ภาคผนวก ฉ

โครงสร้างคำสั่งการติดต่อกับคอมพิวเตอร์ของคอนโทรลยูนิต

การติดต่อกับคอนโทรลยูนิตของคอมพิวเตอร์กระทำผ่านพอร์ตเครื่องพิมพ์ ซึ่งพอร์ตนี้จะมีอยู่ทั้งหมด 3 พอร์ตย่อย คือ 1.Command 2.Data และ 3.Status (ดูรายละเอียดเพิ่มเติมเกี่ยวกับพอร์ตที่ภาคผนวก ค และเกี่ยวกับหน่วยความจำที่ภาคผนวก ข) โดยมีการกำหนดหน้าที่การทำงานดังนี้

1. Command คือพอร์ตที่ใช้สำหรับส่งคำสั่งไปให้คอนโทรลยูนิต มีทั้งหมด 4 บิต โดยใช้บิตที่ 0 ถึง 2 เป็นคำสั่งและบิตที่ 3 เป็นสัญญาณกระตุ้น (Strobe) โดยคอนโทรลยูนิตจะรับคำสั่งเมื่อบิตนี้เปลี่ยนสถานะจากบวกไปลบ ดังนั้นเมื่อต้องการให้คอนโทรลยูนิตรับคำสั่งจะต้องส่งค่ารหัสคำสั่งไปที่บิตที่ 0 ถึง 2 ก่อนแล้วจึงจะทำการส่งสัญญาณกระตุ้น

2. Data คือพอร์ตที่ใช้สำหรับส่งข้อมูลให้กับคอนโทรลยูนิต เช่น ตำแหน่งของหน่วยความจำที่ต้องการอ่าน/เขียน หรือตำแหน่งไคลเอนท์ เป็นต้น พอร์ตนี้มีทั้งหมด 8 บิต

3. Status คือพอร์ตที่ใช้สำหรับรับข้อมูลมาจากคอนโทรลยูนิต ซึ่งพอร์ตนี้จะมีอยู่ 5 บิต คือ บิตที่ 3 ถึง 7 โดยบิตที่ 3 ถึง 6 จะเป็นข้อมูลส่วนบิตที่ 7 จะเป็นบิตแสดงสถานะ คือ ถ้าคอนโทรลยูนิตพร้อมสำหรับการรับคำสั่ง หรือส่งข้อมูลบิตนี้จะมีสถานะเป็น Logic 1

โครงสร้างคำสั่งมีรายละเอียดดังนี้

- [Command = 0] Connection status
  - [Status < 128] Busy
  - [Status > 128] No Busy
- [Command = 1] The pointer of type of memory
  - [Data = 0] External EEPROM
  - [Data = 1] External RAM
- [Command = 2] Client power supply
  - [Data = 1] Power On
  - [Data = 0] Power Off

- [Command = 3] The code of client
  - [Data = 0] Deactivate all signal line
  - [Data = 1] Signal line 1 is activate
  - [Data = 2] Signal line 2 is activate
  - [Data = 3] Signal line 1 and 2 are activate
  - [Data = 4] Signal line 3 is activate
  - [Data = 5] Signal line 1 and 3 are activate
  - [Data = 6] Signal line 3 and 2 are activate
  - [Data = 7] Signal line 1, 2 and 3 are activate
  - [Data = 8] Signal line 4 is activate
  - [Data = 9] Signal line 1 and 4 are activate
  - [Data = 10] Signal line 2 and 4 are activate
  - [Data = 11] Signal line 1, 2 and 4 are activate
  - [Data = 12] Signal line 3 and 4 are activate
  - [Data = 13] Signal line 1, 3 and 4 are activate
  - [Data = 14] Signal line 2, 3 and 4 are activate
  - [Data = 15] Signal line 1, 2, 3 and 4 are activate
  - [Data = 16 to Data = 31] Reserved

- [Command = 4] The Address of client
  - [Data = 1 to 255] Active client
  - [Data = 0] All client are active

- [Command = 5] Measuring
  - [Data = 0] Transfering new data
  - [Data = 1] Sending code and address to client
  - [Data = 2] Trig the terrameter to measuring, recieving the new data

- [Command = 6] Memory Access { * Must be go out by Command = 0 when working complete}

- [Command = 0] Go out of memory access function
- [Command = 1, Data = XXH] Set the pointer of high byte memory address to XXH
- [Command = 2, Data = XXH] Set the pointer of low byte memory address to XXH
- [Command = 3] Read the 4 High bits of data from memory where following by the pointer of type of memory (see Command = 1) and at the dddress where following by address pointer
  - External RAM
  - External EEPROM
- [Command = 4] Read the 4 low bits of data of above command and increase the memory address pointer
- [Command = 5] Read the 4 high bits of data from the microcontroller internal memory at the pointer address
- [Command = 6] Reserved

➢ [Command = 7] Working completed

ตัวอย่างโปรแกรมการอ่านข้อมูลจากแฟ้มในหน่วยความจำของคอนโทรลยูนิต ที่เขียนด้วยภาษาปาสคาล โดยเครื่องมือ Borland Pascal V7.0 มีดังนี้

```
Program Test;
Uses Crt,Dos;
Const
      MaxChr     = 10;
      MaxArray   = 200;
Type
      StrType      = String[MaxChr+2];
      DataAType      = Array[1..MaxArray] Of StrType;
Var
      Data             : DataAType;
```

```
    i,j,k,Electrode,FNum,I    : Integer;
    ABegin,AEnd,Num           : Word;
    Einterval     : Real;
    DataType,N,Remain         : Byte;
    StrWay        : StrType;
    Ch            : Char;

Procedure Command(P888:Byte;P890:Byte); {ส่งคำสั่งให้คอนโทรลยูนิต}
Begin
    Port[888] := P888;
    Repeat
        Port[890] := P890; {ส่งคำสั่งและสัญญาณกระตุ้นไปที่ Command พอร์ต}
      Until Port[889] < 128; {รอจนกว่าคอนโทรลยูนิตมีสถานะไม่พร้อม (ได้รับคำสั่งแล้ว)}
    Repeat
        Port[890] := P890 + 8; {ปรับสัญญาณกระตุ้นให้อยู่ในสถานะปกติ}
      Until Port[889] > 127; รอจนกว่าคอนโทรลยูนิตมีสถานะพร้อม (ทำงานเสร็จแล้ว)}
End;

Function GetData(P890:Byte):Byte;
Var
    Way : Byte;
Begin
    Repeat
        Port[890] := P890;
      Until Port[889] < 128;
      Port[890] := P890 + 8;
    Repeat
        Way := Port[889]
      Until Way > 127;
    GetData := Way;
```

```
End;

Function GetNext : Byte;
Var
        Way : Byte;
Begin
        Way := (GetData(3) Shl 1);
        Way := Way And 240;
        Way := Way Or ((GetData(4) Shr 3) And 15);
        GetNext := Way;
End;

Function GetByte(Address : Word) : Byte;
Begin
        Command((Address Shr 8),1);
        Command(Address,2);
        GetByte := GetNext;
End;

Function GetWord(Address : Word) : Word;
Var
        Way : Word;
Begin
        Way := (GetByte(Address) Shl 8);
        Way := Way Or GetNext;
        GetWord := Way;
End;

Begin
        ClrScr;
        Command(0,6);
```

```
FNum := GetByte(1);
WriteLn('Total ',FNum,' Files');
Command(0,0);
Command(0,7);
Write('Select file ? ');ReadLn(i);
While (i > 0) And (i <= FNum) Do
Begin
        Command(0,6);
        ABegin   := GetWord(16*i);
        Num      := GetNext;
        AEnd     := GetWord(16*i+3);
        EInterval := GetNext / 10;
        DataType  := GetNext;
        N        := GetNext;
        Remain   := GetNext;
        Electrode := GetNext;
        k := 0;
        l := 0;
        StrWay := '';
        j := GetByte(ABegin-1);
        For j := 1 To (AEnd - ABegin) Do
        Begin
        k := k + 1;
        Ch := Chr(GetNext);
        If (Ch='.') And ((k=1) Or (StrWay[1]='.')) Then
        Begin
              StrWay := StrWay + '0';
           End;
           StrWay := StrWay + Ch;
```

```
            If k = MaxChr Then
            Begin
                  k := 0;
                  I := I + 1;
                  Data[I] := StrWay;
                  StrWay := '';
            End;
      End;
      WriteLn('Begin    ',ABegin:6);
      WriteLn('End      ',AEnd:6);
      WriteLn('Num      ',Num:6);
      WriteLn('Interval ',EInterval:6:2);
      WriteLn('Data type ',DataType:6);
      WriteLn('N        ',N:6);
      WriteLn('Remain   ',Remain:6);
      WriteLn('Electrode ',Electrode:6);
      ReadLn;
      i := 0; I := 0;
      Repeat
         i := i + 1;
         Write(i:3,') ');Write(Data[i]:11,'  ');
         If (i Mod 4)= 0 Then WriteLn;
      Until i = Num;
      Command(0,0);
      Command(0,7);
      WriteLn;
      Write('Select file ? ');ReadLn(i);
   End;
End.
```