Greenhouse Monitoring

PIC18F2550 and Arduino Project

Submitted to:

Dr. Neelu Jain

Professor, Electronics and Communication Department

PEC University of Technology

Submitted by:

Divya Khandelwal

ME16207022

Semester: 2nd, M.E (Electronics)

Subject: Microcontroller and Embedded Systems

PREFACE

Greenhouse is a structure that the user used to grow the plants. It is built with a specific need for the type of plant they wish to grow. So the structure varies depending on type of plant and scale of size. Although it creates a perfect environment for plants, it needs human care to control the optimum status of the house such as ventilation.

Automated greenhouse is to ease people when they wish to grow plants. It helps to monitor the situation, when they are not at home. The main aim of this project is monitor the in-house environment status and accordingly the owner can take appropriate measures to provide the plants a suitable environment. A single unit of the greenhouse structure prototype has been constructed and integrated with the sensors. The control system is designed with PIC microcontroller (PIC18F2550) in simulation environment while for the hardware Arduino is used.

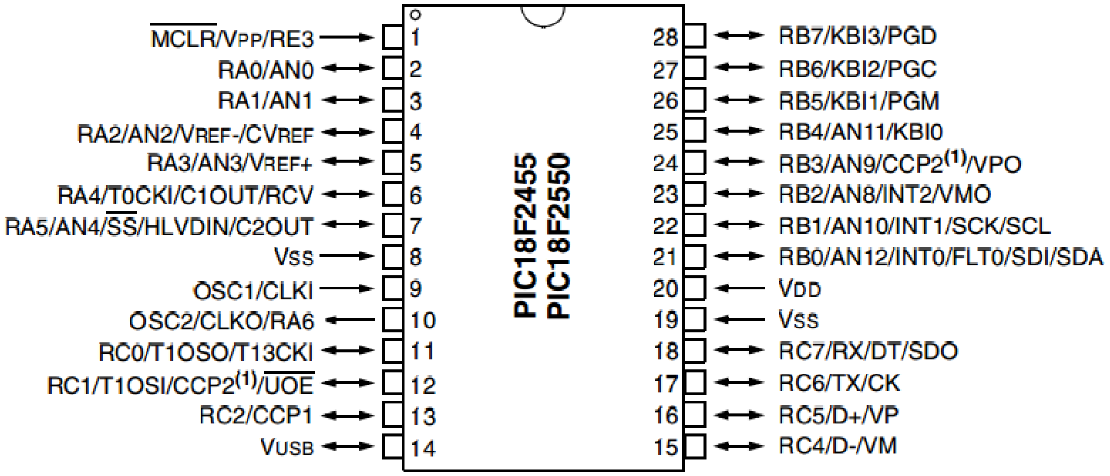
This project is based on simulating the environment of a greenhouse and displaying the whole model on the hardware with parameters like light intensity, humidity, temperature and soil moisture, being monitored using sensors.

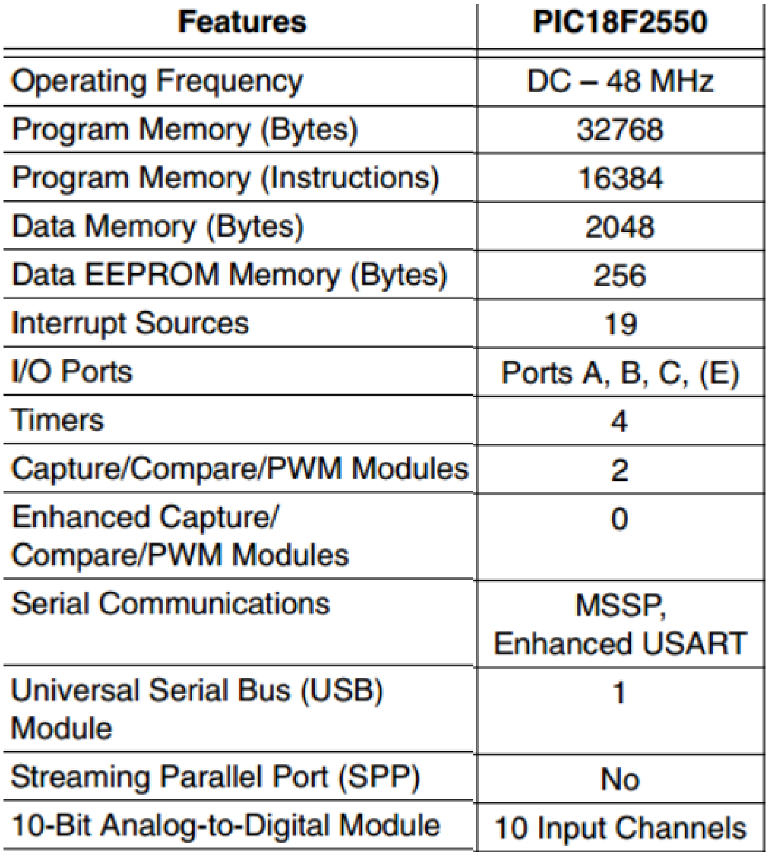
CONTENTS

ABOUT PIC18F2550

This family of devices offers the advantages of all PIC18 microcontrollers – namely, high computational performance at an economical price – with the addition of high endurance, Enhanced Flash program memory. In addition to these features, the PIC18F2455/2550/4455/4550 family introduces design enhancements that make these microcontrollers a logical choice for many high-performance, power sensitive applications.

28-Pin PDIP, SOIC

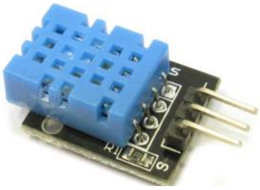




Features	PIC18F2550
Operating Frequency	DC – 48 MHz
Program Memory (Bytes)	32768
Program Memory (Instructions)	16384
Data Memory (Bytes)	2048
Data EEPROM Memory (Bytes)	256
Interrupt Sources	19
I/O Ports	Ports A, B, C, (E)
Timers	4
Capture/Compare/PWM Modules	2
Enhanced Capture/ Compare/PWM Modules	0
Serial Communications	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1
Streaming Parallel Port (SPP)	No
10-Bit Analog-to-Digital Module	10 Input Channels

COMPONENTS USED

1. DHT 11 Sensor



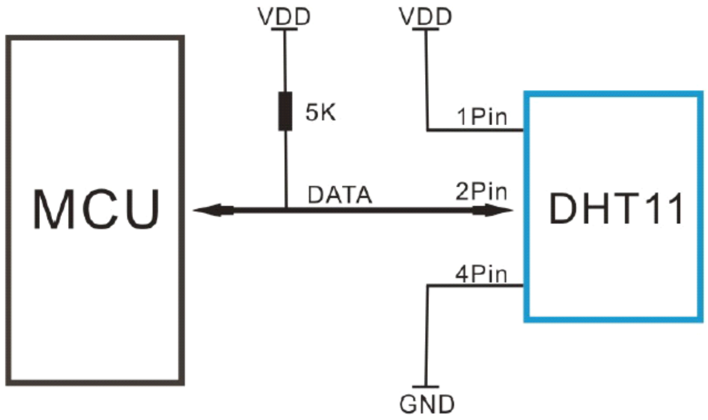
DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output.

This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Technical Specification:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 ℃	±5%RH	±2℃	1	4 Pin Single Row

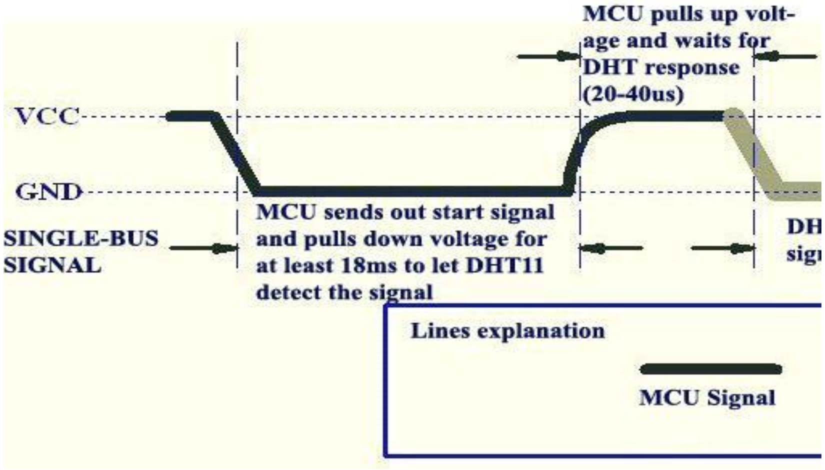
Connection with controller:



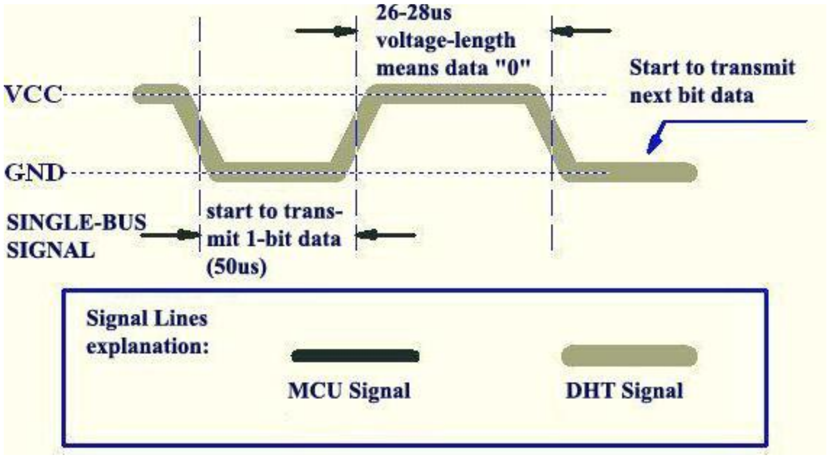
Communication with Controller:

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.
Data consists of decimal and integral parts. A complete data transmission is 40bit, and the sensor sends higher data bit first.
Data format: 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

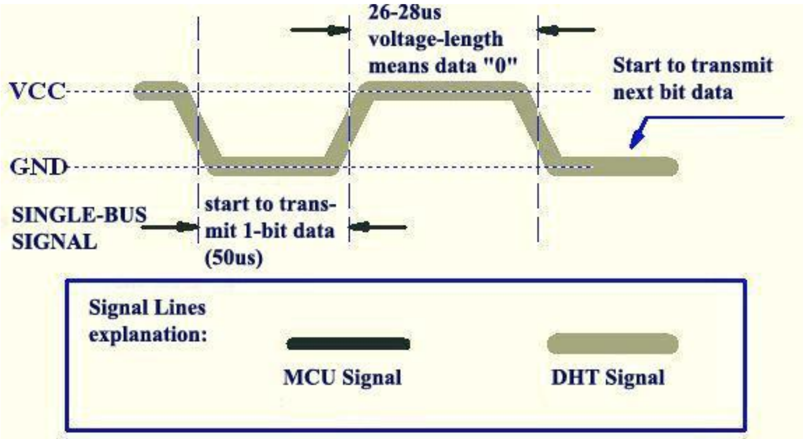
MCU Sends out Start Signal to DHT:


MCU pulls up voltage and waits for DHT response (20-40us)
VCC
GND
SINGLE-BUS SIGNAL
MCU sends out start signal and pulls down voltage for at least 18ms to let DHT11 detect the signal
DH
sig
Lines explanation
MCU Signal

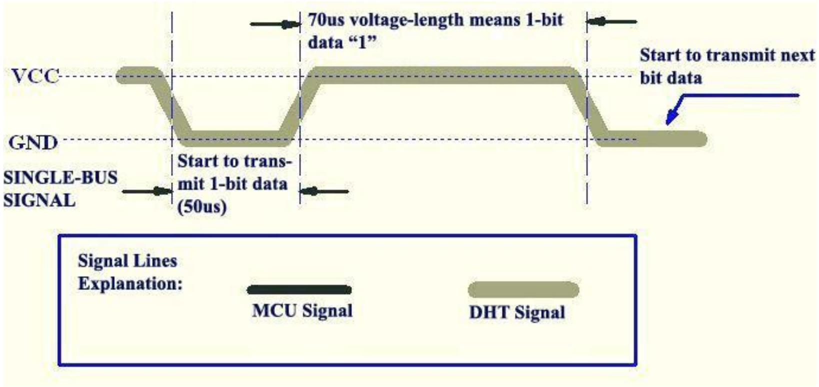
DHT Responses to MCU :


26-28us voltage-length means data "0"
Start to transmit next bit data
VCC
GND
SINGLE-BUS SIGNAL
start to transmit 1-bit data (50us)
Signal Lines explanation:
MCU Signal
DHT Signal

Data ‘0’ Indication



Data "1" Indication

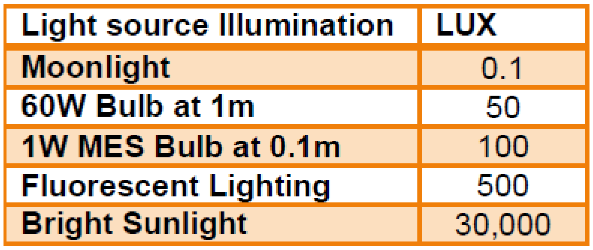


2. LDR Sensor

Two cadmium sulphide(cds) photoconductive cells with spectral responses similar to that of the human eye. The cell resistance falls with increasing light intensity.

Conversion of Resistance to Lux :

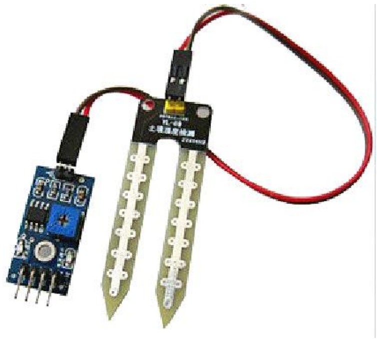
Vout = (LDRvalue * 0.0048828125);
resistance = (10000.0 * (5 - Vout))/Vout;
Lux value = (500 / resistance);



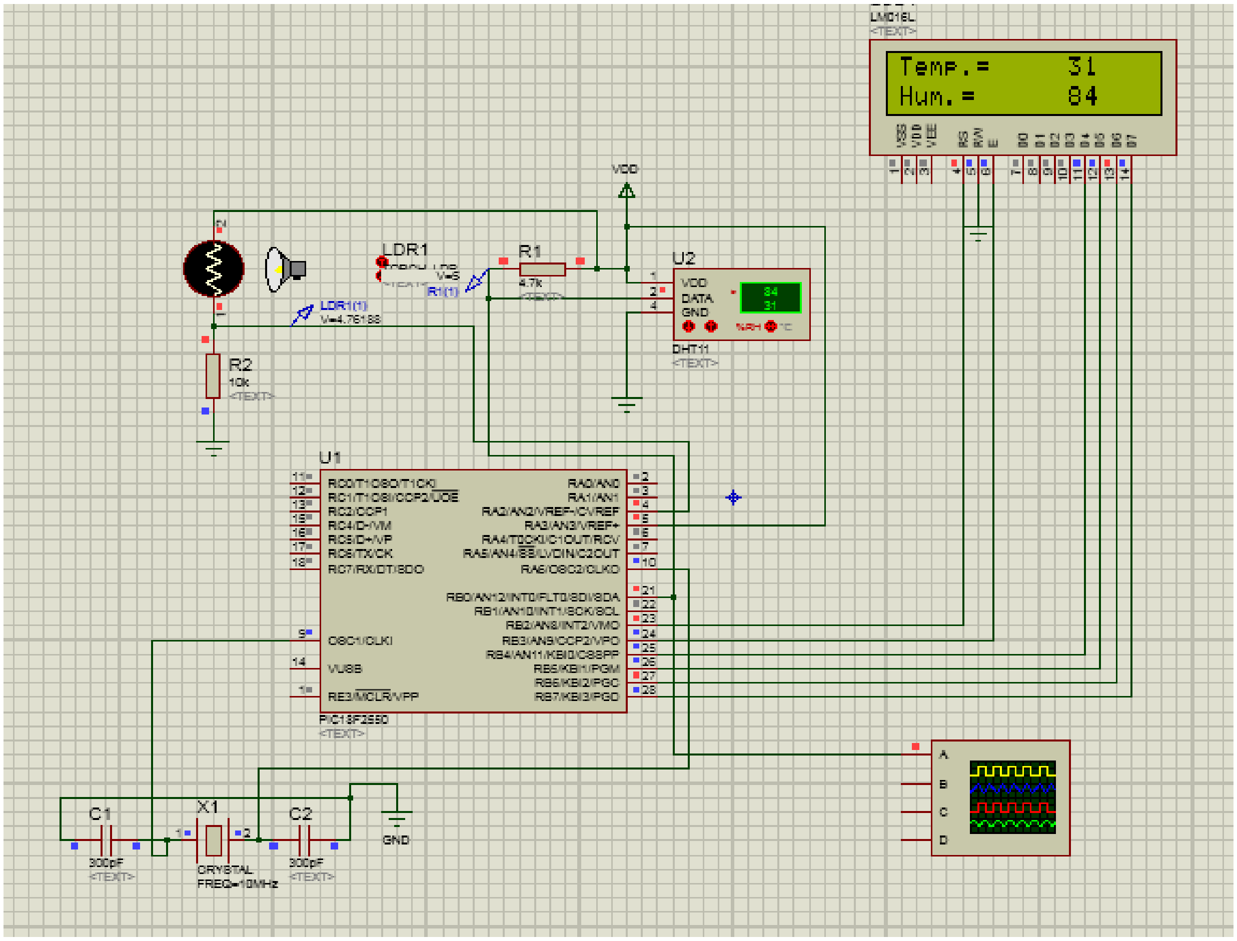
Light source Illumination	LUX
Moonlight	0.1
60W Bulb at 1m	50
1W MES Bulb at 0.1m	100
Fluorescent Lighting	500
Bright Sunlight	30,000

3. Moisture Sensor

This sensor can be used to test the moisture of soil, when the soil is having water shortage, the module output is at high level, else the output is at low level.



PROTEUS SIMULATION WITH PIC18F2550



SOURCE CODE WITH PIC18F2550

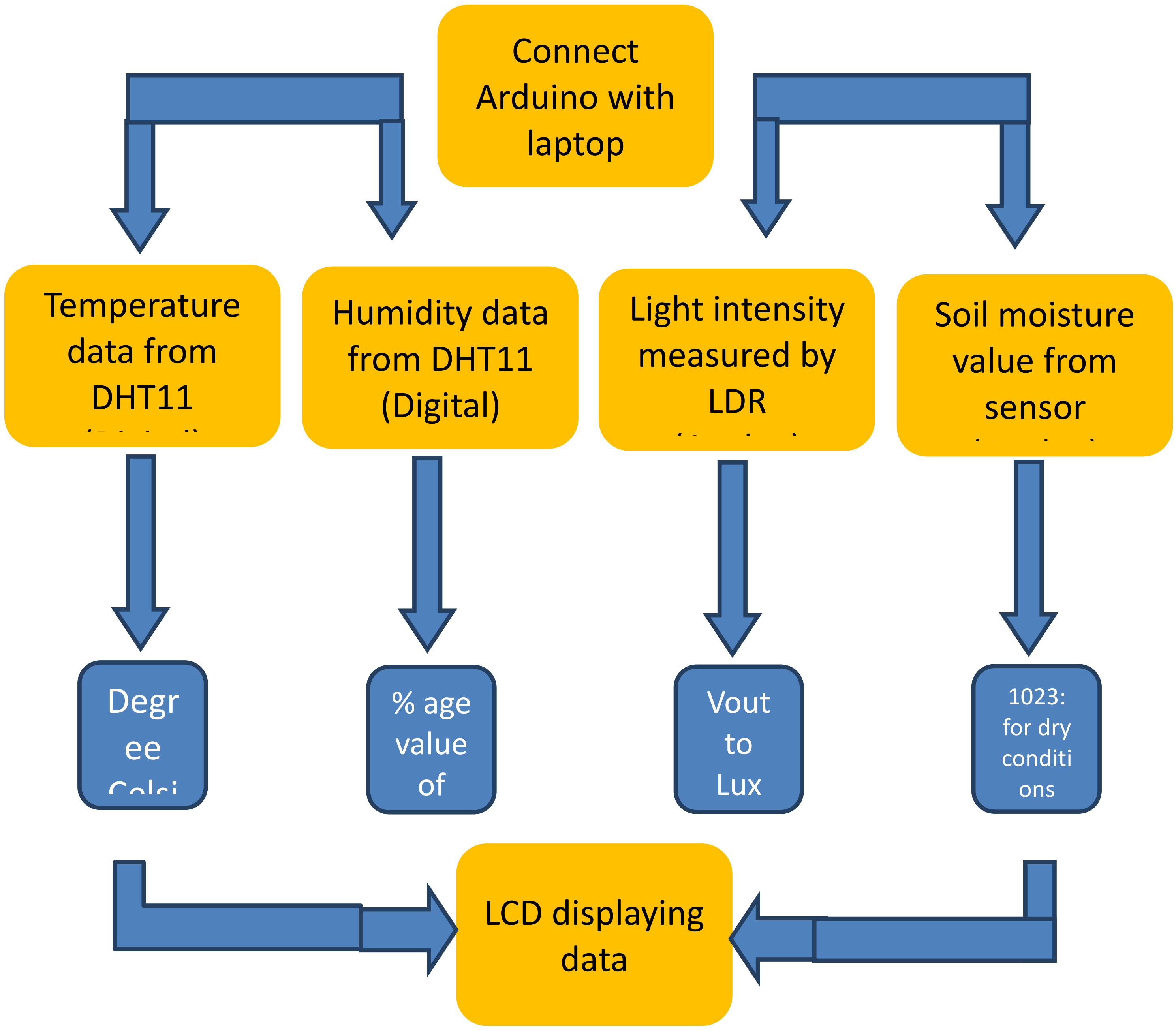
File :MyProject.c

```
#include <built_in.h>
#include "dht11.h"
#include "ldrWorking.c"

sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;

sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;

//DHT11 DEFINITONS
sbit DHT11_Pin at PORTB.B0;
sbit DHT11_DIR at TRISB.B0;
extern unsigned int DHT11_TMP;
extern unsigned int DHT11_HUM;
extern char DHT11_CHKSM;
//END OF DHT11 DEFINITONS

long veri;
int isi,nem;
char bekleme=0;
char txt[7];


//////////////////////////////////////////////

void main() {
  ADCON1=0x0f;
  TRISA = 0xFF; // PORTA is input as it receives values from ldr
  lcd_init();
  DHT11_init();
  while(1)
  {
   Dht11_Start();
   DHT11_Read();


if(DHT11_CHKSM==((DHT11_TMP>>8)+(DHT11_HUM>>8)+(DHT11_TMP&0xff)+
(DHT11_HUM&0xff)))
   {  Lcd_Cmd(_LCD_CURSOR_OFF);
      Lcd_Cmd(_LCD_CLEAR);
      lcd_out(1,1,"Temp.=");
      Lcd_Out(2,1,"Hum.=");
      inttostr(DHT11_TMP>>8,txt);
      lcd_out(1,8,txt);
      inttostr(DHT11_HUM>>8,txt);
       lcd_out(2,8,txt);
       delay_ms(1000);
  }

  Lcd_Cmd(_LCD_CLEAR);
    Lcd_Out(1, 1, "LDR :");    //Display string on LCD position
1,1
    adc_value = ADC_Read(2);    // Get 10-bit results of AD
conversion      (The LDR output is connected to PORTA.2 / ADC
port)
    ShowADC(1,7,adc_value);    //row=1, column=7, char value to
be displayed = adc_value
    delay_ms(1000);

 }
}
```

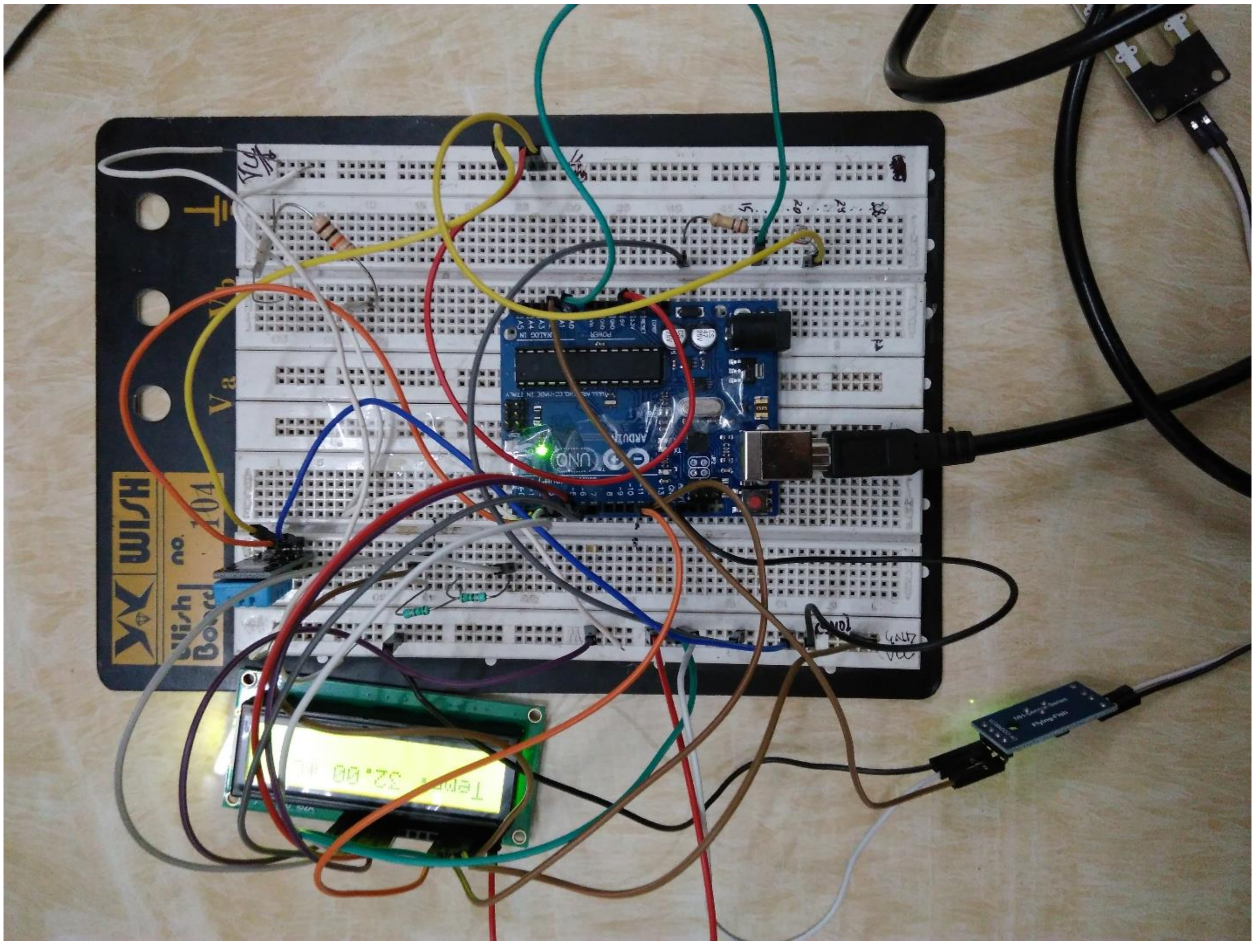
File : ldrWorking.c

```
float RLDR,Vout;
unsigned char *resistance = "00.00";

  // Routine to show the value of the ADC_read
  void ShowADC(int x, int y, float adc_value)
  {
  Lcd_Cmd(_LCD_CURSOR_OFF);

  Vout = (adc_value * 0.0048828125);           // Vout = Output
voltage from potential Divider. [Vout = adc_value * (Vin /
1024)]
   RLDR = (10000.0 * (5 - Vout))/Vout;     // Equation to
calculate Resistance of LDR, [R-LDR =(R1 (Vin - Vout))/ Vout]
   // R1 = 10,000 Ohms , Vin = 5.0 Vdc.
  lux_value = (500 / RLDR);


  floattostr(lux_value, *resistance);
  lcd_out(x,y,*resistance); }
```

FLOWCHART


Connect Arduino with laptop
Temperature data from DHT11
Humidity data from DHT11 (Digital)
Light intensity measured by LDR
Soil moisture value from sensor
Degr ee
% age value of
Vout to Lux
1023: for dry conditi ons
LCD displaying data

HARDWARE IMPLEMENTATION WITH ARDUINO



This image shows the working project with all the sensors working using arduino uno board and all the readings being displayed on LCD.

SOURCE CODE WITH ARDUINO

```
#include <LiquidCrystal.h>

// DHT Temperature & Humidity Sensor

#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>

#define DHTPIN            2         // Pin which is connected to
the DHT sensor.
#define DHTTYPE           DHT11     // DHT 11

DHT_Unified dht(DHTPIN, DHTTYPE);

uint32_t delayMS;
LiquidCrystal lcd(12, 11, 5, 4, 3, 6);   // initialize the
library with the numbers of the interface pins

int LDRpin = A0; // select the input pin for ldr
int LDRvalue = 0; // variable to store the value coming from the
sensor
int moisturePin = A1;
int moistureValue = 0;
int threshold_upper = 400;
int threshold_lower = 250;


void setup() {
  Serial.begin(9600);
  // Initialize device.
  dht.begin();
  lcd.setCursor(0, 0); // top left
  lcd.begin(16, 2);



  sensor_t sensor;
  dht.temperature().getSensor(&sensor);
  dht.humidity().getSensor(&sensor);

// Set delay between sensor readings based on sensor details.
  delayMS = sensor.min_delay / 1000;
```

```
}

void loop() {
  // Delay between measurements.
  delay(delayMS);


  /************ DHT WORKING CODE *******************/
  // Get temperature event and print its value.
  sensors_event_t event;
  dht.temperature().getEvent(&event);
  if (isnan(event.temperature)) {
    Serial.println("Error reading temperature!");
  }
  else {
    Serial.print("Temperature: ");
    Serial.print(event.temperature);
    Serial.println(" *C");
    lcd.clear();
    lcd.write("Temp: ");
    lcd.print(event.temperature);
    lcd.write(" *C");
     delay(1000);

  }
  // Get humidity event and print its value.
  dht.humidity().getEvent(&event);
  if (isnan(event.relative_humidity)) {
    Serial.println("Error reading humidity!");
  }
  else {
    Serial.print("Humidity: ");
    Serial.print(event.relative_humidity);
    Serial.println("%");
    lcd.clear();
    lcd.write("Humidity: ");
    lcd.print(event.relative_humidity);
    lcd.write("%");
     delay(1000);

  }
  /************* DHT CODE ENDS ******************/
  /************* LDR WORKING CODE *******************/
  LDRvalue  =  analogRead(LDRpin);    float  Vout  =  (LDRvalue  *
0.0048828125);
  float RLDR = (10000.0 * (5 - Vout))/Vout;
  float Lux = (500 / RLDR);

  Serial.print("LDR VALUE:   ");
  Serial.println(LDRvalue);  delay(100);
 lcd.clear();
    lcd.write("Light: ");
    lcd.print(Lux);
    lcd.write(" lux");
     delay(1000);

  /******************** END OF LDR ********************/


  /************ MOISTURE SENSOR CODE ****************/

  moistureValue = analogRead(moisturePin);
  Serial.print("MOISTURE VALUE:    ");
  Serial.println(moistureValue);
  delay(100);
  lcd.clear();
  lcd.write("Moisture: ");
  lcd.print(moistureValue);
  lcd.write(" V");
  delay(1000);
if(moistureValue <= threshold_lower)
     {lcd.clear();
      lcd.write("Wet! Leave it.");
        delay(1000);}
if(moistureValue >= threshold_upper)
      {lcd.clear();
        lcd.write("Dry! Water it.");
        delay(1000);}


    /********* END OF MOISTURE SENSOR ***************/



    }
```

CONCLUSION

The Project thereby undertaken has deepend the understanding of the associated course of Embedded Systems and PIC microcontroller. The project is a working prototype of a greenhouse monitoring system. The developed hardware can appropriately measure ambient temperature, humidity, light intensity the system is getting and the soil moisture content. All this information can ease the work of human taking caring of a greenhouse. This gives suitable environment to the plants to grow and a better yield.

This project has been simulated with the PIC18 family controller but the hardware has been built with the arduino Uno board. The LCD interfaced with the system shows the measured parameters.

The project has been successfully completed with good understanding of programming PIC microcontroller.

REFERENCES

1. Automated Greenhouse (2014 IEEE International Symposium on Robotics and Manufacturing Automation)
2. arduino.cc/en/Tutorial/LiquidCrystalSerialDisplay
3. DataSheets : LDR, DTH11, Moisture sensor, PIC182550
4. Forum: MicroElectronika, Arduino Playground