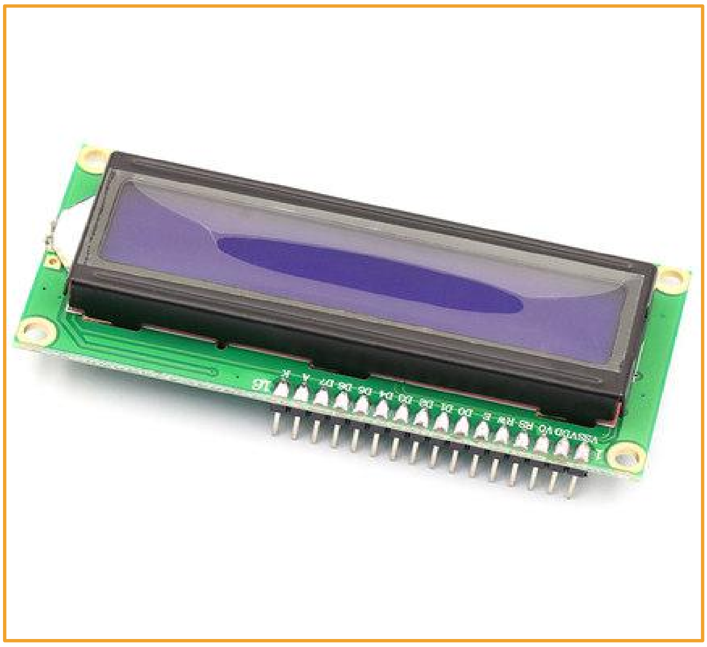
LCD1602 _Display screen experiment

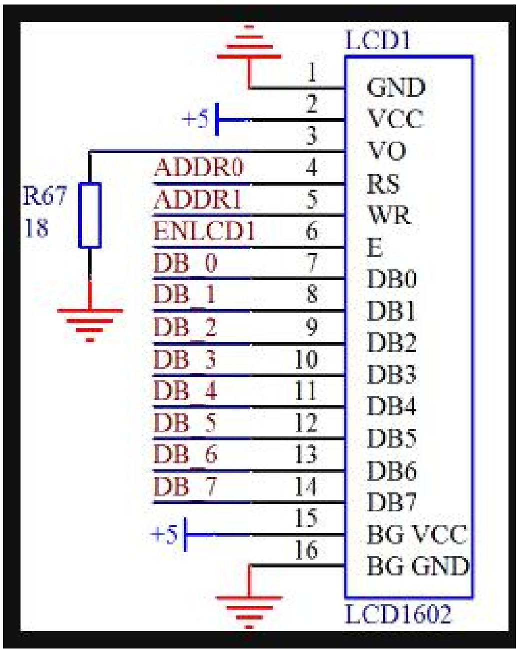
Introduction to LCD1602 _Display screen

LCD1602 is a kind of character LCD module specialized in displaying letters, numbers and symbols. It is widely used in industry, say, electronic clock, temperature display.

Character lcd on the market are mostly based on HD44780 character LCD chip, the control principle is identical. “1602” represents 2 rows and 16 characters of each row.

The physical map of LCD1602:





Function of Each Pin

LCD1602 has standard 14 pins (without backlight) or 16 pins(with backlight) interface, the specifications of each pin is shown in table 10-13.

Number	Pin Name	Pin description	Number	Pin Name	Pin description
1	VSS	GND	9	D2	Data 2
2	VDD	VDD	10	D3	Data 3
3	VL	Liquid Crystal bias	11	D4	Data 4
4	RS	data/command select	12	D5	Data 5
5	R/W	Read/write select	13	D6	Data 6
6	E	Enable signal	14	D7	Data 7
7	D0	Data 0	15	BLA	Backlight VDD
8	D1	Data 1	16	BLK	Backlight VSS

Pin 1: VSS is power supply.

Pin 2: VDD is for +5v.

Pin 3: VL is for LCD contrast adjustment. When VL is connected to the anode of power supply, the contrast is the weakest; when to GND, there will be the highest contrast.Excess contrast will lead to "shadow", so we require a 10k potentiometer to adjust the contrast when using it.

Pin 4: RS stands for register selection. When the pin is high level, we select data registers, otherwise, command registers are selected.

Pin 5: R/W stands for read/write. Read operations are for high level, write operations are for low level. When the RS and R/W are all low level, write operations and address display can be carried out; when the RS is low level and R/W is high, Busy signal is red, otherwise, data can be wrote in.

Pin 6: E is enable pin. When E jumps to low level from high level, the LCD module takes commands.

Pin 7 ~ 14: D0 ~ D7 is 8-bit bidirectional data line.

Pin 15: The anode of backlight.

Pin 16: The cathode of backlight.

Command 1: Clear Display

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Screen	0	0	0	0	0	0	0	0	0	1

Function: cmd code01H .clear all screen content and reset display print cursor to address 0x00H (Show back the top left of monitor) set address counter (AC) a value of 0

Command 2: Reset Cursor

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Reset cursor	0	0	0	0	0	0	0	0	1	x

Reset cursor, set address to 0x00H

Command 3: Mode Setting

Function	Command Code

	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Mode Setting	0	0	0	0	0	0	0	1	I/D	S

Function: Set data to 1 bit at a time after the shift direction of the cursor, and set the write one character at a time if mobile.

I/D : 0= write one bit data and left shift of the cursor 1= write one bit data and right shift of the cursor.

S : 0 = writing new data after the screen does not move 1 = writing new data display overall moves to the right after 1 characters.

Command 4: Display ON/OFF Control

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Display Control	0	0	0	0	0	0	1	D	C	B

Function: Display on/off control, The cursor flashing/closed, The cursor show or not

Control Bit	Setting	
D	0=disable display	1= enable display
C	0=cursor not show	1= show cursor
B	0=cursor flash	1= cursor not flash

Command 5: Set the display instruction or cursor movement direction

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Direction control	0	0	0	0	0	0	S/C	R/L	C	B

Function: Cursor or display screen shift control

Control bit		setting
S/C	R/L	
0	0	The cursor left shift 1, and AC value decrease 1
0	1	The cursor right shift 1, and AC value increase 1
1	0	All characters on a shift to the left one, but the cursor does not move
1	1	All characters on a shift to the right one, but the cursor does not move

Command 6: Function setting

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Function setting	0	0	0	0	1	DL	N	F	X	X

Function： Data bus digits, show the number of rows and fonts

Control bit	Setting	
DL	0= Data bus for 4	1= Data bus for 8
N	0= show one line	1= show two line
F	0= 5 x 7 lattice/per character	1= 5 x 10 lattice/per character

Command 7: CGRAM Address Setting

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
CGRAM Address	0	0	0	0	CGRAM Address（6bit）					

Function： Set the next to deposit the CGRAM address of data.

DB5DB4DB3 char code, the address of the characters(000~111)
DB2DB1DB0 line number(000~111)

Command 8: Set DDRAM address

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DDRAM Address	0	0	0	DDRAM address（7bit）						

Function： Set the next to deposit data of DDRAM addresses.

Command 9:Read the busy signal or AC address instruction

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0

AC/busing	0	0	FB	AC 内容（7bit）

Function： Read busy signal BF bit ,BF=1 means LCD1602 device is busy and can not receive data or command from MCU.

1．BF=0 LCD1602 device is not busy can receive data or command from MCU.

2．read AC register.

Command 10: write data to DDRAM or CGRAM

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
DD/CGRAM	0	0	Write data D0 - D7							

Function：

1．write char data to DDRAM;

2．write graph code to CGRAM. DB7DB6DB5 is ignore, default value “000” DB4DB3DB2DB1DB0 Font data that corresponds to each row 5

Command 11: read DDRAM or CGRAM Value

Function	Command Code									
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read DD/CGRAM	0	0	Read value D0 - D7							

Function ： read value from DDRAM or CGRAM.

Timing：

Read status input：RS=L，RW=H，E=H output：DB0～DB7

Read cmd input：RS=L，RW=L，E= Falling edge pulse，DB0～DB7

Read Data input：RS=H，RW=H，E=H output：DB0～DB7

Write Data input：RS=H，RW=L，E= Falling edge pulse ，DB0～DB7

Experiment Purpose

The first row of LCD1602 displays “hello Arduino !”,the second displays “keywish !”

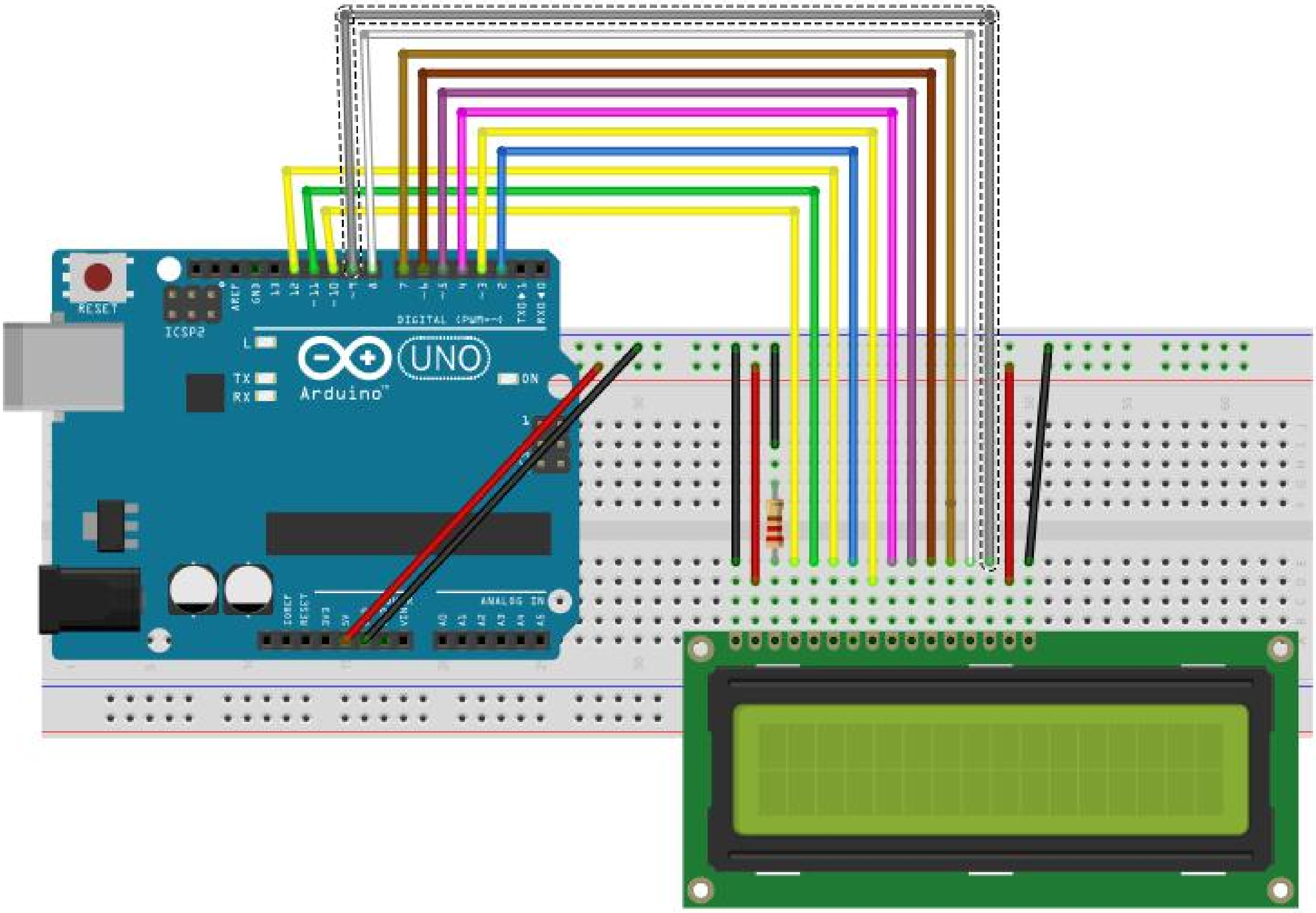
Components List

- Arduinos Uno board
- Breadboard
- USB cable
- LCD1602
- 10k range potentiometer
- Several jumper wires

Wiring

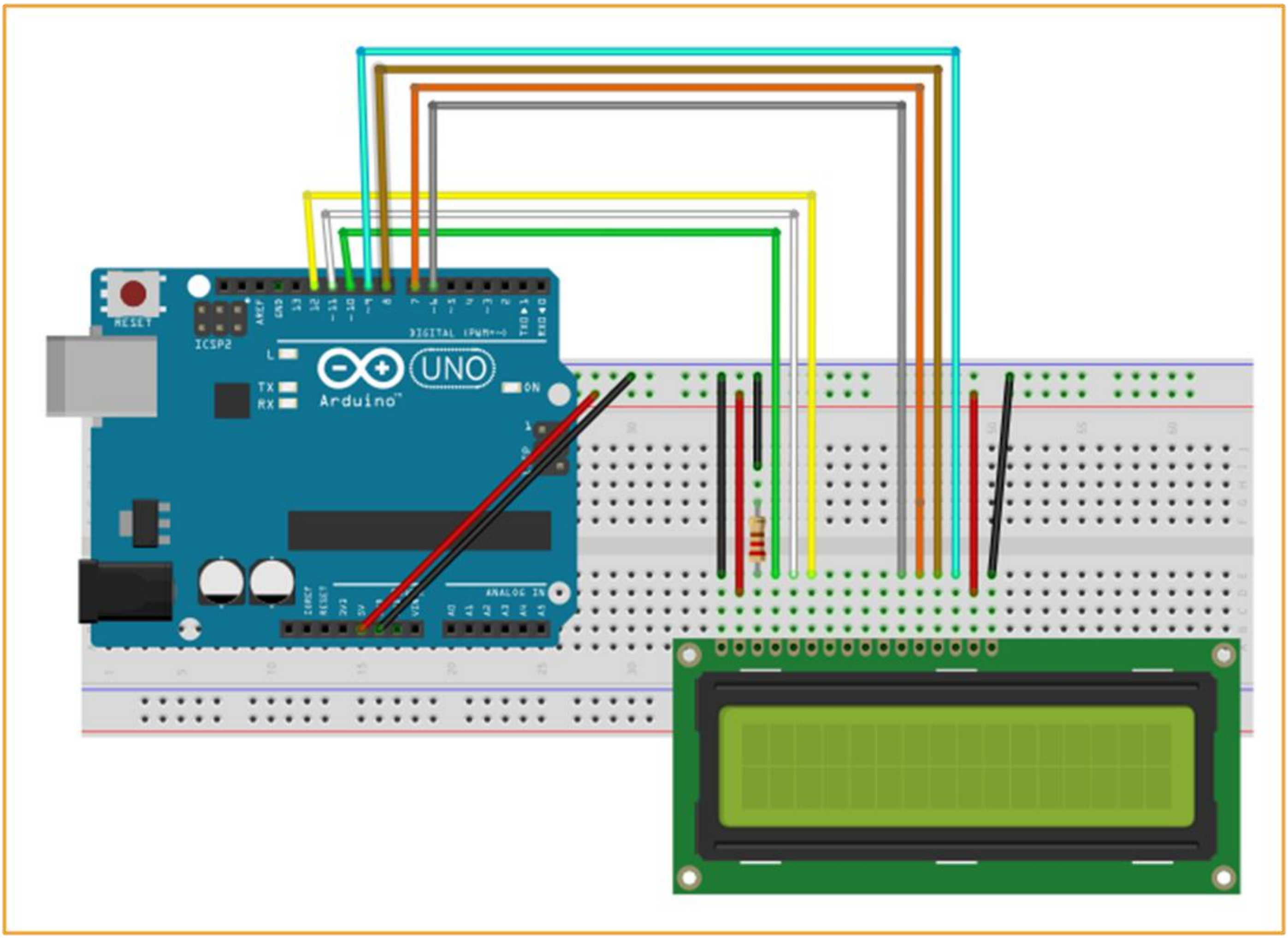
8-wire drive mode of LCD1602 based on Arduino:

Arduino	Lcd1602
GND	1 (VSS)
VCC	2 (VDD)
GND	3 (VO)
10	4 (RS)
11	5 (RW)
12	6 (E)
2	7 (D0)
3	8 (D1)
4	9 (D2)
5	10 (D3)
6	11 (D4)
7	12 (D5)
8	13 (D6)
9	14 (D7)
VCC	15 (V)
GND	16 (K)



4-wire drive mode of LCD1602 based on Arduino:

Arduino	Lcd1602
GND	1 (VSS)
VCC	2 (VDD)
GND	3 (VO)
10	4 (RS)
11	5 (RW)
6	11 (D4)
7	12 (D5)
8	13 (D6)
9	14 (D7)
VCC	15 (V)
GND	16 (K)



Experiment Principle

- First, lcd1602 takes commands above
- Writing lcd1602_write_cmd (0x38) by command 6 initializes 8-bit interface, 2 rows and 5x7 character size.
- Writing lcd1602_write_cmd x(0x06) by command 3; Setting the input as auto-increment, without the display of shifting.
- Writing lcd1602_write_cmd x(0x0E) by command 4; displaying the screen turned on, the cursor display, flicker-free.
- Writing lcd1602_write_cmd (0x01) by command 1; empty the screen, the cursor position return to zero.

Code

8-wire drive mode of LCD1602 based on Arduino:

```
#define   DB0   2   // lcd1602 DB0
#define   DB1   3   // lcd1602 DB1
#define   DB2   4   // lcd1602 DB2
#define   DB3   5   // lcd1602 DB3
#define   DB4   6   // lcd1602 DB4
#define   DB5   7   // lcd1602 DB5
#define   DB6   8   // lcd1602 DB6
#define   DB7   9    // lcd1602 DB7

#define   LCD1602_RS   10
#define   LCD1602_RW   11
#define   LCD1602_E    12// include the library code:
#include <LiquidCrystal.h>// initialize the library by associating any needed LCD interface
pin// with the arduino pin number it is connected to

LiquidCrystal lcd(LCD1602_RS,LCD1602_RW,LCD1602_E,DB0,DB1,DB2,DB3,DB4,DB5,DB6,DB7);

void setup() {// set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
}

void loop() {// set the cursor to (0,0):
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(" Hello Ardunio ! ");
    delay(10);
    lcd.setCursor(2,1);
    lcd.print(" keywish! ");
   while(1);

  }
```

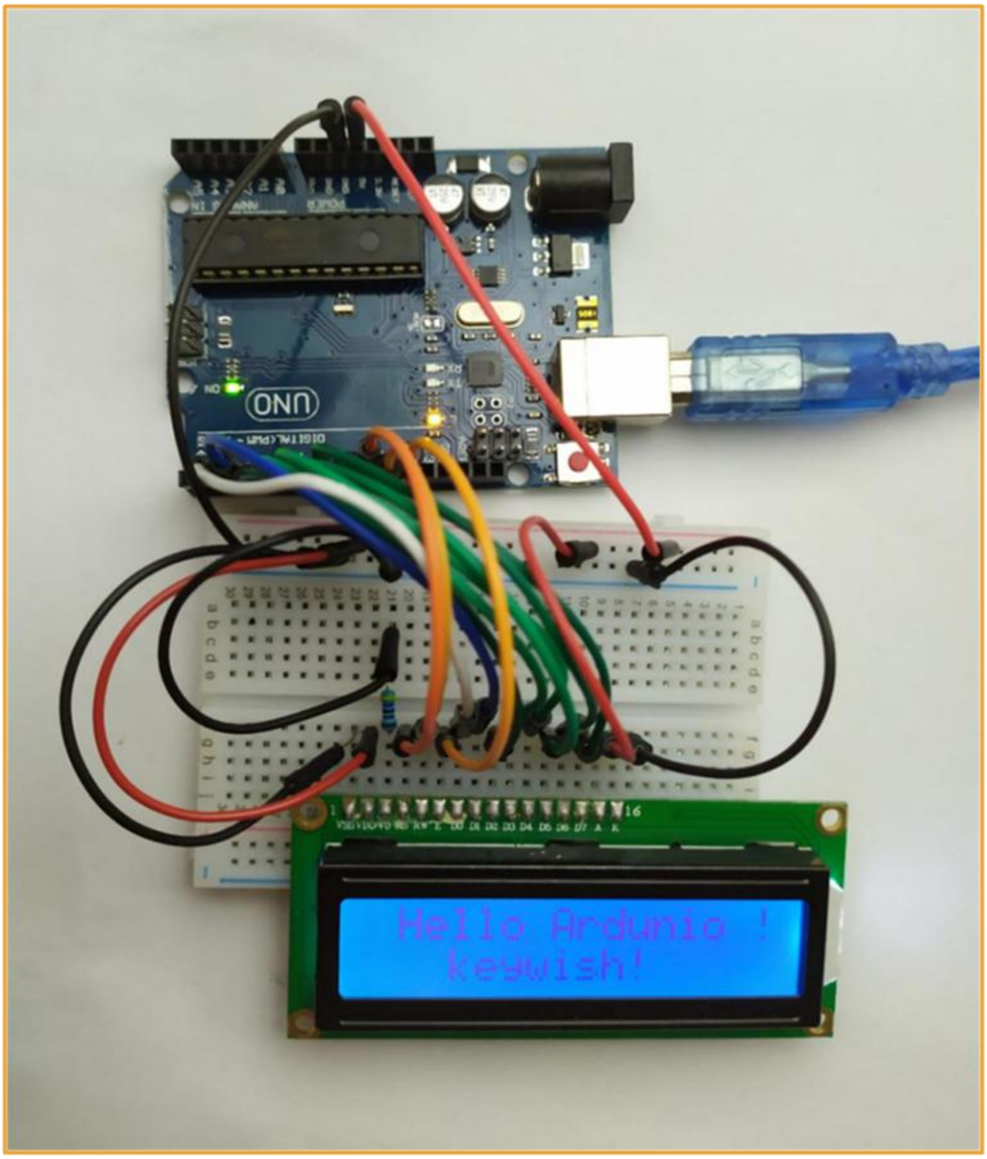
4-wire drive mode of LCD1602 based on Arduino:

```
#define    DB4   6   // lcd1602 DB4
#define    DB5   7   // lcd1602 DB5
#define    DB6   8   // lcd1602 DB6
#define    DB7   9   // lcd1602 DB7

#define    LCD1602_RS   10
#define    LCD1602_RW   11
#define    LCD1602_E    12

#include<LiquidCrystal.h>
LiquidCrystal lcd(LCD1602_RS,LCD1602_RW,LCD1602_E,DB4,DB5,DB6,DB7);
void setup(){
   lcd.begin(16,2);

}
   void loop(){
   lcd.clear();
   lcd.setCursor(0,0);
      lcd.print(" Hello Ardunio ! ");
   delay(10);
   lcd.setCursor(2,1);
   lcd.print(" keywish! ");

   delay(3000);
}
```

Experiment Result

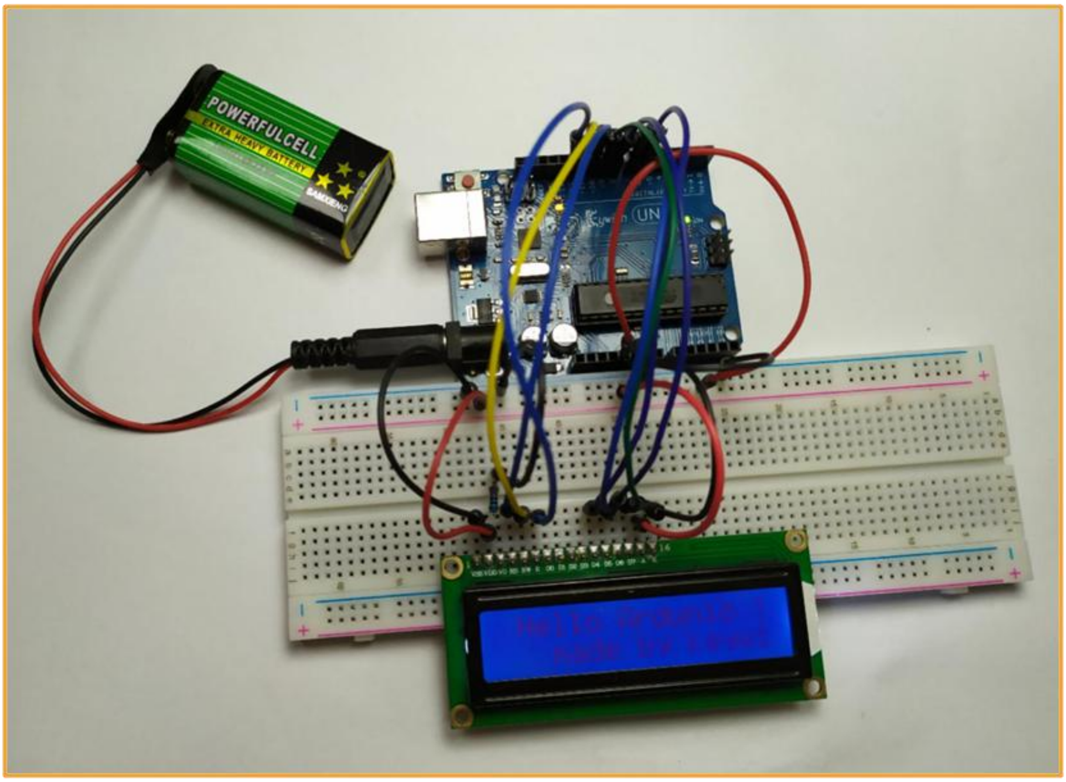
8-wire drive mode of LCD1602 based on Arduino:

The advantage of the eight-wire mode is that each data is transmitted from a separate data line, so the data transmission speed is fast.But the disadvantages are obvious, requiring a large number of ports.



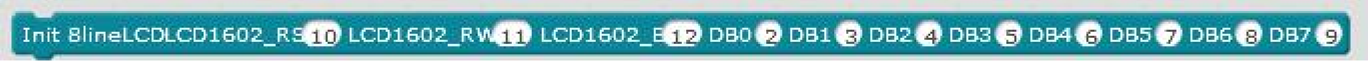
4-wire drive mode of LCD1602 based on Arduino:

The advantage of the four-wire mode over the eight-wire mode is that it occupies fewer ports and reduces the number of four data lines. In other words, eight-bit data needs to be sent through four lines, so the data speed of the four-wire mode is slower than that of the eight-wire mode.



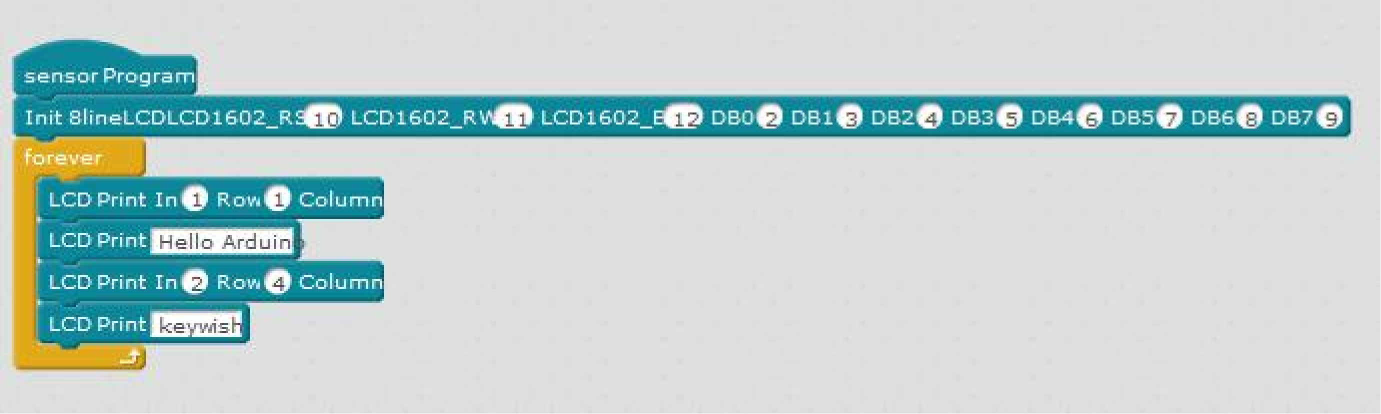
Mblock programming program

8-wire drive mode of LCD1602 based on Arduino:

- 

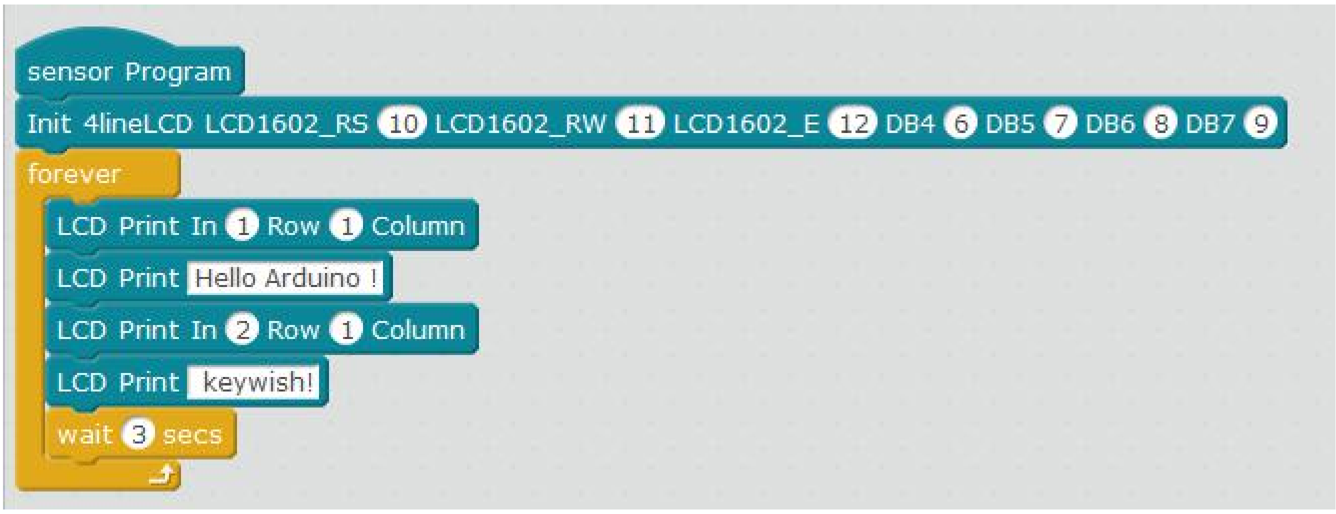
```
Init 8lineLCDLCD1602_RS 10 LCD1602_RW 11 LCD1602_E 12 DB0 2 DB1 3 DB2 4 DB3 5 DB4 6 DB5 7 DB6 8 DB7 9
```

--Initialization of pins



```
sensor Program
Init 8lineLCDLCD1602_RS 10 LCD1602_RW 11 LCD1602_E 12 DB0 2 DB1 3 DB2 4 DB3 5 DB4 6 DB5 7 DB6 8 DB7 9
forever
    LCD Print In 1 Row 1 Column
    LCD Print Hello Arduin
    LCD Print In 2 Row 4 Column
    LCD Print keywish
```

4-wire drive mode of LCD1602 based on Arduino:



```
sensor Program
Init 4lineLCD LCD1602_RS 10 LCD1602_RW 11 LCD1602_E 12 DB4 6 DB5 7 DB6 8 DB7 9
forever
    LCD Print In 1 Row 1 Column
    LCD Print Hello Arduino !
    LCD Print In 2 Row 1 Column
    LCD Print  keywish!
    wait 3 secs
```