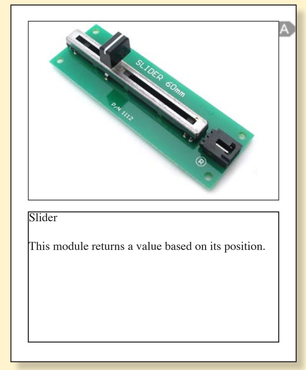
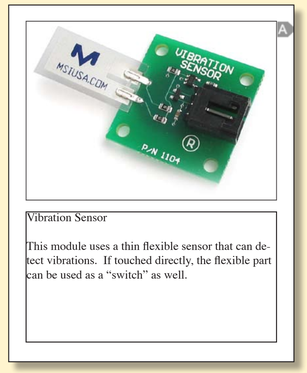
Separating Design and Implementation in Pervasive Computing Education

There are several difficulties that arise when teaching students about all the important aspects of pervasive computing technology. Many of these issues are caused by the tension between the design of a pervasive computing solution and its actual implementation. Students may feel the need to limit their designs based on what they believe they are capable of implementing rather than what is the best overall solution. In this project we developed a method for separating out the design from implementation by providing students with physical cards to represent the various technologies necessary to address a particular problem.



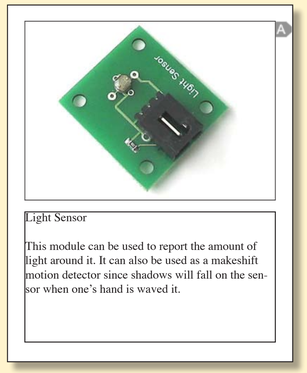
Slider

This module returns a value based on its position.



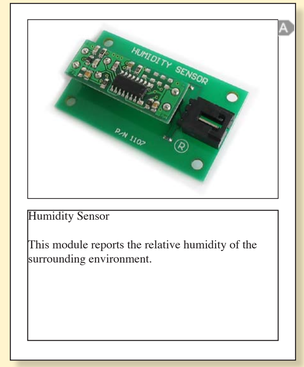
Vibration Sensor

This module uses a thin flexible sensor that can detect vibrations. If touched directly, the flexible part can be used as a "switch" as well.



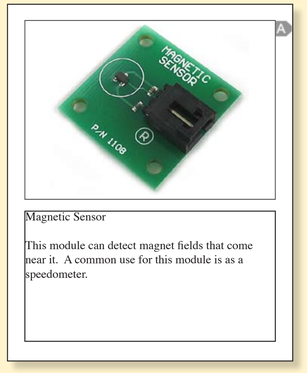
Light Sensor

This module can be used to report the amount of light around it. It can also be used as a makeshift motion detector since shadows will fall on the sensor when one's hand is waved it.



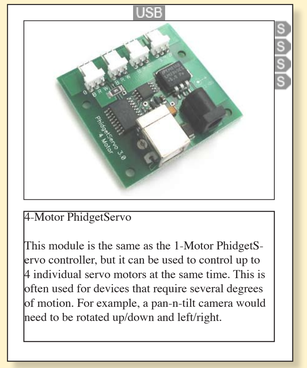
Humidity Sensor

This module reports the relative humidity of the surrounding environment.



Magnetic Sensor

This module can detect magnet fields that come near it. A common use for this module is as a speedometer.



4-Motor PhidgetServo

This module is the same as the 1-Motor PhidgetServo controller, but it can be used to control up to 4 individual servo motors at the same time. This is often used for devices that require several degrees of motion. For example, a pan-n-tilt camera would need to be rotated up/down and left/right.

Students were asked to do the following:

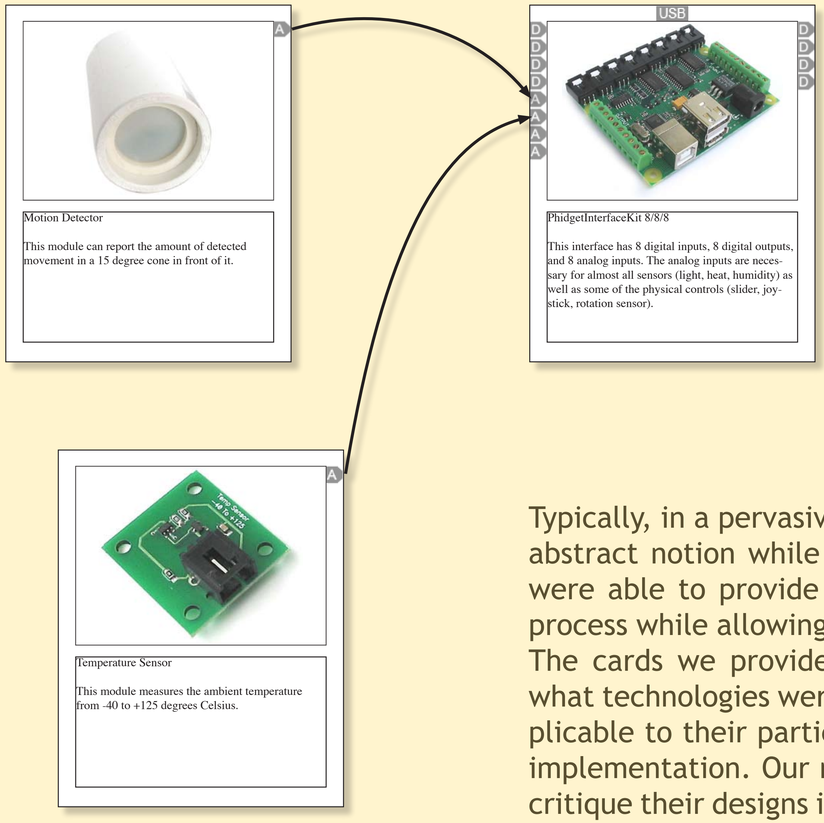
1. Select a specific problem that bothers you in your daily life (e.g. getting settled into bed and realizing that you had forgotten to lock the front door and turn off the television).

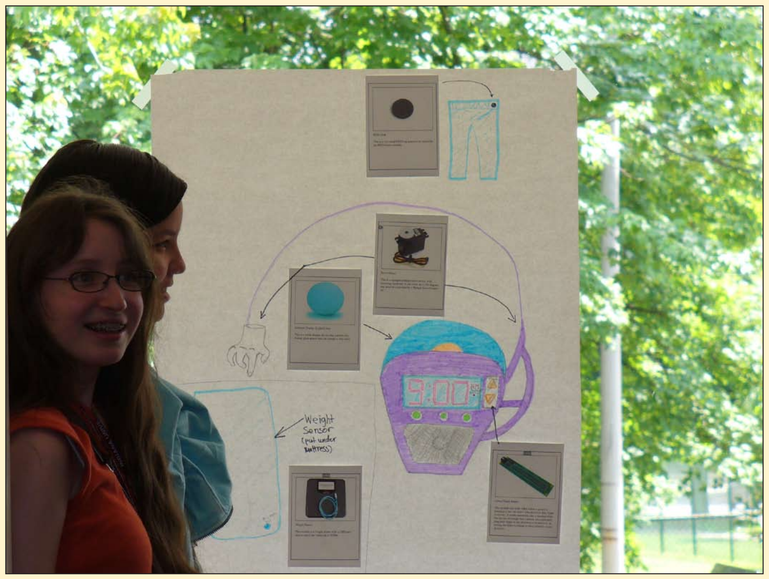
2. Figure out what sorts of things in your environment could be changed to keep this sort of problem (1) from occurring (e.g. make a television that knows whether someone is actually watching).

3. Think about what you would need to know in order to provide the necessary information for (2) (e.g. the television could have built-in motion or heat detection that would tell it when no one is in the room).

Example from the students:

A layout of what might be needed in order to have a television that is aware of whether or not it is being watched using our design cards.





Typically, in a pervasive computing class the design aspect is considered an abstract notion while the implementation is concrete. In our project we were able to provide an environment that solidified the students design process while allowing them to think abstractly about the implementation. The cards we provided gave the inexperienced students information on what technologies were available so they could consider whether it was applicable to their particular problem without being distracted by details of implementation. Our next phase is to develop a way for these students to critique their designs in order to decide what is, or is not, actually plausible for implementation.

William R. Hazlewood and Dennis P. Groth
Indiana University School of Informatics
UbiComp 2006