

# Så gör du hemtermostaten smart



## Ta avstamp i en plattform

**N**ya röststyrda tekniker som Apples Siri Remote och Alexas rösttjänst Amazon Echo – men också system för fjärrstyrning av hemmet som Google Nest – har fått konsumenter sugna på att skaffa smarta produkter för hemmet. Det betyder att företag som tillverkar produkter som styr funktioner i hemmet, såsom termostater, också vill ta fram trådlöst uppkopplade och styrbara versioner (se figur 1).

För detta krävs rf-teknik som wifi eller Bluetooth LE.

Många halvledartillverkare erbjuder lämpliga transceivrar och moduler, liksom färdiga och fria protokollstackar för wifi, Bluetooth och Internetanslutning. Fast om termostaten ska bli smart måste även dess processor uppgraderas – det behövs mer processorkraft, mer adresserbart minne och fler periferafunktioner.

**DEN STORA UTMANINGEN** för den som utvecklar är ändå att kombinera nödvändig hårdvara, firmware och mjukvara för att kunna skapa de funktioner som användarna vill ha. Lösningen ska också fungera med plattformar som HomeKit från Apple och Works with Nest från Google, vilket betyder att



### Av Vikram Kumar Ramanna, Cypress Semiconductors

Vikram Kumar Ramanna, med titeln Principal Software Engineer, har över fem års erfarenhet av mjukvaruutveckling och systemkonstruktion. Genom åren har han arbetat med en mängd trådlösa tekniker, som wifi, Bluetooth, Zigbee och cellulärt.

konstruktörerna framför allt måste lägga tid på att utveckla mjukvaran.

Detta är också anledningen till att en utvecklingsplattform som är komplett och både inkluderar hård- och mjukvara, gör det smidigt att skapa nya smarta produkter för hemmet. Den här artikeln beskriver en sådan plattform.

För att kunna stödja olika typer av fjärr- och röststyrda funktioner, liksom automationsfunktionerna i plattformar som de nyss nämnda från Apple och Google, måste en termostat vara ansluten till Internet. Det är den vanliga via en gateway, som en wifi-router.

För kommunikationen med webb och tillämpningsservrar måste termostaten stödja protokoll som HTTP, SMTP, NTP och

MQTT och kanske även plattformsspecifika protokoll som Apple HomeKit Accessory Protocol (HAP).

Lösningen måste skydda mot intrång i hemnät och skydda användarnas integritet. Säkra transaktioner och utväxling av meddelanden mellan enhet och moln kan göras över SSL (Secure Sockets Layer) och TLS 1.2 (Transport Layer Security) eller andra säkra kommunikationsprotokoll.

**DEN SOFISTIKERADE HANTERINGEN** av tillämpnings-, säkerhets- och kommunikationslager kräver att systemarkitekturen inkluderar ett operativsystem (OS) som sköter prioritering och minnesallokering. I en termostat där resursen är begränsad handlar det om ett realtids-OS som kräver litet minnesutrymme.

Under konstruktionsarbetet dyker flera utmaningar upp. Konstruktören måste skapa en robust wifi-anslutning till en router, och i vissa fall även en WIBREE-anslutning till trådlösa sensorer. Likaså måste han eller hon implementera flera protokoll och tillämpningsprogram i en miljö som kräver robust interoperabilitet med Internetanslutna enheter och tjänster från tredjepart.

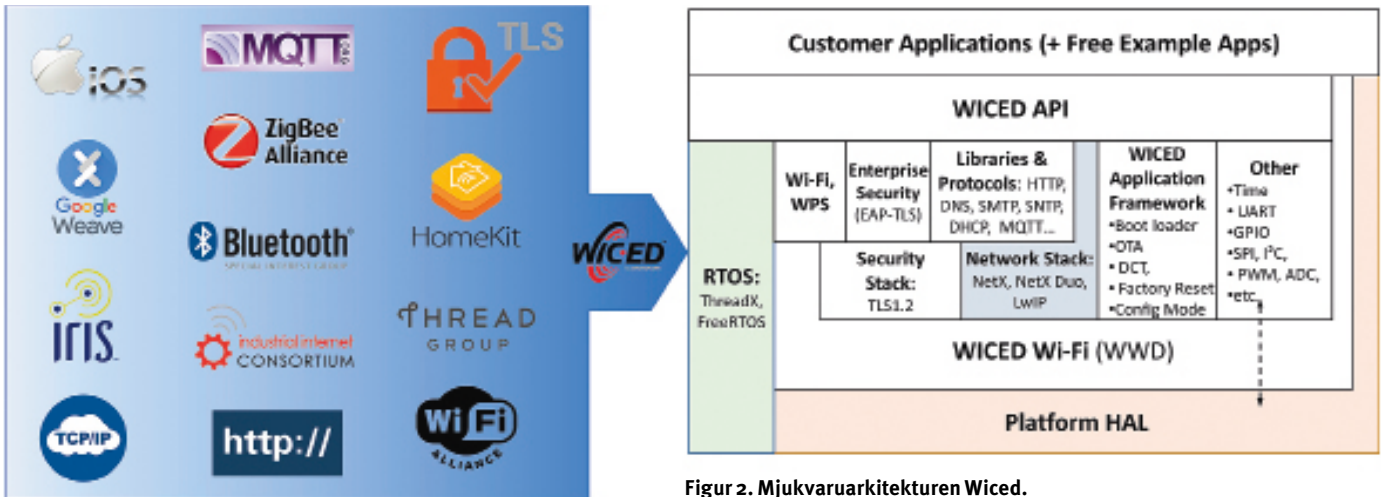
Detta är en verkligt utmanande uppgift och anledningen till att plattformen Wiced (Wireless Internet Connectivity for Embedded Devices) från Cypress utvecklats.

**WICED ÄR EN KOMBINERAD** plattform för utveckling av Internetanslutna produkter som använder wifi, Bluetooth eller Zigbee. "Kombinerad" innebär att plattformen inkluderar fullt kompatibla hård- och mjukvaruelement och består av:

- Rf-kretsar och -moduler (vissa med MCU) från Cypress som ansluter via wifi



Figur 1. Framtida hemtermostater kommer att stödja flera kommunikationsgränssnitt, både en- och tvåvägs.



Figur 2. Mjukvaruarkitekturen Wiced.

och Bluetooth, Bluetooth LE eller Bluetooth och Zigbee

- En omfattande mjukvaruarkitektur för anslutning, säkerhet och tillämpningsprogram (se figur 2)
- Stöd för operativsystemen ThreadX, NuttX och FreeRTOS. Utvecklare kan även portera andra RTOS. ThreadX distribueras som en royaltyfri binär i Wiced
- En komplett utvecklingsmiljö (SDK) byggd ovanpå den öppna utvecklingsmiljön Eclipse IDE. Därmed kan alla funktioner i Eclipse användas för att utveckla och felsöka i tillämpningar som är baserade på Cypress rf-produkter
- Arduinokompatibla socklar som kan ansluta sensorköldar och annan kringutrustning
- Referenskonstruktioner från tredjepart
- Exempelkod som demonstrerar implementation av de vanligaste kommunikations-, nätverks- och tillämpningsprotokollen
- Exempel på appar för kommunikation med molntjänsteleverantörer som Amazon Web Services, Microsoft Azure och IBM Bluemix



- En felsäker process som stöder over-the-air-uppdateringar
- Full dokumentation för API och exempelprogram

**ALLA DELAR ÄR NOGGRANT** testade och verifierade.

Wiced-plattformen stöder dessutom olika hårdvaruarkitekturer. Den kan implementeras i en särskild modul med en styrkrets ombord, i en rf-styrkrets eller i en extern vädprocessor (se figur 3). De processorer som stöder utvecklingsmiljö är Cypress FM4-serie, ST Microelectronics STM32F2- och STM32F4-serie, Microchips AT91SAM4S16B liksom NXP:s Kinetis K61 och LPC17xx- samt LPC18xx-serie.

För den som ska bygga en smart termostat är en bra början att ansluta en temperatursensor, ett relä och Wiced-modul till varandra (se figur 4).

Drivrutinen som är skriven för temperatursensorn kan placeras i biblioteksmappen till Wiced-plattformens utvecklingsmiljö, vilket tillåter att andra tillämpningar använder den. Samma mapp innehåller också drivrutinen för skärmen som SDK:t stöder.

I mappen med källkodsexempel hittar du funktioner som använder I2C-protokollet för att läsa värden från temperatursensorn. Enkla API-anrop kan sedan användas för att läsa rumstemperaturen.

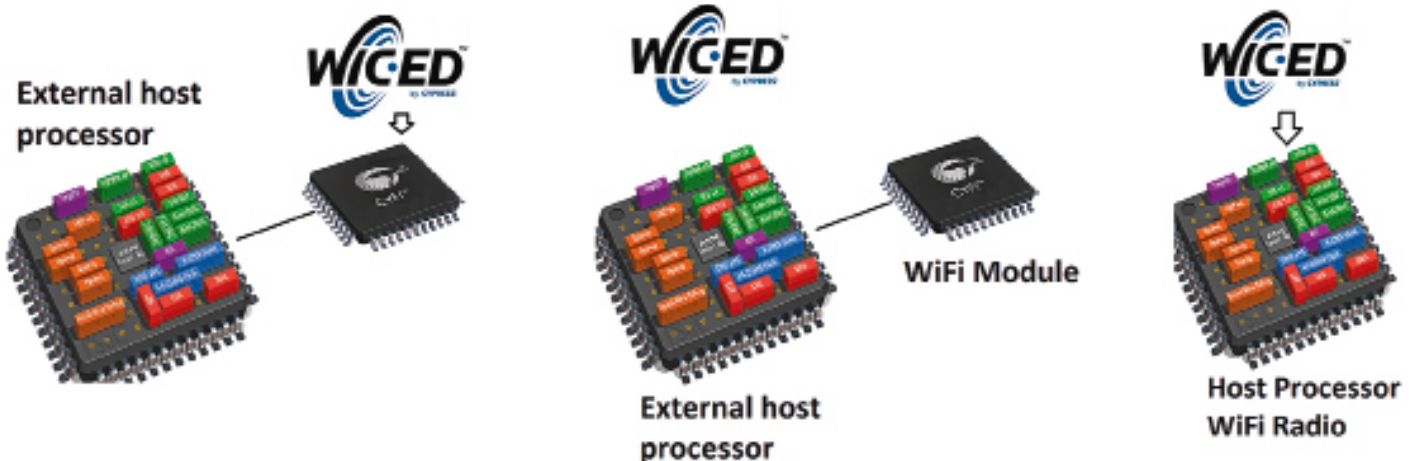
IO:s kan konfigureras i plattformens filer. Filen platform.c mappar alla pin-out liksom deras konfigurationer. När I2C-anslutningarna konfigureras är det nödvändigt att kontrollera:

- vilken I2C-port som temperatursensorn är förbunden med
- SDA- och SCL-anslutningarna
- vilken drivrutinen som ska användas

**ETT ANNAT KODEXEMPEL** ansluter en skärm på 128 x 128 punkter. Genom att kombinera de två går det att läsa rumstemperaturen över I2C-gränssnittet och sedan visa värdet på displayen.

Så här ser flödet ut:

- Initiera Wiced-plattformen
- Initiera I2C-enheten
- Initiera skärmen
- Läs värdet från temperatursensorn via I2C
- Visa värdet på displayen via I2C eller SPI



Figur 3. Hårdvaruarkitekturer som stöds av Wiced-plattformen.



**NÄSTA STEG ÄR** att addera intelligens. Det kan exempelvis vara att jämföra önskad temperatur med den uppmätta temperaturen. Om temperaturen är för låg ska värmen kopplas på via relät.

Därefter är det dags att ansluta termometern till ett wifinät inom räckhåll. I detta läge kan utvecklaren välja ett av flera driftsätt:

- STA-gränssnitt – stationsbaserat, där termostaten är en klient
- Soft AP – termostaten agerar värd för en webbserver
- Soft AP och STA – känt som APSTA-mode, här kan termostaten fungera som både server och klient

**APSTA FINNS TILLGÄNGLIG** i Wiceds utvecklingsmiljö som kodrader. Det betyder att termostaten kan fungera som webbserver och tillåta att andra klienter ansluts till den. Samtidigt kan den som klient ansluta till Internet via hemmets accesspunkt.

Wiced SDK gör det således enkelt att låta termostaten agera HTTP-server, så att önskad temperatur kan hämtas via REST API-anrop som GET eller POST. Denna kraftfulla egenskap gör termostaten tillgänglig för andra enheter i nätet.

Till exempel går det att skriva en liten app till en bärbar enhet, som gör att den kan ställa in temperaturen med hjälp av en POST-begäran. Den skulle kunna se ut så här: <http://mysmarthome.com/thermostat1/setTemp?value=65>

Appen kan även låta termostaten fråga vilken temperatur den mobila enheten har med GET till följande adress

<http://mysmarthome.com/thermostat1/getTemp?value=65>

HTML-sidan som är utvecklad för att vara värd för en webbserver placeras i resursmappen tillsammans med nödvändig javascript, CSS och bildfiler. Allt detta måste inkluderas i appen "Make Target" som ett resursobjekt, vilket tillåter servern att återge och visa när en klient ansluter till enheten.

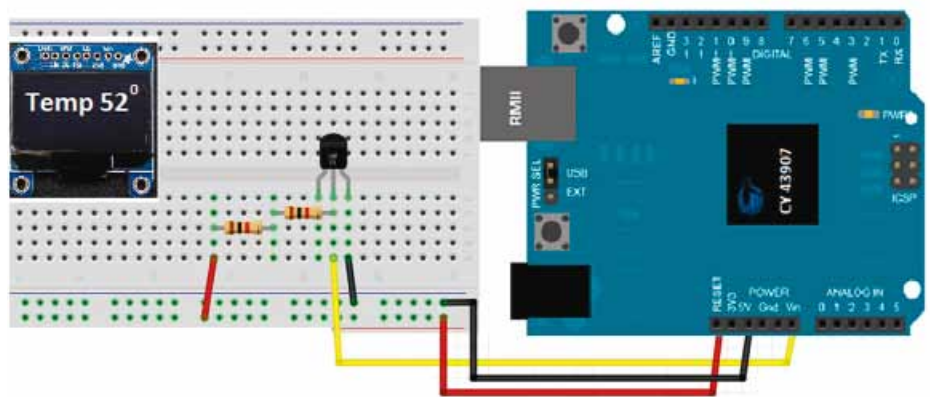
**DET FINNS TVÅ SÄTT** att ansluta termostaten till en önskad åtkomstpunkt:

- lägg SSID och lösenord till åtkomstpunkten i konfigurationsfilen
- be valfri klient, exempelvis en mobiltelefon som är ansluten till Soft AP hos termostaten, att tillåta användaren att skriva in lösenordet för den valda kopplingspunkten. Därefter omstartas termostaten och lagrar åtkomstpunkten i en konfigura-

tionstabell.

- Wiced SDK ger också möjlighet till WPS-säkerhet, för att registrera en ny enhet till en åtkomstpunkt

**NÄR TERMOSTATEN VÄL** är ansluten till Internet kan den börja ladda upp temperaturvärdet till en molntjänst. Att Internetanslutning finns på plats öppnar dessutom för möjligheten att hämta hem dataflöden till programmet, som väderrapporter, uppdateringar av firmware, eller tid och datum via NTP. Wiced stöder också SMTP, som tillåter användaren att konfigurera ett emailkonto för att skicka meddelanden till sin email-adress. ■



**Figur 4.** En första hårdvarukonstruktion av en smart termostat. En temperatursensor med I2C-gränssnitt ansluts till ett utvärderingskort och en liten display som visar systemstatus och temperatur. Utvärderingskortet är bestyckat med Muratas modul LBWA1UZ1GC-958 som är baserad på Cypress wifi-chip CYW43907.