字で not up to date の項がある場合には、 アップデートが必要ですので、内容に応じて次の各項の手順に進んで下さい。アップデートが不 要の場合はアプリケーション開発演習に進んでください。

| • | FEZ Config | - 🗆 × |
|---------------------------------|---|-------------------------|
| <u>A</u> dvanced A <u>b</u> out | | |
| Connection | Device USB V EMX_Gadgeteer Check device for update | |
| Firmware Updater | Output Information | |
| Network Configuration | Loader (TinyBooter) version information: 4.3.3.0 on this computer. 4.2.11.1 on this device. >>> The Loader (TinyBooter) not up to date. <<< | |
| LCD Configuration | Firmware (TinyCLR) version information: 4.3.3.0 on this computer. 4.2.11.2 on this device. >>> The Firmware (TinyCLR) is not up to date. <<< | |
| Deployment (Advanced) | Please wait for the device to reboot Done. | <u>C</u> lear Output |
| h. | EMX | _Gadgeteer is connected |

3.2. TinyCLR だけの更新

>>> The Firmware (TinyCLR) is not up to date <<<

のメッセージが表示された場合、この手順ではオペレーティング・システム**(TinyCLR)** だけ をアップデートします。ユーザーが転送したアプリケーション・プログラムは削除されるので、 注意して下さい。

手順1:アップデートの起動

FEZ Config の右側タブ「Firmware Updater」をクリックして表示内容を確認し、「Next」を クリックします。

| • | FEZ C | Config | - 🗆 🗙 |
|---------------------------------|---------------------------------------|---------------|----------------------------|
| <u>A</u> dvanced A <u>b</u> out | | | |
| Connection | | | |
| | Updating devices based on : | EMX_Gadgeteer | ¥ |
| Firmware Updater | NETMF Version: | v4.3 🗸 | |
| | Port : | USB | |
| Network | | | |
| Configuration | The device's firmware version : | 4.3.4.0 | |
| LCD | Firmware (TinyCLR) on this computer : | 4.3.6.0 | |
| Configuration | | | |
| | | | |
| Deployment | | | |
| (Advanced) | | | Next |
| ii. | | | EMX_Gadgeteer is connected |

手順2:Update 実行

アップデートに使用するファームウェアのファイル名とアドレス情報が表示されます。画面の「Next」をクリックしてアップデートを開始します。

| • | | FEZ Config | | - | | × |
|---------------------------------|--|---|--|----------------------------------|----------------|-------|
| <u>A</u> dvanced A <u>b</u> out | | | | | | |
| Connection | Select Firmware F | EMX Updater | | | | |
| Firmware Updater | ¦>¥Program Files¥0 | GHI Electronics¥GHI NETMF v4.3 SDK¥Firmwares¥EM) | { | | <u>B</u> rowse | |
| | Name | File | Base Address | Size | | _ |
| Network Configuration | Confighex Firmwarehex Firmware2hex | C¥Program Files¥GHI Electronics¥GHI NETMF v C¥Program Files¥GHI Electronics¥GHI NETMF v C¥Program Files¥GHI Electronics¥GHI NETMF v | 0x803fe000 0x00004000 0x80030000 | 0×000009 0×0006fd 0×0011f3 | | |
| LCD Configuration | | | | | | |
| Deployment (Advanced) | Friendly name | Gadgeteer Skip warning mes | ssages B | Bac <u>k</u> | <u>N</u> ext | 5 |
| i | | | EMX_ | _Gadgeteer | is conn | ected |

次に電源不足に関する警告メッセージが出ます。これは消費電力が大きいモジュールを接続したままアップデートを実行すると問題が発生する可能性があることを示すものです。メインボードに USB Client DP モジュールまたは SP モジュールを接続だけの状態では、問題ありませんので「OK」をクリックします。

| FEZ Config |
|--|
| If you are having issues with this application, try connecting external power to the board during operation. |
| OK |

以降、自動的にアップデートが進みます。

| • | FEZ Config – 🗆 🗙 |
|---------------------------------|--|
| <u>A</u> dvanced A <u>b</u> out | |
| Connection | EMX Updater |
| Firmware Updater | Device will be updated automatically! DO NOT disconnect or turn off the device! Connecting to device EMX_Gadgeteer is connected Please wait while rebooting in TinyBooter mode] |
| Network Configuration | |
| LCD Configuration | Progress |
| Deployment (Advanced) | Please wait |
| 1. | Status: No device |

完了すると次の画面が表示されます。更新されたバージョン番号を確認して下さい。



手順3:再起動とバージョン確認

アップデート完了後必ず中央の「Reset」ボタンを押すか、電源を入れ直して、再起動してお きます。

左側「Connection」ボタンをクリックして元の画面に戻り、「Check device for update」をクリックして最新版に更新されたことを確認してください。

| • | FEZ Config - | |
|---------------------------------|--|--------------|
| <u>A</u> dvanced A <u>b</u> out | | |
| Connection | Device USB V EMX_Gadgeteer Check device for update Ping | - |
| Firmware Updater | Output Information- | |
| Network Configuration | Loader (TinyBooter) version information: 4.3.4.0 on this computer. 4.3.4.0 on this device. >>> The Loader (TinyBooter) is up to date. <<< | H |
| LCD Configuration | Firmware (TinyCLR) version information: 4.3.4.0 on this computer. 4.3.4.0 on this device. >>> The Firmware (TinyCLR) is up to date. <<< Please wait for the device to reboot | |
| (Advanced) | Clear Outpu | t |
| i | EMX_Gadgeteer i | is connected |

手順4: LCD 設定

TE35 等の **LCD** を接続して再起動し、画面表示が真っ白や乱れる場合には、LCD パラメータ ーの再設定を行います。右側「LCD Configuration」タブで設定画面を表示させて「Select LCD default:」で「Display TE35」等の使用する LCD 種類を設定し「Apply to device」で設定を完 了します。

| • | | FEZ Con | fig | | - 🗆 🗙 |
|---------------------------------|--------------------------|-----------------|-----------|-----------------------------|----------------------------|
| <u>A</u> dvanced A <u>b</u> out | | | | | |
| Connection | LCD Configuration Selec | st LCD default: | Display T | E85 V | |
| | Width: | 320 | | Horizontal Sync Pulse Width | n : 4 1 |
| Firmware Updater | Height: | 240 | | Horizontal Back Porch: | 29 |
| | Output Enable Is Fixed: | True | ~ | Horizontal Front Porch: | 51 |
| Network Configuration | Output Enable Polarity: | True | ~ | Vertical Sync Pulse Width: | 10 |
| | Horizontal Sync Polarity | False | ~ | Vertical Back Porch: | 3 |
| LCD | Vertical Sync Polarity: | False | ~ | Vertical Front Porch: | 16 |
| Conliguration | Pixel Polarity: | True | ~ | Pixel Clock Rate : | 9000 |
| Deployment (Advanced) | | | | Load from device | Apply to device |
| i | | | | | EMX_Gadgeteer is connected |

3.3. TinyBooter の更新

>>> The Loader (TinyBooter) is not up to date <<<

のメッセージが表示された場合、ブートローダ(TinyBooter) とオペレーティング・システム (TinyCLR)の両方をアップデートします。ブートローダだけをアップデートすることはできま せん。ブートローダの更新時は必ず Advanced メニューを使用して、ブートローダ(TinyBooter) と OS (TinyCLR)の両方をアップデートして下さい。

手順1:アップデートの起動

FEZ Config の上部メニューの Advanced から、Loader (TinyBooter) Update を選択して、サ ブメニューの FEZ Spider を選びます。



アップデート作業のガイド画面に変わるので「Next」をクリックします。



手順2:ブートローダモードでの再起動

「Next」をクリックして画面を次に切り替えてディップスイッチの設定を確認します。



FEZ Spider メインボード中央部のディッチをブートローダモードに(Loader)に切り替えます。 ブートローダモードはディップスイッチの右から3つ(1,2,3)をON(下)に切り替えます。 ディップスイッチは小さいのでシャープペンやボールペンやの先などを使用して設定します。

画面の「See more picture」でガイド画面を切り替えて確認できます。

切り替え後は必ず中央の「Reset」ボタンを押して、再起動しておきます。その後「Update」 をクリックして下さい。

■ヒント:ディップスイッチ■

ディップスイッチの部品は上下反対についているので注意して下さい。

購入直後は、茶色の透明で薄いビニールでスイッチがカバーされている場合があります。その ような場合はカバーを剥がして下さい。



ディップスイッチは次の様に各スイッチのポジションによって役割が決まっています。



手順3:Update 実行

アップデートに使用するファームウェアのファイル名とアドレス情報が表示されます。画面の 「Next」をクリックしてアップデートを開始します。

| • | | FEZ Config | | - 🗆 🗙 |
|--|----------------------------|--|--------------|----------------------|
| <u>A</u> dvanced A <u>b</u> out | | | | |
| Connection | | Spider Updater | | |
| | Select Firmware F | iles: | | |
| the state of the s | ¦⊃¥Program Files (| x86)¥GHI Electronics¥GHI NETMF ∨4.3 SDK¥Firm∞ | ares¥EMX | <u>B</u> rowse |
| Firmware Updater | | | | |
| | Name | File | Base Address | Size |
| Network | Config hex Firmware hex | C:¥Program Files (x86)¥GHI Electronics¥GHI NE C:¥Program Files (x86)¥GHI Electronics¥GHI NE | T 0×803fe000 | 0×000009 0×0006f7 |
| Configuration | Firmware2hex | C:¥Program Files (x86)¥GHI Electronics¥GHI NE | T 0×80030000 | 0×0011db |
| | | | | |
| LCD | | | | |
| Configuration | | | | |
| | | | | |
| Deployment | | | | |
| (Advanced) | Friendly name | Gadgeteer Skip warning | messages | Bac <u>k N</u> ext |
| i | | | | Status: No devic |

次に電源不足に関する警告メッセージが出ます。これは消費電力が大きいモジュールを接続したままアップデートを実行すると問題が発生する可能性があることを示すものです。メインボードに USB Client DP モジュールまたは SP モジュールを接続だけの状態では、問題ありませんので「OK」をクリックします。

| FEZ Config | × |
|--|---|
| If you are having issues with this application, try connecting external power to the board during operation. | |
| OK | 2 |

Flash ROM 内のデータとアプリケーションが全て削除されることに関する確認メッセージで す。「はい」をクリックします。



以降、自動的にアップデートが進みます。

| • | FEZ Config | | × |
|---------------------------------|---|-------|---------|
| <u>A</u> dvanced A <u>b</u> out | | | |
| Connection | Spider Updater | | |
| Firmware Updater | Connecting GHI Bootloader Interface found at Port# COM3 GHI Bootloader Version: 1.02 I | | |
| Network Configuration | | | |
| LCD Configuration | Progress | | |
| Deployment (Advanced) | Connecting | | |
| i | | Statu | s: COM3 |

完了すると次の画面が表示されます。更新されたバージョン番号を確認して下さい。

| | FEZ Config | | | | - | | × |
|---------------------------------|--|------------------------|------------------|---------------|------------------|---------------|-------|
| <u>A</u> dvanced A <u>b</u> out | | | | | | | |
| Connection | Spider was updated succe | essfull y ! | | | | | |
| Firmware Updater | | | DI 1 | 2 2 | ні 3 | | |
| Network | Firmware updated sucessfully. Version: 4.3.4.0 TinvBooter updated sucessfully. Version: 4.3.4.0 | IKUX 4 PUY | FY 5 AOSUY | RESET | CSY 6 AITX | E 7 PUY | |
| Configuration | | 8 | 9 BY | G | 10 RY | 11† | ┛ |
| LCD Configuration | | | 12 | 13 | 14 | | |
| | | Debu | g Debug | Default | Loader T | Inyboot | ar. |
| Deployment | | | | | Star | t Over | |
| (Advanced) | | | | | <u>o</u> tar | C Over | |
| i | | | E | MX_Gad | geteer is | s conne | ected |

手順4:再起動と確認

アップデート完了後必ずメインボードのディップスイッチの右から3つ(1,2,3)をOFF(上) に切り替えて Default モードに設定後、中央の「Reset」 ボタンを押して、再起動しておきます。

左側「Connection」ボタンをクリックして元の画面に戻り、「Check device for update」をクリックして最新版に更新されたことを確認してください。

| • | FEZ Config | - 🗆 🗙 |
|---------------------------------|--|----------------------------|
| <u>A</u> dvanced A <u>b</u> out | | |
| Connection | Device USB V EMX_Gadgeteer Check device for update | Ping |
| Firmware Updater | -Output Information- | |
| Network Configuration | Loader (TinyBooter) version information: 4.3.4.0 on this computer. 4.3.4.0 on this device. >>> The Loader (TinyBooter) is up to date. <<< | |
| LCD Configuration | Firmware (TinyCLR) version information: 4.3.4.0 on this computer. 4.3.4.0 on this device. >>> The Firmware (TinyCLR) is up to date.<<< | - U |
| Deployment (Advanced) | Please wait for the device to reboot | Clear Output |
| i | E | EMX_Gadgeteer is connected |

□解説:ファームウェアとアプリケーションのバージョン整合□

デバイスのファームウェア(TinyCLR)とアプリケーションのメイン・バージョンは一致してい る必要がありますが、ブートローダは互換性があります。従って最新のブートローダを搭載して おけば、一世代前の V4.2 のファームウェア(TinyCLR)を搭載することも可能です。

しかしその場合は開発環境(ライブラリ)の設定も V4.2 にしてアプリケーション開発する必要があります。

■ヒント:TinyCLRの再ロード■

システムリセット直後等 TinyCLR の起動に時間がかかり、TinyCLR がまだロードされてい ないという趣旨で次のメッセージが表示される場合があります。その場合にはしばらく待って再 実行するか、あるいは「Advanced」メニューから強制ロードして確認することも可能です。

| | FEZ Config | - • × |
|--------------------------|--|--------------------------|
| Advanced About | | |
| Connection | Device | ▼ Ping |
| Firmware Updater | -Output Information- | - |
| Network Configuration | Loader (TinyBooter) version information: 4.3.3.0 on this computer. 4.3.3.0 on this device. >>> The Loader (TinyBooter) is up to date. <<< | |
| LCD Configuration | Firmware (TinyCLR) version information: 4.3.3.0 on this computer. Firmware (TinyCLR) Version: Not present on the device. | |
| Deployment (Advanced) | The device is in Loader mode. Click on 'Reboot CLR' under the Advanced menu to reboot the device. | <u>C</u> lear Output |
| h. | ЕМХ | (_Gadgeteer is connected |

| Reboot CLR | を選択すると、 | TinyCLR の再ロー | ドを行います。 |
|------------|---------|--------------|---------|
|------------|---------|--------------|---------|

| • | FEZ Config – 🗆 🗙 |
|---|--|
| Advanced About Loader (TinyBooter) Upd | ate • |
| Flash Sector Map A Device Capabilities A | llt+F B ✓ EMX_Gadgeteer ✓ |
| Reboot CLR A | It+R Information |
| - Disconnect | ader (TinyBooter) version information: |
| Network Configuration | 4.3.30 on this computer. 4.3.3.0 on this device. >>> The Loader (TinyBooter) is up to date. <<< |
| LCD Configuration | Firmware (TinyCLR) version information: 4.3.3.0 on this computer. Firmware (TinyCLR) Version: Not present on the device. |
| Deployment (Advanced) | The device is in Loader mode. Click on 'Reboot CLR' under the Advanced menu to reboot the device. <u>C</u> lear Output |
| in. | EMX_Gadgeteer is connected |

「CLR has been rebooted.」の表示で再ロード完了です。アップデートチェックが可能です。



4. .NETMF アプリケーション開発の基本

4.1. VC#利用前の準備

○演習6)

解説の手順に従って Visual Studio 2013 または Visual C# 2013 Express の利用前の設定をします。

□解説: VC# 2013 express 利用前の準備□

手順1:ベース・ディレクトリの作成

Visual Studio がデフォルトで作成するフォルダの場所は、フルパスの名前が長くなるために 扱いがしにくく、また後からソースコード・ライブラリを追加する際にしばしばエラーとなりま す。このような問題に対応するため、今回の演習では全て C:¥NETMF というフォルダを作成し て、そこで作業します。

最初の手順ではプロジェクト作成前に、エクスプローラーを使用して C: ドライブ直下に、 Visual Studio ソースコード管理様の NETMF ディレクトリを作成します。

手順2:行番号の設定

次ページの様に「ツール」メニュー→「オプション」を開いて「テキスト エディター」の項、 「すべての言語」の設定画面を開いて「行番号」の表示を有効にします。



4.2. 単純なアプリケーションの開発

○演習7)

解説の手順に従って Visual Studio を使用して単純な.NET Micro Framework (以降 NETMF と表記)アプリケーション・プロジェクトを作成してビルドします。

手順1:プロジェクトの作成とメインボード選択

Visual Studio を立ち上げて、新規に Visual C#コンソール・アプリケーションのソリューションを「HelloGadgeteer」という名前で作成します。作成するディレクトリは、C:¥NETMF 以下と指定します。

| | 新しいブ | ロジェクト | ? × |
|--|---------------------------|-----------|--|
| ▶ 最近使用したファイル | 並べ替え基準: 既定 | - # E | インストール済み テンプレート の検索 (Ctrl 👂 - |
| ▲ インストール済み | NET Gadgeteer Application | Visual C# | 種類: Visual C# |
| F>ブレート Visual Basic Visual C# Windows Gadgetee Micro Framework テスト Visual C++ Visual Studio ソリューション サンブル オンライン | | | A template for a Microsoft .NET Gadgeteer application allowing the user to specify a device design comprising a .NET Gadgeteer- compatible mainboard and modules, and write and debug C# software for that device. |
| 名前(<u>N</u>): GadgeteerApp | 1 | | |
| 場所(L): C:¥NETMF¥ | | | |
| シッユーション石(M): 「HeiloGadgetee | 21 | | |
| | | | OK =+7/21/ |

「OK」をクリックすると.NET Gadgeteer Application Wizard のメインボードの選択画面に 移るので、「FEZ Spider」を選択します。

またこの画面では、開発で使用する.NET Micro Framework のバージョンを、右下のプルダ ウン・メニューで「4.3」に設定します。デフォルトのバージョンが「4.2」となっているので、 この「4.3」への変更は必須の作業ですから注意して下さい。(次ページ参照)



手順2:モジュールの追加と配線

ここでは Gadgeteer のキャンバス上に部品を並べて配線する、モジュールの追加と配線作業 を行います。この作業を完了することで、アプリケーションが使用するドライバの選択と初期化 ルーチンの設定、SoC (System on a Chip)のピンアサインが行われます。

□解説:SoC のピンアサイン□

SoC は CPU チップに多数の周辺コントローラを搭載したチップです。SoC は今では組み込み システムでは必須のチップですが、限られた数のピンに機能を様々なコントローラ機能を多重化 して割り当てているため、使用する機能に応じて初期化時に面倒な設定が必要になります。.NET Gadgeteer ではこの画面の作業で自動的に行います。

次の操作は、慣れるとペイントツールの様に簡単に操作できるので、自由に試して下さい。

 ツールボックスを開いて、「Multicolor LED」と「USB Client SP」または「USB Client DP」 をキャンバスにドラッグします。
 配置したモジュールはマウスを使用してキャンバス内で自由に移動することが可能です。また「Delete」キーや右クリック・メニューで削除することも可能です。



② 配線を行います。配線はモジュールのソケットをマウスでクリックして引き出すことで、接続可能なメインボードの場所が緑色に変わるので、そこに接続することが可能です。またキャンバスの何もない場所で右クリック・メニューから「Connect all modules」を選択すると自動的に未配線の配線を行います。結線した配線はマウス選択後、右クリックや「Delete」キーで削除可能です。どこに配線しても物理配線を一致させれば動作可能です。



③ 配線完了

以下に完了した配置と配線の図を示します。配線完了後は必ず Visual Studio を操作して「すべてを保存」の操作を2回行って下さい。



■ヒント:電源モジュールの互換性■

電源供給モジュールである「USB Client SP」と「USB Client DP」は互換性があるため、どちらを使用しても、あるいは両方とも画面上で接続しなくてもシステムは動作するのですが、作業した設定を記録するためにも、使用するモジュールに合わせて配置することをお勧めします。

「USB Client DP」は大容量の 12V(7V~30V)の外部電源でも動作可能なため、ディスプレ イ等の消費電力が大きいモジュールを接続する場合や USB 供給電源が弱い場合に利用します。

手順3:コーディング

ここまでの手順で自動作成されたプロジェクトとソースコードのテンプレートを編集して FEZ Spider で動作するアプリケーション・プログラムを作成します。

C:¥NETMF¥ HelloGadgeteer 以下に「HelloGadgeteer.sln」というソリューションファイル と、C:¥NETMF¥HelloGadgeteer¥HelloGadgeteer 以下に「Program.cs」C#ソースコードが 作成されています。これを少し追加修正して簡単なアプリケーションを作成します。

① スレッド起動の追加

ProgramStarted()のアプリケーションの主処理である BlinkEver() メソッドのスレッドを作成して呼び出す次の処理を ProgramStarted()内、 Debug.Print("Program Started");の前に追加します。

Thread startThread = new Thread(BlinkEver);
startThread.Start();

② BlinkEver() メソッドの追加

ProgramStarted() メソッドの直後に、アプリケーションでの主処理である LED 点滅を永 久に実行する、以下の BlinkEver()メソッドを追加します。

```
void WaitEver()
{
    while (true)
    {
        multicolorLED.TurnGreen();
        Thread.Sleep(1 * 1000);
        multicolorLED.TurnOff();
        Thread.Sleep(1 * 1000);
        Debug.Print("LED Loop");
    }
}
```

追加部分全体を示します。



 $\blacksquare \vdash \succ \vdash : \text{Thread.Sleep} \blacksquare$

Thread.Sleep はパラメーターで指定した時間(ミリ秒)だけスリープするというメソッドで す。このように多くのクラス、メソッドは.NET Framework の C#と互換性があります。 手順4:保存とビルド

追加のコーディングが終わったらば、全てのファイルを保存後、プロジェクトのプロパティを 確認します。

ソリューション エクスプローラーで Properties を開くか、プロジェクト HelloGadgeteer を 右クリックしてプロパティを開き、アプリケーション・タブで「対象のフレームワーク」が「.NET Micro Framework 4.3」となっていることを確認します。ここの表示バージョンが「4.2」の場 合に、そのままでは実機で動作させることができませんので、プロジェクトを作り直してメイン ボードの選択時に「4.3」を選択してやり直します。

その後ビルドメニューからビルドを行います。ビルドにエラーが無い事を確認した後は、次の 演習に進んで実機で動作させます。

| HelloGadgeteer - Microsoft Visual Studio Express 2012 for Windows Desktop カイック記動(Ctrl+Q) アイル(E) 編集(E) 表示(Y) プロジェクト(E) デール(M) ツール(I) デスト(S) ウインドウ(W) ヘルプ(H) O・ 10 留 目 単 ツ ・ ペ・ト 開始 - Debug ・ Any CPU ・ 月 。 | | | |
|---|--|---|--|
| HelloGadgeteer 👳 🗙 Program.g | adgeteer Program.cs | Program.generated.cs 🛎 🗙 🗸 | ▼ リリューション エクスプローラー ▼ 単 × |
| レンジャーション | 構成(C): 該当なし v ブラットフォー | ム(M): 該当なし v | ○ ○ △ ○ · ◇ · ◇ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ ○ |
| ビルド イベント | アセンブリ名(N): | 既定の名前空間(L): | ■ ソリューション 'HelloGadgeteer' (1 プロジェクト) |
| デバッグ | HelloGadgeteer | HelloGadgeteer | Properties |
| リソース 参照パス .NET Micro Framework | 対象のフレームワーク(G): .NET Micro Framework 4.3 v スタートアップ オブジェクト(O): (設定なし) v リリース アプリケーション リソースの管理方法の指定: | 出力の種類(U): コンソール アプリケーション アセンブリ情報(1) | ▶ • ● Statute ▶ ■ Resources ▶ □ Program.gadgeteer ▶ □ Resources.resx |

4.3. 単純なアプリケーションの動作確認

○演習8)

次の手順に従ってメインボードとモジュールを接続し、実際に動作させて確認します。

手順1:組立

Visual Studioの「Program.gadgeteer」タブを開いて、前項で配置・配線した画面に従って モジュールとメインボードのソケットを配線します。

□解説:.NET Gadgeteer のソケット□

メインボードの各ソケットは1~14のソケット番号のほか、各ソケットが対応可能な機能(バ スの仕様)に従って、A~Zのアルファベットが付けられています。メインボード上のソケット に何種類かのアルファベットがプリントされているソケットは複数機能の中から選択して利用 できることを示します。

次のページに実際の接続写真を載せます。キャンバスの画像と比べてみて下さい。



手順2: PC への接続

USB Mini-B ケーブルを使用してメインボードに接続した電源部分の USB Client DP または USB Client SP モジュールから PC の USB ソケットに接続します。

手順3:接続先確認

Visual Studio から HelloGadgeteer プロジェクトの Property を開いて、.NET Micro Framework タブを開き、Deployment の Target が USB、Device が EMX_Gadgeteer になって いることを確認します。USB に接続後 2 ~ 3 秒程度で自動認識されます。

| HelloGadgeteer - Microsod ファイル(E) 編集(E) 表示(Y) プロジ | ft Visual Studio Express 2012 for Windows Desktop エクト(Ⴒ) ビルド(Ⴒ) デバッグ(D) チーム(M) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H) | |
|--|---|----------------------------|
| 🛛 🖸 - O 🛛 🔁 🗳 🗳 🖓 - C | → 】】 開始 → Debug → Any CPU → 月 → 月 → | |
| 년 HelloGadgeteer ㅋ × Program.g | adgeteer Program.cs | Program.generated.cs 🛎 🗙 🧃 |
| レデザー アプリケーション ビルド | 構成(<u>C</u>): アクティブ (Debug) マ プラットフォーム(<u>M</u>): アクティブ (Any CPU) マ | |
| ビルド イベント デバッグ | Deployment | |
| リソース | USB v | |
| SHUX | Device: EMX_Gadgeteer v | |
| | Generate native stubs for internal methods | |
| | Root name for native stub files: | |
| | HelloGadgeteer | |
| | Create stub files in this directory: | |
| | C: #NETMF#HelloGadgeteer#HelloGadgeteer#Stubs# | |

手順4:実行

「F5」キーまたはデバッグメニューから「デバッグ開始」を選択して、プログラムをデバッ グ実行します。LED の点滅と出力 Window へのメッセージ出力を確認します。

| 出力 出力元の表示(<u>S</u>): デバッグ LED Loop LED Loop | 46 47 48 49 50 51 52 53 54 55 56 } 100 % ◄ ◀ | whil { } } | le (true) multicolorLED.TurnGreen(); Thread.Sleep(1 * 1000); multicolorLED.TurnOff(); Thread.Sleep(1 * 1000); Debug.Print("LED Loop"); |
|--|---|---------------------|---|
| 出力元の表示(<u>S</u>): デバッグ LED Loop LED Loop | 出力 | | |
| 4 | 出力元の表示(<u>S</u>): LED Loop LED Loop LED Loop LED Loop LED Loop LED Loop LED Loop LED Loop | デバッグ | - ≌ ≌ ≧ |
| 出力 ローカル ウォッチ 1 | 出力 ローカル ウォ | ୬ ቻ 1 | |
| 準備完了 | 準備完了 | | |

□解説:デバッグと Deployment□

.NET Gadgeteer のデバッグ操作は次の動作を行います。

- ① USB ケーブルを介して Visual Studio からメインボードにデバッグ手順で接続します。
- ② デバッグ手順を使用してビルドしたプログラム(MS-IL 形式の.NET のアセンブリ)を転送します。(Deployment)
- ③ メインボードでは受け取ったプログラムを不揮発性の Flash ROM 内の Deployment 領域に 書き込みます。
- ④ メインボード上で動作している TinyCLR に対して、デバッグ手順で転送したプログラムの 起動を指示します。
- ⑤ TinyCLR は Flash ROM 内の Deployment 領域からプログラムを読み出してインタプリタ 実行します。

従ってデバッグ操作を起動することと、メインボードにアプリケーション・プログラムを転送 することと、プログラムを実行することは全く同じ動作となります。

なおデバッグ停止後も転送したアプリケーション・プログラムはメインボードの不揮発性 Flash ROM 内に残り、また TinyCLR はインタプリタ実行を続けます。 また、Flash ROM 内の Deployment 領域に転送されたプログラムは電源を切ってもそのまま 残るので、次回以降電源が入り TinyCLR が起動すれば Deployment 領域を自動認識して起動す るようになっています。

□解説:接続 Target と Emulator□

手順3でデバイスとの接続を確認した、プロジェクトの Property 画面では.NET Gadgeteer を使用して作成したプログラムの転送先、デバッグ接続方法を設定・確認することができます。 この画面の設定では実機が無い場合でも、メッセージ表示やグラフィック表示だけを行うプログ ラムを Visual Studio に組み込んだ Emulator を使用して動作させたり、デバッグしたりするこ とが可能です。ハードウェアのモジュールを使用するプログラムでは例外が発生します。

Emulator を使用して動作させる場合には、この「.NET Micro Framework」画面で次のよう に Target を「Emulator」に変更します。

| ער אדע גרידע | HelloGadgeteer - Microsof (E) 編集(E) 表示(Y) プロジ + 〇 間 留 単 ピーク - ペ | t Visual Studio Express 2012 for Windows Desktop エクト(P) ビルド(B) デバッグ(D) チーム(M) ツール(I) テスト(S) ウィンドウ(W) ヘルプ(H) - ▶開始 - Debug - Any CPU - 月 _テ | |
|-----------------|---|--|----------------------------|
| ¥ I | ielloGadgeteer 👳 🗙 Program.g | adgeteer Program.cs | Program.generated.cs 🛎 🗙 👻 |
| ルポックス | アプリケーション ビルド | 構成(<u>C</u>): アクティブ (Debug) 		 ブラットフォーム(<u>M</u>): アクティブ (Any CPU) | |
| | ビルドイベント | Deployment | |
| | デバッグ | Transport: | |
| | リソース | Emulator | |
| | 参照パス | | |
| | .NET Micro Framework | Device: | |
| | | Microsoft Emulator | |
| | | | |
| | | Generate native stubs for internal methods | |
| | | Root name for native stub files: | |
| | | HelloGadgeteer | |
| | | Create stub files in this directory: | |
| | | C:¥NETMF¥HelloGadgeteer¥HelloGadgeteer¥Stubs¥ | |
| | | | |

設定すると自動的に Device が Microsoft Emulator」に変更されます。この状態で「F5」キーを押すか、デバッグメニューから「デバッグ開始」を選択することで、ビルドしたプログラムを実行します。

ハードウェアを制御しないプログラムであれば、デバッグの中断、ブレークポイント、変数内 容表示等、Visual Studioのデバッグ機能が利用できることを確認します。

○演習9)

HelloGadgeteer プロジェクトで次のことを確認します。

- ① そのままエミュレータで実行して、例外発生の確認
- ② LED を制御しない様にソースコードを修正してデバッグメッセージ出力の確認

4.4. メインボードのネットワーク設定

ネットワーク・アプリケーションを作成する前に、次の手順でメインボードの Flash ROM の 領域のネットワーク情報を設定し保存します。

○演習10)

手順1:FEZ Config の起動

FEZ Spider メインボードに USB Client SP または Client DP モジュールを接続して PC の USB コネクタに接続します。LED 点滅などのプログラムが動作したままでも構いません。

FEZ Config を起動して「Network Configuration」タブをクリックします。

| • | FEZ Config - | |
|---------------------------------|---|-----------|
| <u>A</u> dvanced A <u>b</u> out | | |
| Connection | Built-in Ethernet ENC28 Ethernet RS910 WiFi | |
| | Static IP Address: 0.0.0.0 | |
| Firmware Updater | Subnet Mask: 0.0.0.0 | |
| | Default Gateway: 0.0.0.0 | |
| Network | DHCP: Enable | |
| Configuration | MAC Address: 00-00-00-00-00 | |
| | DNS Primary Address: 0.0.0 | |
| LCD | DNS Second Address: 0.0.0 | |
| Deployment (Advanced) | Generate MAC Load Built-in Ethernet Config Apply Built-in Ethernet Conf | ig |
| i | EMX_Gadgeteer is | connected |

手順2:既存設定の読み込み

「Load Built-in Ethernet Config」をクリックして既存設定を読み込みます。

| • | FEZ Config – 🗆 🗙 |
|---------------------------------|--|
| <u>A</u> dvanced A <u>b</u> out | |
| Connection | Built-in Ethernet ENC28 Ethernet RS910 WiFi |
| Firmware Updater | Static IP Address: 192.168.1.202 Subnet Mask: 255.255.0 Default Gateway: 0.0.0 |
| Network Configuration | DHCP: ✓ Enable MAC Address: 0-21-8-0-0-1 |
| LCD Configuration | DNS Primary Address: 0.0.0.0 DNS Second Address: 0.0.0.0 |
| Deployment (Advanced) | <u>G</u> enerate MAC Coad Built-in Ethernet Confie Apply Built-in Ethernet Confie |
| i | EMX_Gadgeteer is connected |

手順3:設定の変更(DHCP使用)

この画面でネットワーク関連の設定を確認・変更できます。次の画面では DHCP を有効 (Enable)にして、MAC Address の一番右の桁(一番低位の値)を 0x01 から 0x18 に変更してい ます。DHCP を使用する場合には、MAC Address 以外の項目は参照されません。

設定後「Apply Built-in Ethernet Config」をクリックして設定を保存します。

| • | FEZ Config – 🗆 🗙 |
|---------------------------------|--|
| <u>A</u> dvanced A <u>b</u> out | |
| Connection | Built-in Ethernet ENC28 Ethernet RS910 WiFi |
| Firmware Updater | Static IP Address: 192.168.1.202 Subnet Mask: 255.255.265.0 Default Gateway: 0.0.0.0 |
| Network Configuration | DHCP: MAC Address: 0-21-3-0-0-18 |
| LCD Configuration | DNS Primary Address: 0.0.0.0 DNS Second Address: 0.0.0.0 |
| Deployment (Advanced) | Generate MAC Load Built-in Ethernet Config Apply Built-in Ethernet Config |
| i. | EMX_Gadgeteer is connected |

保存確認のダイアログが出るので「OK」をクリックします。



保存完了のダイアログが出るので「OK」をクリックします。



■ヒント: .NET Gadgeteer と.NET Micro Framework の MAC アドレス■

.NET Gadgeteer と.NET Micro Framework では、Ethernet の MAC アドレス(物理アドレ ス)をメインボード上の Flash ROM 内のコンフィグレーション領域にソフトウェア・データと して保存しています。そのため利用者がユニークな番号を設定・管理する必要があります。ロー カルネットワーク上に接続する FEZ Spider 等の.NET Gadgeteer ボードが 1 台の場合には、デ フォルトの「0-21-3-0-0-18」のまま変更しなくても動作しますが、2 台以上を同時に接続する場 合には、この様に変更しておく必要があります。

またこの様に、自分で MAC アドレスの番号を入力・管理しなくても、右側の「Generate MAC」 をクリックすることで、適当な MAC アドレスを自動生成する機能も用意しています。(下図)

手順4:設定の変更(スタティック IP 使用)

以下にスタティック IP アドレスを使用する場合の設定例を示します。ご使用になる環境に合わせた各アドレス値を設定して下さい。この場合も MAC アドレスを設定し直しています。

| • | FEZ Config – 🗆 🗙 |
|---------------------------------|--|
| <u>A</u> dvanced A <u>b</u> out | |
| Connection | Built-in Ethernet ENC28 Ethernet RS910 WiFi |
| Firmware Updater | Static IP Address: 192.168.51.202 Subnet Mask: 255.255.255.0 Default Gateway: 192.168.51.161 |
| Network Configuration | DHCP: MAC Address: DNS Primary Address 19216851161 |
| LCD Configuration | DNS Second Address 192.168.51.162 Generate MAC Load Built-in Ethernet Config Apply Built-in Ethernet Config |
| Deployment (Advanced) | EMX_Gadgeteer is connected |

★注意:ネットワーク設定情報★

Flash ROM 領域に保存したネットワーク設定情報は、次の条件で容易に書き換えられます。 ① TinyCLR をアップデートした場合、デフォルト状態に設定されます。

 アプリケーション・プログラムがネットワーク設定を変更した場合は、変更した内容に書き 換えられ、そのまま残ります。

ネットワーク設定は、Flash ROM 領域の元の情報とは無関係に、アプリケーション・プログ ラムから強制的に変更して動作が可能です。例えば ROM 領域の DHCP が無効でも無視して DHCP を動作させたり、逆に常時スタティックな IP アドレスで動作させたりすることが可能で す。ROM 領域の設定内容は容易に変更されることを念頭において、利用用途に合わせて最適な 設定方法を選択して下さい。

4.5. ネットワーク・アプリケーションの開発

次の演習手順に従って.NET Gadgeteer V4.3 用のネットワーク・アプリケーション・プログ ラムを開発します。

V4.3 用のネットワーク・アプリケーション開発ではライブラリの一部に不具合があるため、 注意が必要です。

★注意:.NET Gadgeteer V4.3 固有のネットワークの問題★

.NET Gadgeteer V4.3 では V4.2 からネットワーク関連のライブラリが変更されたため、いく つかの問題が発生しています。この章ではこれらの問題に応急的に対応する様に、プログラムと 開発手順を追加しているのでご注意下さい。以降の SDK のリリース後、これらの問題が解決さ れた場合にはご自身で判断して、対応策を取り除くことが可能です。

現在判明している問題点と対応策は次の3点です。

① DHCP でのネットワーク情報取得ができない場合がある。

対応策:

ProgramStarted()メソッドの最初に Thread.Sleep(1);を挿入して、ネットワークの動作 開始を遅らせる。

スタティック IP 利用時に、ネットワークが有効化できない。

対応策:

アプリケーション動作開始時、一旦 DHCP を有効化してネットワークを動作させ、その 後スタティック IP に設定し直す。DHCP を有効化で ROM 内のネットワーク情報が書き換 わるので、DHCP を有効化の前にスタティック IP 設定情報を一時的に保存しておく。

③ Builtin ネットワークデバイスのイベントが上がらない場合がある。
 対応策:

イベントを使用せずにコーディングする。

④ Gadgeteer.Webserver ネームスペースのライブラリに例外が発生する。

対応策:

プログラムのバグが原因のため、Gadgeteer.Webserver ライブラリの DLL を入れ替える。

○演習11)

手順1:Gadgeteer.Webserver ライブラリの DLL の入れ替え

Visual Studio を一旦終了後。管理者権限で、

C:¥Program Files (x86)¥Microsoft .NET Gadgeteer¥Core¥Assemblies¥.NET Micro Framework 4.3 以下に添付 CD の Gadgeteer.Webserver43 フォルダ内のファイルを全て上書きします。

上書き更新するのは次の11個のファイルです。

Gadgeteer.WebServer.dll Gadgeteer.WebServer.pdb Gadgeteer.WebServer.xml be¥Gadgeteer.WebServer.dll be¥Gadgeteer.WebServer.pdb be¥Gadgeteer.WebServer.pdbx be¥Gadgeteer.WebServer.pe le¥Gadgeteer.WebServer.pdb le¥Gadgeteer.WebServer.pdbx

手順2:プロジェクトの作成とメインボード選択

演習7の開発手順と同様に、C:¥METMF以下に「LEDServer」という名前の「FEZ Spider」 のプロジェクトを作成します。メインボード選択画面でバージョン番号を「4.3」に設定するこ とに注意して下さい。「FEZ Spider」メインボードに次のモジュールを組み込み、配線して保存 します。組み込み・配線したキャンバスの例を以下に示します。

① USB Client SP または Client DP モジュール

- ② Smart Multicolor LED
- ③ Ethernet J11D



手順3:ソースコードの修正

ソースコードを次の様に追加修正します。このプログラムでは Flash ROM 領域内の DHCP のチェックに基づいて、DHCP の使用・未使用を決定する様に動作します。

① 10 行目として、次の using 文を using Microsoft.SPOT.Touch;の後に追加します。 using Microsoft.SPOT. Net. NetworkInformation;

□解説:NetworkInformation ネームスペース□

NetworkInformation ネームスペースは IP アドレスの様なネットワークの低レベル処理を扱うために頻繁に呼び出すクラスを含むため、using で指定しておくとプログラムがすっきりして 見通しが良くなります。

② 変数宣言を Debug.Print("Program Started");の前に追加します。

```
string ipAddress = "";
string saveIpAddress = "";
string saveSubnetMask = "";
string saveGatewayAddress = "";
string[] saveDnsAddresses = null;
GHI.Networking.EthernetBuiltIn ei = ethernetJ11D.NetworkInterface;
NetworkInterface en = ethernetJ11D.NetworkSettings;
```

```
□解説: saveIpAddress 等の変数□
```

saveIpAddress 等の変数は本来必要ないものです。今回はスタティック IP アドレス使用時に ネットワークを有効化できないという問題に対処するため、Flash ROM 領域で設定してある情 報を一時保管するために使用します。

③ Sleep と Open 処理を Debug.Print("Program Started");の後に追加します。 Thread. Sleep(1);

```
if (!ei.Opened)
{
    Debug.Print("now open J11D");
    ei.Open();
}
```

```
□解説:Thread.Sleep(1)□
```

Thread.Sleep(1);は、本来必要ない処理です。DHCPを安定動作させるために付加しています。

```
④ 続いて DHCP の設定とスタティック IP の保存処理を追加します。
            if (ei. IsDhcpEnabled)
            {
                Debug. Print("now enable DHCP");
                ei.EnableDhcp();
            }
            else
            {
                // Save Flash ROM settings
                char[] calpAddress = en. IPAddress. ToCharArray();
                saveIpAddress = new string(calpAddress);
                char[] caSubnetMask = en. SubnetMask. ToCharArray();
                saveSubnetMask = new string(caSubnetMask);
                char[] caGatewayAddress = en.GatewayAddress.ToCharArray();
                saveGatewayAddress = new string(caGatewayAddress);
                int numOfDns = en. DnsAddresses. Length;
                saveDnsAddresses = new string[numOfDns];
                if (numOfDns > 0)
                {
                    char[] dnsAddress0 = en. DnsAddresses[0]. ToCharArray();
                    saveDnsAddresses[0] = new string(dnsAddress0);
                }
                if (numOfDns > 1)
                {
                    char[] dnsAddress1 = en. DnsAddresses[1]. ToCharArray();
                    saveDnsAddresses[1] = new string(dnsAddress1);
                }
            }
```

```
PrintNetworkInfo();
```

```
□解説:else 内の処理□
```

else 内の処理は本来必要ないものです。今回はスタティック IP アドレス使用時にネットワークを有効化できないという問題に対処するため、Flash ROM 領域で設定してある情報を一時保管するために記述しています。

```
⑤ インターフェース処理を追加します。
            foreach (var ni in NetworkInterface.GetAllNetworkInterfaces())
            {
                if (ni.NetworkInterfaceType == NetworkInterfaceType.Ethernet)
                {
                    if (ni. IsDhcpEnabled)
                    {
                        ipAddress = ni. IPAddress;
                        if (ipAddress != null && ipAddress != "0.0.0.0")
                        {
                            break;
                        }
                        else
                        {
                            ni.RenewDhcpLease();
                            Thread. Sleep (500);
                            ipAddress = ni. IPAddress;
                        }
                    }
                    else
                    {
                        ni.EnableDhcp();
                        Thread. Sleep (500);
                        ni.EnableStaticIP(saveIpAddress,
                                                                       saveSubnetMask.
saveGatewayAddress);
                        ni.EnableStaticDns(saveDnsAddresses);
                        ipAddress = ni. IPAddress;
                    }
                    break;
                }
            }
```

□解説:インターフェース処理□

.NET Gadgeteer 4.3 では本来、インターフェース処理はイベントハンドラで記述することに なっているのですが、動作しないため V4.2 以前の様にインターフェース内を検索する方法で記 述しています。ここの記述を変更することで Flash ROM の設定内容とは関係なく、意図的に DHCP やスタティック IP の動作をさせることが可能です。 ⑥ デバッグ用メッセージとイベントハンドラの登録処理をメソッドの最後に追加します。
 PrintNetworkInfo();
 Debug.Print("http://" + ipAddress +
 "/LED?status=on|off|red|green|blue|white");

```
try
           {
              WebServer.StartLocalServer(ipAddress, 80);
          }
          catch (Exception e)
           {
              Debug. Print("Network error: " + e. Message);
          }
          var ledEvent = WebServer.SetupWebEvent("LED");
          ledEvent.WebEventReceived += ledEvent WebEventReceived;
⑦ デバッグメッセージ出力用 PrintNetworkInfo()メソッドを追加します。
       void PrintNetworkInfo()
       {
          NetworkInterface en = ethernetJ11D. NetworkSettings;
          Debug. Print ("-----
                                                       -----″);
          Debug. Print("IP Address: " + en. IPAddress);
          Debug.Print("DHCP Enabled: " + en.IsDhcpEnabled);
          Debug. Print("Subnet Mask: " + en. SubnetMask);
          Debug.Print("Gateway:
                                " + en.GatewayAddress);
          Debug. Print ("-----
                                                                 -″);
       }
```

□解説:PrintNetworkInfo()メソッド□

このメソッドが無くてもプログラムは動作します。これは元々、デバッグ用のメッセージ出力 処理で、デバッグ終了後削除する予定でした。しかし V4.3 の DHCP, IP アドレスの処理とイン ターフェースの起動では、様々な問題が発生しているため、トラブル発生時の情報出力のために そのまま残しています。

```
⑧ イベントハンドラ ledEvent_WebEventReceived()をクラスの最後に追加します。
```

void ledEvent_WebEventReceived(string path, WebServer.HttpMethod method, Responder responder)

```
{
    string status = responder.GetParameterValueFromURL("status");
    switch(status)
    ł
        case "red":
        multicolorLED.TurnRed();
            break;
        case "green":
        multicolorLED. TurnGreen();
            break;
        case "blue":
        multicolorLED.TurnBlue();
            break;
        case "white":
        case "on":
            multicolorLED.TurnWhite();
            break;
        default:
            multicolorLED.TurnOff();
            break;
    }
}
```

□解説:ledEvent_WebEventReceived()メソッド□

イベントハンドラで登録した、クライアントからのアクセスがある度に呼び出されるメソッド です。今回の LEDServer の主処理を行います。

手順4:ビルドとデバッグ・

ソースコードを追加修正した後は、実際にモジュールを配線して、ビルドとデバッグを行いま す。デバッグ開始前に必ず、PC と Ethernet J11 アダプタの RJ45 ソケットには、デバッグの PC 同じネットワークに接続している LAN ケーブルを接続する必要があります。

LEDServer フォルダ内に動作確認済のソリューションを置いています。参考にして下さい。

□解説:動作とデバッグ手順□

開発したネットワーク・アプリケーションは、Web サーバー機能を持っています。つまり http のリクエストに応じて、メインボードに接続した Multicolor LED を制御するというものです。

動作・デバッグ手順は、同じネットワークに接続しているマシンのブラウザから次の様に、 URL を入力するだけです。IP アドレスはデバッグ起動時に Visual Studio の「出力」Windows に表示されるので、同じものを入力します。

動作事例)

この場合では、IP アドレスが「192.168.51.202」の場合の入力例です。英字の大文字小文字 は区別するので、注意して下さい。

緑色点灯:http://192.168.51.202/LED?status=green

白色点灯:http://192.168.51.202/LED?status=on

消灯:http://192.168.51.202/LED?status=off

□解説:REST インターフェース□

このように http の GET や POST 等のメソッドを使用して、URL で一意に定まる文字列を送ったり、その結果としてのデータを取得したりする手順を REST と呼びます。この場合はブラウザの窓に入力した文字列が「GET メソッド」で送ったデータになります。一般的に REST インターフェースは、この様にブラウザがあれば容易にデータの送受信ができるため、広く利用されています。

■ヒント:プログラムの改造■

Multicolor LED は任意の色合いで点灯させたり、点灯方法を制御したりすることが可能です。 次の参考文献をヒントにして、ブラウザから制御して様々な光らせ方をする様に、プログラムを 改造することが可能です。

https://www.ghielectronics.com/docs/98/smart-multicolor-led-module

4.6. 参考文献

■ヒント:マニュアルと参考資料■

.NET Micro Framework / .NET Gadgeteer のアプリケーション・プログラムを開発する際に 参考となる API の解説やサンプルプログラムの入手先を次に示します。

●MSDN:.NET Micro Framework V4.3 API オンラインリファレンス http://msdn.microsoft.com/en-us/library/jj610646.aspx

○MSDN:.NET Gadgeteer V4.3 API オンラインリファレンス 現在公開されていません。公開までお待ちください。

●GHI Electronics 社.NET Gadgeteer メインボードとモジュール等の資料 https://www.ghielectronics.com/docs

●GHI Electronics 社開発者向けコミュニティ https://www.ghielectronics.com/community

●Socket 仕様、デザイン使用等.NET Gadgeteer 関連ドキュメント http://gadgeteer.codeplex.com/documentation

5. その他

本テキストについて、誤りや不明な点をみつけた場合には、以下宛にメールでご連絡をお願い します。

メール連絡先)

株式会社デバイスドライバーズ E-Kit 事業部 e-kit@devdrv.co.jp

以上