E-Station へようこそ

目 次

はじめに　E-Station へようこそ

0-1　本書の使い方 6
0-2　E-Station の構成 7
0-3　E-Station の操作方法 9
0-4　E-Station 仕様 11
0-5　設定値の保存と呼び出し 12

第 1 部　E-Station の使い方

実験 1-1 E-Station を使ってみよう 14
実験 1-2 各端子の機能を使ってみよう 18

第 2 部　電子回路基礎

実験 2-1 電圧を測定しよう 22
解説 2-2 電圧とは？ 24
解説 2-3 電気をあらわす単位と記号 25
実験 2-4 電流を測定しよう 26
解説 2-5 オームの法則 28
解説 2-6 電気の源 29
解説 2-7 直流・交流・信号 30
実験 2-8 分圧回路 31
実験 2-9 分流回路 33
解説 2-10 この抵抗、何オーム？ 35
解説 2-11 合成抵抗を計算してみよう 36
実験 2-12 コンデンサの充電と放電 37
解説 2-13 コンデンサのはたらき 39
実験 2-14 CR 回路と過渡現象 40
解説 2-15 信号と周波数 43
解説 2-16 電磁力 44
実験 2-17 コンデンサとコイルの周波数特性 46
実験 2-18 コンデンサとコイルの共振回路 48

©MCIC Inc.

第3部　半導体回路の基本

実験 3-1 ダイオードの順方向特性 52
実験 3-2 ダイオードの逆方向特性 54
実験 3-3 発光ダイオード（LED）を調べる 56
実験 3-4 バイポーラ・トランジスタの B-E 特性 . . 58
実験 3-5 BJT 増幅回路 60
解説 3-6 半導体のしくみ 62
実験 3-7 エミッタ接地増幅回路 68
実験 3-8 スピーカーを鳴らす 70
コラム データシートを見てみよう 72
実験 3-9 スピーカー音を大きくする 73
解説 3-10 トランジスタ発明の意義 75
解説 3-11 BJT と MOSFET 76

第4部　アナログ技術

実験 4-1 オペアンプを動かす 78
実験 4-2 加算回路 82
解説 4-3 信号のかたち 84
実験 4-4 2 音の合成 86
実験 4-5 さまざまな演算回路 88
実験 4-6 ボルテージフォロワ 92
コラム 電子回路の“タンス” 95
解説 4-7 オペアンプの基本動作 96
実験 4-8 コンパレータ 98
実験 4-9 光センサー（CdS フォトセル）を使う . . 100
コラム センサー制御とマイコンプログラミング 101
実験 4-10 暗くなると点灯するライト 102
実験 4-11 明るさの差を判別する 104
実験 4-12 マイク信号の増幅 105
実験 4-13 ホールセンサーで磁気測定 108
実験 4-14 超音波センサーで送受信 109
実験 4-15 配置・配線によるノイズの発生 112
解説 4-16 活躍する半導体製品群 117

©Magic Inc.

第 5 部　デジタル技術

実験 5-1 7 セグメント LED を使う 120
解説 5-2 論理回路 124
解説 5-3 デジタルデータの単位・ビット 126
実験 5-4 2 ビットデコーダ回路 127
実験 5-5 メモリセルを作る 129
解説 5-6 さまざまな論理回路 131
実験 5-7 NOT ゲート（インバータ）を作る 132
実験 5-8 NAND ゲート・NOR ゲートを作る 134
コラム パワー MOSFET の構造 137
実験 5-9 標準ロジック IC を使う 138
解説 5-10 危険なラッチアップ現象 141
実験 5-11 NOT ゲート IC 142
実験 5-12 NAND ゲート IC 144
実験 5-13 OR ゲート IC 146
実験 5-14 XOR ゲートを作る 148
実験 5-15 RS フリップフロップ 150
実験 5-16 D フリップフロップ 152
解説 5-17 組み合わせ回路・順序回路 154
実験 5-18 分周回路 155
実験 5-19 4 ビットバイナリカウンタ 158
実験 5-20 4 ビットアップダウンカウンタ 160
解説 5-21 ハザード・レーシング・チャタリング . . 163
実験 5-22 10 進カウンタ 165

第 6 部　AD/DA 変換

実験 6-1 デジタルーアナログ変換（DAC） 170
解説 6-2 D/A 変換器と分解能 176
実験 6-3 アナログーデジタル変換（ADC） 177
解説 6-4 サンプリング（標本化）・量子化・符号化 . 181

第 7 部　演算装置

解説 7-1 コンピュータ（計算機） 184
実験 7-2 ハーフアダー回路 185
実験 7-3 フルアダー回路 188
実験 7-4 4 ビット加算器 191
コラム ロボット 194
実験 7-5 レジスタと加算器 195

©MC Inc.

はじめに
E-Station へようこそ

0-1 本書の使い方 …… 6
0-2 E-Station の構成 …… 7
0-3 E-Station の操作方法 …… 9
0-4 E-Station 仕様 …… 11
0-5 設定値の保存と呼び出し …… 12

©MGIC Inc.

0-1 本書の使い方

さあ、実験を始めましょう！

本書は、E-Station（イーステーション）を使って電子回路の初歩から応用までを実践・学習するための**実験用テキスト**です。

本テキストは**難解な数式がなるべく登場しないように作られています**。実験を通して多様なエレクトロニクス技術の基礎に触れ、電子回路のふるまいを理解する助けとなることを願っています。

実験ページについて

①学習ポイント

実験ごとのテーマや着目する点について掲載しています。

②チェック

実験時の注意点や、使用するパーツについての解説を掲載しています。

③回路を作ろう！

実験回路の回路図・接続図を掲載しています。接続図は参考例ですので、**回路図通りに作成できるなら接続図とまったく同じ配置にする必要はありません**。

④ヒント

実験時のヒントを掲載しています。

⑤実験のすすめ方

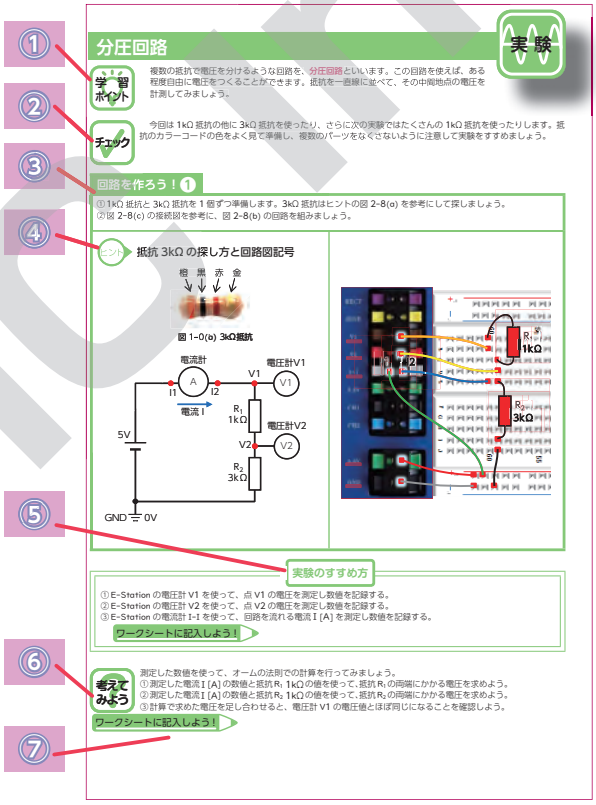
回路を作った後に行う実験手順や記録する内容を掲載しています。

⑥考えてみよう

実験結果を振り返り、考察する内容を掲載しています。

⑦ワークシートに記入しよう

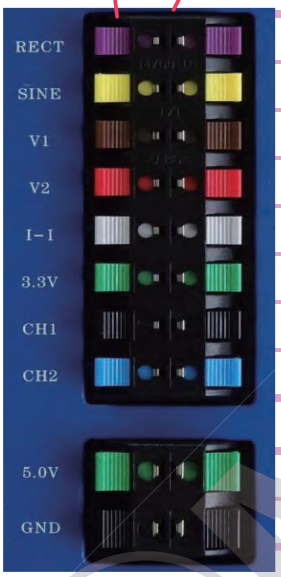
このアイコンは、添付 CD 内にあるワークシートに計測結果や考察を記録することを意味します。ワークシートデータは別途印刷してご利用ください。



©MCI Inc.

0-2 E-Station の構成

E-Station では**ブレッドボード**だけではなく、実験をスムーズにすすめるための**測定機能**や**電源機能**を同時に使う事ができます。



RECT	矩形波の出力
SINE	正弦波の出力
V1	直流電圧測定①
V2	直流電圧測定②
I-I	直流電流測定
3.3V	直流電源
CH1	オシロスコープ①
CH2	オシロスコープ②
5.0V	直流電源
GND	グランド(接地)

注意①　端子台の左右は内部でつながっています。I-I 端子のみ、左(入力)、右(出力)の違いがあります。

注意②　端子台の GND は信号・電源・計測器の GND と内部でつながっています。

注意③　ブレッドボードは端子台とつながっていません。

注意④　**GND 端子・5V 端子・3.3V 端子はそれぞれ直接つないではいけません！**

「電源をオンにしているのにメイン画面が暗くなる」などの異常が起きたときは、**本体の電源をオフにし、回路の接続にまちがいがないかを確認してください。**

① 電源スイッチ
E-Station の電源のオン / オフを切り替えます。
※実験回路を作るときは、必ず電源をオフにするか、電源・GND とブレッドボードの接続を外して行いましょう！

② MENU ダイヤル
メイン画面に表示する設定モードを切り替えます。

③ RANGE ダイヤル
メイン画面の表示形式を切り替えます。
計測の細かい設定も行います。

④ 実験・測定用端子台
様々な機能を持った端子が並んでいます。回路と端子をつないで使います。詳しい使い方は **9 ページ「E-Station の操作方法」**を参考にしてください。

⑤ 電源・GND（グランド）端子台
電源の端子と GND（グランドと読みます）の端子です。

⑥ メイン画面
測定した電圧や電流の表示ができる、オシロスコープの画面です。

⑦ ブレッドボード
電子パーツをさしこみ、回路を作るボードです。

⑧ パーツ BOX
フタを上にスライドさせて開けます。

⑨ マイコンスペース
組み込みプログラミングを学習するためのマイコンを置くスペースです。

©MSC Inc.

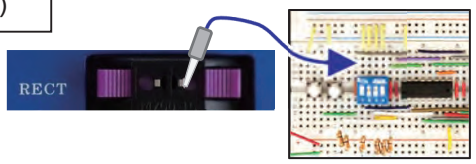
端子台の使い方

端子台と回路のつなぎ方です。
矢印の方向は入力か出力かを意味します。

信号や電流を端子へ入れることを入力と呼び、信号や電流を端子から出すことを出力と呼びます。

RECT 端子（矩形波）

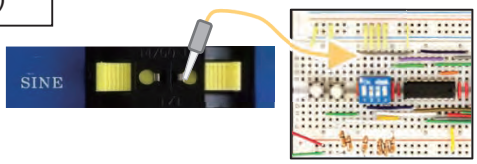
RECT 信号（矩形波）を回路に送ることができます。



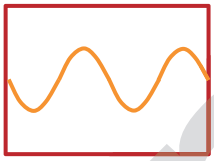
矩形波（RECT）のかたち

SINE 端子（正弦波）

SINE 信号（正弦波）を回路に送ることができます。

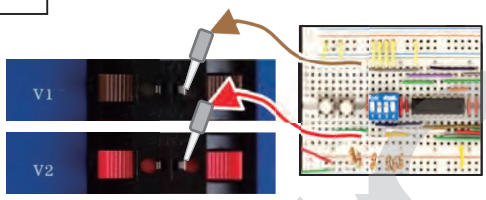


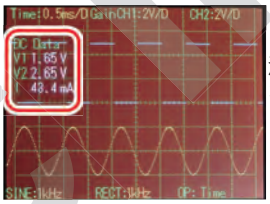
正弦波（SINE）のかたち



V1（V2）端子（電圧計）

つないだ部分の電圧の高さを測定することができます。

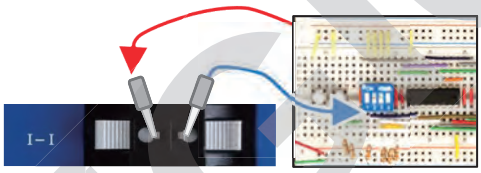


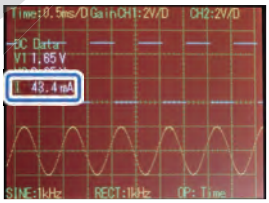


測定電圧は、「DC Data」を表示したときに「V1」「V2」で表示されます

I-I 端子（電流計）

電気の通り道をつくり、つないだ部分を流れる電流を測定することができます。

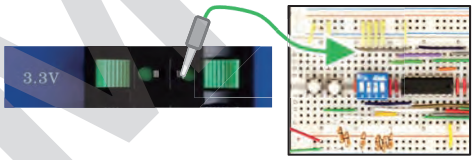




測定電流は、「DC Data」を表示したときに「I」で表示されます。

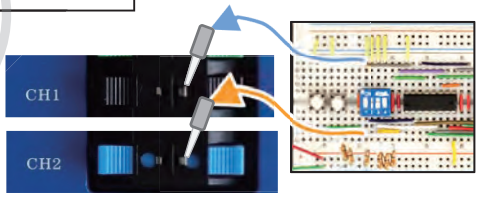
3.3V 端子（電源）

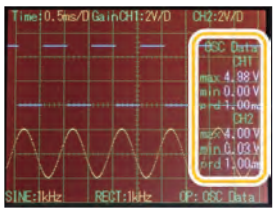
つないだ部分に 3.3V の電源を供給します。



CH1（CH2）端子（オシロスコープ）

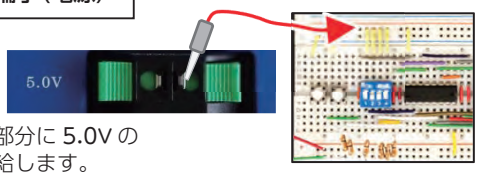
つないだ部分の信号のかたちをオシロスコープ CH1(CH2) に表示します。
CH1 が青色の波形
CH2 が黄色の波形です。





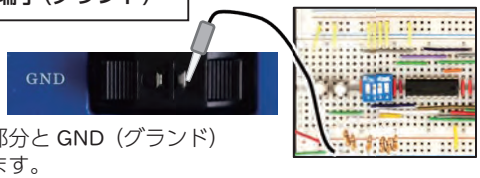
「OSC Data」画面で信号の max（最大の電圧値）、min（最小の電圧値）prd（周期）を確認できます。
「AMP/FREQ」画面では信号の amp（振幅）、freq（周波数）を確認できます。

5.0V 端子（電源）



つないだ部分に 5.0V の電源を供給します。

GND 端子（グランド）



つないだ部分と GND（グランド）をつなぎます。

©MCI inc.

0-4 E-Station 仕様

付 録

①システム電源：商用電源(AC100 ～ 240V)から AC アダプタを使い直流 5V を E-Station に供給します。
- POWER スイッチを押すことにより、電源が E-Station に供給されます。
  基本機能動作時の消費電流　約 200mA

②グランド(GND)：端子台の GND 端子がシステム共通の GND です。

③実験用直流電源
- 直流 5V 電源　端子台 5V 端子より供給　AC アダプタの 5V 電源を使用　電流供給能力　1A
- 直流 3.3V 電源　端子台 3.3V 端子より供給　電流供給能力　200mA

④直流電圧計・電流計
- 直流電圧計　V1、V2 端子に接続し、接続点の直流電圧を測定　　測定電圧範囲 0V ～ 5V
- 直流電流計　I-I 端子に電流の入力、出力を接続し測定　　内部抵抗 30Ω　　測定電流範囲 0.01mA ～ 100mA

⑤信号発生器
- SINE（正弦波）と RECT（矩形波）を各端子から同時に出力(両信号は同期していません)
- SINE（正弦波）　信号レベル 0V ～ 4V　周波数調整 10Hz ～ 76kHz　　出力駆動能力 25mA
- RECT（矩形波）　信号レベル 0V ～ 5V　周波数調整 1Hz ～ 1MHz　　出力駆動能力 8mA(L) ～ 4mA(H)

⑥オシロスコープ
- CH1 と CH2 の 2 現象観測　　時間軸設定 10μs/D ～ 50ms/D　　入力インピーダンス 1MΩ 以上
- ゲイン設定 20mV/D ～ 4V/D
- CH1　青色波形　　CH1 端子の入力信号を検出、トリガーレベルは信号の中央を自動で設定します。
- CH2　黄色波形　　CH2 端子の入力信号を検出、CH1 の信号でトリガーをかけます。

■グランド“GND”について

E-Station ではグランド(Ground)を GND とあらわします。GND の電圧は基準の 0V です。直流電源 5V と言う場合、この GND (0V) を基準にした電圧です。

実験テーマの回路図、接続図の基準点も GND で、E-Station の端子台の GND とつなぎます。回路図では GND を記号 ⏚ であらわします。

■設定を行える範囲とメインモニター表示

MENU ダイヤル	RANGE ダイヤル
Op：Menu	Menu (初期画面) DC Data(電圧計・電流計) OSC Data(波形の最大値・最小値・周期) AMP/FREQ(波形の振幅・ピークトゥピーク値・周波数)　LCD reset(画面リフレッシュ)
Time	10μs　20μs　50μs　0.1ms　0.2ms　0.5ms 1ms　2ms　5ms　10ms　20ms　50ms
Gain CH1	4V/D　2V/D　1V/D　0.5V/D　0.25V/D　125mV/D　63mV/D　20mV/D
Gain CH2	4V/D　2V/D　1V/D　0.5V/D　0.25V/D　125mV/D　63mV/D　20mV/D
Pos CH1	CH1 波形位置の下降 / CH1 波形位置の上昇
Pos CH2	CH2 波形位置の下降 / CH2 波形位置の上昇
SINE C	10Hz　20Hz　40Hz　100Hz　200Hz　400Hz 1kHz　2kHz　4kHz　10kHz　20kHz　40kHz　76kHz
SINE F	SINE 周波数の上昇 / SINE 周波数の下降(微調整)
RECT C	1Hz　2　5　10　20　50　100　200　500 1K　2　5　10　20　50　100　200　500K　1MHz
RECT F	RECT 周波数の上昇 / RECT 周波数の下降(微調整)

©MIC inc.

0-5 設定値の保存と呼び出し

設定値セーブ (保存機能)

メニュー画面でオレンジ色 (RANGE) のダイヤルを左に回すと、「LOAD/SAVE」画面が表示されます。

この画面で青色 (MENU) のダイヤルを右に回すと、設定値のセーブ (保存) が行われます。セーブした設定値は E-Station の電源を切った後も保存され、次にロードを行ったときに同じ設定が再現されます。

設定値ロード (呼び出し機能)

「LOAD/SAVE」画面で青色 (MENU) のダイヤルを右に回すと、設定値のセーブ (保存) が行われます。セーブした設定値は E-Station の電源を切った後も保存され、次にロードを行ったときに同じ設定が再現されます。

セーブ (保存)/ ロード (呼び出し) 機能では Time(時間軸)・Gain(縦軸)・Pos(描画位置)・SINE(正弦波周波数)・RECT(矩形波周波数) の値が保存 / 呼び出しされます。CH1 と CH2 の両方の設定が対象となります。

©MSO Inc.

第 1 部
E-Station の使い方

実験 1-1 E-Station を使ってみよう …… 14
実験 1-2 各端子の機能を使ってみよう …… 18

©MGIC Inc.

1-1 E-Station を使ってみよう

実験

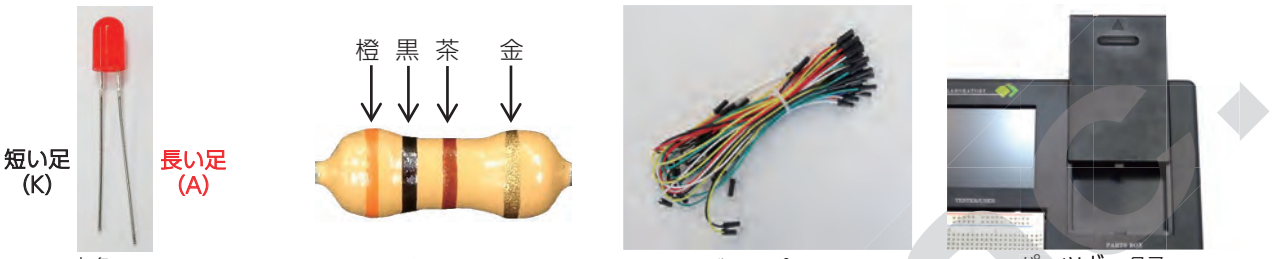
学習ポイント

いざ電子回路を学ぼうとしたとき、「どこで、誰から学ぶか？」「何を用意したらいい？」「パーツの使い方は？」「何の実験をしようか？」といった障害にぶつかってしまうことでしょう。しかし、心配いりません。E-Station とテキスト、電子パーツを使って気軽に実験ができることを体感しましょう。

チェック

実験を行う前に必要なパーツを準備しましょう。回路を作る前に、使うパーツをひととおり取り出しておくと、効率よく進めることができます。実験で使うものを図 1-1(a) に示します。また、本体右上のフタを開けるとパーツボックスとして利用できますので、ぜひ活用してください。

※ LED や**抵抗**についての詳細な解説は**第 2 部以降**で行います。**まずは回路を作ってみましょう。**



赤色 LED　固定抵抗器 300 Ω　ジャンパー線　パーツボックス

図 1-1(a) 実験で使用するパーツ

■電子回路とブレッドボード

電気の性質でとりわけ重要なのは、**金属と金属が接触すると、つながって電気の通り道になる**という事です。電気の通り道になる電線を**導線**と呼び、導線によって電気がひと回りする道のりを**回路**と呼びます。

回路をつくるために導線同士を接触させるには、**ブレッドボード**を使うと便利です。小さな穴がたくさん開いている板で、ほんの少し構造を理解すれば簡単に扱うことができます。

図 1-1(c) のように、ブレッドボードの中には金属の板が入っているため、部品をさしこむと金属板が接触して他の穴とつながります。

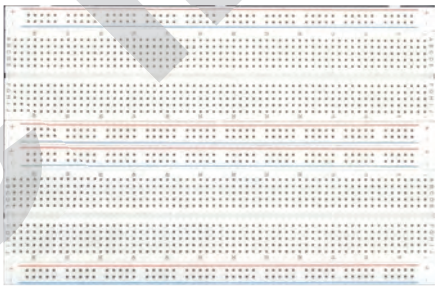


図 1-1(b) ブレッドボード

ブレッドボードの構造

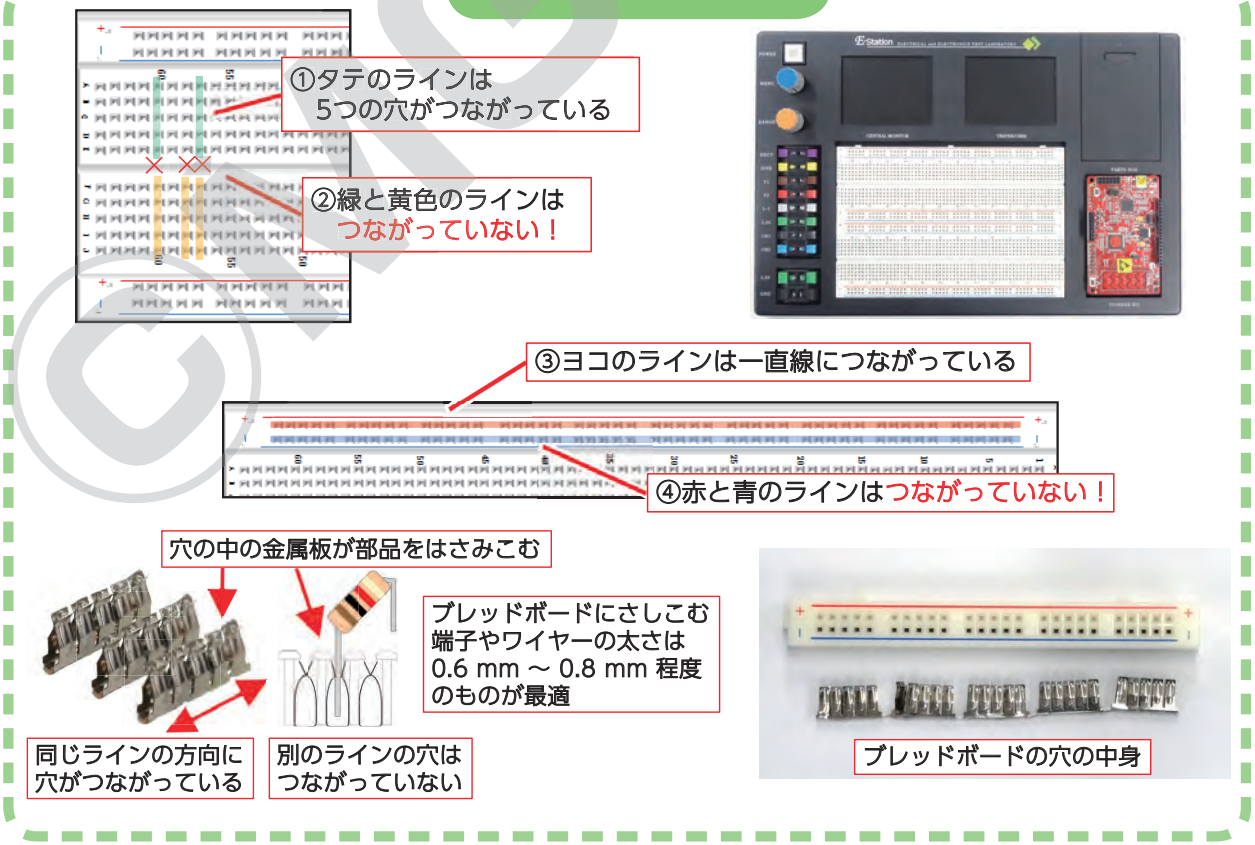


図 1-1(c) タテとヨコのライン

電気回路の設計図を**回路図**と呼び、**回路図上で表現される**電子パーツの記号を**回路図記号**と呼びます。接続方法を変えるだけで、電子回路が動かなくなったり素子が壊れたりする可能性があるため、回路図の配置をしっかりと確認しながらパーツ同士をつなげていくことが大切です。

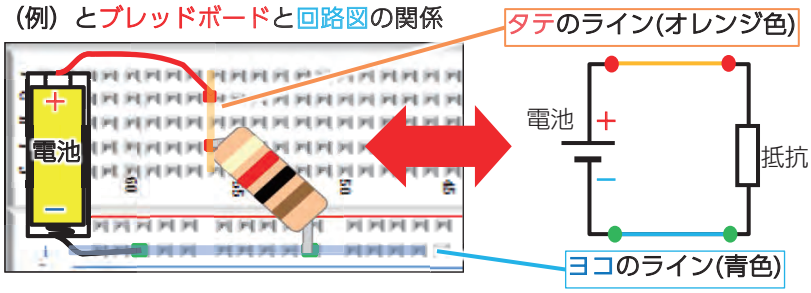


図 1-1(d) **回路のつながりをイメージする**

回路を作ろう！ 1

最初に、LED（発光ダイオード）に電気を流して点灯させる実験を行います。
①**ブレッドボード**にさしこむパーツやワイヤーを図 1-1(a) を参考に準備しておきましょう。
②接続図上で色を付けている穴が配線をさしこむ位置です。まずはこの**接続図の通りに**配線を接続しましょう。
③回路図上の 5.0V 電源のプラス (+) 側と 0V(GND) 側は端子台の「5.0V」「GND」表記のある端子を使用します。

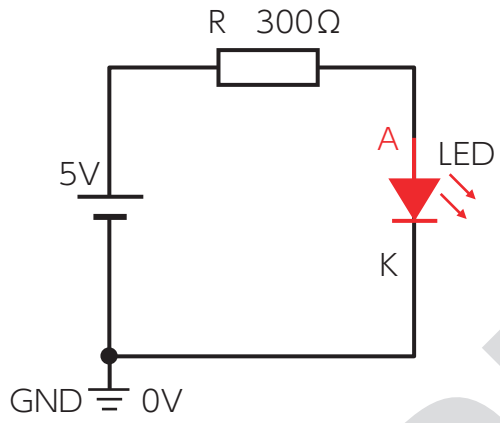


図 1-1(e) **回路図**

図 1-1(f) **接続図**

ヒント 端子台の接続方法

「端子台」　「指で押さえて」　「はさみこむ」

図 1-1(g) **端子台の接続方法**

実験のすすめ方

① E-Station の電源を入れ、E-Station が起動することを確認する。
②モニターが真っ白な表示のままだったり、モニターが表示されたり消えたりと不安定な場合は**端子の接続を間違えている可能性**があります。**すぐに電源を切り、正しい接続を行ってください。**
③赤色 LED が点灯していることを確認する。**点灯しない場合はすぐに電源を切り**、接続の間違いやパーツ・配線に故障がないか確認してください。（間違った配線を行うと、LED が壊れる可能性があります）

補足 5.0V 端子と 3.3V 端子

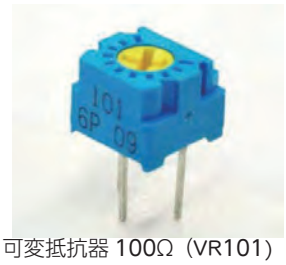
E-Station の 5.0V 端子や 3.3V 端子は**「電源」**として機能する端子で、回路に電気のエネルギーを供給するみなもと（源）となります。
5.0V 端子や 3.3V 端子は常に約 5V、約 3.3V の電圧を維持しようとする**定電圧源**と呼ばれるタイプの電源です。

チェック

次に、作成した電子回路にパーツを追加して、LED の明るさを調整する機能を設けてみましょう。

可変抵抗器(VR)を準備し、回路に追加します。使用する可変抵抗器には側面に「101」という数字が印字されています。

可変抵抗器の VR とは、「抵抗値(Resistance)を可変できる(Variable)」という意味から付けられている名前です。抵抗値についての解説は第 2 部以降に行います。まずは VR のダイヤルを操作して、LED の明るさがどのように変化するかを観察しましょう。



可変抵抗器 100Ω（VR101）

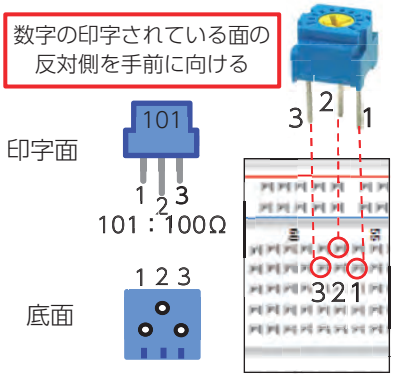


図 1-1(h) 可変抵抗器

回路を作ろう！2

①図 1-1(h) に示す可変抵抗器を準備しましょう。
②図 1-1(j) の接続図を参考に、図 1-1(i) で色付けした位置に可変抵抗器を追加し、配線を変更します。
③図 1-1(k) のヒントを参考に、**最初に VR のダイヤルを左いっぱいに回しておきます。**

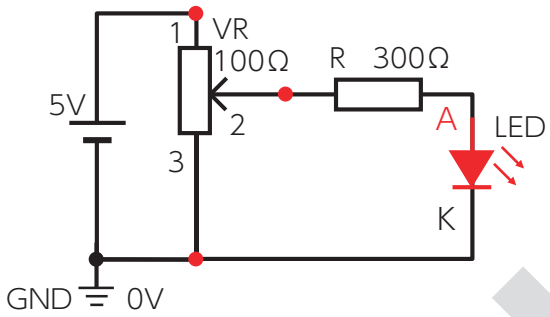


図 1-1(i) 回路図

図 1-1(j) 接続図

ヒント

可変抵抗器 VR のダイヤルを回そう

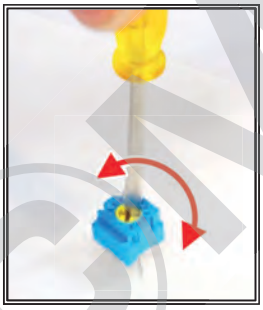


図 1-1(k) ダイヤルの回転方向

- 図 1-1(k) のように、マイナスドライバー等をみぞにさしこむと中心のダイヤルを左右に回すことができます。
- 図 1-1(l) のように、ダイヤルの矢印(→)が VR の位置を示しています。

最大まで回した後に止まったら、それ以上回しすぎないように注意してください！

左いっぱいに回している　中心の位置にある　右いっぱいに回している

図 1-1(l) ダイヤルの矢印

実験のすすめ方

① E-Station の電源を入れ、E-Station が正常に起動することを確認する。
② VR のダイヤルを回転させて、LED の点灯の変化を観察する
③ VR のダイヤルの矢印を左・右のどの方向に回すと LED が明るくなるかを確認する。

考えてみよう

①LED が点灯するのは VR のダイヤルを左・右どちらに回したときでしたか？
②VR のダイヤルが中心を向いているとき、LED はどのような明るさでしたか？ダイヤルを右いっぱいに回したときと比べて明るかったですか？暗かったですか？

ワークシートに記入しよう！

©MSC

回路図の描き方と回路図記号

補足

回路図の描き方は図 1-1(m) を参考にしてください。
電子回路の表現の仕方さえわかれば、誰でも描くことができますよ！

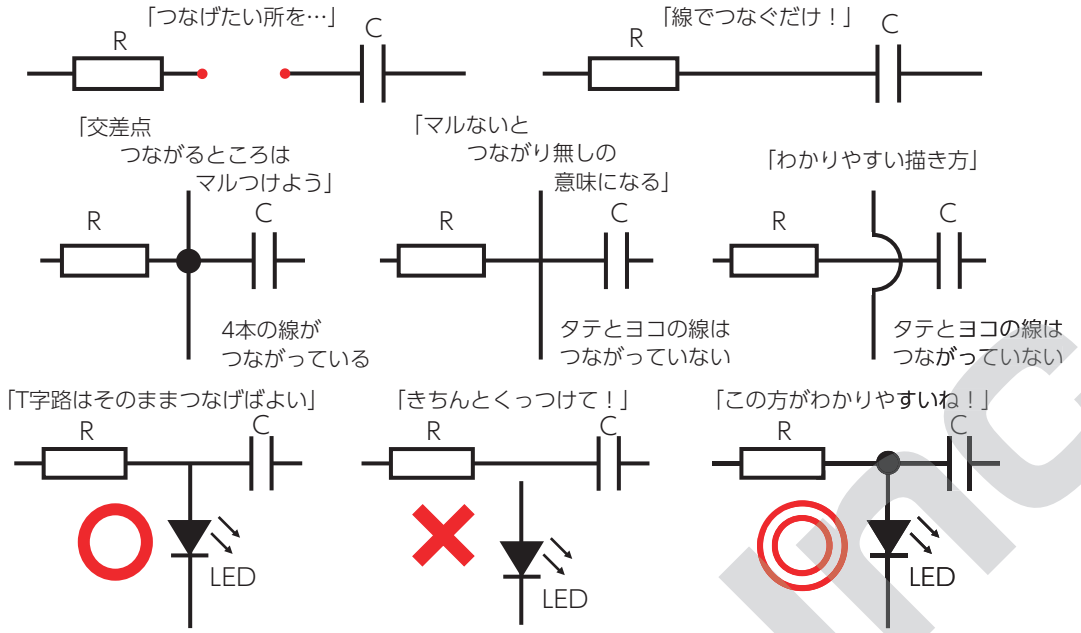
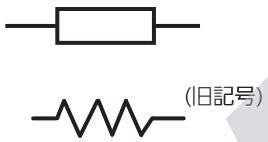


図 1-1(m) 回路図の描き方

回路図記号をよく見てみると、実際の電子パーツのかたちと似ているものがたくさんあります。
中には姿が似ていないものもありますが…そういうものは、外見ではなく中身の構造や機能をあらわしていたりします。

抵抗

呼び方
抵抗 / レジスタンス / R など
回路図記号



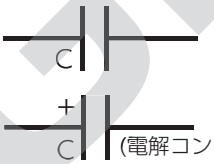
こんな形があります

カーボン抵抗　　チップ抵抗

- 電流を制限する
- 大きな電流から回路をまもる
- 好きな電圧をつくる
- 抵抗値を変えられるものもある
- 温度で抵抗値が変わるものもある

コンデンサ

呼び方
コンデンサ / キャパシタ / C など
回路図記号



(電解コンデンサ
などの有極性のもの)

こんな形があります

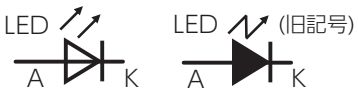
積層セラミック
コンデンサ　　電解コンデンサ
(有極性)

チップコンデンサ

- 電気をたくわえたり、放出したりする
- 電圧の変化を鈍らせる
- **直流をカットする**
- 容量を変えられるものもある
- むだな電力をなくして省エネができる

発光ダイオード

呼び方
発光ダイオード / LED など
回路図記号



電流を流せる方向 (極性)があります
(A側:アノードからK側:カソードへ流れる)

こんな形があります

赤色発光ダイオード　　チップ LED

- 電流を流すと発光する
- 電光掲示板や表示器として使う
- 照明として使う

図 1-1(n) 抵抗・コンデンサ・LED

これらは電子パーツのほんの一部です。国や時代、学校や会社によって使われている回路図記号が、違っていたりすることもあるでしょう。見たことのない描き方を見つけてもあわてずに、調べてみれば「なーんだ、あのパーツか」と思うこともよくあります。

©MCR Inc.

1-2 各端子の機能を使ってみよう

実験

学習ポイント

E-Station には電子回路の実験を簡単に行うための機能が搭載されています。電圧計・電流計・オシロスコープ・信号発生器など、今後これらの機能を活用できるように、まずは使ってみましょう。

チェック

電圧計は、回路上のある 1 点の電圧値を測定する機能で、**電流計**は回路上の 2 点間を流れる電流値を測定します。そして**オシロスコープ (Oscilloscope)** とは、電気の波 (振動) を表示する機能です。

E-Station はこれら測定機能の他にも、**信号発生器**と呼ばれる機能をもっています。信号発生器から出る電気の波をオシロスコープで観測してみましょう。

固定抵抗器 1000 Ω (1k Ω)
4 本使用

図 1-2(a) 実験で使用するパーツ

回路を作ろう！

① E-Station の各端子台を使って回路を 4 種類作成します。
② 図 1-2(b) の接続図通りに配線を行い、回路を作りましょう。
③ この実験によって、E-Station の端子台の各機能 (電圧計・電流計・オシロスコープ・信号発生器) の動作をチェックすることができます。

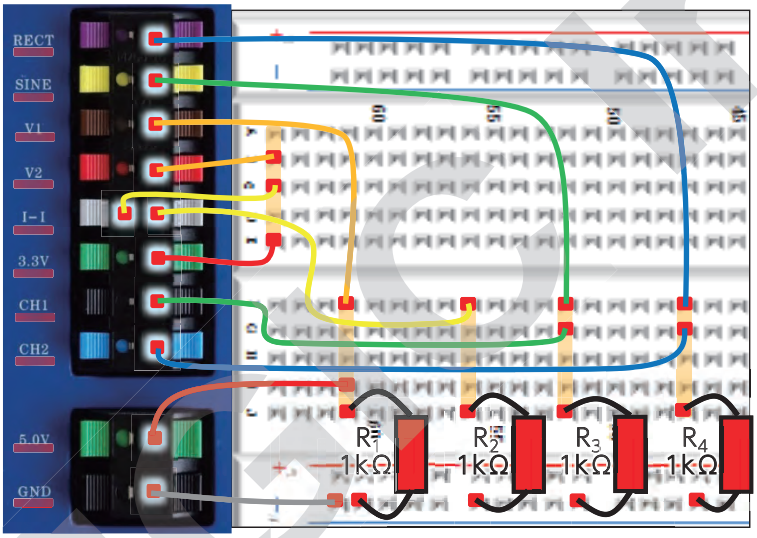


図 1-2(b) 接続図

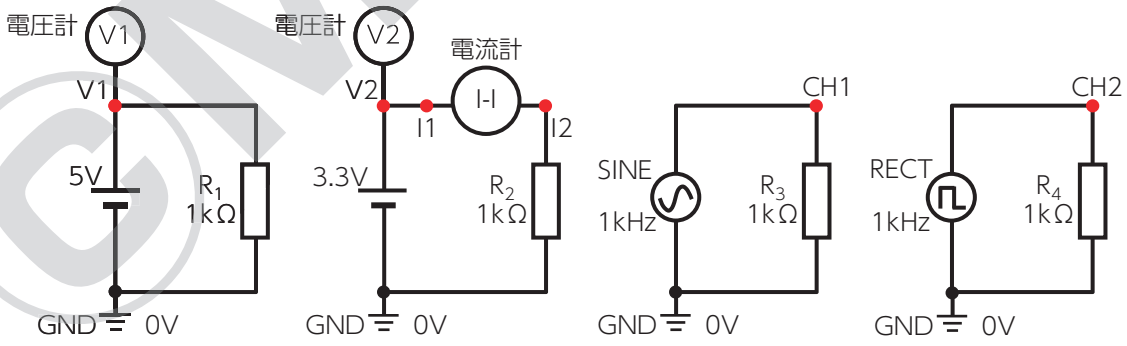


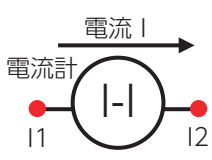
図 1-2(c) 回路図

ヒント

電流計の使い方

電流計は電圧計やオシロスコープ端子とは使い方が異なり、**電流を測定したい 2 点間に埋め込む**ようにして使用します。

図 1-2(d) のように、I1 端子から I2 端子までを流れる電気の量を測定することができます。



I-I 端子の左側→● I1
I-I 端子の右側→● I2
電流計には方向があり、
電流が流れる向きを意識して使います。

図 1-2(d) 電流計

©MSCNC

実験のすすめ方

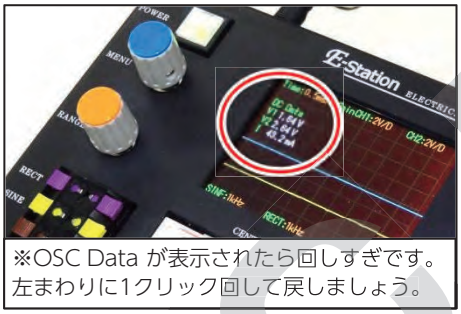
① E-Station の電源を入れ、E-Station が起動することを確認する。
②モニターが真っ白な表示のままだったり、モニターが表示されたり消えたりと不安定な場合は**端子の接続を間違えている可能性**があります。**すぐに電源を切り、正しい接続を行ってください。**
③図 1-2(e) に示すヒントや [0-3]「E-Station の操作方法」を参考に、「DC Data」画面を表示させて電圧 V1・電圧 V2・電流 I の**値を記録**する。
④同様に、「OSC Data」画面を表示させて CH1 と CH2 それぞれの信号の max(最大値)・min(最小値)・prd(周期) の**値を記録する。**(用語については第 2 部以降で解説します)
⑤同様に、「AMP/FREQ」画面を表示させて CH1 と CH2 それぞれの信号の amp(振幅)・Vpp(ピークトゥピーク値)・freq(周波数) の値を記録する。(用語については第 2 部以降で解説します)
⑥オシロスコープ画面に 2 種類の波形が表示されていることを確認する。

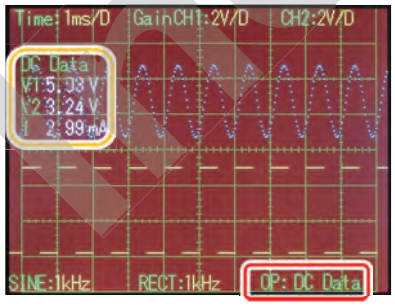
ワークシートに記入しよう！

ヒント 測定機能を表示させよう！

図 1-2(e) のように、**画面右下の表示が「OP：Menu」になっているとき**、オレンジ色の RANGE ダイヤルを右に 1 クリック回すと「DC Data」画面が表示されます。これは電圧計 (V1・V2) と電流計 (I-I) の測定値を表示するモードです。

「メニュー画面で、オレンジのダイヤルを右まわりに 1 クリック回すと…」



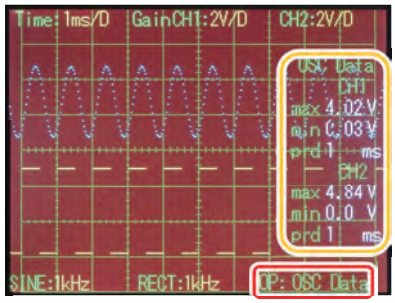


「DC Data が表示される！」

図 1-2(e) 電圧計・電流計の表示

また、「DC Data」画面の状態でオレンジ色の RANGE ダイヤルをさらに右に 1 クリック回すと「OSC Data」が表示され、さらに 1 クリック回すと「AMP/FREQ」が表示されます。これはオシロスコープ (CH1・CH2) に入力された信号の情報を表示するモードです。

「DC Data 画面で、オレンジのダイヤルを右まわりに 1 クリック回すと…」



「OSC Data が表示される！」

「さらにに 1 クリック回すと…」

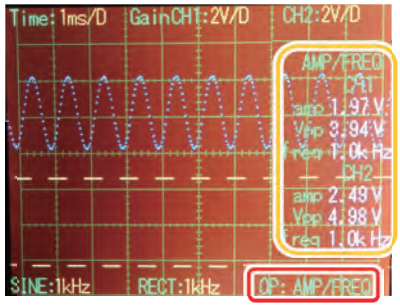


図 1-2(f) オシロスコープ機能

「AMP/FREQ が表示される！」

考えてみよう

①測定した電圧・電流について、電圧 V1 は約 5.0V、電圧 V2 は約 3.3V、電流 I は約 3.3mA になりましたか？
②測定した信号について、CH1 の max(最大値) は約 4V、min(最小値) は約 0V、prd(周期) は約 1ms になりましたか？
③測定した信号について、CH2 の max(最大値) は約 5V、min(最小値) は約 0V、prd(周期) は約 1ms になりましたか？

第 2 部
電子回路基礎

実験 2-1 電圧を測定しよう …… 22
解説 2-2 電圧とは？ …… 24
解説 2-3 電気をあらわす単位と記号 …… 25
実験 2-4 電流を測定しよう …… 26
解説 2-5 オームの法則 …… 28
解説 2-6 電気の源 …… 29
解説 2-7 直流・交流・信号 …… 30
実験 2-8 分圧回路 …… 31
実験 2-9 分流回路 …… 33
解説 2-10 この抵抗、何オーム？ …… 35
解説 2-11 合成抵抗を計算してみよう …… 36
実験 2-12 コンデンサの充電と放電 …… 37
解説 2-13 コンデンサのはたらき …… 39
実験 2-14 CR 回路と過渡現象 …… 40
解説 2-15 信号と周波数 …… 43
解説 2-16 電磁力 …… 44
実験 2-17 コンデンサとコイルの周波数特性 …… 46
実験 2-18 コンデンサとコイルの共振回路 …… 48

©MGC Inc.

2-1 電圧を測定しよう

実験

学習ポイント

光や音、熱は自然界のエネルギーの一種です。照明や暖房器具といった機器は電気のエネルギーを光や熱に変換しており、電子機器に電気のエネルギーを供給するみなもと（源）を**電源**と呼びます。

チェック

電気エネルギーを表現する単位は色々なものがありますが、その基本となる**電圧**（Voltage）について着目しましょう。

「**ある地点から別の地点へと電気を流そうとする力の強さ**」のことを**電圧**と呼びます。電池や電源にはプラス極からマイナス極へと電気を流そうとするはたらきがあるのです。

電圧の単位は「V」という記号であらわされ、この記号を**ボルト**と読みます。

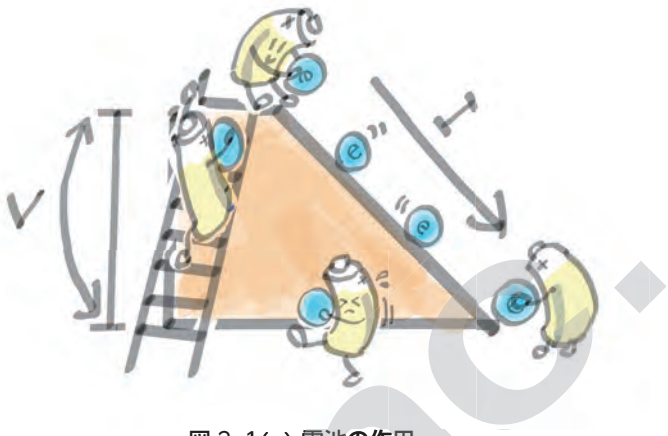


図 2-1(a) 電池の作用

回路を作ろう！ 1

E-Station の電源端子の電圧を電圧計 V1 で測定します。

①図 2-1(c) の接続図を参考に、図 2-1(b) の回路を組みましょう。電圧計 V1 端子と電源 3.3V 端子をワイヤーで接続します。

②回路図で青く色づけしてある **3.3V 端子のマイナス側は E-Station の内部でグランド（GND）部分につながっています**ので、配線の必要はありません。

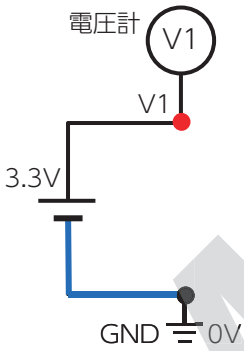


図 2-1(b) 回路図

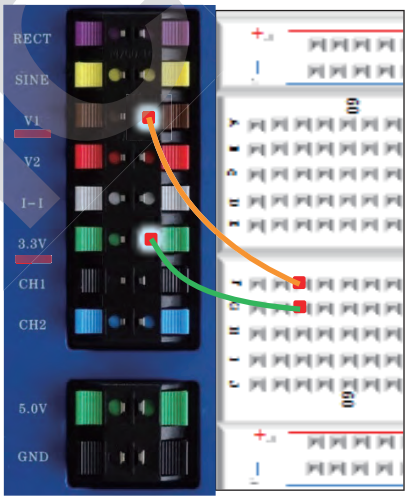


図 2-1(c) 接続図

実験のすすめ方

①E-Station の電圧計 V1 を使って、電源 3.3V 端子の電圧 V1 を測定し数値を記録する。

ワークシートに記入しよう！

回路を作ろう！ 2

①次に、乾電池のプラス（+）端子の電圧を電圧計 V2 で測定します。

②E-Station には乾電池をセットするための電池ボックスが付属しています。このボックスに入れる**単 3 形乾電池を別途用意**してください。

③図 2-1(e) の接続図を参考に、図 2-1(d) の回路を組みましょう。

④前の実験で作る電圧計 V1 と 3.3V 端子は接続したままにしておきます。

©MGC Inc.

回路を作ろう！2

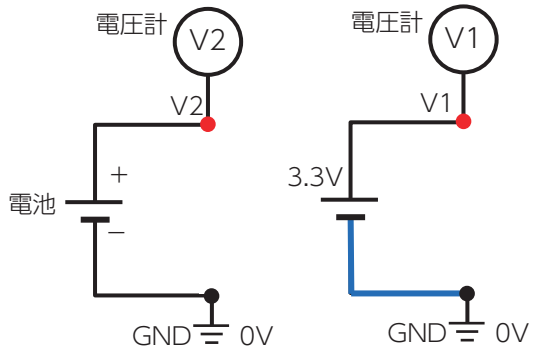


図 2-1(d) 回路図

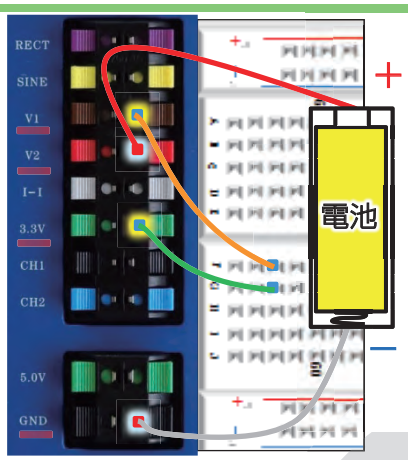


図 2-1(e) 接続図

ヒント E-Station の電源端子と回路図記号

図 2-1(f) のように、電源の回路図記号の上側(線の長い方)が電源端子のプラス側です。今後、回路図を見てパーツを接続する際の参考にしてください。また、電源端子の回路図記号は乾電池と同じ記号であらわし、記号の横に「3.3V」もしくは「5V」を記載します。

図 2-1(f) 電源端子と回路図記号

実験のすすめ方

①E-Station の電圧計 V2 を使って、電池のプラス端子の電圧 V2 を測定し数値を記録する。

ワークシートに記入しよう！

回路を作ろう！3

①二つの電源を同じ直線上につないだときの電圧を測定します。
②図 2-1(h) の接続図を参考に、図 2-1(g) の回路を組みましょう。電池のマイナス端子側をつなぎかえます。

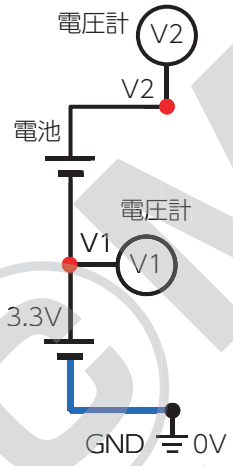


図 2-1(g) 回路図

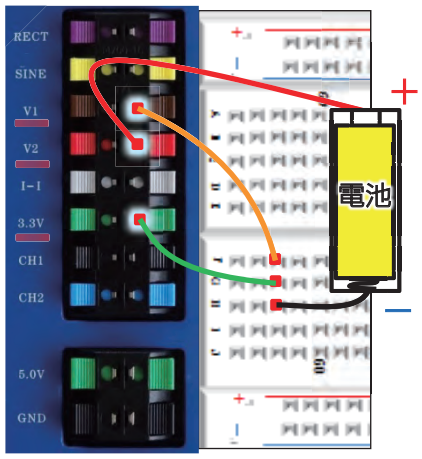


図 2-1(h) 接続図

実験のすすめ方

①E-Station の電圧計 V2 を使って、電池のプラス端子の電圧 V2 を測定し数値を記録する。

ワークシートに記入しよう！

考えてみよう

①測定した結果をもとに、ワークシートの空欄にあてはまる数値を記入しよう。
②回路図 2-1(d) と回路図 2-1(g) の実験では、「同じ電池のプラス端子の電圧」を電圧計 V2 で測定していますが、電圧の値が大きく違うのはなぜでしょうか？

ワークシートに記入しよう！

©MSC Inc.

2-2 電圧とは？

解説

電圧とは「ある地点から別の地点へと電気を流そうとする力の強さ」のことです。**ある地点と別の地点の電圧の差が大きいほどたくさんの電流が流れる**ことになります。

電圧をイメージするときは**高さ**をイメージするとよいでしょう。電気は高いところから低いところへ向かって流れます。標高の高い山から低い海へと川が流れるのと似ています。

「高い」「低い」と表現できるものには「人の身長」「建物の大きさ」「山の標高」や、あるいは「温度」などがあります。これらは、何かを基準にして「高い」か「低い」かを比較しています。

たとえば身長 160cm の A さんと、身長 140cm の B さんとではどちらがより高いでしょうか？ また、B さんが高さ 10m のハシゴの上に登ったときはどちらがより高いでしょうか？

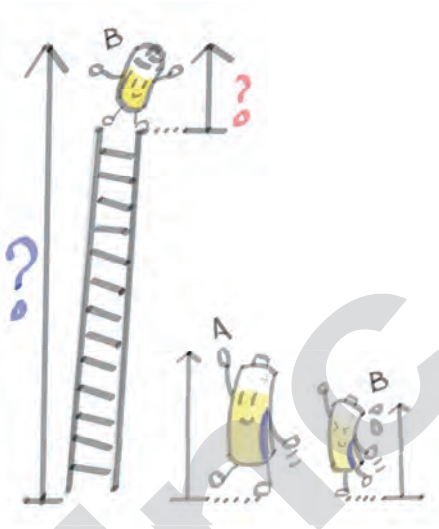


図 2-2(a) 基準と高さ

電圧測定では「約 1.5V」や「約 3.3V」といった数値が出てきました。この電圧値が「**ある地点の高さ**」に相当します。そしてその基準は…**電子回路の「地面」**です。電子回路には「地面」に相当する基準があり、これを**グランド（GND）**と呼んでいます。そしてグランド自体の高さもまた、場所によって変わります。つまり、今お使いの E-Station のグランドや、他の人が使っている E-Station のグランドなど、**別々の機器のグランド（GND）はつなぎ合わせないかぎりまったく別の高さ**なのです。電圧を扱うとき「どこを基準にした高さ」なのか、「どの部分の高さ」なのかを意識しましょう。

電池のはたらきにより、プラス端子の電圧はマイナス端子から見て約1.5V高くなる！

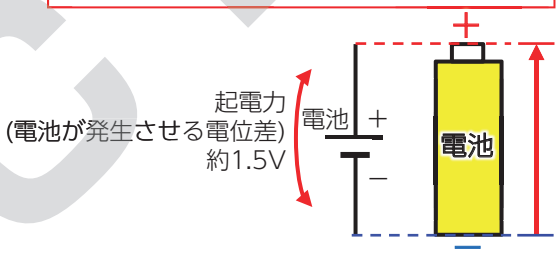


図 2-2(b) 乾電池の起電力

実験 [2.1] の電圧測定では、一般的なアルカリ単 3 形乾電池であれば約 1.5V の値が測定できます。しかし、3.3V 電源をつなぐと乾電池のプラス端子の電圧の値が大きく変化します。

乾電池があたえようとする電圧は、マイナス端子からプラス端子までの電圧の差です。このような 2 点間の電圧の差を**電位差**と呼び、電源のはたらきにより発生する電位差を**起電力**と呼びます。図 2-2(b) のように、一般的なアルカリ単 3 形乾電池の起電力は約 1.5V であるといえます。E-Station の電源端子も 3.3V や 5V の起電力を持っています。

プラス端子の電圧は、
GND端子(0V)から見ると約1.5V高い！
つまり、V1の電圧値は約1.5V！

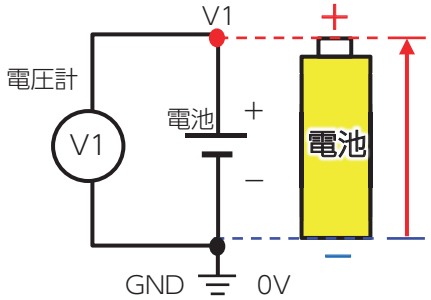
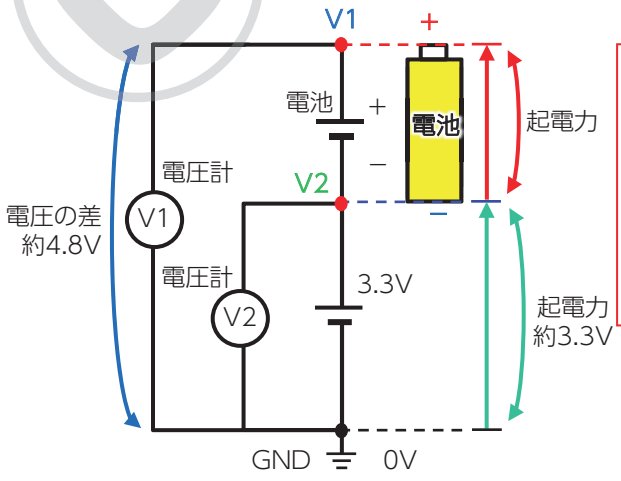


図 2-2(c) 乾電池の電圧測定

E-Station の電圧計は、基準の電圧（GND）「0V」から見たときの電圧計端子の電圧の高さを数値で表示する機能です。この電圧計は E-Station の本体内部で電圧の基準（GND）とつながっています。実験 [2.1] の回路図では省略して表現していますが、電圧計には本来 2 本の足があり、図 2-2(c) のように測定したい 2 点につないで使います。

さて、これに 3.3V の電源が一直線につながると図 2-2(d) のようになります。電池のプラス端子の電圧が約 4.8V になったのは、電圧計が GND からプラス端子までの電圧の差を見ていたからです。このように、電圧測定とは「どこを基準にするか」で見え方が変わってしまうものなのです。



電池のプラス端子の電圧は、
GND端子(0V)から見ると約4.8V高い！
つまり、V1の電圧値は約4.8V！
同様に、V2の電圧値は約3.3V！

電池のマイナス端子からプラス端子までの電位差(起電力)が知りたければ、
(V1-V2)を計算すればよい！
(約4.8V − 約3.3V) = 約1.5V となる。

図 2-2(d) 乾電池と 3.3V 電源の電圧測定

©MSC Inc.

2-5 オームの法則

解 説

実験 [2.4] では、抵抗を使った回路の電圧と電流を測定しました。電圧と電流と抵抗のあいだにはある法則があり、計算して求めることができます。図 2-5(a) に示すこの法則を**オームの法則**といいます。
オームの法則は、次の (1) ～ (3) の 3 通りの使い方ができるとても便利な関係式です。

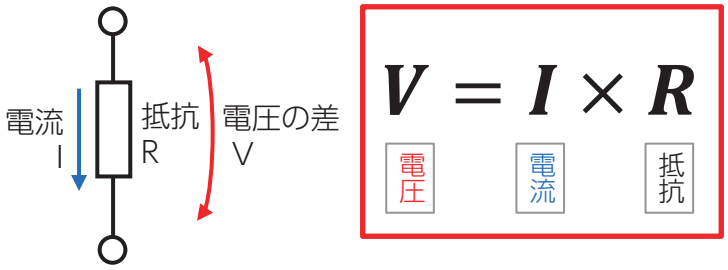


図 2-5(a) オームの法則

(1) 抵抗にかかる電圧の差は、電流と抵抗をかけると計算できる。

$$V = I \times R \quad \cdots (1)$$

(2) 抵抗を流れる電流の量は、電圧を抵抗で割ると計算できる。

$$I = \frac{V}{R} \quad \cdots\cdots\cdots\cdots (2)$$

(3) 抵抗の値は、電圧を電流で割ると計算できる。

$$R = \frac{V}{I} \quad \cdots\cdots\cdots\cdots (3)$$

図 2-5(b) V と I と R の関係

抵抗に電流が流れているとき、「①抵抗の両端にかかる電圧の差」「②抵抗を流れる電流の量」「③抵抗の値」のうちどれか 2 つの値がわかっていれば、残りのひとつを計算することができるのです。さっそく使って練習してみましょう。
(2) の関係式から見えてくる重要な法則のひとつは、「**抵抗が大きいほど流れる電流が小さくなる**」ということです。電流を流すことに「抵抗」するから抵抗、というワケですね。つまり、抵抗は電流を流れにくくするためのパーツとも言えます。

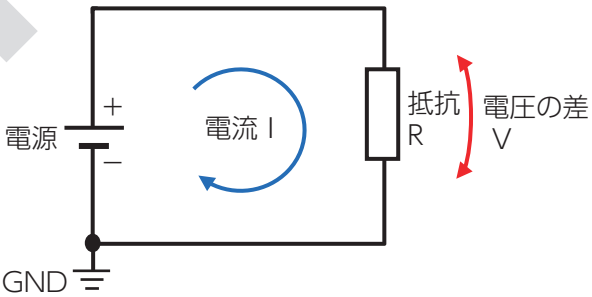


図 2-5(c) 電圧と電流と抵抗の関係

練 習

図 2-5(c) の回路を参考にして、次の問題をオームの法則を使って計算しよう。
①R = 100Ω の抵抗に電流 I = 0.2A が流れているとき、両端にかかる電圧の差 V を求めよう。
②R = 100Ω の抵抗の両端に電圧の差 V = 10V がかかっているとき、流れる電流 I を求めよう。
③電流 I = 0.01A が流れると電圧の差 V= 10V が発生するような抵抗 R の値を求めよう。
④R = 500Ω の抵抗に電流 I = 0.02A が流れているとき、両端にかかる電圧の差 V を求めよう。
⑤R = 10kΩ の抵抗の両端に電圧の差 V= 20V がかかってているとき、流れる電流 I を求めよう。

ワークシートに記入しよう！

©MIC Inc.

2-10 この抵抗、何オーム？

解説

これからの実験でたくさんの電子パーツに触れることになりますが、その中でも固定抵抗器（抵抗）はよく使うパーツです。

抵抗には様々な値や、サイズがあります。ブレッドボードにさし込んで使うタイプの抵抗には色のついた帯（カラーコード）があり、その色を見ればその抵抗が何オームなのかひと目でわかるようになっています。

まずは抵抗をひとつ手に取って、その色を見てみましょう。金色や銀色の部分を右側にして見ると、読みやすくなります。

カラーコードは、同じ色でも何本目にあるのかで意味するものが違います。ここでは、実験でも用いる一般的なカーボン抵抗を例に見ていきましょう。

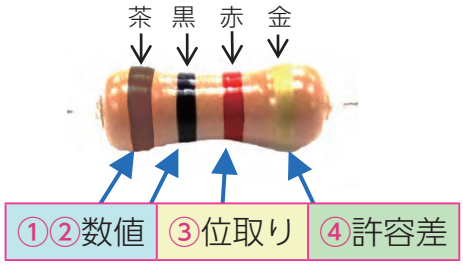


図 2-10(a) 抵抗のカラーコード

図 2-10(a) の抵抗の色は「茶→黒→赤→金」の順です。いったい何オームの抵抗でしょうか？

①②最初の 2 本は、数値をあらわします。「茶」は「1」、「黒」は「0」、つまりこの組み合わせで「10」という意味になります。

③ 3 本目は位取りをあらわします。「赤」は「2」で、数値のケタを「2 ケタ」増やします。数値の「10」と位取りの「00」で「1000Ω」という意味になります。

④ 4 本目は許容差です。工場で作られる抵抗はひとつひとつ値が微妙に違います。金色は「± 5%」で、「1000Ω」の「± 5%」つまり「1050Ω ～ 950Ω」までばらつきがあるという意味になります。

色	(1 本目) 数値	(2 本目) 数値	(3 本目) 位取り	(4 本目) 許容差
黒	0	0	10^0	—
茶	1	1	10^1	± 1%
赤	2	2	10^2	± 2%
橙	3	3	10^3	—
黄	4	4	10^4	—
緑	5	5	10^5	—
青	6	6	10^6	—
紫	7	7	10^7	—
灰	8	8	10^8	—
白	9	9	10^9	—
金	—	—	10^{-1}	± 5%
銀	—	—	10^{-2}	± 10%

図 2-10(b) 色が意味するもの

図 2-10(c) 抵抗値を読もう

図 2-10(b) は各色が意味する値の一覧です。位取りは「10 の何乗か」で表現します。「10^1 なら数値のあとにゼロが 1 個付く」というように考えましょう。実験で使う抵抗を探すとき、カラーコードを見て何オームか調べることは良い練習になります。たくさん実験して、自然に覚えていきましょう。

©MSC inc.

2-12 コンデンサの充電と放電

実験

学習ポイント

コンデンサは身の回りの電化製品のほとんどに使われている、電子回路では抵抗と並び欠かせない電子パーツです。実験を通して、その動作とその役割について理解を深めましょう。

チェック

コンデンサはキャパシタとも呼ばれ、**電気をたくわえたり放出したりする**ことができる素子です。コンデンサがたくわえられる電気の量を**静電容量**と呼び、単位は「F」という記号であらわされ、この記号を**ファラド**と読みます。

(※)電解コンデンサにはプラス(+)とマイナス(−)の向き(極性)があります。使うときは回路記号の＋の側にコンデンサの＋電極をつないでください。向きを間違えて使用すると、電解コンデンサの破損・爆発などの恐れがあります。

図 2-12(a)
様々なコンデンサ

回路を作ろう！

①図 2-12(b) の回路図を見て、必要なパーツを準備します。解説 [2.10] を参考に、必要な抵抗を探しましょう。
②コンデンサ C の回路図記号には「+」のマークが付いています。ここでは、図 2-12(a) にある電解コンデンサを選びます。
③電解コンデンサの容量は、本体の黒い部分に「100μF」と記載されています。
④**タクトスイッチはどの色でも問題ありません。向きに注意して、中央の溝をまたぐように取り付けます。**
⑤図 2-12(c) の接続図を参考に、図 2-12(b) の回路を組みましょう。

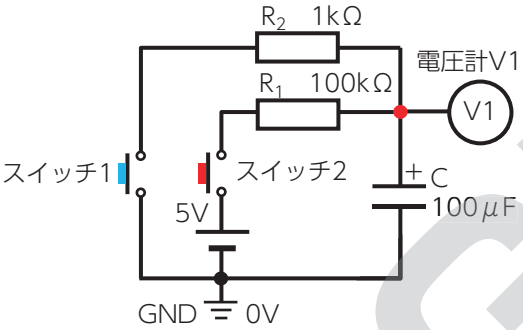


図 2-12(b) 回路図

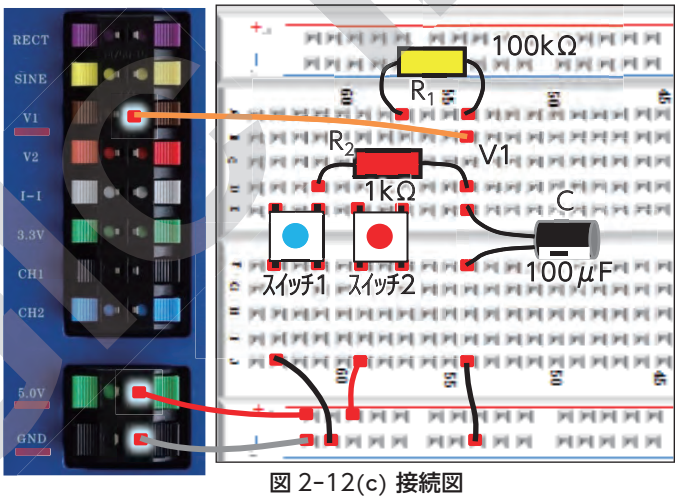


図 2-12(c) 接続図

ヒント

タクトスイッチの構造

タクトスイッチはボタンを押したり離したりして使います。
図 2-12(d) の構造図のように、左側の端子と右側の端子はボタンを押したときにつながり、ボタンを離すとつながらなくなります。

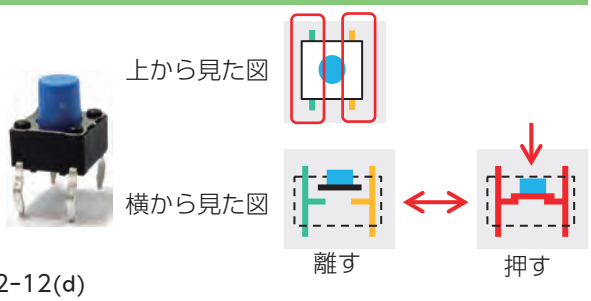


図 2-12(d)
タクトスイッチ

実験のすすめ方

①スイッチ 1 を押し続け、数秒経過したあとの電圧 V1 を測定し数値を記録する。
②スイッチ 1 を離し、スイッチ 2 を押し続ける。電圧 V1 の数値が時間とともにどのように変化するか観察する。
③スイッチ 2 を押し続け、1 分経過したあとの電圧 V1 を測定し数値を記録する。
④スイッチ 2 を離し、スイッチ 1 を押し続ける。電圧 V1 の数値が時間とともにどのように変化するか観察する
⑤スイッチ 1 を押し続ける。時計を用意し、スイッチ 2 を押しはじめてから電圧 V1 が約 3.16V を超えるまでの時間を計測する。
⑥両方のスイッチを同時に押し続け、数秒経過したあとの電圧 V1 を測定し数値を記録する。

ワークシートに記入しよう！

©MCR Inc.

ヒント 回路図の配置と表現

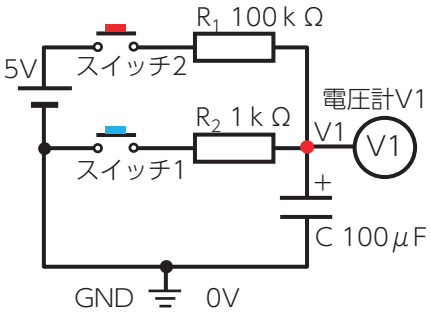


図 2-12(e) 回路図の別の描き方

図 2-12(e) の回路図は、一見すると前ページの図 2-12(b) の回路図と違うように見えますが、**実は同じ回路図です。**図 2-12(b) は接続図での実際の部品配置に合わせて描いてありますが、図 2-12(e) のように描くこともできます。

一般的に、電源や信号の**入力は左側、出力は右側に描き、高い電圧のところを上側に描かれる**ことがしばしばあります。このページでは図 2-12(e) の回路図で動作を解説します。

この回路には大きく分けて下記の 4 種類の状態が存在します。それぞれの状態を図 2-12(f) に示します。

A. スイッチ 1 が押されているとき
B. スイッチ 2 が押されているとき
C. どちらも押されていないとき
D. 両方が押されているとき

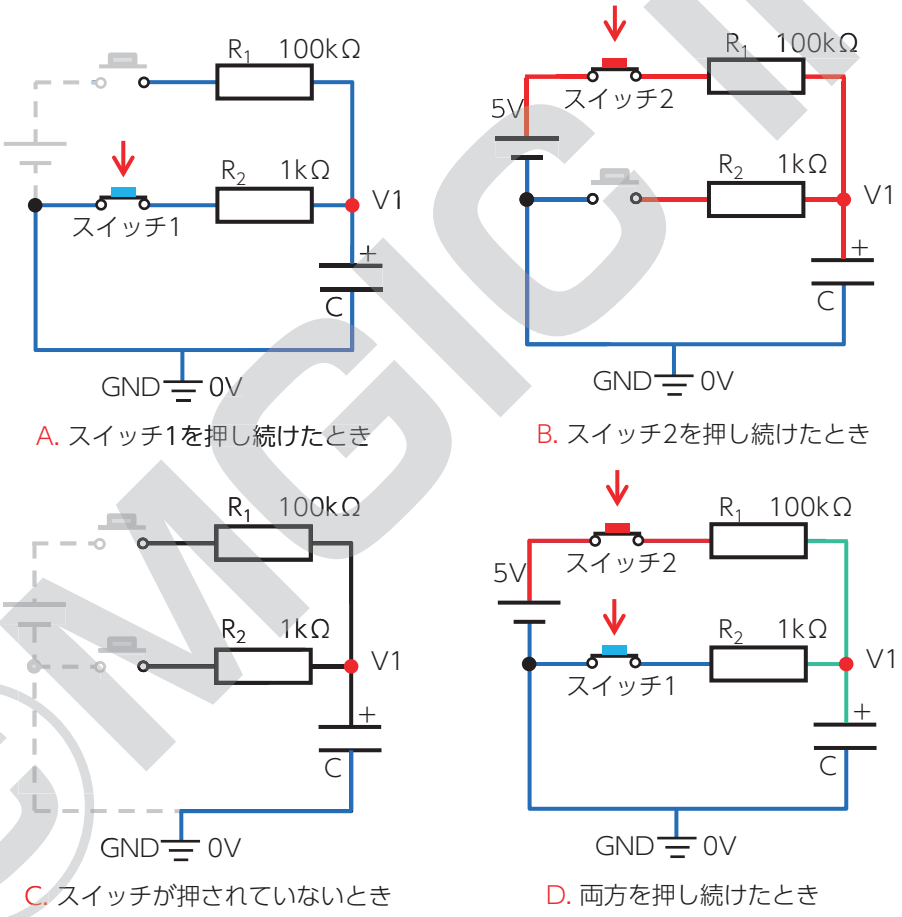


図 2-12(f) スイッチによる回路の状態の変化

考えてみよう

①スイッチ 1 を押し続けたあとの電圧 V1 は、どの電圧に等しくなるといえますか？
②スイッチ 2 を押し続けたあとの電圧 V1 は、どの電圧に等しくなるといえますか？
③両方のスイッチを離したとき、電圧 V1 は変化しますか？
④両方のスイッチを押し続けたときの回路図から電解コンデンサを取りのぞいたとき、電圧 V1 は合成抵抗と分圧回路の考え方で計算することができます。R_1 と R_2 の抵抗値を使って、オームの法則で電圧 V1 を求めよう。

ワークシートに記入しよう！

©MSIC Inc.

2-14 CR 回路と過渡現象

実験

学習ポイント

実験 [2.12] では手でスイッチをオン・オフして充放電を行い、その様子を観察しました。コンデンサの電圧を変化させるには時間がかかります。次は 0V と 5V が自動で切り替わる信号を使って、コンデンサの充放電のようすを**オシロスコープ**機能で見てみましょう。

チェック

E-Station には信号を発生させる端子があり、そのひとつに「RECT」（矩形波）があります。0V と 5V の電圧を繰り返し切り替わる信号で、このかたちの波形を**矩形波**（Rectangle wave）と呼びます。1 秒間に信号が繰り返される数を**周波数**と呼び、繰り返す信号の一回分の長さ（時間）を**周期**と呼びます。例えば 1 周期が 0.01 秒の信号であれば、$\frac{1}{0.01}$ = 100Hz となる計算になり、1 秒間に 10 回繰り返す信号（周波数 10Hz）の周期は $\frac{1}{10}$ = 0.1 秒となる計算です。

信号の中心から変化する電圧幅を**振幅**と呼び、E-Station の RECT 出力は 0V ～ 5V まで変化するので、その振幅は 2.5V です。また、信号の最大値から最小値までの幅を**ピークトゥピーク値 (Vpp/Vp-p)** と表現します。

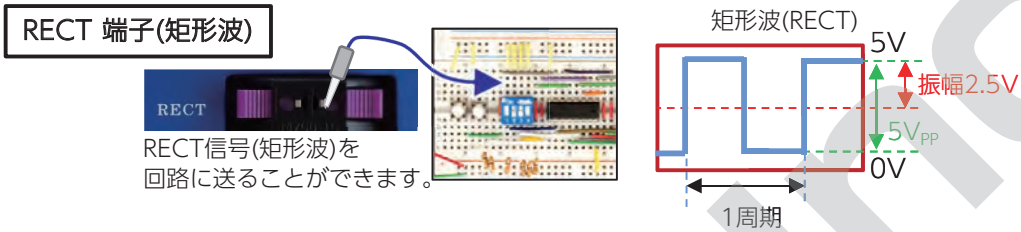


図 2-14(a) RECT 端子と矩形波

回路を作ろう！

①図 2-14(b) の回路図を見て、必要なパーツを準備します。解説 [2.13] を参考に、必要なコンデンサを探しましょう。
②図 2-14(c) の接続図を参考に、図 2-14(b) の回路を組みましょう。
③**RECT 端子の GND 側は E-Station の内部で最初からつながっています。回路は抵抗の側だけをつないで作ってください。**
④回路図上で「CH1」、「CH2」と表記してある点は、それぞれ E-Station の「CH1 端子」と「CH2 端子」につなぎます。

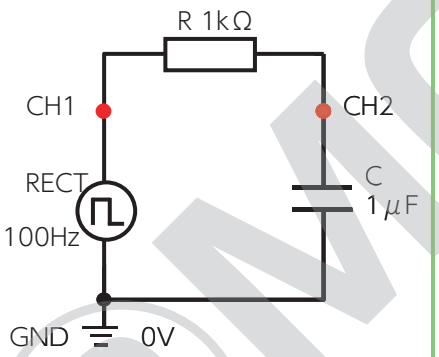


図 2-14(b) 回路図

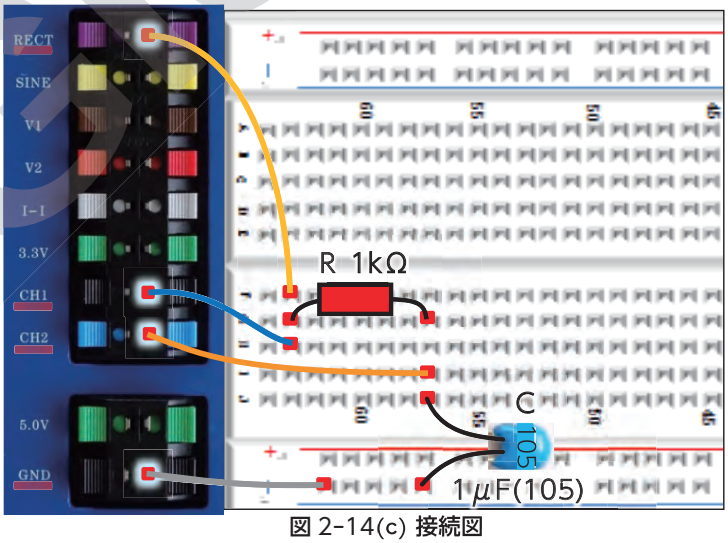


図 2-14(c) 接続図

実験のすすめ方

①E-Station の「OP：Time」の機能を使い、設定を「2ms/D」にする。
②E-Station の「OP：RECT C」の機能を使い、設定を「100Hz」にする。
③CH1（青色）の RECT 信号と、CH2（黄色）のコンデンサの電圧変化の様子を観察する。
④CH1（青色）と CH2（黄色）の信号のかたちを描いて記録する。

ワークシートに記入しよう！

©MGIC Inc.

考えてみよう

この実験では、矩形波(RECT 信号)を回路にあたえ、コンデンサの電圧の変化をとりだして観察しました。回路にあたえる信号を入力、回路からとりだす信号が出力です。入力に対する出力の応答・反応を**過渡応答**と呼びます。

① 入力信号と出力信号の波形を見くらべて、どのような違いがありますか？
② コンデンサの充電時間は、抵抗の値やコンデンサの容量で変わります。コンデンサの容量が小さくなると充電時間はどのように変化すると考えられますか？
③ 抵抗の値が大きくなると充電時間はどのように変化すると考えられますか？

ワークシートに記入しよう！

発展

① 実験回路のコンデンサ C を 0.1μF に変更して、充電時間の変化を確認しよう。
② 実験回路のコンデンサ C を 1μF、抵抗 R を 100Ω に変更して、充電時間の変化を確認しよう。
③ 実験回路のコンデンサ C を 1μF、抵抗 R を 10kΩ に変更して、充電時間の変化を確認しよう。

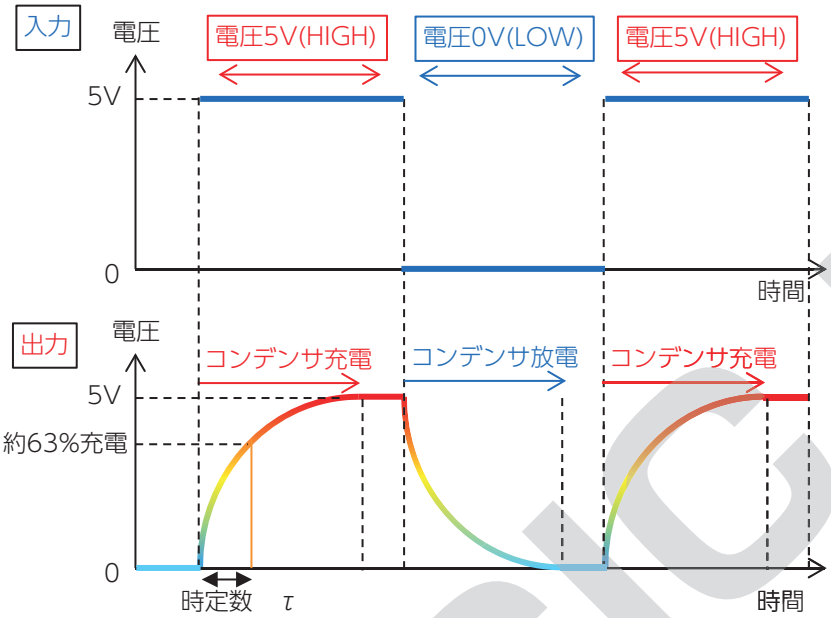


図 2-14(d) **矩形波電圧とコンデンサ電圧の変化**

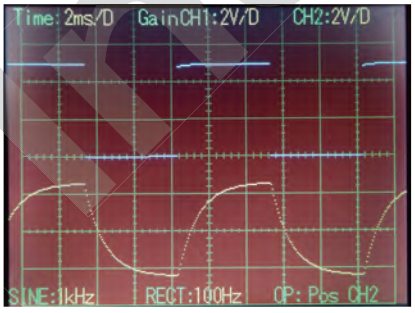
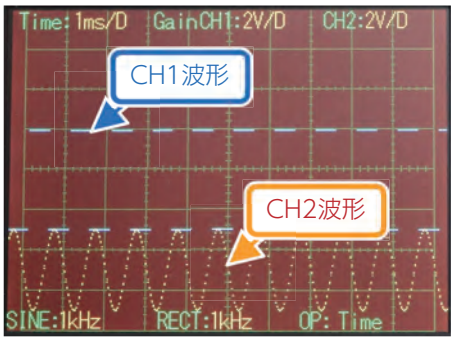


図 2-14(e) **オシロスコープでの観測**

ヒント　オシロスコープの機能を使おう！

E-Station のオシロスコープは、CH1 や CH2 につないだ先の電圧の変化を表示する機能です。
表示の大きさをタテ・ヨコに拡大したり縮小したり、波形が表示される位置を上下に調整したりすることができます。

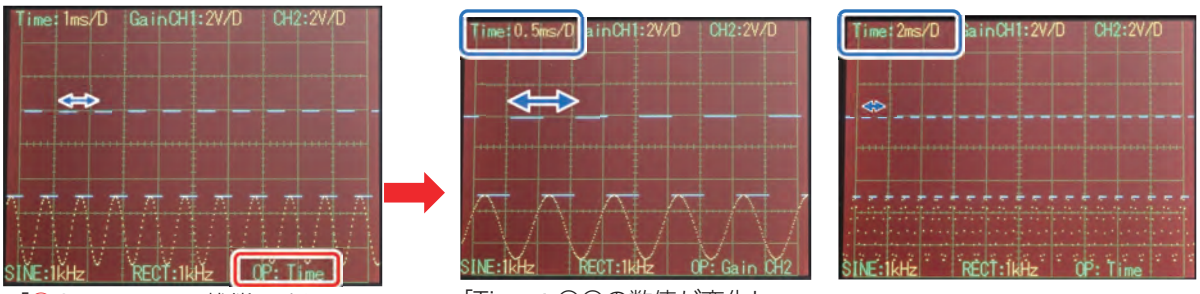
「①メニュー画面で、どちらかのダイヤルを回すと…」



「青が CH1、黄色が CH2 の観測波形！」

図 2-14(f) **オシロスコープの使い方①**

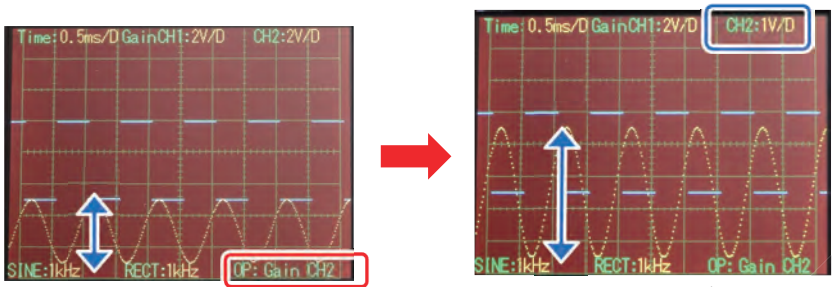
©MGC Inc.



「②OP： Timeの状態でオレンジのダイヤルを回すと…」

「Time：〇〇の数値が変化し、波形表示の時間軸設定(ヨコ方向の縮尺)が変わる！」

「見やすい大きさに調整しよう」



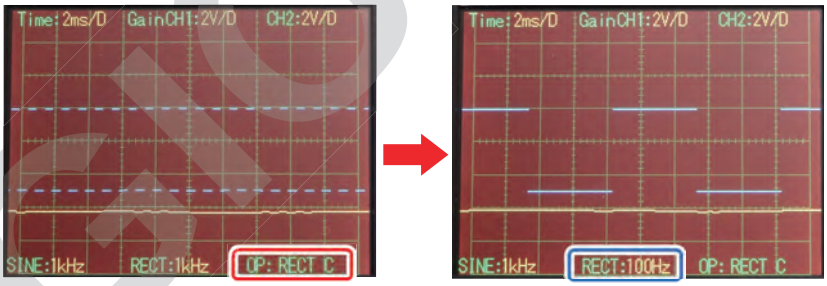
「②OP： Gain CH〇の状態でオレンジのダイヤルを回すと…」

「CH〇：〇〇の数値が変化し、波形表示のゲイン設定(タテ方向の縮尺)が変わる！」

図 2-14(g) オシロスコープの使い方②

ヒント RECT 出力の周波数を変えよう！

オシロスコープの時間軸やゲインの設定は、「信号の見え方」を変えるだけで、信号そのものの速さ(周波数)や大きさ(振幅)は変化しません。E-Station の RECT 出力は周波数を変更することができます。周波数の変更でオシロスコープの表示が見づらくなったときは時間軸などを再調整しましょう。



「①OP： RECT Cの状態でオレンジのダイヤルを回すと…」

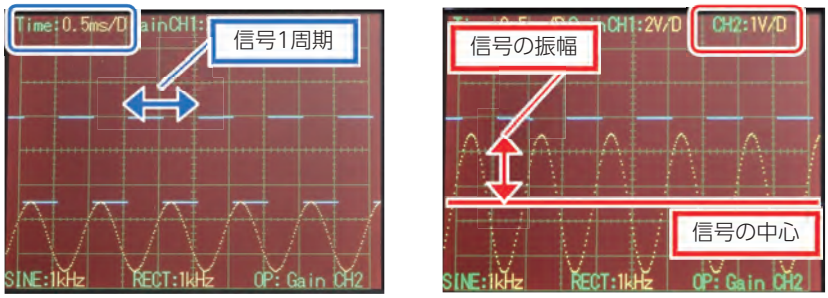
「RECT：〇〇の数値が変化し、出力信号の周波数が変わる！」

図 2-14(h) 出力周波数の設定

補足 「2ms/D」「1V/D」って何？

オシロスコープの時間軸やゲインの設定は、「2ms/D」や「1V/D」のように表示されます。

「ms」は「ミリセック」と読み、「m」は「ミリ」、「s」は秒(second)をあらわします。「2ms」は 2 ミリ秒(0.002 秒)ということです。「D」は「ディビジョン：Division」の頭文字です。ディビジョンとは分割という意味で、画面の 1 マスの間隔が何秒・何ボルトなのかを示します。



「Time：0.5ms/D」…1マスのヨコ幅が0.5ms
信号の 1 周期の長さはおよそ2マス(D)
→ 1周期は0.5×2 = 約1ms！

「CH2：1V/D」…1マスのタテ幅が1V
信号の振幅はおよそ2マス(D)
→ 振幅は1×2 = 約2V！

図 2-14(i) ディビジョンの見方

OMRON Inc.

第 3 部
半導体回路の基本

実験 3-1 ダイオードの順方向特性 …… 52
実験 3-2 ダイオードの逆方向特性 …… 54
実験 3-3 発光ダイオード(LED)を調べる …… 56
実験 3-4 バイポーラ・トランジスタの B-E 特性 …… 58
実験 3-5 BJT 増幅回路 …… 60
解説 3-6 半導体のしくみ …… 62
実験 3-7 エミッタ接地増幅回路 …… 68
実験 3-8 スピーカーを鳴らす …… 70
コラム データシートを見てみよう …… 72
実験 3-9 スピーカー音を大きくする …… 73
解説 3-10 トランジスタ発明の意義 …… 75
解説 3-11 BJT と MOSFET …… 76

©MSIC Inc.

3-1 ダイオードの順方向特性

実験

学習ポイント

電気を通す**導体**と、電気をほとんど通さない**絶縁体**の中間の性質をもつ物質を**半導体**(Semiconductor)と呼び、熱や光などの条件によって**電気の通しやすさが変化**します。半導体の性質をもつ物質を組み合わせて作られる製品や素子そのものを「半導体」と呼ばれることも多くあります。

チェック

一方向のみへ電流を流す作用を**整流作用**と呼び、整流作用を持つ素子が**ダイオード**(Diode)です。電流が流れる方向を**順方向**、電流をほとんど流さない方向を**逆方向**と呼びます。

半導体ダイオード(Semiconductor Diode)は半導体製品の元祖とも言える素子です。半導体を扱うときは**向きに気を付けて**回路を作りましょう。

一般的なダイオードの順方向は、2つの端子でその向きを判断することができます。ダイオードには「アノード(A)」と「カソード(K)」の2つの端子があり、順方向はアノードからカソード、つまり「A→K」の向きに電流が流れます。そしてその逆が逆方向です。アノードは「陽極」、カソードは「陰極」を意味し、真空管の時代から使われて いる名称から由来しています

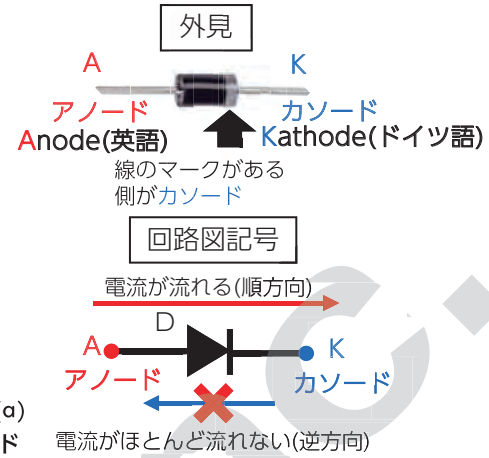


図 3-1(a)
ダイオード

回路を作ろう！

①ダイオードにはつなぐ向きがあるため、図 3-1(a) を参考に注意してつなぎましょう。
②**第 3 部以降の回路図では、電圧計の記号を省略して表記します**。「V1」や「V2」の表記がある赤い点「 • 」や丸「○」が、電圧を測定する部分を意味します。
③図 3-1(c) の接続図を参考に、図 3-1(b) の回路を組みましょう。

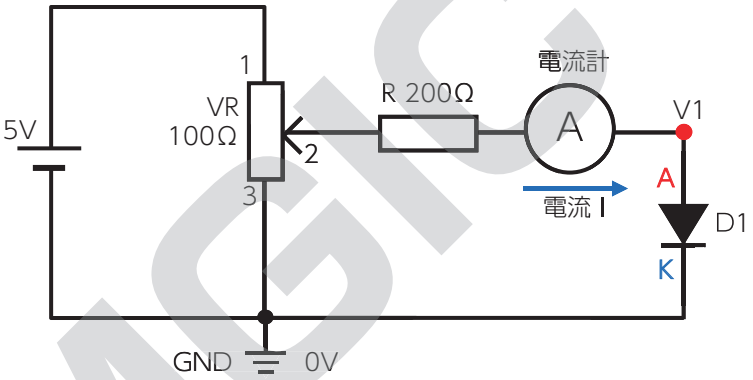


図 3-1(b) 回路図

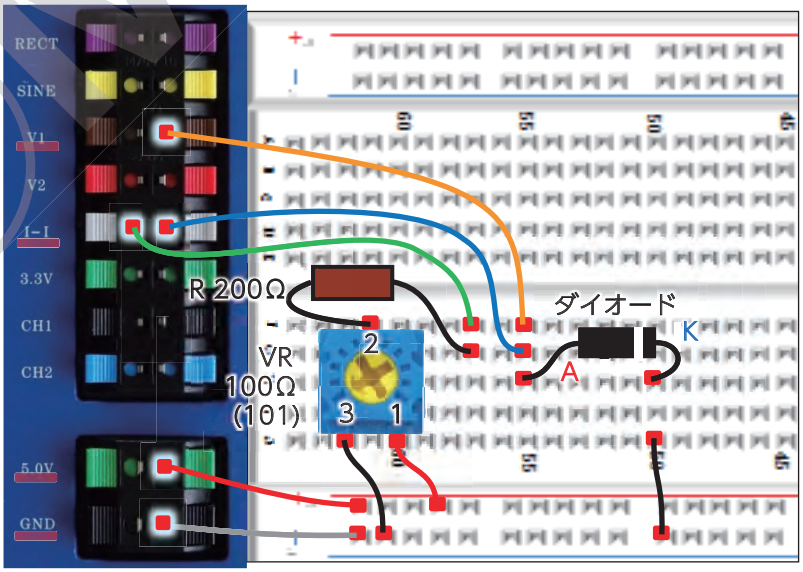


図 3-1(c) 接続図

©MGC inc.

ヒント

電源と回路図のあらわし方

第 2 部までの回路図では、「電源のプラス側から一回りしてマイナス端子へ戻ってくる」ような描き方で回路を表現していました。しかし、複雑な回路になると配線を一周させるとかえって線の交差が複雑になり、非常に見づらい回路図になってしまいます。そこで、「**電源のプラス側からグランド(GND)までの配線**」であらわすような表現も用いられます。これは図 3-1(e)のように、電源のマイナス側とグランド(GND)のつながりを省略したかたちです。もちろん、電源のマイナス側がグランド(GND)とつながっていないときは使えない表現なので注意してください。

回路図の描き方は様々な表現がありますが、配線と部品の接続が同じである必要があります。回路の規模や接続先に応じてよりわかりやすく、その用途に応じた描き方を心がけましょう。

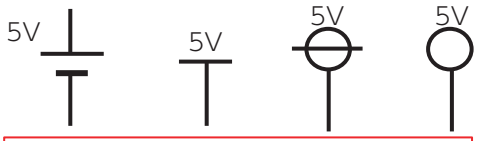


図 3-1(d) 電源記号

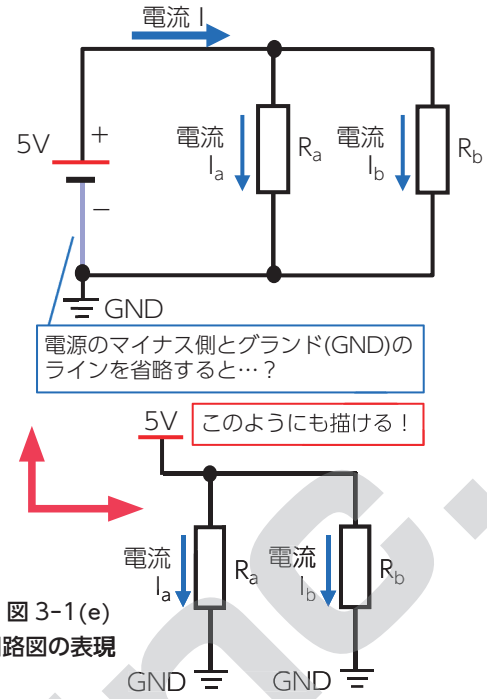


図 3-1(e) 回路図の表現

実験のすすめ方

①VR のダイヤルを左いっぱいに回しているとき、点 V1 の電圧が 0 になることを確認する。
②このときダイオードを流れる電流 I [A] を測定し数値を記録する。
③VR のダイヤルを回して点 V1 の電圧を約 0.1V 上昇させ、ダイオードを流れる電流 I [A] を測定し数値を記録する。
④以降同様に繰り返し 0.1V ずつ上昇させ、ダイオードを流れる電流 I [A] を測定し数値を記録する。
⑤VR を右いっぱいに回しても点 V1 の電圧が上昇しなくなったら、そのときの電圧と電流を記録し、測定を終了する。
⑥記録した電圧と電流の関係から、グラフを描く。

ワークシートに記入しよう！

考えてみよう

① ダイオードの順方向に電圧をかけたとき、どのような電圧でも電流は流れていましたか？
② ダイオードに電流が流れはじめる電圧は、約何ボルトでしたか？
③ ダイオードに電流が流れはじめてから電圧を上昇させたとき、電流はゆるやかに増えましたか？ 急に増えましたか？

ワークシートに記入しよう！

補足

ダイオードの順方向電圧

一定以上の電圧をダイオードにかけると、急に電流が流れはじめます。実験では約 0.5V ～約 0.7V のあたりがさかい目になります。

ダイオードに電流が流れるときに発生する端子間の電圧の差をダイオードの**順方向電圧 V_F** と呼び、順方向に流れる電流を**順方向電流 I_F** と呼びます。V_F や I_F の「F」は「Forward：前方に、先へ」を意味します。**順方向電圧の上昇量と順方向電流の増加量は比例しません。**抵抗と違い、電流が流れ始めたダイオードにかける電圧を上げると、ほんの少しの電圧上昇に対して電流がもっと急激に増えるのです。また、ダイオードにかかる電圧が順方向電圧より高いときに ON になり、順方向電圧より低いときに OFF になるスイッチのようにみなすこともできます。

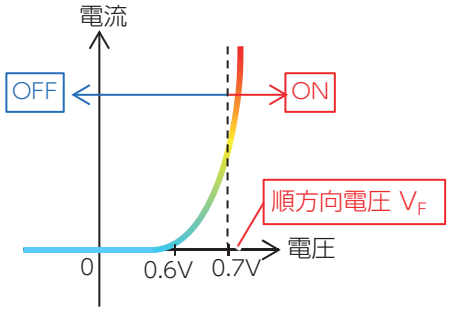


図 3-1(f) ダイオードの電圧電流特性

補足

電圧をかける？

電子回路の世界では、「○○に△ボルトの電圧をかける」という表現がよく用いられます。本来、電圧とは電気を流そうとする力のことですが、この場合は「○○の 2 つの端子の電圧の差を△ボルトにする」というような意味合いがあります。また、ダイオードの順方向に電圧をかけるということは、「アノード（A）の電圧をカソード(K)よりも高くする」という意味にもなります。もし逆方向だと…？

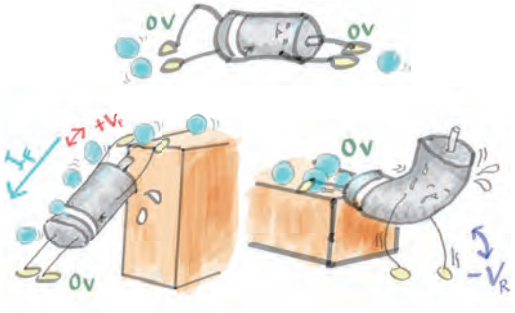


図 3-1(g) 電圧をかける

3-6 半導体のしくみ

解説

ダイオードや BJT のベース－エミッタ間は一定以上の高さの順方向電圧をかけないと電流が流れず、逆方向電圧をかけてもほとんど電流が流れませんでした。なぜこのような半導体作用が起きるのでしょうか。

接合ダイオードをはじめとする半導体素子は、**シリコン**という素材の結晶で作られています。シリコン結晶は**ケイ素原子**と呼ばれる元素（物質を構成するもとになる粒子）がたくさん集まった結晶でできていて、ケイ素原子はその外側に 4 個の電子を持っています。

図 3-6(a) **ケイ素原子**

ケイ素（Si）原子は周囲に電子を「4 個」しか持っていません。しかし隣にケイ素（Si）原子がいると、お互いに電子を 1 個ずつさし出して結びつき、シェアすることができます。これを**共有結合**と呼びます。

共有結合を利用すれば、四方のケイ素（Si）原子と手を結ぶことでまるで自分のまわりに「8 個」の電子があるような状態になり、そのつながりのおかげでケイ素（Si）原子はそれぞれが安定することができます。共有結合結晶はこのようにしてたくさんの原子同士が手を取り合い、強く結びついてできています。鉛筆の芯に使われる黒鉛や、ダイヤモンドなどは同じ炭素原子の共有結合でできています。結びつき方で性質が変わるのです。

電子はマイナスの性質を持つとても小さな粒子です。また、ケイ素原子は「Si」という元素記号であらわされます。

ケイ素（Si）に限らず、原子というものは電子をいくつか持っています。そして、ほとんどの原子は**一番外側の電子の数が「8 個」になるとひとまず安定する**性質があります。そのためチャンスさえあれば原子は外側の電子を「8 個」にしようとします。

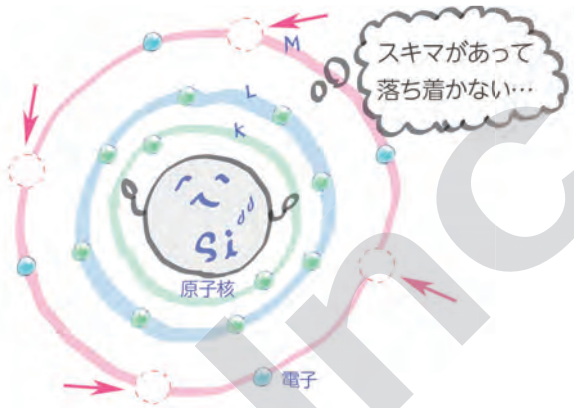


図 3-6(b) **外側の電子**

図 3-6(c) **電子をさし出しあうと…？**

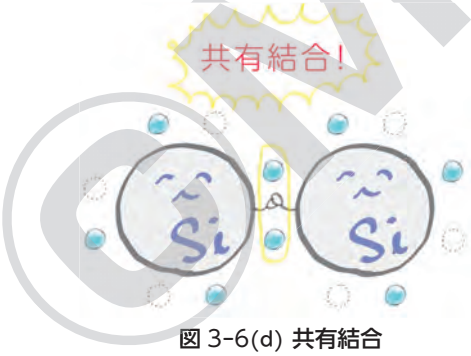


図 3-6(d) **共有結合**

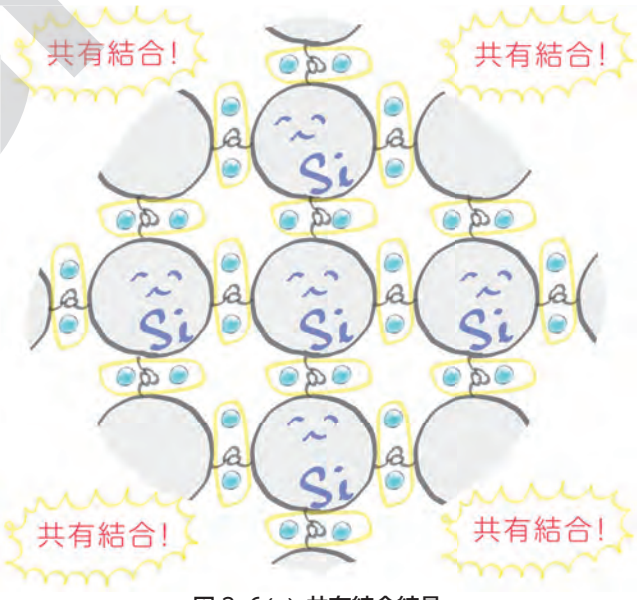


図 3-6(e) **共有結合結晶**

ここで、重要なことをおさらいしましょう。「**電流の正体は、電子の移動（流れ）である**」ということです！

ケイ素（Si）原子の共有結合結晶ではみな「8 個」の電子をもち、とても満足しています。そんな中では、わざわざ電子を手放して移動させようとするケイ素（Si）原子はほとんどいません。せっかく安定しているので、電子を動かすのをいやがるのです。

このままでは電子がなかなか移動せず、電流がほとんど流れないただの絶縁体です。では、どうすれば電子を移動させてくれるでしょうか…？

その答えの一つが、「熱を加える」方法です。物質の温度を上げると、中の原子は小さく振動をはじめ、どんどん振動が大きくなります。振動がとても大きくなると、電子をつなぎとめる力がゆるくなり、頑固なケイ素（Si）原子もがまんできずについ電子を手放してしまいます。

しかし、使うたびに素子を火であぶるわけにはいきませんね。かわりに、**ケイ素（Si）原子の集団の中に「不純物」を混ぜ込む（ドープする）**という方法があります。

©MGI inc.

例えば、シリコンの中にリン（P）という元素を混ぜ込みます。リン（P）原子はケイ素（Si）原子と違って外側に「5個」の電子があります。これをケイ素（Si）原子の共有結合に放り込むと…シェアしきれない電子が「1個」あまってしまいます。

このあまった電子はちょっとしたはずみがあれば移動することができ、純粋なシリコンよりは電流が流れやすくなります。リン（P）原子にとっては、はやくどこかへやってしまいたい電子なのです。

このような、不純物を混ぜ込んで電子をあまらせている素材をN型半導体と呼びます。不純物はリン（P）の他にはヒ素（As）やアンチモン（Sb）などがあります。また、もとの原子に束縛されずに動き回ることができる電子を自由電子と呼びます。電子はマイナスの電気の性質を持ち、これを「負電荷」と呼びます。

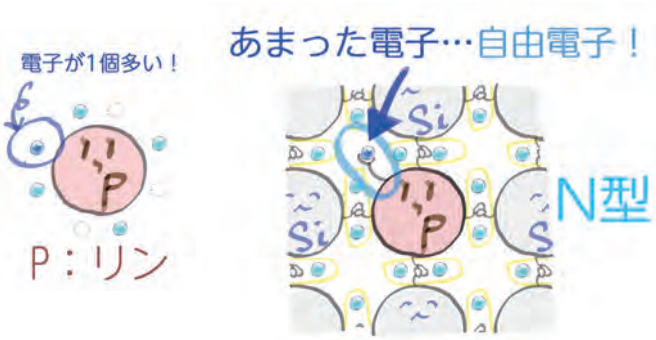


図 3-6(f) 電子をあまらせるリン原子

電子をあまらせる元素があれば、足りない元素もあります。それがアルミニウム（Al）やホウ素（B）、ガリウム（Ga）、インジウム（In）、などの原子です。例えばアルミニウム（Al）原子は外側に「3個」しか電子がありません。これをケイ素（Si）原子の共有結合に放り込むと…電子のシェアが成立しない部分が「1ヵ所」できてしまいます。

この電子の穴のことを、「電子が足りない穴＝正孔（ホール）」と呼びます。「マイナスのマイナスはプラス」といったように、電子の穴はプラスの電気の性質を持つものとみなし、これを「正電荷」と呼びます。このような、不純物を混ぜ込んで電子の足りない穴がある素材をP型半導体と呼びます。

N型半導体の「N」は「負：Negative」、P型半導体の「P」は「正：Positive」ということばから由来しています。

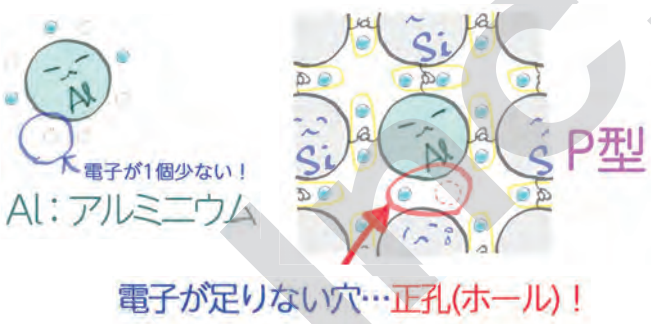


図 3-6(g) 電子が足りないアルミニウム原子

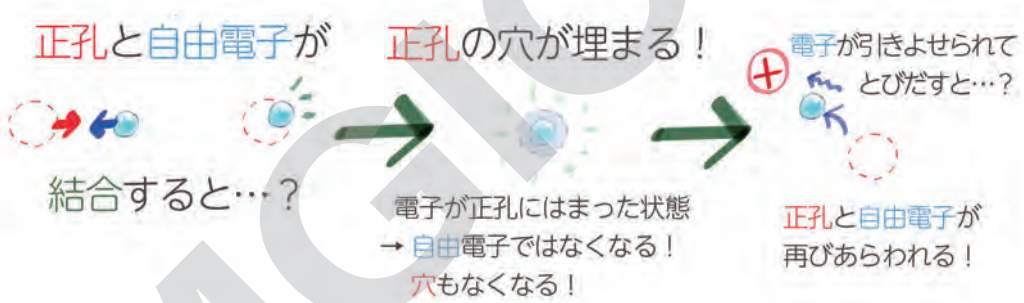


図 3-6(h) 正孔と自由電子

P型半導体やN型半導体はそれぞれ、正電荷や負電荷を移動させる性質があります。そしてこの役割を担っているもの、つまり正孔や自由電子のことをキャリアと呼び、半導体素材の中で量が多い方のキャリアを多数キャリア、少ない方を少数キャリアと呼びます。プラスの電荷である正孔と、マイナスの電荷である自由電子は結合するとお互いの電荷を打ち消し合います。打ち消すといっても、消えるのはあくまで電気のかたよりだけで、正孔も電子もそれ自身が消滅するわけではありません。見た目では、正孔と自由電子がいなくなったように見えるだけです。また、重要な点として次のようなことが言えます。

- P型半導体の中には正電荷のキャリア（正孔）がたくさん存在するが、負電荷のキャリア（自由電子）もほんのすこし存在する。
- N型半導体の中には負電荷のキャリア（自由電子）がたくさん存在するが、正電荷のキャリア（正孔）もほんのすこし存在する。

© Magic Inc.

ところで「正孔は穴なのに、穴が移動するのか？」と不思議に感じるかもしれません。自由電子が次々に隣の穴に移動していくようすを、見方を変えてみればわかりやすくなります。移動するのはあくまで電子ですが、まるで穴が移動するように見えているだけなのです。便宜上、「正電荷が移動している」ように表現されています。

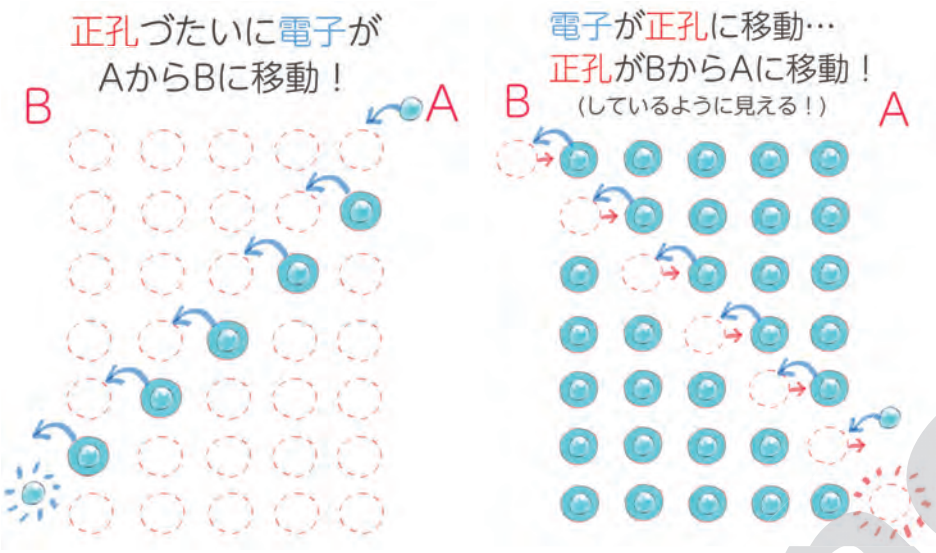


図 3-6(i)　自由電子と正孔の移動

ここまではP型半導体とN型半導体について着目しました。それぞれの半導体素材を、ぴったりくっつけて**接合**したときを見てみましょう。P型とN型の接合面の近くでは、P型の正孔にN型の自由電子が移動する現象が起こります。アルミニウム（Al）原子もリン（P）原子も、近場にいたもの同士で手軽に電子の不足や余分を解消しようとするのです。

このようにして、P型とN型の接合面の近くでは、ほとんどの原子が「8個」ぴったりの電子をもつことができ、みなが満足します。まるで純粋なケイ素（Si）原子の結晶のように、安定した状態になります。接合面近くのこのような層を**空乏層（くうぼうそう）**と呼びます。

図 3-6(j)　正孔と自由電子の結合

…では、接合面より遠いところ、空乏層の外側にいるキャリアはどうなるでしょうか？

P型とN型のあいだには安定した空乏層ができてしまっており、空乏層の原子たちは電子の移動にあまり協力的ではありません。

なぜなら、空乏層のアルミニウム（Al）原子は普段より電子を1個多く得ている状態になるため、マイナスの性質をもつようになります。リン（P）原子も同様に普段より電子が1個少ない状態になるため、プラスの性質をもつようになります。

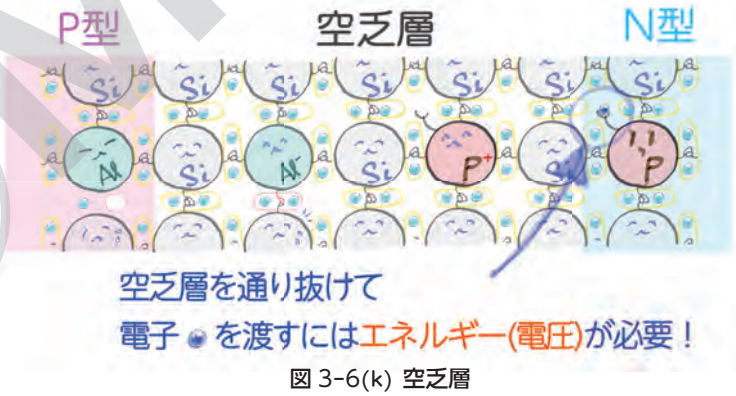


図 3-6(k)　空乏層

空乏層のなかでマイナスとプラスのかたよりが生まれると、N型のキャリアは反対側のP型へ電子を渡しにくくなります。P型近くの空乏層にマイナスの電気（アルミニウム原子）があるせいで、マイナスとマイナスが反発する力が生まれるからです。空乏層の外側のキャリアにとっては、この空乏層をとおりぬけることは一苦労なのです。キャリアが空乏層の反対側へ移動するにはエネルギーが必要になります。空乏層の電気のかたよりが大きいほど、たくさんのエネルギーが必要です。

例えるなら、上り坂を登るとき、急な坂道ほどたいへんな思いをするようなものです。

ダイオードやBJTのベース－エミッタ間には、一定以上の高さの順方向電圧をかけないと電流が流れず、逆方向電圧をかけてもほとんど電流が流れませんでした。半導体としての性質には、物質のふるまいが深く関係していたのです。このような、P型半導体とN型半導体をつなぎ合わせた構造を、**PN接合**と呼びます。

©Mic Inc.

ダイオードのしくみ

PN 接合によって作られる半導体ダイオードを、**接合ダイオード**と呼びます。接合ダイオードでは P 型半導体の側をアノード (A)、N 型半導体の側をカソード (K) と呼びます。

ダイオードのアノード (A) - カソード (K) 間に電圧をかける実験は [3.1] や [3.2] で行いました。ダイオードでは順方向電圧を上げるとやがて電流が流れだします。このように、素子に対して順方向電圧をかけることを「順電圧をかける」もしくは「順バイアスをかける」といった表現がよく使われます。

逆方向電圧では、電流はほとんど流れません。このときも同様に、「逆電圧をかける」もしくは「逆バイアスをかける」と表現されます。

「**バイアス：Bias**」とは「かたより・思い込み・先入観」という意味合いのある言葉です。素子にかける電圧の高さを、順方向や逆方向にかたよらせておくことから、このような言葉が使われています。

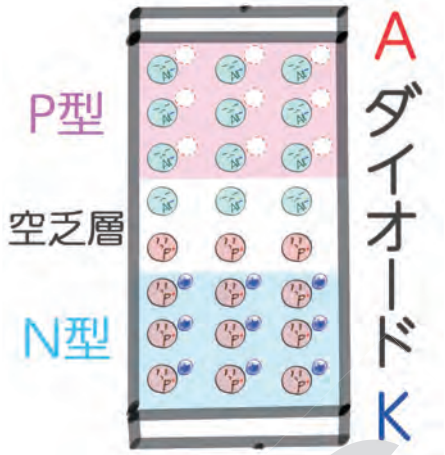


図 3-6(l) 接合ダイオード

ダイオードに「順バイアス」をかけたときを見てみましょう。

- マイナスの電荷 (自由電子) は、プラス極のアノード (A) 側に引き寄せられる
- プラスの電荷 (正孔) は、マイナス極のカソード (K) 側に引き寄せられる

キャリアが空乏層へ向かうエネルギー (電圧) を得ることで、空乏層の作用が小さくなり、空乏層をとびこえて電子が移動をはじめます。つまり、**アノード (A) からカソード (K) に向かって電流が流れる**のです。

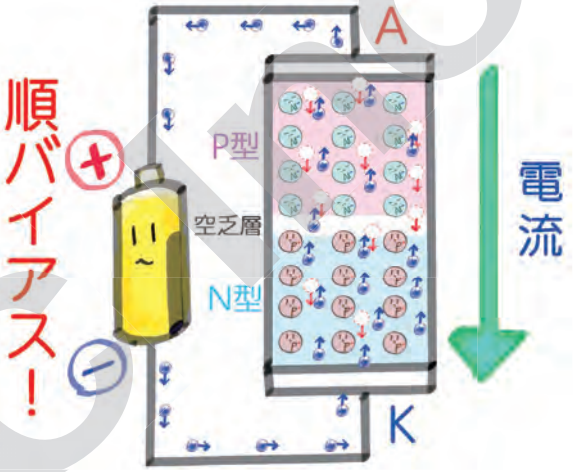


図 3-6(m) 順バイアス

ダイオードに「逆バイアス」をかけたときを見てみましょう。

- マイナスの電荷 (自由電子) は、プラス極のカソード (K) 側に引き寄せられる
- プラスの電荷 (正孔) は、マイナス極のアノード (A) 側に引き寄せられる

キャリアが空乏層と反対側に引き寄せられるため、空乏層がよけいに広がってしまいます。このままでは電子は移動せず、電流はほとんど流れないままです。

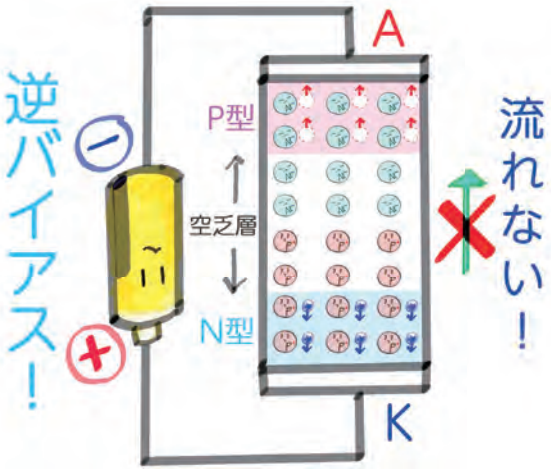


図 3-6(n) 逆バイアス

ダイオードのイラストを見て不思議に思った人もいるかもしれません。電流の方向と、電子の移動する方向が逆になっています。

電流はプラスからマイナスへ向かって流れます。しかし、実は電子は逆向きに移動します。

電流の正体は電子の移動ですが、**電流と電子では流れる方向が逆**です。なぜなら、「電流の向き」を最初に決めた時代では、電子のことがよくわかっていなかったのです。

単に電気製品を利用する上では電子の移動方向はあまり考えなくてもよく、後々になって判明した「電子の移動方向」で人々を混乱させないよう「電流の向き」はそのままにしておこうという約束事を守っています。

ただし、半導体の動作を理解する上では「電子の移動」が重要になることもありますので、ぜひ覚えておいてください。

©mic inc.

■バイポーラ・トランジスタのしくみ

次は、バイポーラ・トランジスタについて見ていきましょう。

接合ダイオードと同じように PN 接合でなりたっていますが、「N-P-N」の順に素材が接合されたかたちになっていて、コレクタ（C）・ベース（B）・エミッタ（E）の端子がそれぞれの半導体素子につながっています。このような構造のトランジスタを NPN トランジスタと呼びます。「NPN」があれば「PNP」もありますが、ここでは NPN について着目します。

コレクタ（C）やベース（B）に使われる半導体の素材の厚さや中に混ぜ込まれる（ドープされる）「不純物」の濃度がそれぞれ異なります。エミッタ（E）には自由電子がたくさん存在し、ベース（B）はとても薄く、コレクタ（C）はエミッタ（E）と比べて自由電子が少なくなっています。

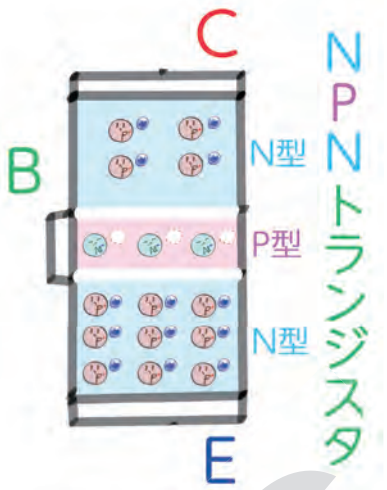


図 3-6(o) NPN トランジスタ

ところで NPN トランジスタでは、ベース（B）から見るとベース（B）- エミッタ（E）間やベース（B）- コレクタ（C）間が接合ダイオードと同じ構造になっていることに気が付きましたか？

「B → C」や「B → E」の方向には電流が流れる構造になっているので、もしも「NPN トランジスタが壊れたかな？」と思ったときはそれぞれの方向に電流が流れるかをテスターなどでチェックすれば、素材が壊れていないかある程度判別することもできます。

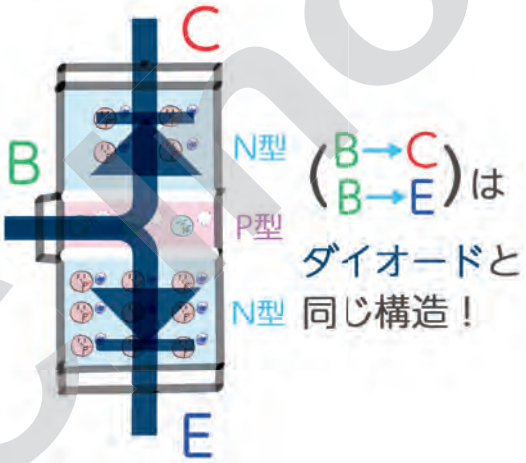


図 3-6(p) ベースから見た構造

NPN トランジスタのコレクタ（C）にプラスの電圧を、エミッタ（E）にマイナスの電圧をかけたときを考えます。

通常、ダイオードでは「N → P」方向へ電流はほとんど流れないはずですが、NPN トランジスタではベース（B）- エミッタ（E）間にかける電圧（バイアス電圧）の向きによって動作のようすが変わります。

- ベース（B）- エミッタ（E）間に逆バイアスがかかっているとき
  「E → B」間の半導体素材の向きは「N → P」になっているので、「E → B」間に電流はほとんど流れません。
  もちろん「C → E」間にも電流はほとんど流れません。ダイオードに逆電圧をかけたときと同じようなふるまいをします。

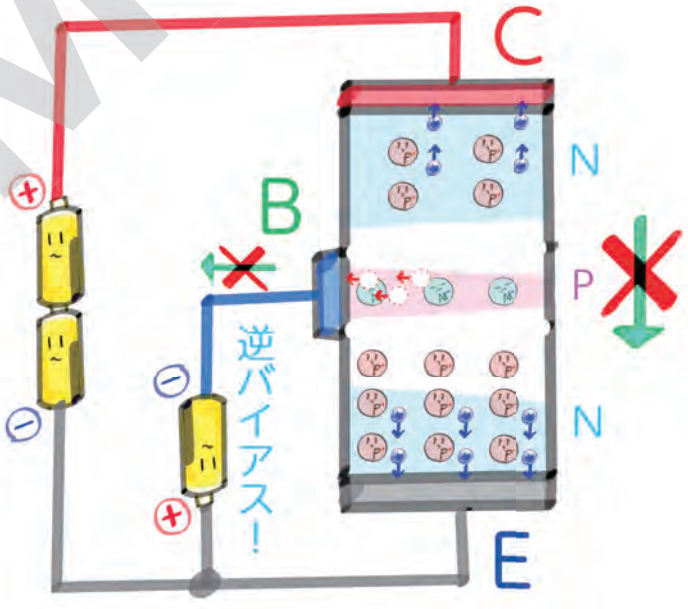


図 3-6(q) 逆バイアス

©MCI Inc.

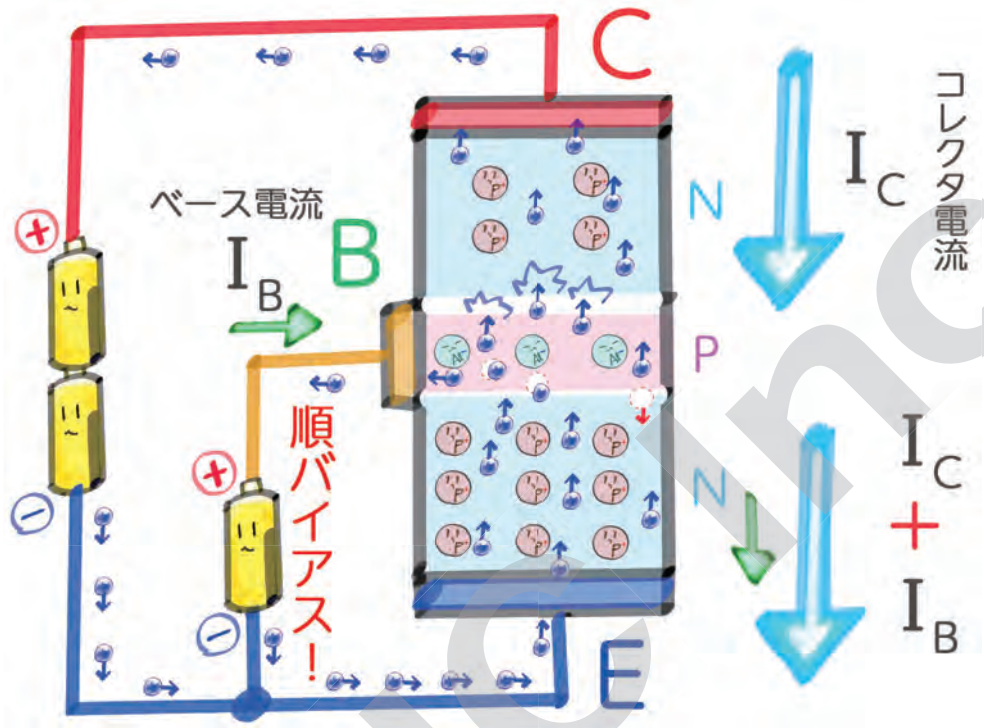
- ベース(B) – エミッタ(E)間に順バイアスがかかっているとき

「B → E」間の半導体素材の向きは「P → N」になっているので、「B → E」間に電流を流すことができます。

ベース(B)からエミッタ(E)に向かって電流が流れるとき、エミッタ(E)のN型半導体の自由電子がベース(B)に向かって移動をはじめます。しかし、エミッタ(E)にはたくさんの自由電子が存在するため、**ベース(B)の正孔と結合しきれなかった自由電子たちが勢いよくコレクタ(C)へと突入**します。

コレクタの電圧はプラスになっているため、突入した自由電子はそのままコレクタに吸い取られていきます。

エミッタから移動を始めた自由電子がコレクタまで移動する、つまりコレクタ – エミッタ間に電流が流れるということです。



エミッタから流れ出るのは
コレクタ電流とベース電流を足したもの！
しかしコレクタ電流とくらべると、
ベース電流は無視できるほど小さい！

図 3-6(r) 順バイアス

エミッタからコレクタに吸い込まれる自由電子の量はベースの正孔につかまる自由電子よりもはるかに多く、この倍率がベース電流に対するエミッタ電流の電流増幅率となります。直流電流に対する電流増幅率を h_{FE} と呼びます。

このような一連の現象が、バイポーラ・トランジスタの増幅動作の原理です。各端子「E」「B」「C」の名前については、「エミッタ(Emitter：発するもの)から放出された電子」を「ベース(Base：基礎)を土台」に「コレクタ(Collector：収集するもの)が吸い寄せる」という動作から由来しています。

また、正孔と自由電子という2つ(2極)のキャリアが電流の道のりに作用することから、**バイポーラ(2極)・トランジスタ**と名が付いています

CQ出版社

第4部
アナログ技術

実験 4-1 オペアンプを動かす …… 78
実験 4-2 加算回路 …… 82
解説 4-3 信号のかたち …… 84
実験 4-4 2音の合成 …… 86
実験 4-5 さまざまな演算回路 …… 88
実験 4-6 ボルテージフォロワ …… 92
コラム 電子回路の“タンス” …… 95
解説 4-7 オペアンプの基本動作 …… 96
実験 4-8 コンパレータ …… 98
実験 4-9 光センサー（CdSフォトセル）を使う …… 100
コラム センサー制御とマイコンプログラミング …… 101
実験 4-10 暗くなると点灯するライト …… 102
実験 4-11 明るさの差を判別する …… 104
実験 4-12 マイク信号の増幅 …… 105
実験 4-13 ホールセンサーで磁気測定 …… 108
実験 4-14 超音波センサーで送受信 …… 109
実験 4-15 配置・配線によるノイズの発生 …… 112
解説 4-16 活躍する半導体製品群 …… 117

©MSC Inc.

4-1 オペアンプを動かす

実験

学習ポイント

トランジスタを用いる増幅回路は部品点数が多くなり、動作点や流す電流の設計と計算が重要となるため、現在ではあまり使われていません。しかし、トランジスタの増幅動作や理論をもとにした集積回路の技術が進んだおかげで、より小型で手軽に増幅動作を行えるオペアンプが登場しました。オペアンプの中にはたくさんのトランジスタ回路が詰め込まれています。

チェック

オペアンプ（OPAmp）は**演算増幅器**（Operational Amplifier）とも呼ばれる増幅素子です。使い方によって増幅のほかにも足し算や引き算、掛け算や割り算、積分や微分といった電気的な「**演算**」を行うことができます。まずは基本的な**増幅回路**を扱ってみましょう。

実験で用いるオペアンプ「7043D」の写真と回路図記号を図4-1(a)に示します。

回路図では三角形の記号が1個分の増幅器を表現します。オペアンプの種類によっては増幅器が2個～4個分入っているものもあり、7043Dには2個の増幅器が搭載されています。

1個の増幅器には入力端子が2本あり、それぞれ非反転入力端子：IN+（プラス入力端子）、反転入力端子：IN－（マイナス入力端子）と呼びます。また、出力は出力端子：OUTです。オペアンプの本体に複数の増幅器が入っている場合は、IN+1やOUT2といった番号を付けて割り振られています。

また、本体上面にある丸いくぼみが1番ピンの位置をあらわしており、1番ピンから左回りに「1～8」番のピン番号が割り振られています。**接続するときは向きとピン番号に注意しましょう。**

V_{DD}端子には電源のプラス側、V_{SS}端子には電源のマイナス側を接続します。

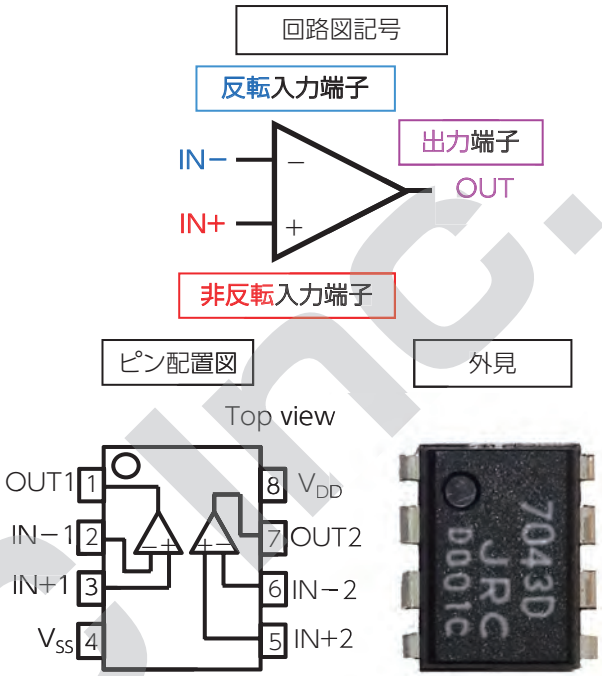


図4-1(a) オペアンプ

回路を作ろう！1

最初に、**非反転増幅回路**を作ります。

①図4-1(c)の接続図を参考に、図4-1(b)の回路を組みましょう。

②オペアンプの8番ピン（VDD）は電源5V、4番ピン（VSS）はグランド（GND）と接続します。**以降の回路図記号では電源ピンとGNDピンの表記は省略します。**

③オペアンプのOUT（出力端子）とIN−（マイナス入力端子）を抵抗でつなぎます。

④出力が入力に戻るようにつなぐことを「**帰還をかける**」や「フィードバックする」と表現します。この回路ではオペアンプのOUT端子とIN−端子（マイナス入力）を抵抗でつなげており、この形を**負帰還**と呼びます。

⑤抵抗**Rf**は**帰還抵抗**とも呼ばれ、「f」は「帰還：Feedback」の言葉から由来しています。

⑥オペアンプの増幅回路は、通常は負帰還をかけて使います。詳しくは実験の後の補足の項で解説します。

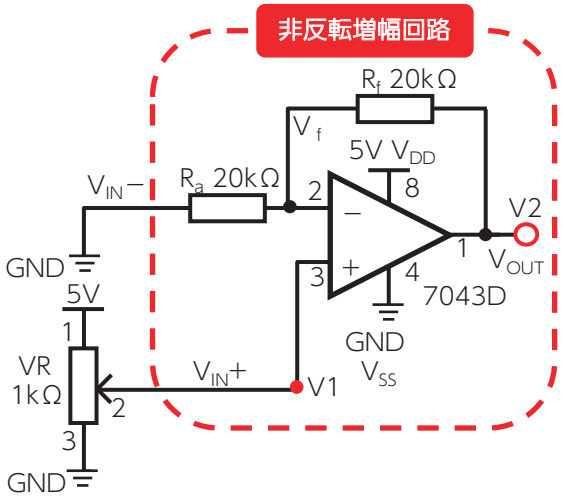


図4-1(b) 回路図

OMRON

回路を作ろう！①

※ 2個入っているオペアンプの1個だけを使います

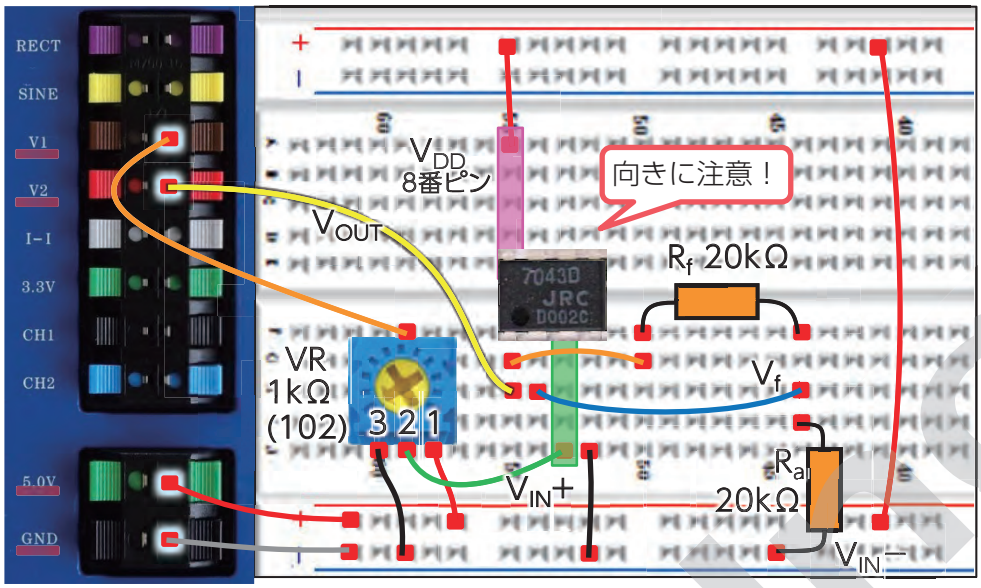


図 4-1(c) 接続図

実験のすすめ方

①VR のダイヤルを調整し、ワークシートの表にそって $V_{IN}+$（プラス入力）に入力する電圧を 0V から 5V に変化させる。
②それぞれの入力電圧について、V_{OUT}（出力）の電圧を記録する。

ワークシートに記入しよう！

考えてみよう

①出力電圧は、「電圧 $V_{IN}+$（プラス入力）と電圧 $V_{IN}-$（マイナス入力）の差」の約何倍になりましたか？ $V_{IN}-$（マイナス入力）はグランド（GND）につないでいるため 0V とします。
②電圧 $V_{IN}+$（プラス入力）を上げても出力が変化しなくなったのは、$V_{IN}+$（プラス入力）が約何ボルトのときからですか？ また、そのときの出力 V_{OUT} は最大何ボルトでしたか？
③オペアンプが増幅できる最大出力電圧は、何の電圧が影響すると言えますか？

ワークシートに記入しよう！

発展

①出力電圧が約 5V より低いときの電圧 V_f を測定し、電圧 $V_{IN}+$（プラス入力）と値を比較してみよう。
②入力抵抗 R_a を 10kΩ に変更して同様の実験を行おう。
③それぞれ得られた結果について出力と入力の関係から増幅度 Ao を求めよう。出力電圧が「電圧 $V_{IN}+$（プラス入力）と電圧 $V_{IN}-$（マイナス入力）の差」の約何倍になったかを計算することで求めることができます。
④計算した増幅度 Ao が $(1 + \frac{R_f}{R_a})$ の値とほぼ同じになることを確認しよう。

ワークシートに記入しよう！

©MIC Inc.

回路を作ろう！②

次は、**反転増幅回路**を作ります。

①図 4-1(e) の接続図を参考に、図 4-1(d) の回路を組みましょう。

②V_{IN}－（マイナス入力）側にコンデンサをつなぎ、VR で正弦波（SINE）を入力します。

③VR のダイヤルは左いっぱいに回しておきます。

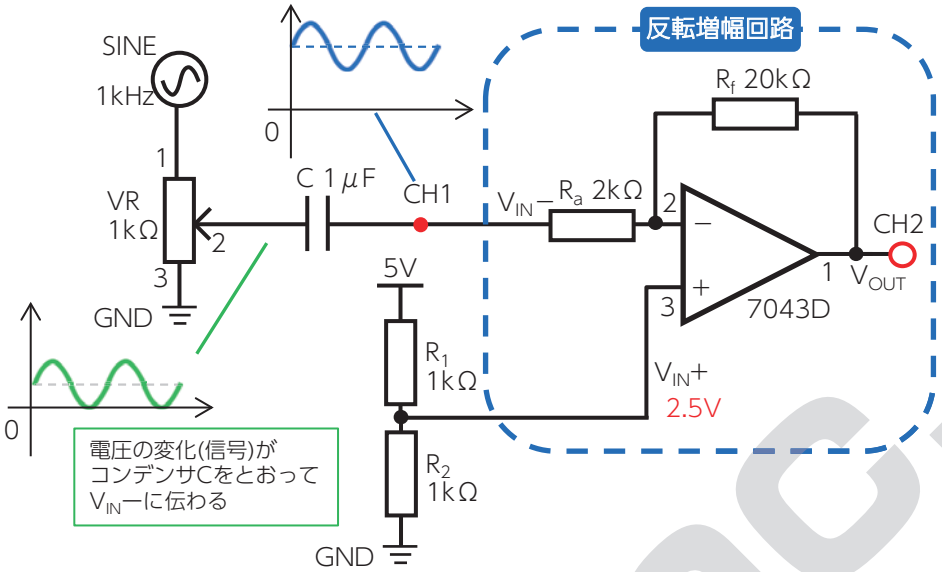


図 4-1(d) 回路図

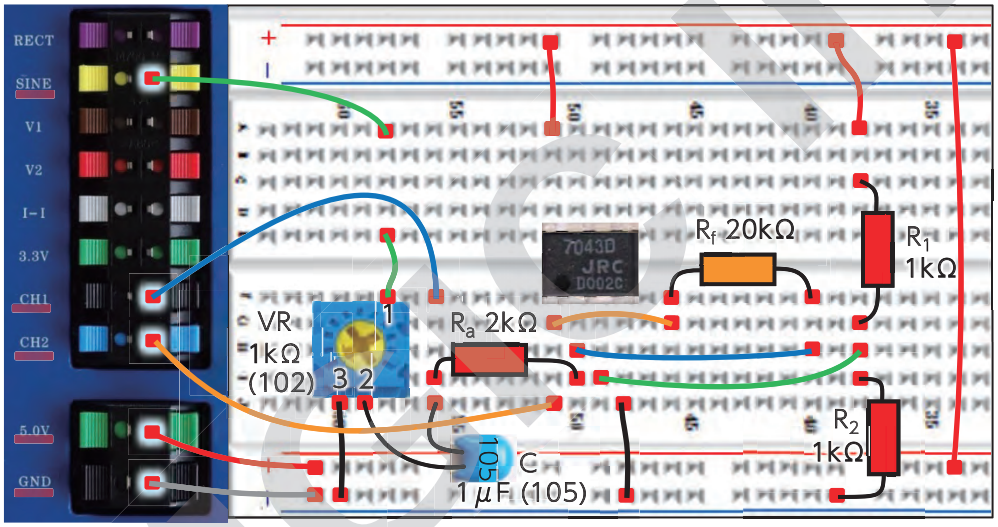


図 4-1(e) 接続図

実験のすすめ方

①VR のダイヤルを左いっぱいに回しているとき、出力(CH2)の最大値・最小値ともに約 2.5V になることを確認する。
②VR のダイヤルを回して出力(CH2)の波形が変化することを確認する。
③入力(CH1)の Vp-p 値（最大値から最小値までの幅）が約 0.3V 程度になるように VR のダイヤルを調整する。
④このときの出力(CH2)の Vp-p 値を記録する。
⑤同様に、入力信号の Vp-p 値をワークシートにそって変化させ、それぞれの出力信号の Vp-p 値を記録する。

ワークシートに記入しよう！

考えてみよう

①出力(CH2)の Vp-p 値は、入力(CH1)の Vp-p 値の約何倍になりましたか？
②入力(CH1)の電圧が上昇すると、出力(CH2)の電圧は上昇しますか？ それとも低下しますか？
③入力(CH1)の Vp-p 値が約 0.5V を超えると、出力(CH2)の波形はどのように変化しましたか？

ワークシートに記入しよう！

発展

①入力抵抗 R_a を 10kΩ に変更して同様の実験を行おう。
②それぞれ得られた結果について出力と入力の関係から増幅度 A_o を求めよう。出力電圧が「電圧 V_{IN}+（プラス入力）と電圧 V_{IN}－（マイナス入力）の差」の約何倍になったかを計算することで求めることができます。
③計算した増幅度 A_o が $\frac{R_f}{R_a}$ の値とほぼ同じになることを確認しよう。

ワークシートに記入しよう！

©Micro Inc.

補足 差動増幅とは？

オペアンプによる信号増幅は、2本の入力端子の「**入力電圧の差を増幅する**」という方式で行われます。これを「**差動増幅**」と呼びます。

一般的なオペアンプの増幅度は数万倍もの倍率があり、ごくわずかな電圧の差も大きく増幅することができます。

ただし、**増幅される信号は電源電圧の範囲を超えることはできません。**

通常、数万倍もある増幅度は扱いにくいものです。そこで、オペアンプの回路に入力抵抗 R_a と帰還抵抗 R_f を置くことで負帰還をかけると、増幅度を実用的な範囲に調整することができます。

補足 反転？ 非反転？

信号の反転とは電圧の変化が上下逆になったり、プラスがマイナスに、マイナスがプラスになったりするようなことを言います。

増幅回路の出力が反転するかしないかは信号を入力する側を非反転入力端子：IN+（プラス入力）、反転入力端子：IN−（マイナス入力）のどちらにするかで変わります。

［差動増幅回路］ $V_{OUT} = Ao \times (V_{IN}^{+} - V_{IN}^{-})$ …(1)
差動増幅回路の出力電圧は、「出力電圧＝増幅度×（プラス入力電圧とマイナス入力電圧の差）」です。

［負帰還抵抗があるときの増幅度］ $Ao = (1 + \frac{R_f}{R_a})$（非反転時）　　$Ao = (\frac{R_f}{R_a})$（反転時）

例えば、マイナス入力を 0V（GND）に固定してプラス入力側に信号を入力すると非反転増幅となり、

式 (1) より $V_{OUT} = (1 + \frac{R_f}{R_a}) \times (V_{IN}^{+} - 0)$ という関係になります。

同様に、プラス入力電圧を 2.5V に固定してマイナス入力側に信号を入力すると反転増幅となり、

式 (1) より $V_{OUT} = \frac{R_f}{R_a} \times (2.5 - V_{IN}^{-})$ という関係になります。

オペアンプの増幅動作を考えるときの参考にしてください。

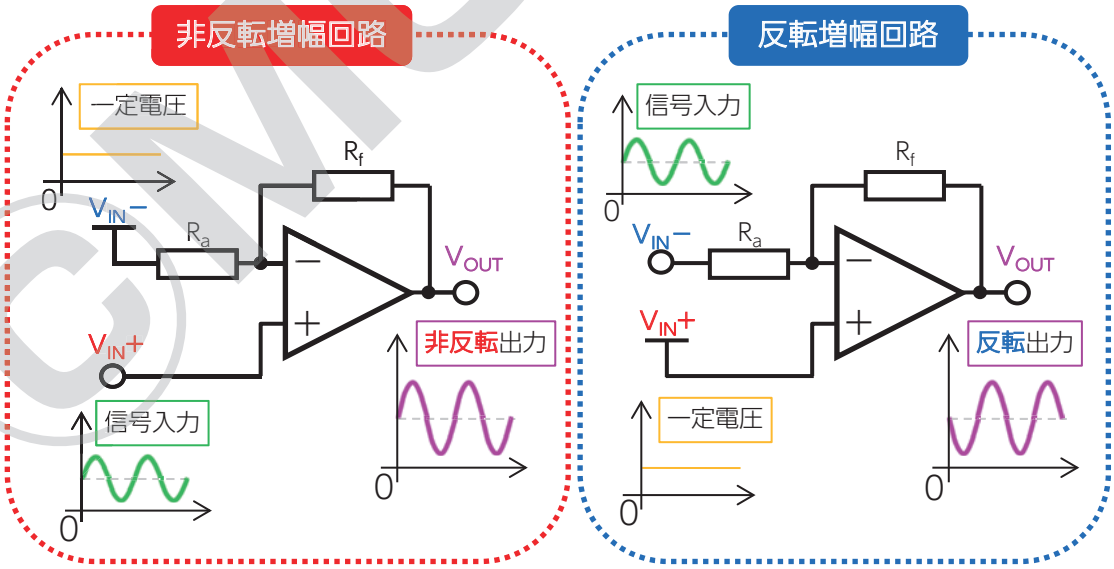


図 4-1(f) 非反転と反転

CMGIC Inc.

4-9 光センサー（CdS フォトセル）を使う

実験

学習ポイント

光センサーである CdS フォトセルを使った実験を行います。ここからオペアンプとセンサー素子を組み合わせた回路を使って、センシング技術を学習していきます。センサー素子の特性を知ることから始めましょう。

チェック

CdS フォトセルはフォトレジスタとも呼ばる半導体の一種で、硫化カドミウム (CdS) を主成分とする素子です。受光部に光が当たると端子間を電子が移動しやすくなるという特性があります。つまり、明るさで抵抗値が変化します。

CdS フォトセルを抵抗と同じように回路に組み込むと、抵抗値の変化によって電圧が変動するため、センサー端子間の電圧を測定することで明るさをとらえることができるのです。明るいときの抵抗値を**明抵抗（めいていこう）**、暗いときの抵抗値を**暗抵抗（あんていこう）**と呼びます。

また、使用するにあたっては**極性（向き）は存在しない**のでどちら向きにつないでも問題ありません。

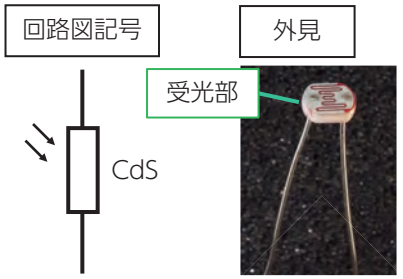


図 4-9(a) 光センサー（CdS フォトセル）

MI527の電気的・光学的特性	
使用温度範囲	−30℃〜+70℃
暗抵抗	1MΩ
明抵抗(10ルクス時)※	10k〜20kΩ

※10ルクスとは、ろうそくの炎程度の明るさを指す

回路を作ろう！

CdS フォトセルに LED の光を当てて、抵抗値の変化を測定します。

①図 4-9(c) の接続図を参考に、図 4-9(b) の回路を組みましょう。
②LED をつなぐ向き（極性）に注意して接続しましょう。
③**VR のダイヤルは左いっぱいに回しておきます。**

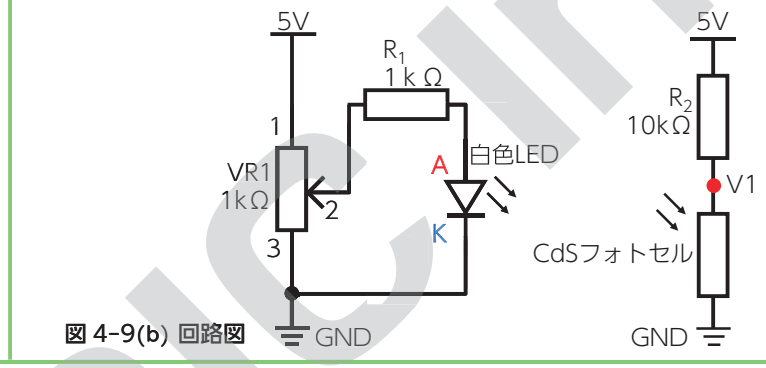


図 4-9(b) 回路図

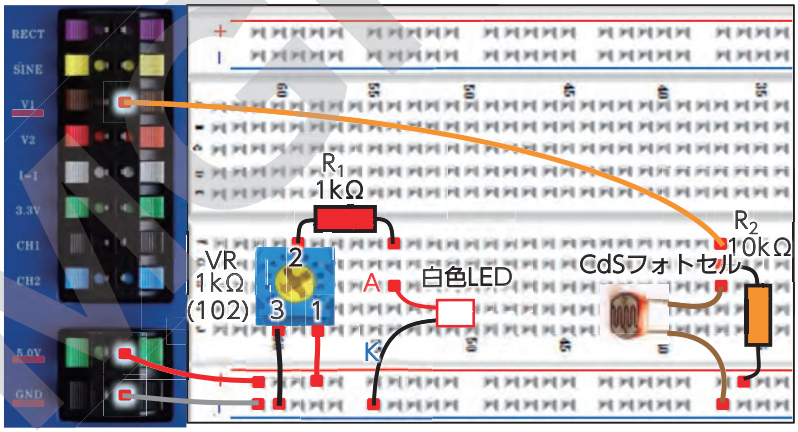


図 4-9(c) 接続図

実験のすすめ方

①**LED が点灯していないとき**、光センサーの電圧 V1 がどのような値になっているかを測定し記録する。
②光センサーの受光部の回りを物体で隠し、受光部の周囲が最も暗いときの電圧 V1 を測定し記録する。
③VR のダイヤルを回して LED を点灯させ、光センサーの受光部を照らす。
④光の強さの変化に合わせて電圧 V1 が変化することを確認する。
⑤VR を右いっぱいに回し、受光部の周囲が最も明るいときの電圧 V1 を測定し記録する。

ワークシートに記入しよう！

考えてみよう

①光センサーの電圧は、受光部の周囲が暗くなったとき上昇しましたか？ それとも低下しましたか？
②受光部の周囲が暗くなると、光センサーの抵抗値はどのように変化すると言えますか？
③受光部の周囲が最も暗いときの光センサーの抵抗値を計算しよう。
④受光部の周囲が最も明るいときの光センサーの抵抗値を計算しよう。

ワークシートに記入しよう！

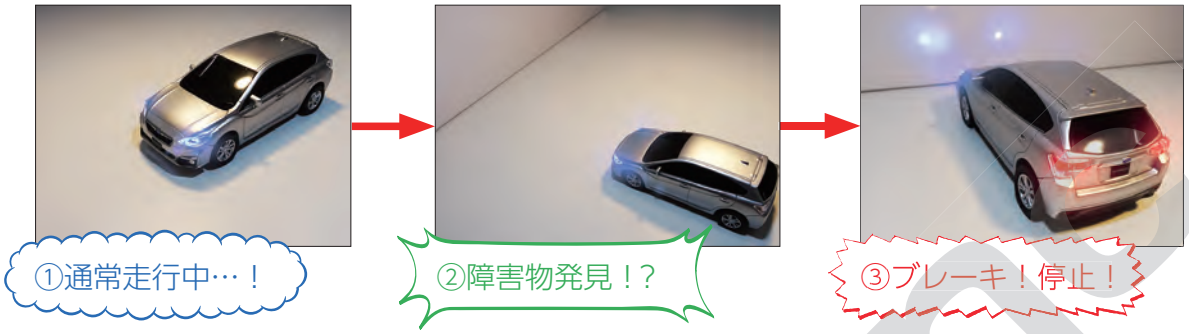
©Micro

コラム

センサー制御とマイコンプログラミング

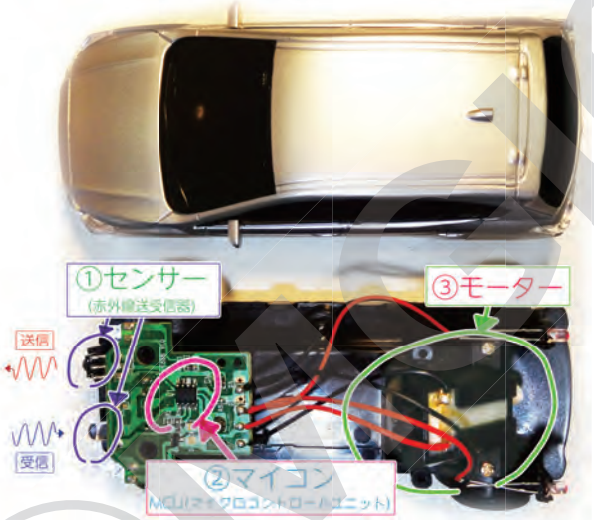
プログラミングによってマイコンに特定の機能をもたせ、センサー素子や負荷などの周辺回路を制御するという手法は「**組み込み技術**」や「**エンベデッド：Embedded**」などとも呼ばれ、このような技術の進歩によってさまざまな製品・産業機械が生み出されています。

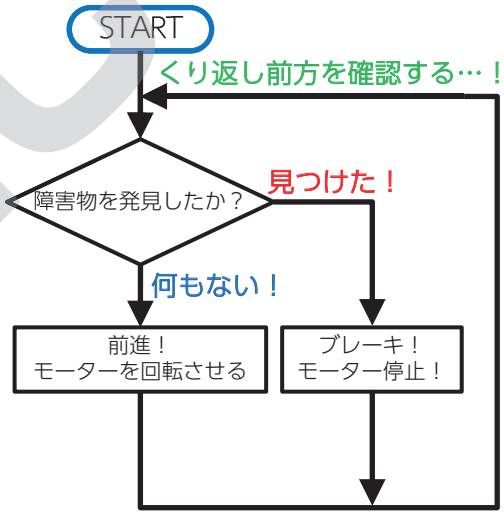
センサー機能をもつミニカーを例に、センシング技術の一端に触れてみましょう。このミニカーは「通常は走行を続ける」「障害物があれば停止する」「障害物がなくなれば再び走行する」という機能をもっています。



センサーが障害物を感知するとミニカーはモーターの回転を止めて停止します。

リモコンはなく、人は「操作」していません。しかし何者かがミニカーを「操作」しているのはまちがいありません。このミニカーの中身を見てみましょう。ライトなどの LED をのぞくと、大きく分けて 3 つの機構で成り立っています。





①センサー

「目」にあたる部分です。赤外線を前方に送りだしたとき、障害物があれば反射されて戻ってくるため、それを受信することで前方の物体の有無を確認することができます。

②マイコン

「脳」にあたる部分です。センサーがとらえた受信信号を確認し、前進を続けても大丈夫かを判定してからモーターを制御します。

③モーター

「足」にあたる部分です。マイコンから「モーター回転」の命令が出ているときだけ前進し、「モーター停止」の命令が出ると停止します。

このシステムの中でマイコンは司令塔としての役割があり、赤外線送受信器の信号やモーターの動力をうまく判定・コントロールしています。

このように制御技術は「制御される対象（センサー・モーター）」と「制御するもの（マイコン・コンピュータ）」からなりたちます。

マイコンプログラミングとは、ただプログラミングができるというだけではなく「**いかに制御対象のふるまいを理解してプログラミングを行えるか**」がとても大切なのです。

第5部
デジタル技術

実験 5-1 7セグメントLEDを使う …… 120
解説 5-2 論理回路 …… 124
解説 5-3 デジタルデータの単位・ビット …… 126
実験 5-4 2ビットデコーダ回路 …… 127
実験 5-5 メモリセルを作る …… 129
解説 5-6 さまざまな論理回路 …… 131
実験 5-7 NOTゲート(インバータ)を作る …… 132
実験 5-8 NANDゲート・NORゲートを作る …… 134
コラム パワーMOSFETの構造 …… 137
実験 5-9 標準ロジックICを使う …… 138
解説 5-10 危険なラッチアップ現象 …… 141
実験 5-11 NOTゲートIC …… 142
実験 5-12 NANDゲートIC …… 144
実験 5-13 ORゲートIC …… 146
実験 5-14 XORゲートを作る …… 148
実験 5-15 RSフリップフロップ …… 150
実験 5-16 Dフリップフロップ …… 152
解説 5-17 組み合わせ回路・順序回路 …… 154
実験 5-18 分周回路 …… 155
実験 5-19 4ビットバイナリカウンタ …… 158
実験 5-20 4ビットアップダウンカウンタ …… 160
解説 5-21 ハザード・レーシング・チャタリング …… 163
実験 5-22 10進カウンタ …… 165

©MSIC Inc.

5-2 論理回路

解説

ここでは、デジタルの情報を扱う上で欠かせないさまざまな考え方について解説します。
アナログは連続した量を扱いますが、**デジタルでは「1個」「1個の次は2個」「2個の次は3個」のようにはっきりと値を数えます**。

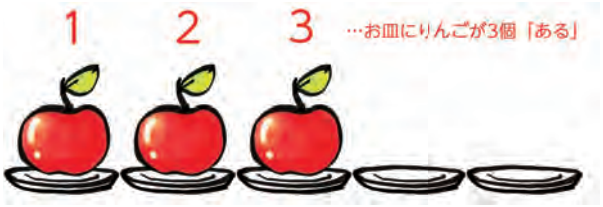


図 5-2(a)「ある」か「ない」か

コンピュータや電子回路でデジタルで情報を扱うとき、**「ある」か「ない」かという状態で考えることがとても大切です**。
デジタル技術ではしばしば、さまざまな情報を「ある」か「ない」か、「1」か「0」かのような2つの状態であらわして扱います。

■論理回路

デジタル情報処理では**論理演算**が用いられます。論理演算とは、**「1」か「0」かを入力すると「1」か「0」が出力されるというものです**。入力が門を通過して出力されるようすから、論理演算を行うものを**論理ゲート**と呼びます。

論理演算そのものは難しいものではありません。ほとんどの論理回路では「1」か「0」かの2種類の値を扱い、この2値のことを**論理値**と呼びます。

論理値は「真(true) = 1」か「偽(false) = 0」かとしてもあらわされます。

電気の世界ではしばしば「電圧や電流などがゼロと言える状態」を「偽(0)」として、「ゼロとは言えない状態」を「真(1)」として論理値を扱います。このような、**論理演算の入力 / 出力を電気で表現する電子回路**のことを**論理回路**と呼びます。

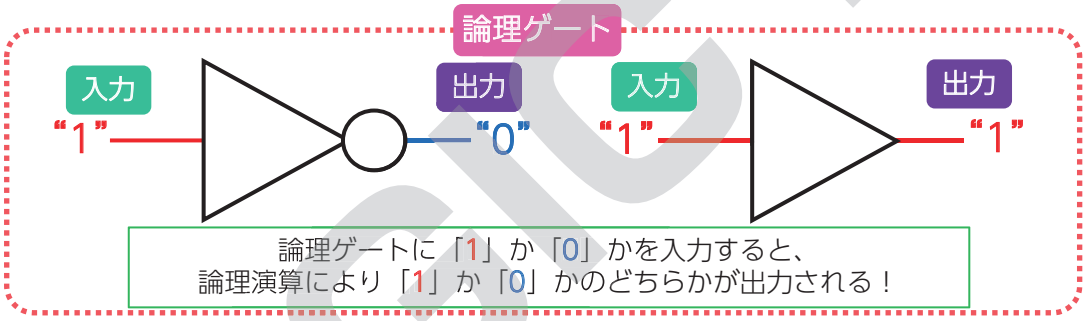


図 5-2(b) 論理ゲート

■基本ゲート：NOT・AND・OR

基本的な論理演算として、NOT 演算・AND 演算・OR 演算の 3 つの論理回路があります。

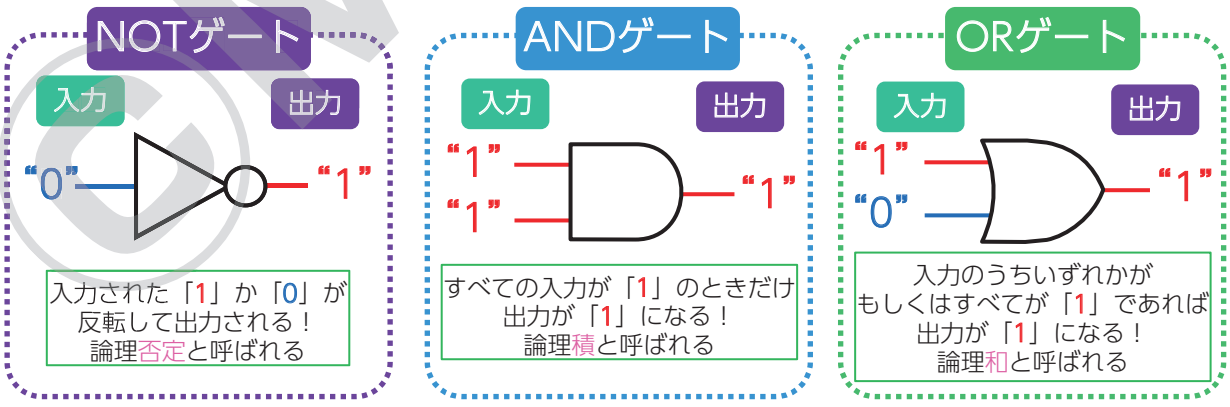


図 5-2(c) NOT・AND・OR

©MCG Inc.

■真理値表

真理値表とは、入力する値のパターン全てに対応する出力を表にしたものです。

例えば OR 演算であれば「入力 A が 0、入力 B が 0 で出力が 0」「入力 A が 1、入力 B が 0 で出力が 1」のような全ての入力パターンの一覧になります。

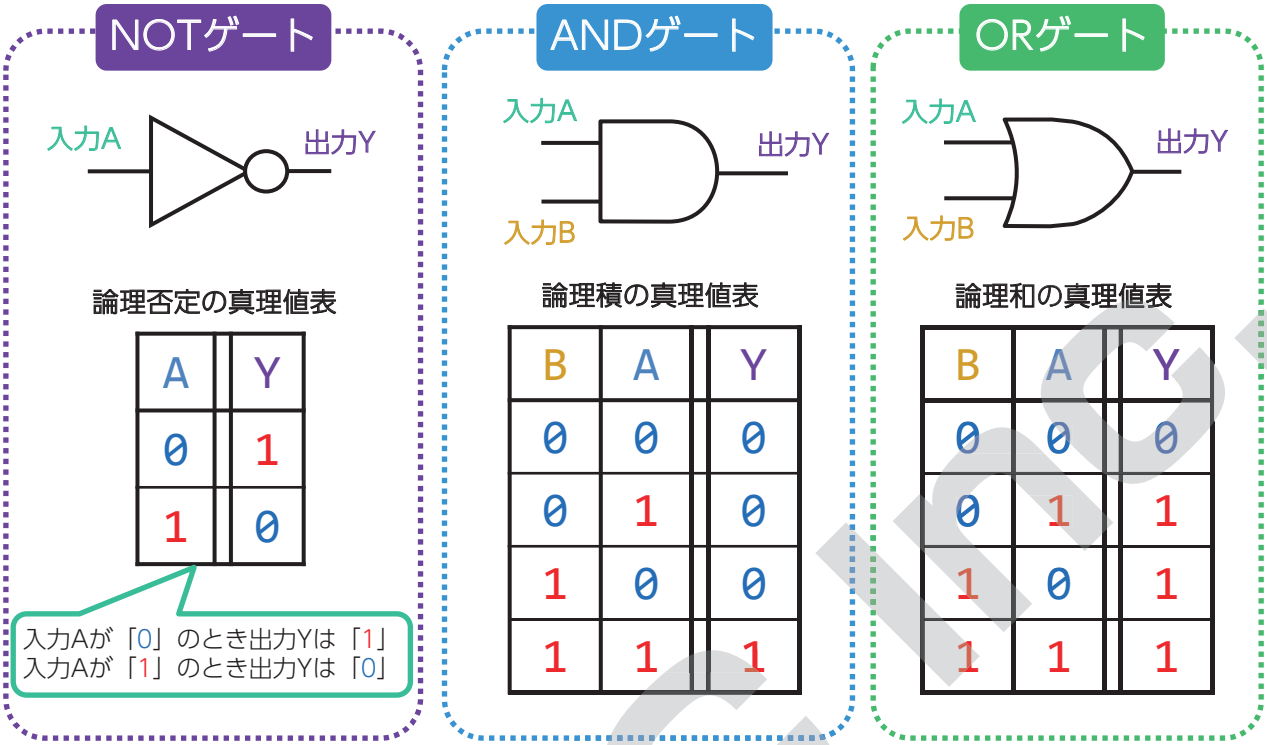
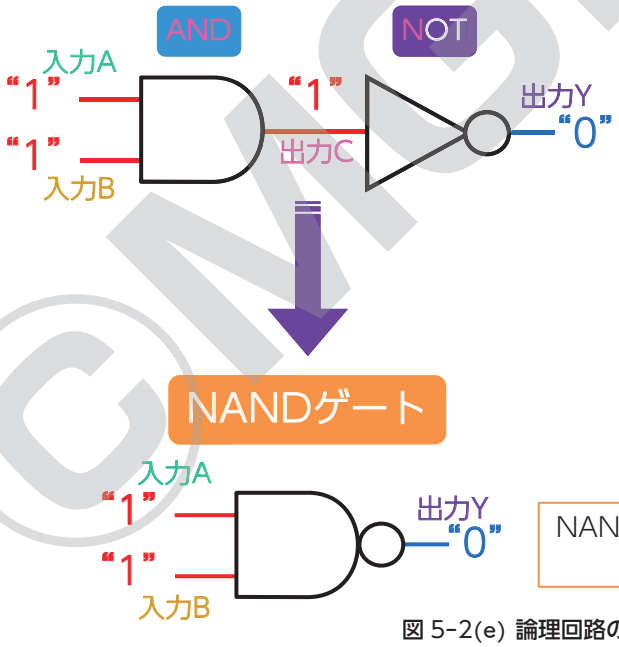


図 5-2(d) **真理値表**

■論理回路の組み合わせ

NOT や AND、OR を組み合わせたり、入力のつなぎ方を変えることでさまざまな入出力のパターンを作ることができます。
その中でも AND ゲートと NOT ゲートを組み合わせたものは **NAND(ナンド)** ゲートと呼ばれ、論理回路でよく用いられます。



AND・NOTを組み合わせた真理値表

C	B	A	Y
0	0	0	1
0	0	1	1
0	1	0	1
1	1	1	0

NANDゲートの出力は、AND出力の否定(NOT)！
否定論理積と呼ばれる

図 5-2(e) **論理回路の組み合わせ**

練習

①ワークシートの論理回路図を見て、入力 / 出力のパターンを真理値表に書き込もう。
②真理値表のパターンを見て、論理回路図を描いてみよう。

ワークシートに記入しよう！

©MSC Inc.

5-11 NOT ゲート IC

実験

学習ポイント

74HC04 の中には 6 回路分の NOT ゲートが入っています。いくつかの NOT ゲートを組み合わせて実験してみましょう。

回路を作ろう！ 1

①図 5-11(c) の接続図を参考に、図 5-11(a) の回路を組みましょう。
②ロジック IC「74HC04」の中の NOT ゲート 4 回路を直列につなぎます。
③使用しない入力ピンはグランド(GND)もしくは 5V ラインにつないでおきましょう。

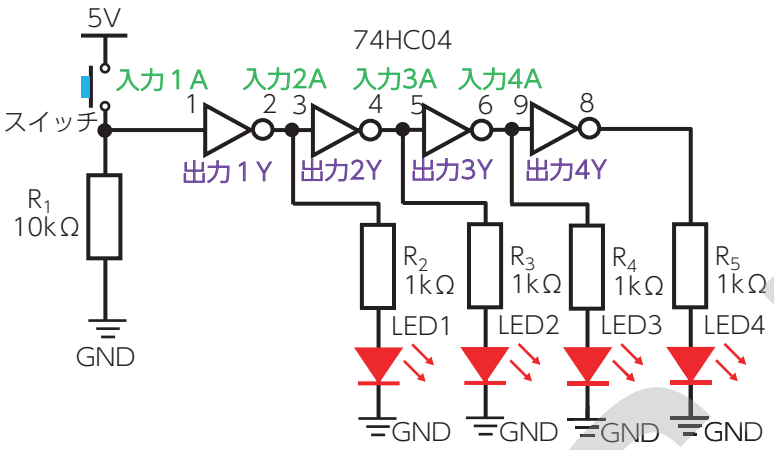


図 5-11(a)　回路図

ヒント

IC のピン配置図

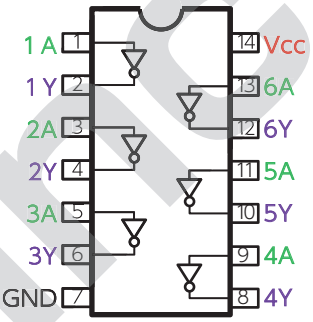


図 5-11(b)　NOT ゲート IC

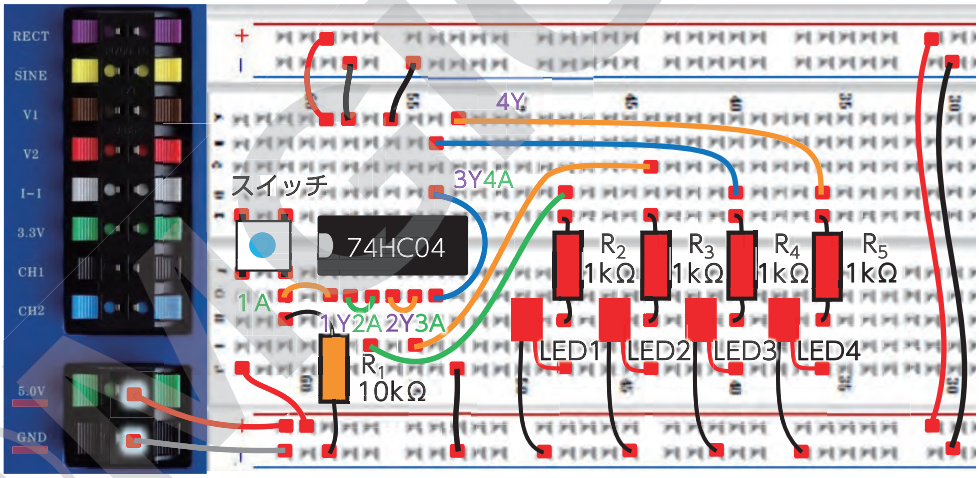


図 5-11(c)　接続図

実験のすすめ方

①スイッチを押していないときにどの LED が点灯しているかを確認する。
②スイッチを押しているときにどの LED が点灯しているかを確認する。

ワークシートに記入しよう！

考えてみよう

①74HC04 の 1 番ピンの入力電圧が 5V のとき、何番の出力ピンが 5V になりますか？ 複数ある場合はそれぞれ答えよう。
②74HC04 の 8 番ピンの出力電圧が 0V になるには、1 番ピンに何ボルトの電圧をかければよいですか？

ワークシートに記入しよう！

©mic Inc.

回路を作ろう！②

①入力電圧の切り替えをスイッチではなく矩形波に変更して、簡単なイルミネーションを作ってみましょう。
②図 5-11(f) の接続図を参考に、図 5-11(d) の回路を組みましょう。
③スイッチ部分を外し、入力を RECT 端子と接続します。

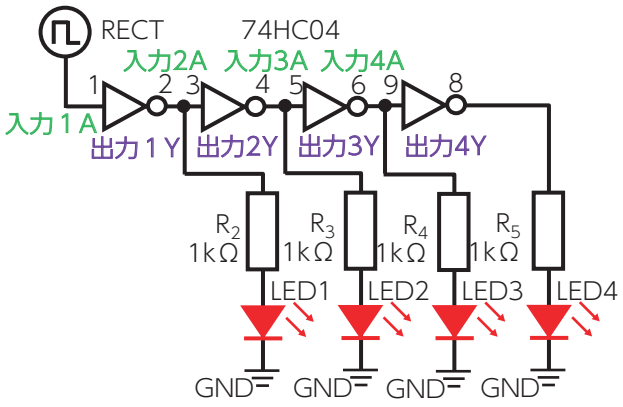


図 5-11(d)　回路図

ヒント　信号を確認しよう

回路がうまく動作しないときはまず**最初に電源・GND の接続を確認**しましょう。次に、オシロスコープや電圧計を活用して**入出力信号や電圧を確認**し、不具合の原因をつきとめましょう。信号の周波数や振幅に合わせて、「OP：Time」や「OP：Gain」の設定を調整してください。

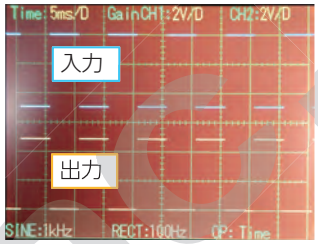


図 5-11(e)　信号を確認

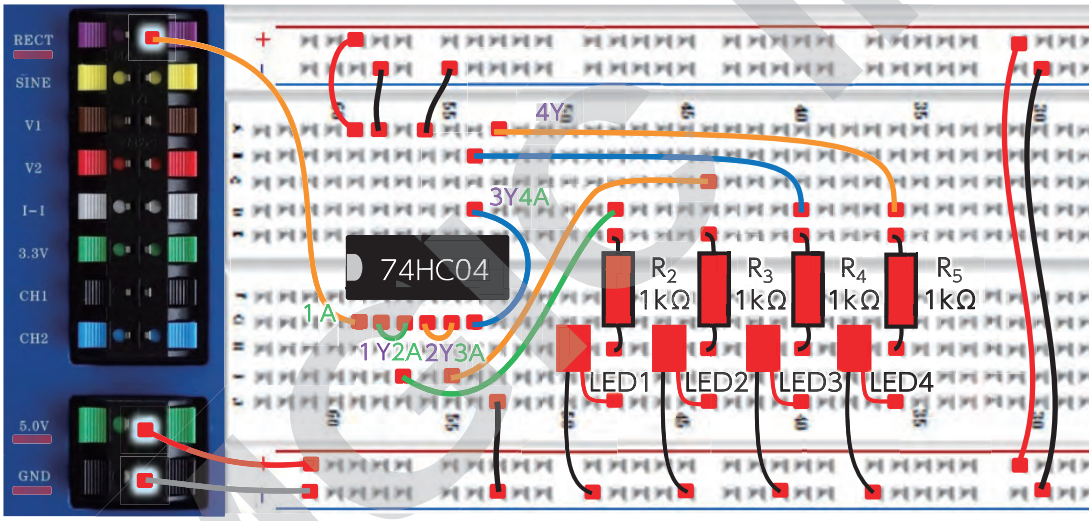


図 5-11(f)　接続図

実験のすすめ方

①矩形波（RECT）の周波数の設定を、電源投入時の初期設定である「1kHz」から少しずつ下げていき、各 LED の点灯 / 消灯のようすを観察する。
②点滅が早すぎて目に見えない場合は、矩形波（RECT）の周波数の設定をさらに下げて観察する。

考えてみよう

①各 LED がはっきりと点滅して見えるようになったのは、矩形波（RECT）の周波数が何ヘルツ（Hz）からでしたか？
②矩形波（RECT）の周波数が「1kHz」のとき、各 LED の明るさは点滅して見えているときと比べて明るかったですか？ それとも暗かったですか？ わからなければ、もう一度周波数を戻して確認しよう。

ワークシートに記入しよう！

補足

LED の明るさ

周波数によって、LED の明るさが違って見えます。LED が点灯 / 消灯をくり返しているために起こる**目の錯覚**ですが、電子回路ではこの原理を応用して照明の明るさやモーター動力などの電力制御を行う方法があります。詳細は本テキスト後半の項で解説します。

5-16 D フリップフロップ

実験

学習ポイント

デジタル技術の中でもよく用いられる順序回路のひとつ、D フリップフロップの動作を学習します。

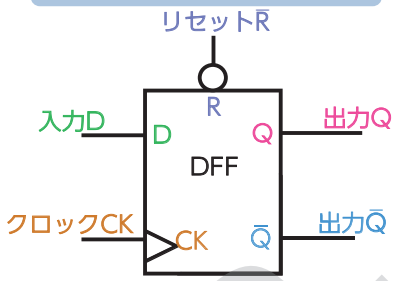
チェック

D フリップフロップはフリップフロップの一種で、クロック（時計：Clock）と呼ばれる信号が「1」になったときの入力の値が出力に反映され、次にクロック信号が再び「1」になるタイミングまで出力が保持されるという動作を行います。出力の変化が遅れることから、遅延を意味する「ディレイ：Delay」の頭文字をとって D フリップフロップ（DFF）と呼ばれています。

通常、D フリップフロップには入力データ D とクロック CK があたえられ、出力 Q からデータが出力されます。

クロックや信号の立ち上がりエッジ、立ち下がりエッジなどの動作については実験後の補足で解説を行います。まずは、DFF が 4 回路入った IC「74HC175」を使用して DFF の動作を簡単に実験してみましょう。

Dフリップフロップ(DFF)



DFFの真理値表

D	CK	R̄	Q
0	↑	1	0
1	↑	1	1
X	↓	1	変化なし
X	X	0	0

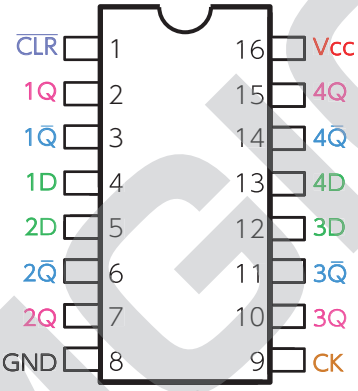
↑はクロック信号の立ち上がりエッジをあらわし、
↓はクロック信号の立ち下がりエッジをあらわす。
Xは「Don’t Care」、つまりどの状態でもよい。

図 5-16(a)　D フリップフロップ

外見

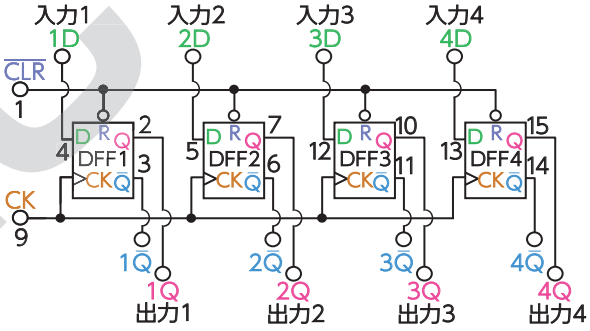
74HC175

ピン配置図



74HC175
“DFF4回路入り”

概略図



CLRは全データクリア用の端子…
反転を意味する「¯」(バー)がついているため、
「LOW」の入力でデータがクリアされる

図 5-16(b)　フリップフロップ IC

回路を作ろう！

①図 5-16(d) の接続図を参考に、図 5-16(c) の回路を組みましょう。
②D フリップフロップ IC「74HC175」の中の DFF1 を用います。
③DFF1 の入力 1D にスイッチ入力回路をつなぎます。
④1Hz の RECT 信号をクロック信号として使うため、RECT 端子とクロック CK をつなぎます。
⑤CLR 端子に 0V が入力されていると出力がクリアされ続けてしまうため、5V とつないでおきます。
⑥使用しない入力ピンはグランド（GND）もしくは 5V ラインにつないでおきましょう。

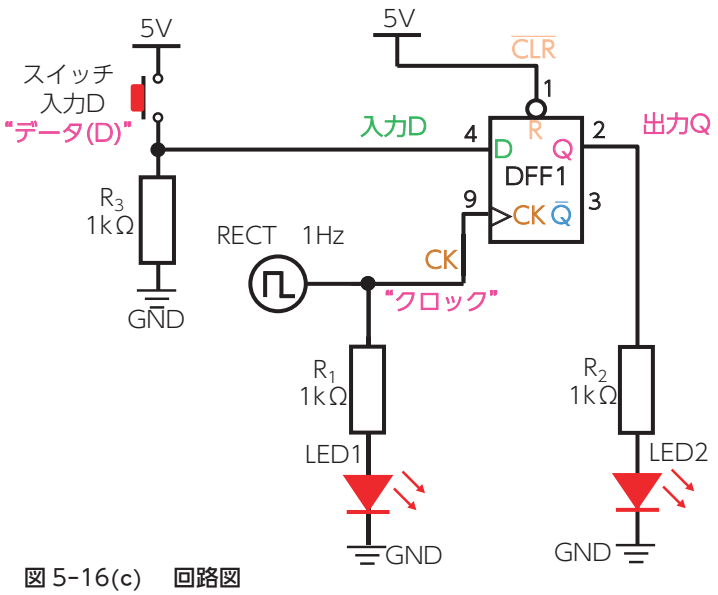


図 5-16(c)　回路図

回路を作ろう！

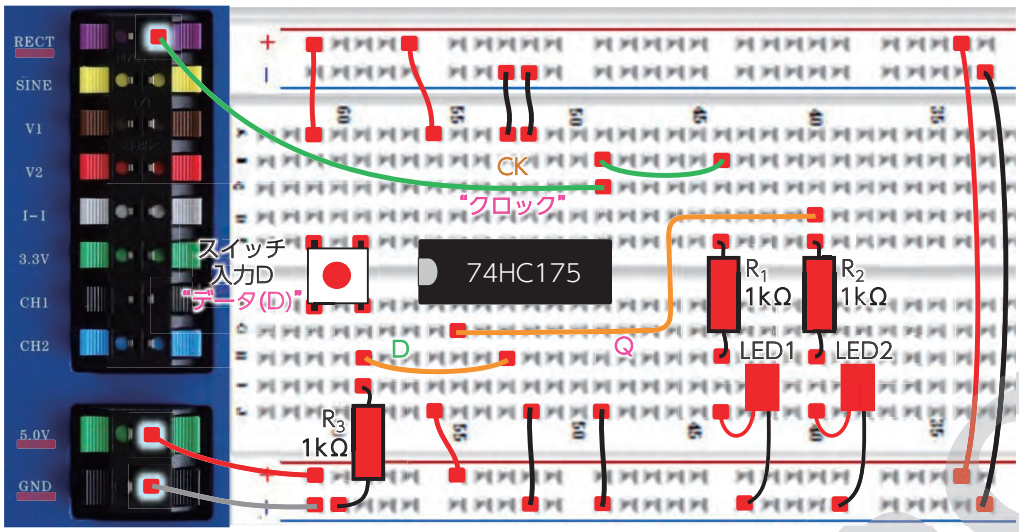


図 5-16(d)　接続図

実験のすすめ方

①矩形波(RECT)の周波数の設定を「1Hz」に設定する。
②スイッチを押し、入力 D が変化するタイミングと出力 LED2 の点灯タイミングを確認する。
③スイッチを離し、入力 D が変化するタイミングと出力 LED2 の消灯タイミングを確認する。
④同様に何度かスイッチ操作を行い、出力 LED2 の点灯 / 消灯タイミングを確認する。。

考えてみよう

①スイッチ操作のタイミングと出力 LED2 の点灯 / 消灯タイミングは毎回同じでしたか？
②出力 LED2 の状態は、クロック信号(LED1)の ON/OFF タイミングのどちらと同時に変化しましたか？

ワークシートに記入しよう！

補足

DFF とクロック

D フリップフロップはデジタル回路において様々な使われ方をする順序回路です。出力が変更されるのはクロック信号の周期がくり返すタイミングです。この「クロック」という信号が DFF の出力タイミングを決定しています。

クロックは「時計：Clock」を意味する言葉で、電子回路においては「回路が動作するタイミング(時間)を知らせる」ために使われる信号を**クロック信号**と呼びます。規則正しく動く時計の秒針やメトロノームを想像するとよいでしょう。

また、信号が「0 から 1」に変化する部分を信号の**立ち上がりエッジ**と呼び、「1 から 0」に変化する部分を**立ち下がりエッジ**と呼びます。実験で作った回路はクロック信号の立ち上がりエッジで出力が変化します。

クロック信号が順序回路の動作タイミングに使われるため、一般的にクロック信号が速ければ速いほど処理が高速になります。パソコンのカタログにある「CPU クロック」というスペックは、基本のクロック信号の周波数のことを指しています。

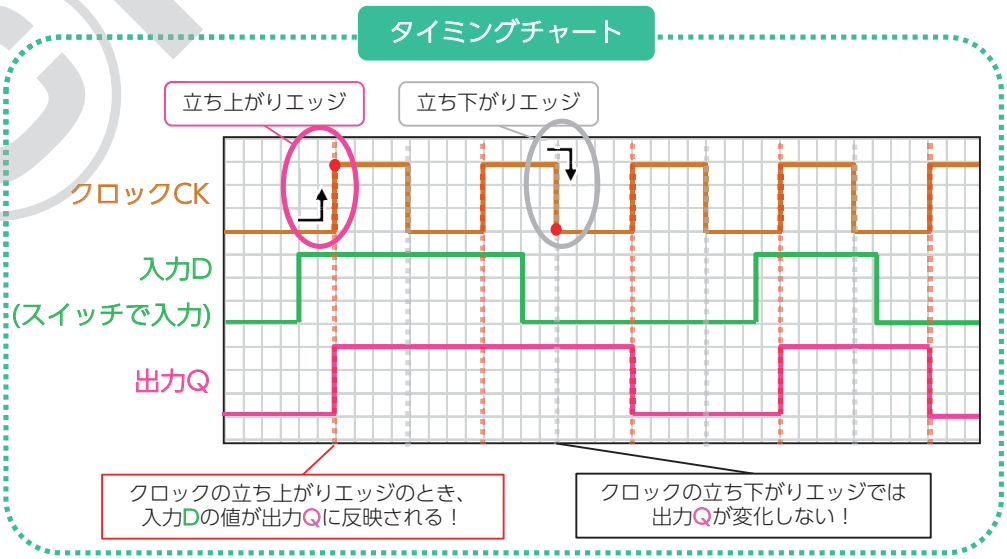


図 5-16(e)　DFF のタイミングチャート

5-20 4ビットアップダウンカウンタ

実験

学習ポイント

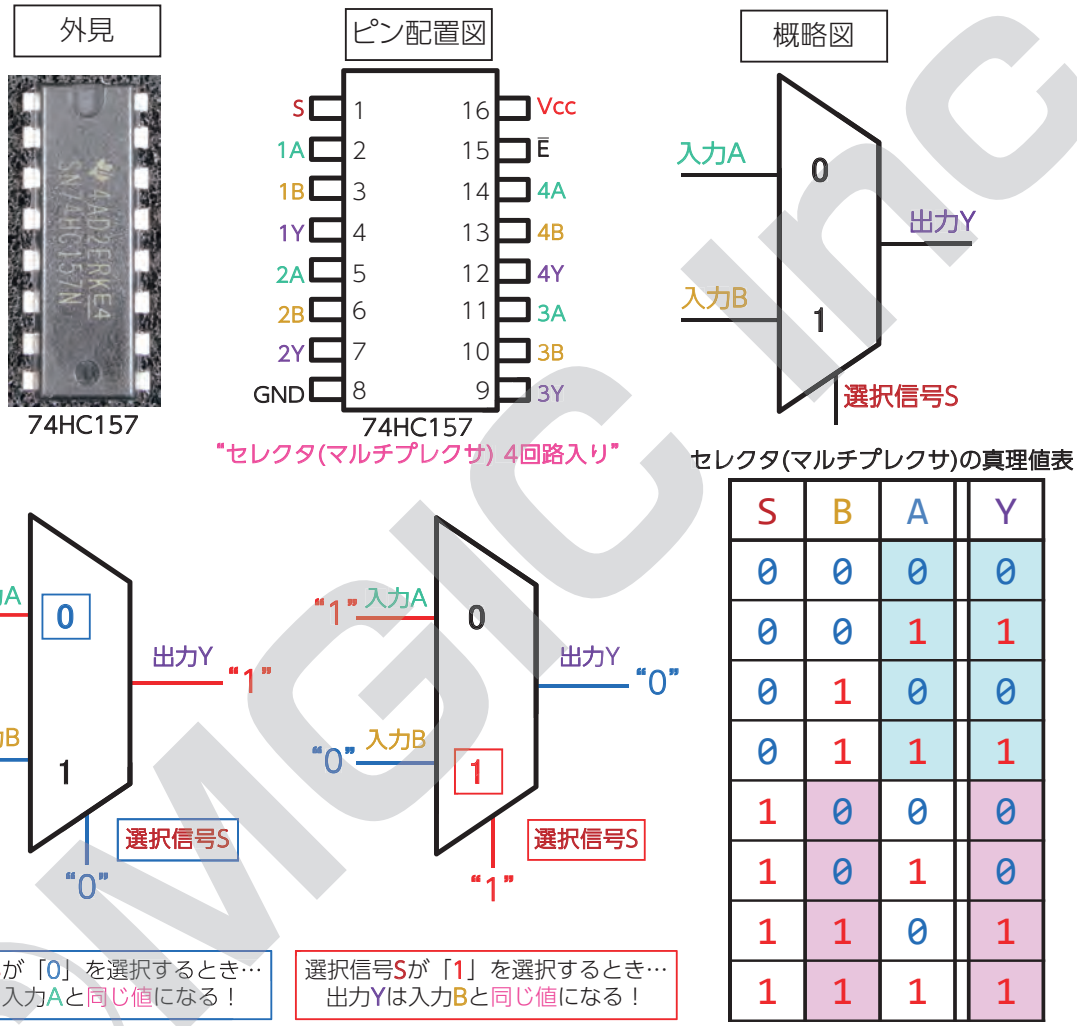
これまでに扱った論理回路を組み合わせてカウントの増加・減少を行うアップダウンカウンタを作りましょう。

チェック

今回の実験回路では、スイッチが押されていないときはカウントを増加させるアップカウンタとして、スイッチが押されているあいだは逆にカウントを減少させるダウンカウンタとして動作します。

カウントを増加させるための信号と、減少させるための信号の2種類のうちどちらかを選択して7セグデコーダへ入力するために、**セレクタ(マルチプレクサ)**と呼ばれる回路を使用します。

実験で使うセレクタIC「74HC157」は、選択信号Sの状態に応じて、**どの入力を選択するか**を決定します。
回路図中ではMUXといった略称もよく用いられ、「マルチプレクサ：Multiplexer」という言葉から由来しています。



S	B	A	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

図5-20(a)　セレクタ(マルチプレクサ) IC

回路を作ろう！

①図5-20(c)の接続図を参考に、図5-20(b)の回路を組みましょう。
②複数のロジックICを組み合わせる大規模な回路となるため、配線に気を付けましょう。
③**各ロジックICの電源も忘れずに接続し、使用しない入力ピンはグランド(GND)もしくは5Vラインにつないでおきましょう。**
④その他の端子(イネーブル $\overline{E}$ やロード $\overline{LOAD}$ など)の機能は使用しないため、回路図通りに接続しましょう。

CMOS Inc.

回路を作ろう！

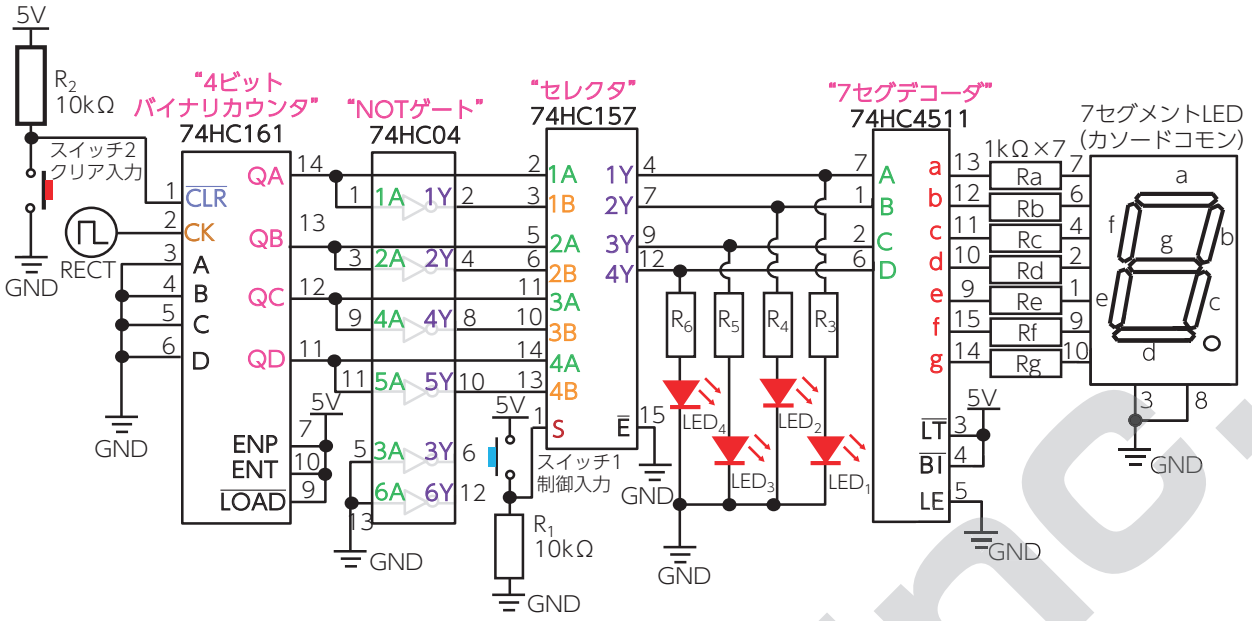


図 5-20(b)　回路図

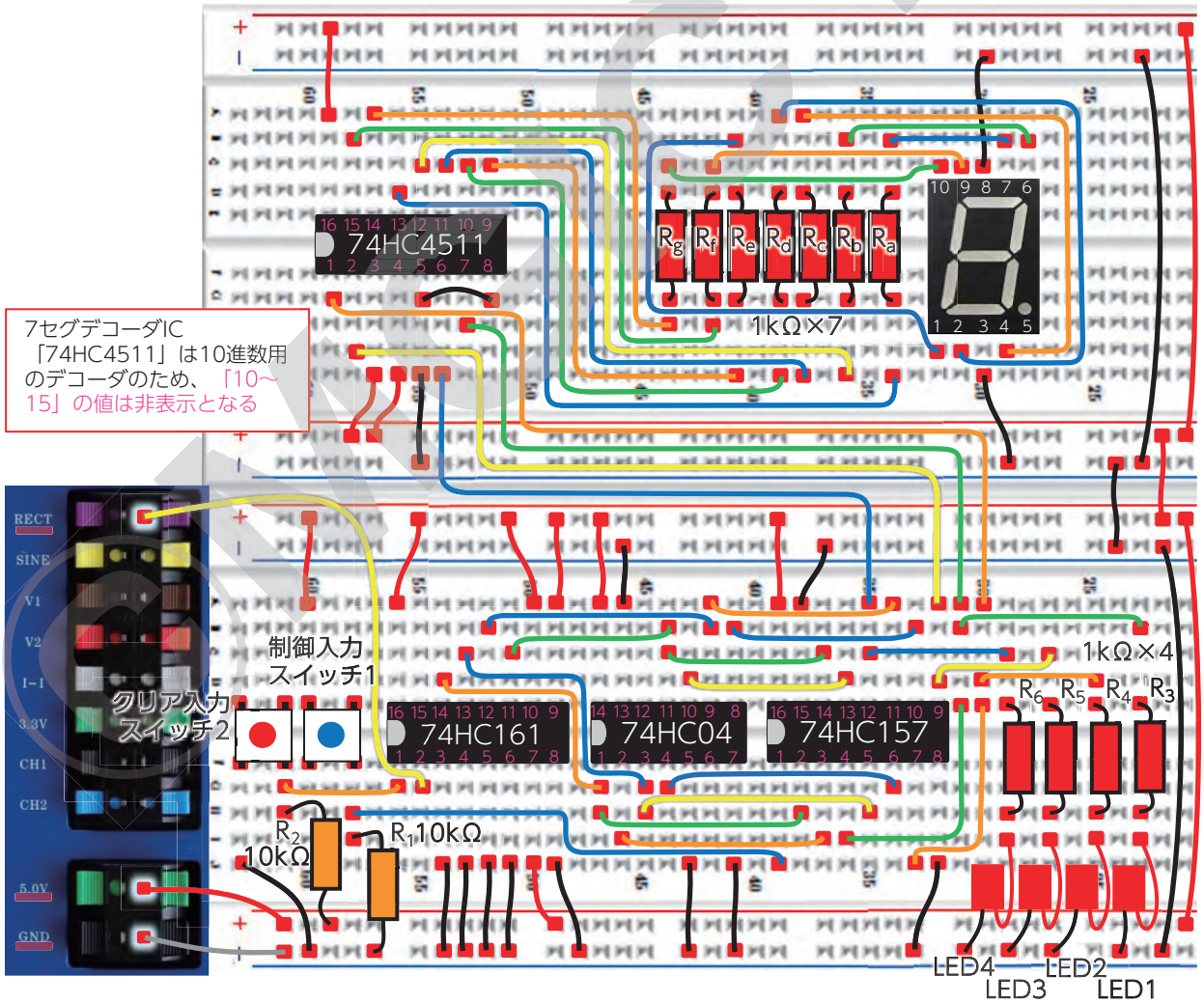


図 5-20(c)　接続図

実験のすすめ方

①クロック信号(RECT)の周波数を「2Hz」に設定し、7 セグメント LED の点灯パターンを観察する。
②スイッチ 1 を押し続けているときの 7 セグメント LED の点灯パターンを観察する。
③スイッチ 1 を離し、スイッチ 2 を押したときの表示数値の変化を確認する。

考えてみよう

①スイッチ 1 を押していないとき、7 セグメント LED の表示はどのように変化しましたか？
②スイッチ 1 を押し続けているとき、7 セグメント LED の表示はどのように変化しましたか？
③スイッチ 1 を離してからスイッチ 2 を押したとき、7 セグメント LED にはどの数字が表示されますか？

ワークシートに記入しよう！

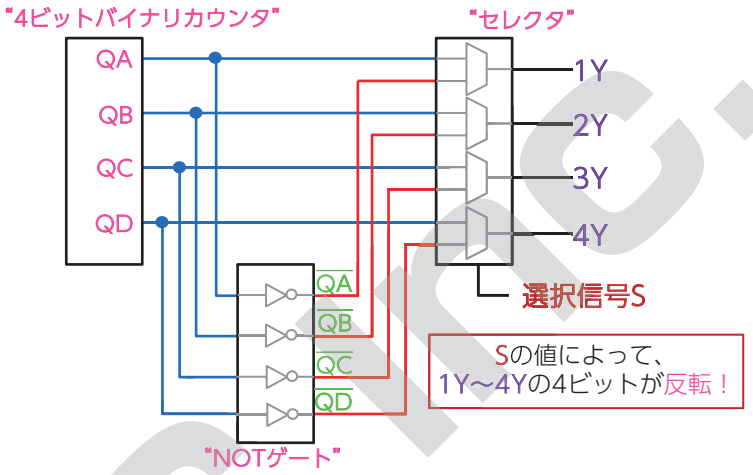
補足

アップカウントとダウンカウント

実験回路では、スイッチ 1 を押し続けているあいだは表示数値が減少していきます。

通常はバイナリカウンタの 4 ビットの値が 7 セグデコーダへ入力されますが、スイッチ 1 を押すことで選択信号が「1」に切り替わり、**4 ビットの値がすべて反転する**ことでデコーダの入力値が減少していくのです。

バイナリカウンタの**カウント動作自体が巻き戻っているわけではない**ことに着目しましょう。



S	4Y	3Y	2Y	1Y
0	0	0	0	0
	QD	QC	QB	QA

S	4Y	3Y	2Y	1Y
1	1	1	1	1
	$\overline{QD}$	$\overline{QC}$	$\overline{QB}$	$\overline{QA}$

図 5-20(d)　選択信号と出力

※ 7セグデコーダIC「74HC4511」は「0～9」までしか表示されないため、LED1～LED4の点灯/消灯を「4Y, 3Y, 2Y, 1Y」の4ビットに見立てて確認しよう！

選択信号Sが「0」のとき

4Y	3Y	2Y	1Y	10進数
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

※7セグに表示されない

アップカウントの進む方向

S=0 でスタート

S→1 選択信号切り替え

選択信号Sが「1」のとき

4Y	3Y	2Y	1Y	10進数
1	1	1	1	15
1	1	1	0	14
1	1	0	1	13
1	1	0	0	12
1	0	1	1	11
1	0	1	0	10
1	0	0	1	9
1	0	0	0	8
0	1	1	1	7
0	1	1	0	6
0	1	0	1	5
0	1	0	0	4
0	0	1	1	3
0	0	1	0	2
0	0	0	1	1
0	0	0	0	0

※7セグに表示されない

ダウンカウントの進む方向

図 5-20(e)　カウント動作

©MGIC Inc.

第6部
AD/DA変換

実験 6-1 デジタル－アナログ変換（DAC） …… 170
解説 6-2 D/A変換器と分解能 …… 176
実験 6-3 アナログ－デジタル変換（ADC） …… 177
解説 6-4 サンプリング（標本化）・量子化・符号化 …… 181

©MGIC Inc.

6-1 デジタル－アナログ変換(DAC)

実験

学習ポイント

デジタル信号をアナログ信号に変換することを**デジタル－アナログ変換（D/A 変換）**、と呼びます。**デジタル－アナログ**変換の考え方や変換回路について実験を通して学習しましょう。

チェック

D/A 変換を行う回路や装置は **D/A 変換器**、または **D/A コンバータ（DAC）**と呼ばれます。コンバータとは「変換する：Convert」という言葉から由来しています。デジタル－アナログ変換では「**デジタル値を入力して、それに対応したアナログ電圧を出力する**」という操作を行います。

例えば、現代では音楽などの音声データはほとんどがデジタルデータとして保存されています。デジタル値をそのままスピーカー回路に入力しても、曲を再生することはできません。デジタル信号を D/A 変換し、アナログ信号に変化させることではじめて音を再生することができるのです。D/A 変換器は、**入力された値をすばやく、正確にアナログ電圧で表現する**ことが求められます。

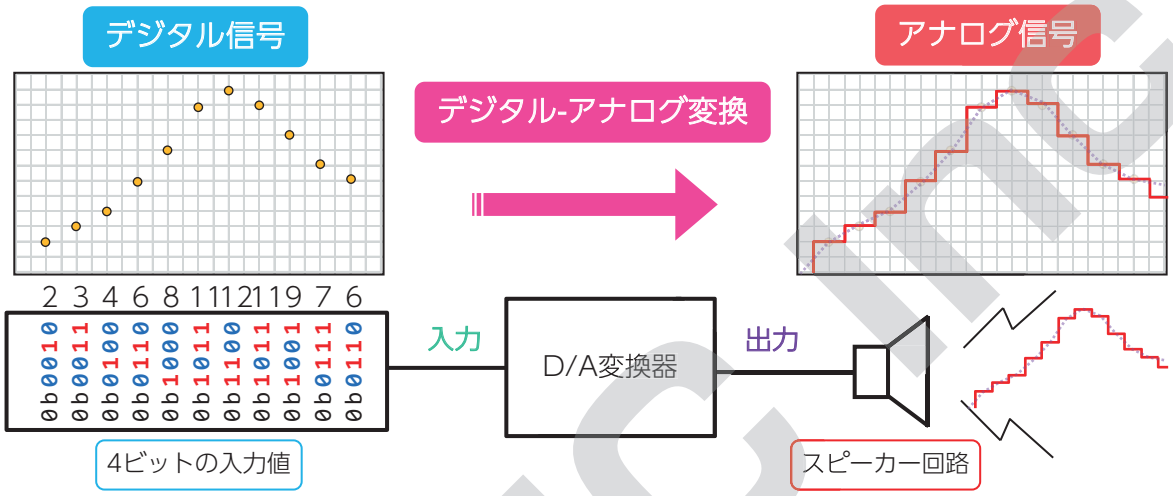


図 6-1(a) D/A 変換

4ビット(D0～D3)のデジタル値とアナログ電圧の対応

D3	D2	D1	D0	10進数	アナログ電圧
0	0	0	0	0	0.00V
0	0	0	1	1	0.3125V
0	0	1	0	2	0.625V
0	0	1	1	3	0.9375V
0	1	0	0	4	1.25V
0	1	0	1	5	1.5625V
0	1	1	0	6	1.875V
0	1	1	1	7	2.1875V
1	0	0	0	8	2.50V
1	0	0	1	9	2.8125V
1	0	1	0	10	3.125V
1	0	1	1	11	3.4375V
1	1	0	0	12	3.75V
1	1	0	1	13	4.0625V
1	1	1	0	14	4.375V
1	1	1	1	15	4.6875V

図 6-1(b) デジタル値とアナログ値

4 ビットの D/A 変換について考えてみます。4 ビットのデジタル値は 2 進数で「0b0000～0b1111」までの **16 通り**です。この **16 段階**のデジタル値で「0V 以上～ 5V 未満」の範囲のアナログ電圧を表現しようとすると、**「5 ÷ 16 ＝ 0.3125V」きざみのアナログ電圧に変換**することになります。

「0b0100」や「0b1100」といったデジタル値を D/A 変換器に入力すれば、そのデジタル値に対応した電圧「1.25V」や「3.75V」が出力されます。

D/A 変換器によく使われる **R-2R ラダー回路**を実験しましょう。難しい理論は必要ありません。**オームの法則**と**直列・並列の合成抵抗**の計算をすることで動作を理解することができます。

ヒント

2 進数の表記とビット数

2 進数の表現には「$(1100)_2$」や「$1100_{(2)}$」などもありますが、**本テキストでは 2 進数の値は先頭に「0b」を付けて表記します。**「b」は「2 進法：Binary」の略です。「0b1100」であれば、10 進数に直すと「12」です。

「**2 進数の 1 ケタ**」を「**ビット（bit）**」と呼び、4 ケタの 2 進数は「4 ビット」と表現します。

また、「**○ビット目**」という表記をよく使いますが、これは「**2 進数の何ケタ目か**」を意味しています。

本テキストでは、2 進数値の一番右端を 1 ビット目とし、左側のケタから 2 ビット目、3 ビット目…と数えていくものとします。

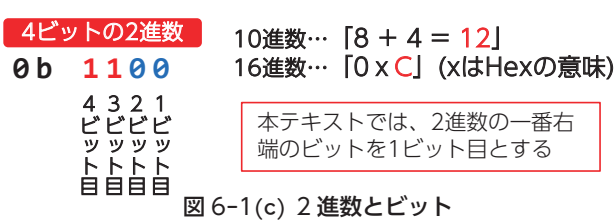


図 6-1(c) 2 進数とビット

CMC Inc.

回路を作ろう！①

①R-2R ラダー回路は、「はしご：Ladder」の名前のとおりに抵抗をはしごのように組み立てる回路です。
②図 6-1(e) の接続図を参考に、図 6-1(d) の回路を組みましょう。
③今回のラダー回路に使う抵抗は R（10kΩ）とその 2 倍の抵抗値の 2R（20kΩ）です。R と 2R を組み合わせて作ります。
④ラダー回路の終端（$2R_7$ と $2R_8$）だけは R-2R ではなく 2R を 2 個並列につなぎます。
⑤2 か所以上の電圧を測定して実験を行います。電圧計 V1・V2 は V_0 ～ V_3 の測定時につなぎかえます。

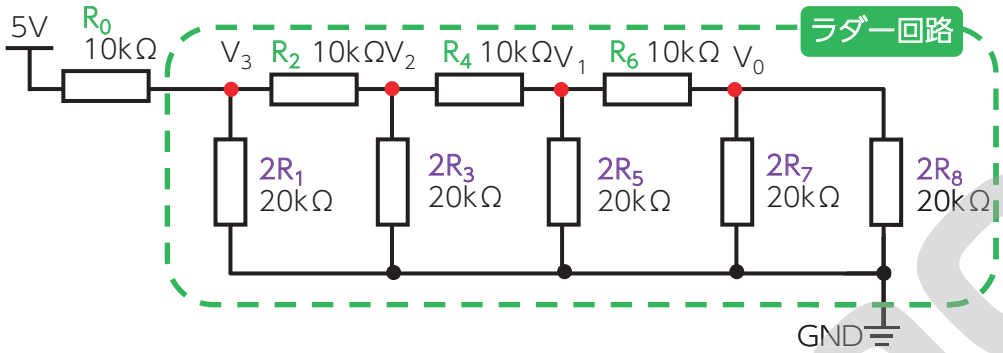


図 6-1(d) 回路図

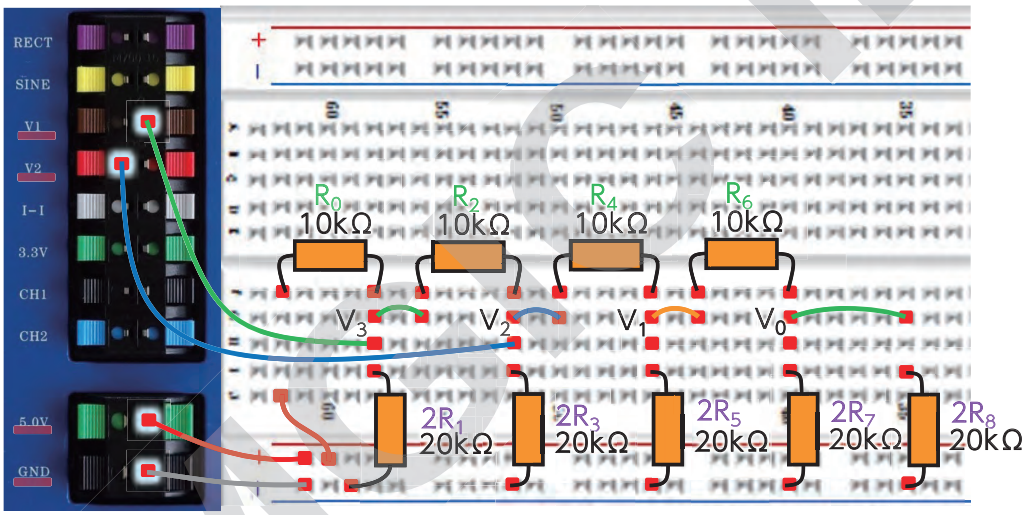


図 6-1(e) 接続図

実験のすすめ方

①回路図上の点 V_0 ～ V_3 の電圧を測定し記録する。

ワークシートに記入しよう！

考えてみよう

①点 V_0 ～ V_3 の電圧にはどのような関係がありましたか？
②$2R_7$ と $2R_8$ の並列合成抵抗「R_{PA}」を計算しよう。「P」は「並列：Parallel」、「S」は「直列：Series」を意味します。
③計算した合成抵抗 R_{PA} と R_6 の直列合成抵抗「R_{SB}」を計算しよう。
④計算した合成抵抗 R_{SB} と $2R_5$ の並列合成抵抗「R_{PB}」を計算しよう。
⑤計算した合成抵抗 R_{PB} と R_4 の直列合成抵抗「R_{SC}」を計算しよう。
⑥計算した合成抵抗 R_{SC} と $2R_3$ の並列合成抵抗「R_{PC}」を計算しよう。
⑦計算した合成抵抗 R_{PC} と R_2 の直列合成抵抗「R_{SD}」を計算しよう。
⑧計算した合成抵抗 R_{SD} と $2R_1$ の並列合成抵抗「R_{PD}」を計算しよう。
⑨R_{PD} は $2R_1$ ～ $2R_8$ までのラダー抵抗をまとめた合成抵抗です。R_0 と R_{PD} の値から電圧 V_3 を計算しよう。

ワークシートに記入しよう！

R-2R ノード電圧

電子回路において、配線同士がつながる接続点「●」を**ノード(結び目：Node)**と呼びます。

実験を行うと、ノード V_3 からノード V_0 までの電圧が 2 分の 1 ずつ低下していることがわかります。R-2R ラダー回路は、**ラダー抵抗が増えるごとに次のノード電圧が 2 分の 1 になっていきます**。これこそが R-2R ラダー回路の特性です。なぜこのような現象が起こるのかは、**R-2R ラダー回路の終端から合成抵抗をつくってまとめていく**と簡単に解明することができます。ラダー回路のはしご部分の終端に着目してみましょう。

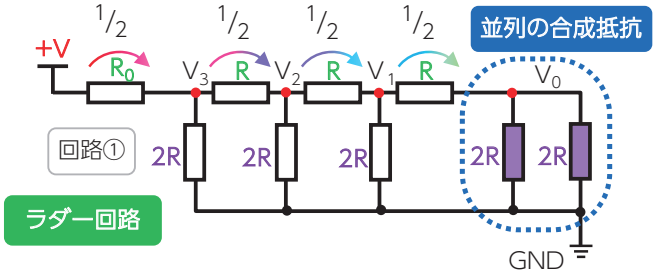


図 6-1(f) ラダー回路

電源電圧 +V、ある抵抗 R とその 2 倍の抵抗値の 2R でラダー回路を作ったときを考えます。終端の 2R と 2R で並列合成抵抗を作ると、**図 6-1(g)** のようにまとまります。並列の合成抵抗 R_{PA} は、

$$\frac{1}{R_{PA}}=\frac{1}{2R}+\frac{1}{2R}\text{ より、}R_{PA}=\frac{2R\times 2R}{2R+2R}=\frac{4R^2}{4R}=R\text{ です。}$$

つまり、2R と 2R の並列合成抵抗を作ると R になります。**同じ値の 2 個の並列抵抗は、合成すると半分の抵抗値**になることを覚えておきましょう。このような回路の変形を**図 6-1(h)** のように続けていきます。

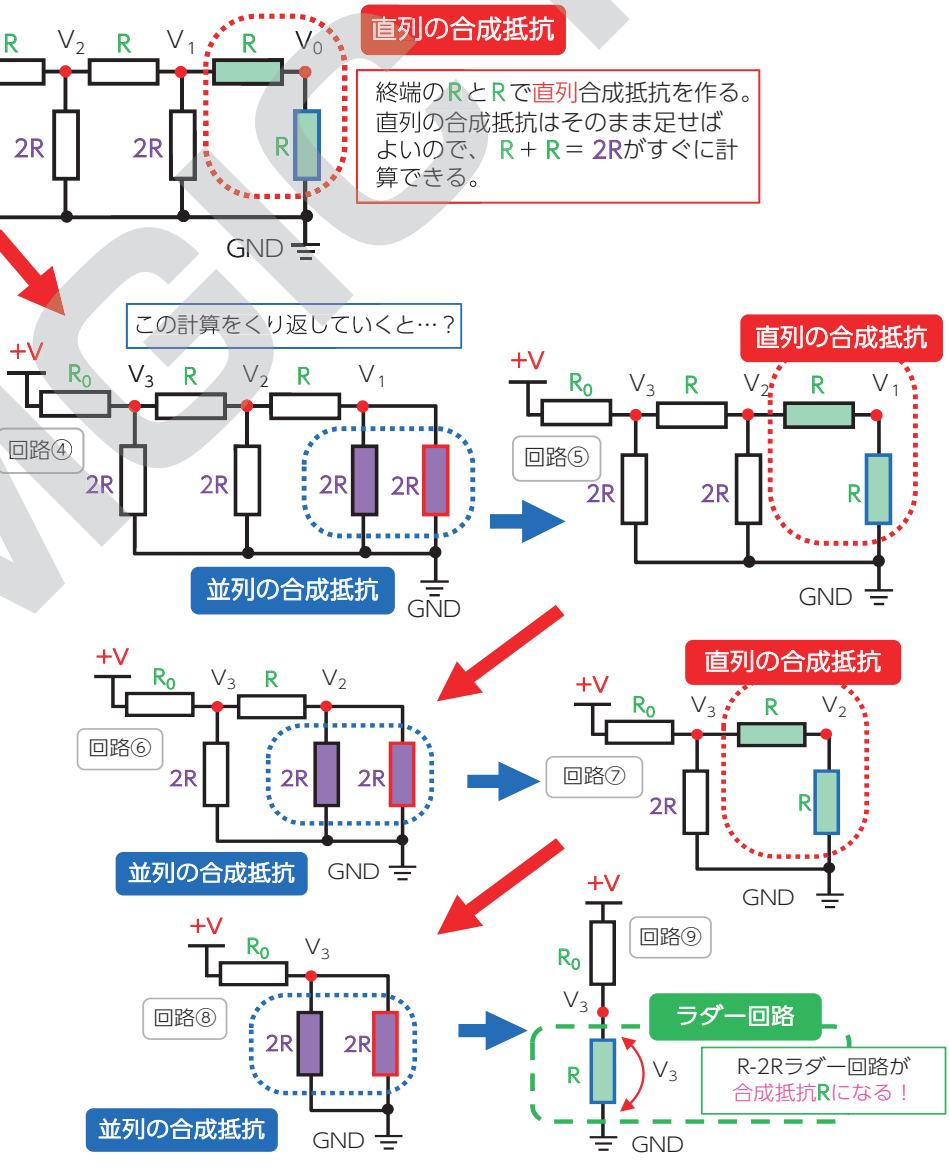


図 6-1(g) 合成抵抗でまとめる

まるではしごをたたむように、**R-2R ラダー回路を 1 個の合成抵抗 R としてまとめる**ことができます。

電圧 V_3 が電源電圧の 2 分の 1 になったのは、同じ値の抵抗 R_0 と合成抵抗 R で分圧されているからです。

同様に、回路⑦を見ると電圧 V_2 は電圧 V_3 の半分であることがわかります。はしごがどれだけ続いても、このくり返しで求めることができます。R-2R ラダー回路は、とてもよくできた電子回路です。

図 6-1(h) ラダー回路をまとめる

©Gijo Inc.

チェック

次に、R-2R ラダー回路を使った **R-2R ラダー方式 D/A 変換回路**を作りましょう。

D/A 変換回路に「0」か「1」のデジタル値を入力するために、DIP スイッチと NOT ゲート IC「74HC04」を使用します。NOT ゲートについては解説 [5-2]「論理回路」を、標準ロジック IC についての詳細は実験 [5-9]「標準ロジック IC を使う」を参照してください。

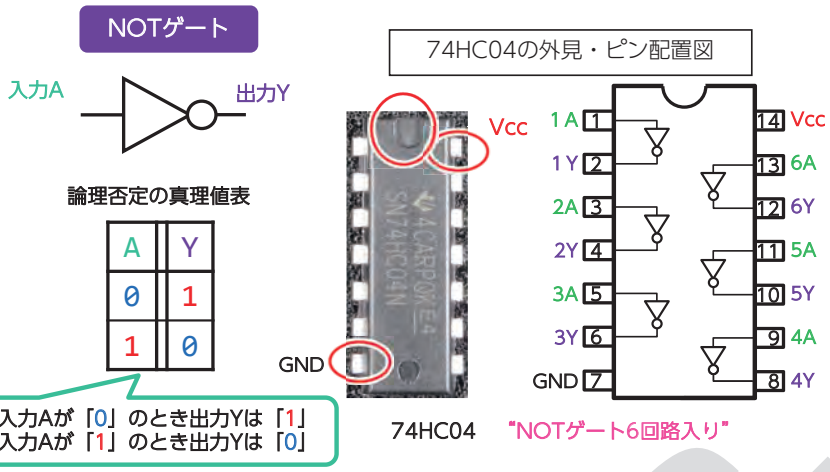


図 6-1(i) NOT ゲートとロジック IC

回路を作ろう！②

①図 6-1(k) の接続図を参考に、図 6-1(j) の回路を組みましょう。
②DIP スイッチで NOT ゲートの出力を「0V」もしくは「5V」に変化させ、4 ビットのデジタル値「D0 ～ D3」を入力します。
D/A 変換された電圧は点 V_{OUT} に出力されます。
③**使用しない入力ピン（3A・4A）はグランド（GND）ラインにつないでおきましょう。**

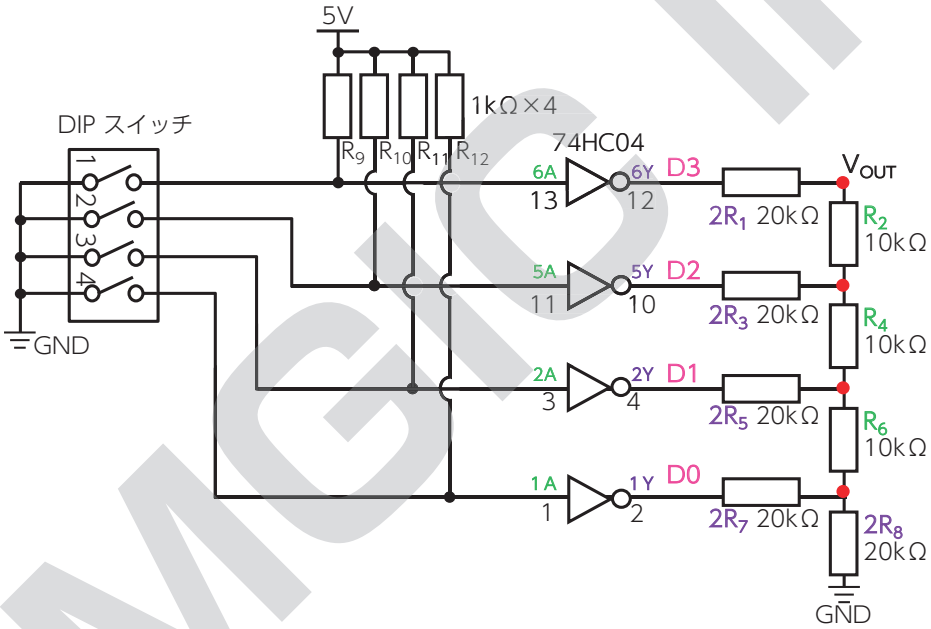


図 6-1(j) 回路図

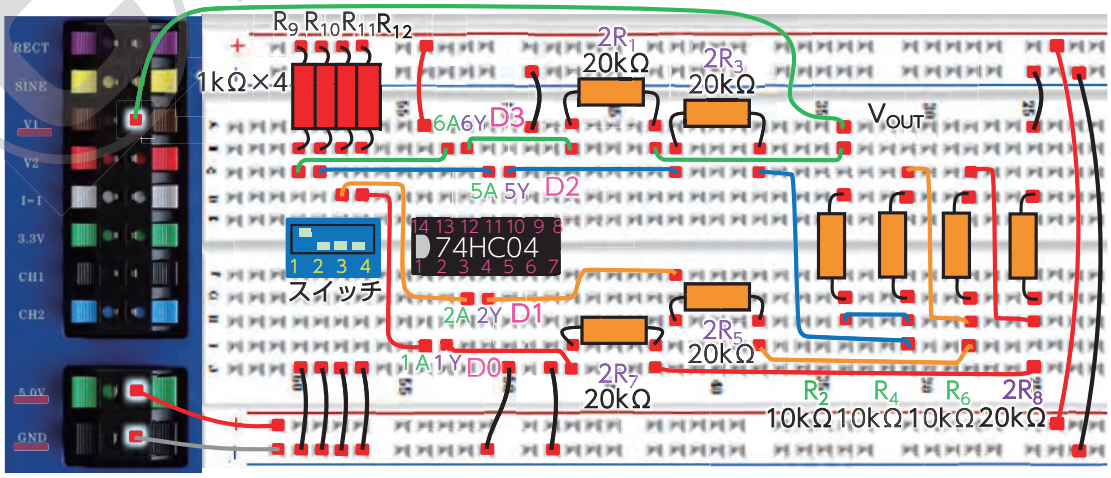


図 6-1(k) 接続図

実験のすすめ方

①DIP スイッチのレバーが下側のときを「0」（切：OFF）、上側の時を「1」（入：ON）とする。
②DIP スイッチのレバーの状態を、左から「0000」の状態にしたときの電圧 V_{OUT} を測定し記録する。
③同様に、ワークシートの表にそって DIP スイッチのレバーの状態を変化させ、電圧 V_{OUT} を測定し記録する。

ワークシートに記入しよう！

考えてみよう

DIP スイッチで入力する 4 ビットの値「D3, D2, D1, D0」と出力電圧 V_{OUT} について考えます。
①1 ビット目の値「D0」が「0 → 1」に変化すると、電圧 V_{OUT} は約何ボルト上昇しましたか？
②2 ビット目の値「D1」が「0 → 1」に変化すると、電圧 V_{OUT} は約何ボルト上昇しましたか？
③3 ビット目の値「D2」が「0 → 1」に変化すると、電圧 V_{OUT} は約何ボルト上昇しましたか？
④4 ビット目の値「D3」が「0 → 1」に変化すると、電圧 V_{OUT} は約何ボルト上昇しましたか？

ワークシートに記入しよう！

ヒント　プルアップ・プルダウン

スイッチは ON のときに両端がつながりますが、OFF の時は切り離された状態になります。このとき、切り離されたノード（接続点）を「5V」のような高い電圧までつり上げることを「プルアップ（Pull Up）」と呼びます。反対に、ノードを「0V」のような低い電圧まで落とすことを「プルダウン（Pull Down）」と呼びます。

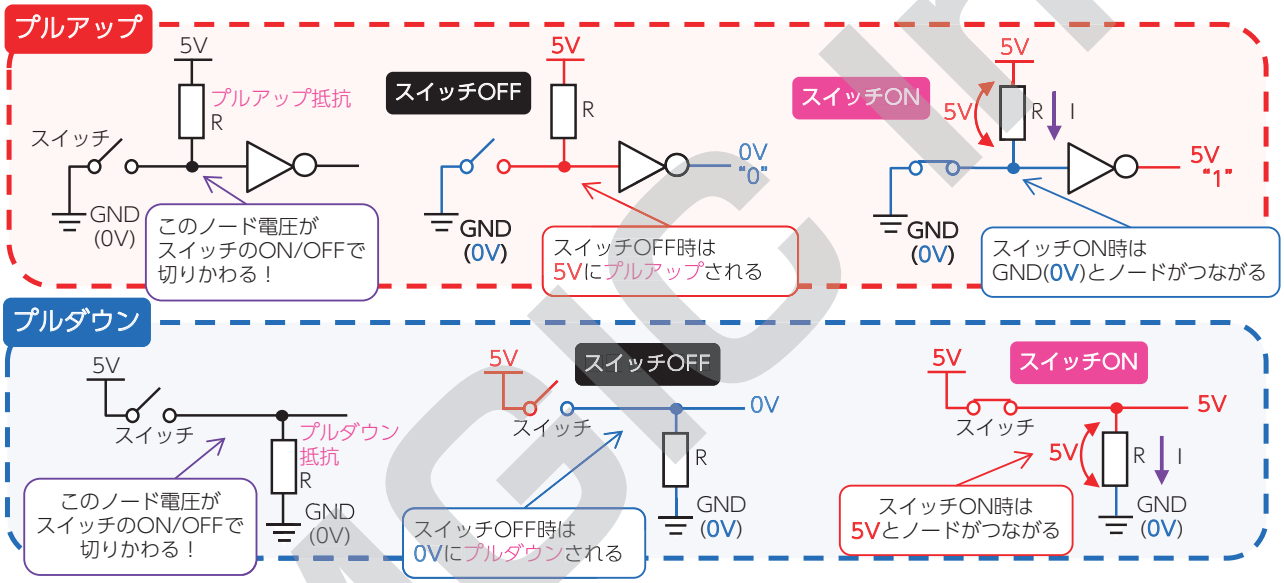


図 6-1(l)　プルアップ・プルダウン

補足　R-2R ラダー方式 D/A コンバータ

今回の実験では、DIP スイッチを操作して「D3, D2, D1, D0」の 4 ビットの値を入力しました。出力電圧 V_{OUT} は、「D0 ～ D3」の値が「0 → 1」に変化したとき、それぞれ対応する位置の R-2R ノード電圧と同じだけ上昇します。

この方式の D/A コンバータでは、入力されたデジタル値に応じてそれぞれ 1/2、1/4、1/8…といった**電圧の重みづけを出力電圧に加えることで、アナログ電圧を表現しています**。このため、厳密にはアナログ値とデジタル値には誤差が発生し、これを**量子化誤差**と呼びます。量子化誤差については解説 [6-4]「サンプリング（標本化）・量子化・符号化」を参照してください。

例えば、「D3」と「D2」のスイッチを ON にすると、
V_{OUT} が「2.5V + 1.25V = 3.75V」上昇します。
R-2R ラダー回路の出力電圧は、**合成抵抗**と**分圧回路**の計算をすることで解析することができます。

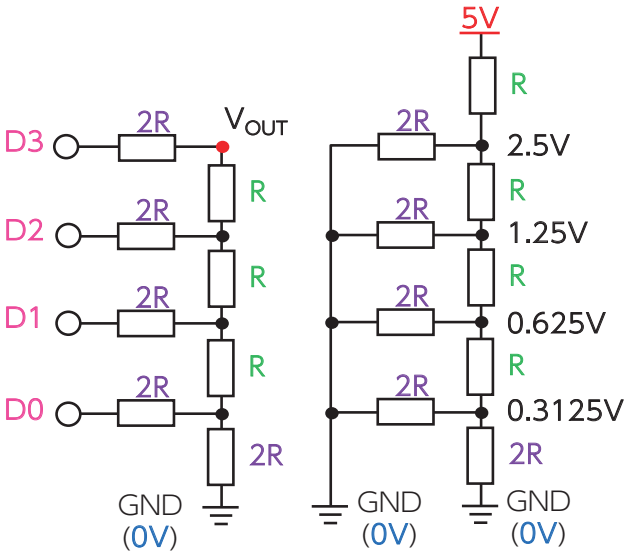


図 6-1(m) R-2R ノード電圧

©MBC Inc.

■「0b1000」入力時の出力電圧

「D3」のスイッチだけを ON にしたときの入力は「0b1000」になります。

このときの D/A 変換回路の状態は図 6-1(n) のようになり、緑枠で囲った部分を R-2R ラダー回路としてひとまとめにできるため、分圧で V_{OUT} が 2.5V になることがわかります。

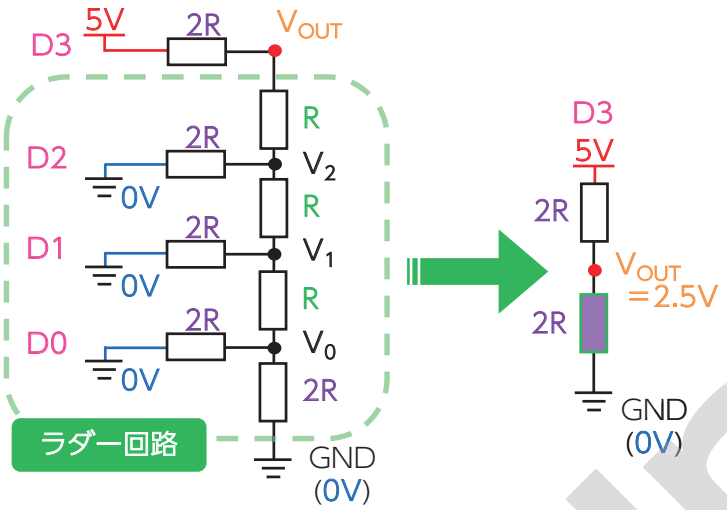


図 6-1(n)「1000」入力時

■「0b1100」入力時の出力電圧

「D3」と「D2」のスイッチを ON にしたときの入力は「0b1100」になります。

このときの D/A 変換回路の状態は図 6-1(o) のようになります。同様に緑枠で囲った部分を R-2R ラダー回路としてひとまとめにしたあと、青枠で囲った部分の合成抵抗を求めると、V_2 が計算できます。

V_2 がわかれば、最後に V_{OUT} の電圧を分圧回路として計算するだけです。

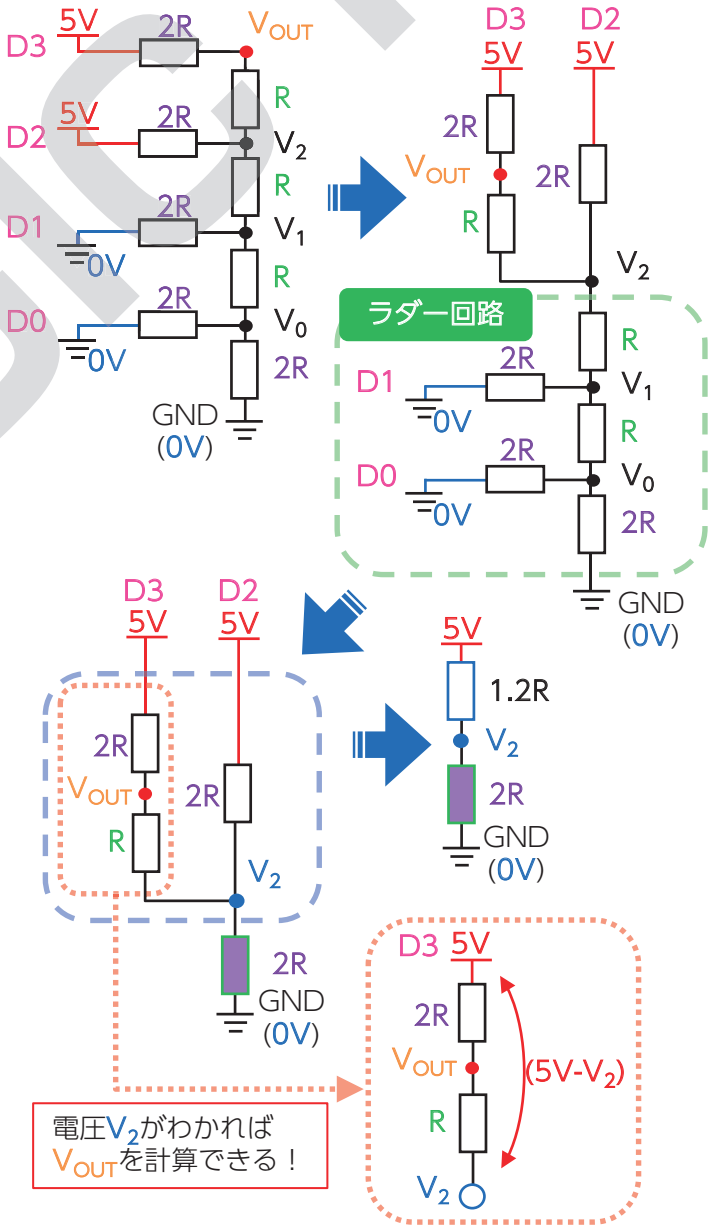


図 6-1(o)「1100」入力時

©MCIC Inc.

第7部
演算装置

解説 7-1 コンピュータ(計算機) …… 184
実験 7-2 ハーフアダー回路 …… 185
実験 7-3 フルアダー回路 …… 188
実験 7-4 4ビット加算器 …… 191
コラム ロボット …… 194
実験 7-5 レジスタと加算器 …… 195

©MGIC Inc.

7-1 コンピュータ（計算機）

解説

「**コンピュータ：Computer**」は「**計算機**」を意味します。私たちが日常で使っているパソコン（PC）やスマートフォンといった端末もまたコンピュータの一種です。コンピュータは通信や事務処理、画像や音楽の再生・保存など、さまざまな目的で利用されており、巨大な計算機であるスーパーコンピュータや小型で低コストな特定用途向けの**マイクロコントローラ（マイコン）**と呼ばれるものもあります。

では、**コンピュータ（Computer）と電卓（Calculator）**にはどのような違いがあるでしょうか？

コンピュータ（計算機）も電卓も、計算を行うという意味ではよく似たツールであり、電卓（電子式卓上計算機）もまたシンプルなコンピュータとしての側面があります。大きな違いがあるとすれば「**コンピュータは設定した計算式の通りに自動で計算する**」という点です。

例えば電卓を使った計算を求められたとき、人間はそのつど計算式どおりの入力を行います。数回程度の手順であれば簡単に実行できるでしょう。しかし、電卓での計算を高速に**数千回、数万回**以上くり返す必要があればどうでしょうか…？これは、ほとんどの人間には不向きな作業です。一方で、コンピュータ（計算機）は**計算式や計算の順番**をあらかじめ教えてあげれば、あとは計算に必要なデータ（値）を入力するだけで処理を続けてくれるのです。これは電卓にはない特別な機能です。

コンピュータ（計算機）に実行してもらう計算式や手順などの**命令**を**プログラム**と呼びます。また、パソコンなどの機器類を**ハードウェア**、ハードウェアを動かすものを一般に**ソフトウェア**と呼び、コンピュータに実行させるプログラム（命令）はソフトウェアと同じ意味の言葉として使われています。

図 7-1(a) 計算機と電卓

コンピュータの 5 大装置

コンピュータ（計算機）は大きく分けて 5 つの機能で構成されており、これを**コンピュータの 5 大装置**と呼びます。

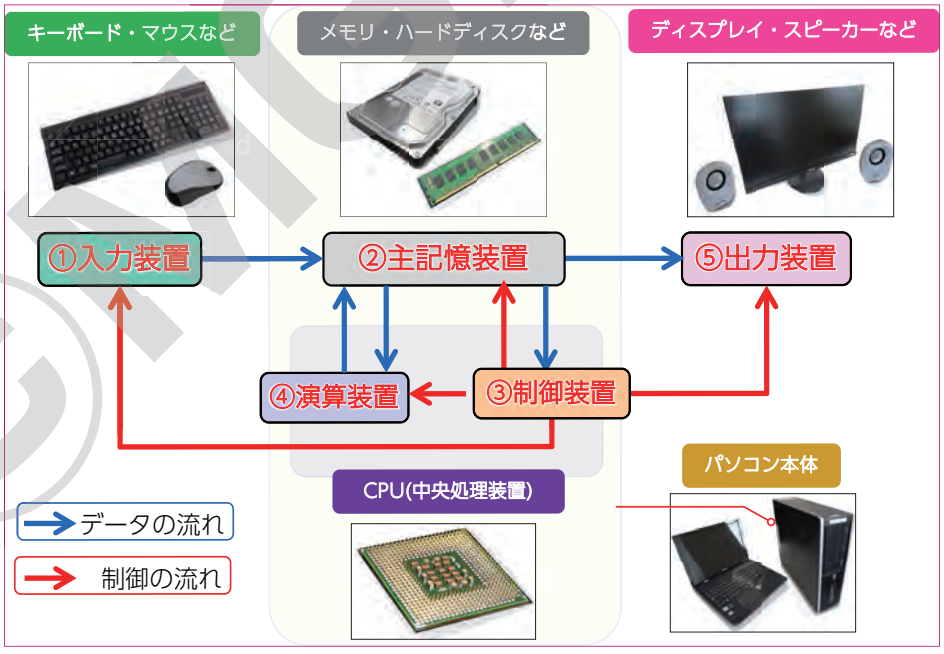


図 7-1(b) 5 大装置

制御装置と演算装置をあわせて CPU（**中央処理装置：Central Processing Unit**）と呼びます。

第 7 部の実験では CPU の演算機能に着目し、ブレッドボード上に簡易的な演算装置を作ることを目標に学習を進めます。まず最初に、基本的な演算である**足し算**を行う回路について実験を行いましょう。

7-2 ハーフアダー回路

実験

学習ポイント

CPU 内部の演算装置は、足し算や引き算などの演算を 2 進数で処理します。足し算を行うための回路を**アダー（加算器：Adder）**と呼びます。足し算回路を作り、演算回路についての学習を始めましょう。

チェック

まず最初に**ハーフアダー（HalfAdder）**と呼ばれる加算器を作って実験します。「半分の：Half」という言葉のとおり、**ハーフアダー（HA）**は**半加算器**とも呼ばれています。図 7-2(a) のように入力 A と B の足し算を行い、出力 S で和の 1 ケタ目を示します。また、ケタが上がるとキャリー信号 C が「0 → 1」に変化します。

2 進数の足し算

1 ビットの入力 A と B の値を 2 進数で足し算するとき、答え（和）は「0b0, 0b1, 0b10」のいずれかです。「0b0」「0b1」は 1 ケタで表記できますが、「2」の値は「0b10」となり、**2 進数値が 1 ケタ（1 ビット）増える**ことになります。このケタ上がりのことを**キャリー（Carry）**と呼びます。

1 ビットの入力 A と B の足し算を行うときには、**和の 1 ケタ目**をあらわす出力 S と 2 ケタ目へのキャリーをあらわす出力 C が必要になります。S は「**和：Sum**」、C は「ケタ上げ：Carry」の頭文字です。

ハーフアダーがなぜ「半」加算器と呼ばれているのか、まずは加算器の実験を行い、学習を進めたうえで解説を行います。

1ケタ(1ビット)の2進数の足し算

A	+	B	=	C	S	10進数
0	+	0	=		0	0
1	+	0	=		1	1
0	+	1	=		1	1
1	+	1	=	1	0	2

入力AとBがどちらも「1」のときにケタ(ビット数)が上がる！ケタ上げ(キャリー)信号Cが必要になる！

図 7-2(a) 2 進数の足し算

半加算器の真理値表

B	A	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

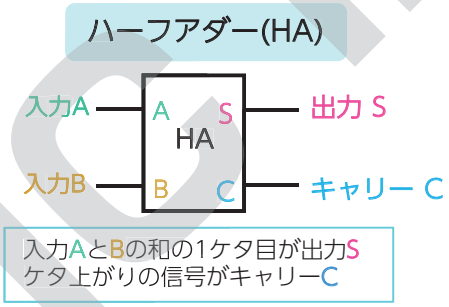


図 7-2(b) ハーフアダー (HA)

ヒント

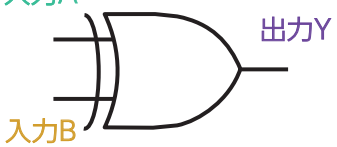
実験で使用するロジック IC

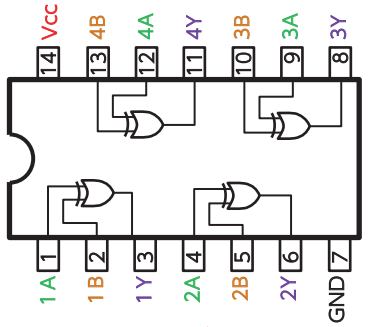
今回の実験では、ハーフアダー（HA）を論理ゲートの組み合わせで作ります。
NAND ゲートや XOR ゲートについては解説 [5-2]「論理回路」、解説 [5-6]「さまざまな論理回路」を参照してください。

排他的論理和の真理値表

XORゲート

B	A	Y
0	0	0
0	1	1
1	0	1
1	1	0



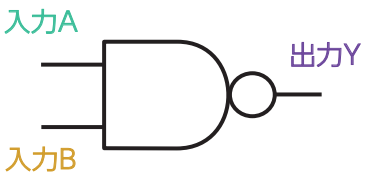


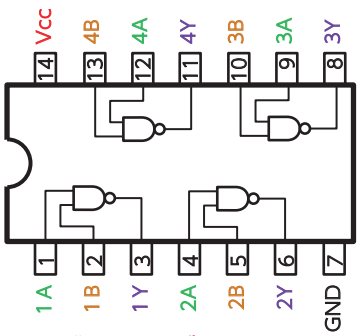
74HC86 “XORゲート4回路入り”

否定論理積の真理値表

NANDゲート

B	A	Y
0	0	1
0	1	1
1	0	1
1	1	0





74HC00 “NANDゲート4回路入り”

図 7-2(c) XOR ゲートと NAND ゲート

回路を作ろう！

①図 7-2(e) の接続図を参考に、図 7-2(d) の回路を組みましょう。
②NAND ゲート IC「74HC00」と XOR ゲート IC「74HC86」を使用します。各 IC のピン配置などは図 7-2(c) のヒントを参考にしてください。
③入力 A と B の XOR をとることで、和の 1 ケタ目の値 S を出力します。
④入力 A と B の AND をとることで、キャリー（ケタ上げ）の値 C を出力します。
⑤回路では NAND ゲートを使って NOT ゲートを作り、NAND 出力につなぐことで AND ゲートを作っています。これは信号を反転させる際によく用いられる NAND ゲートの使い方のひとつです。
⑥**使用しない入力ピンはグランド（GND）もしくは 5V ラインにつないでおきましょう。**

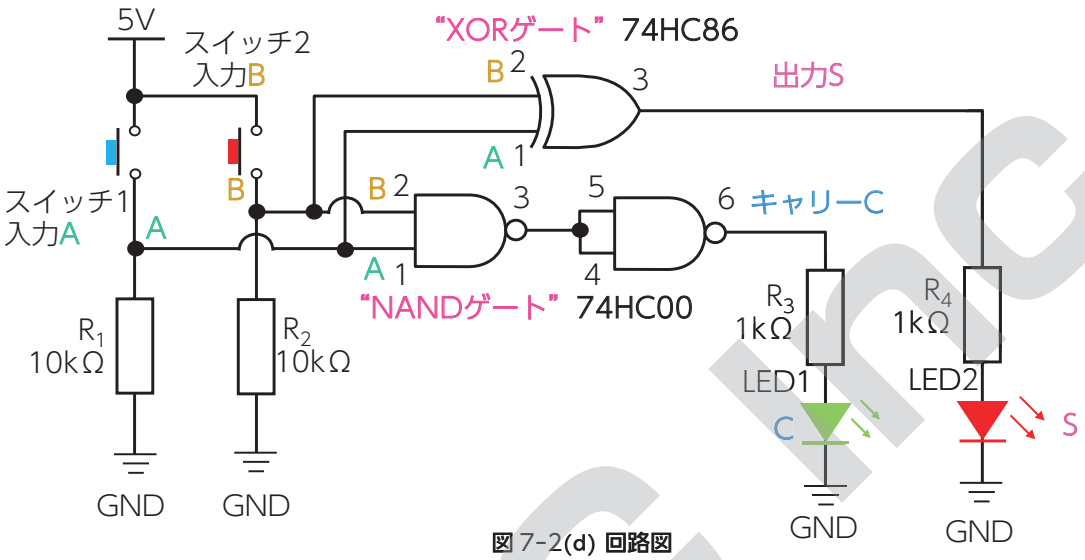


図 7-2(d) 回路図

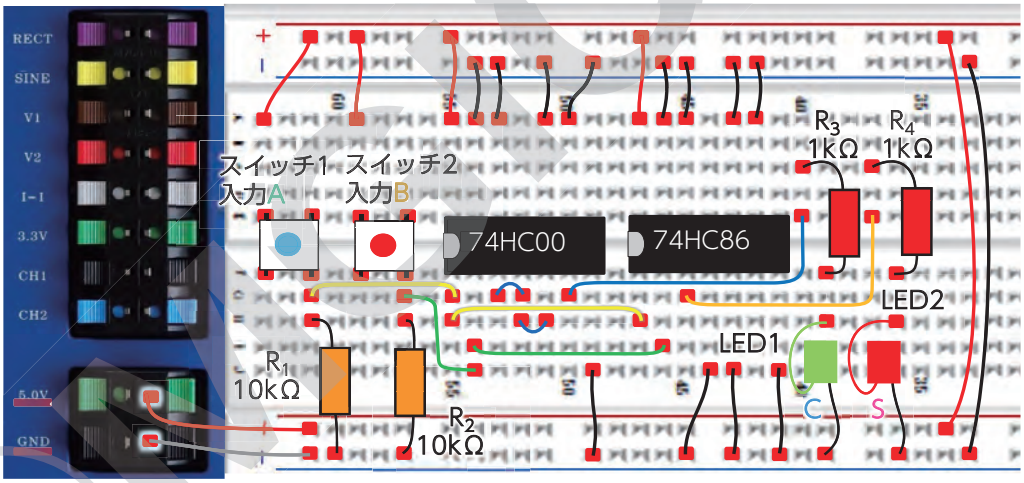


図 7-2(e) 接続図

実験のすすめ方

①スイッチをどちらも押していないときに両方の LED が消灯していることを確認する。
②スイッチ 1 だけを押しているときの各 LED のようすを観察する。
③スイッチ 2 だけを押しているときの各 LED のようすを観察する。
④両方のスイッチを押しているときの各 LED のようすを観察する。

考えてみよう

①スイッチを押しているときを「1」、押していないときを「0」とします。また、LED が点灯しているときを「1」、消灯しているときを「0」とします。
②スイッチ入力と出力 S、キャリー信号 C の関係をワークシートの真理値表に記入しよう。

ワークシートに記入しよう！

©CQ Inc.

ハーフアダーとキャリー信号

補足

足し算(加算)を行い、加算結果のケタが上がるときはキャリー信号を「1」にして次のケタを計算する加算器に知らせます。このような動作が**アダー(加算器)**の基本的な機能です。

半加算器の真理値表

B	A	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

真理値表がAとBのXORと同じ！

真理値表がAとBのANDと同じ！

ハーフアダー(HA)の論理回路図

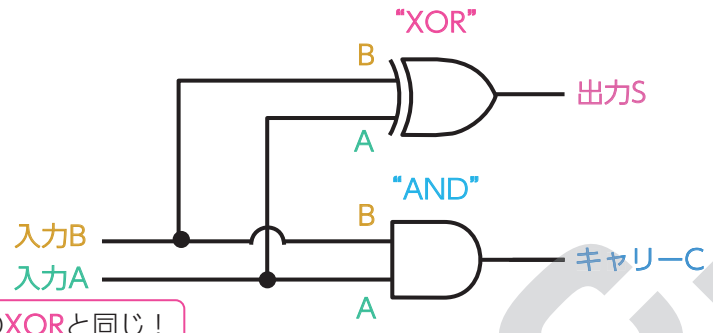


図 7-2(f) ハーフアダー(HA)

ハーフアダー(HA)がなぜ図 7-2(f) のような論理回路になるのかは、**真理値表を見る**と理解しやすくなります。

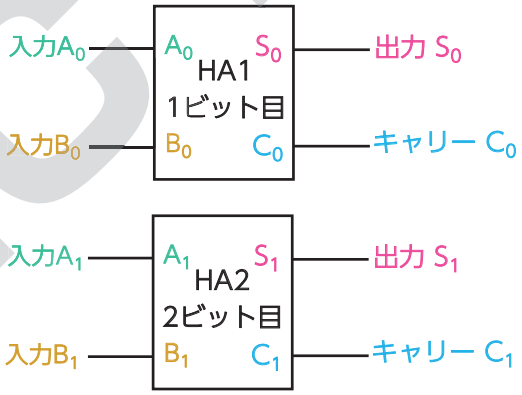
それぞれの入力と出力の関係に着目してみれば、**足し算の和は XOR 演算**で、**キャリーは AND 演算**で出力が決まることがわかります。

ハーフアダー(HA)は**前のビットのキャリーを扱わない**加算器であり、入力が A と B の 2 本しか**ありません**。

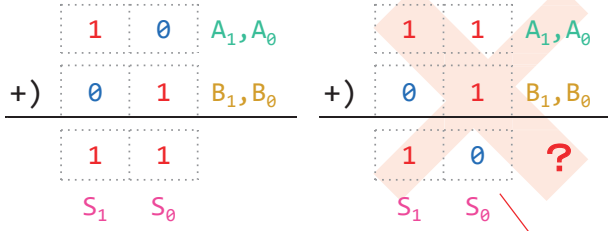
これは、**ハーフアダー 1 個だけでは 2 ビット目以降の足し算ができない**ことを意味します。

図 7-2(g) のように HA2 を 2 ビット目に使っても、HA2 は前のビットの HA1 のキャリー信号を受け取らないためケタ上がりの計算がうまくできません。

ハーフアダー(HA)2個で
2ビットの足し算をすると…？



HA1のキャリー信号がHA2で使われないため、
ケタ上がりの計算がおかしくなってしまう！



ケタ上がりのない加算は
問題なさそうに見えるが…

正解は「0b100」

図 7-2(g) ハーフアダー(HA)の特性

©MGIC Inc.

第8部
組み込みプログラミング入門

はじめに 8-1 組み込みプログラミング入門 …… 8-2
実験 8-2 LEDを点滅させよう …… 8-6
解説 8-3 変数と配列 …… 8-12
実験 8-4 ポーリングと割り込み …… 8-15
解説 8-5 反復処理と分岐処理 …… 8-22
解説 8-6 関数の引数と戻り値 …… 8-24
解説 8-7 マイコンと組み込みシステム …… 8-25

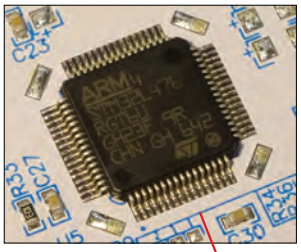
©MGIC Inc.

8-1　組み込みプログラミング入門

はじめに

組み込みプログラミングとは？

特定の機器に必要な機能をプログラムして機器本体のコンピュータに組み込むことを指します。このような技術体系は組み込み技術（Embedded Technology）とも呼ばれ、プログラムを組み込む先のコンピュータとして、**マイコン（MCU：Micro Controller Unit）**と呼ばれるICチップがしばしば用いられます。



組み込み

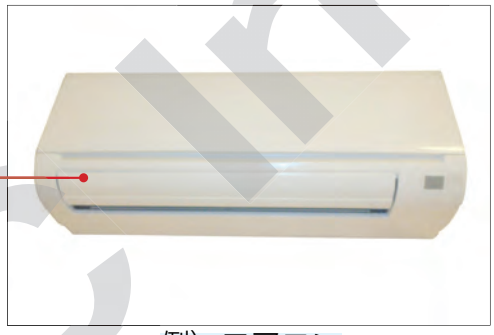
マイコン（マイクロ・コントローラ・ユニット）

コンピュータに必要とされる5大装置のうち、CPU（制御・演算）やメモリ（記憶）などが一つのチップに集約されている。

MCUとも呼ばれる。

エアコンに必要な制御プログラム

- 送風
  - ファン・風向板などの駆動制御
- 室温の測定
  - センサー信号の判定
- タイマー
  - 指定時間での動作切り替え
- リモコン信号の受信
  - 赤外線信号のコード解読



例）エアコン

図 8-1(a) 組み込みプログラミングとマイコン

マイコンとは？

マイコンとは**マイクロ・コントローラ・ユニット**（MCU：Micro Controller Unit）とも呼ばれる集積回路で、小型・低コストで電子機器の制御を行えるため産業・家電問わずさまざまな製品に幅広く用いられています。1チップにCPUだけでなくメモリや外部との接続端子を備えていることを特長とし、さまざまな性能や形状のマイコンが開発されています。

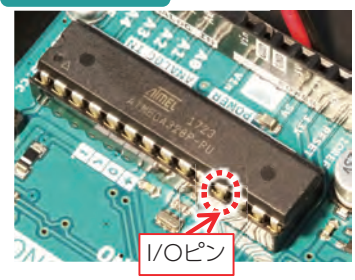
マイコンと外部との入出力用の接続端子を**I/O（Input/Output interface：入出力インターフェイス）**と呼びます。マイコンチップからのびるI/Oの各端子を**I/Oピン（Pin：針）**と呼び、複数のI/Oピンのひとまとまりを**I/Oポート（Port：港）**と呼びます。

マイコンプログラミングで考慮すべきもののひとつにI/Oの制御とマイコン内部での処理があります。

汎用パソコンでのソフトウェアプログラミングとは少し異なり、いかに**マイコンの外の世界の状況**を想定し、外部の機器をコントロールできるかが重要になります。

ARMマイコン

AVRマイコン



PICマイコン

図 8-1(b) マイコン

©MGIC Inc.

汎用 PC と組み込み機器の違い

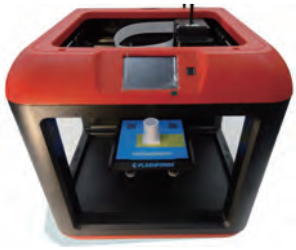
PC やスマートフォン・タブレットなどは、ソフトウェアをユーザーが導入・インストールすることで端末に様々な機能を持たせることができ、通常、機器本体 (ハードウェア) の制御は OS が行います。

一方で、組み込みプログラム開発では特定機器の制御に重点が置かれ、マイコン周辺の電子回路や入出力装置の制御まで考慮する必要があります。一般的な組み込み機器では、組み込まれたプログラムをユーザーが改造したり消したりすることは想定されていないのです。

PC・スマートフォン等

ソフトウェアを導入して様々な機能を持たせることができる。

例) 表計算・カレンダー・動画再生・通話アプリ　etc...



家電・工業機械等（組み込み機器）

その**機器にとって必要な機能**がプログラムされている。

図 8-1(c) PC と組み込み機器

実験に必要なもの

本テキストでは、**ハードウェア**と**ソフトウェア**の両方を作成して実験を進めます。

ハードウェアとして必要なものは E-Station や E-Station 付属品を活用することができますが、ソフトウェアを組み込むための PC や開発ツールは別途用意して頂く必要があります。

開発ツールのインストールについての解説は本テキスト添付 CD 内の「PSoC Creator インストールガイド」を参照してください。

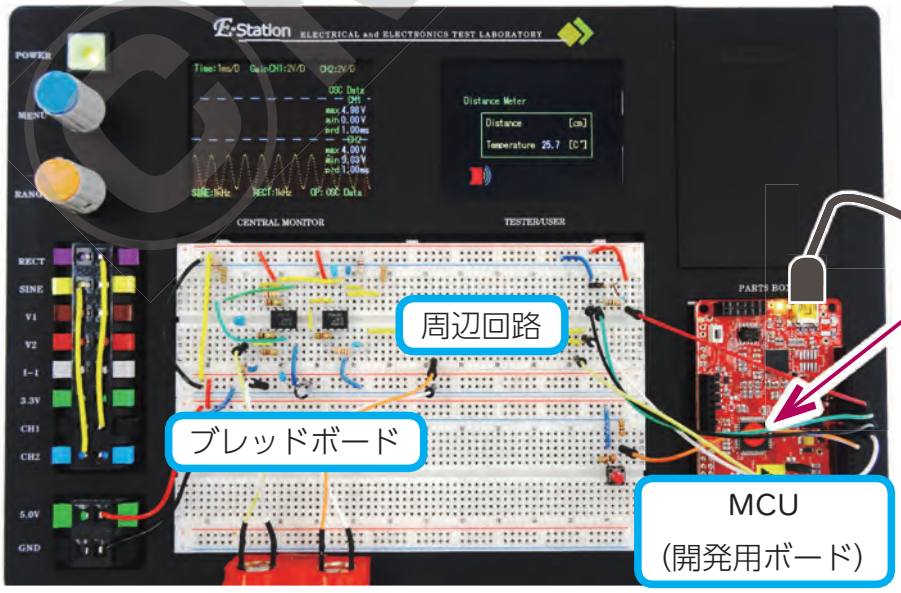
ハードウェア
- 電源 (E-Station)
- 周辺回路 (E-Station)
- 入力装置・出力装置 (E-Station)
- 接続ツール (E-Station 付属 USB ケーブル)
- コンピュータ (E-Station 付属開発用ボード)

ソフトウェア
- **開発用 PC（※別途ご用意してください）**
- **開発ツール（※別途インストールが必要です）**

PSoC　Creator（開発ツール）

プログラムを作成する

PC



E-Station

プログラムを書き込む

図 8-1(d) PC と E-Station

© MYCOM Inc.

開発用ボード：PSoC4 パイオニアキット

組み込みプログラミングを始めるには、コンピュータ (マイコン) がもつ記憶装置にプログラムを書き込むための準備が必要です。実際にはコネクタや、マイコンを動作させるための電源回路や保護素子、コネクタなどが必要ですが、1 枚のボード上にこれらのパーツを実装したマイコンボードを使うと簡単にプログラムを書き込んだり、I/O を使って実験や製品の試作を行ったりすることができます。

本テキストでは、E-Station 付属のマイコンボード「PSoC4 パイオニアキット」を用いて組み込みプログラミングの実験を行います。

ピーソック (PSoC：Programmable System on Chip) は広義にマイコン (MCU) に分類される IC チップです。MCU として機能する CPU を持つだけでなく、メモリやクロック発振器、A/D 変換器、D/A 変換器、その他論理ゲートなどの周辺回路をユーザーの任意に書き変えてチップ内部にハードウェアを作ることができる特長があります。

本テキストでは図 8-1(f) に示すような手順で実験を行います。

なお、添付 CD 内には全実験のサンプルプロジェクトが保存されています。本テキストの前半部は実際にソースコードを記述していただく形式になっていますが、このサンプルプロジェクトを読み込み、使用していただいてもかまいません。

図 8-1(e) PSoC4 パイオニアキット

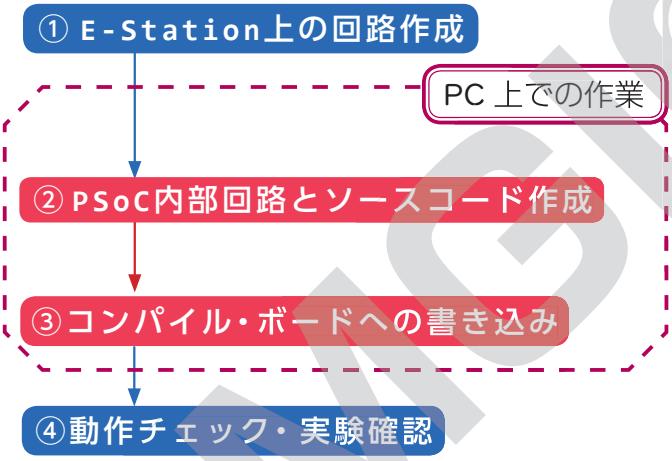


図 8-1(f) プログラミング手順

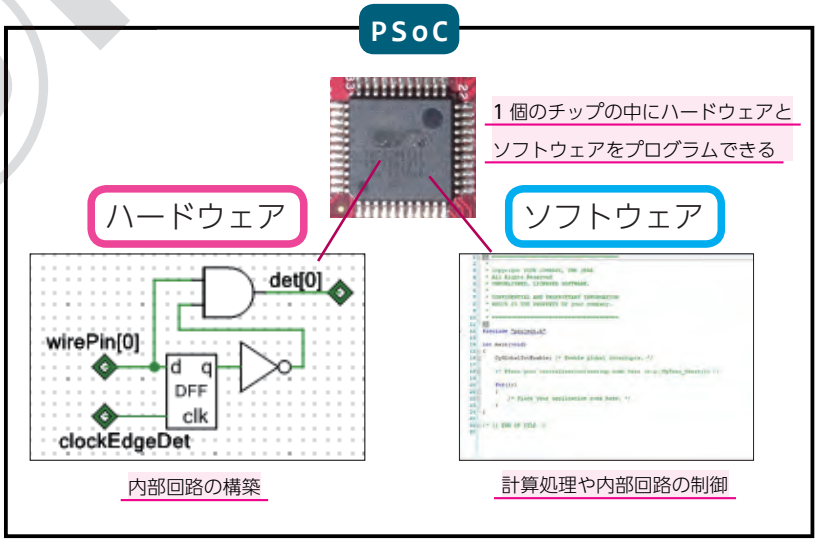


図 8-1(g) PSoC

©MGS

PSoC Creator を使ってみよう

PSoC4 パイオニアキットにチップ内部の回路やプログラムを書き込むための開発ツールとして、Cypress 社が提供している無償ツール「PSoC Creator」を使います。インストール方法については別添の「PSoC Creator インストールガイド」を参照してください。以降、お使いの PC に最新の PSoC Creator がインストールされているものとして解説を行います。

※開発ツールのバージョンアップにより、操作画面が本テキストの図と異なる場合がございます。
本テキストでは PSoC Creator 4.2 を使用しています。

新規プロジェクトの作成方法

① PSoC Creator を起動し、メニューバーを「File > New > Project」の順に選択します。
②開発ターゲットの選択画面では「Target kit」の一覧から「CY8CKIT -042 (PSoC4200)」を選択し、「Next >」をクリックします。

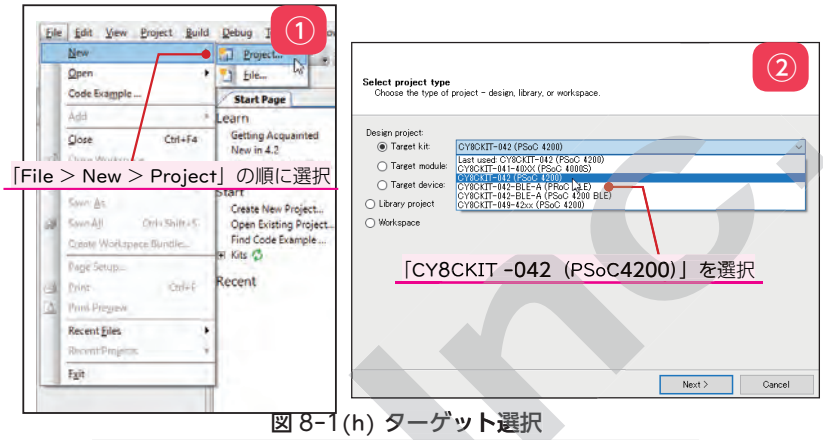


図 8-1(h) ターゲット選択

③「Select project template」画面で「Empty schematic」を選択してから、「Next >」をクリックして、次の「Create Project」画面で「Workspace：Create new workspace」が選択されていることを確認します。
「Workspace name」欄に任意のワークスペース名を入力します。「Location」欄は保存場所の選択です。デフォルトでは「マイドキュメント」内のフォルダが入力されます。
最後に、「Project name」欄にプロジェクト名を入力し、「Finish」をクリックします。
（プロジェクト名は各実験ページで示します）

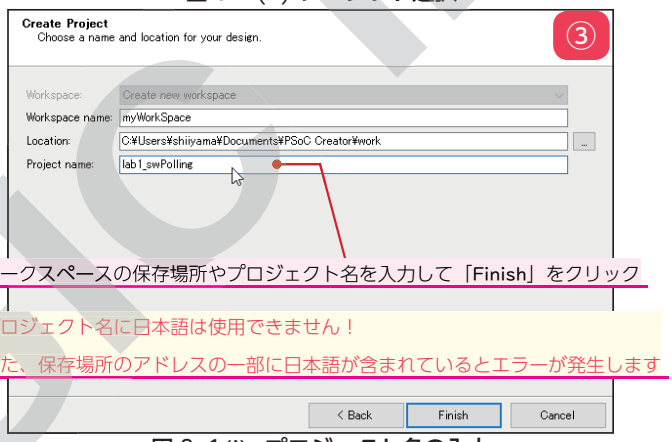


図 8-1(i) プロジェクト名の入力

④この手順により新しいプロジェクトを作成することができます。
以降の実験解説ではこの手順を参考に新規プロジェクトを作成してください。

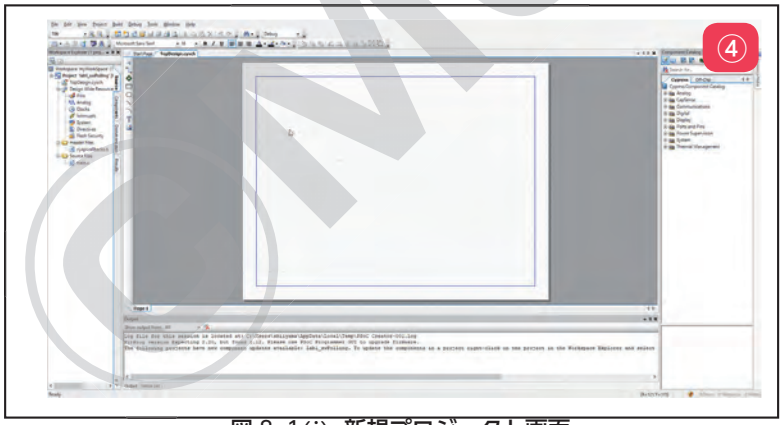


図 8-1(j) 新規プロジェクト画面

プロジェクトとワークスペースの違い

プロジェクトは PSoC の設計データ・ソースコード等のファイル一式を指します。
ワークスペースは、複数のプロジェクトをまとめておくためのフォルダです。通常は、「ワークスペースフォルダ>プロジェクトフォルダ>設計データ・ソースコード」という構成になります。
PSoC Creator の初期設定では画面左側に**ワークスペースエクスプローラー**があり、各プロジェクトの管理を行うことができます。

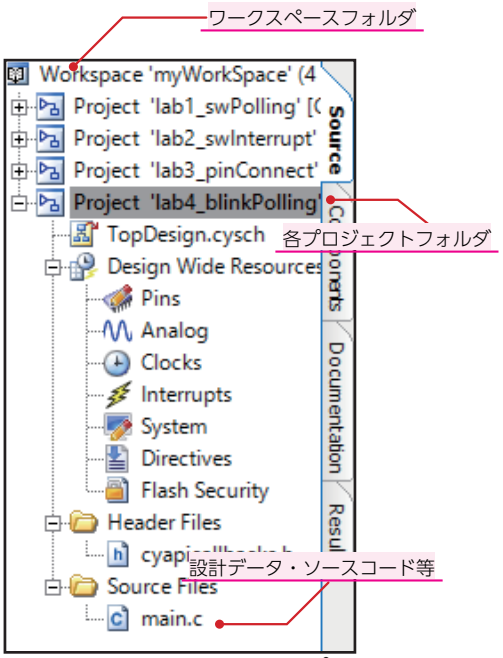


図 8-1(k) ワークスペースエクスプローラー

第9部
組み込みソフトウェア

はじめに 9-1 サンプルプロジェクトの使い方 …… 2
実験 9-2 パルス幅変調制御（PWM） …… 3
解説 9-3 DCモーターの構造 …… 12
実験 9-4 サーミスタで温度測定 …… 14
解説 9-5 サーミスタの温度計算 …… 18
実験 9-6 超音波センサーで距離測定 …… 19
実験 9-7 脈拍モニター …… 26
実験 9-8 スペクトラムアナライザ …… 32
実験 9-9 電子ピアノ …… 36
コラム 音階の周波数 …… 42

©MGIC Inc.

9-6　超音波センサーで距離測定

実験

学習ポイント

別冊の本テキスト実験 [4-14]「超音波センサーで送受信」では超音波センサーの実験を行い、超音波が物体に当たったときの反射波を受信機でとらえ、信号を確認することができます。
超音波送受信機を用いて、距離を求めてみましょう。

チェック

今回は、ごく短時間だけ超音波を放射し、反射波が返ってくるまでの時間を計測するプロジェクトを使います。また、音波の伝搬速度は温度の影響を受けるため、実験 [9-4]「サーミスタで温度測定」の温度測定回路があれば、今回の実験回路に組み込むことができます。
温度測定回路を使用しない場合は後に解説するソースコードの 1 か所を書き換えてください。

プロジェクトの確認

添付 CD 内のフォルダ「work9-6」をお使いの PC 内にフォルダごと保存してください。使用するのはフォルダ内にあるファイル「proj9-6.cywrk」です。
※ PC 内の保存場所について、一部のフォルダ名に日本語が使用されているとエラーで正常に開くことができなくなります。必ず日本語パスの含まれていない場所に保存してください。
今回、次のような動作を行うサンプルプロジェクトを使用します。

動作 A. スイッチを押すと 40kHz の矩形波を約 400 マイクロ秒 (us) 送信し、受信検出までの時間を測定する
動作 B. 測定した時間から超音波の移動距離を計算し、USER MONITOR に表示する

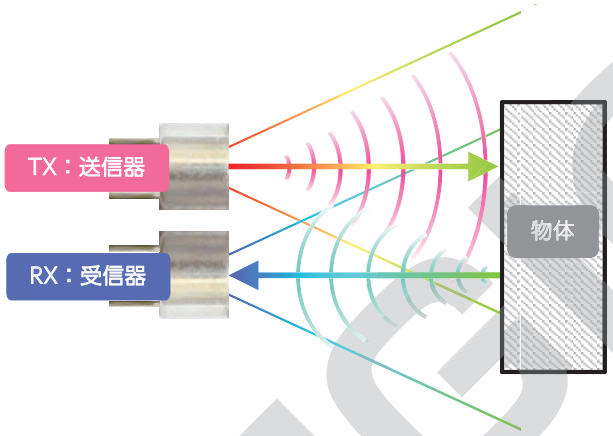


図 9-6(a) 超音波の移動距離

図 9-6(b) 超音波センサー

また、超音波センサーに付属のセンサーユニットを取り付け、図 9-6(d) のように実験装置をセットしてください。
布製の柔らかい物体などは音波の成分のほとんどを吸収してしまうため、反射波の受信がうまくいかないことがあります。
色々な素材で実験を行ってみましょう。

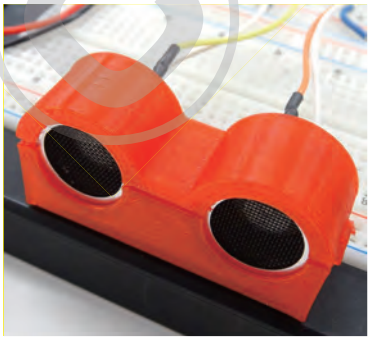


図 9-6(c) センサーユニット

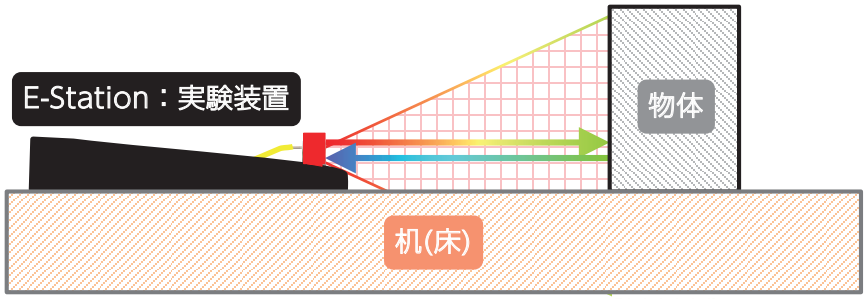


図 9-6(d) 実験装置の構成

回路を作ろう！

① 図 9-6(f) の接続図を参考に、図 9-6(e) の回路を組みましょう。
② 温度測定回路を使用しない場合は、PSoC4 パイオニアキットの P3[4]、P3[6]、P2[6] 番ピンには何も接続しません。
③ PSoC4 パイオニアキットの P0[4] 番ピンを接続するかわりに E-Station の RECT 信号を接続し、OUT1 ～ OUT2 の出力をオシロスコープで確認すると動作チェックを行うこともできます。実験 [4-14] の手順を参考にしてください。

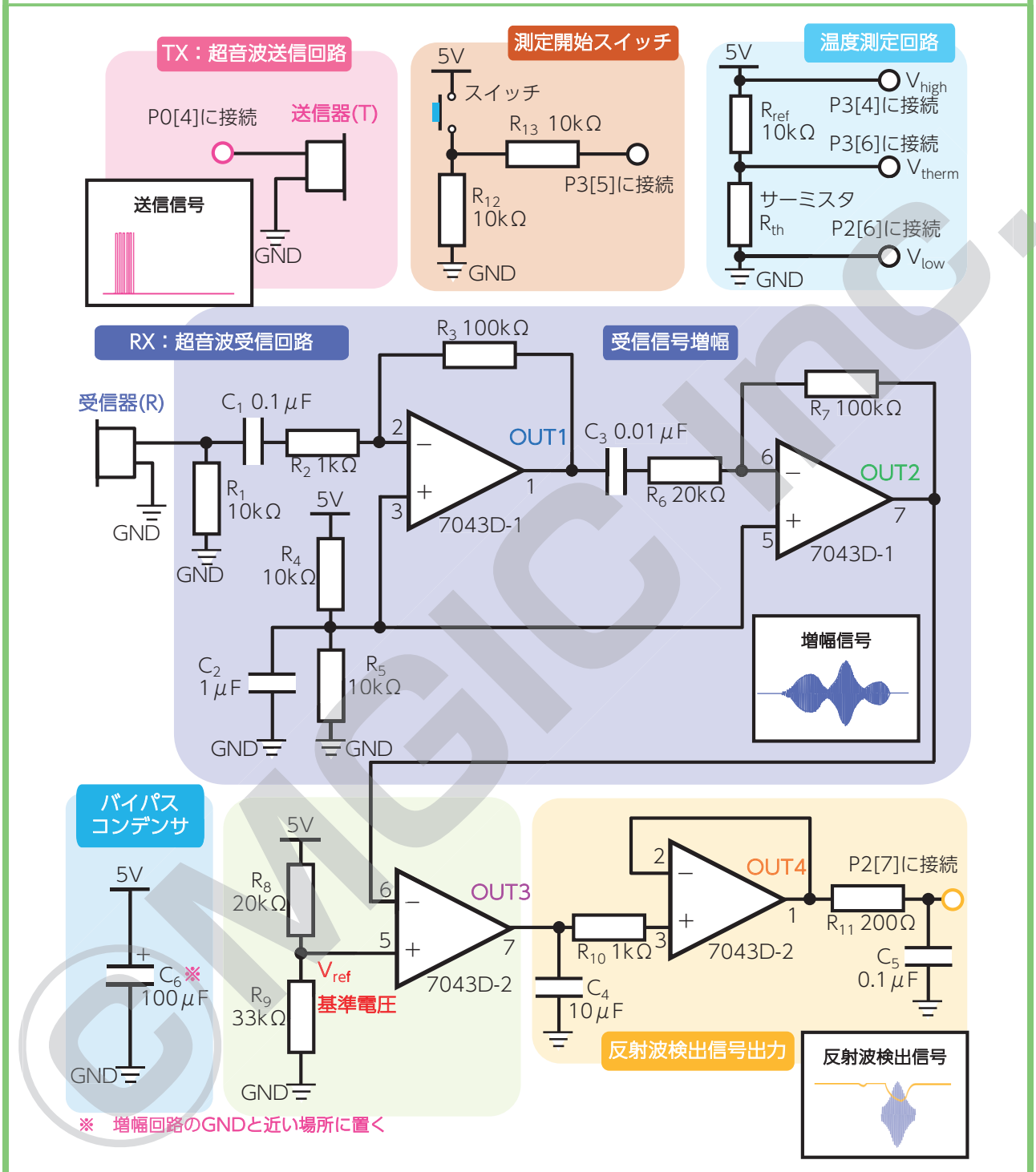


図 9-6(e) 回路図

回路を作ろう！

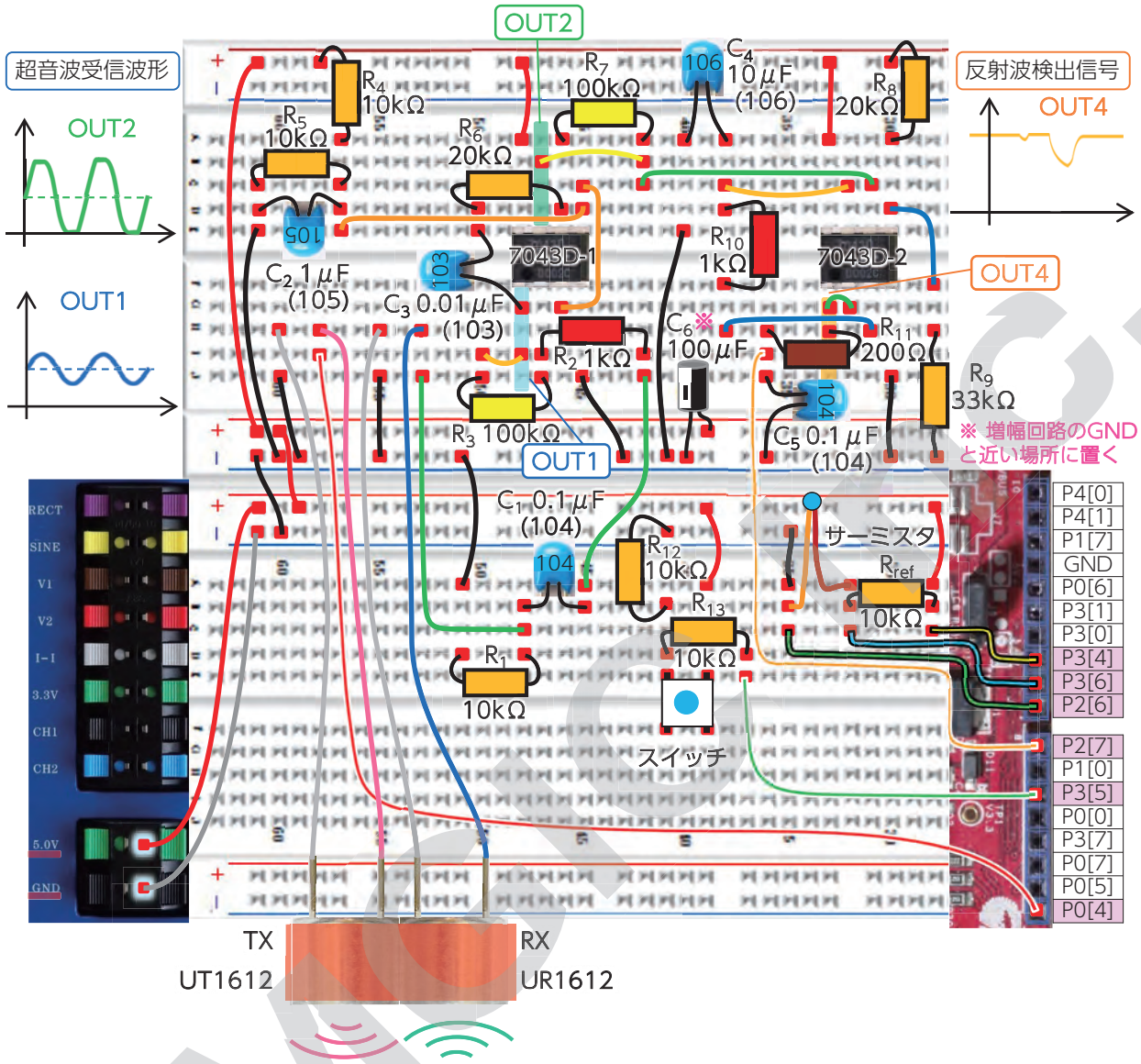


図 9-6(f) 接続図

温度測定回路を使用しない場合

プロジェクト内の「main.c」を開き、10 行目の「#define noTherm (0)」を「#define noTherm (1)」に書き換えてください。
この変更により、プログラムは温度を「25℃」と仮定して距離測定を行います。

```
/* 温度測定回路を併用しない場合は(1)に書き換える */
#define noTherm (0)
```

この 1 か所を変更

図 9-6(g) main.c

サンプルプロジェクトの書き込み

PC 内に保存したプロジェクトファイル「proj9-6.cywrk」を開きましょう。
PSoC ボードと PC を USB ケーブルで接続し、E-Station の電源をオンにします。回路に問題がなければ、メニューバーを「Debug > Program」の順に選択し、書き込みを行います。

正常に書き込みが完了した場合、E-Station 本体右側の USER MONITOR に図 9-6(h) のような画面が描画されます。

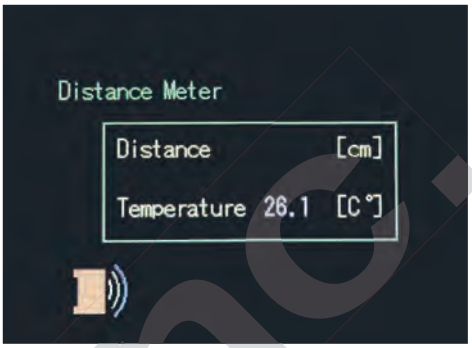


図 9-6(h) USER MONITOR

実験のすすめ方

①E-Station 上の超音波センサーの正面に距離を測定したい物体を置く
②測定開始スイッチを押し、距離が USER MONITOR に表示されることを確認する
③物体の距離を約 10cm ずつ移動させ、測定結果を記録する。
④物体の材質を木材や布などに変更し、測定結果に大きな変化があるかどうかを確認する。

ワークシートに記入しよう！

測定データ例

同じ物体・同じ距離でも、複数回測定すると測定値のばらつき(偏差)が発生します。
これは、回路の遅延やマイコンの読み取りタイミングによって起こる誤差です。ここでは、図 9-6(i) に同回路で行った距離測定時のデータ例を示します。

■測定条件
気温：約 28℃　床面：会議机(メラミン材)
■測定対象
電子情報通信ハンドブック(紙製)
縦：27cm　横：21cm　厚さ：7cm
■試行回数
距離約 10cm 毎に 100cm まで各 10 回測定
※なるべく硬質な床面・測定対象を使用しましょう

■測定データ　単位：[cm]										
測定距離(理想値)	10	20	30	40	50	60	70	80	90	100
平均値	9.8	20.2	30.3	41.4	51.4	61.6	71.5	82.0	92.1	102.0
最大値（max）	10.0	20.3	30.6	41.5	51.7	61.7	71.6	82.1	92.3	102.0
最小値（min）	9.6	20.1	30.2	41.2	51.2	61.5	71.4	81.8	91.9	102.0

図 9-6(i)　実測データ

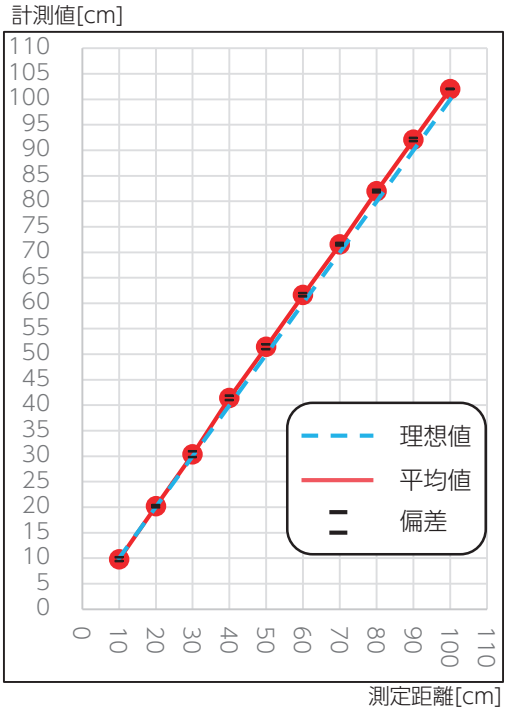


図 9-6(j)　計測距離 - 測定値のグラフ

©MCI Inc.

今回の実験回路の機能について、ブロックで表現したものを図 9-6(k) に示します。

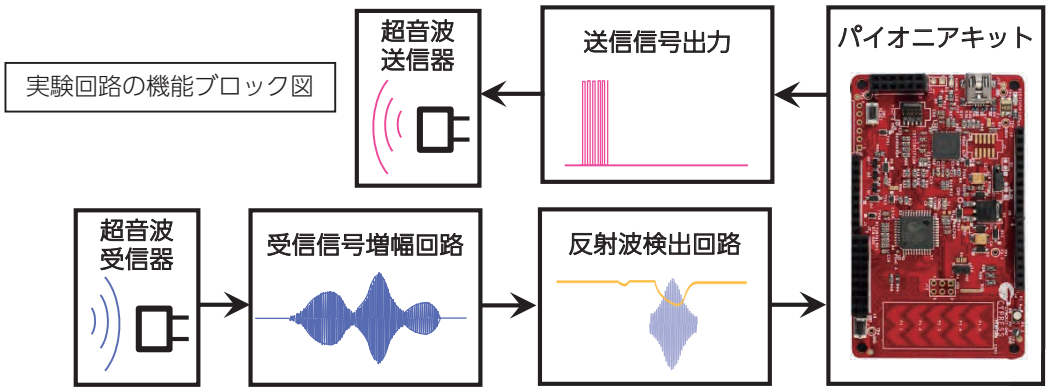


図 9-6(k) ブロック図

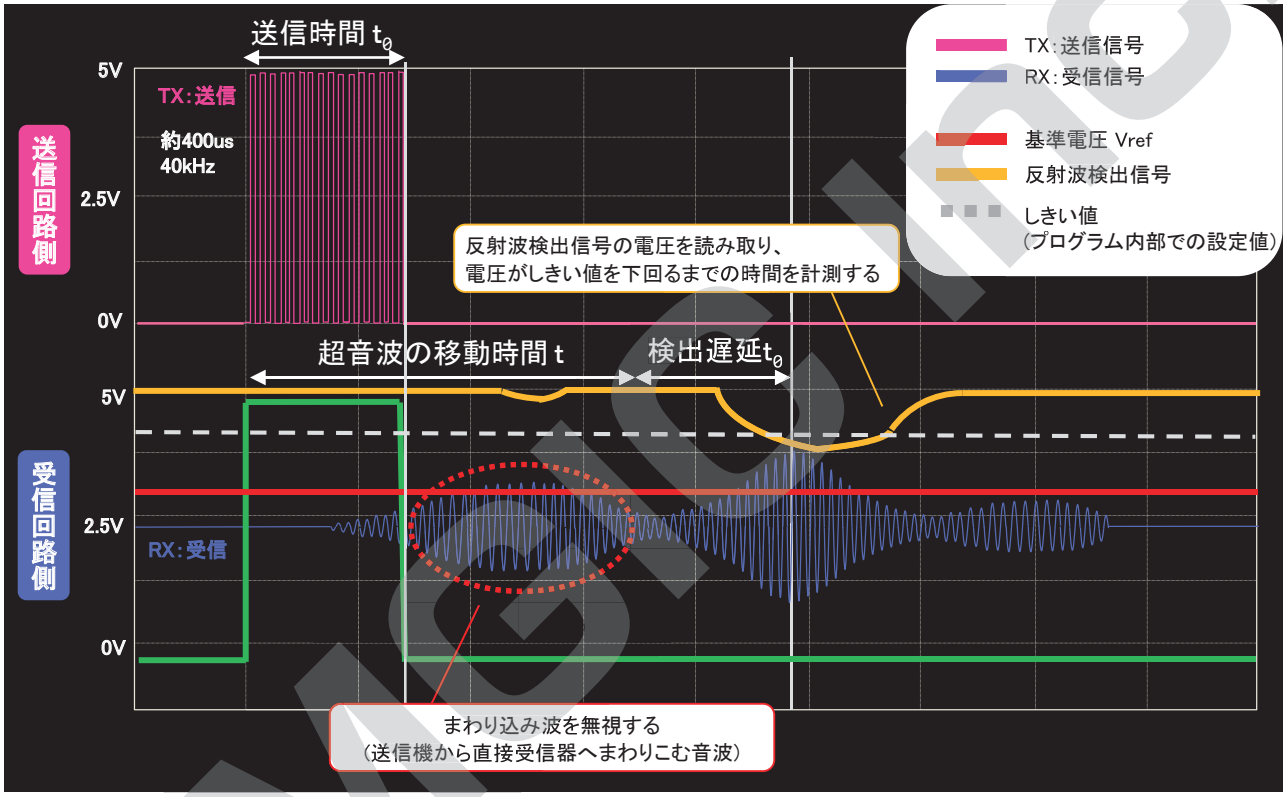


図 9-6(l) 超音波送受信の概要

今回の実験回路は、超音波送信開始から反射波検出までの時間を測定するためのものです。送信器に 40kHz の矩形波を入力すると超音波が放射され、受信器で反射波を受信します。

超音波受信機で受信した信号はそのままでは微弱なので、複数の増幅回路を経由して最終的に安定した検出信号を生成します。

超音波の送信を開始した時点から、反射波検出信号の電圧がしきい値(プログラム内部での設定値：図 9-6(1)(灰色破線)を下回るまでの時間を計測し、超音波の移動時間とします。

時間を求めることができれば、音速と時間から超音波の移動距離がわかります。ただし、超音波の移動距離は往復分であるため、その半分が物体までの距離となることに注意してください。

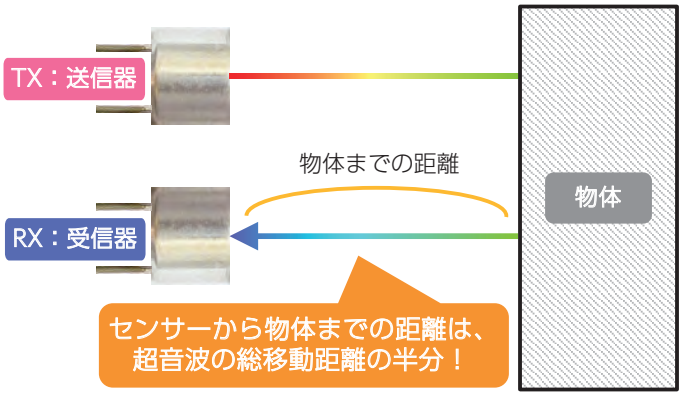


図 9-6(m) 超音波の移動距離

OMRON

サンプルプロジェクト

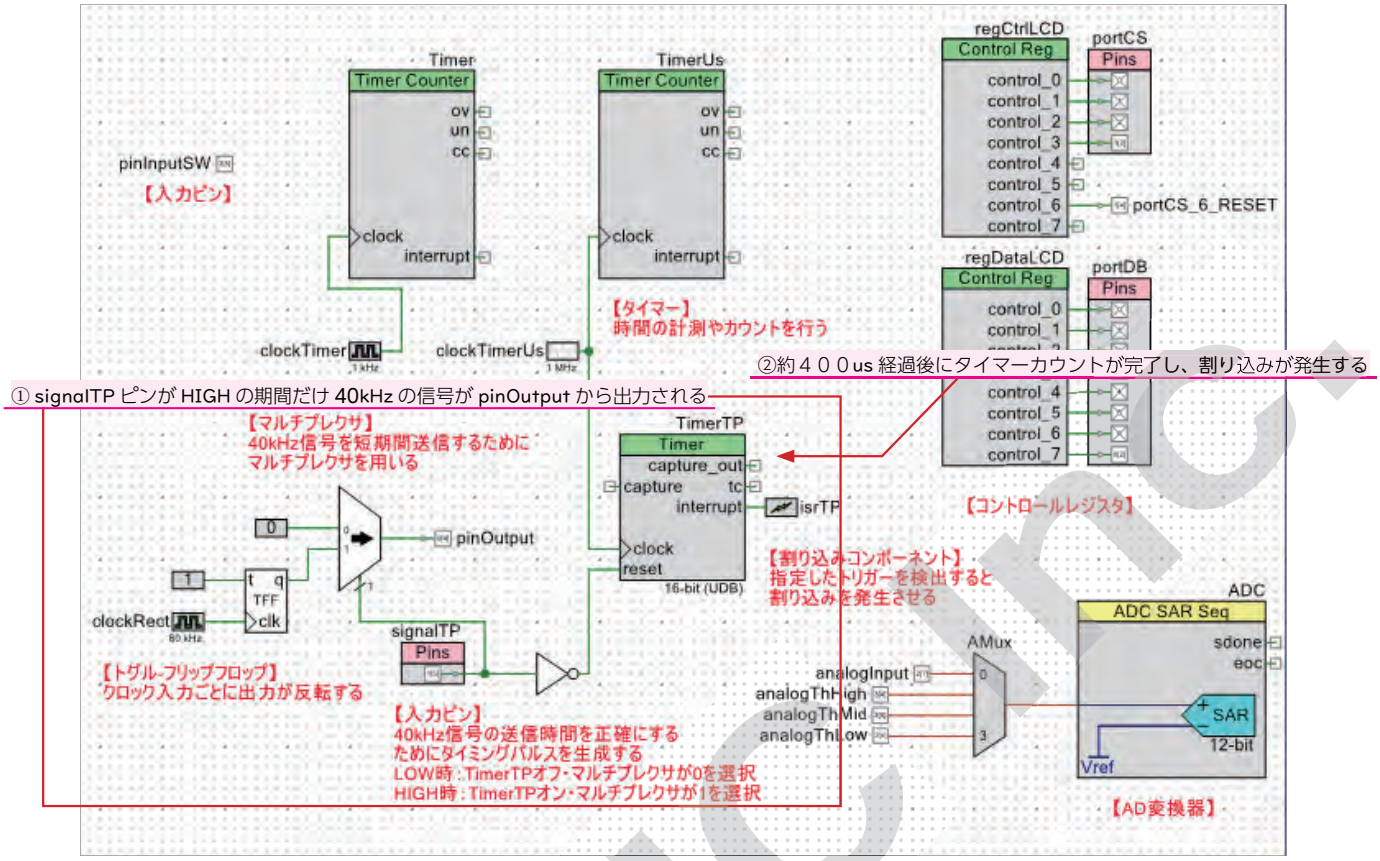


図 9-6(n) PSoC 内部回路図

実験で使用するサンプルプロジェクトの内部回路を図 9-6(n) に示します。

トグル - フリップフロップ (TFF)

クロック入力ごとに出力値が反転するものをトグル - フリップフロップ (TFF) と呼びます。TFF に 80kHz のクロックを入力することで、40kHz の信号を作り出しています。

マルチプレクサ (Mux)

選択信号の値によって、Ch0 もしくは Ch1 のどちらかの信号が選択されて出力されます。通常時は選択信号は「0」が入力されており、出力も「0」のままですが、選択信号に「1」が入力されているあいだは TFF の 40kHz 信号が出力されます。

タイマー

クロック入力ごとにカウントを行い、指定した値まで達したときにイベントを起こしたり、カウント途中の値をプログラムに渡したりすることができます。

このプロジェクトでは、通常時はリセット端子に「1」が入力されているため、リセットがかかり続けることで動作を停止させています。

リセット端子に「0」が入力されているあいだはカウントが進行し、約 400 マイクロ秒 (us) 経過時に割り込み信号を生成し、割り込みを発生させます。

OMRON

次に、ソフトウェアのソースコードの一部を下記に示します。
ソースコード全文はファイル「proj9-6.cywrk」内の「main.c」を開いてください。

```
/* ------------------------------------------------------------ *
   module name     : CY_ISR(isrTPClear)
   function        : 割り込みサービスルーチン
   ------------------------------------------------------------ */
CY_ISR(isrTPClear){
    // 高速で signalTP ピンの値を (0) にクリアする
    CY_SYS_PINS_CLEAR_PIN(signalTP__DR, signalTP_SHIFT);
}
/* ------------------------------------------------------------ *
   module name     : txStart()
   function        : 超音波送信開始
   parameter(引数): uint16 dur     送信時間 (us)
   return(戻り値) :
   ------------------------------------------------------------ */
void txStart(uint16 dur){
    // タイマーに送信時間をセットする
    TimerTP_WritePeriod(dur);
    // 高速で signalTP ピンの値を (1) にセットする
    CY_SYS_PINS_SET_PIN(signalTP__DR, signalTP_SHIFT);
}
/*  TimerTP(タイマー) に作動時間をセットしてから signalTP ピンの値を (1) にすると
TimerTP が動き出す。タイマーカウントが dur の値に達したら割り込みが発生し、signalTP
ピンの値を (0) にする。タイマー終了・割り込み発生までの TimerTP の制御はハードウェ
アが実行するため、ソフトウェアは送信開始直後も別の処理を行える。*/


/* ------------------------------------------------------------ *
   module name     : distMeas
   function        : 距離測定
   parameter(引数): float temp     周囲温度 (℃)
   return(戻り値) : int calc       距離 (mm) もしくは  -1(エラー)
   ------------------------------------------------------------ */
int distMeas(float temp){
    int calc = 0;                       // 戻り値
    const uint16 txDuration = 400;      // 送信時間 (us)
    const long timeOut = 50000;         // 距離測定時のタイムアウト時間 (us)
    const float rxThreshold = 4.4    ;  // 入力しきい値電圧 (0V-5V)
    const int detectDelay = 250;        // 反射波検出までの遅延 (us)
    const float K = 204.6;        // 電圧値を 10 ビットの値に変換する係数 (1023/5)
    unsigned long speedC = 0;           // 音速 C (mm/s)
    int roundTrip = 0;                  // 超音波の往復距離 (mm)
    int voltage = 0;                    // アナログ入力値
    uint32 time = 0;                    // 時間 (us)

    TimerUs_WriteCounter(0);            // マイクロ秒タイマーをリセット
    txStart(txDuration);                // 超音波送信開始

    /* 受信が行われるか、タイムアウトが発生するまでループ */
    while(1) {
        time = TimerUs_ReadCounter();       // 現在の時間を確認
        voltage = ADC_GetResult16(0);       // 現在の入力電圧値を取得

        /* 入力電圧値がしきい値を下回れば受信と判定 */
        if(rxThreshold * K > voltage) {
            speedC = 331500 + (610 * temp); // 音速を [mm/s] に換算して計算
        /* 1 気圧下の音速
        C = 331.5[m/s] + 0.61 * temp[℃] -> (331500+610*temp)[mm/s] */
/* 送信から受信までの時間を元に往復距離を計算、片道分の距離を割り出す */
            roundTrip = (speedC * (time - detectDelay)) / 1000000;
            calc = roundTrip / 2;
            break;
        }
        /* タイムアウト時の処理 */
        if(timeOut <= (long)time) {
            calc = -1;                      // 戻り値にエラー値をセット
            break;
        }
    }
  return calc;                          // 測定値もしくはエラー値を返す
}
```

タイマー割り込み発生時に signalTP ピンの状態を「0」に変更し、タイマーをリセットする

超音波の送信から受信までの時間を取得し、音速から距離を割り出す

PSoC Creator 独自のコード

CY_SYS_PINS_SET_PIN()

CY_SYS_PINS_CLEAR_PIN()

これらのコードは、指定したピンの状態をトグル(切り替え)するものです。
PIN_Write() による値の上書きと違うのは、こちらのコードの方がより高速にトグル(切り替え)が行えるという点です。

Timer_WritePeriod()

このコードは、タイマーがカウント完了する値を設定します。
「400」を指定した場合、1MHz で動作するタイマーの 1 周期は約 1us となるため、約 400us でカウントが完了する計算です。

CMC Inc.

9-9 電子ピアノ

実験

学習ポイント

ここまでの実験では、入力されたアナログ信号を取り込む A/D 変換器を使ったプログラムが主でした。今回は D/A 変換器を使ってデジタル値を疑似的なアナログ信号として出力し、ピアノの音を再現するプログラムを使用します。

チェック

音は空気の振動であり、スピーカーは入力された電気信号に応じて空気を振動させて音を出します。スピーカーに入力する信号を PSoC 内部の D/A 変換器で生成し、電子ピアノを作りましょう。

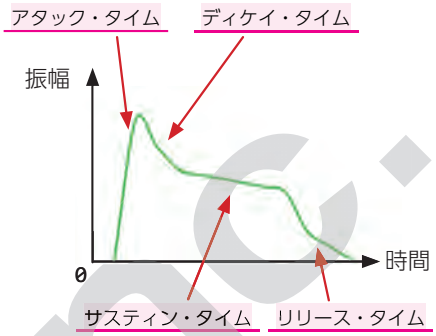
ピアノ音を再現するためのデジタルデータ

矩形波では音が汚く聞こえたり、正弦波をそのまま入力しても実際の楽器の音のようには聞こえません。ピアノのような音を電子的に再現しようとする場合、信号の**強弱を時間とともに変化させる**必要があります。

シンセサイザーなどでは、鍵盤を押した瞬間の大きな音の区間 (アタック・タイム)、瞬間的に大きく減衰する区間 (ディケイ・タイム)、鍵盤が押されている間音が持続する区間 (サスティン・タイム)、鍵盤が離されて音が減衰する区間 (リリース・タイム) の 4 種類の強弱の付け方が用いられています。

そこで、今回の実験用プロジェクトでは正弦波を基本の信号として、一定時間ごとに振幅の強弱情報をソフトウェアで付け加えて出力します。

音の高さは押された鍵盤に応じて周波数を設定します。



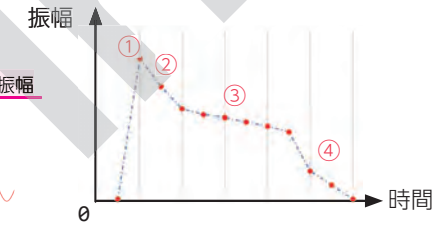


図 9-9(a) 信号の強弱

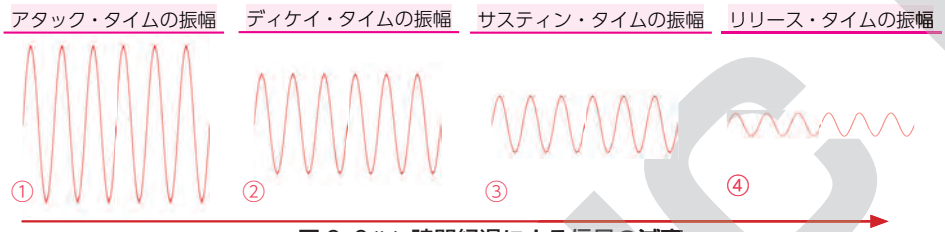


図 9-9(b) 時間経過による信号の減衰

音の高さは押された鍵盤に応じて周波数を設定します。正弦波を 360 分割し、タイマーを使って一定時間ごとに D/A 変換器の出力値を変更して**疑似的な正弦波**を再現します。

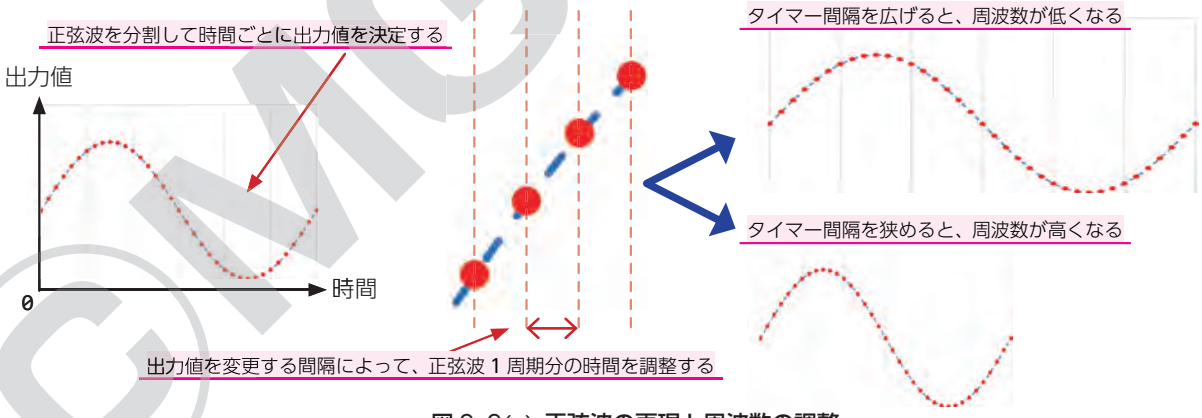


図 9-9(c) 正弦波の再現と周波数の調整

プロジェクトの確認

添付 CD 内のフォルダ「**work9-9**」をお使いの PC 内に**フォルダごと**保存してください。使用するのはフォルダ内にあるファイル「proj9-9.cywrk」です。

※ PC 内の保存場所について、一部のフォルダ名に日本語が使用されているとエラーで正常に開くことができなくなります。必ず日本語パスの含まれていない場所に保存してください。

今回、次のような動作を行うサンプルプロジェクトを使用します。

動作 A. 「ドレミファソラシド」に対応する 8 個のスイッチ入力を判定し、電子ピアノ音を出力する
動作 B. 出力信号には時間とともに強弱を付ける
動作 C. 880Hz の「ラ」の音を基準として、「ドレミファソラシド」1 オクターブ分を扱う

©MCI Inc.

ハードウェア作成

E-Station 上に回路を組み、次に PSoC にプロジェクトを書き込みましょう。

回路を作ろう！

① 図 9-9(e) の接続図を参考に、図 9-9(d) の回路を組みましょう。
② 回路図では省略していますが、8 個分のスイッチ回路を作成します。それぞれ接続先を間違えないように注意して配線してください

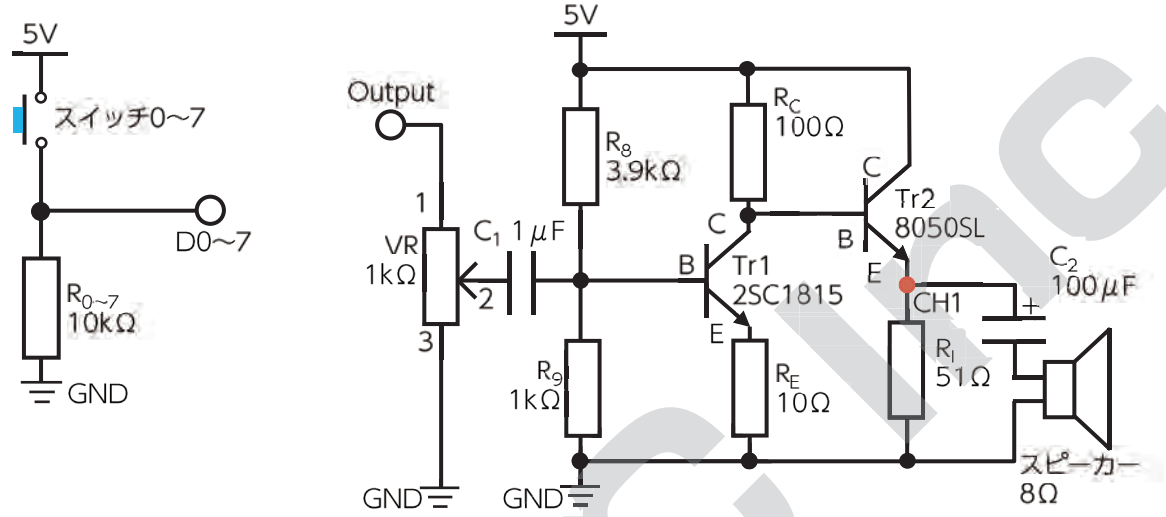


図 9-9(d) 回路図

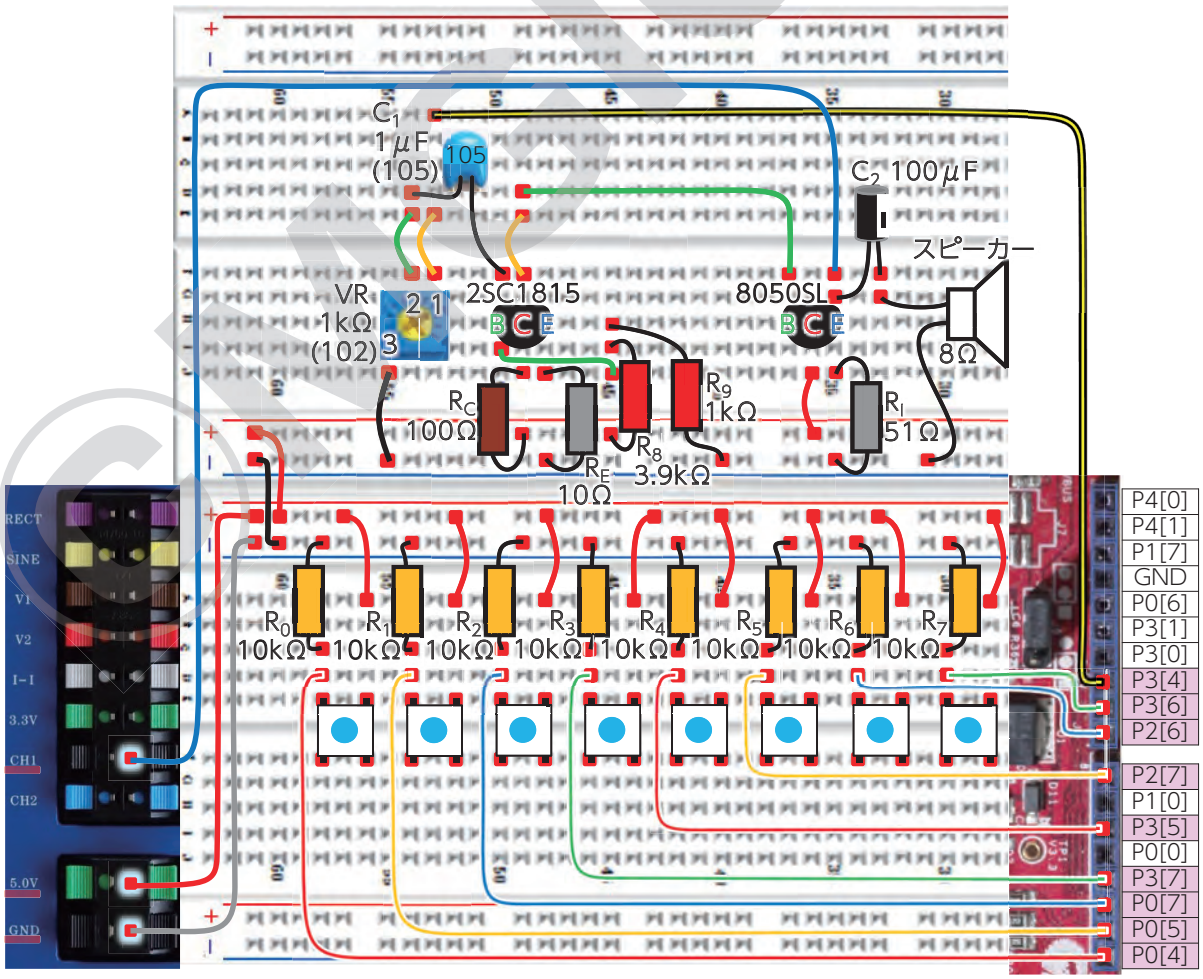


図 9-9(e) 接続図

サンプルプロジェクトの書き込み

PC 内に保存したプロジェクトファイル「proj9-9.cywrk」を開きましょう。
PSoC ボードと PC を USB ケーブルで接続し、E-Station の電源をオンにします。回路に問題がなければ、メニューバーを「Debug ＞ Program」の順に選択し、書き込みを行います。

正常に書き込みが完了した場合、E-Station 本体右側の USER MONITOR に図 9-9(f) のような画面が描画されます。

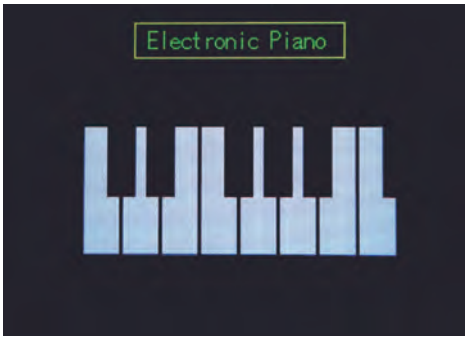


図 9-9(f) USER MONITOR

実験のすすめ方

①スイッチを順番に押して「ドレミファソラシド」の順に音が出力されることを確認する。
②スイッチを押したときの出力信号の変化を、オシロスコープ CH1 で観察する。

サンプルプロジェクト

実験で使用するサンプルプロジェクトの内部回路を図 9-9(g) に示します。

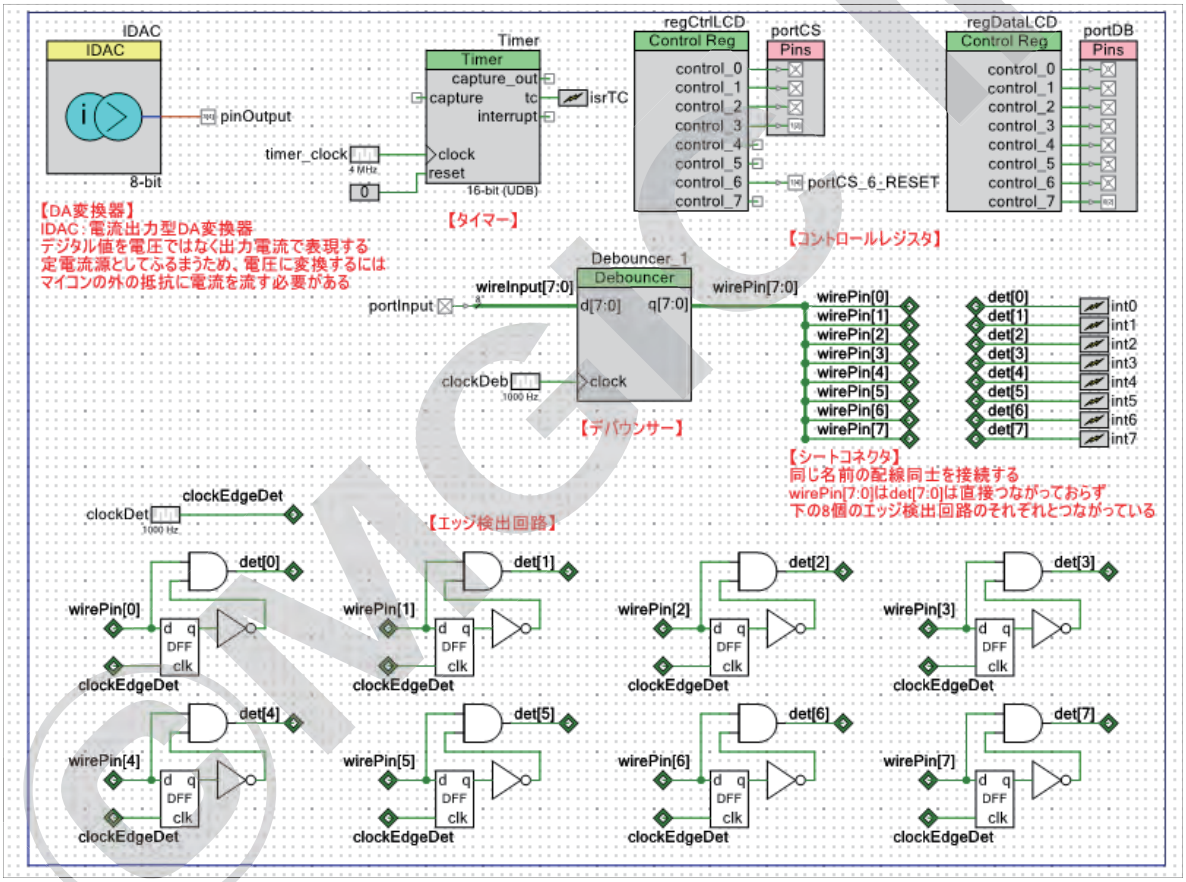


図 9-9(g) PSoC 内部回路図

エッジ検出回路

スイッチを 1 度押されたときに 1 回分のパルスを生成します。

シートコネクタ

同じ名前の配線同士を接続します。複雑な回路を作るときに他の配線との接触を防ぐことができます。

© MCC Inc.

IDAC(Intensity digital to analog converter)

PSoC は内部回路に D/A 変換器を組み込むことができます。実験で使用する PSoC4 では、**IDAC** と呼ばれる電流出力型の D/A 変換器を使用します。

別冊の本テキスト実験 [6-1]「デジタルーアナログ変換(DAC)」で作成したものは、デジタル値を出力電圧に変換するタイプの D/A 変換器ですが、IDAC は**デジタル値を出力電流に変換**します。

IDAC の出力端子には電圧は発生しませんが、デジタル値に応じた電流を流そうとする**電流源**として動作します。

ある値の負荷抵抗 R を出力端子につなぎ、もう一端をグランド (GND) に接続すると、設定された出力電流 I を流そうとするため、負荷抵抗の両端に電圧 V が発生します。

このようにして、出力電流を変化させれば電圧 V を調整することができます。

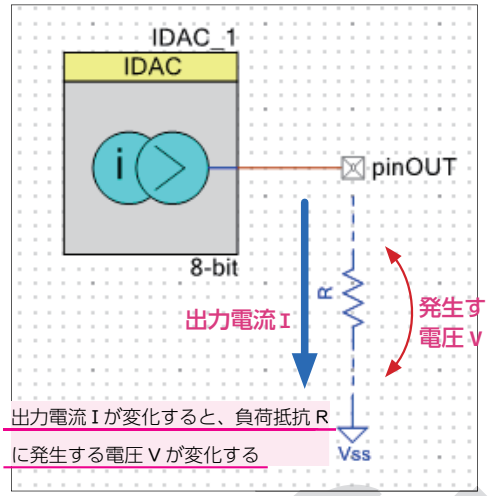


図 9-9(h) IDAC の使い方

次に、ソフトウェアのソースコードの一部を下記に示します。
ソースコード全文はファイル「proj9-9.cywrk」内の「main.c」を開いてください。

```
/* 構造体：音階情報 */
struct _KEY
{
    char *note;                          // 音階名
    char *DoReMi;                        // 音階名
    float freq;                          // 周波数
    uint16 period;                       // 周期 ( タイマー用設定値 )
    int index;                           // 索引
};                                       ← _KEY という名前の構造体の型枠を作成

int main(void)
{
    double aFrq[keyPoints] = {};         // 周波数
    double aPrd[keyPoints] = {};         // 周期
    uint8 pushKeyOld = 0;                // 前回押された鍵盤
    uint8 i = 0;

/* 音階情報 */
    for(int i = 0; i < keyPoints; i++){
        double p1 = 0;
        double p2 = 0;
        double p3 = 0;

        /* 「A5 のラ」 880Hz を基準として、12 音階の周波数・周期を計算 */
        p1 = pow(2.0, 1.0/12.0);
        p2 = pow(p1, 9.0);
        p3 = pow(p1, (double)i);
        aFrq[i] = (880.0 / p2 ) * p3;
    //(880/(2^(1/12))^9)*(2^(1/12))^n [0<=n<13]
        aPrd[i] = (uint16)((1000000/aFrq[i]) * 4) / smpPoints;
        // タイマー用設定値　音の周波数をサンプリング数で割った時間
    }

    /* 構造体：音階情報 */                ← 各要素に構造体変数を持つ配列を宣言
    struct _KEY key[keyPoints] = {
      /* *note     *DoReMi      freq        period     index */
        { "C",      "Do",    aFrq[0],    aPrd[0],     0},
            { "C#", "Do#",   aFrq[1],    aPrd[1],     1},
        { "D",      "Re",    aFrq[2],    aPrd[2],     2},
            { "D#", "Re#",   aFrq[3],    aPrd[3],     3},
        { "E",      "Mi",    aFrq[4],    aPrd[4],     4},
        { "F",      "Fa",    aFrq[5],    aPrd[5],     5},
            { "F#", "Fa#",   aFrq[6],    aPrd[6],     6},
        { "G",      "Sol",   aFrq[7],    aPrd[7],     7},
            { "G#", "Sol#",  aFrq[8],    aPrd[8],     8},
        { "A",      "La",    aFrq[9],    aPrd[9],     9},
            { "A#", "La#",   aFrq[10],   aPrd[10],   10},
        { "B",      "Si",    aFrq[11],   aPrd[11],   11},
        { "C",      "Do",    aFrq[12],   aPrd[12],   12},
    };

    /* 使用する鍵盤のインデックス番号とスイッチ番号の紐づけ */
    int whiteKey[whiteKeys] = {key[0].index, key[2].index, key[4].index,
    key[5].index, key[7].index, key[9].index, key[11].index, key[12].index};
/* 各鍵盤の情報 */
```

構造体 (struct)

C 言語における**構造体**とは、異なる型を要素ごとに決めることができる配列のようなもので、変数の型として扱います。

本来配列は同じ型のデータしか各要素に格納することはできませんが、構造体は中身の要素ごとに型を指定することができます。

構造体の中の各要素を**メンバ**と呼びます。

C 言語由来の関数

`pow(x,y);`

x の y 乗の値を返す数学関数です。引数 x と y および、戻り値はいずれも double 型です。

このサンプルプログラムでは、1 オクターブ分の音の周波数を計算するために使用しています。

構造体のメンバを使用する

構造体の中のメンバにアクセスする場合、その構造体の型として宣言された変数名の後ろにピリオド「.」を付け、メンバ名を追加することで個別に参照・代入することができます。

```
/* 正弦波の値を生成 */
    double pi = acos(-1);                   // 円周率
    double angle = 0;                       // 角度
    double rad = 0;                         // 角度(ラジアン)

    for(int i= 0; i < smpPoints; i++){
            angle = (360/smpPoints) * i;
            rad = pi * angle / 180;
            g_sinTable[i] = (sin(rad) + 1) / 2;
     //-1~1 を 0~1 の範囲に変換
    }

    for(int i= 0; i < whiteKeys; i++){
        g_cycleTime[i] = 50 * (1.0 + (float)i * (1.0 / (float)whiteKeys));
    }

    initComponents();                       //PSoC 内部回路を起動する
    drawDisp();                             //LCD に文字を表示する

    for(;;)
    {
        /* 外部割り込みフラグがセットされていればタイマーを作動させる */
        if(1 == g_keyTriggered){
            /* 各種フラグのクリア */
            drawChar(key[whiteKey[pushKeyOld]].DoReMi,
                58 + (pushKeyOld * 28), 50, BC);
            drawKey(pushKeyOld, initPosX, initPosY, WH);
            g_keyTriggered = 0;
            g_playExit = 0;
            g_cntTable = 0;
            g_cntCycle = 0;
            g_cntEnvelope = 0;
            g_playStart = 1;
            Timer_Start();                  // 音声出力用タイマー作動
            Timer_WritePeriod(key[whiteKey[g_pushKey]].period);
        // タイマー割り込み周期設定
            i = g_pushKey;
            drawChar(key[whiteKey[g_pushKey]].DoReMi,
                58 + (i * 28), 50, RE);
            drawKey(i, initPosX, initPosY, RE);
            pushKeyOld = i;
        }
        /* 出力終了フラグがセットされていればタイマーを停止する */
        else if(1 == g_playExit){
            Timer_Stop();
            drawKey(pushKeyOld, initPosX, initPosY, WH);
            drawChar(key[whiteKey[pushKeyOld]].DoReMi, 58
                + (pushKeyOld * 28), 50, BC);
        }
    }
}
```

時間ごとに 360 点に分けて正弦波のデータを生成する

出力する周波数ごとに再生時間を変える

タイマーが作動すると、一定時間ごとに割り込みが発生するようになる

タイマーカウントのピリオド値を変更することで、出力する正弦波の周波数を決定する

C 言語由来の関数

`acos(x);`

（ただし、`[-1.0 ≤ x ≤ 1.0]` の範囲）

コサイン（余弦）の逆三角関数であり、角度 x でのアークコサイン（逆余弦）を返す関数です。

引数 x と戻り値 y はいずれも double 型です。

$\cos(y)=x$ のとき $y=\cos^{-1}(x)=\mathrm{acos}(x)$ であり、余弦の値が x のときの角度 y を求めます。

$\cos(\pi)=-1$ という関係から、円周率 π を `acos(-1)` で求めることができます。

©MCI Inc.

```
/* ------------------------------------------------------------ *
   module name     : CY_ISR(isrTCount)
   function        : 割り込みサービスルーチン
   ------------------------------------------------------------ */
CY_ISR(isrTCount){
    if(1 == g_playStart){
        /* DA 変換器に出力値をセットする */
        IDAC_SetValue((g_sinTable[g_cntTable] * 255) * g_envelope[g_cntEnvelope]);
        g_cntTable++;
        /* 1 周期分出力したら周期カウントを増やす */
        if(smpPoints <= g_cntTable){
            g_cntTable = 0;
            g_cntCycle++;
            /* 指定の周期分出力したらエンベロープテーブルのカウント値を増やす */
            if(numCycle < g_cntCycle){
                g_cntCycle = 0;
                g_cntEnvelope++;
                /* エンベロープの最後まで出力した段階で終了フラグをセットする */
                if(envPoints <= g_cntEnvelope){
                    g_cntEnvelope = 0;
                    g_playStart = 0;
                    g_playExit = 1;
                }
            }
        }
    }
}
```

タイミングごとに出力値を変化させる

正弦波を複数周期出力した後でエンベロープにそって振幅を変更する

```
/* ------------------------------------------------------------ *
   module name     : CY_ISR(isrPush)
   function        : 割り込みサービスルーチン
------------------------------------------------------------ */
/*  各スイッチ入力に応じた割り込み処理を行う */
CY_ISR(isrPush0){
    g_pushKey = 0;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;          // 音の周波数によって出力回数を調整
する
}
CY_ISR(isrPush1){
    g_pushKey = 1;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush2){
    g_pushKey = 2;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush3){
    g_pushKey = 3;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush4){
    g_pushKey = 4;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush5){
    g_pushKey = 5;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush6){
    g_pushKey = 6;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
CY_ISR(isrPush7){
    g_pushKey = 7;
    g_keyTriggered = 1;
    numCycle = g_cycleTime[g_pushKey] ;
}
```


©MGC inc.

E-Station へようこそ

2019 年　3 月　1 日　初版第 1 刷発行
2019 年　8 月　1 日　初版第 2 刷発行
2019 年　10 月 1 日　初版第 3 刷発行
2020 年　2 月　1 日　初版第 4 刷発行

© 著者　椎 山 晋 太 郎
発行人　近 藤 充 資
発行所　株式会社 MGIC
〒 530-0051
大阪府大阪市北区太融寺町 5-13
東梅田パークビル 8F
TEL　(06) 6195-8680
FAX　(06) 6195-8681
URL　http://m-gic.com
E-Mail　contact@m-gic.com
印刷／製本　株式会社西川印刷所
編　集　竹 下 史 倫
編　集　村 上 達 雄
イラスト　椎 山 晋 太 郎

●本書記載の製品名について
記載されている製品名「E-Station」は株式会社 MGIC の登録商標です。
●本書記載内容についてのご注意
本書で記載されている内容は著作権法により保護されています。また、本書に記載されている内容によって発生した損害は、株式会社 MGIC ならびに著作権者は責任を負いかねますので、あらかじめご了承ください。
●本書の実験内容についてのご注意
本書で記載されている実験については、ユーザー側にて実施した場合の安全性を保障するものではありません。電力を使用するため、実験手順によっては火災等の危険を伴う場合があります。安全を十分確保したうえで実施してください。なお、実験に伴う事故等について、株式会社 MGIC ならびに著作権者は責任を負いかねますので、あらかじめご了承ください。
●本書付属のプロジェクトについてのご注意
本書に記載されているプロジェクト（プログラム、電子データ等）は本書の付属物として著作権法により保護されています。個人使用での改変は認めますが、許可がない限り配布、改変、複写および複製は行えません。
万が一、落丁や乱丁がございましたら、弊社負担にてお取替えいたします。

©MGIC Inc.

株式会社 MGIC