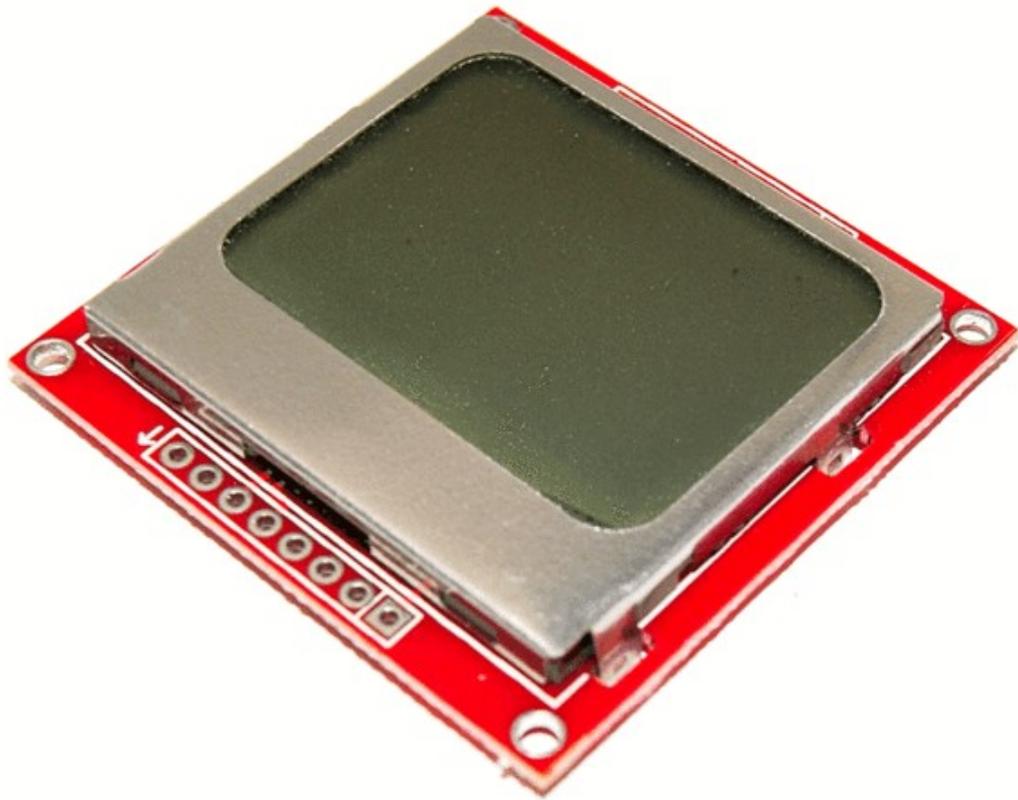


AZ-Delivery

Welcome!

Thank you very much for purchasing our AZ-Delivery Nokia 5110 LCD SPI display. On the following pages, we will introduce you to how to use and setup this handy device.

Have fun!



Az-Delivery

The Nokia 5110 Display is a monochrome LCD display with 84x48 pixels and LED backlighting. As the display was originally used mainly for mobile phones, the working voltage of the display is 3.3V. This means that the supply voltage and the data line voltage use 3.3V.

Voltages higher than 3.3 volts can destroy the display!

In order to be able to work with the display with our Arduino, it is important to bring the data lines of the Arduinos, which work with 5V, to display compatible 3.3V. This voltage conversion can be done by level shifters.

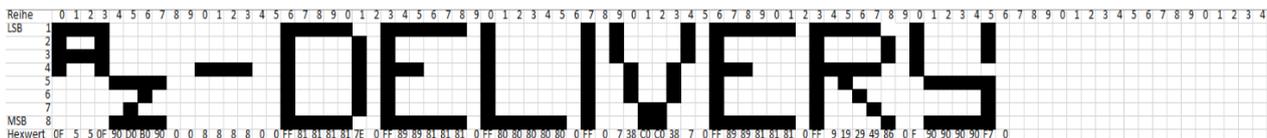
The display has a total of 8 connections, which are assigned as follows:

Pin:	Description:	Additional information:
1	RST (Reset) Input	
2	CE (Chip Enable) Input	
3	DC (Data/Control) Input	
4	DIN (Data in) Input	
5	CLK (Clock) Input	
6	VCC	Power supply - max. 3.3V!
7	LIGHT	LED active when connected to GND
8	GND (Ground)	Ground

Az-Delivery

Making test sketch

Now we come to the actual programming of the display. We want to display the following bit pattern to our screen:



We keep in mind that the display has a resolution of 84 column pixels x 48 row pixels. The line pixels in the internal RAM of the display are organized in bytes, so that 8 line pixels are written with the first output of the first data byte. This is followed by a jump to the next column, in which 8 bits are output as bytes, always starting with the LSB bit first. After reaching column 84, a change to the next 8-bit line takes place. Thus we reach a total of $48/8 = 6$ byte lines with 84 bytes each. The complete screen space in the RAM thus amounts to 6×84 bytes = 504 bytes. After reaching the limit, the following characters are displayed again at the beginning of the display. Writing the byte 0 to all places of the display results a displayed text on the screen.

Az-Delivery



To get our logo into the display, we calculate the corresponding hex value or decimal value from each 8 bit of column 0 to 65. The resulting array of byte values are our data, which we will send to the display later. In the example, this calculation has already been carried out in the line "Hex value". These values serve us in the sketch as display data, which are stored in the array "LOGOTBL" can be deposited. The Arduino function "shiftout" takes over the actual data transmission and outputs the data serially via a clock signal generated in the function itself.

The data is taken over by the display when the clock signal changes from *LOW* to *HIGH*. This can be seen in the following oscilloscope picture:



Pin Clock (CLK) has the color cyan, the data pin the color yellow. Pin Chip Enable (CE) has the color magenta. You can see the transmission of 2 bytes after CE goes to LOW.



The Arduino Code

```
// Control Nokia 5110 LCD Display
#define CLK 8
#define DIN 9
#define DC 10
#define CE 11
#define RST 12
static const byte LOGOTBL[] = { //AZ Delivery Bitmap Logo
    0x0F,0x5,0x5,0x0F,0x90,0xD0,0xB0,0x90,0x00,0x00,0x08,0x08,0x08,0x08,0x00,
    0x00,0xFF,0x81,0x81,0x81,0x81,0x7E,0x00,
    0xFF,0x89,0x89,0x81,0x81,0x81,0x00,0xFF,0x80,0x80,0x80,0x80,0x80,0x00,
    0xFF,0x00,0x07,0x38,0xC0,0xC0,0x38,0x07,
    0x00,0xFF,0x89,0x89,0x81,0x81,0x81,0x00,0xFF,0x09,0x19,0x29,0x49,0x86,
    0x00,0xF,0x90,0x90,0x90,0x90,0xF7
};
// Functions
void LcdWriteCmd(byte order) { // Send command to display
    digitalWrite(DC, LOW); // DC Pin is LOW for commands to the display
    shiftOut(DIN, CLK, MSBFIRST, order); // Transmit data byte serially
    // with MSB (most significant byte) first
}
void LcdWriteData(byte data) { // Send data to display
    digitalWrite(DC, HIGH); // DC Pin is HIGH for data to the display
    shiftOut(DIN, CLK, MSBFIRST, data); // Transmit data byte serial
    // with MSB (most significant byte) first
}
```

AZ-Delivery

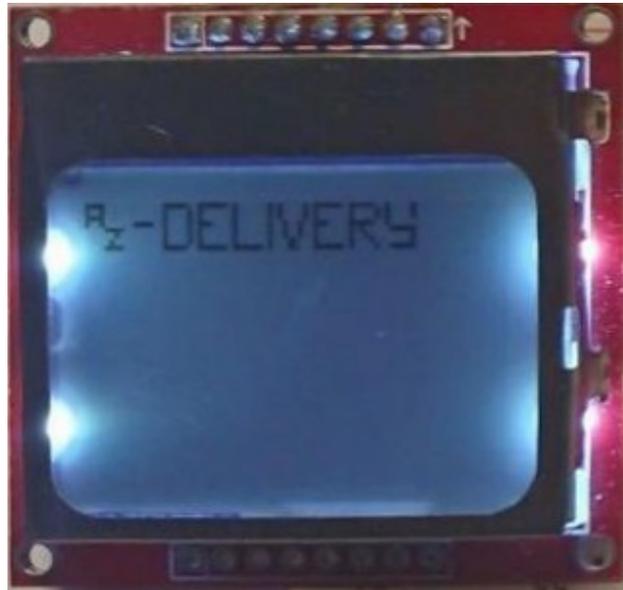
```
void LcdWriteLogo() {
    digitalWrite(CE, LOW); // Chip Enable Pin is LOW,
                          // data transfer is activated
    LcdWriteCmd(0x80); // Jump to row 0 column 0, Jump Column
    LcdWriteCmd(0x40); // Skip row
    // Clear screen completely
    for(int i=0; i < 504; i++) {
        LcdWriteData(0x00); // Delete complete display
    }
    // Output AZ logo
    for(int i=0; i < 66; i++) {
        LcdWriteData(LOGOTBL[i]); // Write logo data into display
    }
    digitalWrite(CE, HIGH); // Chip Enable Pin is HIGH,
                          // data transfer is deactivated
}

void setup() {
    pinMode(CLK, OUTPUT); // Switch pin to output
    pinMode(DIN, OUTPUT); // Switch pin to output
    pinMode(DC, OUTPUT); // Switch pin to output
    pinMode(CE, OUTPUT); // Switch pin to output
    pinMode(RST, OUTPUT); // Switch pin to output
    digitalWrite(RST, HIGH); // Reset signal to Nokia display
    delay(100);
    // forward
    digitalWrite(RST, LOW);
    delay(100);
    digitalWrite(RST, HIGH);
    delay(100);
    // Initialization of the display according to data sheet
    digitalWrite(CE, LOW); // Chip Enable Pin is LOW,
                          // data transfer is activated
    LcdWriteCmd(0x21); // Extended command set LCD
    LcdWriteCmd(0x90); // set LCD Vop
    LcdWriteCmd(0x20); // Normal command LCD
    LcdWriteCmd(0x0C); // LCD Normal Mode
    digitalWrite(CE, HIGH); // Chip Enable Pin is HIGH,
                          // data transfer is deactivated
}

void loop() {
    LcdWriteLogo();
    delay(20000);
}
```

Az-Delivery

After wiring the display and uploading the sketch, we receive the following output:



Further information about the display is available on the Internet in the form of data sheets. Just google to "*PCD8544 data sheet*".

You've done it, you can now use and program your module for your projects.

AZ-Delivery

Now it is time to learn and make the Projects on your own. You can do that with the help of many example scripts and other tutorials, which you can find on the internet.

If you are looking for the high quality products for Arduino and Raspberry Pi, AZ-Delivery Vertriebs GmbH is the right company to get them from. You will be provided with numerous application examples, full installation guides, eBooks, libraries and assistance from our technical experts.

<https://az-delivery.de>

Have Fun!

Impressum

<https://az-delivery.de/pages/about-us>