

Maker L^AT_EX Package

V 1.0

Mg. Fausto Mauricio Lagos Suárez
piratax007@protonmail.ch

15 de julio de 2017

Resumen

El paquete `maker` provee ambientes y comandos basados en el paquete `listings` que permiten incluir rápidamente código `Arduino` o `Processing` utilizando el resaltado de sintaxis de su respectivo IDE.

Guía del usuario

1. ¿Qué puede hacer el paquete <code>maker</code> ?	1
1.1. Incluir código directamente en el documento	1
1.2. Incluir código en línea con el texto	2
1.3. Incluir código desde un archivo <code>.ino</code> o <code>.pde</code>	2
2. Instalación	3
3. Licencia	4

1. ¿Qué puede hacer el paquete `maker`?

El paquete `maker` tiene dos opciones, `Arduino` y `Processing` las cuales pueden utilizarse de forma independiente o conjunta en el mismo documento.

El paquete `maker` le permite incluir código de `Arduino` o `Processing` utilizando el resaltado de sintaxis propio de su respectivo IDE oficial, puede hacerlo de tres formas diferentes:

1. Escribiendo el código directamente en el documento L^AT_EX.
2. Incluyendo comando de `Arduino` o `Processing` en línea con el texto.
3. Cargando el código desde un archivo `.ino` (`Arduino`) o `.pde` (`processing`).

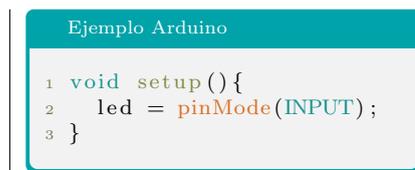
La versión V 1.0 del paquete `maker` esta basada en el resaltado de sintaxis de `Arduino` disponible en <https://www.arduino.cc/en/Reference/HomePage> y en la experiencia de usuario con `Processing`, si encuentra alguna modificación que deba hacerse a este paquete no dude en contactar con su desarrollador.

1.1. Incluir código directamente en el documento

Para incluir código de `Arduino` o `Processing` escribiéndolo directamente en el documento L^AT_EX se utiliza el ambiente `ArduinoSketchBox` o `ProcessingSketchBox` respectivamente.

Este ambiente tiene un parámetro de entrada obligatorio correspondiente al título o caption del código.

```
\begin{ArduinoSketchBox}{Ejemplo Arduino}
void setup(){
  led = pinMode(INPUT);
}
\end{ArduinoSketchBox}
```



```

\begin{ProcessingSketchBox}{Ejemplo Processing}
void draw(){
  ellipse(50, 50, 25, 30);
}
\end{ProcessingSketchBox}

```

Ejemplo Processing

```

1 void draw(){
2   ellipse(50, 50, 25, 30);
3 }

```

Los ambientes `ArduinoSketchBox` y `ProcessingSketchBox` son ideales para pequeñas piezas de código.

1.2. Incluir código en línea con el texto

Los comandos `\ArduinoInline` y `\ProcessingInline` permiten incluir comandos de cualquiera de estos dos lenguajes en línea con el texto, su uso es muy simple ya que tiene un único parámetro de entrada que corresponde con el código a incluir.

Un sketch dinámico de Arduino utiliza las funciones `\ArduinoInline{void setup()}` y `\ArduinoInLine{void loop()}` mientras que su equivalente en Processing utiliza las funciones `\ProcessingInline{void setup()}` y `\ProcessingInline{void draw()}`.

Un sketch dinámico de Arduino utiliza las funciones `void setup()` y `void loop()` mientras que su equivalente en Processing utiliza las funciones `void setup()` y `void draw()`.

1.3. Incluir código desde un archivo .ino o .pde

Incluir código de Arduino o Processing a partir de un archivo `.ino` o `.pde` es muy fácil utilizando el comando `\ArduinoSketch` o `\ProcessingSketch` que tiene dos parámetros de entrada, el nombre del archivo de código sin extensión y el texto del caption.

`\ArduinoSketch{Blink}{Ejemplo de código Arduino a partir de un archivo .ino}`

```

1 /*
2  Blink
3  Turns on an LED on for one second, then off for one second, repeatedly.
4
5  Most Arduinos have an on-board LED you can control. On the UNO, MEGA and ZERO
6  it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN takes care
7  of use the correct LED pin whatever is the board used.
8  If you want to know what pin the on-board LED is connected to on your Arduino model,
9  check
10 the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
11
12 This example code is in the public domain.
13
14 modified 8 May 2014
15 by Scott Fitzgerald
16
17 modified 2 Sep 2016
18 by Arturo Guadalupi
19 */
20 // the setup function runs once when you press reset or power the board
21 void setup() {
22   // initialize digital pin LED_BUILTIN as an output.
23   pinMode(LED_BUILTIN, OUTPUT);
24 }
25
26 // the loop function runs over and over again forever
27 void loop() {
28   digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
29   delay(1000); // wait for a second
30   digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
31   delay(1000); // wait for a second
32 }

```

Listing 1: Ejemplo de código Arduino a partir de un archivo .ino

```
1 gota [] lluvia = new gota[100];
2 float Y1, Y2, v;
3
4 void setup(){
5   size(400, 350);
6   for (int i = 0; i < lluvia.length; i++){
7     Y1 = random(-height, 0);
8     Y2 = Y1 + 5;
9     v = 3.2;
10    lluvia[i] = new gota(Y1, Y2, v);
11  }
12 }
13
14 void draw(){
15   background(125);
16   for (int i = 0; i < lluvia.length; i++){
17     lluvia[i].display();
18     lluvia[i].fall();
19   }
20 }
21
22 class gota{
23   float x, y1, y2, v;
24
25   gota(float Y1, float Y2, float velocity){
26     x = random(width);
27     y1 = Y1;
28     y2 = Y2;
29     v = velocity;
30   }
31
32   void display(){
33     line(x, y1, x, y2);
34   }
35
36   void fall(){
37     if (y2 <= height){
38       y1 += v;
39       y2 += v;
40     }
41     else {
42       y1 = random(-50, 0);
43       y2 = y1 + 5;
44     }
45   }
46 }
```

Listing 2: Ejemplo de código Processing a partir de un archivo .pde

2. Instalación

Probablemente `maker` no este instalado por defecto en su distribución de \LaTeX , si este es el caso puede utilizar el administrador de paquetes de su distribución para instalar `maker` o de otra forma puede instalarlo manualmente siguiendo las instrucciones a continuación.

Deberá descargar el archivo `maker.sty` desde la [página del proyecto](#), una vez disponga localmente del archivo `maker.sty` tendrá que copiarlo en el árbol de directorios de su distribución que variará de acuerdo al sistema operativo que este utilizando, puede referirse a [TEX FAQ](#) para encontrar instrucciones más detalladas. Si quiere trabajar con la última versión deberá reemplazar el archivo `maker.sty` por el más reciente descargado desde la página del proyecto.

3. Licencia

Este trabajo puede ser distribuido o modificado bajo los términos y condiciones de la LaTeX Project Public License (LPPL) v1.3C, o cualquier versión posterior. La última versión de esta licencia puede consultarse en <http://www.latex-project.org/lppl.txt>.

Usted es libre de utilizarlo, modificarlo y compartirlo siempre que se respeten los términos y condiciones de la licencia y se reconozca al autor original.