

ДОДАТОК А
ЛІСТИНГ ПРОГРАМИ КЕРУВАННЯ ДЛЯ ARDUINO

```
#include <Wire.h>
#include "AHT20.h"
#include "LiquidCrystal_I2C.h"
#include "BH1750.h"
#define W_PUMP_RELAY 7
#define COOLER_RELAY 6
#define LED_RELAY 5
AHT20 humiditySensor;
BH1750 lightMeter;

//LiquidCrystal_I2C lcd(0x27,16,4);

int SENSOR_HUMIDITY_ANALOG = A0;
int val_humidity;
bool switcher_grunt = false;
bool switcher_Led = false;
bool switcher_Cooler = false;
String MyData;
int L;

void setup()
{
  Serial.begin(9600);

  Wire.begin();

  pinMode(W_PUMP_RELAY, OUTPUT);
  pinMode(COOLER_RELAY, OUTPUT);
  pinMode(LED_RELAY, OUTPUT);
```

```
}

```

```
void loop() {

```

```
  if(Serial.available())

```

```
  {

```

```
    float temperature = TEM();

```

```
    float humidity = HUM();

```

```
    float LUX_PA = LUX_PARAMETR();

```

```
    MyData = Serial.readString();

```

```
    L = MyData.charAt(0);

```

```

    if(L=='A')

```

```
    {

```

```
      for(;;)

```

```
      {

```

```

          String resultString = String(LUX_PA, 1) + "      " + String(humidity, 1) + "
" + String(temperature, 1) + "      ";

```

```
          Serial.println(resultString);

```

```
          MyData = Serial.readString();

```

```
          L = MyData.charAt(0);

```

```

          if (L=='S'){break;}

```

```

          switcher_grunt = false;

```

```
          if (!switcher_grunt)

```

```
          {

```

```
            if(humidity_ground()>=390)

```

```
{  
    digitalWrite(W_PUMP_RELAY,HIGH);  
    delay(5000);  
    digitalWrite(W_PUMP_RELAY, LOW);  
    delay(8000);  
    switcher_grunt = true;  
} else digitalWrite(W_PUMP_RELAY, LOW);  
}
```

```
switcher_Led = false;
```

```
if (!switcher_Led)
```

```
{
```

```
    if(LUX_PARAMETR()<=500)
```

```
    {
```

```
        digitalWrite(LED_RELAY,HIGH);
```

```
        delay(3000);
```

```
        switcher_Led = true;
```

```
    } else digitalWrite(LED_RELAY, LOW);
```

```
}
```

```
switcher_Cooler = false;
```

```
if (!switcher_Cooler)
```

```
{
```

```
    if(HUM()>=70)
```

```
    {
```

```
        digitalWrite(COOLER_RELAY,HIGH);
```

```
        delay(3000);
```

```
    switcher_Cooler = true;
  } else digitalWrite(COOLER_RELAY, LOW);
}
}
```

```
digitalWrite(W_PUMP_RELAY,LOW);
digitalWrite(COOLER_RELAY,LOW);
digitalWrite(LED_RELAY,LOW);
}
```

```
else if (L=='S')
{
  digitalWrite(W_PUMP_RELAY,LOW);
  digitalWrite(COOLER_RELAY,LOW);
  digitalWrite(LED_RELAY,LOW);
}
else if (L=='D')
{
  digitalWrite(W_PUMP_RELAY,HIGH);
}
else if (L=='F')
{
  digitalWrite(W_PUMP_RELAY,LOW);
}
else if(L=='G')
{
  digitalWrite(LED_RELAY,HIGH);
}
else if (L=='H')
```

```

    {
    digitalWrite(LED_RELAY,LOW);
    }
    else if(L=='J')
    {
    digitalWrite(COOLER_RELAY,HIGH);
    }
    else if (L=='K')
    {
    digitalWrite(COOLER_RELAY,LOW);
    }

}

float temperature = TEM();
float humidity = HUM();
float LUX_PA = LUX_PARAMETR();

String resultString = String(LUX_PA, 1) + " " + String(humidity, 1) + " " +
String(temperature, 1) + " ";
delay(1000);
Serial.println(resultString);
}

int humidity_ground() // Функція зчитування вологості землі з датчика
{
    val_humidity = analogRead(SENSOR_HUMIDITY_ANALOG);
    return val_humidity;
}

```

```
float LUX_PARAMETR() // Функція зчитування освітлення
{
    BH1750 lightMeter;
    lightMeter.begin();
    float lux = lightMeter.readLightLevel();
    return lux;
}
```

```
float HUM() // Функція зчитування вологості повітря
{
    float humidity = humiditySensor.getHumidity();
    return humidity;
}
```

```
float TEM() // Функція зчитування температури повітря
{
    float temperature = humiditySensor.getTemperature();
    return temperature;
}
```

ДОДАТОК Б
ЛІСТИНГ ПРОГРАМИ У VISUAL STUDIO

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO.Ports;
using System.Data.SqlTypes;
using System.Threading.Tasks;

namespace WindowsFormsApp1_Better
{
    public partial class Form1 : Form
    {
        private SerialPort serialPort;
        public Form1()
        {
            InitializeComponent();
            serialPort1.Open();
            red1.Visible = true;
            green1.Visible = false;
        }

        String dataIN;
        String dataT = "0";
```

```
private void SERIAL(object sender, SerialDataReceivedEventArgs e)
{
    BeginInvoke(new EventHandler(ClearTxt));
    dataIN = serialPort1.ReadLine();
}
```

```
private void ClearTxt(object sender, EventArgs e)
{

    dataT = dataIN;
    label5.Text = dataT;

}
```

```
private void Automatic_control_Click(object sender, EventArgs e)
{
    red1.Visible = false;
    green1.Visible = true;
    gren_left.Visible = false;
    red_left.Visible = true;
    serialPort1.Write("A");
}
```

```
private void Manuel_control_Click(object sender, EventArgs e)
{
    red1.Visible = true;
    green1.Visible = false;
    gren_left.Visible = true;
    red_left.Visible = false;
    serialPort1.Write("S");
}
```

```
}
```

```
private void Turn_ON_Water_Click(object sender, EventArgs e)
{
    serialPort1.Write("D");
}
```

```
private void Turn_OFF_Water_Click(object sender, EventArgs e)
{ serialPort1.Write("F");

}
```

```
private void Turn_ON_LED_Click(object sender, EventArgs e)
{
    serialPort1.Write("G");
}
```

```
private void Turn_OFF_LED_Click(object sender, EventArgs e)
{
    serialPort1.Write("H");
}
```

```
private void Turn_ON_COOLER_Click(object sender, EventArgs e)
{
    serialPort1.Write("J");
}
```

```
private void Turn_OFF_COOLER_Click(object sender, EventArgs e)
{
    serialPort1.Write("K");
}
```

```
}  
  
private void Form1_FormClosed(object sender, FormClosedEventArgs e)  
{  
    if (serialPort1.IsOpen) serialPort1.Close();  
}  
}  
}
```

ДОДАТОК В
ДЕМОНСТРАЦІЙНИЙ МАТЕРІАЛ У ВИГЛЯДІ ПРЕЗЕНТАЦІЇ

