

## 課題情報シート

テーマ名 :	無線消費電力計の製作				
担当指導員名 :	三浦 雅嗣	実施年度 :	23 年度		
施設名 :	関東職業能力開発大学校 附属千葉職業能力開発短期大学校				
課程名 :	専門課程	訓練科名 :	電子情報技術科		
課題の区分 :	総合制作実習課題	学生数 :	2	時間 :	18 単位 (324h)

### 課題制作・開発のポイント

#### 【開発（制作）のポイント】

「見える化」によりパソコンへのデータ転送が必要です。消費電力計の設置場所を自由にするために、無線モジュールとしては情報量の多い XBee® シリーズ 1 を利用しました。マイコンへは UART による通信になりますので、学生への負担が減少します。消費電力の計測では力率対応のために電流と電圧を個別に波形計測しています。

また、Window® アプリケーションの作成と Microsoft® Excel® Object Library の利用技術による Excel® へのデータ表示と自動的なファイル化により確実な「見える化」のシステムを開発しました。

【参考文献】 C# による計測制御プログラミング 金藤仁 著

【学生数の内訳】 システム設計・開発 1 名、電子回路製作 : 1 名

#### 【訓練（指導）のポイント】

開発する項目が多いため、開発するスケジュール管理と、事前の基本知識の習得が必要になります。6 月～8 月には電流計測・電圧計測回路の原理の理解と計測実験、各整流波形の電圧(実効値)計測と電流(実効値)計測を行いました。9 月～12 月には電力計測及び位相差計測、XBee® モジュールによるシリアル通信の無線化、Visual C#® による Windows® プログラミングを行いました。1 月～2 月には Visual C#® によるシリアル通信プログラミング、Visual C#® による Excel® プログラミング、無線消費電力計のシステム構築を行いました。

段階的な技術習得により、学生に興味を持たせて習得させることができました。

### 課題に関する問い合わせ先

施設名 : 関東職業能力開発大学校 附属千葉職業能力開発短期大学校

住所 : 〒260-0025 千葉県千葉市中央区問屋町 2-25

電話番号 : 043-242-4166 (代表)

施設 Web アドレス : <http://www3.jeed.or.jp/chiba/college/>

### 課題制作・開発の「予稿」および「テーマ設定シート」

次のページ以降に、本課題の「予稿」および「テーマ設定シート」を掲載しています。

# 無線消費電力計の製作

千葉職業能力開発短期大学校  
電子情報技術科

**要約** 環境問題とエネルギーの大切さを考えた際に身近にある電気に着目し、PIC®マイコンによる無線消費電力計を製作した。消費電力の計測・表示から発展させ、PCによる遠隔操作とデータ収集を可能にさせた。この消費電力計とPCとの無線化通信においてXBee®モジュールを利用した。また、PC側ソフトウェアではVisual C#®により無線消費電力計と通信処理とExcel®にアクセスするアプリケーションを作成した。見やすさを重視しExcel®でグラフ化し・自動的にファイル化させた。追加機能として、1つのアプリケーションから複数の無線消費電力計へアクセス可能であることを実験レベルで検証した。

## 1 はじめに

現在、環境への配慮と節電の大切さがより指摘されている。これに着目し機器の消費電力の認識で節電意識が深まると判断し、消費電力計の製作を考えた。

消費電力を記録することを考慮して PC への接続を考えた。これにより機器の消費電力の変化を随時知ることが可能であり、表計算ソフトに計測データを随時取込むことでグラフ化が可能である。これによる「見える化」により節電意欲が高まる。

計測対象が離れていても現状のデータを把握できるように、無線化を考えた。無線通信モジュール (XBee®) を利用することで、無線化が容易に可能である。この無線化により、1対1のシリアル通信を1対Nに変更可能である。即ち複数の無線消費電力計を1台のPCで操作、またはデータの記録が可能である。

本製作を通して、専門課程で取得したアナログ回路、電気回路、インターフェイス技術、プログラミングの更なるレベルアップが行えた。

## 2 システム構成

システム構成は、PC側からUSB経由で無線コマンド送信をXBee®モジュールで行い、PIC®側のXBee®モジュールにより無線受信通信を行う。システム構成図を図1に示す。

PCから送信された計測コマンドをPIC側が認識すると、電流、電圧を計測し、消費電力へ演算を行う。消費電力をLCDに表示する。またPCに消費電力を転送する。Visual C#®によりExcel®のセルに消費電力を入力することによ

りグラフ化させた。

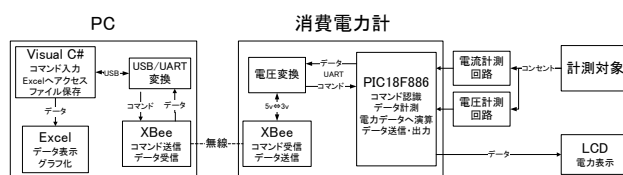


図1 システム構成図

## 3 XBee®について

### 3.1 通信規格

XBee®モジュールはDigiインターナショナル社のXBee®シリーズ1 (XBee 802.15.4 モジュール) を使用した。

短距離無線通信規格としてXBee® (シリーズ1) も「ZigBee®」と呼ばれる場合がある。しかし、ZigBee® Alliance®が認定した製品が正式な「ZigBee®」である。802.15.4部分のみを利用した製品や準拠しただけの非認定製品に対して「ZigBee®」の表現が用いられる事がある。

XBee®は二種類のモジュール (シリーズ1 及びシリーズ2) がある。機能について、図2に示す。

種類	機能	製品
シリーズ1	1対1	IEEE802.15.4
	1対N	
シリーズ2	1対1	ZigBee
	1対N	
	メッシュ接続	

図2 XBee®の種類

シリーズ1は、ZigBee®規格 (シリーズ2) の下位規格IEEE802.15.4だけの製品である。ZigBee®準拠(メ

ッシュ接続)で使用するにはライセンス料の支払い義務が生じる。そのため本製作には、メッシュ接続を使用せず、PC側はコーディネータ(親機)、PIC®側はエンドデバイス(子機)とするスター型を使用するためシリーズ 1 を使用した。XBee®の仕様については以下の通りである。

- ・通信距離 屋内 30m  
屋外 90m
- ・通信速度 最高通信速度 250kbps (2.4GHz 帯)

### 3.2 複数の機器への対応

XBee®はアドレスの指定により 1 対Nの通信が可能である。アドレスの例を図3に示す。今回は動作検証に3台のXBeeを使用した。

機器名	MY 16bit Network Address	Distination Address L
XBeeNo.1(親機)	1	FFFF (ブロードキャストアドレス)
XBeeNo.2(子機)	2	1
XBeeNo.3(子機)	3	1

図3 XBee のアドレス設定

ブロードキャストアドレスに設定された(親機)からのデータ送信は、複数の(子機)でデータ受信される。

1つの(子機)からのデータ送信は(親機)でデータ受信が可能であるが、他の(子機)では受信できない。つまり、(子機)間のデータ送信・受信は不可能である。以上により、PC側を(親機)として(子機)にコマンドを送信することにより、任意の(子機)からデータを取得することができる。

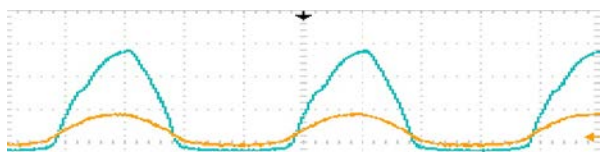
### 4 消費電力の取得方法

電力の演算方法は、データとして電流・電圧・位相差を必要とする。A/D 変換の基準電圧は Maxim 社の MAX6250 で+5V に設定しており、電流計測回路の出力電圧 1V (貫通電流 10A) に対応するために増幅度を 5 倍程度に設定した。各回路の出力波形を図4に示す。電圧計測回路はトランスにより 100V を 8V に変圧し、半波整流から減衰回路により 0.2 倍とした。

また、電力計算は以下の通りである。

$$P(\text{有効電力}) = A \cdot V \cdot \cos\theta \text{ [W]}$$

$$W(\text{皮相電力}) = A \cdot V \text{ [VA]}$$



大振幅が電圧波形 (1V/DIV, 5ms/DIV)

小振幅が電流波形 (1V/DIV, 5ms/DIV)

図4 半田ごての電流電圧波形

皮相電力に位相差を考慮した有効電力を消費電力として表示しなければならないが、最初に位相差のない

機器(半田ごて)を計測対象とした消費電力計を製作した。

電流計測には、電流センサとしてユー・アール・デュー社の CTL-6-P-Z を利用した。このセンサは中央の穴に計測対象の配線を通し、その貫通電流を計測する。使用例を図5に示す。

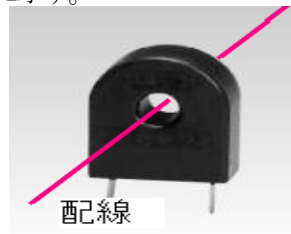


図5 配線例

最大貫通電流は 80Arms であり、センサに接続する抵抗によって出力電圧特性が変化する。出力特性については図6に示す。

電流・電圧の瞬時値の演算は以下の通りである。Aad, Vad は電流電圧の各 A/D 変換値、基準電圧は Vref=5v、分解能=256(8ビット)、Av, Ai (非反転増幅回路の増幅度)、Sva は電流センサの電圧・電流変換係数である。これは PowerMater (YOKOGAWA 製 2533) によって複数の計測対象機器の実効値を計測し、センサ電圧から演算した実効値との補正值である。

$$a = Aad \cdot \frac{Vref}{256} \cdot Av \cdot Sva$$

$$v = Vad \cdot \frac{Vref}{256} \cdot Ai$$

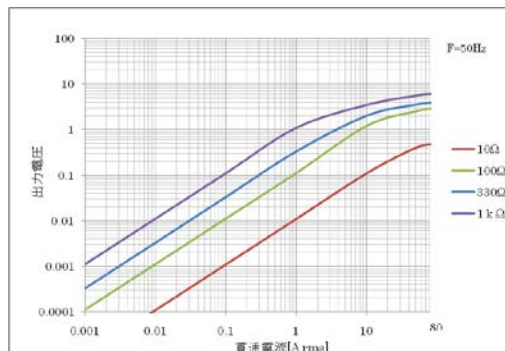


図6 CTL-6-P-Z の出力電圧特性

電圧・電流(電圧レベル)の実効値の取得については、A/D 変換によって得た各瞬時値を二乗平均の平方根のとして下記の演算を行った。この平均においては 10 回の計測データを用いた。

例 1 
$$a_{ave} = (a_0^2 + a_1^2 + \dots + a_9^2) \div 10$$

$$A_{rms\_data} = \sqrt{a_{ave}}$$

電流取得データ(A/D変換後) : a0, a1, a2, a3, a4, a5, a6, a7, a8, a9

更に計測の精度を高めるために 5 回分の実効値の平

均を取り、その値を電流実効値の真値とした。

例 2

$$A_{rms} = (A_0 + A_1 + \dots + A_4) \div 5$$

電流の各実効値：A0, A1, A2, A3, A4

## 5 位相差の取得方法

位相差の取得には、PIC®が持つ2系統のCCPモジュールのキャプチャーモードを使用した。キャプチャーモードとは入力信号が立上り時または立下り時のTimer1の値を取得する機能である。

今回は電流波形 (CCP1) 及び電圧波形 (CCP2) の0V時の立上り信号を取得し、位相が遅い方 (Timer1の値が大きい方) から位相が早い方 (Timer1の値が小さい方) を減算した値を位相差とした。

また、位相が電流と電圧どちらが進み・遅れの判断には以下の式を用いた。電流の立上り時間を $T_{rA}$ 、電圧の立上り時間を $T_{rV}$ 、位相差時間を $Q[s]$ とする。位相差時間 $Q$ が正の数の場合を位相差時間 $Q$ とした。

$$Q[s] = T_{rA} - T_{rV} \dots \textcircled{1}$$

$$Q[s] = T_{rV} - T_{rA} \dots \textcircled{2}$$

つまり、①は電流が進み、②は電圧が進んでいる。

位相差及び  $\cos \theta$  の求め方を以下に示す。周期は  $0.02[s]$  である。

位相差 =  $\cos(\text{rad})$

rad = 角度 $[\text{°}] / 360[\text{°}] \times 2\pi$

角度 $[\text{°}] = Q[s] / \text{周期}[s] \times 360[\text{°}]$



位相が早い波形：電圧波形 $V_r$  (2V/DIV, 5ms/DIV)

位相が遅い波形：電流波形 $I_r$  (2V/DIV, 5ms/DIV)

図 7 位相波形 (計測対象：卓上蛍光灯)

## 6 Visual C#®によるアプリケーションの製作

### 6.1 Windows®アプリケーション

Visual C#®でWindows®アプリケーションを製作し、PIC®で演算された消費電力データをシリアル通信により取込み、Excel®のセルデータとして消費電力を出力させた。Visual C#®とはマイクロソフト社が、NET®に対応したオブジェクト指向プログラミング言語であり、NET®はマイクロソフトが開発したアプリケーション開発/実行環境である。

### 6.2 仮想 COM ポートへのアクセス

本来のデータ通信規格がシリアル通信 (UART) であるため、PC上では仮想COMポートとして認識される。これにより Visual C#® のツールボックスにある SerialPort コントロールを使用した。シリアル通信をするにあたって以下の設定をする。

設定項目	設定値
通信ポート	PCの接続端子。(COM3)
通信速度	XBeeの通信速度と同じにする。(9600)
データビット	データ長。(8)
パリティチェック	(なし)
ストップビット	1
フロー制御	(なし)

図 8 シリアル通信設定値

以上の設定項目は SerialPort コントロールのプロパティで設定する。これはプログラミングで変更できる。次に実際に通信を行うために必要なポート接続、送信、受信メソッドを以下に示す。メソッドとは Visual C#® が持つ操作関数である。

メソッド名	動作
comSerialPort.open()	指定されたポートを開く。
comSerialPort.close()	指定されたポートを閉じる。
comSerialPort.WriteLine()	データを送信する。 例 comSerialPort.WriteLine("start#n#r");
comSerialPort.ReadLine()	データを受信する。

図 9 シリアル通信に必要なメソッド

以上のメソッドを使用して通信を行った。しかしデータ受信については DataReceive イベントを使う必要がある。イベントは、ボタンが押された、データが受信されたなどの動きがあると実施されるプログラム (イベントハンドラ) を持つ。

複数のスレッド (処理中プログラム) が、一つのプロパティにアクセスしない様にスレッドセーフな記述をする必要がある。しかし、フォームのスレッドとシリアル受信のスレッドは別のスレッドとして実行されるが、受信データをフォーム上のテキストボックスに記述したい。このためにデリゲートを行った。デリゲートとはメソッドを呼び出す側と呼び出される側とを仲介する機能である。

今回はフォームのスレッドが操作対象とする TextBox.text プロパティに対して、このプロパティを操作するメソッドを記述する。そして、Serialport クラスの DataReceived イベントハンドラ側でデリゲートにより上記のメソッドを呼び出した。

### 6.3 Excel®へのアクセス

作成した Windows® アプリケーションにおいて Excel®を起動するには、参照の追加より Microsoft® Excel® 12.0 Object Library・Microsoft Office 12.0 Object Libraryを参照設定に追加する。さらに名前空間に Microsoft.Office.Interop.Excel を登録する。以

上によりExcel®の起動・終了などが可能になった。Excel®に必要なメソッド、プロパティ設定を以下に示す。

メソッド名・プロパティ名	動作
Excel.Application.Visible=true	起動
Excel.Application.Quit()	終了
Excel.Worksheet.Cell[,]	行と列を指定しデータを表示
Excel.Workbook.open()	既存ファイルを開く
Excel.Workbook.Save()	上書き保存
Excel.Workbook.SaveAs()	名前を付けて保存

図 10 Excel®に必要なメソッドとプロパティ

これらの技術でデータの表示を行った。動作準備として既存ファイルを用意する。これにはデータがセルに入力されるとグラフ化される。操作においては最初に既存のファイルを開く。データを 30 回セルに入力し、自動でファイルを保存する。ファイル名は PC のシステムの現時刻を指定した。

現時刻の取得方法はツールボックスの Timer コントロールを使用し、Datetime.Now プロパティで時間を読み出した。また、Timer コントロールはインターバル時間を指定するとその時間が立つごとにイベントが発生する。この機能を利用し、コマンドを一定間隔で送信することが可能である。

また、図 11 に示すように Visual C#®のツールボックスよりコンボボックス、テキストボックス、ラベル、ボタンを使用し、ダイアログによる確認、情報通知により、実用性を追加した。

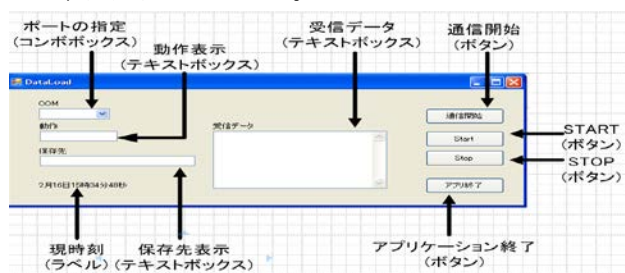


図 11 Visual C#®によるアプリケーション

## 7 製作回路例

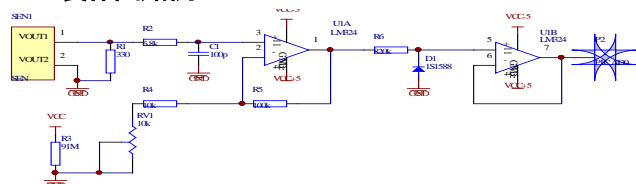


図 12 例 電流計測回路

電流センサが接続した抵抗により電圧出力となるので、電流/電圧で検出対象が異なる二つの回路動作は、ほぼ同じである。各データをローパス・フィルタで高周波数成分（約 200kHz）をカットする。オペアンプ（LM324）の非反転増幅回路で増幅し、ダイオードの半波整流で負電圧

側をカットする。各回路の出カインピーダンスが高いと本来の出力電圧が変化するので、ボルテージフォロワの追加で出カインピーダンスを低下させ、出力信号をPIC®のAD変換に接続した。

## 8 製作結果

### ① XBee®による通信

XBee®により無線化が成功し、コマンドの送受信が可能になった。またXBee®を 3 台使用し、1 対Nの場合でも同様の動作をした。しかし消費電力の取得に関してはハードウェアの製作が遅れているためにブレッドボード上での 1 対 1 の通信のみになった。

### ② 電力データの取得と位相差(ブレッドボード上)

・位相なし（計測対象:半田ごて）の場合  
電力データの取得に成功した。電力データ正確さは±5%であるが稀に±20%のほどの誤差が発生する。原因はブレッドボード上増幅回路の接触不良と考えられる。

・位相あり（計測対象:卓上蛍光灯）の場合  
位相差の取得には成功した。しかし電流、電圧を同時に取得するプログラムを現在製作中である。

### ③ Visual C#®によるアプリケーションの製作

Visual C#®によりPIC®で演算した電力データをExcel®に表示するアプリケーションが完成した。

### ④ ハードウェアの作製

感光基板により製作中。今後とも実施する。

## 9 まとめと謝辞

今回の卒業制作を通して、C 言語によるプログラミング、さらに電気回路、CAD によるハードウェアの製作など多くの知識、技能が身に着いたと実感し、ものづくりの楽しさをより深めた。そして時間内にもものを作ることの難しさ、スケジューリングの大切さを改めて実感した。また本製作にあたり、指導教諭の三浦先生から、丁寧かつ熱心なご指導を賜りました。ここに感謝の意を表します。

## 参考文献

### 1) 消費電力が計れる電力計

<http://www010.upp.so-net.ne.jp/pc-kosaku/index.htm>

### 2) XBee の使い方

<http://www.maroon.dti.ne.jp/koten-kairo/index.html>

### 3) C#による計測制御プログラミング 金藤仁 著

### 4) やさしいC# 高橋麻奈 著

# 課題実習「テーマ設定シート」

作成日：平成23年 8月 19日

科名：電子情報技術科

教科の科目	実習テーマ名
総合制作実習Ⅱ	無線消費電力計の製作
担当教員	担当学生
○三浦 雅嗣	
課題実習の技能・技術習得目標	
無線消費電力計の製作を通して、設計、製作及び組立・調整技術等の総合的な実践力を身に付けるとともに、無線消費電力計の設計を通して、実践的な電子回路設計技術、マイコンプログラミング技術、制御システム設計技術も身に付けます。	
実習テーマの設定背景・取組目標	
実習テーマの設定背景	
XBee®による無線通信を自ら学ぶことで、エンジニアに必要な「自ら学ぶ姿勢」を習得します。 これらを利用したシステムを構築することで、興味や面白さからの発展性を理解します。 製作物が綺麗に、確実に動作することにより、「モノづくり」の面白さを理解します。 期限までに成果物を完成させることでスケジュール管理の重要性を認識します。 複数で製作することで、協調性と責任感を育成します。	
実習テーマの特徴・概要	
XBee®を利用した無線消費電力計を製作します。 プログラム設計部とハード製作部を分担し、協力して製作します。 プログラム設計部はXBee®通信モジュールを利用して通信制御を行い、消費電力計のプログラミングを行います。 ハード製作部は技能検定「電子機器組み立て2級」レベルの課題を行い、電子機器を綺麗に製作可能にします。CADにおいてはライブラリ作成が自由に可能になり、プリント基板の製作工程をマスターします。 最終的に協力して組立・調整・動作試験を行い、報告書を作成します。	
No	取組目標
①	XBee®を利用した無線通信を習得します。
②	消費電力計ハード部の設計・回路特性を行います。
③	消費電力計のプログラミングを行います。
④	消費電力計のシステムプログラミングを行います。
⑤	電子回路製作に必要な技術（CAD・プリント基板製作・束線等）を習得します。※各種回路の特性計測
⑥	作業手順や作業ポイントについては、マニュアルを作成してまとめます。
⑦	想定した動作が行われなかった場合には、問題を分析し、その問題の解決に取り組みます。
⑧	5S（整理、整頓、清掃、清潔、躰）の実現に努め、安全衛生活動を行います。
⑨	発表レジュメ、プレゼンテーション資料の作成、卒論の作成、製作品の展示及び発表会を行います。
⑩	実習の進捗状況や、発生した問題等については、単独、グループの場合にかかわらず、担当教員へ報告します。