

温度補償型高精度水晶発振器搭載

## 超高精度UARTリアルタイムクロックモジュール

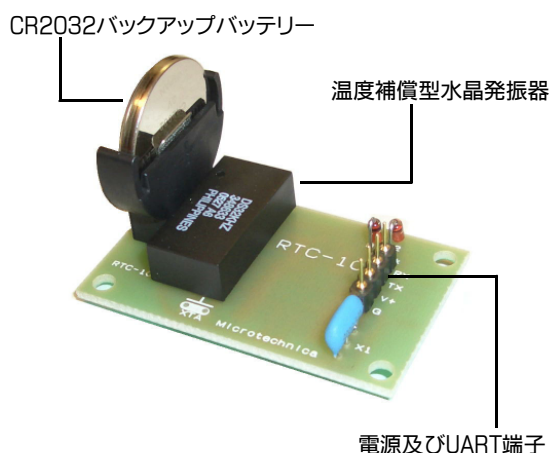
### 取扱説明書

お使いになる前にこの説明書をよくお読みの上正しくお使いください。

(C)2006 マイクロテクニカ

### モジュール本体

#### ■本体上面



### 製品の概要

超高精度UARTリアルタイムクロックモジュール(型式:RTC-10、以下RTC-10と記載)は、動作環境温度0℃~70℃において、約±2.0ppm(約±1分/年)(※)を実現する精度の高いリアルタイムクロックモジュールです。

現在時刻(時、分、秒)及び現在日付(年、月、日、曜日)の情報を非同期シリアル通信(UART)にて簡単なコマンドで取得することができます。

ボタン電池CR2032によるバッテリーバックアップ機能付きですので、電源切断時でも約500時間、正確に時を刻むことができます。

5つの個別登録できるアラーム機能や、毎分ごとに現在時刻を出力する毎分毎時出力、12時間/24時間表示切り替えなど時計アプリケーションに必要な機能を搭載しています。

PICマイコンなどと接続して簡単に超高精度の時計アプリケーションが作成できます。

※時計精度は動作環境周囲温度が0℃~70℃までの時の理論値です。RTC-10はこの精度を保証するものではありません。

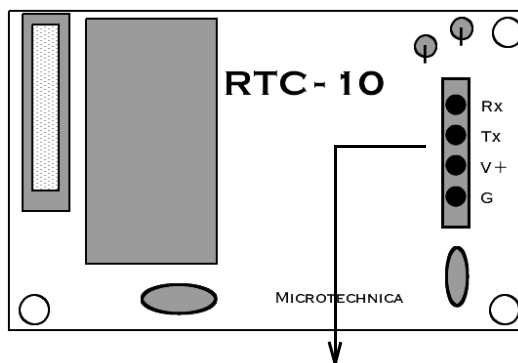
### パッケージの内容

#### ■同梱物

- ・RTC-10モジュール本体
- ・CR2032(バックアップ用バッテリー)
- ・マニュアル(本書)

### 端子の概要と電源について

RTC-10の電源及びシリアル信号の端子は次のようなピン配置になっています。



ピン	端子記号	内容
1	Rx	受信データ (TTLレベル)
2	Tx	送信データ (TTLレベル)
3	V+	+5V電源 (要安定化)
4	G	グランド電位 (GND)

※ピンピッチは2.45mmピッチです。

#### ■電源について

RTC-10の電源電圧はDC5Vです。必ず安定化されたDC5V電源を接続してください。リップルの重畳した電源や高調波が含まれるスイッチング電源などを使用すると時計精度が著しく悪化します。

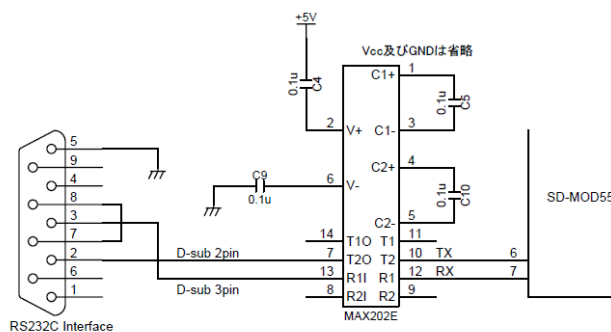
RTC-10では、バックアップバッテリーとしてCR2032を装着しておく、V+端子への電源切断後であっても約500時間、時計機能が動作します。

#### ■シリアル通信について

Txピン及びRxピン及びGNDによる2線式のUARTです。フロー制御は行いません。電圧レベルはTTLレベルとなっています。詳しくは次の項をご覧ください。

### シリアル通信について

RTC-10は、TTLレベルの非同期シリアル通信です。TTLレベルですので、電圧レベルは0V-5Vです。PICマイコン等のTTLレベルのデバイスとは直結できますが、パソコンと接続する場合には、必ずRS232Cレベル変換IC(MAX232など)を介してパソコンと接続します。下記にパソコンのRS232Cポートと接続する場合の参考回路図を示します。



なお、GNDは必ずRTC-10の電源GNDと共通にしてください。

### ■データプロトコル

RTC-10のプロトコルは下記の通りです。

- ・通信速度: 4800bps (変更できません)
- ・データ長: 8ビット長
- ・パリティ: なし
- ・ストップビット: 1

### ■PICマイコンとの接続について

PICマイコンなどのTTLレベルのマイコンと接続する場合、RTC-10のTX及びRXピンはマイコンに直結できます。

通信速度・データ長・ストップビット・パリティの設定を上記のとおり行うことで通信が可能となります。TXピンは送信データ、RXは受信データです。

## RTC-10の基本的な使い方

RTC-10は、各種操作をシリアル通信で行います。コマンドをRTC-10に送信すると、そのコマンドに対応する結果が返ります。

本項では最初に現在時刻の確認と、現在時刻の設定を行う手順を説明します。コマンドの詳しい内容については「制御コマンドについて」の項をお読み下さい。

### ■現在時刻・日付を確認する

RTC-10では工場出荷時に日本標準時に合わせております。まずは現在時刻と日付を確認します。

現在の時刻を取得するコマンドは、"t"(0x74)です。

RTC-10は、各種制御コマンドを送信した後、必ずキャリッジリターン(0x0D)を続けて送信しなければなりません。キャリッジリターンが送信されないとコマンドは受け付けられません。

パソコンと接続している場合には、ハイパーターミナルなどのシリアル通信ターミナルからコマンドを送信します。

※{cr}はキャリッジリターン(0x0D)を意味します。

```
t{cr}
```

上記のようにコマンドを送信すると下記のように結果が返ります。

```
16:00
```

なお工場出荷時には24時間表示設定となっています。戻り値のフォーマットは "hh:mm" です。

次に日付を確認してみます。日付を取得するコマンドは"d"(0x64)です。

```
d{cr}
```

```
061001
```

現在が2006年10月1日の場合には上記のように日付が返ります。戻り値のフォーマットは、"yymmdd" です。

### ■現在時刻の設定

現在時刻を設定したい場合には、"w"コマンド(0x77)を使用します。なおwコマンドで時刻を設定する際には、必ず24進法での指定となります。また時刻だけの設定はできません。必ず日付の設定も同時に行います。

wコマンドは対話形式になります。時→分→秒→年→月→日→曜日の順番で1つずつ、設定していきます。

下記に2006年12月1日(金)、6:15に設定する場合の例を示します。

```
w{cr}
hour(24)=
```

wコマンドとキャリッジリターンを送信すると、上記のように入力待ち状態となります。ここで"時"を24進法で必ず2バイト(2文字)で入力してキャリッジリターンで送信します。

```
06{cr}
OK
min=
```

時刻は6時15分ですので"06"とキャリッジリターンを送信します。

正しいフォーマットで送信すると"OK"が返ります。

続いて"min="と文字列が返りますので分を2バイト(2文字)で入力してキャリッジリターンで送信します。

```
15{cr}
OK
sec=
```

同様にして、秒を2バイト(2文字)で入力してキャリッジリターンで送信します。15分びつたりから始める場合には0秒ですので00を送信します。

```
00{cr}
OK
year(yy)=
```

次に年を下2桁の2バイト(2文字)で送信します。

```
06{cr}
OK
month=
```

同様にして月も2桁の2バイト(2文字)で送信します。

```
12{cr}
OK
date=
```

同様にして日を2桁の2バイト(2文字)で送信します。

```
01{cr}
OK
day(1-7)=
```

最後に曜日を1~7の数値で設定します。1は日曜日始まり、2は月曜日、3は火曜日・・・として7は土曜日になります。

なお、この数値を入力してキャリッジリターンで送信した直後から時間計測が開始されます。正確な時計などを参照しながら6時15分00秒になった瞬間に曜日の数値とキャリッジリターンを送信します。

```
6{cr}
OK
Complete
```

金曜日なので6を送信します。キャリッジリターンを送信すると、OKに続き"Complete"という文字列が返ります。Completeの文字列返れば設定完了です。tコマンド及びdコマンドで正しく日時が設定されているかを確認下さい。

## 制御コマンド一覧

### ■RTC-10のコマンドフォーマット

RTC-10では、コマンドは必ずキャリッジリターン(0x0D)で終端して送信します。コマンドの値の後に必ずCRを送信してください。キャリッジリターンを送信しないとコマンドは実行されません。

RTC-10から返るデータの最後には、キャリッジリターンが付加されています。

RTC-10の制御コマンド一覧を下記に示します。

コマンド <sup>a</sup>	内容	使い方
t	現在時刻を取得	hh:mm 形式
h	現在時を取得	hh 形式
m	現在分を取得	mm 形式
s	現在秒を取得	ss 形式
d	日付を取得	yyymmdd 形式 ※yyは年の下2桁
y	年を取得	yy 形式 ※yyは年の下2桁
o	月を取得	mm 形式
a	日を取得	dd 形式
b	曜日を取得	ww 形式 (01が日曜日→07が土曜日)
p	進法設定	引数 "a"→24進法、"b"→12進法
w	現在時刻日付設定	24進法で現在の時刻・日付を設定
g	アラーム設定	最大5バンクまで登録可能
v	アラーム内容確認	全5バンクのアラーム内容を返す
n	アラーム有効設定	アラームの有効/無効を個別に設定
e	毎分出力設定	分が更新されるごとに現在時刻を出力

コマンドの詳しい使い方は次の項をご覧ください。

## 制御コマンド

※{cr}はRTC-10に対して送信するキャリッジリターン(0x0D)を、(cr)はRTC-10が出力するキャリッジリターンを示します。またRTC-10に送信するコマンドは斜体で表示されています。

### ■ t コマンド

【動作】 現在時刻を hh:mm 形式で出力する

【解説】 pコマンドであらかじめ設定した12進法又は24進法で時刻を出力します。

【例】

```
t{cr}
18:55(cr)
```

### ■ h コマンド

【動作】 現在時を hh 形式で出力する

【解説】 pコマンドであらかじめ設定した12進法又は24進法で時を出力します。

【例】

```
h{cr}
18(cr)
```

### ■ m コマンド

【動作】 現在分を mm 形式で出力する

【解説】 現在の分を2バイト(2文字)で返します。

【例】

```
m{cr}
55(cr)
```

### ■ s コマンド

【動作】 現在秒を ss 形式で出力する

【解説】 現在の秒を2バイト(2文字)で返します。

【例】

```
s{cr}
51(cr)
```

## ■ d コマンド

【動作】 日付を yymmdd 形式で出力する

【解説】 yyは年の下2桁として、計6バイト(6文字)で返します。

【例】  
`d{cr}`  
`061001(cr)`

---

## ■ y コマンド

【動作】 年を yy 形式で出力する

【解説】 yyは年の下2桁で返します。

【例】  
`y{cr}`  
`06(cr)`

---

## ■ o コマンド

【動作】 月を mm 形式で出力する

【解説】 月を2桁で返します。

【例】  
`o{cr}`  
`10(cr)`

---

## ■ a コマンド

【動作】 日を dd 形式で出力する

【解説】 日を2桁で返します。

【例】  
`a{cr}`  
`09(cr)`

---

## ■ b コマンド

【動作】 曜日を ww 形式で出力する

【解説】 曜日の数値を2桁で返します。日曜日から01で始まり、土曜日が07となります。

【例】  
`b{cr}`  
`01(cr)`

## ■ p コマンド

【動作】 12進法・24進法のどちらにするかを設定する

【引数】 12時間表示→b、 24時間表示→a  
※引数は、RTC-10から"24h(a) or 12(b)="という文字列が返ってから送信します。

【解説】 RTC-10の時計表示の進法を設定します。pコマンド送信後に"24h(a) or 12(b)="という文字列がRTC-10から返ります。24時間表示にしたい場合には"a{cr}"を、12時間表示にしたい場合には"b{cr}"を送信します。  
ここで設定した表示方法は、電源及びバックアップバッテリーを切断しても記憶されます。

【例】 12時間表示に設定する場合

`p{cr}`  
`24h(a) or 12(b)=`  
`b{cr}`  
`Set 12hours(cr)`

【補足】 12時間表示に設定した場合であっても、wコマンドによる現在時刻の設定及びgコマンドによるアラーム時刻の設定では、必ず24進法で入力する必要があります。  
これは午前・午後の区別を明確にするためです。

## ■ w コマンド

【動作】 現在時刻・日付・曜日を設定する

【解説】 現在の時刻、日付、曜日をそれぞれ設定します。対話形式での設定となり必ずRTC-10から文字列が返ってから引数を送信します。

wコマンドで時刻を設定する際には、必ず24進法で指定します。また設定値が1桁の場合には、曜日の設定を除きすべての設定において2桁で入力します。(10の位に0を付けます)

設定は、時→分→秒→年→月→日→曜日の順番になり、最後の曜日を送信した直後から時計計測が始まります。

曜日は日～土を1～7の数値で表します。日曜日が1、月曜日が2・・・となり土曜日が7に対応します。

【例】 2006年11月1日(水)、15:30に設定する場合

```
w{cr}
hour(24)=
15{cr}
OK(cr)
min=
30{cr}
OK(cr)
sec=
00{cr}
OK(cr)
year(yy)=
06{cr}
OK(cr)
month=
11{cr}
OK(cr)
date=
01{cr}
OK(cr)
day(1-7)=
4{cr}
OK(cr)
Complete(cr)
```

Completeという文字列が返った瞬間から時間計測が開始されます。

## ■ g コマンド

【動作】 アラーム時刻を設定する

【解説】 RTC-10は、最大で5つまでアラーム時刻を設定でき、個別にアラームの有効/無効を設定できます。アラーム時刻は必ず24進法で設定します。

アラームは下記の手順で設定します。

- ① gコマンドを送信すると、"ALM\_No(1-5)=" という文字列がRTC-10から送信されますので、最初にアラーム番号の指定します。アラーム番号は、1～5までの数値を指定します。
- ② アラーム番号を送信すると、"Time(24hh:mm)=" という文字列がRTC-10から送信されますので、アラーム設定したい時刻を24進法で、hh:mm の形式で送信します。正しい形式で送信すると"OK"が返ります。
- ③ 続いて、"On(a) or Off(b)="という文字列がRTC-10から送信されます。アラームを有効にしたい場合には"a"を、アラームを無効にしたい場合には"b"を送信します。
- ④ アラームが正しく設定されると、"Alarm Data registered"という文字列が返り、設定が完了します。

アラーム設定時刻になると、RTC-10は"armX(cr)"(Xはアラーム番号)という文字列を送信します。

例えば、アラーム番号1に設定した時刻が到来すると、RTC-10は、"arm1(cr)"という文字列を送信し、アラーム時刻になったことを通知します。

アラーム時刻のデータ及び、アラームの有効/無効の設定は、電源及びバックアップバッテリーが切断された後も記憶されます。

なお、アラームは5つ個別に時刻を登録でき、個別に有効/無効を設定できます。5つすべてのアラーム設定の内容を閲覧したい場合には"v"コマンドを、アラーム番号毎に有効/無効を設定したい場合には"n"コマンドを使用します。詳しくは各コマンドの項をご参照下さい。

【例】 アラーム番号2にアラーム時刻16:30を設定する場合

```
g{cr}
ALM_No(1-5)=
2{cr}
Time(24hh:mm)=
16:30{cr}
OK(cr)
On(a) or Off(b)=
a{cr}
On(cr)
Alarm Data registered
```

上記の設定をした場合、16:30になると "arm2" という文字列がRTC-10から返ります。

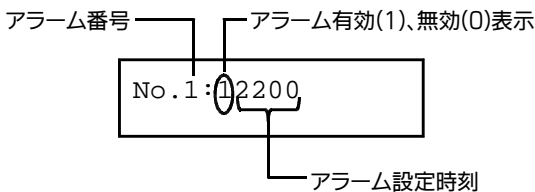
## ■ v コマンド

【動作】 アラーム番号1～5までのすべての設定内容を表示する

【解説】 アラーム番号1～5まで、5つの設定内容を一覧表示します。  
vコマンドを送信すると下記のような内容が返ります。

```
No. 1:12200  
No. 2:01630  
No. 3:00930  
No. 4:01850  
No. 5:12000(cr)
```

アラーム番号に対応した設定内容が表示されます。  
内容は、下記のようなフォーマットになっています。



表示の1万の位はアラームの有効/無効を表しています。  
表示が1の場合にはアラームが有効に、0の場合にはアラームは無効に設定されています。  
例えば、上記の例でアラーム番号2は、アラーム時刻が16:30に設定されており、アラームは無効に設定されています。

【補足】 アラームの有効/無効の設定は、gコマンドによるアラーム時刻設定時の他、nコマンドでアラーム番号毎に設定できます。

## ■ n コマンド

【動作】 指定したアラーム番号のアラーム設定を有効又は無効にする

【解説】 アラーム番号1～5のアラーム設定を個別に有効又は無効に設定するコマンドです。

nコマンドを送信すると、"ALM\_No(1-5)="文字列が返ります。設定したいアラーム番号を1～5の数値で送信します。  
"On(a) or Off(b)="文字列が返りますので、アラームを有効にしたい場合には"a"を、無効にしたい場合には"b"を送信します。正しく設定されると、"On"又は"Off"の文字列が返ります。

※アラームの設定内容は、電源及びバックアップバッテリーが切断されても内容は記憶されます。

【例】 アラーム番号3を無効に設定する場合

```
n{cr}  
ALM_No(1-5)=  
3{cr}  
OK{cr}  
On(a) or Off(b)=  
b{cr}  
Off{cr}
```

## ■ e コマンド

【動作】 毎分時間出力設定をする

【解説】 RTC-10には時計アプリケーションなどを作成する際に便利な毎分時間出力設定機能があります。この機能は、分が更新される毎(毎分)に、現在時間を hh:mm 形式で出力する機能です。

eコマンドを送信すると"Enabled(a) or Disabled(b)="文字列が返ります。この機能を有効にしたい場合には"a"、無効にしたい場合には"b"を送信します。

※設定内容は、電源及びバックアップバッテリーが切断されても内容は記憶されます。

【例】 毎分時間出力設定機能を有効に設定する場合

```
e{cr}  
Enabled(a) or Disabled(b)=  
a{cr}  
Enabled{cr}  
15:30{cr}
```

【補足】 アラーム設定と毎分時間出力設定機能が両方設定されている場合には、アラーム時間が到来すると、先に現在時間を出力し、その後に"armX"のアラーム表示を出力します。  
例えば、9:30にアラーム2が設定されている場合は下記のようになります。

```
09:28{cr}  
09:29{cr}  
09:30{cr}  
arm2{cr}  
09:31
```

## バックアップバッテリーについて

RTC-10には、時間計測を電源切断後も維持するバックアップバッテリーが搭載できます。電池は、CR2032をご使用下さい。その他の電池は使用できません。CR2032にて連続約500時間、時間計測をし続けます。

CR2032を装着する際には、電池の極性に注意してください。電池ホルダー側面に"+"及び"-の印がありますので、極性に注意しながら奥まで押し込んで装着します。

RTC-10では、V+ピンへの電源供給が切断されると自動的にバックアップバッテリーによる給電に切り替わります。再度V+ピンへ電源を供給するとバックアップバッテリーからの給電は停止します。しかしながらV+ピンへの給電中であっても微弱ながら電流を消費していますので、長期間セットしておく、電池の寿命が減りますのでご注意ください。

## 高精度を実現するための設計について

RTC-10には、温度補償型の高精度水晶発振子(TCXO)が搭載されており、この発振子が正確なクロックを作りだし、高精度を実現しています。RTC-10の時計精度は、すべてこの発振子に依存します。

RTC-10に搭載のTCXOは、64秒毎に周囲の温度を計測しその温度から水晶振動子の誤差を計算して発振精度を微調整しています。それにより、動作環境の周囲温度が0°C~40°Cの範囲では、理論的に±2ppmを実現します。

ppmは周波数偏差と呼ばれ、1ppmは100万分の1( $1 \times 10^{-6}$ )です。よって1ppmは1MHzで、1Hzずれることが想定されるという意味になります。RTC-10ではリアルタイムクロックによく使用される32.768KHzのクロックを使用していますので、±2ppmは実時間で1年に±1.05分の誤差が生じることが想定されます。但しこれはあくまでも理想的な環境下で使用した場合の値であり、発振精度は様々な要因で変動し、その変動は直接時間計測の誤差となって現れます。

RTC-10を用いたアプリケーションを作成される際には、できるだけ高精度を保つため下記のような点にご留意の上回路設計をしていただけますようお願いいたします。

### ■急激な温度変化により誤差が大きくなります

RTC-10のTCXOは64秒毎に温度計測をして誤差補正をしています。そのため64秒間に急激な温度変化が繰り返し起こると補正が間に合わず精度が悪くなることがあります。このような現象をサーマルショックと呼んでいます。

### ■電源電圧には安定化された電源を使用してください

RTC-10の電源電圧はDC5Vです。このDC5Vとは安定化された5V電源を指します。ACアダプタやAC-DC回路等によって得られる電源にはリップルと呼ばれる電圧の揺らぎが重畳している場合があります。またRTC-10とグランド電位が共通の回路がある場合、その回路の内容によってはグランド電位の揺らぎが発生する場合があります。こういった電源のリップルや揺らぎはTCXOの発振精度を著しく悪くします。RTC-10に使用する電源は必ず安定化されたものをご使用ください。

### ■電磁ノイズが発振精度に影響を与えることがあります

電子回路、特に周波数の高いデジタル回路から放射される電磁ノイズはRTC-10に悪影響を及ぼすことがあります。周囲に電磁ノイズを発生するような回路や機器がある場合には、金属ケースで周囲を囲うなどの対策が必要になる場合があります。

### ■必要以上のRTC-10へのコマンド送信は控えてください

RTC-10へコマンドを送信すると、それに応じて処理を行います。コマンド処理を行う部分と、時計計測を行う部分は別の回路で処理されていますが、コマンドによっては時計計測部分に影響を与える場合があります。通常の使用では問題ありませんが、例えば現在時刻を取得するコマンドを10ミリ秒間隔で送信し続けるなど、必要以上にコマンドを送信し続けるとRTC-10の時計精度に影響が生じる場合があります。

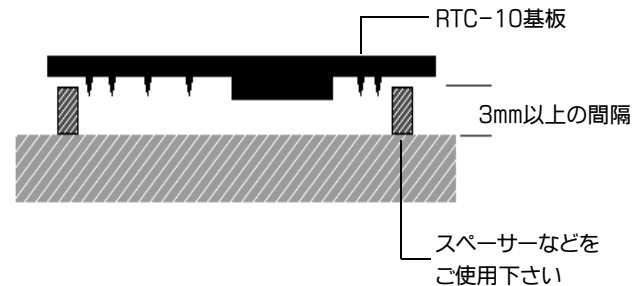
### ■上記以外の要因でも時計精度に狂いが生じる場合があります

その他、本体に振動や衝撃を与えると、TCXO内部の振動領域へ応力がかかり、水晶振動子の特性が変化してしまうことがあります。また、これ以外の様々な要因により時計精度は狂いが生じる恐れがあることをあらかじめご理解頂けますようお願いいたします。

## 実装上の注意

RTC-10を基板に実装する場合や筐体等に固定する場合には、次の点ご注意ください。

RTC-10は部品が両面に実装されております。半田面には面実装部品が実装されておりますので、半田面を下側に固定する際には、必ず高さ3mm以上のスペーサーを挟んで頂き、部品が直接筐体や基板に接触しないようご注意ください。



## 使用上の注意

RTC-10を使用するに際して、必ず下記の注意事項をお守りください。

①RTC-10を使用するに際し、弊社は潜在的な危険が存在する恐れのあるすべての起こり得る状況を予測することはできません。使用する際には、お客様の責任においてこの製品を正しくお使いいただけますようお願い致します。

②RTC-10の信頼性は一般的な使用の範囲に限定されます。本製品を宇宙、航空、原子力、燃焼制御、運輸、交通、各種安全装置、ライフサポート関連の医療機器等のように、特別な品質・信頼性を要求され、その故障や誤動作が直接人命を脅かしたり、人体に危害を及ぼす恐れのある用途向けには使用できません。

③RTC-10では様々な外的要因等(高精度を実現するための設計についての項をご覧ください)によって、時計精度が仕様や理論値を満たせない場合があります。RTC-10を用いた機器等を設計される際にはこの点に十分ご注意ください。設計して頂けますようお願いいたします。また本製品を使用することによって生じた、もしくはこれに関連するいかなる直接・間接損害、懲罰的損害、その他データの破損や消失等を含むいかなる損害についても、弊社では一切責任を負いかねます。あらかじめご理解とご了承頂けますようお願い致します。

④本製品を使用した製品等を製造させる場合には、様々なフェイルセーフ機能(安全設計)を施して頂き、十分に機器のテストをした上で運用されますようお願い致します。

⑤RTC-10は、仕様として周波数偏差の理論値を0°C~40°Cの環境において、±2ppmとしておりますが、RTC-10はこの精度を保証するものではありません。

## 主な仕様

---

電源電圧:	DC5V (要安定化)
消費電流:	+V給電時 1.44mA(typ.) バックアップバッテリー給電時 150 $\mu$ A
動作保証環境:	0 $^{\circ}$ C~40 $^{\circ}$ C
動作可能環境	-20 $^{\circ}$ C~70 $^{\circ}$ C(※1)
時計精度	$\pm$ 2ppm、誤差 $\pm$ 1.05分/年(理論値)
クロック発振子	温度補償型水晶発振器32.768KHz
温度計測インターバル	64秒毎
バックアップバッテリー	CR2032
バックアップ可能時間	約500時間(理論値)
シリアル通信方式:	非同期式シリアル通信(TX及びRX)
信号電圧レベル:	5Vp-p(TTLレベル)
生産国:	日本
外形寸法:	27 $\times$ 45 (mm)

※1:この範囲では誤差精度 $\pm$ 2ppmの仕様は満たせません

マイクロテニカ

〒158-0094 東京都世田谷区玉川1-3-10

(C)2006 Microtechnica All rights reserved