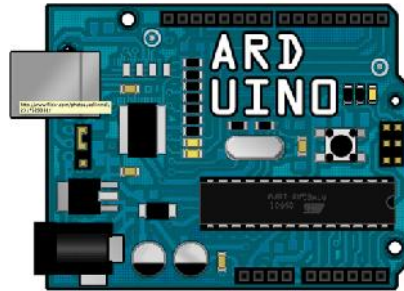


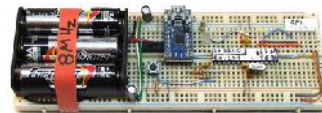
Mechatronics Student Lecture On Arduinos



*Some slides courtesy of Eoin Brazil
<http://www.slideshare.net/eoinbrazil/imediarduino08#btnNext>

Overview

- Hardware: What is Arduino?
 - Physical Computing
 - Versions and Capabilities
 - I/O and interfacing
- Software: Programming Arduinos
 - Sketches and their elements
 - Demo



Physical Computing

“Physical Computing is about prototyping with electronics, turning sensors, actuators and microcontrollers into materials for designers and artists.”

– Massimo Banzi, Tinker.it & Arduino Co-Founder



Daniel Pickem



What is Arduino?

The hardware The development environment The community



Category	Item	Views	Answers
Question	How do I connect my Arduino Uno to a PC?	102	14
Question	How do I connect my Arduino Uno to a PC?	827	103
Question	How do I connect my Arduino Uno to a PC?	91	133
Question	How do I connect my Arduino Uno to a PC?	111	80
Question	How do I connect my Arduino Uno to a PC?	15	48

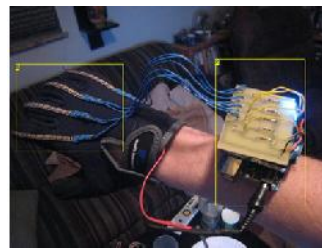
More info at <http://arduino.cc/en/Guide/Introduction>

Daniel Pickem



What can you do with it?

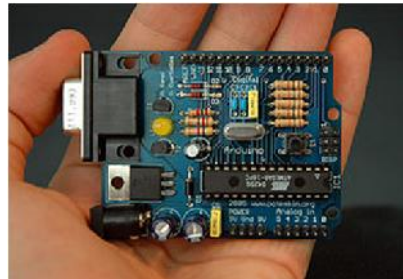
- [The LED Cube](#) – an 8x8x8 3D display
 - Contains 512 LEDs
 - Controlled by one Arduino
 - Instructions to build it yourself freely available
- [Animatronic Hand](#)
 - 5 flex sensors
 - 5 servos
 - 1 Arduino
 - Wireless transmission using XBees



Rowland O'Flaherty

Advantages of Arduinos

- Inexpensive
- Cross-platform
- Simple, clear programming environment
- Open source and extensible software
- Open source and extensible hardware
- Extensive project databases, Howtos and guides online
- Large number of hardware extensions (shields) available online



Daniel Pickem

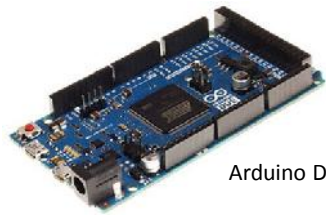
Available Versions



Arduino Uno



Arduino Mega



Arduino Due



Arduino Nano

Daniel Pickem

Comparison

	Nano	Uno	Mega	Due
Clock Speed	16 MHz	16 MHz	16 MHz	84 MHz
Digital I/O	14	14	54	54
Analog I/O	8	6	16	12
Flash Memory	16/32 KB	32	128	512
SRAM	1/2 KB	2	8	96
EEPROM	0.5/1 KB	1	4	0
Price	13 \$ ¹	15 \$ ¹	20 \$ ¹	45.70 \$ ²

¹ www.dealextreme.com
² www.mouser.com

Input / Output

- ★ 14 Digital IO (pins 0 - 13)
- ★ 6 Analog In (pins 0 - 5)
- ★ 6 Analog Out (pins 3, 5, 6, 9, 10, 11)

Daniel Pickem

Layout of an Arduino

- ★ Analog Reference pin (orange)
- ★ Digital Ground (light green)
- ★ Digital Pins 2-13 (green)
- ★ Digital Pins 0-1/Serial In/Out - TX/RX (dark green)
- These pins cannot be used for digital i/o (digitalRead and digitalWrite) if you are also using serial communication (e.g. Serial.begin).
- ★ Reset Button - S1 (dark blue)
- ★ In-circuit Serial Programmer (blue-green)
- ★ Analog In Pins 0-5 (light blue)
- ★ Power and Ground Pins (power: orange, grounds: light orange)
- ★ External Power Supply In (9-12VDC) - X1 (pink)
- ★ Toggles External Power and USB Power (place jumper on two pins closest to desired supply) - SV1 (purple)
- ★ USB (used for uploading sketches to the board and for serial communication between the board and the computer; can be used to power the board) (yellow)

Daniel Pickem

Extending the Hardware

- Done with extension boards or shields
- Plug into an Arduino directly (most are made for Uno and Mega)
- Extend capabilities of the microcontroller
- Examples include
 - Ethernet shield
 - Wifi shield
 - GSM shield
 - Motor control shield



Daniel Pickem

Interfacing with an Arduino

- Serial Rx, Tx Line (UART)
- Serial Peripheral Interface (SPI)
- Inter-Integrated Circuit (I²C)
- External Interrupts (falling or rising edges)
- Pulse Width Modulation (PWM)
- Reset Pin


- More information available at:
<http://en.wikipedia.org/wiki/Arduino>

Daniel Pickem

interact.io
design
centre


IO to/from what?


<p><i>Sensors</i></p> <ul style="list-style-type: none"> ★ LDR / IR ★ Switch / Potentiometer / Joystick / Piezo ★ Accelerometer ★ Ultrasonic <p><i>Indicators</i></p> <ul style="list-style-type: none"> ★ LED / Lamps ★ Buzzers 	<p><i>Actuators</i></p> <ul style="list-style-type: none"> ★ Solenoid ★ Stepper Motor <p><i>Other Circuits</i></p> <ul style="list-style-type: none"> ★ Prototype shields ★ Special ICs
--	---

Daniel Pickem 

interact.io
design
centre

Arduino IO



Daniel Pickem 

interaction design centre

Arduino IO

Bluetooth - BlueSmirf
Internet - MatchPort
Many others:
Wifi, IrDa, Zigbee, etc.




Daniel Pickem




interaction design centre

Arduino IO

Motors:
DC, Steppers, Servos



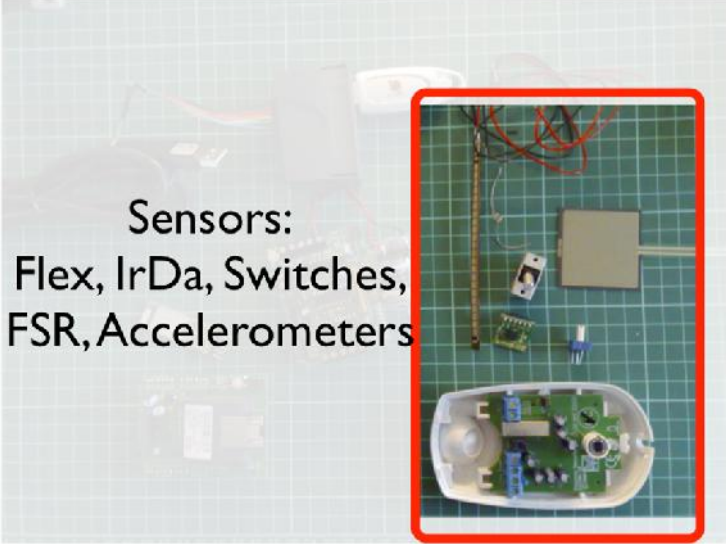
Daniel Pickem




interaction design centre

Arduino IO

Sensors:
Flex, IrDa, Switches,
FSR, Accelerometers




Daniel Pickem




interaction design centre

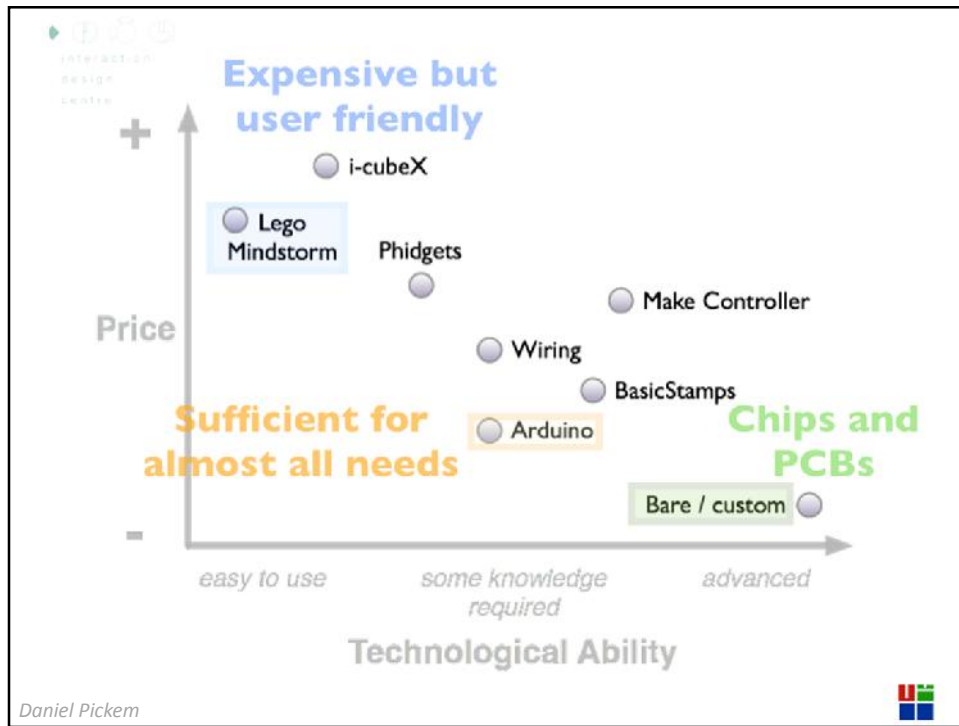
Arduino IO

Custom Hardware:
e.g. VMusic 2 MP3 player




Daniel Pickem






☆ **Lego Mindstorm NXT**




Approx. ~€250

☆ **Arduino**



Approx. ~€25

☆ **ATMega168**

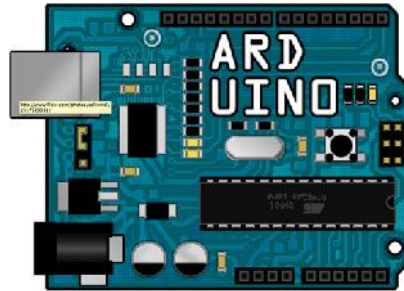


Approx. ~€4

interacti.on design centre

Daniel Pickem

Mechatronics (ME 6405) Student Lecture On Arduinos

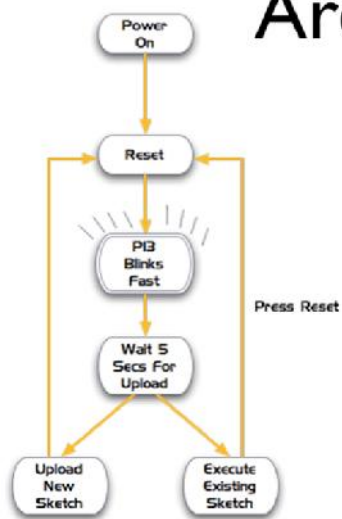


Daniel Pickem and **Rowland O'Flaherty**
12/04/2012

*Some slides courtesy of Eoin Brazil
<http://www.slideshare.net/eoinbrazil/imediarduino08#btnNext>



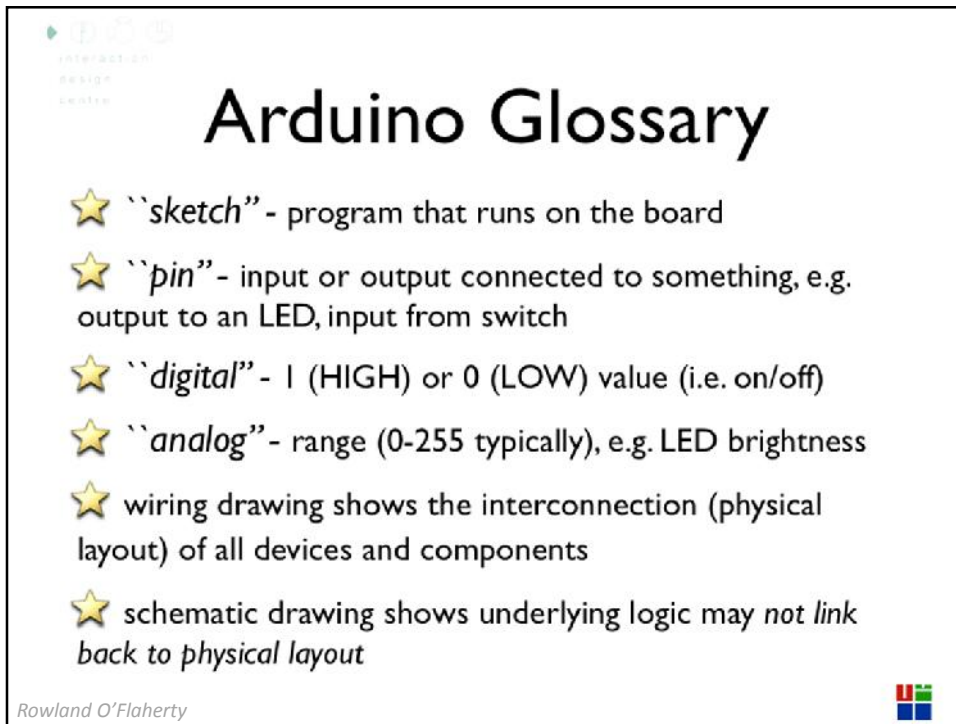
Programming an Arduino



- ☆ Write program
- ☆ Compile (check for errors)
- ☆ Reset board
- ☆ Upload to board

Rowland O'Flaherty




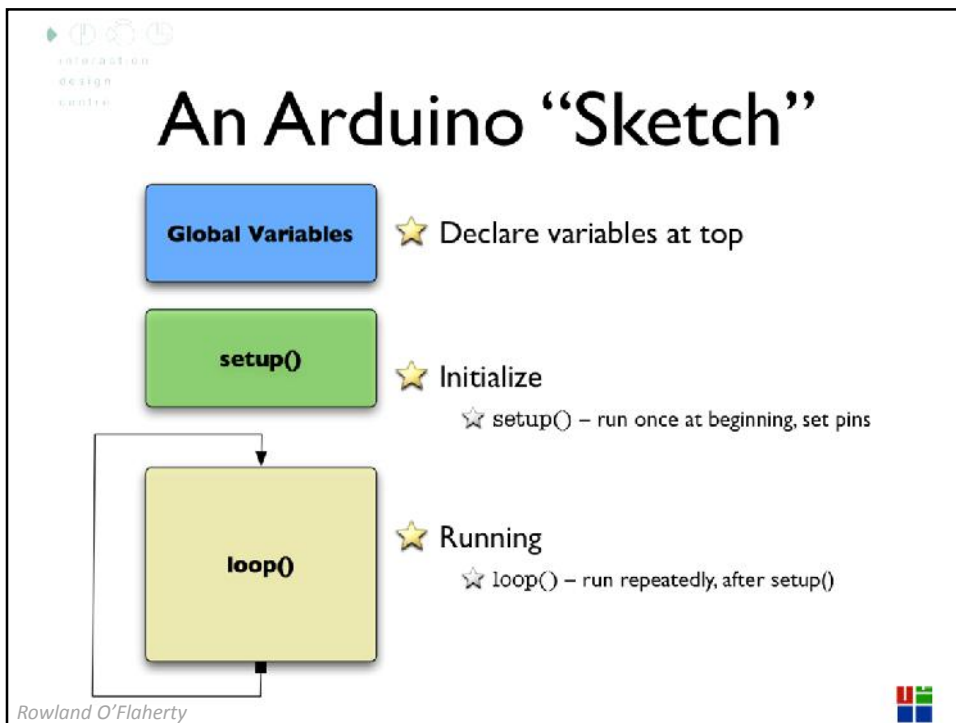


INTERACT.ORG
DESIGN
CENTRE

Arduino Glossary

- ★ ``sketch'' - program that runs on the board
- ★ ``pin'' - input or output connected to something, e.g. output to an LED, input from switch
- ★ ``digital'' - 1 (HIGH) or 0 (LOW) value (i.e. on/off)
- ★ ``analog'' - range (0-255 typically), e.g. LED brightness
- ★ wiring drawing shows the interconnection (physical layout) of all devices and components
- ★ schematic drawing shows underlying logic may *not link back to physical layout*

Rowland O'Flaherty 



INTERACT.ORG
DESIGN
CENTRE

An Arduino "Sketch"

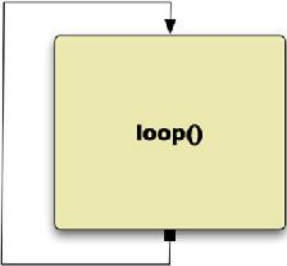
Global Variables

★ Declare variables at top


setup()

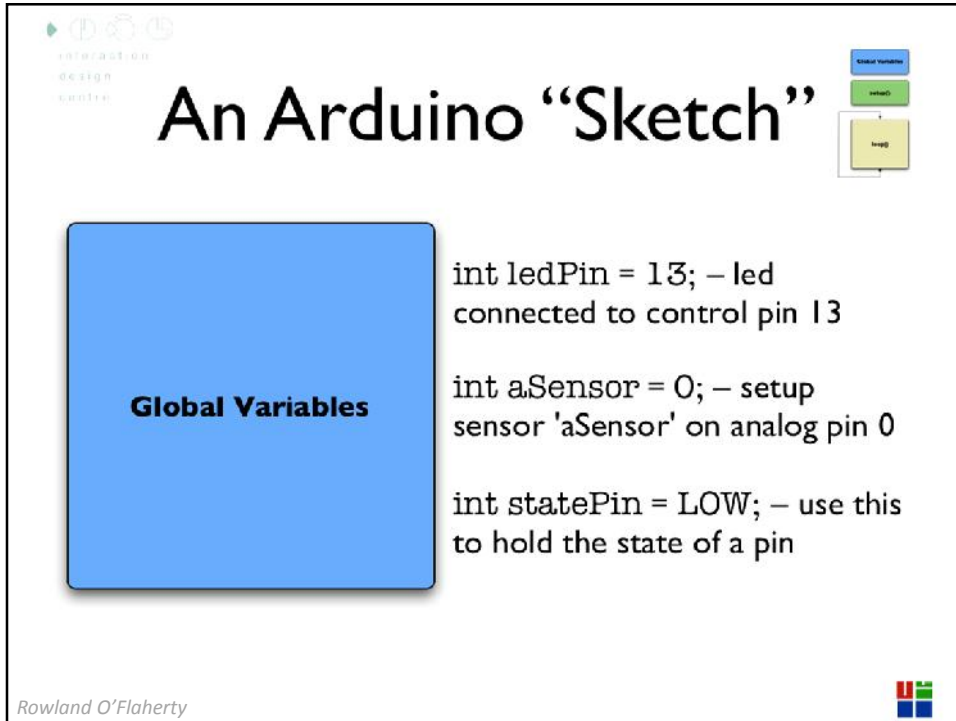
★ Initialize
 ☆ setup() – run once at beginning, set pins

loop()



★ Running
 ☆ loop() – run repeatedly, after setup()

Rowland O'Flaherty 



An Arduino "Sketch"

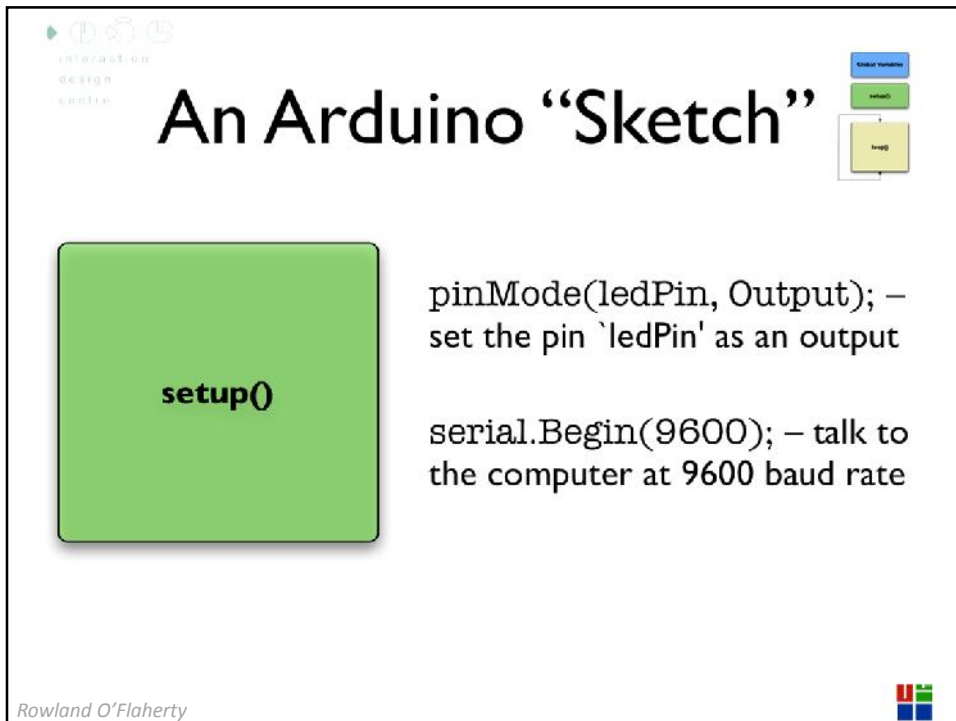
Global Variables

```
int ledPin = 13; - led
connected to control pin 13

int aSensor = 0; - setup
sensor 'aSensor' on analog pin 0

int statePin = LOW; - use this
to hold the state of a pin
```

Rowland O'Flaherty



An Arduino "Sketch"

setup()

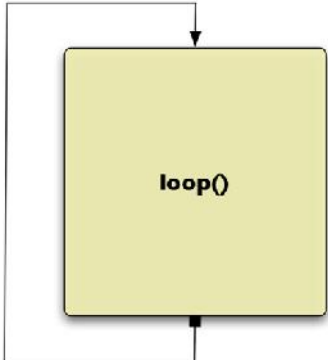
```
pinMode(ledPin, Output); -
set the pin `ledPin' as an output

serial.Begin(9600); - talk to
the computer at 9600 baud rate
```

Rowland O'Flaherty

interaction
design
centre

An Arduino "Sketch"



digitalWrite() – set a digital pin high/low
digitalRead() – read a digital pin's state
analogRead() – read an analog pin
analogWrite() – write an "analog" PWM value
delay() – wait an amount of time
millis() – get the current time

Rowland O'Flaherty

interaction
design
centre

```

/* Blinking LED ...
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, based on data coming over serial
 */

int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;


void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  Serial.begin(19200); // initiate serial communication
}

void loop()
{
  while (Serial.available() > 0) {
    inByte = Serial.read();
  }
  if (inByte > 0) {
    digitalWrite(ledPin, HIGH); // sets the LED on
  } else {
    digitalWrite(ledPin, LOW); // sets the LED off
  }
}

```

Initialise
some of the
variables

Rowland O'Flaherty



INTERACT.ORG
DESIGN
CENTRE

```

/* Blinking LED ...
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, based on data coming over serial
 */


int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;

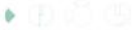
void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  Serial.begin(19200); // initiate serial communication
}

void loop()
{
  while (Serial.available()>0) {
    inByte = Serial.read();
  }
  if (inByte>0) {
    digitalWrite(ledPin, HIGH); // sets the LED on
  } else {
    digitalWrite(ledPin, LOW); // sets the LED off
  }
}

```

Setup LED pin and serial connection

Rowland O'Flaherty 



INTERACT.ORG
DESIGN
CENTRE

```

/* Blinking LED ...
 * turns on and off a light emitting diode(LED) connected to a digital
 * pin, based on data coming over serial
 */


int ledPin = 13; // LED connected to digital pin 13
int inByte = 0;

void setup()
{
  pinMode(ledPin, OUTPUT); // sets the digital pin as output
  Serial.begin(19200); // initiate serial communication
}

void loop()
{
  while (Serial.available()>0) {
    inByte = Serial.read();
  }
  if (inByte>0) {
    digitalWrite(ledPin, HIGH); // sets the LED on
  } else {
    digitalWrite(ledPin, LOW); // sets the LED off
  }
}

```

Loop - Reading the serial for info, when something is received turn the LED on

Rowland O'Flaherty 

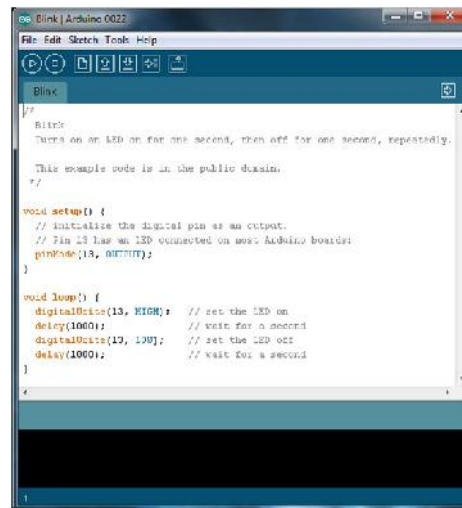
Extending the Software

- Libraries are used to extend software functionality (implemented in a superset of C/C++)
- Large variety of libraries available online as open source
- Examples
 - Communication: GSM, I²C, SPI libraries
 - Localization: GPS
 - Control: PWM, motor control libraries
 - Sensing: temperature, light, acceleration, ...
- Simple to implement your own and make available to IDE

Rowland O'Flaherty

The Arduino IDE

- Used to create sketches, the Arduino term for a program.
- Allows programming the Arduino via USB.
- Allows reflashing of the firmware on the microcontroller.
- Runs on all major operating systems.



```

Blink
//
// Blink:
// Turns on an LED on for one second, then off for one second, repeatedly.
//
// This example code is in the public domain.
//

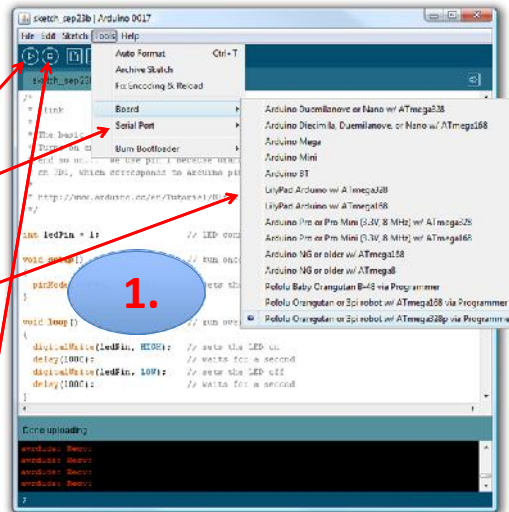
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
  
```

Rowland O'Flaherty

How to program an Arduino

1. Create your sketch in the Arduino IDE
2. Compile and debug your sketch
3. Select the correct serial port for uploading
4. Select the board corresponding to your hardware
5. Upload the sketch to your Arduino



DEMO

Distributors

- www.sparkfun.com
 - Lots and lots of shields, arduinos, and resources
 - Pricy, US-based
- www.dealextreme.com
 - Extremely cheap
 - Long shipping times, based in Hong Kong
- www.adafruit.com
 - US-based distributor
 - Nice selection of Arduino and related products
 - Reasonably priced

Rowland O'Flaherty

References and Resources

- Manufacturer and Introduction
 - <http://www.arduino.cc/en/>
- Projects and Tutorials
 - <http://www.bldr.org/>
- Lectures
 - <http://www.slideshare.net/eoinbrazil/imediardui-no08>

Rowland O'Flaherty

Questions?