User Manual to a Shelter for an Outdoor Aquaponic System

User manual

1) Introduction

This is the User Manual to a Shelter for an Outdoor Aquaponic System. The shelter's function is to measure the temperature and humidity, save the plants and the protect from the bugs, and grant the appropriate temperature for the plants and fishes.

3D model

2) System Overview

The system can work only outside. The policarbonate panels does not prevent the sunlight, that needs for the plants.

The prototype is using passive cooling system, that is meaning , if the wind is blowing it can flow throught the shelter. The other version of the model is using servo motors to open the windows, if the measured temperature is too high and close when the temperature is too low.

3) Technical parameters

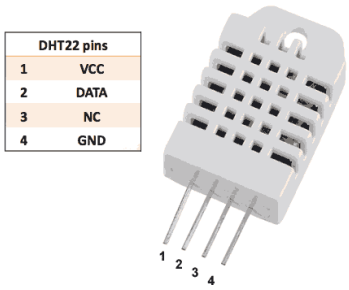
There are presented the most important technical parameters for the electrical system of the shelter:

Product	Description	Value	Accuaracy
Arduino UNO	Microcontroller	12V	-
Power supply	AC/DC	12V / 1.5 A	-
DHT22	temperature sensor	-40 °C to +80°C	± 0.5 °C
DHT22	humidity sensor	0% to 100%	± 2%
MC1602C-SYR	LCD display	2X16 line*column	-

4) How to connect electorinic components

It is a short guide to help to connect the diveces .

Sensor:

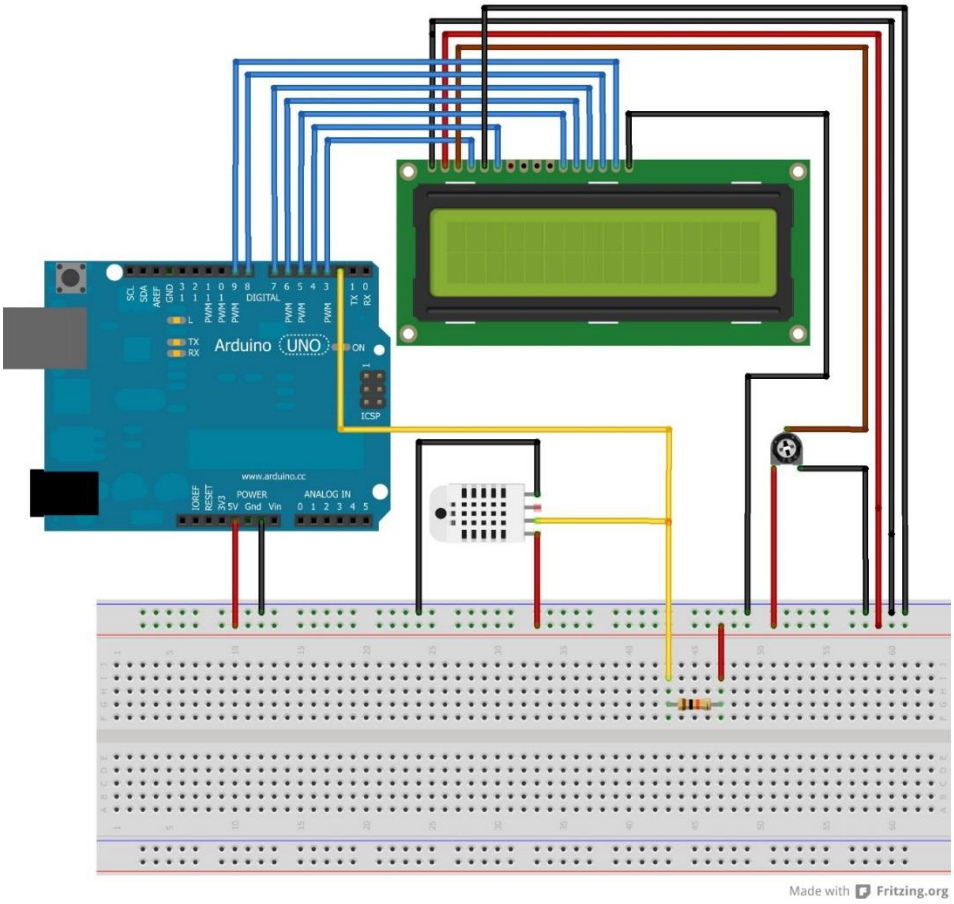


LCD Shield:

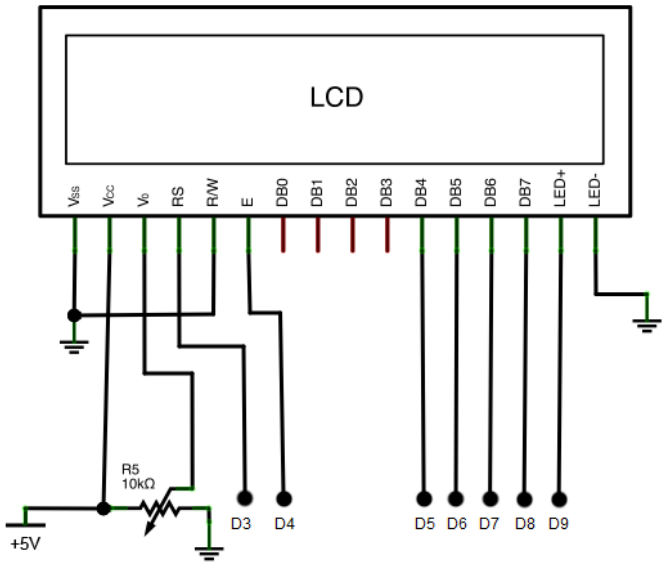
Power Supply:

Arduino UNO:

Electronic schematic:



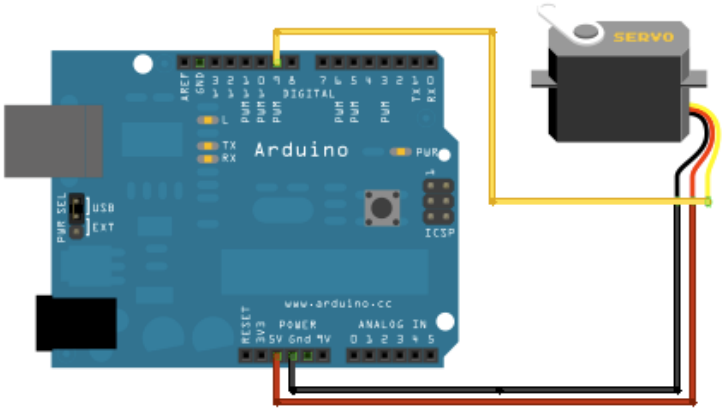
LCD Connection:



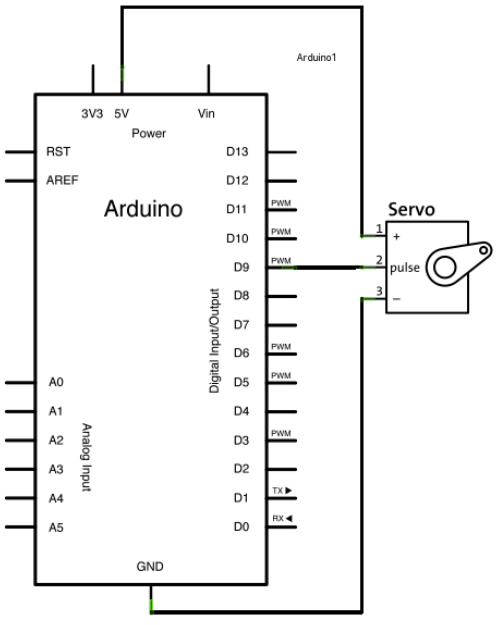
There is an option, if someon use servo motors to open and close the windows.

Servo motor:

Connection for Arduino:



Shematic:



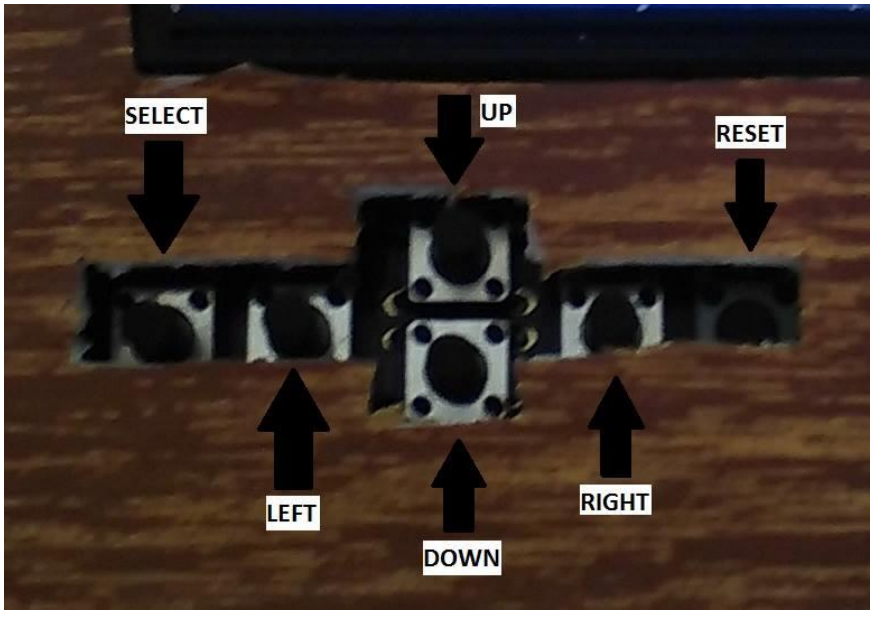
Attention!

The positive and negative pollution should not be mixed up. It is able to damage the Arduino UNO.

5) Programing

There is an option to change the windows opener and closer value. To go into menu we click and hold select button. when menu appears we choose if we want to change maximal temp or minimal by clicking up or down. We select choosen by clicking select. To change temperature we click up and down. To confirm we click select. To leave the menu we click right button.

Allocation of buttons:



There is the code, what the arduino is using to work:

```
// include the libraries:
#include "DHT.h"
#include <LiquidCrystalFast.h>
#define DHTPIN A5     // what pin we're connected to
#define DHTTYPE DHT22   // DHT 22  (AM2302)
DHT dht(DHTPIN, DHTTYPE);
// initialize the library with the numbers of the interface pins
LiquidCrystalFast lcd(8, 9, 4, 5, 6, 7);
int lcd_key     = 0;
int adc_key_in  = 0;
int lcd_key_menu     = 0;
int lcd_key_temp     = 0;
#define btnRIGHT  0
#define btnUP     1
#define btnDOWN   2
#define btnLEFT   3
#define btnSELECT 4
#define btnNONE   5
int maxTemp = 20;
int minTemp = 17;
int button_pressed = 0;
int max_min_temp = 0;               //Defines if you want to change max or min temperature in menu
int temp_changed = 0;               //Defines if temperature changing was finished
int first_menu_exit = 0;            //Defines if user wants to exit from first menu
int read_LCD_buttons(){             // read the buttons
adc_key_in = analogRead(0);       // read the value from the sensor
if (adc_key_in > 1000) return btnNONE;
```

```
   // 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;
   return btnNONE;              // when all others fail, return this.
}
void setup() {
  dht.begin();
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  lcd.print(F("Reading sensor"));
}
void loop() {
  delay(2000);
  int h = dht.readHumidity(); //read humidity
  // Read temperature as Celsius
  float t = dht.readTemperature();
  // Check if any reads failed and exit early (to try again).
  if (isnan(h) || isnan(t)) {
    lcd.println(F("Failed to read from DHT sensor!"));
    return;
  }
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("T:"));
    lcd.print(t);
    lcd.print((char)223);
    lcd.print(F("C "));
    lcd.print(F("H:"));
```

```
 lcd.print(h);
 lcd.print(F("%"));
if(t>maxTemp){
  lcd.setCursor(2,1);
  lcd.print(F("Open windows"));
}
if(t<=maxTemp){
  lcd.setCursor(2,1);
  lcd.print(F("Open windows"));
}
if(t<minTemp){
  lcd.setCursor(2,1);
  lcd.print(F("Close windows"));
}
 float degreees = (maxTemp - minTemp);
 if(t>=(minTemp + (degreees-1))){
   lcd.setCursor(2,1);
   lcd.print(F("Open windows"));
}
lcd_key = read_LCD_buttons();   // read the buttons
 switch (lcd_key){        // depending on which button was pushed, we perform an action
    case btnSELECT:{
      lcd.clear();
      lcd.print(F(">Change max temp"));
      lcd.setCursor(0,1);
      lcd.print(F("Change min temp"));
      delay(500);
      while(1){
         button_pressed = 1;
          lcd_key_menu = read_LCD_buttons();
          switch (lcd_key_menu){
```

```
case btnUP:{
  max_min_temp = 0;
  break;
}
case btnDOWN:{
  max_min_temp = 1;
  break;
}
  case btnRIGHT:{
  first_menu_exit = 1;
  break;
}
case btnSELECT:{
  if(max_min_temp == 0){                          //Change max temperatur
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print(F("Current max: "));
    lcd.print(maxTemp);
    lcd.setCursor(0,1);
    lcd.print(F("New: "));
    lcd.print(maxTemp);
    delay(100);
    while(1){
      lcd_key_temp = read_LCD_buttons();
      switch (lcd_key_temp){
      case btnUP:{
        lcd.setCursor(0,1);
        lcd.print(F("New: "));
        if(((maxTemp+1) <= 40) && (maxTemp>minTemp)){
          maxTemp++;
        }
```

```
      lcd.print(maxTemp);
      delay(300);
      break;
     }
     case btnDOWN:{
      lcd.setCursor(0,1);
      lcd.print(F("New: "));
      if(((maxTemp-1) >= 15) && (maxTemp-1>minTemp)){
       maxTemp--;
      }
      lcd.print(maxTemp);
      delay(300);
      break;
     }
     case btnSELECT:{
      temp_changed = 1;
      break;
     }
     case btnNONE:{
       break;
     }
    }
    if(temp_changed == 1){
     temp_changed = 0;
     lcd.clear();
     break;
    }
  }
}
if(max_min_temp == 1){                          //Change min temperature
  lcd.clear();
```

```
lcd.setCursor(0,0);
lcd.print(F("Current min: "));
lcd.print(minTemp);
lcd.setCursor(0,1);
lcd.print(F("New: "));
lcd.print(minTemp);
delay(500);
while(1){
 lcd_key_temp = read_LCD_buttons();
 switch (lcd_key_temp){
 case btnUP:{
  lcd.setCursor(0,1);
  lcd.print("New: ");
  if(((minTemp+1) < maxTemp) && (minTemp<30)){
   minTemp++;
  }
  lcd.print(minTemp);
  delay(300);
  break;
 }
 case btnDOWN:{
  lcd.setCursor(0,1);
  lcd.print(F("New: "));
  if(((minTemp-1) > 10) && (minTemp-1<maxTemp)){
   minTemp--;
   }
  lcd.print(minTemp);
  delay(300);
  break;
 }
  case btnSELECT:{
```

```
            lcd.clear();
            temp_changed = 1;
            break;
           }
           case btnNONE:{
            break;
           }
          }
         if(temp_changed == 1){
           temp_changed = 0;
           lcd.clear();
           break;
         }
        }
      }
     break;
    }
    case btnNONE:{
    break;
    }
 }
if(max_min_temp == 0){
  lcd.setCursor(0,1);
  lcd.print(F("Change min temp"));
  lcd.setCursor(0,0);
  lcd.print(F(">Change max temp"));
  delay(125);
}

 if(max_min_temp == 1){
  lcd.setCursor(0,0);
```

```
            lcd.print(F("Change max temp"));
            lcd.setCursor(0,1);
            lcd.print(F(">Change min temp"));
            delay(125);
           }
           if(first_menu_exit == 1){
            first_menu_exit = 0;
            break;
           }
       }
       lcd.clear();
        break;
     }
     case btnNONE:{
          button_pressed = 0;
          break;
     }
  }
  if(button_pressed == 0){
    delay(500);
  }
}
```

Dear Costumer!

The ECODOME team is very grateful, that you chose us. Thank you for your trust.