

Arduino Development for Beginners

By Richard Coombs and
Przemyslaw Woznowski

Contents

- Introduction
- Board Types
- Shields
- Background
- Software
- Hardware
- Programming Language/IDE Setup
- Demo
- Wireless
- Arduino @ Cardiff Uni
- Resources

Our Experience

- What?
 - Beginners Arduino Workshop.
- Who?
 - tinker.it - the UK's leading provider of Arduino training.
- Aim?
 - a basic introduction to the Arduino platform for artists, designers and hobbyists.

Introduction

The image features a solid dark blue background. At the bottom, there is a white silhouette of a city skyline with various building shapes and heights. In the center of the image, the word "Introduction" is written in a large, white, sans-serif font.

What?

- Open-source electronics prototyping platform
- Hardware (circuit board)
 - Micro-controller
 - Memory to store code
 - Pins for input/output to/from the CPU
 - Connection for computer
- Software (IDE)
 - Written in Java
 - Designed for novice programmers

Why?

Created in Italy 2005 to be:

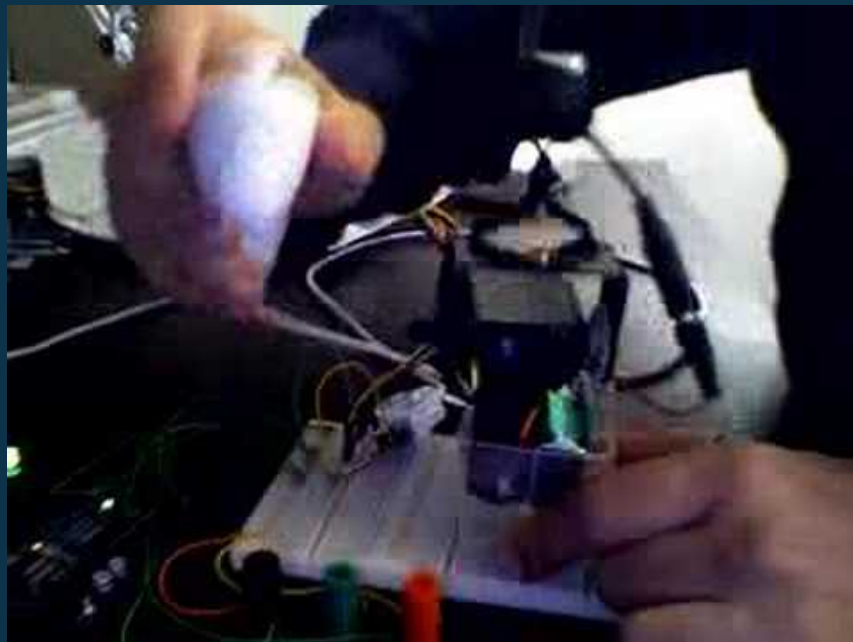
- Inexpensive
- Cross-platform
- Easy to use
- Extensible

Arduinos are for Tinkering!

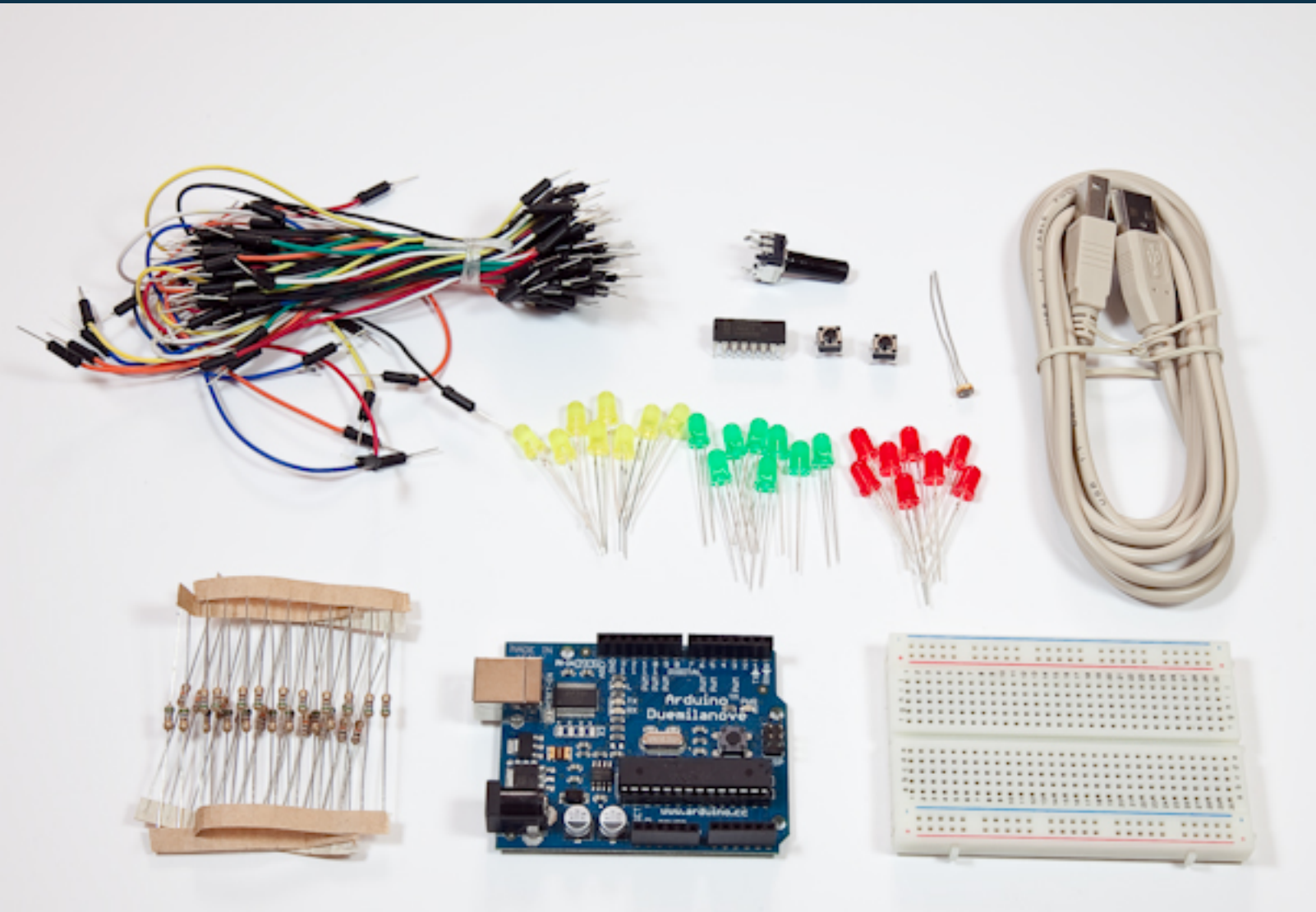
Tinkering

"Tinkering is what happens when you try something you don't quite know how to do, guided by whim, imagination and curiosity" [1]

- Patching
- Circuit bending
- Toy hacking
- Keyboard hacks

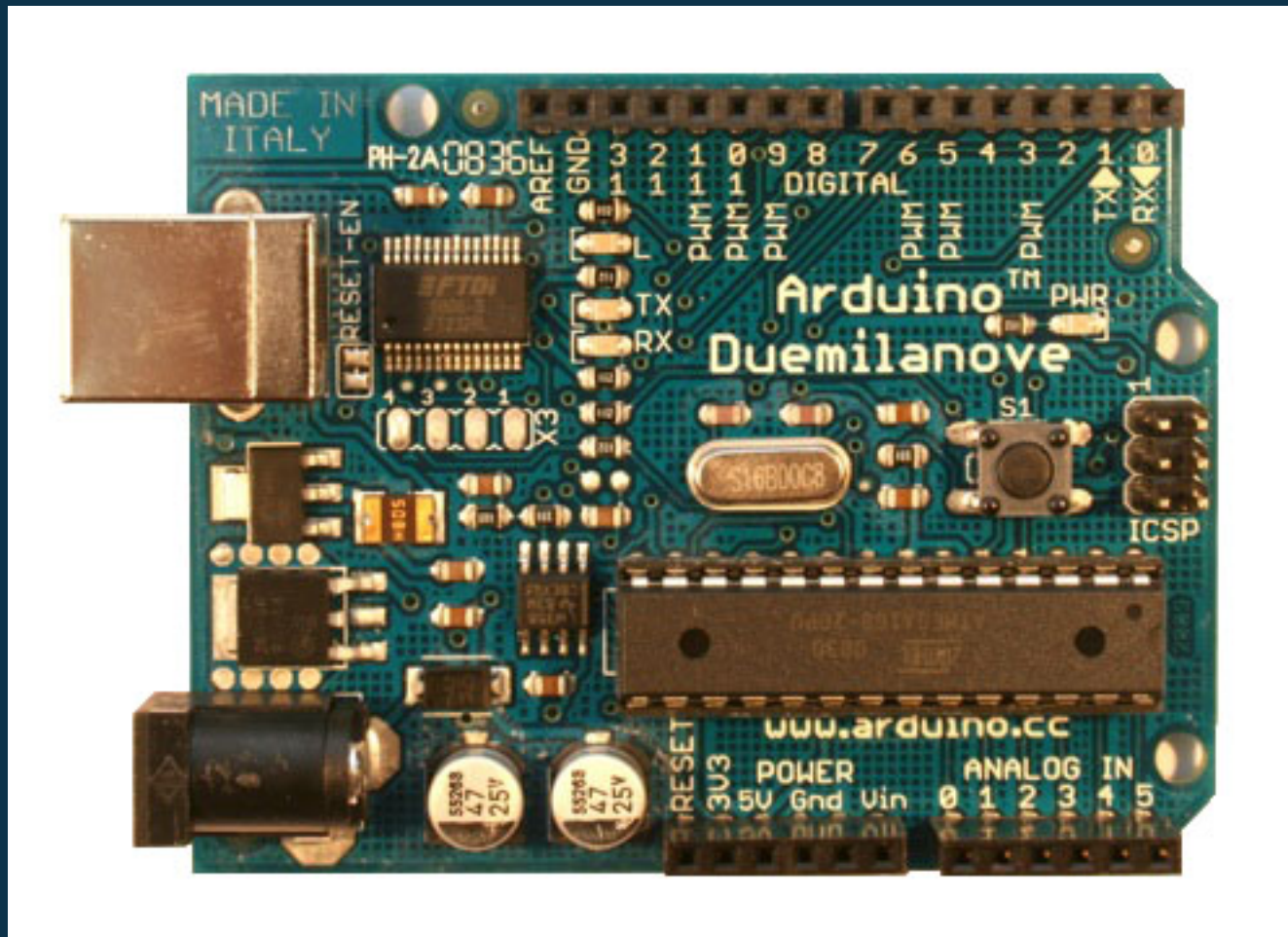


Starter Kit

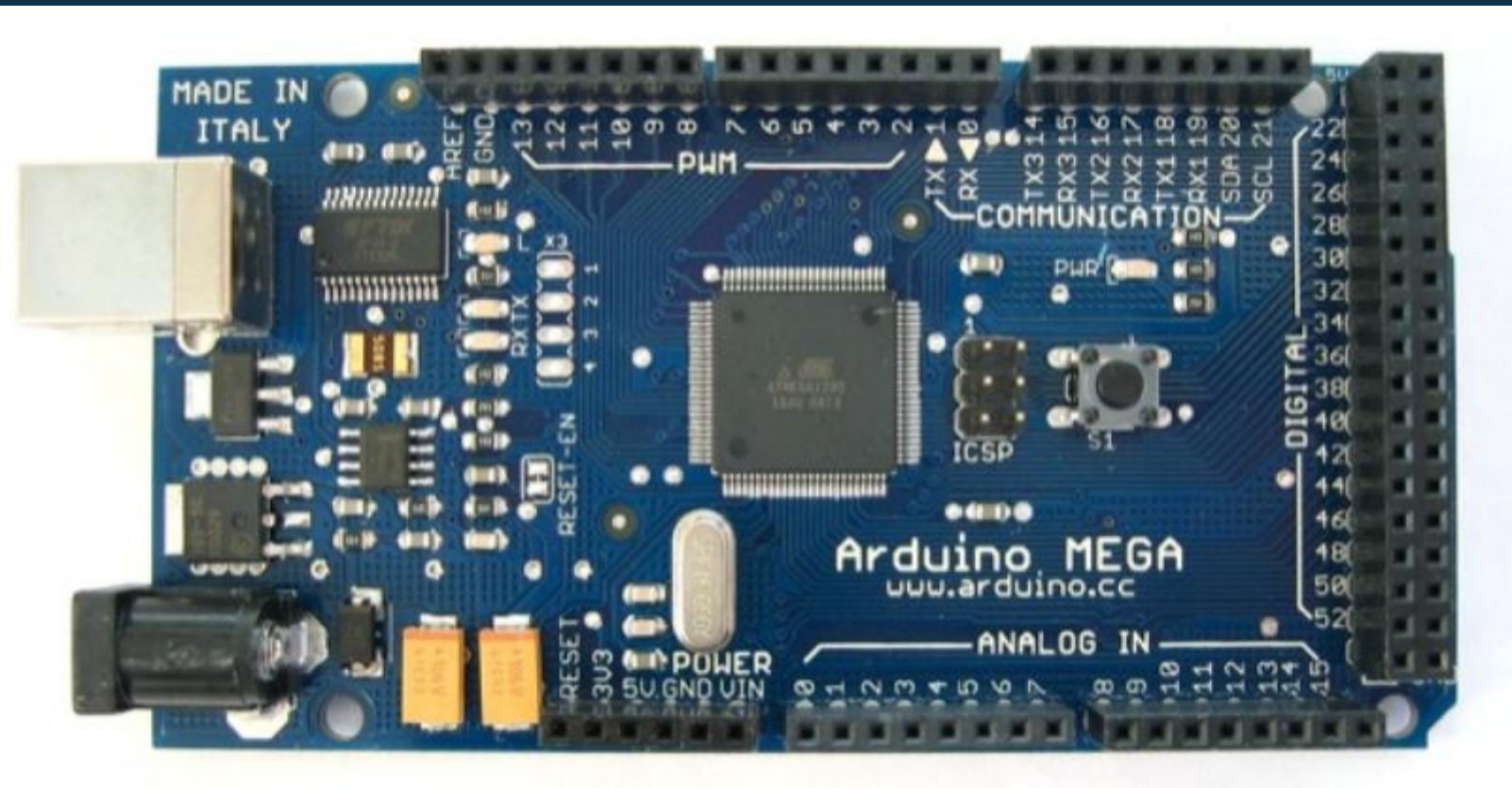


Board Types

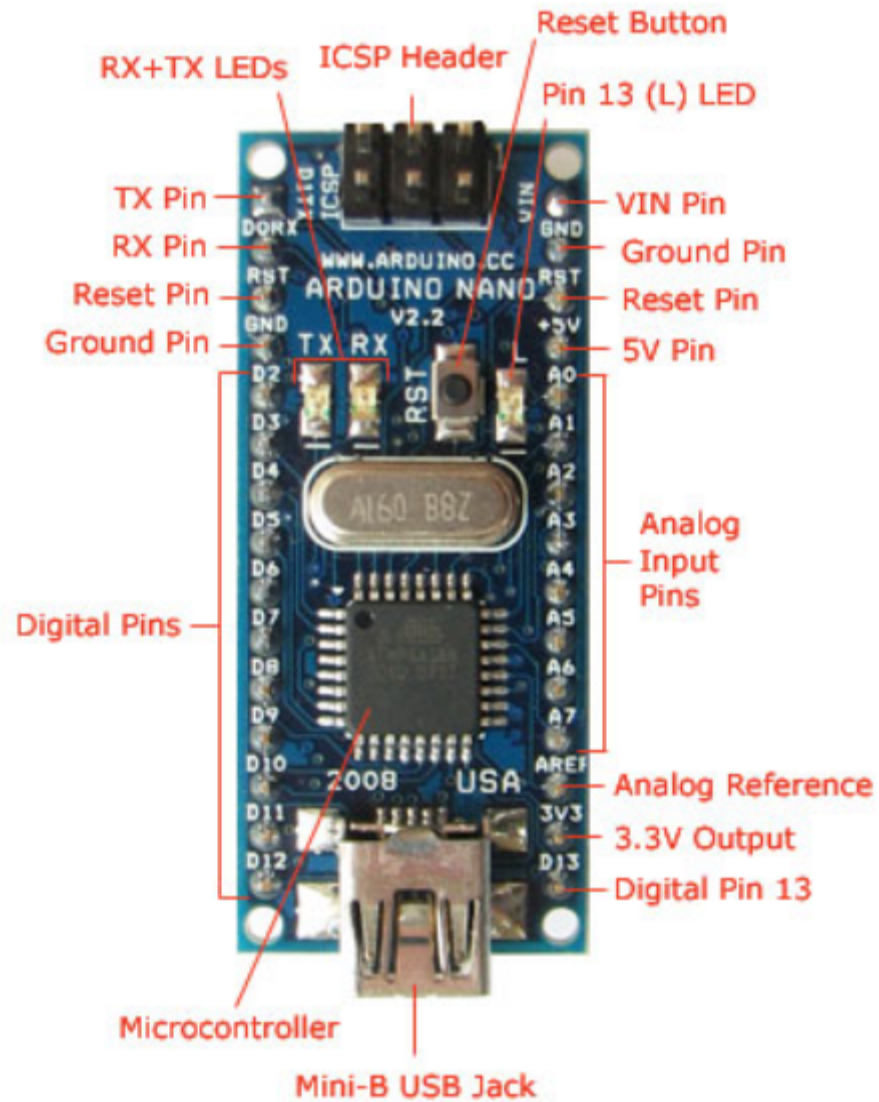
Duemilanove



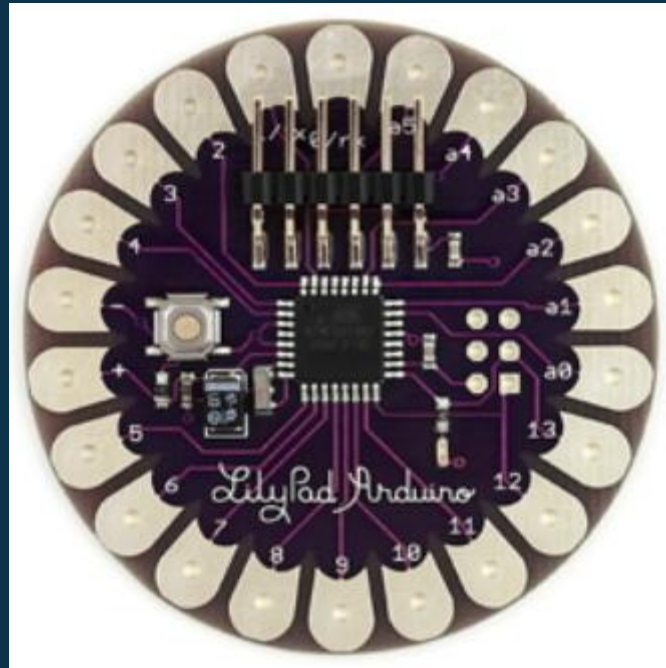
Mega



Nano

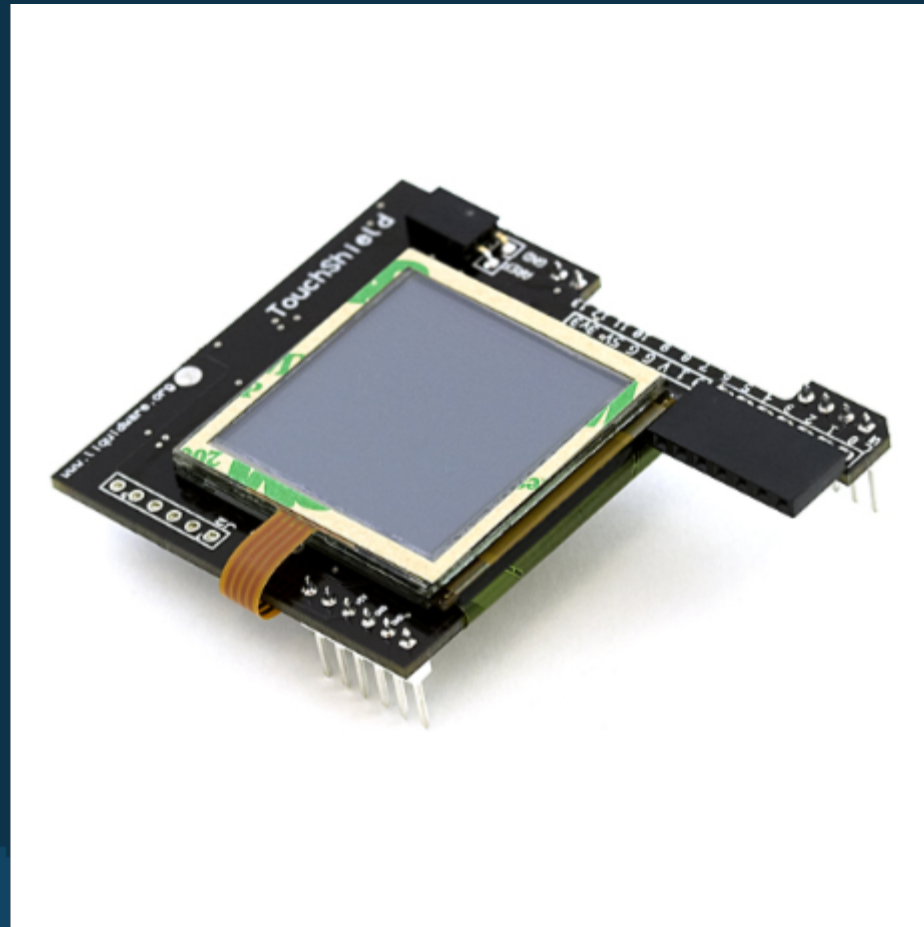


LillyPad



Shields

Boards which can be mounted on an Arduino to add extra functionality

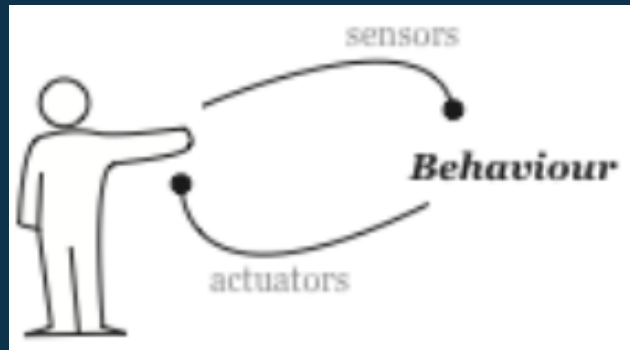


Background

- **sensor** - an electronic device used to measure a physical quantity such as temperature, pressure or loudness and convert it into an electronic signal of some kind (e.g a voltage). [2]
- **actuator** - a mechanism that puts something into automatic action. [3]

The Interactive Device

- Most of the objects build using Arduino follow, so called “Interactive Device” pattern.



- “It is an electronic circuit that is able to sense the environment using components called “sensors” and processing the information through “behaviour” implemented as software. The device will then be able to interact back with the world using “actuators”.”[1]

“Ingredients” needed

- Software/ Development Environment
- Hardware
- Program sketch
- Often patience

Software

- **Cross platform** - runs on Windows, Mac OS X and Linux
- Written in **Java** and based on **Processing** programming language, avr-gcc, and other open source software.
- **Arduino** – IDE designed to introduce programming to artists and other users unfamiliar with software development. Includes a code editor with brace matching, automatic indentation, API lookup, capable of compiling and uploading programs to the board with one click

Software cont.

- **Processing** - Processing is an open source programming language and environment for people who want to program images, animation, and interactions. It is used by students, artists, designers, researchers, and hobbyists for learning, prototyping, and production. It is created to teach fundamentals of computer programming within a visual context and to serve as a software sketchbook and professional production tool. [4]

Hardware

- Arduino board
- USB (standard A-B) cable
- Broken (or not) toys, keyboards, etc.
- Resistors, potentiometers, LEDs, buttons, etc.
- Wires, breadboards

Program Sketch

- Arduino IDE comes with a C/C++ library called “Wiring”, which makes input/output operations easier.
- Programs are written in C/C++ but user only has to define 2 functions to make a runnable program:
 - `setup()` - called once at the start of a program. This is where you initialize things.
 - `loop()` – called repeatedly until the board is powered off.

Programming Language

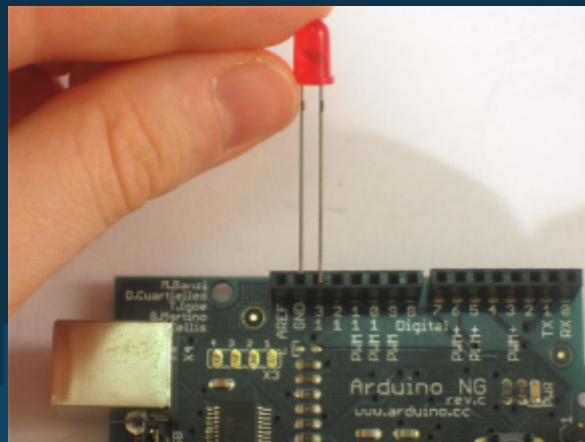
- Code written in Arduino is translated into C language.
- Passed on to the `avr-gcc` compiler that makes the translation into the language understood by the microcontroller.
- Uploaded to the board.

IDE Setup

- Download and install Arduino (<http://www.arduino.cc/en/Main/Software>) and Processing (<http://processing.org>).
- Connect the board to the USB port on your computer.
- Run Arduino IDE.
- Set the correct port your device is using
 - Tools -> Serial Port -> select port beginning with “/dev/cu.usbserial-“
- Set the Arduino board type your are using
 - Tools -> Board -> select correct board type (printed on the chip)

Hello World

- Open: **File -> Examples -> Digital -> Blink**
- Before uploading the code, you can verify it (**Sketch -> Verify/Compile**), however this is not compulsory, as if there are errors while sending program to the board, Arduino will abort the operation and print appropriate warnings/exceptions.
- Connect your LED to port 13 and GND (Ground)



LED

- **LEDs** are directional components. That means if you put it in backwards it will not work!
- To help you put the LED in right, the LED factory cuts the legs at **different lengths**.
- The **longer leg** goes in the hole marked 13 and the shorter one goes in the hole marked GND

Hello World cont.

- Make sure your board is connected via USB cable
- Click the “**Upload**” button or select **File -> Upload to I/O Board**
- Watch for any errors (if wrong board/port selected or bugs in the code)
- Hopefully it works – “**Let there be light**”

Button Demo

- Open: **File -> Examples -> Digital -> Button**
- Build the circuit as described in the file, or
- Look up the website address in this file, in order to see the picture of the circuit and its description: <http://www.arduino.cc/en/Tutorial/Button>
- Connect Arduino via USB
- Upload it to the board

IMPORTANT: Always disconnect the board while building the circuit!!!

Analog Input

- Open: **File -> Examples -> Analog -> Analog Input**
- Build the circuit as described in the file, or
- Look up the website address in this file, in order to see the picture of the circuit and its description: <http://arduino.cc/en/Tutorial/AnalogInput>
- Connect Arduino via USB
- Upload it to the board

Graph

- Open: **File -> Examples -> Communication -> Graph**
- Circuit previously built can remain the same
- Open Processing
- Copy and paste code commented out in Arduino to Processing
- Run the program from Processing

Simple Paint App

- In Arduino: Open: **File -> Examples -> Firmata -> StandardFirmata**
- Upload it to the board
- In Processing open and run test program: **File -> Sketchbook -> Libraries -> arduino -> examples -> arduino_input**
- In processing open and run: **File -> Sketchbook -> drawing_app**

Wireless

- Cardiff Sensor Network project (CaSeNet?)
- CaSeNet to use Bluetooth sensors
- Arduino is compatible with Pachube
- Possible to use Arduinos for sensor nodes?
- Arduino + Bluetooth shield
 - Bluetooth shield could not discover devices
- Arduino + ZigBee shield
 - Some progress made
 - Couldn't get it to work
 - Couldn't keep shields to continue work

Arduino @ Cardiff University

- Matthew Morgan is setting up a group for Arduino developers
- Held every fortnight
- Undergraduate/Postgraduate students/Lecturers/Staff
- Hopefully we can get some funding and buy couple of boards
- Aim: exchange experiences, share ideas, build something funky that can be used/displayed in our department
- **Interested?** On your way out put your name down on the Arduino Cafe list.

Resources

- Forum, tutorials, blog, useful links, software:
 - <http://arduino.cc>
- Shop:
 - <http://www.coolcomponents.co.uk>
 - <http://www.skpang.co.uk>
 - <http://www.robotbits.co.uk>
- Training:
 - <http://tinker.it>

Questions? Comments?
Feedback?

References

- [1]. Banzi, M. , Getting Started with Arduino 3rd release
- [2]. <http://encyclopedia2.thefreedictionary.com/sensor>
- [3]. <http://www.thefreedictionary.com/actuator>
- [4]. <http://processing.org>