21 December 2019
Advanced Input Device
高级输入设备
Week 4 assignments:

- Publish, modify and re-publish several times the page using git workflow
  使用 Git 工作流程发布、修改、反复发布你的网页
- Trying out another sensor
  试一试其它的传感器
- Trying out to fade an LED In and Out, like on a sleeping apple computer
  可选：试试 LED 的渐亮和渐淡编程，就像是苹果电脑的呼吸灯一样。
- Create a page to document your progress of week 4
  创建一个网页去记录本周的收获，拍下项目成功运行的照片上传。
    - What did you do and learn? 你做了什么？学到了什么？
    - What are your experiences? (problems, solution, etc.). 你经历了什么？（困难有哪些？解决方案是？等）
    - Upload a picture and the programming code to your website. 上传照片，编你的网站。

Please bring with you: 带上

- Your personal laptop 个人笔记本电脑
- A mouse 鼠标

And please install the following software: 下载下面软件

- Solidworks
- Cura
  https://ultimaker.com/software/ultimaker-cura

Do not hesitate to contact us if you have any question. 有疑问可以随时联系老师。

Reference:
https://wiki.dfrobot.com/
https://wiki.dfrobot.com/Gravity__DS18B20_Temperature_Sensor__Arduino_Compatible__V2_SKU__DFR0024
https://wiki.dfrobot.com/DFRobot_Ambient_Light_Sensor_SKU_DFR0026

Algorithm 算法 ：

A step-by-step procedure for solving a problem or accomplishing some end.

Value 值

A numerical quantity that is assigned or is determined by calculation or measurement.

Variable 变量

A quantity that may assume any one of a set of values.

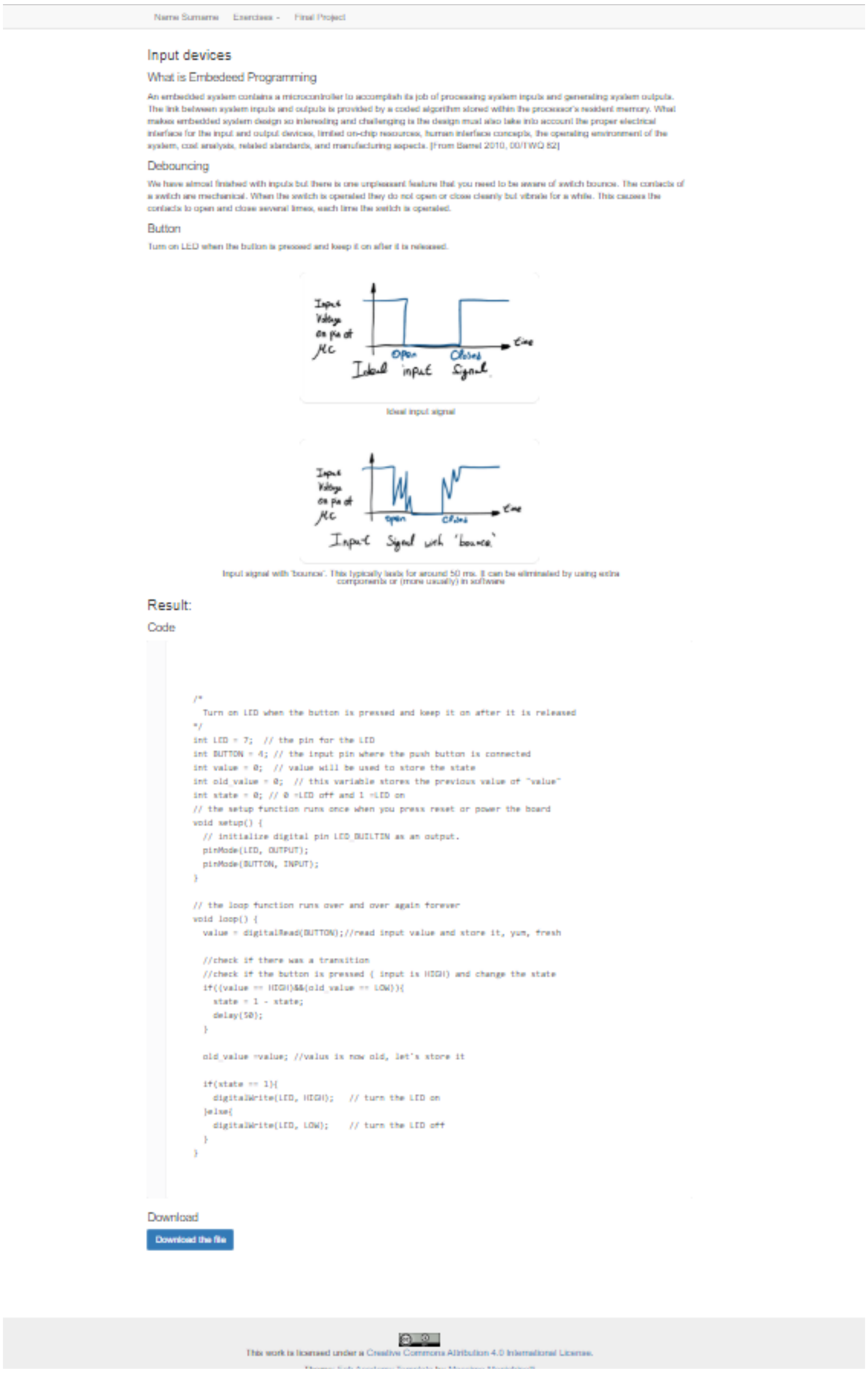
Documentation

This is the example of the documentation for input device
黄色的部分是需要学生用自己的内容去填充的代码

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <meta name="description" content="">
    <meta name="author" content="Name Surname">
    <link rel="icon" href="media/favicon.ico">

    <title>Input devices</title>

    <!-- Bootstrap core CSS -->
    <link href="bootstrap/css/bootstrap.min.css" rel="stylesheet">

    <!-- Custom styles for this template -->
    <link href="media/fabacademy.css" rel="stylesheet">

    <!-- 3D files viewer -->
    <script type="text/javascript" src="media/jsc3d_ie.min.js"></script>
      <script type="text/javascript" src="media/jsc3d.min.js"></script>
    <script type="text/javascript" src="media/jsc3d.webgl.js"></script>
    <script type="text/javascript" src="media/jsc3d.touch.js"></script>
    <!-- Just for debugging purposes. Don't actually copy these 2 lines! -
->
    <!--[if lt IE 9]><script src="bootstrap/js/ie8-responsive-file-
warning.js"></script><![endif]-->
    <script src="bootstrap/js/ie-emulation-modes-warning.js"></script>

    <!-- HTML5 shim and Respond.js for IE8 support of HTML5 elements and
media queries -->
    <!--[if lt IE 9]>
      <script
src="https://oss.maxcdn.com/html5shiv/3.7.2/html5shiv.min.js"></script>
      <script
src="https://oss.maxcdn.com/respond/1.4.2/respond.min.js"></script>
    <![endif]-->

    <!-- Load the menu file -->
    <script>
      function menu() {
                                $('#exercises').load("exercises-
menu.html");
                                $('#project').load("project-menu.html");
                                $('#cclicense').load("license.html");
                                }
      </script>

  </head>

  <body onload="menu()">

    <!-- Static navbar -->
    <nav class="navbar navbar-default navbar-static-top">
      <div class="container">
        <div class="navbar-header">
          <button type="button" class="navbar-toggle collapsed" data-
toggle="collapse" data-target="#navbar" aria-expanded="false" aria-
controls="navbar">
            <span class="sr-only">Toggle navigation</span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
            <span class="icon-bar"></span>
          </button>
        </div>
        <div id="navbar" class="navbar-collapse collapse">
```

```
          <ul class="nav navbar-nav">
            <li><a href="index.html">Name Surname</a></li>
            <li class="dropdown">
              <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-expanded="false">Exercises <span
class="caret"></span></a>
              <ul id="exercises" class="dropdown-menu" role="menu">
              </ul>
            </li>
              <li><a href="project.html">Final Project</a></li>
        </ul>
        </div><!--/.nav-collapse -->
      </div>
    </nav>
    <div class="container">

      <!-- Insert your content here below! -->


     <h2>Input devices</h2>

    <h3>What is Embedeed Programming</h3>
      <p>
      An embedded system contains a microcontroller to accomplish its job
of processing system inputs and generating system outputs. The link between
system inputs and outputs is provided by a coded algorithm stored within
the processor's resident memory. What makes embedded system design so
interesting and challenging is the design must also take into account the
proper electrical interface for the input and output devices, limited on-
chip resources, human interface concepts, the operating environment of the
system, cost analysis, related standards, and manufacturing aspects. [From
Barret 2010, 00/TWQ 82]
      </p>

    <h3>Debouncing</h3>
      <p>
      We have almost finished with inputs but there is one unpleasant
feature that you need to be aware of switch bounce. The contacts of a
switch are mechanical. When the switch is operated they do not open or
close cleanly but vibrate for a while. This causes the contacts to open and
close several times, each time the switch is operated.
      </p>

    <h3>Button</h3>
    <p>Turn on LED when the button is pressed and keep it on after it is
released.</p>


    <p class="pic"><img
src="http://archive.fabacademy.org/2018/labs/fablabkamplintfort/students/ma
rcello-tania/media/exercise8/switchBounce.png"
    style="width:500px">
      <legend>Ideal input signal</legend>
      </p>

    <p class="pic"><img
src="http://archive.fabacademy.org/2018/labs/fablabkamplintfort/students/ma
rcello-tania/media/exercise8/switchBounce1.png"
    style="width:500px">
      <legend>Input signal with 'bounce'. This typically lasts for around
50 ms. It can be eliminated by using extra components or (more usually) in
software</legend>
      </p>


    <h2>Result:</h2>

      <h3>Code</h3>
      <pre class="prettyprint linenums">
    <xmp>

        /*
          Turn on LED when the button is pressed and keep it on after it is
released
        */
        int LED = 7;  // the pin for the LED
        int BUTTON = 4; // the input pin where the push button is connected
        int value = 0;  // value will be used to store the state
        int old_value = 0;  // this variable stores the previous value of
"value"
        int state = 0; // 0 =LED off and 1 =LED on
        // the setup function runs once when you press reset or power the
board
        void setup() {
          // initialize digital pin LED_BUILTIN as an output.
          pinMode(LED, OUTPUT);
          pinMode(BUTTON, INPUT);
        }
        // the loop function runs over and over again forever
        void loop() {
          value = digitalRead(BUTTON);//read input value and store it, yum,
fresh

          //check if there was a transition
          //check if the button is pressed ( input is HIGH) and change the
state
          if((value == HIGH)&&(old_value == LOW)){
            state = 1 - state;
            delay(50);
          }

          old_value =value; //valus is now old, let's store it

          if(state == 1){
            digitalWrite(LED, HIGH);    // turn the LED on
          }else{
            digitalWrite(LED, LOW);     // turn the LED off
          }
        }
    </xmp>
</pre>


    <h3>Download</h3>

      <p>
      <a href="media/pushButton/">
      <button type="button" class="btn btn-primary btn-lg">Download the
file</button>
      </a>
      </p>

    <!-- End of your content -->

    </div> <!-- /container -->

      <!-- footer -->

    <footer id="footer">
        <p id="cclicense">
        </p>
        <p class="license">
        Theme: <a
href="https://github.com/openp2pdesign/FabAcademy_Template">Fab Academy
Template</a> by <a href="http://openp2pdesign.org">Massimo Menichinelli</a>
<br>
        Based on <a href="http://getbootstrap.com/">Twitter
Bootstrap</a>+<a href="http://jquery.com/">JQuery</a>+<a
href="https://code.google.com/p/google-code-prettify/">google-code-
prettify</a>+<a href="http://jmblog.github.io/color-themes-for-google-code-
prettify/github/">GitHub theme for google-code-prettify</a>+<a
href="https://code.google.com/p/jsc3d/">JSC3D</a>+<a
href="https://github.com/thegrubbsian/jquery.ganttView">jquery.ganttView</a
>.
        </p>
    </footer>


      <!-- Do not touch this! -->
    <!-- Bootstrap core JavaScript
    ================================================== -->
    <!-- Placed at the end of the document so the pages load faster -->
    <script src="media/jquery-1.9.1.min.js"></script>

    <!-- Syntax Highlighter -->
    <script src="https://google-code-
prettify.googlecode.com/svn/loader/run_prettify.js">
    </script>
    <!-- From https://github.com/jmblog/color-themes-for-google-code-
prettify -->
    <link href="media/github.css" type="text/css" rel="stylesheet">
      <script type="text/javascript">
        !function ($) {
            $(function(){
              window.prettyPrint && prettyPrint()
            })
        }(window.jQuery)
      </script>

    <script src="bootstrap/js/bootstrap.min.js"></script>
    <!-- IE10 viewport hack for Surface/desktop Windows 8 bug -->
    <script src="bootstrap/js/ie10-viewport-bug-workaround.js"></script>

  </body>
</html>
```

Result　网页最终显示的样板



Name Surname Exercises - Final Project

Input devices

What is Embedeed Programming

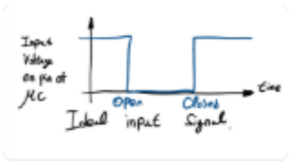
An embedded system contains a microcontroller to accomplish its job of processing system inputs and generating system outputs. The link between system inputs and outputs is provided by a coded algorithm stored within the processor's resident memory. What makes embedded system design so interesting and challenging is the design must also take into account the proper electrical interface for the input and output devices, limited on-chip resources, human interface concepts, the operating environment of the system, cost analysis, related standards, and manufacturing aspects. [From Barrel 2010, 00/TWQ 82]

Debouncing

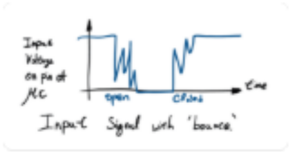
We have almost finished with inputs but there is one unpleasant feature that you need to be aware of switch bounce. The contacts of a switch are mechanical. When the switch is operated they do not open or close cleanly but vibrate for a while. This causes the contacts to open and close several times, each time the switch is operated.

Button

Turn on LED when the button is pressed and keep it on after it is released.



Ideal input signal



Input signal with 'bounce'. This typically lasts for around 50 ms. It can be eliminated by using extra components or (more usually) in software

Result:

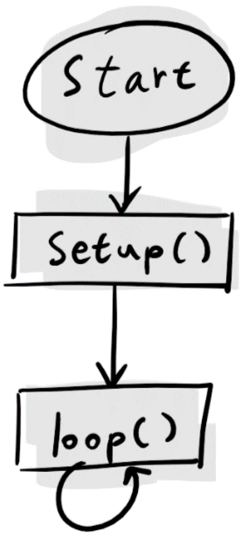
Code

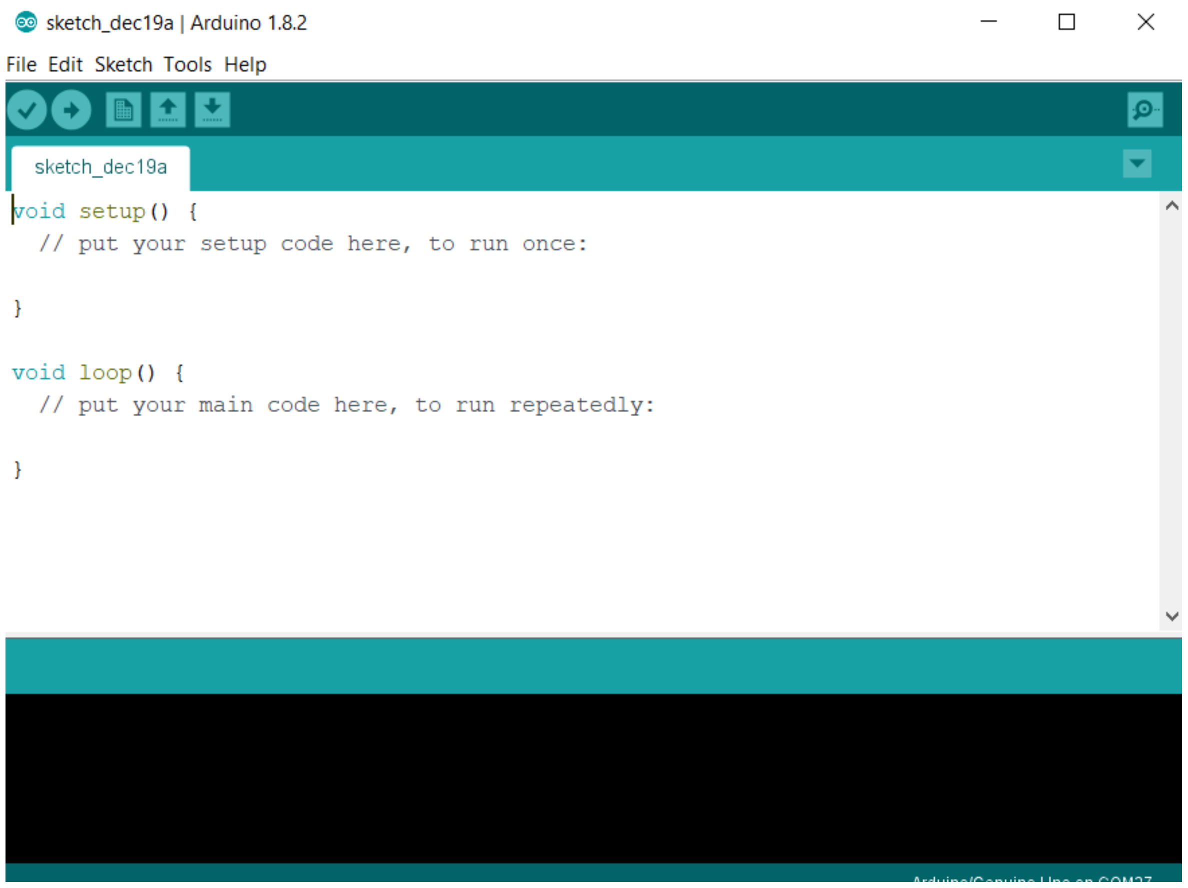
```
/*
  Turn on LED when the button is pressed and keep it on after it is released
*/
int LED = 7;  // the pin for the LED
int BUTTON = 4; // the input pin where the push button is connected
int value = 0;  // value will be used to store the state
int old_value = 0;  // this variable stores the previous value of "value"
int state = 0; // 0 =LED off and 1 =LED on
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

// the loop function runs over and over again forever
void loop() {
  value = digitalRead(BUTTON);//read input value and store it, yum, fresh

  //check if there was a transition
  //check if the button is pressed ( input is HIGH) and change the state
  if((value == HIGH)&&(old_value == LOW)){
    state = 1 - state;
    delay(50);
  }

  old_value =value; //valus is now old, let's store it

  if(state == 1){
    digitalWrite(LED, HIGH);   // turn the LED on
  }else{
    digitalWrite(LED, LOW);    // turn the LED off
  }
}
```

Download

Download the file

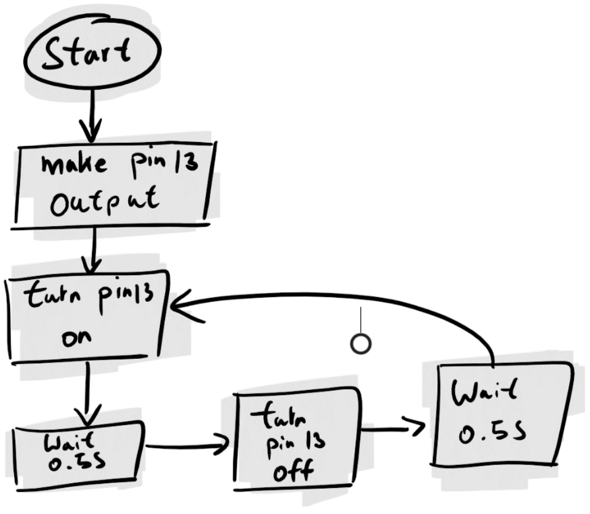
Arduino IDE and Flow Chart

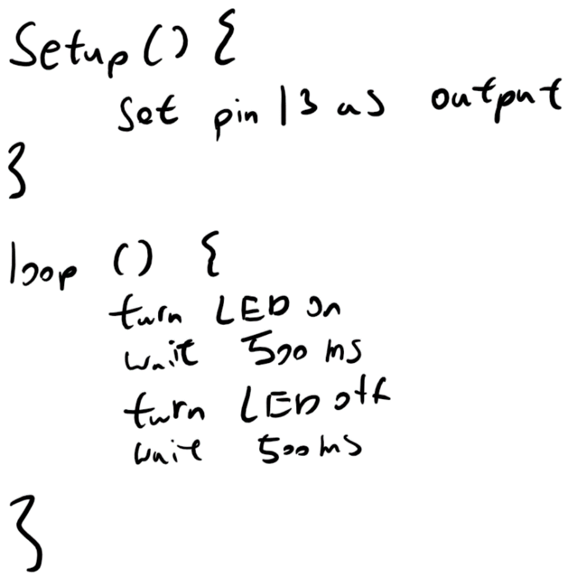


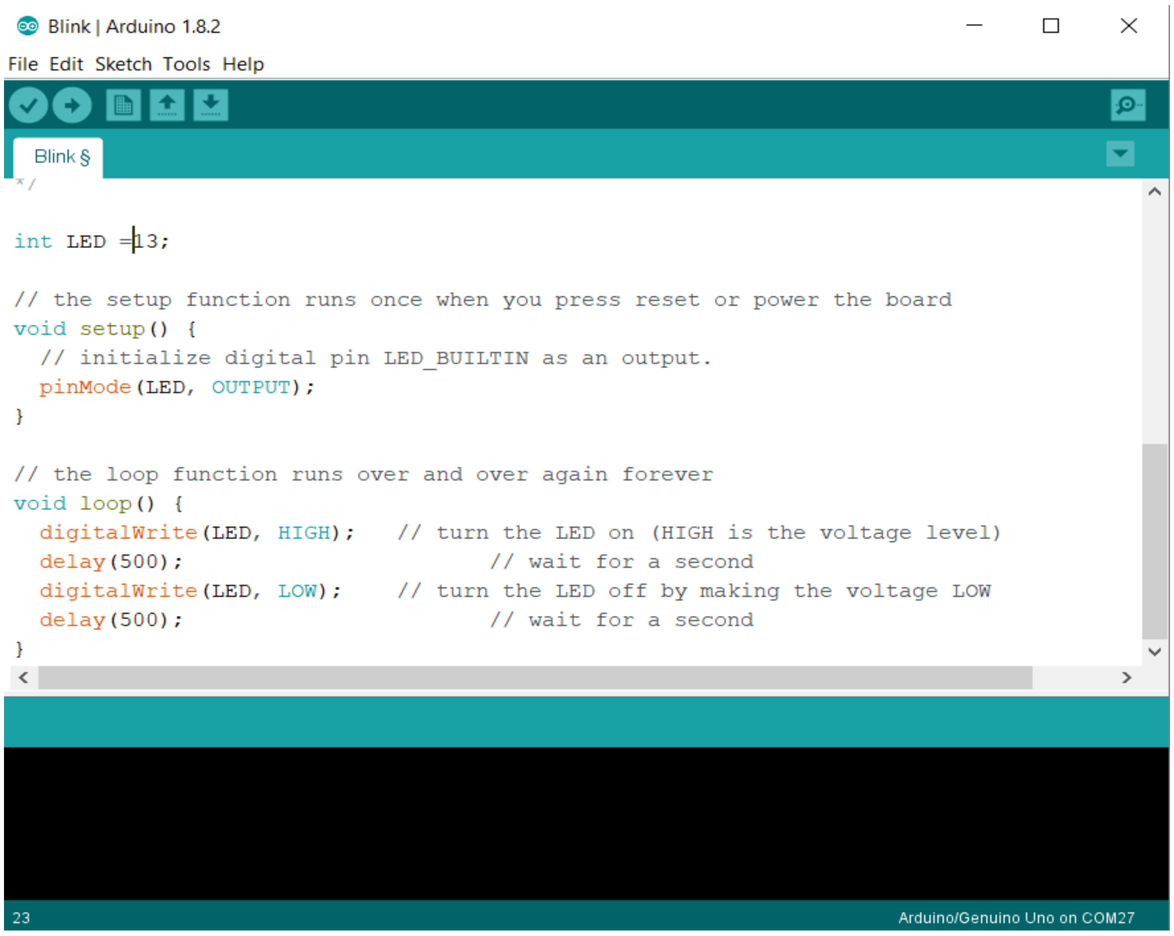


```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Blink Example LED 闪烁代码逻辑图

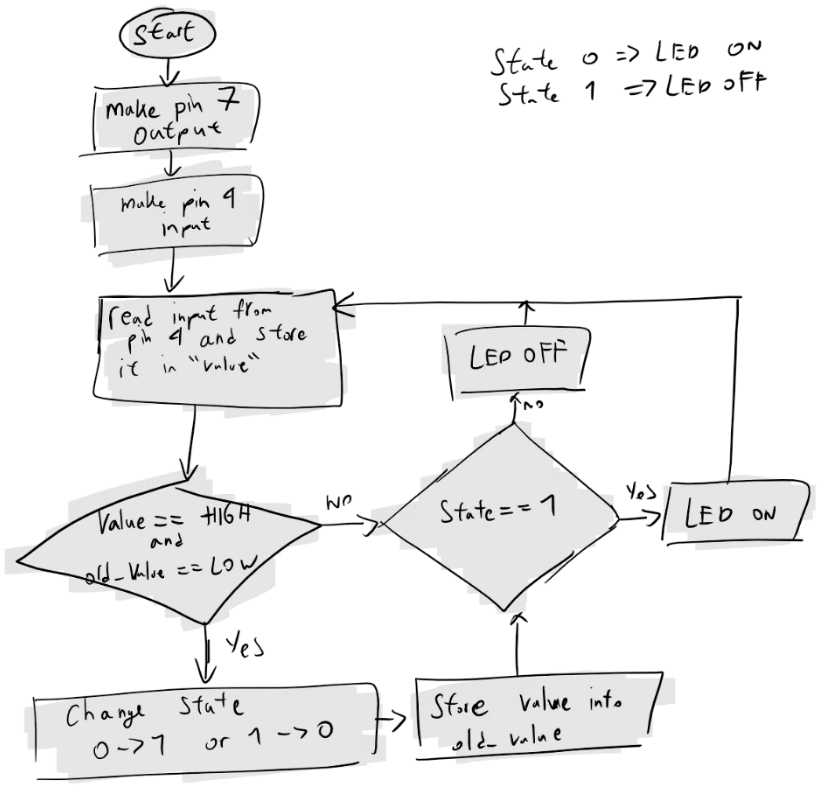






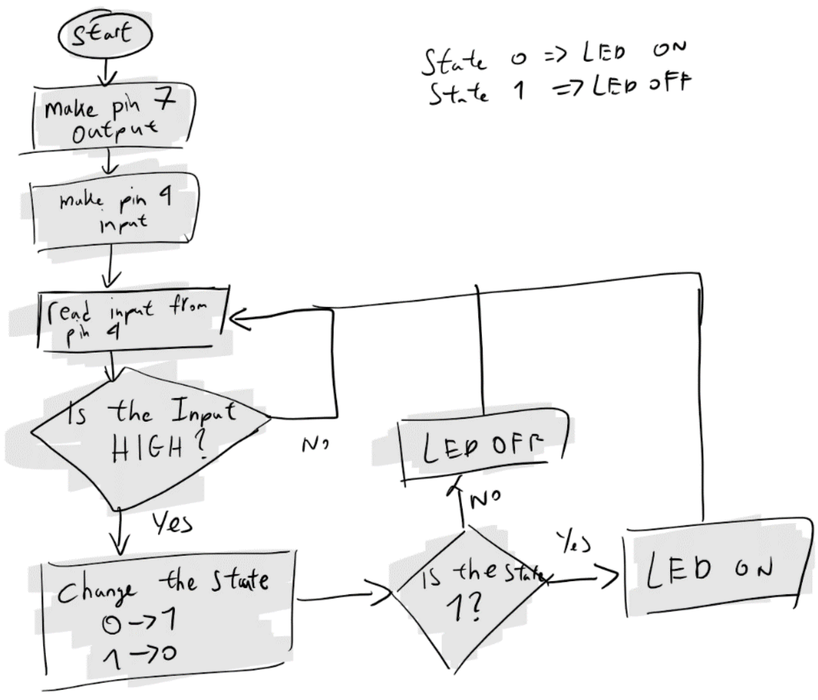
```
*/

int LED =13;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(500);                       // wait for a second
  digitalWrite(LED, LOW);    // turn the LED off by making the voltage LOW
  delay(500);                       // wait for a second
}
```

Turn on LED when the button is pressed and keep it on after it is released (Not perfect) 按按钮 LED 亮，按完后 LED 保持亮（不完美案例）。

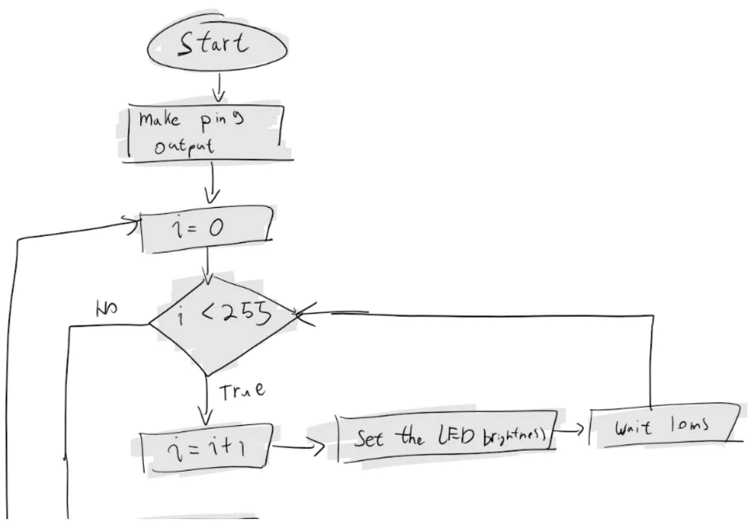

Start
make pin 7 output
make pin 4 input
read input from pin 4 and store it in "value"
Value == HIGH and old_Value == LOW
Yes
No
Change State 0->1 or 1->0
Store value into old_value
State == 1
Yes
No
LED ON
LED OFF
State 0 => LED ON
State 1 => LED OFF

```
int LED = 7;
int BUTTON = 4;
int value = 0;
int state = 0;

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

// the loop function runs over and over again forever
void loop() {
  value = digitalRead(BUTTON);

  //check if the button is pressed ( input is HIGH) and change the state
  if(value == HIGH){
    state = 1 - state;
  }
  if(state == 1){
    digitalWrite(LED, HIGH);    // turn the LED on (HIGH is the voltage level)
  }else{
    digitalWrite(LED, LOW);     // turn the LED off by making the voltage LOW
  }
}
```

Turn on LED when the button is pressed and keep it on after it is released. 按按钮 LED 亮，按完后 LED 保持亮。



Fade an LED In and Out, Like on a Sleeping Apple Computer。渐亮和渐淡，就像是苹果电脑的呼吸灯。



```
/*
  Turn on LED when the button is pressed and keep it on after it is released
*/
int LED = 7;  // the pin for the LED
int BUTTON = 4; // the input pin where the push button is connected
int value = 0;  // value will be used to store the state
int old_value = 0;  // this variable stores the previous value of "value"
int state = 0; // 0 =LED off and 1 =LED on
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED, OUTPUT);
  pinMode(BUTTON, INPUT);
}

// the loop function runs over and over again forever
void loop() {
  value = digitalRead(BUTTON);//read input value and store it, yum, fresh

  //check if there was a transition
  //check if the button is pressed ( input is HIGH) and change the state
  if((value == HIGH)&&(old_value == LOW)){
    state = 1 - state;
  }

  old_value =value; //valus is now old, let's store it

  if(state == 1){
    digitalWrite(LED, HIGH);   // turn the LED on
  }else{
    digitalWrite(LED, LOW);    // turn the LED off
  }
}
```

```
const int LED = 9; // the pin for the LED
int i = 0; // We'll use this to count up and down

void setup() {
 pinMode(LED, OUTPUT); // tell Arduino LED is an output
}
void loop(){
 for (i = 0; i < 255; i++) { // loop from 0 to 254 (fade in)
 analogWrite(LED, i); // set the LED brightness
 delay(10); // Wait 10ms because analogWrite
 // is instantaneous and we would
 // not see any change
 }
 for (i = 255; i > 0; i--) { // loop from 255 to 1 (fade out)
 analogWrite(LED, i); // set the LED brightness
 delay(10); // Wait 10ms
 }
}
```