Dathan Pompa  
CS 350 Emerging Sys Architecture

12/22/2024

Report

1. Explain how the thermostat supports the peripherals used in the project. Ensure you have included all the required details from the scenario in your report. Discuss the three outlined hardware architectures: TI, Microchip, and Freescale.  
     
   - My thermostat supports the peripherals because it performs GPIO, UART, I2C, and Timer tasks. GPIO will control the LEDS and detect the buttons being pressed to adjust our set point. UART helps communicate with the external system to show the real time status updates on the terminal. I2C helps to interface with the temperature sensor, it reads the ambient temperature data and when it's initialized, it gets identified through the connected sensor from a list. The Timer, this is what allows the time intervals to take place, it will be used for updating LEDs, checking for buttons, and it will send data so we can see how responsive our system is.  
   So for the TI devices, these are made well for Internet of Things devices because it offers a wide range of tools like Wi-Fi, peripheral support and it won't consume large amounts of power. It has Flash and RAM, flash to store the code and RAM to run real time operations. Microchips like PIC32MX can be useable for a thermostat program, it has Flash 512 KB to 2MB and RAM 128 KB to 512KB also, it also has Peripheral support like MPLAB Harmony which can help give support for the GPIO, UART, I2C and Timer tasks. This microchip also has Wi-Fi capabilities with external modules that will connect to it through the PIC microcontroller with SPI or UART. Lastly Freescale has Kinetis series microcontrollers, for example I'll talk about the K64F, it has Flash and RAM. Flash is up to 2MB, and RAM is 256KB up to 512. Its peripheral support offers the Kinetis SDK that has libraries for what we need, GPIO, UART, I2C and Timer. For Wi-Fi, they will pair up with NXP external Wi-Fi module or you could get something third party, this will allow you to connect to the internet and the cloud, using protocols like MQTT, HTTP and even WebSocket's.
2. Explain how the thermostat connects to the cloud via Wi-Fi. Discuss all three architectures in your work.  
     
   - The thermostat will be connecting to the cloud over Wi-Fi which lets us remotely monitor it and control it. For TI, this has built in WI-FI, and we can use protocols MQTT or HTTP to send our state updates and so it can get commands from us. For Microchips, the PIC32MX will need an external module for WI-FI, for example the ATWINC1500 will help us connect to SPI and UARTs, this gives us access to send HTTPS or MQTT protocols. For Freescale, the Kinetis series will be paired with Wi-Fi modules usually and one for example is named ESP8266 and this will let us connect to the cloud. It will send updates every so often and will listen for commands which helps give us a streamlines experience for integrating with cloud services.
3. Discuss the architecture’s Flash and RAM that supports the code. Include all three architectures in your discussion.  
     
   - For TI, these give us 1MB of Flash which is essential for storing our code such as the task scheduler code, the Wifi stack, and all the drivers needed to operate. The 256KB of RAM will be used for handling operations in real time like switching tasks, sensor data processing and when the data buffers. The Microchip usually gives 512KB and 2MB of Flash storage and around 512KB of RAM. The Freescale is going to give us up to 2MB of Flash and 512KB of RAM.