บทที่ 4

โครงสร้างทางซอฟต์แวร์

งานของตัวควบคุม ดังที่ได้กล่าวในบทที่ 2 มี 2 ลักษณะ คือ งานที่มีข้อจำกัดทางเวลามาก (Hard-time constraint) และงานที่มีข้อจำกัดทางเวลารองลงมา (Soft-time constraint) สำหรับงานในลักษณะแรก ได้แก่ งานการคำนวณการควบคุมแบบ PID ซึ่งเป็นการทำงานในลักษณะของงานเวลาจริง (Real-time task) โดยในทุกเวลาการสุ่มตัวควบคุมจะทำงานดังกล่าว ส่วนงานในลักษณะหลัง เช่น งานการรับข้อมูลจากผู้ใช้ งานการแสดงผล เป็นต้น เป็นงานที่สามารถรอได้ โดยอาจทำหลังจากการคำนวณการควบคุมแบบ PID เสร็จก่อน

งานการคำนวณแบบ PID ซึ่งเป็นงานที่มีข้อจำกัดทางเวลามาก เนื่องจากตัวควบคุมจะทำการคำนวณในทุกๆ เวลาการสุ่ม โปรแกรมของการทำงานดังกล่าว จึงได้จัดให้เป็นโปรแกรมอินเทอร์รัพต์ของตัวตั้งเวลาการสุ่ม ส่วนโปรแกรมของงานที่เหลือได้จัดไว้ในส่วนของโปรแกรมหลัก

4.1 โปรแกรมอินเทอร์รัพต์ (Interrupt program)

โปรแกรมอินเทอร์รัพต์เป็นโปรแกรมที่เกิดจากตัวตั้งเวลาที่ใช้มอดูลของ PCA ของซีพียู ได้แก่ ตัวตั้งเวลาการสุ่ม (Sampling timer) และตัวตั้งเวลาจับเหตุการณ์ (Event timer) สำหรับโปรแกรมอินเทอร์รัพต์ที่เกิดจากตัวตั้งเวลาการสุ่ม จะทำงานเกี่ยวกับการคำนวณการควบคุม PID ส่วนโปรแกรมของตัวตั้งเวลาจับเหตุการณ์จะทำงานเมื่อมีการกดปุ่มนานกว่า 1 วินาที เพื่อตรวจสถานะการกดปุ่มของผู้ใช้

เพื่อป้องกันความผิดพลาด (Error) ที่อาจเกิดขึ้นในขณะทำงานอันเนื่องมาจากส่วนประมวลผลทำงานผิดพลาด โปรแกรมอินเทอร์รัพต์จะถูกตรวจสอบระยะเวลาการทำงานไม่ให้เกินเวลาการสุ่มด้วยตัวตั้งเวลาแบบวอชด็อก (Watchdog timer)

สำหรับผังงานของโปรแกรมแสดงดังรูปที่ 4.1

4.2 โปรแกรมหลัก (Main program)

ดังที่ได้กล่าวในบทที่ 2 โปรแกรมหลักแบ่งออกได้เป็น 2 แบบ

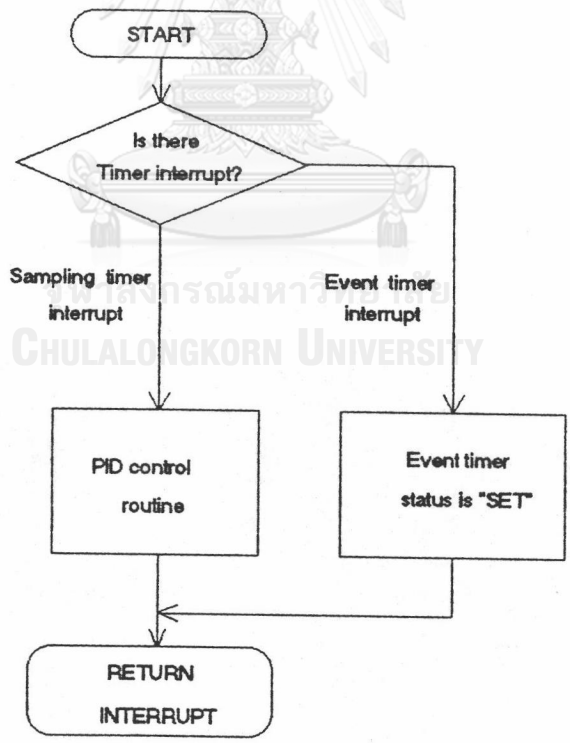
4.2.1 โปรแกรมที่ทำงานเพียงครั้งเดียว

ประกอบด้วยการทำงานในการกำหนดค่าเริ่มต้นของตัวแปร กำหนดค่าเริ่มต้นของฮาร์ดแวร์ที่โปรแกรมได้ เช่น ฮาร์ดแวร์ของส่วนแสดงผล และจะทำการวินิจฉัย (Diagonostic) ฮาร์ดแวร์ที่ใช้การตรวจสอบในตอนเริ่มเปิดเครื่อง เช่น การตรวจสอบแรม (RAM) นอกจากนี้แล้วในตอนต้นของโปรแกรม ถ้ามีการรายงานการทำงานผิดพลาดโดยการจับเวลาของวอชด็อกไทม์เมอร์ เครื่องจะหยุดการทำงานและรายงานผลการผิดพลาดที่ส่วนแสดงผล

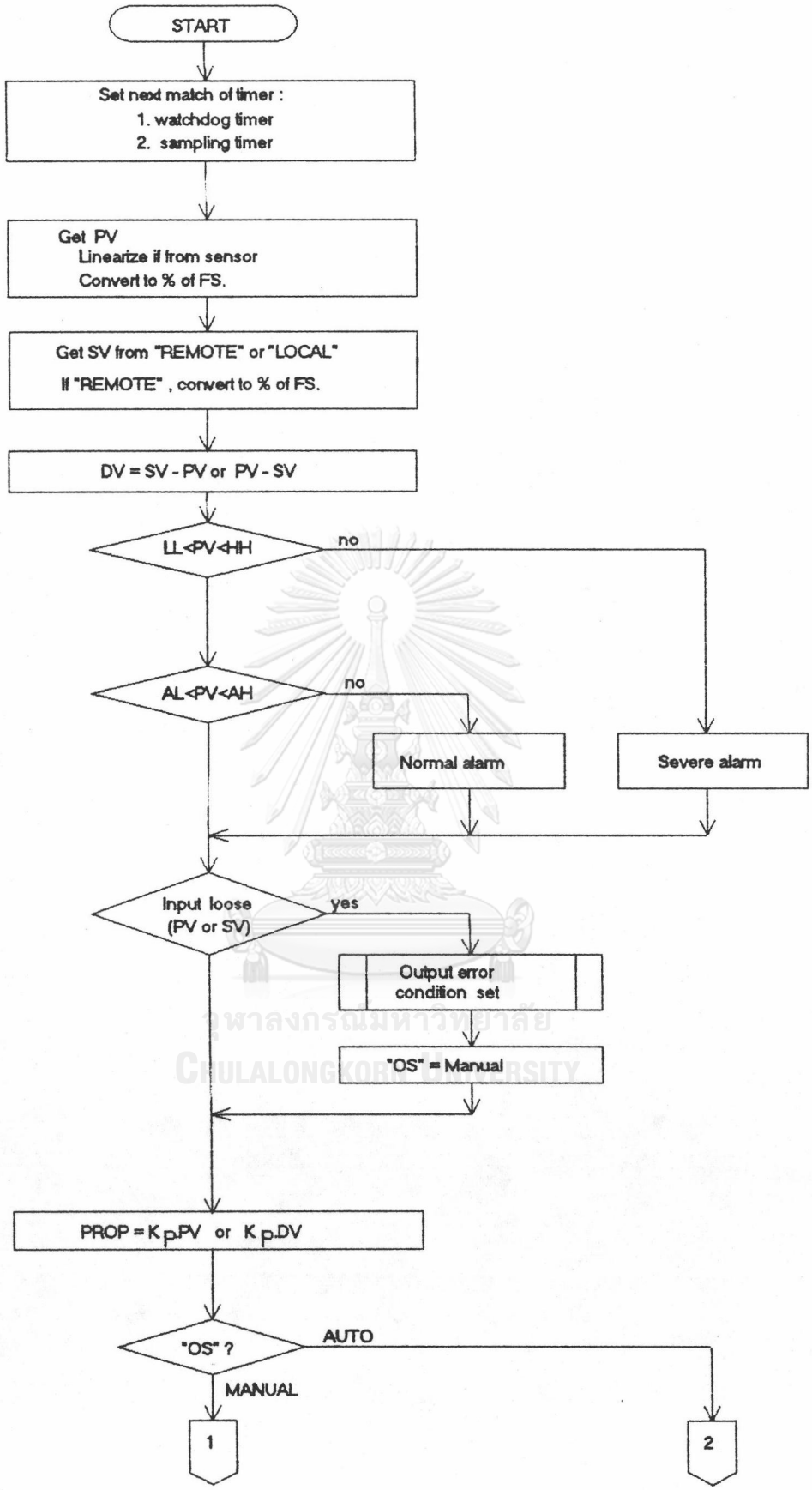
4.2.2 โปรแกรมที่ทำงานแบบวนรอบ

ประกอบด้วยโปรแกรมบริการผู้ใช้ในส่วนของการรับข้อมูลและการแสดงผลจากแผงหน้าปัด โปรแกรมการวินิจฉัยฮาร์ดแวร์ในกรณีที่ฮาร์ดแวร์นั้นๆ ต้องการตรวจสอบความถูกต้องก่อนการใช้งาน เช่น วงจรแปลงผันแอนะลอกเป็นดิจิตอล (A/D Converter) เป็นต้น และโปรแกรมการคำนวณสัมประสิทธิ์ของ PID

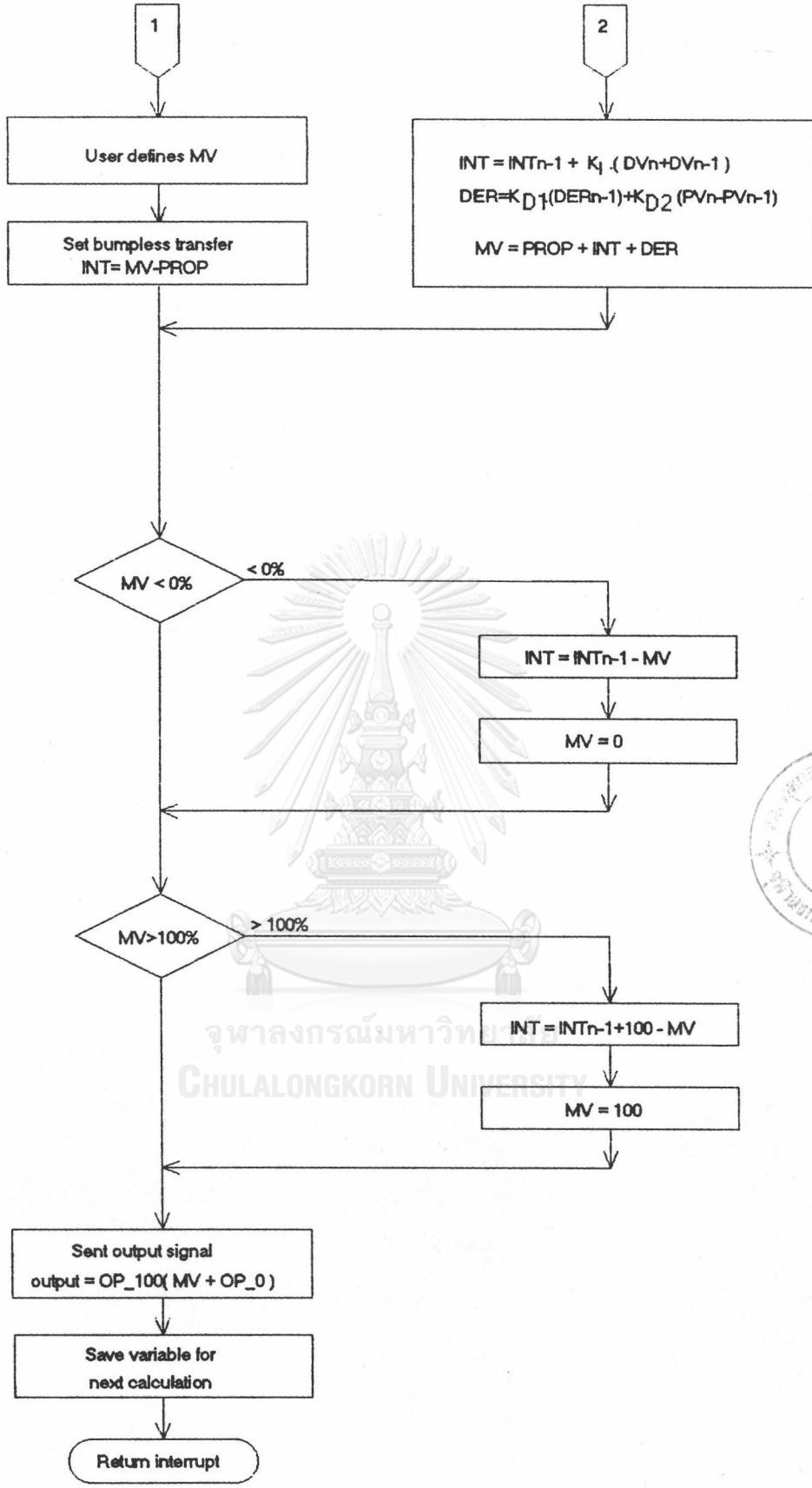
โครงสร้างการทำงานของโปรแกรมหลัก แสดงได้ดังรูปที่ 4.2



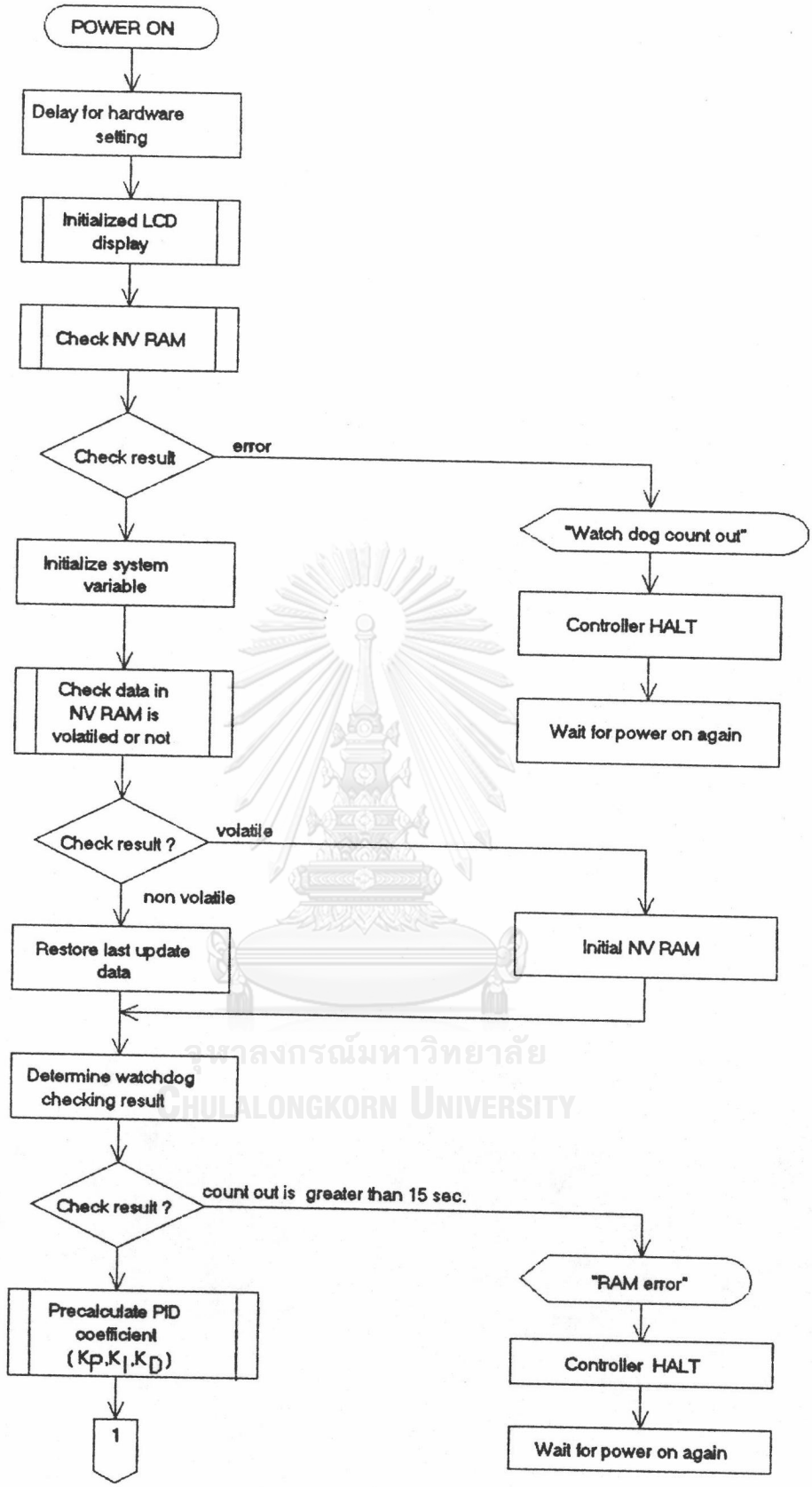
รูปที่ 4.1 ก) แสดงผังงานของโปรแกรมอินเทอร์รัพต์



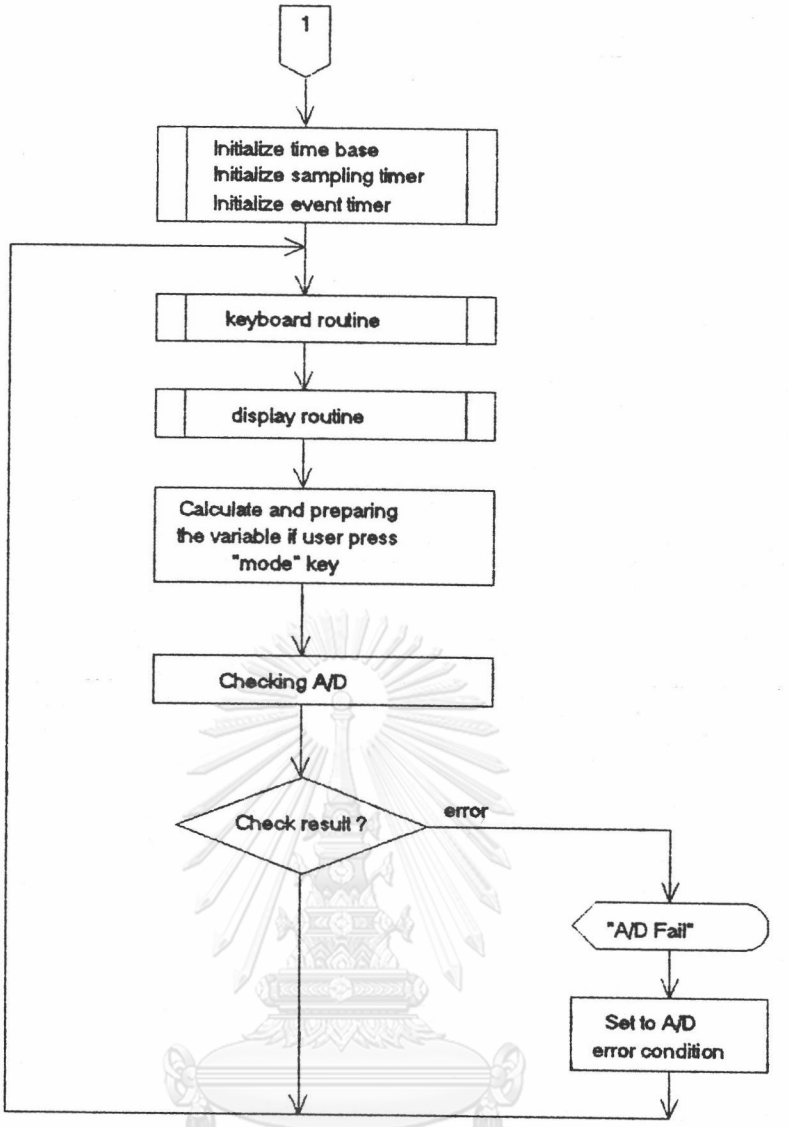
รูปที่ 4.1 ข) แสดงผังงานของการคำนวณการควบคุมแบบ PID (PID control routine)



รูปที่ 4.1 ข) แสดงผังงานของการคำนวณการควบคุมแบบ PID (ต่อ)



รูปที่ 4.2 แสดงผังงานของโปรแกรมหลัก



รูปที่ 4.2 แสดงผังงานของโปรแกรมหลัก (ต่อ)

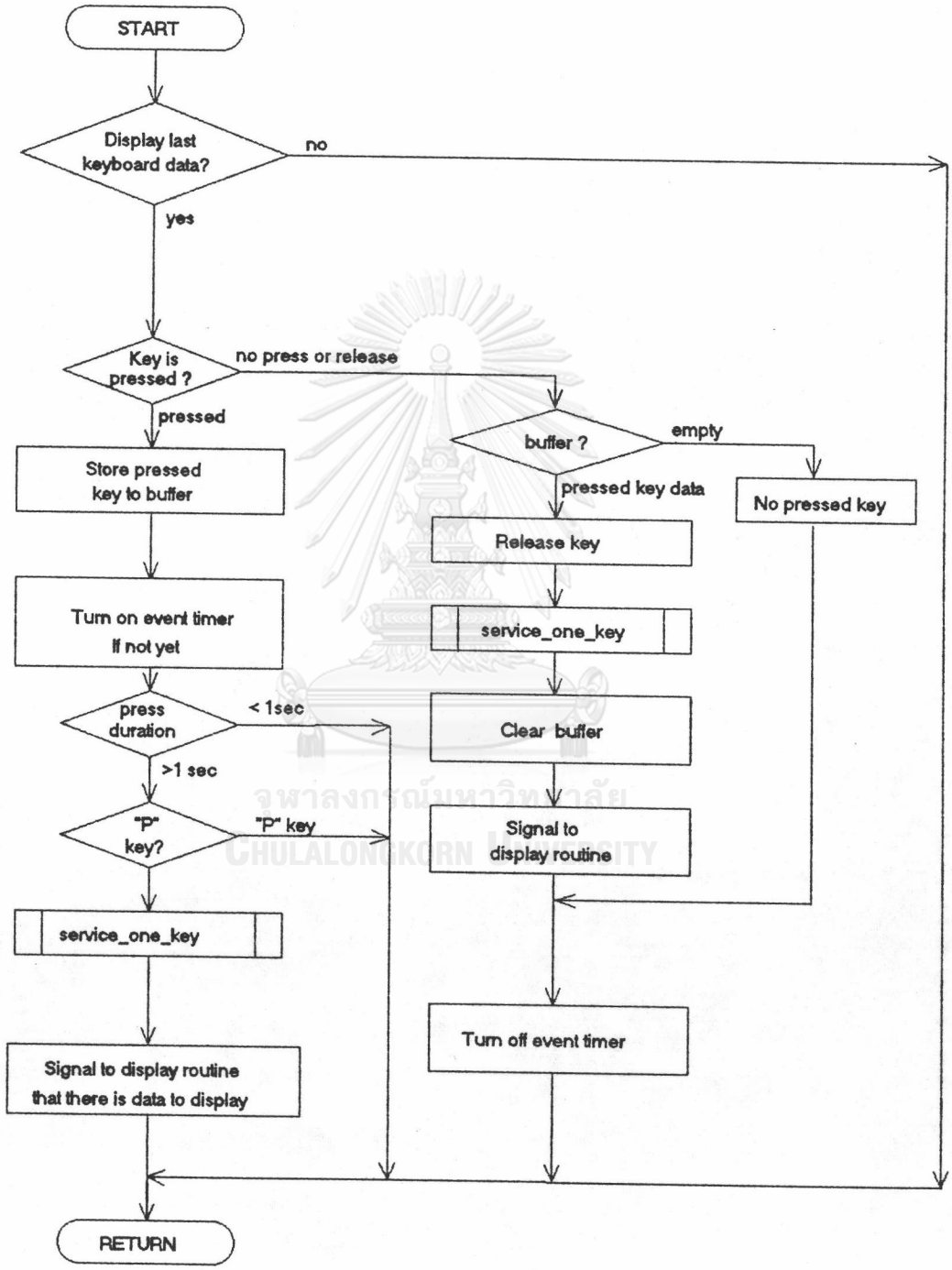
4.3 รายละเอียดของโปรแกรมต่างๆ

ในส่วนนี้ จะอธิบายถึงรายละเอียดของโปรแกรมย่อยในแต่ละส่วน ทั้งส่วนของโปรแกรมอินเทอร์รัพต์ และโปรแกรมหลัก

4.3.1 โปรแกรมการรับข้อมูลปุ่มกดจากผู้ใช้ทางแผงหน้าปัด "Keyboard routine"

เป็นโปรแกรมที่มีหน้าที่ในการรับข้อมูลการกดปุ่มจากผู้ใช้ โดยการกดปุ่มแบ่งได้เป็น 2 แบบคือ กดปล่อยทันที และกดค้างเกิน 1 วินาที เพื่อเรียกใช้ฟังก์ชันปกติ และฟังก์ชันพิเศษของปุ่มตามลำดับ สำหรับการแยกความแตกต่างของลักษณะการกดปุ่มจะใช้ PCA มอดูล 1 เป็นตัวตั้งเวลาเพื่อจับเวลาเหตุการณ์ (Event timer) ดังกล่าว ในการแบ่งแยก

ความแตกต่าง โดยเมื่อมีการกดปุ่มจะทำให้ตัว Event timer ทำงาน และถ้ามีการกดปุ่มครบเวลา 1 วินาที สถานะของ Event timer จะเปิด (On) ค้าง เมื่อมีการตรวจสอบการกดปุ่มอีกครั้งก็จะทำให้มีการทำงานตามปุ่มที่กดทันที ผังงานของโปรแกรมแสดงได้ดังรูปที่ 4.3



รูปที่ 4.3 แสดงผังงานของส่วนการรับข้อมูลจากปุ่มกด

โปรแกรมรับข้อมูลการกดปุ่มนี้จะเรียกใช้โปรแกรมการบริการปุ่มกด คือ "Service_one_key" ซึ่งรายละเอียดจะได้กล่าวถึงต่อไป

4.3.2 โปรแกรมการบริการปุ่มกด "Service_one_key"

เป็นโปรแกรมย่อย (Sub routine) ในส่วนของโปรแกรมการกดปุ่ม งานของโปรแกรมจะเป็นการถอดรหัสว่ากดปุ่มใด และทำการกดในลักษณะกดค้างเกิน 1 วินาที หรือไม่ และทำการบริการการทำงานของปุ่มกดนั้นๆ ผังงานของโปรแกรมแสดงได้ดังรูปที่ 4.4

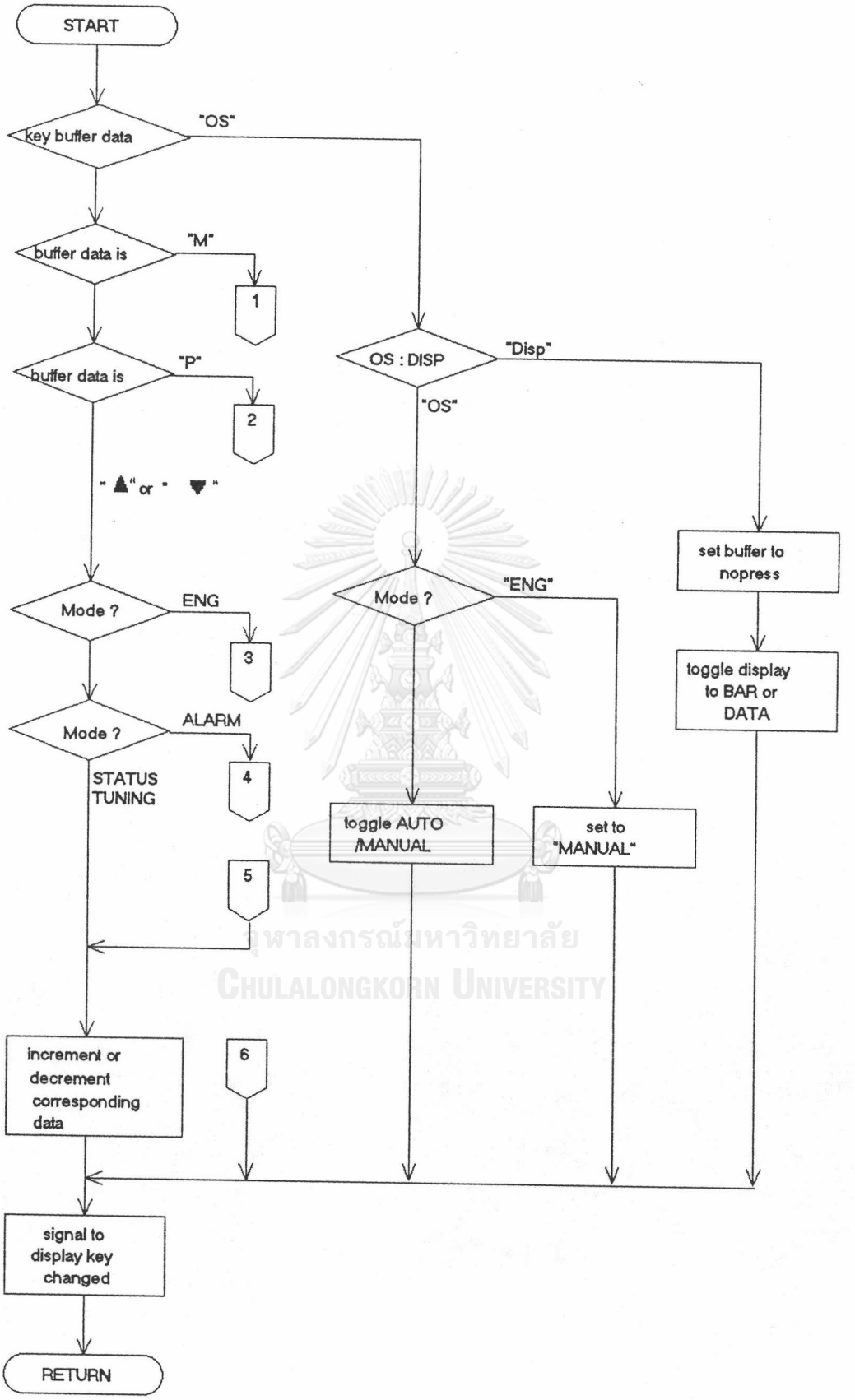
4.3.3 โปรแกรมการแสดงผล "Display routine"

เป็นโปรแกรมในส่วนของโปรแกรมหลักแบบวนรอบที่จัดการเกี่ยวกับการแสดงผลทั้ง 2 แบบ ได้แก่ การแสดงค่าและแก้ไขข้อมูล (Data and Modified display) และแสดงกราฟแท่ง (Bar graph display) ของตัวแปรในโปรเซสโดยการทำงานของโปรแกรมจะทำการตรวจสอบว่ามีข้อมูลที่จะแสดงผลหรือไม่ซึ่งข้อมูลดังกล่าวจะได้จากการกดปุ่มเพื่อตอบสนอง (response) ผู้ใช้หรือได้จากการโปรแกรมอินเทอร์รัพท์เพื่อที่จะปรับค่า (Update) ตัวแปรในโปรเซสในแบบ (Real-time) จากนั้นจะทำการตรวจสอบดูว่ามีข้อมูลที่เกี่ยวกับความผิดพลาด แล้วจะเลือกการทำงานตามลักษณะการแสดงผลนั้นๆ การทำงานของโปรแกรมการแสดงผลแสดงได้ด้วยผังงานในรูปที่ 4.5

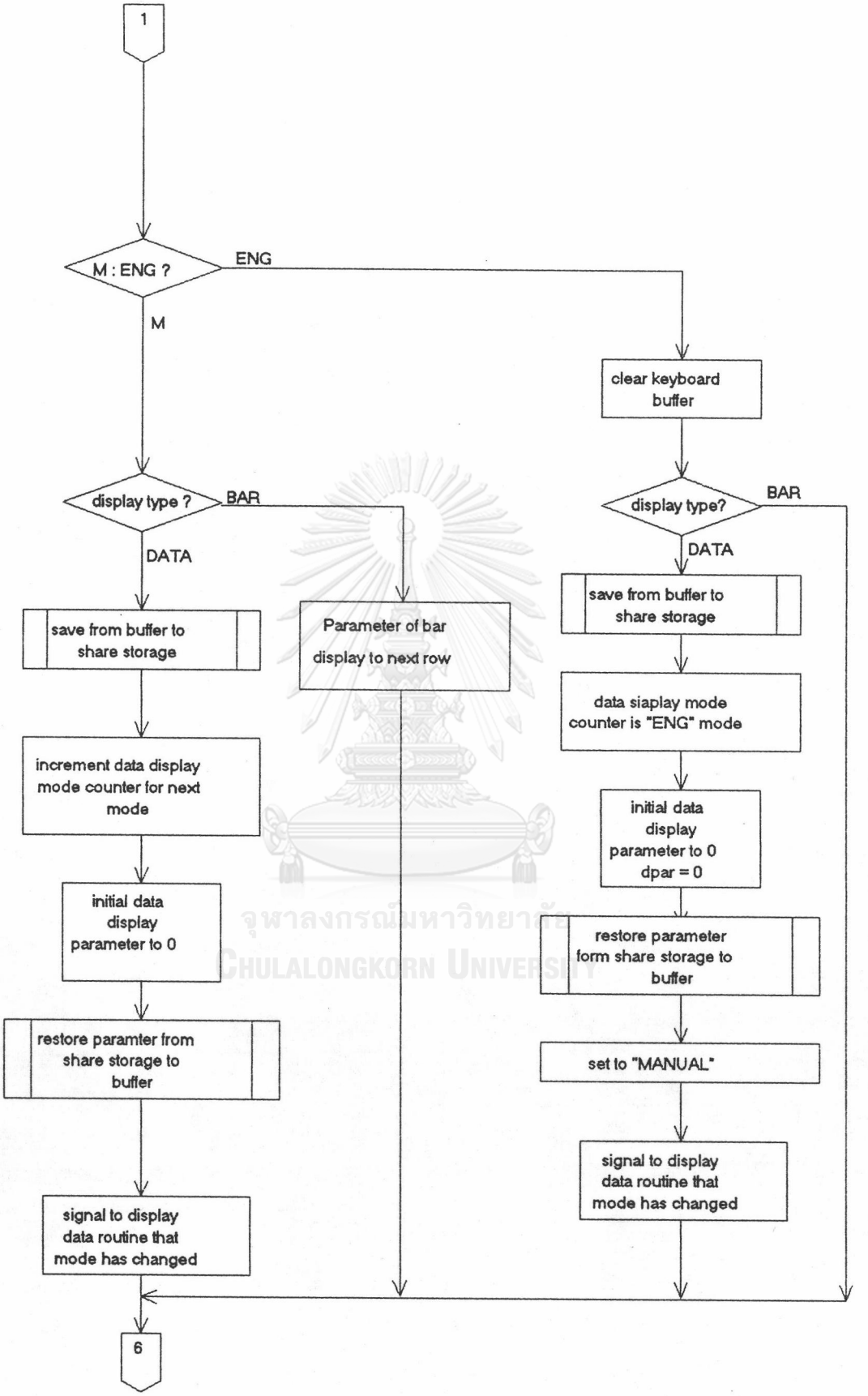
โปรแกรมการแสดงผลจะทำการเรียกโปรแกรมย่อย ได้แก่ โปรแกรมการแสดงผลแบบแสดงค่าและแก้ไขข้อมูล "Display_data" และโปรแกรมการแสดงผลแบบกราฟแท่ง "Display_bar"

4.3.4 โปรแกรมการแสดงผลแบบแสดงค่าและแก้ไขข้อมูล "Display_data"

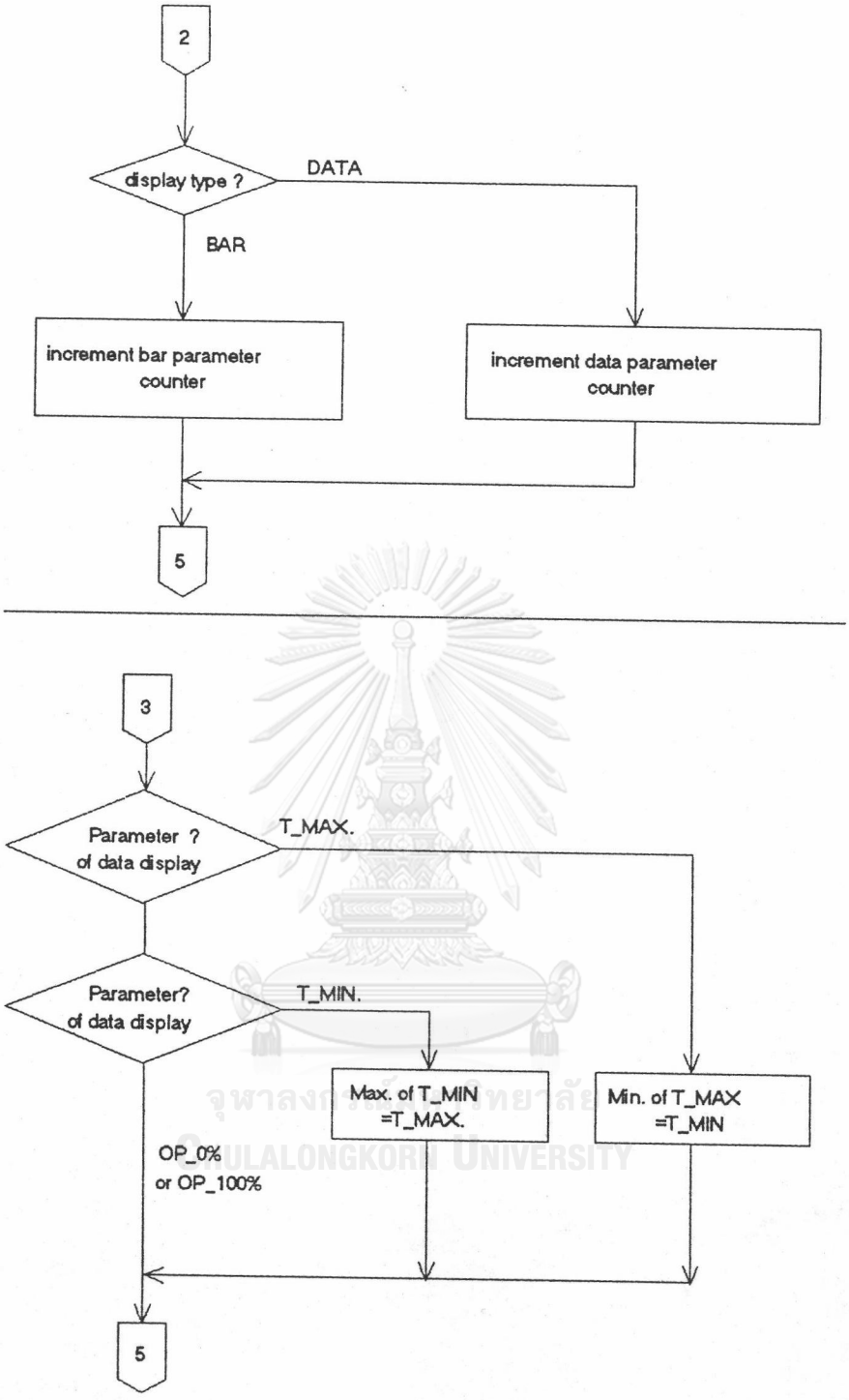
เป็นโปรแกรมที่ทำหน้าที่ในการแสดงผล โดยแสดงค่าข้อมูลจริงของตัวแปรในแต่ละโหมด เพื่อดูค่าหรือทำการแก้ไข ผังงานการทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.6



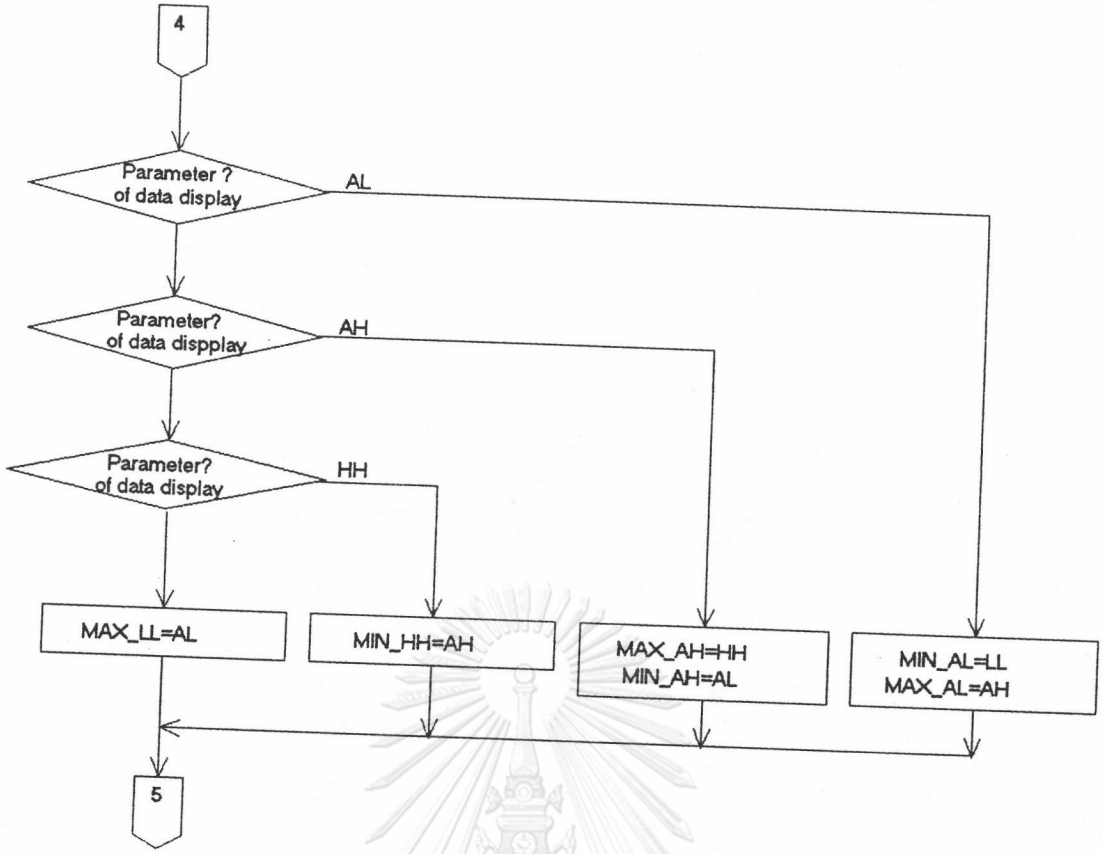
รูปที่ 4.4 แสดงผังงานของโปรแกรมการบริการปุ่มกด



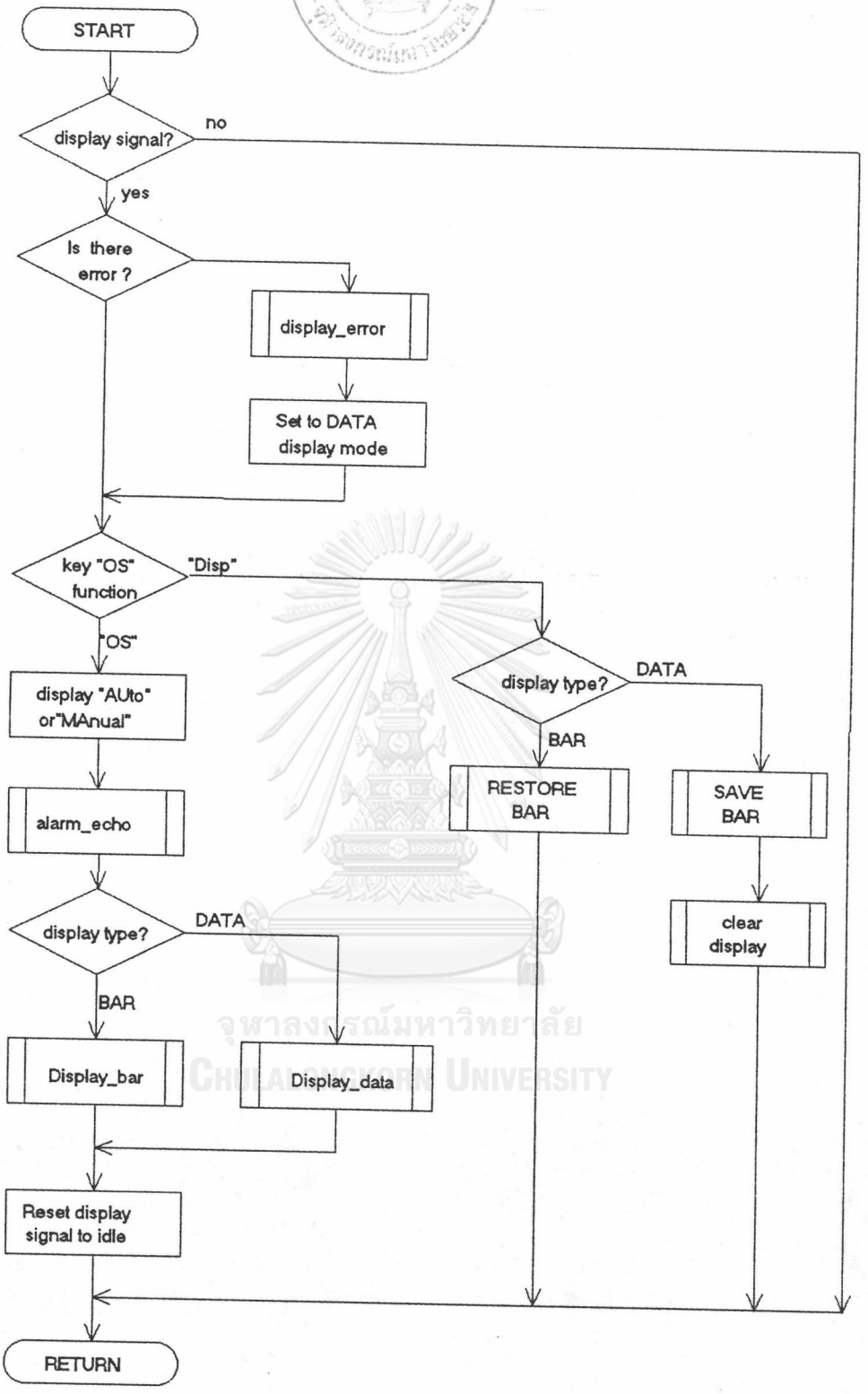
รูปที่ 4.4 แสดงผังงานของโปรแกรมการบริการปุ่มกด (ต่อ)



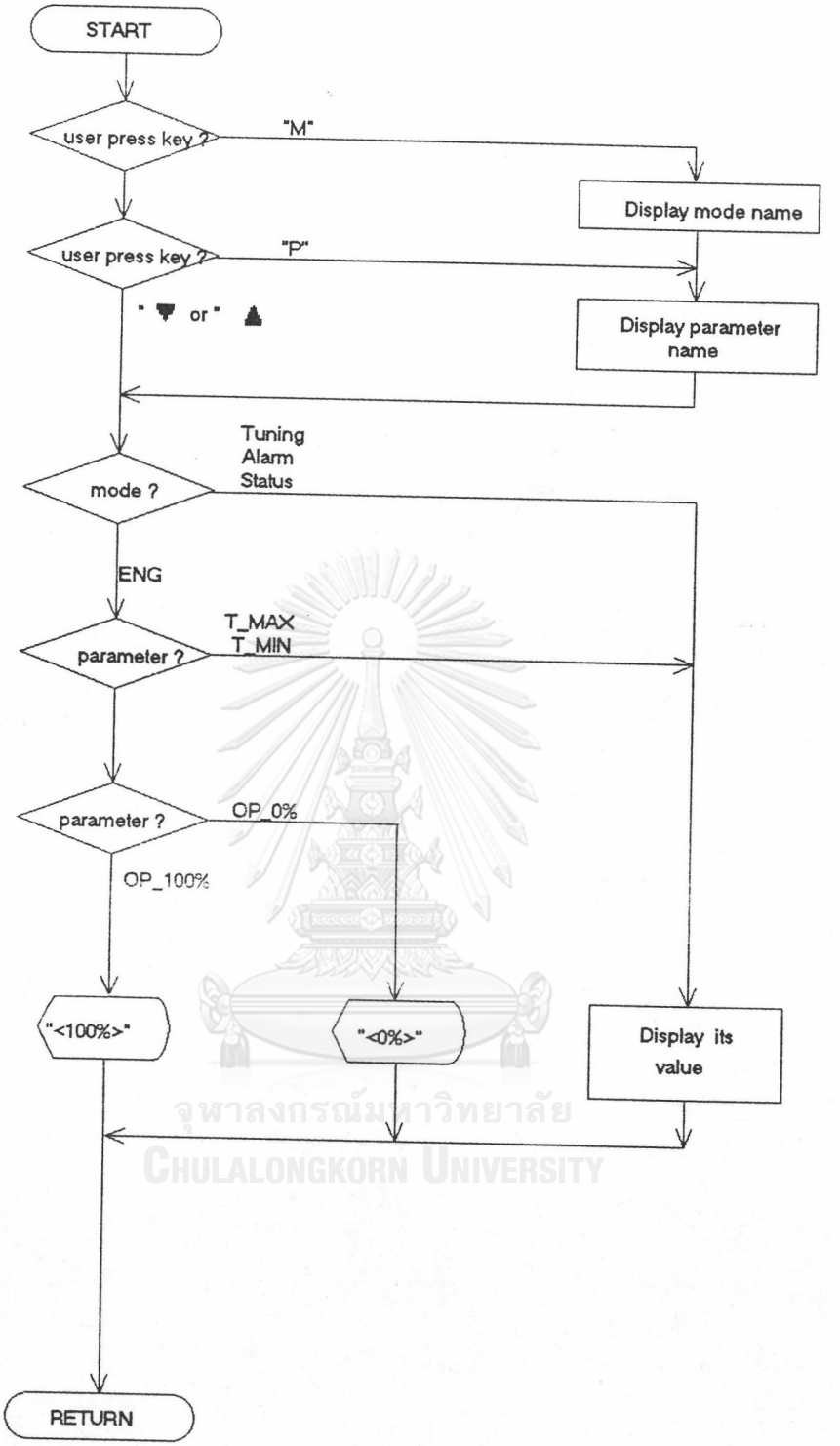
รูปที่ 4.4 แสดงผังงานของโปรแกรมการบริการปุ่มกด (ต่อ)



รูปที่ 4.4 แสดงผังงานของโปรแกรมการบริการปุ่มกด (ต่อ)



รูปที่ 4.5 แสดงผังงานของโปรแกรมการแสดงผล



รูปที่ 4.6 แสดงผังงานการทำงานของส่วนแสดงผลแบบแสดงค่าข้อมูล

4.3.5 โปรแกรมการแสดงผลแบบกราฟแท่ง "Display_bar"

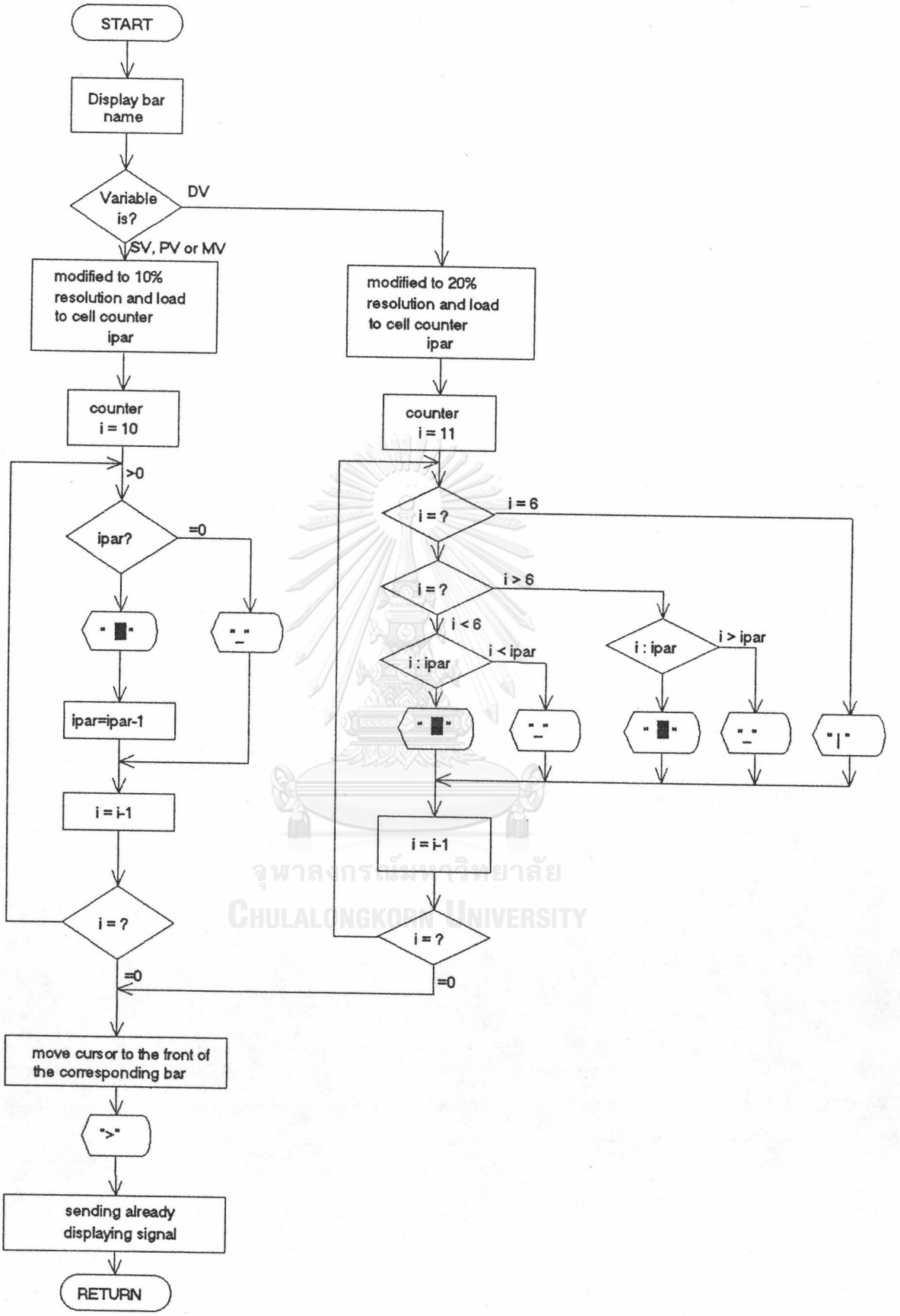
เป็นโปรแกรมที่ทำการแสดงค่าของตัวแปรในโปรเซส โดยแสดงในลักษณะของรูปกราฟแท่ง (Bar graph) การทำงานของโปรแกรมจะเริ่มจากการพิมพ์ชื่อของตัวแปร จากนั้นทำการคำนวณ โดยเริ่มจากการแปลงหาค่าจำนวนเซลล์ และจำนวนเซลล์ของกราฟแท่งใน 1 เซลล์ จะมีความละเอียดเป็น 10% ของค่าเต็มสเกล สำหรับในกรณีของ SV,PV และ MV ส่วน DV จะเป็น 20% ของค่าเต็มสเกล ในการคำนวณหาค่าจำนวนเซลล์ของ SV, PV และ MV จะหาได้จาก

$$\text{จำนวนเซลล์ของ SV, PV หรือ MV} = \frac{\text{ค่าของตัวแปร}}{10}$$

ในกรณี DV จะมีค่าทั้งบวก และลบ โดยตำแหน่งตรงกลางจะเป็นตำแหน่งที่ 6

$$\text{จำนวนเซลล์ของ DV} = \frac{\text{ค่าของตัวแปร DV}}{20} + 6$$

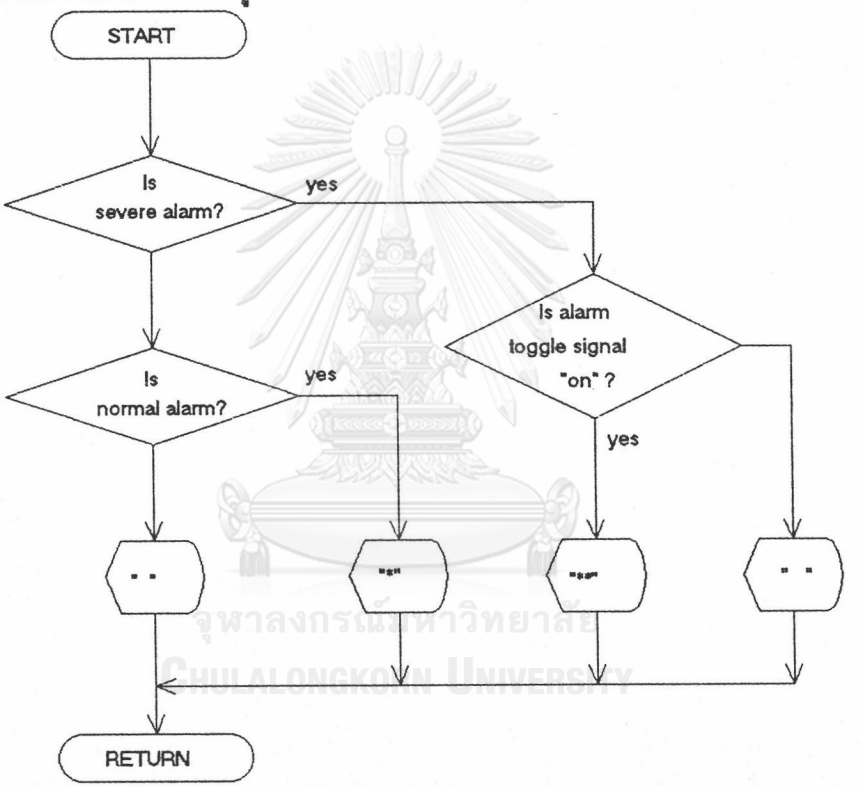
สำหรับผังงานแสดงการทำงานของการแสดงผลแบบกราฟแท่ง แสดงไว้ในรูปที่ 4.7



รูปที่ 4.7 แสดงผังงานการทำงานของส่วนแสดงผลแบบกราฟแท่ง

4.3.6 โปรแกรมแสดงข้อมูลการเตือน "Alarm_echo"

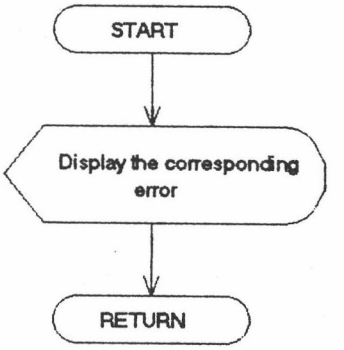
เป็นโปรแกรมย่อยในส่วนของโปรแกรมการแสดงผลเพื่อแสดงการเตือนบนส่วนแสดงผลเมื่อ PV มีค่าออกนอกช่วงกำหนดของตัวแปรในโหมด "Alarm" ในกรณีที่ออกนอกช่วงของตัวแปร AL หรือ AH จะเป็นการเตือนแบบปกติ (Normal alarm) และแสดงด้วยเครื่องหมาย "*" สำหรับกรณีเมื่อออกนอกช่วงของ LL หรือ HH จะเป็นการเตือนแบบแบบรุนแรง (Severe alarm) จะแสดงด้วยเครื่องหมาย "**" ที่กระพริบด้วยความถี่ 1 เฮิรตซ์ ผังงานของการทำงานแสดงได้ดังรูปที่ 4.8



รูปที่ 4.8 แสดงผังงานของโปรแกรมแสดงข้อมูลการเตือน

4.3.7 โปรแกรมการแสดงการทำงานผิดพลาด "Display_error"

เป็นโปรแกรมย่อยในส่วนของโปรแกรมการแสดงผล การทำงานจะทำการตรวจถอดรหัสความผิดพลาดที่ได้จากส่วนต่างๆ ของการวินิจฉัยฮาร์ดแวร์ของตัวควบคุม ผังงานการทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.9



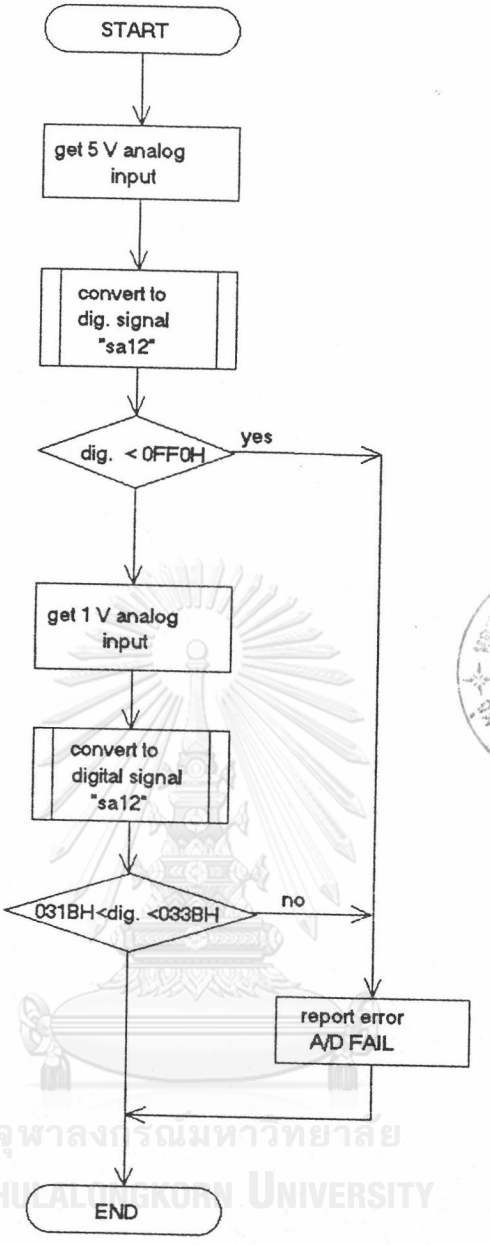
รูปที่ 4.9 แสดงผังงานของโปรแกรมการแสดงการทำงานผิดพลาด

4.3.8 โปรแกรมตรวจสอบอินพุตเกินพิสัย

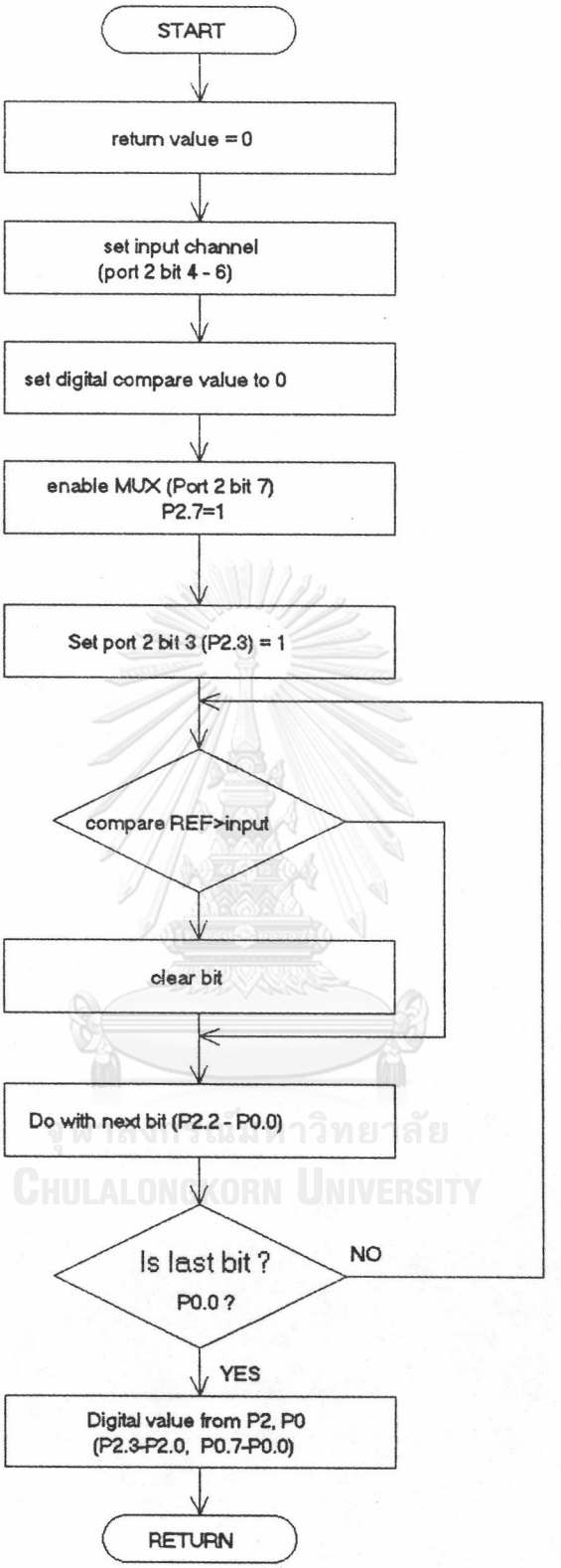
เป็นส่วนของการทำงานในโปรแกรมหลักแบบวนรอบ ในการตรวจสอบค่าความผิดพลาดของการแปลงผัน (Convert) โดยการออกแบบฮาร์ดแวร์จะมีส่วนของอินพุตภายในที่มีขนาด 5 V และ 1 V ตามลำดับ ส่วนดังกล่าวจะใช้ในการตรวจสอบการแปลงผันสัญญาณอินพุต โดยวิธีการประมาณค่าแบบ Successive ว่าถูกต้องหรือไม่ สำหรับค่า 5 V และ 1 V จะให้ค่าเชิงเลขเป็น 0FFFH และ 0333H ตามลำดับ แต่ในทางปฏิบัติค่า 1 V ซึ่งได้จากวงจรแบ่งแรงดัน จะให้ค่าเป็น 1 V โดยตรงเป็นไปได้ยาก จึงยอมให้ค่าดังกล่าวผิดพลาดอยู่ในช่วงของ +/- 0.02 V ผังงานของการทำงานแสดงได้ดังรูปที่ 4.10

4.3.9 โปรแกรมการประมาณค่าอินพุตแบบ Successive "sa12"

เป็นโปรแกรมที่ใช้ในการแปลงค่าเชิงอุปมาน (Analog) ไปเป็นค่าเชิงเลข (Digital) จากฮาร์ดแวร์ของวงจรดังกล่าวซึ่งใช้พอร์ต 0 และพอร์ต 2 ของซีพียู ในการให้ค่าเชิงเลขขนาด 12 บิต โดยพอร์ต 2 ใช้ 4 บิตล่าง (P2.3-P2.0) และพอร์ต 0 ใช้ 7 บิต (P0.7-P0.0) เพื่อทำการประมาณค่าสัญญาณอินพุต ซึ่งซีพียูสามารถสั่งแต่ละบิตของพอร์ตได้โดยตรง ทำให้ลดเวลาที่ใช้ในการประมาณค่าอินพุตลดลง [8] การทำงานของโปรแกรมแสดงได้ดังรูปที่ 4.11



รูปที่ 4.10 แสดงผังงานของการทำงานของโปรแกรมการตรวจสอบอินพุตเกินพิสัย



รูปที่ 4.11 แสดงผังงานของโปรแกรมการประมาณค่าอินพุตแบบ Successive

4.3.10 โปรแกรมการตรวจสอบ RAM "Xramchk"

โปรแกรมการตรวจสอบ RAM เป็นส่วนของโปรแกรมหลักแบบทำงานเพียงครั้งเดียวของการวินิจฉัยฮาร์ดแวร์ขณะเริ่มต้นทำงาน โดยการทดสอบจะทำการเขียนข้อมูลลงลงในหน่วยความจำที่ตำแหน่งที่ต้องการทดสอบ และอ่านข้อมูลขึ้นมา เพื่อทำการเปรียบเทียบ การทดสอบจะทดสอบเกี่ยวกับการเซต และรีเซตของแต่ละบิตของหน่วยความจำที่จะทำการทดสอบ โดยการเขียนข้อมูลเป็น 055H และ 0AAH และทำการทดสอบแบบวิธี Walking ones [9] ในท้ายที่สุดจะทดสอบเกี่ยวกับปัญหาของการอ้างแอดเดรส (Addressing problem) อัลกอริธึมของการทดสอบ RAM แสดงได้ดังรูปที่ 4.12

4.3.11 โปรแกรมการกำหนดค่าเริ่มต้นสำหรับอินพุต "Cal_sens_parameter"

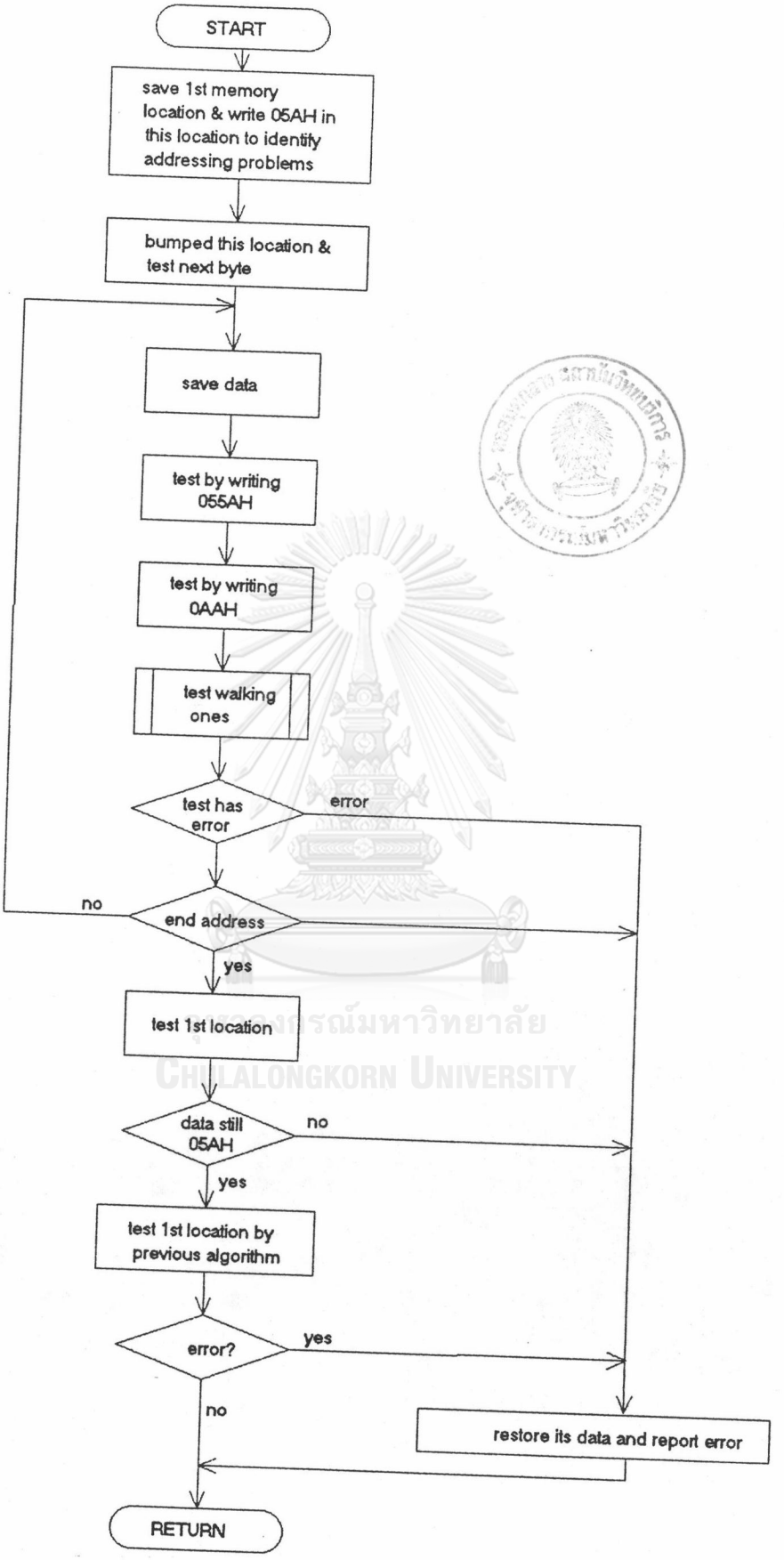
เป็นโปรแกรมย่อยในส่วนของโปรแกรมหลักแบบการทำงานเพียงครั้งเดียว เพื่อทำการคำนวณค่าสัมประสิทธิ์ของอินพุตที่เป็นเซนเซอร์ ดังแสดงในสมการ 2.15 2.16 และ 2.18 โดยโปรแกรมจะทำงานก็ต่อเมื่อมีการเปลี่ยนแปลงช่วงการทำงานของอินพุต โดยการกำหนดค่าตัวแปร T_MAX, T_MIN ในโหมด ENG ผังงานของการทำงานแสดงได้ดังรูปที่ 4.13

4.3.12 โปรแกรมการประมาณค่าข้อมูลแบบเชิงเส้น "Linear_and_convert"

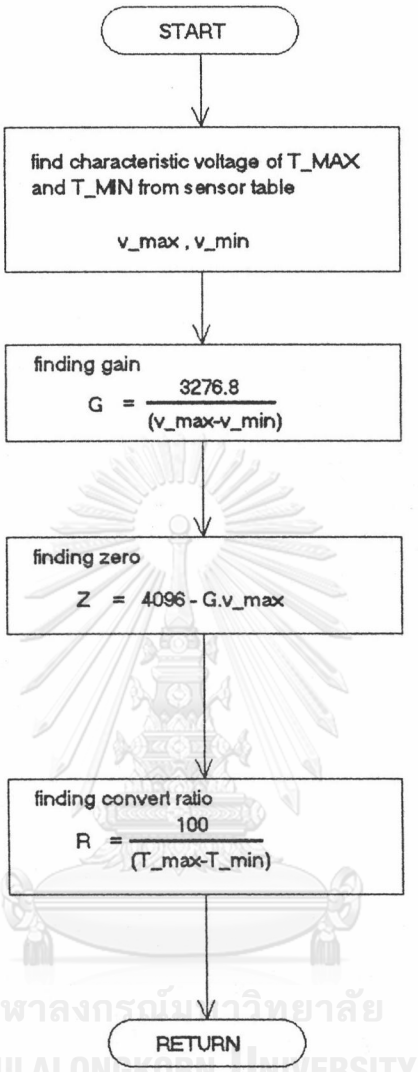
เป็นโปรแกรมย่อยที่ใช้ในการประมาณค่า PV ในกรณีที่ PV เป็นสัญญาณที่ได้จากเซนเซอร์ โดยทำการคำนวณตามสมการ 2.14 และ 2.17 ผังงานของโปรแกรม แสดงดังรูปที่ 4.14

4.3.13 โปรแกรมการคำนวณสัมประสิทธิ์ของ PID "Calcoef"

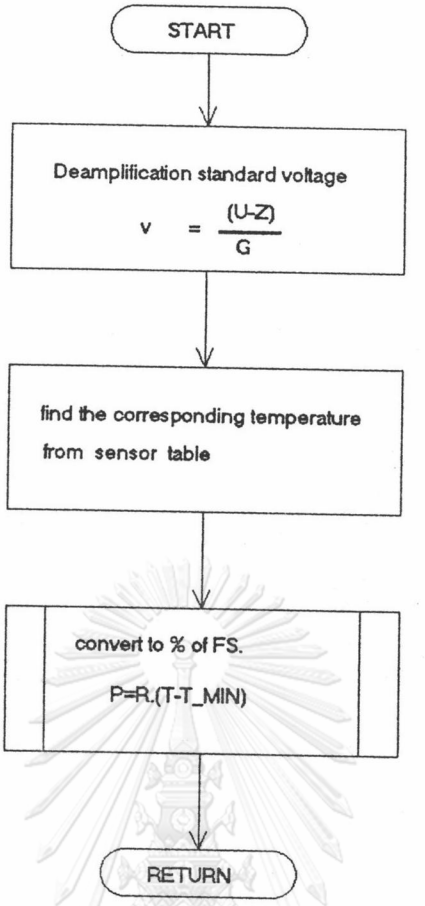
เป็นโปรแกรมที่ใช้ในการคำนวณสัมประสิทธิ์ของสมการการควบคุมแบบ PID ตามสมการ 2.10 - 2.13 ผังงานของการคำนวณแสดงดังรูปที่ 4.15



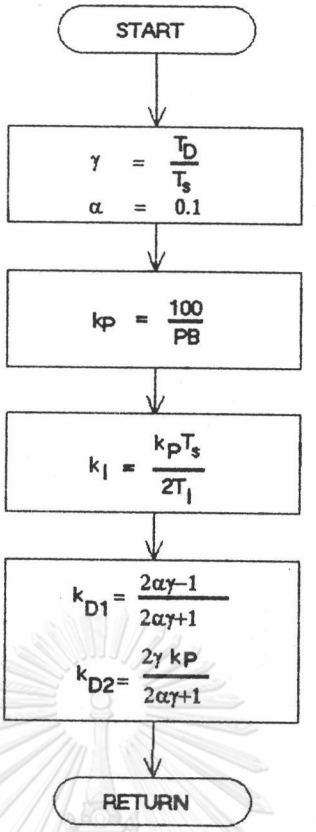
รูปที่ 4.12 แสดงผังงานของโปรแกรมการตรวจสอบ RAM



รูปที่ 4.13 แสดงผังการทำงานของโปรแกรม Cal_sens_parameter



รูปที่ 4.14 แสดงผังงานของโปรแกรม Linear_and_convert



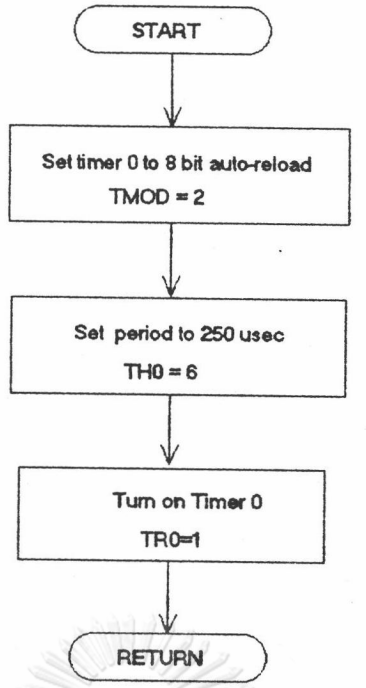
รูปที่ 4.15 แสดงผังงานของการคำนวณสัมประสิทธิ์ของ PID

4.3.14 โปรแกรมกำหนดค่าเริ่มต้นสำหรับฐานเวลา "init_timebase"

เป็นโปรแกรมย่อยในส่วนของโปรแกรมหลักแบบทำงานเพียงครั้งเดียวเพื่อกำหนดค่าฐานเวลาให้กับมอดูล PCA โดยใช้ไทม์เมอร์มอดูล 0 (Timer 0) (ซึ่งฮาร์ดแวร์ดังกล่าวเป็นฮาร์ดแวร์ที่อยู่ภายในตัวซีพียู) การทำงานของไทม์เมอร์มอดูล 0 ได้ถูกออกแบบให้อยู่ในโหมดของการทำงานเป็น 8 บิต ที่โหลดค่าโดยอัตโนมัติ (8 bit Auto-reload) โดยค่าดังกล่าวจะโหลดให้กับรีจิสเตอร์ TH หรือ TL สำหรับค่าที่โหลดให้กับรีจิสเตอร์ TH หรือ TL เนื่องจากไทม์เมอร์ 0 มีการทำงานในลักษณะของไทม์เมอร์แบบนับขึ้น (Count-up timer) ดังนั้นจะได้ว่า

ค่าข้อมูลของรีจิสเตอร์ TH หรือ TL = 256 - ค่าของฐานเวลาที่ต้องการ

ในการออกแบบเพื่อความสะดวกผู้วิจัยออกแบบให้ค่าฐานเวลาที่ต้องการเป็น 250 ไมโครวินาที จะได้ค่าข้อมูลในริจิสเตอร์ TH หรือ TL เป็น 6 สำหรับผังงานของโปรแกรมจะเป็นการกำหนดค่าเริ่มต้นให้กับรีจิสเตอร์ที่เกี่ยวข้อง ดังแสดงในรูปที่ 4.16



รูปที่ 4.16 โปรแกรมการกำหนดค่าฐานเวลาสำหรับมอดูล PCA "Init_timebase"

4.3.15 โปรแกรมกำหนดค่าเริ่มต้นสำหรับตัวตั้งเวลาการสุ่ม "Init_sampling_time"

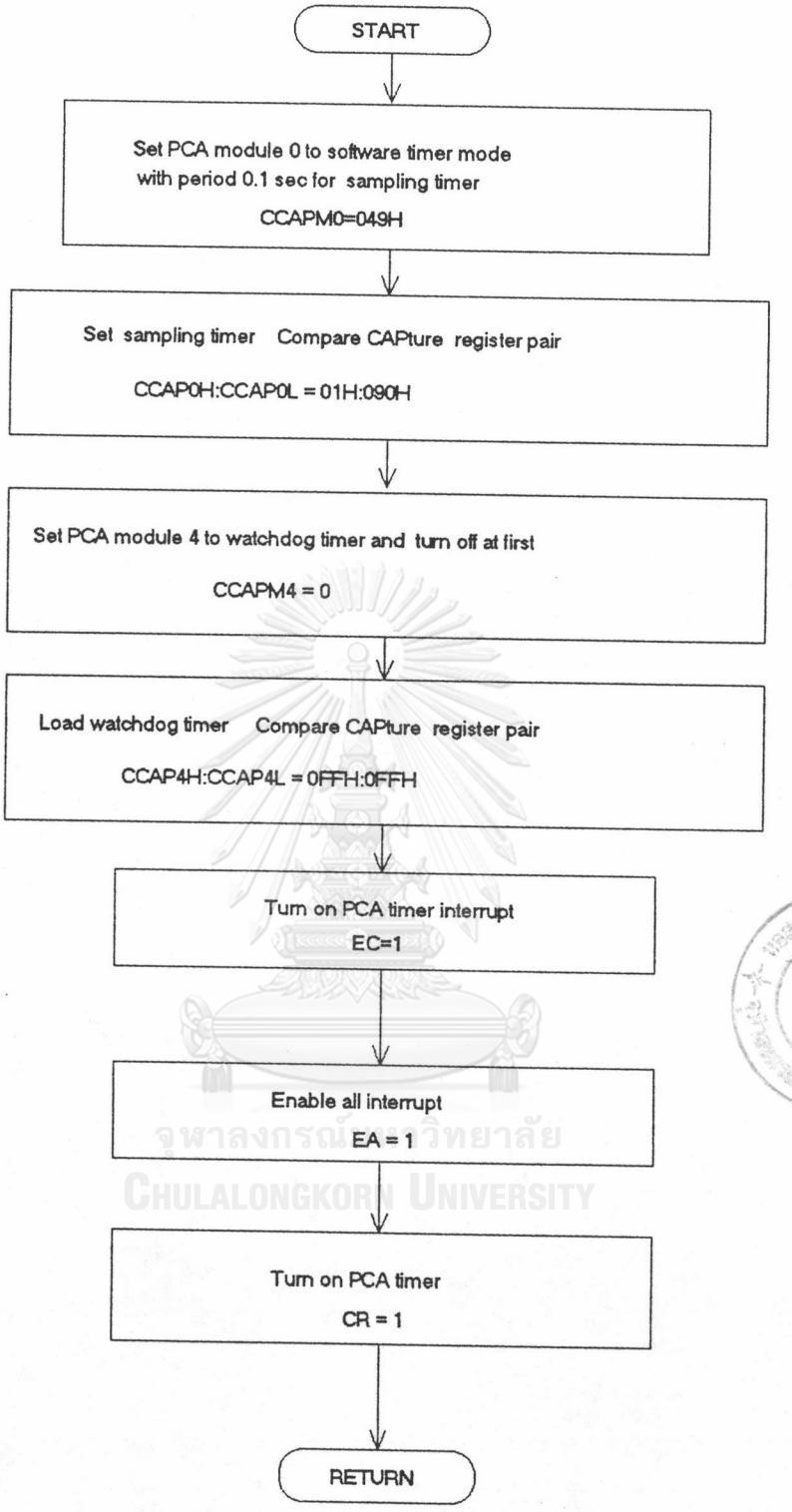
เป็นโปรแกรมย่อยในส่วนของโปรแกรมหลักที่ทำงานเพียงครั้งเดียว โดยจะทำการกำหนดค่าเริ่มต้นให้กับ PCA มอดูล 0 และ มอดูล 4 ซึ่งใช้เป็นตัวตั้งเวลาการสุ่มและตัวตั้งเวลาวอชด็อก ค่าเวลาที่ตั้งของมอดูลทั้งสองจะมีขนาด 16 บิต ที่เก็บไว้ในคู่รีจิสเตอร์ภายใน ได้แก่ CCAP0H : CCAP0L และ CCAP4H : CCAP4L สำหรับตัวตั้งเวลาการสุ่มและตัวตั้งเวลาวอชด็อกตามลำดับ ในการคำนวณหาค่าดังกล่าวเป็นดังนี้

ค่าข้อมูลของ CCAP0H,CCAP0L = ค่าเวลาสุ่ม/ฐานเวลา

สำหรับเวลาการสุ่ม 0.1 วินาที ค่าของตัวตั้งเวลาสุ่มจะได้ว่า CCAP0H : CCAP0L เป็น 01H : 090H

สำหรับรีจิสเตอร์ CCAP4H : CCAP4L จะถูกตั้งค่าให้เป็น 0FFH : 0FFH เพื่อป้องกันการทำงานของตัวตั้งเวลาวอชด็อกทำงานในครั้งแรก

ผังงานของโปรแกรมจะเป็นการกำหนดค่าเริ่มต้นให้กับรีจิสเตอร์ที่เกี่ยวข้องดังแสดงในรูปที่ 4.17



รูปที่ 4.17 ผังงานของโปรแกรมการกำหนดค่าเริ่มต้นสำหรับตัวตั้งเวลาการสุ่ม