PHYSICAL COMPUTING, LAB EXERCISE 2 (Part 1 and 2)

ARDUINO

The purpose of this exercise is for you to familiarize yourselves with microcontrollers and how they can be used to interface the digital and physical worlds, by connecting sensors, actuators and software together. You will also design a small interactive system.

We will use Arduino, a physical computing platform based on a simple I/O microcontroller board and a development environment that can be used to develop stand-alone interactive objects or can be connected to software running on a computer. It implements the Processing/Wiring language, is open-source and has a large community of users that make their work using Arduino available online. Website (and everything you will ever need to know about Arduino: http://www.arduino.cc)

The following is needed (for each group):

- A computer with USB port
- An Arduino microcontroller
- A breadboard (make sure the supply lines on the board are connected all the way!)
- A USB cable
- 1 red LED
- Piezos
- Analogue sensors
- Suitable resistors and possible other components to connect the sensor to the microcontroller, depending on needs
- A switch
- A straw

Getting Started

- Read about Arduino:
  http://www.arduino.cc/en/Guide/Introduction
  http://arduino.cc/en/Main/Hardware
- Get Arduino running, either following the instructions below:
  Windows: http://www.arduino.cc/en/Guide/Windows
  Mac OS: http://www.arduino.cc/en/Guide/MacOSX
  UNIX: http://www.arduino.cc/playground/Learning/Linux
  Or follow these more detailed tutorial instructions:
  http://www.ladyada.net/learn/arduino/lesson0.html
  http://www.ladyada.net/learn/arduino/lesson1.html
- Read more about the software environment
  http://www.arduino.cc/en/Guide/Environment
- Read more about the foundations of working with Arduino:
  http://www.arduino.cc/en/Tutorial/Foundations

For references about the programming language:
http://www.arduino.cc/en/Reference/HomePage
http://www.arduino.cc/playground/uploads/Main/arduino_notebook_v1-1.pdf

Problems? Arduino troubleshooting: http://www.arduino.cc/en/Guide/Troubleshooting

Exercise 1: Analogue and Digital I/O

In this exercise, you will learn how to

- Control digital / analogue values from Arduino
- Read digital / analogue values with Arduino
- Control output values based on sensor values through using Arduino

Digital I/O

1) Turn a LED on and off:

   - Build the circuit described in http://www.arduino.cc/en/Tutorial/Blink
   - Read the following webpage and execute the exercises at the bottom of the page: http://www.ladyada.net/learn/arduino/lesson2.html
   - Browse through http://www.ladyada.net/learn/arduino/lesson3.html to learn more about connecting LEDs to Arduino (do not execute the exercise included in the page, only do a quick read through)

2) Use a push button to control a LED

   - Browse through http://www.ladyada.net/learn/arduino/lesson5.html (do not execute the exercise included in the page, only do a quick read through)
   - Read about digital I/O at http://todbot.com/blog/wp-content/uploads/2006/10/arduino_spooky_projects_class2.pdf, pages 14-17
   - Build the circuit described in http://itp.nyu.edu/physcomp/Labs/DigitalInOut (if there are not enough push buttons, adapt the circuit to other available switches or just use wires to close a circuit. See also http://www.arduino.cc/en/Tutorial/Button for different wiring.)

Analogue I/O

Read about analogue I/O at http://todbot.com/blog/wp-content/uploads/2006/10/arduino_spooky_projects_class2.pdf, pages 28-36.

1) Use an analog output (PWM pin) to fade an LED:

   Build the circuit described in http://www.arduino.cc/en/Tutorial/Fading

2) Use a potentiometer to control the blinking of an LED:

   Build the circuit described in http://www.arduino.cc/en/Tutorial/AnalogInput

3) Read an IR-sensor for sensing proximity

   - Connect a proximity sensor following these instructions: http://www.blueink.com/CLASS/physcom1/SharpGP2D12.htm)
   - Read the sensor values using the same code as in (2).

Exercise 2: Serial Communication

In this exercise, you will start to learn how to control computer programmes with Arduino and control Arduino with a computer, by learning how to set up and use serial communication.

1) Read about serial communication with Arduino at http://todbot.com/blog/wp-content/uploads/2006/10/arduino_spooky_projects_class3.pdf, pages 4-21 and execute the exercises included in these pages.
(For more information about serial communication with Arduino, you can read http://www.ladyada.net/learn/arduino/lesson4.html)

2) Build the following circuits:

- Detect a knock with a piezo and send the info through the serial port: http://arduino.cc/en/Tutorial/Knock
(make sure that the sensor is connected to the same pin as assigned in the code)
- Produce sound using PWM: http://www.arduino.cc/en/Tutorial/PlayMelody
- Playing sounds using the computer's keyboard: http://www.arduino.cc/en/Tutotial/KeyboardSerial

Exercise 3: Interfacing with Software

In this exercise, you will learn how interface hardware with software by connecting Arduino to an interactive graphics software (Processing) and to an interactive music programming environment (Pd). A complete list of software working with Arduino and of corresponding tutorials is available at http://216.38.50.214/playground/Main/InterfacingWithSoftware.

Processing

Read about Processing at http://todbot.com/blog/wp-content/uploads/2006/10/arduino_spooky_projects_class4.pdf pages 24-31 and http://itp.nyu.edu/physcomp/Labs/SerialOut without doing the exercises.

1) Install Processing:

Dowload Processing from http://processing.org/. Follow these instructions: http://www.arduino.cc/playground/Interfacing/Processing - stop right before "references". For more information about Processing, you can go to http://processing.org/
(If you receive an error message about "framerate", change the word with FrameRate" in the code.)

2) Display sensor behaviour with Processing

Execute the following exercises in order to see have a closer look at the behaviour of various sensors by visualizing it with Processing:

- Switch: http://webzone.k3.mah.se/projects/arduino-workshop/projects/arduino_meets_processing/instructions/switch.html
- Potentiometer: http://webzone.k3.mah.se/projects/arduino-workshop/projects/arduino_meets_processing/instructions/poti.html
- LDR light sensor: http://webzone.k3.mah.se/projects/arduino-workshop/projects/arduino_meets_processing/instructions/ldr.html
- Do same as potentiometer only using other analogue sensors such as IR sensor
- Piezo as knock-sensor: http://webzone.k3.mah.se/projects/arduino-workshop/projects/arduino_meets_processing/instructions/piezo.html

3) Control the brightness of an LED by moving a mouse while watching the same movement in Processing: http://www.arduino.cc/en/Tutorial/Dimmer

Pd (Pure Data)

Read about Pd on this front page: http://puredata.info/

1) Install Pd-extended

Download the Pd version that corresponds to your computer model and follow the corresponding instructions: http://puredata.info/downloads

2) Install Arduino2PD, a Pd programme that reads sensor values from Arduino, and Simple Message System, a library for using Arduino with Pd

- Read about Arduino2PD and Simple Message System at http://arduino.cc/playground/Interfacing/PD
- Download Arduino2PD from the link below: http://www.arduino.cc/playground/uploads/Interfacing/Arduino2PD.zip
- Follow the instructions in the Arduino2PD readme file included in the zip-folder, only use this page for downloading Simple Message System instead (Likewise, follow the instructions in the readme file to complete the SMS installation): http://www.arduino.cc/playground/Code/SimpleMessageSystem
(Make sure to put SMS in the right folder!)
- Use the Arduino_for_PD_input.pde code included in the Arduino2PD zip-folder to programme Arduino
- Run Arduino2PD, making sure the right port and the right baud rate are used (same baud rate as Arduino). Ports might vary depending on computer model.
- See how sensor values affect values on the Pd window!

3) Try out making sounds with Pd, using Arduino

Pd has a lot of patches available in its documentation. You can easily control them with sensors by using Arduino.

- Look for a patch that behaves in an interesting way, and let Arduino control this patch by copy-and-pasting the Arduino2PD patch into the window and connecting the output signal to the signal input of an object in the patch.
- Connect a sensor to Arduino using the Arduino_for_PD_input.pde code and watch how the sensor can control sound parameters.

Exercise 4: Design of an Interactive System

Grading: If you have done everything up until now, you will get 3. To get 4, also do part (1), (2) and (4) OR (1), (3) and (4) of this exercise. To get a 5, do everything.

1) Using what you have learned so far during the lab exercise, imagine an interactive system that interfaces the digital world to the physical one, either by sensing it or actuating it. This should be something *simple* and feasible for you to build an implementation prototype of. Write an algorithm for how this system would work

Depending on how much time you have left before the end of the lab, prototype this system doing a combination of the following points (one or both)

2) Make an a look & feel prototype and a role prototype (scenarios, storyboards…).
3) Build the actual circuit.

4) Present the final result using all the prototypes you have built.