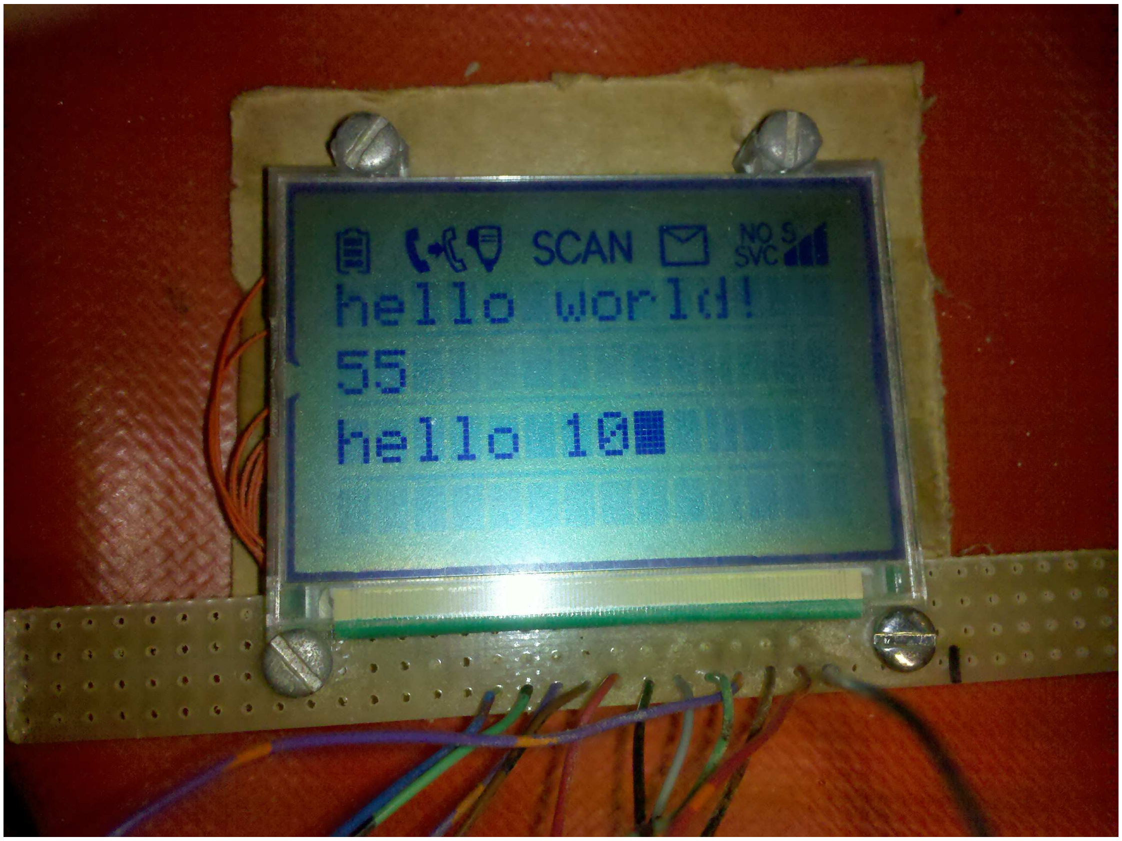
HUGP002A LCD Datasheet



The HUGP002A is a 14 character by 4 line LCD. It is generally only available as surplus and was sold by earthlcd.com under the name CLCD414. It is believed to use the nju6426 controller although this has not been verified. It responds to the same command set as the common Hitachi HD44780 controller chip, but has a set of icons at the top that are controlled in a different manner.

Pinout

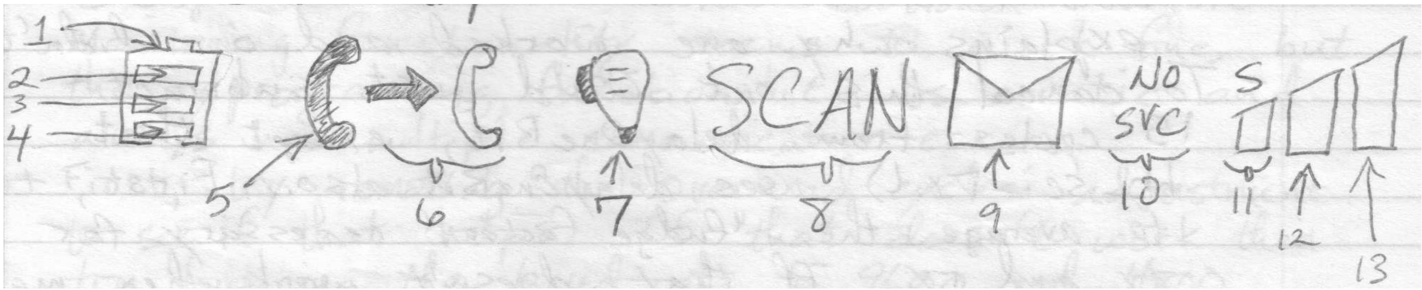
Pin	Description
1	GND
2	VCC (5V)
3	Batt (LED power)
4	LED Control (0V = ON)
5	GND
6	RS
7	R/W
8	Enable
9	DB0
10	DB1
11	DB2
12	DB3

Row Address Offsets

The row address offsets also differ from the standard row address offsets of typical 4 line LCDs.

Row	Offset
1	0x00
2	0x10
3	0x40
4	0x50

Icon Information



The icons on the HUGP002A are accessed by setting bits in the CGRAM region. Note that 0x30 and 0x02 commands must be written before writing to the CGRAM locations and following the CGRAM writes a 0x29 command must be written. Writing a "1" in the enable bit position turns on that icon, and writing a "1" in the blink bit position makes that icon blink. See the example code for more information.

CGRAM Icon Locations

Icon	CGRAM Address	Enable Bit Number	Blink Bit Number
1	0x01	0	7
2	0x00	1	7
3	0x01	1	7
4	0x00	0	7
5	0x02	0	7
6	0x03	0	7
7	0x04	0	7
8	0x05	0	7
9	0x08	0	7
10	0x09	0	7
11	0x0B	0	7
12	0x0D	1	7
13	0x0D	0	7

Sample Code

This code is meant to be run on an Arduino. The electrical connections are described in the comments. Note that this example uses the standard LCDCrystal library included with Arduino for common HD44780-compatible commands.

```
/*
*
*       Test program for interfacing the HUGP002A character LCD to the Arduino
*
```

```
*    Author: Ken Sharp
*    Date: 8/27/2011
*
*     The circuit:
*     LCD RS pin to digital pin 12
*     LCD Enable pin to digital pin 11
*     LCD D4 pin to digital pin 5
*     LCD D5 pin to digital pin 4
*     LCD D6 pin to digital pin 3
*     LCD D7 pin to digital pin 2
*     LCD R/W pin to ground
*    LCD backlight control to pin 8
*/

#include <LiquidCrystal.h>

//icon code defines
#define LCD_ICON_BATTERY_OUTLINE 0x11
#define LCD_ICON_BATTERY_SEG1 0x20
#define LCD_ICON_BATTERY_SEG2 0x21
#define LCD_ICON_BATTERY_SEG3 0x10
#define LCD_ICON_BATTERY_OUTLINE_AND_SEG2 0x31
#define LCD_ICON_BATTERY_SEG1_AND_SEG3 0x30
#define LCD_ICON_BLACK_PHONE 0x12
#define LCD_ICON_WHITE_PHONE 0x13
#define LCD_ICON_PEN 0x14
#define LCD_ICON_SCAN 0x15
#define LCD_ICON_ENVELOPE 0x18
#define LCD_ICON_NO_SVC 0x19
#define LCD_ICON_BAR1 0x1B
#define LCD_ICON_BAR2 0x2D
#define LCD_ICON_BAR3 0x1D
#define LCD_ICON_BAR2_AND_BAR3 0x3D

char row_offsets[4] = {0x00, 0x10, 0x40, 0x50};
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {
    lcd.begin(14,4);
    lcd.print("hello world!");
    hugp002a_iconsOff();

    // Turn on backlight
    digitalWrite(8, 0);
}

void loop() {
    static int i = 0;
    delay(5000);

    hugp002a_iconControl(LCD_ICON_BATTERY_OUTLINE_AND_SEG2, i%2, 1);
    hugp002a_iconControl(LCD_ICON_BATTERY_SEG1_AND_SEG3, i%2, 1);
    hugp002a_iconControl(LCD_ICON_BLACK_PHONE, i%2, 1);
    hugp002a_iconControl(LCD_ICON_WHITE_PHONE, i%2, 1);
    hugp002a_iconControl(LCD_ICON_PEN, i%2, 1);
    hugp002a_iconControl(LCD_ICON_SCAN, i%2, 1);
    hugp002a_iconControl(LCD_ICON_ENVELOPE, i%2, 1);
    hugp002a_iconControl(LCD_ICON_NO_SVC, i%2, 1);
    hugp002a_iconControl(LCD_ICON_BAR1, i%2, 1);
    hugp002a_iconControl(LCD_ICON_BAR2_AND_BAR3, i%2, 1);

    hugp002a_setCursor(0,1);
    lcd.print(millis()/1000);
    hugp002a_setCursor(0,2);
    lcd.print("hello ");
    lcd.print(i);
    i++;
}

void hugp002a_setCursor(char col, char row) {
    lcd.command(LCD_SETDDRAMADDR | col + row_offsets[row]);
}

/* must call at least one print() prior to calling this */
void hugp002a_iconsOff() {
    int i;
    lcd.command(0x30);
    lcd.command(0x02);
    for (i=0x40; i <= 0x4d; i++) {
        lcd.command(i);
        lcd.write(0x00);
    }
    lcd.command(0x29);
}

void hugp002a_iconControl(unsigned char icon, char blink, char state) {
    unsigned char icon_byte, icon_addr;

    icon_addr = (0x0f & icon) + 0x40;

    if (state > 0) {
        icon_byte = (icon & 0xf0) >> 4;
        if (blink > 0)
            icon_byte |= 0x80;
    }
    else {
        icon_byte = 0;
    }

    lcd.command(0x30);
    lcd.command(0x02);

    lcd.command(icon_addr);
    lcd.write(icon_byte);

    lcd.command(0x29);
}

// This function doesn't seem to work
// I found this on the internet so I've kept
// it here for documentation and future investigation
void hugp002a_iconsBlink(char state) {
    lcd.command(0x30);
    lcd.command(0x02);
    lcd.command((state) ? 0x2e : 0x2c);
    lcd.command(0x29);
```

```
            }
```