

Intro to Arduino Zero to Virtual Prototyping in Seven Hours

Linz Craig, Erik Hallqvist, Jordan McConnell, Dave Stillman and Aaron Weiss

Overview of Class

Getting Started: Installation, Applications and Materials Electrical: Components, Ohm's Law, Input and Output, Analog and Digital

One Hour Break For Lunch

Programming: Split into groups depending on experience Serial Communication Basics: Troubleshooting and Debugging Virtual Prototyping: Schematics and PCB Layout in Fritzing



Getting Started				
Installation:	Arduino (v.22) Java and Drivers			
Materials:	SIK Guide Analog I/O, Digital I/O, Serial,			
Applications:	Arduino IDE for programming board Processing			



















<section-header><section-header><section-header><list-item><list-item><list-item><list-item>



Analog and Digital

- All Arduino signals are either Analog or Digital
- All computers including Arduino, only understand Digital
- It is important to understand the difference between Analog and Digital signals since Analog signals require an Analog to Digital conversion

Output

Output is any signal exiting an electrical system

- Almost all systems that use physical computing will have some form of output
- The outputs in SIK include LEDs, a motor, a servo, a piezo element, a relay and an RGB LED

Output

Output is always Digital

To Output a Digital signal (On or Off) use this code: *digitalWrite (pinNumber , value);* Where value is **HIGH** or **LOW**

To output a signal that pretends to be Analog use this code:

analogWrite (pinNumber, value); Where value is a number 0 - 255

Output

Output is always Digital

Using a Digital signal that pretends to be an Analog signal is called Pulse Width Modulation

Use Pulse Width Modulation, or P.W.M., for anything that requires a signal between HIGH and LOW

P.W.M. is available on Arduino pins # 3, 5, 6, 9, 10, and 11



Input Input is any signal entering an electrical system · Both digital and analog sensors are forms of input · Input can also take many other forms: Keyboards, a mouse, infared sensors, biometric sensors, or just plain voltage from a circuit

Analog Input

- · To connect an analog Input to your Arduino use Analog Pins # 0 - 5
- · To get an analog reading: analogRead (pinNumber);
- Analog Input varies from 0 to 1023 on an Arduino

Analog Sensors Examples: Sensors Variables Values Signals Mic Photoresisto Potentiomet

Mic	Volume	Decibels	Voltage
Photoresistor	Light	Photons	Voltage
Potentiometer	Dial Position	Resistance	Voltage
Temp Sensor	Temperature	Celsius	Voltage
Flex Sensor	Bend	Resistance	Voltage
Accelerometer	Motion/Tilt/Acceleration	Acceleration	Voltage

Digital Input To connect digital input to your Arduino use Digital Pins # 0 – 13 (Although pins # 0 & 1 are also used for serial) • Digital Input needs a pinMode command: pinMode (pinNumber, INPUT); Make sure to use caps for INPUT • To get a digital reading: digitalRead (pinNumber); • Digital Input values are only HIGH (On) or LOW (Off)

Digital Sensors

- Digital sensors are more straight forward than Analog
- No matter what the sensor there are only two settings: On and Off
- Signal is always either HIGH (On) or LOW (Off)
- Voltage signal for HIGH will be a little less than 5V on your Uno
- Voltage signal for LOW will be 0V on most systems

Questions?

