「ロボットの作り方2012」
実習1　実習キット組立

6月16日 14:10-16:40

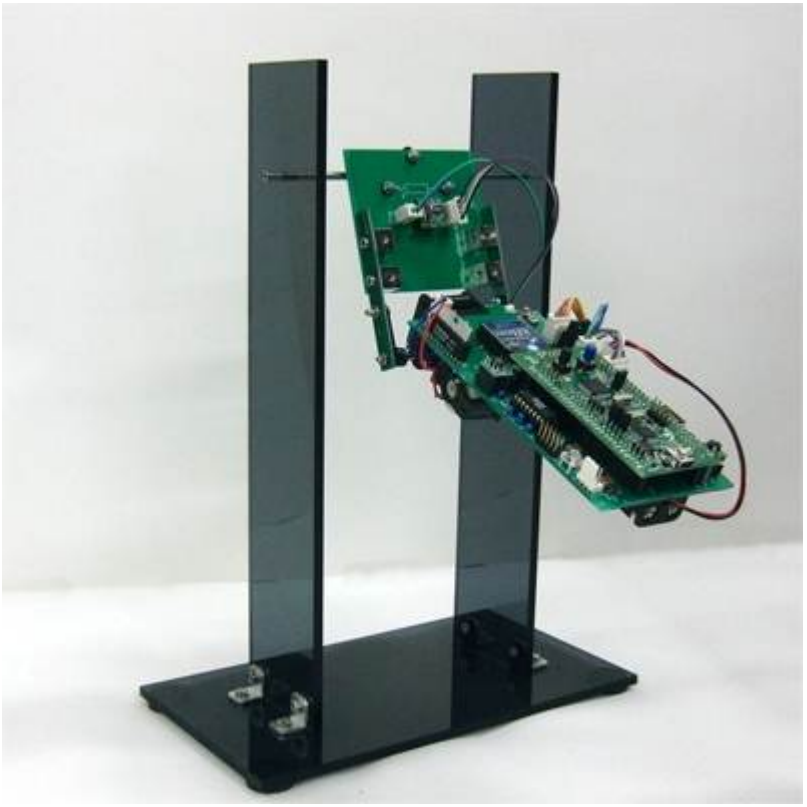
日本工業大学　滝田　謙介

滝田謙介略歴

- 1999年3月～ 北海道大学大学院研究科情報エレクトロニクス系専攻群システム情報工学専攻 複雑系工学講座 自律系工学分野にて、知能ロ ボット、進化計算論、行動型人工知能に関する研究による、システム工学・情報工学の博士号を取得。
- 1999年4月～ ロボット研究の第一人者である東京工業大学 広瀬茂男教授の研究室の研究員に着任。恐竜型2足歩行ロボット、ロボット用コントローラ等の開発に従事。電子回路・ コンピュータ・プログラムに関する知識を活かして、各種ロボットを開発
- 2003年4月～ 特定非営利活動法人国際レスキュー システム研究機構の研究員に就任。レスキューロボット・極限作業ロボットの制御システム、画像安定化システムなどの開発に従事。
- 2004年4月～ (株)ハイボットを 設立し、代表取締役に就任。電線点検ロ ボット、水陸両用ヘビ型ロボット(2005年愛知万博出展)などを開発。電子回路の知識を活かして、マイコンボード・ モータドライバなどを開発し、同社より、販売。
- 2009年8月　(株)ハイボットを退社。
- 2009年9月～ 滝田技研(株)として独立し、IRT製品の開発を始める。マイコンボート、モータドライ バ、充電マネージャー、配電回路などを開発
- 2011年4月～ 日本工業大学 創造システム工学科准教授に着任

本セミナーで組み立てるキット

- STM32VLDISCOVERY を搭載した鉄棒ロボット。
- ギア付きモータx1を搭載
- 光学式エンコーダ、フルカラーLED、7セグメント、ジャイロセンサなどを搭載
- PWM制御によるモータ制御実習用に開発



STM32VLDiscovery

ST Micro semiconductor社の
STM32シリーズの評価ボード

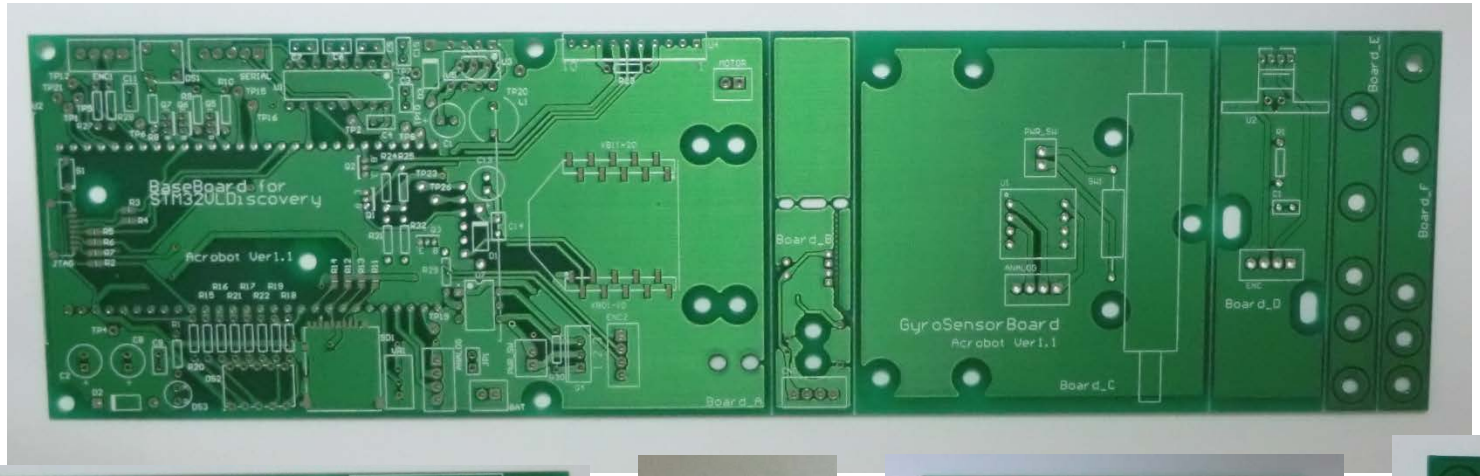
- STM32F100RBT6B, 128 KB Flash, 8 KB RAM
- ST-Link(STM社のデバッグインターフェース)
- 2つの赤色LED; LD1(USB通信), LD2(電源ランプ)
- 電源: USBバスパワー、外部5V、3.3V のいずれか
- ユーザー制御LED(LD3(緑)・LD4(青))
- 2つのプッシュボタン(User・Reset)

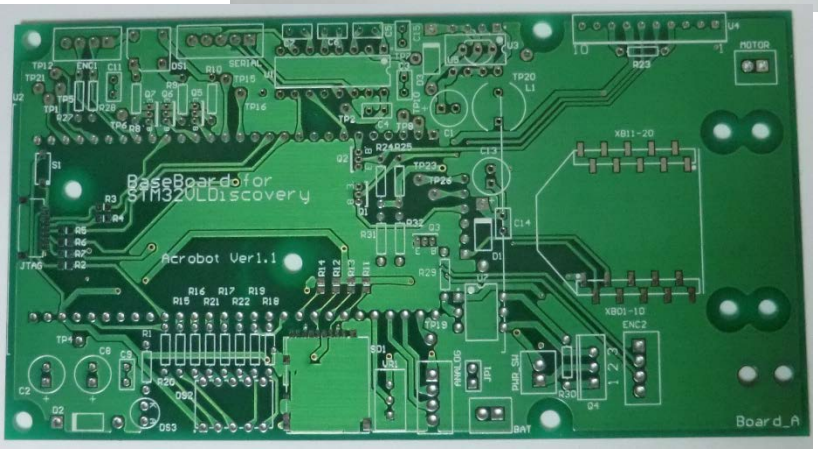
参考資料

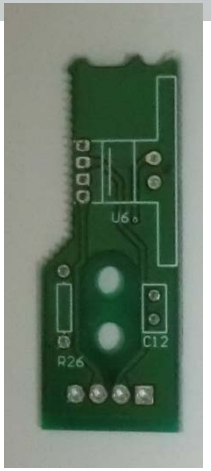
世界の定番ARMマイコン
超入門キット STM32ディスカバリ(CQ出版)

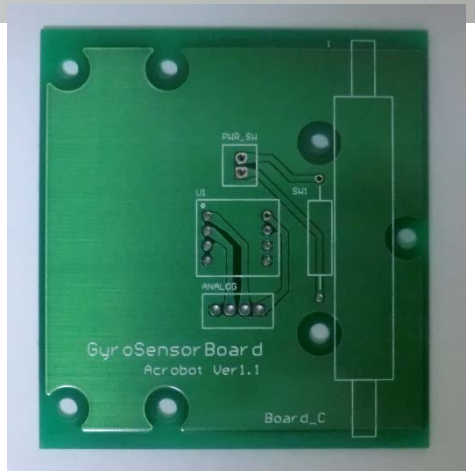
製作

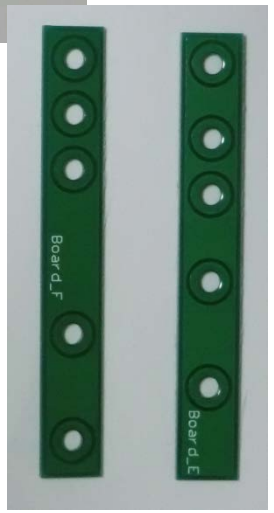
基板


BaseBoard for
STM32LDiscovery
Acrobot Ver1.1
Board_A
Board_B
GyroSensorBoard
Acrobot Ver1.1
Board_C
Board_D
Board_E
Board_F


BaseBoard for
STM32LDiscovery
Acrobot Ver1.1
Board_A


U6
R26
C12

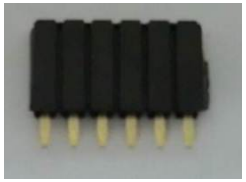

GyroSensorBoard
Acrobot Ver1.1
Board_C


Board_F
Board_E

基板の半田付けの準備

- ピンソケットの準備

1x28ピン ：2本

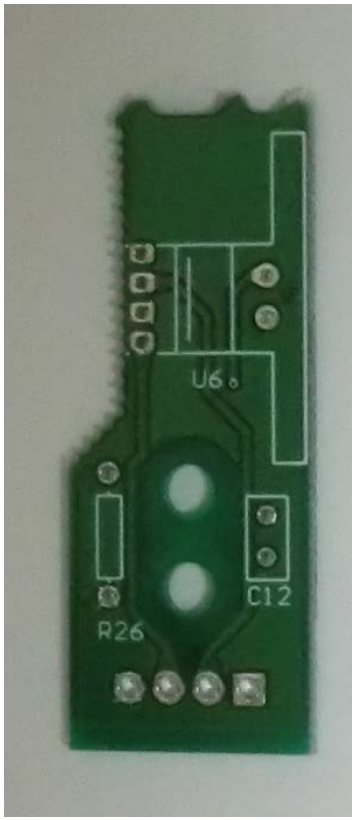


1x6ピン ：1本

1x4ピン ：2本

Board_Bについて(お詫びと訂正)

- 光学式エンコーダをBoard_Bの上に半田付けします。
- コネクタの印刷が裏表間違っていました。
- コネクタを半田付けする際は、R26，C12と同じ面に半田付けしてください。(四角いパッドが1番ピンです。)



部品表(Board_A、Board_B)

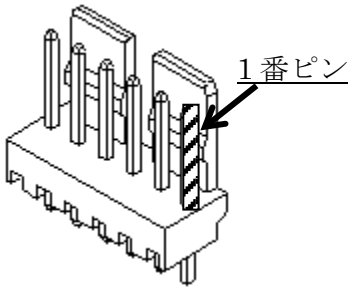
部品番号	メーカー	製品名	数	値	実装の有無
ANALOG, ENC, ENC1	MOLEX	5045-04A	3	4ピン	
C1, C2, C8, C13			4	100uF	
C3, C4, C5, C6, C7, C9, C11, C12, C14, C15			10	0.1uF	
D1, D2, D3	Panjit	1S3	3	1.0A 0.5V	
DS1	OptoSupply	OSTA71A1D-A	1	角形フルカラー	
DS2	Sharp	GL9A040G	1	7セグメント	
DS3	OptoSupply	OSHR5111A-TU	1	赤	
JP1		ピンヘッダ	1	2ピン	
L1	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	

部品上の表記

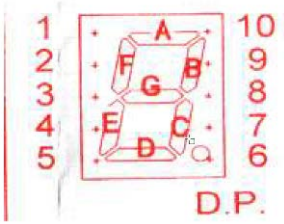
104

茶 黒 赤 金

茶 黒 橙 金
黄 紫 茶 金
橙 橙 赤 金



5045-0xAの向き(一番ピンを四角いパッドに挿入する)



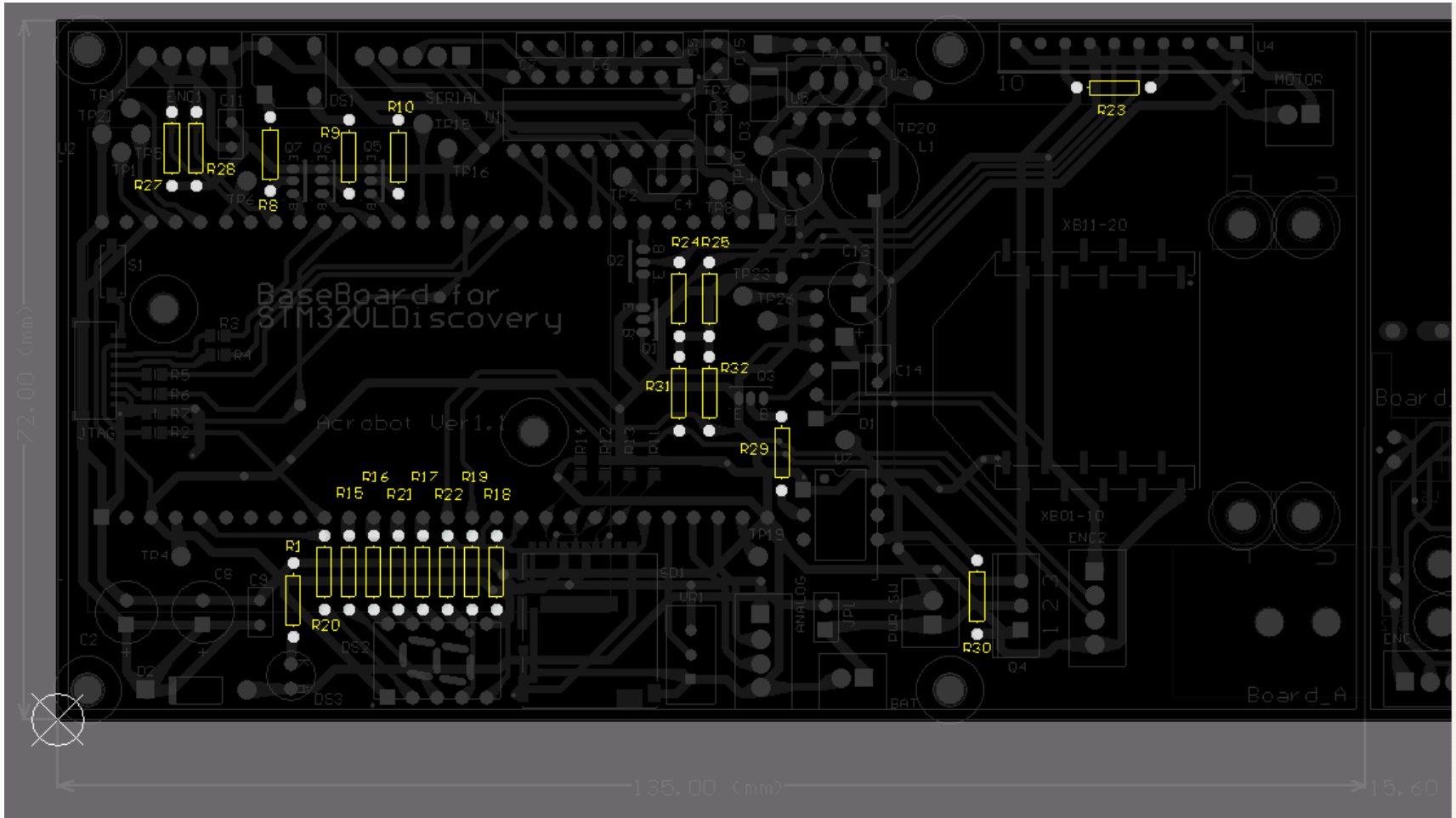
DS2(GL9A040G)のピン配置

部品表(Board_C、Board_D)

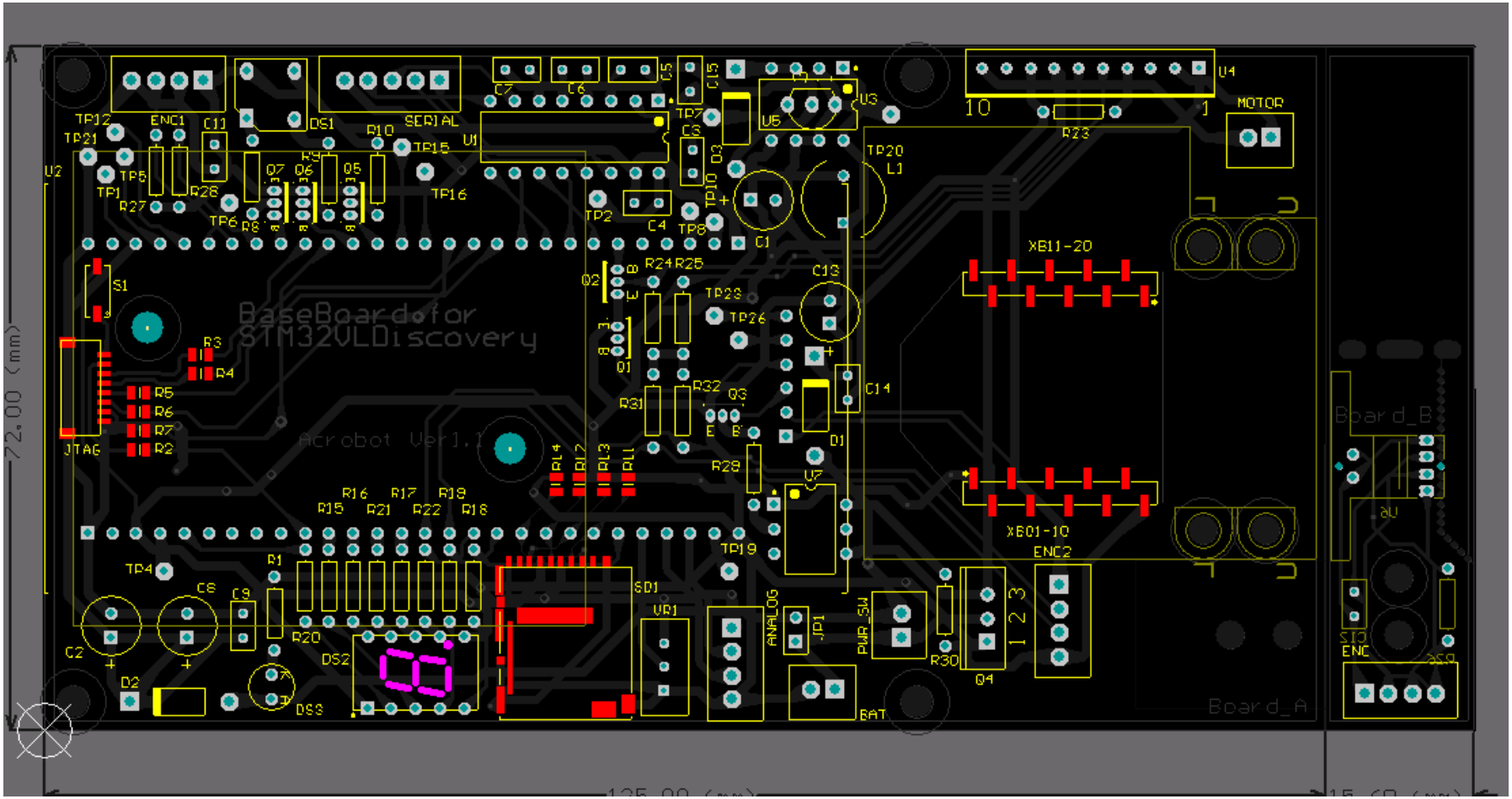
部品番号	メーカー	製品名	数	値	実装の有無
U1	秋月電子	K-04912	1		
ANALOG	MOLEX	5045-04A	1		
PWR_SW	MOLEX	5045-02A	1		
SW1	NECトーキン	RD-7B	1		

半田付けの順番(背の低いものから)

抵抗→ ダイオード→セラミックコンデンサ・・・

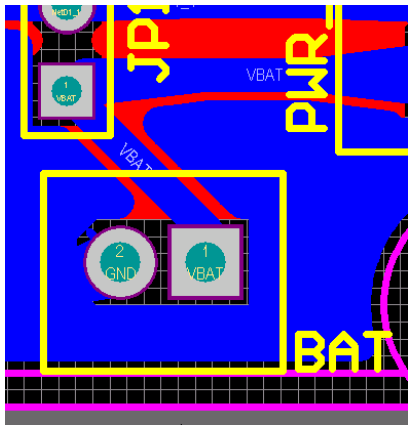


部品配置

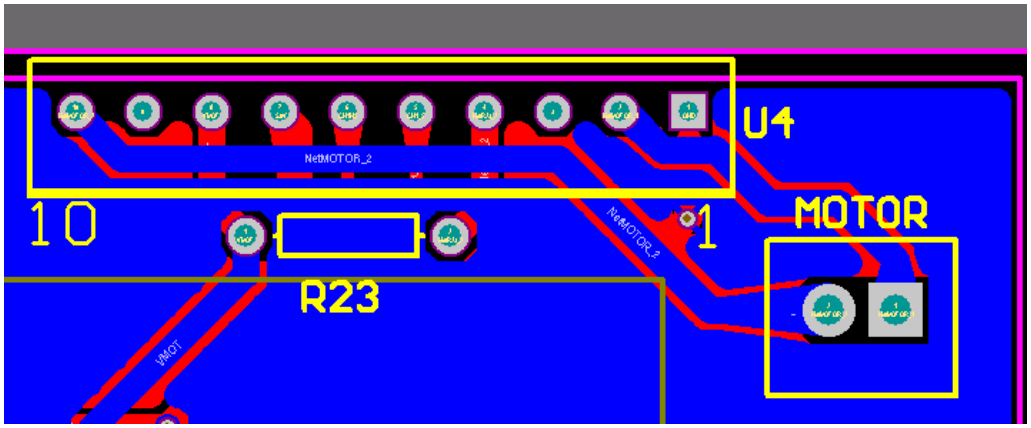

BaseBoardfor
STM32VLDiscovery
Acrobot Ver1.1
MOTOR
SERIAL
ENC1
ENC2
JTAG
XB11-20
XB01-10
Board_A
Board_B
72.00 (mm)

電源･モータ端子

四角いパッドが1番ピン



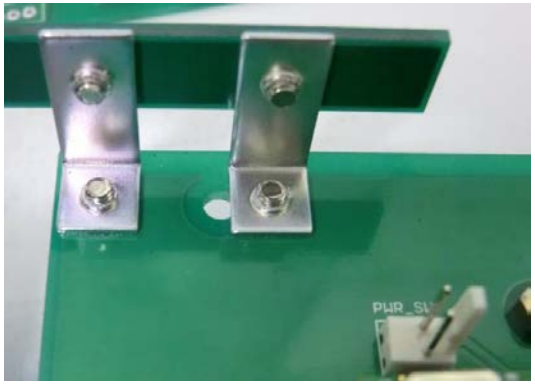
1番 → 赤
2番 → 黒

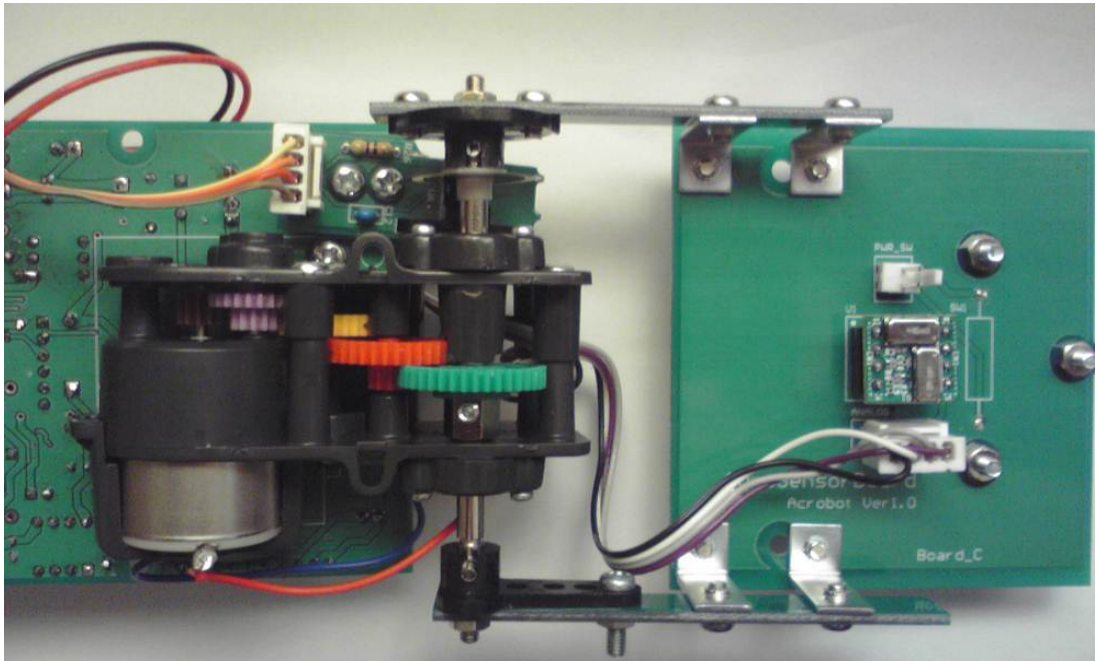


1番 →赤い配線
2番 →青い配線

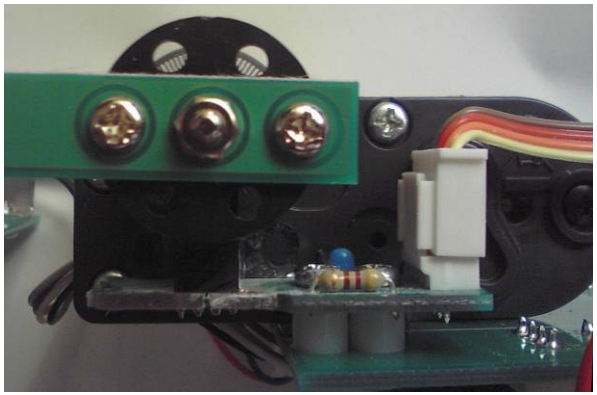
Board__CとBoard__E，Fとの接続

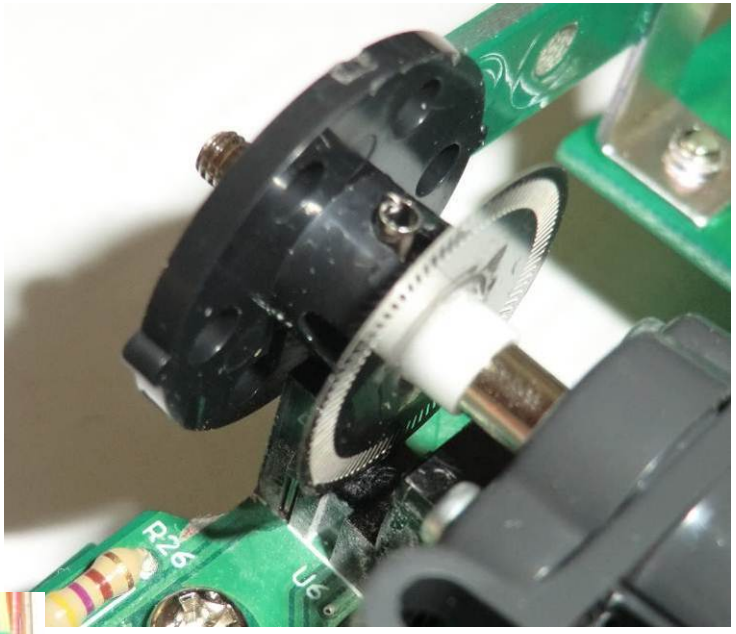
- L字金具を使って取り付け。





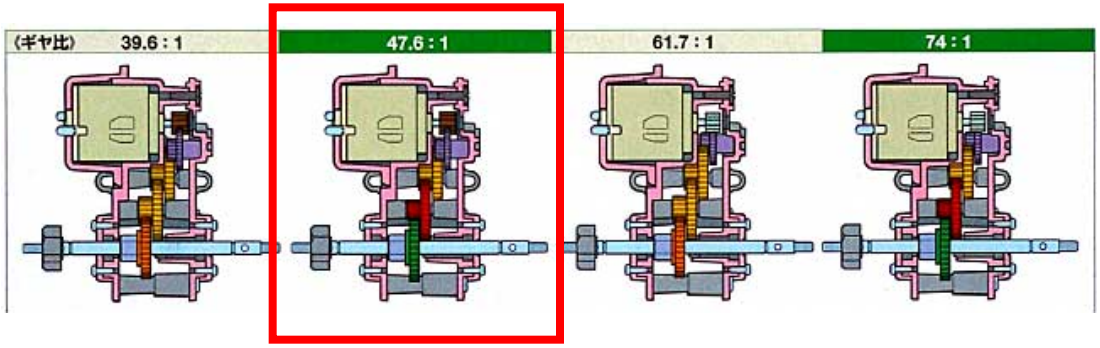
エンコーダ用ディスクの取り付け





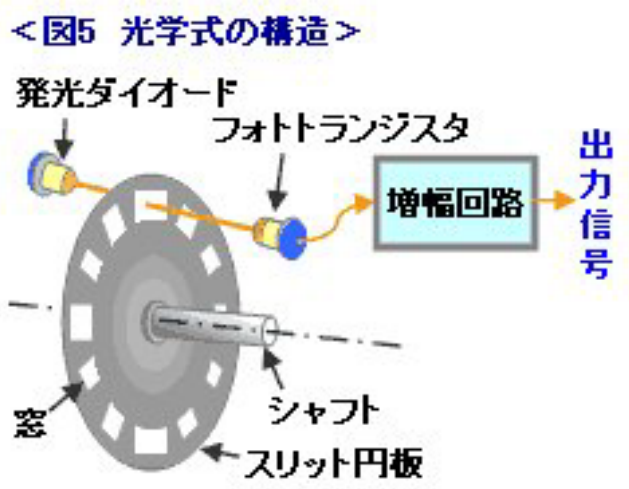
ギアボックスの組み立て

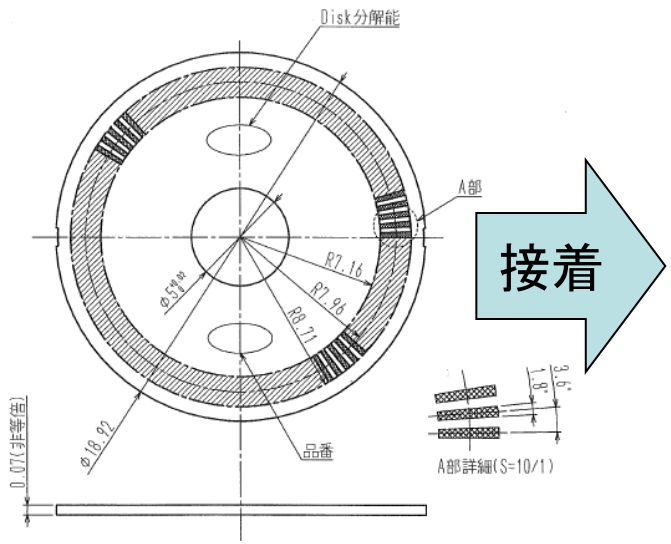
- 47.6:1を選択して組み立てます



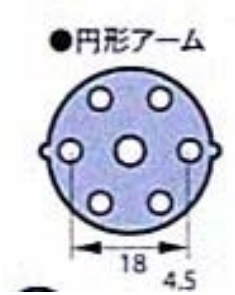
光学式エンコーダ

- 外周部に窓を持つスリット円板と光電検出装置をそなえ、光のオン／オフにより信号を発生させる方式

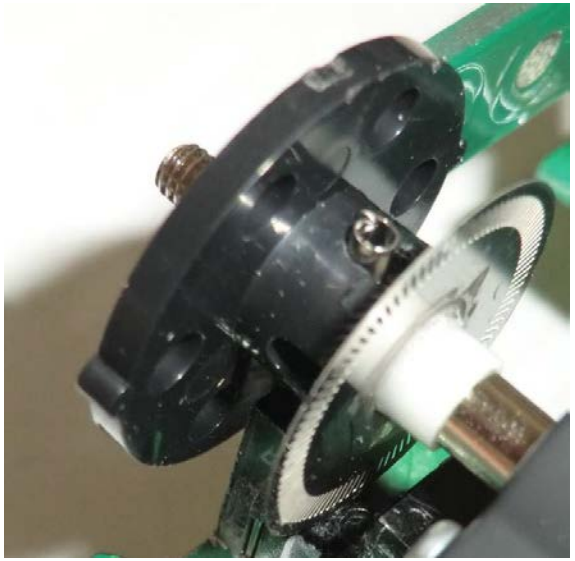




接着



接着



「ロボットの作り方2012」

実習2　ARMマイコンによる
フィードバック制御

6月17日 11:10-12:00

日本工業大学　滝田　謙介

「ロボットの作り方2012」

実習３　鉄棒ロボットの制御

6月17日 14:00-16:30

日本工業大学　滝田　謙介

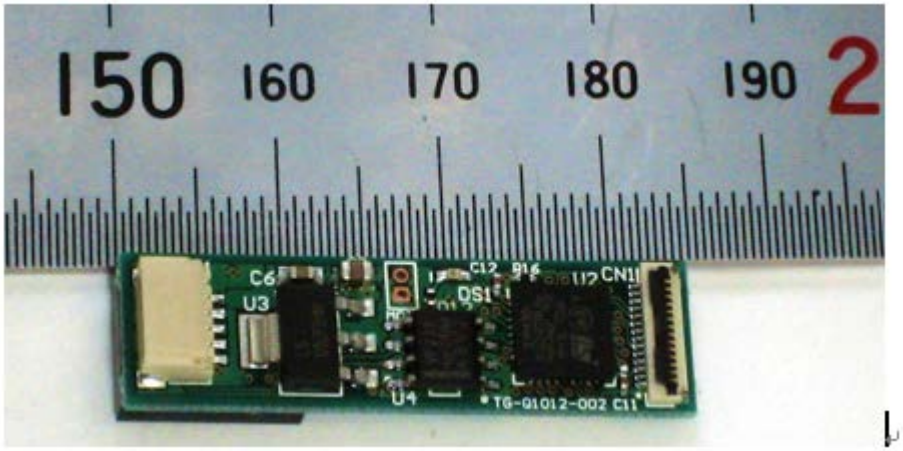
STM32

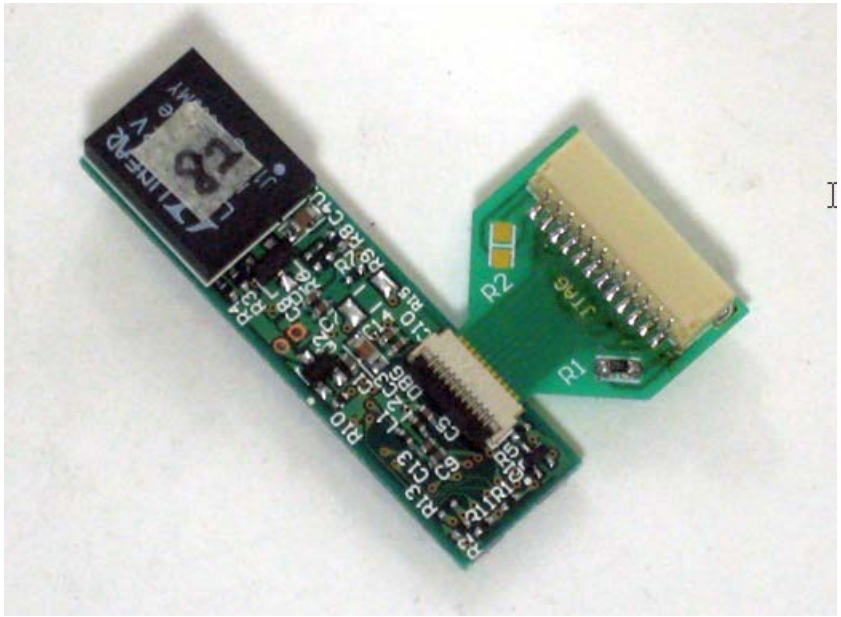
- CORTEX-M3コアを搭載したマイコン
- モータ制御を始め、さまざまなロボットの制御要素として使用可能。

STM32の使用例

先端基板 TG-Q1101-002

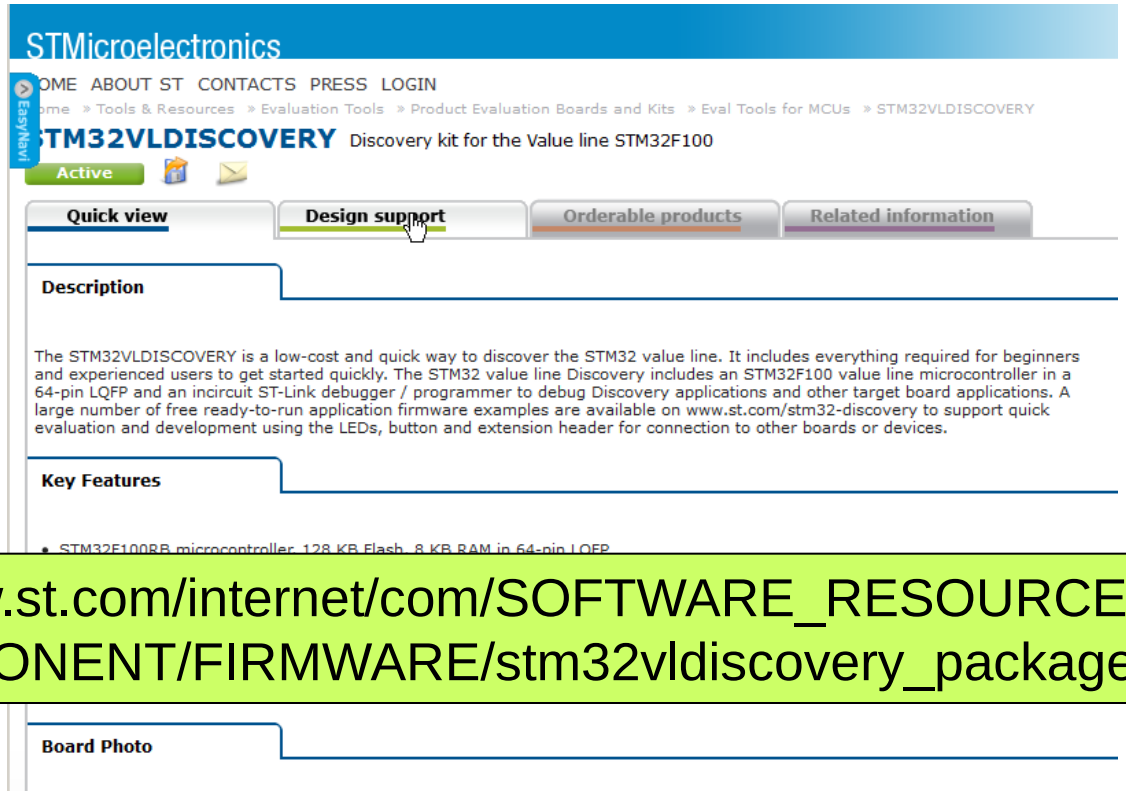
- CPU : STMicroelectronics STM32F103T8U6
- クロック : 64MHz (8MHz内蔵クロックから生成)
- 電源電圧･電流: 3.3V 1A(max) (入力 3.6V～36V)
- 通信 : CAN(CAN2.0B準拠)
- 大きさ : 35mm × 10mm





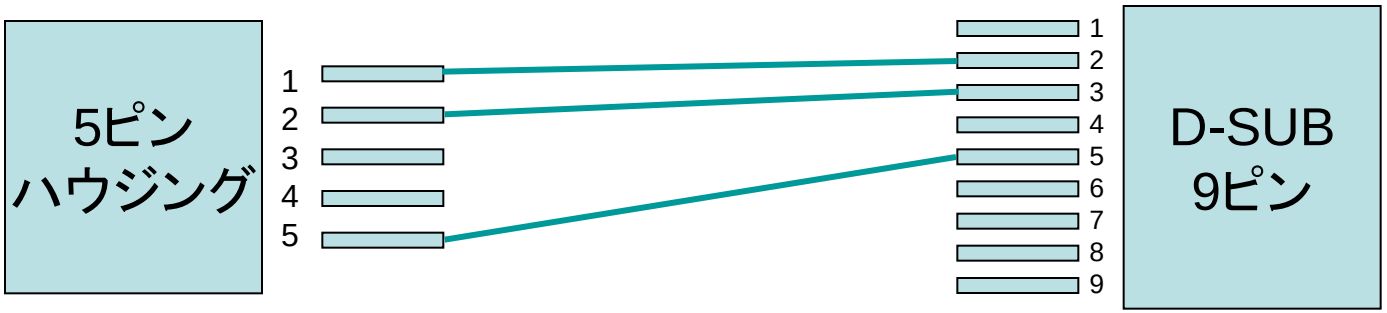
ARMマイコンによるソフトウェア開発

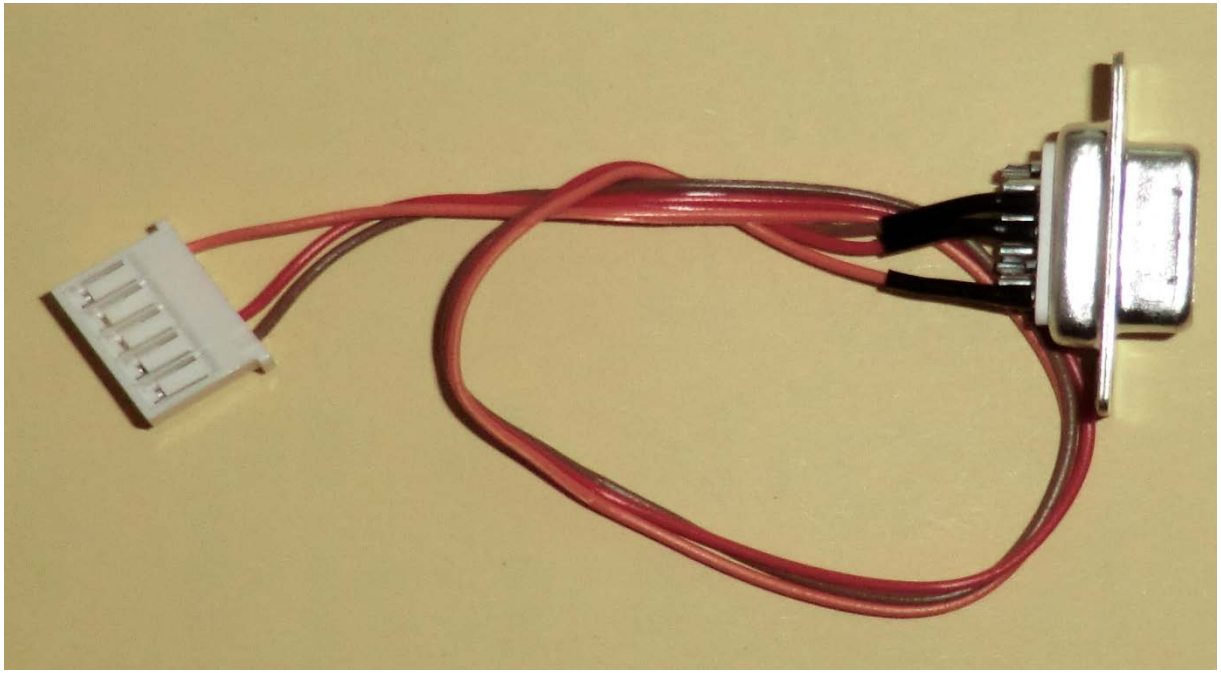
- STM32VLDISCOVERYサンプルファームウェアを使って、STM32マイコンのソフトウェア開発について解説



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

シリアルケーブル





AcRobotサンプルプログラム

- サンプルプログラムを複数ご用意していますのでそちらを下記からダウンロードして下さい。

無線LAN で SSID ROBOT のルータに接続して、下記の共有フォルダーを確認してください。

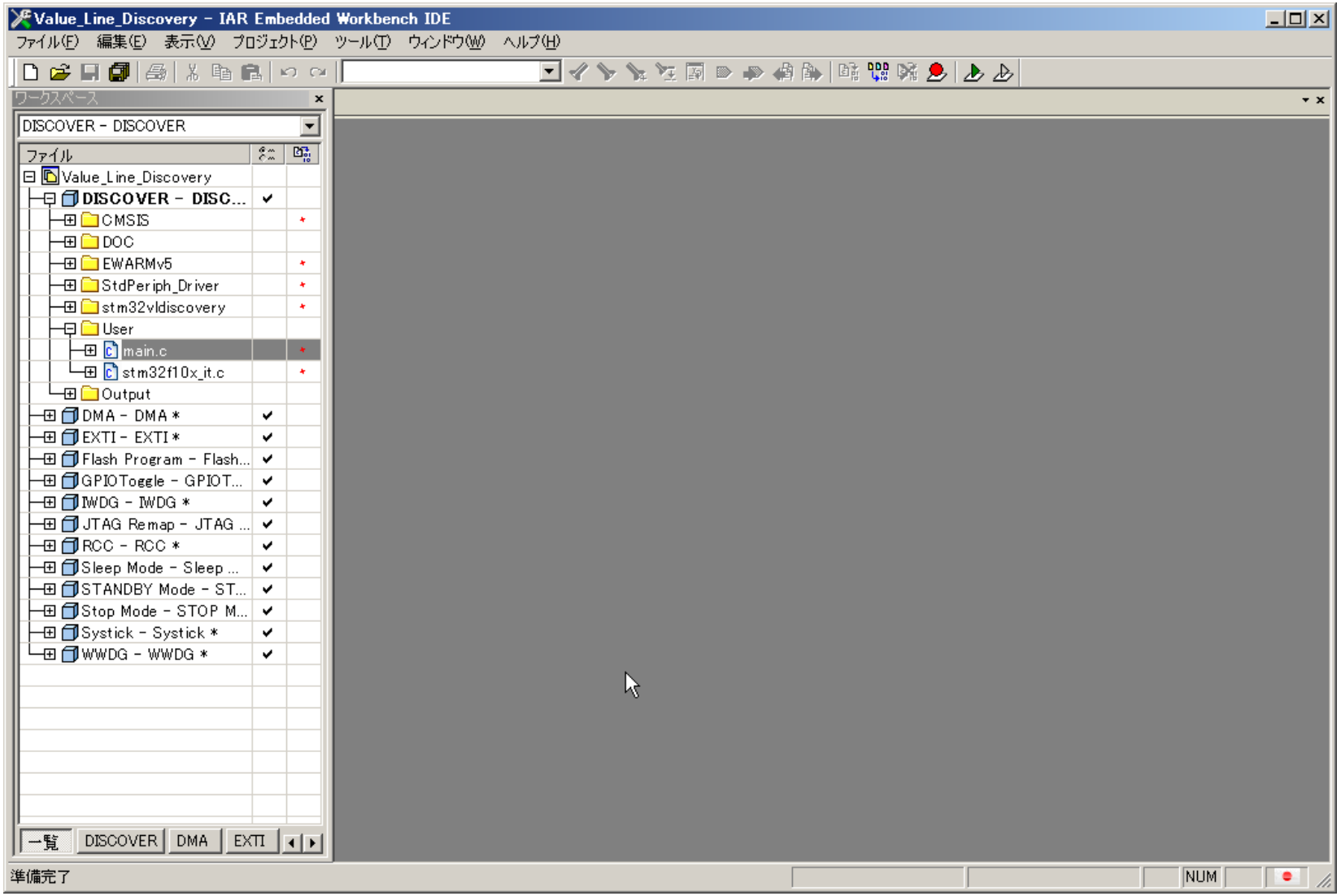
¥¥192.168.11.1¥disk1_pt1

stm32vldiscovery_package.zipをご自身のPCにコピーし、展開してください。

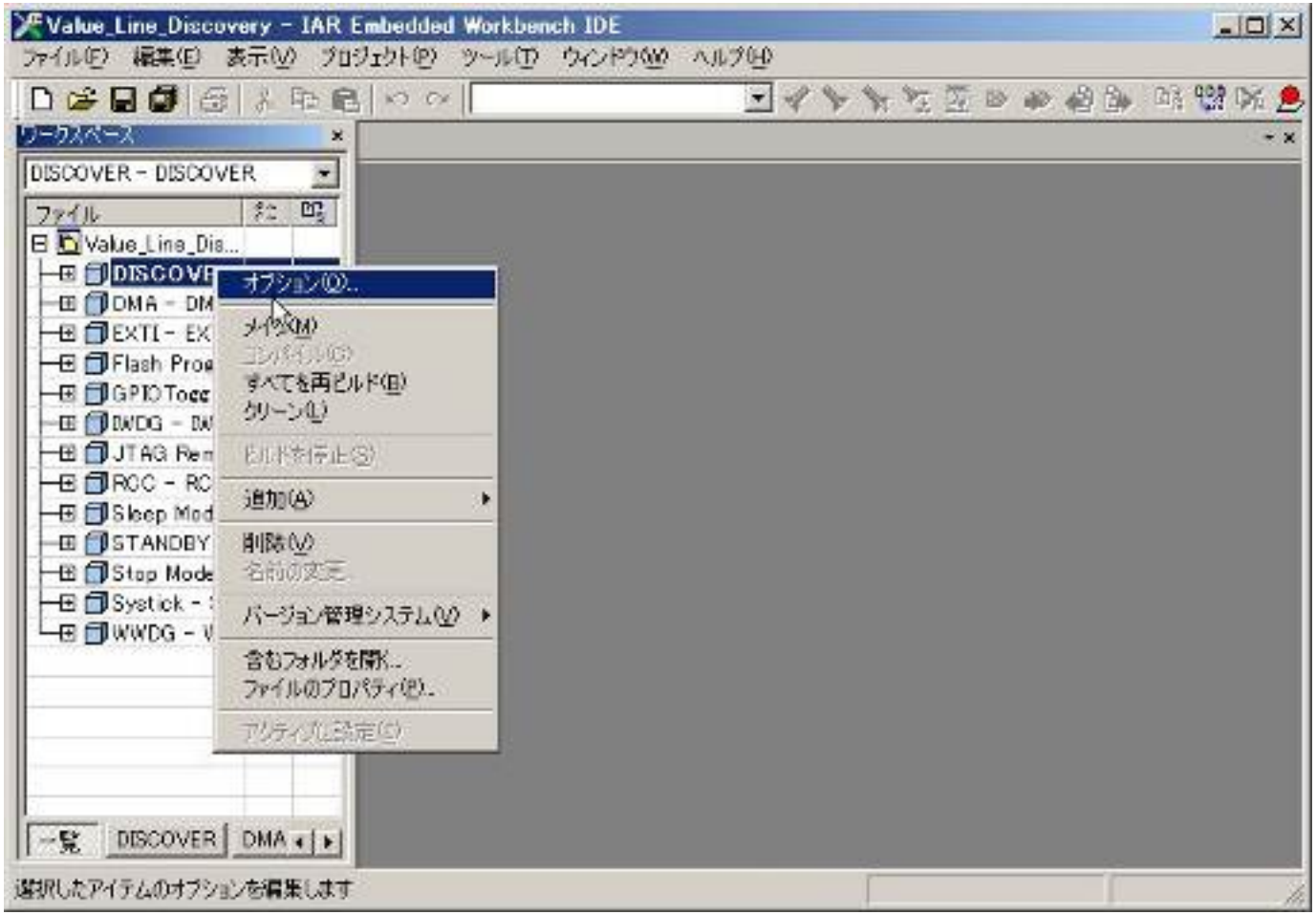
- an3268というディレクトリ中のstm32vldiscovery_packageを、開発用ワーキングディレクトリ(たとえば、C:¥STM32_WORKSPACE など）にコピーする。
- ワーキングディレクトリ中の以下のファイルをEWARMを使って読み込む。

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

サンプルワークスペースの読み込み

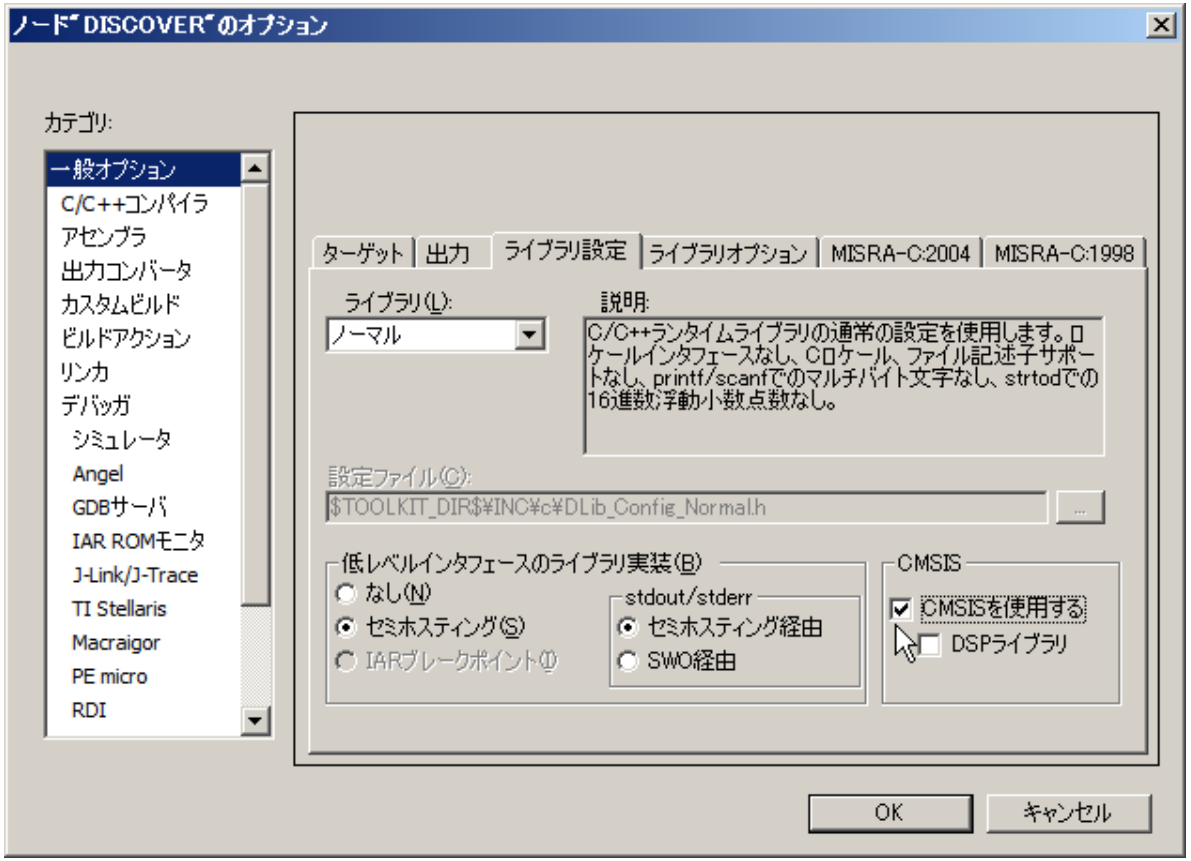


オプションの変更


Value_Line_Discovery - IAR Embedded Workbench IDE
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ツール(T) ウィンドウ(W) ヘルプ(H)
ワークスペース
DISCOVER - DISCOVER
ファイル
Value_Line_Dis...
DISCOVER
DMA - DM
EXTI - EX
Flash Proc
GPIOTogg
IWDG - IW
JTAG Rem
RCC - RC
Sleep Mod
STANDBY
Stop Mode
Systick -
WWDG - W
オプション(O)...
メイク(M)
コンパイル(C)
すべてを再ビルド(B)
クリーン(L)
ビルドを停止(S)
追加(A)
削除(V)
名前の変更
バージョン管理システム(V)
含むフォルダを開く...
ファイルのプロパティ(P)...
アクティブに設定(C)
一覧
DISCOVER
DMA
選択したアイテムのオプションを編集します

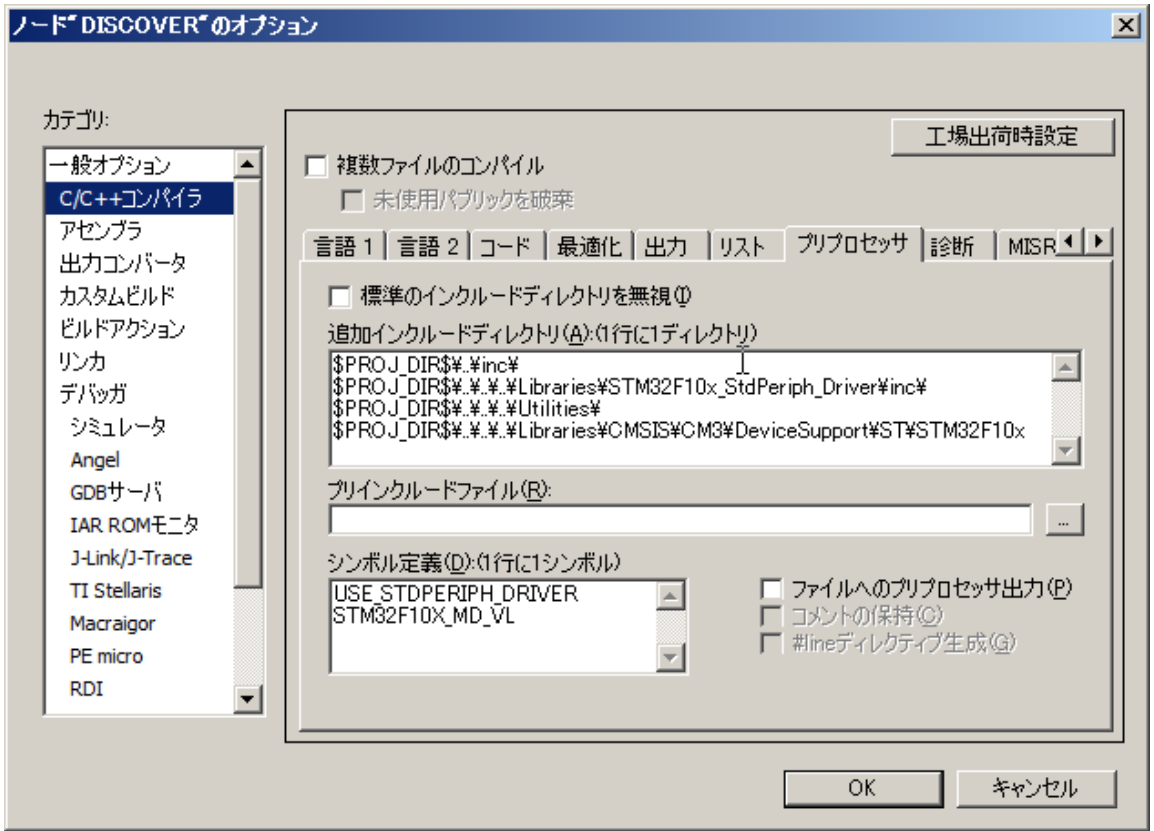
CMSISを設定する

- 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

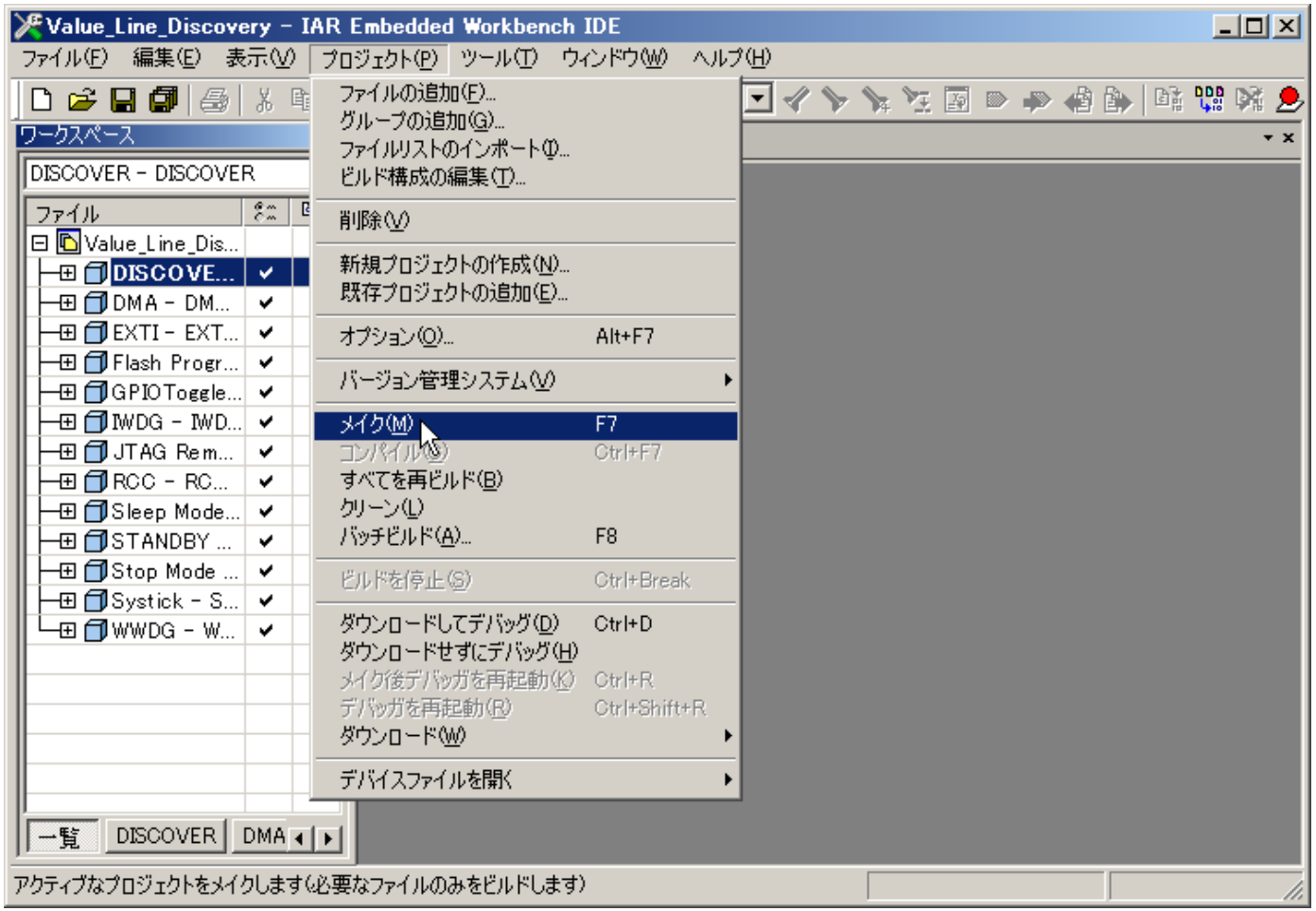


エントリの削除

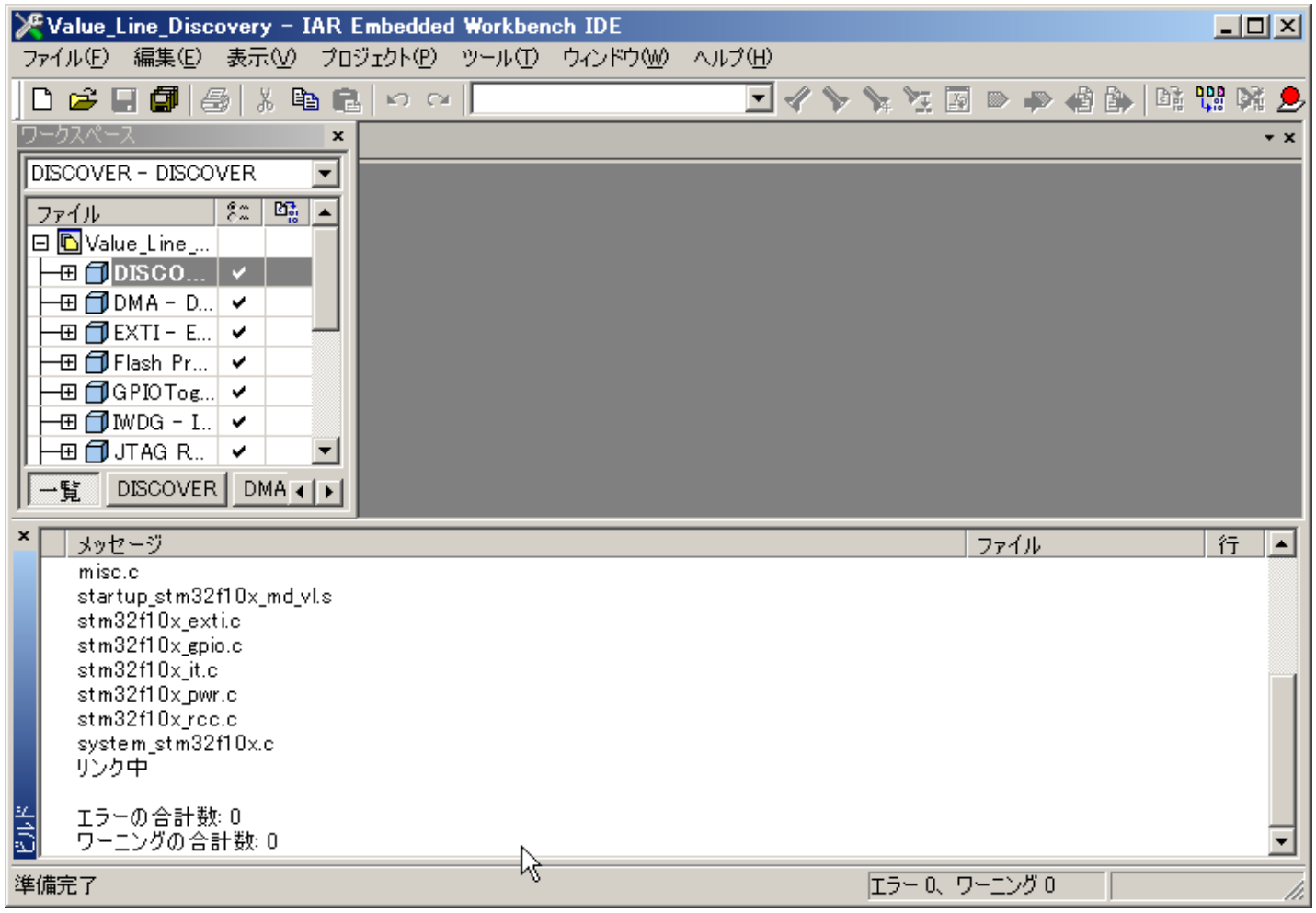
- C/C++コンパイラのプリプロセッサ から $PROJ_DIR$¥..¥..¥..¥Libraries¥CMSIS¥CM3¥CoreSupport¥を削除



メイクの実行


Value_Line_Discovery - IAR Embedded Workbench IDE
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ツール(T) ウィンドウ(W) ヘルプ(H)
ファイルの追加(F)...
グループの追加(G)...
ファイルリストのインポート(I)...
ビルド構成の編集(T)...
削除(V)
新規プロジェクトの作成(N)...
既存プロジェクトの追加(E)...
オプション(O)... Alt+F7
バージョン管理システム(V)
メイク(M) F7
コンパイル(C) Ctrl+F7
すべてを再ビルド(B)
クリーン(L)
バッチビルド(A)... F8
ビルドを停止(S) Ctrl+Break
ダウンロードしてデバッグ(D) Ctrl+D
ダウンロードせずにデバッグ(H)
メイク後デバッガを再起動(K) Ctrl+R
デバッガを再起動(R) Ctrl+Shift+R
ダウンロード(W)
デバイスファイルを開く
ワークスペース
DISCOVER - DISCOVER
ファイル
Value_Line_Dis...
DISCOVE...
DMA - DM...
EXTI - EXT...
Flash Progr...
GPIOToggle...
IWDG - IWD...
JTAG Rem...
RCC - RC...
Sleep Mode...
STANDBY ...
Stop Mode ...
Systick - S...
WWDG - W...
一覧
DISCOVER
DMA
アクティブなプロジェクトをメイクします(必要なファイルのみをビルドします)

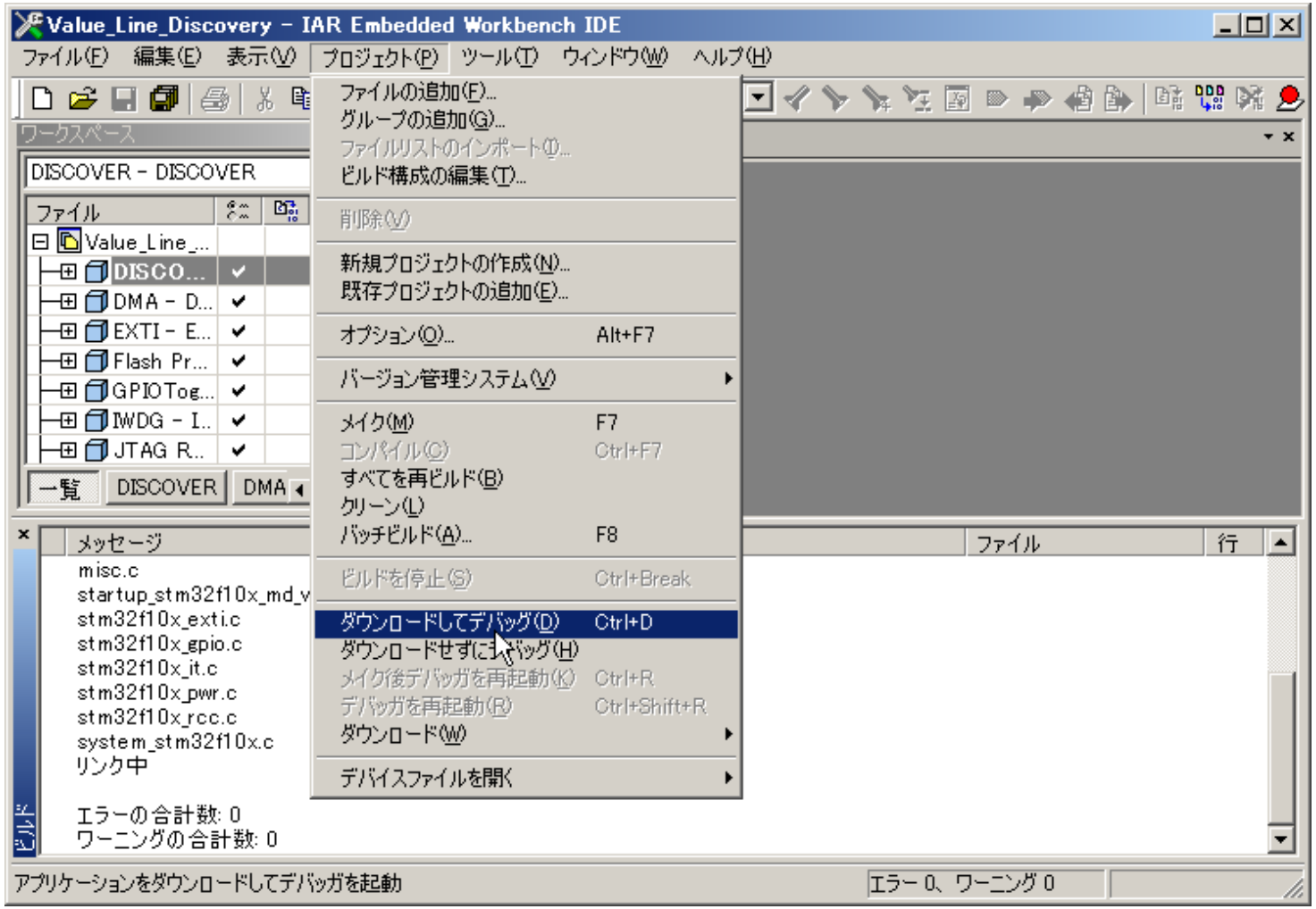
メイクの成功


Value_Line_Discovery - IAR Embedded Workbench IDE
ファイル(F) 編集(E) 表示(V) プロジェクト(P) ツール(T) ウィンドウ(W) ヘルプ(H)
ワークスペース
DISCOVER - DISCOVER
ファイル
Value_Line_...
DISCO...
DMA - D...
EXTI - E...
Flash Pr...
GPIOTog...
IWDG - I...
JTAG R...
一覧
DISCOVER
DMA
メッセージ
ファイル
行
misc.c
startup_stm32f10x_md_vl.s
stm32f10x_exti.c
stm32f10x_gpio.c
stm32f10x_it.c
stm32f10x_pwr.c
stm32f10x_rcc.c
system_stm32f10x.c
リンク中
エラーの合計数: 0
ワーニングの合計数: 0
ビルド
準備完了
エラー 0、ワーニング 0

デバッグ

1. STM32VLDiscoveryをPCとUSBケーブルで接続する。
2. ST−LinkがPCに認識される。
3. EWARM上で、ダウンロードを実行
4. プログラムを実行。
5. エラーがあれば、ソースファイルを変更→メイクを実行し、ダウンロード、プログラムの実行を繰り返す。

ダウンロードしてデバッグ

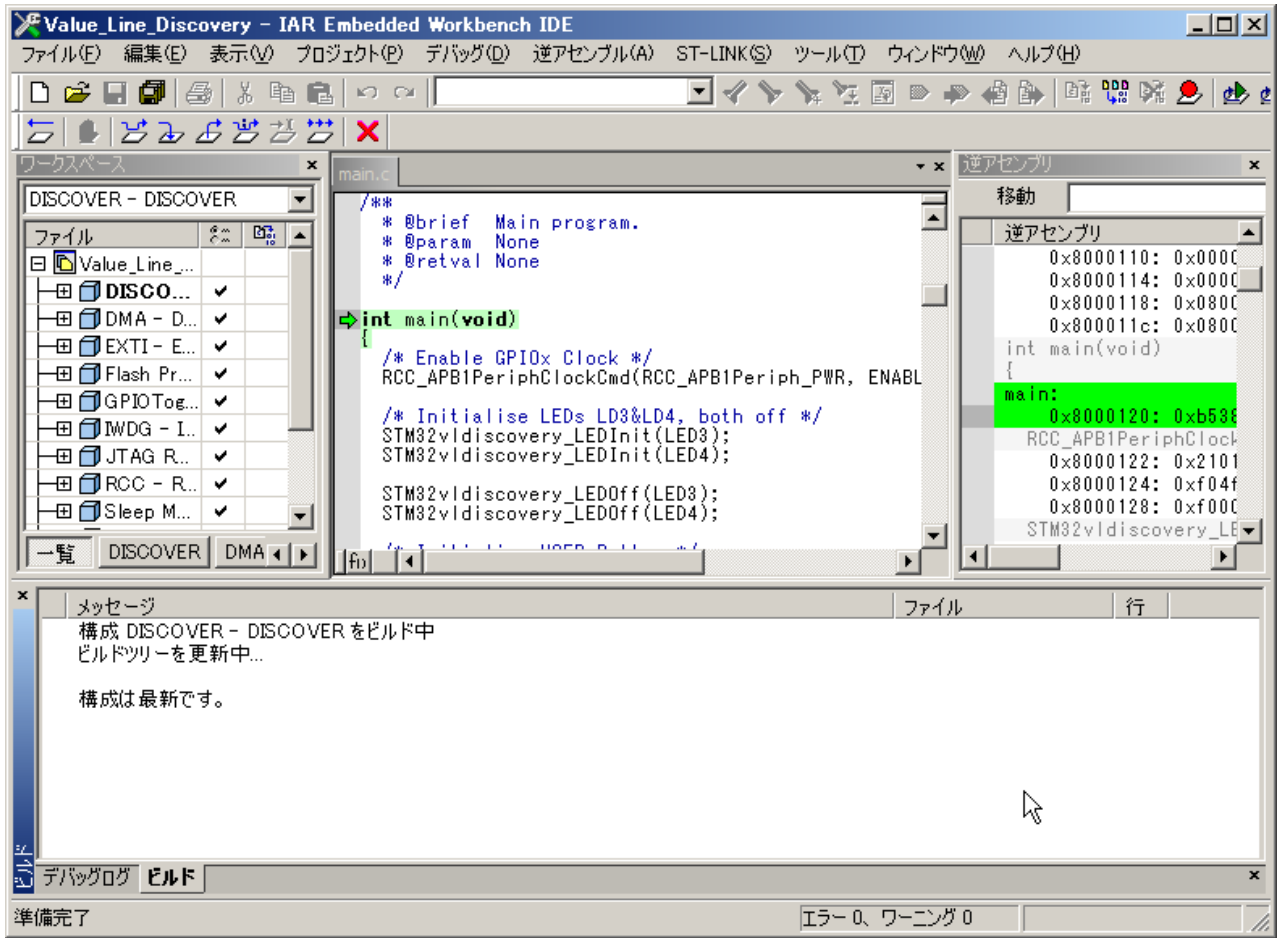


プログラムのデバッグ実行画面

- F5キーを押して、プログラムを実行

USERボタンを押す

青色LEDが点灯して
緑色LEDの点滅周期が
変化



マイコン制御の流れ

- 一般にOSを使用しないマイコンプログラムでは、以下の流れで、処理が行われる。

各種周辺機器の設定

↓

処理の開始

↓

while文で無限ループ

割込でイベント処理

周辺機器の初期化

```
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);.
```

.
.
.

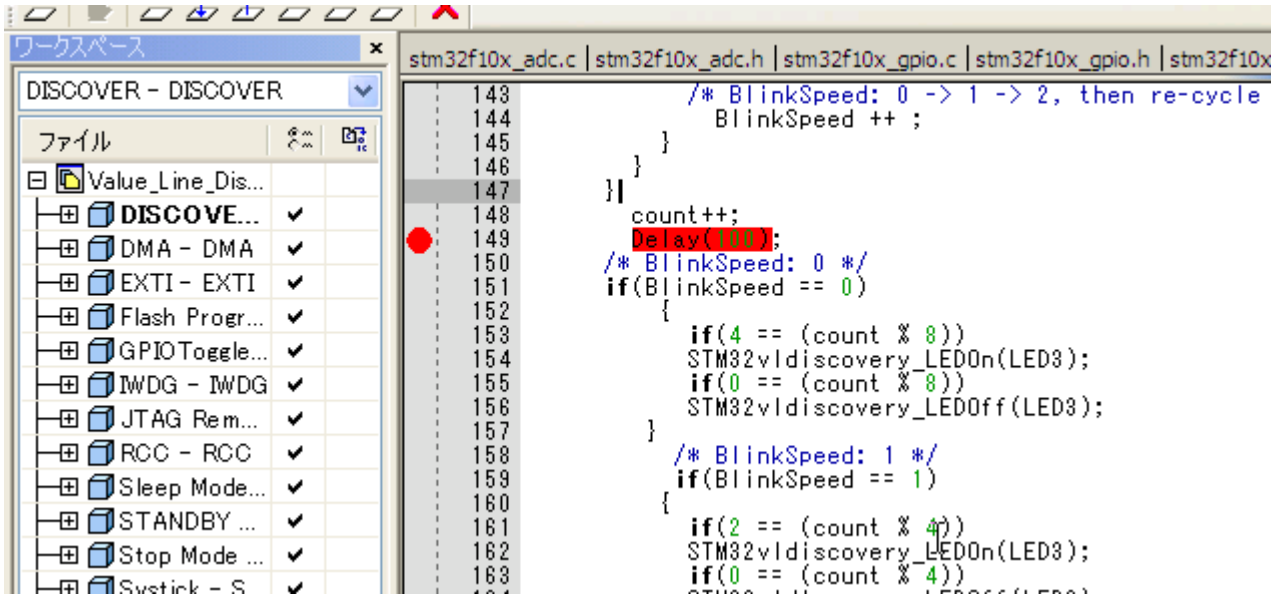
無限ループ

```
114.  /* main while */
115.  while(1)
116.  {
117.    if(0 == STM32vldiscovery_PBGetState(BUTTON_USER))
118.      {
119.        if(KeyState == 1)
120.        {
121.          if(0 == STM32vldiscovery_PBGetState(BUTTON_USER))
122.          {
```

- while(1)で無限ループに入り、ボタンを読み取り、処理を実行する。

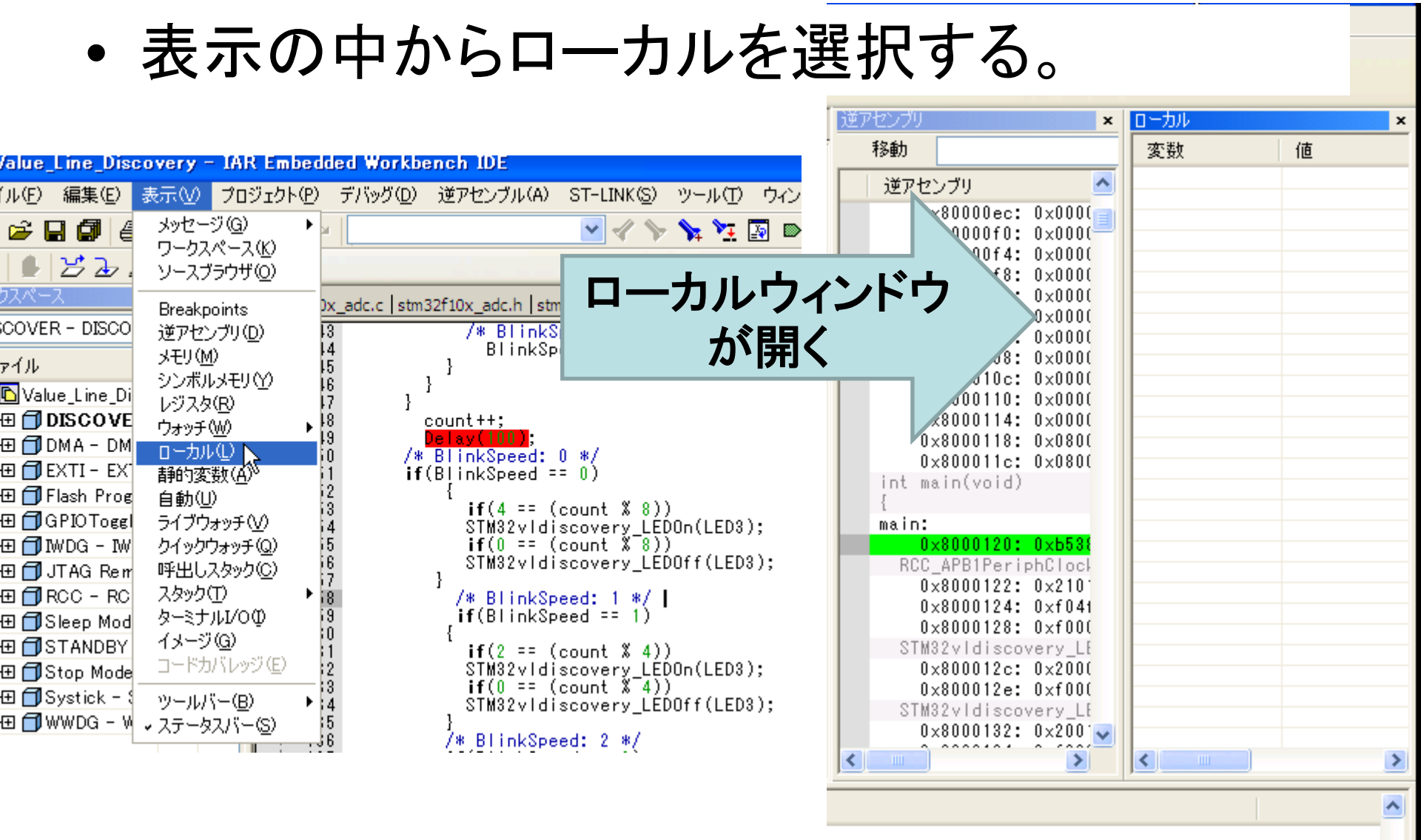
実際のデバッグ

- 実行中の変数の状態を監視するためには，ブレークポイント(プログラムを一時停止する場所)を設定します。ブレークポイントは，ソースコードの停止したい分の左側ダブルクリックします。



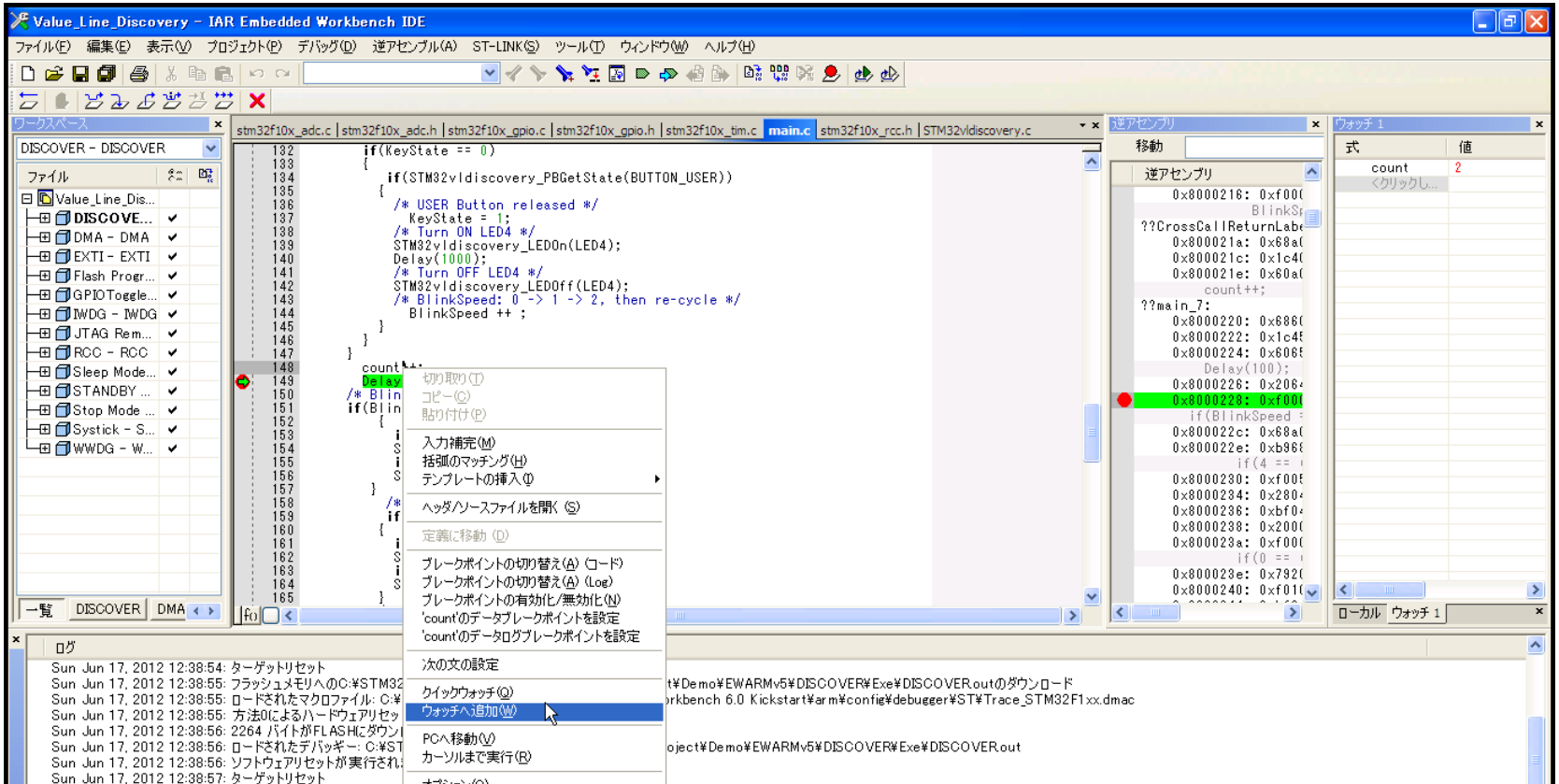
ローカル変数の監視

- 表示の中からローカルを選択する。



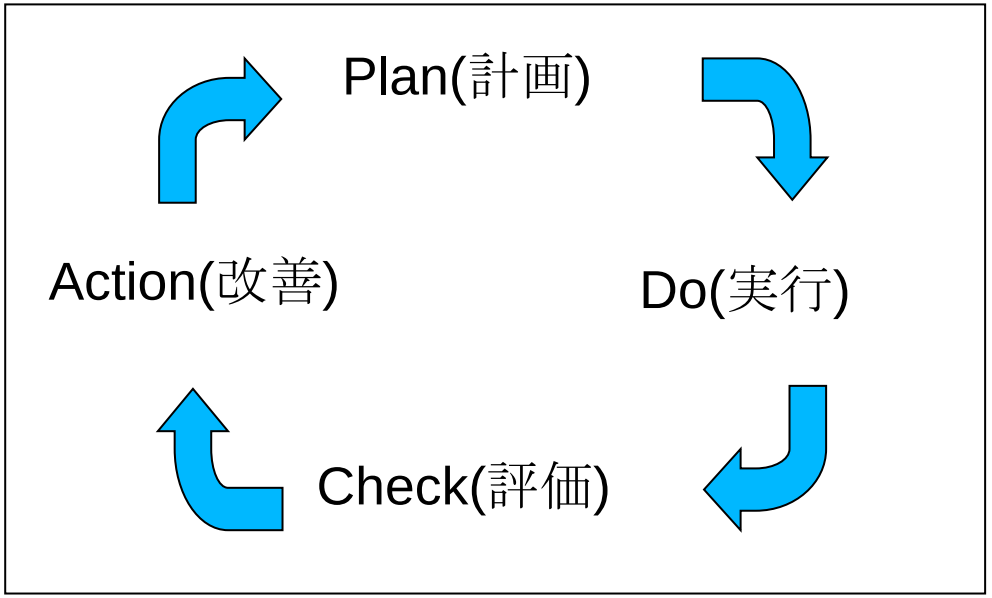
ウォッチの指定

- count変数を監視対象にするために,count変数の上で右クリックをして,ウォッチに指定する。
- [F5]キーでプログラムすると先ほど指定したBPまで実行されてcount変数が更新される。



プログラムが動くようになるまで

- 基本的には プログラム→実行→動作確認→改良→プログラム→・・・ という流れを繰り返し、動作する状態にする



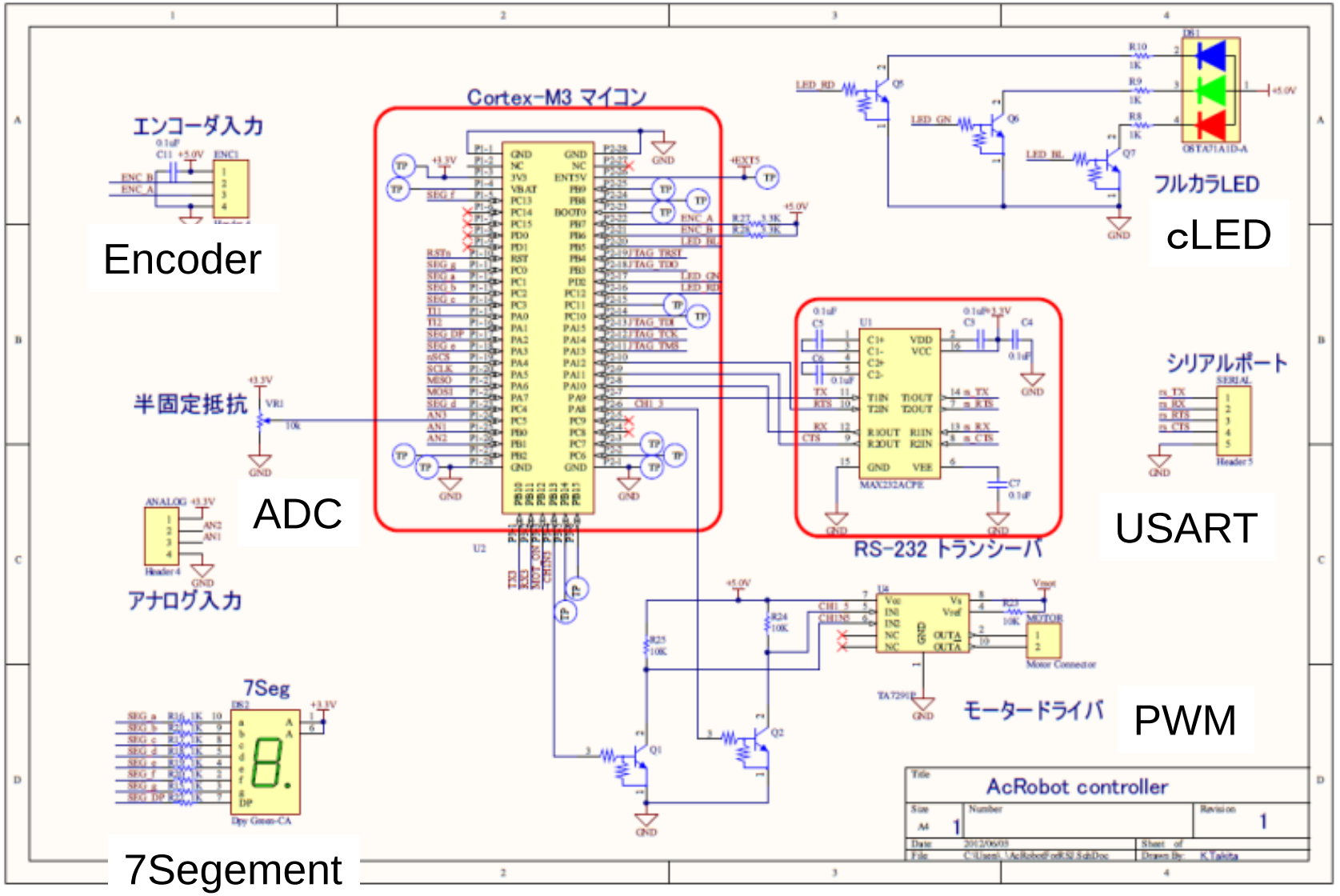
AcRobotサンプルプログラム

- サンプルプログラムを複数ご用意していますのでそちらを下記からダウンロードして下さい。

無線LAN で SSID ROBOT のルータに接続して、下記の共有フォルダーを確認してください。

¥¥192.168.11.1¥disk1_pt1

サンプルプログラム


Cortex-M3 マイコン
エンコーダ入力
Encoder
半固定抵抗
ADC
アナログ入力
7Seg
7Segement
フルカラLED
cLED
シリアルポート
USART
RS-232 トランシーバ
モータードライバ
PWM
AcRobot controller

サンプルプログラム

- AcRobot_7SEGMENT: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.

- AcRobot_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。

AcRobot_7segment.h

7セグメントLEDを点灯するための関数を以下に示します。

- void Acrobot_7SEG_Init(void);

IOポートの初期化をする関数。

- void Acrobot_7SEG_LEDON(uint8_t ch);

それぞれのLEDを点灯するための関数

- void Acrobot_7SEG_NO(uint8_t num,uint8_t dp);

0～9、a～fを表示するための関数。上記のAcrobot_7SEG_LEDONにそれぞれの文字に使用するchを指定。

void Acrobot_7SEG_Init(void)

- void Acrobot_7SEG_Init(void)
- {
- GPIO_InitTypeDef GPIO_InitStructure;

IOポートにクロックを供給して，
IOポートを使える用にする。

- RCC_APB2PeriphClockCmd(ACROBOT_7SEG_CLK, ENABLE);
-
- GPIO_InitStructure.GPIO_Pin = ACROBOT_7SEG_A_PIN;
- GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
- GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
- GPIO_Init(ACROBOT_7SEG_A_PORT, &GPIO_InitStructure);

GPIOピン・モード・スピードを指定して、該当ピンを初期化

AcRobot_Utils.h

- void TIM2_delayus(uint16_t us);

TIM2 (タイマー2)を使用してマイクロ秒の時間を計測する関数。

- void TIM2_delayms(uint16_t ms);

TIM2 (タイマー2)を使用してミリ秒の時間を計測する関数。

TIM2_delayus(uint16_t us)

TIM2 (タイマー2)を使用してマイクロ秒の時間を計測する関数。

```
void TIM2_delayus(uint16_t us)
{
  TIM_TimeBaseInitTypeDef TIM_TimeBaseStruct;

  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2 , ENABLE);

  if(us>1000)us = 1000;

  TIM_TimeBaseStruct.TIM_Period = 24*us-1; // 1us
  TIM_TimeBaseStruct.TIM_Prescaler = 0;
  TIM_TimeBaseStruct.TIM_ClockDivision = 0;
  TIM_TimeBaseStruct.TIM_CounterMode = TIM_CounterMode_Up;
  TIM_TimeBaseStruct.TIM_RepetitionCounter = 0;
  TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStruct);

  TIM_SelectOnePulseMode(TIM2, TIM_OPMode_Single);
  TIM_SetCounter(TIM2,0);
  TIM_Cmd(TIM2, ENABLE);
  while (TIM_GetCounter(TIM2)){};

  TIM_Cmd(TIM2, DISABLE);
}
```

TIM2にクロックを供給して，TIM2を起動する。

CPUが周波数24MHzで動作している→ 24クロック数えれば、1us

クロックを数え終わるのを待つ。

サンプルプログラム

- AcRobot_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。

AcRobot_USART.h

- void Acrobot_USARTInit(void);

USART(非同期・同期シリアルトランシーバ)を初期化する。

- void Acrobot_USART_SendChar(char c);

一文字分のデータを送信する

- void Acrobot_USART_SendString(char *str);

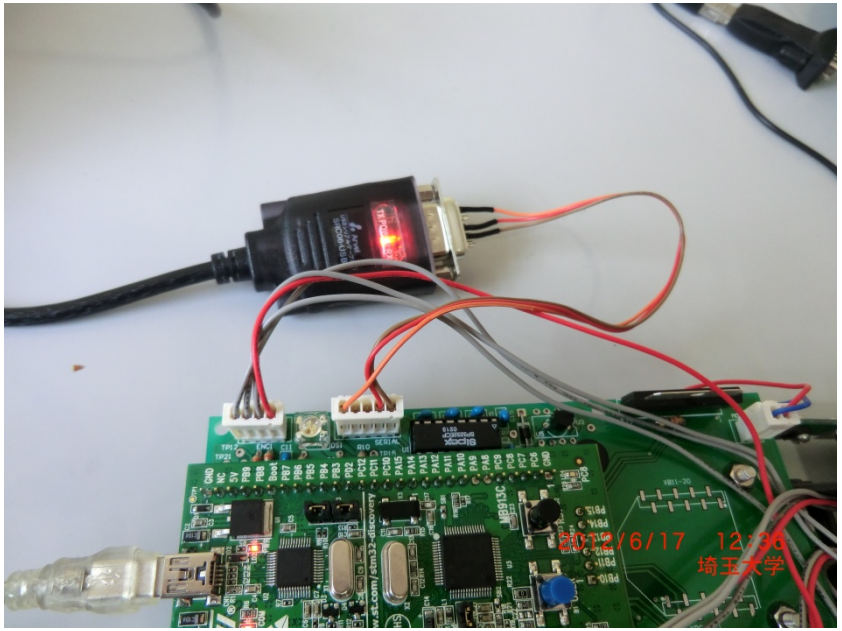
Acrobot_USART_SendCharを使用して文字列を送信する。

- int Acrobot_USART_Printf(const char *fmt, ...);

書式付きprintfを実装したもの

シリアルケーブルの接続

- SERIALにDSUB9ピンケーブルを接続し、D-SUB9ピンをUSBシリアルケーブルに接続する



サンプルプログラム

- AcRobot_ADC: 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。
- 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。

AcRobot_ADC.h

- void Acrobot_ADC_Init(void);

ADコンバータモジュールを初期化する。

- uint16_t Acrobot_ADC_Read(uint8_t ch);

指定したチャンネルのAD変換した値を取得する。

サンプルプログラム

- AcRobot_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。

AcRobot_Encoder.h

- void Acrobot_QEI_Init(void);

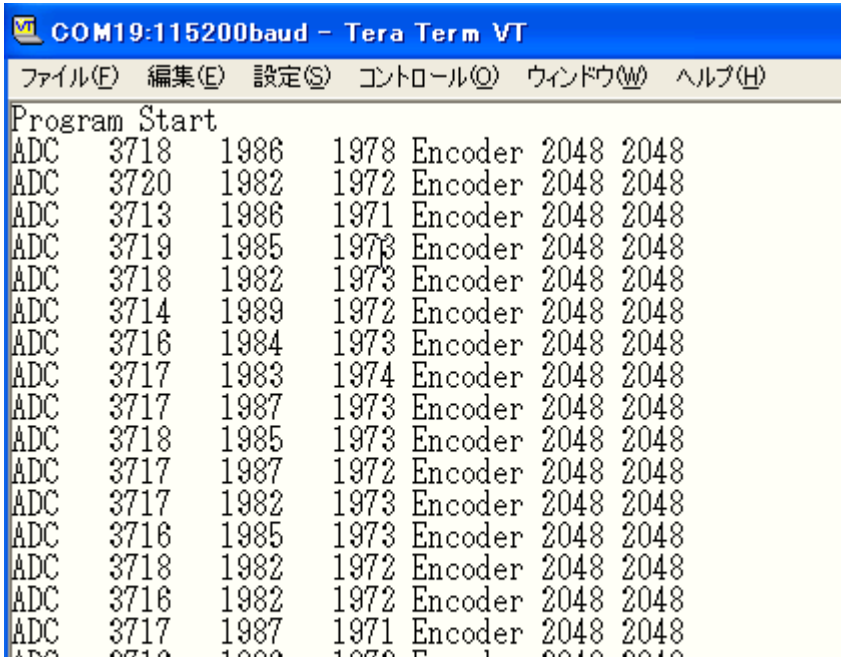
TIM4のカウンタ機能を設定。

- uint32_t Acrobot_QEI_GetCount(void);

エンコーダのカウント値を取得。

シリアル出力結果

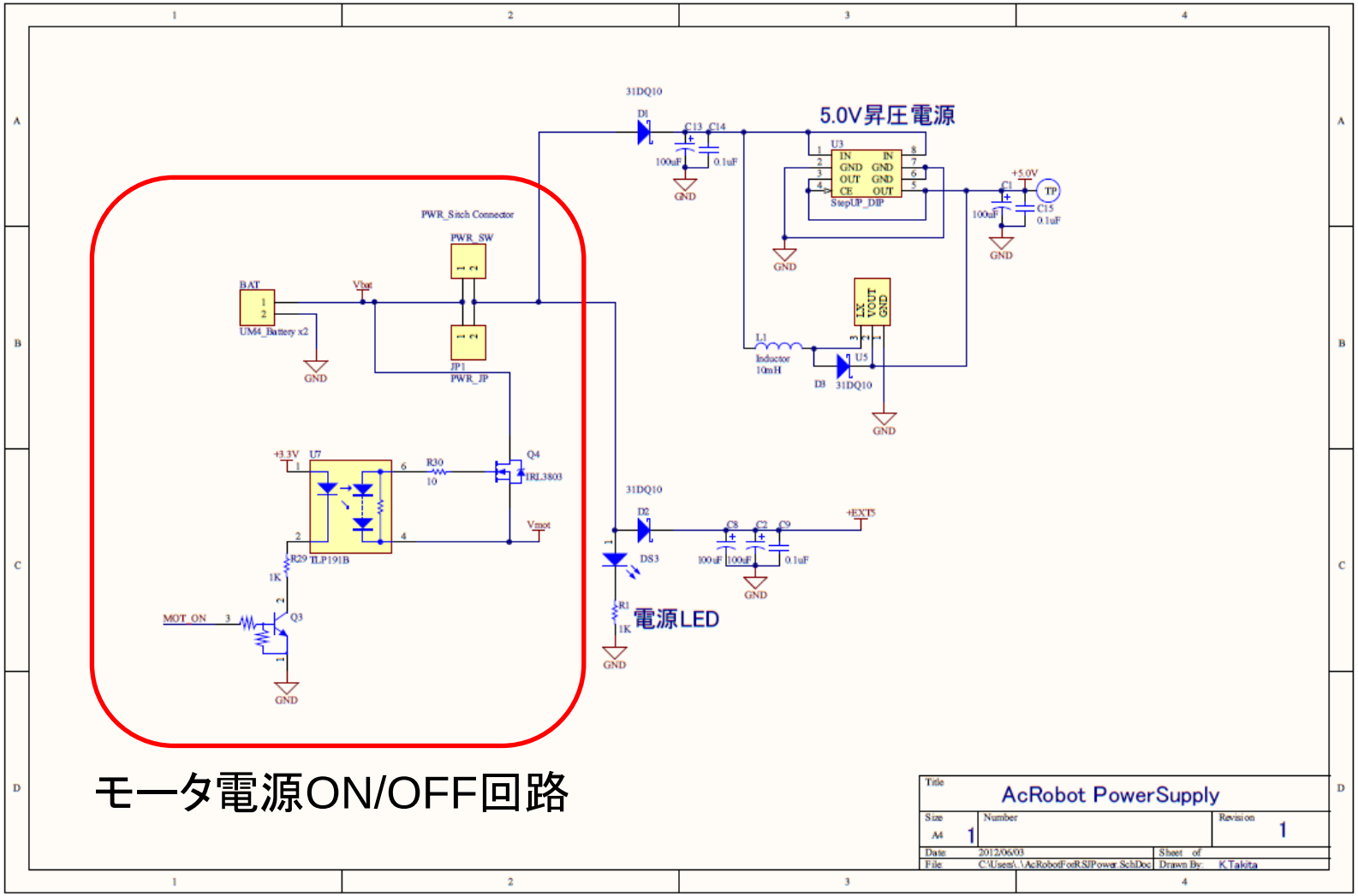
- 最初の3つがAD変換した値
- (基板上の半固定抵抗、ジャイロセンサの値x2)
- Encoder値(Acrobot_QEI_GetCountで取得したもの、TIM4のカウント値)



サンプルプログラム

- AcRobot_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。AcRobot_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みます。

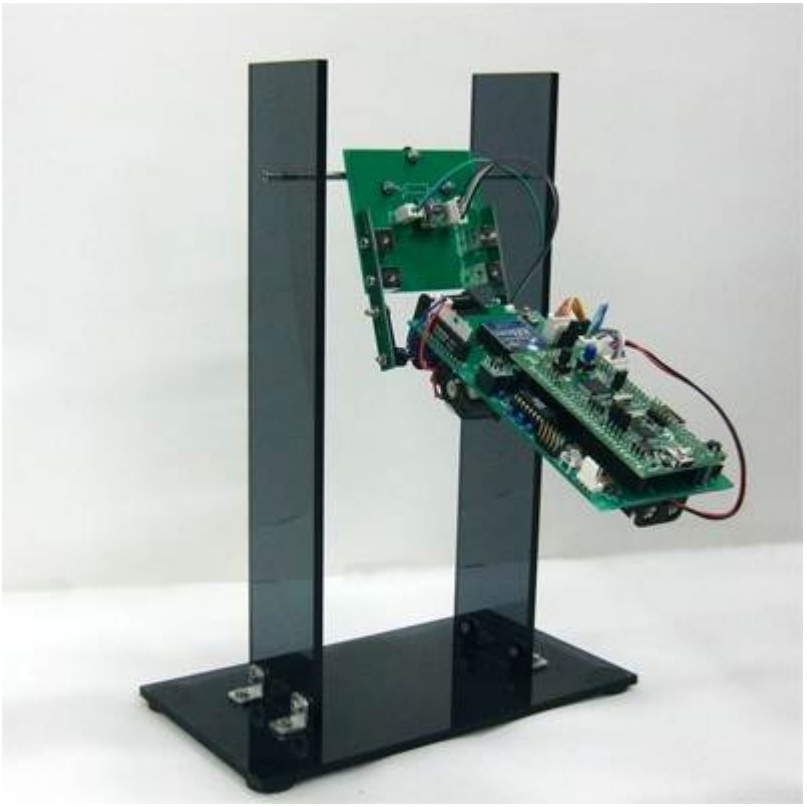
電源回路



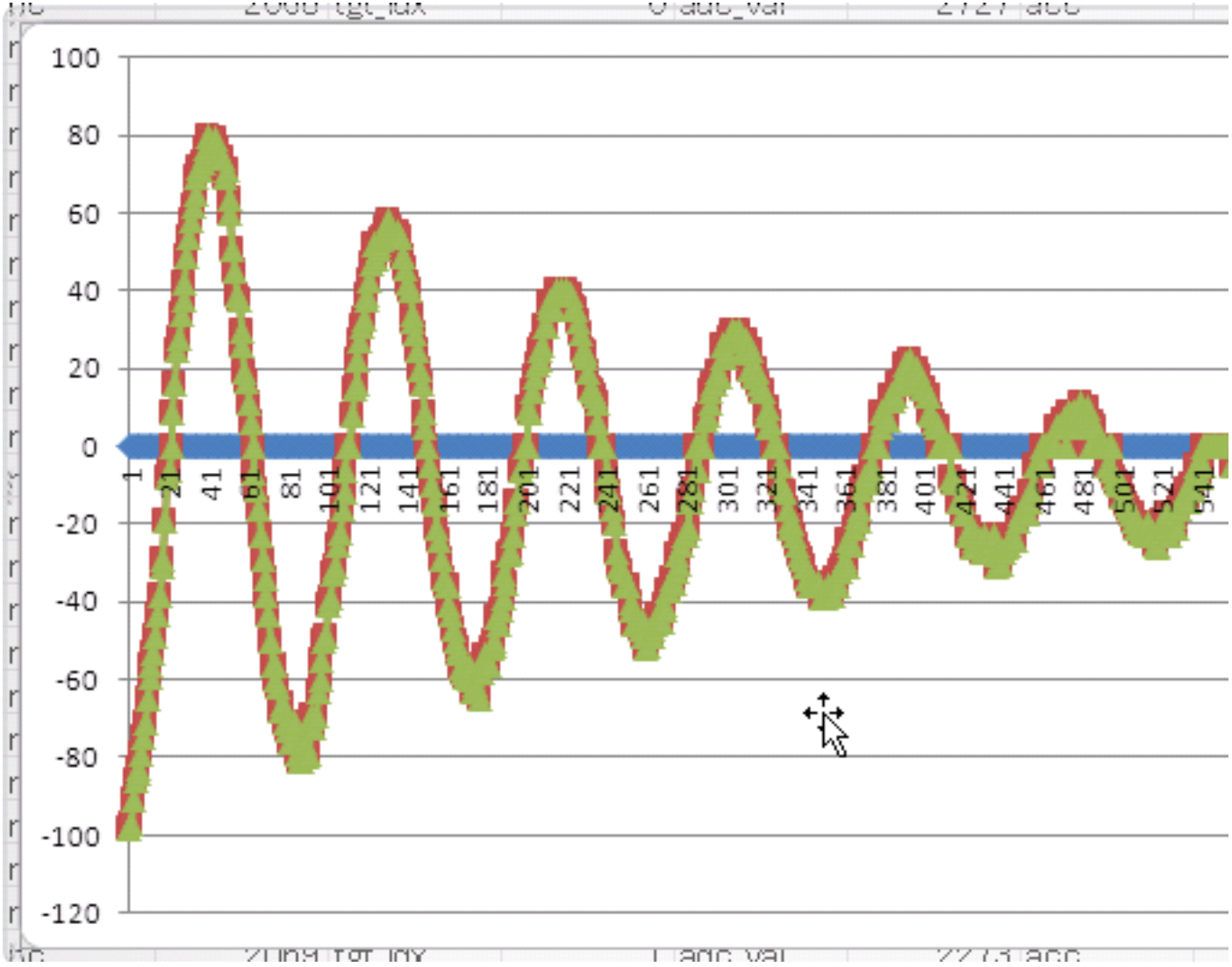
モータ電源ON/OFF回路

本セミナーで組み立てるキット

- STM32VLDISCOVERY を搭載した鉄棒ロボット。
- ギア付きモータx1を搭載
- 光学式エンコーダ、フルカラーLED、7セグメント、ジャイロセンサなどを搭載
- PWM制御によるモータ制御実習用に開発



摩擦がある振り子なので、いずれ止まってしまう。



サンプルプログラム

- AcRobot_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。
- ジャイロセンサで角速度を計算し、速度と加速度の向きが等しいとき(加速しているとき)に体を伸ばして、異なるときに体を曲げる。