

LEVEL 4 ASSESSMENT SPECIFICATION

Student name:	Samuel Davies	Student ID number:	2221719
Programme:	HNC Electrical/Electronic Engineering & Instrumentation Engineering		
Module:	Embedded Systems		
Module code:	PCEE4004	Contribution to Overall Module Assessment (%):	100%
Lecturer:	Sarah Williams	Internal Verifier:	Jason Pointer
Assignment Title:	Office Environment Monitor	Word count (or equivalent):	2000 words
Submission deadline:	12 th May 2023	Return date of provisional marks & written feedback:	10 th June 2023
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.		

Module Learning Outcomes (from module syllabus)

1. Demonstrate a good understanding of the fundamental principles of programming an embedded system.
2. Describe the basic elements of typical embedded hardware and apply that knowledge in the analysis of a number of simple system applications.

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.

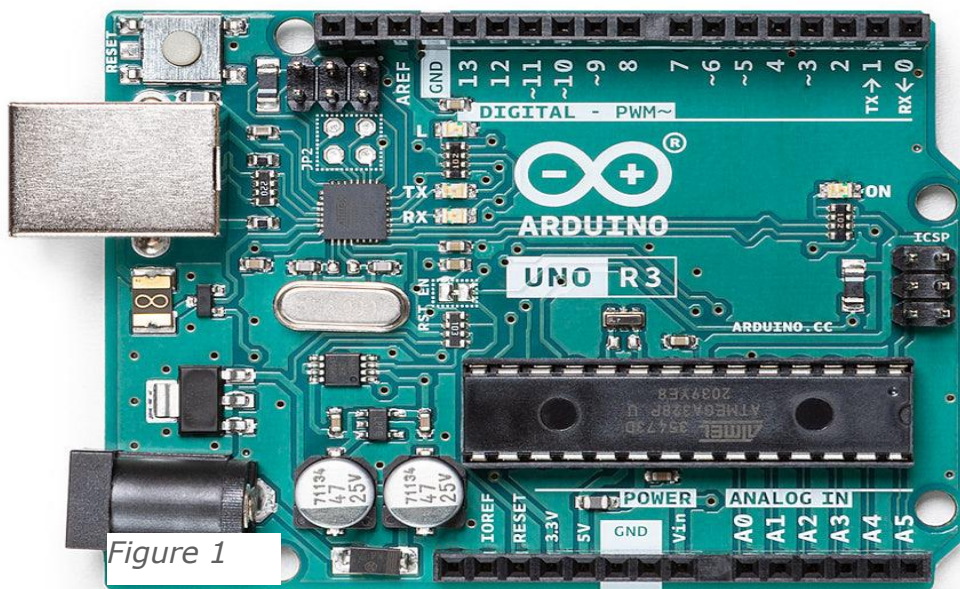
Generic Assessment Criteria	Marks available	Marks awarded
1. Engagement with Literature Skills		
	0	
2. Knowledge and Understanding Skills		
Hardware Competent knowledge of the basic underlying concepts and principles. Components selected have been justified and some limited discussion on alternatives included. Audible components could also be considered. Schematic included is for a functional system and is accurate apart from the 8 x 8 led matrix which is a slightly different component to the one used and as such is wired slightly differently. 9/15 Software Limited knowledge and understanding of the basic underlying concepts. Flowchart logic is flawed - there is an overlap for the value 20 and 16 will completely bypass the system. Symbols used in the diagram are correct as in the indication of the flow. Reasonable attempt made at pseudocode, however it should be clear what is global, what is in the setup section of the sketch, and what is in the loop. It convention to end an IF code block with an END IF. 7/15	30	16
3. Cognitive and Intellectual Skills		
Discussion, Evaluation and Conclusion Limited attempt at evaluation of the underlying concepts and principles, tending towards description. In the conclusion the omission of certain temperature ranges was acknowledged, this would have been more appropriate to be picked up and discussed in the testing, however it was detected. Consideration has also been given to some of the shortcomings of the components selected. No discussion has been made in regards to the effectiveness of the C language when working with embedded systems. 7/15	15	7
4. Practical Application Skills		
Implementation An adequate awareness and mostly appropriate application of basic methods, slight error in method of implementation used. Program is in the most part functional and meets the requirements of the assignment. Variables and constants are used and appropriately named. Program starts in setup by showing a too hot read out- I am not sure why you would start with this without taking a reading. Perhaps a start-up image to show system is booting up would be more appropriate. You read in humidity- but never do anything with it. As the Arduino has a limited memory capacity you should not declare variables that are not used.	50	24.5

Code is commented however correct indentation is not used. Consider including a function to provide the loop that outputs the images on the led matrix as this code is duplicated. The use of serial monitor is good for testing, but as they system is unlikely to be tethered to a PC that is running the Arduino IDE, it is good practice to comment the code out. The issues detected in the flowchart are implemented into the code, so as a result of the DHT11 reading to the neared 1/10 of a degree, your program does not respond to temperatures in the range of 16 to 16.9 degrees. 19.5/40		
Testing A good and appropriate application of basic testing techniques has been used. This program would really have benefited from boundary testing as it would have detected that between the values 16 and 16.9 nothing is displayed due to the flaw in the if statement. All states have been tested and evidenced. 5/10		
5. Transferable Skills for Life and Professional Practice		
Work is coherent, very fluent and is presented professionally. Can work effectively independently. Work is set out formally, well structured, including contents page, referenced statements where required and labelled diagrams. 4/5	5	4

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)		51.5%
	Up to 1 week late (40% Max)		
	Over 1 week late (0%)		

EMBEDDED SYSTEMS PCE4004

Year 1- HNC Instrumentation Engineering



Samuel Davies

Level 4 HNC in Instrumentation Engineering

Pembrokeshire College / UOW Trinity St David's

Student no: 146387 / 2221719 [146387@pembrokeshire.ac.uk]



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

TABLE OF CONTENTS

INTRODUCTION: A BRIEF OVERVIEW OF THE DESIRED OUTCOME FROM READING THE ENCLOSED REPORT (PAGE 3)

TASK 1: HARDWARE DESIGN (PAGES 4-5)

TASK 2: SOFTWARE DESIGN (PAGES 6-7)

TASK 3: IMPLEMENTATION (PAGES 8-10)

TASK 4: TESTING (PAGE 11)

TASK 5: DISCUSSION, EVALUATION AND CONCLUSION (PAGE 12)

REFERENCES: A LIST OF PUBLISHED SOURCES OF MATERIAL REFERRED TO IN THE TEXT (PAGE 13)

BIBLIOGRAPHY: A LIST OF PUBLISHED SOURCES OF MATERIAL NOT REFERRED TO IN THE TEXT (PAGE 14)

APPENDIX: A LIST OF ADDITIONAL READING MATERIALS SUCH AS DATA SHEETS, TESTING EVIDENCE & FIGURES (PAGES 15-23)

INTRODUCTION-

The following report should demonstrate a system created in order to monitor the effectiveness of heating and air condition systems in a workplace. The device created will alert someone should the temperature become too hot or too cold, based on “normal” comfortable working conditions. Normal, comfortable working temperature is defined as being “at least 16 degrees C, or 13degrees if the work involves rigorous effort”, (GOV, H. 2019). There is no law on the maximum temperature in the workplace, but every effort should be made to ensure employees are comfortable. Based on this, the device will alert should the temperature drop below 16°C, or above 20°C.

The embedded system will contain both input and output devices, with a microcontroller at the core. C programming language is used to control the microcontroller, and there is also a flow chart and pseudo code enclosed, to help the reader understand the desired function from the program. The report details the code used, the testing carried out to ensure validity of the system, and an evaluation of the desired outcomes, and whether the brief was achieved. Lastly, there is recommendations on what can be changed should the system be re-designed to improve the overall performance.

TASK 1 (HARDWARE DESIGN)-

Below is the Fritzing diagram showing the layout of the project circuitry. The main components include; Arduino Uno controller, DHT11 temperature and humidity sensor, 8x8 dot matrix display, 10k Ω resistor and breadboard.

The Arduino controller has been used, despite many alternatives, as the programming software and accessories were readily available. Also, the online support available for this choice on controller was second to none. The Arduino Uno is also cost effective, as opposed to alternatives such as the PJRC Teensy 4.1, and accessing and modifying the code was as simple as plugging in via USB to the Comms port. The only function that the micro-controller lacks, is Wi-Fi connectivity, but this wasn't essential to this project.

The DHT11 temperature sensor was chosen due to its relatively low cost, compared to DHT21, and its ease of use and programming. Again, the support available online helped sway choosing this component. Also, it gives its output on a single as opposed to multiple pins as some of its competitors do, reducing the complexity of wiring.

An 8x8 dot matrix display was used, as it is a beneficial output device, that is easily programmed to give logos, figure, letters etc. on the display. This device was chosen as opposed to an LCD display, as there was ample support online to gather the byte arrays for the required display.

The 10k Ω resistor is used in the circuit to limit the current entering the device, as the manufacturer states, "When the connecting cable is shorter than 20 meters, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 meters, choose an appropriate pull-up resistor as needed", (Electronics, M, 2018). Based on the above, a 10k Ω was used as good practice, despite the cable run being shorter than 20 meters.

EMBEDDED SYSTEMS – PCE 4004

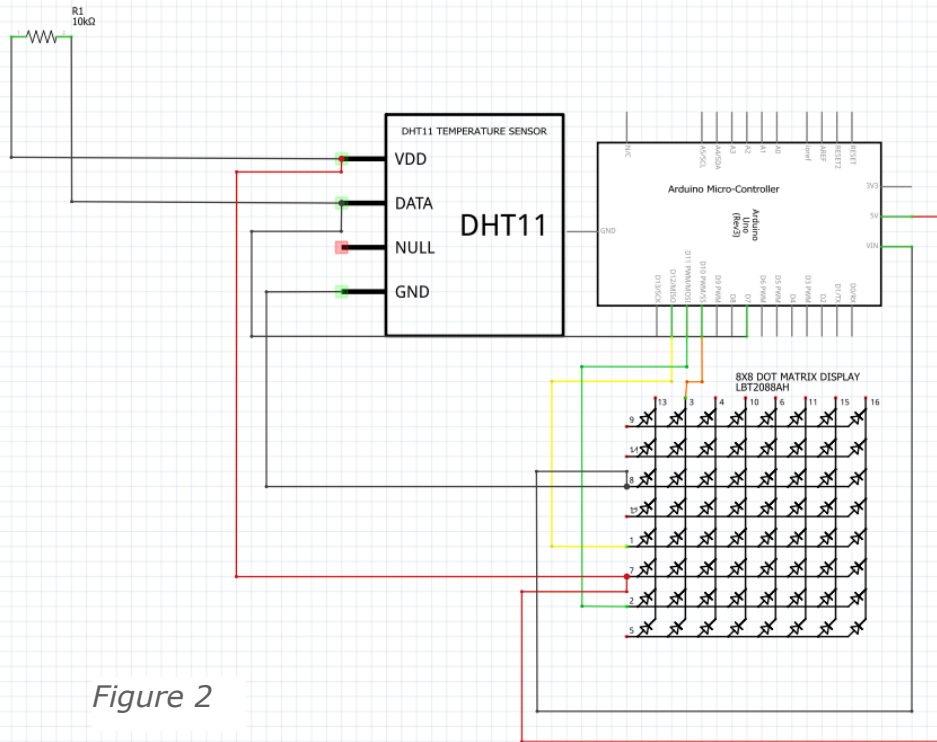


Figure 2

EMBEDDED SYSTEMS – PCE 4004

TASK 2 (SOFTWARE DESIGN)-

The code needs to control both the DHT11 input device, and the 8x8 dot matrix output device, the below Flow Chart shows an overview of the process that will be followed in the C Program. Based on the information listed in the introduction, on HSE recommendations, the device will alert the user should the temperature fall below 16°C, or above 20°C, and will also output an ok alert when the temperature is >17°C and <=20°C.

C programme was chosen due to its ability to access hardware at low level, and as a language it is easier to develop than others such as assembler. As a method of coding, it is generally easy to programme, requires smaller memory usage and is compatible from various manufacturers. C programme is a simple method of achieving an application, with “your first step is to decide what the application should do. Once you've done that, you write the C source code and compile it into an executable format. Finally, you can run the executable application”, (Ullman & Liyanage, 2005).

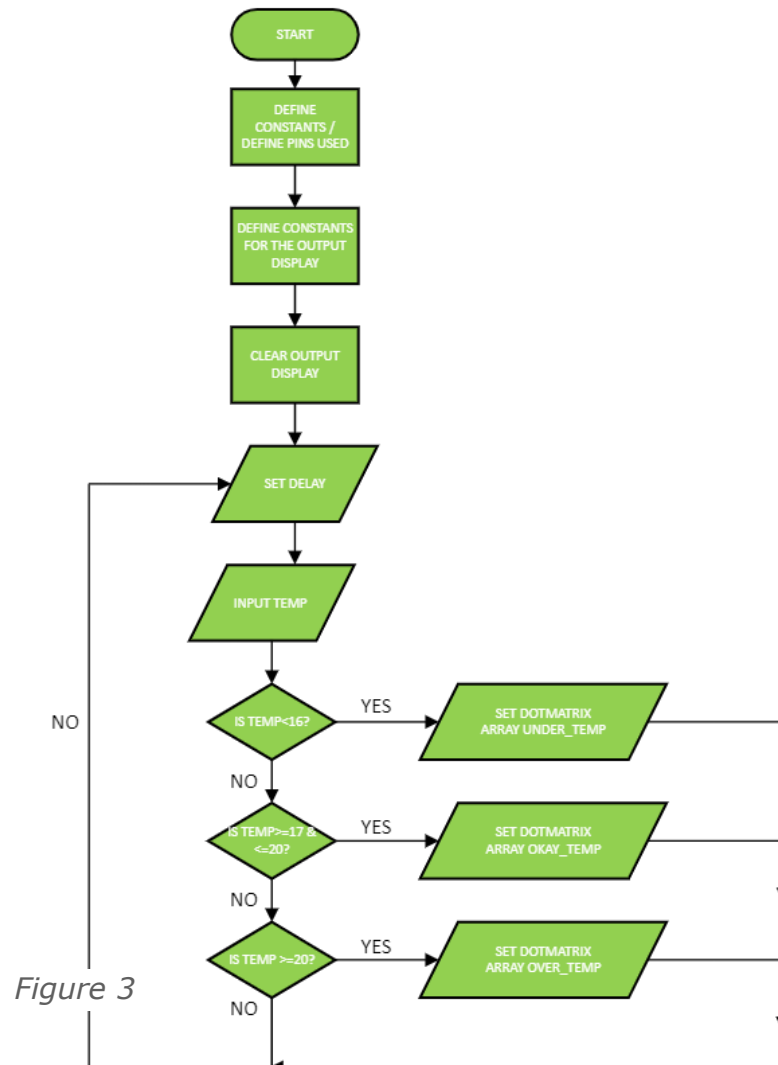


Figure 3

EMBEDDED SYSTEMS – PCE 4004

In order to further aid implementation, pseudo code has also been produced, as it is similar to C code, but understandable by any programmer or code writer, and “Pseudo code can be transformed into any programming language code with the help of appropriate libraries or API”, (Rakib & Vinh, 2019). Pseudo code, along with the flow chart above will aid understanding of the desired function of the system.

define DHT11_PIN 7

LedControl matrix = LedControl (12,10,11,1

SetDHT11 as INPUT

SetMatrix.Display as OUTPUT

int Temp= read(PIN 7)

if temp=<16°C

write(MATRIX>;UNDER_TEMP)

if temp ->17°C & =<20°C

write(MATRIX.OK_TEMP)

if temp =>20°C

write(MATRIX>OVER_TEMP)

Matric.clear.dispay

Delay(5000)

EMBEDDED SYSTEMS – PCE 4004

TASK 3 (IMPLEMENTATION)-

The below C code was produced using Arduino IDE, accessed through Pembrokeshire College's "Zen Works", (by Samuel Davies, January-May 2023)

```
#include "DHT.h"           // # is part of language, doesn't mean anything  '.h' is extension of
                             file type just like '.pdf'
#include "LedControl.h"
#define DHT11_PIN 7
```

```
LedControl matrix = LedControl (12,10,11,1);  // this is the dot matrix output display device
```

```
DHT tempSensor (DHT11_PIN,DHT11);           // this is the input temp / humidity sensor
const uint64_t OVER_TEMP[] =                 // the below byte array is a sad face with "H" also
displayed. This will inform the user OVER_TEMP
```

```
{
    0b000000101,
    0b000000111,
    0b000000101,
    0b00100100,
    0b00000000,
    0b00111100,
    0b01000010,
    0b10000001
};
```

```
const uint64_t OK_TEMP[] =                   // the below byte array is a smiley face displayed.
This will inform the user OK_TEMP
```

```
{
    0b00000000,
    0b00000000,
    0b00100100,
    0b00000000,
    0b00000000,
    0b10000001,
    0b01000010,
    0b00111100
};
```

```
const uint64_t UNDER_TEMP[] =               // the below byte array is a sad face with "C" also
displayed. This will inform the user UNDER_TEMP
```

```
{
    0b00000011,
    0b00000010,
    0b00000011,
    0b00100100,
```

EMBEDDED SYSTEMS – PCE 4004

```
0b00000000,  
0b00111100,  
0b01000010,  
0b10000001  
  
};  
  
void setup() {  
    // put your setup code here, to run once:  
  
    matrix.shutdown(0,false);  
    matrix.setIntensity(0,4);    //first parameter is 0, intensity is 4/8, brightness of dots.  
    matrix.clearDisplay(0);    // this switches of any LEDs that were on, it clears the display  
  
    Serial.begin(9600);  
    delay(500);    //Delay to let system boot up  
    // put booting up code/ comment  
    Serial.println("DHT11 Humidity & temperature Sensor\n\n");  
    delay(1000);    //Wait before accessing Sensor  
  
    tempSensor.begin();  
    int rows;  
    for(rows=0;rows<8;rows++){  
        matrix.setRow(0,rows,OVER_TEMP[rows]);  
    }  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
    float humidity = tempSensor.readHumidity();    //float command reads the amount after  
    the decimal places.  
    float temp = tempSensor.readTemperature();  
  
    Serial.println(temp);  
  
    int rows;  
  
    if (temp> 20)    // if temperature is above 20degrees, the output device  
        will display the OVER_TEMP array  
  
    {  
  
        for(rows=0;rows<8;rows++){
```

EMBEDDED SYSTEMS – PCE 4004

```
matrix.setRow(0,rows,OVER_TEMP[rows]);
}
delay(10000);
matrix.clearDisplay(0);
}

if (temp>= 17 & temp<=20)           // if temperature is below 20degrees & above
17degress, the output device will display the OK_TEMP array

{

for(rows=0;rows<8;rows++){
matrix.setRow(0,rows,OK_TEMP[rows]);
}
delay(5000);
matrix.clearDisplay(0);
}

if (temp< 16)                       // if temperature is below 16degrees, the output device will
display the UNDER_TEMP array

{

for(rows=0;rows<8;rows++){
matrix.setRow(0,rows,UNDER_TEMP[rows]);
}
delay(5000);
matrix.clearDisplay(0);
}

Serial.print("Current humidity=");
Serial.print(humidity);
Serial.print("% ");
Serial.print("temperature = ");
Serial.print(temp);
Serial.print("C ");

delay(5000);           //wait 5 seconds before accessing temperature sensor again
}
```

EMBEDDED SYSTEMS – PCE 4004

TASK 4 (TESTING)-

There have been 3 tests carried out on the temperature monitoring device, with the results listed below. The purpose of testing such an embedded system is to try and find any unexpected results that may have been overlooked during the early stages of software / hardware design. The main principle of testing such as system is to trigger expected inputs, and see if the expected output occurs, “testing demonstrates to yourself (and regulatory agencies, if appropriate), that the system and software works correctly and as designed”, (Ganssle et al., 2008)

Photographic evidence can be found in the appendix;

Test number	Test	Inputs	Expected Outputs	Actual Outputs	Issues
1	Testing UNDER_TEMP function	Temp=<16°C	8x8 dot matrix display= 😞 sad face & letter “C”	The actual display was as should, with 😞 a sad face and letter C both displayed	No issues to report
2	Testing OK_TEMP function	Temp=>17°C & <20°C	8x8 dot matrix display= 😊 smiley face	The actual output display was as should, with 😊 a smiley face displayed	No issues to report
3	Testing OVER_TEMP function	Temp=>20°C	8x8 dot matrix display= 😞 sad face & letter “H”	The actual output displayed was as should, with a 😞 sad face & the letter “H” displayed	No issues to report, although the display was quite congested with the letter “H” being hard to read.

EMBEDDED SYSTEMS – PCE 4004

TASK 5 (DISCUSSION, EVALUATION & CONCLUSION)-

Upon review of the embedded system produced, the brief stated to create “a simple device that will report if the environment is outside of “normal” comfortable working conditions”, and the device achieves that, through the 8x8 dot matrix display. Upon reflection, the device does report should the temperature be outside of the defined set points, and it also send an OK output to let the user know the temperature is within the range, but there is no audible alert, so the user would have to physically view the display to know if the office temperature is okay. If the system was to be designed again, an audible device such as a buzzer, would be incorporated to ensure that the device gains the users attention should the temperature within the office not be okay.

Upon reviewing the program, it is effective for its purpose and the testing proves that the inputs actuate to desired outputs. Evaluating the C program, it appears that although the lower limit ($<16^{\circ}\text{C}$), upper limit ($>20^{\circ}\text{C}$) are both reflected, the middle temperature is programmed at ($>17^{\circ}\text{C}$ & $<20^{\circ}\text{C}$), so this leaves $>16^{\circ}\text{C}$ & $<17^{\circ}\text{C}$ excluded, resulting in the display not outputting anything. In future systems, it will be ensured that the full range of temperature will be covered in the scale of input detection.

Another realization upon testing, was the difficulty in viewing a dot matrix display at times. It was noticed that the letter as well as a smiley or sad face can be quite difficult to define, particularly if the room is in direct sunlight. Taking this into account, in future projects LCD displays could be utilized to ensure the output display is easily accessible and clear in its information.

In conclusion, the systems designed and tested, does meet the brief set out, by alerting should the temperature be outside of the defined temperatures, but moving forward, some small hardware and software changes would make the system more user friendly, accessible and effective.

REFERENCES-

Electronics, M. (2018) DHT11 humidity & temperature sensor - Mouser Electronics, DHT!11 Humidity & Temperature Sensor. Available at: <https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf> (Accessed: 01 May 2023).

Ganssle, J.G. and Ball, S.R. (2008) 'Testing', in Embedded systems. Amsterdam: Elsevier/Newnes, pp. 49–49.

GOV, H. (2019) Temperature in the workplace, Managing workplace temperature. Available at: <https://www.hse.gov.uk/temperature/employer/managing.htm#:~:text=a%20reasonable%20working%20temperature%20in,hot%20and%20cold%20manufacturing%20processes> (Accessed: 03 May 2023).

Rakib, A. and Vinh, P.C. (2019) Text to Code: Pseudo Code Generation, Library Search. Available at: https://librarysearch.cardiff.ac.uk/discovery/fulldisplay?docid=cdi_proquest_ebookcentralchapters_5970556_25_29&context=PC&vid=44WHELF_CAR:44WHELF_CAR_VU1&lang=en&search_scope=CU_Search_ALL&adaptor=Primo%20Central&tab=CSCOP_EVERYTHING&query=any,contains,pseudo%20code&offset=0 (Accessed: 07 May 2023).

Ullman, L.E. and Liyanage, M. (2005) C programming, Library Search. Berkeley, CA: Peachpit Press. Available at: https://learning.oreilly.com/library/view/c-programming-visual/0321287630/0321287630_ch01.html (Accessed: 18 April 2023).

BIBLIOGRAPHY-

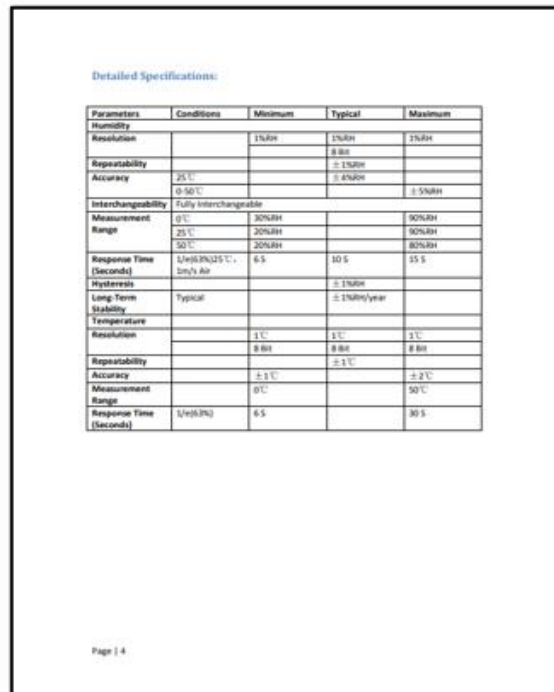
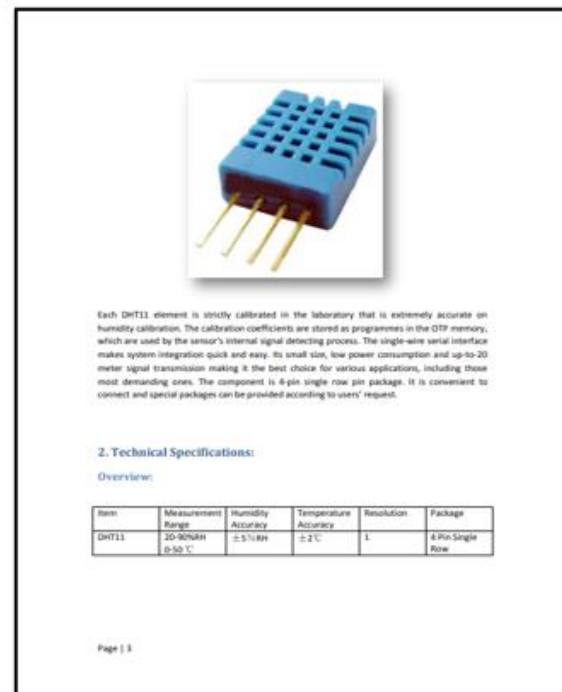
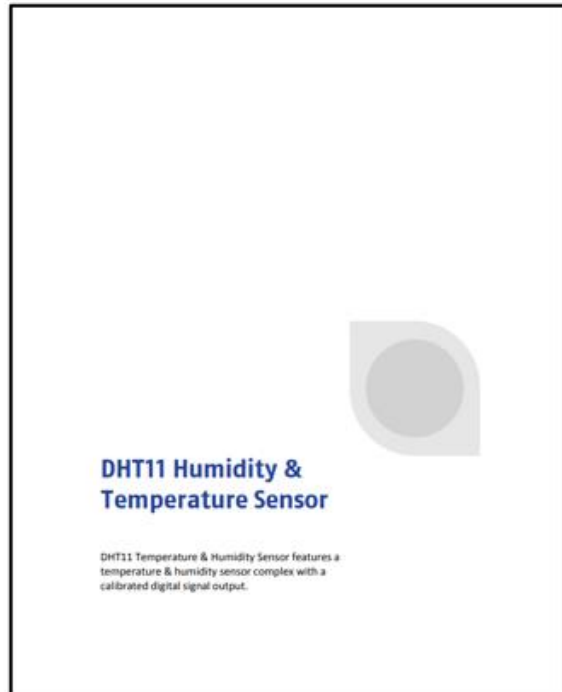
- Bird, J. (2014) “Resistor colour coding and ohmic value,” in Electrical Circuit Theory and Technology (Electrical Circuit Theory and Technology series). Routledge, pp. 669–669. Available at: <https://ebookcentral-proquest-com.pembrokeshire.idm.oclc.org/lib/pembrokeshire-ebooks/reader.action?docID=307211> (Accessed: May 11, 2023).
- Blum, R. (2014) Sam’s teach yourself Arduino programming in 24 hours, VLeBooks. Available at: <https://www.vlebooks.com> (Accessed: 12 May 2023).
- Deitel, P. (2016) C: How to Program, VLeBooks. Available at: <https://www.vlebooks.com> (Accessed: 11 May 2023).
- Harris, S.L. and Harris, D. (eds.) (2021) C Programming, Digital Design and Computer Architecture. Available at: <https://www.sciencedirect.com/science/article/pii/B9780128200643000179> (Accessed: 06 May 2023).
- Robertson, C.R. (2015) Fundamental Electrical and Electronic principles, ProQuest eBook Central. London: Routledge. Available at: <https://ebookcentral-proquest-com.pembrokeshire.idm.oclc.org/lib/pembrokeshire-ebooks/reader.action?docID=349341> (Accessed: April 19, 2023).
- Sebesta, R.W. (2016) Concepts of programming languages, VLeBooks. Available at: <https://www.vlebooks.com/Product/Index/751388?page=0&startBookmarkId=0> (Accessed: 07 May 2023).

EMBEDDED SYSTEMS – PCE 4004

APPENDIX-

Data Sheets

DHT11 Humidity & Temperature Sensor;



EMBEDDED SYSTEMS – PCE 4004

3. Typical Application (Figure 3)

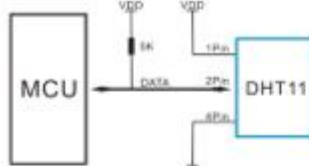


Figure 3 Typical Application

Note: 3Pin = 14.4, MCU = Microcontroller (either on single-chip Computer)

When the connecting cable is shorter than 20 metres, a 10k pull-up resistor is recommended, when the connecting cable is longer than 20 metres, choose an appropriate pull-up resistor as needed.

4. Power and Pin

DHT11's power supply is 3.3 to 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 VCC and GND for power filtering.

5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for characterization 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 40bits, and the sensor sends higher data 30bits.

Data format: Bit0 integral 8th data + Bit1 decimal 8th data + Bit2 integral 7th data + Bit3 decimal 7th data + Bit4 check sum. If the data transmission is right, the check sum should be the last Bit of "Bit0 integral 8th data + Bit1 decimal 8th data + Bit2 integral 7th data + Bit3 decimal 7th data".

Page | 6

5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Then can choose to collect (read) sensor data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

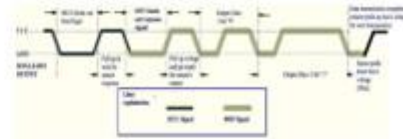


Figure 2 Overall Communication Process

5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Single-bus data format is used for characterization and synchronization between MCU and DHT11. When the communication between MCU and DHT11 begins, the programme of MCU will set data single-bus voltage level from high to low and this process must take at least 40ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40ms for DHT's response.

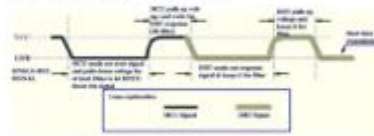


Figure 3 MCU Sends out Start Signal & DHT Response

Page | 6

5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80ms. Then the programme of DHT sets data single-bus voltage level from low to high and keeps it for 80ms for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80ms and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50ms low-voltage-level and the length of the following high-voltage level (signal determines) whether data bit is "0" or "1" (see Figures 4 and 5 below).

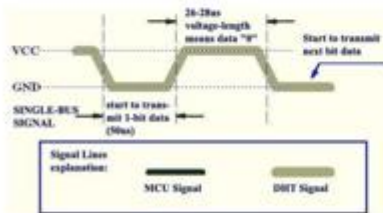


Figure 4 Data "0" Indication

Page | 7



Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage level, it suggests that DHT is not responding properly and please check the connection. When the last 8th data is transmitted, DHT11 pulls down the voltage level and keeps it for 80ms. Then the single-bus voltage will be pulled up by the resistor to set it back to the free status.

6. Electrical Characteristics

VCC=3.3V, T = 25 °C (unless otherwise stated)

Parameter	Symbol	Condition	Minimum	Typical	Maximum
Power Supply	VCC	3.3V	3.3V	3.3V	3.3V
Current	Measuring	100ms	100mA	100mA	100mA
Supply	Average	100ms	100mA	100mA	100mA
Sampling period	Second	1	1	1	1

Note: Sampling period at intervals should be no less than 1 second.

7. Attention of application

(1) Operating condition

Applying the DHT11 sensor beyond its working range stated in this datasheet can result in 100% signal drift/inaccuracy. The DHT11 sensor can recover to the calibrated status gradually when it gets back to the normal operating condition and works within its range. Please refer to [4] of

Page | 8

EMBEDDED SYSTEMS – PCE 4004

this section to accelerate its recovery. Please be aware that operating the DHT11 sensor in the non-normal working conditions will accelerate sensor's aging process.

(2) Attention to chemical materials

Vapor from chemical materials may interfere with DHT's sensitive elements and debase its sensitivity. A high degree of chemical contamination can permanently damage the sensor.

(3) Restoration process when (1) & (2) happen

Step one: Keep the DHT sensor at the condition of Temperature 50~60Celsius, humidity <30%RH for 2 hours;

Step two: Keep the DHT sensor at the condition of Temperature 20~30Celsius, humidity >70%RH for 5 hours.

(4) Temperature Affect

Relative humidity largely depends on temperature. Although temperature compensation technology is used to ensure accurate measurement of RH, it is still strongly advised to keep the humidity and temperature sensors working under the same temperature. DHT11 should be mounted at the place as far as possible from parts that may generate heat.

(5) Light Affect

Long time exposure to strong sunlight and ultraviolet may debase DHT's performance.

(6) Connection wires

The quality of connection wires will affect the quality and distance of communication and high quality shielding wire is recommended.

(7) Other attentions

* Welding temperature should be below 260Celsius and contact should take less than 30 seconds.

* Avoid using the sensor under dew condition.

* Do not use this product in safety or emergency stop devices or any other occasion that failure of DHT11 may cause personal injury.

* Storage: Keep the sensor at temperature 50~40°C, humidity <60%RH.

Disclaimer

This is a translated version of the manufacturer's data sheet. OSEPP is not responsible for the accuracy of the translated information.

Page | 9

Mouser Electronics

Authorized Distributor

[Click to View Pricing, Inventory, Delivery & Lifecycle Information:](#)

OSEPP Electronics:

[OSEPP](#)

Electronics, M. (2018) *DHT11 humidity & temperature sensor - Mouser Electronics, DHT!11 Humidity & Temperature Sensor*. Available at:
<https://www.mouser.com/datasheet/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf> (Accessed: 01 May 2023).

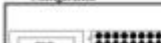
EMBEDDED SYSTEMS – PCE 4004

8X8 LED Dot Matrix Display;



General Description
The 8x8 DMM, 8x8 DMM and the 8x8 DMM are 8 by 8 dot matrix LED displays combined with a multiplexed drive, integrated circuit with a compatible serial interface. The displays are designed so that they can be mounted in a horizontal plane and can also be expanded in a vertical plane allowing versatile displays to be built. The cascade connection is also available. The displays are available in high efficiency red, green and yellow colour variants. LED sizes include (active display area) 3mm by 3mm, 3mm by 3mm and 3mm by 3mm.

The on-board driver chip has a serial input format that formats serial data, clock and chip enable. A single 5 volt supply is required. The serial bus operates at 100KHz levels and the format is compatible with SPI, I2C and I2C-BUS. It features a low level bus interface (I2C). The on-board driver chip allows the device to be programmed with row and column data thus allowing a wide range of characters to be generated. An external resistor sets the maximum brightness that can be digitally obtained to 16 levels. A test mode forces all LEDs on. Low power shutdown reduces current to 150µA.

Configuration	Dimensions			
	Rows (mm)	25.4	25.4	mm
A	2.54	2.54	2.54	
B	4.0	4.0	5.62	
C	10.0	10.0	10.0	
D	32.0	32.0	40.0	
E	2.75	5.0	5.0	
Depth	100.0	100.0	110.0	

EMBEDDED SYSTEMS – PCE 4004

The LED current is set to a maximum value (ie at its rated current). This provides the maximum level of brightness. The LED display controller also features a facility to 'latch' the display brightness from the host microcontroller. This control allows the microcontroller to maintain the segment current and thus the brightness.

Segment Brightness Parameter

Device Code	LED Cat	Red	Green	Blue	Max. Cath
15/11	Over 1.0mm	0	0	0	50mA
15/12	Over 1.0mm	0	0	0	50mA
15/13	Over 1.0mm	0	0	0	50mA
15/14	Over 1.0mm	0	0	0	50mA
15/15	Over 1.0mm	0	0	0	50mA
15/16	Over 1.0mm	0	0	0	50mA
15/17	Over 1.0mm	0	0	0	50mA
15/18	Over 1.0mm	0	0	0	50mA
15/19	Over 1.0mm	0	0	0	50mA
15/20	Over 1.0mm	0	0	0	50mA
15/21	Over 1.0mm	0	0	0	50mA
15/22	Over 1.0mm	0	0	0	50mA
15/23	Over 1.0mm	0	0	0	50mA
15/24	Over 1.0mm	0	0	0	50mA
15/25	Over 1.0mm	0	0	0	50mA
15/26	Over 1.0mm	0	0	0	50mA
15/27	Over 1.0mm	0	0	0	50mA
15/28	Over 1.0mm	0	0	0	50mA
15/29	Over 1.0mm	0	0	0	50mA
15/30	Over 1.0mm	0	0	0	50mA
15/31	Over 1.0mm	0	0	0	50mA
15/32	Over 1.0mm	0	0	0	50mA
15/33	Over 1.0mm	0	0	0	50mA
15/34	Over 1.0mm	0	0	0	50mA
15/35	Over 1.0mm	0	0	0	50mA
15/36	Over 1.0mm	0	0	0	50mA
15/37	Over 1.0mm	0	0	0	50mA
15/38	Over 1.0mm	0	0	0	50mA
15/39	Over 1.0mm	0	0	0	50mA
15/40	Over 1.0mm	0	0	0	50mA
15/41	Over 1.0mm	0	0	0	50mA
15/42	Over 1.0mm	0	0	0	50mA
15/43	Over 1.0mm	0	0	0	50mA
15/44	Over 1.0mm	0	0	0	50mA
15/45	Over 1.0mm	0	0	0	50mA
15/46	Over 1.0mm	0	0	0	50mA
15/47	Over 1.0mm	0	0	0	50mA
15/48	Over 1.0mm	0	0	0	50mA
15/49	Over 1.0mm	0	0	0	50mA
15/50	Over 1.0mm	0	0	0	50mA
15/51	Over 1.0mm	0	0	0	50mA
15/52	Over 1.0mm	0	0	0	50mA
15/53	Over 1.0mm	0	0	0	50mA
15/54	Over 1.0mm	0	0	0	50mA
15/55	Over 1.0mm	0	0	0	50mA
15/56	Over 1.0mm	0	0	0	50mA
15/57	Over 1.0mm	0	0	0	50mA
15/58	Over 1.0mm	0	0	0	50mA
15/59	Over 1.0mm	0	0	0	50mA
15/60	Over 1.0mm	0	0	0	50mA
15/61	Over 1.0mm	0	0	0	50mA
15/62	Over 1.0mm	0	0	0	50mA
15/63	Over 1.0mm	0	0	0	50mA
15/64	Over 1.0mm	0	0	0	50mA
15/65	Over 1.0mm	0	0	0	50mA
15/66	Over 1.0mm	0	0	0	50mA
15/67	Over 1.0mm	0	0	0	50mA
15/68	Over 1.0mm	0	0	0	50mA
15/69	Over 1.0mm	0	0	0	50mA
15/70	Over 1.0mm	0	0	0	50mA
15/71	Over 1.0mm	0	0	0	50mA
15/72	Over 1.0mm	0	0	0	50mA
15/73	Over 1.0mm	0	0	0	50mA
15/74	Over 1.0mm	0	0	0	50mA
15/75	Over 1.0mm	0	0	0	50mA
15/76	Over 1.0mm	0	0	0	50mA
15/77	Over 1.0mm	0	0	0	50mA
15/78	Over 1.0mm	0	0	0	50mA
15/79	Over 1.0mm	0	0	0	50mA
15/80	Over 1.0mm	0	0	0	50mA
15/81	Over 1.0mm	0	0	0	50mA
15/82	Over 1.0mm	0	0	0	50mA
15/83	Over 1.0mm	0	0	0	50mA
15/84	Over 1.0mm	0	0	0	50mA
15/85	Over 1.0mm	0	0	0	50mA
15/86	Over 1.0mm	0	0	0	50mA
15/87	Over 1.0mm	0	0	0	50mA
15/88	Over 1.0mm	0	0	0	50mA
15/89	Over 1.0mm	0	0	0	50mA
15/90	Over 1.0mm	0	0	0	50mA
15/91	Over 1.0mm	0	0	0	50mA
15/92	Over 1.0mm	0	0	0	50mA
15/93	Over 1.0mm	0	0	0	50mA
15/94	Over 1.0mm	0	0	0	50mA
15/95	Over 1.0mm	0	0	0	50mA
15/96	Over 1.0mm	0	0	0	50mA
15/97	Over 1.0mm	0	0	0	50mA
15/98	Over 1.0mm	0	0	0	50mA
15/99	Over 1.0mm	0	0	0	50mA
15/100	Over 1.0mm	0	0	0	50mA

The display also features a write latch register that sets how many columns are displayed from 1 to 4. These are displayed in a multiplexed manner with a typical display write rate of 80Hz. Since the number of digits displayed affects the overall display intensity, the write latch register should be set to 4.

WARNING: Exercise care to ensure the driver chip does not dissipate excessive power.

Using the device in elevated temperatures without decreasing the driver chip dissipation may result in damage to the device. This can be done by reducing the brightness at elevated temperatures.

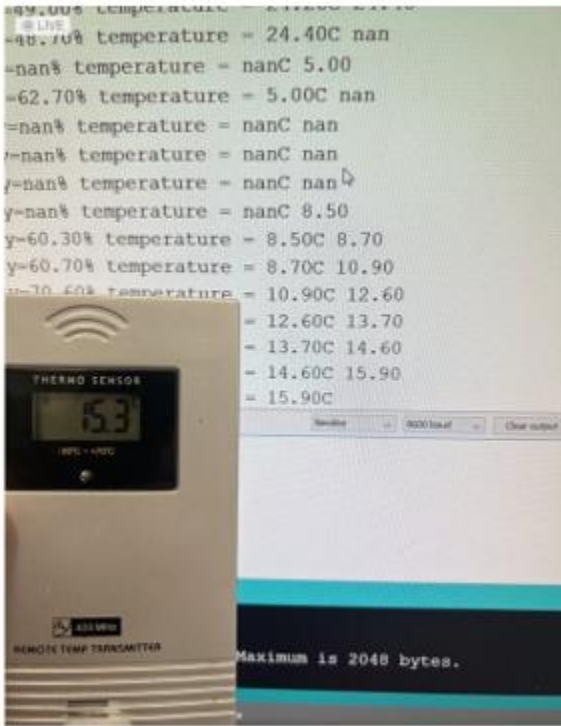
Write Latch Register Format

Write Latch	100-000	101	100	100	Max. Cath
Column 0	Over 1.0mm	0	0	0	50mA
Column 1	Over 1.0mm	0	0	0	50mA
Column 2	Over 1.0mm	0	0	0	50mA
Column 3	Over 1.0mm	0	0	0	50mA
Column 4	Over 1.0mm	0	0	0	50mA
Column 5	Over 1.0mm	0	0	0	50mA
Column 6	Over 1.0mm	0	0	0	50mA
Column 7	Over 1.0mm	0	0	0	50mA
Column 8	Over 1.0mm	0	0	0	50mA
Column 9	Over 1.0mm	0	0	0	50mA
Column 10	Over 1.0mm	0	0	0	50mA
Column 11	Over 1.0mm	0	0	0	50mA
Column 12	Over 1.0mm	0	0	0	50mA
Column 13	Over 1.0mm	0	0	0	50mA
Column 14	Over 1.0mm	0	0	0	50mA
Column 15	Over 1.0mm	0	0	0	50mA
Column 16	Over 1.0mm	0	0	0	50mA
Column 17	Over 1.0mm	0	0	0	50mA
Column 18	Over 1.0mm	0	0	0	50mA
Column 19	Over 1.0mm	0	0	0	50mA
Column 20	Over 1.0mm	0	0	0	50mA
Column 21	Over 1.0mm	0	0	0	50mA
Column 22	Over 1.0mm	0	0	0	50mA
Column 23	Over 1.0mm	0	0	0	50mA
Column 24	Over 1.0mm	0	0	0	50mA
Column 25	Over 1.0mm	0	0	0	50mA
Column 26	Over 1.0mm	0	0	0	50mA
Column 27	Over 1.0mm	0	0	0	50mA
Column 28	Over 1.0mm	0	0	0	50mA
Column 29	Over 1.0mm	0	0	0	50mA
Column 30	Over 1.0mm	0	0	0	50mA
Column 31	Over 1.0mm	0	0	0	50mA
Column 32	Over 1.0mm	0	0	0	50mA
Column 33	Over 1.0mm	0	0	0	50mA
Column 34	Over 1.0mm	0	0	0	50mA
Column 35	Over 1.0mm	0	0	0	50mA
Column 36	Over 1.0mm	0	0	0	50mA
Column 37	Over 1.0mm	0	0	0	50mA
Column 38	Over 1.0mm	0	0	0	50mA
Column 39	Over 1.0mm	0	0	0	50mA
Column 40	Over 1.0mm	0	0	0	50mA
Column 41	Over 1.0mm	0	0	0	50mA
Column 42	Over 1.0mm	0	0	0	50mA
Column 43	Over 1.0mm	0	0	0	50mA
Column 44	Over 1.0mm	0	0	0	50mA
Column 45	Over 1.0mm	0	0	0	50mA
Column 46	Over 1.0mm	0	0	0	50mA
Column 47	Over 1.0mm	0	0	0	50mA
Column 48	Over 1.0mm	0	0	0	50mA
Column 49	Over 1.0mm	0	0	0	50mA
Column 50	Over 1.0mm	0	0	0	50mA
Column 51	Over 1.0mm	0	0	0	50mA
Column 52	Over 1.0mm	0	0	0	50mA
Column 53	Over 1.0mm	0	0	0	50mA
Column 54	Over 1.0mm	0	0	0	50mA
Column 55	Over 1.0mm	0	0	0	50mA
Column 56	Over 1.0mm	0	0	0	50mA
Column 57	Over 1.0mm	0	0	0	50mA
Column 58	Over 1.0mm	0	0	0	50mA
Column 59	Over 1.0mm	0	0	0	50mA
Column 60	Over 1.0mm	0	0	0	50mA
Column 61	Over 1.0mm	0	0	0	50mA
Column 62	Over 1.0mm	0	0	0	50mA
Column 63	Over 1.0mm	0	0	0	50mA
Column 64	Over 1.0mm	0	0	0	50mA
Column 65	Over 1.0mm	0	0	0	50mA
Column 66	Over 1.0mm	0	0	0	50mA
Column 67	Over 1.0mm	0	0	0	50mA
Column 68	Over 1.0mm	0	0	0	50mA
Column 69	Over 1.0mm	0	0	0	50mA
Column 70	Over 1.0mm	0	0	0	50mA
Column 71	Over 1.0mm	0	0	0	50mA
Column 72	Over 1.0mm	0	0	0	50mA
Column 73	Over 1.0mm	0	0	0	50mA
Column 74	Over 1.0mm	0	0	0	50mA
Column 75	Over 1.0mm	0	0	0	50mA
Column 76	Over 1.0mm	0	0	0	50mA
Column 77	Over 1.0mm	0	0	0	50mA
Column 78	Over 1.0mm	0	0	0	50mA
Column 79	Over 1.0mm	0	0	0	50mA
Column 80	Over 1.0mm	0	0	0	50mA
Column 81	Over 1.0mm	0	0	0	50mA
Column 82	Over 1.0mm	0	0	0	50mA
Column 83	Over 1.0mm	0	0	0	50mA
Column 84	Over 1.0mm	0	0	0	50mA
Column 85	Over 1.0mm	0	0	0	50mA
Column 86	Over 1.0mm	0	0	0	50mA
Column 87	Over 1.0mm	0	0	0	50mA
Column 88	Over 1.0mm	0	0	0	50mA
Column 89	Over 1.0mm	0	0	0	50mA
Column 90	Over 1.0mm	0	0	0	50mA
Column 91	Over 1.0mm	0	0	0	50mA
Column 92	Over 1.0mm	0	0	0	50mA
Column 93	Over 1.0mm	0	0	0	50mA
Column 94	Over 1.0mm	0	0	0	50mA
Column 95	Over 1.0mm	0	0	0	50mA
Column 96	Over 1.0mm	0	0	0	50mA
Column 97	Over 1.0mm	0	0	0	50mA
Column 98	Over 1.0mm	0	0	0	50mA
Column 99	Over 1.0mm	0	0	0	50mA
Column 100	Over 1.0mm	0	0	0	50mA
Column 101	Over 1.0mm	0	0	0	50mA
Column 102	Over 1.0mm	0	0	0	50mA
Column 103	Over 1.0mm	0	0	0	50mA
Column 104	Over 1.0mm	0	0	0	50mA
Column 105	Over 1.0mm	0	0	0	50mA
Column 106	Over 1.0mm	0	0	0	50mA
Column 107	Over 1.0mm	0	0	0	50mA
Column 108	Over 1.0mm	0	0	0	50mA
Column 109	Over 1.0mm	0	0	0	50mA
Column 110	Over 1.0mm	0	0	0	50mA
Column 111	Over 1.0mm	0	0	0	50mA
Column 112	Over 1.0mm	0	0	0	50mA
Column 113	Over 1.0mm	0	0	0	50mA
Column 114	Over 1.0mm	0	0	0	50mA
Column 115	Over 1.0mm	0	0	0	50mA
Column 116	Over 1.0mm	0	0	0	50mA
Column 117	Over 1.0mm	0	0	0	50mA
Column 118	Over 1.0mm	0	0	0	50mA
Column 119	Over 1.0mm	0	0	0	50mA
Column 120	Over 1.0mm	0	0	0	50mA
Column 121	Over 1.0mm	0	0	0	50mA
Column 122	Over 1.0mm	0	0	0	50mA
Column 123	Over 1.0mm	0	0	0	50mA
Column 124	Over 1.0mm	0	0	0	50mA
Column 125	Over 1.0mm	0	0	0	50mA
Column 126	Over 1.0mm	0	0	0	50mA
Column 127	Over 1.0mm	0	0	0	50mA
Column 128	Over 1.0mm	0	0	0	50mA
Column 129	Over 1.0mm	0	0	0	50mA

EMBEDDED SYSTEMS – PCE 4004

Testing Evidence

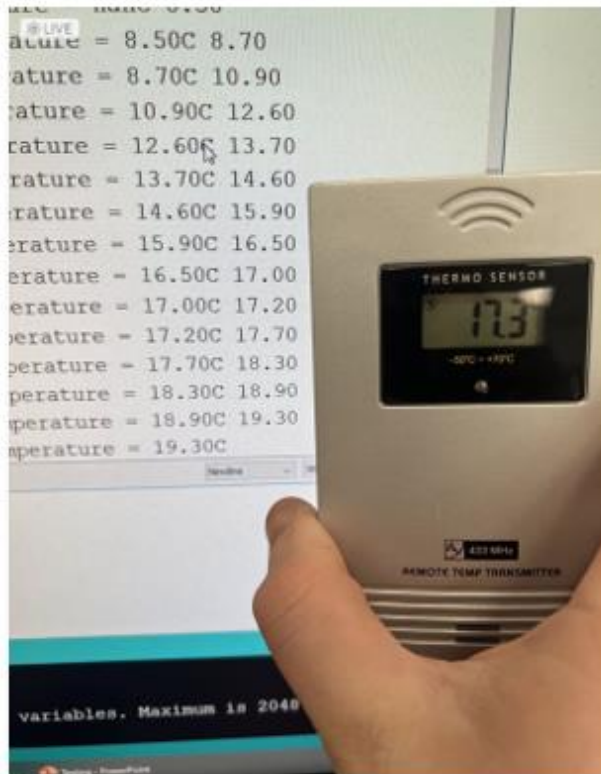
Test 1-



These pictures above show the output display result when both the DHT11 and the pre-existing room thermostat measure $Temp \leq 16$. It also shows the real time measure value through the programming software “Arduino IDE”.

EMBEDDED SYSTEMS – PCE 4004

Test 2-



These pictures above show the output display result when both the DHT11 and the pre-existing room thermostat measure $Temp \geq 17$ & $Temp \leq 20$. It also shows the real time measure value through the programming software “Arduino IDE”

EMBEDDED SYSTEMS – PCE 4004

Test 3-



These pictures above show the output display result when the both the DHT11 and the pre-existing room thermostat measure $\text{Temp} \geq 20$. It also shows the real time measure value through the programming software “Arduino IDE”

EMBEDDED SYSTEMS – PCE 4004

Figures

Figure 1- Store, A. (2018) *Arduino Uno REV3*, *Arduino Official Store*. Available at: <https://store.arduino.cc/products/arduino-uno-rev3> (Accessed: 12 May 2023).

Figure 2- Produced in “Fritzing” software, accessed through Pembroke College “Zen Works”, (by Samuel Davies on 20 April, 2023)

Figure 3- Produced in “Microsoft Visio” software, accessed through Pembroke College “Office 365”, (by Samuel Davies on 24 April, 2023)