


<div><div><div>RV</div><div>UNIVERSITY</div><div>Go, change the world</div><div>an initiative of RV EDUCATIONAL INSTITUTIONS</div></div></div>		USN <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div> <div></div>													
		RV University, Bengaluru School of Computer Science and Engineering B.Tech (Hons.) CP-1 Question Paper – Set1 with Answer keys Academic Year 2024-2025													
		Course: IoT and Edge Computing						Course Code: CS3100				Semester: V			
		Date: 3 Sep 2024		Duration: 90 minutes				Max Marks: 20							
Sl. No.	Questions								Marks	L1-L6		CO			
1.	Draw the layered architecture of IoT and give a short description of each of the layers mentioning their responsibilities.								2	L2		1			
2.	What is the ISM band? Is it licensed or unlicensed? Give a couple of wireless protocols that work on this band.								2	L2		1			
3.	Write a note on 5G and explain the roles of front haul and backhaul in network architecture and evaluate how each contributes to overall network performance in 5G and its use cases in the IoT domain.								2	L5		1			
4.	Consider the following C code which is executed on a 32-bit system. Assume that the variable num is located in the memory address 0x00A1234 . int main() { int num = 0x10; int* iPtr; iPtr = # iPtr++; num++; printf("The values in iPtr = %p and num = %d\n", iPtr, num); return 0; } What will be printed by the program when it runs?								2	L4		2			
5.	L1 = [1, 4, 'Hi', 5] L2 = [5.0, 2.3, 'Hello', 4.0] for elem in L1: for x in L2: if type(elem) == type(x): print(elem, x) What will be printed by this program when it runs?								2	L4		2			

6.	<p>Conisder the definition of union type below.</p> <pre>typedef union { int i; float f; char c[2]; } MY_DATA; MY_DATA myData;</pre> <p>a. What is the size of myData in a 32-bit system?</p> <p>b. If myData.i = 0x41424344, what will be the values of myData.c[0], myData.c[1], myData.c[2], and myData.c[3] assuming little-endian memory representation?</p>	2	L3	2
7.	<p>What is the differene between the APIs xTaskCreate() and xTaskCreatePinnedToCore()?</p> <p>Give sample invocation of these APIs with suitable parameters and illustrate the role of stack size as one of the paarmeters to the above APIs</p>	2	L5	3
8.	Compare and Contrast the key differences between NB-IoT and LTE-M in terms of latency, throughput, and deployment?	1	L4	1
9.	Write a single line note on the four characteristics of WSN.	1	L2	1
10	<pre>void myFn(void) { char charArr[] = "ABCDE"; int num1 = charArr[2] - charArr[1]; int num2 = charArr[4] - charArr[1]; printf("num1 = %d and num2 = %d\n", num1, num2); }</pre> <p>What will be printed when the above function executes?</p>	1	L4	2
11	<p>Write a Python program to flatten the given matrix by using comprehension and the output should be the square of the individual elements.</p> <pre>myMatrix = [[1, 2, 3], [4, 5], [6, 7, 8, 9, 10]]</pre>	1	L6	2
12	What is the OS running on ESP32 (lab execrcises) ? How it is different from any other commercial OS's like Windows/ Linux ?	1	L2	3
13	Write a short note on each of the macros used to access variable parameters passed to a function in a C program.	1	L2	2

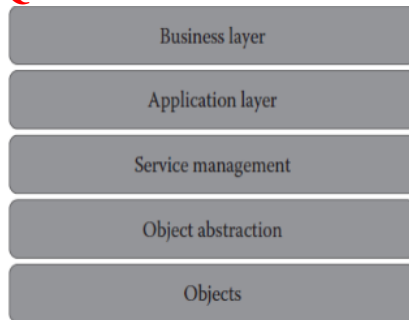
Course Outcomes

1. Choose a suitable wireless protocol based on the problem domain of an IoT product
2. Understand NumPy, TensorFlow framework and the need for TFLite for Edge Devices
3. Identify an IoT solution based on the features supported by ESP32 and FreeRTOS
4. Demonstrate integration of cloud computing platforms with ESP32

Marks Distribution									
L1	L2	L3	L4	L5	L6	CO1	CO2	CO3	CO4
0	7	2	6	4	1	8	9	3	0

Answers

Q1.



Short description about each layer and its responsibilities. You can also draw the below diagram with a short note on each layer.

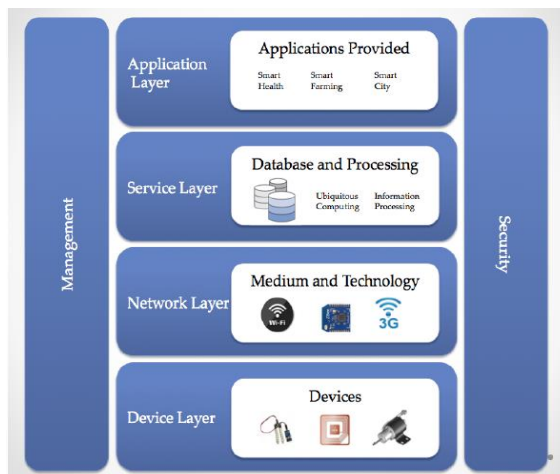


Diagram-1 mark

Explanation-1 mark

Q2. Industrial, scientific, and medical (ISM) bands are radio frequency (RF) bands that are reserved for use in science, industry, and medicine, rather than telecommunications.

They are unlicensed frequency spectrum.

WiFi, Bluetooth, UWB, Zigbee, NFC and RFID are the wireless protocols using ISM band and they are of LAN and PAN type networks.

Q3. 5G features-MU-MIMO, Beamforming, data rate and latency

Backhaul connects the mobile network to the wired network for large data transfer, while fronthaul connects remote cell sites to the baseband processing unit (BBU) for higher data speed.

Q4. The values in iPtr = 00A1238 and num = 17

Q5. Hi Hello

Q6. Union allocates the maximum size required by the element within in it. Since it is mentioned that it is

on 32-bit system, int here will be of 4 bytes and float is always four bytes.

Size of myData = 4 bytes – 1 mark

Assuming little endian machine, the lower address or c[0] and c[1] are the valid elements since the given char array is only of two bytes. Accessing myData.c[2] and c[3] will give a compilation error. But the memory contents will be as shown below.

myData.c[0] = 44, myData.c[1] = 43, myData.c[2] = 42 and myData.c[3] = 41 – 1 mark

Note: No marks will be reduced even if compilation error is not mentioned.

Q7. xTaskCreate() creates task that can be scheduled on any available core and **xTaskCreatePinnedToCore()** creates task that needs to be run on a specific core (last parameter)

```
void setup() {  
    // Create a task  
    xTaskCreate(  
        vTaskCode,      /* Task function. */  
        "Task 1",        /* Name of the task. */  
        1000,            /* Stack size in words, not bytes. */  
        NULL,            /* Task input parameter. */  
        1,               /* Priority of the task. */  
        NULL             /* Task handle. */  
    );  
}  
  
// Create a task pinned to core 1  
xTaskCreatePinnedToCore(  
    vTaskCode,          /* Task function. */  
    "Task 2",           /* Name of the task. */  
    1000,               /* Stack size in words, not bytes. */  
    NULL,               /* Task input parameter. */  
    1,                 /* Priority of the task. */  
    NULL,               /* Task handle. */  
    1                  /* Core on which the task should run. */  
);  
}
```

The stack size parameter is essential for defining how much memory is allocated to a task's stack. This allocation is crucial for the stability and reliability of task execution, as it directly impacts the task's ability to handle local data and function calls without running into stack overflow issue. – **1 Mark**

Q8.

- ☐ **Latency:** LTE-M offers lower latency compared to NB-IoT, making it better for real-time applications.
- ☐ **Throughput:** LTE-M provides higher throughput, suitable for applications requiring more data transmission, whereas NB-IoT is designed for lower throughput needs.
- ☐ **Deployment:** NB-IoT is optimized for deep coverage and high device density, while LTE-M is more suitable for applications requiring mobility and higher data rates.

Q9. WSN: Wireless Sensor Networks.

The four characteristics are: Redundancy Exploitation, Data-centric Routing, Data Aggregation and Localized Algorithms. Write a single line note on each.

Q10. num1 = 1 and num2 = 3 - 0.5 marks each

Q11. # Flatten it using comprehension and get the square of each element

flatMatrix = [x*x for row in myMatrix for x in row]

Q12. FreeRTOS is the Real Time OS running on ESP32.

FreeRTOS is an open-source Real-Time Operating System (RTOS) designed for embedded systems. It provides multitasking capabilities, allowing multiple tasks to run concurrently on a single microcontroller. This is highly responsive to events and predictable latencies involved in its execution compared to other commercial OS such as Windows and Linux. FreeRTOS is also having a smaller footprint in terms of its memory requirements and also has simpler structure in terms of process/task management. It does not maintain parent-child relationship as well as concept of threads running within the context of tasks or processes.

Note: Need to mention any two of the above need to be mentioned. Additional points which are valid will also be considered for evaluation.


Q13. va_start(), va_arg() and va_end()

va_list args; - Variable of type va_list is declared to hold the starting address of arguments.

va_start(args, fmt); - Initializes the starting addr of list of variables passed into the function

va_arg(args, type) accesses the argument of a particular type by updating the argument list

va_end(args); - Clears the list created above.

 RV UNIVERSITY <i>Go, change the world</i> <small>an initiative of RV EDUCATIONAL INSTITUTIONS</small>		USN <table border="1"><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr></table>																									
<p align="center">RV University, Bengaluru School of Computer Science and Engineering B.Tech (Hons.) CP-1 Question Paper – Set2 with Answer keys Academic Year 2024-2025</p>																											
Course: IoT and Edge Computing						Course Code: CS3100				Semester: V																	
Date: 3 Sep 2024			Duration: 90 minutes			Max Marks: 20																					
Sl. No.	Questions							Marks	L1-L6		CO																
1.	Draw the layered architecture of IoT and give a short description of each of the layers mentioning their responsibilities.							2	L2		1																
2.	Consider a smart home scenario where the operation of a fan is controlled based on temperature and humidity levels. Traditionally, a simple rule-based system might turn the fan on or off when the temperature exceeds a certain threshold and the humidity falls below a specific level. How could Artificial Intelligence (AI) and Machine Learning (ML) enhance this system in Improving the accuracy and responsiveness of the fan control system?							2	L5		1																
3.	With a neat diagram explain how messages are transmitted in AMQP using publish/subscribe model.							2	L2		1																
4.	Consider the following C code, which is executed on a 64-bit system. Assume that the variable data is located at the memory address 0x7FFFC1234560. int main() { float data = 3.14; float* fPtr; fPtr = &data; fPtr++; data += 1.0; printf("The values in fPtr = %p and data = %.2f\n", fPtr, data); return 0; } a) What will be the output of the printf statement? b) Explain how the value of fPtr is determined.							2	L4		2																

5.	<p>Consider the following Python code:</p> <pre>A = [10, 'apple', 3, 'banana'] B = [20.0, 100, 30.0, 2.0]</pre> <p>for item in A: for value in B: if type(item) == type(value): print(item, value)</p> <p>a) What will be the output of this code? b) Explain the reasoning behind the output.</p>	2	L4	2
6.	<p>Consider the definition of a union type below:</p> <pre>typedef union { int i; double d; char c[4]; } MY_UNION;</pre> <p>MY_UNION myUnion;</p> <p>a) What is the size of myUnion in a 64-bit system? b) If myUnion.i = 0x12345678, what will be the values of myUnion.c[0], myUnion.c[1], myUnion.c[2], and myUnion.c[3] assuming little-endian memory representation?</p>	2	L4	2
7.	<p>Answer the questions based on the code snippet given below:</p> <pre>xTaskCreate(genericTask, /* Task function. */ "genericTask", /* String with name of task. */ 10000, /* Stack size in words. */ (void *)&param1, /* Parameter passed as input of the task */ 2, /* Priority of the task. */ NULL); /* Task handle. */</pre> <p>delay(2000);</p> <p>a) What core will the task created with xTaskCreate run on if you do not specify a core? b) Explain what happens if you pass NULL as the task handle in xTaskCreate.</p>	2	L3	3
8.	Compare and Contrast the key differences between NB-IoT and LTE-M in terms of latency, throughput, and deployment?	1	L4	1
9.	Give a list of salient features of WiFi 6.	1	L2	1

10	<pre>#include <stdio.h> void myFn(void) { static int myCounter = 0; myCounter++; printf("myCounter = %d\n", myCounter); } int main() { myFn(); myFn(); myFn(); return 0; }</pre> <p>What will be the output of the above program?</p>	1	L4	2
11	<p>Write a Python function to flatten given matrix into a 1D list using comprehension and the output should have double that of individual elements in the matrix,</p> <pre>matrix = [[1, 2, 3], [4, 5], [6, 7, 8, 9], [10]]</pre> <p>a) What will be the output of the function when applied to the given matrix?</p>	1	L6	2
12	What is FreeRTOS? List a couple of APIs related to tasks.	1	L2	3
13	Write a short note on each of the macros used to access variable parameters passed to a function in a C program.	1	L3	2

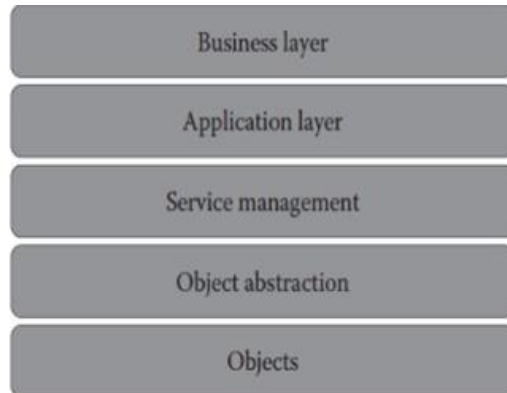
Course Outcomes

1. Choose a suitable wireless protocol based on the problem domain of an IoT product
2. Understand NumPy, TensorFlow framework and the need for TFLite for Edge Devices
3. Identify an IoT solution based on the features supported by ESP32 and FreeRTOS
4. Demonstrate integration of cloud computing platforms with ESP32

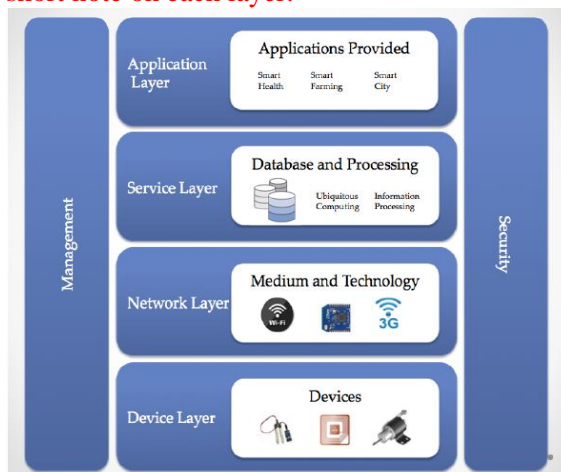
Marks Distribution									
L1	L2	L3	L4	L5	L6	CO1	CO2	CO3	CO4
0	6	3	8	2	1	8	9	3	0

Answers

1.

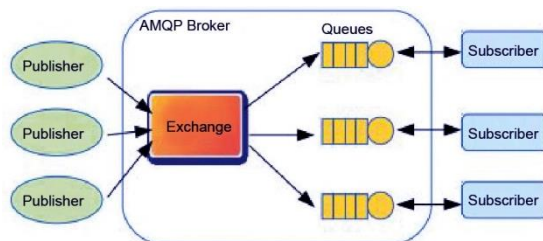


Short description about each layer and its responsibilities. You can also draw the below diagram with a short note on each layer.



2. Artificial Intelligence (AI) and Machine Learning (ML) can significantly enhance a fan control system in a smart home by making it more accurate and responsive to varying conditions. like Data-Driven Decision Making, Personalized Control, Improved Accuracy and Efficiency

3.



AMQP is based on Publish/Subscribe pattern which is known as pub/sub.

It provides a framework for exchanging messages between publishers and subscribers.

The publishers and the subscribers rely on a message broker that relays messages from the publisher to the subscribers.

The host (publisher) publishes messages (events) to a channel that subscribers can then sign up to

4. A) The values in fPtr = 0x7FFFC1234564 and data = 4.14
 B) Below points explain the reasoning behind the above answer.
 - The value of fPtr is determined by pointer arithmetic.
 - Initially, fPtr is assigned the address of data, which is 0x7FFFC1234560.
 - When fPtr is incremented (fPtr++), the pointer moves to the next float location in memory. Since a float typically occupies 4 bytes, the address stored in fPtr increases by 4 bytes.
5. The output is:
 10 100
 3 100
Explanation: It matches the data type within the list and prints the matched datatype (here integers) with its element.
6. A) Since double d is of 8 bytes, and that is the largest member within the union, the size of myUnion will be 8 bytes.
 B) In a little-endian system, the least significant byte is stored at the lowest memory address. This means that the integer **0x12345678** will be stored in memory in reverse order, byte by byte.
myUnion.c[0] = 0x78, myUnion.c[1] = 0x56, myUnion.c[2] = 0x34, myUnion.c[3] = 0x12
7. a) It will be scheduled on any core based on its availability.
 b) When you pass NULL as the task handle in xTaskCreate, it means that you are not interested in keeping a reference to the task handle. The task will still be created and run as usual, but you will not be able to interact with it later using the handle.
8. **Latency:** LTE-M offers lower latency than NB-IoT, making it suitable for more real-time applications.
Throughput: LTE-M provides higher throughput compared to NB-IoT, which is more appropriate for applications that require more data to be transmitted.
Deployment: NB-IoT is more flexible in deployment, especially in areas with poor coverage, while LTE-M leverages existing LTE infrastructure and is easier to implement in areas with good LTE coverage.
9. Beam forming, MU-MIMO antennas and Target Wakeup Time (TWT), simultaneous operation at 2.4 GHz & 5GHz are some of the major features of WiFi6. Any two features are sufficient.
10. myCounter = 1
 myCounter = 2
 myCounter = 3
11. flatList = [2*x for row in matrix for x in row]
 # Implementaiton without using comprehension – 0.5 marks will be reduced.
 def flatten_matrix(matrix):
 flat_list = []
 for row in matrix:
 for element in row:
 flat_list.append(element)
 return flat_list

 # Example matrix
 matrix = [
 [1, 2, 3],

```

[4, 5],
[6, 7, 8, 9],
[10]
]
# Apply the function
flattened = flatten_matrix(matrix)
print(flattened)
output is [2, 4, 6, 8, 10, 12, 14, 16, 18, 20] - corrected

```

12. FreeRTOS is an open-source Real-Time Operating System (RTOS) designed for embedded systems. It provides multitasking capabilities, allowing multiple tasks to run concurrently on a single microcontroller. This highly responsive to events and predictable latencies involved in its execution compared to other commercial OS such as Windows and Linux. FreeRTOS is also having a smaller footprint in terms of its memory requirements and also has simpler structure in terms of process/task management. It does not maintain parent-child relationship as well as concept of threads running within the context of tasks or processes.. – Any of above two points are to be mentioned.

Any two of the below APIs are to be mentioned.

xTaskCreate(), xTaskCreatePinnedToCore(), pcTaskGetName(), xPortGetCoreID() are few APIs

13. `va_list args; va_start(args, fmt); va_end(args);`
va_start(), va_arg() and va_end()
va_list args; - Variable of type `va_list` is declared to hold the starting address of arguments.
va_start(args, fmt); - Initializes the starting addr of list of variables passed into the function
va_arg(args, type) accesses the argument of a particular type by updating the argument list
va_end(args); - Clears the list created above.
