Chipkit Portable GPS

By Justin Lex and Petr Kocián

Accomplishments

- Able to read and write to UART bus
- Able to interpret UBX-NAV-PVT packets from GPS
- Able to tell time from the GPS, using either a RTC or GPS synchronization
- Able to convert 32x32 bitmaps into the image format used by display_image()
- Able to rotate images by a specified amount of radians
- Implemented a page-based UI, using button interrupts
- Implemented a date page that tells the current date
- Implemented a time page that tells the current time
- Implemented a humorous spinner page that shows a spinning arrow

Solution

In implementing our portable GPS, we faced many changes due to the difficulty of debugging the UART communication between the Chipkit and the GPS.

Our goal of creating a user interface was fully successful, however, and we were able to implement a UI that allows the user to switch between pages of data. This UI uses the Change Notification subsystem of the Chipkit to send interrupts immediately when a button is pressed, allowing for an responsive, asynchronous UI. The graphical parts of the UI are also nearly complete, we were able to implement a compass graphic that can be rotated by arbitrary amounts and printed to the screen. Due to the missing compass functionality, this was instead used to create a spinning arrow animation on one of the UI pages. This graphic utilizes two functions that we made, one that can rotate a given 32x32 bitmap by an arbitrary amount (utilizing rotation matrices), and one that can convert a 32x32 bitmap into the array of 128 bytes that display_image() uses.

On the I/O front, we were eventually successful in allowing for bi-directional communication on the UART bus. Packets can be sent and received on the bus, and we are able to receive and interpret “UBX-NAV-PVT” packets from the gps module, which gives us information such as the current UTC time, down to the nanosecond, as well as our current GPS coordinates. This communication is all done over the UART2 bus integrated into the Chipkit, and is controlled with special functions that do a variety of operations, ranging from enabling the bus, enabling RX or TX operation, sending predefined packets, and interpreting packets in the UBX protocol.

Verification

We have verified the code works properly by moving through the UI, making sure that it performs as expected, and that the time and date the Chipkit prints is accurate. We also verified that the UART bus works correctly by connecting the serial port up to a PC and printing and reading characters from a serial console.

Contributions

Though we both made our own contributions to the majority of the code, helping each other as needed, we did split up the code into separate UI and I/O portions. Justin handled the I/O portions, writing the code for the UART bus and packet interpretation, and Petr handled the UI portions, writing the code for the CN subsystem, designing the UI, and crafting the graphics algorithms used.

Reflections

Communicating with GPS module was by far the most challenging part of this project. Though there was good documentation for the GPS module and even a PC-based control panel tool, it proved to be very difficult to consistently send and receive on the UART bus. We ended up using undocumented interrupts on the Chipkit to operate the UART bus, and faced many difficult bugs, such as a bug where we used the interrupt U2E, which flags errors on the UART2 bus, instead of U2RX, which flags when the RX port receives bytes.