

```

#include <PulseSensorPlayground.h>
#include <LCD5110_Graph.h>

LCD5110 myGLCD(8, 9, 10, 11, 12);

// Pulse Sensor bağlantı pini
const int PulseWire = A1;
// Nabız sayısını saklayacağımız değişken
int BPM;

PulseSensorPlayground pulseSensor;
extern uint8_t SmallFont[];
extern uint8_t MediumNumbers[];
extern uint8_t BigNumbers[];

const int buttonPin = 2; // Düğme pini
int buttonState = 0;     // Düğme durumu
int lastButtonState = LOW; // Son düğme durumu

bool showEKG = false;    // EKG veya BPM göstermek için değişken

void setup() {
    // LCD ekranı başla
    myGLCD.InitLCD();
    myGLCD.setContrast(65);
    Serial.begin(9600);
    myGLCD.clrScr();

    // Pulse Sensor ayarları
    pulseSensor.analogInput(PulseWire);
    pulseSensor.setThreshold(510); // Eşik değerini sensörünüzün hassasiyetine göre
    ayarlayın

    // Pulse Sensor başlat
    if (!pulseSensor.begin()) {
        Serial.println("Pulse sensor start failed!");
        while (true); // Don't proceed, loop forever
    }

    // Button pini giriş olarak ayarla
    pinMode(buttonPin, INPUT);
}

void loop() {
    // Düğme durumunu oku

```

```

buttonState = digitalRead(buttonPin);

// Düğme basımı kontrolü (debounce yapılabilir)
if (buttonState == HIGH && lastButtonState == LOW) {
    showEKG = !showEKG; // Modu değiştir
    myGLCD.clrScr(); // Ekranı temizle
    delay(50); // Debounce
}
lastButtonState = buttonState;

if (showEKG) {
    showEKGDisplay();
} else {
    showBPM();
}
}

void showBPM() {
    // Nabız sayısını al
    BPM = pulseSensor.getBeatsPerMinute();

    // Yeni bir nabız atışı algılandığında true döner
    pulseSensor.outputSample();

    // Her nabız atışını algıladığında ekranda BPM'i güncelle
    if (pulseSensor.sawStartOfBeat()) {
        myGLCD.setFont(SmallFont);
        myGLCD.print("NABIZ", CENTER, 0);
        myGLCD.setFont(MediumNumbers);
        myGLCD.printNumI(BPM, CENTER, 10);
        myGLCD.update();
    }
}

void showEKGDisplay() {
    int piksel[84];
    for (int i = 1; i < 82; i++) {
        int x = analogRead(A0);
        int z = map(x, 100, 600, 1, 47);
        int y = 48 - z;
        piksel[1] = 24;
        piksel[i + 1] = y;
        myGLCD.drawLine(i, piksel[i], i + 1, piksel[i + 1]);
        myGLCD.update();
        delay(20);
    }
}

```

```
}  
myGLCD.clrScr();  
}
```