Use of Arduino Hardware to Upgrade Obsolete Simulator Components

Andrew Jedlowski
Exelon IT – Simulator Applications

Exelon Generation

Topics

- Using Arduino hardware in the simulator
- **Using an Xbox controller as a simulator testing tool**
- **Questions**

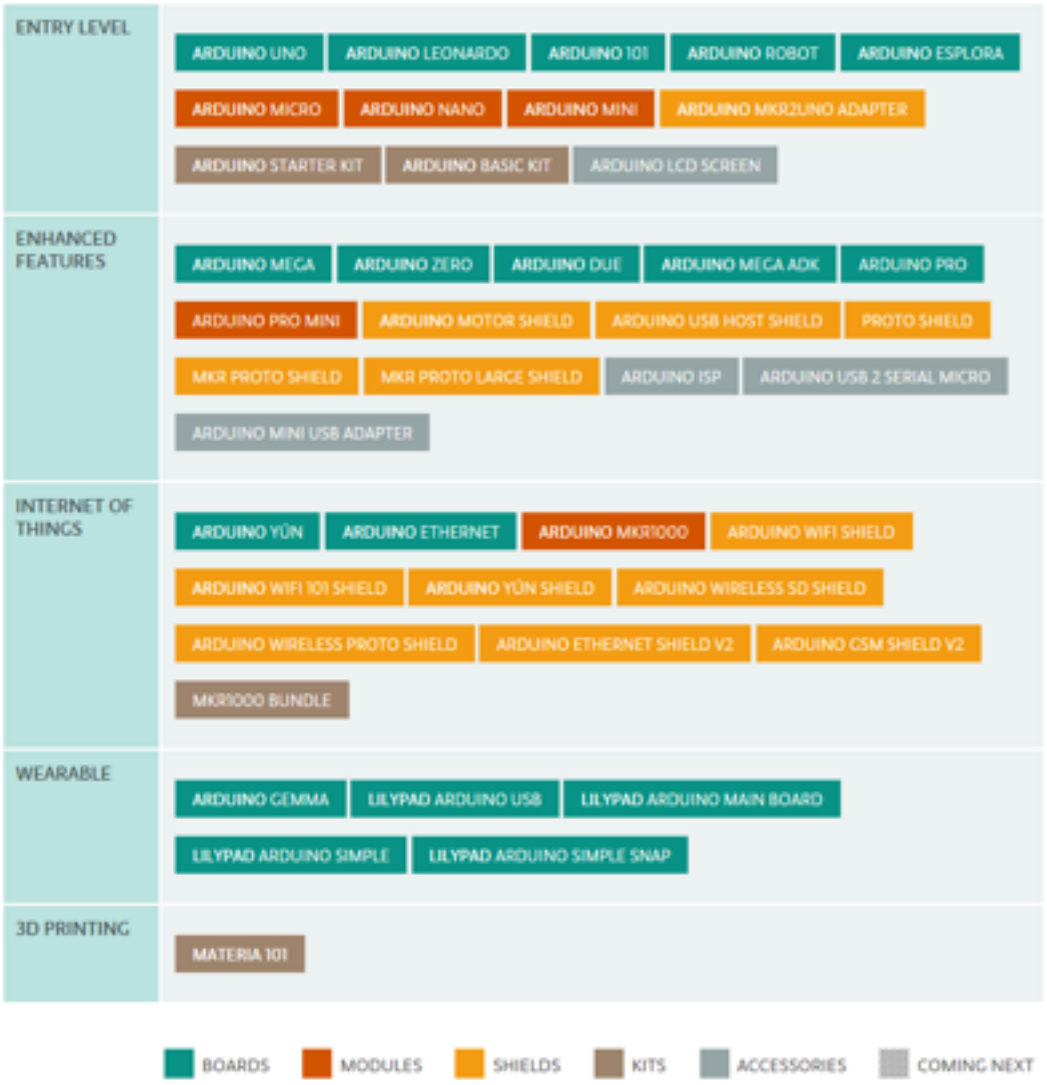
What is Arduino?

- Arduino is an open-source prototyping platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing.

- All Arduino boards are completely open-source, empowering users to build them independently and eventually adapt them to their particular needs. The software is open-source, and it is growing through the contributions of developers and the Arduino community worldwide.

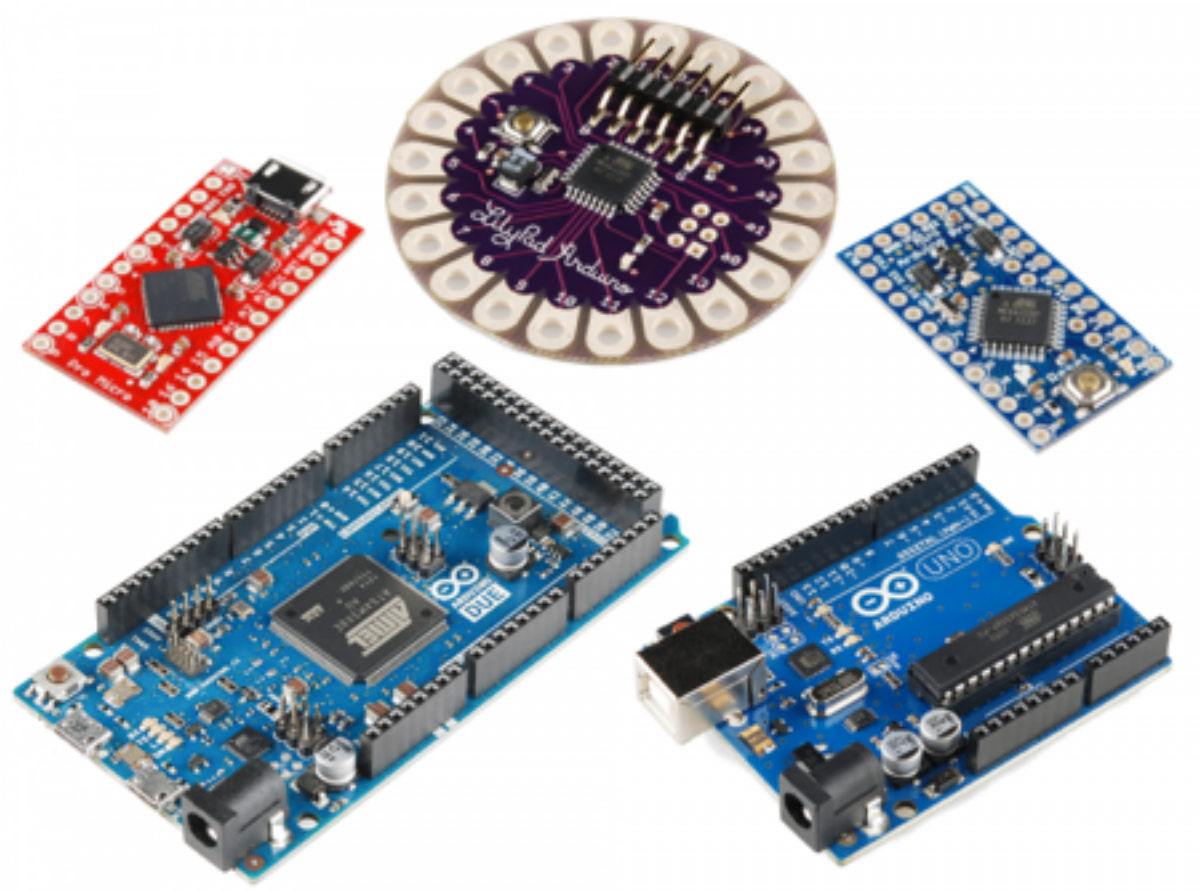
- Arduino shields are modular circuit boards that piggyback onto your Arduino to instill it with extra functionality

Arduino Products



Exelon Generation

Arduino Products



Dresden/Quad Cities SPING Arduino Upgrade

- Current Dresden/Quad Cities simulator radiation monitor is simulated on a VME based embedded system that was developed in house in the ’90s.
- The hardware interface is built around a custom designed wire wrapped board that we do not have resources to maintain.
- RS232 interface back to the simulator
- We also no longer have a Sun Unix workstation that will run the cross compilers for software development.
- There has been talk of upgrading the plant system several times throughout the years, so we’ve just lived with the system as is.

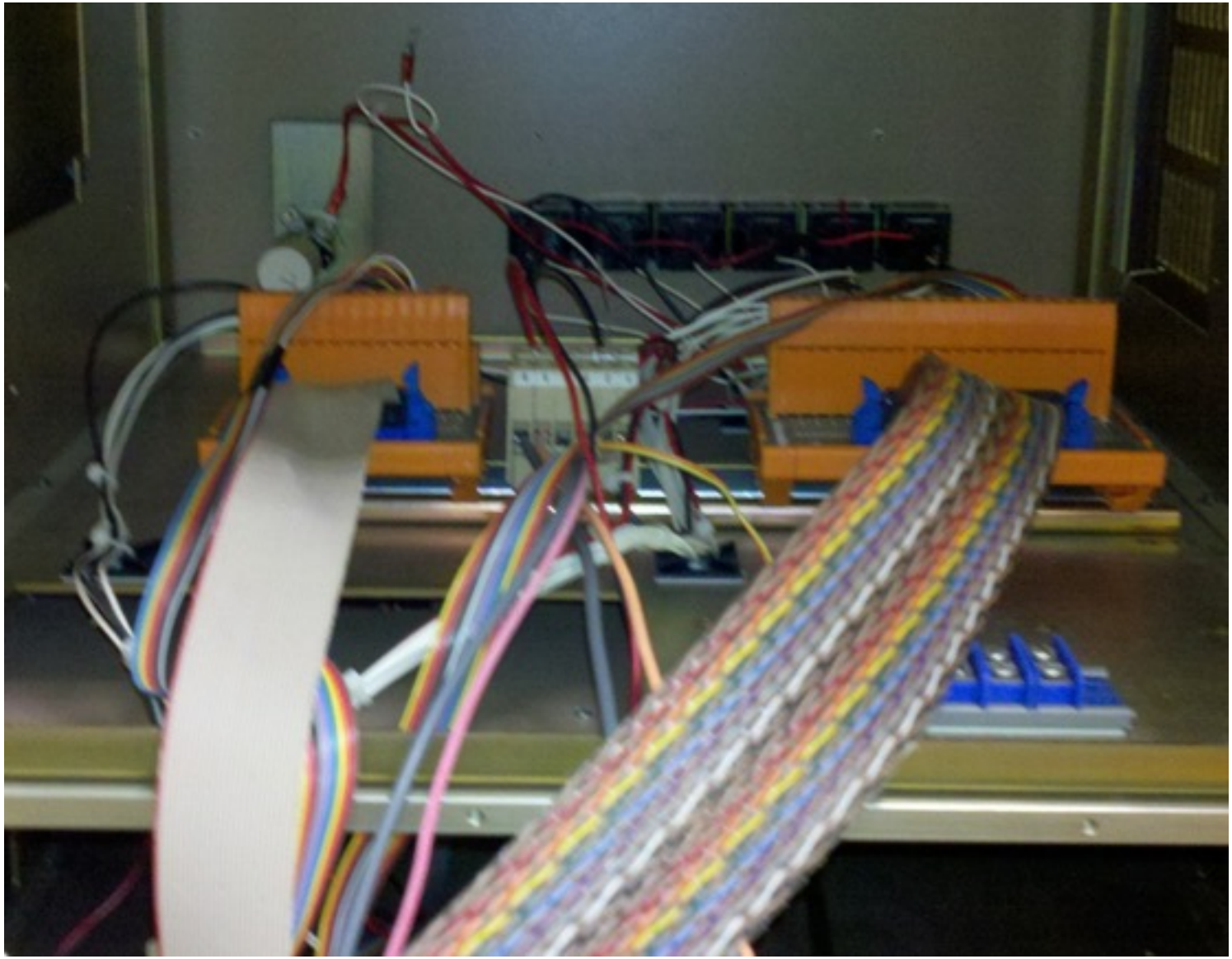

STATION RAD SAMPLING
CONTROL TERMINAL A
STATUS LIGHTS
1
2
3
4
5
6
NORM
MAINT
FAIL
TREND
ALERT
PRESS TO
SILENCE ANNUNCIATORS
SYSTEM ANNUNCIATOR
eberline
SYSTEM OPERATOR CONSOLE
CONTROL
MONITOR
MASS STORAGE
PRINTER FAIL
INTEGRATED RESOURCES INC
TL-PRINTER 02X
PRINTER A
DISABLE
ENABLE
COMMAND ENABLE
EDIT ENABLE

Dresden/Quad Cities SPING – Hardware Requirements

I/O Counts per SPING unit (2 units total)

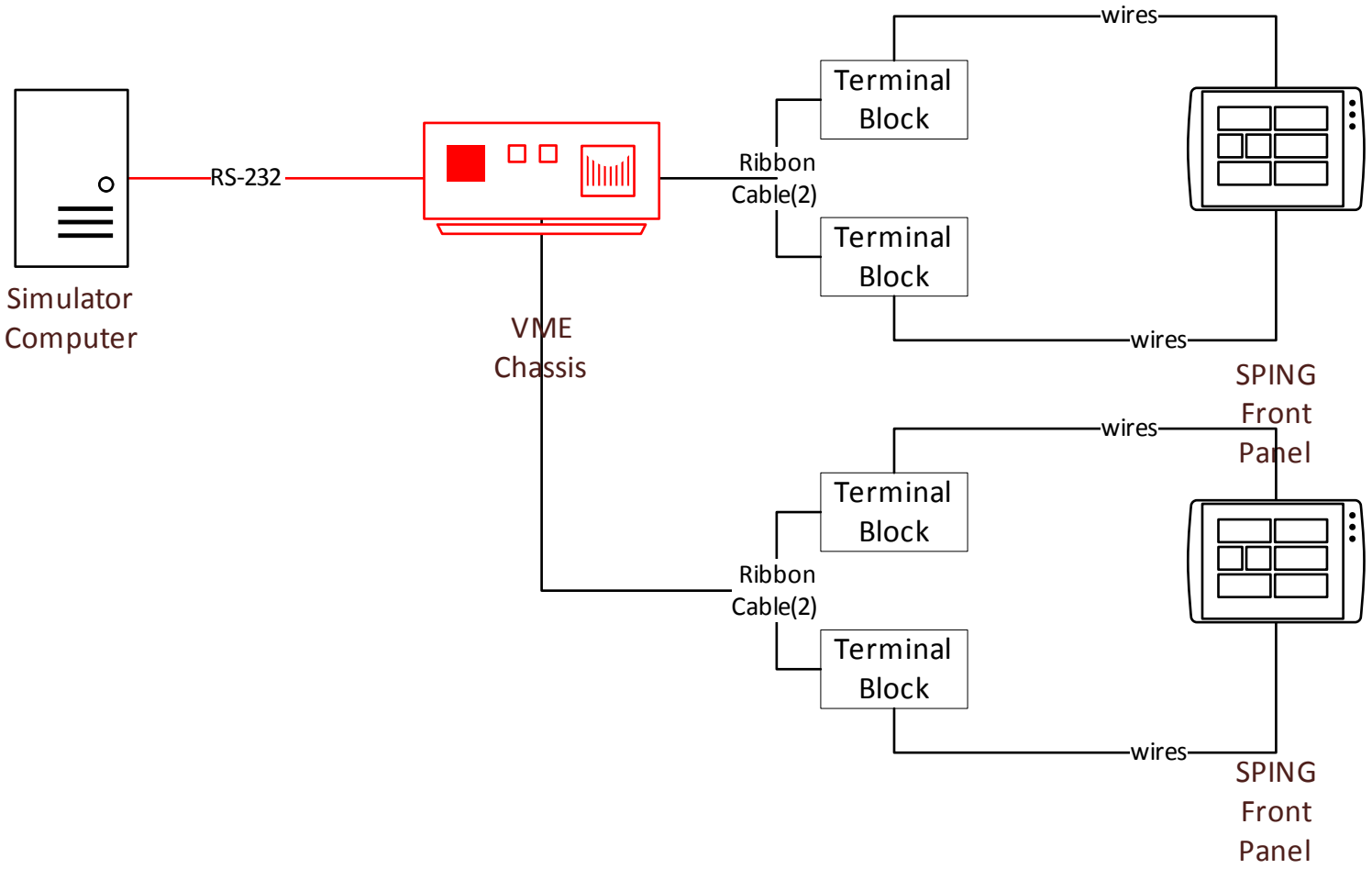
- 16 Digital Outputs
  - 6 Alarms
  - 1 Horn
  - 8 LEDs
  - 1 Transmit for LCD Display
- 20 Digital inputs
  - 16 2 4x4 keypads
  - 2 Keylocks
  - 1 paper change switch
  - 1 Alarm reset pushbutton
- 1 Serial Port (Printer)

Requirement to use Existing Ribbon cables

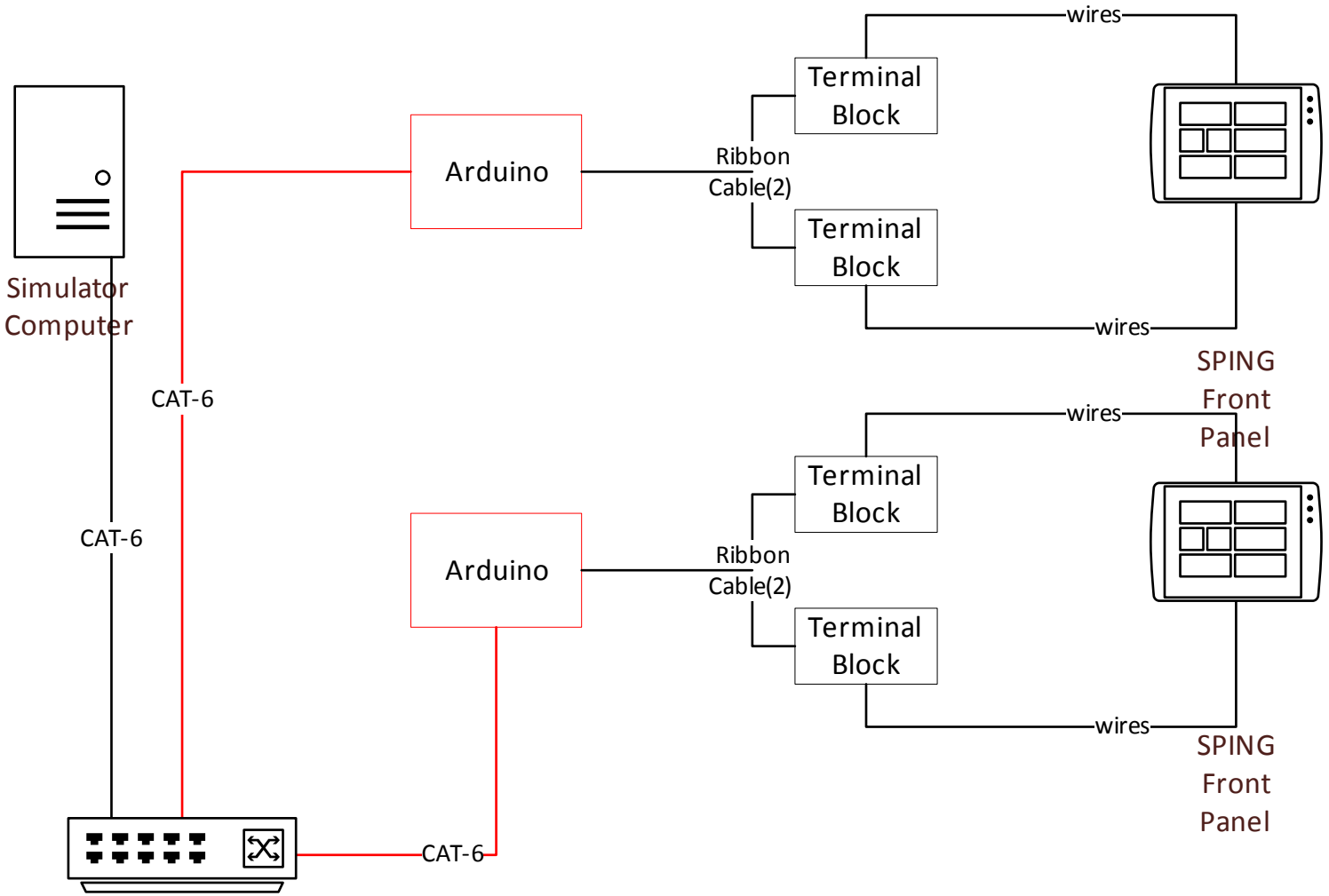



MVME
105
FAIL
HALT
RUN
ABORT
RESET

Dresden/Quad Cities SPING – Current Architecture



Dresden/Quad Cities SPING – Current Architecture



Simulator Network

What Arduino Hardware Did We Choose?

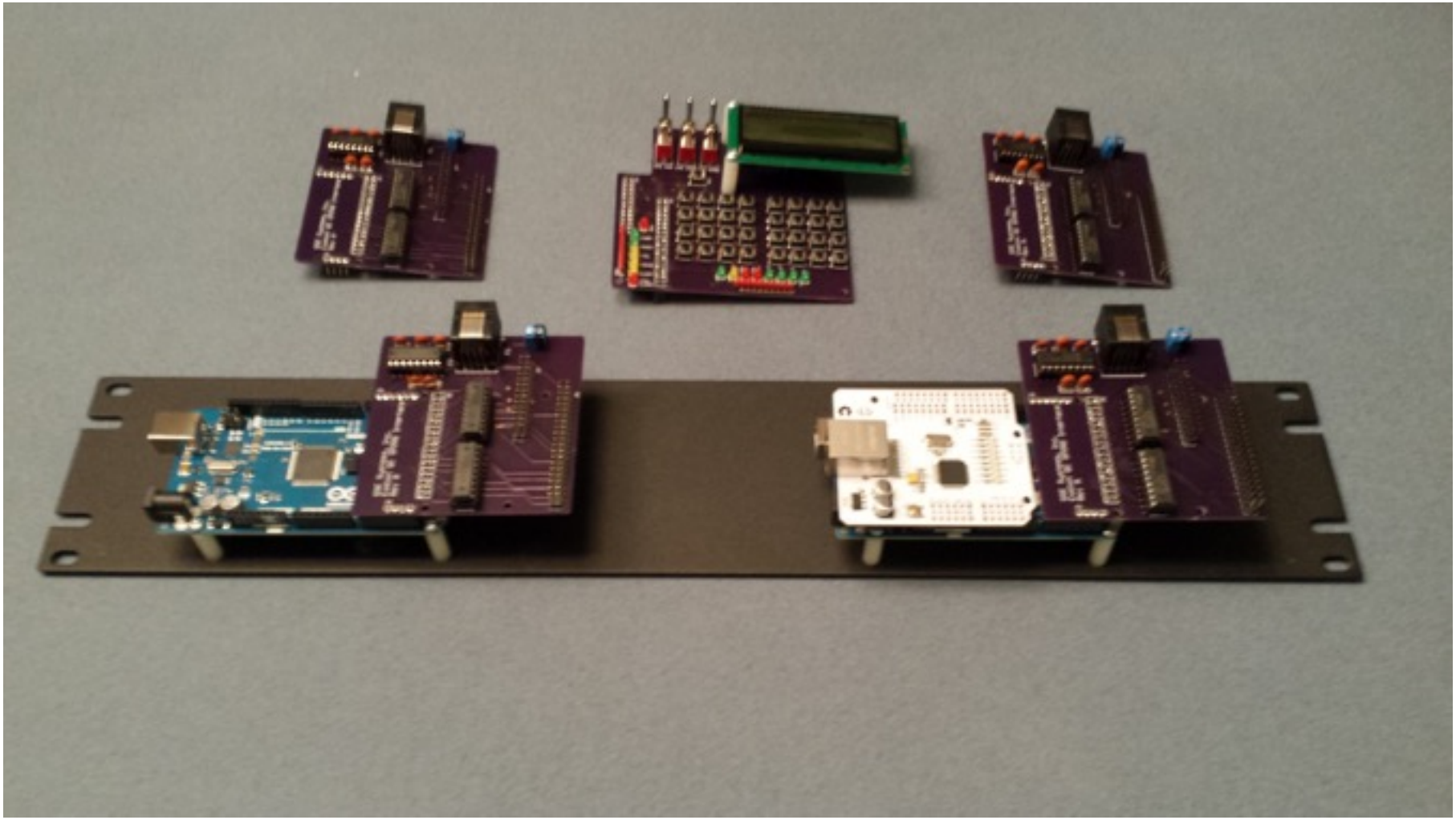
- **The Mega 2560 is a microcontroller board based on the ATmega2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs (hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.**

- **The Arduino Ethernet Shield V1 allows an Arduino board to connect to the internet. It is based on the Wiznet W5100 ethernet chip. The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections.**

Dresden/Quad Cities SPING – Hardware

Hardware - per SPING unit (2 units total)

- 1 Arduino Mega
- 1 Ethernet Shield
- 1 LCD Display
- Custom Interface Board
  - Routes signals from existing ribbons to Arduino
  - RS232 Interface and conversion from TTL



Dresden/Quad Cities SPING – Installation

- Run network cables
- Mount Arduino hardware
- Mount and wire new LCD (3 wires into spare points on terminal block)
- Move ribbon cables to interface board
- Run new serial cable from interface board to printer

In all about 2 hours to install and test

Topics

- **Using Arduino hardware in the simulator**
- Using an Xbox controller as a simulator testing tool
- **Questions**

Xbox Controller for Simulator Testing

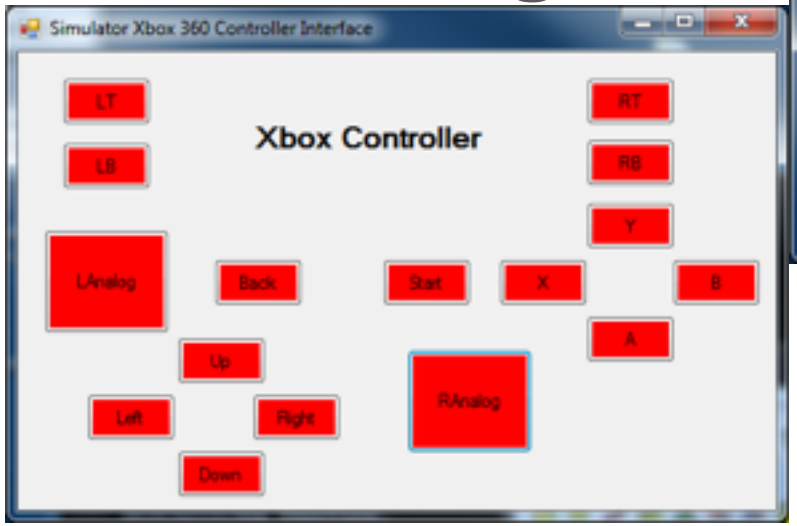
- **After a recent FAT test, the Braidwood simulator coordinator made the following request:**

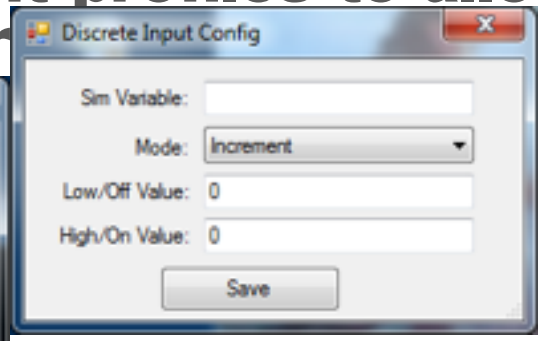
Since the FAT I have been thinking about the difficulty of running certain things from a desktop. Rod control is probably the least easy to use since you have no fine control using the graphic input and a mouse. My thought was what if we took an I/O device like a game controller and pirated the hardware buttons, stick etc and replaced with our input devices, like an in-hold-out in place of the joystick and switches/buttons as digital I/O.

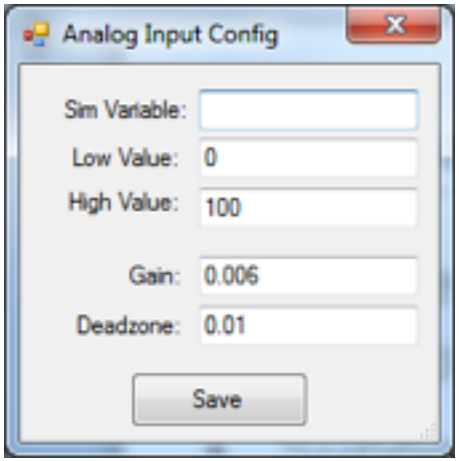
- **After thinking about this for a while, it didn't seem as crazy as I first thought.**
- **Latency in touch screens and graphics programs makes the little control tweaks difficult**
- **I happened to have an intern available, and an old Xbox controller in a drawer at home, so….**

Xbox Controller for Simulator Testing

- We developed a tool that will map controls on an Xbox controller to the simulator.
- Supports both DI's and AI's
- DI's support latched, toggle and rotary type switches
- AI's support custom gain for configuring sensitivity of control
- Plan to support different profiles to allow for different testing scenar



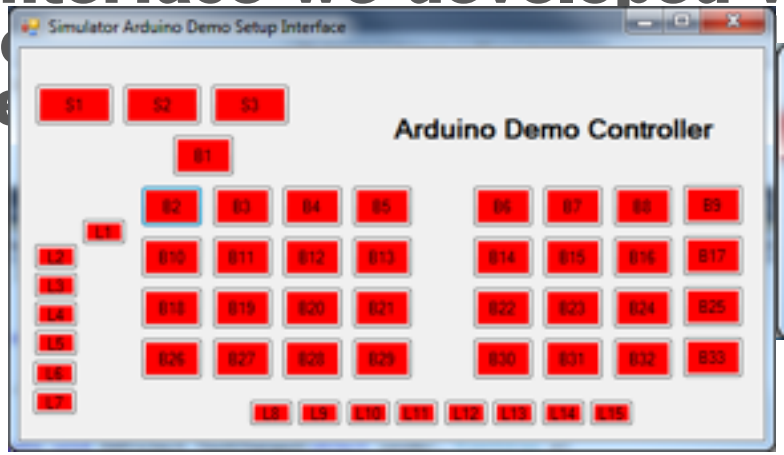


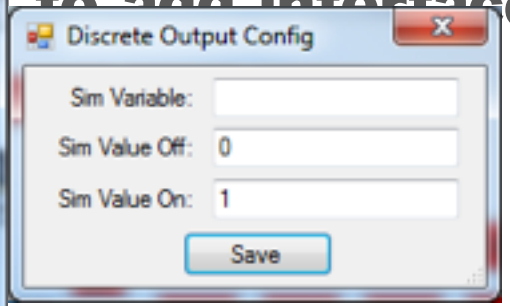


Exelon Generation

Next Step, Build a Arduino Based Test Control Panel

- The next step in this project is to build a small portable box with real simulator switches and pots that can be configured using the previously developed software.
- The switches and pots will be wired into an Arduino controller
- The Arduino will communicate to the simulator via USB
- We did a proof of concept of this using the and the results look promising.
- The initial interface we developed will also support configurable to add interface lights to the





Exelon Generation

Questions??