```
#include <LiquidCrystal.h>
#include <math.h>
#include <SimpleDHT.h>

LiquidCrystal lcd(8, 9, 4, 5, 6, 7);

const int menuCount = 5; // number of sensors
const char *menuLabels[menuCount] = { // list of active sensors
  //1234567890123456
    "Temperature     ", // A1
    "Light           ", // A2
    "Humidity        ", // A3
    "Magnetism       ", // A4
    "Proximity       "  // A5
    };

int i = 0; // "i" indicates the current sensor
int m = 1; // "m" marks the status of the menu
           // (1) being in the main menu
          // (0) displaying the value measured by the selected sensor

enum Button { // Menu buttons
    right,
    up,
    down,
    left,
    select,
    none
};
Button readLcdButtons() { // function that checks which button has been pressed
    const int keyIn = analogRead(0); // "keyIn" stores the value transmitted by
the board by pressing a button
    if (keyIn < 50) return Button::right;
    if (keyIn < 250) return Button::up;
    if (keyIn < 450) return Button::down;
    if (keyIn < 650) return Button::left;
    if (keyIn < 850) return Button::select;
    return Button::none;
}

void initMenu() { // (returns to) displaying the first line of the main menu
    m = 1; // mark the status of the menu at the main menu
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Choose (UP/DOWN)");
}

void setup() {
    //Serial.begin(115200);
    lcd.begin(16, 2);
    initMenu();
}

long lastUpdate = 0;

void loop() { // main
    const long currentTime = millis();
    if (m == 0 && (currentTime - lastUpdate) > 200) {
      // a sensor has been selected and it passed 200 milliseconds since the last
update
        lastUpdate = currentTime;
        lcd.setCursor(9, 1);
        lcd.print("         "); //clean the LCD portion where the measured value
will be written
        lcd.setCursor(9, 1);
        // check the selected sensor
        switch (i) {
        case 0: { // *** Temperature
****************************************************************************
            double temp = 0;
            // will calculate the average of 10 measurements
            for (int i = 1; i < 10; i++) {
                temp += analogRead(1);
            }
            temp /= 50.0;
           lcd.print(round(temp)); // an approximation of the measurements made
will be displayed
            lcd.print(' ');
            char c=222; // the symbol for Celsius degree
            lcd.print(c);
            lcd.print('C');
            } break;
        case 1: { // *** Light
**********************************************************************************
            const int value = analogRead(2);
            lcd.print(value);
            } break;
        case 2: { // *** Humidity
*******************************************************************************
            const int value = analogRead(3);
            const int temperature=25;
            float humidity= (161.0*value/1023.0 - 25.8) / (1.0546 - 0.
0026*temperature); // calculate the humidity in percent
            lcd.print(humidity);
            } break;
        case 3: { // *** Magnetism
******************************************************************************
            int value = analogRead(4); // the value measured by sensor gives us
the polarity
            if (value < 500) {
                lcd.print("North");
            } else if (value > 600) {
                lcd.print("South");
            } else {
                lcd.print("none");
            }
            } break;
        case 4: { // *** Proximity
******************************************************************************
           lcd.print(analogRead(5)); // print the value measured by sensor
            } break;
        }
    }

    const Button button = readLcdButtons();
    // check which button has been pressed
    switch (button) {
    case Button::right: { // we do not use this button for anything
        } break;
    case Button::left: { // (returns to) displaying the first line of the main
menu
        initMenu();
        } break;
    case Button::up: { // navigate the list of sensors upwards
        if (i == 0) { // if the index reaches the top of the list you will need
to jump to the end of it
            i = menuCount - 1;
            delay(500);
        } else {
           i--; // the index decreases by marking the previous sensor in the list
            delay(500);
        }
        } break;
    case Button::down: { // navigate the list of sensors downwards
        if(i == menuCount - 1) { // if the index reaches the end of the list it
will be necessary to jump to its beginning
            i = 0;
            delay(500);
        } else {
           i++; // the index increases by marking the next sensor in the list
            delay(500);
        }
        } break;
    case Button::select: { // a sensor has been selected and the measured value
is to be displayed
        m = 0; // mark the status of the menu to display the measured value for
the selected sensor
```

```
        lcd.setCursor(0, 0); // set the cursor on the first line of the LCD
        lcd.print("< ");
        lcd.setCursor(2, 0);
        lcd.print(menuLabels[i]); // the name of the current sensor will be
displayed from the list
        lcd.setCursor(0,1); // set the cursor on the second line of the LCD
        lcd.print("  Value: ");
        } break;
    case Button::none: {
        if (m==1) { // being in the main menu
            lcd.setCursor(0, 1); // set the cursor on the second line of the LCD
            lcd.print(menuLabels[i]); // the name of the current sensor will be
displayed from the list
        }
        } break;
    }
}
```