

ロボット博士養成講座

ロボティクスプロフェッサーコース

不思議アイテムⅢ-2①

第2回

液晶ディスプレイに
センサー情報を表示させる

講師用

目 次

0. 液晶ディスプレイにセンサー情報を表示させる

0.0. 「液晶ディスプレイにセンサー情報を表示させる」でやること

0.1. 必要なもの

1. 液晶ディスプレイに文字を表示させる

1.0. 組み立てと注意点の確認

1.1. 好きな文字を表示させる

1.2. 文字の大きさ・色・位置を変更する

2. 液晶ディスプレイにセンサー情報を表示させる

2.0. タッチセンサーの情報を表示させる

2.1. タクトスイッチの情報を表示させる

2.2. ボリュームセンサー（可変抵抗ボリューム）の情報を表示させる

2.3. 超音波距離センサーの情報を表示させる

3. まとめ

○ 授業開始にあたって

授業のはじめは、着席させ、大きな声であいさつしてから始めます。

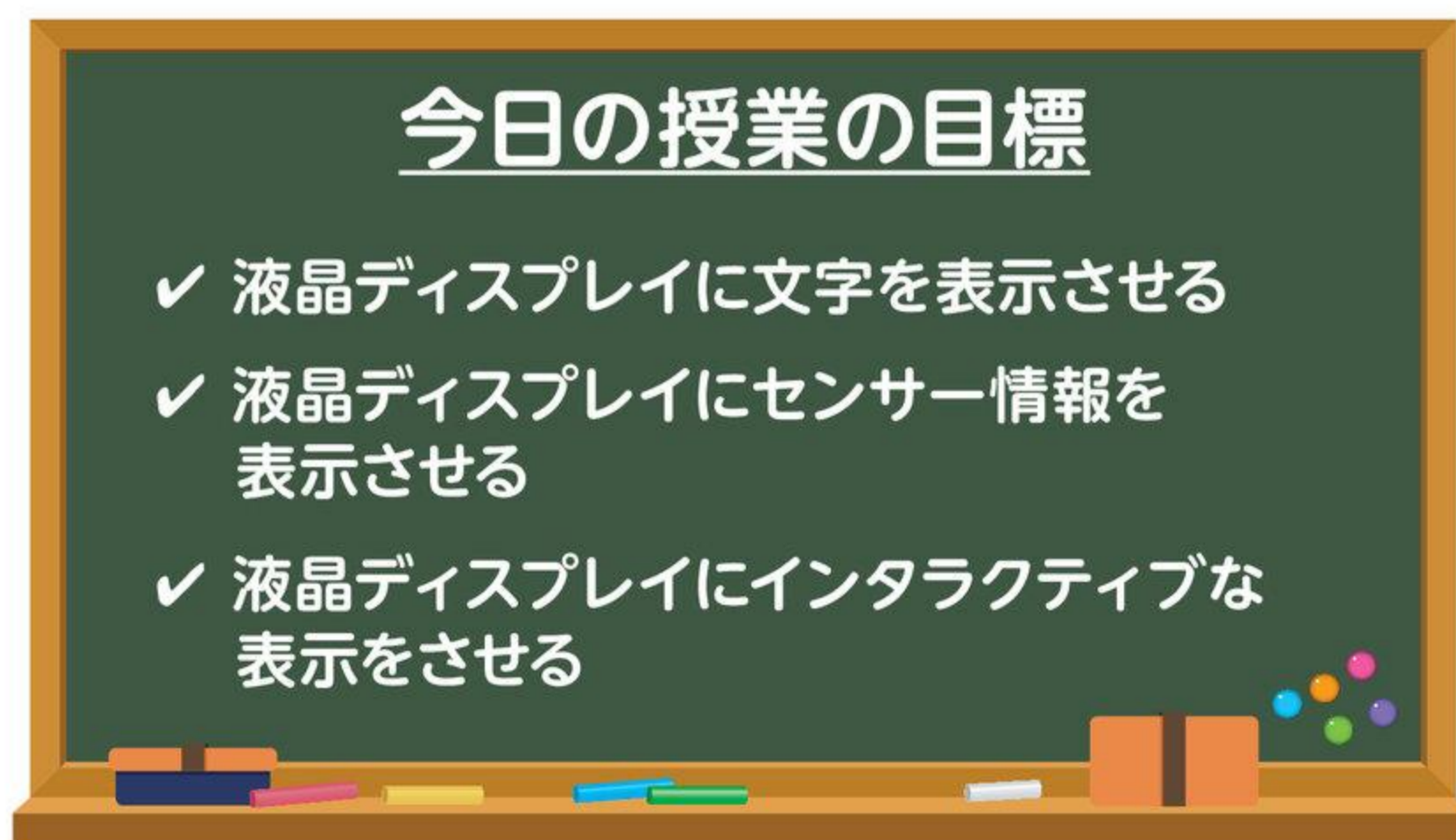
○ 今回の目標をパネルで用意するか、黒板に予め書いておきます。

（授業の目標を明確化することは大変重要なことですので、生徒によく理解させます）

目安時間は授業時間 120 分のうち、休憩 10 分程度取ることを想定しています。
生徒の進捗状況により、休憩時間などを調整して授業を行ってください。

0. 液晶ディスプレイにセンサー情報を表示させる (目安 10分)

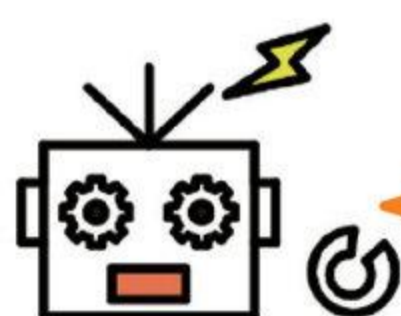
0.0. 「液晶ディスプレイにセンサー情報を表示させる」でやること



不思議アイテムⅢ-2、第2回です。今回は、液晶ディスプレイに四角形や円といった基本的な図形を表示する練習をしました。今回はもう少し発展して、液晶ディスプレイに文字などを表示させる方法を学びましょう！

文字が表示できるようになると、ロボットの状態を文章で読み取ることができるようになるので、たとえばロボットに異常が見つかったとき、どこに問題があるのか判別しやすくなったり、これまでロボプロで扱ってきた各種センサーをつないで「目標までの距離は〇〇cm」「今は左のタッチセンサーがオンになっている」などの情報が見られるようになるとりします！よりカシコイロボットに近づきますね！

ところで、一般的なゲームはコントローラーのボタンを押すと画面に変化が起こり、その変化に合わせて次に押すボタンを決めるといった流れになっていますよね。このように、人とコンピューターがお互い情報をやりとりしながら操作を進めることを「インタラクティブ」と言います。このタームで学ぶ事をうまく組み合わせれば液晶ディスプレイでゲームを遊ぶこともできるようになります。ゲーム製作に活かせるようにインタラクティブな表示についても学んでおきましょう！



体を使ってロボットとコミュニケーションがとれる？

0.1. 必要なもの

今回は、以下のパーツを使用します。






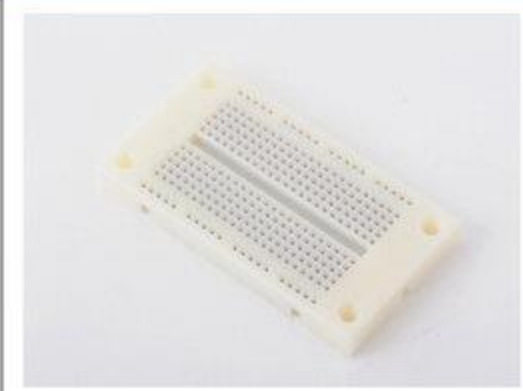

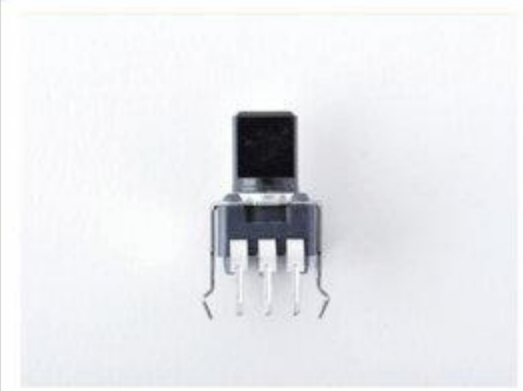



USBケーブル	1	マイコンボード	1	ロボプロシールド	1	タッチセンサー	2
							
超音波距離センサー	1	301ブレッドボード	1	ジャンパー線	65	可変抵抗ボリューム	2
							
タクトスイッチ	10	姿勢検出シールド	1	液晶ディスプレイシールド	1		
							

図0-0 必要なもの

液晶ディスプレイシールドの表面に貼ってある保護シートは剥がして使用しましょう。
また、ディスプレイに傷がつくのを防ぐために、他の部品とぶつからないように保管してください。

1. 液晶ディスプレイに文字を表示させる (目安 20 分)

1.0. 組み立てと注意点の確認

1) 組み立て

図1-0が、今回使う液晶ディスプレイシールドの組み立て完成図です。

前回の組み合わせに、ロボプロシールドを追加しましょう。下から、マイコンボード・ロボプロシールド・姿勢検出シールド・液晶ディスプレイシールド、という順になるように組み立てます。



図1-0 液晶ディスプレイシールドの取り付け完成図

2) 注意点の確認

この液晶ディスプレイの座標に関しては注意が必要でしたね。

図1-1のように座標をとります。一般的な数学とはちょっと^{ちが}違う座標の取り方をしていました。この液晶ディスプレイの場合、X座標は160まで、Y座標は128までです。

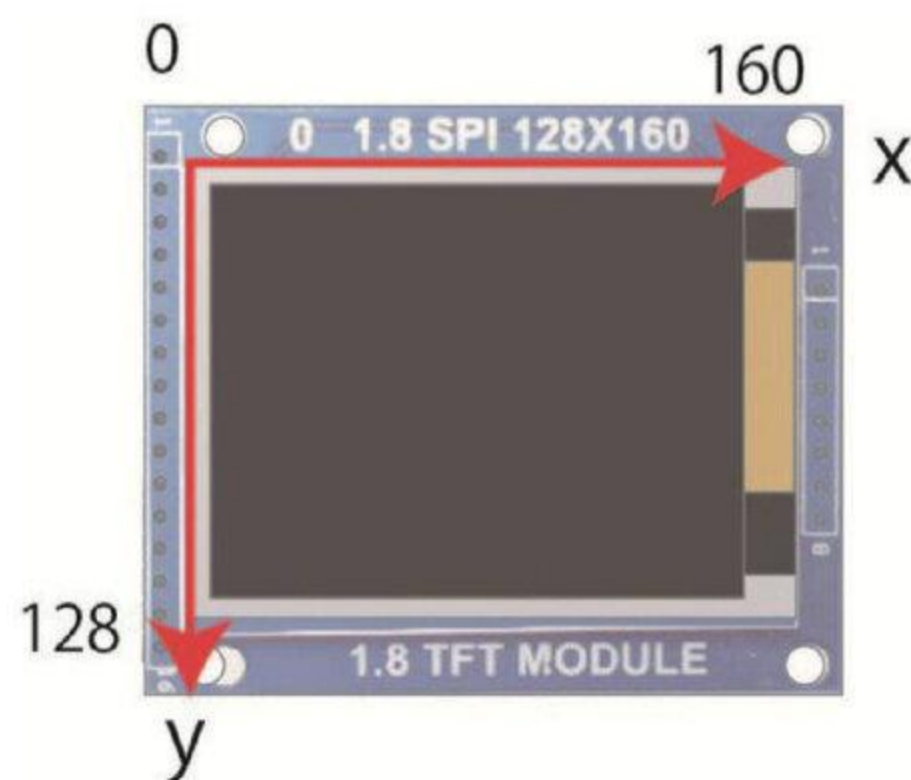


図1-1 液晶ディスプレイシールドの表示

1.1. 好きな文字を表示させる

ここではディスプレイに好きな文字を表示させてみます。
まずは、次のプログラムを実行してみましょう。

🔄 プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD1 > drawText

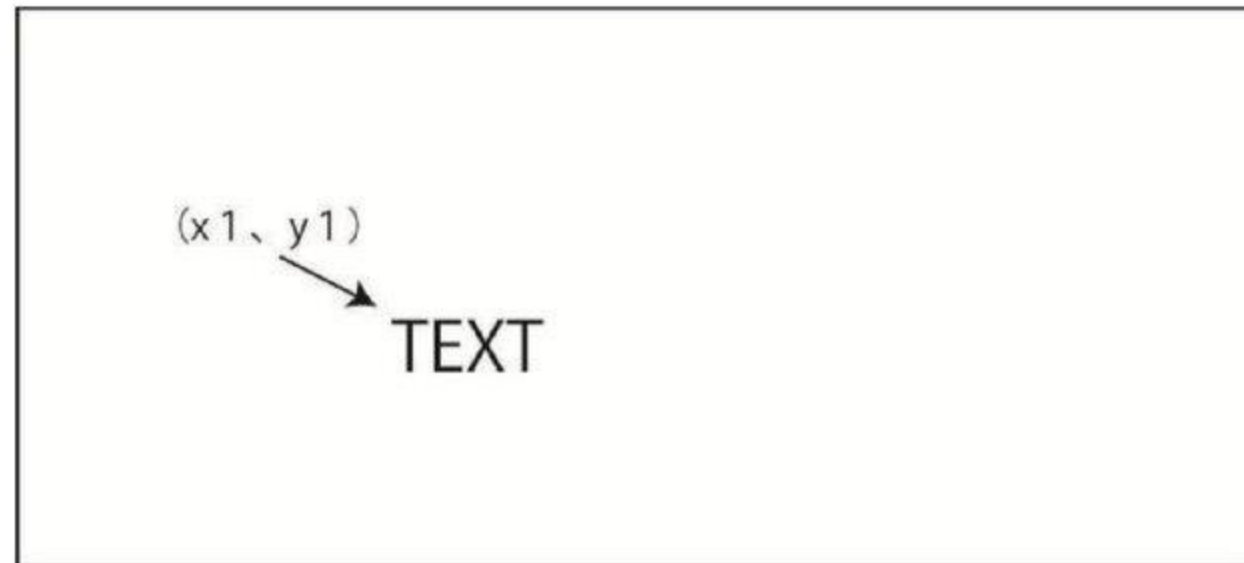


図1-2 プログラム「drawText」の解説

実行結果：「Hello World!」と表示される。

では、プログラム「drawText」の中を見てみましょう。文字の表示方法は以下ようになります。

📄 プログラム「drawText」より抜粋 ばっすい

```
void setup() {
  TFTscreen.begin();
  TFTscreen.background(0, 0, 0);           // 背景色は黒
  TFTscreen.stroke(255, 255, 255);        // 線の色は白
  // TFTscreen.setTextSize(2);           // 文字サイズ変更
  TFTscreen.text("Hello World!", 5, 30);  // (5,30)にテキストを表示する
}
```

以下が新しい命令です。

命 令：「TFTscreen.setTextSize」 「TFTscreen.text」

実行結果：文字を描く

使い方： TFTscreen.setTextSize([サイズの値]); // サイズを指定
TFTscreen.text (" [文字列]" , [x 座標], [y 座標]);

例： TFTscreen.setTextSize(1);
TFTscreen.text("Helloworld", 5, 30);

1.2. 文字の大きさ・色・位置を変更する

やってみよう!

表示される文字を変更してみよう。

1. □内の文字を変更してみよう。ただし日本語は扱えないので注意しよう。

例) "I LOVE ROBOTS!"

2. 文字の大きさを変更してみよう。//を削除してから入力してね。

例) TFTscreen.setTextSize(5);

3. 文字の表示位置(座標)を変更してみよう。

例) TFTscreen.text("I LOVE ROBOTS!",1,110);

4. 文字の色を黄色、背景色を青にしてみよう。

表1-0のテキストの色データを参照して変更してみよう。

表1-0 RGBカラーコード

Red	Green	Blue	色
255	255	255	
0	0	0	黒
255	0	0	赤
0	0	255	青
0	255	0	緑
255	255	0	黄
0	255	255	水色
255	0	255	紫
110	60	170	紺
50	50	50	黒

講

4. の解答プログラムは以下となります。

RoboticsProfessorCourse3 > MagicItemLCD1 > drawTextAns1

RGB カラーコードなどを使用して、背景色や大きさなどを自由に変えて課題を行ってください。プログラムの変更方法を反復して理解を深めてください。

2. 液晶ディスプレイにセンサー情報を表示させる (目安 70 分)

2.0. タッチセンサーの情報を表示させる

1) タッチセンサーを接続する

最初はタッチセンサーを使ってみます。次のようにケーブルを [D2]、[D3] に接続してみましょう。

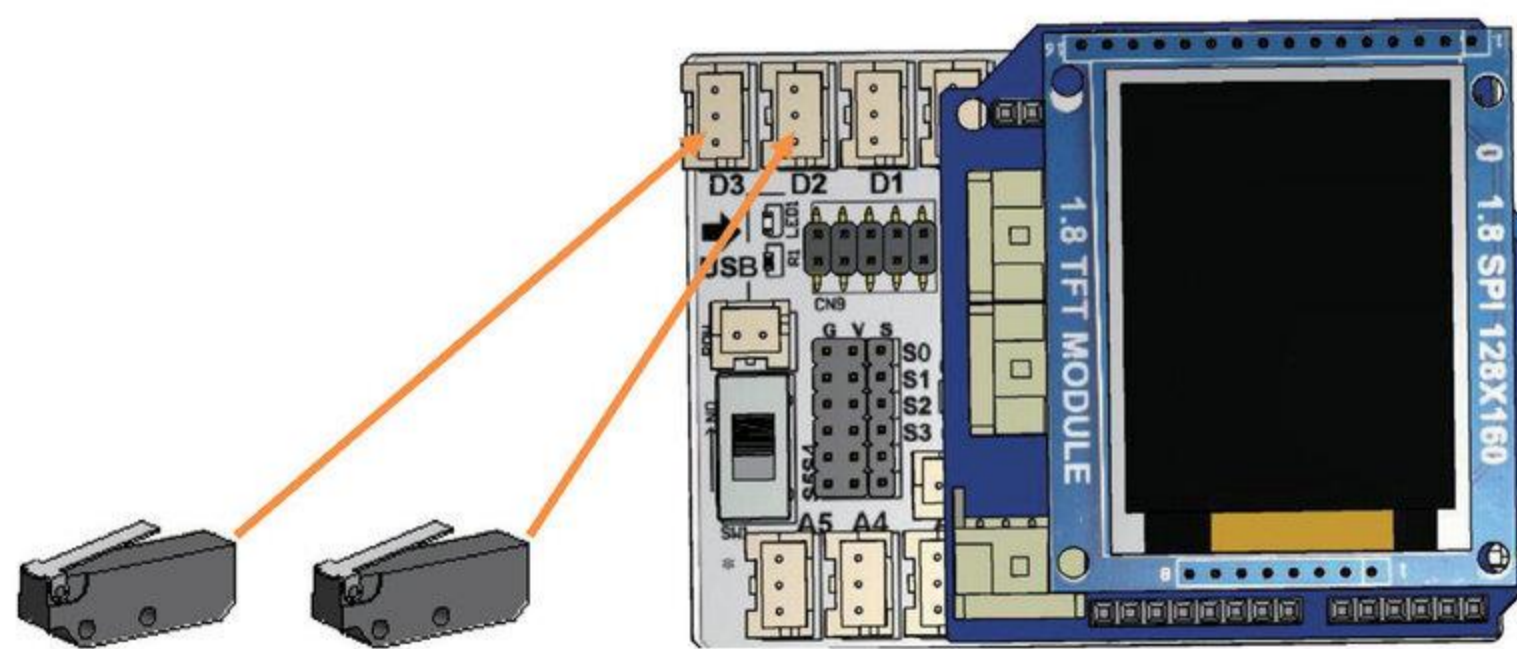


図2-0 タッチセンサーの配線指示図

2) スイッチを読み取る

タッチセンサーの接続が完了したら、次のプログラムを実行しましょう。

🔄 プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > TouchLCD1

実行結果：[D2] のタッチセンサーを押すと「push D2」、[D3] のタッチセンサーを押すと「push D3」と表示される。

では、プログラムを確認していきましょう。

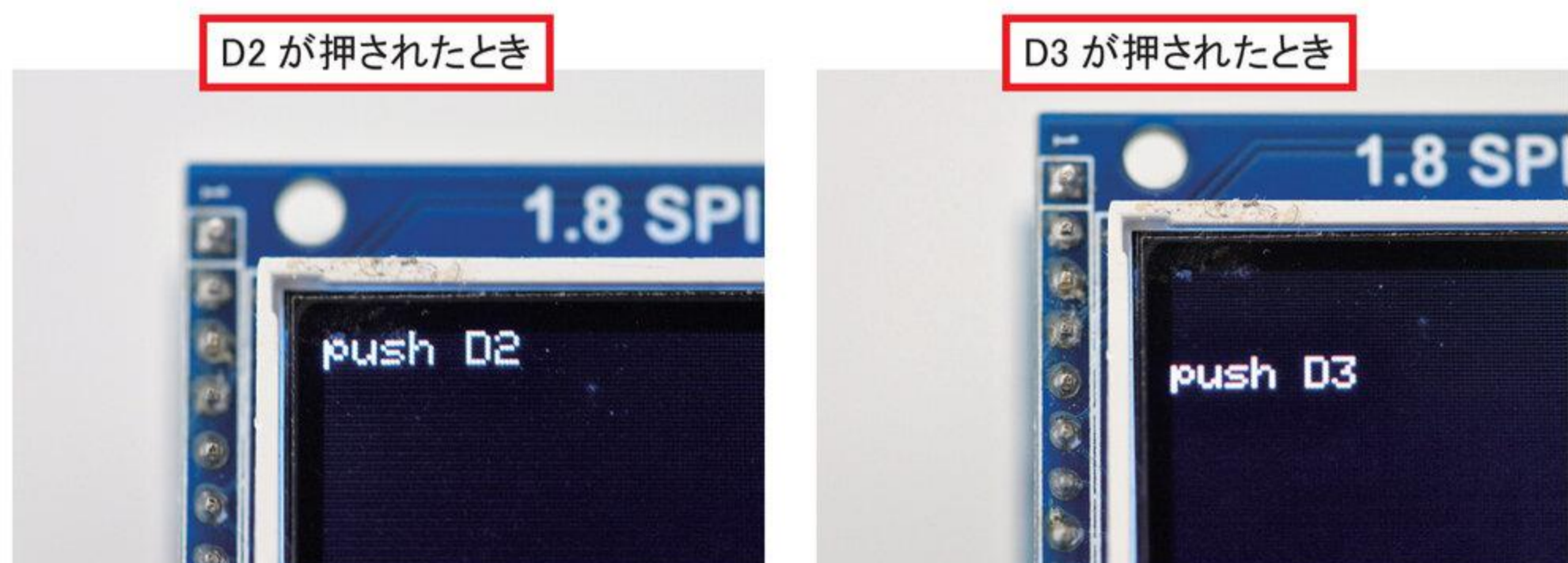


図2-1 タッチセンサーテスト

□ プログラム「TouchLCD1」より^{ぼっすい}抜粋

```
if (digitalRead(D2) == ON && digitalRead(D3) == OFF)
//もしD2がおさされていてD3がおされていないならば
    TFTscreen.text("push D2", 0, 0);           //(0,0)にテキストを表示する
if (digitalRead(D2) == OFF && digitalRead(D3) == ON)
//もしD2がおされておらずD3がおされていれば
    TFTscreen.text("push D3", 0, 20);          //(0,20)にテキストを表示する
```

`digitalRead` は、スイッチの状況を読み取る命令でしたね。
また、プログラムの初期化の `setup()` でピンを設定をしています。

□ プログラム「TouchLCD1」より^{ぼっすい}抜粋

```
pinMode(D2, INPUT_PULLUP); // D2にタッチセンサーを接続
pinMode(D3, INPUT_PULLUP); // D3にタッチセンサーを接続
```

なお、復習になりますが、`digitalRead` は次のような書き方でしたね。

命 令：「digitalRead」

実行結果：指定のピンに電流が流れているかチェックする

使い方： `digitalRead([ポート名]);`

例： `if(digitalRead(D1) == ON)`

// もしD1ピンに電流が流れている（ONである）ならば

やってみよう!

プログラム「TouchLCD1」を^{へんこう}変更して、タッチセンサーが同時に押された場合に画面に「push 1/2 ON」と表示されるようにしてみよう。

講

解答プログラムは以下となります。
RoboticsProfessorCourse3 > MagicItemLCD2 > TouchLCD2

ステップアップ

前回の図形描画のプログラムを参考に、以下のようなプログラムを書いてみよう。

- ・ D2 タッチセンサーが反応したら四角をかく
 - ・ D3 タッチセンサーが反応したら円をかく
 - ・ D2、 D3 タッチセンサー両方が反応したら三角をかく
- 内容は前回の復習だよ。

講

解答プログラムは以下となります。
RoboticsProfessorCourse3 > MagicItemLCD2 > TouchLCD3

3) ポートを^{へんこう}変更する

続いて、タッチセンサーのポートを^{へんこう}変更してみましょう。まずタッチセンサーを A4、 A5 に接続してください。接続を終えたら、以下のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > TouchLCD4

実行結果: A4 に接続したタッチセンサーを押すと「push A4」、 A5 に接続したタッチセンサーを押すと「push A5」、両方同時に押されたら「push A4/A5 ON」と表示される。

これでタッチセンサーの表示はばっちりですね。タッチセンサーは、ロボプロシールドから外しておきましょう。

2.1. タクトスイッチの情報を表示させる

1) 301ブレッドボード (タクトスイッチ)

続いてタクトスイッチを使ってみましょう。図2-2のように301ブレッドボードとジャンパー線、タクトスイッチを使い配線してみましょう。なお、ジャンパー線やタクトスイッチの色は図と異なっても構いません。

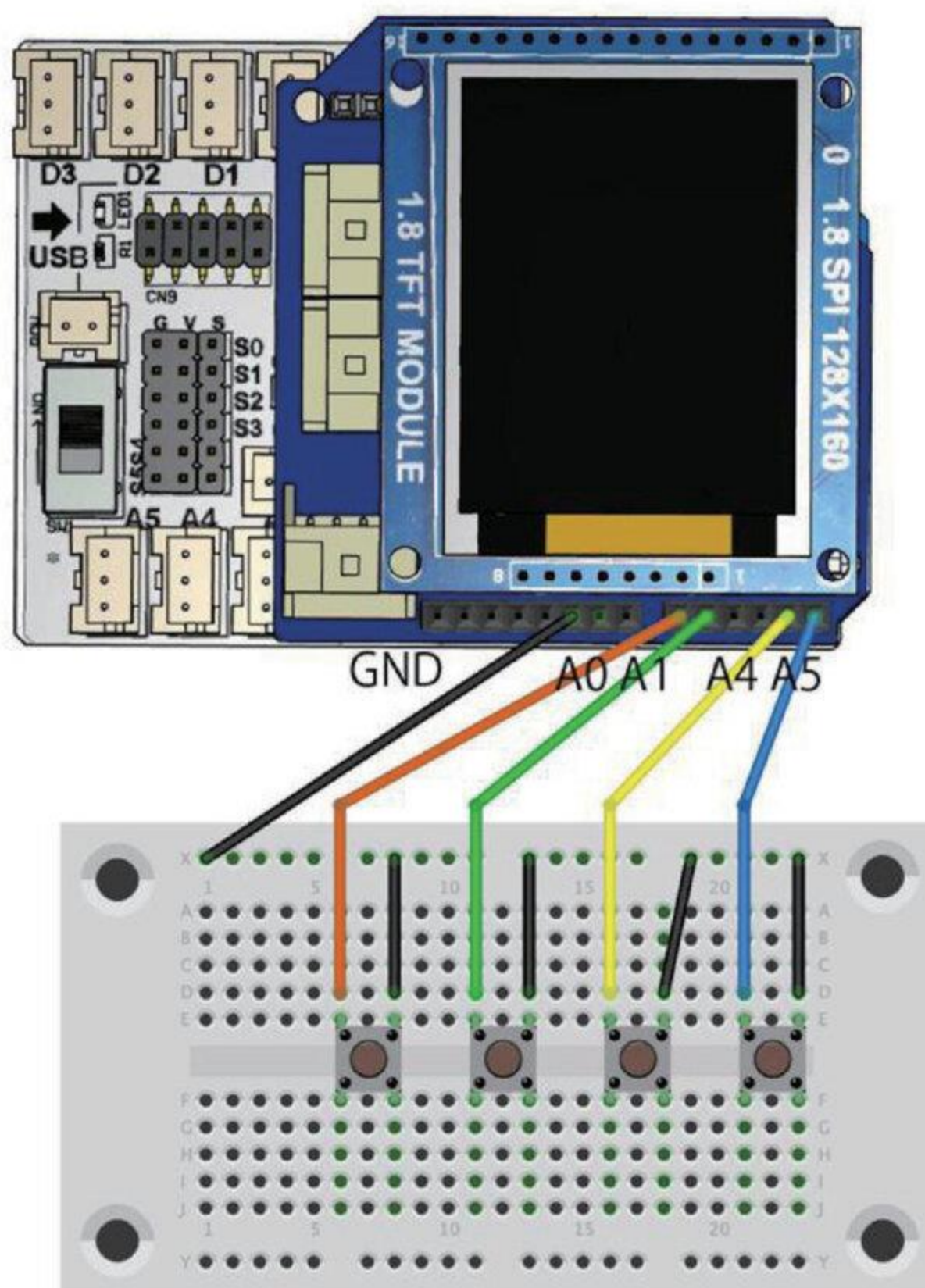


図2-2 回路

やってみよう!

4つのタクトスイッチを使用すると、ぜんぶで16通りの条件をつくることができるよ。表2-0の空欄をうめて、16通りのスイッチの状態をすべて書き出してみよう!

 ヒント

2つのタッチセンサーの場合は、以下のように4つの状態に分けることができるよ。同じように考えてみると、4つのタクトスイッチであればいくつの条件分けができるかな?

	状態1	状態2	状態3	状態4
タッチセンサー1	ON	OFF	ON	OFF
タッチセンサー2	OFF	ON	ON	OFF

表2-0 16通りのスイッチの状態

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Sw0	OFF															ON
Sw1	OFF															ON
Sw2	OFF															ON
Sw3	OFF															ON

4つだけしかスイッチがないと、16通りしか物事を表現できないということがわかります。これがデジタルの世界なのです。状況を表す手段がないわけですね。人の世界とは大きく異なります。もっと具体的に考えてみましょう。4つだけしかスイッチのない世界では、たとえば「はい」「いいえ」「いいね」「悪いね」「初めまして」「久しぶり」「こんにちは」「さようなら」「おやすみ」「おはよう」「ありがとう」「どういたしまして」「すみません」「お願いします」「いくらですか」「これください」…ほどしか、言葉が話せないのです。これでは、細かいことは全く伝えられません。

この不自由な世界はスイッチが4つあることから「4ビットデジタル」と表現します。

2) スイッチを読み取る

配線ができたら、以下のプログラムを実行してください。

 プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > TactSwLCD1

実行結果：各タクトスイッチを押すと、画面に様々な数字が表示される。タクトスイッチを押す組み合わせによって表示される数字は変化する。

やってみよう!

実際にタクトスイッチを16通りすべてのパターンで押してみよう!

スイッチを押すと、0～15の数字が表示されると思います。それ以外の文字は表示できません。

3) switch case文

今度は、次のプログラムを実行しましょう。

プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > TactSwLCD2

実行結果：プログラム「TactSwLCD1」と同じ。

プログラムの中をみてください。どちらが読みやすいでしょうか？

今回の場合、プログラム「TactSwLCD2」では「switch case文」を使うことで、プログラム「TactSwLCD1」と比べて、プログラムがうまくまとまっています。

さてここでこれまでも出てきた「switch case文」をもう一度復習しておきましょう。



POINT

switch case文はif文と同じように、場合分け（条件分岐）に使われます。switchで指定された変数が、それぞれのcaseと一致すると、そのあとに続く文が実行されます。また、処理が終わったら、breakと書いてswitch文から抜け出す必要があります。必ずcaseの最後に「:」を付けるところも覚えておきましょう。

プログラム「TactSwLCD2」より抜粋

```
char c;  
(中略)  
switch (c){  
    //0000  
    case 0:  
        TFTscreen.text("0000 0", 0, 0); // (0,0)にテキストを表示する  
        delay(100);  
        TFTscreen.stroke(0, 0, 0); // 線の色は黒  
        TFTscreen.text("0000 0", 0, 0); // 黒で上書きして消す  
        break;
```

4) デジタルの世界の情報量

さて、デジタルの世界をもう少し考えてみましょう。

スイッチのON/OFFの2つの値を使って数字を表現するというのは、2進数の話でした。

では、もっと多くのことを表現したいと思ったらどうしたらよいのでしょうか？ スイッチを増やせばよいですね。

スイッチが8個あったら「8ビット」、16個あったら「16ビット」と表現します。聞いたことはありますか？ なお、「8ビット」の場合は「1バイト」と表現します。「16ビット」は「2バイト」です。さらに広げていくと、「1024バイト」は「1K(キロ)バイト」、「1024Kバイト」は「1M(メガ)バイト」、「1024M(メガ)バイト」は「1G(ギガ)バイト」です。

つまり、「1G(ギガ)バイト」ともなると、8,000,000,000個以上のスイッチがあります！

身近にある携帯電話やデジタルカメラ、SDカードの凄さがわかってきましたか？

私たちの日常はスイッチまみれとも考えられますね。膨大なスイッチを身近なマイコンが肩代わりして、ここでいうタクトスイッチのボタンを押しまくっているといったイメージです。人間は10個同時くらい、足を使っても20個くらいで限界ですね。つまりたった20ビットです。

ちなみに、「このデータは8,000,000,000個分(ビット)のデータです。」のようにかくのは大変ですね。そこで、記憶容量など大きい数値(情報量)をあらわすM(メガ)やG(ギガ)などの補助単位と呼ばれるものを使っているわけです。



図2-3 情報量の補助単位

2.2. ボリュームセンサー（可変抵抗ボリューム）の情報を表示させる

1) 可変抵抗ボリュームを接続する

続いては、ボリュームセンサーの情報を表示してみましょう。先ほどまではデジタルを読み込んで表示する方法でしたが、今度はアナログを読み込んで表示するプログラムを学びましょう。まずは図2-4を見て配線してみましょう。

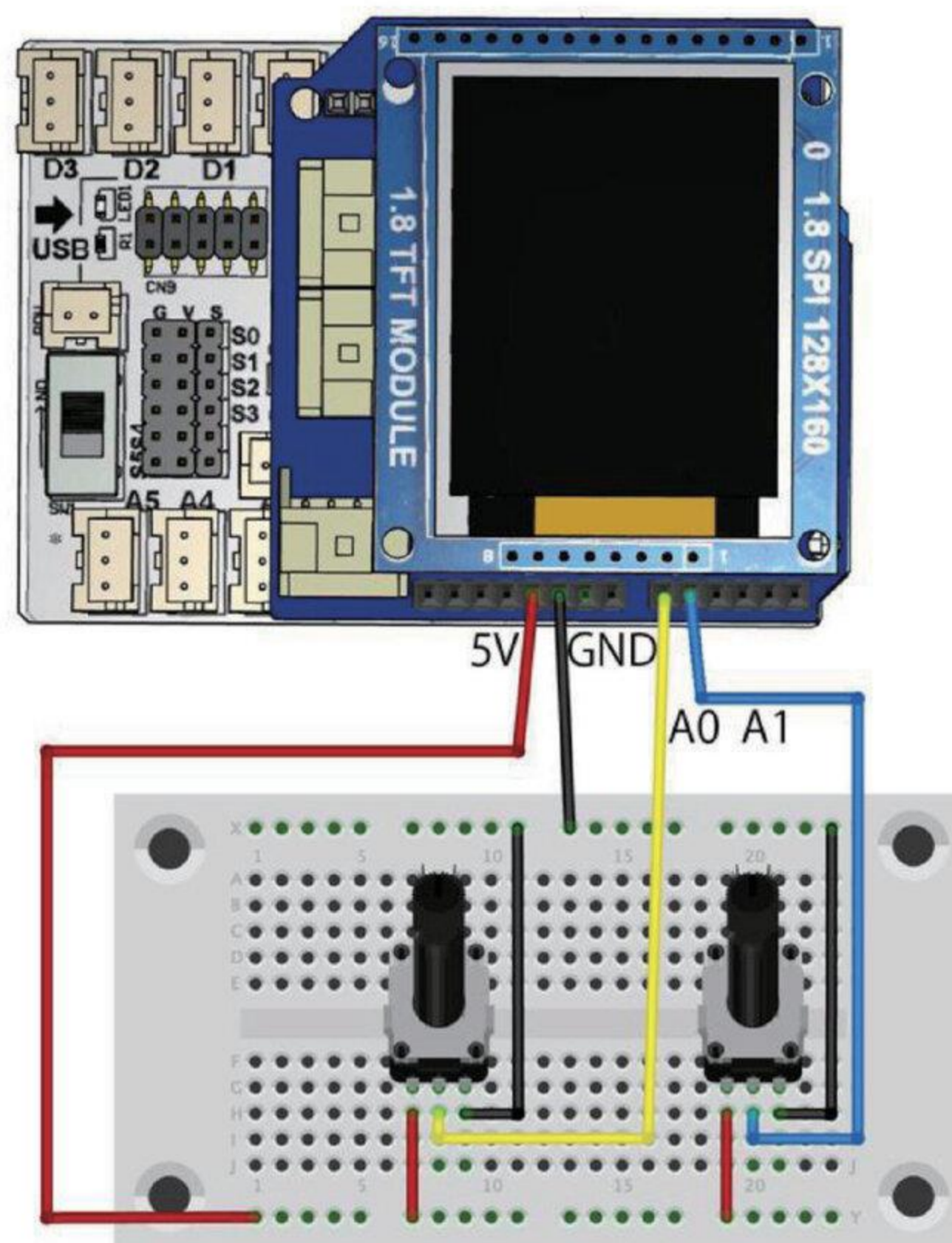


図2-4 可変抵抗ボリューム回路

2) 文字を表示する

配線が完了したら、以下のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > VolumeLCD1

実行結果：両方のボリュームの数値が表示される。可変抵抗ボリュームのつまみを回すと値が変化する。

やってみよう!

可変抵抗ボリュームのつまみを回して数値の変化を見てみよう。そのときの変化の最小と最大を確認しておこう。先ほどの話でいうと、何ビットにあたるか逆算できるかな？

命令: [analogRead]

実行結果: A0/1/2/3/4/5のマイコンのポートに繋がれた、電圧(0 ~ 5V)を検出する。
5Vが0 ~ 1023で検出されてその値を返してくる。

使い方: analogRead ([ポート名])

例: a=analogRead(A0) // A0に接続されたセンサーの電圧をはかって変数aに入れる

digitalReadが0 (OFF) と1 (ON) の2通りしか判定できなかったのに対し、こちらは1つの部品で微妙な数値の違いを入力できますね。

ちなみに、1024通りの結果に分かれるのに必要なスイッチは10個(10ビットデジタル)です。

3) 文字を移動させる

プログラム「VolumeLCD1」を変更して、好きな文字が可変抵抗ボリュームで画面上を動くように改造してみます。

プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > VolumeLCD2

```
void loop(){
  int x = map(analogRead(A0), 0, 1023, 0, 160);
  TFTscreen.stroke(255, 255, 255); //線の色は白
  TFTscreen.text("*", x, 64);      //ボリュームの値に応じた場所に*を表示
  delay(100);
  TFTscreen.stroke(0, 0, 0);       //線の色は黒
  TFTscreen.text("*", x, 64);     //黒で上書きして消す
}
```

ここで重要になるのは、緑の部分のmapです。センサーから返ってくる値の範囲は0 ~ 1023でした。これを画面の範囲に当てはめるため、0 ~ 160の範囲で値を変換しているわけです。

もちろんそのままでもよいのですが、画面外にキャラクターが出て行ってしまいう結果になってしまいます。



POINT

map関数

数値をある範囲から、別の範囲に変更します。現在の範囲の下限と同じ値を与えると、変更後の範囲の下限を返し、現在の範囲の上限と同じ値なら、変更後の範囲の上限を返します。その中間の値は、2つの範囲の大きさの比に基づいて計算されます。

例) `int x= map(analogRead(A0),0,1023,0,160);`



ステップアップ

プログラム「VolumeLCD2」を変更して、**A1** に接続されている可変抵抗ボリュームを使ってy軸方向にも動かせるようにしてみよう。

少しの変更で大幅に動きが変わりましたね！



解答プログラムは以下となります。

RoboticsProfessorCourse3 > MagicItemLCD2 > VolumeLCD3

チャレンジ課題

「VolumeLCD2」をさらに書きかえて、可変抵抗ボリュームのつまみを回すと画面の背景色が変わるプログラムにしてみよう！

なお、変更したプログラムは以下になります。



プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > VolumeLCD4

2.3. 超音波距離センサーの情報を表示させる

1) 超音波距離センサーを接続させる

最後は超音波距離センサーです。次のように、超音波距離センサーを接続します。図2-5を見ながら配線してください。

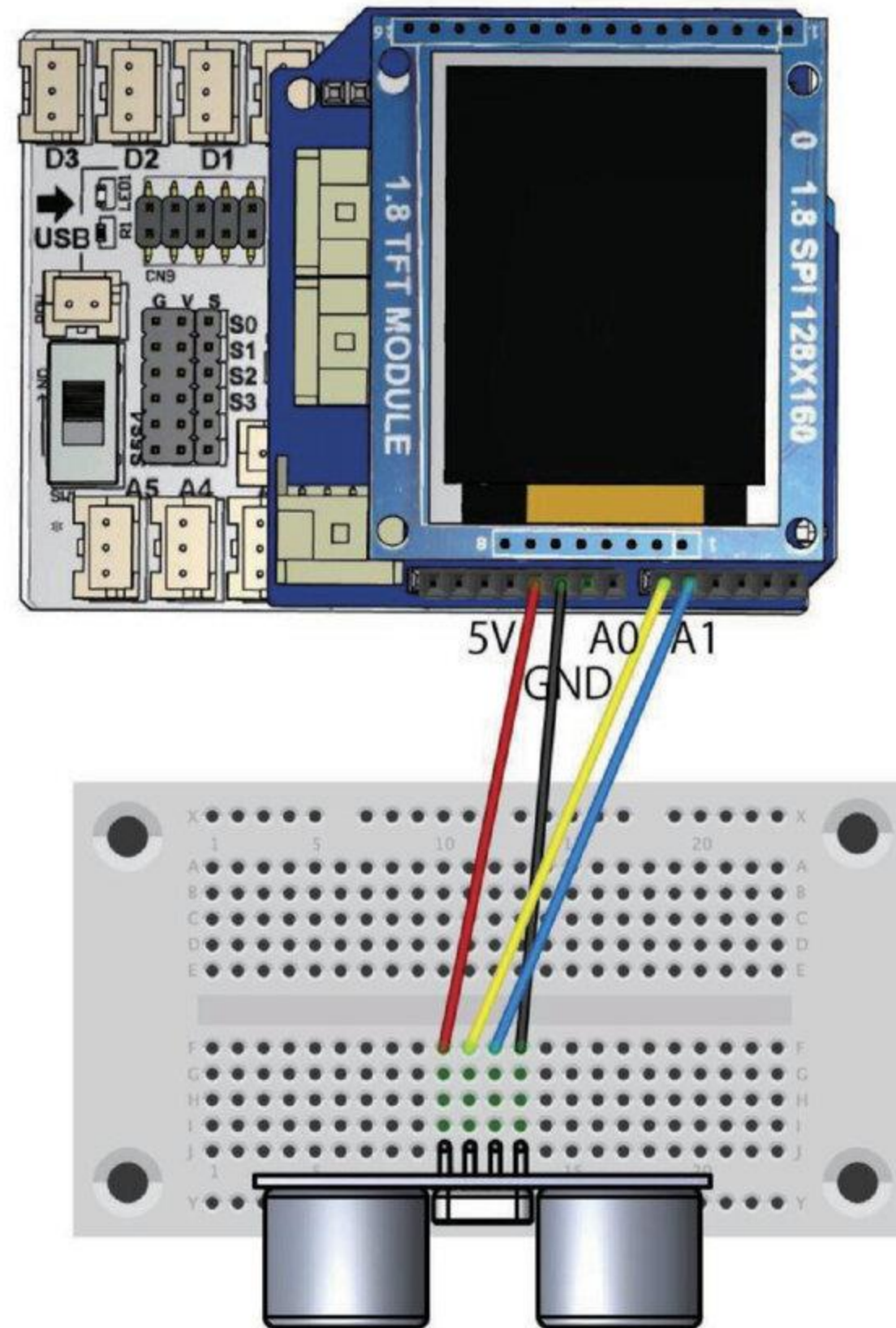


図2-5 超音波距離センサーの接続

講

回路を接続して反応が無い場合、ジャンパー線の接続や断線の確認をしてください。

2) ボリューム (距離) を読み取る

配線できたら、次のプログラムを実行してください。

プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > UltrasonicLCD1

超音波距離センサーに手をかざしてみると、距離情報が液晶ディスプレイに表示されます。0～200cmくらいの間で変化します。

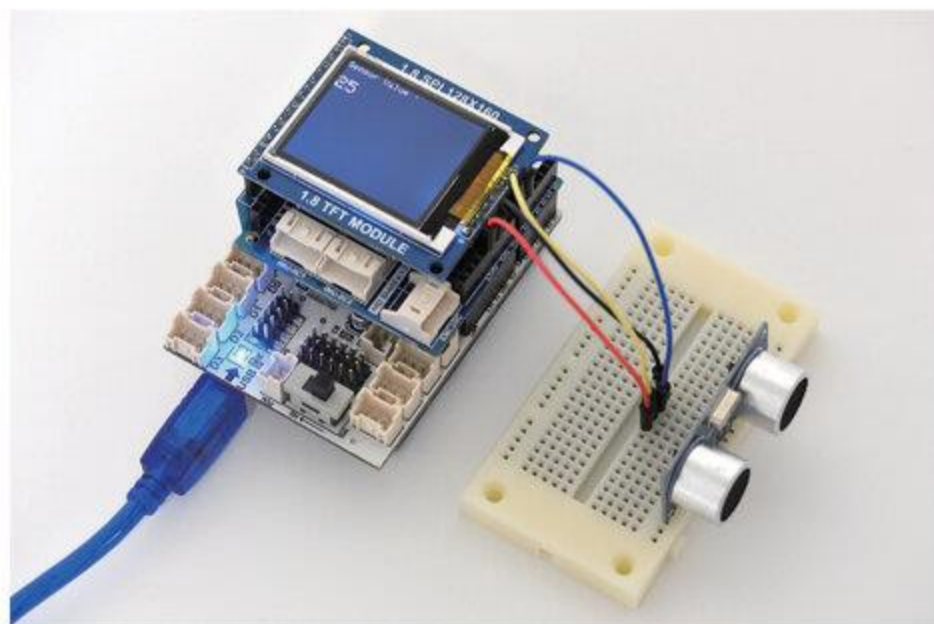


図2-6 距離情報の表示

3) グラフを表示させる

続いてグラフを表示させましょう。次のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > UltrasonicLCD2

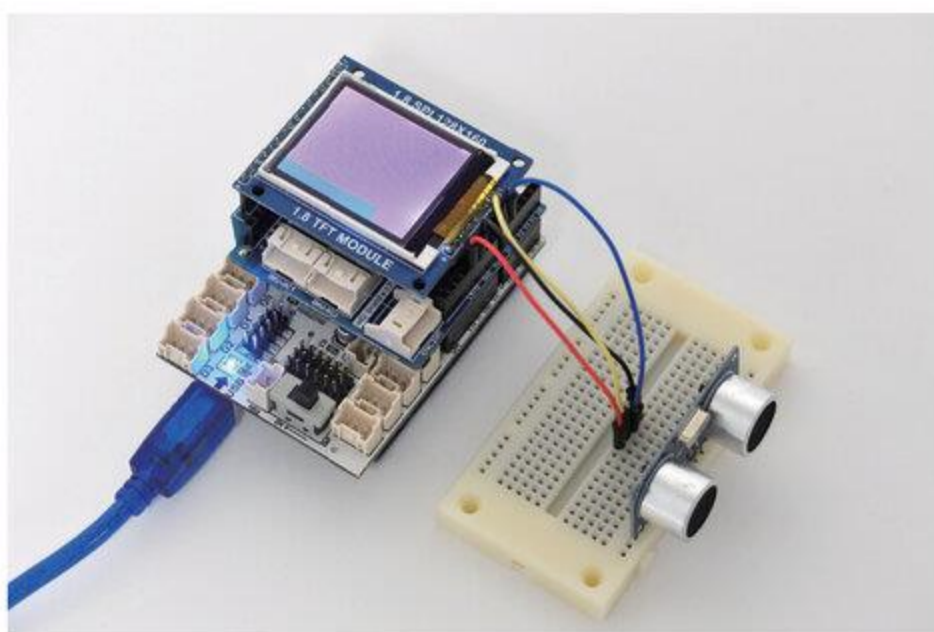


図2-7 グラフの表示

実行結果：超音波距離センサーの読み取った数値がグラフで表示される。数値の変化が読み取りやすくなりましたね。

4) バー表示させる

続いてバーを表示させます。次のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > UltrasonicLCD3

実行結果：超音波距離センサーの読み取った数値がバーで表示される。距離が視覚的に読み取りやすくなりましたね。

5) 円を表示させる

最後は円を表示させます。次のプログラムを実行してください。

∞ プログラムの書き込み

RoboticsProfessorCourse3 > MagicItemLCD2 > UltrasonicLCD4

実行結果：超音波距離センサーの読み取った数値が円の大小で表示される。
距離1つとっても様々な表現方法がありますね。

やってみよう!

センサーの値に応じ、大きさが変化するような四角を表示させてみよう。プログラム「UltrasonicLCD4」を参考に作ってみてね。

講

解答プログラムは以下となります。

RoboticsProfessorCourse3 > MagicItemLCD2 > UltrasonicLCD5

いかがでしょうか？ 自分なりに、「こうだったらわかりやすいのでは？」というイメージは湧いてきましたか？ 人間には、驚くような方法を生み出す能力があります。最近のエンジニアリングには、このようなデザインや人間工学といった概念も必要になってきています。誰が見てもわかるというデザインは、最近では「ユニバーサルデザイン」と呼ばれて、年齢、言語、知識に関係なく直感的にわかるデザインとして注目されています。興味がある人は、調べてみましょう。

講

constrain

constrain() は、“数値をある範囲に制限する” というときに使います。

constrain(x, a, b)

x 範囲を制限したい数値 (任意の型)

a 範囲の下限値 (任意の型)

b 範囲の上限値 (任意の型)

という使い方になります。

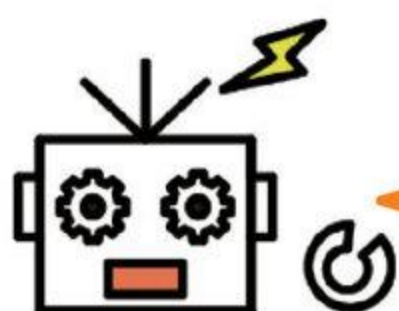
3. まとめ (目安 5 分)

今回は、復習も兼ねて、今まで使ってきたセンサーの情報をもとに、液晶ディスプレイにいろいろな形を描いてみました。数字だけのときよりも、直感的にわかりやすくなる部分もあったかと思います。これが、俗にいう「GUI (グラフィカルユーザインターフェース)」というものです。

昔は、数字の結果と、みなさんの今使っているプログラム (文字の羅列) だけだったのですが、現在は、Windows、OSX、Linux…と「GUI」の世界が広がっています。しかし、今回わかったように、その根底には、文字ばかりの世界 (CUI: キャラクタユーザインターフェース) があることを覚えておいてください。その世界のことを覚えておくことは、将来GUIのプログラムをするときに大きなアドバンテージとなりえます。

実際のところ、GUIプログラミングを行うと、今回のプログラムよりも簡単にできる部分もありますが、それは先人たちが苦勞をしてきたおかげです。その苦勞の一端を知った上で、さらに高度なプログラミング技術を学んでほしいと思います。

さて、次回は、さらに複雑な三次元プログラムを学んでいきます。複雑といっても、難しい作業はマイコンが行ってくれます! 人の手で行うと大変なことでも、マイコンはさらりとこなしてくれるのです。



さらなる未知の領域にチャレンジだよ!

《次回必要なもの》



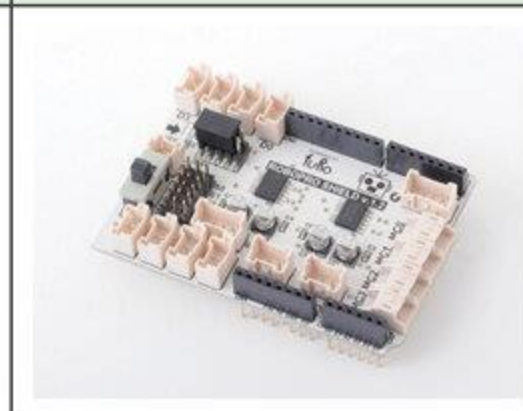


USBケーブル 1	マイコンボード 1	ロボプロシールド 1	姿勢検出シールド 1
			
液晶ディスプレイシールド 1			
			

図3-0 次回必要なもの

講

- 今回の理解度を確認しましょう。
- ・液晶ディスプレイに文字を表示させる
- ・液晶ディスプレイにセンサー情報を表示させる
- ・液晶ディスプレイにインタラクティブな表示をさせる

○次回は、「メモリーとビット演算」です。