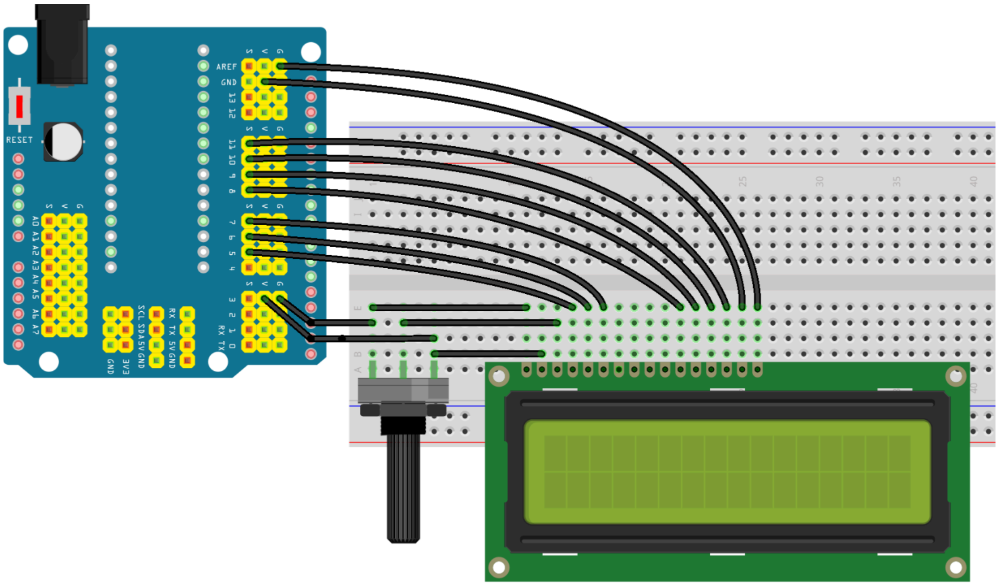
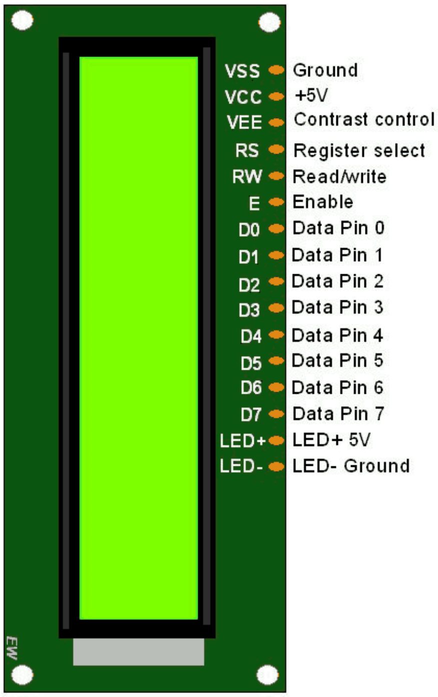
Task 1. Connect the circuit as shown in the picture.





Arduino board	LCD board
G	Vss
V	Vcc
var. resistor	Vee
5	RS
6	RW
7	E
8	D4
9	D5
10	D6
11	D7
V	A
G	K

`LiquidCrystal` library allows an Arduino board to control LCDs with the **HD44780** (or a compatible) controller. Reference:

www.arduino.cc/en/Reference/LiquidCrystal

Code example:

```
#include <LiquidCrystal.h> // library


#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

//LCD pins order: RS;R/W;E;D0-D7
LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                  D4_PIN, D5_PIN, D6_PIN, D7_PIN);

void setup() {
  pinMode(LED_BUILTIN,OUTPUT);
  lcd.begin(16,2);
  lcd.clear();
  lcd.print("Basic Eng.Course");
}

void loop() {
  led_blink(LED_BUILTIN,250);
}

void led_blink(int pin,int time) {
  digitalWrite(pin,HIGH);
  delay(time);
  digitalWrite(pin,LOW);
  delay(time);
}
```

Modify the *loop()* function and observe results. Are all the characters visible?

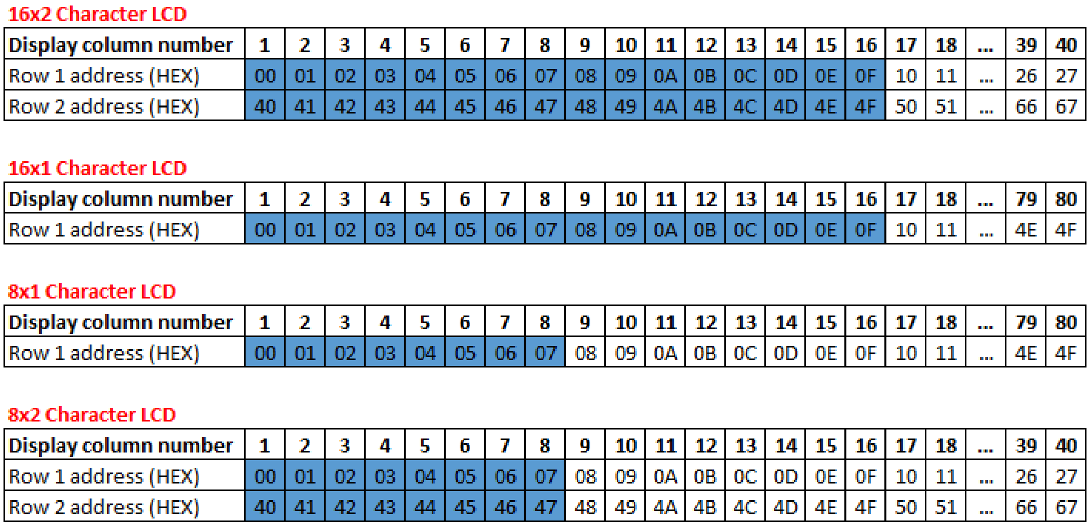
```
void loop() {
  led_blink(LED_BUILTIN,250);
  lcd.clear();
  lcd.print("Basic Engineering Course");
}
```

What is visible after uploading the following code?

```
void loop() {
  led_blink(LED_BUILTIN,250);
  lcd.clear();
  lcd.print("This is the Basic Engineering Course classes in 
2023");
}
```

Conclusions:

1. Always clear the places on the LCD You are not using - either with *lcd.clear* or filling them with *SPACE* character.
2. The text is not divided automatically between the lines.
3. A display memory map is presented below.



16x2 Character LCD

Display column number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	39	40
Row 1 address (HEX)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	26	27
Row 2 address (HEX)	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	...	66	67

16x1 Character LCD

Display column number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	79	80
Row 1 address (HEX)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	4E	4F

8x1 Character LCD

Display column number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	79	80
Row 1 address (HEX)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	4E	4F

8x2 Character LCD

Display column number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	...	39	40
Row 1 address (HEX)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	...	26	27
Row 2 address (HEX)	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	...	66	67

Task 2. Create a program for the Arduino board that allows You to display your name on the first line of an LCD and your surname on the second line.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

//LCD pins order: RS;R/W;E;D0-D7
LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                  D4_PIN, D5_PIN, D6_PIN, D7_PIN);


char myname[] = “Tomasz”;
char mysurname[] = “Ocetkiewicz”;

void setup() {
 lcd.begin(16,2);
 lcd.clear();
}

void loop() {
 lcd.setCursor(5,0);
 lcd.print(myname);
 lcd.setCursor(3,1);
 lcd.print(mysurname);
 while(1);
}
```

What’s new:

`lcd.print` method, `lcd.setCursor` method

Task 3. Implement the following code. Observe results.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                   D4_PIN, D5_PIN, D6_PIN, D7_PIN);

char lcd_line1_buffer[] = “Cursor”;

void setup() {
 lcd.begin(16,2);
 lcd.clear();
 lcd.home();
 lcd.print(lcd_line1_buffer);
}

void loop() {
 lcd.blink();        // turns on the blinking cursor
 delay(4000);
 lcd.noBlink(); // turns off the blinking cursor
 delay(4000);
 lcd.cursor();       // turns on the cursor
 delay(4000);
 lcd.noCursor();     // turns off the cursor
 delay(4000);

 lcd.cursor();
 lcd.setCursor(0,0);  // top left
 delay(4000);
 lcd.setCursor(15,0); // top right
 delay(4000);
 lcd.setCursor(0,1);   // bottom left
 delay(4000);
 lcd.setCursor(15,1); // bottom right
 delay(4000);
}
```

What’s new:

lcd.blink method, *lcd.noBlink* method

lcd.cursor method, *lcd.noCursor* method

Task 4. Implement the following code. Observe results.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                   D4_PIN, D5_PIN, D6_PIN, D7_PIN);

void setup() {
 lcd.begin(16,2); }

void loop() {
 lcd.clear();
 lcd.home();

 for (int pos=0; pos<10; pos++) {
  lcd.print(pos);
  delay(500); }

 lcd.setCursor(16, 1);
 lcd.autoscroll();
 for (int pos=0; pos<10; pos++) {
  lcd.print(pos);
  delay(500); }
 lcd.noAutoscroll();
 lcd.clear(); }
```

What’s new:

lcd.autoscroll method, *lcd.noAutoscroll* method,

Task 5. Implement the following code. Observe results. Prepare Your code to present *lcd.scrollDisplayRight* method.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                  D4_PIN, D5_PIN, D6_PIN, D7_PIN);

void setup() {
 lcd.begin(16,2);
 lcd.clear();
 lcd.home();
 lcd.print(“Basic Eng. Course”); }

void loop() {
 for(int pos=0; pos< 10; pos++) {
  lcd.scrollDisplayLeft();
  delay(500); } }
```

What’s new:

lcd.scrollDisplayLeft method, *lcd.scrollDisplayRight* method

Task 6. Rewrite the following code. Observe results.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                   D4_PIN, D5_PIN, D6_PIN, D7_PIN);

char digit = '0';

void setup() {
 lcd.begin(16,2);
 lcd.clear();
 lcd.home(); }

void loop() {
 if (digit == '3')
  lcd.rightToLeft();

 if (digit == '6')
  lcd.leftToRight();

 if (digit > '9') {
  lcd.home();
  digit = '0'; }

  lcd.write(digit);
  delay(1000);
  digit++;
}
```

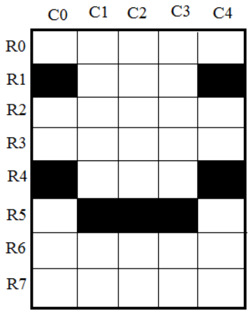
What’s new:

lcd.rightToLeft method, *lcd.leftToRight* method

lcd.write method

Task 7. Implement the following code. Observe results.

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                   D4_PIN, D5_PIN, D6_PIN, D7_PIN);

byte smiley[8] = {
  B00000,
  B10001,
  B00000,
  B00000,
  B10001,
  B01110,
  B00000,
};

void setup() {
  lcd.createChar(0,smiley);     // create a new char (no 0)
  lcd.begin(16,2);
  lcd.write(byte(0));}          // display custom char no 0

void loop() {
}
```

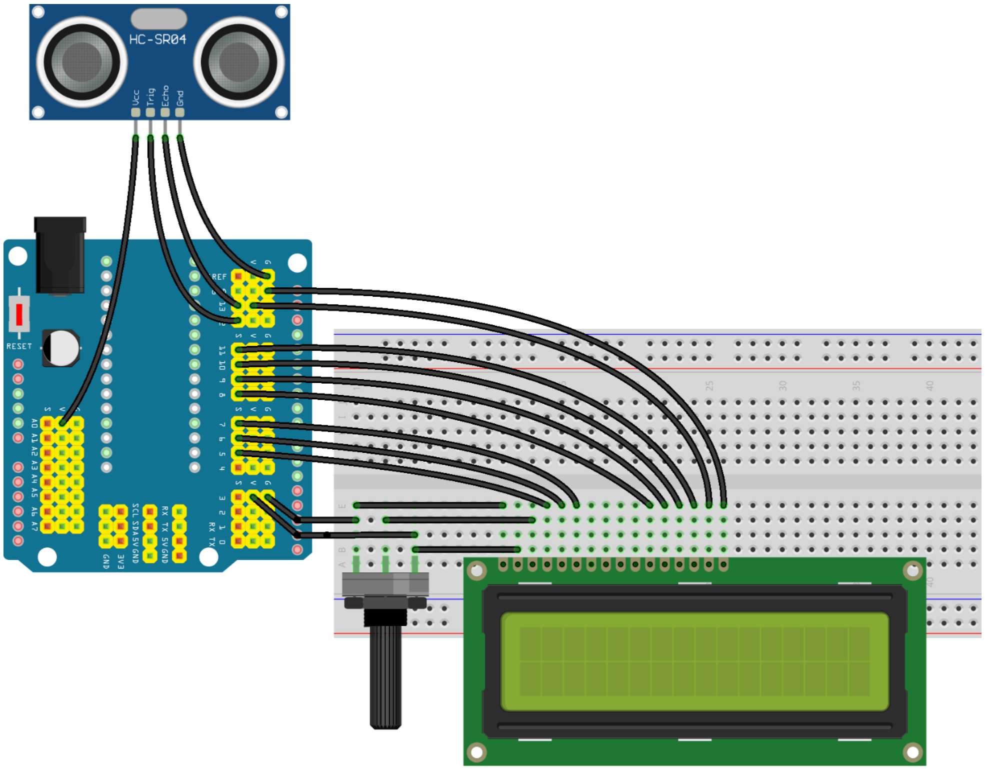


What's new:

lcd.createChar method

Up to eight characters of 5x8 pixels are supported. They are numbered from 0 to 7. The appearance of each custom character is specified by an array of 8 bytes, one for each row. The 5 LSBs of each byte determine the pixels in that row.

Task 8. Connect the circuit as shown in the picture.



Prepare code that implements range finder. Project parameters are as follows:

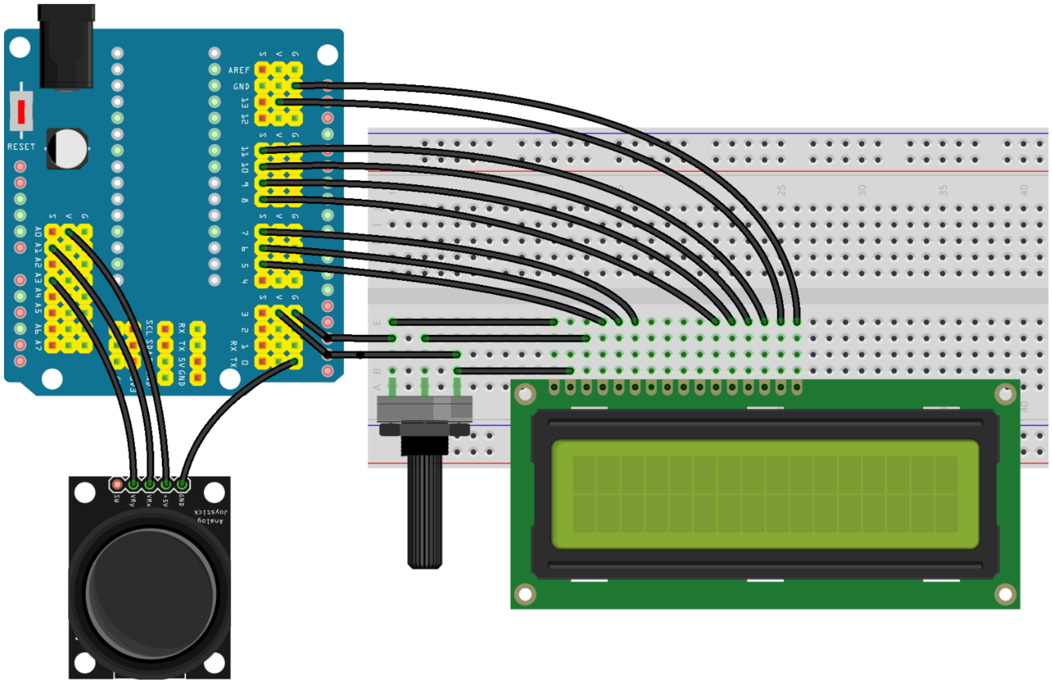
1. Distance measurement results should be displayed on the LCD.
2. The longest measured distance should be displayed on the LCD.
3. The shortest measured distance should be displayed on the LCD.
4. It should be possible to reset the longest and the shortest measured distances.

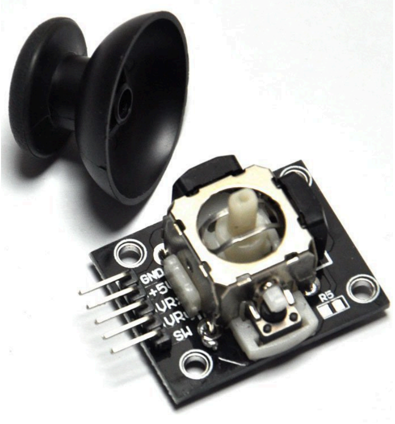
Hint:

```
String lcd_line1;
int number = 5;
lcd_line1 = "Result " + String(number);
lcd.print(lcd_line1);
```

```
#include <LiquidCrystal.h>
#define LCD_RS_PIN 5
#define LCD_RW_PIN 6
#define LCD_E_PIN 7
#define D4_PIN 8
#define D5_PIN 9
#define D6_PIN 10
#define D7_PIN 11

#define ECHO_PIN 13
#define TRIGGER_PIN 12

LiquidCrystal lcd(LCD_RS_PIN,LCD_RW_PIN,LCD_E_PIN,
                   D4_PIN, D5_PIN, D6_PIN, D7_PIN);

long distance = 0;
long max_distance = 0;
long p_millis = 0;

String lcd_line1,lcd_line2;

void setup() {
 pinMode(LED_BUILTIN,OUTPUT);
 pinMode(ECHO_PIN,INPUT);
 pinMode(TRIGGER_PIN,OUTPUT);

 lcd.begin(16,2);
 lcd.clear();
 lcd.print("* Range finder *");
 delay(2000);
 lcd.clear();
 lcd.print("Distance:");
}

#define M_PERIOD 500

void loop() {
 if(millis() - p_millis > M_PERIOD) {
  distance = find_range(ECHO_PIN, TRIGGER_PIN);
  lcd.setCursor(10,0);
  lcd.print("       ");
  lcd.setCursor(10,0);
  lcd.print(String(distance));
  lcd.print("cm");
  p_millis = millis();
 }
}

long find_range(int EchoPin, int TriggerPin) {
 long range=0;
 do {
  digitalWrite(TriggerPin,LOW);
  delayMicroseconds(2);
  digitalWrite(TriggerPin,HIGH);
  delayMicroseconds(10);
  digitalWrite(TriggerPin,LOW);
  range = pulseIn(EchoPin,HIGH)/58;
 } while(range>280);
 return range;
}
```

Task 9. Create a program, for an Arduino board, that fills the LCD screen with a character of Your choice (eg. ‘*’) proportionally to the X-axis joystick inclination.





Arduino board	Joystick board
G	Gnd
V	+5V
A3	VRx
A1	VRy
-	SW

Task 10. Using the circuit from Task no 2 create a program for the Arduino board that allows You to move the **custom** character (Task no 7) on the LCD screen. All places on the LCD should be accessible for the character.

Task 11. Create a prototype of a thermometer. Project parameters are as follows:

1. Temperature measurement results should be displayed on the LCD.
2. The highest measured temperature should be displayed on the LCD.
3. The lowest measured temperature should be displayed on the LCD.
4. It should be possible to reset the highest and the lowest results.

For those interested:

1. SparkFun - Basic Character LCD Hookup Guide:
   learn.sparkfun.com/tutorials/basic-character-lcd-hookup-guide/all

2. lastminuteengineers.com web page - Interfacing 16×2 Character LCD Module with Arduino:
   lastminuteengineers.com/arduino-1602-character-lcd-tutorial/

3. arduinogetstarted.com LCD tutorial:
   arduinogetstarted.com/reference/library/arduino-lcd-library