

ARM マイコンを用いたロボット開発の基礎講座

1. はじめに

本稿では、「ロボットの作り方 2012」にて製作する鉄棒ロボットの製作方法について、記述してものです。製作する鉄棒ロボットは、体を構成する二つの基板をつないだ能動関節と鉄棒周りの回転の受動関節を持ち、モータを使って能動関節を起動します。CORTEX-M3 コアのマイコンを搭載した CPU ボードに、モータ制御回路・シリアルレベルコンバータ・LED 転倒回路など豊富な周辺機能を持っている STM32VLDISCOVERY を使用しています。

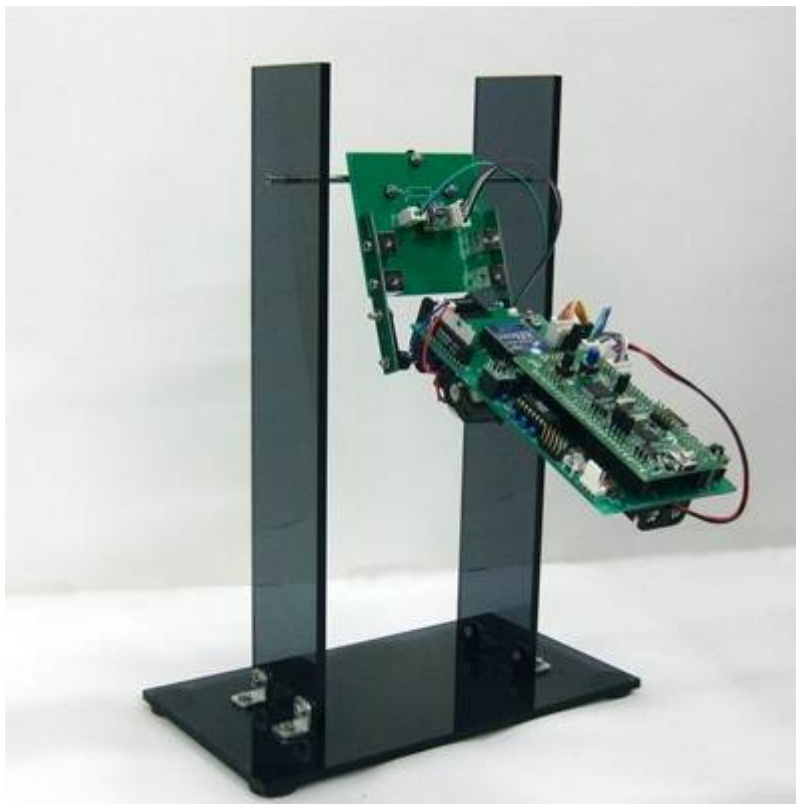


図 1 鉄棒ロボット

2. ARM (CORTEX-M3) マイコン

CORTEX-M3 は、ARM 社が 2005 年に発表した CPU ファミリーの中に属する、CORTEX-M シリーズのうち、主に 16/32bit アプリケーションをターゲットとして開発されたモデルです。サービスロボットの制御に使用されるほかの有名な CPU で言えば、H8 や SH2 などが対象とするエリアです。その他に、32bit/DSP アプリケーション用の CORTEX-M4、8bit/16bit アプリケーション向けの CORTEX-M0 などがあります。

ARM シリーズは、近年は、Android 携帯に使用されて有名になっていますが、古くはゲームボーイなどにも、ARM7TDMI という CORTEX-M3 の前身のモデルが採用されていまし

た. ARM 社自体は, ARM を搭載した IC を製造・販売していません. ARM 社は単純に言うてしまうと CPU の設計データを販売している会社です. マイコンとして動かすために必要な周辺機能を載せた IC を製造するのは, ARM 社から CPU の設計データを使用することを許可された企業です. 本キットで使用する STM32VLDISCOVERY には, ST Micro semiconductor 社 (以下 STM 社) の STM32F100RBT6B という IC が搭載されています.

3. STM32VLDISCOVERY

STM32VLDISCOVERY は, STM 社のマイコン評価ボードである DISCOVERY シリーズの STM32 Value Line に属するマイコンを搭載した評価ボードです. STM32 Value Line は, STM32 F1 に属する複数の製品シリーズのうち, クロック周波数が 24MHz でモータ制御や家電製品の制御をターゲットとしたシリーズで, STM32 F1 の入門的な位置づけになります. STM32VLDISCOVERY は, プログラムの対象となるメインの CPU に STM32F100RB を搭載し, STM 社のマイコン用デバッグインターフェースの ST-Link が接続された構成になっています. ST-Link 自体は, 単体でほかのボードをデバッグするためにも使用可能です. また, 評価ボードですので, いくつかの LED とボタンが搭載されています. 以下のような特徴を持っています.

- STM32F100RBT6B, 128 KB Flash, 8 KB RAM
- ST-Link(STM 社のデバッグインターフェース)
- 2つの赤色 LED; LD1(USB 通信), LD2(電源ランプ)
- 電源: USB バスパワー、外部 5V、3.3V のいずれか
- ユーザー制御 LED(LD3(緑)・LD4(青))
- 2つのプッシュボタン(User・Reset)

図 2 が STM32VLDISCOVERY のパッケージ外観です. この STM32VLDISCOVERY をコアにモータや追加の LED などを搭載した基板などを組み立てると本セミナーで製作する鉄棒ロボット(図 1)が完成します. 以下に、その作り方を解説します.



図 2 STM32VLDISCOVERY

4. 4速パワーギアボックスHE

本キットで使用しているギアボックスは、タミヤ社の4速パワーギアボックスHEです(図3)。このギアボックスは、歯車の構成を変えることで4つのギア比を選ぶことができます。個々では、ギア比として、47.6:1を選択して組み立てます(図4、図5)。組み立て方の詳細は、4速パワーギアボックスHEに含まれる説明書をご覧ください。

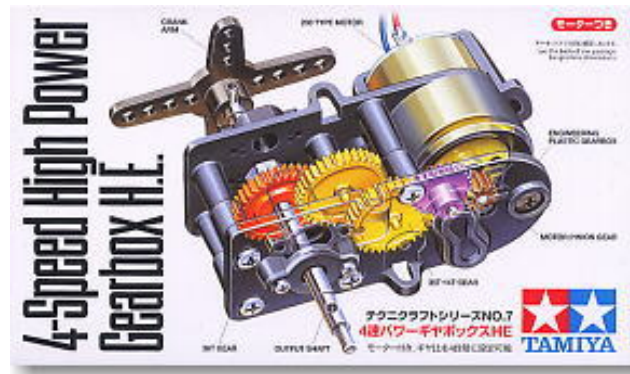


図 3 4速パワーギアボックスHE

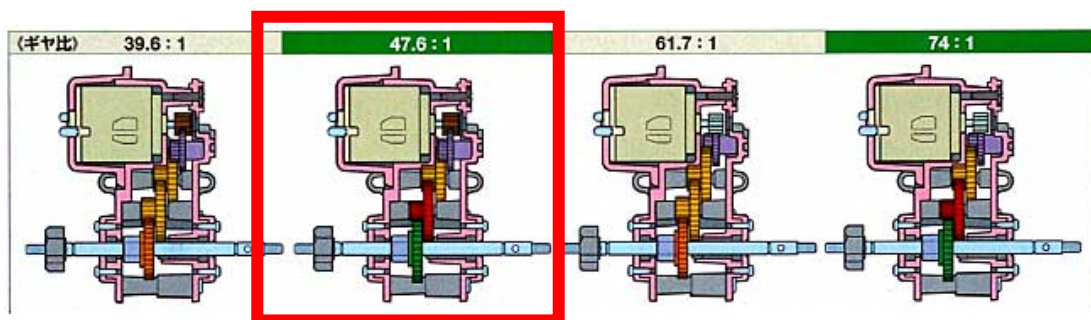


図 4 4速パワーギアボックスHEのギアの組み合わせ



図 5 組み立てた4速パワーギアボックスHE

5. エンコーダの組み立て

関節角度を測定するために本キットでは、光学式エンコーダを使用します。光学式エンコーダは、LED と受光素子の間に、光が通るスリットが入った円盤をはさんで、スリットを通して入ってきた光の数で何度回転したかを計測するセンサです。本キットでは、エンコーダとして、コーデンシ社の KE203 と図 6 に示すスリット板を使用して光学式エンコーダを構成します。スリット板を軸の回転と一緒に回転させるため、スリット板とギアボックスの円形アームを接着します。まず、はじめに、図 7 のフランジ付きスペーサを、ギアボックスの使用していないシャフトに通します。次に、スリット板に、フランジ付きスペーサを接着します。さらに、スペーサをギアボックスの円形アームのボス側に接着します(図 8)。できあがった様子を図 9 に示します。

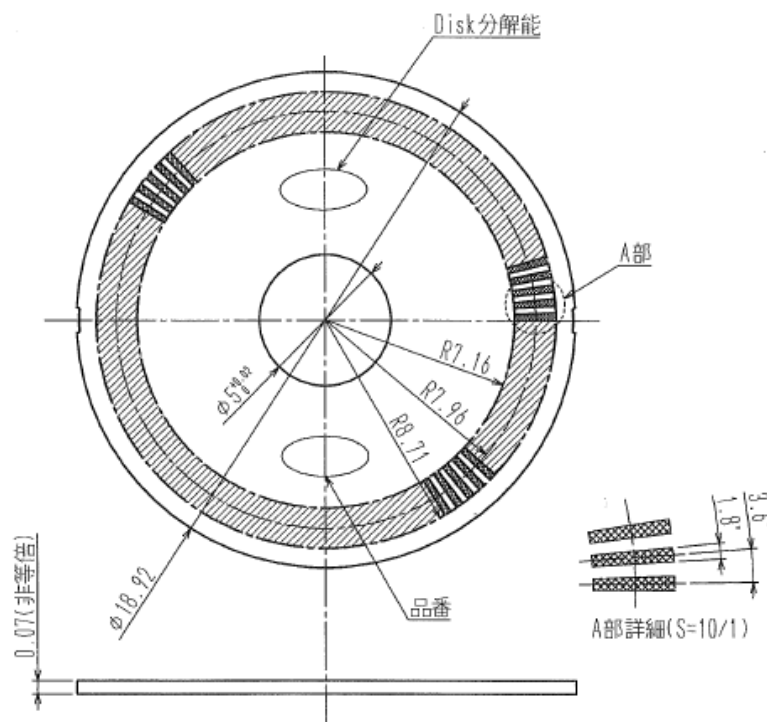


図 6 スリット板の仕様



図 7 フランジ付きスペーサ

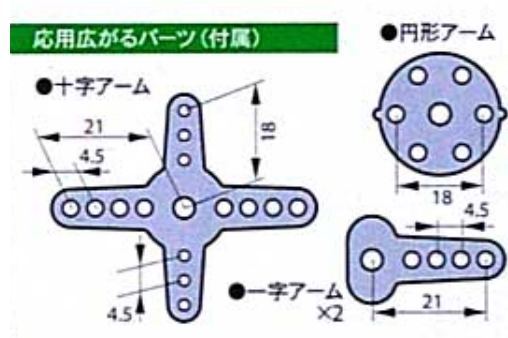


図 8 ギアボックスのアーム

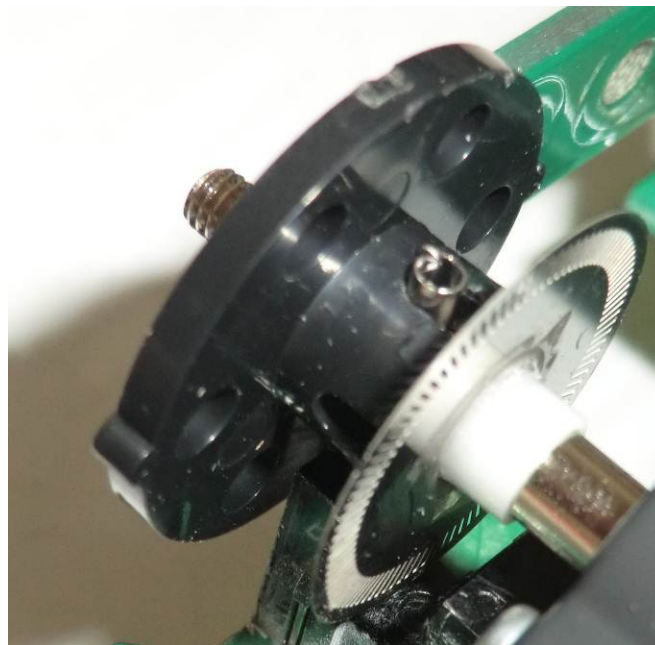


図 9 エンコーダ用スリット板

6. 基板の半田付け

基板の半田付けの前に部品の準備をします。STM32VLDISCOVERYのピンヘッダは、28ピン x 2、6ピン x 1 という構成なので、1x42のピンソケットを切って、1x28ピンを2本、1x6ピンを1本作ります(図10, 図11)。また、ジャイロセンサを取り付けるためにも、ピンソケットを使用しますので、1x4ピンを2本作ります(図12)。

基板は、図13のように、Vカットが入った状態で複数の基板がつながっていますので、それぞれの基板に折り分けます。Board_AがCPUなどを載せるメインの基板、Board_Bがエンコーダ用の基板、Board_Cがジャイロセンサ・リードスイッチなどを載せる基板、Board_E、Board_Fが、Board_A、Board_Cをつなぐ腕になる基板です。Board_Dは、オプションのエンコーダを取り付ける基板ですが、今回は使用しません。



図 10 28ピン x2



図 11 6ピン x1



図 12 4ピン x2

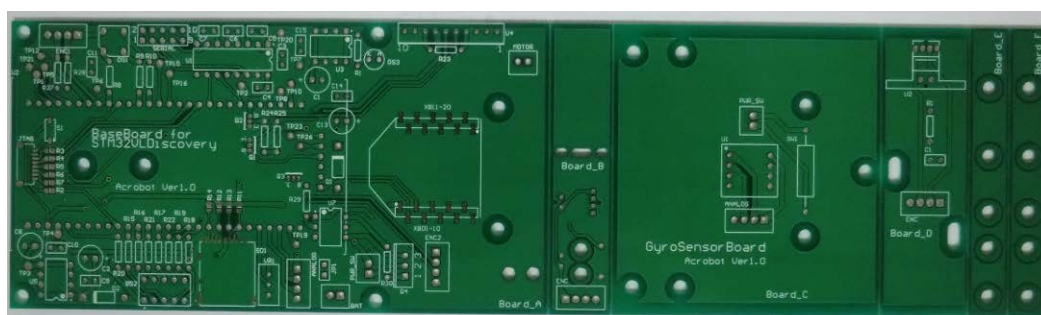


図 13 集合基板(写真は開発中のもの)

基板の上には、半田付けする部品を示した番号が印刷されていますので、その番号に、部品表にある部品をさして、半田付けしてください。部品をさす順番は、背の低い部品(抵抗やダイオード)から順にさすようにすると作業がしやすくなります。JP1には、ピンヘッダをはんだ付けします。なお、Board_B上のENCコネクタは、U6、R26、C12と同じ面(ENCと印刷されている面の反対側)に取り付けます。

Board_AおよびBoard_Bの部品表を表1に示します。また、Board_CおよびBoard_D

の部品表を表 2 に示します。表中では「実装しない」と表記されている部品は、本セミナーで使用しないので今回は実装しません。

表 1 部品表(Board_A、Board_B)

部品番号	メーカー	製品名	数	値	実装の有無
ANALOG, ENC, ENC1	MOLEX	5045-04A	3	4ピン	
C1, C2, C8, C13			4	100uF	
C3, C4, C5, C6, C7, C9, C11, C12, C14, C15			10	0.1uF	
D1, D2, D3	Panjit	1S3	3	1.0A 0.5V	
DS1	OptoSupply	OSTA71A1D-A	1	角形フルカラー	
DS2	Sharp	GL9A040G	1	7セグメント	
DS3	OptoSupply	OSHR5111A-TU	1	赤	
JP1		ピンヘッド	1	2ピン	
L1	太陽誘電株式会社	LHL08NB470K	1	47uH	
PWR_SW	MOLEX	5045-02A	1	2ピン	
Q1, Q2, Q3, Q5, Q6, Q7	Toshiba	RN1201 or RN1202	6		
Q4	Fairchild	FQPF14N30	1	Nch MOSFET	
R1, R8, R9, R10, R15, R16, R17, R18, R19, R20, R21, R22, R29			13	1K	
R23, R24, R25			3	10K	
R26			1	470	
R27, R28			2	3.3K	
R30			1	10	
SERIAL	MOLEX	5045-05A	1	5ピン	
U1	Intersil	ICL3232CPZ	1		
U2	STM	STM32VLDISCOVERY	1		
U4	東芝	TA7291P	1	1A(ave) 2A(peak)	
U5	HOLTEK	Ht7750A	1	5V	
U6	KODENSHI	KE203	1		
U7	Toshiba	TLP191B	1	7V 24uA	
VR1	Bourns	3296W	1	10k	
ENC2	MOLEX	5045-04A	1		実装しない
JTAG	日本圧着端子製造	BM08B-SRSS-TB	1	10ピン	実装しない
R11, R12, R13, R14			4	4.7K	実装しない
R2, R3, R4, R5, R6, R7		RMC1/16S270F	6	27Ω	実装しない
R31, R32			2	3.3K	実装しない
S1	日本電産コパル	CHS-01TA	1		実装しない
SD1	ヒロセ	DM3AT-SF-PEJ	1		実装しない
U3	秋月電子	STEP-UP_DIP_5.0V	1	3.3V 0.2A	実装しない
XB01-10, XB11-20			2		実装しない

表 2 部品表(Board_C、Board_D)

部品番号	メーカー	製品名	数	値	実装の有無
U1	秋月電子	K-04912	1		
ANALOG	MOLEX	5045-04A	1		
PWR_SW	MOLEX	5045-02A	1		
SW1	NECTーキン	RD-7B	1		
C1			1		実装しない
U2	KODENSHI	KE203	1		実装しない
ENC	MOLEX	5045-04A	1		実装しない
R1			1		実装しない

必要な部品を半田付けしたら、電池ボックスを取り付けます。電池ボックスと基板の間に、高さ 5mm のスペーサーを入れて、M3 のサラねじで、電池ボックスを取り付けます (図 14)。

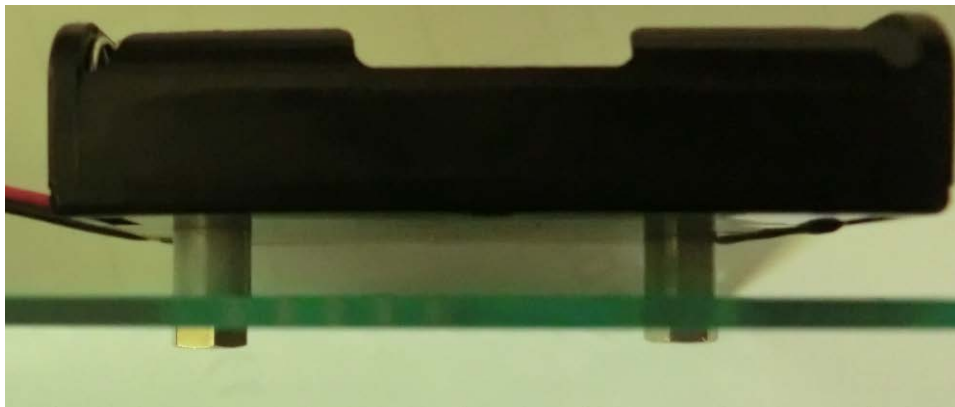


図 14 電池ボックスの取り付け部分

電池ボックスから出てる赤・黒の線は、基板上に BAT と書かれたエリアの四角いパッドと丸いパッドにそれぞれ刺して半田付けします。四角いパッドが+になるので赤い線を、丸いパッドが-になるので黒い線をはんだ付けします (図 15)。

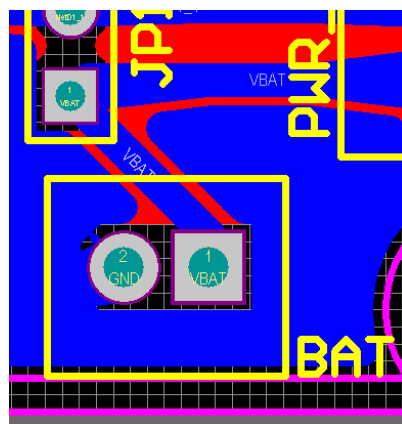


図 15 電池端子

組み立てたギアボックスを基板 Board_A の裏面に長さ 10mm の M3 ナベねじを使用して、基板上 L と書かれた穴を使用して取り付けてください。モータの端子は U4 右下の基板上 MOTOR と書かれたエリアの四角いパッドにモータから出ている赤い配線を、丸いパッドに青い配線を半田付けします (図 16)。

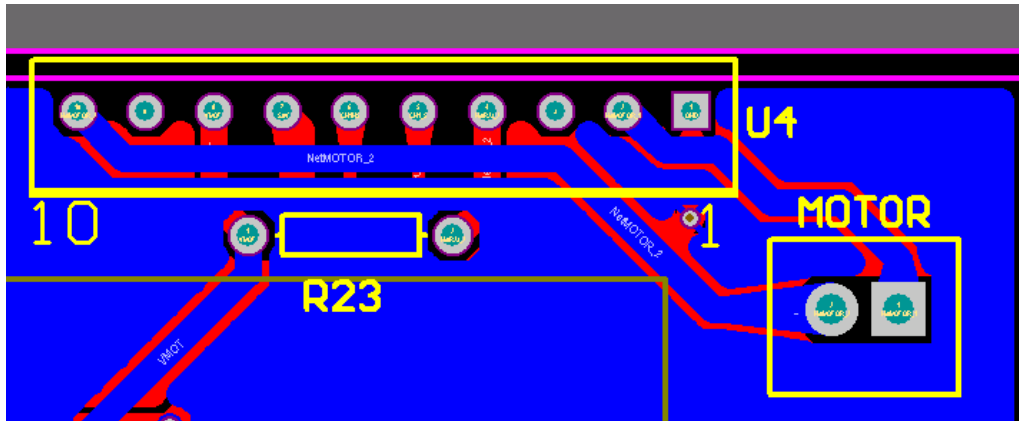


図 16 モータ端子

ギアボックスを Board_A に取り付けられた後に、Board_B を Board_A 上に取り付けます。まず、先ほど、スリット板を接着した円形アームを、ギアボックスの出力軸に取り付けます。その際、上下の基板をつなぐアーム(Board_F)と穴の位置を合わせてください(図 17)。アームの取り付け方については、ギアボックスの説明書をご覧ください。割ピンをいれてアームを固定したら、スリット板がエンコーダの中央の隙間に入るように、Board_B を取り付けます(図 18)。Board_A と Board_B の間には、高さ 5mm のスペーサを間に入れて、ねじで固定します。

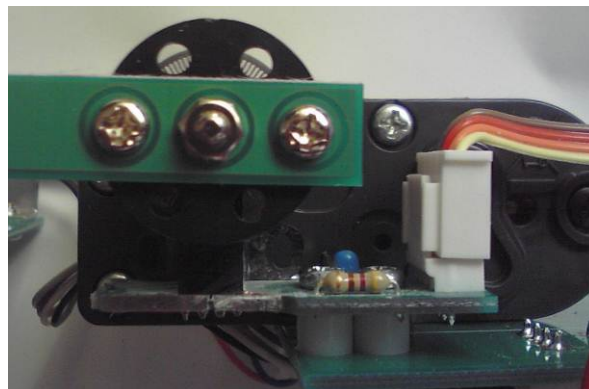


図 17 Board_B の取り付け方

Board_C 上には、ジャイロセンサとリードスイッチを取り付けます。ジャイロセンサ基板は、ジャイロセンサに付属のピンヘッドをはんだ付けします。ジャイロセンサは、1 番ピンが基板上の印刷にある点の位置にくる向きに、基板表面に取り付けます。リードスイッ

チを半田付けする前に、軸受けを取り付けます。リードスイッチは、図 19 のように、リードスイッチ本体が、鉄棒の近くに来るように、ガラス管を破損しないように、注意して足を曲げて、穴に半田付けします。

Board_C の上に L 字金具をねじ止めします。L 字金具の短い方を Board_C 上に固定し、Board_E、Board_F を長い方に取り付けます(図 20)。Board_A、Board_B、Board_C、Board_E、Board_F を図 21 のように接続します。付属のケーブルで Board_A 上の ENC1 と Board_B 上の、Board_C メイン基板と、ジャイロ基板本体の完成です。

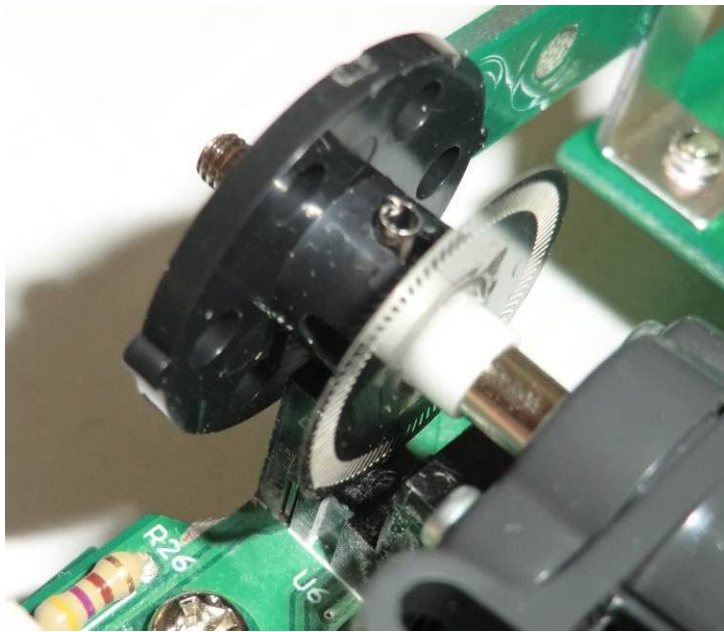


図 18 Board_B とスリット板の位置関係



図 19 Board_C 上の軸受けとリードスイッチの位置関係



図 20 Board_C と Board_E、Board_F との取り付け方

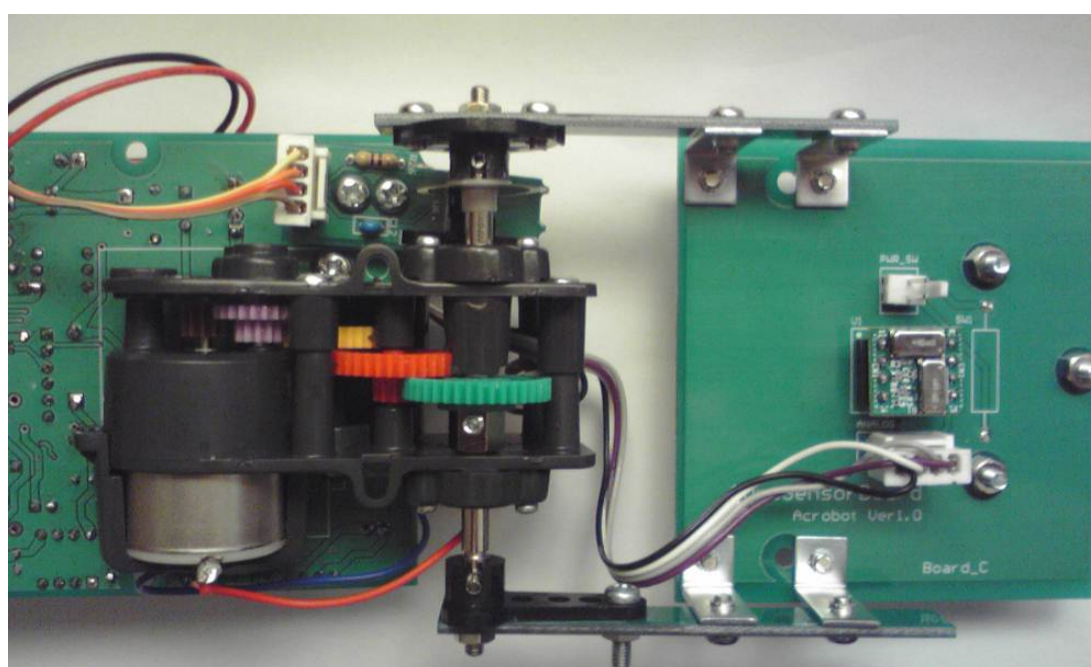


図 21 Board_A、Board_B、Board_C、Board_E、Board_F を接続した状態

図 19 の軸受けに鉄棒を通します。軸受けの左右に、ワッシャ・ゴムブッシュを通して、本体が鉄棒を左右にずれないようにします。アクリル板と L 字金具で土台を組み立て、先ほどの鉄棒を通して、アクリル板の外側からゴムブッシュで鉄棒が抜けないようにして、全体が完成します。

7. ソフトウェア開発

STM32VLDISCOVERY に搭載されている STM32 シリーズのマイコンは、様々な周辺機器を内蔵しています。本章では、そのうちロボットの制御に使用する主要な周辺機器を使用したプログラムについて解説します。

まず、STM32VLDISCOVERY の基本的な使い方を学ぶために、STM 社の WEB ページから、STM32VLDISCOVERY のサンプルファームウェアのパッケージをダウンロードし

ます。次の URL(図 22)を開き、Design Support を選択します。

<http://www.st.com/internet/evalboard/product/250863.jsp>

ページ下部にある、[Software & Development Tools]の欄から、STM32VLDISCOVERY firmware package (AN3268)をダウンロードします(図 23)。

http://www.st.com/internet/com/SOFTWARE_RESOURCES/SW_COMPONENT/FIRMWARE/stm32vldiscovery_package.zip

stm32vldiscovery_package.zip を展開すると、an3268 とディレクトリができるので、その中の[stm32vldiscovery_package]ディレクトリを、開発するプログラムを入れるディレクトリ(たとえば C:\STM32_WORKSPACE など)にコピーします。[stm32vldiscovery_package]ディレクトリ中の次のファイルを EWARM 用のワークスペースファイルで開きます(図 24)。

stm32vldiscovery_package¥Project¥Master
Workspace¥EWARMv5¥Value_Line_Discovery.eww

The screenshot shows the product page for STM32VLDISCOVERY. The page has a blue header with the STMicroelectronics logo and navigation links. Below the header, there is a breadcrumb trail: Home > Tools & Resources > Evaluation Tools > Product Evaluation Boards and Kits > Eval Tools for MCUs > STM32VLDISCOVERY. The main heading is 'STM32VLDISCOVERY Discovery kit for the Value line STM32F100'. There are four tabs: 'Quick view', 'Design support' (which is selected), 'Orderable products', and 'Related information'. Under the 'Design support' tab, there is a 'Description' section with the following text: 'The STM32VLDISCOVERY is a low-cost and quick way to discover the STM32 value line. It includes everything required for beginners and experienced users to get started quickly. The STM32 value line Discovery includes an STM32F100 value line microcontroller in a 64-pin LQFP and an in-circuit ST-Link debugger / programmer to debug Discovery applications and other target board applications. A large number of free ready-to-run application firmware examples are available on www.st.com/stm32-discovery to support quick evaluation and development using the LEDs, button and extension header for connection to other boards or devices.' Below the description is a 'Key Features' section with a bulleted list:

- STM32F100RB microcontroller, 128 KB Flash, 8 KB RAM in 64-pin LQFP
- On-board ST-Link with selection mode switch to use the kit as a stand-alone ST-Link (with SWD connector)
- Designed to be powered by USB or an external supply of 5 V or 3.3 V
- Can supply target application with 5 V and 3 V
- Two user LEDs (green and blue)
- One user push button
- Extension header for all QFP64 I/Os for quick connection to prototyping board or easy probing

At the bottom, there is a 'Board Photo' section.

図 22 STM32VLDISCOVERY の解説ページ

Software & Development Tools		
<ul style="list-style-type: none"> FIRMWARE OPERATING SYSTEMS SOFTWARE FUNCTIONS 	<ul style="list-style-type: none"> CONFIGURATION UTILITIES SOFTWARE DEMOS TEST UTILITIES 	<ul style="list-style-type: none"> DEVICE PROGRAMMERS SOFTWARE DRIVERS
FIRMWARE		
Description	Version	Size
STM32VLDISCOVERY firmware package (AN3268)	1.0.0	1630KB

図 23 Software & Development Tools

8¥stm32vldiscovery_package¥Project¥Master Workspace¥EWARMv5			
名前	サイズ	種類	更新日時
settings		ファイル フォルダ	2012/05/18 2:23
Value_Line_Discovery.eww	2 KB	IAR IDE Workspace	2012/05/14 14:59

図 24 Value_Line_Discovery.eww

開くと、複数のサンプルプロジェクトが記録されたワークスペースが展開されます (図 25)。

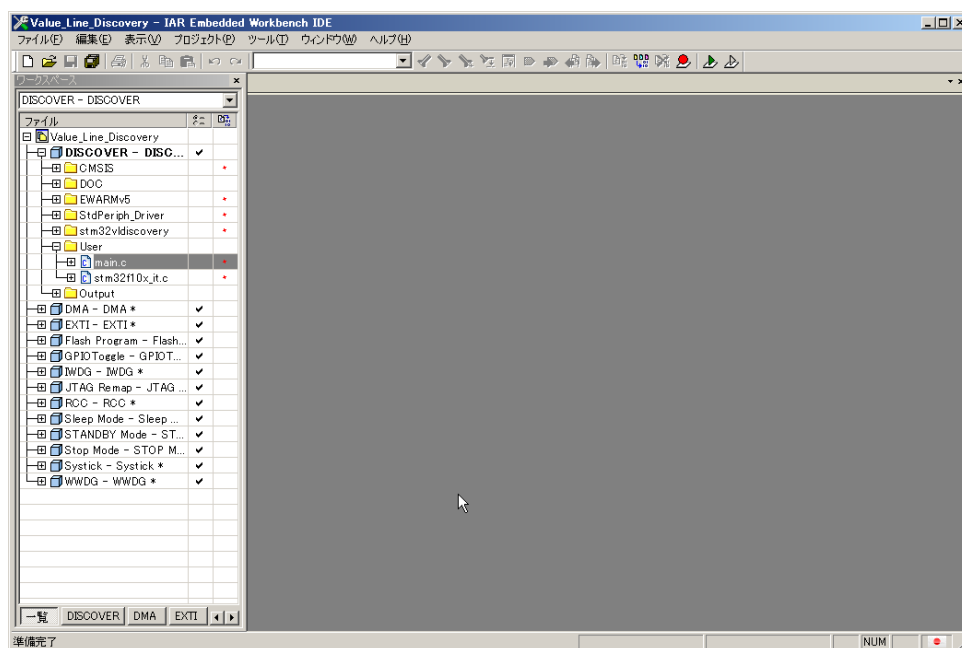


図 25 EWARM

EWARM のバージョンアップに伴い、一部変更がありますので、サンプルプロジェクトはそのままで、コンパイルが成功しません。STM32VLDISCOVERY に出荷時にダウンロードされているファームウェアである DISCOVERY プロジェクトをコンパイル・実行できるようにし、STM32VLDISCOVERY のプログラム開発の流れを学びます。

ワークスペース中の DISCOVERY の項目で右クリックし、オプションを選択します (図

26)。DISCOVERY の一般オプションのライブラリ設定で、“CMSIS を設定する”にチェックを入れます (図 27)。次に C/C++コンパイラのプリプロセッサから、追加のインクルードディレクトリの以下のエントリを削除し、OK ボタンを押して、オプションの変更を終了します (図 28)。

```
$PROJ_DIR$¥..¥..¥Libraries¥CMSIS¥CM3¥CoreSupport¥
```

プロジェクトメニューからメイクを選択し、コンパイルを実行します (図 29)。エラーの合計数 0 と表示され、書き込みに使用されるバイナリファイルが作成されます (図 30)。

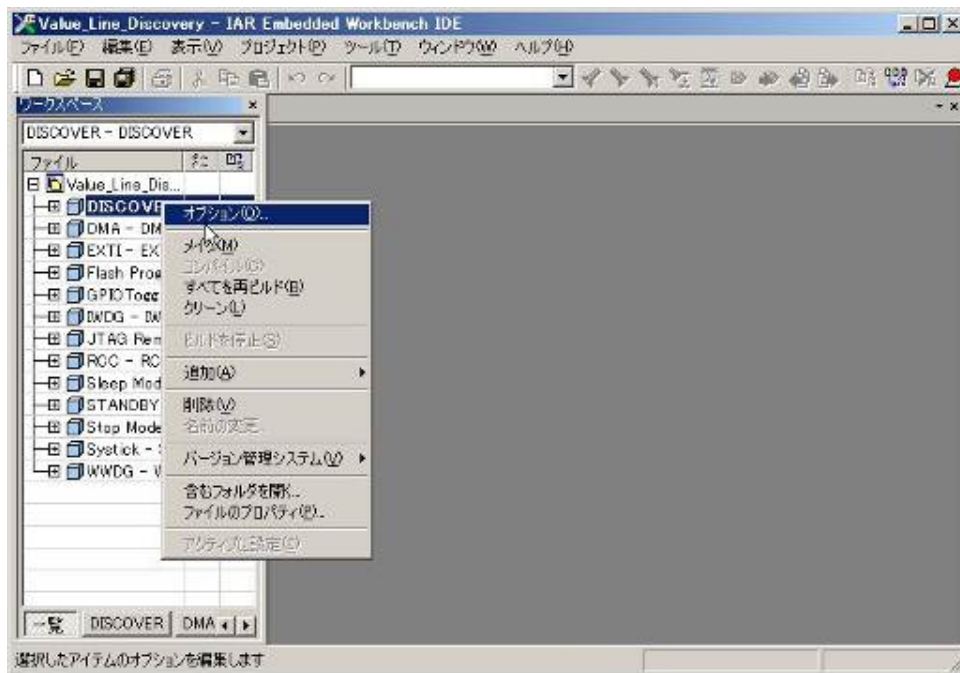


図 26 プロジェクト”DISCOVERY”

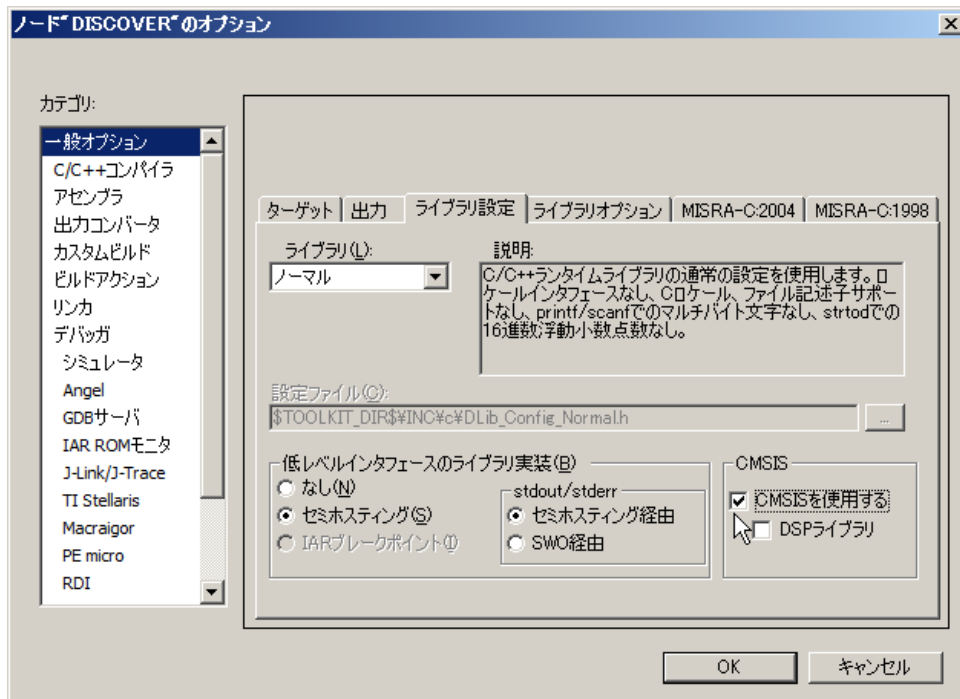


図 27 プロジェクト”DISCOVERY”の一般オプション

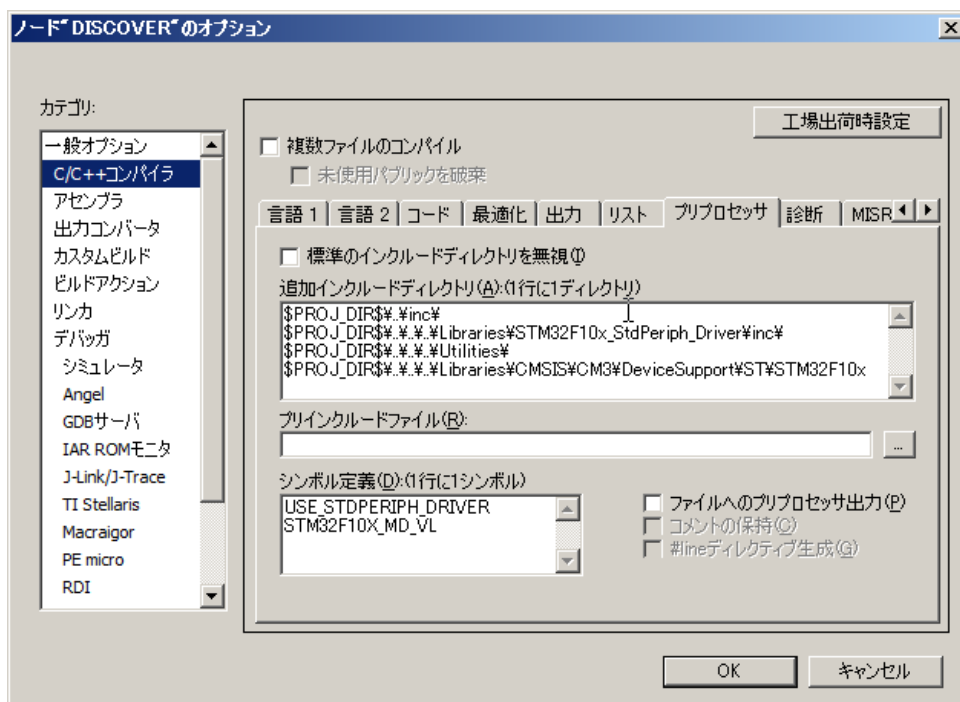


図 28 プロジェクト”DISCOVERY”の C/C++コンパイラ

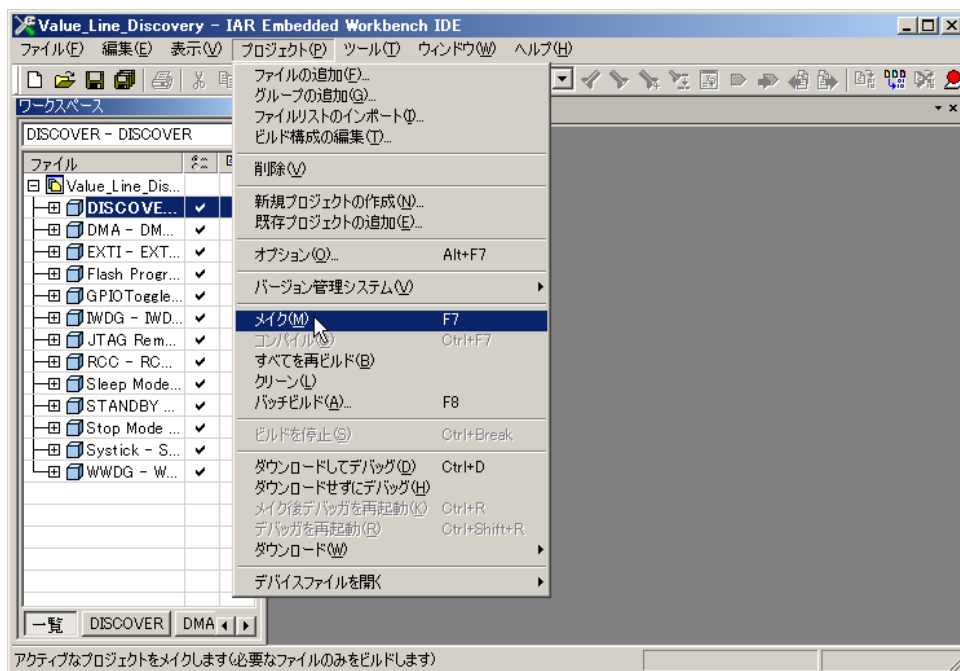


図 29 プロジェクトのメイク

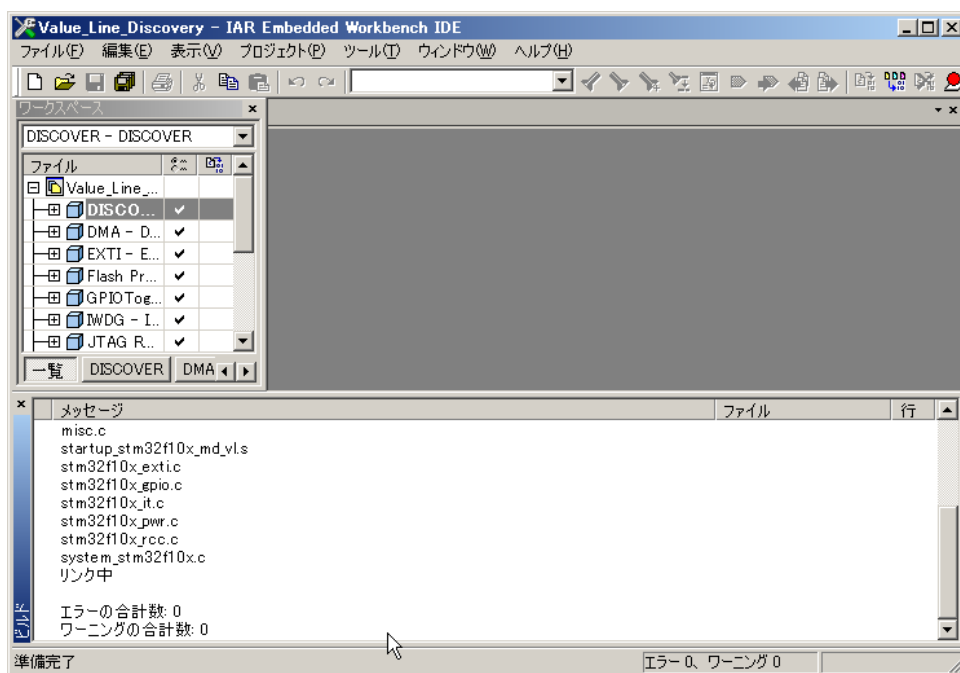


図 30 コンパイルの成功

メイクが終了したら、プログラムをマイコンにダウンロードして、デバッグを実行します (図 31)。図 31 USB ケーブルを使ってマイコンと PC を接続します。プロジェクトメニューから”ダウンロードしてデバッグ”を選択し、ファームウェアをダウンロードします。

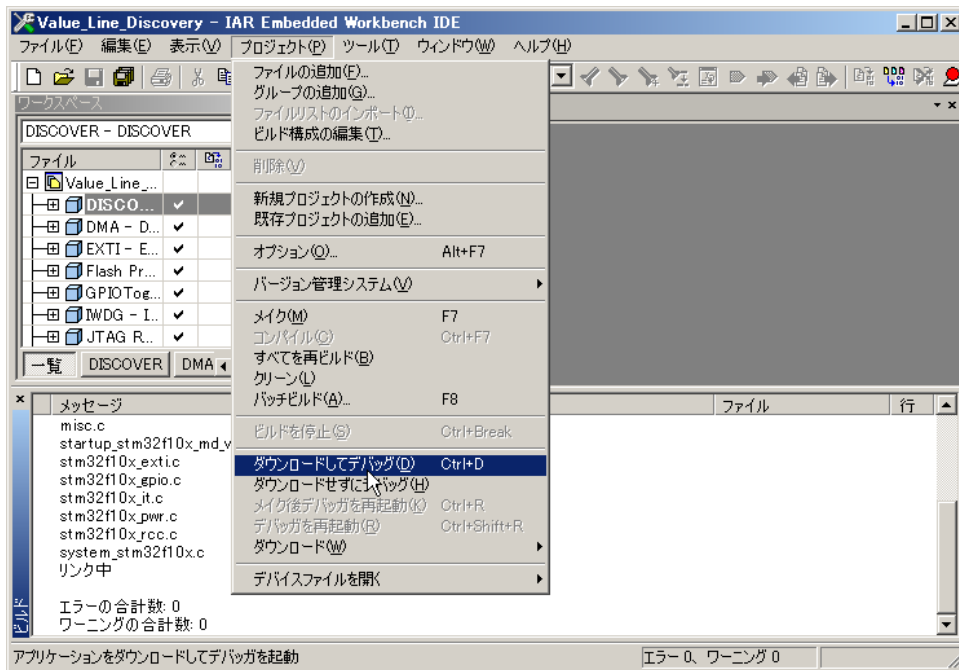


図 31 ダウンロードしてデバッグ

ファームウェアのダウンロードが終了すると、図 32 のデバッグ画面に移行します。ダウンロードした状態では、まだプログラムは実行されていないので、[F5]キーを押して、プログラムを実行します

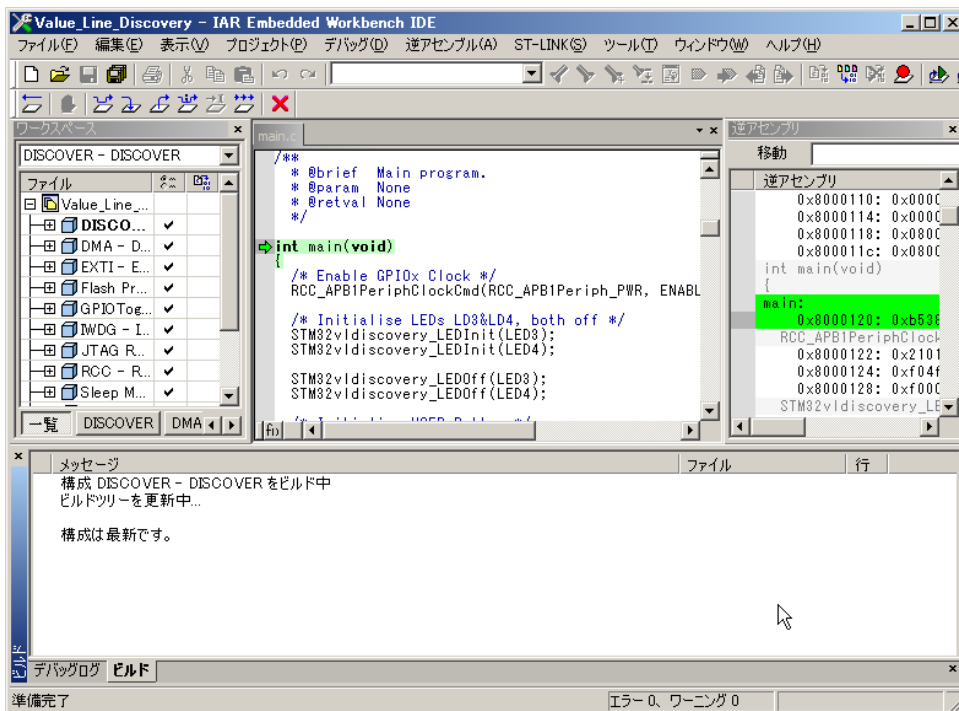


図 32 プログラムのデバッグ

プログラムが正常に実行されると、STM32VLDISCOVERY 上の LD1 が点滅し始めます。DISCOVERY プロジェクトは、USER ボタンが押されると、LD4 が点灯し、LD3 の点灯周期が変化するプログラムですので、USER ボタンを押すと、LD4 の青色 LED が点灯し、消灯すると LD3 の点灯周期が変化します。

DISCOVERY プロジェクトのプログラムについて、詳しく見ていきます。DISCOVERY プロジェクトは、複数のファイルを含むいくつかのグループで構成されています。

CMSIS グループは、CMSIS という ARM 社が推奨するライブラリ規格の CPU コアに関する処理を含んだファイルが登録されています。EWARMv5 グループは、プログラムの実行開始時に必要な初期化処理などを含んだスタートアップファイルが登録されています。

StdPeriph_Driver グループは、STM 社が配布している STM32 シリーズに搭載されている周辺機能のドライバプログラムなどが含まれています。Stm32vldiscovery グループは、

STM32VLDISCOVERY に関する処理を含むファイルが登録されています。User グループは、ユーザーが作成したファイルが含まれるグループです。Output グループは、コンパイルの結果作成されるファイルが含まれています。この中で、ユーザーが作成するプログラムは、通常 User グループに含まれるファイルです。図 33 では、main.c が、ユーザーが作成すべきファイルになります。

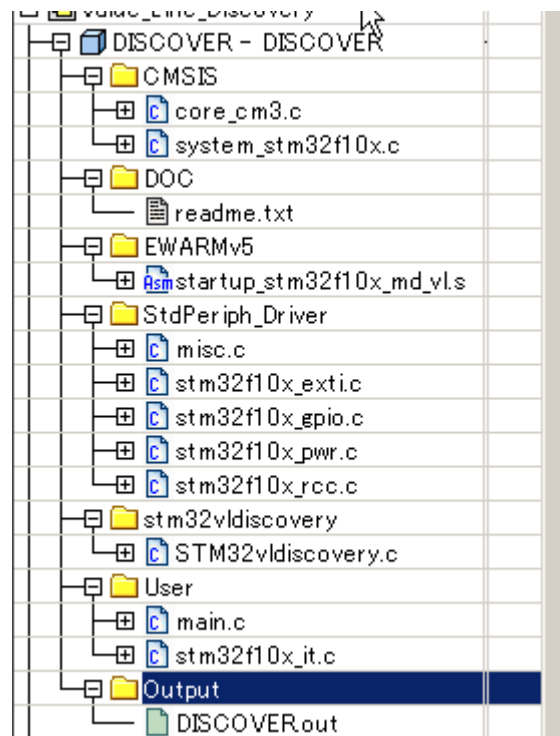


図 33 DISCOVERY プロジェクトの構造

リスト 1 は、DISCOVERY プロジェクトの main.c 中の main 関数です。この部分から CORTEX-M3 のプログラムの基本的な流れを学ぶことができます。52～53 行で、周辺機器

にクロックを供給し周辺機器を利用可能にします。CORTEX-M3 では、使わない周辺機器には、通常クロックを供給せず停止状態になっているので、設定などを始める前に、使用する周辺機器にクロックを供給して動作を開始させます。55～60 行で、STM32VLDISCOVERY 上の LED の初期化を行っています。具体的な処理を行っている関数が知りたい場合は、図 34 のように、調べたい関数・変数などの上で右クリックで表示されるメニューの中の”定義に移動”を選択すると、その関数・変数などの詳細を知ることができます。ここでは、STM32vldiscovery_LEDInit という関数の詳細を調べています。STM32vldiscovery_LEDInit 関数は、STM32vldiscovery.c に定義されている関数で、LED3, LED4 を駆動するための初期化処理を担当する関数です。汎用入出力にクロックを供給し、入出力方向を出力に設定しています。63 行の STM32vldiscovery_PBInit 関数は、STM32VLDISCOVERY 上の押しボタンを読み込むための処理を行っている関数です。66 行で、1m 秒ごとに割り込みが入るように設定をしています。その割り込を使って、140 行および 149 行の Delay 関数が実装されています。72～112 行でシステムクロックの設定を行っています。115 行の While 文からが LED を点滅させる部分になります。

このように、基本的なプログラムの流れは、

各種周辺機器の設定→処理の開始→while 文で無限ループ

となります。

```

50  int main(void)
51  {
52      /* Enable GPIOx Clock */
53      RCC_APB1PeriphClockCmd(RCC_APB1Periph_PWR, ENABLE);
54
55      /* Initialise LEDs LD3&LD4, both off */
56      STM32vldiscovery_LEDInit(LED3);
57      STM32vldiscovery_LEDInit(LED4);
58
59      STM32vldiscovery_LEDOff(LED3);
60      STM32vldiscovery_LEDOff(LED4);
61
62      /* Initialise USER Button */
63      STM32vldiscovery_PBInit(BUTTON_USER, BUTTON_MODE_GPIO);
64
65      /* Setup SysTick Timer for 1 msec interrupts  */
66      if (SysTick_Config(SystemCoreClock / 1000))
67      {
68          /* Capture error */

```

```
69     while (1);
70 }
71
72 /* Enable access to the backup register => LSE can be enabled */
73 PWR_BackupAccessCmd(ENABLE);
74
75 /* Enable LSE (Low Speed External Oscillation) */
76 RCC_LSEConfig(RCC_LSE_ON);
77
78 /* Check the LSE Status */
79 while(1)
80 {
81     if(LSE_Delay < LSE_FAIL_FLAG)
82     {
83         -----中略-----
84
85         /* Disable LSE */
86         RCC_LSEConfig(RCC_LSE_OFF);
87         break;
88     }
89 }
90
91 /* main while */
92 while(1)
93 {
94     if(0 == STM32vldiscovery_PBGetState(BUTTON_USER))
95     {
96         if(KeyState == 1)
97         {
98             if(0 == STM32vldiscovery_PBGetState(BUTTON_USER))
99             {
100                 /* USER Button released */
101                 KeyState = 0;
102                 /* Turn Off LED4 */
103                 STM32vldiscovery_LEDOff(LED4);
104             }
105         }
106     }
107 }
```

```
129     }
130     else if(STM32vldiscovery_PBGetState(BUTTON_USER))
131     {
132         if(KeyState == 0)
133         {
134             if(STM32vldiscovery_PBGetState(BUTTON_USER))
135             {
136                 /* USER Button released */
137                 KeyState = 1;
138                 /* Turn ON LED4 */
139                 STM32vldiscovery_LEDOn(LED4);
140                 Delay(1000);
141                 /* Turn OFF LED4 */
142                 STM32vldiscovery_LEDOff(LED4);
143                 /* BlinkSpeed: 0 -> 1 -> 2, then re-cycle */
144                 BlinkSpeed ++ ;
145             }
146         }
147     }
148     count++;
149     Delay(100);
150     /* BlinkSpeed: 0 */
151     if(BlinkSpeed == 0)
152     {
153         if(4 == (count % 8))
154             STM32vldiscovery_LEDOn(LED3);
155         if(0 == (count % 8))
156             STM32vldiscovery_LEDOff(LED3);
157     }
158     /* BlinkSpeed: 1 */
159     if(BlinkSpeed == 1)
160     {
161         if(2 == (count % 4))
162             STM32vldiscovery_LEDOn(LED3);
163         if(0 == (count % 4))
164             STM32vldiscovery_LEDOff(LED3);
```

```

165     }
166     /* BlinkSpeed: 2 */
167     if(BlinkSpeed == 2)
168     {
169         if(0 == (count % 2))
170             STM32vldiscovery_LEDOn(LED3);
171         else
172             STM32vldiscovery_LEDOff(LED3);
173     }
174     /* BlinkSpeed: 3 */
175     else if(BlinkSpeed == 3)
176         BlinkSpeed = 0;
177 }
178 }

```

リスト 1 DISCOVERY プロジェクトの main.c(抜粋)

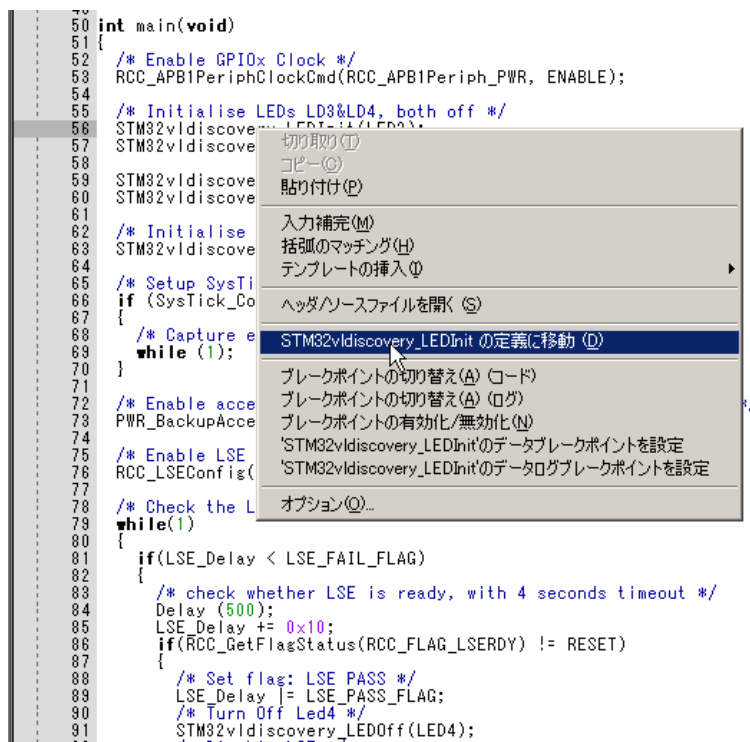


図 34 詳細を調べる方法

8. AcRobotサンプルプログラム

下記の URL から AcRobot 向けのサンプルプログラムをダウンロードし、作業ディレクト

りの下に展開します。

http://www.takitagiken.com/~takita/RSJ/STM32VLDISCOVERY_SAMPLE.zip

8.1. サンプルプログラムSTM32_7SEGMENT

STM32_7SEGMENT プロジェクトは、7セグメント LED を点灯するプログラムです。7セグメント LED の点灯のタイミングは、for 文を使用しています。

8.2. サンプルプログラムSTM32_7SEGMENT_TIMER

CORTEX-M3 が持つタイマーを使用して時間を計測する関数を含んだ7セグメント LED の点灯プログラムです。

8.3. サンプルプログラムSTM32_USART

本キットでは、STM32 が持つ同期・非同期シリアル通信モジュールの内 USART1 の入出力ピンが RS-232 レベルコンバータに接続されています。STM32_USART は、USART1 を使用して、printf 関数に相当する Acrobot_USART_Printf 関数を実装しています。

8.4. サンプルプログラムSTM32_ADC

本キットには、半固定抵抗が一つ搭載されています。半固定抵抗のツマミを回すと、出力ピンの電圧が変化します。また、ジャイロセンサからの出力を角速度に応じて電圧が変化します。そのような電圧の変化をマイコンに取り込むのが AD コンバータです。

STM32_ADC プロジェクトは、STM32VLDISCOVERY がもつ AD コンバータを使用するための関数を含むプログラムです。AD 変換された値は、Acrobot_USART_Printf 関数によりシリアルポートに出力されます。

8.5. サンプルプログラムSTM32_Encoder

本キットでは、上下の基板の角度を、光学式エンコーダを用いて計測します。STM32_Encoder プロジェクトは、光学式エンコーダの信号を取り込むカウンタ回路の処理を含んでいます。取り込んだ値は、Acrobot_USART_Printf 関数によりシリアルポートに出力されます。

8.6. サンプルプログラムSTM32_PWM

本キットでは、PWM 信号を用いて、モータを駆動することができます。STM32_PWM プロジェクトは、タイマー回路を用いて、一定の周期で、エンコーダから現在の関節角度を取得・比較を行い、目標角度に向かってモータを回転させる相補 PWM 信号を生成する処理が実装されています。本キットでは、モータの電源は、フォトカプラを使ったスイッチ回路で ON/OFF できる構成になっており、そのための制御も含まれます。

8.7. サンプルプログラムSTM32_ACROBOT

本キットのように、複数の自由度(動作可能な方向などの数)を持ち、全体の自由度よりも制御することができる自由度が少ないシステムを劣駆動システムと言います。本キットでは、上下の基板をつないでいる関節を曲げることで、鉄棒周りの振りを制御することができます。

STM32_ACROBOT プロジェクトは、AD コンバータを用いて取得したジャイロセンサの

値を使って、鉄棒周りの角速度が大きくなるように、関節の曲げ伸ばしを実行します。

9. 終わりに

本稿では、劣駆動システムである鉄棒ロボットの作り方を通して、マイコン、モータ、モータドライバ、エンコーダを使ってロボットを制御する方法を説明しました。現在開発されている多くのロボットも、個々の要素の性能は異なりますが、基本的な制御構造は似ています。次は是非自分のアイデアで、周囲をあっと思かせるような面白いロボットを作ってみてください。

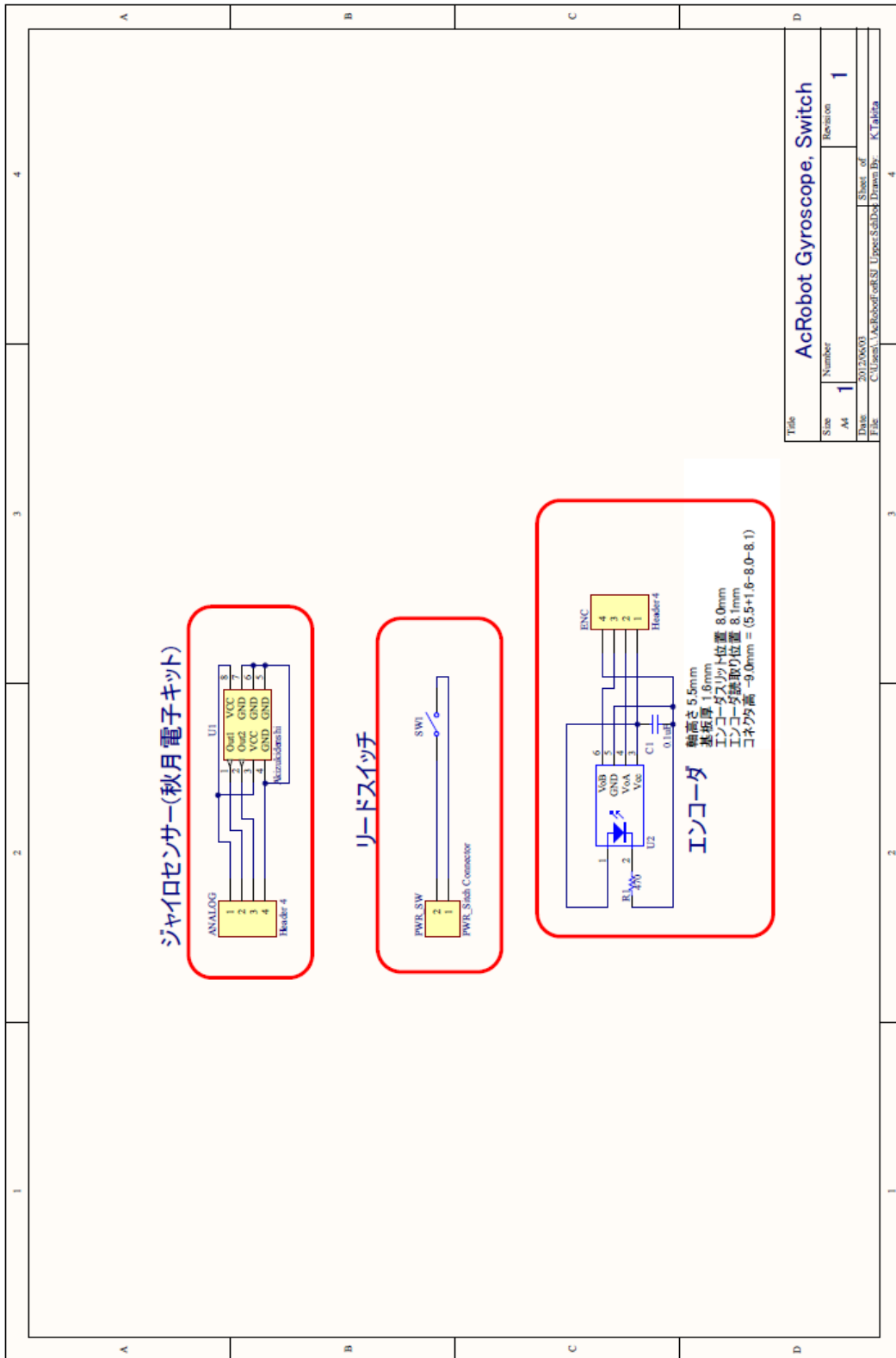


図 35 回路図(ジャイロセンサ・リードスイッチ・エンコーダ(オプション))

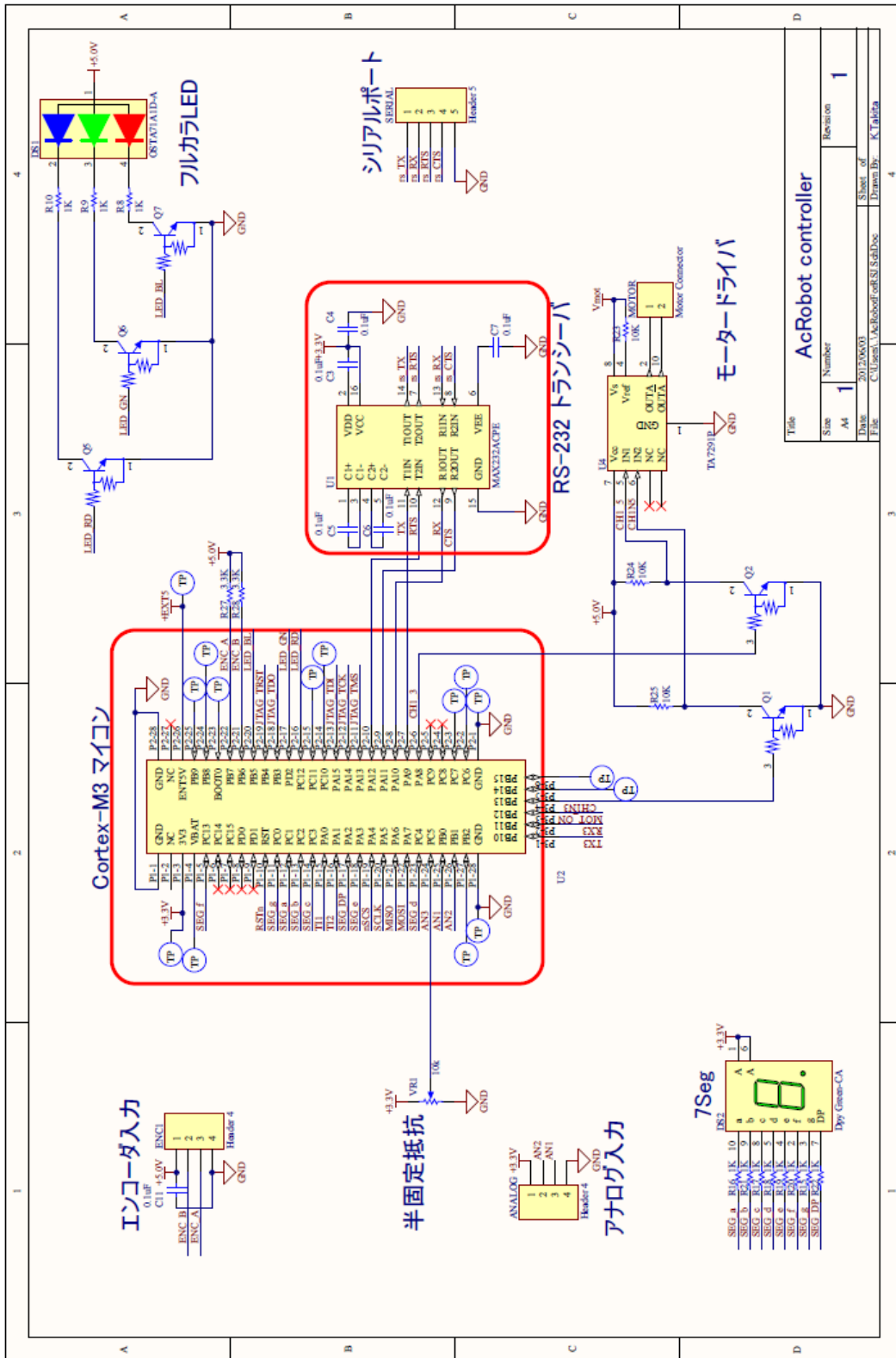


図 36 回路図(STM32VLDiscovery 周辺)

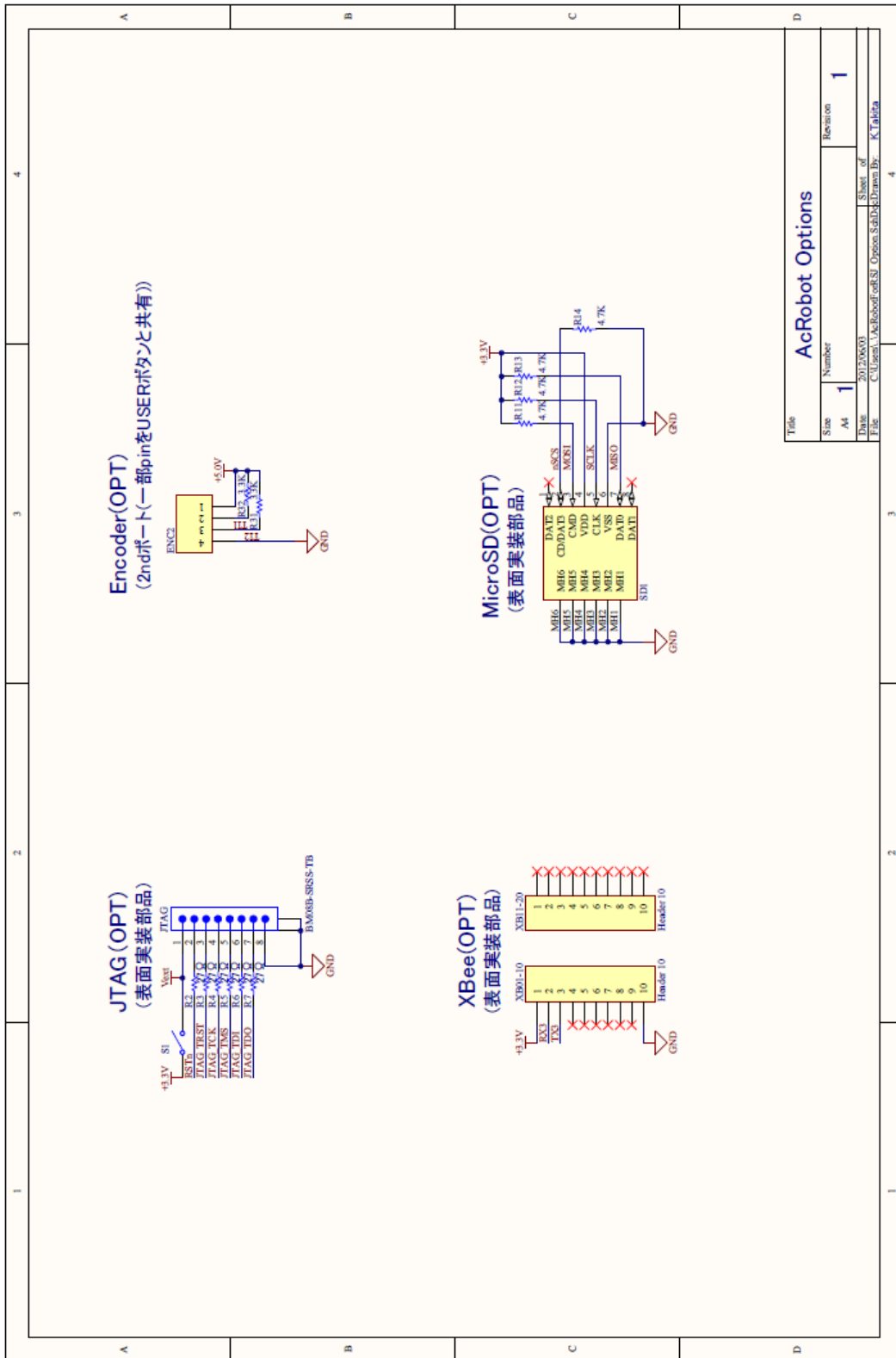


図 37 回路図(本セミナーで実装しない部品)

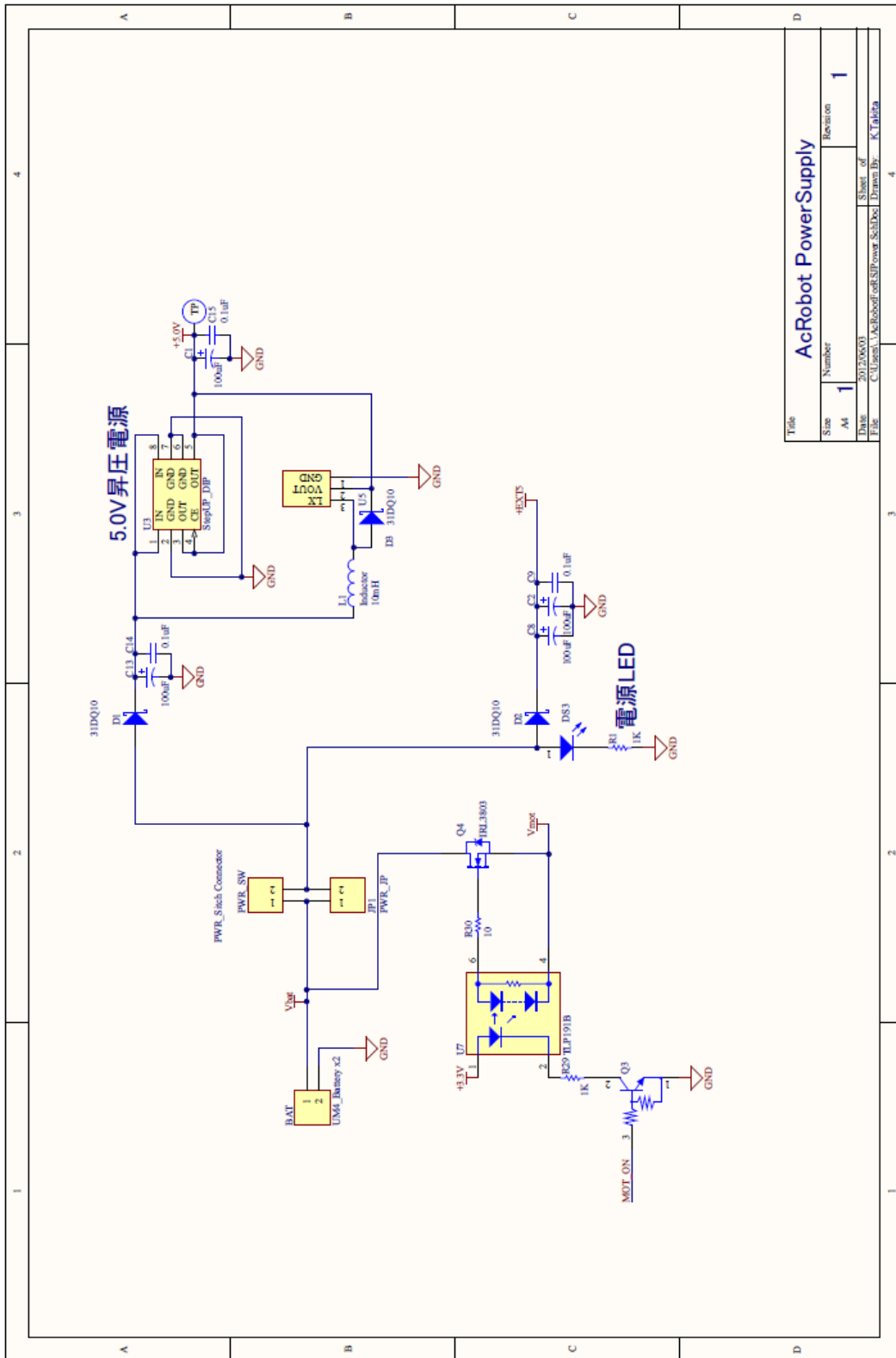


図 39 回路図(電源)

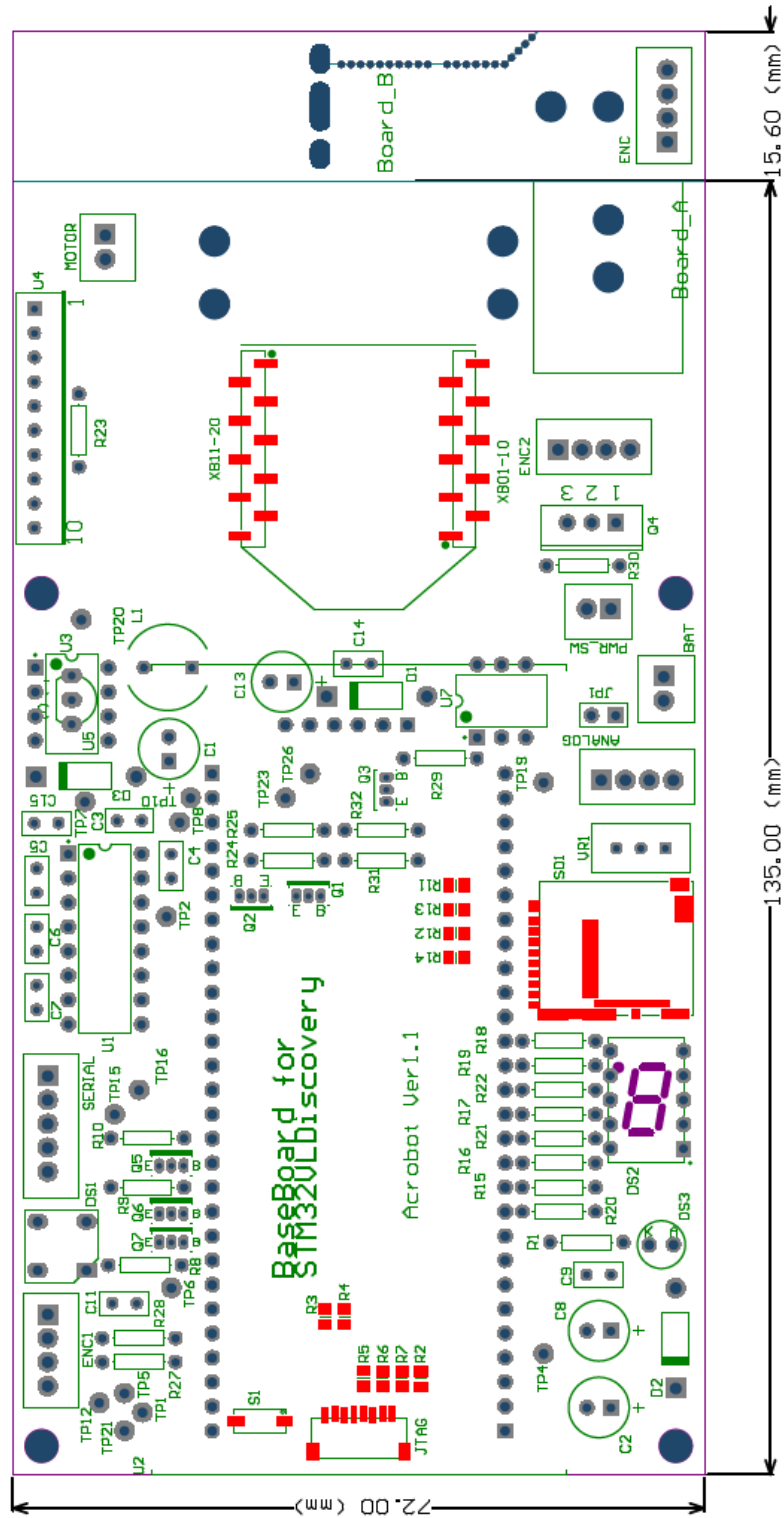


図 40 基板部品配置図表面(Board_A,Board_B)

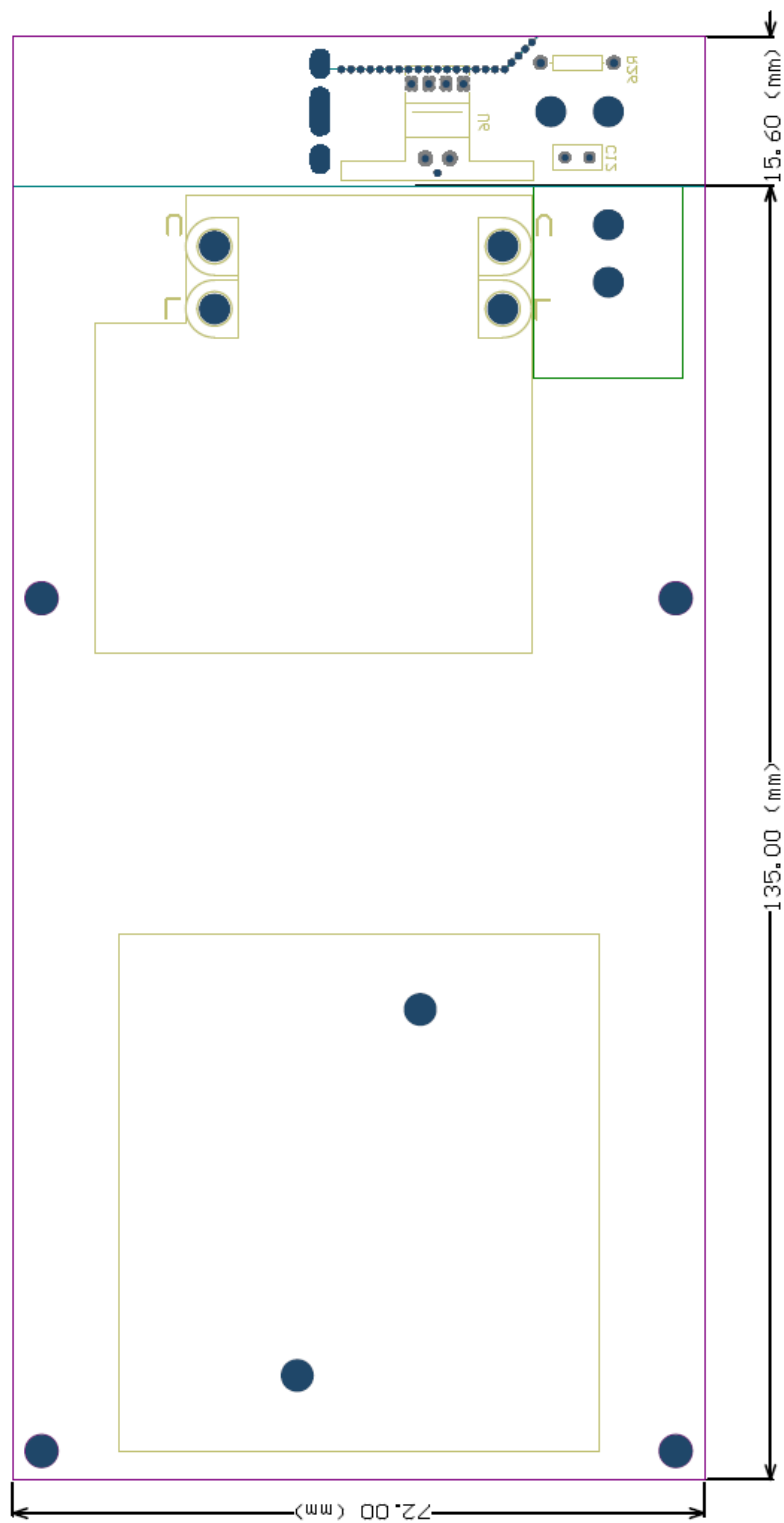


图 41 基板部品配置图裏面(Board_A,Board_B)

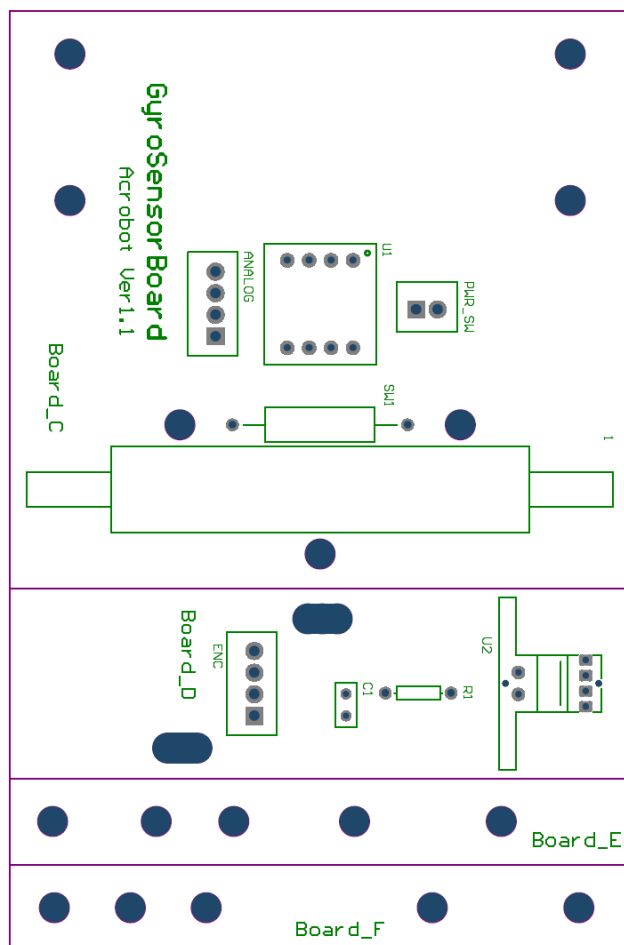


図 42 基板部品配置図表面(Board_C, Board_D, Board_E, Board_F)

表 3 部品表(Board_A、Board_B)

部品番号	メーカー	製品名	数	値	実装の有無
ANALOG, ENC, ENC1	MOLEX	5045-04A	3	4ピン	
C1, C2, C8, C13			4	100uF	
C3, C4, C5, C6, C7, C9, C11, C12, C14, C15			10	0.1uF	
D1, D2, D3	Panjit	1S3	3	1.0A 0.5V	
DS1	OptoSupply	OSTA71A1D-A	1	角形フルカラー	
DS2	Sharp	GL9A040G	1	7セグメント	
DS3	OptoSupply	OSHR5111A-TU	1	赤	
JP1		ピンヘッダ	1	2ピン	
L1	太陽誘電株式会社	LHL08NB470K	1	47uH	
PWR_SW	MOLEX	5045-02A	1	2ピン	
Q1, Q2, Q3, Q5, Q6, Q7	Toshiba	RN1201 or RN1202	6		
Q4	Fairchild	FQPF14N30	1	Nch MOSFET	
R1, R8, R9, R10, R15, R16, R17, R18, R19, R20, R21, R22, R29			13	1K	
R23, R24, R25			3	10K	
R26			1	470	
R27, R28			2	3.3K	
R30			1	10	
SERIAL	MOLEX	5045-05A	1	5ピン	
U1	Intersil	ICL3232CPZ	1		
U2	STM	STM32VLDISCOVERY	1		
U4	東芝	TA7291P	1	1A(ave) 2A(peak)	
U5	HOLTEK	Ht7750A	1	5V	
U6	KODENSHI	KE203	1		
U7	Toshiba	TLP191B	1	7V 24uA	
VR1	Bourns	3296W	1	10k	
ENC2	MOLEX	5045-04A	1		実装しない
JTAG	日本圧着端子製造	BM08B-SRSS-TB	1	10ピン	実装しない
R11, R12, R13, R14			4	4.7K	実装しない
R2, R3, R4, R5, R6, R7		RMC1/16S270F	6	27Ω	実装しない
R31, R32			2	3.3K	実装しない
S1	日本電産コパル	CHS-01TA	1		実装しない
SD1	ヒロセ	DM3AT-SF-PEJ	1		実装しない
U3	秋月電子	STEP-UP_DIP_5.0V	1	3.3V 0.2A	実装しない
XB01-10, XB11-20			2		実装しない

表 4 部品表(Board_C、Board_D)

部品番号	メーカー	製品名	数	値	実装の有無
U1	秋月電子	K-04912	1		
ANALOG	MOLEX	5045-04A	1		
PWR_SW	MOLEX	5045-02A	1		
SW1	NECTーキン	RD-7B	1		
C1			1		実装しない
U2	KODENSHI	KE203	1		実装しない
ENC	MOLEX	5045-04A	1		実装しない
R1			1		実装しない

表 5 機械系部品

部品名	数	関係部品
ナベネジ1	M3-5mm	10 Board_C, Board_E, Board_F, 円形アーム, L字金具
ナベネジ2	M3-10mm	8 鉄棒軸受け, ギアボックス, Board_A, Board_B, Board_C
サラネジ1	M3-12mm	10 電池ボックス, 鉄棒台
スペーサ	内径3mm-5mm	4
ナット	M3	20
アクリル板		
アクリル板		
磁石	1.5mm x 10mm	1
パイプ	外径4mm, 内径2mm	1

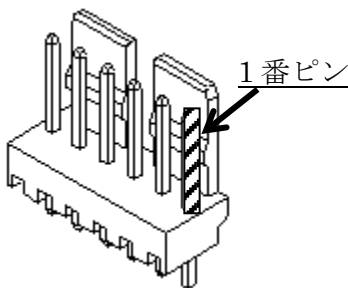


図 43 5045-0xA の向き(一番ピンを四角いパッドに挿入する)

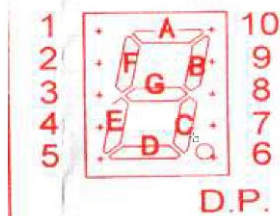
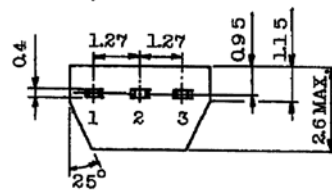
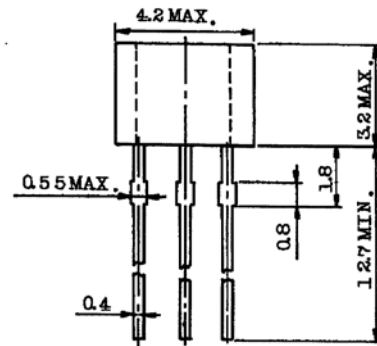


図 44 DS2(GL9A040G)のピン配置



- 1. EMITTER
- 2. COLLECTOR
- 3. BASE

図 45 RN120x の向き