

Nom : Prénom: Classe:

Matière :	Programmation Mobile	Filière : DDSI,IA	Niveau : 4 ^e
Date de l'épreuve :	XX/YY/2026 14:00 - 16:00	Durée de l'épreuve : 2H00	

Consignes à Respecter

- La clarté et la simplicité des réponses et du code seront prises en considération.
- Toute communication entre candidats est formellement interdite.
- Tout appareil électronique (smartphone, montre connectée, tablette, etc.) doit être éteint et rangé hors de portée.

Partie I : Complétez les phrases suivantes en choisissant le mot ou l'expression correcte parmi les propositions données :(5 pts).

Banque de mots (12 propositions – 5 réponses) : 6LoWPAN | CoAP | Fog Computing | InfluxDB | Isolation Forest | MiNiFi | QoS 2 | Spark Streaming | Time Series | transducteur | Z-Score | Zigbee

- 1.1.** Dans l'architecture IoT en 5 couches (Physique, Liaison, Réseau, Transport, Application), le protocole _____ de la couche Réseau permet d'encapsuler et de transporter des paquets IPv6 sur des réseaux à faible puissance et faible débit (LLN).
- 1.2.** Le _____ est le composant fondamental d'un capteur qui convertit une grandeur physique (température, pression, lumière) en signal électrique exploitable par un microcontrôleur.
- 1.3.** L'algorithme de Machine Learning _____ détecte les anomalies en isolant récursivement les observations via des arbres de décision aléatoires ; il est particulièrement efficace sur des données IoT de haute dimension en temps réel.
- 1.4.** Dans l'écosystème Apache NiFi, _____ est la version allégée (C++) conçue spécifiquement pour le déploiement sur des passerelles Edge avec des ressources CPU/mémoire très limitées, assurant la collecte locale et la transmission vers NiFi central.
- 1.5.** La base de données _____ est optimisée pour le stockage et la requête de données temporelles (Time Series) ; elle utilise un modèle de données basé sur des mesures (measurement), des tags (indexés) et des champs (non indexés), avec un langage de requête Flux.

Partie II : Questions de compréhension (5 pts)

- 2.1.** Dessinez et décrivez l'architecture IoT à 5 couches. Précisez le rôle de chaque couche et citez une technologie associée.

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2.2. Définissez ce qu'est un capteur. Expliquez la différence entre un capteur analogique, un capteur numérique et un capteur logique (TOR). Donnez un exemple concret pour chaque type.

.....

.....

.....

.....

.....

.....

2.3. Décrivez les trois niveaux de Quality of Service (QoS) du protocole MQTT. Pour chaque niveau, expliquez le mécanisme de livraison et précisez dans quel cas d'usage il est préférable.

.....

.....

.....

.....

.....

.....

2.4. Comparez les protocoles MQTT et CoAP selon quatre critères : modèle d'échange, protocole de transport, overhead et cas d'usage privilégié.

.....

.....

.....

.....

.....

.....

2.5. Comparez les bases de données InfluxDB et MongoDB dans le contexte IoT. Pour chacune, précisez : le type de données optimisé, le modèle de stockage, les forces et une limitation.

.....

.....

.....

.....

.....

.....

.....

.....

Partie III : Exercices de raisonnement (10 pts)

- 3.1. Expliquez la difference entre une broche configuree en sortie numerique (OUTPUT) et une broche configuree en entree numerique (INPUT) sur Arduino. Citez les fonctions Arduino utilisees pour chaque operation (configuration, ecriture, lecture). 2pts

.....

.....

.....

.....

.....

.....

.....

.....

- 3.2. Dans le montage d'une LED controlee par un bouton-poussoir, a quoi sert la resistance de pull-down de 10k ohms connectee au bouton ? Que se passerait-il si cette resistance etait absente ? 2pts

.....

.....

.....

.....

.....

.....

.....

.....

- 3.3. Expliquez la difference entre la fonction delay() et la fonction millis() en Arduino. Pourquoi millis() est-elle preferable pour des applications interactives ? 2pts

.....

.....

.....

.....

.....

.....

.....

.....

3.4. Question 5 - Programmation ESP32 - Entrees / Sorties Numeriques

4pts

Contexte : Un ESP32 possede une LED integree sur GPIO 2 et un bouton-poussoir connecte sur GPIO 4. L'ESP32 lit l'etat du bouton et allume la LED uniquement lorsque le bouton est presse. On utilise la resistance de pull-up interne de l'ESP32 (pas de resistance externe necessaire).

Completez les parties manquantes (indiquees par ____):

Rappel : avec INPUT_PULLUP, la broche est HIGH par défaut. Quand on presse le bouton, elle passe à LOW.

```
# _____ // GPIO de la LED integree ESP32
# _____ // GPIO du bouton-poussoir

void setup() {
  Serial.begin(115200);

  // Configurer la LED comme sortie
  _____(LED_PIN, _____);

  // Configurer le bouton avec pull-up interne
  _____(BUTTON_PIN, _____);
}

void loop() {
  // Lire l'etat du bouton (LOW = presse car pull-up)
  int etat = _____(BUTTON_PIN);

  if (etat == _____) { // Bouton presse ?
    digitalWrite(LED_PIN, _____); // Allumer LED
    Serial.println("LED ON");
  } else {
    digitalWrite(LED_PIN, _____); // Eteindre LED
    Serial.println("LED OFF");
  }

  _____ // Petite pause en millisecondes
}
```