

AZURE RTOS

Introduction	2
Définition des termes	2
OS (Operating System)	2
RTOS (Real Time Operating System)	2
IoT (Internet of Objects)	3
MCU (MicroController Unit)	3
Azure RTOS	3
Définition	3
Avantages	4
Composants	4
Azure RTOS ThreadX	4
Azure RTOS FileX	4
Azure RTOS USBX	5
Azure RTOS NetX et NetX Duo	5
Azure RTOS GUIX Studio et GUIX	5
Azure RTOS TraceX	5
Carte	6
Démonstration	6
Installation	6
Bibliographie	8
Bibliographie	8
Figure 1 : Fonctionnement d'un OS	2
Figure 2 : Sortie dans Termite - Démarrage	7
Figure 3 : IoT Central – Visualisation des données	7
Figure 4 : Sortie sur Termite - Allumage de la LED	8
Figure 5 : IoT Central – Allumage de la LED	8

Introduction

Selon Oracle, on peut définir l'IoT comme suit :

L'Internet of Things (IoT) décrit le réseau de terminaux physiques (objets) qui intègrent des capteurs, des softwares et technologies et qui se connectent à d'autres terminaux et systèmes sur Internet pour échanger des données avec eux.

En 2022, on compte dorénavant plus de 244 millions objets connectés en France, ce qui représente en moyenne presque plus de 350 objets connectés par tranche de 100 habitants. L'omniprésence de l'IoT soulève de nombreux questionnements. L'un des enjeux majeurs de cette technologie en pleine expansion est de pouvoir intégrer des systèmes complexes communiquant avec le cloud dans des environnements embarqués, qui disposent de ressources limitées (calcul comme mémoire). Azure RTOS se présente comme une des solutions à cette problématique.

Définition des termes

OS (Operating System)

Ensemble des programmes qui permettent l'utilisation des ressources physiques d'un ordinateur (mémoires, processeur, périphériques...) par les applications. L'OS gère les demandes d'accès à ces ressources par les applications en évitant les interférences et en distribuant les ressources nécessaires à chaque demande.

RTOS (Real Time Operating System)

Un système d'exploitation en temps réel est conçu dans un but particulier : le temps maximum entre la réception d'un message en entrée et l'émission d'une réponse en sortie doit être précisément déterminé. Il est ainsi destiné à des applications en temps réel : thermostats programmables, robots industriels, vaisseaux spatiaux, matériel de recherche scientifique...

Contrairement aux systèmes d'exploitation à temps partagé (plusieurs utilisateurs ont accès à du temps processeur, le processeur commutant souvent pour que chaque utilisateur ait un temps processeur minimal, ce qui donne l'illusion à un utilisateur qu'il est le seul à utiliser le processeur), un RTOS est dit déterministe : les résultats de sortie et leur heure d'apparition sont connus en fonction des entrées.

Dans un RTOS, les threads ont des priorités, ce qui assure l'exécution rapide d'un thread si il possède une haute priorité. Si un thread avec une haute priorité (comparée au thread courant) a besoin d'être exécuté, une commutation de contexte a lieu (sauvegarder le contexte du thread courant dans la mémoire et restaurer le contexte du thread à exécuter).

Ils sont donc plus prévisibles et stables, ce qui permet de contrôler des tâches spécifiques. Les systèmes embarqués doivent souvent répondre à des événements dans une limite de temps définie, donc on utilise des RTOS car le fait de pouvoir prédire leur comportement et leur temps d'exécution facilite le respect de ces contraintes.

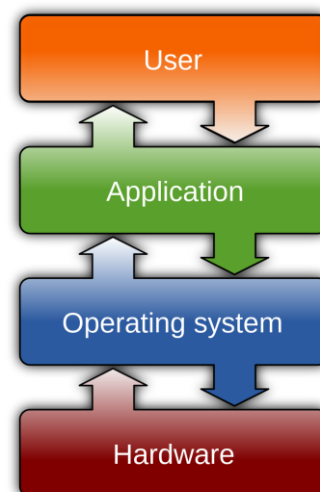


Figure 1 : Fonctionnement d'un OS

Exemple

Un RTOS pourrait être utilisé dans une machine industrielle qui exécute une tâche régulièrement : un bras robot qui doit serrer un boulon sur un châssis d'une voiture, par exemple. Avec un OS classique, on ne pourrait pas être assuré que la tâche soit réalisée à temps. En revanche, avec un RTOS, on sait avec quasi-certitude que la tâche va être faite en un temps déterminé, et on peut donc également effectuer toutes les autres tâches qui suivent.

IoT (Internet of Objects)

L'internet des objets, c'est l'interconnexion entre l'Internet et des objets, des lieux et des environnements physiques. Cette définition désigne en fait tous les objets qui sont reliés à Internet, et peuvent ainsi communiquer avec d'autres objets. Ces connexions permettent donc de rassembler et de partager des données en un seul endroit.

MCU (MicroController Unit)

Un microcontrôleur est un circuit électronique intégré, composé des éléments essentiels d'un ordinateur (processeur, mémoires mortes et vives, périphériques d'entrées et sorties...). Ils se distinguent de ces derniers par leur capacité à être intégrés à d'autres systèmes, leur consommation électrique réduite, leur puissance plus faible et leurs coûts réduits. Ils sont ainsi fréquemment utilisés dans des systèmes embarqués diverses : jouets, télécommandes, électroménager, téléphonie mobile...

Azure RTOS

Définition

Azure RTOS est développé par Microsoft. Il est notamment conçu pour être optimisé pour les appareils IoT et les MCU. En effet, il peut fonctionner avec des ressources très contraintes (moins de 64 ko de mémoire flash), ce qui est particulièrement adapté à ce type d'appareils. Il est déployé sur plus de 6 milliards d'appareils à travers le monde :

Table 1 : Répartition du déploiement d'Azure RTOS (2017)

Domain	Total Deployed Units
<i>Mobile Phones</i>	3 442 371 819
<i>Consumer Electronics</i>	1 392 879 607
<i>Office/Business Automation</i>	923 150 041
<i>Retail Automation</i>	195 680 210
<i>Industrial Automation & Energy/Power</i>	161 670 672
<i>Communications & Networking</i>	86 693 829
<i>Aerospace & Defense</i>	19 368 625
<i>Other & General Purpose</i>	12 877 099
<i>Automotive & Transportation</i>	12 537 911
<i>Medical Devices</i>	12 432 839
TOTAL	6 259 662 652

En plus de fournir un environnement sécurisé, il respecte diverses spécifications :

- Niveau d'assurance d'évaluation EAL4+ : conçu, testé et vérifié méthodiquement (l'EAL est une méthode d'évaluation de la sécurité des applications)
- Sécurité de la couche IP avec IPsec (protocole de sécurité pour les réseaux IP qui fonctionne par échange de clés)
- Sécurité couche socket avec TLS et DTLS (chiffrement, authentification et intégrité des données)
- Chiffrement logiciel certification FIPS 140-2 (permet de valider l'efficacité du matériel cryptographique)

Avantages

Azure RTOS est à ce jour le RTOS le plus déployé à travers le monde. Il offre une API conçue dans un objectif de facilité d'utilisation, et permet d'établir des connexions faciles grâce aux services cloud Azure. Il est aussi efficace (faible encombrement du code, exécution rapide) et sécurisé, deux points essentiels pour des appareils mobiles comme ceux de l'IoT et les MCU. Il prend en charge une grande partie des architectures 32 et 64 bits les plus populaires. Enfin, une licence payante n'est nécessaire que sur des appareils non pré-concédés, et diverses ressources, documentations ainsi que le code source sont disponibles gratuitement.

Composants

Quand on parle de Azure RTOS, on parle en général de la collection de composants (solutions d'exécution) mise à disposition. Il y a en tout 6 composants différents.

Azure RTOS ThreadX

Kernel de Azure RTOS.

- Compact : ~2 KB; empreinte minimale.
- Rapide : commutation de contexte en moins de 1 milliseconde.
- Fonctionnalités avancées :
 - Planification des seuils de préemption (priorités pour les threads, pour réduire la commutation de contexte)
 - Event chaining (événements qui déclenchent d'autres événements)
 - Auto-scaling (Build uniquement les composants utilisés, par ex si les sémaphore ne sont pas utilisés alors ils ne seront pas inclus dans l'image)

Azure RTOS FileX

Système de fichiers haute performance compatible avec FAT, entièrement intégré à ThreadX.

- Compact : ~9 KB, empreinte minimale.
- Rapide : Cache optimisé.
- Fonctionnalités avancées :
 - Tolérance des fautes (Par exemple en cas de panne de courant, évite la corruption du système de fichier).
 - Compatible FAT 12/16/32 et exFAT.
 - Auto-scaling.

Azure RTOS USBX

Pile embarquée haute performance d'hôtes et de périphériques USB, entièrement intégré à ThreadX. Permet la communication en USB pour un système embarqué sous ThreadX.

- Compact : ~20 KB.
- Rapide : Utilise le DMA (Direct Memory Access), pas de passage par le processeur.

Azure RTOS NetX et NetX Duo

Implémentation des normes TCP/IP (IPv4 et IPv6) avec TLS et MQTT, entièrement intégré à ThreadX.

- Compact : ~50KB pour une connexion minimale au cloud.
- Rapide : usage CPU minimal.
- Fonctionnalités avancées :
 - Chiffrement logiciel certification FIPS 140-2.
 - Auto-scaling.
 - Zero copy (évite les tâches de copies d'une zone de mémoire à une autre, pour économiser des cycles processeur).

Azure RTOS GUIX Studio et GUIX

Package d'interface graphique utilisateur, qui permet de concevoir des interfaces graphiques pour systèmes embarqués. GUIX Studio est un logiciel de bureau pour créer des interfaces graphiques facilement et exporter le code GUIX produit vers une cible.

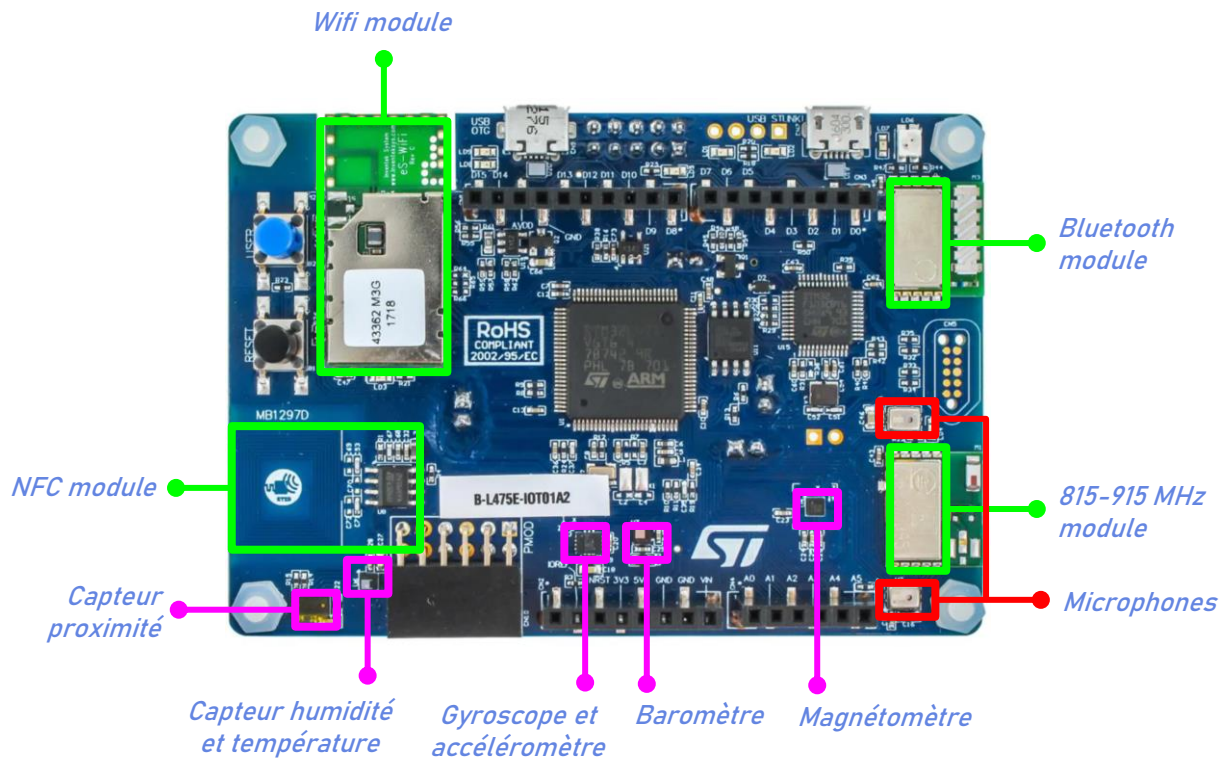
- Compact : ~13KB, support basique.
- Rapide : écrit en C, optimisation du clipping, du dessin et des évènements.
- Fonctionnalités avancées :
 - Ensembles de widgets prédéfinis.
 - Auto-scaling.
 - Zero copy.

Azure RTOS TraceX

Outil d'analyse, fournissant une vue graphique des évènements en temps réel pour comprendre et visualiser le comportement du système (dont interruptions et permutations de contexte qui ne sont pas visible dans des outils de débogage standards).

Démonstration

Carte



Installation

- 1) Installer Cmake (build), ARM GCC (compilation), Termite (gère les sorties série des appareils connectés).
- 2) Créer une application sur IoT Central (composant cloud qui va gérer les évènements et la communication avec les appareils) et configurer un nouvel appareil.
- 3) Ajouter les paramètres de configuration dans le fichier :
`getting-started\STMicroelectronics\B-L475E-IOT01A\app\azure_config.h`
- 4) Construire l'image :
`getting-started\STMicroelectronics\B-L475E-IOT01A\tools\rebuild.bat`
- 5) Flasher l'image sur la carte.
- 6) Lancer Termite pour observer la sortie :

```

Starting Azure thread

Initializing WiFi
Module: ISM43362-M3G-L44-SPI
MAC address: C47F:51:06:A8:E9
Firmware revision: C3.5.2.5.STM
SUCCESS: WiFi initialized

Connecting WiFi
Connecting to SSID 'POCO F3'
Attempt 1...
SUCCESS: WiFi connected

Initializing DHCP
IP address: 192.168.162.43
Mask: 255.255.255.0
Gateway: 192.168.162.215
SUCCESS: DHCP initialized

Initializing DNS client
DNS address 1: 192.168.162.215
DNS address 2: 8.8.8.8
SUCCESS: DNS client initialized

Initializing SNTP time sync
SNTP server 0.pool.ntp.org
SNTP time update: Dec 11, 2022 13:58:10.881 UTC
SUCCESS: SNTP initialized

Initializing Azure IoT DPS client
DPS endpoint: global.azure-devices-provisioning.net
DPS ID scope: 0ne008A1024
Registration ID: STM32-L475E
SUCCESS: Azure IoT DPS client initialized

Initializing Azure IoT Hub client
Hub hostname: iotc-869c6ac9-1fae-43ec-9a02-884fcd50fb.azure-devices.net
Device id: STM32-L475E
Model id: dtmi:azurerτος:devkitgsgstml4s5.2
SUCCESS: Connected to IoT Hub

Receive properties: {"desired":{"$version":1},"reported":{"deviceInformation":{"__t":"c","manufacturer":"STMMicroelectronics","model":"B-L475E-IOT01A","swVersion":"1.0.0","osName":"Azure RTOS"}
Sending property: $iothub/win/PATCH/properties/reported/?$rid=3{"deviceInformation":{"__t":"c","manufacturer":"STMMicroelectronics","model":"B-L475E-IOT01A","swVersion":"1.0.0","osName":"A
Sending property: $iothub/win/PATCH/properties/reported/?$rid=5{"ledState":false}
Sending property: $iothub/win/PATCH/properties/reported/?$rid=7{"telemetryInterval":{"ac":200,"ev":1,"value":10}}

Starting Main loop
Telemetry message sent: {"humidity":35.59,"temperature":21.84,"pressure":866.15}

```

Figure 2 : Sortie dans Termite – Démarrage

7) Observer les données et allumer la LED sur IoT Central :

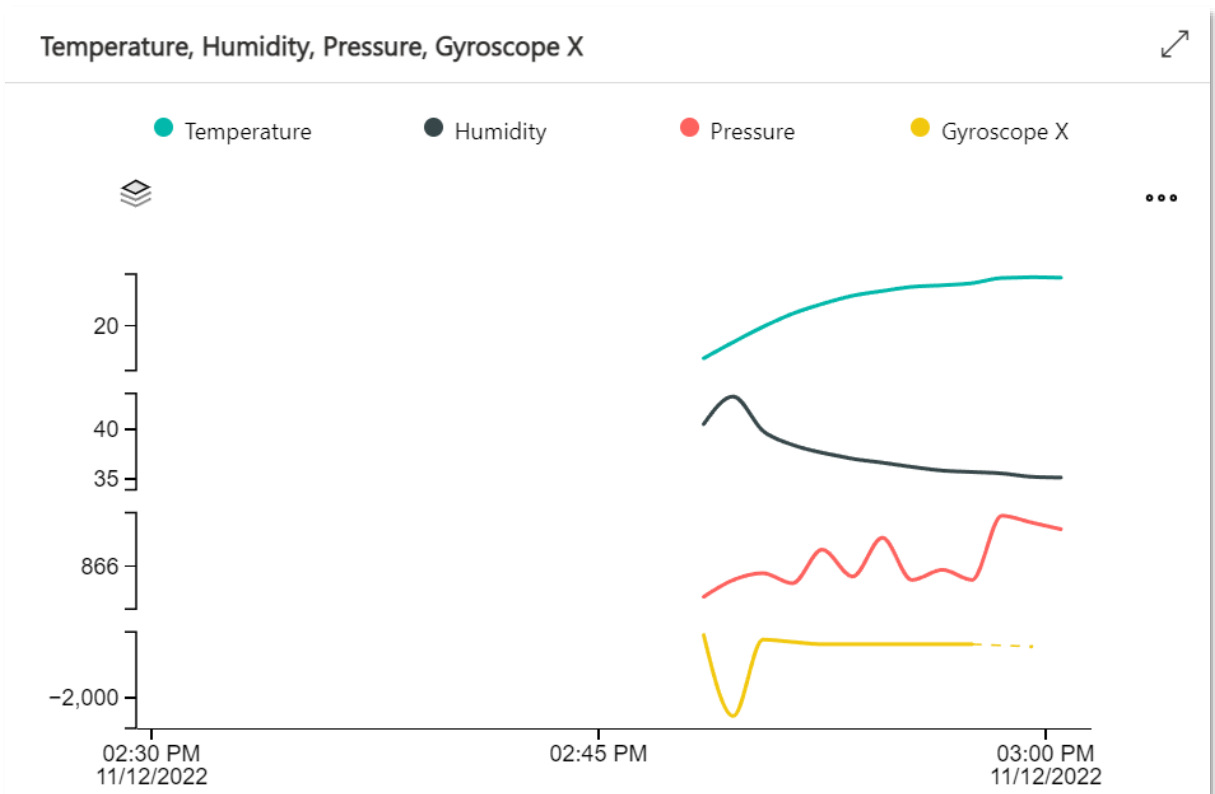


Figure 3 : IoT Central – Visualisation des données

```
Received command: setLedState
  Payload: true
  LED is turned ON
Sending property: $iothub/twin/PATCH/properties/reported/?$rid=9{"ledState":true}
```

Figure 4 : Sortie sur Termite - Allumage de la LED

STM L4S5 Getting Started Guide / Set LED state ⓘ

State ⓘ

Figure 5 : IoT Central - Allumage de la LED

Bibliographie

Site	Description
HWLibre - RTOS	Définition d'un RTOS
FreeRTOS	Définition d'un RTOS
Microsoft - Azure RTOS	Définition de Azure RTOS
Wikipédia - RTOS	Page wikipédia sur les systèmes d'exploitation temps réel
Wikipédia - MCU	Page wikipédia sur les microcontrôleurs
Wikipédia - IoT	Définition de l'IoT
Evaluation Assurance Level	Définition de l'EAL