เครื่องให้อาหารสุนัข

DOG FEEDING MACHINE

โดย

นายปณิธาน สีทน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

เครื่องให้อาหารสุนัข
DOG FEEDING MACHINE

โดย
นายปณิธาน สีทน

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2557

เครื่องให้อาหารสุนัข
DOG FEEDING MACHINE

โดย

นายปณิธาน สีทน 54010771

อาจารย์ที่ปรึกษา

รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2557

ปริญญานิพนธ์ปีการศึกษา 2557
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เรื่อง เครื่องให้อาหารสุนัข
DOG FEEDING MACHINE

ผู้จัดทำ
นายปณิธาน สีทน 54010771

..... อาจารย์ที่ปรึกษา
รศ.ดร.ยุทธพงษ์ รังสรรค์เสรี

กิตติกรรมประกาศ

การที่ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดีนั้น ก่อนอื่นต้องขอขอบคุณอาจารย์ที่ปรึกษา รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี ที่ได้ช่วยแนะแนวความคิดในการทำโครงงานนี้ ตลอดจนช่วยชี้แนะปัญหาต่างๆที่เกิดขึ้นระหว่างการทำงาน รวมทั้งคำแนะนำต่างๆ ที่เป็นประโยชน์ในการทำโครงงานฉบับนี้เป็นอย่างดี

สุดท้ายนี้ทางผู้จัดทำขอขอบพระคุณอาจารย์ทุกคนที่ให้ความรู้ รวมทั้งแนวความคิดต่างๆที่ทำให้ผู้จัดทำสามารถทำปริญญานิพนธ์สามารถทำงานจนสำเร็จลุล่องไปได้

นายปณิธาน สีทน
ผู้จัดทำ

เครื่องให้อาหารสุนัข
DOG FEEDING MACHINE

โดย นายปณิธาน สีทน 54010771

อาจารย์ที่ปรึกษา รศ.ดร. ยุทธพงษ์ รังสรรค์เสรี

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นการออกแบบการทำงานของเครื่องให้อาหารสุนัขออกแบบให้เครื่องสามารถใช้งานได้ในกรณีที่ผู้ใช้อยู่ไกลจากตัวเครื่องแต่ต้องการให้อาหารสุนัขในเวลานั้นโดยเครื่องให้อาหารจะทำงานเมื่อได้รับสัญญาณโทรศัพท์จากผู้ใช้ ซึ่งหลักการทำงานของเครื่องคือส่งสัญญาณโทรศัพท์มายัง GSM MODULE แล้ว GSM MODULE จะส่งสัญญาณไปยังไมโครคอนโทรลเลอร์เพื่อควบคุมการเปิดปิดของกล่องอาหารและหากอาหารในกล่องมีน้อยกว่าที่กำหนดโดยเซนเซอร์เซนเซอร์ก็จะส่งเอาท์พุตไปยังไมโครคอนโทรลเลอร์และไมโครคอนโทรลเลอร์ก็จะสั่งให้ GSM MODULE ส่งสัญญาณไปยังโทรศัพท์ของผู้ใช้

ABSTRACT

This project is design of Dog Feeding Machine which design for user that can use when they do not stay home and they want to feed dog. Dog Feeding Machine will start to work when it receive telephone signal from user and when user call the signal was sent to GSM MODULE then GSM MODULE transfer signal to Microcontroller for control open and close of feeding box. If dog's food in feeding box has less than control value then sensor will transfer output to Microcontroller and then GSM MODULE will send signal to user.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	III
สารบัญรูป	V

บทที่ 1

บทนำ	**เลขหน้า**
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	1

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง	**เลขหน้า**
2.1 การใช้งานพอร์ตอนุกรม RS232	2
2.2 การต่อใช้งาน RS232 / MAX232	6
2.3 ระบบ GSM	8
2.4 อินฟราเรด	9

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์	**เลขหน้า**
3.1 การออกแบบ	11
3.2 เครื่องมือที่ใช้ในการทดลอง	13
3.3 การจัดเก็บผลการทดลอง	14

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**เลขหน้า**
	4.1 การส่ง SMS เพื่อสั่งงานมอเตอร์	15
	4.2 การใช้งานเซนเซอร์อินฟราเรด	16
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**เลขหน้า**
	5.1 สรุปผล	17
	5.2 ข้อเสนอแนะ	17
บรรณานุกรม		**เลขหน้า**
ภาคผนวก ก	การต่อใช้งาน AT89C51	18
ภาคผนวก ข	CODE AT89C51	20

สารบัญรูป

รูปที่		หน้า
2.1	การต่อพอร์ต RS232	2
2.2	พอร์ตอนุกรมของPC DB9ตัวผู้และพอร์ตอนุกรมของอุปกรณ์ภายนอก DB9ตัวเมีย	2
2.3	DB9 ตัวผู้ เมื่อมองจากด้านหลัง	2
2.4	การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ NULL MODEM	3
2.5	การต่ออุปกรณ์ภายนอกผ่าน DB แบบ 3 เส้น	3
2.6	ระดับสัญญาณของ RS232	4
2.7	การรับส่งข้อมูลแบบซิงโครนัส	5
2.8	การรับส่งข้อมูลแบบอะซิงโครนัส	6
2.9	การต่อ ICMAX 232	6
2.10	วิธีต่อใช้งาน MAX232 โดยมี TTL 0-5V	7
2.11	การติดต่อกัน ระหว่าง MCU 2 ตัว	7
2.12	แสดงการต่อMAX 232 กับ MCUทั้ง 2 ฝั่ง	8
2.13	แสดงโครงสร้างระบบ GSM	9
2.14	แสงอินฟราเรด	10
3.1	บล็อกไดอะแกรมระบบ	11
3.2	โครงสร้างภายนอกกล่อง	12
3.3	เครื่องมือที่ใช้ในการทดลอง	14
3.4	การจัดเก็บผลการทดลอง	14
4.1	การส่ง SMS โดย GSM MODULE	15
4.2	เซนเซอร์อินฟราเรดกระตุ้นการส่ง SMS	16

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

สัตว์เลี้ยงก็เปรียบเสมือนเพื่อนของเรา เพราะฉะนั้นเราจึงอยากให้มันมีความสุข โดยอาหารก็เป็นปัจจัยหนึ่งที่ทำให้มันมีความสุขได้ซึ่งเราควรให้อาหารมันเป็นเวลาแต่ในปัจจุบันนี้ผู้คนต้องไปทำงานนอกบ้านและไม่มีเวลาที่จะมากลับมาให้อาหารมันเราจึงได้จัดทำกล่องให้อาหารสุนัขผ่านทางโทรศัพท์ โดยกล่องจะทำการปล่อยอาหารออกมาให้สุนัขกินเมื่อมีการส่ง SMS เข้ามาที่กล่องและเมื่อมีอาหารน้อยกว่าปริมาณที่กำหนดกล่องก็จะทำการส่ง SMS ไปยังเบอร์โทรศัพท์ที่ตั้งค่าไว้(เบอร์เจ้าของสุนัข)

1.2 วัตถุประสงค์

1) เพื่อศึกษาการใช้งานไมโครคอนโทรลเลอร์ในการควบคุมอุปกรณ์อื่น
2) เพื่อศึกษาการทำงานของ GSM Module และประยุคต์ใช้
3) เพื่อศึกษาการทำงานของเซนเซอร์อินฟราเรด
4) สามารถให้อาหารสุนัขจากที่ไกลได้

1.3 ขอบเขตของปริญญานิพนธ์

1) สามารถใช้ไมโครคอนโทรลเลอร์ควบคุมการเปิดปิดกล่องอาหารได้
2) สามารถใช้เซนเซอร์อินฟราเรดในการตรวจสอบอาหารในกล่อง
3) สามารถสั่งสั่งงานให้กล่องเปิดปิดได้ผ่านทางโทรศัพท์

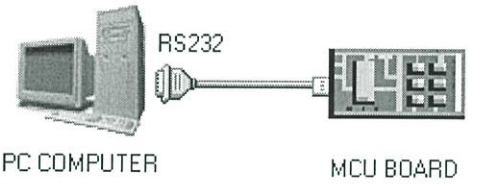
บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การใช้งานพอร์ตอนุกรม RS232

การสื่อสารแบบอนุกรมนับว่ามีความสำคัญต่อการใช้งานไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์และจอภาพของ PC เป็นอินพุตและเอาต์พุตในการติดต่อหรือควบคุมไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

- สายส่งสัญญาณ TX
- สายรับสัญญาณ RX
- และสายGND



รูปที่ 2.1การต่อพอร์ต RS232

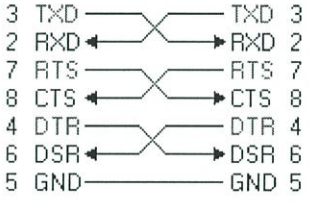
โดยปกติพอร์ตอนุกรม RS-232C จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณขึ้นอยู่กับชนิดของสายสัญญาณ, ระยะทาง,และปริมาณสัญญาณรบกวน

รูปที่ 2.2 พอร์ตอนุกรมของPC DB9ตัวผู้และพอร์ตอนุกรมของอุปกรณ์ภายนอก DB9 ตัวเมีย

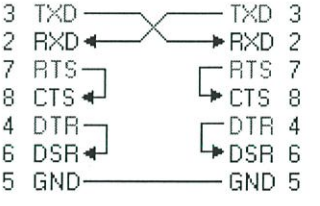
รูปที่ 2.3 DB9 ตัวผู้เมื่อมองจากด้านหลัง

Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

การเชื่อมต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์ด้วยสาย DB9



รูปที่ 2.4 การเชื่อมต่ออุปกรณ์ภายนอกผ่าน DB9 แบบ Null modem



รูปที่ 2.5 การต่ออุปกรณ์ภายนอกผ่าน DBแบบ 3 เส้น

การทำงานของขาสัญญาณ DB9

TXD เป็นขาที่ใช้ส่งข้อมูล

RXD เป็นขาที่ใช้รับข้อมูล

DTR แสดงสภาวะพอร์ตว่าเปิดใช้งานDSRตรวจสอบว่าพอร์ตที่ติดต่อด้วยเปิดอยู่

- เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย

- ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่ RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล CTSตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่

- เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF

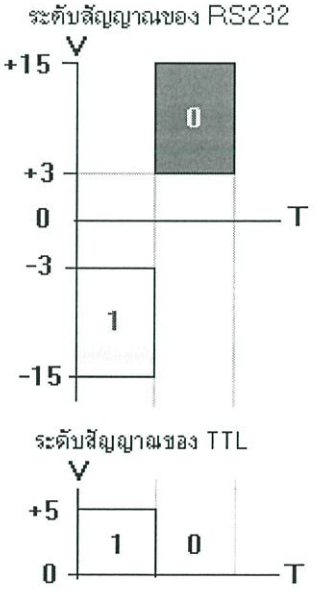
- ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่ GND ขา ground

ระดับสัญญาณของ RS232C และระดับสัญญาณของ TTL

สัญญาณรบกวนที่เกิดขึ้นในสายนำสัญญาณมักจะมีแรงดันเป็นบวกเมื่อเทียบกับกราวน์

- เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดันของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -15V ส่วนแรงดัน ของโลจิก "0" อยู่ในช่วง +3V ถึง +15V

- และเหตุที่ระดับสัญญาณของ RS232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้น ดังนั้นจึงจำเป็นจะต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL



รูปที่ 2.6 ระดับสัญญาณของ RS232

อัตราการส่งข้อมูล (Baud rate)

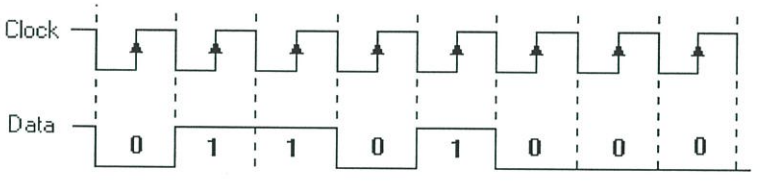
- คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิทต่อวินาทีเช่น 300 1,200 2,400 4,800 9,600 14,400 19,200 38,400 56,000 เป็นต้น
- การเลือกอัตราการส่งข้อมูลขึ้นอยู่กับชนิดของสายสัญญาณ ระยะทาง และปริมาณสัญญาณรบกวน

รูปแบบการสื่อสารแบบอนุกรม

มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัสและแบบอะซิงโครนัส

การสื่อสารแบบซิงโครนัส (Synchronous)

การรับส่งข้อมูลจะมีสัญญาณนาฬิกาซึ่งเป็นตัวกำหนดจังหวะเวลาการส่งข้อมู ร่วมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด



รูปที่ 2.7 การรับส่งข้อมูลแบบซิงโครนัส

การสื่อสารแบบอะซิงโครนัส (Asynchronous)

การรับส่งข้อมูลโดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยแต่จะใช้ให้ตัวส่งและตัวรับมีอัตราส่งข้อมูลที่เท่ากัน

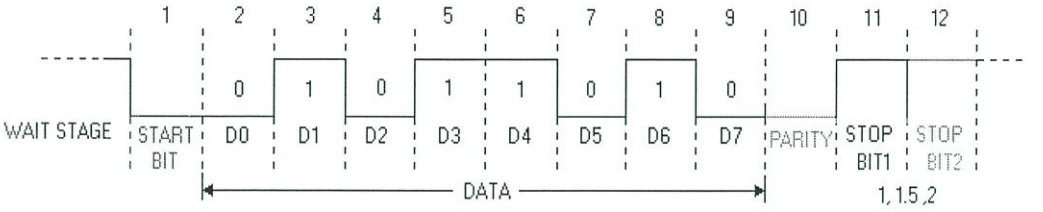
รูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

1.บิตเริ่มต้น (Start bit) มีขนาด1 บิต

2.บิตข้อมูล (Data) มีขนาด 5,6,7 หรือ 8 บิต

3.บิตตรวจสอบพาริตี้ (Parity bit) มีขนาด 1 บิตหรือไม่มี

4.บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต



รูปที่ 2.8 การรับส่งข้อมูลแบบอะซิงโครนัส

- เมื่อไม่มีการส่งข้อมูล ขา data จะมีสถานะเป็นโลจิก "1" หรือ สถานะหยุดรอ (Waiting stage)

- เมื่อเริ่มต้นส่งข้อมูลจะให้ขา data เป็นโลจิก "0" เป็นจำนวน 1 บิต เรียกว่าบิตเริ่มต้น (Start bit)

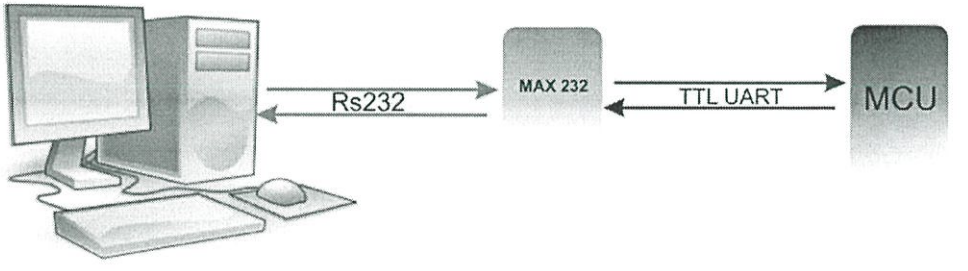
- จากนั้นก็จะเริ่มต้นส่งข้อมูล โดยส่งบิตต่ำไปก่อน (LSB)

- แล้วตามด้วยพาริตี้บิต (จะมีหรือไม่ก็ได้ ขึ้นอยู่กับการติดตั้งค่า ของทั้งสองฝ่าย)

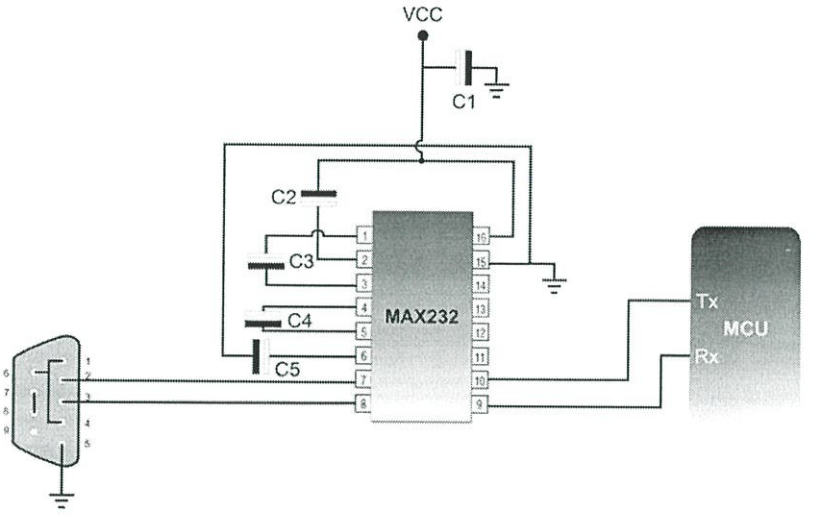
- สุดท้ายตามด้วยโลจิก "1" อย่างน้อย 1 บิต (มีขนาด1, 1.5, หรือ 2 บิต) เพื่อแสดงว่าสิ้นสุดข้อมูล

2.2 การต่อใช้งาน RS232 / MAX232

เป็น IC ที่ใช้เปลี่ยน TTL เป็น RS232 ในฝั่งส่งและเปลี่ยน RS232 เป็น TTL ในฝั่งรับ



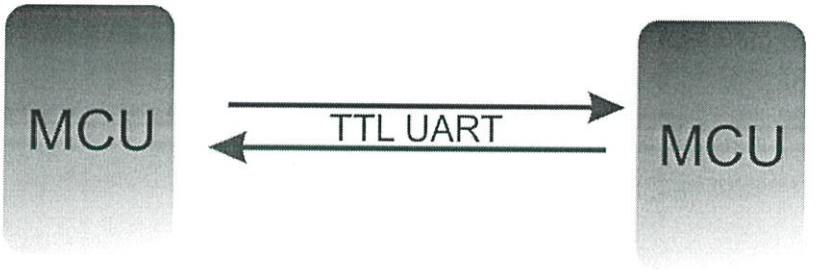
รูปที่ 2.9 การต่อใช้งาน MAX 232



Vcc = 5V

C1 – C5 = 10 uF

รูปที่ 2.10 วิธีต่อใช้งาน MAX232 โดยมี TTL 0-5V



รูปที่ 2.11 การติดต่อกัน ระหว่าง MCU 2 ตัว สามารถต่อ Rx ->Tx , Tx -> Rx กันโดยตรงได้เลย เนื่องจาก ทั้ง 2 ตัวมีระดับแรงดันเป็น TTL เหมือนกัน

รูปนี้จะเห็นว่าจะมี MAX 232 ต่อกับ MCU ทั้ง 2 ฝั่ง วิธีต่อแบบนี้มีข้อดีคือสามารถส่งข้อมูลผ่านสายได้ไกลมากขึ้นเนื่องจาก RS 232 ใช้แรงดันในสายสัญญาณสูงทำให้สามารถส่งได้ไกลกว่าใช้ TTL และเมื่อ MCU ตัวหนึ่งส่งข้อมูลมาในรูปแบบ มาตรฐาน RS232 ทำให้อุปกรณ์ หรือ MCU อีกตัวก็จะต้อง รับ และ ส่ง ข้อมูล แบบ RS232 ด้วย จากตัวอย่างที่ได้แสดงไปข้างต้น นั้น ได้ใช้ MCU เป็นการยกตัวอย่าง ซึ่งหากเราไปต่อใช้กับ อุปกรณ์ตัวอื่นๆ ก็สามารถใช้หลักการข้างต้นเพื่อตัดสินใจในการต่อวงจร

รูปที่ 2.12 การต่อMAX 232 กับ MCUทั้ง 2 ฝั่ง

2.3 ระบบ GSM

GSM ย่อมาจาก Global System for Mobile communication ซึ่งเป็นระบบโทรศัพท์เคลื่อนที่ระบบดิจิตอลระบบหนึ่ง ซึ่งใช้กันอย่างกว้างขวางทั่วโลก โดยทั่วโลกมีผู้ใช้ถึงกว่าพันล้านคน และมีใช้ในกว่า 190 ประเทศ โดยเฉพาะในยุโรป ซึ่งถือเป็นมาตรฐานของยุโรปไปโดยปริยาย สำหรับประเทศไทย เราก็ใช้ระบบ GSM กันอย่างกว้างขวางเช่นกัน โดยมีส่วนแบ่งการตลาดสูงที่สุดเช่นกัน ดังนั้นผู้ใช้ระบบ GSM จึงสามารถทำ International Roaming ได้ง่าย โดยใช้เครื่องเดิมของตัวเองGSM นั้นใช้เทคนิคการส่งข้อมูลแบบ TDMA หรือการแบ่งการเข้าถึงข้อมูลหลาย ๆ ชุดตามช่วงเวลา ซึ่งเป็นเทคนิคที่ได้รับความนิยมที่สุดในระบบการสื่อสารไร้สาย โดยกระบวนการเริ่มจากการเปลี่ยนสัญญาณเสียงให้เป็นดิจิตอล และ บีบอัดข้อมูล จากนั้นก็ส่งสัญญาณที่บีบอัดไปพร้อมข้อมูลของผู้ใช้งาน โดยผู้ใช้งานจะได้รับการจองลำดับสัญญาณไว้ในลำดับเดิม หรือสลอตเดิมตลอดเวลาใน timeslot โดยมากจะใช้ช่องสัญญาณในย่านความถี่900MHz และ 1800MHz ในการส่งข้อมูล ซึ่งจะใช้ Bandwidth รวมประมาณ 25 MHz จากนั้นซอยช่องสัญญาณในการส่องออกเป็นช่องย่อยๆโดยมีความกว้างของสสัญญาณ (Bandwidth)200kHz สำหรับ 1 ช่องสัญญาณ ดังนั้นโดยส่วนมากจำนวนช่องสัญญาณของGSM จะมีได้ประมาณ 124 ช่องสัญญาณ โดยใน 1 ช่องสัญญาณจะร้องรับผู้ใช้งานได้ 8 คน โดยแบ่งเป็น 8 timeslot ดังแสดงในรูปด้านล่าง โดยความเร็วในการสื่อสารที่ทำได้สูงสุดสำหรับ8 timeslots ต่อช่อง ก็คือ 270.833 kbit/s โดยสรุป หลักการทำงานของระบบ GSM คือ เสียงที่เราพูดจะถูกกรองโดย Filter จากนั้นจะแปลงเป็นสัญญาณดิจิตอล และทำการบีบอัดข้อมูลเพื่อลดจำนวนข้อมูลที่ต้องส่ง ก่อนจะแบ่งสัญญาณที่เราพูดนี้ออกเป็นส่วน ย่อย ๆ โดยแต่ละส่วนย่อยๆจะมีข้อมูลผู้ใช้ติดไปกับชุดข้อมูลทุกชุดสำหรับใน 1 ช่องสัญญาณจะมีทั้งหมด 8 timeslot นั่นหมายความว่ารองรับผู้ใช้ได้ 8 คนพร้อมๆกันจากรูปด้านล่างหมายเลข 1 หมายถึงผู้ใช้คนที่ 1 เมื่อผู้ใช้พูด เสียงจะถูกแบ่งออกเป็นส่วนๆแล้วส่งเรียงไปตามลำดับ โดยเรียงจากคนที่ 1 ถึง คนที่ 8 โดยเสียงส่วนที่สองของผู้ใช้คนที่ 1 ก็จะถูกส่งในรอบที่ 2, 3, 4 ไปเรื่อย ๆ เมื่อไปถึงปลายทางก็จะมีการนำกลับมาประกอบกันตามลำดับ เพื่อแปลงกลับเป็นเสียงต้นฉบันให้เหมือนกับที่ส่งไป ก่อนส่งไปยังผู้รับ โดยกระบวนการเหล่านี้เกิดขึ้นเร็วมาก จนเราไม่อาจรู้สึกได้เลยว่าเสียงของเราได้ถูกแบ่งกันส่งไปเป็นส่วน ๆ

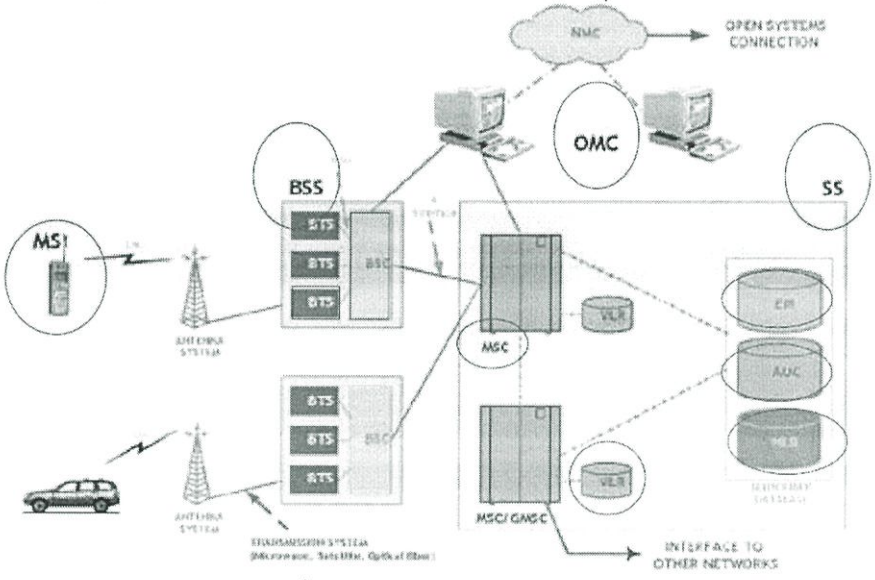
โครงสร้างระบบ GSM

1.Mobile Station แบ่งเป็นเครื่องโทรศัพท์เครื่อนที่ หรือซิมการ์ด

2. Base Station Systemแบ่งเป็นสถานีฐาน (BTS),ส่วนควบคุมสถานีฐาน

3. Switching System แบ่งเป็นชุมสายโทรศัพท์เคลื่อนที่(MSC), หน่วยเก็บข้อมูลท้องถิ่น , หน่วยเก็บข้อมูลหลักของผู้ใช้บริการ , ศูนย์ตรวจสอบการใช้งาน , หน่วยเก็บข้อมูลหมายเลขประจำเครื่อง

4. Operation and Maintenance Center ควบคุมบริหารการทำงานของโครงข่าย

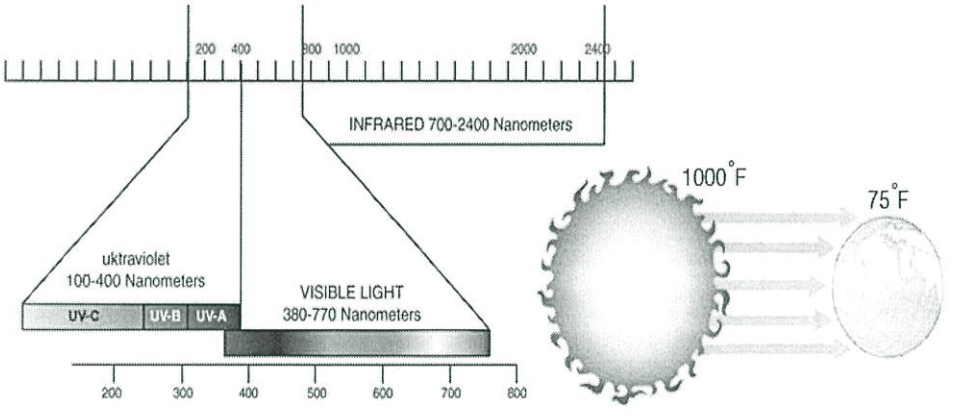


รูปที่ 2.13 โครงสร้างระบบ GSM

2.4 อินฟราเรด

วงจรเซ็นเซอร์อินฟราเรดชุดนี้เป็นวงจรตรวจจับแบบเอนกประสงค์ที่สามารถนำไปประยุกต์ใช้งานได้หลายอย่าง เช่น ชุดกันขโมยและตัวเปิดปิดไฟอัตโนมัติเมื่อเดินผ่าน ซึ่งเซนเซอร์อินฟราเรดนั้นคือการนำแสงอินฟราเรดนั้นมาประยุคต์ใช้อีกที รังสีอินฟราเรด หรือ รังสีความร้อน เป็นคลื่นแม่เหล็กไฟฟ้าชนิดหนึ่งแผ่มาจากดวงอาทิตย์ รังสีแม่เหล็กไฟฟ้าจากดวงอาทิตย์นั้นมีหลายชนิดซึ่งแต่ละชนิดนั้นมีความยาว คลื่นต่างกัน เช่น แสงที่ตามองเห็น (Visible light) มีความยาวคลื่นอยู่ระหว่าง 400 -700 นาโนเมตร, รังสีอุลตราไวโอเล็ต(Ultraviolet radiation) มีความยาวคลื่น 1 - 400 นาโนเมตร, รังสีแกมมา (Gamma ray) มีความยาวคลื่นน้อยกว่า 0.01 นาโนเมตร, รังสีเอ็กซ์ (X-ray) ความยาวคลื่น 0.01 - 1 นาโนเมตร เป็นต้น สำหรับรังสีอินฟราเรด มีความยาวคลื่น 700 นาโนเมตร -1 มิลลิเมตร เป็นคลื่นที่มีความถี่ถัดจากความถี่ของสีแดงลงมา มนุษย์จึงไม่สามารถมองเห็นรังสีอินฟราเรด แต่ก็รู้สึกถึงความร้อนได้ ส่วนสัตว์บางชนิด เช่น งูมีประสาทสัมผัสรังสีอินฟราเรดสามารถทราบตำแหน่งของเหยื่อได้โดยการสัมผัสรังสีอินฟราเรดซึ่งแผ่ออกมาจาก

ร่างกายของเหยื่อ เมื่อโลกได้รับรังสีจากดวงอาทิตย์ โลกจะแผ่รังสีสะท้อนกลับสู่บรรยากาศเรียกว่า รังสีโลก (terrestrial radiation) ซึ่งเป็นรังสีอินฟราเรดในคลื่นยาว ซึ่งแตกต่างจากรังสีอินฟราเรดจากดวงอาทิตย์ซึ่งเป็นช่วงคลื่นสั้น ตามปกติแล้วไอน้ำและคาร์บอนไดออกไซด์ในบรรยากาศจะมีปริมาณพอเหมาะและสามารถ ดูดซึมพลังงานส่วนนี้ไว้ ทำให้ โลกเก็บความร้อนไว้อยู่ในระดับที่เหมาะสมต่อการดำรงชีวิตของคน สัตว์และพืช ในปัจจุบันมีการนำรังสีอินฟราเรดมาประยุกต์ใช้งานหลายอย่างด้วยกัน เช่น ใช้เป็นตัวกลางในการสื่อสารของอุปกรณ์ไร้สายหลายชนิดทั้งโทรศัพท์มือถือและ คอมพิวเตอร์ กล้องอินฟาเรดที่สามารถจับรูปได้แม้ในเวลากลางคืน ล่าสุดมีผลงานการวิจัยจาก University of Arizonaแสดงให้เห็นว่ารังสีหรือแสงอินฟาเรดนั้นสามารถนำมาประยุกต์ใช้ประโยชน์อื่นๆ ได้อีก โดยเฉพาะอย่างยิ่งการตรวจวิเคราะห์เซลล์มะเร็ง รวมไปถึงการวิเคราะห์สารปนเปื้อนในแหล่งต่างๆ



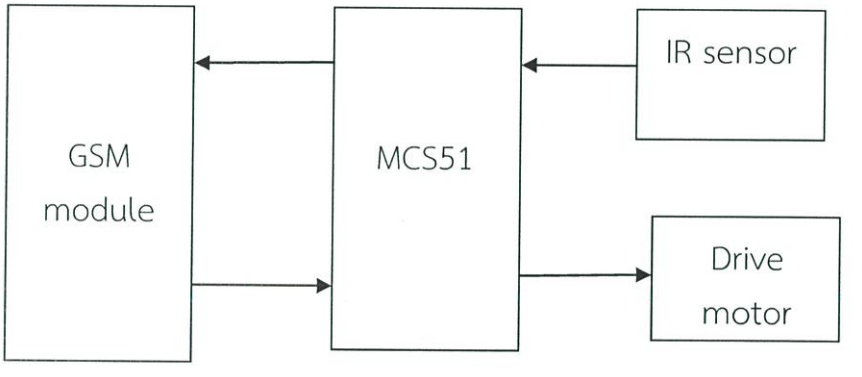
รูปที่ 2.14 แสงอินฟราเรด

บทที่ 3

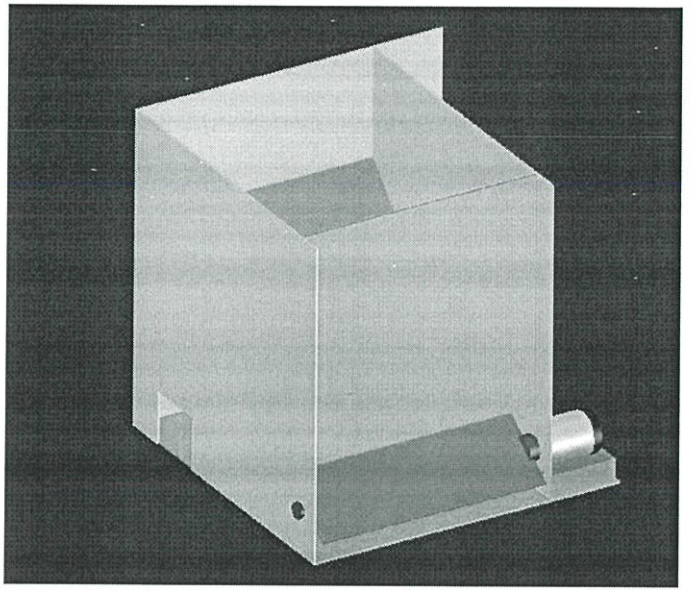
การออกแบบและการจัดทำปริญญานิพนธ์

ในการออกแบบกล่องให้อาหารสุนัขนี้ออกแบบให้ใช้ได้จริงในกรณีที่วางกล่องไว้ในที่ร่ม ปราศจากน้ำใดๆ โดยกล่องจะสามารถใช้งานได้เมื่อมารจ่ายไฟเลี้ยง 12 โวลต์ให้กับตัวกล่อง เมื่อมีการส่ง SMS จากผู้ใช้มายัง GSM Module ไมโครคอนโทรลเลอร์ก็จะทำการตรวจจับว่ามี SMS มายัง GSM Module หรือไม่ หากมีก็จะทำการสั่งให้มอเตอร์หมุนเป็นการเปิดปิดกล่องอาหาร และในกล่องจะมีเซนเซอร์อินฟราเรดสำหรับแจ้งเตือนผู้ใช้ในกรณีที่อาหารเหลือน้อยกว่าแนวระดับแสงอินฟราเรดที่ติดตั้งไว้ ไมโครคอนโทลเลอร์ก็จะทำการสั่งให้ GSM Module ส่ง SMS ไปยังผู้ใช้

3.1 การออกแบบ



รูปที่ 3.1 บล็อกไดอะแกรมระบบ



รูปที่ 3.2 โครงสร้างภายนอกกล่อง

3.1.1 การเปิดปิดกล่อง

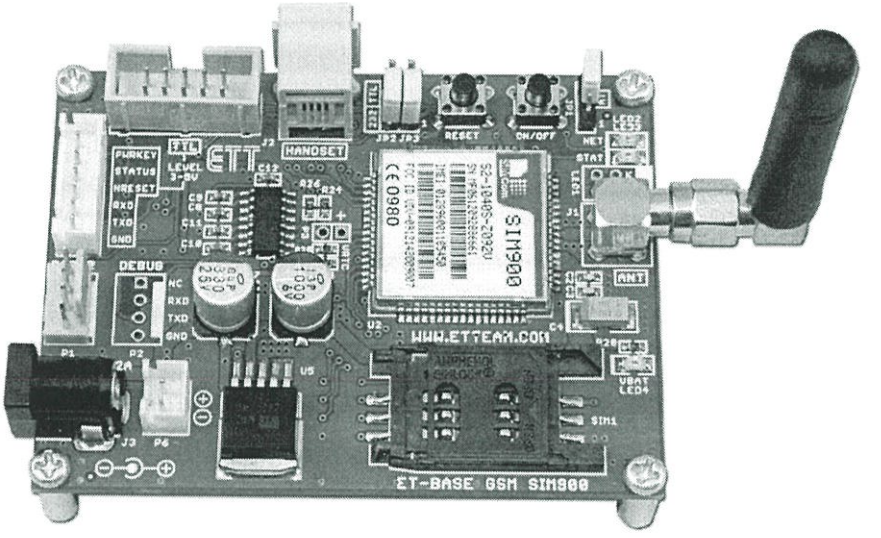
การเปิดปิดกล่องนั้นทำงานโดยใช้ GSM Module รับ SMS คำสั่งที่ส่งมาจากผู้ใช้ ซึ่งไมโครคอนโทรลเลอร์จะคอยตรวจว่ามี SMS ส่งมายัง GSM Module หรือไม่ หากมีก็จะทำการสั่งให้มอเตอร์หมุนเป็นการเปิดปิดกล่องอาหาร

3.1.2 การแจ้งเตือน

การแจ้งเตือนผู้ใช้คือการแจ้งผู้ใช้ให้ทราบว่าอาหารในกล่องใกล้หมดแล้ว โดยการใช้เซนเซอร์อินฟราเรดเป็นตัวตรวจสอบปริมาณอาหารในกล่อง เมื่อระดับอาหารต่ำกว่าแสงอินฟราเรดแล้วจากนั้นไมโครคอนโทรลเลอร์จะทำการสั่งให้ GSM Module ส่ง SMS ไปยังผู้ใช้

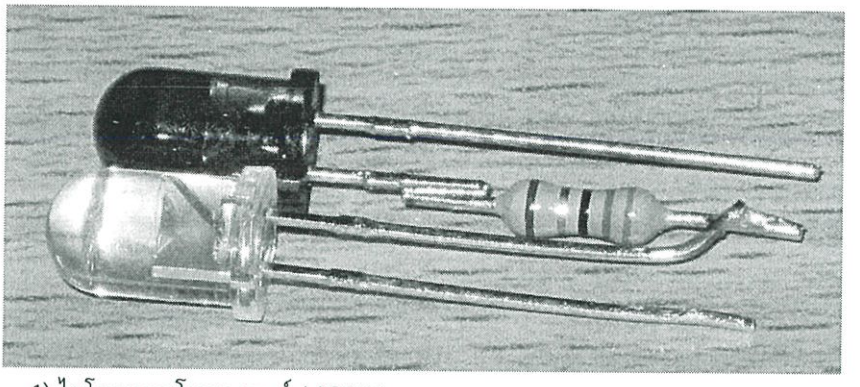
3.2 เครื่องมือที่ใช้ในการทดลอง

1) GSM Module ETT SIM900

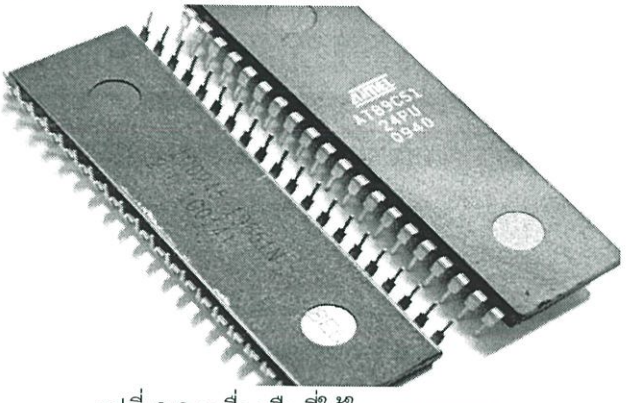


2) DC Motor

3) ตัวรับและตัวส่งอินฟราเรด



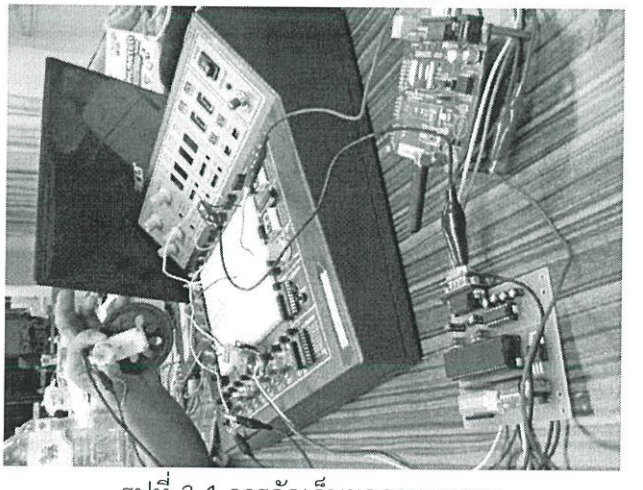
4) ไมโครคอนโทรลเลอร์ MCS51



รูปที่ 3.3 เครื่องมือที่ใช้ในการทดลอง

3.3 การจัดเก็บผลการทดลอง

การจัดเก็บผลโดยใช้ POWER SUPPLY เป็นตัวจ่ายไฟเลี้ยง 12 Volt ให้กับเครื่องให้อาหารสุนัขและสังเกตผลการทดลอง แล้วบันทึกด้วยการถ่ายภาพ



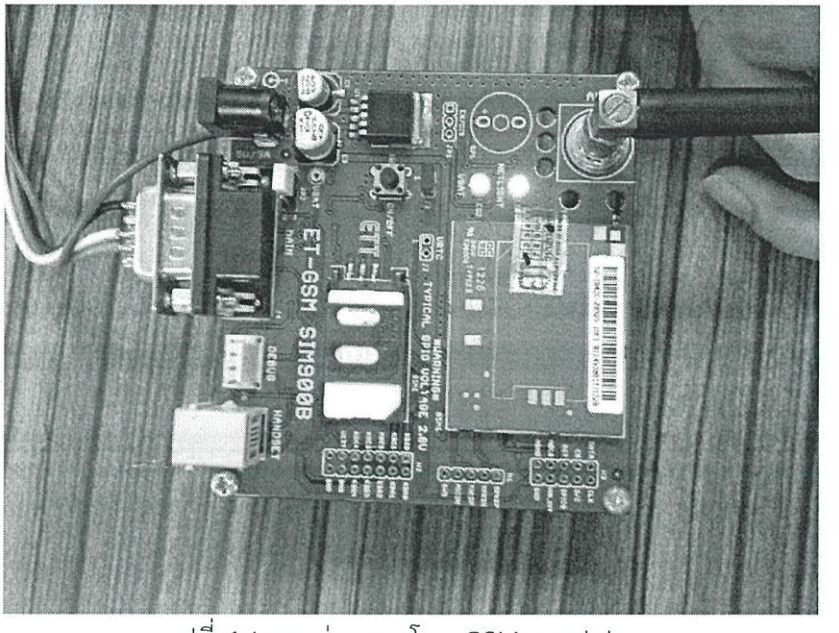
รูปที่ 3.4 การจัดเก็บผลการทดลอง

บทที่ 4

ผลการทดลอง

4.1 การส่ง SMS เพื่อสั่งงานมอเตอร์

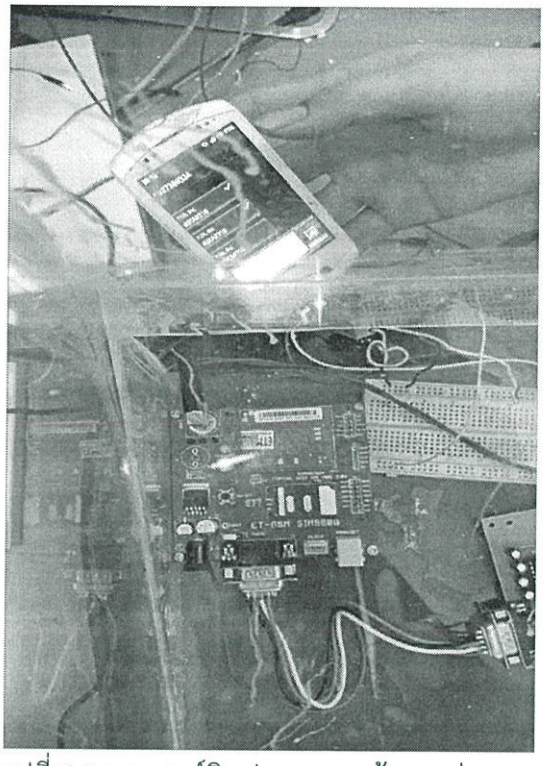
การทดลองส่ง SMS เพื่อสั่งงานมอเตอร์ทำโดยการเชื่อมต่อไมโครคอนโทรลเลอร์กับเครื่องรับส่ง SMS (GSM MODULE) จากนั้นจ่ายไฟเลี้ยง 12 Volt และส่ง SMS เข้าเครื่องรับ SMS แล้วสังเกตมอเตอร์



รูปที่ 4.1 การส่ง sms โดย GSM module

4.2 การใช้งานเซนเซอร์อินฟราเรด

นำตัวส่งอินฟราเรดมาส่องใส่ตัวรับจากนั้นสังเกตที่โทรศัพท์หมายเลขปลายทางว่ามีการส่งข้อความเข้าจากเครื่องรับส่ง SMS (GSM MODULE)



รูปที่ 4.2 เซนเซอร์อินฟราเรดกระตุ้นการส่ง sms

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

การดำเนินการทดลองในเบื้องต้นในการควบคุมการเปิดปิดกล่องอาหารผ่านโทรศัพท์นั้นเป็นการโปรแกรมการทำงานของมอเตอร์เพื่อคุมการการเปิดปิดของกล่องอาหารมีการพัฒนาตามวัตถุประสงค์ที่ตั้งไว้ดังนี้

1.สามารถสั่งเปิดปิดกล่องได้จากโทรศัพท์

2.สามารถทำการส่งข้อความผ่าน GSM Module

3.มีการส่งข้อความกลับเมื่อปริมาณอาหารน้อย

4.สามารถออกแบบกล่องเพื่อรองรับอาหารสุนัขได้

5.2 ข้อเสนอแนะ

จากการดำเนินงานพบว่ากล่องให้อาหารสุนัขนั้นมีราคาในการลงทุนค่อนข้างแพงเกินกว่าประโยชน์ที่ได้มา ดังนั้นผู้สร้างจึงออกแบบกล่องนี้ขึ้นมาเพื่อเป็นแนวทางให้ผู้ที่ต้องการใช้และสนใจได้นำไปเป็นตัวอย่างในการสร้างเพื่อขายหรือสร้างไว้ใช้เองก็ตามซึ่งควรจะให้ต้นทุนในการสร้างต่ำกว่าที่ข้าพเจ้าได้ทำไว้จึงจะสามารถนำออกมาขายในท้องตลาดได้

บรรณานุกรม

[1] ธีรวัฒน์ ประกอบผล. การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี. พิมพ์ครั้งที่ 6 : สำนักพิมพ์ ส.ส.ท.(สมาคมเทคโนโลยี ไทย-ญี่ปุ่น), 2548.

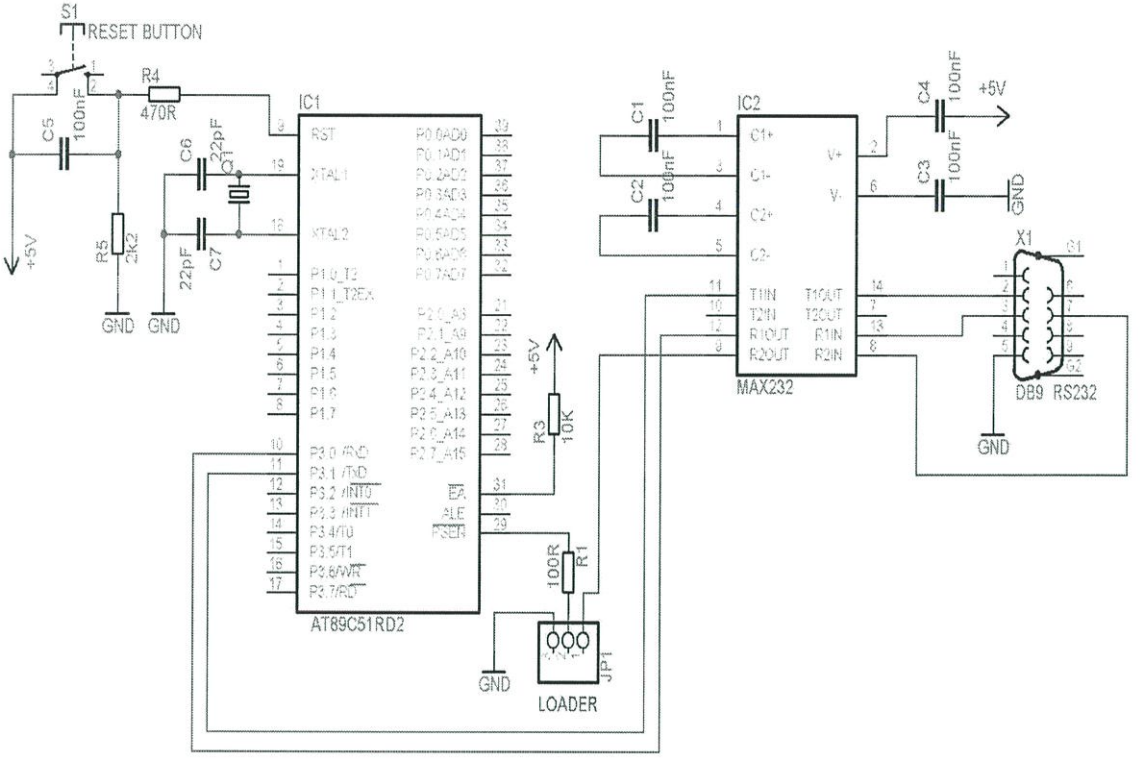
[2] ทีมงานสมาร์ทเลิร์นนิ่ง. เรียนรู้ไมโครคอนโทรลเลอร์ mcs-51 ด้วยภาษา c พร้อมโครงงาน. พิมพ์ครั้งที่ 1 : ห้างหุ้นส่วนสามัญสมาร์ทเลิร์นนิ่ง, 2552.

[3] ทิพวิลย์ คำน้ำนอง. ไมโครโทรลเลอร์และการประยุกต์ใช้งานpic. พิมพ์ครั้งที่ 4 : สำนักพิมพ์ ส.ส.ท.(สมาคมเทคโนโลยี ไทย-ญี่ปุ่น), 2553.

[4] ประภาพร ช่างไม้. คู่มือเขียนโปรแกรมภาษา c ฉบับผู้เริ่มต้น. พิมพ์ครั้งที่ 1 : นนทบุรี ไอดีซี อินโฟอิสทบิวเตอร์ เซ็นเตอร์ จำกัด, 2551.

ภาคผนวก ก

การต่อใช้งาน AT89c51


S1
RESET BUTTON
R4
470R
C5
100nF
C6
22pF
Q1
C7
22pF
R5
2K2
+5V
GND
GND
IC1
RST
XTAL1
XTAL2
P1.0_T2
P1.1_T2EX
P1.2
P1.3
P1.4
P1.5
P1.6
P1.7
P3.0 /RXD
P3.1 /TXD
P3.2 /INT0
P3.3 /INT1
P3.4/T0
P3.5/T1
P3.6/WR
P3.7/RD
P0.0/AD0
P0.1/AD1
P0.2/AD2
P0.3/AD3
P0.4/AD4
P0.5/AD5
P0.6/AD6
P0.7/AD7
P2.0_A8
P2.1_A9
P2.2_A10
P2.3_A11
P2.4_A12
P2.5_A13
P2.6_A14
P2.7_A15
EA
ALE
PSEN
AT89C51RD2
+5V
R3
10K
100R
R1
GND
JP1
LOADER
C1
100nF
C2
100nF
IC2
C1+
C1-
C2+
C2-
V+
V-
T1IN
T2IN
R1OUT
R2OUT
T1OUT
T2OUT
R1IN
R2IN
MAX232
C4
100nF
+5V
C3
100nF
GND
X1
DB9 RS232
GND

ภาคผนวก ข

Code AT89c51

```
;       SIM900A SMS TIMER
;       RFID4.PCB
;

JUMP:   DS      48
BUFF:   DS      1
LATC:   DS      1

HOLD:   DS      1
CT00:   DS      1
CT01:   DS      1
DP01:   DS      1
DP02:   DS      1

BF01:   DS      1
BF02:   DS      1
BF03:   DS      1
BF04:   DS      1
BF05:   DS      1
BF06:   DS      1
BF07:   DS      1
BF08:   DS      1
BF09:   DS      1
BF10:   DS      1
BF11:   DS      1
BF12:   DS      1
BF13:   DS      1
BF14:   DS      1
BF15:   DS      1
BF16:   DS      1

NM00:   DS      1
NM01:   DS      1
NM02:   DS      1
NM03:   DS      1
```

```
NM04:   DS      1
NM05:   DS      1
NM06:   DS      1
NM07:   DS      1
NM08:   DS      1
NM09:   DS      1

NM10:   DS      1
NM11:   DS      1
NM12:   DS      1
NM13:   DS      1
NM14:   DS      1
NM15:   DS      1
NM16:   DS      1
NM17:   DS      1
NM18:   DS      1
NM19:   DS      1

KLOC    BIT     00H
SL01    BIT     01H
SL02    BIT     02H
SL03    BIT     03H
SL04    BIT     04H
MSGL    BIT     05H

        ORG     0000H
        LJMP    INIT

        ORG     000BH
        LJMP    TIME

;--->   Initial System  ----------------------------------------------

        ORG     0050H
INIT:   MOV     P0,#0FFH
        MOV     P1,#0FFH
```

```
        MOV     P2,#0FFH
        MOV     P3,#0FFH

        MOV     CT00,#128
        MOV     CT01,#32
        MOV     HOLD,#00
        MOV     DP01,#00
        MOV     DP02,#00

        SETB    KLOC
        SETB    SL01
        SETB    SL02
        SETB    SL03
        SETB    SL04

        MOV     TMOD,#22H           ; Timer 1 8 Bit Auto Reload
        MOV     TH0,#256-225        ; Timer 0
        MOV     TL0,#256-225
        CLR     TR0                 ; Stop Timer
        SETB    IE.1                ;
        SETB    IE.7                ;

        MOV     SCON,#50H           ; Serial Mode 1
        MOV     TH1,#0FDH           ; 9600 Bps
        MOV     TL1,#0FDH
        SETB    TR1                 ; Start Timer
        CLR     RI                  ; Clear Receive Bit
        CLR     TI                  ; Clear Transmitt Bit

        LCALL   INLD
        LCALL   DISP


;--->  Timer 0 Interrupt ----------------------------------------

TIME:   DJNZ    CT00,ETIM           ; 0.01 Sec
```

```
        MOV     CT00,#128
        DJNZ    CT01,ETIM           ; 1 Sec
        MOV     CT01,#32

        DJNZ    HOLD,ETIM           ; Time Out
        CLR     TR0
        CLR     KLOC

ETIM:   RETI


;--->   Main Program  ----------------------------------------------

MAIN:   DJNZ    R4,MAN1
        LCALL   RSMS                ; Read SMS

MAN1:   JB      KLOC,MN10           ; Timer Out
        SETB    KLOC
        SETB    P3.5


        ; Finish MSG


MN10:   JNB     P0.7,MN11           ; Sensor 01
        LJMP    EMN1

MN11:       LCALL   PBR1            ; Read Phone Book
        LCALL   HLAY

        MOV     DPTR,#CMGS          ; AT+CMGS=
        LCALL   ATSD
        LCALL   PBSD                ; Phone No.
        LCALL   CRHH

        MOV     DPTR,#MSGA          ; SMS Text
```

```
        LCALL   ATSD
        LCALL   ATCZ                ; ^Z

EMN1:   NOP

MN20:   MOV     A,HOLD              ; LED
        CPL     A
        MOV     P2,A

        MOV     A,HOLD              ; LCD
        MOV     B,#10
        DIV     AB
        MOV     DP01,A
        MOV     A,B
        MOV     DP02,A
        LCALL   DISP

MN30:   NOP
        LJMP    MAIN


;--->  Read SMS  ------------------------------------------------

RSMS:   CLR     RI                  ; Clear Old Data
        LCALL   ATOK                ; Check Phone
        LCALL   HLAY
        JNB     RI,RS00
        CLR     RI
        CLR     MSGL
        LJMP    RS10

RS00:   CLR     P3.0                ; Error Lamp
        CLR     P3.1
        LCALL   HHLY
        SETB    P3.0
        SETB    P3.1
```

```
        LCALL   HLAY
        LJMP    ENRS

RS10:   MOV     DPTR,#CMGR          ; Read SMS
        LCALL   ATSD
        LCALL   ATCR

RS11:   MOV     R0,#00              ; Time Out
RS12:   MOV     R1,#00
RS13:   JB      RI,RS14
        DJNZ    R1,RS13
        DJNZ    R0,RS12
        LJMP    ENRS

RS14:   CLR     RI
        MOV     A,SBUF
        MOV     BF01,BF02
        MOV     BF02,BF03
        MOV     BF03,BF04
        MOV     BF04,BF05
        MOV     BF05,BF06
        MOV     BF06,BF07
        MOV     BF07,BF08
        MOV     BF08,BF09
        MOV     BF09,BF10
        MOV     BF10,BF11
        MOV     BF11,BF12
        MOV     BF12,BF13
        MOV     BF13,BF14
        MOV     BF14,BF15
        MOV     BF15,BF16
        MOV     BF16,A

RS20:   MOV     A,BF05              ; +66
        CJNE    A,#'+',RS21
        MOV     A,BF06
```

```
        CJNE    A,#'6',RS21
        MOV     A,BF07
        CJNE    A,#'6',RS21
        SETB    MSGL                ; Message In

        MOV     NM01,BF08           ; Save Phone In
        MOV     NM02,BF09
        MOV     NM03,BF10
        MOV     NM04,BF11
        MOV     NM05,BF12
        MOV     NM06,BF13
        MOV     NM07,BF14
        MOV     NM08,BF15
        MOV     NM09,BF16
        LJMP    RS11

RS21:   MOV     A,BF14              ; OK
        CJNE    A,#10,RS22
        MOV     A,BF15
        CJNE    A,#'O',RS22
        MOV     A,BF16
        CJNE    A,#'K',RS22

        LCALL   CONT
        LCALL   HLAY
        LCALL   HLAY
        JNB     MSGL,ENRS
        LJMP    RS30

RS22:   MOV     A,BF12              ; ERROR
        CJNE    A,#'E',RS23
        MOV     A,BF13
        CJNE    A,#'R',RS23
        MOV     A,BF14
        CJNE    A,#'R',RS23
        MOV     A,BF15
```

```
        CJNE    A,#'O',RS23
        MOV     A,BF16
        CJNE    A,#'R',RS23
        LJMP    ENRS

RS23:   NOP
        LJMP    RS11

RS30:   MOV     DPTR,#CMGD          ; Delete SMS
        LCALL   ATSD
        LCALL   CRHH

ENRS:   RET


;--->  Phone Book Read  ----------------------------------------

PBR1:   MOV     DPTR,#PB01          ; Phone Book 1
        LCALL   ATSD
        LCALL   ATCR
        LJMP    PB11

PBR2:   MOV     DPTR,#PB02          ; Phone Book 2
        LCALL   ATSD
        LCALL   ATCR
        LJMP    PB11

PB11:   MOV     R0,#00              ; Time Out
PB12:   MOV     R1,#00
PB13:   JB      RI,PB14
        DJNZ    R1,PB13
        DJNZ    R0,PB12
        LJMP    PB30

PB14:   CLR     RI
        MOV     R0,#00
```

```
        MOV     R1,#00
        MOV     A,SBUF
        MOV     BF01,BF02
        MOV     BF02,BF03
        MOV     BF03,BF04
        MOV     BF04,BF05
        MOV     BF05,BF06
        MOV     BF06,BF07
        MOV     BF07,BF08
        MOV     BF08,BF09
        MOV     BF09,BF10
        MOV     BF10,BF11
        MOV     BF11,BF12
        MOV     BF12,BF13
        MOV     BF13,BF14
        MOV     BF14,BF15
        MOV     BF15,BF16
        MOV     BF16,A

PB20:   MOV     A,BF06          ; "08
        CJNE    A,#"",PB21
        MOV     A,BF07
        CJNE    A,#'0',PB21
        MOV     A,BF08
        CJNE    A,#'8',PB21

        MOV     NM01,BF08       ; Save Phone In
        MOV     NM02,BF09
        MOV     NM03,BF10
        MOV     NM04,BF11
        MOV     NM05,BF12
        MOV     NM06,BF13
        MOV     NM07,BF14
        MOV     NM08,BF15
        MOV     NM09,BF16
        LJMP    PB11
```

```
PB21:   MOV     A,BF14              ; OK
        CJNE    A,#10,PB22
        MOV     A,BF15
        CJNE    A,#'O',PB22
        MOV     A,BF16
        CJNE    A,#'K',PB22

        LCALL   HLAY
        LJMP    PB30

PB22:   LJMP    PB11

PB30:   NOP
        RET


;--->  Send Phone Number  ----------------------------------------

PBSD:   MOV     SBUF,#""            ; Header
        LCALL   SDDL
        MOV     SBUF,#'0'           ; #0
        LCALL   SDDL

        MOV     A,NM01
        MOV     SBUF,A              ; #8
        LCALL   SDDL

        MOV     A,NM02
        MOV     SBUF,A              ; #1
        LCALL   SDDL

        MOV     A,NM03
        MOV     SBUF,A              ; #2
        LCALL   SDDL
```

```
        MOV     A,NM04
        MOV     SBUF,A          ; #3
        LCALL   SDDL

        MOV     A,NM05
        MOV     SBUF,A          ; #4
        LCALL   SDDL

        MOV     A,NM06
        MOV     SBUF,A          ; #5
        LCALL   SDDL

        MOV     A,NM07
        MOV     SBUF,A          ; #6
        LCALL   SDDL

        MOV     A,NM08
        MOV     SBUF,A          ; #7
        LCALL   SDDL

        MOV     A,NM09
        MOV     SBUF,A          ; #8
        LCALL   SDDL

        MOV     SBUF,#""        ; Header
        LCALL   SDDL

        RET

SDDL:   JNB     TI,$            ; Send & Delay
        CLR     TI
        MOV     R7,#00
        DJNZ    R7,$

        RET
```

```
;--->  AT Command Send ----------------------------------------

ATSD:   MOV     A,#00                   ; Max Text
        MOVC    A,@A+DPTR
        MOV     R0,A

        MOV     R1,#01
ATS0:   MOV     A,R1                    ; Send Table
        MOVC    A,@A+DPTR
        MOV     SBUF,A
        JNB     TI,$
        CLR     TI

        MOV     R7,#00
        DJNZ    R7,$
        INC     R1
        DJNZ    R0,ATS0
        MOV     R7,#00
        DJNZ    R7,$

        RET


;--->  Check Phone ----------------------------------------------

ATOK:   MOV     SBUF,#'A'               ; AT
        JNB     TI,$
        CLR     TI
        LCALL   DLAY
        MOV     SBUF,#'T'
        JNB     TI,$
        CLR     TI

ATCR:   LCALL   DLAY
        MOV     SBUF,#13                ; CR
```

```
        JNB     TI,$
        CLR     TI
;       MOV     SBUF,#10            ; LF
;       JNB     TI,$
;       CLR     TI

        RET

CRHL:   LCALL   DLAY
        MOV     SBUF,#13            ; CR
        JNB     TI,$
        CLR     TI
;       MOV     SBUF,#10            ; LF
;       JNB     TI,$
;       CLR     TI
        LCALL   HLAY
        LCALL   HLAY
        RET

CRHH:   LCALL   DLAY
        MOV     SBUF,#13            ; CR
        JNB     TI,$
        CLR     TI
;       MOV     SBUF,#10            ; LF
;       JNB     TI,$
;       CLR     TI
        LCALL   HHLY
        RET


ATCZ:   LCALL   DLAY

        MOV     SBUF,#26            ; ^Z
        JNB     TI,$
        CLR     TI
        LCALL   HHLY
```

```
        LCALL   HHLY
        LCALL   HHLY
        LCALL   HHLY
        LCALL   HHLY

        RET


;--->   AT Command Table  ----------------------------------------------

CMGF:   DB      09,'AT+CMGF=1'
CMGS:   DB      08,'AT+CMGS='
CMGR:   DB      09,'AT+CMGR=1'
CMGD:   DB      09,'AT+CMGD=1'
PB01:   DB      09,'AT+CPBR=1'
PB02:   DB      09,'AT+CPBR=2'

MSGA:   DB      12,'< FOOD LOW >'
MSGB:   DB      12,'OUTPUT -OFF-'


;--->   Control Command  ----------------------------------------------

CONT:   NOP

CNT7:   MOV     A,BF04                  ; Start
        CJNE    A,#'S',CNTA
        MOV     A,BF05
        CJNE    A,#'T',CNTA
        MOV     A,BF06
        CJNE    A,#'A',CNTA
        MOV     A,BF07
        CJNE    A,#'R',CNTA
        MOV     A,BF08
        CJNE    A,#'T',CNTA
```

```
        ;MOV    A,BF09              ; Time
        ;CJNE   A,#'X',CNTA
        ;MOV    A,BF10
        ;CJNE   A,#'X',CNTA

        LCALL   HHLY
        MOV     A,BF09              ; Convert
        ANL     A,#00001111B
        MOV     B,#10
        MUL     AB
        MOV     HOLD,A
        MOV     A,BF10
        ANL     A,#00001111B
        ADD     A,HOLD
        MOV     HOLD,A
        SETB    TR0
        CLR     P3.5

CNTA:   NOP

ECNT:   RET


;--->  Display Scan  ----------------------------------------------

DISP:   MOV     A,#80H              ; LCD Address
        LCALL   LCDX

        MOV     A,#'T'              ; 01
        LCALL   WRCH
        MOV     A,#'I'              ; 02
        LCALL   WRCH
        MOV     A,#'M'              ; 03
        LCALL   WRCH
        MOV     A,#'E'              ; 04
        LCALL   WRCH
```

```
MOV     A,#' '              ; 05
LCALL   WRCH
MOV     A,#'C'              ; 06
LCALL   WRCH
MOV     A,#'O'              ; 07
LCALL   WRCH
MOV     A,#'U'              ; 08
LCALL   WRCH

MOV     A,#0C0H               ; LCD Address
LCALL   LCDX

MOV     A,#'N'              ; 09
LCALL   WRCH
MOV     A,#'T'              ; 10
LCALL   WRCH

MOV     A,#' '              ; 11
LCALL   WRCH
MOV     A,#'-'              ; 12
LCALL   WRCH
MOV     A,#'<'              ; 13
LCALL   WRCH

MOV     A,DP01                ; 14
ORL     A,#00110000B
LCALL   WRCH
MOV     A,DP02                ; 15
ORL     A,#00110000B
LCALL   WRCH

MOV     A,#'>'              ; 16
LCALL   WRCH

RET
```

```
;--->  LCD Activity  ----------------------------------------------

LCDP    EQU     P1                  ; LCD Data Port
LCRS    EQU     P3.2                ; LCD RS
LCEN    EQU     P3.3                ; LCD Chip Select

INLD:   CLR     LCRS                ; Initial LCD
        MOV     LCDP,#00111000B         ; 8 Bit 1/16 Duty
        LCALL   WRT
        MOV     LCDP,#00001100B         ; Display On
        LCALL   WRT
        MOV     LCDP,#00000001B         ; Clear Display
        LCALL   WRT
        LCALL   HHLY                ; Delay
        LCALL   HHLY
        MOV     DPTR,#LGO1              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        MOV     DPTR,#LGO2              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        MOV     DPTR,#LGO3              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        MOV     DPTR,#LGO4              ; Display Logo
        LCALL   LCD0
        LCALL   HHLY
        RET

LCD0:   MOV     LCDP,#80H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        MOV     LCDP,#0C0H
        CLR     LCRS
```

```
        LCALL   WRT
        LCALL   WDAT
        RET

LCD1:   MOV     LCDP,#80H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        RET

LCD2:   MOV     LCDP,#0C0H
        CLR     LCRS
        LCALL   WRT
        LCALL   WDAT
        RET

LCDX:   MOV     LCDP,A
        CLR     LCRS
        LCALL   WRT
        RET

WDAT:   MOV     R0,#08
WDT1:   MOV     A,#00
        MOVC    A,@A+DPTR
        MOV     LCDP,A
        SETB    LCRS
        LCALL   WRT
        INC     DPTR
        DJNZ    R0,WDT1
        MOV     LCDP,#0FFH
        RET

WRCH:   MOV     LCDP,A
        SETB    LCRS
        LCALL   WRT
        MOV     LCDP,#0FFH
```

```
        RET

WRT:    MOV     R7,#00
        DJNZ    R7,$
        CLR     LCEN
        MOV     R7,#00
        DJNZ    R7,$
        SETB    LCEN
        RET


LGO1:   DB      'Initial System !'
LGO2:   DB      '.....'
LGO3:   DB      'SMS TimerControl'
LGO4:   DB      '----------'


DLAY:   MOV     R7,#00
        DJNZ    R7,$
        RET

HLAY:   MOV     R6,#00
HY01:   MOV     R7,#00
        DJNZ    R7,$
        DJNZ    R6,HY01
        RET

HHLY:   MOV     R5,#06
HH01:   MOV     R6,#00
HH02:   MOV     R7,#00
        DJNZ    R7,$
        DJNZ    R6,HH02
        DJNZ    R5,HH01
        RET
```