MTAT.03.262 Mobile Application Development

Pervasive Mobile Applications
Micromechanical technologies

Huber Flores
huber@ut.ee

Tartu, Estonia, 2013
Outline

• Background
• Micromechanical technologies
  – Embedded
  – Distributed (e.g. microcontrollers)
• Pervasive development
  – Android + underlying hardware
  – Android + Arduino
• Demos
BACKGROUND
Pervasive Mobile Applications

• Ubiquitous sensor data
• Wireless communication
• Wearable devices
• Energy consumption
• Participatory, opportunistic and social sensing
MICROMECHANICAL TECHNOLOGIES
Mobile Sensing

- Embedded sensors
Mobile Sensing

• Accelerometer

Sensor pattern (e.g. walking)
Mobile Sensing

• Gyroscope

http://www.ovation.co.uk/video-stabilization.html
Mobile Sensing

• Magnetic field
Mobile Sensing

• Proximity
• Air pressure
Mobile Sensing

• Distributed (e.g microcontrollers)

http://makezine.com/2012/03/02/processing-for-android-and-arduino-tutorial-and-samples-from-tellart/
Mobile Sensing

- Microcontroller

Arduino Mega ADK
Mobile Sensing

• Shields

Wireless shield

Sensor shield
Mobile Sensing

• Modules

Usb to serial converter

Wireless module
Mobile Sensing

• Sensors
PERVASIVE DEVELOPMENT
Mobile Sensing in Android

• Physical device
• Emulator?
Mobile Sensing in Android

• Sensor simulator tool
  – Download and extract
    • http://code.google.com/p/openintents/wiki/SensorSimulator (SensorSimulator 2.0-rc1.zip)
    • Install app in Android emulator
      – $ adb install SensorSimulatorSettings-2.0-rc1.apk
    • Execute sensor simulator
      – $ java –jar sensorsimulator-2.0-rc1.jar
      – Set IP address in Android emulator
  – Alternatively, this repository contains everything you need
    (https://github.com/huberflores/AndroidSensorSimulator.git)
Mobile Sensing in Android
Mobile Sensing in Android

Sensor Simulator

Choose Device

Basic Orientation

Extended Orientation

Write emulator command port and click on set to create connection. Possible IP addresses:

10.0.2.2
192.168.1.79
Mobile Sensing in Android

Huber Flores
Mobile Sensing in Android

```java
public void onAccelerationChanged(float x, float y, float z)
public static void startListening

// wrapper class
private static Sensor sensor;
private static SensorManager sensorManager;

public static boolean isSupported()
private static AccelerometerListener listener;
```
Exercise

• Try and install sensor simulator
Android + Arduino
Android + Arduino

• Accessory Mode
  – Usb cable
Android + Arduino

• Development requirements
  – Arduino IDE
    • Sketch (C, C++)
  – Hterm console
    • Communication testing
const int ledPin1 = 13;  // pin that LED 1 is attached to
const int ledPin2 = 12;  // pin that LED 2 is attached to
int statePin1 = 0;        // state of pin 1
int statePin2 = 0;        // state of pin 2
int data = 0;             // which LED to turn ON/OFF

void setup() {
  Serial.begin(115200);
  pinMode(ledPin1, OUTPUT);
  pinMode(ledPin2, OUTPUT);
}

void loop() {
  // if there is any available data
  if(Serial.available() > 0) {
    // read it and store it to
    data = Serial.read();
    // received data is going to be either 1 or 2 (it's the identifier)
    // we are going to use on our Android app - if LED 1 button is pressed,
    // it sends out value 1 and if LED 2 button is pressed, it sends out value 2
    if(data==1) {
      // state 0 means it's off
      if(statePin1==0) {
        // turn on LED and set state to 1
        digitalWrite(ledPin1, HIGH);
        statePin1=1;
        // state 1 means it's on
      } else if(statePin1==1) {
        digitalWrite(ledPin1, LOW);
        statePin1=0;
      }
    } else if(data==2) {
      if(statePin2==0) {
        digitalWrite(ledPin2, HIGH);
        statePin2=1;
      } else if(statePin2==1) {
        digitalWrite(ledPin2, LOW);
        statePin2=0;
      }
    }
  }
}
Android + Arduino

Bind to an Android view

Establish communication
Android + Arduino

```java
/**
 * Write broadcast packet.
 */

public void write(byte[] buffer) {
    try {
        String data = new String(buffer);
        DatagramPacket packet = new DatagramPacket(data.getBytes(), data.length(), myBcastIP, BCAST_PORT);
        mSocket.send(packet);
    } catch (Exception e) {
        Log.e(TAG, "Exception during write", e);
    }
}
```

```xml
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="stim.puniste.com"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk android:minSdkVersion="10" />
    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
    <uses-permission android:name="android.permission INTERNET" />
    <application...
        android:label="@string/app_name">
            ...
        </activity>
</manifest>
```

Send streaming

Set permissions in Manifest.xml
DEMO
QUESTIONS
References

• Suggested readings
  – Arduino tutorial
  – Source code
    https://github.com/huberflores?tab=repositories
Home Assignment 2

• We already explored the utilization of micromechanical sensors within the mobile applications
  – We will focus in the accelerometer
• You can use sensor emulator or a physical device for this homework
  – If you don’t have a device, we can provide you one, but request it with anticipation
• Sensor emulator demonstration was provided along with some code for sensor simulator
• Sensor working code for a real device was also provided
  – It can be utilized as core if you find it usable, but you can always code yours
Home Assignment 2

• The home assignment consists to create an image editor application with at least 3 features (e.g. cut, draw, etc) that implements redo and undo functions using the accelerometer

• The idea is to set two different thresholds (one for redo and one for undo) and to apply those functions to the images once the device detects the specified movement
Home Assignment 2

• Delivery date:
  – Two weeks after this lecture.
Last year app...