

TEI2017 Studio: The Body as a Source of Aesthetic Qualities for Design

## SOFTWARE SETUP

Please install the following software on your laptop, in preparation for the studio.

1. Arduino IDE <http://arduino.cc/en/Main/Software>

Note that you can use an older version.

1. FTDI Bus Serial Driver <http://www.ftdichip.com/Drivers/VCP.htm>
2. Mosquito Pattern Arduino sketch “Pattern\_Mosquito.ino”
3. Personal Pattern Arduino sketch “personal\_pattern.ino”

See screenshots over page ...



Buy

Software

Products

Learning

Forum

Support

Blog

LOG IN

SIGN UP

## Download the Arduino IDE



### ARDUINO 1.8.1

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app [Get](#)

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

Release Notes

Source Code

Checksums (sha512)

### ARDUINO SOFTWARE HOURLY BUILDS

LAST UPDATE  
9 March 2017 0:13:42 GMT

Download a preview of the incoming release with the most updated features and bugfixes.

[Windows](#)

[Mac OS X](#) (Mac OSX Lion or later)

[Linux 32 bit](#), [Linux 64 bit](#), [Linux ARM](#)

ARDUINO 1.0.6 / 1.5.x / 1.6.x

### PREVIOUS RELEASES

Download the [previous version of the current release](#), the classic [Arduino 1.0.x](#), or the [Arduino 1.5.x Beta version](#).

All the [Arduino 00xx versions](#) are also available for download. The Arduino IDE can be used on Windows, Linux (both 32 and 64 bits), and Mac OS X.

## Source Code

Active development of the Arduino software is hosted by GitHub. See the instructions for [building the code](#). Source code of Arduino is available [here](#).

- Home
- Products
- Drivers
  - VCP Drivers
  - D2XX Drivers
  - D3XX Drivers
- Firmware
- Support
- Android
- EVE
- MCU
- Sales Network
- Web Shop
- Newsletter
- Corporate
- Contact Us



Virtual COM Port Drivers

This page contains the VCP drivers currently available for FTDI devices.

For D2XX Direct drivers, please click [here](#).

Installation guides are available from the [Installation Guides](#) page of the [Documents](#) section of this site for selected operating systems.

VCP Drivers

Virtual COM port (VCP) drivers cause the USB device to appear as an additional COM port available to the PC. Application software can access the USB device in the same way as it would access a standard COM port.

This software is provided by Future Technology Devices International Limited "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall future technology devices international limited be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

FTDI drivers may be used only in conjunction with products based on FTDI parts.

FTDI drivers may be distributed in any form as long as license information is not modified.

If a custom vendor ID and/or product ID or description string are used, it is the responsibility of the product manufacturer to maintain any changes and subsequent WHCK re-certification as a result of making these changes.

For more detail on FTDI Chip Driver licence terms, please [click here](#).

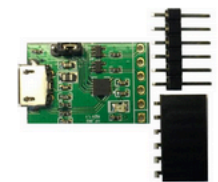
Currently Supported VCP Drivers:

Operating System	Release Date	Processor Architecture							Comments
		x86 (32-bit)	x64 (64-bit)	PPC	ARM	MIPSII	MIPSIIV	SH4	
Windows*	2016-10-10	2.12.24	2.12.24	-	-	-	-	-	WHQL Certified. Includes VCP and D2XX. Available as a <a href="#">setup executable</a> Please read the <a href="#">Release Notes</a> and <a href="#">Installation Guides</a> .
Linux	2009-05-14	1.5.0	1.5.0	-	-	-	-	-	1.0-19 Linux
Mac OS X 10.3 to 10.8	2012-08-10	2.2.18	2.2.18	2.2.18	-	-	-	-	10.0-10.9 Mac OS
Mac OS X 10.9 and above	2015-04-15	-	2.3	-	-	-	-	-	This driver is signed by Apple
Windows CE 4.2-5.2**	2012-01-06	1.1.0.20	-	-	1.1.0.20	1.1.0.10	1.1.0.10	1.1.0.10	
Windows CE 6.0/7.0	2016-11-03	1.1.0.22 CE 6.0 CAT CE 7.0 CAT	-	-	1.1.0.22 CE 6.0 CAT CE 7.0 CAT	1.1.0.10	1.1.0.10	1.1.0.10	For use of the CAT files supplied for ARM and x86 builds refer to <a href="#">AN_319</a>

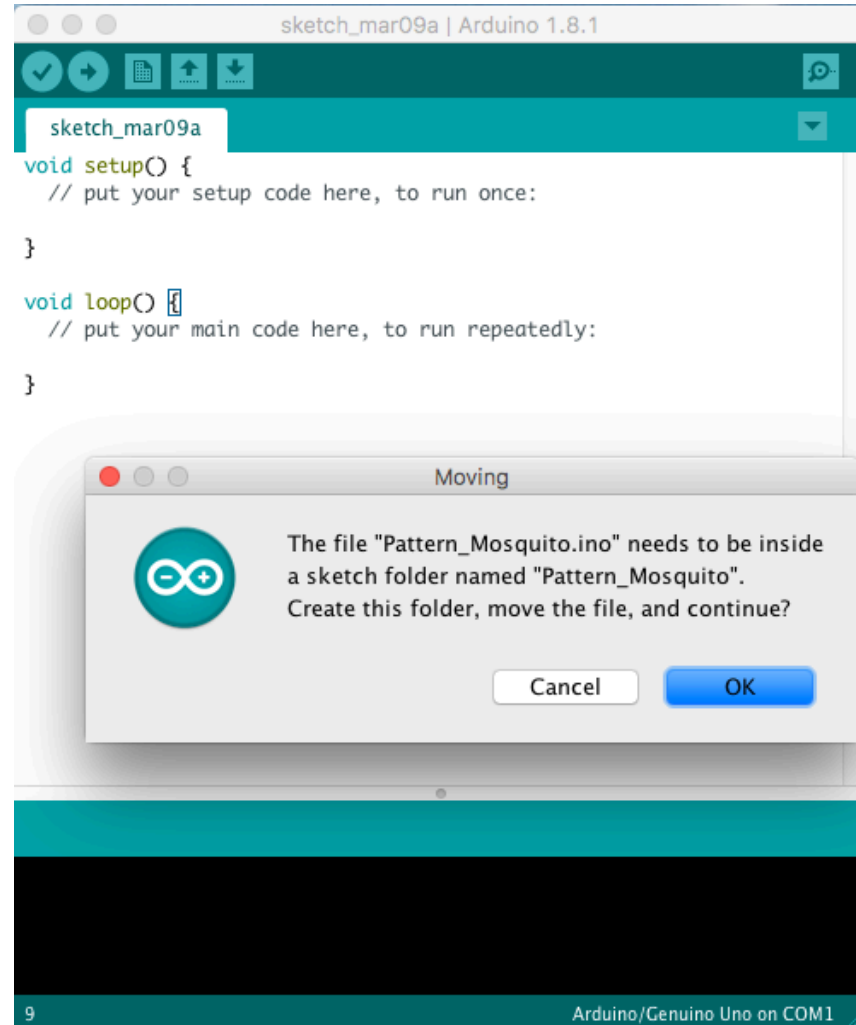
Select the one for your operating system/processor

Google

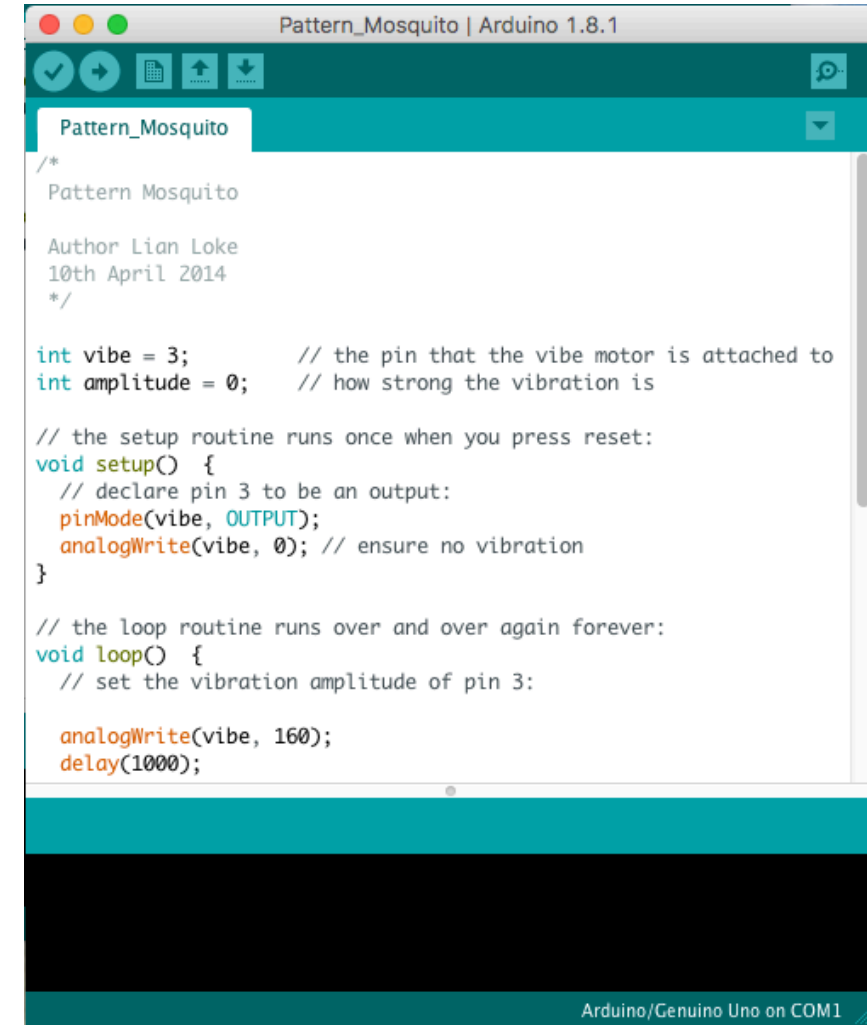
Sign up for your FREE LC234X module [here](#)



Open the Arduino software.  
Then select File > Open ...  
Select the file "Pattern\_Mosquito.ino",  
and click OK.



The code for "Pattern\_Mosquito.ino" sketch

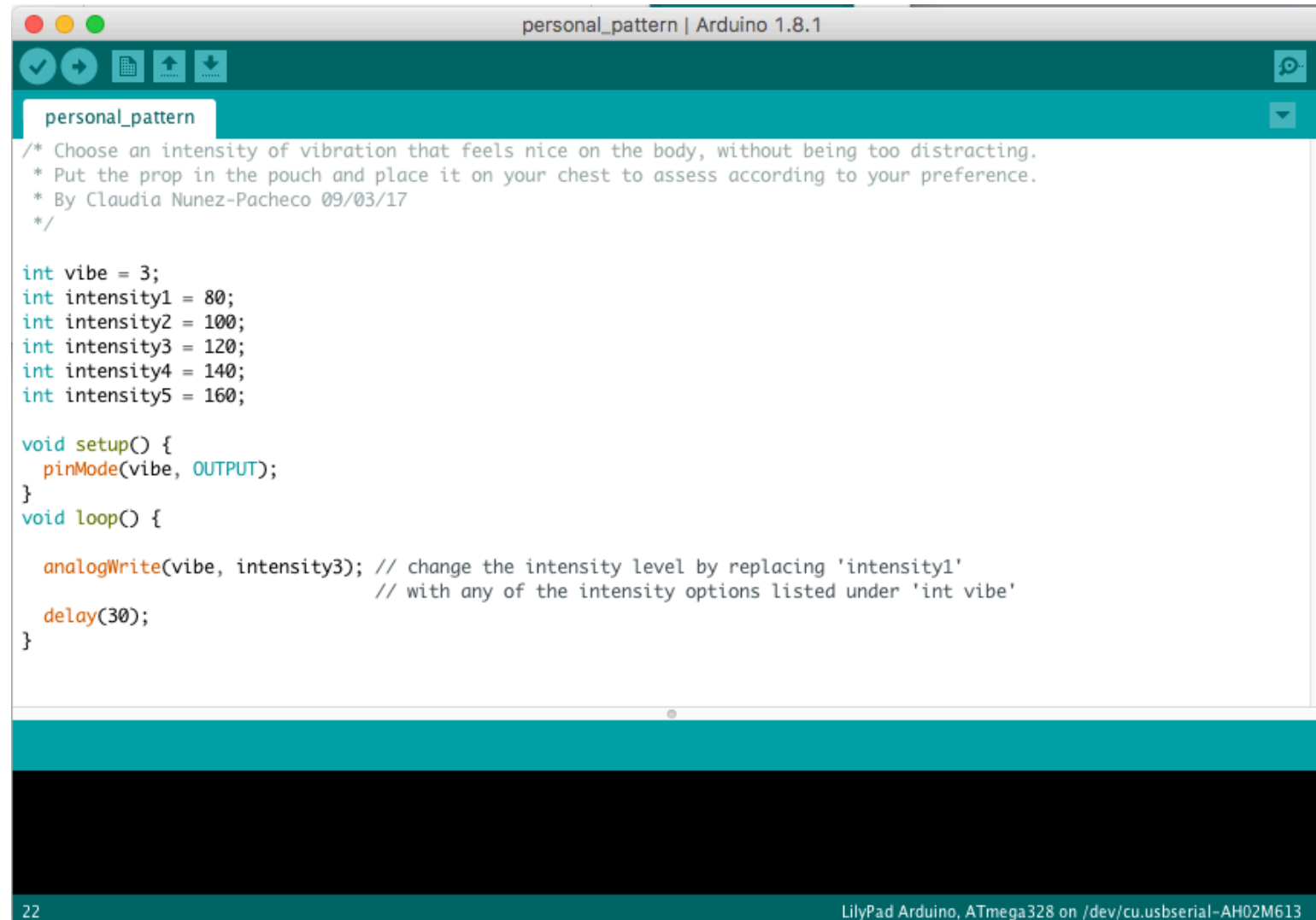


Open the Arduino software.

Then select File > Open ...

Select the file “personal\_pattern.ino”, and click OK.

The code for “personal\_pattern.ino” sketch

A screenshot of the Arduino IDE interface. The window title is 'personal\_pattern | Arduino 1.8.1'. The top toolbar shows icons for checking, running, and saving. The main editor area displays the code for the 'personal\_pattern' sketch. The code includes a multi-line comment at the top, followed by variable declarations for 'vibe' and five 'intensity' values. It then defines 'setup()' to set the pin mode and 'loop()' to write an analog value and delay. The status bar at the bottom shows '22' and 'LilyPad Arduino, ATmega328 on /dev/cu.usbserial-AH02M613'.

personal\_pattern

```
/* Choose an intensity of vibration that feels nice on the body, without being too distracting.  
 * Put the prop in the pouch and place it on your chest to assess according to your preference.  
 * By Claudia Nunez-Pacheco 09/03/17  
 */
```

```
int vibe = 3;  
int intensity1 = 80;  
int intensity2 = 100;  
int intensity3 = 120;  
int intensity4 = 140;  
int intensity5 = 160;
```

```
void setup() {  
  pinMode(vibe, OUTPUT);  
}
```

```
void loop() {  
  
  analogWrite(vibe, intensity3); // change the intensity level by replacing 'intensity1'  
                                // with any of the intensity options listed under 'int vibe'  
  delay(30);  
}
```