

Laboratoires IoT

Mise en oeuvre des architectures IoT à la base de IoT-DevKit de SmartComputerLab

Table de Contenu

0. Introduction	4
0.1 ESP32 Soc – une unité avancée pour les architectures IoT.....	5
0.2 Carte Heltec WiFi LoRa.....	5
0.3 IoT DevKit une plate-forme de développement IoT.....	6
0.3.1 Cartes d'extension – quelques exemples.....	7
0.4 L'installation de l'Arduino IDE sur un OS Ubuntu.....	8
0.4.1 Installation des nouvelles cartes ESP32 et ESP8266.....	8
0.4.2 Préparation d'un code Arduino pour la compilation et chargement.....	10
Laboratoire 1	11
1.1 Premier exemple – l'affichage des données.....	11
A faire:.....	11
1.2 Deuxième exemple – capture et affichage des valeurs.....	12
1.2.1 Capture de la température/humidité par HTU31D.....	12
A faire :.....	12
1.2.2 Capture de la luminosité par BH1750.....	13
A faire :.....	13
Laboratoire 2 – communication en WiFi et serveur ThingSpeak.fr	14
2.1 Introduction.....	14
2.1.1 Un programme de test – scrutation du réseau WiFi.....	14
2.2 Mode WiFi – STA, client WEB et serveur ThingSpeak.....	15
2.2.1 Accès WiFi – votre Phone ou un routeur WiFi-4G.....	16
A faire.....	17
A faire (comme dans l'exemple précédent) :.....	19
2.4 Le programme de réception des données à partir d'un serveur ThingSpeak.....	20
A faire.....	21
Laboratoire 3 – Serveur Thinger.io	22
3.1 Introduction.....	22
3.1.1 Un simple code Arduino pour allumer une LED (pin 25).....	22
A faire :.....	23
3.2 Les mécanismes de Thinger.io sur l'exemple du capteur DHT22.....	24
3.2.1 Code Arduino.....	24
3.2.2 Le <i>bucket</i> de Thinger.io.....	26
3.2.3 <i>Dashboards</i>	27
A faire :.....	27
3.3 Gestion d'une interruption pour l'envoi des données vers un serveur Thinger.....	28
3.3.1 Programme avec le code principal dans la boucle loop().....	28
3.3.2 Programme avec le code principal dans une tâche spécifique.....	30
A faire :.....	31
Laboratoire 4 – communication longue distance avec LoRa (<i>Long Range</i>) et WiFi en mode ESP-NOW	32
4.1 Introduction.....	32
4.1.1 Modulation LoRa.....	32
4.1.2 Paquets LoRa.....	33
4.2 Premier exemple – émetteur et récepteur des paquets LoRa.....	34
A faire :.....	35
4.3 onReceive() – récepteur des paquets LoRa avec une interruption.....	36

A faire :	36
4.4 Communication directe entre cartes par le protocole ESP-NOW.....	37
4.4.1 Récupération de l'adresse MAC.....	37
4.4.2 L'envoi de trames MAC après une interruption sur capteur PIR.....	38
4.4.3 La réception d'une trame MAC.....	39
A faire :	40
Laboratoire 5: Développement des serveurs locaux WEB-IoT.....	41
5.1 Simple serveurs WEB en mode station et mode point d'accès.....	41
5.1.1 Serveur WEB en mode station - STA.....	41
5.1.2 Un simple serveur WEB en mode softAP.....	43
Code complet.....	43
5.2 Un serveur WEB avec système SPIFFS.....	45
5.2.1 Le fichier HTML.....	46
5.2.2 Le fichier CSS.....	46
5.2.3 Le fichier principal - script Arduino.....	47
5.3 Mini serveur WEB avec une carte SD.....	48
5.3.1 Un programme de test de la carte SD.....	49
5.3.1 Le programme de mini serveur WEB avec une carte SD.....	50
A faire.....	55
Laboratoire 6: Terminaux « intelligents » avec FreeRTOS.....	56
6.1 Création d'une tâche.....	56
A faire :	56
6.2 Création d'une file d'attente et communication entre 2 tâches.....	57
6.2.1 Modèle Producteur-Consommateur avec une file d'attente.....	57
6.2.2 Tâche capteur - sensorTask(void * pvParameters).....	58
A faire :	58
6.3 Gestion d'une interruption avec un sémaphore binaire.....	59
A faire :	59
6.4 L'envoi des données captées par une tâche FreeRTOS.....	60
A faire :	62
Laboratoire 7 - Développement des passerelles IoT.....	63
7.1 Passerelle LoRa-ThingSpeak.....	63
7.1.1 Le principe de fonctionnement.....	63
7.1.2 Le code.....	63
A faire :	64
7.2 Passerelle LoRa-Thingier.io.....	65
7.2.1. Code d'émetteur.....	65
7.2.2 Code de la passerelle.....	66
A faire :	68