

Physical Computing: Controlling The World Around You with Arduino

By Ryan Hagan, Monta Elkins and Thomas Weeks

Hardware Gracefully provided by ACI, Modea, and Eksdyne Research

Software Setup

- Linux
 - RHAT: # yum -y install arduino' # (reqs: uisp avr-libc avr-gcc-c++ rxtx avrdude)
 - Ubuntu: \$ sudo apt-get install arduino # (reqs uisp avr-libc gcc-avr avrdude librxtx-java)
 - or
 - Download & Install Software from: <http://www.arduino.cc/en/Main/Software>
- Linux T-Shooting:
 - check permissions of /dev/ttyUSB0
 - sudo usermod -a -G uucp,dialout,lock \$USER
 - or may have to chmod 777 /dev/ttyUSB0
- Windows T-Shooting:
 - Check COM port settings
- Mac T-Shooting:
 - check device permissions

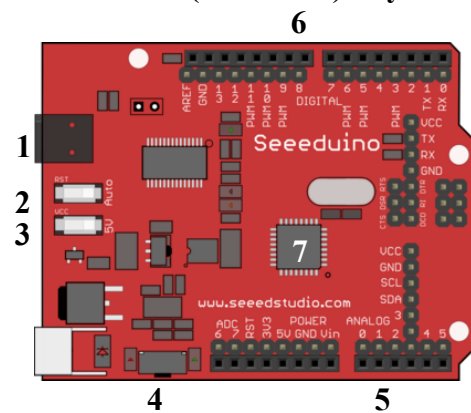
Part-I: Physical Computing Intro & Overview

- Physical Computing (or Having Fun with Magic Blue Smoke)
 - Make / Hackaday , Seeeduno
 - Arduino, Uno, Teensy, Mega
 - 16KiB Flash, 512B EEPROM, 1KiB SRAM
 - Digital I/O, Analog I/O, Actuators
 - Shields
- Hardware Hacking Tips
 - Success lies on the far side of failure
 - Fail Cheap
 - Loosely coupled subsystems

Part-II: Load and Test "Blink" Program

- Select Tools / Board and "Arduino Diecimila ... w/AtMega168"
- Select correct Tools / Serial Port
- Load or type "Blink" program
 - Click "upload & run" icon to compile, upload and run your program
- Discuss and play with Analog In

The Arduino (seeeduno) Layout



- | | |
|--------------------------|--------------------------------|
| 1) Mini-USB/B connector | 5) Analog inputs |
| 2) Reset man/auto switch | 6) Digital/PWM I/O lines |
| 3) 3.3v / 5v switch | 7) AtMega 168 μ Controller |
| 4) Reset switch | |

```
// Blink
int ledPin = 13; // LED connected to pin 13

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

void loop()
{
  digitalWrite(ledPin, HIGH); // turn LED on
  delay(1000);                // wait a second
  digitalWrite(ledPin, LOW); // turn LED off
  delay(1000);                // wait a second
}
```

- **Part-II "Blink" Hardware build**, breadboard + 1 LED + resistor
 - LED positive (long) to digital pin2
 - LED negative (short) to resistor
 - Resistor to GND (after digital pin 13)
 - in program, change digitalWrite(13) to pin "2"
 - upload & run

• **Part-III: "Chaser & Pot Delay"**

A) **Hardware build: Chase Lights**,

breadboard + 7 LEDs + resistor

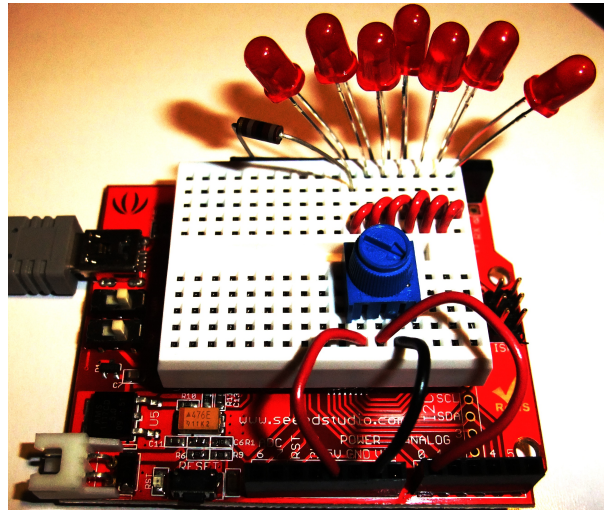
- LED positive (long) legs to digital pins 2,3,4,5,6,7,8 [area 6]
- LED negative (short) legs to breadboard, tied together with jumper
- All LED negative legs to resistor
- Resistor to GND ([area 6] after pin 13)
- Modify Program to output to LED pins
- Testing, mentoring, t-shooting

B) **Hardware build: Analog Read & Delay**

breadboard + 7 LEDs + resistor + potentiometer

- add potentiometer (blue knob) for variable chase light speed
- Wire pot leg pin 1 to "5V" (on Power header side between [4] and [5])
- Wire pot leg pin 2 (middle) to "Analog5" [5] (shown above hooked to Analog0 pin)
- Wire pot leg pin 3 to GND (on Power header between [4] and [5])
- Use "analogRead(5);" in program to digitize the pot value (from 0 to 1023 max)
- Replace "delay(1000);" with delay(analogRead(5));" to use the pot read as the delay
- Compile, upload and run code
- Twist knob to adjust chase light speed (delay)

Try this: See data with Serial.print(analogRead(5));" to see your pot value in the serial console



-Useful Arduino Links:

Arduino Playground: <http://www.arduino.cc/playground/>

Arduino Cheat Sheet: <http://sites.google.com/site/mechatronicsguy/arduinocheatsheet>

Hardware/Parts Sites:

Microcontroller Hardware

<http://www.sparkfun.com/>

<http://www.seeedstudio.com/>

Misc. Parts and Surplus

<http://www.allelectronics.com/>

<http://www.jameco.com/>

<http://www.mouser.com/>

<http://www.alltronics.com/>

<http://www.mpja.com/>

If interested in free classes like this, help us start a "MakerSpace" here in Blacksburg!

Indicate interest on our sign-in sheet, or contact us directly:

ryan@ryanhagan.net

monta.rbtc@geekslunch.com

tweeks-rbtc@theweeks.org