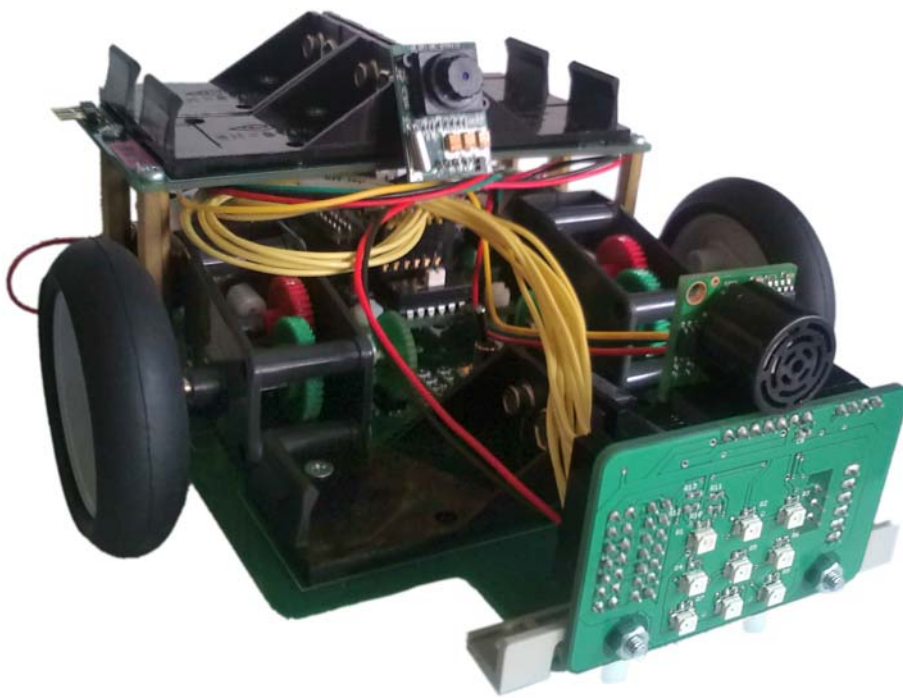


ライントレースカー KTM-02 取扱説明書



氏名 _____

RENESAS, SH-7125F, Renesas C/C++ Compiler Package for SuperH RISC engine family, High-performance Embedded Workshop は、ルネサス エレクトロニクス株式会社の商標および登録商標です。

ALPHA PROJECT, STK-7125, Flash Writer EX for SH7125F は、株式会社 アルファ プロジェクトの商標および登録商標です。

XC9500 シリーズ (XC95108, XC9536), ISE Web PACK は、ザイリンクス株式会社の登録商標です。

Windows 7 はマイクロソフト株式会社の登録商標です。
その他記載されている会社名、製品名は、各社の商標および登録商標です。

本書の一部または全部を許可なく複製または転載することを禁じます。

目次

第 1 章	はじめに	1
1.1	本書の構成	1
1.2	セット内容	2
1.2.1	ベースボード	2
1.2.2	MCU 搭載ボード	5
1.2.3	ドットマトリクスディスプレイボード	8
1.2.4	電源ボード	11
1.2.5	ケーブル類	12
1.2.6	その他	15
1.3	開発に必要なソフトウェア	15
第 2 章	接続方法	16
2.1	部品の接続	16
2.1.1	ケーブルコネクタによる接続	18
2.1.2	ピンソケットによる接続	27
第 3 章	デバイスの詳細	30
3.1	シリアル通信によりデータを送受信するデバイス	30
3.1.1	赤外線センサ	30
3.1.2	カメラ	30
3.1.3	距離センサ	31
3.2	LCD とドットマトリクスディスプレイ用 LED	31
3.3	タイヤ回転数検知用フォトインタラプタ	31
第 4 章	電源	32
4.1	AC アダプタと 006P 充電電池の切替	34
4.2	モータ用電源コネクタ	34
4.3	モータドライバ	34
第 5 章	Bluetooth の使用方法	35
5.1	Bluetooth-USB アダプタ	35
5.1.1	セットアップ	35
5.2	Bluetooth モジュール	44
5.2.1	設定可能時間	45
5.3	無線のシリアル通信	49
第 6 章	各デバイスの調整方法	50
6.1	LCD	50
6.2	赤外線センサ	50

第 7 章	開発手順	51
7.1	HDL による CPLD への回路開発	52
7.2	C/C++による MCU へのプログラム開発	54
7.2.1	新規ワークスペースの作成	54
7.2.2	プログラミングとビルド	60
7.2.3	ダウンロード	62
7.2.4	プログラムの開始	67
第 8 章	端子の機能	68
8.1	ポート A	68
8.2	ポート B	70
8.3	ポート E	71
8.4	ポート F	71
付 録 A	部品リスト	73
A.1	ベースボード	73
A.2	MCU 搭載ボード	74
A.3	ドットマトリクスディスプレイボード	75
A.4	電源ボード	75
付 録 B	回路図	76

目 次

1.1	ベースボード (表)	2
1.2	ベースボード (裏)	3
1.3	MCU 搭載ボード	5
1.4	MCU 搭載ボード (一部拡大)	7
1.5	Bluetooth モジュール	7
1.6	ドットマトリクスディスプレイボード (表)	8
1.7	ドットマトリクスディスプレイボード (裏)	8
1.8	カメラ	9
1.9	距離センサ	9
1.10	LCD	9
1.11	電源ボード	11
1.12	USB ケーブル	12
1.13	電源ボード用電源ケーブル	12
1.14	MCU 搭載ボード-ドットマトリクスディスプレイボード接続ケーブル	13
1.15	AC アダプタ	13
1.16	006P 充電電池	15
1.17	Bluetooth-USB アダプタ	15
2.1	部品の接続	17
2.2	ドットマトリクスディスプレイボードの撮影方向	18
2.3	ドットマトリクスディスプレイボードとカメラの接続	18
2.4	ドットマトリクスディスプレイボードと MCU 搭載ボードの接続	19
2.5	MCU 搭載ボードとドットマトリクスディスプレイボードの接続	19
2.6	ドットマトリクスディスプレイボードと MCU 搭載ボードの接続 (詳細)	20
2.7	MCU 搭載ボードとドットマトリクスディスプレイボードの接続 (詳細)	20
2.8	電源ボードの接続端子	21
2.9	電源ボードに電源ボード用電源ケーブルを取り付けた様子	21
2.10	ベースボードに電源ボード用電源ケーブルを接続した様子	22
2.11	MCU 搭載ボードに電源ボード用電源ケーブルを接続した様子	22
2.12	MCU 搭載ボードに左側モータを接続した様子	23
2.13	MCU 搭載ボードに右側モータを接続した様子	23
2.14	ベースボード用 JTAG 端子にダウンロードケーブルを接続した様子	24
2.15	ドットマトリクスディスプレイボード用 JTAG 端子にダウンロードケーブルを接続した様子	25
2.16	USB ケーブルによりパソコンと MCU 搭載ボードを接続した様子	26
2.17	ベースボードに AC アダプタを取り付けた様子	26
2.18	ドットマトリクスディスプレイボードに距離センサを接続した様子 (裏)	27
2.19	ドットマトリクスディスプレイボードに距離センサを接続した様子 (表)	27

2.20	ドットマトリクスディスプレイボードに LCD を接続した様子	28
2.21	MCU 搭載ボードに Bluetooth モジュールを接続した様子	28
2.22	MCU 搭載ボード裏側にある電解コンデンサ	29
3.1	赤外線センサ	31
4.1	電源に注目した接続状況	33
4.2	電源切替スイッチ	34
5.1	ユーザアカウント制御	36
5.2	インストールウィザード 1	37
5.3	使用許諾契約	37
5.4	インストール準備完了	37
5.5	Bluetooth Stack for Windows by Toshiba	38
5.6	ドライバのインストール開始	38
5.7	Bluetooth-USB アダプタの取り付け	38
5.8	ドライバのインストール中	39
5.9	ドライバのインストール終了	39
5.10	再起動	39
5.11	再起動後に現れるダイアログ	40
5.12	Bluetooth 設定のためのインジケータ	40
5.13	Bluetooth 設定ウィザード	41
5.14	Bluetooth デバイスのスキャン	41
5.15	Bluetooth デバイス	41
5.16	Bluetooth モジュールの対応プロファイルを検知中	42
5.17	初期設定完了	42
5.18	Bluetooth モジュールと接続	43
5.19	Bluetooth モジュールのフロー	44
5.20	COM ポートの確認	45
5.21	設定可能状態	46
7.1	開発時の流れ	51
7.2	HEW のアイコン	54
7.3	ワークスペースの選択	54
7.4	ワークスペース名の入力	55
7.5	MCU の選択	55
7.6	オプションの選択	56
7.7	イニシャルルーチンの設定	56
7.8	ライブラリの選択	57
7.9	スタックポインタの設定	57
7.10	ベクタテーブルの設定	58
7.11	デバッガの設定	58
7.12	確認	59
7.13	最終確認	59
7.14	HEW の用語	60

7.15	main 関数の編集	61
7.16	ビルド	61
7.17	MCU モード切替スイッチ	62
7.18	COM ポートの番号	63
7.19	FW のアイコン	64
7.20	COM ポートの設定	64
7.21	ダウンロードファイルを開く	65
7.22	ダウンロード開始 1	66
7.23	ダウンロード開始 2	66
7.24	ダウンロード終了	67

表 目 次

1.1	ベースボードの部品	4
1.2	MCU 搭載ボードの部品	6
1.3	ドットマトリクスディスプレイボードの部品	10
1.4	電源ボードの部品	11
1.5	ケーブル類	14
1.6	その他	15
3.1	シリアル通信によりデータを送受信するデバイス	30
5.1	基本設定表示	47
5.2	拡張設定表示	47
5.3	その他の設定の表示	48
7.1	ドットマトリクスディスプレイのパターンコード	52
7.2	ドットマトリクスディスプレイのパターンコードを送信する信号線	53
8.1	デバイスの略称	68
8.2	PACRL4	68
8.3	PACRL3	69
8.4	PACRL2	69
8.5	PACRL1	69
8.6	PBCRH1	70
8.7	PBCRL2	70
8.8	PBCRL1	70
8.9	PECRL4	71
8.10	PECRL3	71
8.11	PECRL2	71
8.12	PECRL1	72
A.1	ベースボード上の部品リスト	73
A.2	MCU 搭載ボード上の部品リスト	74
A.3	ドットマトリクスディスプレイボード上の部品リスト	75
A.4	電源ボード上の部品リスト	75
B.1	回路図一覧	76

第1章 はじめに

本書は、マイコン応用の教材として開発したライントレースカー (以下、KTM-02) の取扱説明書である。この KTM-02 には赤外線センサ、カメラ、キャラクタ LCD、フォトインタラプタ、距離センサ、無線シリアル通信 Bluetooth などのデバイスが備わっている。これらのデバイスを MCU および CPLD で制御する方法を学ぶことができる。KTM-02 は一般的なライントレースカーと比べて大きく異なる。一般的なライントレースカーは、線上を永続的にたどり続けるが、KTM-02 は線が交差している箇所 (以下、交差点) が多くある格子状の線上を移動することができる。理由は底面に装備した 25 個の赤外線センサにより交差点を検知することができるからである。

以上の機能を組み合わせると、さまざまなアプリケーションを構築できる。たとえば、格子状の線上を移動するライントレースカーに対して直進、90 度右旋回および 90 度左旋回の 3 つの命令を組合わせて目的地まで移動させるアプリケーションが考えられる。ここで、命令は格子状の交差点から次の交差点までの区間 (以下、1 ブロック) ごとに行われる。これらの命令は、パソコン上にある専用アプリケーションで編集し、無線のシリアル通信により KTM-02 に送られる。また、移動の途中に障害物を設置し、それを避けるようにすることで命令の組合わせの難易度を高めることができる。さらに、距離センサやカメラを組み合わせると、より高度なアプリケーションが作成できる。

本教材で、MCU や CPLD によるデバイスの制御方法を身につける一助になることを期待する。

1.1 本書の構成

本書の構成は以下のとおりである。

- 第 1 章:KTM-02 を構成するデバイス、その部位および付属のケーブル類の名称について述べる。あわせて、開発に必要なソフトウェアも記す。
- 第 2 章:デバイス、基板および付属品との接続方法について説明する。
- 第 3 章:デバイスの中で使用上注意が必要な箇所を詳説する。
- 第 4 章:KTM-02 を動作させる電源の接続方法とその時の注意点について述べる。
- 第 5 章:Bluetooth モジュールおよび Bluetooth-USB アダプタの使用方法について述べる。
- 第 6 章:赤外線センサの感度調整および LCD のコントラスト調整について説明する。
- 第 7 章:開発手順について詳説する。
- 第 8 章:KTM-02 に搭載されている MCU である SH-7125F の端子と各デバイスの接続状況について述べる。
- 付録 A:使用している部品一覧を示す。
- 付録 B:KTM-02 の回路図を示す。

1.2 セット内容

ここでは、KTM-02 を構成する部品について写真とともに示す。あわせて、各部品の名称を定義する。

1.2.1 ベースボード

KTM-02 の基礎的な基板であるベースボードには、表 1.1 に示す部品がある。なお、図 1.1 では電源ボードとドットマトリクスディスプレイボードは取り外してある。

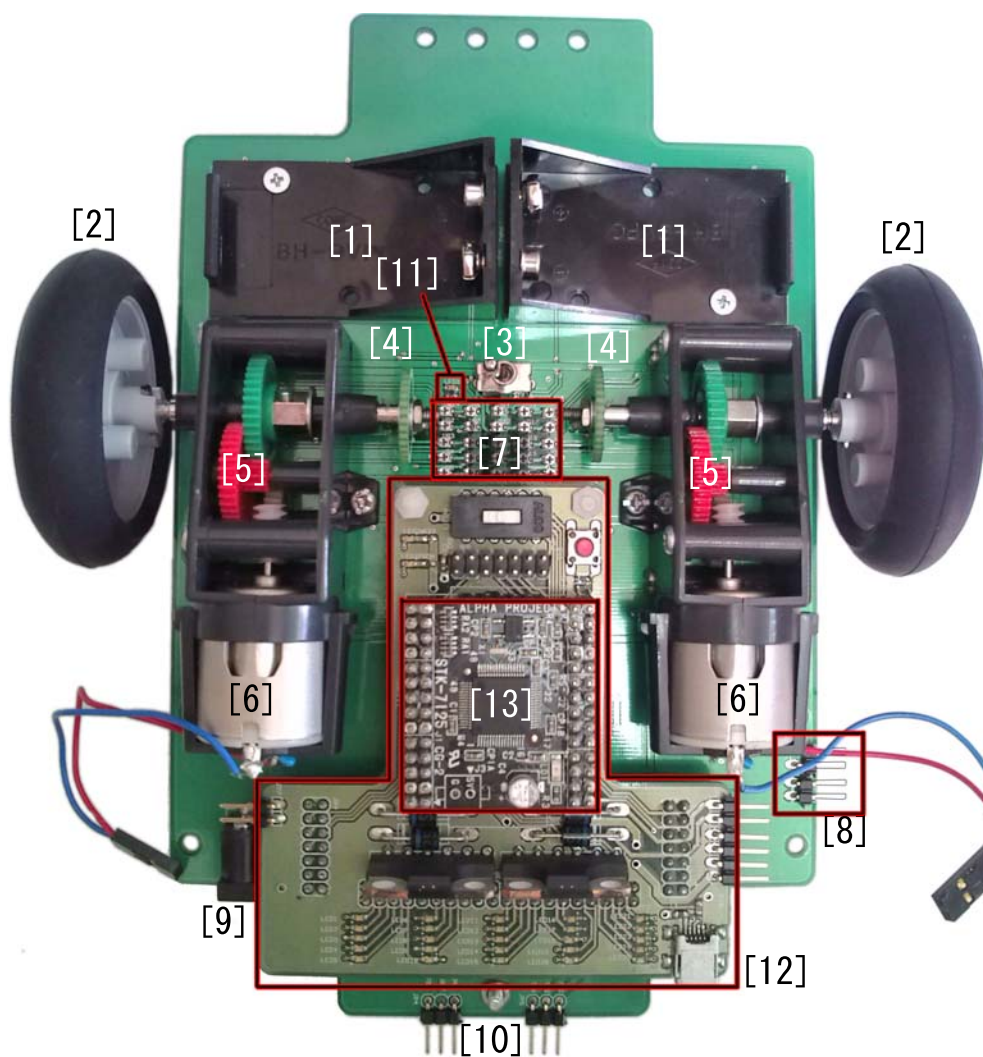


図 1.1: ベースボード (表)

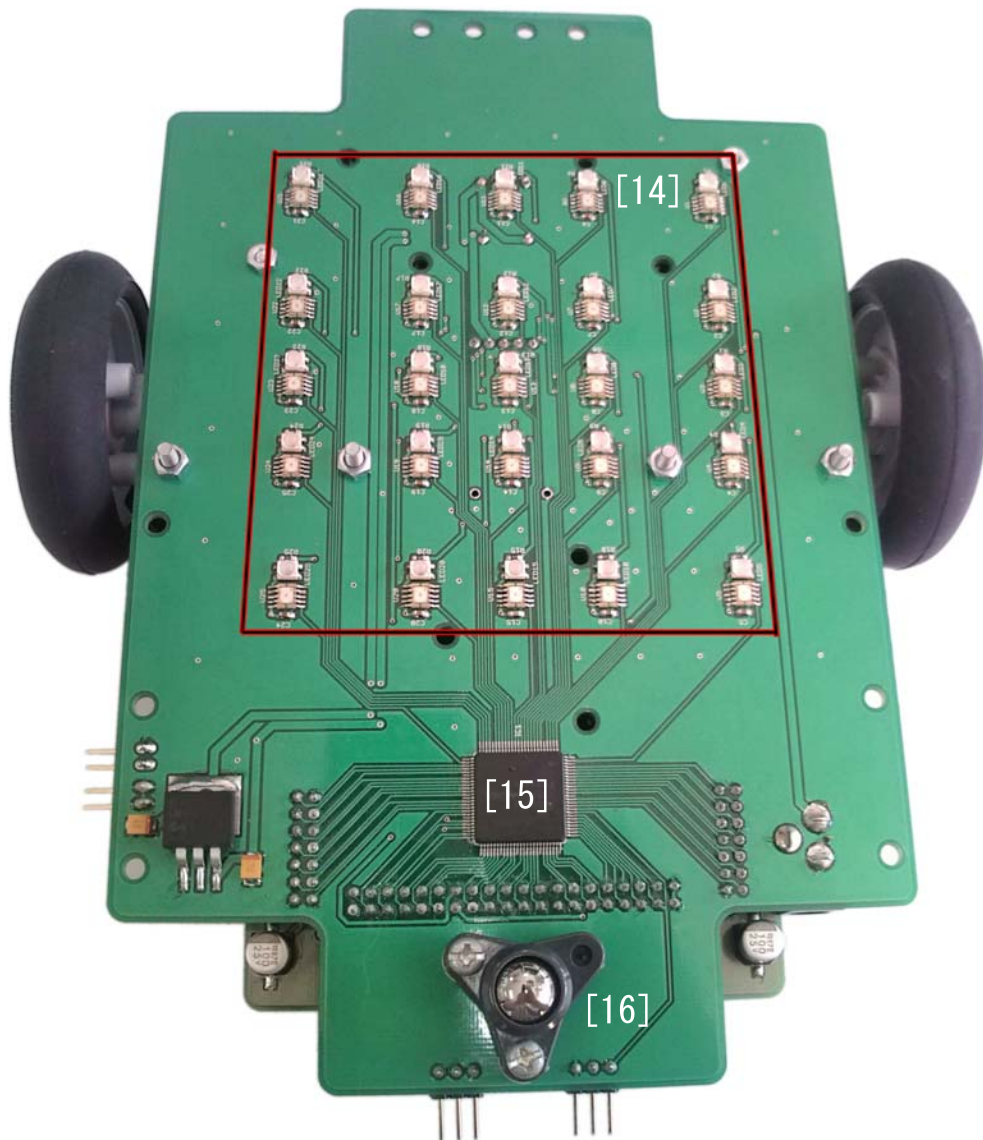


図 1.2: ベースボード (裏)

表 1.1: ベースボードの部品

名称	写真	役割
電池ボックス (ベースボード)	図 1.1-[1]	KTM-02 を動作させるための電源。 開発中は DC ジャックに AC アダプタを 装着し、電源とする。
タイヤ	図 1.1-[2]	KTM-02 が移動するときの足。
電源切替スイッチ	図 1.1-[3]	AC アダプタと電池のいずれを 電源として用いるか決めるスイッチ。
タイヤ回転数検知用 フォトインタラプタ	図 1.1-[4]	穴のあいた円盤をギアボックスの軸 に取り付けて、フォトインタラプタで 穴の有無を読み取ることで回転数や スピードを得る。
ギアボックス	図 1.1-[5]	ウォームギア。
モータ	図 1.1-[6]	DC モータ。
赤外線センサ用 トリマ	図 1.1-[7]	赤外線センサの感度を調整する。
電源ボード接続用端子	図 1.1-[8]	電源ボードの電源を使用するとき に用いる接続端子。
DC ジャック	図 1.1-[9]	AC アダプタを接続する端子。
ベースボード用 JTAG 端子	図 1.1-[10]	CPLD にダウンロードするとき に用いる端子。
電源パイロット LED	図 1.1-[11]	電源が供給されたときに光る LED。
MCU 搭載ボード	図 1.1-[12]	MCU ボードを搭載するボード。 ベースボードから取り外せる。
MCU ボード	図 1.1-[13]	SH-7125F があるボード。 MCU 搭載ボードから取り外せる。
赤外線センサおよび 赤外線 LED	図 1.2-[14]	ラインを読み取るためのセンサ
赤外線センサ通信用 CPLD	図 1.2-[15]	赤外線センサにより読み取った 情報を MCU に送信するための素子。
ボールキャスタ	図 1.2-[16]	

1.2.2 MCU 搭載ボード

図 1.3 にベースボードに接続された MCU 搭載ボードを示す。この図では、MCU ボードは接続されていない。また、図中の [a] には多くのデバイスが密集しているため、図 1.3[a] に描いた矢印の方向から見た場合の様子を図 1.4 に拡大して示す。さらに、図 1.5 には MCU 搭載ボードに取り付けられる Bluetooth モジュールを示す。

表 1.2 に、MCU 搭載ボードにある部品の名称を定義する。

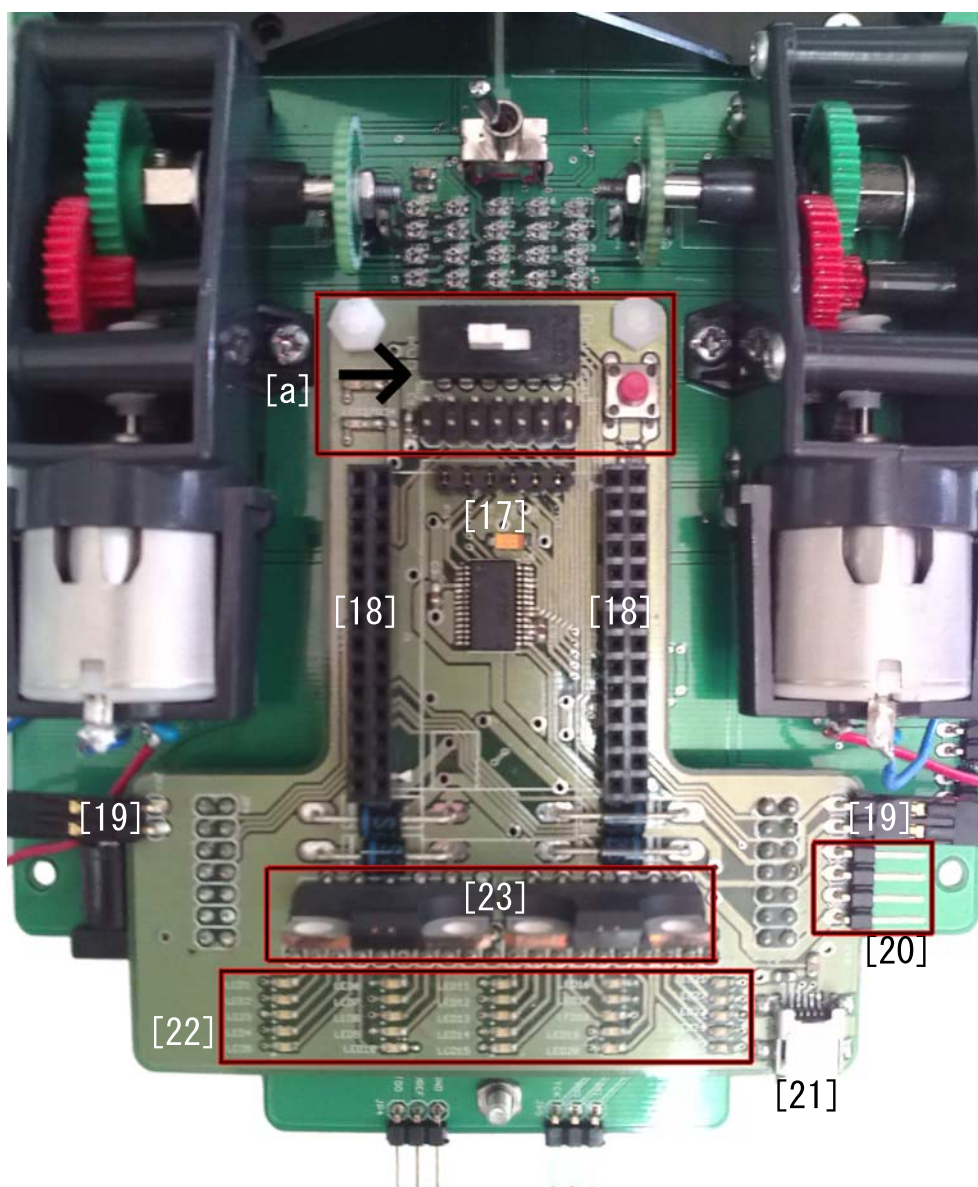


図 1.3: MCU 搭載ボード

表 1.2: MCU 搭載ボードの部品

名称	写真	役割
シリアル通信用 Bluetooth モジュール 接続コネクタ	図 1.3-[17]	KTM-02 と無線のシリアル通信で 命令などを送受信するための Bluetooth モジュールを接続 するコネクタ。
MCU 接続コネクタ	図 1.3-[18]	MCU ボードと接続するためのコネクタ。
モータ接続コネクタ	図 1.3-[19]	モータと接続するためのコネクタ。
モータ用電源コネクタ	図 1.3-[20]	モータを駆動するための電源と接続する コネクタ。
USB コネクタ	図 1.3-[21]	KTM-02 と有線のシリアル通信を行うとき 用いる USB コネクタ。
赤外線センサ確認用 LED	図 1.3-[22]	赤外線センサの感知状況を確認するた めの LED。
モータドライバ	図 1.3-[23]	モータの制御をおこなうドライバ。
リセットスイッチ	図 1.4-[24]	MCU をリセットするスイッチ
MCU モード切替 スイッチ	図 1.4-[25]	MCU にダウンロードするモードと プログラムが動作するモードを 切り替えるスイッチ。
ドットマトリクスディスプレイボード 接続端子	図 1.4-[26]	交差点の形状を表示するボードと 接続するための端子。
送信状態表示 LED	図 1.4-[27]	有線のシリアル通信をしたとき、MCU への 送信データの有無を表示する LED。
受信状態表示 LED	図 1.4-[28]	有線のシリアル通信をしたとき、MCU への 受信データの有無を表示する LED。
Bluetooth モジュール	図 1.5-[29]	無線のシリアル通信を行うための モジュール。

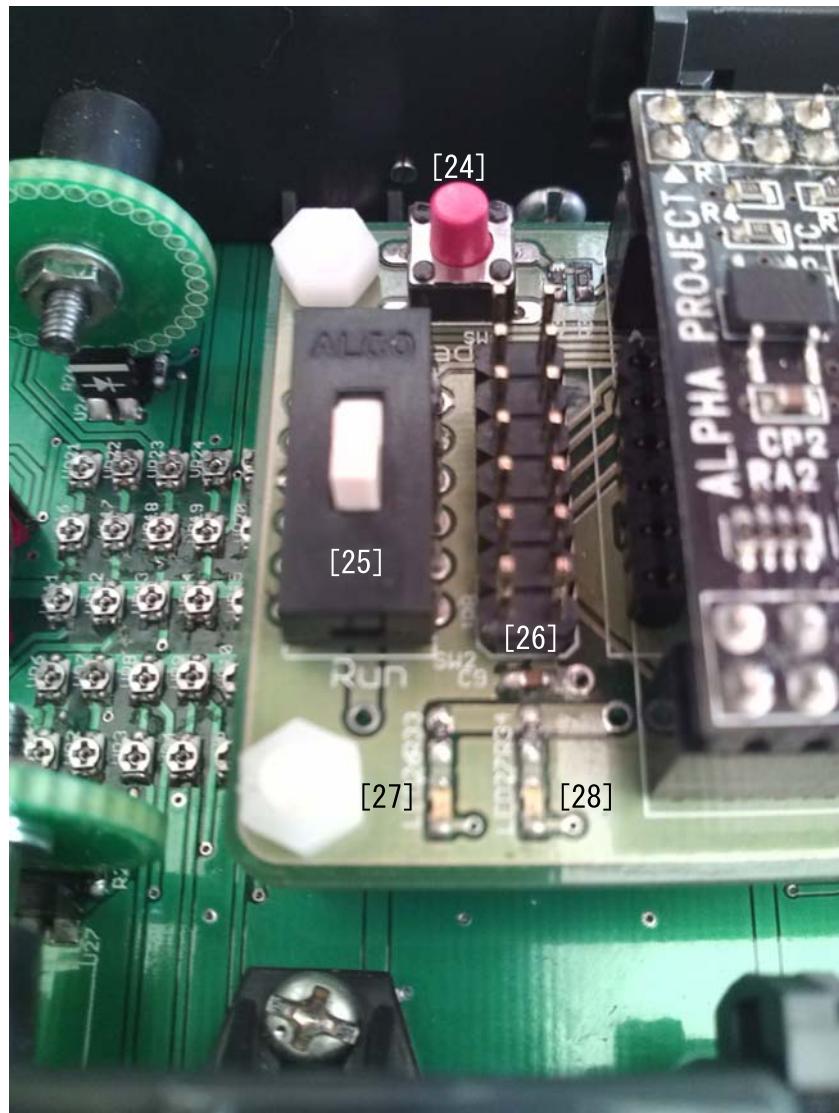


図 1.4: MCU 搭載ボード (一部拡大)



図 1.5: Bluetooth モジュール

1.2.3 ドットマトリクスディスプレイボード

赤外線センサより得られた情報をもとに、MCU による交差点形状の認識をし、その結果を表示するためのボードである。図 1.6 に表面、図 1.7 に裏面の様子を示す。また、各部品名を表 1.3 に示す。

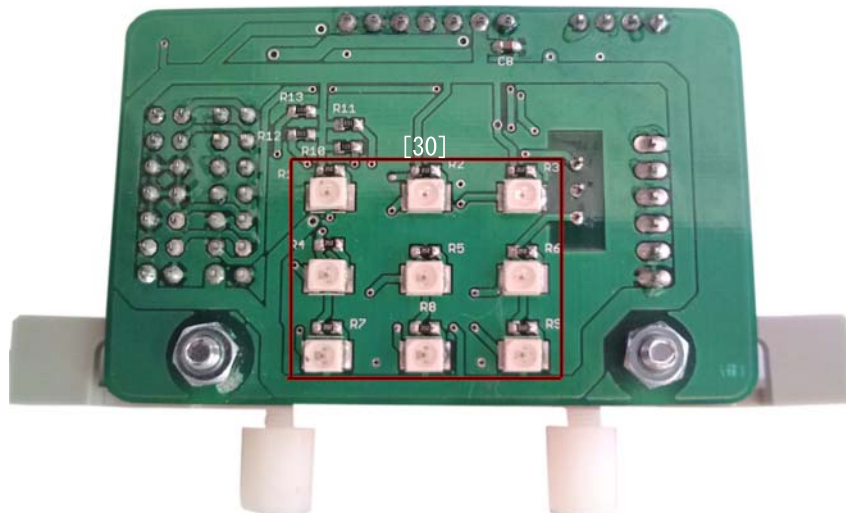


図 1.6: ドットマトリクスディスプレイボード (表)

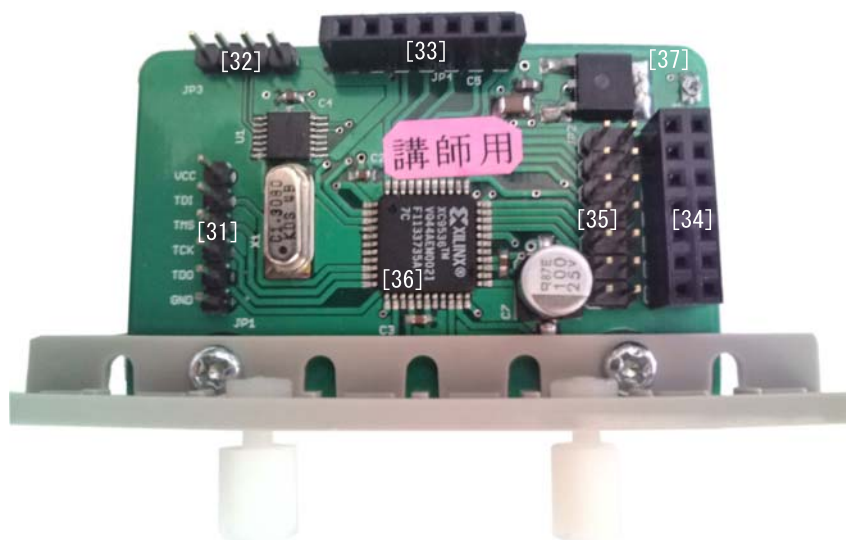


図 1.7: ドットマトリクスディスプレイボード (裏)



図 1.8: カメラ



図 1.9: 距離センサ



図 1.10: LCD

表 1.3: ドットマトリクスディスプレイボードの部品

名称	写真	役割
ドットマトリクス ディスプレイ用 LED	図 1.6-[30]	マトリクス状に配置した LED .
ドットマトリクス ディスプレイ用 JTAG 端子	図 1.7-[31]	ドットマトリクスディスプレイボードに搭載された CPLD にダウンロードする時に用いる JTAG 端子 .
カメラ接続端子	図 1.7-[32]	カメラと接続するときに用いる端子 .
距離センサ端子	図 1.7-[33]	距離センサと接続するときに用いる端子 .
LCD 接続端子	図 1.7-[34]	LCD と接続するときに用いる端子 .
MCU 搭載ボード接続端子	図 1.7-[35]	MCU 搭載ボードと接続するときに用いる端子 .
カメラ-距離センサ切替用 およびドットマトリクス ディスプレイ用 CPLD	図 1.7-[36]	カメラと距離センサを切替えたり, MCU から送られる交差点情報を LED で表示するための CPLD .
LCD 輝度調整用トリマ	図 1.7-[37]	LCD に表示する文字の輝度を調整するトリマ .
カメラ	図 1.8	シリアルカメラ .
距離センサ	図 1.9	距離センサ .
LCD	図 1.10	2 行 8 文字だけ表示できるキャラクタ LCD .

1.2.4 電源ボード

無線のシリアル通信により KTM-02 を操作するとき，AC アダプタより電源を得ていては無線にした意味がない．そこで 006P 充電電池を用いるが，電池ボックス (ベースボード) のみでは必要な電力が得られないため，電源ボードを追加する．

図 1.11 に電源ボード，表 1.4 に部品一覧を示す．



図 1.11: 電源ボード

表 1.4: 電源ボードの部品

名称	写真	役割
電源供給端子	図 1.11-[37]	006P 充電電池により得られる電力を供給する端子．
電池ボックス (電源ボード)	図 1.11-[38]	006P 充電電池フォルダ．

1.2.5 ケーブル類

KTM-02 にはいくつかの端子があり、それらをつなぐためのケーブルが備わっている。図 1.12 から図 1.15 にケーブルの外観を示し、表 1.5 にはそれらの名称および役割を示す。



図 1.12: USB ケーブル

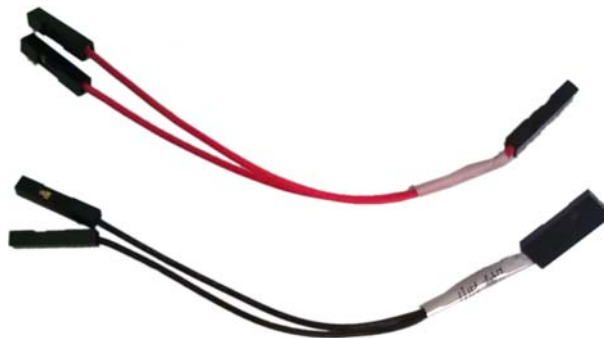


図 1.13: 電源ボード用電源ケーブル



図 1.14: MCU 搭載ボード-ドットマトリクスディスプレイボード接続ケーブル



図 1.15: AC アダプタ

表 1.5: ケーブル類

名称	写真	役割
USB ケーブル	図 1.12	有線のシリアル通信をするときに用いる .
電源ボード用電源ケーブル	図 1.13	電源ボードからの電源をモータやその他のデバイスを駆動させる電源とすると用いるケーブル .
デバッグ用モータ電源ケーブル	図??	プログラム作成時にモータを動作させるために用いるケーブル .
MCU 搭載ボード- ドットマトリクスディスプレイボード 接続ケーブル	図 1.14	MCU 搭載ボードとドットマトリクスディスプレイボードを接続するためのケーブル .
AC アダプタ	図 1.15	100[V] 交流電源を 5[V] 直流電源に変換するアダプタ .

1.2.6 その他

これまで紹介した部品のほかに、表 1.6 に示す部品が備わっている。

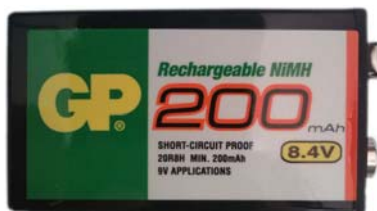


図 1.16: 006P 充電電池



図 1.17: Bluetooth-USB アダプタ

表 1.6: その他

名称	写真	役割
006P 充電電池	図 1.16	無線のシリアル通信をするときに用いる電源。6つある。
Bluetooth-USB アダプタ	図 1.17	パソコンに装着することで Bluetooth による無線のシリアル通信が行える。

1.3 開発に必要なソフトウェア

次に示すソフトウェアを用いて KTM-02 を開発する。

- High-performance Embedded Workshop(54 ページにワークスペースの作成方法を記載。)
- Flash Writer EX for SH7125F(64 ページに使用方法を記載。)
- Bluetooth-USB アダプタのドライバ (35 ページにインストール方法を記載。)
- Tera Term
- ISE Web PACK(CPLD に対して回路を構成する場合)

第2章 接続方法

この章では，部品および各個所の接続方法を説明する．

誤った接続をすると故障の原因になるため，熟読すること．

2.1 部品の接続

図 2.1 で部品の接続状況，方法および個数を示す．ここで，赤色の線はケーブルコネクタにより接続していることを，青色の線はピンソケットにより接続していることを，黒色の線はパターンにより接続していることを意味している．次節以降では，ケーブルコネクタおよびピンソケットにより接続する方法について詳しく説明する．あわせて，図 2.1 にある接続番号である《 》を節の題名に付す．

図 2.1 には，部品の接続状況以外に MCU および CPLD を実装するとき用いる言語も表している．SH-7125F に対しては，C/C++言語にて記述したプログラムをルネサスエレクトロニクス社製 SuperH ファミリ用 C/C++コンパイラによりコンパイルする．また，CPLD に対しては，Verilog HDL にて記述したコードをザイリンクス社製 ISE Web PACK により論理合成など行う．

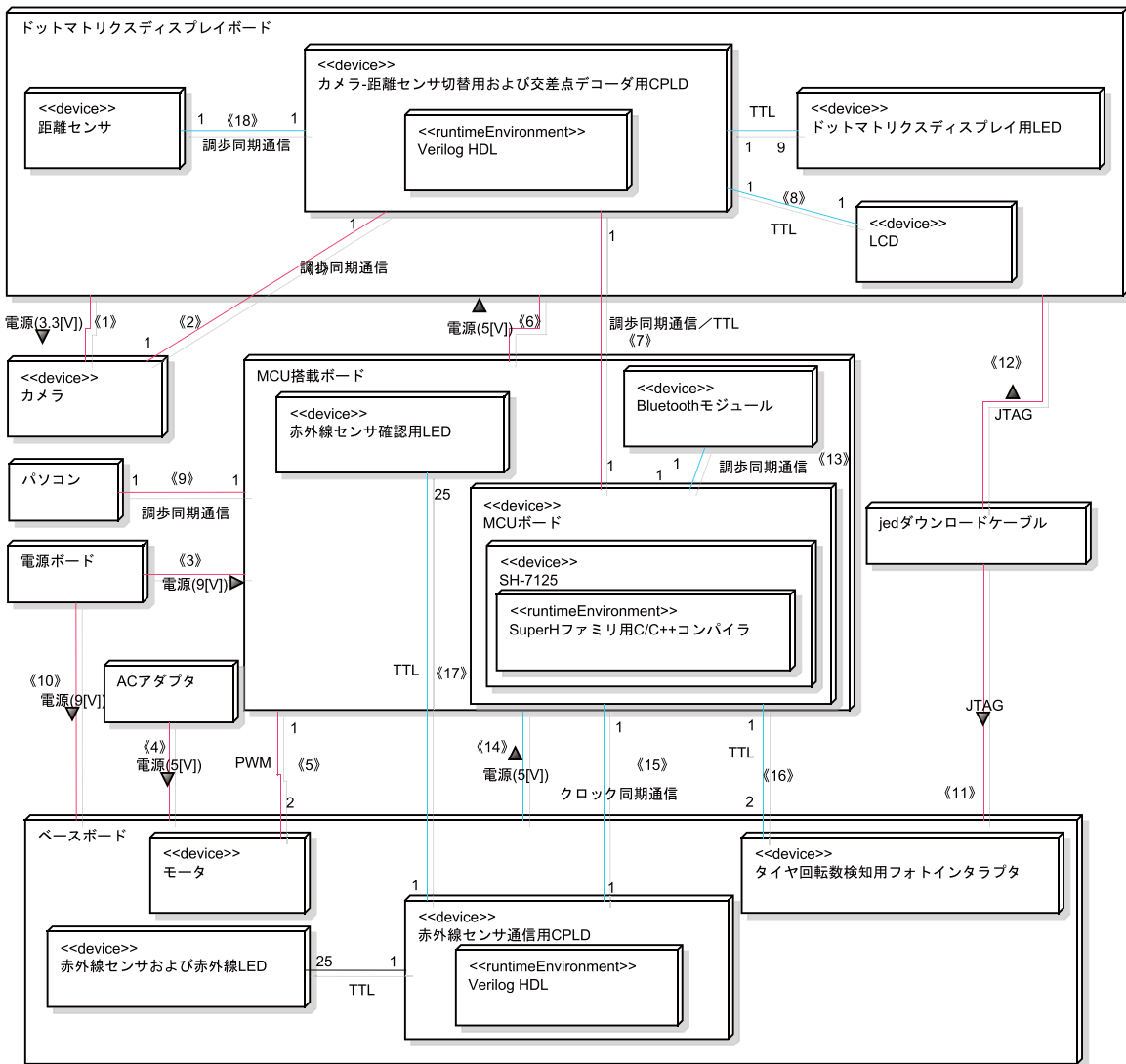


図 2.1: 部品の接続

2.1.1 ケーブルコネクタによる接続

ケーブルコネクタによる接続は、もっとも誤る可能性が高いので、特に熟読すること。

ドットマトリクスディスプレイボードとカメラ (《1》および《2》)

接続した様子を図 2.2 の赤色矢印から撮影した様子を図 2.3 に示す。左から黒、緑、黄および赤になるようにケーブルコネクタを接続すること。

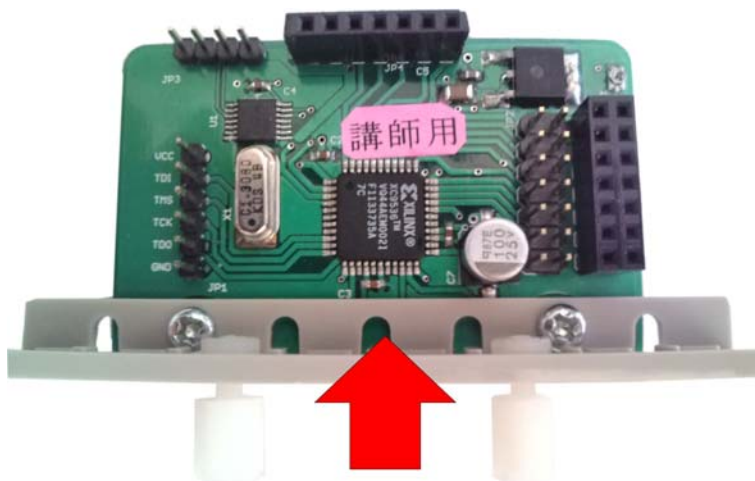


図 2.2: ドットマトリクスディスプレイボードの撮影方向



図 2.3: ドットマトリクスディスプレイボードとカメラの接続

MCU 搭載ボードとドットマトリクスディスプレイボード (《6》および《7》)

ケーブルをドットマトリクスディスプレイボードに接続した様子を図 2.4 に、MCU 搭載ボードに接続した様子を図 2.5 に示す。

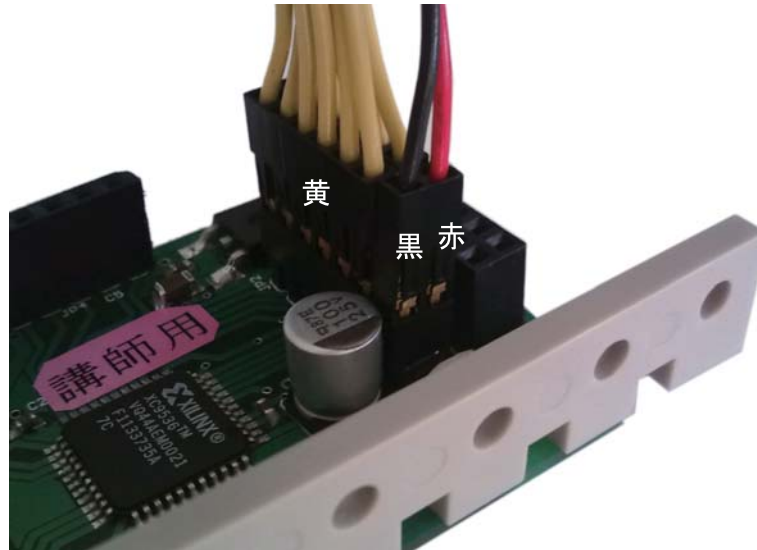


図 2.4: ドットマトリクスディスプレイボードと MCU 搭載ボードの接続

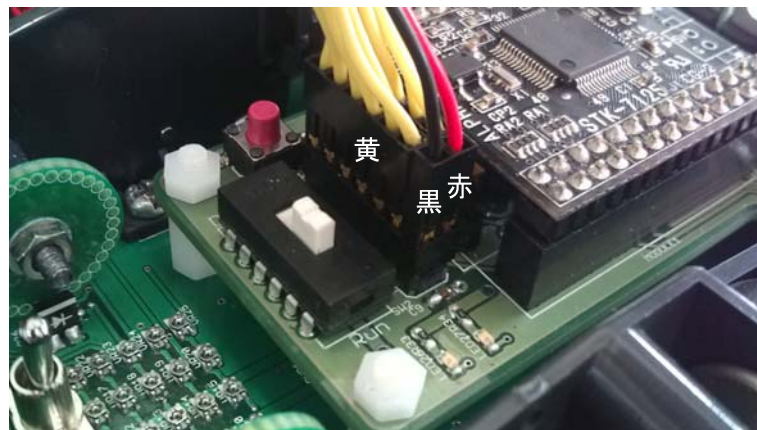


図 2.5: MCU 搭載ボードとドットマトリクスディスプレイボードの接続

今回の接続には MCU 搭載ボード-ドットマトリクスディスプレイボード接続ケーブルを用いる。このケーブルには電源用である 2 端子のケーブル 1 本と、信号伝達用である 6 端子のケーブル 2 本がある。このケーブルを図 2.6 および 2.7 に示す番号と対応するように接続する。

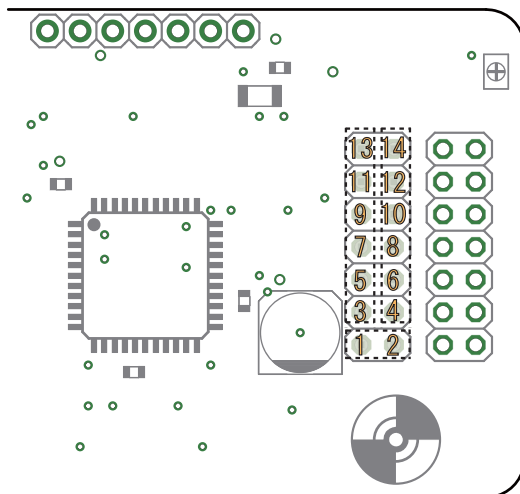


図 2.6: ドットマトリクスディスプレイボードと MCU 搭載ボードの接続 (詳細)

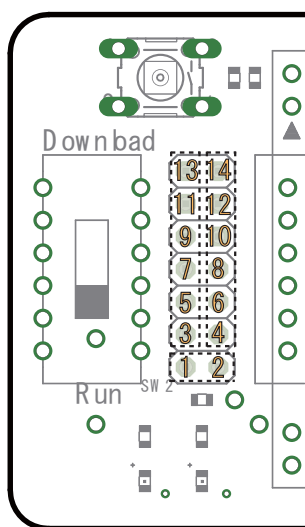


図 2.7: MCU 搭載ボードとドットマトリクスディスプレイボードの接続 (詳細)

電源ボード (ベースボード: 《10》, MCU 搭載ボード: 《3》)

006P 充電電池は無線のシリアル通信により KTM-02 を操作するとき用いられる。ベースボードには2つの 006P 充電電池フォルダが備え付けられているが、モータを含めてすべての電力を賄うことは難しいため、006P 充電電池フォルダが4つある電源ボードを追加して用いる。

電源ボードには図 2.8 に示す端子があり、ここから供給される電力を分配し、モータやその他のデバイスで用いる。この端子に 12 ページの図 1.13 で示した電源ボード用電源ケーブルを取り付ける。取り付けの様子を図 2.9 に示す。

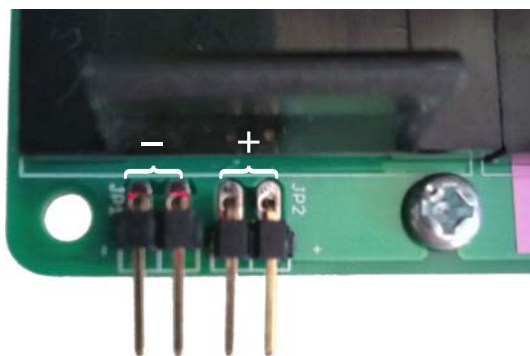


図 2.8: 電源ボードの接続端子



図 2.9: 電源ボードに電源ボード用電源ケーブルを取り付けた様子

電源ボード用電源ケーブルを取り付けるうえで注意点が2点ある。1つはプラス極性側は赤色、マイナス極性側は黒色の電源ボード用電源ケーブルを用いなければならない。もう1つは、ケーブルの端が2つのレセプタと、そこから1つのレセプタ2つに分かれているが、電源ボードには2つのレセプタを接続する必要がある。1つのレセプタに分かれているのは、必要に応じてモータ駆動用電源とそれ以外の電源それぞれに供給するためである。なお、KTM-02はモータ駆動用電源と、それ以外の電源に分かれている。詳しくは第4章で説明する。

電源ボードとベースボードを接続している様子を図2.10に示す。これは2ページの図1.1にある[8]を拡大したものである。電源ボードとMCU搭載ボードを接続している様子を図2.11に示す。これは2ページの図1.1にある[20]を拡大したものである。なお、上記のいずれの画像もわかりやすくするため、電源ボードを取り外して撮影した。

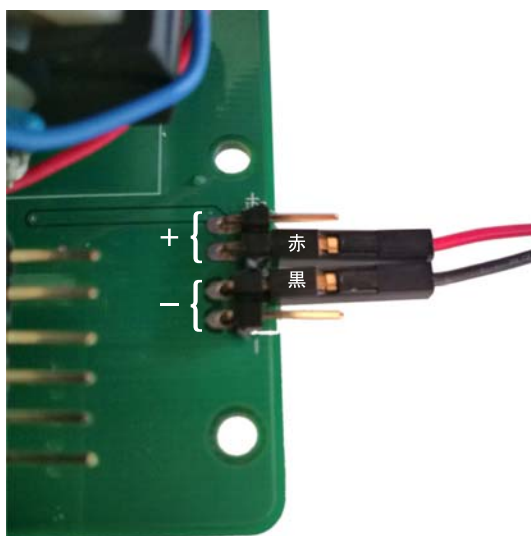


図 2.10: ベースボードに電源ボード用電源ケーブルを接続した様子

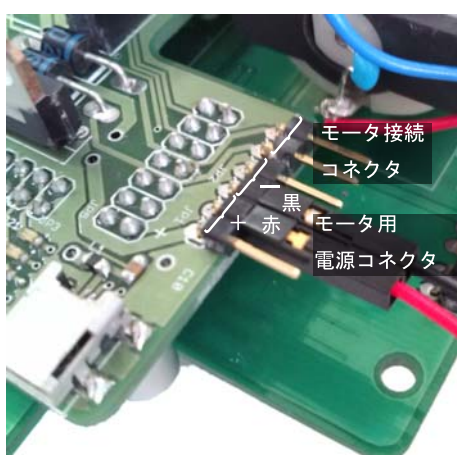


図 2.11: MCU 搭載ボードに電源ボード用電源ケーブルを接続した様子

モータ (《5》)

モータはベースボードにあるにもかかわらず、MCU 搭載ボードと接続する理由は、MCU 搭載ボード上にあるモータドライバにより制御されるためである。5 ページの図 1.3 で示した [19] および [20] はモータ接続コネクタである。これらを拡大した様子を図 2.12 および図 2.13 に示す。青色と赤色のケーブルを反対に接続すると、モータが逆回転してしまうので注意すること。

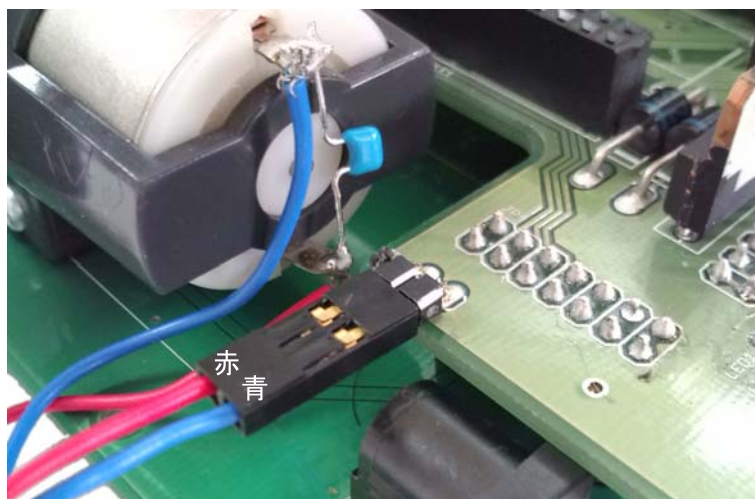


図 2.12: MCU 搭載ボードに左側モータを接続した様子



図 2.13: MCU 搭載ボードに右側モータを接続した様子

ベースボード用 JTAG 端子 (《11》)

この端子は赤外線センサ通信用 CPLD に HDL で構成した回路をダウンロードするとき用いる。CPLD へのダウンロードを行うには、図 2.14 のようにザイリンクス社製ダウンロードケーブルを接続する。ダウンロードケーブルには、端子の名称が書かれているので、それとボード上に書かれている文字(シルク)と照らし合わせて接続する。なお、ダウンロードケーブルには 1 本だけ使用しない端子がある。ダウンロードは ISE Web PACK に収録されている iMPACKT により行うが、方法については割愛する。



図 2.14: ベースボード用 JTAG 端子にダウンロードケーブルを接続した様子

ドットマトリクスディスプレイボード用 JTAG 端子 (《12》)

2.1.1 節と同様に、この端子を用いることで CPLD にダウンロードすることができる。ただしこの端子は電源供給源として使用することはない。図 2.15 に、ダウンロードケーブルを接続した様子を示す。



図 2.15: ドットマトリクスディスプレイボード用 JTAG 端子にダウンロードケーブルを接続した様子

パソコンと MCU 搭載ボード (《9》)

MCU に対して作成したプログラムをダウンロードしたり、有線のシリアル通信をしたりするとき、パソコンと MCU 搭載ボードを 12 ページの図 1.12 の USB ケーブルで接続する。図 2.16 には、MCU 搭載ボードとパソコンを USB ケーブルにより接続した様子を示す。

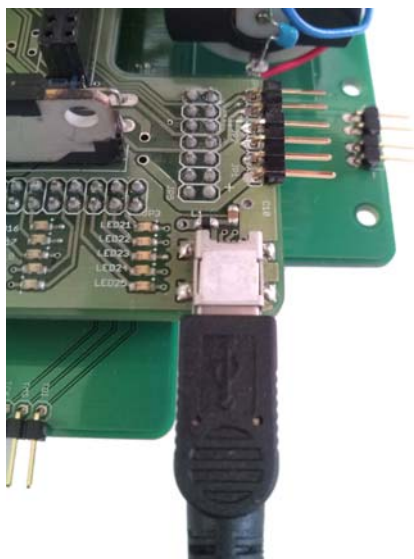


図 2.16: USB ケーブルによりパソコンと MCU 搭載ボードを接続した様子

AC アダプタ (《4》)

AC アダプタは、プログラム作成時など、KTM-02 を無線のシリアル通信により操作するとき以外に用いられる。2 ページの図 1.1 で示した [9](DC ジャック) に AC アダプタを取り付けた様子を図 2.17 に示す。

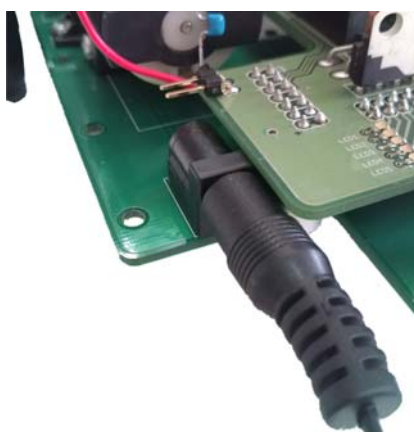


図 2.17: ベースボードに AC アダプタを取り付けた様子

2.1.2 ピンソケットによる接続

ピンソケットによる接続は、ケーブルコネクタによる接続と比べて間違えることが少ない。しかし、強引に脱着するとピンヘッドが変形したり、場合によっては折れてしまう場合があるので注意が必要である。

距離センサ (《18》)

距離センサは、8 ページの図 1.7 にある [33] と接続する。その様子を図 2.18 および図 2.19 に示す。ドットマトリクスディスプレイボードは KTM-02 前方に取り付けられるため、結果として距離センサにより前方にある物体までの距離を測ることができる。

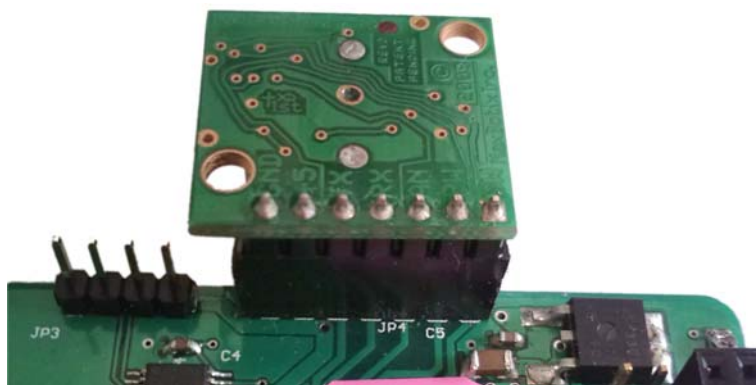


図 2.18: ドットマトリクスディスプレイボードに距離センサを接続した様子 (裏)

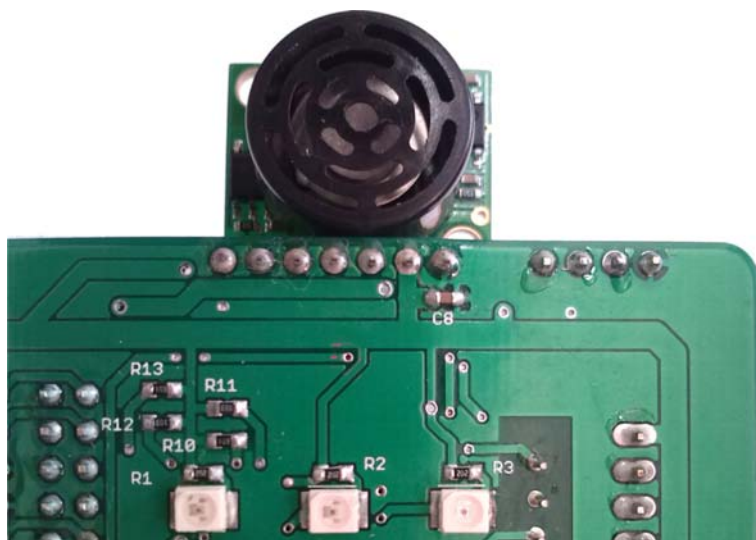


図 2.19: ドットマトリクスディスプレイボードに距離センサを接続した様子 (表)

LCD(《8》)

LCD は図 2.20 に示すように、ドットマトリクスディスプレイボードにある LCD 接続端子と接続して使用される。主な LCD の用途は、プログラムのデバッグ中に変数の値などを確認するために用いる。このため、ドットマトリクスディスプレイボードと LCD はピンヘッダとピンソケットのみの簡易的な固定しかしておらず、強度は期待できない。よって、ベースボードにドットマトリクスディスプレイボードを固定したときには、コネクタが破損してしまう可能性があるため、LCD は取り付けないようにすること。



図 2.20: ドットマトリクスディスプレイボードに LCD を接続した様子

Bluetooth モジュール(《13》)

無線のシリアル通信を行うとき、5 ページの図 1.3 にある [17] に Bluetooth モジュールを接続する。接続した様子を図 2.21 に示す。このコネクタはロープロファイルであり、ピンの長さが短いため、確実に挿入すること。

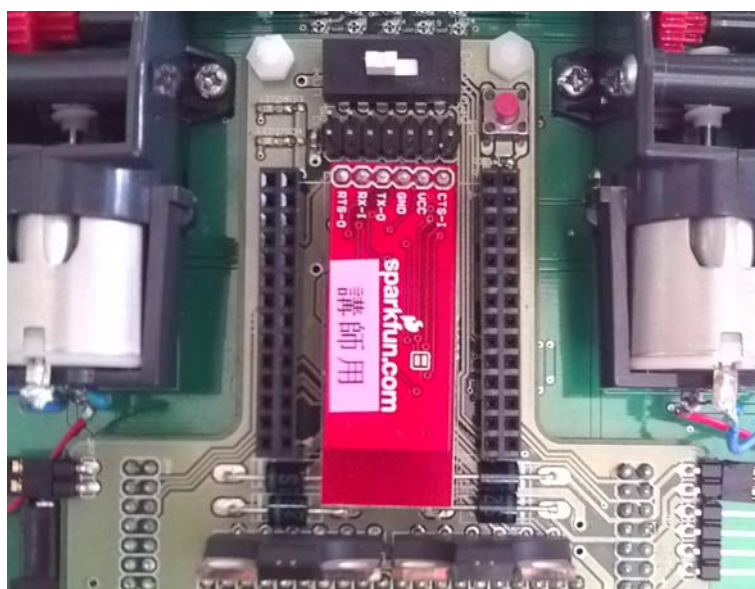


図 2.21: MCU 搭載ボードに Bluetooth モジュールを接続した様子

ベースボードと MCU 搭載ボード (《14》～《17》)

ベースボードと MCU 搭載ボードはコの字に配置された 3つのピンソケットとピンヘッダで接続される。注意点として、MCU 搭載ボードの裏側にある面実装の電解コンデンサ (図 2.22) が大変脆いので MCU 搭載ボードを取り外すとき、これを触らないようにすること。なお、KTM-02 を使用するうえで基本的には MCU 搭載ボードを取り外す必要はない。

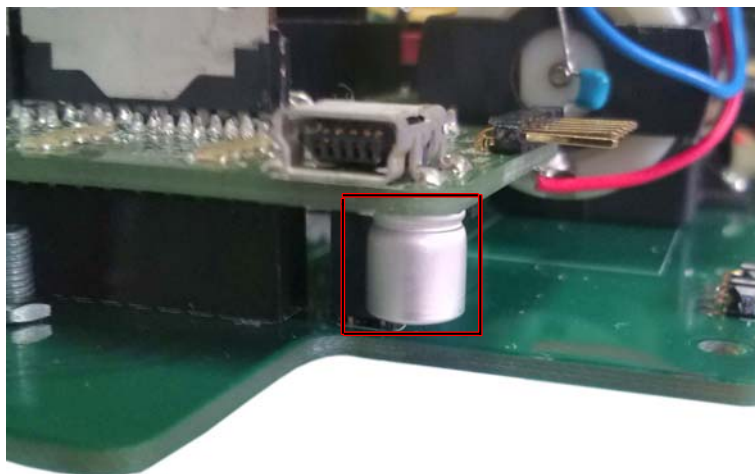


図 2.22: MCU 搭載ボード裏側にある電解コンデンサ

第3章 デバイスの詳細

3.1 シリアル通信によりデータを送受信するデバイス

KTM-02 にはシリアル通信によりデータを送受信するデバイスが備わっており、それらの中には排他的に使用するものもある。この章では、このようなデバイスが MCU に接続されている状況について説明する。MCU の各チャンネルにつながっているデバイスと通信方式を表 3.1 に記す。

表 3.1: シリアル通信によりデータを送受信するデバイス

チャンネル	デバイス	通信方式
0	赤外線センサ	クロック同期通信
1	パソコン	調歩同期通信
2	カメラ 距離センサ	調歩同期通信 調歩同期通信

表 3.1 を補足説明する。チャンネル 0 につながっている赤外線センサは、正確にはベースボードにある赤外線センサ通信用 CPLD に接続されている。次に、チャンネル 2 につながっているカメラおよび距離センサは、正確にはドットマトリクスディスプレイボードにあるカメラ-距離センサ切替用およびドットマトリクスディスプレイ用 CPLD に接続されている。この 2 つのデバイスは、共通のチャンネルを使用するため、排他的に用いる。デバイスの切替は、端子 PA1 の出力により行うことができ、Low 出力にするとカメラ、High 出力にすると距離センサが使用できる。この切替機能は CPLD によるものであり、デフォルトではそのような回路が CPLD に構築されている。別の回路を構成する方法については、7.1 節を参照すること。

3.1.1 赤外線センサ

赤外線センサは図 3.1 のように赤外線 LED と隣接しており、路面の状況、言い換えれば黒いか白いかを検知する。図 3.1 では、路面が黒く、赤外線があまり反射しない様子を示している。このようなセンサと LED のペアがベースボード裏側に 25 個設置してある。そして、検知した結果は CPLD に送られ、クロック同期通信にて MCU に送られる。

3.1.2 カメラ

今回用いているカメラは、調歩同期通信にて撮影した画像データを送信し、また、さまざまなコマンドを受信する。なお、このデバイスは動作電圧が 3.3[V] であるので、レベルシフタ IC により信号レベルを 5[V] と 3.3[V] に変換している。カメラへのコマンドについては、カメラのデータシートを参照すること。

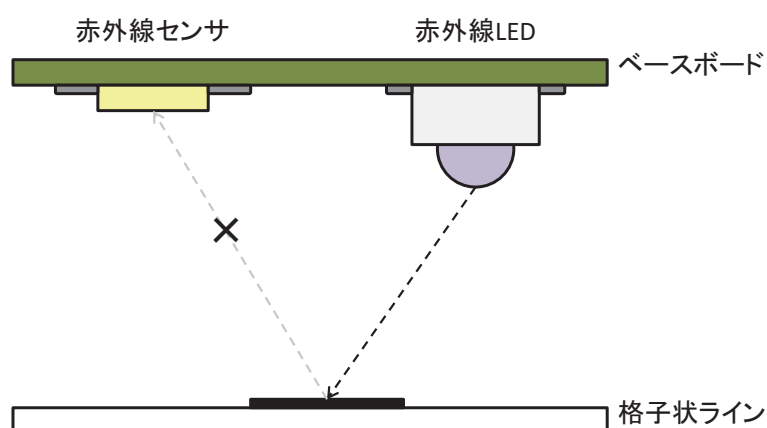


図 3.1: 赤外線センサ

3.1.3 距離センサ

今回用いている距離センサは、センサ感知部の先にある物体までの距離を超音波にて検知する。インタフェースは、調歩同期通信、デューティ比およびアナログ値の3種類となっている。主に調歩同期通信により距離をMCUに送っているが、アナログ値でも読み取れるようにするため、PF0にその信号を送るように設計してある。アナログ値の場合には、MCUの周辺機能であるA/D変換を使用する必要がある。調歩同期通信の場合に送られてくるデータフォーマットは、距離センサのデータシートを参照すること。

3.2 LCDとドットマトリクスディスプレイ用LED

ドットマトリクスディスプレイボードにあるLCDおよびドットマトリクスディスプレイ用LEDは同一のI/Oポートを用いているため、排他的に制御する必要がある。この切替は、端子PA5の出力により行われ、High出力するとドットマトリクスディスプレイ用LED、Low出力するとLCDが使用できるよう、CPLD上に回路が構成されている。また、ドットマトリクスディスプレイ用LEDへのデコードを行う回路もCPLDに構成されている。デコード方法については52ページの表7.1を参照すること。

3.3 タイヤ回転数検知用フォトインタラプタ

ギアボックスの回転軸には小さな孔のある円盤が取り付けられており、その円盤の下には透過型フォトインタラプタがある。これにより、円盤が回転し、孔がない個所では発光部から照射される光が遮られ、ある個所では光が透過し、受光部で感知する。この光の感知する時間的間隔を調べることで、回転数がタイヤの回転スピードを割り出せる。また、光を感知した回数を数えることで、タイヤの進んだ距離が割り出せる。

フォトインタラプタから得られた信号はCPLDを介さず、直接MCUにつながっている。MCUでは、回転数および回転スピードを容易に計測ようにするため、MTUの外部クロックおよびインプットキャプチャ機能用端子それぞれにフォトインタラプタからの信号を接続している。接続している端子の詳細は、表8.4および表8.10を参照すること。

第4章 電源

この章では、各デバイスの動作電圧、電源の接続時の注意点を述べる。KTM-02 には 9[V]、5[V] および 3.3[V] で動作するデバイスが混在しており、やや複雑になっている。誤った電源を接続するとデバイスを破損することがあるので、注意すること。図 4.1 は、各デバイスの電源を表したものであり、各デバイスの動作電圧が書かれている。以降、この図を中心に説明していくので、必要に応じて参照すること。

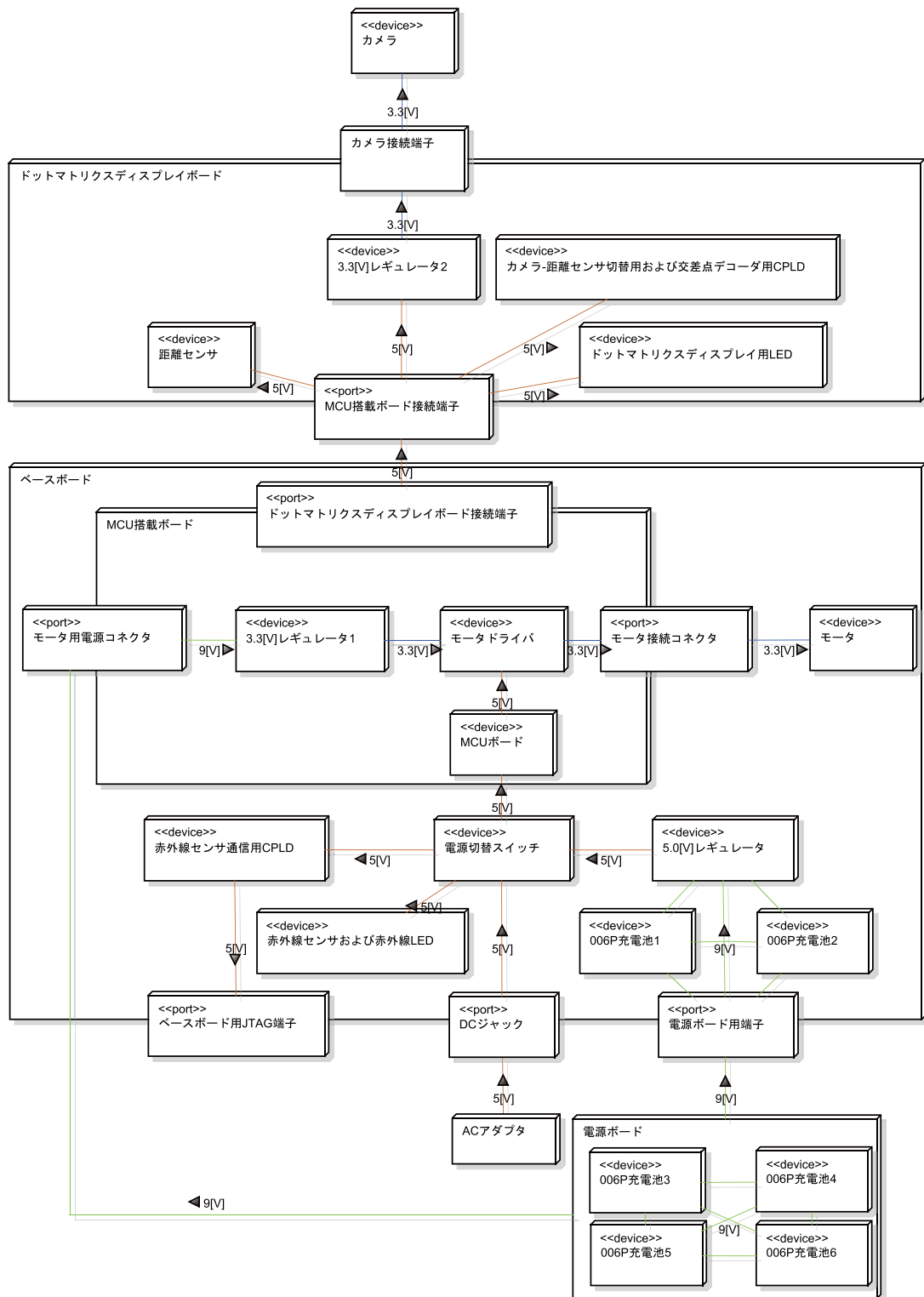


図 4.1: 電源に注目した接続状況

4.1 ACアダプタと006P充電電池の切替

KTM-02の電源として、ACアダプタおよび006P充電電池が用いられる。ベースボードにある電源切替スイッチにより、ベースボード上にある006P充電電池からの電源とDCジャックに接続したACアダプタからの電源を切り替えることができる。その様子を図4.2に示す。なお、電源ボード接続用端子からの電源は図4.1に示したようにベースボード上の006P充電電池と並列につながっているため、電源切替スイッチが006P充電電池側に倒されているとき、電源ボードの006P充電電池は使用される。また、ベースボード用JTAG端子にあるVREFおよびGNDは、上記いずれかの電源をつなげていれば電流が流れる。

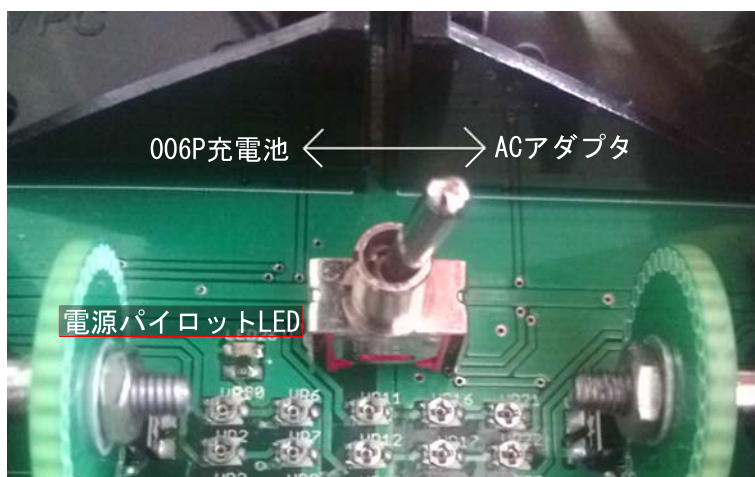


図 4.2: 電源切替スイッチ

4.2 モータ用電源コネクタ

モータ用電源コネクタは、場合に応じて5[V]もしくは9[V]の電源を接続することで、その電源はレギュレータにより3.3[V]にした後、モータの駆動電源として使用される。当然のことながら、5[V]と9[V]を同時にモータの駆動電源として使用することはできないので、決してモータ用電源コネクタに2種類の電源を同時に接続しないこと。

4.3 モータドライバ

図4.1に示したモータドライバには、3.3[V]および5[V]の電源が使用されていることが書かれている。通常、アナログデバイス用電源である駆動電源と、デジタルデバイス用電源は分離するため、このモータドライバは駆動電源は3.3[V]、制御をするための電源は5[V]とした。

第5章 Bluetoothの使用法

この章では、無線のシリアル通信を行う時に用いられる Bluetooth モジュールおよび Bluetooth-USB アダプタについて説明する。通常、プログラムの開発およびデバッグ中は、USB 接続によりパソコンと有線のシリアル通信を行う。そのうち、無線のシリアル通信を行う時にこれらの機材を用いる。

USB ケーブルと同時に Bluetooth モジュールを接続しないこと。

当然ながら、無線と有線を併用することはできないため、Bluetooth モジュールを使用するときには必ず USB ケーブルを抜くようにすること。これを誤ると、関係するデバイスの破壊を招く。

以降の節では、Bluetooth-USB アダプタならびに Bluetooth モジュールそれぞれ分けて説明し、最後に一連の通信方法について説明する。

5.1 Bluetooth-USB アダプタ

Bluetooth-USB アダプタは、パソコンの USB ポートに接続することで、Bluetooth のさまざまなプロファイルに基づく通信を行う機材である。ここで、プロファイルとは Bluetooth で接続する機材ごとに策定されたプロトコルのことである。代表的なものとしてマウスやキーボードなどの入力デバイスのために策定された HID(Human Interface Device profile)、携帯電話や PHS を介してインターネットにダイヤルアップ接続するためのプロファイルである DUN(Dial-up Networking Profile) などがある。今回、KTM-02 とパソコンの間ではシリアル通信を行うが、このとき用いられるプロファイルが SPP(Serial Port Profile) である。これは、パソコンに Bluetooth-USB アダプタを接続するとパソコンに COM ポートがあるように見え、また、MCU に Bluetooth モジュールを接続しデータを送受信すると Bluetooth の通信方式に変換されるプロファイルである。これにより、パソコンおよび MCU は、ともに一般のシリアル通信と同じように通信が行える。

近年のパソコンでは標準で Bluetooth-USB アダプタの代わりとなる Bluetooth トランシーバを搭載しているものも見受けられる。その場合には、教材に同梱してあるものを接続してはならない。最悪の場合、内蔵の Bluetooth を使用できなくなる。

5.1.1 セットアップ

この節では、本教材に含まれる Bluetooth-USB アダプタであるブラネックスコミュニケーション社製 BT-MicroEDR1X のセットアップ方法について述べる。なお、手持ちにパソコンに Bluetooth トランシーバが内蔵されている場合には読み飛ばしてよい。

インストール前に Bluetooth-USB アダプタを接続しないこと .

これを誤ると、正しく使えないうえ、修正する手間が大変かかる。必ず先にドライバのインストール作業を行ってから Bluetooth-USB アダプタを接続すること。

インストール

この節では、Windows 7 上で BT-MicroEDR1X を使用するときに行うインストール作業について述べる。Windows7 以外の OS については、BT-MicroEDR1X に添付されているユーザズ・マニュアルを参照し、インストールすること。

まず、下記 URL より Windows7 対応のドライバをダウンロードする。

<http://www.planex.co.jp/support/download/bluetooth/bt-microedr1x.shtml>

ダウンロードした圧縮ファイルを解凍すると、“Setup.exe” ファイルがあるので実行する。Setup.exe が開始されると、ユーザアカウント制御が働き、図 5.1 のようなダイアログが表示される。ここでは「はい」を選択する。次に、図 5.2 のようにインストールウィザードが開始される。続いて、図 5.3 のように使用許諾契約をする。次に、図 5.4 のようにインストール準備が完了する。その後、Bluetooth 用ドライバのインストールが開始される。



図 5.1: ユーザアカウント制御



図 5.2: インストールウィザード 1

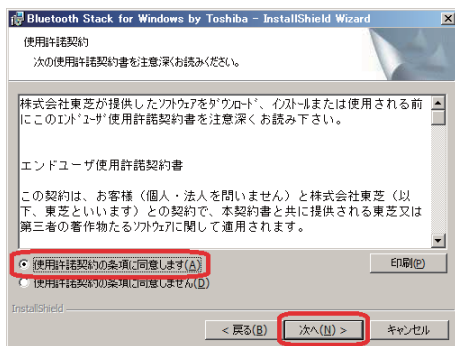


図 5.3: 使用許諾契約

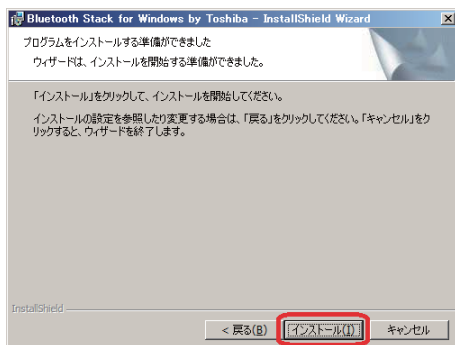


図 5.4: インストール準備完了

インストールが開始されてしばらくすると、図 5.5 のようなダイアログが表示される。さらにドライバのインストールが開始されると、図 5.6 のようなダイアログが表示される。その後図 5.7 のように Bluetooth デバイスを取り付けるように促されるので、Bluetooth-USB アダプタをパソコンに取り付けて OK ボタンを押す。これにより、ドライバのインストールが図 5.8 のように開始され、終了すると図 5.9 のダイアログが表示される。インストールが終了すると、図 5.10 のように再起動するように促されるので、適当なときに再起動すること。

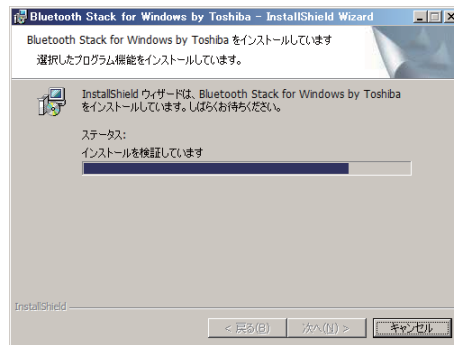


図 5.5: Bluetooth Stack for Windows by Toshiba

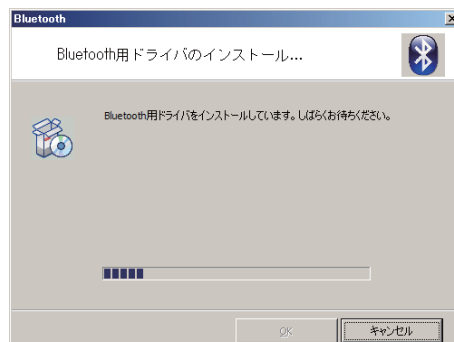


図 5.6: ドライバのインストール開始

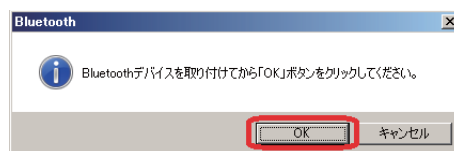


図 5.7: Bluetooth-USB アダプタの取り付け

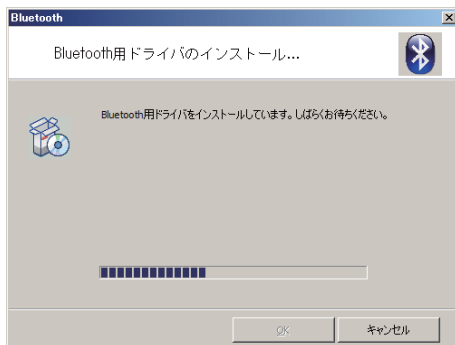


図 5.8: ドライバのインストール中

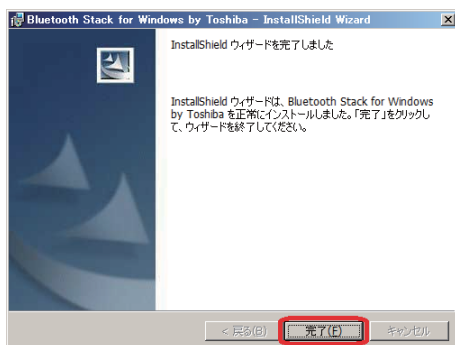


図 5.9: ドライバのインストール終了



図 5.10: 再起動

ペアリング

Bluetooth による通信では、サーバとクライアントの間でペアリングしたのちに通信を行う。Bluetooth-USB アダプタ付属のユーザーズ・マニュアルにはペアリングの方法が書かれているが、この節ではより具体的に述べる。

前節で Bluetooth-USB アダプタのインストールが終了し、OS の再起動を行うと図 5.11 のようなダイアログが表示される。ここではキャンセルをし、次の手順でペアリングを行う。

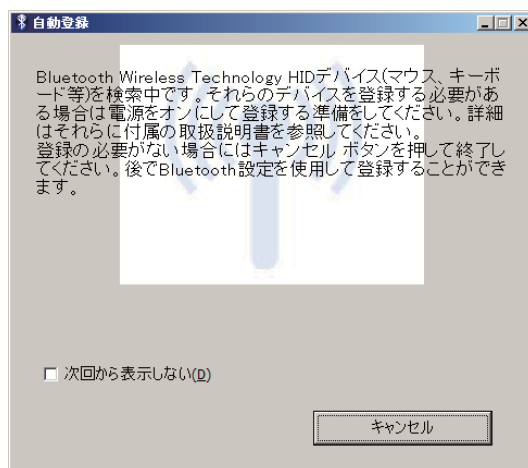


図 5.11: 再起動後に現れるダイアログ

一般的に、画面右下にあるインジケータの中から、図 5.12 に示すものを選び、ダブルクリックすると図 5.13 のような Bluetooth 設定ウィザードのダイアログが表示される。次に、28 ページに示したように Bluetooth モジュールを MCU 搭載ボードに取り付け、電源を入れる。その後「エクスプレスモード」を選択し、「次へ」ボタンを押すと、近くにある Bluetooth デバイスを図 5.14 のようにスキャンする。その結果、図 5.15 のように、Bluetooth デバイスが表示される。



図 5.12: Bluetooth 設定のためのインジケータ



図 5.13: Bluetooth 設定ウィザード



図 5.14: Bluetooth デバイスのスキャン

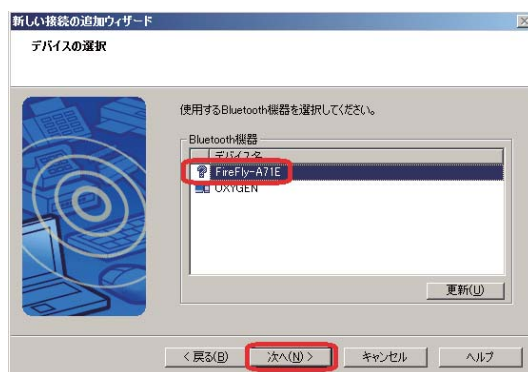


図 5.15: Bluetooth デバイス

今回、Bluetooth モジュールとの接続をしたいので、図 5.15 にある”FireFly-A71E” を選択する。なお、今回は”A71E” となったが、ほかの Bluetooth モジュールであるところの 4 文字は異なる。次に、Bluetooth モジュールと接続して対応するプロファイルを検知するとき図 5.16 のようなダイアログが表示される。その後、図 5.17 のように設定が完了したことを伝えるダイアログが表示される。以上で、初期設定は終了となる。



図 5.16: Bluetooth モジュールの対応プロファイルを検知中



図 5.17: 初期設定完了

接続

図 5.18 で Firefly-A71E をダブルクリックすることで接続がなされる。このダイアログをいったん閉じたのち、再び表示するには、図 5.12 に示したインジケータをダブルクリックすればよい。



図 5.18: Bluetooth モジュールと接続

取り外し

図 5.12 に示したインジケータを右クリックするとポップアップメニューが表示される。この中に「終了」があるので選択し、その後に Bluetooth-USB アダプタを取り外す。

5.2 Bluetooth モジュール

Bluetooth-USB アダプタと Bluetooth モジュールの間で調歩同期通信するには、少なくとも Bluetooth モジュールにボーレートの設定をしなければならない。Bluetooth モジュールに電源を入れてから通信開始までの流れを図 5.19 に示す。ボーレートの設定は、設定開始から設定完了までの間 (以下、設定可能時間) に行わなければならない、この時間を過ぎると調歩同期通信が開始される。なお、設定可能時間の初期値は 60 秒であり、この時間も設定可能である。

この節では Bluetooth モジュールの設定方法について説明する。

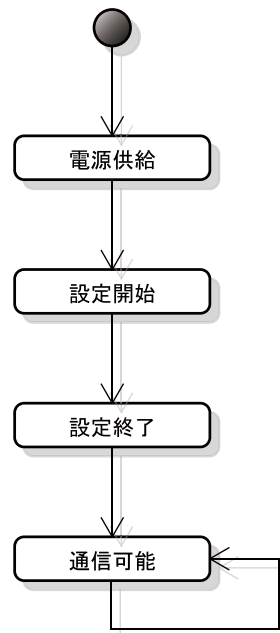


図 5.19: Bluetooth モジュールのフロー

5.2.1 設定可能時間

設定可能時間中，Bluetooth モジュールの赤色 LED がやや速く点滅している．そして時間が経過すると点滅は遅くなる．この LED により，設定可能時間中であるか見極めることができる．

Bluetooth モジュールの設定方法は，シリアル通信ターミナルソフトからコマンドを送信して行う．本書では，ターミナルソフトとして Tera Term を用いる．まず，電源投入後，すみやかにシリアル通信を開始する．今回の設定では，COM40 を使用している．もし，COM ポートがわからない場合，図 5.18 に表れるターゲットデバイスを右クリックするとメニューが表示されるので詳細を選択する．その結果，図 5.20 のようなダイアログが表示され，この中に COM ポートが明記されている．

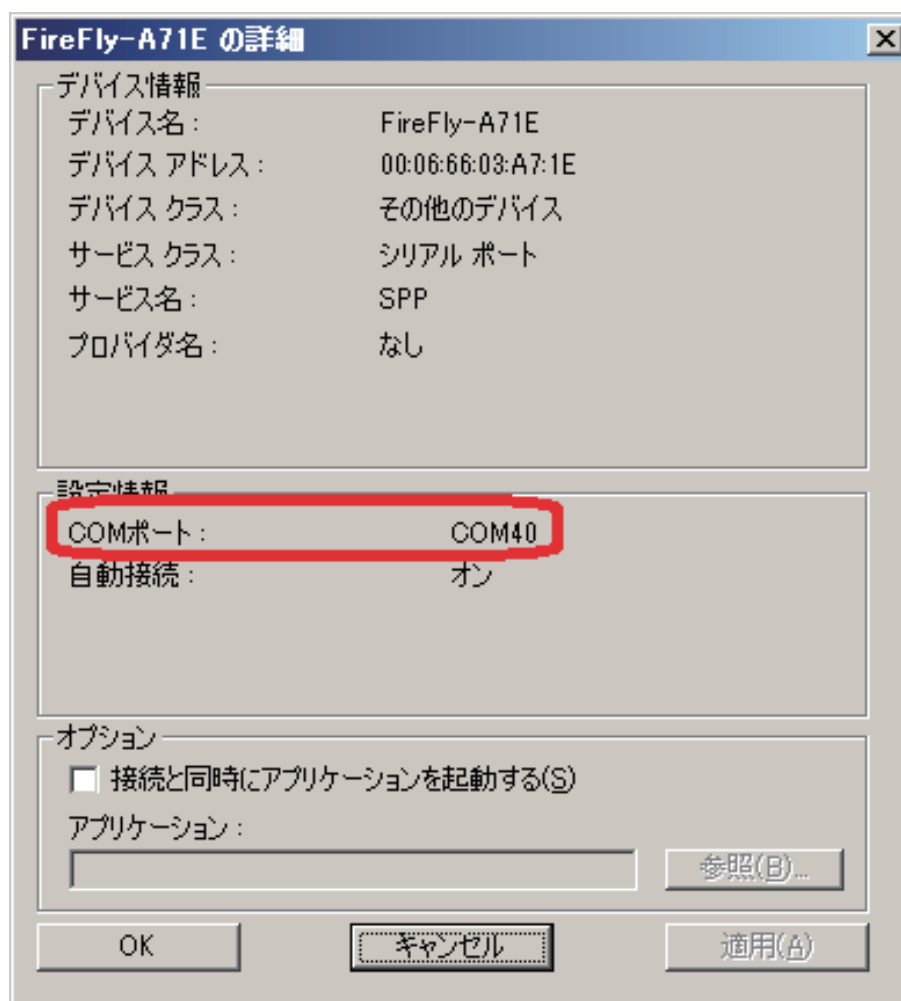


図 5.20: COM ポートの確認

Tera Term により COM40 と接続した後，“\$\$\$” を押すと，図 5.21 のように”CMD” と表示される．この状態は，設定可能状態であることを示している．なお，Bluetooth モジュールにある赤色 LED はこれまでよりさらに速く点滅をしているが，これは設定可能状態を表している．また，設定可能状態のときには Tera Term のボーレートが 115[kbps] 以下であれば通信が可能である．

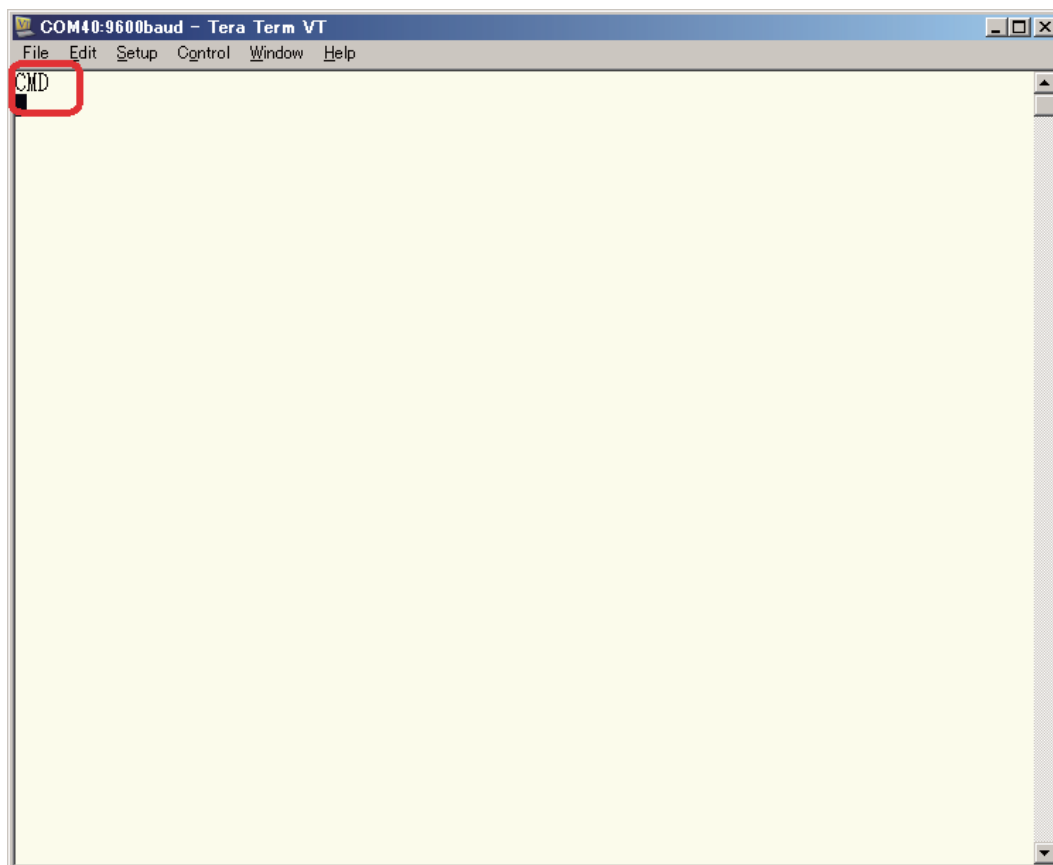


図 5.21: 設定可能状態

設定可能状態になった後，Bluetooth モジュールに対してさまざまな設定を行える．詳しくは Bluetooth モジュールのデータシートを参照すること．ここでは，最低限必要な設定のみを述べる．

設定の確認

Bluetooth モジュールの設定を確認するために下記のコマンドが用意されている．ボールド体で書かれている文字がコマンドであり，これを入力およびリターンキーを押すと現在の設定が表示される．

D :基本設定を表示する．表 5.1 にある名称および値が表示される．

E :拡張設定を表示する．表 5.2 を参照のこと．

O :その他の設定を表示する．表 5.3 を参照のこと．

表 5.1: 基本設定表示

名称	値	意味
BTA	00066603A71E	Bluetooth モジュールの bluetooth アドレス .
BTName	FireFly-A71E	Bluetooth モジュールの名前 .
Baudrt(SW4)	115k	ボーレート .
Parity	None	パリティビット . None=なし, Even=偶数ビット, Odd=奇数 .
Mode	Slav	動作モード . Slav=スレーブ, Mstr=マスタ, Trig=トリガ, Auto=オート .
Authen	0	認証の有無 . 0=なし, 1=あり .
Encryp	0	暗号化の有無 . 0=なし, 1=あり .
PinCod	1234	セキュリティPin コード .
Bonded	0	ボンディングの有無 . 0=無効, 1=有効
Rem	NONE SET	リモートアドレス .

表 5.2: 拡張設定表示

名称	値	意味
SrvName	SPP	サービス名 .
DevClass	1F00	デバイスクラス .
InqWindw	0100	デバイス検出可能時間 .
PagWindw	0100	デバイス接続可能時間 .
CfgTimer	60	設定可能時間 . 単位は秒 .
StatuStr	NULL	状態表示時拡張文字列 .

ボーレート設定

ボーレートを設定するためのコマンドを下記の書式で表す .

書式 : `SU,<1200 | 2400 | 4800 | 9600 | 19.2 | 38.4 | 57.6 | 115k | 230k | 460k | 921k >`

ここで、引数はボーレートを表しており、また、はじめの 2 文字だけでコマンドが受理される . 例えば、19.2[kbps] で通信をする場合、”SU,19” と入力する .

ビット長

送信時のビット長を設定する . 書式を下に示す .

書式 : `S7,<1 | 0>`

ビット長を 7 とする場合、S7 の引数を 1 とし、そうでなかったら 0 とする .

表 5.3: その他の設定の表示

名称	値	意味
Profile	SPP	プロファイル名。 SPP=シリアル通信, DCE=DUN DCE, DTE=DUN DTE, MDM=MDM.
CfgChar	\$	設定可能状態遷移文字.
SniffEna	0	SNIFF モード.
LowPower	0	低消費電力.
TX Power	0	送信時電力.
IOPorts	0	I/O ポート.
IOValues	0	I/O の値.
DebugMod	0	デバッグモードの有無.
RoleSwch	0	ロールスイッチの有無.

パリティビット

通信時のエラー検出のために用いるパリティビットを設定する。書式を下に示す。

書式: SL,<E | O | N>

ここで, "E" は Even, "O" は Odd, "N" は None を表す。

設定可能時間の設定

設定可能時間は, 下記に示す ST コマンドにより設定できる。

書式: ST,<num>

ここで, num は設定可能時間を表しており, また, 範囲は 0~255 となっている。ただし 0 と設定してはならない。理由は, 設定する時間がなくなってしまい, 今後は設定ができなくなってしまうためである。ある程度余裕を持った設定にすることが望ましいため, 10 以下にしないことを勧める。

省電力モード

Bluetooth モジュールには, 省電力化を図る手法がいくつか存在する。まず, SNIFF モードについて説明する。Bluetooth モジュールにはスリープ状態があるが, この状態になると極端に消費電力が低下する。SNIFF モードとは, 通常状態から一定時間ごとにスリープ状態にするものである。

書式: SW,<16 進数 >

上記の書式にある「16 進数」を例えば「10」, 10 進数で 16 とすると, $625 \times 16 = 10000$ [us] ごとにスリープ状態になる。ただし, スリープ状態を大きくしすぎると, 通信することができなくなってしまうため, 注意が必要である。

次に、デバイス検出可能時間の調整について説明する。Bluetooth によりデバイスを検出する (Inquiry Scan) 時間を少なくすることで、消費電力が抑えられる。

書式： `SI,<16進数>`

ここで、16進数に入れられる値は 0x1000 から 0x0000 であり、デフォルト値は 0x0200 となっている。なお、0x0000 とした場合にはデバイスの検出は行われない。

最後に、デバイス接続可能時間の調整について説明する。デバイスと接続する (Page Scan) 時間を少なくすることで消費電力が抑えられる。

書式： `SJ,<16進数>`

ここで、16進数に入れられる値は 0x1000 から 0x0000 であり、デフォルト値は 0x0200 となっている。なお、0x0000 とした場合にはデバイスの接続は行われない。

この教材では、以下のような設定とする。

```
SU,38
ST,20
SW,0010
SI,0200
SJ,0200
```

5.3 無線のシリアル通信

この節では、無線のシリアル通信を開始するまでの手順について述べる。

1. Bluetooth-USB アダプタをパソコンに取り付ける。
2. 図 5.12 に示した Bluetooth のインジケータをダブルクリックする。
3. MCU モード切替スイッチが”RUN” になっていることを確認し、Bluetooth モジュールを MCU 搭載ボードに取り付け、電源を入れる。
4. 図 5.18 に示した接続先をダブルクリックする。
5. KTM-02 の電源投入後、30 秒経過した後、リセットボタンを押す。なお、この 30 秒間は前節で設定した設定可能時間である。

以上の手順により、プログラムが開始され、あわせて無線のシリアル通信が可能となる。

第6章 各デバイスの調整方法

この章では、LCD および赤外線センサのように調整が必要であるデバイスの調整方法を記す。

6.1 LCD

LCD には、輝度調整機能が備わっており、具体的には図 1.7-[37] にある LCD 輝度調整用トリマにより調整できる。時計方向に回転すると文字が薄くなり、反時計方向に回転すると文字が濃くなる。ただし、今回用いているトリマは 1 回転すると元に戻ってしまうことに注意すること。

6.2 赤外線センサ

赤外線センサの近傍には赤外線 LED が備わっており、この明るさを変更することで感度を調整できる。赤外線 LED は、固定抵抗と図 1.1-[7] に示したトリマが直列接続されている。このトリマを時計方向に回転させると LED が明るくなり、反時計方向に回転させると暗くなる。感度の確認には、図 1.3-[22] に示した LED を用いる。なお、赤外線 LED、トリマ、確認用 LED はすべて同じ配列になっている。

第7章 開発手順

この章では、KTM-02の標準的な開発手順について説明する。全体の流れを図7.1に示す。一連の流れを大きく分けると、HDLによるCPLDへの回路開発、C/C++によるMCUへのプログラム開発の2つになる。すべての開発が終わった後、Bluetoothモジュールを取り付けてパソコンからの命令を無線で受け付けるようにする。以下では、上記2つの開発手順について概説する。

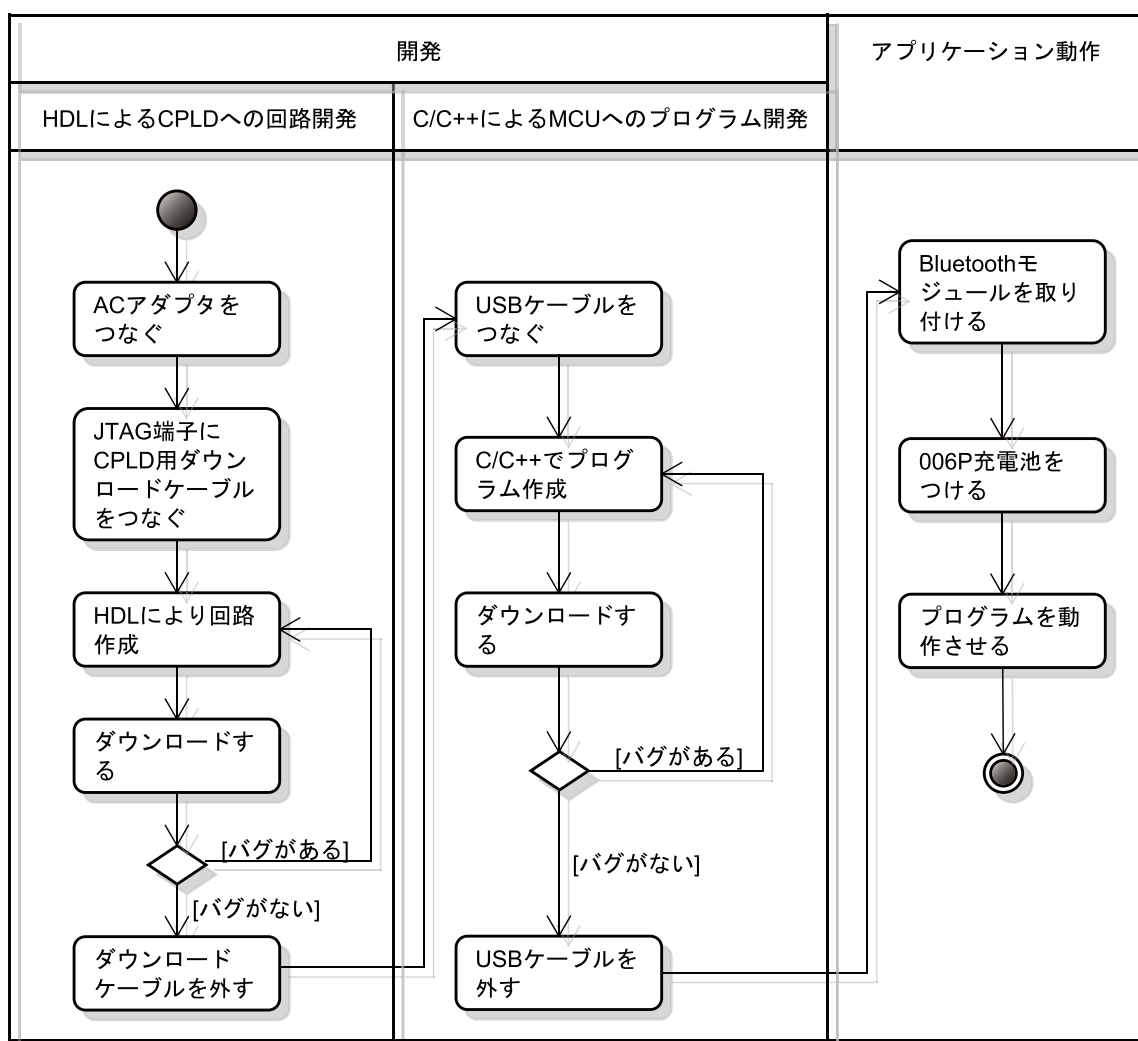


図 7.1: 開発時の流れ

7.1 HDL による CPLD への回路開発

KTM-02 にはベースボードとドットマトリクスディスプレイボードにそれぞれ CPLD があるが、ともにザイリンクス社製であるため、開発手順は同じである。はじめに、HDL により回路を作成するが、HDL を論理合成、配置配線およびダウンロードファイルを作成する。これらの処理を行うための開発ツールである ISE Web PACK を入手し、インストールする。ダウンロードファイルを CPLD にダウンロードするには、まず図 2.14 や図 2.15 のように、ザイリンクス社製ダウンロードケーブルを接続する。その後、ISE Web PACK に収録されている iMPACK を用いてダウンロードする。このツールはザイリンクスのホームページより無償でダウンロードできるので、入手してインストールする。

本書では HDL の解説を行わないが、回路図により各デバイスの CPLD への接続状況を把握できるので、参考にして回路を作成すること。なお、デフォルトでは下記の動作をする回路を CPLD に作成してある。

- 赤外線センサ通信用 CPLD(ベースボード)
 - 赤外線センサおよび赤外線 LED により路面状況を読み取り、その結果を MCU にクロック同期通信で送る。通信開始前に \overline{SS} 端子 (PA2 端子) をアサート (Low アクティブ) する必要があることに注意すること。
 - MCU の指示 (PA13 端子) により、赤外線センサ確認用 LED の点消灯をする。
- カメラ-距離センサ切替用およびドットマトリクスディスプレイ用 CPLD(ドットマトリクスディスプレイボード)
 - カメラと距離センサのいずれを使用するかを MCU の指示により選択する。
 - MCU と、カメラもしくは距離センサとの間を調歩同期通信でデータを送受信する。
 - MCU から送信されるドットマトリクスディスプレイのパターンコード (表 7.1) をデコードし、交差点表示用 LED によりパターンを表示する。また、パターンコードを送信する信号線について表 7.2 に示す。

表 7.1: ドットマトリクスディスプレイのパターンコード

コード	パターン	コード	パターン	コード	パターン	コード	パターン
0		1		2		3	
4		5		6		7	
8		9		10		11	
12		13		14		15	

表 7.2: ドットマトリクスディスプレイのパターンコードを送信する信号線

ビット	ビット3	ビット2	ビット1	ビット0
端子	PB2	PB1	PB5	PB3

7.2 C/C++によるMCUへのプログラム開発

SH-7125Fにプログラムするには、ルネサスエレクトロニクス社製の開発統合環境である High-performance Embedded Workshop(以下、HEW)を用いる。これを用いることで、アセンブリ、C および C++ 言語を用いるプログラミングおよびデバッグを統一的行える。プログラムを終了した後、アルファプロジェクト社製 Flash Writer EX for SH7125F(以下、FW)により、MCU にダウンロードする。

この章では、HEW によるプログラミング手順および FW によるダウンロード手順について説明する。

7.2.1 新規ワークスペースの作成

はじめに、HEW で用いられる言葉を説明する。HEW では、1つのアプリケーションを作成するときに用いられるソースやヘッダの集まりをプロジェクトという。さらに、いくつかのプロジェクトを集めたものをワークスペースという。1つのアプリケーションを作成するには少なくとも1つのワークスペースを作成する必要がある。

ワークスペースの作成方法

まず、HEW を起動する。スタートメニューなどから図 7.2 のアイコンを探し、ダブルクリックする。



図 7.2: HEW のアイコン

起動すると、図 7.3 に示すダイアログが表示され、ワークスペースを新規に作成するか、既存のものを開くか選択できる。今回は、新規にワークスペースを作成するので、「新規プロジェクトワークスペースの作成」を選択する。

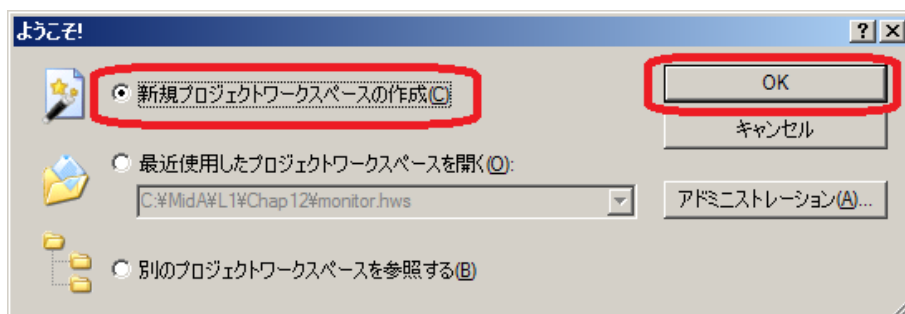


図 7.3: ワークスペースの選択

開発するプロジェクトの種類とワークスペース名の入力

今回はもっともオーソドックスなアプリケーションを作成するため、プロジェクトの種類については既存の選択でよい。また、ワークスペース名とプロジェクト名を異なるものにすることも可能であるが、ここでは同一とする。この場合、図 7.4 のようにワークスペース名を入力すると自動的にプロジェクト名も入力される。

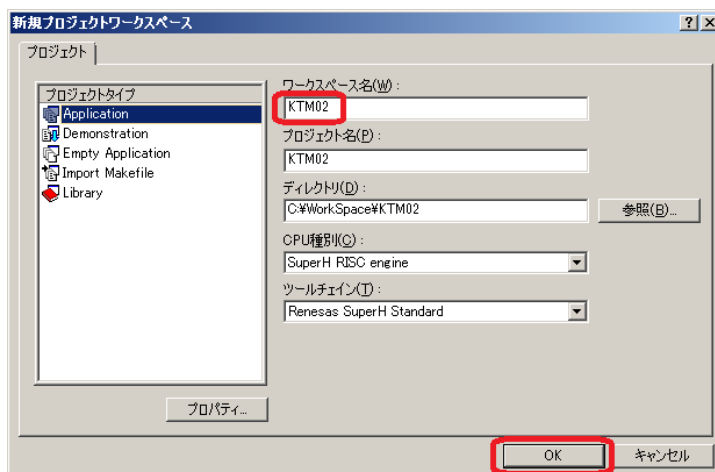


図 7.4: ワークスペース名の入力

MCU の種類を選択

HEW では、選択した MCU の周辺機能を構造体でアクセスできるヘッダファイル”iodefine.h”を生成する。図 7.5 に示すように、今回は CPU シリーズとして”SH-2”，CPU タイプとして”SH-7125”を選択する。

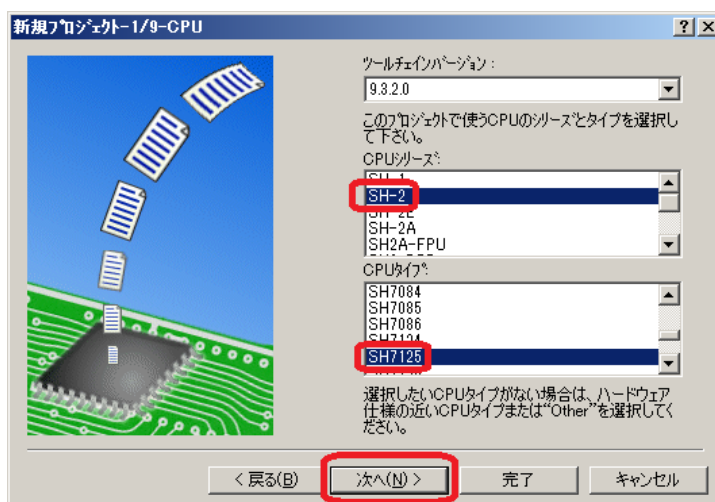


図 7.5: MCU の選択

オプションの設定

除算器の選択, エンディアンの選択, C++関連の設定などを行うが, 今回は図 7.6 のようにデフォルトのままでもよい. 必要に応じて選択すること.

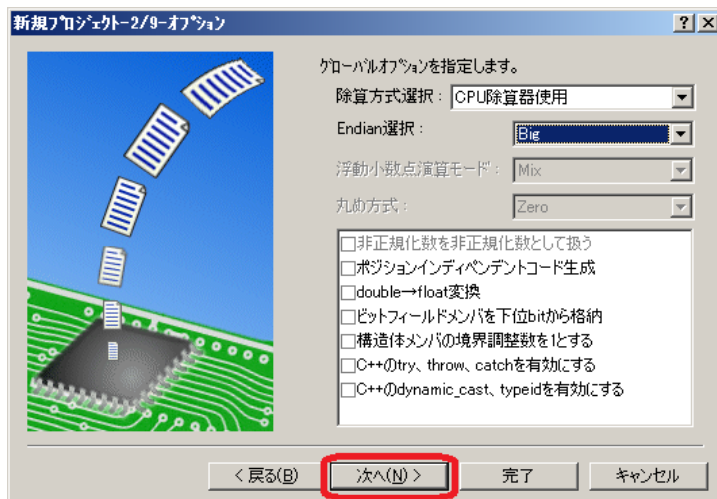


図 7.6: オプションの選択

イニシャルルーチンの設定

HEW で自動生成されるファイルには, main 関数が含まれるソースファイル, main 関数が呼ばれる前に実行される, ハードウェアセットアップ関数が含まれるソースファイルなどがある. これらのファイルを生成するために, ここでは設定を行う. 図 7.7 のように, 今回は main 関数を”C++ source file”, ハードウェアセットアップ関数生成を”C/C++ source file” とする.

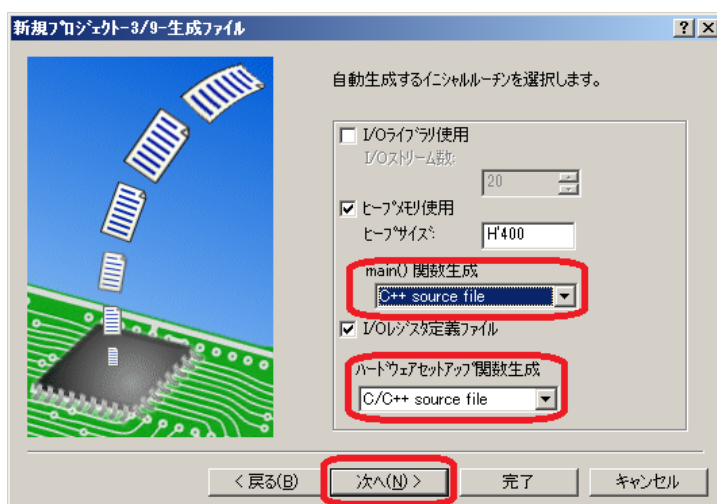


図 7.7: イニシャルルーチンの設定

使用するライブラリを選択

ここでは必要最低限のライブラリとするため、図 7.8 のようにデフォルトで選択されている”stdlib.h” および”string.h” をはずす。なお、必要とあらば後で選択しなおすことができる。



図 7.8: ライブラリの選択

スタックポインタの設定

ここでは図 7.9 のように、デフォルトのままの設定とする。



図 7.9: スタックポインタの設定

ベクタテーブルの設定

MCU に電源が入ったときおよびリセットがかかったときに呼び出されるプログラムをベクタテーブルに記述する。ここでは図 7.10 のように、デフォルトのままの設定とする。



図 7.10: ベクタテーブルの設定

デバッガの設定

HEW では、エミュレータによるデバッガを使ってレジスタの値やプログラムのステップ実行が行える。今回はデバッガがないので図 7.11 のようにデフォルトのままでもよい。



図 7.11: デバッガの設定

確認

図 7.12 では、生成されるファイルの確認をするダイアログである。HEW により、どのようなファイルが生成されるか一通り確認する。

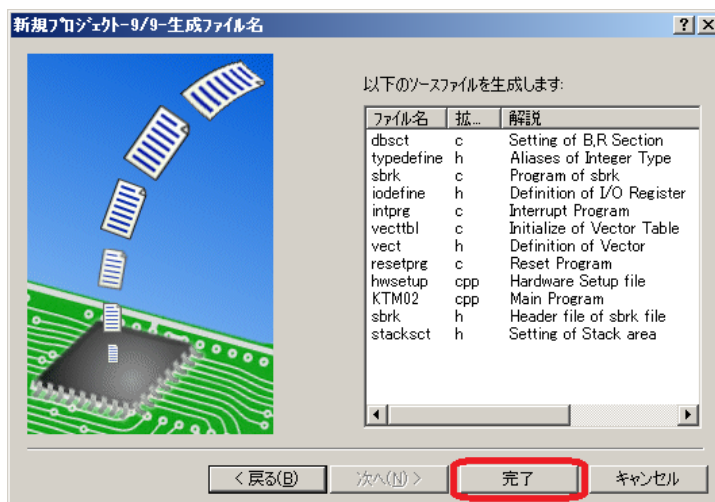


図 7.12: 確認

最終確認

図 7.13 のように、最終的な確認が行われる。設定項目の一覧が表示されるので、確認をする。

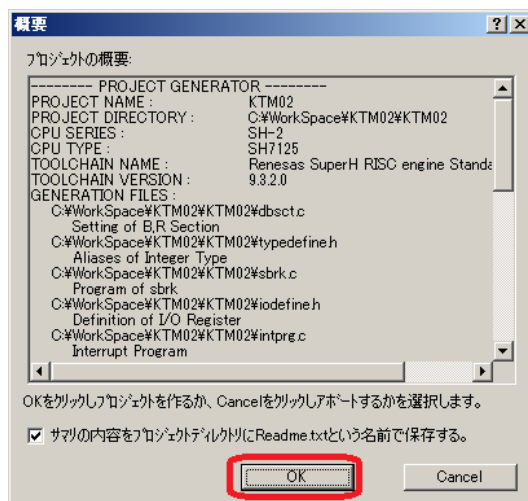


図 7.13: 最終確認

7.2.2 プログラミングとビルド

この節では、HEW によるプログラミングおよびビルドの方法について述べる。まずは図 7.14 を用いて HEW の用語を説明する。HEW には基本的な 3 つのウィンドウがあり、1 つは左端にあるワークスペースウィンドウである。これはワークスペースに帰属するソースファイルやヘッダファイルを階層的に表す領域である。2 つめはアウトプットウィンドウであり、これはコンパイル結果が出力される領域である。3 つめはプログラムエディタウィンドウであり、開かれたソースファイルなどをここで編集する。また、HEW の上部にはツールチップやメニューがあり、ビルドを行ったり、プロジェクトの設定を行ったりする。

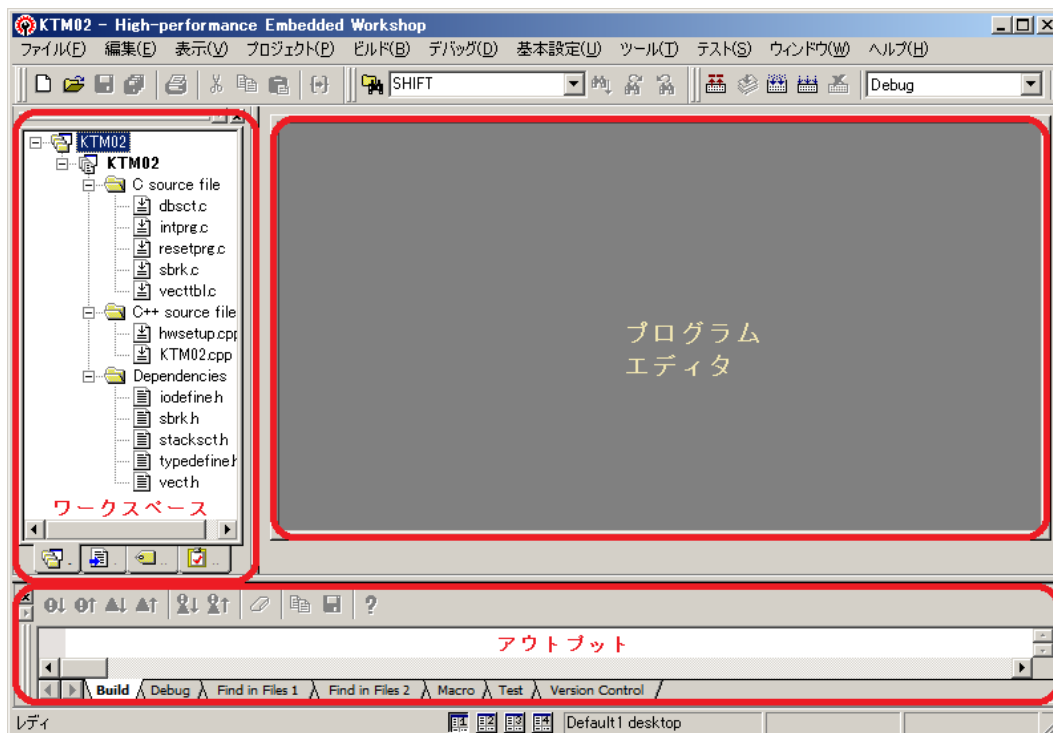


図 7.14: HEW の用語

次節では、プログラムの記述からビルドまで具体的に説明する。

プログラムの記述

まず、7.2.1 節で説明した手順により自動生成された main 関数をエディタで編集する。図 7.15 のように、ワークスペースウィンドウにある”KTM02.cpp” をダブルクリックするとこのファイルが開かれる。このファイルには main 関数があるので、ここに必要な処理を記述する。なお、main 関数が呼び出される前にハードウェアの設定を行いたい場合には、”hwsetup.cpp” にある HardwareSetup 関数に記述すること。

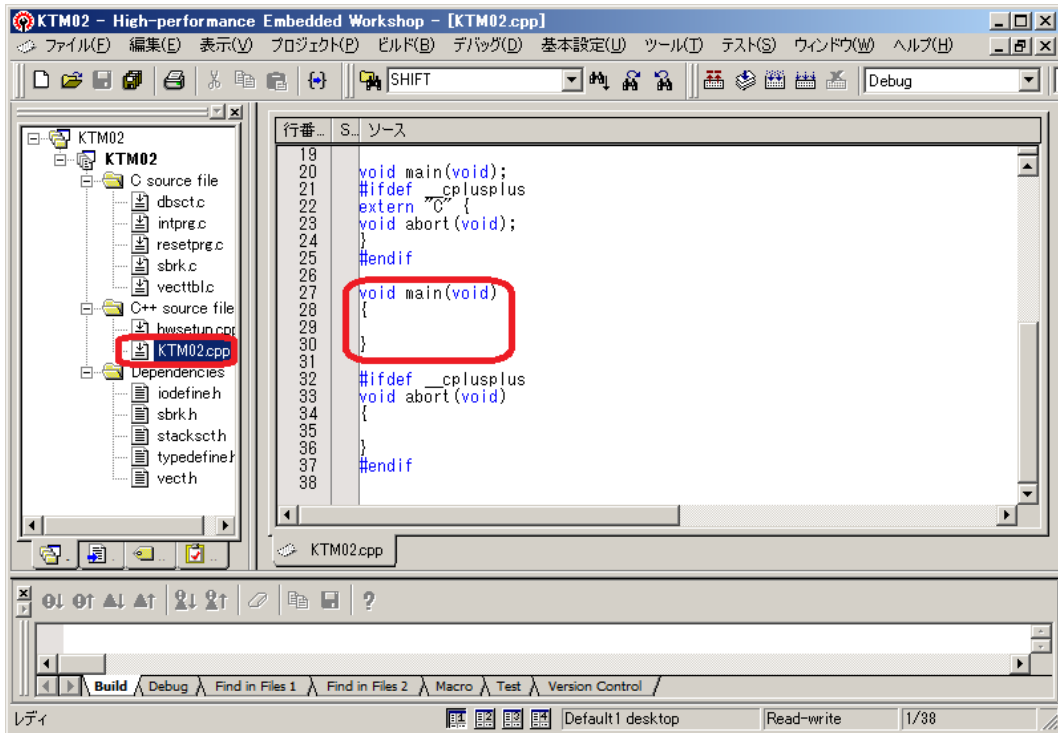


図 7.15: main 関数の編集

ビルド

プログラムを記述したらビルドを行う。図 7.16 のように、メニューにある「ビルド」の中の「ビルド」を選択するか、ショートカットである”F7”を押下するとビルドが開始される。ワークスペースを作って初めてのビルドをすると、ランタイムライブラリもビルドするため通常より時間がかかる。ビルドに成功すると、アウトプットウィンドウに”0 Errors, 0 Warnings” と表示される。

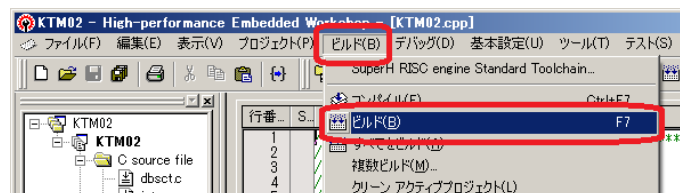


図 7.16: ビルド

7.2.3 ダウンロード

ビルドに成功すると、KTM02.mot ファイルが Debug フォルダ内に生成される。このファイルを Flash Writer EX for SH7125F により MCU ヘダウンロードする。

ケーブルおよびスイッチの準備

はじめに、MCU 搭載ボードにある MCU モード切替スイッチを図 7.17 のように Download 側にスライドさせる。

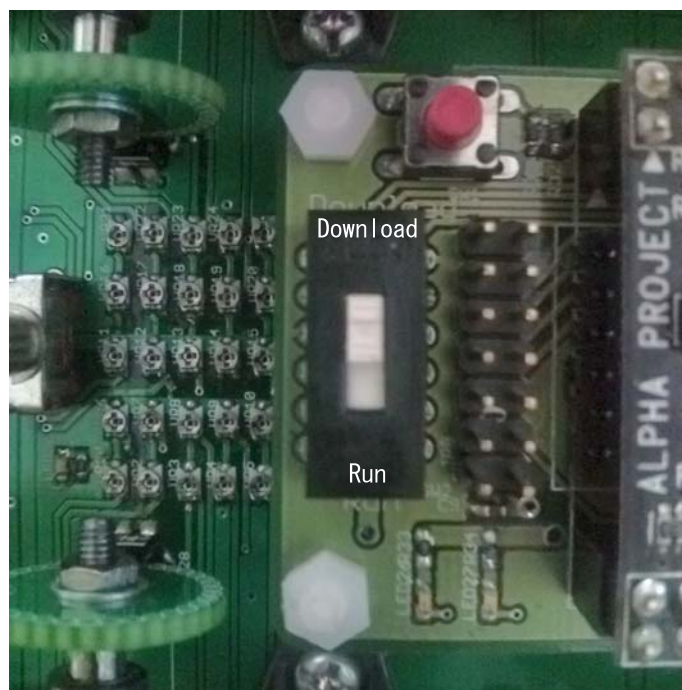


図 7.17: MCU モード切替スイッチ

次に、26 ページの図 2.16 に示したように、MCU 搭載ボードにある USB コネクタとパソコンを接続する。ここで、接続した USB ケーブルの先には USB-シリアル変換 IC があるため、パソコン側からは仮想 COM ポートが現れる。図 7.18 のように、デバイスマネージャを開き、COM 番号を確認しておく。

さらに、26 ページの図 2.17 のように AC アダプタを接続し、電源切替スイッチを 34 ページの図 4.2 のように AC アダプタを電源として使用する側にスイッチを倒し、MCU に電源を供給する。

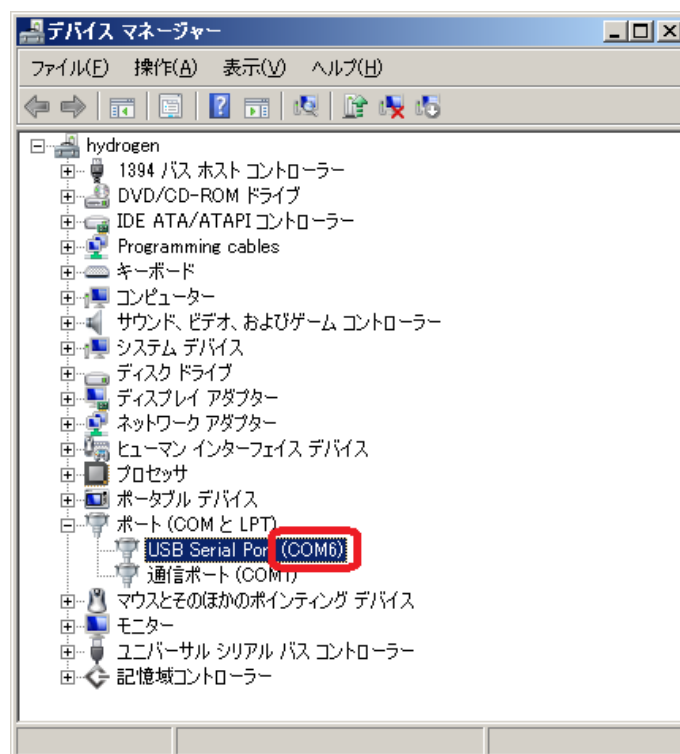


図 7.18: COM ポートの番号

FW の起動

スタートメニューなどから図 7.19 のアイコンを探し , FW を起動する .



図 7.19: FW のアイコン

COM ポートの設定

図 7.20 のように , 先ほど調べた COM ポートの番号を設定する .

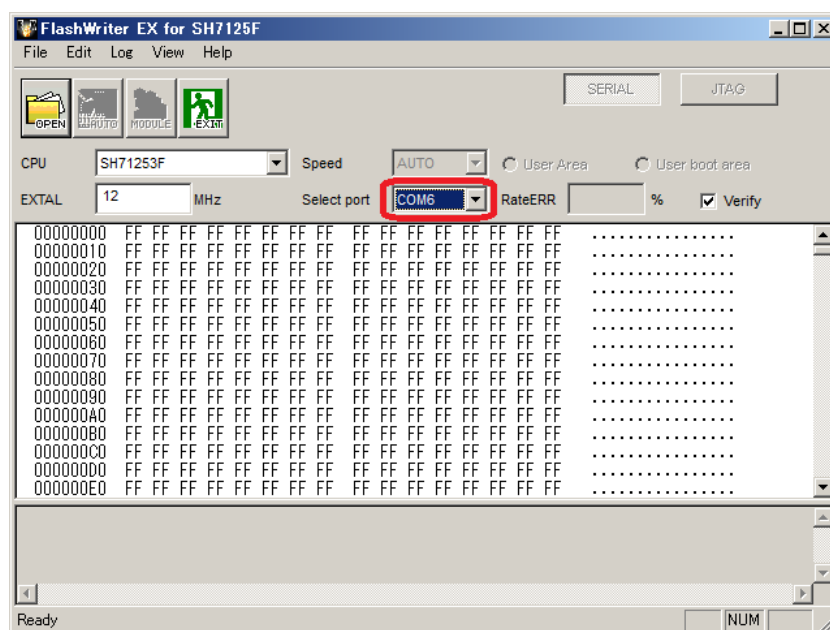


図 7.20: COM ポートの設定

ダウンロードファイルを開く

作成したダウンロードファイルを FW で開く。図 7.21 のように，“OPEN” ボタンを押すとファイルセレクタが表示されるので，下記のフォルダ内にある”KTM02.mot” を選択する。

C:\¥Workspace¥KTM02¥KTM02¥Debug¥KTM02.mot

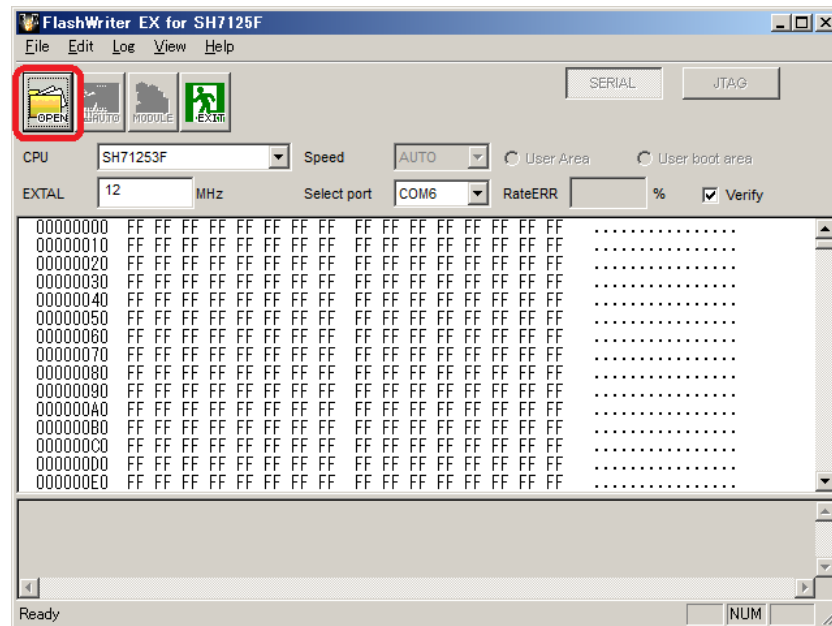


図 7.21: ダウンロードファイルを開く

ダウンロード開始

選択されたダウンロードファイルの内容を図 7.22 のように見ることができたら，“AUTO” ボタンを押す．次に，図 7.23 のダイアログが表示されるので，“START” ボタンを押す．そののちにダウンロードが開始され，正常に終了すると図 7.24 のダイアログが表示される．

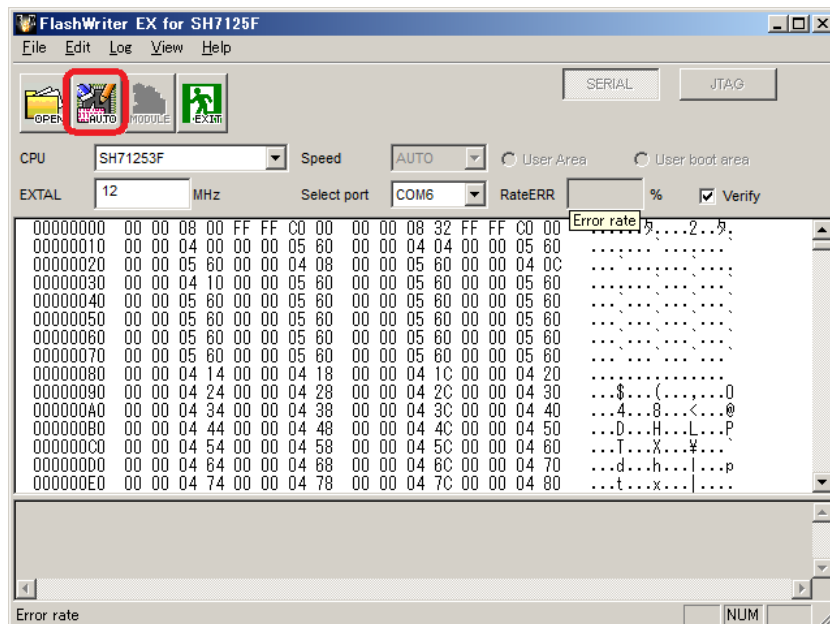


図 7.22: ダウンロード開始 1

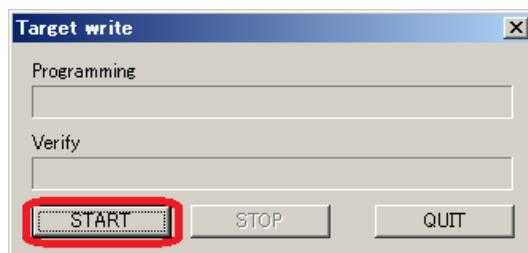


図 7.23: ダウンロード開始 2

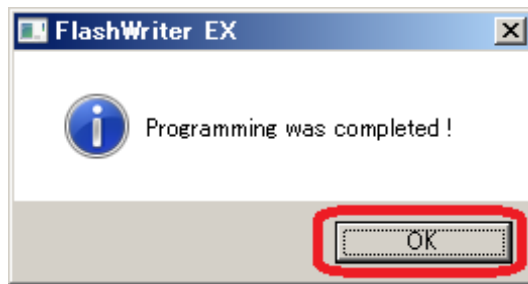


図 7.24: ダウンロード終了

ダウンロードの終了

ダウンロードが終了したら、電源切替スイッチを反対側に倒すか、ACアダプタを抜くことにより電源を切る。また、必要に応じてFWを終了する。

7.2.4 プログラムの開始

まずは、電源が切れていることを確認する。次に、MCUモード切替スイッチを”RUN”側にスライドさせる。そして、電源を入れるとプログラムが動作し始める。

第8章 端子の機能

はじめに各デバイスの略称を表 8.1 に示す．この略称を用いて各端子の機能などを次章で説明する．

表 8.1: デバイスの略称

略称	正式名称
モータ	タイヤを回転させるモータのドライバ．
カメラ/距離	ドットマトリクスディスプレイボードに接続されるカメラおよび距離センサおよびそれらを切り替えるセレクタ．
フォト	タイヤ回転数検知用フォトインタラプタ．
CPLD	赤外線センサ通信用 CPLD ．
センサモニタ	赤外線センサ確認用 LED ．
赤外線センサ	ベースボードの裏側になる赤外線センサ．
LCD/DMD	ドットマトリクスディスプレイボードに接続される LCD およびドットマトリクスディスプレイ．
パソコン	KTM-02 を操作するためのパソコン ．
ダウンロード	SH-7125F にプログラムをダウンロードするための端子 ．

8.1 ポート A

ポート A の端子を表 8.2～8.5 で説明する ．

表 8.2: PACRL4

端子名	ビット	機能	デバイス	説明
PA15	14-12	TXD1	パソコン	調歩同期通信でデータをパソコンへ送信するための端子 ．
PA14	10-8	RXD1	パソコン	調歩同期通信でデータをパソコンへ受信するための端子 ．
PA13	6-4	IO	センサモニタ	赤外線センサのモニタ用 LED の点灯/消灯するための端子 ．
PA12	2-0	SCK0	赤外線センサ	クロック同期通信を赤外線センサと行うとき用いるクロックを送るための端子 ．

表 8.3: PACRL3

端子名	ビット	機能	デバイス	説明
PA11	14-12	TXD0	赤外線センサ	クロック同期通信を赤外線センサと行うとき用いるデータを送信するための端子。
PA10	10-8	RXD0	赤外線センサ	クロック同期通信を赤外線センサと行うとき用いるデータを受信するための端子。
PA9	6-4	TXD2	カメラ/距離	調歩同期通信でデータをカメラもしくは距離センサへ送信するための端子。
PA8	2-0	RXD2	カメラ/距離	調歩同期通信でデータをカメラもしくは距離センサから受信するための端子。

表 8.4: PACRL2

端子名	ビット	機能	デバイス	説明
PA7	14-12	TCLKB	フォト	右タイヤの回転数を得るための端子。
PA6	10-8	TCLKA	フォト	左タイヤの回転数を得るための端子。
PA5	6-4	IO	LCD/DMD	LCD とドットマトリクスディスプレイ用 LED を切り替えるためのセレクト端子。
PA4	2-0	-	ダウンロード	プログラムをダウンロードするとき用いられている端子。

表 8.5: PACRL1

端子名	ビット	機能	デバイス	説明
PA3	14-12	-	ダウンロード	プログラムをダウンロードするとき用いられている端子。
PA2	10-8	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PA1	6-4	IO	カメラ/距離	カメラと距離センサを切り替えるためのセレクト端子。
PA0	2-0	IO	LCD	LCD の Enable 端子

8.2 ポート B

ポート B の端子を表 8.6~8.8 で説明する。

表 8.6: PBCRH1

端子名	ビット	機能	デバイス	説明
PB16	0	IO	LCD	LCD の RS 端子

表 8.7: PBCRL2

端子名	ビット	機能	デバイス	説明
PB5	6-4	IO	LCD/DMD	LCD のデータバス D5 端子もしくは ドットマトリクスディスプレイ用 LED のデータのビット 1 .

表 8.8: PBCRL1

端子名	ビット	機能	デバイス	説明
PB3	14-12	IO	LCD/DMD	LCD のデータバス D4 端子もしくは ドットマトリクスディスプレイ用 LED のデータのビット 0 .
PB2	10-8	IO	LCD/DMD	LCD のデータバス D7 端子もしくは ドットマトリクスディスプレイ用 LED のデータのビット 3 .
PB1	6-4	IO	LCD/DMD	LCD のデータバス D6 端子もしくは ドットマトリクスディスプレイ用 LED のデータのビット 2 .

8.3 ポート E

ポート E の端子を表 8.9~8.12 で説明する。

表 8.9: PECRL4

端子名	ビット	機能	デバイス	説明
PE15	14-12	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PE14	10-8	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PE13	5-4	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PE12	2-0	IO	CPLD	ベースボードにある CPLD につながっている補助端子。

表 8.10: PECRL3

端子名	ビット	機能	デバイス	説明
PE11	14-12	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PE10	10-8	IO	CPLD	ベースボードにある CPLD につながっている補助端子。
PE9	6-4	IO	フォト	左タイヤのスピード数を得るための端子。
PE8	2-0	IO	フォト	右タイヤのスピード数を得るための端子。

表 8.11: PECRL2

端子名	ビット	機能	デバイス	説明
PE7	14-12	IO	モータ	右モータの ENABLE 端子。
PE6	10-8	IO	モータ	右モータの IN 端子。
PE5	6-4	IO	モータ	右モータの BRAKE 端子。
PE4	2-0	MTU2	モータ	右モータの CHOP 端子。

8.4 ポート F

PF0 は、距離センサから出力されている距離を表すアナログ信号を受け取っている。距離センサは、基本的に調歩同期通信により MCU へ距離を送信しているが、カメラと同時に使用したいときには、この PF0 から得られたアナログ値をデジタル値に変換して距離を得ることも可能である。

表 8.12: PECRL1

端子名	ビット	機能	デバイス	説明
PE3	14-12	IO	モータ	左モータの ENABLE 端子 .
PE2	10-8	IO	モータ	左モータの IN 端子 .
PE1	6-4	IO	モータ	左モータの BRAKE 端子 .
PE0	2-0	MTU2	モータ	左モータの CHOP 端子 .

付録A 部品リスト

A.1 ベースボード

表 A.1: ベースボード上の部品リスト

記号	部品名	型番・サイズ	値	メーカ
C1~32	積層セラコン	1608	0.1[μ F]	
EC1,2	タンタルコン	3528	10[μ F]	AVX
IC1	CPLD	XC95108-TQ100		XILINX
J1	DC ジャック		内径=2.1[mm]	
JP1	ピンソケット	0.1[inch] ピッチ	2×20	
JP2,3	ピンソケット	0.1[inch] ピッチ	2×7	
JP4,5	L字ピンヘッダ	0.1[inch] ピッチ	1×3	
JP6,7	L字ピンヘッダ	0.1[inch] ピッチ	1×2	
LED1~25	赤外線 LED	SFH 4258	λ =850[nm]	OSRAM
LED26	LED	1608	黄色	
PWR1~3	006P 電池ボックス			
R1~25	抵抗器	1608	2.7[k Ω]	
R26,28	抵抗器	1608	1[k Ω]	
R27,29	抵抗器	1608	10[k Ω]	
R30	抵抗器	1608	1.5[k Ω]	
VR1~25	トリマ	TMC2KJ-B470	470[Ω]	帝国通信工業
REG1	レギュレータ	LM1085IS-5.0	5.0[V]	NI
SW1	トグルスイッチ		単極双投	COSLAND
U1~25	赤外線センサ	S10053		浜松ホトニクス
U26,27	フォトインタラプタ	KRB031		Kingbright

A.2 MCU 搭載ボード

表 A.2: MCU 搭載ボード上の部品リスト

記号	部品名	型番・サイズ	値	メーカー
C1~3,5,9	積層セラコン	1608	0.1[μ F]	
C4	積層セラコン	2012	1[μ F]	
C6	タンタルコン	3216	4.7[μ F]	Vishy
C7,8	電解コンデンサ	面実装	100[μ F]	ニチコン
C10	積層セラコン	1608	0.01[μ F]	
EC1,2	タンタルコン	3216	10[μ F]	AVX
CN1	USB コネクタ	mini-B		
D1~4	ショットキバリアダイオード	D2S6M	60[V]2[A]	SHINDENGEN
JP1,2,6,7	L 字ピンヘッダ	0.1[inch] ピッチ	1×2	
JP3	ピンヘッダ	0.1[inch] ピッチ	2×20	
JP4,5	ピンヘッダ	0.1[inch] ピッチ	2×7	
JP8	ピンヘッダ	0.1[inch] ピッチ	2×7	
L1	フェライトビーズ	2012	40[m Ω]	
LED1~27	LED	1608	黄色	
MODULE1	MCU ボード	STK-7125		Alpha Project
R1~25,34,35	抵抗器	1608	1.5[k Ω]	
R26~30,32,35,36	抵抗器	1608	10[k Ω]	
R31	抵抗器	1608	100[k Ω]	
REG1	レギュレータ	LM1085IS-3.3	3.3[V]	NI
U1,2	モータドライバ	TA8440H		東芝
U3	Bluetooth モジュール	WRL-00582		ROVING NETWORK
U4	USB-シリアル変換	FT232R		FTDI
SW1	タクトスイッチ		単極単投	
SW2	スライドスイッチ	ASE4204	4 極双投	tyco

A.3 ドットマトリクスディスプレイボード

表 A.3: ドットマトリクスディスプレイボード上の部品リスト

記号	部品名	型番・サイズ	値	メーカー
C1~4,8	積層セラコン	1608	0.1[μ F]	
C5	積層セラコン	1608	0.33[μ F]	
C6	積層セラコン	3216	2.2[μ F]	
C7	電解コンデンサ	面実装	100[μ F]	ニチコン
IC1	CPLD	XC9536-VQ44		XLINX
JP1	ピンヘッダ	0.1[inch] ピッチ	1×6	
JP2	ピンヘッダ	0.1[inch] ピッチ	2×7	
JP3	ピンヘッダ	0.1[inch] ピッチ	1×4	
JP4	ピンソケット	0.1[inch] ピッチ	1×7	
JP5	ピンソケット	0.1[inch] ピッチ	2×7	
LED1~9	LED	HSML-A101-S00J1	オレンジ	
R1~9	抵抗器	1608	2[k Ω]	
R10~13	抵抗器	1608	10[k Ω]	
REG1	レギュレータ	NJM2845DL	3.3[V]	
U1	レベルシフタ	TXB0104	3.3[V] 5.0[V]	TI
VR1	トリマ	TMC2KJ-B22K	22[k Ω]	帝国通信工業
X1	発振器	DOC-49S	1.3[MHz]	大真空

A.4 電源ボード

表 A.4: 電源ボード上の部品リスト

記号	部品名	型番・サイズ	値	メーカー
PWR1~4	006P 電池ボックス			
JP1,2	L字ピンヘッダ	0.1[inch] ピッチ	1×2	

付録B 回路図

KTM-02 の回路図を次ページ以降に示す．表 B.1 に示すページにそれぞれの回路図を表す．

表 B.1: 回路図一覧

基板名	ページ
ベースボード	77, 78
MCU 搭載ボード	79, 80
ドットマトリクスディスプレイボード	81
電源ボード	82

