Fletcher Harrington

Yero Rudzinskas

Design for Chuck the Growbot….

Youtube video: https://youtu.be/s9Rh_62YpA0

Overview:

This project presented some really interesting challenges in implementing both hardware and software. We chose the Arduino platform because we had both been wanting to play with it for a while and due to its popularity, there was a reasonable assumption of solid online support.

The hardware challenges included the basics of circuitry, which neither of us had any experience in whatsoever. But by talking with people and doing a ton of online research we were able to get there. We utilized a breadboard for both prototyping and for our final presentation. We chose the SHT 11 sensor because of its reliability and accuracy. The Powerswitch tail was a cool find, and though it took a few hours for us to figure out how to use it, it is also a great piece of the project. Essentially this is an Ac\Dc converter that also has built in pin inputs so that it may be powered by a micro-controller.

The Arduino environment itself took a little bit of getting used to, but because it uses C++, we were able to use our knowledge of C to implement our logic.

Much more difficult was the implementation of the data collection and visualization in 'Processing'. We used the base code of a project on the Arduino site, but that code accepted only one variable and made a really ugly graph. Ours is, well… much better. It was a significant challenge to utilize the serial port data for both logging and real time visualization but we managed to reach a reasonable implementation of both.

Overall this project offered us a real appreciation of how much complexity can go into even relatively simple micro-controlled systems. Add in the externalities that arise when sensors are used in an changing physical environment and the electrical needs of devices that respond and manipulate that environment and the challenge offered a great learning experience.

A rundown of the hardware environment:

The Arduino received sensor data from SHT11 and sent data to output pins which changed states of fan and AC/DC devices based on logic uploaded to the ATMEGA chip on the microcontroller.

Basic Arduino and circuit design:

The fan:

A transistor was utilized on a breadboard to take in a PWM signal on its base pin from a digital Arduino pin and based on the level of that signal it switched a circuit which controlled power (from 9v battery) to the 12v DC fan. A 40004 diode was used to on the 9volt power (which was modulated by the transistor

to power the fan) and a 100 Ohm resistor was used on the PWM line. Software on the Arduino used sensor data to decide what signal was sent to the fan.

The AC/DC powerswitch:

The Powerswitch Tail II is a mains device which allows control via a wire over a the 120volt signal it is attached to. It takes in DC 5v signal and based on the level either blocks the power, turning off the attached device, or allows it to power on. We controlled this 5v signal via the Arduino pin out and the signal was modulated via the software based on sensor data (humidity). When the humidity was sensed low, the mains circuit was unblocked and the humidifier turned on.

The software:

Our zip file looks like this as far as software goes:

```
main project folder
|
 ----      chucks_brain
|          --contains the basic logic (c++) for the sensing environment and serial data
|          *check out chucks_brain.ino
|
-----      graph
|          --contains the Processing (java) code which implements the visual systems and logging
|          *check out graph.pde
|
|
------     docs
```

Functional details:

The Arduino C++ environment was used to program the controller to create the effects described above. Pins are monitored for sensor data which triggers the devices described above based on thresholds and reports the data over the serial connection (usb). The data is received by the emulated serial connection on the connected PC which is running "Processing v2." The processing environment, programmed in Java, reads the data from the port and creates a realtime graph based on the values. It also creates a log file and writes the data to it.

That's the basic operation of Chuck the Growbox. It isn't quite perfected yet but the major systems are in place. Now we just need to tweak our code, polish our visualization and logging as well as our sensor placement. As it stands, it is a viable automatically controlled, reporting /growing environment for fungus, etc!

Thanks!