

簡単フリー・ツールとサンプル回路で
今すぐやってみよう!



FPGA版Arduino!!

Papilioで作るデジタル・ガジェット

見本

VGA表示/音楽再生からSDR/動画処理まで、
できることがパッと広がる

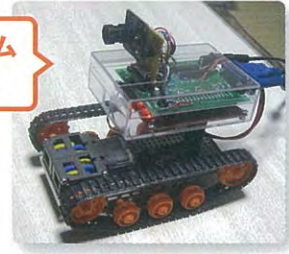
横溝憲治, 岩田利王, 土井滋貴, 平 一平 著



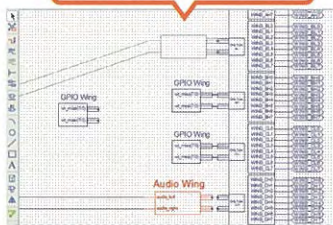
USBで
書き込みOK

FPGA
Spartan-6

リアルタイム
動画処理



音源モジュールを
グラフィカルに追加



ソフトウェア・ラジオ (SDR)



拡張
コネクタ

VGA表示

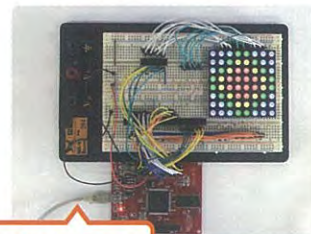


Papilio Pro
FPGAボード

64Mビット
フラッシュ



64Mビット
SDRAM



LEDマトリクス表示

デジタル
温度計



ストップウォッチ



CQ出版社

本書にボード, 部品類は付属していません

CQ出版社

第1章

Papilio の世界へようこそ

オープン・ソース FPGA プロジェクト Papilio とは

Papilio は, Gadget Factory (<http://www.gadgetfactory.net/>) が行っているオープン・ソース FPGA プロジェクトです. Papilio の主な目的は, 教育, ホビー, ラピッド・プロトタイピングです.

Papilio では,

- FPGA ボード Papilio
- 拡張ボード Wing, MegaWing
- プログラム (スケッチ) 開発環境 ZAP IDE
- FPGA 書き込みソフトウェア Papilio Loader

などを提供しており, ハードウェアの回路図, HDL ソース・コードなども公開しています.

FPGA 版 Arduino としての Papilio

Papilio の一番の特徴は, Arduino と同じ手順で FPGA 上のプロセッサ用スケッチ (プログラム) が作成できることです.

FPGA 上のプロセッサとしては, ZPUino (Zylin 社の 32bit プロセッサ ZPU ベースの SoC), AVR8 互換プロセッサが用意されており, その回路情報 (BIT ファイル) が提供されています.

Arduino IDE と同じ操作で使える ZAP IDE

プログラム開発環境としては, Arduino IDE をカスタマイズした ZAP IDE (ZPUino Arduino Papilio Integrated Development Environment) が提供されています (図 1-1) .

ZAP IDE では, Papilio に拡張ボード MegaWing を搭載した状態別に開発環境が整えられています.

例えば, 拡張ボード LogicStart MegaWing を使ってプログラミングしたいときは, ZAP IDE のメニューからその組み合わせを選択するだけですぐにプログラミングが始められます.



図 1-1 プログラム開発環境 ZAP IDE

第2章

ZPUino スケッチ入門

この章では、ZPUino で動作するスケッチ例を紹介します。スケッチ例では、Papilio Pro のほか、拡張ボード LogicStart MegaWing, RetroCade MegaWing といくつかの部品を使っています。LogicStart MegaWing, RetroCade MegaWing は一部の回路だけを使っているため、ブレッドボード上に同じ回路を組むことも簡単にできます。

サンプル・スケッチ Blink で LED の点滅

初めにサンプル・スケッチで LED を点滅させてみましょう。Papilio Pro にはユーザ用 LED が 1 個搭載されているのでこれを使用します。

サンプル・スケッチ Blink を開く

ZPUino の書き込みからスケッチを実行するまでの詳細手順は第 1 章を参照してください。

第 1 章でインストールした ZAP IDE (zap.exe) を起動します。メニューから「ツール」→「シリアルポート」を選び、USB Serial Converter B に割り当てられた COM ポートを選択します。

Papilio Pro に、ブートローダを書き込みます。ZAP IDE のメニューから、「ツール」→「Board」→「ZPUino 1.0 (32-bits) Boards」→「ZPUino on Papilio Pro (LX9), Vanilla Variant」を選択してクリックします。「Vanilla Variant」は拡張ボードを取り付けない場合の BIT ファイルを指します。「ツール」→「ブートローダを書き込む」を選択してクリックします。しばらくすると BIT ファイルが Papilio Pro に書き込まれます。

ZAP IDE からサンプルのスケッチを読み出します。ZAP IDE のメニューで、「ファイル」→「スケッチの例」→「01.Basics」→「Blink」を選択してクリックします [図 1(a)]。

新しいウィンドウで Blink スケッチが開きます [図 1(b)]。テキスト・ウィンドウにスケッチが表示されます。

サンプル・スケッチ Blink のピン番号を Papilio に合わせて変更

このスケッチの 10 行目、

```
int led = 13;
```

の設定値 13 は Arduino の LED 接続ピン番号です。Papilio Pro の LED は 49 番ピンに接続されているので値を変更します（詳細は後述）。

第3章

ZPUino へ回路を追加する (回路図編)

この章では、ZPU ベースの SoC ZPUino へ新たに回路を追加する方法を紹介します。内部回路を追加/変更ができることはマイコンを使っている Arduino にはない大きなメリットです。

Papilio シリーズの製造元 Gadget Factory では、ZPUino のトップ回路を回路図で扱えるようにデータをまとめた Papilio Schematic Library (回路図ベースのライブラリ) を提供しています。Papilio Schematic Library を使うと、HDL の知識がなくても FPGA 開発環境 ISE の回路図エディタから回路ブロックの追加や削除が可能になります。

本章では、Papilio Schematic Library を使って、ZPUino へ新たに回路を追加する方法を紹介します。

Papilio Schematic Library の準備 ～ライブラリを ISE に登録

Papilio Schematic Library は、ZAP IDE と一緒にインストールされています (最新版は Gadget Factory のサイトから zip アーカイブでダウンロードできる)。

Papilio Schematic Library には、サンプル・デザインのスケッチと ISE 用の設計データ、および使用する回路ブロックのシンボルと HDL 記述が用意されています。

Papilio Schematic Library の使用手順を、サンプル・スケッチ Audio_SID_simple (コモドール 64 の SID。SID は Sound Interface Device の略) を例に、[図 3-1](#) に示します。

ZAP IDE から Papilio Schematic Library のサンプル・スケッチ Audio_SID_simple を開きます。サンプル・スケッチのフォルダには ISE の設計プロジェクトも含まれています。オリジナルのデータを変更したくないので、サンプル・スケッチを「名前を付けて保存」で別のフォルダに保存します [[図 \(b\)](#)]。このときに ISE 設計データも一緒にコピーされます。

スケッチ上の ISE 設計プロジェクトをクリックすると、ISE が起動してプロジェクトが開きます [[図 \(c\)](#)、[\(d\)](#)]。ISE からトップ回路図を開くとライブラリが未登録なのでエラーになり、回路図のシンボルも正常に表示されません [[図 \(f\)](#)]。

ライブラリとして Papilio Schematic Library を登録し、トップ回路図を閉じてから再度開くと、回路図上のシンボルが正常に表示されます [[図 \(g\)](#)~[図 \(m\)](#)]。

さらに BIT ファイルを作成し正常終了すれば、ライブラリの登録は正常です。

ライブラリは一度登録するとほかのプロジェクトでも利用可能になります。

第4章

HDL/FPGA 開発環境 ISE 入門

前章では、ISE の回路図エディタとライブラリ (Papilio Schematic Library) を使って、ZPUino の回路の変更と追加を行いました。本章では、FPGA 開発を HDL ベースで行う際の ISE の使い方と HDL の基礎、そしてサンプルとしてストップウォッチの製作例を紹介します。使用する ISE は ISE Design Suite WebPACK 14.7 です。

ISE には、Project Navigator と PlanAhead の二つの開発環境が用意されています。Papilio で利用できる ZPUino は Project Navigator ベースで開発されているので、ここでは主に Project Navigator を使って説明します。設計ツールや FPGA ベンダが違ってても、作業フローはほぼ同じです。一つのツールで作業フローに慣れていれば、別のツールに移っても、どの作業をしているかを意識することで、それほど迷うことなく対応可能です。

ISE WebPACK チュートリアル

回路設計の流れ

FPGA 用回路設計の作業フローを図 4-1 に示します。

回路は RTL (Register Transfer Level) ソース・コードで作成します。そして、テストベンチを作成しシミュレーションで動作確認を行います。

次に、論理合成で RTL ソース・コードを回路に変換します。入出力信号のピン配置を設定し、配置配線を実施します。

配置配線に成功したら、FPGA へ書き込む BIT ファイルを作成します。

Papilio Pro を利用する場合、BIT ファイルを FPGA に書き込むには Papilio Loader を使用します。もし、ザイリンクス純正ダウンロード・ケーブルと書き込みソフトウェア iMPACT を使う場合は、第 11 章を参照してください。

最後に、実機で動作確認ができれば設計完了です。

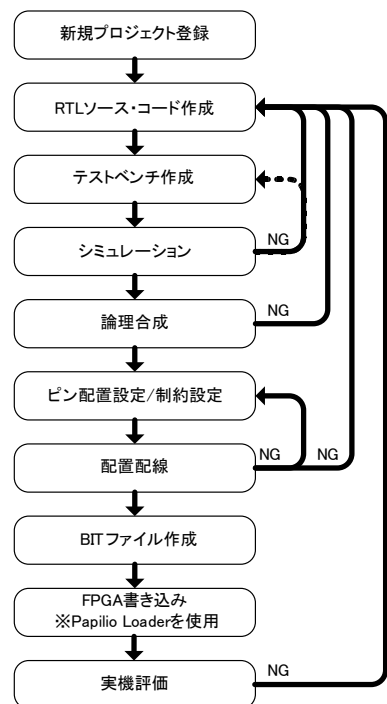


図 4-1 FPGA 用回路設計の作業フロー

第5章

ZPUino へ回路を追加する (HDL 編)

ZPUino へ回路を追加する方法として第 3 章では、Papilio Schematic Library を使って回路図ベースで新たに回路を追加する方法を紹介しました。

ここでは、HDL を使って ZPUino へ新たに回路を追加する方法を紹介します。ZPUino の HDL ソース・コード（設計データ）は公開されているので、この HDL ソース・コードへ回路を追加して、独自の SoC にすることができます。使用する FPGA 開発環境は ISE 14.7 です。

この章の内容を以下に示します。

1. ZPUino の HDL ソース・コード（設計データ）の入手
2. 設計データ内の Papilio Pro 用 ISE プロジェクト・ファイルの動作確認
3. 7セグメント LED 制御回路の追加
4. ユーザ・オリジナル設計回路の追加
5. 第 4 章で作成したストップウォッチ回路の追加
6. I²C コントローラの追加
7. LED マトリクス・ダイナミック点灯制御回路の追加

ZPUino のソース・コードの入手

ZPUino の設計データは、ZPUino のサイトのダウンロード・ページ (<http://www.alvie.com/zpuino/download.html>) の一番下にある「ZPU HDL core:」のリンク先 (<https://github.com/alvieboy/ZPUino-HDL/>) にあります。このページの右下の「Download ZIP」をクリックすると設計データ (ZPUino-HDL-master.zip) のダウンロードが開始されます (図 5-1)。

ダウンロードした ZPUino-HDL-master.zip を任意のフォルダへ解凍します。ZPUino の HDL ソース・コードは VHDL で書かれています。解凍したデータで主に使うフォルダを表 5-1 に示します。

Papilio Pro 用既存プロジェクトで動作確認

最初に、Papilio Pro 用の既存プロジェクトが ISE 14.7 で使えるか確認します。

Project Navigator を起動して Papilio Pro 用 ISE プロジェクト・ファイル（解凍フォルダ /ZPUino-HDL-master/Papilio/papilio_pro_lx9_vanilla_ise/papilio_pro_lx9_vanilla_ise.xise）を Open Project で開きます。

第6章

MicroBlaze MCS の動かし方

この章では、ザイリンクスの提供する無償プロセッサ MicroBlaze Micro Controller System (MicroBlaze MCS) を Papilio Pro 上で動かしてみます。

MicroBlaze MCS とは

ザイリンクスでは、自社の FPGA 向けに 32bit RISC プロセッサ MicroBlaze を用意しています。MicroBlaze を使用するためには、有償設計ツール Vivado Design Suite に含まれる ISE Design Suite: Embedded Edition を利用する必要があります。

一方、MicroBlaze MCS は、機能を限定して無償設計ツール ISE WebPACK で開発を可能にしたプロセッサです。MicroBlaze MCS の制約事項は、メモリ容量が 64KByte まで、ペリフェラルはシリアル通信、フィックスド・インターバル・タイマ (FIT) 4 個、プログラマブル・インターバル・タイマ (PIT) 4 個、32bit GPIO が入力用、出力用にそれぞれ 4 個が利用可能で、これ以上は追加できないことです。接続可能な外部バスも I/O バスのみです。また、アドレス・マップは固定されていて、表 6-1 のようになっています。

MicroBlaze MCS の設計フロー

MicroBlaze MCS の設計フローを図 6-1 に示します。ハードウェアの設計には Project Navigator

表 6-1 MicroBlaze MCS のアドレス・マップ

アドレス	名称
0x0~0xffff	内部メモリ
0x80000000	UART_RX
0x80000004	UART_TX
0x80000008	UART_STATUS
0x8000000C	IRQ_MODE
0x80000010	GPO
0x80000020	GPI
0x80000030	IRQ_STATUS
0x80000034	IRQ_PENDING
0x80000038	IRQ_ENABLE
0x8000003C	IRQ_ACK
0x80000040	PIT
0x80000080~0x800000FC	IRQ_VECTOR_0~IRQ_VECTOR_31
0xC0000000~0xFFFFFFFF	IO_BUS

第7章

カメラ画像無線転送装置を作ろう！

ZPUino は、SPI モードでの SD メモリーカード（以下、SD カードと省略）の読み書きに対応しています。

この機能を利用して、カメラ・モジュールの画像を Wi-Fi 対応 SD カードへ保存する装置を製作します。Wi-Fi 対応 SD カードへ保存したデータは、Wi-Fi 経由で PC などから見るすることができます。

装置の動作イメージ

図 7-1 に本装置の動作イメージを示します。カメラ・モジュールからの画像データを FPGA のメモリに保存して、この画像データを Wi-Fi 機能内蔵の SD カードへ書き込みます。Wi-Fi 経由でカードに保存された画像は、PC やタブレット端末で確認できます。装置には以下の機能を持たせています。

- カメラ・モジュールからの画像（128×128 ピクセル）をメモリに保存（ハードウェア）
- SD カードの初期化設定（ソフトウェア）
- I²C でカメラ・モジュールの設定（ソフトウェア）
- 画像データを BMP フォーマットで Wi-Fi 対応 SD カードへ書き込み（ソフトウェア）

SD カードの書き込みはファイル・システムに対応する必要がありますが、Arduino の SD カード用ライブラリを利用することで簡単に書き込み処理が行えます。

装置のハードウェア構成

回路は第 5 章で紹介した I²C コントローラを追加した ZPUino をベースにしています。図 7-2 はブロック図、写真 7-1 は実機の外観です。

ZPUino に、カメラ・モジュールからの画像データを一時的に保存する `zpuino_camera` を追加して

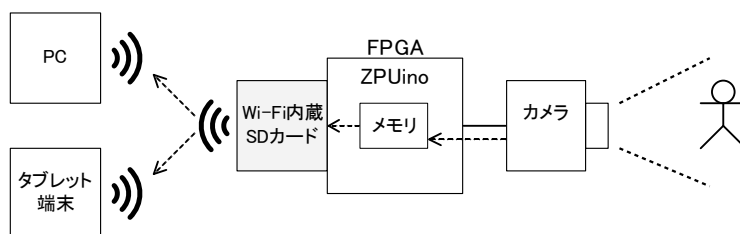


図 7-1 カメラ画像無線転送装置の動作イメージ

第8章

カメラの目を持つリモコン戦車を作ろう！

カメラ画像をリアルタイムに動き検出して標的を追いかける

小型で高性能な Papilio Pro ならではのアプリケーション

Papilio Pro は 86mm×68mm と手のひらに収まるほど小型のボードです。また FPGA (Field Programmable Gate Array) Spartan-6 LX9 を搭載しているので高速・並列な処理も難なくこなせます。従って、Papilio Pro をロボットなどに組み込めば、汎用マイコン・ボードには困難だった高度なアプリケーションも可能になりそうです。

カメラの目を持つ Papilio 戦車誕生！

写真 8-1 は今回製作した「Papilio 戦車」です。戦車には CMOS カメラ・モジュールが搭載されており、動く物体を追いかけることが出来ます。そのアルゴリズムは、カメラからの動画を Papilio Pro で解析し、左側に動きを検出すれば右のキャタピラを回し、右側に動きがあれば左のキャタピラを回すというものです (写真 8-5)。

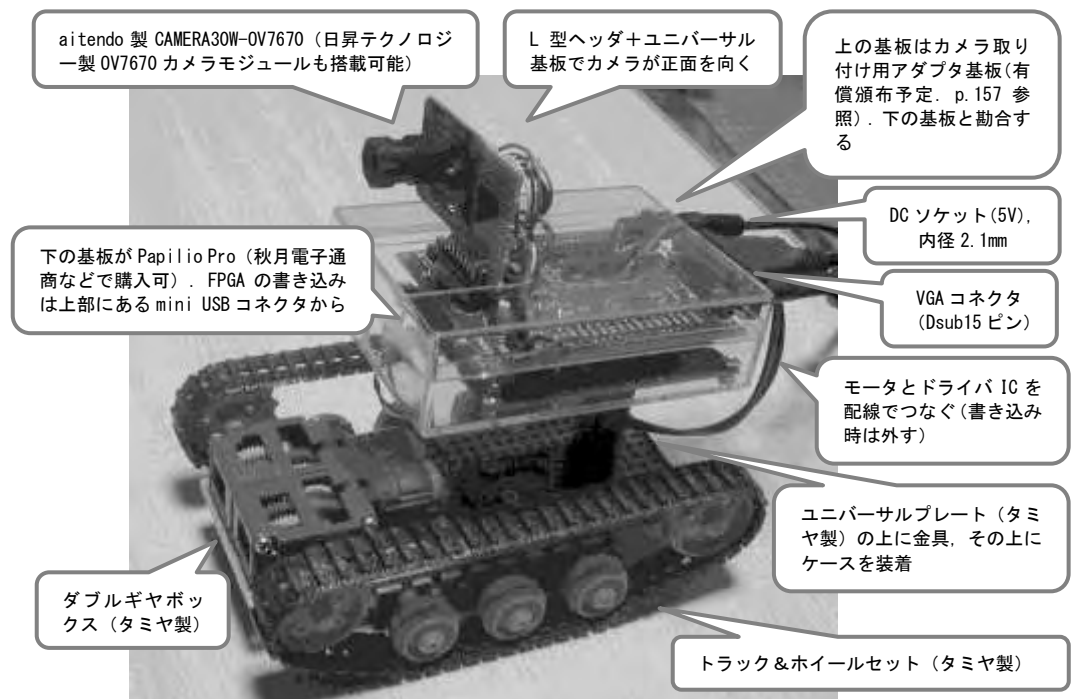


写真 8-1 カメラの目で標的を追いかけるーこれが Papilio 戦車だ！

第9章

AM ソフトウェア・ラジオの実験と製作

FPGA らしい高速な実時間デジタル信号処理を使った応用として、ソフトウェア・ラジオ (Software Defined Radio, SDR) があります。ソフトウェア・ラジオとは、ソフトウェアによる信号処理で変調/復調を行う無線技術です。ここでの“ソフトウェア”の意味は少し広くて、アナログ信号をデジタル化した後、ソフトウェア処理、ハードウェア処理を問わずにデジタル信号処理技術で処理するものを指すようです。

一般にソフトウェア・ラジオは、専用の RF (Radio Frequency) アナログ・フロントエンド回路さえ用意すれば、AM (Amplitude Modulation), FM (Frequency Modulation), PM (Phase Modulation), そして多値のデジタル変調も信号処理方法を切り替えることで復調できるようになります。

本章では、復調処理を全て Papilio Pro 上の FPGA Spartan-6 LX9 で行うソフトウェア・ラジオを紹介します (写真 9-1)。なお、フロントエンド部分も製作する関係から、比較的ラフな工作でもそれなりの性能が得られる低い周波数の AM 放送 (500k~1600kHz) を対象にします。

復習 : AM ラジオの原理

AM とは

無線で情報をやりとりするには、送信側は搬送波の上に信号波を載せた電波をアンテナから送信し、受信側はアンテナで受信した電波から信号波を取り出す必要があります。この搬送波に信号波を載せる処理を変調、電波から信号波を取り出す処理を復調または検波と呼びます。また、信号波 (変調信号波) で変調された搬送波を変調波と呼びます。

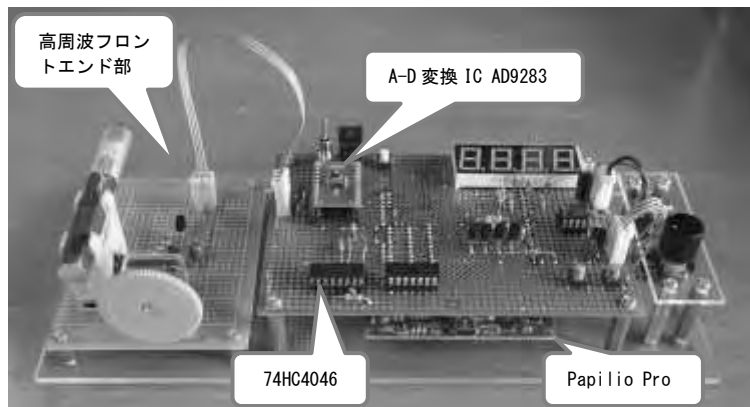


写真 9-1 製作したソフトウェア・ラジオ全景

第10章

gcc, PyMite で使う ZPUino

簡単なプログラム作成には ZPUino のスケッチで十分対応可能ですが、少し複雑なプログラムを作りたい場合は、やはり C でプログラムを書きたくくなります。

そこで、本章では ZPUino 向けプログラムを gcc で作成する方法について解説します。また、組み込み用スクリプト言語 PyMite についても触れます。本章に必要なドキュメント類を下記に示します。

- ZPU のドキュメント
http://repo.or.cz/w/zpu.git?a=blob_plain;f=zpu/docs/zpu_arch.html;hb=HEAD
- ZPUino のドキュメント
<http://www.alvie.com/zpuino/downloads/zpuino-1.0.pdf>
- ZPUino のウェブ・ページ
<http://www.alvie.com/zpuino/index.html>

開発環境の準備

使用する Linux 環境

ここでは、開発環境として Linux 環境を使用することとし、Windows 7 32bit 版で動作する Oracle VM VirtualBox 上で Ubuntu 12.04 32bit を使用しています。VirtualBox と Ubuntu のインストール方法については、参考文献を参照してください。

作業用ディレクトリとして /zpu を作成しておきます。ターミナル（端末）を起動し、次のコマンドを実行します（先頭の \$ はプロンプト、# 以降はコメント）。

```
$ sudo mkdir /zpu # ルート・ディレクトリにディレクトリ作成できるのは su のみなので sudo で実行
$ sudo chown <ユーザ名> : <ユーザ名> /zpu # 作成したディレクトリのオーナーとグループを変更しておく
$ mkdir /zpu/downloads # ダウンロード・ファイル格納用ディレクトリ
```

サンプル・プログラム（p.252 参照）を展開しておきます。

```
$ cd /zpu
$ tar xjvf proj.tar.bz2
```

展開したサンプル・プログラムの構成は図 10-1 のとおりです。

シリアル・ポートを使用するため、グループ dialout にユーザを追加しておきます。

```
$ sudo gpasswd -a <ユーザ名> dialout
```

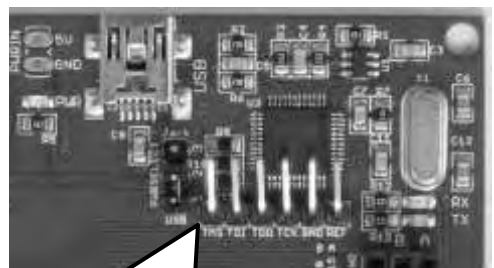
いったんログアウトして、再ログインすると有効になります。

ダウンロード・ケーブルで書き込む方法

ザイリンクス純正ダウンロード・ケーブルを使うための設定

Papilio には、BIT ファイル書き込みなどの FPGA のコンフィグレーションに使用できる JTAG 端子が、ボード上にパターンとして用意されています。この JTAG 端子は、ザイリンクス純正ダウンロード・ケーブル（品名：プラットフォームケーブル USB）を接続して使用することができます。

ただし、Papilio には JTAG 端子のパターンのみでピン・ヘッダは取り付けられていないので、ユーザがピン・ヘッダをはんだ付けする必要があります（写真 11-1）。また、Papilio は、出荷時には FTDI 社の FT2232D が JTAG 端子を制御しているので、ダウンロード・ケーブルを安定動作させるためには、FT2232D の JTAG 端子をハイ・インピーダンスにする必要があります。



はんだ付けしたピン・ヘッダ

写真 11-1 Papilio Pro の JTAG 端子パターンにピン・ヘッダをはんだ付け

FT2232D の JTAG 端子をハイ・インピーダンスにする方法

FT2232D の JTAG 機能をハイ・インピーダンスにするには、FTDI 社のチップ設定用プログラム FT_prog を使用します。FT_prog は FTDI 社のサイトからダウンロードして PC にインストールします。

操作手順は、FT_prog を起動し、虫眼鏡のアイコン「Scan and Parse」をクリック→「Device: 0」→「FT EEPROM」→「Hardware Specific」→「Port A」→「Hardware」を選択して、「OPTO Isolate」にチェックを入れて、雷のアイコン「Program Device」をクリックします（図 11-1）。

表示された「Program Devices」ダイアログで「Program」をクリックすると（図 11-2）、FT2232D の設定用 EEPROM が書き換えられ、

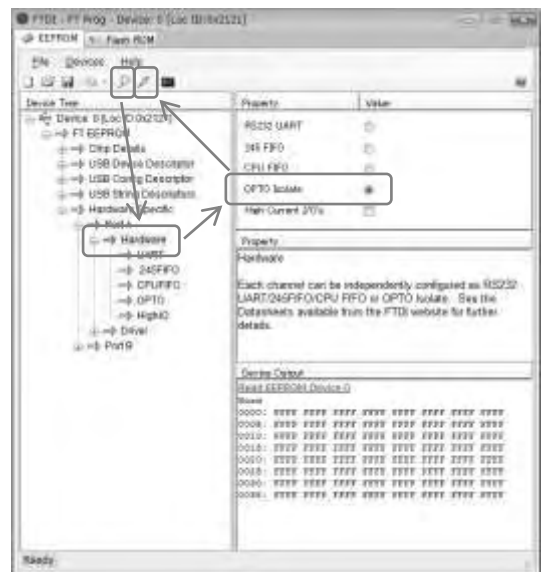


図 11-1 FT_prog の操作手順

ISBN978-4-7898-4814-5

C3055 ¥3600E

CQ出版社

定価：本体3,600円（税別）



9784789848145



1923055036000

このPDFは、CQ出版社発売の「FPGA版Arduino!!Papilioで作るデジタル・ガジェット」の一部見本です。

内容・購入方法などにつきましては以下のホームページをご覧ください。

内容 <http://shop.cqpub.co.jp/hanbai/books/48/48141.htm>

購入方法 <http://www.cqpub.co.jp/order.htm>

これがPapilioの
メリット!

ZPUinoとしての メリット

- ・ ZPUino開発ソフト
ZAP IDE
書き込みソフト
Papilio Loaderは無料
- ・ スケッチを書くと動く
- ・ 画像出力が可能
- ・ 高速動作プロセッサ
(96MHz)
- ・ 回路はBITファイルで提供
- ・ 拡張ボードごとに
専用回路を用意

組み合わせによる メリット

- ・ ZPUinoは
オープン・ソースなので
改造/回路追加が可能
- ・ 処理を高速化できる
- ・ 画像出力が
簡単に扱える
- ・ シリアル通信が簡単に扱える
- ・ ソフトウェア処理が
追加可能

FPGAボードとしての メリット

- ・ FPGA開発ソフト
ISE WebPACKは無料
- ・ 評価ボードとして容量の
大きいFPGAを搭載
- ・ ダウンロード・ケーブル不要
- ・ 汎用ピン・ヘッダを採用
- ・ 48本のI/Oピン
- ・ 低価格

開発ツール
一式が
無料で揃う!

Papilioとは：Papilioはラテン語では蝶を意味する。さなぎから蝶の姿になることを、いろいろなデジタル機器に変化できるFPGAの能力(DIGITAL METAMORPHOSIS)に重ねてPapilioと名付けられた。