

การเชื่อมโยงคอมพิวเตอร์กับเครื่องมือ

อ.มนู เฟื่องฟุ้ง และ อ.ดร.ทวี จินอ้อ¹

ถ้าจะถามว่า"เราจะได้ประโยชน์อะไรจากการเชื่อมโยงคอมพิวเตอร์กับเครื่องมือ?" ก่อนตอบคำถามนี้ คงจะต้องทราบก่อนว่าเครื่องมือในที่นี้เราแบ่งเป็นสองประเภทคือ

ก) เครื่องมือตรวจวัด (Measuring equipment) เช่น ตรวจวัดอุณหภูมิ ตรวจวัดความดัน ตรวจวัดเวลา และตรวจวัดอัตราการไหล เป็นต้น

ข) เครื่องมือควบคุมกระบวนการ (Process-control equipment) เช่น ควบคุมอุณหภูมิ ควบคุมความดัน ควบคุมเวลา และควบคุมอัตราการไหล เป็นต้น

เครื่องมือทั้งสองประเภทดังกล่าวข้างต้นมีประโยชน์มากมาย เช่น แทนที่นักฟิสิกส์จะต้องมานั่งบันทึกค่าที่อ่านได้จาก เครื่องมือวัดอุณหภูมิ เครื่องวัดความต่างศักย์ เครื่องวัดกระแส แล้วป้อนข้อมูลเข้าเครื่องคอมพิวเตอร์ทางคีย์บอร์ดเพื่อคำนวณในสิ่งที่ต้องการ ก็อาจเชื่อมโยงคอมพิวเตอร์กับเครื่องวัดโดยตรงแล้วปล่อยให้คอมพิวเตอร์ทำหน้าที่บันทึกค่าที่ต้องการลงในหน่วยความจำตามโปรแกรมที่เรากำหนด ก็จะเป็นประโยชน์ทั้งด้านประหยัดเวลาและความแม่นยำที่สูง

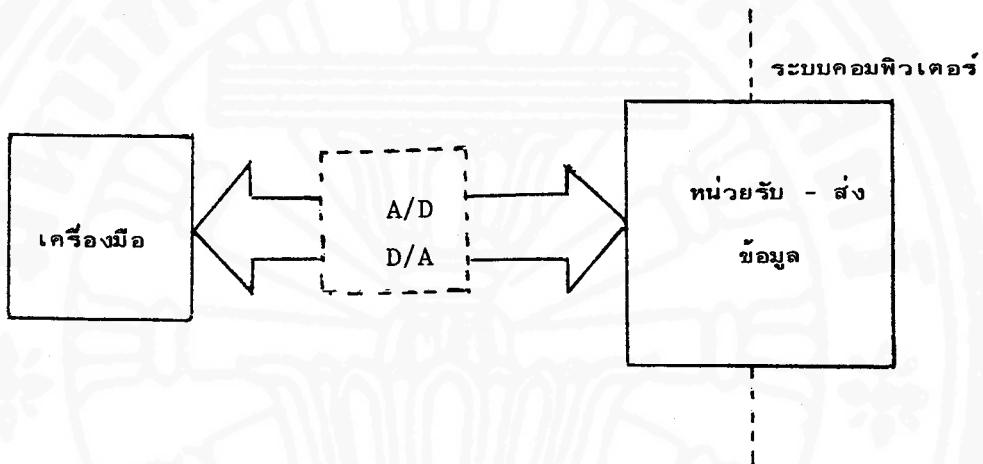
ก่อนที่จะกล่าวถึงวิธีการออกแบบการเชื่อมต่อระหว่างคอมพิวเตอร์กับเครื่องมือ เราต้องมาทำความเข้าใจเกี่ยวกับธรรมชาติของคอมพิวเตอร์และเครื่องมือก่อน

เนื่องจากคอมพิวเตอร์ทำงานในลักษณะประมวลผลข้อมูลแบบเลขฐานสอง ซึ่งเลขฐานสองหนึ่งหลักมีหน่วยที่เรียกว่า บิต (bit) และถ้ามองเลขฐานสองในลักษณะสัญญาณไฟฟ้าแล้วจะเป็นสัญญาณไฟฟ้าสองระดับ คือ 0 และ 1

สำหรับเครื่องมือ เรามักจะหมายถึงเครื่องมือที่ทำงานด้วยกลไกของสัญญาณไฟฟ้าได้หรือเปลี่ยนสัญญาณที่มีอยู่ในรูปของพลังงานอื่นเป็นสัญญาณไฟฟ้าได้ เช่น เปลี่ยนสัญญาณในรูปของปริมาณความร้อน(อุณหภูมิ)เป็นสัญญาณไฟฟ้า หรือ เปลี่ยนสัญญาณในรูปของความดันเป็นสัญญาณไฟฟ้าได้ เป็นต้น ตัวที่ทำหน้าที่เปลี่ยนปริมาณจากรูปอื่นเป็นปริมาณไฟฟ้าได้ เราเรียกว่า "ตัวแปลงปริมาณ" (transducer หรือ sensor) ตัวอย่างเช่น เซลล์แสงอาทิตย์ทำหน้าที่แปลงปริมาณแสงสว่างเป็นปริมาณไฟฟ้า เป็นต้น ดังนั้นจะเห็นได้ว่าการเชื่อมโยงระหว่างคอมพิวเตอร์กับเครื่องมือเป็นการถ่ายเทข้อมูลในลักษณะของเลขฐานสองซึ่งกันและกันนั่นเอง

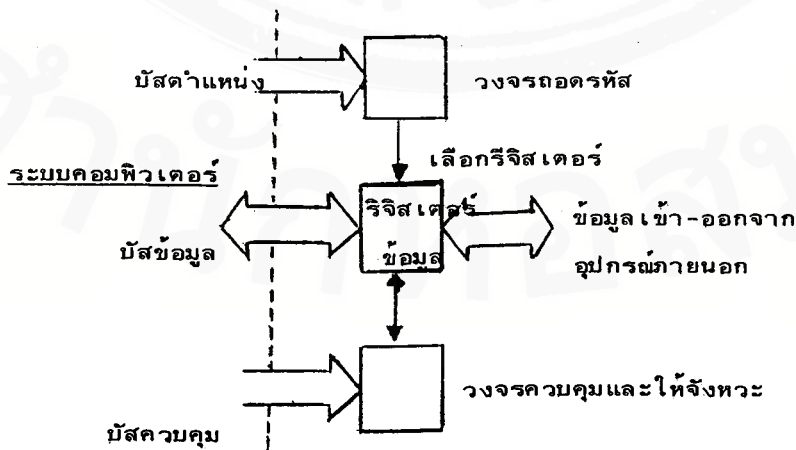
¹ สาขาวิชาฟิสิกส์ คณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยธรรมศาสตร์

สำหรับการเชื่อมโยงระหว่างคอมพิวเตอร์กับโลกภายนอก(เครื่องมือ)นั้น หน่วยรับส่งข้อมูลเข้าออก(input/output unit)จะทำหน้าที่เป็นตัวกลาง(ทำหน้าที่อินเตอร์เฟส) โดยตัวมันจะจัดข้อมูลที่ได้รับจากภายนอกให้อยู่ในรูปที่เหมาะสมแล้วจึงส่งเข้าสู่คอมพิวเตอร์ ในทางกลับกันก็สามารถรับข้อมูลจากคอมพิวเตอร์แล้วจัดให้อยู่ในรูปที่เหมาะสมเพื่อส่งให้กับเครื่องมือ การที่ต้องจัดการกับข้อมูลก่อนที่จะส่งเข้าคอมพิวเตอร์หรือส่งออกก็เพราะว่าข้อมูลที่ใช้ในระบบคอมพิวเตอร์กับเครื่องมือโดยปกติแล้วจะอยู่ในรูปแบบที่ต่างกัน



รูปที่ 1 แสดงการอินเตอร์เฟสระหว่างคอมพิวเตอร์กับเครื่องมือ

ในตัวอย่างของการออกแบบการอินเตอร์เฟสของบทความนี้จะใช้วิธีจัดวงจรเพิ่มเติมเพื่อแบ่งตำแหน่งของหน่วยรับ-ส่งข้อมูล โดยมีลักษณะพื้นฐานดังแสดงตามรูปที่ 2



รูปที่ 2 แสดงการต่อหน่วยรับ-ส่งข้อมูล

จากรูปที่ 2 เราก็มารู้ถึงการออกแบบการ์ด ซึ่งมีขั้นตอนที่พอสรุปได้ดังนี้

ขั้นตอนที่ 1 ก่อนอื่นต้องกำหนดคุณสมบัติของคร่าว ๆ ของการ์ดที่เราต้องการเสียก่อนว่า เป็นอย่างไร ตัวอย่างเช่น ต้องการให้มัลติพอร์ท 2 พอร์ต เอาท์พอร์ท 1 พอร์ต จำนวนช่อง รับสัญญาณ 8 ช่องและสามารถขยายเพิ่มเติมได้ เป็นต้น

ขั้นตอนที่ 2 กำหนดตำแหน่ง (address) ของการ์ดว่าจะให้อยู่ในช่วงใด ซึ่งกระทำได้ โดยพิจารณาจากแผนผังการจัดตำแหน่งหมายเลขพอร์ทของระบบไมโครคอมพิวเตอร์ดังแสดงตามรูปที่ 3 จะเห็นว่าในช่วงที่ยังไม่ถูกใช้งานอยู่หลายช่วงรวมทั้งส่วนที่สงวนไว้ (reserve) สมมติว่าเราเลือกให้ การ์ดของเราอยู่ที่ตำแหน่ง 2F0-2F7 ซึ่งเป็นช่วงที่ยังว่างอยู่

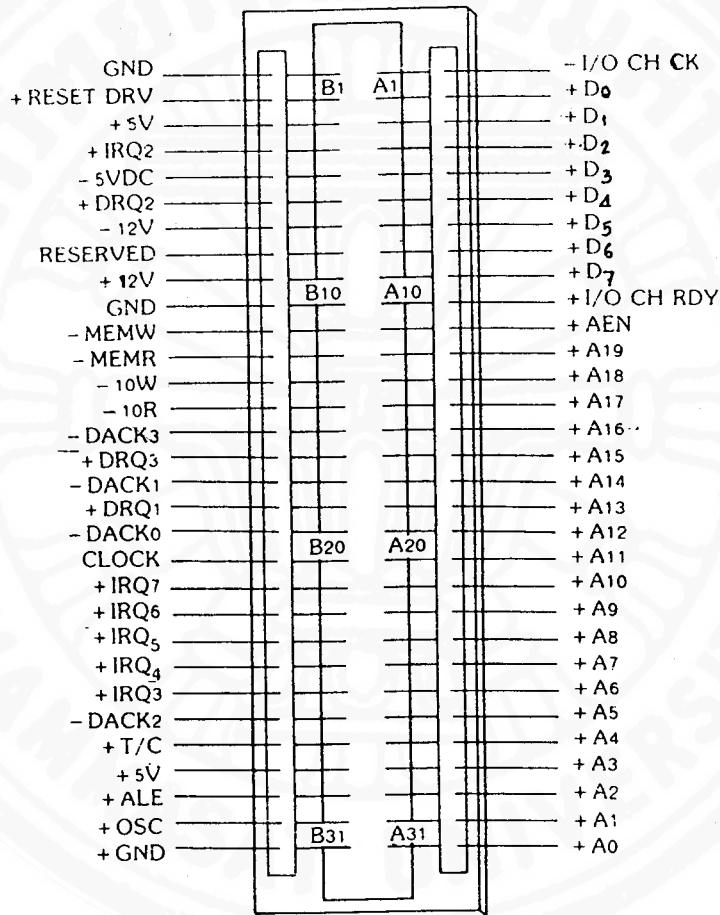
Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Registers
0AX*	NMI Mask Register
OCX	Reserved
OEX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Parallel Printer
380-38F	SDLC Communications
3A0-3AF	Reserved
3B0-3BF	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communications (Primary)

*At power-on time, the Non Mask Interrupt (NMI) into the 8088 is masked off
This mask bit can be set and reset via system software as follows
Set mask: write hex 80 to I/O Address hex A0 (enable NMI)
Clear mask: write hex 00 to I/O Address hex A0 (disable NMI)

รูปที่ 3 memory map

ขั้นตอนที่ 3 ตอนนี้เป็นการเลือกอุปกรณ์ที่จะนำมาออกแบบให้การ์ดของเรา มีคุณสมบัติตามที่ได้กำหนดไว้ โดยในที่นี้เลือกใช้ชิพไอซีเบอร์ ADC0808 เป็นตัวเปลี่ยนสัญญาณอนาล็อกเป็นดิจิทัล ขนาด 8 บิต 8 ช่องรับสัญญาณ⁽¹⁾ สำหรับพอร์ทได้เลือกใช้ไอซี เบอร์ 8255 PPI⁽²⁾ ซึ่งใช้ตำแหน่งในการกำหนดรีจิสเตอร์ภายใน 4 ตำแหน่ง เพราะฉะนั้นการ์ดของเราจะมีตำแหน่งอยู่ที่ 2F0, 2F1, 2F2 และ 2F3

ขั้นตอนที่ 4 ตอนนี้จะต้องไปดูลักษณะสัญญาณบนสล็อตของเครื่องก่อนว่ามีการจัดขาอย่างไร เพื่อจะได้เลือกขาที่เกี่ยวข้องมาใช้งานได้ถูก ซึ่งขาที่ใช้งานได้แก่ขา address A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 ขา RESET ขา IO/R ขา IO/W ขา V_{CC} ขา GND ขา AEN และขาบั๊ตข้อมูล



รูปที่ 4 แสดงลักษณะสัญญาณบนสล็อตของไมโครคอมพิวเตอร์ ระบบ XT

ขั้นตอนที่ 5 คราวนี้เป็นขั้นที่สำคัญที่เดิวนั้นคือต้องออกแบบวงจรถอดรหัสให้ได้ตรงตามตำแหน่งที่กำหนด เพราะไม่เช่นนั้นแล้วการติดต่อระหว่างคอมพิวเตอร์กับภายนอกก็ไม่สามารถสื่อสารกันได้ เหมือนกับส่งจดหมายแล้วไม่ได้บอกที่อยู่ของผู้รับ-ส่ง

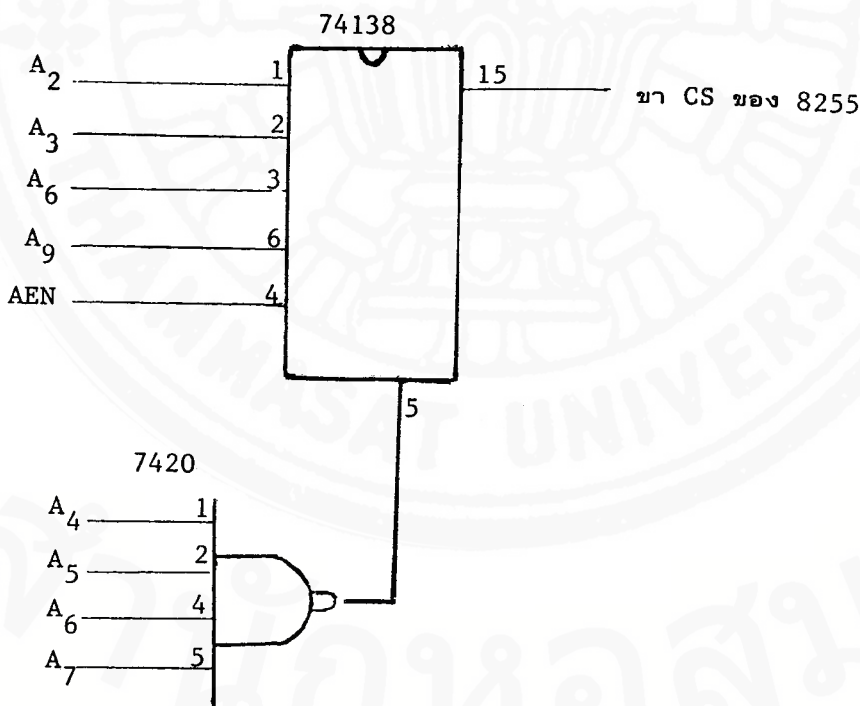
การออกแบบวงจรลอจิกที่ได้โดยนำตำแหน่งของการวัดที่เรากำหนด (คือ 2F0) มาเขียนในรูปของเลขฐานสอง ดังนี้

$$\begin{array}{rcccccccccccc} \text{บิตตำแหน่ง} & A_{10} & A_9 & A_8 & A_7 & A_6 & A_5 & A_4 & A_3 & A_2 & A_1 & A_0 \\ 2F0 & = & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \end{array}$$

เลือก A_7, A_6, A_5, A_4 มากระทำทางขบวนการลอจิกเพื่อให้ได้ผลลัพธ์ออกมาเป็นลอจิก "0" (จริง ๆ แล้วจะเลือกบิตตำแหน่งตัวไหนจากข้างบน มากระทำทางลอจิกแบบใดก็ได้ขอให้ได้ผลลัพธ์ออกมาแล้วอยู่ในช่วงของตำแหน่งที่เรากำหนดก็แล้วกัน)

$$A_7 \cdot A_6 \cdot A_5 \cdot A_4 = 0 \quad (1)$$

จากความสัมพันธ์ตามสมการข้างบน เราเลือกอุปกรณ์เกทพื้นฐานมาทำเป็นวงจรอินเวิร์ตและลอจิกได้ดังรูปที่ 5 ต่อไปนี้



รูปที่ 5 วงจรลอจิกที่ได้จากสมการ(1) ส่วนไอซีเบอร์ 74138 ใช้สำหรับในกรณีที่

เราต้องการขยายความสามารถของการวัดออกไป



การที่เราออกแบบมาเสร็จแล้วนั้นจะไม่มีประโยชน์อะไรเลยถ้าปราศจากโปรแกรมคอมพิวเตอร์
การทำงาน ซึ่งในหัวข้อนี้จะเป็นเรื่องตัวอย่างโปรแกรมคอมพิวเตอร์การอ่านค่าโวลเตจจากความต้านทาน
ปรับค่า (ซึ่งสมมติว่าเป็นสัญญาณจากเครื่องมือ)

โปรแกรมทดสอบเขียนด้วยภาษาเบสิก

```

10 cls
12 out 755,146 ***กำหนดให้พอร์ท A,Bเป็นอินพุตพอร์ท พอร์ท C เป็นเอาต์พุต
15 print"                                ANALOG TO DIGITAL TESTER"
16 print"                                ====="
20 print" "
30 out 754,0 *** ส่งสัญญาณไป START ให้ ADC ทำการวัดจากช่องที่ 1
50 out 754,8
70 out 754,0
100 x = inp(752)*5/255 *** อ่านข้อมูลจากพอร์ท A
110 print"                                Input Voltage 1 = ";x *** แสดงข้อมูลที่อ่านได้
120 goto 80
130 out 754,1
150 out 754,9
170 out 754,1
200 x = inp(752)*5/255
210 print"                                Input Voltage 2 = ";x
230 out 754,2
250 out 754,10
270 out 754,2
300 x = inp(752)*5/255
310 print"                                Input Voltage 3 = ";x
330 out 754,3
350 out 754,11
370 out 754,3
400 x = inp(752)*5/255
410 print"                                Input Voltage 4 = ";x
430 out 754,4
450 out 754,12
470 out 754,4
500 x = inp(752)*5/255

```

```

510 print"                               Input Voltage 5 = ";x
530 out 754,5
550 out 754,13
570 out 754,5
600 x = inp(752)*5/255
610 print"                               Input Voltage 6 = ";x
630 out 754,6
650 out 754,14
670 out 754,6
700 x = inp(752)*5/255
710 print"                               Input Voltage 7 = ";x
730 out 754,7
750 out 754,15
770 out 754,7
800 x = inp(752)*5/255
810 print"                               Input Voltage 8 = ";x
820 for i = 1 to 1500 :next i ***หน่วยเวลาในการอ่านรอบต่อไป
830 cls
840 goto 15

```

เท่าที่กล่าวมาโดยคร่าว ๆ นี้คงจะทำให้ผู้ที่ไม่คุ้นเคยกับการนำคอมพิวเตอร์มาใช้กับเครื่องมือ ได้มองเห็นแนวและประโยชน์ในอีกลักษณะหนึ่งของคอมพิวเตอร์ซึ่งแทนที่จะนำมาใช้เพียงเพื่อเก็บข้อมูลหรือพิมพ์รายงานเท่านั้น

ตัวอย่างโปรแกรมภาษาปาสคาล

Program Analog Digital;

{ โปรแกรมทดสอบ การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล
กำหนดให้พอร์ท A, B เป็นอินพุตพอร์ท พอร์ม C เป็นเอาต์พุต }

uses crt, dos ;

var data : real ;

channel : integer ;

hour, minute, second, sec100 : word ;

{ โปรแกรมย่อย กำหนดช่องต่อสัญญาณ }

function inport (ich : integer) : real ;

var val : real ;

begin

delay(100) ; { หน่วงเวลา 100 มิลลิวินาที }

port[754] := ich ; { ส่งสัญญาณไป Start ให้ ADC ทำการวัดสัญญาณช่อง ICH }

delay(300) ;

val := port[752]*5./255.0 ; { อ่านข้อมูลจากพอร์ท A }

inport := val ;

end ;

```

begin          { โปรแกรมหลัก }

    clrscr ;
    textcolor (yellow) ;
    textbackground (red) ;
    gotoxy(25,2) ; write (' Analog to Digital ') ;
    gotoxy(25,20);write(' Press anykey to stop ');
    textcolor(yellow);textbackground(0);
    gotoxy(20,4);write(' Enter Channel input(0,1,2,3,4) : ');
    readln(channel);
    port[755]:=146;   { กำหนดให้พอร์ท A,B เป็นอินพุตพอร์ท }
    gotoxy(20,8);write(' Time      :      ');
    gotoxy(20,9);write(' Voltage   :      Volts');
    repeat
        gettime(hour,minute,second,sec100); { อ่านเวลา ฟังก์ชันใน DOS }
        gotoxy(33,8);write('      ');
        gotoxy(33,9);write('      ');
        data:=inport(channel);                { อ่านข้อมูลจากช่องที่ ich }
        gotoxy(34,8);write(hour:2,':',minute:2,':',second:2);
        gotoxy(34,9);write(data:7:3);
        delay(2000);
    until keypressed;
    clrscr;

end.

```