# SPECIFICATION

**PRODUCT : LCD MODULE**

**MODEL NO. : JLX12864G-091FW-BN**

**SUPPLIER : Shenzhen JLX Electronics CO., LTD**

**VERSION : A**

# CONTENTS

| WRITTEN BY / DATE | CHECKED BY / DATE | APPROVED BY / DATE |
|---|---|---|
| Shen | Yang | Ken Yip |
| May 19, 2020 | May 19, 2020 | May 19, 2020 |

# 1 BASIC SPECIFICATIONS

## 1.1 Display Specification

Table 1

| ITEM | SPECIFICATION |
|---|---|
| DISPLAY TYPE | FSTN/POSITIVE/TRANSFLECTIVE |
| COLOR | DISPLAY DOT: BLACK |
| | DISPLAY BACKGROUND:GRAY |
| DUTY | 1/64 |
| BIAS | 1/9 |
| VIEW ANGLE | 6 O'CLOCK |
| CONTROLLER | UC1701X |
| BACKLIGHT | LED (WHITE) |
| OPERATING TEMPERATURE | -20 °C ~70 °C |
| STORAGE TEMPERATURE | -30 °C ~ 80 °C |

## 1.2 Mechanical Specification

Table 2

| ITEM | SPECIFICATION | UNIT | NOTE |
|---|---|---|---|
| DIMENSIONAL OUTLINE | 43.5(L)*33.8(W)*3.5Max.(H) | mm | Excluding FPC |
| VIEW AREA | 39.8 (L)*25.5(W) | mm | |
| ACTIVE AREA | 37.09(L)*23.01(W) | mm | |
| DOT MATRIX | 128*64 | --- | |
| DOT SIZE | 0.26(W)*0.33(H) | mm | |
| DOT PITCH | 0.29(W)×0.36(H) | mm | |

## 1.3 Block Diagram

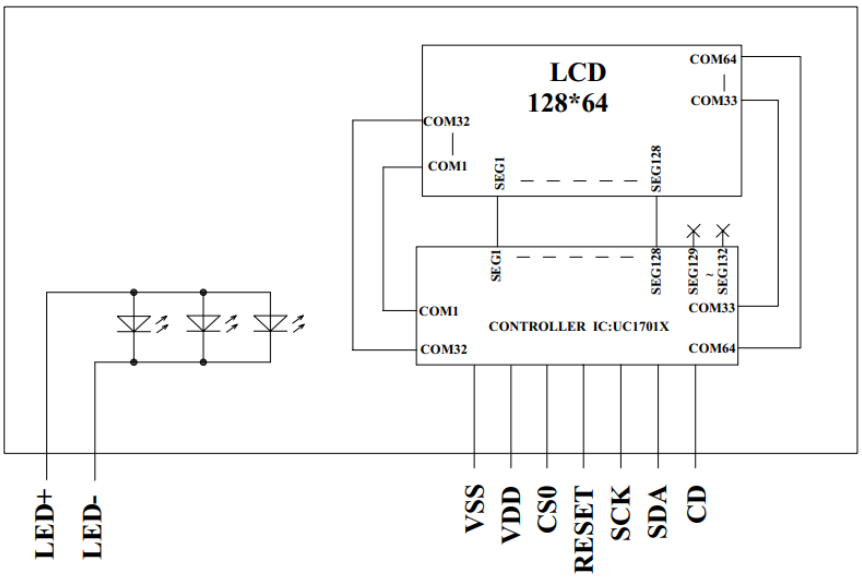
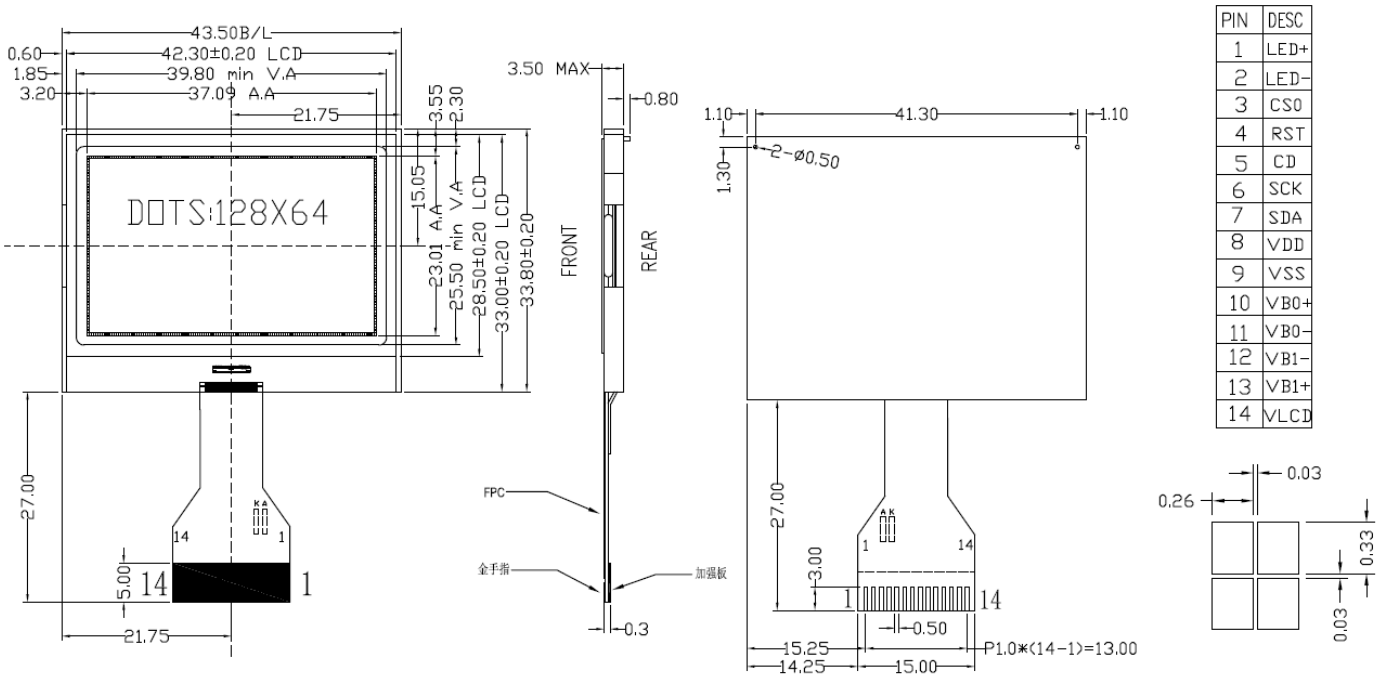



Figure 1 Block Diagram

## 1.4 Dimensions Outline




| PIN | DESC |
|---|---|
| 1 | LED+ |
| 2 | LED- |
| 3 | CS0 |
| 4 | RST |
| 5 | CD |
| 6 | SCK |
| 7 | SDA |
| 8 | VDD |
| 9 | VSS |
| 10 | VB0+ |
| 11 | VB0- |
| 12 | VB1- |
| 13 | VB1+ |
| 14 | VLCD |

Figure 2 Dimensions Outline

## 1.5 Terminal Function

Table 3

| Pin NO. | Symbol | Input/output | Level | Description |
|---|---|---|---|---|
| 1 | LED+ | I | H/L | Anode of LED backlight power supply. H: On, L: Off |
| 2 | LED- | I | 0V | Cathode of LED backlight power supply |
| 3 | CS0 | I | H/L | Chip select. Chip is selected when CS0='L'. |
| 4 | RESET | I | H/L | Hardware reset input pin. When RESET is "L", internal initialization is executed and the internal registers will be initialized. |
| 5 | CD | I | H/L | Register selects. H: Data code input; L: Control code input. |
| 6 | SCK | I | H/L | Serial clock input |
| 7 | SDA | I | H/L | Serial data input. |
| 8 | VDD | I | VDD | Power supply. |
| 9 | VSS | I | Ground | Ground. |
| 10 | VB0+ | | | LCD Bias Voltages. These voltages are generated internally. Connect a capacitor between VB0+ and VB0-. Connect a capacitor between VB1+ and VB1-. |
| 11 | VB0- | | | |
| 12 | VB1- | | | |
| 13 | VB1+ | | | |
| 14 | VLCD | | | Main LCD power supply. This voltage is generated internally. Connect a capacitor between VLCD and VSS. |

# 2 ABSOLUTE MAXIMUM RATING

Table 4 （Vss=0V）

| PARAMETER | SYMBOL | RATINGS | UNITS |
|---|---|---|---|
| POWER SUPPLY FOR LOGIC | $V_{DD}$-$V_{SS}$ | -0.3~ 4.0 | V |
| POWER SUPPLY FOR LCD DRIVER | $V_{LCD}$ | -0.3~ +13.2 | V |
| OPERATING TEMPERATURE | Topr | -20~70 | °C |
| STORAGE TEMPERATURE | Tstg | -30~80 | °C |

# 3 DC CHARACTERISTICS

Table 5 (Ta = –0 to 50 deg. C, Vss = 0 V)

| **CHARACTERISTIC** | **SYMBOL** | **TEST CONDITION** | **MIN** | **TYP** | **MAX** | **UNIT** |
|---|---|---|---|---|---|---|
| LOGIC CIRCUIT POWER SUPPLY VOLTAGE | $V_{DD}$ | | 2.5 | 3.3 | 3.6 | V |
| LCD OPERATION VOLTAGE | $V_{LCD}$ | | | | 11 | V |
| INPUT HIGH VOLTAGE | $V_{IH}$ | $V_{DD}$ = 3.3V | 0.8$V_{DD}$ | | | V |
| INPUT LOW VOLTAGE | $V_{IL}$ | | | | 0.2$V_{DD}$ | V |
| OUTPUT HIGH VOLTAGE | $V_{OH}$ | | 0.8$V_{DD}$ | | | V |
| OUTPUT VOLTAGE LOW | $V_{OL}$ | | | | 0.2$V_{DD}$ | V |
| LOGIC CIRCUIT POWER SUPPLY CURRENT | $I_{DD}$ | | | | 0.5 | mA |
| LED BACKLIGHT CURRENT | $I_{LED}$ | $V_{LED+}$=3.0V | 24 | 30 | 45 | mA |

# 4 TIMING CHARACTERISTICS

## 4.1 AC Characteristics

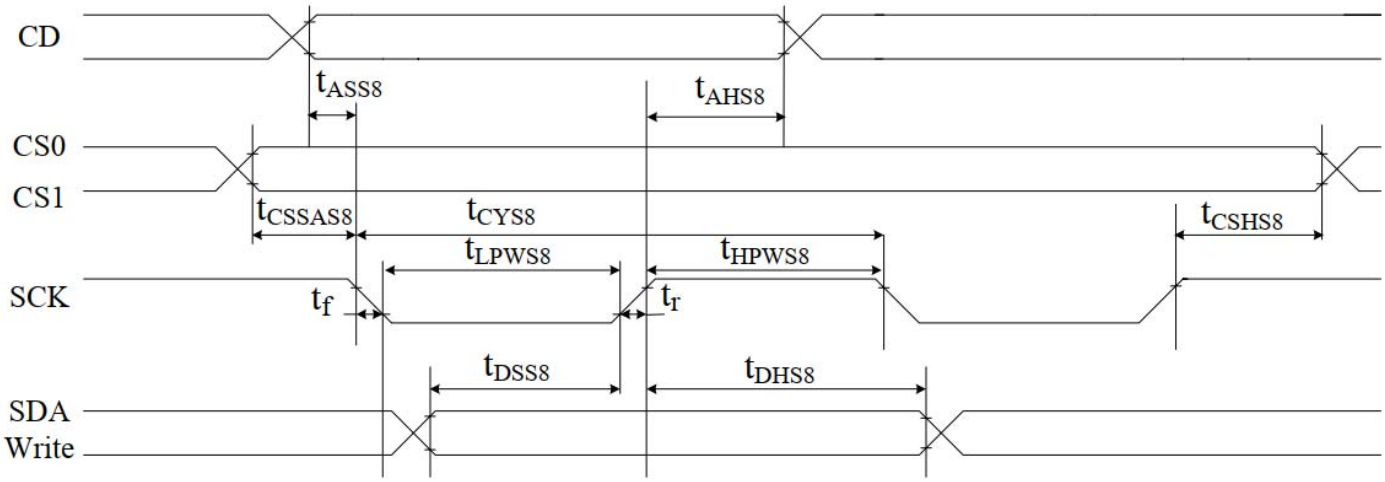



Figure 4

Table 6

| Symbol | Signal | Description | Condition | Min. | Max. | Unit |
|---|---|---|---|---|---|---|
| (2.5V ≤ $V_{DD}$ < 3.3V, Ta= –30 to +85°C) | | | | (Read / Write) | | |
| $t_{ASS8}$<br>$t_{AHS8}$ | CD | Address setup time<br>Address hold time | | 0<br>0 | – | nS |
| $t_{CSSAS8}$<br>$t_{CSHS8}$ | CS1/CS0 | Chip select setup time<br>Chip select hold time | | 5<br>5 | – | nS |
| $t_{CYS8}$ | SCK | Cycle time | | 130 / 60 | – | nS |
| $t_{LPWS8}$ | | Low pulse width | | 50 / 15 | | |
| $t_{HPWS8}$ | | High pulse width | | 50 / 15 | | |
| $t_{DSS8}$<br>$t_{DHS8}$ | SDA<br>(Write) | Data setup time<br>Data hold time | | --- / 12<br>--- / 0 | – | nS |
| (1.65V ≤ $V_{DD}$ < 2.5V, Ta= –30 to +85°C) | | | | (Read / Write) | | |
| $t_{ASS8}$<br>$t_{AHS8}$ | CD | Address setup time<br>Address hold time | | 0<br>0 | – | nS |
| $t_{CSSAS8}$<br>$t_{CSHS8}$ | CS1/CS0 | Chip select setup time<br>Chip select hold time | | 10<br>10 | – | nS |
| $t_{CYS8}$ | SCK | Cycle time | | 160 / 90 | – | nS |
| $t_{LPWS8}$ | | Low pulse width | | 65 / 30 | | |
| $t_{HPWS8}$ | | High pulse width | | 65 / 30 | | |
| $t_{DSS8}$<br>$t_{DHS8}$ | SDA<br>(Write) | Data setup time<br>Data hold time | | --- / 24<br>--- / 0 | – | nS |

**Note:** tr (rising time), tf (falling time) : ≤ 15nS

## 4.2 AC Characteristics of 'RESET'

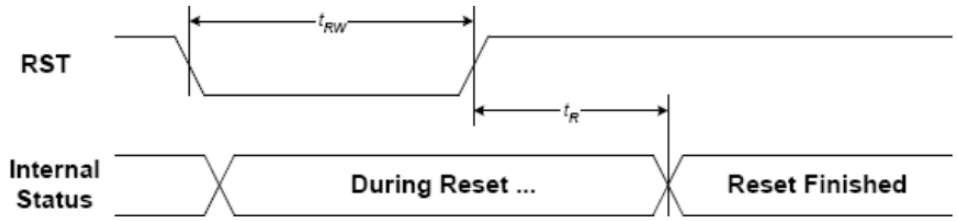



Figure 5

| Symbol | Signal | Description | Condition | Min. | Max. | Unit |
|---|---|---|---|---|---|---|
| $t_{RW}$ | RST | Reset low pulse width | | 3 | – | μS |
| $t_R$ | RST,, Internal Status | Reset to Internal Status pulse delay | | 6 | – | mS |

# 5 COMMANDS AND FUNCTION DESCRIPTIONS

## 5.1 Command Table

TABLE 7

The following is a list of host commands supported by UC1701x

C/D: 0: Control, 1: Data
W/R: 0: Write Cycle, 1: Read Cycle

# Useful Data bits – Don't Care

| | Command | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Action | Default |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1. | Write Data Byte | 1 | 0 | # | # | # | # | # | # | # | # | Write 1 byte | N/A |
| 2. | Read Data Byte | 1 | 1 | # | # | # | # | # | # | # | # | Read 1 byte | N/A |
| 3. | Get Status | 0 | 1 | BZ | MX | DE | RST | 0 | 0 | 0 | 0 | Get Status | -- |
| 4. | Set Column Address LSB | 0 | 0 | 0 | 0 | 0 | 0 | # | # | # | # | Set CA [3:0] | 0 |
| | Set Column Address MSB | | | 0 | 0 | 0 | 1 | # | # | # | # | Set CA [7:4] | 0 |
| 5. | Set Power Control | 0 | 0 | 0 | 0 | 1 | 0 | 1 | # | # | # | Set PC[2:0] | 000b |
| 6. | Set Scroll Line | 0 | 0 | 0 | 1 | # | # | # | # | # | # | Set SL[5:0] | 0 |
| 7. | Set Page Address | 0 | 0 | 1 | 0 | 1 | 1 | # | # | # | # | Set PA[3:0] | 0 |
| 8. | Set $V_{LCD}$ Resistor Ratio | 0 | 0 | 0 | 0 | 1 | 0 | 0 | # | # | # | Set PC[5:3] | 110b |
| 9. | Set Electronic Volume (double-byte command) | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | Set PM[5:0] | 10H |
| | | | | 0 | 0 | # | # | # | # | # | # | | |
| 10. | Set All-Pixel-ON | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | # | Set DC[1] | 0b |
| 11. | Set Inverse Display | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | # | Set DC[0] | 0b |
| 12. | Set Display Enable | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | # | Set DC[2] | 0b |
| 13. | Set SEG Direction | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | # | Set LC[0] | 0b |
| 14. | Set COM Direction | 0 | 0 | 1 | 1 | 0 | 0 | # | - | - | - | Set LC[1] | 1b: MY ON |
| 15. | System Reset | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | Software Reset | N/A |
| 16. | NOP | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | No operation | N/A |
| 17. | Set LCD Bias Ratio | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | # | Set BR | 0b |
| 18. | Set Cursor Update Mode | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | AC3=1, CR=CA | N/A |
| 19. | Reset Cursor Update Mode | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | AC3=0, CA=CR. | N/A |
| 20. | Set Static Indicator OFF | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | NOP | N/A |
| 21. | Set Static Indicator ON | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | NOP | N/A |
| | Set Static Indicator | | | - | - | - | - | - | - | - | - | | |
| 22. | Set Booster Ratio (double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | NOP | 00b |
| | | | | 0 | 0 | 0 | 0 | 0 | 0 | # | # | | |
| 23. | Set Power Save (compound command) | 0 | 0 | # | # | # | # | # | # | # | # | Display OFF & All Pixel ON | N/A |
| 24. | Set Test Control (double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | TT | | For UCI only Do NOT use | N/A |
| | | | | - | # | # | # | # | # | # | # | | |
| 25. | Set Adv. Program Control 0 (double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | Set TC, WA[1:0] | 90H |
| | | | | # | 0 | 0 | 1 | 0 | 0 | # | # | | |
| 26. | Set Adv. Program Control 1 (double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | For UCI only Set APC1 | N/A |
| | | | | # | # | # | # | # | # | # | # | | |

* Other than commands listed above, all other bit patterns result in NOP (No Operation).

## 5.2 Function Descriptions

### 5.2.1 Write Data Byte to Memory

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Write data | 1 | 0 | 8-bit data write to SRAM | | | | | | | |

### 5.2.2 Read Data Byte from Memory

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Read data | 1 | 1 | 8-bit data read from SRAM | | | | | | | |

Write/Read Data Byte (Command 1,2) access Display Data RAM based on Page Address (PA) register and Column Address (CA) register. PA and CA can also be programmed directly by issuing *Set Page Address* and *Set Column Address* commands.

### 5.2.3 Get Status

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Get Status | 0 | 1 | BZ | MX | DE | RST | 0 | 0 | 0 | 0 |

*BZ:* BZ=1 when busy. The system accepts commands only when BZ=0.
*MX*: Mirror X. Status of register LC[0]
*DE*: Display Enable flag. DE=1 when display is enabled.
*RST*: RST flag. RST=1 when reset is in progress.

### 5.2.4 Set Column Address

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Column Address LSB, CA[3:0] | 0 | 0 | 0 | 0 | 0 | 0 | CA3 | CA2 | CA1 | CA0 |
| Set Column Address MSB, CA[7:4] | 0 | 0 | 0 | 0 | 0 | 1 | CA7 | CA6 | CA5 | CA4 |

Set the SRAM column address before Write/Read memory from host interface.

CA value range: 0~131

### 5.2.5 Set Power control

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Power Control, PC[2:0] | 0 | 0 | 0 | 0 | 1 | 0 | 1 | PC2 | PC1 | PC0 |

Set PC[2:0] to enable the built-in charge pump.

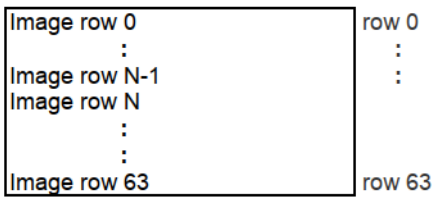
PC[2] : 0 – Boost OFF 1 – Boost ON
PC[1] : 0 – Voltage Regular OFF 1 – Voltage Regular ON
PC[0] : 0 – Voltage Follower OFF 1 – Voltage Follower ON

### 5.2.6 Set Scroll Line

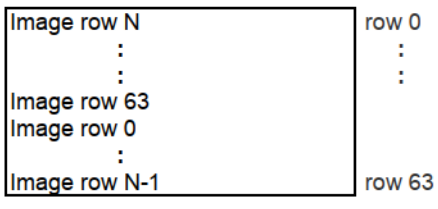
| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Scroll Line, SL[5:0] | 0 | 0 | 0 | 1 | SL5 | SL4 | SL3 | SL2 | SL1 | SL0 |

Set the scroll line number. Range : 0~63

Scroll line setting will scroll the displayed image up by *SL* rows. Icon output CIC will not be affected by `Set Scroll Line` command.




SL=0




SL=N

### 5.2.7 Set Page Address

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Page Address, PA[3:0] | 0 | 0 | 1 | 0 | 1 | 1 | PA3 | PA2 | PA1 | PA0 |

Set the SRAM page address before write/read memory from host interface. Each page of SRAM corresponds to 8 COM lines on LCD panel, except for the last page. The last page corresponds to the icon output CIC.

Possible value = 0~8.

### 5.2.8 Set $V_{LCD}$ Resistor Ratio

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set $V_{LCD}$ Resistor Ratio, PC[5:3] | 0 | 0 | 0 | 0 | 1 | 0 | 0 | PC5 | PC4 | PC3 |

Configure PC[5:3] to set internal Resistor Ratio, Rb/Ra, for the $V_{LCD}$ Voltage regulator to adjust the contrast of the display panel:

PC[5:3] : 000b~111b – 1+Rb/Ra ratio. **Default : 110b**. Refer to $V_{LCD}$ Quick Reference for "1+Rb/Ra" ratio.

$V_{LCD}=((1+Rb/Ra) \times Vev) \times (1+(T-25)xC_T\%)$ $Vev=(1-(63-PM)/162)xV_{REF}$

where Rb and Ra are internal resistors,
$V_{REF}$ is on-chip contrast voltage, and
PM is a vaule of electronic volume

### 5.2.9 Set Electronic Volume

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Electronic Volume, PM[5:0] | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | | | 0 | 0 | PM5 | PM4 | PM3 | PM2 | PM1 | PM0 |

Set PM[5:0] for electronic volume "PM" for VLCD voltage regulator to adjust contrast of LCD panel display

Effective range : 0~63. **Default : 16 (10h)**

### 5.2.10 Set All Pixel ON

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set All Pixel ON, DC [1] | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | DC1 |

Set DC[1] to force all SEG drivers to output ON signals. This function has no effect on the existing data stored in display RAM. **Default : 0**

### 5.2.11 Set Inverse Display

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Inverse Display, DC [0] | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | DC0 |

Set DC[0] to force all SEG drivers to output the inverse of the data (bit-wise) stored in display RAM. This function has no effect on the existing data stored in display RAM.

### 5.2.12 Set Display Enable

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Display Enable, DC[2] | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | DC2 |

This command is for programming register DC[2]. When DC[2] is set to 1, UC1701x will first exit from sleep mode, restore the power and then turn on COM drivers and SEG drivers.

### 5.2.13 Set SEG Direction

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Segment Direction, LC[0] | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | MX |

Set LC[0] for SEG (column) mirror (MX). **Default : 0**

MX is implemented by reversing the mapping order between RAM and SEG (column) electrodes. The data stored in RAM is not affected by MX command. Yet, MX has immediate effect on the display image.

### 5.2.14 Set COM Direction

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Common Direction, LC[1] | 0 | 0 | 1 | 1 | 0 | 0 | MY | - | - | - |

Set LC[1] for COM (row) mirror (MY). **Default : 1b**

MY is implemented by reversing the mapping between RAM and COM (row) electrodes. The data stored in RAM is not affected by MY command. Yet, MY has immediate effect on the display image.

### 5.2.15 System Reset(Software Reset)

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Software Reset | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

This command will activate the system reset.

Some control register values will be reset to their default values. Yet, data stored in RAM will not be affected. See the “Reset and Power Management” section for more details.

### 5.2.16 NOP

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| No Operation | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |

This command is used for “no operation”.

### 5.2.17 Set LCD Bias Ratio

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Bias Ratio, BR | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | BR |

Select voltage bias ratio required for LCD. **Default : 0**

The setting of Bias ratio varies according to Duty:

| DUTY | BR = 0 | BR = 1 |
|---|---|---|
| 1/65 | 1/9 | 1/7 |
| 1/49 | 1/8 | 1/6 |
| 1/33 | 1/6 | 1/5 |
| 1/55 | 1/8 | 1/6 |

### 5.2.18 Set Cursor Update Mode

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Cursor Update Mode | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |

This command is used for set cursor update mode function. When cursor update mode sets, UC1701x will update register CR with the value of register CA. The column address CA will increment with write RAM data operation but the address wraps around will be suspended no matter what WA setting is. However, the column address will not increment in read RAM data operation. The set cursor update mode can be used to implement “write after read RAM” function. The column address (CA) will be restored to the value, which is before the set cursor update mode command, when reset cursor update mode.

The purpose of this pair commands and their feature is to support “write after read” function for cursor implementation.

### 5.2.19 Reset Cursor Update Mode

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Reset Cursor Update Mode | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 |

Set AC3=0 and CA=CR.

### 5.2.20 Set Static Indicator OFF

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Turn OFF Static Indicator | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |

No Operation.

### 5.2.21 Set Static Indicator ON

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Turn ON Static Indicator | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 0 | 0 | - | - | - | - | - | - | - | - |

No Operation.

### 5.2.22 Set Booster Ratio

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Booster Ratio (Double-byte command) | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| | | | 0 | 0 | 0 | 0 | 0 | 0 | - | - |

This command is used for "No Operation".

### 5.2.23 Set Power Save

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Power Save (Compound Command) | 0 | 0 | # | # | # | # | # | # | # | # |

### 5.2.24 Set Test Control

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set TT (Double-byte command) | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | TT | |
| | | | - | # | # | # | # | # | # | # |

This command is for UltraChip's Test only. Do NOT use.

### 5.2.25 Set Advanced Program Control 0

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Adv. Program Control, APC0 [7:0] (Double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 |
| | | | TC | 0 | 0 | 1 | 0 | 0 | WA1 | WA0 |

TC : APC0 [7], $V_{BIAS}$ temperature compensation coefficient (%-per-degree-C)

Temperature compensation curve definition:
TC : 0b = -0.05%/°C, **1b = -0.11%/°C**

WA : APC0 [1:0], Automatic column/row wrap around.
WA[0] : **0: PA WA disable** 1: PA WA enable.
WA[1] : **0: CA WA disable** 1: CA WA enable.

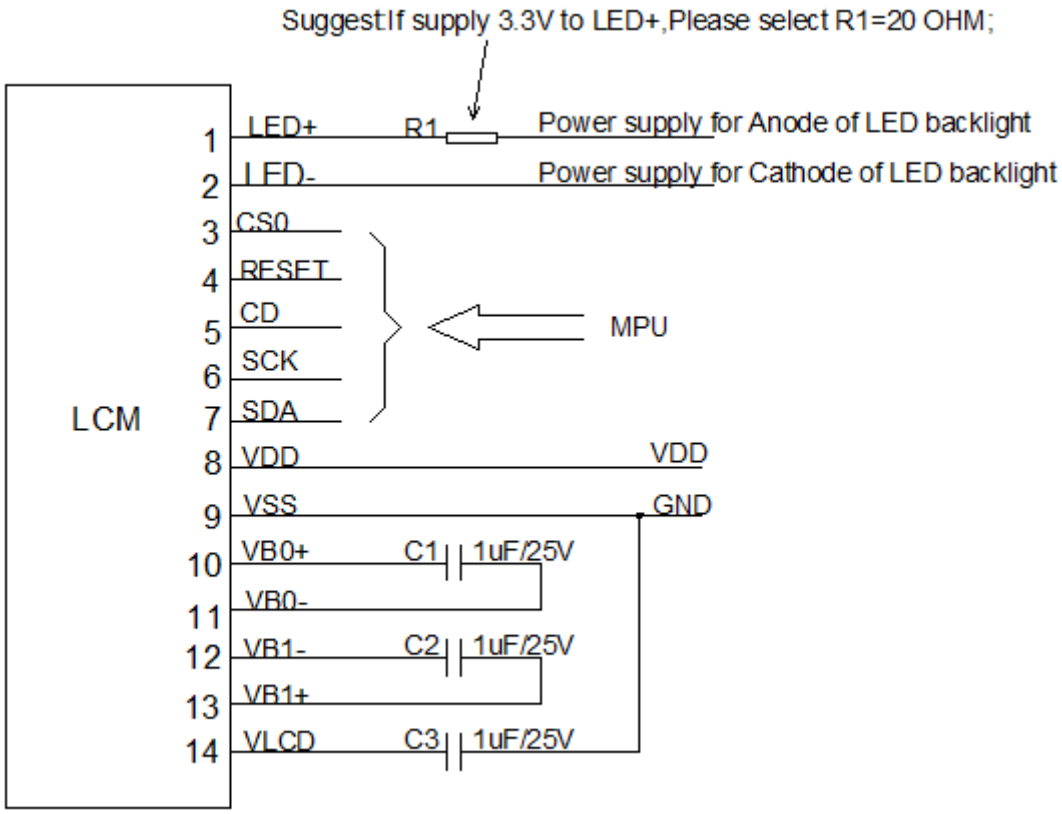
### 5.2.26 Set Advanced Program Control 1

| Action | C/D | W/R | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 |
|---|---|---|---|---|---|---|---|---|---|---|
| Set Adv. Program Control, APC1 [7:0] (Double-byte command) | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 |
| | | | APC1 register parameter | | | | | | | |

For UltraChip only. Please Do *NOT* use.

# 6 REFERENCE CIRCUIT AND PROGRAM

## 6.1 Reference Circuit




## 6.2 Reference Program

```
//REFERENCE PROGRAM FOR LCM OF JLX12864G-091
//LCD CONTROLLER IC:UC1701X
//March 10,2020

#include <reg52.H>
#include <intrins.h>

sbit cs0=P1^1;
sbit cd=P3^0;
sbit reset=P1^0;
sbit sck=P3^2;
sbit sda=P3^1;

sbit key=P2^0;

#define uchar unsigned char
#define uint unsigned int

uchar code BMP12864_PANDA[];
uchar code number_16x32_0[];
uchar code number_16x32_1[];
uchar code number_16x32_2[];
```

```
uchar code number_16x32_3[];
uchar code number_16x32_4[];
uchar code number_16x32_5[];
uchar code number_16x32_6[];
uchar code number_16x32_7[];
uchar code number_16x32_8[];
uchar code number_16x32_9[];

uchar code ascii_table_8x16[95][16];
uchar code ascii_table_5x8[95][5];



//WRITE COMMAND TO LCM
void transfer_command(int data1)
{
    char i;
    cs0=0;
    cd=0;
    for(i=0;i<8;i++)
    {
        sck=0;
        if(data1&0x80) sda=1;
        else sda=0;
        sck=1;
        data1=data1<<=1;
    }
    cs0=1;
}

//WRITE DATA TO LCM
void transfer_data(int data1)
{
    char i;
    cs0=0;
    cd=1;
    for(i=0;i<8;i++)
    {
        sck=0;
        if(data1&0x80) sda=1;
        else sda=0;
        sck=1;
        data1=data1<<=1;
    }
    cs0=1;
}

//DELAY
void delay(int i)
{
 int j,k;
 for(j=0;j<i;j++)
 for(k=0;k<990;k++);
}

//Wait a key
void waitkey()
{
repeat:  if(key==1)    goto repeat;
         else delay(600);
}



//INITIALIZE LCM
void initial_lcd()
{
    cs0=0;
    reset=0;              //Hardware Reset
    delay(100);
    reset=1;          //Reset finished
```

```
    delay(20);
    transfer_command(0xe2);    //Software Reset
    delay(5);

    transfer_command(0x2f);  //To enable the built-in charge pump.
    delay(5);
    transfer_command(0x23);  //Set internal resistor ratio to adjust the contrast. Adjust range:0x20~0x27.
    transfer_command(0x81);  //Fine tune contrast,don't change "0x81"
    transfer_command(0x28);  //Fine tune contrast,adjust range:0x00~0x3f
    transfer_command(0xa2);  //Set bias=1/9
    transfer_command(0xc8);  //Row scan direction:from top to bottom.
    transfer_command(0xa0);  //Column scan direction:from left to right.
    transfer_command(0x40);  //Set start line
    transfer_command(0xaf);  //Turn on display
    cs0=1;
}

//SET PAGE ADDRESS AND COLUMN ADDRESS OF LCM
void lcd_address(uchar page,uchar column)
{
    cs0=0;
    column=column-1;
    page=page-1;
    transfer_command(0xb0+page);
    transfer_command(((column>>4)&0x0f)+0x10);
    transfer_command(column&0x0f);
}

//CLEAR THE LCD SCREEN
void clear_screen()
{
    unsigned char i,j;
    cs0=0;
    for(i=0;i<9;i++)
    {
        lcd_address(1+i,1);
        for(j=0;j<132;j++)
        {
            transfer_data(0x00);
        }
    }
    cs0=1;
}


//Display graphic of 128x64 dot matrix
void display_graphic_128x64(uchar *dp)
{
    uchar i,j;
    cs0=0;
    for(j=0;j<8;j++)
    {
        lcd_address(1+j,1);
        for (i=0;i<128;i++)
        {
            transfer_data(*dp);   //Write the data to the LCD, and the column address will be increased by 1 automatically
            dp++;
        }
    }
    cs0=1;
}


//Display graphic of 16x32 dot matrix
void display_graphic_16x32(uchar page,uchar column,uchar *dp)
{
    uchar i,j;
    cs0=0;
    for(j=0;j<4;j++)
    {
        lcd_address(page+j,column);
```

```
        for (i=0;i<16;i++)
        {
            transfer_data(*dp);   //Write the data to the LCD, and the column address will be increased by 1 automatically
            dp++;
        }
    }
    cs0=1;
}

//Display graphic of 16x16 dot matrix
void display_graphic_16x16(uchar page,uchar column,uchar *dp)
{
    uchar i,j;

    cs0=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for (i=0;i<16;i++)
        {
            transfer_data(*dp);//Write the data to the LCD, and the column address will be increased by 1 automatically
            dp++;
        }
    }
    cs0=1;
}

//Display graphic of 8x16 dot matrix
void display_graphic_8x16(uchar page,uchar column,uchar *dp)
{
    uchar i,j;
    cs0=0;
    for(j=0;j<2;j++)
    {
        lcd_address(page+j,column);
        for (i=0;i<8;i++)
        {
            transfer_data(*dp);//Write the data to the LCD, and the column address will be increased by 1 automatically
            dp++;
        }
    }
    cs0=1;
}


//Display ASCII string of 8x16 dot matrix
void display_string_8x16(uint page,uint column,uchar *text)
{
    uint i=0,j,k,n;
    cs0=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<=0x7e))
        {
            j=text[i]-0x20;
            for(n=0;n<2;n++)
            {
                lcd_address(page+n,column);
                for(k=0;k<8;k++)
                {
                    transfer_data(ascii_table_8x16[j][k+8*n]);//Write the data to the LCD, and the column address will be increased by 1 automatically
                }
            }
            i++;
            column+=8;

        }
        else
        i++;
    }
}
```

```
//Display ASCII string of 5x8 dot matrix
void display_string_5x8(uint page,uint column,uchar *text)
{
    uint i=0,j,k;
    cs0=0;
    while(text[i]>0x00)
    {
        if((text[i]>=0x20)&&(text[i]<0x7e))
        {
            j=text[i]-0x20;
            lcd_address(page,column);
            for(k=0;k<5;k++)
            {
                transfer_data(ascii_table_5x8[j][k]);//Write the data to the LCD, and the column address will be increased by 1 automatically
            }
            transfer_data(0x00);//Insert a column of spaces between characters
            i++;
            column+=6;
        }
        else
        i++;
    }
}



void main(void)
 {

    while(1)
    {
        initial_lcd();

        clear_screen();
        display_graphic_128x64(BMP12864_PANDA);     //Display a graphic of 128x64 dot matrix
        delay(1000);
        //waitkey();
        clear_screen();                                  //clear all dots
        display_graphic_16x32(3, 1,number_16x32_0);     //Display a graphic of 16x32 dot matrix
        display_graphic_16x32(3,17,number_16x32_1);     //Display a graphic of 16x32 dot matrix
        display_graphic_16x32(3,33,number_16x32_2);     //Display a graphic of 16x32 dot matrix
        display_graphic_16x32(3,49,number_16x32_3);     //Display a graphic of 16x32 dot matrix

        delay(1000);
        //waitkey();
        clear_screen();                                  //clear all dots
        display_string_8x16(1,1," !\"#$%&'()*+,-./");   //Display ASCII string of 8x16 dot matrix
        display_string_8x16(3,1,"0123456789:;<=>?");
        display_string_8x16(5,1,"@ABCDEFGHIJKLMNO");
        display_string_8x16(7,1,"PQRSTUVWXYZ[\\]^_");
        delay(1000);
        //waitkey();
        clear_screen();
        display_string_8x16(1,1,"`abcdefghijklmno");
        display_string_8x16(3,1,"pqrstuvwxyz{|}~");

        delay(1000);
        //waitkey();                                      //clear all dots
        clear_screen();
        display_string_5x8(1,1," !\"#$%&'()*+,-./01234"); //Display ASCII string of 5x8 dot matrix
        display_string_5x8(2,1,"56789:;<=>?@ABCDEFGHI");
        display_string_5x8(3,1,"JKLMNOPQRSTUVWXYZ[\\]^");
        display_string_5x8(4,1,"_`abcdefghijklmnopqrs");
        display_string_5x8(5,1,"tuvwxyz{|}~");
        delay(1000);
        //waitkey();

    }
}
```

```
uchar code number_16x32_0[]={
/*--Character:0, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0x00,0x80,0xC0,0xC0,0x40,0x40,0xC0,0xC0,0x80,0x00,0x00,0x00,0x00,
0x00,0xF8,0xFE,0xFF,0x7F,0x03,0x00,0x00,0x00,0x00,0x03,0xFF,0xFF,0xFE,0xF8,0x00,
0x00,0x3F,0xFF,0xFF,0xFC,0x80,0x00,0x00,0x00,0x00,0x80,0xFF,0xFF,0xFF,0x1F,0x00,
0x00,0x00,0x00,0x01,0x03,0x07,0x06,0x04,0x04,0x06,0x07,0x03,0x01,0x00,0x00,0x00,
};
uchar code number_16x32_1[]={
/*--Character:1, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x80,0x80,0xC0,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x01,0x01,0x01,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x04,0x04,0x04,0x07,0x07,0x07,0x07,0x04,0x04,0x04,0x00,0x00,0x00,
};
uchar code number_16x32_2[]={
/*--Character:2, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0x80,0x80,0xC0,0x40,0x40,0x40,0xC0,0xC0,0xC0,0x80,0x00,0x00,0x00,
0x00,0x0C,0x1F,0x1F,0x1F,0x08,0x00,0x00,0x00,0x80,0xE1,0xFF,0xFF,0x7F,0x00,0x00,
0x00,0x00,0x80,0xC0,0xE0,0x70,0x38,0x1C,0x0E,0x07,0x03,0x01,0xC0,0xC0,0x40,0x00,
0x00,0x07,0x07,0x07,0x06,0x06,0x06,0x06,0x06,0x06,0x06,0x07,0x07,0x07,0x00,0x00,
};
uchar code number_16x32_3[]={
/*--Character:3, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0x80,0xC0,0xC0,0x40,0x40,0xC0,0xC0,0xC0,0x80,0x00,0x00,0x00,0x00,
0x00,0x00,0x0F,0x0F,0x0F,0x80,0x80,0x80,0xC0,0xE1,0xFF,0x7F,0x3F,0x0E,0x00,0x00,
0x00,0xC0,0xE0,0xE0,0xE0,0x00,0x00,0x00,0x00,0x01,0x03,0xFF,0xFE,0xFC,0x70,0x00,
0x00,0x00,0x03,0x03,0x07,0x06,0x04,0x04,0x04,0x06,0x07,0x03,0x03,0x01,0x00,0x00,
};
uchar code number_16x32_4[]={
/*--Character:4, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x80,0xC0,0xC0,0xC0,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x80,0xE0,0x78,0x3C,0x0F,0xFF,0xFF,0xFF,0xFF,0x00,0x00,0x00,0x00,
0x08,0x0C,0x0F,0x0F,0x09,0x08,0x08,0x08,0xFF,0xFF,0xFF,0xFF,0x08,0x08,0x08,0x00,
0x00,0x00,0x00,0x00,0x00,0x04,0x04,0x04,0x07,0x07,0x07,0x07,0x04,0x04,0x04,0x00,
};
uchar code number_16x32_5[]={
/*--Character:5, HxV=16x32 dot matrix  --*/
0x00,0x00,0x00,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0x00,0x00,
0x00,0x00,0xFE,0xFF,0xFF,0x60,0x60,0x20,0x20,0x60,0xE0,0xE0,0xC0,0x80,0x00,0x00,
0x00,0xC0,0xE1,0xE1,0xE1,0x00,0x00,0x00,0x00,0x00,0x80,0xFF,0xFF,0xFF,0x3C,0x00,
0x00,0x00,0x01,0x03,0x07,0x06,0x04,0x04,0x04,0x06,0x07,0x03,0x03,0x01,0x00,0x00,
};
uchar code number_16x32_6[]={
/*--Character:6, HxV=16x32  --*/
0x00,0x00,0x00,0x00,0x80,0x80,0xC0,0xC0,0x40,0x40,0xC0,0xC0,0x80,0x00,0x00,0x00,
0x00,0xF0,0xFE,0xFF,0xFF,0x83,0xC0,0x40,0x40,0xC0,0xC3,0xC3,0x83,0x03,0x00,0x00,
0x00,0x7F,0xFF,0xFF,0xF7,0x01,0x00,0x00,0x00,0x00,0x00,0xFF,0xFF,0xFF,0x7E,0x00,
0x00,0x00,0x01,0x03,0x03,0x07,0x06,0x04,0x04,0x04,0x06,0x03,0x03,0x01,0x00,0x00,
};
uchar code number_16x32_7[]={
/*--Character:7, HxV=16x32  --*/
0x00,0x00,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0xC0,0x00,0x00,
0x00,0x06,0x07,0x07,0x01,0x00,0x00,0x80,0xE0,0xF8,0x7E,0x0F,0x03,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0xF0,0xFE,0xFF,0xFF,0x03,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x07,0x07,0x07,0x07,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
};
uchar code number_16x32_8[]={
/*--Character:8, HxV=16x32  --*/
0x00,0x00,0x00,0x80,0xC0,0xC0,0x40,0x40,0x40,0x40,0xC0,0x80,0x80,0x00,0x00,0x00,
0x00,0x1E,0x3F,0x7F,0xF3,0xE0,0xC0,0xC0,0x80,0x80,0xC0,0x7F,0x7F,0x3F,0x0C,0x00,
0x00,0xF8,0xFE,0xFE,0x07,0x03,0x01,0x01,0x03,0x03,0x07,0xDF,0xFE,0xFC,0x70,0x00,
0x00,0x00,0x01,0x03,0x07,0x06,0x04,0x04,0x04,0x04,0x06,0x03,0x03,0x01,0x00,0x00,
};
uchar code number_16x32_9[]={
/*--Character:9, HxV=16x32  --*/
0x00,0x00,0x00,0x80,0xC0,0xC0,0x40,0x40,0x40,0xC0,0xC0,0x80,0x00,0x00,0x00,0x00,
0x00,0xFE,0xFF,0xFF,0x87,0x00,0x00,0x00,0x00,0x00,0x01,0xFF,0xFF,0xFE,0xF8,0x00,
0x00,0x01,0x83,0x83,0x87,0x86,0x04,0x04,0x04,0x06,0xE3,0xFF,0xFF,0x7F,0x0F,0x00,
0x00,0x00,0x03,0x07,0x07,0x07,0x04,0x04,0x06,0x07,0x03,0x01,0x00,0x00,0x00,0x00,
};
```

```
uchar code ascii_table_8x16[95][16]={

//ASCII CODE:8x16 DOT MATRIX
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,     //- -(SPACE) ASCII CODE: 0X20
0x00,0x00,0x38,0xFC, 0xFC,0x38,0x00,0x00, 0x00,0x00,0x00,0x0D, 0x0D,0x00,0x00,0x00,     //-!-            ASCII CODE: 0X21
0x00,0x0E,0x1E,0x00, 0x00,0x1E,0x0E,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,         //-"-
0x20,0xF8,0xF8,0x20, 0xF8,0xF8,0x20,0x00, 0x02,0x0F,0x0F,0x02, 0x0F,0x0F,0x02,0x00,         //-#-
0x38,0x7C,0x44,0x47, 0x47,0xCC,0x98,0x00, 0x06,0x0C,0x08,0x38, 0x38,0x0F,0x07,0x00,         //-$-
0x30,0x30,0x00,0x80, 0xC0,0x60,0x30,0x00, 0x0C,0x06,0x03,0x01, 0x00,0x0C,0x0C,0x00,         //-%-
0x80,0xD8,0x7C,0xE4, 0xBC,0xD8,0x40,0x00, 0x07,0x0F,0x08,0x08, 0x07,0x0F,0x08,0x00,     //-&-
0x00,0x10,0x1E,0x0E, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,     //-'-
0x00,0x00,0xF0,0xF8, 0x0C,0x04,0x00,0x00, 0x00,0x00,0x03,0x07, 0x0C,0x08,0x00,0x00,     //-(-
0x00,0x00,0x04,0x0C, 0xF8,0xF0,0x00,0x00, 0x00,0x00,0x08,0x0C, 0x07,0x03,0x00,0x00,     //-)-

0x80,0xA0,0xE0,0xC0, 0xC0,0xE0,0xA0,0x80, 0x00,0x02,0x03,0x01, 0x01,0x03,0x02,0x00,     //-*-            ASCII CODE: 0X2A
0x00,0x80,0x80,0xE0, 0xE0,0x80,0x80,0x00, 0x00,0x00,0x00,0x03, 0x03,0x00,0x00,0x00,     //-+-
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x10,0x1E, 0x0E,0x00,0x00,0x00,     //-,-
0x80,0x80,0x80,0x80, 0x80,0x80,0x80,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00,     //---
0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x0C, 0x0C,0x00,0x00,0x00,     //-.-
0x00,0x00,0x00,0x80, 0xC0,0x60,0x30,0x00, 0x0C,0x06,0x03,0x01, 0x00,0x00,0x00,0x00,     //-/-
0xF8,0xF8,0x0C,0xC4, 0x0C,0xF8,0xF0,0x00, 0x03,0x07,0x0C,0x08, 0x0C,0x07,0x03,0x00,     //-0-            ASCII CODE: 0X30
0x00,0x10,0x18,0xFC, 0xFC,0x00,0x00,0x00, 0x00,0x08,0x08,0x0F, 0x0F,0x08,0x08,0x00,     //-1-
0x08,0x0C,0x84,0xC4, 0x64,0x3C,0x18,0x00, 0x0E,0x0F,0x09,0x08, 0x08,0x0C,0x0C,0x00,     //-2-
0x08,0x0C,0x44,0x44, 0x44,0xFC,0xB8,0x00, 0x04,0x0C,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-3-

0xC0,0xE0,0xB0,0x98, 0xFC,0xFC,0x80,0x00, 0x00,0x00,0x00,0x08, 0x0F,0x0F,0x08,0x00,     //-4-            ASCII CODE: 0X34
0x7C,0x7C,0x44,0x44, 0x44,0xC4,0x84,0x00, 0x04,0x0C,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-5-
0xF0,0xF8,0x4C,0x44, 0x44,0xC0,0x80,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-6-
0x0C,0x0C,0x04,0x84, 0xC4,0x7C,0x3C,0x00, 0x00,0x00,0x0F,0x0F, 0x00,0x00,0x00,0x00,     //-7-
0xB8,0xFC,0x44,0x44, 0x44,0xFC,0xB8,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-8-
0x38,0x7C,0x44,0x44, 0x44,0xFC,0xF8,0x00, 0x00,0x08,0x08,0x08, 0x0C,0x07,0x03,0x00,     //-9-
0x00,0x00,0x00,0x30, 0x30,0x00,0x00,0x00, 0x00,0x00,0x00,0x06, 0x06,0x00,0x00,0x00,     //-:-
0x00,0x00,0x00,0x30, 0x30,0x00,0x00,0x00, 0x00,0x00,0x08,0x0E, 0x06,0x00,0x00,0x00,     //-;-
0x00,0x80,0xC0,0x60, 0x30,0x18,0x08,0x00, 0x00,0x00,0x01,0x03, 0x06,0x0C,0x08,0x00,     //-<-
0x00,0x20,0x20,0x20, 0x20,0x20,0x20,0x00, 0x00,0x01,0x01,0x01, 0x01,0x01,0x01,0x00,     //-=-

0x00,0x08,0x18,0x30, 0x60,0xC0,0x80,0x00, 0x00,0x08,0x0C,0x06, 0x03,0x01,0x00,0x00,     //->-            ASCII CODE: 0X3E
0x18,0x1C,0x04,0xC4, 0xE4,0x3C,0x18,0x00, 0x00,0x00,0x00,0x0D, 0x0D,0x00,0x00,0x00,     //-?-
0xF0,0xF0,0x08,0xC8, 0xC8,0xF8,0xF0,0x00, 0x07,0x0F,0x08,0x0B, 0x0B,0x0B,0x01,0x00,     //-@-
0xE0,0xF0,0x98,0x8C, 0x98,0xF0,0xE0,0x00, 0x0F,0x0F,0x00,0x00, 0x00,0x0F,0x0F,0x00,     //-A-            ASCII CODE: 0X41
0x04,0xFC,0xFC,0x44, 0x44,0xFC,0xB8,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0F,0x07,0x00,     //-B-
0xF0,0xF8,0x0C,0x04, 0x04,0x0C,0x18,0x00, 0x03,0x07,0x0C,0x08, 0x08,0x0C,0x06,0x00,     //-C-
0x04,0xFC,0xFC,0x04, 0x0C,0xF8,0xF0,0x00, 0x08,0x0F,0x0F,0x08, 0x0C,0x07,0x03,0x00,     //-D-
0x04,0xFC,0xFC,0x44, 0xE4,0x0C,0x1C,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0C,0x0E,0x00,     //-E-
0x04,0xFC,0xFC,0x44, 0xE4,0x0C,0x1C,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,     //-F-
0xF0,0xF8,0x0C,0x84, 0x84,0x8C,0x98,0x00, 0x03,0x07,0x0C,0x08, 0x08,0x07,0x0F,0x00,     //-G-

0xFC,0xFC,0x40,0x40, 0x40,0xFC,0xFC,0x00, 0x0F,0x0F,0x00,0x00, 0x00,0x0F,0x0F,0x00,     //-H-            ASCII CODE: 0X48
0x00,0x00,0x04,0xFC, 0xFC,0x04,0x00,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,     //-I-
0x00,0x00,0x00,0x04, 0xFC,0xFC,0x04,0x00, 0x07,0x0F,0x08,0x08, 0x0F,0x07,0x00,0x00,     //-J-
0x04,0xFC,0xFC,0xC0, 0xE0,0x3C,0x1C,0x00, 0x08,0x0F,0x0F,0x00, 0x01,0x0F,0x0E,0x00,     //-K-
0x04,0xFC,0xFC,0x04, 0x00,0x00,0x00,0x00, 0x08,0x0F,0x0F,0x08, 0x08,0x0C,0x0E,0x00,     //-L-
0xFC,0xFC,0x38,0x70, 0x38,0xFC,0xFC,0x00, 0x0F,0x0F,0x00,0x00, 0x00,0x0F,0x0F,0x00,     //-M-
0xFC,0xFC,0x38,0x70, 0xE0,0xFC,0xFC,0x00, 0x0F,0x0F,0x00,0x00, 0x00,0x0F,0x0F,0x00,     //-N-
0xF8,0xFC,0x04,0x04, 0x04,0xFC,0xF8,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-O-
0x04,0xFC,0xFC,0x44, 0x44,0x7C,0x38,0x00, 0x08,0x0F,0x0F,0x08, 0x00,0x00,0x00,0x00,     //-P-
0xF8,0xFC,0x04,0x04, 0x04,0xFC,0xF8,0x00, 0x07,0x0F,0x08,0x0E, 0x3C,0x3F,0x27,0x00,     //-Q-

0x04,0xFC,0xFC,0x44, 0xC4,0xFC,0x38,0x00, 0x08,0x0F,0x0F,0x00, 0x00,0x0F,0x0F,0x00,     //-R-
0x18,0x3C,0x64,0x44, 0xC4,0x9C,0x18,0x00, 0x06,0x0E,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-S-
0x00,0x1C,0x0C,0xFC, 0xFC,0x0C,0x1C,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,     //-T-
0xFC,0xFC,0x00,0x00, 0x00,0xFC,0xFC,0x00, 0x07,0x0F,0x08,0x08, 0x08,0x0F,0x07,0x00,     //-U-
0xFC,0xFC,0x00,0x00, 0x00,0xFC,0xFC,0x00, 0x01,0x03,0x06,0x0C, 0x06,0x03,0x01,0x00,     //-V-
0xFC,0xFC,0x00,0x00, 0x00,0xFC,0xFC,0x00, 0x07,0x0F,0x0E,0x03, 0x0E,0x0F,0x07,0x00,     //-W-
0x0C,0x3C,0xF0,0xE0, 0xF0,0x3C,0x0C,0x00, 0x0C,0x0F,0x03,0x01, 0x03,0x0F,0x0C,0x00,     //-X-
0x00,0x0C,0x7C,0xC0, 0xC0,0x7C,0x3C,0x00, 0x00,0x00,0x08,0x0F, 0x0F,0x08,0x00,0x00,     //-Y-
0x1C,0x0C,0x84,0xC4, 0x64,0x3C,0x1C,0x00, 0x0E,0x0F,0x09,0x08, 0x08,0x0C,0x0E,0x00,     //-Z-
0x00,0x00,0xFC,0xFC, 0x04,0x04,0x00,0x00, 0x00,0x00,0x0F,0x0F, 0x08,0x08,0x00,0x00,     //-[-

0x38,0x70,0xE0,0xC0, 0x80,0x00,0x00,0x00, 0x00,0x00,0x00,0x01, 0x03,0x07,0x0E,0x00,     //-\-
0x00,0x00,0x04,0x04, 0xFC,0xFC,0x00,0x00, 0x00,0x00,0x08,0x08, 0x0F,0x0F,0x00,0x00,     //-]-
```

```
0x08,0x0C,0x06,0x03,  0x06,0x0C,0x08,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  //-^-
0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x20,0x20,0x20,0x20,  0x20,0x20,0x20,0x20,  //-_-
0x00,0x00,0x03,0x07,  0x04,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  //-`-
0x00,0xA0,0xA0,0xA0,  0xE0,0xC0,0x00,0x00,  0x07,0x0F,0x08,0x08,  0x07,0x0F,0x08,0x00,  //-a-            ASCII CODE: 0X61
0x04,0xFC,0xFC,0x20,  0x60,0xC0,0x80,0x00,  0x00,0x0F,0x0F,0x08,  0x08,0x0F,0x07,0x00,  //-b-
0xC0,0xE0,0x20,0x20,  0x20,0x60,0x40,0x00,  0x07,0x0F,0x08,0x08,  0x08,0x0C,0x04,0x00,  //-c-
0x80,0xC0,0x60,0x24,  0xFC,0xFC,0x00,0x00,  0x07,0x0F,0x08,0x08,  0x07,0x0F,0x08,0x00,  //-d-
0xC0,0xE0,0xA0,0xA0,  0xA0,0xE0,0xC0,0x00,  0x07,0x0F,0x08,0x08,  0x08,0x0C,0x04,0x00,  //-e-

0x40,0xF8,0xFC,0x44,  0x0C,0x18,0x00,0x00,  0x08,0x0F,0x0F,0x08,  0x00,0x00,0x00,0x00,  //-f-
0xC0,0xE0,0x20,0x20,  0xC0,0xE0,0x20,0x00,  0x27,0x6F,0x48,0x48,  0x7F,0x3F,0x00,0x00,  //-g-
0x04,0xFC,0xFC,0x40,  0x20,0xE0,0xC0,0x00,  0x08,0x0F,0x0F,0x00,  0x00,0x0F,0x0F,0x00,  //-h-
0x00,0x00,0x20,0xEC,  0xEC,0x00,0x00,0x00,  0x00,0x00,0x08,0x0F,  0x0F,0x08,0x00,0x00,  //-i-
0x00,0x00,0x00,0x00,  0x20,0xEC,0xEC,0x00,  0x00,0x30,0x70,0x40,  0x40,0x7F,0x3F,0x00,  //-j-
0x04,0xFC,0xFC,0x80,  0xC0,0x60,0x20,0x00,  0x08,0x0F,0x0F,0x01,  0x03,0x0E,0x0C,0x00,  //-k-
0x00,0x00,0x04,0xFC,  0xFC,0x00,0x00,0x00,  0x00,0x00,0x08,0x0F,  0x0F,0x08,0x00,0x00,  //-l-
0xE0,0xE0,0x60,0xC0,  0x60,0xE0,0xC0,0x00,  0x0F,0x0F,0x00,0x07,  0x00,0x0F,0x0F,0x00,  //-m-
0x20,0xE0,0xC0,0x20,  0x20,0xE0,0xC0,0x00,  0x00,0x0F,0x0F,0x00,  0x00,0x0F,0x0F,0x00,  //-n-
0xC0,0xE0,0x20,0x20,  0x20,0xE0,0xC0,0x00,  0x07,0x0F,0x08,0x08,  0x08,0x0F,0x07,0x00,  //-o-

0x20,0xE0,0xC0,0x20,  0x20,0xE0,0xC0,0x00,  0x40,0x7F,0x7F,0x48,  0x08,0x0F,0x07,0x00,  //-p-
0xC0,0xE0,0x20,0x20,  0xC0,0xE0,0x20,0x00,  0x07,0x0F,0x08,0x48,  0x7F,0x7F,0x40,0x00,  //-q-
0x20,0xE0,0xC0,0x60,  0x20,0xE0,0xC0,0x00,  0x08,0x0F,0x0F,0x08,  0x00,0x00,0x00,0x00,  //-r-
0x40,0xE0,0xA0,0x20,  0x20,0x60,0x40,0x00,  0x04,0x0C,0x09,0x09,  0x0B,0x0E,0x04,0x00,  //-s-
0x20,0x20,0xF8,0xFC,  0x20,0x20,0x00,0x00,  0x00,0x00,0x07,0x0F,  0x08,0x0C,0x04,0x00,  //-t-
0xE0,0xE0,0x00,0x00,  0xE0,0xE0,0x00,0x00,  0x07,0x0F,0x08,0x08,  0x07,0x0F,0x08,0x00,  //-u-
0x00,0xE0,0xE0,0x00,  0x00,0xE0,0xE0,0x00,  0x00,0x03,0x07,0x0C,  0x0C,0x07,0x03,0x00,  //-v-
0xE0,0xE0,0x00,0x80,  0x00,0xE0,0xE0,0x00,  0x07,0x0F,0x0C,0x07,  0x0C,0x0F,0x07,0x00,  //-w-
0x20,0x60,0xC0,0x80,  0xC0,0x60,0x20,0x00,  0x08,0x0C,0x07,0x03,  0x07,0x0C,0x08,0x00,  //-x-
0xE0,0xE0,0x00,0x00,  0x00,0xE0,0xE0,0x00,  0x47,0x4F,0x48,0x48,  0x68,0x3F,0x1F,0x00,  //-y-

0x60,0x60,0x20,0xA0,  0xE0,0x60,0x20,0x00,  0x0C,0x0E,0x0B,0x09,  0x08,0x0C,0x0C,0x00,  //-z-             //
0x00,0x40,0x40,0xF8,  0xBC,0x04,0x04,0x00,  0x00,0x00,0x00,0x07,  0x0F,0x08,0x08,0x00,  //-{-
0x00,0x00,0x00,0xBC,  0xBC,0x00,0x00,0x00,  0x00,0x00,0x00,0x0F,  0x0F,0x00,0x00,0x00,  //-|-
0x00,0x04,0x04,0xBC,  0xF8,0x40,0x40,0x00,  0x00,0x08,0x08,0x0F,  0x07,0x00,0x00,0x00,  //-}-
0x08,0x0C,0x04,0x0C,  0x08,0x0C,0x04,0x00,  0x00,0x00,0x00,0x00,  0x00,0x00,0x00,0x00,  //-~-           ASCII CODE: 0X7E


};



uchar code ascii_table_5x8[95][5]={
//ASCII CODE:5x8 DOT MATRIX
0x00,0x00,0x00,0x00,0x00,//space
0x00,0x00,0x4f,0x00,0x00,//!
0x00,0x07,0x00,0x07,0x00,//"
0x14,0x7f,0x14,0x7f,0x14,//#
0x24,0x2a,0x7f,0x2a,0x12,//$
0x23,0x13,0x08,0x64,0x62,//%
0x36,0x49,0x55,0x22,0x50,//&
0x00,0x05,0x07,0x00,0x00,//]
0x00,0x1c,0x22,0x41,0x00,//(
0x00,0x41,0x22,0x1c,0x00,//)
0x14,0x08,0x3e,0x08,0x14,//*
0x08,0x08,0x3e,0x08,0x08,//+
0x00,0x50,0x30,0x00,0x00,//,
0x08,0x08,0x08,0x08,0x08,//-
0x00,0x60,0x60,0x00,0x00,//.
0x20,0x10,0x08,0x04,0x02,///
0x3e,0x51,0x49,0x45,0x3e,//0
0x00,0x42,0x7f,0x40,0x00,//1
0x42,0x61,0x51,0x49,0x46,//2
0x21,0x41,0x45,0x4b,0x31,//3
0x18,0x14,0x12,0x7f,0x10,//4
0x27,0x45,0x45,0x45,0x39,//5
0x3c,0x4a,0x49,0x49,0x30,//6
0x01,0x71,0x09,0x05,0x03,//7
0x36,0x49,0x49,0x49,0x36,//8
0x06,0x49,0x49,0x29,0x1e,//9
0x00,0x36,0x36,0x00,0x00,//:
```

```
0x00,0x56,0x36,0x00,0x00,//;
0x08,0x14,0x22,0x41,0x00,//<
0x14,0x14,0x14,0x14,0x14,//=
0x00,0x41,0x22,0x14,0x08,//>
0x02,0x01,0x51,0x09,0x06,//?
0x32,0x49,0x79,0x41,0x3e,//@
0x7e,0x11,0x11,0x11,0x7e,//A
0x7f,0x49,0x49,0x49,0x36,//B
0x3e,0x41,0x41,0x41,0x22,//C
0x7f,0x41,0x41,0x22,0x1c,//D
0x7f,0x49,0x49,0x49,0x41,//E
0x7f,0x09,0x09,0x09,0x01,//F
0x3e,0x41,0x49,0x49,0x7a,//G
0x7f,0x08,0x08,0x08,0x7f,//H
0x00,0x41,0x7f,0x41,0x00,//I
0x20,0x40,0x41,0x3f,0x01,//J
0x7f,0x08,0x14,0x22,0x41,//K
0x7f,0x40,0x40,0x40,0x40,//L
0x7f,0x02,0x0c,0x02,0x7f,//M
0x7f,0x04,0x08,0x10,0x7f,//N
0x3e,0x41,0x41,0x41,0x3e,//O
0x7f,0x09,0x09,0x09,0x06,//P
0x3e,0x41,0x51,0x21,0x5e,//Q
0x7f,0x09,0x19,0x29,0x46,//R
0x46,0x49,0x49,0x49,0x31,//S
0x01,0x01,0x7f,0x01,0x01,//T
0x3f,0x40,0x40,0x40,0x3f,//U
0x1f,0x20,0x40,0x20,0x1f,//V
0x3f,0x40,0x38,0x40,0x3f,//W
0x63,0x14,0x08,0x14,0x63,//X
0x07,0x08,0x70,0x08,0x07,//Y
0x61,0x51,0x49,0x45,0x43,//Z
0x00,0x7f,0x41,0x41,0x00,//[
0x02,0x04,0x08,0x10,0x20,/* \ */
0x00,0x41,0x41,0x7f,0x00,//]
0x04,0x02,0x01,0x02,0x04,//^
0x40,0x40,0x40,0x40,0x40,//_
0x01,0x02,0x04,0x00,0x00,//`
0x20,0x54,0x54,0x54,0x78,//a
0x7f,0x48,0x48,0x48,0x30,//b
0x38,0x44,0x44,0x44,0x44,//c
0x30,0x48,0x48,0x48,0x7f,//d
0x38,0x54,0x54,0x54,0x58,//e
0x00,0x08,0x7e,0x09,0x02,//f
0x48,0x54,0x54,0x54,0x3c,//g
0x7f,0x08,0x08,0x08,0x70,//h
0x00,0x00,0x7a,0x00,0x00,//i
0x20,0x40,0x40,0x3d,0x00,//j
0x7f,0x20,0x28,0x44,0x00,//k
0x00,0x41,0x7f,0x40,0x00,//l
0x7c,0x04,0x38,0x04,0x7c,//m
0x7c,0x08,0x04,0x04,0x78,//n
0x38,0x44,0x44,0x44,0x38,//o
0x7c,0x14,0x14,0x14,0x08,//p
0x08,0x14,0x14,0x14,0x7c,//q
0x7c,0x08,0x04,0x04,0x08,//r
0x48,0x54,0x54,0x54,0x24,//s
0x04,0x04,0x3f,0x44,0x24,//t
0x3c,0x40,0x40,0x40,0x3c,//u
0x1c,0x20,0x40,0x20,0x1c,//v
0x3c,0x40,0x30,0x40,0x3c,//w
0x44,0x28,0x10,0x28,0x44,//x
0x04,0x48,0x30,0x08,0x04,//y
0x44,0x64,0x54,0x4c,0x44,//z
0x08,0x36,0x41,0x41,0x00,//{
0x00,0x00,0x77,0x00,0x00,//|
0x00,0x41,0x41,0x36,0x08,//}
0x04,0x02,0x02,0x02,0x01,//~
};

uchar code BMP12864_PANDA[]={
```

```
//PANDA_128x64.bmp
//128X64DOT MARTRIX
0xFF,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x81,0xC1,0xE1,0xF1,0xF9,0xF9,0xF9,0xFD,0xFD,0xFD,0xFD,0xF9,0xF9,0xF1,
0xF1,0xE1,0xC1,0xC1,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0x41,0xC1,0x81,0x81,
0x81,0xC1,0xE1,0xF1,0xF9,0xF9,0xFD,0xFD,0xFD,0xFD,0xFD,0xFD,0xF9,0xF9,0xF1,0xE1,
0xC1,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0xC1,0xE1,0xE1,0xF1,0xF1,0xF9,0xF9,0xF9,0xF9,0xF9,0xF9,0xF1,0xF1,
0xE1,0xC1,0x81,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,
0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0x01,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0xFE,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0x1F,0x0F,0x07,0x07,0x03,0x01,
0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x01,0x01,0x03,0x07,0x07,0x0F,0x1F,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0xFF,0x80,0xF0,0xF0,0x60,0x20,0x10,0x10,0x08,0x08,0x0C,0x04,0x04,0x04,0x04,
0x06,0x06,0x07,0x07,0x07,0x07,0x07,0x07,0x0F,0x0F,0x0F,0x1F,0x3F,0x3F,0x7F,0xFF,
0xFF,0xFF,0xFF,0x7E,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x03,0x07,0xFF,0x3F,0x07,0xC1,0xE0,0xF0,0xF8,0xF8,0xF8,0xFC,0xFC,0xFC,
0xFC,0xF8,0xF8,0xF8,0xF0,0xE0,0xC0,0x00,0x00,0x80,0xC0,0xE0,0xF0,0xF0,0xF8,0xF8,
0xF8,0xF8,0xF8,0xF8,0xF8,0xF0,0xF0,0xE0,0xC0,0x83,0x0F,0x7F,0xDF,0xEF,0xFF,0xF7,
0x19,0x04,0x03,0x01,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0xE0,0xF0,0xF8,0xFC,0xFC,0xFC,0xFE,0xFE,0xFE,0xFE,0xFC,0xFC,0xF8,0xF8,
0xF1,0xC3,0x0F,0x38,0xE0,0x80,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x7F,0xD8,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x8F,0x07,0x63,
0x23,0x27,0x07,0xFF,0xFF,0xFF,0xFF,0x7F,0x7E,0xFF,0xFF,0xFF,0xFF,0x8F,0x47,0x23,
0x23,0x23,0x47,0x8F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x7C,0xC0,0x7F,0x11,0xFF,0x00,
0xC0,0xF0,0xF8,0xFC,0xFE,0xFE,0xFF,0x7F,0x3F,0x3F,0x3F,0x3F,0x3E,0xFE,0xFC,0xF8,
0xF0,0xFF,0xFF,0xFF,0xFF,0xFF,0xF3,0xC1,0x80,0x9C,0x88,0x80,0xC1,0xFF,0xFF,0xFF,
0xFF,0x7F,0x0F,0x80,0x80,0x7F,0x40,0x40,0x40,0xC0,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x03,0x6E,0x99,0x93,0x97,0x9F,0x9F,0xBF,0x7F,0x1F,0x1E,
0x0E,0x0E,0x0F,0x07,0x07,0x13,0xB9,0x78,0x38,0xB9,0x03,0x07,0x0F,0x0F,0x1F,0x1E,
0x1E,0x1E,0xFE,0xFF,0x3F,0x3F,0x2F,0x27,0x33,0xF9,0xCE,0x03,0x00,0x00,0x0F,0x78,
0xCF,0x3F,0xFF,0xFF,0xFF,0xFF,0xFF,0xF0,0xF0,0xE1,0xE3,0xF0,0xF0,0xFC,0xFF,0xFF,
0x7F,0x1F,0x00,0x70,0xE1,0xF3,0x77,0x07,0x07,0x07,0x07,0x07,0x07,0x07,0x03,0x01,
0x00,0x06,0xC5,0xF4,0x9C,0xC4,0xC2,0xE2,0xE1,0xE0,0xE0,0xE0,0xE0,0xE0,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0xC0,0xF0,0xF8,0xF8,0xFC,0xFD,0xFD,0xFF,0xFE,0xFC,0xFC,
0xE8,0xC8,0x18,0x10,0x30,0xF0,0xF0,0xF1,0xF1,0xF0,0x30,0x10,0x10,0x18,0x08,0x08,
0x0C,0x04,0x0E,0x1F,0x3F,0x7F,0xFF,0xFF,0xFF,0xFE,0xFE,0xFC,0xFC,0xFC,0xF8,0xF8,
0xE1,0x1F,0xFC,0xF9,0xF1,0xE3,0xA3,0x93,0xCB,0xCF,0xBF,0x83,0x03,0x01,0x01,0x00,
0x00,0x00,0x00,0x00,0x00,0x80,0x80,0x82,0xC0,0xC0,0xC0,0xE0,0xF0,0xF0,0xC8,0x0C,
0x1E,0x3F,0x3F,0x7F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0x7F,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x3F,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,
0xFF,0x7F,0x3F,0x70,0xFC,0xFF,0xFF,0xFF,0xFF,0xFF,0xFF,0xFC,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x01,0x03,0x0F,0x7F,0xFF,0x3F,0x3F,0x1F,0x1F,
0x67,0xF8,0xFF,0xFF,0xFF,0xFF,0xFF,0x3F,0x1F,0x07,0x01,0x01,0x01,0x01,0x0F,0x1F,
0x3F,0x3F,0x3F,0x3F,0x1F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x0F,0x07,0x00,
0x00,0x00,0x00,0x00,0x00,0x03,0x07,0x1F,0xFF,0x87,0x03,0x01,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x00,0xFF,
0xFF,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0xBF,0x81,0x80,0x81,0x81,0x83,0x83,0x83,0x81,0x81,
0x80,0x80,0x80,0x80,0x81,0x83,0x87,0x87,0x87,0x87,0x83,0x81,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xBF,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x81,0xB1,0xBF,0x81,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xBF,0x80,0x80,0x80,0x80,0x80,0x80,
0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0x80,0xFF,
};
```