Creative Programming.

Pablo Miranda Carranza, 2014-02-18

‘semantic aspects of communication are irrelevant to the engineering problem’.

Claude E. Shannon *A Mathematical Theory of Communication* , The Bell System Technical Journal, Vol. 27, July-October 1948, p. 379.

‘ the medium is the message’.

Marshall McLuhan, Understanding Media: the Extensions of Man,Routledge, London, 1964, p.7.

Some examples:

Processing and other “creative programming” frameworks.

Processing: http://processing.org, http://openprocessing.org

Arduino: http://arduino.cc

Open Frameworks: http://www.openframeworks.cc

Cinder: http://libcinder.org

Information Visualisation.

Infosthetics: http://infosthetics.com

Visual Complexity: http://www.visualcomplexity.com/vc/

Design.

JurgLehni: http://juerglehni.com/works/hektor/

Troika: http://troika.uk.com/works

Kram Weisshaar: http://www.kramweisshaar.com

For today:

We are going to download Arduino, download and install a plugin, and lear the basic interaction with Arduino.

1.

Download and set-up Arduino:
arduino.cc

Optionally download and install the Sublime plug-in for Arduino(called Arduino-like IDE) , through package control.

When installing Arduino, it will create a directory/folder for your sketches (usually something like: /Users/username/Arduino/).This is your Sketchbook location. You can see what folder this is in the Arduino application, in Preferences. This is important, because it is where Arduino sketches and libraries should go.

2.

Download and install the web socket server library by Per Ejeklint:
https://github.com/ejeklint/ArduinoWebsocketServer

If you download it as a zip file and unzip it, the resulting folder will be called WebsocketServer-master. Remove the -master bit (so it is called only WebsocketServer) and copy it into the "library" folder in your Sketchbook location (explained in 1.). Restart the Arduino application if it is open, so it can find the library.

3.

Download Servo Code
automatic.se/websocket_servo.zip
Download, unzip and copy the folder into your Sketchbook location.

4.

(optionally an easy version that controls an LED).
automatic.se/websocket_servo.zip

5.

In the Arduino application, open the "Websocket_Servo" project (it should be visible in the "file/Sketchbook" menu, if you have copied it to the right location) Compile and upload the project into the Arduino, which should be connected through the USB to your computer, by clicking the arrow on top of the screen (it says Upload...).

6.

Open the serial port monitor (set the baudrate to 57600) and listen to get the correct IP from the Arduino. If you are connecting many Arduinos to the same network, you should give them different mac addresses in the code. You will see this in the first lines of the Wevsocket_Servo file, where it says something like:" byte mac[] = { 0x52, 0x4F, 0x43, 0x4B, 0x45, 0x54 }; “ change one of the hexadecimal numbers…

7.

Connect the Servos:
Connect ground (brown or black cable) to Arduino’s ground.
Connect Control cables (white or yellow) to pins 7,8 or 9.
Connect Power cable (red)to 5V.

8.

Set the power supply(to 7.5 volts…) and connect it to the Arduino board.

9.

Open the control.html file (it is in the same folder as the Websocket_Servo project, but you can copy and place it anywhere) and input the IP you have got from the serial..

(Hopefully) Control the servos!

You can also use a copy of the html file in a server… for example
http://automatic.se/control.html

10.

You can now change the way you control the servos by writing programs in Javascript and using websockets to send control commands to the Arduino (check the code of the control.html file to do that), or you can of course just play around with the code in the Arduino. The basic principle is that you have javascript talking to the Arduino through websockets.