

eDAQ EPU352 バイオセンサー USB isoPod™



- ソフトウェアでコントロール
- USB対応、専用Pod-Vuソフトウェア
- 電気アイソレート式
- 入力レンジ $\pm 20\text{nA} \sim \pm 20\mu\text{A}$
- ゼロ調整オフセットコントロール付き

《概要》

電気アイソレーション式でUSB対応の小型測定器です。PCに接続するだけで、低電流アンペロメトリックセンサーのシグナルを連続的にモニターします。

《使用電極》

この isoPod は、通常の 2 本電極 (作用電極と補助電極) 式低電流タイプのアンペロメトリックセンサーに使用します。また、クラークタイプの酸素電極や一酸化窒素 (NO) 電極にも使えますが、NOや酸素測定にはそれぞれ専用の isoPod も発売しています。

USB端子を持つWindows XP 以降のコンピュータに対応します。専用のPod-Vu、またはシリアルに互換性のある下記ソフトウェア (RS232) を使うことができます：

- Connect™, www.labtronics.com/DI/RS232_Software.htm
- WinWedge®, www.taltech.com/products/winwedge.html
- Tera Term, <http://logmett.com/>

または、LabView、Visual Basic、C++ などを使って独自のソフトウェアを作成します。

《アプリケーション》

アンペロメトリックバイオセンサーを使って連続モニターを必要とする分析化学、生化学、生理学等の研究に便利です。オキシダーゼを固定化した酵素電極では、作用電極表面で酸化反応が起こりオキシダーゼ酵素によって過酸化水素を産出します。その酸化電流を測定します。

《作動原理》

バイオセンサー isoPod は、センサー電極に電解電圧を印加し、その結果作用電極で起こる基質の酸化、又は還元反応から生ずる電流をモニターし、この電流の大きさに比例するアナログ電圧を出力します。

印加する電解電圧は $-2000 \sim +2000\text{mV}$ まで設定できます。酸素電極では -800mV 、NO電極では $+800\text{mV}$ 、オキシダーゼ固定化酵素電極は $+450\text{mV}$ が一般的に用いられる電解電圧です。

この isoPod はUSBネットワークからDC電源を受けますので、ファラデーケージ内でも最小のノイズレベルで測定できます。4 台以上の isoPods を使う場合は、電源付きのUSBハブをお使いください。

《Pod-Vuソフトウェア》

eDAQ Pod-Vuソフトウェアのデモンストレーションバージョンが、付属のUSBメモリスティックの中に入っています。このPod-Vuを購入するとライセンスコードが提供されますので、Pod-Vuの全ての機能が利用できます。

Pod-Vuは自動的にバーチャルシリアルポートを構築し、接続した全てのUSB isoPodにロケーションを割り当てます。PCのUSB端子に接続するだけで、ソフトウェアからコントロールできます。操作の詳細はマニュアルをご覧ください。

Pod-Vuは最大 8 台のisoPodに対応し、使用するセンサーの較正、データの収録、グラフィック表示をリアルタイムで実行します。

《仕様》

入力レンジ (及び分解能):	20 nA (6.25 fA) 200 nA (62.5 fA) 2 μA (625 fA) 20 μA (6.25 pA)
電解電圧設定:	-2000 to $+2000$ mV
電解分解能:	1 mV
オフセットゼロ:	± 20 μA
RMS ノイズ (標準):*	20 pA
入力コネクタ:	BNC
出力データ:	ASCII or 32 bit binary IEEE floating point
出力速度:	100 /s
電源:	< 50 mA、USB 接続から
サイズ (l x w x h):	108 x 58 x 35 mm
重さ:	200 g

eDAQ 社は、上記の仕様を予告なく変更する権利を留保します。

*ファラデーケージ内でレンジ 20 nA、1 Gohm 負荷時。

保証期間: eDAQ 社のハードウェアは 1 年間の品質保証が付いています。

www.eDAQ.jp

E-mail: info@edaq.jp

e-corder® と isoPod™ は eDAQ 社の登録商標です。それ以外は、それぞれの所有者の商標です。



バイオリサーチセンター株式会社 eDAQ事業部
〒461-0001 名古屋市東区泉2-28-24 Tel: 052-932-6421

シリアルコミュニケーション

このisoPod はUSB端子に対応し、Windows、Mac OSX、及び Linux コンピュータのバーチャルシリアルポートとして機能します。Windows XP 以降のコンピュータ用のUSBドライバーと互換性があります。ドライバーは付属のUSBメモリースティック、又は下記からダウンロードできます。

<http://www.ftdichip.com/Drivers/VCP.htm>

isoPod をシリアル(RS232)デバイスとして扱いますので、通信用のソフトウェアを LabView、Visual Basic、C++などを使って作成しコントロールします。

まず、シリアルポートの番号を規定します。Windows コンピュータのシリアル番号 1 と 2 はマザーボード用に割り当てられていますので、isoPod には通常シリアルポート 3 (COM3) かそれ以上の番号を割り当てます。

このバーチャルシリアルポートのコンフィギュレーションを 115200 波特、8 ビット、1 ストップビット、no parity とし flow control は NONE とします。

正しくコンフィギュレートされるとコマンドプロンプト **EPU352>** が表示し、新規コマンドが送信できます。

双方向通信

ターミナルエミュレーションソフト (例えば Tera Term) を使えば、手入力での isoPod と双方向通信ができます：

1. <http://logmett.com> から Tera Term のインストララーをダウンロードする。
2. 不必要なものをインストールしないために、'Compact Installation' オプションを選んでインストールしてください。
3. isoPod を接続し、USBドライバーが必要なら付属の isoPod インストララー USB メモリースティックをコンピュータにつなぎます。
4. Windows のデバイスマネージャーを開き、isoPod とそれに対応する COM ポートを 'Ports (COM & LPT)' リストから見つけてください。どの COM ポートが対応しているかは判りませんが、COM3 以上のポートです。
5. Tera Term ソフトウェアを立ち上げ、Setup メニューの serai port... を選び、上記のシリアルポートを指定します。OKをクリックして Tera Term メインウィンドウに戻し、リターンキーで **EPU352>** プロンプトを呼び出します。

双方向通信が設定できたらシリアル通信が確保されますので、ソフトウェアを書き込む操作を進めてください。

シリアルプロトコル

EPU352> help

EPU352> ?

コマンドリストを返す。

EPU352> set range <r>

ナノアンペアでレンジを設定。<r> = 20, 200, 2000, or 20000.

確認ストリングスを返す。

EPU352> get range

レンジ設定を返す。

EPU352> set Vex <amount>

印加/静止電圧を設定。<amount> ±2.000 V.

確認ストリングスを返す。

EPU352> get Vex

印加/静止電圧をボルトで返す。

EPU352> set offset auto

入力するシグナルが自動的にゼロ nA になるように設定。

EPU352> set offset <amount>

シグナルに <amount> nA (～2000 nA) のオフセットを適用。

EPU352> set offset 0

EPU352> set offset off

適用したオフセットを除去。

EPU352> set units

較正後に使う単位名を設定。例えば ng/L、確認ストリングスを返す。

EPU352> get units

単位名を返す。

EPU352> r

シグナルの測定値を返す。10 Hz(10回/秒)まで繰り返し可能。

EPU352> v

シングル値を返す。測定値は数値として、単位名などの文字は除く。

10 Hz (10回/秒) まで繰り返し可能。

EPU352> sample ascii <freq> [N]

EPU352> sample binary <freq> [N]

周波数 <freq> で測定し、<freq> を1～100Hzの整数で返す。測定値は ASCII か 32 bit binary IEEE 浮動小数点データとして返す。[N] に任意の整数を設定し、それをサンプル数として返す。! でこのモードは終了。<freq> を文字 # にすると、毎回シングル値を # で送信。

EPU352> interval ascii <time> [N]

EPU352> interval binary <time> [N]

ワンサンプルを毎時 <time> 秒で返す。[N] に任意の整数を設定し、設定したそのサンプル数を返す。! でこのモードは終了。

EPU352> cal set <N> <value>

N番目の較正ポイント数 <N> を設定し、測定値は指定した単位で表わされ <value> とする。ただし、<N> は 1～15 の整数。<N> = 1 (一点較正) の時は、ゼロシグナルをゼロと見なし、較正曲線は原点を通る最適化直線とする。

EPU352> cal set <N> <value> <current>

N番目の較正ポイント数 <N> を設定し、測定した値 <current> ナノアンペアを指定した単位の値 <value> で表す。ただし、<N> は 1～15 の整数。<N> = 1 (一点較正) の時は、ゼロシグナルをゼロと見なし、較正曲線は原点を通る最適化直線とする。

EPU352> cal remove <N>

N番目の較正ポイント <N> を消去。それ以後のポイント番号は 1 ポイントずつ減らす。その較正ポイントで再較正する。

EPU352> cal remove all

全ての較正ポイントを消去。

EPU352> cal get

全ての較正情報を返す。

EPU352> res

現行シグナルを読み取り、Vexをこの値で割って抵抗値を算出する。この演算を示すストリングスを返す。

EPU352> version

ファームウェアのバージョン番号を返す。

EPU352> prompt off

'EPU352>' プロンプトを終了。

prompt on

プロンプトに戻す。

スタートの方法

上記のプロトコルに従って独自のソフトウェアを作成する場合は、Ters Term のような端末エミュレーションソフトを使い、各コマンドを手入力での isoPod に送ることから始めてください。期待通りの応答が得られ、プロトコルが変更されても必ず対応できる筈です。