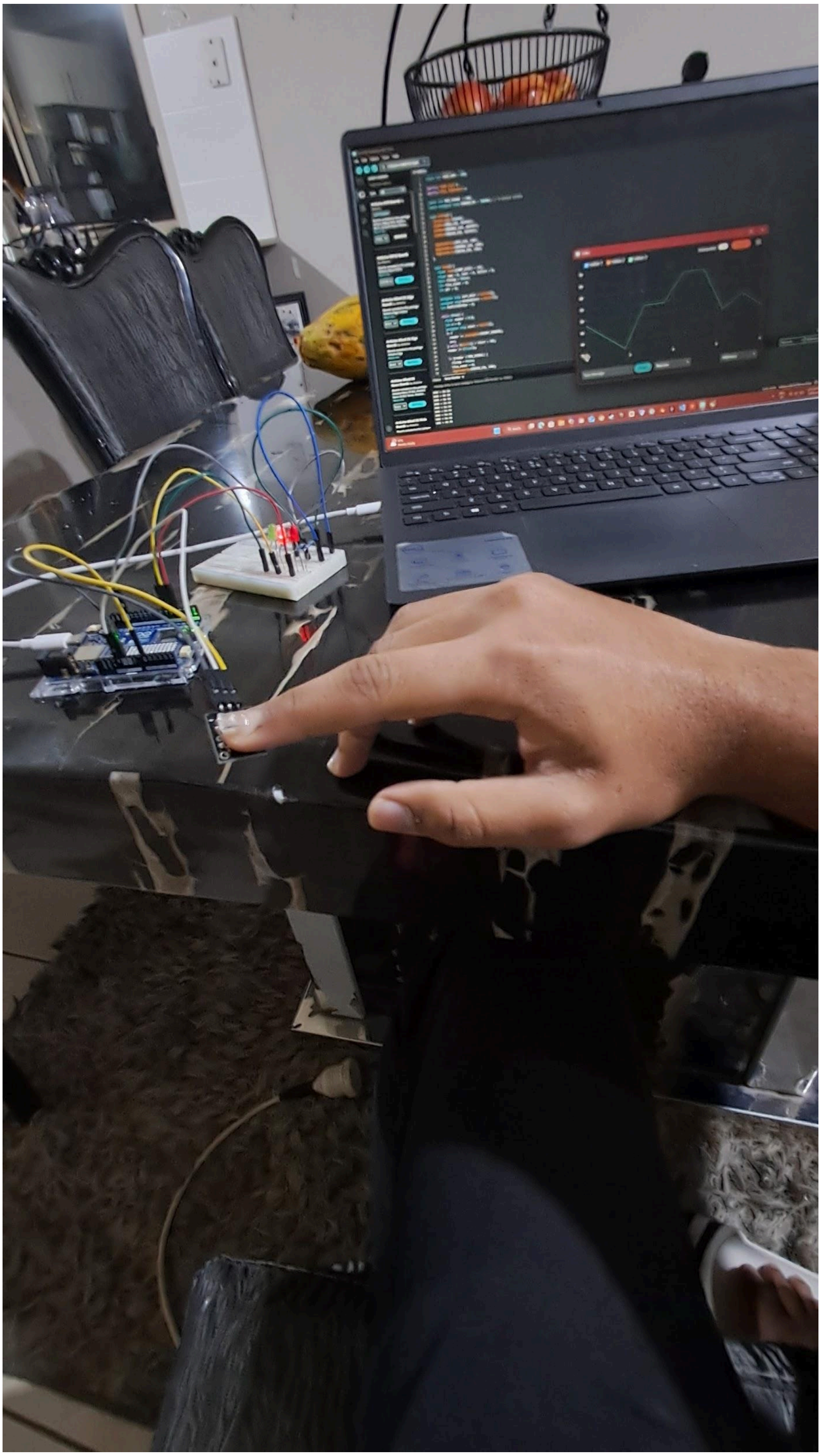
Mechatronics Heartbeat Monitor Project



By Ekambir Singh

Project Overview

This project involves designing and building a heartbeat monitoring system utilising a combination of mechatronic components and software engineering principles. The system will ensure that the heartbeat sensor will be connected to the Arduino and will record the heartbeat data, which will be processed through the human finger to the sensor, then the Arduino to the laptop, where the data will be shown.

Requirements Definition

The project will involve the design and construction of a heartbeat sensor that is to be integrated with mechatronic components and embedded software features. The core sensing element will be the KY-039 heartbeat sensor module. This sensor uses an infrared (IR) LED and a phototransistor to detect changes in blood flow through the fingertip. When a person places their finger between the LED and the receiver, the blood vessels with each heartbeat alter the amount of IR light received. These variations are output as an analog voltage signal in the range of 0-1023. The system will use a fingertip to detect pulse, and the sensor will be connected to an Arduino Uno to detect variation in pulses and blood flow, which is caused by each heartbeat. These variations produce an analog voltage signal, which will have a purpose to read the Arduino through an analog input pin. Written in the embedded program, the Arduino IDE software will process this signal to identify heartbeat peaks, which will be calculated in BPM (beats per minute), and display the result in real time to a serial monitor.

The problem that is to be addressed or solved in this project is the lack of affordable, portable, and easy-to-use biometric monitoring devices within community and education settings. While the sensors used in the project will not achieve medical-grade accuracy due to having limited capabilities of hardware, it will demonstrate the capabilities of low-cost, real-time heartbeat sensors to perform basic health tracking and data visualisation. This is suitable for applications such as sport training feedback, classroom demonstrations, or basic wellness monitoring.

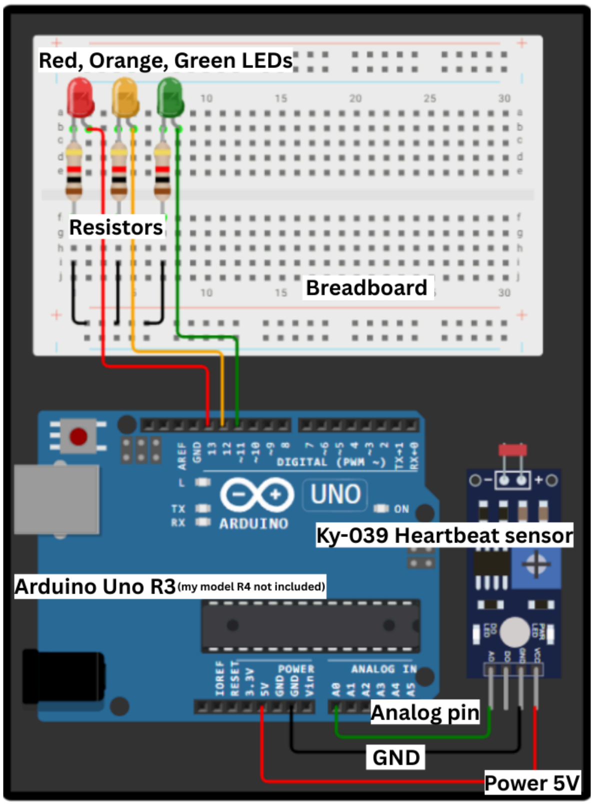
The primary focus of the data being collected will be the user's heart rate in BPM. This is obtained by placing a fingertip on the pulse sensor, which will detect the changes in light absorption caused by blood flow. The KY-039 module is small and compact, being approximately 18x7.8x3 mm, and operates a 5V, making it fully compatible with the Arduino Uno and ideal for portable or space-constrained applications..The sensors output an analog signal to the Arduino Uno, which will process the data using filtering and peak detection algorithms. The BPM value is then calculated and displayed on the Serial Monitor. Adding three LEDs as a visual indication of the heart rate zones: orange for resting (<60 BPM), green for normal (60-100 BPM) and red for high (>100 BPM).

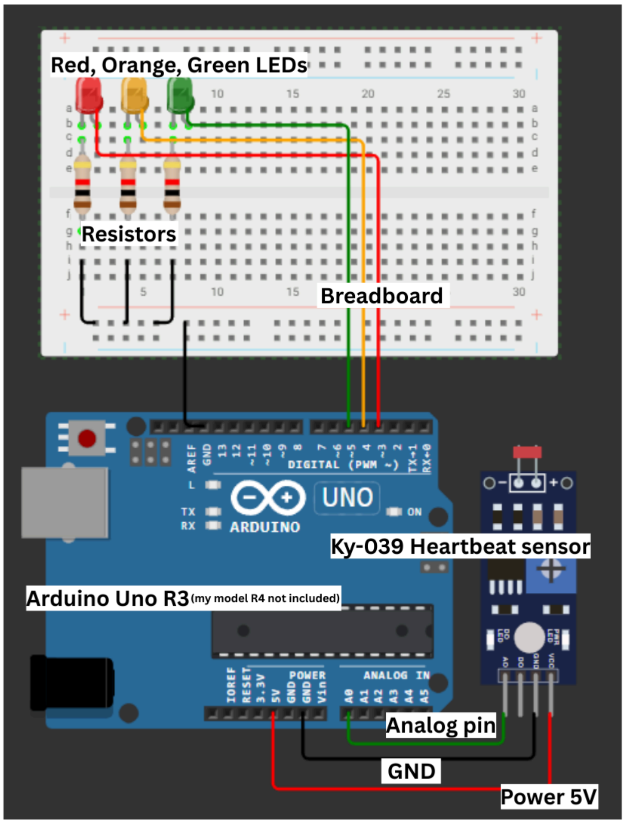
Data is then stored temporarily in Arduino variables using "***int***" and "***float***" types of data during data processing.

The program will use control structures to loop and conditional statements, such as “***if***” statements, to manage signal sampling, filtering, and BPM calculation. Constant will be used to define BPM thresholds for LED colour changes. The display will be handled through serial communication to the laptop, with the option to include LED feedback for a quick visual reference. The wiring will follow the standard KY-039 pinout GND to GND, +5 to 5V, and Signal to Analog (A0-5). This approach will ensure that the system demonstrates the need for both hardware and software integration and the structured programming techniques required for effective mechatronic solutions.

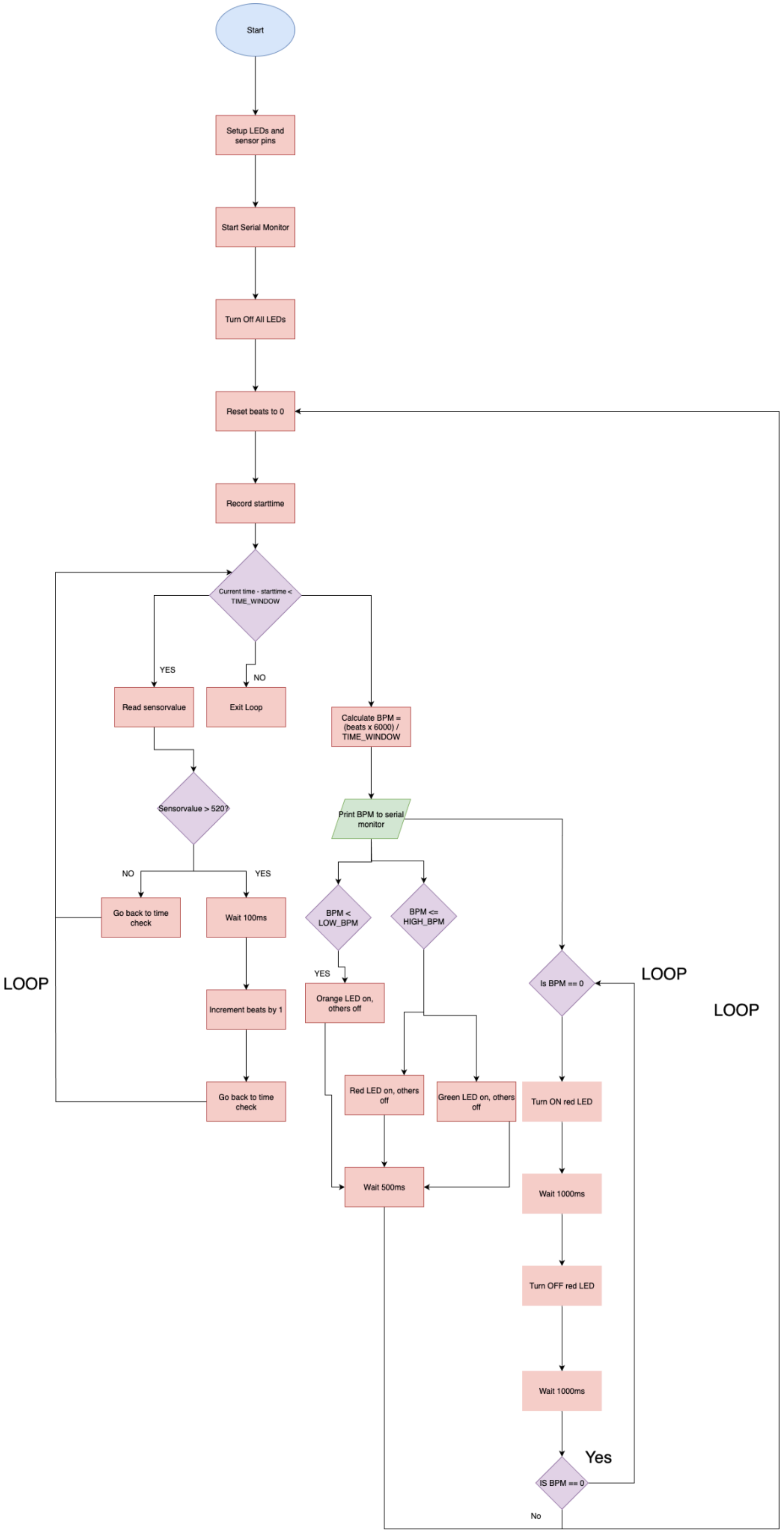
Project sketches and designs

Old version

Versions	Image
Version 1	 Red, Orange, Green LEDs Resistors Breadboard Arduino Uno R3 (my model R4 not included) Ky-039 Heartbeat sensor Analog pin GND Power 5V

Version 2	 Red, Orange, Green LEDs Resistors Breadboard Ky-039 Heartbeat sensor Arduino Uno R3 (my model R4 not included) Analog pin GND Power 5V

Algorithm design and desk checks

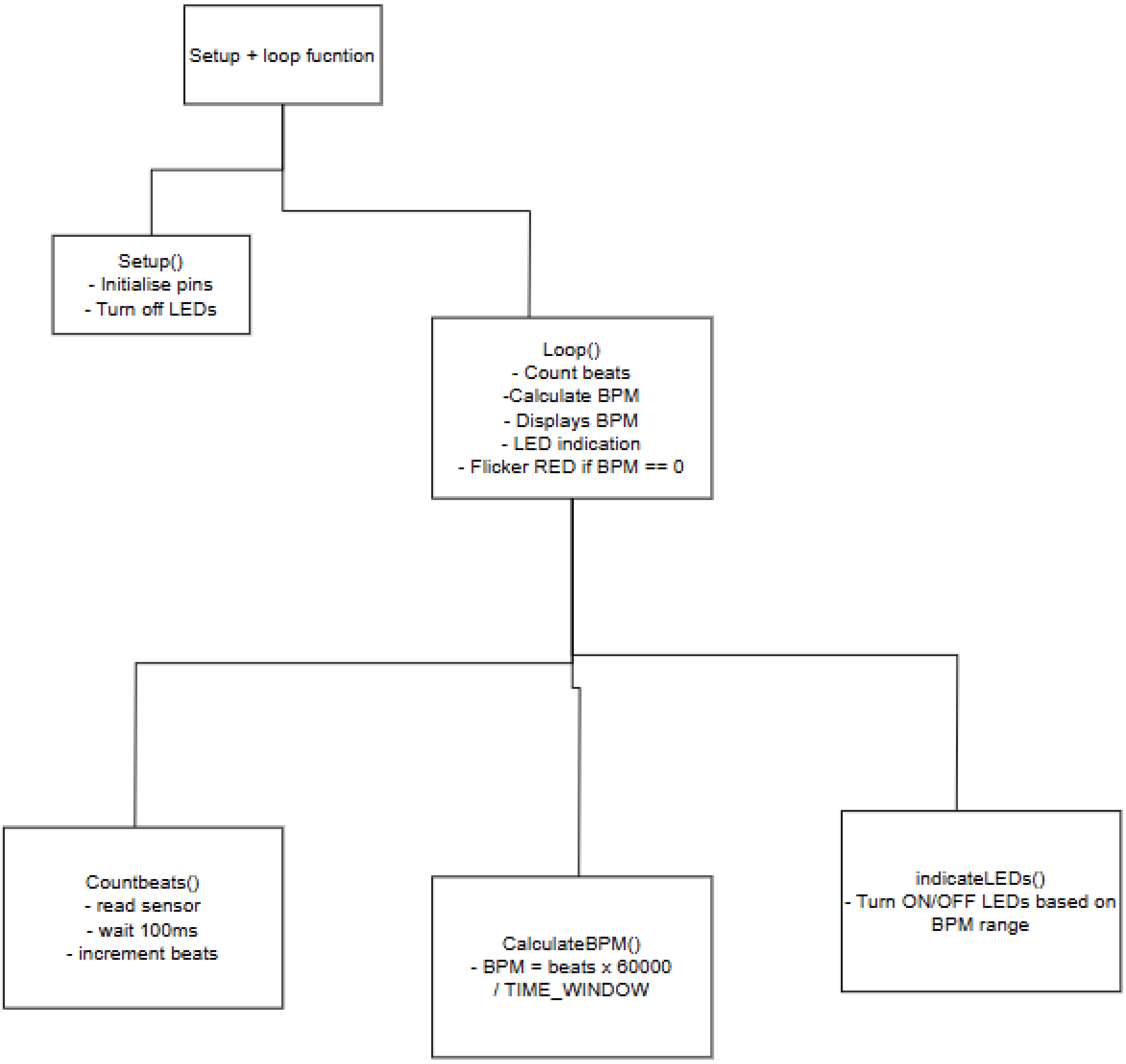

Start
Setup LEDs and sensor pins
Start Serial Monitor
Turn Off All LEDs
Reset beats to 0
Record starttime
Current time - starttime < TIME_WINDOW
YES
NO
Read sensorvalue
Exit Loop
Calculate BPM = (beats x 6000) / TIME_WINDOW
Sensorvalue > 520?
Print BPM to serial monitor
NO
YES
Go back to time check
Wait 100ms
BPM < LOW_BPM
BPM <= HIGH_BPM
LOOP
YES
Is BPM == 0
LOOP
LOOP
Increment beats by 1
Orange LED on, others off
Red LED on, others off
Green LED on, others off
Turn ON red LED
Go back to time check
Wait 500ms
Wait 1000ms
Turn OFF red LED
Wait 1000ms
Yes
IS BPM == 0
No

None

```
// Pseudocode

// Assign pin nmbers to each LEDs and heartbeat sensor
SET RED_LED to pin 2
SET ORANGE_LED to pin 3
SET GREEN_LED to pin 4
SET HEART_SENSOR to analog pin A0

// Set BPM threshold
SET LOW_BPM = <60 // Less then 60 considered low (resting)
SET NORMAL_BPM = 60-100 // Between 60-100 conpsided normal (active)
SET HIGH_BPM = >100  // Greater than 100 is considered high (fast heart rate)
SET TIME_WINDOW = 5000 milliseconds // 5 seconds

// Function Setup
// Runs once when Arduino is powered on
FUNCTION setup()
       START serial monitor 9600 speed // Begain serial commnication
       SET LED pins to OUTPUT // Allow LEDs to be controlled
       TURN OFF al LEDs // Make sure all LEDs start off
END FUNCTION

// loop function running forever
FUNCTION loop()
       SET beats = 0 // reset heart beat counter
       SET starttime = current time // Get the current itme to start timing

       // read heartbeasts for 5 seconds
       WHILE current time - starttime < TIME_WINDOW
              READ sensorvalue from HEART_SENSOR // Get sensor value

       IF sensorvalue > 520 then // If value higher then thresold
              WAIT 100 milliseconds // prent double counting the same one beat
              ADD 1 to beats // count one heartbeat
       END IF
END WHILE

       // calcualte heartrate
       SET BPM = (beats * 60000) / TIME_WINDOW // convert numbers of beats to BPM

       // Show BPM in serial mointor
       PRINT "BPM = " + BPM // show the restuilt in the serial monitor


       // Check for 0 BPm and blink RED rapidly
       IF BPM == 0 THEN
              PRINT "No heartbeat detected."
              SET blinkStart = current time
              WHILE current time - blinkStart < 5000 // blink for 5 seconds
              TURN ON RED_LED
```

```
             WAIT 200 milliseconds
             TURN OFF RED_LED
             WAIT 200 milliseconds
        END WHILE
        RETURN // skip LED indication and restart loop
    END IF


       // LED indication
       IF BPM < LOW_BPM THEN //If heartrate is low
             TURN ON ORANGE_LED
             TURN OFF RED_LED and GREEN_LED
       ELSE IF BPM <= HIGH_BPM THEN // if heartrate is high
             TURN ON RED_LED
             TURN OFF GREEN_LED and ORANGE_LED
       ELSE
             TURN ON GREEN_LED // If heartaret is normal
             TURN OFF RED_LED and ORANGE_LED
       END IF

       WAIT 500 milliseconds // small pause before starting again
       // then loop starts again automaticlaly
END FUNCTION

END PROGRAM
```

Structure chart:



For Desk checking we must confirm if our code would work in three scenarios.

1. Low BPM (<60)
2. Normal BPM (60-100)
3. High BPM (>100)
4. No BPM (0)

BPM Formula:
BPM = beats x 6000 /* 10000 = *beats x 6

Trial 1: low BPM

- Start program, betas = 0
- 8 beats counted in 5 seconds
- BPM = 8 x 6 = 48
- Print “BPM = 48”
- Orange LED ON, others OFF

Trial 2: Normal BPM

- Start program, beats = 0
- 12 beats counted in 5 seconds
- BPM = 12 x 6 =72
- Print “BPM = 72”
- Green LED ON, others OFF

Trial 3: High BPM

- Start program, beats = 0
- 20 beats counted in 5 seconds
- BPM = 20 x 6 = 120
- Print “BPM = 120”
- Red LED ON, others OFF

Trail 4: No BPM

- Start program beats = 0
- 0 beats counted in 5 seconds
- BPM = 0 x 6 = 0
- Print “No heartbeat detected
- Red LED BLINK every 1 seconds, Alert or check sensor

Desk Check Table

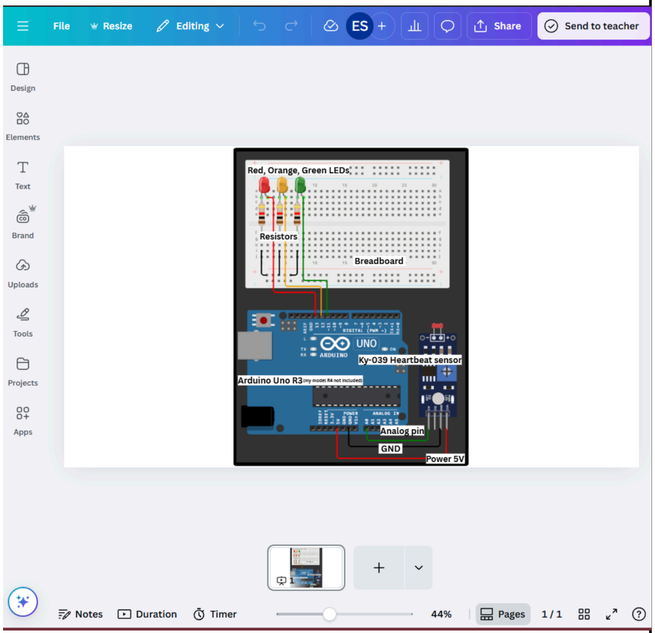
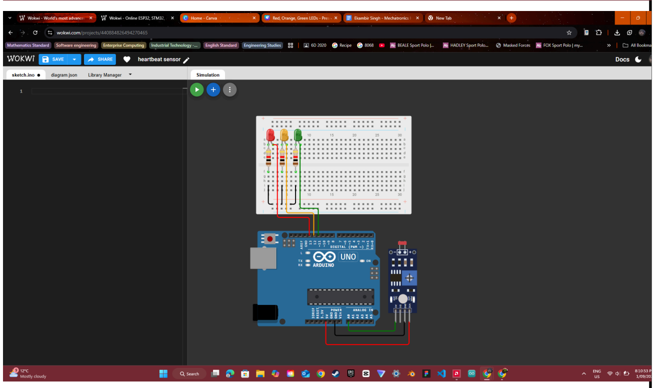
Beats in 5 seconds	**BPM = beats x 6**	**Result**	**LED ON**
8	48	BPM = 48	Orange (low)
12	72	BPM = 72	Green (Normal)
20	120	BPM = 120	Red (High)
0	0	No heartbeat detected.	Red LED BLINK every 1 (Alert or check sensor)

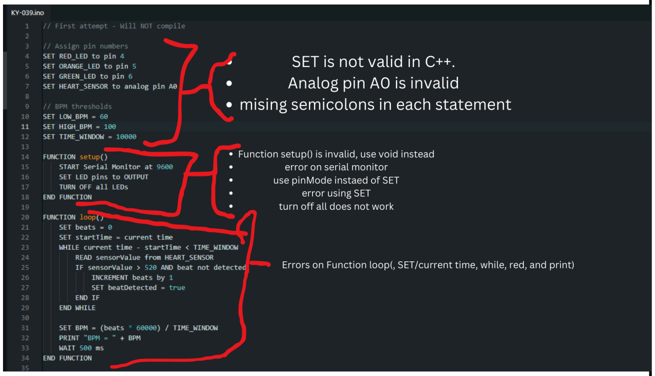
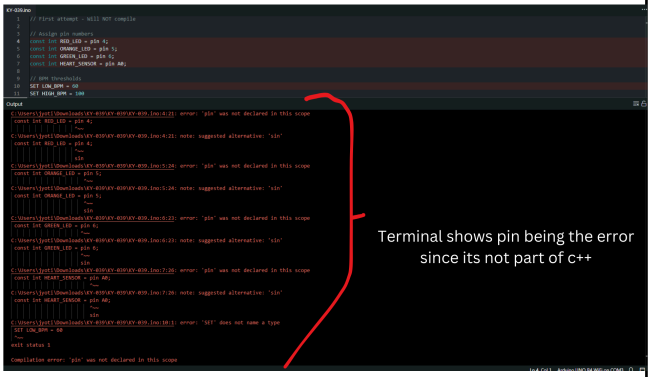
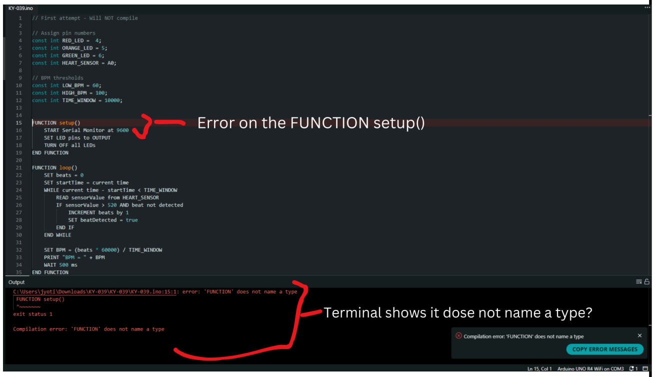
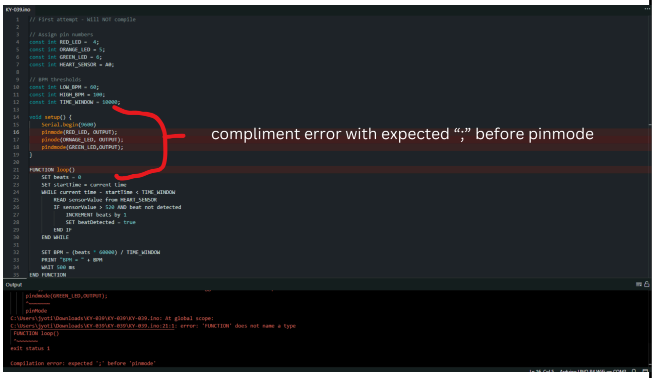
Data dictionary and requirements

Name	Properties	Use in project
RED_LED	Integer	Turns on when BPM is high or blinks rapidly when BPm is zero

ORANGE_LED	Integer	Turns on when BPM is low
GREEN_LED	Integer	Turns on when BPM is normal
HEART_SENSOR	Analog PIN	Along pin A0 reads heartbeat signal from the sensor
reader	Integer	Stores the raw analog data value from the heartbeat sensor to detect pulse peaks.
beatsInWindow	Integer	Counts the number of heartbeats detected during the 5 second time window
windowStart	Timestamp	Stores timestamp for the start of the 5 seconds and determines BPM calculating it and reset the beat count
WINDOW_MS	Integer	Defines the duration of the measurement of 5 second nad controls the BPM calculated and the LEDs
Avg BPM	Integer	Stores the calculated BPM and decides which LED light it should turn on.
LOW_BPM	Integer	Threshold for low heartrate
HIGH_BPM	Integer	Threshold for the high heartrate
blinkStart	TimeStamp	Flickers the RED LED when their is no beats (BPM = 0)
SerialMonitor	Interface	Displays real time heart rate reading and prints BPM stats
Loop()	Function	Loop's main purpose is to continuously read the sensor and repeat it.
setup()	Function	Set pin modes for LEDs. and start the serial monitor for output.

Development journal

Date	Task completed / undergoing process	Screenshot	Notes / Challenges
1/09/2025	- Researched about the KY-039 heartbeat sensor and system requirements - Designed and built a circuit and breadboard in Wokwi.	 File Resize Editing Share Send to teacher Design Elements Text Brand Uploads Tools Projects Apps Red, Orange, Green LEDs Resistors Breadboard Arduino Uno R3 Ky-039 Heartbeat sensor Analog pin GND Power 5V Notes Duration Timer 44% Pages 1 / 1 WOKWI SAVE SHARE heartbeat sensor Docs sketch.ino diagram.json Library Manager Simulation	- Creating the circuit diagram.
2/09/2025	- Designed on the pseudocode to outline how my system should function - Started creating the flowchart to show and understand the logic and		n/a

	decision making steps.		
5/09/2025	- Completed on flow charts and deskchecking to make sure the logic works. - Starting coding the heartbeat sensor in Arduino IDE	 SET is not valid in C++. • Analog pin A0 is invalid • mising semicolons in each statement • Function setup() is invalid, use void instead • error on serial monitor • use pinMode instaed of SET • error using SET • turn off all does not work Errors on Function loop(, SET/current time, while, red, and print)  Terminal shows pin being the error since its not part of c++  Error on the FUNCTION setup() Terminal shows it dose not name a type?  compliment error with expected “;” before pinmode	- Never used C++, so coding this was difficult. Wasn’t sure whether the code would work without showing the errors. - Still working on improving accuracy of heartbeat sensor

KY-039.ino

```
// First attempt - Will NOT compile

// Assign pin numbers
SET RED_LED to pin 4
SET ORANGE_LED to pin 5
SET GREEN_LED to pin 6
SET HEART_SENSOR to analog pin A0

// BPM thresholds
SET LOW_BPM = 60
SET HIGH_BPM = 100
SET TIME_WINDOW = 10000

FUNCTION setup()
    START Serial Monitor at 9600
    SET LED pins to OUTPUT
    TURN OFF all LEDs
END FUNCTION

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

KY-039.ino

```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED = pin 4;
const int ORANGE_LED = pin 5;
const int GREEN_LED = pin 6;
const int HEART_SENSOR = pin A0;

// BPM thresholds
SET LOW_BPM = 60
SET HIGH_BPM = 100
```

Output

```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:4:21: error: 'pin' was not declared in this scope
 const int RED_LED = pin 4;
                     ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:4:21: note: suggested alternative: 'sin'
 const int RED_LED = pin 4;
                     ^~~
                     sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:5:24: error: 'pin' was not declared in this scope
 const int ORANGE_LED = pin 5;
                        ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:5:24: note: suggested alternative: 'sin'
 const int ORANGE_LED = pin 5;
                        ^~~
                        sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:6:23: error: 'pin' was not declared in this scope
 const int GREEN_LED = pin 6;
                       ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:6:23: note: suggested alternative: 'sin'
 const int GREEN_LED = pin 6;
                       ^~~
                       sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:7:26: error: 'pin' was not declared in this scope
 const int HEART_SENSOR = pin A0;
                          ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:7:26: note: suggested alternative: 'sin'
 const int HEART_SENSOR = pin A0;
                          ^~~
                          sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:10:1: error: 'SET' does not name a type
 SET LOW_BPM = 60
 ^~~
exit status 1

Compilation error: 'pin' was not declared in this scope
```

Ln 4, Col 1   Arduino UNO R4 WiFi on COM3

KY-039.ino

```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED =  4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;
const int TIME_WINDOW = 10000;


FUNCTION setup()
    START Serial Monitor at 9600
    SET LED pins to OUTPUT
    TURN OFF all LEDs
END FUNCTION

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

Output

```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:15:1: error: 'FUNCTION' does not name a type
 FUNCTION setup()
 ^~~~~~~~
exit status 1

Compilation error: 'FUNCTION' does not name a type
```

Compilation error: 'FUNCTION' does not name a type

Ln 15, Col 1   Arduino UNO R4 WiFi on COM3

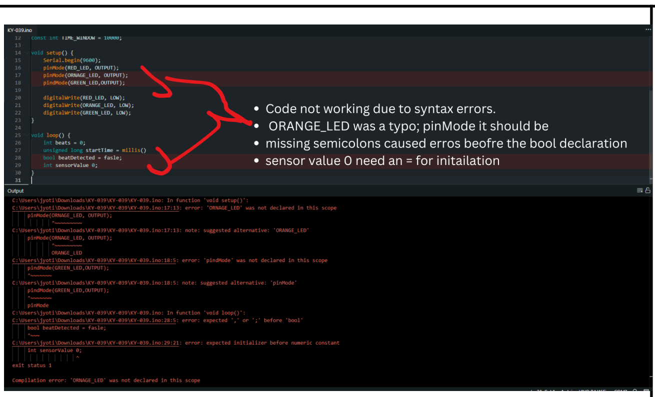
KY-039.ino

```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED =  4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;
const int TIME_WINDOW = 10000;

void setup() {
    Serial.begin(9600)
    pinmode(RED_LED, OUTPUT);
    pinode(ORNAGE_LED, OUTPUT);
    pindmode(GREEN_LED,OUTPUT);
}

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

Output

```
     pindmode(GREEN_LED,OUTPUT);
     ^~~~~~~~
     pinMode
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: At global scope:
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:21:1: error: 'FUNCTION' does not name a type
 FUNCTION loop()
 ^~~~~~~~
exit status 1

Compilation error: expected ';' before 'pinmode'
```

Ln 16, Col 5   Arduino UNO R4 WiFi on COM3

6/09/2025	- Coding my arduino and ensuring that it works and no bugs or glitches occur during testing.	 • Code not working due to syntax errors. • ORANGE_LED was a typo; pinMode it should be • missing semicolons caused erros beofre the bool declaration • sensor value 0 need an = for initailation	Challenges were the constant glitches and bugs occurring during the BPM and LED displaying wrong LEDs or not displaying a BPM at all. To fix this I checked the official page of the project to see whether it worked, so I then incorporated my version of the code and the sample version of the code, not copying but taking ideas and notes of it.

KY-039.ino

```
const int TIME_WINDOW = 10000;

void setup() {
    Serial.begin(9600);
    pinMode(RED_LED, OUTPUT);
    pinMode(ORNAGE_LED, OUTPUT);
    pindMode(GREEN_LED,OUTPUT);

    digitalWrite(RED_LED, LOW);
    digitalWrite(ORANGE_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
}

void loop() {
    int beats = 0;
    unsigned long startTime = millis()
    bool beatDetected = fasle;
    int sensorValue 0;
}
```

Output

```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: In function 'void setup()':
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:17:13: error: 'ORNAGE_LED' was not declared in this scope
   pinMode(ORNAGE_LED, OUTPUT);
           ^~~~~~~~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:17:13: note: suggested alternative: 'ORANGE_LED'
   pinMode(ORNAGE_LED, OUTPUT);
           ^~~~~~~~~~
           ORANGE_LED
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:18:5: error: 'pindMode' was not declared in this scope
   pindMode(GREEN_LED,OUTPUT);
   ^~~~~~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:18:5: note: suggested alternative: 'pinMode'
   pindMode(GREEN_LED,OUTPUT);
   ^~~~~~~~
   pinMode
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: In function 'void loop()':
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:28:5: error: expected ',' or ';' before 'bool'
   bool beatDetected = fasle;
   ^~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:29:21: error: expected initializer before numeric constant
   int sensorValue 0;
                   ^
exit status 1

Compilation error: 'ORNAGE_LED' was not declared in this scope
```

Error capturing

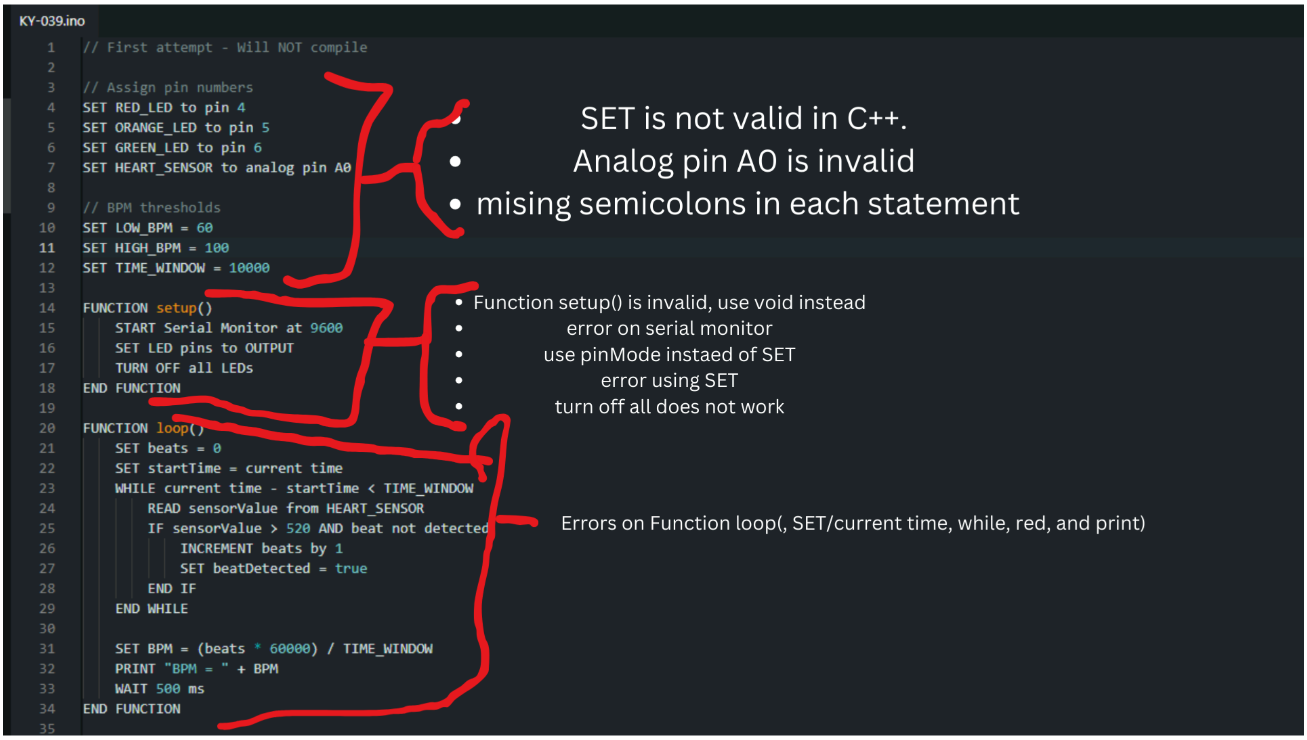
ERROR 1

Attempts: 1

Task: First attempt of coding heartbeat sensor

Goal: assign pins, set up BPM threshold, and start the loop and setup functions

Result: Fail



```
// First attempt - Will NOT compile

// Assign pin numbers
SET RED_LED to pin 4
SET ORANGE_LED to pin 5
SET GREEN_LED to pin 6
SET HEART_SENSOR to analog pin A0

// BPM thresholds
SET LOW_BPM = 60
SET HIGH_BPM = 100
SET TIME_WINDOW = 10000

FUNCTION setup()
    START Serial Monitor at 9600
    SET LED pins to OUTPUT
    TURN OFF all LEDs
END FUNCTION

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

- SET is not valid in C++.
- Analog pin A0 is invalid
- mising semicolons in each statement

- Function setup() is invalid, use void instead
- error on serial monitor
- use pinMode instaed of SET
- error using SET
- turn off all does not work

Errors on Function loop(, SET/current time, while, red, and print)

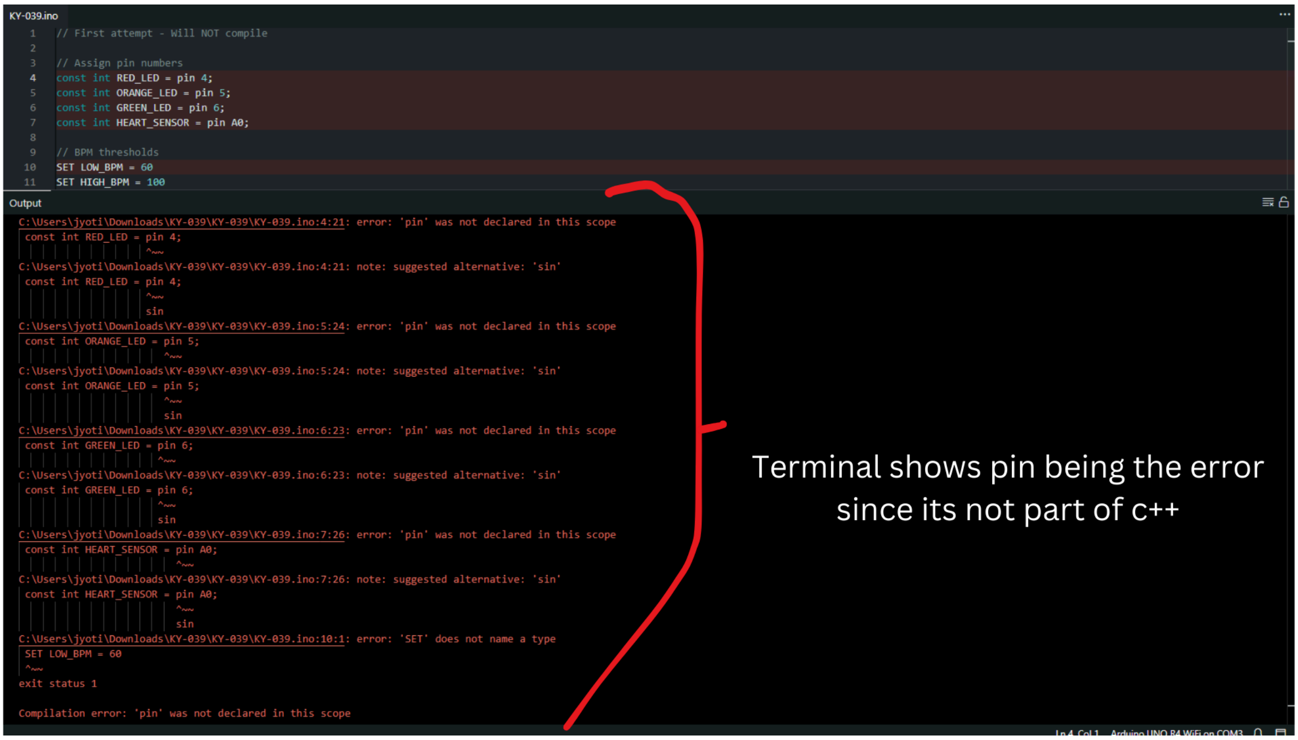
ERROR 2

Attempts: 2

Task: Second attempt of coding heart beat sensor

Goal: make section 1 of the code work and gradually work my way up

Result: Fail



```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED = pin 4;
const int ORANGE_LED = pin 5;
const int GREEN_LED = pin 6;
const int HEART_SENSOR = pin A0;

// BPM thresholds
SET LOW_BPM = 60
SET HIGH_BPM = 100
```

```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:4:21: error: 'pin' was not declared in this scope
 const int RED_LED = pin 4;
|                    ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:4:21: note: suggested alternative: 'sin'
 const int RED_LED = pin 4;
|                    ^~~
|                    sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:5:24: error: 'pin' was not declared in this scope
 const int ORANGE_LED = pin 5;
|                       ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:5:24: note: suggested alternative: 'sin'
 const int ORANGE_LED = pin 5;
|                       ^~~
|                       sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:6:23: error: 'pin' was not declared in this scope
 const int GREEN_LED = pin 6;
|                      ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:6:23: note: suggested alternative: 'sin'
 const int GREEN_LED = pin 6;
|                      ^~~
|                      sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:7:26: error: 'pin' was not declared in this scope
 const int HEART_SENSOR = pin A0;
|                         ^~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:7:26: note: suggested alternative: 'sin'
 const int HEART_SENSOR = pin A0;
|                         ^~~
|                         sin
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:10:1: error: 'SET' does not name a type
 SET LOW_BPM = 60
 ^~~
exit status 1

Compilation error: 'pin' was not declared in this scope
```

Terminal shows pin being the error since its not part of c++

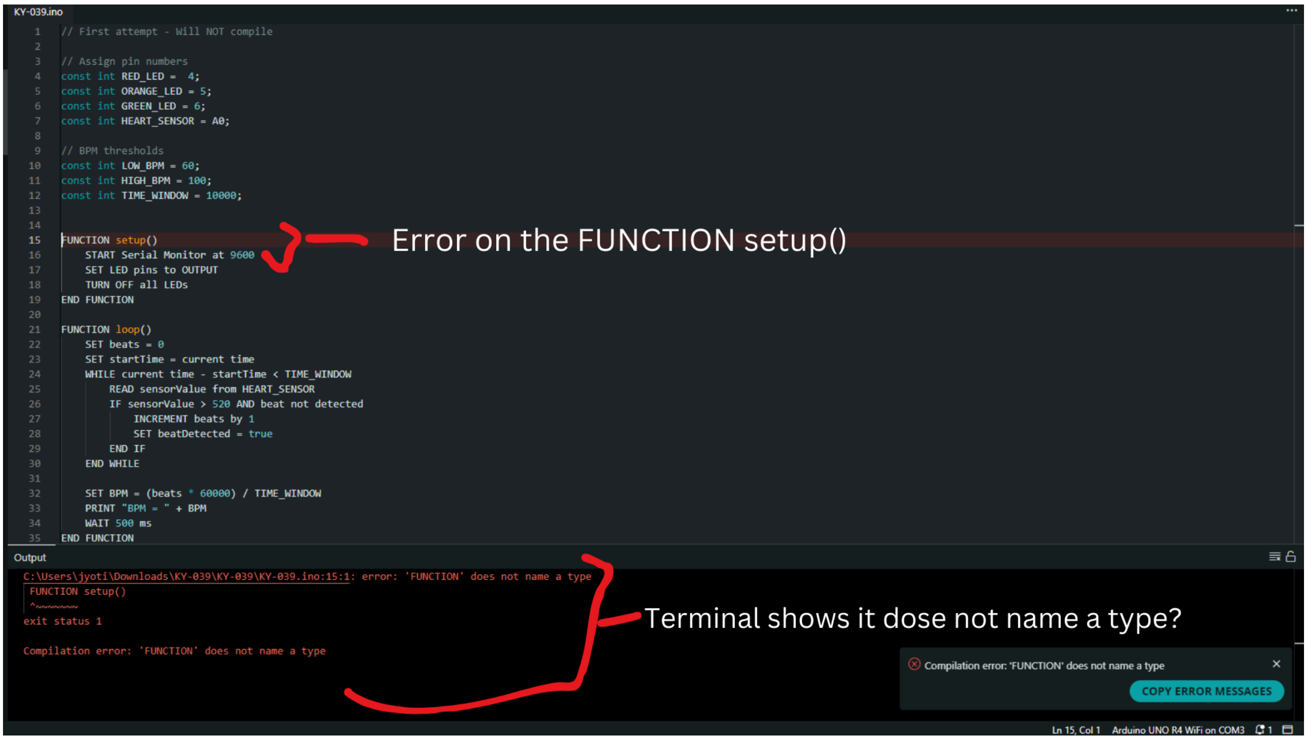
ERROR 3

Attempts: 3

Task: third attempt of coding heart beat sensor

Goal: fixing section 1 to cause no errors on my work

Result: success but fail on the rest of the code



```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED =  4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;
const int TIME_WINDOW = 10000;


FUNCTION setup()
    START Serial Monitor at 9600
    SET LED pins to OUTPUT
    TURN OFF all LEDs
END FUNCTION

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

Error on the FUNCTION setup()

```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:15:1: error: 'FUNCTION' does not name a type
 FUNCTION setup()
 ^~~~~~~~
exit status 1

Compilation error: 'FUNCTION' does not name a type
```

Terminal shows it dose not name a type?

Compilation error: 'FUNCTION' does not name a type

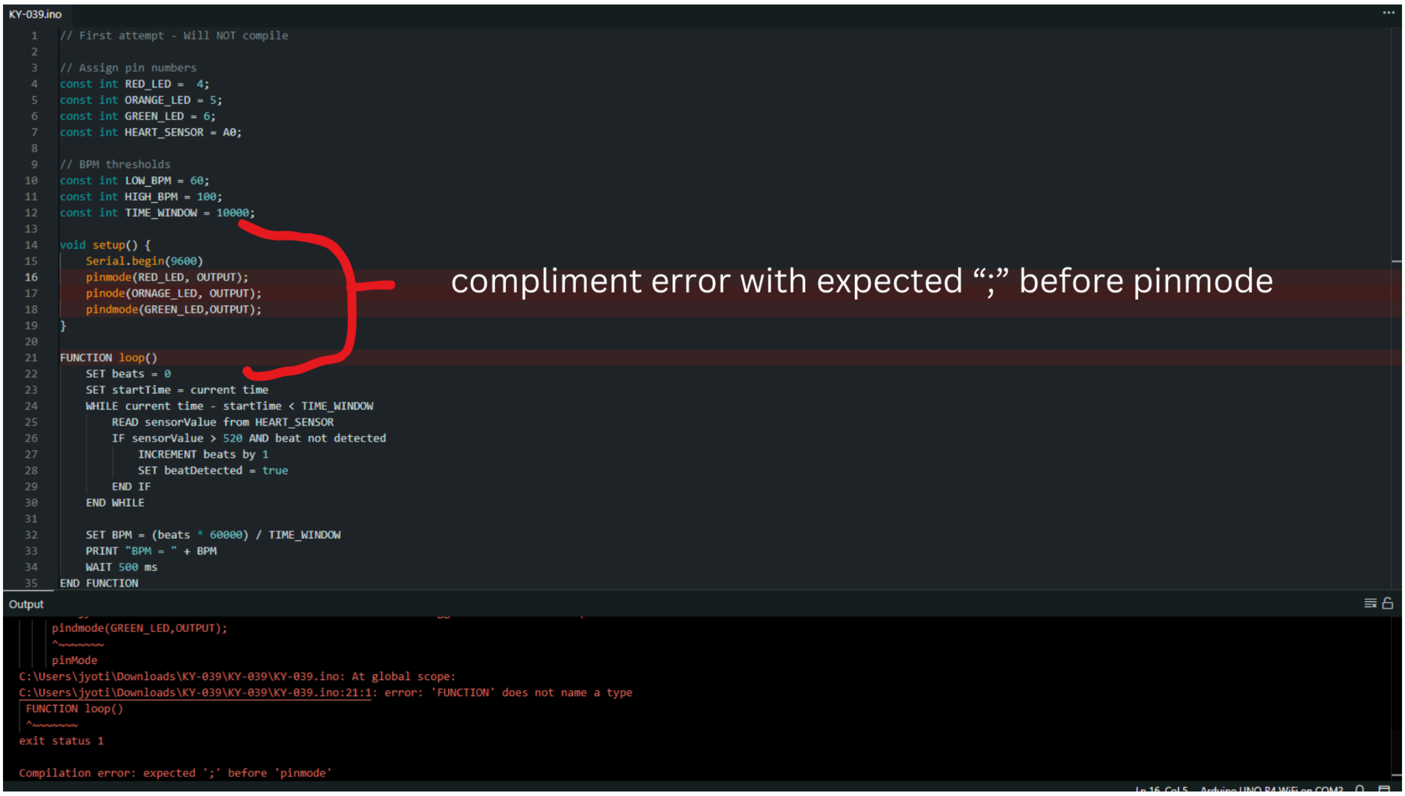
ERROR 4

Attempt 4:

Task: making the code work in section 2 of the fourth attempt

Goal: to make sure no errors in the code and work step by step

Result: Fail



```
// First attempt - Will NOT compile

// Assign pin numbers
const int RED_LED =  4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;
const int TIME_WINDOW = 10000;

void setup() {
    Serial.begin(9600)
    pinmode(RED_LED, OUTPUT);
    pinode(ORNAGE_LED, OUTPUT);
    pindmode(GREEN_LED,OUTPUT);
}

FUNCTION loop()
    SET beats = 0
    SET startTime = current time
    WHILE current time - startTime < TIME_WINDOW
        READ sensorValue from HEART_SENSOR
        IF sensorValue > 520 AND beat not detected
            INCREMENT beats by 1
            SET beatDetected = true
        END IF
    END WHILE

    SET BPM = (beats * 60000) / TIME_WINDOW
    PRINT "BPM = " + BPM
    WAIT 500 ms
END FUNCTION
```

compliment error with expected “;” before pinmode

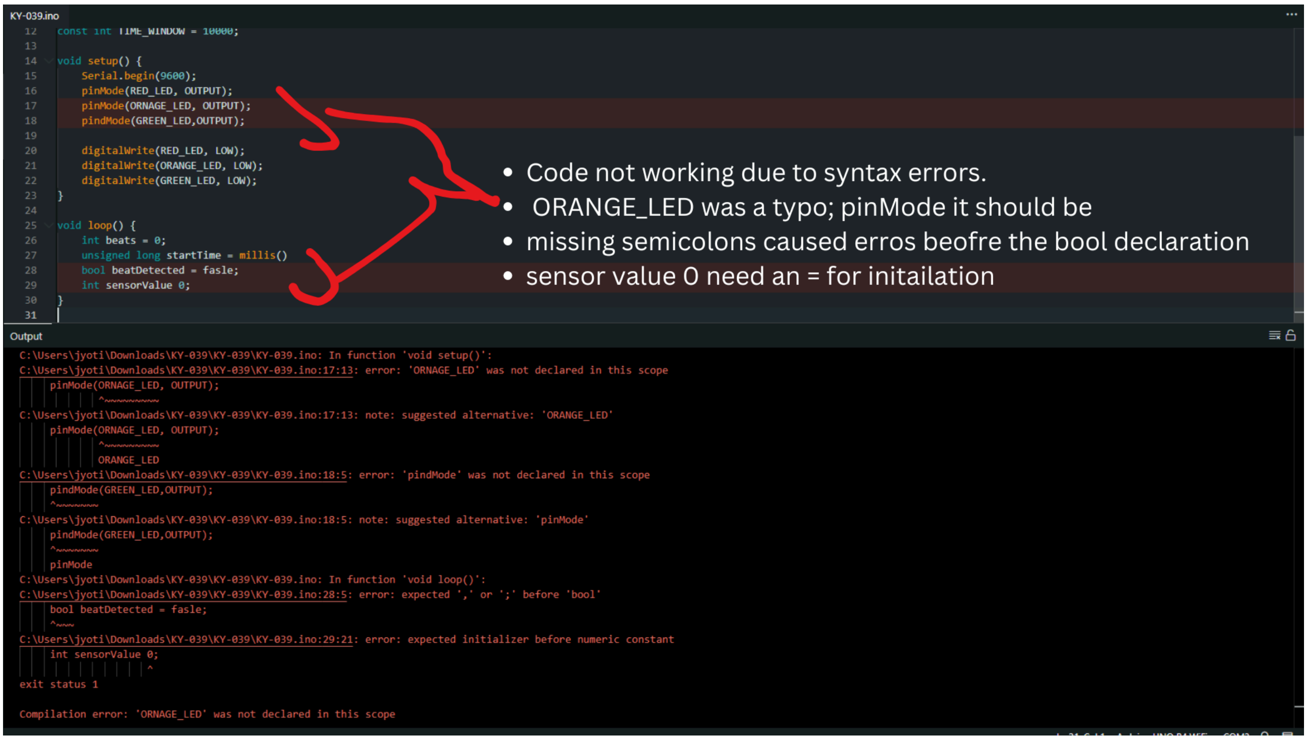
```
      pindmode(GREEN_LED,OUTPUT);
      ^~~~~~~~
      pinMode
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: At global scope:
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:21:1: error: 'FUNCTION' does not name a type
 FUNCTION loop()
 ^~~~~~~~
exit status 1

Compilation error: expected ';' before 'pinmode'
```

Attempt 5:

Task: refining the code and erasing old data

Goal: Make the arduino have no errors and verify to check

Result: Fail



```
const int TIME_WINDOW = 10000;

void setup() {
    Serial.begin(9600);
    pinMode(RED_LED, OUTPUT);
    pinMode(ORNAGE_LED, OUTPUT);
    pindMode(GREEN_LED,OUTPUT);

    digitalWrite(RED_LED, LOW);
    digitalWrite(ORANGE_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
}

void loop() {
    int beats = 0;
    unsigned long startTime = millis()
    bool beatDetected = fasle;
    int sensorValue 0;
}
```

- Code not working due to syntax errors.
- ORANGE_LED was a typo; pinMode it should be
- missing semicolons caused erros beofre the bool declaration
- sensor value 0 need an = for initailation

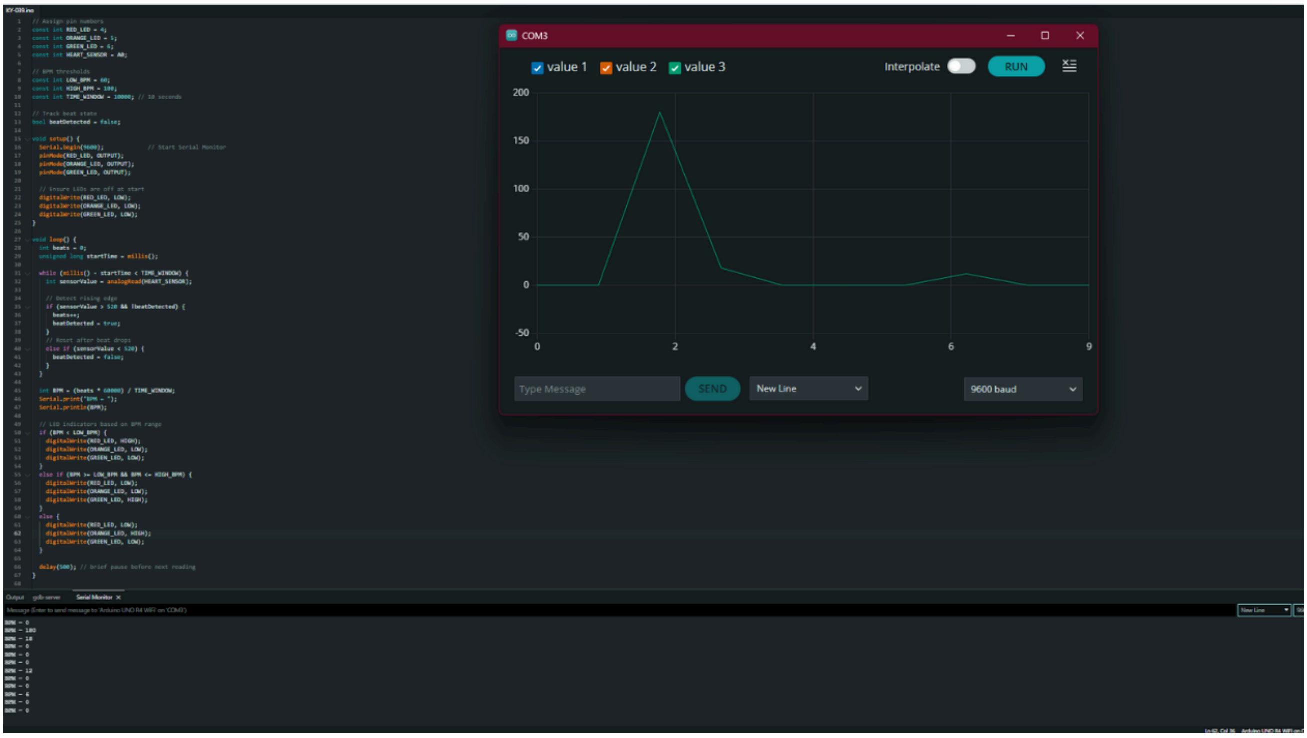
```
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: In function 'void setup()':
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:17:13: error: 'ORNAGE_LED' was not declared in this scope
      pinMode(ORNAGE_LED, OUTPUT);
              ^~~~~~~~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:17:13: note: suggested alternative: 'ORANGE_LED'
      pinMode(ORNAGE_LED, OUTPUT);
              ^~~~~~~~~~
              ORANGE_LED
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:18:5: error: 'pindMode' was not declared in this scope
      pindMode(GREEN_LED,OUTPUT);
      ^~~~~~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:18:5: note: suggested alternative: 'pinMode'
      pindMode(GREEN_LED,OUTPUT);
      ^~~~~~~~
      pinMode
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino: In function 'void loop()':
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:28:5: error: expected ',' or ';' before 'bool'
      bool beatDetected = fasle;
      ^~~~
C:\Users\jyoti\Downloads\KY-039\KY-039\KY-039.ino:29:21: error: expected initializer before numeric constant
      int sensorValue 0;
                      ^
exit status 1

Compilation error: 'ORNAGE_LED' was not declared in this scope
```

Attempt 6:

Task: Fix all the errors of the code

Goal: To check if the BPM status work

Result: success, although it worked the reading was not displaying accurately and was only showing 0 BPM even when the finger was placed. For this to fix I used Projecthub arduino and took a look at their code, tested it and that worked meaning my system can work, so then I integrated some parts of the code to mine to work.



```
// Assign pin numbers
const int RED_LED = 4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;
const int TIME_WINDOW = 10000; // 10 seconds

// Track beat state
bool beatDetected = false;

void setup() {
  Serial.begin(9600);             // Start Serial Monitor
  pinMode(RED_LED, OUTPUT);
  pinMode(ORANGE_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  // Ensure LEDs are off at start
  digitalWrite(RED_LED, LOW);
  digitalWrite(ORANGE_LED, LOW);
  digitalWrite(GREEN_LED, LOW);
}

void loop() {
  int beats = 0;
  unsigned long startTime = millis();

  while (millis() - startTime < TIME_WINDOW) {
    int sensorValue = analogRead(HEART_SENSOR);

    // Detect rising edge
    if (sensorValue > 520 && !beatDetected) {
      beats++;
      beatDetected = true;
    }
    // Reset after beat drops
    else if (sensorValue < 520) {
      beatDetected = false;
    }
  }

  int BPM = (beats * 60000) / TIME_WINDOW;
  Serial.print("BPM = ");
  Serial.println(BPM);

  // LED indicators based on BPM range
  if (BPM < LOW_BPM) {
    digitalWrite(RED_LED, HIGH);
    digitalWrite(ORANGE_LED, LOW);
    digitalWrite(GREEN_LED, LOW);
  }
  else if (BPM >= LOW_BPM && BPM <= HIGH_BPM) {
    digitalWrite(RED_LED, LOW);
    digitalWrite(ORANGE_LED, LOW);
    digitalWrite(GREEN_LED, HIGH);
  }
  else {
    digitalWrite(RED_LED, LOW);
    digitalWrite(ORANGE_LED, HIGH);
    digitalWrite(GREEN_LED, LOW);
  }

  delay(500); // brief pause before next reading
}
```

Message (Enter to send message to 'Arduino UNO R4 WiFi' on 'COM3')

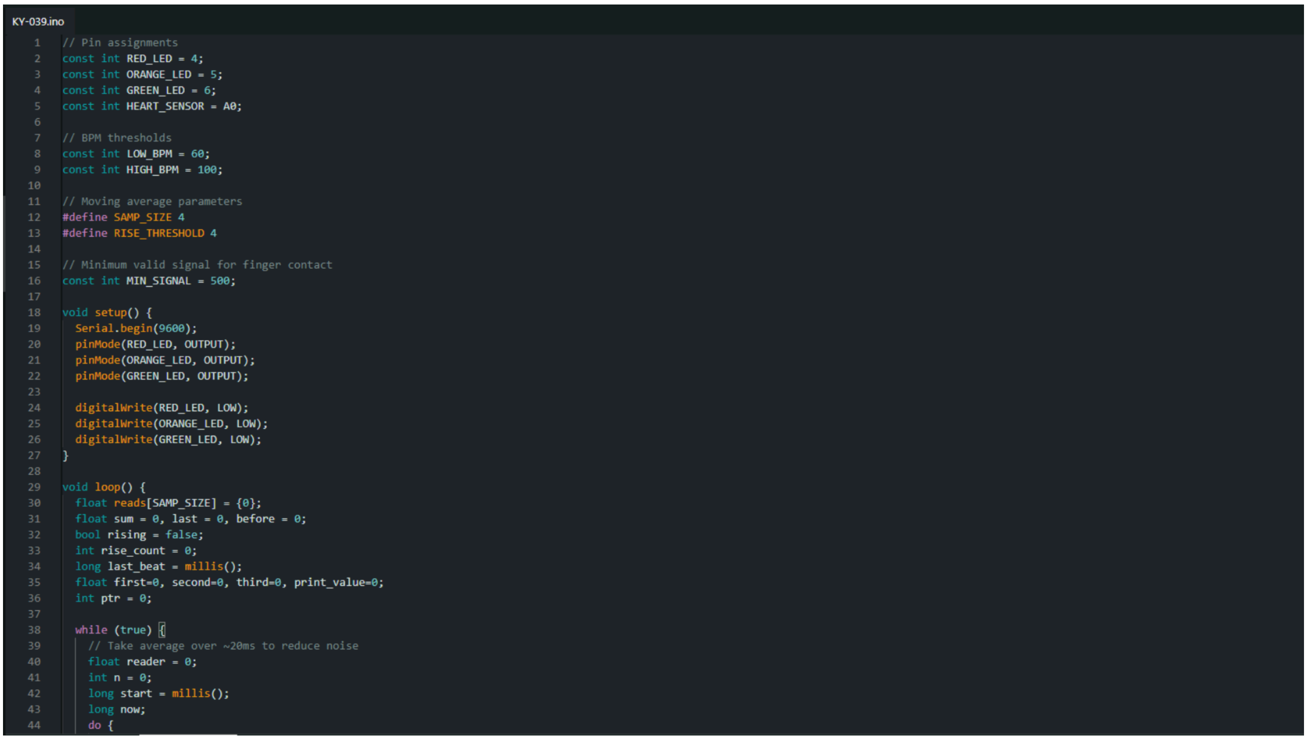
```
BPM = 0
BPM = 180
BPM = 18
BPM = 0
BPM = 0
BPM = 0
BPM = 12
BPM = 0
BPM = 0
BPM = 6
BPM = 0
BPM = 0
```

Attempt 7:

Task: To test if the code finally works

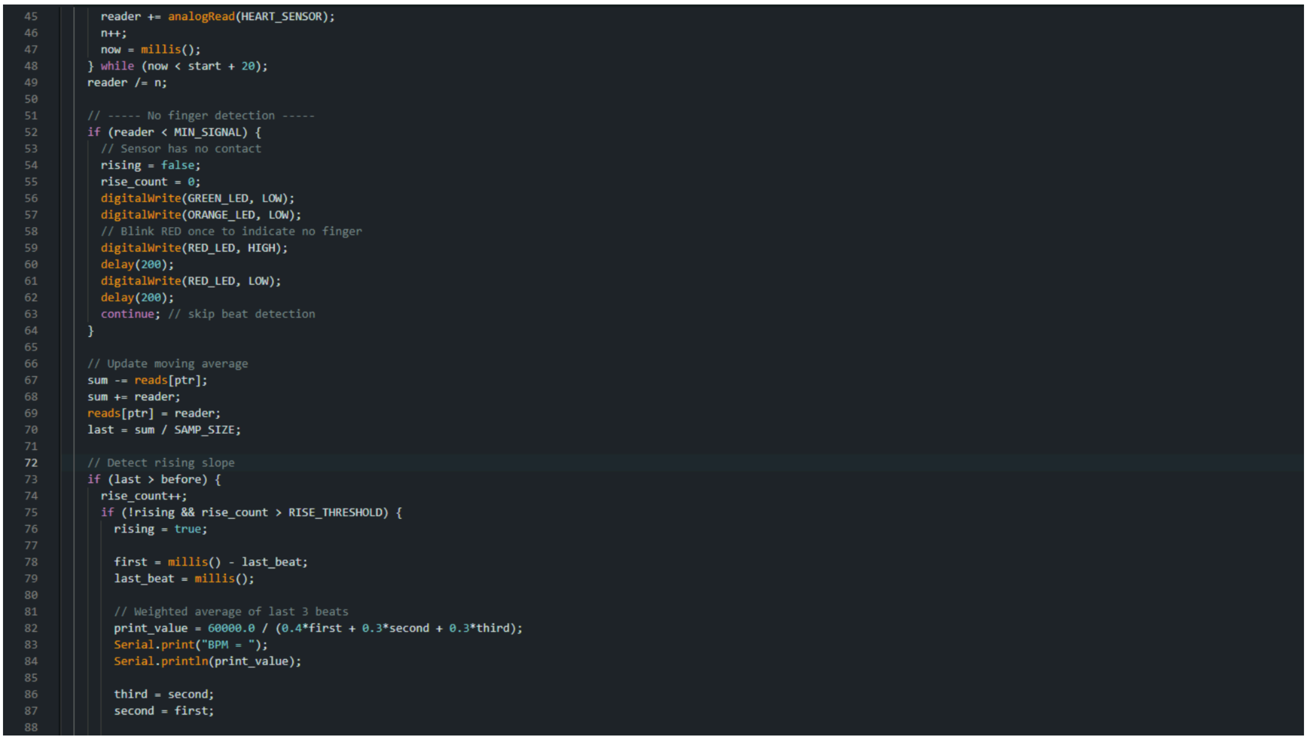
Goal: Make sure the whole project work

Result: success but the BPM wasn't being reliable and was just no being accurate

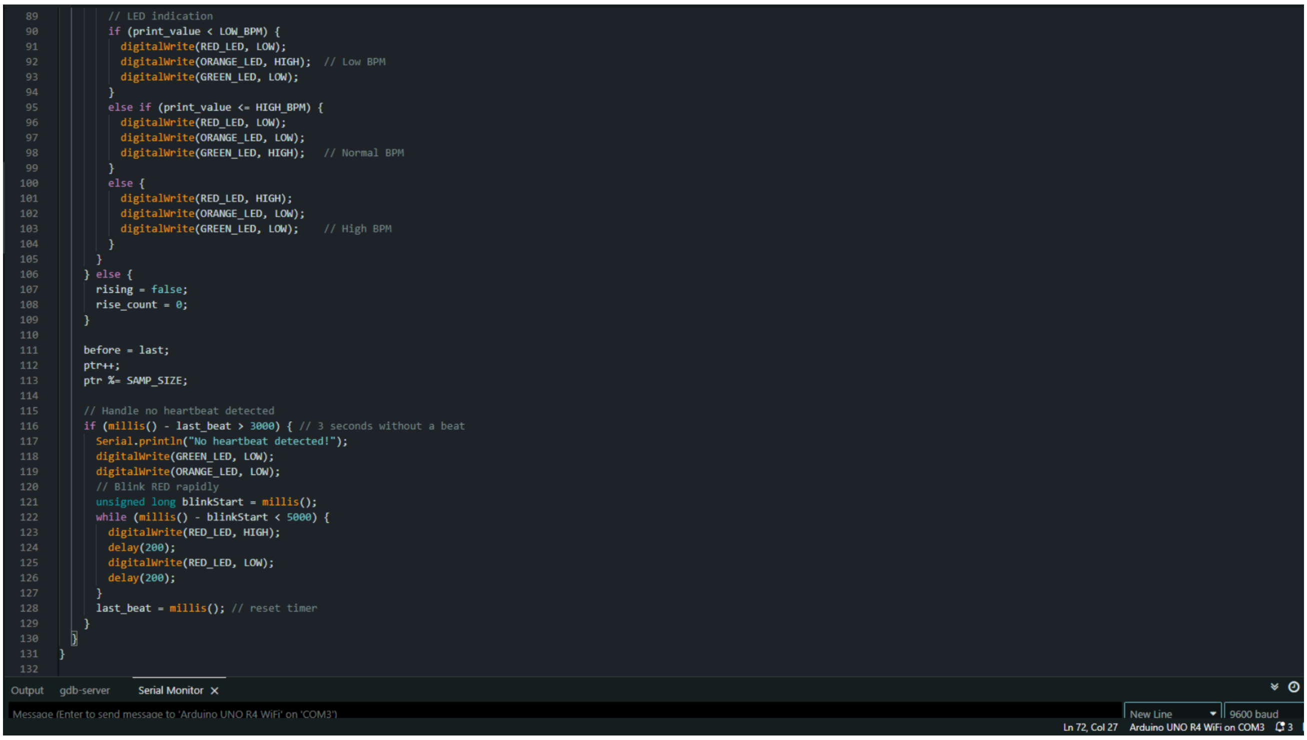


KY-039.ino

```
// Pin assignments
const int RED_LED = 4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

// BPM thresholds
const int LOW_BPM = 60;
const int HIGH_BPM = 100;

// Moving average parameters
#define SAMP_SIZE 4
#define RISE_THRESHOLD 4

// Minimum valid signal for finger contact
const int MIN_SIGNAL = 500;

void setup() {
  Serial.begin(9600);
  pinMode(RED_LED, OUTPUT);
  pinMode(ORANGE_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  digitalWrite(RED_LED, LOW);
  digitalWrite(ORANGE_LED, LOW);
  digitalWrite(GREEN_LED, LOW);
}

void loop() {
  float reads[SAMP_SIZE] = {0};
  float sum = 0, last = 0, before = 0;
  bool rising = false;
  int rise_count = 0;
  long last_beat = millis();
  float first=0, second=0, third=0, print_value=0;
  int ptr = 0;

  while (true) {
    // Take average over ~20ms to reduce noise
    float reader = 0;
    int n = 0;
    long start = millis();
    long now;
    do {
```



```
      reader += analogRead(HEART_SENSOR);
      n++;
      now = millis();
    } while (now < start + 20);
    reader /= n;

    // ----- No finger detection -----
    if (reader < MIN_SIGNAL) {
      // Sensor has no contact
      rising = false;
      rise_count = 0;
      digitalWrite(GREEN_LED, LOW);
      digitalWrite(ORANGE_LED, LOW);
      // Blink RED once to indicate no finger
      digitalWrite(RED_LED, HIGH);
      delay(200);
      digitalWrite(RED_LED, LOW);
      delay(200);
      continue; // skip beat detection
    }

    // Update moving average
    sum -= reads[ptr];
    sum += reader;
    reads[ptr] = reader;
    last = sum / SAMP_SIZE;

    // Detect rising slope
    if (last > before) {
      rise_count++;
      if (!rising && rise_count > RISE_THRESHOLD) {
        rising = true;

        first = millis() - last_beat;
        last_beat = millis();

        // Weighted average of last 3 beats
        print_value = 60000.0 / (0.4*first + 0.3*second + 0.3*third);
        Serial.print("BPM = ");
        Serial.println(print_value);

        third = second;
        second = first;

```



```
        // LED indication
        if (print_value < LOW_BPM) {
          digitalWrite(RED_LED, LOW);
          digitalWrite(ORANGE_LED, HIGH);  // Low BPM
          digitalWrite(GREEN_LED, LOW);
        }
        else if (print_value <= HIGH_BPM) {
          digitalWrite(RED_LED, LOW);
          digitalWrite(ORANGE_LED, LOW);
          digitalWrite(GREEN_LED, HIGH);   // Normal BPM
        }
        else {
          digitalWrite(RED_LED, HIGH);
          digitalWrite(ORANGE_LED, LOW);
          digitalWrite(GREEN_LED, LOW);    // High BPM
        }
      }
    } else {
      rising = false;
      rise_count = 0;
    }

    before = last;
    ptr++;
    ptr %= SAMP_SIZE;

    // Handle no heartbeat detected
    if (millis() - last_beat > 3000) { // 3 seconds without a beat
      Serial.println("No heartbeat detected!");
      digitalWrite(GREEN_LED, LOW);
      digitalWrite(ORANGE_LED, LOW);
      // Blink RED rapidly
      unsigned long blinkStart = millis();
      while (millis() - blinkStart < 5000) {
        digitalWrite(RED_LED, HIGH);
        delay(200);
        digitalWrite(RED_LED, LOW);
        delay(200);
      }
      last_beat = millis(); // reset timer
    }
  }
}
```

Output
gdb-server
Serial Monitor ×
Message (Enter to send message to 'Arduino UNO R4 WiFi' on 'COM3')
New Line
9600 baud
Ln 72, Col 27
Arduino UNO R4 WiFi on COM3
3

Attempt 8:

Task: recheck the code and find glitches or bugs that might cause this.

Goal: Make sure all the led work in their specific order by the BPM

Result: Success the code is all working and the code is finished and the given code is now below.

```
const int RED_LED = 4;
const int ORANGE_LED = 5;
const int GREEN_LED = 6;
const int HEART_SENSOR = A0;

const int LOW_BPM = 60;
const int HIGH_BPM = 100;

#define SAMP_SIZE 4
#define RISE_THRESHOLD 4

const int MIN_SIGNAL = 500;
const unsigned long WINDOW_MS = 5000UL; // 5-second window

void setup() {
  Serial.begin(9600);
  pinMode(RED_LED, OUTPUT);
  pinMode(ORANGE_LED, OUTPUT);
  pinMode(GREEN_LED, OUTPUT);

  digitalWrite(RED_LED, LOW);
  digitalWrite(ORANGE_LED, LOW);
  digitalWrite(GREEN_LED, LOW);
}

void loop() {
  float reads[SAMP_SIZE] = {0};
  float sum = 0, last = 0, before = 0;
```

```
bool rising = false;
int rise_count = 0;
int ptr = 0;

unsigned long last_beat = millis();
unsigned long windowStart = millis();
int beatsInWindow = 0;

while (true) {
  float reader = 0.0;
  int n = 0;
  unsigned long start = millis();
  do {
    reader += analogRead(HEART_SENSOR);
    n++;
  } while (millis() < start + 20);
  reader /= (float)n;

  if (reader < MIN_SIGNAL) {
    rising = false;
    rise_count = 0;
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(ORANGE_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    delay(120);
    digitalWrite(RED_LED, LOW);
    delay(120);
  } else {
    sum -= reads[ptr];

    sum += reader;

    reads[ptr] = reader;

    last = sum / SAMP_SIZE;


    if (last > before) {

      rise_count++;

      if (!rising && rise_count > RISE_THRESHOLD) {

        rising = true;

        last_beat = millis();

        beatsInWindow++;

      }

    } else {

      rising = false;

      rise_count = 0;

    }


    before = last;

    ptr++;

    ptr %= SAMP_SIZE;

  }


  if (millis() - windowStart >= WINDOW_MS) {

    float avgBPM = beatsInWindow * 6.0; // 5s -> multiply by 6 for BPM

    Serial.print("BPM = ");

    Serial.println(avgBPM);


    if (beatsInWindow == 0) {
```

```
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(ORANGE_LED, LOW);
    unsigned long blinkStart = millis();
    while (millis() - blinkStart < 1000) {
      digitalWrite(RED_LED, HIGH);
      delay(200);
      digitalWrite(RED_LED, LOW);
      delay(200);
    }
  } else {
    if (avgBPM < LOW_BPM) {
      digitalWrite(RED_LED, LOW);
      digitalWrite(ORANGE_LED, HIGH);
      digitalWrite(GREEN_LED, LOW);
    } else if (avgBPM <= HIGH_BPM) {
      digitalWrite(RED_LED, LOW);
      digitalWrite(ORANGE_LED, LOW);
      digitalWrite(GREEN_LED, HIGH);
    } else {
      digitalWrite(RED_LED, HIGH);
      digitalWrite(ORANGE_LED, LOW);
      digitalWrite(GREEN_LED, LOW);
    }
  }

  beatsInWindow = 0;
  windowStart = millis();
  before = 0;
```

```
    }

    if (millis() - last_beat > WINDOW_MS) {
      digitalWrite(GREEN_LED, LOW);
      digitalWrite(ORANGE_LED, LOW);
      unsigned long blinkStart = millis();
      while (millis() - blinkStart < 800) {
        digitalWrite(RED_LED, HIGH);
        delay(150);
        digitalWrite(RED_LED, LOW);
        delay(150);
      }
      last_beat = millis();
      windowStart = millis();
      beatsInWindow = 0;
    }
  }
}
```

Result: success as my code finally works and with no errors displaying meaning my code finally works the way it should have been working.

Evaluations

At the end the heartbeat sensor was a success and records the heartbeats using hte KY-039 sensor and Arduino Uno R4 system. The results were displayed on the Serial monitor in real time every 5 seconds, while the LEDs provide a visual contrast showing feedback whether its low, normal, and high rate. The system to detect pulses and calculate them in PBM, showing the effective application skill and the structured programming techniques.

This project is well addressed for its need of low cost, but accessible biometric monitoring device in an education system. Testing the various scenarios of the low, normal, high, and zero BPM made sure my code worked and correctly processed data and triggered the LED based on the BPM displayed. Initially it was a challenging project with the inaccurate BPM reading or LED misbehaviour, this was resolved through testing, debugging, and refining code constantly.

Overall with the desk checking and detailed journal of the project, the solution is a safe, non-invasive and ethical approach. Since it does not store sensitive personal data, this project is effectively demonstrating real time data and processing it every 5 seconds showing the reliability of the output given.

Bibliography

- Joy-it.net. (2025). KY-039 Heartbeat sensor | Joy-IT. [online] Available at: https://joy-it.net/en/products/sen-ky039hs#:~:text=The%20KY%2D039%20heartbeat%20sensor,wearables%20or%20DIY%20healthcare%20technology. [Accessed 1 Sep. 2025].
- Phipps Electronics (2021). *Using the Heart Beat Sensor KY-039 with Arduino - Phipps Electronics*. [online] Phipps Electronics. Available at: https://www.phippselectronics.com/using-the-heart-beat-sensor-ky-039-with-arduino/?srsltid=AfmBOop3uhtkNX0HaXZ6n7gJaAqMptfUetWPVCrjlqI1j2XYPFzLZq28 [Accessed 1 Sep. 2025].
- Wokwi.com. (2019). *Wokwi - World's most advanced ESP32 Simulator*. [online] Available at: https://wokwi.com/ [Accessed 1 Sep. 2025].
- Canva. (2025). Available at: https://www.canva.com/ [Accessed 1 Sep. 2025].
- Newark, F. (2023). *Resistor Colour Code Calculator | element14 Australia*. [online] Element14.com. Available at: https://au.element14.com/resistor-colour-code-calculator [Accessed 2 Sep. 2025].
- Arduino Project Hub. (2018). *From KY-039 To Heart Rate*. [online] Available at: https://projecthub.arduino.cc/Johan_Ha/from-ky-039-to-heart-rate-8c660b#section5 [Accessed Sep. 2025].