

試作検討 4

創造設計第二 TA：松本 浩, 草野 正巳

平成 20 年度 11 月 5, 6 日

1. はじめに

今回の試作検討では、既製品のセンサーやモジュールの使い方を学ぶことを目的とする。

本日使用するセンサは 200mm 距離センサである。センサからのアナログ出力を、マイコンボードの A/D 変換入力を使って計測する。また、センサからの出力によって、ポート割り込みがおきるようにする。

2. 反射型距離センサ (200mm 光センサ)

2.1 フォトトランジスタ

フォトトランジスタは光によって駆動するトランジスタであり、近接センサやロータリエンコーダのパターン読み取りなど、光センサとしてさまざまな応用が考えられる。

Fig.1 のエミッタ出力の回路では入射光と同位相の出力が得られ、Fig.2 のコレクタ出力では入射光と逆位相の出力が得られる。

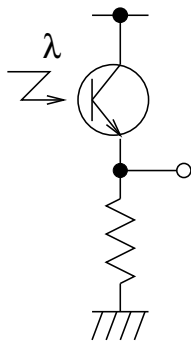


Fig.1: エミッタ出力

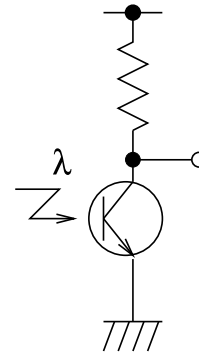


Fig.2: コレクタ出力

また、LED と組み合わせることによってフォトカプラを構成することもできる。Fig.3 の回路では LED の光がフォトトランジスタに入射している場合は 5[V]、入射していない場合は 0[V] の出力となる。

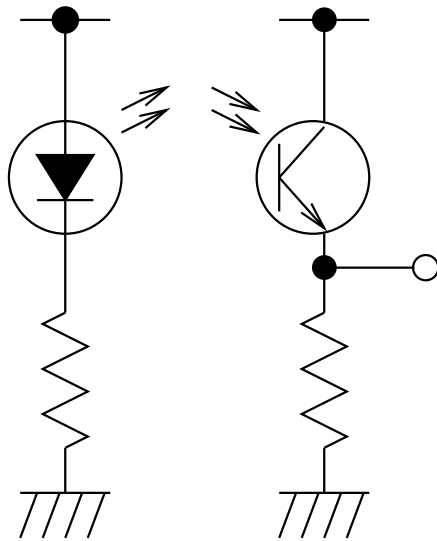
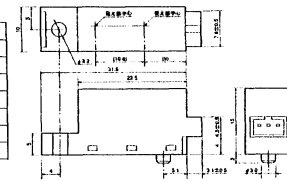


Fig. 3: フォトカプラ回路

2. 基本仕様

電気的特性	
動作温度	-20~+60°C
電源電圧	DC5V±5%
消費電流	11mA(非投光時) 200mA(投光時)
応答速度	1ms
許容照度	外乱光 2000 lx
出力I/f	オープンコレクタ方式
出力信号	物体検知時:OH=L VOL=0.4V以下 物体非検知:OFF=H VOH=4.5V以上

外形寸法



ピン接続

PIN No.	名称	説明
1	VCC	電源DC5V端子
2	VOUT	センサの反応出力端子
3	GND	電源グランド端子



なお、推奨コネクタはIL-S-3S-S2C2-S(日本航空電子製)です。

Fig. 4: 200mm センサ基本仕様 (データシートより)

これを利用すればLEDとフォトトランジスタの間に障害物が入っていないかを検出する光センサとすることが可能である。また、入力信号によりLEDを駆動し、フォトトランジスタ側に外部回路を接続することによって、2つの回路を電気的に絶縁した状態で駆動することも可能である。

2.2 200mm 光センサ (ベストテクノロジー - 社製『BTE019』)

創造設計第2では近接センサ(物体の接近を検出する)用に「200mm 光センサ」が用意されている。これは、赤外線LEDとフォトトランジスタがワンパッケージで組み込まれたもので、機能としては、センサから最大200mm離れた白いもの(光を反射するもの)を検出する。

コネクタ端子は3つあって、電源とGNDを1,3ピン間につなぎ、出力を2ピンから取る。

Fig. 4に基本仕様を示す。なお、BTE019のデータシートは、マシンショップのリンクをたどることによりダウンロード可能である。他のセンサー類も同じである。



注意

当然のことながら、電源とGNDや信号線を間違えると、センサは破壊する。これを防ぐ方法は以下の手順を踏み、一つ一つのステップを確実に行うことである。まず、センサをつなぐ前に、電源回路や外部回路を構成する。そして、センサをつなぐ前に、コネクタのセンサ側入力の電圧を確認する。こうすることで、センサの破壊はほとんど防ぐことができる。



警告

今回の試作設計で扱うモジュールは、高価なものである。これらの取り扱いには十分注意すること。壊してしまった場合には、ロボット製作において使用できなくなる可能性があるので十分注意しながらやること。

2.3 課題

— 200mm センサを使った課題 —

- 配布されている 200mm センサをマイコンに接続し、センサの出力によって LCD の表示が変わるプログラムを作成する。(試作検討 1 の資料を参照するとよい)
- MCU から 200mm センサのつないであるポートをプルアップする。
- 余裕があれば、色や角度などを変化させて 200mm センサの反応の違いをみる。

3. 参考:PSD センサ

創造設計第二では A 類部品として PSD センサ (GP2D12) が配布されている。PSD センサを利用することで対象との距離を計測することができる。

3.1 PSD センサの概要

PSD はセンサ自身が赤外線を発し、反射光の戻って来る位置を測定することで距離情報を計測する測距センサである。PSD は反射光の強度ではなく位置を測定に用いるため、反射物の色、反射率の影響を受けにくいという特徴がある。また、物体の接近だけでなく、物体への距離を数値的に計測することができる。GP2D12 のピン配置は、Fig.6 のようになっている。

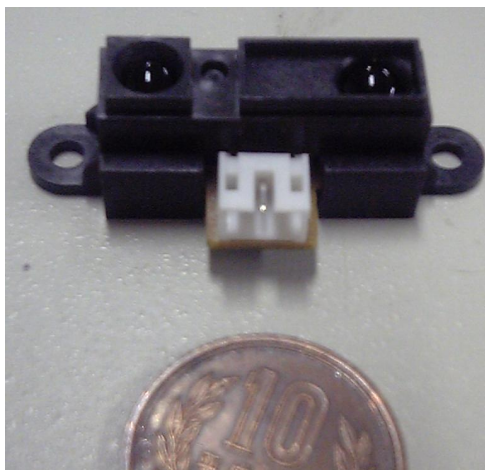
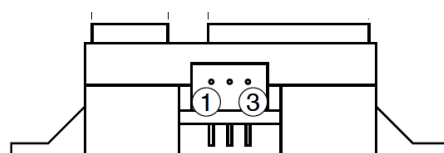


Fig. 5: GP2D12(PSD センサ)



Terminal connection

- ① V_O (Signal)
- ② GND
- ③ V_{CC} (5V)

Fig. 6: GP2D12(PSD センサ) のコネクタ

3.2 PSD の出力と A/D 変換

PSD センサからの出力はアナログ出力である。今回配布する PSD センサは距離に対応した電圧を出力する。この電圧を A/D 変換してマイコンボードに取り込む。Fig.7 にデータシートを示す。

Analog Output Voltage vs.Distance to Reflective Object

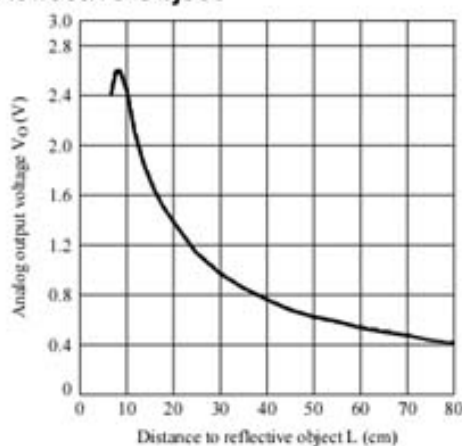


Fig. 7: データシート



注意

PSD センサは個々で特性線図が異なるため、精密な測定をする場合は実験をして個々の特性線図を作成する必要がある。

3.3 A/D 変換

本マイコンボードには 10 ビット分解能の A/D 変換器が 1 個搭載されている。この A/D 変換器にアナログマルチプレクサを接続することで、8 チャンネルのアナログ入力が可能となっている。A/D 変換器は逐次

変換型なのでクロックが必要になる。このクロックによって、A/D 変換速度を変化させることができるが、A/D 変換速度が速いほど消費電流が多くなるので注意が必要である。

設定するレジスタは sfr26.h というヘッダファイル内で定義されている。以下に 10 ビット分解能、10[MHz] クロックの設定を示す。

3.4 回路例

以下に回路の例を示す。必ずしもこの通りにしなければならないわけではない。Fig.8 に出力と MCU ボードの端子台の間に Op-Amp によるバッファをはさんだ回路の例を示す。必ずしもこの通りにしなければならないわけではないことに注意する。

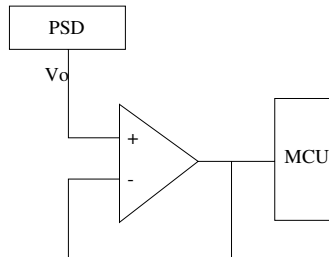


Fig. 8: バッファ

また、Op-Amp は LM358(ピン配置は Fig.9) を用いる。

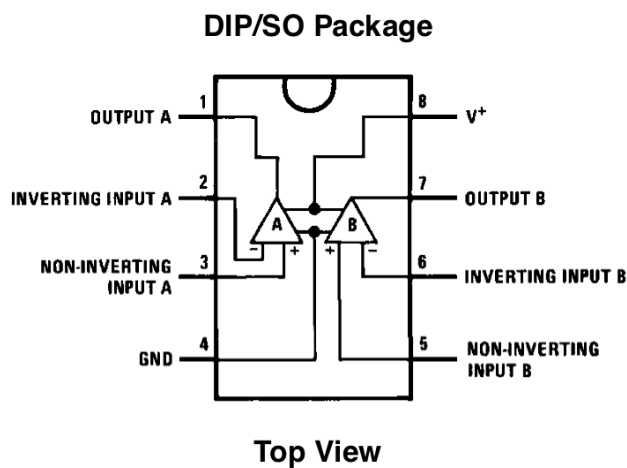


Fig. 9: LM358 ピン配置

4. RFID モジュールと RFID タグ

4.1 RFID とは

JR の Suica などの非接触 IC カードは所定の場所に近づくだけで、情報のやり取りができる。これらは、図 10 のようなものを、カードに内蔵することにより実現しているシステムである。IC チップは、大きなコイルアンテナを用いて外部からの電磁波を変換し電源としている。このコイルアンテナを使用して、外部と通信を行う。さらに、IC チップは不揮発性のメモリを内蔵しており、外部からデータを読んだり、書き込んだりできるようになっている。

多くの場合、このチップとアンテナをシール状にしたり、プラスチックのカードの中に内蔵したりして使用する。これを IC タグ（または RFID タグ）と呼ぶ。

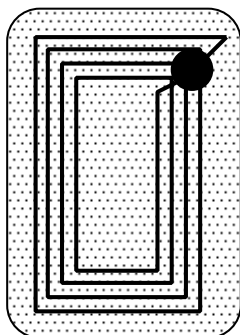


Fig. 10: RFID タグ

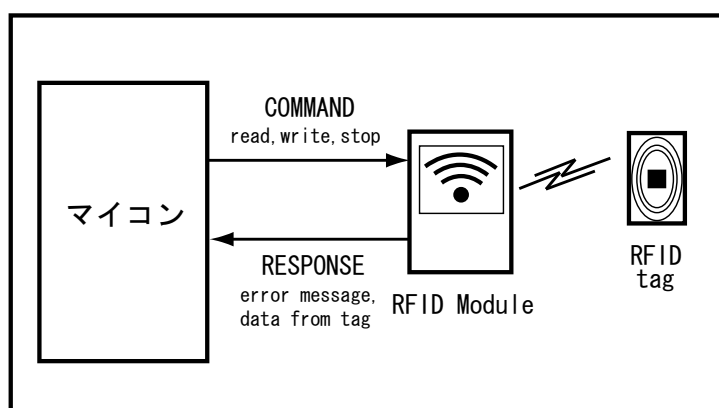


Fig. 11: RFID システムの構成及び関係

4.2 マイコン RFID モジュール タグの関係

今回使う RFID システムを構成するマイコン、RFID モジュール、そしてタグの関係を図 11 に示す。まずマイコンからタグに必要な情報を読むか、または書くかのコマンドを RFID モジュールに送る。そのコマンドを RFID モジュールが受信し、コマンドに応じてタグと情報のやり取りを行う。情報のやり取りが終わった後、RFID モジュールはタグとのやり取りがうまくできたかについてマイコン側に報告（送受信時のエラー発生有無、及びタグから読み出した情報）し、一通りの仕事が終了する。

	注意 マイコンからコマンドを出してからその結果がマイコンに戻るまで時間がかかるので、リアルタイムプログラムの構成としてはコマンド関数と受信データ取る関数を別々に設定するとよい
--	--

マイコンと RFID モジュール間のコマンドやデータの通信はシリアル通信によって実現されている。今年の SS2 ではルール上マイコン間の UART 通信を行わないため、試作検討では通信に関する解説は省略する。メカトロニクスラボ C コースの資料などから各自で勉強してもらいたい。

4.3 RFID リーダライタモジュール V720S-HMC73

競技時に貸与されるオムロン社製 RFID リーダライタモジュール V720S-HMC73（以下、V720）には様々な機能がサポートされ、これらの機能を活用することで、より効率の良い V720 との通信や V720 を介したタグとの交信が実現できるようになる。また、V720 をタグ検出センサとして用いるなど、新たな活用方法も見えてくるだろう。

4.3.1 交信モード

V720 では、様々な交信モードでタグとの交信を行うことができる。交信モードは、更新後のタグに対する処理の違いによりシングルモードと FIFO モードの 2 種類に、コマンドの処理手順、実行タイミングの違いによりトリガモード、オートモード、リピートモードの 3 種類に分けられ、これらの組み合わせにより計 6 種類（実際は FIFO 連続モードがあるので 7 種類）がある（試作検討時に使用した RFID ライブラリでは、シングルトリガモードを使用。）以下に、各モードについて簡単に説明する。なお、各モードの動作シーケンスについては V720 のデータシート（4-7、4-8）を参照。

- 交信後のタグに対する処理の違い

1. シングルモード (Single Mode)

一度交信 (Read/Write コマンド処理) を終了したタグに対して特別な処理を施さない。タグとの交信時間は FIFO モードに比べて短い。

2. FIFO モード (FIFO Mode)

タグとの交信 (Read/Write コマンド処理) 後、無変調発振を継続することで一度交信を終了したタグにアクセス禁止処理を施す。アクセス禁止されたタグは、アンテナの交信範囲外に出ると (もしくは、Stop コマンドを受信して発振を停止した後) 再び交信可能となる (アクセス禁止中のタグは、モジュールからはタグ不在と見なされるので注意。)

● コマンドの処理手順、実行タイミングの違い

1. トリガモード (Trigger Mode)

Read/Write コマンド受信後、すぐにタグと交信してレスポンス (交信範囲内にタグがなければタグ不在エラー) を送信。レスポンス送信後、新たなコマンドを受信するまで待機。

2. オートモード (Auto Mode)

Read/Write コマンド受信後、タグが交信範囲内に入ってくるのを待ってタグと交信し、レスポンスを送信。送信後 (タグを待っている状態では Stop コマンドを受信後)、新たなコマンドを受信するまで待機。

3. リピートモード (Repeat Mode)

Read/Write コマンド受信後、タグが交信範囲内に入ってくるのを待ってタグと交信し、レスポンスを送信。送信後は再びタグの検出を開始し、Stop コマンドを受信するまで新たなコマンドは受け付けずに ⇨ タグ検出 → 交信 → レスポンス送信 ⇩ のサイクルを繰り返す。

トリガモードだけでは、複数の V720 を切り替えて使用しても結局はそれぞれを単独で使っていることになりませんが、オートモード、リピートモードを用いれば、複数の V720 にコマンドを送信して並行して同時に使うことができるようになる。また、タグ検出時にレスポンスが返ってくるので、V720 をタグ検出センサとして使うこともできる。

4.3.2 シリアル通信速度

V720 では、上位制御ユニットとの通信速度 (転送レート) を 9600bps/38400bps から選択することができる。通信速度を 38400bps とした場合には、9600bps の場合と比較して V720 側のコマンド受信時、もしくは上位制御ユニット側のレスポンス受信時にオーバーラン・エラーが若干発生しやすくなるが、コマンド・レスポンスの通信時間は短縮される (V720 との通信速度なので、タグとの交信時間はシリアル通信速度とは関係ない。交信時間を短縮するためにはシングルモードを用いること。)

V720 側の通信速度を変更するには、通信速度設定スイッチを変更 (OFF: 9600bps, ON: 38400bps) する必要がある。

4.3.3 同バンクデータの一括読み書き

SS2 で使用される RFID タグには、1024 バイトのメモリを持つフィリップス社製の IC チップ SL2-ICS20 が搭載されているが、この V720IC チップのメモリマップには「ページ」と「バンク」という区分がある。「ページ」とは V720 の最小アクセス単位 4 バイト (ASCII コード 4 文字分) であり、「バンク」とは同時にアクセス可能な最大ページ数 16 ページのことである。つまり、同バンクのデータであれば一度のコマンド処理で任意のページを一括で読み書きすることができる。

4.4 V720S ライブラリ

RFID ライブラリに加え、以上の V720 の未使用機能を中心に、新たな機能を追加した C 言語ライブラリが V720S ライブラリである。

shisaku04¥rfid¥src にあるので内容を見ておくことを薦める。

4.4.1 ライブラリ構成

V720S ライブラリの構成は以下のようになっている。本ライブラリのインターフェース (4.4.2 節参照) を使用するためには、ユーザー側のアプリケーションプログラムで公開ヘッダファイル `v720s.h` をインクルードする。なお、UART2 ライブラリは、V720 を制御する際のハードウェア依存部分である OAKS16mini の UART2¹ (OAKS16mini のマニュアル第 13 章参照) を用いたシリアル通信のための関数群を提供するもので、V720S ライブラリ内で使われている。

¹UART (Universal Asynchronous Receiver Transmitter) シリアルポートなどに使われる通信回路で、送信時にパレルなバイトデータをシリアルビットストリームに変換、受信時にその逆に復元する

ライブラリ構成	
v720s.c (V720S Library C Source File)	上位ライブラリ関数の定義
v720s.h (V720S Library Public Header File)	上位ライブラリ関数 (公開部) に関連する定義・宣言
uart2.c (UART2 Library C Source File)	下位ライブラリ関数の定義
uart2.h (UART2 Library Private Header File)	下位ライブラリ関数 (非公開) に関連する定義・宣言

ライブラリのヘッダファイルは通常、インターフェースを提供するためのもの (公開ヘッダファイル) と、ライブラリ自身を作るために必要なもの (非公開ヘッダファイル) とに分けられる。これは、バージョンアップ (インターフェース以外の中身の変更) の際のトラブルを避けるために、アプリケーションプログラムではライブラリの中身に依存した書き方をして欲しくないからである。今回の V720S ライブラリを使ったマシンのソフトウェア開発でも、今後のバージョンアップによるサポートを受けたい場合には、公開ヘッダファイル **v720s.h** 以外に依存したコーディングは止めておくこと (非公開とは「ユーザ」ではなく「アプリケーションプログラム」に対して非公開という意味です。きちんと中身を理解してライブラリを使えるように、他の非公開ヘッダファイルやライブラリのソースにも必ず一度は目を通しておく。) ただし、V720S ライブラリは完全なオープンソースであるため、自分たちで独自にカスタマイズしたい場合にはこの限りではない。もちろん、ライブラリのソース自体を変更して使ってもらっても構わない。

4.4.2 インターフェース

V720S ライブラリでは、RFID リーダライタモジュール V720S-HMC73 を制御するための以下のライブラリ関数、グローバル変数、およびマクロ関数を提供する。

ライブラリ関数	
v720s_init	UART2 の初期化・V720 との通信テスト
v720s_putcmd_read	V720 への Read (タグデータ読みだし) コマンドの送信
v720s_putcmd_write	V720 への Write (タグデータ書き込み) コマンドの送信
v720s_putcmd_stop	V720 への Stop (実行中のコマンド停止、発振停止) コマンドの送信
v720s_getdata	V720 からのレスポンス (V720 の終了コードと読み出しデータ) 受信

グローバル変数	
uart2_received	UART2 側へ V720 のレスポンス受信完了通知
v720s_busy	一通りの仕事が終わったことを通知

4.4.3 ライブラリ関数

各ライブラリ関数の形式、機能、および返値は以下のようにになっている。

v720s_init	
形式	int v720s_init(void)
機能	UART2 初期化と V720 との通信テスト (TEST コマンド送信、レスポンス受信) を行う。
返値	正常処理時 0、それ以外の時エラーコード 999、998、110 ~ 114 を返す。

v720s_putcmd_read	
形式	int v720s_putcmd_read(unsigned char mode_type, unsigned char data_bank, unsigned char pagenum_fs_st, unsigned char pagenum_fs_end)
機能	mode_type の交信モードで data_bank と pagenum_fs_st ~ pagenum_fs_end で指定されるアドレスのデータを読み出す要求を V720 に出す。data_bank で示したバンクの pagenum_fs_st から pagenum_fs_end ページまで読み出しを要求する。
返値	正常処理時 0、それ以外の時エラーコード 997 ~ 993、991 を返す。

v720s_putcmd_write	
形式	int v720s_putcmd_write(unsigned char mode_type, unsigned char data_bank, unsigned int pagenum_fs_st, char* data_write)
機能	mode_type の交信モードで指定するアドレスに data_write で指定されるデータを書く要求を V720 に出す。バンク data_bank のページ pagenum_fs_st から書き始める。データがページ単位に満たない時はデータの終端にはスペースが挿入される。
返値	正常処理時 0、それ以外の時エラーコード 997 ~ 992 を返す。

v720s_putcmd_stop	
形式	int v720s_putcmd_stop(void)
機能	実行中のコマンド停止（発振状態の場合は発振停止）の要求を V720 に出す。
返値	正常処理時 0，それ以外の時エラーコード 997，996 を返す。

v720s_getdata	
形式	int v720s_getdata(char* data_read)
機能	V720 からのレスポンス（要求に対する終了コード，読み出しデータ）を共有メモリに格納する．読み出しデータは data_read で指定される char 型配列に文字データとして格納される．読み出しデータがない場合は data_read にはヌル文字が格納される．
返値	正常処理時は V720 の終了コード，それ以外の時エラーコード 995，110～114 を返す。

上記ライブラリ関数に関するいくつかの注意事項をまとめる．

- v720s_init 関数は「OAKS16mini の初期化」関数です。「V720 の初期化」は接続さえきちんとしていれば必要ない．そのため，複数の V720 を使う場合も v720s_init 関数は一度だけ実行すれば十分．
- v720s_putcmd_xxx 関数の返値は「V720S ライブラリのエラーコード」で，単にライブラリ関数をその仕様通りに呼び出しているかを確認するためのもの．従って，この返値が正常処理を示す 0 であっても V720 がコマンド処理を完了したことにはならない．V720 がコマンド処理を完了したかどうかを確認するためには，v720s_getdata の返値のうちの「V720 の終了コード」を確認する（v720s_getdata の返値には V720S，UART2 ライブラリのエラーコードも含まれている）．
- v720s_getdata 関数では，ポインタで指定された char 型配列に受信した UART1 の受信バッファデータをコピーする．この時，バッファ・オーバーフローを起こして他のメモリを破壊しないようにするためには，char 型配列のサイズを 65（1 バンク分のデータ + ヌル文字）以上にしておく必要がある（1 バンク以下の読み出しの場合はそれ以下）．残念ながら C 言語ではパラメータとして渡された文字配列のサイズを呼ばれた関数側で求める方法はないので，ライブラリ内でのエラーチェックは不可能．

4.4.4 グローバル変数

各グローバル変数の形式，初期値，および機能は以下のようになっている．

uart2_received	
形式	unsigned int uart2_received
初期値	0
機能	V720 からのレスポンス受信完了 (UART2 側) を通知する．UART2 の受信許可を出してから，受信が終了したときに 1 となり新たな受信を許可した時に 0 にセットされる．

v720s_busy	
形式	unsigned int v720s_busy
初期値	0
機能	コマンドを出してもいいことを通知する．V720S ライブラリ関数 v720s_getdata の呼出しにより前回受信したレスポンスが共有メモリに格納された時 0 でクリア，次のコマンドを出してからレスポンス受信完了時に 1 にセットされる．

上記グローバル変数の使い方について簡単にまとめておく．

- 送信データは V720 との間の共通の転送レートで 1 バイト分ずつ V720 ヘシリアル送信される．従って，コマンドは v720s_putcmd_xxx 関数の呼び出しから復帰した時に送信完了している訳ではない．複数の V720 を同時に使う（オート，リピートモードでコマンドを送信して，レスポンスを受信する前に別の V720 に切り替えてコマンドを送信するなどの）場合，コマンド送信完了通知が届いてから切り替えを行うと良い．
- v720s_getdata 関数は，V720 からのレスポンスが届いた時に呼び出さなければ意味がない．v720s_getdata 関数の実行タイミングはレスポンス受信通知より判断．

4.4.5 ライブラリ関数引数

ライブラリ関数の引数 *mode_type* , *data_type* , *fs_pagenum* の指定方法を以下に示す .

交信モード <i>mode_type</i>	
V720S_MODE_ST	シングルトリガモード : Single Trigger
V720S_MODE_SA	シングルオートモード : Single Auto
V720S_MODE_SR	シングルリピートモード : Single Repeat (読みだしのみ可)
V720S_MODE_FT	FIFO トリガモード : FIFO Trigger
V720S_MODE_FA	FIFO オートモード : FIFO Auto
V720S_MODE_FR	FIFO リピートモード : FIFO Repeat

アクセスデータ <i>data_type</i>	
V720S_DATA_UID	シリアルナンバー : Unique Identification (読み出しのみ可)

フリースペース読み出し , 書き込みページ指定 <i>pagenum_fs_st or _end</i>	
0 ~ 15	フリースペースから読み出す時は最初と最後ページ (0 ~ 15) を指定 書き込む時は書き込む初期ページ (0 ~ 15) を指定

4.4.6 エラーコード一覧

V720S ライブラリ関数の返値である V720S , UART ライブラリのエラーコード , V720 の終了コードの一覧を以下に示す (V720 の終了コードは一部のみなので , 詳しくはデータシート (4-9) を参照すること .)

V720S ライブラリエラーコード (~ 999)	
999	V720 と通信できない .
998	TEST コマンド実行後の V720 のレスポンスの終了コードが正しくない .
997	UART の初期化 (v720s_init 関数の呼び出し) が行なわれていない .
996	前回コマンドの送信が完了していません (コマンド送信完了通知がまだ届いていない)
995	V720 からのレスポンスは未受信です (レスポンス受信完了通知がまだ届いていない)
994	交信モード <i>mode_type</i> の指定が正しくない .
993	アクセスするタグデータ <i>data_type</i> の指定が正しくない .
992	書き込みデータの文字数が正しくない (フリースペース : 最大 64 文字)
991	フリースペースの読み出しページ数 <i>fs_pagenum_st or _end</i> の指定が正しくない .

UART ライブラリ受信エラーコード (100 ~)	
102	UART 送受信中に受信バッファにバッファ・オーバーフローが発生 .
111	UART 受信中にオーバーラン・エラーが発生 .
112	UART 受信中にフレーミング・エラーが発生 .
114	UART 受信中にパリティ・エラーが発生 .

V720 終了コード (0 ~)	
0	コマンドを正常に実行した .
70	タグとの交信中にノイズ等の障害が発生し , 正常に完了しなかった .
72	コマンド処理時に交信範囲内にタグが存在しない . (Trigger mode だけ有効)

4.4.7 通信速度の変更方法

シリアル通信速度 (転送レート) はデフォルトでは 9600bps となっている . 転送レートを 38400bps に変更するためには以下を行う .

- V720 側の設定 : V720 の通信速度設定スイッチを変更 (4.3.2 節参照)
- OAKS16mini 側の設定 : uart2.h の「#define UART2_BOWRATE 129」を「#define UART2_BOWRATE 31」に変更

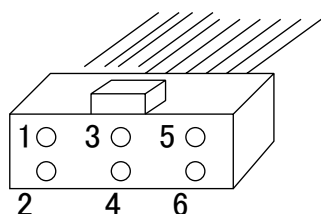
4.5 マイコンと RFID モジュールと接続

4.5.1 RFID コンポーネント

RFID リードライトモジュール V720S は、固定して使うことを目的として作られているため、ケーブルを着脱したり、競技ごとにはがしたりするような使い方は想定されていない。このような使い方をすると、コネクタ部分が破損するなど、耐久性に問題がある。

そこで、これらの耐久性の向上と、保守性のため、V720S、ケースやケーブルを含めて、「RFID コンポーネント」とする。

RFID リードライトモジュールからの信号も、7 番から 10 番までのピンの信号を省いている。信号と、コネクタ形状について、図 12 に示す。



Pin No.	記号	機能
1	5V	+5V 電源
2	GND	グランド
3	RxD	シリアル入力
4	TxD	シリアル出力
5	GND	グランド
6	GND	グランド

Fig. 12: RFID からのケーブルのコネクタ

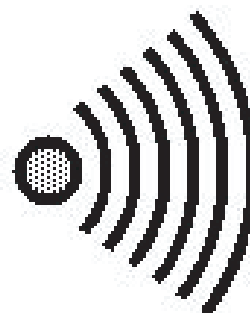


Fig. 13: アンテナ面を示すシール

RFID コンポーネントでは、図 13 のシールが張ってある方が、RFID リードライトモジュールのアンテナ面。(図 13 の丸が交信範囲の中心。) タグには、シールの着いている面を向けるようにする。

RFID コンポーネントのマシンへの取り付けは、マジックテープにて行う。詳細については後日発表。

4.5.2 マイコンボードとの接続

実際にロボットを作る際には、ユーザーボード上などに、外部回路を組む。RFID を切り替えるための推奨回路とフィールド上のタグの位置情報については後日発表。

今回の試作検討では、外部回路は作成せず、一番上についているデバッグボードをはがして、SS2 ボードから直接ポートにつなぐことで通信を行う。

注意 SS2 ボード上の IC 類は非常に高価なので、静電気や不注意によるショートなどで破壊しないように気をつけること

OAKS16mini では、P70 が UART2 の送信ポート、P71 が UART2 の受信ポートになっている。したがって、RFID モジュールの受信ポートと P70 を、RFID モジュールの送信ポートと P71 を接続する。この際 V850/KG1 と RFID モジュールの送受信ポートをそれぞれクロスでつなぐことに気をつける。(また、TXD2 は N チャンネルオープンドレインだが、すでにプルアップされていますので、抵抗を付ける必要はない。)

図 14 に、SS2 ボードから 5V、GND、P70 および P71 をとる方法を示す。電源と GND を間違えないように気をつけること。

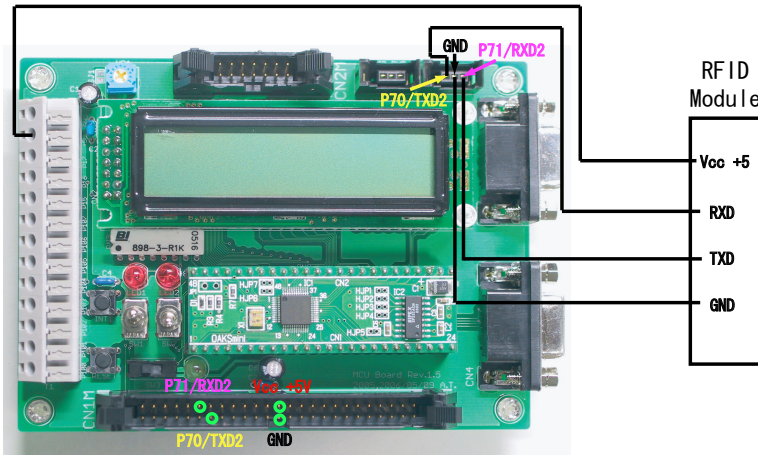


Fig. 14: SS2 ボードから 5V , GND , P70 および P71 をとる方法

課題

1. サンプルプログラム shisaku04¥rfid を実際にビルド・デバッグを行い, 動作を確認する.
 - RFID モジュールの蓋を開けてチップ LED の動作を確認する.
 - トグルスイッチを切り替えてタグとのデータの読み書きをやる.
 - 自分たちでオリジナルのメッセージを考え, タグに読み書きする.
2. main.c を修正してプログラムの開発に慣れる.
 - FIFO モードでの動作を確認する.
 - 読み書きするバンク, ページを変えてみる.

余力があるものは積極的にソースに改良を加えていくとよい.

なお, 競技で使用される RFID タグについて, コースの情報 (通過するゲートの順番や, 次のゲートまでの距離) をどのように記憶させておくかは現在検討中である. 場合によってはライブラリの修正などが発生すると考えられる.