```
#include “DHT.h”                          // Include the dht library to interface with the DHT sensor
#include <Wire.h>                         // IC library to use the light sensor
#include <BH1750.h>                       // Import BH1750 library in order to be able to use the light sensor
//---------------------------------------------- Define the control pins ----------------------------------------------------
Const int Pump=4;                         // We’ve used this pin to control the motor pump
Const int Fan=5;                          // Use this pin as a PWM output to control the Fan speed
Const int Light=6;                        // Use this pin in order to control the LED brightness
Const int TempHum=7;                      // Input pin for DHT sensor
Const int Moisture=8;                     // Digital Input pin to read the moisture command signals
Const int TestLED=9;                      // Use this output pin to test the right soldering on your PCB by turning on and off the test LEDs
Const int Brightness=12;                  // Input pin to read the light sensor signals
Const int analogMoistue=0;                // Analog Input to read the analog signal from moisture sensor
//-------------------------------------------------------------------------------------------------------------------------
#define DHTTYPE DHT11                     // DHT 22  (AM2302), AM2321
DHT dht(TempHum, DHTTYPE);                // Temperature and humidity sensor constructor
BH1750 lightMeter;                        // Light sensor constructor
Char Data=’x’;                            // Store Serial data in this variable
String cmd=””;                            // Read the full instruction sent from the android app
Int flagModeAuto=0;                       // flag to activate the auto mode
Int sprayCMD=0;                           // CMD limit spray
Int LightCMD=0;                           // CMD limit brightness
Int lightLevelPlus=0;                     // Variable to control the light brightness
Float temperature=0;                      // Variable to store the temperature value
Float humidity=0;                         // Variable to store the humidity value
Uint16_t lux=0;                           // Variable to read light brightness from the light sensor
//---------------------------------------------- Start the pin configuration ----------------------------------------------
```

```
Void setup()
{

  Wire.begin();                          // Initialize the I2C bus (BH1750 library doesn't do this
automatically)
  Dht.begin();                          // Start the temperature and humidity sensor reading
  lightMeter.begin();                      // Start the light sensor reading
  Serial.begin(9600);                      // Set the baudrate up to 9600 BPS to communicate with the
android app through Bluetooth
  Serial.setTimeout(100);                    // Set the time to wait for data before closing the Serial
port (after 100 ms)
  pinMode(Pump,OUTPUT);
  pinMode(Fan,OUTPUT);
  pinMode(Light,OUTPUT);
  pinMode(Moisture,INPUT);
  pinMode(TestLED,INPUT);
  delay(1000);
  digitalWrite(Pump,LOW);
  digitalWrite(Fan,LOW);
  digitalWrite(Light,LOW);
}

//------------------------------------------------ Start the Process code ------------------------------------------------------
/*Void loop()
{
  While(Serial.available())                  // Read the serial data once available
  {
    Delay(10);
    Data=Serial.read();
    Cmd+=Data;
```

```
}
If(cmd==”dt”)                          // Send the humidity value to the android app
{
  Serial.print(humidity);
}
If(cmd==”dh”)                          // Send the brightness value to the android app
{
  Serial.print(lux);
}
If(cmd==”db”)                          // Send the temperature value to the android app
{
  Serial.print(temperature);
}
If(cmd==”o”)                           // Activate the automatic mode
{
  flagModeAuto=1;
}
If(cmd==”m”)                           // Disactivate the automatic mode
{
  flagModeAuto=0;
}
If(flagModeAuto==1)
{
  autoPump();
  lightBrightness();
  autoFan();
}
If(flagModeAuto==0)
{
```

```
  If(cmd==”f”)                          // Turn ON the FAN
  {
    analogWrite(Fan,255);
  }
  If(cmd==”x”)                          // Turn OFF the FAN
  {
    analogWrite(Fan,0);
  }
  If(cmd==”l”)                          // Turn ON the Lights
  {
    analogWrite(Light,255);
  }
  If(cmd==”k”)                          // Turn OFF the Lights
  {
    analogWrite(Light,0);
  }
  If(cmd==”w”)                          // Turn ON the Pump
  {
    analogWrite(Pump,255);
  }
  If(cmd==”y”)                          // Turn OFF the Pump
  {
    analogWrite(Pump,0);
  }
 }
  Cmd=””;                               // Clear the cmd variable to make it available for the next
instruction
  Lux = lightMeter.readLightLevel();            // Get the brightness level from the light sensor
```

```
  Temperature=dht.readTemperature();                    // Get the temperature value from the DHT sensor
©
  Humidity=dht.readHumidity();                          // Get the humidity value from the DHT sensor (%)
}
//------------------------------------------------ Auto control function for Pump spray ------------------------------------
-------------------
Void autoPump()
 {
   If(analogRead(analogMoistue)<sprayCMD)
   {
     digitalWrite(Pump,HIGH);
     delay(1000);
     digitalWrite(Pump,LOW);
     delay(1000);
   }
 }

//------------------------------------------------ Auto control function for Brightness LED ---------------------------------
-----------------------
Void lightBrightness()
{
  While(lux<LightCMD)
   {
     analogWrite(Light,lightLevelPlus);                  // Increase the light brightness
     delay(100);
     lightLevelPlus++;
     lux = lightMeter.readLightLevel();                  // Read the light brightness level
   }
}
```

```
//------------------------------------------------ Auto control function for FAN ------------------------------------------------
Void autoFan()
{
  If(temperature>30)
  {
    analogWrite(Fan,255);                          // Turn ON the fan if the temperature exceed 30C
  }
  Else
  {
    analogWrite(Fan,0);                           // Turn OFF the fan if the temperature exceed 30C
  }
}
```