Transformer un thermostat domestique en appareil intelligent : le défi de la connectivité

par Vikram Ramanna Senior Staff Applications Engineer, Cypress

Les consommateurs développent un intérêt croissant pour les appareils domestiques intelligents, en partie grâce à la nouvelle sophistication des technologies à contrôle vocal telles que Siri Remote® d'Apple ou Alexa Voice Service avec Amazon Echo, et des systèmes de commande à distance d'appareils domotiques comme les produits Nest de Google.

Cela explique que de nombreux fabricants d'appareils de contrôle, par exemple des thermostats, s'attaquent à la tâche de transformer un produit conventionnel 'bête' en un produit 'intelligent' et connecté (voir Figure 1). Ajouter de la connectivité implique d'utiliser pour la communication une technologie RF standard comme Wi-Fi ou Bluetooth Low Energy (BLE). Les concepteurs ne sont pas à court de choix en transmetteurs et modules Wi-Fi ou Bluetooth, de la part d'un grand nombre de fournisseurs. Les fabricants de composants RF proposent également des piles de protocole gratuites et prêtes à l'emploi pour la connectivité Wi-Fi, Bluetooth et Internet.

Et si un design de thermostat doit être rendu plus intelligent, les fabricants de microcontrôleurs proposent toujours un chemin de migration pour augmenter la puissance de traitement, la mémoire adressable et l'environnement périphérique du microcontrôleur équipant le thermostat.

Donc d'un point de vue purement matériel la transformation d'un design existant en un appareil connecté intelligent paraît facilement réalisable à l'aide des composants sur étagère disponibles sur ce marché florissant. La véritable question pour le développeur embarqué, alors, est sur le meilleur moyen de combiner hardware, firmware et logiciel d'application pour répondre aux attentes du client en terme de fonctionnalités et pour assurer une interopérabilité avec des plateformes telles que HomeKit d'Apple ou Works with Nest de Google. En réalité la réussite d'un design de thermostat intelligent implique bien plus que la connectivité sans fil : c'est le développement du logiciel qui occupera la plus grande partie du temps de l'équipe de design.

C'est la raison pour laquelle l'utilisation d'une plateforme de développement matériel/logiciel totalement intégrée est une aide essentielle aux ingénieurs pour limiter la durée et les difficultés rencontrées lors du design d'appareils domestiques intelligents connectés. Cet article décrit les éléments constitutifs et l'opération d'une telle plateforme.



Figure 1 : La prochaine génération de thermostats domestiques supportera de nombreuses interfaces de communication mono- et bi-directionnelles

Les protocoles pour appareils connectés

Pour supporter les types de contrôle à distance et vocal et les autres fonctionnalités intellligentes disponibles avec des plateformes comme HomeKit ou Works With Nest, un thermostat a besoin d'une connexion Internet, généralement fournie via une passerelle telle qu'un routeur Wi-Fi.

L'appareil devra également supporter les protocoles d'application tels que HTTP, SMTP, NTP et MQTT pour la communication avec le Web et avec des serveurs d'application sur l'Internet. Au niveau de la couche application le thermostat peut également incorporer des éléments logiciels spécifiques tels que HAP (Homekit Accessory Protocol) d'Apple.

La sécurité est une exigence primordiale des objets connectés, pour protéger la vie privée de l'utilisateur et empêcher les intrusions sur son réseau privé. Les transactions et les échanges de messages entre l'appareil et le cloud peuvent être effectués via SSL (Secure Sockets Layer) et TLS 1.2 (Transport Layer Security) ou d'autres protocoles sécurisés.

En support des opérations complexes de la couche applicative, de la couche sécurité et de la couche communication l'architecture système a besoin d'un système d'exploitation (OS) pour gérer les priorités et l'allocation mémoire. Dans un appareil à ressources limitées tel qu'un thermostat ce sera un OS temps réel doté d'une empreinte mémoire minimum.

Et voici le défi du concepteur de thermostat : non seulement garantir une connexion Wi-Fi robuste avec le routeur – et parfois une connexion Bluetooth Low Energy à des périhériques tels que des capteurs sans fil – mais aussi implanter de multiples protocoles et le logiciel d'application dans un environnement imposant une interopérabilité sans faille avec des appareils et des services tierce partie connectés à Internet.

C'est une tâche de développement considérable, avec de nombreuses difficultés techniques à maîtriser, pas toutes familières. C'est la raison pour laquelle Cypress a développé la plateforme WICED™ (Wireless Internet Connectivity for Embedded Devices).

Une plateforme unifiée pour le développement du matériel et du logiciel

L'objectif de WICED est de fournir une plateforme unifiée pour le développement d'appareils connectés à Internet via une connexion Wi-Fi, Bluetooth ou ZigBee. Que signifie "unifiée" ? La plateforme WICED regroupe des éléments hardware et software totalement interopérables, elle est constituée de :

- Composants et modules RF de Cypress, certains avec un microcontrôleur embarqué, permettant une connectivité via Wi-Fi, Wi-Fi + Bluetooth, Bluetooth + Bluetooth Low Energy, ou Bluetooth + ZigBee.
- Une architecture logicielle complète couvrant la connectivité, la sécurité et les applications (voir Figure 2).

- Le support natif des systèmes d'exploitation ThreadX, NuttX et FreeRTOS. ThreadX est livré gratuitement sous forme de binaires partageables. Les développeurs peuvent également porter d'autres OS sur la plateforme WICED.
- Le support natif des piles réseau NetX, NetXDuo et LwIP. NetX et NetXDuo sont livrées gratuitement sous forme de binaires partageables. D'autres piles TCP/IP peuvent être facilement portées sous WICED.
- Un kit de développement logiciel (SDK) complet bâti sur l'environnement de développement intégré open-source Eclipse, donnant accès aux développeurs à l'ensemble des fonctionnalités d'Eclipse pour le développement et la mise au point d'applications utilisant les produits RF de Cypress.
- Des cartes de développement compatibles Arduino auxquelles les utilisateurs peuvent connecter des extensions capteurs et d'autres périphériques.
- Des designs de référence fournis par des développeurs WICED partenaires.
- Des exemples de code applicatif explicitant la mise en oeuvre de tous les protocoles courants de communication, de réseau et d'applications.
- Des exemples d'applications de communication avec des fournisseurs de services cloud tels que Web Services d'Amazon, Azure® de Microsoft et Bluemix®d'IBM.
- Une procédure fiabilisée pour les mises à jour OTA (Over-The-Air).
- Une documentation complète des API et des exemples d'application.

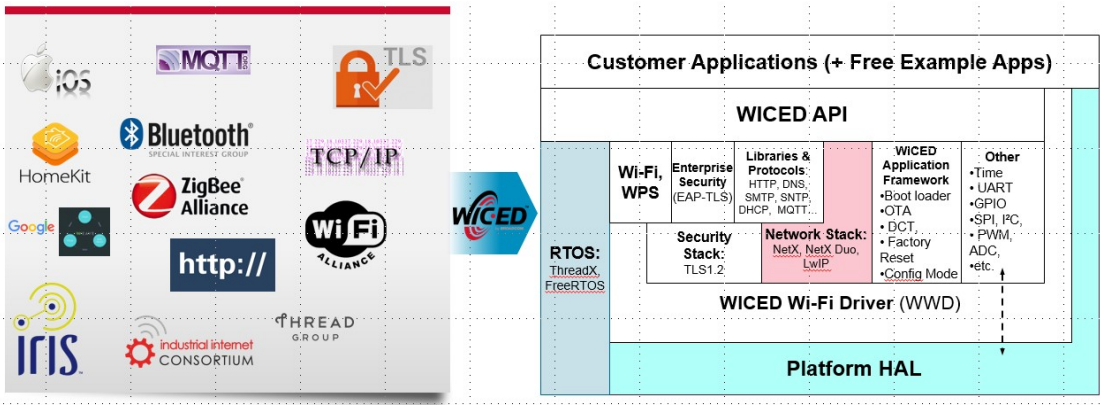


Figure 2 : L'architecture logicielle WICED

Tous les éléments sont rigoureusement testés de telle sorte que les utilisateurs soient sûrs, par exemple, que le code WICED HAP fonctionnera parfaitement sous environnement logiciel ThreadX sur un CYW43907, contrôleur Wi-Fi à coeur ARM® Cortex®-R4.

La plateforme WICED supporte différentes architectures hardware, laissant le concepteur libre d'implanter son design dans un module dédié avec microcontrôleur embarqué, dans un microcontrôleur

avec RF totalement intégrée, ou dans un processeur hôte externe (voir Figure 3). Les microcontrôleurs supportés par le kit SDK incluent :

- Cypress : famille FM4
- STMicroelectronics : familles STM32F2 et STM32F4
- Microchip : AT91SAM4S16B
- NXP Semiconductors : Kinetis K61, LPC17xx, famille LPC18xx

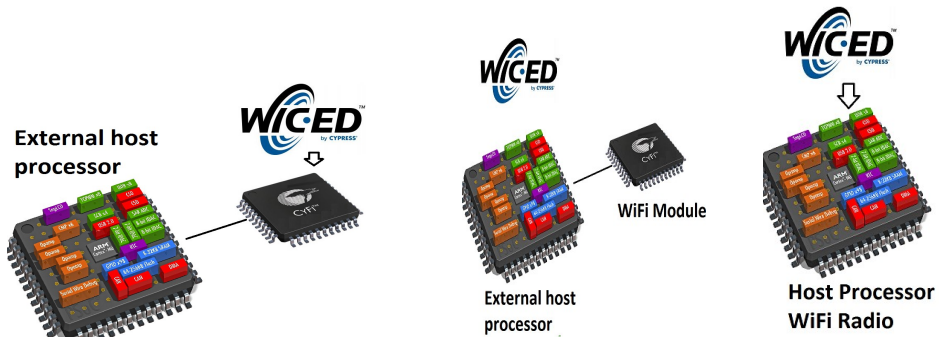


Figure 3: Architectures hardware supportées par la plateforme WICED

Thermostat intelligent : le processus de développement WICED

- Un bon point de départ consiste à interconnecter un capteur de température, un relais et un module WICED. La Figure 4 montre l'arrangement correspondant, consistant en un capteur de température avec interface I^2C connecté à une carte d'évaluation CYW43907AEVAL1F, équipée d'un module Murata LBWA1UZ1GC-958 à base du système-sur-puce Wi-Fi CYW43907, le tout complété d'un petit afficheur pour visualiser l'état du système et la température.

Le pilote logiciel écrit pour le capteur de température peut être placé dans le dossier des bibliothèques du SDK WICED, permettant à d'autres applications de l'utiliser. Ce même dossier contient le pilote de l'afficheur, fourni avec le SDK.

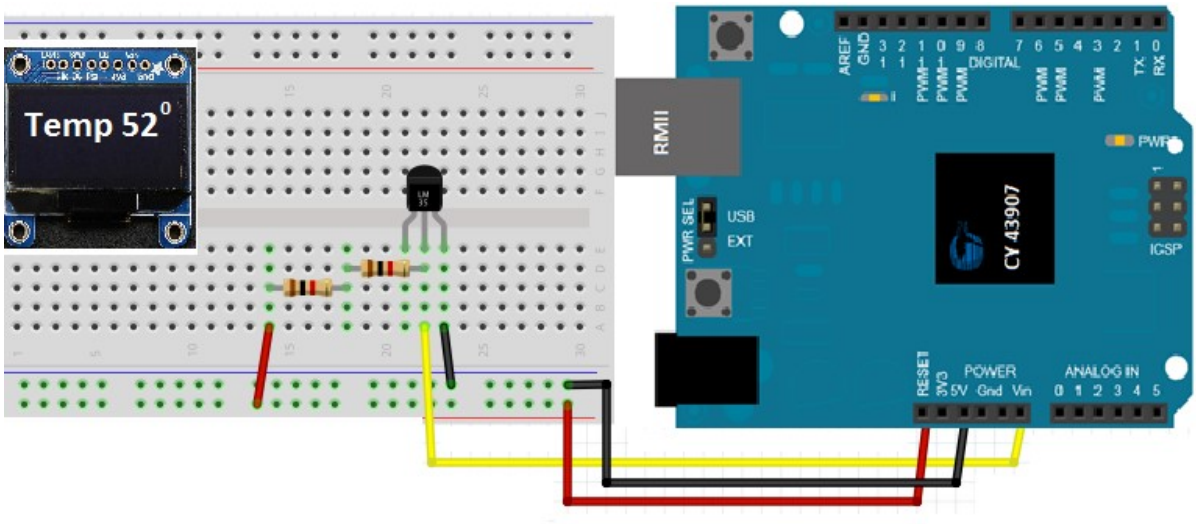


Figure. 4: Configuration hardware initiale du projet thermostat intelligent

Le dossier application inclut un exemple d'application explicitant comment utiliser le protocole I^2C pour lire la valeur de la température fournie par le capteur. La température ambiante peut ensuite être accédée via des appels API basiques.

Les E/S sont configurées dans les fichiers plateforme. Le fichier platform.c regroupe les affectations d'E/S et leur configuration respective. Lors de la configuration des E/S I^2C il est nécessaire de définir :

- le port I2C auquel est attaché le capteur de température
- les broches SDA et SCL
- le pilote à utiliser

Un autre extrait de code applicatif permet d'intégrer au design un affichage 128x128 pixels. En combinant les deux applications il est alors possible de lire la température ambiante via l'interface I2C et de la visualiser sur l'afficheur.

Pour écrire cette application la procédure est la suivante :

1) initialiser WICED
2) initialiser le capteur I^2C
3) initialiser l'afficheur
4) lire la valeur de la température via l'I^2C
5) visualiser cette valeur sur l'afficheur via I^2C ou SPI

Jusqu'à maintenant le système ne met en oeuvre qu'une fontion thermomètre basique, sans intelligence. L'étape suivante consiste à ajouter l'intelligence : par exemple comparer la température ambiante à une température de consigne et activer la mise en route d'un chauffage via le relais si elle est inférieure.

Le développeur voudra ensuite permettre au thermostat de se connecter à un réseau Wi-Fi à portée. L'initialisation de la pile réseau offre à l'utilisateur un choix du mode de fonctionnement :

a) interface STA – le thermostat est un client de la station
b) Soft AP – le thermostat intègre un serveur web
c) SoftAP et STA – appelé aussi mode APSTA, dans lequel le thermostat peut opérer à la fois en mode serveur et en mode client

APSTA est disponible dans le SDK WICED sous la forme d'un extrait d'application. Cela signifie que le thermostat peut embarquer un serveur web et permettre à d'autres clients de se connecter à lui ; en même temps il peut se connecter en tant que client au point d'accès résidentiel pour une connexion à Internet.

Le SDK WICED simplifie la tâche du développeur pour héberger un serveur http dans le thermostat avec une API REST pour obtenir (GET) ou définir (POST) la température. Cette fonction rend le thermostat accessible par d'autres appareils du réseau.

Par exemple une petite application sur un appareil mobile peut être développée pour lui permettre de fixer la température à l'aide d'une simple requête POST qui dirait quelque chose comme :
http://mysmarthome.com/thermostat1/setTemp?value=65

L'application peut également demander au thermostat d'indiquer quelle est la valeur actuelle de la consigne de température, par une requête GET telle que :
http://mysmarthome.com/thermostat1/getTemp?value=65

La page html développée pour héberger un serveur web est placée dans le dossier des ressources avec les fichiers javascript, css et image correspondants. Il faut l'inclure dans le Make Target de l'application comme un objet ressource, permettant au serveur d'effectuer le rendu et l'affichage quand un client se connecte à l'appareil.

Il y a deux mécanismes pour connecter le thermostat au point d'accès souhaité :

- fournir le SSID et le mot de passe du point d'accès dans le fichier de configuration
- demander à tout client tel qu'un téléphone portable connecté au softAP hébergé par le thermostat d'autoriser l'utilisateur à entrer le mot de passe du point d'accès choisi. Le thermostat redémarre alors et stocke les identifiants du point d'accès dans une table de configuration.
- le SDK WICED fournit également l'option d'une sécurisation WPS pour l'enregistrement d'un nouvel appareil par un point d'accès.

Une fois connecté à Internet le thermostat peut commencer la remontée des valeurs de température vers un service cloud de données.

Bien entendu une connexion Internet ouvre également le thermostat à des flux de données entrants, tels que prévisions météo, mises à jour firmware, réglage de date et d'heure via le protocole NTP. WICED supporte également le protocole SMTP permettant à l'utilisateur de configurer un compte de messagerie pour l'envoi de notifications à son adresse email.

Le SDK accélère le développement système

Le hardware WICED – composants radio et microcontrôleurs RF disponibles auprès de Cypress – constitue la base fondamentale d'un appareil connecté intelligent. Mais comme le montre cet exemple la valeur ajoutée de la plateforme WICED repose sur l'intégration d'une architecture logicielle complète supportée par un SDK, adossé à l'environnement IDE Eclipse et intégrant toute une gamme d'exemples d'applications. C'est cette combinaison de hardware, de software, d'exemples et d'outils qui rend le développement d'appareils connectés intelligents bien plus rapide et plus simple qu'auparavant.