Physical Computing
What We’ve Learned So Far @ITP

Tom Igoe
ITP
Tisch School of the Arts
NYU
ITP

• 2-year grad program, 230 students

• No technical requirements to admissions:
  • Artists, designers, musicians, dancers, chefs, therapists, sword swallowers, sandwich artists, engineers, butchers, bakers, candlestick makers, etc.

• Collaboration is central to the program

• A plurality of voices and a large community makes for better interactive design work
Physical Computing: methods for facilitating a greater range of human physical expression using computers.
Sensor-to-Screen

Foosball
Dennis Crowley
Ramps

John Schimmel, Tristan Perich, Wlodek Koss
Standalone Objects

SmartHug
Heather Dewey-Hagborg
Vaishali Khandare
Therapy Incentives Through Play
Stephen Kerrigan, Mike Bukhin
Wireless Objects

Brett Shulz, Daniel Perlin, Amos Bloomberg

Needies
A few microcontroller modules for physical computing

- BX-24
- Basic Stamp 2
- Basic Atom Pro24
- PIC 18F452
Hardware is Hard

- Keep setup cost really low
- Consider the cost of multiple controllers
- The fewer components, the better
- The things that seem most minor to experienced people are the most major stumbling blocks for beginners
- Abstract the housekeeping, focus on the interface
Basic Stamp 2
<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>10K Resistor</td>
<td>$0.05</td>
</tr>
<tr>
<td>7805 Voltage Regulator</td>
<td>$0.50</td>
</tr>
<tr>
<td>DC Power Jack</td>
<td>$1.79</td>
</tr>
<tr>
<td>Jumper Wires</td>
<td>$10.99</td>
</tr>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Breadboard</td>
<td>$5.95</td>
</tr>
<tr>
<td>BS-2</td>
<td>$49.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$68.45</strong></td>
</tr>
<tr>
<td>Item</td>
<td>Price</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>10K Resistor</td>
<td>$0.05</td>
</tr>
<tr>
<td>7805 Voltage Regulator</td>
<td>$0.50</td>
</tr>
<tr>
<td>DC Power Jack</td>
<td>$1.79</td>
</tr>
<tr>
<td>Jumper Wires</td>
<td>$10.99</td>
</tr>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Breadboard</td>
<td>$5.95</td>
</tr>
<tr>
<td>BX-24</td>
<td>$49.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$68.45</strong></td>
</tr>
<tr>
<td>Component</td>
<td>Price</td>
</tr>
<tr>
<td>---------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>10K Resistor</td>
<td>$0.05</td>
</tr>
<tr>
<td>7805 Voltage Regulator</td>
<td>$0.50</td>
</tr>
<tr>
<td>DC Power Jack</td>
<td>$1.79</td>
</tr>
<tr>
<td>Jumper Wires</td>
<td>$10.99</td>
</tr>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Breadboard</td>
<td>$5.95</td>
</tr>
<tr>
<td>4MHz Crystal</td>
<td>$2.25</td>
</tr>
<tr>
<td>2 - 22pF Capacitors</td>
<td>$0.07</td>
</tr>
<tr>
<td>PIC 18F252</td>
<td>$6.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$27.84</strong></td>
</tr>
<tr>
<td>Item</td>
<td>Price</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>10K Resistor</td>
<td>$0.05</td>
</tr>
<tr>
<td>7805 Voltage Regulator</td>
<td>$0.50</td>
</tr>
<tr>
<td>DC Power Jack</td>
<td>$1.79</td>
</tr>
<tr>
<td>Jumper Wires</td>
<td>$10.99</td>
</tr>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Breadboard</td>
<td>$5.95</td>
</tr>
<tr>
<td>4MHz Crystal</td>
<td>$2.25</td>
</tr>
<tr>
<td>2 - 22pF Capacitors</td>
<td>$0.07</td>
</tr>
<tr>
<td>PIC 18F252</td>
<td>$6.00</td>
</tr>
<tr>
<td>PicBasic Pro Compiler</td>
<td>$249.00</td>
</tr>
<tr>
<td>MElabs Programmer</td>
<td>$79.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$355.84</strong></td>
</tr>
</tbody>
</table>
Arduino
LED $0.17
Arduino Board $30.00
Total $30.17
Software Must Be Soft
(Because Hardware is Hard)

• Simple interface is important
• Simple language is important
• Big function/feature list isn’t so important
• Minor language elements are major stumbling blocks
• Abstract the housekeeping, focus on the interface
' {$STAMP BS2}
' {$PBASIC 2.5}

main:
  HIGH 0
  PAUSE 1000
  LOW 0
  PAUSE 1000
GOTO main
main:
   HIGH 0
   PAUSE 1000
   LOW 0
   PAUSE 1000
   GOTO main
Option Explicit

Public Sub Main()

End Sub
DEFINE OSC 20

OUTPUT portc.4

main:
    HIGH portc.4
    PAUSE 500
    LOW portc.4
    PAUSE 500
    GOTO main
```c
int ledPin = 13; // LED connected to digital pin 13

void setup()
{
    pinMode(ledPin, OUTPUT); // sets the digital pin as output
}

void loop()
{
    digitalWrite(ledPin, HIGH); // sets the LED on
    delay(1000); // waits for a second
    digitalWrite(ledPin, LOW); // sets the LED off
    delay(1000); // waits for a second
}
```
void setup() {
    pinMode(2, OUTPUT);
}

void loop() {
    digitalWrite(2, HIGH);
    delay(1000);
    digitalWrite(2, LOW);
    delay(1000);
}
void setup() {
    pinMode(2, OUTPUT);
}

void loop() {
    digitalWrite(2, HIGH);
    delay(1000);
    digitalWrite(2, LOW);
    delay(1000);
}

millis() !!!!!!
Phidgets

• Love that there’s no need to program hardware
Phidgets

• Love that there’s no need to program hardware

• Standalone?
LED $0.17
Make Controller $149.00
Total $149.17
<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Make Controller</td>
<td>$149.00</td>
</tr>
<tr>
<td>Rowley Crossworks</td>
<td>$1000.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$1,149.17</strong></td>
</tr>
</tbody>
</table>
Welcome to CrossStudio for ARM

CrossStudio for ARM is a streamlined integrated development environment (IDE) for building, testing, and deploying ARM applications.

CrossStudio provides:

- **Source Code Editor** - A powerful source code editor with multi-level undo and redo, makes editing your code a breeze.
- **Project System** - A complete project system organises your source code and build rules.
- **Build System** - With a single key press you can build all your applications in a solution, ready for them to be loaded onto a developer card or into the debugger.
- **JTAG Interface Support** - You can download and debug your applications using the Rowley USB CrossConnect for ARM, Macraigor Wiggler or Segger JLINK.
- **ARM Simulator** - The simulator mimics the behaviour of the ARM CPU, so you can test your applications without hardware.
- **Integrated Debugger** - The debugger will help you to quickly find problems in your applications.
- **ARM Flash Programming and Debug** - You can download your programs directly into Flash and debug them seamlessly from within the IDE.
- **Integrated Help system** - The built-in help system provides context-sensitive help and a complete reference to the CrossStudio IDE and tools.

**Getting Started**

The [CrossStudio Tutorial](http://www.crossstudio.org) section gives an overview of how to get started with CrossWorks for ARM.

**Did you know...** Ctrl+Tab cycles through the open editors. Ctrl+Shift+Tab cycles through them backwards.
Parallax Propeller

<table>
<thead>
<tr>
<th>Item</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>LED</td>
<td>$0.17</td>
</tr>
<tr>
<td>Propeller Dev. Kit</td>
<td>$129.00</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>$129.17</strong></td>
</tr>
</tbody>
</table>
Parallax Propeller

Premise  Propeller

---

`/*

* (C) 2006 Parallax, Inc.

* Author: Chris Savage & Jeff Martin

* Started: 05-08-2006

*/

***************************************

'Interface to Ping))) sensor and measure its ultrasonic travel time. Measureme'

Each method requires one parameter, Pin, that is the I/O pin that is connected

![Diagram]

Connection To Propeller

Remember PING))) Requires

+5V Power Supply

CON

TO_IN = 73_746
TO_CM = 29_034

PUB Ticks(Pin) : Microseconds | cnt1, cnt2

'Return Ping)))'s one-way ultrasonic travel time in microseconds

outa[Pin]-
dira[Pin]--
outa[Pin]--
The Future
Thanks!

More info:
http://itp.nyu.edu
http://tigoe.net/pcomp/