プログラマブルタイマー基板　KP-TMAR08　取扱説明書
応用編(Arduino)

第 1 版　190416

■はじめに

この説明書は「プログラマブルタイマー基板」(KP-TMAR08) の応用編です。
本基板には出荷時にあらかじめ、汎用的なタイマーとしての機能がプログラムされています。
出荷時に書き込まれている機能については、基本編の説明書をご覧ください。

本基板はAVRをコントローラに使ったArduino UNOと機能互換があります。プロトタイピングのスタンダードとも言えるArduino UNOボードに、リレー、フォトカプラ等の回路をあわせて一枚の基板とした構造になっています。
本基板は出荷時からタイマー機能を持っていますが、Arduino UNOボードとしてプログラムを直接製作することによって、結線マクロでは得られない細かな制御を実現する事ができます。

▼プログラマブルタイマー基板 KP-TMAR08 製品ページ
http://prod.kyohritsu.com/KP-TMAR08.html

▼出荷時のプログラマブルタイマー機能スケッチ
出荷時のプログラマブルタイマー機能を実装しているスケッチ(Arduino のプログラム)は、弊社のWebページからダウンロード可能です。出荷時の機能に戻したい場合は、下記ページにて配布のスケッチを本文書で説明する方法により再書き込みし、同ページ配布の結線マクロデータを転送してご利用ください。
また、これらのデータは、本基板で動作させる目的には自由に編集改変してご利用頂けます。
スケッチおよび結線マクロの配布ページ
http://prod.kyohritsu.com/PROGTIMER/index.html

▼ご注意
共立電子産業では、Arduino UNOボードとしてのプログラム作成サポートや作成したプログラムの動作に関するご質問への回答はいたしかねます。
ご理解の上、ご利用ください。

■本基板のハードウェア構成
本基板は、Arduino UNO互換CPUにリレーやフォトカプラ等の入出力回路を1枚のボードにまとめた構成であり、Arduino UNO本体と入出力回路シールドをあわせた機能に相当します。
また、Arduino UNOとして使用するためのブートローダーは出荷時にあらかじめ書き込まれていますので、Arduino IDE(プログラム開発用ソフトウェア)がインストールされたPCに接続して、プログラムを書き換えて機能を変更・拡張する事ができます。

自作のプログラム(Arduinoでは「スケッチ」と呼ばれます)を書き込んだ時点で、出荷時に書かれていた結線マクロを管理するプログラム(これもArduinoのスケッチとして書かれています)は上書きされて消去されます。
出荷時の状態に戻す場合は、弊社製品ページからKP-TMAR08用初期スケッチをダウンロードして、Arduino IDEからコンパイルとKP-TMAR08に書き込みを行ってください。
本プログラム(スケッチ)は本機のIOを利用する際の参考例としてもご利用頂けます。

▼本基板のピン構成一覧
本基板上の各回路と接続ピンの関係は、下記の通りです。

リレー ch0,ch1,ch2,ch3	出力	D2,D3,D4,D5
シリアルポート RXD(受信)	入力	D0
シリアルポート TXD(送信)	出力	D1
ジャンパー T0,T1,T2,T3,M0,M1	入力※1	D6,D7,A4,A5,D8,D9
フォトカプラ ch0,ch1,ch2,ch3	入力※2	D10,D11,D12,D13
半固定抵抗 V0,V1,V2,V3	入力※2	A0,A1,A2,A3

※1 短絡ソケットを外して使う場合は、出力ピンとして使用可能です。
※2 このピンを出力に指定しないでください。

▼純正Arduino UNOと本基板の異なる点
・電源は5V入力専用となります。USB Micro-B コネクタ(CN1)またはネジターミナル(CN12)のいずれかから入力します。電源出力端子はフォトカプラ用電源端子(CN10)にあります。
・Arduino UNOではDIGITAL端子D0～D13とANALOG IN端子A0～A5がありますが、本機では基板上に搭載した回路に各ピンを割り当てているため、下表に示す制約があります。

D0, D1	シリアル入出力(純正のArduino UNOと同等機能です)
D2 ～ D5	リレー出力 ch0～ch3に接続されており、出力専用となります。入力ピンに設定しても故障することはありません。
D6 ～ D9	ジャンパー入力端子 T0,T1,M0,M1に接続されており、Arduino UNOと同様、信号の入出力に使用できます。
D10 ～ D13	フォトカプラ入力ch0～ch3に接続されており、入力専用となります。この端子を出力ピンに設定しないでください。
A0 ～ A3	半固定抵抗V0～V3に接続されており、外部から信号を入れる事はできません。この端子を出力ピンに設定しないでください。
A4, A5	ジャンパー入力端子T2,T3に接続されており、Arduino UNOと同様、信号の入出力に使用できます。

【注意】
KP-TMAR08基板で使用しているネジ端子の上面にある電線取り付けネジは、端子との電気的に接続されません。(ネジ止めの状態によっては接触する場合もあります)
テスター等で電圧を測定する場合は端子のネジ部分ではなく、電線挿入口の金属部分もしくは接続した電線に直接行ってください。

■スケッチ作成のヒント
ArduinoのスケッチはPCを使い、「Arduino IDE」という開発用ソフトウェアを用いて行います。Arduino IDEは、無料でダウンロードすることができます。Arduinoスケッチの開発言語はC++で、豊富な事例がネット上に多数公開されています。また、IDE内部に基本的な機能を説明するサンプルのスケッチが収録されています。

［1. デバイスドライバの組み込み］
本基板はArduino UNO等のマイコンボードと同様、USBシリアルと呼ばれる機構を通してPCと接続します。PCに本基板を接続の間は、シリアルポートと呼ばれる通信用の切り口が作成されます。このポートを用いて、PCから本基板上に通信(プログラムの転送)を行います。

▼Windowsをお使いの場合
・インターネットに接続されている環境で実行してください。
・本基板を接続する前に、「デバイス マネージャー」を起動してください。
・デバイス マネージャーの中に「ポート(COM と LPT)」の項目が存在する場合は展開してください。展開したリスト内に「通信ポート(COM1)」など、COM□(□には番号が入ります) が表示されますので、現在の番号をメモしてください。
「ポート(COM と LPT)」の項目が存在しない場合は次に進んでください。
・本基板をUSBに接続してください。
接続後、デバイスの認識中を表す文言が表示される場合もありますが、数分でデバイスドライバの組み込みが完了します。
・デバイス マネージャーの「ポート(COM と LPT)」内に、先程メモした COM 番号以外に、新たに「USB Serial Port (COM□)」(□には番号が入ります) が増えていれば成功です。
この□に入る番号が、接続したPCにおける本基板のポート番号となります。
・デバイスマネージャの項目に黄色の［?］と「不明なデバイス」の項目が出てきた場合、デバイスドライバの自動組み込みに失敗しています。この場合は手動でデバイスドライバを組み込む必要があります。下記「USBシリアルコンバータのデバイスドライバ」に移り、手動でインストールしてください。

▼Mac、Linuxをお使いの場合
・本基板を接続する前に、「ターミナル」を起動してください。(Linuxでは「端末」という名前の環境もあります)
・現れたターミナル画面に、下記のコマンド文字を入力し、［Enter］を入力して実行してください。

ls /dev/tty*

・画面上に、"/dev/tty"から始まる結果が出力された場合は、その結果をメモしてください。
全く表示されず次の入力待ち表示が出た場合は、そのまま次に進んでください。
・本基板をUSBに接続してください。
・再度ターミナル画面に、先程と同じコマンドを入力し、[Enter]を入力して実行してください。

ls /dev/tty*

・先程メモした結果の項目以外に、新たに"/dev/tty"から始まる項目が増えていれば成功です。
この項目の名前が接続したPCにおける本基板のポート名となります。
・コマンドを実行しても項目が増えていない場合は、デバイスドライバの自動組み込みに失敗
しています。この場合は手動でデバイスドライバを組み込む必要があります。
下記「USBシリアルコンバータのデバイスドライバ」に移り、手動でインストールしてください。

▼USBシリアルコンバータのデバイスドライバ

・本基板は、USBとシリアルコミュニケーションを仲介するUSBシリアルコンバータチップとして、
FTDI 社の「FT231XS」を使用しています。
・ドライバの自動組み込みが行われなかった場合は、下記URLより、各環境に合わせたデバイス
ドライバをダウンロードし、インストールしてください。

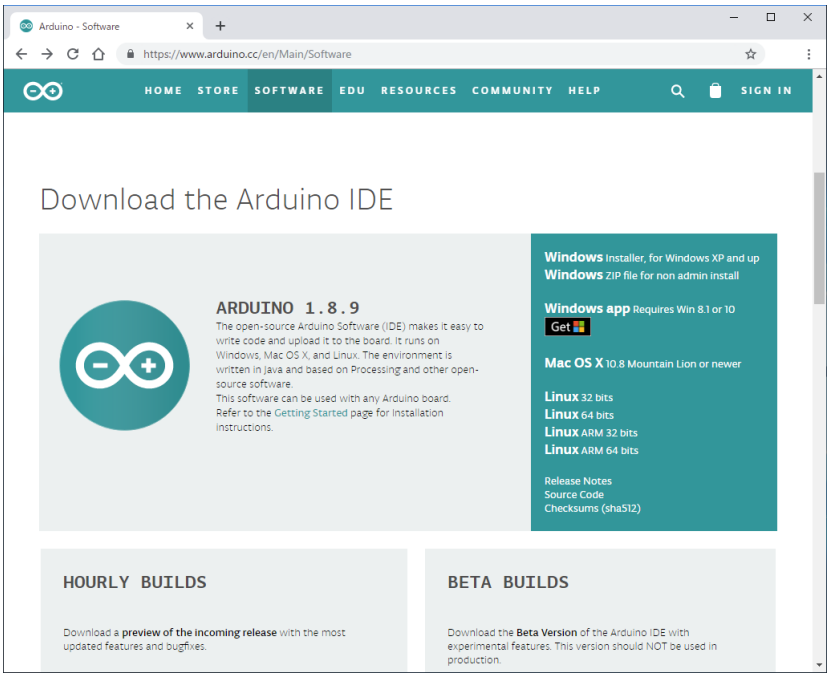
https://www.ftdichip.com/Drivers/VCP.htm

[2. Arduino IDEのインストール]

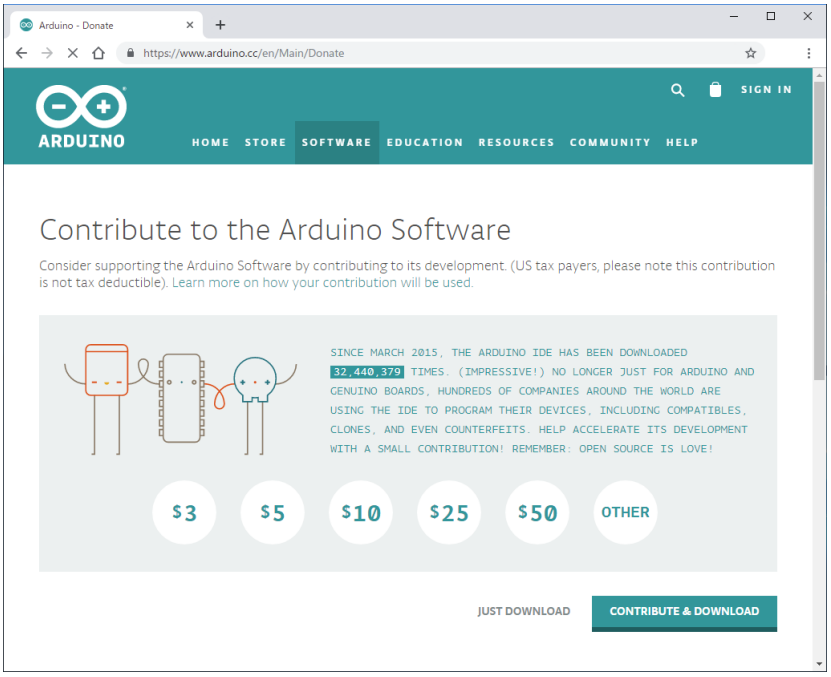
下記の配布ページから、Arduino IDEをダウンロードします。

Arduino - Software

https://www.arduino.cc/en/Main/Software



ページ内「Download the Arduino IDE」の項目をさがし、お使いのPC環境に応じたプログラムを入手します。
(Windows の場合は「Windows Installer, for Windows XP and up」、Macの場合は「Mac OS X 10.8 Mountain Lion or newer」を選択します)



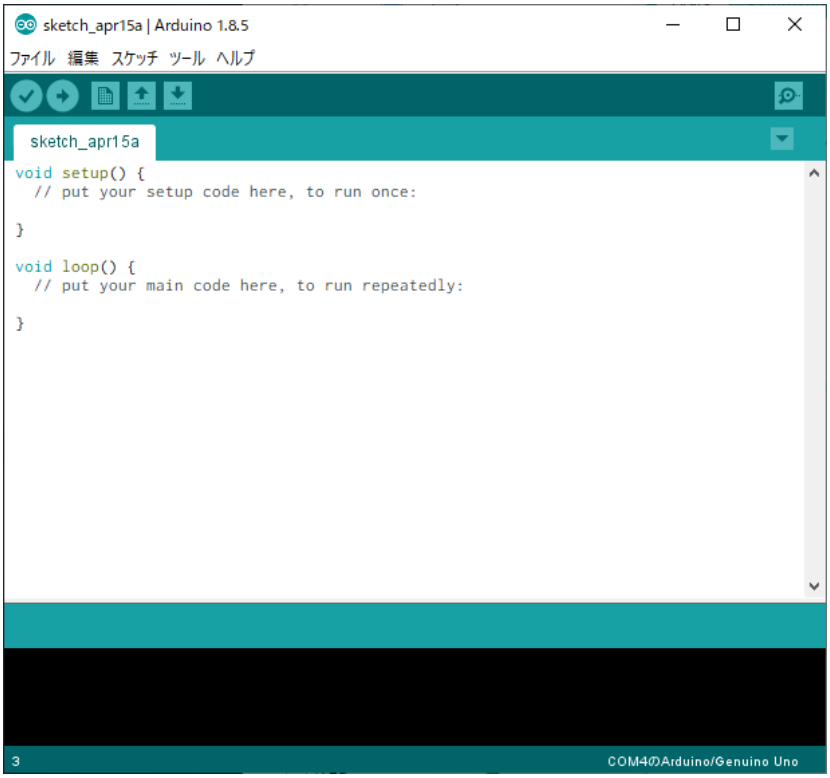
上記の文字をクリックすると「Contribute to the Arduino Software」というページが開きます。
Arduinoの開発チームに対する寄付を募る旨のメッセージが表示されますので、寄付をせずにダウンロードする場合は［JUST DOWNLOAD］をクリックするとダウンロードが始まります。

「CONTRIBUTE & DOWNLOAD」ボタンは、寄付をしたい場合に押すボタンになっています。
寄付は気持ちとして、してもしなくても同じ結果(同じソフトを入手)になりますので、適時判断してください。
ダウンロードが完了すればファイルを開いて、PC に Arduino IDE をインストールしてください。

［3. Arduino IDEの環境］

本基板をPCのUSBポートから外している場合は、接続してください。

インストールしたIDEを起動すると、下記のような初期画面が表示されます。



上部のメニューから「ツール」を選択し、表示されたリストの中の「ボード："……"」(……にはボードの品種名が入ります) という項目が、「ボード："Arduino/Genuino Uno"」となっている事を確認してください。もし他の項目である場合は、「ボード："……"」の上にマウスを重ねると右側に表示されるボードのリストから「Arduino/Genuino Uno」をさがしてクリックしてください。

次に、同じく「ツール」メニュー内に表示される「シリアルポート："……"」という項目が、手順1で組み込んだデバイスドライバのシリアルポート名となっていることを確認してください。

違う場合は、同じくマウスを重ねて右側のリストを表示し、そこから選択してください。リストにも表示されない場合は、再度手順1に戻り、ドライバの組込みが出来ているか確認してください。

以上で準備は完了です。このボードとシリアルポートの設定は、再変更するまで保持されます。

［4. プログラムの作成］

収録済みのサンプルプログラムをベースにして、プログラムを作成してみます。

IDE上部のメニューから「ファイル」を選択し「スケッチ例」にマウスを重ねると右側に項目が表示されますので、スケッチ例 ＞ 01.Basics の順に選び、その中にあるBlinkをクリックしてください。

サンプルスケッチ「Blink」の画面がもう一枚開きます。最初に起動した画面がじゃまな場合は閉じても問題ありません。

このスケッチは、俗に『Lチカ』と呼ばれる、出力ポートに接続されたLEDを点滅させるだけの単純なプログラムですが「基板の動作が正常か」「プログラムが正常に書き込めているか」等、動作を確かめるには最適です。
このプログラムを改変して、リレーをON/OFFするように変更していきます。

▼関数と実行手順
Arduinoスケッチには、setupとloopの2つの関数が必須となります。その他に、自分で関数を追加で作成することもできます。
○setup関数
スケッチの実行開始時に1度だけ呼び出されます。
一般的には、設定や前準備などプログラムの最初に行う処理を配置します。
○loop関数
setup関数を終えた後に呼び出されます。loop関数の終了後は再度loop関数が呼び出されるようになっているため、この関数内に配置した処理は反復されます。一般的には、プログラムの実行部の本体処理を配置します。

▼プログラムの修正
LEDはArduinoの慣例ではD13ピンに接続しますが、KP-TMAR08基板ではリレーのch0が接続されているD2ピンを代わりに指定します。
各リレーとピン番号の関係は下表の通りです。

ch3（CN5）	ch2（CN4）	ch1（CN3）	ch0（CN2）
D5	D4	D3	D2

24行目に、下記の行を追加してください。

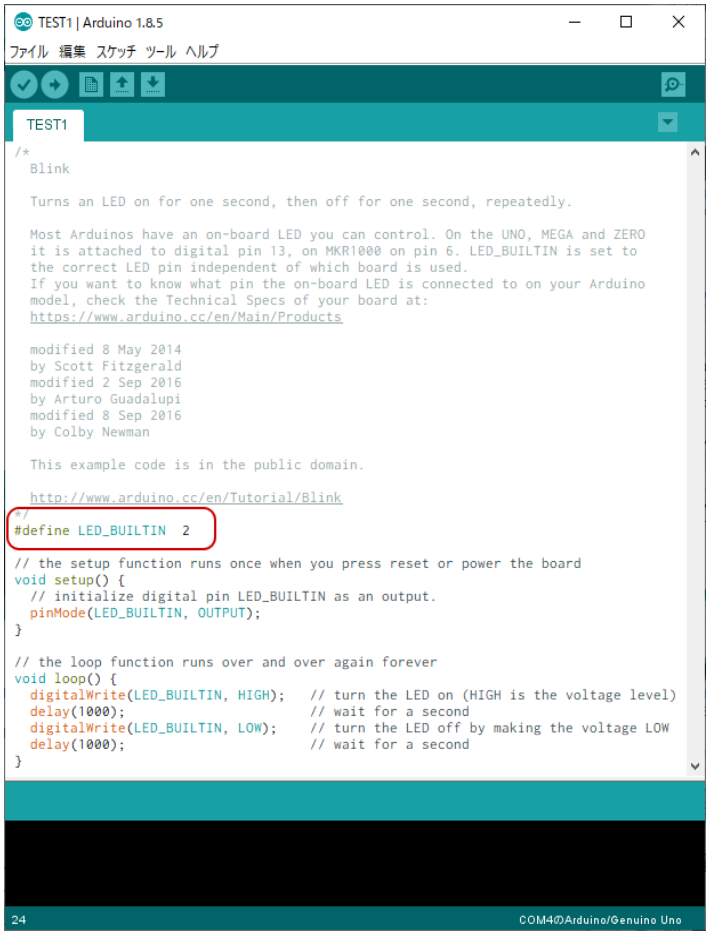
```
#define LED_BUILTIN  2
```

❖図挿入箇所❖
この記述は、「プログラム中にLED_BUILTIN と書かれた場所に2を入れよ」と指示した事になります。2はリレーch0が接続されているピンの番号です。
(D2からDを取った番号として指定します)
以上により、例えば以後のプログラム中に現れる

```
 pinMode(LED_BUILTIN, OUTPUT);
```

という表記は、

```
 pinMode(2, OUTPUT);
```



と記述したのと同じになります。
これを直接 2 と書かずに LED_BUILTIN という名前を付けることは、下記のような利点があります。
○名前があると目的がはっきりする
(LED_BUILTINという表記を見るだけで、内蔵のLEDに関するものであることの推測につながります)
○多数の場所にある数字を一括で変更することができる
例えば、リレー ch0(2番ピン)を ch1(3番ピン)に変更したい場合には、

```
#define LED_BUILTIN  3
```

とするだけで、全箇所のLED_BUILTINを2から3へ変更できます。
※例題として、元サンプルスケッチにあるLED_BUILTINという名前のまま使いましたが、本来は
RELAY_BUILTIN 等の名前にした方がより適切です。

▼基本の入出力関数

入出力の制御を行うための基本となる関数の一部を解説します。
このほかにも多数の関数が用意されていますので、詳細はArduinoの公式リファレンス等を参照してください。

○pinMode

---

pinMode(pin, mode)
  pin：設定を行う対象のピン番号…0 ～ 13 および A0 ～ A5
  mode：入出力設定…INPUT(入力) または OUTPUT(出力)

KP-TMAR08基板でのピンの接続先と番号の関係については、本説明書内「▼本基板のピン構成一覧」を参照してください。ただし、D0 ～ D13は、実際にはDを省略して0 ～ 13と記載します。
【注意】D10 ～ D13 および A0 ～ A3 ピンは、本基板では入力ピンとして使用されています。
出力(OUTPUT)に指定しないでください。

○digitalWrite
digitalWriteは、出力モードのピンに対し、HIGH(5V) または LOW(0V)のいずれを出力するかを指定するための関数です。

---

digitalWrite(pin, value)
  pin: 出力を行う対象のピン番号…0 ～ 9 および A4, A5
  value: 入出力設定…HIGH(5V) または LOW(0V)

※あらかじめ pinMode 関数で出力 (OUTPUT) を設定する必要があります。
※リレー出力(ピン番号 D2,D3,D4,D5)は、HIGH出力でオン、LOW出力でオフとなります。

○digitalRead
digitalReadは、入力モードのピン電圧を読み取り、HIGH(5V) または LOW(0V)のいずれであるかを結果として返す関数です。

---

digitalRead(pin)
  pin：入力を行う対象のピン番号…0 ～ 13 および A0 ～ A5
  戻り値 : HIGH または LOW

※あらかじめpinMode関数で入力(INPUT)を設定する必要があります。
※入力電圧が約1.5V以下(≦0.3Vcc)の時に 0、約3.0V以上(≧0.6Vcc)の時に1となることが保証されます。その中間の電圧では不定となります。

[5. スケッチのコンパイルと書き込み]
IDEの画面上部にある[✓(検証)] ボタンをクリックすると、スケッチに間違いがないかチェックされます。終了すると画面下部に「コンパイルが完了しました。」の表記と共に、エラーがある場

合は最下部の黒色背景のメッセージ出力内に赤字で表示されます。赤字のメッセージが無ければ正常です。

「ファイル」メニューから「名前を付けて保存」をクリックし、適当な名前を付けて保存を行ってください。このIDEは英語圏のプログラムであるため、日本語や全角文字を含まない名前を指定することをお勧めします。

例えば TEST1 の名前を指定すると「TEST1」フォルダが作られ「TEST1.ino」が中に保存されます(.ino がスケッチの拡張子になります。)

KP-TMAR08基板が接続されているのを確認し、IDE画面上部[➡(マイコンボードに書き込む)]ボタンをクリックしてください。
再度プログラムのコンパイルが行われ、生成されたプログラムがKP-TMAR08基板に自動で書き込まれます。書き込みが終われば作成したプログラムの動作が開始されます。
今回の例ではKP-TMAR08基板のリレー ch0 が 2 秒周期(1秒ON、1秒OFF)で動作を繰り返せば完成です。