

情報工学実験2

携帯電話による施錠制御

担当教員：宮里 智樹

実験日 2008/01/26

2008/02/02

提出日 2008/02/09

075730G：澤岷千明

課題 1 : USB について簡潔にまとめよ。

USB とは、Universal Serial Bus の略称で、コンピュータに周辺機器を接続するためのシリアルバス規格の一つである。最初の規格である、USB 1.0 は、本体との間であまり大量のデータをやりとりしない機器を接続するための規格として出た。規格にバスパワー、すなわちホストアダプタからの電源供給を規定している。電源機器とデータのやり取りがケーブル一本で済ませられることから、現在のパーソナルコンピュータ及びその周辺において、最も普及した汎用インターフェイス規格の一つ。

課題 2 : USB-IO について調べよ。

電源不要の USB インターフェイスを使った入出力基盤で、12 点のデジタル入出力ピンがある。Pin0 ~ Pin7 は Port0 と呼ばれ、Pin8 ~ Pin11 は Port1 と呼ばれる。

電気の入出力を制御するものなので、これ単体では使えない。コンピュータやスイッチ、センサー等と接続する必要がある。

課題 3 : ドアの施錠を以下に示す様に制御するプログラムを作成せよ。

「解錠する。その後ドアの開閉を監視し、ドアを閉じられると施錠する。」

初期化の記述は jikken2_sample.hsp をそのまま使用しているので、省略した。

省略

```
; ***** key open *****鍵を開けるだけ
log_str="**開錠します\n":gosub *show_log; ログ表示
uio_out 0, _PIN6_KEYOPEN_, 0 ; 開錠
while; 無限ループ
    uio_inp inp, 1, 0 ; 入力を取る
    inp=inp & _PIN9_KEYOPENED_; 開錠入力を抽出
    if inp!0 : _break ; 開錠完了したらループを抜ける
    wait 10 ; しばらく待つ
wend
uio_out 0, 0, 0; 開錠おわり アクチュエータ停止
log_str="**鍵が開きました。 \n":gosub *show_log; ログ表示

; *****ここから
log_str="**ドアが開けられる状態です。 \n":gosub *show_log; ログ表示
;
uio_out 0, _PIN8_DOOROPEN_, 0;
while; ドアが開いたかどうかの判定
    uio_inp inp, 1, 0 ; 入力を取る
    inp=inp & _PIN8_DOOROPEN_; ドアが開いた入力を抽出
    if inp=1 : _break ; ドアが開いたらループを抜ける
    wait 10;
wend
log_str="**ドアが開きました。 \n":gosub *show_log; ログ表示
uio_out 0, 0, 0; 開錠おわり アクチュエータ停止
; *****ブザーを鳴らす
; *****ドアの監視
log_str="**ドアを閉めてください。 \n":gosub *show_log; ログ表示
while; 無限ループ
```

```

        uio_inp inp,1,0          ; 入力を取る
inp=inp & _PIN8_DOOROPEN_; 開錠入力を抽出
        if inp=0 : _break      ; 開錠完了したらループを抜ける
        wait 10 ; しばらく待つ
wend
log_str="**ドアが閉まりました。 \n":gosub *show_log; ログ表示

;*****鍵を閉める
log_str="**施錠します\n":gosub *show_log; ログ表示
uio_out 0, _PIN7_KEYCLOSE_,0      ; 開錠
while; 無限ループ
        uio_inp inp,1,0          ; 入力を取る
inp=inp & _PIN10_KEYCLOSED_; 開錠入力を抽出
        if inp!0 : _break      ; 開錠完了したらループを抜ける
        wait 10;
wend
;*****ドアの監視
log_str="**施錠しました\n":gosub *show_log;
uio_out 0,0,0; 施錠おわり アクチュエータ停止

;
while;
;
        uio_inp inp,1,0
;
        inp=inp & _PIN9_KEYOPENED_; あいたことを確認
;
        ;if inp!0 ;: _break
;
        wend
;/
log_str="**閉錠する":gosub *show_log
;
while
;
        uio_inp inp,1,0
;
        inp=inp & _PIN8_DOOROPEN_          ; ドア閉
;
        inp=inp & _PIN10_KEYCLOSED_      ; 鍵閉位置
//
        uio_inp 0, _PIN8_DOOROPEN_,0      ; ドア閉
//
        uio_inp 1, _PIN10_KEYCLOSED_,1    ; 鍵閉位置
//
        switch
//
        case 0, _PIN8_DOOROPEN_,0          ; ドア閉
//
        case 1, _PIN10_KEYCLOSED_,1      ; 鍵閉位置
//
        swbreak
;
wend
;
        uio_out 1, _PIN7_KEYCLOSE_,1      ; 施錠
;
        uio_out 0,0,0;
;
        log_str="閉錠しました\n":gosub *show_log;
;*****

stop; 止まります

```

省略

始めの key open は sample プログラムと同様である。ドアが開いているかどうかの判断、閉じられた時の施錠の処理が書き加えられている。

課題4：ドアの施錠を以下の示す様に制御するプログラムを作成せよ。

「解錠する。その後ドアの開閉を監視し、5秒以上ドアが開けられるとブザーを鳴らす。ドアが閉じられると施錠しブザーが鳴っていればそれを止める。」

こちらも課題3と同様に、初期化部分は省略している。

省略

```
; ***** key open *****鍵を開けるだけ
log_str="**開錠します\n":gosub *show_log; ログ表示
uio_out 0, _PIN6_KEYOPEN_, 0 ; 開錠
while; 無限ループ
    uio_inp inp, 1, 0 ; 入力を取る
    inp=inp & _PIN9_KEYOPENED_; 開錠入力を抽出
    if inp!0 : _break ; 開錠完了したらループを抜ける
    wait 10 ; しばらく待つ
wend
uio_out 0, 0, 0; 開錠おわり アクチュエータ停止
log_str="**鍵が開きました。 \n":gosub *show_log; ログ表示

;*****ここから
i=0
log_str="**ドアが開けられる状態です。 \n":gosub *show_log; ログ表示
;
uio_out 0, _PIN8_DOOROPEN_, 0;
while; ドアが開いたかどうかの判定
    uio_inp inp, 1, 0 ; 入力を取る
    inp=inp & _PIN8_DOOROPEN_; ドアが開いた入力を抽出
    if inp=1 : _break ; ドアが開いたらループを抜ける
    wait 10;
wend
log_str="**ドアが開きました。 \n":gosub *show_log; ログ表示
uio_out 0, 0, 0; 開錠おわり アクチュエータ停止
;*****5sec count...

;
while;
;
    i =i+1
;
    if i>50 : _break
;
    wait 10;
;
wend
;
log_str="**5秒たったのでブザーを鳴らします。 \n":gosub *show_log; ログ表示

;*****ブザーを鳴らす
;
log_str="**peeeeeeeeeeee。 \n":gosub *show_log; ログ表示
;
while;
;
    uio_inp inp, 1, 0 ; 入力を取る
;
    inp=inp & _PIN5_BEEP_ ; ビープ音の入力を抽出
;
    if inp=1 : _break ; 音が鳴ったらループを抜ける
;
    wait 10;
;
wend
;
;*****ドアの監視
log_str="**ドアを閉めてください。 \n":gosub *show_log; ログ表示
while; 無限ループ
    i = i+1
```

```

        if i=50 : uio_out 0,_PIN5_BEEP_,0
        log_str="**peeeeeeeeeeeee. \n":gosub *show_log; ログ表示
        uio_inp inp,1,0          ; 入力を取る
        inp=inp & _PIN8_DOOROPEN_; 開錠入力を抽出
        if inp=0 : _break        ; 開錠完了したらループを抜ける
        wait 10 ; しばらく待つ
    wend
;*****ブザーを消す
    uio_out 0,0,0
    log_str="**ドアが閉まりました。 \n":gosub *show_log; ログ表示
;*****鍵を閉める
    log_str="**施錠します\n":gosub *show_log; ログ表示
    uio_out 0,_PIN7_KEYCLOSE_,0      ; 開錠
    while; 無限ループ
        uio_inp inp,1,0          ; 入力を取る
        inp=inp & _PIN10_KEYCLOSED_; 開錠入力を抽出
        if inp!0 : _break        ; 開錠完了したらループを抜ける
        wait 10;
    wend
;*****ドアの監視
    log_str="**施錠しました\n":gosub *show_log;
    uio_out 0,0,0; 施錠おわり アクチュエータ停止

;
    while;
;
        uio_inp inp,1,0
;
        inp=inp & _PIN9_KEYOPENED_; あいたことを確認
;
        ;if inp!0 ;: _break
;
        wend
;/
    log_str="**閉錠する":gosub *show_log
;
    while
;
        uio_inp inp,1,0
;
        inp=inp & _PIN8_DOOROPEN_          ; ドア閉
;
        inp=inp & _PIN10_KEYCLOSED_      ; 鍵閉位置
//
        uio_inp 0,_PIN8_DOOROPEN_,0      ; ドア閉
//
        uio_inp 1,_PIN10_KEYCLOSED_,1    ; 鍵閉位置
//
        switch
//
            case 0,_PIN8_DOOROPEN_,0      ; ドア閉
//
            case 1,_PIN10_KEYCLOSED_,1    ; 鍵閉位置
//
        swbreak
;
    wend
;
        uio_out 1,_PIN7_KEYCLOSE_,1      ; 施錠
;
        uio_out 0,0,0;
;
        log_str="閉錠しました\n":gosub *show_log;
;*****

    stop; 止まります

; ***** ログ表示 *****
;
; 使い方
; log_str に好きな文字列を入れて、*show_log を gosub で呼び出す
;
*show_log
    LOG_BUFF=log_str+LOG_BUFF;

```

```

objprm LOG_OBJ,LOG_BUFF;
return;

*beep
    log_str="**peeeeeeeee\n":gosub *show_log; ログ表示
uio_out 0,_PIN5_BEEP_,0      ; 開錠
while; 無限ループ
    uio_inp inp,1,0          ; 入力を取る
    inp=inp & _PIN8_DOOROPEN_; 開錠入力を抽出
    if inp=0 : _break        ; 開錠完了したらループを抜ける
    wait 10 ; しばらく待つ

wend
uio_out 0,0,0; 開錠おわり アクチュエータ停止
log_str="**peeee----stop. \n":gosub *show_log; ログ表示
return

*owari
uio_out 0,0,0;
uio_out 1,0,0;
uio_free;
end

```

課題3のソースに、新たに書き加えたもの。「ドアの監視」に5秒開けるとブザーがなる処理を書き加え、ドアが閉まるとブザーが消える処理も加えた。最後のログ表示にも、ブザー音がなっている時にも、なっていることを示す処理を書き加えた。

課題5：VAIOにて、ウェブサーバを起動し、表示可能なページを作成せよ。

プログラムの中から、[Apache HTTP Server] [Control Apache Serever] [Monitor Apache Servers] と辿り、Apache Service Monitor を起動。その後、タスクトレイ (画面の左下) に表示される apache service monitor のアイコンをクリック。start で apache を起動することができる。この時、C:\Program Files\Apache Group\Apache\htdocs 以下に html ファイルを置くこと。html を編集などした時は、restart でその更新を反映させる必要がある。

以下に、その html ファイルのソースを示す。

door.html のソース

```

<html>
<body>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<br>
<center>情報工学実験 2 </center>
<br>
<a href="oc.php">開け閉め</a>
</body>
</html>

```

課題6：携帯電話から、1週間目の課題4で作成したプログラムを実行可能にせよ。

HSP スクリプトエディタで、課題4で作成したソースを exe ファイルに変換する。それで生成された hsptmp.exe というファイルを C:\Program Files\Apache Group\Apache\htdocs 以下に置き、apache を restart させればよい。なお、ここでは全てのファイルは同階層に置かれている。

以下に、oc.php のソースを示す。

oc.php のソース

```
<!DOCTYPE html PUBLIC "-//OPENWAVE//DTD XHTML 1.0//EN"
"http://www.openwave.com/DTD/xhtml-basic.dtd">
<html>
<body>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<h1>実行中</h1>
<?php
exec("hsptmp");
?>
</body>
</html>
```

課題7：遠隔制御を実現する上で、他に追加すべき機能は何か、考察せよ。

現状だと、誰でも簡単に操作できるので、特定の人しか制御できないようにパスワード等を設定が必要だと考える。

使用するのが一人ならまだいいが、複数で使うのなら何時使われたかといった機能も必要だと思われる。

参考

LaTeX コマンドシート一覧

<http://www002.upp.so-net.ne.jp/latex/index.html>

Wikipedia

<http://ja.wikipedia.org/wiki/メインページ>

IT用語辞典 e-Words

<http://e-words.jp/>

Km2Net

<http://km2net.com/usb-io/index.shtml>