

# Arduino IDE Documentation



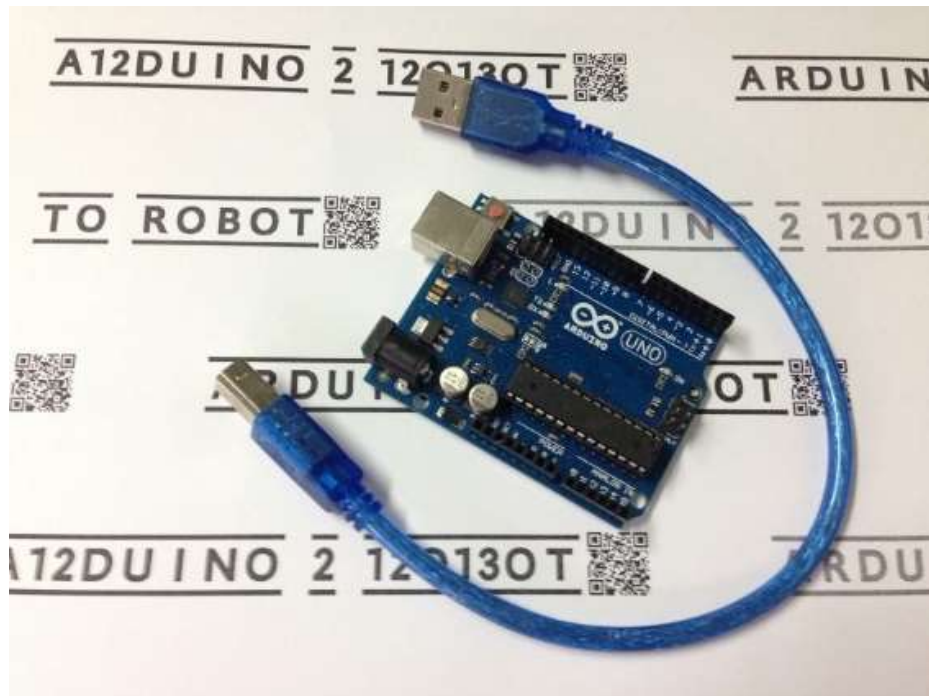
## สมาชิกกลุ่ม

- |    |                     |           |            |
|----|---------------------|-----------|------------|
| 1. | 61015004 ว่าที่ ร.ต | กฤษณะ     | วิป็นเขตร์ |
| 2. | 63015049 นาย        | ณัฐพล     | สุขสมรูป   |
| 3. | 63015097 นาย        | นวัต      | การสำเร็จ  |
| 4. | 63015108 นาย        | ปรีทัศน์  | วิลัยำ     |
| 5. | 63015121 นาย        | พศิน      | จันทร์ทัน  |
| 6. | 63015208 นาย        | อภิสิทธิ์ | ทับแสง     |

Github source code: [Arduino/Arduino: open-source electronics platform \(github.com\)](https://github.com/Arduino/Arduino)

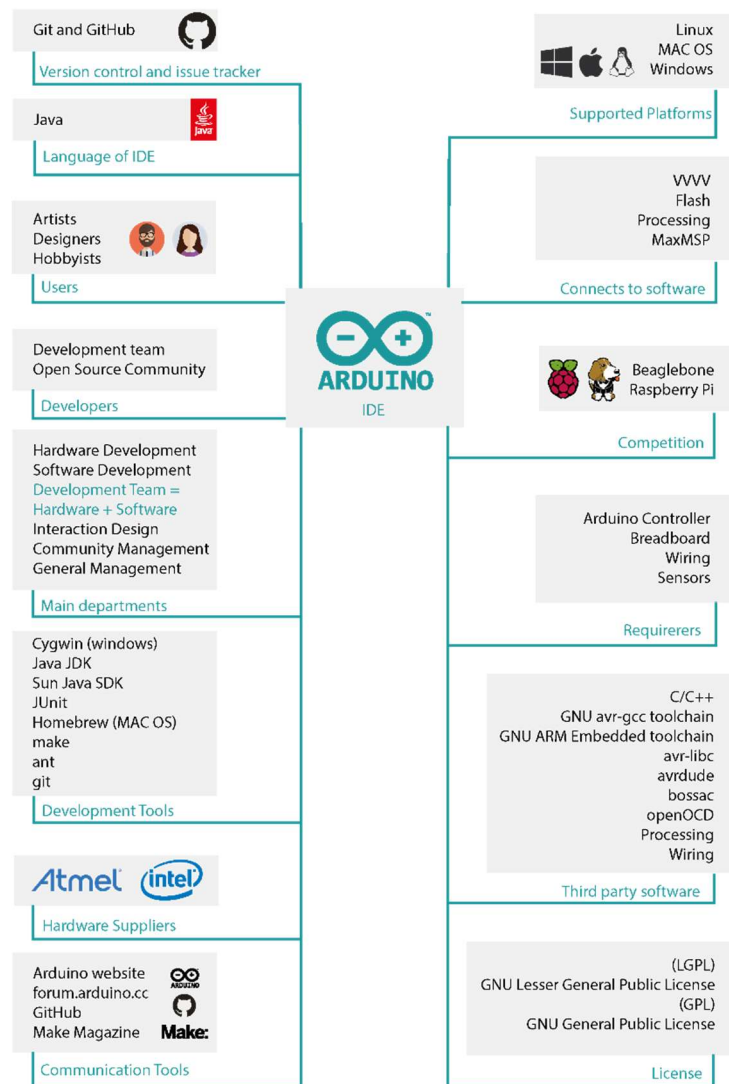
## Arduino คืออะไร?

Arduino คือ ไมโครคอนโทรลเลอร์ (Microcontroller) ชนิดหนึ่ง ซึ่งหมายถึง อุปกรณ์ที่ใช้สำหรับควบคุม หรืออ่านค่าบางสิ่ง ถ้าให้เปรียบเทียบมันก็คือคอมพิวเตอร์ขนาดเล็กนั่นเอง แต่มีราคาที่ถูกกว่าคอมพิวเตอร์มาก ซึ่งคำว่า Arduino นั้นไม่ใช่ชื่อของไมโครชิพ (Microchip) เป็นแค่เพียงชื่อแบรนด์ที่ออกแบบรูปทรงและเพิ่มอุปกรณ์เสริมเข้าไปบนบอร์ดเพื่อให้เราใช้งานได้ง่ายขึ้น เช่น ช่องสำหรับเสียบ USBช่องเสียบสายสัญญาณ ชุดแปลงไฟฟ้าก่อนเข้าไปเลี้ยงบอร์ด เป็นต้น



(อ้างอิงจาก [Arduino คืออะไร](#) )

## Arduino IDE คืออะไร?



Context view: [Arduino · Delft Students on Software Architecture: DESOSA 2017 \(gitbooks.io\)](#)

IDE ย่อมาจาก (Integrated Development Environment) คือ ส่วนเสริมของระบบการพัฒนา หรือตัวช่วยต่าง ๆ ที่จะคอยช่วยเหลือ Developer หรือช่วยเหลือคนที่พัฒนา Application เพื่อเสริมให้เกิดความรวดเร็ว ถูกต้อง แม่นยำ ตรวจสอบระบบที่จัดทำได้ ทำให้การพัฒนางานต่าง ๆ เร็วมากขึ้น ซึ่ง Arduino IDE เป็นโปรแกรม Open source ทำหน้าที่ ติดต่อระหว่างคอมพิวเตอร์ ไม่ว่าจะเป็นระบบ Windows, Mac OS X หรือ Linux กับ บอร์ด Arduino ซึ่งโปรแกรมนี้ออกแบบให้ง่ายต่อการเขียนโค้ด และ อัปโหลดโปรแกรมที่เราเขียนเข้าสู่บอร์ด Arduino

## Module structure and organization

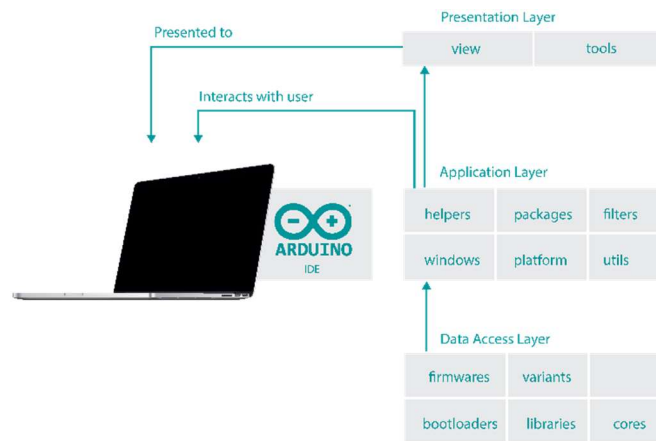
โดยตัว file ต่างๆ จำนวนมากและมีความซับซ้อนของตัว Code ที่ค่อนข้างมาก ทาง team Developer เองก็ได้ทำการจัดระเบียบส่วนต่างๆ ไว้เป็น Module เพื่อใช้งานง่ายต่อการเข้าถึงไว้ดังนี้

Module	หน้าที่
app	เก็บในส่วนของ Graphical User-Interface (GUI)
arduino-core	เก็บในส่วนของ ไฟล์หลักต่างๆ ของ IDE
build	ใช้ในการ building ตัว Arduino IDE เพื่อ test
hardware	Hardware ต่างๆที่ IDE รองรับ
libraries	เก็บพวก standard libraries ต่างๆที่ต้องใช้งาน

## Architecture ของ Arduino IDE

Software Architecture Styles ที่มีลักษณะตรงกับ Arduino IDE คือ

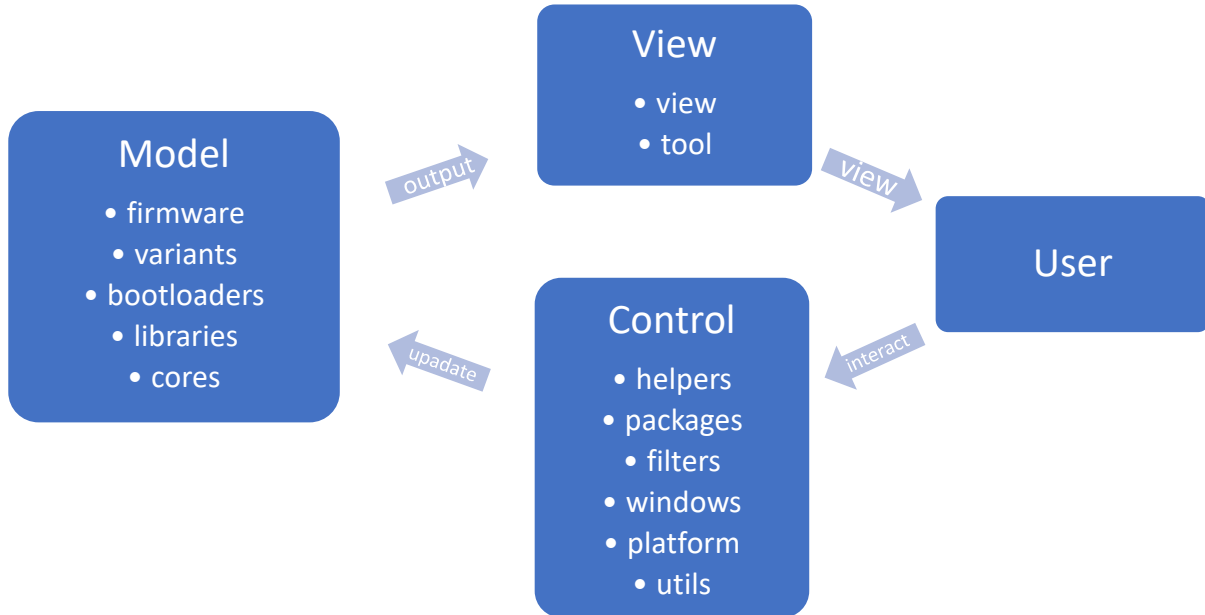
### Layer



ถ้าเป็นทางตัว UML class Diagram อาจจะมองเห็น Software Architecture Styles ได้ยากจึงได้ทำแบบ module view ออกมาก็สามารถเห็นได้เลยว่ามีการทำงานที่เป็นลำดับชั้น เริ่มจากการทำ data access layer ที่เป็นส่วนของการเข้าถึงทรัพยากรต่างๆ แล้วส่งต่อการทำงานไปให้ applications layer ที่เป็นส่วนของการประมวลผลหลักที่มี interacts กับผู้ใช้ จากนั้นก็ส่งให้กับ Presentation layer ทำการแสดงผลกับผู้ใช้

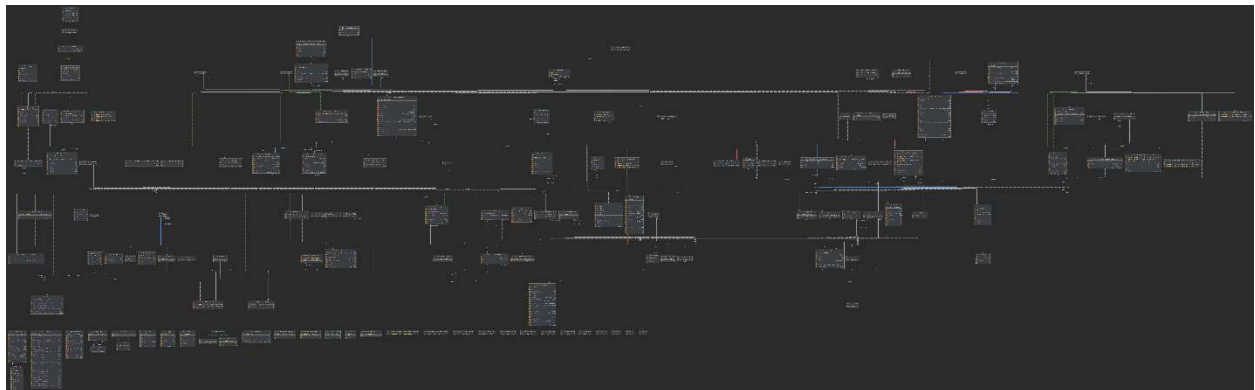
แต่ ก็มีความเป็นไปได้อีกส่วนหนึ่งคือ Software Architecture Styles แบบ **MVC**

### **MVC (Model View Control)**



โดยเดิมนั้น MVC เองก็เหมาะกับการทำ Software ที่ต้องมี Interacts ต่างๆ กับทางผู้ใช้อยู่แล้ว เพราะเช่นนั้นจึงคิดว่า MVC ก็อาจจะเป็น 1 ใน Software Architecture Style ของ software นี้เช่นกัน โดยทาง Model เราก็จะให้เป็นส่วนของการเข้าถึงทรัพยากรต่างๆ โดยถูกส่งมาโดยฝั่ง Control ที่ interacts กับผู้ใช้อยู่ และแสดงผลการทำงานต่างๆ ผ่าน View

### **UML ของ Arduino IDE : [ARCHSTYLE](#)**



Full Ver. : <https://drive.google.com/drive/folders/1jydK1zOL0cX2TNmblo78tsF8g3a-HwT1?usp=sharing>

## Quality Attributes ของ Arduino IDE


### 1. Portability

เนื่องจาก Arduino ได้มีการทำการ Compile ตัวโปรเจก Arduino ไว้รองรับการทำงานแบบ 32-bit และ 64-bit ดังนั้น Arduino IDE จึงสามารถนำไปทำงานทั้งระบบ Windows, OS X และระบบปฏิบัติการ Linux หลาย ๆ ตัวที่ได้รับความนิยมได้ ทั้งสถาปัตยกรรม CPU อื่นเช่น ARM อีกด้วย นอกจากนี้ ยังรองรับการใช้กับ Arduino board ทุกรุ่นบอร์ด

#### Tactic for achieve Portability

Arduino IDE ได้มีการพัฒนาโดยใช้ภาษา Java ซึ่งเป็นภาษาที่รองรับสำหรับ Compilers ที่สามารถรันได้บน Platforms ต่างๆมากมายที่มีการรองรับตัวภาษานี้อยู่

#### Proof of Evidence



**Arduino IDE 1.8.19**

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

Refer to the [Getting Started](#) page for Installation instructions.

**SOURCE CODE**

Active development of the Arduino software is [hosted by GitHub](#). See the instructions for [building the code](#). Latest release source code archives are available [here](#). The archives are PGP-signed so they can be verified using [this](#) gpg key.

**DOWNLOAD OPTIONS**

**Windows** Win 7 and newer  
**Windows** ZIP file

**Windows app** Win 8.1 or 10 [Get](#)

**Linux** 32 bits  
**Linux** 64 bits  
**Linux** ARM 32 bits  
**Linux** ARM 64 bits

**Mac OS X** 10.10 or newer

[Release Notes](#)  
[Checksums \(sha512\)](#)

Ref : [Software | Arduino](#)

[Arduino IDE 1 | Arduino Documentation | Arduino Documentation](#)

[Arduino/Arduino: open-source electronics platform \(github.com\)](#)

## 2. Modifiability

เนื่องจาก Arduino IDE นั้นได้มีการรองรับ Third-Party Hardware ที่ไม่ใช่จากทาง Arduino เอง ซึ่งทำให้ต้องมีการ ออกแบบ ให้ตัว IDE รองรับ libraries อื่นที่ไม่ใช่จากฝั่งของ Arduino เอง และอีกส่วนหนึ่งคือเรื่องของการใช้ custom theme ตัว IDE นี้สามารถนำเข้า custom theme จากผู้อื่นที่สร้างขึ้นมาได้อีกด้วย

### Tactic for achieve modifiability

ทาง Arduino ได้มีการทำข้อเสนอแนะในการ add library ที่เป็น Third-Party ให้ทำผู้ใช้ที่ต้องการ และได้มีการแจก document platform-specification ที่มีไว้เพื่อให้ผู้ที่ต้องการ พัฒนา Third-Party library นั้นทราบถึงข้อมูลจำเพาะบางอย่างหรือข้อมูลที่จำเป็นต่อการพัฒนาเพื่อให้สามารถนำมาใช้ร่วมกับ IDE ของทาง Arduino ได้ง่ายมากขึ้น

### Proof of Evidence

#### Third-Party Hardware

Support for third-party hardware can be added to the **hardware** directory of your sketchbook directory. Platforms installed there may include board definitions (which appear in the board menu), core libraries, bootloaders, and programmer definitions. To install, create the **hardware** directory, then unzip the third-party platform into its own sub-directory. (Don't use "arduino" as the sub-directory name or you'll override the built-in Arduino platform.) To uninstall, simply delete its directory.

For details on creating packages for third-party hardware, see the Arduino Platform specification.

Ref : [Overview of the Arduino IDE 1 | Arduino Documentation](#) | [Arduino Documentation](#)

[Platform specification - Arduino CLI](#)

[Use a custom theme for Arduino IDE 1.8 – Arduino Help Center](#)

### 3. Usability

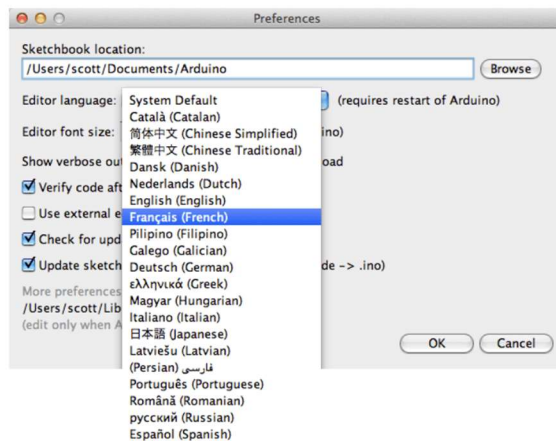
Arduino IDE มีการใช้งานที่แพร่หลายในหลากหลายพื้นที่ และหลากหลายระดับ สิ่งหนึ่งที่ทำให้ Arduino IDE เป็นที่นิยมมากคือ ตัว UI ที่สามารถเข้าใจได้ง่ายและสามารถใช้ custom theme ในการปรับแต่งรูปแบบให้สะดวกกับตัวผู้ใช้งานมากที่สุด และ Tool ต่างๆหรือ Function ต่างๆ ก็มีการทำ Document ออกมาอธิบายให้ผู้ใช้งานได้รับข้อมูลการทำงานเป็นจำนวนมาก และฝั่งผู้ใช้อเองก็สามารถปรับเปลี่ยน Library ของตัว board ในการ compile ได้เองอย่างง่ายดาย

### Tactic for achieve usability

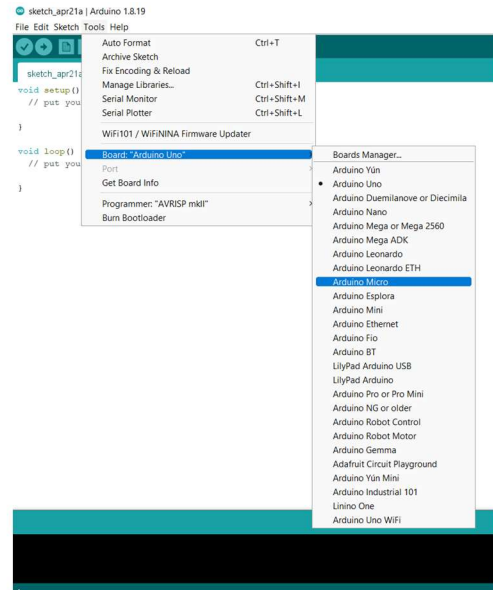
ทาง Arduino ได้มีการออกแบบ หน้า UI โดยผู้พัฒนาจำนวนมากซึ่ง ต่างเคยใช้ IDE อื่นๆ มา ก่อน จึงถือได้ว่าเป็น User เช่นกันจึงออกแบบ UI ให้เหมาะสมที่สุด ง่ายต่อการใช้งาน รวมไปถึงการ มีภาษาต่างๆให้เลือก(ยังไม่มีภาษาไทย)

## Proof of Evidence

## Language Support



Language preferences.



Ref: [Overview of the Arduino IDE 1](#) | [Arduino Documentation](#) | [Arduino Documentation](#)

[Arduino/Arduino: open-source electronics platform \(github.com\)](https://github.com/arduino/Arduino)

[Use a custom theme for Arduino IDE 1.8 – Arduino Help Center](#)



#### 4. Interoperability

ตัว Arduino IDE เองนั้นก็มีอีกความสามารถหนึ่งนั่นก็คือการที่สามารถเชื่อมต่อกับ Software อื่นภายนอกได้ (Third party software) เช่น Flash, VVVV, Processing หรือ Max/MSP ที่จะถูกเรียกใช้ บางครั้งในตอนที่อยู่ในช่วงเวลา run-time

##### **Tactic for achieve interoperability**

การที่ตัว Arduino IDE เองนั้นเป็น Open source จึงทำให้มีผู้พัฒนาที่มีความเชี่ยวชาญจาก หลากหลายด้าน บางส่วนเล็งเห็นตัวประโยชน์ของการทำการเชื่อมต่อกับ Third party software และรับ ฟัง feedback จากผู้ใช้อย่างหลายคนเพื่อสนองต่อความต้องการของผู้ใช้ให้ได้มากที่สุด

##### **Proof of Evidence**

###### **Development Tools**

When you are developing software for Arduino you can make your life easier by using at least the following tools: Cygwin, Java JDK and Homebrew [7].

**Third party software** The IDE can be connected to third party software: Flash, VVVV, Processing or Max/MSP. Finally, Arduino has some dependencies during run-time [8]. These can be found in the top right corner of figure 2.

###### **Hardware Boards**

When using the Arduino IDE you also need some hardware to start a project, for this an Arduino controller, breadboard, wiring and sensors are needed. Arduino has two main suppliers of the hardware: Intel and Atmel.

อ้างอิง : [Arduino - Delft Students on Software Architecture: DESOSA 2017 \(gitbooks.io\)](https://gitbooks.io/Arduino-Delft-Students-on-Software-Architecture-DESOSA-2017)

## จุดอ่อนของ Architecture MVC

### Testability

จุดอ่อนของ Architecture MVC Testability ที่ใช้อยู่คือข้อจำกัดในเรื่องของการที่สามารถ Test ได้ยาก เนื่องจากการที่ทุกส่วนมีการเชื่อมต่อกันหมดจึงทำให้การ test เฉพาะส่วน รวมไปถึงทรัพยากรของเครื่องในการ test เอง

### Tactic for achieve Testability

เพื่อที่จะลบปัญหาในจุดนี้ ทาง Arduino ได้ทำสร้างตัว test แยกแต่ละส่วนเอาไว้ให้สำหรับผู้ที่ต้องการพัฒนาโดยเฉพาะ เพื่อที่จะรองรับปัญหาในด้านนี้ คล้ายกับการทำ sandbox และมีการกำหนด ทรัพยากรขั้นต่ำเอาไว้

## Building and testing

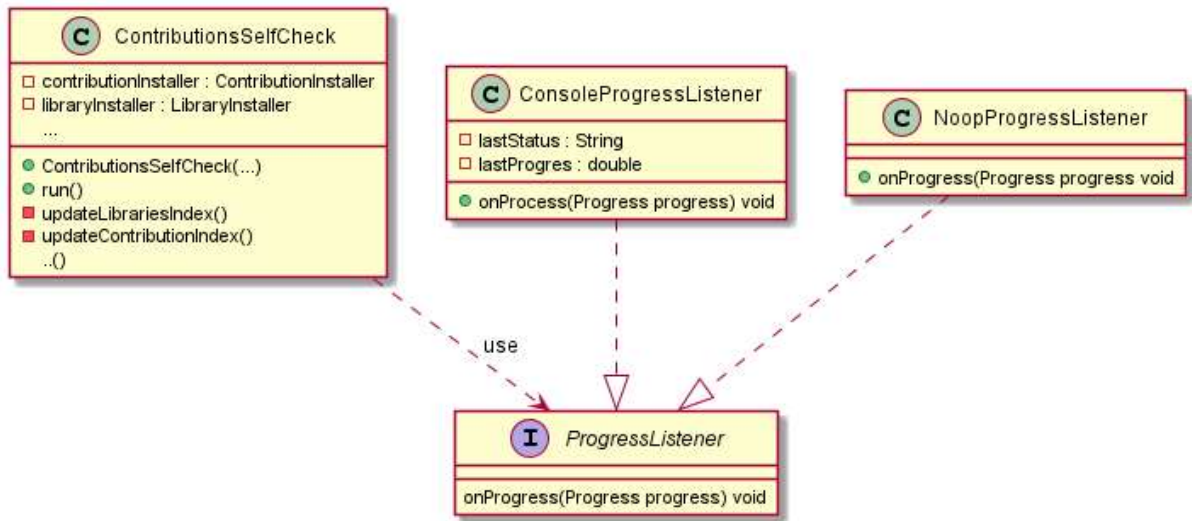
Instructions for building the IDE and running unit tests can be found on the wiki:

- <https://github.com/arduino/Arduino/wiki/Building-Arduino>
- <https://github.com/arduino/Arduino/wiki/Testing-Arduino>

Reference: [Testing Arduino · arduino/Arduino Wiki \(github.com\)](https://github.com/arduino/Arduino/wiki/Testing-Arduino)

## Design Pattern ของ Arduino IDE

### 1. Null Object



Null Object ออกแบบมาเพื่อทำหน้าที่เป็นค่าเริ่มต้นของ Object โดยเสนอทางเลือกทดแทนที่เหมาะสมให้กับค่าเริ่มต้น

ในที่นี้ก็คือ class ที่ชื่อว่า ContributionsSelfCheck ซึ่งถ้าหากมีการเรียกคลาสขึ้นมา ก็จะมีการสร้าง Object ContributionsSelfCheck ขึ้นมาโดยภายในก็จะมี Attribute ที่ชื่อ progressListener อยู่ โดยตัว progressListener เองจะมีค่าเป็น NoopProgressListener หรือเป็น Null object นั้นเอง ซึ่งถ้าจะทำการปรับเปลี่ยนตัว progressListener ให้เป็น object อื่นก็จะไปทำที่ method updateContributionIndex และ updateLibrariesIndex

#### Code

ContributionsSelfCheck.java(68 เรียกใช้ progressListener = new NoopProgressListener

211-220 เปลี่ยนแปลง progressListener ให้ update)

NoopProgressListener.java

[Arduino/NoopProgressListener.java at 57a931c9c4c084057417837239ad8136f8a7b1aa · arduino/Arduino \(github.com\)](#)

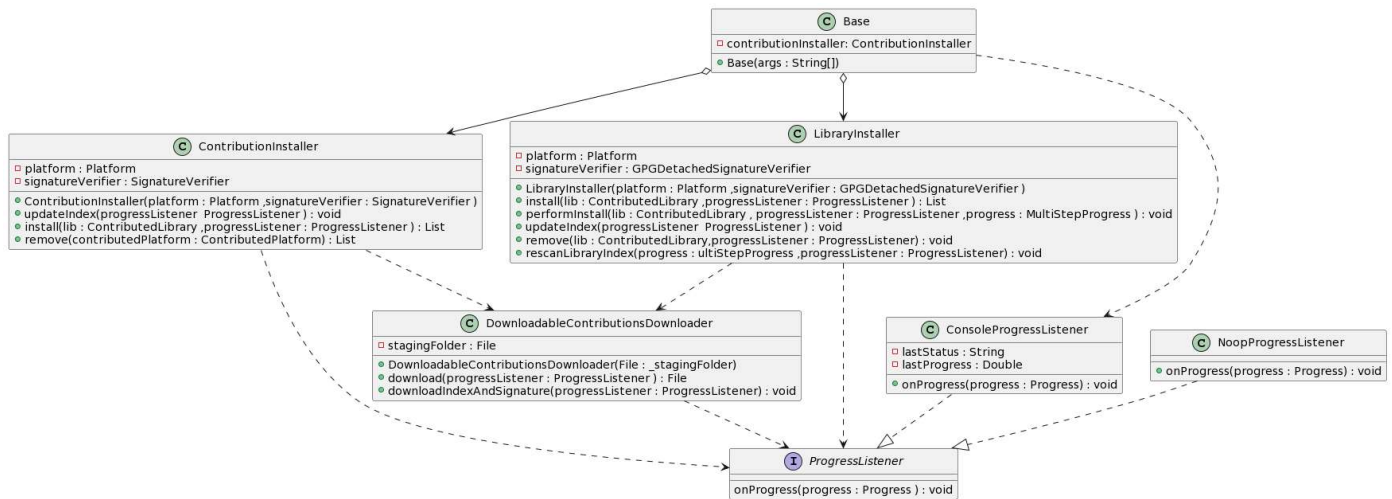
ProgressListener.java(Interface)

[Arduino/ProgressListener.java at 57a931c9c4c084057417837239ad8136f8a7b1aa · arduino/Arduino \(github.com\)](#)

ConsoleProgressListener(34-51)

[Arduino/ConsoleProgressListener.java at 57a931c9c4c084057417837239ad8136f8a7b1aa · arduino/Arduino \(github.com\)](#)

## 2. Observer



นี่จะเป็นการออกแบบเพื่อสร้างการติดต่อรับข้อมูลกันแบบ one to many ซึ่งเหมาะสมกับงานประเภทที่ถ้ามีการเปลี่ยนแปลงส่วนหนึ่งของการทำงานก็ต้องแจ้ง เกือบทุกส่วนว่ามีการเปลี่ยนแปลง

ซึ่งใน Arduino IDE นั้นมีการให้เลือก Library ในการ compile and verify รวมไปถึงการเลือกตัว Broad ที่ทำการเลือกเข้ามาใช้ หรือ แม้กระทั่งการติดตั้ง Library เพิ่มเติมนอกจากเดิม ทั้งหมดที่กล่าวมาจึงมการแจ้งเตือนเพื่อนให้ส่วนอื่นนั้นมีการปรับเปลี่ยนให้สามารถรับการเปลี่ยนแปลงได้

### Code

Base.java -> 96,294,312,341,345 (Base o--> ContributionInstaller)

-> 354,295,97,391,389 (Base o--> LibraryInstaller)

-> 310 (Base ..> ConsoleProgressListener)

[Arduino/Base.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

ContributionInstaller.java -> 73,132,154,280-303 (ContributionInstaller ..> ProgressListener)

-> 92,100,108,283(ContributionInstaller ..>DownloadableContributionsDownloader)

[Arduino/ContributionInstaller.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

LibraryInstaller.java -> 66,107,113,125,161,180,189,197,199(LibraryInstaller ..> ProgressListener)

-> 69,89,144,148(LibraryInstaller ..> DownloadableContributionsDownloader)

[Arduino/LibraryInstaller.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

ProgressListener.java -> 34-38 (Subscribe Interface)

[Arduino/ProgressListener.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

ConsoleProgressListener.java -> 34 (ConsoleProgressListener ..|> ProgressListener)  
39-50 (update)

[Arduino/ConsoleProgressListener.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

NoopProgressListener.java -> 34 (NoopProgressListener ..|> ProgressListener)  
37-39 (update)

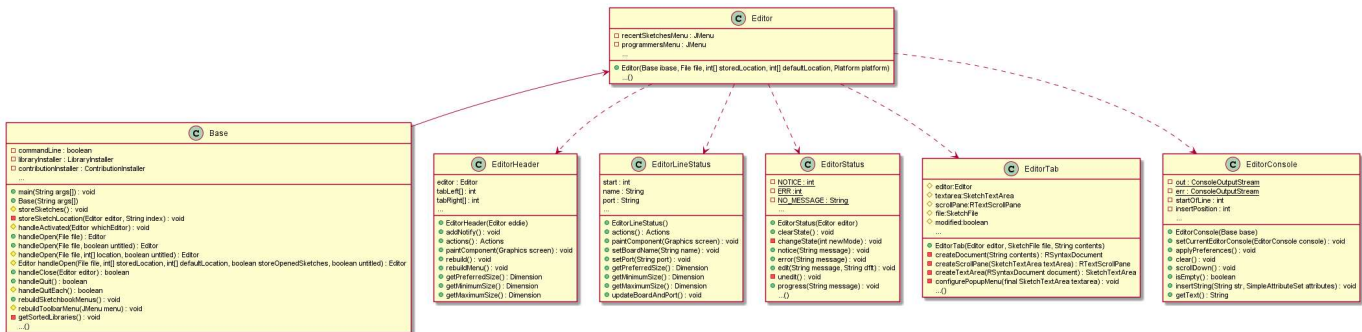
[Arduino/NoopProgressListener.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

DownloadableContributionsDownloader.java -> 59,85,129,141,149,195

(DownloadableContributionsDownloader ..> ProgressListener)

[Arduino/DownloadableContributionsDownloader.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

### 3. Façade



Façade ออกแบบไว้ใช้แก้ปัญหาคำสั่งที่ต้องทำงานหลายอย่าง ให้สามารถเรียกใช้ได้ด้วยคำสั่งเดียว โดยที่ไม่ต้องทำการเรียก คลาสที่เหลือ หรือ subclass ต่างๆมาด้วยตัวเอง

เช่นในที่นี้ Base มีการเรียกใช้ method จากหลากหลาย Editor class เช่น EditorHeader , EditorConsole เป็นต้น ซึ่งถ้าหากไม่มีการทำเป็น Façade ก็ได้แต่ทำให้การเรียกใช้นั้นซับซ้อนมากขึ้น เพราะจำนวนของ Editor class ที่มีเยอะและการเลือกใช้งานที่ดูคล้ายกันไปหมด ดังนั้นแทนที่ base class จะไปเรียกใช้ทีละตัวเอง ก็สามารถเรียกใช้ผ่าน class Editor เพื่อให้เกิดความง่ายและ ลดความซับซ้อนของการทำงานเองด้วย

Code(เนื่องจากการทำงานที่หลากหลาย จึงขอยกตัวอย่างมา 1 การทำงาน)

Base (1942 - เรียกใช้ Editor - applyPreferences)

[Arduino/Base.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

Editor (306,309,316,319,325 – = new Editor class ต่างๆเข้ามา)

(491-502 –เป็นการเรียกใช้ applyPreferences ของ EditorTab)

[Arduino/Editor.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

EditorTab (289)

[Arduino/EditorTab.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

EditorConsole (120)

[Arduino/EditorConsole.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

EditorHeader

[Arduino/EditorHeader.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

EditorLineStatus

[Arduino/EditorLineStatus.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

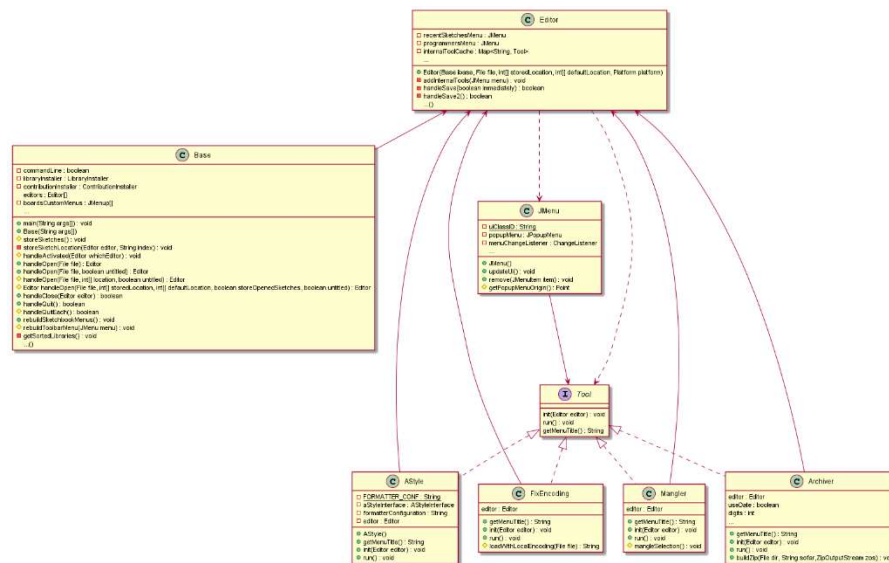
EditorStatus

[Arduino/EditorStatus.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

EditorToolbar

[Arduino/EditorToolbar.java at 18f55445ab95b193671306688f2a8f4877e95e17 · arduino/Arduino \(github.com\)](#)

## 4. Command



เป็น pattern เกี่ยวกับ “คำสั่ง” รูปแบบเดียวกันที่ติดต่อกับระบบอื่นที่มีความหลากหลายในการควบคุม เช่น การคำสั่ง Save จะถูกเรียกโดยหลากหลายรูปแบบ เช่น การกด Key ลัด “Ctrl + C” , การกดปุ่ม Save หรือ ไปเข้า Menu > file > save เป็นต้น ทั้งหมดนี้เป็นการทำคำสั่ง Save เดียวกันแต่ต่างแค่การวิธีการเรียกใช้ ซึ่งพบได้บ่อยใน Software ที่มีส่วน GUI

Command: Tool เป็นตัวกำหนดรูปแบบคำสั่งที่ทุกๆ Command จะมีเหมือนกัน เช่น Init, run โดย subclass ของ Tool มีทั้งหมด 4 class ได้แก่ AStyle, FixEncoding, Magler และ Archiver ซึ่งจะทำงานแตกต่างกันไป

Invoker: JMenu ทำหน้าที่เป็น Invoker หรือตัวที่ใช้เก็บ Command เอาไว้ โดยวิธีการจะแตกต่างจาก

Command design pattern ทั่วไปตรงที่ JMenu จะไม่ได้เก็บ Object ของ Command โดยตรง

แต่จะทำการอ้างอิงถึง Path ของ Command แต่ละตัวและทำการ add เข้าไปใน Object ของ JMenu

Receiver: เป็นส่วนที่จะนำ Command ไปใช้งาน ซึ่ง Editor จะทำหน้าที่ เรียกใช้ JMenu โดยจะส่ง Object ของ Editor เองไปให้ Command แต่ละตัวเก็บไว้ เพื่อแสดงผลแต่ละ Command ให้กับ Editor ต่อไป

Client: เป็นส่วนหลักในการเรียกใช้คำสั่งทั้งหมด โดย Base จะทำหน้าที่เป็น Client ที่มี Attribute เป็น

Editor โดยจะเรียกใช้งาน Editor ทันทีเมื่อมีการเรียกใช้ Base และ Editor จะไปเรียกใช้ JMenu ให้เก็บ

Command แต่ละตัวเอาไว้ และ ทำการ save เป็นลำดับถัดมา เพื่อบันทึก Command แต่ละที่อยู่ใน JMenu



## Code

Base -> 882 (Base --> Editor)

[Arduino/Base.java at master · arduino/Arduino \(github.com\)](#)

Editor -> 985-1,000 (Editor ..> JMenu)

971-983, 1908-1909 (Editor ..> Tool)

[Arduino/Editor.java at master · arduino/Arduino \(github.com\)](#)

Tool -> 32-40 (Command Interface)

[Arduino/Tool.java at master · arduino/Arduino \(github.com\)](#)

JMenu -> (Add Command to Menu)

AStyle -> 43-93 (Command Subclass)

43 (AStyle ..|> Tool)

49 (AStyle --> Editor)

[Arduino/AStyle.java at 57a931c9c4c084057417837239ad8136f8a7b1aa · arduino/Arduino \(github.com\)](#)

FixEncoding -> 37-98 (Command Subclass)

37 (FixEncoding..|> Tool)

38 (FixEncoding --> Editor)

[Arduino/FixEncoding.java at master · arduino/Arduino \(github.com\)](#)

Mangler -> 39-95 (Command Subclass)

39 (Mangler ..|> Tool)

40 (Mangler --> Editor)

[Arduino/Mangler.java at 57a931c9c4c084057417837239ad8136f8a7b1aa · arduino/Arduino \(github.com\)](#)

Archiver -> 44-181 (Command Subclass)

44 (FixEncoding.<|> Tool)

45 (FixEncoding --> Editor)

[Arduino/Archiver.java at master · arduino/Arduino \(github.com\)](#)

## อ้างอิงเพิ่มเติม

[Arduino · Delft Students on Software Architecture: DESOSA 2017 \(gitbooks.io\)](#)

[The Arduino IDE \(tudelft.nl\)](#)

[www.ArduinoFreeloader.com/no\\_help/decode\\_base64](#)

[arduino/Arduino: open-source electronics platform \(github.com\)](#)

[“Massive” View Controllers or bad coding style? | by Besar | Medium](#)

[MVC MVP MVVM คืออะไร และต่างกันอย่างไร | by Nutti Saelor | Medium](#)