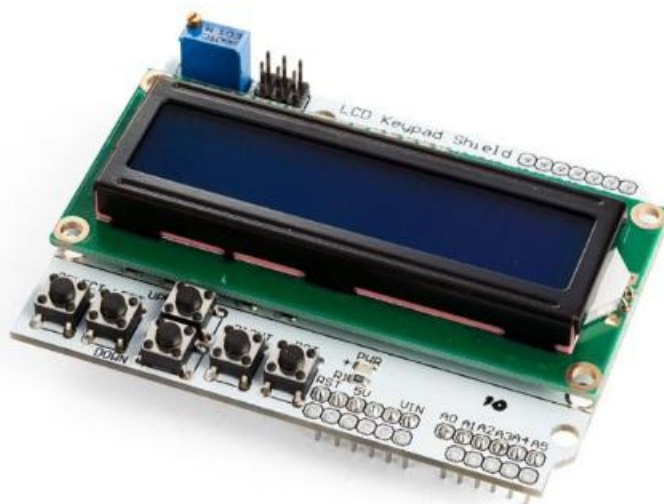


MAKEVMA203

LCD & KEYPAD SHIELD FOR ARDUINO® - LCD1602



USER MANUAL



USER MANUAL

1. Introduction

To all residents of the European Union

Important environmental information about this product



This symbol on the device or the package indicates that disposal of the device after its lifecycle could harm the environment. Do not dispose of the unit (or batteries) as unsorted municipal waste; it should be taken to a specialized company for recycling. This device should be returned to your distributor or to a local recycling service. Respect the local environmental rules.

■ If in doubt, contact your local waste disposal authorities.

Please read the manual thoroughly before bringing this device into service. If the device was damaged in transit, do not install or use it and contact your dealer.

2. Safety Instructions



- This device can be used by children aged from 8 years and above, and persons with reduced physical, sensory or mental capabilities or lack of experience and knowledge if they have been given supervision or instruction concerning the use of the device in a safe way and understand the hazards involved. Children shall not play with the device. Cleaning and user maintenance shall not be made by children without supervision.



- Indoor use only.
Keep away from rain, moisture, splashing and dripping liquids.

3. General Guidelines



- Familiarise yourself with the functions of the device before actually using it.
- All modifications of the device are forbidden for safety reasons. Damage caused by user modifications to the device is not covered by the warranty.
- Only use the device for its intended purpose. Using the device in an unauthorised way will void the warranty.
- Damage caused by disregard of certain guidelines in this manual is not covered by the warranty and the dealer will not accept responsibility for any ensuing defects or problems.
- The dealers cannot be held responsible for any damage (extraordinary, incidental or indirect) – of any nature (financial, physical...) arising from the possession, use or failure of this product.
- Due to constant product improvements, the actual product appearance might differ from the shown images.
- Product images are for illustrative purposes only.
- Do not switch the device on immediately after it has been exposed to changes in temperature. Protect the device against damage by leaving it switched off until it has reached room temperature.
- Keep this manual for future reference.

4. What is Arduino®

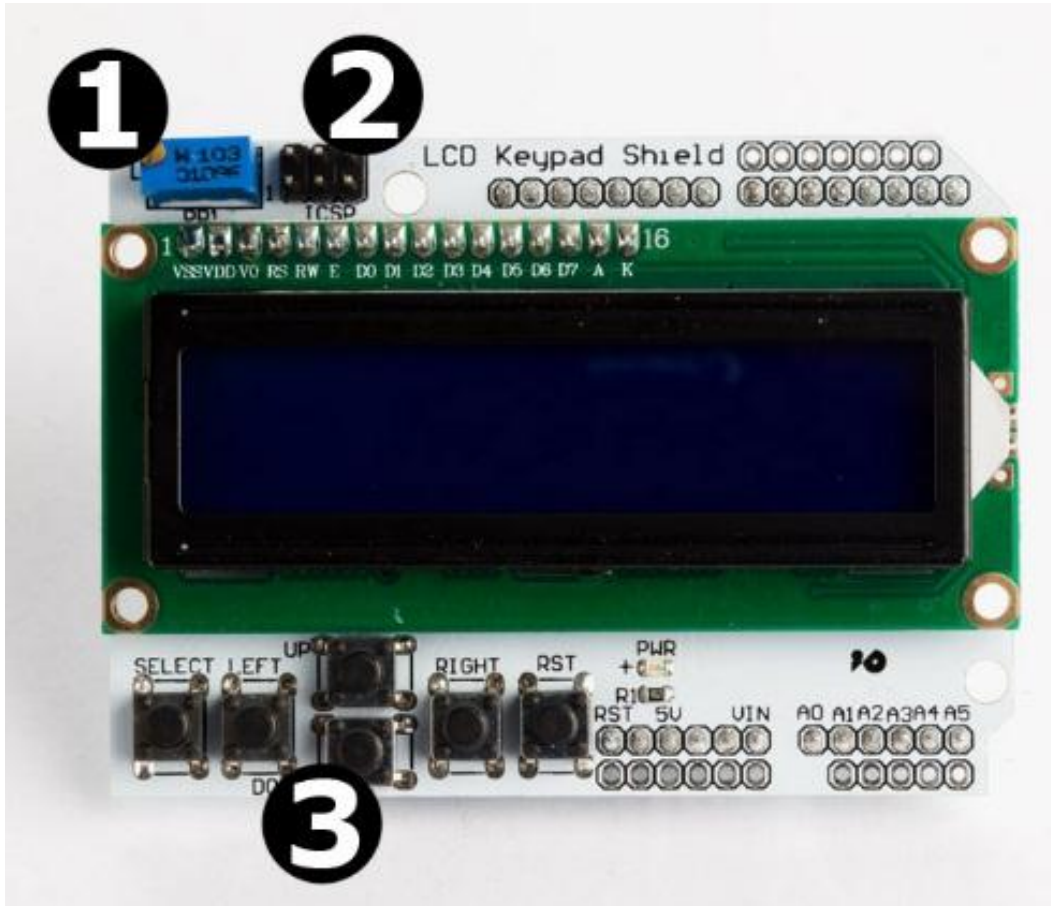
Arduino® is an open-source prototyping platform based in easy-to-use hardware and software. Arduino® boards are able to read inputs – light-on sensor, a finger on a button or a Twitter message – and turn it into an output – activating of a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so, you use the Arduino programming language (based on Wiring) and the Arduino® software IDE (based on Processing).

Surf to www.arduino.cc and www.arduino.org for more information.

5. Overview

MAKEVMA203

The 16x2 LCD and keypad shield for Arduino® Uno, Mega, Diecimila, Duemilanove and Freeduino boards.



1	LCD contrast potentiometer
2	ICSP

3	control keys (connect to analogue input 0)
----------	--

dimensions 80 x 58 x 20 mm

6. Pin Layout

Analogue 0	UP, DOWN, RIGHT, LEFT, SELECT
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS
Digital 9	E
Digital 10	Backlight

7. Example

```

*****code begin*****
//Sample using LiquidCrystal library
#include <LiquidCrystal.h>

/*****

This program will test the LCD panel and the buttons

*****/

// select the pins used on the LCD panel
LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

// define some values used by the panel and buttons
int lcd_key    = 0;
int adc_key_in = 0;
#define btnRIGHT  0
#define btnUP     1
#define btnDOWN   2
#define btnLEFT   3

#define btnSELECT 4
#define btnNONE   5

// read the buttons
int read_LCD_buttons()
{
  adc_key_in = analogRead(0); // read the value from the sensor
  // my buttons when read are centered at these values: 0, 144, 329, 504, 741
  // we add approx 50 to those values and check to see if we are close
  if (adc_key_in > 1000) return btnNONE; // We make this the 1st option for speed reasons since it
  will be the most likely result
  // For V1.1 us this threshold
  if (adc_key_in < 50)   return btnRIGHT;
  if (adc_key_in < 250) return btnUP;
  if (adc_key_in < 450) return btnDOWN;
  if (adc_key_in < 650) return btnLEFT;
  if (adc_key_in < 850) return btnSELECT;
}

```

```

// For V1.0 comment the other threshold and use the one below:
/*
if (adc_key_in < 50)   return btnRIGHT;
if (adc_key_in < 195) return btnUP;
if (adc_key_in < 380) return btnDOWN;
if (adc_key_in < 555) return btnLEFT;
if (adc_key_in < 790) return btnSELECT;
*/

return btnNONE; // when all others fail, return this...
}

void setup()
{
  lcd.begin(16, 2);           // start the library
  lcd.setCursor(0,0);
  lcd.print("Push the buttons"); // print a simple message
}

void loop()
{
  lcd.setCursor(9,1);        // move cursor to second line "1" and 9 spaces over
  lcd.print(millis()/1000);  // display seconds elapsed since power-up

  lcd.setCursor(0,1);        // move to the beginning of the second line
  lcd_key = read_LCD_buttons(); // read the buttons

  switch (lcd_key)           // depending on which button was pushed, we perform an
  action
  {
    case btnRIGHT:
      {
        lcd.print("RIGHT ");
        break;
      }
    case btnLEFT:
      {
        lcd.print("LEFT  ");
        break;
      }
    case btnUP:
      {
        lcd.print("UP   ");
        break;
      }
  }
}

```

```
case btnDOWN:
{
  lcd.print("DOWN ");
  break;
}
case btnSELECT:
{
  lcd.print("SELECT");
  break;
}
case btnNONE:
{
  lcd.print("NONE ");
  break;
}
}

}
*****Code End*****
```

© COPYRIGHT NOTICE

All worldwide rights reserved. No part of this manual may be copied, reproduced, translated or reduced to any electronic medium or otherwise without the prior written consent of the copyright holder.