A PRIMER ON MICROCONTROLLERS

Presented by Connor Taylor

Microcontrollers are all around you

- Modern Cars
  - adjust engine parameters
- Remote Controls
- Smart Watches
  - Pebble Smartwatch: STM32
- MP3 Players
- Traffic Lights
- Avionics
- Factory Controls

Common Microcontrollers

- Atmel AVR (Focus of today's presentation)
- Texas Instruments MSP
- Microchip PIC
- STMicroelectronics STM32
- And many more…

Blog.xeltek.com

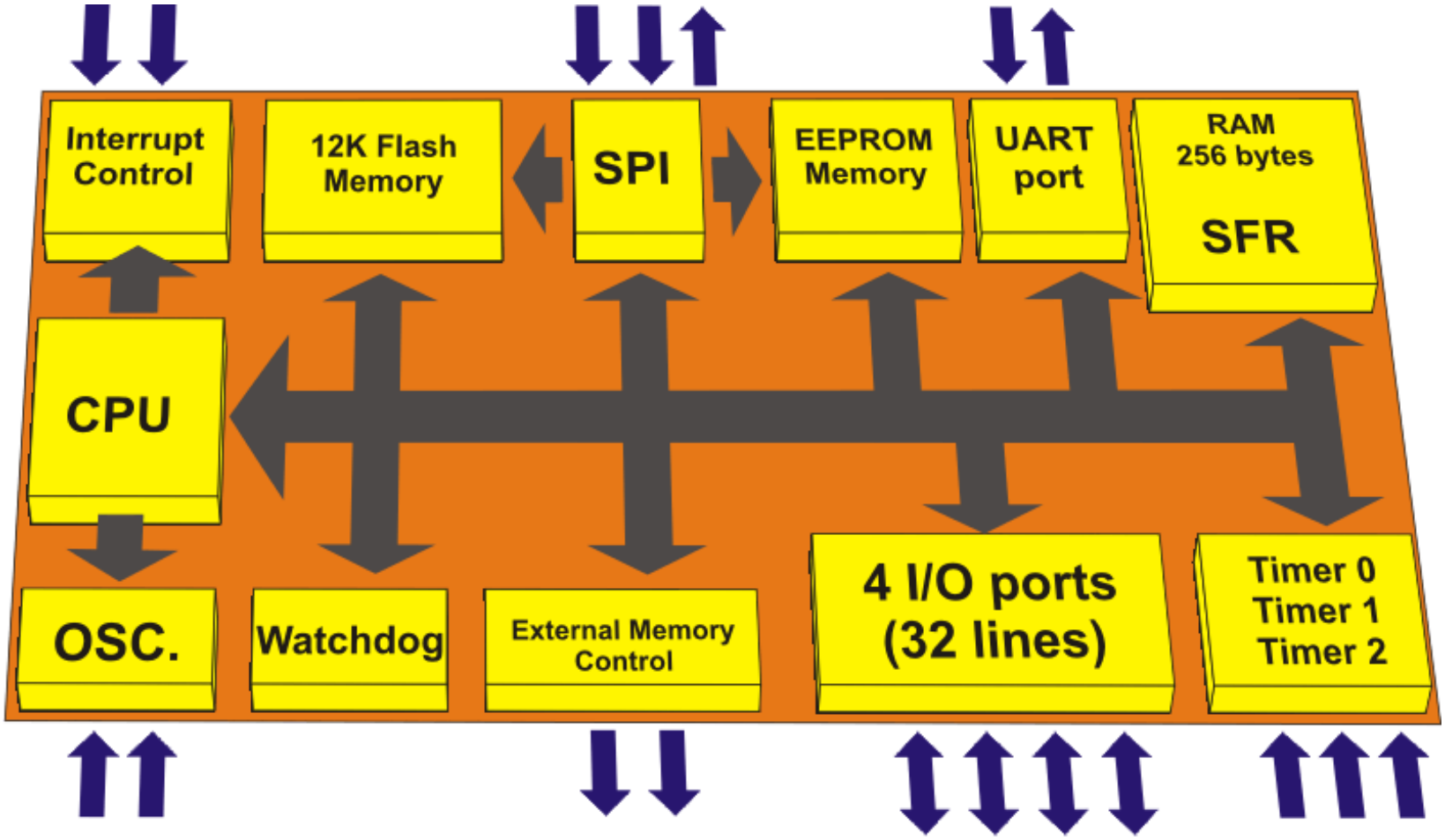
- There are over 36 different manufacturers of microcontrollers

What Is a Microcontroller?

- A computer in a single chip
  - Processor
    - Single core
    - multi-core (e.g. Propeller 1)
  - Memory: Flash, SRAM, EEPROM
  - IO Peripherals
    - GPIO, UART, CAN, SPI, I2C, ADC, DAC
- Lower speed/power
  - Typically 1-64MHz
- Typically runs a single program repeatedly

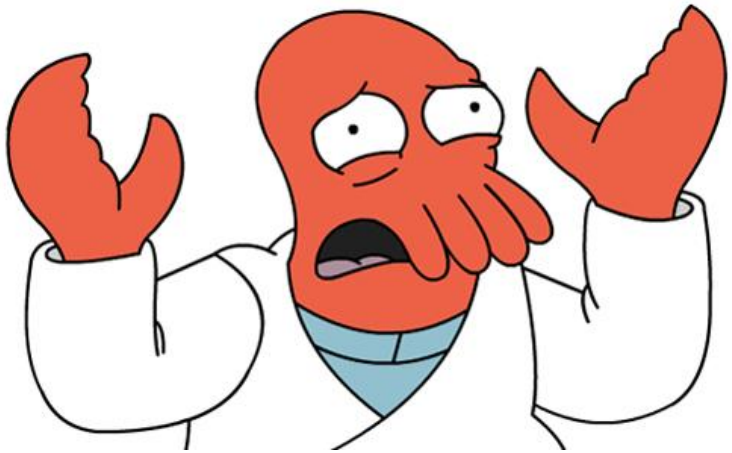
en.wikipedia.org

What's Inside a Microcontroller?



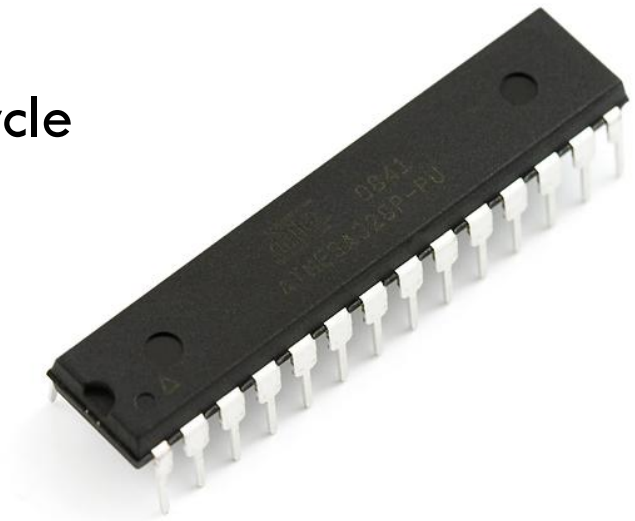
Why Use Microcontrollers?

- Programs can easily interface with the real world through peripherals
  - Interrupt driven program events
- Low cost
  - Most a few dollars or less
- Small size
  - Some less than 5mm x 5mm
- Low Power
  - Typically less than 100mW
  - Ultra low power microcontrollers: < 5uW
- Easy firmware changes
  - Updated through USB (on the nicer ones)
- "Simple" implementation
  - Easier than FPGA, PLCs (Ladder Logic)



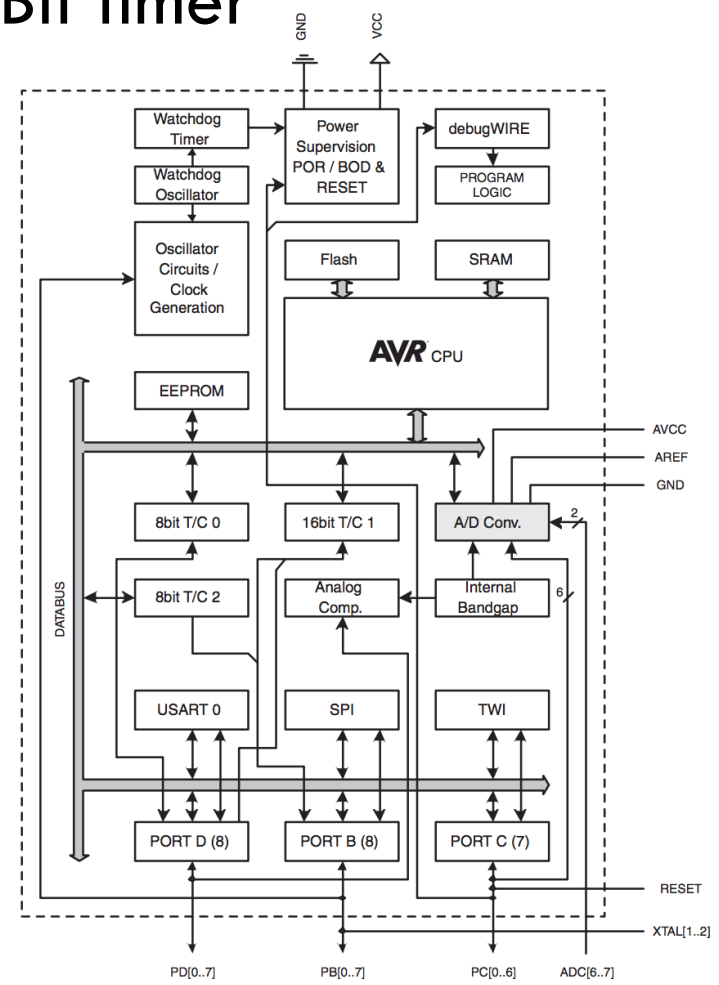
ATmega328P – Processor and Memory

- Produced by Atmel
- AVR Core
  - 8-Bit
  - Based on RISC architecture
  - Most operations take one clock cycle
- 32KB of Flash
  - Program storage
- 1KB of EEPROM
  - Data modifiable by program
  - Non-Volatile
- 2KB of SRAM
  - Stores program variables



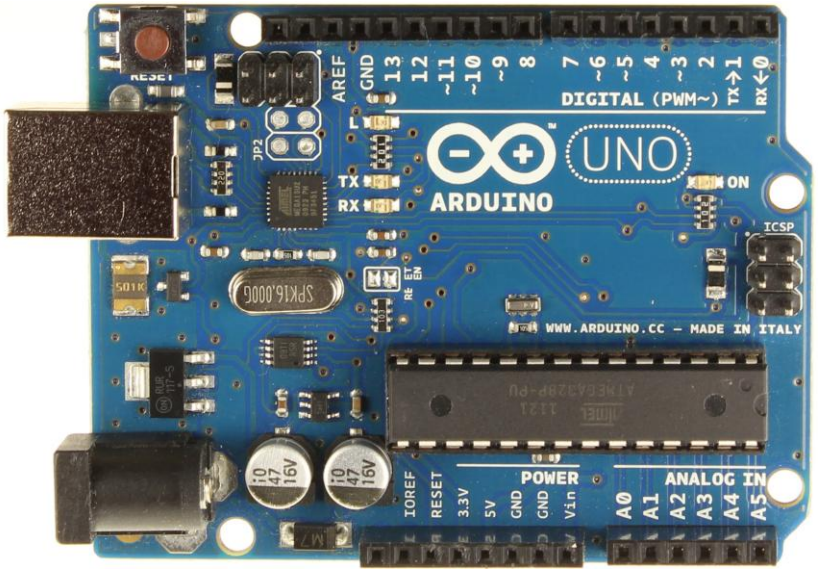
ATmega328P – Features

- Two 8-Bit timers and one 16-Bit timer
  - PWM, Counters
- Six channel 10-Bit ADC
- USART
- SPI Bus
- I2C Bus
- 23 GPIO
- 0-20MHz Clock



Arduino

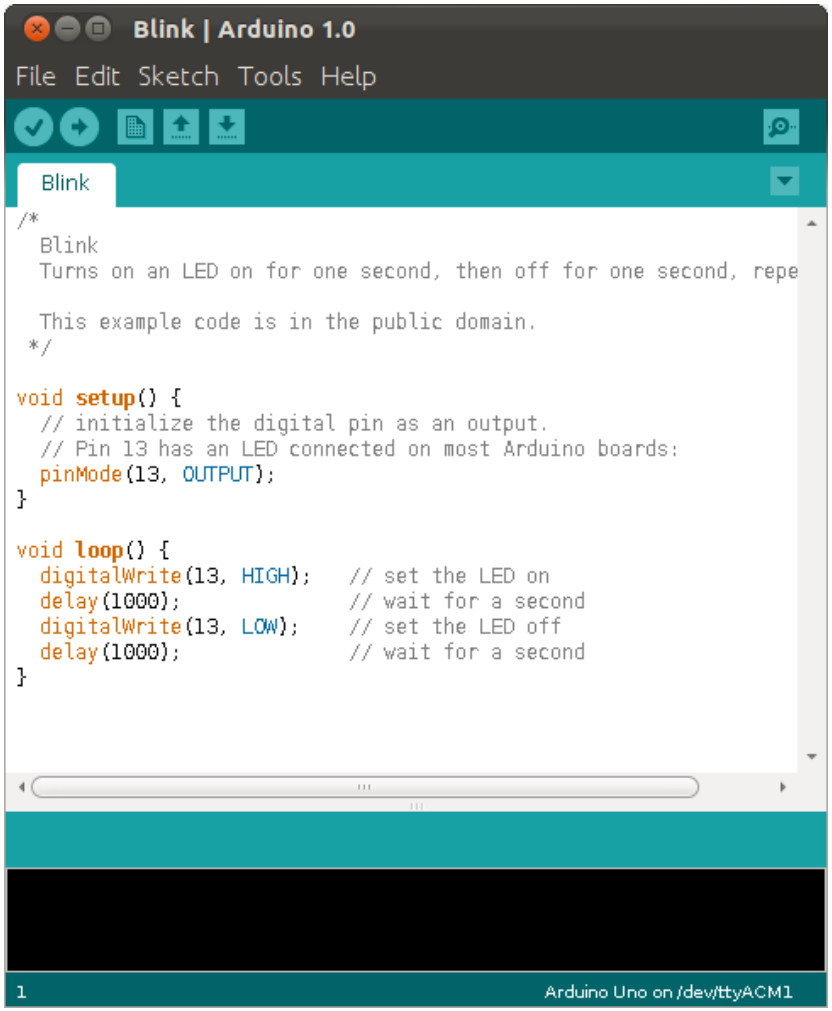
- Uses ATmega328P
- Onboard
  - USB-UART
  - Voltage Regulators
  - Indicator LEDs
- Standardized Form Factor
  - Many useful daughterboards
- Useful Bootloader
  - No external programmer needed

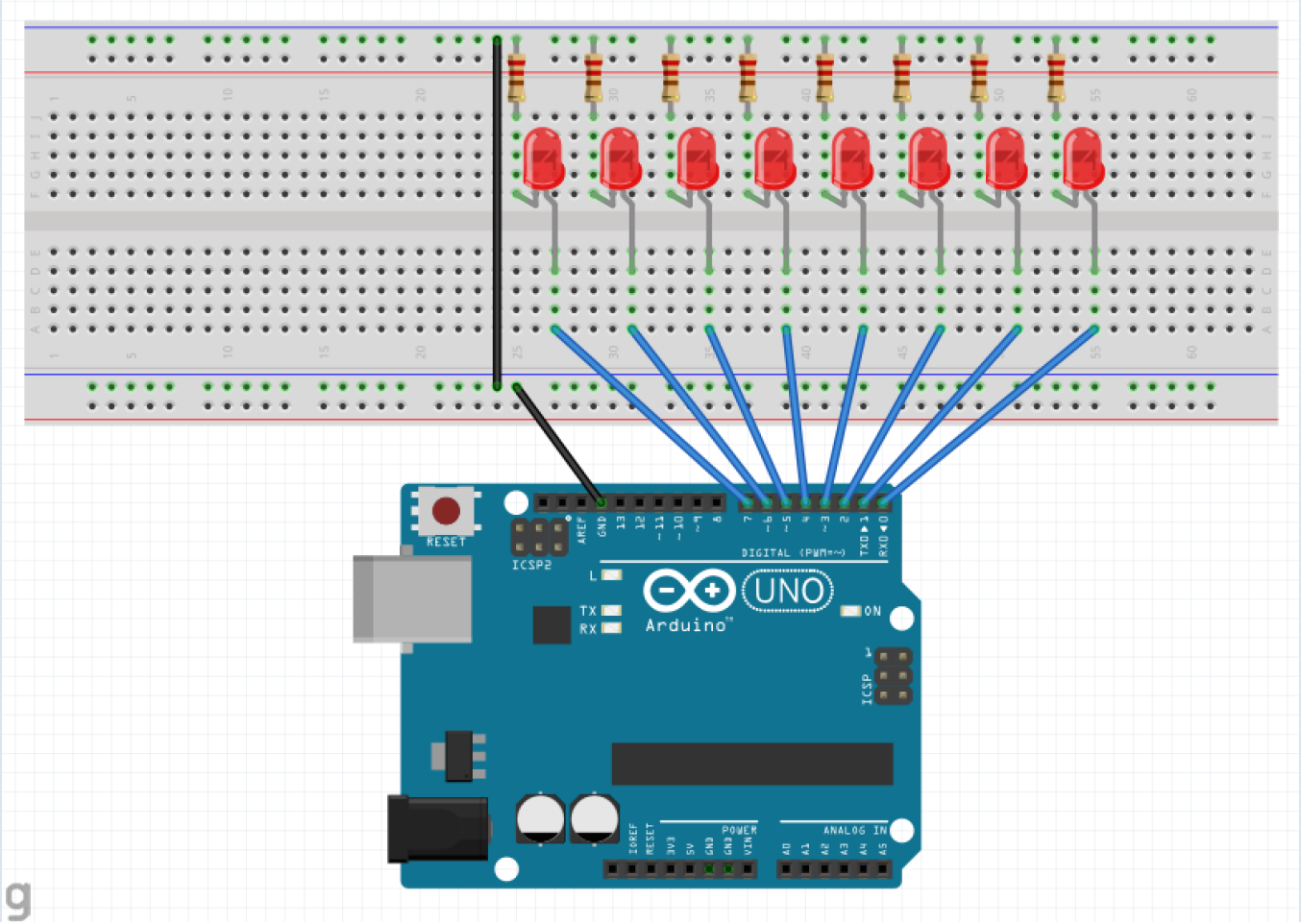


Arduino IDE

- Programming at the click of a button
- Useful libraries pre-included

- void setup()
  - Runs once at startup
- void loop()
  - Repeats forever





Activity: Nightrider Strobe

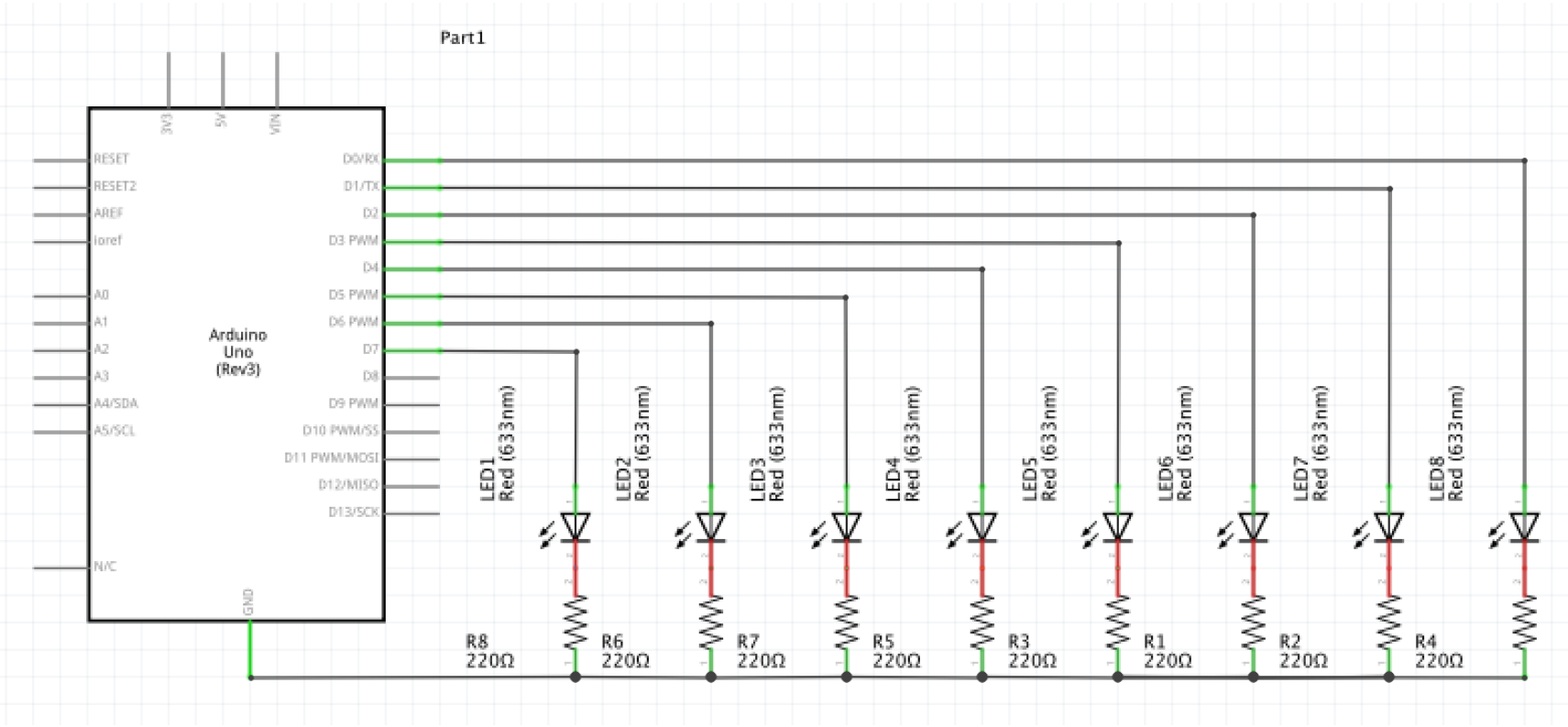
Required Materials

- Breadboard
- Arduino (and cable)
- Jumpers
- 8 LEDs
- 8 220Ω Resistors

Procedure

- Step 1: Open Blink example in Arduino IDE
- Step 2: Upload Program
  - Check to see that onboard LED blinks
- Step 3: Build the circuit on breadboard
  - Don't forget LEDs are polarized: Longer lead is positive
- Step 4: Try to write the code necessary to make the LED go back and forth
- Step 5 (Optional/Advanced): Use two buttons to make a game of pong

Schematic


Part1
Arduino
Uno
(Rev3)
3V3
5V
VIN
RESET
RESET2
AREF
Ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
LED1 Red (633nm)
LED2 Red (633nm)
LED3 Red (633nm)
LED4 Red (633nm)
LED5 Red (633nm)
LED6 Red (633nm)
LED7 Red (633nm)
LED8 Red (633nm)
R8 220Ω
R6 220Ω
R7 220Ω
R5 220Ω
R3 220Ω
R1 220Ω
R2 220Ω
R4 220Ω

Useful Functions

- pinMode(PIN_NUMBER, DIRECTION);
  - PIN_NUMBER = 0-13 (corresponding to D0-D13)
  - DIRECTION = INPUT, OUTPUT
- digitalWrite(PIN_NUMBER, STATE);
  - PIN_NUMBER = 0-13 (corresponding to D0-D13)
  - STATE = HIGH (5V), LOW (0V)
- delay(DELAY_MS);
  - DELAY_MS = 0 - about 4 billion milliseconds

Example Code

```
#define DELAY_TIME_MS 100
void setup(){
  for(int i = 0; i < 8; i++)
    pinMode(i, OUTPUT);
}
void loop(){
  //Sequence each LED in order
  for(byte i = 0; i < 8; i++){
    for(int k = 0; k < 8; k++)
      digitalWrite(k, LOW);
    digitalWrite(i, HIGH);
    delay(DELAY_TIME_MS);
  }
  //Sequence each LED in reverse order
  for(byte i = 6; i > 0; i--){
    for(int k = 0; k < 8; k++)
      digitalWrite(k, LOW);
    digitalWrite(i, HIGH);
    delay(DELAY_TIME_MS);
  }
}
```

Direct Port Access Example

```
#define DELAY_TIME_MS 100
void setup(){
  DDRD = 0xFF;    //Set pins D0-D7 as outputs
}
void loop(){
  //Sequence each LED in order
  for(byte i = 0; i < 8; i++){
    PORTD = (1 << i);
    delay(DELAY_TIME_MS);
  }
  //Sequence each LED in reverse order
  for(byte i = 6; i > 0; i--){
    PORTD = (1 << i);
    delay(DELAY_TIME_MS);
  }
}
```