



การออกแบบเครื่องควบคุมที่โปรแกรมได้ด้วย

ไมโครโปรเซสเซอร์

PROGRAMMABLE CONTROLLER (Z 84C11)

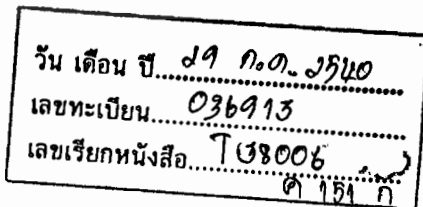


โดย

นาย คมเพชร แก้วเหล็ก 36012092

นาย ชลิต สารอด 36012094

นาย นิพนธ์ พุทธิมา 36012104



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขา วิศวกรรมการวัดคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขา วิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง

การออกแบบเครื่องควบคุมที่โปรแกรมได้ด้วยไมโครโปรเซสเซอร์  
PROGRAMMABLE CONTROLLER (Z84C11)

ผู้จัดทำ

นาย คมเพชร แก้วเหล็ก 36012092

นาย ชลิต สารอด 36012094

นาย นิพนธ์ พุทธิมา 36012104

  
.....อาจารย์ที่ปรึกษา  
(อาจารย์ วิริยะ กองรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบเครื่องควบคุมที่โปรแกรมได้ด้วย  
ไมโครโปรเซสเซอร์  
PROGRAMMABLE CONTROLLER (Z 84C11)

โดย นาย คมเพชร แก้วเหล็ก 36012092  
นาย ชลิต สารอด 36012094  
นาย นิพนธ์ พุทธิมา 36012104

อาจารย์ที่ปรึกษา

อาจารย์ วิริยะ กองรัตน์

**บทคัดย่อ**

โครงการ การออกแบบเครื่องควบคุมที่โปรแกรมได้ด้วยไมโครโปรเซสเซอร์  
ฉบับนี้ เป็นการศึกษาถึงการเขียนโปรแกรมควบคุมการทำงาน ของเครื่องควบคุม  
ที่สามารถโปรแกรมได้ โดยอาศัยโครงสร้างคำสั่งของ PLC ( Programmable  
Logic Control ) แต่จะนำไปใช้ควบคุมกระบวนการในระบบ PID

ลักษณะการควบคุมกระบวนการ ของตัวควบคุมจะ ประกอบไปด้วยโหมดการ  
ทำงาน 2 โหมด คือโหมด RUN และโหมด EDITOR โดยที่ทั้ง 2 โหมดสามารถจะ  
เซตค่าขณะทำงานในโหมดนั้น ๆ ได้ ซึ่งการทำตัวควบคุมในลักษณะนี้ทำให้ง่าย และ  
สะดวกในการควบคุมกระบวนการ โดย CPU ที่ใช้เป็นตัวควบคุมจะใช้ CPU เบอร์  
Z84C11

## **ABSTRACT**

Project **Desing of programmable controller (Z84C11)** is study to writting programe for control operating of programmable controller by uses the algorithm command of programmable logic control ( PLC ) but it will be use for control process in PID system.

The characteristic of this controller compose with two modes. The first is editor mode. The other is run mode. The editor mode uses for programming and setting parameter value. The run mode uses for run programe. This controller give us easy and comfortable for control process.



## กิตติกรรมประกาศ

โครงการนี้สามารถสำเร็จล่วงตามวัตถุประสงค์ที่วางไว้ เนื่องจากได้รับคำแนะนำและความช่วยเหลือจากอาจารย์หลายท่าน คณะผู้จัดทำโครงการขอขอบพระคุณอาจารย์ทุกท่านในภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรมเป็นอย่างสูง ไว้ ณ.ที่นี้ และขอขอบคุณเพื่อนๆทุกท่านที่ช่วยเหลือและให้กำลังใจมาโดยตลอด

คณะผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ

### บทที่ 1 ชนิดของเครื่องควบคุมแบบป้อนกลับ

- เครื่องควบคุมแบบสัดส่วน [ Proportional Controller (P) ] 1-1
- เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัล [ Proportional - Integral Control (PI) ] 1-6
- เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัลและเดอริเวทีฟ [Proportional - Integral - Derivative Control (PID) ] 1-9
- เครื่องควบคุมแบบสัดส่วนและเดอริเวทีฟ [Proportional - Derivative (PD) ] 1-11
- เครื่องควบคุมแบบดิจิทัล [Digital Controller] 1-12
- รีเซ็ต - วินด์อัฟ 1-13

### บทที่ 2 คุณสมบัติของ CP - Z84C11 PLUS

- ข้อมูลของบอร์ด CP - Z84C11 PLUS 2-1
- Z80 BUS 2-3
- CTC 2-4
- WATCH DOG (WDTOUT) 2-5

### บทที่ 3 รายละเอียดและหลักการของโปรแกรม

- การจัดตำแหน่งของโปรแกรมที่เขียนบน Z84C11 DEBUGGER 3-1
- การกำหนดตำแหน่งของพอร์ท 3-1
- โฟลชาร์ทการทำงานของโปรแกรม 3-2

### บทที่ 4 หลักการทำงานและการแสดงผลของเครื่องควบคุม

- คำสั่งพื้นฐานสำหรับการเขียนโปรแกรม 4-1
- การใช้งานโปรแกรมคอนโซล 4-2
- การใช้ฟังก์ชันคีย์ 4-5
- หลักการเขียนโปรแกรม 4-6

# บทที่ 5 บทสรุป

-ผลการทดลอง

5-1

ภาคผนวก ก โปรแกรม

ภาคผนวก ข รายละเอียดข้อมูล

บรรณานุกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1

## ชนิดของเครื่องควบคุมแบบป้อนกลับ

หนทางหนึ่งที่เครื่องควบคุมแบบป้อนกลับจะทำการตัดสินใจเพื่อที่จะรักษาผลของการควบคุมไว้ที่จุดเป้าหมายโดยการคำนวณค่าทางออกบนพื้นฐานของความแตกต่างระหว่างตัวแปรที่ถูกควบคุมและค่าเป้าหมาย ในส่วนนี้เราจะมองถึงเครื่องควบคุมที่พบกันทั่วไปโดยคู่ที่สมการซึ่งเป็นการอธิบายการทำงานของเครื่องควบคุมเหล่านี้

เครื่องควบคุมแบบสัดส่วน **Proportional Controller (P)** เครื่องควบคุมแบบสัดส่วน คือเป็นชนิดที่ง่ายที่สุดของเครื่องควบคุม โดยยกเว้นเครื่องควบคุมแบบเปิด - ปิด ซึ่งจะไม่ขอกล่าวถึงในที่นี้ สมการที่อธิบายการทำงานของเครื่องควบคุมแบบสัดส่วนเป็นดังนี้

$$m(t) = \bar{m} + Kc[r(t) - c(t)] \quad (1.1)$$

หรือ

$$m(t) = \bar{m} + Kc.e(t) \quad (1.2)$$

เมื่อ

$m(t)$  = ทางออกของเครื่องควบคุม , *psig* หรือ *mA*

$r(t)$  = ค่าเป้าหมาย , *psig* หรือ *mA*

$c(t)$  = ตัวแปรที่ถูกควบคุม , *psig* หรือ *mA* เป็นสัญญาณที่ได้จากอุปกรณ์ส่งสัญญาณ

$e(t)$  = สัญญาณผิดพลาด , *psig* หรือ *mA* เป็นความแตกต่างระหว่างค่าเป้าหมายกับตัวแปรที่ถูกควบคุม

$Kc$  = อัตราการขยายของเครื่องควบคุม , *psi / psi*, *mA / mA*

$\bar{m}$  = ค่าไบอัส , *psig* หรือ *mA* เป็นค่าทางออกของเครื่องควบคุมเมื่อสัญญาณผิดพลาดเป็นศูนย์ค่านี้ปกติจะถูกตั้งไว้ในระหว่างการปรับเทียบเครื่องควบคุมอยู่ที่กึ่งกลางสเกล คือ 9 *psig* หรือ 12 *mA*

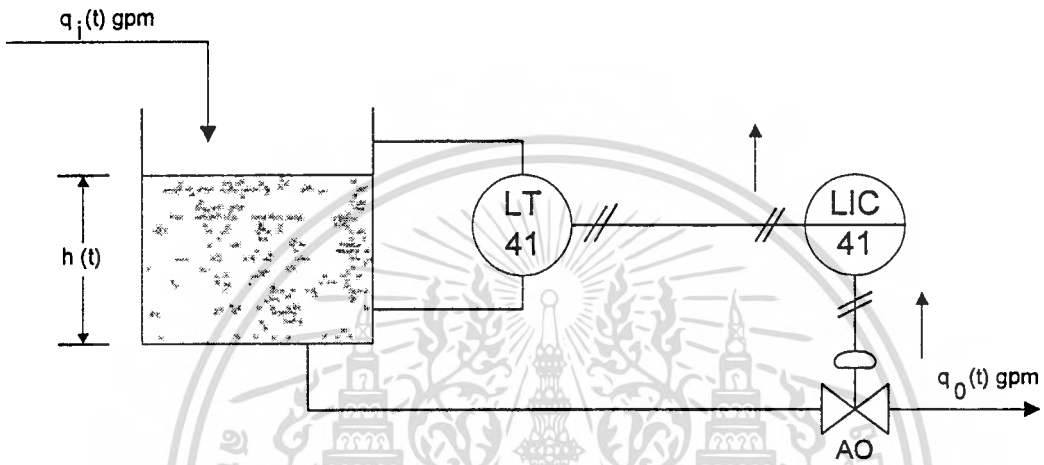
โดยที่พิกัดทางเข้าและทางออกนั้นเหมือนกัน (3-15 *psig* หรือ 4-20 *mA*) และสัญญาณทางเข้าและสัญญาณทางออกตลอดจนค่าเป้าหมายเหมือนกัน ซึ่งในบางครั้งอาจจะเป็นเศษส่วนหรือเปอร์เซ็นต์ของพิกัด มันเป็นเรื่องที่น่าสนใจในสมการที่ (1.1) ที่ถูกเขียนขึ้นมาสำหรับเครื่องควบคุมที่เป็นกริยาทางออกแบบกลับ คือถ้าตัวแปรที่ถูกควบคุมมีค่าเพิ่มมากขึ้นเหนือค่าเป้าหมายและค่าผิดพลาดกลายเป็นค่าลบ จากสมการแสดงให้เห็นว่าทางออกของเครื่องควบคุมนั้นได้ลดลงในทางที่จะแสดงให้เห็นถึงเครื่องควบคุมที่เป็นกริยาทางออกแบบตรง ในทางคณิตศาสตร์สามารถทำได้ โดยการให้อัตราการขยายของเครื่องควบคุมที่เป็นลบ แต่เราจะต้องจำไว้อย่างหนึ่งว่าในเครื่องควบคุมในอุตสาหกรรมจะไม่มีอัตราการขยายที่เป็นค่าลบแต่จะมีเฉพาะค่าบวกอย่างเดียว สวิตช์เลือก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปเผยแพร่โดยไม่เสียค่าใช้จ่าย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

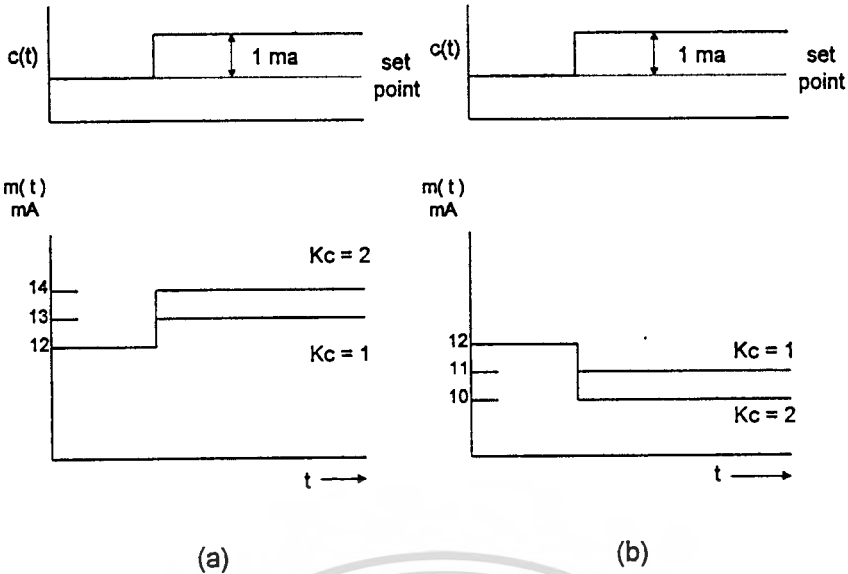
แบบกลับ/แบบตรง จึงได้ถูกนำมาใช้สำหรับอัตราการขยายที่เป็นลบถูกใช้เมื่อมีการกระทำทางคณิตศาสตร์ของระบบควบคุมเพื่อที่จะวิเคราะห์เครื่องควบคุมกรียาทางออกแบบตรง

สมการ (1.1) และ (1.2) แสดงให้เห็นทางออกของเครื่องควบคุมที่เป็นสัดส่วนกับค่าผิดพลาดระหว่างค่าเป้าหมายกับตัวแปรที่ถูกควบคุม ความเป็นสัดส่วนนี้ถูกกำหนดโดยค่าอัตราขยายของเครื่องควบคุม อัตราการขยายหรือความไวของเครื่องควบคุมนี้จะบอกให้รู้ว่าทางออกจะมีขนาดมากเท่าใดจากการเปลี่ยนแปลงของค่าผิดพลาดที่เกิดขึ้นอันนี้จะได้แสดงในรูปที่ 1-2

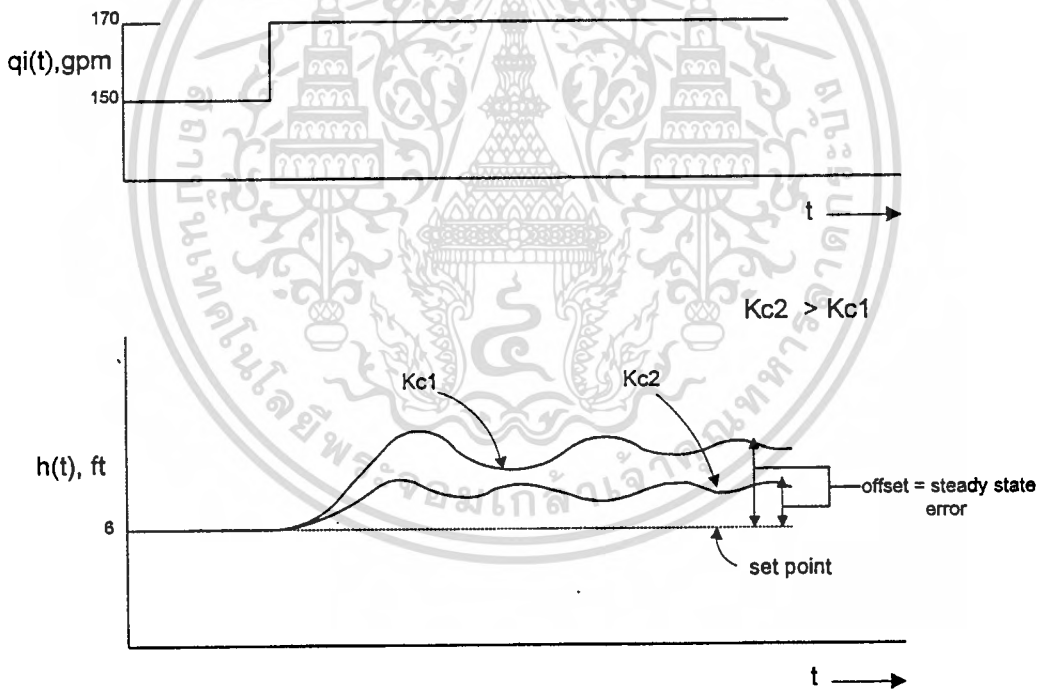


รูปที่ 1-1 วงจรการควบคุมของระดับของเหลว

เครื่องควบคุมแบบสัดส่วนอย่างเดียวมีข้อดีคือมีตัวปรับค่าอัตราการขยายเพียงตัวเดียว อย่างไรก็ตามข้อเสียของเครื่องควบคุมแบบสัดส่วนนั้นในการทำงานจะเกิดค่าผิดพลาดที่สภาวะคงที่ (Steady - State Error) OFFSET เพื่อที่จะแสดงถึงค่าผิดพลาดที่สภาวะคงที่เป็นกราฟพิจารณวงรอบการควบคุมระดับของเหลวในรูปที่ 1-1 กำหนดให้เงื่อนไขการทำงานที่  $\bar{q}_i = \bar{q}_o = 150 \text{ gpm}$  และ  $\bar{h} = 6 \text{ ft}$  เพื่อที่จะให้ค่าที่ทางออกของวาล์วเป็น  $150 \text{ gpm}$  จะต้องให้แรงดันลมแก่วาล์ว  $9 \text{ psig}$  แต่ถ้าอัตราไหลทางเข้าเพิ่มขึ้นผลตอบสนองของระบบร่วมกับเครื่องควบคุมแบบสัดส่วนแสดงดังรูปที่ 1-3 เครื่องควบคุมได้นำค่าของตัวแปรที่ถูกควบคุมกลับเข้าสู่สภาวะคงที่แต่ค่าที่สภาวะคงที่นี้จะไม่ใช่ค่าที่จุดเป้าหมาย ความแตกต่างระหว่างค่าเป้าหมายและค่าที่สภาวะคงที่ของตัวแปรที่ถูกควบคุมคือออฟเซ็ท รูปที่ 1-3 แสดงให้เห็นผลตอบสนองของค่าตัวแปรที่ถูกควบคุมทั้งสองค่าที่แตกต่างกันของค่าอัตราการขยายที่ถูกปรับเปลี่ยน จะพบว่าค่าอัตราการขยายของเครื่องควบคุมที่มากจะให้ค่าออฟเซ็ทน้อยแต่จะเกิดการแกว่งมากกว่า อย่างไรก็ตามจะมีค่าที่มากที่สุดของอัตราการขยายของเครื่องควบคุมที่ทำให้กระบวนการเข้าสู่ความไม่เสถียรภาพการคำนวณค่าที่มากที่สุดของอัตราการขยายจะนี้เรียกว่า Ultimate gain



รูปที่ 1-2 ผลของอัตราขยายในเครื่องควบคุมต่อทางออกของเครื่องควบคุม  
 a) เครื่องควบคุมกริยาทางออกแบบตรง b) เครื่องควบคุมกริยาทางออกแบบกลับ



รูปที่ 1-3 ผลตอบสนองของกระบวนการควบคุมระดับของเหลว

การพิจารณาระบบควบคุมระดับของเหลวเดียวกันนี้ในรูปที่ 1-1 ด้วยเงื่อนไขของการทำงานเหมือนเดิม และจำไว้ว่าอย่างหนึ่งว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวที่กริยาทางออกแบบตรงนั้นอัตราขยายเป็นลบ ( $-K_c$ ) ดูสมการต่อไปนี้

$$m(t) = 9 + (-K_c).e(t) \tag{1.3}$$

ตอนนี้กำหนดให้อัตราไหลทางเข้าเพิ่มขึ้นเป็น 170 *gpm* เมื่อเป็นเช่นนี้แล้วระดับของเหลวจะต้องเพิ่มขึ้นจากนั้นเครื่องควบคุมจะส่งสัญญาณทางออกไปเปิดวาล์วเพื่อที่จะทำให้ระดับของเหลวกลับ

นอกจากนี้ยังมีข้อควรระวังในการใช้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้าสู่สภาวะคงที่ในตำแหน่งเดิม อัตราไหลทางออกตอนนี้จะต้องเป็น 170 *gpm* เพื่อที่จะให้วาล์วผ่านอัตราไหลค่าใหม่ที่เพิ่มขึ้นก็จะต้องเปิดให้มากขึ้นกว่าเดิม วาล์วที่ใช้เป็นแบบแรงดันเพื่อเปิดสมมติให้ค่าแรงดันค่าใหม่ที่กระทำกับวาล์วนี้เป็น 10 *psig* นั่นคือเครื่องควบคุมจะต้องให้ทางออกเป็น 10 *psig* แก้วาล์ว มองกลับไปที่สมการ (1.3) จะเห็นว่ามืออย่างเดียวกันเท่านั้นที่ทางออกของเครื่องควบคุมเท่ากับ 10 *psig* ได้ก็ต่อเมื่อเทอมที่สองทางขวามือจะต้องเป็น +1 *psig* และในเทอมค่าผิดพลาดไม่สามารถที่จะเป็นศูนย์ที่สภาวะคงที่ได้ และค่าผิดพลาดที่สภาวะคงที่ก็คือออฟเซ็ท ค่าผิดพลาดที่เป็นลบอันเนื่องมาจากค่าของตัวแปรที่ถูกควบคุมมีค่ามากกว่าค่าเป้าหมาย

มือคู่สองจุดที่น่าสนใจในตัวอย่างนี้ อันแรกคือขนาดของเทอมความผิดพลาดขึ้นอยู่กับค่าอัตราการขยายของเครื่องควบคุมโดยที่ทั้งเทอมรวมกันแล้วมีค่าเท่ากับ +1 *psig* ดังนี้

$-K_c$	$-e(\infty)(\text{offset})$
1	1
2	0.5
4	0.25

จากที่กล่าวมาแล้วว่าอัตราการขยายที่มากจะเกิดออฟเซ็ทน้อย ผู้อ่านจะต้องจำไว้ว่าอัตราขยายที่มากกว่าขอบเขตอันหนึ่งแล้วจะทำให้ขบวนการเข้าสู่ความไม่เสถียรภาพ

อันดับที่สองเพื่อเป็นการสรุปในตัวอย่างนี้ จะเห็นว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวนี้จะพยายามทำหน้าที่โดยให้การเปลี่ยนแปลงนั้นเข้าสู่สภาวะคงที่ และจำนวนของความเบี่ยงเบนจากค่าเป้าหมายหรือออฟเซ็ท ขึ้นอยู่กับอัตราการขยายของเครื่องควบคุม

ผู้ผลิตเครื่องควบคุมส่วนมากจะไม่ใช้เทอมของอัตราการขยายเพื่อที่จะบ่งบอกความไวของเครื่องควบคุม แต่จะใช้เทอมของ Proportional Band PB โดยจะเป็นความสัมพันธ์กับอัตราขยาย

$$\text{คือ } PB = \frac{100}{K_c} \quad (1.4)$$

ดังนั้น สมการที่ได้อธิบายสำหรับเครื่องควบคุมแบบสัดส่วนเขียนใหม่ดังนี้

$$m(t) = \bar{m} + \frac{100}{PB}(r(t) - c(t)) \quad (1.5)$$

หรือ

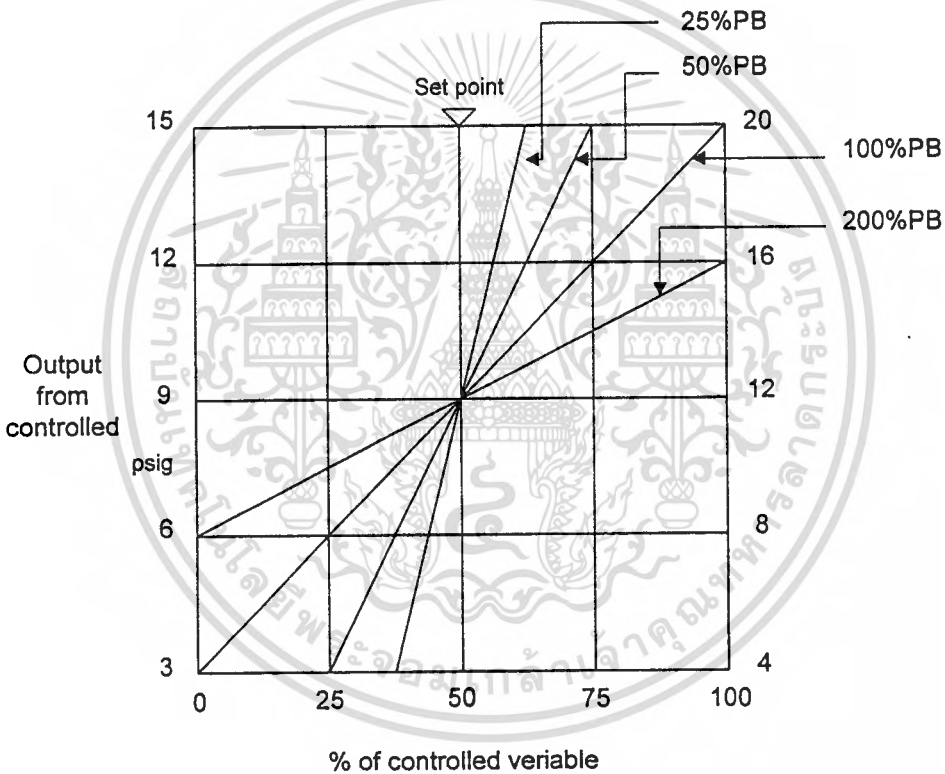
$$m(t) = \bar{m} + \frac{100}{PB}e(t) \quad (1.6)$$

เทอม 100 ถูกนำมาใช้เพราะว่า PB โดยปกติจะอ้างถึง Percent Proportional Band

สมการ (1.6) แสดงให้เห็นจริงว่าอัตราการขยายมากก็เหมือนกับมี Proportional Band ต่ำหรือแคบ และถ้าอัตราขยายต่ำแล้วจะเหมือนกับมี Proportional Band มากหรือกว้าง นั่นก็หมายความว่าก่อนที่จะเริ่มใช้งานเครื่องควบคุมอย่างน้อยที่สุดก็ต้องรู้ว่าเครื่องควบคุมนี้มีการปรับแบบ

อัตราการขยายหรือปรับ Proportional band

คำจำกัดความอีกอย่างหนึ่งของ Proportional Band ก็คือจะมองที่ผิดพลาด (ที่กระจายอยู่ในเทอมของเปอร์เซ็นต์พิคคของตัวแปรที่ถูกควบคุม) ที่ต้องการที่จะเคลื่อนค่าที่ทางออกของเครื่องควบคุมจากค่าที่ต่ำสุดไปสู่สูงสุด พิจารณาวงรอบการควบคุมกระบวนการแลกเปลี่ยนความร้อน อุปกรณ์ส่งสัญญาณมีพิคคจาก 100-300 องศาเซลเซียส และค่าเป้าหมายของเครื่องควบคุมอยู่ที่ 200 องศาเซลเซียส รูปที่ 1-4 จะแสดงกราฟที่เป็นคำจำกัดความของ PB ในรูปนี้แสดงค่า PB 100% ก็แสดงว่าค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 100% ของพิคค ถ้า PB 50% เช่นเดียวกันก็แสดงว่าค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 50% ของพิคค และถ้า PB 200% ถูกต้องค่าตัวแปรที่ถูกควบคุมสามารถที่จะแปรค่าได้ 200% ของพิคค แต่อย่างไรก็ตามเครื่องควบคุมแบบสัดส่วนอย่างเดียวเครื่องนี้ไม่สามารถที่จะให้ค่าทางออกเกินกว่าพิคคที่เป็นอยู่ได้ PB 200% จะหมายถึงว่าความไว้น้อยมากต่อความผิดพลาดหรืออัตราการขยายน้อยมากนั่นเอง



Controller output

	3 psig	9 psig	5 psig
	4 mA	2 mA	20 mA
PB = 100%	100 C	200 C	300 C
PB = 50%	150 C	200 C	300 C
PB = 25%	175 C	200 C	300 C
PB = 200%		200 C	300 C

รูปที่ 1-4 คำจำกัดความของ Proportional Band

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะหาฟังก์ชันถ่ายโอนของเครื่องควบคุมแบบสัดส่วนอย่างเดียว สมการ 5-27 สามารถจะเขียนได้ดังนี้

$$m(t) - \bar{m} = Kc.(e(t) - 0) \quad (1.7)$$

กำหนดตัวแปรเบี่ยงเบนได้ดังนี้

$$M(t) = m(t) - \bar{m} \quad (1.8)$$

$$E(t) = e(t) - 0 \quad (1.9)$$

ดังนั้น

$$M(t) = Kc.E(t) \quad (1.10)$$

แปลงลาปลาซจะได้ฟังก์ชันถ่ายโอนดังนี้

$$\frac{M(s)}{E(s)} = Kc \quad (1.11)$$

สรุปได้ว่าเครื่องควบคุมแบบสัดส่วนอย่างเดียวเป็นนั้นเป็นชนิดที่ง่ายมากด้วยข้อดีที่มีพารามิเตอร์เพียงตัวเดียวในการปรับแต่งอัตราขยายหรือ PB ส่วนข้อเสียก็คือว่าการทำงานของเครื่องควบคุมแบบนี้จะเกิดออฟเซ็ทแก่ตัวแปรที่ถูกควบคุมในกรณีที่เครื่องควบคุมแบบสัดส่วนอย่างเดียวสามารถจะควบคุมได้ก็เป็นกระบวนการที่ตัวแปรที่ถูกควบคุมอยู่ในขอบเขตของค่าเป้าหมายอย่างไรก็ตามกระบวนการที่การควบคุมจะต้องอยู่ที่ค่าเป้าหมายเพียงค่าเดียวนั้นเครื่องควบคุมดังกล่าวไม่สามารถนำมาใช้งานได้

### เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัล Proportional-Integral Controller

(PI) กระบวนการส่วนใหญ่ไม่สามารถที่จะถูกควบคุมให้เกิดค่าออฟเซ็ทได้โดยค่าตัวแปรที่ถูกควบคุมจะต้องอยู่ที่ค่าเป้าหมายเลยจำเป็นจะต้องมีเทอมบางเทอมที่ชาญฉลาดกระทำร่วมกับการทำงานแบบสัดส่วนอย่างเดียวของเครื่องควบคุมเพื่อที่จะขจัดค่าออฟเซ็ทออกไป และเทอมที่ชาญฉลาดอันนี้เองคือ กริยาอินทิกรัล หรือ รีเซ็ท เพราะฉะนั้นเครื่องควบคุมก็จะกลายเป็นแบบสัดส่วนร่วมกับอินทิกรัล โดยได้ถูกจำกัดความไว้ในสมการดังนี้

$$m(t) = \bar{m} + Kc[r(t) - c(t)] + \frac{Kc}{\tau_i} \int [r(t) - c(t)] dt \quad (1.12)$$

หรือ

$$m(t) = \bar{m} + Kc[r(t) - c(t)] + \frac{Kc}{\tau_i} \int e(t) dt \quad (1.13)$$

เมื่อ  $\tau_i$  = เวลาในการอินทิเกรต หรือ เวลารีเซ็ท นาที/ จำนวนครั้ง

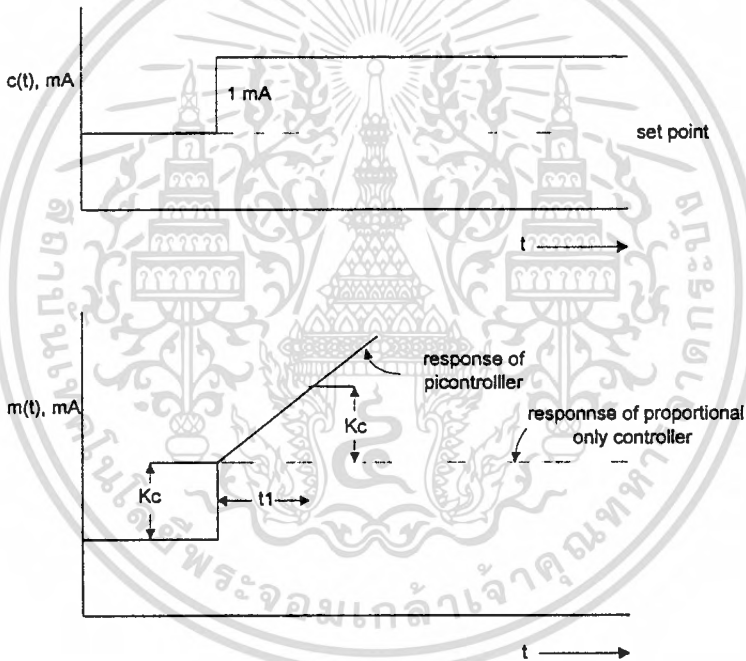
เพราะฉะนั้นเครื่องควบคุมแบบ PI จะมีสองพารามิเตอร์คืออัตราการขยายและเวลาในการอินทิเกรตซึ่งจะต้องถูกปรับแต่งเพื่อให้การควบคุมนั้นเหมาะสมที่สุด เพื่อที่จะให้เข้าใจคุณลักษณะ

ของเวลารีเซ็ท  $\tau_i$  โดยพิจารณาตัวอย่างสมมติที่แสดงในรูปที่ 1-5  $\tau_i$  คือเวลาที่จะทำให้เครื่อง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมกระทำกริยาแบบสัดส่วนที่ซ้ำกัน โดยหน่วยของเวลาอันนี้เป็นนาที/จำนวนครั้ง ค่าเวลาที่เล็กของ  $\tau_i$  จะทำให้เส้นโค้งของผลตอบสนองนั้นชันมาก ซึ่งจะหมายความว่า เป็นผลตอบสนองที่เร็ว อีกทางหนึ่งที่จะใช้ขยายความโดยดูที่สมการ (1.13) ค่าของ  $\tau_i$  ที่เล็กจะเป็นค่าที่มากในเทอมอยู่น้ำอินทิเกรต และจะเป็นตัวคูณของกริยาอินทิเกรตหรือรีเซ็ต

ต่อไปเราจะมาดูระบบควบคุมระดับของเหลวอีกครั้งที่จะถูกใช้สำหรับอธิบายว่าทำไมออฟเซ็ทจึงเกิดขึ้น เมื่ออัตราไหลทางเข้าเพิ่มขึ้นเป็น  $170 \text{ gpm}$  อัตราการไหลทางออกก็จำเป็นจะต้องพยายามให้เป็น  $170 \text{ gpm}$  ด้วย เพื่อให้มีอัตราไหลผ่านที่วาล์ว  $170 \text{ gpm}$  วาล์วจะต้องเปิดกว้างโดยมีแรงดันลมเพื่อเปิดเป็น  $10 \text{ psig}$  ด้วยเครื่องควบคุมแบบ PI ไม่ว่าค่าผิดพลาดจะเกิดด้วยเวลาที่ยาวนานเท่าใดเครื่องควบคุมก็จะเก็บค่าผิดพลาดนี้รวมกันไว้เพื่อชดเชยให้ที่ทางออกของเครื่องควบคุมในการแก้ความผิดพลาดและความผิดพลาดนั้นก็จะหมดไป เมื่อค่าผิดพลาดหมดไปแล้วทางออกของเครื่องควบคุมจะอยู่ในสภาวะ



รูปที่ 1-5 แสดงผลตอบสนองของเครื่องควบคุมแบบ PI (แบบกริยาทางออกตรง)

ต่อการเปลี่ยนแปลงของความผิดพลาดแบบเป็นขั้น

$$m(t) = \bar{m} + \frac{K_c}{\tau_i} (\int_0^t dt) \quad (1.14)$$

ในความเป็นจริงแล้วถ้าความผิดพลาดเป็นศูนย์ไม่ได้หมายความว่าเทอมอินทิเกรตจะเป็นศูนย์ด้วย แต่จะหมายถึงเครื่องควบคุมกำลังอินทิเกรตค่าผิดพลาดที่เป็นศูนย์โดยการบวกด้วยศูนย์เข้าไปที่ทางออกซึ่งยังคงที่อยู่ สำหรับกระบวนการระดับของเหลวนี้อินทิเกรตเทอมจะมีค่าเท่า  $1 \text{ psig}$  เพราะฉะนั้นทางออกของเครื่องควบคุมจะเป็น  $10 \text{ psig}$  โดยปราศจากค่าผิดพลาดและที่เกิดขึ้นนี้

ได้เป็นการขยายความว่าทำไมกริยารีเซ็ตจึงได้ขจัดออฟเซ็ทออกไปได้

$$\frac{Kc}{\tau_i} \int_0^t dt \quad (1.15)$$

บางผู้ผลิตจะไม่ใช้เทอมของเวลารีเซ็ต  $\tau_i$  สำหรับเป็นค่าปรับแต่งแต่จะใช้อัตราการรีเซ็ต reset rate

$$\frac{R}{I} = \frac{1}{\tau_i}, \text{ จำนวนครั้ง/นาที} \quad (1.16)$$

เพราะฉะนั้นก่อนที่จะทำการปรับพารามิเตอร์เหล่านี้จะต้องรู้ว่าเครื่องควบคุมใช้ค่าของเวลารีเซ็ตหรืออัตราการรีเซ็ต ต่อไปจะขอแสดงสมการสำหรับเครื่องควบคุมแบบ PI ในอุตสาหกรรมที่ผู้ผลิตได้กำหนดไว้

Foxboro Co.

$$m(t) = \bar{m} + \frac{100}{PB} e(t) + \frac{100}{PB \cdot \tau_i} \int e(t) dt \quad (1.17)$$

Fisher Controls

$$m(t) = \bar{m} + \frac{100}{PB} e(t) + \frac{100\tau_i R}{PB I} \int e(t) dt \quad (1.18)$$

Taylor Co. Honeywell, Inc.

$$m(t) = \bar{m} + Kc \cdot e(t) + Kc \cdot \tau_i \frac{R}{I} \int e(t) dt \quad (1.19)$$

เป็นสิ่งที่น่าสนใจคือ Honeywell ได้ทำการพัฒนาเครื่องควบคุมที่ใช้ไมโครโปรเซสเซอร์ รุ่น TDC 2000 จะใช้สมการ (1.13) ส่วน Fisher Controls ได้ใช้ไมโครโปรเซสเซอร์ รุ่น PROVOX ใช้สมการ (1.16) และ (1.17) สมการที่ (1.13) ก็เป็นอันหนึ่งที่มีจุดประสงค์เพื่อเป็นมาตรฐานโดยสมาคมเครื่องมือวัดแห่งอเมริกา Instrument Society of America (ISA) สิ่งที่สำคัญที่จะต้องจดจำก็คือความสัมพันธ์ระหว่างอัตราขยายกับ Proportional Band และความสัมพันธ์ระหว่างค่ารีเซ็ตกับอัตรารีเซ็ต

เพื่อที่จะหาฟังก์ชันถ่ายโอนสำหรับเครื่องควบคุมแบบ PI สมการ (1.13) เขียนได้ดังต่อไปนี้

$$m(t) - \bar{m} = Kc(e(t) - 0) + \frac{Kc}{\tau_i} \int (e(t) - 0) dt \quad (1.20)$$

โดยใช้ข้อกำหนดเดิมของตัวแปรที่เบี่ยงเบนในสมการ (1.9) และ (1.10) แปลงลาปลาซจัดรูปใหม่จะได้

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} \right) \quad (1.21)$$

สรุปเครื่องควบคุมแบบสัดส่วนร่วมกับอินทิเกรตจะมีสองพารามิเตอร์ที่จะต้องปรับแต่งได้แก่ อัตราขยายหรือ Proportional Band และค่าเวลารีเซ็ต ข้อดีของเครื่องควบคุมแบบนี้ก็คือ กริยาอินทิกรัลหรือรีเซ็ตจะทำการขจัดออฟเซตให้หมดไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

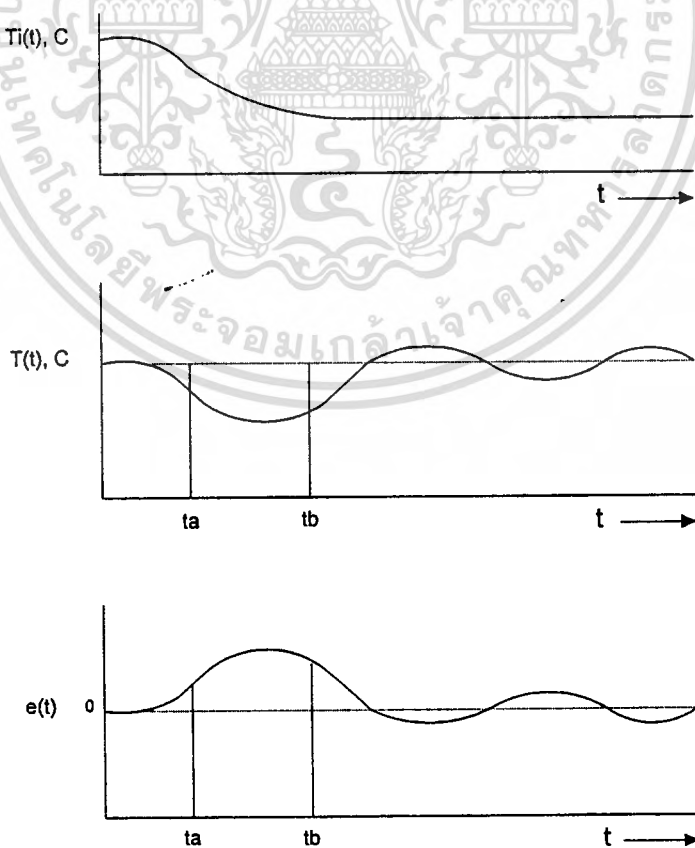
เครื่องควบคุมแบบสัดส่วนร่วมกับอินทิกรัลและเดอริเวทีฟ Proportional-Integral-Derivative Controller (PID) โหมดการควบคุมอันใหม่นี้ได้เพิ่มเข้าไปกับเครื่องควบคุมแบบ PI ก็คือ เดอริเวทีฟ หรืออาจเรียกกริยาของอัตราการเปลี่ยนแปลง rate หรือ peract . จุดประสงค์ก็เพื่อที่จะเป็นการคาดการณ์ล่วงหน้าเมื่อกระบวนการทำงานโดยมองที่อัตราการเปลี่ยนแปลงต่อเวลาของความผิดพลาด ซึ่งมันเป็นค่าของอนุพันธ์ สมการที่ใช้อธิบายเป็นดังต่อไปนี้

$$m(t) = \bar{m} + K_c e(t) + \frac{K_c}{\tau_i} \int e(t) dt + K_c \tau_D \frac{de(t)}{dt} \quad (1.22)$$

เมื่อ  $\tau_D$  = เดอริเวทีฟ หรือ อัตราหน่วยเป็นเวลา, นาที

เพราะฉะนั้น เครื่องควบคุมแบบ PID จะมีสามพารามิเตอร์ที่จะต้องปรับแต่งเพื่อให้การควบคุมนั้นเหมาะสม ส่วนค่าเวลาเดอริเวทีฟจะเป็นหน่วยของเวลาที่ทุกผู้ผลิตต่างใช้เป็นแบบเดียวกันและจากที่กล่าวมากริยาเดอริเวทีฟจะทำให้เครื่องควบคุมมีความสามารถในการคาดเดาเหตุการณ์ล่วงหน้าได้ด้วยการมองเหตุการณ์ไปข้างหน้า โดยการคำนวณค่าอนุพันธ์ของความผิดพลาด จำนวนของค่าคาดการณ์จะมากน้อยเท่าใดก็จะขึ้นอยู่กับพารามิเตอร์ของเวลาเดอริเวทีฟที่ได้ถูกปรับแต่ง

ต่อไปจะพิจารณากระบวนการแลกเปลี่ยนความร้อนและใช้เพื่อให้ความหมายของค่าว่าการคาดเดาเหตุการณ์ล่วงหน้า สมมุติว่าอุณหภูมิทางเข้าของของเหลวในกระบวนการลดลงไปจำนวนหนึ่งและอุณหภูมิทางออกเริ่มที่จะลดลงที่จะได้แสดงในรูปที่ 1-6



รูปที่ 1-6 การควบคุมในกระบวนการแลกเปลี่ยนความร้อน

ที่เวลา  $t_a$  จำนวนของค่าผิดพลาดเป็นบวกแต่อาจจะเล็กน้อย จำนวนของการแก้ไขค่าผิดพลาดของระบบควบคุมโดยเฉพาะค่าการแก้ไขแบบสัดส่วนและอินทิกรัลจะน้อย อย่างไรก็ตามค่าเดริเวทีฟของความผิดพลาดอันนี้โดยสังเกตจากความชันของความผิดพลาดจะมีค่ามากและเป็นบวกจะเป็นผลให้ค่าการแก้ไขเฉพาะเดริเวทีฟนั้นมีค่ามาก เครื่องควบคุมจะรู้ว่าตัวแปรที่ถูกควบคุมกำลังมุ่งหน้าออกจากเป้าหมายด้วยความเร็ว และที่เวลา  $t_b$  ค่าผิดพลาดก็ยังคงเป็นบวกและมากขึ้นกว่าเดิม จำนวนของค่าแก้ไขของแบบสัดส่วนและแบบอินทิกรัลนั้นก็มากกว่าแต่ก่อนและก็ยังคงรวมเข้าไปกับค่าแก้ไขเพื่อที่จะเปิดวาล์วให้กว้างขึ้น แต่ค่าเดริเวทีฟของความผิดพลาดที่เวลานี้จะเป็นลบ เป็นผลให้ค่าที่แก้ไขก่อนหน้านี้ถูกลดลง นั่นคือตัวแปรที่ถูกควบคุมได้เริ่มที่จะกลับมาสู่ค่าเป้าหมายที่จุดนี้เองค่าเดริเวทีฟก็เริ่มที่หักค่าแก้ไขของตัวเองออกจากค่าแก้ไขทั้งหมด เมื่อเป็นเช่นนี้ก็จะต้องใช้เวลาที่ยาวนานขึ้นเพื่อที่จะทำให้ค่าตัวแปรที่ถูกควบคุมกลับเข้าหาเป้าหมายอย่างไรก็ตามการพุ่งเลยขอบเขตและการแกว่งรอบค่าเป้าหมายจะถูกลดลงไป

เครื่องควบคุมแบบ PID ถูกใช้ในกระบวนการที่มีค่าเวลาคงตัวที่ยาวนาน ยกตัวอย่างเช่นวงรอบการควบคุมของอุณหภูมิและความเข้มข้น กระบวนการที่มีค่าเวลาคงตัวที่สั้น (ค่าความจุที่น้อย) จะเร็วและอ่อนไหวต่อสัญญาณรบกวนในกระบวนการมาก กระบวนการเหล่านี้ได้แก่วงรอบการควบคุมของอัตราไหลและการควบคุมความดันของกระแสของเหลว (ของเหลวที่พุ่งเป็นลำ เช่น น้ำพุ) พิจารณาผลการบันทึกของอัตราการไหลดังรูปที่ 1-7 เมื่อนำโหมดเดริเวทีฟมาใช้โดยจะเห็นว่าเป็นการขยายผลต่อสัญญาณรบกวนเท่านั้นเพราะว่าค่าเดริเวทีฟจะไวต่อการเปลี่ยนแปลงของสัญญาณรบกวนซึ่งจะมีค่ามาก กระบวนการที่มีเวลาคงตัวมาก (ค่าความจุที่มาก) โดยปกติจะเป็นแบบถูกหน่วงหรือช้าและไม่ไวต่อสัญญาณรบกวน อย่างไรก็ตามถ้าจะใช้เครื่องควบคุมแบบ PID กับกระบวนการการอุณหภูมิแล้วก็ต้องตรวจสอบอุปกรณ์ส่งสัญญาณที่ไม่เป็นแหล่งกำเนิดสัญญาณรบกวนเสียเอง

ฟังก์ชันถ่ายโอนของเครื่องควบคุมแบบ PID ในอุดมคติจะเป็นในสมการ (1.19) ดังต่อไปนี้



รูปที่ 1-7 แสดงผลการบันทึกอัตราการไหล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$m(t) - \bar{m} = Kc(e(t) - 0) + \frac{Kc}{\tau_i} \int (e(t) - 0) dt + Kc\tau_D \frac{d(e(t) - 0)}{dt} \quad (1.23)$$

โดยใช้ข้อกำหนดที่แล้วมาของตัวแปรที่ถูกเบี่ยงเบนโดยสมการ (1.9) และ (1.10) แปลงลาปลาซ และจัดรูปจะได้

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} + \tau_D s \right) \quad (1.24)$$

ฟังก์ชันถ่ายโอนนี้ถูกเรียกว่าเป็นอุดมคติเพราะว่าในทางปฏิบัติในค่าของเดอริเวทีฟเป็นไปไม่ได้ที่จะทำการคำนวณหา ดังนั้นค่าเดอริเวทีฟจะถูกประมาณโดยใช้เทอมของ นำหน้า/ล่าหลัง ผลที่ได้จะเป็นฟังก์ชันถ่ายโอนที่ใกล้เคียงและแม่นยำกว่า

$$\frac{M(s)}{E(s)} = Kc \left( 1 + \frac{1}{\tau_i s} \right) \left( \frac{\tau_D s + 1}{\alpha \tau_D s + 1} \right) \quad (1.25)$$

ค่าของ  $\tau_{DS}$  อยู่ในพิคเจอร์ระหว่าง 0.05 ถึง 0.1 รูปเครื่องควบคุมแบบ PID จะมีสามพารามิเตอร์ที่จะต้องถูกปรับแต่ง ได้แก่ อัตราขยายหรือ Proportional Band เวลารีเซ็ตหรืออัตรารีเซ็ต อัตราต่อเวลา อัตราต่อเวลาหรือเดอริเวทีฟนี้ค่าเวลาจะเป็นหน่วยนาที เครื่องควบคุมแบบ PID จะถูกใช้กับกระบวนการที่มีเวลาคงที่ยาวนานและโดยไม่มีสัญญาณรบกวน ข้อดีของเดอริเวทีฟก็คือจะมีความสามารถในการคาดการณ์ข้างหน้าขณะที่กระบวนการหรือตัวแปรที่ถูกควบคุมกำลังมุ่งหน้าไป

**เครื่องควบคุมแบบสัดส่วนและเดอริเวทีฟ Proportional-Derivative Controller (PD)** เครื่องควบคุมแบบนี้จะถูกใช้ในกระบวนการที่ใช้เครื่องควบคุมแบบสัดส่วนอย่างเดียวแต่ต้องการค่าของการคาดการณ์ล่วงหน้าอีกน้อย แสดงสมการได้ดังนี้

$$m(t) = \bar{m} + Kc.e(t) + Kc.\tau_D \frac{de(t)}{dt} \quad (1.26)$$

และฟังก์ชันถ่ายโอนในอุดมคติคือ

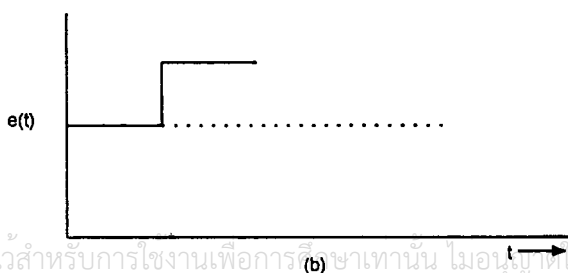
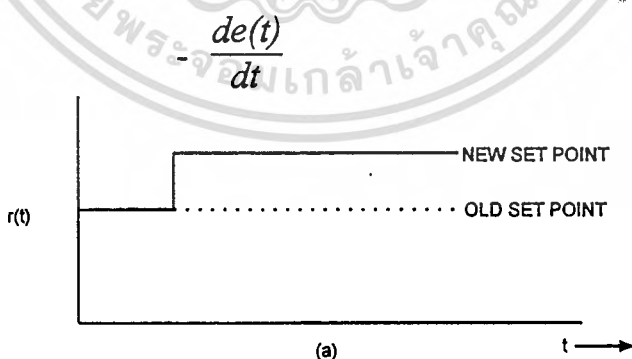
$$\frac{M(s)}{E(s)} = Kc(1 + \tau_D s) \quad (1.27)$$

ข้อเสียของเครื่องควบคุมแบบ PD ไม่สามารถที่จะแก้ค่าผิดพลาดที่สภาวะคงที่ได้ จะเฉพาะกรียาอินทริกัลเท่านั้นที่สามารถจะขจัดออฟเซ็ทให้หมดไปได้ อย่างไรก็ตามเครื่องควบคุมแบบ PD สามารถที่จะมีอัตราขยายที่สูงขึ้นกว่าแบบสัดส่วนอย่างเดียวจึงเป็นผลให้ออฟเซ็ทมีค่าน้อยไปได้ในวงรอบการควบคุมแบบง่าย

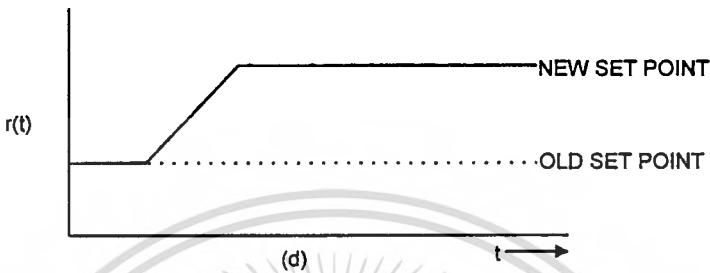
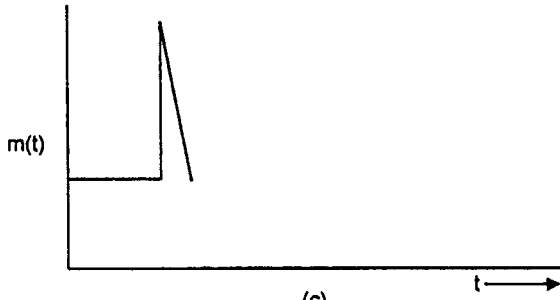
**เครื่องควบคุมแบบดิจิทัล Digital Controller** จากที่ได้กล่าวมาแล้วในสมการ (1.25) เป็นฟังก์ชันถ่ายโอนของเครื่องควบคุมในอุตสาหกรรมแบบอนาลอก อย่างไรก็ตามสมการของเครื่องควบคุมแบบดิจิทัลที่เป็นแบบที่ไม่ต่อเนื่องในสมการ (1.19) วิธีการในการปรับแต่งจะไม่ได้แตกต่างจากการปรับแต่งเครื่องควบคุมแบบอนาลอก

มีเรื่องราวของเครื่องควบคุมอีกเล็กน้อย ที่จะได้อธิบายต่อไปก่อนที่จะจบในส่วนนี้ได้แก่สมการ (1.19) แสดงให้เห็นว่าทุกเวลาที่มีการเปลี่ยนแปลงค่าอัตราขยายแล้วก็จะจะมีผลกระทบต่อกริยาอินทรีย์ลและเดอริเวทีฟเพราะ  $\tau_i, \tau_d$  ถูกคูณและหารโดยค่าอัตราขยายหมายความว่าถ้าต้องเปลี่ยนแปลงเฉพาะอัตราขยายอย่างเดียวโดยไม่ต้องให้มีผลกับอินทรีย์ลและเดอริเวทีฟแล้ว ก็จะต้องเปลี่ยนแปลงค่าทั้งอินทรีย์ลและเดอริเวทีฟทุกค่าของอัตราขยายที่ถูกเปลี่ยนไปซึ่งเครื่องควบคุมแบบอนาลอกจะเป็นแบบนี้ทั้งหมด บางครั้งเครื่องควบคุมดังกล่าวนี้ถูกเรียกว่า **Interacting Controller** เครื่องควบคุมที่ใช้ไมโครโปรเซสเซอร์โดยทั่วไปจะเป็นแบบนี้ด้วยอย่างไรก็ตามจะมีเครื่องควบคุมบางแบบที่แก้ปัญหานี้โดยการแทน  $Kc/\tau_i$  ด้วยเทอม  $Ki$  และ  $Kc \cdot \tau_d$  ด้วยเทอม  $Kd$  และพารามิเตอร์ทั้งสามคือ  $Kc, Ki, Kd$

สุดท้ายเป็นความสัมพันธ์ของกริยาเดอริเวทีฟ ในทางหนึ่งที่จะเปลี่ยนแปลงค่าเป้าหมายของเครื่องควบคุมโดยการเปลี่ยนแปลงในรูปที่ 1-8a เมื่อเป็นแบบการเปลี่ยนแปลงแบบเป็นขั้นในค่าผิดพลาดแสดงในรูปที่ 1-8b ดังนั้นเครื่องควบคุมจะต้องให้ค่าเดอริเวทีฟต่อความผิดพลาดนี้ และกริยาเดอริเวทีฟนี้จะมีการเปลี่ยนแปลงในทันทีที่ทางออกของเครื่องควบคุมที่แสดงในรูปที่ 1-8c การเปลี่ยนแปลงที่ทางออกนี้ไม่เป็นที่ต้องการและอาจก่อให้เกิดความเสียหายต่อการทำงานของกระบวนการ เพื่อที่จะแก้ปัญหาอันนี้จะใช้เพียงค่าลบของอนุพันธ์ในตัวแปรที่ถูกควบคุมเป็นค่าเดอริเวทีฟต่อความผิดพลาดดังกล่าว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1-8 แสดงผลกระทบต่อค่าเป้าหมาย

ค่าอนุพันธ์ของทั้งสองจะเท่ากันเมื่อค่าเป้าหมายยังคงที่สามารถแสดงได้ดังต่อไปนี้

$$\frac{de(t)}{dt} = \frac{d[r(t) - c(t)]}{dt} = \frac{dr(t)}{dt} - \frac{dc(t)}{dt} \quad (1.28)$$

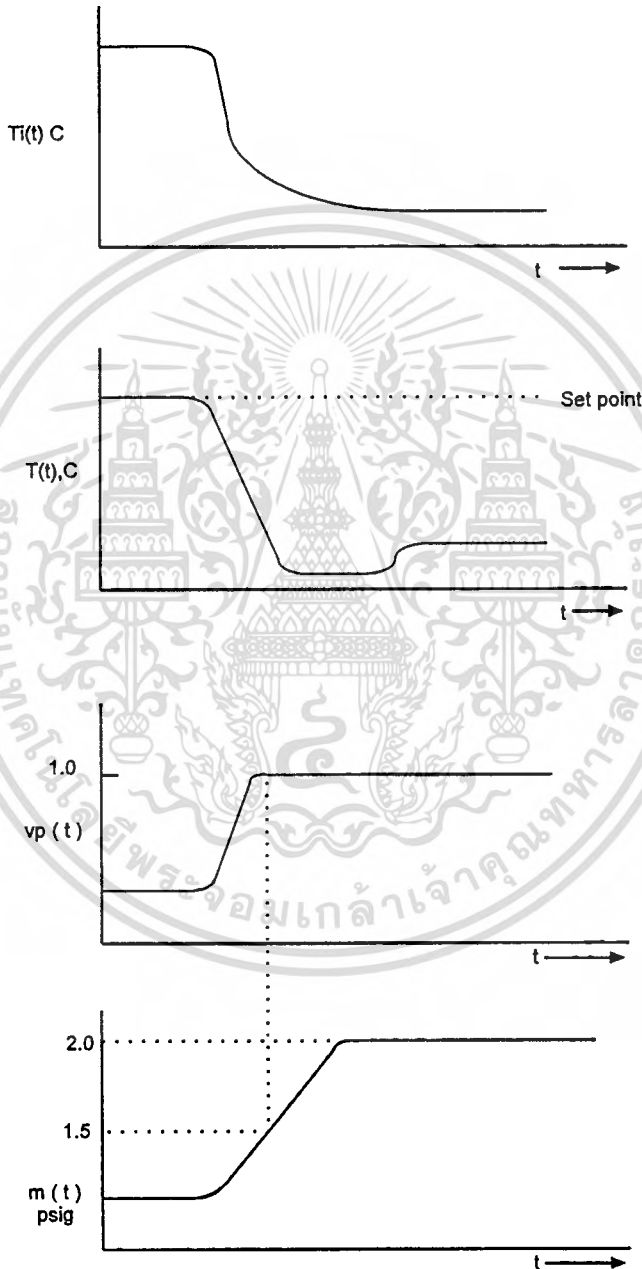
ขณะนี้ค่าเป้าหมายได้เปลี่ยนเป็นค่าใหม่แล้วค่าเดอริเวทีฟไม่ได้สร้างกรียาทันทีได้ออกมา ในขณะที่เดียวกันพฤติกรรมต่างๆ ก็ยังคงเหมือนแต่ก่อน การเลือกที่จะใช้เครื่องควบคุมแบบอนาลอกหรือแบบไมโครโปรเซสเซอร์จะต้องคำนึงถึงกรียาเดอริเวทีฟที่จะเป็นผลต่อตัวแปรที่ถูกควบคุมด้วย

### รีเซ็ต-วินด์อัฟ

ปัญหา รีเซ็ต-อัฟนี้เป็นปัญหาที่สำคัญและเกิดขึ้นจริงในกระบวนการ โดยเกิดได้ทุกเวลาที่เครื่องควบคุมมีการใช้โหมดควบคุมแบบอินทิกรัลรวมอยู่ด้วย วงรอบการควบคุมของกระบวนการแลกเปลี่ยนความร้อนที่แสดงในรูปที่ 5-8 ถูกใช้เพื่อแสดงให้เห็นปัญหาอันนี้

สมมติว่าอุณหภูมิของเหลวที่ทางเข้าของกระบวนการลดลงไปจำนวนหนึ่ง สิ่งรบกวนอันนี้จะกระทบไปจนถึงอุณหภูมิที่ทางออกของกระบวนการด้วย เครื่องควบคุม PI และ PID ก็จะทำกรเปิดวาล์วไอน้ำให้กว้างขึ้น โดยเพิ่มสัญญาณแรงดันลมให้กับวาล์วที่เป็นแบบใช้แรงดันเพื่อเปิด จนอุณหภูมิทางออกเพิ่มขึ้นจนเท่ากับค่าเป้าหมายที่ต้องการ แต่ถ้าสมมติว่าถ้าตัวแปรที่ถูกควบคุมยังไม่เพิ่มขึ้นเท่าใด เครื่องควบคุมก็จะต้องอินทิเกรตค่าผิดพลาดไปจนที่ทางออกเป็น 15 psig ที่จุดนี้วาล์วไอน้ำก็จะเปิดกว้างสุดเพราะฉะนั้นวงรอบการควบคุมไม่สามารถที่จะทำอะไรได้อีกแล้ว กระบวนการอยู่นอกเหนือจากการควบคุมดังแสดงเป็นกราฟในรูปที่ 1-9 ในรูปนี้แสดงให้เห็นว่าเมื่อวาล์วถูกเปิดสุด ตัวแปรที่ถูกควบคุมหรืออุณหภูมิทางออกก็ยังคงไม่เข้าหาค่าเป้าหมาย ก็ยังคงเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีค่าผิดพลาดอยู่ตลอด เครื่องควบคุมก็พยายามที่จะแก้ (โดยการอินทิเกรตค่าผิดพลาด) ออกที่ทางออกมากขึ้นแต่ว่าวาล์วไม่สามารถที่จะเปิดกว้างกว่านี้ได้แล้ว และที่ทางเครื่องควบคุมจะอินทิเกรตมากขึ้นจนไปถึงที่แรงดันแหล่งจ่ายซึ่งโดยปกติจะประมาณ 20 psig ที่จุดนี้เองเครื่องควบคุมไม่สามารถที่จะเพิ่มค่าที่ทางออกได้มากกว่านี้ทางออกที่จุดอิมิตัวนี้ได้แสดงไว้ในรูปที่ 1-9 ด้วย จุดอิมิตัวจะเป็นผลมาจากกรียาอินทริกัลของเครื่องควบคุมที่ผิดพลาดได้เกิดขึ้นเป็นเวลานาน และสภาวะอิมิตัวนี้จึงถูกเรียกว่า รีเซ็ท - วินด์อัฟ



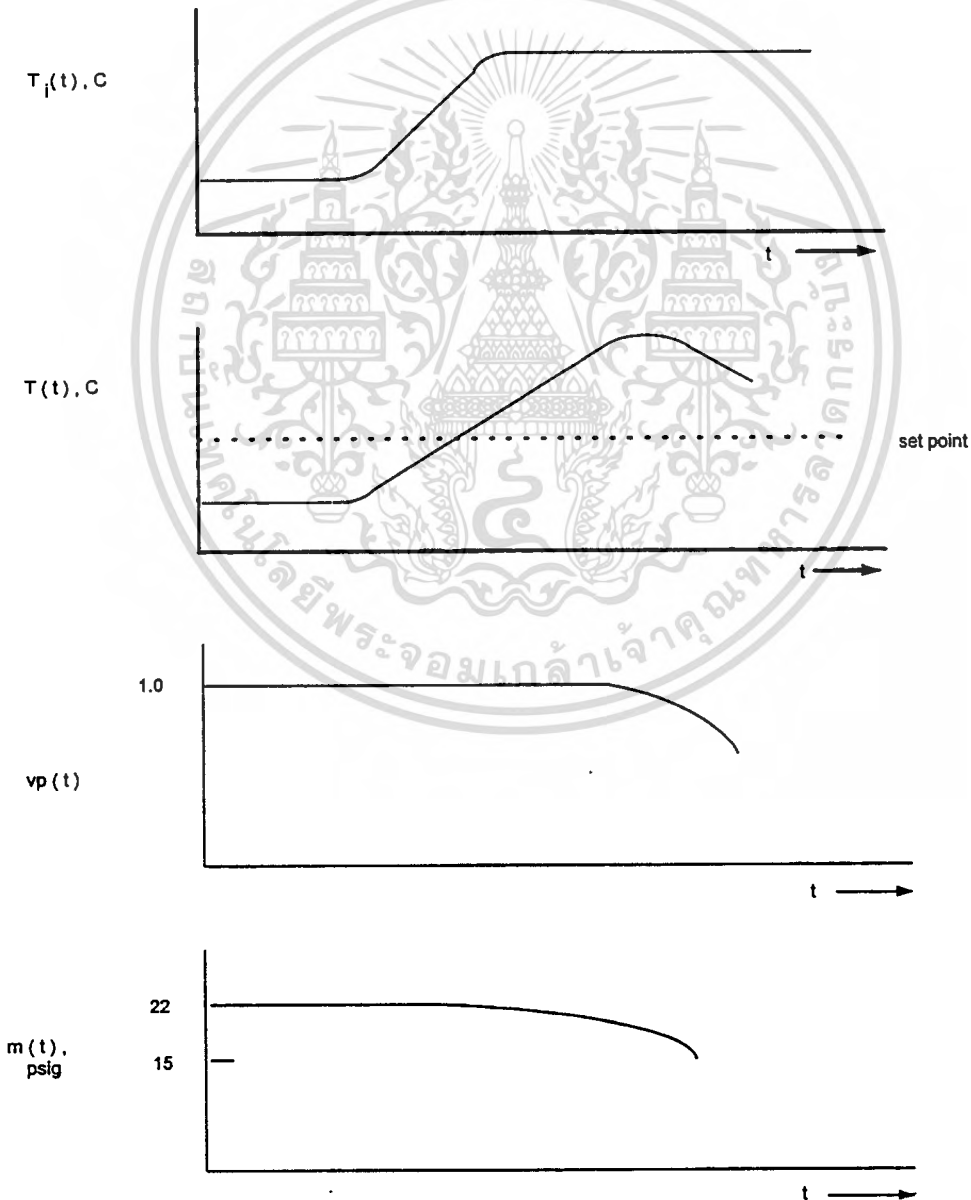
รูปที่ 1-9 แสดงผลตอบสนองในวงรอบการควบคุมของกระบวนการแลกเปลี่ยนความร้อน และในขณะที่สมมติว่าอุณหภูมิทางเข้าได้กลับมาตามเดิมอุณหภูมิทางออกของกระบวนการก็เริ่มต้นที่จะเพิ่มขึ้นแสดงในรูปที่ 1-10 รูปนี้แสดงอุณหภูมิทางออกเพิ่มขึ้นเข้าสู่ค่าเป้าหมายและผ่านเลยไป

แต่ว่าวาล์วก็ยังคงเปิดกว้างสุดอยู่ ซึ่งในความเป็นจริงแล้วน่าจะเปิดไปได้แล้ว เหตุผลที่มันไม่ปิดก็

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการเรียนการสอนเท่านั้น มิอนุญาตให้เผยแพร่โดยไม่ขออนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะว่าเครื่องควบคุมจะต้องอินทรีเกรดค่าที่ค้างอยู่เดิมออกเสียก่อนจาก 20 psig ไปเป็น 15 psig จากนั้นถึงจะเริ่มถึงที่จะปิดวาล์วได้ ในเวลานี้เองทำให้อุณหภูมิทางออกเกิดการพุ่งเลขขอบเขตของค่าเป้าหมายไปจำนวนหนึ่ง การแก้ไขในกรณีรีเซ็ต-วินด์อัฟ โดยการเปลี่ยนโหมดการควบคุมไปที่ควบคุมด้วยมือเครื่องควบคุมก็จะหยุดการอินทรีเกรดจากนั้นก็เปลี่ยน โหมดการควบคุมกลับไปใช้อัตโนมัตินี้เพื่อเริ่มที่จะลดค่าที่ทางออกต่อไป ข้อเสียของการกระทำแบบนี้ก็คือไม่มีการเตือนว่ากรณีนี้จะเกิดขึ้นเมื่อใด อย่างไรก็ตามเครื่องควบคุมส่วนมากจะมีอุปกรณ์ป้องกันในกรณีเกิดรีเซ็ต-วินด์อัฟ ไว้ด้วย การป้องกันโดยอัตโนมัติจะหยุดการอินทรีเกรดของเครื่องควบคุมถ้าค่าที่ทางออกเข้าสู่ 15 psig (20mA) หรือ 3 psig (4mA) รีเซ็ต-วินด์อัฟ มักจะเกิดกับกระบวนการแบบแบทช์ (batch process) หรือในการควบคุมแบบแคสเคด (cascade control) และเมื่ออุปกรณ์ควบคุมขั้นสุดท้ายถูกควบคุมจากเครื่องควบคุมที่มากกว่าหนึ่ง



รูปที่ 1-10 แสดงผลตอบสนองในวงรอบการควบคุมของกระบวนการแลกเปลี่ยนความร้อน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้เผยแพร่โดยไม่ระบุชื่อหน่วยงานการศึกษานี้  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 2

### คุณสมบัติของ CP - Z84C11 PLUS

บอร์ด CP - Z84C11 PLUS พัฒนาคือจากบอร์ด CP - Z84C11 เพิ่มในส่วนของวงจร RTC เข้าไปอีกเป็นบอร์ดที่ใช้ CPU Z84C11 ของบริษัท ZILOG มาเป็น CPU ประจำบอร์ด RUN ที่ SPEED 10 MHz CPU Z84C11 นี้ก็คือ CPU ที่รวมเอาชิพวงจรต่างๆ ของ ZILOG เข้าด้วยกันคือ

- Z84 C00 เป็น CPU Z80 แบบ CMOS RUN ที่ 10 MHz
- Z84 C30 เป็น Z80 CTC แบบ CMOS RUN ที่ 10 MHz
- CGC เป็น CLOCK GENERATOR CONTROL CIRCUIT
- WDT เป็นวงจร WATCH DOG TIMER
- POWER ON RESET เป็นวงจร RESET CPU เมื่อ VOLT VCC ต่ำกว่า 2.2 V
- 40 BIT PARALLEL PORT เป็น PORT 8 BIT จำนวน 5 PORT ใช้งาน

จะเห็นว่าบอร์ด CP - Z84 C11 PLUS ก็คือบอร์ดที่มี CPU เป็น Z80 พร้อมด้วยวงจรต่างๆ เพิ่มเข้ามาเหมาะสำหรับผู้ที่ใช้งาน Z80 ที่ต้องการขีดความสามารถสูงๆ ขึ้นอีก ประมวลผลได้รวดเร็วขึ้นพร้อมทั้งบอร์ด Z84 C11 นี้ยังสามารถต่อกับอุปกรณ์ต่างๆ ทั่วไปได้อีก เช่น LCD, PRINTER, KEYBOARD, EEPROM หรือต่อลำโพงด้วยก็ได้เหมาะเป็นบอร์ดใช้ในการ CONTROL เป็นอย่างยิ่ง

### ข้อมูลของบอร์ด CP - Z84C11 PLUS

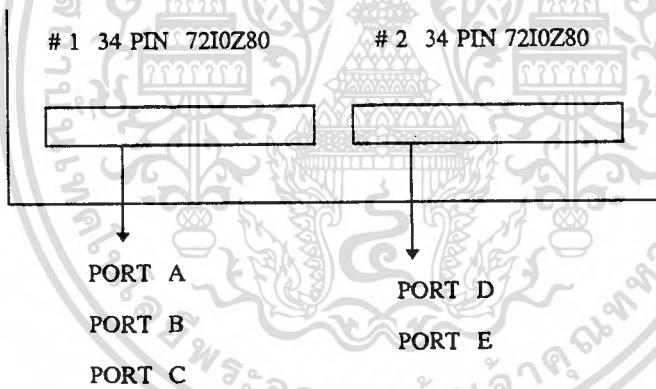
#### CPU

Z84C11 ของบริษัท ZILOG โดยเป็น HIGH SPEED OPERATION โดยมีให้เลือก 2 รุ่น คือ RUN ที่ความเร็ว 6 Mhz และ RUN ที่ความเร็ว 10 MHz CPU เบอร์นี้ของ ZILOG จะเป็นแบบ CMOS ทำให้กินกระแสต่ำมากในขณะที่เรา SET ให้อยู่ใน STOP MODE จะกินกระแสเพียง 50  $\mu$ A ข้อดีอีกอย่างหนึ่งของ CPU ตัวนี้ก็คือ ถึงแม้จะ RUN 10 MHz ตามเราก็สามารถ SET ให้อ่านเพียงครั้งเดียวก็ได้เพียง 5 MHz เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ใช้ RAM หรือ ROM ที่มี ACCESS TIME ต่ำๆ โดยเมื่อ POWER ON แล้ว CPU จะ RUN เพียง 5 MHz เท่านั้นตอนเริ่มต้นและเมื่อเราต้องการ RUN 10 MHz ก็สามารถใช้โปรแกรม รูปลักษณะของ CPU จะเป็น PACKAGE แบบ 100 PIN CP - Z84C11 สามารถใส่หน่วยความจำได้สูงสุด 64 KBYTE โดยแบ่งเป็น

- EPROM สามารถใส่ EPROM ขนาด 32 KBYTE ได้โดยเป็นเบอร์ 27256 มีหน่วยความจำเริ่มจาก ADDRESS 0000H ถึง ADDRESS 7FFFH
- RAM สามารถใส่ RAM ขนาด 8 - 32 KBYTE ได้โดยเป็นเบอร์ 6264 หรือ 62256 มีหน่วยความจำเริ่มจาก ADDRESS 8000H ถึง ADDRESS FFFFH

CP - Z84C11 จะมี PORT ใช้งาน 40 BIT I/O หรือ 5 PORT ด้วยกัน โดย 5 PORT นี้จะเป็น PORT ในตัว CPU โดยจะมีตำแหน่งต่อออกมาดังรูป



PORT ที่ต่อออกมานี้จะอยู่เป็น CONNECTER 34 PIN ( 72 IO Z80) สามารถต่อร่วมใช้กับ อุปกรณ์สนับสนุนต่าง ๆ ได้มากมาย

CHANNEL PORT	ADDRESS
PORT A DATA PORT	50 H
PORT B DATA PORT	51 H
PORT C DATA PORT	52 H
PORT D DATA PORT	30 H
PORT E DATA PORT	40 H

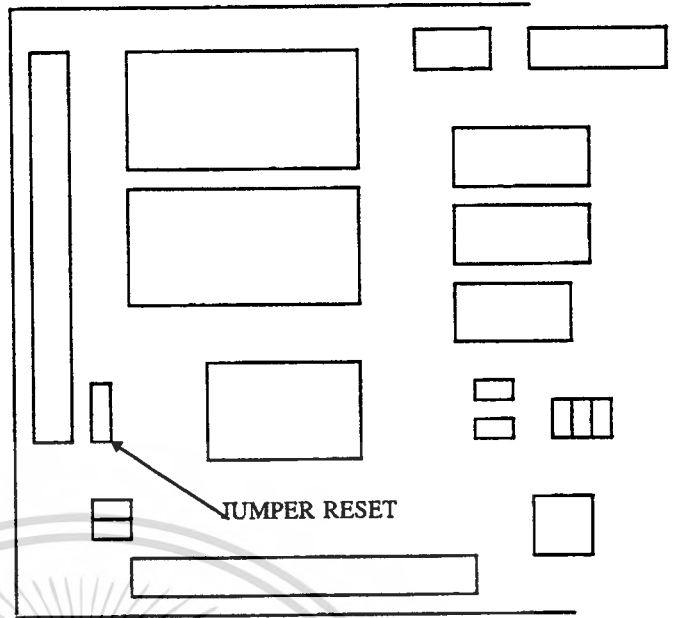
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ 40 H ขนด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHANNEL CONTROL PORT	ADDRESS
PORT A DATA DIRECTION REGISTER	54 H
PORT B DATA DIRECTION REGISTER	55 H
PORT C DATA DIRECTION REGISTER	56 H
PORT D DATA DIRECTION REGISTER	34 H
PORT E DATA DIRECTION REGISTER	44 H

เราสามารถสั่งให้ PORT ของ Z84C11 เป็น INPUT หรือ OUTPUT ได้โดยอิสระ BIT ต่อ BIT โดยถ้าเราสั่ง SET ค่าออกที่ PORT DATA DIRECTION REGISTER ถ้าให้ BIT ใดเป็น 1 ก็คือให้ PORT DATA ของ BIT นั้น ๆ เป็น OUTPUT PORT ถ้า SET PORT นั้นเป็น OUTPUT PORT ค่าเริ่มต้นจะเป็น 0 และถ้าเราต้องการให้เป็น INPUT PORT ก็ SET ค่าออกที่ PORT DATA DIRECTION REGISTER เป็นค่า 0

## Z80BUS

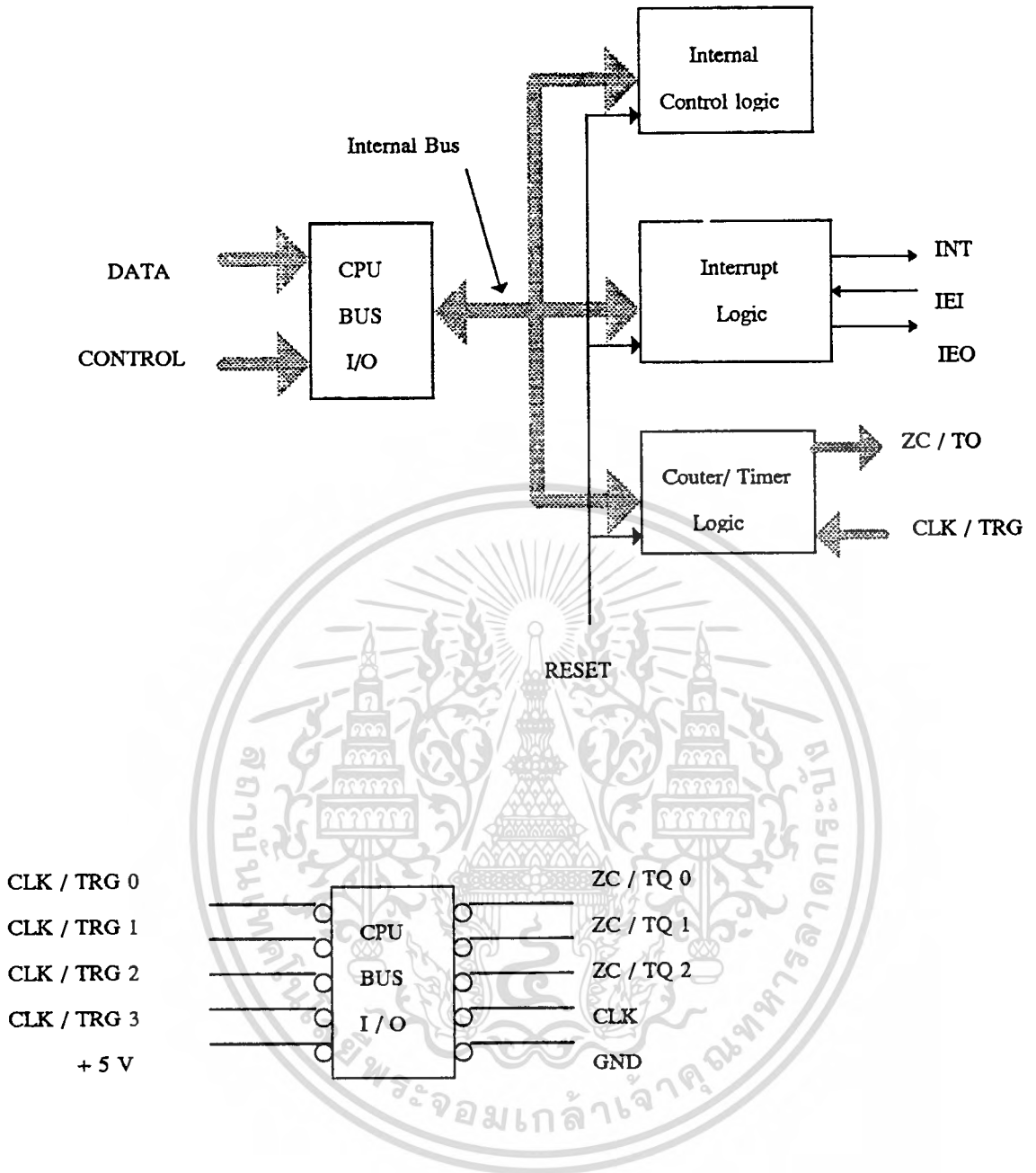
PC - Z84C11 จะมีส่วนขยายระบบได้ทาง Z80 BUS โดยมีลักษณะขาคู่เหมือนกับ CPU Z80 40 PIN ทำให้เราสามารถต่อขยาย PORT 8255 ได้อีกโดยใช้ บอร์ด 72 IO Z80. เป็นต้น มีข้อพิเศษคือเนื่องจาก CPU Z84C11 นี้มีส่วนวงจร POWER ON, RESET ในตัวอยู่แล้วไม่จำเป็นต้องใช้วงจร POWER ON RESET ต่อจากภายนอก เช่น CPU เบอร์อื่น ๆ ฉะนั้นถ้าเราไปต่อร่วมกับ Z80 BUS อื่น ๆ ที่มี R, C ต่ออยู่แล้วอาจทำให้ CPU Z84C11 นี้อยู่ในสภาพ RESET ตลอดเวลาได้ บอร์ด CP - Z84C11 จึงมี JUMPER ในการเลือกไม่ให้ต่อ RESET จากภายนอก โดยถ้ามีการต่อ R,C ที่ RESET PIN ภายนอกก็ให้ถอด JUMPER RESET ออก



### CTC

CP - Z84C11 จะมีวงจร Z84C30 (Z80 CTC) ต่อรวมกันอยู่ในตัวเรียบร้อยแล้วโดย บอร์ด CP - Z84C11 จะต่อขาใช้งาน CTC ออกมาที่ CONNECTER 10 PIN แล้ว ค้างรูป ส่วนขา INT ของ CTC นั้นจะต่อกับขา INT ของ Z80 ในตัวโดยเป็นแบบ WIRED - OR เรียบร้อยในตัวแล้วรายละเอียดของ CTC ในบอร์ดนี้จะเหมือนกับ Z 80 CTC ทั้งหมด

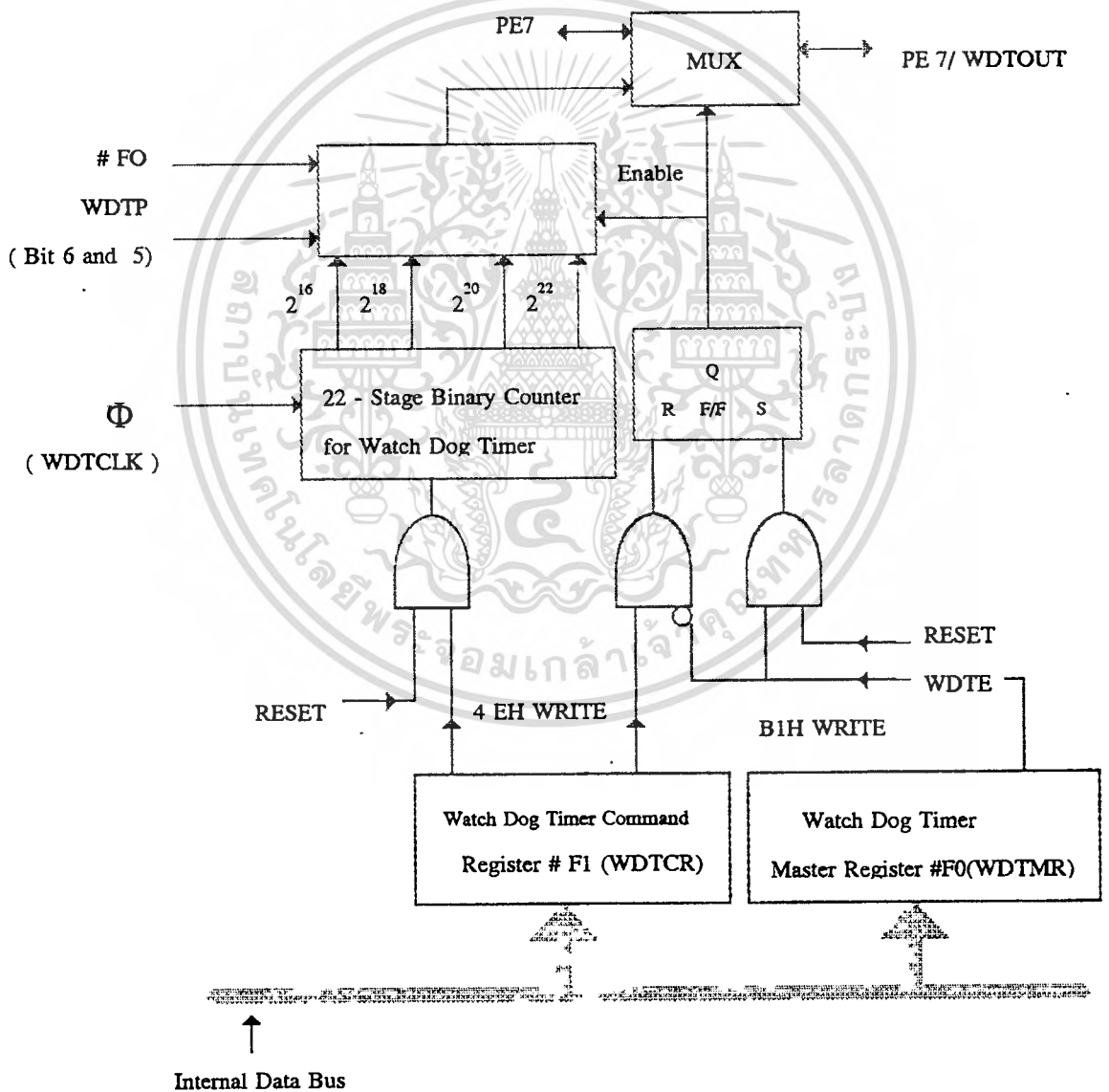
CHANNEL	CTC ADDRESS PORT
CH 0	10 H
CH 1	11 H
CH 2	12 H
CH 3	13 H



## WDTOUT

WATCH DOG เป็นลักษณะวงจรที่จะทำการ RESET CPU อยู่เสมอตามค่าเวลาที่เรากำหนด ซึ่งถ้าเราไม่ทำการ DISABLE WATCH DOG ภายในเวลากำหนด CPU นั้นก็จะถูก RESET เช่นในโปรแกรมทำงานปกติเราจะ CALL DISABLE WATCH DOG อยู่เสมอ แต่ถ้า CPU กำลัง RUN อยู่ นั้นจะเกิดมีสัญญาณรบกวนขึ้น ทำให้ไม่อาจสามารถ RUN โปรแกรมปกติที่มีการเรียกใช้โปรแกรม CALL DISABLE WATCH DOG ได้ CPU ก็จะเกิดการ RESET ขึ้นทันทีเพื่อให้กลับไปเริ่ม RUN ใหม่อีก

เป็น PIN จาก PORT PE BIT 7 ซึ่งทำหน้าที่ 2 อย่างคือ เป็น DATA PORT  
 ทั่ว ๆ ไปแล้ว ก็ยังเป็น WATCH DOG TIMER OUTPUT ด้วยโดยจะเป็น OPEN -  
 DRAIN I/O เมื่อถูก SET ให้เป็น WDTOUT โดย PIN นี้จะถูกต่อโดยตรงเข้ากับ  
 PIN RESET ของ CPU โดยปกติถ้าเราไม่มีการ SET WDTOUT แล้วให้ถอด JUMPER  
 ออกห้ามต่อถ้าไม่มีการใช้ WDTOUT เพราะเมื่อเราใช้ PORT PE 7 จะทำให้ CPU ถูก  
 RESET ตามคำสั่ง OUT PORT PE 7 นั้นด้วย



**Watch Dog Timer Block Diagram**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### บทที่ 3

#### รายละเอียดและหลักการของโปรแกรม

ในโครงการนี้ได้ใช้การเขียนโปรแกรมบน Z84C11 DEBUGER ซึ่งเป็นตัวช่วยในการเขียนโปรแกรม เนื่องจากมีคำสั่งช่วยอำนวยความสะดวกในการแก้ไขโปรแกรม และคำสั่งในการรัน

การพัฒนาโปรแกรมของเครื่องควบคุมนี้ เริ่มจาก

- การเขียนซอสโคช แอสแซมบลีบน Q-EDITER ซึ่งจะได้ไฟล์ข้อมูลนามสกุล .ASM แล้ว
- ทำการแปลงให้เป็นไฟล์ข้อมูลนามสกุล .HEX โดยใช้โปรแกรม (C16)
- ใช้โปรแกรม โปรคอม (PROCOM) โหลดไฟล์ข้อมูล .HEX ลงบน Z84C11 DEBUGER ณ.ตำแหน่ง (8000)

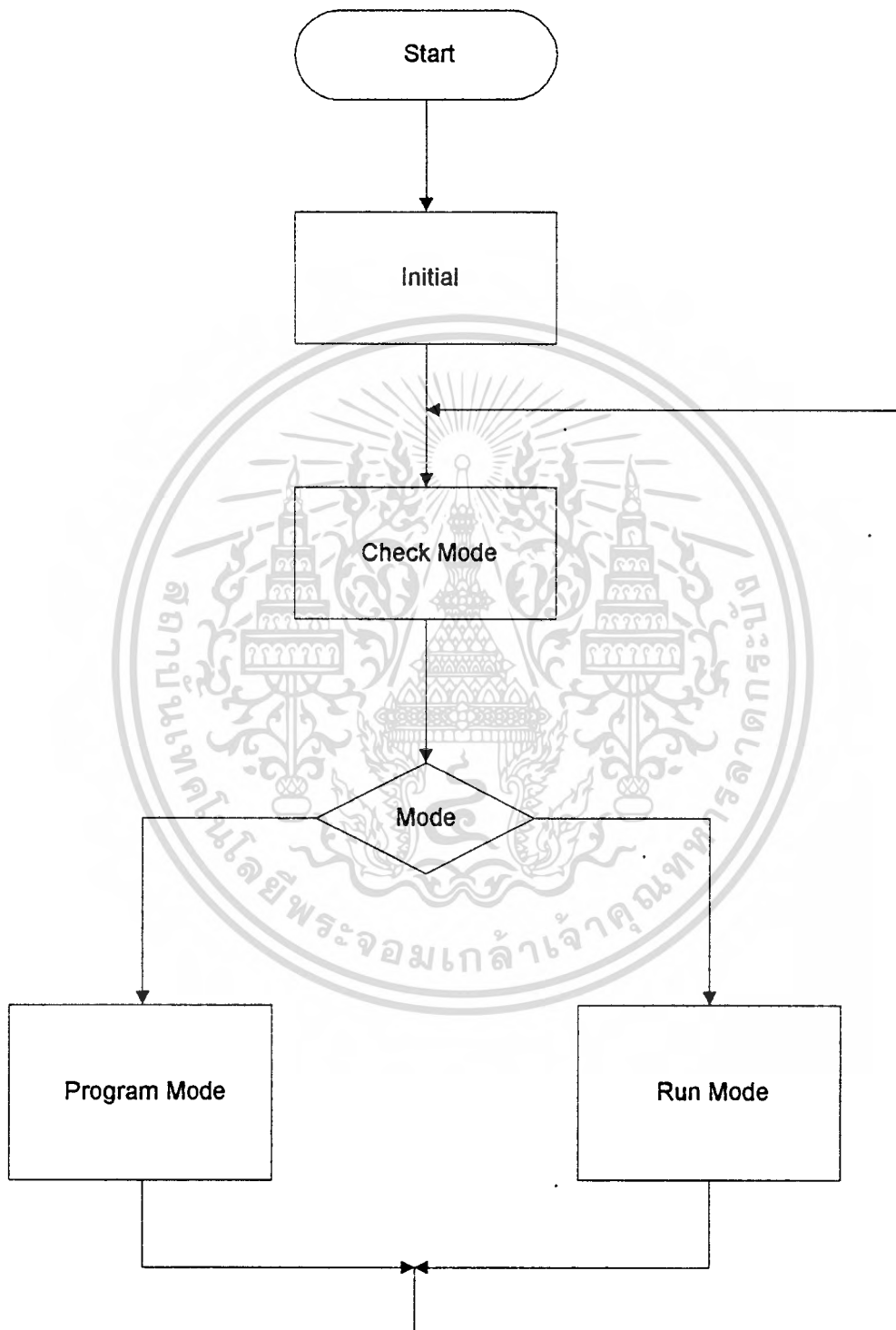
การจัดตำแหน่งของโปรแกรมที่เขียนบน Z84C11 DEBUGER

8000 - 0F5FF H DATA AREA
0F600 - 0AFC H PROGRAM EXECUTE AREA (255 LINE)
0FB00 - 0FEFC H PROGRAM EDITOR AREA (255 LINE)

การกำหนดตำแหน่งของพอร์ท

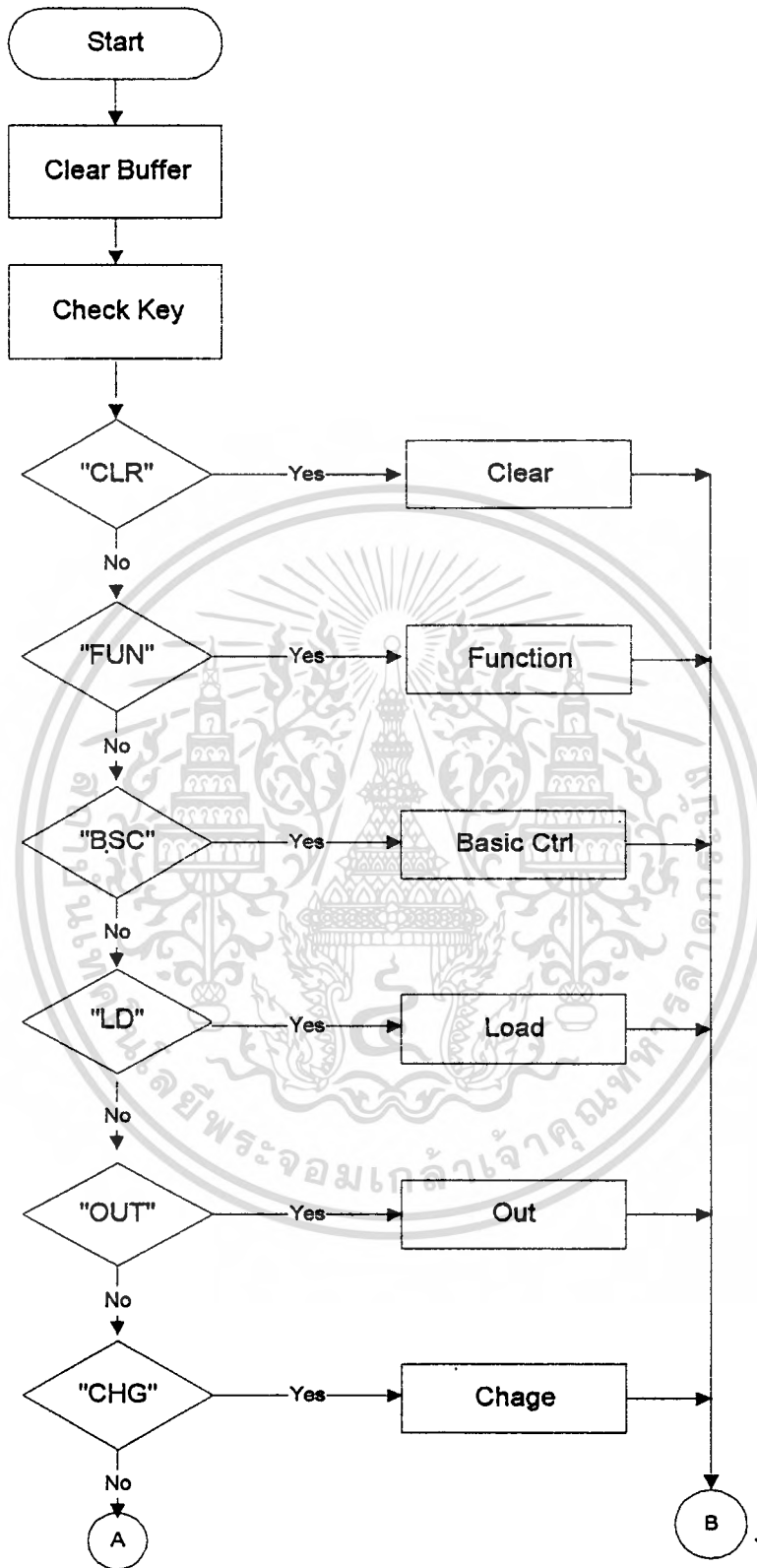
- พอร์ท B (51H) ใช้เลือกโหมดระหว่าง RUN และ EDITOR  
1 = EDITOR  
0 = RUN
- พอร์ททีย์ (BOH) ใช้รับข้อมูลที่ส่งมาจากคีย์บอร์ด

หลักการเขียนโปรแกรมควบคุมการทำงานของเครื่องควบคุมได้แสดงไว้ดังนี้



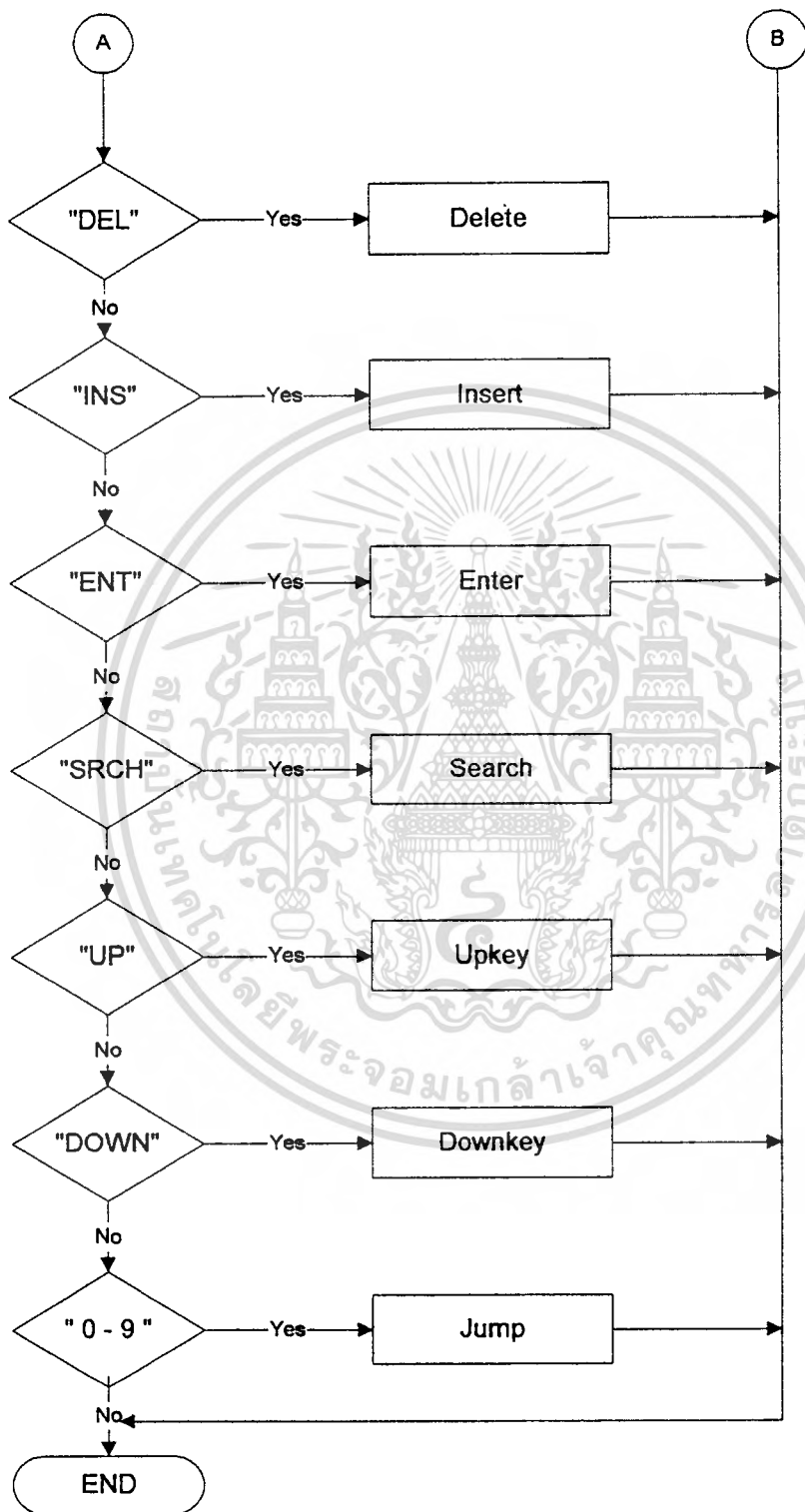
### แสดงการทำงานในการเลือกโหมดของระบบ PID - Programmable Controller

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



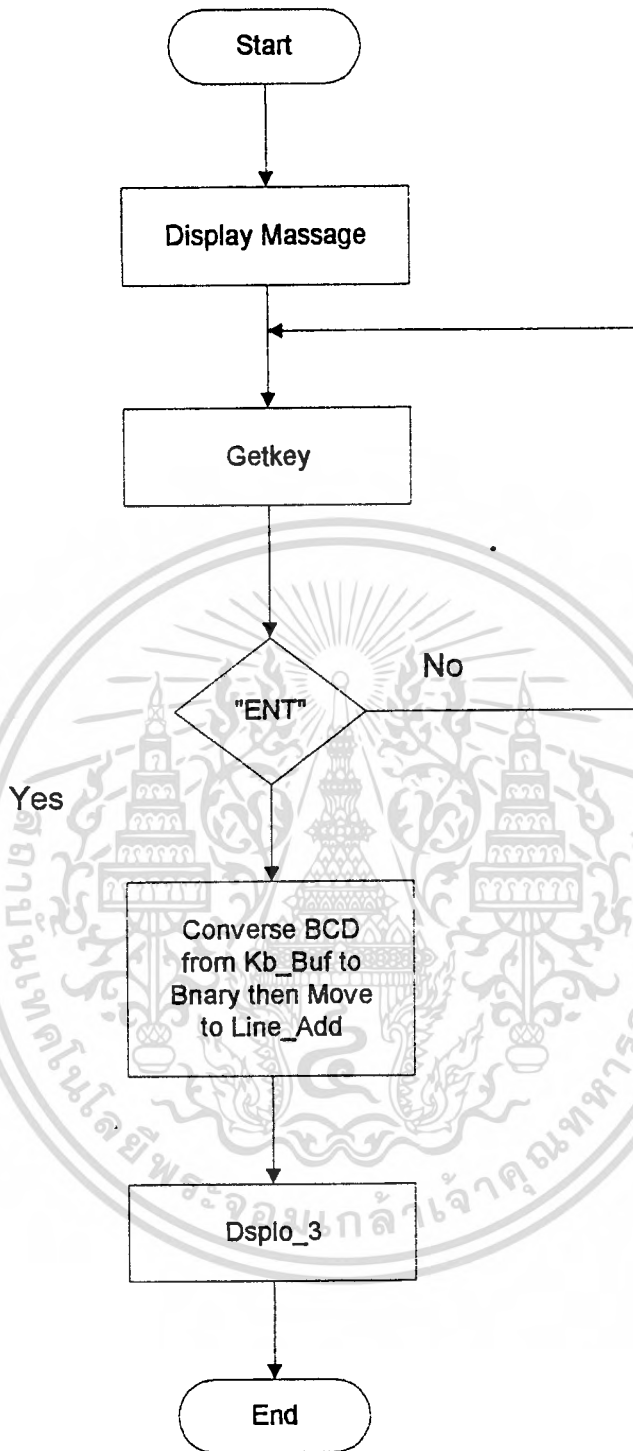
### แสดงการทำงานของ PID-Programmable Controller ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



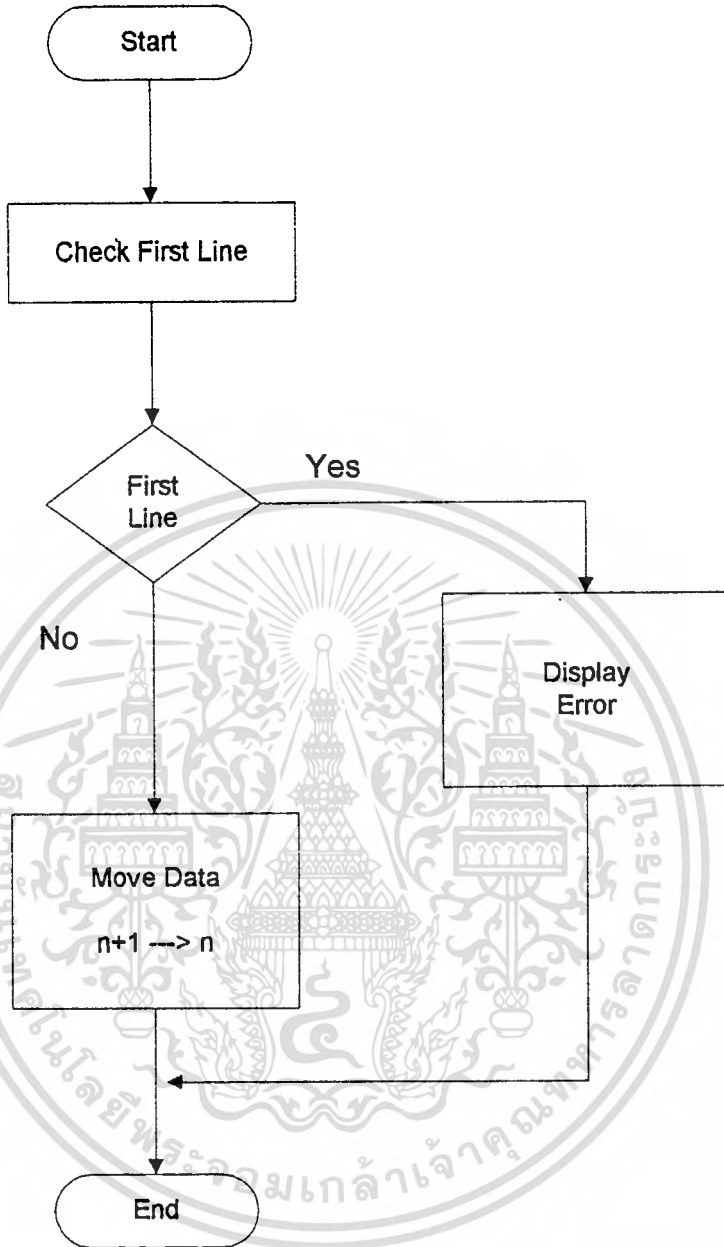
แสดงการทำงานของ PID-Programmable Controller ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



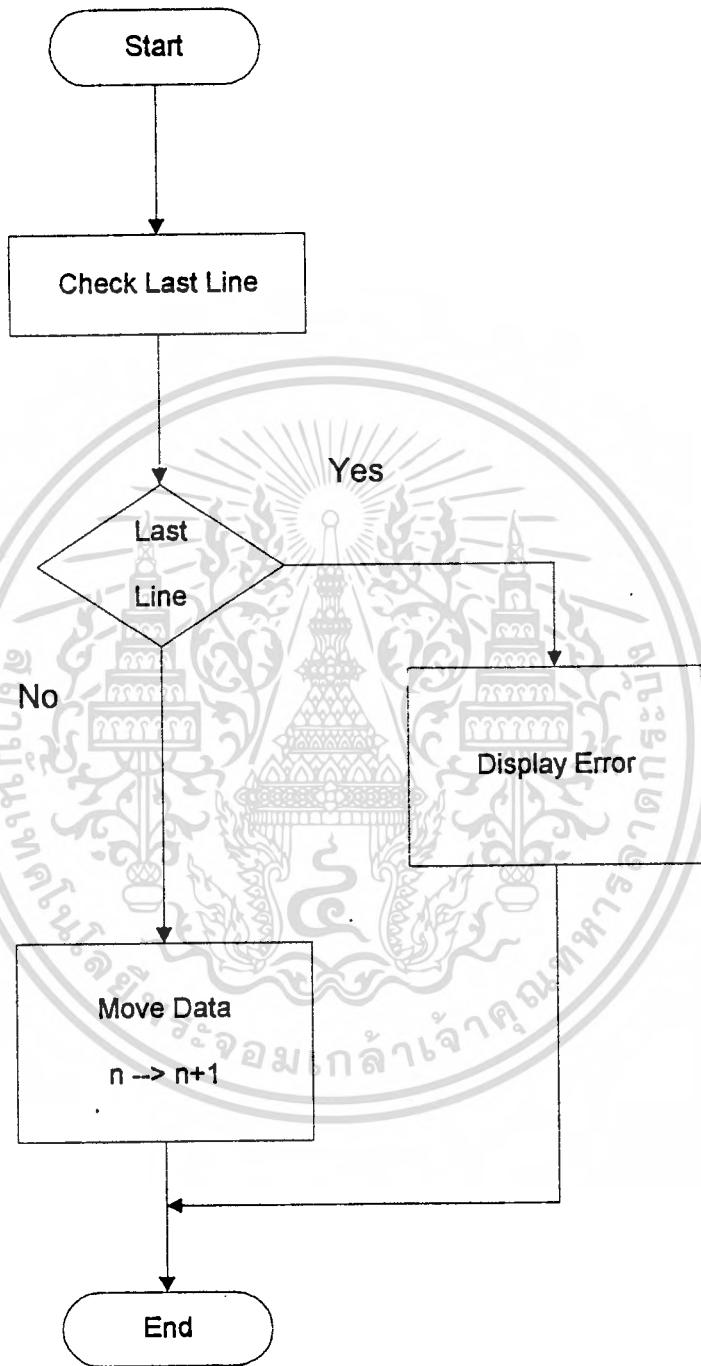
### แสดงการทำงานของ Jump ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



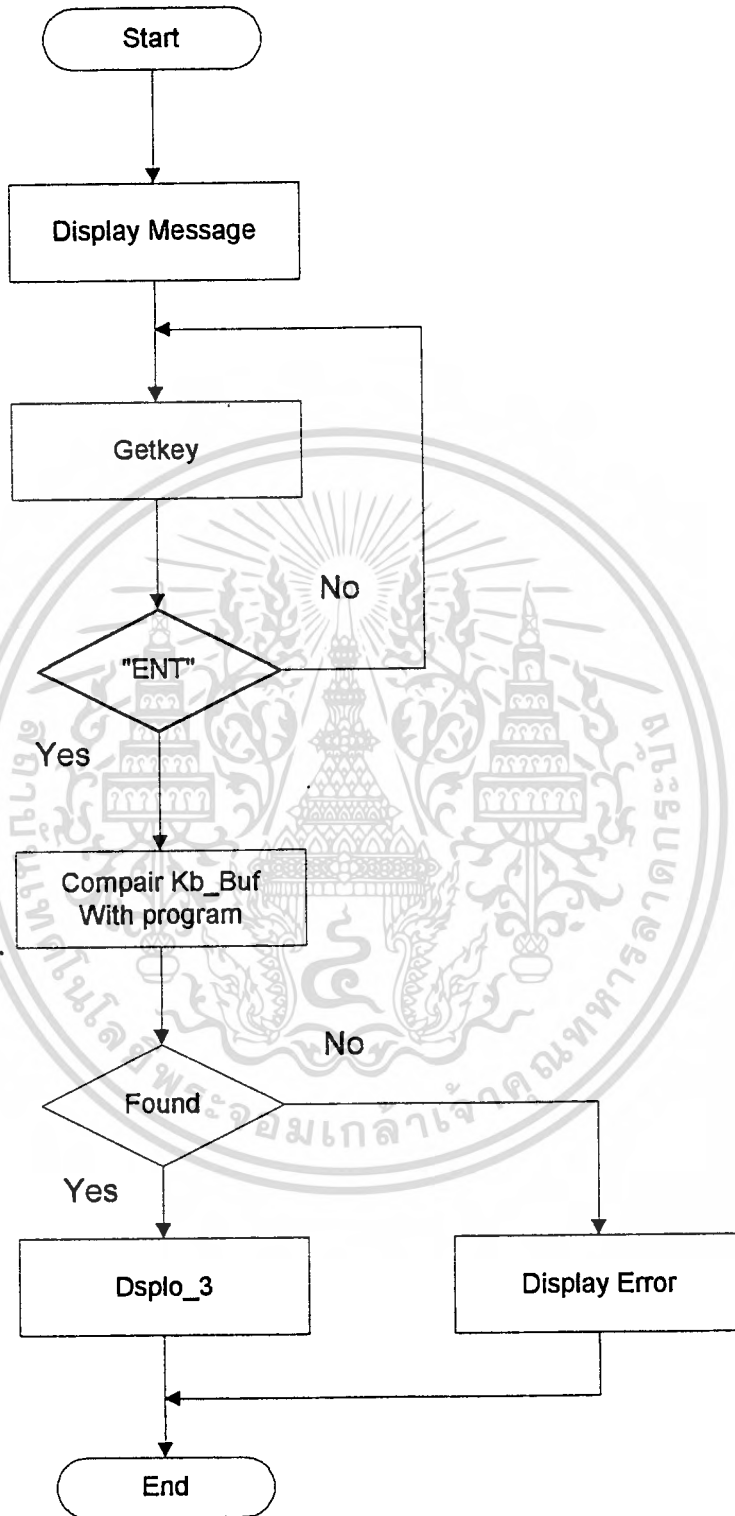
### แสดงการทำงานของ Delete ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



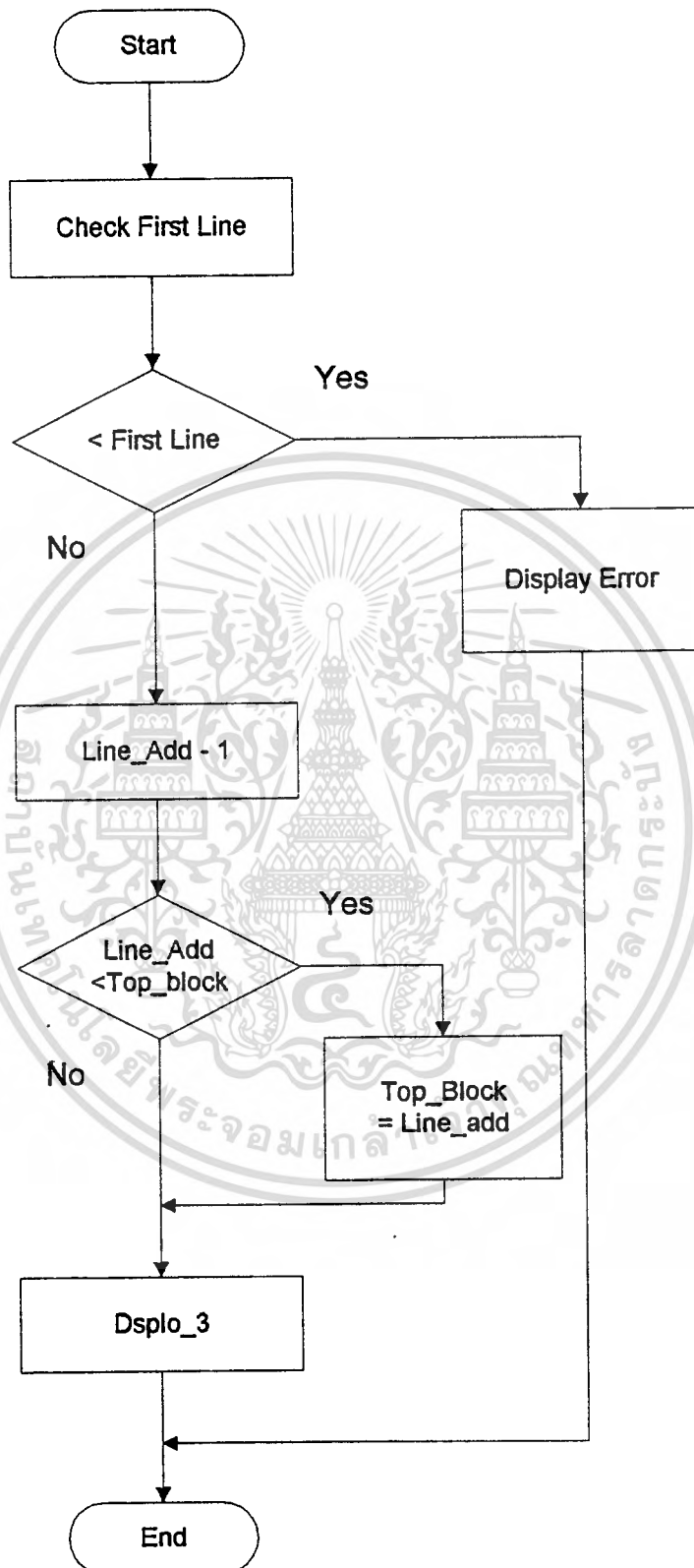
### แสดงการทำงานของ Insert ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



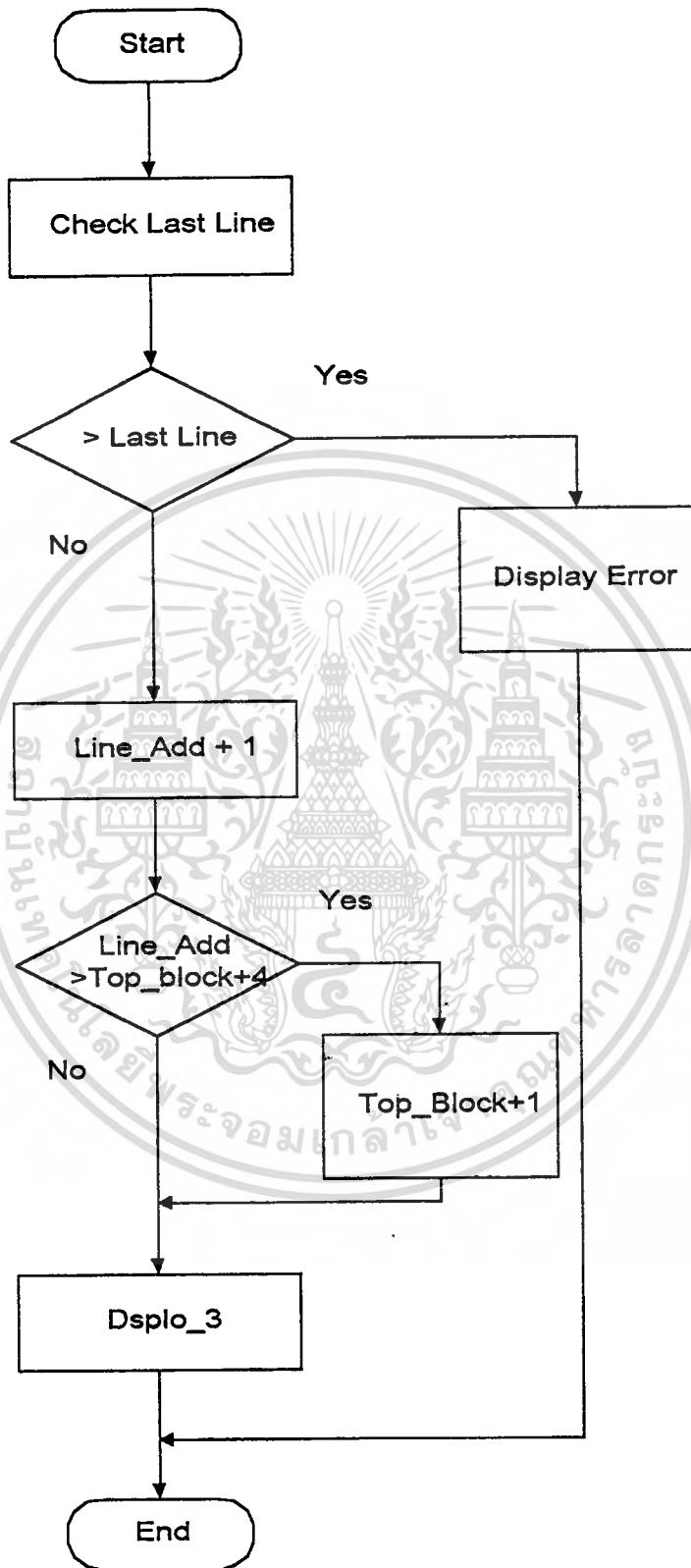
### แสดงการทำงานของ Search ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### แสดงการทำงานของ Up Key ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



### แสดงการทำงานของ Down Key ในโหมดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### หลักการงานและการแสดงผลของเครื่องควบคุม

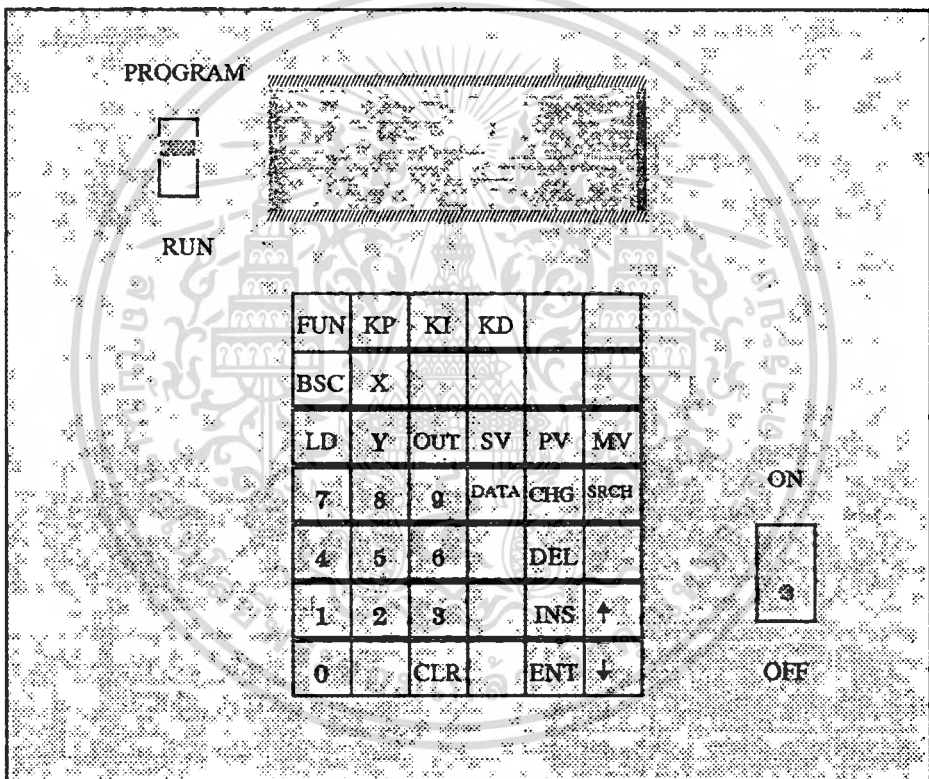
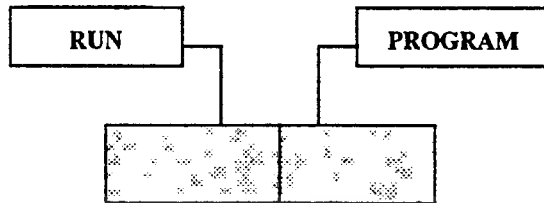
#### 4.1 คำสั่งพื้นฐานสำหรับการเขียนโปรแกรม (BASIC INSTRUCTION PROGRAMME) ของเครื่องควบคุม

สัญลักษณ์	ข้อมูลที่ใช้	ความหมาย
LD	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆที่กำหนดเข้ามา คำสั่งนี้จะใช้ในการเริ่มต้นของการเขียนโปรแกรม
OUT	Y,SV,PV,MV	เป็นการให้ค่าแก่อุปกรณ์ต่างๆที่กำหนด
BSC	X	เป็นการนำค่าอินพุตเข้ามาประมวลผลในรูปแบบ PID
SUM	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมาบวกกัน
SUB	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมาลบกัน โดยใช้ค่าสถานะบรรทัดแรกตั้งใช้ค่าสถานะบรรทัดต่อมาเป็นตัวลบ
MUL	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมากระทำการคูณ
DIV	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมาหารกัน โดยใช้ค่าสถานะบรรทัดแรกตั้งใช้ค่าสถานะบรรทัดต่อมาเป็นตัวหาร
SQRT	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมาหารากที่สอง
SQR	X,Y,SV,PV,MV	เป็นการนำค่าสถานะปัจจุบันตำแหน่งต่างๆมายกกำลังสอง
END		ใช้เมื่อสิ้นสุดการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

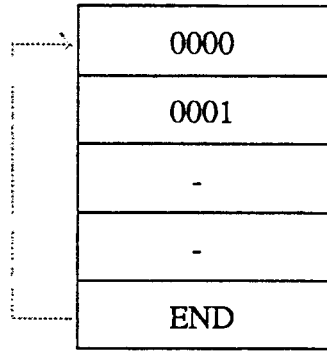
## 4.2 การใช้งานโปรแกรมคอนโซล (PROGRAM CONSOLE)

### 4.2.1 การเลือกโหมดการใช้งาน (MODE SELECTOR SWITCH)



**โหมดโปรแกรม :** เป็นโหมดการเขียนโปรแกรม เพื่อใช้ในการสร้าง แก้ไข ลบ แทรกข้อมูล และในขณะที่เดียวกันก็สามารถตรวจสอบค่าต่างๆของ SV,PV,MV,KP,KI,KD ได้

**โหมดรัน :** ใช้ในการรันโปรแกรม โดยโปรแกรมจะถูก Executed จากตำแหน่ง 0000 จนกระทั่งถึงคำสั่ง END

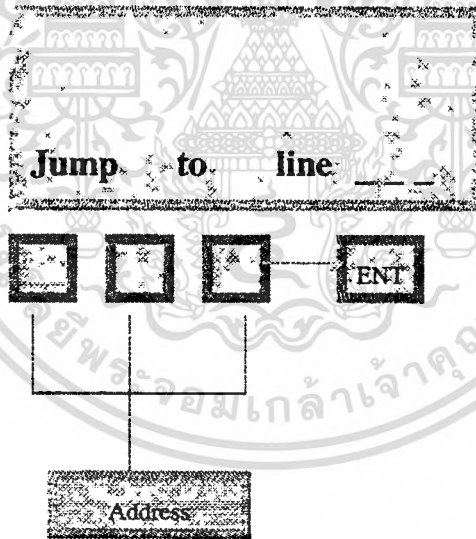


4.2.2 การป้อนโปรแกรม

ทุกครั้งที่มีการกดคีย์คำสั่ง ผู้ใช้จะต้องกด ENT เสมอเป็นการเก็บข้อมูลคำสั่งลงหน่วยความจำ

4.2.3 การตั้งค่าหมายเลขบรรทัด

เพื่อให้สามารถเลือกไปทำตำแหน่งใดๆของโปรแกรมได้(000-255) โดยการข้ามไปที่ ตำแหน่งนั้นๆทันที โดยกดคีย์ดังนี้ คีย์ตัวเลขใดๆ (0-9) เมื่อได้หมายเลขบรรทัดที่ต้องการแล้วกด ENT



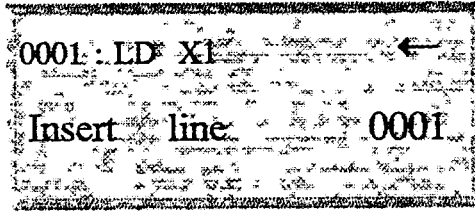
4.2.4 การค้นหาข้อมูลทั้งบรรทัดโดยใช้คำสั่งค้นหา(SRCH)

การค้นหาข้อมูลแบบนี้จะทำการค้นตั้งแต่บรรทัดที่เคอร์เซอร์อยู่ขึ้นไปเรื่อยๆจนถึงบรรทัด 255 การใช้งานทำได้ดังนี้



กดคีย์ข้อมูลในบรรทัดที่ต้องการจะค้นหา ถ้าไม่พบจะขึ้น Not found

#### 4.2.5 การแทรกข้อมูลโดยใช้คำสั่งแทรก ( INS )



เลื่อนเคอร์เซอร์ไปที่บรรทัดที่ต้องการแทรกข้อมูล แล้วกด ENT เครื่อง  
จะทำการเลื่อนข้อมูลในบรรทัดที่เคอร์เซอร์ปรากฏลงมาหนึ่งบรรทัด

#### 4.2.6 การลบบางส่วนของโปรแกรมโดยใช้คำสั่งลบ ( DEL )

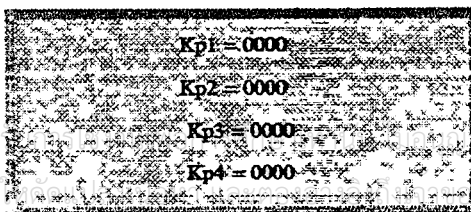
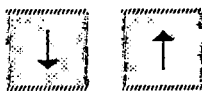


เลื่อนเคอร์เซอร์ไปที่บรรทัดที่ต้องการลบข้อมูล ถ้าถูกต้องแล้วให้กด  
ENT ข้อมูลในบรรทัดนั้นจะถูกลบ และจะเลื่อนข้อมูลในบรรทัด  
ด้านล่างขึ้นมาแทน

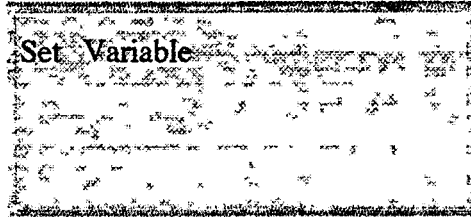
#### 4.2.7 การขอดูค่า KP,KI,KD,SV,PV และ MV



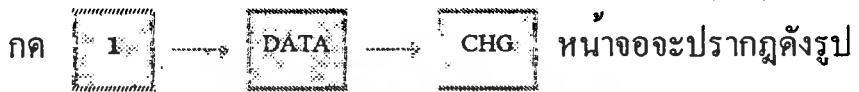
ตัวอย่างการขอดูค่า KP โดย กด KP หน้าจอจะขึ้นดังรูป จากนั้นกด  
DATA จะปรากฏข้อความดังรูป(ล่าง) ถ้าต้องการดูค่าที่เหลืออีกสี่  
ชุด ให้ใช้



#### 4.2.8 การเปลี่ยนแปลงค่าของ (KP,KI,KD,SV,MV)

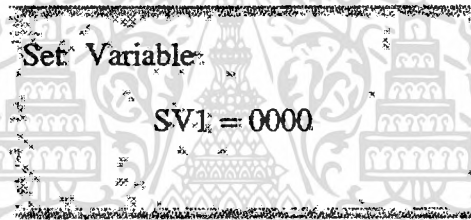


ตัวอย่างการเปลี่ยนค่า SV1 กดคีย์ SV หน้าจอจะปรากฏดังรูป(บน)



(ล่าง) ใส่ค่าที่ต้องการในตำแหน่งที่มีการกระพริบ เมื่อได้ค่าที่ต้องการให้กด

ENT

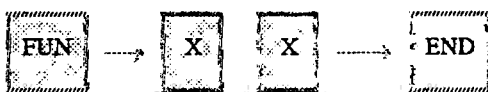


#### 4.3 การใช้ฟังก์ชันคีย์ (FUNCTION KEY)

นอกเหนือจากการใช้คำสั่งพื้นฐานแล้ว ยังมีฟังก์ชันคำสั่งอีก 7 คำสั่งดังนี้

- FUN 01 หมายถึงคำสั่ง END
- FUN 02 หมายถึงคำสั่ง บวก (SUM)
- FUN 03 หมายถึงคำสั่ง ลบ (SUB)
- FUN 04 หมายถึงคำสั่ง คูณ (MUL)
- FUN 05 หมายถึงคำสั่งหาร (DIV)
- FUN 06 หมายถึงคำสั่ง รากที่สอง (SQRT)
- FUN 07 หมายถึงคำสั่ง ยกกำลังสอง (SQR)

การใช้งานฟังก์ชัน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.4 หลักการเขียนโปรแกรม

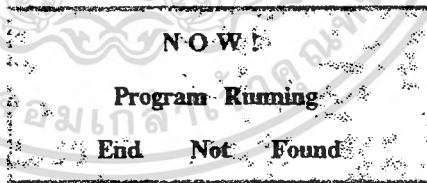
##### 4.4.1 การใช้คำสั่ง BSC

เป็นคำสั่งที่ใช้ในการประมวลผลแบบ PID ซึ่งมีอยู่ทั้งหมด 8 ชุด โดยแต่ละชุดจะรับข้อมูลจาก X(1-8)

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	X1
0001	LD	SV1
0002	BSC	
0003	OUT	Y1
0004	END	

##### 4.4.2 การใช้คำสั่ง END (FUN 01)

ใช้เมื่อสิ้นสุดการเขียนโปรแกรม แต่ถ้าไม่มีคำสั่ง END แล้วเมื่อผู้ใช้ทำการรัน โปรแกรมจะแสดงข้อความ



##### 4.4.3 การใช้คำสั่ง บวก (SUM : FUN 02)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการบวกกัน

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	X1
0001	LD	SV1

0002	SUM	
0003	OUT	Y1
0004	END	

#### 4.4.4 การใช้คำสั่ง ลบ (SUB : FUN 03)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการลบกัน โดยให้ข้อมูลในบรรทัดแรกเป็นตัวตั้ง และข้อมูลในบรรทัดต่อมาเป็นตัวลบ

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	X1
0001	LD	SV1
0002	SUB	
0003	OUT	Y1
0004	END	

#### 4.4.5 การใช้คำสั่ง คูณ (MUL : FUN 04)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการคูณกัน

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	SV1
0001	LD	SV2
0002	MUL	
0003	OUT	Y1
0004	END	

#### 4.4.6 การใช้คำสั่ง หาร (DIV : FUN 05)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการหารกัน  
โดยให้ข้อมูลในบรรทัดแรกเป็นตัวตั้ง และข้อมูลในบรรทัดต่อมา  
เป็นตัวหาร

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	SV1
0001	LD	SV2
0002	DIV	
0003	OUT	Y1
0004	END	

#### 4.4.7 การใช้คำสั่ง รากที่สอง (SQRT : FUN 06)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการหารากที่สอง

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	SV1
0001	SQRT	
0003	OUT	Y1
0004	END	

#### 4.4.8 การใช้คำสั่ง ยกกำลังสอง (SQR : FUN 07)

เป็นการนำข้อมูลในตำแหน่งที่กำหนดมาทำการยกกำลังสอง

ตำแหน่ง	คำสั่ง	ข้อมูล
0000	LD	SV1

0001	SQR	
0003	OUT	Y1
0004	END	



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### บทสรุป

#### ผลการทดลอง

การทำงานของเครื่องควบคุมชุดนี้ ในส่วนของการแสดงผลทางหน้าจอ ทางคณะผู้จัดทำได้ทำสำเร็จเรียบร้อยแล้ว โดยโครงสร้างหลักๆ ที่จำเป็นต้องใช้ในส่วนนี้ ที่ทางผู้จัดทำได้ทำไว้ คือ

- โครงสร้าง และ คำสั่งการเขียนโปรแกรม
- การรับค่าทางคีย์บอร์ด
- การแสดงค่าออกทาง LCD
- ฟังก์ชันการบวก
- คำสั่ง DELETE
- คำสั่ง SERCH
- คำสั่ง JUMP
- คำสั่งการเปลี่ยนค่า PARAMETER
- คำสั่งการเซทค่า PARAMETER

โดยที่รายละเอียดในส่วนต่าง ๆ สามารถจะศึกษาได้จากเนื้อหาในปริญญา นิพนธ์เล่มนี้ แต่อย่างไรก็ตามยังมีฟังก์ชันย่อยการทำงานของเครื่องควบคุมบาง ฟังก์ชันยังขาดไป โดยที่ทางคณะผู้จัดทำได้ทำการเขียนเป็น โครงสร้างหลักรองรับ ไว้แล้ว เพียงแต่นำฟังก์ชันย่อยที่ขาดไปใส่ลงไปในโครงสร้างหลักแล้วกำหนดพื้นที่ สำหรับการส่งค่า และ การรับค่าของฟังก์ชันย่อยนั้น ๆ ให้ตรงกับโครงสร้างหลัก ของโปรแกรม เครื่องควบคุมก็สามารถเรียกใช้ฟังก์ชันย่อยนั้น ๆ ได้เลย

ฟังก์ชันย่อยที่ต้องเพิ่มเติม คือ

- PID ฟังก์ชัน หรือ BSC ฟังก์ชัน
- SQUARE ROOT ฟังก์ชัน
- MULTIPLE ฟังก์ชัน
- DIVIDED ฟังก์ชัน
- SUBTRACT ฟังก์ชัน
- SQUARE ฟังก์ชัน



ภาคผนวก ก โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CPU "Z80.TBL"
```

```
HOF "INT8"
```

```
ORG 8000H
```

```
-----
```

```
; CONSTANT VALUE
```

```
-----
```

```
LCD_Ctrl: EQU 0A0h
```

```
Rd_Bflg: EQU 0A4h
```

```
Wr_Data: EQU 0A2h
```

```
Rd_Data: EQU 0A6h
```

```
PortB: EQU 51h
```

```
PortKey: EQU 0B0h
```

```
TopProg: EQU 0FB00H ; address for start program editor
```

```
BotProg: EQU 0FEFCH ; address for end program editor area <255 Line>  
; BotProg = TopProg + 3FCh
```

```
TopExec: EQU 0F600H ; address for start program execute
```

```
BotExec: EQU 0FAFCH ; address for end program execute area <255 Line>
```

```
Line_0: EQU 0
```

```
Line_1: EQU 1
```

```
Line_2: EQU 2
```

```
Line_3: EQU 3
```

```
-----
```

```
; ScanKey
```

```
-----
```

fun_k:	EQU	70h
bsc_k:	EQU	71h
ld_k:	EQU	72h
n07_k:	EQU	73h
n04_k:	EQU	74h
n01_k:	EQU	75h
n00_k:	EQU	76h
kp_k:	EQU	60h
x_k:	EQU	61h
y_k:	EQU	62h
n08_k:	EQU	63h
n05_k:	EQU	64h
n02_k:	EQU	65h
p_k:	EQU	66h
ki_k:	EQU	50h
reg_k:	EQU	51h
out_k:	EQU	52h
n09_k:	EQU	53h
n06_k:	EQU	54h
n03_k:	EQU	55h
clr_k:	EQU	56h
kd_k:	EQU	40h
f_r_k:	EQU	41h
sv_k:	EQU	42h
dta_k:	EQU	43h
set_k:	EQU	44h
rst_k:	EQU	45h
ver_k:	EQU	46h
min_k:	EQU	30h
flt_k:	EQU	31h
pv_k:	EQU	32h



```

chg_k: EQU 33h
del_k: EQU 34h
ins_k: EQU 35h
ent_k: EQU 36h
shft_k: EQU 20h
smp_k: EQU 21h
mv_k: EQU 22h
srch_k: EQU 23h
mont_k: EQU 24h
up_k: EQU 25h
down_k: EQU 26h

```

```

;-----
; MAIN PROGRAM BEGIN
;-----

```

```

main: CALL init
      CALL infodsp
      CALL delay
      CALL clrProg
      CALL clrdsp
      CALL beginval

```

```

loop: IM 1
      EI
      CALL chkport
      JP loop

```

```

;-----
; PROGRAM END
;-----

```

```

;-----
; Routine   : edit
; Description : edit program <simulation>
; Register  : A
;-----

```

```
edit: PUSH af
```

```
CALL dspl0_3
```

```
IN a,(PortB) ; check edit mode or run mode
```

```
BIT 5,a
```

```
JP z,exitedit
```

```
CALL ReadKey
```

```
branch1: LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,clr1
```

```
CP ld_k
```

```
JP z,ld1
```

```
CP out_k
```

```
JP z,out1
```

```
CP bsc_k
```

```
JP z,bsc1
```

```
CP fun_k
```

```
JP z,funcl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP srch\_k

JP z,srch1

CP kp\_k

JP z,kp1

CP ki\_k

JP z,ki1

CP kd\_k

JP z,kd1

CP sv\_k

JP z,sv1

CP pv\_k

JP z,pv1

CP mv\_k

JP z,mv1

CP min\_k

JP z,min1

CP shft\_k

JP z,shft1

CP n00\_k

JP z,number1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n01\_k

JP z,number1

CP n02\_k

JP z,number1

CP n03\_k

JP z,number1

CP n04\_k

JP z,number1

CP n05\_k

JP z,number1

CP n06\_k

JP z,number1

CP n07\_k

JP z,number1

CP n08\_k

JP z,number1

CP n09\_k

JP z,number1

CP del\_k

JP z,del1

เอกสารนี้เป็นเอกสาร **CP ins\_k** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP z,ins1

CP up\_k

JP z,up1

CP down\_k

JP z,down1

JP exitedit

ld1: CALL FLine0\_3

LD hl,clrline

CALL nextchr

CALL ldkey

JP exitedit

out1: CALL FLine0\_3

LD hl,clrline

CALL nextchr

CALL outkey

JP exitedit

bsc1: CALL FLine0\_3

LD hl,clrline

CALL nextchr

CALL bsckey

JP exitedit

func1: CALL FLine0\_3

LD hl,clrline

CALL nextchr

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**CALL funckey**

**JP exitedit**

**srch1: CALL srchkey**

**JP exitedit**

**kp1: CALL kpkey**

**JP exitedit**

**ki1: CALL kikey**

**JP exitedit**

**kd1: CALL kdkey**

**JP exitedit**

**sv1: CALL svkey**

**JP exitedit**

**pv1: CALL pvkey**

**JP exitedit**

**mv1: CALL mvkey**

**JP exitedit**

**min1: CALL minscale**

**JP exitedit**

**max1: CALL maxscale**

**JP exitedit**

เอกสารนี้เป็น **clr1: CALL clear** ทรัพย์สินของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitedit

```
number1: LD a,080h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL clrdisp
```

```
CALL jump
```

```
CALL delay
```

```
JP exitedit
```

```
del1: LD a,080h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL clrdisp
```

```
CALL delkey
```

```
CALL delay
```

```
JP exitedit
```

```
ins1: LD a,080h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL clrdisp
```

```
CALL inskey
```

```
CALL delay
```

```
JP exitedit
```

```
up1: LD a,080h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL clrdisp ; can't move clrdisp
```

```
CALL upkey
```

```
JP exitedit
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

down1: LD  a,080h    ;set dd ram
        CALL LCDctrl
        CALL clr dsp ; can't move clr dsp
        CALL downkey
        JP  exitedit

```

```

shft1: CALL readkey
        LD  a,(Kb_Buf)
        CP  clr_k
        JP  z,exitedit

```

```

        CP  min_k

```

```

        JP  z,max1

```

```

        JP  shft1

```

```

exitedit: POP af

```

```

        RET

```

```

;-----
; Routine   : Int_Serv

```

```

; Description : Interup service for keypressed

```

```

; Register  : A,B,DE,HL
;-----

```

```

Int_Serv: DI

```

```

        PUSH af

```

```

        PUSH bc

```

```

        PUSH de

```

```

        PUSH hl

```

LD de,(getout)

XOR a

SBC hl,de

LD a,l

CP 0fh

JP z,Buf\_full

LD hl,(getout)

LD de,(getin)

XOR a

SBC hl,de

LD a,l

CP 01h

JP z,Buf\_full

debouce: IN a,(PortKey)

LD c,a

CALL delayint

IN a,(PortKey)

CP c

JP nz,debouce

LD hl,(getin)

LD (hl),a

INC hl

LD (getin),hl

LD hl,(getin)

LD de,Buffer

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XOR a
SBC hl,de
LD a,l
CP 0fh
JP z,exitint
JP c,exitint

```

```

LD hl,Buffer
LD (getin),hl
JP exitint

```

```

Buf_full: CALL putdsp1

```

```

LD a,0d0h ;set dd ram

```

```

CALL LCDctrl

```

```

LD hl,BufErr

```

```

LD b,16

```

```

fulldsp: LD a,(hl)

```

```

CALL waitbusy

```

```

OUT (Wr_Data),a

```

```

CALL waitbusy

```

```

INC hl

```

```

DJNZ fulldsp

```

```

CALL delay

```

```

CALL putdsp1

```

```

exitint: POP hl

```

```

POP de

```

```

POP bc

```

```

POP af

```

```

CALL delayint

```

## RETI

```

;-----
; Routine   : readkey
; Description : read key from
; Register  : A,DE,HL
;-----

```

ReadKey: PUSH af

PUSH de

PUSH hl

LD hl,(getin) ;--high byte for getin = getout (30h)

LD de,(getout) ;

XOR a

SBC hl,de

JP z,clr\_Kbuf ; getin == getout

LD hl,(getout)

LD a,(hl)

LD (Kb\_Buf),a

INC hl

LD (getout),hl

LD de,Buffer

XOR a

SBC hl,de

LD a,l

CP 0fh

JP z,exitread

JP c,exitread

```
LD hl,Buffer
LD (getout),hl
JP exitread
```

```
clr_Kbuf: LD a,00
          LD (Kb_Buf),a
```

```
exitread: POP hl
          POP de
          POP af
          RET
```

```
-----
; Routine   : beginval
; Description : set begin value to variable
; Register  : HL
;-----
```

```
beginval: PUSH hl
```

```
LD hl,TopProg
LD (Line_Add),hl
LD (Top_Block),hl
LD (Cur_Addr),hl
LD (EndProg),hl
Ld (MaxLiUse),hl
```

```
LD hl,Buffer
LD (getin),hl
```

```
LD (getout),hl
```

LD a,01

LD (page),a

POP hl

RET

```

;-----
; Routine   : readprog
; Description: read data from program area
; Register  : A,HL
;-----

```

readprog: PUSH af

PUSH hl

LD hl,(Line\_Add)

LD a,(hl)

LD (firstchr),a

INC hl

LD a,(hl)

LD (secondch),a

INC hl

LD a,(hl)

LD (thirdchr),a

INC hl

LD a,(hl)

LD (fourthchr),a

first: LD a,(firstchr)

CP bsc\_k

JP z,bsc

CP ld\_k

JP z,ldd

CP out\_k

JP z,outt

CP fun\_k

JP z,func

JP exit\_Rd

func: CALL readfunc

JP exit\_Rd

bsc: LD hl,bscmess

CALL nextchr

JP second

ldd: LD hl,ldmess

CALL nextchr

JP second

outt: LD hl,outmess

CALL nextchr

เอกสารนี้เป็นเอกสาร **JP second** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exit\_Rd

second: LD a,(secondch)

CP x\_k

JP z,xx

CP y\_k

JP z,yy

CP reg\_k

JP z,r

CP mv\_k

JP z,mv

CP pv\_k

JP z,pv

CP sv\_k

JP z,sv

CALL comp0\_9

JP third

xx: LD hl,xbotton

CALL nextchr

JP third

yy: LD hl,ybotton

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

;-----
; Routine   : run
; Description : run program from edit area <simulation>
; Register  : hl
;-----

```

```
run: PUSH af
```

```
    PUSH hl
```

```
    LD hl,TopProg
```

```
    LD (Pointpro),hl
```

```
    LD hl,TopExec
```

```
    LD (Pointexe),hl
```

```
    LD a,01
```

```
    LD (Ldflag),a
```

```
    LD a,80h ;set dd ram
```

```
    CALL LCDctrl
```

```
    LD hl,runmess1
```

```
    CALL nextchr
```

```
    LD a,0c0h ;set dd ram
```

```
    CALL LCDctrl
```

```
    LD hl,runmess2
```

```
    CALL nextchr
```

```
    LD a,90h ;set dd ram
```

```
    CALL LCDctrl
```

```
    LD hl,runmess3
```

```
    CALL nextchr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,runmess4
```

```
CALL nextchr
```

```
chkf01: LD hl,(Pointpro)
```

```
LD a,(hl)
```

```
CP fun_k
```

```
JP nz,chkBot
```

```
INC hl
```

```
LD a,(hl)
```

```
CP n00_k
```

```
JP nz,chkBot
```

```
INC hl
```

```
LD a,(hl)
```

```
CP n01_k
```

```
JP nz,chkBot
```

```
JP z,runstart
```

```
chkBot: LD hl,(Pointpro)
```

```
LD de,04
```

```
XOR a
```

```
ADC hl,de
```

```
LD (Pointpro),hl
```

```
EX de,hl
```

```
LD hl,(BotProg)
```

```
XOR a
```

```

JP z,RunErr
JP chkf01

runstart: LD hl,TopProg
          LD (Pointpro),hl

checkrun: IN a,(PortB)      ; check edit mode or run mode
          BIT 5,a           ; 0 = runmode , 1 = edit mode
          JP nz,exitrun

          LD hl,(Pointpro)
          LD a,(hl)

          CP ld_k
          JP z,ldexe1

          CP out_k
          JP z,outexe1

          CP bsc_k
          JP z,bscexe1

          CP fun_k
          JP z,funexe1

          INC hl
          INC hl
          INC hl
          INC hl
          LD (Pointpro),hl

          JP checkrun

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ldexe1: CALL Ldexe2

JP checkrun

outexe1: CALL Outexe2

JP checkrun

bscexe1: CALL Bscexe2

JP checkrun

funexe1: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP n00\_k

JP z,F0\_exe

; CP n01\_k

; JP F1\_exe

INC hl

INC hl

INC hl

LD (Pointpro),hl

JP checkrun

F0\_exe: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n01\_k  
JP z,F01run

CP n02\_k  
JP z,F02run

CP n03\_k  
JP z,F03run

CP n04\_k  
JP z,F04run

CP n05\_k  
JP z,F05run

CP n06\_k  
JP z,F06run

CP n07\_k  
JP z,F07run

INC hl  
INC hl  
LD (Pointpro),hl  
JP checkrun

```
RunErr: LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,RunErr01
LD b,16
```

เอกสารนี้ **RunErrdsp**: ที่ **LD a,(hl)** ทรัพยากรใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL waitbusy
OUT (Wr_Data),a
CALL waitbusy
INC hi
DJNZ RunErdsp

```

```

runloop: IN a,(PortB)      ; check edit mode or run mode
        BIT 5,a
        JP z,runloop
        JP exitrun

```

```
F01run: JP F01Exec
```

```
F02run: CALL F02Exec
        JP checkrun
```

```
F03run: CALL F03Exec
        JP checkrun
```

```
F04run: CALL F04Exec
        JP checkrun
```

```
F05run: CALL F05Exec
        JP checkrun
```

```
F06run: CALL F06Exec
        JP checkrun
```

```
F07run: CALL F07Exec
        JP checkrun
```

**exitrun: CALL clrdsp**

**POP hl**

**POP af**

**RET**

-----

**; Routine : PutCall**

**; Description : Put CALL code for run**

**; Register :**

-----

**PutCall: LD hl,(Pointpro)**

**INC hl**

**INC hl**

**LD (Pointpro),hl**

**LD hl,(Pointexe)**

**LD (hl),0cdh**

**LD hl,(Pointexe)**

**INC hl**

**LD (Pointexe),hl**

**RET**

-----

**; Routine : CallAddr**

**; Description : Put CALL code for run**

**; Register :**

-----

**CallAddr: LD hl,(Pointexe)**

**LD (hl),e**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

INC hl
LD (hl),d
INC hl
LD (Pointxe),hl

```

```

RET

```

```

;-----
; Routine   : Ldexe2
; Description : Put code for run LD .
; Register  :
;-----

```

```

Ldexe2: PUSH hl

```

```

LD hl,(Pointpro)
INC hl
LD (Pointpro),hl
LD a,(hl)

```

```

CP sv_k
JP z,LdsvX

```

```

CP pv_k
JP z,LdpvX

```

```

CP mv_k
JP z,LdmvX

```

```

CP x_k
JP z,LdxX

```

INC hl

INC hl

INC hl

LD (Pointpro),hl

JP exLdexe2

LdsvX: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP n01\_k

JP z,Lsv1exe

CP n02\_k

JP z,Lsv2exe

CP n03\_k

JP z,Lsv3exe

CP n04\_k

JP z,Lsv4exe

CP n05\_k

JP z,Lsv5exe

CP n06\_k

JP z,Lsv6exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n07\_k  
JP z,Lsv7exe

CP n08\_k  
JP z,Lsv8exe

INC hl  
INC hl  
LD (Pointpro),hl  
JP exLdexe2

Lsv1exe: CALL PutCall  
LD de,Ldsv1exe  
CALL CallAddr  
JP exLdexe2

Lsv2exe: CALL PutCall  
LD de,Ldsv2exe  
CALL CallAddr  
JP exLdexe2

Lsv3exe: CALL PutCall  
LD de,Ldsv3exe  
CALL CallAddr  
JP exLdexe2

Lsv4exe: CALL PutCall  
LD de,Ldsv4exe  
CALL CallAddr  
JP exLdexe2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lsv5exe: CALL PutCall

LD de,Ldsv5exe

CALL CallAddr

JP exLdexe2

Lsv6exe: CALL PutCall

LD de,Ldsv6exe

CALL CallAddr

JP exLdexe2

Lsv7exe: CALL PutCall

LD de,Ldsv7exe

CALL CallAddr

JP exLdexe2

Lsv8exe: CALL PutCall

LD de,Ldsv8exe

CALL CallAddr

JP exLdexe2

LdpvX: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP n01\_k

JP z,Lpv1exe

CP n02\_k

JP z,Lpv2exe

CP n03\_k

JP z,Lpv3exe

CP n04\_k

JP z,Lpv4exe

CP n05\_k

JP z,Lpv5exe

CP n06\_k

JP z,Lpv6exe

CP n07\_k

JP z,Lpv7exe

CP n08\_k

JP z,Lpv8exe

INC hl

INC hl

LD (Pointpro),hl

JP exLdexe2

Lpv1exe: CALL PutCall

LD de,Ldpv1exe

CALL CallAddr

JP exLdexe2

Lpv2exe: CALL PutCall

LD de,Ldpv2exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL CallAddr

JP exLdexe2

Lpv3exe: CALL PutCall

LD de,Ldpv3exe

CALL CallAddr

JP exLdexe2

Lpv4exe: CALL PutCall

LD de,Ldpv4exe

CALL CallAddr

JP exLdexe2

Lpv5exe: CALL PutCall

LD de,Ldpv5exe

CALL CallAddr

JP exLdexe2

Lpv6exe: CALL PutCall

LD de,Ldpv6exe

CALL CallAddr

JP exLdexe2

Lpv7exe: CALL PutCall

LD de,Ldpv7exe

CALL CallAddr

JP exLdexe2

Lpv8exe: CALL PutCall

LD de,Ldpv8exe

CALL CallAddr

JP exLdexe2

LdmvX: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP n01\_k

JP z,Lmv1exe

CP n02\_k

JP z,Lmv2exe

CP n03\_k

JP z,Lmv3exe

CP n04\_k

JP z,Lmv4exe

CP n05\_k

JP z,Lmv5exe

CP n06\_k

JP z,Lmv6exe

CP n07\_k

JP z,Lmv7exe

CP n08\_k

JP z,Lmv8exe

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INC hl  
 INC hl  
 LD (Pointpro),hl  
 JP exLdexe2

Lmv1exe: CALL PutCall  
 LD de,Ldmv1exe  
 CALL CallAddr  
 JP exLdexe2

Lmv2exe: CALL PutCall  
 LD de,Ldmv2exe  
 CALL CallAddr  
 JP exLdexe2

Lmv3exe: CALL PutCall  
 LD de,Ldmv3exe  
 CALL CallAddr  
 JP exLdexe2

Lmv4exe: CALL PutCall  
 LD de,Ldmv4exe  
 CALL CallAddr  
 JP exLdexe2

Lmv5exe: CALL PutCall  
 LD de,Ldmv5exe  
 CALL CallAddr  
 JP exLdexe2

Lmv6exe: CALL PutCall

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD de,Ldmv6exe
CALL CallAddr
JP exLdexe2
```

```
Lmv7exe: CALL PutCall
LD de,Ldmv7exe
CALL CallAddr
JP exLdexe2
```

```
Lmv8exe: CALL PutCall
LD de,Ldmv8exe
CALL CallAddr
JP exLdexe2
```

```
LdxX: LD hl,(Pointpro)
INC hl
LD (Pointpro),hl
LD a,(hl)
```

```
CP n01_k
JP z,Lx1exe
```

```
CP n02_k
JP z,Lx2exe
```

```
CP n03_k
JP z,Lx3exe
```

```
CP n04_k
JP z,Lx4exe
```

CP n05\_k  
JP z,Lx5exe

CP n06\_k  
JP z,Lx6exe

CP n07\_k  
JP z,Lx7exe

CP n08\_k  
JP z,Lx8exe

INC hl  
INC hl  
LD (Pointpro),hl  
JP exLdexe2

Lx1exe: CALL PutCall  
LD de,Ldx1exe  
CALL CallAddr  
JP exLdexe2

Lx2exe: CALL PutCall  
LD de,Ldx2exe  
CALL CallAddr  
JP exLdexe2

Lx3exe: CALL PutCall  
LD de,Ldx3exe  
CALL CallAddr

JP exLdexe2

Lx4exe: CALL PutCall  
 LD de,Ldx4exe  
 CALL CallAddr  
 JP exLdexe2

Lx5exe: CALL PutCall  
 LD de,Ldx5exe  
 CALL CallAddr  
 JP exLdexe2

Lx6exe: CALL PutCall  
 LD de,Ldx6exe  
 CALL CallAddr  
 JP exLdexe2

Lx7exe: CALL PutCall  
 LD de,Ldx7exe  
 CALL CallAddr  
 JP exLdexe2

Lx8exe: CALL PutCall  
 LD de,Ldx8exe  
 CALL CallAddr  
 JP exLdexe2

exLdexe2: POP hl

RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; Routine   : Ldsv1exe
; Description : LD Sv1 execute
; Register  :
;-----

```

Ldsv1exe: LD a,(Ldflag)

CP 01

JP z,L1sv1exe

CP 02

JP z,L2sv1exe

JP xLsv1

L1sv1exe: LD hl,(Sv1Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv1

L2sv1exe: LD hl,(Sv1Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLsv1

xLsv1: RET

```

;-----
; Routine   : Ldsv2exe
; Description : LD Sv2 execute
; Register  :
;-----

```

Ldsv2exe: LD a,(Ldflag)

CP 01

JP z,L1sv2exe

CP 02

JP z,L2sv2exe

JP xLsv2

L1sv2exe: LD hl,(Sv2Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv2

L2sv2exe: LD hl,(Sv2Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLsv2

xLsv2: RET

-----  
 ; Routine : Ldsv3exe  
 ; Description : LD Sv3 execute  
 ; Register :  
 -----

Ldsv3exe: LD a,(Ldflag)

CP 01

JP z,L1sv3exe

CP 02

JP z,L2sv3exe

JP xLsv3

L1sv3exe: LD hl,(Sv3Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv3

L2sv3exe: LD hl,(Sv3Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLsv3

xLsv3: RET

-----

; Routine : Ldsv4exe

; Description : LD Sv4 execute

; Register :

-----

Ldsv4exe: LD a,(Ldflag)

CP 01

JP z,L1sv4exe

CP 02

JP z,L2sv4exe

JP xLsv4

L1sv4exe: LD hl,(Sv4Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv4

L2sv4exe: LD hl,(Sv4Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLsv4

xLsv4: RET

-----

```

; Routine   : Ldsv5exe
; Description : LD Sv5 execute
; Register  :

```

```

;-----

```

```

Ldsv5exe: LD a,(Ldflag)

```

```

          CP 01

```

```

          JP z,L1sv5exe

```

```

          CP 02

```

```

          JP z,L2sv5exe

```

```

          JP xLsv5

```

```

L1sv5exe: LD hl,(Sv5Buf)

```

```

          LD (Input1),hl

```

```

          LD a,02

```

```

          LD (Ldflag),a

```

```

          JP xLsv5

```

```

L2sv5exe: LD hl,(Sv5Buf)

```

```

          LD (Input2),hl

```

```

          LD a,01

```

```

          LD (Ldflag),a

```

```

          JP xLsv5

```

```

xLsv5:   RET

```

```

;-----

```

```

; Routine   : Ldsv6exe
; Description : LD Sv6 execute
; Register  :

```

```

;-----

```

```

Ldsv6exe: LD a,(Ldflag)

```

```

          CP 01

```

```

          JP z,L1sv6exe

```

```

          CP 02

```

JP z,L2sv6exe

JP xLsv6

L1sv6exe: LD hl,(Sv6Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv6

L2sv6exe: LD hl,(Sv6Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLsv6

xLsv6: RET

-----  
 ; Routine : Ldsv7exe  
 ; Description : LD Sv7 execute  
 ; Register :  
 -----

Ldsv7exe: LD a,(Ldflag)

CP 01

JP z,L1sv7exe

CP 02

JP z,L2sv7exe

JP xLsv7

L1sv7exe: LD hl,(Sv7Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLsv7

L2sv7exe: LD hl,(Sv7Buf)

```

LD (Input2),hl
LD a,01
LD (Ldflag),a
JP xLsv7
xLsv7:  RET

```

```

;-----
; Routine   : Ldsv8exe
; Description : LD Sv8 execute
; Register  :
;-----

```

```

Ldsv8exe: LD a,(Ldflag)
          CP 01
          JP z,L1sv8exe
          CP 02
          JP z,L2sv8exe
          JP xLsv8
L1sv8exe: LD hl,(Sv8Buf)
          LD (Input1),hl
          LD a,02
          LD (Ldflag),a
          JP xLsv8
L2sv8exe: LD hl,(Sv8Buf)
          LD (Input2),hl
          LD a,01
          LD (Ldflag),a
          JP xLsv8
xLsv8:   RET

```

```

;-----
; Routine   : Ldpv1exe

```

; Description : LD Pv1 execute

; Register :

-----

Ldpv1exe: LD a,(Ldflag)

CP 01

JP z,L1pv1exe

CP 02

JP z,L2pv1exe

JP xLpv1

L1pv1exe: LD hl,(Pv1Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv1

L2pv1exe: LD hl,(Pv1Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLpv1

xLpv1: RET

-----

; Routine : Ldpv2exe

; Description : LD Pv2 execute

; Register :

-----

Ldpv2exe: LD a,(Ldflag)

CP 01

JP z,L1pv2exe

CP 02

JP z,L2pv2exe

JP xLpv2

L1pv2exe: LD hl,(Pv2Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv2

L2pv2exe: LD hl,(Pv2Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLpv2

xLpv2: RET

-----  
 ; Routine : Ldpv3exe  
 ; Description : LD Pv3 execute  
 ; Register :  
 -----

Ldpv3exe: LD a,(Ldflag)

CP 01

JP z,L1pv3exe

CP 02

JP z,L2pv3exe

JP xLpv3

L1pv3exe: LD hl,(Pv3Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv3

L2pv3exe: LD hl,(Pv3Buf)

LD (Input2),hl

```

LD a,01
LD (Ldflag),a
JP xLpv3
xLpv3: RET

```

-----

```

; Routine : Ldpv4exe
; Description : LD Pv4 execute
; Register :

```

-----

```

Ldpv4exe: LD a,(Ldflag)
          CP 01
          JP z,L1pv4exe
          CP 02
          JP z,L2pv4exe
          JP xLpv4
L1pv4exe: LD hl,(Pv4Buf)
          LD (Input1),hl
          LD a,02
          LD (Ldflag),a
          JP xLpv4
L2pv4exe: LD hl,(Pv4Buf)
          LD (Input2),hl
          LD a,01
          LD (Ldflag),a
          JP xLpv4
xLpv4: RET

```

-----

```

; Routine : Ldpv5exe

```

```

; Description : LD Pv5 execute

```

; Register :

-----

Ldpv5exe: LD a,(Ldflag)

CP 01

JP z,L1pv5exe

CP 02

JP z,L2pv5exe

JP xLpv5

L1pv5exe: LD hl,(Pv5Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv5

L2pv5exe: LD hl,(Pv5Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLpv5

xLpv5: RET

-----

; Routine : Ldpv6exe

; Description : LD Pv6 execute

; Register :

-----

Ldpv6exe: LD a,(Ldflag)

CP 01

JP z,L1pv6exe

CP 02

JP z,L2pv6exe

JP xLpv6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

L1pv6exe: LD hl,(Pv6Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv6

L2pv6exe: LD hl,(Pv6Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLpv6

xLpv6: RET

-----

; Routine : Ldpv7exe

; Description : LD Pv7 execute

; Register :

-----

Ldpv7exe: LD a,(Ldflag)

CP 01

JP z,L1pv7exe

CP 02

JP z,L2pv7exe

JP xLpv7

L1pv7exe: LD hl,(Pv7Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLpv7

L2pv7exe: LD hl,(Pv7Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLpv7

```

LD (Ldflag),a
JP xLpv7
xLpv7: RET

```

```

;-----
; Routine : Ldpv8exe
; Description : LD Pv8 execute
; Register :
;-----

```

```
Ldpv8exe: LD a,(Ldflag)
```

```
CP 01
```

```
JP z,L1pv8exe
```

```
CP 02
```

```
JP z,L2pv8exe
```

```
JP xLpv8
```

```
L1pv8exe: LD hl,(Pv8Buf)
```

```
LD (Input1),hl
```

```
LD a,02
```

```
LD (Ldflag),a
```

```
JP xLpv8
```

```
L2pv8exe: LD hl,(Pv8Buf)
```

```
LD (Input2),hl
```

```
LD a,01
```

```
LD (Ldflag),a
```

```
JP xLpv8
```

```
xLpv8: RET
```

```

;-----
; Routine : Ldmv1exe
; Description : LD Mv1 execute

```

เอกสาร; Register : ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-----

Ldmv1exe: LD a,(Ldflag)

CP 01

JP z,L1mv1exe

CP 02

JP z,L2mv1exe

JP xLmv1

L1mv1exe: LD hl,(Mv1Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLmv1

L2mv1exe: LD hl,(Mv1Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv1

xLmv1: RET

-----

; Routine : Ldmv2exe

; Description : LD Mv2 execute

; Register :

-----

Ldmv2exe: LD a,(Ldflag)

CP 01

JP z,L1mv2exe

CP 02

JP z,L2mv2exe

JP xLmv2

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD (Input1),hl
```

```
LD a,02
```

```
LD (Ldflag),a
```

```
JP xLmv2
```

```
L2mv2exe: LD hl,(Mv2Buf)
```

```
LD (Input2),hl
```

```
LD a,01
```

```
LD (Ldflag),a
```

```
JP xLmv2
```

```
xLmv2: RET
```

```
-----
```

```
; Routine : Ldmv3exe
```

```
; Description : LD Mv3 execute
```

```
; Register :
```

```
-----
```

```
Ldmv3exe: LD a,(Ldflag)
```

```
CP 01
```

```
JP z,L1mv3exe
```

```
CP 02
```

```
JP z,L2mv3exe
```

```
JP xLmv3
```

```
L1mv3exe: LD hl,(Mv3Buf)
```

```
LD (Input1),hl
```

```
LD a,02
```

```
LD (Ldflag),a
```

```
JP xLmv3
```

```
L2mv3exe: LD hl,(Mv3Buf)
```

```
LD (Input2),hl
```

```
LD a,01
```

```
LD (Ldflag),a
```

JP xLmv3

xLmv3: RET

-----

; Routine : Ldmv4exe

; Description : LD Mv4 execute

; Register :

-----

Ldmv4exe: LD a,(Ldflag)

CP 01

JP z,L1mv4exe

CP 02

JP z,L2mv4exe

JP xLmv4

L1mv4exe: LD hl,(Mv4Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLmv4

L2mv4exe: LD hl,(Mv4Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv4

xLmv4: RET

-----

; Routine : Ldmv5exe

; Description : LD Mv5 execute

; Register :

Ldmv5exe: LD a,(Ldflag)

CP 01

JP z,L1mv5exe

CP 02

JP z,L2mv5exe

JP xLmv5

L1mv5exe: LD hl,(Mv5Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLmv5

L2mv5exe: LD hl,(Mv5Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv5

xLmv5: RET

-----  
 ; Routine : Ldmv6exe  
 ; Description : LD Mv6 execute  
 ; Register :  
 -----

Ldmv6exe: LD a,(Ldflag)

CP 01

JP z,L1mv6exe

CP 02

JP z,L2mv6exe

JP xLmv6

L1mv6exe: LD hl,(Mv6Buf)

เอกสารนี้เป็นเอกสารที่ LD (Input1),hl ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD a,02

LD (Ldflag),a

JP xLmv6

L2mv6exe: LD hl,(Mv6Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv6

xLmv6: RET

-----

; Routine : Ldmv7exe

; Description : LD Mv7 execute

; Register :

-----

Ldmv7exe: LD a,(Ldflag)

CP 01

JP z,L1mv7exe

CP 02

JP z,L2mv7exe

JP xLmv7

L1mv7exe: LD hl,(Mv7Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLmv7

L2mv7exe: LD hl,(Mv7Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv7

xLmv7: RET

-----

; Routine : Ldmv8exe

; Description : LD Mv8 execute

; Register :

-----

Ldmv8exe: LD a,(Ldflag)

CP 01

JP z,L1mv8exe

CP 02

JP z,L2mv8exe

JP xLmv8

L1mv8exe: LD hl,(Mv8Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLmv8

L2mv8exe: LD hl,(Mv8Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLmv8

xLmv8: RET

-----

; Routine : Ldx1exe

; Description : LD X1 execute

; Register :

-----

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ldx1exe: LD a,(Ldflag)

CP 01

JP z,L1x1exe

CP 02

JP z,L2x1exe

JP xLx1

L1x1exe: LD hl,(X1Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLx1

L2x1exe: LD hl,(X1Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLx1

xLx1: RET

-----

; Routine : Ldx2exe

; Description : LD X2 execute

; Register :

-----

Ldx2exe: LD a,(Ldflag)

CP 01

JP z,L1x2exe

CP 02

JP z,L2x2exe

JP xLx2

L1x2exe: LD hl,(X2Buf)

เอกสารนี้เป็นเอกสาร LD (Input1),hl ราชการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD a,02

LD (Ldflag),a

JP xLx2

L2x2exe: LD hl,(X2Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLx2

xLx2: RET

-----

; Routine : Ldx3exe

; Description : LD X3 execute

; Register :

-----

Ldx3exe: LD a,(Ldflag)

CP 01

JP z,L1x3exe

CP 02

JP z,L2x3exe

JP xLx3

L1x3exe: LD hl,(X3Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLx3

L2x3exe: LD hl,(X3Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLx3

xLx3: RET

-----  
 ; Routine : Ldx4exe  
 ; Description : LD X4 execute  
 ; Register :  
 -----

Ldx4exe: LD a,(Ldflag)

CP 01

JP z,L1x4exe

CP 02

JP z,L2x4exe

JP xLx4

L1x4exe: LD hl,(X4Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLx4

L2x4exe: LD hl,(X4Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLx4

xLx4: RET

-----  
 ; Routine : Ldx5exe  
 ; Description : LD X5 execute  
 ; Register :  
 -----

Ldx5exe: LD a,(Ldflag)

CP 01

JP z,L1x5exe

CP 02

JP z,L2x5exe

JP xLx5

L1x5exe: LD hl,(X5Buf)

LD (Input1),hl

LD a,02

LD (Ldflag),a

JP xLx5

L2x5exe: LD hl,(X5Buf)

LD (Input2),hl

LD a,01

LD (Ldflag),a

JP xLx5

xLx5: RET

-----  
 ; Routine : Ldx6exe  
 ; Description : LD X6 execute  
 ; Register :  
 -----

Ldx6exe: LD a,(Ldflag)

CP 01

JP z,L1x6exe

CP 02

JP z,L2x6exe

JP xLx6

L1x6exe: LD hl,(X6Buf)

LD (Input1),hl

LD a,02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD (Ldflag),a
JP xLx6
L2x6exe: LD hl,(X6Buf)
LD (Input2),hl
LD a,01
LD (Ldflag),a
JP xLx6
xLx6: RET

```

```

;-----
; Routine   : Ldx7exe
; Description : LD X7 execute
; Register  :
;-----

```

```

Ldx7exe: LD a,(Ldflag)
CP 01
JP z,L1x7exe
CP 02
JP z,L2x7exe
JP xLx7

```

```

L1x7exe: LD hl,(X7Buf)
LD (Input1),hl
LD a,02
LD (Ldflag),a
JP xLx7

```

```

L2x7exe: LD hl,(X7Buf)
LD (Input2),hl
LD a,01
LD (Ldflag),a
JP xLx7

```

```

xLx7: RET

```

```

;-----
; Routine   : Ldx8exe
; Description : LD X8 execute
; Register  :
;-----

```

```
Ldx8exe: LD a,(Ldflag)
```

```
CP 01
```

```
JP z,L1x8exe
```

```
CP 02
```

```
JP z,L2x8exe
```

```
JP xLx8
```

```
L1x8exe: LD hl,(X8Buf)
```

```
LD (Input1),hl
```

```
LD a,02
```

```
LD (Ldflag),a
```

```
JP xLx8
```

```
L2x8exe: LD hl,(X8Buf)
```

```
LD (Input2),hl
```

```
LD a,01
```

```
LD (Ldflag),a
```

```
JP xLx8
```

```
xLx8: RET
```

```

;-----
; Routine   : Outexe2
; Description : Put code for run OUT
; Register  :
;-----

```

```
Outexe2: PUSH hl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP sv\_k

JP z,OusvX

INC hl

INC hl

INC hl

LD (Pointpro),hl

JP exOusvex

OusvX: LD hl,(Pointpro)

INC hl

LD (Pointpro),hl

LD a,(hl)

CP n01\_k

JP z,Osv1exe

CP n02\_k

JP z,Osv2exe

CP n03\_k

JP z,Osv3exe

CP n04\_k

JP z,Osv4exe

CP n05\_k

JP z,Osv5exe

CP n06\_k

JP z,Osv6exe

CP n07\_k

JP z,Osv7exe

CP n08\_k

JP z,Osv8exe

INC hl

INC hl

LD (Pointpro),hl

JP exOsvex

Osv1exe: CALL PutCall

LD de,Osv1exe

CALL CallAddr

JP exOsvex

Osv2exe: CALL PutCall

LD de,Osv2exe

CALL CallAddr

JP exOsvex

Osv3exe: CALL PutCall

LD de,Osv3exe

CALL CallAddr

JP exOsvex

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Osv4exe: CALL PutCall  
 LD de,Ousv4exe  
 CALL CallAddr  
 JP exOusvex

Osv5exe: CALL PutCall  
 LD de,Ousv5exe  
 CALL CallAddr  
 JP exOusvex

Osv6exe: CALL PutCall  
 LD de,Ousv6exe  
 CALL CallAddr  
 JP exOusvex

Osv7exe: CALL PutCall  
 LD de,Ousv7exe  
 CALL CallAddr  
 JP exOusvex

Osv8exe: CALL PutCall  
 LD de,Ousv8exe  
 CALL CallAddr  
 JP exOusvex

exOusvex: POP hl  
 RET

```

; Routine   : Ousv1exe
; Description : LD Sv1 execute
; Register  :

```

```

;-----

```

```

Ousv1exe: LD hl,(Output)
          LD (Sv1Buf),hl
          LD a,01
          LD (Ldflag),a

```

```

RET

```

```

;-----

```

```

; Routine   : Ousv2exe
; Description : LD Sv2 execute
; Register  :

```

```

;-----

```

```

Ousv2exe: LD hl,(Output)
          LD (Sv2Buf),hl
          LD a,01
          LD (Ldflag),a

```

```

RET

```

```

;-----

```

```

; Routine   : Ousv3exe
; Description : LD Sv3 execute
; Register  :

```

```

;-----

```

```

Ousv3exe: LD hl,(Output)
          LD (Sv3Buf),hl

```

```

LD a,01

```

LD (Ldflag),a

RET

-----  
 ; Routine : Ousv4exe  
 ; Description : LD Sv4 execute  
 ; Register :

Ousv4exe: LD hl,(Output)  
 LD (Sv4Buf),hl  
 LD a,01  
 LD (Ldflag),a  
 RET

-----  
 ; Routine : Ousv5exe  
 ; Description : LD Sv5 execute  
 ; Register :

Ousv5exe: LD hl,(Output)  
 LD (Sv5Buf),hl  
 LD a,01  
 LD (Ldflag),a  
 RET

-----  
 ; Routine : Ousv6exe  
 ; Description : LD Sv6 execute

; Register :

-----

Ousv6exe: LD hl,(Output)

LD (Sv6Buf),hl

LD a,01

LD (Ldflag),a

RET

-----

; Routine : Ousv7exe

; Description : LD Sv7 execute

; Register :

-----

Ousv7exe: LD hl,(Output)

LD (Sv7Buf),hl

LD a,01

LD (Ldflag),a

RET

-----

; Routine : Ousv8exe

; Description : LD Sv8 execute

; Register :

-----

Ousv8exe: LD hl,(Output)

LD (Sv8Buf),hl

LD a,01

LD (Ldflag),a

RET

```

;-----
; Routine   : Bscexe2
; Description : Put code for run BSC
; Register  :
;-----

```

Bscexe2: nop

RET

```

;-----
; Routine   : F01Exec
; Description : Function 01 END (use by JP)
; Register  :
;-----

```

F01Exec: LD hl,(Pointexe)

LD (hl),0c3h ; Code for JP

LD hl,(Pointexe)

INC hl

LD (Pointexe),hl

LD de,FuncEND

LD (hl),e

INC hl

LD (hl),d

INC hl

LD (Pointexe),hl

JP TopExec

```

;-----

```

เอกสารนี้เป็นเอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : Function 01 END (use by JP)

; Register :

-----

FuncEND: CALL Readkey

LD a,(Kb\_Buf)

CP kp\_k

JP z,kpexe

CP ki\_k

JP z,kiexe

CP kd\_k

JP z,kdexe

CP sv\_k

JP z,svexe

CP pv\_k

JP z,pvexe

CP mv\_k

JP z,mvexe

CP min\_k

JP z,minexe

CP shft\_k

JP z,shftexe

JP chkmode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

kpexe: CALL kpkey  
JP chkmode

kiexe: CALL kikey  
JP chkmode

kdexe: CALL kdkey  
JP chkmode

svexe: CALL svkey  
JP chkmode

pvexe: CALL pvkey  
JP chkmode

mvexe: CALL mvkey  
JP chkmode

minexe: CALL minscale  
JP chkmode

maxexe: CALL maxscale  
JP chkmode

shftexe: CALL readkey  
LD a,(Kb\_Buf)  
CP clr\_k  
JP z,chkmode

CP min\_k

JP z,maxexe

JP shftexe

chkmode: IN a,(PortB) ; check edit mode or run mode

BIT 5,a ; 0 = runmode , 1 = edit mode

JP z,TopExec

JP nz,exitrun

-----

; Routine : F02Exec

; Description : Function 02 SUM

; Register :

-----

F02Exec: CALL PutCall

LD de,FuncSum

CALL CallAddr

RET

-----

; Routine : FuncSum

; Description : Function 02 SUM

; Register :

-----

FuncSum: PUSH hl

PUSH de

LD hl,(Input1)

LD de,(Input2)

XOR a

ADC hl,de

LD (Output),hl

เอกสารนี้เป็นเอกสารให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POP de

POP hl

RET

-----

; Routine : F03Exec

; Description : Function 03 SUB

; Register :

-----

F03Exec: nop

RET

-----

; Routine : F04Exec

; Description : Function 04 MUL

; Register :

-----

F04Exec: nop

RET

-----

; Routine : F05Exec

; Description : Function 05 DIV

; Register :

-----

F05Exec: nop

RET

-----

; Routine : F06Exec

; Description : Function 06 SQRT

; Register :

-----

F06Exec: nop

RET

```

;-----
; Routine   : F07Exec
; Description : Function 07 SQR
; Register  :
;-----

```

F07Exec: nop

RET

```

;-----
; Routine   : clrProg
; Description : Clear Program area
; Register  : A,DE,HL
;-----

```

```

clrProg: PUSH af
        PUSH de
        PUSH hl

```

```

        LD hl,TopProg

```

```

        LD de,BotProg+4

```

```

clrloop: LD (hl),00

```

```

        INC hl

```

```

        PUSH hl

```

```

        XOR a

```

```

        SBC hl,de

```

```

        POP hl

```

```

        JP c,clrloop

```

POP de

POP af

RET

```

;-----
; Routine   : chkport
; Description : simulate port checking run mode or edit mode
; Register  : A,HL
;-----

```

```

chkport: PUSH af
        PUSH hl

        IN a,(PortB) ; check edit mode or run mode
        BIT 5,a
        JP z,runmode
        JP nz,editmode
        JP exitport ;---- else exit chkport

```

```

runmode: CALL run
        JP exitport

```

```

editmode: CALL edit
        JP exitport

```

```

exitport: POP hl
        POP af

```

RET

```

; Routine   : dspl0_3
; Description : shift up 1 line
; Register  : DE,HL

```

```

;-----

```

```

dspl0_3: PUSH af
        PUSH de
        PUSH hl

```

```

CALL curdsp

```

```

CALL chkline

```

```

CALL chkline

```

```

CALL chkline

```

```

CALL chkline

```

```

XOR a    ;clear carry

```

```

LD hl,(Line_Add)

```

```

LD de,0010h

```

```

SBC hl,de

```

```

LD (Line_Add),hl

```

```

CALL getdsp1

```

```

POP hl

```

```

POP de

```

```

POP af

```

```

RET

```

```

;-----

```

```

; Routine   : getdsp1

```

```

; Description : read data from LCD then write to Screen buffer

```

```

; Register  : A,BC,HL

```

```

getdsp1:  PUSH af
          PUSH bc
          PUSH hl

          LD  a,080h    ;set dd ram
          CALL LCDctrl
          LD  hl, Scr1L0
          LD  b,72

WrL:     CALL waitbusy
          IN  a,(Rd_Data)
          LD  (hl),a
          INC hl
          DJNZ WrL

          POP hl
          POP bc
          POP af

          RET

```

```

;-----
; Routine      :  putdsp1
; Description   :  read data from Screen buffer then write to LCD
; Register     :  A,B,HL
;-----

```

```

putdsp1:  PUSH af
          PUSH bc
          PUSH hl

          LD  a,80h    ;set dd ram
          CALL LCDctrl

```

```
LD hl, Scr1L0
```

```
LD b,16
```

```
RdL0: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ RdL0
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl, Scr1L1
```

```
LD b,16
```

```
RdL1: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ RdL1
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl, Scr1L2
```

```
LD b,16
```

```
RdL2: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ RdL2
```

```

LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl, Scr1L3
LD b,16
RdL3: LD a,(hl)
CALL waitbusy
OUT (Wr_Data),a
CALL waitbusy
INC hl
DJNZ RdL3

POP hl
POP bc
POP af
RET

```

```

;-----
; Routine : chkmaxli
; Description : check maximum line used
; Register : A,DE,HL <must PUSH value>
;-----

```

```

chkmaxli: PUSH af
          PUSH de
          PUSH hl

          LD hl,(Cur_Addr)
          LD de,(MaxLiUse)
          XOR a
          SBC hl,de

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นสำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JP c,exitmax ; Cur_Addr < MaxLiUse
```

```
LD hl,(Cur_Addr)
```

```
LD (MaxLiUse),hl
```

```
exitmax: POP hl
```

```
POP de
```

```
POP af
```

```
RET
```

```
-----  
; Routine : comp0_9  
; Description : compare second char with 0 - 9  
; Register : A <not PUSH value>  
-----
```

```
comp0_9: CP n00_k
```

```
JP z,zero
```

```
CP n01_k
```

```
JP z,one
```

```
CP n02_k
```

```
JP z,two
```

```
CP n03_k
```

```
JP z,three
```

```
CP n04_k
```

```
JP z,four
```

CP n05\_k

JP z,five

CP n06\_k

JP z,six

CP n07\_k

JP z,seven

CP n08\_k

JP z,eight

CP n09\_k

JP z,nine

JP exitn0\_9

zero: CALL Write0

JP exitn0\_9

one: CALL Write1

JP exitn0\_9

two: CALL Write2

JP exitn0\_9

three: CALL Write3

JP exitn0\_9

four: CALL Write4

JP exitn0\_9

five: CALL Write5

JP exitn0\_9

six: CALL Write6

JP exitn0\_9

seven: CALL Write7

JP exitn0\_9

eight: CALL Write8

JP exitn0\_9

nine: CALL Write9

JP exitn0\_9

exitn0\_9: RET

-----

; Routine : clrbuf4k

; Description : clear buffer key before move to program area

; Register : B,HL

-----

clrbuf4k: LD b,04

LD hl,Buff4Key

clrb: LD (hl),00

INC hl

DJNZ clrb

**RET**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; Routine   : FLine0_3
; Description : calculate position for first charactor line 0 - 3
; Register  : A,DE,HL <must PUSH value>
;-----

```

```
FLine0_3: PUSH af
```

```
    PUSH de
```

```
    PUSH hl
```

```
    XOR a      ;clear carry
```

```
    LD hl,(Cur_addr)
```

```
    LD de,(Top_Block)
```

```
    SBC hl,de
```

```
    SRL l
```

```
    SRL l
```

```
    LD a,l
```

```
    CP Line_0
```

```
    JP z,actF10
```

```
    CP Line_1
```

```
    JP z,actF11
```

```
    CP Line_2
```

```
    JP z,actF12
```

```
    CP Line_3
```

```
    JP z,actF13
```

```
    JP exitF0_3
```

```
actF10: LD a,86h ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitF0_3
```

```
actF11: LD  a,0c6h      ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitF0_3
```

```
actF12: LD  a,96h      ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitF0_3
```

```
actF13: LD  a,0d6h     ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitF0_3
```

```
exitF0_3: POP hl
```

```
POP de
```

```
POP af
```

```
RET
```

```
-----
```

```
; Routine      : SLine0_3
```

```
; Description  : calculate position for second charactor line 0 - 3
```

```
; Register    : A,DE,HL <must PUSH value>
```

```
-----
```

```
SLine0_3: PUSH af
```

```
        PUSH de
```

```
        PUSH hl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
XOR a      ;clear carry
```

```
LD hl,(Cur_addr)
```

```
LD de,(Top_Block)
```

```
SBC hl,de
```

```
SRL l
```

```
SRL l
```

```
LD a,l
```

```
CP Line_0
```

```
JP z,actS10
```

```
CP Line_1
```

```
JP z,actS11
```

```
CP Line_2
```

```
JP z,actS12
```

```
CP Line_3
```

```
JP z,actS13
```

```
JP exitS0_3
```

```
actS10: LD  a,8bh      ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitS0_3
```

```
actS11: LD  a,0cbh     ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitS0_3
```

```
actS12: LD  a,9bh      ;set dd ram
```

```
CALL LCDctrl
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
actS13: LD  a,0dbh      ;set dd ram
        CALL LCDctrl
        JP  exitS0_3
```

```
exitS0_3: POP hl
         POP de
         POP af
```

```
RET
```

```
;-----
; Routine   : TLine0_3
; Description : calculate position for third charactor line 0 - 3
; Register  : A,DE,HL <must PUSH value>
;-----
```

```
TLine0_3: PUSH af
         PUSH de
         PUSH hl
         .
         XOR a      ;clear carry
         LD hl,(Cur_addr)
         LD de,(Top_Block)
         SBC hl,de
         SRL l
         SRL l
         LD a,l
         CP Line_0
         JP z,actT10
```

JP z,actT11

CP Line\_2

JP z,actT12

CP Line\_3

JP z,actT13

JP exitT0\_3

actT10: LD a,8ch ;set dd ram

CALL LCDctrl

JP exitT0\_3

actT11: LD a,0cch ;set dd ram

CALL LCDctrl

JP exitT0\_3

actT12: LD a,9ch ;set dd ram

CALL LCDctrl

JP exitT0\_3

actT13: LD a,0dch ;set dd ram

CALL LCDctrl

JP exitT0\_3

exitT0\_3: POP hl

POP de

POP af

RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; Routine      : FoLin0_3
; Description  : calculate position for third charactor line 0 - 3
; Register    : A,DE,HL <must PUSH value>
;-----

```

```
FoLin0_3: PUSH af
```

```
    PUSH de
```

```
    PUSH hl
```

```
    XOR a      ;clear carry
```

```
    LD hl,(Cur_addr)
```

```
    LD de,(Top_Block)
```

```
    SBC hl,de
```

```
    SRL l
```

```
    SRL l
```

```
    LD a,l
```

```
    CP Line_0
```

```
    JP z,actFol0
```

```
    CP Line_1
```

```
    JP z,actFol1
```

```
    CP Line_2
```

```
    JP z,actFol2
```

```
    CP Line_3
```

```
    JP z,actFol3
```

```
    JP exitFo03
```

```
actFol0: LD a,8dh ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitFo03
```

```
actFol1: LD a,0cdh ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitFo03
```

```
actFol2: LD a,9dh ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitFo03
```

```
actFol3: LD a,0ddh ;set dd ram
```

```
CALL LCDctrl
```

```
JP exitFo03
```

```
exitFo03: POP hl
```

```
POP de
```

```
POP af
```

```
RET
```

```
;
```

```
; Routine : multi
```

```
; Description : multiplier 8 bit
```

```
; Register : A,B,DE,HL <no PUSH value>
```

```
;
```

```
multi: LD b,8
```

```
LD d,0
```

```
LD h,0
```

```
LD l,0
```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RLCA

JP nc,notadd

ADD hl,de

notadd: DJNZ mul

RET

```

;-----
; Routine   : DIV16
; Description : devison for 16 bit
; Register  : BC<divisor>,DE<dividend>,HL<result>
;-----

```

```

DIV16: XOR a      ;clear a=0
        LD h,a    ;clear result=0
        LD l,a    ;
        LD a,16   ;loop counter

div1:  RL e      ;shift left ,store partial result in bit 0
        RL d
        ADC hl,hl ;rotate hl left
                ;if hl greater than bc, subtrac from bc
        SBC hl,bc
        JP nc,div2
        ADD hl,bc ;if negative, restore hl

div2:  CCF      ;partial result in carry flag
        DEC a
        JP nz,div1
        EX de,hl
        ADC hl,hl ;restore last bit of result

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## RET

```

;-----
; Routine   : curdsp
; Description : show cursor
; Register  : A,DE,HL
;-----

```

```
curdsp: PUSH af
```

```
        PUSH de
```

```
        PUSH hl
```

```
        XOR a     ;clear carry
```

```
        LD hl,(Cur_Addr)
```

```
        LD de,(Top_Block)
```

```
        SBC hl,de
```

```
        SRL l
```

```
        SRL l
```

```
        LD a,l
```

```
        CP Line_0
```

```
        JP z,lin0
```

```
        CP Line_1
```

```
        JP z,lin1
```

```
        CP Line_2
```

```
        JP z,lin2
```

```
        CP Line_3
```

```
        JP z,lin3
```

```
        JP exitcur
```

```

lin0: LD  a,8fh      ;set dd ram
      CALL LCDctrl
      LD  hl,cur_on
      CALL nextchr
      JP  exitcur

```

```

lin1: LD  a,0cfh     ;set dd ram
      CALL LCDctrl
      LD  hl,cur_on
      CALL nextchr
      JP  exitcur

```

```

lin2: LD  a,9fh      ;set dd ram
      CALL LCDctrl
      LD  hl,cur_on
      CALL nextchr
      JP  exitcur

```

```

lin3: LD  a,0dfh     ;set dd ram
      CALL LCDctrl
      LD  hl,cur_on
      CALL nextchr
      JP  exitcur

```

```

exitcur: POP hl
        POP de
        POP af

```

```
RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;-----
; Routine   : curoff
; Description : off cursor
; Register  : A,DE,HL
;-----

```

```

curoff: PUSH af

```

```

        PUSH de

```

```

        PUSH hl

```

```

        XOR a      ;clear carry

```

```

        LD hl,(Cur_Addr)

```

```

        LD de,(Top_Block)

```

```

        SBC hl,de

```

```

        SRL l

```

```

        SRL l

```

```

        LD a,l

```

```

        CP Line_0

```

```

        JP z,lin00

```

```

        CP Line_1

```

```

        JP z,lin11

```

```

        CP Line_2

```

```

        JP z,lin22

```

```

        CP Line_3

```

```

        JP z,lin33

```

```

        JP exitcurf

```

```

lin00: LD   a,8fh      ;set dd ram

```

เอกสารนี้เป็นเอกสารการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD hl,cur_off
CALL nextchr
JP exitcurf
```

```
lin11: LD a,0cfh ;set dd ram
CALL LCDctrl
LD hl,cur_off
CALL nextchr
JP exitcurf
```

```
lin22: LD a,9fh ;set dd ram
CALL LCDctrl
LD hl,cur_off
CALL nextchr
JP exitcurf
```

```
lin33: LD a,0dfh ;set dd ram
CALL LCDctrl
LD hl,cur_off
CALL nextchr
JP exitcurf
```

```
exitcurf: POP hl
          POP de
          POP af

          RET
```

เอก; Routine ออกระบบ chkline สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : check for line 0-3

; Register : A,DE,HL

-----

chkline: PUSH af

PUSH de

PUSH hl

XOR a ;clear carry

LD hl,(Line\_Add)

LD de,(Top\_Block)

SBC hl,de

SRL l

SRL l

LD a,l

CP Line\_0

JP z,line0

CP Line\_1

JP z,line1

CP Line\_2

JP z,line2

CP Line\_3

JP z,line3

JP exitchk

line0: LD a,80h ;set dd ram

CALL LCDctrl

CALL calcline

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL PLine_No
```

```
LD hl,colon
```

```
CALL nextchr
```

```
CALL readprog
```

```
XOR a ;clear carry
```

```
LD hl,(Line_Add)
```

```
LD de,0004h
```

```
ADC hl,de
```

```
LD (Line_Add),hl
```

```
JP exitchk
```

```
line1: LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL calcline
```

```
CALL bitode
```

```
CALL PLine_No
```

```
LD hl,colon
```

```
CALL nextchr
```

```
CALL readprog
```

```
XOR a ;clear carry
```

```
LD hl,(Line_Add)
```

```
LD de,0004h
```

```
ADC hl,de
```

```
LD (Line_Add),hl
```

```
JP exitchk
```

เอกสารนี้ line2: LD a,90h สำหรับ ;set dd ram เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL LCDctrl
CALL calcline
CALL bitode
CALL PLine_No
LD hl,colon
CALL nextchr
```

```
CALL readprog
```

```
XOR a ;clear carry
```

```
LD hl,(Line_Add)
```

```
LD de,0004h
```

```
ADC hl,de
```

```
LD (Line_Add),hl
```

```
JP exitchk
```

```
line3: LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL calcline
```

```
CALL bitode
```

```
CALL PLine_No
```

```
LD hl,colon
```

```
CALL nextchr
```

```
CALL readprog
```

```
XOR a ;clear carry
```

```
LD hl,(Line_Add)
```

```
LD de,0004h
```

```
ADC hl,de
```

```
LD (Line_Add),hl
```

JP exitchk

exitchk: POP hl

POP de

POP af

RET

-----

; Routine : lineAct

; Description : check for line 0-3

; Register : A,DE,HL

-----

lineAct: PUSH af

PUSH bc

PUSH de

PUSH hl

XOR h

XOR l

LD hl,(Cur\_Addr)

LD de,TopProg

SBC hl,de

SRL l

SRL l

LD a,h

AND 03h

RLC a

RLC a

RLC a

RLC a

RLC a

RLC a

OR l

SRL h

SRL h

LD (Line\_No),a

LD de,(Line\_No)

LD d,0 ; clear Line\_No+1

LD bc,100

CALL DIV16

LD (result),hl

LD bc,10

CALL DIV16

LD (result+2),hl

LD (result+4),de

POP hl

POP de

POP bc

POP af

RET

-----

; Routine : calcline

; Description : for calculate line number

; Register : A,HL,BC

-----

calcline: PUSH af

PUSH bc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PUSH hl

XOR h

XOR l

LD hl,(Line\_Add)

LD bc,TopProg

SBC hl,bc

SRL l

SRL l

LD a,h

AND 03h

RLC a

RLC a

RLC a

RLC a

RLC a

RLC a

OR l

SRL h

SRL h

LD (Line\_No),a

POP hl

POP bc

POP af

RET

-----

; Routine : clrdisp

; Description : for clear LCD display

เอกสารนี้เป็นทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; Register : A
```

```
-----
```

```
clrdspl: LD a,01h ;clear display
```

```
CALL LCDctrl
```

```
RET
```

```
-----
```

```
; Routine : PLine_No
```

```
; Description : for print line number to LCD display
```

```
; Register : DE<form result>,B<counter>
```

```
-----
```

```
PLine_No: PUSH bc
```

```
PUSH de
```

```
LD de,result
```

```
LD b,03h
```

```
repeat: CALL numtochr
```

```
INC de
```

```
INC de
```

```
DJNZ repeat
```

```
POP de
```

```
POP bc
```

```
RET
```

```
-----
```

```
; Routine : init
```

```
; Description : for initial LCD
```

เอกสารที่ A นี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----
init: LD a,00111000b
```

```
CALL LCDctrl
```

```
LD a,00001100b ;display control d=1,c=1,b=1
```

```
CALL LCDctrl
```

```
LD a,01h ;clear display
```

```
CALL LCDctrl
```

```
LD a,00000110b ;entry mode set
```

```
CALL LCDctrl
```

```
RET
```

```
-----
; Routine : infodsp
```

```
; Description : for send string to LCD testing
```

```
; Register : A,HL
-----
```

```
infodsp: PUSH af
```

```
PUSH hl
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,message1
```

```
CALL nextchr
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,message2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL nextchr
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,message3
```

```
CALL nextchr
```

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,message4
```

```
CALL nextchr
```

```
POP hl
```

```
POP af
```

```
RET
```

```
-----  
; Routine : nextchr  
; Description : for sending string to LCD and ending by 20h  
; Register : A,HL  
-----
```

```
nextchr: LD a,(hl)
```

```
CP 0ffh ; end string
```

```
JP z,exit_loop
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
JP nextchr
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
exit_loop: RET
```

```
-----
```

```
; Routine      : waitbusy
; Description   : waiting for busy flag = 0
; Register     : AF
```

```
-----
```

```
waitbusy: PUSH af
```

```
waitbusy1: IN a,(Rd_Bflg)
```

```
BIT 7,a
```

```
JP nz,waitbusy1
```

```
POP af
```

```
RET
```

```
-----
```

```
; Routine      : jump
; Description   : jump to other line
; Register     : A,B,HL <must PUSH value>
```

```
-----
```

```
jump: PUSH af
```

```
PUSH bc
```

```
PUSH hl
```

```
LD b,3
```

```
LD hl,jumpbcd
```

```
jpp: LD (hl),00
```

```
INC hl
```

DJNZ jpp

CALL putdsp1

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,jumpk

LD b,13

jumpdsp: LD a,(hl)

CALL waitbusy

OUT (Wr\_Data),a

CALL waitbusy

INC hl

DJNZ jumpdsp

LD a,(Kb\_Buf)

CALL keytonum

ro: CALL Jline\_No

CALL getkjump

LD a,(Kb\_Buf)

CP ent\_k

JP z,exitjump

CP clr\_k

JP z,exitjump

JP ro

exitjump: POP hl

POP bc

POP af

RET

```

;-----
; Routine   : getkjump
; Description : for print line number to LCD display
; Register  : A,B<counter>,DE<form result>,HL
;-----

```

```
getkjump: PUSH af
```

```
        PUSH de
```

```
        PUSH hl
```

```
        CALL ReadKey
```

```
        IN a,(PortB) ; check edit mode or run mode
```

```
        BIT 5,a
```

```
        JP z,exitget
```

```
        LD a,(Kb_Buf)
```

```
        CP ent_k
```

```
        JP z,chk_Bot
```

```
        CP clr_k
```

```
        JP z,clrjump
```

```
        CP n00_k
```

```
        JP z,noOk
```

```
        CP n01_k
```

```
        JP z,noOk
```

```
        CP n02_k
```

```
        JP z,noOk
```

CP n03\_k

JP z,noOk

CP n04\_k

JP z,noOk

CP n05\_k

JP z,noOk

CP n06\_k

JP z,noOk

CP n07\_k

JP z,noOk

CP n08\_k

JP z,noOk

CP n09\_k

JP z,noOk

JP exitget

noOk: CALL rojmpbcd

LD a,(Kb\_Buf)

CALL keytonum

JP exitget

clrjump: CALL putdsp1

JP exitget

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
chk_Bot: CALL jmpaddr      ; check > BotProg ?
```

```
LD hl,(Jump_To)  ;
```

```
LD de,BotProg    ;
```

```
XOR a
```

```
SBC hl,de
```

```
JP z,lessBot
```

```
JP nc,jumpErr    ; > BotProg
```

```
lessBot: CALL putdsp1     ; if < BotProg
```

```
CALL curoff
```

```
CALL getdsp1
```

```
LD hl,BotProg    ; check > BotProg-12 (3 line)
```

```
LD de,0ch        ;
```

```
XOR a            ;
```

```
SBC hl,de        ;
```

```
EX de,hl
```

```
LD hl,(Jump_To)
```

```
XOR a
```

```
SBC hl,de
```

```
JP c,topdsp      ; > BotProg-12
```

```
JP z,topdsp
```

```
LD hl,(Jump_To)
```

```
LD (Cur_Addr),hl
```

```
LD (Line_Add),de
```

```
LD (Top_Block),de
```

```
CALL clr dsp
```

```
JP exitget
```

```
LD (Cur_Addr),hl
LD (Line_Add),hl
LD (Top_Block),hl
CALL clr dsp
JP exitget
```

```
jumpErr: LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,JumpErr
```

```
LD b,16
```

```
overdsp: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ overdsp
```

```
CALL delay
```

```
CALL putdsp1
```

```
JP exitget
```

```
exitget: POP hl
```

```
POP de
```

```
POP af
```

```
RET
```

```
;
```

```
; Routine : JLine_No
```

```
; Description : for print line number to LCD display
```

```
; Register : DE<form result>,B<counter>
```

JLine\_No: PUSH af

PUSH bc

PUSH de

LD a,0ddh ;set dd ram

CALL LCDctrl

LD de,jumpbcd

LD b,03h

re: CALL numtochr

INC de

DJNZ re

POP de

POP bc

POP af

RET

-----  
 ; Routine : rojmpbcd |-----|  
 ; Description : rotate data |--jumpbcd<--jumpbcd+1<--jumpbcd+2<--|  
 ; Register : A,DE <must PUSH value>

rojmpbcd: PUSH af

PUSH de

LD a,(jumpbcd)

LD d,a

LD a,(jumpbcd+1)

LD (jumpbcd),a

```
LD a,(jumpbcd+2)
```

```
LD (jumpbcd+1),a
```

```
LD a,d
```

```
LD (jumpbcd+2),a
```

```
POP de
```

```
POP af
```

```
RET
```

```
-----
; Routine      : jmpadrs
; Description  : converse decimal to binary  address in (Jump_To)
; Register    : A,DE,HL <no PUSH value>
```

```
jmpadrs: LD de,(jumpbcd)
```

```
LD a,00
```

```
AND d
```

```
LD d,a
```

```
LD a,100
```

```
CALL multi
```

```
LD (jumpsum),hl
```

```
LD de,(jumpbcd+1)
```

```
LD a,00
```

```
AND d
```

```
LD d,a
```

```
CALL multi
LD (jumpsum+1),hl
```

```
XOR a ;clear carry
LD hl,(jumpsum+1)
LD a,00
AND h
LD h,a
```

```
LD de,(jumpbcd+2)
LD a,00
AND d
LD d,a
ADC hl,de
LD de,(jumpsum)
LD a,00
AND d
LD d,a
XOR a
ADC hl,de
```

```
ADD hl,hl ;
ADD hl,hl ; hl * 4
LD de,TopProg
XOR a
ADC hl,de
LD (Jump_To),hl
```

```

;-----
; Routine   : bitode
; Description : converse binary to decimal
; Register  : A,BC<divisor>,DE<dividend>,HL<result>
;-----

```

```
bitode: PUSH af
```

```
PUSH bc
```

```
PUSH de
```

```
PUSH hl
```

```
LD de,(Line_No)
```

```
LD d,0 ; clear Line_No+1
```

```
LD bc,100
```

```
CALL DIV16
```

```
LD (result),hl
```

```
LD bc,10
```

```
CALL DIV16
```

```
LD (result+2),hl
```

```
LD (result+4),de
```

```
POP hl
```

```
POP de
```

```
POP bc
```

```
POP af
```

```
RET
```

```
;-----
```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : converse number 0-9 to character for LCD display  
 ; Register : A,DE<number 0-9>,HL<numtable address> (not PUSH)

-----

```
numtochr: LD a,(de)
          LD hl,numtable
          ADD a,l
          LD l,a
          LD a,(hl)
          CALL LCDwrite
```

```
RET
```

-----

; Routine : keytonum  
 ; Description : compare third char with 0 - 9  
 ; Register : HL <not PUSH value>

-----

```
keytonum: CP n00_k
          JP z,kzero

          CP n01_k
          JP z,kone

          CP n02_k
          JP z,ktwo

          CP n03_k
          JP z,kthree

          CP n04_k
```

CP n05\_k

JP z,kfive

CP n06\_k

JP z,ksix

CP n07\_k

JP z,kseven

CP n08\_k

JP z,keight

CP n09\_k

JP z,knine

JP exitkn

kzero: LD a,00

LD (jumpbcd+2),a

JP exitkn

kone: LD a,01

LD (jumpbcd+2),a

JP exitkn

ktwo: LD a,02

LD (jumpbcd+2),a

JP exitkn

kthree: LD a,03

LD (jumpbcd+2),a

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitkn

kfour: LD a,04

LD (jumpbcd+2),a

JP exitkn

kfive: LD a,05

LD (jumpbcd+2),a

JP exitkn

ksix: LD a,06

LD (jumpbcd+2),a

JP exitkn

kseven: LD a,07

LD (jumpbcd+2),a

JP exitkn

keight: LD a,08

LD (jumpbcd+2),a

JP exitkn

knine: LD a,09

LD (jumpbcd+2),a

JP exitkn

exitkn: RET



; Description : shift down line display

; Register : A,B,DE,HL <must PUSH value>

-----

downkey: PUSH af

PUSH de

PUSH hl

LD hl,(Cur\_Addr)

LD de,04

ADC hl,de

LD (Cur\_Addr),hl

XOR a ;clear carry

LD hl,(Top\_Block)

LD de,10h ; compare line cursor = top block +10h

ADC hl,de

LD de,(Cur\_Addr)

XOR a

SBC hl,de

JP z,chgpgedn

JP exitdwn

chgpgedn: XOR a

LD hl,(Top\_Block) ; else

LD de,0004h

ADC hl,de

LD (Top\_Block),hl

XOR a ;clear carry

LD hl,(Line\_Add)

LD de,0004h

```
ADC hl,de
```

```
LD (Line_Add),hl
```

```
XOR a
```

```
LD hl,BotProg
```

```
LD de,0004h
```

```
ADC hl,de
```

```
LD de,(Cur_Addr)
```

```
SBC hl,de
```

```
JP z,bottom
```

```
JP' exitdwn
```

```
=====if over than BotProg
```

```
bottom: XOR a
```

```
LD hl,BotProg
```

```
LD (Cur_Addr),hl
```

```
XOR a
```

```
LD hl,(Top_block) ; else
```

```
LD de,0004h
```

```
SBC hl,de
```

```
LD (Top_Block),hl
```

```
LD (Line_Add),hl
```

```
=====display Error "Bottom" or Beep sound
```

```
CALL putdsp1 ; display cleared before must put old display
```

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl, BotErr
```

```
LD b,16
```

```
BotError: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
INC hl
DJNZ BotError
CALL delay
```

```
CALL putdsp1
JP exitdwn
```

```
exitdwn: POP hl
```

```
POP de
```

```
POP af
```

```
RET
```

```
;-----
; Routine   : upkey
; Description : shift up line display
; Register  : A,B,DE,HL <must PUSH value>
```

```
;-----
upkey: PUSH af
```

```
PUSH de
```

```
PUSH hl
```

```
XOR a
```

```
LD hl,(Cur_Addr)
```

```
LD de,04
```

```
SBC hl,de
```

```
LD (Cur_Addr),hl
```

```
XOR a ;clear carry
```

```
LD hl,(Top_Block)
```

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD de,04h ; compare line cursor = top block - 4
```

```
SBC hl,de
```

```
LD de,(Cur_Addr)
```

```
XOR a
```

```
SBC hl,de
```

```
JP z,chgpggeup
```

```
JP exitup
```

```
=====check not less than begin program
```

```
chgpggeup: XOR a
```

```
LD hl,(Top_Block) ; else
```

```
LD de,0004h ;
```

```
SBC hl,de
```

```
LD (Top_Block),hl
```

```
XOR a ;clear carry
```

```
LD hl,(Line_Add)
```

```
LD de,0004h
```

```
SBC hl,de
```

```
LD (Line_Add),hl
```

```
XOR a
```

```
LD hl,TopProg
```

```
LD de,04
```

```
SBC hl,de
```

```
LD de,(Cur_Addr)
```

```
XOR a
```

```
SBC hl,de
```

```
JP z,topover
```

```
JP exitup
```

```

topover: XOR a
          LD hl,TopProg
          LD (Line_Add),hl
          LD (Cur_Addr),hl
          LD (Top_Block),hl
          ;==display Error "Top" or Beep sound
          CALL putdsp1      ; display cleared before must put old display
          LD a,0d0h        ;set dd ram
          CALL LCDctrl
          LD hl, TopErr
          LD b,16
TopError: LD a,(hl)
          CALL waitbusy
          OUT (Wr_Data),a
          CALL waitbusy
          INC hl
          DJNZ TopError
          CALL delay
          CALL putdsp1
          JP exitup

exitup:  POP hl
        POP de
        POP af

        RET

```

```

;-----
; Routine   : readfunc
; Description : read data function key from program area
; Register  : A <not PUSH value>

```

---

```
readfunc: PUSH hl
```

```
LD a,(secondch)
```

```
CP n00_k
```

```
JP z,read3th0
```

```
; CP n01_k
```

```
; JP z,read3th1
```

```
JP exrdfunc
```

```
read3th0: LD a,(thirdchr)
```

```
CP n01_k
```

```
JP z,f_01
```

```
CP n02_k
```

```
JP z,f_02
```

```
CP n03_k
```

```
JP z,f_03
```

```
CP n04_k
```

```
JP z,f_04
```

```
CP n05_k
```

```
JP z,f_05
```

```
CP n06_k
```

```
JP z,f_06
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n07\_k

JP z,f\_07

JP exrdfunc

; read3th1: LD a,(thirdchr)

; CP n01\_k

; JP z,f\_11

;

; CP n02\_k

; JP z,f\_12

;

; JP exrdfunc

f\_01: LD hl,func01

CALL nextchr

JP exrdfunc

f\_02: LD hl,func02

CALL nextchr

JP exrdfunc

f\_03: LD hl,func03

CALL nextchr

JP exrdfunc

f\_04: LD hl,func04

CALL nextchr

JP exrdfunc

f\_05: LD hl,func05

CALL nextchr

JP exrdfunc

f\_06: LD hl,func06

CALL nextchr

JP exrdfunc

f\_07: LD hl,func07

CALL nextchr

JP exrdfunc

exrdfunc: POP hl

RET

;-----  
; Routine : funckey

; Description : test jump to function key

; Register : HL <must PUSH value>

;-----

funckey: PUSH af

PUSH bc

PUSH hl

CALL clrbuf4k

LD a,(Kb\_Buf)

```
LD hl, Buff4Key
LD (hl), a
CALL FLine0_3
LD hl, funcmess
CALL nextchr
```

```
funcchr2: CALL ReadKey ;second charactor
```

```
LD a, (Kb_Buf)
LD hl, Buff4Key+1
LD (hl), a
```

```
CP clr_k
JP z, clrfunc
```

```
CP n00_k
JP z, chrn0
```

```
JP funcchr2
```

```
chrn0: CALL SLine0_3
```

```
CALL Write0
```

```
JP chr3th
```

```
chr3th: CALL ReadKey ;third charactor
```

```
LD a, (Kb_Buf)
LD hl, Buff4Key+2
LD (hl), a
```

```
CP clr_k
JP z, clrfunc
```

CP n01\_k

JP z,f01

CP n02\_k

JP z,f02

CP n03\_k

JP z,f03

CP n04\_k

JP z,f04

CP n05\_k

JP z,f05

CP n06\_k

JP z,f06

CP n07\_k

JP z,f07

JP chr3th

f01: CALL TLine0\_3

CALL Write1

JP funcent

f02: CALL TLine0\_3

CALL Write2

เอกสารนี้เป็นเอกสาร **JP funcent** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f03: CALL TLine0\_3  
CALL Write3  
JP funcent

f04: CALL TLine0\_3  
CALL Write4  
JP funcent

f05: CALL TLine0\_3  
CALL Write5  
JP funcent

f06: CALL TLine0\_3  
CALL Write6  
JP funcent

f07: CALL TLine0\_3  
CALL Write7  
JP funcent

funcent: CALL ReadKey  
LD a,(Kb\_Buf)  
CP clr\_k  
JP z,clrfunc  
  
CP ent\_k  
JP z,entfunc  
JP funcent

```
entfunc: LD b,04
         LD hl,Buff4Key
         LD de,(Cur_Addr)
```

```
Wrfunc: LD a,(hl)
        LD (de),a
        INC hl
        INC de
        DJNZ Wrfunc
```

```
CALL chkmaxli
```

```
LD hl,BotProg
LD de,(Cur_Addr)
XOR a
SBC hl,de
JP z,exitfunc
```

```
CALL dsp10_3
LD a,080h ;set dd ram
CALL LCDctrl
CALL clrdisp ; can't move clrdisp
CALL downkey
JP exitfunc
```

```
clrfunc: CALL putdsp1
        JP exitfunc
```

```
exitfunc: POP hl
         POP bc
         POP af
```

RET

```

;-----
; Routine   : ldkey
; Description : LD key input & write to Program area
; Register  : A,B,HL <must PUSH value>
;-----

```

ldkey: PUSH af

PUSH bc

PUSH hl

CALL clrbuf4k

LD a,(Kb\_Buf)

LD hl,Buff4Key

LD (hl),a

CALL FLine0\_3

LD hl,ldmess

CALL nextchr

waitld2: CALL ReadKey ;second charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+1

LD (hl),a

CP clr\_k

JP z,clrld

CP x\_k

JP z,ldchr2x

CP reg\_k

JP z,ldchr2r

CP pv\_k

JP z,ldchr2pv

CP sv\_k

JP z,ldchr2sv

CP mv\_k

JP z,ldchr2mv

JP waitld2

ldchr2x: CALL SLine0\_3

LD a,"X"

CALL LCDwrite

JP ldchr3x

ldchr2pv: CALL SLine0\_3

LD hl,pvbotton

CALL nextchr

JP ldchr3v

ldchr2sv: CALL SLine0\_3

LD hl,svbotton

CALL nextchr

JP ldchr3v

ldchr2mv: CALL SLine0\_3

LD hl,mvbotton

CALL nextchr

JP lchr3v

ldchr2r: CALL SLine0\_3

LD a,"R"

CALL LCDwrite

ldr0or1: CALL ReadKey

LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrld

CP n00\_k

JP z,Lr0

CP n01\_k

JP z,Lr1

JP ldr0or1

Lr0: CALL TLine0\_3

CALL Write0

ldr01\_09: CALL ReadKey ;fourth charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+3

LD (hl),a

CP clr\_k

JP z,clrld

CP n01\_k

JP z,Lr01

CP n02\_k

JP z,Lr02

CP n03\_k

JP z,Lr03

CP n04\_k

JP z,Lr04

CP n05\_k

JP z,Lr05

CP n06\_k

JP z,Lr06

CP n07\_k

JP z,Lr07

CP n08\_k

JP z,Lr08

CP n09\_k

JP z,Lr09

JP ldr01\_09

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lr01: CALL FoLin0\_3

CALL Write1

JP Ident

Lr02: CALL FoLin0\_3

CALL Write2

JP Ident

Lr03: CALL FoLin0\_3

CALL Write3

JP Ident

Lr04: CALL FoLin0\_3

CALL Write4

JP Ident

Lr05: CALL FoLin0\_3

CALL Write5

JP Ident

Lr06: CALL FoLin0\_3

CALL Write6

JP Ident

Lr07: CALL FoLin0\_3

CALL Write7

JP Ident

Lr08: CALL FoLin0\_3

CALL Write8

JP Ident

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lr09: CALL FoLin0\_3

CALL Write9

JP Ident

Lr1: CALL TLine0\_3

CALL Write1

ldr10\_16: CALL ReadKey ;fourth charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+3

LD (hl),a

CP clr\_k

JP z,clrld

CP n00\_k

JP z,Lr10

CP n01\_k

JP z,Lr11

CP n02\_k

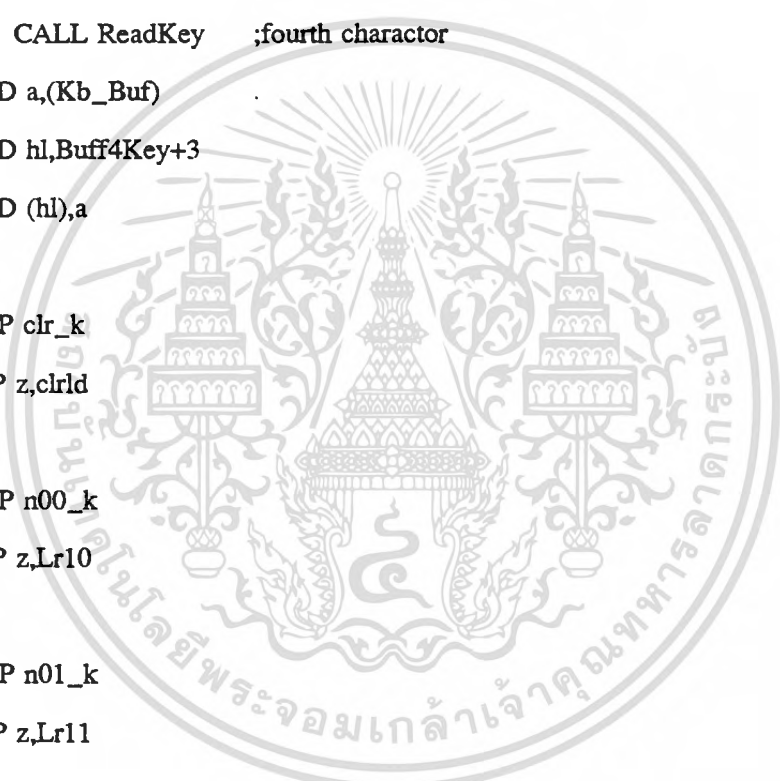
JP z,Lr12

CP n03\_k

JP z,Lr13

CP n04\_k

JP z,Lr14



CP n05\_k

JP z,Lr15

CP n06\_k

JP z,Lr16

JP ldr10\_16

Lr10: CALL FoLin0\_3

CALL Write0

JP Ident

Lr11: CALL FoLin0\_3

CALL Write1

JP Ident

Lr12: CALL FoLin0\_3

CALL Write2

JP Ident

Lr13: CALL FoLin0\_3

CALL Write3

JP Ident

Lr14: CALL FoLin0\_3

CALL Write4

JP Ident

Lr15: CALL FoLin0\_3

CALL Write5

JP Ident

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันวิจัยและพัฒนาเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่สามารถนำเอกสารไปใช้โดยไม่ได้รับอนุญาตจากสถาบันฯ  
 ไม่สามารถนำเอกสารไปใช้เพื่อการค้า  
 ไม่สามารถนำเอกสารไปใช้เพื่อการศึกษา  
 ไม่สามารถนำเอกสารไปใช้เพื่อเผยแพร่  
 ไม่สามารถนำเอกสารไปใช้เพื่อประชาสัมพันธ์  
 ไม่สามารถนำเอกสารไปใช้เพื่อวัตถุประสงค์อื่นใด  
 ไม่สามารถนำเอกสารไปใช้เพื่อวัตถุประสงค์อื่นใด  
 ไม่สามารถนำเอกสารไปใช้เพื่อวัตถุประสงค์อื่นใด

Lr16: CALL FoLin0\_3

CALL Write6

JP Ident

ldchr3x: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl, Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrld

CP n01\_k

JP z,Lon1x

CP n02\_k

JP z,Ltw2x

CP n03\_k

JP z,Lth3x

CP n04\_k

JP z,Lfo4x

CP n05\_k

JP z,Lfi5x

CP n06\_k

JP z,Lsi6x

CP n07\_k

JP z,Lse7x

CP n08\_k

JP z,Lei8x

JP ldchr3x

Lon1x: CALL TLine0\_3

CALL Write1

JP lident

Ltw2x: CALL TLine0\_3

CALL Write2

JP lident

Lth3x: CALL TLine0\_3

CALL Write3

JP lident

Lfo4x: CALL TLine0\_3

CALL Write4

JP lident

Lfi5x: CALL TLine0\_3

CALL Write5

JP lident

Lsi6x: CALL TLine0\_3

CALL Write6

JP lident

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lse7x: CALL TLine0\_3

CALL Write7

JP Ident

Lei8x: CALL TLine0\_3

CALL Write8

JP Ident

ldchr3v: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl, Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrld

CP n01\_k

JP z,Lon1

CP n02\_k

JP z,Ltw2

CP n03\_k

JP z,Lth3

CP n04\_k

JP z,Lfo4

JP z,Lfi5

CP n06\_k

JP z,Lsi6

CP n07\_k

JP z,Lse7

CP n08\_k

JP z,Lei8

JP ldchr3v

Lon1: CALL FoLin0\_3

CALL Writel

JP lident

Ltw2: CALL FoLin0\_3

CALL Write2

JP lident

Lth3: CALL FoLin0\_3

CALL Write3

JP lident

Lfo4: CALL FoLin0\_3

CALL Write4

JP lident

Lfi5: CALL FoLin0\_3

CALL Write5

JP lident

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นโดยพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่สามารถนำเอกสารไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Lsi6: CALL FoLin0\_3  
 CALL Write6  
 JP Ident

Lse7: CALL FoLin0\_3  
 CALL Write7  
 JP Ident

Lei8: CALL FoLin0\_3  
 CALL Write8  
 JP Ident

Ident: CALL ReadKey  
 LD a,(Kb\_Buf)  
 CP clr\_k  
 JP z,clrld  
  
 CP ent\_k  
 JP z,entld  
 JP Ident

entld: LD b,04  
 LD hl,Buff4Key  
 LD de,(Cur\_Addr)

WrProglD: LD a,(hl)  
 LD (de),a  
 INC hl  
 INC de

DJNZ WrProglD

```
CALL chkmaxli
```

```
LD hl,BotProg
```

```
LD de,(Cur_Addr)
```

```
XOR a
```

```
SBC hl,de
```

```
JP z,exitld
```

```
CALL dsp10_3
```

```
LD a,080h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL clrdisp ; can't move clrdisp
```

```
CALL downkey
```

```
JP exitld
```

```
clrld: CALL putdsp1
```

```
JP exitld
```

```
exitld: POP hl
```

```
POP bc
```

```
POP af
```

```
RET
```

```
#####
```

```
ORG 0A000H
```

```
#####
```

```
;
```

```
; Routine : outkey
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : OUT key input & write to Program area

; Register : A,B,HL <must PUSH value>

-----

outkey: PUSH af

PUSH bc

PUSH hl

CALL clrbuf4k

LD a,(Kb\_Buf)

LD hl,Buff4Key

LD (hl),a

CALL FLine0\_3

LD hl,outmess

CALL nextchr

waitout2: CALL ReadKey ;second charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+1

LD (hl),a

CP clr\_k

JP z,clrout

CP y\_k

JP z,outchr2y

CP reg\_k

JP z,outchr2r

CP pv\_k

JP z,outchr2pv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP sv\_k

JP z,outch2sv

CP mv\_k

JP z,outch2mv

JP waitout2

outchr2y: CALL SLine0\_3

LD a,"Y"

CALL LCDwrite

JP outchr3y

outch2pv: CALL SLine0\_3

LD hl,pvbotton

CALL nextchr

JP outchr3v

outch2sv: CALL SLine0\_3

LD hl,svbotton

CALL nextchr

JP outchr3v

outch2mv: CALL SLine0\_3

LD hl,mvbotton

CALL nextchr

JP outchr3v

outchr2r: CALL SLine0\_3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD a,"R"

CALL LCDwrite

outr0or1: CALL ReadKey

LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrout

CP n00\_k

JP z,Outr0

CP n01\_k

JP z,Outr1

JP outr0or1

Outr0: CALL TLine0\_3

CALL Write0

outr0109: CALL ReadKey ;fourth charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+3

LD (hl),a

CP clr\_k

JP z,clrout

CP n01\_k

JP z,Outr01

CP n02\_k

JP z,Outr02

CP n03\_k

JP z,Outr03

CP n04\_k

JP z,Outr04

CP n05\_k

JP z,Outr05

CP n06\_k

JP z,Outr06

CP n07\_k

JP z,Outr07

CP n08\_k

JP z,Outr08

CP n09\_k

JP z,Outr09

JP outr0109

Outr01: CALL FoLin0\_3

CALL Write1

JP outent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Outr02: CALL FoLin0\_3  
 CALL Write2  
 JP outent

Outr03: CALL FoLin0\_3  
 CALL Write3  
 JP outent

Outr04: CALL FoLin0\_3  
 CALL Write4  
 JP outent

Outr05: CALL FoLin0\_3  
 CALL Write5  
 JP outent

Outr06: CALL FoLin0\_3  
 CALL Write6  
 JP outent

Outr07: CALL FoLin0\_3  
 CALL Write7  
 JP outent

Outr08: CALL FoLin0\_3  
 CALL Write8  
 JP outent

Outr09: CALL FoLin0\_3  
 CALL Write9

JP outent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Outr1: CALL TLine0\_3

CALL Write1

outr1016: CALL ReadKey ;fourth charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+3

LD (hl),a

CP clr\_k

JP z,clrout

CP n00\_k

JP z,Outr10

CP n01\_k

JP z,Outr11

CP n02\_k

JP z,Outr12

CP n03\_k

JP z,Outr13

CP n04\_k

JP z,Outr14

CP n05\_k

JP z,Outr15

CP n06\_k

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP z,Outr16

JP outr1016

Outr10: CALL FoLin0\_3

CALL Write0

JP outent

Outr11: CALL FoLin0\_3

CALL Write1

JP outent

Outr12: CALL FoLin0\_3

CALL Write2

JP outent

Outr13: CALL FoLin0\_3

CALL Write3

JP outent

Outr14: CALL FoLin0\_3

CALL Write4

JP outent

Outr15: CALL FoLin0\_3

CALL Write5

JP outent

Outr16: CALL FoLin0\_3

CALL Write6

JP outent

outchr3y: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl, Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrout

CP n01\_k

JP z,Oonly

CP n02\_k

JP z,Otw2y

CP n03\_k

JP z,Oth3y

CP n04\_k

JP z,Of04y

CP n05\_k

JP z,Ofi5y

CP n06\_k

JP z,Osi6y

CP n07\_k

JP z,Ose7y

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n08\_k

JP z,Oei8y

JP outchr3y

Oonly: CALL TLine0\_3

CALL Write1

JP outent

Otw2y: CALL TLine0\_3

CALL Write2

JP outent

Oth3y: CALL TLine0\_3

CALL Write3

JP outent

Ofo4y: CALL TLine0\_3

CALL Write4

JP outent

Ofi5y: CALL TLine0\_3

CALL Write5

JP outent

Osi6y: CALL TLine0\_3

CALL Write6

JP outent

Ose7y: CALL TLine0\_3

CALL Write7

JP outent

Oei8y: CALL TLine0\_3

CALL Write8

JP outent

outchr3v: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrout

CP n01\_k

JP z,Oon1

CP n02\_k

JP z,Otw2

CP n03\_k

JP z,Oth3

CP n04\_k

JP z,Ofo4

CP n05\_k

JP z,Ofi5

CP n06\_k

JP z,Osi6

CP n07\_k

JP z,Ose7

CP n08\_k

JP z,Oei8

JP outchr3v

Oon1: CALL FoLin0\_3

CALL Write1

JP outent

Otw2: CALL FoLin0\_3

CALL Write2

JP outent

Oth3: CALL FoLin0\_3

CALL Write3

JP outent

Ofo4: CALL FoLin0\_3

CALL Write4

JP outent

Ofi5: CALL FoLin0\_3

CALL Write5

JP outent

Osi6: CALL FoLin0\_3

CALL Write6

JP outent

Ose7: CALL FoLin0\_3

CALL Write7

JP outent

Oei8: CALL FoLin0\_3

CALL Write8

JP outent

outent: CALL ReadKey

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrout

CP ent\_k

JP z,entout

JP outent

entout: LD b,04

LD hl,Buff4Key

LD de,(Cur\_Addr)

WrProgout: LD a,(hl)

LD (de),a

INC hl

INC de

DJNZ WrProgout

CALL chkmaxli

LD hl,BotProg

เอกสารนี้เป็นเอกสาร LD de,(Cur\_Addr) ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

XOR a
SBC hl,de
JP z,exitout

```

```

CALL dsp10_3
LD a,080h ;set dd ram
CALL LCDctrl
CALL clrdisp ; can't move clrdisp
CALL downkey
JP exitout

```

```

clrout: CALL putdsp1
JP exitout
exitout: POP hl
POP bc
POP af
RET

```

```

;-----
; Routine : bsckey
; Description : bsc key input & write to Program area
; Register : A,B,HL <must PUSH value>
;-----

```

```

bsckey: PUSH af
PUSH bc
PUSH hl
CALL clrbuf4k

```

เอกสารนี้เป็นเอกสาร LD a,(Kb\_Buf)การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD hl, Buff4Key
LD (hl), a
CALL FLine0_3
LD hl, bscmess
CALL nextchr

```

```
waitzero: CALL ReadKey ;second charactor
```

```

LD a, (Kb_Buf)
LD hl, Buff4Key+1
LD (hl), a

```

```

CP clr_k
JP z, clrbsc

CP n00_k
JP z, zeroOK
JP waitzero

```

```

zeroOK: CALL SLine0_3
CALL Write0

```

```
waitnum: CALL ReadKey ;third charactor
```

```

LD a, (Kb_Buf)
LD hl, Buff4Key+2
LD (hl), a

```

```

CP clr_k
JP z, clrbsc

```

```
CP n01_k
```

เอกสารนี้เป็นเอกสาร **JP z, on1** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n02\_k

JP z,tw2

CP n03\_k

JP z,th3

CP n04\_k

JP z,fo4

CP n05\_k

JP z,fi5

CP n06\_k

JP z,si6

CP n07\_k

JP z,se7

CP n08\_k

JP z,ei8

JP waitnum

on1: CALL TLine0\_3

CALL Write1

JP waitent

tw2: CALL TLine0\_3

CALL Write2

JP waitent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

th3: CALL TLine0\_3

CALL Write3

JP waitent

fo4: CALL TLine0\_3

CALL Write4

JP waitent

fi5: CALL TLine0\_3

CALL Write5

JP waitent

si6: CALL TLine0\_3

CALL Write6

JP waitent

se7: CALL TLine0\_3

CALL Write7

JP waitent

ei8: CALL TLine0\_3

CALL Write8

JP waitent

waitent: CALL ReadKey

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrbsc

CP ent\_k

JP z,entbsc

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP waitent

entbsc: LD b,04

LD hl, Buff4Key

LD de, (Cur\_Addr)

WrProg: LD a, (hl)

LD (de), a

INC hl

INC de

DJNZ WrProg

CALL chkmaxli

LD hl, BotProg

LD de, (Cur\_Addr)

XOR a

SBC hl, de

JP z, exitbsc

CALL dsp10\_3

LD a, 080h ;set dd ram

CALL LCDctrl

CALL clrdspl ; can't move clrdspl

CALL downkey

JP exitbsc

clrbsc: CALL putdsp1

JP exitbsc

exitbsc: POP hl

POP bc

POP af

RET

-----  
 ; Routine : delkey  
 ; Description : delete line at cursor address pointing  
 ; Register : A,BC,HL <must PUSH value>  
 -----

delkey: PUSH af  
 PUSH bc  
 PUSH hl  
 CALL putdsp1

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,delk

LD b,13

deldsp: LD a,(hl)

CALL waitbusy

OUT (Wr\_Data),a

CALL waitbusy

INC hl

DJNZ deldsp

CALL lineAct

CALL Pline\_No

waitdOK: CALL ReadKey

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นโดยศูนย์เทคโนโลยีสารสนเทศและการสื่อสารเพื่อใช้ในการศึกษาวิจัยและพัฒนาเทคโนโลยีสารสนเทศและการสื่อสาร  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT 5,a

JP z,exitdel

LD a,(Kb\_Buf)

CP ent\_k

JP z,delOK

CP clr\_k

JP z,clrdel

JP waitdOK

; if MaxLiUse <> Cur\_Addr

delOK: XOR a

LD hl, (MaxLiUse)

LD de, (Cur\_Addr)

SBC hl,de

JP z,equalMax

JP c,exitdel

LD b,h

LD c,l

XOR a

LD hl,(Cur\_Addr)

LD de,04

ADC hl,de ; line Cur + 4

LD de,(Cur\_Addr)

ldir

```

LD hl,(MaxLiUse)
delclr: LD (hl),0000h
INC hl
DJNZ delclr
; if MaxLiUse < TopProg
XOR a
LD hl, (MaxLiUse)
LD de, TopProg
SBC hl,de
JP z,exitdel
maxloop: XOR a ; decrement Max. line used
LD hl,(MaxLiUse)
LD de,04
SBC hl,de
LD (MaxLiUse),hl
LD a,(hl)
CP 00
JP nz,exitdel
JP maxloop

clrdel: CALL putdsp1
JP xdel

exitdel: CALL clrdisp
CALL dsp10_3

xdel: POP hl
POP bc
POP af

```

RET

```

;-----
; Routine   : inskey
; Description : test jump to inskey
; Register  : A,BC,HL <must PUSH value>
;-----

```

```

inskey: PUSH af
        PUSH bc
        PUSH hl
        CALL putdsp1

```

```

        LD a,0d0h ;set dd ram

```

```

        CALL LCDctrl

```

```

        LD hl,inskey

```

```

        LD b,13

```

```

insdsp: LD a,(hl)

```

```

        CALL waitbusy

```

```

        OUT (Wr_Data),a

```

```

        CALL waitbusy

```

```

        INC hl

```

```

        DJNZ insdsp

```

```

        CALL lineAct

```

```

        CALL Pline_No

```

```

waitiOK: CALL ReadKey

```

```

        IN a,(PortB) ; check edit mode or run mode

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT 5,a

JP z,exitins

LD a,(Kb\_Buf)

CP ent\_k

JP z,insOK

CP clr\_k

JP z,clrins

JP waitiOK

; if MaxLiUse < BotProg

insOK: XOR a

LD hl, BotProg

LD de, (MaxLiUse)

SBC hl,de

JP z,equalBot

JP c,exitins ; BotProg not less than MaxLiUse check by adding record.

XOR a

LD hl, (MaxLiUse)

LD de, (Cur\_Addr)

SBC hl,de

JP c,exitins

XOR a

LD hl, (MaxLiUse)

LD de,04

ADC hl,de

XOR a

LD de, (Cur\_Addr)

SBC hl,de

LD b,h

LD c,l

XOR a

LD hl,(MaxLiUse)

LD de,07

ADC hl,de ; MaxLiUse + 7 target

EX de,hl

PUSH de

XOR a

LD hl,(MaxLiUse)

LD de,03

ADC hl,de ; source

POP de

lddr

LD b,04

LD hl,(Cur\_Addr)

insclr: LD (hl),0000h

INC hl

DJNZ insclr

XOR a ; increment Max. line used

LD hl,(MaxLiUse)

LD de,04

ADC hl,de

LD (MaxLiUse),hl

```
equalBot: LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl, insErr
```

```
LD b,16
```

```
insError: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ insError
```

```
CALL delay
```

```
CALL putdsp1
```

```
JP exitins
```

```
clrins: CALL putdsp1
```

```
JP xins
```

```
exitins: CALL clrdsp
```

```
CALL dsp10_3
```

```
xins: POP hl
```

```
POP bc
```

```
POP af
```

```
RET
```

```
-----
```

```
; Routine : srchkey
```

```
; Description : test jump to srchkey
```

เอก; Register : A,B,DE,HL <must PUSH value> ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

---

srchkey: PUSH af

PUSH bc

PUSH de

PUSH hl

CALL putdsp1

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,srchk

LD b,16

srchdsp: LD a,(hl)

CALL waitbusy

OUT (Wr\_Data),a

CALL waitbusy

INC hl

DJNZ srchdsp

LD a,0d8h ;set dd ram

CALL LCDctrl

CALL clrbuf4k

srchchr1: CALL readkey ; first charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key

LD (hl),a

CP ld\_k

JP z,srchld

CP bsc\_k

JP z,srchbsc

CP out\_k

JP z,srchout

CP fun\_k

JP z,srchfunc

CP clr\_k

JP z,clrsrch

JP srchchr1

srchld: CALL srch\_Id

JP exitsrch

srchout: CALL srch\_out

JP exitsrch

srchbsc: CALL srch\_bsc

JP exitsrch

srchfunc: CALL srch\_fun

JP exitsrch

clrsrch: CALL putdsp1

JP exitsrch

exitsrch: POP hl

POP de

POP bc

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POP af

RET

```

;-----
; Routine      : srch_ld
; Description  : search LD
; Register    : <not PUSH value>
;-----

```

```

srch_ld:  LD hl,ldmess
          CALL nextchr
          LD  a,0ddh ;set dd ram
          CALL LCDctrl

sldchr2:  CALL ReadKey ;second charactor
          LD  a,(Kb_Buf)
          LD  hl,Buff4Key+1
          LD  (hl),a

          CP  clr_k
          JP  z,clrs_ld

          CP  x_k
          JP  z,sldch2x

          CP  pv_k
          JP  z,sldch2pv

          CP  mv_k

```

CP sv\_k

JP z,sldch2sv

CP reg\_k

JP z,sldch2r

JP sldchr2

sldch2x: LD a,"X"

CALL LCDwrite

LD a,0deh ;set dd ram

CALL LCDctrl

JP sld3num

sldch2r: LD a,"R"

CALL LCDwrite

LD a,0deh ;set dd ram

CALL LCDctrl

JP sldr0or1

sldch2pv: LD hl,pvbotton

CALL nextchr

LD a,0dfh ;set dd ram

CALL LCDctrl

JP sld3num

sldch2mv: LD hl,mvbotton

CALL nextchr

LD a,0dfh ;set dd ram

CALL LCDctrl

JP sld3num

sldch2sv: LD hl,svbotton

CALL nextchr

LD a,0dfh ;set dd ram

CALL LCDctrl

JP sld3num

sldr0or1: CALL ReadKey ;third charactor LD r

- LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrs\_ld

CP n00\_k

JP z,sLdr0

CP n01\_k

JP z,sLdr1

JP sldr0or1

sLdr0: CALL Write0

LD a,0dfh ;set dd ram

CALL LCDctrl

JP sldr0109

```
LD a,0dfh ;set dd ram
```

```
CALL LCDctrl
```

```
JP sldr1016
```

```
sldr0109: CALL ReadKey ;fourth charactor LD r0
```

```
LD a,(Kb_Buf)
```

```
LD hl,Buff4Key+3
```

```
LD (hl),a
```

```
CP clr_k
```

```
JP z,clrs_ld
```

```
CP n01_k
```

```
JP z,sLdr01
```

```
CP n02_k
```

```
JP z,sLdr02
```

```
CP n03_k
```

```
JP z,sLdr03
```

```
CP n04_k
```

```
JP z,sLdr04
```

```
CP n05_k
```

```
JP z,sLdr05
```

```
CP n06_k
```

```
JP z,sLdr06
```

```
CP n07_k
```

JP z,sLdr07

CP n08\_k

JP z,sLdr08

CP n09\_k

JP z,sLdr09

JP sldr0109

sLdr01: CALL Write1

JP srcldent

sLdr02: CALL Write2

JP srcldent

sLdr03: CALL Write3

JP srcldent

sLdr04: CALL Write4

JP srcldent

sLdr05: CALL Write5

JP srcldent

sLdr06: CALL Write6

JP srcldent

sLdr07: CALL Write7

JP srcldent

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sLdr08: CALL Write8

JP srcldent

sLdr09: CALL Write9

JP srcldent

sldr1016: CALL ReadKey ;fourth charactor LD r1

LD a,(Kb\_Buf)

LD hl,Buf4Key+3

LD (hl),a

CP clr\_k

JP z,clrs\_ld

CP n00\_k

JP z,sLdr10

CP n01\_k

JP z,sLdr11

CP n02\_k

JP z,sLdr12

CP n03\_k

JP z,sLdr13

CP n04\_k

JP z,sLdr14

CP n05\_k

JP z,sLdr15

CP n06\_k

JP z,sLdr16

JP sldr1016

sLdr10: CALL Write0

JP srcldent

sLdr11: CALL Write1

JP srcldent

sLdr12: CALL Write2

JP srcldent

sLdr13: CALL Write3

JP srcldent

sLdr14: CALL Write4

JP srcldent

sLdr15: CALL Write5

JP srcldent

sLdr16: CALL Write6

JP srcldent

sld3num: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl, Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrs\_ld

CP n01\_k

JP z,sLon1

CP n02\_k

JP z,sLtw2

CP n03\_k

JP z,sLth3

CP n04\_k

JP z,sLfo4

CP n05\_k

JP z,sLfi5

CP n06\_k

JP z,sLsi6

CP n07\_k

JP z,sLse7

CP n08\_k

JP z,sLei8

JP sld3num

JP srcldent

sLtw2: CALL Write2

JP srcldent

sLth3: CALL Write3

JP srcldent

sLfo4: CALL Write4

JP srcldent.

sLfi5: CALL Write5

JP srcldent

sLsi6: CALL Write6

JP srcldent

sLse7: CALL Write7

JP srcldent

sLei8: CALL Write8

JP srcldent

srcldent: CALL ReadKey

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrs\_ld

CP ent\_k

JP z,entsrcld

JP srcldent

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
entsrcld: CALL entsrch
          JP ExsrcLd
```

```
clrs_ld:  CALL putdsp1
          JP ExsrcLd
```

```
ExsrcLd: RET
```

```
;-----
; Routine   : srch_bsc
; Description : search BSC
; Register  : <not PUSH value>
;-----
```

```
srch_bsc: LD hl,bscmess
          CALL nextchr
          LD  a,0ddh      ;set dd ram
          CALL LCDctrl

sbscchr2: CALL ReadKey   ;second charactor
          LD a,(Kb_Buf)
          LD hl,Buff4Key+1
          LD (hl),a

          CP clr_k
          JP z,clrs_bsc

          CP n00_k
          JP z,sbsc0
```

JP sbscchr2

sbsc0: CALL Write0

LD a,0deh ;set dd ram

CALL LCDctrl

JP sbsc3num

sbsc3num: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrs\_bsc

CP n01\_k

JP z,sBsc01

CP n02\_k

JP z,sBsc02

CP n03\_k

JP z,sBsc03

CP n04\_k

JP z,sBsc04

CP n05\_k

JP z,sBsc05

CP n06\_k

JP z,sBsc06

CP n07\_k

JP z,sBsc07

CP n08\_k

JP z,sBsc08

JP sbsc3num

sBsc01: CALL Write1

JP sBscEnt

sBsc02: CALL Write2

JP sBscEnt

sBsc03: CALL Write3

JP sBscEnt

sBsc04: CALL Write4

JP sBscEnt

sBsc05: CALL Write5

JP sBscEnt

sBsc06: CALL Write6

JP sBscEnt

sBsc07: CALL Write7

JP sBscEnt

sBsc08: CALL Write8

JP sBscEnt

JP sBscEnt

sBscEnt: CALL ReadKey

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrs\_bsc

CP ent\_k

JP z,EntsBsc

JP sBscEnt

EntsBsc: CALL entsrch

JP ExsrcBsc

clrs\_bsc: CALL putdsp1

JP ExsrcBsc

ExsrcBsc: RET

-----

; Routine : Mulvar

; Description : Change data from Keyboard change Buffer to Binary

; Register : CA<Multiplier>,DE<Multiplicand>,HI<Result>

-----

Mulvar: LD de,(KchgBuf)

LD d,0

LD a,03H ; set multiplier = 1000

LD c,a ;

LD a,0e8H

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรรมใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL Mul16

LD (result),hl

XOR a

LD de,(KchgBuf+1)

LD d,0

LD a,00H ; set multiplier = 100

LD c,a

LD a,100

CALL Mul16

LD de,(result)

XOR a

ADC hl,de

LD (result),hl

XOR a

LD de,(KchgBuf+2)

LD d,0

LD a,00H ; set multiplier = 10

LD c,a

LD a,10

CALL Mul16

LD de,(result)

XOR a

ADC hl,de

LD de,(KchgBuf+3) ; set multiplier = 1

LD d,0

XOR a

ADC hl,de

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

-----  
 ; Routine : kGtoKch4  
 ; Description : Display data from GainBcd and move to KchgBuf  
 ; Register : DE <not PUSH value>

kGtoKch4: CALL Kbcd4

LD de,GainBcd  
 CALL numtochr  
 LD de,GainBcd+2  
 CALL numtochr  
 LD de,GainBcd+4  
 CALL numtochr  
 LD de,GainBcd+6  
 CALL numtochr  
 LD a,(GainBcd)  
 LD (KchgBuf),a  
 LD a,(GainBcd+2)  
 LD (KchgBuf+1),a  
 LD a,(GainBcd+4)  
 LD (KchgBuf+2),a  
 LD a,(GainBcd+6)  
 LD (KchgBuf+3),a

RET

-----  
 ; Routine : Kbcd4

; Description : converse binary to decimal

; Register : AF,BC<divisor>,DE<dividend>,HL<result> ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

---

Kbcd4: PUSH af

PUSH bc

PUSH hl

LD bc,1000

CALL DIV16

LD (GainBcd),hl

LD bc,100

CALL DIV16

LD (GainBcd+2),hl

LD bc,10

CALL DIV16

LD (GainBcd+4),hl

LD (GainBcd+6),de

POP hl

POP bc

POP af

RET

---

; Routine : GtoNum

; Description : change Gainbcd to number display

; Register :

เอกสารนี้เป็นเอกสารที่เผยแพร่โดยศูนย์บริการวิชาการที่หอศิลปวัฒนธรรมแห่งกรุงเทพมหานคร ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GtoNum: CALL Kbcd4
        LD de,GainBcd
        CALL numtochr
        LD de,GainBcd+2
        CALL numtochr
        LD de,GainBcd+4
        CALL numtochr
        LD de,GainBcd+6
        CALL numtochr

```

```
RET
```

```

;-----
; Routine   : clear
; Description : clear data
; Register  :
;-----

```

```
clear: CALL readkey
        LD a,(Kb_Buf)

```

```
CP set_k
JP z,clr_prog

```

```
CP clr_k
JP z,clrclr
JP clear

```

```
clr_prog: LD a,0d0h ;set dd ram

```

```
CALL LCDctrl
LD hl,clrprg

```

```
LD b,16
```

```

clrpgdsp: LD a,(hl)
          CALL waitbusy
          OUT (Wr_Data),a
          CALL waitbusy
          INC hl
          DJNZ clrpgdsp

```

```

clrconf: CALL readkey
          LD a,(Kb_Buf)

```

```

          CP ent_k
          JP z,progOK

```

```

          CP clr_k
          JP z,clrclr
          JP clrconf

```

```

clrclr:  CALL putdsp1
          JP exitclr

```

```

progOK:  CALL clrProg
          CALL clrdsp
          JP exitclr

```

```

exitclr: RET

```

-----

```
; Routine : LCDctrl
```

```
; Description : Control LCD
```

```
; Register :
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LCDctrl: CALL waitbusy
        OUT (LCD_Ctrl),a
        CALL waitbusy

```

```

RET

```

```

;-----
; Routine   : LCDwrite
; Description : Write data to LCD
; Register  :
;-----

```

```

LCDwrite: CALL waitbusy
        OUT (Wr_Data),a
        CALL waitbusy
        RET

```

```

;-----
; Routine   : Write0
; Description : Write data to LCD for display "0"
; Register  :
;-----

```

```

Write0: LD a,"0"
        CALL LCDwrite
        RET

```

```

; Routine   : Write1
; Description : Write data to LCD for display "1"
; Register  :

```

```

;-----

```

```

Write1: LD a,"1"
        CALL LCDwrite

```

```

RET

```

```

; Routine   : Write2
; Description : Write data to LCD for display "2"
; Register  :

```

```

;-----

```

```

Write2: LD a,"2"
        CALL LCDwrite

```

```

RET

```

```

; Routine   : Write3
; Description : Write data to LCD for display "3"
; Register  :

```

```

;-----

```

```

Write3: LD a,"3"
        CALL LCDwrite

```

---

```

; Routine   : Write4
; Description : Write data to LCD for display "4"
; Register  :

```

---

```
Write4: LD a,"4"
```

```
CALL LCDwrite
```

```
RET
```

---

```

; Routine   : Write5
; Description : Write data to LCD for display "5"
; Register  :

```

---

```
Write5: LD a,"5"
```

```
CALL LCDwrite
```

```
RET
```

---

```

; Routine   : Write6
; Description : Write data to LCD for display "6"
; Register  :

```

---

```
Write6: LD a,"6"
```

เอกสารนี้เป็นเอกสาร CALL LCDwrite ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

-----  
 ; Routine : Write7  
 ; Description : Write data to LCD for display "7"  
 ; Register :  
 -----

Write7: LD a,"7"

CALL LCDwrite

RET

-----  
 ; Routine : Write8  
 ; Description : Write data to LCD for display "8"  
 ; Register :  
 -----

Write8: LD a,"8"

CALL LCDwrite

RET

-----  
 ; Routine : Write9  
 ; Description : Write data to LCD for display "9"  
 ; Register :  
 -----

```

Write9: LD a,"9"
        CALL LCDwrite

        RET

```

```

;-----
; Routine   : srch_out
; Description : search OUT
; Register  : <not PUSH value>
;-----

```

```

srch_out: LD hl,outmess
          CALL nextchr
          LD  a,0ddh ;set dd ram
          CALL LCDctrl

soutchr2: CALL ReadKey ;second charactor
          LD  a,(Kb_Buf)
          LD  hl,Buff4Key+1
          LD  (hl),a

          CP  clr_k
          JP  z,clrs_out

          CP  y_k
          JP  z,sout2y

          CP  pv_k
          JP  z,sout2pv

```

```
JP z,sout2mv
```

```
CP sv_k
```

```
JP z,sout2sv
```

```
CP reg_k
```

```
JP z,sout2r
```

```
JP soutchr2
```

```
sout2y: LD a,"Y"
        CALL LCDwrite
        LD a,0deh ;set dd ram
        CALL LCDctrl
        JP sout3num

sout2r: LD a,"R"
        CALL LCDwrite
        LD a,0deh ;set dd ram
        CALL LCDctrl
        JP sout0or1
```

```
sout2pv: CALL waitbusy
         LD hl,pvbotton
         CALL nextchr
         LD a,0dfh ;set dd ram
         CALL LCDctrl
         JP sout3num
```

```
sout2mv: CALL waitbusy
```

```
LD hl,mvbotton
```

```

CALL nextchr
LD  a,0dfh    ;set dd ram
CALL LCDctrl
JP  sout3num

```

```

sout2sv: CALL waitbusy
LD  hl,svbotton
CALL nextchr
LD  a,0dfh    ;set dd ram
CALL LCDctrl
JP  sout3num

```

```

soutr0or1: CALL ReadKey ;third charactor LD r
LD  a,(Kb_Buf)
LD  hl,Buff4Key+2
LD  (hl),a

CP  clr_k
JP  z,clrs_out

CP  n00_k
JP  z,sOutr0

CP  n01_k
JP  z,sOutr1

JP  soutr0or1

```

```
sOutr0: CALL Write0
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CALL LCDctrl
```

```
JP soutr0109
```

```
sOutr1: CALL Write1
```

```
LD a,0dfh ;set dd ram
```

```
CALL LCDctrl
```

```
JP soutr1016
```

```
soutr0109: CALL ReadKey ;fourth charactor LD r0
```

```
LD a,(Kb_Buf)
```

```
LD hl,Buff4Key+3
```

```
LD (hl),a
```

```
CP clr_k
```

```
JP z,clrs_out
```

```
CP n01_k
```

```
JP z,sOutr01
```

```
CP n02_k
```

```
JP z,sOutr02
```

```
CP n03_k
```

```
JP z,sOutr03
```

```
CP n04_k
```

```
JP z,sOutr04
```

```
CP n05_k
```

```
JP z,sOutr05
```

CP n06\_k

JP z,sOutr06

CP n07\_k

JP z,sOutr07

CP n08\_k

JP z,sOutr08

CP n09\_k

JP z,sOutr09

JP sOutr0109

sOutr01: CALL Writel

JP sOutEnt

sOutr02: CALL Write2

JP sOutEnt

sOutr03: CALL Write3

JP sOutEnt

sOutr04: CALL Write4

JP sOutEnt

sOutr05: CALL Write5

JP sOutEnt

sOutr06: CALL Write6

เอกสารนี้เป็นเอกสาร **JP sOutEnt** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sOutr07: CALL Write7

JP sOutEnt

sOutr08: CALL Write8

JP sOutEnt

sOutr09: CALL Write9

JP sOutEnt

sOutr1016: CALL ReadKey ;fourth charactor LD r1

LD a,(Kb\_Buf)

LD hl, Buff4Key+3

LD (hl),a

CP clr\_k

JP z,clrs\_out

CP n00\_k

JP z,sOutr10

CP n01\_k

JP z,sOutr11

CP n02\_k

JP z,sOutr12

CP n03\_k

JP z,sOutr13

JP z,sOutr14

CP n05\_k

JP z,sOutr15

CP n06\_k

JP z,sOutr16

JP soutr1016

sOutr10: CALL Write0

JP sOutEnt

sOutr11: CALL Write1

JP sOutEnt

sOutr12: CALL Write2

JP sOutEnt

sOutr13: CALL Write3

JP sOutEnt

sOutr14: CALL Write4

JP sOutEnt

sOutr15: CALL Write5

JP sOutEnt

sOutr16: CALL Write6

JP sOutEnt

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sout3num: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl,Buff4Key+2

LD (hl),a

CP clr\_k

JP z,clrs\_out

CP n01\_k

JP z,sOon1

CP n02\_k

JP z,sOtw2

CP n03\_k

JP z,sOth3

CP n04\_k

JP z,sOfo4

CP n05\_k

JP z,sOfi5

CP n06\_k

JP z,sOsi6

CP n07\_k

JP z,sOse7

CP n08\_k

JP z,sOei8

JP sout3num

sOon1: CALL Write1

JP sOutEnt

sOtw2: CALL Write2

JP sOutEnt

sOth3: CALL Write3

JP sOutEnt

sOf4: CALL Write4

JP sOutEnt

sOfi5: CALL Write5

JP sOutEnt

sOsi6: CALL Write6

JP sOutEnt

sOse7: CALL Write7

JP sOutEnt

sOei8: CALL Write8

JP sOutEnt

sOutEnt: CALL ReadKey

LD a,(Kb\_Buf)

CP clr\_k

JP z,clr\_s\_out

```

CP ent_k
JP z,EntsOut
JP sOutEnt

```

```

EntsOut: CALL entsrch
          JP ExsrcOut

```

```

clrs_out: CALL putdsp1
          JP ExsrcOut

```

```

ExsrcOut: RET

```

```

;-----
; Routine   : srch_fun
; Description : search Function
; Register  : <not PUSH value>
;-----

```

```

srch_fun: LD hl,funcmess
          CALL nextchr
          LD  a,0ddh      ;set dd ram
          CALL LCDctrl

sfunchr2: CALL ReadKey   ;second charactor
          LD  a,(Kb_Buf)
          LD  hl,Buff4Key+1
          LD  (hl),a

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP clr\_k

JP z,clrs\_fun

CP n00\_k

JP z,sfunc0

JP sfuncr2

sfunc0: CALL Write0

LD a,0deh ;set dd ram

CALL LCDctrl

JP sfun3num

sfun3num: CALL ReadKey ;third charactor

LD a,(Kb\_Buf)

LD hl,Buf4Key+2

LD (hl),a

CP clr\_k

JP z,clrs\_fun

CP n01\_k

JP z,sFun01

CP n02\_k

JP z,sFun02

CP n03\_k

JP z,sFun03

CP n04\_k

JP z,sFun04

CP n05\_k

JP z,sFun05

CP n06\_k

JP z,sFun06

CP n07\_k

JP z,sFun07

JP sfun3num

sFun01: CALL Write1  
 CALL delayint  
 LD a,0d8h ;set dd ram  
 CALL LCDctrl  
 LD hl,func01  
 CALL nextchr  
 JP sFunEnt

sFun02: CALL Write2  
 CALL delayint  
 LD a,0d8h ;set dd ram  
 CALL LCDctrl  
 LD hl,func02  
 CALL nextchr  
 JP sFunEnt

sFun03: CALL Write3

CALL delayint

```

LD  a,0d8h      ;set dd ram
CALL LCDctrl
LD  hl,func03
CALL nextchr
JP  sFunEnt

```

sFun04: CALL Write4

```

CALL delayint
LD  a,0d8h      ;set dd ram
CALL LCDctrl
LD  hl,func04
CALL nextchr
JP  sFunEnt

```

sFun05: CALL Write5

```

CALL delayint
LD  a,0d8h      ;set dd ram
CALL LCDctrl
LD  hl,func05
CALL nextchr
JP  sFunEnt

```

sFun06: CALL Write6

```

CALL delayint
LD  a,0d8h      ;set dd ram
CALL LCDctrl
LD  hl,func06
CALL nextchr
JP  sFunEnt

```

sFun07: CALL Write7

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL delayint
LD a,0d8h ;set dd ram
CALL LCDctrl
LD hl,func07
CALL nextchr
JP sFunEnt

```

```
sFunEnt: CALL ReadKey
```

```

LD a,(Kb_Buf)
CP clr_k
JP z,clrs_fun

CP ent_k
JP z,EntsFun
JP sFunEnt

```

```

EntsFun: CALL entsrch
JP ExsrcFun

```

```

clrs_fun: CALL putdsp1
JP ExsrcFun

```

```
ExsrcFun: RET
```

```
-----
```

```

; Routine : entsrch
; Description : enter for search
; Register : <not PUSH value>

```

```
-----
```

```
entsrch: LD hl,(Cur_Addr)
```

เอกสารนี้เป็นเอกสาร LD (SrcAddr),hl การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

chks1: LD a,(Buff4Key)

LD d,a

LD hl,(SrchAddr)

LD a,(hl)

CP d

JP z,chks2

LD hl,(SrchAddr)

LD de,04

XOR a

ADC hl,de

LD (SrchAddr),hl

LD de,BotProg+4

XOR a

SBC hl,de

JP z,notfound

JP nc,notfound

JP chks1

chks2: LD hl,(SrchAddr)

INC hl

LD a,(Buff4Key+1)

LD d,a

LD a,(hl)

CP d

JP z,chks3

LD hl,(SrchAddr)

LD de,04

XOR a

ADC hl,de

LD (SrchAddr),hl

JP chks1

chks3: LD hl,(SrchAddr)

INC hl

INC hl

LD a,(Buff4Key+2)

LD d,a

LD a,(hl)

CP d

JP z,chks4

LD hl,(SrchAddr)

LD de,04

XOR a

ADC hl,de

LD (SrchAddr),hl

JP chks1

chks4: LD hl,(SrchAddr)

INC hl

INC hl

INC hl

LD a,(Buff4Key+3)

LD d,a

LD a,(hl)

CP d

JP z,srchOK

```

LD hl,(SrchAddr)
LD de,04
XOR a
ADC hl,de
LD (SrchAddr),hl
JP chks1

```

```
srchOK: LD hl,(SrchAddr)
```

```
LD (Cur_Addr),hl
```

```

CALL putdsp1 ; if < BotProg
CALL curoff
CALL getdsp1

LD hl,BotProg ; check > BotProg-12 (3 line)
LD de,0ch ;
XOR a ;
SBC hl,de ;
EX de,hl
LD hl,(Cur_Addr)
XOR a
SBC hl,de
JP c,topsdsp ; > BotProg-12
JP z,topsdsp

```

```
LD (Line_Add),de
```

```
LD (Top_Block),de
```

```
CALL cirdsp
```

```
JP exentsrc
```

```
LD (Line_Add),hl
LD (Top_Block),hl
CALL clr dsp
JP exentsrc
```

```
notfound: LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,srchErr
```

```
LD b,16
```

```
srchEdsp: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

```
CALL waitbusy
```

```
INC hl
```

```
DJNZ srchEdsp
```

```
CALL delay
```

```
CALL putdsp1
```

```
JP exentsrc
```

```
exentsrc: RET
```

```
-----
; Routine : FchgX999
; Description : First character for change variable
; Register :
-----
```

```
FchgX999: LD hl,KchgBuf
```

```
LD (hl),a
```

```
LD a,9ah ;set dd ram
```

```
CALL LCDctrl
```

RET

-----  
 ; Routine : Schg9X99  
 ; Description : Second charactor for change variable  
 ; Register :  
 -----

Schg9X99: LD hl,KchgBuf+1  
 LD (hl),a  
 LD a,9bh ;set dd ram  
 CALL LCDctrl  
 RET

-----  
 ; Routine : Tchg99X9  
 ; Description : Third charactor for change variable  
 ; Register :  
 -----

Tchg99X9: LD hl,KchgBuf+2  
 LD (hl),a  
 LD a,9ch ;set dd ram  
 CALL LCDctrl  
 RET

-----

เอก; Routine เอก: Fchg999X สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : Third charactor for change variable

; Register :

-----

Fchg999X: LD hl,KchgBuf+3

LD (hl),a

LD a,99h ;set dd ram

CALL LCDctrl

RET

-----

; Routine : delay

; Description : for delay time

; Register : A,B<loop counter>,HL<loop couter>

-----

delay: PUSH af

PUSH bc

PUSH hl

LD b,02h ;set loop

delay1: LD hl,0ffffh

delay2: DEC hl

LD a,h

OR 1

JP nz,delay2

DJNZ delay1

POP hl

POP bc

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

POP af

RET

---

```
; Routine      : delayint
; Description  : for delay time
; Register    : A,B<loop counter>,HL<loop counter>
```

---

```
delayint: PUSH af
          PUSH bc
          PUSH hl

          LD b,01h ;set loop
de1:     LD hl,4ff0h
de2:     DEC hl
          LD a,h
          OR l
          JP nz,de2
          DJNZ de1
```

POP hl

POP bc

POP af

RET

---

```
; Routine      : Mul16
```

```
; Description  : Multiply 16 bit result not over FFFFH
```

เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; Register : CA<Multiplier>,DE<Multiplicand>,HI<Result>
```

```
-----
```

```
Mul16: LD b,16
```

```
LD hl,0
```

```
Mult: SRL c
```

```
RR a
```

```
JP nc,Noadd
```

```
ADD hl,de
```

```
--Noadd: EX de,hl
```

```
ADD hl,hl
```

```
EX de,hl
```

```
DJNZ Mult
```

```
RET
```

```
; Routine : Setvalue
```

```
; Description : Set data to Variable
```

```
; Register :
```

```
-----
```

```
Setvalue: LD a,0
```

```
LD (clrset),a
```

```
LD a,99h ;set dd ram
```

```
CALL LCDctrl
```

```
LD a,00001111b ;display control d=1,c=1,b=1
```

```
CALL LCDctrl
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

setfnum: CALL ReadKey ;first no. ?999

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrflag

CP ent\_k

JP z,exitset

CP n00\_k

JP z,setfnum0

CP n01\_k

JP z,setfnum1

CP n02\_k

JP z,setfnum2

CP n03\_k

JP z,setfnum3

CP n04\_k

JP z,setfnum4

CP n05\_k

JP z,setfnum5

CP n06\_k

JP z,setfnum6

CP n07\_k

JP z,setfnum7

CP n08\_k

JP z,setfnum8

CP n09\_k

JP z,setfnum9

LD a,99h ;set dd ram

CALL LCDctrl

JP setfnum

setfnum0: CALL Write0

LD a,00

CALL FchgX999

JP setsnum

setfnum1: CALL Write1

LD a,01

CALL FchgX999

JP setsnum

setfnum2: CALL Write2

LD a,02

CALL FchgX999

JP setsnum

setfnum3: CALL Write3

LD a,03

CALL FchgX999

JP setsnum

เอกสารนี้เป็นเอกสารตัวอย่างสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**setfnum4: CALL Write4**

LD a,04

CALL FchgX999

JP setsnum

**setfnum5: CALL Write5**

LD a,05

CALL FchgX999

JP setsnum

**setfnum6: CALL Write6**

LD a,06

CALL FchgX999

JP setsnum

**setfnum7: CALL Write7**

LD a,07

CALL FchgX999

JP setsnum

**setfnum8: CALL Write8**

LD a,08

CALL FchgX999

JP setsnum

**setfnum9: CALL Write9**

LD a,09

CALL FchgX999

JP setsnum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

setsnum: CALL ReadKey ;second no. 9?99

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrflag

CP ent\_k

JP z,exitset

CP n00\_k

JP z,setsnum0

CP n01\_k

JP z,setsnum1

CP n02\_k

JP z,setsnum2

CP n03\_k

JP z,setsnum3

CP n04\_k

JP z,setsnum4

CP n05\_k

JP z,setsnum5

CP n06\_k

JP z,setsnum6

CP n07\_k

JP z,setsnum7

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n08\_k

JP z,setsnum8

CP n09\_k

JP z,setsnum9

LD a,9ah ;set dd ram

CALL LCDctrl

JP setsnum

setsnum0: CALL Write0

LD a,00

CALL Schg9X99

JP settnum

setsnum1: CALL Write1

LD a,01

CALL Schg9X99

JP settnum

setsnum2: CALL Write2

LD a,02

CALL Schg9X99

JP settnum

setsnum3: CALL Write3

LD a,03

CALL Schg9X99

JP settnum

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

setsnum4: CALL Write4

LD a,04

CALL Schg9X99

JP settnum

setsnum5: CALL Write5

LD a,05

CALL Schg9X99

JP settnum

setsnum6: CALL Write6

LD a,06

CALL Schg9X99

JP settnum

setsnum7: CALL Write7

LD a,07

CALL Schg9X99

JP settnum

setsnum8: CALL Write8

LD a,08

CALL Schg9X99

JP settnum

setsnum9: CALL Write9

LD a,09

CALL Schg9X99

JP settnum

LD a,(Kb\_\_Buf)

CP clr\_k

JP z,clrflag

CP ent\_k

JP z,exitset

CP n00\_k

JP z,settnum0

CP n01\_k

JP z,settnum1

CP n02\_k

JP z,settnum2

CP n03\_k

JP z,settnum3

CP n04\_k

JP z,settnum4

CP n05\_k

JP z,settnum5

CP n06\_k

JP z,settnum6

CP n07\_k

JP z,settnum7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n08\_k

JP z,settnum8

CP n09\_k

JP z,settnum9

LD a,9bh ;set dd ram

CALL LCDctrl

JP settnum

settnum0: CALL Write0

LD a,00

CALL Tchg99X9

JP sefonum

settnum1: CALL Write1

LD a,01

CALL Tchg99X9

JP sefonum

settnum2: CALL Write2

LD a,02

CALL Tchg99X9

JP sefonum

settnum3: CALL Write3

LD a,03

CALL Tchg99X9

JP sefonum

settnum4: CALL Write4

LD a,04

CALL Tchg99X9

JP sefonum

settnum5: CALL Write5

LD a,05

CALL Tchg99X9

JP sefonum

settnum6: CALL Write6

LD a,06

CALL Tchg99X9

JP sefonum

settnum7: CALL Write7

LD a,07

CALL Tchg99X9

JP sefonum

settnum8: CALL Write8

LD a,08

CALL Tchg99X9

JP sefonum

settnum9: CALL Write9

LD a,09

CALL Tchg99X9

JP sefonum

LD a,(Kb\_Buf)

CP clr\_k

JP z,clrflag

CP ent\_k

JP z,exitset

CP n00\_k

JP z,sefonum0

CP n01\_k

JP z,sefonum1

CP n02\_k

JP z,sefonum2

CP n03\_k

JP z,sefonum3

CP n04\_k

JP z,sefonum4

CP n05\_k

JP z,sefonum5

CP n06\_k

JP z,sefonum6

CP n07\_k

JP z,sefonum7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n08\_k

JP z,sefonum8

CP n09\_k

JP z,sefonum9

LD a,9ch ;set dd ram

CALL LCDctrl

JP sefonum

sefonum0: CALL Write0

LD a,00

CALL Fchg999X

JP setfnum

sefonum1: CALL Write1

LD a,01

CALL Fchg999X

JP setfnum

sefonum2: CALL Write2

LD a,02

CALL Fchg999X

JP setfnum

sefonum3: CALL Write3

LD a,03

CALL Fchg999X

JP setfnum

sefonum4: CALL Write4

LD a,04

CALL Fchg999X

JP setfnum

sefonum5: CALL Write5

LD a,05

CALL Fchg999X

JP setfnum

sefonum6: CALL Write6

LD a,06

CALL Fchg999X

JP setfnum

sefonum7: CALL Write7

LD a,07

CALL Fchg999X

JP setfnum

sefonum8: CALL Write8

LD a,08

CALL Fchg999X

JP setfnum

sefonum9: CALL Write9

LD a,09

CALL Fchg999X

JP setfnum

clrflag: LD a,01

LD (clrset),a

```
exitset: RET
```

```
-----
; Routine   : minscale
; Description : Display and change data Min scale
; Register  :
-----
```

```
minscale: CALL getdsp1
```

```
CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,mindsp
```

```
CALL nextchr
```

```
mn2ch: CALL readkey
```

```
LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,exitmn
```

```
CP dta_k
```

```
JP z,Mindtdsp
```

```
CP n01_k
```

```
JP z,dspmn1
```

```
CP n02_k
```

```
JP z,dspmn2
```

CP n03\_k

JP z,dspmn3

CP n04\_k

JP z,dspmn4

CP n05\_k

JP z,dspmn5

CP n06\_k

JP z,dspmn6

CP n07\_k

JP z,dspmn7

CP n08\_k

JP z,dspmn8

JP mn2ch

Mindtdsp: JP mn1\_4

dspallmn: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,mnpgedwn

CP up\_k

JP z,mnpgeup

CP clr\_k

JP z,exitmn

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP dspallmn

mnpgeup: LD a,(page)

CP 01

JP z,mn1\_4

JP dspallmn

mnpgedwn: LD a,(page)

CP 02

JP z,mn5\_8

JP dspallmn

mn1\_4: CALL clrdsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,mn1dsp

CALL nextchr

LD de,(Min1Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,mn2dsp

CALL nextchr

LD de,(Min2Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

```
LD hl,mn3dsp
CALL nextchr
LD de,(Min3Buf)
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,mn4dsp
CALL nextchr
LD de,(Min4Buf)
CALL GtoNum
```

```
LD a,02
LD (page),a
JP dspallmn
```

```
mn5_8: CALL clrdsp
LD a,80h ;set dd ram
CALL LCDctrl
LD hl,mn5dsp
CALL nextchr
LD de,(Min5Buf)
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
CALL LCDctrl
LD hl,mn6dsp
CALL nextchr
LD de,(Min6Buf)
CALL GtoNum
```

```

LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mn7dsp
CALL nextchr
LD de,(Min7Buf)
CALL GtoNum

LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,mn8dsp
CALL nextchr
LD de,(Min8Buf)
CALL GtoNum

LD a,01
LD (page),a
JP dspallmn

dspmn1: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mn1dsp
CALL nextchr
LD a,01
LD (NoDspAct),a
JP Minonlyd

dspmn2: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mn2dsp
CALL nextchr

```

เอกสารนี้เป็นเอกสาร LD a,02 ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD (NoDspAct),a

JP Minonlyd

dspm3: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mn3dsp

CALL nextchr

LD a,03

LD (NoDspAct),a

JP Minonlyd

dspm4: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mn4dsp

CALL nextchr

LD a,04

LD (NoDspAct),a

JP Minonlyd

dspm5: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mn5dsp

CALL nextchr

LD a,05

LD (NoDspAct),a

JP Minonlyd

dspm6: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mn6dsp

CALL nextchr

```
LD a,06
LD (NoDspAct),a
JP Minonlyd
```

```
dsprmn7: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mn7dsp
CALL nextchr
LD a,07
LD (NoDspAct),a
JP Minonlyd
```

```
dsprmn8: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mn8dsp
CALL nextchr
LD a,08
LD (NoDspAct),a
JP Minonlyd
```

```
Minonlyd: CALL readkey
LD a,(Kb_Buf)
CP clr_k
JP z,exitmn

CP dta_k
JP z,Mindtonl
JP Minonlyd
```

```
Mindtonl: LD a,(NoDspAct)
```

CP 01

JP z,Minno\_1

CP 02

JP z,Minno\_2

CP 03

JP z,Minno\_3

CP 04

JP z,Minno\_4

CP 05

JP z,Minno\_5

CP 06

JP z,Minno\_6

CP 07

JP z,Minno\_7

CP 08

JP z,Minno\_8

JP exitmn

Minno\_1: LD de,(Min1Buf)

CALL kGtoKch4

JP chkextmn

Minno\_2: LD de,(Min2Buf)

CALL kGtoKch4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP chkextmn

Minno\_3: LD de,(Min3Buf)

CALL kGtoKch4

JP chkextmn

Minno\_4: LD de,(Min4Buf)

CALL kGtoKch4

JP chkextmn

Minno\_5: LD de,(Min5Buf)

CALL kGtoKch4

JP chkextmn

Minno\_6: LD de,(Min6Buf)

CALL kGtoKch4

JP chkextmn

Minno\_7: LD de,(Min7Buf)

CALL kGtoKch4

JP chkextmn

Minno\_8: LD de,(Min8Buf)

CALL kGtoKch4

JP chkextmn

chkextmn: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmn

CP chg\_k

JP z,chgmn

JP chkextmn

chgmn: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgmn

CP 01

JP z,exitmn

JP exitmn

entchgmn: LD a,(NoDspAct)

CP 01

JP z,MinOK\_1

CP 02

JP z,MinOK\_2

CP 03

JP z,MinOK\_3

CP 04

JP z,MinOK\_4

CP 05

JP z,MinOK\_5

CP 06

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP 07

JP z,MinOK\_7

CP 08

JP z,MinOK\_8

JP exitmn

MinOK\_1: CALL Mulvar

LD (Min1Buf),hl

JP exitmn

MinOK\_2: CALL Mulvar

LD (Min2Buf),hl

JP exitmn

MinOK\_3: CALL Mulvar

LD (Min3Buf),hl

JP exitmn

MinOK\_4: CALL Mulvar

LD (Min4Buf),hl

JP exitmn

MinOK\_5: CALL Mulvar

LD (Min5Buf),hl

JP exitmn

MinOK\_6: CALL Mulvar

LD (Min6Buf),hl

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MinOK\_7: CALL Mulvar

LD (Min7Buf),hl

JP exitmn

MinOK\_8: CALL Mulvar

LD (Min8Buf),hl

JP exitmn

exitmn: LD a,00001100b ;display control d=1,c=1,b=1

CALL LCDctrl

CALL putdsp1

RET

-----  
 ; Routine : maxscale  
 ; Description : Display and change data Max scale  
 ; Register :

-----

maxscale: CALL getdsp1

CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,maxdsp

CALL nextchr

mx2ch: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmx

CP dta\_k

JP z,Maxdtdsp

CP n01\_k

JP z,dspmx1

CP n02\_k

JP z,dspmx2

CP n03\_k

JP z,dspmx3

CP n04\_k

JP z,dspmx4

CP n05\_k

JP z,dspmx5

CP n06\_k

JP z,dspmx6

CP n07\_k

JP z,dspmx7

CP n08\_k

JP z,dspmx8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP mx2ch

Maxdtdsp: JP mx1\_4

dspallmx: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,mxpgedwn

CP up\_k

JP z,mxpgoup

CP clr\_k

JP z,exitmx

JP dspallmx

mxpgoup: LD a,(page)

CP 01

JP z,mx1\_4

JP dspallmx

mxpgedwn: LD a,(page)

CP 02

JP z,mx5\_8

JP dspallmx

mx1\_4: CALL clrdsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,mx1dsp

```
CALL nextchr
LD de,(Max1Buf)
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
CALL LCDctrl
LD hl,mx2dsp
CALL nextchr
LD de,(Max2Buf)
CALL GtoNum
```

```
LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mx3dsp
CALL nextchr
LD de,(Max3Buf)
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,mx4dsp
CALL nextchr
LD de,(Max4Buf)
CALL GtoNum
```

```
LD a,02
LD (page),a
JP dspallmx
```

**mx5\_8: CALL clrdsp**

เอกสารนี้เป็นเอกสาร LD ส่ง a,80h สำหรับ ;set dd ram เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL LCDctrl

LD hl,mx5dsp

CALL nextchr

LD de,(Max5Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,mx6dsp

CALL nextchr

LD de,(Max6Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mx7dsp

CALL nextchr

LD de,(Max7Buf)

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,mx8dsp

CALL nextchr

LD de,(Max8Buf)

CALL GtoNum

LD a,01

LD (page),a

JP dspallmx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dspmx1: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx1dsp
        CALL nextchr
        LD  a,01
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx2: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx2dsp
        CALL nextchr
        LD  a,02
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx3: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx3dsp
        CALL nextchr
        LD  a,03
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx4: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx4dsp
        CALL nextchr
        LD  a,04
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx5: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx5dsp
        CALL nextchr
        LD  a,05
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx6: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx6dsp
        CALL nextchr
        LD  a,06
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx7: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx7dsp
        CALL nextchr
        LD  a,07
        LD  (NoDspAct),a
        JP  Maxonlyd

```

```

dspmx8: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD  hl,mx8dsp
        CALL nextchr
        LD  a,08

```

JP Maxonlyd

Maxonlyd: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmx

CP dta\_k

JP z,Maxdtonl

JP Maxonlyd

Maxdtonl: LD a,(NoDspAct)

CP 01

JP z,Maxno\_1

CP 02

JP z,Maxno\_2

CP 03

JP z,Maxno\_3

CP 04

JP z,Maxno\_4

CP 05

JP z,Maxno\_5

CP 06

JP z,Maxno\_6



CP 07

JP z,Maxno\_7

CP 08

JP z,Maxno\_8

JP exitmx

Maxno\_1: LD de,(Max1Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_2: LD de,(Max2Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_3: LD de,(Max3Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_4: LD de,(Max4Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_5: LD de,(Max5Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_6: LD de,(Max6Buf)

CALL kGtoKch4

JP chkextmx



Maxno\_7: LD de,(Max7Buf)

CALL kGtoKch4

JP chkextmx

Maxno\_8: LD de,(Max8Buf)

CALL kGtoKch4

JP chkextmx

chkextmx: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmx

CP chg\_k

JP z,chgmx

JP chkextmx

chgmx: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgmx

CP 01

JP z,exitmx

JP exitmx

entchgmx: LD a,(NoDspAct)

CP 01

JP z,MaxOK\_1

CP 02

JP z,MaxOK\_2

CP 03

JP z,MaxOK\_3

CP 04

JP z,MaxOK\_4

CP 05

JP z,MaxOK\_5

CP 06

JP z,MaxOK\_6

CP 07

JP z,MaxOK\_7

CP 08

JP z,MaxOK\_8

JP exitmx

MaxOK\_1: CALL Mulvar

LD (Max1Buf),hl

JP exitmx

MaxOK\_2: CALL Mulvar

LD (Max2Buf),hl

JP exitmx

MaxOK\_3: CALL Mulvar

LD (Max3Buf),hl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitmx

MaxOK\_4: CALL Mulvar

LD (Max4Buf),hl

JP exitmx

MaxOK\_5: CALL Mulvar

LD (Max5Buf),hl

JP exitmx

MaxOK\_6: CALL Mulvar

LD (Max6Buf),hl

JP exitmx

MaxOK\_7: CALL Mulvar

LD (Max7Buf),hl

JP exitmx

MaxOK\_8: CALL Mulvar

LD (Max8Buf),hl

JP exitmx

exitmx: LD a,00001100b ;display control d=1,c=1,b=1

CALL LCDctrl

CALL putdsp1

RET

-----

เอกสาร; Routine ออกส: kpkkey ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; Description : Display and change data KP

; Register :

-----

kpkey: CALL getdsp1

CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,kpdsp

CALL nextchr

kp2ch: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitkp

CP dta\_k

JP z,Kpdt dsp

CP n01\_k

JP z,dspkp1

CP n02\_k

JP z,dspkp2

CP n03\_k

JP z,dspkp3

CP n04\_k

JP z,dspkp4

เอกสารนี้เป็นเอกสารที่ให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n05\_k

JP z,dspkp5

CP n06\_k

JP z,dspkp6

CP n07\_k

JP z,dspkp7

CP n08\_k

JP z,dspkp8

JP kp2ch

Kpdtdsp: JP kpl\_4

dspallkp: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,kppgedwn

CP up\_k

JP z,kppgeup

CP clr\_k

JP z,exitkp

JP dspallkp

CP 01

JP z,kp1\_4

JP dspallkp

kppgedwn: LD a,(page)

CP 02

JP z,kp5\_8

JP dspallkp

kp1\_4: CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,kp1dsp

CALL nextchr

LD de,(Kp1Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,kp2dsp

CALL nextchr

LD de,(Kp2Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kp3dsp

CALL nextchr

LD de,(Kp3Buf)

CALL GtoNum

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,kp4dsp
```

```
CALL nextchr
```

```
LD de,(Kp4Buf)
```

```
CALL GtoNum
```

```
LD a,02
```

```
LD (page),a
```

```
JP dspallkp
```

```
kp5_8: CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,kp5dsp
```

```
CALL nextchr
```

```
LD de,(Kp5Buf)
```

```
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,kp6dsp
```

```
CALL nextchr
```

```
LD de,(Kp6Buf)
```

```
CALL GtoNum
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,kp7dsp
```

```
CALL nextchr
```

```
LD de,(Kp7Buf)
```

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,kp8dsp

CALL nextchr

LD de,(Kp8Buf)

CALL GtoNum

LD a,01

LD (page),a

JP dspallkp

dspkp1: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kp1dsp

CALL nextchr

LD a,01

LD (NoDspAct),a

JP Kponlyd

dspkp2: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kp2dsp

CALL nextchr

LD a,02

LD (NoDspAct),a

JP Kponlyd

dspkp3: LD a,90h ;set dd ram

CALL LCDctrl

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD hl,kp3dsp
CALL nextchr
LD a,03
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp4: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp4dsp
CALL nextchr
LD a,04
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp5: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp5dsp
CALL nextchr
LD a,05
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp6: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp6dsp
CALL nextchr
LD a,06
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp7: LD a,90h ;set dd ram
```

```
LD hl,kp3dsp
CALL nextchr
LD a,03
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp4: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp4dsp
CALL nextchr
LD a,04
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp5: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp5dsp
CALL nextchr
LD a,05
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp6: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,kp6dsp
CALL nextchr
LD a,06
LD (NoDspAct),a
JP Kponlyd
```

```
dspkp7: LD a,90h ;set dd ram
```

CALL LCDctrl

LD hl,kp7dsp

CALL nextchr

LD a,07

LD (NoDspAct),a

JP Kponlyd

dspkp8: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kp8dsp

CALL nextchr

LD a,08

LD (NoDspAct),a

JP Kponlyd

Kponlyd: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitkp

CP dta\_k

JP z,Kpdtonl

JP Kponlyd

Kpdtonl: LD a,(NoDspAct)

CP 01

JP z,Kpno\_1

CP 02

JP z,Kpno\_2

CP 03

JP z,Kpno\_3

CP 04

JP z,Kpno\_4

CP 05

JP z,Kpno\_5

CP 06

JP z,Kpno\_6

CP 07

JP z,Kpno\_7

CP 08

JP z,Kpno\_8

JP exitkp

Kpno\_1: LD de,(Kp1Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_2: LD de,(Kp2Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_3: LD de,(Kp3Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_4: LD de,(Kp4Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_5: LD de,(Kp5Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_6: LD de,(Kp6Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_7: LD de,(Kp7Buf)

CALL kGtoKch4

JP chkextkp

Kpno\_8: LD de,(Kp8Buf)

CALL kGtoKch4

JP chkextkp

chkextkp: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitkp

CP chg\_k

JP z,chgkp

JP chkextkp

LD a,(clrset)

CP 00

JP z,entchgkp

CP 01

JP z,exitkp

JP exitkp

entchgkp: LD a,(NoDspAct)

CP 01

JP z,KpOK\_1

CP 02

JP z,KpOK\_2

CP 03

JP z,KpOK\_3

CP 04

JP z,KpOK\_4

CP 05

JP z,KpOK\_5

CP 06

JP z,KpOK\_6

CP 07

JP z,KpOK\_7

เอกสารนี้เป็นเอกสาร **CP 08** นี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP z,KpOK\_8

JP exitkp

KpOK\_1: CALL Mulvar

LD (Kp1Buf),hl

JP exitkp

KpOK\_2: CALL Mulvar

LD (Kp2Buf),hl

JP exitkp

KpOK\_3: CALL Mulvar

LD (Kp3Buf),hl

JP exitkp

KpOK\_4: CALL Mulvar

LD (Kp4Buf),hl

JP exitkp

KpOK\_5: CALL Mulvar

LD (Kp5Buf),hl

JP exitkp

KpOK\_6: CALL Mulvar

LD (Kp6Buf),hl

JP exitkp

KpOK\_7: CALL Mulvar

LD (Kp7Buf),hl

JP exitkp

KpOK\_8: CALL Mulvar

LD (Kp8Buf),hl

JP exitkp

exitkp: LD a,00001100b ;display control d=1,c=1,b=1

CALL LCDctrl

CALL putdsp1

RET

-----  
; Routine : kikey

; Description : Display and change data KI

; Register :  
-----

kikey: CALL getdsp1

CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,kidsp

CALL nextchr

ki2ch: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitki

CP dta\_k

JP z,Kidtdsp

CP n01\_k

JP z,dspki1

CP n02\_k

JP z,dspki2

CP n03\_k

JP z,dspki3

CP n04\_k

JP z,dspki4

CP n05\_k

JP z,dspki5

CP n06\_k

JP z,dspki6

CP n07\_k

JP z,dspki7

CP n08\_k

JP z,dspki8

JP ki2ch

Kidtdsp: JP ki1\_4

dspallki: CALL readkey

เอกสารนี้เป็นเอกสารที่สร้างขึ้นเพื่อการใช้งานในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP down\_k

JP z,kipgedwn

CP up\_k

JP z,kipgeup

CP clr\_k

JP z,exitki

JP dspallki

kipgeup: LD a,(page)

CP 01

JP z,ki1\_4

JP dspallki

kipgedwn: LD a,(page)

CP 02

JP z,ki5\_8

JP dspallki

ki1\_4: CALL clrdsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,ki1dsp

CALL nextchr

LD de,(Ki1Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

```
CALL LCDctrl
```

```
LD hl,ki2dsp
```

```
CALL nextchr
```

```
LD de,(Ki2Buf)
```

```
CALL GtoNum
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,ki3dsp
```

```
CALL nextchr
```

```
LD de,(Ki3Buf)
```

```
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,ki4dsp
```

```
CALL nextchr
```

```
LD de,(Ki4Buf)
```

```
CALL GtoNum
```

```
LD a,02
```

```
LD (page),a
```

```
JP dspallki
```

```
ki5_8: CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,ki5dsp
```

```
CALL nextchr
```

```
LD de,(Ki5Buf)
```

```
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
CALL LCDctrl
LD hl,ki6dsp
CALL nextchr
LD de,(Ki6Buf)
CALL GtoNum
```

```
LD a,90h ;set dd ram
CALL LCDctrl
LD hl,ki7dsp
CALL nextchr
LD de,(Ki7Buf)
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,ki8dsp
CALL nextchr
LD de,(Ki8Buf)
CALL GtoNum
```

```
LD a,01
LD (page),a
JP dspalki
```

```
dspki1: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,ki1dsp
CALL nextchr
```

เอกสารนี้เป็นเอกสาร LD a,01 ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LD (NoDspAct),a  
JP Kionlyd

dspki2: LD a,90h ;set dd ram  
CALL LCDctrl  
LD hl,ki2dsp  
CALL nextchr  
LD a,02  
LD (NoDspAct),a  
JP Kionlyd

dspki3: LD a,90h ;set dd ram  
CALL LCDctrl  
LD hl,ki3dsp  
CALL nextchr  
LD a,03  
LD (NoDspAct),a  
JP Kionlyd

dspki4: LD a,90h ;set dd ram  
CALL LCDctrl  
LD hl,ki4dsp  
CALL nextchr  
LD a,04  
LD (NoDspAct),a  
JP Kionlyd

dspki5: LD a,90h ;set dd ram  
CALL LCDctrl  
LD hl,ki5dsp

LD a,05

LD (NoDspAct),a

JP Kionlyd

dspki6: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,ki6dsp

CALL nextchr

LD a,06

LD (NoDspAct),a

JP Kionlyd

dspki7: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,ki7dsp

CALL nextchr

LD a,07

LD (NoDspAct),a

JP Kionlyd

dspki8: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,ki8dsp

CALL nextchr

LD a,08

LD (NoDspAct),a

JP Kionlyd

**Kionlyd:** CALL readkey

เอกสารนี้เป็นเอกสาร (LD a,(Kb\_Buf) รับการจ้างงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP clr\_k

JP z,exitki

CP dta\_k

JP z,Kidtonl

JP Kionlyd

**Kidtonl:** LD a,(NoDspAct)

CP 01

JP z,Kino\_1

CP 02

JP z,Kino\_2

CP 03

JP z,Kino\_3

CP 04

JP z,Kino\_4

CP 05

JP z,Kino\_5

CP 06

JP z,Kino\_6

CP 07

JP z,Kino\_7

CP 08

JP z,Kino\_8

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitki

Kino\_1: LD de,(Ki1Buf)

CALL kGtoKch4

JP chkextki

Kino\_2: LD de,(Ki2Buf)

CALL kGtoKch4

JP chkextki

Kino\_3: LD de,(Ki3Buf)

CALL kGtoKch4

JP chkextki

Kino\_4: LD de,(Ki4Buf)

CALL kGtoKch4

JP chkextki

Kino\_5: LD de,(Ki5Buf)

CALL kGtoKch4

JP chkextki

Kino\_6: LD de,(Ki6Buf)

CALL kGtoKch4

JP chkextki

Kino\_7: LD de,(Ki7Buf)

CALL kGtoKch4

JP chkextki

เอกสารนี้ Kino\_8: LD de,(Ki8Buf) ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL kGtoKch4

JP chkextki

chkextki: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitki

CP chg\_k

JP z,chgki

JP chkextki

chgki: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgki

CP 01

JP z,exitki

JP exitki

entchgki: LD a,(NoDspAct)

CP 01

JP z,KiOK\_1

CP 02

JP z,KiOK\_2

CP 03

JP z,KiOK\_3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP 04

JP z,KiOK\_4

CP 05

JP z,KiOK\_5

CP 06

JP z,KiOK\_6

CP 07

JP z,KiOK\_7

CP 08

JP z,KiOK\_8

JP exitki

KiOK\_1: CALL Mulvar

LD (Ki1Buf),hl

JP exitki

KiOK\_2: CALL Mulvar

LD (Ki2Buf),hl

JP exitki

KiOK\_3: CALL Mulvar

LD (Ki3Buf),hl

JP exitki

KiOK\_4: CALL Mulvar

LD (Ki4Buf),hl

เอกสารนี้เป็นเอกสาร **JP exitki** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

KiOK\_5: CALL Mulvar

LD (Ki5Buf),hl

JP exitki

KiOK\_6: CALL Mulvar

LD (Ki6Buf),hl

JP exitki

KiOK\_7: CALL Mulvar

LD (Ki7Buf),hl

JP exitki

KiOK\_8: CALL Mulvar

LD (Ki8Buf),hl

JP exitki

exitki: LD a,00001100b ;display control d=1,c=1,b=1

CALL LCDctrl

CALL putdsp1

RET

-----

; Routine : kdkey

; Description : Display and change data KD

; Register :

-----

```
CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,kddsp
```

```
CALL nextchr
```

```
kd2ch: CALL readkey
```

```
LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,exitkd
```

```
CP dta_k
```

```
JP z,Kddtdsp
```

```
CP n01_k
```

```
JP z,dspkd1
```

```
CP n02_k
```

```
JP z,dspkd2
```

```
CP n03_k
```

```
JP z,dspkd3
```

```
CP n04_k
```

```
JP z,dspkd4
```

```
CP n05_k
```

```
JP z,dspkd5
```

```
CP n06_k
```

JP z,dspkd6

CP n07\_k

JP z,dspkd7

CP n08\_k

JP z,dspkd8

JP kd2ch

Kddtdsp: JP kd1\_4

dspallkd: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,kdpgedwn

CP up\_k

JP z,kdpgeup

CP clr\_k

JP z,exitkd

JP dspallkd

kdpgeup: LD a,(page)

CP 01

JP z,kd1\_4

JP dspallkd

CP 02

JP z,kd5\_8

JP dspallkd

kd1\_4: CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,kd1dsp

CALL nextchr

LD de,(Kd1Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,kd2dsp

CALL nextchr

LD de,(Kd2Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kd3dsp

CALL nextchr

LD de,(Kd3Buf)

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,kd4dsp

CALL nextchr

LD de,(Kd4Buf)

CALL GtoNum

LD a,02

LD (page),a

JP dspallkd

kd5\_8: CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,kd5dsp

CALL nextchr

LD de,(Kd5Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,kd6dsp

CALL nextchr

LD de,(Kd6Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kd7dsp

CALL nextchr

LD de,(Kd7Buf)

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,kd8dsp

```
CALL nextchr
LD de,(Kd8Buf)
CALL GtoNum
```

```
LD a,01
LD (page),a
JP dspallkd
```

```
dspkd1: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd1dsp
CALL nextchr
LD a,01
LD (NoDspAct),a
JP Kdonlyd
```

```
dspkd2: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd2dsp
CALL nextchr
LD a,02
LD (NoDspAct),a
JP Kdonlyd
```

```
dspkd3: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd3dsp
CALL nextchr
LD a,03
LD (NoDspAct),a
JP Kdonlyd
```

```

dspkd4: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd4dsp
CALL nextchr
LD a,04
LD (NoDspAct),a
JP Kdonlyd

```

```

dspkd5: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd5dsp
CALL nextchr
LD a,05
LD (NoDspAct),a
JP Kdonlyd

```

```

dspkd6: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd6dsp
CALL nextchr
LD a,06
LD (NoDspAct),a
JP Kdonlyd

```

```

dspkd7: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,kd7dsp
CALL nextchr
LD a,07
LD (NoDspAct),a

```

JP Kdonlyd

dspkd8: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,kd8dsp

CALL nextchr

LD a,08

LD (NoDspAct),a

JP Kdonlyd

Kdonlyd: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitkd

CP dta\_k

JP z,Kddtonl

JP Kdonlyd

Kddtonl: LD a,(NoDspAct)

CP 01

JP z,Kdno\_1

CP 02

JP z,Kdno\_2

CP 03

JP z,Kdno\_3

CP 04

JP z,Kdno\_4

CP 05

JP z,Kdno\_5

CP 06

JP z,Kdno\_6

CP 07

JP z,Kdno\_7

CP 08

JP z,Kdno\_8

JP exitkd

Kdno\_1: LD de,(Kd1Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_2: LD de,(Kd2Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_3: LD de,(Kd3Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_4: LD de,(Kd4Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_5: LD de,(Kd5Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_6: LD de,(Kd6Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_7: LD de,(Kd7Buf)

CALL kGtoKch4

JP chkextkd

Kdno\_8: LD de,(Kd8Buf)

CALL kGtoKch4

JP chkextkd

chkextkd: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitkd

CP chg\_k

JP z,chgkd

JP chkextkd

chgkd: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgkd

JP z,exitkd

JP exitkd

entchgkd: LD a,(NoDspAct)

CP 01

JP z,KdOK\_1

CP 02

JP z,KdOK\_2

CP 03

JP z,KdOK\_3

CP 04

JP z,KdOK\_4

CP 05

JP z,KdOK\_5

CP 06

JP z,KdOK\_6

CP 07

JP z,KdOK\_7

CP 08

JP z,KdOK\_8

JP exitkd

KdOK\_1: CALL Mulvar

LD (Kd1Buf),hl

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitkd

KdOK\_2: CALL Mulvar

LD (Kd2Buf),hl

JP exitkd

KdOK\_3: CALL Mulvar

LD (Kd3Buf),hl

JP exitkd

KdOK\_4: CALL Mulvar

LD (Kd4Buf),hl

JP exitkd

KdOK\_5: CALL Mulvar

LD (Kd5Buf),hl

JP exitkd

KdOK\_6: CALL Mulvar

LD (Kd6Buf),hl

JP exitkd

KdOK\_7: CALL Mulvar

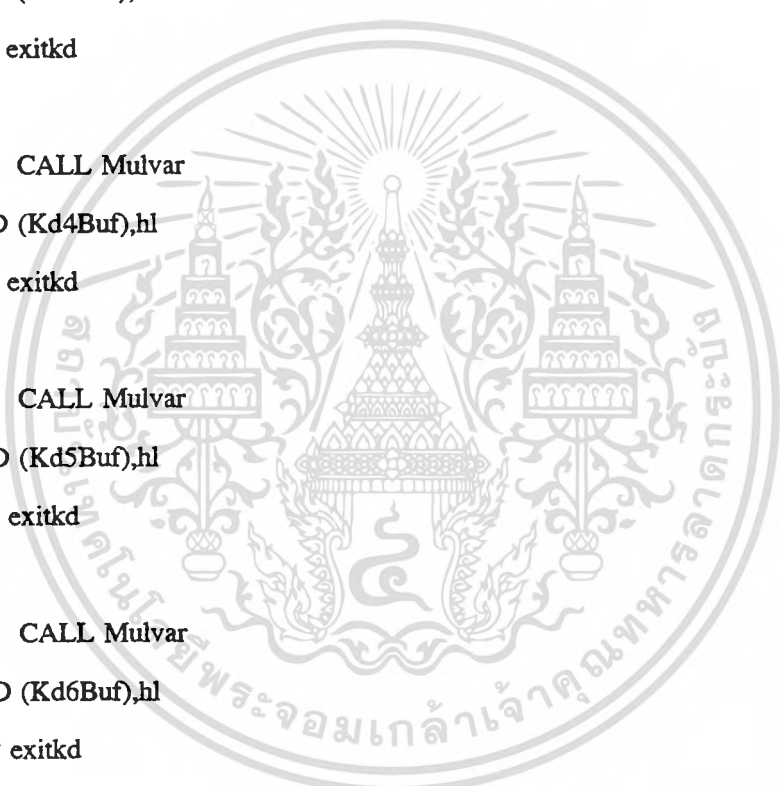
LD (Kd7Buf),hl

JP exitkd

KdOK\_8: CALL Mulvar

LD (Kd8Buf),hl

JP exitkd



```
exitkd: LD a,00001100b ;display control d=1,c=1,b=1
```

```
CALL LCDctrl
```

```
CALL putdsp1
```

```
RET
```

```
-----
```

```
; Routine : svkey
```

```
; Description : Display and change data SV
```

```
; Register :
```

```
-----
```

```
svkey: CALL getdsp1
```

```
CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,svdsp
```

```
CALL nextchr
```

```
sv2ch: CALL readkey
```

```
LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,exitsv
```

```
CP dta_k
```

```
JP z,Svdt dsp
```

```
CP n01_k
```

```
JP z,dspsv1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n02\_k

JP z,dspsv2

CP n03\_k

JP z,dspsv3

CP n04\_k

JP z,dspsv4

CP n05\_k

JP z,dspsv5

CP n06\_k

JP z,dspsv6

CP n07\_k

JP z,dspsv7

CP n08\_k

JP z,dspsv8

JP sv2ch

Svtdtosp: JP sv1\_4

dspallsv: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,svpgedwn

CP up\_k

เอกสารนี้เป็นเอกสาร **JP z,svpgedwn** สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP clr_k
JP z,exitsv

JP dspallsv

```

```
svpgeup: LD a,(page)
```

```

CP 01
JP z,sv1_4
JP dspallsv

```

```
svpgedwn: LD a,(page)
```

```

CP 02
JP z,sv5_8
JP dspallsv

```

```
sv1_4: CALL clrdsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv1dsp
```

```
CALL nextchr
```

```
LD de,(Sv1Buf)
```

```
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv2dsp
```

```
CALL nextchr
```

```
LD de,(Sv2Buf)
```

```
CALL GtoNum
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv3dsp
```

```
CALL nextchr
```

```
LD de,(Sv3Buf)
```

```
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv4dsp
```

```
CALL nextchr
```

```
LD de,(Sv4Buf)
```

```
CALL GtoNum
```

```
LD a,02
```

```
LD (page),a
```

```
JP dspallsv
```

```
sv5_8: CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv5dsp
```

```
CALL nextchr
```

```
LD de,(Sv5Buf)
```

```
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,sv6dsp
```

```
CALL nextchr
```

LD de,(Sv6Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,sv7dsp

CALL nextchr

LD de,(Sv7Buf)

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,sv8dsp

CALL nextchr

LD de,(Sv8Buf)

CALL GtoNum

LD a,01

LD (page),a

JP dspallsv

dspsv1: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,sv1dsp

CALL nextchr

LD a,01

LD (NoDspAct),a

JP Svonlyd

dspsv2: LD a,90h ;set dd ram

CALL LCDctrl

เอกสารนี้เป็นเอกสารลิขสิทธิ์ภายใต้การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD hl,sv2dsp
CALL nextchr
LD a,02
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv3: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,sv3dsp
CALL nextchr
LD a,03
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv4: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,sv4dsp
CALL nextchr
LD a,04
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv5: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,sv5dsp
CALL nextchr
LD a,05
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv6: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,sv6dsp
CALL nextchr
LD a,06
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv7: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,sv7dsp
CALL nextchr
LD a,07
LD (NoDspAct),a
JP Svonlyd
```

```
dspsv8: LD a,90h ;set dd ram
```

```
CALL LCDctrl
LD hl,sv8dsp
CALL nextchr
LD a,08
LD (NoDspAct),a
JP Svonlyd
```

```
Svonlyd: CALL readkey
LD a,(Kb_Buf)
CP clr_k
JP z,exitsv
```

```
CP dta_k
```

JP Svonlyd

Svdtonl: LD a,(NoDspAct)

CP 01

JP z,Svno\_1

CP 02

JP z,Svno\_2

CP 03

JP z,Svno\_3

CP 04

JP z,Svno\_4

CP 05

JP z,Svno\_5

CP 06

JP z,Svno\_6

CP 07

JP z,Svno\_7

CP 08

JP z,Svno\_8

JP exitsv

Svno\_1: LD de,(Sv1Buf)

CALL kGtoKch4

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Svno\_2: LD de,(Sv2Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_3: LD de,(Sv3Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_4: LD de,(Sv4Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_5: LD de,(Sv5Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_6: LD de,(Sv6Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_7: LD de,(Sv7Buf)  
 CALL kGtoKch4  
 JP chkextsv

Svno\_8: LD de,(Sv8Buf)  
 CALL kGtoKch4  
 JP chkextsv

chkextsv: CALL readkey

LD a,(Kb\_Buf)

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP clr\_k

JP z,exitsv

CP chg\_k

JP z,chgsv

JP chkextsv

chgsv: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgsv

CP 01

JP z,exitsv

JP exitsv

entchgsv: LD a,(NoDspAct)

CP 01

JP z,SvOK\_1

CP 02

JP z,SvOK\_2

CP 03

JP z,SvOK\_3

CP 04

JP z,SvOK\_4

CP 05

JP z,SvOK\_5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP 06

JP z,SvOK\_6

CP 07

JP z,SvOK\_7

CP 08

JP z,SvOK\_8

JP exitsv

SvOK\_1: CALL Mulvar

LD (Sv1Buf),hl

JP exitsv

SvOK\_2: CALL Mulvar

LD (Sv2Buf),hl

JP exitsv

SvOK\_3: CALL Mulvar

LD (Sv3Buf),hl

JP exitsv

SvOK\_4: CALL Mulvar

LD (Sv4Buf),hl

JP exitsv

SvOK\_5: CALL Mulvar

LD (Sv5Buf),hl

JP exitsv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SvOK_6: CALL Mulvar
        LD (Sv6Buf),hl
        JP exitsv
```

```
SvOK_7: CALL Mulvar
        LD (Sv7Buf),hl
        JP exitsv
```

```
SvOK_8: CALL Mulvar
        LD (Sv8Buf),hl
        JP exitsv
```

```
exitsv: LD  a,00001100b ;display control d=1,c=1,b=1
        CALL LCDctrl
        CALL putdsp1
        RET
```

```
;-----
; Routine   : pvkey
```

```
; Description : Display and change data PV
```

```
; Register  :
```

```
;-----
```

```
pvkey: CALL getdsp1
        CALL clr dsp
```

```
LD  a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,pvdsp
```

CALL nextchr

pv2ch: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitpv

CP dta\_k

JP z,Pvdtdsp

CP n01\_k

JP z,dsppv1

CP n02\_k

JP z,dsppv2

CP n03\_k

JP z,dsppv3

CP n04\_k

JP z,dsppv4

CP n05\_k

JP z,dsppv5

CP n06\_k

JP z,dsppv6

CP n07\_k

JP z,dsppv7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP n08\_k

JP z,dsppv8

JP pv2ch

Pvtdtsp: JP pv1\_4

dspallpv: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,pvpgedwn

CP up\_k

JP z,pvpgeup

CP clr\_k

JP z,exitpv

JP dspallpv

pvpgeup: LD a,(page)

CP 01

JP z,pv1\_4

JP dspallpv

pvpgedwn: LD a,(page)

CP 02

JP z,pv5\_8

JP dspallpv

pv1\_4: CALL clrdspp

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,pv1dsp
```

```
CALL nextchr
```

```
LD de,(Pv1Buf)
```

```
CALL GtoNum
```

```
LD a,0c0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,pv2dsp
```

```
CALL nextchr
```

```
LD de,(Pv2Buf)
```

```
CALL GtoNum
```

```
LD a,90h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,pv3dsp
```

```
CALL nextchr
```

```
LD de,(Pv3Buf)
```

```
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,pv4dsp
```

```
CALL nextchr
```

```
LD de,(Pv4Buf)
```

```
CALL GtoNum
```

```
LD a,02
```

```
LD (page),a
```

```
LD dspallpv
```

```

pv5_8: CALL clr dsp
        LD  a,80h      ;set dd ram
        CALL LCDctrl
        LD  hl,pv5dsp
        CALL nextchr
        LD  de,(Pv5Buf)
        CALL GtoNum

        LD  a,0c0h    ;set dd ram
        CALL LCDctrl
        LD  hl,pv6dsp
        CALL nextchr
        LD  de,(Pv6Buf)
        CALL GtoNum

        LD  a,90h    ;set dd ram
        CALL LCDctrl
        LD  hl,pv7dsp
        CALL nextchr
        LD  de,(Pv7Buf)
        CALL GtoNum

        LD  a,0d0h   ;set dd ram
        CALL LCDctrl
        LD  hl,pv8dsp
        CALL nextchr
        LD  de,(Pv8Buf)
        CALL GtoNum

```

LD (page),a

JP dspallpv

dsppv1: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv1dsp

CALL nextchr

LD a,01

LD (NoDspAct),a

JP Pvonlyd

dsppv2: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv2dsp

CALL nextchr

LD a,02

LD (NoDspAct),a

JP Pvonlyd

dsppv3: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv3dsp

CALL nextchr

LD a,03

LD (NoDspAct),a

JP Pvonlyd

dsppv4: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv4dsp

CALL nextchr

LD a,04

LD (NoDspAct),a

JP Pvonlyd

dsppv5: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv5dsp

CALL nextchr

LD a,05

LD (NoDspAct),a

JP Pvonlyd

dsppv6: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv6dsp

CALL nextchr

LD a,06

LD (NoDspAct),a

JP Pvonlyd

dsppv7: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv7dsp

CALL nextchr

LD a,07

LD (NoDspAct),a

JP Pvonlyd

dsppv8: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,pv8dsp

CALL nextchr

LD a,08

LD (NoDspAct),a

JP Pvonlyd

Pvonlyd: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitpv

CP dta\_k

JP z,Pvdtonl

JP Pvonlyd

Pvdtonl: LD a,(NoDspAct)

CP 01

JP z,Pvno\_1

CP 02

JP z,Pvno\_2

CP 03

JP z,Pvno\_3

CP 04

JP z,Pvno\_4

CP 05

JP z,Pvno\_5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP 06

JP z,Pvno\_6

CP 07

JP z,Pvno\_7

CP 08

JP z,Pvno\_8

JP exitpv

Pvno\_1: LD de,(Pv1Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_2: LD de,(Pv2Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_3: LD de,(Pv3Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_4: LD de,(Pv4Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_5: LD de,(Pv5Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_6: LD de,(Pv6Buf)

การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL kGtoKch4

JP chkextpv

Pvno\_7: LD de,(Pv7Buf)

CALL kGtoKch4

JP chkextpv

Pvno\_8: LD de,(Pv8Buf)

CALL kGtoKch4

JP chkextpv

chkextpv: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitpv

CP chg\_k

JP z,chgpv

JP chkextpv

chgpv: CALL Setvalue

LD a,(clrset)

CP 00

JP z,entchgpv

CP 01

JP z,exitpv

JP exitpv

entchgpv: LD a,(NoDspAct)

เอกสารนี้เป็นเอกสาร CP 01 หน่วยงานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP z,PvOK\_1

CP 02

JP z,PvOK\_2

CP 03

JP z,PvOK\_3

CP 04

JP z,PvOK\_4

CP 05

JP z,PvOK\_5

CP 06

JP z,PvOK\_6

CP 07

JP z,PvOK\_7

CP 08

JP z,PvOK\_8

JP exitpv

PvOK\_1: CALL Mulvar

LD (Pv1Buf),hl

JP exitpv

PvOK\_2: CALL Mulvar

LD (Pv2Buf),hl

เอกสารนี้เป็นเอกสาร **JP exitpv** ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PvOK\_3: CALL Mulvar

LD (Pv3Buf),hl

JP exitpv

PvOK\_4: CALL Mulvar

LD (Pv4Buf),hl

JP exitpv

PvOK\_5: CALL Mulvar

LD (Pv5Buf),hl

JP exitpv

PvOK\_6: CALL Mulvar

LD (Pv6Buf),hl

JP exitpv

PvOK\_7: CALL Mulvar

LD (Pv7Buf),hl

JP exitpv

PvOK\_8: CALL Mulvar

LD (Pv8Buf),hl

JP exitpv

exitpv: LD a,00001100b ;display control d=1,c=1,b=1

CALL LCDctrl

CALL putdsp1

```

-----
; Routine      : mvkey
; Description  : Display and change data MV
; Register    :
-----

```

```
mvkey: CALL getdsp1
```

```
CALL clr dsp
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl,mv dsp
```

```
CALL nextchr
```

```
mv2ch: CALL readkey
```

```
LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,exitmv
```

```
CP dta_k
```

```
JP z,Mvdt dsp
```

```
CP n01_k
```

```
JP z,dspmv1
```

```
CP n02_k
```

```
JP z,dspmv2
```

```
CP n03_k
```

JP z,dspmv3

CP n04\_k

JP z,dspmv4

CP n05\_k

JP z,dspmv5

CP n06\_k

JP z,dspmv6

CP n07\_k

JP z,dspmv7

CP n08\_k

JP z,dspmv8

JP mv2ch

Mvtdtsp: JP mv1\_4

dspallmv: CALL readkey

LD a,(Kb\_Buf)

CP down\_k

JP z,mvpgedwn

CP up\_k

JP z,mvpggeup

CP clr\_k

JP z,exitmv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP dspallmv

mvpgeup: LD a,(page)

CP 01

JP z,mv1\_4

JP dspallmv

mvpgedwn: LD a,(page)

CP 02

JP z,mv5\_8

JP dspallmv

mv1\_4: CALL clr dsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,mv1 dsp

CALL nextchr

LD de,(Mv1Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,mv2dsp

CALL nextchr

LD de,(Mv2Buf)

CALL GtoNum

LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mv3dsp

CALL nextchr

LD de,(Mv3Buf)

CALL GtoNum

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl,mv4dsp

CALL nextchr

LD de,(Mv4Buf)

CALL GtoNum

LD a,02

LD (page),a

JP dspallmv

mv5\_8: CALL clrdsp

LD a,80h ;set dd ram

CALL LCDctrl

LD hl,mv5dsp

CALL nextchr

LD de,(Mv5Buf)

CALL GtoNum

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl,mv6dsp

CALL nextchr

LD de,(Mv6Buf)

CALL GtoNum

LD a,90h ;set dd ram

```
CALL LCDctrl
LD hl,mv7dsp
CALL nextchr
LD de,(Mv7Buf)
CALL GtoNum
```

```
LD a,0d0h ;set dd ram
CALL LCDctrl
LD hl,mv8dsp
CALL nextchr
LD de,(Mv8Buf)
CALL GtoNum
```

```
LD a,01
LD (page),a
JP dspallmv
```

```
dspmv1: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mv1dsp
CALL nextchr
LD a,01
LD (NoDspAct),a
JP Mvonlyd
```

```
dspmv2: LD a,90h ;set dd ram
CALL LCDctrl
LD hl,mv2dsp
CALL nextchr
LD a,02
```

```
LD (NoDspAct),a
```

JP Mvonlyd

```
dspmv3: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD hl,mv3dsp
        CALL nextchr
        LD a,03
        LD (NoDspAct),a
        JP Mvonlyd
```

```
dspmv4: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD hl,mv4dsp
        CALL nextchr
        LD a,04
        LD (NoDspAct),a
        JP Mvonlyd
```

```
dspmv5: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD hl,mv5dsp
        CALL nextchr
        LD a,05
        LD (NoDspAct),a
        JP Mvonlyd
```

```
dspmv6: LD  a,90h      ;set dd ram
        CALL LCDctrl
        LD hl,mv6dsp
        CALL nextchr
```

LD (NoDspAct),a

JP Mvonlyd

dspm7: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mv7dsp

CALL nextchr

LD a,07

LD (NoDspAct),a

JP Mvonlyd

dspm8: LD a,90h ;set dd ram

CALL LCDctrl

LD hl,mv8dsp

CALL nextchr

LD a,08

LD (NoDspAct),a

JP Mvonlyd

Mvonlyd: CALL getdspm

LD a,99h ;set dd ram

CALL LCDctrl

mvdata: CALL readkey

LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmv

CP dta\_k

JP z,Mvdtonl

JP mvdata

Mvdtonl: LD a,(NoDspAct)

CP 01

JP z,Mvno\_1

CP 02

JP z,Mvno\_2

CP 03

JP z,Mvno\_3

CP 04

JP z,Mvno\_4

CP 05

JP z,Mvno\_5

CP 06

JP z,Mvno\_6

CP 07

JP z,Mvno\_7

CP 08

JP z,Mvno\_8

JP exitmv

Mvno\_1: LD de,(Mv1Buf)

CALL kGtoKch4

JP chkextmv

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mvno\_2: LD de,(Mv2Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_3: LD de,(Mv3Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_4: LD de,(Mv4Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_5: LD de,(Mv5Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_6: LD de,(Mv6Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_7: LD de,(Mv7Buf)  
 CALL kGtoKch4  
 JP chkextmv

Mvno\_8: LD de,(Mv8Buf)  
 CALL kGtoKch4  
 JP chkextmv

chkextmv: CALL readkey  
 LD a,(Kb\_Buf)

CP clr\_k

JP z,exitmv

CP chg\_k

JP z,chgmv

JP chkextmv

chgmv: CALL Setmv

LD a,(clrset)

CP 00

JP z,entchgmv

CP 01

JP z,exitmv

JP exitmv

entchgmv: LD a,(NoDspAct)

CP 01

JP z,MvOK\_1

CP 02

JP z,MvOK\_2

CP 03

JP z,MvOK\_3

CP 04

JP z,MvOK\_4

CP 05

JP z,MvOK\_5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CP 06

JP z,MvOK\_6

CP 07

JP z,MvOK\_7

CP 08

JP z,MvOK\_8

JP exitmv

MvOK\_1: LD hl,(MvBuff)

LD (Mv1Buf),hl

JP exitmv

MvOK\_2: LD hl,(MvBuff)

LD (Mv2Buf),hl

JP exitmv

MvOK\_3: LD hl,(MvBuff)

LD (Mv3Buf),hl

JP exitmv

MvOK\_4: LD hl,(MvBuff)

LD (Mv4Buf),hl

JP exitmv

MvOK\_5: LD hl,(MvBuff)

LD (Mv5Buf),hl

JP exitmv

```
LD (Mv6Buf),hl
```

```
JP exitmv
```

```
MvOK_7: LD hl,(MvBuff)
```

```
LD (Mv7Buf),hl
```

```
JP exitmv
```

```
MvOK_8: LD hl,(MvBuff)
```

```
LD (Mv8Buf),hl
```

```
JP exitmv
```

```
exitmv: CALL putdsp1
```

```
RET
```

```
-----
```

```
; Routine : Setmv
```

```
; Description : Set data to manipulate variable
```

```
; Register :
```

```
-----
```

```
Setmv: LD a,0
```

```
LD (clrset),a
```

```
LD (Fastmv),a
```

```
LD a,(NoDspAct)
```

JP z,SetMv\_1

CP 02

JP z,SetMv\_2

CP 03

JP z,SetMv\_3

CP 04

JP z,SetMv\_4

CP 05

JP z,SetMv\_5

CP 06

JP z,SetMv\_6

CP 07

JP z,SetMv\_7

CP 08

JP z,SetMv\_8

SetMv\_1: LD hl,(Mv1Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_2: LD hl,(Mv2Buf)

LD (MvBuff),hl

เอกสารนี้เป็นเอกสารการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JP exitsmv

SetMv\_3: LD hl,(Mv3Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_4: LD hl,(Mv4Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_5: LD hl,(Mv5Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_6: LD hl,(Mv6Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_7: LD hl,(Mv7Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

SetMv\_8: LD hl,(Mv8Buf)

LD (MvBuff),hl

CALL incdecmv

JP exitsmv

exitsmv: RET

-----

; Routine : Signfast

; Description : display sign for faster

; Register :

-----

Signfast: LD hl,Sfast

CALL nextchr

RET

-----

; Routine : Signslow

; Description : display sign for slower

; Register :

-----

Signslow: LD hl,Sslow

CALL nextchr

RET

-----

; Routine : incdecmv

; Description : inc & dec manipulate variable

; Register :

-----

incdecmv: LD a,0d0h ;set dd ram

เอกสารนี้เป็นเอกสารให้บริการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรรมใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LD a,00001100b ;display control d=1,c=1,b=1
```

```
CALL LCDctrl
```

```
LD a,(Fastmv)
```

```
CP 00
```

```
JP z,Sgnslow
```

```
CP 01
```

```
JP z,Sgnfast
```

```
indec: LD a,99h ;set dd ram
```

```
CALL LCDctrl
```

```
CALL ReadKey
```

```
LD a,(Kb_Buf)
```

```
CP clr_k
```

```
JP z,clrsetmv
```

```
CP ent_k
```

```
JP z,mvsetOK
```

```
CP up_k
```

```
JP z,incmv
```

```
CP down_k
```

```
JP z,decmv
```

JP z,fastflag

JP incdecmv

Sgnfast: CALL Signfast

JP incdec

Sgnslow: CALL Signslow

JP incdec

fastflag: LD a,(Fastmv)

CP 0

JP z,fast

CP 1

JP z,slow

JP incdecmv

fast: LD a,01

LD (Fastmv),a

JP incdecmv

slow: LD a,00

LD (Fastmv),a

JP incdecmv

incmv: LD a,(Fastmv)

CP 0

JP z,incnorm



JP incdecnv

incnorm: LD hl,(MvBuff)

INC hl

PUSH hl

XOR a

LD de,10000

SBC hl,de

POP hl

JP z,overmv

JP nc,overmv

LD (MvBuff),hl

EX de,hl

CALL putdspmv

LD a,99h ;set dd ram

CALL LCDctrl

CALL kGtoKch4

JP incdecnv

incfast: LD hl,(MvBuff)

INC hl

INC hl

INC hl

INC hl

INC hl

PUSH hl

XOR a

LD de,10000

SBC hl,de

POP hl

JP z,overmv

JP nc,overmv

LD (MvBuff),hl

EX de,hl

CALL putdspmv

LD a,99h ;set dd ram

CALL LCDctrl

CALL kGtoKch4

JP incdecmv

overmv: CALL putdspmv

LD a,99h ;set dd ram

CALL LCDctrl

LD hl,9999

LD (MvBuff),hl

EX de,hl

CALL kGtoKch4

JP incdecmv

decmv: LD a,(Fastmv)

CP 0

JP z,decnorm

CP 1

JP z,decfast

JP incdecmv

decnorm: LD hl,(MvBuff)

PUSH hl

```
LD de,00
SBC hl,de
POP hl
JP z,lessmv
JP c,lessmv
```

```
DEC hl
LD (MvBuff),hl
EX de,hl
CALL putdspmv
LD a,99h ;set dd ram
CALL LCDctrl
CALL kGtoKch4
JP incdecmv
```

```
decast: LD hl,(MvBuff)
PUSH hl
XOR a
LD de,04
SBC hl,de
POP hl
JP z,decnorm
JP c,decnorm
```

```
DEC hl
DEC hl
DEC hl
DEC hl
DEC hl
```

```
LD (MvBuff),hl
```

```
EX de,hl
```

```

CALL putdspmv
LD  a,99h      ;set dd ram
CALL LCDctrl
CALL kGtoKch4
JP  incdecmv

```

```

lessmv:  CALL putdspmv
         LD  a,99h      ;set dd ram
         CALL LCDctrl
         LD  hl,00
         LD  (MvBuff),hl
         EX  de,hl
         CALL kGtoKch4
         JP  incdecmv

```

```

clrsetmv: LD  a,01
          LD  (clrset),a

```

```

mvsetOK: RET

```

-----

```

; Routine   : getdspmv
; Description : Save MV display
; Register  :

```

-----

```

getdspmv:  PUSH af
          PUSH bc
          PUSH hl

```

```
CALL LCDctrl
```

```
LD hl, Scr1L0mv
```

```
LD b,72
```

```
WrLmv: CALL waitbusy
```

```
IN a,(Rd_Data)
```

```
LD (hl),a
```

```
INC hl
```

```
DJNZ WrLmv
```

```
POP hl
```

```
POP bc
```

```
POP af
```

```
RET
```

```
;-----  
; Routine : putdspmv
```

```
; Description : read data from Screen buffer then write to LCD
```

```
; Register : A,B,HL  
;-----
```

```
putdspmv: PUSH af
```

```
PUSH bc
```

```
PUSH hl
```

```
LD a,80h ;set dd ram
```

```
CALL LCDctrl
```

```
LD hl, Scr1L0mv
```

```
LD b,16
```

```
RdL0mv: LD a,(hl)
```

```
CALL waitbusy
```

```
OUT (Wr_Data),a
```

CALL waitbusy

INC hl

DJNZ RdL0mv

LD a,0c0h ;set dd ram

CALL LCDctrl

LD hl, Scr1L1mv

LD b,16

RdL1mv: LD a,(hl)

CALL waitbusy

OUT (Wr\_Data),a

CALL waitbusy

INC hl

DJNZ RdL1mv

LD a,90h ;set dd ram

CALL LCDctrl

LD hl, Scr1L2mv

LD b,16

RdL2mv: LD a,(hl)

CALL waitbusy

OUT (Wr\_Data),a

CALL waitbusy

INC hl

DJNZ RdL2mv

LD a,0d0h ;set dd ram

CALL LCDctrl

LD hl, Scr1L3mv

LD b,16

RdL3mv: LD a,(hl)

```
CALL waitbusy
OUT (Wr_Data),a
CALL waitbusy
INC hl
DJNZ RdL3mv
```

```
POP hl
POP bc
POP af
```

```
RET
```

```
;
; TABLE
```

```
;
numtable: DFB "0"
          DFB "1"
          DFB "2"
          DFB "3"
          DFB "4"
          DFB "5"
          DFB "6"
          DFB "7"
          DFB "8"
          DFB "9"
```

```
;
; DATA MESSAGE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

message1: DFB " K-M-I-T-L ",0ffh  
 message2: DFB " PID-CONTROLLER ",0ffh  
 message3: DFB " INS-V1.0 ",0ffh  
 message4: DFB " 1994 ",0ffh

runmess1: DFB " ",0ffh  
 runmess2: DFB " N O W ! ",0ffh  
 runmess3: DFB "Program Running.",0ffh  
 runmess4: DFB " ",0ffh

TopErr: DFB "Top. "  
 BotErr: DFB "Bottom. "  
 insErr: DFB "Full memory. "  
 BufErr: DFB "Key buffer full. "  
 JmpErr: DFB "Over line jump. "  
 chgErr: DFB "No Data. "  
 srchErr: DFB "Not found. "  
 RunErr01: DFB "END Not found."

bscmess: DFB "BSC ",0ffh  
 ldmess: DFB "LD ",0ffh  
 outmess: DFB "OUT ",0ffh  
 funcmess: DFB "Func ",0ffh  
 func01: DFB "END ",0ffh  
 func02: DFB "SUM ",0ffh  
 func03: DFB "SUB ",0ffh  
 func04: DFB "MUL ",0ffh  
 func05: DFB "DIV ",0ffh  
 func06: DFB "SQRT ",0ffh

func07: DFB "SQR ",0ffh

clrline: DFB " ",0ffh

xbotton: DFB "X",0ffh

ybotton: DFB "Y",0ffh

rbotton: DFB "R",0ffh

mvsbotton: DFB "MV",0ffh

pvsbotton: DFB "PV",0ffh

svbotton: DFB "SV",0ffh

jumpk: DFB "Jump to line "

colon: DFB " : ",0ffh

cur\_on: DFB 7fh,0ffh ;"<-"

cur\_off: DFB " ",0ffh

Sfast: DFB "Fast >> ",0ffh

Sslow: DFB "Norm. > ",0ffh

kpdsp: DFB "Prop. Gain ",0ffh

kp1dsp: DFB " Kp1 = ",0ffh

kp2dsp: DFB " Kp2 = ",0ffh

kp3dsp: DFB " Kp3 = ",0ffh

kp4dsp: DFB " Kp4 = ",0ffh

kp5dsp: DFB " Kp5 = ",0ffh

kp6dsp: DFB " Kp6 = ",0ffh

kp7dsp: DFB " Kp7 = ",0ffh

kp8dsp: DFB " Kp8 = ",0ffh

kidsp: DFB "Integral Gain ",0ffh

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ki1dsp: DFB " Ki1 = ",0ffh  
 ki2dsp: DFB " Ki2 = ",0ffh  
 ki3dsp: DFB " Ki3 = ",0ffh  
 ki4dsp: DFB " Ki4 = ",0ffh  
 ki5dsp: DFB " Ki5 = ",0ffh  
 ki6dsp: DFB " Ki6 = ",0ffh  
 ki7dsp: DFB " Ki7 = ",0ffh  
 ki8dsp: DFB " Ki8 = ",0ffh

kddsp: DFB "Diff. Gain ",0ffh

kd1dsp: DFB " Kd1 = ",0ffh  
 kd2dsp: DFB " Kd2 = ",0ffh  
 kd3dsp: DFB " Kd3 = ",0ffh  
 kd4dsp: DFB " Kd4 = ",0ffh  
 kd5dsp: DFB " Kd5 = ",0ffh  
 kd6dsp: DFB " Kd6 = ",0ffh  
 kd7dsp: DFB " Kd7 = ",0ffh  
 kd8dsp: DFB " Kd8 = ",0ffh

maxdsp: DFB "Maximum Scale ",0ffh

mx1dsp: DFB " Max1 = ",0ffh  
 mx2dsp: DFB " Max2 = ",0ffh  
 mx3dsp: DFB " Max3 = ",0ffh  
 mx4dsp: DFB " Max4 = ",0ffh  
 mx5dsp: DFB " Max5 = ",0ffh  
 mx6dsp: DFB " Max6 = ",0ffh  
 mx7dsp: DFB " Max7 = ",0ffh  
 mx8dsp: DFB " Max8 = ",0ffh

mindsp: DFB "Minimum Scale ",0ffh

mn1dsp: DFB "Min1 = ",0ffh

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mn2dsp: DFB " Min2 = ",0ffh  
 mn3dsp: DFB " Min3 = ",0ffh  
 mn4dsp: DFB " Min4 = ",0ffh  
 mn5dsp: DFB " Min5 = ",0ffh  
 mn6dsp: DFB " Min6 = ",0ffh  
 mn7dsp: DFB " Min7 = ",0ffh  
 mn8dsp: DFB " Min8 = ",0ffh  
  
 svdsp: DFB "Set Variable ",0ffh  
 sv1dsp: DFB " SV1 = ",0ffh  
 sv2dsp: DFB " SV2 = ",0ffh  
 sv3dsp: DFB " SV3 = ",0ffh  
 sv4dsp: DFB " SV4 = ",0ffh  
 sv5dsp: DFB " SV5 = ",0ffh  
 sv6dsp: DFB " SV6 = ",0ffh  
 sv7dsp: DFB " SV7 = ",0ffh  
 sv8dsp: DFB " SV8 = ",0ffh  
  
 pvdsp: DFB "Process Variable",0ffh  
 pv1dsp: DFB " PV1 = ",0ffh  
 pv2dsp: DFB " PV2 = ",0ffh  
 pv3dsp: DFB " PV3 = ",0ffh  
 pv4dsp: DFB " PV4 = ",0ffh  
 pv5dsp: DFB " PV5 = ",0ffh  
 pv6dsp: DFB " PV6 = ",0ffh  
 pv7dsp: DFB " PV7 = ",0ffh  
 pv8dsp: DFB " PV8 = ",0ffh  
  
 mvdsp: DFB "Manipulate Var.",0ffh

mv1dsp: DFB " MV1 = ",0ffh

เอกสารฉบับนี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

mv2dsp: DFB " MV2 = ",0ffh  
 mv3dsp: DFB " MV3 = ",0ffh  
 mv4dsp: DFB " MV4 = ",0ffh  
 mv5dsp: DFB " MV5 = ",0ffh  
 mv6dsp: DFB " MV6 = ",0ffh  
 mv7dsp: DFB " MV7 = ",0ffh  
 mv8dsp: DFB " MV8 = ",0ffh

clrprg: DFB "Delete Program ?"  
 srchk: DFB "Search: "  
 chgk: DFB "Change: "  
 delk: DFB "Delete line "  
 insk: DFB "Insert line "

Kp1Buf: DFB 00,00  
 Kp2Buf: DFB 00,00  
 Kp3Buf: DFB 00,00  
 Kp4Buf: DFB 00,00  
 Kp5Buf: DFB 00,00  
 Kp6Buf: DFB 00,00  
 Kp7Buf: DFB 00,00  
 Kp8Buf: DFB 00,00

Ki1Buf: DFB 00,00  
 Ki2Buf: DFB 00,00  
 Ki3Buf: DFB 00,00  
 Ki4Buf: DFB 00,00  
 Ki5Buf: DFB 00,00  
 Ki6Buf: DFB 00,00

Ki7Buf: DFB 00,00

Ki8Buf: DFB 00,00

Kd1Buf: DFB 00,00

Kd2Buf: DFB 00,00

Kd3Buf: DFB 00,00

Kd4Buf: DFB 00,00

Kd5Buf: DFB 00,00

Kd6Buf: DFB 00,00

Kd7Buf: DFB 00,00

Kd8Buf: DFB 00,00

Max1Buf: DFB 00,00

Max2Buf: DFB 00,00

Max3Buf: DFB 00,00

Max4Buf: DFB 00,00

Max5Buf: DFB 00,00

Max6Buf: DFB 00,00

Max7Buf: DFB 00,00

Max8Buf: DFB 00,00

Min1Buf: DFB 00,00

Min2Buf: DFB 00,00

Min3Buf: DFB 00,00

Min4Buf: DFB 00,00

Min5Buf: DFB 00,00

Min6Buf: DFB 00,00

Min7Buf: DFB 00,00

Min8Buf: DFB 00,00

X1Buf: DFB 00,00

X2Buf: DFB 00,00

เอกสาร X2Buf: DFB 00,00 สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

X3Buf: DFB 00,00

X4Buf: DFB 00,00

X5Buf: DFB 00,00

X6Buf: DFB 00,00

X7Buf: DFB 00,00

X8Buf: DFB 00,00

Y1Buf: DFB 00,00

Y2Buf: DFB 00,00

Y3Buf: DFB 00,00

Y4Buf: DFB 00,00

Y5Buf: DFB 00,00

Y6Buf: DFB 00,00

Y7Buf: DFB 00,00

Y8Buf: DFB 00,00

Sv1Buf: DFB 00,00

Sv2Buf: DFB 00,00

Sv3Buf: DFB 00,00

Sv4Buf: DFB 00,00

Sv5Buf: DFB 00,00

Sv6Buf: DFB 00,00

Sv7Buf: DFB 00,00

Sv8Buf: DFB 00,00

Pv1Buf: DFB 00,00

Pv2Buf: DFB 00,00

Pv3Buf: DFB 00,00

Pv4Buf: DFB 00,00

Pv5Buf: DFB 00,00

Pv6Buf: DFB 00,00

Pv7Buf: DFB 00,00

Pv8Buf: DFB 00,00

Mv1Buf: DFB 00,00

Mv2Buf: DFB 00,00

Mv3Buf: DFB 00,00

Mv4Buf: DFB 00,00

Mv5Buf: DFB 00,00

Mv6Buf: DFB 00,00

Mv7Buf: DFB 00,00

Mv8Buf: DFB 00,00

MvBuff: DFS 2

Fastmv: DFS 1

; DEFINE SPACE

result: DFS 6

Kb\_Buf: DFS 1

Buff4Key: DFS 4

firstchr: DFS 1

secondch: DFS 1

thirdchr: DFS 1

fourchr: DFS 1

Top\_Block: DFS 2

Cur\_Addr: DFS 2

Line\_Add: DFS 2

Line\_No: DFS 1

เอกสารนี้ Cur\_Addr: DFS 2 ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Jump\_To: DFS 2  
 jumpbcd: DFS 3  
 jumpsum: DFS 2  
 SrchAddr: DFS 2

;+++++

Buffer: DFS 16  
 getin: DFS 2  
 getout: DFS 2  
 EndProg: DFS 2  
 MaxLiUse: DFS 2

Scr1L0: DFS 16  
 Scr1L2: DFS 16  
 nouse1: DFS 8  
 Scr1L1: DFS 16  
 Scr1L3: DFS 16

Scr1L0mv: DFS 16  
 Scr1L2mv: DFS 16  
 nouse1mv: DFS 8  
 Scr1L1mv: DFS 16  
 Scr1L3mv: DFS 16

GainBcd: DFS 8  
 KchgBuf: DFS 4  
 NoDspAct: DFS 1

page: DFS 1  
 clrset: DFS 1

Pointexe: DFS 2

Pointpro: DFS 2

Input1: DFS 2

Input2: DFS 2

Ldflag: DFS 1

Output: DFS 2

org 9F93H

intser: DFB low Int\_Serv,high Int\_Serv

END





**ภาคผนวก ข รายละเอียดขอมูล**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



## Z84011/C11

### PARALLEL I/O CONTROLLER

#### FEATURES

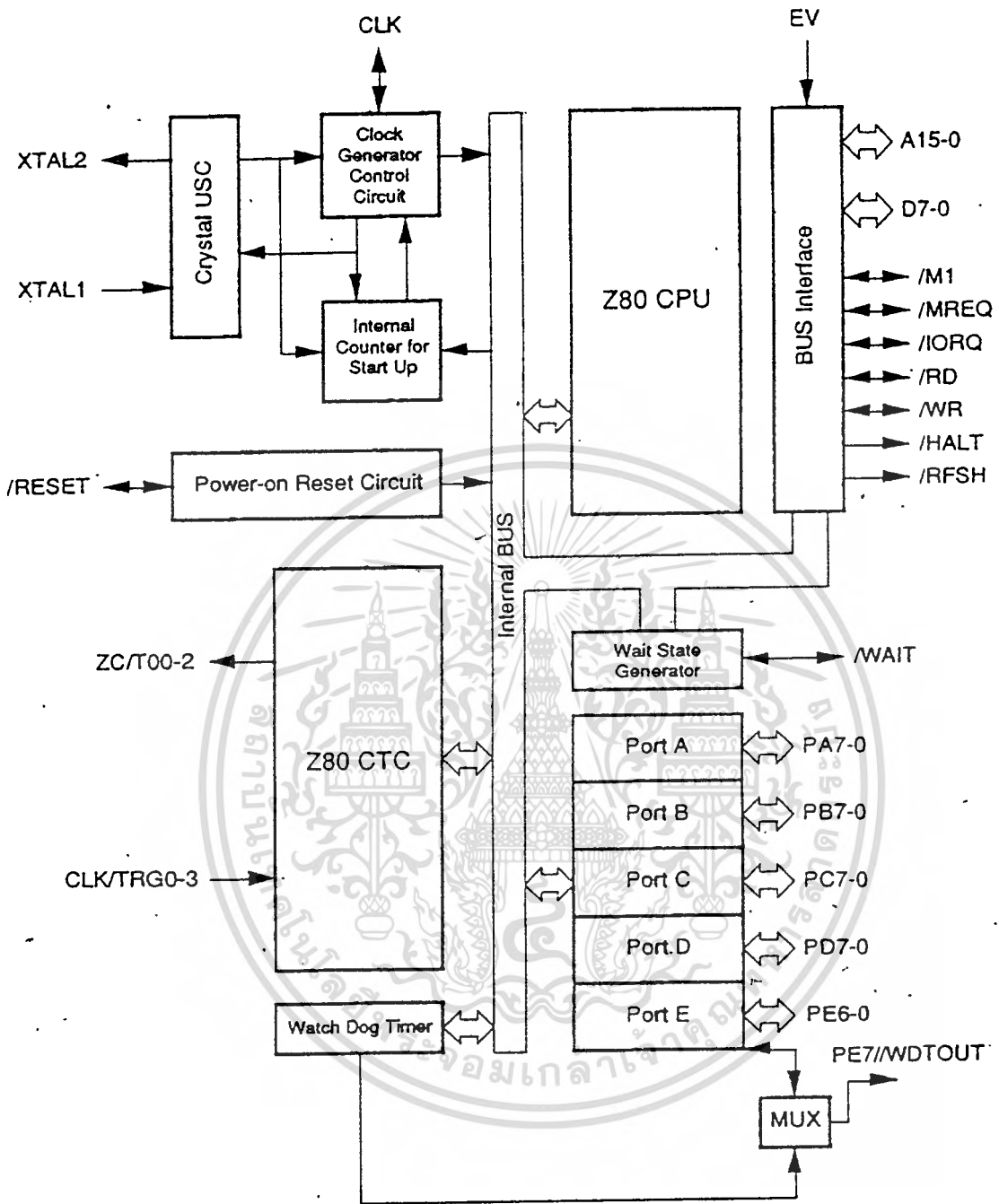
- Z84C00: Z80 CPU with CGC, Z84C30 Z80 CTC, five 8-bit parallel ports.
- High speed operation (6/10 MHz).
- Low power consumption in four operation modes:
  - .45 mA Typ. (Run mode).
  - 6 mA Typ. (Idle1 mode).
  - 9 mA Typ. (Idle2 mode; Not applicable to Z84011).
  - 1  $\mu$ A Typ. (Stop mode).
- Wide operational voltage range (5V  $\pm$  10%).
- TTL/CMOS compatible.
- Z84011 features:
  - Z84C00 Z80 CPU.
  - On-chip four channel Counter Timer Controller (Z80 CTC).
  - Built-in Clock Generator Controller (CGC).
  - Five 8-bit parallel ports.
  - 100-pin QFP Package.
  - Noise filter to CLK/TRG inputs of the Z80 CTC.
- Z84C11 features:
  - All Z84011 features.
  - Support of Idle 2 Mode.
  - Built-in Watch Dog Timer (WDT).
  - Power-on Reset and Reset Extension.
  - Wait State Generator.
  - Simplified EV Mode Selection.
  - Crystal Divide-by-One Option.
  - External Clock Input Option.

#### GENERAL DESCRIPTION

The Z84011 and Z84C11 Parallel I/O Controllers (PIC) are CMOS 8-bit microprocessors. They are integrated with the CTC and five 8-bit parallel ports into a single 100-pin QFP (Quad Flat Pack) package. The Z84C11 is an upward compatible version of the Z84011. Figure 1(a) shows the block diagram of Z84011, and Figure 1(b) shows the block diagram of the Z84C11. Figure 2 has pin assignments for both versions. These high end superintegrated Parallel I/O Controllers are targeted for a broad range of applica-

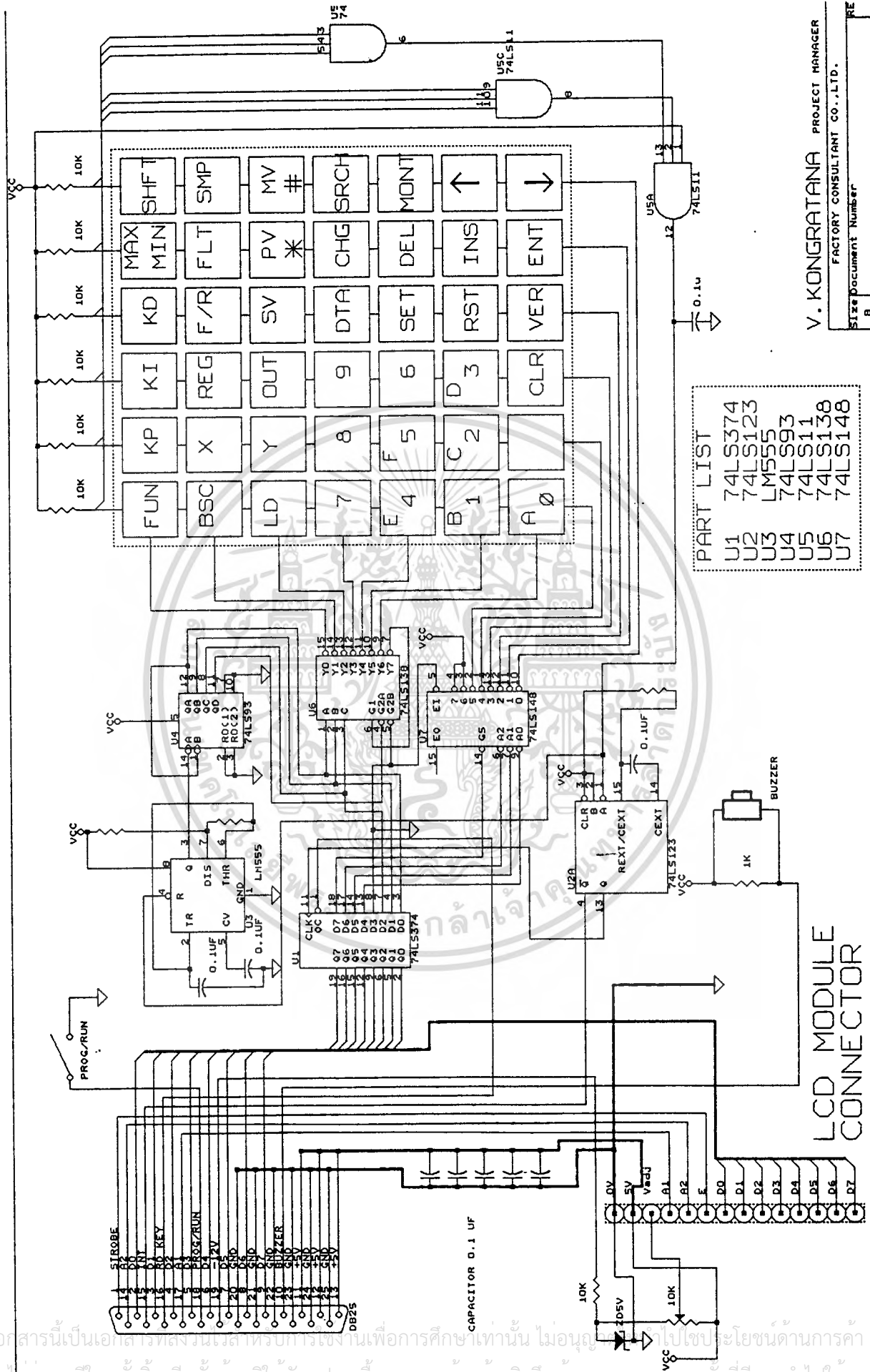
tions ranging from embedded controller to enhancement/cost reductions of existing hardware using Z80 based discrete peripherals.

Hereinafter, the word PIC on the description covering both versions (Z84C11 and Z84011) is used. Use Z84C11 on the description which applies only to the Z84C11, and use Z84011 which applies only to the Z84011.



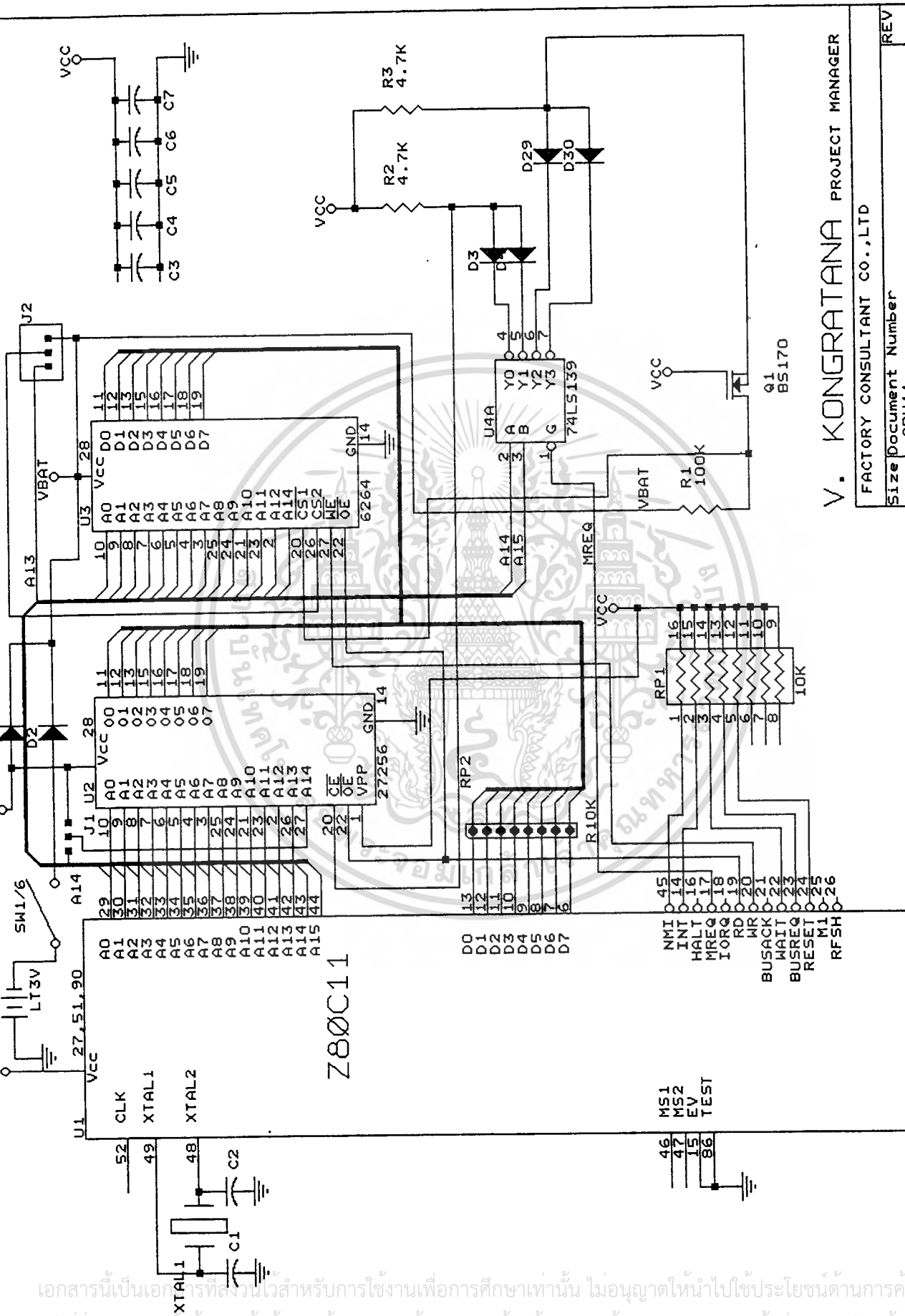
(b) Z84C11 Functional Block Diagram

Figure 1. PIC Functional Block Diagram (Continued)



PART LIST

U1	74LS374
U2	74LS123
U3	LM555
U4	74LS93
U5	74LS11
U6	74LS138
U7	74LS148

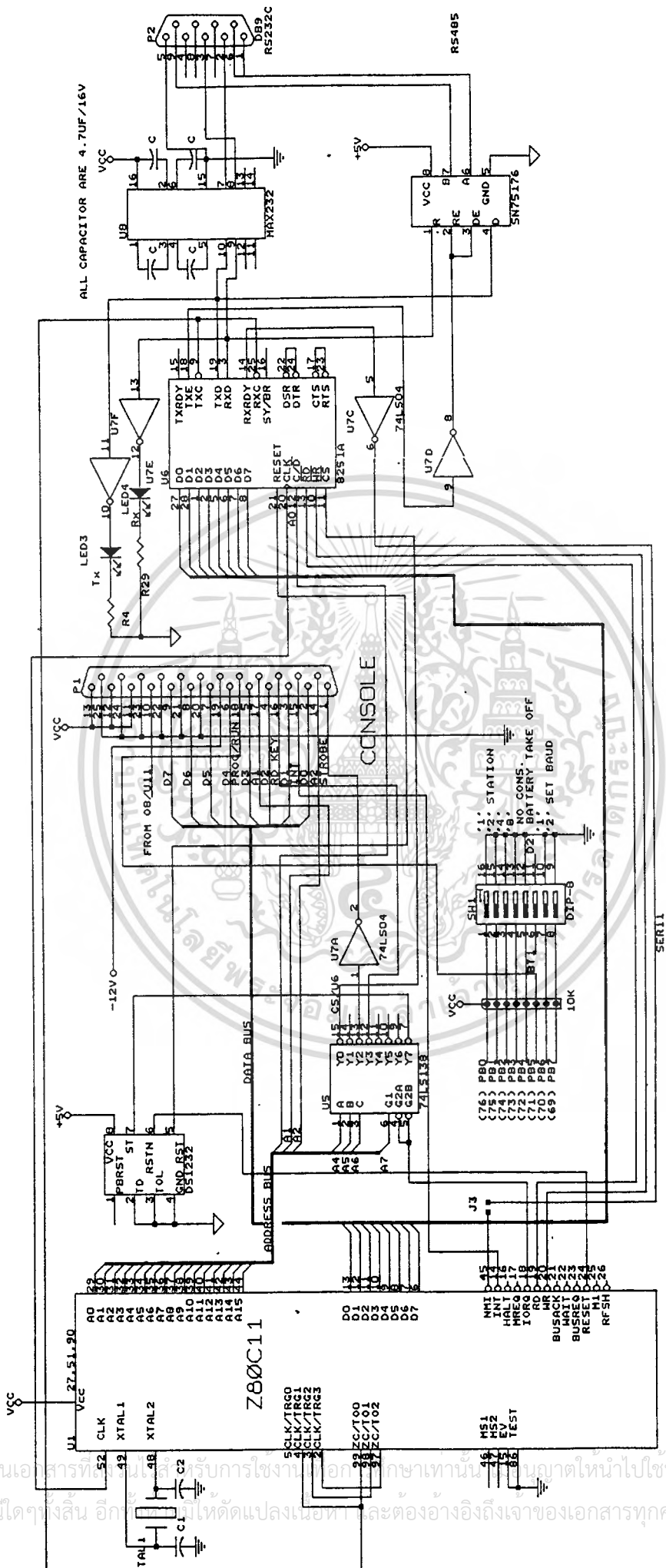


V. KONGRATANA PROJECT MANAGER

FACTORY CONSULTANT CO.,LTD

Size Document Number  
A CPU11

Date: October 16, 1994 Sheet of



SER11

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการให้บริการเชิงพาณิชย์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี โดยสงวนลิขสิทธิ์ในเนื้อหาและข้อมูลทั้งหมด ไม่สามารถนำข้อมูลไปใช้โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัยฯ



## บรรณานุกรม

วิบูลย์ ชื่นแขก. ไมโครโปรเซสเซอร์. สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ZILOG. INTELLIGENT PERIPHERAL CONTROLLERS

Ernest F. Johnson. Automatic Process Control. Mc Graw Hill



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้