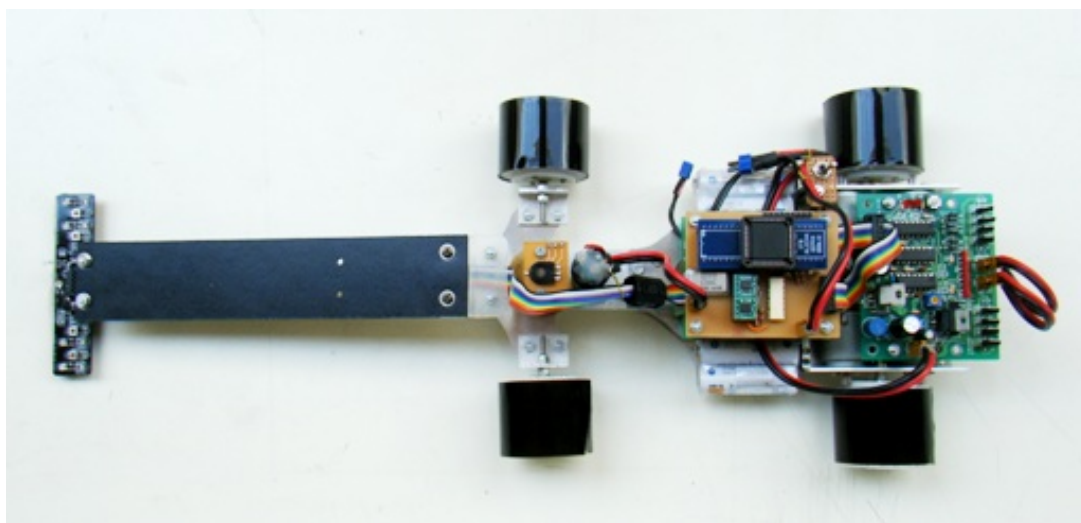


# CPLD モータードライブ基板 製作マニュアル



北海道稚内商工高等学校

立蔵 久範

## 目次

1. はじめに .....	2
2. モータードライバの必要性 .....	2
3. CPLD を用いたモータードライバ .....	3
4. CPLD を用いたモータードライバの構成 .....	4
5. 製作方法 .....	7
6. 車体への取り付け方法 .....	19
7. 開発環境のインストール .....	24
8. サーボドライバサンプル回路 .....	41
9. CPLDへの回路ダウンロード .....	46
10. サンプルプログラム .....	53
参考文献 .....	58
付録 .....	58

## 1. はじめに

近年のマイコンカーラリーは、競技の成熟とともにマシンの高速化が著しくなっています。

強力な自作サーボをはじめとする高い技術に取り組んでいるチームもあれば、ラジコンサーボとデジタルセンサマシンからのステップアップに苦労しているチームもあり、技術力の二極化がみられる状況にあります。

しかし、Basic のマイコンカーキットから自作サーボマシンへのステップアップでは、ハードウェアの製作、とりわけ制御に関わるドライバ類の製作がその難易度を上げています。

そこで、本マニュアルでは、回路構成を簡単にでき、しかも基板加工機などの設備を必要とせず、手配線で十分に製作可能な CPLD を用いたモータードライバの製作方法を紹介します。

本マニュアルは以下のような方を対象・想定し、CPLD に取り組むための環境を整えることを目的としています。

- ①電気電子・制御分野が専門外で、自作サーボ機構と車体の製作はできるが、モータードライバやプログラムの作成が難しいと考えている方
- ②これから CPLD に取り組みたいと考えている方
- ③CPLD について VHDL などの参考書を読んでいたが、実装で難しさを感じている方

本マニュアルがマイコンカーに限らず、技術向上のきっかけやヒントになることを願っています。

## 2. モータードライバの必要性

Basic のマイコンカーキットから、自作サーボとアナログセンサのマシンに移行するためには、CPU とモータードライブ基板を除いて、ほとんどを作り変えなければなりません。マクソンモータを用いたサーボや車体本体については、これまでのマイコンカーラリー参加者のレポートなどを参考に製作することができるかと思います。

サーボと車体ができると、次はドライバ類に取り組まなければなりません。走行用モーターについては、キットのモータードライブ基板がそのまま使えます。しかし、サーボの駆動には別途ドライバを用意しなければなりません。

簡単に済ませるならば、キットのモータードライブ基板を2枚積みればいいのですが、技術的な挑戦の点から考えると、やはり自作に挑戦したいところです。

また、さらなる高速化の対応を考えると、FET の選定とそれに合わせたドライバを自作できることは、大きな武器になると考えられます。

### 3. CPLD を用いたモータードライバ

自作でモータードライバを用意することにしても、次に問題になるのが、FET を正確に駆動させるための回路部分の構成です。一般にはゲートドライバという形で市販されていますが、選定や入手性を考えると自分で製作したくなります。また、キットのモータードライブ基板のようにICとコンデンサで回路を組むことも、専門でない限りハードルが高い技術になってしまいます。

そこで、回路を簡単にしながら、専門でなくても十分に自作可能で、幅広いFETにも対応可能なモータードライバを実現するためにCPLDが有効です。

CPLD はプログラムを組むようにして回路を生成できるデバイスで、電子機器の幅広い分野で使用されています。

マイコンとの違いは、パッケージ内部に実際に回路が生成されることで、電源が切れると回路が消えてしまったり、プログラムを読み込む必要がないので、高速動作が可能になっている点です。さらに、ロジックのICを組んで、回路に変更が必要になった場合には、基板や配線そのものを作り変えなければなりませんが、CPLDではパソコンでプログラムのように回路を組みなおし、書き込むだけで変更が可能です。

全国大会で上位を占めるチームに限らず、ロボット競技では一般的になってきているCPLDの技術は、今後の可能性が大きいと言えます。



#### 4. CPLD を用いたモータードライバの構成

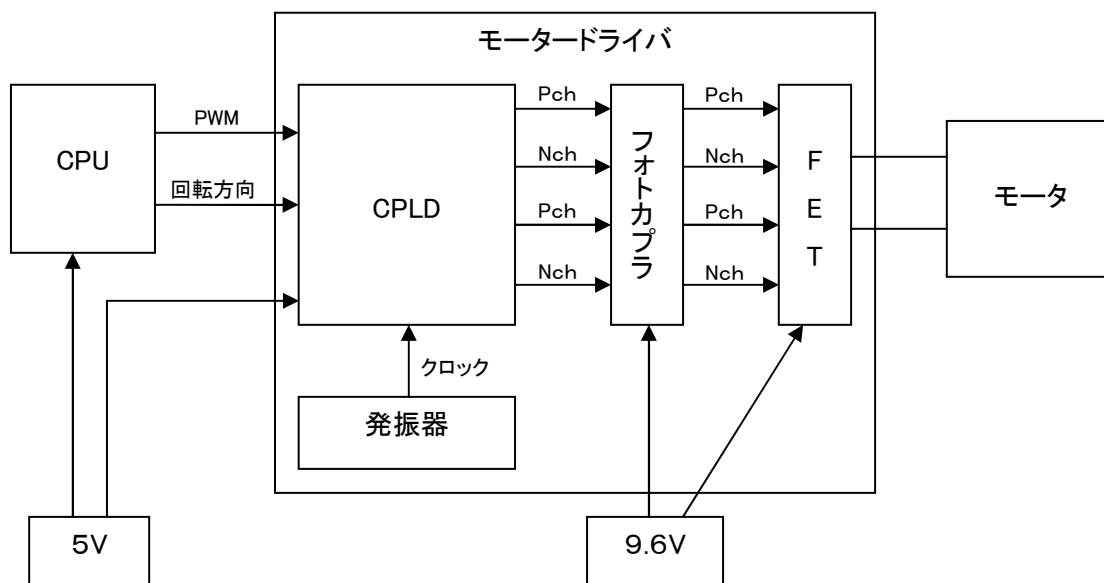


図4. 1 CPLD を用いたモータードライバの構成

上の図4. 1に CPLD を用いたモータードライバの構成を示します。

各部品の役割は以下の通りです。

①CPU

サーボを動作させる PWM と回転方向の信号を出力します。

②発振器

デットタイムを計るための基準となる信号を出力します。CPLD ではこの信号の回数をカウントし、時間としています。

③CPLD

発振器の信号から時間を計測し、回転方向が変化するとき設定した時間だけ FET を OFF にします。

④フォトカプラ

CPLD までの5V の信号を FET を駆動できる電圧に変換します。FET の中にはマイコンから直接駆動できるようなものもありますが、大きな電圧がかかる場所なので、安全のために変換し、CPU 側と FET 側を絶縁します。

⑤FET

モータを駆動します。

#### 4. 1 CPLD を用いたモータドライバの動作

- ①CPU から PWM と回転方向が出力されます。
- ②回転方向の信号で決められた FET の組に回路が選択されます。
- ③PWM の ON 幅だけ FET を ON します。(図4. 2)

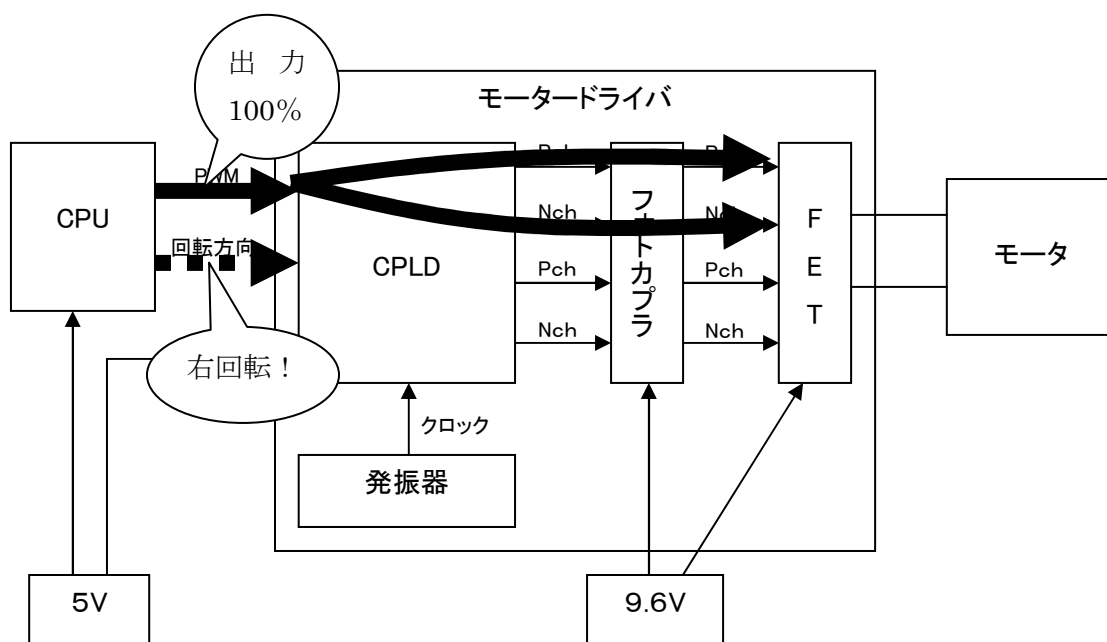


図4. 2

- ④回転方向が変わると、クロックをカウントしながら、その間(デッドタイム)はすべての FET を OFF にします。(図4. 3)
- ⑤指定のクロックをカウントしたら、再び PWM の信号を待って FET を ON します。(図4. 4)

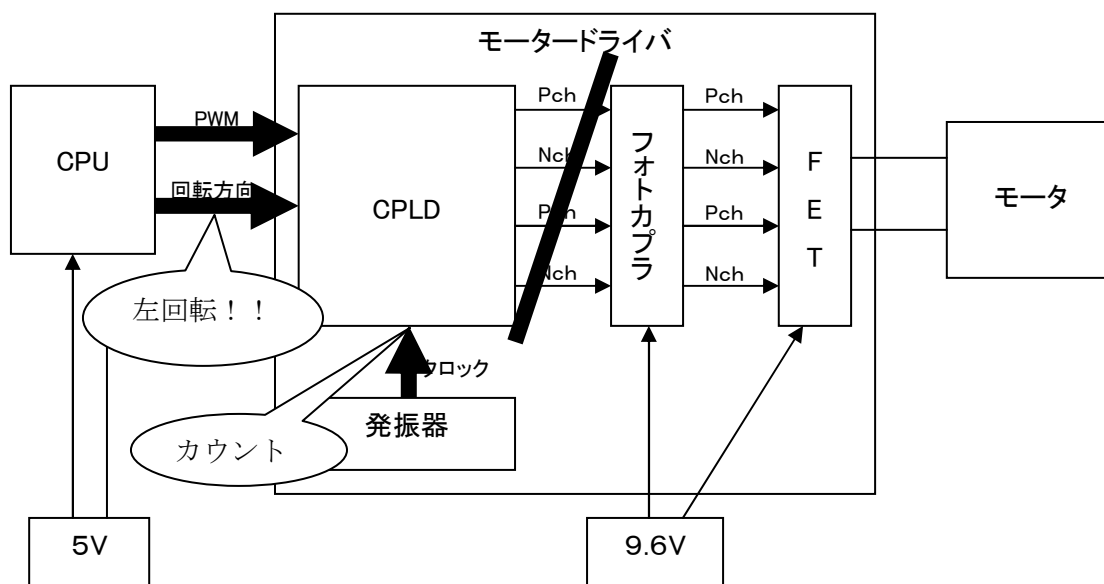


図4. 3

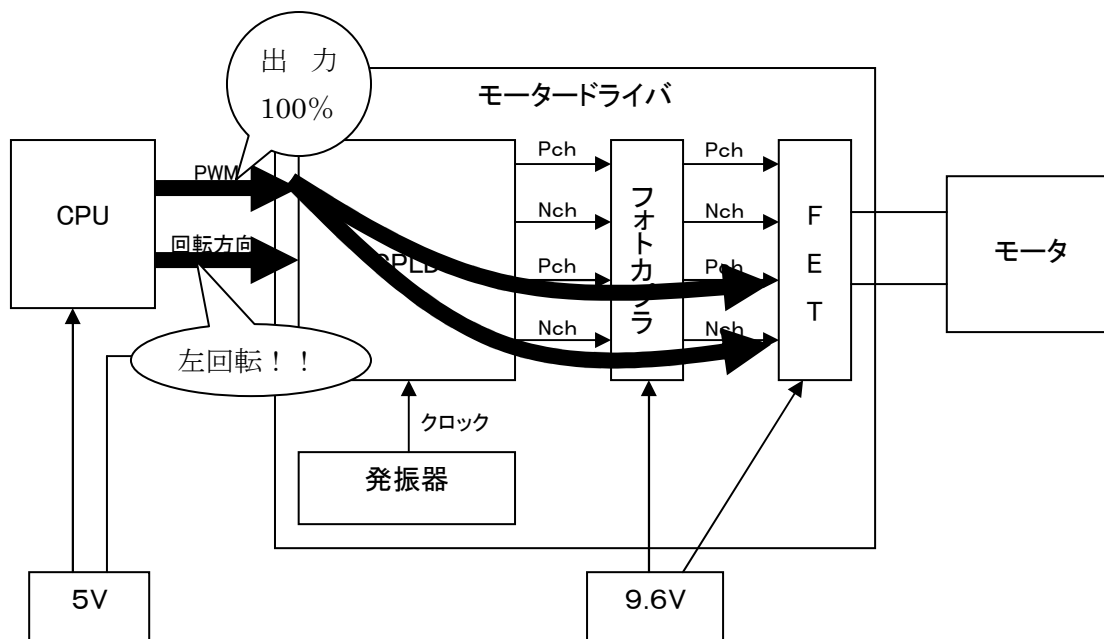


図4. 4

## 5. 製作方法

### 5.1 基板と部品の準備

下の図5. 1に、本マニュアルで製作する CPLD モータードライブ基板のハンダ面からみた配線図を示します。図中の太線が配線回路とハンダ面のジャンパで、細線は部品面の部品およびジャンパを示しています。

ここでは、図5. 1の配線と同様のものを卓上CNCで生基板から削りだした図5. 2の基板を使い、CPLD モータードライブ基板を製作します。

図5. 1と同じく配線ができれば、エッチングの環境や、基板加工機などがなくても、穴あきのユニバーサル基板で作ることができます。ユニバーサル基板で製作した例を図5. 3に示します。

また、図5. 4にはCPLD モータードライブ基板の回路図を示します。これを基にしてブレッドボードなどで試作することも可能です。

表5. 1には使用部品の一覧を示します。数量欄のカッコ内は入手先の参考です。

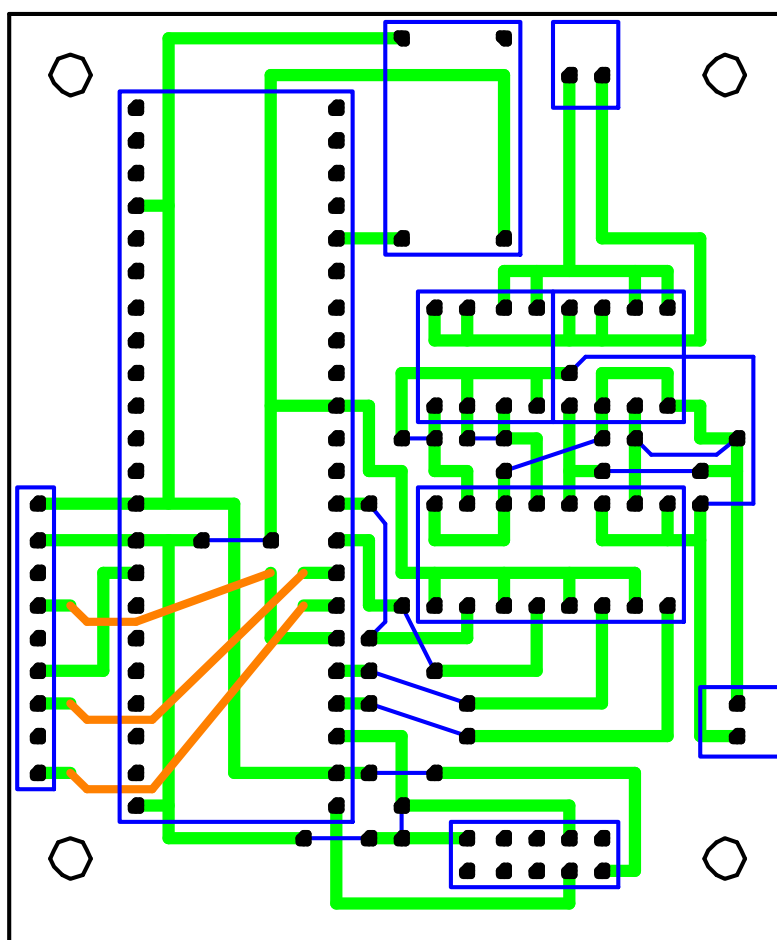


図5. 1 ハンダ面からみた配線図

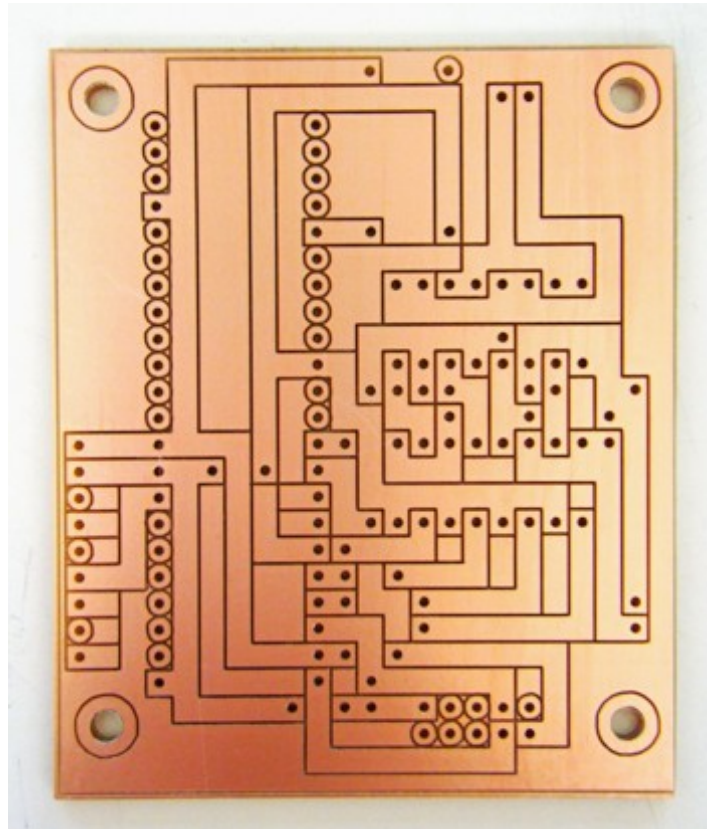


図5. 2 組立て前の CPLD モータードライブ基板

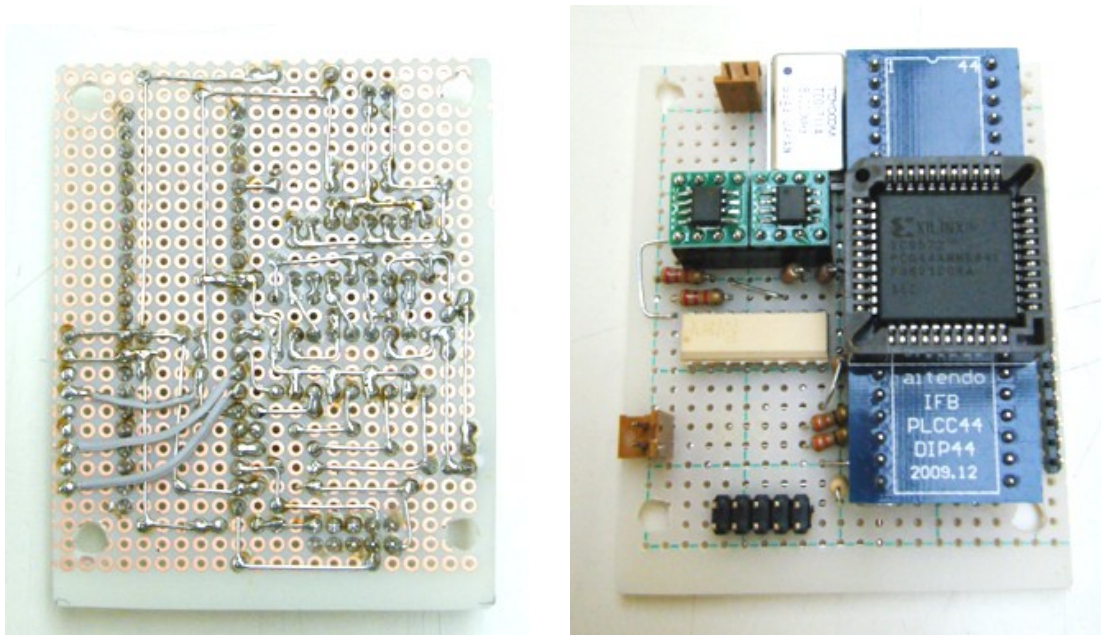


図5. 3 ユニバーサル基板を使った製作例

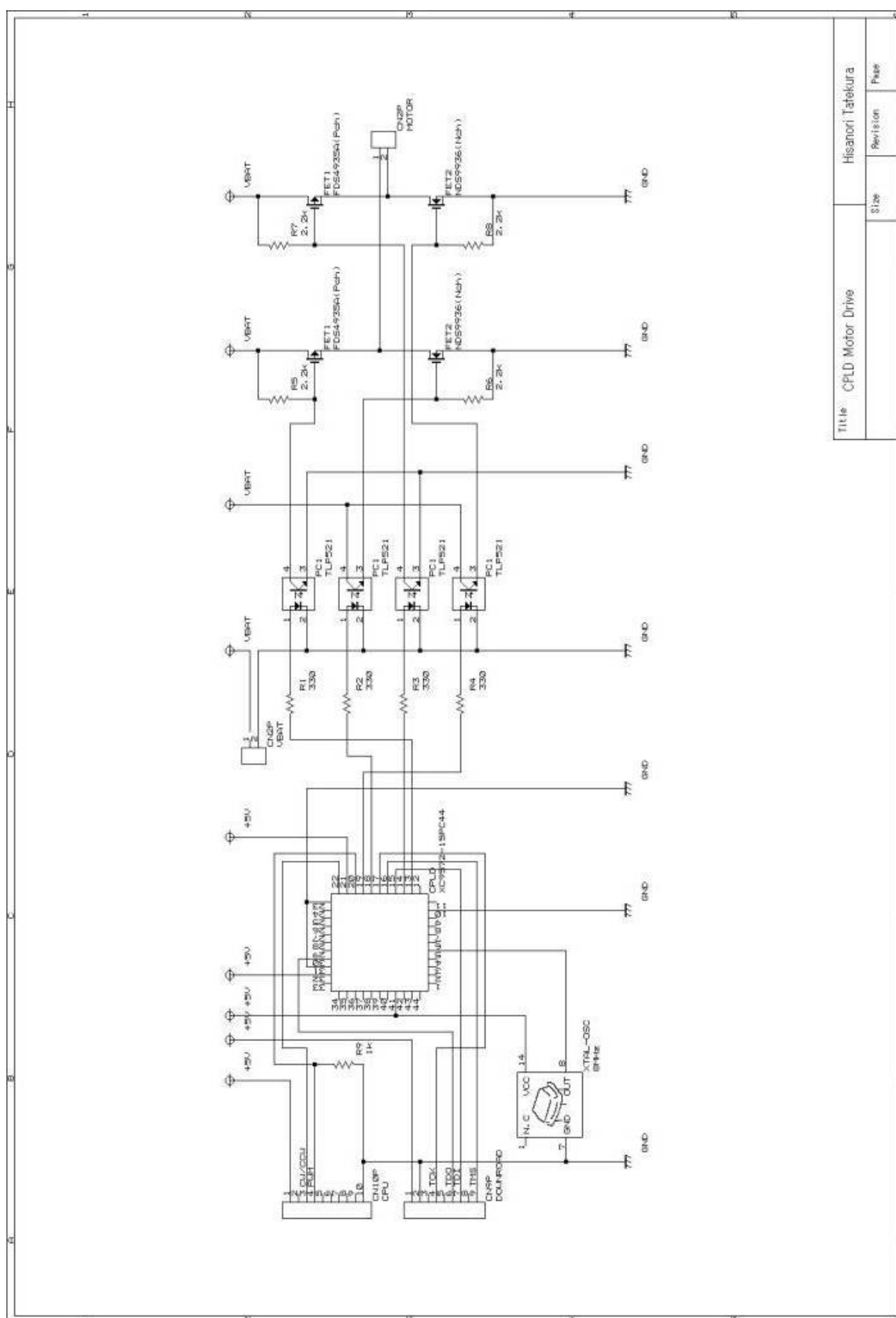


表5. 1 部品一覧

名称	形式	写真	メーカー	数量
CPLD	XC9572-15PC44 PLCC パッケージ (XC9572XL の 3.3V 版 ではない)		Xilinx	1
PLCC-DIP 変換 基板 (44P)	IFB-PLCC44-DIP44		aitendo ( <a href="http://www.aitendo.co.jp">www.aitendo.co.jp</a> )	1 (@680 aitendo)
デュアル Pch チッ プ MOS-FET	FDS4935A (30V 7A)		Fairchild Semiconductor	1 (5 個入り¥200 秋月)
デュアル Nch チッ プ MOS-FET	NDS9936 (30V 5A)		Fairchild Semiconductor	1 (5 個入り¥200 秋月)
SOP8 (1.27mm) DIP 変換基板	AE-SOP8-DIP8		秋月電子	2 (9 枚入り¥100 秋月)
抵抗 330Ω (橙橙茶金)	1/4~1/8W		各社	4
抵抗 2.2kΩ (赤赤赤金)	1/4~1/8W		各社	4
抵抗 1kΩ (茶黒赤金)	1/4~1/8W		各社	1
クリスタル オシレータ	8MHz(長方形)		TOYOKOM 他	1 (@100 秋月)
4 回路入り フォトカプラ	TLP521-4		東芝セミコンダクター	1 (@100 秋月)

表5. 1 続き				
名称	形式	写真	メーカー	数量
ピンヘッダ(オス)	2×5(10P)		各社	1
ピンヘッダ(オス)	1×9		各社	1 ※1
細ピンヘッダ	1×4		各社	4 (@5 秋月) ※1
ピンソケット(メス)	1×22 ピン分 ※2		各社	2 ※3
ピンソケット(メス)	1×8		各社	2 (@30 秋月) ※4
2P オスコネクタ ストレート	IL-2P-S3EN2(錫メッキ)			2
10P メスコネクタ	PS-10SM-D4P1-1D または PS-10SM-D4P1-1C (右写真)		日本航空電子工業	6
フラットケーブル	ピッチ 1.27mm 10 心		沖電線他	合計 640mm ※5

※1 20P のものを切って使うと得です。

※2 最長 20P しかないので、切って付け足します。

※3 加工が難しい場合は、1×10(@40 秋月)を1個と1×6(@20 秋月) 2個を組み合わせて 22P とします。

※4 加工できる環境があれば、20P のものを切って使うと得です。

※5 本マニュアル車両の参考長です。センサ・CPU 間 420mm、CPU・ドライブ基板間 170mm、CPU ポート A 結線用 50mm です。



## 5.2 組立て

表5. 1の部品を下の図5. 5の部品配置図に従って取りつけてください。フォトカプラとクリスタルオシレータは角部に黒丸印がありますので、この向きが図と同じになるように注意して取りつけてください。ジャンパは導線または抵抗などの余った足を使ってください。

部品面からの取り付けが終わったら、ハンダ面で図5. 6のジャンパ3本を取りつけてハンダ面の作業は完了です。ハンダ面の完成図を図5. 7に示します。

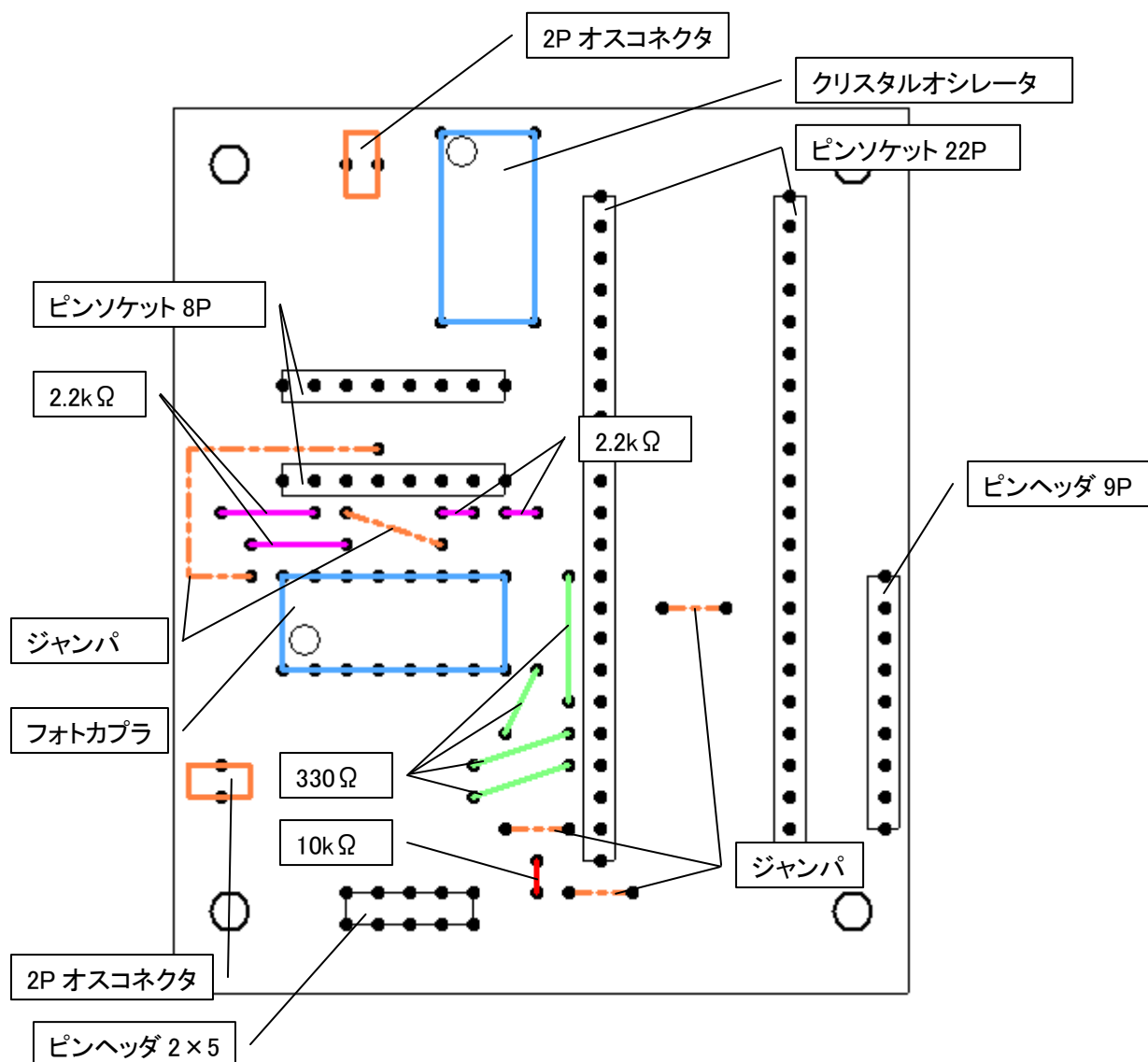


図5. 5 部品配置図

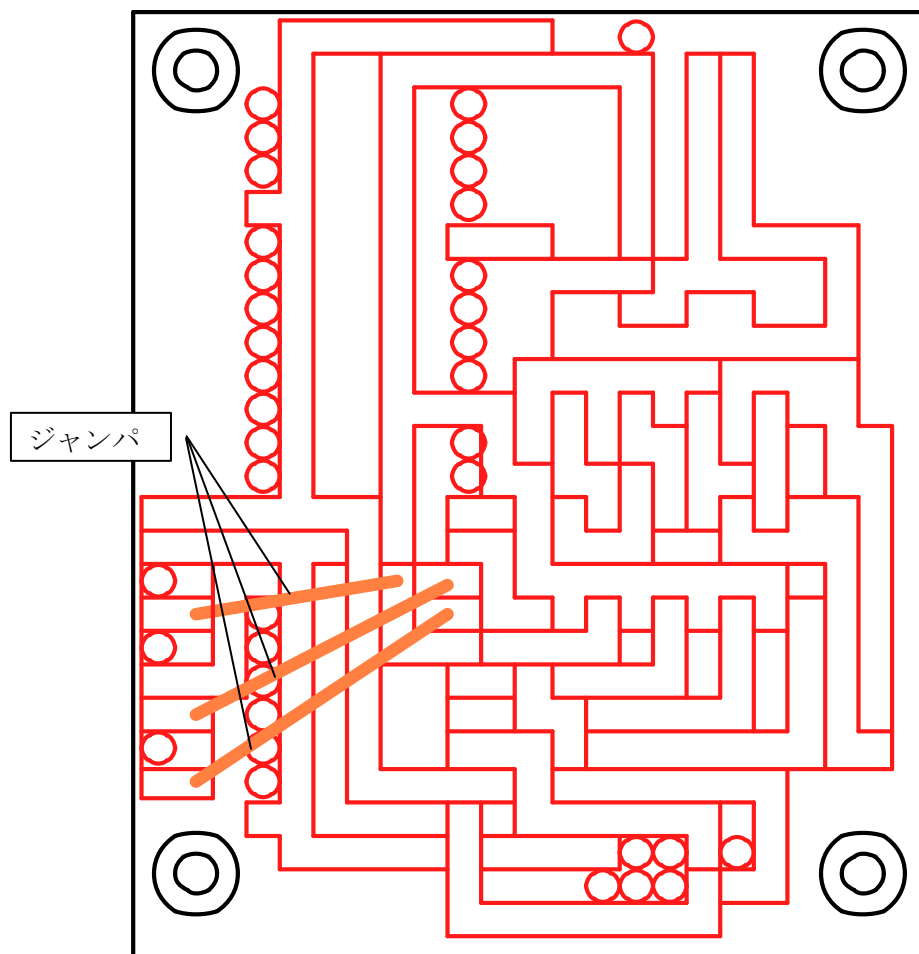


図5. 6 ジャンパ配線図

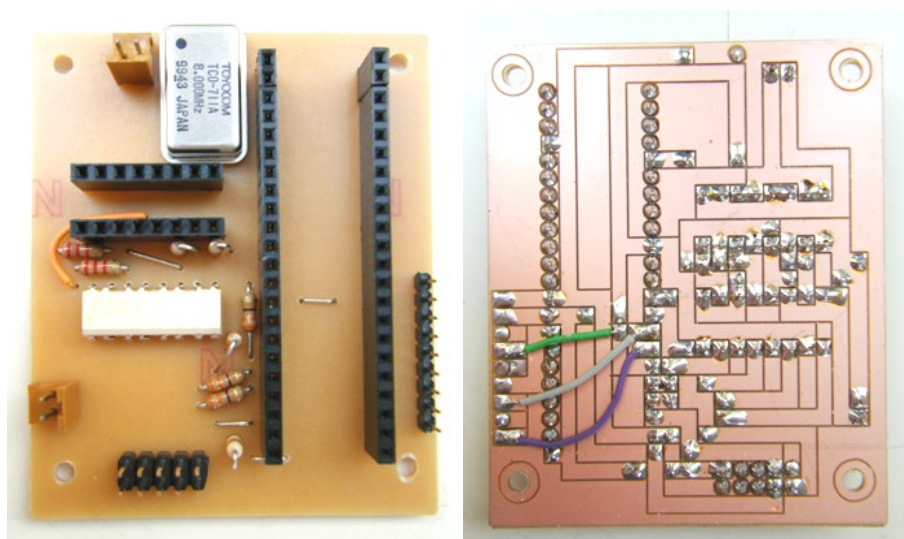


図5. 7 ハンダ面完成図

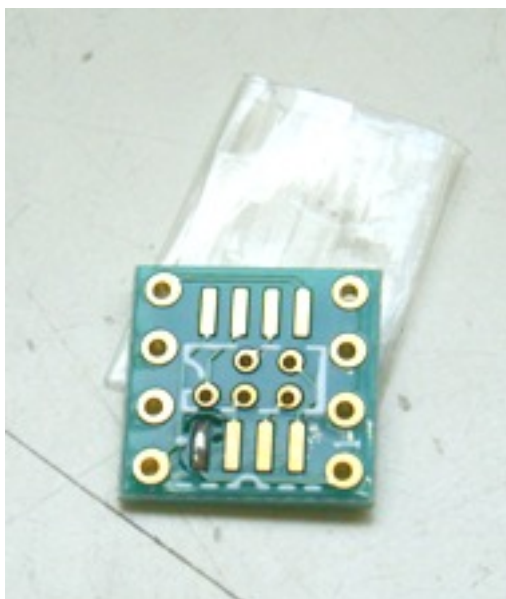
### 5.3 FETの取り付け

SOP8(1.27mm) DIP 変換基板にデュアル Pch チップ MOS-FET(FDS4935A)とデュアル Nch チップ MOS-FET(NDS9936)をハンダ付けします。

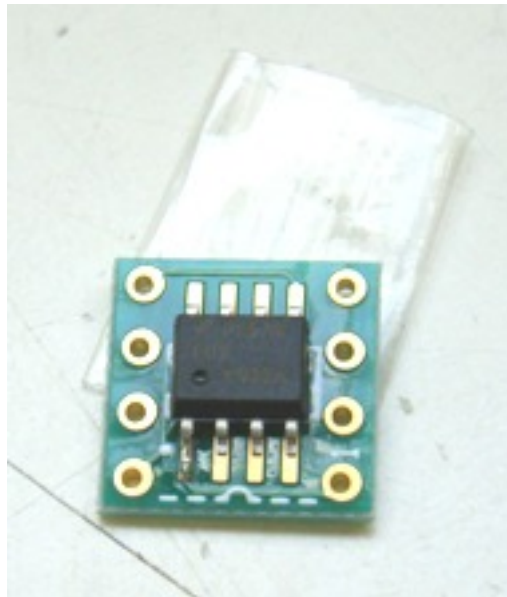
①基板を用意し、テープなどで作業台に固定します。



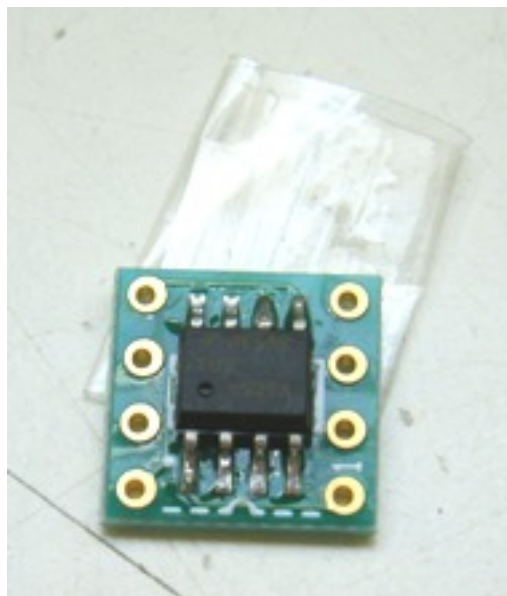
②1ピンにハンダを少量付けます。



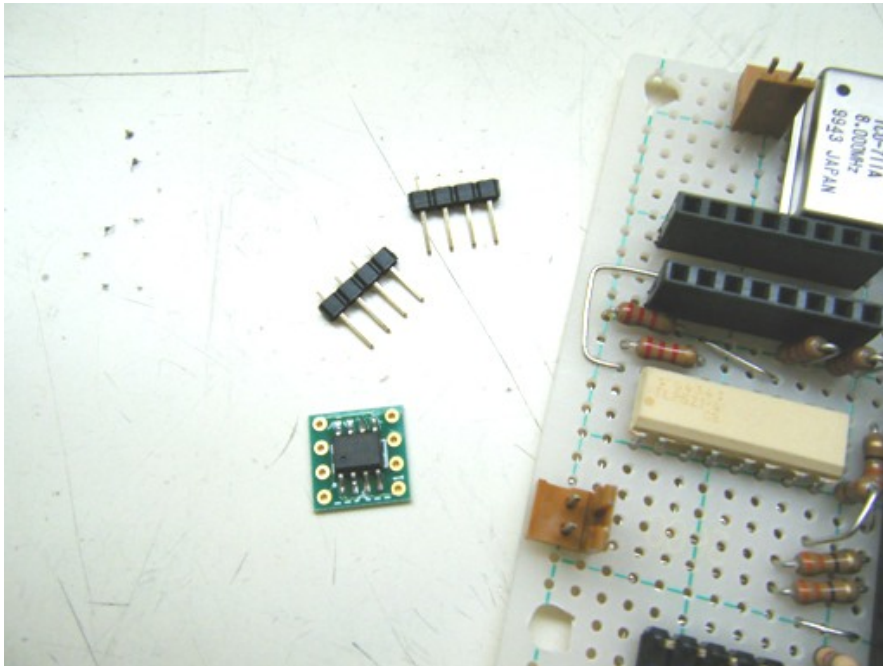
- ③基板の白点とFETのくぼみ(1ピン)が一致するように注意して、付けたハンダを温めながら、FETの場所を決めます。



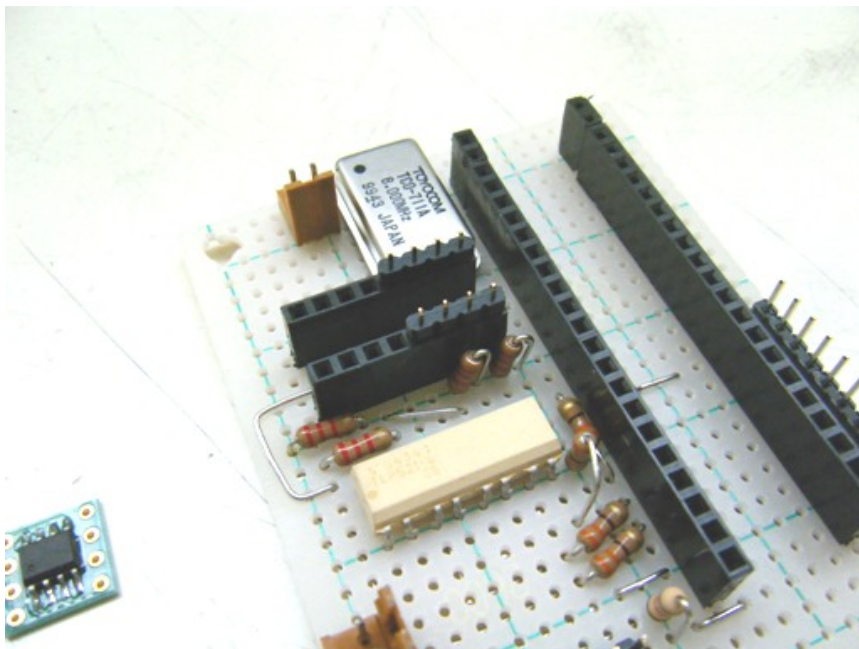
- ④残りのピンをハンダ付けします。



⑤細ピンヘッダ1×4を用意します。

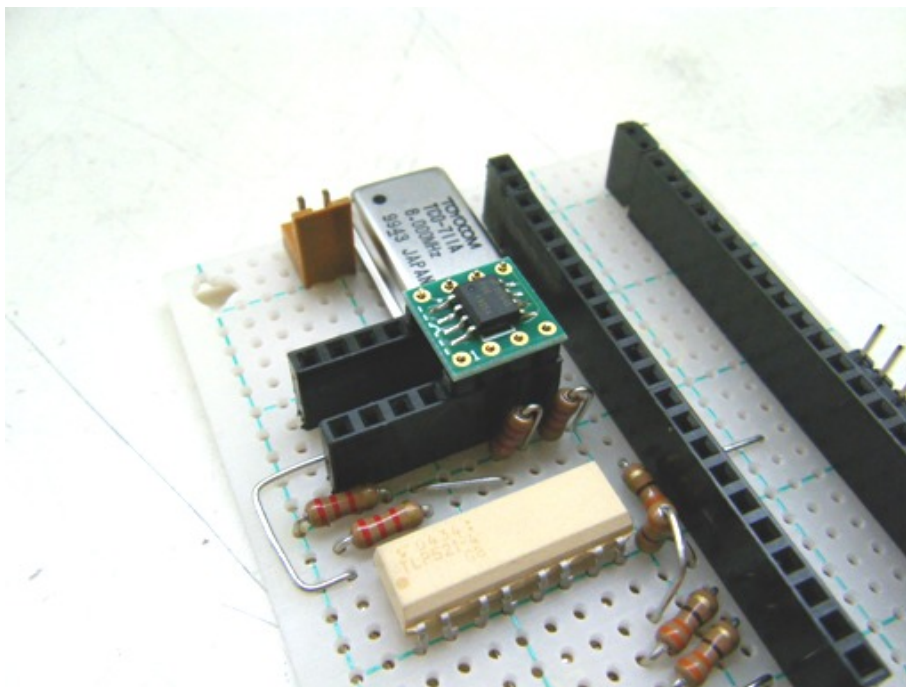


⑥ピンヘッダをピンソケットに挿します。

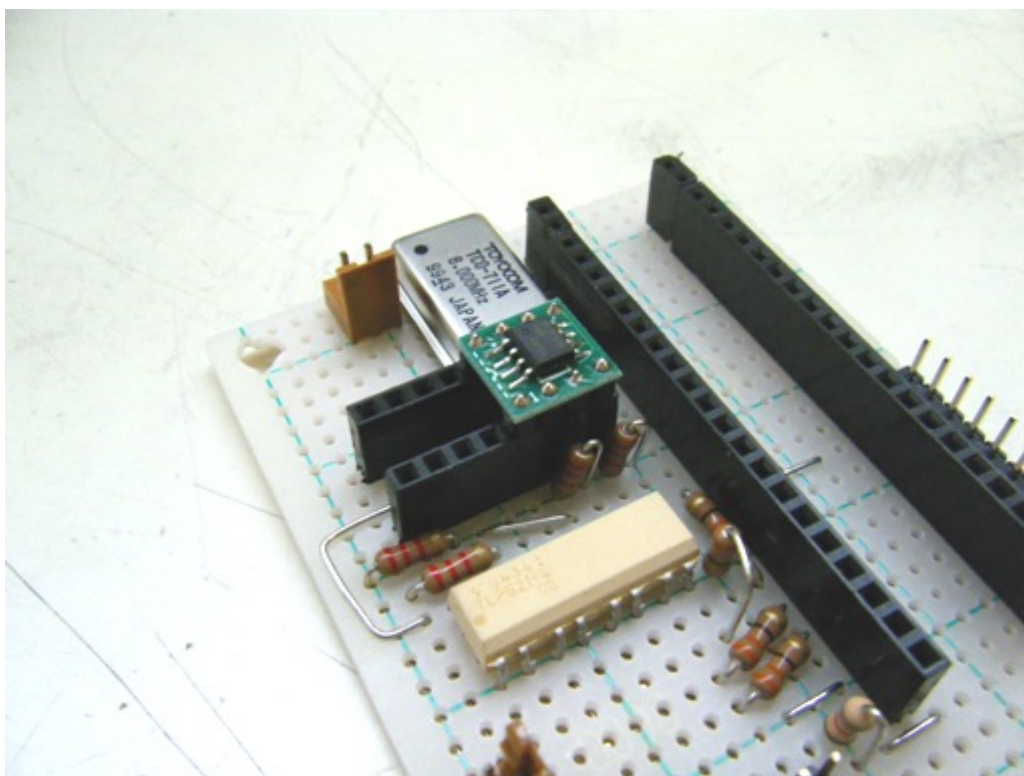




⑦FET基板をピンヘッダの上に載せます。



⑧ピンヘッダとFET基板をハンダ付けします。



## 5.4 組付け

最後にFETとCPLDを基板に組付けて、CPLDモータードライブ基板は完成します。

下の図5. 8にFETとCPLDの組付け向きを示します。CPLDは角が欠けている方を PLCC-DIP 変換基板のソケットの欠けている角に合わせて押し込みます。

完成図を図5. 9に示します。

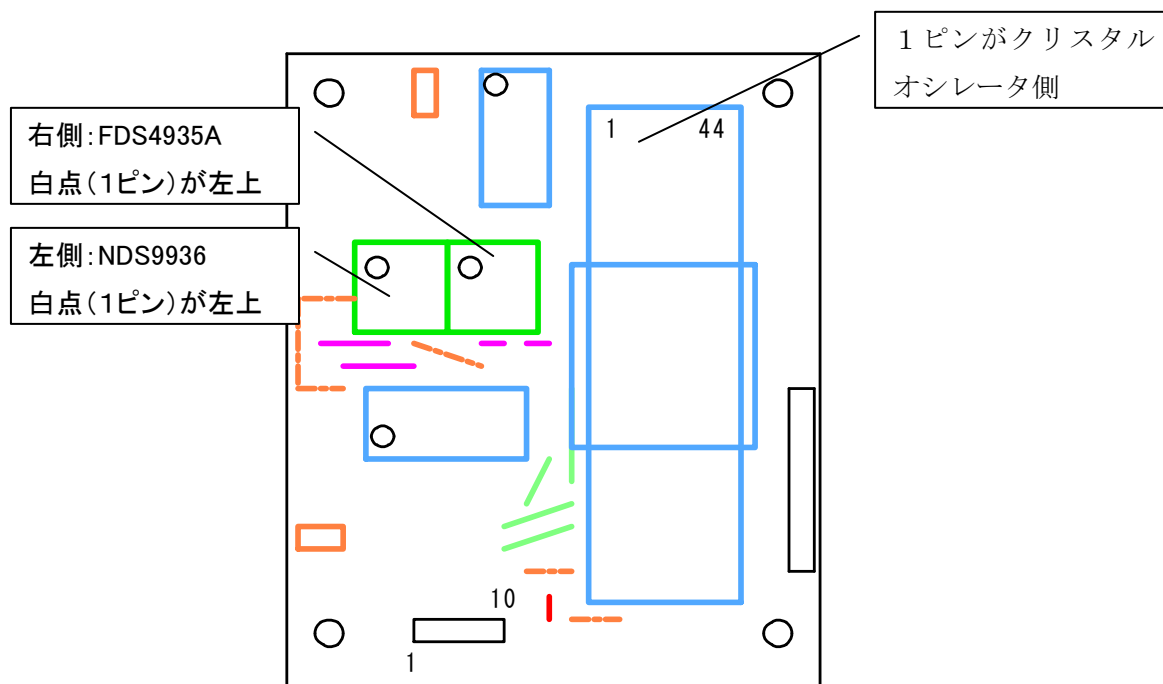


図5. 8 FETとCPLDの組付け向き

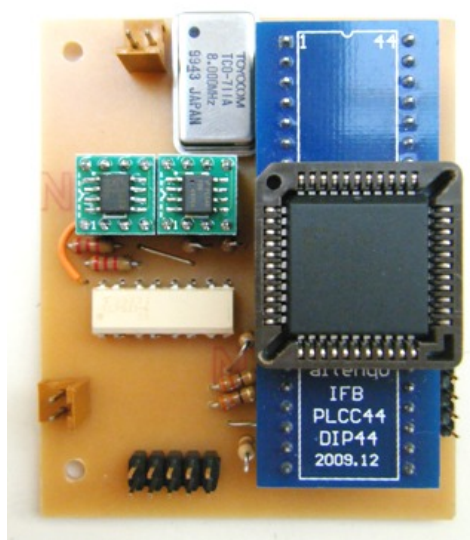


図5. 9 完成図

## 6. 車体への取り付け方法

本章では、車体に CPLD モータードライブ基板を実装する方法を紹介します。車体構成は以下の通りです。

センサ: TypeS

サーボ: マクソン RE16 を用いた自作

CPU: H8-RY3048Fone

走行用モータードライバ: モータードライブ基板 Ver3(追加キット付)

サーボモータードライバ: CPLD モータードライブ基板

エンコーダ: ロータリーエンコーダキット

### 6.1 センサ・CPU 接続ケーブルの製作

本提案では、結線による配線が必要です。実際に各基板を接続するフラットケーブルを解いて下の図6.1のように結線します。1ピン側(5V)を茶、10ピン側(GND)を黒とします。図中のコネクタは省略しても構いません。

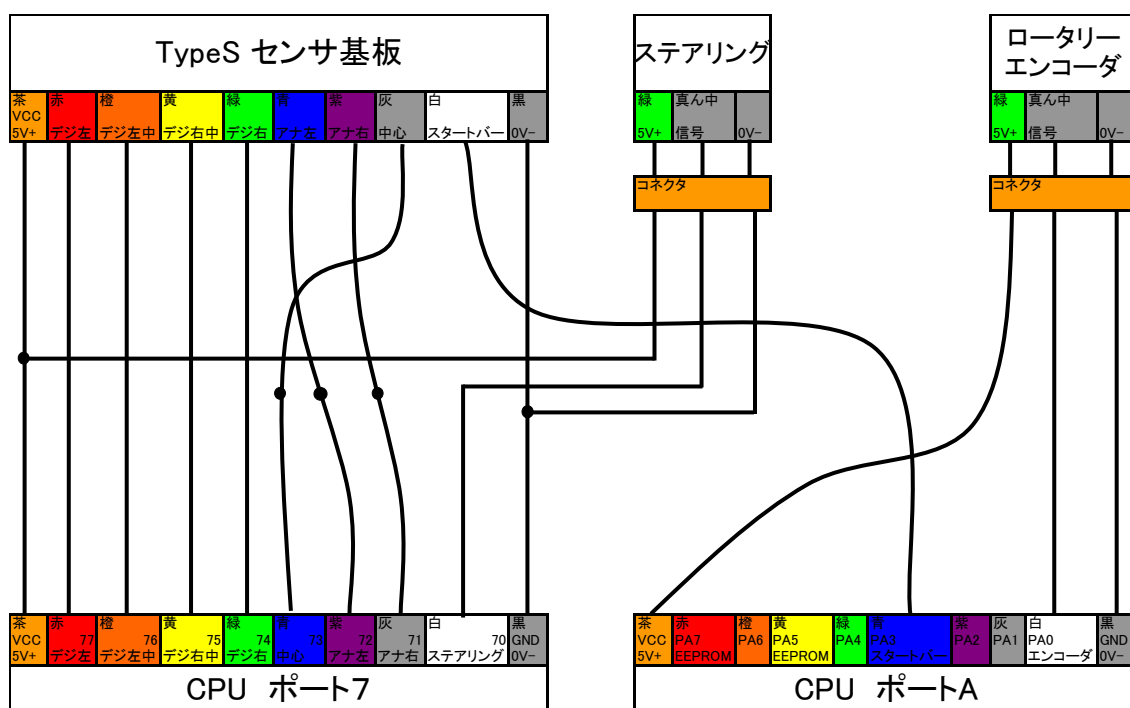


図6.1 センサ・CPU 接続ケーブル結線図



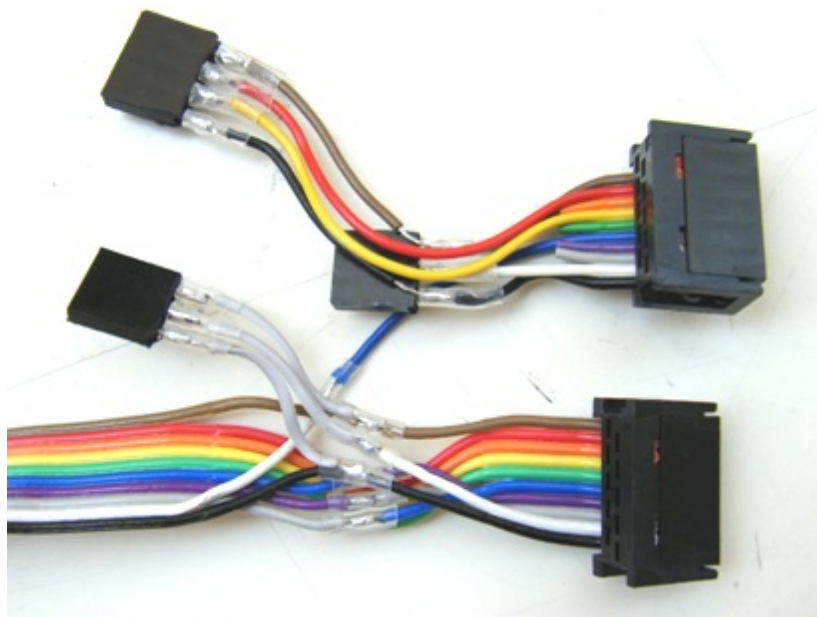


図6. 2 ケーブル製作例

上の図6. 2は、図6. 1センサ・CPU 接続ケーブル結線図をもとに作った実際のケーブルです。図6. 2ではEEPROM用の4ピンのソケットを追加しています。このようにまとめれば、見やすく確実に接続ができますので、参考にしてください。

## 6. 2 CPU・モータードライブ基板接続ケーブルの製作

モータードライブ基板 Ver3(追加キット付)と CPLD モータードライブ基板を接続するケーブルを下の図6. 3のように作成します。使用するフラットケーブルは、それぞれの基板に十分届くよう、長さに注意し、ケーブルの茶色線をコネクタの三角印に合わせてください。また、モータードライブ基板 Ver3 のコネクタは PS-10SM-D4P1-1D を使うと加工無しでそのまま圧着できますが、PS-10SM-D4P1-1C を使うときは、ストッパ部で断線しないように図6. 4のように削ってください。

PS-10SM-D4P1-1D と PS-10SM-D4P1-1C はストッパ部の有無が違います。マイコンカーキットに入っているのは、ストッパ部がある PS-10SM-D4P1-1C ですので、入手しやすい方を使ってください。

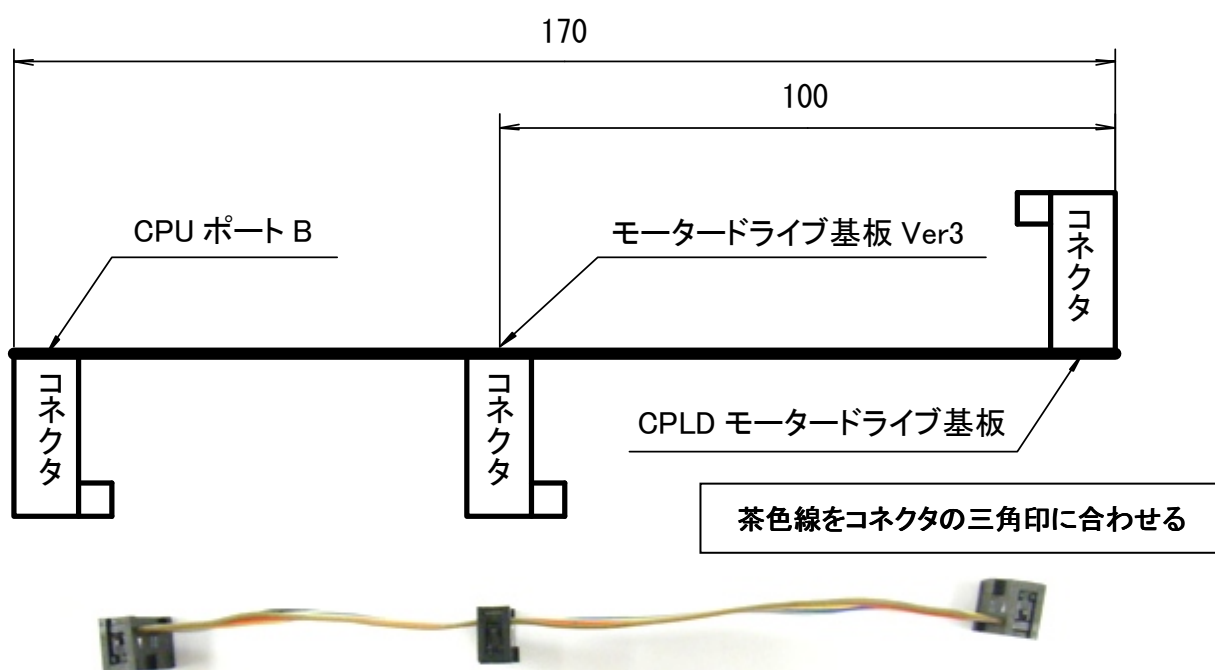


図6. 3 CPU・モータードライブ基板接続ケーブル組立図と実物

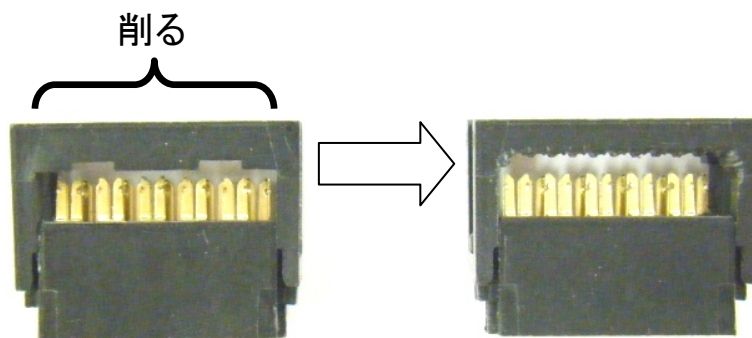


図6. 4 コネクタストッパ部

以上のケーブルを用意して TypeS センサ基板、ステアリングセンサ、ロータリーエンコーダ、CPU ボード、モータードライブ基板 Ver3、サーボモータードライバと各モータ、バッテリーを接続すると次の図6. 5のようになります。

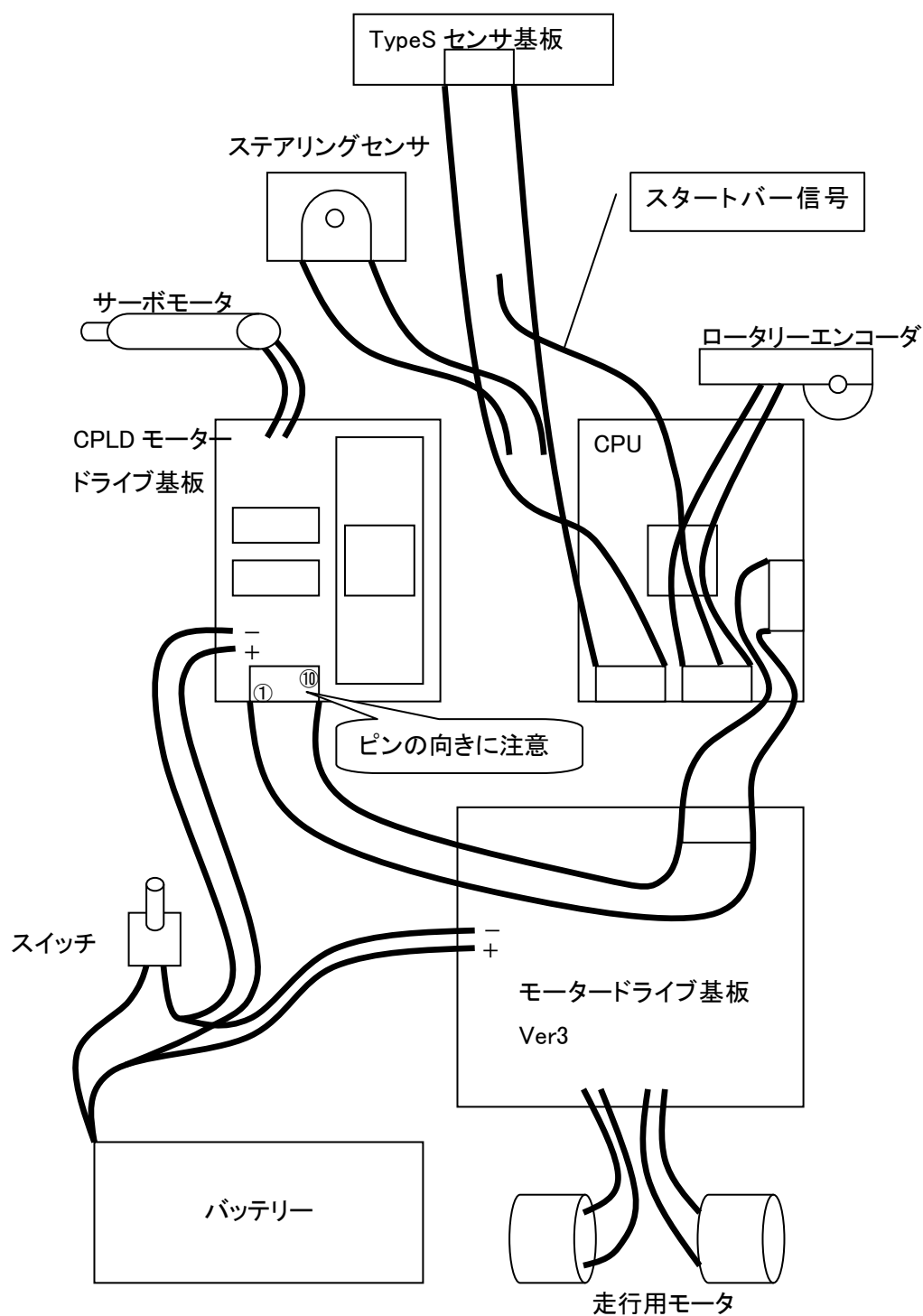


図6. 5 各ケーブル接続図

車体に取り付けると、下の図6. 6のようになります。本マニュアルの CPLD モータードライブ基板は CPU ボードと同サイズとなっているので、重ねて配置が可能です。大会で車検を受ける場合を考えると、CPU ボードが上にある方が望ましいです。全体図を図6. 7に示します。

今回は、この後の CPLD の回路生成や、車体の納まりを考えて CPU ボードを下にしています。ある程度マシンが動作するものになったら、レイアウトなどは工夫してみてください。

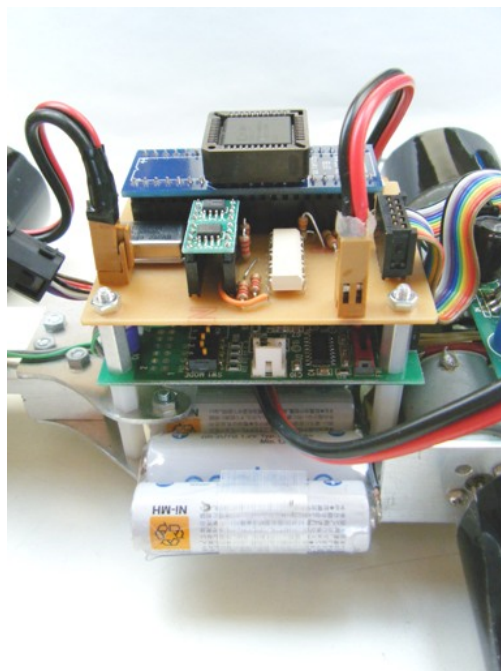


図6. 6 車体搭載図

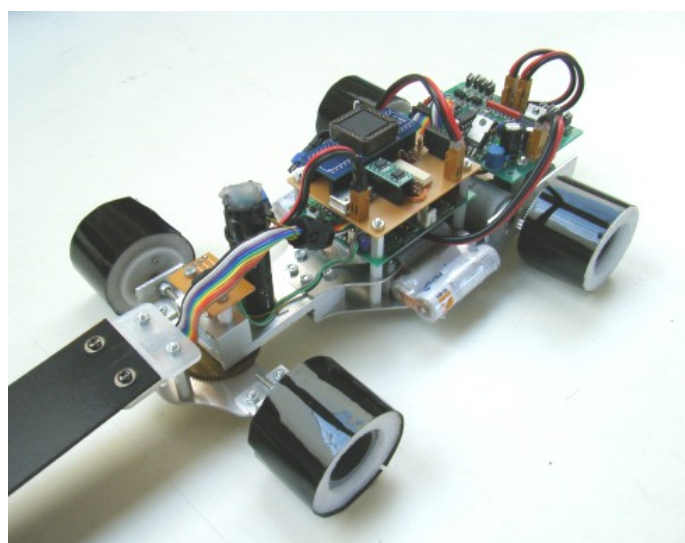


図6. 7 車体全体図

## 7. 開発環境のインストール

CPLDに回路を作るためには、VHDLと呼ばれる言語で回路を記述する必要があります。ここでは、メーカーから無料で入手可能な開発環境「ISE Project Navigator」の Ver11.1 を例にインストールの手順を紹介します。

この開発環境は、回路の生成の他に、回路のシミュレーションや回路の書き込みなどが行えるもので、CPLDを使うときには必須と言えます。

### 7.1 開発環境とライセンスファイルのダウンロード

1. Xilinx の公式ホームページ (japan.xilinx.com/)にて ISE WebPACK 11.1 をダウンロードします。



## 2. 「ALL Platforms(WebPACK)」を選択します。

The screenshot shows the Xilinx download portal interface. On the left, a sidebar lists version numbers from 14.1 down to 10.1, with 'アーカイブ' (Archive) at the bottom. The main content area displays a list of download options. The option 'All Platforms (WebPACK)' (TAR/GZ - 2.87GB) is highlighted with a red circle. Below it, a note states 'ISE WebPACK 11.1' and 'すべてのプラットフォーム用ダウンロードファイル' (Download file for all platforms). To the right, a table lists the contents of the download, including ISE Design Suite (All Editions), ChipScope Pro, PlanAhead, System Generator, Platform Studio, and Lab Tools. The download type is 'Full Product Installation' and the last updated date is '05/03/2010'. Other options include 'All Platforms (Full DVD)' and 'All Platforms (Foundation)'. At the bottom, there are links for 'ISE Design Suite 資料 - 11.1' and 'ISE Japanese Documentation Update (Windows)'.

## 3. アカウントを取得し、サイトにサインインします。

The screenshot shows the Xilinx website homepage. The top navigation bar includes links for 'サインイン' (Sign In), '言語の選択' (Language Selection), '資料' (Documentation), 'ダウンロード' (Download), and 'お問い合わせ' (Contact Us). The main content area features a large heading 'サインイン' (Sign In) and a login form with fields for 'ユーザー ID' (User ID) and 'パスワード' (Password), along with a 'Sign In' button. To the right of the login form, there is a section titled 'サインイン アカウントをお持ちでない方' (Sign In Account not yet created) with a list of benefits and a 'Create Account' button highlighted with a red circle. The footer contains information about Xilinx, Inc., a list of links for '会社概要' (Company Overview), 'サポート' (Support), and 'サインインとつながろう' (Sign In and Connect), and social media links for Facebook, Twitter, LinkedIn, and YouTube.

4. 表示の項目を確認して「次へ」をクリックします。

**XILINX**  
ALL PROGRAMMABLE

マイアカウント | サインアウト | 言語の選択 | 資料 | ダウンロード | お問い合わせ  
キーワードやパーツ番号を入力  
アドバンスサーチ

インフォメーション | 製品 | アプリケーション | サポート | 購入 | サイリンクスについて

### Download Center

米国輸出規制により、お客様のリクエストにお答えする前に、氏名、会社名、送付先住所などを確認させていただいております。米国輸出コンプライアンスシステムは英語のみが有効のため以下のすべての情報をご提供ください。

First Name  
Last Name  
Company/Organization  
Address 1  
City  
State/Province(都道府県)  
Postal/Zip Code  
Country  
Phone  
\*空の箇所は必須入力です。

名(半角英数)\*  
姓(半角英数)\*  
会社メールアドレス  
[プロフィールを更新](#)  
部署名(日本語で記載)\*  
住所(半角英数)\*  
都市名(半角英数)\*  
州(半角英数)\*  
郵便番号(半角数字)\*  
国名\* Japan  
電話番号(市外局番も含む)\*  
ファックス番号(市外局番も含む)  
会社名(半角英数)\*  
会社名(日本語で記載)\*

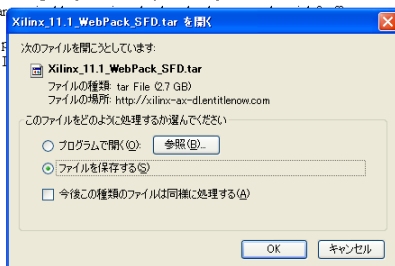
次へ

5. ファイルを保存します。ファイルは大きいので注意してください。

#### About the Download Manager

The Download Manager helps ensure that your Xilinx product download is successful. If your internet connection is temporarily lost or your computer is restarted, the Download Manager will resume the download.

You are being prompted to accept the Akamai Technologies Inc. security certificate, the I



Download Manager. By accepting the Akamai Technologies Inc. security

## 6. ライセンスファイルを取得します。

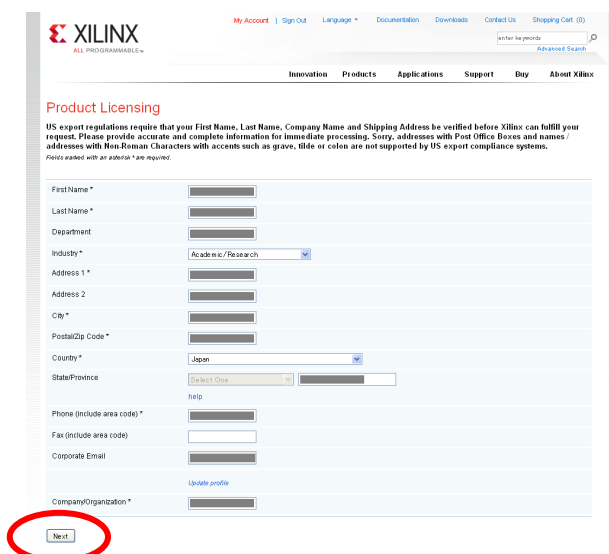
The screenshot shows the Xilinx website interface. At the top, there is a navigation bar with links for 'サインイン' (Sign In), '言語' (Language), '資料' (Documentation), 'ダウンロード' (Downloads), and 'お問い合わせ' (Contact Us). Below this is a search bar with the text 'キーワードの入力' (Enter keywords) and a 'アドバンス検索' (Advanced Search) button. The main navigation menu includes 'イノベーション' (Innovation), '製品' (Products), 'アプリケーション' (Applications), 'サポート' (Support), '購入情報' (Purchase Information), and 'ザイリンクスについて' (About Xilinx). The 'サポート' (Support) menu is highlighted with a red circle, and a red arrow points to the 'ダウンロードとライセンス' (Download and License) link. Below this, the 'ライセンスファイルの取得' (License File Download) link is also highlighted with a red circle. The page content shows the 'ISE Design Suite - 14.1 Full Product Installation' section, which includes a table of download links for various platforms (Windows, Linux) and a list of download includes (ISE WebPACK, ISE Design Suite, ChipScope Pro, etc.).

## 7. サインインします。

The screenshot shows the Xilinx website interface for signing in. At the top, there is a navigation bar with links for 'Sign in', 'Language', 'Documentation', 'Downloads', 'Contact Us', and 'Shopping Cart (0)'. Below this is a search bar with the text 'enter keywords' and an 'Advanced Search' button. The main navigation menu includes 'Innovation', 'Products', 'Applications', 'Support', 'Buy', and 'About Xilinx'. The page content shows the 'Sign in to the Xilinx Licensing Site' section, which includes a login form with fields for 'User ID' and 'Password', a 'Forgot your password?' link, and a 'Sign in' button. To the right of the login form, there is a section titled 'Don't have a Xilinx account yet?' with a list of benefits (Choose to receive important news and product information, Gain access to special content, Personalize your web experience on Xilinx.com) and a 'Create Account' link. At the bottom of the page, there is a footer section with links for 'Xilinx, Inc.', 'Company' (About Us, Investor Relations, Xilinx in the News, Xcell Journal, Jobs at Xilinx), 'Support' (Documentation, Downloads, Licensing, Training, Solution Centers), and 'Connect' (Contact Us, Connect on Facebook, Follow us on Twitter, Connect on LinkedIn, Watch us on YouTube, Join our Community, Subscribe to Newsletter).

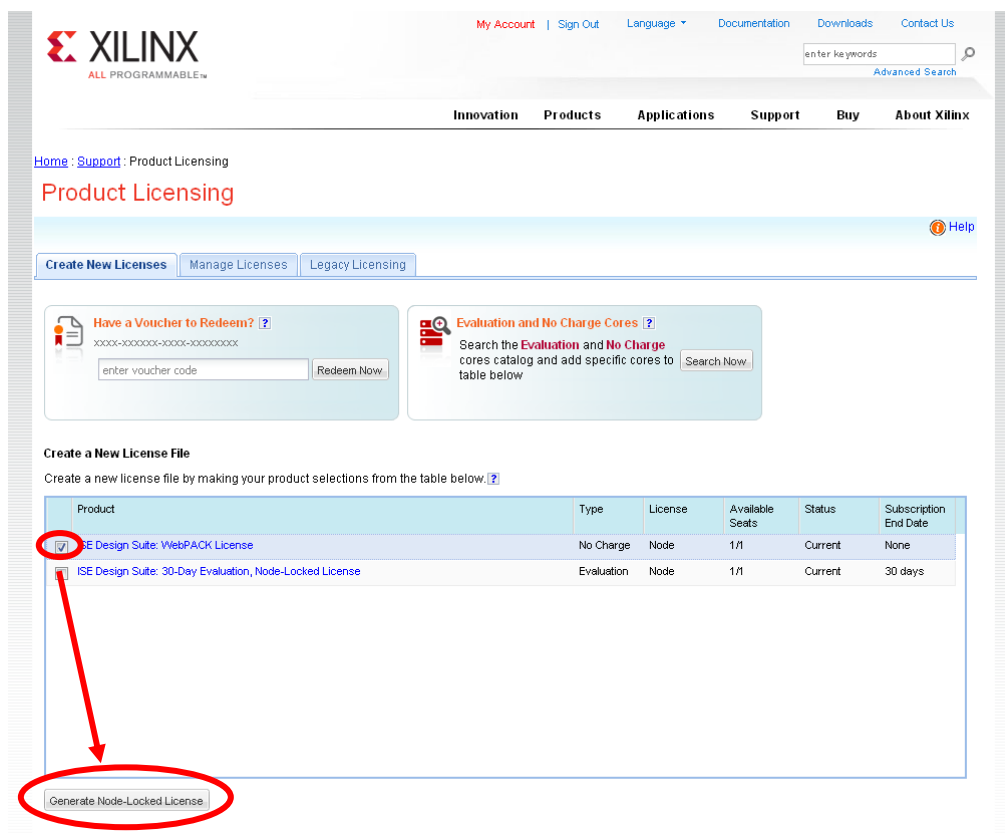


8. 表示の項目を確認して「Next」をクリックします。



The image shows the Xilinx Product Licensing registration form. It includes fields for First Name, Last Name, Department, Industry (set to Academic/Research), Address 1, Address 2, City, Postal/Zip Code, Country (set to Japan), State/Province, Phone, Fax, Corporate Email, and Company/Organization. A red circle highlights the 'Next' button at the bottom left of the form.

9. 「ISE Design Suite:WebPACK License」にチェックを付け、「Generate Node-Locked License」をクリックします。



The image shows the 'Create a New License File' screen. It features a table with product selections. The first row, 'ISE Design Suite: WebPACK License', is selected with a checkmark. A red arrow points from this checkmark to the 'Generate Node-Locked License' button at the bottom, which is also circled in red.

Product	Type	License	Available Seats	Status	Subscription End Date
<input checked="" type="checkbox"/> ISE Design Suite: WebPACK License	No Charge	Node	1/1	Current	None
<input type="checkbox"/> ISE Design Suite: 30-Day Evaluation, Node-Locked License	Evaluation	Node	1/1	Current	30 days

10. 「Next」をクリックします。

The screenshot shows the Xilinx Product Licensing interface. The main form is titled 'Generate Node License' and is at Step 1: PRODUCT SELECTION. It includes a table for product selections and a 'Next' button circled in red.

Product Selections *	Product	Type	Available Seats	Subscription End Date	Requested Seats
<input checked="" type="checkbox"/>	ISE Design Suite: WebPACK License	No Charge	1/1	None	1

At the bottom of the form, the 'Next' button is circled in red.

11. 「Next」をクリックします。

The screenshot shows the Xilinx Product Licensing interface at Step 4: REVIEW LICENSE REQUEST. It includes a table for product selections and a 'Next' button circled in red.

Product Selections	Product	Subscription End Date	Available Seats	Requested Seats
<input checked="" type="checkbox"/>	ISE Design Suite: WebPACK License		1/1	1

At the bottom of the form, the 'Next' button is circled in red.

12. 登録したメールアドレスにライセンスファイルが届きます。

Host Name	Host Type	Host ID	License Type	OS	Created By	Created Date
*	*	*	Node	*	KGCLUB SHOKO	29 JUN 2012

Comments

License File Details

Node License  
Host ID: ANY

Products

ISE Design Suite: WebPACK License (No Charge): 1 seats

Subscription ID Date	Activated Seats
ie	1

Modify License

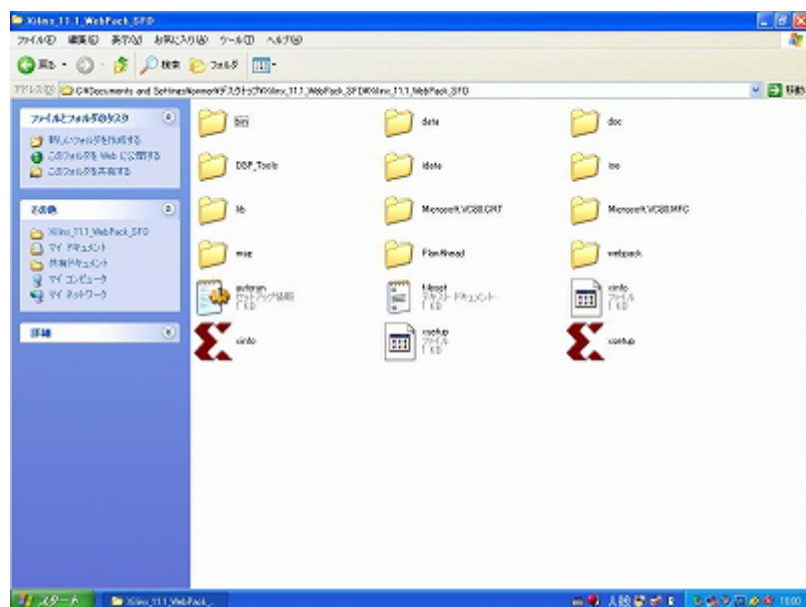
https://xilinx-ax-user.entitlenow.com/Across...sp?licenseType=8&product=8&tab=CreateLicense&#...

Success

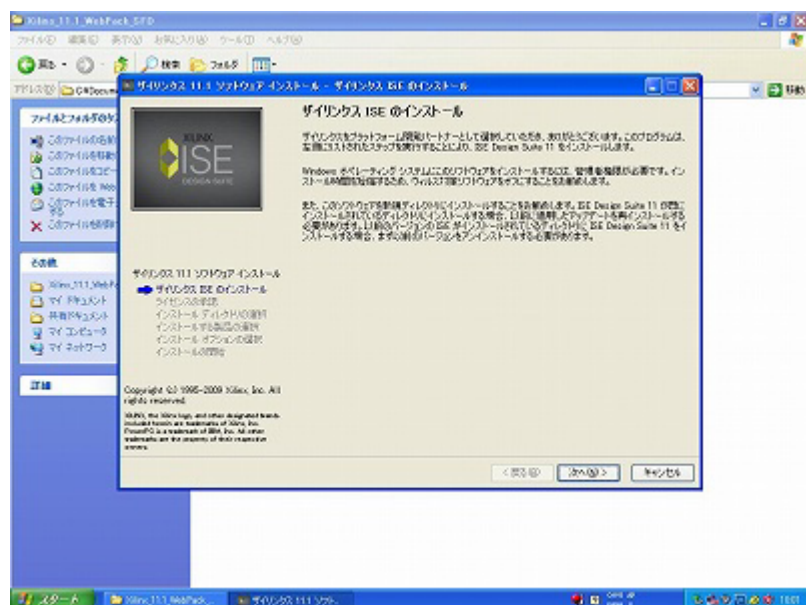
Cancel

## 7.2 開発環境のインストール

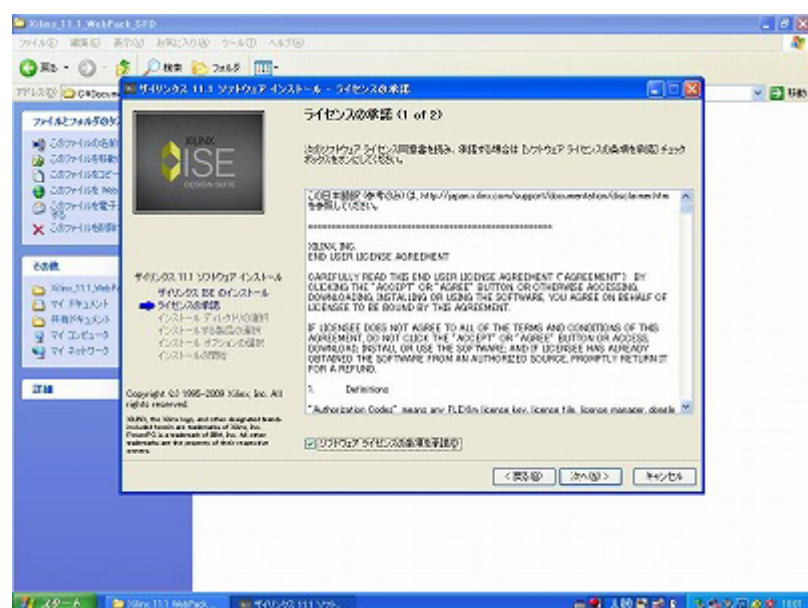
1. ダウンロードした.tar ファイルを解凍ソフトで展開し、「xsetup」を開きます。



2. 「次へ」をクリックします。



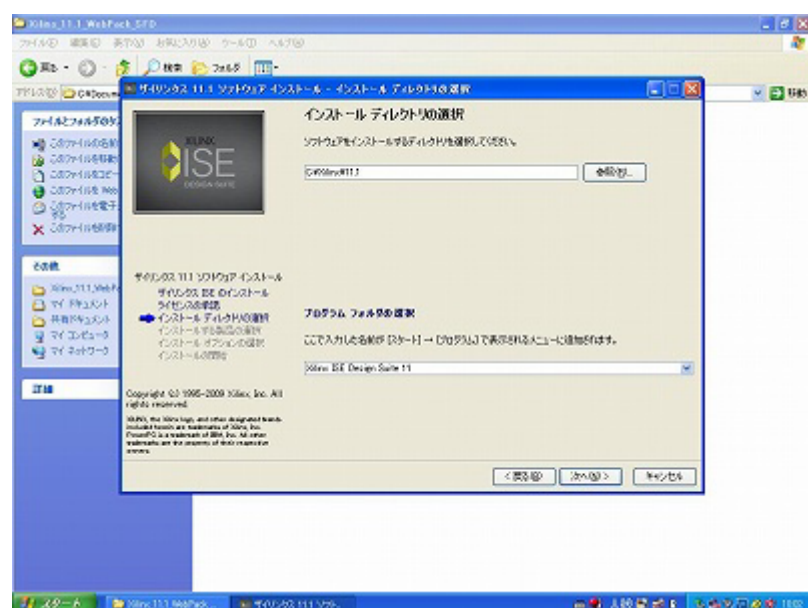
3. 「ソフトウェアライセンスの条項を承諾」にチェックを付け、「次へ」をクリックします。



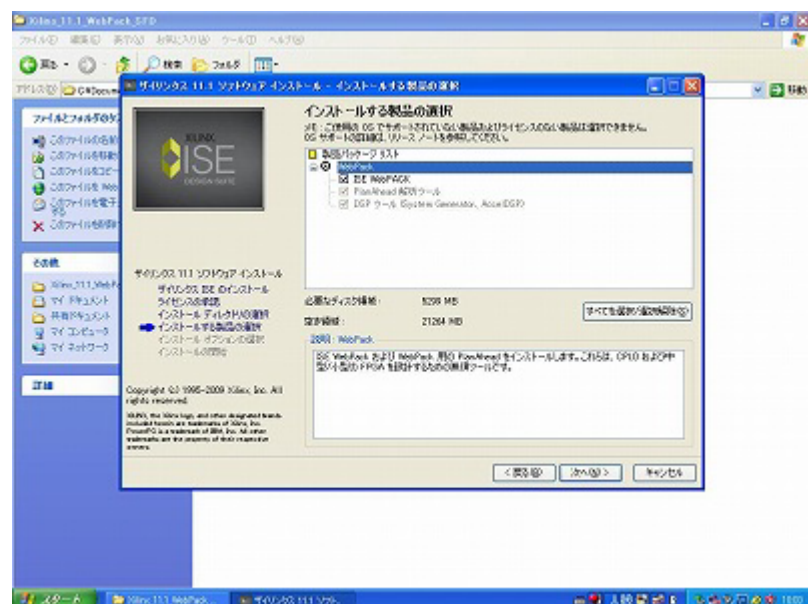
4. 「次へ」をクリックします。



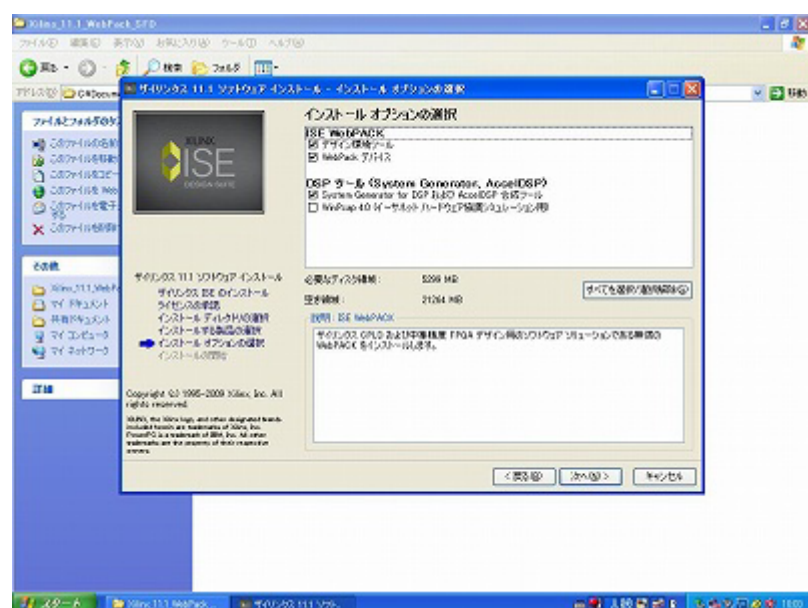
5. 「次へ」をクリックします。



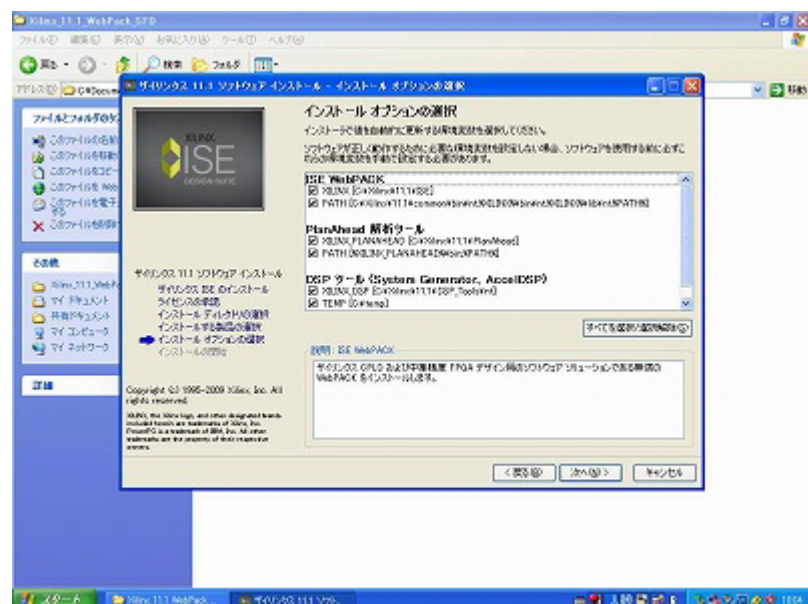
6. 「WebPack」の「ISE WebPACK」にチェックを確認して、「次へ」をクリックします。



7. 「WinPcap 4.0」以外にチェックを付けて、「次へ」をクリックします。

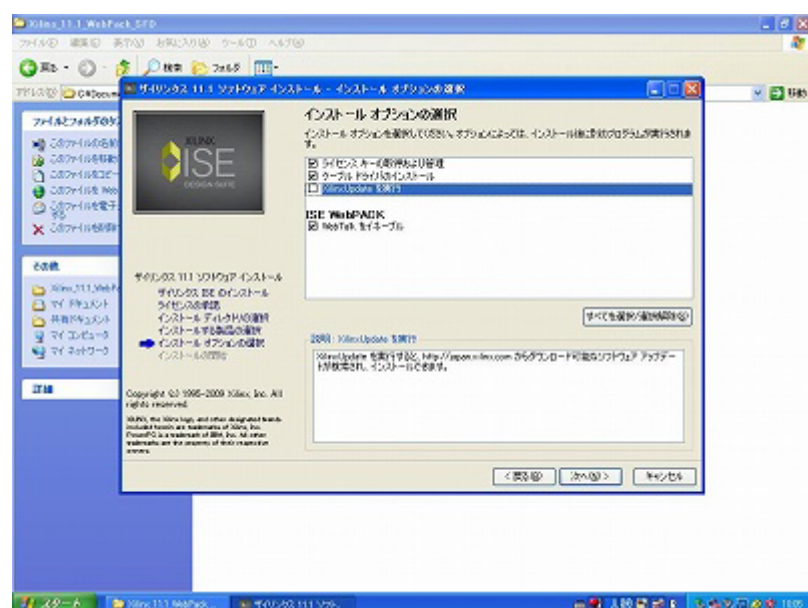


8. 全てにチェックが付いているのを確認して「次へ」をクリックします。

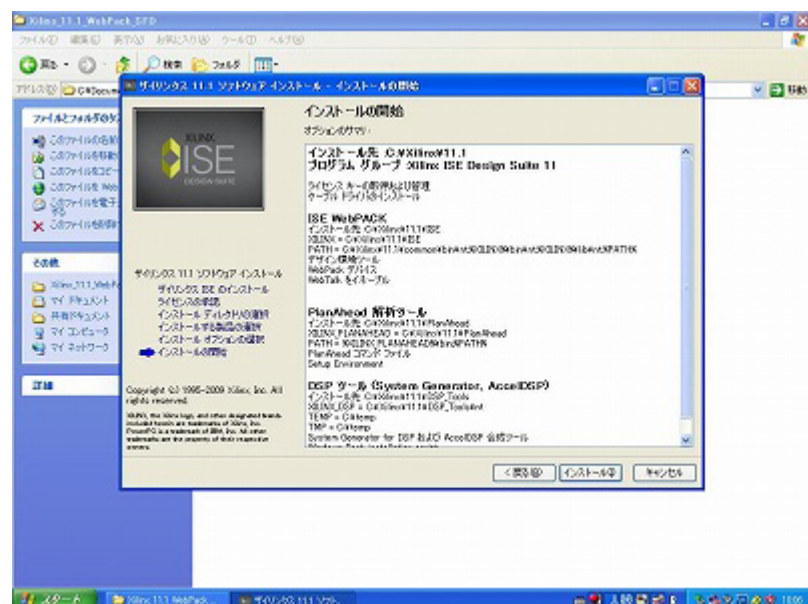




9. 「XilinxUpdate」のチェックをはずして、「次へ」をクリックします。

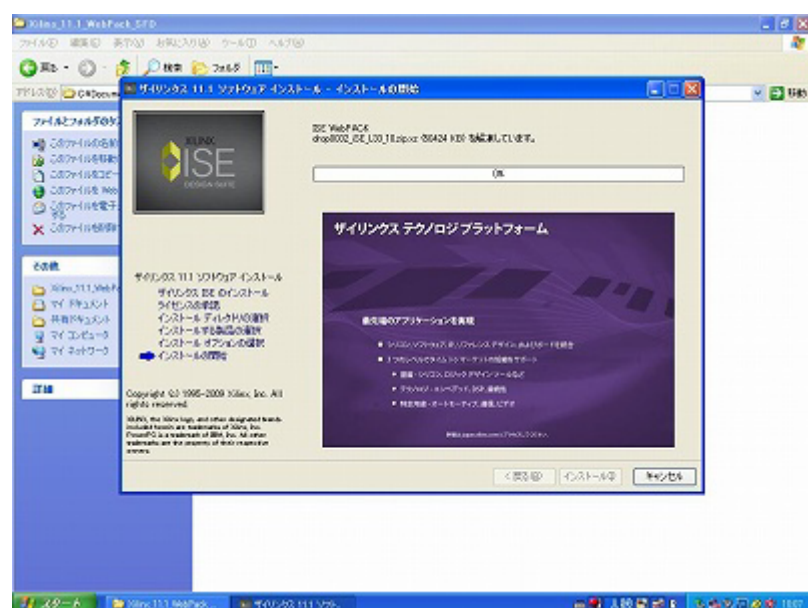


10. 「インストール」をクリックします。

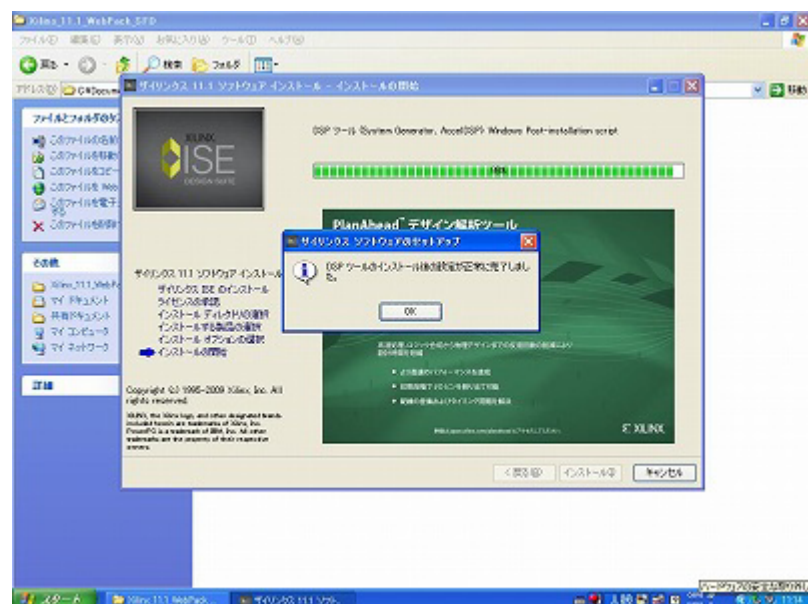




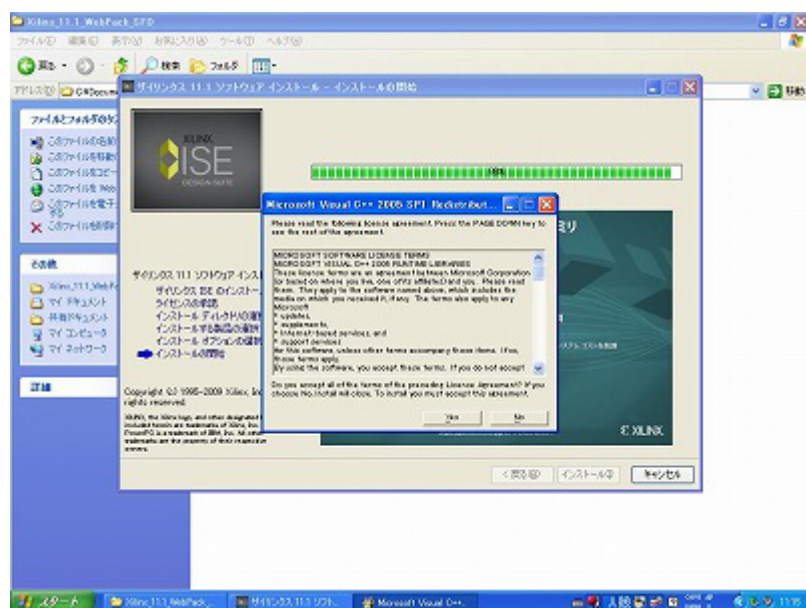
## 11. インストールが開始されます。



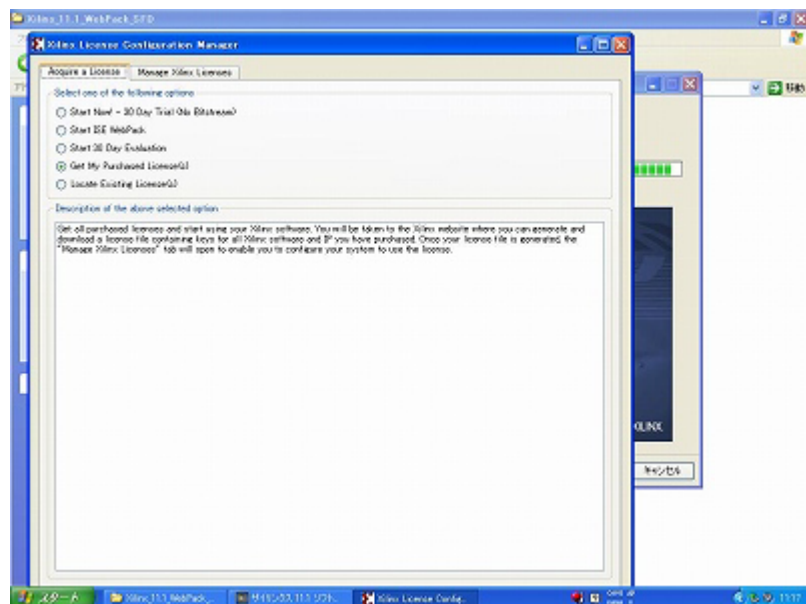
## 12. 「OK」をクリックします。



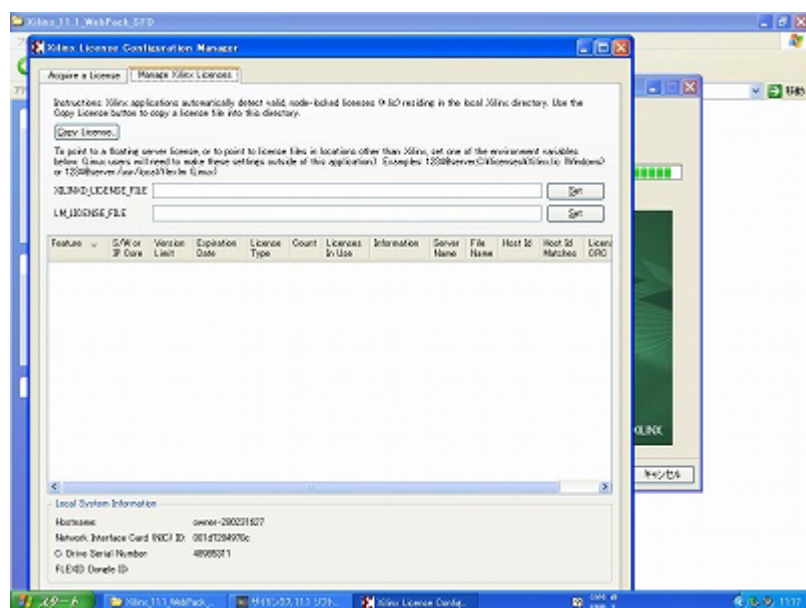
13. 「Yes」をクリックします。



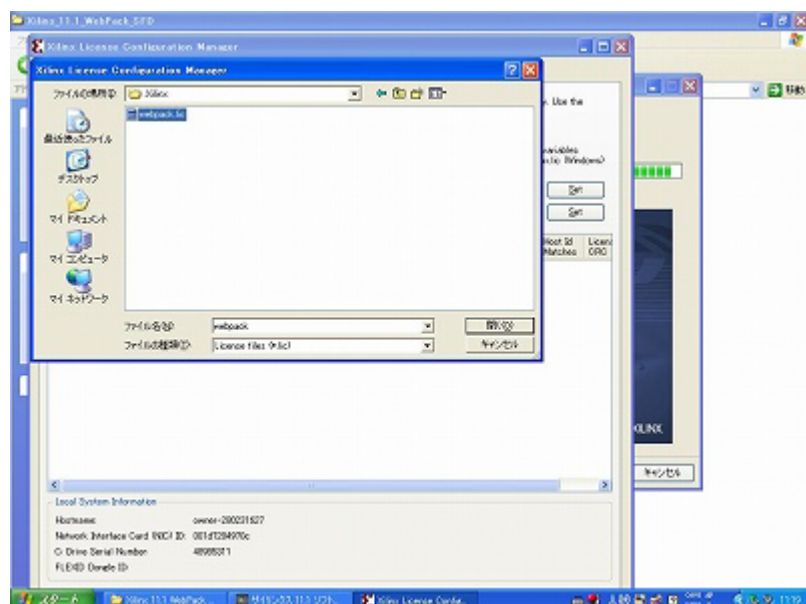
14. 「Get My Purchased License(s)」にチェックします。



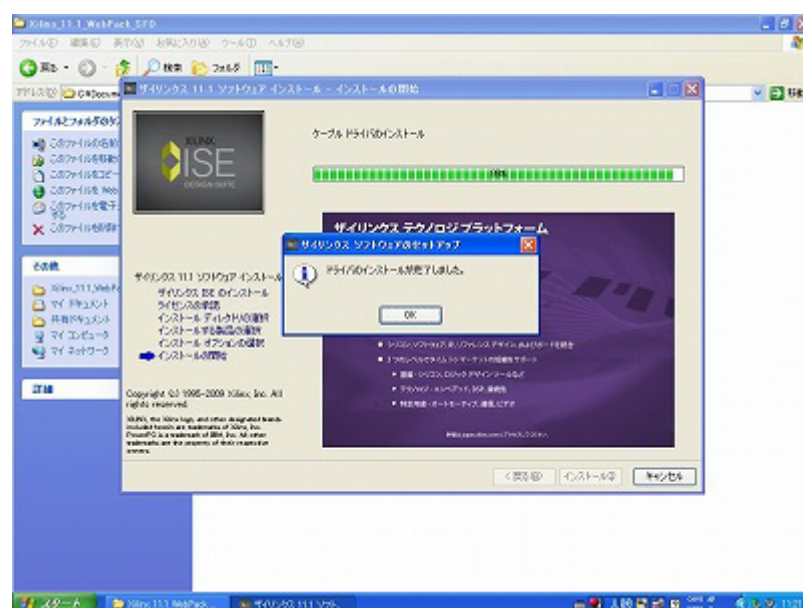
15. 「Manage Xilinx Licenses」タブを選択し、「Copy License…」をクリックします。



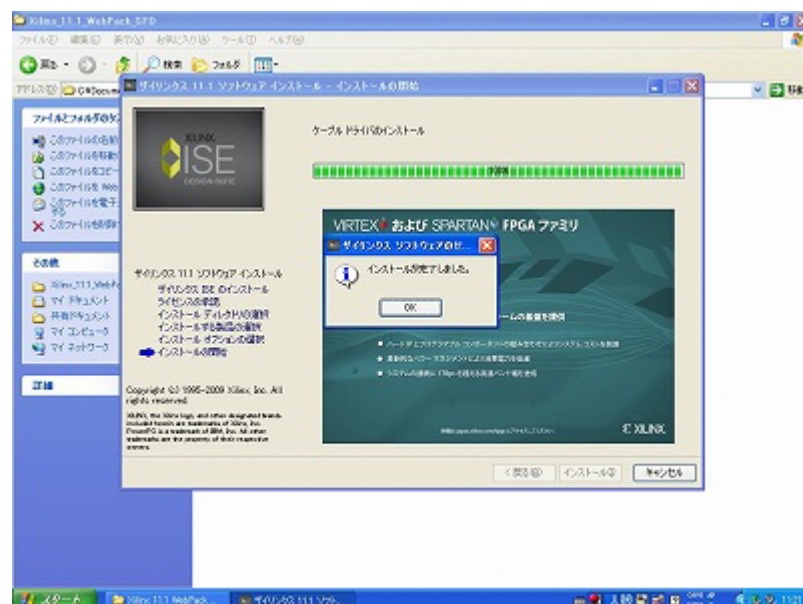
16. メールで届いた.lic ファイルを選択します。



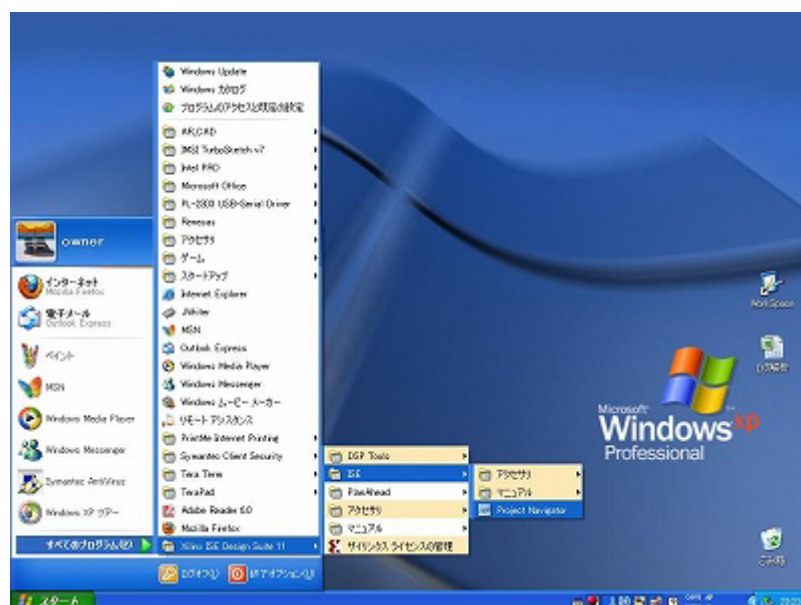
17. 「OK」をクリックします。



18. 「OK」をクリックします。



19. インストールが完了すると、「Xilinx ISE Design Suite 11」―「ISE」―「Project Navigator」で開くことができます。



## 8. サーボドライバサンプル回路

本ドライバの基本動作概要は、「4. 1CPLD を用いたモータードライバの動作」で説明した通りです。ここでは、具体的にどのような設計になっているのか、VHDL による解説を示します。詳しい VHDL の記述方法については専門書に譲りますが、回路を実際に設計する際の参考になればと思います。

下の図8. 1は開発環境の画面です。また、次頁の図8. 2に本ドライバの VHDL による回路を示します。

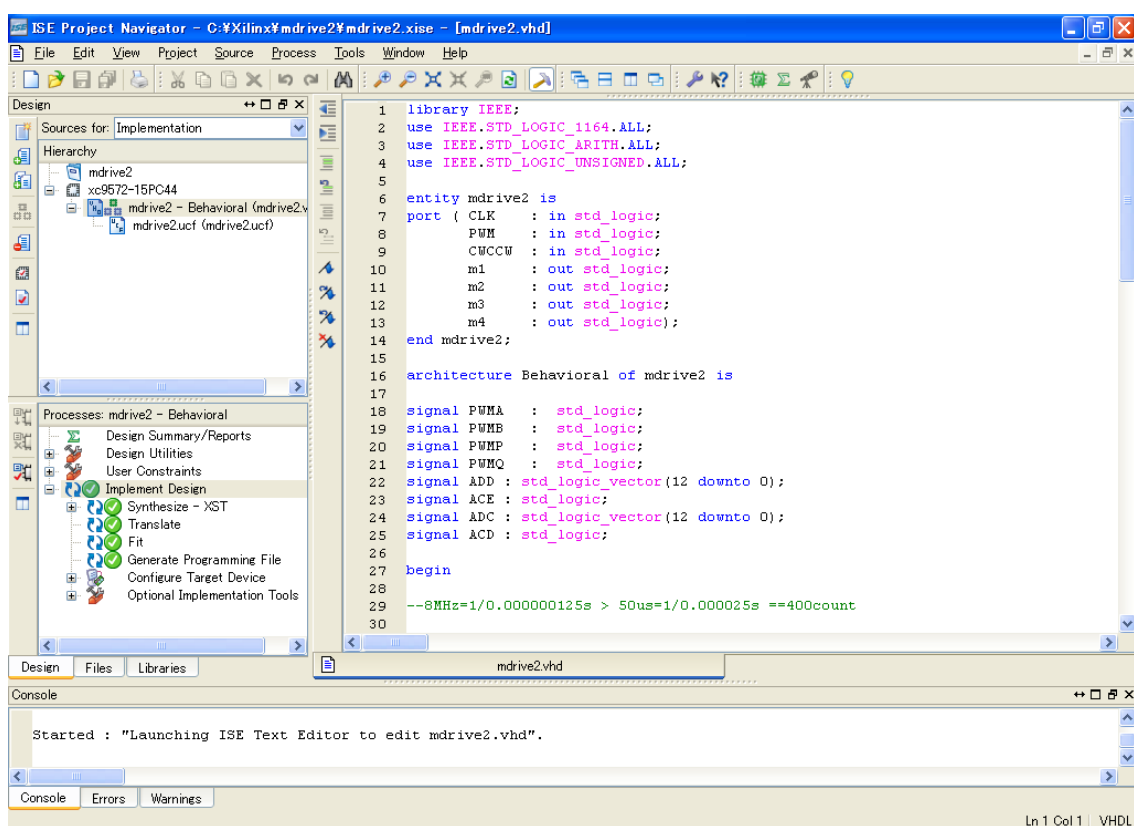


図8. 1 開発環境の画面

```

1  library IEEE;
2  use IEEE.STD_LOGIC_1164.ALL;
3  use IEEE.STD_LOGIC_ARITH.ALL;
4  use IEEE.STD_LOGIC_UNSIGNED.ALL;
5
6  entity mdrive2 is
7  port ( CLK      : in std_logic;
8        PWM       : in std_logic;
9        CWCCW     : in std_logic;
10       m1        : out std_logic;
11       m2        : out std_logic;
12       m3        : out std_logic;
13       m4        : out std_logic);
14 end mdrive2;
15
16 architecture Behavioral of mdrive2 is
17
18   signal PWMA : std_logic;
19   signal PWMB : std_logic;
20   signal PWMP : std_logic;
21   signal PWMQ : std_logic;
22   signal ADD : std_logic_vector(12 downto 0);
23   signal ACE : std_logic;
24   signal ADC : std_logic_vector(12 downto 0);
25   signal ACD : std_logic;
26
27 begin
28
29   --8MHz=1/0.000000125s > 50us=1/0.000025s ==400count
30
31   process (CLK,PWM) --立ち上がりが遅く、立下りが早い波形を生成
32   begin
33     if (PWM = '0') then
34       ADD <= "00000000000000";
35     elsif (CLK'event and CLK='1') then
36       ADD <= ADD+'1';
37     end if;
38   end process;
39
40   process (ADD,PWM)
41   begin
42     if (ADD > "0000110001111") then
43       ACE <= '1';
44     else
45       ACE <= '0';
46     end if;
47   end process;
48
49   PWMQ <= ACE and PWM;
50   PWMB <= PWMQ;
51
52   -----
53
54   process (PWM,CLK) --立ち上がりが早く、立下りが遅い波形を生成
55   begin
56     if (PWM = '1') then
57       ADC <= "00000000000000";
58     elsif (CLK'event and CLK='1') then
59       ADC <= ADC+'1';
60     end if;
61   end process;
62
63   process (ADC,PWM)
64   begin
65     if (ADC > "0000110001111" ) then
66       ACD <= '0';
67     else
68       ACD <= '1';
69     end if;
70   end process;
71
72   PWMP <= ACD or PWM;
73   PWMA <= PWMP;
74
75   -----Hブリッジに接続
76   process (CWCCW,PWMA,PWMB)
77   begin
78     if (CWCCW = '0') then
79       m1 <= PWMB;
80       m2 <= '0';
81       m3 <= '0';
82       m4 <= PWMA;
83     else
84       m1 <= '0';
85       m2 <= PWMA;
86       m3 <= PWMB;
87       m4 <= '0';
88     end if;
89   end process;
90
91 end Behavioral;

```

図8. 2 VHDLによるサーボドライバ



以下に上の図8. 2中の各部の動作や役割を解説します。

#### 6～14行

CPLD 本体に接続される信号の名前と入出力の区別を設定しています。信号の名前と説明を以下の通りです。

CLK : CPLD に入力される信号で、クリスタルオシレータを接続します。

PWM : マイコンから出力される PWM 信号を入力します。

CWCCW : マイコンから出力されるモーターの回転方向の信号を入力します。0V で正回転、5V で逆回転となっています。

m1～m4 : FET を実際に動かす信号を出力します。

#### 18～25行

CPLD 内部で扱う信号の名前を定義しています。0または1の信号なのか、数値なのか設定しています。ADD と ADC は13桁の2進数を扱うもので、モーターの回転方向が変わるときに 50us だけ FET を OFF にする時間を数えます。ADD と ADC 以外は01の信号で、処理上の信号が代入されます。

#### 31～38行

クロックを読み込んで処理する部分です。PWM が0の時は ADD を0にクリアし、それ以外の時はクロックが入力されるたびに ADD の値を1ずつ増やしていきます。

本ドライバで使用するクロックは 8MHz で 50us を計るためには信号を400個数えれば良いことになります。記述上は400を含んでこれより大きい場合としたいので、-1して399を数えています。これを13桁で2進数に変換すると0000110001111となります。

#### 30～47行

ADD が400以上=50us 以上、つまり、PWM が1になってから 50us 後に ACE を1にします。PWM が0の場合は ACE を0にします。

#### 49～50行

ACE と PWM を and 処理して PWMQ とし、PWMB に代入します。

以上の31～50行部分で ACE は、PWM が1になってから 50us 後に1になり、PWM が0になると同時に ACE も0となります。これにより、立ち上がりが遅く、立下りが早い信号がでかかります。

下の図8. 3に31～50行の信号の変化を示します。



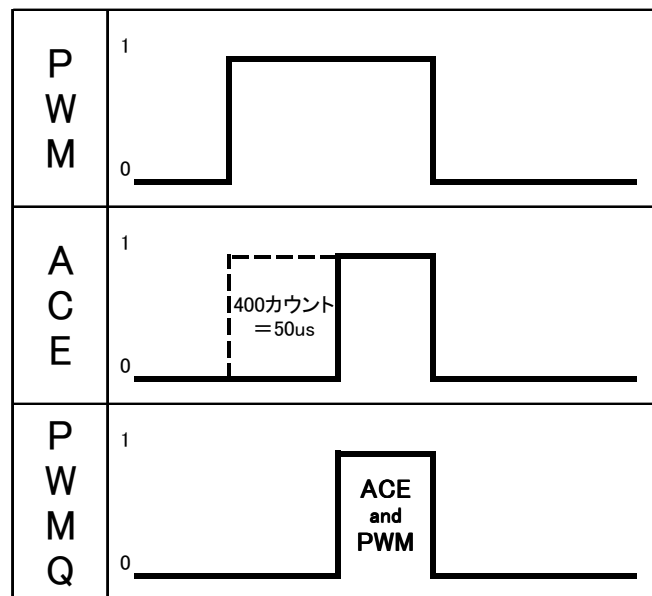


図8. 3 31～50行の信号の変化

#### 54～73行

PWMが0になるとACDを1にしてクロックのカウントを始め、50us 経つとACDを0にしています。これをPWM と or 演算し、立ち上がりが早く、立下りが遅い信号 PWMP を作っています。下の図8. 4に54～73行の信号の変化を示します。

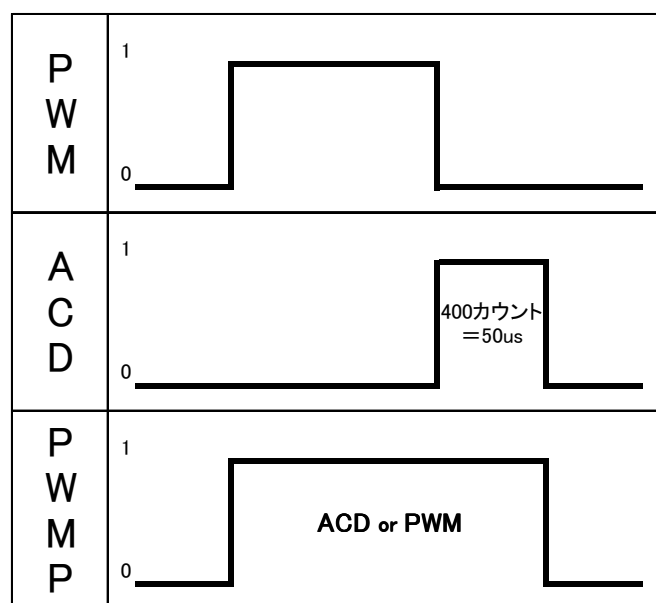


図8. 4 54～73行の信号の変化

## 76～89行

完成した信号を FET に接続します。m1は PchFET の右側m2は PchFET の左側、m3は NchFET の右側、m4は NchFET の左側にそれぞれ対応していて、m1とm4、m2とm3を組で使います。

信号 CWCCW が0V の時は正回転でm2とm3を OFF にしてm1とm4に PWM を入力し、CWCCW が5V の時は逆回転でm2とm3を PWM を入力してm1とm4を OFF にしています。

そして、PWMが変化するときには、50us のデッドタイムが入ってFET が燃えないようにすべてのFET を OFF にしています。

## 本ドライバのピン配置

CPLD では、生成した回路の信号を CPLD 本体のどのピンに接続するか選ぶことができます。下の表8. 1に本ドライバのピン配置を示します。

表8. 1 ピン配置

機能	信号	ピン		信号	機能
		1	44	5V	電源
		2	43		
		3	42		
		4	41		
クロック入力	CLK	5	40		
		6	39		
		7	38		
		8	37		
		9	36		
OV	GND	10	35		
		11	34		
		12	33		
Pch	m1	13	32	5V	電源
Pch	m2	14	31	GND	OV
回路ダウン ロード用	TDI	15	30	TDO	
	TMS	16	29		
	TCK	17	28		
Nch	m3	18	27		
Nch	m4	19	26		
PWM	PWM	20	25		
電源	5V	21	24		
回転方向	CWCCW	22	23	GND	OV

以上のように本ドライバは CPLD を動作させています。基本的な機能はキットのモータードライバ基板と変わりませんが、デッドタイムを作る部分がコンデンサの充放電作用を使ったものではなく、時間を正確に計って作っているところが違います。このデッドタイムはクリスタルオシレータからの信号ですので、狂いも少ないですし、カウントする時間を変えれば様々なFET に応用することが可能です。しかも、回路はプログラムを組むようにコンピュータ上で変更でき、部品点数も少なくシンプルです。その結果、故障しにくく安定した動作を実現できることにもなり、マイコンカーの性能を大きく向上させることができるようになります。

## 9. CPLDへの回路ダウンロード

### 9.1 ダウンロードケーブル(書込器)

#### ①J-Writer

USBに対応したダウンロードケーブル。現在のパソコンであればほぼ間違いなく使えます。

定価9800円(送料込)で、ザイリンクス、アルテラどちらのCPLDも書き込めます。コマンドプロンプトを使っての操作ですが、簡単で動作も軽快です。図9. 1に J-Writer 本体を示します。

通販サイト: <http://www.tokudenkairo.co.jp/jwriter.html>



図9. 1 J-Writer 本体

#### ②PLD-PGM キット

ザイリンクス、アルテラ両社で公表している書込回路をキットにした、ストロベリーリナックス製のダウンロードケーブル組立キット。1台2000円。パラレルポートが用意できるなら、こちらの方が安価です。図9. 2に PLD-PGM キットの完成写真を示します。

通販サイト: <http://strawberry-linux.com/catalog/items?code=20001>

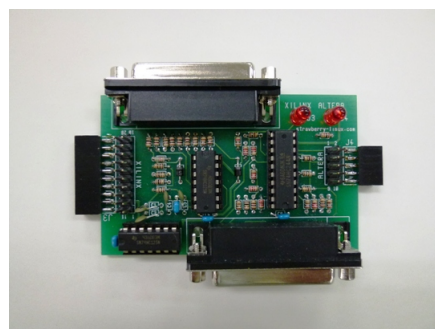


図9. 2 PLD-PGM キット

CPLD に回路を生成するためには、以上のような書込器が必要です。パラレルポートがあれば②の PLD-PGM キットが手頃ですが、最近のパソコンでは難しいのが現状です。

そこでUSBに対応して、1台約1万円と比較的安価な特殊電子回路社製の書込器「J-Writer」を使った方法を次項で紹介します。ちなみにメーカー純正のものは1台3万円弱と高価です。

## 9.2 J-Writer 専用ソフトのインストール

このソフトは本体のシリアルナンバーが必要です。J-Writer 本体を準備の上インストールしてください。

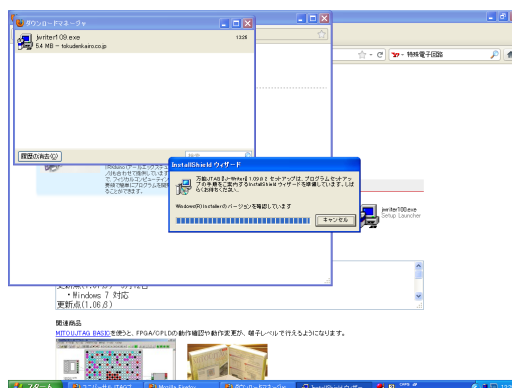
1. <http://www.tokudenkairo.co.jp/jwriter.html> から jwriter109.exe をダウンロードし、実行します。



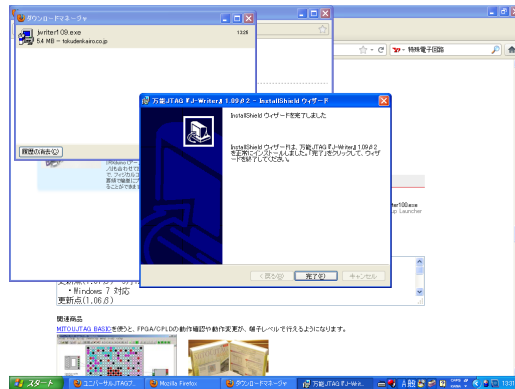
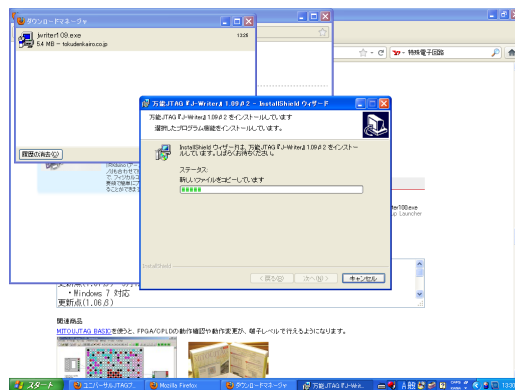
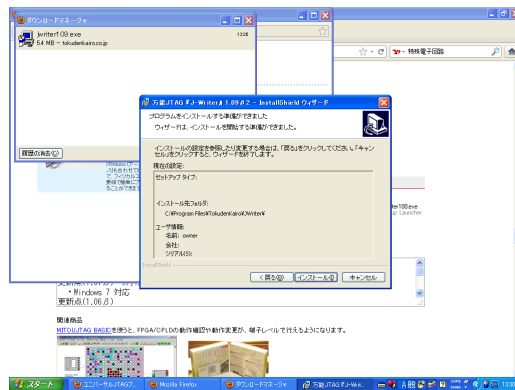
2. J-Writer 本体裏に貼られているシリアルナンバーを入力します。



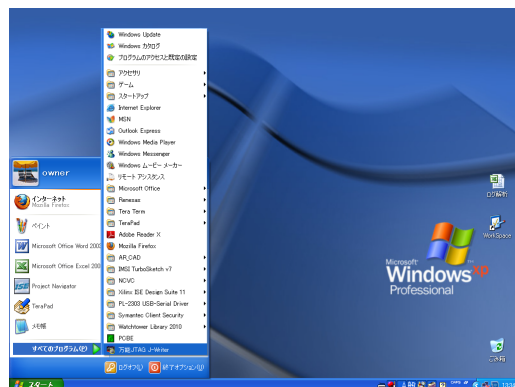
3. 表示される手順に沿ってインストールします。





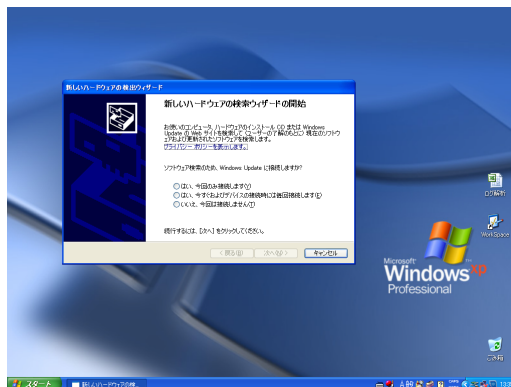


5. 特に重いソフトではないので、問題なくインストールされると思います。

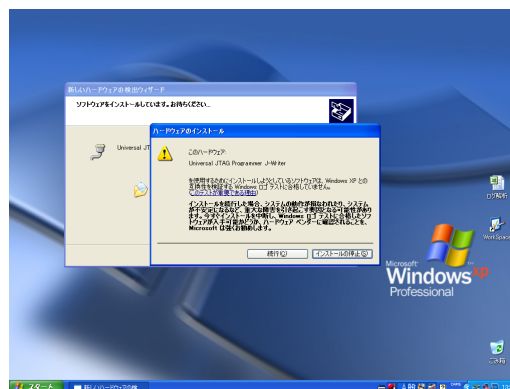
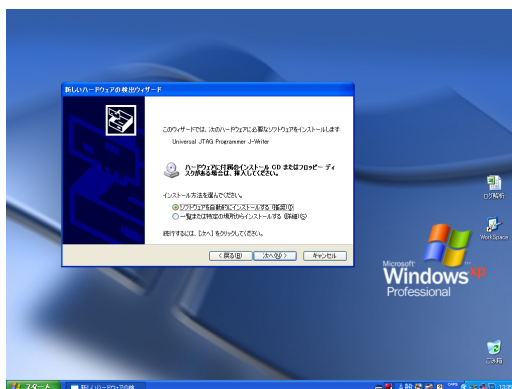


## 9.3 デバイスドライバのインストール

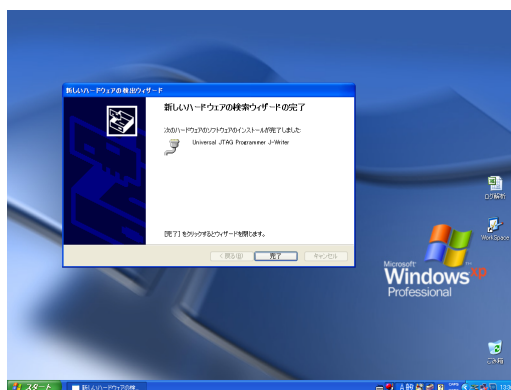
1. J-Writer 本体とパソコンを接続するとデバイスドライバがインストールされます。「新しいハードウェアの検索ウィザードの開始」では、「いいえ、今回は接続しません」を選択します。



2. 次に「ソフトウェアを自動的にインストールする」を選択して「続行」をクリックします。



3. 以上でインストールが完了します。



## 9.4 CPLD へのダウンロード(回路生成)手順

①まずはパソコン側で書き込む元になるデータを準備します。

本マニュアル付属の mdrive2 フォルダ内にある「mdrive2.jed」というファイルをコピーし、J-Writer をインストールしたフォルダ内に貼り付けておきます。通常のインストールであれば、C:\Program Files-TokudenKiro¥JWriter です。

②J-Writer 付属の9ピン1列コネクタを本体の表示色とコードの色を合わせて接続します。

次に本ドライバのダウンロードピンに、向かって左側から「緑・茶・白・紫・青・黄・灰・黒・赤」の順にコードを接続します。J-Writer と本ドライバの接続を下の図9. 3に示します。



図9. 3 J-Writer と本ドライバの接続

③フラットケーブルを車体側のモータードライブ基板 Ver3 から本ドライバに接続し、電源スイッチを入れて5Vを供給します。このとき、CPU、本ドライバへのバッテリーとモーターを接続しないでください。また、FETは取り外しておいてください。下の図9. 4にダウンロード時の各ケーブルの接続を示します。図9. 5にはパソコン～本ドライバの接続図を示します。

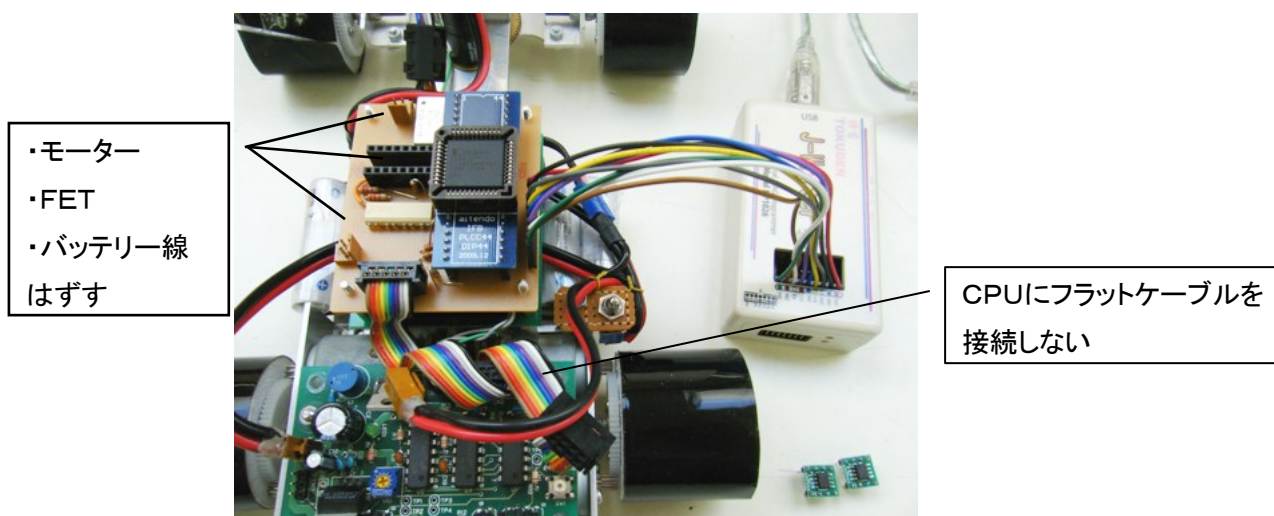


図9. 4 ダウンロード時の各ケーブル接続図



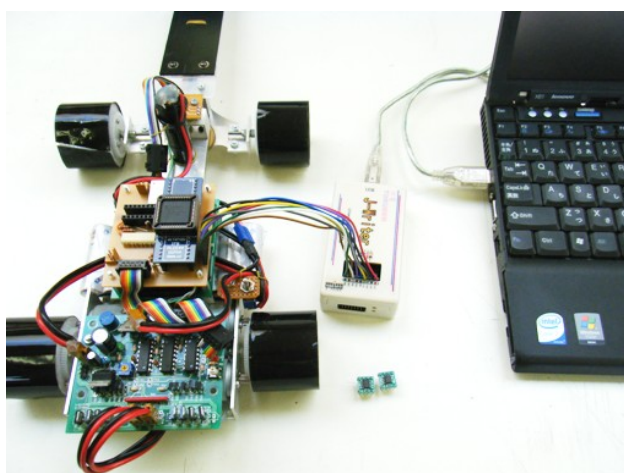


図9. 5 パソコン・J-Writer・ドライバ接続図

- ④すべてのプログラムから「万能 JTAG J-Writer」を機動します。起動すると図9. 6の画面が表示されるので、「jwriter -auto mdrive2.jed」と打ち込みます。

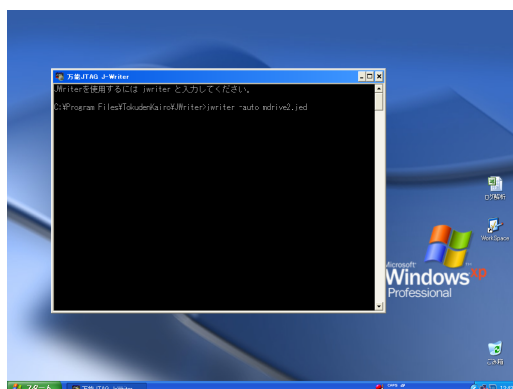


図9. 6 J-Writer 操作画面

- ⑤write、verify と続き、図9. 7のような画面が表示されると、書込完了です。

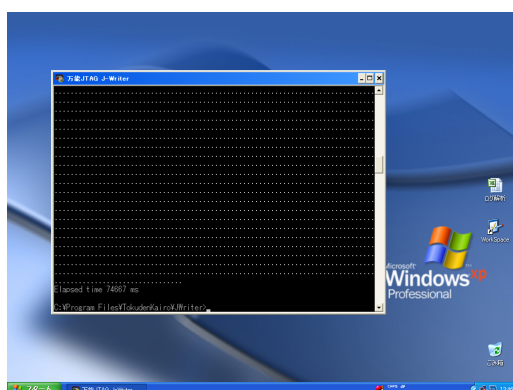


図9. 7 書込完了

## 10. サンプルプログラム

本章では、ここまで製作した CPLD モータードライブ基板を使ったマイコンカーを走行させるサンプルプログラムを紹介します。

このプログラムは、すでに公開されている「anaservo2008」というプログラムに最小限の変更を加えたものとなっています。ですから、レーンチェンジやサーボの角度指定制御、EEP-ROM などを使った走行ログの保存、ロータリーエンコーダには対応していません。

しかし、これから加えていく変更と同じように必要な部分を他のプログラムを参考にしたり、真似しながら、少しずつ追加・変更してプログラムを成長させていくことができます。ぜひ、ひとつずつ機能を追加して、これまで思っていたよりも簡単にプログラムが作れることを知っていただけたらと思います。

### 10. 1 ポート入出力設定の変更

```
227  /* **** */
228  /* H8/3048F-ONE 内蔵周辺機能の初期化 */
229  /* **** */
230  void init( void )
231  {
232      /* ポートの入出力設定 */
233      P1DDR = 0xff;
234      P2DDR = 0xff;
235      P3DDR = 0xff;
236      P4DDR = 0xff;
237      P5DDR = 0xff;
238      P6DDR = 0xf0; /* CPU基板上のDIP SW */
239      P8DDR = 0xff;
240      P9DDR = 0xf7;
241      PADDR = 0xf7; /* bit3:スタートバー */
242      /* */
243      PBDR = 0x80; /* */
244      PBDDR = 0xfe; /* モータドライブ基板Ver3 */
245      /* A/D */
246      AD_CSR = 0x12; /* スキャンモードAN2-0 */
247      AD_CSR |= 0x20; /* A/D変換開始 */
248      /* ITU0 1msごとの割り込み */
249      ITU0_TCR = 0x23; /* カウンタ、クリアの設定 */
250      ITU0_GRA = TIMER_CYCLE; /* 割り込み周期の設定 */
251      ITU0_IER = 0x01; /* 割り込み制御 */
252      /* ITU3,4 リセット同期PWMモード 左右モータ、サーボ用 */
253      ITU3_TCR = 0x23; /* カウンタ、クリアの設定 */
254      ITU3_FCR = 0x3e; /* リセット同期PWMモード */
255      ITU3_GRA = PWM_CYCLE; /* 周期の設定 */
256      ITU3_GRB = ITU3_BRB = 0; /* 左モータのPWM設定 */
257      ITU4_GRA = ITU4_BRB = 0; /* 右モータのPWM設定 */
258      ITU4_GRB = ITU4_BRB = 0; /* サーボのPWM設定 */
259      ITU_TOER = 0x38; /* 出力端子選択 */
260      ITU_STR = 0x09; /* ITUのカウントスタート */
261  }
```

図10. 1 ポート入出力設定部分

以降のプログラムの変更箇所は「anaservo2008」の行番号で説明します。上の図10. 1にポート入出力設定部分を示します。この中の241、243、247行目に変更を加えます。

## 241行

ポート A の入出力設定です。ポート A の3bit、ピン番では6ピンにはスタートバーが接続されています。

出力のビットは1、入力のビットは0とし、bit3 を入力、その他は出力に設定します。これを16進数に変換するとポート A の設定は 0xf7 となります。下の表10. 1にポート A の入出力設定値をまとめます。

表10. 1 ポート A の入出力設定値

bit	7	6	5	4	3	2	1	0
入出力	1	1	1	1	0	1	1	1
16 進数	f				7			

## 243行

ポート B の bit6 をサーボの回転方向に変更したので、電源投入直後の誤作動を防ぐために PBDDR の前に PBDR = 0x80 をセットしておきます。

## 247行

アナログ値を扱うポートをいくつ用意するかを設定しています。今回は、センサ基板のアナログセンサ2つとステアリングセンサの計3つを使うようにしていますので、設定値を 0x12 とします。これでアナログポートの0～2がアナログ入力となります。ピンはポート7の9～7ピンに対応しています。

## 10. 2 中心デジタルセンサとスタートバーの変更

下の図10. 2に中心デジタルセンサとスタートバー部のプログラムを示します。

```

296  /* **** */
297  /* アナログセンサ基板TypeSの中心デジタルセンサ読み込み */
298  /* 引数 なし */
299  /* 戻り値 中心デジタルセンサ 0:黒 1:白 */
300  /* **** */
301  unsigned char center_inp( void )
302  {
303      unsigned char sensor;
304
305      sensor = ~P7DR & 0x08; /* アナログセンサ基板TypeSの */
306      sensor = !!sensor; /* 中心デジタルセンサ読み込み */
307
308      return sensor;
309  }
310
311  /* **** */
312  /* アナログセンサ基板TypeSのスタートバー検出センサ読み込み */
313  /* 引数 なし */
314  /* 戻り値 0:スタートバーなし 1:スタートバーあり */
315  /* **** */
316  unsigned char startbar_get( void )
317  {
318      unsigned char sensor;
319
320      sensor = ~PADR & 0x08; /* アナログセンサ基板TypeSの */
321      /* スタートバーセンサ読み込み */
322
323      return sensor;
324  }

```

図10. 2 中心デジタルセンサとスタートバー部

### 305行

本マシンでは中心デジタルセンサをポート7の bit3 に接続していますので、この関数内のセンサ値読み込みは、`~P7DR & 0x08` とします。

### 320行

スタートバーはポート A の bit3 に接続していますので、中心デジタルセンサと同様にセンサ値読み込みは、`~PADR & 0x08` とします。意味は「ポート A の値が 08 となるビット、つまり bit3 だけを読み込み」となります。「~」は値を反転させるためのものです。

## 10. 3 LED 制御部の変更

```
353  /*~~~~~*/
354  /* LED制御 */
355  /* 引数 スイッチ値 LED0:bit0 LED1:bit1 "0":消灯 "1":点灯 */
356  /* 例)0x3→LED1:ON LED0:ON 0x2→LED1:ON LED0:OFF */
357  /*~~~~~*/
358  void led_out( unsigned char led )
359  {
360      unsigned char data;
361
362      led = ~led;
363      led <<= 7;
364      data = PBDR & 0x7f;
365      PBDR = data | led;
366  }
367
```

図10. 3 LED 制御部分

上の図10. 3にモータードライブ基板 ver3 の LED 制御部分のプログラムを示します。

本マシンではモータードライブ基板 Ver3 の LED1に接続されているポート B の bit6 をサーボの回転方向に割り当てているため、LED の表示をポート B の bit7 の LED0にだけするように変更が必要です。

363行では出力値を7ビット左にシフトさせます。そして、364行で `PBDR & 0x7f` としてポート B の bit7 だけに LED の出力値が入るようにします。

## 10. 4 サーボ回転方向設定の変更

下の図10. 4にサーボ回転方向設定の部分を示します。この部分ではサーボの回転方向を決める信号を処理しています。

本マシンでは、サーボの回転方向信号をポートBの bit6 に接続しています。この信号が0の時には正転、1の時には逆転となるように出力を変更します。

```

434  /*~~~~~ サーボハンドルの動作角度設定~~~~~*/
435  /* サーボハンドルの動作角度設定~~~~~*/
436  /* 引数 サーボ動作角度：-100～100~~~~~*/
437  /*~~~~~ サーボ動作角度設定~~~~~*/
438  void servoPwmOut( int pwm )
439  {
440      unsigned long   pwm_max;
441
442      pwm_max = PWM_CYCLE - 1;
443
444      if( pwm > 0 ) {
445          PBDR &= 0xbf;
446          ITU4_BRB = pwm_max * pwm / 100;
447      } else if( pwm < 0 ) {
448          PBDR |= 0x40;
449          pwm = -pwm;
450          ITU4_BRB = pwm_max * pwm / 100;
451      }
452  }
453

```

図10. 4 サーボ回転方向設定部分

#### 445行

ここではPWMが0より大きい場合の回転方向の設定をしています。ポートBの bit6 が0で、その他の bit は、それまでの値を保つように 0xbf を and 処理して「PBDR &= 0xbf;」とします。

#### 448行

ここはPWMが0より小さい場合の回転方向を設定しています。上の445行とは反対に、ポートBの bit6 が1でその他の bit が、それまでの値を保つようにするため、0x40 を or 処理して「PBDR |= 0x40;」とします。

### 10. 5 アナログセンサ、ステアリングセンサの変更

下の図10. 5にアナログセンサ設定部分を示します。この部分では、ステアリングセンサとセンサ基板のアナログセンサを読み込んで処理しています。

本マシンでは、ステアリングセンサをポート7の bit0、右アナログセンサをポート7の bit1、左アナログセンサをポート7の bit2 にそれぞれ接続しています。さらにこれらの値をマイコン側で扱うときには、bit0 の値は AD\_DRA、bit1 の値は AD\_DRB、bit2 の値は AD\_DRC となります。

#### 481行

ここではステアリングの角度を処理しています。スタート時のステアリング角度と現在の角度の差を計算して「getServoAngle」に出力しています。現在の角度は、ステアリングセンサのアナログ値を読み込めばよいので、ステアリングセンサが接続されたポート7の bit0、つまり AD\_DRA となります。

```

474  /*~~~~~*/
475  /* サーボ角度取得 */
476  /* 引数 なし */
477  /* 戻り値 中心を0 左側でプラス、右側でマイナスの数値 */
478  /*~~~~~*/
479  int getServoAngle( void )
480  {
481      return( (AD_DRA>>6) - iAngle0 ); /*----- 変更 -----*/
482  }
483
484  /*~~~~~*/
485  /* アナログセンサ値取得 */
486  /* 引数 なし */
487  /* 戻り値 センサ値 */
488  /*~~~~~*/
489  int getAnalogSensor( void )
490  {
491      int ret;
492
493      ret = (AD_DRC>>6) - (AD_DRB>>6); /* アナログセンサ情報取得 */ /*----- 変更 -----*/
494
495      if( !crank_mode ) {
496          /* クランクモードでなければ補正処理 */
497          switch( iSensorPattern ) {
498              case 0:
499                  if( sensor_inp() == 0x04 ) {
500                      ret = -650;
501                      break;
502                  }

```

図10. 5 アナログセンサ設定部分

#### 493行

ここはセンサ基板のアナログセンサ値を処理しています。アナログセンサのマシンは、左右のセンサ値の差を計算して、値が大きくなるほど右、値が小さくなるほど左にハンドルを切るように制御しています。

本マシンでは、左アナログセンサはAD\_DRC、右アナログセンサはAD\_DRB ですので、493行目は「 ret = (AD\_DRC>>6) - (AD\_DRB>>6); 」となります。

以上の変更で「とりあえず」動作するマシンのプログラムとなります。

このようにポートの設定ができるようになると、ハードウェア設計の幅を広げることができます。例えば、センサを増設したり、出力部分を変更したりなど、キットのものに限定されない自前のハードウェアが製作しやすくなります。また、マイコンカーだけに限らない他の分野への応用ができるようになります。

## 参考文献

1. マイコンカーラリー事務局、2008年度マイコンカーラリー講習会マニュアル
2. 森北出版、堀桂太郎著、図解VHDL実習―ゼロからわかるハードウェア記述言語―
3. 川内康雄、マイコン徹底入門、<http://miqn.net/>

## 付録

### 本マニュアル付属の各データについて

#### ① サーボドライバサンプル回路「mdrive2」

開発環境で使用するデータ一式を含みます。この中の「mdrive2.jed」だけを取り出して J-Writer で CPLD に回路を生成することもできます。

#### ② CPLD 講習サンプルプログラム

10章で作成した走行用のサンプルプログラムです。お使いのコンピュータの WorkSpace にコピーして使うことができますが、手順に示したように作ってみることをお勧めします。

## 謝辞

本マニュアル作成にあたり、たくさんのご協力をいただきました。苫小牧工業高校の田中先生をはじめ、北海道MCRプロジェクトに関係の諸先生方に感謝いたします。(2012年)