INTELLIGENT
DISPLAY SOLUTIONS

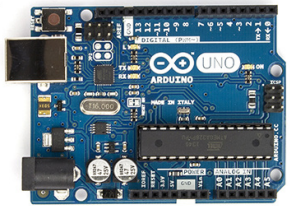
CI064-4001 LCD – Arduino application note for 8bit and 4bit control

Overview

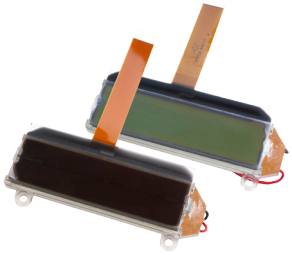
This guide will show you how to connect the CI064-4001-xx series of COG displays to your Arduino. By adding an LCD to your projects this will allow you to display messages and also debug issues easier. The guide will show you both 4Bit control as well as 8Bit control.

What you need

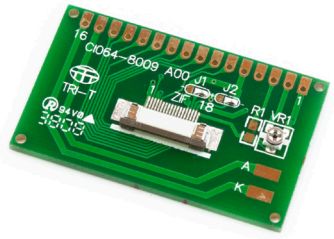
- Arduino Uno



- CI064-4001 16x2 LCD

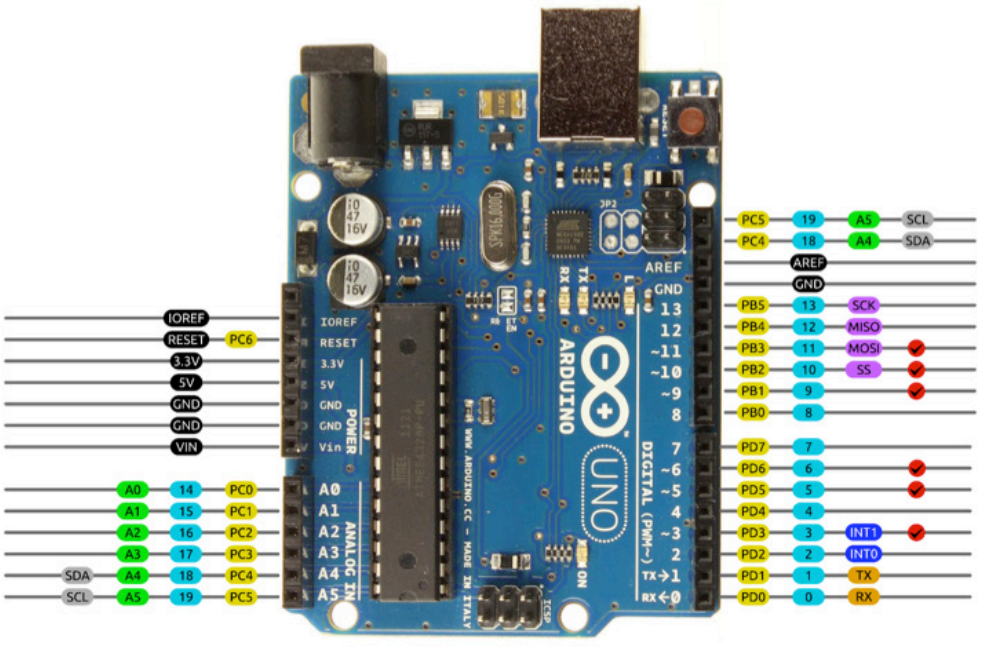


- IDB-CI064-4001-xx-01



- Jumper Wires

Arduino Pinout



AVR DIGITAL ANALOG POWER SERIAL SPI I2C PWM INTERRUPT

Intelligent Display Solutions, Unit 2, Berkshire Business Centre, Berkshire Drive, Thatcham, Berkshire, RG19 4EW
Telephone : +44 (0)1635 294600 Fax : +44 (0)1635 869200 Email: info@i-lcd.com www.i-lcd.com
A division of Intelligent Group Solutions Ltd

QUICK START

Wiring the LCD to the Arduino

The following wiring connections need to be made to allow the LCD to work with the Arduino in 8Bit mode.

LCD pin number	Arduino pin number	Comment
1	GND	Vss
2	+5V	Vdd
3	N/C	VD
4	10	Register Select
5	N/C	R/W
6	8	Enable
7	0	DB0
8	1	DB1
9	2	DB2
10	3	DB3
11	4	DB4
12	5	DB5
13	6	DB6
14	7	DB7
15	11	BACKLIGHT+
16	GND	BACKLIGHT-

Note: If the Arduino is only going to be configured for 4bit control then DB0-DB3 do not need to be connected.

There are 8 data lines called DB0-DB7. These pins will either send high or low to the LCD.

The enable pin toggle to allow for data to be wrote to the LCD.

The register select pin can either be to send commands to the LCD such as clear the display or to move to a new line. To send this data the pin must be low. When the pin is high the pin will go into data send mode and allow for data to be send to the display.

The BACKLIGHT+ connection must be connected with a 89R resistor in series. This is to limit the current going to the backlight.

Intelligent Display Solutions, Unit 2, Berkshire Business Centre, Berkshire Drive, Thatcham, Berkshire, RG19 4EW
Telephone : +44 (0)1635 294600 Fax : +44 (0)1635 869200 Email: info@i-lcd.com www.i-lcd.com
A division of Intelligent Group Solutions Ltd

Software Testing

The guide will set up the LCD using 8Bit control first then move onto 4Bit control after. For the LCD to operate with the Arduino a library must be used. The Arduino software when installed will also install several useful libraries which help get peripherals working with your Arduino to make your projects better. To use the LCD display the 'liquidcrystal' library will be used. A library will add extra functions and features to your sketch.

To import the library sketch > include library > liquidcrystal. To ensure that the library has been imported correctly at the top of the sketch if should read '#include <LiquidCrystal.h>'.

With the library imported the setup of each pin can now be defined. Each display control lines needs to be assigned to the correct Arduino pin in the software. This is done using the predefined function 'LiquidCrystal LCD (10, 8, 0, 1, 2, 3, 4, 5, 6, 7);'. With the first two pin numbers presenting register select and enable then the rest represent the data lines. Now the Arduino has imported the correct library and also setup the pins for the LCD. This set is shown in figure 1.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(10, 8, 0, 1, 2, 3, 4, 5, 6, 7);
```

Figure 1. Initialisation

The brightness of the backlight can be control through a pin on the Arduino. This needs to be defined in the sketch after the setup of the library and the pins. To setup in the pin type 'int backlight = 11;' This will assign the backlight control from the LCD to pin 11 of the Arduino in the software.

```
  int backlight = 11;
```

Figure 2. Backlight Initialisation

With the initialisation completed, it's now time to setup the LCD. As there are many different sizes of displays available such as 16x2 or 20x4, we need to define the size of our display. This is show in figure 3. The function 'lcd.begin(16,2)', defines the number of columns and the rows respectively.

Also during set up we will send the LCD a message to display. This messaged is displayed using the lcd.print function followed by your message. The last stage before the main loop is to set up the backlight pin as either an input or an output. As we would like to adjust the brightness of the backlight, this requires the pin to be an output. To define the pin as an output type the following into the sketch 'pinMode(backlight, OUTPUT)'.

```
int backlight = 11;

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Test! 8 Bit");

  pinMode(backlight, OUTPUT);   // sets the pin as output
}
```

Figure 3. Setup

Intelligent Display Solutions, Unit 2, Berkshire Business Centre, Berkshire Drive, Thatcham, Berkshire, RG19 4EW
Telephone : +44 (0)1635 294600 Fax : +44 (0)1635 869200 Email: info@i-lcd.com www.i-lcd.com
A division of Intelligent Group Solutions Ltd

Now that the setup of the display is completed the main loop which will continually run can be created. The setup will only run once then enter into the main loop and not leave that loop until the reset button has been pushed or the Arduino has been powered down. Within the main loop we will set a timer to count how long in seconds since the last reset. To change the positon of the cursor we use the function setCursor. As we want the cursor to be in position zero, we first send a 0, followed by a 1 as we would like the cursor on the next line of the display.

```
void loop() {
  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins with 0):
  lcd.setCursor(0, 1);
  // print the number of seconds since reset:
  lcd.print(millis() / 1000);

 analogWrite(backlight, 255);
}
```

Figure 4. Main Loop

In figure 4, it shows how to display the time since the last reset, with the predefined function milis(). The final stage of the code is change the brightness of the backlight. As we would like the voltage of the backlight pin to change the function analogwrite is used. Then the voltage of the pin is determined upon the value in the function. The highest the value can be being 255 and the lowest is 0. The higher the value the brighter the backlight of the LCD. As this is in the main loop, this will continue to run until reset.

4Bit Control

To enable your LCD to operate with 4Bit control, the following changes need to be made. The number of pins initialised in the first few lines of code will change. Instead of defining 10 pins after the function liquidcrystal lcd(), we will only define 6 pins. These will be register select, enable and then the data control lines. This is shown in figure 5.

```
// include the library code:
#include <LiquidCrystal.h>

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(10, 8, 4, 5, 6, 7);
```

Figure 5. 4Bit Mode

This is all that is needed to change the display from 8Bit mode into 4Bit mode. The sketch will detect that only 4 data lines have been set and then change accordingly. Then repeat the steps to run the code.