



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

19



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

20



VSX-6117-X DM&P Vortex86SX 300MHz Mity-Mite CPU Module



Product Name	Package
VSX-6117-X	<ul style="list-style-type: none">● Embedded Vortex86SX CPU All-in-One Board● Manual & Drivers CD x 1● RS232 cable x 1● IDE cable x 1● GPIO cable x 1● PS/2 Keyboard cable x 1● USB cable x 1 (Optional)

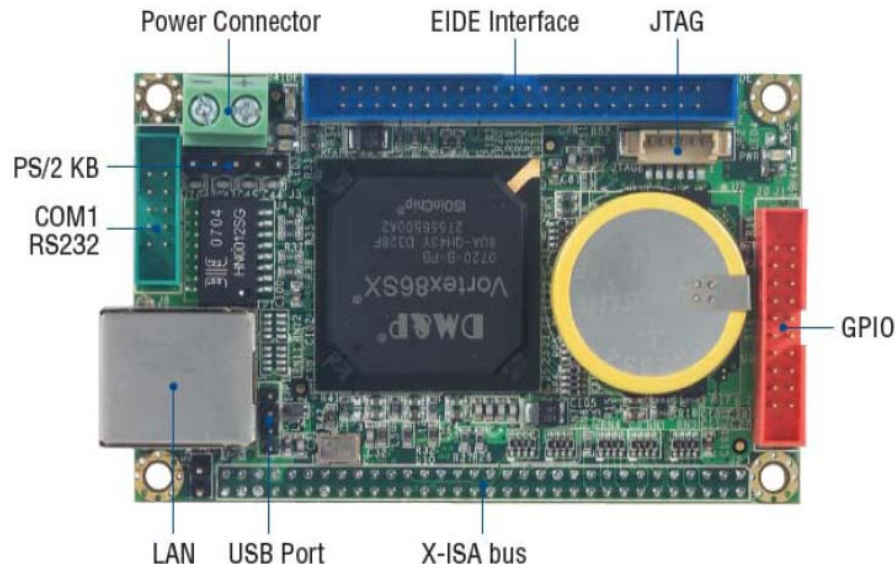


RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

21

VSX-6117-X DM&P Vortex86SX 300MHz Mity-Mite CPU Module



VSX-6117-X

DM&P Vortex86SX 300MHz

Mity-Mite CPU Module

with 1S/USB/LAN/GPIO 128MB DDR2 Onboard



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

22

VSX-6117-X DM&P Vortex86SX 300MHz Mity-Mite CPU Module

- 300 MHz Vortex86SX System-On-Chip
- 128MB DDR2 system memory
- Enhanced IDE (UltraDMA-100/66/33)
- 10/100Mbps Ethernet
- 1 USB 2.0 (host)
- Up to 1 serial ports
- 16-bit GPIO x1
- x-ISA bus
- 2 watchdog timer
- 3 PWM channels
- JTAG interface
- AMI BIOS
- 2MB SPI flash
- Single voltage +5V DC
- Support extended operating temperature range of -20°C to +70°C

OS : DOS, Linux, WinCE



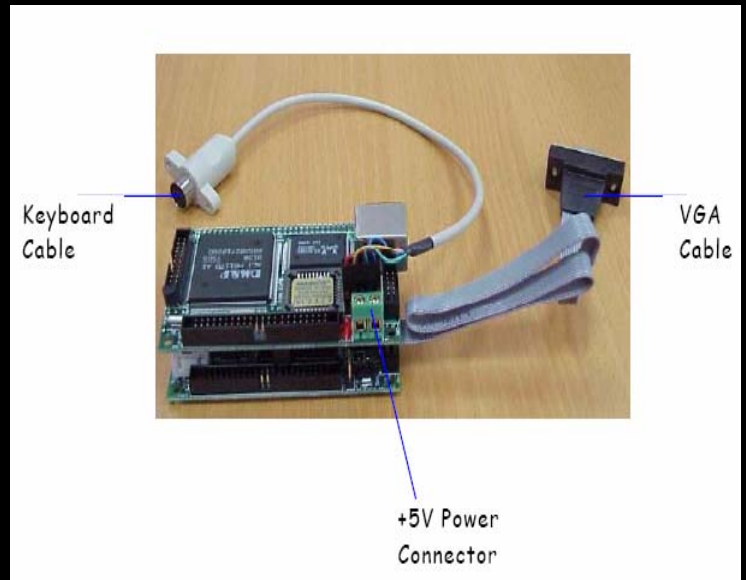
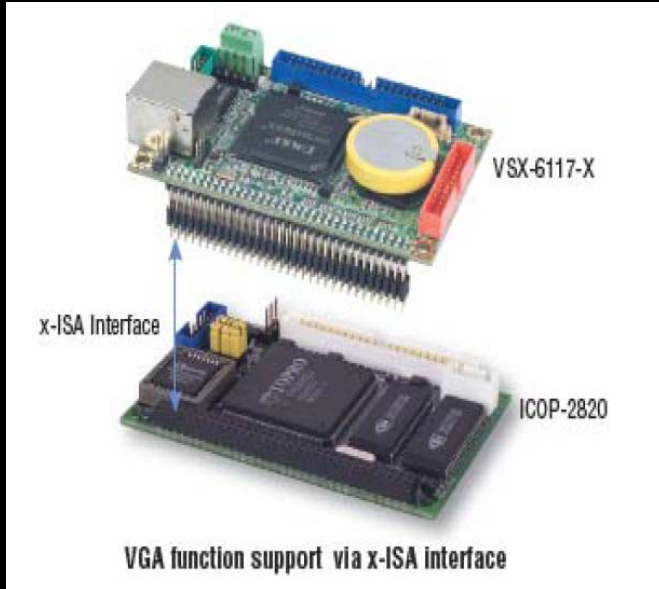
RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

23



VSX-6117-X DM&P Vortex86SX 300MHz Mity-Mite CPU Module



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

24



VSX-6117-X DM&P Vortex86SX 300MHz Mity-Mite CPU Module

There is still around 300KByte free in onboard electronics Floppy A:.. You can simply connect hard-disk on IDE interface of Mity-Mite Module and become Drive C:.. Please refer below picture.

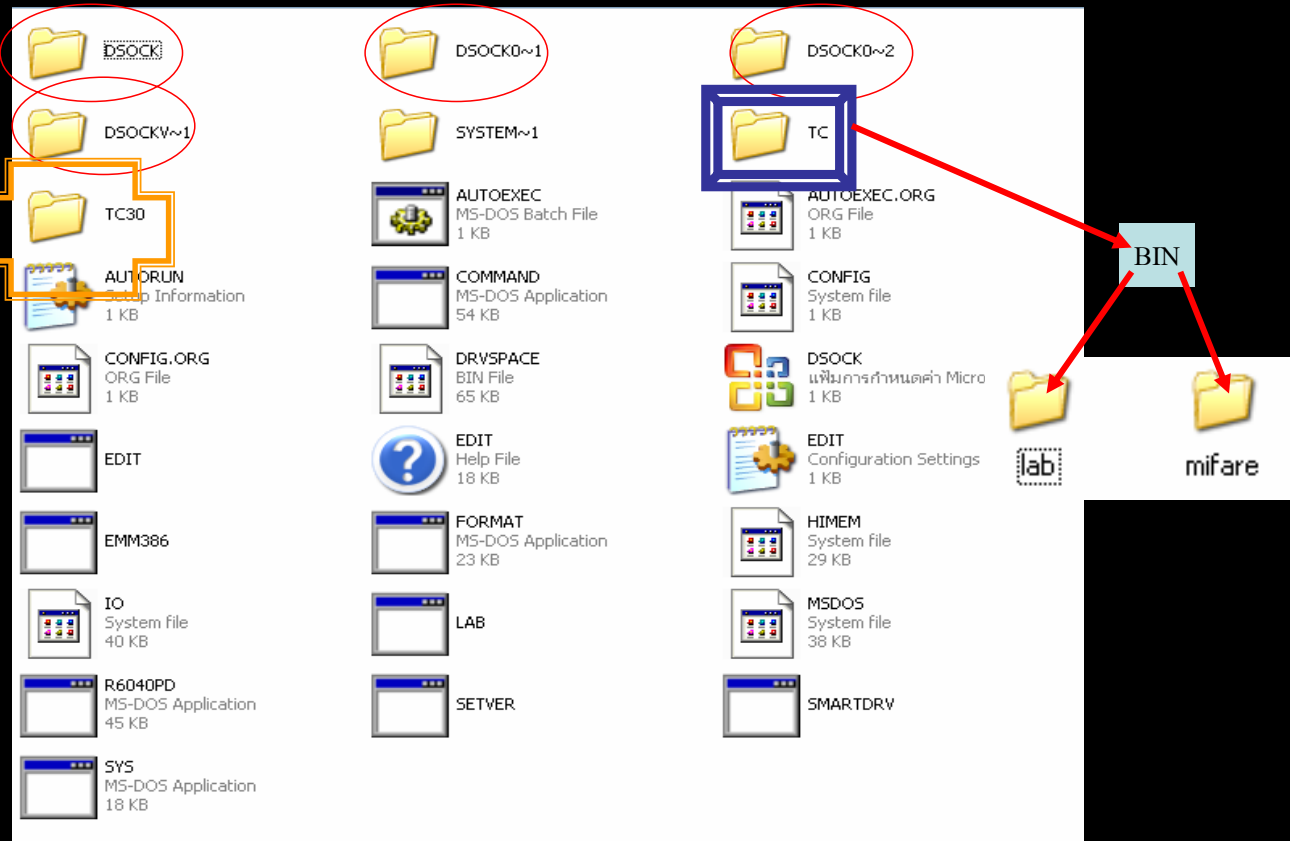




RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

25



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

26





File Edit Search

- New
- Open... F3
- Save F2
- Save as...
- Save all
- Change dir...
- Print
- DOS shell
- Quit Alt+X

Edit Search Run Compile

- Undo Alt+BkSp
- Redo Shift+Alt+BkSp
- Cut Shift+Del
- Copy Ctrl+Ins
- Paste Shift+Ins
- Clear Ctrl+Del
- Copy example
- Show clipboard

Compile Debug Project

- Compile Alt+F9
- Make F9
- Link
- Build all
- Information...
- Remove messages

ject Options Window Help

- Contents
- Index Shift+F1
- Topic search Ctrl+F1
- Previous topic Alt+F1
- Help on help
- About...

Run Compile Debug Project

- Run Ctrl+F9
- Program reset Ctrl+F2
- Go to cursor F4
- Trace into F7
- Step over F8
- Arguments...



โครงสร้างของโปรแกรมภาษา C

1	<pre>#include <stdio.h> #include <conio.h> int fibonacci(int a); void main() { int a,b; printf("Input Number"); scanf("%d", &a); b = fibonacci(a); printf("Output = %f",b); }</pre>	} Header ประกอบด้วย Preprocessor Directive และส่วนของการกำหนดข้อมูล
2	<pre>int fibonacci(int a) { int i,c; c = 1; for(i=2;i<=a;i++) c *= i; return(c); }</pre>	} Body ประกอบด้วยชุดคำสั่ง ของโปรแกรมหลักอยู่ภายใน void main(){...} และ ฟังก์ชันย่อยในตัวอย่างนี้คือ int fibonacci(int a)
3	<pre>int fibonacci(int a) { int i,c; c = 1; for(i=2;i<=a;i++) c *= i; return(c); }</pre>	



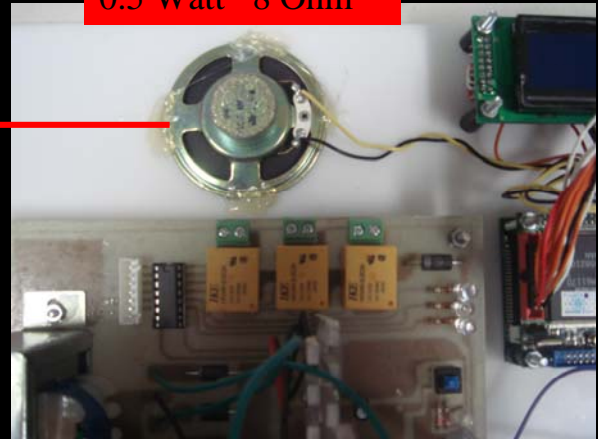
```
#include <dos.h>
```

การทดลองที่ 1

C:\TC\BIN\LAB\ SOUND1.CPP

```
sound(data);  
delay(100);  
nosound();
```

0.5 Watt 8 Ohm



Data = Integer Hz

```
1 #include <conio.h>  
2 #include <iostream.h>  
3 #include <dos.h>  
4 main()  
5 {  
6   int data,data1;  
7   data= ____; //Ex ...1000... Hz  
8   data1= ____; //Ex ...2000... Hz  
9   clrscr();  
10  cout<<"sound="<<data<<" Hz"<<endl;  
11  cout<<"sound1="<<data1<<" Hz"<<endl;  
12  do  
13  {  
14    sound(data);  
15    delay(100);  
16    nosound();  
17    delay(100);  
18    sound(data1);  
19    delay(100);  
20    nosound();  
21    delay(500);  
22  } while(!kbhit());  
23  
24  nosound();  
25  return 0;  
26 }
```

การทดลองที่ 1

C:\TC\BIN\LAB\ SOUND1.CPP



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

31



ทบทวนระบบเลขฐาน

ฐานสอง	ฐานสิบ
0000 0000	0
0000 0001	1
0000 0010	2
0000 0100	4
0000 1000	8
0001 0000	16
0010 0000	32
0100 0000	64
1000 0000	128
1111 1111	255



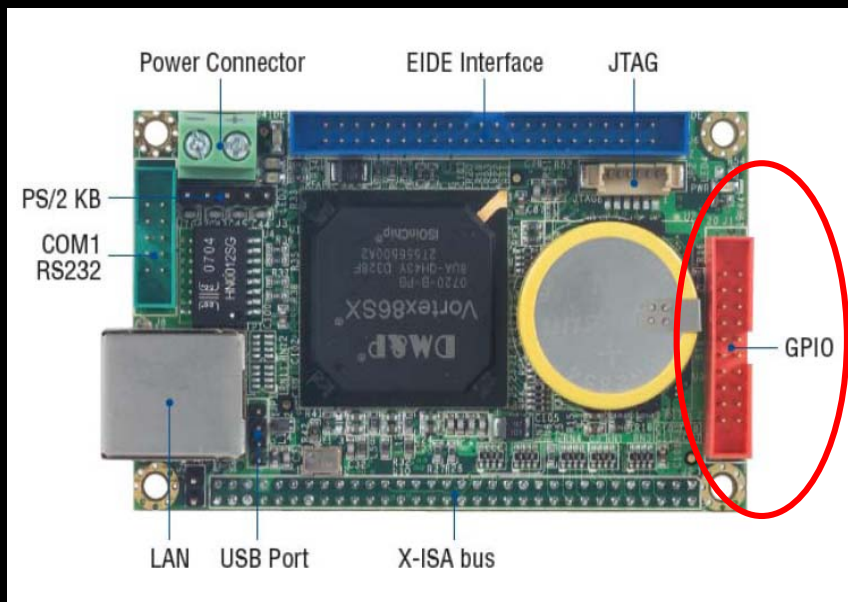
RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

32



General Purpose I/O (GPIO)



Pin	Description	Pin	Description
1	GND	2	Vcc
3	GP0	4	GP8
5	GP1	6	GP9
7	GP2	8	GP10
9	GP3	10	GP11
11	GP4	12	GP12
13	GP5	14	GP13
15	GP6	16	GP14
17	GP7	18	GP15
19	Vcc	20	GND

GPIO[7..0]

GPIO[15..8]



การ Config มี port ที่เกี่ยวข้องอยู่ 2 port คือ

0x22 = Control port หรือ Index port

0x23 = Data port

มีขั้นตอนในการใช้งาน GPIO เบื้องต้นดังนี้

1. กำหนดปลดล็อก GPIO เพื่อที่จะกำหนดสถานะการทำงาน (Unlock Configuration)

```
outputb(0x22,...0x13...);
```

```
outputb(0x23,...0xc5...);
```

2. กำหนดสถานะการทำงานให้เป็นอินพุตพอร์ทหรือเอาต์พุตพอร์ท โดย

กรณีเป็นเอาต์พุตพอร์ททั้งหมด กำหนดด้วย 0xff

กรณีเป็นอินพุตพอร์ททั้งหมด กำหนดด้วย 0x00

3. การกำหนดว่าจะส่งข้อมูลออก GPIO Port ไດ ด้วยค่าอะไร กรณีต้องการส่งข้อมูลออกทางGPIO Port GPIO[15..8] ให้ส่งค่า 0x4d ไปยัง Control Port(0x22) จากนั้นตามด้วยข้อมูลที่ต้องการส่งไปยัง Data Port(0x23)

```
outputb(0x22,...0x4d...);
```

0x4c = In put

```
outputb(0x23, ...data...); // When Data มีค่าตั้งแต่ 0-255
```

- กรณีต้องการส่งข้อมูลออกทางGPIO Port GPIO[7..0] ให้ส่งค่า 0x47 ไปยัง Control Port(0x22) จากนั้นตามด้วยข้อมูลที่ต้องการส่งไปยัง Data Port(0x23)

```
outputb(0x22,...0x47...);
```

0x46 = In put

```
outputb(0x23, ...data...); // When Data มีค่าตั้งแต่ 0-255
```



GPIO CONFIG

Unlock Config

```
outputb(0x22,...0x13...);
```

```
outputb(0x23,...0xc5...);
```

IN/OUT Set

```
outputb(0x22,0x4e);
```

```
outputb(0x22,0x4f);
```

```
outputb(0x23,0xff); | outputb(0x23,0x00);
```

```
outputb(0x23,0xff); | outputb(0x23,0x00);
```

```
outputb(0x22,0x47); | outputb(0x22,0x46);
```

```
outputb(0x22,0x4d); | outputb(0x22,0x4c);
```

```
outputb(0x23, data.); | data = inportb(0x23);
```

```
outputb(0x23, data.); | data = inportb(0x23);
```

OUT

IN

OUT

IN

GPIO7..0

GPIO 15..8


```

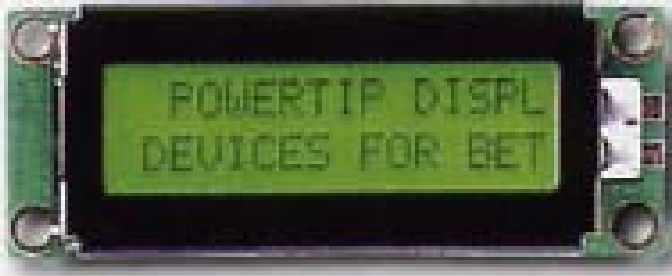
1 #include <conio.h>
2 #include <iostream.h>
3 #include <dos.h>
4
5 main()
6 { int data; // int = -32,768 to 32,767
7
8  outportb(0x22,    ); // Unlock Configuration
9  outportb(0x23,    ); // Unlock Configuration
10
11  outportb(0x22,    ); // Set GPIO[7..0]
12  outportb(0x23,    ); // Select OUT Mode
13
14  outportb(0x22,    ); // Set GPIO[15..8]
15  outportb(0x23,    ); // Select OUT Mode
16
17  clrscr();
18  do
19  {
20    cout<<"\nData < 0 exit program"<<endl;
21    cout<<"Input Data :=";
22    cin>>data;
23    outportb(0x22,    ); // {out GPIO[15..8] = 0x4d}    {out GPIO[7..0] = 0x47}
24    outportb(    ,data); // Out GPIO data
25  } while(data>=0);
26
27  outportb(0x22,    ); // {out GPIO[15..8] = 0x4d}    {out GPIO[7..0] = 0x47}
28  outportb(    ,0); // Out GPIO data
29
30  return 0;
31 }

```

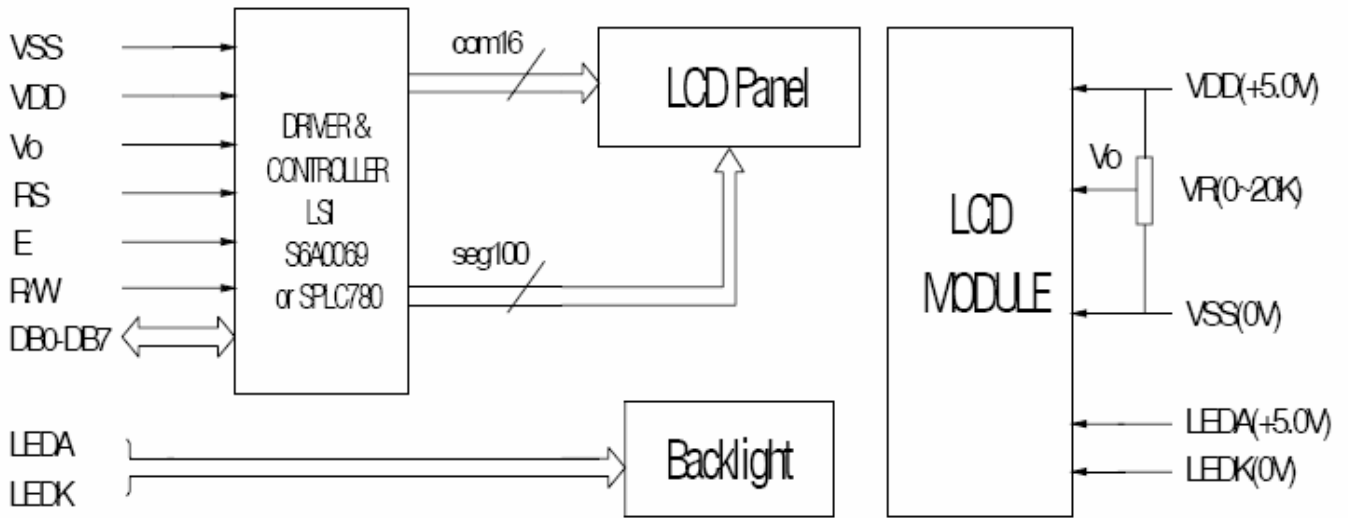
```

1 #include <conio.h>
2 #include <iostream.h>
3 #include <dos.h>
4 int main(void)
5 { int data; // int = -32,768 to 32,767
6
7  outportb(0x22,    ); // Unlock Configuration
8  outportb(0x23,    ); // Unlock Configuration
9
10  outportb(0x22,    ); // Set GPIO[15..8]
11  outportb(0x23,    ); // Set 00= Active Low FF=Active High
12
13  clrscr();
14  do
15  {
16    outportb(0x22,    ); // Set GPIO[15..8] input port
17    data= inportb(    ); // data = GPIO[15..8]
18    cout<<"Input Data :="<<data<<endl;
19    delay(100);
20  } while(!kbhit());
21  return 0;
22 }
23
24

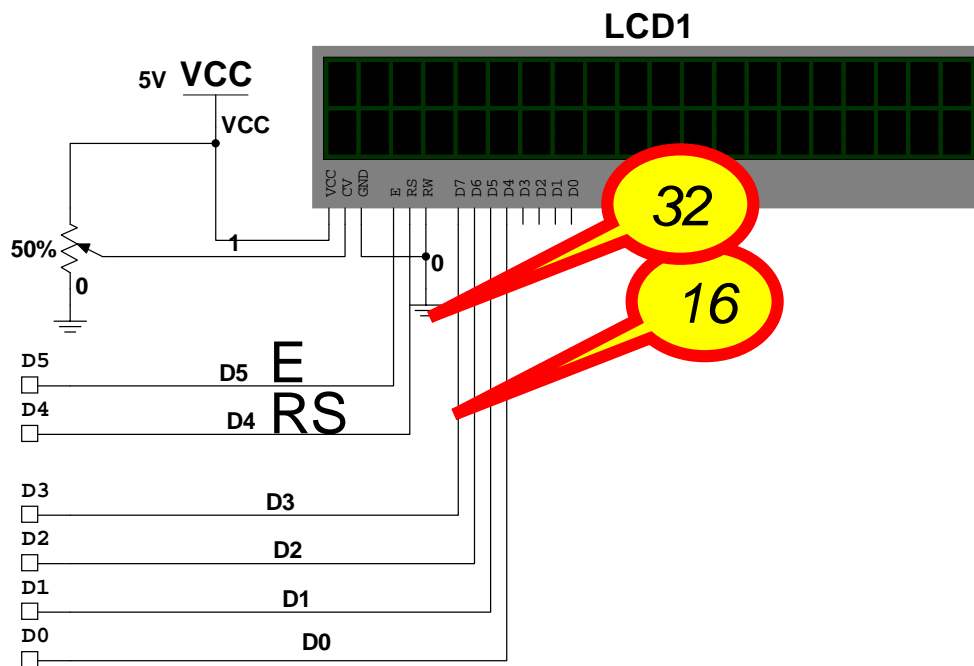
```



3. BLOCK DIAGRAM

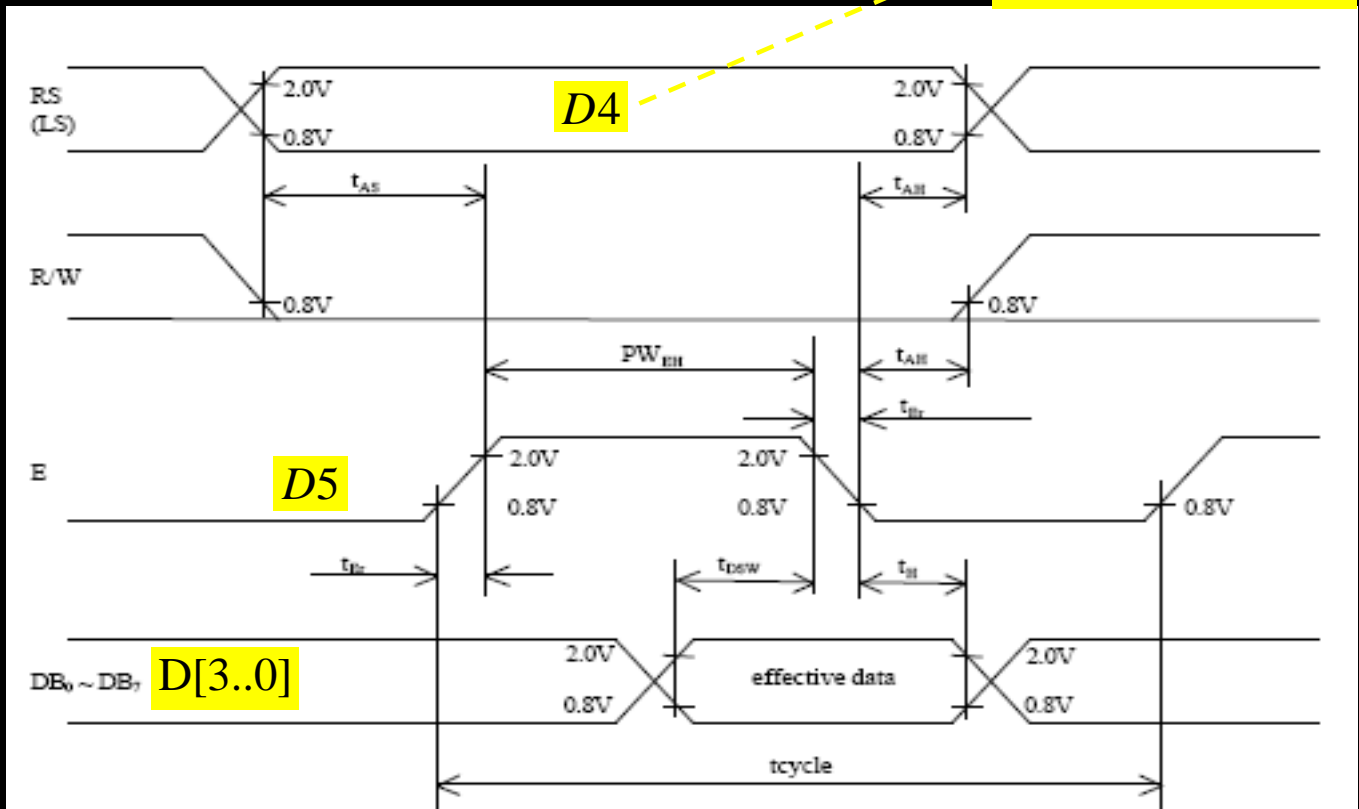


การจัดวางตำแหน่งขาสัญญาณ LCD



LCD Setting

RS=0 ชุดคำสั่ง
RS=1 แสดงผล



การกำหนดเงื่อนไขการทำงานของ LCD

LCD Setting

3.1.6 Function Set

Sets the interface data length, the number of lines, and character font.

Code	RS	R/W	DB ₇	DB ₆	DB ₅	DB ₄	DB ₃	DB ₂	DB ₁	DB ₀
	0	0	0	0	1	DL	N	F	x	x

Note: x = Don't Care

DL: Sets interface data length. Data is sent or received in 8-bit length (DB₇~DB₀) when DL = "1", and in 4-bit length (DB₇~DB₄) when DL = 0. When the 4-bit length is selected, data must be sent or received twice.

N: Sets the number of lines

N = "0" : 1 line display (1/8 duty)

N = "1" : 2 line display (1/16 duty)

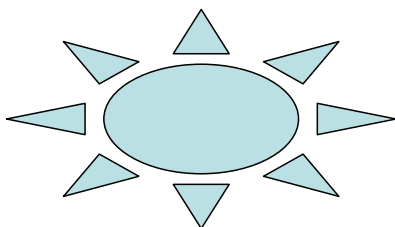
F: Sets character font.

F = "1" : 5 x 10 dots

F = "0" : 5 x 7 dots

001 1 1000

3 8



LCD Setting

3.1.3 Entry mode set

	RS	S/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	0	0	0	0	1
							I/D
							S

Sets the Increment/Decrement and Shift modes to the desired settings.

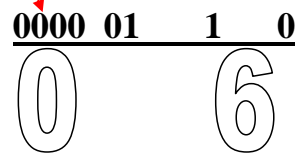
I/D: Increments (I/D = 1) or decrements (ID = 0) the DD RAM address by 1 when a character code is written into or read from the DD RAM.

The cursor or blink moves to the right when incremented by +1.

The same applies to writing and reading the CG RAM.

S: Shifts the entire display either to the right or to the left when S = 1; shift to the left when I/D = 1 and to the right when I/D = 0. Thus it looks as if the cursor stands still and only the display seems to move.

The display does not shift when reading from DD RAM nor when S = 0.



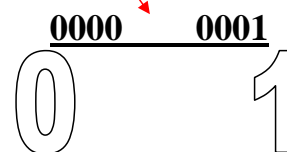
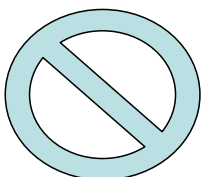
การกำหนดเงื่อนไขการทำงานของ LCD

LCD Setting

3.1.1 Clear Display

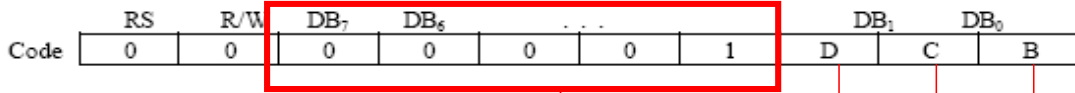
	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	0	0	0	0	1

Writes the space code "20" (hexadecimal) into all addresses of DD RAM. Returns display to its original position if it was shifted. In other words the display clears and the cursor or blink moves to the upper left edge of the display. The execution of clear display instruction sets entry mode to increment mode.



LCD Setting

3.1.4 Display ON/OFF Control

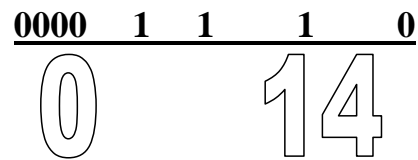
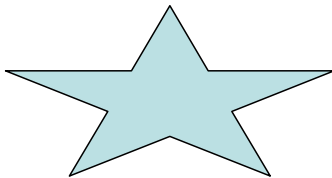


Controls the display ON/OFF status, Cursor ON/OFF and Cursor Blink function.

D: The display is ON when D = 1 and OFF when D = 0. When OFF due to D = 0, display data remains in the DD RAM. It can be displayed immediately by setting D = 1.

C: The cursor displays when C = 1 and does not display when C = 0. The cursor is displayed on the 8th line when 5 x 7 dot character font has been selected.

B: The character indicated by the cursor blinks when B = 1. The blink is displayed by switching between all blank dots and display characters at 0.4 sec intervals. The cursor and the blink can be set to display simultaneously.



3
3
3
2

Power ON When interface is 4-bit long.

Wait more than 15ms after Vcc rises to 4.5v

RS R/W DB7 DB6 DB5 DB4

0 0 0 0 1 1



BF cannot be checked before this instruction.
Function set (interface is 8 bits long)

Wait more than 4.1ms

RS R/W DB7 DB6 DB5 DB4

0 0 0 0 1 1



BF cannot be checked before this instruction.
Function set (interface is 8 bits long)

Wait more than 100 uS

RS R/W DB7 DB6 DB5 DB4

0 0 0 0 1 1



BF cannot be checked before this instruction.
Function set (interface is 8 bits long)

RS R/W DB7 DB6 DB5 DB4

0 0 0 0 1 0



BF can be checked after the following instructions.
When BF is not checked, the waiting time between instructions is longer than the execution instruction time.

0 0 0 0 1 0

0 0 N F * *

0 0 0 0 0 0

0 0 1 0 0 0

0 0 0 0 0 0

0 0 0 0 0 1

0 0 0 0 0 0

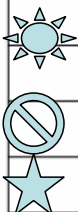
0 0 0 1 ID S

Function Set (Set interface to 4 bits long.) Interface is 8 bits length.

Function Set (interface is 4 bits long. Specify the number of display lines and character font) The number of display lines and character font cannot be changed afterwards.

Initialization ends.

Display OFF
Display ON
Entry Mode Set.



```

1 #include <stdio.h>
2 #include <conio.h>
3 #include <iostream.h>
4 #include <bios.h>
5 #include <string.h>
6 #include <time.h>
7 #include <dos.h>
8 void lcd_set();
9 void dis(int code);
10 int msb,lsb;
11 char data;
12 main()
13 {
14     outportb(0x22,0x13); // Unlock Configuration
15     outportb(0x23,0xc5); // Unlock Configuration
16     outportb(0x22,0x4e); // GPIO 7..0 Output
17     outportb(0x23,0xff); // GPIO 7..0 Output
18     outportb(0x22,0x4f); // GPIO 15..8 Output
19     outportb(0x23,0xff); // GPIO 15..8 Output
20     lcd_set(); ★
21     clrscr();
22     do {
23         cout<<"\nData = Q exit program"<<endl;
24         cout<<"Input Data :=";
25         cin>>data;
26         dis(data); ★
27     } while((data!='Q')&&(data!='q'));
28     return 0;
29 } // End Main Program

```

การทดลองที่ 4 : \TC\BIN\LAB\LCD_GPIO.CPP

```

32 void lcd_set()
33 {
34     int i;
35     int set_code[20]={ , , , , , , , , , , , };
36     for(i=0; i<12; i++)
37     {
38         outportb(0x22,0x47);
39         outportb(0x23,set_code[i]); //RS=0 E=0
40         delay(1);
41
42         outportb(0x23,set_code[i]+32); //RS=0 E=1
43         delay(1);
44
45         outportb(0x23,set_code[i]); //RS=0 E=0
46         delay(1);
47     }
48 }

```

การทดลองที่ 4 : \TC\BIN\LAB\LCD_GPIO.CPP

```

50 void dis(int code)
51 {
52     int msb,lsb;
53     msb=((code&240)>>4);
54     lsb=code&15;
55     //cout<<" msb :="<<msb<<endl;
56     //cout<<" lsb :="<<lsb<<endl;    // E,RS
57     outportb(0x22,0x47);
58     outportb(0x23,msb+ + ); // RS=1 E=0
59     delay(1);
60
61     outportb(0x23,msb+ + ); // RS=1 E=1
62     delay(1);
63
64     outportb(0x23,msb+ + ); // RS=1 E=0
65     delay(1);
66
67     outportb(0x23,lsb+ + ); // RS=1 E=0
68     delay(1);
69
70     outportb(0x23,lsb+ + ); // RS=1 E=1
71     delay(1);
72
73     outportb(0x23,lsb+ + ); // RS=1 E=0
74     delay(1);
75
76 }

```

การทดลองที่ 4 : \TC\BIN\LAB\LCD_GPIO.CPP



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม



ASCII Code (Hex Code)

Row Number	Column Number							
	000	001	010	011	100	101	110	111
0000	NUL	DLE	◊	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BELL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	_	o	DEL

LCD Setting

3.1.8 Set DD RAM Address

	RS	R/W	DB ₇	DB ₆	...	DB ₁	DB ₀
Code	0	0	1	A	A	A	A
	← Higher order bit			Lower order bits →			

Sets the address counter to the DD RAM address AAAAAA. Data is then written/read to from the DD RAM.

For a 1-line display module AAAAAA is "00" ~ "4F" (hexadecimal). For 2-line display module AAAAAA is "00" ~ "27" (hexadecimal) for the first line and "40" ~ "67" (hexa decimal) for the second line. (See section 1.7.6 "DD RAM addressing")

Out 0x22,0x47
Out 0x23,0xC0 // set DD Ram Address 1100 0000

4 0

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

จงเขียนโปรแกรมให้ LCD แสดงผลดังนี้

R	F	I	D		L	A	B		2	-	5		0	7		2	5	5	3
40	41	42																	

คำถามท้ายการทดลองที่ 4

LCD_TIME.CPP

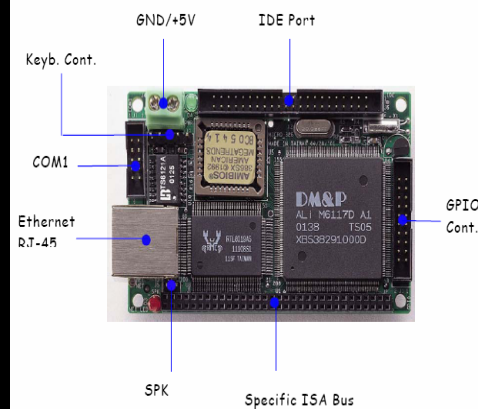
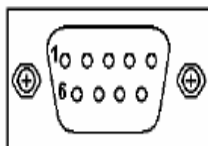
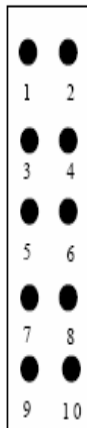
โจทย์ทำท่ายฝากให้คิด สร้างโปรแกรมแกรมย่อยให้แสดง วัน เดือน ปีในบรรทัดที่ 2



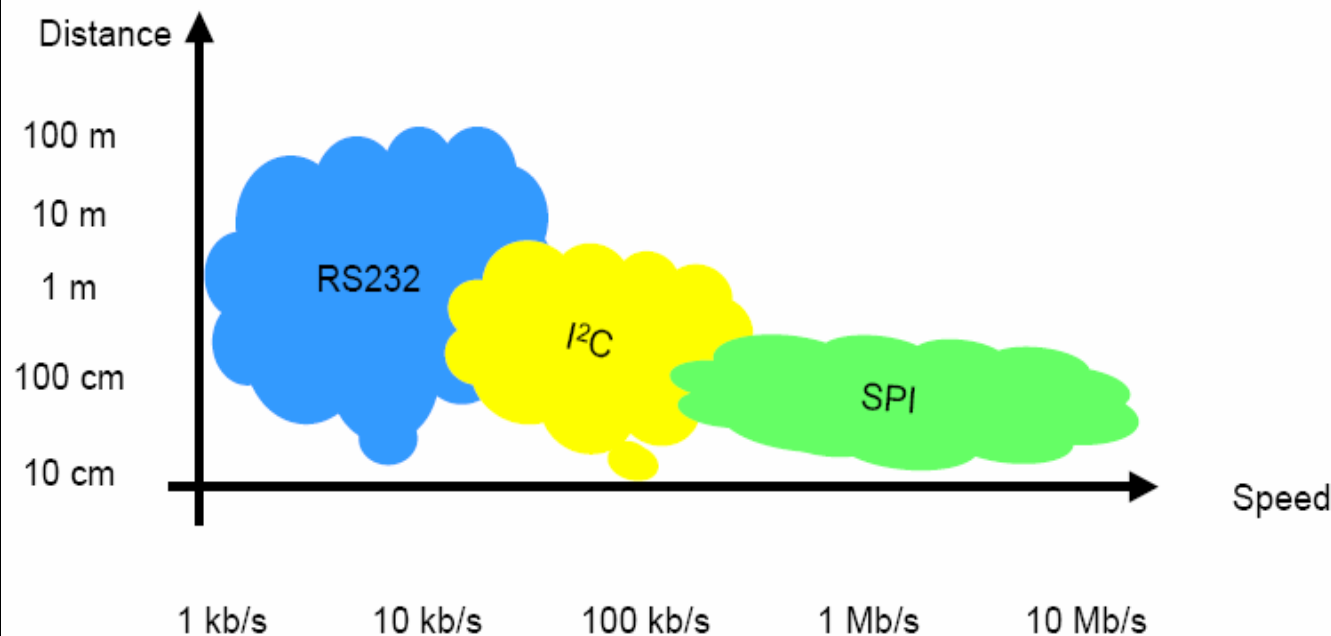
Serial Port RS232

Serial Port Interface

10- Pin	D-Type	Description
1	1	Data Carrier Detect (DCD)/5V/12V
2	2	Receive Data (RXD)
3	3	Transmit Data (TXD)
4	4	Data Terminal Ready (DTR)
5	5	Ground (GND)
6	6	Data Set Ready (DSR)
7	7	Request to Send (RTS)
8	8	Clear to Send (CTS)
9	9	Ring Indicator (RI)/5V/12V
10	X	NC



Serial Port RS232





Serial Port RS232

(Serial Communication)

การสื่อสารข้อมูลแบบอนุกรม เป็นการรับส่งข้อมูลที่ละบิต แทนที่จะทำการรับส่งข้อมูลพร้อมกันทุกบิตในเวลาเดียวกัน ข้อดีของการสื่อสารแบบนี้คือ ใช้จำนวนสายในการสื่อสารน้อย สามารถรับส่งได้ในระยะทางที่ ไกลๆ แต่ก็มีข้อเสียในด้านเวลา เพราะต้องใช้เวลาในการสื่อสารมาก เมื่อเทียบกับการสื่อสารแบบขนาน อีกทั้งโอกาสเกิด การผิดพลาดของข้อมูลก็สูงกว่าแบบขนาน



การสื่อสารแบบอนุกรมแบ่งออกได้เป็น 2 แบบ

การสื่อสารอนุกรมแบบซิงโครนัส จะมีสัญญาณนาฬิกา ร่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ คีย์บอร์ดของคอมพิวเตอร์แบบซิงโครนัสนี้จะต้องใช้สายในการเชื่อมต่ออย่างน้อยที่สุด 3 เส้น คือ สัญญาณนาฬิกา ข้อมูล และกราวด์

การสื่อสารอนุกรมแบบอะซิงโครนัส การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิกา ร่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะใช้การกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้ในการกำหนดค่าให้ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็น บิตต่อวินาที (*bit per secone : bps*)

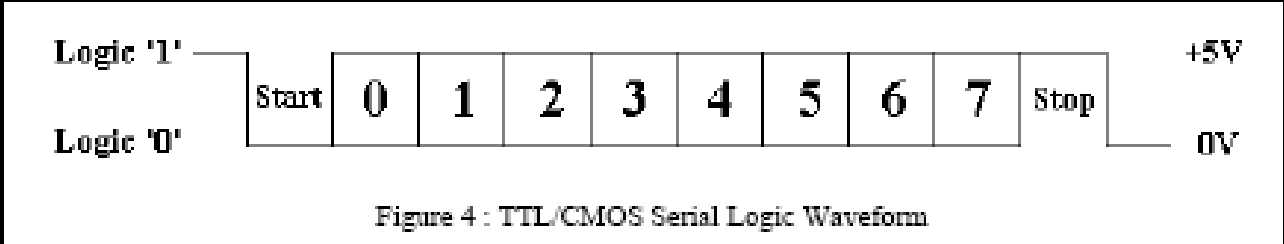
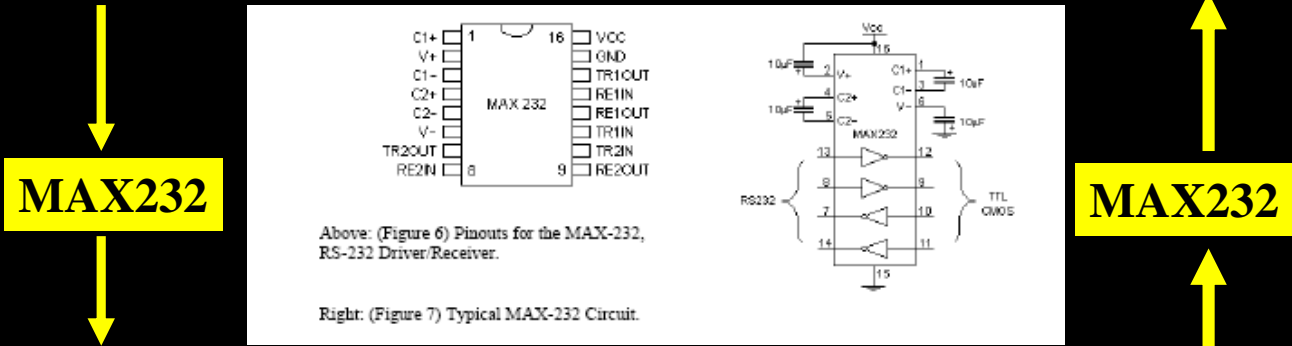


Figure 4 : TTL/CMOS Serial Logic Waveform



Above: (Figure 6) Pinouts for the MAX-232, RS-232 Driver/Receiver.

Right: (Figure 7) Typical MAX-232 Circuit.

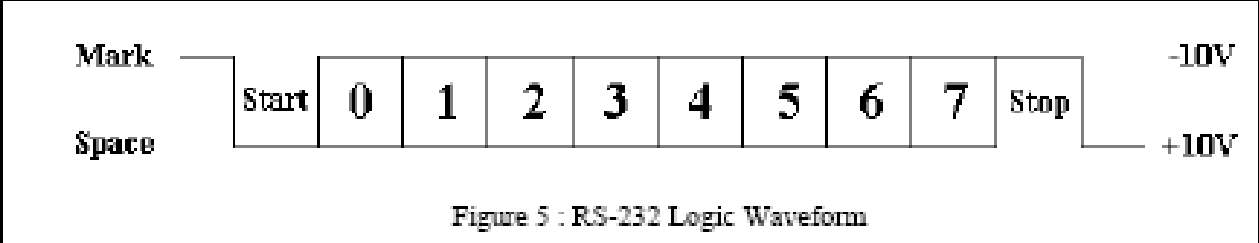


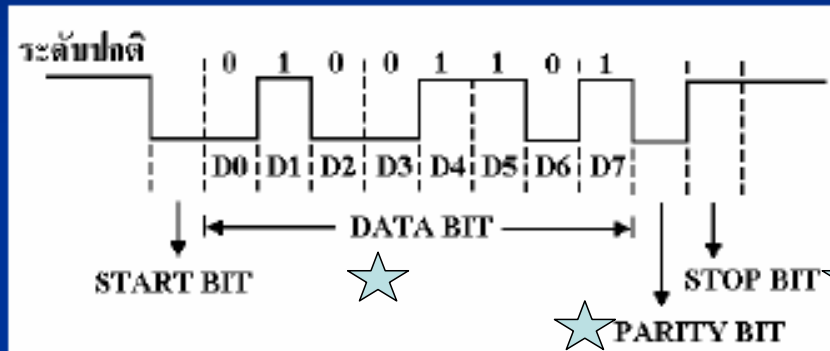
Figure 5 : RS-232 Logic Waveform



RFID Technology Research Unit

กลุ่มโปรแกรมวิศวกรรมศาสตร์และเทคโนโลยี มหาวิทยาลัยราชภัฏนครปฐม

แบบนี้ใช้มากในเครื่องไมโครคอมพิวเตอร์พีซี
รูปแบบการส่งข้อมูลจะเป็นการส่งครั้งละ 1 ไบต์ โดยมีรูปแบบดังนี้



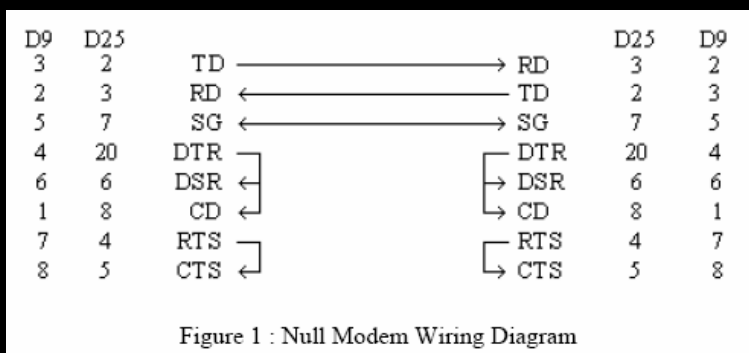
- Start Bit บอกจุดเริ่มต้นข้อมูล มีขนาด 1 บิต
- Data Bit ค่าข้อมูลมีได้ 5 ถึง 8 บิต
- Parity Bit บิตสำหรับใช้ตรวจสอบความผิดพลาดของข้อมูล มีได้ 0 ถึง 1 บิต
- Stop Bit บิตใช้บอกจุดสิ้นสุดข้อมูล มีได้ 1 1.5 และ 2 บิต



การสื่อสารแบบอนุกรม

Name	Address	IRQ
COM 1	3F8	4
COM 2	2F8	3
COM 3	3E8	4
COM 4	2E8	3

Speed (BPS)
50
300
600
2400
4800
9600
19200
38400
57600
115200



การสื่อสารแบบอนุกรม

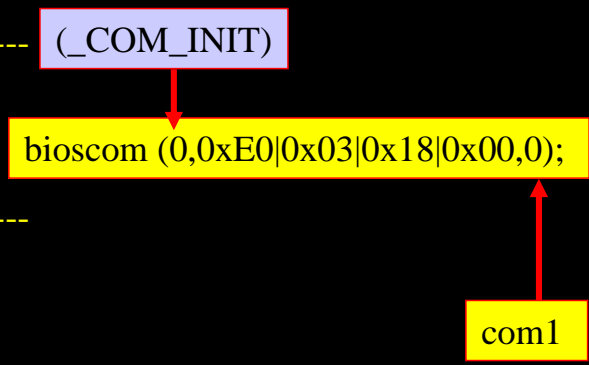
RS-232 communications (serial I/O) Declaration:

0x02 _COM_CHR7 7 data bits
 0x03 _COM_CHR8 8 data bits

0x00 _COM_STOP1 1 stop bit
 0x04 _COM_STOP2 2 stop bits

0x00 _COM_NO PARITY No parity
 0x08 _COM_ODD PARITY Odd parity
 0x18 _COM_EVEN PARITY Even parity

0x00 _COM_110 110 baud
 0x20 _COM_150 150 baud
 0x40 _COM_300 300 baud
 0x60 _COM_600 600 baud
 0x80 _COM_1200 1200 baud
 0xA0 _COM_2400 2400 baud
 0xC0 _COM_4800 4800 baud
 0xE0 _COM_9600 9600 baud



RS-232 communications (serial I/O)

Declaration: Value of cmd Lower 8 bits of return value

0 (_COM_INIT) or The lower bits are defined as shown

1 (_COM_SEND)

`bioscom(1,0x31, 0);`

`bioscom(1,49, 0);`

`bioscom(1,'1', 0);`

2 (_COM_RECEIVE) The byte read is in the lower bits of the return value--if there is no error (no upper bits are set to 1).

3 (_COM_STATUS) in the preceding diagram.

```
rfid1=bioscom(3,0,0);
if (rfid1&0x100)
{
    rfid2=bioscom(2,0,0);
    cout<<rfid2;
}
```