

■仕様

電源電圧：DC5V

消費電流：180mA（付属の蛍光管を使用した場合）

表示桁数：9桁 *桁ドライブ出力は10桁対応

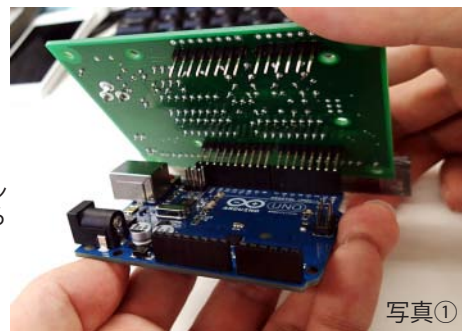
信号形式：ARDUINO 用

■概要

蛍光標示管を使用した、9桁の数字表示ユニットです。Arduino からスケッチを用いて、数字を表示する事ができます。表示管は、ダイナミック表示形式のため、ソフトウェアでダイナミック点灯をコントロールして、表示します。また、表示に必要な高電圧電源も、ARDUINO から出力されるパルスを用いて作成しています。

■ARDUINO に実装

写真の様に、VFD 基板を ARDUINO のコネクタに挿入します。180度反対向きでは、コネクタのピン数が一致しません。(写真①)ARDUINO の種類によっては Digital コネクタの内、2本分が無いものも存在します。



■蛍光標示管について

蛍光標示管は、ヒータを持った真空管で、ヒータ兼用のカソードと、表示桁をコントロールするグリッド、及び電子の衝突で発光するアノードがあります。アノードには蛍光体が塗られており、カソードからの電子の衝突で青緑色の蛍光色で発光します。

またアノードは、「日」の字型表示や特定文字形のセグメントに分割されて、任意に発光制御ができます。ヒータは、熱源のため、LCD 等に比べ、少々消費電力が大きくなる傾向にあります。自発光素子のため、視野角に捉われない美しい表示が行えます。本セットの付属蛍光管は、7セグメントに小数点、カンマ、アポストロフィーを加えた10セグメントを1桁として、9桁分の数字表示が行えます。

また、最右桁には1/2の専用セグメントがあります。

ヒータの電圧は3.5V程度と低いですが、アノードに加える電圧は15V~18Vと比較的高電圧になります。

■回路構成

蛍光表示管を点灯させるためには、ヒータ用の3.5V程度の電圧と、点灯用の17V程度の二つの電圧が必要になります。構成方法としては、17V位の比較的高い電圧から、ARDUINO用の5Vと、ヒータ用の3.5Vを作る方法と、逆に5V供給からヒータの3.5Vと17Vの高圧を作る方法が考えられます。本セットでは、後者の5Vを電源とし、17Vを作る回路で、電圧を上昇させて、表示管に供給しています。セグメント（アノード）及び、グリッドのドライブ回路については、11個のセグメントと9個のグリッド（10桁表示管なら10個）において、17VをON/OFFしなければなりません。合計で20個（21個）の5V->17Vコンバータが必要になり、また制御側の出力も同数必要になる事から、ARDUINOのポート数では足りないため、シリアル入力、パラレル出力のシフトレジスタを使用し、このシフトレジスタの電源電圧を17Vとする事で、必要な電圧変換回路の個数を削減しています。さらに、少々トリッキーな回路構成を採用した事で、電圧変換回路を1個に削減しています。

注意

本セットのシフトレジスタには、CMOS系4094が、表示デコーダに、同じくCMOS系の4028を使用しています。

これらCMOSのICですが、使用できる電源の最大値が15Vのものとして18Vのタイプが市販されています。

本器では、17Vの電圧で使用できる様、18Vの耐圧を持ったメーカーの製品を使用しています。

何らかの理由で、ICを交換される場合は、使用電圧が18Vになったタイプを選択してください。

*ヒータ点灯について

本来、蛍光表示管のヒータは交流で点灯させます。本セットでは、コストダウンのため、直流点灯としています。

このため、表示の輝度が、右の桁に比べ、左の桁が暗くなります。表示輝度はアノード電圧からカソード電圧を引いた値になります。

ヒータの電圧は、左端が3.5V、右端は0V（GND）になります。ヒータはカソードと兼用されているため、左端は17V-3.5V=13.5V、右端は17V-0V=17Vとなり、右に行くほど表示電圧が高くなって、明るく発光する事になります。

（静電界防止のため、ヒータ右端は、VFD管の表示面ガラスの裏に塗布された透明の導電膜と接続されています。GND電位が理想と判断し、ヒータの右側をGND、左側を3.5Vにしています）

*電圧表記について

高圧側の電圧は、極力明るく表示させるため、17.6V程度を目標に、回路を構成しています。

表記上、この電圧を17Vと記載しています。またヒータ電圧は、5Vから整流ダイオード2本による電圧降下を利用しているため、大体の値が3.5Vになります。

*高圧電源の生成

高圧側、約17Vの電源は、DC5Vを電源とし、トランジスタスイッチでコイルに流れる電流をON/OFFして、OFF時のバックパルスを利用した昇圧回路で電圧を持ち上げています。回路的には、殆どの部分がハード構成ですが、少しソフトが介在しています。

昇圧回路の構成上、トランジスタをON/OFFするパルスが必要ですが、省力化のため、この部分はソフトを使用しています。（ソフトを使用しない、自励発振回路を追加しても動作します）

予め設定した電圧に到達すると、ハード的にスイッチ動作が止まる様になっていますので、ソフト部分は単にパルスを出し続けているだけです。任意の輝度に設定できる様、高圧側の電圧をA/Dコンバータで測定して、フィードバックさせる様な構成も考えられます。（測定回路の追加、ソフトの作成が必要です）

*電源について

電源には、安定化された5Vをご用意ください。5V出力のスイッチング式ACアダプタが最良です。

一方、ARDUINO上にも、5Vの安定化回路が搭載されており、7V~8VをARDUINOのDCジャックに接続する事で、本セットにも電源供給を行う事ができます。しかしながら、本セットの消費電流が比較的大きいため、ARDUINO上の安定化回路の発熱が多くなってしまいます。消費電流は許容値内ですが、発熱の関係で、ARDUINOのDC端子に電源供給する場合は7V~8Vの間にしてください。

ARDUINOの最大許容電圧、12Vを加えても、直ちに壊れるわけではありませんが、30秒程度で、発熱による温度上昇が、上限値に達してしまいます。なるべく、本セットのDCジャックにDC5Vを供給する様にしてください。

*USBからの電源供給について

標準仕様のUSB端子からの供給電力に不足はありません。が、機構上、電圧電圧が5Vから低下してしまいます。

結果的に、高電圧が十分に得られず、表示輝度の低下や表示ムラが出る場合があります。

（USB端子の供給電圧に依存します）

輝度ムラが発生する事がある点を除いて、ソフト制作中の動作実験には、問題なく利用できます。この場合、本器のDC端子に電源を供給しなくても、USB端子からの電源で動作します。

■基本プログラム

*ライブラリに付いて

本器のスケッチは、タイマー割り込みとして「TimerOne」を利用しています。

最新のバージョンは「<http://code.google.com/p/arduino-timerone/>」から入手できます。

2013/4 現在、同ページの下行にある「<http://arduino.cc/playground/Code/Timer1>」リンクの先を開き、「Download -> TimerOne Google Code download」から「TimerOne-v9.zip」をダウンロードできます。

zip ファイルを展開後、生成されたフォルダを arduino をインストールしたフォルダの中の libraries フォルダにコピーします。windows の場合、arduino のインストールフォルダは、arduino のショートカットアイコンを右クリックしたメニュー内から、プロパティを表示し、「リンク先を探す」または「ファイルの場所を開く」ボタンで表示する事ができます。

MAC の場合、

Arduino アイコンを右クリック (dock 内では無くアプリケーションフォルダ内内)>「パッケージの内容を表示」をクリック
>Contents>Resources>Java>libraries> の中にダウンロードしたファイルをコピーします。

ソフトの構成。

VFD 数字表示ユニットは、ハードを簡略化するため、ソフトによる駆動で補いを行っています。

VFD を直接駆動しているのは、シリアル入力、パラレル出力のシフトレジスタです。

この表示用シフトレジスタには、データ端子、クロック端子、ラッチ端子の3つの端子があり、それぞれに信号が必要です。データ端子に入力されている信号を、シフトさせながら取り込むのがクロック端子です。

ラッチ端子は、シリアルシフトレジスタに揃ったデータ (本器仕様では 16bit) のデータを並列ラッチに取り込む仕事をします。

動作は、「データを用意->クロックを出す->16 回前記の作業をくりかえしたら、ラッチ信号を出す」となります。

ただ、3 個の端子用の信号は 17V 程度の論理レベルが必要なため、ARDUINO が扱う 5V の論理振幅をトランジスタ等を用いて 17V 程度の振幅に変換しなければなりません。

本器はこの部分を大幅に省略して、1 個のレベル変換器で済ませています。

それでは、3 つの信号はどうやって区別しているのかとなりますが、それぞれ、駆動パルスの時間が異なるため、時間中で判別させています。具体的には、抵抗とコンデンサを利用した積分回路を使用して区別を行います。クロック端子はレベル変換回路と直接接続されています。データ端子は 20 μ S 程度の遅れを持った積分回路、ラッチ端子は非対称な応答時間を持った積分回路を経由して接続されています。ソフト上では、レベル変換器を 50 μ S 周期の割り込みタイミングで駆動します。

割り込み周期毎にデータとクロックを合体した信号を生成し変換回路に出力します

(レベル変換器は、エミッタ接地のトランジスタスイッチですので、ここを通過すると信号の論理が反転します)

データが H の場合は L を出力、50 μ S 経過、信号を一瞬 H にし、即 L に戻し、H->L の変化点で、データ端子の H レベルがシフトレジスタに取り込まれます。レベルシフト回路を通過した信号が、L になっている時間が短いと、積分回路が入っているデータ端子は H レベルを維持したままとなります。一方データが L の場合は H を出力、50 μ S 経過、H->L に変化させる事で、L データがシフトレジスタに取り込まれます。ラッチ信号は、クロック、データの変化を L レベルで停止させる事で、レベルシフト回路の通過後では、200 μ S 程度経過すれば H レベルに到達する事を利用します。

この様な動作を行うため、タイマー割り込みを止める事ができませんが、割り込み内で、処理されるため、表示関係の処理を気にする事なく、メインルーチンは作業に専念する事ができます。

メインルーチンからは、中継に使用する表示メモリにセグメント表示用のデータを入れるだけで、後は自動的に VFD に表示させる事ができます。

■スケッチ上の VFD 割り当て

VFD の 7 セグメントと上の' 記号、下の、記号と右側桁の 1/2 記号のメモリ割り当て情報です。

44 行目付近にある、

```
volatile uint16_t seg_data[10];
```

が表示用のメモリです。このメモリに内容を書き込むと、割り込みルーチンが、VFD に内容を転送します。

このため、スケッチとして、VFD に何かを表示させる場合は、このメモリにデータを書くだけの作業となります。

なお、表示が 9 桁にも係わらず、10 桁分 (seg_data[0] ~ seg_data[9]) 存在するのは、10 桁表示の VFD も使用可能とするため、本シールドでは 10 桁目 (seg_data[9]) は使用せず、seg_data[0] ~ seg_data[8] の 9 個分が表示データとなります。

(割り込みルーチン内では 10 桁分のデータを順次送信しています。volatile uint16_t seg_data[10]; を [9] に変更しないで下さい)

seg_data[?] は 16bit の変数ですが、使用しているのは 11bit 分だけです。

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
---	---	---	---	---	seg	seg	seg	seg	seg	seg	seg	seg	seg	seg	seg
					1/2	,	'	.	g	f	e	d	c	b	a

表示位置は左の桁から右の桁に向かって順に、seg_data[0]、seg_data[1]... seg_data[8] に対応します。

なお、セグメント 1/2 は一番右の桁、seg_data[8] にしか存在しません。

表示用メモリは論理 1 を書き込むと該当セグメントが点灯、0 を書き込むと消燈します。

セグメントデータですので、数字の形に表示させたい場合、数字からセグメントの配置に変換しなければなりません。

変換用として先頭付近にある配列

```
const uint16_t seg[]={
```

を宣言しています。

数字 -> セグメントパターン変換ですので、データを作成すれば、色々な形 (H とか L との形) を表示する事ができます。

もちろん、seg_data[?] に直接表示データを組み立てても問題ありません。

■ | / O の割り当て

ARDUINO の Digital、12 ~ 9 を使用しています。

12 番：出力：表示データ+クロック+ストロブ信号

11 番：出力：DC-DC コンバータの駆動信号

10 番：入力：押しボタンスイッチ S1

9 番：入力：押しボタンスイッチ S2

■VFD表示シールドを使用した、時計プログラム

VFD数字表示器を利用した時計プログラムです。

時刻の基準は、20KHzの表示割り込みから得ていますが、ARDUINO UNO R3 のクロックの精度が悪く、時刻の進行をソフト的に補正して、時刻表示としています。桁数が9桁あるため、1/100秒から表示を行っています。

時刻をカウントする部分は「1/100秒」から「時」まで、ソフト的にカウントするだけの単純なものです。

得られた時刻データはセグメントデータに変換して、表示用メモリに書き込むだけの作業で時計になります。

しかしながら、実際の時計プログラムは少々長くなっています。

殆どの部分を、押しボタンを使った現在時刻の修正とARDUINOのクロック補正用データの修正が占めています。

現在時刻の修正は、ボタン押しの検査と、カウントのインクリメントを、時と分について行うだけの作業ですが、どうしても時計部分よりコードが長くなります。

時刻の進行を司る、1/100秒信号を作成している部分も割り込み周期を分周するだけですが、少し説明が必要と思います。

割り込み周期20KHzから1/100秒(100Hz)の信号を作るので、200で分周するカウントを行います(20,000÷200=100)

一方、1/100秒が作られる度に、補正データを加算し続ける16bitの変数を用意します。

この変数に、補正データを加算して、32767を超える(16bit目がHになる)瞬間だけ、200で分周する値を201で分周する様に切り換え、Hになった16bit目はクリアします。これにより1秒当り、補正データ÷32768×100×50μSの修正が入ります。

この修正により、1/100秒(10ms)が少し長い区間が存在する事になります。(正確には10msは補正の回だけ、10.05msになります)

この方式の利点は、補正を細かく設定できる事です。

時計ソフトの使い方。

基板上的スイッチ S1,S2を使用して、時刻修正を行います。

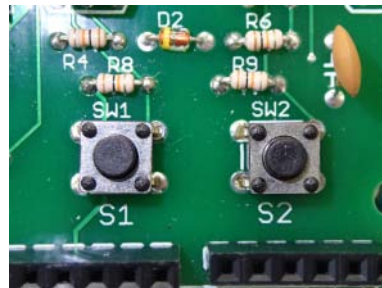
・定常表示状態

S1:短押:未使用

長押:時刻修正

S2:押ししている間、秒、1/100秒をリセット。

*S2の秒1/100秒リセットは、29秒切り捨て、30秒切り上げ機能があります。



・時刻修正状態

定常状態でS1を長押しする事で、時刻修正状態になります。

修正状態になると、時刻の進行が停止し、「時」の表示が点滅表示となります。S2ボタンを押すと、「時」の数値(またはAM/PM表示)が+1されます。

「時」の修正後、S1を押すと「分」の桁が点滅表示となります。S2ボタンで「分」の数値が+1されます。

S1ボタンで修正が終了、通常の時刻表示に戻ります。

(注意:時刻修正は、時刻の進行がHOLDされます。何も修正しなくても、修正モードに入っている間は、時刻が止まったままになります)

時刻修正で、秒、1/100秒の合わせこみは、通常表示状態でのS2によるリセットをご利用ください。

・起動オプション

S1またはS2または両方を押したまま、電源を加える事で、動作設定を行う事ができます。

⇒S1またはS2のどちらかを押した状態で、電源ON

S1を押して起動:24時間表示に設定します。VFDに24Hと表示されるまで、S1ボタンを押したままとしてください。

S2を押して起動:AM/PM表示モードになります。VFDに12Hと表示されるまで、S2ボタンを押したままとしてください。

表示が出れば、ボタンを放してください。通常の時刻表示になります。

24H表示とAM/PM表示の区別は、ARDUINOの内部(CPUチップAVRのEEPROM)に記録されます。

以後、操作で表示状態を変更するまで、電源からの起動は、設定した状態に従います。

⇒S1とS2の両方のボタンを押した状態で、電源ON時刻カウントの遅れ進みを補正する数値を編集するモードです。

VFDにAdjと右の方に数値が表示されるまで、両ボタンを押したままとしてください。表示後、ボタンを放すと、数字が点滅状態になります。

数字はS1で数字が大きくなる方向、S2で数字が小さくなる方向に修正できます。

また、押したままにする事でオートリピートが働き、連続的に数字を送る事ができます。

修正完了後、通常の表示になるまで、触らずに置いてください。約15秒で、時刻表示になります。

通常の時刻表示になる時に、設定(修正)した数値が、ARDUINOの内部(CPUチップAVRのEEPROM)に記録されます。

通常の時刻表示状態では、ここの補正数値に従って、補正しながら時刻をカウントします。

数値の値について。

プログラムは、遅らせる方向(自然状態でARDUINOの進行が早い=時計が進む)に補正する様になっています。

プログラムを作成した固体では、3370程で進みが0になりました。

初期値は3000としています。遅れ進みをちょっと長い期間で測って、1日(24時間)当りの秒数に換算後、75.85を掛け算して、補正数値の修正を行ってください。数字を増やすほど、遅くなります。

また、進みや遅れが激しいほど、1秒/日の補正数値は、75.85から外れて行きます。

*ARDUINOのクロックは水晶発振を使用していない様です。このため、周辺温度の変化で遅れ進みが変動する可能性があります。

時計の時刻カウントは、ARDUINOの発振クロックから得ているため、発振周波数の安定度が直接影響します。

*ARDUINOのロットによっては、CPUのクロック周波数が早い(=進む)のではなく、遅い(=遅れる)事があるかも知れません。

その場合は、時計プログラムの割り込みルーチン内にある、1/100秒を作る数値、通常値200、修正値201を、通常値200、修正値199にした後に

ARDUINOに書き込みます。タイマー割り込みは、`void timerIsr()` ですが、数字の宣言自体は、先頭寄りにあります。

```
#define Adj_SPC_VAL 201 //Clock Adjust speal vaule
```

これを、

```
#define Adj_SPC_VAL 199 //Clock Adjust speal vaule
```

としてください。

VFD シールドを使用して時計を作る方法

他の資料は応用例になっていますので、こちらの資料ではVFDシールドを使用し時計を作成する方法を明記しておきます。

- 1.Arduino SDKをダウンロード <http://arduino.cc/en/>
後インストールしておきます。
(詳しくはオンラインマガジン「Arduinoをはじめよう 第2版」をご覧ください)
- 2.下記URLからArduino用プログラムを表示させます。
http://www.technobase.jp/eclib/WONDERKIT/PROGRAM/VFD_CLOCK.ino
メモ帳などテキストエディターにプログラムをコピー&ペーストして保存しておきます。
- 3.VFDを駆動する為に必要なライブラリを入れます。
*ライブラリについて
本器のスケッチは、タイマー割り込みとして「TimerOne」を利用しています。
最新のバージョンは「<http://code.google.com/p/arduino-timerone/>」から入手できます。
2013/4 現在、同ページの下行にある「<http://arduino.cc/playground/Code/Timer1>」リンクの先を開き、「Download -> TimerOne Google Code download」から「TimerOne-v9.zip」をダウンロードできます。
zip ファイルを展開後、生成されたフォルダを arduino をインストールしたフォルダの中の libraries フォルダにコピーします。
windows の場合、arduino のインストールフォルダは、arduino のショートカットアイコンを右クリックしたメニュー内から、プロパティを表示し、「リンク先を探す」または「ファイルの場所を開く」ボタンで表示する事ができます。
- MAC の場合、
Arduino アイコンを右クリック (dock 内では無くアプリケーションフォルダ内)>「パッケージの内容を表示」をクリック
>Contents>Resources>Java>libraries> の中にダウンロードしたファイルをコピーします。
- 4.Arduino SDKを起動しダウンロードした「VFD_CLOCK.ino」を開きます。
(起動時にエラーが出る場合はライブラリフォルダの名前をTimerOneに変更お願いします)
- 5.→(マイコンボードに書き込む)をクリックします。
エラーが出る場合はライブラリがちゃんと入っていない可能性があります。
「スケッチ→ライブラリを使用」の中にTimerOneがちゃんと入っているか確認して下さい。

時計以外のソフトを作られる方へ

VFD基板を使って、カウンタなどのスケッチを製作したい場合、この時計プログラムから、時計部分を削除して、自分のプログラムを追加しても良いですが、時計部分だけ削除するのに手間がかかります。

下記URL

http://www.technobase.jp/eclib/WONDERKIT/PROGRAM/VFD_Skeleton.ino

に、「VFD_Skeleton.ino」の名前で、必要部分だけのスケッチを置いています。
このスケッチを読み込み、自分のコードを追加する事で、手間を省く事ができます。

近日webページ&wikiでも情報公開予定となっています。

アップ次第下記に情報掲載予定ですのでぜひご覧ください。(2013年5月末予定)

<http://blog.livedoor.jp/kyohritsu/>

VFDシールドの使用方法に関して電話サポートなどは行っておりません。

お問い合わせの際はメール wonderkit@keic.jp までご連絡をお願いします。

