RAiO

RA8806

Two Layers
Character/Graphic
LCD Controller
Specification

Preliminary Version 1.2

March 2, 2009

RAiO Technology Inc.

Update History		
Version	**Date**	**Description**
1.0	March 25, 2008	Preliminary Version
1.1	June 20, 2008	1. 6-4-3 Touch panel sampling time reference table 2. Add Section 6-15 Eliminating Flicker Mode 3. Update table 8-2 DC characteristic 4. Add Section 8-3 for RA8806 driver I/F timing chart 5. Update font table of Appendix D and Appendix E
1.2	March 2, 2009	1. Add the description of Japanese Kanji font version of RA8806-J. 2. Update Figure 6-38、Figure 6-39 PWM Reference Circuits 3. Update the Section 6-10-1-3 for the usage of Chinese font and Japanese Kanji font.

Chapter Content Page

1. General Description 6
2. Feature 6
3. Block Diagram 7
4. Pin Definition 8
4-1 MPU Interface 8
4-2 Clock Interface 8
4-3 Peripheral Interface 9
4-4 LCD Driver Interface 10
4-5 Power 10
5. Register Description 11
5-1 Register List Table 11
5-2 Register Description 13
6. Function Description 27
6-1 MPU Interface 27
6-1-1 MPU Type 27
6-1-2 Command Write 30
6-1-3 Memory Write/Read 31
6-1-4 Status Read 31
6-2 Driver Interface 32
6-2-1 Display Resolution 35
6-2-2 Display Window and Active Window 35
6-2-3 Com/Seg Scan Direction 38
6-2-4 Idle Time Counter (ITCR) 38
6-3 Display Data RAM (DDRAM) 40
6-3-1 Display Layer and Display Mode Selection 40
6-3-2 Access Memory Selection 40
6-4 Touch Panel 41
6-4-1 Auto Mode 43
6-4-2 Manual Mode 45
6-4-2-1 External Interrupt Mode 45
6-4-2-2 Polling Mode 48
6-4-3 Touch Panel Sampling Time Reference Table 51
6-5 Key-Scan 52
6-6 Clock and Reset 59
6-6-1 OSC Circuit 59
6-6-2 External Clock 59
6-6-3 Reset 60
6-7 Power 61
6-7-1 Power Architecture 61
6-7-2 3V Application Circuit 61
6-7-3 5V Application Circuit 62

6-7-4 Sleep Mode 63
6-8 Interrupt and Busy 64
6-8-1 Interrupt 64
6-8-2 Busy 65
6-9 PWM 68
6-10 Display Function 70
6-10-1 Character/Graphic Mode 70
6-10-1-1 Graphic Display 70
6-10-1-2 Half Size Font 71
6-10-1-3 Full Size Font 72
6-10-1-4 Bold and Inverse 75
6-10-1-5 Two Layer Display 76
6-10-1-6 Line Gap 77
6-10-2 Gray Scale Display 77
6-10-3 Font Size Adjustment and Font Write-Time 80
6-10-4 Font Vertical Display 82
6-11 User-Defined Font 84
6-11-1 Create Font in CGRAM 84
6-11-2 Create Font in DDRAM 88
6-11-3 Create Symbol 92
6-12 Scroll Function 94
6-12-1 Horizontal Scrolling 94
6-12-2 Vertical Scrolling 96
6-13 Cursor 97
6-13-1 Cursor Position and Shift 97
6-13-2 Full Alignment 97
6-13-3 Cursor Blinking 100
6-13-4 Cursor Width and Height 100
6-14 Extension Mode for Display 101
6-15 Eliminating Flicker Mode 104
7. Package Information 108
7-1 Bonding Pad 108
7-2 Pad X/Y Coordinate 109
7-3 Pin Assignment 110
7-4 Package Dimension 111
7-5 Part Number 115
8. Electrical Characteristic 116
8-1 Absolute Maximum Ratings 116
8-2 DC Characteristic 117
8-3 Timing Characteristic Wavaform 118
Appendix A . Application Circuit 119
Appendix B . Frame Rate Table 120
Appendix C . Font Table - ASCII 123
Appendix D . Font Table - GB Code 131

Appendix E . Font Table - BIG-5 Code.....153
Appendix F . Font Table - JIS Code.....184

1. General Description

RA8806 is a LCD controller for Dot-Matrix type STN-LCD which supports both character and graphic mode display. The RA8806 has built-in two Display Data RAM(DDRAM) for two layers display, and has an embedded font ROM which is capable of displaying the full-size(16x16 pixels) traditional Chinese font(BIG5, 13973 characters) or simplified Chinese font(GB, 9216 characters). RA8806-J is one controller in this series that consisting of Kanji and Hiragana according to the JIS standard Level-1 & 2 Kanji font (6,355 characters). RA8806 also contains 4x256 embedded half-size (8x16 pixels) characters that can display ISO8859-1 ~ 4(or called Latin-1 ~ 4) alphabets using at most of English speaking and Europe countries.

RA8806 supports 8080/6800 MPU protocol interface, which is capable of switching the interface with 4-bits or 8-bits data bus. For LCD driver interface, it can be set to 4-bits or 8-bits data bus. The maximum resolution of RA8806 is 320x240 pixels in normal mode, and 640x240 or 320x480 pixels in extension mode. By using the font rotation mode, which can implement the “vertical” font display. The embedded intelligence touch panel controller provides the 4-wires resistance-type Touch Panel interface. The PWM output provides an easy contrast or back-light control method for LCD panel. RA8806 also provides a 4x8(32 keys) or 8x8(64 keys) powerful and smart Key-Scan interface includes long-key function. The flexible interrupt and polling mechanism can make it easy to control touch panel, key-scan and power mode functions. Also it can greatly reduce the MPU loading. The embedded 512Byte character generation RAM (CGRAM) allows user to build maximum 16 full-size or 32 half-size fonts. Even with the single layer display, the other unused layer can be used as CGRAM too. In this setting, the amazing 300 full-size and 600 half-size user created fonts or symbols are supported.

In addition, RA8806 supports 4-gray-scale display in FRC mode. The bit-arrangement is compatible for most gray level picture and easy to program. RA8806 also includes many useful functions, like area scroll, font inverse, font bold, font enlargement, memory clear function and so on. An innovative mechanism of “no-flicker” mode is provided in RA8806. It’s effective for removing the “flicker” in frequently display data Read/Write. User can easily improve the display quality by RA8806.

RA8806 is a powerful and flexible LCD controller. It provides the total solution for the middle-size mono LCD controller. User can save large amount of time for system development and the cost of hardware system.

2. Feature

- ◆ Support text and graphics mode.
- ◆ Maximum resolution: 320x240 with 2-Layers overlay display (AND, OR, NOR and XOR).
- ◆ Extension Mode: 640x240 or 320x480 with single layer.
- ◆ Support 4/8-bits of 6800/8080 MPU interface and 4/8-bits driver interface.
- ◆ Built-In smart 8x8 or 4x8 key-scan circuit with programmable long key function.
- ◆ Support horizontal and vertical area scrolling
- ◆ Built-In GB/BIG5 and ASCII font ROM.
- ◆ Support 90°, 180°, 270° font and display rotation.
- ◆ Support font enlargement(x1 ~ x4 in Horizontal and Vertical direction)
- ◆ Built-In 512Byte CGRAM for user-created font:
  - _ Half-size: 8x16
  - _ Full-size: 16x16
- ◆ Un-used DDRAM could be used as a CGRAM of 300 full-size or 600 half-size characters.
- ◆ Flexible interrupt/polling mechanism for touch panel, key-scan and power mode programming.
- ◆ Support font alignment function.
- ◆ Support 4-gray-scale display (FRC mode).
- ◆ Support bold font and row-row interval setting
- ◆ Built-In smart resister type touch panel controller.
- ◆ Built-In PWM for contrast or back-light control
- ◆ Power mode to reduce power consumption.
- ◆ Clock source: 4M ~ 12MHz crystal or external clock
- ◆ Built-In a 5V-to-3V DC/DC converter
- ◆ Power supply: 2.4V ~ 5.5V
- ◆ Package: Die, LQFP-100, TQFP-80 Pins

3. Block Diagram

Figure 3-1 is the internal block diagram of RA8806. The RA8806 consists of two Display Data RAM, Font ROM, Register Block, Analog to Digital Converter (ADC), Pulse Width Modulation (PWM), LCD driver interface and microprocessor interface. Figure 3-2 is the system block for application of RA8806.

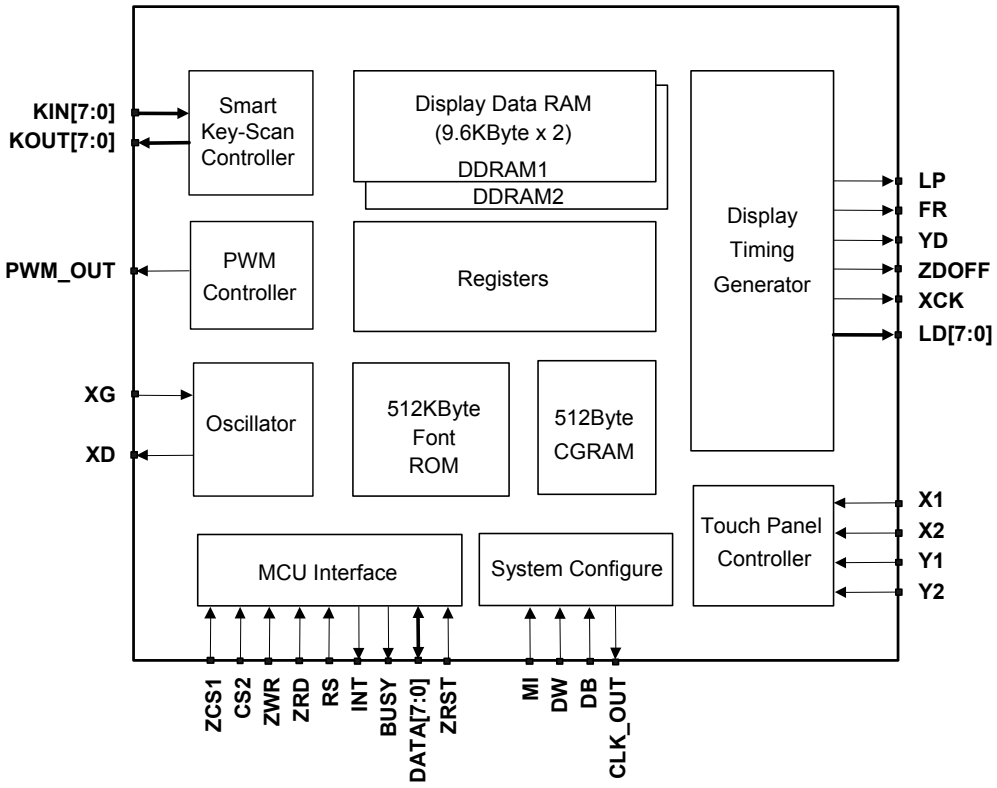


Figure 3-1 : RA8806 Block Diagram

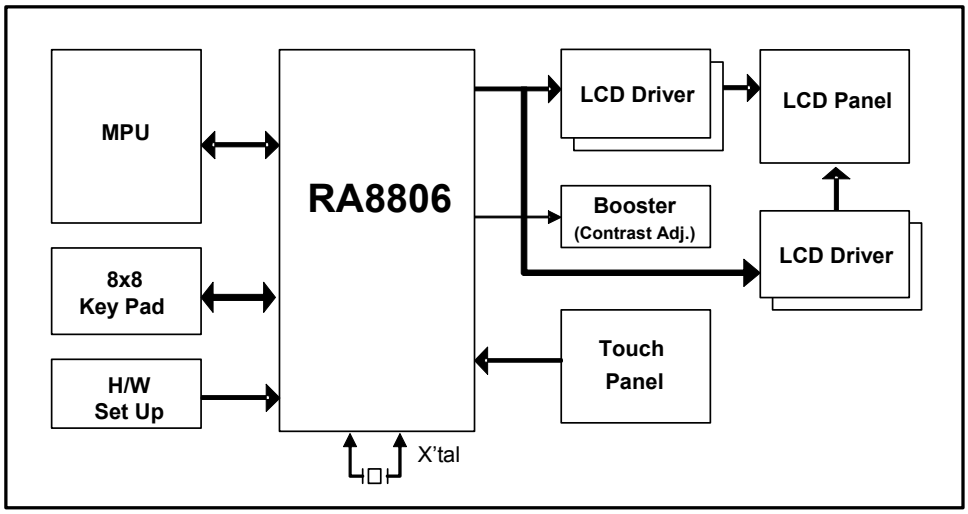


Figure 3-2 : RA8806 System Block Diagram

4. Pin Definition

4-1 MPU Interface

Pin Name	I/O	Description
DATA[7:0]	I/O	**Data Bus** These are data bus for data transfer between MPU and RA8806. The high nibble DATA[7:4] is output and should keep floating when 4-bits data bus mode is used.
ZRD (EN)	I	**Enable/Read Enable** When MPU interface(I/F) is 8080 series, this pin (ZRD) is used as data read, active low. When MPU I/F is 6800 series, this pin (EN) is used as Enable, active high.
ZWR (ZRW)	I	**Write/Read-Write** When MPU I/F is 8080 series, this pin (ZWR) is used as data write, active low. When MPU I/F is 6800 series, this pin(ZRW) is used as data read/write control. Active high for read and active low for write.
RS	I	**Command / Data Select Input** The pin is used to select command/data cycle. RS = 0, data Read/Write cycle is selected. RS = 1, status read/command write cycle is selected. In 8080 interface, usually it connects to "A0" address pin.
ZCS1 CS2	I	**Chip Select Input** The MPU interface of RA8806 is active only when ZCS1 is low and CS2 is high.
INT	O	**Interrupt Signal Output** The interrupt output for MPU to indicate the status of RA8806. It could be setup active high or low.
BUSY	O	**Busy Signal Output** This is a busy output to indicate the RA8806 is in busy state. It could be set to active high or active low by register. The RA8806 can't access MPU cycle when BUSY pin is active. It could be used for MPU to poll busy status by connecting it to I/O port.

RS	ZWR	Access Cycle
0	0	Data Write
0	1	Data Read
1	0	CMD Write
1	1	Status Read

4-2 Clock Interface

Pin Name	I/O	Description
XG	I	**X'tal Input** In internal clock mode, this pin connects to external X'tal(4M ~ 12MHz). In external clock mode, it connects to external clock.
XD	O	**X'tal Output** This pin connects to external X'tal(4M ~ 12MHz). In external clock mode, it keeps floating.

4-3 Peripheral Interface

Pin Name	I/O	Description
ZRST	I	**Reset Signal Input** This active-low input performs a hardware reset on the RA8806. It is a Schmitt-trigger input with pull-up resistor for enhanced noise immunity; however, care should be taken to ensure that it is not triggered if the supply voltage is lowered.
X1	I	**Touch Panel Input** The left analog input pin(XL) of 4-wires touch panel.
X2	I	**Touch Panel Input** The right analog input pin(XR) of 4-wires touch panel.
Y1	I	**Touch Panel Input** The top analog input pin(YU) of 4-wires touch panel. If user want to use Touch Panel function, please add a 39K~51Kohm external pull-up resistor on this pin.
Y2	I	**Touch Panel Input** The bottom analog input pin(YD) of 4-wires touch panel.
PWM_OUT	O	**PWM Output Signal** This output signal is used to control back-light module or booster circuit.
KIN[7:0]	I	**Key Pad Input** These pins are keypad inputs with pull-up resistors. For un-used input, please keep floating.
KOUT[7:0]	O	**Key Pad Output** These pins are keypad outputs. For un-used pin, please keep floating.
CLK_OUT	O	**Clock Output** This is a multi-function output pin that depending on the value of register REG[01h] Bit-6. REG[01h] Bit-6 = 0: The pin is the output of internal system clock. REG[01h] Bit-6 = 1: The pin indicate the SLEEP state of Status Register(0: Normal Mode, 1: Sleep Mode).
DW	I	**LCD Driver Data Bus Select** This pin is used to select data bus of LCD driver is 8-bits or 4-bits: 0 : LCD driver data bus is 4-bits, LD[3:0] is used. 1 : LCD driver data bus is 8-bits, LD[7:0] is used. When 4-bits data bus is used, LD[7:4] need to keep floating. RA8806T1N does not support this function, its LCD driver data bus is fix 4-bits.
MI	I	**MPU Type Select** This pin is used to select MPU interface protocol: 0 : Intel 8080 series MPU interface. 1 : Motorola 6800 series MPU interface.
DB	I	**8080/6800 MPU Data Bus Select** This pin is used to select data bus width. 0 : 4-bits MPU I/F, DATA[3:0] is used. 1 : 8-bits MPU I/F, DATA[7:0] is used. When 4-bits data bus is used, DATA[7:4] need to keep floating.

4-4 LCD Driver Interface

Pin Name	I/O	Description
YD	O	**Start Signal of LCD Per Frame** YD is the reset pulse for the COM driver, it's active during the last COM period of each frame and latched by LP signal.
FR	O	**LCD AC Wave Output** This signal controls the Level Shift of LCD driver. Normally it works as VDD/GND interlacing to prevent the liquid cystal polarization.
LP	O	**LCD Common Latch** LP is the latch pulse for the shift register of SEG driver to SEG output. It is also used as COM driver shift clock.
XCK	O	**LCD Clock** XCK is the latch pulse of the LCD driver data(LD[7:0]) for SEG driver. The falling edge of XCK will latch the LD[7:0] (or LD[3:0] for 4-bits driver) to the shift register.
ZDOFF	O	**LCD Display Off** This signal is used to control the LCD Display On or Off. 0 : Display off. 1 : Display on.
LD[7:0]	O	**LCD Driver Data Bus** When 8-bits LCD driver IC is used. LD[7:0] are connected to LCD driver data bus. When 4-bits driver is used, LD[3:0] are connected to LCD driver data bus and LD[7:4] keep floating. RA8806T1N supports LD[3:0] only.

4-5 Power

Pin Name	I/O	Description
VDDH	P	**5V Power** This is the source power for DC to DC converter. In 5V power application, it is connected to 5V. If 3V application is used, then keep this pin floating.
VDD	P	**3V Power** If the pin VDDH connects to 5V power then the pin will driving 3V power, and must add an external 1uF capacitor to GND. If 3V application is used, then connecting this pin to external 3V power directly.
VDDP	P	**Power for I/O Buffer** VDDP can be 3V or 5V.
AVDD	P	**Analog Power for ADC Touch Panel Controller** AVDD can be 3V or 5V.
GND **GNDP**	P	**Ground**
AGND	P	**Analog Ground for ADC Touch Panel Controller** Connect this pin to 0V earth ground(GND).
TESTMD	I	**Test Mode** This pin is used for test only. It has internal pull-low and need to keep floating.
TESTI	I	**Test Pin** The pin is used for test function, It has internal pull-low and need to keep floating.

5. Register Description

5-1 Register List Table

Table 5-1 : Cycle List Table

CYC_NAME	RS	ZWR	Description
CMD	1	0	Command write cycle, for writting register number(REG#)
STATUS	1	1	Status read cycle, using to check Interrupt or Sleep status.
DATW	0	0	Data write cycle, using to write register data or memory data.
DATR	0	1	Data read cycle, using to read register data or memory data.

Table 5-2 : Register Table

REG#	Name	D7	D6	D5	D4	D3	D2	D1	D0	Default
--	**STATUS**	MBUSY	SBUSY	SLEEP			WAKE_STS	KS_STS	TP_STS	--
00h	**WLCR**	PWR	LINEAR	SRST	--	TEXT_MD	ZDOFF	GBLK	GINV	00h
01h	**MISC**	NO_ FLICKER	CLKO_SEL	BUSY_ LEV	INT_LEV	XCK_SEL1	XCK_SEL0	SDIR	CDIR	04h
03h	**ADSR**	SCR_PEND	--	--	--	BIT_INV	SCR_DIR	SCR_HV	SCR_EN	00h
0Fh	**INTR**	--	WAKI_EN	KEYI_EN	TPI_EN	TP_ACT	WAK_STS	KEY_STS	TP_STS	00h
10h	**WCCR**	CUR_INC	FULL_OFS	BIT_REV	BOLD	T90DEG	CUR_EN	CUR_BLK	---	00h
11h	**CHWI**	CURH3	CURH2	CURH1	CURH0	ROWH3	ROWH 2	ROWH 1	ROWH 0	00h
12h	**MAMR**	CUR_HV	DISPMD2	DISPMD1	DISPMD0	L_MIX1	L_MIX 0	MW_MD1	MW_MD0	11h
20h	**AWRR**	--	--	AWR5	AWR4	AWR3	AWR2	AWR1	AWR0	27h
21h	**DWWR**	--	--	DWW5	DWW 4	DWW 3	DWW 2	DWW 1	DWW 0	27h
30h	**AWBR**	AWB7	AWB6	AWB5	AWB4	AWB3	AWB2	AWB1	AWB0	EFh
31h	**DWHR**	DWH7	DWH6	DWH5	DWH4	DWH3	DWH2	DWH1	DWH0	EFh
40h	**AWLR**	--	--	AWL5	AWL4	AWL3	AWL2	AWL1	AWL0	00h
50h	**AWTR**	AWT7	AWT6	AWT5	AWT4	AWT3	AWT2	AWT1	AWT0	00h
60h	**CURX**	--	--	CURX5	CURX4	CURX3	CURX2	CURX1	CURX0	00h
61h	**BGSG**	--	--	BGSG5	BGSG4	BGSG3	BGSG2	BGSG1	BGSG0	00h
62h	**EDSG**	EDSG7	EDSG6	EDSG5	EDSG4	EDSG3	EDSG2	EDSG1	EDSG0	00h
70h	**CURY**	CURY7	CURY6	CURY5	CURY4	CURY3	CURY2	CURY1	CURY0	00h
71h	**BGCM**	BGCM7	BGCM6	BGCM5	BGCM4	BGCM3	BGCM2	BGCM1	BGCM0	00h
72h	**EDCM**	EDCM7	EDCM6	EDCM5	EDCM4	EDCM3	EDCM2	EDCM1	EDCM0	00h
80h	**BTMR**	BLKT7	BLKT6	BLKT5	BLKT4	BLKT3	BLKT2	BLKT1	BLKT0	00h
90h	**ITCR**	ITC7	ITC6	ITC5	ITC4	ITC3	ITC2	ITC1	ITC0	00h
A0h	**KSCR1**	KEY_EN	KEY4X8	KSAMP1	KSAMP0	LKEY_EN	KF2	KF1	KF0	00h
A1h	**KSCR2**	KWAK_EN	--	--	--	LKEY_T1	LKEY_T0	KEYNO1	KEYNO0	00h
A2h	**KSDR0**	KSD07	KSD06	KSD05	KSD04	KSD03	KSD02	KSD01	KSD00	00h
A3h	**KSDR1**	KSD17	KSD16	KSD15	KSD14	KSD13	KSD12	KSD11	KSD10	00h
A4h	**KSDR2**	KSD27	KSD26	KSD25	KSD24	KSD23	KSD22	KSD21	KSD20	00h
B0h	**MWCR**	MWD7	MWD6	MWD5	MWD4	MWD3	MWD2	MWD1	MWD0	--
B1h	**MRCR**	MRD7	MRD6	MRD5	MRD4	MRD3	MRD2	MRD1	MRD0	--

(Continued)

REG#	Name	D7	D6	D5	D4	D3	D2	D1	D0	Default
C0h	**TPCR1**	TP_EN	TP_SMP2	TP_SMP1	TP_SMP0	TPWAK_EN	ACLK2	ACLK1	ACLK0	00h
C1h	**TPXR**	TPX9	TPX8	TPX7	TPX6	TPX5	TPX4	TPX3	TPX2	00h
C2h	**TPYR**	TPY9	TPY8	TPY7	TPY6	TPY5	TPY4	TPY3	TPY2	00h
C3h	**TPZR**	TPX1	TPX0	--	--	TPY1	TPY0	--	--	00h
C4h	**TPCR2**	MTP_MD	--	--	--	--	--	MTP_PH1	MTP_PH2	00h
D0h	**PCR**	PWM_EN	PWM_DIS_LEV	--	--	PCLK_R3	PCLK_R2	PCLK_R1	PCLK_R0	00h
D1h	**PDCR**	PDUTY7	PDUTY6	PDUTY5	PDUTY4	PDUTY3	PDUTY2	PDUTY1	PDUTY0	00h
E0h	**PNTR**	PND7	PND6	PND5	PND4	PND3	PND2	PND1	PND0	00h
F0h	**FNCR**	ISO8859_EN	--	--	--	MCLR	ASC	ASC_SEL1	ASC_SEL0	00h
F1h	**FVHT**	FH1	FH0	FV1	FV0	--	--	--	--	00h

5-2 Register Description

STATUS Register (RS = 1, ZWR = 1)

Bit	Description	Access
7	**Memory Write Busy Flag** 0 : Not busy. 1 : Busy, when font write or memory clear cycle is running, the busy flag = 1.	R
6	**SCAN_BUSY** 0 : Not busy. 1 : When driver scan logic is not idle(i.e. XCK is active), SCAN_BUSY = 1.	R
5	**SLEEP** 0 : Normal mode. 1 : Sleep mode.	R
4-3	**NA**	R
2	**Wakeup Status bit** (The same with REG[0Fh] Bit-2.)	R
1	**KS Status bit** (The same with REG[0Fh] Bit-1.)	R
0	**TP Status bit** (The same with REG[0Fh] Bit-0.)	R

REG [00h] Whole Chip LCD Controller Register (**WLCR**)

Bit	Description	Default	Access
7	**Power Mode** 0 : Normal Mode. All of the functions of RA8806 are available in this mode. 1 : Sleep Mode. When RA8806 is in Sleep mode, all of functions enter off mode, except the wake-up trigger block. If wake-up event occurred, RA8806 would wake-up and return to Normal mode.	0	R/W
6	**Linear Decode mode** This bit is used to define the Font ROM address mapping rule. The standard product is set to 0. And 1 for special application that when user a want to create a new Mask Code. 0 : BIG5/GB ROM mapping rule. 1 : User-defined ROM mapping rule.	0	R/W
5	**Software Reset** 0 : Normal Operation. 1 : Reset all registers except the contents of Display Data RAM (Only work at Normal mode). When this bit set to “1”, the next MPU cycle for RA8806 have to wait 3 clocks at least.	0	R/W
4	**Reserved**	0	R
3	**Text Mode Selection** 0 : Graphical Mode. The written data will be treated as a bit-map pattern. 1 : Text Mode. The written data will be treated as an ASCII, BIG5 or GB code.	0	R/W

2	**Set Display On/Off Selection** The bit is used to control LCD Driver Interface signal – "DISP_OFF". 0 : DISP_OFF pin output low(Display Off). 1 : DISP_OFF pin output high(Display On).	0	R/W
1	**Blink Mode Selection** 0 : Normal Display. 1 : Blink Full Screen. The blink time is set by register BTMR.	0	R/W
0	**Inverse Mode Selection** 0 : Normal Display. 1 : Inverse Full Screen. It will cause the display inversed.	0	R/W

REG [01h] Misc. Register (**MISC**)

Bit	Description	Default	Access
7	**Eliminating Flicker** 1 : Eliminating flicker mode, scan will auto-pending when busy. 0 : Normal mode.	0	R/W
6	**Clock Output** (Pin CLK_OUT) Control 1 : The pin "CLK_OUT" indicates the SLEEP state of Status Register(0: Normal Mode, 1: Sleep Mode). 0 : The pin "CLK_OUT" is the output of Internal system clock.	0	R/W
5	**Busy Polarity (for "BUSY" pin)** 1 : Set Active High. 0 : Set Active Low.	0	R/W
4	**Interrupt Polarity (for "INT" pin)** 1 : Set Active High. 0 : Set Active Low.	0	R/W
3-2	**Driver Clock Selection** These two bits are used to select the clock frequency of XCK. 0 0 : XCK = CLK/8 0 1 : XCK = CLK/4 (Default) 1 0 : XCK = CLK/2 1 1 : XCK = CLK The "CLK" means system clock.	01	R/W
1	**SEG Scan Direction**(SDIR) 0 : SEG order is 0 ~ 319. 1 : SEG order is 319 ~ 0.	0	R/W
0	**COM Scan Direction**(CDIR) 0 : COM order 0 ~ 239. 1 : COM order 239 ~ 0.	0	R/W

REG [03h] Advance Display Setup Register (**ADSR**)

Bit	Description	Default	Access
7	**Scroll Function Pending** 1 : Scroll function pending 0 : Scroll function keep active **Note:** When SCR_HV(Bit-1) and SCR_EN(Bit-0) are changed, the function does not support.	0	R/W
6-4	**Reserved**	000	R

3	**BIT_ORDER**(Set driver data output bit order) 1 : Inverse driver output data order(Bit-7 to Bit-0, Bit-6 to Bit-1 and so on) 0 : Normal Mode	0	R/W
2	**SCR_DIR**(Scroll Direction) When SCR_HV = 0(Horizontal Scroll) 0 : Left → Right. 1 : Right → Left. When SCR_HV = 1(Vertical Scroll) 0 : Top → Bottom. 1 : Bottom → Top.	0	R/W
1	**SCR_HV**(Scroll Horizontal/Vertical) 0 : Segment Scrolling(Horizontal). 1 : Common Scrolling(Vertical).	0	R/W
0	**SCR_EN**(Scroll Enable) 1 : Scroll function enable. 0 : Scroll function disable.	0	R/W

REG [0Fh] Interrupt Setup and Status Register (**INTR**)

Bit	Description	Default	Access
7	**Reserved**	0	R
6	**Wakeup Interrupt Mask** 1 : Enable wake-up Interrupt. 0 : Disable wake-up Interrupt.	0	R/W
5	**Key-Scan Interrupt Mask** 1 : Enable Key-Scan Interrupt. 0 : Disable Key-Scan Interrupt.	0	R/W
4	**Touch Panel Interrupt Mask** 1 : Generate interrupt output if touch panel was detected. 0 : Don't generate interrupt output if touch panel was detected.	0	R/W
3	**Touch Panel Event**(Only activate in TP Manual mode) 1 : Touch panel is touched. 0 : Touch panel is not touched.	0	R
2	**Wakeup Interrupt Status bit** 1 : Interrupt that indicate wake-up event happen from Sleep mode. 0 : No wake-up interrupt happen. User must write "0" to clear the Status bit.	0	R/W
1	**Key-Scan Interrupt Status bit** 1 : Key-Scan Detects Key Input. 0 : Key-Scan doesn't Detect Key Input. User must write "0" to clear the Status bit.	0	R/W
0	**Touch Panel Detect Status bit** 1 : Touch Panel Touched. 0 : Touch Panel Untouched. User must write "0" to clear the Status bit.	0	R/W

REG [10h] Whole Chip Cursor Control Register (**WCCR**)

Bit	Description	Default	Access
7	**CUR_INC** (Auto Increase Cursor Position in Reading/Writing DDRAM Operation.) 1 : Disable. 0 : Enable(Auto Increase).	0	R/W
6	**FULL_OFS (Full-size and Half-size Character Alignment)** 1 : Enable, in Full-size and Half-size character mixed mode. Chinese always start at full-size alignment. 0 : Disable.	0	R/W
5	**Reversed Data Write mode** 0 : Store Current Data to DDRAM Directly. 1 : Store Current Data to DDRAM Inversely.(i.e. 01101101➔ 10010010)	0	R/W
4	**Bold Font (Character Mode Only)** 1 : Bold Font 0 : Normal Font	0	R/W
3	**Font Rotate mode(T90DEG)** 1 : Font rotates 90 degree. (See Section 6-10-4 for detail) 0 : Normal font.	0	R/W
2	**Cursor Display** 1 : Set Cursor Display On. 0 : Set Cursor Display Off.	0	R/W
1	**Cursor Blinking** 1 : Blink Cursor. The blink time is determined by register BTMR. 0 : Normal.	0	R/W
0	**Reserved**	0	R

REG [11h] Cursor Height and Word Interval Register (**CHWI**)

Bit	Description	Default	Access
7-4	**Set Cursor Height** 0000 b → Height = 1 pixel. 0001 b → Height = 2 pixels. 0010 b → Height = 3 pixels. : : 1111 b → Height = 16 pixels. **Note:** In normal font, the cursor width fixed to one byte(8 pixels). And cursor's height is from 1~16pixels that depends on Bit[7:4]. In vertical font, the cursor height fixed to 16 pixels, and width is from 1~8 pixels that depends on Bit[6:4].	0000	R/W
3-0	**Set Line Gap** 0000 b → Gap = 1 pixel. 0001 b → Gap = 2 pixels. 0010 b → Gap = 3 pixels. : : 1111 b → Gap = 16 pixels.	0000	R/W

REG [12h] Memory Access Mode Register (**MAMR**)

Bit	Description	Default	Access
7	**Cursor Auto Shifting Direction** 0 : Cursor moves horizontally (left to right) first then vertically (top to down). 1 : Cursor moves vertically first then horizontally. **Note:** In graphic mode, the cursor moving is treated as unit of bytes in horizontal direction. At vertical direction, it's treated as unit of bit. At text mode, the bit is ignored, and the cursor moving is always in horizontal direction.	0	R/W
6-4	**Display Layer and Display Mode Selection** 0 0 0 : Gray Mode. In this mode, each pixel consists with 2 continuous bits in memory data. With the FRC methodology, 4-level-gray mode is implemented. The bit mapping is list as below. bit1 bit0 Gray 0 0 Level1 (Lightest) 0 1 Level2 1 0 Level3 1 1 Level4 (Darkest) **Note:** Gray mode doesn't support text-mode input. 0 0 1 : Show DDRAM1 data on screen. 0 1 0 : Show DRRAM2 data on screen. 0 1 1 : Show Two Layer Mode. The display rule depends on Bit-3 and Bit-2 as following. 1 0 X : NA. 1 1 0 : Extension Mode (1), the panel will show both DDRAM1 and DDRAM2 data on the screen. The RA8806 is available for 640x240 pixels panel. 1 1 1 : Extension Mode (2), the panel will show both DDRAM1 and DDRAM2 on the screen. The RA8806 is available for 320x480 pixels panel.	001	R/W
3-2	**Two Layer Mode Selection** Combine the data of DDRAM1 and DDRAM2 on the screen when Bit[6:4] is set as "011". 0 0 : DDRAM1 "OR" DDRAM2. 0 1 : DDRAM1 "XOR" DDRAM2. 1 0 : DDRAM1 "NOR" DDRAM2. 1 1 : DDRAM1 "AND" DDRAM2.	00	R/W
1-0	**MPU Read/Write Layer Selection** 0 0 : Access CGRAM.(512Byte) 0 1 : Access DDRAM1. 1 0 : Access DDRAM2. 1 1 : Access both DDRAM1 and DDRAM2 concurrently	01	R/W

REG [20h] Active Window Right Register (**AWRR**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Active Window Right Position → Segment-Right** **Note:** AWRR must be equal or larger then AWLR, and less or equal then the value 27h (40 in decimal).	27h	R/W

Note:

REG[20h, 30h, 40h, and 50h] are used to dominate an active window for line/row changing when writing data. Users can use these four registers to set the left/right/top/bottom boundary of active window. When data goes beyond the right boundary of it, the cursor will automatically change the next line to write data. It will move to the left boundary of new line in active window. When the data comes to the right-bottom corner, the next write will cause the cursor to move to the left-top corner.

REG [21h] Display Window Width Register (**DWWR**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Set Display Window Width Position → Segment-Width** **Segment-Right = (Segment Number / 8) – 1** If LCD panel resolution is 320x240, the value of the register is: (320 / 8) - 1 = 39 = 27h	27h	R/W

Note:

REG[21h, 31h] are used to set Display Window Resolution. Users can set the viewing scope of Display Data RAM. Column width (DWWR) of RA8806 can be set between 0h ~ 27h, and Row height (DWHR) can be set between 0h ~ EFh.

REG [30h] Active Window Bottom Register (**AWBR**)

Bit	Description	Default	Access
7-0	**Active Window Bottom Position → Common-Bottom** **Note:** AWBR must be equal or larger then AWTR, and less or equal then the value EFh(239 in decimal)	EFh	R/W

REG [31h] Display Window Height Register (**DWHR**)

Bit	Description	Default	Access
7-0	**Display Window Height Position → Common- Height** **Common_ Height = LCD Common Number –1** If LCD panel resolution is 320x240, the value of the register is: 240 – 1 = 239 = EFh	EFh	R/W

REG [40h] Active Window Left Register (**AWLR**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Active Window Left Position → Segment-Left** **Note:** AWLR must be equal or less then AWRR, and less then the value 27h(39 in decimal)	00h	R/W

REG [50h] Active Window Top Register (**AWTR**)

Bit	Description	Default	Access
7-0	**Active Window Top Position → Common-Top** **Note:** AWTR must be equal or less then AWBR, and less then the value EFh (239 in decimal)	00h	R/W

REG [60h] Cursor Position X Register (**CURX**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Cursor Position of Segment / RAM0 Address[4:0]** Define the cursor address of segment, a value from 0h ~ 27h(0 ~ 40 in decimal) When CGRAM write mode is selected (REG[12h] Bit[1:0] = 00b), the Bit[4:0] is the address for writing bit-map data. When create a full-size font, normally set to 0. When create an odd half-size font, normally set to 0, and set 10h for even font.	00h	R/W

REG [61h] Begin Segment Position Register of Scrolling (**BGSG**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Segment Start Position of Scrolling Mode** REG[61h] defines the start position (left boundary) of scroll window, it must be a value that less or equal to the REG[62h], which defines the end position(right boundary) of scroll window. Also it must be less then the value of 27h (40 in decimal), for the Display Data RAM limit.	00h	R/W

Note:

REG[61h, 62h, 71h, 72h] dominate a named scroll window for scroll function. They must be set before the scroll function is enable.

REG [62h] End Segment Position Register of Scrolling (**EDSG**)

Bit	Description	Default	Access
7-6	**Reserved**	00	R
5-0	**Segment End Position of Scrolling Mode** REG[62h] defines the end position(right boundary) of scroll window, it must be a value that larger or equal to the REG[61h], which defines the end position(left boundary) of scroll window. Also it must be less or equal then the value of 27h(40 in decimal), for the Display Data RAM limit.	00h	R/W

REG [70h] Cursor Position Y Register (**CURY**)

Bit	Description	Default	Access
7-0	**Cursor Position of Common / RAM0 Address[8:5]** Define the cursor address of common, a value from 0h ~ EFh(0 ~ 239 in decimal). When CGRAM write mode is selected (REG[12h] Bit[1:0] = 00b), the Bit[3:0] is indicate which font will be created. And Bit[7:4] are not available.	00h	R/W

REG [71h] Scrolling Action Range Begin Common Register (**BGCM**)

Bit	Description	Default	Access
7-0	**Common Start Position of Scrolling Mode** REG[71h] defines the begin position(top boundary) of scroll window, it must be a value that less or equal to the REG[72h], which defines the end position(bottom boundary) of scroll window. Also it must be less then the value of EFh (239 in decimal), for the Display Data RAM limit.	00h	R/W

REG [72h] Scrolling Action Range END Common Register (**EDCM**)

Bit	Description	Default	Access
7-0	**Common Ending Position of Scrolling Mode** REG[72h] defines the end position(bottom boundary) of scroll window, it must be a value that larger or equal to the REG[71h], which defines the end position(top boundary) of scroll window. Also it must be less or equal then the value of EFh (239 in decimal), for the Display Data RAM limit.	00h	R/W

REG [80h] Blink Time Register (**BTMR**)

Bit	Description	Default	Access
7-0	**Cursor Blink Time and Scroll Time** **Blinking Time = Bit[7:0] x (Frame width)** **Frame width = 1/Frame Rate** The Frame Rate is depends on the DWWR and DWHR and ITCR setting.	00h	R/W

Notes:

1. The Setting also determines the scroll moving speed.
2. The Frame width is the time that the controller scan whole panel, it depends on the system clock frequency, setting of display window, driver interface (4-bits/8-bits), Idle time (ITCR), and dual mode or gray scale mode, etc.

REG [90h] Idle Time Counter Register (**ITCR**)

Bit	Description	Default	Access
7-0	**Idle Time Setting, in count of system clock.** The value can determine the scan time of each COM of the LCD. **COM_PRD = (COM_SCAN + ITCR) x XCK_PRD** In which, **COM_SCAN = (SEG_NO/LD_WIDTH) x (1 + EXT_MD)** **XCK_PRD = 1 / XCK** **COM_PRD:** The finally scan period for each COM(Unit : ns). **COM_SCAN:** The really scan time for each COM. **XCK_PRD:** One cycle time of XCK. XCK is depends on the system clock(CLK) and REG[01h] Bit[3:2]. If system clock is 8MHz, REG[01h] Bit[3:2] = 10b, then XCK_PRD = 250ns. **SEG_NO:** Segment number, i.e. 240x160 panel, SEG_NO = 240. **EXT_MD:** In extension mode1 or 2(REG[12h] Bit[6:4] = 111b or 110b), the EXT_MD = 1, otherwise EXT_MD = 0. **LD_WIDTH:** Driver data width. If LCD driver data bus is 4-bits then LD_WIDTH = 4. If LCD driver data bus is 8-bits then LD_WIDTH = 8. Please refer pin “DW” description of Section 4-3.	00h	R/W

REG [A0h] Key-Scan Control Register 1 (**KSCR1**)

Bit	Description	Default	Access
7	**Key-Scan Enable Bit** 1 : Enable. 0 : Disable.	0	R/W
6	**Key-Scan Matrix Selection** 1 : 4x8 Matrix(KOUT[3:0] is used, KOUT[7:4] please keep floating) 0 : 8x8 Matrix(KOUT[7:0] is used)	0	R/W
5-4	**Key-Scan Data Sampling Times** De-bounce times of scan frequency. 0 0 : 4 0 1 : 8 1 0 : 16 1 1 : 32	00	R/W
3	**LNGKEY_EN : Long Time Key Function Enable** LNGKEY_EN = 0 → Long key function is disable. LNGKEY_EN = 1 → Long key function is enable.	0	R/W

<table>
<tr>
<td>2-0</td>
<td>KF2-0: Key-Scan frequency.
If system clock is 10MHz, then the related Key-Scan timing are as following:
<table>
<tr><th>KF2</th><th>KF1</th><th>KF0</th><th>Key-Scan Pulse Width (KOUT period)</th><th>Key-Scan Cycle (4x8)</th><th>Key-Scan Cycle (8x8)</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>16µs</td><td>64µs</td><td>128µs</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>32µs</td><td>128µs</td><td>256µs</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>64µs</td><td>256µs</td><td>512µs</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>128µs</td><td>512µs</td><td>1.024ms</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>256µs</td><td>1.024ms</td><td>2.048ms</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>512µs</td><td>2.048ms</td><td>4.096ms</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1.024ms</td><td>4.096ms</td><td>8.192ms</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2.048ms</td><td>8.192ms</td><td>16.384ms</td></tr>
</table>
</td>
<td>000</td>
<td>R/W</td>
</tr>
</table>

REG [A1h] Key-Scan Controller Register 2(**KSCR2**)

Bit	Description	Default	Access
7	**Key-Scan Wakeup Function Enable Bit** 0: Key-Scan Wakeup function is disable. 1: KEY-SCAN Wakeup function is enable.	0	R/W
6-4	**Reserved**	000	R
3-2	**Long Key Timing Adjustment** 00 : About 0.625sec(for 8MHz Clock source) 01 : About 1.25sec(for 8MHz Clock source) 10 : About 1.875 sec(for 8MHz Clock source) 11 : About 2.5 sec(for 8MHz Clock source)	00	R/W
1-0	**Numbers of Key Hit.** 00 : No key is pressed 01 : One key is pressed, read REG[A2h] for the key number. 10 : Two key is pressed, read REG[A2h ~ A3h] for the key number. 11 : Three key is pressed, read REG[A2h ~ A4h] for the key number.	00	R

REG [A2h ~ A4h] Key-Scan Data Register (**KSDR0 ~ 2**)

Bit	Description	Default	Access
7-0	**Key Strobe Data** The corresponding key number that is pressed. Please reference Section 6-5 "Key-Scan".	00h	R

REG [B0h] Memory Write Command Register (**MWCR**)

Bit	Description	Default	Access
7-0	**Memory data write command from the cursor position.** **Note:** Write memory data, user must write the MWCR command first, then write DATA cycle.	NA	R/W

REG [B1h] Memory Read Command Register (**MRCR**)

Bit	Description	Default	Access
7-0	**Memory data read command from the cursor position.** **Note:** Memory read cycle in text mode, the cursor move in same behavior like graphic mode. B1h will perform a pre-read function. So the cursor position will increase after the MRCR command is write.	NA	R/W

REG [C0h] Touch Panel Control Register 1 (**TPCR1**)

Bit	Description	Default	Access
7	**Touch Panel Enable Bit** 1 : Enable. 0 : Disable.	0	R/W
6-4	**TP Sample Time Adjusting** 000 : Wait 50μs for ADC data ready. 001 : Wait 100μs for ADC data ready. 010 : Wait 200μs for ADC data ready. 011 : Wait 400μs for ADC data ready. 100 : Wait 800μs for ADC data ready. 101 : Wait 1.6ms for ADC data ready. 110 : Wait 3.2ms for ADC data ready. 111 : Wait 6.4ms for ADC data ready. **Note:** When touch panel detects the Touch event, to avoid the signal instability, the sampled time is delayed to wait the signal stable. The TP Sample Time Adjusting and ADC Clock Convert Speed relation just refer to Section 6-4-3.	000	R/W
3	**Touch Panel Wake-up Enable:** 1 : Touch panel can wake-up the Sleep mode(At the condition that ADC is enabled). 0 : Disable the touch panel wake-up function	0	R/W
2-0	**ADC Clock Convert Speed** 0 0 0 : CLK / 4 0 0 1 : CLK / 8 0 1 0 : CLK /16 0 1 1 : CLK / 32 1 0 0 : CLK / 64 1 0 1 : CLK / 128 1 1 0 : CLK / 256 1 1 1 : CLK / 512 The “CLK” means system clock.	000	R/W

REG [C1h] Touch Panel X High Byte Data Register (**TPXR**)

Bit	Description	Default	Access
7-0	**Touch Panel X Data Bit[9:2](Segment)**	00h	R

REG [C2h] Touch Panel Y High Byte Data Register (**TPYR**)

Bit	Description	Default	Access
7-0	**Touch Panel Y Data Bit[9:2] (Common)**	00h	R

REG [C3h] Touch Panel Segment/Common Low Byte Data Register (**TPZR**)

Bit	Description	Default	Access
7-4	**Reserved**	0000	R
3-2	**Touch Panel Y Data Bit[1:0] (Common)**	00	R
1-0	**Touch Panel X Data Bit[1:0] (Segment)**	00	R

REG [C4h] Touch Panel Control Register 2 (**TPCR2**)

Bit	Description	Default	Access
7	**TP Manual Mode Enable** 1 : Using the manual mode. 0 : Auto mode.	0	R/W
6-2	**Reserved**	00h	R
1-0	**Mode selection for TP Manual Mode** 00 : IDLE mode: ADC idles. 01 : Wait for TP event, touch panel event could cause the interrupt or be read from REG[0Fh] B3. 10 : Latch X data, in the phase, X Data can be latched in REG[C1h] and REG[C3h]. 11 : Latch Y data, in the phase, Y Data can be latched in REG[C2h] and REG[C3h].	00	R/W

REG [D0h] PWM Control Register (**PCR**)

Bit	Description	Default	Access
7	**PWM enable** 1 : Enable 0 : Disable, PWM_OUT level depends on the REG[D0h] Bit-6.	0	R/W
6	**PWM Disable Level** 0 : PWM_OUT is Normal L when PWM disable or Sleep mode. 1 : PWM_OUT is Normal H when PWM disable or Sleep mode.	0	R/W
5-4	**Reserved**	00	R
3-0	**PWM Clock Source Divide Ratio** 0000 b → CLK / 1 0001 b → CLK / 2 0010 b → CLK / 4 0011 b → CLK / 8 : : 1111 b → CLK / 32768 The “CLK” means system clock. For example, CLK is 8MHz: 0000 b ➔ PWM clock source = 8MHz, 0001 b ➔ PWM clock source = 4MHz, : : 1111 b ➔ PWM clock source = 244Hz.	0000	R/W

REG [D1h] PWM Duty Cycle Register (**PDCR**)

Bit	Description	Default	Access
7-0	**PWM Cycle Duty Selection Bit** 00h → 1 / 256 01h → 2 / 256 High period 02h → 3 / 256 High period : : FFh → 256 / 256 High period	00h	R/W

REG [E0h] Pattern Data Register (**PNTR**)

Bit	Description	Default	Access
7-0	**Data Written to DDRAM(Display Data RAM)** The pattern that will be filled to active window in memory clear function. When REG[F0h] Bit-3 is '1', the data in the REG[E0h] will be filled to the whole active window.	00h	R/W

REG [F0h] Font Control Register (**FNCR**)

Bit	Description	Default	Access
7	**ISO8859 Mode** 0 : Disable. The contents of ASCII block 1 ~ 4 are show as Table C-1~ Table C-4 of Appendix B. 1 : Enable. The ASCII block 1 ~ 4 indicate the ISO8859-1 ~ 4 standard and show as Table C-5 ~ Table C-8 of Appendix C.	0	R/W
6-4	**Reserved**	000	R
3	**Memory Clear Function** Write Function 0 : No Action. 1 : Memory clear function active, fill the data of FNTR to Active window. When this bit is set to "1", RA8806 will automatically read PNTR data, and fill it to Active window (Range: [AWLR, AWTR] ~ [AWRR, AWBR]), after clear completed, this bit will be cleaned to "0".	0	R/W
2	**ASCII Mode Enable** 1 : All input data will be decoded as ASCII (00h ~ FFh) 0 : In text mode (REG[00h] Bit-3), the RA8806 will check the first written byte data first. If less then 80h then it's treated as ASCII (Half-size). Or it's treated as a full-size text(GB, BIG5 or User-created font).	0	R/W
1-0	**ASCII Blocks Select** 0 0 : Map to ASCII block 1. (Table C-1 and Table C-5 of Appendix C.) 0 1 : Map to ASCII block 2. (Table C-2 and Table C-6 of Appendix C.) 1 0 : Map to ASCII block 3. (Table C-3 and Table C-7 of Appendix C.) 1 1 : Map to ASCII block 4. (Table C-4 and Table C-8 of Appendix C.)	00	R/W

REG [F1h] Font Size Control Register (**FVHT**)

Bit	Description	Default	Access
7-6	**Set Character Horizontal Size** 0 0 : One Time of normal font width. 0 1 : Two Times of normal font width. 1 0 : Three Times of normal font width. 1 1 : Four Times of normal font width.	00	R/W
5-4	**Set Character Vertical Size** 0 0 : One Time of normal font height. 0 1 : Two Times of normal font height. 1 0 : Three Times of normal font height. 1 1 : Four Times of normal font height.	00	R/W
3-0	**Reserved**	0000	R

6. Function Description

6-1 MPU Interface

6-1-1 MPU Type

The RA8806 support 8080 or 6800 compatible MPU interface. When the pin “MI” is pull low then the MPU interface is set to 8080 compatible. If “MI” pulls high then the MPU interface is defined as 6800 compatible. And the pin “DB” is used to select the 8080 MPU data bus is 4-bits or 8-bits. When “DB” is pulled low, then the data bus for data transition is 4-bits. If pin “DB” pull high, the data transition is 8-bits. The option of 4-bits or 8-bits data bus is for both 8080 and 6800 MPU. Of course, if used 4-bits interface then the 8080 MPU has to take double time to communicate with RA8806. In order to reduce the transmission line interference between MPU interface and RA8806, we suggest that a small capacitor to the GND should be added at the signal of ZCS1、ZRD、ZWR, please see the following Figure 6-1.

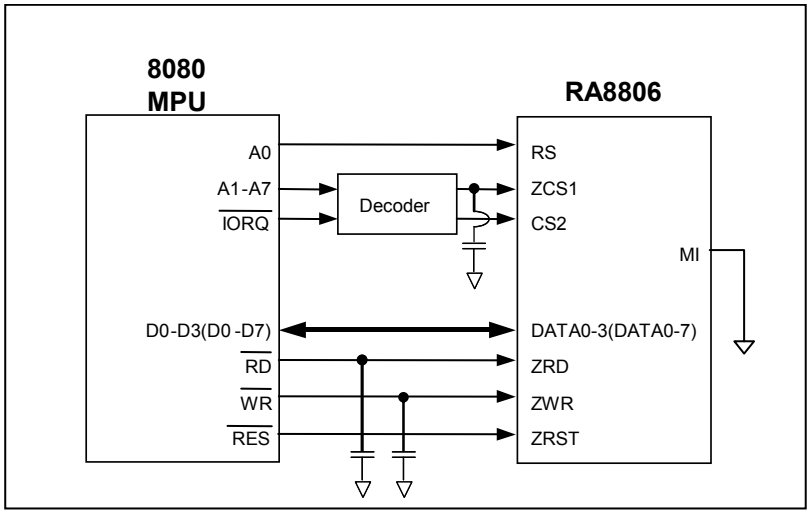


Figure 6-1 : 8080 (4/8-bits) MPU Interface

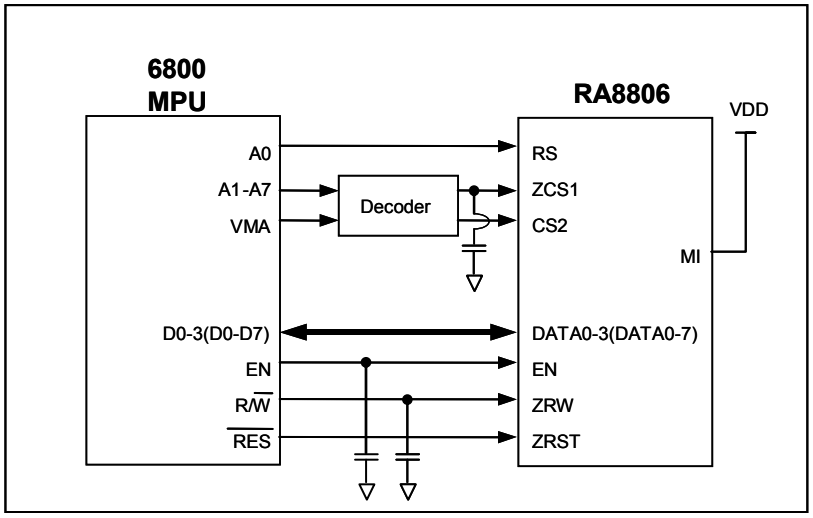


Figure 6-2 : 6800 (4/8-bits) MPU Interface

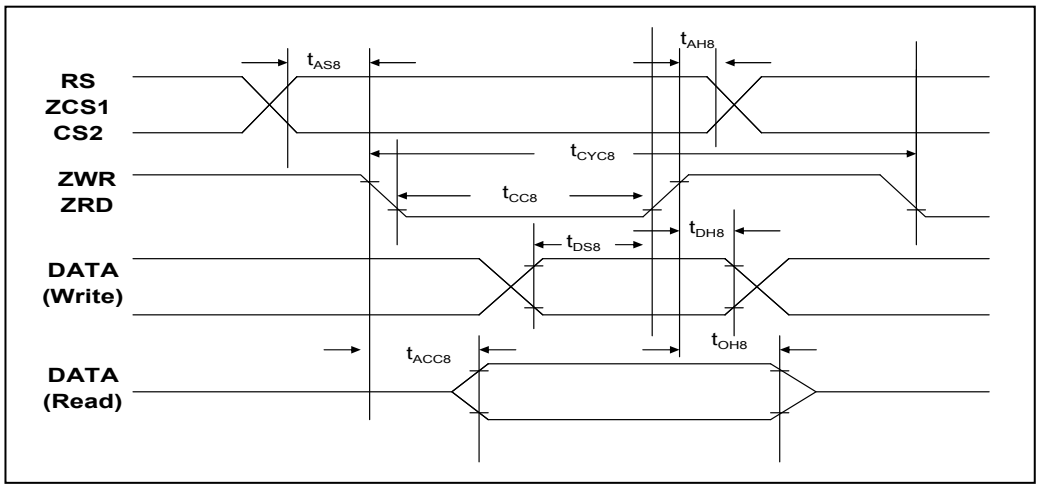


Figure 6-3 : 8080 MPU Interface Waveform

Table 6-1 : 8080 MPU Interface Timing

Symbol	Description	Rating		Unit	Condition
		Min.	Max.		
t_{CYC8}	Cycle time	2*tc	--	ns	tc = one system clock period
t_{CC8}	Strobe Pulse width	50	--	ns	
t_{AS8}	Address setup time	0	--	ns	
t_{AH8}	Address hold time	20	--	ns	
t_{DS8}	Data setup time	30	--	ns	
t_{DH8}	Data hold time	20	--	ns	
t_{ACC8}	Data output access time	0	20	ns	
t_{OH8}	Data output hold time	0	10	ns	

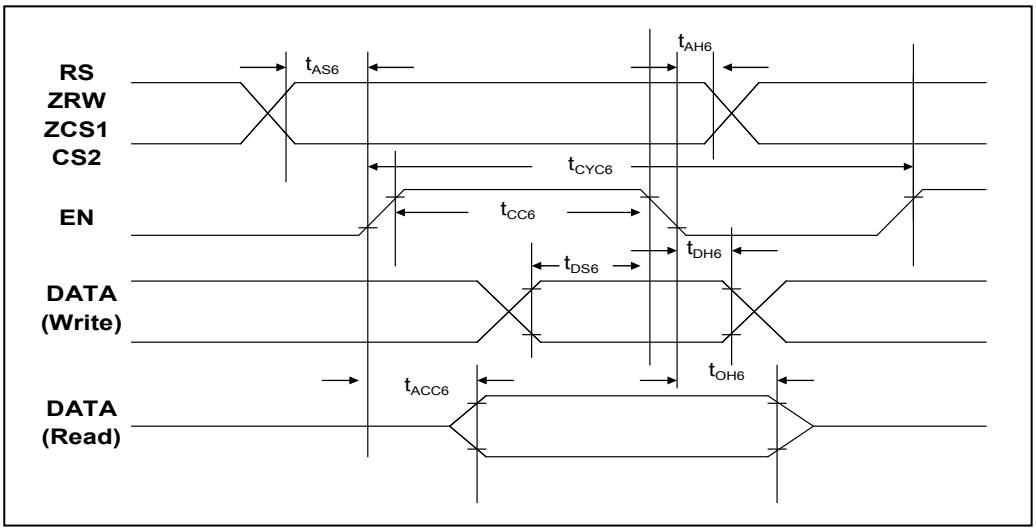


Figure 6-4 : 6800 MPU Interface Waveform

Table 6-2 : 6800 MPU Interface Timing

Symbol	Description	Rating		Unit	Condition
		Min.	Max.		
t_{CYC6}	Cycle time	2*tc	--	ns	tc is one system clock period: tc = 1/CLK
t_{CC6}	Strobe Pulse width	50	--	ns	
t_{AS6}	Address setup time	0	--	ns	
t_{AH6}	Address hold time	20	--	ns	
t_{DS6}	Data setup time	30	--	ns	
t_{DH6}	Data hold time	20	--	ns	
t_{ACC6}	Data output access time	0	20	ns	
t_{OH6}	Data output hold time	0	10	ns	

6-1-2 Command Write

According to the Table 5-1, RA8806 accept 4 cycles through MPU interface. If users want to write command to RA8806, then a Command cycle has to execute first, and then execute a Data Write cycle. The “Command Write” means write function data to register. After these two cycles, the Data will write into the indicative Register. Please see the following Figure 6-5 (1).

In Table 6-1 of Section 6-1-1, each command of RA8806 is take 2 cycles, and the minimum cycle time is 2*tc. So totally the minimum time of command write need 4*tc. See following Table 6-3.

If the secondary cycle is a “Data Read”, then user could read the register content. See the following Figure 6-5 (2). Note the Figure 6-5 to Figure 6-7 are use the 8080 MPU interface as examples.

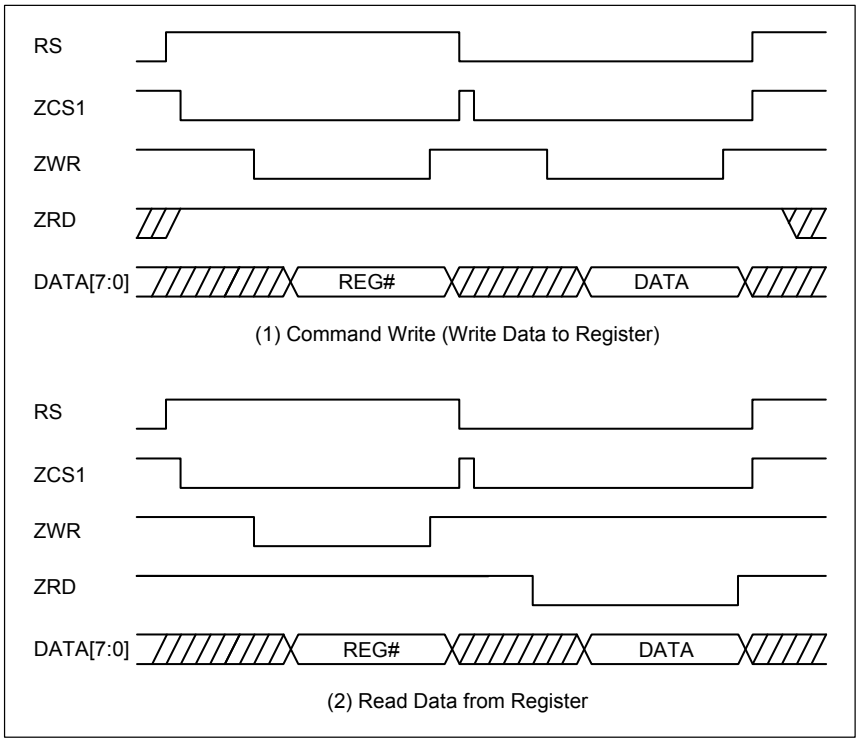


Figure 6-5 : Command Write and Register Read Cycle

Table 6-3 : Command Access Time Table

System Clock	Command Access Time
4MHz	1μs
6 MHz	667ns
8 MHz	500ns
10 MHz	400ns
12 MHz	333ns

6-1-3 Memory Write/Read

When users want to write data to memory – DDRAM or CGRAM, then a special Command cycle has to execute first, the register have to assign to “B0h” on Data Bus. Then the following Data Write cycle will write data into memory. If users want to read data from memory, then the register has to assign to “B1h” on Data Bus in Command Write cycle. Please see the following Figure 6-6 (1) and (2).

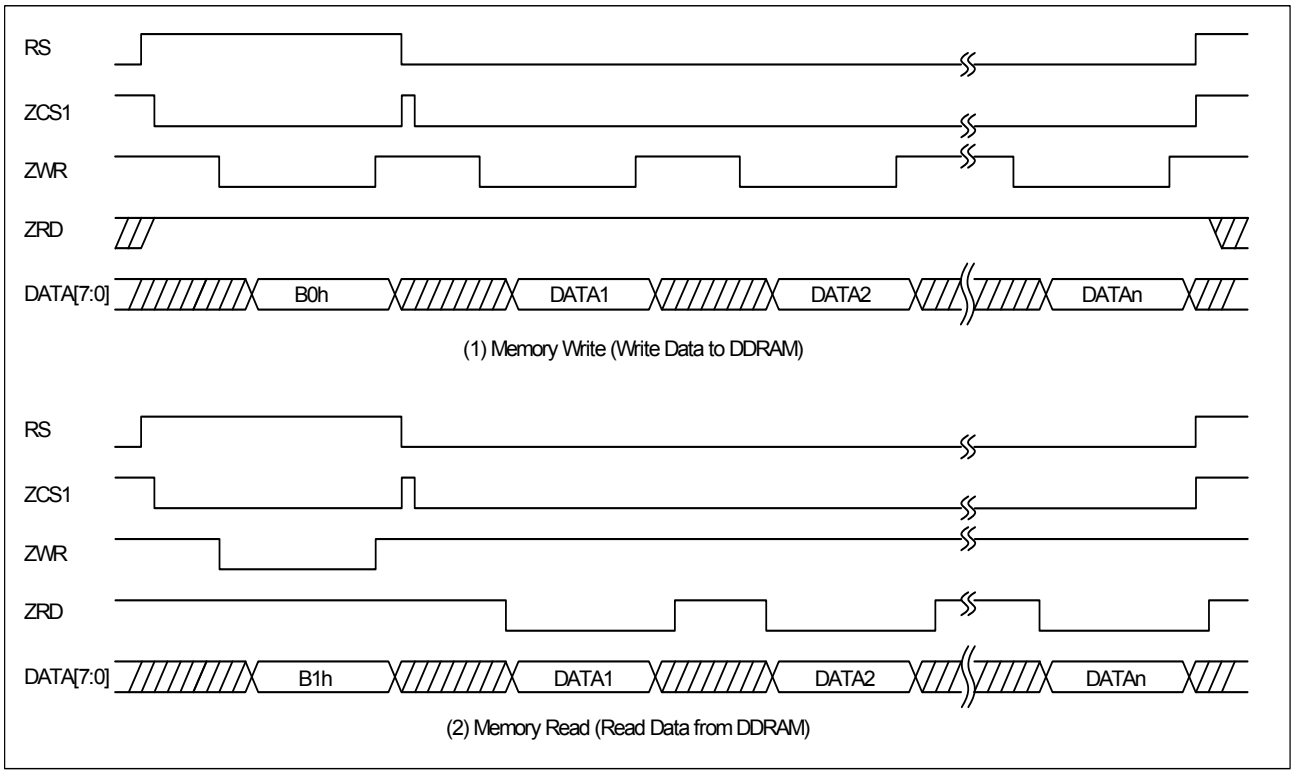


Figure 6-6 : Memory Write/Read Cycle

6-1-4 Status Read

RA8806 provides a dedicate Status Read cycle to help users know the status of RA8806. Please refer to following Figure 6-7 and the beginning of Section 5-2 “Register Description”.

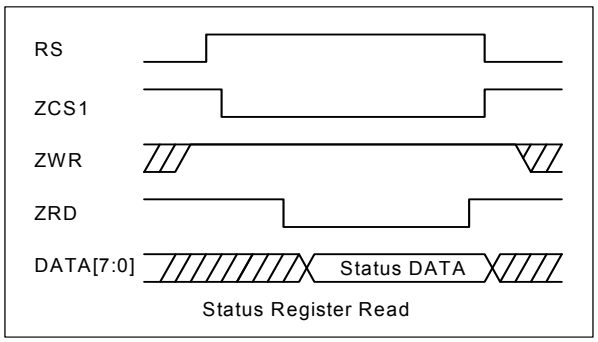


Figure 6-7 : Status Read Cycle

6-2 Driver Interface

The main function of Driver Interface is to generate Frame (Pin “FR”), Latch Pulse (Pin “LP”), “YD” and Data Bus (LD[7:0]) for external LCD driver IC. RA8806 could both support 4-bits and 8-bits LCD driver interface. Pin “DW” is for LCD driver data bus selection. If “DW” pulls high then 8-bits LCD driver is used. If pull low then 4-bits LCD driver is used. Figure 6-8 is the LCD interface of RA8806.

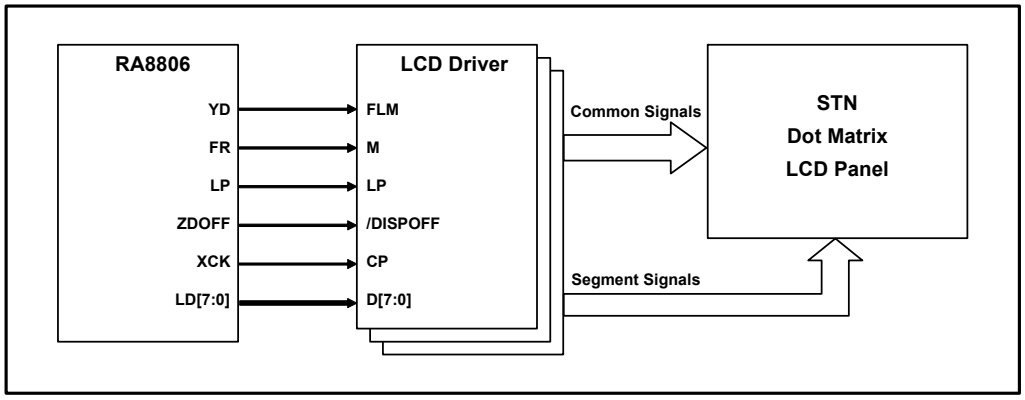


Figure 6-8 : The Interface of RA8806 and LCD Driver

Figure 6-9 is the timing waveform of RA8806 and LCD Driver. Users could also refer to Section 4-4 “LCD Driver Interface” for LCD driver pin description.

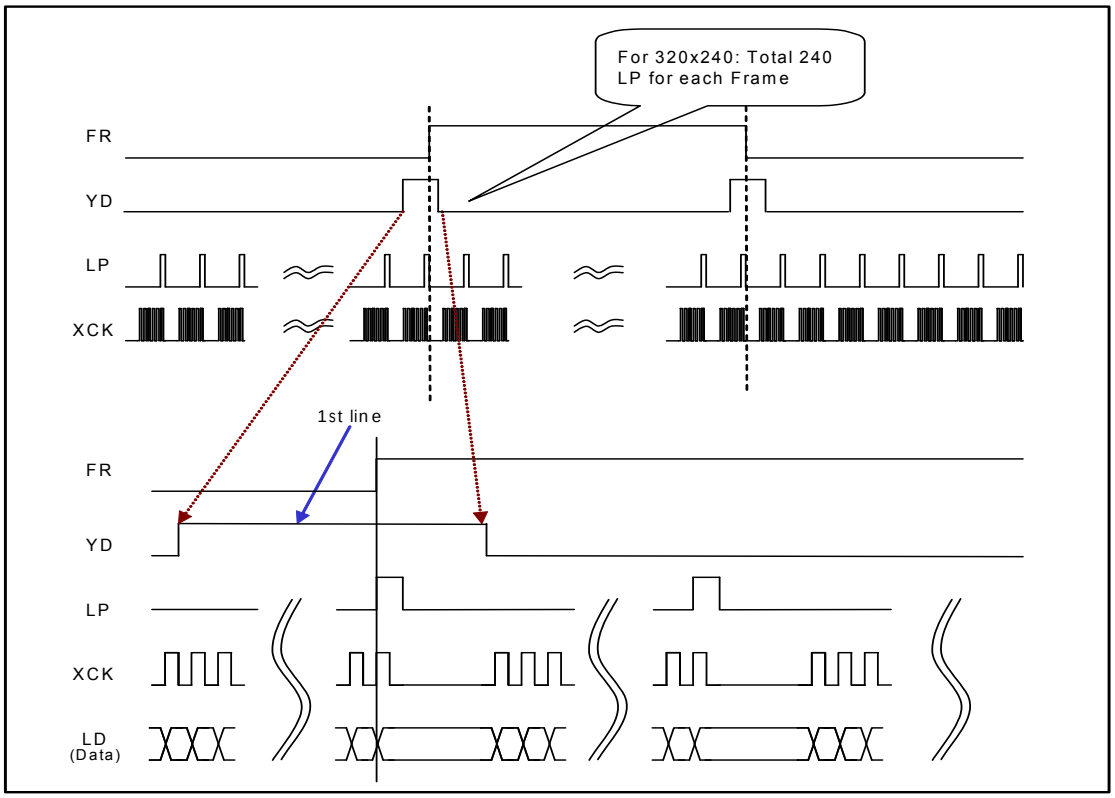


Figure 6-9 : The Waveform of RA8806 and LCD Driver

Figure 6-10 is the application diagram of RA8806. In this example, we use 80-channels LCD Drivers to process Common and Segment activity of 320x240 LCD Panel. RA8806 send FR, LP, YD, Data Bus(LD[3:0] and clock(XCK) signals to Segment/Common drivers. The Figure 6-11 is the example for 160x160 panel application.

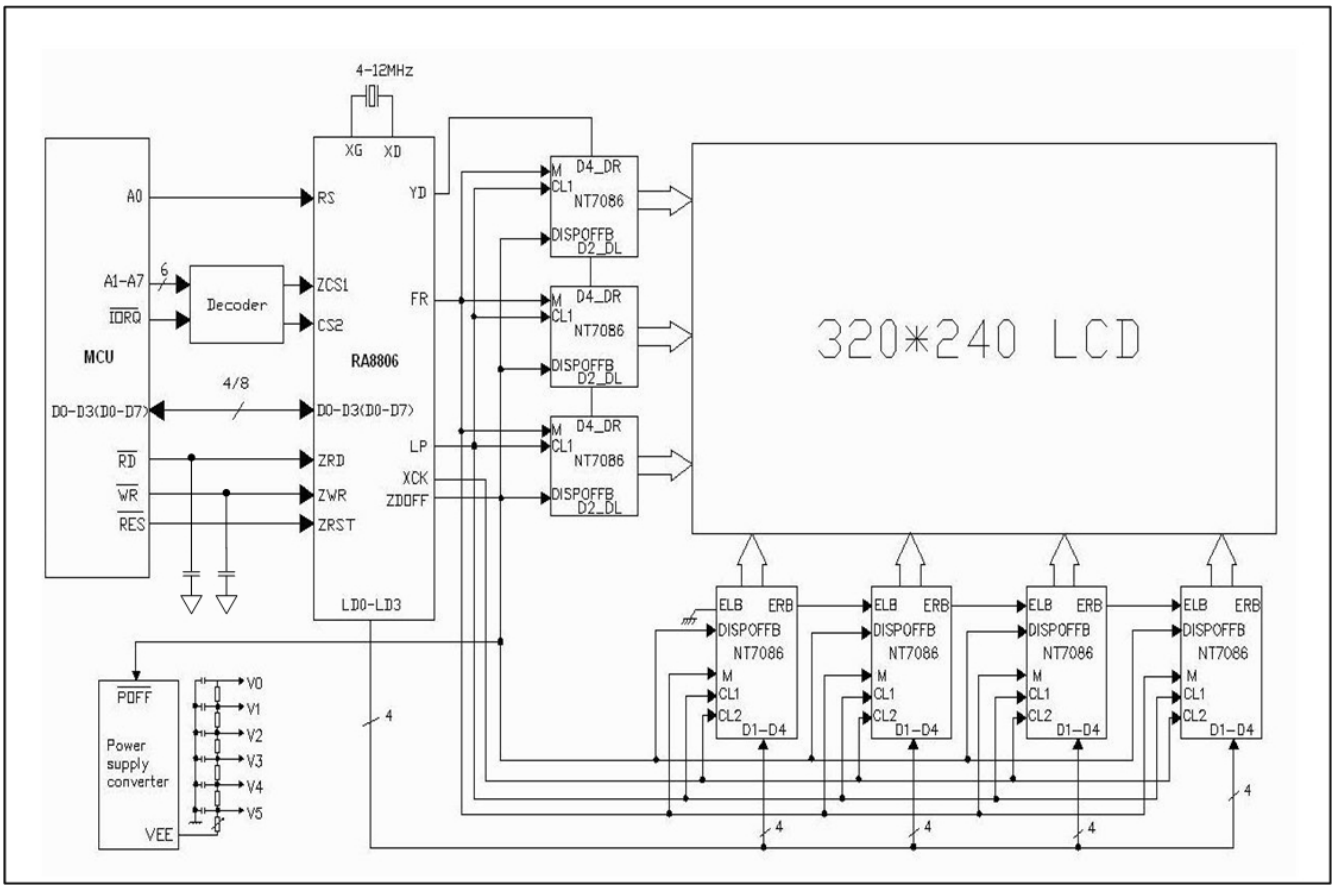


Figure 6-10 : Application Circuit Diagram for 320x240 Panel

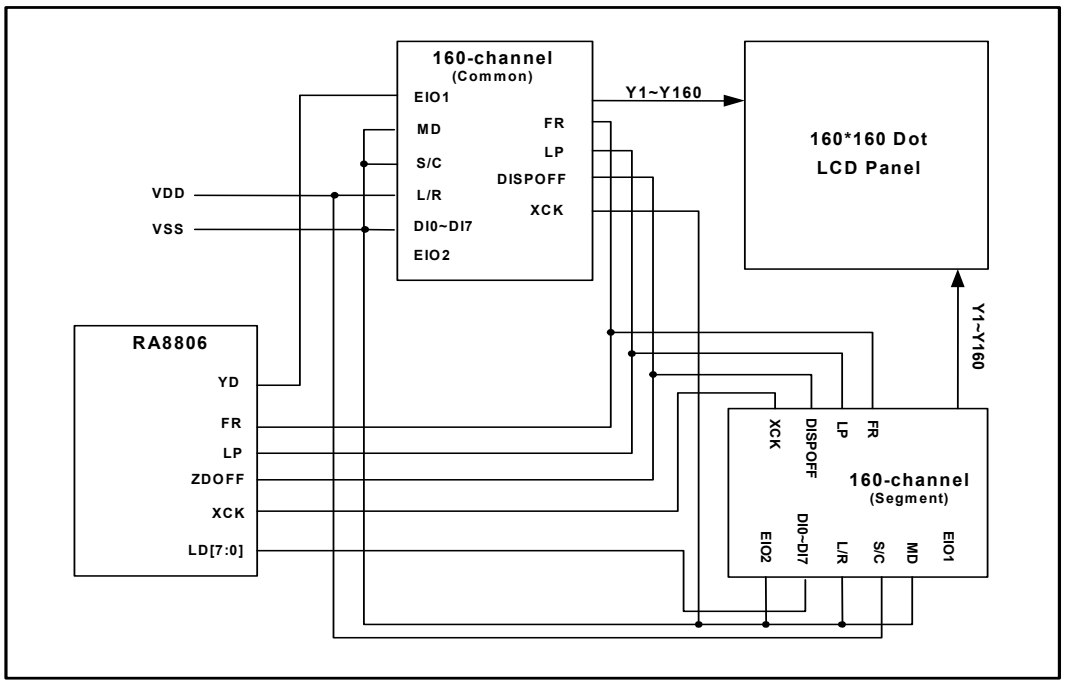


Figure 6-11 : Application Circuit Diagram for 160x160 Panel

Table 6-4 is the RA8806 driver signals mapping to different driver ICs' pin names. **Note, do not need add any capacitor on these for LCD driver signals. If add external capacitor on these pins to ground or VDD, then the maximum limitation is 30pF.**

Table 6-4 : Driver IC I/F Name vs. RA8806

RA8806 Driver I/F	Driver IC I/F Name	Definition of Driver IC I/F
LP	LP	Data Latch Clock Latch Pulse in one line
	LOAD	Latch pulse of display data
	CL1	Data Latch Pulse
XCK	CP	Data Shift Clock Clock pulse for segment shift register
	SCP	Shift Clock Pulse for X-Drivers
	CL2	Data Shift Pulse
	HSCP	Shift Clock Pulse
YD	FLM	Scan Start-up Signal First Line Marker
	FR	Frame Pulse
	FRAME	Frame start signal(First line mark of common signal)
	CDATA	Synchronous Data
FR	DF(M)	Switch signal to convert LCD drive waveform into AC
LD[7:0]	D[7:0]	LCD Data Bus
ZDOFF	/DISPOFF	Display OFF
	/D.OFF	Display OFF
	DISP	Display OFF

6-2-1 Display Resolution

RA8806 support many different resolution of LCD panel. For different resolution of panel, RA8806 could change the setting of some registers like DWWR and DWHR to modify display window size. And use registers AWRR, AWBR, AWLR and AWTR to change the active window size.

For example, if 320x240 LCD panel is used, then the related register setting is as following:

DWWR = (320 / 8) - 1 = 39 = 27h
DWHR = 240 – 1 = 239 = EFh

The active window range is less than display window. So user has to care the rule as following:

1. DWWR ≥ AWRR ≥ AWLR
2. DWHR ≥ AWBR ≥ AWTR

RA8806 supports a variety of LCD modules, the setting of register depnding on different resolution of LCD module is list at below table.

Table 6-5 : Registers Setting for LCM Resolution

Panel Resolution	Segment	Common	REG[21h] DWWR	REG[31h] DWHR
160*80	160	80	13h	4Fh
160*128	160	128	13h	7Fh
160*160	160	160	13h	9Fh
240*64	240	64	1Dh	3Fh
240*128	240	128	1Dh	7Fh
240*160	240	160	1Dh	9Fh
320*240	320	240	27h	EFh

6-2-2 Display Window and Active Window

The RA8806 provides two windows for real application -- Display Window and Active Window. The Display Window is the actual resolution of LCD panel. Active is a sub-window in Display Window. The boundary of cursor shift depends on the active window. The relative registers of the two windows are as the following table.

Table 6-6

Reg.	Bit_Num	Description	Reference
AWLR	Bit [5:0]	Define left boundary of the active window	REG[40h]
AWRR	Bit [5:0]	Define right boundary of the active window	REG[20h]
AWTR	Bit [7:0]	Define top boundary of the active window	REG[50h]
AWBR	Bit [7:0]	Define bottom boundary of the active window	REG[30h]
DWWR	Bit [5:0]	Define the width of the display window	REG[21h]
DWHR	Bit [5:0]	Define the height of the display window	REG[31h]

For RA8806, if LCD panel resolution is 320x240 pixels then the display window resolution is 320x240. We can create an active window in the display window like Figure 6-12. This figure show the display resolution is 320x240, and a 160x160 active window is on the upper-middle. The relative setting of the active window as following:

```
LCD_CmdWrite ( 0x40 );      // AWLR = 09h = 9 ➔ ( 80 / 8) – 1
LCD_DataWrite ( 0x09 );

LCD_CmdWrite ( 0x20 );      // AWRR = 1Dh = 29 ➔ ( 240 / 8 ) – 1
LCD_DataWrite ( 0x1D );

LCD_CmdWrite ( 0x50 );      // AWTR = 00h = 0
LCD_DataWrite ( 0x00 );

LCD_CmdWrite ( 0x30 );      // AWBR = 9Fh = 159 ➔ 160 – 1
LCD_DataWrite ( 0x9F );

LCD_CmdWrite ( 0x21 );      // DWWR = 27h = 39 ➔ ( 320 / 8 ) – 1
LCD_DataWrite ( 0x27 );

LCD_CmdWrite ( 0x31 );      // DWHR = EFh = 239 ➔ 240 – 1
LCD_DataWrite ( 0xEF );
```

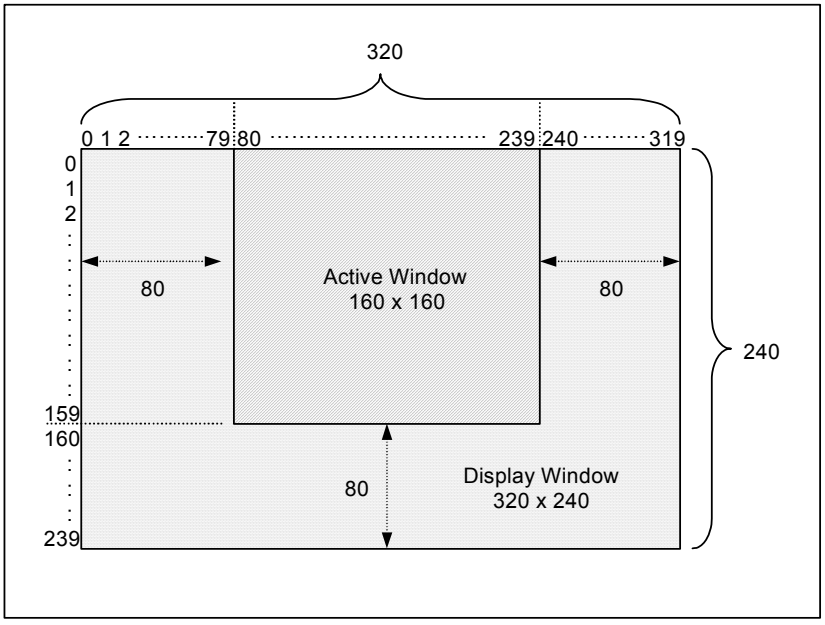


Figure 6-12 : RA8806 Display Window and Active Window

For RA8806, if LCD resolution is 240x160 pixels then the display window resolution is 240x160. We can create an active window in the display window like Figure 6-13. This figure show the display resolution is 240x160, and a 120x120 active window is on the upper-left. The relative setting of the active window as following:

```
LCD_CmdWrite ( 0x40 );      // AWLR = 00h = 0
LCD_DataWrite ( 0x00 );

LCD_CmdWrite ( 0x20 );      // AWRR = 0Eh = 14 ➔ ( 120 / 8 ) – 1
LCD_DataWrite ( 0x0E );

LCD_CmdWrite ( 0x50 );      // AWTR = 00h = 0
LCD_DataWrite ( 0x00 );

LCD_CmdWrite ( 0x30 );      // AWBR = 77h = 119 ➔ 120 – 1
LCD_DataWrite ( 0x77 );

LCD_CmdWrite ( 0x21 );      // DWWR = 1Dh = 29 ➔ ( 240 / 8 ) – 1
LCD_DataWrite ( 0x1D );

LCD_CmdWrite ( 0x31 );      // DWHR = 9Fh = 159 ➔ 160 – 1
LCD_DataWrite ( 0x9F );
```

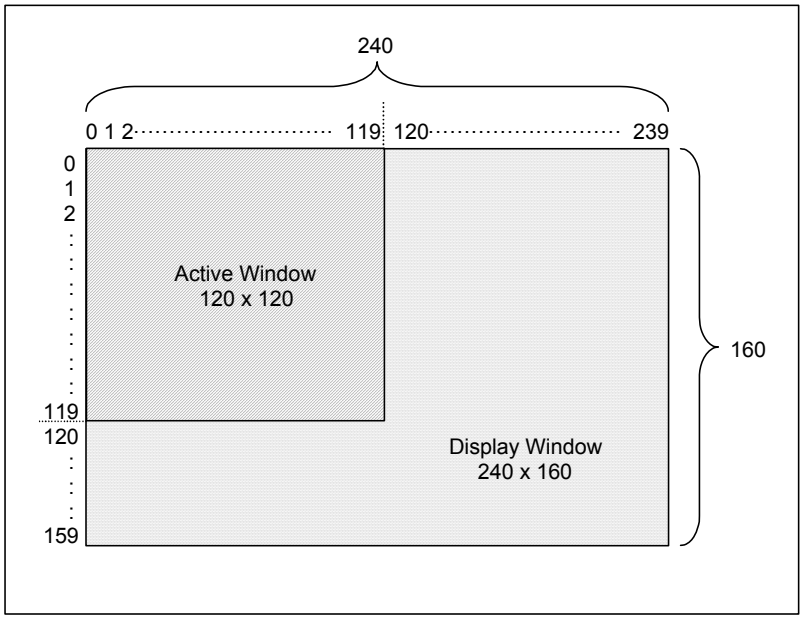


Figure 6-13 : Display Window and Active Window

6-2-3 Com/Seg Scan Direction

RA8806 supports a special feature to reverse the Common and Segment signal. User could use this feature to change the direction of cursor and display data. If use text rotate function and set CDIR = 1, then use could show text in vertical display. Please refer to Section 6-10-4.

Table 6-7

Reg.	Bit_Num	Description	Reference
MISC	Bit 1	Define the order of Segment signals. (SDIR)	REG[01h]
	Bit 0	Define the order of Common signals. (CDIR)	

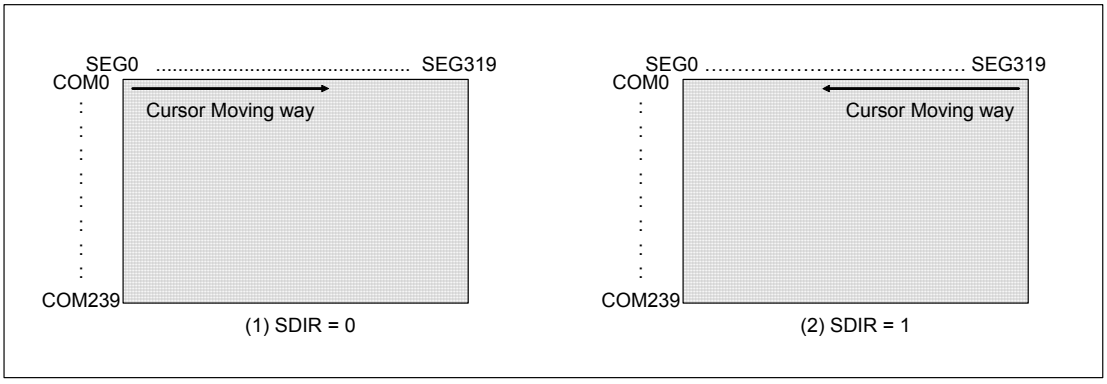


Figure 6-14 : Example for Change Segment Direction

6-2-4 Idle Time Counter (ITCR)

ITCR(REG[90h]) is used to determine the idle time during the LP peer-to-peer time. It has following meanings in function.

1. Adjusting the Frame Rate.(By extending the scan time of each COM)
2. Avoiding the generation of “Flicker”.

The “Flicker” is generated by the violation of LCD scan cycle and Memory write cycle. “Flicker” means the noise of the scan data at such violation. By setting the ITCR, user can write the display memory only at “Idle” time to eliminate the “Flicker”.

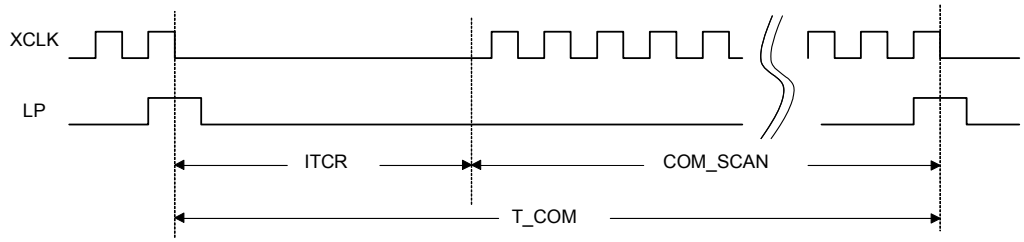


Figure 6-15 : Idle Time Period

RA8806 scan time of each COM line can be calculated by the formulas.

COM_PRD = ((SEG_NO/LD_WIDTH) x (1 + EXT_MD)) + ITCR) x XCK_PRD

In which the EXT_MD means extension mode is set or not, if REG[12h] Bit[6:4] = 110b/111b, EXT_MD = 1, or EXT_MD = 0. The "XCK_PRD" is one clock period of XCK. The XCK frequency is base on the system clock. It depends on the setting of Bit[3:2] of REG[MISC]. As to the Frame time and frame rate calculation. It would be:

FRM_PRD = COM_PRD x COM#

And

FRM_Rate = 1 / FRM_PRD

For example, when panel size is set to 320x240, system clock frequency is 8MHz, REG[MISC] Bit[3:2] = 10b, LCD driver data bus width is 4-bits, what the frame rate would be?

The System Clock(CLK) is 8MHz, and REG[MISC] Bit[3:2] = 10b → XCK = CLK/2, so the XCK_PRD is 250ns.

XCK_PRD = 1 / (CLK/2) = 1/4MHz = 250ns

COM_PRD = (320 / 4 + ITCR) x XCK_PRD = (80+ ITCR) x 250(ns)

If the ITCR = A0h(160 in decimal):

COM_PRD = (80+160) x 250ns = 240 x 250ns = 60μs

The COM number is 240, so the frame period is:

FRM_PRD = 60μs x 240 = 14.4 ms

And the frame rate is:

Frame Rate = 1 / 14.4 ms = 69.4 Hz

We can see the effect that the ITCR setting to the corresponding frame rate. So we can use it to adjust the frame rate. Please refer to Appendix B - the Table B-1 ~ Table B-3. In those tables, we show some Frame Rate setting for difference resolution, system clock. But note the display quality is also depends on the module design and the material of liquid crystal.

Table 6-8

Reg.	Bit_Num	Description	Reference
ITCR	Bit [7:0]	Define the idle time during the LP peer-to-peer time.	REG[90h]
MISC	Bit [3:2]	Select the clock frequency of XCK.	REG[01h]

6-3 Display Data RAM (DDRAM)

The RA8806 embedded two 9.6Kbyte Display Data RAMs – DDRAM1 and DDRAM2. It can use for two layers mono-display or one layer 4-gray-levels display. The maximum resolution of supporting LCD panel is 320Column x 240Row. There are two modes to write the DDRAM, text mode and graphics mode. It provides a flexible and easy way to make the display.

6-3-1 Display Layer and Display Mode Selection

There are 4 possible displays way at 2 Modes combination.

1. ***Only show DDRAM1 or DDRAM2:*** One DDRAM data will show on the screen, and another one DDRAM can be standby or as a CGRAM to created special font or symbol by users. Please refer to Section 6-11 “User Defined Font” for detail description.

2. ***Two Layer Mode:*** In this mode, the screen will show the combination data of DDRAM1 and DDRAM2. There are 4 types combination that set by REG[12H] Bit[3:2].

DDRAM1 “OR” DDRAM2
DDRAM1 “XOR” DDRAM2
DDRAM1 “NOR” DDRAM2
DDRAM1 “AND” DDRAM2

For detail please refer to Section 6-10-1-5 “Two Layer Display”.

3. ***4-Gray-level Mode:*** In gray mode, each pixel consists with 2 continuous bits in both DDRAM.

4. ***Extension Mode:*** Two extension modes is support, and both DDRAM1 and DDRAM2 data will show on the screen.

Horizontal extension mode (Max. resolution is 640 x 240 pixels).
Vertical extension mode (Max. resolution is 320 x 480 pixels).

6-3-2 Access Memory Selection

There are one 512Byte CGRAM and two 9.6KByte DDRAMs in RA8806. The 512Byte CGRAM is used to generate user-define font, and the two DDRAM are used to store the display data. One of both DDRAM can be also store special font or symbol, that when only one DDRAM is active on the screen. Which RAM is available for MPU access is depending on the REG[12h] Bit[1:0]. Please refer to Section 5-2 “Register Description”.

Table 6-9

Reg.	Bit_Num	Description	Reference
MAMR	Bit [6:4]	Display Layer and Display Mode Selection	REG[12h]
	Bit [3:2]	Two Layer Mode Selection	
	Bit [1:0]	MPU Read/Write Memory Selection	

6-4 Touch Panel

RA8806 built in a 10-bit ADC and control circuits to connect with 4-wires resistance type touch panel. Resistive Touch Panel is composed of two layer extremely thin resistive panel, such as Figure 6-16. There is a small gap between these two-layer panels. When external force press a certain point, the two-layer resistive panels will be touched, which is short. Because the end points of two-layer have electrodes (XL, XR, YU, YD), such as Figure 6-17, a comparative location will be detected with some switches in coordination.

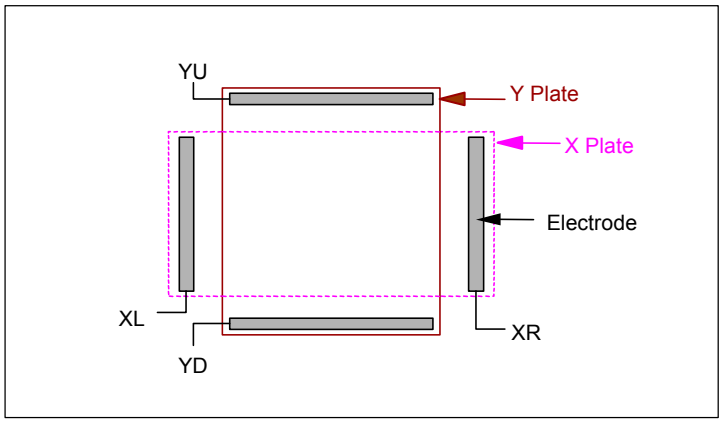


Figure 6-16：Touch Panel

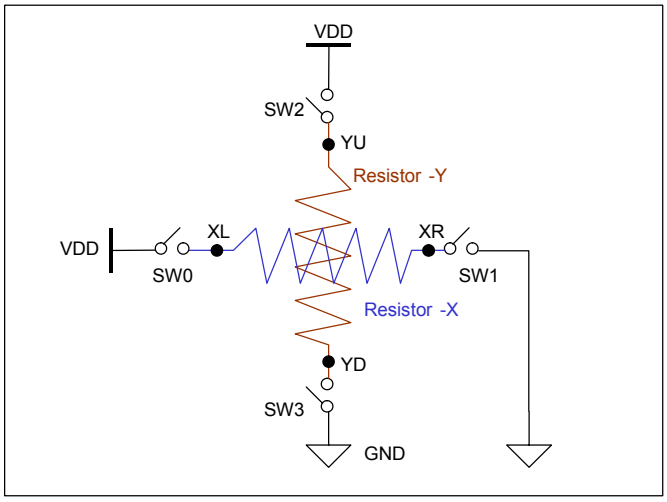


Figure 6-17 : Control Switch of Touch Panel

Users only need to connect the touch panel signals -- XL, XR, YU, and YD to RA8806. It will continuously monitor the panel and wait for touch event. When the event is occurred, a divided voltage on panel caused by touch is sensed and transferred by ADC to determine the location. After the value of X-axis and Y-axis are transferred and stored in corresponding registers respectively, the touch panel controller will issue an interrupt to inform MPU to process it.

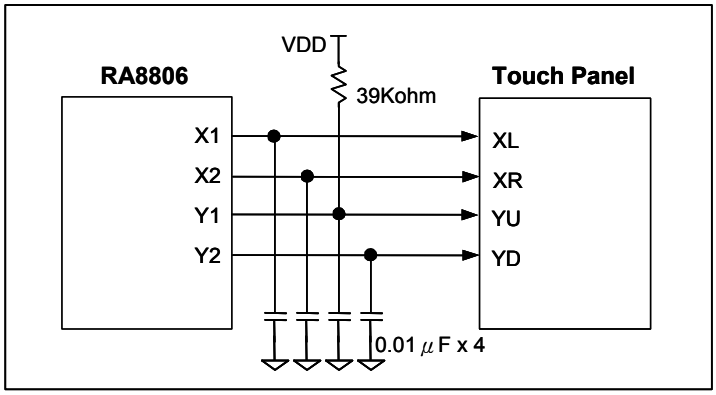


Figure 6-18 : RA8806 Touch Panel Circuit

RA8806 provides 2 modes (Auto and Manual) for touch panel application.

Table 6-10

Operation mode	Event detection	Description
Auto	Interrupt	When touch event happens, Read the corresponding X, Y coordination.
Manual	Interrupt	When touch event happens, Read the corresponding X, Y coordination.
	Polling	Polling the touch event, and read the corresponding X, Y coordination.

6-4-1 Auto Mode

Auto mode is the easiest way to implement touch panel application. Please refer to the flow chart below.

(1) Flowchart:

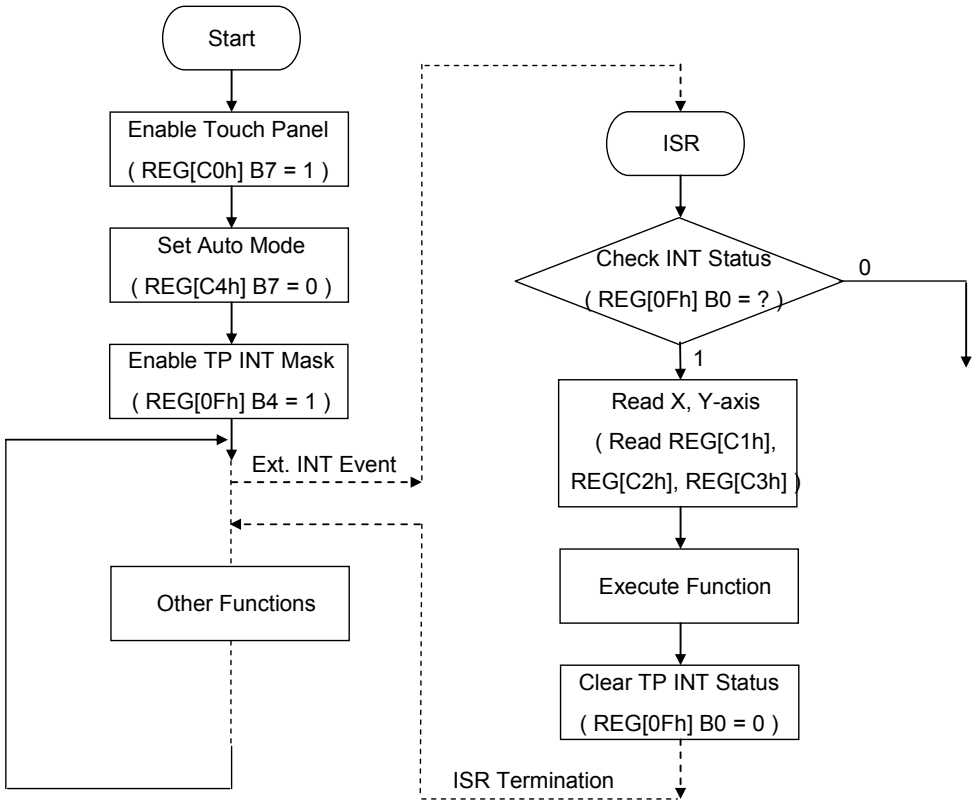


Figure 6-19 : Auto Mode Flowchart for Touch Screen

Table 6-11 lists the used registers.

Table 6-11

Reg.	Bit_Num	Description	Reference
TPCR1	Bit 7	EnableTouch Panel function	REG[C0h]
TPCR2	Bit 7	“Auto-Mode” or “Manual Mode” selection bit	REG[C4h]
INTR	Bit 4	Touch Panel Hardware Interrupt enable bit	REG[0Fh]
	Bit 0	Touch event status bit	
TPXR	Bit [7:0]	Touch Panel SEG data MSB byte	REG[C1h]
TPYR	Bit [7:0]	Touch Panel COM data MSB byte	REG[C2h]
TPZR	Bit [3:2]	Touch panel COM data LSB 2bit	REG[C3h]
	Bit [1:0]	Touch panel SEG data LSB 2bit	

(2) Program Example:

```
Unsigned char  X1,X2,Y1,Y2;
Touch_Panel_Enable ( );                          // Set TPCR1 Bit 7 to 1
TP_Auto_Enable ( );                              // Set TPCR2 Bit 7 to 0
TP_INT_Mask_Enable ( );                          // Set INTR Bit-4 to 1
        :
        :
      Execute other function                     // Jump to ISR when interrupt
        :
        :

Int  EXT_INT_Service_Routine                     // ISR entry
{
      LCD_CmdWrite ( INTR );                     // Check INT status
      INT_Sta  =  LCD_DataRead ( );
      If ( INT_Sta  &  0x01 )                    // Check If TP interrupt
      {
            LCD_CmdWrite(TPXR);
            X1 = LCD_DataRead( );                // MSB of X
            LCD_CmdWrite(TPYR);
            Y1 = LCD_DataRead( );                // MSB of Y
            LCD_CmdWrite(TPZR);
            X2 = LCD_DataRead( )  &  0x03;       // LSB two Bits of X
            LCD_CmdWrite(TPZR);
            Y2 = LCD_DataRead( )  &  0x0C;       // Least two Bits of Y
                  :
                  :
             Execute corresponding function
                  :
                  :
             LCD_CmdWrite ( INTR );              // Clear Touch Panel status
             temp  =  LCD_DataRead ( ) &  0xfe;
             LCD_CmdWrite ( INTR );
             LCD_DataWrite ( temp );
       }
 Else if (INT_Sta  &  0x02)                      // Check if Key-Scan interrupt
      {
            :
            :
      }
 Else if (INT_Sta  &  0x04)                      // Check if Wakeup interrupt
     {
            :
            :
      }
 }
```

6-4-2 Manual Mode

The " Manual Mode" means that the operation process from "Touch event checking function" to "input Latch X data Y data", the whole operation and setting process (includes TPCR2[1:0]) and receiving data from XY coordinates are manual operated by programmer. The advantage of using Manual Mode is it allows programmer more flexible applications. In the condition that is over the range of RA8806 register setting, the user can still use the software method to control the TP function in a correct way.

Touch Event can be detected from "Interrupt Mode" or "Polling Mode" that depend on the system configuration. The difference between the "Interrupt Mode" and "Polling Mode" are explained as following.

6-4-2-1 External Interrupt Mode

Under the "Interrupt Mode" the touch event detecting way is almost the same as " Auto Mode". The major processes are list as follows:

1. Enable Touch Panel function.
2. Change mode to "Manual mode".
3. Set the switch to 「Wait for touch event 」, Set TPCR2[1:0] to 01b.
4. When interrupt asserts, check if TP interrupt.
5. If yes, change the switch to 「Latch X data」, Set TPCR2[1:0] to 10b, wait for enough time to make the latch data stable and latched to TPXR and TPZR.
6. Change the switch to「Latch Y data」, Set TPCR2[1:0] to 11b, wait for enough time to make the latch data stable and latched to TPYR and TPZR.
7. Read X, Y data from TPXR, TPYR and TPZR, and clear the interrupt status.

The registers for Interrupt Mode are explained as below:

Table 6-12

Reg.	Bit_Num	Description	Reference
TPCR1	Bit 7	Enable Touch Panel function	REG[C0h]
TPCR2	Bit 7	TP Manual mode enable	REG[C4h]
	Bit [1:0]	Mode selection for TP manual mode	
INTR	Bit 4	Touch Panel Interrupt Mask	REG[0Fh]
	Bit 0	Touch Panel Detect Status bit	
TPXR	Bit [7:0]	Touch Panel X Data Bit[9:2](Segment)	REG[C1h]
TPYR	Bit [7:0]	Touch Panel Y Data Bit[9:2](Common)	REG[C2h]
TPZR	Bit [3:2]	Touch Panel Y Data Bit[1:0] (Common)	REG[C3h]
	Bit [1:0]	Touch Panel X Data Bit[1:0] (Segment)	

Please refer to the following flow chart and the setting examples for applying Interrupt Mode:

(1) Flowchart:

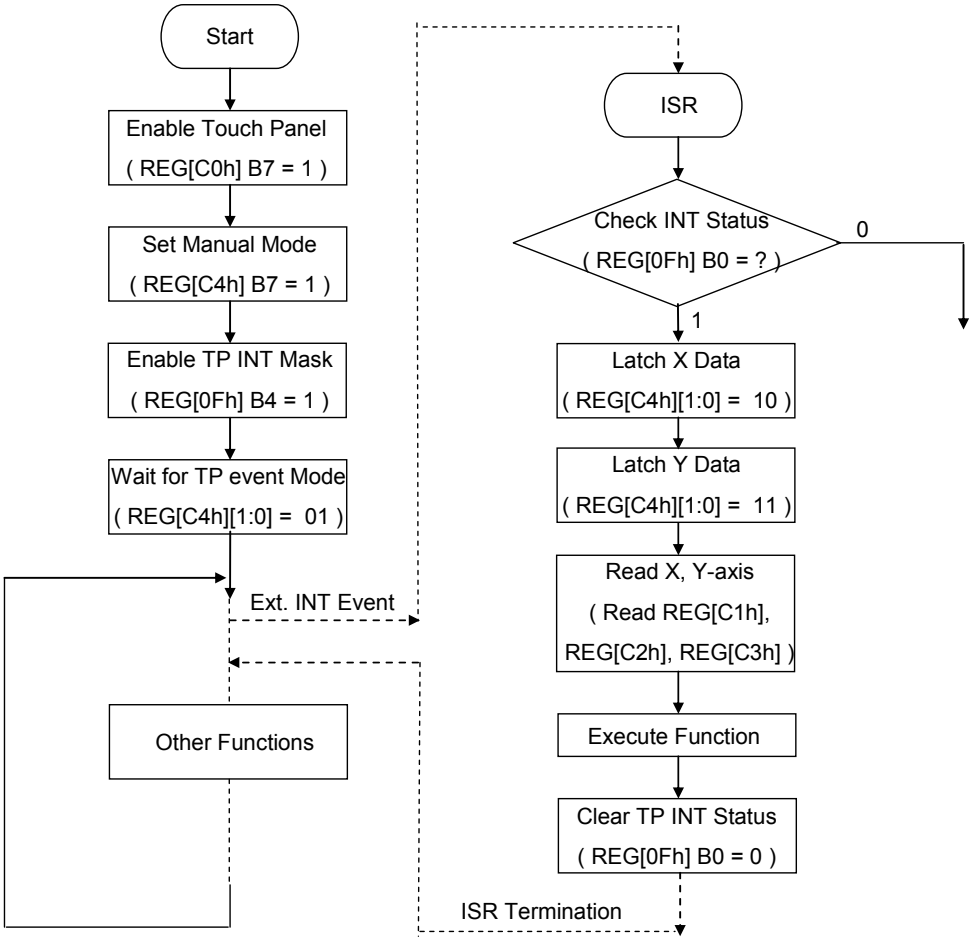


Figure 6-20 : Manual Mode Flowchart for Touch Screen

(2) Program Example:

```
Unsigned char  X1,X2,Y1,Y2;
Touch_Panel_Enable ( );                    // Set TPCR1 Bit-7 to 1
TP_Manual_Enable ( );                      // Set TPCR2 Bit 7 to 1
TP_INT_Mask_Enable ( );                    // Set  INTR Bit-4 to 1
Switch_Wait_TP_Event( );                   // Set TPCR2[1:0] to 01b
                :
                :
        Execute other function             // Jump to ISR when interrupt
                :
                :
Int  EXT_INT_Service_Routine               // ISR entry
{
      LCD_CmdWrite ( INTR );               // Check INT status
      INT_Sta  =  LCD_DataRead ( );
      If ( INT_Sta  &  0x01)               // Check If TP interrupt
      {
            Switch_Latch_X_data( );        // Set TPCR2[1:0] to 10b
            Delay_Time( );                 // Delay enough time for X data stable
            Switch_Latch_Y_data( );        // Set TPCR2[1:0] to 11b
            Delay_Time( );                 // Delay enough time for Y data stable
            LCD_CmdWrite(TPXR);
            X1 = LCD_DataRead( );          // MSB of X
             LCD_CmdWrite(TPYR);
            Y1 = LCD_DataRead( );          // MSB of Y
            LCD_CmdWrite(TPZR);
            X2 = LCD_DataRead( )  &  0x03; // LSB two Bits of X
            LCD_CmdWrite(TPZR);
            Y2 = LCD_DataRead( )  &  0x0C; // LSB two Bits of Y
                        :
                        :
               Execute corresponding function
                        :
                        :
            LCD_CmdWrite ( INTR );         // Clear Touch Panel status
            temp  =  LCD_DataRead ( ) &  0xfe;
            LCD_CmdWrite ( INTR );
            LCD_DataWrite ( temp );
       }
 Else if (INT_Sta  &  0x02)                // Check if Key-Scan interrupt
      {
            :
            :
      }
 Else if (INT_Sta  &  0x04)                // Check if Wakeup interrupt
     {
            :
            :
      }
}
```

6-4-2-2 Polling Mode

Under the ”Polling Mode”, users need to decide and set the de-bounce time after the touch event, as well as the sampling time after latch by considering the real situation, thus more flexibilities for users apply this mode.

The development procedures are explained as follows:

1. Enable Touch Panel function
2. Change mode to “Manual mode”
3. Set the switch to「Wait for Touch event」, i.e., set TPCR2[1:0] to 01b.
4. Read Touch Panel Event status from status register, check if the “Touch event” happens.
5. When touch event happens, confirm the stability of it and set the switch to 「Latch X data」, i.e., TPCR2[1:0] set to 10b, wait for enough time to make the latch data stable and latched to TPXR and TPZR
6. Set the switch to「Latch Y data」, i.e., TPCR2[1:0] set to 11b, wait for enough time to make the latch data stable and latched to TPYR and TPZR
7. Read X, Y data from TPXR, TPYR and TPZR, and clear the interrupt status

The settings for manual interrupt mode are described in the following table:

Table 6-13

Reg.	Bit_Num	Description	Reference
TPCR1	Bit 7	Enable Touch Panel function	REG[C0h]
TPCR2	Bit 7	Select operation mode to Auto-mode or Manual-mode.	REG[C4h]
	Bit [1:0]	The switch of ADC controller for manual mode	
INTR	Bit 3	Touch panel event(Only activate in TP Manual mode)	REG[0Fh]
	Bit 0	Touch Panel Detect Status bit	
TPXR	Bit [7:0]	Touch Panel X Data Bit[9:2](Segment)	REG[C1h]
TPYR	Bit [7:0]	Touch Panel Y Data Bit[9:2] (Common)	REG[C2h]
TPZR	Bit [3:2]	Touch Panel Y Data Bit[1:0] (Common)	REG[C3h]
	Bit [1:0]	Touch Panel X Data Bit[1:0] (Segment)	

Programmer can check the status of Touch Panel Event from the Bit-3 or Bit-0 of INTR, the difference between those two bits is :

1. The Bit-3 of INTR reflects the current Touch status. When Touch event occurs, the Bit-3 is 1. When there is no Touch event, Bit-3 will be 0 and read only.
2. The Bit-0 of INTR records the Touch status. When a Touch event occurs, the Bit-3 will be 1 and however it won’t be clear automatically, it has to clear by programmer.

It needs to be noted is that the REG[0Fh] Bit-3 is the direct output from ADC circuit, when touch panel is touched, the bit will respond with the event. When touch event is not stable, it needs to be de-bounced to check if it’s legal. The bit is only active at “Manual mode”. When setting RA8806 to “Auto-mode, the touch event will be automatically checked if it’s legal or not. Only the legal touch event will cause the interrupt.

Please refer to the flowchart and setting examples for applying above methods:

(1) Flowchart:

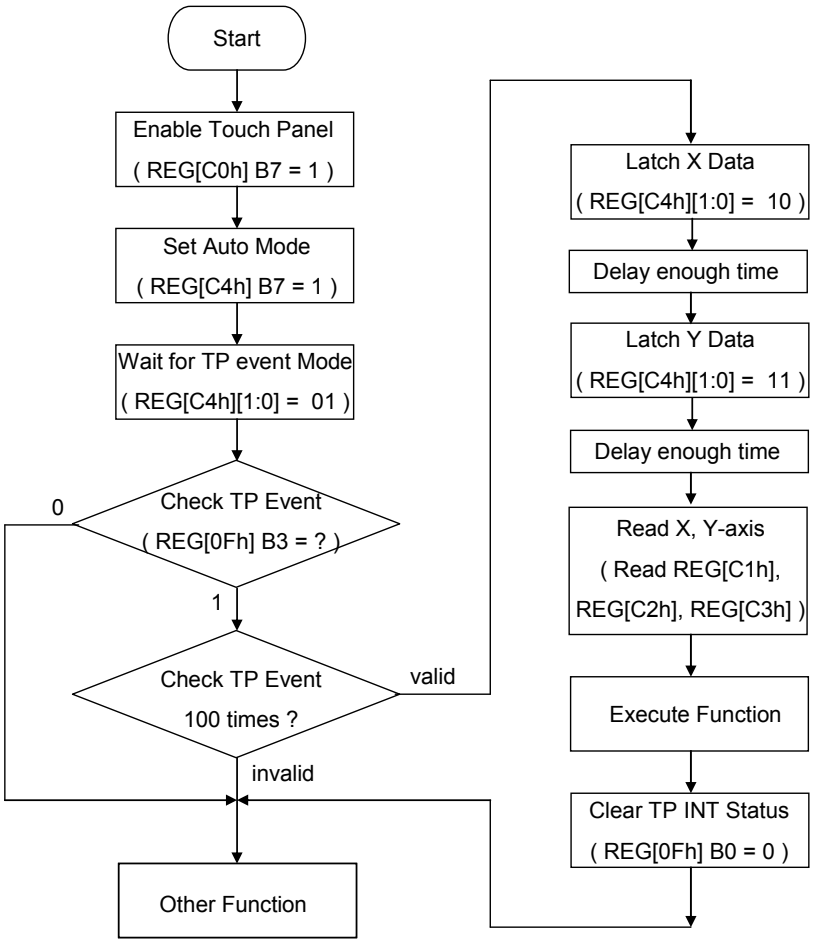


Figure 6-21 : Polling Mode Flowchart for Touch Screen

(2) Program Example:

```
Touch_Panel_Enable ( );                     // Set REG[C0h]. Bit-7 = 1
TP_Manual_Enable ( );                       // Set REG[C4h]. Bit-7 = 1
Switch_Wait_TP_Event( );                    // Set REG[C4h][1:0] = 01

Touch_Sta_Valid = 0;                        // Initial Touch state
LCD_CmdWrite ( INTR );
INT_Sta  =  LCD_DataRead ( );
If ( INT_Sta &  0x08 )                      // Check INTR.Bit-3
{
    for ( count = 0 ; count < 100 ; count++ )   // Check 100 times
    {
        LCD_CmdWrite ( INTR );
        INT_Sta = LCD_DataRead( );
        if (INT_Sta == 0)                   // When  no touch
        {
            Touch_Sta_Valid = 0;            // Touch is invalid
            break;
        }
        if ( count == 99 )                  // When count 100 times, touch is valid
            Touch_Sta_Valid = 1;
    }
    if (Touch_Sta_Valid )
    {
            Switch_Latch_X_data( );         // Set REG[C4h][1:0] = 10
            Delay_Time( );                  // Delay enough time
            Switch_Latch_Y_data( );         // Set REG[C4h][1:0] = 11
            Delay_Time( );                  // Delay enough time
            LCD_CmdWrite(TPXR);
            X1 = LCD_DataRead( );           // Read high byte of X-axis
             LCD_CmdWrite(TPYR);
            Y1 = LCD_DataRead( );           // Read high byte of Y-axis
            LCD_CmdWrite(TPZR);
            X2 = LCD_DataRead( )  &  0x03;  // Read Least two Bits of X-axis
            LCD_CmdWrite(TPZR);
            Y2 = LCD_DataRead( )  &  0x0C;  // Read Least two Bits of Y-axis
                    :
                Execute corresponding function
                    :
            LCD_CmdWrite ( INTR );          // Clear REG[0Fh]. Bit-0
            temp  =  LCD_DataRead ( ) &  0xfe;
            LCD_CmdWrite ( INTR );
            LCD_DataWrite ( temp );
    }
}
            :
        Execute other function
            :
            :
```

6-4-3 Touch Panel Sampling Time Reference Table

When using the auto mode of touch panel function, It is suggested to select suitable TP Sampling wait time while the ADC Clock Convert speed change, in order to avoid the TP Sample wait time error. Please refer to the following table for the ADC Sampling wait time REG[C0h][6:4]。

Figure 6-21-A : Touch Panel Sampling Time Reference Table

ADC Sampling Wait Time - REG[C0h] Bit[6:4]					
SYSTEM_CLK / REG[C0h] [2:0]	4M	6M	8M	10M	12M
000	000	000	000	000	000
001	000	000	000	000	000
010	000	000	000	000	000
011	001	001	000	000	000
100	010	010	001	001	001
101	011	011	010	010	010
110	100	100	011	011	011
111	101	101	100	100	100

6-5 Key-Scan

RA8806 features with Key-Scan circuit, and could be used as Keyboard function. It will help to integrate the system circuit that includes keyboard application. The below Figure 6-22 shows the basic application circuit of 8x8 Key-Pad. RA8806 already built-in pull-up resistors in the pins “KIN[7:0]”.

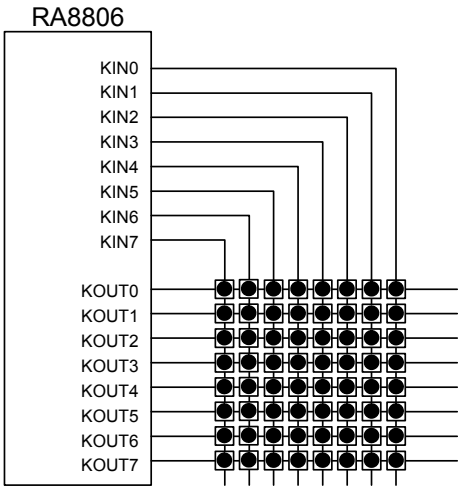


Figure 6-22 : 8x8 Key-Pad Application

The related Registers of Key-san function are KSCR, KSDR, and KSER. The RA8806 Key-Scan controller features are given below:

1. Support with 4x8 or 8x8 Key-Scan Matrix
2. Programmable setting of sampling times and scan frequency of Key-Scan
3. Adjustable long key-press Timing
4. Multi-Key is available (maximum three keys at the same time)
5. Allows the function of “ Key press to wake-up the system”

Table 6-14 is the key code of mapping to key-pad matrix for normal press. The key code will be stored in REG[A2h] when key was pressed. If it was a long time press, then the key code is show as Table 6-15.

Table 6-14 : Key Number Mapping Table (Normal Key)

		ROW #							
		0	1	2	3	4	5	6	7
COL #	0	00h	01h	02h	03h	04h	05h	06h	07h
	1	10h	11h	12h	13h	14h	15h	16h	17h
	2	20h	21h	22h	23h	24h	25h	26h	27h
	3	30h	31h	32h	33h	34h	35h	36h	37h
	4	40h	41h	42h	43h	44h	45h	46h	47h
	5	50h	51h	52h	53h	54h	55h	56h	57h
	6	60h	61h	62h	63h	64h	65h	66h	67h
	7	70h	71h	72h	73h	74h	75h	76h	77h

Table 6-15 : Key Number Mapping Table (Long Key)

COL # \ ROW #	0	1	2	3	4	5	6	7
0	80h	81h	82h	83h	84h	85h	86h	87h
1	90h	91h	92h	93h	94h	95h	96h	97h
2	A0h	A1h	A2h	A3h	A4h	A5h	A6h	A7h
3	B0h	B1h	B2h	B3h	B4h	B5h	B6h	B7h
4	C0h	C1h	C2h	C3h	C4h	C5h	C6h	C7h
5	D0h	D1h	D2h	D3h	D4h	D5h	D6h	D7h
6	E0h	E1h	E2h	E3h	E4h	E5h	E6h	E7h
7	F0h	F1h	F2h	F3h	F4h	F5h	F6h	F7h

When the Multi-Key function is applied, the pressed keys will be saved in the system each with KSDR0, KSDR1 and KSDR2. Note that the priority of keys saving is determined on the value size of Key-code, not the orders of keys pressing, please refer to the following example:

Press the Key-code in turn of 0x44, 0x00 and 0x22, press Multi-Key at the same time, the Key-code will be saved in KSDR:

KSDR0 = 0x00
KSDR1 = 0x22
KSDR2 = 0x44

The definition of Key-code is described in the registers KSDR0 ~ 2 [A2h ~ A4h]. The basic features of above Key-Scan settings are introduced as follows:

Table 6-16

Reg.	Bit_Num	Description	Reference
KSCR1	Bit 7	Key-Scan enable bit	REG[A0h]
	Bit 6	Key-Scan size selection	
	Bit [5:4]	Key-Scan sampling times setting	
	Bit 3	Long key function enable	
	Bit [2:0]	Key-Scan scan frequency setting	
KSCR2	Bit [3:2]	long key timing adjustment	REG[A1h]
	Bit [1:0]	Pressed key number	
KSDR0 KSDR1 KSDR2	Bit [7:0]	Key code for pressed key	REG[A2h ~ A4h]
INTR	Bit 5	Key-Scan interrupt enable	REG[0Fh]
	Bit 1	Key-Scan Interrupt Status bit	

Besides, in the Sleep Mode, RA8806 allows the “Wake-up on Keystroke” function, if any key is pressed, the System Clock starts to oscillate, and de-bounce to check if the wake-up event is legal. If legal then after two Frame time, the Display on the panel will be lighted, then system will recover from the Sleep Mode. If not legal, the system clock is off again and keeps on the Sleep mode. Please refer to the description of “Power Mode” for the detail introduction of “Sleep Mode”. The settings for Wake-up function are explained as follows:

Table 6-17

Reg.	Bit_Num	Description	Reference
KSCR2	Bit 7	Enable Key-Scan wake-up function	REG[A1h]
INTR	Bit 6	Wake-up interrupt enable bit	REG[0Fh]
	Bit 2	Wake-up Interrupt Status bit	

Enabling the Key-Scan functions, programmer cans uses following methods to check keystroke.

1) Software check method: to know the key be pressed from keeping check the status of Key-Scan (Bit-1 of INTR)

2) Hardware check method: to know the key be pressed from external interrupt signal

Please be aware that the status of Key-Scan (Bit-1 of INTR) has been set to “1” no matter which method is used, programmer have to clear the status to 0 after reading the correct Key Code, otherwise the interrupt will be kept that no more interrupt is visible again.

The flowchart and sample settings for above applications are shown as follows:

1. Software Method:

(1) Flowchart:

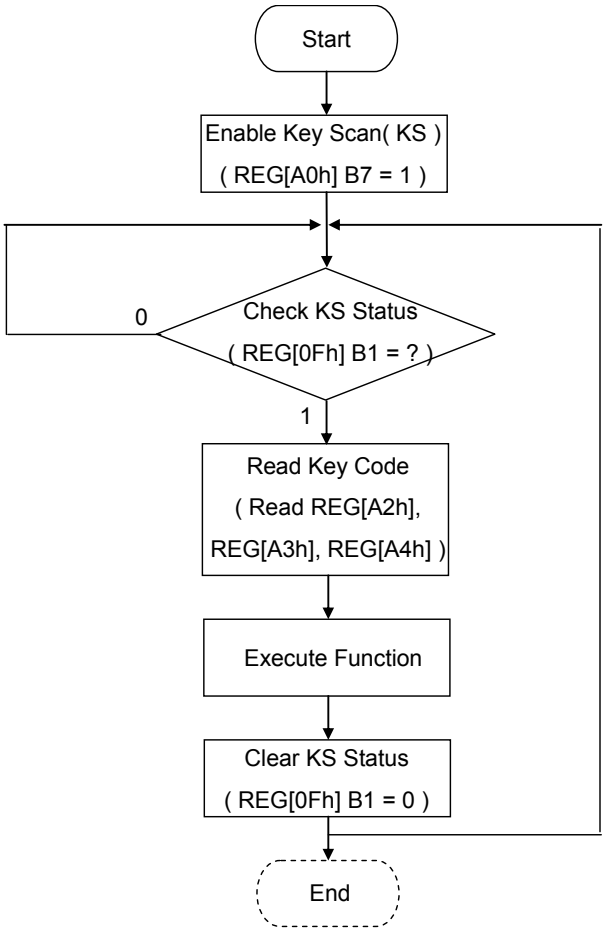


Figure 6-23 : Key-Scan Flowchart (1)

(2) Program Example:

```
Key-Scan_Enable ( );                    // KSCR Bit-7 is set to 1
while (1)
{
      LCD_CmdWrite ( INTR );            // Check Key-Scan status
      KS_Sta  =  LCD_DataRead ( );
      KS_Sta  =  KS_Sta  &  0x02;
      If ( KS_Sta )
      {
            LCD_CmdWrite ( KSDR0 );     // Read first Key Code
            KeyCode1 = LCD_DataRead ( );

            Switch ( KeyCode1 )         // Execute corresponding action
            {
                  case 0 :
                              :
                              :
                        break;
                   case 1 :
                              :
                              :
            }

            LCD_CmdWrite ( INTR );      // Clear Key-Scan status
            temp  =  LCD_DataRead ( );
            temp   =  temp  &  0xfd;
            LCD_CmdWrite ( INTR );
            LCD_DataWrite ( temp );
      }
 }
```

2. Hardware Method:

(1) Flowchart:

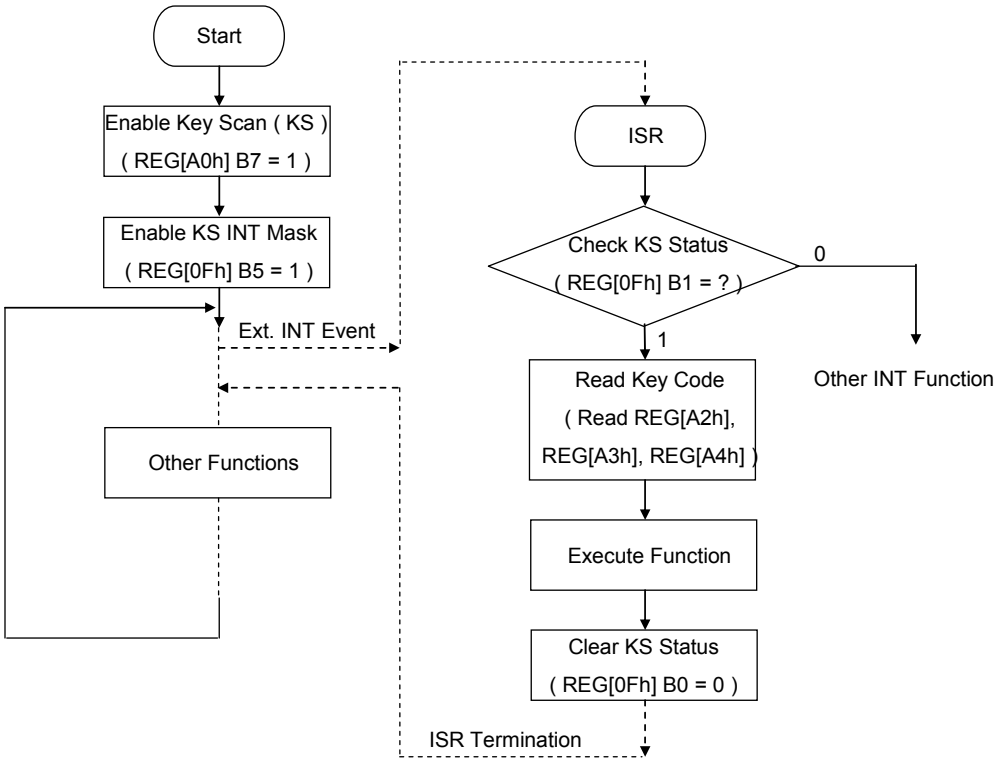


Figure 6-24 : Key-Scan Flowchart (2)

(2) Program Example:

```
Key-Scan_Enable ( );                        // Set Reg. KSCR1 Bit-7=1
Key-Scan_INT_Mask_Enable ( );               // Set Reg. INTR Bit-5=1
                :
                :
          Execute other functions           // Jump to ISR when external interrupt
                :
                :

Int  EXT_INT_Service_Routine                // ISR entry
{
      LCD_CmdWrite ( INTR );                // Check INT status
      INT_Sta  =  LCD_DataRead ( );


      If ( INT_Sta  &  0x02 )               // Check if Key-Scan interrupt
      {
           LCD_CmdWrite ( KSDR0 );          // Read Key Code
           KeyCode1 = LCD_DataRead ( );

           Switch ( KeyCode1 )              // Execute keystroke function
           {
                  case 0 :
                                :
                                :
                           break;
                   case 1 :
                                :
                                :
            }
            LCD_CmdWrite ( INTR );          // Clear Key-Scan status
            temp  =  LCD_DataRead ( );
            temp   =  temp  &  0xfd;
            LCD_CmdWrite ( INTR );
            LCD_DataWrite ( temp );
       }
       else if (INT_Sta  &  0x01)           // Check if Touch Panel interrupt
            {
                 :
                 :
             }
            else if (INT_Sta  &  0x04)      // Check if Wakeup interrupt
                  {
                        :
                        :
                   }
}
```

6-6 Clock and Reset

RA8806 supports internal or external clock source without extra select pin.

6-6-1 OSC Circuit

RA8806 contains a built-in OSC circuits. It generates corresponding clock by connecting an external 4M ~ 12MHz crystal between XG and XD pins. Figure 6-25 is the clock circuit for X'tal oscillator and external clock.

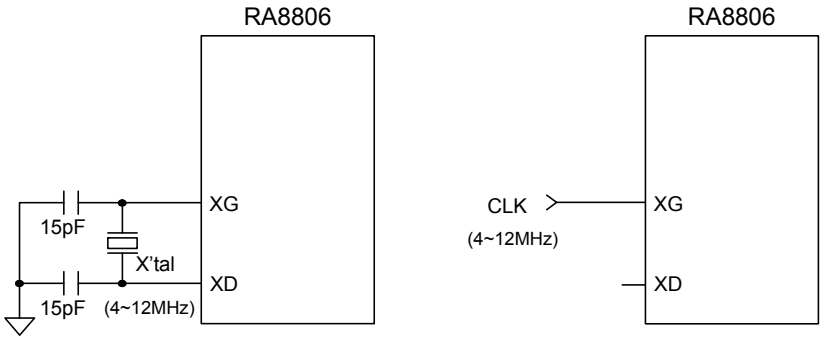


Figure 6-25: Clock Circuit

6-6-2 External Clock

RA8806 can also accept external clock for system clock source. The external clock source can directly connect to XG pin, and XD pin must be kept floating.

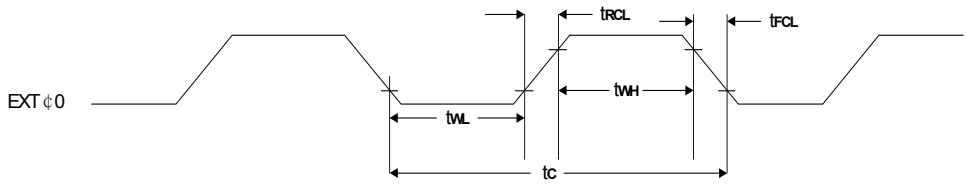


Figure 6-26 : External Clock

Table 6-18 : Clock Timing **Ta = –20 to 75℃**

Signal	Symbol	Parameter	V_{DD} = 5V		V_{DD} = 3.3V		Unit	Condition
			Min.	Max.	Min.	Max.		
EXT Φ0	t_{RCL}	External clock rise time	—	10	—	10	ns	
	t_{FCL}	External clock fall time	—	10	—	10	ns	
	t_{WH}	External clock HIGH-level pulse width	Note 1.	Note 2.	Note 1.	Note 2.	ns	
	t_{WL}	External clock LOW-level pulse width	Note 1.	Note 2.	Note 1.	Note 2.	ns	
	t_C	External clock period	66.6	—	83.3	—	ns	

Notes:

1. $(t_C - t_{RCL} - t_{FCL}) \times \frac{475}{1000} < t_{WH}, t_{WL}$
2. $(t_C - t_{RCL} - t_{FCL}) \times \frac{525}{1000} > t_{WH}, t_{WL}$

6-6-3 Reset

The RA8806 requires a reset pulse at least 1024*tc long after power-on in order to re-initialize its internal state. If the oscillator frequency is 6Mhz, then the Reset pulse is at least 170.7μs. For maximum reliability, it is not recommended to apply a DC voltage to the LCD panel while the RA8806 is reset. Turn off the LCD power supplies for at least one frame period after the start of the reset pulse.

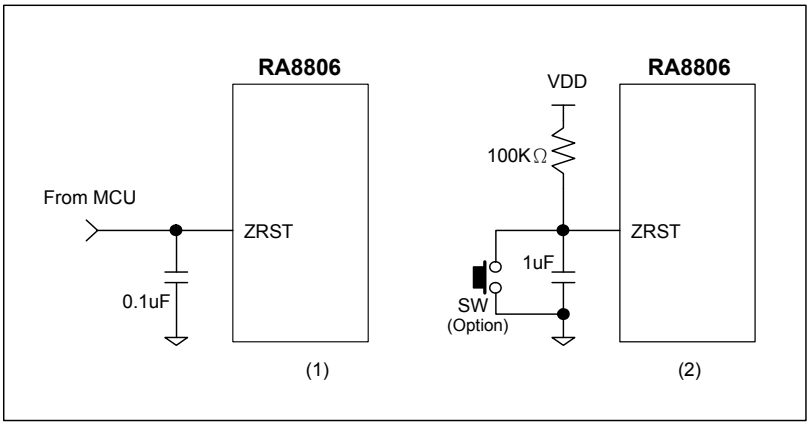
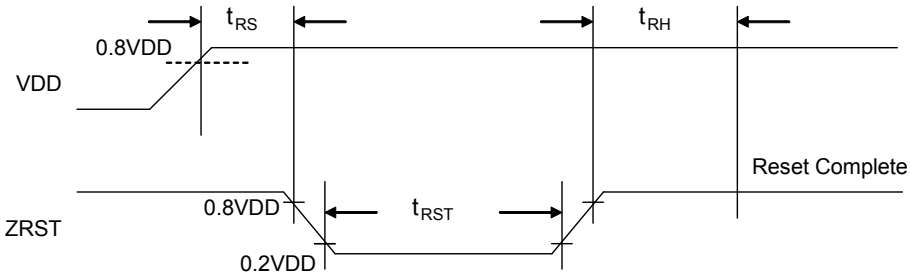


Figure 6-27: Examples of ZRST Pin

Figure 6-27 is an example for ZRST application circuit. It could be controlled by MPU such as (1) of Figure 6-27. Or, generated by a RC circuit such as (2) of Figure 6-27.

The RA8806 cannot receive commands while it is reset. Commands to initialize the internal registers should be issued soon after a reset. During reset, the LCD drive signals XD, LP and FR are halted. A delay of 1ms (minimum) is required following the rising edges of both ZRST and VDD to allow for system stabilization. Please refer to Figure 6-28 for more detail description.



		Max.	Typ.	Min.	Unit
t_{RS}	Reset setup time	--	--	1	ms
t_{RH}	Reset hold time	--	--	1	ms
t_{RST}	Reset active time	--	--	1024	t_c (*)

*t_C is the period of system clock,
for example: 10MHz, t_C = 100ns

Figure 6-28: Reset Timing

6-7 Power

6-7-1 Power Architecture

The power architecture of RA8806 is shown as Figure 6-29. VDDP, GNDP are I/O powers and AVDD, AGND are analog powers for internal ADC. There is an embedded 5V-to-3V DC/DC Converter which is without any external controlled pin. The application circuits of 5V and 3V system will be described below.

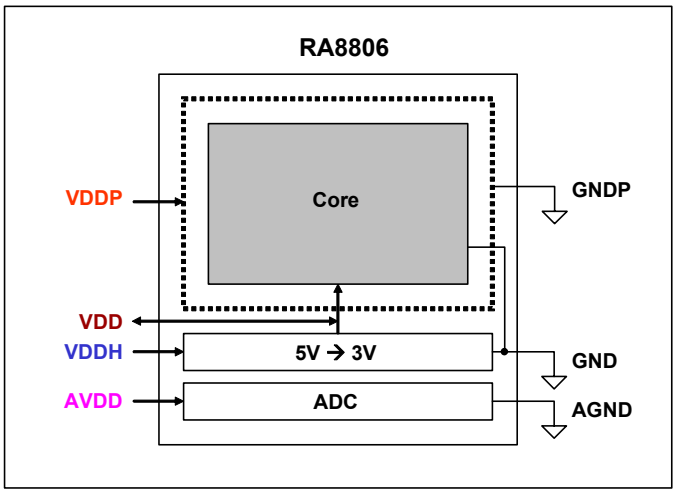


Figure 6-29 : Power Architecture

6-7-2 3V Application Circuit

When RA8806 operates in 3V system, the power consumption will be smaller. The Figure 6-30 is the example for 3V system. Users need to connect 3V power to VDDP, VDD and AVDD respectively. The DC/DC Converter is not used, so VDDH needs to be kept floating.

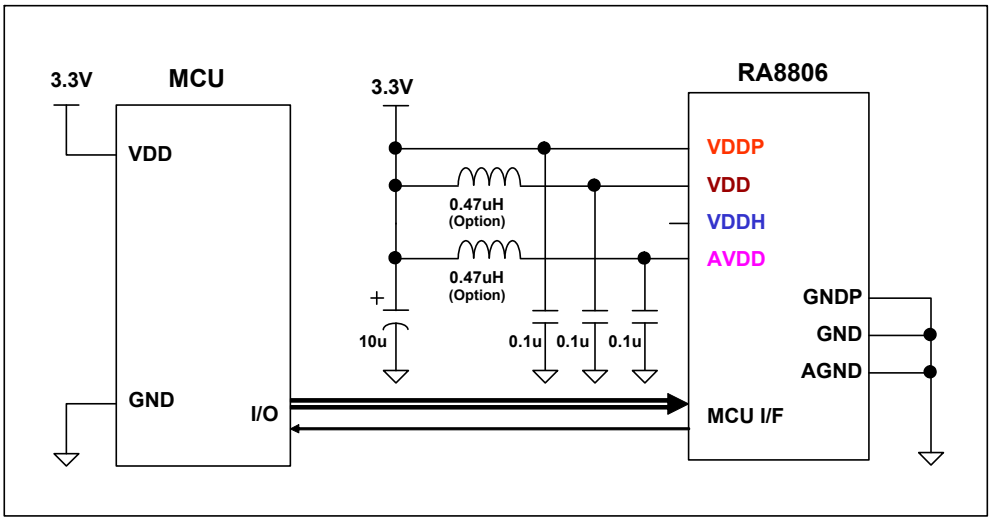


Figure 6-30: The Power Connection of 3V

6-7-3 5V Application Circuit

When RA8806 operates in 5V system, 5V power inputs from VDDH and 3V power will output at VDD that is the power to supply internal core. Users need to add 1μF and 0.1μF capacitors in parallel at VDD to increase power stability.

Please note that DC-to-DC converter will increase some power consumption. So, if the RA8806 enter Sleep mode then it will keep around 20μA static current.

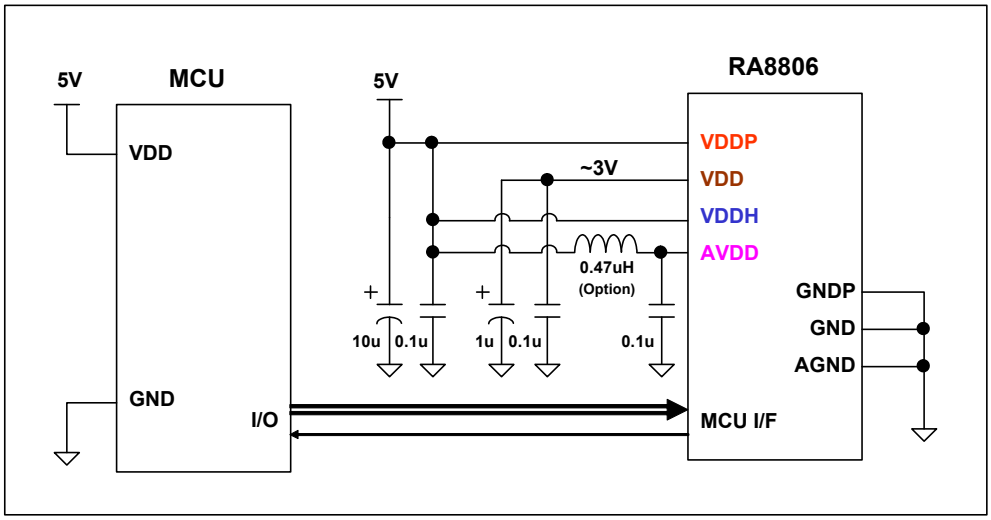


Figure 6-31 : The Power Connection of 5V

6-7-4 Sleep Mode

The RA8806 provide two operation modes: Normal mode and Sleep mode. Please refer to Chapter 5 “Register Description” for Register WLCR explanation. RA8806 will shut down the system clock, to achieve the minimum power consumption. When RA8806 is under the Sleep mode, only the status register can be read, other registers are not allowed to read. Other command besides REG[00h] Bit-7(Sleep mode) can’t be written. When RA8806 is under Sleep Mode, it can accept the following three methods to wake-up.

1. Write REG[00h] Bit-7 = 0, then it will return to Normal Mode.
2. Touch event is detected.
3. Key-Scan is detected.

When RA8806 quit from Sleep mode, to avoid the incorrect activity, RA8806 must de-bounce the wake-up event for enough time(about 1,000 system clock periods) to make sure it, then quit from Sleep mode.

Besides, when MPU program RA8806 to enter Sleep mode, RA8806 can’t receive command from MPU anymore till quit the Sleep mode. The command will be lost. User program must avoid the possibility of the condition. For example, if RA8806 be programmed to Sleep mode, at the same time, the MPU receives an external interrupt, and cause the MPU enter the Interrupt Service Routine(ISR). That will cause RA8806 can’t correctly receive and recognize the command.

From the view of hardware design, the condition can’t be prevented. Because the external interrupt is an individual mechanism of MPU. We can’t predicate the happen of it from the role of RA8806. It’s a possible risk even the possibility is rare.

So it’s suggested that MPU must be mask the interrupt for RA8806 before commanding the RA8806 to enter the Sleep mode. The interrupt mask is off after RA8806 quit from Sleep mode. Besides, because the status register can be read normally at Sleep mode. Also user can check the status in the ISR for judgment. If RA8806 is checked under Sleep mode, MPU can quit ISR and execute nothing to prevent incorrect activity.

Table 6-19

Reg.	Bit_Num	Description	Reference
WLCR	Bit 7	Normal mode and Sleep mode selection.	REG[00h]

6-8 Interrupt and Busy

The RA8806 provides an interrupt output (INT) for MPU to indicate the status of RA8806. And a busy output to indicate the RA8806 is in busy state. Both signals could be set to active high or active low by register.

6-8-1 Interrupt

RA8806 provides an Interrupt signal (INT) for following event:

- Touch Panel touched event is happen.
- Key-Scan enable and key is pressed
- Wake-up event occurs when Sleep mode.

These interrupt events can be masked or active respectively. It is controlled by the REG[0Fh], the mask and status registers.

Besides, RA8806 also provide a software access interrupt function. When user system don't support hardware interrupt signal, they can use polling method to achieve the software interrupt.

To access hardware interrupt, user needs to set corresponding Interrupt Mask bit to 1(Refer to register INTR). The steps are:

- RA8806 generate the interrupt signal to MPU.
- After finishing the current instruction, MPU hardware will jump the PC(Program counter) to Interrupt Service Routine(ISR).
- In the same time that RA8806 generate the interrupt, the interrupt status is set to "1"(REG[0Fh] Bit[2:0]). (For example, if key-scan interrupt activates, the Key-Scan status will be set to 1).

To apply the software interrupt method, users don't need extra setting. Just reading the status bit of register INTR to check the interrupt event. BTW, interrupt mask can only disable the hardware interrupt, but can't disable the status of INTR.

Example-1:

The Figure 6-32 is an example of INT timing for Wake-up event. RA8806 provides three wake-up event. Please refer to Section 6-7-4 "Sleep Mode" for the detail.

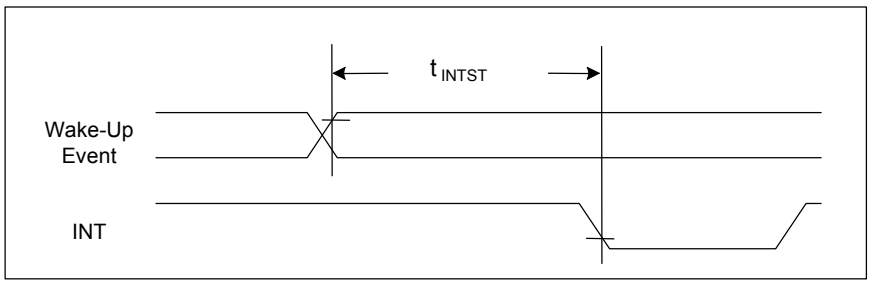


Figure 6-32: Interrupt Timing(1)

t_{INTST} = Clock Stable Time + 1024*t_C

If user use a 6MHz X'tal for system clock, the Clock Stable Time is about 3~3.5ms and t_C is 167ns.

Example-2:

The Figure 6-33 is an example of INT timing for Key-Scan event.

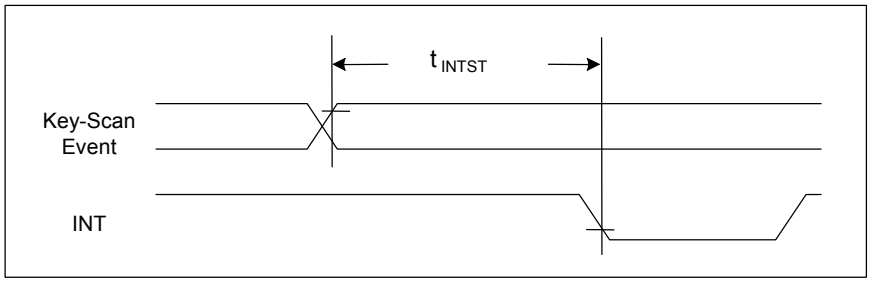


Figure 6-33 : Interrupt Timing(2)

t_{INTST} = De-bounce Time + t_{CKEY}

The "De-bounce Time" is set by REG[A0h] Bit[5:4], and the t_{CKEY} is the Key-Scan Cycle that depends on the setting of REG[A0h] Bit[2:0].

Additionally, because the status bit will not be cleared to 0 automatically. So user must manually clear it after processing the interrupt routine.

6-8-2 Busy

The RA8806 also provides a "BUSY" signal. When Busy Flag is "1", which means RA8806 is in busy state that RA8806 couldn't access data from DDRAM. Two busy conditions are provided. One is Scan Busy and another is Memory Write Busy. The detail explains is as follows.

Scan Busy:

When LCD panel displays, RA8806 scan circuit will access the DDRAM. If another DDRAM access cycle happens. It will cause the data lose in one of them. So when scan circuit is active, it causes a busy condition, called Scan Busy. Figure 6-34 shows the data flow diagram of scan circuit and MPU memory access cycle. Figure 6-35 is same as Figure 6-15 to show RA8806 scan waveform. It describes the condition that RA8806 scan the display for each COM line. The COM scan time is combined with an Idle time and a Scan time. The Idle time period can be set by register ITCR. The Scan period is the time of Scan Busy. Also accessing data at Scan Busy time will cause the scan data lose. But it will not cause a fatal error. The scan data lose will cause the display defect. But if it's not too frequent, the defect will not infect the display too much.

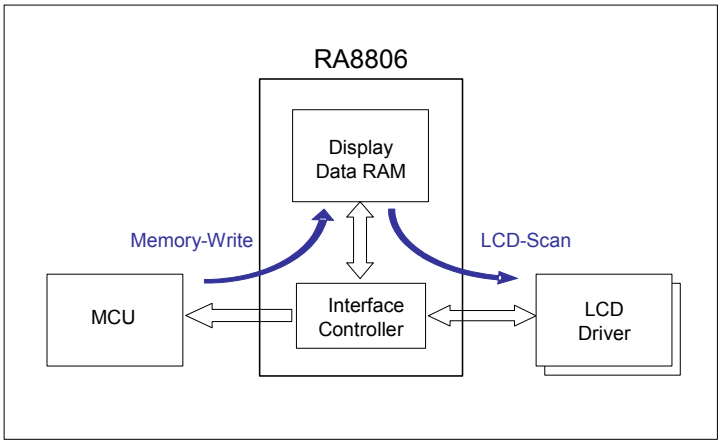


Figure 6-34 : Data Flow of DDRAM

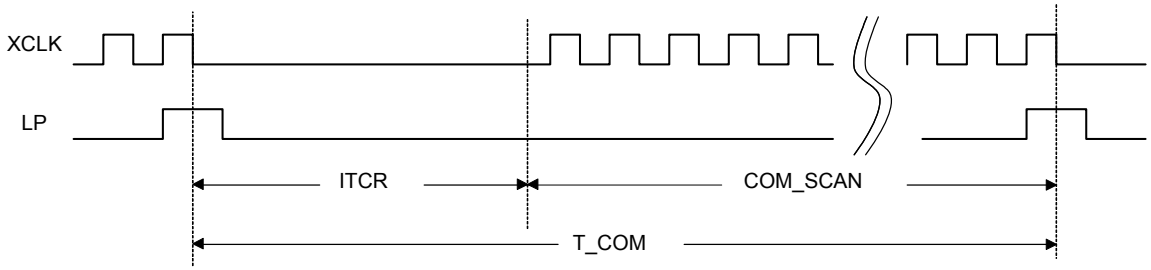


Figure 6-35 : Scan for Each COM Line

Memory Write Busy:

Following two conditions will cause the Memory write busy:

1. When MPU write data in text-mode. Depending on different size of font, it needs an enough time to write the font to DDRAM. At the period, RA8806 can't access the DDRAM again, it's a memory write busy condition.
2. When MPU program RA8806 as a memory clear function(FNCR Bit-3 = 1). The period of clear DDRAM also cause a memory write busy.

It will cause the DDRAM lost when accessing DDRAM in the period of Memory Write Busy. So user must check the busy status after the upper two conditions is done.

Besides, RA8806 provide polarity setting for "BUSY" and interrupt "INT" pin. (Please refer to register MISC)

Normally, this "BUSY" pin is connected to MPU I/O input, and then MPU have to monitor this pin before accessing RA8806. The following is the timing of BUSY pin. The BUSY can be active high or low that depend on the setting of REG[01h] Bit-5.

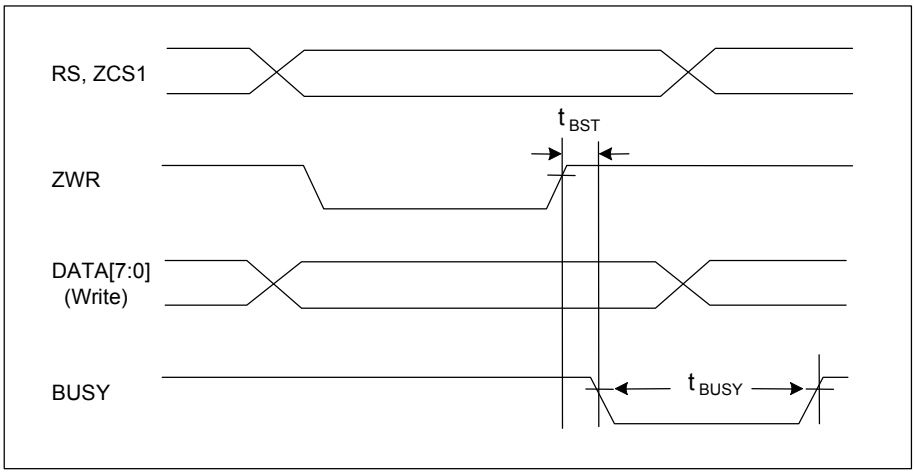


Figure 6-36 : BUSY Timing Chart

Table 6-20 : Busy Timing

Signal	Symbol	Parameter		Rating		Unit	Condition
				Min	Max		
BUSY	t_{BST}	Busy Setup Time	Half Size Font	150	--	ns	System Clock: 8MHz VDD: 5V tc = 125ns
			Full Size Font	250			
	t_{BUSY}	Busy Active Time	Half Size Font	--	50*tc	ns	
			Full Size Font	--	100*tc	ns	

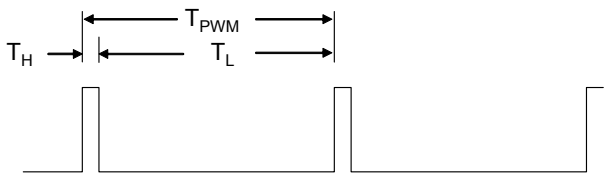
6-9 PWM

RA8806 provide a set of programmable PWM (Pulse Width Modulation) for LCD contrast adjustment. The PWM frequency and duty can be set by register. And the driving capability of PWM output pin is larger than other output pin, about 4 multiples of normal output. Besides, if the PWM function is disable, it can use as normal IO signal. The relative function setting please refers to the table below.

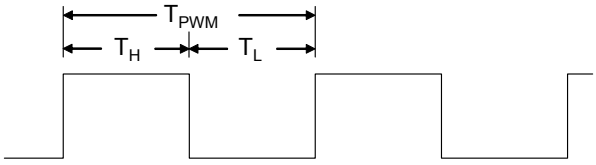
Table 6-21

Reg.	Bit_Num	Description	Reference
PCR	Bit 7	PWM function enable	REG[D0h]
	Bit [3:0]	Clock source divide ratio select	
PDCR	Bit [7:0]	PWM Duty Cycle Select	REG[D1h]

The following are two examples for the PWM output (pin "PWM_OUT"):



Example-1:
System Clock = 10Mhz,
Register PCR Bit[3:0] = 0001b → Clock Source = 10MHz/2 = 5MHz
T_{PWM} = 256*(1/5MHz) = 51.2μs
Register PDCR Bit[7:0] = 0Fh →
T_H = 16*(1/5MHz) = 3.2μs
T_L = (256-16) * (1/ 5MHz) = 48μs



Example-2:
System Clock = 10Mhz,
Register PCR Bit[3:0] = 0010b → Clock Source = 10MHz/4 = 2.5MHz
T_{PWM} = 256*(1/2.5MHz) = 102.4 μs
Register PDCR Bit[7:0] = 7Fh →
T_H = 128*(1/2.5MHz) = 51.2μs
T_L = (256-128) * (1/ 2.5MHz) = 51.2μs

Figure 6-37 : PWM_OUT Pulse

Figure 6-38 shows the reference circuit of PWM contrast application for positive VLCD voltage. Figure 6-39 shows the reference circuit for negative VLCD voltage.

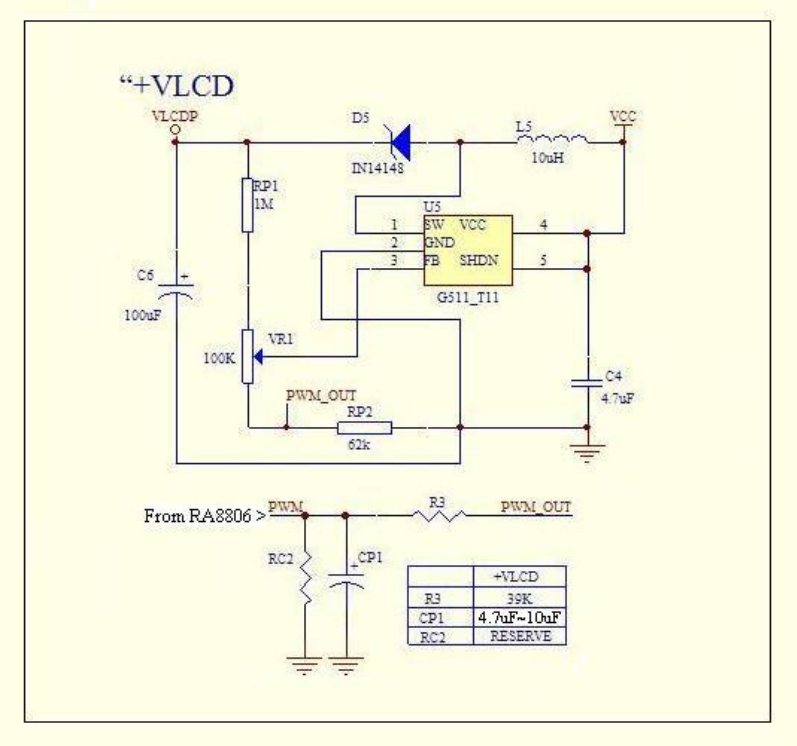


Figure 6-38 : PWM Reference Circuit for Positive VLCD

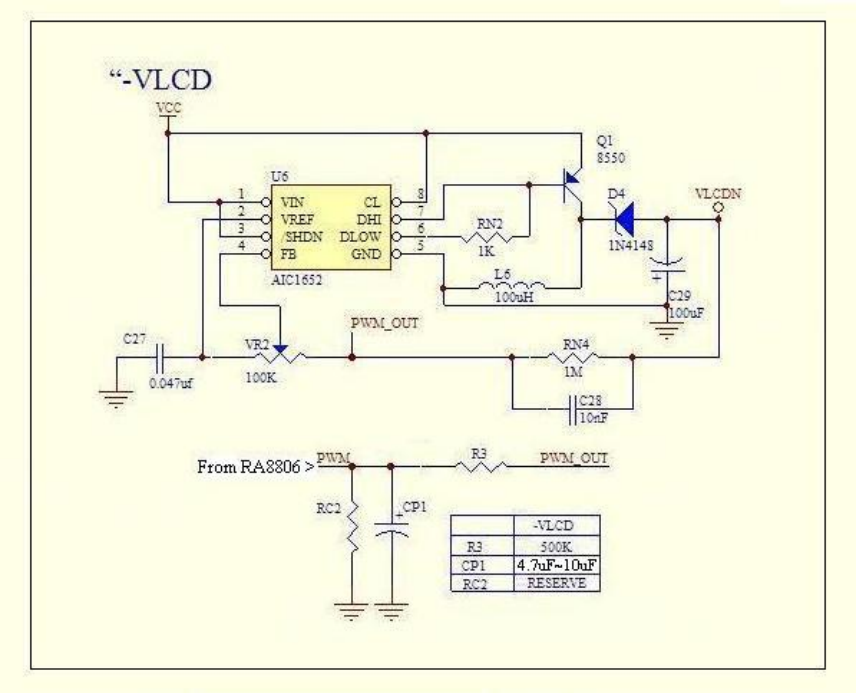
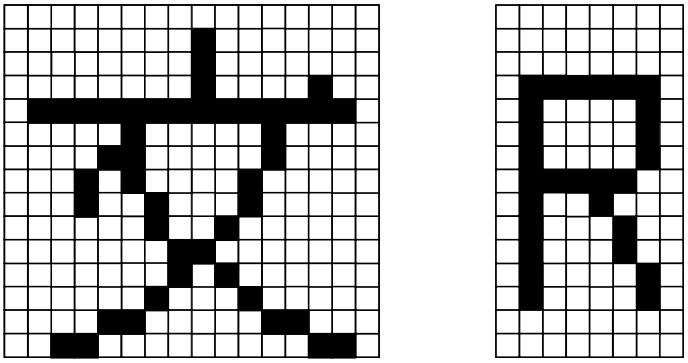


Figure 6-39 : PWM Reference Circuit for Negative VLCD

6-10 Display Function

6-10-1 Character/Graphic Mode

There are two modes for MPU to write data to RA8806, i.e., character mode and graphic mode. In graphic mode, data is written directly to DDRAM in bit-map format. In character mode, data is written in code format, the font bit-map in the CGROM will be written to DDRAM by this way. RA8806 stores two different sizes of characters in its font ROM - 1) half size font(8x16 pixels), 2) full size font(16x16). Figure 6-40 shows the examples.



16X16 font 8X16 font

Figure 6-40 : Full Size and Half Size Font

6-10-1-1 Graphic Display

The RA8806 graphics mode is use bit map to fill the data on the DDRAM. The Figure 6-41 is an example to show how to set graphics mode.

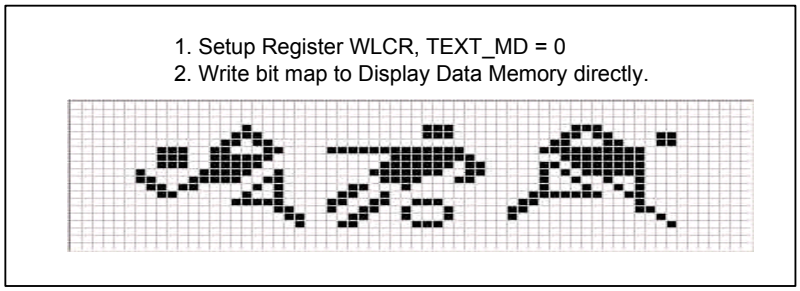


Figure 6-41 : Graphics Mode

The RA8806 support maximum resolution is 320x240 pixels, therefore it need 9.6Kbyte(320x240/8 = 9600) Display Data RAM(DDRAM) to store each pixel data. Figure 6-42 is an example to show the DDRAM data mapping to the LCD panel.

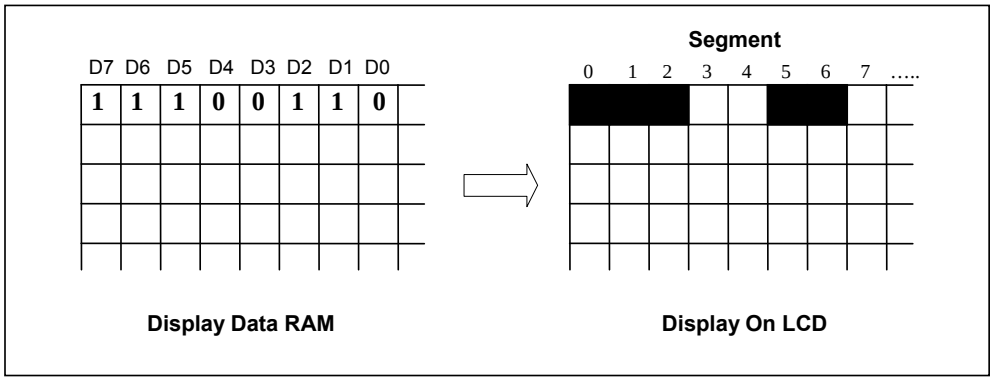


Figure 6-42 : The Mapping of Display Data to LCD Panel

The RA8806 provide an Auto-Write feature to fill a data to all of the DDRAM. At first, user writes the data to Register PNTR then initials the Auto-Write function(Register FNCR Bit-3). RA8806 will fill the data to DDRAM in very short time. Normally this feature is used to clear screen or want to fill fixed pattern or background on screen.

Table 6-22

Reg.	Bit_Num	Description	Reference
WLCR	Bit 3	Text Mode Selection	REG[00h]

6-10-1-2 Half Size Font

RA8806 built in 4 half-size font blocks, each block contains 0h ~ FFh(256) characters, and could be selected by register REG[F0h] Bit[1:0]. By setting the REG[F0h] Bit[1:0], the corresponding table is selected. About the font table, please reference Appendix C. The RA8806 also supports ISO8859 standard (ISO - International Organization for Standardization). Which can fully supports most Latin characters set. REG[F0h] Bit-7 is the option of RA8806 default coding or ISO8859 mode. In this mode, RA8806 internal ASCII table1 ~ 4 is mapping to the coding of ISO8859-1 ~ ISO8859-4.

Table 6-23

Reg.	Bit_Num	Description	Reference
FNCR	Bit 7	ISO8859 mode	REG[F0h]
	Bit 2	ASCII Mode Enable	
	Bit [1:0]	ASCII Blocks Select	

6-10-1-3 Full Size Font

There are three types of CGROM in RA8806, i.e. RA8806-S, RA8806-T and RA8806-J. RA8806-S contains the Simplified Chinese characters with the coding of GB standard. RA8806-T contains the Traditional Chinese characters with the coding of BIG5 standard. RA8806-J contains the JIS Japanese Kanji Level 1 & 2 font characters. About the detail, please refer the Appendix D, Appendix E, and Japanese Kanji font code table (RA8806_DS_V12_Font_JIS.pdf).

Table 6-24

Reg.	Bit_Num	Description	Reference
WLCR	Bit 3	Text Mode Selection	REG[00h]

The RA8806 displays the Chinese characters in text mode. Key in the Chinese code (GB or BIG5 code) directly , then it will display the Chinese characters in cursor position. If display Chinese , RA8806-S accepts the GB code, RA8806-T accepts the BIG-5 code. The Chinese character occupies two Byes, hence the MPU should write the Chinese code (High Byte & Low Byte) separate into RA8806 in case the MPU interface is 8-Bit. The English code or numeric code occupies only one Byte, hence it will write the code into the RA8806 at a time. i.e. for RA8806-T, the BIG-5 code of character ”世” is “A540”, MPU will send the “A5” and “40” in proper order to RA8806-T, then it will display the Chinese character of ”世”.

If use the Japanese Kanji font version of RA8806, that is RA8806-J, it should transfer the JIS code for a start , then the MPU send the code to RA8806-J. Firstly, add 80h to the Hi-Byte of JIS code, then to judge the Low-Byte, if it is larger or equal to 60h, then adds 40h, otherwise adds 20h. Then it will display the Japanese Kanji font if the new code sends to RA8806-J by MPU. i.e. the JIS code of character ”粟” is “3040”, to transfer to the new code is “B060”, to send the “B0” and “60” in proper order, then it will display the Japanese Kanji font of ”粟”.

The Font ROM of RA8806-J mainly refer to the coding of JIS code, if the user use Shift-JIS code (called for short S-JIS), then it should be transfered by another procedure, firstly, transfer S-JIS code to JIS code. It is in common use of S-JIS coding for Japanese computer system, because it contains the full font and half font of Latin character, Hiragana, Katakana, notation and the Japanese Kanji font. It is named by Shift_JIS, for put in full font, to avoid put in the 0xA1~0xDF of the Half-width Katakana.

The Figure 6-43 is the flow chart for the S-JIS code transfers to JIS code and JIS code transfers to RA8806-J, furthermore the attachment is the program for the user to refer. i.e. the S-JIS code of character ”粟” is “88BE”, transfers to JIS code is “3040”, as the description above, JIS transfers new code to “B060”, then send “B0” and “60” in proper order to RA8806-J, it will display the Japanese Kanji font of ”粟”. i.e. the character ” 甌” of J-JIS code is “E14D”, transfers to JIS code is “612E”, as the mention above, the JIS transfer new code to “E14E”, then send “E1” and “4E” in proper order to RA8806-J, in the meantime display the Japanese Kanji font of ” 甌”.

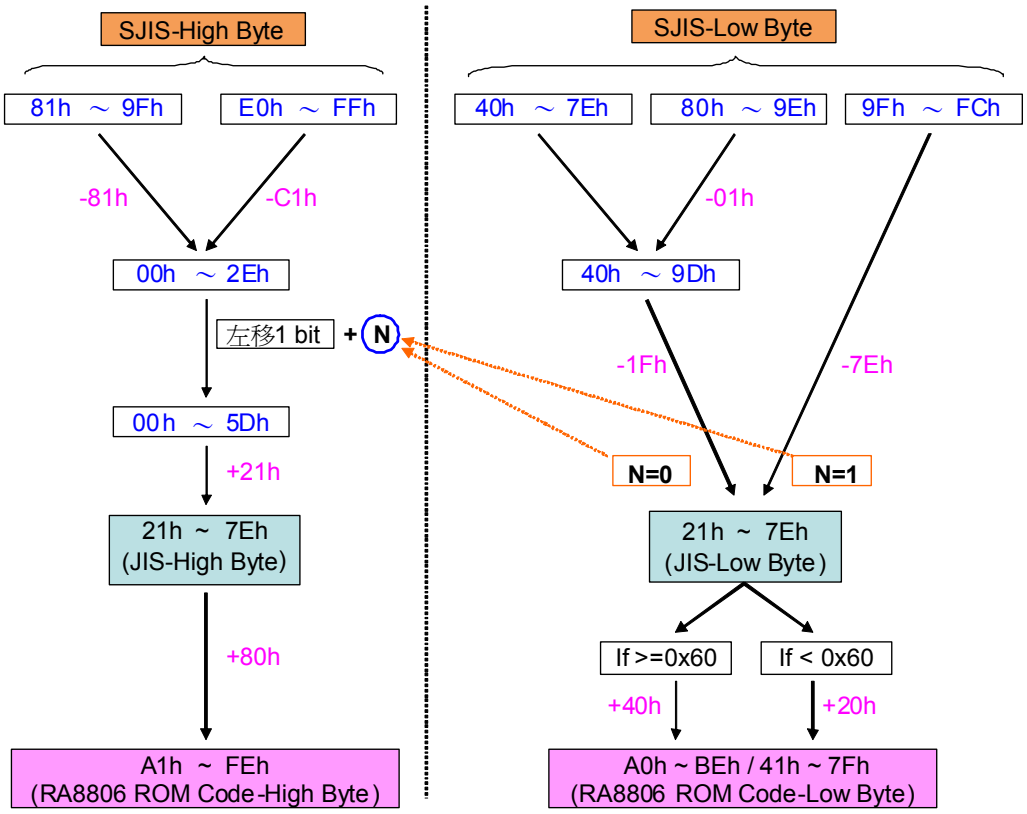


Figure 6-43 : The flow chart of the S-JIS code transfer to RA8806-J

```
//=====================================================
// Main Program
//=====================================================
void main(void)
{
 LCD_Reset();
 LCD_Initial();

 LCD_ON();       // Turn on the screen
 LCD_Clear();    // Clear display memory all
 LCD_Text();     // Switch display mode to “Text mode”

 LCD_GotoXY(0, 0);
 LCD_Print_J_Str(SJIS_string, 20);    // SJIS_string is a string include Shifit - JIS Code
}

//=====================================================
// Subroutine : Transform SJIS to JIS
//=====================================================
void LCD_Print_JIS_Str(uchar *ptr, int char_num)     // string pointer , char numbers
{
 int temp = 0;
 unsigned char SJIS_HB, SJIS_LB, JIS_HB, JIS_LB, LSB;
```

```
while(temp < char_num)
{
  if(ptr[temp] <= 0x7F)      // ASCII Code : 0x00 ~ 0x7F
  {
    LCD_DataWrite(ptr[temp]);
    temp++;
  }
  else    // Full Size font display
  {
      SJIS_HB = ptr[temp];
    temp++;
    SJIS_LB = ptr[temp];
    temp++;

    //=======================================
    // Transform SJIS Low Byte
    //=======================================
    if(SJIS_LB >= 0x9F)
    {
      JIS_LB = SJIS_LB - 0x7E;
      LSB = 0x01;
    }
    else if(SJIS_LB <= 0x7E)
    {
      JIS_LB = SJIS_LB - 0x1F;
      LSB = 0x00;
    }
    else
    {
      JIS_LB = SJIS_LB - 0x20;
      LSB = 0x00;
    }

    //=======================================
    // Transform SJIS High Byte
    //=======================================
    if(SJIS_HB >= 0xE0)
    {
      SJIS_HB = SJIS_HB - 0xC1;
    }
    else if(SJIS_HB <= 0x9F)
    {
      SJIS_HB = SJIS_HB - 0x81;
    }

    JIS_HB = (((SJIS_HB << 1) & 0xFE) | LSB) + 0x21;

    //=======================================
    // Write JIS Code
    //=======================================
    JIS_HB = JIS_HB + 0x80;

    if(JIS_LB >= 0x60)
      JIS_LB = JIS_LB + 0x40;
    else
      JIS_LB = JIS_LB + 0x20;
```

```
        LCD_DataWrite(JIS_HB);
        LCD_DataWrite(JIS_LB);
    }
  }
}
```

6-10-1-4 Bold and Inverse

The RA8806 supports the bold and inverse font. The user only needs to set up the related register bit that before send the character code to RA8806. Figure 6-44 shows the examples.

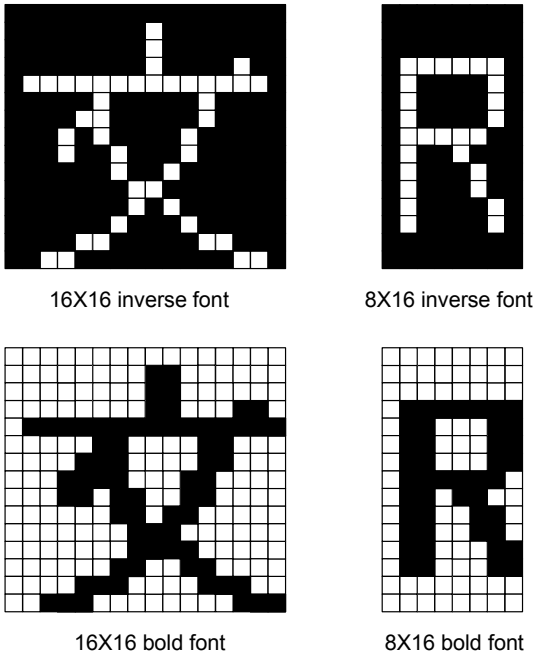


Figure 6-44 : Inverse and Bold Font Example.

Table 6-24

Reg.	Bit_Num	Description	Reference
WLCR	Bit 3	Text Mode Selection	REG[00h]
	Bit 2	Set Display On/Off Selection	
	Bit 1	Blink Mode Selection	
	Bit 0	Inverse Mode Selection	
WCCR	Bit 4	Bold Font (Character Mode Only)	REG[10h]

6-10-1-5 Two Layer Display

The RA8806 embedded two DDRAM for two layers display. The register MAMR is used to show the visible display for DDRAM1 and DDRAM2. It provides six display modes:

Display DDRAM1
Display DDRAM2
Display DDRAM1 OR DDRAM2
Display DDRAM1 XOR DDRAM2
Display DDRAM1 NOR DDRAM2
Display DDRAM1 AND DDRAM2

Please refer Figure 6-45 and Register description of MAMR Bit[6:4] and Bit[3:2].

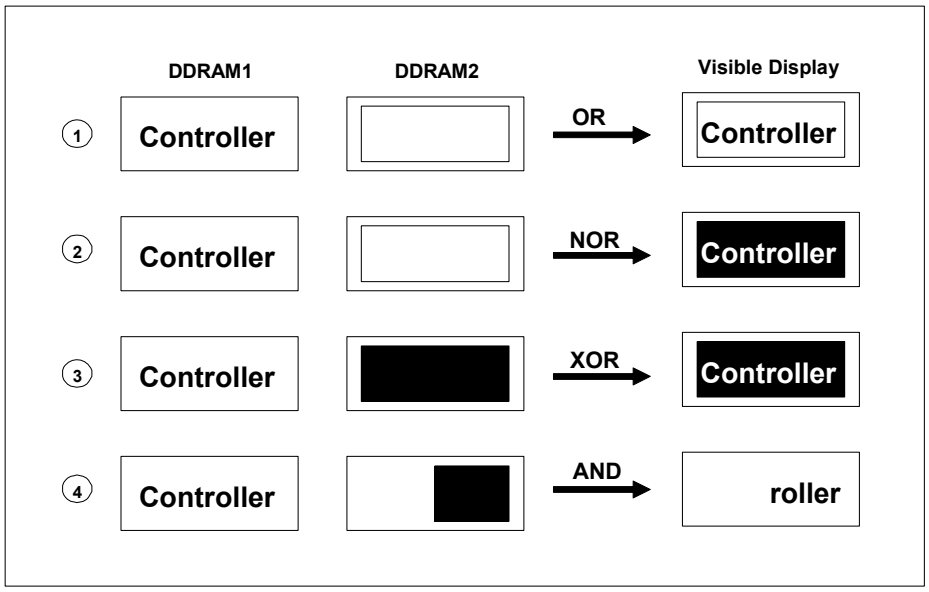


Figure 6-45 : Two Layers Display

Table 6-25

Reg.	Bit_Num	Description	Reference
MAMR	Bit [6:4]	Display Layer Selection	REG[12h]
	Bit [3:2]	Two Layer Mode Selection	

6-10-1-6 Line Gap

The RA8806 provide Line Gap feature. Especially in Chinese display, if add some space in each line will look better. The range of line gap is 1 ~ 16 pixel. Once the line gap is setup, the cursor will automatically move to property position for each line.

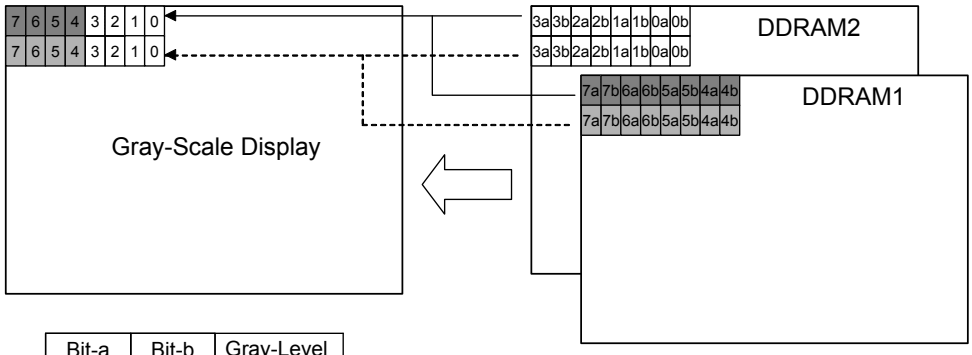
Setting the low nibble of the register CHWI, user can adjust the line gap. To deserve to be mentioned, when RA8806 operating in 90 degree mode, the line gap is either 0 pixel or 8 pixels, no matter what the font size. The selection of two conditions is according to Bit-3 of the register CHWI.

Table 6-26

Reg.	Bit_Num	Description	Reference
CHWI	Bit [7:4]	Set Cursor Height	REG[11h]
	Bit [3:2]	Set Line Gap	

6-10-2 Gray Scale Display

The RA8806 also provide 4-gray-scale display implemented by FRC method. In gray level display mode, RA8806 combines the data of DDRAM1 and DDRAM2 as a gray picture. Each gray pixel occupies 2 bits of DDRAM data for display. Data [00b] indicate an empty pixel display and [11b] is solid display. [01b] & [10b] will be time sharing as 1/3 and 2/3 lightness. Please refer to Figure 6-46.



Bit-a	Bit-b	Gray-Level
0	0	1
0	1	2
1	0	3
1	1	4

DDRAM1: Display Data RAM #1
DDRAM2: Display Data RAM #2
In Gray-Scale mode, both DDRAM are used.

Figure 6-46 : Mapping Rule of 4-Gray-Scale Display

According to the data write order, RA8806 support 2 data input methodology for Gray-Scale display. User can write the data as the same way of mono graphic display.

1. Horizontal moving first then Vertical, refer Figure 6-47.
2. Vertical moving first then Horizontal, refer Figure 6-48.

Horizontal-Write (Panel Resolution: 320 x 240)

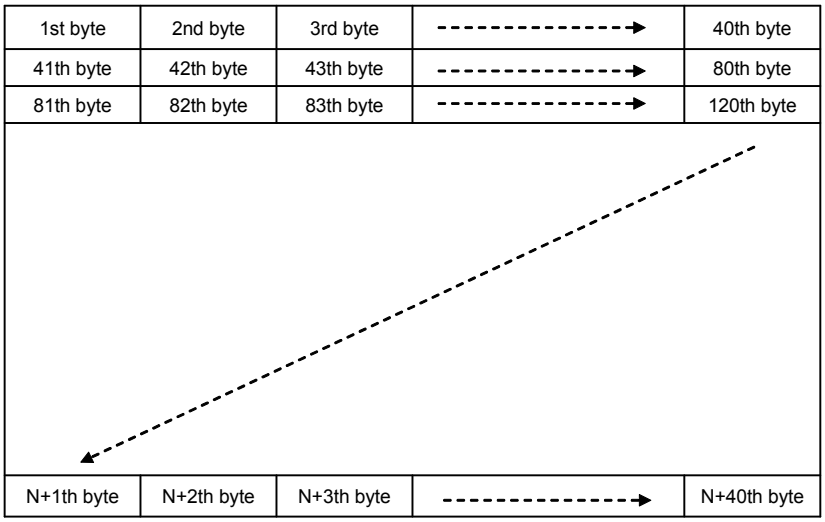


Figure 6-47 : Horizontal Write

Vertical-Write (Panel Resolution: 320 x 240)

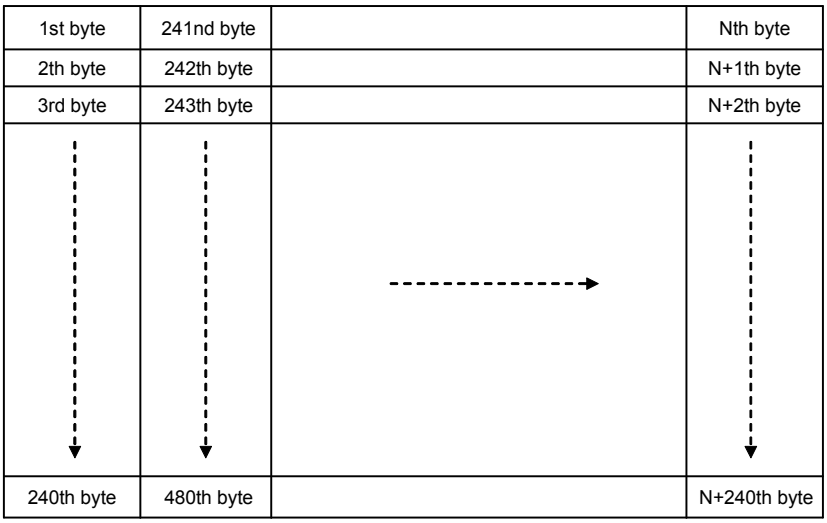


Figure 6-48 : Vertical Write

About the register setting, please refer to following register:

Table 6-27

Reg.	Bit_Num	Description	Reference
MAMR	Bit 7	Cursor Auto Shifting Direction	REG[12h]
	Bit [6:4]	Gray Mode selection	

The picture below is an example of 4-gray-level picture, the program example are attached below it:

Figure 6-49 : 4-Gray-Scale Example

Program Example:

```
 LCD_Graphic();                 // Set graphic mode
Gray_Mode();                    // Set REG[12h] Bit[6:4] to 000b, as gray level display
 Scan_Diret_H_V();              // Set REG[12h] Bit-7 to 0, Cursor moves from left to right
 LCD_GotoXY( 0 , 0 );
 LCD_CmdWrite( 0xB0 );          // Write memory command
 for ( i = 0 ; I < 4800 ; i++ ) // Write data
 {
     LCD_DataWrite( 0x00 ) ;    // Gray Level 1 ( 00 )
 }
for ( i = 0 ; I < 4800 ; i++ )
 {
     LCD_DataWrite( 0x55 ) ;    // Gray Level 2 ( 01 )
 }
for ( i = 0 ; I < 4800 ; i++ )
 {
     LCD_DataWrite( 0xAA ) ;    // Gray Level 3 ( 10 )
 }
for ( i = 0 ; I < 4800 ; i++ )
 {
     LCD_DataWrite( 0xFF ) ;    // Gray Level 4 ( 11 )
 }
```

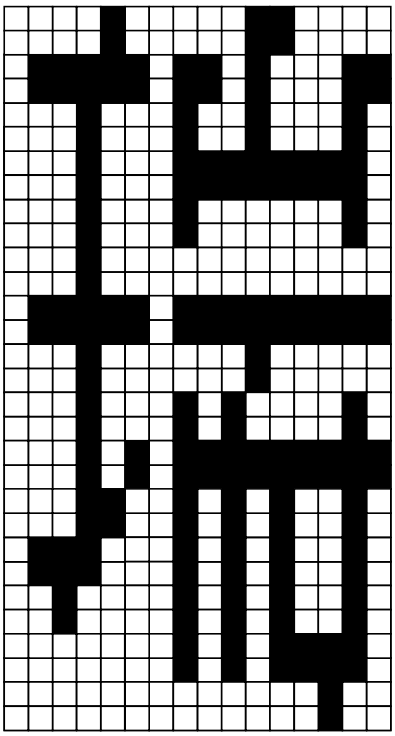
6-10-3 Font Size Adjustment and Font Write-Time

RA8806 supports Font size adjustment. There are two types of characters stored in RA8806 font ROM - Half size (8x16 pixels) and full size (16x16 pixels) characters. The“ Font size adjustment” function supports the font enlargement from 1 to 4 times bigger in vertical and horizontal respective directions. Programmer can adjust the character size from setting FVHT[7:4] :

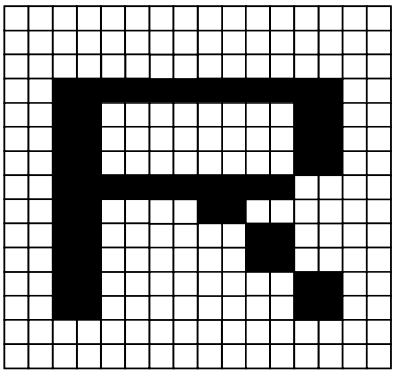
Table 6-28

Reg.	Bit_Num	Description	Reference
FVHT	Bit [7:6]	Horizontal width adjustment for character	REG[F1h]
	Bit [5:4]	Vertical width adjustment for character	

Figure 6-50 shows the example of character enlargement:



1x2 enlarged 16x16 font



2x1 enlarged 8x16 font

Figure 6-50 : Font Size Adjustment

Additionally, the Font Sized Adjustment is not only available for the character display mode, but also for the following display modes:

- For displaying the user created fonts
- 90° font and display rotation

After writing completed font code(One byte for half size font and two bytes for full-size font), it will takes an enough time for data to be written into the DDRAM. In the mean while no data can be read or wrote into DDRAM. (Please refer to Section 6-8 “Interrupt and Busy”, the description of “Memory Write Busy”). The writing time is differing from the enlargement multiplier of font size. Its calculation formula is as follows :

T_{fw} = (16*tc + 32*tc x (HW x VH)) x (1 + 10%)

T_{fw} : Font Write time
tc : System clock period
HW : Horizontal enlargement multiplier.
VH : Vertical enlargement multiplier.

Here 10% is for the flexible error consideration. Please refer to the following examples: (Estimate the system clock is 4MHz and the tc is 250ns)

Table 6-29 : Delay Time for Font Enlargement

Font enlargement multiplier	Standard delay time(μS)	Suggested delay time(μS)
1 x 1	12	13.2
2 x 2	36	39.6
3 x 3	76	83.6
4 x 4	132	145.2

In addition to calculate writing time and delay, programmer can also check the “Busy” status to know if font characters were written into the DDRAM. Please refer to the “ Status Register” for the description of “Busy”.

6-10-4 Font Vertical Display

RA8806 features of its “Font Vertical Display” function. This function enables the normal display change to vertical display. This means the original 320x240 panel can be used as a 240x320 panel to display embedded character font or data in vertical way, and no need to change scan direction or the LCD driver layout. The LCD panel could display from both vertical and horizontal ways. This function is option on setting the Bit-3 of register WCCR. When WCCR Bit-3 =1 (Font Rotation Mode), the font character will be written in vertical display. Figure 6-51 shows the example of font vertical display.

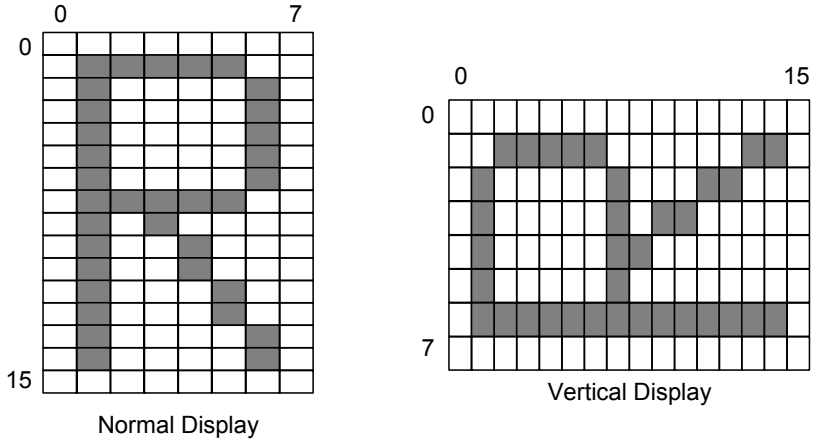


Figure 6-51: Font Vertical Display Function

Under the vertical display mode, the cursor moves from the direction of up to down. And the cursor display is also differing to normal display mode. (Please refer to Section 6-13-4 “Cursor Width and Height” for detail descriptions).

Due to the advance rotate function, if user chose a 240x160 module, then it will be very easy to use same module but as an 160x240 module. The following Figure 6-52 shows the example of 320x240 panel rotate to as 240X320 application. In normal mode, if write text font “RAIO” then the screen show as (1) of Figure 6-52. If set the T90DEG(Bit-3 of REG[10h]) = 1 and CDIR(Bit-0 of REG[01h]) = 1 first, then write text font “RAIO”, the screen will show as (2) of Figure 6-52.

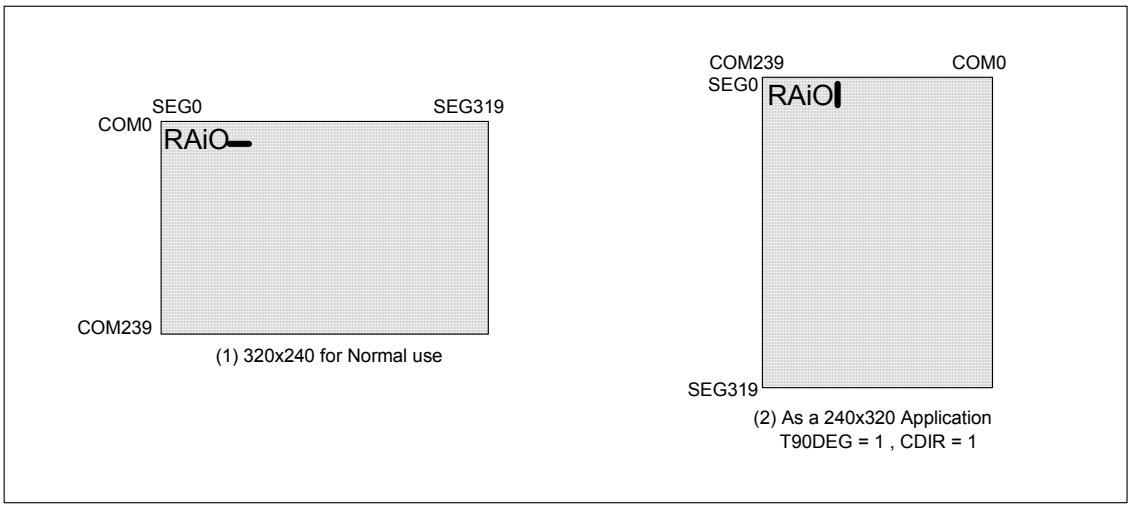


Figure 6-52 : Font Display in Rotate Mode

The relative register list about font rotation function is:

Table 6-30

Reg.	Bit_Num	Description	Reference
WCCR	Bit 3	Font rotation mode enable	REG[10h]
MISC	Bit 0	COM direction selection	REG[01h]

Besides the cursor display function, the other functions in the font rotation mode work as the same behavior as normal mode, including:

1) Font enlargement function
2) Full alignment function
3) Line Gap adjustment function
4) Scroll function

6-11 User-Defined Font

The RA8806 embedded a 512Byte CGRAM for user to create new character. The user could create font, special symbol or pattern in this memory. If user only uses one DDRAM, then the other one could also use as CGRAM.

Therefore, the RA8806 provides 3 regions for user to create new character(symbol or pattern) as following:

1. 512 Bytes CGRAM
2. 9.6K Bytes DDRAM1 (When user only shows DDRAM2 data on screen)
3. 9.6K Bytes DDRAM2 (When user only shows DDRAM1 data on screen)

Table 6-31 : Font Code for User-Created

Region	Size	Code and Range	Capacity
CGRAM	512 bytes	Half-Size : 8F00h ~ 8F1Fh Full-Size : 9F00h ~ 9F0Fh	32 Half-Size chars 16 Full-Size chars
DDRAM1	9.6 Kbytes	Half-Size : 8000h ~ 8E27h Full-Size : 9000h ~ 9E13h	600 Half-Size chars 300 Full-Size chars
DDRAM2	9.6 Kbytes	Half-Size : 8000h ~ 8E27h Full-Size : 9000h ~ 9E13h	600 Half-Size chars 300 Full-Size chars

6-11-1 Create Font in CGRAM

The CGRAM is a 512Byte RAM, so it could create 32 Half-Size or 16 Full-Size characters. The Table 6-32 is the font code mapping table in CGRAM. Note the font codes were fix in RA8806. And the Font# was defined in Bit[3:0] of REG[CURY].

Table 6-32 : CGRAM Font Code Mapping Table

Font# / Code	0		1		2		3		4		5		6		7	
Half-Size	8F00	8F01	8F02	8F03	8F04	8F05	8F06	8F07	8F08	8F09	8F0A	8F0B	8F0C	8F0D	8F0E	8F0F
Full-Size	9F00		9F01		9F02		9F03		9F04		9F05		9F06		9F07	

Font# / Code	8		9		10		11		12		13		14		15	
Half-Size	8F10	8F11	8F12	8F13	8F14	8F15	8F16	8F17	8F18	8F19	8F1A	8F1B	8F1C	8F1D	8F1E	8F1F
Full-Size	9F08		9F09		9F0A		9F0B		9F0C		9F0D		9F0E		9F0F	

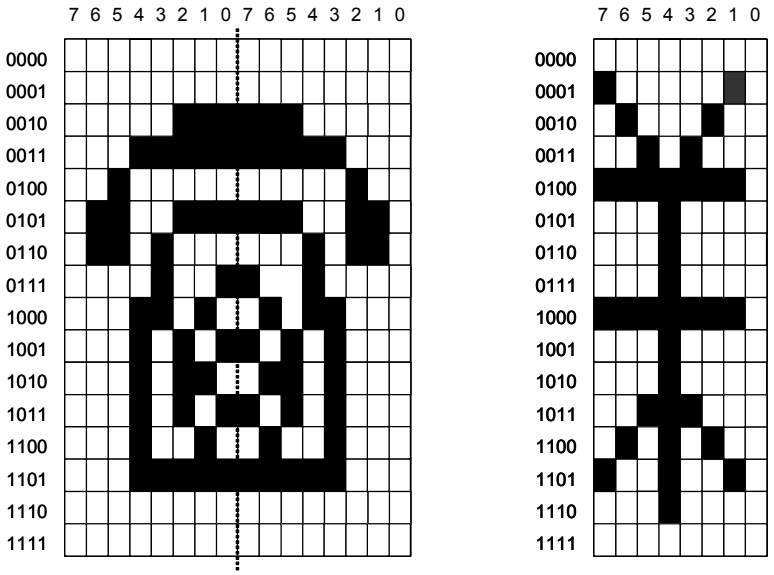


Figure 6-53 : Example of Created Font Bit-map Data

According to the user-created font register setting in CGRAM, please refer to the below table.

Table 6-33 : Register Definition for Created-font in CGRAM

Reg.	Bit_Num	Description	Reference
MAMR	Bit 7	Cursor Increase enable	REG[12h]
	Bit [6:4]	Display mode	
	Bit [1:0]	Select memory for write cycle	
CURX	Bit [4:0]	The Bit[4:0] is the address for writing bit-map data of create font. When create a full-size font, normally set to 0. When create an odd half-size font, normally set to 0, and set 10h for even font.	REG[60h]
CURY	Bit [3:0]	Indicate which font code data will be created.	REG[70h]

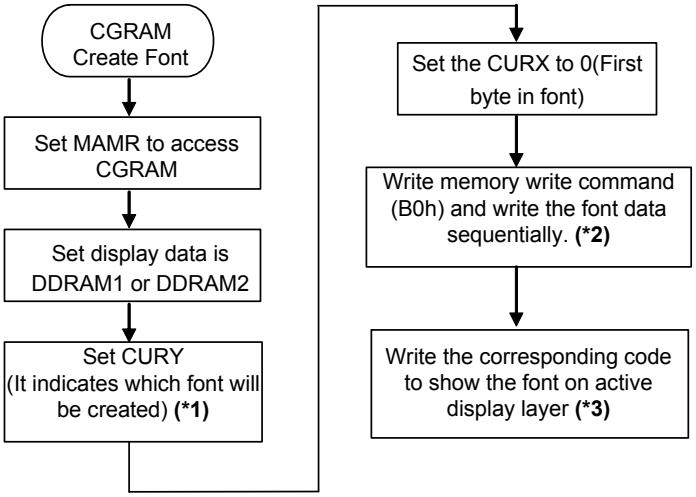


Figure 6-54 : The Flow-chart of User-created Font on CGRAM

Note:

1. CGRAM contains 512 bytes and can contain 16 16x16 full-size or 32 8x16 half-size user-created font. Each full-size font contains 32bytes.
2. For created full-size font, after CURY is assign(0~15) and CURX is set to 0, a created-font bit-map can be filled by the 32 continuous memory writes cycle. And then CURY will be increased by 1. Please refer the following *Program Example(1)*. The result is show (1) of Figure 6-53 on screen.
3. For created half-size font, after CURY is assign(0~15) and CURX is set to 0(odd font) or 10h(even font), a created-font bit-map can be filled by the 16 continuous memory writes cycle. Please refer the following *Program Example(2)*. The result is show (2) of Figure 6-53 on screen.
4. The CGRAM font code is 8F00~8F1F for Half-size font, and 9F00~9F0Fh for full-size font. Refer to Table 6-31.

Program Example(1):

```
// Create a full-size font in CGRAM and show on the screen.
// font_data[] = {00, 00, 07, 1F, 20, 67, 68, 09, 1A, 15, 16, 15, 12, 1F, 00, 00,
// 00, 00, E0, F8, 04, E6, 16, 90, 58, A8, 68, A8, 48, F8, 00, 00}

Access_CGRAM();                     // MAMR[1:0] = 00b
Only_Show_DDRAM1();                 // MAMR[6:4] = 001b
LCD_CmdWrite( CURY );               // Create 6th full-size character
LCD_DataWrite( 0x05 );
LCD_CmdWrite( CURX );               // Write font-data from 1st byte
LCD_DataWrite( 0x00 );
LCD_CmdWrite( MWCR);                // B0h, Memory write command
for( i = 0; i < 32;  i++ )          // 32 continuous write for font bit-map
  {  LCD_DataWrite( font_data[i] );
     Delay2us(50);
  }
Access_DDRAM1();
LCD_Text();
LCD_GotoXY( 5 , 5 );                // Set Cursor position ( 5 , 5 )
LCD_CmdWrite( MWCR);                // Corresponding font code( 9F05h )
LCD_DataWrite( 0x9F );
```

```
LCD_DataWrite( 0x05 );          // Show as (1) of Figure 6-53
```

Program Example(2):

```
// Create a half-size font in CGRAM and show on the screen.
// font_data[] = {00, 82, 44, 28, FE, 10, 10, 10, FE, 10, 10, 38, 54, 92, 10, 00}
Access_CGRAM();                    // MAMR[1:0] = 00b
Only_Show_DDRAM1();                // MAMR[6:4] = 001b
LCD_CmdWrite( CURY );              // Create 6th half-size char
LCD_DataWrite( 0x02 );
LCD_CmdWrite( CURX );              // Write font-data from 1st byte
LCD_DataWrite( 0x10 );
LCD_CmdWrite( MWCR);               // B0h, Memory write command
for( i = 0; i < 16;  i++ )         // 16 continuous write for font bit-map
  {
     LCD_DataWrite( font_data[i] );
     Delay2us(50);
  }
Access_DDRAM1();
LCD_Text();
LCD_GotoXY( 5 , 5 );               // Set Cursor position ( 5 , 5 )
LCD_CmdWrite( MWCR);               // Corresponding font code( 8F05h )
LCD_DataWrite( 0x8F );
LCD_DataWrite( 0x05 );           // Show as (2) of Figure 6-53
```

6-11-2 Create Font in DDRAM

The DDRAM are 9.6KByte RAM, so it could create 600 Half-Size or 300 Full-Size characters. Table 6-35 is the font code mapping table in DDRAM for Half-Size characters. Table 6-36 is the font code mapping table in DDRAM for Full –Size characters. These font codes were also fix in RA8806. The register CURX and CURY are used to point the address that bit-map data to create font.

According to the register setting of user-created font in DDRAM, please refer to the below table.

Table 6-34 : Register Definition for Created-font in DDRAM

Reg.	Bit_Num	Description	Reference
MAMR	Bit 7	Cursor Increase enable	REG[12h]
	Bit [6:4]	Display mode	
	Bit [1:0]	Select memory for write cycle	
CURX	Bit [5:0]	Both registers are Indicate which font code data will be created.	REG[60h]
CURY	Bit [7:0]		REG[70h]

Table 6-35 : DDRAM1 and DDRAM2 Half-size Font Code Mapping

	CURX / CURY	0	1	2	. . .	26	27
0	00	8000	8001	8002	. . .	8026	8027
1	10	8100	8101	8102	. . .	8126	8127
2	20	8200	8201	8202	. . .	8226	8227
3	30	8300	8301	8302	. . .	8326	8327
4	40	8400	8401	8402	. . .	8426	8427
5	50	8500	8501	8502	. . .	8526	8527
6	60	8600	8601	8602	. . .	8626	8627
7	70	8700	8701	8702	. . .	8726	8727
8	80	8800	8801	8802	. . .	8826	8827
9	90	8900	8901	8902	. . .	8926	8927
10	A0	8A00	8A01	8A02	. . .	8A26	8A27
11	B0	8B00	8B01	8B02	. . .	8B26	8B27
12	C0	8C00	8C01	8C02	. . .	8C26	8C27
13	D0	8D00	8D01	8D02	. . .	8D26	8D27
14	E0	8E00	8E01	8E02	. . .	8E26	8E27

Table 6-36 : DDRAM1 and DDRAM2 Full-size Font Code Mapping Table

	CURX / CURY	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27
0	00	9000	9001	9002	9003	9004	9005	9006	9007	9008	9009	900A	900B	900C	900D	900E	900F	9010	9011	9012	9013																				
1	10	9100	9101	9102	9103	9104	9105	9106	9107	9108	9109	910A	910B	910C	910D	910E	910F	9110	9111	9112	9113																				
2	20	9200	9201	9202	9203	9204	9205	9206	9207	9208	9209	920A	920B	920C	920D	920E	920F	9210	9211	9212	9213																				
3	30	9300	9301	9302	9303	9304	9305	9306	9307	9308	9309	930A	930B	930C	930D	930E	930F	9310	9311	9312	9313																				
4	40	9400	9401	9402	9403	9404	9405	9406	9407	9408	9409	940A	940B	940C	940D	940E	940F	9410	9411	9412	9413																				
5	50	9500	9501	9502	9503	9504	9505	9506	9507	9508	9509	950A	950B	950C	950D	950E	950F	9510	9511	9512	9513																				
6	60	9600	9601	9602	9603	9604	9605	9606	9607	9608	9609	960A	960B	960C	960D	960E	960F	9610	9611	9612	9613																				
7	70	9700	9701	9702	9703	9704	9705	9706	9707	9708	9709	970A	970B	970C	970D	970E	970F	9710	9711	9712	9713																				
8	80	9800	9801	9802	9803	9804	9805	9806	9807	9808	9809	980A	980B	980C	980D	980E	980F	9810	9811	9812	9813																				
9	90	9900	9901	9902	9903	9904	9905	9906	9907	9908	9909	990A	990B	990C	990D	990E	990F	9910	9911	9912	9913																				
10	A0	9A00	9A01	9A02	9A03	9A04	9A05	9A06	9A07	9A08	9A09	9A0A	9A0B	9A0C	9A0D	9A0E	9A0F	9A10	9A11	9A12	9A13																				
11	B0	9B00	9B01	9B02	9B03	9B04	9B05	9B06	9B07	9B08	9B09	9B0A	9B0B	9B0C	9B0D	9B0E	9B0F	9B10	9B11	9B12	9B13																				
12	C0	9C00	9C01	9C02	9C03	9C04	9C05	9C06	9C07	9C08	9C09	9C0A	9C0B	9C0C	9C0D	9C0E	9C0F	9C10	9C11	9C12	9C13																				
13	D0	9D00	9D01	9D02	9D03	9D04	9D05	9D06	9D07	9D08	9D09	9D0A	9D0B	9D0C	9D0D	9D0E	9D0F	9D10	9D11	9D12	9D13																				
14	E0	9E00	9E01	9E02	9E03	9E04	9E05	9E06	9E07	9E08	9E09	9E0A	9E0B	9E0C	9E0D	9E0E	9E0F	9E10	9E11	9E12	9E13																				

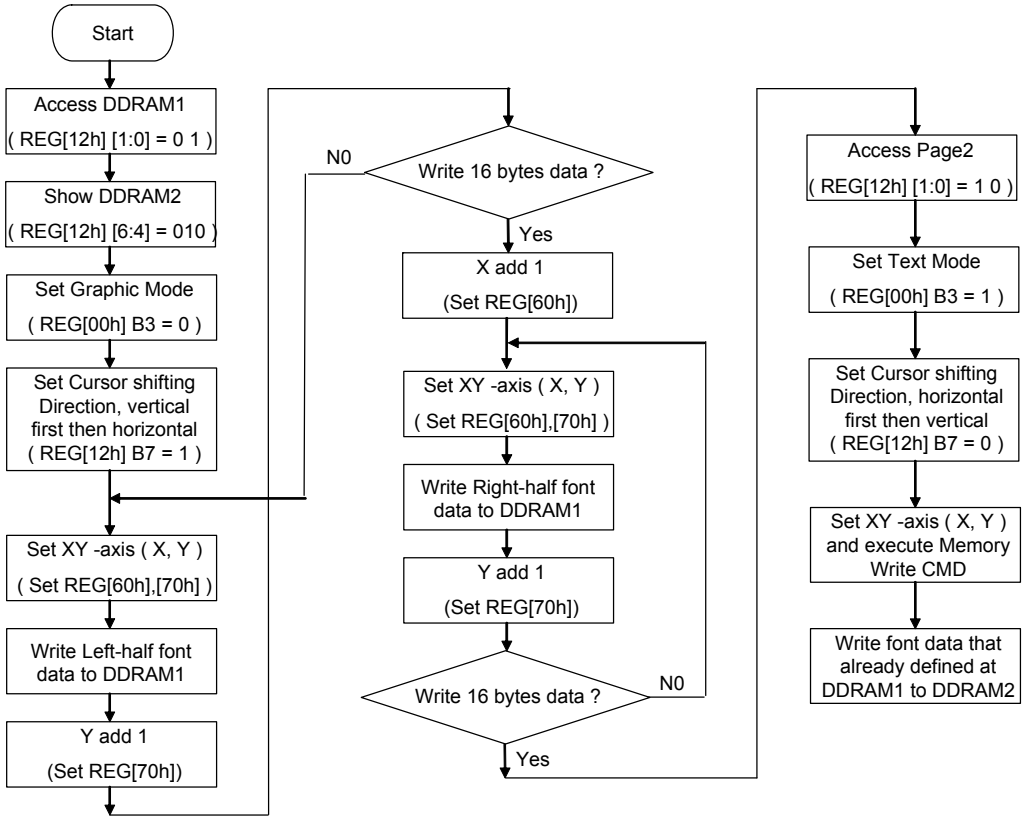


Figure 6-55 : Flowchart of User-created font in DDRAM1

The above flow-chart is an example to create a Full-size in DDRAM1 and show the new font on screen(DDRAM2). At first, user have to set up the CURX and CURY to define which code data will be created, and then write 32Bytes bit-map data to DDRAM1. The CURX need to set twice for each 16Bytes data – the left and right of full-size font. Please refer the following *Program Example(3)*. If we want to create a full-size font as (1) of Figure 6-53 to mapping code – 9205h in Table 6-36. The result is show (1) of Figure 6-53 on screen.

Program Example(3):

```
// Create a full-size font in DDRAM1 and show on the screen.
// font_data[] = {00, 00, 07, 1F, 20, 67, 68, 09, 1A, 15, 16, 15, 12, 1F, 00, 00,
// 00, 00, E0, F8, 04, E6, 16, 90, 58, A8, 68, A8, 48, F8, 00, 00}

Access_DDRAM1();                // MAMR[1:0] = 01
Only_Show_DDRAM2();             // MAMR[6:4] = 010

LCD_Graphic();                  // WLCR. Bit-3 = 0
Cursor_Shift_Direct_VH();       // MAMR. Bit-7 = 1
                                // Set the cursor moving in vertical
LCD_GotoXY(0x0A, 0x20);         // Write the left part of full-size font
LCD_CmdWrite( MWCR);            // Memory write command
for(i=0;i<16;i++)
{
    LCD_DataWrite(font_data_L[i]);
    Delay2us(50);
}

LCD_GotoXY(0x0B, 0x20);         // Write the right part of full-size font
LCD_CmdWrite( MWCR);            // Memory write command
for(i=16;i<32;i++)
{
    LCD_DataWrite(font_data_R[i]);
    Delay2us(50);
}

Access_DDRAM2();                // MAMR[1:0] = 10
LCD_Text();                     // WLCR.Bit-3 = 1
Cursor_Shift_Direct_HV();       // MAMR. Bit-7 = 0

LCD_GotoXY( 3 , 3 );            // set coordinate to ( 3 , 3 )
LCD_CmdWrite( MWCR);            // Memory write command
LCD_DataWrite(0x92);            // Write the code(9205h) of user-defined font
LCD_DataWrite(0x05);            // Show as (1) of Figure 6-53
Delay2us(50);
```

6-11-3 Create Symbol

RA8806 provides font creator function that allows user to design and create new fonts, special symbols or logos and store into its embedded CGRAM or DDRAM. The user-created characters each defined as a full-size (16x16) or half-size (8x16) character bitmap font formats, thus 24x16, 40x16, 16x32 or bigger sized fonts are also possible to create accordingly. The following Figure 6-56 represents a user created symbol with size 24x16.

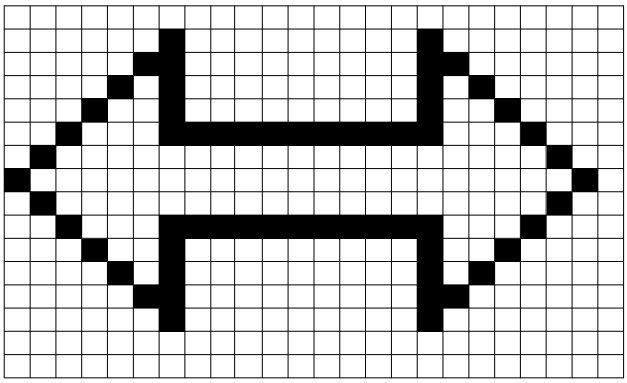


Figure 6-56 : A 24x16 Symbol Example

For creating a new symbol, in addition to write command to CGRAM or DDRAM, note the setting of the font codes before trying to write characters to be displayed. See the following 2 examples of creating this 24x16 symbol in DDRAM:

When the first address - (CURX, CURY) of data write to DDRAM is (0, 0), the font code of this symbol are 0x8000, 0x8001 and 0x8002, as Figure 6-57 below:

When the first address of date write to DDRAM is (2, 10h), the font code of this symbol are 0x8105，0x8106，0x8107 and so on. Please refer the following *Program Example(4)*. The result is show Figure 6-56 on screen.

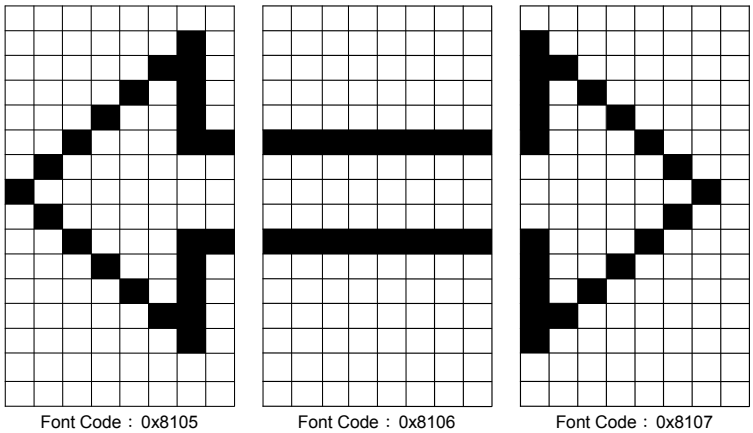


Figure 6-57 : Font Code of User's Created Symbol for Program Example(4)

Program Example(4):

```
// Create a 24x16 symbol in DDRAM1 and show on the screen.
// font_data[] = {00, 02, 06, 0A, 12, 23, 40, 80, 40, 23, 12, 0A, 06, 02, 00, 00,
// 00, 00, 00, 00, 00, FF, 00, 00, 00, FF, 00, 00, 00, 00, 00, 00,
// 00, 80, C0, A0, 90, 88, 04, 02, 04, 88, 90, A0, C0, 80, 00, 00}

Access_DDRAM1();                // MAMR[1:0] = 01
Only_Show_DDRAM2();             // MAMR[6:4] = 010

LCD_Graphic();                  // WLCR. Bit-3 = 0
Cursor_Shift_Direct_VH();       // MAMR. Bit-7 = 1
                                // Set the cursor moving in vertical
LCD_GotoXY(0x02, 0x10);         // Write the left part of symbol
LCD_CmdWrite( MWCR);            // Memory write command
for(i=0;i<16;i++)
{
    LCD_DataWrite(font_data_L[i]);
    Delay2us(50);
}

LCD_GotoXY(0x03, 0x10);         // Write the middle part of symbol
LCD_CmdWrite( MWCR);            // Memory write command
for(i=16;i<32;i++)
{
    LCD_DataWrite(font_data_R[i]);
    Delay2us(50);
}

LCD_GotoXY(0x04, 0x10);         // Write the right part of symbol
LCD_CmdWrite( MWCR);            // Memory write command
for(i=32;i<48;i++)
{
    LCD_DataWrite(font_data_R[i]);
    Delay2us(50);
}


Access_DDRAM2();                // MAMR[1:0] = 10
LCD_Text();                     // WLCR.Bit-3 = 1
Cursor_Shift_Direct_HV();       // MAMR. Bit-7 = 0

LCD_GotoXY( 3 , 3 );            // set coordinate to ( 3 , 3 )
LCD_CmdWrite( MWCR);            // Memory write command
LCD_DataWrite(0x81);            // Write the code(8102, 8103, 8104) of symbol
LCD_DataWrite(0x02);            // Show as Figure 6-57
Delay2us(10);
LCD_DataWrite(0x81);
LCD_DataWrite(0x03);
Delay2us(10);
LCD_DataWrite(0x81);
LCD_DataWrite(0x04);
Delay2us(10);
```

Besides, note the display address for each of the font code to be display on the LCD screen,

especially under Rotate 90 degree and Font Enlargement mode.

6-12 Scroll Function

6-12-1 Horizontal Scrolling

The RA8806 provide Horizontal Scrolling function. You can assign an area by Register BGSG and EDSG. Once start the horizontal scrolling, the assigned area will shift step by step and each step is 8 pixels width.

About Horizontal Scrolling register setting, please refer to the list below.

Table 6-37

Reg.	Bit_Num	Description	Reference
ADSR	Bit 2	SCR_DIR(Scroll direction)	REG[03h]
	Bit 1	SCR_HV(Scroll Horizontal/Vertical)	
	Bit 0	SCR_EN(Scroll Enable)	
BGSG	Bit [5:0]	The SEG start address in scroll mode	REG[61h]
EDSG	Bit [5:0]	The SEG end address in scroll mode	REG[62h]
BGCM	Bit [7:0]	The COM start address in scroll mode	REG[71h]
EDCM	Bit [7:0]	The COM end address in scroll mode	REG[72h]

(1)Flowchart: The flow-chart of horizontal scrolling is as following:

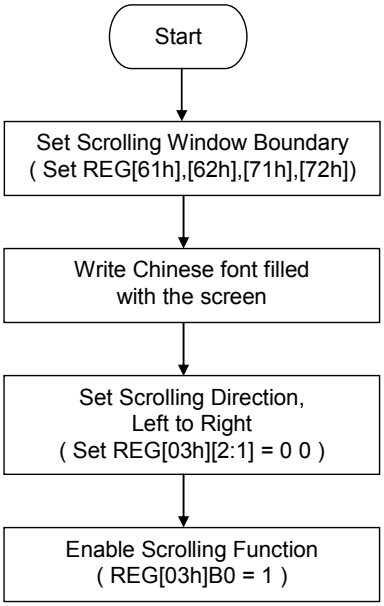


Figure 6-58 : Horizontal Scrolling

(2)Program Example:

```
LCD_Text();                 // Text Mode
//================================================
// Set Scrolling Window
//================================================
LCD_CmdWrite(0x61);         // SEG Start Position of Scrolling Mode
LCD_DataWrite(0x05);
LCD_CmdWrite(0x62);         // SEG End Position of Scrolling Mode
LCD_DataWrite(0x22);

LCD_CmdWrite(0x71);         // COM Start Position of Scrolling Mode
LCD_DataWrite(0x20);
LCD_CmdWrite(0x72);         // COM End Position of Scrolling Mode
LCD_DataWrite(0xd0);

for( i = 0 ; i < 300 ; i++ )        // Write the font to fill the screen
{
    LCD_DataWrite( RAiO [ i ] );
    Delay2us(50);
}

//================================================
// Set Scrolling Direction and Enable scroll function
//================================================
LCD_CmdWrite(0x03);
LCD_DataWrite(0x01);        // L → R
//  LCD_DataWrite(0x05);    // R → L
```

6-12-2 Vertical Scrolling

The RA8806 also provide Vertical Scrolling function. You can start the scrolling by control Register ADSR Bit-2. Once start the vertical scrolling, the whole screen will shift step by step and each step is 1 pixel height.

(1)Flowchart: The flow-chart of vertical scrolling is as following:

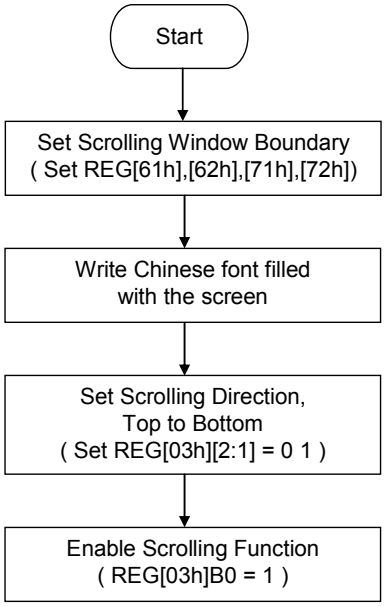


Figure 6-59 : Vertical Scrolling

(2)Program Example:

```
LCD_Text();                 // Text Mode

LCD_CmdWrite(0x61);         // SEG Start Position of Scrolling Mode
LCD_DataWrite(0x05);
LCD_CmdWrite(0x62);         // SEG End Position of Scrolling Mode
LCD_DataWrite(0x22);

LCD_CmdWrite(0x71);         // COM Start Position of Scrolling Mode
LCD_DataWrite(0x20);
LCD_CmdWrite(0x72);         // COM End Position of Scrolling Mode
LCD_DataWrite(0xd0);

for( i = 0 ; i < 300 ; i++ )    // Fill Chinese font on the screen
{
    LCD_DataWrite( RAiO [ i ] );
    Delay2us(50);
}

LCD_CmdWrite(0x03);         // Scrolling Enable
LCD_DataWrite(0x03);        // T → B
// LCD_DataWrite(0x07);     // B → T
```

6-13 Cursor

6-13-1 Cursor Position and Shift

The cursor-moving unit of segment is one byte(or eight pixels). But the moving unit of common is one pixel. The cursor position is controlled by Register CURX and CURY for both text and graphics mode. You can also setup the Auto-Increase mode for write to DDRAM or read data from DDRAM. The cursor-moving boundary depends on the active window.

6-13-2 Full Alignment

The “Full alignment function” provides the “Auto-Align” feature when full-size and half-size characters combined display. Programmer can adjust it from the settings of Bit-6 of WCCR. (Refer to register WCCR for detail description.)

Appendix A When Bit-6 is 1, the auto-align feature is enabled.
Appendix B When Bit-6 is 0, the auto-align feature is not being used.

The Figure 6-60 and Figure 6-61 shows the above settings.

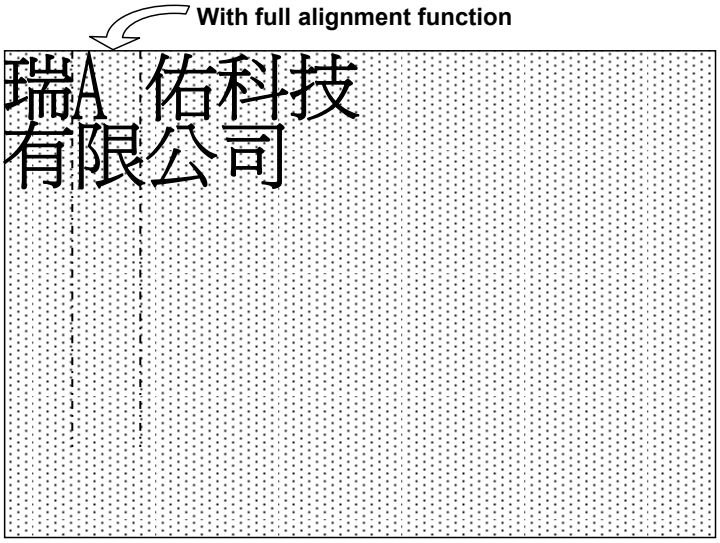


Figure 6-60 : Font Write with Full Alignment

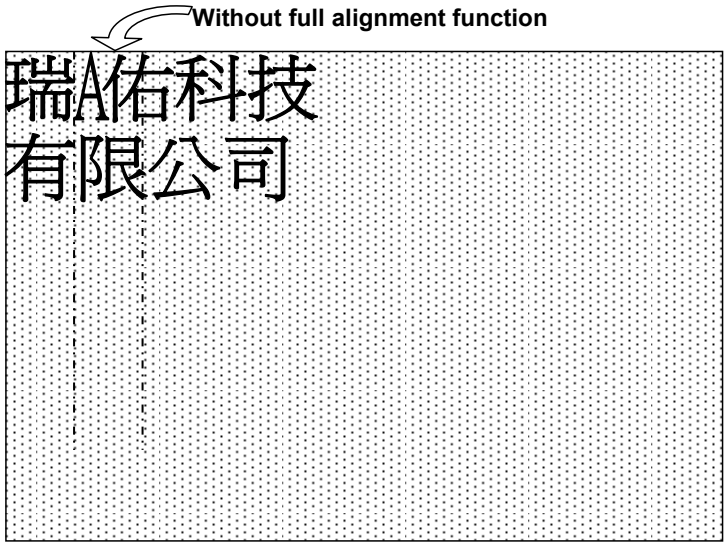


Figure 6-61 : Font Write without Full Alignment

When the ”Full-alignment” feature is used, the display shows all the full-size characters start from the set start address. And there will be an even bytes alignment space between these fonts. For half-size character, the function doesn’t restrict the position of it.

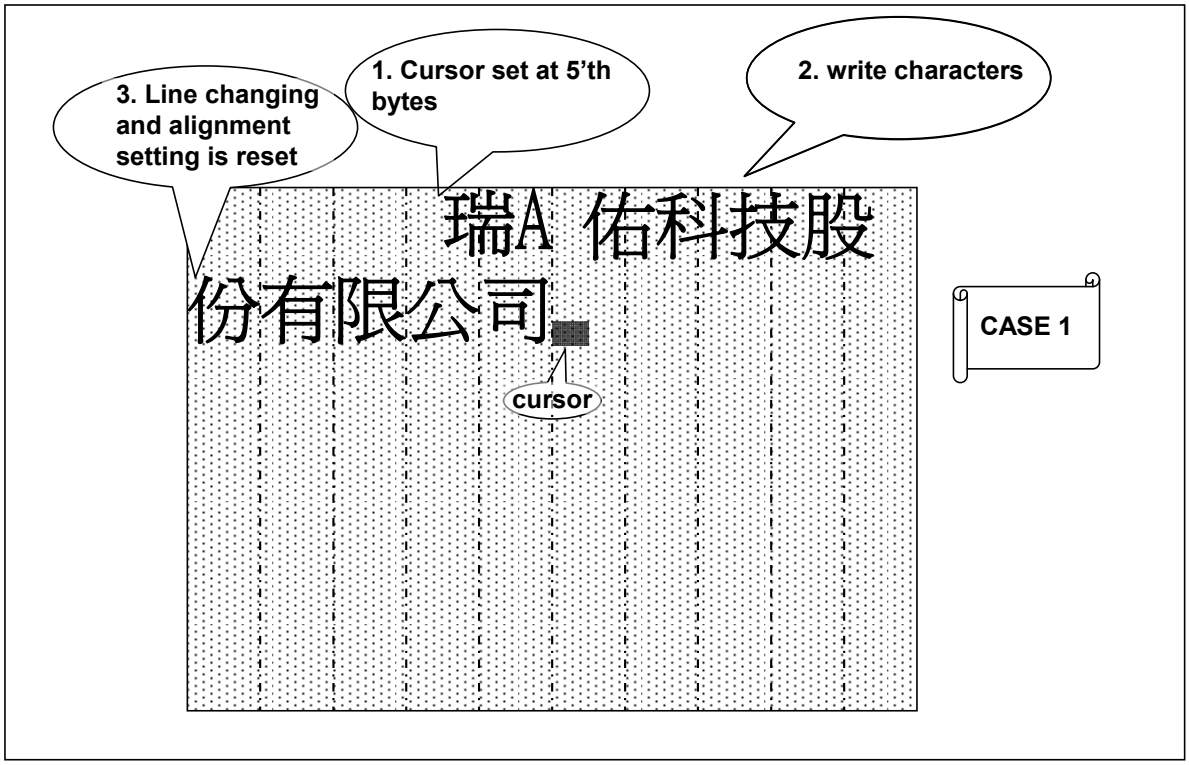


Figure 6-62 : Full-alignment Example(1)

The full-alignment display rule is base on cursor position be set at 1st row. After changing row, the alignment base is reset. That is, the start segment address of active window. There are serial conditions for the changing line with the full alignment feature. Please refer to below figures for the condition.

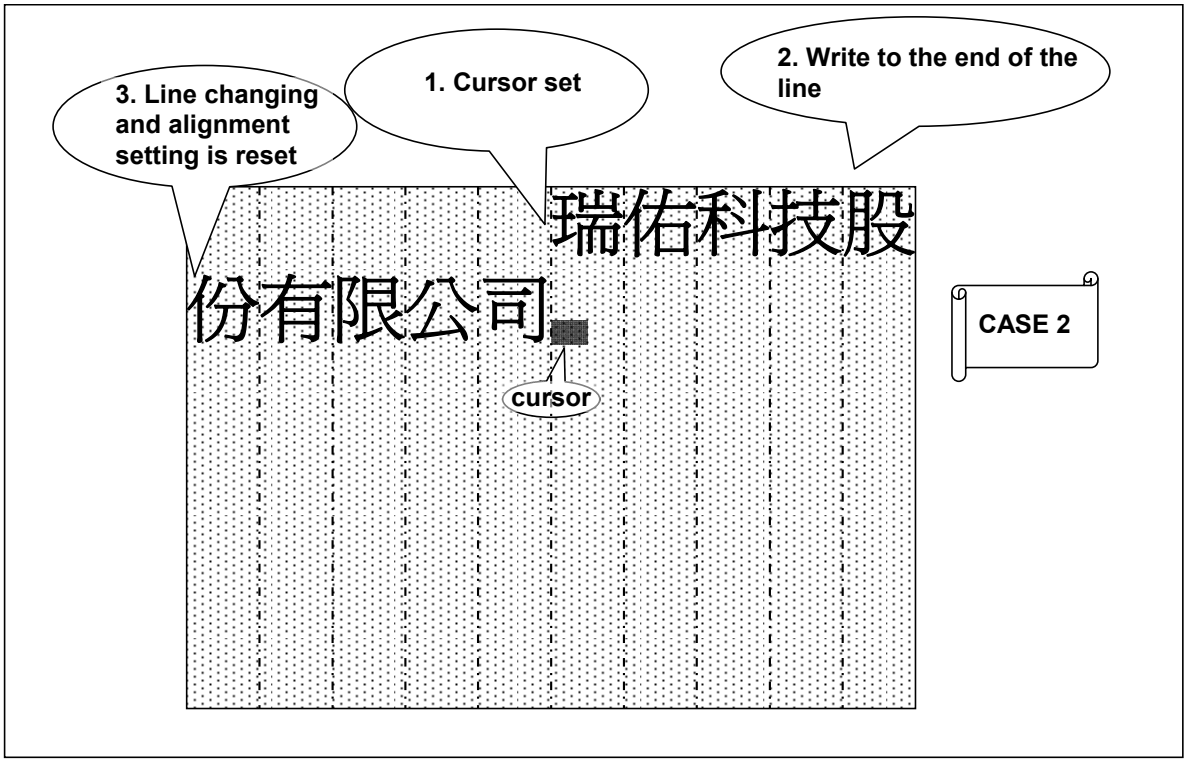


Figure 6-63 : Full-alignment Example(2)

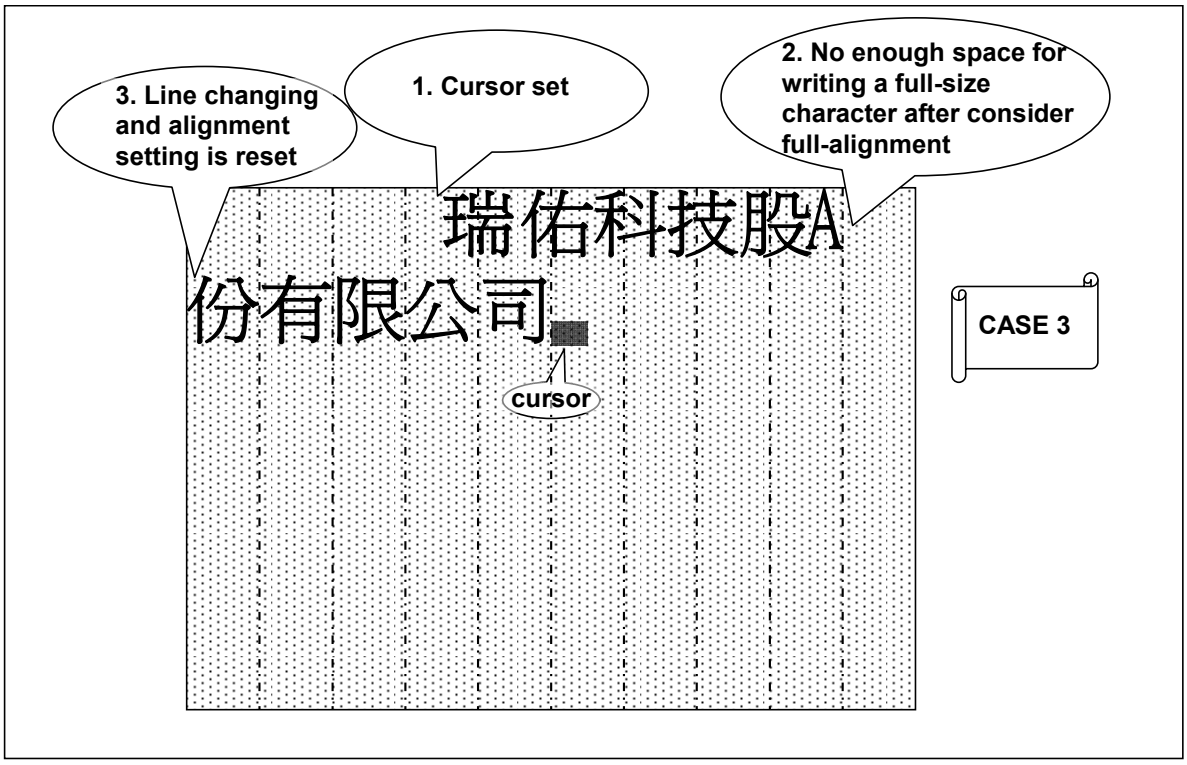


Figure 6-64 : Full-alignment Example(3)

6-13-3 Cursor Blinking

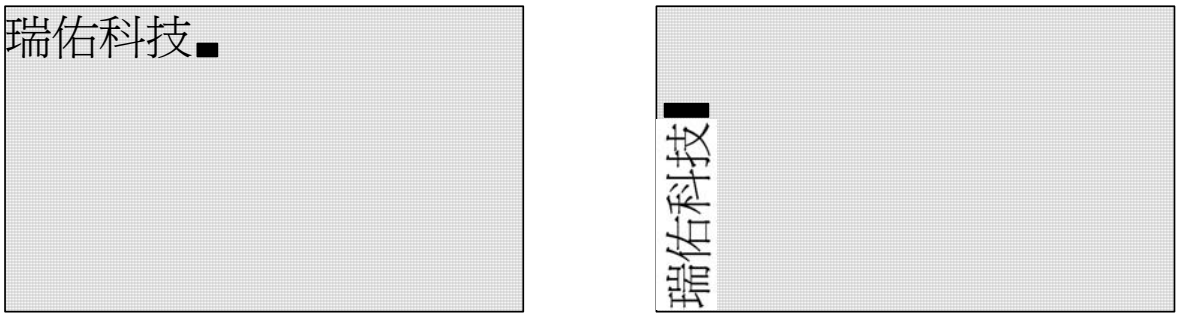
The user could control cursor On/Off or Blinking. The register [80h] BTMR is used to set up the blinking time.

- Blinking Time = BTMR[80h] Bit[7:0] x (1/Frame_Rate)

6-13-4 Cursor Width and Height

The cursor height is controlled by register CHWI Bit[7:4], and the height is setting from 1 ~ 16 pixel. It does depend on user's requirement.

In normal font, the cursor width fixed to one byte(8 pixels). And cursor's height is from 1~16pixels that depends on CHWI Bit[7:4]. In vertical font, the cursor height fixed to 16 pixels, and width is from 1~8 pixels that depends on CHWI Bit[6:4]. See Figure 6-65 and Figure 6-66 as following.



Cursor movement for Normal-Font　　Cursor movement for Vertical-Font

Figure 6-65 : Cursor Movement

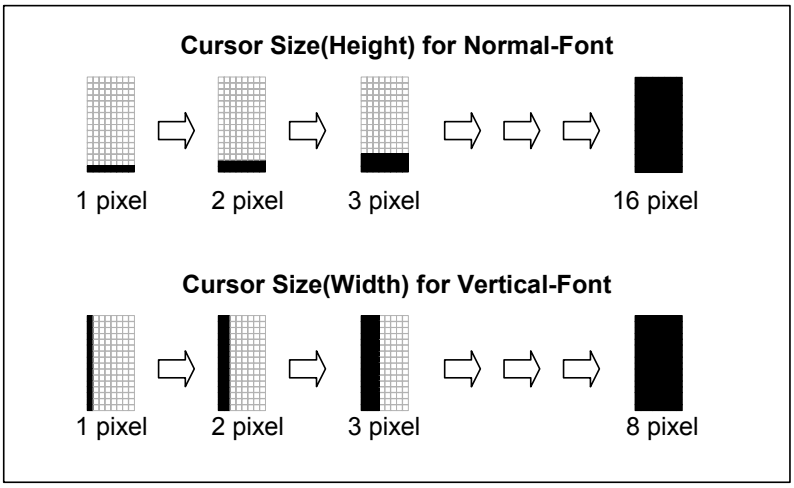


Figure 6-66 : Cursor Size

6-14 Extension Mode for Display

Normally the maximum support resolution of RA8806 is 320x240 dots. But RA8806 support a special display mode – Extension mode. And the maximum support resolution is 640x240 or 320x480 dots. In Extension mode, the dual DDRAM data can be show on the bigger panel. This mode is set by the Bit[6:4] of register MAMR. Please refer to Figure 6-68 of the example by using 640x240 panel, and Figure 6-69 by using 320x480 panel.

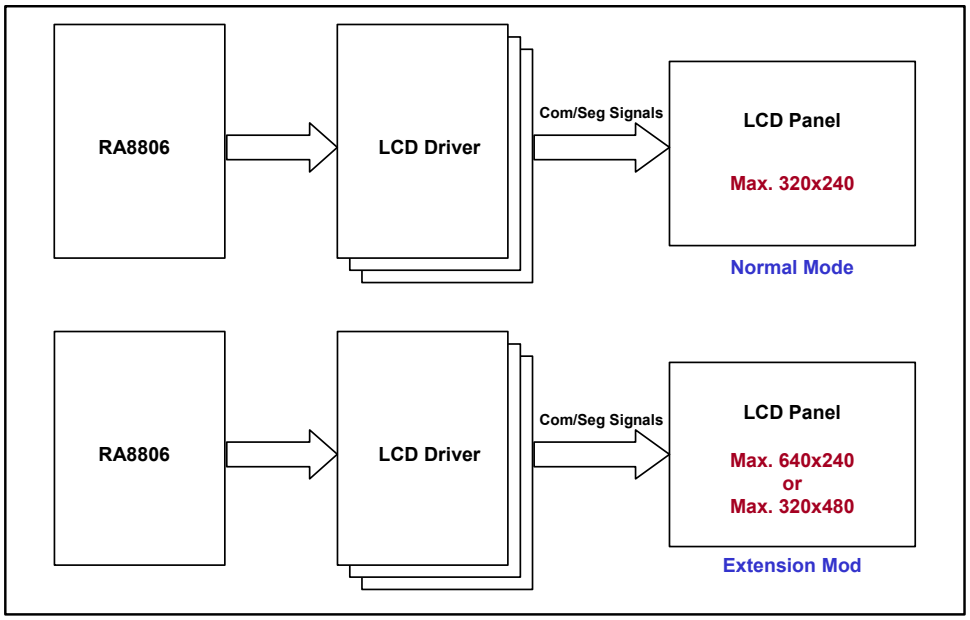


Figure 6-67 : Maximum Resolution for RA8806

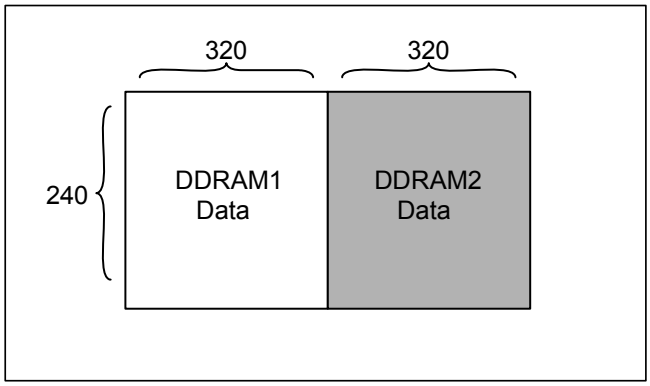


Figure 6-68 : Extension Mode(1) Register MAMR Bit[6:4] = 110h

If MAMR Bit[6:4] = 110b, then RA8806 supports maximum resolution is 640x240 pixels. The left side of screen shows the DDRAM1 data, the right side shows the DDRAM2 data. Please refer to Figure 6-68.

If MAMR Bit[6:4] = 111b, then RA8806 supports maximum resolution is 320x480 pixels. The up side of screen shows the DDRAM1 data, the down side shows the DDRAM2 data. Please refer to Figure 6-69.

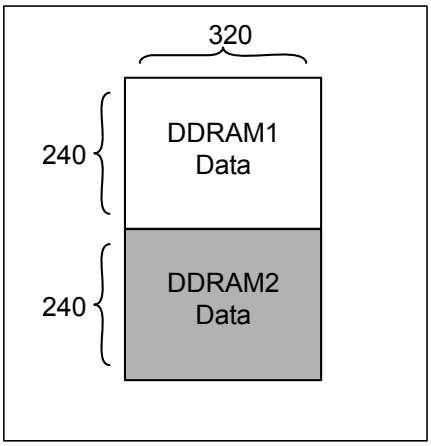


Figure 6-69 : Extension Mode(2) Register MAMR Bit[6:4] = 111h

Note that the extension mode is only display the combination of the DDRAM1 and DDRAM2 data. The cursor movement or characters display is not continuous across them. So user must set two pages separately on the screen. The RA8806 will combine the DDRAM1 and DDRAM2 data as a single display in extension mode. So, there is some limitation on Extension Mode. For example, if RA8806 set to Extension Mode(1) as Figure 6-68 for 640x240 dots panel. The data have to write into DDRAM1 and DDRAM2. But the cursor moving of Common is not from 0 to 639. The users have to separate the screen into 2 blocks and write data into DDRAM1 and DDRAM2 to create a complete 640x240 dots picture. The Figure 6-70 shows the description.

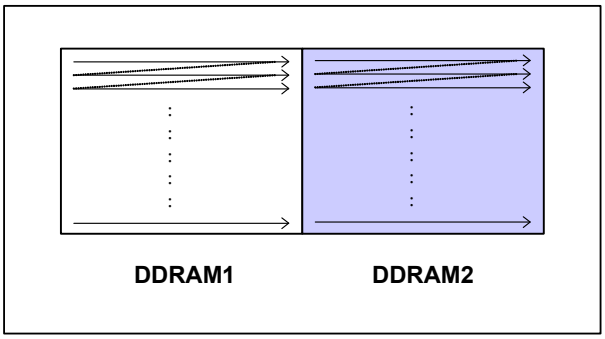


Figure 6-70：Extension Mode(1) Cursor Moving

Of course, if use Horizontal Scrolling, then the screen is like the following Figure 6-71.

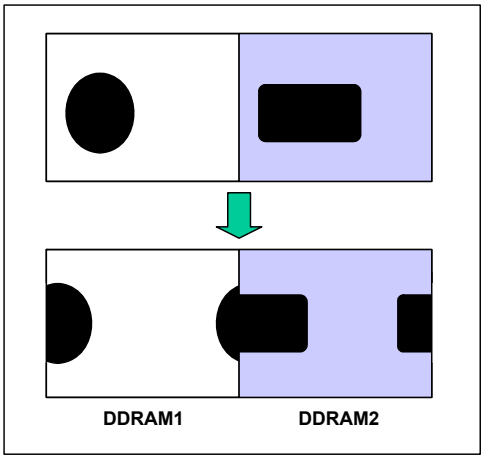


Figure 6-71：Extension Mode(1) Horizontal Scrolling

But if RA8806 use Extension mode(2) like Figure 6-69 for 320x480 dots panel, then the cursor moving is same as normal mode – the Common is from 0 to 319. The data scan of DDRAM is as Figure 6-72. The effect of Horizontal Scrolling is show as Figure 6-73.

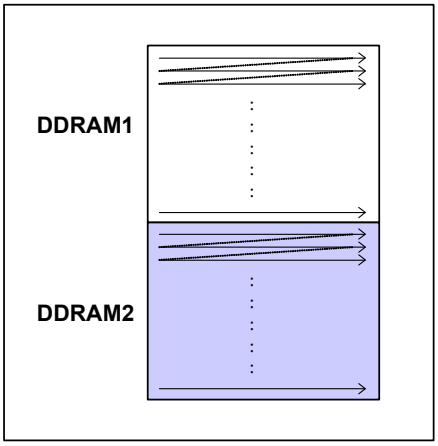


Figure 6-72：Extension Mode(2) Cursor Moving

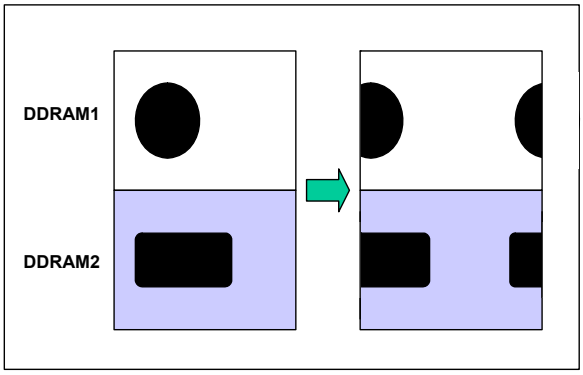


Figure 6-73：Extension Mode(2) Horizontal Scrolling

6-15 Eliminating Flicker Mode

RA8806 also provides “Eliminating Flicker Mode” in order to eliminate display errors at scan/MPU confliction. When the scan logic is processing scan task and at the same time the DDRAM is accessed by the MPU, the scan data will force to get a wrong one and may experience side effects such as “flicker”. When the flicker is too much, it will infect the display effect.

Therefore, RA8806 is designed to disable scan logic when MPU is accessing the DDRAM. And after it is completed, the scan logic active again. The “Eliminating Flicker Mode” separate Write and Read operating therefore no confliction will happen between scan logic and MPU cycle. That is, no data will lose. Ideally, it will have great improvement at display effect.

In real application, sometimes the cycle time that MPU writes data will be different(font write and memory clear), although it accesses the RA8806 periodly. On the other hand, the display performance is still limited when operating under the eliminating-flicker mode in some conditions. The limits are :

(1) The eliminating-flicker mode is suggested to operate under the graphic mode(bit-7 of register[01h] is setting 1).

(2) Please disable eliminating-flicker function before the memory clear mode. After the memory clear function is finished, then the eliminating-flicker function could be enable again.

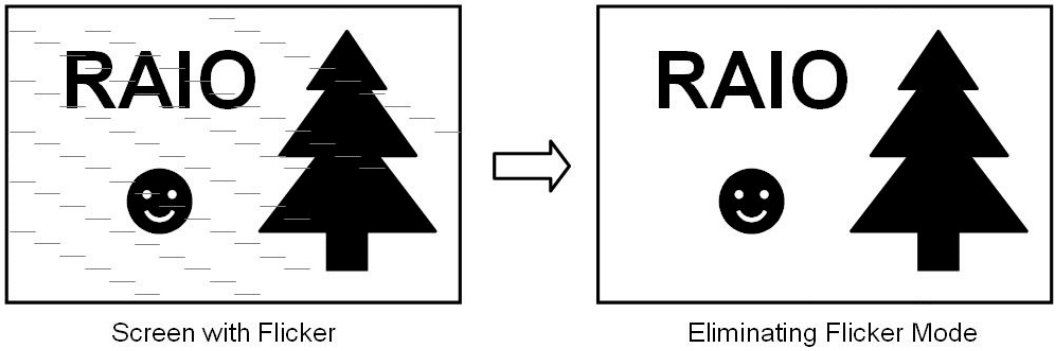


Figure 6-74 : Eliminating Flicker Mode

Table 6-38

Reg.	Bit_Num	Description	Reference
MISC	Bit 7	The LCD Driver-Scan will auto-pending when busy.	REG[01h]

Additionly, RA8806 also supply a method to eliminate flicker by check status. By check the status bit-6(SCAN BUSY), user can know whether can access memory that doesn’t conflict the scan data period. The application figure is below :

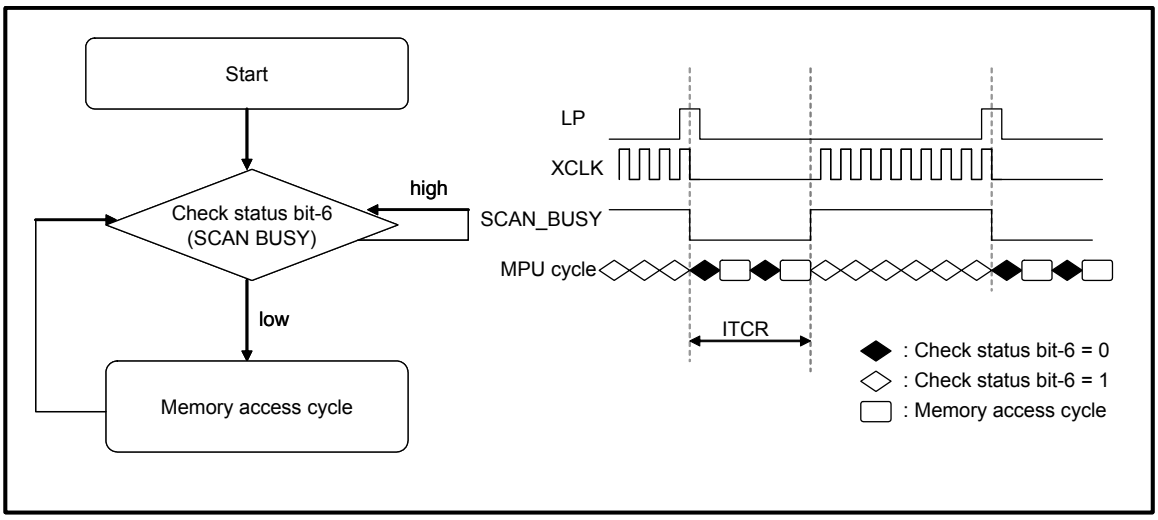


Figure 6-75 : Eliminate Flicker (1) Ideal

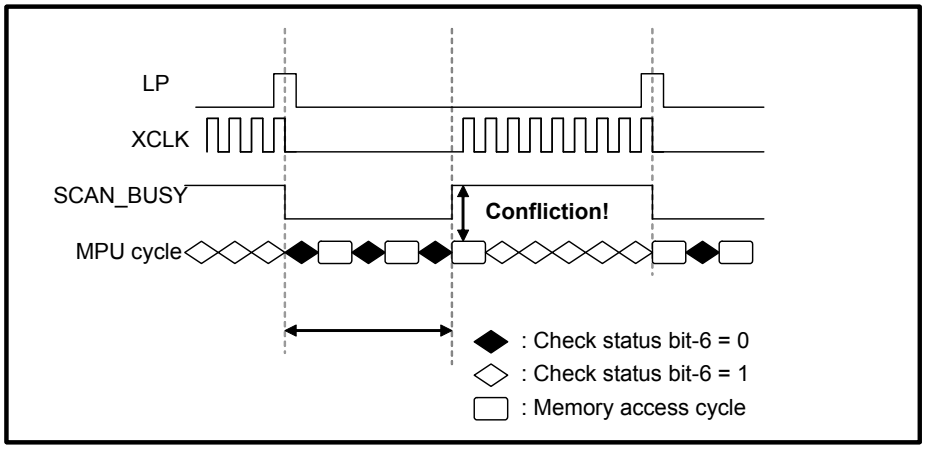


Figure 6-76 : Eliminate Flicker (1) Confliction

The ideal condition just likes the Figure 6-75. But if the condition likes Figure 6-76, the little part of confliction that we can not avoid. So if user wants to eliminate the flicker completely by checking SCAN_BUSY, user can use the method just like the Figure 6-77.

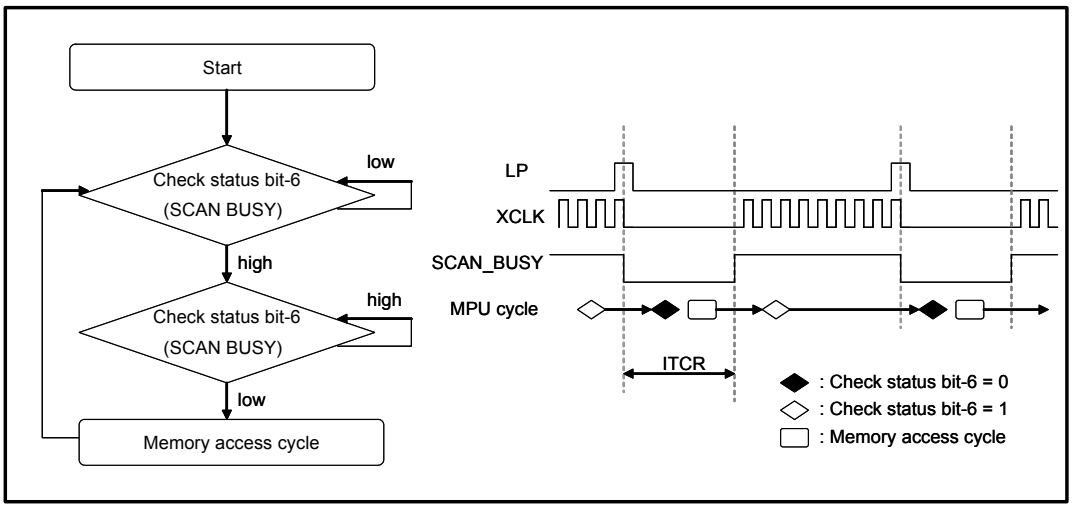


Figure 6-77 : Eliminate Flicker (2) Ideal

In order to improve the effect of eliminating flicker by check busy, we have two suggestions in real application :

MPU speed cannot be too slow(The MPU cycle period cannot be too long, just like Figure 6-78), otherwise the effect of eliminating flicker will be not eminent.

The ITCR value cannot be set too small.(just like Figure 6-79), otherwise the effect of eliminating flicker will be not eminent.

So if you want to eliminate flicker by check busy of this way, we support a suggestion ITCR setting value just like Figure 6-80 and Figure 6-81.

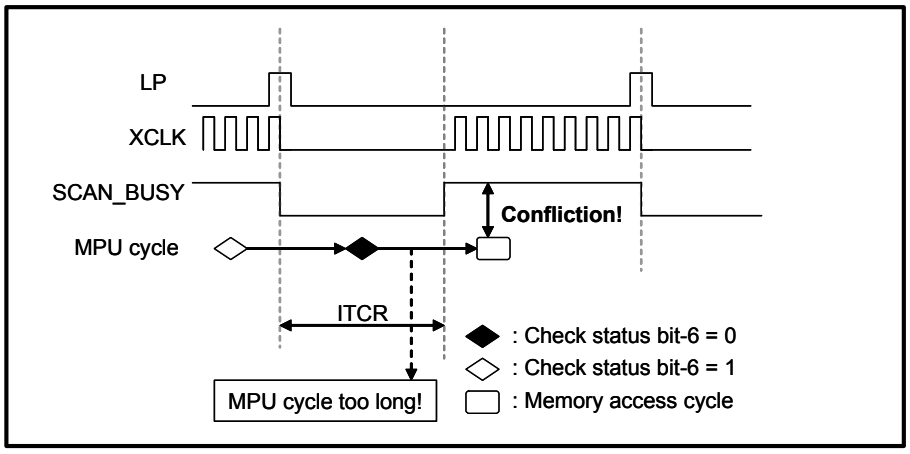


Figure 6-78 : Eliminate Flicker (2) MPU Cycle Too Long

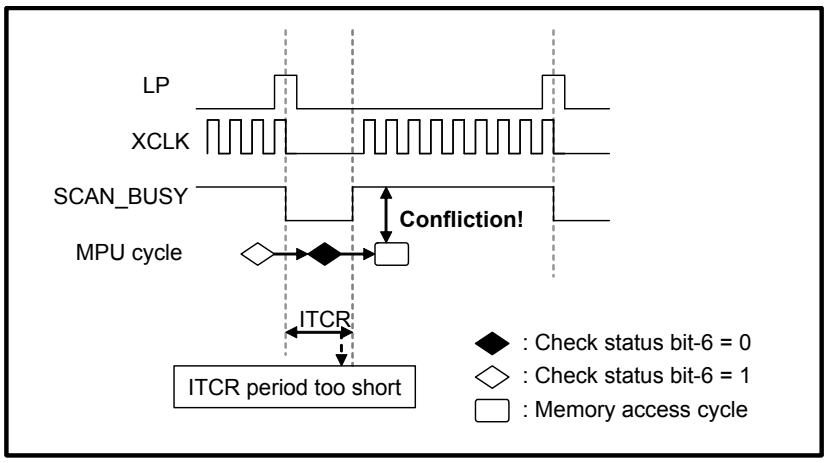


Figure 6-79 : Eliminate Flicker (2) ITCR Cycle Too Short

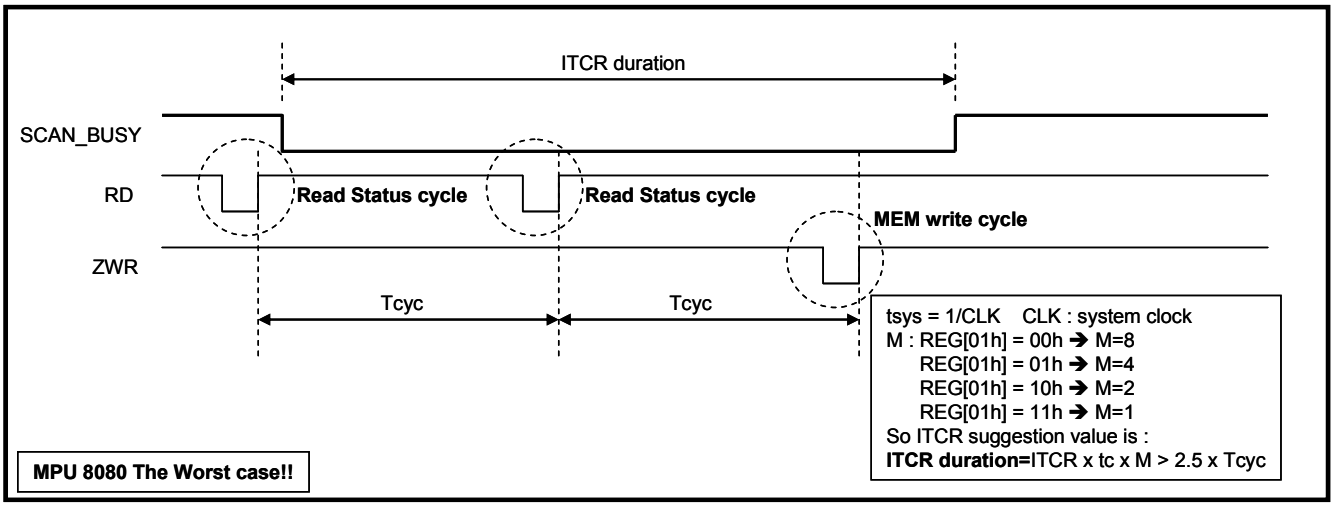


Figure 6-80 : The Suggestion ITCR value for MPU 8080 Interface to Eliminate Flicker

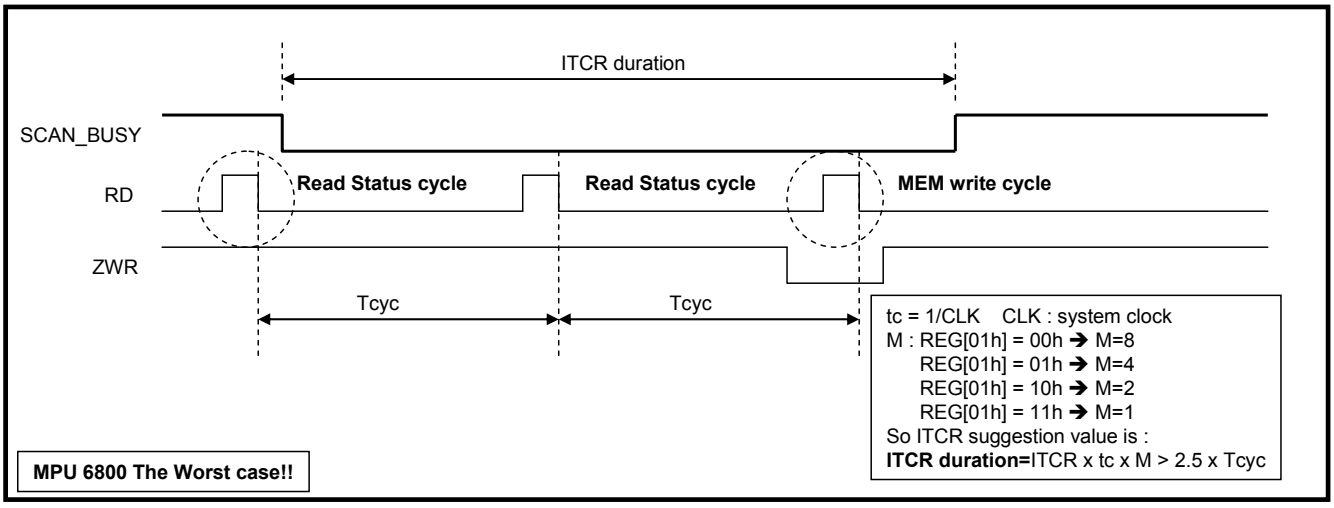


Figure 6-81 : The Suggestion ITCR value for MPU 6800 Interface to Eliminate Flicker

7. Package Information

7-1 Bonding Pad

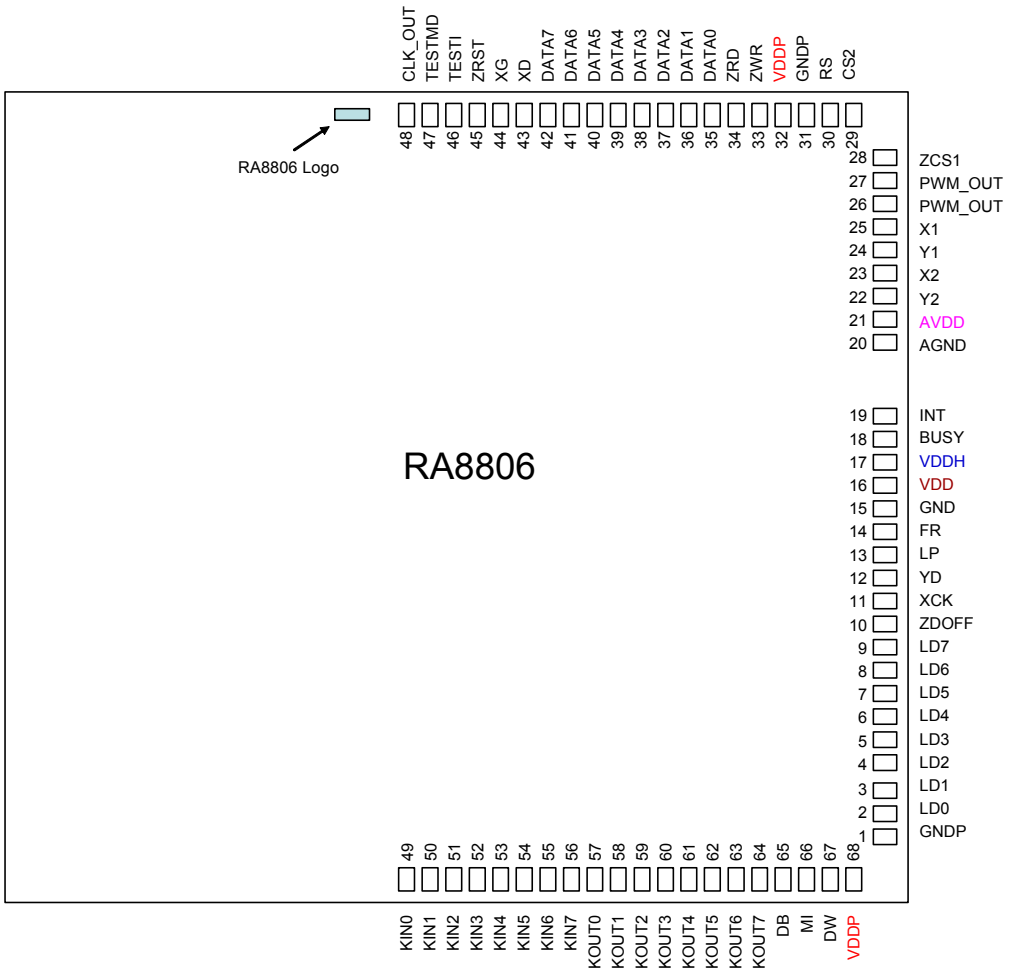


Figure 7-1 : RA8806 Bonding Pad

7-2 Pad X/Y Coordinate

Table 7-1 : RA8806 Pad Coordinate

Pad No.	Pad Name	X	Y	Pad No.	Pad Name	X	Y
1	GNDP	2301.05	-1649.70	35	DATA0	1429.80	1833.05
2	LD0	2301.05	-1536.80	36	DATA1	1314.80	1833.05
3	LD1	2301.05	-1421.80	37	DATA2	1199.80	1833.05
4	LD2	2301.05	-1306.80	38	DATA3	1084.80	1833.05
5	LD3	2301.05	-1191.80	39	DATA4	969.80	1833.05
6	LD4	2301.05	-1076.80	40	DATA5	854.80	1833.05
7	LD5	2301.05	-961.80	41	DATA6	739.80	1833.05
8	LD6	2301.05	-846.80	42	DATA7	624.80	1833.05
9	LD7	2301.05	-731.80	43	XD	509.80	1833.05
10	ZDOFF	2301.05	-616.80	44	XG	394.80	1833.05
11	XCK	2301.05	-501.80	45	ZRST	279.80	1833.05
12	YD	2301.05	-386.80	46	TESTI	164.80	1833.05
13	LP	2301.05	-271.80	47	TESTMD	49.80	1833.05
14	FR	2301.05	-156.80	48	CLK_OUT	-65.20	1833.05
15	GND	2301.05	-39.70	49	KIN0	-65.20	-1833.05
16	VDD	2301.05	75.30	50	KIN1	49.80	-1833.05
17	VDDH	2301.05	186.10	51	KIN2	164.80	-1833.05
18	BUSY	2301.05	303.20	52	KIN3	279.80	-1833.05
19	INT	2301.05	418.20	53	KIN4	394.80	-1833.05
20	AGND	2301.05	702.20	54	KIN5	509.80	-1833.05
21	AVDD	2301.05	817.20	55	KIN6	624.80	-1833.05
22	Y2	2301.05	934.30	56	KIN7	739.80	-1833.05
23	X2	2301.05	1049.30	57	KOUT0	854.80	-1833.05
24	Y1	2301.05	1164.30	58	KOUT1	969.80	-1833.05
25	X1	2301.05	1279.30	59	KOUT2	1084.80	-1833.05
26	PWM_OUT	2301.05	1394.30	60	KOUT3	1199.80	-1833.05
27	PWM_OUT	2301.05	1509.30	61	KOUT4	1314.80	-1833.05
28	ZCS1	2301.05	1624.30	62	KOUT5	1429.80	-1833.05
29	CS2	2119.80	1833.05	63	KOUT6	1544.80	-1833.05
30	RS	2004.80	1833.05	64	KOUT7	1659.80	-1833.05
31	GNDP	1887.70	1833.05	65	DB	1774.80	-1833.05
32	VDDP	1772.70	1833.05	66	MI	1889.80	-1833.05
33	ZWR	1659.80	1833.05	67	DW	2004.80	-1833.05
34	ZRD	1544.80	1833.05	68	VDDP	2121.90	-1833.05

7-3 Pin Assignment

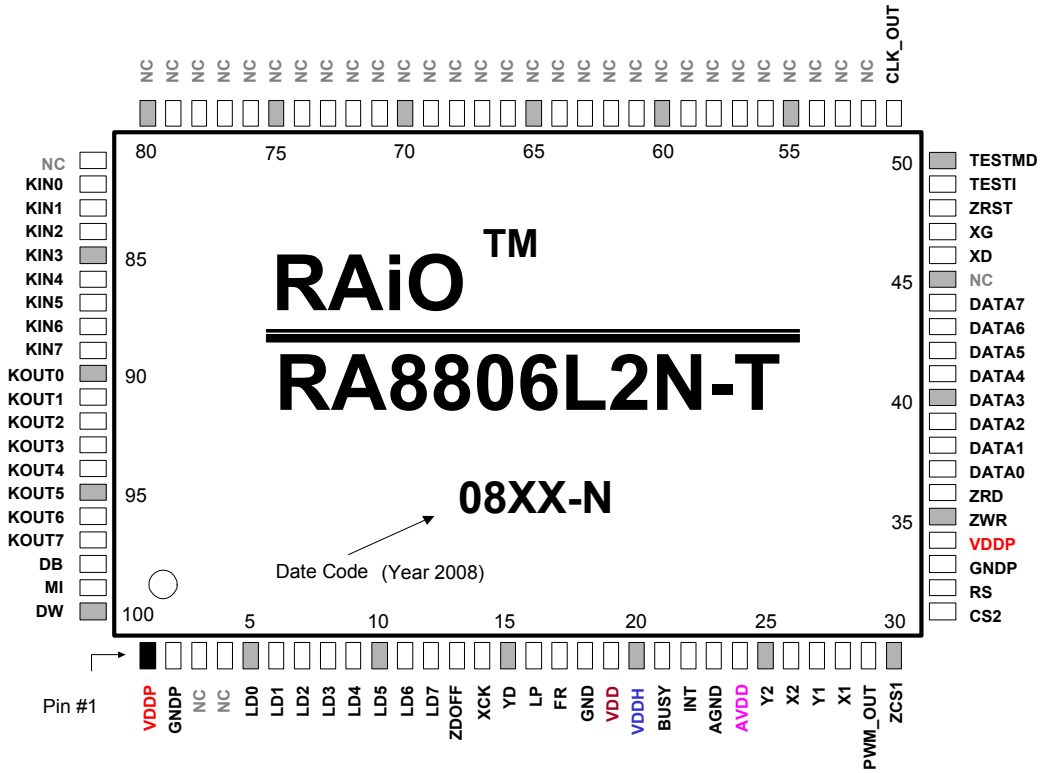


Figure 7-2 : LQFP-100Pin Pin Assignment

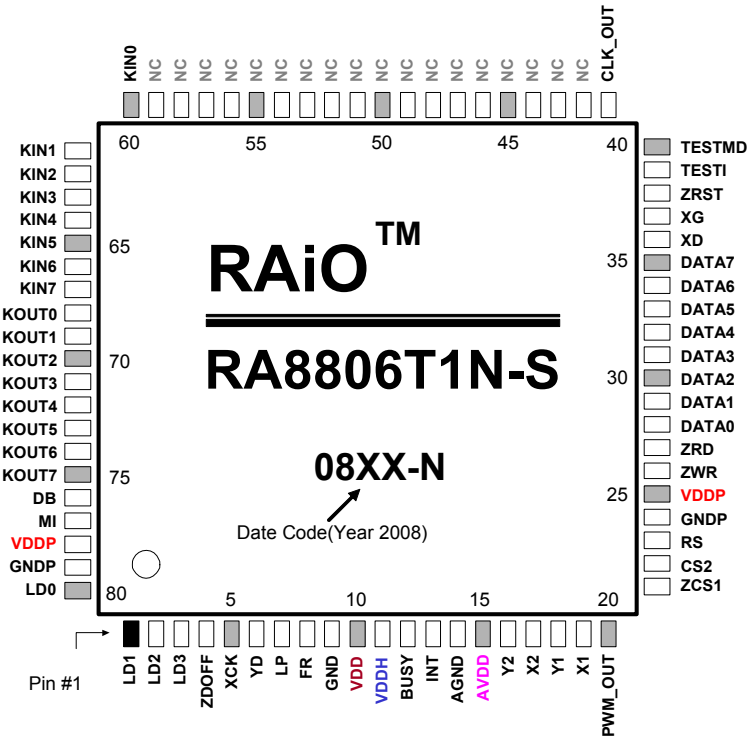


Figure 7-3 : TQFP-80Pin Pin Assignment

7-4 Package Dimension

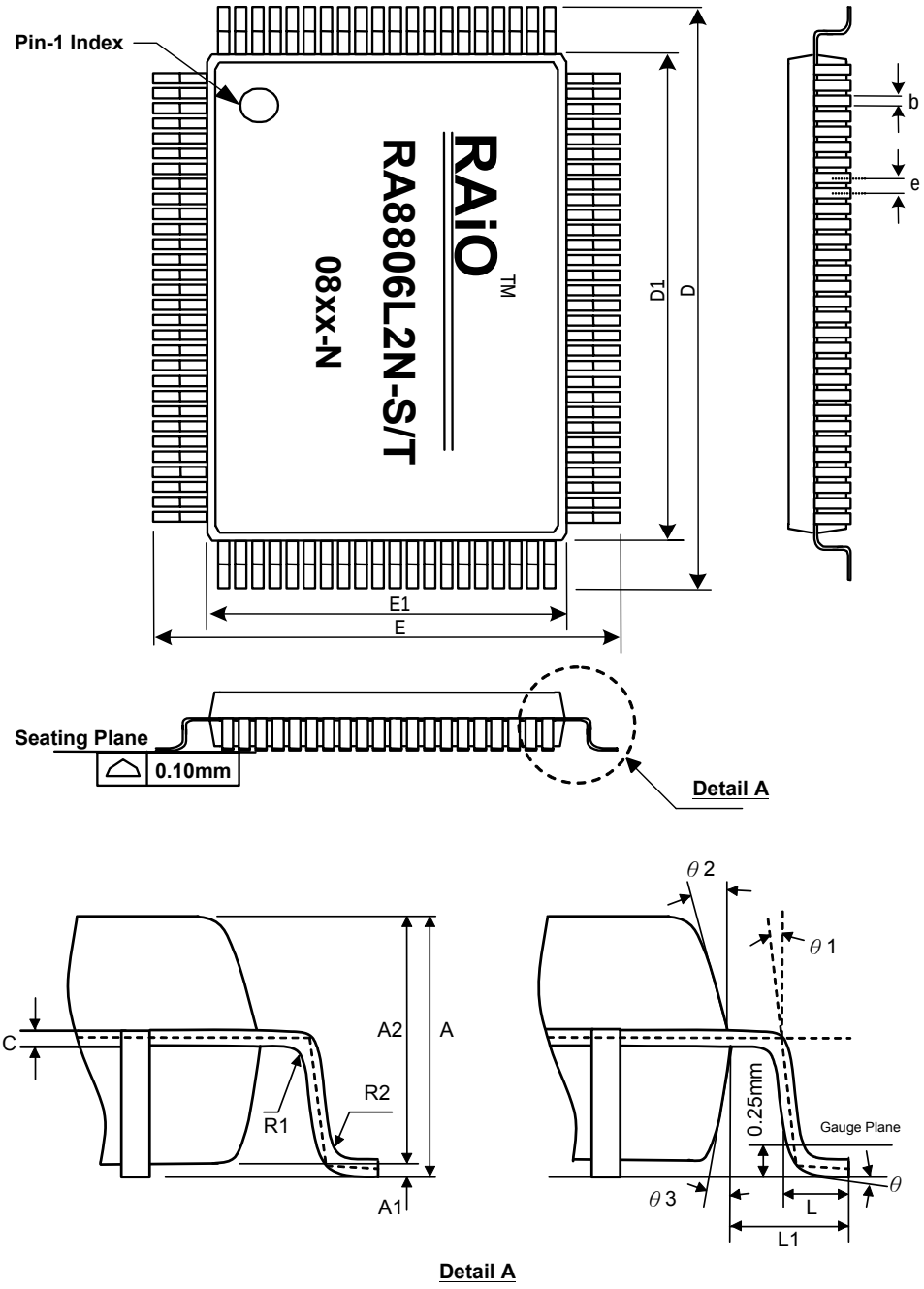


Figure 7-4 : LQFP-100Pin Mechanical

Table 7-2 : LQFP-100 Package Dimension

Symbols	Dimensions in Millimeters			Dimensions in Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	--	--	1.60	--	--	0.063
A1	0.05	--	0.15	0.002	--	0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
b	0.22	0.3	0.33	0.009	0.012	0.013
c	0.09	--	0.16	0.004	--	0.006
e	0.65 BSC.			0.026 BSC.		
D	22.00 BSC.			0.866 BSC.		
D1	20.00 BSC.			0.787 BSC.		
E	16.00 BSC.			0.630 BSC.		
E1	14.00 BSC.			0.551 BSC.		
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 Ref.			0.039 Ref.		
R1	0.08	--	--	0.003	--	--
R2	0.08	--	0.20	0.003	--	0.008
θ	0	3.5°	7°	0	3.5°	7°
θ1	0	--	--	0	--	--
θ2	11°	12°	13°	11°	12°	13°
θ3	11°	12°	13°	11°	12°	13°

Note:

Dimension “D1” and “E1” do not include mold protrusion. Allowable protrusion is 0.25mm per side. “D1” and “E1” are maximum plastic body size dimensions including mold mismatch.

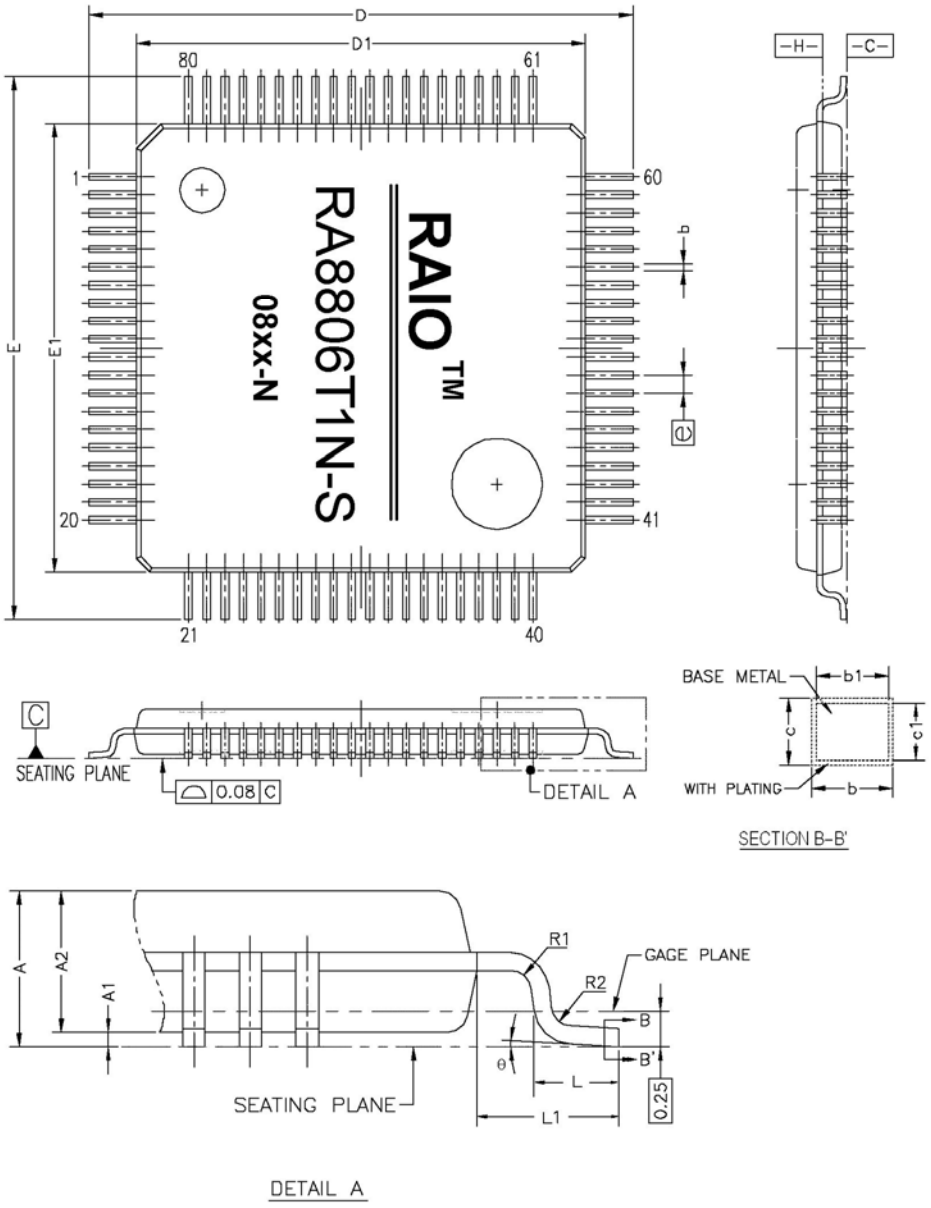


Figure 7-5 : TQFP-80Pin Mechanical

Table 7-3 : TQFP-80 Package Dimension

Symbols	Dimensions in Millimeters			Dimensions in Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A	--	--	1.20	--	--	0.047
A1	0.05	0.10	0.15	0.002	0.004	0.006
A2	0.95	1.00	1.05	0.037	0.039	0.041
b	0.13	0.18	0.23	0.005	0.007	0.009
b1	0.13	0.16	0.19	0.005	0.006	0.007
c	0.09	--	0.20	0.004	--	0.008
c1	0.09	0.127	0.16	0.004	0.005	0.006
e	0.40 BSC.			0.016 BSC.		
D	11.90	12.00	12.10	0.469	0.472	0.476
D1	9.90	10.00	10.10	0.390	0.394	0.398
E	11.90	12.00	12.10	0.469	0.472	0.476
E1	9.90	10.00	10.10	0.390	0.394	0.398
L	0.45	0.60	0.75	0.018	0.024	0.030
L1	1.00 Ref.			0.040 Ref.		
R1	0.08	--	--	0.003	--	--
R2	0.08	--	0.20	0.003	--	0.008
θ	0	3°	7°	0	3°	7°

7-5 Part Number

Table 7-4 : Part Number

Product Name (Full Name)	Resolution (Max)	Package	Font ROM	ASCII ROM	RoHs Compliance
RA8806L2N-T	320x240 (Note 1)	LQFP-100 (20x14)	Traditional Chinese (Note 2)	ISO-8859-1 ~ 4	Yes
RA8806L2N-S			Simple Chinese (Note 2)	ISO-8859-1 ~ 4	Yes
RA8806L2N-J			Japanese Kanji	ISO-8859-1 ~ 4	Yes
RA8806T1N-T		TQFP-80 (10x10)	Traditional Chinese (Note 2, 3)	ISO-8859-1 ~ 4	Yes
RA8806T1N-S			Simple Chinese (Note 2, 3)	ISO-8859-1 ~ 4	Yes
RA8806T1N-J			Japanese Kanji (Note 3)	ISO-8859-1 ~ 4	Yes
RA8806-T		Die	Traditional Chinese	ISO-8859-1 ~ 4	Yes
RA8806-S			Simple Chinese	ISO-8859-1 ~ 4	Yes

Notes:

1. In Extension Mode, the maximum resolution is 640x240 or 320x480. See Section 6-14 “Extension Mode and Display”.
2. In both Traditional and Simple Chinese font, it built-in 52 basic Japanese font.
3. LCD driver data bus of RA8806T1N is 4-bits.
4. All of the parts of RA008 are RoHS compliance and pass the detection of free PFOS and PFOA.

Table 7-5 : RA8806L2N vs. RA8806T1N

Difference	RA8806L2N	RA8806T1N
Package	LQFP-100Pins 20mm x 14mm	TQFP-80Pins 10mm x 10mm
LCD Data Bus	4-bits or 8-bits	4-bits

8. Electrical Characteristic

8-1 Absolute Maximum Ratings

Table 8-1 : Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Supply Voltage Range	V_{DD}	-0.3 to 6.5	V
Input Voltage Range	V_{IN}	-0.3 to V_{DD}+0.3	V
Power Dissipation (VDD = 5V)	P_D	≦30	mW
Operation Temperature Range	T_{OPR}	-30 to +85	℃
Storage Temperature	T_{ST}	-45 to +125	℃
Soldering temperature (10 seconds). See Note 1.	T_{SOLDER}	260	℃

Notes:

1. The humidity resistance of the flat package may be reduced if the package is immersed in solder. Use a soldering technique that does not heat stress the package.
2. If the power supply has a high impedance, a large voltage differential can occur between the input and supply voltages. Take appropriate care with the power supply and the layout of the supply lines.
3. All supply voltages are referenced to Gnd = 0V.

8-2 DC Characteristic

Table 8-2 : DC Characteristic

Parameter	Symbol	Min.	Typ.	Max.	Unit	Condition
Operating Voltage(1)	V_{DDP} / V_{DDH}	4.5	5.0	5.5	V	V_{DDP} = V_{DDH} (Refer to Figure 6-31)
Operating Voltage(2)	V_{DDP} / V_{DD}	2.4	3.3	3.6	V	V_{DDP} = V_{DD} V_{DDH} Open (Refer to Figure 6-30)
Oscillator frequency	F_{OSC}	4	8	12	MHz	V_{DD} = 5V
External clock frequency	F_{CLK}	4	8	12	MHz	V_{DD} = 5V
DC to DC Output Voltage	V_{DD}	2.8V	3.0	3.3	V	Add external 1uF Capacitor
Input						
Input High Voltage	V_{IH}	0.8xV_{DD}	--	V_{DD}	V	See Note 1, 3
Input Low Voltage	V_{IL}	Gnd	--	0.2xV_{DD}	V	See Note 1, 3
Output						
Output High Voltage	V_{OH}	V_{DD}-0.4	--	V_{DD}	V	See Note 2, 3
Output Low Voltage	V_{OL}	Gnd	--	V_{DD}+0.4	V	See Note 2, 3
Schmitt-trigger						
Output High Voltage	V_{OH}	0.5xV_{DD}	0.7xV_{DD}	0.8xV_{DD}	V	See Note 4
Output Low Voltage	V_{OL}	0.2xV_{DD}	0.3xV_{DD}	0.5xV_{DD}	V	See Note 4
Input Leakage Current 1	I_{IH}	--	--	+1	μA	
Input Leakage Current 2	I_{IL}	--	--	-1	μA	
Operation Current	I_{OPR}	1	5	10	mA	
Standby Mode Current	I_{SB}	--	1.5	1.8	mA	Case1
(Normal Mode Current)			1.8	2.1	mA	Case2
Display Off Current	$I_{DISPLAY}$	--	120	140	μA	Case1
			140	160	μA	Case2
Sleep Mode	I_{SLP}	--	0.5	1	μA	Case1
		--	20	25	μA	Case2

Notes:
1. ZCS1, CS2, ZWR, ZRD, RS, MI, DW, DB, KIN[7:0], TESTMD and TESTI are inputs. KIN[7:0] built-in pull up resistors. The TESTMD and TESTI built-in pull down resistors.
2. INT, BUSY, CLK_OUT, PWM_OUT, KOUT[7:0], LP, FR, YD, ZDOFF, XCK and LD[7:0] are outputs.
3. DATA[7:0] are Bi-direction.
4. The ZRST are Schmitt-trigger with pull-up input. The pulse width on ZRST must be at least 1024*tc. Note that pulses of more than a few seconds will cause DC voltages to be applied to the LCD panel.

Case1: V_{DDP} = V_{DD} = A_{VDD} = 3.3V, V_{DDH} = NC, LCD Driver VDD = 5V, CLK = 4MHz, CLK_OUT: Off, Segment=160, Common=160, FRM = 78Hz, T_A=25℃.
Case2: V_{DDP} = V_{DDH} = 5V, V_{DD} = A_{VDD} = 3V, LCD Driver VDD = 3.3V, CLK = 4MHz, CLK_OUT: Off, Segment=160, Common=160, FRM = 78Hz, T_A=25℃.

8-3 Timing Characteristic Wavaform

Figure 8-1 is timing characteristic waveform and parameters for RA8806 driver interface signal.

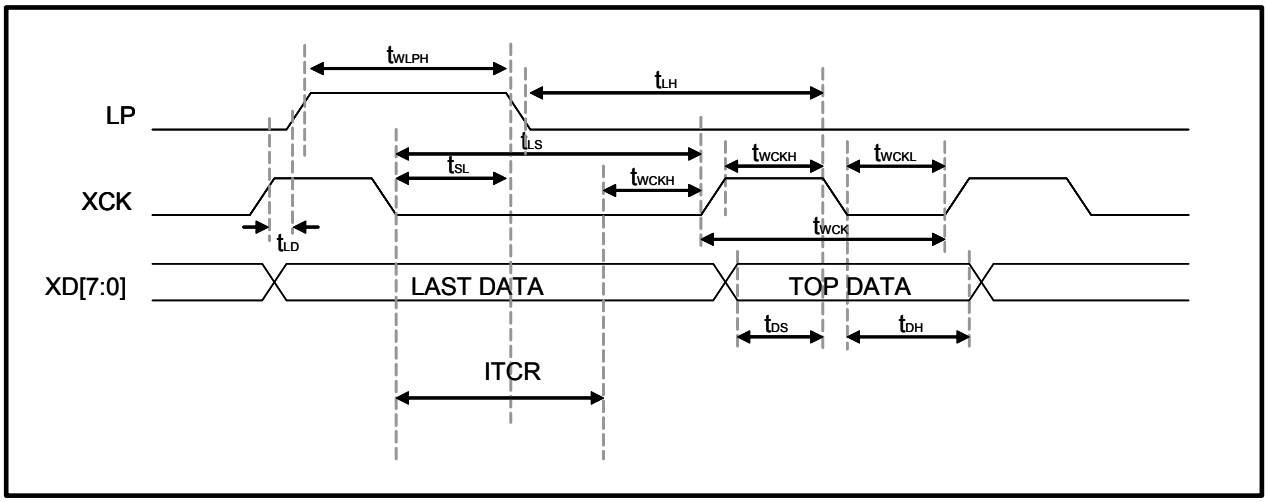


Figure 8-1 : Timing Characteristic Waveform

Table 8-3: Parameters

Parameter	Symbol	condition	Min.	Typ.	Max.	Unit	Note
Shift Clock period	t_{WCK}			T_{sys}/D		ns	*1
Shift Clock "H" Pulse Width	t_{WCKH}		t_{WCK}/2 -10		t_{WCK}/2 +10	ns	
Shift Clock "L" Pulse Width	t_{WCKL}		t_{WCK}/2 -10		t_{WCK}/2 +10	ns	
Data Setup Time	t_{DS}		t_{WCK}/2 -30		t_{WCK}/2	ns	
Data Hold Time	t_{DH}		t_{WCK}/2		t_{WCK}/2 +30	ns	
Latch Pulse "H" Pulse Width	t_{WLPH}		t_{WCK}-10		t_{WCK}+10	ns	
Shift Clock Rise to Latch Pulse Rise Time	t_{LD}		0			ns	
Shift Clock Fall to Latch Pulse Fall Time	t_{SL}		t_{WCK}/2 -10		t_{WCK}/2 +10	ns	
Latch Clock Rise to Shift Pulse Rise Time	t_{LS}		t_{WCK}/2 -10			ns	*2
Latch Clock Rise to Shift Pulse Rise Time	t_{LH}		t_{WCK}/2 -10			ns	*2

Note :

1. Tsys : system clock period(i.e. System clock = 12MHz, Tsys = 83.3ns)

   D : Driver clock selection(REG[01h] B3-2)
   - 0 0 : XCK = CLK/8
   - 0 1 : XCK = CLK/4（initial value）
   - 1 0 : XCK = CLK/2
   - 1 1 : XCK = CLK

2.The period also depends on the setting of register ITCR.

Appendix A . Application Circuit

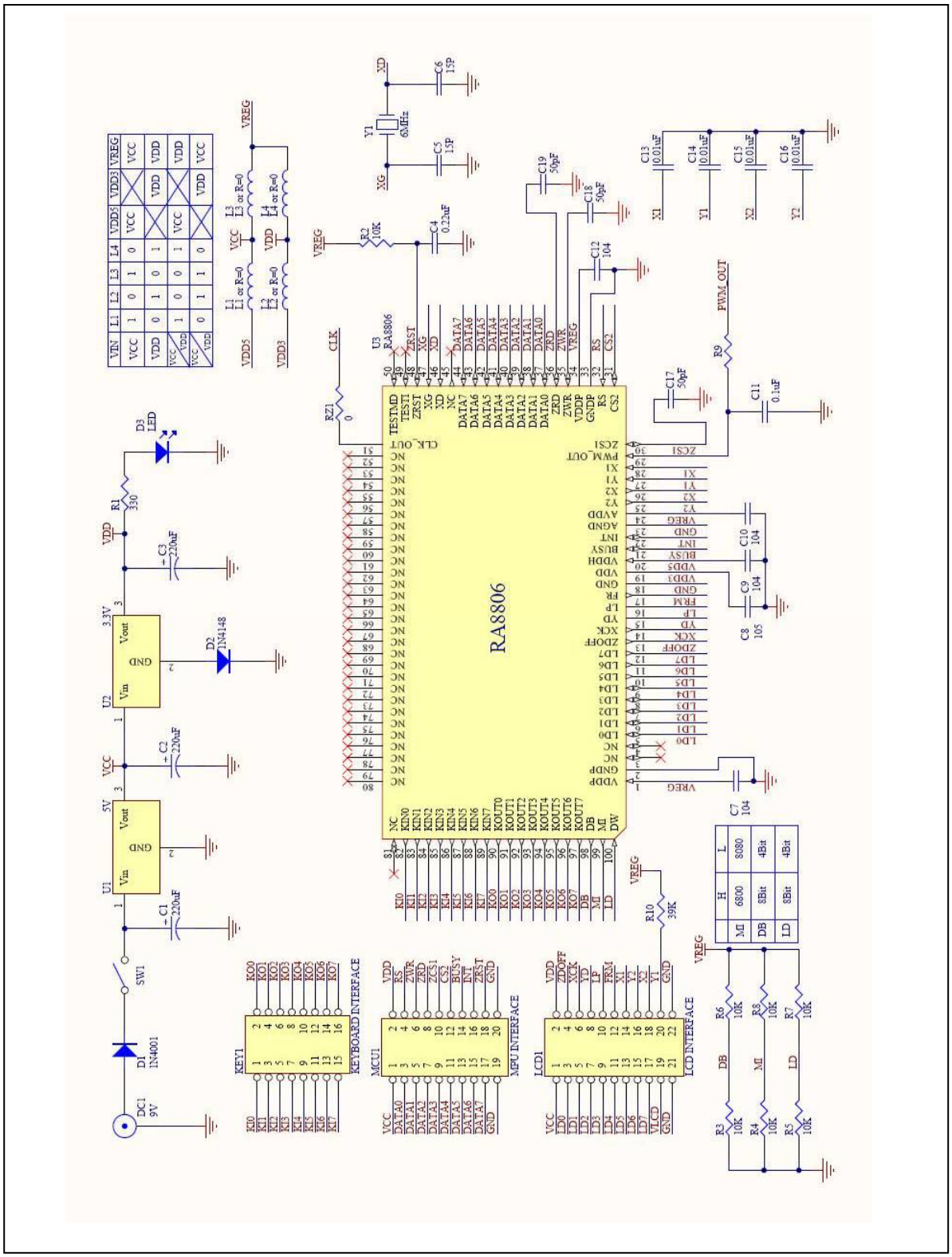


Figure A-1 : Application Circuit for 3V or 5V

Appendix B . Frame Rate Table

Table B-1 : Frame Rate Table(1)

Seg	Com	CLK (MHz)	XCK=CLK/2 REG[01h] Bit[3:2] = 10		XCK=CLK/4 REG[01h] Bit[3:2] = 01		XCK=CLK/8 REG[01h] Bit[3:2] = 00	
			Frame Rate (Hz)	REG[90h] ITCR	Frame Rate (Hz)	REG[90h] ITCR	Frame Rate (Hz)	REG[90h] ITCR
320	240	4	55	72	55	--	55	--
320	240	4	60	59	60	--	60	--
320	240	4	65	48	65	--	65	--
320	240	4	70	39	70	--	70	--
320	240	4	75	31	75	--	75	--
320	240	6	55	147	55	34	55	--
320	240	6	60	128	60	24	60	--
320	240	6	65	112	65	16	65	--
320	240	6	70	99	70	9	70	--
320	240	6	75	87	75	3	75	--
320	240	8	55	223	55	72	55	--
320	240	8	60	198	60	59	60	--
320	240	8	65	176	65	48	65	--
320	240	8	70	158	70	39	70	--
320	240	8	75	142	75	31	75	--
320	240	10	55	--	55	109	55	15
320	240	10	60	--	60	94	60	7
320	240	10	65	241	65	80	65	--
320	240	10	70	218	70	69	70	--
320	240	10	75	198	75	59	75	--
320	240	12	55	--	55	147	55	34
320	240	12	60	--	60	128	60	24
320	240	12	65	--	65	112	65	16
320	240	12	70	--	70	99	70	9
320	240	12	75	253	75	87	75	3
240	160	4	55	167	55	54	55	--
240	160	4	60	148	60	44	60	--
240	160	4	65	132	65	36	65	--
240	160	4	70	119	70	29	70	--
240	160	4	75	107	75	23	75	--
240	160	6	55	--	55	110	55	25
240	160	6	60	--	60	96	60	18
240	160	6	65	228	65	84	65	12
240	160	6	70	208	70	74	70	7
240	160	6	75	190	75	65	75	3
240	160	8	55	--	55	167	55	54
240	160	8	60	--	60	148	60	44
240	160	8	65	--	65	132	65	36
240	160	8	70	--	70	119	70	29
240	160	8	75	--	75	107	75	23
240	160	10	55	--	55	224	55	82
240	160	10	60	--	60	200	60	70
240	160	10	65	--	65	180	65	60
240	160	10	70	--	70	163	70	52
240	160	10	75	--	75	148	75	44
240	160	12	55	--	55	--	55	110
240	160	12	60	--	60	253	60	96
240	160	12	65	--	65	228	65	84
240	160	12	70	--	70	208	70	74
240	160	12	75	--	75	190	75	65

Table B-2 : Frame Rate Table(2)

Seg	Com	CLK (MHz)	XCK=CLK/2 REG[01h] Bit[3:2] = 10		XCK=CLK/4 REG[01h] Bit[3:2] = 01		XCK=CLK/8 REG[01h] Bit[3:2] = 00	
			Frame Rate (Hz)	REG[90h] ITCR	Frame Rate (Hz)	ITCR	Frame Rate (Hz)	REG[90h] ITCR
160	160	4	55	187	55	74	55	17
160	160	4	60	168	60	64	60	12
160	160	4	65	152	65	56	65	8
160	160	4	70	139	70	49	70	5
160	160	4	75	127	75	43	75	2
160	160	6	55	--	55	130	55	45
160	160	6	60	--	60	116	60	38
160	160	6	65	248	65	104	65	32
160	160	6	70	228	70	94	70	27
160	160	6	75	210	75	85	75	23
160	160	8	55	--	55	187	55	74
160	160	8	60	--	60	168	60	64
160	160	8	65	--	65	152	65	56
160	160	8	70	--	70	139	70	49
160	160	8	75	--	75	127	75	43
160	160	10	55	--	55	244	55	102
160	160	10	60	--	60	220	60	90
160	160	10	65	--	65	200	65	80
160	160	10	70	--	70	183	70	72
160	160	10	75	--	75	168	75	64
160	160	12	55	--	55	--	55	130
160	160	12	60	--	60	--	60	116
160	160	12	65	--	65	248	65	104
160	160	12	70	--	70	228	70	94
160	160	12	75	--	75	210	75	85
160	128	4	55	244	55	102	55	31
160	128	4	60	220	60	90	60	25
160	128	4	65	200	65	80	65	20
160	128	4	70	183	70	72	70	16
160	128	4	75	168	75	64	75	12
160	128	6	55	--	55	173	55	67
160	128	6	60	--	60	155	60	58
160	128	6	65	--	65	140	65	50
160	128	6	70	--	70	127	70	44
160	128	6	75	--	75	116	75	38
160	128	8	55	--	55	244	55	102
160	128	8	60	--	60	220	60	90
160	128	8	65	--	65	200	65	80
160	128	8	70	--	70	183	70	72
160	128	8	75	--	75	168	75	64
160	128	10	55	--	55	--	55	138
160	128	10	60	--	60	--	60	123
160	128	10	65	--	65	--	65	110
160	128	10	70	--	70	239	70	100
160	128	10	75	--	75	220	75	90
160	128	12	55	--	55	--	55	173
160	128	12	60	--	60	--	60	155
160	128	12	65	--	65	--	65	140
160	128	12	70	--	70	--	70	127
160	128	12	75	--	75	--	75	116

Table B-3 : Frame Rate Table(3)

Seg	Com	CLK (MHz)	XCK=CLK/2 REG[01h] Bit[3:2] = 10		XCK=CLK/4 REG[01h] Bit[3:2] = 01		XCK=CLK/8 REG[01h] Bit[3:2] = 00	
			Frame Rate (Hz)	REG[90h] ITCR	Frame Rate (Hz)	REG[90h] ITCR	Frame Rate (Hz)	REG[90h] ITCR
240	128	4	55	224	55	82	55	11
240	128	4	60	200	60	70	60	5
240	128	4	65	180	65	60	65	--
240	128	4	70	163	70	52	70	--
240	128	4	75	148	75	44	75	--
240	128	6	55	--	55	153	55	47
240	128	6	60	--	60	135	60	38
240	128	6	65	--	65	120	65	30
240	128	6	70	--	70	107	70	24
240	128	6	75	253	75	96	75	18
240	128	8	55	--	55	224	55	82
240	128	8	60	--	60	200	60	70
240	128	8	65	--	65	180	65	60
240	128	8	70	--	70	163	70	52
240	128	8	75	--	75	148	75	44
240	128	10	55	--	55	--	55	118
240	128	10	60	--	60	--	60	103
240	128	10	65	--	65	240	65	90
240	128	10	70	--	70	219	70	80
240	128	10	75	--	75	200	75	70
240	128	12	55	--	55	--	55	153
240	128	12	60	--	60	--	60	135
240	128	12	65	--	65	--	65	120
240	128	12	70	--	70	--	70	107
240	128	12	75	--	75	253	75	96
240	64	4	55	--	55	224	55	82
240	64	4	60	--	60	200	60	70
240	64	4	65	--	65	180	65	60
240	64	4	70	--	70	163	70	52
240	64	4	75	--	75	148	75	44
240	64	6	55	--	55	--	55	153
240	64	6	60	--	60	--	60	135
240	64	6	65	--	65	--	65	120
240	64	6	70	--	70	--	70	107
240	64	6	75	--	75	253	75	96
240	64	8	55	--	55	--	55	224
240	64	8	60	--	60	--	60	200
240	64	8	65	--	65	--	65	180
240	64	8	70	--	70	--	70	163
240	64	8	75	--	75	--	75	148
240	64	10	55	--	55	--	55	--
240	64	10	60	--	60	--	60	--
240	64	10	65	--	65	--	65	240
240	64	10	70	--	70	--	70	219
240	64	10	75	--	75	--	75	200
240	64	12	55	--	55	--	55	--
240	64	12	60	--	60	--	60	--
240	64	12	65	--	65	--	65	--
240	64	12	70	--	70	--	70	--
240	64	12	75	--	75	--	75	253

Note: The value of ITCR is in decimal.

Appendix C . Font Table - ASCII

When Bit-7 of register “FNCR” is “0”, the contents of ASCII Block Tables 1-4 are show as following Table C-1~ Table C-3.

Table C-1 : ASCII Block 1

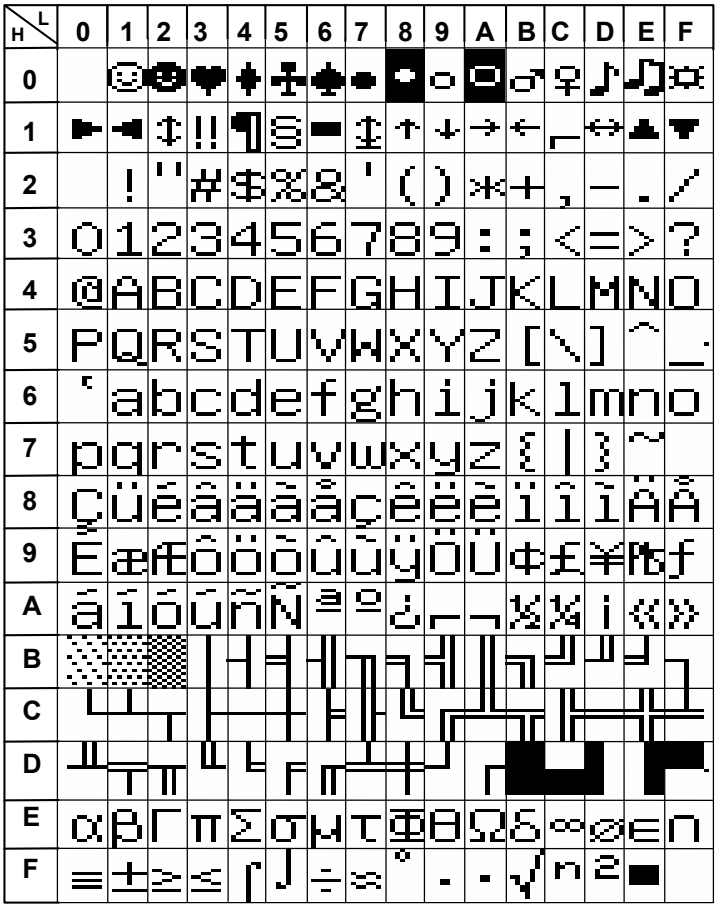


Table C-2 : ASCII Block 2

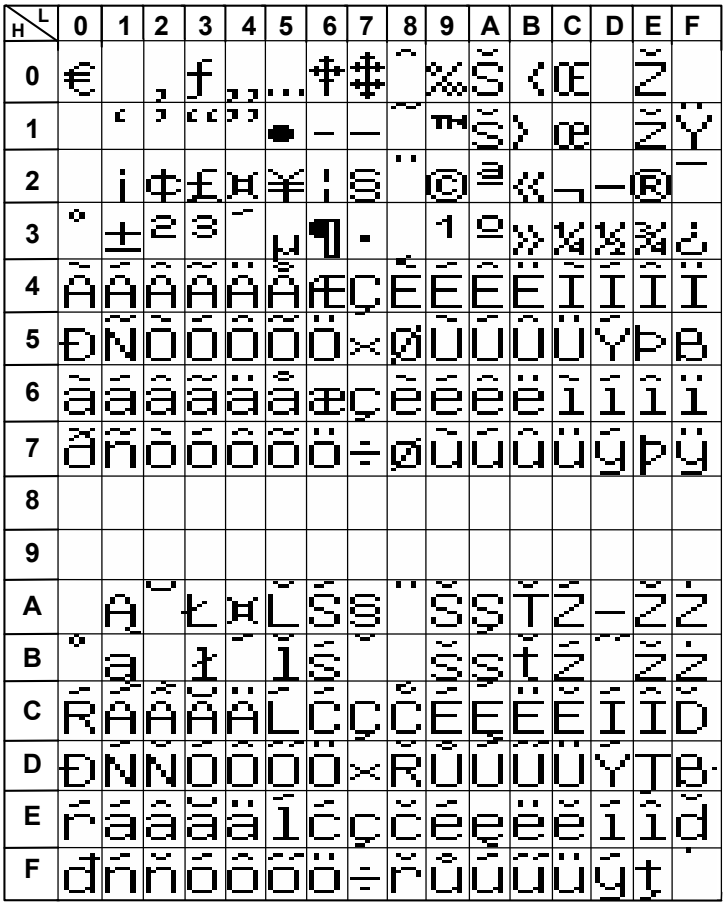


Table C-3 : ASCII Block 3

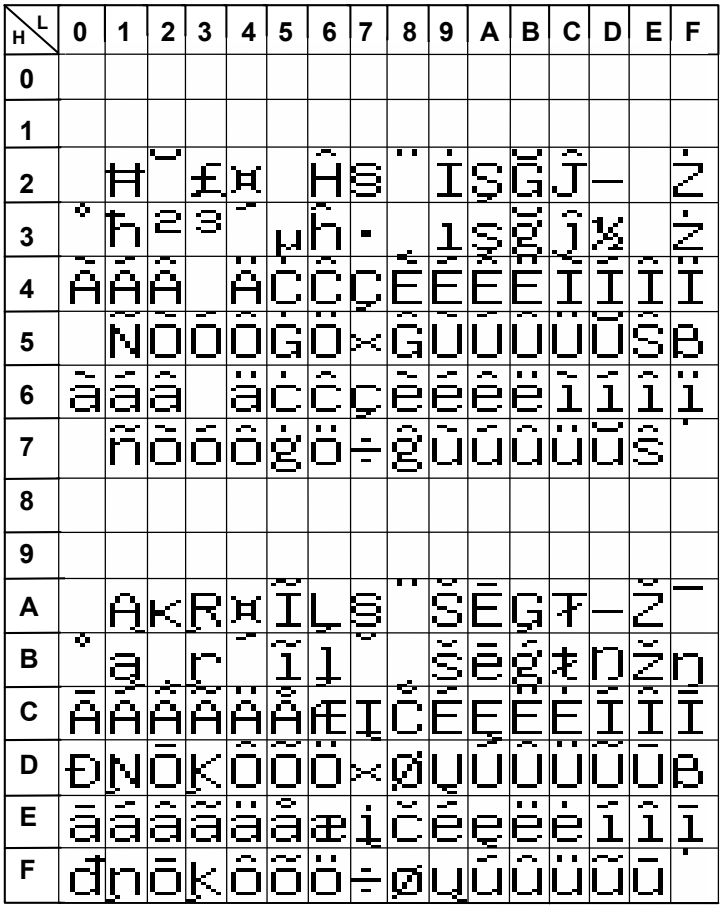
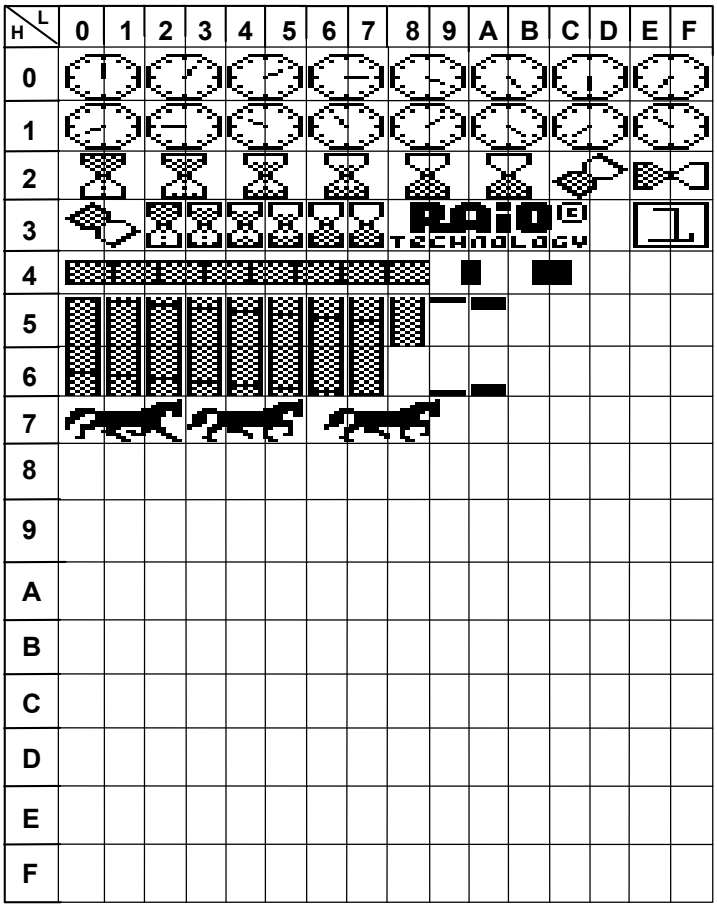


Table C-4 : ASCII Block 4



When Bit-7 of register “FNCR” is “1”, the contents of ASCII Block Tables 1-4 are show as following Table C-5 ~ Table C-8.

Table C-5 shows the standard character encoding of ISO/IEC 8859-1. ISO means International Organization for Standardization. The ISO 8859-1 also less formally called “Latin-1” is he first eight-bits coded character sets that developed by the ISO. It refers to ASCII that consisting of 192 characters from the Latin script in range 0xA0-0xFF. This character encoding is used throughout Western Europe, includes Albanian, Afrikaans, Breton, Danish, Faroese, Frisian, Galician, German, Greenlandic, Icelandic, Irish, Italian, Latin, Luxembourgish, Norwegian, Portuguese, Rhaeto-Romanic, Scottish Gaelic, Spanish, Swedish. English letters with no accent marks also can use ISO 8859-1.
In addition, it also commonly used in many languages outside Europe, such as Swahili, Indonesian, Malaysian and Tagalog.

Table C-5 : ASCII Block 1(ISO 8859-1)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A		¡	¢	£	¤	¥	¦	§	¨	©	ª	«	¬		®	¯
B	°	±	2	3	´	µ	¶	·	¸	1	0	»	¼	½	¾	¿
C	À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D	Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
E	à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
F	ð	ñ	ò	ó	ô	õ	ö	÷	ø	ù	ú	û	ü	ý	þ	ÿ

Table C-6 shows the standard characters of ISO/IEC 8859-2. ISO 8859-2 also cited as Latin-2 is the part 2 of the eight-bits coded character sets developed by ISO/IEC 8859. These code values can be used in almost any data interchange system to communicate in the following European languages : Croatian, Czech, Hungarian, Polish, Slovak, Slovenian, and Upper Sorbian. The Serbian, English, German, Latin can use ISO 8859-2 as well. Furthermore it is suitable to represent some western European languages like Finnish (with the exception of **å** used in Swedish and Finnish)

Table C-6 : ASCII Block 2(ISO 8859-2)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	●	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	!!	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A	SP	Ą	˘	Ł	¤	Ľ	Ś	§	¨	Š	Ş	Ť	Ź		Ž	Ż
B	°	ą	˛	ł	´	ľ	ś	ˇ	¸	š	ş	ť	ź	˝	ž	ż
C	Ŕ	Á	Â	Ă	Ä	Ĺ	Ć	Ç	Č	É	Ę	Ë	Ě	Í	Î	Ď
D	Đ	Ń	Ň	Ó	Ô	Ő	Ö	×	Ř	Ů	Ú	Ű	Ü	Ý	Ţ	ß
E	ŕ	á	â	ă	ä	ĺ	ć	ç	č	é	ę	ë	ě	í	î	ď
F	đ	ń	ň	ó	ô	ő	ö	÷	ř	ů	ú	ű	ü	ý	ţ	˙

Table C-7 shows the standard characters of ISO/IEC 8859-3. ISO 8859-3 also known as Latin-3 or “South European” is an eight-bits character encoding, third part of the ISO 8859 standard. It was designed originally to cover Turkish, Maltese and Esperanto, though the introduction of ISO 8859-9 superseded it for Turkish. The encoding remains popular with users of Esperanto and Maltese, though it also supports English, German, Italian, Latin and Portuguese.

Table C-7 : ASCII Block 3(ISO 8859-3)

H \ L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	●	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	!!	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A	SP	Ħ	˘	£	¤		Ĥ	§	¨	İ	Ş	Ğ	Ĵ			Ż
B	°	ħ	2	3	´	μ	ĥ	·	¸	ı	ş	ğ	ĵ	½		ż
C	À	Á	Â		Ä	Ċ	Ĉ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
D		Ñ	Ò	Ó	Ô	Ġ	Ö	×	Ĝ	Ù	Ú	Û	Ü	Ŭ	Ŝ	ß
E	à	á	â		ä	ċ	ĉ	ç	è	é	ê	ë	ì	í	î	ï
F		ñ	ò	ó	ô	ġ	ö	÷	ĝ	ù	ú	û	ü	ŭ	ŝ	˙

Table C-8 shows the standard characters of ISO/IEC 8859-4. ISO 8859-4 is known as Latin-4 or " North European"， is the forth part of the ISO 8859 eight-bits character encoding. It was designed originally to cover Estonian, Greenlandic, Latvian, Lithuanian, and Sami. This character set also supports Danish, English, Finnish, German, Latin, Norwegian, Slovenian, and Swedish.

Table C-8 : ASCII Block 4(ISO 8859-4)

H\L	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0		☺	☻	♥	♦	♣	♠	●	◘	○	◙	♂	♀	♪	♫	☼
1	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2	SP	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	
8																
9																
A	SP	Ą	ĸ	Ŗ	¤	Ĩ	Ļ	§	¨	Š	Ē	Ģ	Ŧ		Ž	¯
B	°	ą	˛	ŗ	´	ĩ	ļ	ˇ	¸	š	ē	ģ	ŧ	Ŋ	ž	ŋ
C	Ā	Á	Â	Ã	Ä	Å	Æ	Į	Č	É	Ę	Ë	Ė	Í	Î	Ī
D	Đ	Ņ	Ō	Ķ	Ô	Õ	Ö	×	Ø	Ų	Ú	Û	Ü	Ũ	Ū	ß
E	ā	á	â	ã	ä	å	æ	į	č	é	ę	ë	ė	í	î	ī
F	đ	ņ	ō	ķ	ô	õ	ö	÷	ø	ų	ú	û	ü	ũ	ū	˙

Appendix D . Font Table - GB Code

A1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A			、	。	·	ˉ	ˇ	¨	〃	々	—	～	‖	…	‘	’
B	“	”	〔	〕	〈	〉	《	》	「	」	『	』	〖	〗	【	】
C	±	×	÷	∶	∧	∨	∑	∏	∪	∩	∈	∷	√	⊥	∥	∠
D	⌒	⊙	∫	∮	≡	≌	≈	∽	∝	≠	≮	≯	≤	≥	∞	∵
E	∴	♂	♀	°	′	″	℃	$	¤	¢	£	‰	§	№	☆	★
F	○	●	◎	◇	◆	□	■	△	▲	※	→	←	↑	↓	〓	

A2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B		1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.	13.	14.	15.
C	16.	17.	18.	19.	20.	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)
D	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)	①	②	③	④	⑤	⑥	⑦
E	⑧	⑨	⑩			㈠	㈡	㈢	㈣	㈤	㈥	㈦	㈧	㈨	㈩	
F		Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ	Ⅷ	Ⅸ	Ⅹ	Ⅺ	Ⅻ			

A3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		!	"	#	¥	%	&	‘	(	)	*	+	,	—	.	/
B	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
C	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
D	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
E	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
F	p	q	r	s	t	u	v	w	x	y	z	{	|	}	￣	

A4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ	お	か	が	き	ぎ	く
B	ぐ	け	げ	こ	ご	さ	ざ	し	じ	す	ず	せ	ぜ	そ	ぞ	た
C	だ	ち	ぢ	っ	つ	づ	て	で	と	ど	な	に	ぬ	ね	の	は
D	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ	べ	ぺ	ほ	ぼ	ぽ	ま	み
E	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ	ら	り	る	れ	ろ	ゎ	わ
F	ゐ	ゑ	を	ん												

A5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ｧ	ｱ	ｨ	ｲ	ｩ	ｳ	ｪ	ｴ	ｫ	ｵ	ｶ	ｶﾞ	ｷ	ｷﾞ	ｸ
B	ｸﾞ	ｹ	ｹﾞ	ｺ	ｺﾞ	ｻ	ｻﾞ	ｼ	ｼﾞ	ｽ	ｽﾞ	ｾ	ｾﾞ	ｿ	ｿﾞ	ﾀ
C	ﾀﾞ	ﾁ	ﾁﾞ	ｯ	ﾂ	ﾂﾞ	ﾃ	ﾃﾞ	ﾄ	ﾄﾞ	ﾅ	ﾆ	ﾇ	ﾈ	ﾉ	ﾊ
D	ﾊﾞ	ﾊﾟ	ﾋ	ﾋﾞ	ﾋﾟ	ﾌ	ﾌﾞ	ﾌﾟ	ﾍ	ﾍﾞ	ﾍﾟ	ﾎ	ﾎﾞ	ﾎﾟ	ﾏ	ﾐ
E	ﾑ	ﾒ	ﾓ	ｬ	ﾔ	ｭ	ﾕ	ｮ	ﾖ	ﾗ	ﾘ	ﾙ	ﾚ	ﾛ	ﾜ	ﾜ
F	ｷ	ﾝ	ﾞ	ﾟ	ヴ	ヵ	ヶ									

A6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ	Ν	Ξ	Ο
B	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω							
C		α	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο
D	π	ρ	σ	τ	υ	φ	χ	ψ	ω							
E																
F																

A7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
B	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
C	Ю	Я														
D		а	б	в	г	д	е	ё	ж	з	и	й	к	л	м	н
E	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ъ	ы	ь	э
F	ю	я														

A8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		ō	ó	ǒ	ò	ē	é	ě	è	ī	í	ǐ	ì	ō	ó	ǒ
B	ò	ū	ú	ǔ	ù	ǖ	ǘ	ǚ	ǜ	ü	ê	ɑ	ḿ	ń	ň	ṅ
C	ɡ					ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ
D	ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ	ㄟ
E	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ	ㄧ	ㄨ	ㄩ						
F																

A9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A					—	—	|	|	---	---	¦	¦	----	----	¦	¦
B	⌐	⌐	⌐	⌐	¬	¬	¬	¬	└	└	└	└	┘	┘	┘	┘
C	├	├	├	├	├	├	├	├	┤	┤	┤	┤	┤	┤	┤	┤
D	┬	┬	┬	┬	┬	┬	┬	┬	┴	┴	┴	┴	┴	┴	┴	┴
E	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼	┼
F																

AA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

AB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

AC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

AD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

AE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

AF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A																
B																
C																
D																
E																
F																

B0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		啊	阿	埃	挨	哎	唉	哀	皑	癌	蔼	矮	艾	碍	爱	隘
B	鞍	氨	安	俺	按	暗	岸	胺	案	肮	昂	盎	凹	敖	熬	翱
C	袄	傲	奥	懊	澳	芭	捌	扒	叭	吧	笆	八	疤	巴	拔	跋
D	靶	把	耙	坝	霸	罢	爸	白	柏	百	摆	佰	败	拜	稗	斑
E	班	搬	扳	般	颁	板	版	扮	拌	伴	瓣	半	办	绊	邦	帮
F	梆	榜	膀	绑	棒	磅	蚌	镑	傍	谤	苞	胞	包	褒	剥	

B1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		薄	雹	保	堡	饱	宝	抱	报	暴	豹	鲍	爆	杯	碑	悲
B	卑	北	辈	背	贝	钡	倍	狈	备	惫	焙	被	奔	苯	本	笨
C	崩	绷	甭	泵	蹦	迸	逼	鼻	比	鄙	笔	彼	碧	蓖	蔽	毕
D	毙	毖	币	庇	痹	闭	敝	弊	必	辟	壁	臂	避	陛	鞭	边
E	编	贬	扁	便	变	卞	辨	辩	辫	遍	标	彪	膘	表	鳖	憋
F	别	瘪	彬	斌	濒	滨	宾	摈	兵	冰	柄	丙	秉	饼	炳	

B2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		病	并	玻	菠	播	拨	钵	波	博	勃	搏	铂	箔	伯	帛
B	舶	脖	膊	渤	泊	驳	捕	卜	哺	补	埠	不	布	步	簿	部
C	怖	擦	猜	裁	材	才	财	睬	踩	采	彩	菜	蔡	餐	参	蚕
D	残	惭	惨	灿	苍	舱	仓	沧	藏	操	糙	槽	曹	草	厕	策
E	侧	册	测	层	蹭	插	叉	茬	茶	查	碴	搽	察	岔	差	诧
F	拆	柴	豺	搀	掺	蝉	馋	谗	缠	铲	产	阐	颤	昌	猖	

B3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		场	尝	常	长	偿	肠	厂	敞	畅	唱	倡	超	抄	钞	朝
B	嘲	潮	巢	吵	炒	车	扯	撤	掣	彻	澈	郴	臣	辰	尘	晨
C	忱	沉	陈	趁	衬	撑	称	城	橙	成	呈	乘	程	惩	澄	诚
D	承	逞	骋	秤	吃	痴	持	匙	池	迟	弛	驰	耻	齿	侈	尺
E	赤	翅	斥	炽	充	冲	虫	崇	宠	抽	酬	畴	踌	稠	愁	筹
F	仇	绸	瞅	丑	臭	初	出	橱	厨	躇	锄	雏	滁	除	楚	

B4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		础	储	矗	搐	触	处	揣	川	穿	椽	传	船	喘	串	疮
B	窗	幢	床	闯	创	吹	炊	捶	锤	垂	春	椿	醇	唇	淳	纯
C	蠢	戳	绰	疵	茨	磁	雌	辞	慈	瓷	词	此	刺	赐	次	聪
D	葱	囱	匆	从	丛	凑	粗	醋	簇	促	蹿	篡	窜	摧	崔	催
E	脆	瘁	粹	淬	翠	村	存	寸	磋	撮	搓	措	挫	错	搭	达
F	答	瘩	打	大	呆	歹	傣	戴	带	殆	代	贷	袋	待	逮	

B5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		怠	耽	担	丹	单	郸	掸	胆	旦	氮	但	惮	淡	诞	弹
B	蛋	当	挡	党	荡	档	刀	捣	蹈	倒	岛	祷	导	到	稻	悼
C	道	盗	德	得	的	蹬	灯	登	等	瞪	凳	邓	堤	低	滴	迪
D	敌	笛	狄	涤	翟	嫡	抵	底	地	蒂	第	帝	弟	递	缔	颠
E	掂	滇	碘	点	典	靛	垫	电	佃	甸	店	惦	奠	淀	殿	碉
F	叼	雕	凋	刁	掉	吊	钓	调	跌	爹	碟	蝶	迭	谍	叠	

B6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		丁	盯	叮	钉	顶	鼎	锭	定	订	丢	东	冬	董	懂	动
B	栋	侗	恫	冻	洞	兜	抖	斗	陡	豆	逗	痘	都	督	毒	犊
C	独	读	堵	睹	赌	杜	镀	肚	度	渡	妒	端	短	锻	段	断
D	缎	堆	兑	队	对	墩	吨	蹲	敦	顿	囤	钝	盾	遁	掇	哆
E	多	夺	垛	躲	朵	跺	舵	剁	惰	堕	蛾	峨	鹅	俄	额	讹
F	娥	恶	厄	扼	遏	鄂	饿	恩	而	儿	耳	尔	饵	洱	二	

B7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		贰	发	罚	筏	伐	乏	阀	法	珐	藩	帆	番	翻	樊	矾
B	钒	繁	凡	烦	反	返	范	贩	犯	饭	泛	坊	芳	方	肪	房
C	防	妨	仿	访	纺	放	菲	非	啡	飞	肥	匪	诽	吠	肺	废
D	沸	费	芬	酚	吩	氛	分	纷	坟	焚	汾	粉	奋	份	忿	愤
E	粪	丰	封	枫	蜂	峰	锋	风	疯	烽	逢	冯	缝	讽	奉	凤
F	佛	否	夫	敷	肤	孵	扶	拂	辐	幅	氟	符	伏	俘	服	

B8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		浮	涪	福	袱	弗	甫	抚	辅	俯	釜	斧	脯	腑	府	腐
B	赴	副	覆	赋	复	傅	付	阜	父	腹	负	富	讣	附	妇	缚
C	咐	噶	嘎	该	改	概	钙	盖	溉	干	甘	杆	柑	竿	肝	赶
D	感	秆	敢	赣	冈	刚	钢	缸	肛	纲	岗	港	杠	篙	皋	高
E	膏	羔	糕	搞	镐	稿	告	哥	歌	搁	戈	鸽	胳	疙	割	革
F	葛	格	蛤	阁	隔	铬	个	各	给	根	跟	耕	更	庚	羹	

B9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		埂	耿	梗	工	攻	功	恭	龚	供	躬	公	宫	弓	巩	汞
B	拱	贡	共	钩	勾	沟	苟	狗	垢	构	购	够	辜	菇	咕	箍
C	估	沽	孤	姑	鼓	古	蛊	骨	谷	股	故	顾	固	雇	刮	瓜
D	剐	寡	挂	褂	乖	拐	怪	棺	关	官	冠	观	管	馆	罐	惯
E	灌	贯	光	广	逛	瑰	规	圭	硅	归	龟	闺	轨	鬼	诡	癸
F	桂	柜	跪	贵	刽	辊	滚	棍	锅	郭	国	果	裹	过	哈	

BA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		骸	孩	海	氦	亥	害	骇	酣	憨	邯	韩	含	涵	寒	函
B	喊	罕	翰	撼	捍	旱	憾	悍	焊	汗	汉	夯	杭	航	壕	嚎
C	豪	毫	郝	好	耗	号	浩	呵	喝	荷	菏	核	禾	和	何	合
D	盒	貉	阂	河	涸	赫	褐	鹤	贺	嘿	黑	痕	很	狠	恨	哼
E	亨	横	衡	恒	轰	哄	烘	虹	鸿	洪	宏	弘	红	喉	侯	猴
F	吼	厚	候	后	呼	乎	忽	瑚	壶	葫	胡	蝴	狐	糊	湖	

BB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		弧	虎	唬	护	互	沪	户	花	哗	华	猾	滑	画	划	化
B	话	槐	徊	怀	淮	坏	欢	环	桓	还	缓	换	患	唤	痪	豢
C	焕	涣	宦	幻	荒	慌	黄	磺	蝗	簧	皇	凰	惶	煌	晃	幌
D	恍	谎	灰	挥	辉	徽	恢	蛔	回	毁	悔	慧	卉	惠	晦	贿
E	秽	会	烩	汇	讳	诲	绘	荤	昏	婚	魂	浑	混	豁	活	伙
F	火	获	或	惑	霍	货	祸	击	圾	基	机	畸	稽	积	箕	

BC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		肌	饥	迹	激	讥	鸡	姬	绩	缉	吉	极	棘	辑	籍	集
B	及	急	疾	汲	即	嫉	级	挤	几	脊	己	蓟	技	冀	季	伎
C	祭	剂	悸	济	寄	寂	计	记	既	忌	际	妓	继	纪	嘉	枷
D	夹	佳	家	加	荚	颊	贾	甲	钾	假	稼	价	架	驾	嫁	歼
E	监	坚	尖	笺	间	煎	兼	肩	艰	奸	缄	茧	检	柬	碱	硷
F	拣	捡	简	俭	剪	减	荐	槛	鉴	践	贱	见	键	箭	件	

BD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		健	舰	剑	饯	渐	溅	涧	建	僵	姜	将	浆	江	疆	蒋
B	桨	奖	讲	匠	酱	降	蕉	椒	礁	焦	胶	交	郊	浇	骄	娇
C	嚼	搅	铰	矫	侥	脚	狡	角	饺	缴	绞	剿	教	酵	轿	较
D	叫	窖	揭	接	皆	秸	街	阶	截	劫	节	桔	杰	捷	睫	竭
E	洁	结	解	姐	戒	藉	芥	界	借	介	疥	诫	届	巾	筋	斤
F	金	今	津	襟	紧	锦	仅	谨	进	靳	晋	禁	近	烬	浸	

BE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		尽	劲	荆	兢	茎	睛	晶	鲸	京	惊	精	粳	经	井	警
B	景	颈	静	境	敬	镜	径	痉	靖	竟	竞	净	炯	窘	揪	究
C	纠	玖	韭	久	灸	九	酒	厩	救	旧	臼	舅	咎	就	疚	鞠
D	拘	狙	疽	居	驹	菊	局	咀	矩	举	沮	聚	拒	据	巨	具
E	距	踞	锯	俱	句	惧	炬	剧	捐	鹃	娟	倦	眷	卷	绢	撅
F	攫	抉	掘	倔	爵	觉	决	诀	绝	均	菌	钧	军	君	峻	

BF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		俊	竣	浚	郡	骏	喀	咖	卡	咯	开	揩	楷	凯	慨	刊
B	堪	勘	坎	砍	看	康	慷	糠	扛	抗	亢	炕	考	拷	烤	靠
C	坷	苛	柯	棵	磕	颗	科	壳	咳	可	渴	克	刻	客	课	肯
D	啃	垦	恳	坑	吭	空	恐	孔	控	抠	口	扣	寇	枯	哭	窟
E	苦	酷	库	裤	夸	垮	挎	跨	胯	块	筷	侩	快	宽	款	匡
F	筐	狂	框	矿	眶	旷	况	亏	盔	岿	窥	葵	奎	魁	傀	

C0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		馈	愧	溃	坤	昆	捆	困	括	扩	廓	阔	垃	拉	喇	蜡
B	腊	辣	啦	莱	来	赖	蓝	婪	栏	拦	篮	阑	兰	澜	谰	揽
C	览	懒	缆	烂	滥	琅	榔	狼	廊	郎	朗	浪	捞	劳	牢	老
D	佬	姥	酪	烙	涝	勒	乐	雷	镭	蕾	磊	累	儡	垒	擂	肋
E	类	泪	棱	楞	冷	厘	梨	犁	黎	篱	狸	离	漓	理	李	里
F	鲤	礼	莉	荔	吏	栗	丽	厉	励	砾	历	利	傈	例	俐	

C1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		痢	立	粒	沥	隶	力	璃	哩	俩	联	莲	连	镰	廉	怜
B	涟	帘	敛	脸	链	恋	炼	练	粮	凉	梁	粱	良	两	辆	量
C	晾	亮	谅	撩	聊	僚	疗	燎	寥	辽	潦	了	撂	镣	廖	料
D	列	裂	烈	劣	猎	琳	林	磷	霖	临	邻	鳞	淋	凛	赁	吝
E	拎	玲	菱	零	龄	铃	伶	羚	凌	灵	陵	岭	领	另	令	溜
F	琉	榴	硫	馏	留	刘	瘤	流	柳	六	龙	聋	咙	笼	窿	

C2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		隆	垄	拢	陇	楼	娄	搂	篓	漏	陋	芦	卢	颅	庐	炉
B	掳	卤	虏	鲁	麓	碌	露	路	赂	鹿	潞	禄	录	陆	戮	驴
C	吕	铝	侣	旅	履	屡	缕	虑	氯	律	率	滤	绿	峦	挛	孪
D	滦	卵	乱	掠	略	抡	轮	伦	仑	沦	纶	论	萝	螺	罗	逻
E	锣	箩	骡	裸	落	洛	骆	络	妈	麻	玛	码	蚂	马	骂	嘛
F	吗	埋	买	麦	卖	迈	脉	瞒	馒	蛮	满	蔓	曼	慢	漫	

C3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		谩	芒	茫	盲	氓	忙	莽	猫	茅	锚	毛	矛	铆	卯	茂
B	冒	帽	貌	贸	么	玫	枚	梅	酶	霉	煤	没	眉	媒	镁	每
C	美	昧	寐	妹	媚	门	闷	们	萌	蒙	檬	盟	锰	猛	梦	孟
D	眯	醚	靡	糜	迷	谜	弥	米	秘	觅	泌	蜜	密	幂	棉	眠
E	绵	冕	免	勉	娩	缅	面	苗	描	瞄	藐	秒	渺	庙	妙	蔑
F	灭	民	抿	皿	敏	悯	闽	明	螟	鸣	铭	名	命	谬	摸	

C4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		摹	蘑	模	膜	磨	摩	魔	抹	末	莫	墨	默	沫	漠	寞
B	陌	谋	牟	某	拇	牡	亩	姆	母	墓	暮	幕	募	慕	木	目
C	睦	牧	穆	拿	哪	呐	钠	那	娜	纳	氖	乃	奶	耐	奈	南
D	男	难	囊	挠	脑	恼	闹	淖	呢	馁	内	嫩	能	妮	霓	倪
E	泥	尼	拟	你	匿	腻	逆	溺	蔫	拈	年	碾	撵	捻	念	娘
F	酿	鸟	尿	捏	聂	孽	啮	镊	镍	涅	您	柠	狞	凝	宁	

C5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		拧	泞	牛	扭	钮	纽	脓	浓	农	弄	奴	努	怒	女	暖
B	虐	疟	挪	懦	糯	诺	哦	欧	鸥	殴	藕	呕	偶	沤	啪	趴
C	爬	帕	怕	琶	拍	排	牌	徘	湃	派	攀	潘	盘	磐	盼	畔
D	判	叛	乓	庞	旁	耪	胖	抛	咆	刨	炮	袍	跑	泡	呸	胚
E	培	裴	赔	陪	配	佩	沛	喷	盆	砰	抨	烹	澎	彭	蓬	棚
F	硼	篷	膨	朋	鹏	捧	碰	坯	砒	霹	批	披	劈	琵	毗	

C6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		啤	脾	疲	皮	匹	痞	僻	屁	譬	篇	偏	片	骗	飘	漂
B	瓢	票	撇	瞥	拼	频	贫	品	聘	乒	坪	苹	萍	平	凭	瓶
C	评	屏	坡	泼	颇	婆	破	魄	迫	粕	剖	扑	铺	仆	莆	葡
D	菩	蒲	埔	朴	圃	普	浦	谱	曝	瀑	期	欺	栖	戚	妻	七
E	凄	漆	柒	沏	其	棋	奇	歧	畦	崎	脐	齐	旗	祈	祁	骑
F	起	岂	乞	企	启	契	砌	器	气	迄	弃	汽	泣	讫	掐	

C7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		恰	洽	牵	扦	钎	铅	千	迁	签	仟	谦	乾	黔	钱	钳
B	前	潜	遣	浅	谴	堑	嵌	欠	歉	枪	呛	腔	羌	墙	蔷	强
C	抢	橇	锹	敲	悄	桥	瞧	乔	侨	巧	鞘	撬	翘	峭	俏	窍
D	切	茄	且	怯	窃	钦	侵	亲	秦	琴	勤	芹	擒	禽	寝	沁
E	青	轻	氢	倾	卿	清	擎	晴	氰	情	顷	请	庆	琼	穷	秋
F	丘	邱	球	求	囚	酋	泅	趋	区	蛆	曲	躯	屈	驱	渠	

C8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		取	娶	龋	趣	去	圈	颧	权	醛	泉	全	痊	拳	犬	券
B	劝	缺	炔	瘸	却	鹊	榷	确	雀	裙	群	然	燃	冉	染	瓤
C	壤	攘	嚷	让	饶	扰	绕	惹	热	壬	仁	人	忍	韧	任	认
D	刃	妊	纫	扔	仍	日	戎	茸	蓉	荣	融	熔	溶	容	绒	冗
E	揉	柔	肉	茹	蠕	儒	孺	如	辱	乳	汝	入	褥	软	阮	蕊
F	瑞	锐	闰	润	若	弱	撒	洒	萨	腮	鳃	塞	赛	三	叁	

C9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		伞	散	桑	嗓	丧	搔	骚	扫	嫂	瑟	色	涩	森	僧	莎
B	砂	杀	刹	沙	纱	傻	啥	煞	筛	晒	珊	苫	杉	山	删	煽
C	衫	闪	陕	擅	赡	膳	善	汕	扇	缮	墒	伤	商	赏	晌	上
D	尚	裳	梢	捎	稍	烧	芍	勺	韶	少	哨	邵	绍	奢	赊	蛇
E	舌	舍	赦	摄	射	慑	涉	社	设	砷	申	呻	伸	身	深	娠
F	绅	神	沈	审	婶	甚	肾	慎	渗	声	生	甥	牲	升	绳	

CA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		省	盛	剩	胜	圣	师	失	狮	施	湿	诗	尸	虱	十	石
B	拾	时	什	食	蚀	实	识	史	矢	使	屎	驶	始	式	示	士
C	世	柿	事	拭	誓	逝	势	是	嗜	噬	适	仕	侍	释	饰	氏
D	市	恃	室	视	试	收	手	首	守	寿	授	售	受	瘦	兽	蔬
E	枢	梳	殊	抒	输	叔	舒	淑	疏	书	赎	孰	熟	薯	暑	曙
F	署	蜀	黍	鼠	属	术	述	树	束	戍	竖	墅	庶	数	漱	

CB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		恕	刷	耍	摔	衰	甩	帅	栓	拴	霜	双	爽	谁	水	睡
B	税	吮	瞬	顺	舜	说	硕	朔	烁	斯	撕	嘶	思	私	司	丝
C	死	肆	寺	嗣	四	伺	似	饲	巳	松	耸	怂	颂	送	宋	讼
D	诵	搜	艘	擞	嗽	苏	酥	俗	素	速	粟	僳	塑	溯	宿	诉
E	肃	酸	蒜	算	虽	隋	随	绥	髓	碎	岁	穗	遂	隧	祟	孙
F	损	笋	蓑	梭	唆	缩	琐	索	锁	所	塌	他	它	她	塔	

CC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		獭	挞	蹋	踏	胎	苔	抬	台	泰	酞	太	态	汰	坍	摊
B	贪	瘫	滩	坛	檀	痰	潭	谭	谈	坦	毯	袒	碳	探	叹	炭
C	汤	塘	搪	堂	棠	膛	唐	糖	倘	躺	淌	趟	烫	掏	涛	滔
D	绦	萄	桃	逃	淘	陶	讨	套	特	藤	腾	疼	誊	梯	剔	踢
E	锑	提	题	蹄	啼	体	替	嚏	惕	涕	剃	屉	天	添	填	田
F	甜	恬	舔	腆	挑	条	迢	眺	跳	贴	铁	帖	厅	听	烃	

CD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		汀	廷	停	亭	庭	挺	艇	通	桐	酮	瞳	同	铜	彤	童
B	桶	捅	筒	统	痛	偷	投	头	透	凸	秃	突	图	徒	途	涂
C	屠	土	吐	兔	湍	团	推	颓	腿	蜕	褪	退	吞	屯	臀	拖
D	托	脱	鸵	陀	驮	驼	椭	妥	拓	唾	挖	哇	蛙	洼	娃	瓦
E	袜	歪	外	豌	弯	湾	玩	顽	丸	烷	完	碗	挽	晚	皖	惋
F	宛	婉	万	腕	汪	王	亡	枉	网	往	旺	望	忘	妄	威	

CE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		巍	微	危	韦	违	桅	围	唯	惟	为	潍	维	苇	萎	委
B	伟	伪	尾	纬	未	蔚	味	畏	胃	喂	魏	位	渭	谓	尉	慰
C	卫	瘟	温	蚊	文	闻	纹	吻	稳	紊	问	嗡	翁	瓮	挝	蜗
D	涡	窝	我	斡	卧	握	沃	巫	呜	钨	乌	污	诬	屋	无	芜
E	梧	吾	吴	毋	武	五	捂	午	舞	伍	侮	坞	戊	雾	晤	物
F	勿	务	悟	误	昔	熙	析	西	硒	矽	晰	嘻	吸	锡	牺	

CF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		稀	息	希	悉	膝	夕	惜	熄	烯	溪	汐	犀	檄	袭	席
B	习	媳	喜	铣	洗	系	隙	戏	细	瞎	虾	匣	霞	辖	暇	峡
C	侠	狭	下	厦	夏	吓	掀	锨	先	仙	鲜	纤	咸	贤	衔	舷
D	闲	涎	弦	嫌	显	险	现	献	县	腺	馅	羡	宪	陷	限	线
E	相	厢	镶	香	箱	襄	湘	乡	翔	祥	详	想	响	享	项	巷
F	橡	像	向	象	萧	硝	霄	削	哮	嚣	销	消	宵	淆	晓	

D0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		小	孝	校	肖	啸	笑	效	楔	些	歇	蝎	鞋	协	挟	携
B	邪	斜	胁	谐	写	械	卸	蟹	懈	泄	泻	谢	屑	薪	芯	锌
C	欣	辛	新	忻	心	信	衅	星	腥	猩	惺	兴	刑	型	形	邢
D	行	醒	幸	杏	性	姓	兄	凶	胸	匈	汹	雄	熊	休	修	羞
E	朽	嗅	锈	秀	袖	绣	墟	戌	需	虚	嘘	须	徐	许	蓄	酗
F	叙	旭	序	畜	恤	絮	婿	绪	续	轩	喧	宣	悬	旋	玄	

D1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		选	癣	眩	绚	靴	薛	学	穴	雪	血	勋	熏	循	旬	询
B	寻	驯	巡	殉	汛	训	讯	逊	迅	压	押	鸦	鸭	呀	丫	芽
C	牙	蚜	崖	衙	涯	雅	哑	亚	讶	焉	咽	阉	烟	淹	盐	严
D	研	蜒	岩	延	言	颜	阎	炎	沿	奄	掩	眼	衍	演	艳	堰
E	燕	厌	砚	雁	唁	彦	焰	宴	谚	验	殃	央	鸯	秧	杨	扬
F	佯	疡	羊	洋	阳	氧	仰	痒	养	样	漾	邀	腰	妖	瑶	

D2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		摇	尧	遥	窑	谣	姚	咬	舀	药	要	耀	椰	噎	耶	爷
B	野	冶	也	页	掖	业	叶	曳	腋	夜	液	一	壹	医	揖	铱
C	依	伊	衣	颐	夷	遗	移	仪	胰	疑	沂	宜	姨	彝	椅	蚁
D	倚	已	乙	矣	以	艺	抑	易	邑	屹	亿	役	臆	逸	肄	疫
E	亦	裔	意	毅	忆	义	益	溢	诣	议	谊	译	异	翼	翌	绎
F	茵	荫	因	殷	音	阴	姻	吟	银	淫	寅	饮	尹	引	隐	

D3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		印	英	樱	婴	鹰	应	缨	莹	萤	营	荧	蝇	迎	赢	盈
B	影	颖	硬	映	哟	拥	佣	臃	痈	庸	雍	踊	蛹	咏	泳	涌
C	永	恿	勇	用	幽	优	悠	忧	尤	由	邮	铀	犹	油	游	酉
D	有	友	右	佑	釉	诱	又	幼	迂	淤	于	盂	榆	虞	愚	舆
E	余	俞	逾	鱼	愉	渝	渔	隅	予	娱	雨	与	屿	禹	宇	语
F	羽	玉	域	芋	郁	吁	遇	喻	峪	御	愈	欲	狱	育	誉	

D4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		浴	寓	裕	预	豫	驭	鸳	渊	冤	元	垣	袁	原	援	辕
B	园	员	圆	猿	源	缘	远	苑	愿	怨	院	曰	约	越	跃	钥
C	岳	粤	月	悦	阅	耘	云	郧	匀	陨	允	运	蕴	酝	晕	韵
D	孕	匝	砸	杂	栽	哉	灾	宰	载	再	在	咱	攒	暂	赞	赃
E	脏	葬	遭	糟	凿	藻	枣	早	澡	蚤	躁	噪	造	皂	灶	燥
F	责	择	则	泽	贼	怎	增	憎	曾	赠	扎	喳	渣	札	轧	

D5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		铡	闸	眨	栅	榨	咋	乍	炸	诈	摘	斋	宅	窄	债	寨
B	瞻	毡	詹	粘	沾	盏	斩	辗	崭	展	蘸	栈	占	战	站	湛
C	绽	樟	章	彰	漳	张	掌	涨	杖	丈	帐	账	仗	胀	瘴	障
D	招	昭	找	沼	赵	照	罩	兆	肇	召	遮	折	哲	蛰	辙	者
E	锗	蔗	这	浙	珍	斟	真	甄	砧	臻	贞	针	侦	枕	疹	诊
F	震	振	镇	阵	蒸	挣	睁	征	狰	争	怔	整	拯	正	政	

D6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		帧	症	郑	证	芝	枝	支	吱	蜘	知	肢	脂	汁	之	织
B	职	直	植	殖	执	值	侄	址	指	止	趾	只	旨	纸	志	挚
C	掷	至	致	置	帜	峙	制	智	秩	稚	质	炙	痔	滞	治	窒
D	中	盅	忠	钟	衷	终	种	肿	重	仲	众	舟	周	州	洲	诌
E	粥	轴	肘	帚	咒	皱	宙	昼	骤	珠	株	蛛	朱	猪	诸	诛
F	逐	竹	烛	煮	拄	瞩	嘱	主	著	柱	助	蛀	贮	铸	筑	

D7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		住	注	祝	驻	抓	爪	拽	专	砖	转	撰	赚	篆	桩	庄
B	装	妆	撞	壮	状	椎	锥	追	赘	坠	缀	谆	准	捉	拙	卓
C	桌	琢	茁	酌	啄	着	灼	浊	兹	咨	资	姿	滋	淄	孜	紫
D	仔	籽	滓	子	自	渍	字	鬃	棕	踪	宗	综	总	纵	邹	走
E	奏	揍	租	足	卒	族	祖	诅	阻	组	钻	纂	嘴	醉	最	罪
F	尊	遵	昨	左	佐	柞	做	作	坐	座						

D8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		亍	丌	兀	丐	廿	卅	丕	亘	丞	鬲	孬	噩	丨	禺	丿
B	匕	毛	夭	爻	卮	氐	囟	胤	馗	毓	睾	鼗	丶	亟	鼐	乜
C	乩	亓	芈	孛	啬	嘏	仄	厍	厝	厣	厥	厮	靥	匚	叵	
D	匦	匮	匾	赜	卦	卣	刂	刈	刎	刭	刳	刿	剀	剌	剞	剡
E	剜	蒯	剽	劂	劁	劐	剖	冂	罔	亻	仃	仉	仂	仨	仡	仫
F	仞	伛	仳	伢	佤	仵	伥	伧	伉	伫	佞	佧	攸	佚	佝	

D9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		佟	佗	伲	伽	佶	佴	侑	侉	侃	侏	佾	佻	侪	佼	侬
B	侔	俦	俨	俪	俅	俚	俣	俜	俑	俟	俸	倩	偌	俳	倬	倏
C	倮	倭	俾	倜	倌	倥	倨	偾	偃	偕	偈	偎	偬	偻	傥	傧
D	傩	傺	僖	儆	僭	僬	僦	僮	儇	儋	仝	氽	佘	佥	俎	龠
E	汆	籴	兮	巽	黉	馘	冁	夔	勹	匍	訇	匐	凫	夙	兕	亠
F	兖	亳	衮	袤	亵	脔	裒	禀	嬴	蠃	羸	冫	冱	冽	冼	

DA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		凇	冖	冢	冥	讠	讦	讧	讪	讴	讵	讷	诂	诃	诋	诏
B	诎	诒	诓	诔	诖	诘	诙	诜	诟	诠	诤	诨	诩	诮	诰	诳
C	诶	诹	诼	诿	谀	谂	谄	谇	谌	谏	谑	谒	谔	谕	谖	谙
D	谛	谘	谝	谟	谠	谡	谥	谧	谪	谫	谮	谯	谲	谳	谵	谶
E	卩	卺	阝	阢	阡	阱	阪	阽	阼	陂	陉	陔	陟	陧	陬	陲
F	陴	隈	隍	隗	隰	邗	邛	邝	邙	邬	邡	邴	邳	邶	邺	

DB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		邸	邰	郏	郅	邾	郐	郄	郇	郓	郦	郢	郜	郗	郛	郫
B	郯	郾	鄄	鄢	鄞	鄣	鄱	鄯	鄹	酃	酆	刍	奂	劢	劬	劭
C	劾	哿	勐	勖	勰	叟	燮	矍	廴	凵	凼	鬯	厶	弁	畚	巯
D	坌	垩	垡	塾	墼	壅	壑	圩	圬	圪	圳	圹	圮	圯	坜	圻
E	坂	坩	垅	坫	垆	坼	坻	坨	坭	坶	坳	垭	垤	垌	垲	埏
F	垧	垴	垓	垠	埕	埘	埚	埙	埒	垸	埴	埯	埸	埤	埝	

DC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		堋	堍	埽	埭	堀	堞	堙	塄	堠	塥	塬	墁	墉	墚	墀
B	馨	鼙	懿	艹	艽	艿	芏	芊	芨	芄	芎	芑	芗	芙	芫	芸
C	芾	芰	苈	苊	苣	芘	芷	芮	苋	苌	苁	芩	芴	芡	芪	芟
D	苄	苎	芤	苡	茉	苷	苤	茏	茇	苜	苴	苒	苘	茌	苻	苓
E	茑	茚	茆	茔	茕	苠	苕	茜	荑	荛	荜	茈	莒	茼	茴	茱
F	莛	荞	茯	荏	荇	荃	荟	荀	茗	荠	茭	茺	茳	荦	荥	

DD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		荨	茛	荩	荬	荪	荭	荮	莰	荸	莳	莴	莠	莪	莓	莜
B	莅	荼	莶	莩	荽	莸	荻	莘	莞	莨	莺	莼	菁	萁	菥	菘
C	堇	萘	萋	菝	菽	菖	萜	萸	萑	萆	菔	菟	萏	萃	菸	菹
D	菪	菅	菀	萦	菰	菡	葜	葑	葚	葙	葳	蒇	蒈	葺	蒉	葸
E	萼	葆	葩	葶	蒌	蒎	萱	葭	蓁	蓍	蓐	蓦	蒽	蓓	蓊	蒿
F	蒺	蓠	蒡	蒹	蒴	蒗	蓥	蓣	蔌	甍	蔸	蓰	蔹	蔟	蔺	

DE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		蕖	蔻	蓿	蓼	蕙	蕈	蕨	蕤	蕞	蕺	瞢	蕃	蕲	蕻	薤
B	薨	薇	薏	蕹	薮	薜	薅	薹	薷	薰	藓	藁	藜	藿	蘧	蘅
C	蘩	蘖	蘼	廾	弈	夼	奁	耷	奕	奚	奘	匏	尢	尥	尬	尴
D	扌	扪	抟	抻	拊	拚	拗	拮	挢	拶	挹	捋	捃	掭	揶	捱
E	捺	掎	掴	捭	掬	掊	捩	掮	掼	揲	揸	揠	揿	揄	揞	揎
F	摒	揆	掾	摅	摁	搋	搛	搠	搌	搦	搡	摞	撄	摭	撖	

DF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		摺	撷	撸	撙	撺	擀	擐	擗	擤	擢	攉	攥	攮	弋	忒
B	甙	弑	卟	叱	叽	叩	叨	叻	吒	吖	吆	呋	呒	呓	呔	呖
C	呃	吡	呗	呙	吣	吲	咂	咔	呷	呱	呤	咚	咛	咄	呶	呦
D	咝	哐	咭	哂	咴	哒	咧	咦	哓	哔	呲	咣	哕	咻	咿	哌
E	哙	哚	哜	咩	咪	咤	哝	哏	哞	唛	哧	唠	哽	唔	哳	唢
F	唣	唏	唑	唧	唪	啧	喏	喵	啉	啭	啁	啕	唿	啐	唼	

E0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		唷	啖	啵	啶	啷	唳	唰	啜	喋	嗒	喃	喱	喹	喈	喁
B	喟	啾	嗖	喑	啻	嗟	喽	喾	喔	喙	嗪	嗷	嗉	嘟	嗑	嗫
C	嗬	嗔	嗦	嗝	嗄	嗯	嗥	嗲	嗳	嗌	嗍	嗨	嗵	嗤	辔	嘞
D	嘈	嘌	嘁	嘤	嘣	嗾	嘀	嘧	嘭	噘	嘹	噗	嘬	噍	噢	噙
E	噜	噌	噔	嚆	噤	噱	噫	噻	噼	嚅	嚓	嚯	囔	囗	囝	囡
F	囵	囫	囹	囿	圄	圊	圉	圜	帏	帙	帔	帑	帱	帻	帼	

E1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		帷	幄	幔	幛	幞	幡	岌	屺	岍	岐	岖	岈	岘	岙	岑
B	岚	岜	岵	岢	岽	岬	岫	岱	岣	峁	岷	峄	峒	峤	峋	峥
C	崂	崃	崧	崦	崮	崤	崞	崆	崛	嵘	崾	崴	崽	嵬	嵛	嵯
D	嵝	嵫	嵋	嵊	嵩	嵴	嶂	嶙	嶝	豳	嶷	巅	彳	彷	徂	徇
E	徉	後	徕	徙	徜	徨	徭	徵	徼	衢	彡	犭	犰	犴	犷	犸
F	狃	狁	狎	狍	狒	狨	狯	狩	狲	狴	狷	猁	狳	猃	狺	

E2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		狻	猗	猓	猡	猊	猞	猝	猕	猢	猹	猥	猬	猸	猱	獐
B	獍	獗	獠	獬	獯	獾	舛	夥	飧	夤	夂	饣	饧	饨	饩	饪
C	饫	饬	饴	饷	饽	馀	馄	馇	馊	馍	馐	馑	馓	馔	馕	庀
D	庑	庋	庖	庥	庠	庹	庵	庾	庳	赓	廒	廑	廛	廨	廪	膺
E	忄	忉	忖	忏	怃	忮	怄	忡	忤	忾	怅	怆	忪	忭	忸	怙
F	怵	怦	怛	怏	怍	怩	怫	怊	怿	怡	恸	恹	恻	恺	恂	

E3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		恪	恽	悖	悚	悭	悝	悃	悒	悌	悛	惬	悻	悱	惝	惘
B	惆	惚	悴	愠	愦	愕	愣	惴	愀	愎	愫	慊	慵	憬	憔	憧
C	憷	懔	懵	忝	隳	闩	闫	闱	闳	闵	闶	闼	闾	阃	阄	阆
D	阈	阊	阋	阌	阍	阏	阒	阕	阖	阗	阙	阚	丬	爿	戕	氵
E	汔	汜	汊	沣	沅	沐	沔	沌	汨	汩	汴	汶	沆	沩	泐	泔
F	沭	泷	泸	泱	泗	沲	泠	泖	泺	泫	泮	沱	泓	泯	泾	

E4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		洹	洧	洌	浃	浈	洇	洄	洙	洎	洫	浍	洮	洵	洚	浏
B	浒	浔	洳	涑	浯	涞	涠	浞	涓	涔	浜	浠	浼	浣	渚	淇
C	淅	淞	渎	涿	淠	渑	淦	淝	淙	渖	涫	渌	涮	渫	湮	湎
D	湫	溲	湟	溆	湓	湔	渲	渥	湄	滟	溱	溘	滠	漭	滢	溥
E	溧	溽	溻	溷	滗	溴	滏	溏	滂	溟	潢	潆	潇	漤	漕	滹
F	漯	漶	潋	潴	漪	漉	漩	澉	澍	澌	潸	潲	潼	潺	濑	

E5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		濉	澧	澹	澶	濂	濡	濮	濞	濠	濯	瀚	瀣	瀛	瀹	瀵
B	灏	灞	宀	宄	宕	宓	宥	宸	甯	骞	搴	寤	寮	褰	寰	蹇
C	謇	辶	迓	迕	迥	迮	迤	迩	迦	迳	迨	逅	逄	逋	逦	逑
D	逍	逖	逡	逵	逶	逭	逯	遄	遑	遒	遐	遨	遘	遢	遛	暹
E	遴	遽	邂	邈	邃	邋	彐	彗	彖	彘	尻	咫	屐	屙	孱	屣
F	屦	羼	弪	弩	弭	艴	弼	鬻	屮	妁	妃	妍	妩	妪	妣	

E6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		妗	姊	妫	妞	妤	姒	妲	妯	姗	妾	娅	娆	姝	娈	姣
B	姘	姹	娌	娉	娲	娴	娑	娣	娓	婀	婧	婊	婕	娼	婢	婵
C	胬	媪	媛	婷	婺	媾	嫫	媲	嫒	嫔	媸	嫠	嫣	嫱	嫖	嫦
D	嫘	嫜	嬉	嬗	嬖	嬲	嬷	孀	尕	尜	孚	孥	孳	孑	孓	孢
E	驵	驷	驸	驺	驿	驽	骀	骁	骅	骈	骊	骐	骒	骓	骖	骘
F	骛	骜	骝	骟	骠	骢	骣	骥	骧	纟	纡	纣	纥	纨	纩	

E7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		纭	纰	纾	绀	绁	绂	绉	绋	绌	绐	绔	绗	绛	绠	绡
B	绨	绫	绮	绯	绱	绲	缍	绶	绺	绻	绾	缁	缂	缃	缇	缈
C	缋	缌	缏	缑	缒	缗	缙	缜	缛	缟	缡	缢	缣	缤	缥	缦
D	缧	缪	缫	缬	缭	缯	缰	缱	缲	缳	缵	幺	畿	巛	甾	邕
E	玎	玑	玮	玢	玟	珏	珂	珑	玷	玳	珀	珉	珈	珥	珙	顼
F	琊	珩	珧	珞	玺	珲	琏	琪	瑛	琦	琥	琨	琰	琮	琬	

E8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		琛	琚	瑁	瑜	瑗	瑕	瑙	瑷	瑭	瑾	璜	璎	璀	璁	璇
B	璋	璞	璨	璩	璐	璧	瓒	璺	韪	韫	韬	杌	杓	杞	杈	杩
C	枥	枇	杪	杳	枘	枧	杵	枨	枞	枭	枋	杷	杼	柰	栉	柘
D	栊	柩	枰	栌	柙	枵	柚	枳	柝	栀	柃	枸	柢	栎	柁	柽
E	栲	栳	桠	桡	桎	桢	桄	桤	梃	栝	桕	桦	桁	桧	桀	栾
F	桊	桉	栩	梵	梏	桴	桷	梓	桫	棂	楮	棼	椟	椠	棹	

E9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		椤	棰	椋	椁	楗	棣	椐	楱	椹	楠	楂	楝	榄	楫	榀
B	榘	楸	椴	槌	榇	榈	槎	榉	楦	楣	楹	榛	榧	榻	榫	榭
C	槔	榱	槁	槊	槟	榕	槠	榍	槿	樯	槭	樗	樘	橥	槲	橄
D	樾	檠	橐	橛	樵	檎	橹	樽	樨	橘	橼	檑	檐	檩	檗	檫
E	猷	獒	殁	殂	殇	殄	殒	殓	殍	殚	殛	殡	殪	轫	轭	轱
F	轲	轳	轵	轶	轸	轷	轹	轺	轼	轾	辁	辂	辄	辇	辋	

EA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		辍	辎	辏	辘	辚	軎	戋	戗	戛	戟	戢	戡	戥	戤	戬
B	臧	瓯	瓴	瓿	甏	甑	甓	攴	旮	旯	旰	昊	昙	杲	昃	昕
C	昀	炅	曷	昝	昴	昱	昶	昵	耆	晟	晔	晁	晏	晖	晡	晗
D	晷	暄	暌	暧	暝	暾	曛	曜	曦	曩	贲	贳	贶	贻	贽	赀
E	赅	赆	赈	赉	赇	赍	赕	赙	觇	觊	觋	觌	觎	觏	觐	觑
F	牮	犟	牝	牦	牯	牾	牿	犄	犋	犍	犏	犒	挈	挲	掰	

EB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		搿	擘	耄	毪	毳	毽	毵	毹	氅	氇	氆	氍	氕	氘	氙
B	氚	氡	氩	氤	氪	氲	攵	敕	敫	牍	牒	牖	爰	虢	刖	肟
C	肜	肓	肼	朊	肽	肱	肫	肭	肴	肷	胧	胨	胩	胪	胛	胂
D	胄	胙	胍	胗	朐	胝	胫	胱	胴	胭	脍	脎	胲	胼	朕	脒
E	豚	脶	脞	脬	脘	脲	腈	腌	腓	腴	腙	腚	腱	腠	腩	腼
F	腽	腭	腧	塍	媵	膈	膂	膑	滕	膣	膪	臌	朦	臊	膻	

EC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		臁	膦	欤	欷	欹	歃	歆	歙	飑	飒	飓	飕	飙	飚	殳
B	彀	毂	觳	斐	齑	斓	於	旆	旄	旃	旌	旎	旒	旖	炀	炜
C	炖	炝	炻	烀	炷	炫	炱	烨	烊	焐	焓	焖	焯	焱	煳	煜
D	煨	煅	煲	煊	煸	煺	熘	熳	熵	熨	熠	燠	燔	燧	燹	爝
E	爨	灬	焘	煦	熹	戾	戽	扃	扈	扉	礻	祀	祆	祉	祛	祜
F	祓	祚	祢	祗	祠	祯	祧	祺	禅	禊	禚	禧	禳	忑	忐	

ED	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		怼	恝	恚	恧	恁	恙	恣	悫	愆	愍	慝	憩	憝	懋	懑
B	戆	肀	聿	沓	泶	淼	矶	矸	砀	砉	砗	砘	砑	斫	砭	砜
C	砝	砹	砺	砻	砟	砼	砥	砬	砣	砩	硎	硭	硖	硗	砦	硐
D	硇	硌	硪	碛	碓	碚	碇	碜	碡	碣	碲	碹	碥	磔	磙	磉
E	磬	磲	礅	磴	礓	礤	礞	礴	龛	黹	黻	黼	盱	眄	眍	盹
F	眇	眈	眚	眢	眙	眭	眦	眵	眸	睐	睑	睇	睃	睚	睨	

EE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		睢	睥	睿	瞍	睽	瞀	瞌	瞑	瞟	瞠	瞰	瞵	瞽	町	畀
B	畎	畋	畈	畛	畲	畹	疃	罘	罡	罟	詈	罨	罴	罱	罹	羁
C	罾	盍	盥	蠲	钅	钆	钇	钋	钊	钌	钍	钏	钐	钔	钗	钕
D	钚	钛	钜	钣	钤	钫	钪	钭	钬	钯	钰	钲	钴	钶	钷	钸
E	钹	钺	钼	钽	钿	铄	铈	铉	铊	铋	铌	铍	铎	铐	铑	铒
F	铕	铖	铗	铙	铘	铛	铞	铟	铠	铢	铤	铥	铧	铨	铪	

EF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		铩	铫	铮	铯	铳	铴	铵	铷	铹	铼	铽	铿	锃	锂	锆
B	锇	锉	锊	锍	锎	锏	锒	锓	锔	锕	锖	锘	锛	锝	锞	锟
C	锢	锪	锫	锩	锬	锱	锲	锴	锶	锷	锸	锼	锾	锿	镂	锵
D	镄	镅	镆	镉	镌	镎	镏	镒	镓	镔	镖	镗	镘	镙	镛	镞
E	镟	镝	镡	镢	镤	镥	镦	镧	镨	镩	镪	镫	镬	镯	镱	镲
F	镳	锺	矧	矬	雉	秕	秭	秣	秫	稆	嵇	稃	稂	稞	稔	

F0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		稹	稷	穑	黏	馥	穰	皈	皎	皓	皙	皤	瓞	瓠	甬	鸠
B	鸢	鸨	鸩	鸪	鸫	鸬	鸲	鸱	鸶	鸸	鸷	鸹	鸺	鸾	鹁	鹂
C	鹄	鹆	鹇	鹈	鹉	鹋	鹌	鹎	鹑	鹕	鹗	鹚	鹛	鹜	鹞	鹣
D	鹦	鹧	鹨	鹩	鹪	鹫	鹬	鹱	鹭	鹳	疒	疔	疖	疠	疝	疬
E	疣	疳	疴	疸	痄	疱	疰	痃	痂	痖	痍	痣	痨	痦	痤	痫
F	痧	瘃	痱	痼	痿	瘐	瘀	瘅	瘌	瘗	瘊	瘥	瘘	瘕	瘙	

F1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		瘛	瘼	瘢	瘠	癀	瘭	瘰	瘿	瘵	癃	瘾	瘳	癍	癞	癔
B	癜	癖	癫	癯	翊	竦	穸	穹	窀	窆	窈	窕	窦	窠	窬	窨
C	窭	窳	衤	衩	衲	衽	衿	袂	袢	裆	袷	袼	裉	裢	裎	裣
D	裥	裱	褚	裼	裨	裾	裰	褡	褙	褓	褛	褊	褴	褫	褶	襁
E	襦	襻	疋	胥	皲	皴	矜	耒	耔	耖	耜	耠	耢	耥	耦	耧
F	耩	耨	耱	耋	耵	聃	聆	聍	聒	聩	聱	覃	顸	颀	颃	

F2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		颉	颌	颍	颏	颔	颚	颛	颞	颟	颡	颢	颥	颦	虍	虔
B	虬	虮	虿	虺	虼	虻	蚨	蚍	蚋	蚬	蚝	蚧	蚣	蚪	蚓	蚩
C	蚶	蛄	蚵	蛎	蚰	蚺	蚱	蚯	蛉	蛏	蚴	蛩	蛱	蛲	蛭	蛳
D	蛐	蜓	蛞	蛴	蛟	蛘	蛑	蜃	蜇	蛸	蜈	蜊	蜍	蜉	蜣	蜻
E	蜞	蜥	蜮	蜚	蜾	蝈	蜴	蜱	蜩	蜷	蜿	螂	蜢	蝽	蝾	蝻
F	蝠	蝰	蝌	蝮	螋	蝓	蝣	蝼	蝤	蝙	蝥	螓	螯	螨	蟒	

F3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		蟆	螈	螅	螭	螗	螃	螫	蟥	螬	螵	螳	蟋	蟓	螽	蟑
B	蟀	蟊	蟛	蟪	蟠	蟮	蠖	蠓	蟾	蠊	蠛	蠡	蠹	蠼	缶	罂
C	罄	罅	舐	竺	竽	笈	笃	笄	笕	笊	笫	笏	筇	笸	笪	笙
D	笮	笱	笠	笥	笤	笳	笾	笞	筘	筚	筅	筵	筌	筝	筠	筮
E	筻	筢	筲	筱	箐	箦	箧	箸	箬	箝	箨	箅	箪	箜	箢	箫
F	箴	篑	篁	篌	篝	篚	篥	篦	篪	簌	篾	篼	簏	簖	簋	

F4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		簟	簪	簦	簸	籁	籀	臾	舁	舂	舄	臬	衄	舡	舢	舣
B	舭	舯	舨	舫	舸	舻	舳	舴	舾	艄	艉	艋	艏	艚	艟	艨
C	衾	袅	袈	裘	裟	襞	羝	羟	羧	羯	羰	羲	籼	敉	粑	粝
D	粜	粞	粢	粲	粼	粽	糁	糇	糌	糍	糈	糅	糗	糨	艮	暨
E	羿	翎	翕	翥	翡	翦	翩	翮	翳	糸	絷	綦	綮	繇	纛	麸
F	麴	赳	趄	趔	趑	趱	赧	赭	豇	豉	酊	酐	酎	酏	酤	

F5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		酢	酡	酰	酩	酯	酽	酾	酲	酴	酹	醌	醅	醐	醍	醑
B	醢	醣	醪	醭	醮	醯	醵	醴	醺	豕	鹾	趸	跫	踅	蹙	蹩
C	趵	趿	趼	趺	跄	跖	跗	跚	跞	跎	跏	跛	跆	跬	跷	跸
D	跣	跹	跻	跤	踉	跽	踔	踝	踟	踬	踮	踣	踯	踺	蹀	踹
E	踵	踽	踱	蹉	蹁	蹂	蹑	蹒	蹊	蹰	蹶	蹼	蹯	蹴	躅	躏
F	躔	躐	躜	躞	豸	貂	貊	貅	貘	貔	斛	觖	觞	觚	觜	

F6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		觥	觫	觯	訾	謦	靓	雩	雳	雯	霆	霁	霈	霏	霎	霪
B	霭	霰	霾	龀	龃	龅	龆	龇	龈	龉	龊	龌	黾	鼋	鼍	隹
C	隼	隽	雎	雒	瞿	雠	銎	銮	鋈	錾	鍪	鏊	鎏	鐾	鑫	鱿
D	鲂	鲅	鲆	鲇	鲈	稣	鲋	鲎	鲐	鲑	鲒	鲔	鲕	鲚	鲛	鲞
E	鲟	鲠	鲡	鲢	鲣	鲥	鲦	鲧	鲨	鲩	鲫	鲭	鲮	鲰	鲱	鲲
F	鲳	鲴	鲵	鲶	鲷	鲺	鲻	鲼	鲽	鳄	鳅	鳆	鳇	鳊	鳋	

F7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A		鳌	鳍	鳎	鳏	鳐	鳓	鳔	鳕	鳗	鳘	鳙	鳜	鳝	鳟	鳢
B	靼	鞅	鞑	鞒	鞔	鞯	鞫	鞣	鞲	鞴	骱	骰	骷	鹘	骶	骺
C	骼	髁	髀	髅	髂	髋	髌	髑	魅	魃	魇	魉	魈	魍	魑	飨
D	餍	餮	饕	饔	髟	髡	髦	髯	髫	髻	髭	髹	鬈	鬏	鬓	鬟
E	鬣	麽	麾	縻	麂	麇	麈	麋	麒	鏖	麝	麟	黛	黜	黝	黠
F	黟	黢	黩	黧	黥	黪	黯	鼢	鼬	鼯	鼹	鼷	鼽	鼾	齄	

Appendix E . Font Table - BIG-5 Code

A1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4		，	、	。	·	•	；	：	？	！	：	…	‥	，	、	.
5	·	；	：	？	！	｜	–	｜	—	｜	_	﹏	﹏	（	）	︵
6	﹀	｛	｝	︵	﹀	〔	〕	︵	﹀	【	】	┌	┘	《	》	︽
7	﹀	〈	〉	︿	﹀	「	」	┐	└	『	』	╕	╘	（	）	
A		｛	｝	〔	〕	‘	’	“	”	〝	〞	‵	′	#	&	*
B	※	§	〃	○	●	△	▲	◎	☆	★	◇	◆	□	■	▽	▼
C	㊣	℅	¯	―	＿	﹍	‐	‐	﹉	﹍	﹋	﹋	#	&	*	+
D	－	×	÷	±	√	<	>	＝	≦	≧	≠	∞	≒	≡	+	－
E	<	>	=	～	∩	∪	⊥	∠	∟	⊿	log	ln	∫	∮	∵	∴
F	♀	♂	⊕	⊙	↑	↓	←	→	↖	↗	↙	↘	∥	｜	／	

A2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	＼	／	\	$	￥	〒	¢	£	%	@	℃	℉	$	%	‰	㎜
5	㎜	㎝	㎞	㏎	㎡	㎎	㎏	㏄	°	兙	兛	兝	兡	兣	兤	冓
6	凪	糎	▁	▂	▃	▄	▅	▆	▇	▉	▏	▎	▍	▌	▋	▊
7	▉	┼	┴	┬	┤	├	━	─	│	┃	┌	┐	└	┘	╭	
A		╮	╰	╯	═	╞	╪	╡	◢	◣	◥	◤	／	＼	╳	0
B	1	2	3	4	5	6	7	8	9	Ⅰ	Ⅱ	Ⅲ	Ⅳ	Ⅴ	Ⅵ	Ⅶ
C	Ⅷ	Ⅸ	Ⅹ	〡	〢	〣	乂	〆	一	二	三	文	十	廿	卅	A
D	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
E	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f	g
F	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	

A3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	w	x	y	z	Α	Β	Γ	Δ	Ε	Ζ	Η	Θ	Ι	Κ	Λ	Μ
5	Ν	Ξ	Ο	Π	Ρ	Σ	Τ	Υ	Φ	Χ	Ψ	Ω	α	β	γ	δ
6	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ
7	φ	χ	ψ	ω	ㄅ	ㄆ	ㄇ	ㄈ	ㄉ	ㄊ	ㄋ	ㄌ	ㄍ	ㄎ	ㄏ	
A		ㄐ	ㄑ	ㄒ	ㄓ	ㄔ	ㄕ	ㄖ	ㄗ	ㄘ	ㄙ	ㄚ	ㄛ	ㄜ	ㄝ	ㄞ
B	ㄟ	ㄠ	ㄡ	ㄢ	ㄣ	ㄤ	ㄥ	ㄦ	ㄧ	ㄨ	ㄩ	•		ˊ	ˇ	ˋ
C																
D																
E		€														
F																

A4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	一	乙	丁	七	乃	九	了	二	人	儿	入	八	几	刀	刁	力
5	匕	十	卜	又	三	下	丈	上	丫	丸	凡	久	么	也	乞	于
6	亡	兀	刃	勺	千	叉	口	土	士	夕	大	女	子	孑	孓	寸
7	小	尢	尸	山	川	工	己	已	巳	巾	干	廾	弋	弓	才	
A		丑	丐	不	中	丰	丹	之	尹	予	云	井	互	五	亢	仁
B	什	仃	仆	仇	仍	今	介	仄	元	允	内	六	兮	公	冗	凶
C	分	切	刈	勻	勾	勿	化	匹	午	升	卅	卞	厄	友	及	反
D	壬	天	夫	太	夭	孔	少	尤	尺	屯	巴	幻	廿	弔	引	心
E	戈	户	手	扎	支	文	斗	斤	方	日	曰	月	木	欠	止	歹
F	毋	比	毛	氏	水	火	爪	父	爻	片	牙	牛	犬	王	丙	

A5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	世	丕	且	丘	主	乍	乏	乎	以	付	仔	仕	他	仗	代	令
5	仙	仞	充	兄	冉	冊	冬	凹	出	凸	刊	加	功	包	匆	北
6	匝	仟	半	卉	卡	占	卯	卮	去	可	古	右	召	叮	叩	叨
7	叼	司	叵	叫	另	只	史	叱	台	句	叭	叻	四	囚	外	
A		央	失	奴	奶	孕	它	尼	巨	巧	左	市	布	平	幼	弁
B	弘	弗	必	戊	打	扔	扒	扑	斥	旦	朮	本	未	末	札	正
C	母	民	氐	永	汁	汀	氾	犯	玄	玉	瓜	瓦	甘	生	用	甩
D	田	由	甲	申	疋	白	皮	皿	目	矛	矢	石	示	禾	穴	立
E	丞	丟	乒	乓	乩	亙	交	亦	亥	仿	伉	伙	伊	伕	伍	伐
F	休	伏	仲	件	任	仰	仳	份	企	伋	光	兇	兆	先	全	

A6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	共	再	冰	列	刑	划	刎	刖	劣	匈	匡	匠	印	危	吉	吏
5	同	吊	吐	吁	吋	各	向	名	合	吃	后	吆	吒	因	回	囝
6	圳	地	在	圭	圬	圯	圩	夙	多	夷	夸	妄	奸	妃	好	她
7	如	妁	字	存	宇	守	宅	安	寺	尖	屹	州	帆	并	年	
A		式	弛	忙	忖	戎	戌	戍	成	扣	扛	托	收	早	旨	旬
B	旭	曲	曳	有	朽	朴	朱	朵	次	此	死	氖	汝	汗	汙	江
C	池	汐	汕	污	汎	汍	汎	灰	牟	牝	百	竹	米	糸	缶	羊
D	羽	老	考	而	耒	耳	聿	肉	肋	肌	臣	自	至	臼	舌	舛
E	舟	艮	色	艾	虫	血	行	衣	西	阡	串	亨	位	住	佇	佗
F	佞	伴	佛	何	估	佐	佑	伽	伺	伸	佃	佔	似	但	佣	

Example: The BIG-5 code of “中” is A4, A4, and BIG-5 code of “世” is A5, 40.

A7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	作	你	伯	低	伶	余	佝	佈	佚	兌	克	免	兵	冶	冷	別
5	判	利	刪	刨	劫	助	努	劬	匣	即	卵	吝	吭	吞	吾	否
6	呎	吧	呆	呃	吳	呈	呂	君	吩	告	吹	吻	吸	吮	吵	呐
7	吠	吼	呀	吱	含	吟	听	囪	困	囤	囫	坊	坑	址	坍	
A		均	坎	圾	坐	坏	坼	壯	夾	妝	妒	妨	妞	妣	妙	妖
B	妍	妤	妓	妊	妥	孝	孜	孚	孛	完	宋	宏	尬	局	屁	尿
C	尾	岐	岑	岔	岌	巫	希	序	庇	床	廷	弄	弟	彤	形	彷
D	役	忘	忌	志	忍	忱	快	忸	忪	戒	我	抄	抗	抖	技	扶
E	抉	扭	把	扼	找	批	扳	抒	扯	折	扮	投	抓	抑	抒	改
F	攻	攸	旱	更	束	李	杏	材	村	杜	杖	杞	杉	杆	杠	

A8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	杓	杗	步	每	求	汞	沙	沁	沈	沉	沅	沛	汪	決	沐	汰
5	沌	汨	沖	沒	汽	沃	汲	汾	汴	沆	汶	沍	沔	沘	沂	灶
6	灼	災	炙	牢	牡	牠	狄	狂	玖	甬	甫	男	甸	皂	盯	矣
7	私	秀	禿	究	系	罕	肖	肓	肝	肘	肛	肚	育	良	芒	
A		芋	芍	見	角	言	谷	豆	豕	貝	赤	走	足	身	車	辛
B	辰	迂	迆	迅	迄	巡	邑	邢	邪	邦	那	酉	釆	里	防	阮
C	阱	阪	阬	並	乖	乳	事	些	亞	享	京	佯	依	侍	佳	使
D	佬	供	例	來	侃	佰	併	侈	佩	佻	侖	佾	侏	侑	佺	兔
E	兒	兕	兩	具	其	典	冽	函	刻	券	刷	刺	到	刮	制	剁
F	劾	勗	卒	協	卓	卑	卦	卷	卸	卹	取	叔	受	味	呵	

A9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	咖	呸	咕	咀	呻	呷	咄	咒	咆	呼	咐	呱	呶	和	咚	呢
5	周	咋	命	咎	固	垃	坷	坪	坩	坡	坦	坤	坼	夜	奉	奇
6	奈	奄	奔	妾	妻	委	妹	妮	姑	姆	姐	姍	始	姓	姊	妯
7	妳	姒	姅	孟	孤	季	宗	定	官	宜	宙	宛	尚	屈	居	
A		屆	岷	岡	岸	岩	岫	岱	岳	帘	帚	帖	帕	帛	帑	幸
B	庚	店	府	底	庖	延	弦	弧	弩	往	征	彿	彼	忝	忠	忽
C	念	忿	怏	怔	怯	怵	怖	怪	怕	怡	性	怩	怫	怛	或	戕
D	房	戾	所	承	拉	拌	拄	抿	拂	抹	拒	招	披	拓	拔	拋
E	拈	抨	抽	押	拐	拙	拇	拍	抵	拚	抱	拘	拖	拗	拆	抬
F	拎	放	斧	於	旺	昔	易	昌	昆	昂	明	昀	昏	昕	昊	

AA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	昇	服	朋	杭	枋	枕	東	果	杳	杷	枇	枝	林	杯	杰	板
5	枉	松	析	杵	枚	枓	杼	杪	杲	欣	武	歧	歿	氓	氛	泣
6	注	泳	沱	泌	泥	河	沽	沾	沼	波	沫	法	泓	沸	泄	油
7	況	沮	泗	泅	泱	沿	治	泡	泛	泊	沬	泯	泜	泖	泠	
A		炕	炎	炒	炊	炙	爬	爭	爸	版	牧	物	狀	狎	狙	狗
B	狐	玩	玨	玫	玫	玥	甽	疝	疙	疚	的	盂	盲	直	知	矽
C	社	祀	祁	秉	秈	空	穹	竺	糾	罔	羌	羋	者	肺	肥	肢
D	肱	股	肫	肩	肴	肪	肯	臥	臾	舍	芳	芝	芙	芭	芽	芟
E	芹	花	芬	芥	芯	芸	芣	芰	芾	芷	虎	虱	初	表	軋	迎
F	返	近	邵	邸	邱	邶	采	金	長	門	阜	陀	阿	阻	附	

AB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	陂	隹	雨	青	非	亟	亭	亮	信	侵	侯	便	俠	俑	俏	保
5	促	侶	俘	俟	俊	俗	侮	俐	俄	係	俚	俎	俞	侷	兗	冒
6	冑	冠	剎	剃	削	前	刺	剋	則	勇	勉	勃	勁	匍	南	卻
7	厚	叛	咬	哀	咨	哎	哉	咸	咦	咳	哇	哂	咽	咪	品	
A		哄	哈	咯	咫	咱	咻	咩	唎	唄	囿	垂	型	垠	垣	垢
B	城	垮	垓	奕	契	奏	奎	奐	姜	姘	姿	姣	姨	娃	姥	姪
C	姚	姦	威	姻	孩	宣	宦	室	客	宥	封	屎	屏	屍	屋	峙
D	峒	巷	帝	帥	帘	幽	庠	度	建	弈	弭	彥	很	待	徊	律
E	徇	後	徉	怒	思	怠	急	怎	怨	恍	恰	恨	恢	恆	恃	恬
F	恫	恪	恤	扁	拜	挖	按	拼	拭	持	拮	拽	指	拱	拷	

AC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	拯	括	拾	拴	挑	挂	政	故	斫	施	既	春	昭	映	昧	是
5	星	昨	昱	昤	曷	柿	染	柱	柔	某	柬	架	枯	柵	柩	柯
6	柄	柑	柷	柚	查	枸	柏	柞	柳	枰	柙	柢	柝	柒	歪	殃
7	殆	段	毒	毗	氟	泉	洋	洲	洪	流	津	洌	洱	洞	洗	
A		活	洽	派	洶	洛	泵	洹	洧	洸	洩	洮	洵	洎	洫	炫
B	爲	炳	炬	炯	炭	炸	炮	炤	爰	牲	牯	牴	狩	狼	狡	玷
C	珊	玻	玲	珍	珀	玳	甚	甭	畏	界	畎	畋	疫	疤	疥	疢
D	疣	癸	皆	皇	皈	盈	盆	盃	盅	省	盹	相	眉	看	盾	盼
E	眇	矜	砂	研	砌	砍	祆	祉	祈	祇	禹	禺	科	秒	秋	穿
F	突	竿	竽	籽	紂	紅	紀	紉	紇	約	紆	缸	美	羿	耄	

AD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	耐	耍	耑	耶	胖	胥	胚	胃	胄	背	胡	胛	胎	胞	胤	胝
5	致	舢	苧	范	茅	苣	苛	苦	茄	若	茂	茉	苒	苗	英	茁
6	苜	苔	苑	苞	苓	苟	苯	茆	虐	虹	虻	虺	衍	衫	要	觔
7	計	訂	訃	貞	負	赴	赳	趴	軍	軌	述	迦	迢	迪	迥	
A		迭	迫	迤	迨	郊	郎	郁	郃	酋	酊	重	閂	限	陋	陌
B	降	面	革	韋	韭	音	頁	風	飛	食	首	香	乘	亳	倌	倍
C	倣	俯	倦	倥	俸	倩	倖	倆	值	借	倚	倒	們	俺	倀	倔
D	倨	俱	倡	個	候	倘	俳	修	倭	倪	俾	倫	倉	兼	冤	冥
E	冢	凍	凌	准	凋	剖	剜	剔	剛	剝	匪	卿	原	厝	叟	哨
F	唐	唁	唷	哼	哥	哲	唆	哺	唔	哩	哭	員	唉	哮	哪	

AE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	哦	唧	唇	哽	唏	圃	圄	埂	埔	埋	埃	堉	夏	套	奘	奚
5	娑	娘	娜	娟	娛	娓	姬	娠	娣	娩	娥	娌	娉	孫	屘	宰
6	害	家	宴	宮	宵	容	宸	射	屑	展	屐	峭	峽	峻	峪	峨
7	峰	島	崁	峴	差	席	師	庫	庭	座	弱	徒	徑	徐	恙	
A		恣	恥	恐	恕	恭	恩	息	悄	悟	悚	悍	悔	悌	悅	悖
B	扇	拳	挈	拿	捎	挾	振	捕	捂	捆	捏	捉	挺	捐	挽	挪
C	挫	挨	捍	捌	效	敉	料	旁	旅	時	晉	晏	晃	晒	晌	晅
D	晁	書	朔	朕	朗	校	核	案	框	桓	根	桂	桔	栩	梳	栗
E	桌	桑	栽	柴	桐	桀	格	桃	株	桅	栓	栘	桁	殊	殉	殷
F	氣	氧	氨	氦	氤	泰	浪	涕	消	涇	浦	浸	海	浙	涓	

AF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	浬	涉	浮	浚	浴	浩	涌	涊	浹	涅	浥	涔	烊	烘	烤	烙
5	烈	烏	爹	特	狼	狹	狽	狸	狷	玆	班	琉	珮	珠	珪	珞
6	畔	畝	畜	畚	留	疾	病	症	疲	疳	疽	疼	疹	痂	疸	皋
7	皰	益	盍	盎	眩	真	眠	眨	矩	砰	砧	砸	砝	破	砷	
A		砥	砭	砠	砟	砲	祕	祐	祠	祟	祖	神	祝	祗	祚	秤
B	秣	秧	租	秦	秩	秘	窄	窈	站	笆	笑	粉	紡	紗	紋	紊
C	素	索	純	紐	紕	級	紜	納	紙	紛	缺	罟	羔	翅	翁	耆
D	耘	耕	耙	耗	耽	耿	胱	脂	胰	脅	胭	胴	脆	胸	胳	脈
E	能	脊	胼	胯	臭	臬	舀	舐	航	舫	舨	般	芻	茫	荒	荔
F	荊	茸	荐	草	茵	茴	荏	茲	茹	茶	茗	荀	茱	茨	莖	

B0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	虔	蚊	蚪	蚓	蚤	蚩	蚌	蚣	蚜	衰	衷	袁	袂	衽	衹	記
5	訐	討	訌	訕	訊	託	訓	訖	訏	訑	豈	豺	豹	財	貢	起
6	躬	軒	軔	軏	辱	送	逆	迷	退	迺	迴	逃	追	逅	迸	邕
7	郡	郝	郢	酒	配	酌	釘	針	釗	釜	釙	閃	院	陣	陡	
A		陛	陝	除	陘	陞	隻	飢	馬	骨	高	鬥	鬲	鬼	乾	偺
B	偽	停	假	偃	偌	做	偉	健	偶	偎	偕	偵	側	偷	偏	倏
C	偯	偭	兜	冕	凰	剪	副	勒	務	勘	動	匐	匏	匙	匿	區
D	匾	參	曼	商	啪	啦	啄	啞	啡	啃	啊	唱	啖	問	啕	唯
E	啤	唸	售	啜	唬	啣	唳	啁	啗	圈	國	圉	域	堅	堊	堆
F	埠	埤	基	堂	堵	執	培	夠	奢	娶	婁	婉	婦	婪	婀	

B1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	娼	婢	婚	婆	婊	孰	寇	寅	寄	寂	宿	密	尉	專	將	屠
5	屜	屝	崇	崆	崎	崛	崖	崢	崑	崩	崔	崙	崤	崧	崗	巢
6	常	帶	帳	帷	康	庸	庶	庵	庾	張	強	彗	彬	彩	彫	得
7	徙	從	徘	御	徠	徜	恿	患	悉	悠	您	惋	悴	惦	悽	
A		情	悻	悵	惜	悼	惘	惕	惆	惟	悸	惚	惇	戚	戛	扈
B	掠	控	捲	掖	探	接	捷	捧	掘	措	捱	掩	掉	掃	掛	捫
C	推	掄	授	掙	採	掬	排	掏	掀	捻	捩	捨	捺	敝	敖	救
D	教	敗	啓	敏	敘	敕	敔	斜	斛	斬	族	旋	旌	旎	晝	晚
E	晤	晨	晦	晞	曹	勗	望	梁	梯	梢	梓	梵	桿	桶	梱	梧
F	梗	械	梃	棄	梭	梆	梅	梔	條	梨	棗	椪	椳	欲	殺	

B2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	毫	毬	氫	涎	涼	淳	淙	液	淡	淌	淤	添	淺	清	淇	淋
5	涯	淑	涮	淞	淹	涸	混	淵	淅	淒	渚	涵	淚	淫	淘	淪
6	深	淮	淨	淆	淄	涪	淬	涿	淦	烹	焉	焊	烽	烯	爽	牽
7	犁	猜	猛	猖	猓	猙	率	琅	琊	球	理	現	琍	瓠	瓶	
A		瓷	甜	產	略	畦	畢	異	疏	痔	痕	疵	痊	痍	皎	盔
B	盒	盛	眷	眾	眼	眶	眸	眺	硫	硃	硎	祥	票	祭	移	窒
C	窕	笠	笨	笛	第	符	笙	笞	笮	粒	粗	粕	絆	絃	統	紮
D	紹	紼	絀	細	紳	組	累	紿	紲	紱	缽	羞	羚	翌	翎	習
E	耜	聊	聆	脯	脖	脣	脫	脩	脰	脤	舂	舵	舷	舶	船	莎
F	莞	莘	荸	莢	莖	莽	莫	莒	莊	莓	莉	莠	荷	荻	荼	

B3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	莆	莧	處	彪	蛇	蛀	蚶	蛄	蚵	蛆	蛋	蚱	蚯	蛉	術	袞
5	袈	被	袒	袖	袍	袋	覓	規	訪	訝	訣	訥	許	設	訟	訛
6	訢	豉	豚	販	責	貫	貨	貪	貧	赧	赦	趾	趺	軛	軟	這
7	逍	通	逗	連	速	逝	逐	逕	逞	造	透	逢	逖	逛	途	
A		部	郭	都	酗	野	釵	釦	釣	釧	釭	釩	閉	陪	陵	陳
B	陸	陰	陴	陶	陷	陬	雀	雪	雩	章	竟	頂	頃	魚	鳥	鹵
C	鹿	麥	麻	傢	傍	傳	備	傑	傀	傖	傘	傚	最	凱	割	剴
D	創	剩	勞	勝	勛	博	厥	啻	喀	喧	啼	喊	喝	喘	喂	喜
E	喪	喔	喇	喋	喃	喳	單	喟	啐	喲	喚	喻	喬	喱	啾	喉
F	喫	喙	圍	堯	堪	場	堤	堰	報	堡	堝	堠	壹	壺	奠	

B4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	婷	媚	婿	媒	媛	媧	孳	孱	寒	富	寓	寐	尊	尋	就	嵌
5	嵐	崴	嵇	巽	幅	帽	幀	幃	幾	廊	廁	廂	廄	弼	彭	復
6	循	徨	惑	惡	悲	悶	惠	愜	愣	惺	愕	惰	惻	惴	慨	惱
7	愎	惶	愉	愀	愒	戟	扉	掣	掌	描	揀	揩	揉	揆	揍	
A		插	揣	提	握	揖	揭	揮	捶	援	揪	換	摒	揚	揹	敞
B	敦	敢	散	斑	斐	斯	普	晰	晴	晶	景	暑	智	晾	晷	曾
C	替	期	朝	棺	棕	棠	棘	棗	椅	棟	棵	森	棧	棹	棒	棲
D	棣	棋	棍	植	椒	椎	棉	棚	楮	棻	款	欺	欽	殘	殖	殼
E	毯	氮	氯	氬	港	游	湔	渡	渲	湧	湊	渠	渥	渣	減	湛
F	湘	渤	湖	湮	渭	渦	湯	渴	湍	渺	測	湃	渝	渾	滋	

B5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	溉	渙	湎	潛	湄	溼	湩	湟	焙	焚	焦	焰	無	然	煮	焜
5	牌	犄	犀	猶	猥	猴	猩	琺	琪	琳	琢	琥	琵	琶	琴	琯
6	琛	琦	琨	甥	甦	畫	番	痢	痛	痣	痙	痘	痞	痠	登	發
7	皖	皓	皴	盜	睏	短	硝	硬	硯	稍	稈	程	稅	稀	窘	
A		窗	窖	童	竣	等	策	筆	筐	筒	答	筍	筋	筏	筑	粟
B	粥	絞	結	絨	絕	紫	絮	絲	絡	給	絢	絰	絳	善	翔	翕
C	耋	聒	肅	腕	腔	腋	腑	腎	脹	腆	脾	腌	腓	腴	舒	舜
D	菩	萃	菸	萍	菠	菅	萋	菁	華	菱	菴	著	萊	菰	萌	菌
E	菽	菲	菊	萸	萎	萄	菜	萇	菔	菟	虛	蛟	蛙	蛭	蛔	蛛
F	蛤	蛐	蛞	街	裁	裂	袱	覃	視	註	詠	評	詞	証	詁	

B6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	詔	詛	詐	詆	訴	診	訶	詖	象	貂	貯	貼	貳	貽	賁	費
5	賀	貴	買	貶	貿	貸	越	超	趁	跎	距	跋	跚	跑	跌	跛
6	跆	軻	軸	軼	辜	逮	逵	週	逸	進	逶	鄂	郵	鄉	郾	酣
7	酥	量	鈔	鈕	鈣	鈉	鈞	鈍	鈐	鈇	鈑	閔	閏	開	閑	
A		間	閒	閎	隊	階	隋	陽	隅	隆	隍	陲	隄	雁	雅	雄
B	集	雇	雯	雲	韌	項	順	須	飧	飪	飯	飩	飲	飭	馮	馭
C	黃	黍	黑	亂	傭	債	傲	傳	僅	傾	催	傷	傻	傯	僇	剿
D	剷	剽	募	勦	勤	勢	勣	匯	嗟	嗨	嗓	嗦	嗎	嗜	嗇	嗑
E	嗣	嗤	嗯	嗚	嗡	嗅	嗆	嗥	嗉	園	圓	塞	塑	塘	塗	塚
F	塔	填	塌	塭	塊	塢	塒	塋	奧	嫁	嫉	嫌	媾	媽	媼	

B7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	媳	婦	媲	嵩	嵯	幌	幹	廉	廈	弒	彙	徬	微	愚	意	慈
5	感	想	愛	惹	愁	愈	慎	慌	慄	溫	愾	愴	愧	慇	慾	愷
6	戡	戢	搓	搾	搞	搪	搭	搽	搬	搏	搜	搔	損	搶	搖	搗
7	搆	敬	斟	新	暗	暉	暇	暈	暖	暄	暘	暍	會	榔	業	
A		楚	楷	楠	楔	極	椰	概	楊	楨	楫	楞	楓	楹	榆	楝
B	楣	楛	歇	歲	毀	殿	毓	毽	溢	溯	滓	溶	滂	源	溝	滇
C	滅	溥	溘	溼	溺	溫	滑	準	溜	滄	滔	溪	溧	溴	煎	煙
D	煩	煤	煉	照	煜	煬	煦	煌	煥	煞	煆	煨	煖	爺	牒	猷
E	獅	猿	猾	瑯	瑚	瑕	瑟	瑞	瑁	琿	瑙	瑛	瑜	當	畸	瘀
F	痰	瘁	痲	痱	痺	痿	痴	痳	盞	盟	睛	睫	睦	睞	督	

B8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	睹	睪	睬	睜	睥	睨	睢	矮	碎	碰	碗	碘	碌	碉	硼	碑
5	碓	碚	祺	祿	禁	萬	禽	稜	稚	稠	稔	稟	稞	窟	窠	筷
6	節	筠	筮	筧	粱	粳	粵	經	絹	綑	綁	綏	絛	置	罩	罪
7	署	義	羨	群	聖	聘	肆	肄	腱	腰	腸	腥	腮	腳	腫	
A		腹	腺	腦	舅	艇	蒂	葷	落	萱	葵	葦	葫	葉	葬	葛
B	萼	萵	葡	董	葩	葭	葆	虞	虜	號	蛹	蜓	蜈	蜇	蜀	蛾
C	蛻	蜂	蜃	蜆	蜊	衙	裟	裔	裙	補	裘	裝	裡	裊	裕	裒
D	覜	解	詫	該	詳	試	詩	詰	誇	詼	詣	誠	話	誅	詭	詢
E	詮	詬	詹	諂	詻	談	豢	貊	貉	賊	資	賈	賄	貲	賃	賂
F	賅	跡	跟	跨	路	跳	踩	跪	跤	跦	躲	較	載	軾	輊	

B9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	辟	農	運	遊	道	遂	達	逼	違	遐	遇	遏	過	遍	遑	逾
5	遁	鄒	鄗	酬	酪	酩	釉	鈷	鉗	鈸	鈽	鉀	鈾	鉛	鉋	鉤
6	鉑	鈴	鉉	鉍	鉅	鈹	鈿	鉚	閘	隘	隔	隕	雍	雋	雉	雊
7	雷	電	雹	零	靖	靴	靶	預	頑	頓	頊	頒	頌	飼	飴	
A		飽	飾	馳	馱	馴	髡	鳩	麂	鼎	鼓	鼠	僧	僮	僥	僖
B	僭	僚	僕	像	僑	僱	僎	僩	競	竦	劃	劂	匱	厭	嗾	嘀
C	嘛	嘗	嗽	嘔	嘆	嘉	嘍	嘎	嗷	嘖	嘟	嘈	嘐	嘩	團	圖
D	塵	塾	境	墓	墊	塹	墅	塽	壽	夥	夢	夤	奪	奩	嫡	嫦
E	嫩	嫗	嫖	嫘	嫣	孵	寞	寧	寡	寥	實	寨	寢	寤	察	對
F	屢	嶄	嶇	幛	幣	幕	幗	幔	廓	廖	弊	彆	彰	徹	慇	

BA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	愿	態	慷	慢	慣	慟	慚	慘	慵	截	撇	摘	摔	撤	摸	摟
5	摺	摑	摧	搴	摭	摻	敲	斡	旗	旖	暢	暨	暝	榜	榨	榕
6	槁	榮	槓	構	榛	榷	榻	槔	榴	槐	槍	榭	槌	榦	槃	榣
7	歉	歌	氳	漳	演	滾	漓	滴	漩	漾	漠	漬	漏	漂	漢	
A		滿	滯	漆	漱	漸	漲	漣	漕	漫	漯	澈	澍	澐	漁	滲
B	滌	滷	熔	熙	煽	熊	熄	熒	爾	犒	犖	獄	獐	瑤	瑣	瑪
C	瑰	瑭	甄	疑	瘧	瘍	瘋	瘉	瘓	盡	監	瞄	瞭	睿	睡	磁
D	碟	碧	碳	碩	碣	禎	福	禍	種	稱	窪	窩	竭	端	管	箕
E	箋	箸	算	箝	箔	箏	箸	箇	箄	粹	粽	精	綻	綰	綜	綽
F	綾	綠	緊	綴	網	綱	綺	綢	綿	綵	綸	維	緒	緇	綬	

BB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	罰	翠	翡	翟	聞	聚	肇	腐	膀	膏	膈	膊	腿	膂	臧	臺
5	與	舔	舞	艋	蓉	蒿	蓆	蓄	蒙	蒞	蒲	蒜	蓋	蒸	蓀	蓓
6	蒐	蒼	蓑	蓊	蜿	蜜	蜻	蜢	蜥	蜴	蜘	蝕	蜷	蜩	裳	褂
7	裴	裹	裸	製	裨	褚	裯	誦	誌	語	誣	認	誡	誓	誤	
A		說	誥	誨	誘	誑	誚	誧	豪	貍	貌	賓	賑	賒	赫	趙
B	趕	跼	輔	輒	輕	輓	辣	遠	遘	遜	遣	遙	遞	遢	遝	遛
C	鄙	鄘	鄞	酵	酸	酷	酴	鉸	銀	銅	銘	銖	鉻	銓	銜	銨
D	鉼	銑	閡	閨	閩	閣	閥	閤	隙	障	際	雌	雒	需	靼	鞅
E	韶	頗	領	颯	颱	餃	餅	餌	餉	駁	骯	骰	髦	魁	魂	鳴
F	鳶	鳳	麼	鼻	齊	億	儀	僻	僵	價	儂	儈	儉	儅	凜	

BC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	劇	劈	劉	劍	劊	勰	厲	嘮	嘻	嘹	嘲	嘿	嘴	嘩	噓	噎
5	噗	噴	嘶	嘯	嘰	墀	墟	增	墳	墜	墮	墩	墦	奭	嬉	嫻
6	嬋	嫵	嬌	嬈	寮	寬	審	寫	層	履	嶝	嶔	幢	幟	幡	廢
7	廚	廟	廝	廣	廠	彈	影	德	徵	慶	慧	慮	慝	慕	憂	
A		慼	慰	慫	慾	憧	憐	憫	憎	憬	憚	憤	憔	憮	戮	摩
B	摯	摹	撞	撲	撈	撐	撰	撥	撓	撕	撩	撒	撮	播	撫	撚
C	撬	撙	撢	撳	敵	敷	數	暮	暫	暴	暱	樣	樟	槨	樁	樞
D	標	槽	模	樓	樊	槳	樂	樅	槭	樑	歐	歎	殤	毅	毆	漿
E	潼	澄	潑	潦	潔	澆	潭	潛	潸	潮	澎	潺	潰	潤	澗	潘
F	滕	滔	澠	潟	熟	熬	熱	熨	牖	犛	獎	獗	瑩	璋	璃	

BD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	瑾	璀	畿	瘠	瘩	瘟	瘤	瘦	瘡	瘢	皚	皺	盤	瞎	瞇	瞌
5	瞑	瞋	磋	磅	確	磊	碾	磕	碼	磐	稿	稼	穀	稽	稷	稻
6	窯	窮	箭	箱	範	箴	篆	篇	篁	箠	篌	糊	締	練	緯	緻
7	緘	緬	緝	編	緣	線	緞	緩	綞	緙	緲	緹	罵	罷	羯	
A		翩	耦	膛	膜	膝	膠	膚	膘	蔗	蔽	蔚	蓮	蔬	蔭	蔓
B	蔑	蔣	蔡	蔔	蓬	蔥	蓿	蔆	螂	蝴	蝶	蝠	蝦	蝸	蝨	蝙
C	蝗	蝌	蝓	衛	衝	褐	複	褒	褓	褕	褊	誼	諒	談	諄	誕
D	請	諸	課	諉	諂	調	誰	論	諍	誶	誹	諛	豌	豎	豬	賠
E	賞	賦	賤	賬	賭	賢	賣	賜	質	賡	赭	趟	趣	踫	踐	踝
F	踢	踏	踩	踟	踡	踞	躬	輝	輛	輟	輩	輦	輪	輜	輞	

BE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	輥	適	遮	遨	遭	遷	鄰	鄭	鄧	鄱	醇	醉	醋	醃	鋅	銻
5	銷	鋪	銬	鋤	鋁	銳	銼	鋒	鋇	鋰	銲	閭	閱	霄	霆	震
6	霉	靠	鞍	鞋	鞏	頡	頫	頜	颳	養	餓	餒	餘	駝	駐	駟
7	駛	騖	駕	駒	駙	骷	髮	髯	鬧	魅	魄	魷	魯	鴆	鴉	
A		鴃	麩	麾	黎	墨	齒	儒	儘	儔	儐	儕	冀	冪	凝	劑
B	劓	勳	噙	噫	噹	噩	噤	噸	噪	器	噥	噱	噯	噬	噢	噶
C	壁	墾	壇	壅	奮	嬝	嬴	學	寰	導	彊	憲	憑	憩	憊	懍
D	憶	憾	懊	懈	戰	擅	擁	擋	撻	撼	據	擄	擇	擂	操	撿
E	擒	擔	撾	整	曆	曉	暹	曄	曇	暸	樽	樸	樺	橙	橫	橘
F	樹	橄	橢	橡	橋	橇	樵	機	橈	歙	歷	氅	濂	澱	澡	

BF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	濃	澤	濁	澧	澳	激	澹	澶	澦	澠	澴	熾	燉	燐	燒	燈
5	燕	熹	燎	燙	燜	燃	燄	獨	璜	璣	璘	璟	璞	瓢	甌	甍
6	瘴	瘸	瘺	盧	盥	瞠	瞞	瞟	瞥	磨	磚	磬	磧	禦	積	穎
7	穆	穌	穋	窺	窩	簑	築	篤	篛	篡	篩	篦	糕	糖	縊	
A		縑	縈	縛	縣	縞	縝	縉	縐	罹	羲	翰	翱	翮	耨	膳
B	膩	膨	臻	興	艘	艙	蕊	蕙	蕈	蕨	蕩	蕃	蕉	蕭	蕪	蕞
C	螃	螟	螞	螢	融	衡	褪	褲	褥	褫	褡	親	覦	諦	諺	諫
D	諱	謀	諜	諧	諮	諾	謁	謂	諷	諭	諳	諶	諼	豫	豭	貓
E	賴	蹄	踱	踴	蹂	踹	踵	輻	輯	輸	輳	辨	辦	遵	遴	選
F	遲	遼	遺	鄴	醒	錠	錶	鋸	錳	錯	錢	鋼	錫	錄	錚	

C0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	錐	錦	錡	錕	錮	錙	閻	隧	隨	險	雕	霎	霑	霖	霍	霓
5	霏	靛	靜	靦	鞘	頰	頸	頻	頷	頭	頹	頤	餐	館	餞	餛
6	餡	餚	駭	駢	駱	骸	骼	髻	髭	鬨	鮑	鴕	鴣	鴦	鴨	鴿
7	鴛	默	黔	龍	龜	優	償	儡	儲	勵	嚎	嚀	嚐	嚅	嚇	
A		嚏	壕	壓	壑	壎	嬰	嬪	嬤	孺	尷	屨	嶼	嶺	嶽	嶸
B	幫	彌	徽	應	懂	懇	懦	懋	戲	戴	擎	擊	擘	擠	擰	擦
C	擬	擱	擢	擭	斂	斃	曙	曖	檀	檔	檄	檢	檜	櫛	檣	橾
D	檗	檐	檠	歜	殮	毚	氈	濘	濱	濟	濠	濛	濤	濫	濯	濉
E	濬	濡	濩	濕	濮	濰	燧	營	燮	燦	燥	燭	燬	燴	燠	爵
F	牆	獰	獲	璩	環	璦	璨	癆	療	癌	盪	瞳	瞪	瞰	瞬	

C1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	瞧	瞭	矯	磷	磺	磴	磯	礁	禧	禪	穗	窿	簇	簍	篾	篷
5	簌	篠	糠	糜	糞	糢	糟	糙	糝	縮	績	繆	縷	縲	繃	縫
6	總	縱	繅	繁	縴	縹	繈	縵	縿	繽	罄	翳	翼	聱	聲	聰
7	聯	聳	臆	臃	膺	臂	臀	膿	膽	臉	膾	臨	舉	艱	薪	
A		薄	蕾	薜	薑	薔	薯	薛	薇	薨	薊	虧	蟀	蟑	螳	蟒
B	蟆	螯	螻	螺	蟈	蟋	褻	褶	襄	褸	褽	覬	謎	謗	謙	講
C	謊	謠	謝	謄	謐	豁	谿	豳	賺	賽	購	賸	賻	趨	蹉	蹋
D	蹈	蹊	轄	輾	轂	轅	輿	避	遽	還	邁	邂	邀	鄹	醣	醞
E	醜	鍍	鎂	錨	鍵	鍊	鍥	鍋	錘	鍾	鍬	鍛	鍰	鍚	鍔	闊
F	闋	闌	闈	闆	隱	隸	雖	霜	霞	鞠	韓	顆	颶	餵	騁	

C2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	駿	鮮	鮫	鮪	鮭	鴻	鴿	麋	黏	點	黜	黝	黛	鼾	齋	叢
5	嚕	嚮	壙	壘	嬸	彝	懣	戳	擴	擲	擾	攆	擺	擻	擷	斷
6	曜	朦	檳	檬	櫃	檻	檸	櫂	檮	檯	歟	歸	殯	瀉	瀋	濾
7	瀆	濺	瀑	瀏	燻	燼	燾	燸	獷	獵	璧	璿	甕	癖	癘	
A		癒	皚	皛	瞻	瞼	礎	禮	穡	穢	穠	竄	竅	簫	簧	簪
B	簞	簣	簡	糧	織	繕	繞	繚	繡	繪	繙	罈	翹	翻	職	聶
C	臍	臏	舊	藏	薩	藍	藐	藉	薰	薺	臺	薦	蟯	蟬	蟲	蟠
D	覆	覲	觴	謨	謹	謬	謫	豐	贅	蹙	蹣	蹦	蹤	蹟	蹕	軀
E	轉	轍	邇	邃	邈	醫	醬	釐	鎔	鎊	鎖	鎢	鏤	鎮	鎬	鎰
F	鍋	鎚	鎗	闔	闖	闐	闕	離	雜	雙	雛	雞	霤	鞣	鞦	

C3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	鞭	韹	額	顏	題	顎	顯	颺	餾	餿	餽	餮	馥	騎	髁	鬃
5	鬆	魏	魎	魍	鯊	鯉	鯽	鯈	鯀	鵑	鵝	鵠	黠	鼙	鼬	儳
6	嚥	壞	壟	壢	寵	龐	廬	懲	懷	懶	懵	攀	攏	曠	曝	櫥
7	櫝	欄	櫓	灝	瀟	瀨	瀚	瀝	瀕	瀘	爆	爍	牘	犢	獸	
A		獺	璽	瓊	瓣	疇	疆	癟	癡	矇	礙	禱	穫	穩	簾	簿
B	簸	簽	簷	籀	繫	繭	繹	繩	繪	羅	繳	羶	羹	贏	臘	藩
C	藝	藪	藕	藤	藥	藷	蟻	蠅	蠍	蟹	蟾	襠	襟	襖	襞	譁
D	譜	識	證	譚	譎	譏	譆	譙	贈	贊	蹼	蹲	躇	蹶	蹬	蹺
E	蹴	轔	轎	辭	邊	邋	醱	醮	鏡	鏑	鏟	鏃	鏈	鏜	鏝	鏖
F	鏢	鏍	鏘	鏤	鏗	鏨	關	隴	難	霪	霧	靡	韜	韻	類	

C4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	願	顛	颼	饅	饉	騖	騙	鬍	鯨	鯧	鯖	鯛	鶉	鵡	鵲	鵪
5	鵬	麒	麗	麓	麴	勸	嚨	嚷	嚶	嚴	嚼	壤	孀	孃	孽	寶
6	巉	懸	懺	攘	攔	攙	曦	朧	櫬	瀾	瀰	瀲	爐	獻	瓏	癢
7	癥	礦	礪	礬	礫	竇	競	籌	籃	籍	糯	糰	辮	繽	繼	
A		纂	罌	耀	臚	艦	藻	藹	蘑	藺	蘆	蘋	蘇	蘊	蠔	蠕
B	襤	覺	觸	議	譬	警	譯	譟	譫	贏	贍	躉	躁	躅	躂	醴
C	釋	鐘	鐃	鏽	闡	霰	飄	饒	饑	馨	騫	騰	騷	騙	鰓	鰍
D	鹹	麵	黨	鼯	齟	齣	齡	儷	儸	囁	囀	囂	夔	屬	巍	懼
E	懾	攝	攜	斕	曩	櫻	欄	櫺	殲	灌	爛	犧	瓖	瓔	癩	矓
F	籐	纏	續	羼	蘗	蘭	蘚	蠣	蠢	蠡	蠟	襪	襬	覽	譜	

C5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	護	譽	贓	躊	躍	躋	轟	辯	醺	鐮	鐳	鐵	鐺	鐸	鐲	鐫
5	闢	霸	霹	露	響	顧	顥	饗	驅	驃	驀	騾	髏	魔	魑	鰭
6	鰥	鶯	鶴	鷂	鶸	麝	黯	鼙	齜	齦	齧	儼	儻	囈	囊	囉
7	孿	巔	巒	彎	懿	攤	權	歡	灑	灘	玀	瓤	疊	癮	癬	
A		禳	籠	籟	聾	聽	臟	襲	襯	觼	讀	贖	贗	躑	躓	轡
B	酈	鑄	鑑	鑒	霽	霾	韃	韁	顫	饕	驕	驍	髒	鬚	鱉	鰱
C	鰾	鰻	鷓	鷗	鼴	齬	齪	龔	囌	巖	戀	攣	攫	攪	曬	欐
D	瓚	竊	籤	籣	籥	纓	纖	纔	臢	蘸	蘿	蠱	變	邐	邏	鑣
E	鑠	鑤	靨	顯	饜	驚	驛	驗	髓	體	髑	鱔	鱗	鱖	鷥	麟
F	黴	囑	壩	攬	灞	癱	癲	矗	罐	羈	蠶	蠹	衢	讓	讒	

C6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	讖	豔	贛	釀	鑪	靂	靈	靄	韆	顰	驟	鬢	魘	鱟	鷹	鷺
5	鹼	鹽	鼇	齷	齲	廳	欖	灣	籬	籮	蠻	觀	躡	釁	鑲	鑰
6	顱	饞	髖	鬣	黌	灤	矚	讚	鑷	韉	驢	驥	纜	讜	躪	釅
7	鑽	鑾	鑼	鱷	鱸	黷	豔	鑿	鸚	爨	驪	鬱	鸛	鸞	籲	
A		①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	⑴	⑵	⑶	⑷	⑸
B	⑹	⑺	⑻	⑼	⑽											丶
C	丿	亅	亠	冂	冖	冫	勹	匚	卩	厶	夊	宀	巛	幺	广	廴
D	彐	彡	攴	无	疒	癶	辵	隶	¨	ˆ	ヽ	ヾ	ゝ	ゞ	〃	仝
E	々	〆	〇	ー	［	］	✽	ぁ	あ	ぃ	い	ぅ	う	ぇ	え	ぉ
F	お	か	が	き	ぎ	く	ぐ	け	げ	こ	ご	さ	ざ	し	じ	

C7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	す	ず	せ	ぜ	そ	ぞ	た	だ	ち	ぢ	っ	つ	づ	て	で	と
5	ど	な	に	ぬ	ね	の	は	ば	ぱ	ひ	び	ぴ	ふ	ぶ	ぷ	へ
6	べ	ぺ	ほ	ぼ	ぽ	ま	み	む	め	も	ゃ	や	ゅ	ゆ	ょ	よ
7	ら	り	る	れ	ろ	ゎ	わ	ゐ	ゑ	を	ん	ァ	ア	ィ	イ	
A		ゥ	ウ	ェ	エ	ォ	オ	カ	ガ	キ	ギ	ク	グ	ケ	ゲ	コ
B	ゴ	サ	ザ	シ	ジ	ス	ズ	セ	ゼ	ソ	ゾ	タ	ダ	チ	ヂ	ッ
C	ツ	ヅ	テ	デ	ト	ド	ナ	ニ	ヌ	ネ	ノ	ハ	バ	パ	ヒ	ビ
D	ピ	フ	ブ	プ	ヘ	ベ	ペ	ホ	ボ	ポ	マ	ミ	ム	メ	モ	ャ
E	ヤ	ュ	ユ	ョ	ヨ	ラ	リ	ル	レ	ロ	ヮ	ワ	ヰ	ヱ	ヲ	ン
F	ヴ	ヵ	ヶ	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	

C8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
5	Ы	Ь	Э	Ю	Я	а	б	в	г	д	е	ё	ж	з	и	й
6	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
7	ъ	ы	ь	э	ю	я	⇧	↸	↹	㇏	𠃌	乚	𠂊	𠂆	亠	
A																
B																
C														￢	￤	＇
D	＂	㈱	№	℡												
E																
F																

C9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	乂	乜	凵	匚	厂	万	丌	乇	亍	囗	兀	屮	彳	丏	冇	与
5	丮	亓	仂	仉	仈	冘	勼	卬	厹	圠	夃	夬	尐	巿	旡	殳
6	毌	气	爿	丱	丼	仨	仜	仩	仡	仝	仚	刌	匜	卌	圢	圣
7	夗	夯	宁	宄	尒	尻	屴	屳	帄	庀	庂	忉	戉	扐	氕	
A		氶	汃	氿	氻	犮	犰	玊	禸	肊	阞	伎	优	伬	仵	伔
B	仱	伀	价	伈	伝	伂	伅	伢	伓	伄	仴	伒	冱	刓	刉	刐
C	劦	匢	匟	卍	厊	吇	囡	囟	圮	圪	圴	夼	妀	奼	妅	奻
D	奾	奷	奿	孖	尕	尥	屼	屺	屻	屾	巟	幵	庄	异	弚	彴
E	忕	忔	忏	扜	扞	扤	扡	扦	扢	扙	扠	扚	扥	旯	旮	朾
F	朹	朸	朻	机	朿	朼	朳	氘	汆	汒	汜	汏	汊	汔	汋	

CA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	汌	灱	牞	犴	犵	玎	甪	癿	穵	网	艸	艼	芀	艽	艿	虍
5	襾	邙	邗	邘	邛	邔	阢	阤	阠	阣	佖	伻	佢	佉	体	佤
6	伾	佧	佒	佟	佁	佘	伭	伳	伿	佡	冏	冹	刜	刞	刡	劭
7	劮	匉	卣	卲	厎	厏	吰	吷	吪	呔	呅	吙	吜	吥	吘	
A		吽	呏	呁	吨	吤	呇	囮	囧	囥	坁	坅	坌	坉	坋	坒
B	夆	奀	妦	妘	妠	妗	妎	妢	妐	妏	妧	妡	宎	宒	尨	尪
C	岍	岏	岈	岋	岉	岒	岊	岆	岓	岕	巠	帊	帎	庋	庉	庌
D	庈	庍	弅	弝	彸	彶	忒	忑	忐	忭	忨	忮	忳	忡	忤	忣
E	忺	忯	忷	忻	怀	忴	戺	抃	抌	抎	抏	抔	抇	扱	扻	扺
F	扰	抁	抈	扷	扽	扲	扴	攷	旰	旴	旳	旲	旵	杅	杇	

CB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	杙	杕	杌	杈	杝	杍	杚	杋	毐	氙	氚	汸	汧	汫	沄	沋
5	沏	汱	汯	汩	沚	汭	沇	沕	沜	汦	汳	汥	汻	沎	灴	灺
6	牣	犿	犽	狃	狆	狁	犺	狅	玕	玗	玓	玔	玒	町	甹	疔
7	疕	皁	礽	耴	肕	肙	肐	肒	肜	芐	芏	芅	芎	芑	芓	
A		芊	芃	芄	豸	迉	辿	邟	邡	邥	邞	邧	邠	阰	阨	阯
B	阭	丳	侘	佼	侅	佽	侀	侇	佶	佴	侉	侄	佷	佌	侗	佪
C	侚	佹	侁	佸	侐	侜	侔	侞	侒	侂	侕	佫	佮	冞	冼	冾
D	刵	刲	刳	剆	刱	劼	匊	匋	匼	厒	厔	咇	呿	咁	咑	咂
E	咈	呫	呺	呾	呥	呬	呴	呦	咍	呯	呡	呠	咘	呣	呧	呤
F	囷	囹	坯	坲	坭	坫	坱	坰	坶	垀	坵	坻	坳	坴	坢	

CC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	坨	坽	夌	奅	姙	姀	姏	姎	妲	姌	姁	妶	妼	姃	姖	妱
5	妽	妺	姈	妿	妴	孢	孥	宓	宕	屄	屆	岮	岤	岠	岵	岯
6	岨	岬	岟	峋	岺	岢	岪	岧	岸	岥	岶	岰	岦	帗	帔	帙
7	弨	弢	弣	弤	彔	徂	彾	彽	忞	忥	怭	怦	怙	怲	怋	
A		怴	怊	怗	怳	怚	怞	怬	怢	怍	怐	怮	怓	怑	怌	怉
B	怜	戔	戽	抭	抴	拑	抾	抪	抶	拊	抮	抳	抯	抻	抩	抰
C	抸	攽	斨	斻	昉	旼	昄	昒	昈	旻	昃	昋	昍	昅	旽	昑
D	昐	曶	朊	枅	杬	枎	枒	杶	杻	枘	枆	构	杴	枍	枌	杺
E	枟	枑	枙	杓	杽	极	杸	杹	枔	欥	殀	歾	毞	氝	沓	泬
F	泫	泮	泙	沶	泔	沭	泧	沷	泐	泂	沺	泃	泆	泭	泲	

CD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	沠	泝	沴	炏	沝	杼	泞	泀	洰	泍	泇	沰	泹	泏	泩	泑
5	炔	炘	炅	炓	炆	炄	炑	炖	炂	炚	炃	牪	狖	狋	狘	狉
6	狜	狒	狔	狚	狌	狑	玤	玡	玭	玦	玢	玠	玬	玝	瓝	瓨
7	甿	畀	甾	疌	疘	皯	盳	盱	盰	盵	矸	矼	矹	矻	矺	
A		矷	祂	礿	秅	穸	穻	竻	籵	糽	耵	肏	肮	肣	肸	肵
B	肭	舠	芠	苀	芫	芚	芘	芛	芵	芧	芮	芼	芞	芺	芴	芨
C	芡	芩	苂	芤	芰	芶	芢	虰	虯	虭	虮	豖	迒	迋	迓	迍
D	迖	迕	迗	邲	邴	邯	邳	邰	阹	阽	阼	阺	陃	俍	俅	俓
E	侲	俉	俋	俁	俔	俜	俙	侻	侳	俛	俇	俖	侺	俀	侹	俬
F	剄	剉	勀	勂	匽	卼	厗	厖	厙	厘	咺	咡	咭	咥	哏	

CE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	哃	苟	咷	咮	哖	咶	哅	哆	咠	呰	咼	咢	咾	呲	哞	咰
5	垵	垞	垟	垤	垌	垗	垝	垛	垔	垘	垏	垙	垥	垚	垕	壴
6	复	奓	姡	姞	姮	娀	姱	姝	姺	姽	姼	姶	姤	姲	姷	姛
7	姩	姳	姵	姠	姾	姴	姭	宨	屌	峐	峘	峌	峗	峋	峛	
A		峞	峚	峉	峇	峊	峖	峓	峔	峏	峈	峆	峎	峟	峸	巹
B	帡	帢	帣	帠	帤	庰	庤	庢	庛	庣	庥	弇	弮	彖	徆	怷
C	怹	恔	恲	恞	恅	恓	恇	恉	恛	恌	恀	恂	恟	怤	恄	恘
D	恦	恮	扂	扃	拏	挍	挋	拵	挎	挃	拫	拹	挏	挌	拸	拶
E	挀	挓	挔	拺	挕	挻	挩	敁	敔	斪	斿	旂	昡	昈	昮	昜
F	昪	昢	昳	昫	昺	昝	昴	昶	昜	朏	朐	柁	柲	柈	枺	

CF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	柜	枻	柸	柘	柀	枷	柅	柫	柤	柟	枵	柍	枳	柷	柶	柮
5	柣	柂	枹	柎	柧	柰	枲	柼	柆	柭	柌	枮	柦	柛	柺	柉
6	柊	柃	柪	柋	欨	殂	殄	殶	毖	毘	毠	氠	氡	洨	洴	洭
7	洟	洼	洿	洒	洊	泚	洳	洄	洙	洺	洚	洑	洀	洝	浂	
A		洁	洘	洷	洃	洏	浀	洇	洴	渢	洈	洢	洉	洐	炷	炟
B	炾	炱	炰	炡	炴	炵	炶	牁	牉	牊	牬	牰	牳	牮	狊	狤
C	狨	狫	狟	狪	狦	狣	玅	玌	珂	珈	珅	玹	玶	玵	玴	珫
D	玿	珇	玾	珃	珆	玸	珋	瓬	瓮	甮	畇	畈	疧	疪	癹	盄
E	眈	眃	眄	眅	眊	盷	盻	盺	矧	矨	砆	砑	砒	砅	砐	砏
F	砎	砉	砃	砓	祊	祌	祋	祅	祄	秕	种	秏	秖	秎	窀	

D0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	穾	竑	笀	笁	籺	籸	籹	籿	粀	粁	紃	紈	紁	罘	羑	羍
5	羾	耇	耎	耏	耔	耷	胘	胇	胠	胑	胈	胂	胐	胅	胣	胙
6	胜	胊	胕	胉	胏	胗	胦	胍	臿	舡	芔	苙	苾	苹	茇	苨
7	茀	苕	茺	苫	苖	苴	苬	苡	苲	苵	茌	苻	苶	苰	苪	
A		苤	苠	苺	苳	苭	虷	虴	虼	虳	衁	衎	衧	衪	衩	觓
B	訄	訇	赲	迣	迡	迮	迠	邦	邽	邿	郕	郅	郜	郇	郈	郔
C	釔	釓	陔	陏	陑	陓	陊	陎	倞	倅	倇	倓	倢	倰	倛	俵
D	俴	倳	倷	倬	俶	俷	倗	倜	倠	倧	倵	倯	倱	倎	党	冔
E	冓	凊	凄	凅	凈	凎	剡	剚	剒	剞	剟	剕	剢	勍	匎	厞
F	唦	哢	唗	唒	哧	哳	哤	唚	哿	唄	唈	哫	唑	唅	哱	

D1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	唊	哻	哷	哸	哠	唎	唃	唋	圁	圂	埌	堲	埕	埒	垺	埆
5	垽	垼	垸	垶	垿	埇	埐	垹	埁	夎	奊	娙	娖	娭	娮	娕
6	娏	娗	娊	娞	娳	孬	宧	宭	宬	尃	屖	屔	峬	峿	峮	峱
7	崒	崀	峹	帩	帨	庨	庮	庪	庬	弳	弰	彧	恝	恚	恧	
A		恁	悢	悈	悀	悒	悁	悝	悃	悕	悛	悗	悇	悜	悎	戙
B	扆	拲	挐	捖	挬	捄	捅	挶	捃	揤	挹	捋	捊	挼	挩	捁
C	挴	捘	捔	捙	挭	捇	挳	捚	捑	挸	捗	捀	捈	敊	敆	旆
D	旃	旄	旂	晊	晟	晇	晑	朒	朓	栟	栚	桉	栲	栳	栻	桋
E	桏	栖	栱	栜	栵	栫	栭	栯	桎	桄	栴	栝	栒	栔	栦	栨
F	栮	桍	栺	栥	栠	欬	欯	欭	欱	欴	歭	肂	殈	毦	毤	

D2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	毨	毣	毢	毧	氥	浺	浣	浤	浶	洍	浡	涒	浘	浢	浭	浯
5	涑	涍	淯	浿	涆	浞	浧	浠	涗	浰	浼	浟	涂	涘	洯	浨
6	涋	浾	涀	涄	洖	涃	浻	浽	浵	涐	烜	烓	烑	烝	烋	缹
7	烢	烗	烒	烞	烠	烔	烍	烅	烆	烇	烚	烎	烡	牂	牸	
A		牷	牶	猀	狺	狴	狾	狶	狳	狻	猁	珓	珙	珥	珖	玼
B	珧	珣	珩	珜	珒	珛	珔	珝	珚	珗	珘	珨	瓞	瓟	瓴	瓵
C	甡	畛	畟	疰	痁	疻	痄	痀	疿	疶	疺	皊	盉	眝	眛	眐
D	眓	眒	眣	眑	眕	眙	眚	眢	眧	砣	砬	砢	砵	砯	砨	砮
E	砫	砡	砩	砳	砪	砱	祔	祛	祏	祜	祓	祒	祑	秫	秬	秠
F	秮	秭	秪	秜	秞	秝	窆	窉	窅	窋	窌	窊	窇	竘	笐	

D3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	笄	笓	笅	笏	笈	笊	笎	笉	笒	粄	粑	粊	粌	粈	粍	粅
5	紞	紝	紑	紎	紘	紖	紓	紟	紒	紏	紌	罜	罡	罞	罠	罝
6	罛	羖	羒	翃	翂	翀	耖	耾	耹	胺	胲	胹	胵	脁	胻	脀
7	舁	舯	舥	茳	茭	荄	茙	荑	茥	荖	茿	荁	茦	茜	茢	
A		荂	荎	茛	茪	茈	茼	荍	茖	茤	茠	茷	茯	茩	荇	荅
B	荌	荓	茞	茬	荋	茧	荈	虓	虒	蚢	蚨	蚖	蚍	蚑	蚞	蚇
C	蚗	蚆	蚋	蚚	蚅	蚥	蚙	蚡	蚧	蚕	蚘	蚎	蚝	蚐	蚔	衃
D	衄	衭	衵	衶	衲	袀	衱	衿	衯	袃	衾	衴	衼	訒	豇	豗
E	豻	貤	貣	赶	赸	趵	趷	趶	軑	軓	迾	迵	适	迿	迻	逄
F	迼	迶	郖	郠	郙	郚	郣	郟	郥	郘	郛	郗	郜	郤	酐	

D4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	酎	酏	釕	釢	釚	陜	陟	隼	飣	髟	鬯	乿	偰	偪	偡	偞
5	偠	偓	偋	偝	偲	偈	偍	偁	偛	偊	偢	倕	偅	偟	偩	偫
6	偣	偤	偆	偀	偮	偳	偗	偑	凐	剫	剭	剬	剮	勖	勓	匭
7	厜	啵	啶	唼	啍	啐	唴	唪	啑	啢	唶	唵	唰	啒	啅	
A		唌	唲	啥	啎	唹	啈	唭	唻	啀	啋	圊	圇	埻	堔	埢
B	埶	埜	埴	堀	埭	埽	堈	埸	堋	埳	埏	堇	埮	埣	埲	埥
C	埬	埡	堎	埼	堐	埧	堁	堌	埱	埩	埰	堍	堄	奜	婠	婘
D	婕	婧	婞	娸	娵	婭	婐	婟	婥	婬	婓	婤	婗	婃	婝	婒
E	婄	婛	婈	媎	娾	婍	娹	婌	婰	婩	婇	婑	婖	婂	婜	孲
F	孮	寁	寀	屙	崞	崋	崝	崚	崠	崌	崨	崍	崦	崥	崏	

D5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	崰	崒	崣	崟	崮	帾	帴	庱	庴	庹	庲	庳	弶	弸	徛	徖
5	徟	悊	悐	悆	悾	悰	悺	惓	惔	惏	惤	惙	惝	惈	悱	惛
6	悷	惊	悿	惃	惍	惀	挲	捥	掊	掂	捽	掽	掞	掭	掝	掗
7	掫	掎	捯	掇	掐	据	掯	捵	掜	捭	掮	捼	掤	挻	掟	
A		捸	掅	掁	掑	掍	揰	敓	旍	晥	晡	晛	晙	晜	晢	朘
B	桹	梇	梐	梜	桭	桮	梮	梫	楖	桯	梣	梬	梩	桵	桴	梲
C	梏	桷	梒	桼	桫	桲	梪	梀	桱	桾	梛	梖	梋	梠	梉	梤
D	桸	桻	梑	梌	梊	桽	欶	欳	欷	欸	殑	殏	殍	殎	殌	氪
E	淀	涫	涴	涳	湴	涬	淩	淢	涷	淶	淔	渀	淈	淠	淟	淖
F	涾	淥	淜	淝	淛	淴	淊	涽	淭	淰	涺	淕	淂	淏	淉	

D6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	淐	淲	淓	淽	淗	淍	淣	涻	烺	焍	烷	焗	烴	焌	烰	焄
5	烳	焐	烼	烿	焆	焓	焀	烸	烶	焋	焂	焎	牾	牻	牼	牿
6	猝	猗	猇	猑	猘	猊	猈	狿	猏	猞	玈	珶	珸	珵	琄	琁
7	珽	琇	琀	珺	珼	珿	琌	琋	珴	琈	畤	畣	痎	痒	痏	
A		痋	痌	痑	痐	皏	皉	盓	眹	眯	眭	眱	眲	眴	眳	眽
B	眥	眻	眵	硈	硒	硉	硍	硊	硌	砦	硅	硐	祤	祧	祩	祪
C	祣	祫	祡	离	秺	秸	秶	秷	窏	窔	窐	笵	筇	笴	笥	笰
D	笢	笤	笳	笘	笪	笝	笱	笫	笭	笯	笲	笸	笚	笣	粔	粘
E	粖	粣	紵	紽	紸	紶	紺	絅	紬	紩	絁	絇	紾	紿	絊	紻
F	紨	罣	羕	羜	羝	羛	翊	翋	翍	翐	翑	翇	翏	翉	耟	

D7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	耞	耛	聇	聃	聈	脘	脥	脙	脛	脭	脟	脬	脞	脡	脕	脧
5	脝	脢	舑	舸	舳	舺	舴	舲	艴	莐	莣	莨	莍	荺	荳	莤
6	荴	莏	莁	莕	莙	荵	莔	莩	荽	莃	莌	莝	莛	莪	莋	荾
7	莥	莯	莈	莗	莰	荿	莦	莇	莮	荶	莚	虙	虖	蚿	蚷	
A		蛂	蛁	蛅	蚺	蚰	蛈	蚹	蚳	蚸	蛌	蚴	蚻	蚼	蛃	蚽
B	蚾	衒	袉	袕	袨	袢	袪	袚	袑	袡	袟	袘	袧	袙	袛	袗
C	袤	袬	袲	袓	袎	覂	觖	觙	觕	訰	訧	訬	訞	谹	谻	豜
D	豝	豽	貥	赽	赻	赹	趼	跂	趹	趿	跁	軘	軞	軝	軜	軗
E	軠	軡	逤	逋	逑	逜	逌	逡	郯	郪	郰	郴	郲	郳	郔	郫
F	郬	郩	酖	酘	酚	酓	酕	釬	釴	釱	釳	釸	釤	釹	釪	

D8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	釫	釭	釨	釯	镺	閆	閈	陼	陭	陫	陱	陯	隿	靪	頄	飥
5	馗	傛	傕	傔	傞	傋	傣	傃	傌	傎	傝	偨	傜	傒	傂	傇
6	兟	凔	匒	匑	厤	厧	喑	嗁	喥	喭	喞	嗚	喢	喓	喈	喏
7	喵	喁	喣	喒	喤	啍	喌	喦	啿	喕	喡	喎	圌	堩	堷	
A		堙	堞	堧	堣	堨	埵	塈	堥	堜	堛	堳	堿	堶	堮	堹
B	堸	堭	堬	堻	奡	媯	媔	媟	婺	媢	媞	婸	媦	婼	媥	媬
C	媕	媮	娷	媄	媊	媗	媃	媋	媩	婻	婽	媌	媜	媏	媓	媝
D	寪	寍	寋	寔	寑	寊	寎	尌	尰	崷	嵃	嵫	嵁	嵋	崿	崵
E	嵑	嵎	嵕	崳	崺	嵒	崽	崱	嵙	嵂	崹	嵉	嵦	崼	崲	尌
F	嵀	嵗	幄	幁	彘	徦	徥	徫	惉	悹	惌	惢	惎	惄	愔	

D9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	惲	愊	愖	愅	惵	愓	惸	惼	惾	惁	愃	愘	愝	愐	惿	愄
5	愋	扊	掔	掱	掰	揎	揥	揨	揯	揃	撝	揳	揊	揠	揶	揕
6	揲	揵	摡	揟	掾	揝	揜	揄	揘	揓	揂	揇	揌	揋	揈	揰
7	揗	揙	攲	敧	敪	敤	敜	敨	敥	斌	斝	斞	斮	旐	旒	
A		晼	晬	晻	暀	晱	晹	晪	晲	朁	椌	棓	椄	棜	椪	棬
B	棪	棱	椏	棖	棷	棫	棤	棶	椓	椐	棳	棡	椇	棌	椈	楰
C	梴	椑	棯	棆	椔	棸	棐	棽	棼	棨	椋	椊	椗	棎	棈	棝
D	棞	棦	棴	棑	椆	棔	棩	椕	椥	棇	欹	欻	欿	欼	殔	殗
E	殙	殕	殽	毰	毲	毳	氰	淼	湆	湇	渟	湉	溈	渼	渽	湅
F	湢	渫	渿	湁	湝	湳	渜	渳	湋	湀	湑	渻	渃	渮	湞	

DA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	湨	湜	湡	渱	渨	湠	湱	湫	渹	渢	渰	湓	湥	渧	湸	湤
5	湷	湕	漄	湒	湦	渵	渶	湚	焠	焞	焯	烻	焮	焱	焣	焥
6	焢	焲	焟	焨	焊	閄	牋	牚	犈	犉	犆	犅	犋	猒	猋	猰
7	猢	猱	猳	猧	猲	猭	猦	猣	猵	猌	琮	琬	琰	琫	琖	
A		琚	琡	琭	琱	琤	琣	琝	琩	琠	琲	瓻	甯	畯	畬	痧
B	痚	痡	痦	痝	痟	痤	痗	皕	皒	盜	睆	睇	睄	睍	睅	睊
C	睎	睋	睌	矞	矬	硠	硤	硥	硜	硭	硒	硪	确	硲	硩	碑
D	硞	硢	祴	祳	祲	祰	稂	稊	稃	稌	稄	窐	竦	竑	笅	笻
E	笵	笘	笙	笳	筀	笡	笐	粢	粞	粨	粡	絘	絯	絣	絓	絖
F	絧	絪	絏	絭	絜	絫	絒	絔	絩	絑	絟	絎	缾	缿	罥	

DB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	罦	羢	羠	羡	翗	聑	聏	聐	胾	胔	腃	腊	腒	腏	腇	脽
5	腍	脺	臦	臮	臷	臸	臹	舄	舼	艂	舿	艴	茻	菏	菹	萣
6	菀	萎	莌	菧	菤	菼	菶	萐	菆	菈	董	菽	莿	萁	菝	菥
7	菘	菿	菡	菋	菎	菖	萵	菉	萉	萏	萂	萑	萆	菂	菳	
A		菕	菺	菇	菑	萿	萱	萓	菭	萳	菄	菻	萂	菢	萛	萭
B	蒤	蛘	蛢	蛦	蛓	蛣	蛚	蛪	蛝	蛫	蛜	蛬	蛩	蛗	蛨	蛑
C	衈	衖	衕	袺	裗	袹	袸	裀	袾	袶	袼	袷	袽	袲	裋	裉
D	覜	覘	覗	觝	觚	觛	詎	詍	訹	詙	詀	詗	詘	詄	詅	詒
E	詈	詑	詊	詌	詏	豟	貁	貀	貺	貾	貰	貹	貵	趄	趀	趉
F	跘	跓	跍	跇	跖	跜	跏	跕	跙	跈	跗	跅	軒	軷	軺	

DC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	軹	軦	軮	軥	軵	軧	軨	軶	軫	軱	軬	軴	軩	逭	逴	逯
5	鄆	鄬	鄄	鄘	鄛	鄈	鄍	鄎	鄁	鄐	鄊	鄔	鄇	酡	酤	酟
6	酢	酠	鈁	鈊	鈥	鈃	鈚	鈦	鈏	鈌	鈀	鈒	釿	釽	鈆	鈄
7	鈧	鈂	鈜	鈤	鈙	鈗	鈅	鈖	镻	閍	閌	閐	陙	陜	隈	
A		隉	隃	隀	雂	雈	雃	雱	雰	靬	靰	靮	頇	颩	飫	鬿
B	冓	氂	亄	亶	傽	傿	僆	傮	僄	僊	傴	僈	僂	傰	僁	傺
C	傱	僋	僉	傶	傸	凗	勞	剸	剻	剼	嗃	嗛	嗌	嗐	噌	嗊
D	嗝	嗀	嗔	嗄	嗩	喿	嗒	喍	嗏	嗕	嘔	嗖	嘊	嗲	嗍	嗙
E	嗂	圔	塓	塨	塤	塏	塍	塉	塯	塕	塎	塝	塙	塥	塛	堽
F	塣	塱	壼	嫇	嫄	嫋	嫰	媏	媱	媵	嫡	媿	嫈	嫛	嫆	

DD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	媷	嫀	嫊	嫄	婧	嫍	媹	嫛	寖	寘	寙	尟	尳	嵱	嵣	嵊
5	嵥	嵲	嵬	嵞	嵨	嵧	嵢	巰	幏	幎	幘	幍	幋	廅	廌	廆
6	廋	廇	彀	徯	徭	惷	慉	慊	愫	慅	愶	愲	愮	慆	愯	慏
7	愩	慀	戠	截	戣	戢	戤	撆	掣	擊	搐	搒	搉	搠	搤	
A		搳	揁	摲	搕	搘	搹	搷	搢	揻	搌	搦	搰	搨	摁	搵
B	搯	擉	擼	摀	搥	搧	搋	掣	搛	摽	搸	搎	毃	斒	旓	暆
C	暌	暕	暐	暋	暊	暙	暔	㬰	朠	楦	楟	椸	楎	檜	楱	椿
D	楅	楪	椹	楂	楗	楙	楺	楈	楉	椵	楬	椳	椽	楥	棰	楸
E	椴	楩	楀	楯	楄	楶	楘	楁	楴	椮	椻	楋	楲	楜	楏	楑
F	椲	楒	椯	楻	椼	歆	欼	歃	歂	歛	歁	殛	殼	毷	毸	

DE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	毻	毰	毸	渷	滖	滈	溏	滀	溟	溓	溔	溠	溱	溹	滆	滒
5	溽	滁	溞	滉	溷	溰	滍	激	漨	溲	溾	滃	滜	漝	溙	溒
6	漼	溍	溤	漭	滸	溳	滐	瀛	溗	漸	淪	煇	粘	煒	煣	煠
7	煁	煝	煢	煲	煸	煪	煡	熚	煘	煃	煋	煰	煟	煐	煓	
A		煄	煍	牑	牏	犍	犌	犑	犐	犎	猼	獂	猻	猺	獀	獊
B	獉	瑄	瑊	瑋	瑒	瑑	瑗	瑀	瑏	瑜	瑎	瑂	瑆	瑍	瑔	瓡
C	瓿	甀	甇	甝	畹	畷	畓	痯	瘏	瘕	痷	痾	痼	痺	瘌	瘐
D	痻	痶	痭	痚	痽	皙	皵	盝	睕	睟	睠	睒	睖	睚	睩	睧
E	睔	睙	睭	矠	碇	碚	碔	碏	碄	碕	碅	碆	碡	碃	硹	碙
F	碀	碖	硻	祼	禂	祽	祹	稑	稘	稙	稒	稗	稈	稢	稓	

DF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	稛	稐	窣	窢	窞	竫	筦	筤	筭	筴	筩	筲	筥	筳	筱	筰
5	筡	筸	箁	筣	粲	粴	粯	綈	綆	綀	綍	絿	綅	絺	綎	絻
6	綃	絼	綌	綔	綄	絽	綒	罭	罫	罧	罨	罬	羦	羥	羧	翛
7	翣	耡	腤	腠	腷	腜	腩	腛	腢	腲	朡	腞	腶	腧	腯	
A		腄	腡	舝	艉	艄	艀	艂	艅	蓱	萿	葖	葶	葹	蒏	蒍
B	葥	葑	葀	萋	葧	萰	葍	葽	葚	葙	葴	葳	葝	蔇	葞	萷
C	萱	萴	葺	葃	葸	蔆	葅	萩	菙	萮	萯	葂	萬	葟	葰	萹
D	葎	葌	葒	葯	蓅	蒎	萻	葇	萶	萳	萺	葾	葄	萫	葠	葔
E	葭	葢	蜋	蜄	蛷	蜌	蛺	蜧	蛵	蜖	蛸	蜎	蜉	蜓	蛶	蜍
F	蜅	裖	裋	裍	裎	裞	裛	裚	裌	裐	覅	覛	觟	觥	觤	

E0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	觡	觠	觢	觜	触	詶	誆	詿	詡	訿	詷	誂	誄	詵	誃	誁
5	誧	詻	谼	豋	豊	豥	豤	豦	貆	貄	貅	貣	赨	赩	趀	趌
6	趎	趏	趍	趓	趔	趐	趒	跰	跠	跬	跱	跮	跐	跩	跣	跢
7	跧	跲	跫	跴	輆	軿	輁	輀	輅	輇	輈	輂	輋	遒	逿	
A		遄	遉	逽	鄐	鄍	鄏	鄑	郥	鄔	鄕	郞	酮	酯	鉈	鉒
B	鈰	鈺	鉦	鈳	鉥	鉞	銃	鈮	鉊	鉆	鉅	鉬	鉏	鈌	鉧	鈒
C	鈶	鉼	鉰	鈱	鉔	鉣	鉐	鉡	鉎	鉓	鉌	鉖	鈲	閟	閜	閞
D	閛	隒	隓	隑	隗	雎	雺	雽	雸	雵	靳	靷	靸	靲	頏	頍
E	頎	颬	飶	飹	馯	馲	馰	馵	骭	骫	魛	鳪	鳭	鳧	麀	黽
F	僦	僔	僗	僨	僳	僛	僪	僝	僤	僓	僬	僰	僯	僣	僠	

E1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	凘	劀	劁	勩	勫	匰	厬	嗋	嘕	嘌	嘒	嗼	嘏	嘜	嘁	嘓
5	嘂	嗺	嘝	嘄	嗿	嗹	墉	塼	墐	墘	墆	墁	塿	塴	墋	塺
6	墇	墑	墎	墶	墂	墈	塻	墔	墏	壾	奫	嫜	嫮	嫥	嫕	嫪
7	嫚	嫭	嫫	嫳	嫢	嫠	嫛	嫬	嫞	嫝	嫙	嫨	嫟	孷	寠	
A		寣	屧	嶂	嶀	嵽	嶆	嵺	嶁	嵷	嶊	嶉	嶈	嵾	嵼	嶍
B	嵹	嵿	幘	幙	幓	廘	廑	廗	廎	廜	廕	廙	廒	廔	彄	彈
C	彯	徶	愬	愨	慁	慞	慱	慳	慒	慓	慲	慬	憀	慴	慔	慺
D	慛	慥	愻	慪	慡	慖	戩	戧	戫	摮	摍	摛	摝	摴	摶	摲
E	摳	摽	摵	摦	撦	摎	摺	摞	摜	撥	摓	摠	摐	摿	摒	摬
F	摫	摙	摥	摷	敳	斠	暡	暠	暟	朅	朄	朢	榱	榶	槉	

E2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	榠	槎	榖	槢	榬	榼	槫	榙	椱	榧	榍	榎	榾	構	榿	槄
5	榽	榤	槔	榹	槊	榚	槏	榳	榓	榪	榡	榞	槙	榗	榐	槂
6	榵	榥	槆	歊	歍	歋	殞	殟	殠	毃	毄	毾	滎	滵	滱	漃
7	漥	滸	漷	漨	漮	漉	漇	漙	漚	漧	漘	漻	漒	澤	漊	
A		漶	潳	滹	滮	漭	潀	漰	漼	滶	潃	漽	漎	滫	漅	滽
B	潡	潙	澘	潕	漺	漟	漍	漞	漈	漡	熇	熐	熉	熀	熅	熂
C	熏	煻	熆	熁	熗	牄	牓	犗	犕	犓	獃	獍	獑	獌	瑢	瑳
D	瑱	瑵	瑲	瑧	瑮	甀	甂	甃	畽	疐	瘖	瘈	瘌	瘕	瘑	瘊
E	瘔	皸	瞁	睼	瞅	瞂	睮	瞀	瞖	睾	睶	碲	碪	碴	碭	碨
F	硾	碫	碞	碥	碠	碬	碢	碤	禘	禊	禋	禖	禕	禔	禓	

E3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	禗	禈	禒	禐	稫	穊	稰	稯	稨	稦	窨	窫	窬	竮	箈	箜
5	箊	箑	箐	箖	箍	箌	箛	箎	箅	箘	劄	箙	箤	箂	粻	粿
6	粼	粺	綧	綷	緂	綣	綪	緁	緀	緅	綝	緎	緄	緆	緋	緌
7	綯	綹	綖	綼	綟	綦	綮	綩	綡	緉	罳	翢	翣	翥	獠	
A		耤	聝	聜	膉	膆	膃	膇	膍	膌	膋	舕	蒗	蒤	蒡	蒟
B	蒺	蓎	蓂	蒬	蒮	蒫	蒹	蒴	蓁	蓍	蒪	蒚	蒱	蓐	蒝	蒧
C	蒻	蓧	蒔	蓇	蔆	蒛	蒩	蒯	蒨	蓖	蓻	蒶	蓏	蒠	蓗	蒸
D	蓒	蓛	蒰	蔎	虡	蜳	蜣	蜨	蝫	蝀	蜮	蜞	蜡	蜙	蜛	蝃
E	蜬	蝁	蜾	蝆	蜠	蜲	蜪	蜭	蜼	蜒	蜺	蜱	蜵	蝂	蜦	蜧
F	蜸	蜤	蜚	蜰	蜑	裷	裧	裱	裲	裺	裾	裮	裼	裶	裻	

E4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	裰	裬	裫	覎	覒	覛	覞	觩	觫	觨	誫	誙	誋	誒	誏	誖
5	谽	豨	豩	賕	賏	賗	趖	踉	踂	跿	踍	跽	踊	踃	踇	踆
6	踅	跫	踶	踄	輐	輑	輎	輍	郼	鄘	鄠	鄢	鄟	鄝	鄚	鄤
7	鄡	鄛	酺	酲	酻	酳	鈫	銤	鉶	銛	鉺	銠	鉼	銪	銍	
A		銦	銚	銫	鉹	銗	鉿	鉫	鉞	銐	銂	銕	銢	鉽	鉒	銈
B	銊	銆	鋍	銙	銑	鋒	銾	銽	鉥	鋌	鈭	隞	隡	雭	艴	靬
C	靺	靾	鞃	鞀	鞂	靻	鞄	鞁	靿	韎	韍	頖	颭	颮	餂	餀
D	餇	馝	馜	駃	馹	馻	馺	馽	鬵	駇	骱	髣	髧	鬾	魀	魠
E	魡	魟	鴀	鳲	鳵	麧	僆	儃	儰	僸	儆	儇	僶	僾	儋	儌
F	儊	僜	劋	劇	勱	勯	噈	噂	噌	嘵	噁	噊	噉	噆	噘	

E5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	噚	噀	嘳	嘽	嘬	嘾	嘸	嘪	嘺	圚	墫	墝	墱	墠	墣	墯
5	墬	墥	墡	壿	嫿	嫴	嫽	嫮	嫶	嬃	嫸	嬂	嫹	嬁	嬇	嬅
6	嬏	屧	嶙	嶗	嶟	嶒	嶢	嶓	嶕	嶠	嶜	嶡	嶚	嶞	幩	幝
7	幠	幜	縻	廛	廞	廡	彉	徲	憋	憃	慹	憱	憰	憢	憉	
A		憛	憓	憯	憭	憟	憒	憪	憡	憍	憼	憳	戫	擎	摰	撽
B	撠	撅	撗	撜	撏	撋	撊	撌	撣	撟	摨	撱	撘	敶	敺	敹
C	敻	斲	斳	暵	暰	暩	暲	暷	暪	暯	樀	樆	樗	槥	槸	樕
D	標	槤	樠	槿	槬	槢	樛	樝	槾	樧	槲	槮	樔	槷	槧	榼
E	樈	槦	槻	樍	槳	槫	樉	樄	樘	樥	樏	槶	槎	樇	槴	翕
F	歔	殯	殣	殢	殦	氁	氀	毿	氂	潁	澂	潾	澇	濆	澒	

E6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	澍	澈	澌	潢	潏	澅	潚	澖	潶	潬	澂	潕	潲	潒	潐	潗
5	澔	澓	潝	漀	潡	潫	潽	潧	澐	潓	澋	潩	潿	澕	潣	潷
6	潪	潻	熲	熯	熛	熰	熠	熚	熩	熵	熝	熥	熞	熤	熡	熪
7	熜	熧	熳	犘	犚	獘	獒	獞	獟	獠	獝	獛	獡	獚	獙	
A		獢	璇	璉	璊	璆	璁	瑽	璅	璈	瑼	瑹	甈	甇	畾	瘥
B	瘞	瘙	瘝	瘜	瘣	瘚	瘨	瘛	皜	皝	皞	皛	瞍	瞏	瞉	瞈
C	磍	碻	磏	磌	磑	磎	磔	磈	磃	磄	磉	禚	禡	禠	禜	禢
D	禛	歶	稹	窲	窴	窳	箷	篋	箾	箬	篎	箯	箹	篊	箵	糅
E	糈	糌	糋	緷	緛	緪	緧	緗	緡	縃	緺	緦	緶	緱	緰	緮
F	緟	罶	羬	羰	羭	翭	翫	翪	翬	翦	翨	聤	聧	膣	膟	

E7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	膞	膕	膢	膙	膍	舖	艏	艓	艒	艐	艎	艑	蔤	蔻	蔏	蔀
5	蔩	蔎	蔉	蔍	蔟	蔊	蔧	蔜	蓻	蔫	蓺	蔈	蔌	蓴	蔪	蓲
6	蔕	蓷	蓫	蓳	蓼	蔒	蓪	蓩	蔖	蓾	蔨	蔝	蔮	蔂	蓽	蔞
7	蓶	蔱	蔦	蓧	蓨	蓰	蓯	蓹	蔘	蔠	蔰	蔋	蔙	蔯	虢	
A		蝖	蝣	蝤	蝷	蟡	蝳	蝘	蝔	蝛	蝒	蝡	蝚	蝑	蝞	蝭
B	蝪	蝐	蝎	蝟	蝝	蝯	蝬	蝺	蝮	蝜	蝥	蝏	蝻	蝵	蝢	蝧
C	蝩	衚	褅	褌	褔	褋	褗	褘	褙	褆	褖	褑	褎	褉	覢	覤
D	覣	觭	觰	觬	諏	諆	謚	諓	諑	諔	諕	誻	諗	誾	諀	諅
E	諘	諃	諫	説	諙	谾	豍	貏	賥	賟	賙	賨	賚	賝	賧	趠
F	趜	趡	趛	踠	踣	踥	踤	踮	踕	踛	踖	踑	踙	踦	踧	

E8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	踔	踒	踘	踓	踜	踗	踚	輬	輤	輘	輚	輠	輣	輖	輗	遳
5	遰	遯	遧	遫	鄯	鄫	鄩	鄪	鄲	鄦	鄮	醅	醆	醊	醁	醂
6	醄	醀	鋐	鋃	鋄	鋀	鋙	銶	鋏	鋱	鋟	鋘	鋩	鋗	鋝	鋌
7	鋯	鋂	鋨	鋊	鋈	鋎	鋦	鋍	鋕	鋉	鋠	鋞	鋧	鋑	鋓	
A		鎉	鋡	鋆	鋻	閜	閬	閫	閮	閰	隤	隥	雒	雺	雽	雸
B	靚	鞊	鞎	鞈	韐	韏	頞	頝	頦	頩	頨	頠	頛	頧	颲	餈
C	飺	餑	餔	餖	餗	餕	駜	駍	駏	駓	駔	駎	駉	駖	駘	駋
D	駗	駌	骳	髬	髫	髳	髲	髱	魆	魃	魧	魴	魱	魦	魶	魵
E	魰	魨	魤	魬	鳼	鳺	鳽	鳿	鳷	鴇	鴅	鴃	鴆	鴈	鴁	鴄
F	麀	黕	鼏	鼐	儜	儓	儗	儚	儑	凞	匴	叡	噰	噠	噮	

E9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	噳	噦	噣	噭	噲	噞	噉	圜	圛	壈	墽	壉	墿	墺	壂	墼
5	壆	嬗	嬙	嬛	嬡	嬔	嬓	嬐	嬖	嬨	嬚	嬠	嬞	寯	嶬	嶱
6	嶩	嶧	嶵	嶰	嶮	嶪	嶨	嶲	嶭	嶯	嶴	幧	幨	幦	幯	廩
7	廧	廦	廨	廥	彋	徼	憝	憨	憖	懅	憴	懆	懁	懌	憺	
A		憿	憸	憌	擗	擖	擐	撖	擉	撽	擏	擃	擛	擳	擙	擿
B	斀	斁	斢	曈	曔	曀	曊	曋	曏	暽	曍	暺	曡	朣	樴	橦
C	橉	橧	樲	樍	檅	橝	椲	橶	橜	橑	樨	橚	橨	樿	檹	橪
D	橤	橐	橏	橔	橯	橩	橀	樨	橞	檏	橕	橌	橎	橆	歕	歔
E	歖	殧	殪	殫	毈	毇	氄	氅	氃	澭	濋	澣	濇	澼	濎	濈
F	潞	濄	澽	澞	濊	澨	瀄	澥	澮	澺	濱	澪	濏	澿	澸	

EA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	澢	濰	澫	濍	濸	澲	澰	燅	燂	熿	熸	燖	燀	燁	燋	燔
5	燊	燇	燏	熽	燗	熼	燆	燚	燛	犝	犞	獩	獦	獧	獬	獥
6	獫	獪	瑿	璚	璠	璔	璒	璕	璡	甋	甐	瘯	瘭	瘱	瘽	瘳
7	瘼	瘵	瘲	瘰	皻	盦	瞚	瞝	瞡	瞜	瞛	瞢	瞘	瞕	瞙	
A		瞗	礉	礆	磥	磪	磞	磣	磛	礪	磢	磸	磟	磠	禰	穄
B	穈	穇	窶	窸	窵	窱	竀	篁	篣	篧	篝	篕	篥	篚	篨	篹
C	篔	篪	篢	篜	篫	篘	篟	糒	糔	糗	糐	糑	縒	縡	縗	縰
D	縟	縠	縓	縃	縜	縕	縚	縢	縋	縏	縖	縍	縔	縥	縤	罃
E	罻	罼	罺	羱	翯	耪	耩	聬	膱	膦	膮	膹	膵	膫	膰	膬
F	膴	膲	膷	膧	臲	艕	艖	艗	蕖	蕅	蕫	蕍	蕓	蕡	蕘	

EB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	蕀	蕆	蕤	蕁	蕢	蕄	蕑	蕇	蕣	蔾	蕛	蕱	蕎	蕮	蕵	蕕
5	蕧	蕠	薌	蕦	蕝	蕔	蕥	蕬	虣	虥	虦	螛	螏	螗	螓	螒
6	螈	螁	螖	螘	蝹	螇	螣	螅	螐	螑	螝	螄	螔	螜	螚	螉
7	褞	褦	褰	褭	褮	褧	褱	褢	褩	褣	褯	褬	褟	觱	諠	
A		諢	諲	諴	諵	諝	謔	諤	諟	諰	諈	諞	諡	諨	諿	諯
B	諻	貑	貒	貐	賵	賮	賱	賰	賳	赬	赮	趥	趧	踳	踾	踸
C	蹀	蹅	踶	踼	踽	蹁	踰	踿	躽	輶	輮	輵	輲	輹	輷	輴
D	遶	遹	遻	邆	郺	鄳	鄵	鄶	醓	醐	醑	醍	醏	錧	錞	錈
E	錟	錆	錏	鍺	錸	錼	錹	錣	錒	錁	鍆	錭	錎	錍	鋋	錝
F	鋺	錥	錓	鋹	鋷	錴	錂	錤	鋿	錩	錉	錵	錪	錔	錊	

EC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	錋	鋾	錉	錀	鋻	鎜	閯	闍	閾	閹	閿	閶	閺	闈	闇	隩
5	雔	霋	霒	霐	鞙	鞗	鞔	鞓	韸	頛	頨	頲	餤	餟	餧	餕
6	馞	駮	駬	駥	駤	駰	駣	駪	駩	駧	骹	骿	骴	骻	髶	髺
7	髹	髷	鬳	鮀	鮅	鮇	魼	魾	鮋	鮂	鮓	鮒	鮐	鮣	鮎	
A		鮏	鮈	鮄	鴂	鴠	鴺	鴥	鴩	鴘	鴗	鴝	鴐	鴔	鴇	麀
B	麇	麋	麩	麭	默	黖	黺	鼒	鼽	儦	儥	儢	儤	儳	儩	勴
C	嚓	嚌	嚧	嚆	嚄	嚃	嚐	嚂	嚀	嚁	壖	壔	壏	壒	嬭	嬥
D	嬲	嬣	嬬	嬧	嬦	嬯	嬮	孻	寱	寲	嶷	幬	幪	徾	徻	懃
E	憵	憼	懧	懠	懥	懤	懨	懞	擯	擩	擣	擫	擤	擨	斁	斀
F	斶	旚	曒	檍	檖	檁	檥	檉	檟	檛	檡	檞	檇	檓	檎	

ED	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	檕	檃	檏	檤	檑	橿	檦	檚	檅	檌	檒	歛	殭	氉	黏	燢
5	濛	濔	濣	濾	濫	濧	濦	濞	濲	濝	濹	濨	燡	燱	燨	燲
6	燤	燰	燢	獳	獮	獯	璗	璲	璫	璐	璪	璭	璱	璥	璯	甐
7	甑	甒	甏	疄	癕	癈	癉	癇	皤	盩	瞵	瞫	瞲	瞷	瞶	
A		瞴	瞱	瞨	矰	磳	磽	礂	磻	磼	磲	礅	磹	磾	礄	禫
B	禨	穜	穛	穖	穘	穔	穚	窾	竀	竁	簅	簏	篲	簀	篿	篻
C	簎	篴	簋	篳	簂	簉	簃	簁	篸	篽	簆	篰	篱	簐	簊	糕
D	縭	縼	繂	縳	顈	縸	縪	繉	繀	繇	縩	繌	縰	縻	縶	繄
E	縺	罅	罿	罾	罽	翴	翲	耬	膻	臄	臌	臊	臅	臇	膼	臩
F	艛	艚	艜	薃	薀	薏	薧	薕	薠	薋	薣	蕻	蓶	薚	蕿	

EE	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	蕷	蕼	薉	蕭	蕺	藘	蕗	蔑	藡	薆	薍	薙	薝	薁	薢	薂
5	薈	薅	蕹	薵	薘	藆	薟	虨	螓	螪	螭	蟅	螔	螬	螹	螵
6	螼	螮	蟉	蟃	蟂	蟌	螷	螯	蟄	蠡	蟘	蟲	螿	螢	螽	蟼
7	螳	褵	褳	褼	褾	襁	襒	褷	襂	覭	覯	覮	觲	觳	謞	
A		謘	謖	謑	謅	謋	謢	謏	謒	謕	謇	謍	謈	謆	謜	謓
B	謚	謙	豰	豲	豱	豯	貕	貔	賯	赯	蹎	蹍	蹓	蹐	蹌	蹇
C	轃	轀	邅	遾	鄸	醚	醓	醛	醙	醟	醡	醝	醢	鎡	鎃	鄊
D	鍤	鍖	鍇	鍼	鍘	鍜	鍶	鍉	鍐	鍑	鍠	鍭	鎏	鍌	鍪	鍹
E	鍗	鍕	鍒	鍏	鍱	鍷	鍻	鍡	鍞	鍣	鍧	鎀	鍎	鍙	闉	闋
F	闌	闃	閿	䌉	隮	隰	隬	霠	霟	霘	霝	霙	鞚	鞡	鞜	

EF	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	韡	韝	韟	韔	韱	頲	頣	顊	顉	顅	顃	餮	餫	餬	餪	餳
5	餲	餯	餭	餱	餰	馘	馣	馡	騂	駺	駴	駷	駹	駸	駶	駻
6	駽	駾	駼	騃	骾	髾	髸	鬁	鬇	魋	鮚	鮨	鮞	鮇	鮦	鮡
7	鮥	鴷	鴜	鮢	鮠	鮯	鴽	鴔	鵧	鴶	鵜	鴯	鴱	鵥	鴰	
A		鴿	鵂	鵃	鵏	鴑	鵟	鴐	鵴	鵖	麆	麛	麑	麰	黈	黚
B	黻	黿	鼔	鼣	鼢	齔	龠	儱	儭	儮	嚘	嚜	嚗	嚚	嚝	嚙
C	嬰	嬼	屩	屫	巀	幭	幱	懘	懟	懬	懮	懱	懪	懰	懫	懖
D	懹	擿	攄	擽	攕	攁	攃	擼	斀	旛	曚	曞	曘	櫅	櫇	檽
E	櫤	櫆	檬	檶	檷	櫴	檴	櫷	歞	毉	氉	濱	瀌	瀘	瀁	瀅
F	瀔	瀎	濿	瀀	瀆	瀦	濼	瀿	瀯	爁	燿	燹	爃	燾	獶	

F0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	璸	瓀	璵	瓁	璾	瓃	璻	瓂	甔	甓	瘕	癠	癲	癐	癓	癗
5	癚	皦	皾	盬	矂	瞺	磿	礌	礓	礔	礉	礐	礒	礑	禭	禬
6	穟	簜	簩	簙	簠	簟	簭	簝	簦	簨	簢	簥	簰	繜	繐	繖
7	繣	繘	繢	繟	繑	繠	繗	繓	羵	羳	翷	翸	聵	臑	臒	
A		臐	艟	艞	藒	藆	藇	藅	藔	薳	薵	薽	薁	藄	薿	藋
B	藎	藑	藚	藞	藧	藲	藶	薸	藳	薾	虩	蟧	蟦	蟢	蟛	蟫
C	蟪	蟥	蟟	蟳	蟤	螵	蟜	蟓	蟭	蟘	蟣	蟨	蠆	蟙	蟚	蟼
D	蠁	蟂	襓	襋	襏	襌	襆	襐	襑	襉	謪	謧	謣	謳	謰	謵
E	譇	謯	謼	謾	謱	謥	謷	謦	謶	謮	謤	謻	謽	謺	豂	豵
F	貙	貘	貗	賾	贄	賿	賢	蹈	蹢	蹠	蹗	蹐	蹞	蹥	蹧	

F1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	蹛	蹚	蹡	蹝	蹩	蹔	轆	轇	轈	轋	鄨	鄺	鄻	鄾	醨	醥
5	醧	醘	醪	鎵	鎌	鎒	鎷	鎛	鎝	鎉	鎧	鎎	鎪	鎞	鎦	鎕
6	鎈	鎙	鎟	鎍	鎱	鎑	鎲	鎤	鎨	鎴	鎣	鎥	闒	闓	闑	隳
7	雗	雚	巂	雟	雘	雝	霣	霢	霥	鞬	鞮	鞨	鞫	鞤	鞪	
A		鞢	鞥	韗	韙	韖	韘	韺	顐	顑	顒	颸	饁	餼	餺	騏
B	騋	騉	騍	騄	騑	騊	騅	騇	騆	髀	髜	鬈	鬄	鬅	鬩	鬵
C	魊	魌	魋	鯇	鯆	鯃	鮿	鯁	鮵	鮸	鯓	鯝	鯄	鮹	鯀	鵛
D	鵊	鵓	鵈	鵄	鵅	鵙	鵖	鵌	鵎	鵒	鵔	鵟	鵐	鵚	麋	麜
E	黟	鼁	鼀	鼖	鼥	鼫	鼪	鼩	鼨	齌	齕	儴	儵	劀	勷	厴
F	嚫	嚭	嚦	嚧	嚪	嚬	壚	壝	壛	夒	嬽	嬾	嬿	巃	幰	

F2	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	儱	懮	攄	攘	攍	攉	攌	攎	檾	旞	旝	曞	櫧	櫅	櫌	櫑
5	櫖	櫋	櫟	櫜	櫐	豶	櫏	櫇	櫞	歠	殰	氌	瀙	瀧	瀠	瀖
6	瀫	瀡	瀢	瀊	瀨	濾	瀤	瀜	瀪	爌	爊	爇	爂	爅	犥	犦
7	犤	犣	犡	瓋	瓅	璬	瓃	疉	癠	矉	矊	矄	矱	礝	礛	
A		礡	礜	礗	礞	禰	穧	穨	簳	簼	簹	簬	簻	糬	糪	繶
B	繵	繸	繰	繷	繯	繺	繲	繴	繨	罋	罊	羃	羆	羷	翽	翾
C	聸	臏	臕	艤	艡	艣	藫	藱	藰	藙	藡	薕	藚	藗	藬	藴
D	藸	藘	藟	蘢	藜	蘷	藰	藦	藯	藞	藢	蟦	蟺	蠃	蟶	蟷
E	蠉	蠌	蠋	蠆	螫	蟞	蟿	蠊	蠂	襢	襚	襛	襗	襡	襜	襘
F	襝	襙	覈	覷	覶	觶	譐	譈	譊	譀	譓	譖	譔	調	譕	

F3	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	譑	譂	譒	譇	徼	豷	豶	貚	贆	贇	贉	趬	趪	趭	趫	蹭
5	蹸	蹳	蹪	蹯	蹻	軂	轒	轑	轏	轐	轓	辴	酀	鄿	醰	醭
6	鏞	鏇	鏏	鏂	鏚	鏐	鏃	鏬	鏌	鏙	鏋	鏦	鏊	鎮	鎌	鏣
7	鏕	鏄	鏎	鏀	鏒	鎜	鐐	闃	闉	雛	霩	霫	霬	霨	霚	
A		鞈	鞷	鞶	鞲	韞	韟	顜	顙	顝	顗	颾	颽	颼	飀	饈
B	饇	饃	馦	馧	騟	騕	騥	騝	騤	騑	騢	騠	騧	騣	䮗	騜
C	騔	髂	鬋	鬊	鬎	鬌	鬷	鯪	鯫	鯠	鯞	鯤	鮥	鯢	鯰	鯔
D	鯗	鯊	鯘	鯙	鯥	鯕	鯡	鯚	鵷	鵇	鵏	鶕	鵱	鵲	鶀	鵨
E	鶜	鶝	鶌	鵽	鵯	鵸	鵵	鵰	鵬	鶂	鶣	雝	鵼	鶋	鶩	鶡
F	鶇	鴿	麔	麑	麌	黼	鼭	鼿	鼩	齍	齖	齗	齘	匷	彈	

F4	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	嚵	嚳	瓕	孅	巆	巇	廮	廯	忀	懪	懹	攈	攖	攕	攓	旟
5	曨	曣	曤	櫳	櫰	櫪	櫨	櫹	櫱	櫮	橅	瀼	灘	瀯	瀷	瀴
6	灟	灂	瀸	灋	瀺	瀹	灀	瀻	瀳	瀾	爓	爔	犨	獽	獼	璺
7	皫	皪	皾	盭	矌	矎	矏	矍	矲	礥	礣	礧	礨	礤	礩	
A		禲	穮	穬	穭	竷	籉	籈	籊	籇	籅	糮	繻	繾	纁	纀
B	羺	翷	聹	臛	臙	舋	艨	艩	蘢	藿	蘁	藾	蘛	蘀	藶	蘄
C	蘉	蘅	蘌	藽	蠙	蠐	蠑	蠗	蠓	蠖	襣	襦	覹	觷	譠	譪
D	譝	譨	譣	譥	譧	譭	趮	躆	躈	躄	轙	轖	轗	轕	轘	轚
E	邍	酃	酁	醷	醵	醲	醳	鍚	鐓	鏻	鐠	鐏	鐔	鏉	鐕	鐐
F	鐨	鐙	鐍	鏵	鐀	鏷	鐇	鐎	鐖	鐒	鏺	鐉	鏸	鐊	鐰	

F5	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	鏷	鐌	鏶	鐑	鑋	闠	闟	闤	霿	靌	鞾	韂	韽	韾	顝	顟
5	顣	顙	颾	飂	饐	饎	饙	饌	饋	饓	騲	騴	騱	騬	騪	騶
6	騩	騮	騸	騭	髇	髊	髆	鬐	鬒	鬚	鰋	鰈	鯷	鰅	鰒	鯸
7	鱀	鰇	鰎	鰆	鰗	鰔	鰉	鶟	鶙	鶤	鶝	鶒	鶘	鶐	鶛	
A		鶠	鶔	鶜	鶪	鶗	鶡	鶚	鶢	鶨	鶞	鶣	鶿	鶩	鶖	鶦
B	鶬	麌	麔	麚	黥	黭	黨	黦	鼰	鼮	齛	齠	齞	齝	齙	龑
C	儺	儹	劘	劗	囃	嚽	嚾	孈	孇	巋	巏	廱	懽	攛	欂	櫼
D	欃	櫸	欀	灃	灄	灊	灈	灉	灅	灆	爝	爚	爙	獾	甗	癪
E	矐	礭	礱	礫	籔	籓	糲	纊	纇	纈	纋	纆	纍	罍	羻	耰
F	贏	蘘	蘪	蘦	蘟	蘣	蘜	蘙	蘧	蘮	蘡	蘠	蘩	蘞	蘥	

F6	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	蠩	蠉	蠛	蠠	蠤	蠥	蠫	蠬	襭	襩	襮	襫	覻	譹	譸	譿
5	譺	譻	贕	贔	趯	躎	躌	轞	轛	轝	酆	酄	酅	醹	鐿	鐻
6	鐶	鐩	鐽	鐼	鐰	鐹	鐪	鑈	鑯	鑀	鐱	闥	闤	闟	霮	霰
7	鞿	韡	顤	飋	飖	飀	饘	饖	騹	騽	驆	驄	驂	驁	驇	
A		驔	髍	鬠	鬗	鬘	鬖	鬺	魒	鰫	鰝	鰜	鰬	鰣	鰨	鰩
B	鰤	鰡	鶷	鶬	鶭	鶹	鷇	鷊	鷏	鶾	鷅	鷃	鶻	鶸	鷈	鷂
C	鷆	鷌	麡	鶱	鷔	鷌	鶳	鷍	鷚	鹺	麷	黫	黮	黭	鼛	鼘
D	鼜	鼸	齎	齵	龤	龒	亹	囆	囅	囋	奱	孋	孌	巕	巑	廲
E	攡	攟	攞	攢	欋	欌	欉	氍	灕	灖	灗	灒	爞	爟	犩	獿
F	瓘	瓕	瓙	瓗	癭	皭	礵	禴	穰	穱	籗	籜	籙	籛	籚	

F7	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	籭	糶	纑	罏	羇	臞	艫	蘴	蘵	蘳	蘬	蘲	蘶	蠬	蠨	蠦
5	蠪	蠥	襱	覿	覾	觻	譾	讄	讂	讆	讅	譿	贕	躕	躔	躚
6	躒	躐	躖	躗	轀	轢	酇	鑌	鑐	鑊	鑋	鑏	鑇	鑅	鑈	鑉
7	鑆	霿	韣	顪	顩	飋	饔	饛	驎	驓	驔	驌	驏	驈	驊	
A		驉	驒	驐	髐	鬙	鬫	鬻	魖	魕	鱆	鱈	鰿	鱄	鱋	鱍
B	鱁	鰼	鰷	鰴	鰲	鰽	鰶	鷛	鷒	鷞	鷚	鷋	鷐	鷜	鷑	鷟
C	鷩	鷙	鷘	鷖	鷵	鷕	鷝	麶	黰	鼵	鼳	鼲	齂	齫	龕	龢
D	儽	劙	壨	壧	奲	孍	巘	蠯	彏	戁	戃	戄	攩	攥	斖	曫
E	欑	欒	欏	毊	灛	灚	爢	玂	玁	玃	癰	矔	籧	籦	纕	艬
F	蘺	虀	蘹	蘼	蘱	蘻	蘾	蠰	蠲	蠮	蠳	襶	襴	襳	觾	

F8	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	讌	[illegible]	讐	讈	襱	贊	躘	轤	轣	醼	鑢	鑕	鑝	鑗	鑞	韄
5	韅	護	驖	驙	鬞	鬟	鬠	鱒	鱘	鱐	鱊	鱍	鱸	鱕	鱙	鱌
6	鱎	驐	鷷	[illegible]	鷣	鷫	鷸	鷤	鷶	鷡	鷮	鷦	鷩	鷰	鷢	鷬
7	鷴	鷳	[illegible]	鷭	麡	黐	黲	黳	鼆	鼜	鼸	鼷	鼶	齃	齏	
A		齱	齰	齮	齯	囓	囍	孎	屭	攭	曭	曮	欓	灟	灡	灝
B	灠	爣	瓛	蠡	[illegible]	礸	襰	禶	籪	纗	羉	艭	瀻	蠸	蠷	蠵
C	衋	讔	讕	躞	躟	躠	躝	醾	醽	釂	鑫	鑨	鑩	雥	靆	靃
D	靇	韇	韥	驞	髕	[illegible]	鱣	鱧	鱦	鱢	鱞	鱠	鸂	鷾	鸇	鸃
E	鸆	鸅	鸀	鸁	鸉	鷿	鸒	鷔	麢	鼞	鼲	鬺	齺	齶	囔	攮
F	斸	欘	欙	欗	欚	灢	爦	犪	矘	矙	礹	籩	籫	糶	纚	

F9	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4	纘	纛	纙	臠	臡	虆	虉	虈	襹	襺	襼	襻	觿	讘	讙	躥
5	躤	躣	鑮	鑭	鑯	鑱	鑳	靉	顲	饟	鱨	鱮	鱭	鸋	鸍	鸐
6	鸏	鸒	鸎	麠	黵	鼉	齇	齸	齻	齹	蘁	圞	灦	籯	蠼	趲
7	躦	釃	鑴	鑸	鑶	鑵	驠	鱴	鱳	鱱	鱵	鸔	鸓	黶	鼊	
A		龤	灨	灥	糷	虪	蠾	蠽	蠿	讞	貜	躩	軉	靋	顳	顴
B	飌	饡	馫	驤	驦	驧	鬤	鸕	鸗	齈	贛	欞	爧	蘢	躨	钂
C	钀	钁	驩	驨	鬮	鸙	爩	虋	讟	钃	鱹	麷	癵	驫	鱺	鸝
D	灩	灪	麤	齾	齉	龘	碁	銹	裏	墻	恒	粧	嫺	╔	╦	╗
E	╠	╬	╣	╚	╩	╝	╒	╤	╕	╞	╪	╡	╘	╧	╛	╓
F	╥	╖	╟	╫	╢	╙	╨	╜	║	═	╭	╮	╰	╯	■	

FA	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
A																
B																
C																
D																
E																
F																

FB	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
A																
B																
C																
D																
E																
F																

FC	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
A																
B																
C																
D																
E																
F																

FD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4																
5																
6																
7																
A																
B																
C																
D																
E																
F																

Appendix F . Font Table - JIS Code

Please refer to the Japanese Kanji font table(RA8806_DS_V12_Font_JIS.pdf).