04/03/2016

Mini Project- Autonomous Window Operator

Ziqi Xia (960118-1564) and Oscar Bystedt (960404-6137)

Objective:
Our project is a prototype of a window controller. It accepts inputs from both the on-board temperature sensor, as well as the buttons and switches on the ChipKIT. These inputs are then used to drive a motor which operates the opening and closing of the window. The display gives user feedback to whether the system is operating in manual or automatic mode, the current temperature and the status of the window(motor), as well as a pleasing graphic of a window.

Features of the project:
- The values of the temperature sensor and the buttons/switches are read and processed by a program that we wrote in C.
- Depending on the temperature value, the program decides whether to drive the motor to open and close the window, or not.
- The user can also control the operation of the motor using the buttons on the ChipKIT after enabling manual mode using switches.
- The temperature value and motor status are displayed on the ChipKIT screen.

Solution:
We used the ChipKIT Uno32 board together with the basic I/O shield as well as the on-board temperature sensor, and a servo motor for this project. The intended use for our product is in schools and other buildings to provide a safe automatic way of controlling the temperature and opening the windows in a room. The switches give access to the buttons on the ChipKIT board, which will manually operate the motor as well as disable the automatic controller. This allows for an adult or other responsible person to open the window, but prevents children from accidentally opening it and possibly falling out. The motor is controlled using a PWM signal created by the program. Depending on which buttons are pressed, the angle of the motor is updated in steps. The user can accurately increment the opening and closing of the window by sending a pulse with a different length. One of the switches enables reverse operation by the user. The temperature sensor automatically sends a pulse to the motor to open and close the window depending on how warm it is. The reading from the temperature sensor and the window status are displayed on the screen in real time. All of the programming was written in C and we used the MCB32 tools as the interface between the compiler and the ChipKIT.

Verification:
When testing the project, we made a scale out of paper that replicated how the motor would operate the window. When the temperature was below 20ºC, the window was closed, and therefore the arrow on our scale was pointing to a fully closed window.

When the temperature increased, our scale showed that the window would open more to let more cool air in from outside. We tested it in different environments, and it provided an accurate representation of how far the window would open. The ChipKIT generated heat during operation, so the actual values were not so accurate to the indoor temperature, however the concept of operating a window depending on the temperature still held.

Contribution:
Oscar worked mostly with the user interface and IO, including the graphics on the display and the functionality of the buttons and switches. Ziqi focused on interrupt handling so that the manual override worked smoothly, as well as the larger program syntax and structure. Both members researched the data sheets for all components as well as provided active suggestions to any problems that arose.

Reflection:
Our original project plan implemented an external thermometer to prevent the window from opening if it was too cold outside, as well as the RFID module to prevent it from being opened by someone who was not authorised to do so. Unfortunately, we spent too much time trying to get the external thermometer to work. Since it was controlled using a One-Wire interface which is not natively supported on the ChipKIT, and there is no documentation relevant to our application, we had to abandon that idea with less than a week to the expo. Because of this, our initial objective of authorising using RFID could not be completed, as we did not have enough time to implement the SPI module. Through working on this project, we learned that it is important to check that a device is compatible with an interface before trying to implement the two together. The data sheets were useful to provide specific information on both the ChipKIT as well as our external components. To conclude, we feel like we did not have enough time to implement our ideas and that although we did not create the project we had planned, we still managed to create a working system and learn about microcontrollers in the process.