Microprocessor Based Process Control

Temperature and Humidity Module

Made by: Ahmed Tarek Fahmy

Reg. No: 14103030

Presented to: Dr. Walid Ghoneim

Table of Contents

Abstract

In this project we have made a device, which can measure the temperature and humidity. The device contains a temperature sensor, a humidity sensor and an LCD Screen to display the reading in addition to an Arduino Nano to control the whole system. The system in implemented in a cuboid on Acrylic (11cm * 8cm * 4cm). The system is compact in size so that it can be placed anywhere home, office, school, library… etc. The system can also measure the maximum and average temperature in addition to measuring the maximum and average humidity.

Device layout:

The device consists of Am 2301, LCD Screen, Arduino Nano, as shown in figure (1).

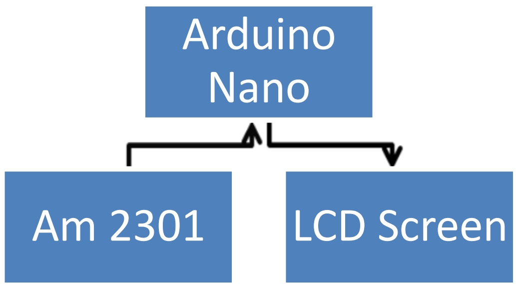


Figure 1: System Block Diagram

The function of each component is shown below:

- Am 2301: Temperature and Humidity module, to measure the temperature and the relative humidity.
- LCD Screen: Liquid Crystal Display, to show the reading if the system
- Arduino Nano: Read the temperature from the Temperature and Humidity module and then display them on the LCD Screen.

Operation Principles of Device Components:

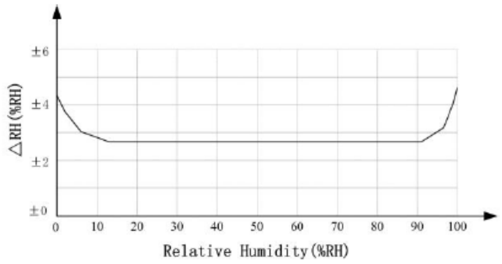
Temperature and Humidity Module (Am2301)

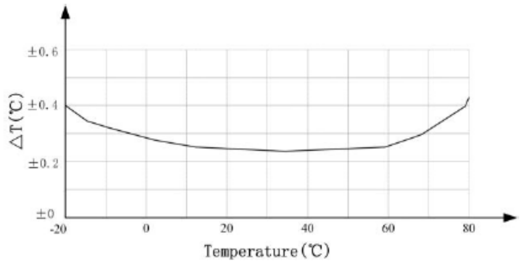
The module includes a capacitive sensor wet components and a high-precision temperature measurement device, and connected with a high-performance 8-bit microcontroller. Each sensor is extremely accurate humidity calibration chamber calibration. The module has small size, low power consumption, signal transmission distance up to 20 meters.

Sensor Performance:

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
Range		0		99.9	%RH
Accuracy [1]	25℃		±3		%RH
Repeatability			±1		%RH
Exchange		Completely interchangeable			
Response [2]	1/e(63%)		<6		S
Sluggish			±0.3		%RH
Drift [3]	Typical		<0.5		%RH/yr

Parameter	Condition	min	typ	max	Unit
Resolutio			0.1		℃
n			16		bit
Accuracy			±0.3	±1	℃
Range		−40		80	℃
Repeat			±0.2		℃
Exchange		Completely interchangeable			
Response	1/e(63%)		<10		S
Drift			±0.3		℃/yr





Electrical Characteristics:

Parameter	Condition	min	typ	max	Unit
Voltage		3.3	5	5.2	V
Power consumption [4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low level output voltage	I_{OL} [5]	0		300	mV
High output voltage	Rp<25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High Voltage	Rise	70%		100%	VDD
Rpu [6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	turn on		8		mA
	turn off	10	20		μA
Sampling period		2			S

[1] the accuracy of the factory inspection, the sensor 25 ° C and 5V, the accuracy specification of test conditions, it does not include hysteresis and nonlinearity, and is only suitable for non−condensing environment.

[2] to achieve an order of 63% of the time required under the conditions of 25 ℃ and 1m / s airflow.

[3] in the volatile organic compounds, the values may be higher. See the manual application to store information.

[4] this value at VDD = 5.0V when the temperature is 25 ℃, 2S / time, under the conditions of the average.

[5] low output current.

[6] that the pull−up resistor.

Single-bus communication protocol

- Single bus Description

AM2301 device uses a simplified single-bus communication. Single bus that only one data line, data exchange system, controlled by the data line to complete. Equipment (microprocessor) through an open-drain or tri-state port connected to the data line to allow the device does not send data to release the bus, while other devices use the bus; single bus usually require an external about 5.1kΩ pull-up resistor, so when the bus is idle, its status is high. Because they are the master-slave structure, only the host calls the sensor, the sensor will answer, so the hosts to access the sensor must strictly follow the sequence of single bus, if there is a sequence of confusion, the sensor will not respond to the host.

- Single bus to send data definition

SDA For communication and synchronization between the microprocessor and the AM2301, single-bus data format, a transmission of 40 data, the high first-out. Specific communication timing is shown in figure (2), the comm. format is depicted in Table 1.

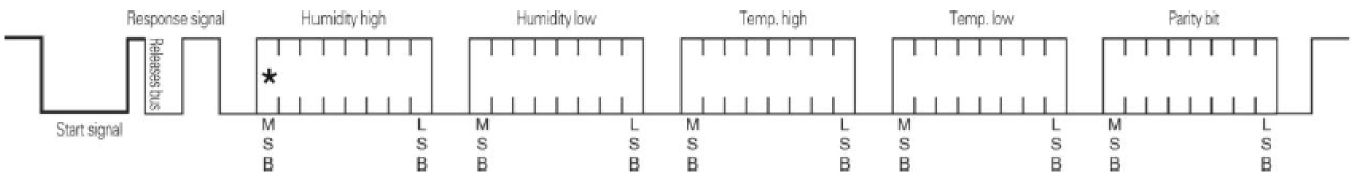


Figure 2: AM2301 Single-bus communication protocol

Table 1: AM2301 Communication format specifier

Name	Single−bus format definition
Start signal	Microprocessor data bus (SDA) to bring down a period of time (at least 800μ s) [1] notify the sensor to prepare the data.
Response signal	Sensor data bus (SDA) is pulled down to 80μ s, followed by high−80μ s response to host the start signal.
Data format	Host the start signal is received, the sensor one−time string from the data bus (SDA) 40 data, the high first−out.
Humidity	Humidity resolution of 16Bit, the previous high; humidity sensor string value is 10 times the actual humidity values.
Temp.	Temperature resolution of 16Bit, the previous high; temperature sensor string value is 10 times the actual temperature value; The temperature is the highest bit (Bit15) is equal to 1 indicates a negative temperature, the temperature is the highest bit (Bit15) is equal to 0 indicates a positive temperature; Temperature in addition to the most significant bit (Bit14 ~ bit 0) temperature values.
Parity bit	Parity bit = humidity high + humidity low + temperature high + temperature low

Single-bus communication timing

User host (MCU) to send a start signal (data bus SDA line low for at least 800μs) after AM2301 from Sleep mode conversion to high-speed mode. The host began to signal the end of the AM2301 send a response signal sent from the data bus SDA serial 40Bit's data, sends the byte high; data sent is followed by: Humidity high、 Humidity low、 Temperature high、Temperature low、Parity bit, Send data to the end of trigger information collection, the collection end of the sensor is automatically transferred to the sleep mode, the advent until the next communication. Single-bus communication timing diagram is shown in figure (2); detailed timing signals characteristics are shown in Table 2.

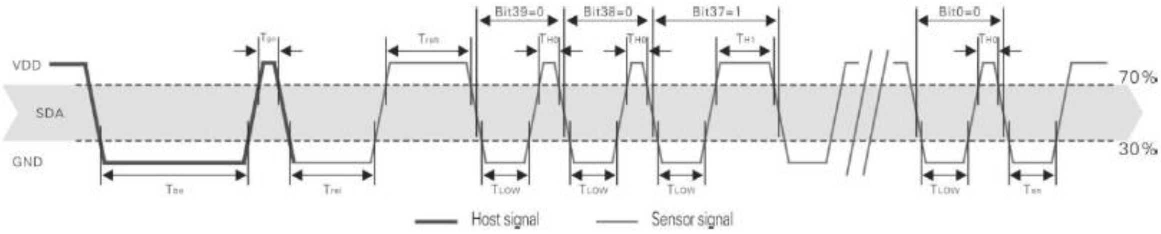


Figure 3: Single-bus communication timing diagram

Table 2: Detailed timing signals characteristics

Symbol	Parameter	min	typ	max	Unit
T_{be}	Host the start signal down time	0.8	1	20	mS
T_{go}	Bus master has released time	20	30	200	μS
T_{rel}	Response to low time	75	80	85	μS
T_{reh}	In response to high time	75	80	85	μS
T_{LOW}	Signal "0", "1" low time	48	50	55	μS
T_{H0}	Signal "0" high time	22	26	30	μS
T_{H1}	Signal "1" high time	68	70	75	μS
T_{en}	Sensor to release the bus time	45	50	55	μS

LCD Display

The Display used is 16×2 Liquid Crystal Display which will display 32 characters at a time in two rows (16 characters in one row). Each character in the display of size 5×8 pixel matrix, there are 16 pins in the LCD module, the pin configuration is shown in table (3).

Table 3: Pin configuration of LCD Screen

Pin no.	Symbol	External connection	Function
1	Vss	Power supply	Signal ground for LCM
2	V_{DD}		Power supply for logic for LCM
3	V_0		Contrast adjust
4	RS	MPU	Register select signal
5	R/W	MPU	Read/write select signal
6	E	MPU	Operation (data read/write) enable signal
7~10	DB0~DB3	MPU	Four low order bi-directional three-state data bus lines. Used for data transfer between the MPU and the LCM. These four are not used during 4-bit operation.
11~14	DB4~DB7	MPU	Four high order bi-directional three-state data bus lines. Used for data transfer between the MPU
15	LED+	LED BKL power supply	Power supply for BKL
16	LED-		Power supply for BKL

Communicating with LCD Display

In order to communicate with the LCD display you should specify first what is needed to be sent, as the LCD can receive commands and data, as you need to send to the LCD some commands to configure it in the right way then send the data needed to be displayed. In order to configure this we use pin (4) “Register select signal” where:

- **Command Register**
  - ❖ When we send commands to LCD it goes to Command register
  - ❖ When RS=0 Command Register is Selected.
- **Data Register**
  - ❖ When we send Data to LCD it goes to data register.
  - ❖ When RS=1 Data Register is selected.

We can configure pin (5) “Read and write”, where:

- When RW=1 we want to read data from LCD.
- When RW=0 we want to write to LCD.
- In Our Project we connected it permanently to ground as we only want to write to the LCD

We can configure pin (6) “Enable”, to give voltage push to execute the instruction. Usually we make it en=0 and when we want to execute the instruction we make it high en=1 for some milliseconds. After this we again make it ground en=0.

List of LCD Commands

1. The command 0x30 means we are setting 8-bit mode LCD having one lines and character shape between 5x7 matrix.
2. The command 0x38 means we are setting 8-bit mode LCD having two lines and character shape between 5x7 matrix.
3. The command 0x20 means we are setting 4-bit mode LCD having 1 line and character shape between 5x7 matrix.
4. The command 0x28 means we are setting 4-bit mode LCD having 2 lines and character shape between 5x7 matrix.
5. The command 0x06 is entry mode it tells the LCD that we are going to use.
6. The command 0x08 displays cursor off and display off but without clearing DDRAM contents.
7. The command 0x0E displays cursor on and display on.
8. The command 0x0c display on cursor off(displays cursor off but the text will appear on LCD)

9. The command 0x0F display on cursor blink (text will appear on screen and cursor will blink).
10. The command 0x18 shift entire display left (shift whole off the text on the particular line to its left).
11. The command 0x1C shift entire display right (shift whole off the text on the particular line to its right).
12. The command 0x10 Moves cursor one step left or move cursor on step ahead to left whenever new character is displayed on the screen.
13. The command 0x14 Moves cursor one step right or move cursor on step ahead to right whenever new character is displayed on the screen.
14. The command 0x01 clears all the contents of the DDRAM and also clear the LCD removes all the text from the screen.
15. The command 0x80 initialize the cursor to the first position means first line first matrix (start point) now if we add 1 in 0x80+1=0x81 the cursor moves to second matrix.

Operating LCD

- For Initializing LCD:
  - 0x38 is used for 8-bit data initialization.
  - 0xFH for making LCD on and initializing the cursor.
  - 0X6H for incrementing the cursor which will help to display another character in the LCD
  - 0x1H for clearing the LCD.

- For Sending Data to the LCD:
  - E=1; enable pin should be high
  - RS=1; Register select should be high for writing the data
  - Placing the data on the data registers
  - R/W=0; Read/Write pin should be low for writing the data.

Device Hardware

Electrical Configuration

The device consists of Am 23-01, LCD Screen, Arduino Nano. The Am 2301 has 3 wires the first one connected to 5v and the second one is connected to the digital pin 6 (D6) in the Arduino Nano, while the third one is connected to ground. The LCD has 16 pins, 3 of them are connected to ground (VSS, R/W, LED-), 2 of them connected to 5V (Vcc, LED+ “Through a 220 Ω resistance”), 4 of them are unconnected (DB0, DB1, DB2, DB3), 6 are connected to the digital pins of the Arduino Nano (DB4, DB5, DB6, DB7, RS, E) and the pin (V_o) is connected to a potentiometer while both ends of the potentiometer are connected to 5v and ground. The detailed circuit layout is shown in figure (4).

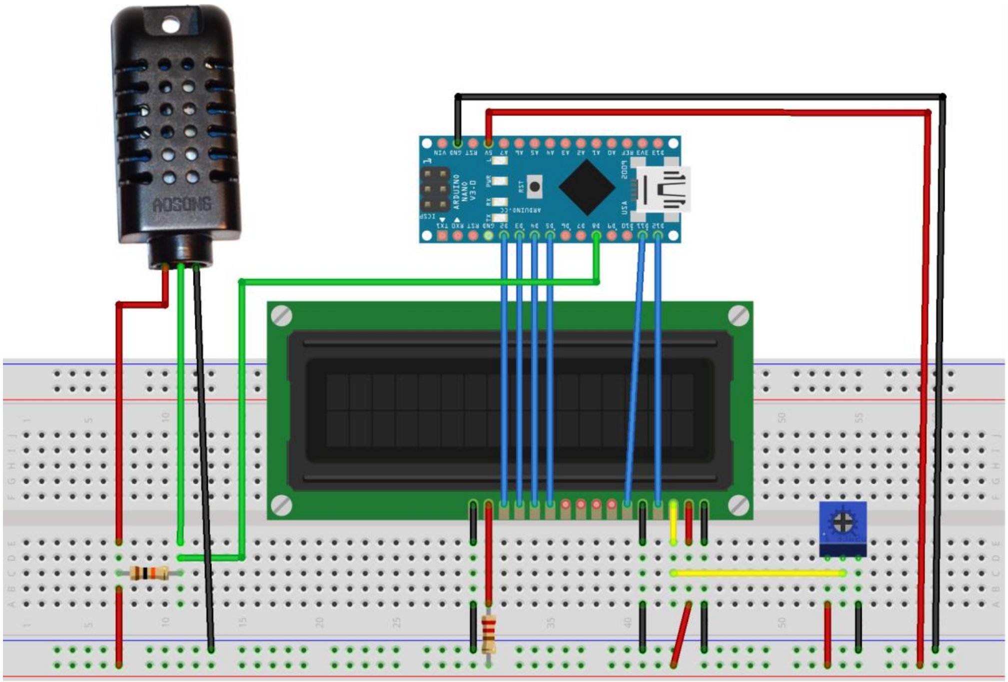


Figure 4: Circuit Layout

The circuit was implemented on a printed circuit board. The design was made especially for this project using Fritzing Program as shown in figure (5).

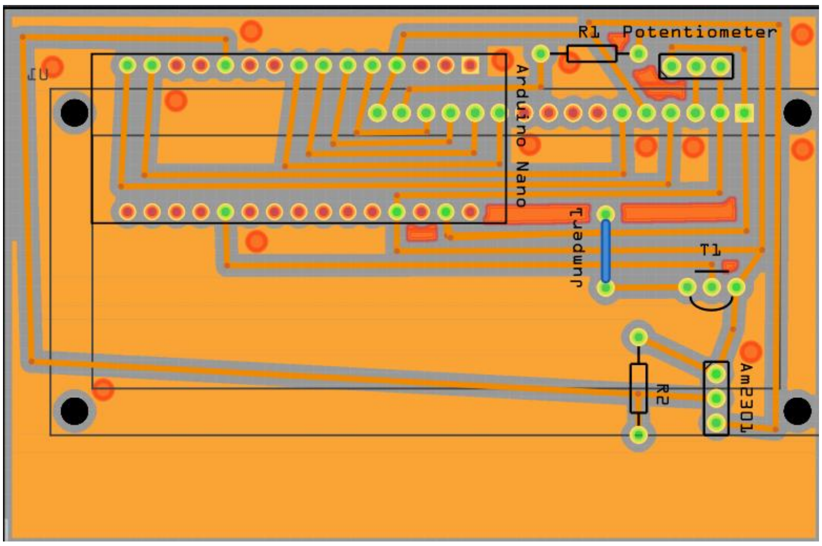


Figure 5: Circuit Schematic

All the components were put on the top layer while only the LCD screen was put on the bottom layer, as shown in figure (6).

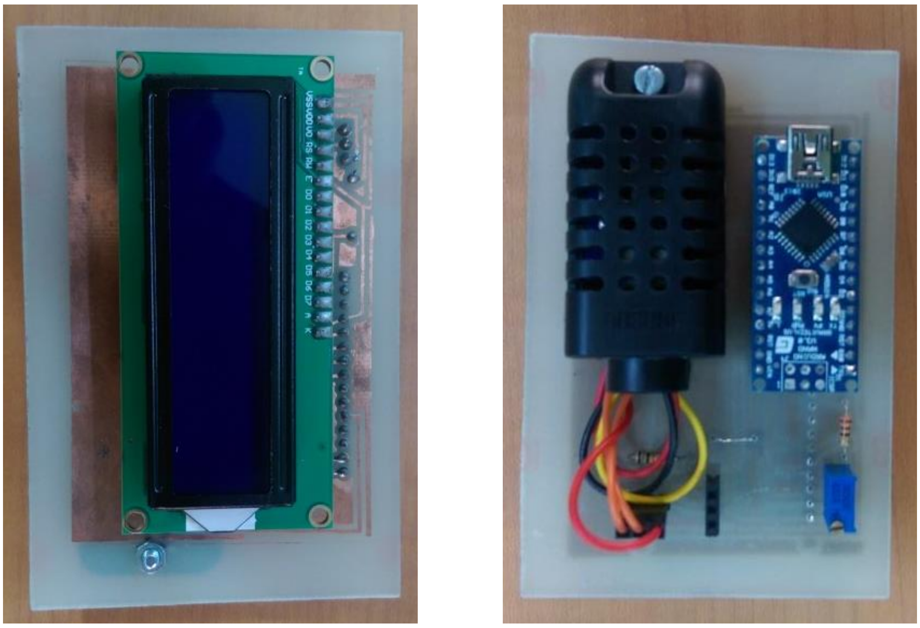


Figure 6: Printed Circuit Board

Mechanical Configuration

The circuit was placed inside an acrylic box with length: 11 cm, Width: 8 cm, Height: 4 cm. The box was specially designed for the project using AutoCAD as shown in figure (7).

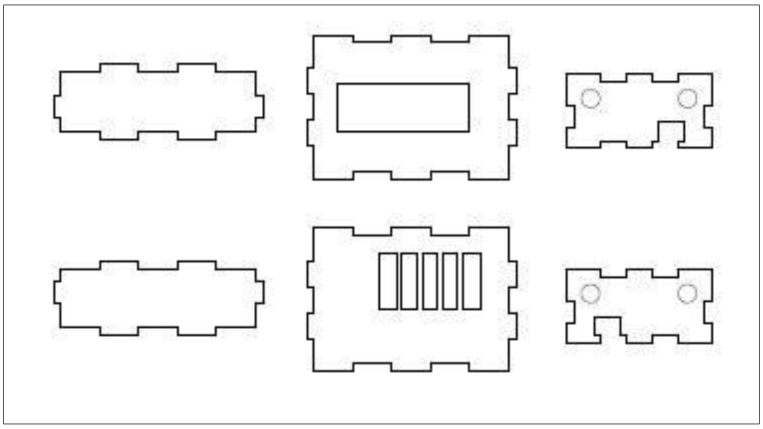


Figure 7: Box Design

The box has an opening suitable for a micro USB Cable to power and the device. Also it has 4 holes for ventilation in addition to holes above the Am 2301 to be able to sense the atmosphere. The final image of the device is shown in figure (8):

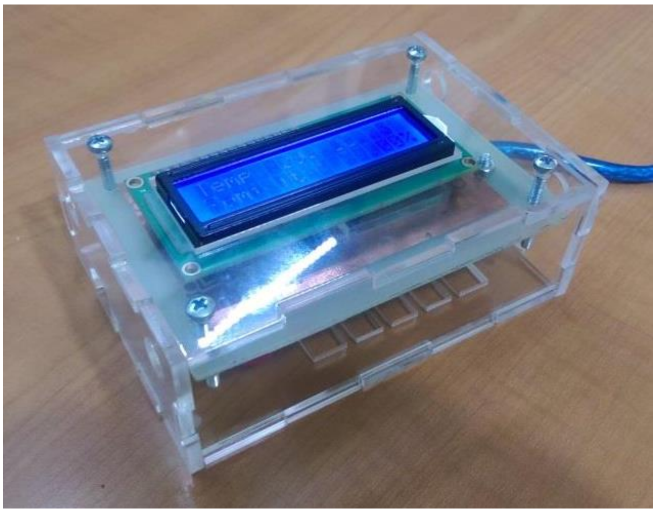


Figure 8: Final Image of device

Device Software:

A Program was implemented inside the Arduino Nano to control this device this program aims to:

- Read Temperature and relative humidity form Am2301
- Calculate the average and the maximum temperature
- Calculate the average and the maximum relative humidity
- Display all the results on the LCD Screen

The flow chart indicating the details of the software in shown in figure (9), the detailed code is attached in Appendix I.

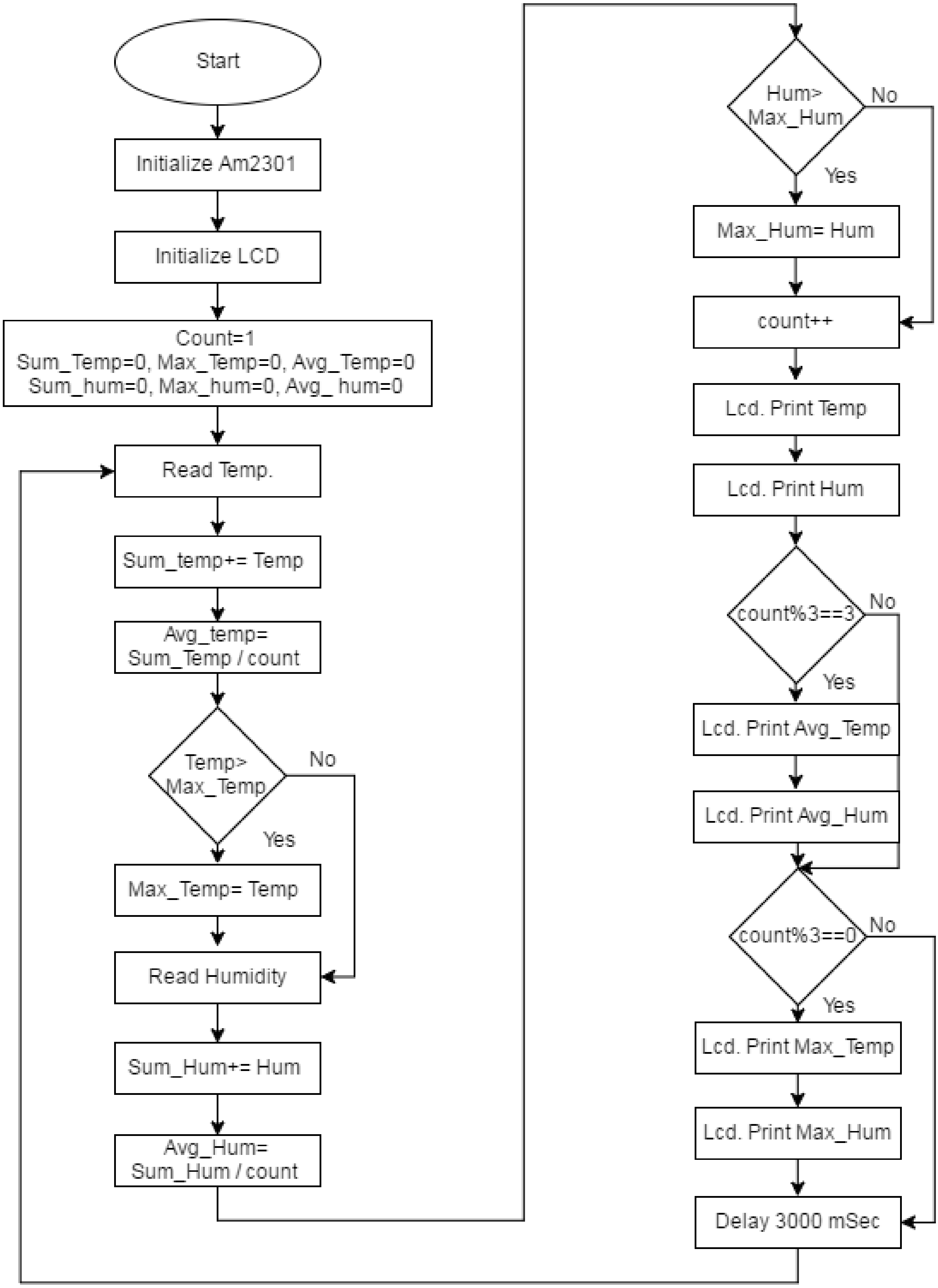


Figure 9: Flow Chart of the Software

Appendix I (Detailed code of the device):

```
#include <LiquidCrystal.h>
#include <SPI.h>
#include "DHT.h"

#define DHTPIN 8     // what digital pin we're connected to

#define DHTTYPE DHT21   // DHT 21 (AM2301)

DHT dht(DHTPIN, DHTTYPE);

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
float sum_hum=0, max_hum=0, avg_hum ;
float sum_t=0, max_t=0, avg_t ;

int  count=1;

void setup() {
 lcd.begin(16, 2);
 Serial.begin(9600);
 Serial.println("Intializing!");
 lcd.setCursor(0, 0);
 lcd.print("Intializing!");

 dht.begin();
 }

void loop() {

 float h = dht.readHumidity();
 // Read temperature as Celsius (the default)
 float t = dht.readTemperature();


 // Check if any reads failed and exit early (to try again).
 if (isnan(h) || isnan(t) ) {
  Serial.println("Failed to read from DHT sensor!");
  return;
 }

 sum_t+=t;
 avg_t=sum_t/(count);

 if(t>max_t)
 max_t=t;

 sum_hum+=h;
 avg_hum=sum_hum/(count);

 if(h>max_hum)
 max_hum=h;

 count++;
```

```
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp: ");
  lcd.print(t);
  lcd.print(" C");

  lcd.setCursor(0, 1);
  lcd.print("Hum : ");
  lcd.print(h);
  lcd.print(" %");

if (count%4==3)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Temp Avg: ");
  lcd.print(avg_t);

  lcd.setCursor(0, 1);
  lcd.print("Temp Max: ");
  lcd.print(max_t);
}

if (count%4==0)
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Hum Avg: ");
  lcd.print(avg_hum);

  lcd.setCursor(0, 1);
  lcd.print("Hum Max: ");
  lcd.print(max_hum);
}

delay(3000);
}
```

Appendix II (Cost Analysis):

Item	Price
Am2301	70 EGP
LCD Display	35 EGP
Arduino Nano	70 EGP
PCB	15EGP
Acrylic Box	28 EGP
Total	**218 EGP**