UOW MALAYSIA KDU PENANG UNIVERSITY COLLEGE

PART OF THE UNIVERSITY OF WOLLONGONG AUSTRALIA GLOBAL NETWORK

ASSIGNMENT COVER PAGE

Validated by: UNIVERSITY OF LINCOLN UNITED KINGDOM

Programme	**Course code and title**
Bachelor of Computer Science (Hons)/ Bachelor of Computer Science (Hons) in Computer and Network Technology/ Bachelor of Information Systems (Hons)/ Bachelor of Software Engineering (Hons)	CET3063/N/CET3064 Internet of Things
Student's name / student's ID	**Lecturer's name**
0204677 Lim Zhe Yuan	Dr. Khoo Hee Kooi

Date issued	**Submission deadline**	**Indicative weighting**
13rd February 2023 (Week 3)	6th March 2023 (Week 6)	30%
Assignment 1 title	Implement an IoT system (400 words)	

This assessment assesses the following course learning outcomes

# as in course guide	**UOW Malaysia KDU Penang University College learning outcomes**
CLO1	Review the various components of Internet of Things (IoT) at various scales.
CLO2	Evaluate and design IoT system architecture for a real world application.
CLO3	N/A
CLO4	N/A
# as in course guide	**University of Lincoln learning outcomes**
CLO1	Critically evaluate the strengths, weaknesses, and resource constraints of IoT computing systems in comparison to traditional computing models
CLO2	Design and implement a connected prototype IoT software system that utilises sensor data
CLO3	Implement industry standard IoT messaging protocols
CLO4	N/A

Student's declaration

I certify that the work submitted for this assignment is my own and research sources are fully acknowledged.

Student's signature: Submission date: **6 March 2023**

Zhe Yuan

TurnItIn Similarity Report

Asgn1

ORIGINALITY REPORT

0%	0%	0%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

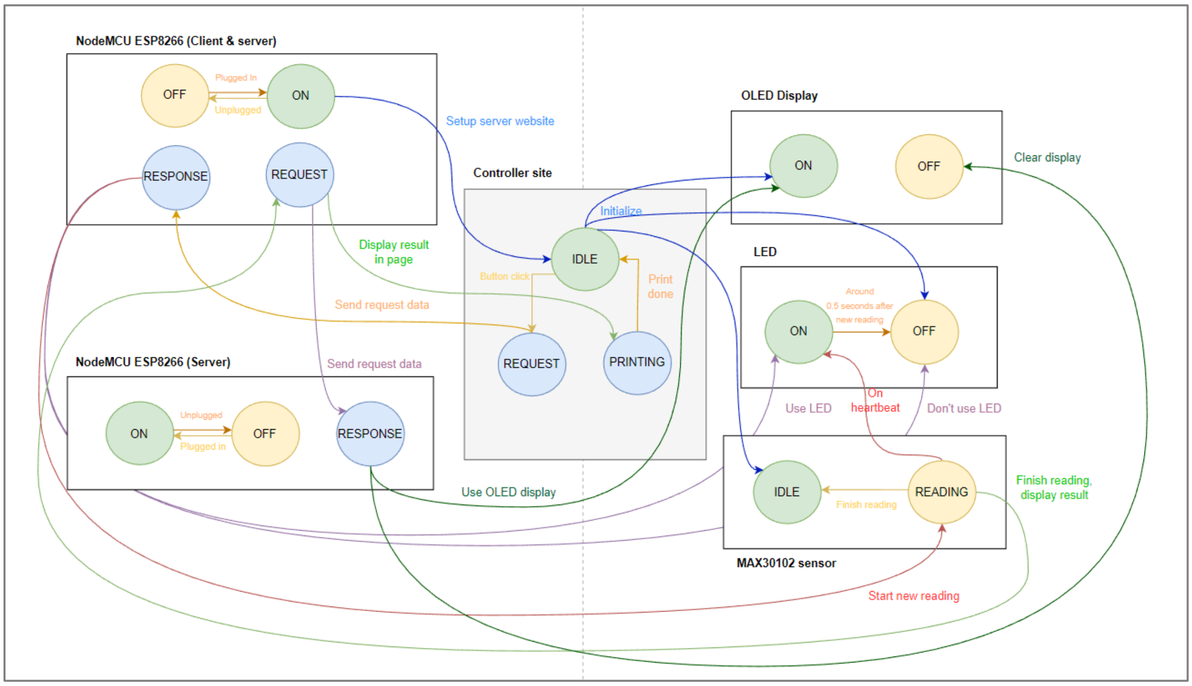
Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

Table of Contents

Main Report

Question 1

- **Finite state machine (FSM) diagram**



When plugged in, initialization code is run to establish connections between the microcontroller units (MCU) and components such as MAX30102 sensor, red LED, and OLED display to create the **sensor layer**. At the same time, both MCUs are configured to serve as servers as well. While one of the MCU is responsible for serving the display, another MCU is tasked to serve the application platform. To enable the usage of the display, the application MCU is also used to connect to the display MCU server as a client via Arduino WiFiClient methods to enable communication between the display MCU and the application MCU, establishing the **network layer**. After these processes are finished, users can access the application platform by entering the IP address of the platform server in a browser. All user interactions are handled on the platform. Therefore, the **platform layer** has also been configured to allow users to request for different actions from both servers through HTTP queries when required. It processes user request inputs and handles different user requests with grace to avoid generating errors when users use the system. The use of all three layers mentioned previously allows the system to store heart rate data and display them accurately and interactively for user convenience, providing a feasible **application layer** for system users.

- **Functional requirements**

A few requirements have been considered for the heart rate monitoring system. The following explains these requirements and justifies their importance to the system.

1. Take user heart rate readings

Users must be able to take their heart rate readings as it is the main requirement that is essential for a heart rate monitoring system. Heart rate readings should be taken using the given formula, *bpm = 60(c - l) / 1000*, where c is the current time and l is the time from the previous beat. At least 10 heart rate readings should be taken to be able to derive a more accurate average heart rate, and the outputs from using the given formula should be sensible for a typical user.

2. Enable users to configure component usage

Users should be able to toggle component usages such as the LED and OLED display according to their preferences using a settings panel. This feature should be added for the system as not all users will require the use of these components and they might want to turn them off. This feature will also help prevent the activation of certain syndromes that react to light, such as epileptic seizures.

3. Allow users to retake another heart rate reading after finishing one.

Users should be able to retake another heart rate reading immediately after completing one. It allows the user to use shorter time to restart the process instead of going back to the initial landing page. When using this feature, the new process can be done in a new tab as well so old readings can remain available for the users to compare results later on.

- **Non-functional requirements**

When developing the system's requirements, some non-functional requirements are also taken into consideration to provide a pleasant user experience to the user when using the system.

1. Feasibility of starting heart rate readings

Users should be able to start heart rate readings easily without going through too many user interaction processes. It helps reduce the user's effort and time to perform the main task, which will eventually lead to increased user reliance on the system because the system leaves a good impression on the users as a very efficient system.

2. Visibility of the heart rate reading process

Users should be able to see the exact process of an ongoing heart rate reading. It makes it easier for users to recognize the system's status and allows them to identify problems in the process when they find them taking unusually long.

3. Responsive interfaces

Application interfaces should be responsive to system changes. This increases the interactivity of the application with the user and creates a sense of realism as virtual displays are being rendered off from real world data. Users may also find the interface to be less laggy or buggy if interfaces are responsive to their modifications instantly.

- **Bill of Materials (BOM)**

No.	Item name	Unit Price	Quantity	Total Price
1	MAX30102 heartbeat sensor	13.00	1	13.00
2	Breadboards	0.75	2	1.50
3	Red LED (1.8 – 2.2V)	1.20	1	1.20
4	Resistor (150ohm)	1.00	1	1.00
5	Jumper wires, male to female (40pc per pack)	3.60	1	3.60
6	OLED display (0.96", I2C)	12.50	1	12.50
7	NodeMCU (ESP8266)	19.90	2	39.80
				72.60#

Each item in the BOM were bought to serve a purpose in the proposed heart rate monitoring system. A MAX30102 heartbeat sensor was procured as it is the main sensor that obtains heart rate data. MAX30102 sensor is chosen over MAX30100 sensor because it is more sensitive to IR receiver voltage changes, has lower power consumption, and has more data storage which results in quicker data transmission (Goyal, 2021). An OLED display was also bought to display heart rate calculation results to give visual feedback from the microcontroller unit to the user.

Additionally, a red LED was bought to simulate the heartbeat effect when the heart rate reading process is ongoing. It allows users to know the status and progress of a reading session. The LED is paired with a resistor to lower down the voltage supplied to the LED as the LED only accepts around 1.8 – 2.2V compared to the microcontroller unit's 3.3V. 2 breadboards were also acquired to lock these components in place and prevent them from hanging loosely during the handling of these components.

To connect these components together on breadboards, male-to-female jumper wires were also bought to configure the pinouts of the components to the microcontroller units. Although the jumper wires that were bought come in 40 pieces per pack, only 10 of them are used in the developed heart rate monitoring system. This is because 8 of them are used to connect the MAX30102 sensor and OLED display that use the I2C communication interface to the microcontroller units as they require jumper wires for their serial data (SDA), serial clock (SCL), current input (VCC), and ground (G) pins, and the other 2 wires are used to supply current to the LED using the 3.3V and ground pins of a microcontroller unit.

Lastly, 2 NodeMCU ESP8266's were also bought to implement a client-server connection between both microcontroller units. Both microcontroller units are servers that serves the main application platform and the OLED display function respectively, and the microcontroller unit that serves the platform connects to the OLED display server as a client to allow it to send requests to the display server and modify the state of the display. This enables system code to be distributed between the two microcontroller units and reduces the chances of crowding wires on a single microcontroller unit, which leads to the need to reconfigure wiring methods.

Question 2

- **Demonstration video link**

https://drive.google.com/file/d/1aRgX0fSjHNNKOXKCLcGCAKYiK-8uoRio/view?usp=share_link

- **Arduino sketch link**

https://drive.google.com/drive/folders/1qCN6IXYi825VTxvTzWucw3ElGslKID8p?usp=share_link

References

Goyal, A. (2021) *Difference between Max30100 and Max30102 Sensor?* Available at https://www.hnhcart.com/blogs/sensors-modules/a [Accessed 17 March 2023].

Microcontrollerslab (2023) *MAX30102 Pulse Oximeter and Heart Rate Sensor with ESP8266*. Available at https://microcontrollerslab.com/esp8266-heart-rate-pulse-oximeter-max30102/ [Accessed 15 March 2023].

CET3063/N/CET3064 Internet of Things MARKING RUBRIC Assignment 1 Implement an IoT system (Weighted marks: 30%)							
Question 1: Reviews on an IoT system (Score: 50%)							
LEARNING OUTCOME	**MARKING CRITERIA**	**SCALE**					
		Failed (0% to 49%)	**3rd class (50% to 59%)**	**2nd lower (60% to 69%)**	**2nd upper (70% to 79%)**	**1st class (80% to 100%)**	**YOUR MARKS/COMMENTS**
CLO1: Review the various components of Internet of Things (IoT) at various scales.	**1(a) Finite-state machine (FSM) (10%)**	No FSM has illustrated for the IoT system.	Brief FSM has drawn for the IoT system. There are major flaws in the FSM.	Good FSM has drawn for the IoT system with minor flaws, however lack of technical details.	Good FSM has drawn for the IoT system with appropriate concepts and technical details are stated.	Excellent FSM has drawn for the IoT system with proper concepts and in-depth technical details are stated.	
	1(b) Sensors justifications (10%)	No discussion on the sensors involved.	Brief discussion on the sensors involved and there are misconceptions for certain terms.	Good discussion on the sensors involved, however lack of technical details.	Good discussion on the sensors involved with proper concepts and technical details are stated.	Excellent discussion on the sensors involved with proper concepts and in-depth technical details are stated.	
	1(c) Functional requirements (20%)	No discussion on the functional requirements for the IoT system.	Brief discussion on the functional requirements for the IoT system. There are misconceptions in the discussion.	Good discussion on the functional requirements for the IoT system, however lack of technical justifications.	Good discussion on the functional requirements for the IoT system with proper concepts and technical details are stated.	Excellent discussion on the functional requirements for the IoT system with proper concepts and in-depth technical details are stated.	
	1(d) Non-functional requirements (10%)	No discussion on the non-functional requirements for the IoT system.	Brief discussion on the non-functional requirements for the IoT system. There are misconceptions in the discussion.	Good discussion on the non-functional requirements for the IoT system, however lack of technical justifications.	Good discussion on the non-functional requirements for the IoT system with proper concepts and technical details are stated.	Excellent discussion on the non-functional requirements for the IoT system with proper concepts and in-depth technical details are stated.	
	Total (50%)						

Question 2: Develop an IoT system (Score: 50%)							
LEARNING OUTCOME	**MARKING CRITERIA**	**SCALE**					
		Failed (0% to 49%)	**3rd class (50% to 59%)**	**2nd lower (60% to 69%)**	**2nd upper (70% to 79%)**	**1st class (80% to 100%)**	**YOUR MARKS/COMMENTS**
CLO2: Evaluate and design IoT system architecture for a real world application.	**2(a) Sensor layer (10%)**	No development on sensor layer.	Erroneous in the program code for sensor layer, however there are some relevant codes are implemented.	Good developments of sensor layer with error free. However, there are some bugs occurred during runtime.	Good developments of sensor layer with error free. There are shortcomings for the flow during runtime.	Outstanding developments of sensor layer with error free. Efficient process flow during runtime.	
	2(b) Network layer (10%)	No development on network layer.	Erroneous in the program code for network layer, however there are some relevant codes are implemented.	Good developments of network layer with error free. However, there are some bugs occurred during runtime.	Good developments of network layer with error free. There are shortcomings for the flow during runtime.	Outstanding developments of network layer with error free. Efficient process flow during runtime.	
	2(c) Platform layer (10%)	No development on platform layer.	Erroneous in the program code for platform layer, however there are some relevant codes are implemented.	Good developments of platform layer with error free. However, there are some bugs occurred during runtime.	Good developments of platform layer with error free. There are shortcomings for the flow during runtime.	Outstanding developments of platform layer with error free. Efficient process flow during runtime.	
	2(d) Application layer (20%)	No development on application layer.	Erroneous in the program code for application layer, however there are some relevant codes are implemented.	Good developments of application layer with error free. However, there are some bugs occurred during runtime.	Good developments of application layer with error free. There are shortcomings for the flow during runtime.	Outstanding developments of application layer with error free. Efficient process flow during runtime.	
	Total (50%)						
Overall score (100%)							
Weighted marks (30%)							