G – 16782

Register No. | | | | | | | | |

BE Degree Examination November 2024

Fifth Semester

Computer Science and Engineering

22CST51 – INTERNET OF THINGS AND CLOUD COMPUTING

(Regulations 2022)

Time: Three hours | Maximum: 100 marks

Answer all Questions

Part – A (10 × 2 = 20 marks)

1. Distinguish between IoT and M2M. Give an example for each. [CO1,K2]
2. Identify the types of data generated for a forest fire detection system. [CO1,K2]
3. Consider an intelligent energy conservation system. For this system, identify the challenges in using LPWAN. [CO2,K3]
4. Find the appropriate network technologies for the following: [CO2,K3]
   1) Connected ambulance for emergency medical services
   2) Smart home security system.
5. How does Raspberry Pi differ from a traditional computer? [CO3,K2]
6. Write a program to control the turning on and off of an LED using a Raspberry Pi. [CO3,K3]
7. Identify the type of cloud for the following applications: [CO4,K3]
   1) Hosting websites for global accessibility
   2) Hospitals and clinics collaborate on patient data
8. Provide any two examples for device cloud. [CO4,K2]
9. Identify the AWS services needed for storing and retrieving sensor data. [CO5,K2]
10. State the use of list-topic-rules command. Give an example for using it. [CO5,K2]

Part – B (5 × 16 = 80 marks)

11. a. Consider a smart irrigation system for a home garden with soil moisture sensors. The soil moisture sensors monitor moisture levels and if the soil is too dry, the system activates a water valve. Design an IoT system for this use case and also suggest an appropriate IoT level and justify the same. (16) [CO1,K4]

(OR)

b. An IoT based ICU patient monitoring system enables continuous, automated monitoring of patients by collecting health parameters like blood pressure, heart rate and temperature through sensors. These parameters are sent to the cloud via internet connectivity, allowing doctors and users to access real-time health data from anywhere, ensuring timely updates and reducing the burden of manual monitoring. Answer the following: (16) [CO1,K4]
   1) Identify the purpose and requirement specification.
   2) Suggest a domain model specification.
   3) Choose a suitable IoT level for the above system and justify the same.

12. a. i) Classify LPWAN network topologies. How these topologies enable device connectivity? Illustrate with suitable sketches. (10) [CO2,K2]

ii) Draw the layered architecture of IoT. Highlight the role of each layer. (6) [CO2,K2]

(OR)

b. i) Name the technologies that are available for IoT device discovery. Provide a brief note of them (10) [CO2,K2]

ii) Compare the features of various protocols used in IoT service discovery. (6) [CO2,K2]

13. a. i) Enumerate the various interfaces of Raspberry Pi and their purposes. (6) [CO3,K2]

ii) Develop and IoT system that monitors room temperature and sends the measurements to a cloud platform. The temperature monitoring device shall be built with a micro controller. (10) [CO3,K2]

(OR)

b. i) What are packages in Python that supports development of IoT based projects? Provide a brief note of them. (6) [CO3,K2]

ii) Design and implement a Python program to simulate an automated traffic light control system at a busy junction using a Raspberry. (10) [CO3,K2]

14. a. A set of IoT sensors monitor water quality and detect leakages in water pipelines. Data collected from these sensors is processed in the cloud, where the alerts are generated for leaks, water quality issues or maintenance needs. Authorities and residents are notified in real-time. For this use case, illustrate how sensors and actuators are being quickly integrated to do a real-time capture of data and use them to arrive at appropriate and actionable insights. (16) [CO4,K3]

(OR)

b. Apply edge analytics to design a system that detects the traffic rule violation such as speeding, red light violations etc at intersections. Also describe the architectural components of the smart traffic system. (16) [CO4,K3]

15. a. i) How does AWS IoT registry help in managing devices? Explain with appropriate examples. (8) [CO5,K2]

ii) How do we connect and communicate with AWS IoT core for providing an IoT solution? (8) [CO5,K2]

(OR)

b. i) How does AWS IoT Device shadow allow us to manage and interact with IoT devices more effectively by creating a persistent virtual representation of the state of the device in cloud? (8) [CO5,K2]

ii) Present a brief note of using tags to categorize IoT resources in different ways. (8) [CO5,K2]

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	–	49	33	18	–	–

Kongu Engineering College.

Answer Key

22CST51 - Internet of Things and Cloud Computing.

PART-A

1. Distinguish between IoT and M2M.

IoT (1mark)	M2M (1mark)
* Connect devices over the Internet	* Direct communication b/w machines.
* Allows data processing and analytics.	* Typically does not involve Internet connectivity.
* Ex. Smarthome device like Alexa	Ex. Vending machine.

2. Identify the type of data generated for forest fire detection system.

Environmental data: Temparature, humidity, wind speed (1mark)
Sensor data: Smoke levels, gas (CO, CO_2) } 1 mark
Geolocation data: GPS coordinates of the fire }

3. Identify the challenges in using LPWAN. (any 2) 2marks)

* Limited bandwidth & data rate.
* Limited device mobility
* Higher latency

4. Find the appropriate n/w technologies.

* Connected ambulance - cellular, LPWAN; bLowPWAN (1mark)
* Smart home security system - WiFi, Zigbee. (1 Mark)

5. How does Raspberry Pi differs from a traditional computer?

Raspberry Pi (1mark)	traditional computer (1mark)
* Compact, Low cost	Larger, more expensive
* Designed for specific projects like IoT and prototyping	general purpose.

6. Write a Program to control the turning on and off an LED. using a Raspberry Pi

```
import RPI.GPIO as GPIO
import time

LED_PIN =17
GPIO.setmode(GPIO.BCM)
GPIO.setup(LED_PIN, GPIO.OUT)

while True:
    GPIO.output(LED_PIN, GPIO.HIGH)
    time.sleep(1)
    GPIO.output(LED_PIN, GPIO.LOW)
    time.sleep(1)
GPIO.cleanup()
```

} (1 mark)

} (1 mark)

7. Identify the type of cloud for following application.

(i) Hosting Website for global accessibility: (1 mark)

Public cloud.— AWS Elastic Beanstalk.
Google platform.

(ii) Hospitals and clinics collaborate on Patient data (1 mark)

Hybrid cloud - IBM Hybrid,
Microsoft Azure.

8. Provide any two examples for device cloud.

1. Amazon Web services (AWS IoT cloud)
2. Microsoft Azure.
3. Google cloud
4. IBM IoT platform
5. Thing speak.

any 2

9. Identify the AWS services needed for storing and retrieving sensor data. (any 2) - 2 mark

AWS IoT core
Amazon S3
Amazon DynamoDB

10. State the use of list-topic-rules command. Give an example.

list-topic-rule - AWS IoT core for message Processing and routing. (1 mark)

Ex:
aws iot list-topic-rules - max results 5
(1 mark)

PART-B

11 a. Smart irrigation System. (16 marks)

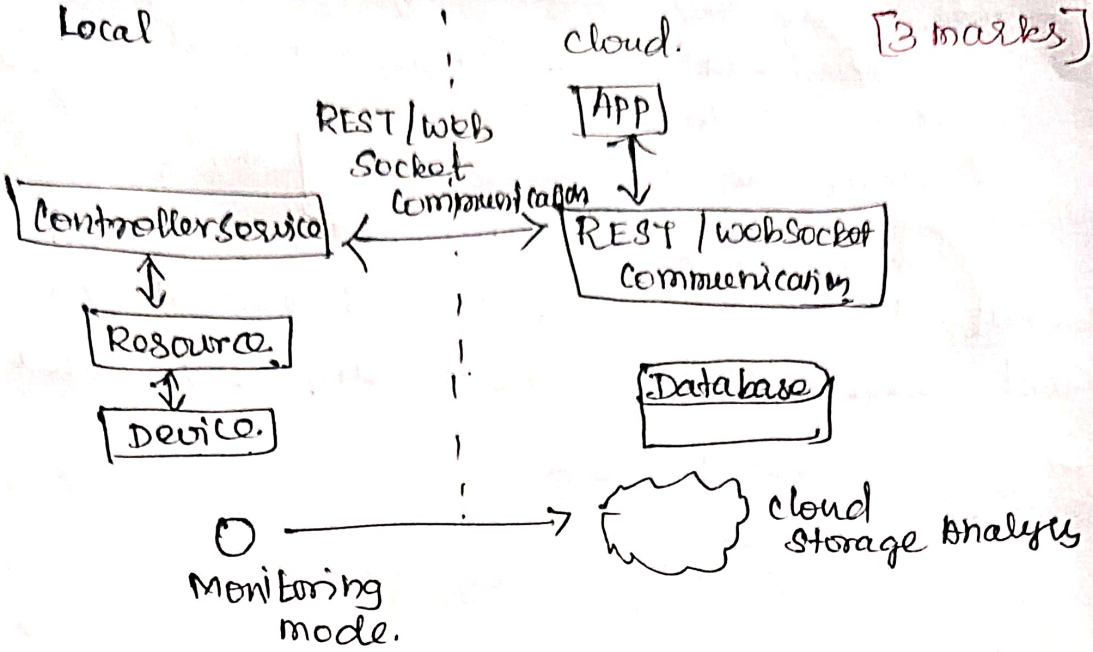
IoT Level - Level 3 | 4 | 5 | 6 — [3 Marks]

Justification - Real time monitoring — [5 Marks]
Data Analytics.
Automation
Scalability.

IoT system Design. [2 Marks]

Sensor → Actuators → Raspberry pi → WIFI/ LPWAN → Thing Speak → Mobile application
↳ Soil Moisture
↳ Temperature

Working Explanation. [3 Marks]



11 b. ICU Patient Monitoring System. (16 marks)

1. Purpose and requirement specification. [4 marks]

↳ ICU monitoring, automated.
- blood pressure, heart rate, temparature, oxygen

↳ Provide Realtime monitoring, improving decision making, reduce manual efford.

Requirements
- Data collection - blood pressure, heart rate, temp.
- Data transmisson, Alerts Notification, Remote Access

2. Domain model Specification [4 marks]

entities, & Relationships.

Patient associate with Heath Parameters
Sensor collect data from Patient send to Cloudservice

3 IoT Level [8 marks]

Level 3 to b. 3+3

Justification - Sensor collect the Data
Contentual information
Cloud integration, Realtime integration

Diagram.

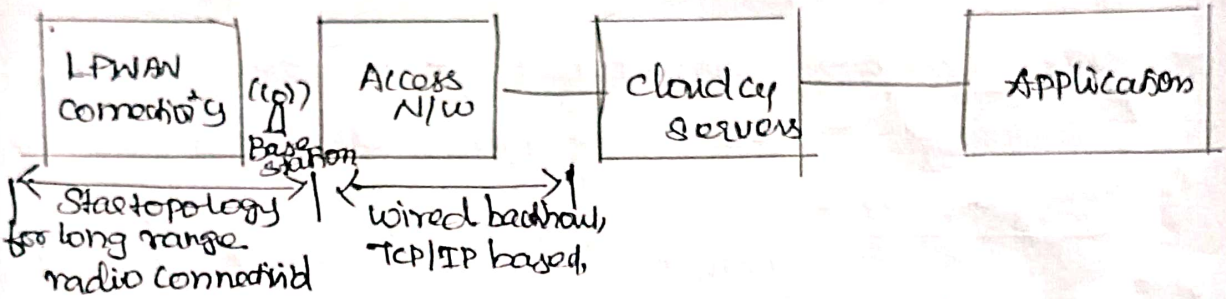
12 a (i) LPWAN N/w topologies. (10 marks)

Low Power Wide Area N/w [2 marks]

* Direct Device connectivity (basestation)
* Indired Device connectivity through an LPWAN gateway

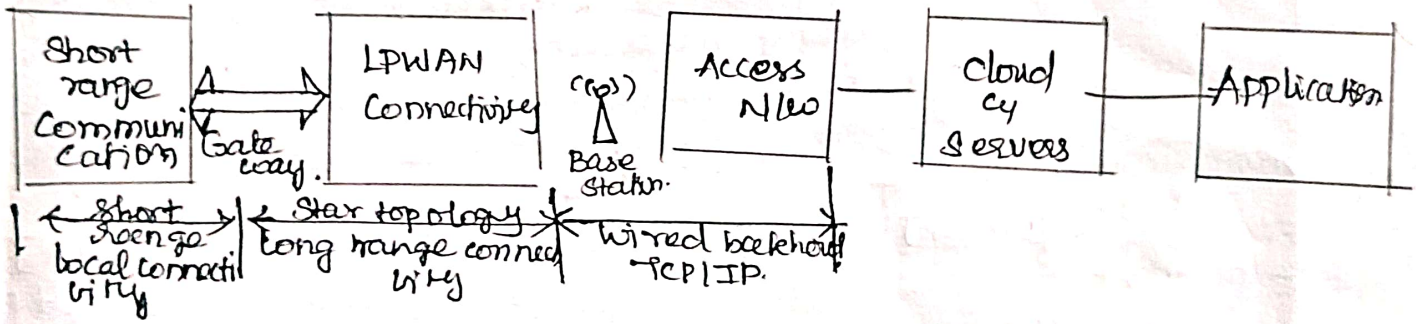
Direct Device [4 marks]

Base station is responsible for translation of protocol from IoT protocols such as MQTT or CoAP to specific device application protocols.



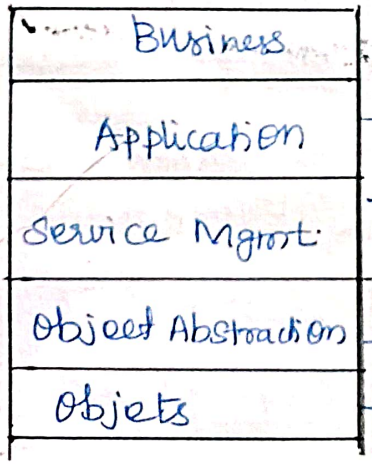
Indirect Device Connectivity [4 marks]

→ local gateway used – support large numbers of device
– short range radio (SRD) Ex. Zigbee BLE



12 a(ii) Layeed Architectue. (6 marks)

Diagram – 2 mark, Explanation 4 marks



– Build a bussiness model. for analyse, implement
– Provides Services requested by Customer
– middleware – Processing data, necessary decision taking
– data transmission Object to Service. RFID, WiFi, Blue tooth, zigbee, Infrared.
→ physical devices. – Sensons.

12 bii) Name the technologies that are available for IOT device discovery. Brief Note. (10 marks)

* Self configured, dynamic, & efficient.

(any 5 each 2 marks)

* **Bluetooth Becons** – smartphone, tablets.
  - low power, and allows devices to discover nearby devices using beaconing.
  - Proximity based services.
* **WiFi Aware.** – similar to Bluetooth. connect share updates → photo, play games.
* **Physical Web** – Device broadcast beacon – using URL
  - detect the physical web signal
* **Open Hybrid** – directly map a digital interface to physical object.
* **Shazam** – identify audio content present in TV, Connect to Bluetooth, detect ultrasonic signals
* **Chirp** – it slw.
  - Short string data (encoded format)
  → Encoding, decoding
  - split into huge amount of data.
  - highly reliable short distance.

12 bii) IoT service Discovery. (6 marks)

mDNS [2 mark]	DNS-SD [2 mark]	UPnP [2 mark]
– multicast DNS (local n/w)	DNS-Standard.	Universal Plug and Play
– n/w type LAN	LAN & WAN	LAN.
– limited to small n/ws	Scalable	Limited to medium n/ws
– Protocol Application layer	Application layer with DNS records	Application layer for transport.
– Zero configuration	DNS or mDNS servers.	Simple set up – Specific configure
– small n/w home automation	Enterprise n/w	Home n/ws

(1)

13.a.i) Various interfaces of Raspberry Pi & its purpose

* Serial — (2 marks)
  - Has receive (Rx) & transmit (Tx) pins to enable communication with peripheral devices
  - can be used for low-speed data communication

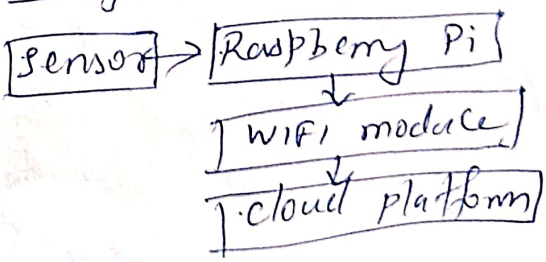
* Serial Peripheral Interface — (2 marks)
  - Synchronous serial data protocol
  - used to establish communication with one or more peripheral devices
  - Has one master device & one or more slave devices
  - Pins used: MISO, MOSI, SCK, CE0, CE1

* I2C — (2 marks)
  - Allows synchronous data transfer with just 2 pins - SDA (Serial Data) & SCL (Serial clock)
  - It is a multi-master, multi-slave design
  - Ideal for low-speed devices

13.a.ii) Monitoring room Temperature (10 marks)

Components + Tools — (2 marks)
* Arduino / ESP32
* Temperature sensor
* Thingspeak
* WIFI module if Arduino is used

Design — (2 marks)



Code — (6 marks)

```
import time
import request

def read_temp():
    hum, temp = DHT.read_retry(sensor, pin)
    if temp is not None:
        return round(temp, 2)
    else:
        print("sensor failed")
        return 0

def uploadTemp(temp):
    res = request.get(THINGSPEAK_URL, params={
                'apiKey': THINGSPEAK_API_KEY,
                'field1': temp})
    if res.status_code == 200:
        print("Temperature = {temp}°c")
    else
        print("failed")

def main():
    while True:
        temp = read_temp()
        if temp != 0:
            uploadTemp(temp)
        else:
            print("Failed")
```

13.b.i) Packages in Python (b)

(Any three with example 2 marks each)

* JSON -
  - Javascript Object Notation
  - Data received from sensors or APIs or format the data to send to cloud platform

* XML - Extensible Markup Language
  - Provides features to parse & create XML data
  - Used in IoT system for structured representation

* Httplib
  - used to develop network or Internet - based applications
  - httplib2 - HTTP client library

* Urllib
  - Used to retrieve & parse URLs
  - uses urlopen() function to fetch URLs

* smtplib
  - used to send mail through applications
  - Routes email between mail servers

13.b.ii) Traffic Light controller — (10 marks)

Components Required — (1 mark)

* Raspberry Pi
* LEDs - Red, Green & Yellow
* Jumper wires

Code

```
// Importing necessary packages & initial setup - (3 marks)
import RPI.GPIO as gpio
import time
red = 11
green = 13
Yellow = 15
gpio.setmode(GPIO.BOARD)
gpio.setup(red, GPIO.out)
gpio.setup(green, GPIO.out)
gpio.setup(Yellow, GPIO.out)

// Turning on or off specific LEDs for traffic
light simulation - (6 marks)

while True:
    GPIO.output(green, GPIO.HIGH)
    GPIO.output(red, GPIO.LOW)
    GPIO.output(Yellow, GPIO.LOW)
    time.sleep(50000)
    GPIO.output(green, GPIO.LOW)
    GPIO.output(red, GPIO.HIGH)
    GPIO.output(Yellow, GPIO.LOW)
    time.sleep(50000)
    GPIO.output(green, GPIO.LOW)
    GPIO.output(red, GPIO.LOW)
    GPIO.output(Yellow, GPIO.HIGH)
    time.sleep(30000)
```

14-a) Monitoring water quality (16 marks)

Required components, Tools, & Protocols – (4 marks)

components – water leakage & water quality / flow sensor, Raspberry Pi / Arduino / ESP8266 / ESP32, WIFI Module of Arduino, Actuator

cloud services - AWS IOT core

Protocols - MQTT, HTTP for REST API

Cloud Deployment Type – (6 marks each 3)

Hybrid model – combination of public & private cloud

Private cloud – Data regarding water quality & leakage is stored

public cloud – Host the app or website to control the function & send alerts & status notification

cloud Service Model (6 marks - each 3)

IaaS – S3 / DynamoDB / RDS for data storage
– EC2 for running the app

PaaS – develop, test, & deploy application using ElasticBeanStalk / Green Grass & AWS IOT core

---

14-b) Architectural Components of Smart Traffic

Features – (2 marks)

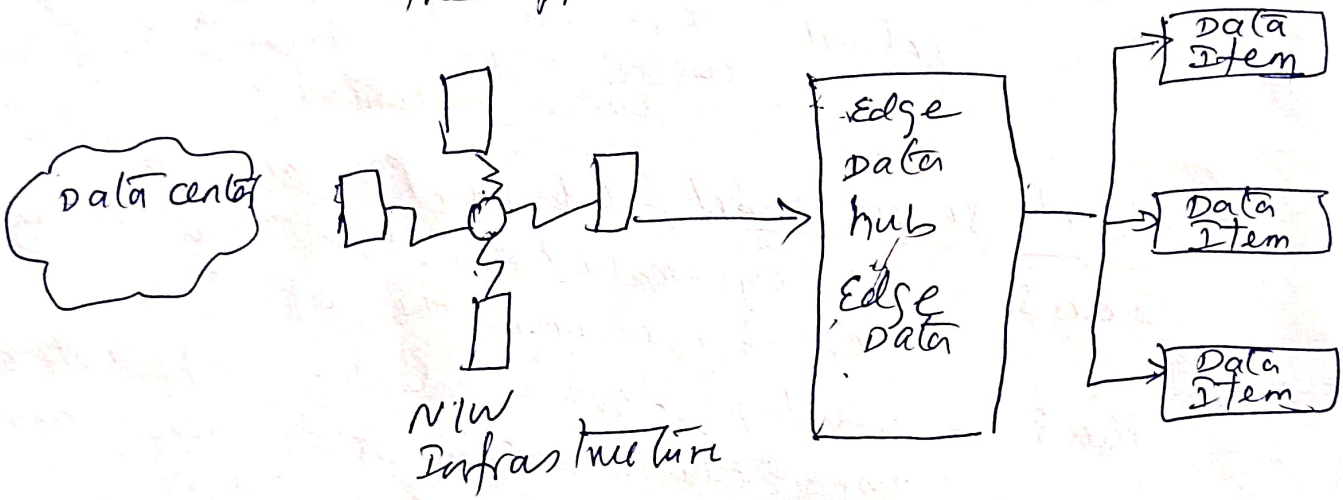
* Monitor Traffic violations
* Perform real-time data processing
* Trigger alerts
* Send traffic data

3 layers (each layer 4 marks - 12 marks)

1) Edge layer (location sit devices)
   - sensors, edge computing devices, Gateway, ML, Local Actions

2) Communication layer
   - Data transfer using MQTT, HTTP & LoRaWAN

3) Cloud layer
   - Data aggregation, Bigdata Analytics, ML
   - Paas - for application development & deployment
   - Iaas for running instances that host the applications.



. Architecture Diagram - (2 marks)

15 9i

AWS IoT Registry [8 marks]

Def (2 mark)

→ Provided by AWS.
→ Managing & organizing devices connected.
→ Centralized Repository. (Step 6 marks)

Step 1. Device Registration - Unique Identifier.
2. Thing types - attributes - temp, humidity
3 Device Metadata - device status - active/inactive
4. Device Authentication - authentication
5. Integration rules - common rule
6. Device shadow - track their state.

15 9ii) AWS IoT core. [8 marks] [each step 1 mark]

Step 1 Create an AWS account and setup IoT core
Step 2. Register IoT devices (Things) in AWS IoT core
Step 3: Setup Secure communication.
↳ Authentication & Authorization
Step 4: Configure the Device to communicate with AWS IoT core. - AWS IoT SDK,
MQTT communication
IoT Endpoint
Step 5: Publish & Subscribe
Step 6: Monitoring & manage Device
Step 7: Real time Analytics.
Step 8: Lambada functions

15 bi AWS IoT Device shadow. [8 marks] Def - 2 mark
each step 1 mark (6 mks)

1) Persistent Virtual Representation of Device state
2) Offline Device Management
3) Synchronization of Desired and Reported States
4) cloud device communication without Direct connecting.
5) Automation 6) Device updates.

15 bii Tags to Categorize IoT resources (8 Marks)

- Tags are key value pairs – associate with each resource (2 marks)
- Organization, Access control, Cost tracking, Automation

Way to use tags (4 Marks)

* Tagging IoT Devices
* Tag - MQTT topics
* IoT Rules
* IoT Analytics
* Cloud based Services

Benefits (2 marks)

- Improve Resource Mgmt.
- Better Access control
- Automation
- Cost allocation.

13a.i) Interfaces of Raspberry Pi – (6 marks)

* Serial – (2 marks)
  - has receive (Rx) & transmit (Tx) pins to enable communication with peripheral devices
* Serial Peripheral Interface (SPI) – (2 marks)
  - Synchronous serial data protocol used to establish communication with one or more peripheral devices
  - Has one master device & one or more peripheral devices
  - Has following pins: MISO, MOSI, SCK, CE0, CE1
* I2C – (2 marks)
  - Allows to connect to hardware modules
  - Allows synchronous data transfer with just 2 pins: SDA & SCL