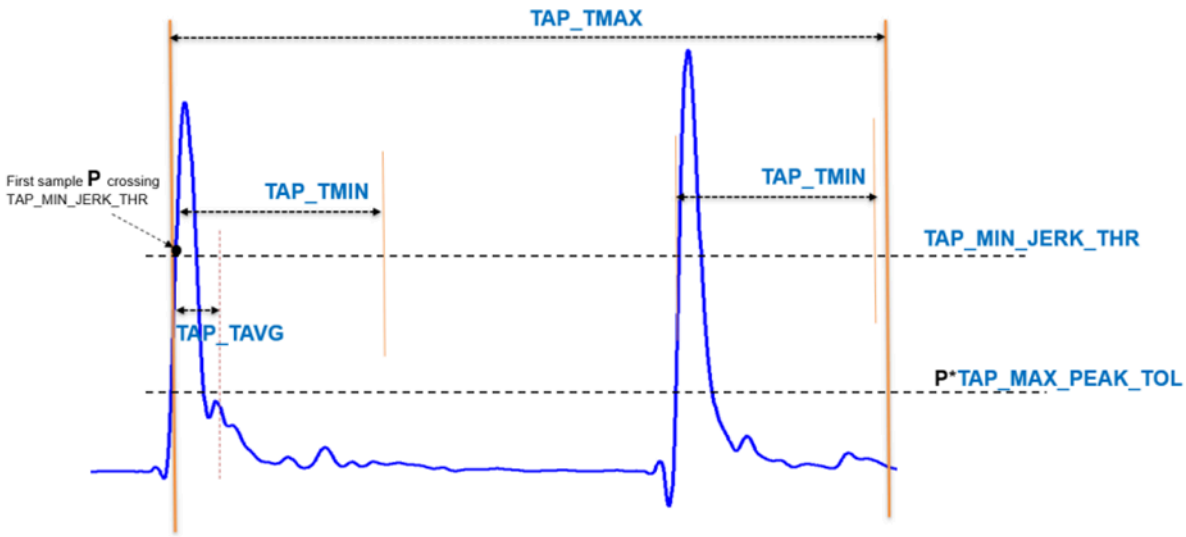

Figure 13. Timing diagram showing configurable input parameters.

The TAP_MIN_JERK_THR, TAP_MAX_PEAK_TOL, T_MIN, T_MAX and T_AVG values should be adjusted properly for the tap to be detected in the way that we want it to be with minimum noise. The T_MAX and T_AVG values have no significant effect on single tap detection, so the values given in the datasheet can be used.

The following code can be used for varying the threshold and tolerance value by just pressing the boot button.

```
#include <Wire.h>

int now;
int start = millis();
const int ICM = 0x68;
int mode;
int flag = 0;
int count = 0;
int detect;b.      I ..
int state=0;
//uint8_t threshold = 254;
//const int ledPin = 2;     // GPIO pin connected to the built-in LED
const int buttonPin = 9; // GPIO pin connected to the boot button

int Reg_read(unsigned long add){      // reading data from the
particular register
  Wire.beginTransmission(ICM);
  Wire.write(add);
  Wire.endTransmission();
  Wire.requestFrom(ICM, 1);
  if (Wire.available()){
    return Wire.read();}

}

void Reg_write(unsigned long add, uint8_t and_=255, uint8_t or_=0){
// write data to specific bitsm
  Wire.beginTransmission(ICM);
  Wire.write(add);
  Wire.endTransmission();
  Wire.requestFrom(ICM, 1);
  if (Wire.available()){
    mode=(Wire.read()&and_)|or_;
    Wire.beginTransmission(ICM);
    Wire.write(add);
    Wire.write(mode);
    Wire.endTransmission();
  }
}

void Reg_dir_write(unsigned long add, uint8_t val){      // write data to whole byte
  Wire.beginTransmission(ICM);
  Wire.write(add);
  Wire.write(val);
  Wire.endTransmission();
}

void Tap_detect(){     // ISR
  flag=1;
}

void setup() {

  Serial.begin(115200);
  //pinMode(ledPin, OUTPUT);
  pinMode(buttonPin, INPUT_PULLUP);
//
//  while (Serial.available() == 0) {
//  }
  now = millis();
  while(now-start<=10000){
  Serial.println("One");
  if (digitalRead(buttonPin) == LOW) { // Check if the button is pressed
    state++;        // Turn on the LED
    delay(750);
  }
//  for(int i = 0; i<state;i++)
//  {digitalWrite(ledPin, HIGH);
//   delay(50);
//   digitalWrite(ledPin, LOW);
//  }
  now = millis();
 }


  // while (Serial.available() == 0) {
  // }

  // uint8_t Tmin = Serial.parseInt();
  // Serial.println(Tmin);


  Wire.begin();

  Reg_dir_write(0x76, 0);   // set bank =0

  Reg_dir_write(0x50, 15);  //  ACCEL_ODR = 15 (500 Hz)

  Reg_write(0x4E, 254, 2);  // accel mode =2
  Reg_write(0x4D, 247);  // accel clk = 0
  Reg_write(0x53, 253, 4);  // ACCEL_DEC2_M2_ORD = 2
  Reg_write(0x52, 15, 64);  // ACCEL_UI_FILT_BW = 4

  delay(500);

  Reg_dir_write(0x76, 4);   // set bank =4

  Reg_write(0x47, 219, 91);  // TAP_TMAX to 2, TAP_TMIN to 3, TAP_TAVG to 3   // changes

  if (state == 0){
  Reg_dir_write(0x46, 70);  // TAP_MIN_JERK_THR to 31, TAP_MAX_PEAK_TOL to 0
  Serial.println("70");
  }
  else if (state == 1){
  Reg_dir_write(0x46, 252);  // TAP_MIN_JERK_THR to 31,
TAP_MAX_PEAK_TOL to 0
  Serial.println("252");
  }
  else if (state == 2){
  Reg_dir_write(0x46, 253);  // TAP_MIN_JERK_THR to 31,
TAP_MAX_PEAK_TOL to 0
  Serial.println("253");
  }
  else if (state == 3){
  Reg_dir_write(0x46, 255);  // TAP_MIN_JERK_THR to 31,
TAP_MAX_PEAK_TOL to 0
  Serial.println("255");
  }
  else if (state ==4){
  Reg_dir_write(0x46, 203);
  Serial.println("203");
  }
  else if (state >=5){
  Reg_dir_write(0x46, 163);
  Serial.println("163");
  }
  delay(500);


  Reg_write(0x4D, 255, 1);  // enable INT1 = 1

  delay(500);

  Reg_dir_write(0x76, 0);   // set bank =0

  Reg_write(0x56, 255, 64);  // enable INT1 = 1

  attachInterrupt(digitalPinToInterrupt(5), Tap_detect, RISING);    //
calling ISR at interrupt   //changes

  delay(100);
  Serial.println("Setup Done");

}
```

```
void loop() {
  if (flag==1){           // from ISR after detection
    //Serial.println("Tapp detected");
    detect = Reg_read(0x38);
    if (detect%2==1){                 // double check register for tap detection
      Serial.println("Tapp");
      flag = 0;
      count++;
      Serial.println(count);
      //digitalWrite(ledPin, HIGH);
      //delay(50);
      //digitalWrite(ledPin, LOW);
    }
  }
}
```

The link for the heat map is bellow
--> *Heat Map for variation of thres, tol and tmin* --> *thres and tol map.xlsx*

Thre\Tol	63
0	Sensitive
1	less sensitive, better than 0
2	less sensitive, better than 1
3	more sensitive

Thus we found that the threshold value of 63 and tolerance value of 2 is best fit for single tap detection with least noise. Thus the decimal value that should be given to 0x46 is 254.