**Details**

The hardware revolution is just beginning. Nest, Ring, Mellow; these are just the tip of the iceberg. Advances in hardware and software have opened a new world of connected possibilities that extends beyond just the consumer and into the industrial and commercial IoT world. And if you can develop apps, you can build hardware!  
  
Howdy folks, Bryan Costanich here, and I'm super excited to announce the very first Hardware Hackers Seattle meetup. Come and get inspired to build your own connected things as I walk through hacking household appliances into connected things that you can control with your phone using Netduino and Xamarin. Come learn how Netudino.Foundation ([http://Netduino.Foundation](http://netduino.foundation/)) makes creating connected things in C# a piece of cake, with a massive library of peripheral drivers for sensors, LCDs, etc., and a handrolled API that makes building smart hardware a plug and play experience.  
  
Many thanks to City University (<https://www.cityu.edu/>) for sponsoring our space!

Xamarin Forms – easily deploy i0S

*Easier to learn Xamarin / Xamarin Forms?*

*Answer: Xamarin Forms is easier. The control kits are a lot nicer.*

*Android is like a web, every view is a page.*

*We want hardware as fast as easy to develop with software.*

Hardware Revolution

*Pretty soon, you cannot buy something that is not connected*

Consumers will demand sophisticated hardware.

June Oven. Tovala, and other

3 types of IoT: Consumer, Commercial and Industrial.

Smart Tech

Connected fridge of tomorrow:

* Inventory + automated grocery ordering.
* Meal & diet plan, receipt assistant.
* User recognition, experience tailoring.
* Allergy, food expiration, etc. warnings.
* Integration with oven, microwave, etc.

Microcontrollers will make the revolution possible.

Real IoT is powered by microcontrollers (MCUs).

Commodity chips. $2-$10

Low-energy, high-performance

General purpose input-output (GPIO)

Digital + Analog

Built-in Protocol Support (SPI, I2C, Serial, CAN, and others)

Analog-to-Digital (ADC) Converters

Digital-to-Analog Converters

Gateway Connectivity (BLE, WiFi, others)

*Microcontrollers can run for a long-time*

Netduino = Arduino form factor running the .Net MicroFramework.

STM32f4 Microcontroller 164k RAM 1.4mb Flash

Visual Studio (Windows + Mac)

Debugging, events, etc. No generics.

Excellent prototyping/ dev board.

Lots of commercial NetMF projects; it’s even in space with NASAJPL

Can embed for <$15

developer.wildernesslabs.com

community.wildernesslabs.com

Dehydrator3000 (a low temperature oven)

*Web server that forecast it’s iep???*

*Control the temperature*

*Turning off the mode will cool down the machine.*

**Hacking Connected Appliances**

Start building the hardware of tomorrow, today.

Controlling **household mains electricity** with a relay.

Regulating temperature with Proportional, Integral, Derivative (PID) controllers.

Rotary encoder powered LCD menu UI.

Exposing control via Web API

Connecting to that Web API from a Xamarin mobile app.

*Dehydrators in the market is around $300, died after 1 year.*

*We reversed engineered the circuit.*

*Turns out there is a 1972 microcontroller*

**System**

Dehydrator App

* Encoder
* LCD
* Dehydrated Controller
  + PID
    - Fan
    - Heater
* Web API => Xamarin Forms App

**Enclosure**

Netduino

Breadboard

Relay

Power Distribution

LCD + Rotary Encoder

github.com/wildernesslabs/3D\_Print\_Designs

*Atom A8, 270usd, print everything needs here.*

Netduino.Foundation beta by wilderness labs

Nugetized hardware and peripheral framework API

**Power Control**

Output Port relay =

New Output Port(Pins.GPIO\_PIN\_D2, false);

relay.Write(true);

Netduino.Foundation:

Var.relay = new Relay(N.Pins.GPIO\_PIN\_D1);

relay.IsOn = true;

House hold electricity (110V/240V) is controlled by a relay.

Relays are electromechanical and isolate circuits

Controlled by a simple on/off via a digital I/O pin.

Baseboard @ 3D Print Designs repo

BOM = Build Of Materials?

**TextDisplayMenu** (library)

JSON-powered

Use with any LCD via Netduino.Foundation (GPIO, I2C, Serial, SPI)

Navigate with IRotaryEncoder, or IButtons.

Editable Items.

protected void InitializeMenu()

{

// initialize menu

menu = new Menu(\_display, \_encoder,

Resources.GetBytes(Resources.BinaryResources.menu),

true);

\_menu.ValueChanged += HandleMenu Value Change

……

}

switch (e.Command)…

*What is the version C# support? .NET?*

*Answer: basically .NET 2.0 (no generics), Frankenstein of C#.*

*You don’t have tasks.*

PID

Proportional, Integral, Derivative

*Originally started with Ships.*

*The simplest way to control temperature is to turn on the heater, if you hit my desired temp, turn off the heater. Then would have heat inertia. This cycles again.*

*PID tries to go to the value as fast as possible, then uses the on and off cycle.*

i.e. 20 celsius to 100 celsius, then 95-100-95-100….

PID keeps drone to be not out of flight

PID for cars on cruise control

PID is on your HVAC  
PID is on your oven

PID Guide

Netduino.Foundation

StandardPIDController

IdealPIDController

PID – keeping the robot upright, steering, etc.

Dehydrator App Solution Architure – Controllers

Main() launches App.

App instantiates peripherals

Features managed by controllers

app.cs

Peripherals

* LCD
* RotaryEncoder
* FanRelay
* HeaterRelay
* AnalogTempSensor
* DehydratorContoller.cs
* ….

Feedback loop

-> input [ CONTROLLER] -> output --- *control signal (gain) ---*> [SYSTEM] – *actual signal (feedback)* ---> loops

PID in Action – Controller ctor

**PWM = Post Swift Modulation**

*High frequency of turning off and on.*

*This will average.*

PID in Action – Temperature Thread

*Hey run this 50% of the time, PID will calculate the duty cycle.*

*This runs every 5 seconds.*

PID Tuning

**Web Server - Maple**

Purpose-built for Netduino

Modern, RESTful Web API/

Built-in JSON Support

get:/Status

post:/Turnon

post:/Turnoff

*we parse the form…*

One-Line UDP.

­\_server.Start(“dehydrator3000”, Initializer.CurrentNetworkInterface.IPAddress.ToString());

**Name IP**

Xamarin Mobile App

Xamarin.Forms + HttpClient

*I think mobile is the gateway to communicate between things.*

*A lot of these stuff will be interface-less*

**What future tech will you build?**

*If you can build .NET things, you can build it.*

hackster.io

Netduino Contest

Device Deadline – July 2nd

Project Deadline – July 31st

**Thanks**

slideshare.net/bryancostanich

newsletter: bit.ly/2rBfP4Y

github

blog.wildernesslabs.co

*do you support SSL? Yes and No. we are currently fixing.*

*We don’t support modern SSL for now. We are fixing that.*

*Next..*

*Smart Chicken Coop*

*Also, includes door opening.*

*Also, adding machine vision.*

*Smart Dumb Coffee*

*Has simon says*