Short Press and Long Press with debouncing During Pressing

```
/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page:
https://arduinogetstarted.com/tutorials/arduino-button-long-press-short-pres
s
 */

#include <ezButton.h>

const int SHORT_PRESS_TIME = 1000; // 1000 milliseconds
const int LONG_PRESS_TIME  = 1000; // 1000 milliseconds

ezButton button(7);  // create ezButton object that attach to pin 7;

unsigned long pressedTime  = 0;
unsigned long releasedTime = 0;
bool isPressing = false;
bool isLongDetected = false;

void setup() {
  Serial.begin(9600);
  button.setDebounceTime(50); // set debounce time to 50 milliseconds
}

void loop() {
  button.loop(); // MUST call the loop() function first

  if(button.isPressed()){
    pressedTime = millis();
    isPressing = true;
    isLongDetected = false;
  }

  if(button.isReleased()) {
    isPressing = false;
    releasedTime = millis();

    long pressDuration = releasedTime - pressedTime;

    if( pressDuration < SHORT_PRESS_TIME )
      Serial.println("A short press is detected");
  }
```

```
  if(isPressing == true && isLongDetected == false) {
    long pressDuration = millis() - pressedTime;

    if( pressDuration > LONG_PRESS_TIME ) {
      Serial.println("A long press is detected");
      isLongDetected = true;
    }
  }
}
```

Short Press and Long Press with debouncing after released

```
/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page:
https://arduinogetstarted.com/tutorials/arduino-button-long-press-short-pres
s
 */

#include <ezButton.h>

const int SHORT_PRESS_TIME = 1000; // 1000 milliseconds
const int LONG_PRESS_TIME  = 1000; // 1000 milliseconds

ezButton button(7);  // create ezButton object that attach to pin 7;

unsigned long pressedTime  = 0;
unsigned long releasedTime = 0;

void setup() {
  Serial.begin(9600);
  button.setDebounceTime(50); // set debounce time to 50 milliseconds
}

void loop() {
  button.loop(); // MUST call the loop() function first

  if(button.isPressed())
    pressedTime = millis();

  if(button.isReleased()) {
    releasedTime = millis();

    long pressDuration = releasedTime - pressedTime;
```

```
    if( pressDuration < SHORT_PRESS_TIME )
      Serial.println("A short press is detected");

    if( pressDuration > LONG_PRESS_TIME )
      Serial.println("A long press is detected");
  }
}
```

Si no posem el **debouncing** podem tenir problmes de rebot "**chattering phenomenon**" els següents poden tenir aquest problema:

Short Press and Long Press During pressing

```
/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page:
https://arduinogetstarted.com/tutorials/arduino-button-long-press-short-pres
s
 */

// constants won't change. They're used here to set pin numbers:
const int BUTTON_PIN = 7; // the number of the pushbutton pin
const int SHORT_PRESS_TIME = 1000; // 1000 milliseconds
const int LONG_PRESS_TIME  = 1000; // 1000 milliseconds

// Variables will change:
int lastState = LOW;  // the previous state from the input pin
int currentState;     // the current reading from the input pin
unsigned long pressedTime  = 0;
unsigned long releasedTime = 0;
bool isPressing = false;
bool isLongDetected = false;

void setup() {
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
  // read the state of the switch/button:
  currentState = digitalRead(BUTTON_PIN);

  if(lastState == HIGH && currentState == LOW) {        // button is pressed
    pressedTime = millis();
    isPressing = true;
    isLongDetected = false;
  } else if(lastState == LOW && currentState == HIGH) { // button is
released
```

```
    isPressing = false;
    releasedTime = millis();

    long pressDuration = releasedTime - pressedTime;

    if( pressDuration < SHORT_PRESS_TIME )
      Serial.println("A short press is detected");
  }

  if(isPressing == true && isLongDetected == false) {
    long pressDuration = millis() - pressedTime;

    if( pressDuration > LONG_PRESS_TIME ) {
      Serial.println("A long press is detected");
      isLongDetected = true;
    }
  }

  // save the the last state
  lastState = currentState;
}
```

Short Press and Long Press after released

```

/*
 * Created by ArduinoGetStarted.com
 *
 * This example code is in the public domain
 *
 * Tutorial page:
https://arduinogetstarted.com/tutorials/arduino-button-long-press-short-pres
s
 */

// constants won't change. They're used here to set pin numbers:
const int BUTTON_PIN = 7; // the number of the pushbutton pin
const int SHORT_PRESS_TIME = 1000; // 1000 milliseconds
const int LONG_PRESS_TIME  = 1000; // 1000 milliseconds

// Variables will change:
int lastState = LOW;  // the previous state from the input pin
int currentState;     // the current reading from the input pin
unsigned long pressedTime  = 0;
unsigned long releasedTime = 0;


void setup() {
  Serial.begin(9600);
  pinMode(BUTTON_PIN, INPUT_PULLUP);
```

```
}

void loop() {
  // read the state of the switch/button:
  currentState = digitalRead(BUTTON_PIN);

  if(lastState == HIGH && currentState == LOW)        // button is pressed
    pressedTime = millis();
  else if(lastState == LOW && currentState == HIGH) { // button is released
    releasedTime = millis();

    long pressDuration = releasedTime - pressedTime;

    if( pressDuration < SHORT_PRESS_TIME )
      Serial.println("A short press is detected");

    if( pressDuration > LONG_PRESS_TIME )
      Serial.println("A long press is detected");
  }

  // save the the last state
  lastState = currentState;
}
```