

# **iPhone/iPod Touch as a Data Acquisition and Control Device**

**By Lindrick Outerbridge &  
Jared Alan Frank**

# Introducing the iPhone

- Marketed by Apple Inc.
- Internet-able mp3 player, camera, and smartphone
- Less than 140g (5oz)
- Over 20 million sold worldwide
- Offers software development kit (SDK) for developers to write custom applications
- Fast & reliable performance
- user-friendly graphical



# Features and Technology

- 1<sup>st</sup> Generation:quad-band GSM with EDGE
- 2<sup>nd</sup> Generation added UMTS with HSDPA  
(3G mobile technologies and protocols)
- Multi-touch screen
- Internal 3-axis accelerometer
- GPS
- Camera

# Features and Technology

- *Original & 3G:* Samsung 32-bit RISC ARM1176JZ(F)-S v1.0  
620 MHz underclocked to 412 MHz  
PowerVR MBX Lite 3D GPU

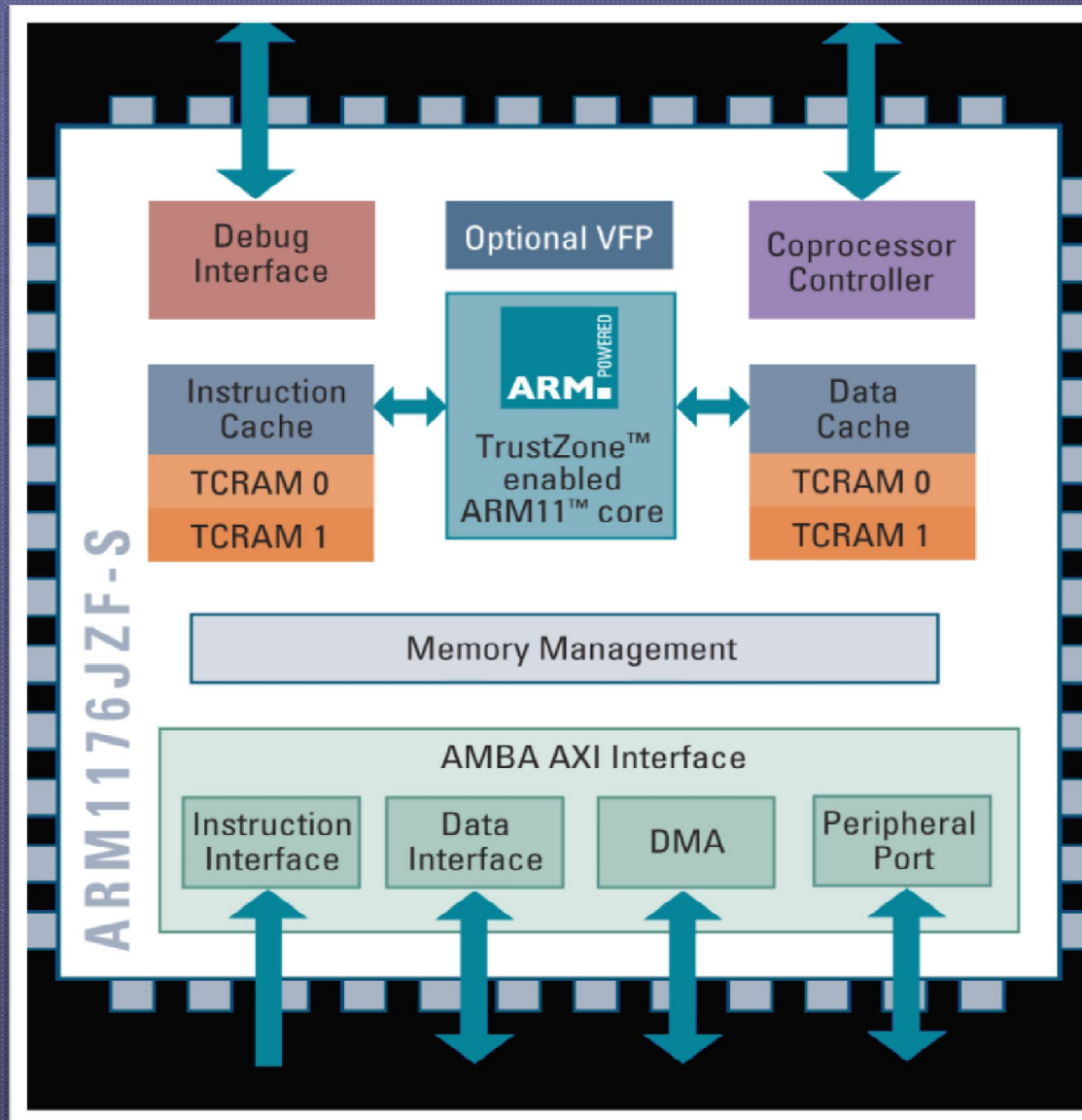
*3GS:* ARM Cortex-A8  
833 MHz underclocked to 600 MHz  
PowerVR SGX GPU

- “Wi-Fi (802.11b/g), Bluetooth 2.0+EDR (3GS: 2.1),  
USB 2.0/Dock connector  
Quad band GSM 850 900 1800 1900 GPRS/EDGE  
3G : A-GPS; Tri band UMTS/HSDPA 850, 1900,  
2100

# Features and Technology

- headset controls
- proximity and ambient light sensors  
3GS: digital compass
- *Original & 3G: 128 MB DRAM*  
3GS: 256 MB
- Flash memory (*Original: 4, 8, & 16 GB;*  
*3G: 8 & 16 GB; 3GS: 16 & 32 GB*)

# Internal Architecture



# Object-Oriented Programming

- iPhone is programmed in Objective-C language

- Objective-C is superset of C:

**Objective-C = C + Object-Oriented capabilities**

- In object-oriented programming:

- Data (variables) and Operations on Data (functions)

become

Instance variables and Methods

# Object-Oriented Programming

- Classes are defined that objects become instances of
- Subclasses inherit properties (instance variables) and behaviors (methods) of the Superclass

(Ex. Table and chair are subclasses of furniture class. A round table and a wheeled chair are instances of the classes with some inherited properties but with some customized properties)



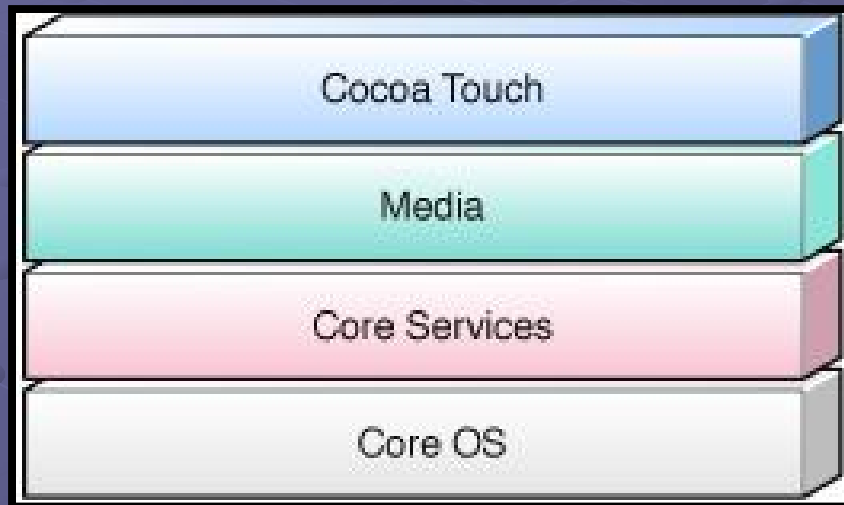
# Object-Oriented Programming

- Frameworks contain broadly-defined classes to perform particular goals (Ex. Audiotoolbox framework, externalaccessory framework, coregraphics framework)
- Objects are declared as instances of an already defined (in terms of variables and methods) class
- Objects perform the methods defined by its class by sending messages to one another

**Message Syntax:** `[receiver methodname]`

**Ex:** `[myBox open]`

# iPhone OS Technology



Each layer of the iPhone OS stack contains a list of associated frameworks that are responsible for the features and functionality of a specific technology of the phone and can be accessed through the methods of its classes

# iPhone Programming

## Xcode Tools :

Xcode – IDE; manage, edit, compile, run, and debug projects. Integrates with other tools; the main application you use during development

Interface Builder - assemble your user interface visually. The interface objects created are saved to a special resource file format & loaded into the application at runtime

Instruments - runtime performance analysis and debugging tool. To gather behavior/performance info and identify potential problems

iPhone Simulator - simulates the iPhone technology stack to test iPhone applications on an Intel-based Mac

# Mac Mini

- Cannot develop for the iPhone/iPod touch without an apple computer
- Mac mini is an affordable Intel-based Macintosh desktop
- DVI to VGA adapter needed for VGA-supported PC monitors
- 5 USB ports
- Wifi/Ethernet, Bluetooth
- Mac OS X v10.5.6 and up



# What's been done

- Bluetooth and wired USB (dock connector) data transfer protocols are off limits to iPhone developers. The only inter-device communication protocol available requires WiFi connectivity. Aside from being the only hardware option for the iPhone (at the moment), OBD-II WiFi offers some technical advantages over Bluetooth and USB:
  - Bluetooth headsets would be unusable during Rev operation.
  - Bluetooth hardware may never work with the iPod Touch (it has a very limited subset of protocol support).
  - A USB dock connection to an OBD-II device would have to replace any car-charger/music dock connectors.

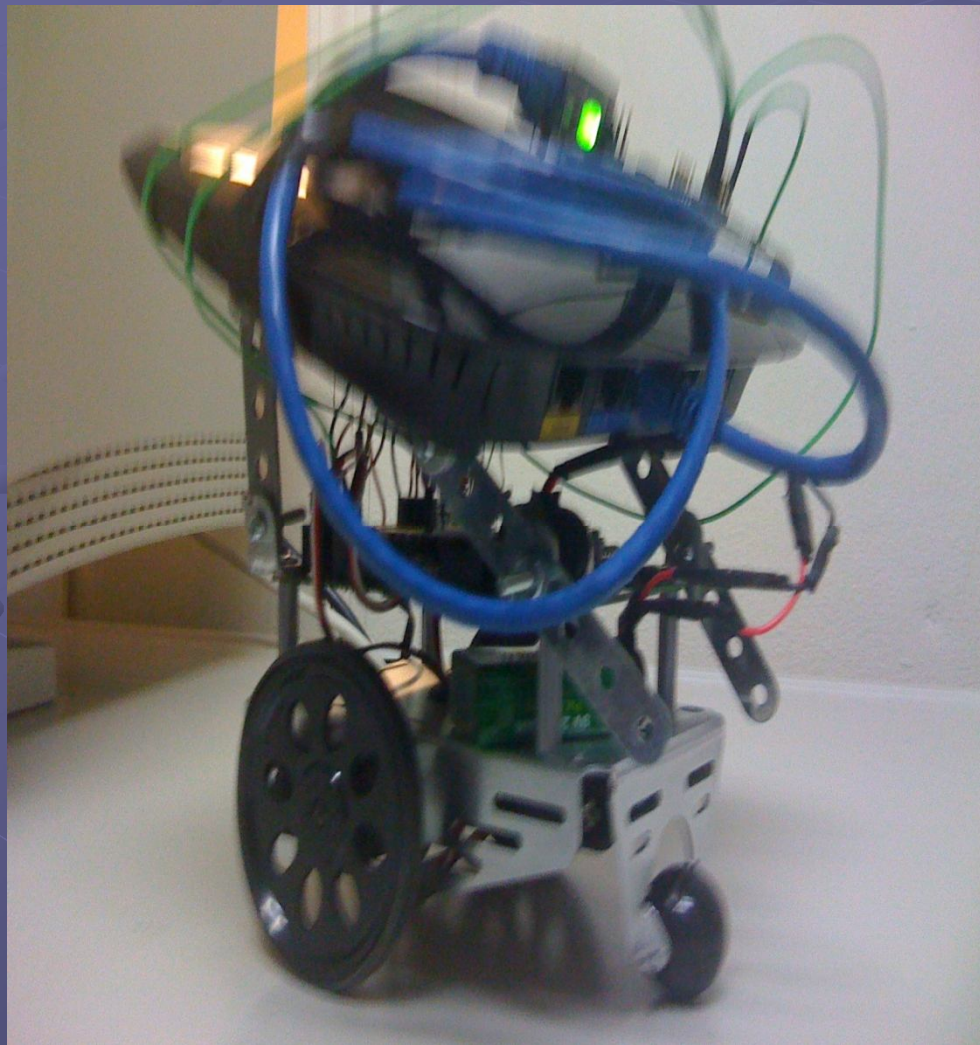
# What's been done

- Military
- Medical
- R2D2 robot
- RC car
- Robot plane squad
- Packbot

# Our Long-term Goals

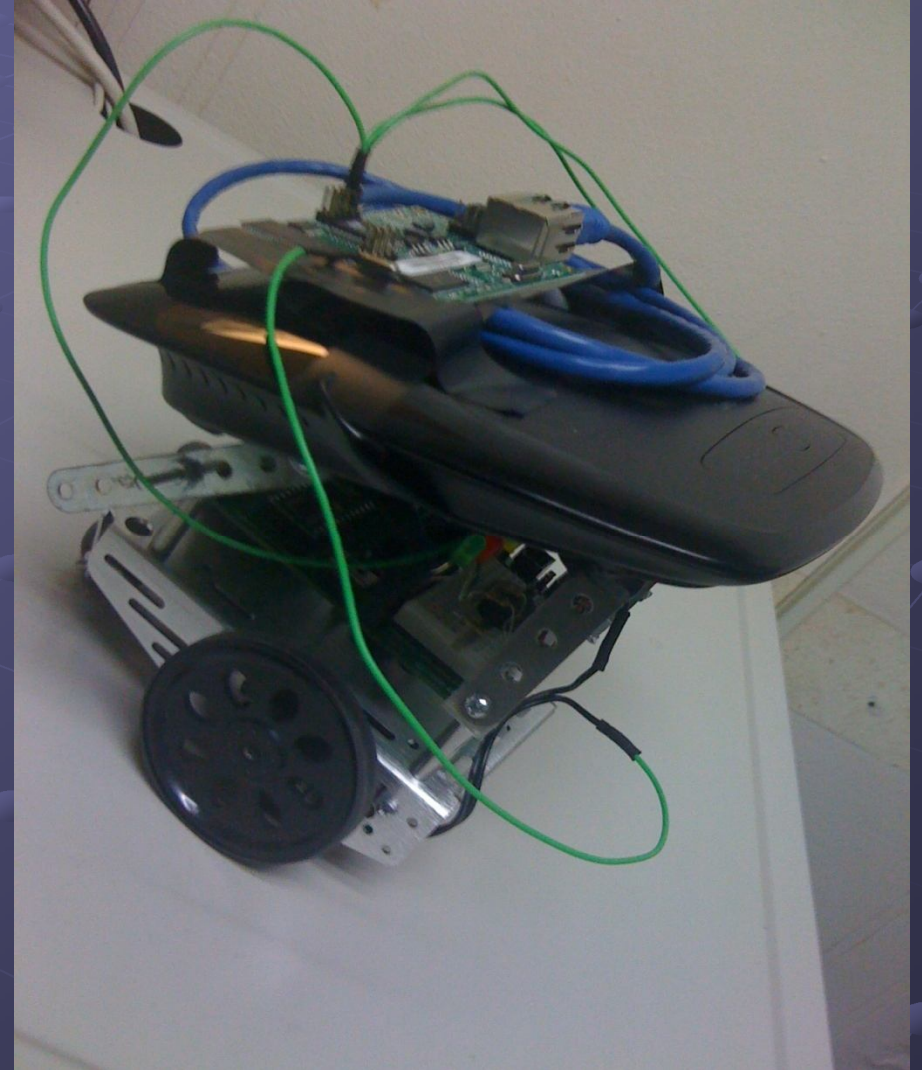
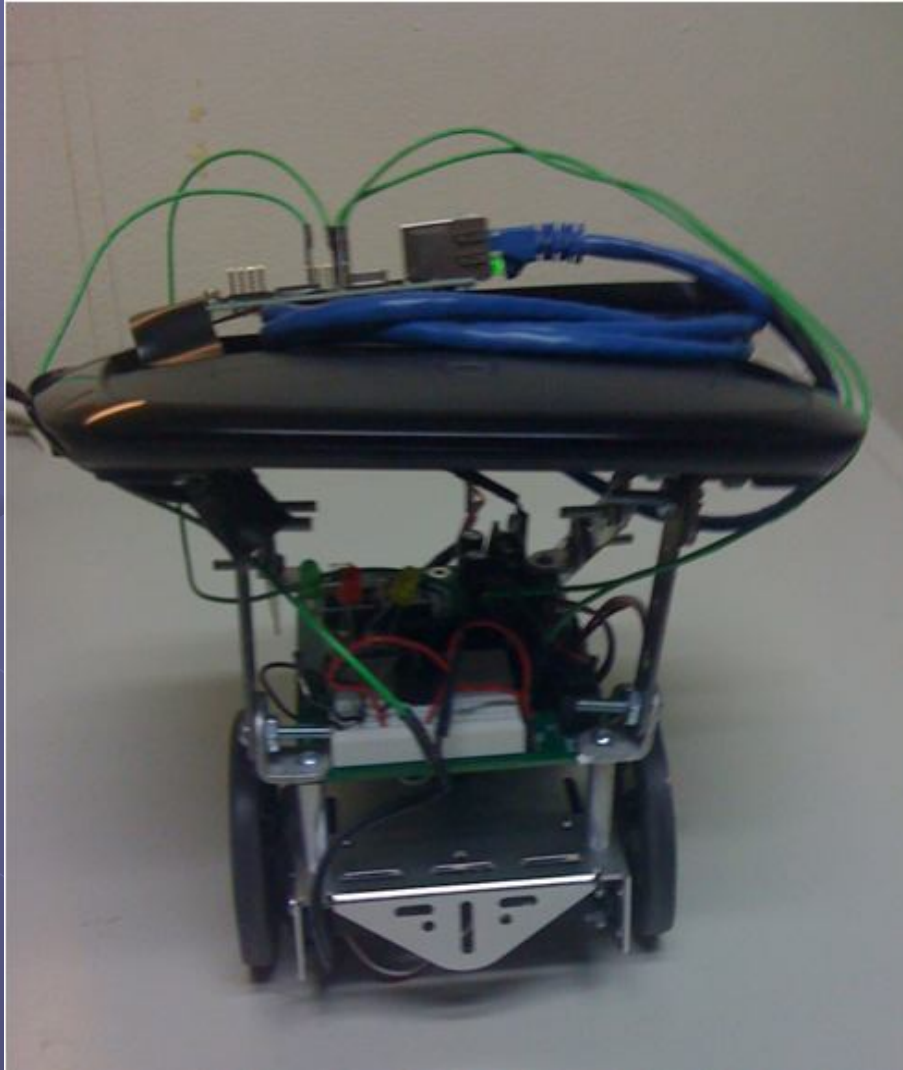
- A remote CPU for mechatronics applications
- Data Acquisition and Control Device
- Experiment/Equipment communication
- BASIC Stamp interfacing
- Wireless sensor networks

# Mobile Robot with Wireless Router





# Mobile Robot



# Motor Control Desktop Icon



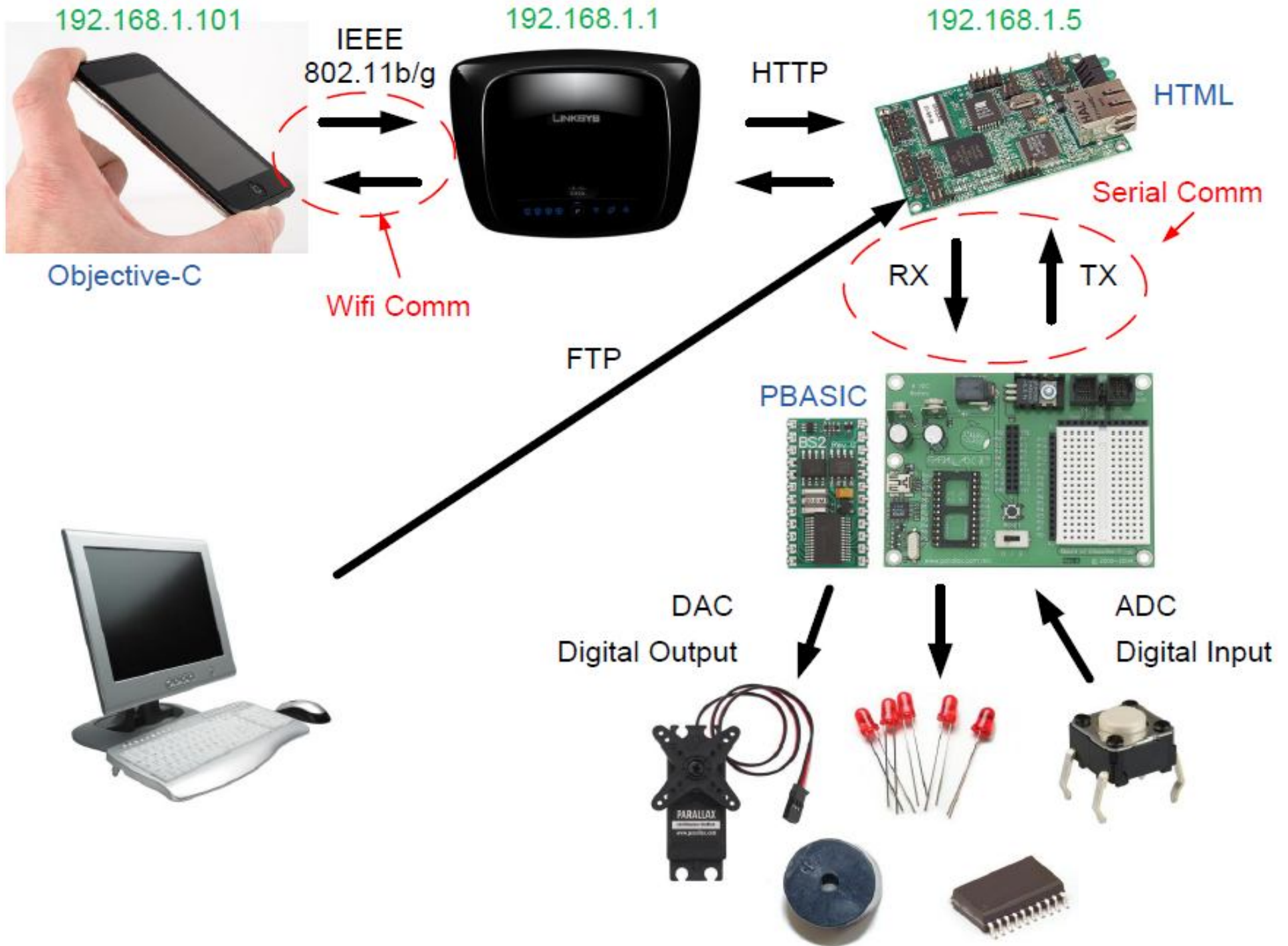
# Failure of bluetooth

- Bluetooth failed
- Jail breaking / Downgrading and the search for 3<sup>rd</sup> party apps and hacked Bluetooth
- SPP, GAP, SDAP, GOEP – the four most basic, low-level Bluetooth profiles
- SPP especially essential – sets up a virtual serial connection – most important data transfer profile. ALL SPP supported devices always communicate with each other
- FTP attempt (input and output streams [obj-c objects] with CFNetwork.framework (CFFTP classes))

# Supported Bluetooth

Device	Hands-Free Profile (HFP 1.5)	Phone Book Access Profile (PBAP)	Advanced Audio Distribution Profile (A2DP)	Audio/Video Remote Control Profile (AVRCP) <sup>*</sup>	Personal Area Network Profile (PAN)
iPhone 3GS	√	√	√	√	√
iPhone 3G	√	√	√	√	√
Original iPhone	√	√	-	-	-
iPod touch (2nd generation)	-	-	√	√	√

<sup>\*</sup> iPhone 3GS, iPhone 3G, and iPod touch (2nd generation) support pause, play, and stop for AVRCP.

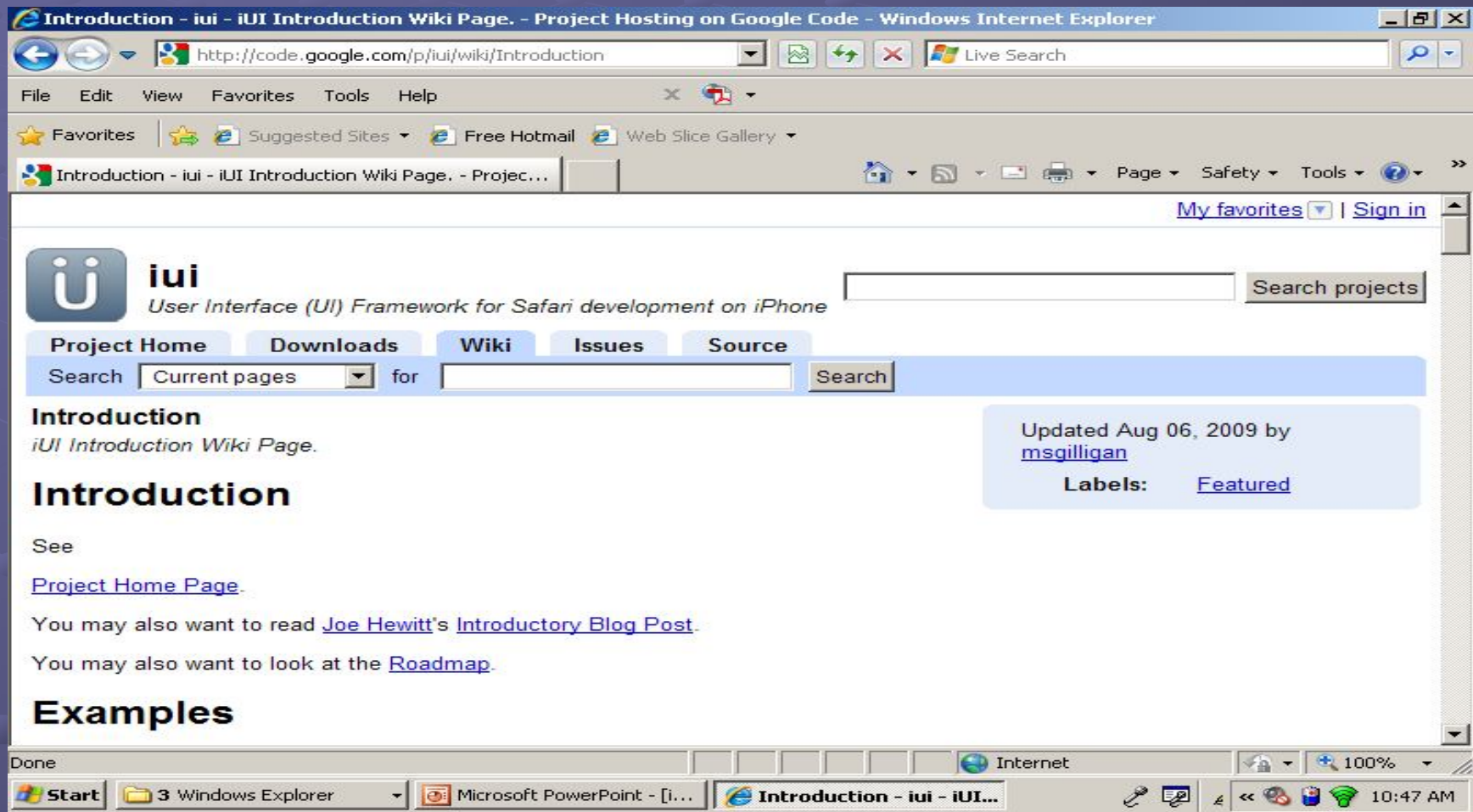


# Implementation

- Create an HTML web page using iUI Interface
- Open FTP Connection to Embedded WebServer. Upload HTML File.



# iUi User Interface Framework for Safari development on iPhone



# Writing to Variables

- Writing to variables via web page can be accomplished with HTML POST Method

```
<form name="iDAC" method="post"
```

```
<h2>Speaker</h2>
```

```
<fieldset>
```

```
<div class="row">
```

```
<input value="1" name="Nb_var03"  
type="radio">On
```

```
<input value="0" name="Nb_var03"  
type="radio">Off
```

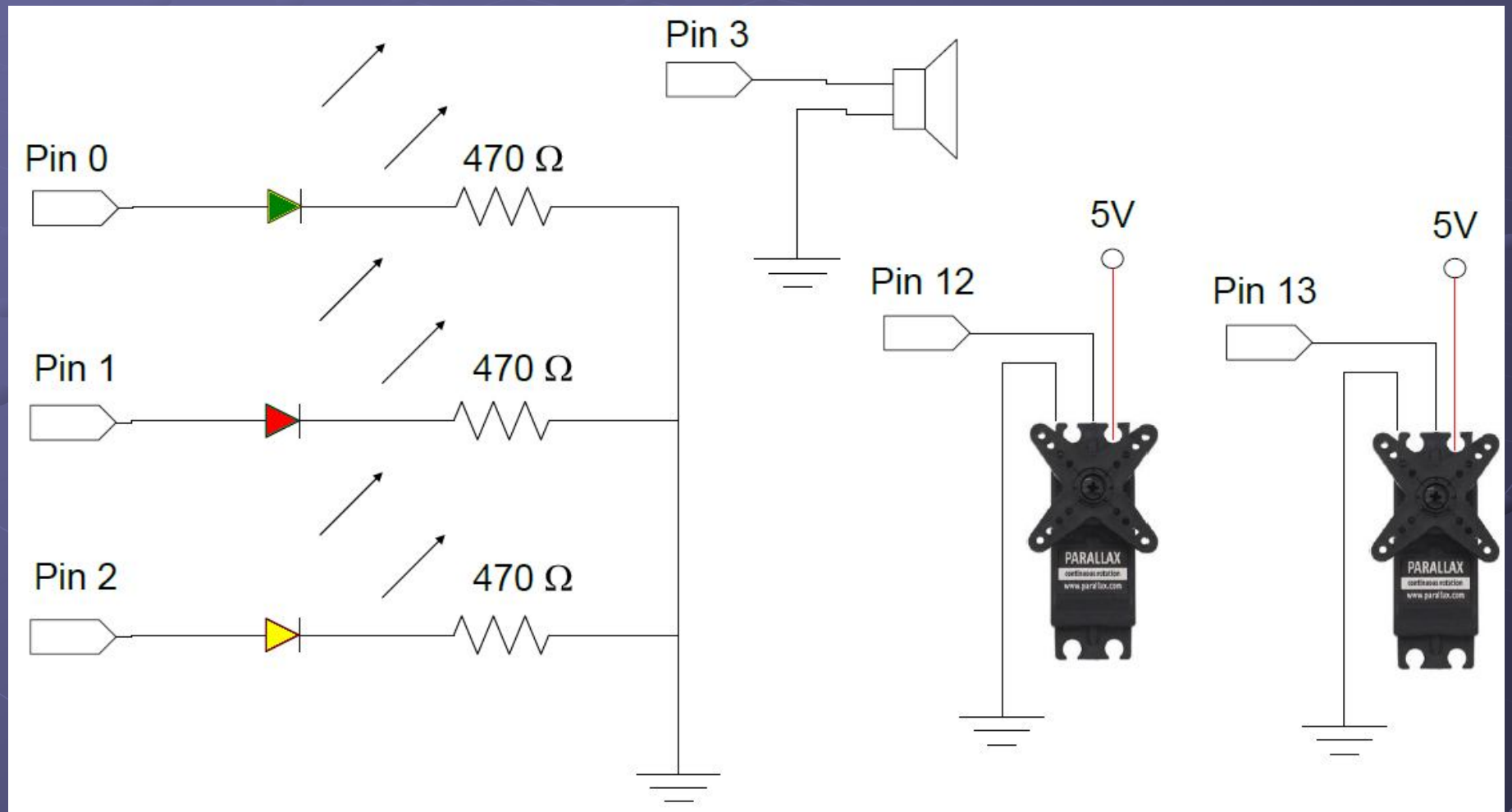


# Access a variable serially from an HTML page

Read the variable in the format variable !NB0R05

- NEXT
- 'SEROUT TX, Baud, ["!NB0R05"] ' Send Command To Read variable in Flash memory
- 'SERIN RX, Baud, 100, Timeout, [DEC nbvar]
- Baud Rate is 2400 bps

# Breadboard Circuit



# Problems

## ● Channel

- 802.11g transmits in 2.4Ghz range. The channel had to be configured to a working channel (5)

## ● Timeout

- The servo motors drained power from our circuit causing the PINK to timeout

## ● Power

- Separate power supplies for router and motors

## ● FTP 0 Bytes There is a file limit for FTP app

# Future

- OSC using UDP/IP & Ethernet

