

LEVEL 4 ASSESSMENT SPECIFICATION

Student name:	Ben Hathaway	Student ID number:	1810525
Programme:			
Module:	Embedded Systems		
Module code:	PCEE4004	Contribution to Overall Module Assessment (%):	100%
Lecturer:	Sarah Williams	Internal Verifier:	
Assignment Title:	Office Environment Monitor	Word count (or equivalent):	2000
Submission deadline:	10th May 2019	Return date of provisional marks & written feedback:	24th May 2019
Submission method:	All written assessments, where practical and possible, must be submitted via Turnitin unless otherwise instructed by the Lecturer. (Please DO NOT put this assessment specification into Turnitin or it will match many similarities with other students' submissions.) Alternative submission method (if applicable): Late submission of the assessment will result in a late penalty mark. Penalties for late submission: Up to one week late, maximum mark of 40%. Over one week late, 0%. Only the Extenuating Circumstances Panel may approve a change to submission dates.		
Academic honesty / referencing:	Academic honesty is required. In the main body of your submission you must give credit to authors on whose research and ideas your work is based. Append to your submission a reference list that indicates the books, articles, etc. that you have used, cited or quoted in order to complete this assessment.		

STUDENT FEEDBACK FORM

This section details the *extent* to which the assessment criteria are demonstrated by you, which in turn determines your mark. The marks available for each category of skill are shown. Lecturers will use the space provided to comment on the achievement of the task(s), including those areas in which you have performed well and areas that would benefit from development/improvement.

Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills		
Evidence of wide range of independently sourced research provided.	10	6
2. Knowledge and Understanding Skills		
Evidence of planning and justification of decisions made for the components used. Hardware and software design provided The flowchart only considers the “loop” section of the program and does not consider any initialisation of the components/pins, also flow implies that it “restarts”, whilst the program will sit in a loop, the system does not reboot. Boundary conditions are also a concern, what happens at 16°C and 24 °C? The flowchart and circuit diagrams are created adhering to standard conventions and demonstrate good understanding of the topic area.	20	10
3. Cognitive and Intellectual Skills		
Evidence is provided to show that the system has been tested effectively, although you should include in the testing table what the system “should” do, so that it can be confirmed that the test has actually been successful or not. Project was demonstrated as working as required in class. Evaluation discussion provides valid and justified ways in which the system might be further improved.	25	18
4. Practical Application Skills		
Basic circuit however it effectively allows for the user to identify when the temperature is within acceptable ranges, and when outside. Code is well commented and uses constants which is good practice, however there is no use of functions. A good description of the code is included in the report. Code is simple, and differs slightly from flowchart provided at design stage, however this is beneficial deviation due to the flaws in the original flowchart. Overall the project demonstrates a very good deployment of both practical programming and electrical circuit design and construction.	35	20

5. Transferable Skills for Life and Professional Practice		
Work is well formatted and presented, however you ensure it is proof read before submission.	10	6

Assessment Mark (Assessment marks are subject to ratification at the Exam Board. These comments and marks are to give feedback on module work and are for guidance only until they are confirmed.)	Late Submission Penalties (tick if appropriate)		60%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

GENERIC ASSESSMENT CRITERIA

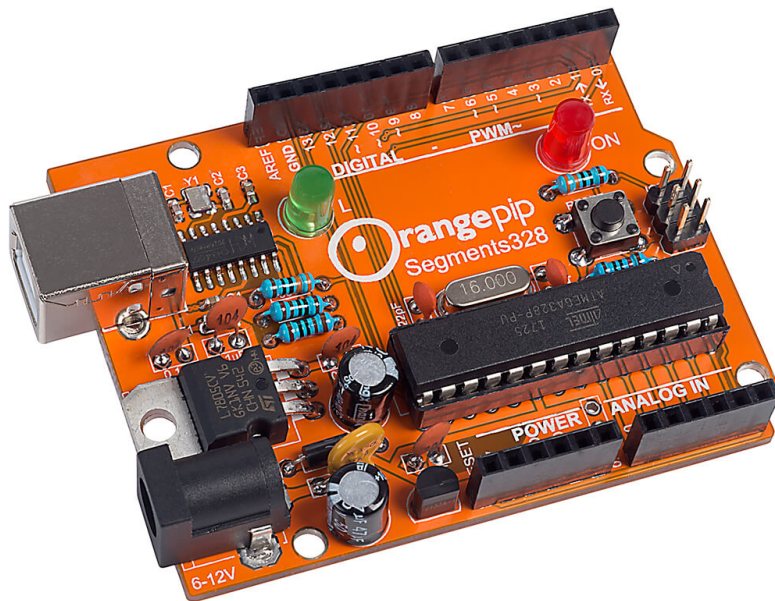
Level 4

In accordance with the Framework for Higher Education Qualifications, at the end of Level 4 students will be expected to have demonstrated knowledge of the basic underlying concepts and principles of a subject, and an ability to evaluate and interpret these within the context of that area of study. They should be able to present, evaluate and interpret qualitative and quantitative data in order to develop lines of argument and make sound judgements in accordance with basic theories and concepts of their subject(s) of study. They will have learned how to evaluate different approaches to solving problems, and will be able to communicate the results of their study/work accurately and reliably, and with structured and coherent arguments. They will be able to undertake further training and develop new skills within a structured and managed environment and will have the qualities and transferable skills necessary for employment requiring the exercise of some personal responsibility.

Level 4	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)	EXCEPTIONAL (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-84%	85-100%
Engagement with literature (including reading, referencing, academic conventions and academic honesty)	Little or no evidence of reading and/or reliance on inappropriate sources. Views and findings mostly unsupported and non-authoritative. Referencing conventions used incoherently or largely absent.	Poor engagement with essential reading. No evidence of wider reading. Reliance on inappropriate sources, and/or indiscriminate use of sources. Heavily reliant on information gained through class contact. Inconsistent and weak use of referencing.	Engagement with a limited range of mostly relevant and credible sources but with some reliance on information gained through class contact. Some omissions and minor errors. Referencing conventions evident though not always applied accurately or consistently.	Engagement with an appropriate range of literature, including sources retrieved independently. Some over-reliance on texts rather than other sources. Referencing may show minor inaccuracies or inconsistencies.	Engagement with a wide range of literature, including sources retrieved independently. Selection of relevant and credible sources. Generally sound referencing, with no/very few inaccuracies or inconsistencies.	Engagement with an extensive range of relevant and credible literature. Consistently accurate application of referencing.	Exceptional engagement with an extensive range of relevant and credible literature. High-level referencing skills consistently applied.
Knowledge and understanding (Knowledge of the basic underlying concepts and principles of a subject.)	Major gaps in knowledge and understanding of the basic underlying concepts and principles of the subject matter. Inclusion of irrelevant material. Substantial inaccuracies.	Gaps in knowledge of the basic underlying concepts and principles, with flawed or superficial understanding. Some significant inaccuracies and/or irrelevant material.	Limited knowledge and understanding of the basic underlying concepts and principles within the subject area. Some elements may be missing.	Knowledge of the basic underlying concepts and principles is accurate with a good understanding of the field of study but lacks depth and/or breadth.	Competent knowledge of the basic underlying concepts and principles. Exhibits very good understanding.	Excellent knowledge and understanding of the basic underlying concepts and principles of the subject.	Exceptional, detailed knowledge and understanding of the basic underlying concepts and principles

Level 4	FAIL	MARGINAL FAIL	SATISFACTORY (3 rd / Pass)	GOOD (2.2 / Pass)	VERY GOOD (2.1 / Merit)	EXCELLENT (1 st / Distinction)	EXCEPTIONAL (1 st / Distinction)
Category	0-29%	30-39%	40-49%	50-59%	60-69%	70-84%	85-100%
Cognitive and intellectual skills (Evaluate underlying concepts and principles of a subject and interpret qualitative and quantitative data in order to develop lines of argument and make sound judgements.)	Wholly or almost wholly descriptive work. Little or no evaluation of the underlying concepts and principles. Failure to develop arguments, leading to illogical or invalid judgements. Minimal or no use of evidence to back up views.	Largely descriptive work, with superficial evaluation of the underlying concepts and principles. Weak interpretation of data, flawed development of arguments and judgements. Information accepted uncritically, uses generalised statements made with scant evidence and unsubstantiated opinions. Ideas sometimes illogical and contradictory.	Limited attempt at evaluation of the underlying concepts and principles, tending towards description. Can interpret qualitative and quantitative data but with some errors. Some evidence to support emerging arguments and judgements but these may be underdeveloped or with a little inconsistency / mis-interpretation.	Good evaluation of the underlying concepts and principles. Can interpret qualitative and quantitative data, with minor errors. An emerging ability to use evidence to support the argument. Mostly valid arguments and logical judgements.	Sound evaluation of the underlying concepts and principles. Can interpret qualitative and quantitative data accurately. Ability to devise arguments using evidence to make mostly appropriate and valid judgements.	Excellent evaluation of the underlying concepts and principles. Can interpret qualitative and quantitative data accurately and with some insight. Excellent ability to devise arguments using evidence and make appropriate and valid judgements.	Exceptional evaluation of the underlying concepts and principles based evidence. Outstanding interpretation of qualitative and quantitative data. Exceptional ability to devise arguments using evidence and make wholly appropriate and valid judgements.
Practical skills (Different approaches to solving problems in particular contexts.)	Limited or no use of taught, basic methods, materials, tools and/or techniques. Little or no appreciation of the context of the application. Very weak evidence of different approaches to problem-solving in particular contexts.	Rudimentary application of taught, basic methods, materials, tools and/or techniques but without consideration and competence. Flawed appreciation of the context of the application. Weak evidence of different approaches to problem-solving in particular contexts.	An adequate awareness and mostly appropriate application of basic methods, materials, tools and/or techniques. Basic appreciation of the context of the application. Can identify problems in particular contexts and propose basic alternative approaches or solutions though there may be errors.	A good and appropriate application of basic methods, materials, tools and/or techniques. Clear appreciation of the context of the application. Good evidence of different approaches to problem-solving in particular contexts and proposes mostly appropriate solutions.	A very good application of a range of basic methods, materials, tools and/or techniques. Very good consideration of the context of the application. Very good evidence of different approaches to problem-solving in particular contexts and proposes appropriate solutions.	An advanced application of a range of taught, basic methods, materials, tools and/or techniques. The context of the application is well considered, and insightful. Excellent evidence of different approaches to problem-solving in particular contexts and proposes appropriate solutions.	Exceptional levels of application and deployment skills in particular practical contexts. Outstanding identification of problems in particular contexts and formulation of wholly appropriate, thoughtful solutions / different approaches.

Transferable skills for life and professional practice (Communicate the results of their study/work accurately and reliably, and with structured and coherent arguments; the qualities needed for employment requiring the exercise of some personal responsibility within a structured and managed environment.)	Work is poorly structured, disorganised, inaccurate and/or confusingly expressed. Very weak use of language and/or very inappropriate style. Failure to work effectively individual or as part of a group. Little or no evidence of the skills for employment requiring the exercise of some personal responsibility.	Work is poorly presented in a disjointed manner. It is loosely, and at times incoherently, structured, with information and ideas often poorly expressed. Weak use of language and/or inappropriate style. Flawed approach to individual or group work, meeting only partial obligations to others. Limited evidence of the skills for employment requiring the exercise of some personal responsibility.	Mostly ordered presentation and structure in which relevant ideas / concepts are reasonably expressed. Work may lack coherence and/or accuracy in places. Can work as part of a group, meeting most obligations to others but perhaps with limited involvement in group activities. Demonstrates the basic skills for employment requiring the exercise of some personal responsibility, with some areas of minor weakness.	Mostly coherent, organised and accurate work, in a suitable structure and is for the most part clearly expressed. Can work effectively independently and/or as part of a team, with clear contribution to group activities. Demonstrates the skills for employment requiring the exercise of some personal responsibility, with some areas of strength and some of minor weakness.	Work is accurate, coherent, fluent, well-structured and organised. Can work effectively independently and/or as part of a team, with very good contribution to group activities. Demonstrates very good skills for employment requiring the exercise of some personal responsibility, with just occasional minor weakness.	Work is coherent, very fluent and is presented professionally. Can work effectively independently and/or as part of a team, with an excellent contribution to group activities. Demonstrates excellent skills for employment requiring the exercise of some personal responsibility with an appetite for further development.	Work is accurate, exceptionally coherent, very fluent and presented well. Can work effectively independently and/or as part of a team, with an exceptional contribution to group activities. Demonstrates exceptional skills for employment requiring the exercise of some personal responsibility with an appetite for further development.
---	--	--	--	---	---	--	---



EMBEDDED SYSTEMS

PCEE4004

Ben Hathaway

Ben Hathaway

Contents page

Introduction – Page 2

Flow chart – Page 3

Hardware Design – Page 4

Breadboard design of circuit – Page 5

Schematic of circuit – Page 6

Software Design – Page 7

Brief explanation of code – Page 8

Testing – Pages 9, 10

Testing images – Pages 11, 12, 13, 14

Evaluation and improvements - Page 15

Appendix – Pages 16 – 27

References - 28

Biblogpraghy - 29

Introduction

The purpose of the system I have produced is to give indication to someone working in an office based environment if the room temperature is too hot or too cold as well as being within a comfortable working range.

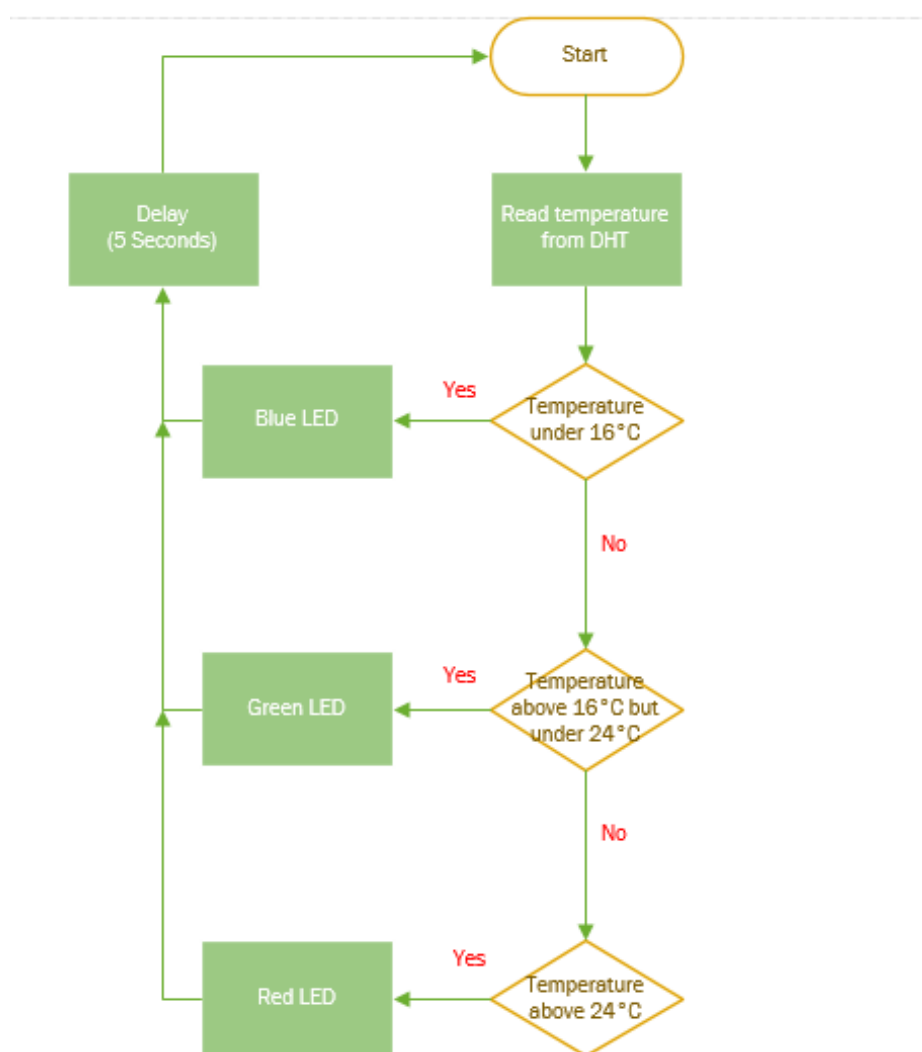
From research on the HSE website the figures I have decided to use are 16°C for the minimum temperatures as this is what it states under the law section of the page " The law does not state a minimum or maximum temperature, but the temperature in workrooms should normally be at least 16°C".

A maximum figure cannot be given due to the differences in work environments and conditions and the fact that people can work in very high temperatures as long as there are suitable safety procedures in places, based on this I set my higher boundary at 24°C as I believe it would be uncomfortable to work in that temperature all day.

I plan to implement this by using a microcontroller based system and using c programming language to create a successful code to work the microcontroller system. The microcontroller being using to complete this task is an orange pip Kona 328 board in addition to a breadboard to connect all the components and power them correctly. The programme being used to create the code is the Arduino IDE and this will then be loaded onto the microcontroller and then put through a testing procedure to ensure it works correctly.

Flow Chart

Below is the image of a flow chart I created to help me plan the hardware and software design. It starts when the board is powered; it then reads the temperature from the DHT component. This component reads the temperature relaying the reading back to the code, and then as the flow chart shows if the temperature is under 16°C the tri coloured LED will turn blue before waiting five seconds and then taking the temperature again. If the temperature is above 16°C and under 24°C the tri coloured LED will turn green and then wait five seconds before reading the temperature again and finally, if the temperature is over 24°C the tri coloured LED will turn red before a five-second delay and the programme starting again.



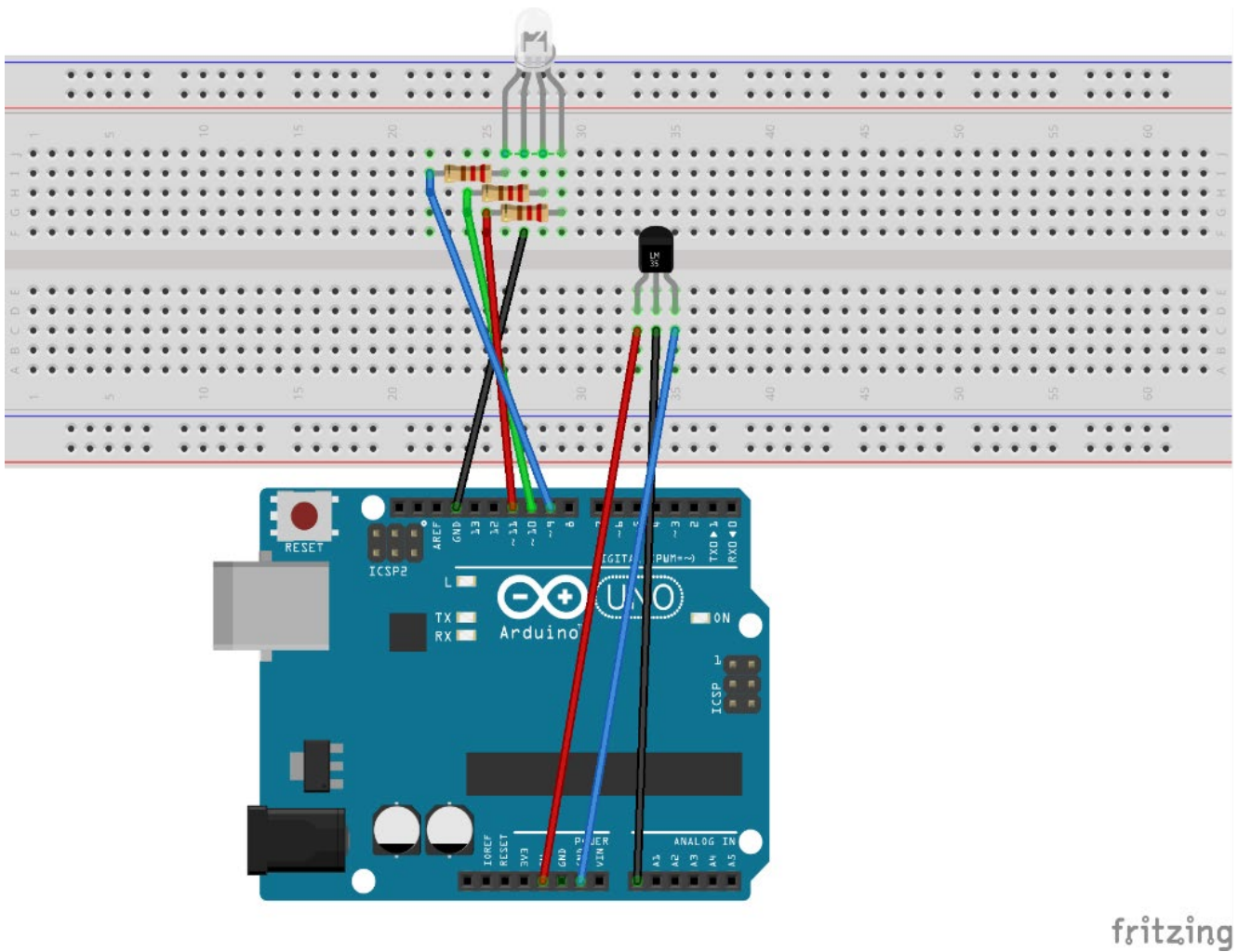
Hardware Design

The circuit I have created includes an orange pi Kona328 board, 7 different wires, a DHT temperature and humidity sensor, three 220-ohm resistors and a tri colour light emitting diode. The circuit is going to work with the DHT reading the temperature within a certain environment and reading this back to the code I have written.

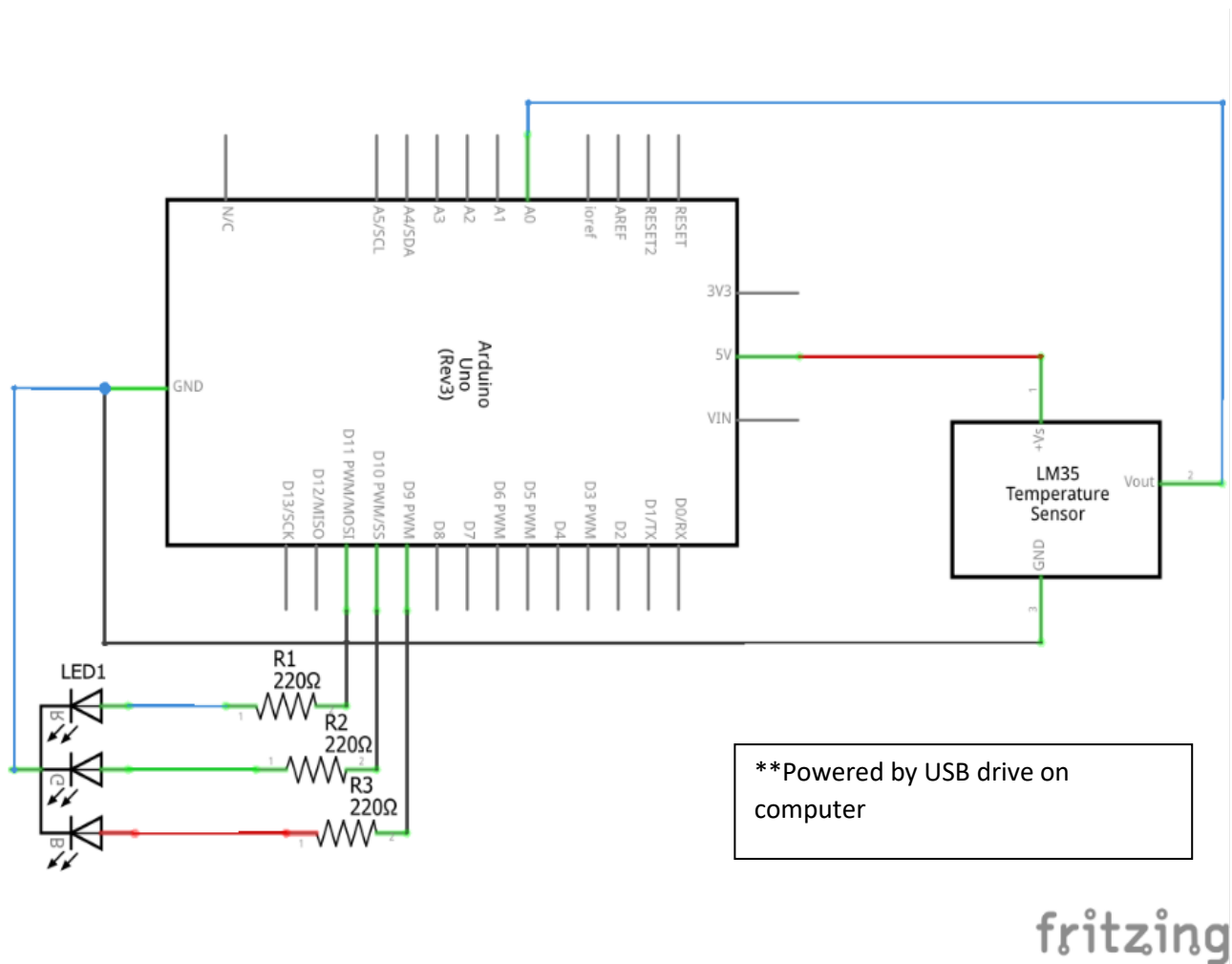
If the temperature is under 16°C the tri coloured LED will turn blue. After a length of time, the DHT will take another reading of the temperature if it is over 24°C the tri coloured LED will change to a red colour. If the temperature is within 16 – 24°C the LED will be green. I choose to use these colours as it gives a good indication of what the temperature is doing as, blue is associated with colder weather, red is associated with warmer weather and the green gives a good indication that the temperature is within a good range.

The three 220 ohm resistors that are used in the circuit to limit the current flowing to the tri coloured LED to a safe amount. The reason I have used this size resistor as it is in the manufacturer's guidebook and is a value that is commonly used with light emitting diodes. I used the colour code system to ensure I had picked the correct resistors and I used a multimeter to prove that the resistors were within the correct tolerances, which is +/- 10%.

Breadboard diagram of hardware design



Schematic of hardware design



Software Design

Below is a print screen of the code I produced to take a reading from the DHT changing the tricoloured LED to give an indication of the temperature within a work environment. The print screens includes general comments explaining what functions are doing.

```
* Untitled x

#include "dht.h" //This includes the DHT library and means it can be used on the arduino programme
#define dht_pin A0 // Analog Pin sensor is connected to relays the temperature reading

const int Rled = 11; // Red LED powered from pin 11 on the Orangepip board
const int Gled = 10; // Green LED powered from pin 10 on the Orangepip board
const int Bled = 9; // Blue LED powered from pin 9 on the Ornagepip board
const int TEMPERATURE = 0;
const int MIN_TEMP = 16; //This is the minimum temp, Blue LED comes on when below this
const int MAX_TEMP = 24; // This is the maximum temp, Red Led come on when above this
int val = 0;
int temp = 0;
dht DHT;

void setup(){

  Serial.begin(9600);
  delay(500); //Delay to let system boot
  Serial.println("DHT11 Humidity & temperature Sensor\n\n");
  delay(1000); //Wait before accessing Sensor so the LED has time to change before the temp chnages again

  pinMode(Rled, OUTPUT); //Red colour of tri coloured LED is an output
  pinMode(Gled, OUTPUT); //Green colour of tri coloured LED is an output
  pinMode(Bled, OUTPUT); //Blue colour of tri coloured LED is an output
}

void loop(){

  DHT.read11(dht_pin);

  Serial.print("Current humidity = ");
  Serial.print(DHT.humidity);
  Serial.print("% ");
  temp=DHT.temperature; // Take the temperature reading off the DHT
  Serial.print("temperature = ");
  Serial.print(temp);
  Serial.println("C "); // Serial prints the readings so they can be seen on screen, used for testing purposes
  delay(5000); // Delay before taking temperature again giving the LED a chnace to chnage from previous reading

  if (temp >=MAX_TEMP) // If temperature is above the max (24) the LED will be RED
  {
    analogWrite(Rled, 255); //Sends a voltage to RED led pin as temp is above maximum
    analogWrite(Gled, 0); // No voltage to green pin as condidion not met
    analogWrite(Bled, 0); // No voltage to blue pin as condition not met
  }

  else if (temp > MIN_TEMP && temp < MAX_TEMP) // If temperature above min value (16) and below max (24) LED will be green
  {
    analogWrite(Rled, 0); // No voltage to Red pin as condition not met
    analogWrite(Gled, 255); //Send a voltage to Green LED pin as temp is above the minimum and below the maximum
    analogWrite(Bled, 0); // No voltage to Blue pin as condition not met
  }
  else // If temperature is below the min the LED will be blue
  {
    analogWrite(Rled, 0); // No voltage to red pin as condition not met
    analogWrite(Gled, 0); // No voltage to green pin as condition not met
    analogWrite(Bled, 255); // Voltage send to blue pin as temperature is under maximum
  }
}

File 'Fritz ldr.ino' was restored from the .fzz file; save a local copy to work with an external editor.
```

Brief explanation of code – System explanation

I started by defining the library of the DHT so it could be included within the code, this had to be pre-downloaded and placed into the code before the void setup and void loop section of the code. As well as this before the code I defined constant integers, these were the pins on the orangepi board for the three colours of the tri-coloured LED, as well as that I defined the minimum and maximum temperature as constant integers. Once these are defined at the start they cannot be changed throughout.

The void setup section is where the green, blue and red LED's colours are defined as outputs which means a signal can be sent from the orangepi board. This section is also where the serial print is started that was setup for the testing purposes of both the code and the hardware design.

The next section is the void which is a return method this is where the DHT gives a reading back from the temperature and humidity sensor onto the serial print so I could tell it was reading the temperature correctly. This section had the delay in which meant that after the temperature and humidity were shown on the serial print it was five seconds before the sensor then tested the environmental conditions again.

The next part of the code is the 'if' statement, if the statement is correct and the temperature is higher than the maximum temperature then the red LED will come on. The next statement is an 'else if' this means that if the first statement was not true then move onto this one and this one stated if the temperature is above the minimum but below the maximum the green LED will come on. Finally the last statement is an 'else' and it covers the final condition which would be that if the temperature is below the minimum the blue LED will come on.

Testing

The first test carried out on the circuit was the verifying check after I had written the code, this is already installed on the Arduino software and has to be done before loading the programme onto the ornaagepip board. This ensures there are no errors anywhere within the code and proves that it is correct, although this test only proves that there are no errors within the code it doesn't ensure that the code and the circuit are both going to work correctly.

To ensure the functionality of my test I created a table. The first test I carried out was to ensure when the temperature was within the green zone, which I thought to be a comfortable working temperature of 16°C – 24°C the LED was green. To prove this I took two pictures one with the temperature at 24°C and another at 17°C which would be the upper and lower boundaries of the comfortable working zone.

The next test I carried out was to ensure when the temperature drifted below 17°C that the tri-coloured LED changed from the green to the blue. To create this temperature I used freeze spray as the classroom I was working in was above this temperature. To prove that this worked I took a photo with the temperature at 16°C with the LED showing the blue colour. I also carried on this test by using the freeze spray again and ensuring that the LED stayed blue through all values 16°C and lower.

I had to ensure the LED turned red when the temperature was above 24°C. To do this I used a heat gun and started to heat up the DHT, as the temperature started to rise I was paying attention for when it rose above 16°C to ensure the LED changed from blue to green. I kept heating the DHT all the way through the comfortable working zone until I reached 25°C and this was when the LED changed from green to red. I kept the heat gun on to prove that the red LED stayed on when the temperature was above 24°C. To prove this I took photos with the temperature in the background.

There was one more test I needed to carry out and that was proving that if the temperature was in the red zone so above 24°C that when it started to reduce into the green working zone so between 16°C – and 24°C the LED changed from red to green. Likewise, I ensured that when the temperature was in the green working zone when it came under 16°C the LED would change from green to blue which it did. This meant I had fully tested my circuit moving both up and down the temperature scales making sure that the LED changed colour at the right time. This proves that both the circuit was wired correctly and the code was written correctly.

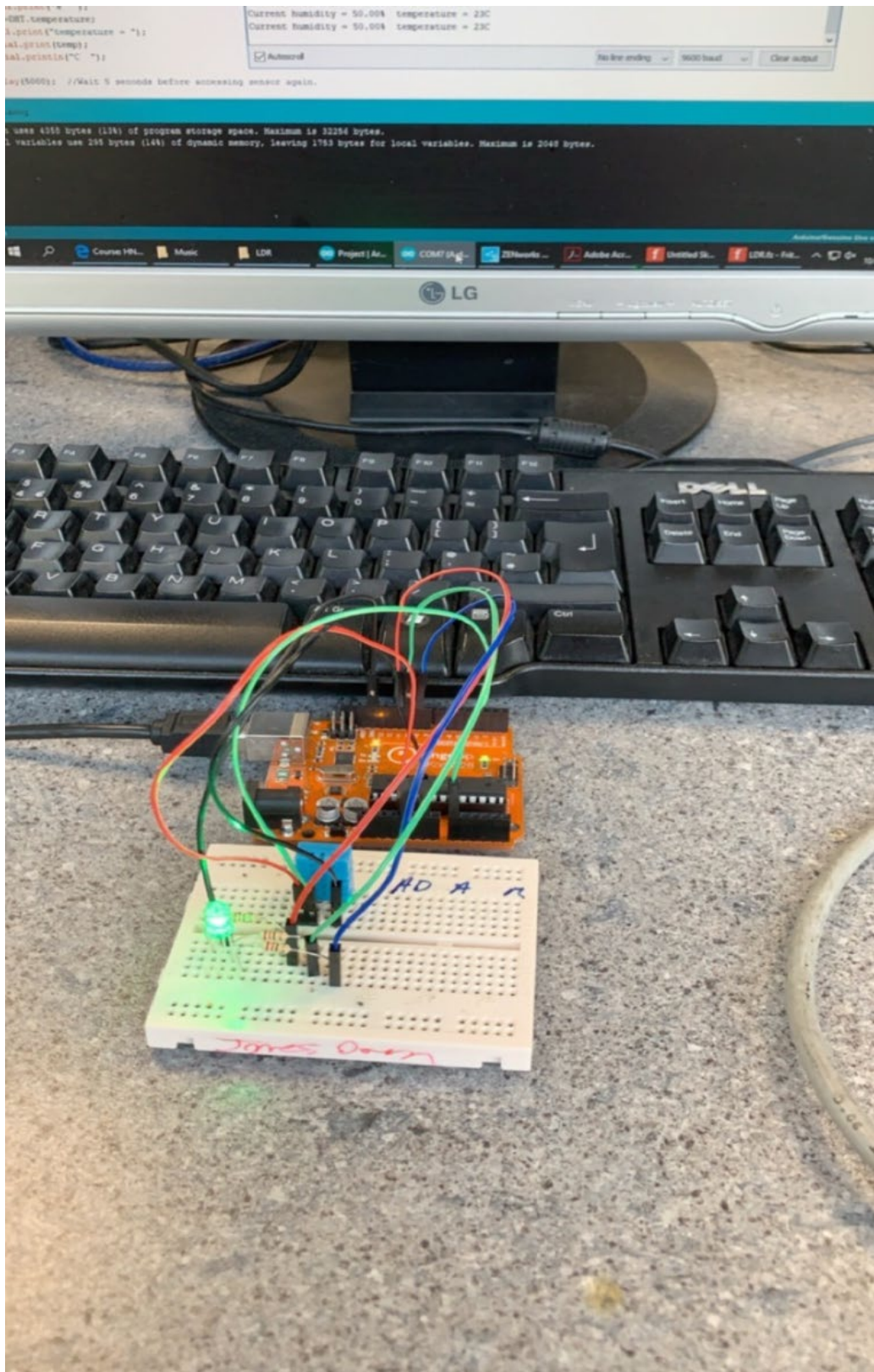
The table below shows the different colours of the LED at different temperatures I tested at. The table along with the images below shows that my circuit is working correctly.

Temperature (°C)	LED Colour
14	Blue
15	Blue
16	Blue
17	Green
18	Green
19	Green
20	Green
21	Green
22	Green
23	Green
24	Green
25	Red
26	Red
27	Red
28	Red

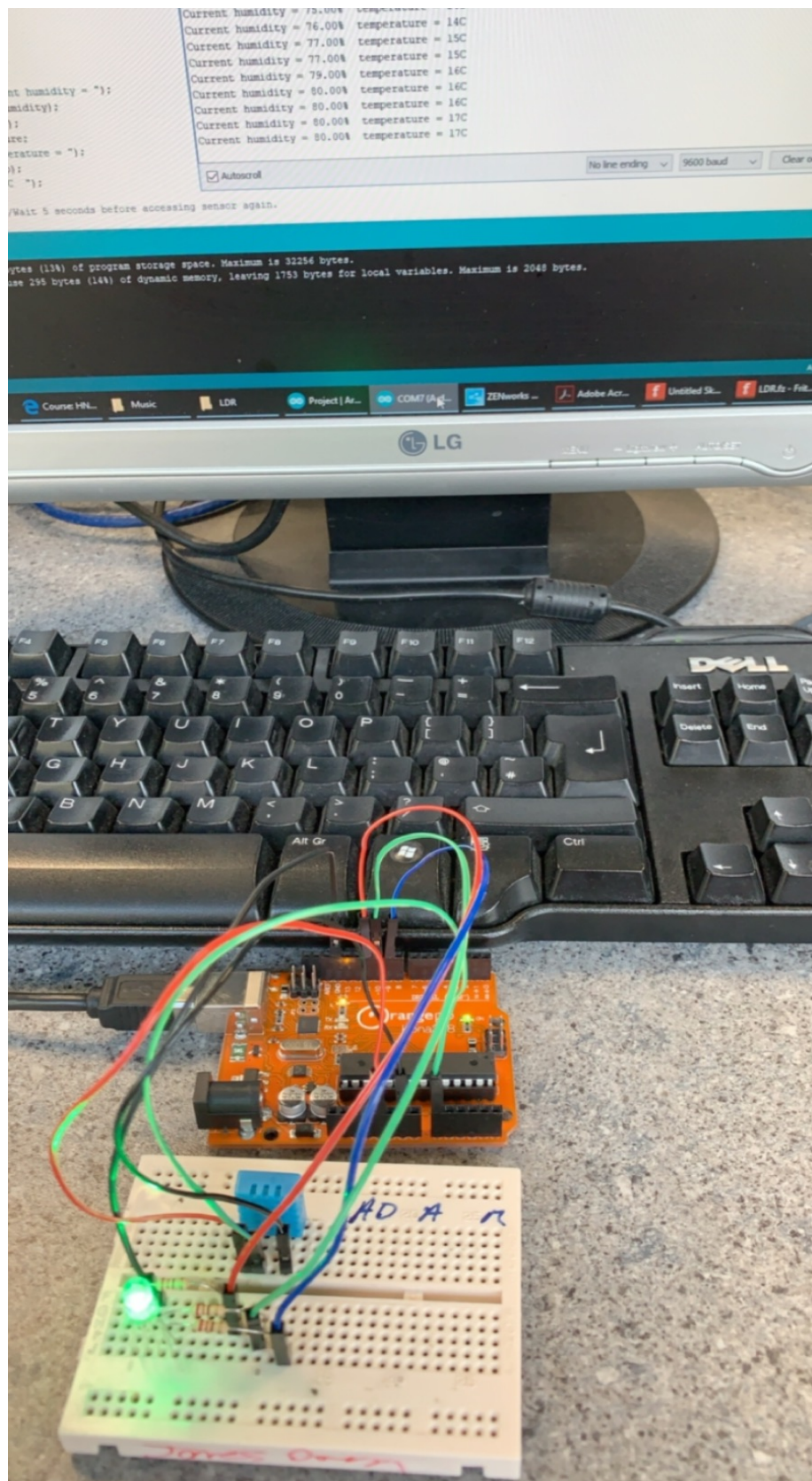
Images of testing

These first images show the LED when it is green and in the comfortable working zone.

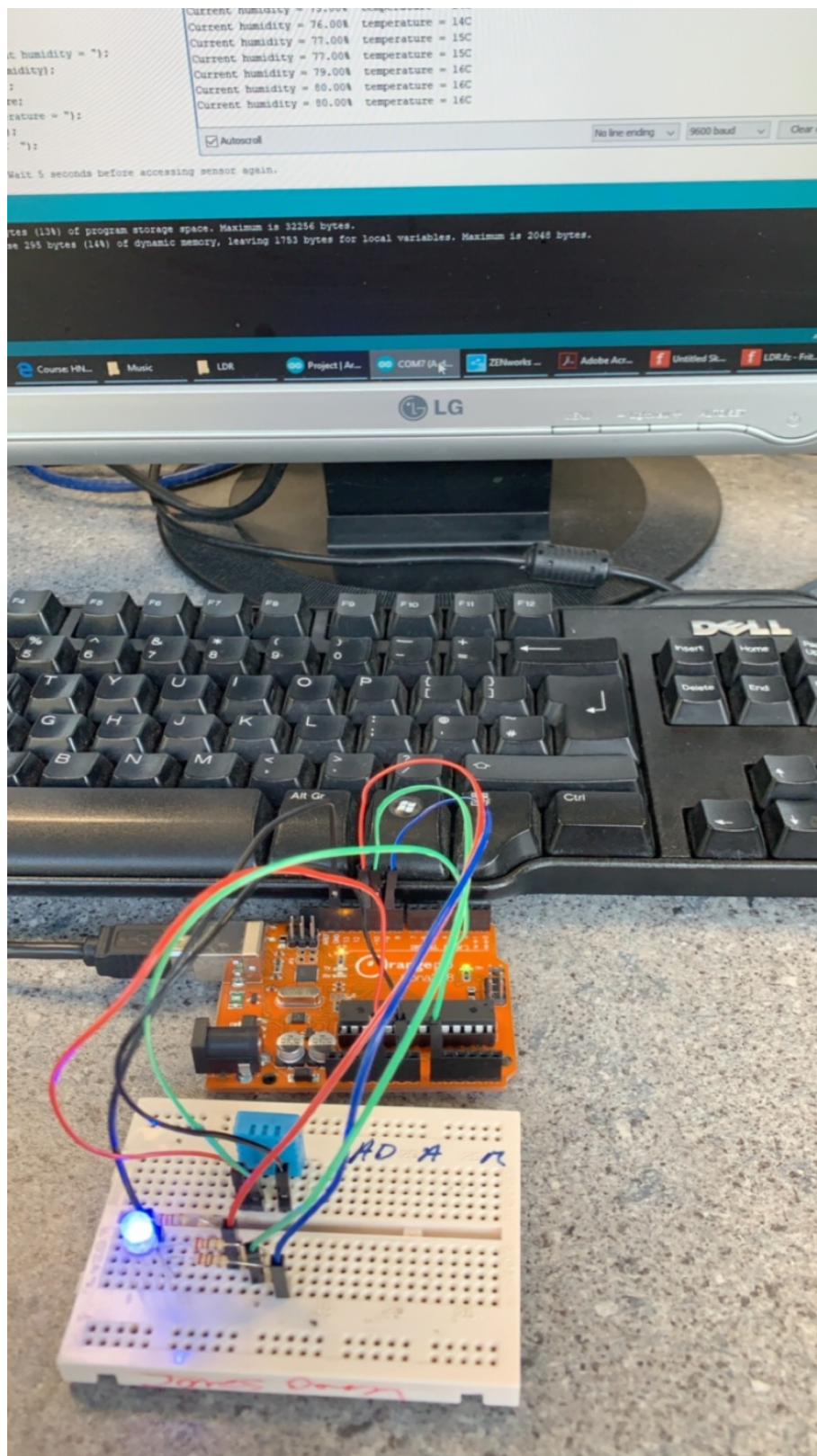
This first image shows the LED is green when the temperature is 24°C.



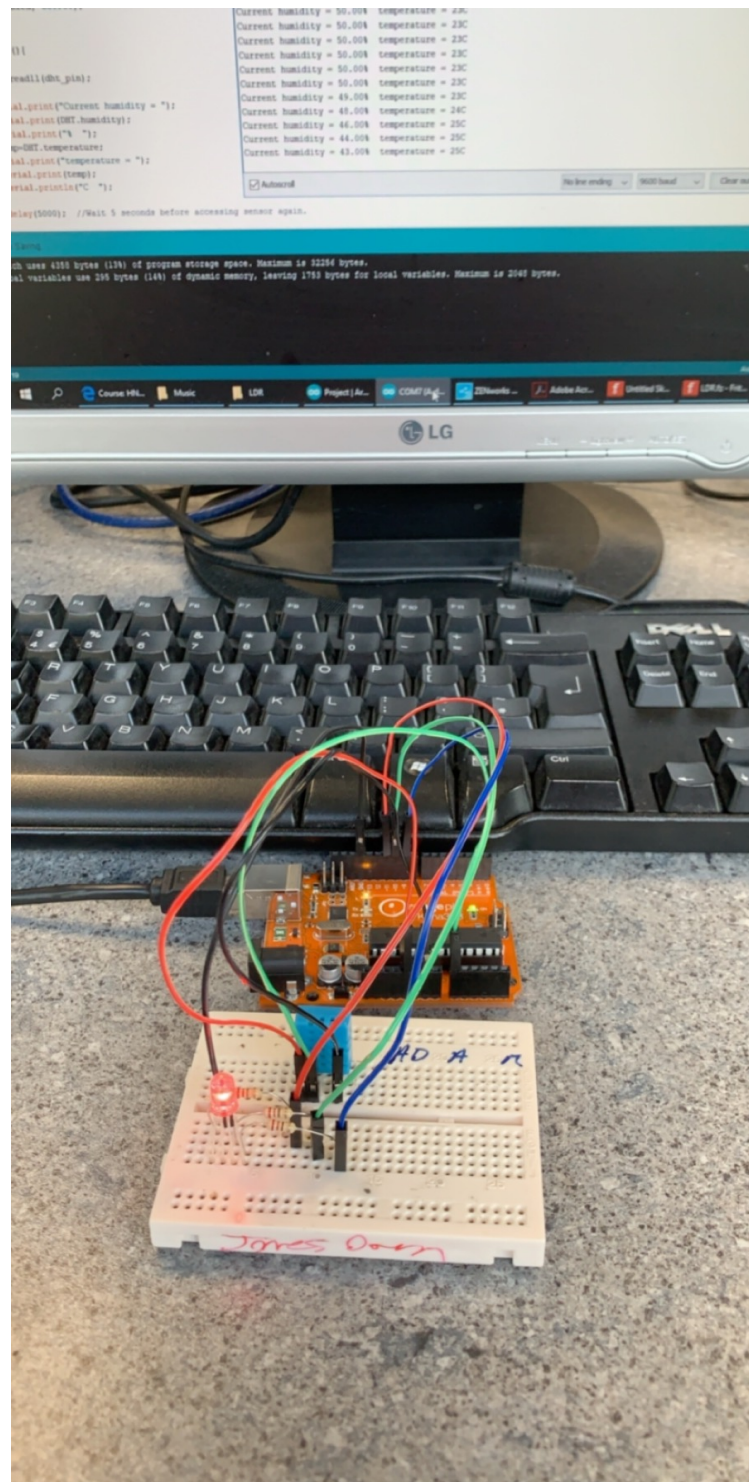
This image shows the LED is green when the temperature is 15°C



This next image shows that when the temperature changed from 15°C to 14°C the LED changed from green to blue.



The last image shows that when the temperature raised to 24°C the LED was red.



Evaluation and improvements

Overall, the circuit worked well and met the criteria needed, I used the tri coloured LED with resistors. I used the red colour of the three colours to show the temperature being too warm, I used the green LED to show that the temperature was within a comfortable working zone and finally I used the blue colour to show that the temperature was too cold. Both the circuit and the code that I had written worked. The circuit worked as the testing stage proves that as when the temperature falls into the different zones it works correctly with the colours I stated earlier in the assignment showing.

The circuit I created was very simple and did not have many components which made the code easier to write this meant there were improvements I could have made overall to the both the code and the circuit.

The first change I would have made was to power the board and the circuit with a batteries. This would make the circuit more suitable and would mean it would not have to be connected via a usb lead in front of the computer. This means the circuit could be taken into different environments not just in an office. Another change I would make to the circuit and code is to include audio representation as well as visual this would have meant that the orangepi board and breadboard circuit would not of had to sit in front of the individual working all day and could of been moved behind the computer or out of the way with a buzzer sounding this way the individual could have just grabbed the board and then seen the colour of the LED and made changes to the environment according to the indication on the LED. There is one other change that could be made and that is substituting the LED out for a LCD display screen and having the message show up on the screen when the temperature changes. This would be ideal if the person using the circuit was colour blind and could not actually see the change in colour of the tri coloured LED.

Another improvement that could have been made was using the humidity sensor as well as the temperature sensor as the DHT is capable of doing both. I could have included this into the coding part of the project and wouldn't have made any changes to the hardware. This would have been a good improvement as the humidity can have an effect on how hot or cold someone feels.

The final improvement I think I could have made was introducing something into the code which allows the person that is using the code to pick the temperatures that they feel comfortable in rather than having them already inside the code. This would of meant my code was a lot more complex but also meant with the addition of the battery powered source it could be used anywhere.

Appendixes

Orangepip Kona328 board – Datasheet



Kona328 Arduino UNO Compatible Development Board

Order code: **75-0550**

Overview

The Kona328 is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

The Kona328 differs from all preceding boards in that it does not use the FTDI USB-to-serial driver chip. Instead, it features the ATmega8U2 programmed as a USB-to-serial converter. Revision 2 of the Kona328 board has a resistor pulling the 8U2 HWB line to ground, making it easier to put into DFU mode.

Summary

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Power

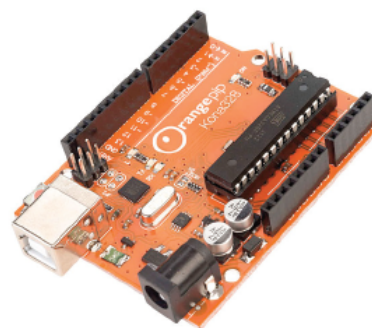
The Kona328 can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.

The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

VIN The input voltage to the Kona328 board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.



5V The regulated power supply used to power the microcontroller and other components on the board. This can come either from VIN via an on-board regulator, or be supplied by USB or another regulated 5V supply.

3V3 A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.

GND Ground pins.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Kona328 can be used as an input or output, using `pinMode()`, `digitalWrite()`, and `digitalRead()` functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

Serial: 0 (RX) and 1 (TX) Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

External Interrupts: 2 and 3 These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the `attachInterrupt()` function for details.

PWM: 3, 5, 6, 9, 10, and 11 Provide 8-bit PWM output with the `analogWrite()` function.

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK) These pins support SPI communication using the SPI library.

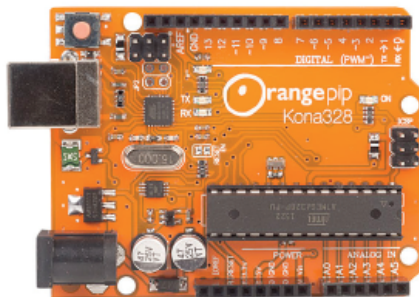
Page 1 of 3

www.orangepipboards.com



Kona328 Arduino UNO Compatible Development Board

Order code: **75-0550**



LED: 13 There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.

The Kona328 has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though it is possible to change the upper end of their range using the AREF pin and the `analogReference()` function. Additionally, some pins have specialized functionality:

TWI: A4 (SDA) and A5 (SCL) Support TWI communication using the Wire library. There are a couple of other pins on the board:

AREF Reference voltage for the analog inputs. Used with `analogReference()`.

Reset Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

See also the mapping between OrangePi pins and ATmega328 ports?

Communication

The Kona328 has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega8U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '8U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Kona328's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; see the documentation for details. For SPI communication, use the SPI library. Programming

The Kona328 can be programmed with the Arduino software. Select "Arduino Uno" from the Tools > Board menu (according to the microcontroller on your board). For details, see the reference and tutorials.

The ATmega328 on the Kona328 comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the original STK500 protocol (reference, C header files).

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header; see these instructions for details.

The ATmega8U2 firmware source code is available. The ATmega8U2 is loaded with a DFU bootloader, which can be activated by connecting the solder jumper on the back of the board and then resetting the 8U2. You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer.

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Kona328 is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Kona328 is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Kona328. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

Page 2 of 3



Kona328 Arduino UNO Compatible Development Board

Order code: **75-0550**

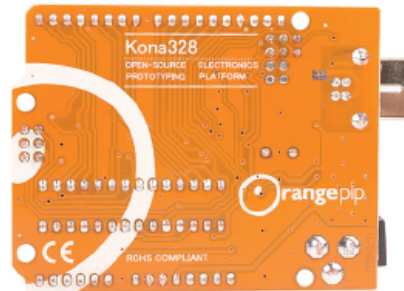
The Kona328 contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Overcurrent Protection

The Kona328 has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Kona328 PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.



Datasheet	5mm RGB LED common cathode http://www.kitronik.co.uk
-----------	---

Kitronik Ltd – 5mm RGB LED Common Cathode

TECHNOLOGY DATA SHEET & SPECIFICATIONS



Device Selection Guide

Chip		Lens Colour
Material	Emitted Color	
AlGaInP	Red	Water clear
InGaN	Green	
InGa1N	Blue	

Features

- Uniform light output.
- Low power consumption.
- I.C. compatible.
- Long life solderability.
- Common Cathode.

Descriptions

- The Red source colour devices are made with AlGaInP on GaAs substrate.
- The Green source colour devices are made with InGaN on sic.
- The Blue source colour devices are made with InGaA1N on sic.

Usage Notes:

- The ultra bright LED is an electrostatic sensitive device, so static electricity and surge will damage the LED.
- It is required to wear a wrist-band when handling the LED. All device, equipment, machinery, desk and ground must be properly grounded.
- When using LED, it must use a protective resistor in series with DC current about 20mA.

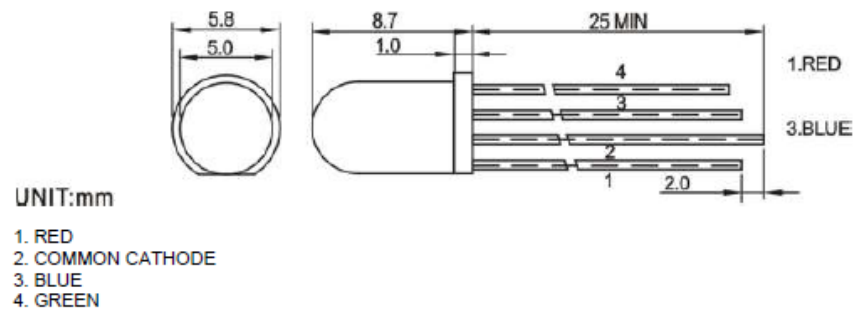
Applications

- Status indicators.
- Commercial use.
- Advertising signs.
- Back lighting.



Datasheet	5mm RGB LED common cathode http://www.kitronik.co.uk
-----------	---

Package Dimensions



Notes:

- Other dimensions are in millimetres, tolerance is 0.25mm except being specified.
- Protruded resin under flange is 1.5mm Max LED.
- Bare copper alloy is exposed at tie-bar portion after cutting.

Datasheet	5mm RGB LED common cathode http://www.kitronik.co.uk
-----------	---

Absolute Maximum Rating (Ta=25°C)

Parameter	Symbol	Absolute Maximum Rating	Unit
Forward Pulse Current	IFPM	70	mA
Forward Current	IFM	30	mA
Reverse Voltage	VR	5	V
Power Dissipation	PD	140	mW
Operating Temperature	Topr	-40~+80	°C
Storage Temperature	Tstg	-40~+100	°C
Soldering Temperature	Tsol	Reflow Soldering : 260 °C for 10 sec. Hand Soldering : 350 °C for 3 sec.	°C

Electro-Optical Characteristics (Ta=25°C)

Parameter	Symbol	Device	Min.	Typ.	Max.	Unit	Test Condition
Luminous Intensity	Iv	Red Green Blue	1000 1200 1000	1500 2000 1500	2300 2700 2200	mcd	IF=20mA
Viewing Angle	2θ1/2	Red Green Blue	40	---	50	Deg	
Peak Emission Wavelength	λp	Red Green Blue	635 520 460	640 525 465	650 530 470	nm	IF=20mA
Spectral Line Half-Width	Δλ	Red Green Blue	15 15 25	20 20 30	25 25 35	nm	IF=20mA
Forward Voltage	VF	Red Green Blue	1.9 2.9 2.9	---	2.5 3.5 3.5	V	IF=20mA
Reverse Current	IR	Red Green Blue	---	---	10	μA	VR=5V



DATA SHEET

Prototyping Boards

Order code	Manufacturer code	Description
34-0650	n/a	PROTOBLOC 1 BREADBOARD (RE)
34-0655	n/a	PROTOBLOC 2 BREADBOARD (RE)
34-0660	n/a	PROTOBLOC 2A BREADBOARD(RH21B) (RE)
34-0662	n/a	PROTOBLOC 2B PROTOBLOC (AD-11) (RE)
34-0664	n/a	PROTOBLOC 2C PROTOTYPING BOARD (RE)
34-0665	n/a	PROTOBLOC 3 BREADBOARD (RH-74)
34-0670	n/a	2T SOCKET SET (RH-10)
34-0675	n/a	2D SOCKET SET

Protobloc prototyping boards



Material:

Back plate : ABS polymer

External body : ABS polymer

Internal contact terminal : Alloy of silver and nickel

Resistance : $< 10\text{m } \Omega$ (0.6 ϕ mm lead / at 1KHz)

Capacitance : $< 10\text{pF}$ (0.6 ϕ mm lead / at 1MHz)

Insulation : $> 10^9 \Omega$ (500VDC / 1min)

HI-POT Test : The appearance 、 instruction no change (1500VDC / 1min / Breakdown current : 30mA)

Contact current : 3A (0.6 ϕ mm lead)

Wire size acceptability : 0.3 ϕ mm $\sim$ 0.8 ϕ mm

Note : For security purpose , please do not use high voltage on the bread board (AC $> 50\text{V}$, DC $> 75\text{V}$)



For more products visit our website <http://www.sunrom.com>

Document: Datasheet

Date: 20-Jun-12

Model #: 3732

Product's Page: www.sunrom.com/p-1141.html

DHT11 - Humidity and Temperature Sensor

The DHT11 is a basic, low-cost digital temperature and humidity sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and spits out a digital signal on the data pin (no analog input pins needed).

Its fairly simple to use, but requires careful timing to grab data. The only real downside of this sensor is you can only get new data from it once every 2 seconds.

Features

- Full range temperature compensated
- Relative humidity and temperature measurement
- Calibrated digital signal
- Outstanding long-term stability
- Extra components not needed
- Long transmission distance
- Low power consumption
- 4 pins packaged and fully interchangeable



Details

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. Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process.

The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package.



Specifications

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

Parameters	Conditions	Minimum	Typical	Maximum
Humidity				
Resolution		1%RH	1%RH	1%RH
			8 Bit	
Repeatability			± 1%RH	
Accuracy	25°C		± 4%RH	
	0-50 °C			± 5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			± 1%RH	
Long-Term Stability	Typical		± 1%RH/year	
Temperature				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			± 1°C	
Accuracy		± 1°C		± 2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

Item	Condition	Min	Typical	Max	Unit
Power supply	DC	3	5	5.5	V
Current supply	Measuring	0.5		2.5	mA
	Stand-by	100	Null	150	uA
	Average	0.2	Null	1	mA

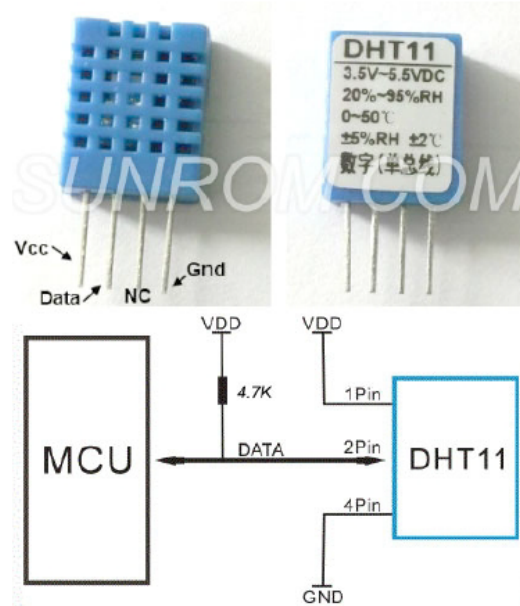
2

Sunrom



www.sunrom.com

Typical Application



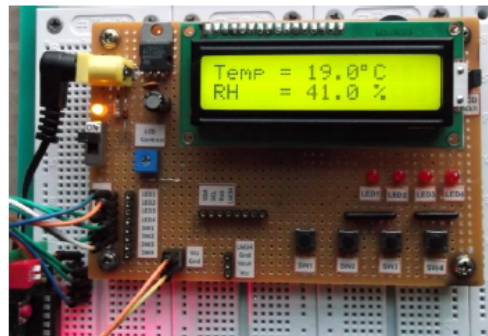
DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

SDK (Software Development Kit)

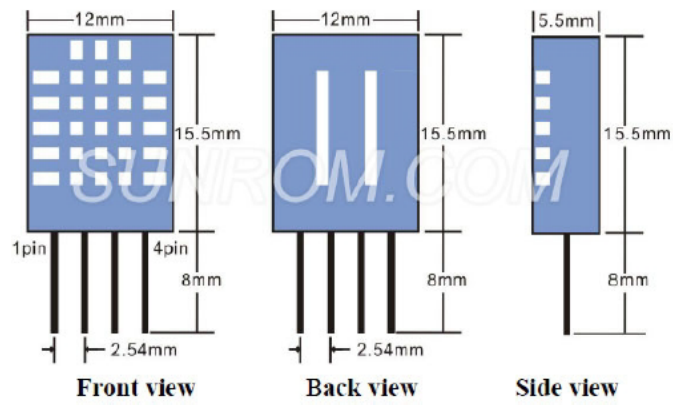
Download source code + project articles by clicking following link

<http://www.sunrom.com/files/3732.zip>

It contains details for AVR, PIC and Arduino projects.



Dimensions (mm)



7

Sunrom



7

/ 7



www.sunrom.com

References

Appendixes – Datasheets

<https://www.rapidonline.com/orangepip-kona328-arduino-uno-compatible-development-board-75-0550>

https://www.kitronik.co.uk/pdf/5mm_RGB_led_common_cathode.pdf

<https://www.rapidonline.com/k-h-rh-21-prototyping-breadboard-840-tie-points-34-0655>

<http://robocraft.ru/files/datasheet/DHT11.pdf>

References in text

<http://www.hse.gov.uk/temperature/faq.htm>

Bibliography

Noergaard T. *Embedded Systems Architecture: A Comprehensive Guide for Engineers and Programmers*, 2nd Edition

Ganguly A K. *Embedded Systems: Design, Programming and Applications*. Alpha Science International Ltd

Blum R. *Arduino Programming in 24 Hours, Sams Teach Yourself*

Lewis D.W, *Fundamentals of Embedded Software*. Prentice Hall

Deitel D J. *C How To Program*, 8th Edition

<https://www.gov.uk/workplace-temperatures>

<http://www.hse.gov.uk/temperature/faq.htm>