Dominic Clapper

CS350

August 13, 2023

Temperature/LED Project Reflection

The thermostat prototype that I developed in this project utilized various types of peripherals. In this project, I used the TI development board that I bought through the school store. The board I used for testing development was a CC3220S, which is a microcontroller development board that supports wireless connectivity. This board had LEDs, buttons, and a temperature sensor, which were all necessary to properly complete the project. The LEDs and buttons were great support for GPIO usage, and the temperature sensor supported the I2C usage. For UART, the connection cord provided with the TI board was very useful for allowing the microcontroller/board to connect with my computer. All these peripherals were essential to making it so my thermostat application could work. Despite this, the board, in my experience, is meant for a development environment, and not for a production level environment. This leads me to the point that if we were to make this program usable in an actual production version, then it would be highly recommended to use an actual microcontroller. Basically, instead of using a microcontroller development board, we would be using a smaller, more compact microcontroller that could fit inside of a thermostat.

Now, TI does have microcontrollers that support the necessary peripherals and are very compact, but there are other businesses like Freescale and Microchip that have developed microcontrollers that meet similar requirements. For example, there are microcontroller series made by Freescale (who are now owned by NXP Semiconductors) that are good options for our thermostat program. Some of these series of microcontrollers use a standard architecture/protocol which is called Zigbee. (Zigbee is essentially a protocol that is used in wireless control and monitoring solutions to allow devices to communicate with one another.) One such microcontroller series that uses the Zigbee protocol architecture, and is made by Freescale, is the JN516x series. The JN516x series of microcontrollers do support the UART peripheral (2 UARTs), some of the GPIO peripheral processes (20 pins), I2C, and even has wireless connectivity capabilities. Overall, the JN516X series of microcontrollers are a good option for a production version in terms of its support for peripherals and its other capabilities. Another good microcontroller to consider in this situation is the PIC32CX-BZ2 microcontroller which is made by Microchip. This microcontroller also utilizes the Zigbee protocol to facilitate certain wireless connectivity and communication processes, which makes it a viable option. Not only that, the PIC32CX-BZ2 microcontroller supports USART, which is pretty much an extended version of the UART peripheral, I2C, and GPIO peripherals (29 pins). This microcontroller from Microchip is definitely a possible option for the production level microcontroller we need to make our thermometer work. The last option I want to mention is CC1350 microcontroller, which is a part of the CC13x0 series of microcontrollers from TI. This microcontroller supports Wi-Fi connectivity, Bluetooth connection, Ethernet connection, as well as Zigbee connection protocol capabilities. According to the guide and webpages related to the CC1350, the microcontroller does support the UART peripheral (1 UART), GPIO peripherals (31 pins), and I2C.

Beyond the support for peripherals, there are also things like Flash and RAM to consider when determining which microcontroller to select. For the JN516x series microcontrollers, they seem to have embedded FLASH and memory in varying sizes. If we were to go with the JN516x, there is 32 KB of RAM, and about 262Kbytes of FLASH memory. The FLASH memory in this microcontroller is separated into two parts, a very small part for boot code, and a larger part for application code. For the PIC32CX-BZ2, there is 1 MB of FLASH memory with ECC (Error Correction Code), 32 KB of non-volatile FLASH memory, and 128 KB of RAM. And, for the CC1350 microcontroller, there is 128 KB of FLASH memory, as well as 28 KB of RAM. All three microcontrollers have a fair amount of RAM and FLASH memory that would allow for powerful operation of certain devices and their software, such as our thermostat.

The last thing to mention before making a recommendation is the ways in which the microcontrollers connect to the cloud via Wi-Fi. As I mentioned before, all three of the microcontrollers I brought up do have wireless connectivity capabilities, which is very beneficial for a smart device like our thermostat. Due to them being wireless and having certain software built into them, they can connect to other devices and the cloud. The JN516x uses the Zigbee connection protocol I mentioned before, alongside a wireless transceiver. The PIC32CX-BZ2 is similar in that it uses Zigbee as a part of the process in connecting this device to other devices such as the cloud. The PIC32CX-BZ2 can use a Bluetooth interface to handle connections between the cloud and the microcontroller as well. And from what I have found, the CC1350 is capable of using a Bluetooth interface to help with connections to other devices. So, from the research I have done, there a quite a few ways in which a production level microcontroller can connect to a cloud or other devices. Most of these ways seem to be through some form of middleware protocol or architecture.

If I had to make a recommendation between the three microcontrollers I mentioned, I personally would go with the TI CC1350 microcontrollers because I have a little bit of experience working with same brand products like the TI development board kit. All three microcontrollers I mentioned meet very similar standards and are all great choices. The TI CC1350 microcontrollers do have quite a fair amount of FLASH memory and RAM, which will be very handy for fast operation of the software. Also, because of how much peripheral support the TI CC1350 microcontroller has, I feel more confident in this decision. Thus, I would go with the TI CC1350. (However, there are other factors like the price and necessary amount to consider before making such a decision.)

References

*CC1350*. CC1350 data sheet, product information and support | TI.com. (n.d.). [https://www.ti.com/product/CC1350#params](https://www.ti.com/product/CC1350)

NXP Semiconductors. (n.d.). *Data sheet: JN516x - NXP Semiconductors*. NXP.com. <https://www.nxp.com/docs/en/data-sheet/JN516X.pdf>

*PIC32CX1012BZ25048 - Microchip Technology*. Microchip.com - Product - PIC32CX1012BZ25048. (n.d.). <https://www.microchip.com/en-us/product/PIC32CX1012BZ25048>

*PIC32CX-BZ2 and WBZ45 Family Data Sheet*. Microchip technology. (n.d.). <https://ww1.microchip.com/downloads/aemDocuments/documents/WSG/ProductDocuments/DataSheets/PIC32CX-BZ2-and-WBZ45-Family-Data-Sheet-DS70005504.pdf>

*Support Resources for JN516X MCUs*. Support Resources for JN516x MCUs | NXP Semiconductors. (n.d.). <https://www.nxp.com/products/wireless/zigbee/support-resources-for-jn516x-mcus:SUPPORT-RESOURCES-JN516X-MCUS>