UNIVERSITY OF BOHOL
COLLEGE of ENGINEERING, TECHNOLOGY
ARCHITECTURE, and FINE ARTS
Computer Engineering Department

Second Semester

HEARTBEAT SENSOR USING ATMEGA328 ASSEMBLY LANGUAGE

In partial fulfillment of the requirements for the subject

CPEP 322 – Computer Architecture & Organization Lec/Lab

Submitted to:

Engr. Liduvina L. Namocatcat, CPE

Instructor

Submitted by:

Taguran, John Claidy Ken O.

June 2024

Introduction

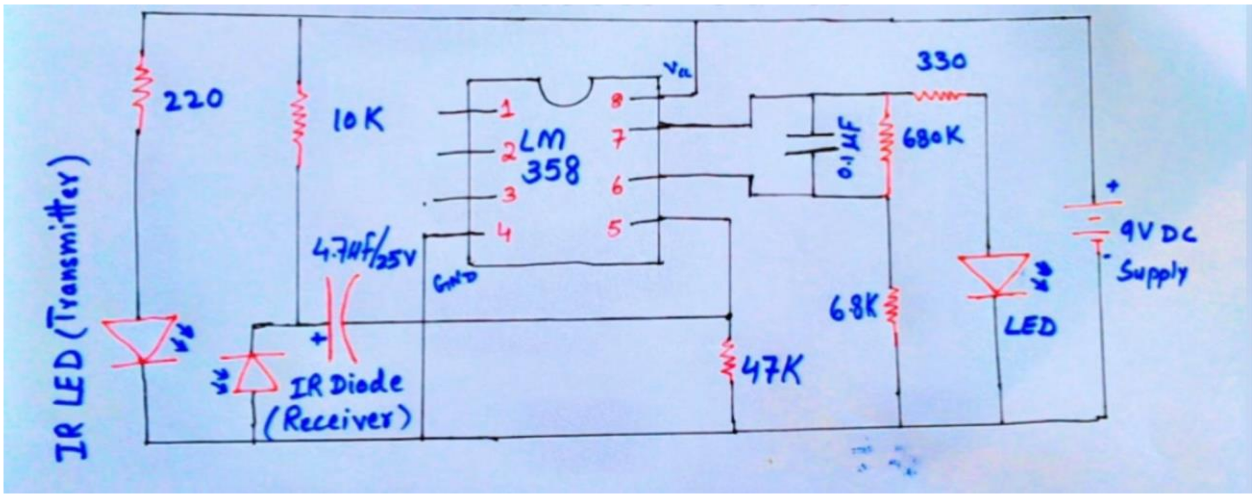
In the realm of modern electronics and embedded systems, heartbeat monitoring is crucial for numerous applications. Accurate and reliable heart rate sensing is essential for ensuring safety, efficiency, and optimal performance of devices and systems. This project focuses on the development and implementation of a heartbeat sensor using the ATMEGA328 microcontroller, programmed in assembly language.

The ATMEGA328 microcontroller, a core component of the popular Arduino platform, offers a robust and versatile solution for embedded system applications. Its features, including a rich set of I/O ports, built-in analog-to-digital converters (ADCs), and efficient power management, make it an ideal choice for implementing a precise and responsive heartbeat sensing project.

The primary objective of this project is to use the ATMEGA328 microcontroller and assembly language to create a heartbeat sensor that can accurately detect heart rate changes and control an external device based on the sensed data. The system utilizes a photoplethysmogram (PPG) sensor to measure heart rate variations. The PPG sensor detects blood volume changes in the microvascular bed of tissue, allowing the microcontroller to convert these changes into corresponding heart rate readings.

This project is distinctive in its use of assembly language for programming the ATMEGA328 microcontroller. This approach enables the optimization of code for speed and memory usage, which is crucial for real-time applications.**Logic**

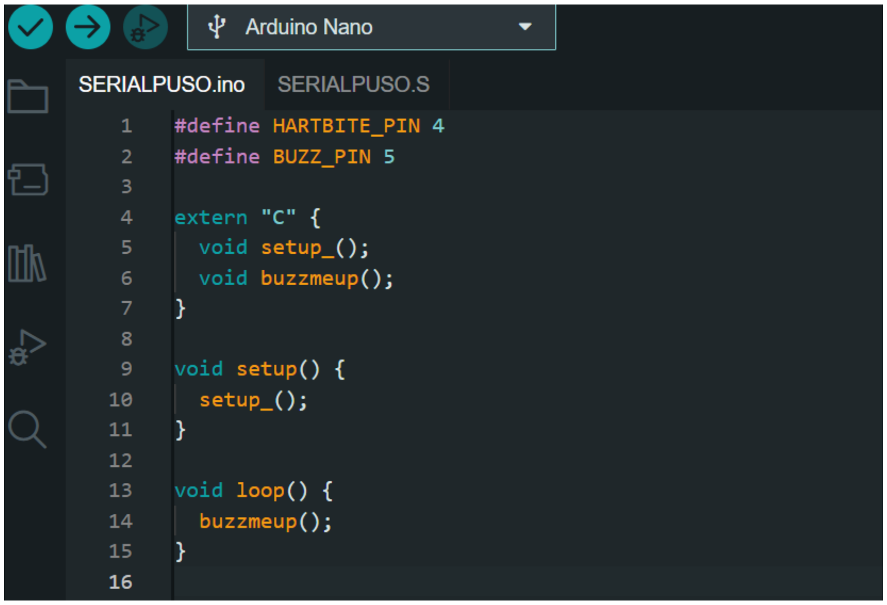
LOGIC DIAGRAM



CODE

.ino

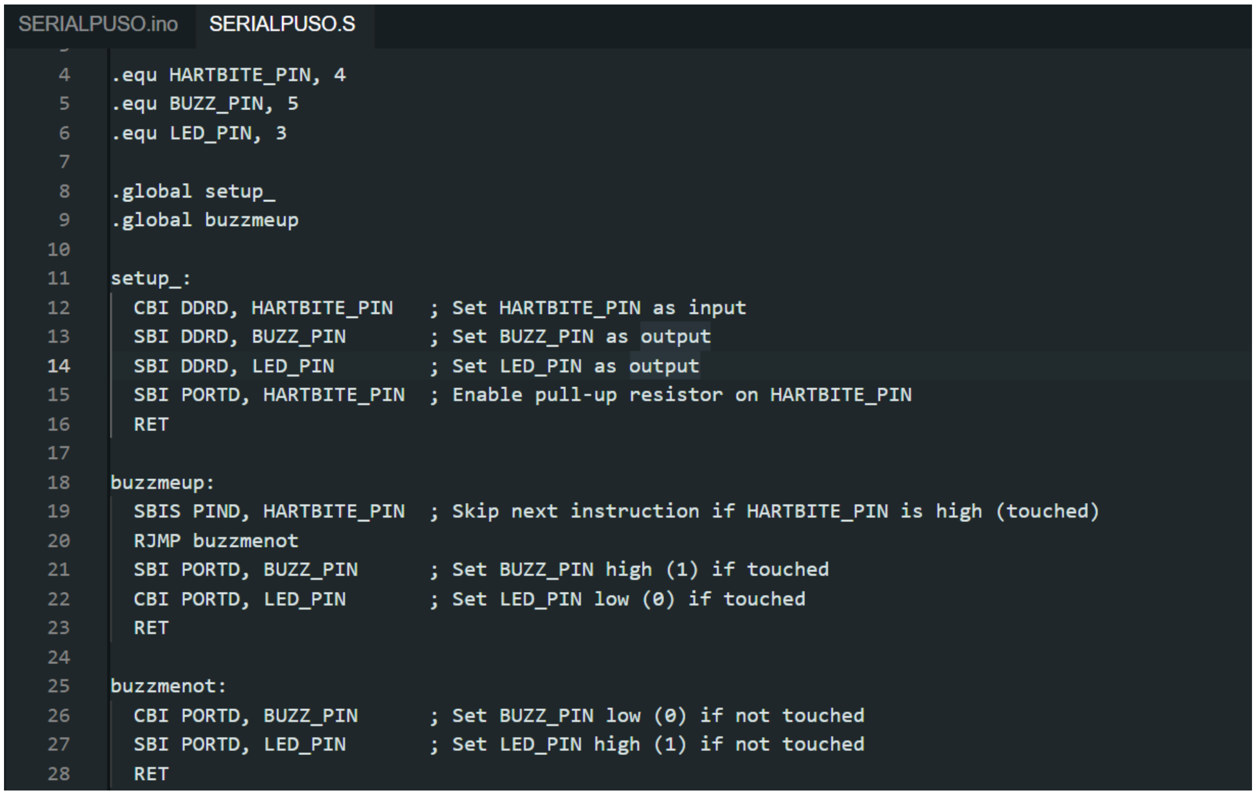
```
#define HARTBITE_PIN 4
#define BUZZ_PIN 5

extern "C" {
void setup_();
void buzzmeup();
}

void setup() {
setup_();
}

void loop() {
buzzmeup();
}
```



```
#define HARTBITE_PIN 4
#define BUZZ_PIN 5

extern "C" {
  void setup_();
  void buzzmeup();
}

void setup() {
  setup_();
}

void loop() {
  buzzmeup();
}

```

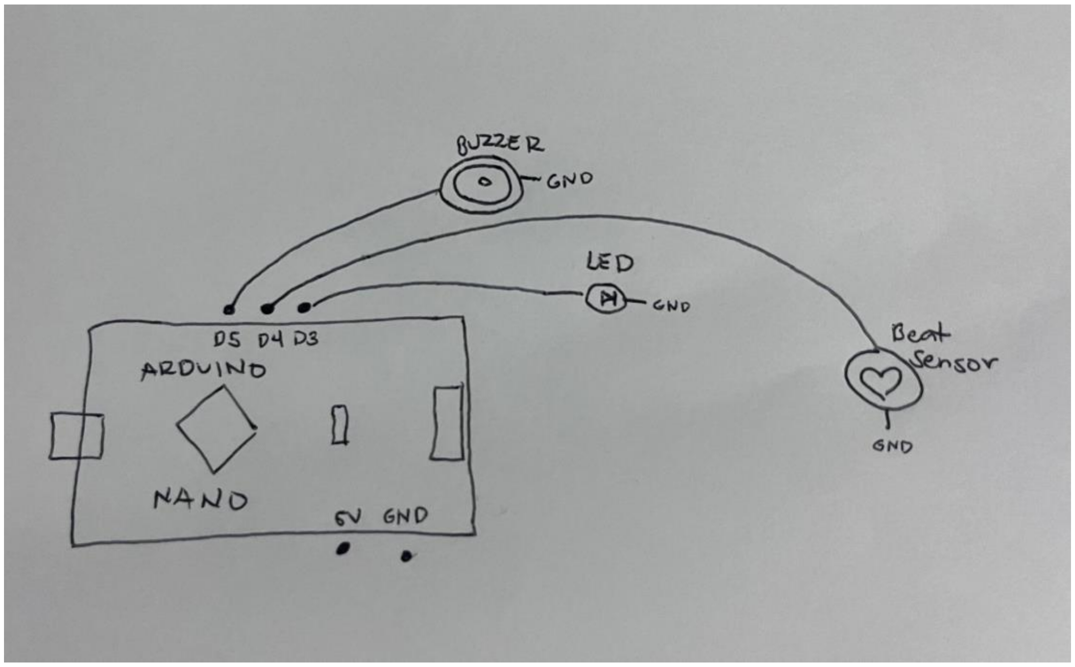
```
.S
#define __SFR_OFFSET 0x00
#include "avr/io.h"


.equ HARTBITE_PIN, 4
.equ BUZZ_PIN, 5
.equ LED_PIN, 3


.global setup_
.global buzzmeup


setup_:
  CBI DDRD, HARTBITE_PIN   ; Set HARTBITE_PIN as input
  SBI DDRD, BUZZ_PIN       ; Set BUZZ_PIN as output
  SBI DDRD, LED_PIN        ; Set LED_PIN as output
  SBI PORTD, HARTBITE_PIN  ; Enable pull-up resistor on HARTBITE_PIN
  RET
```

```
buzzmeup:
  SBIS PIND, HARTBITE_PIN  ; Skip next instruction if HARTBITE_PIN is high (touched)
  RJMP buzzmenot
  SBI PORTD, BUZZ_PIN      ; Set BUZZ_PIN high (1) if touched
  CBI PORTD, LED_PIN       ; Set LED_PIN low (0) if touched
  RET

buzzmenot:
  CBI PORTD, BUZZ_PIN      ; Set BUZZ_PIN low (0) if not touched
  SBI PORTD, LED_PIN       ; Set LED_PIN high (1) if not touched
  RET
```

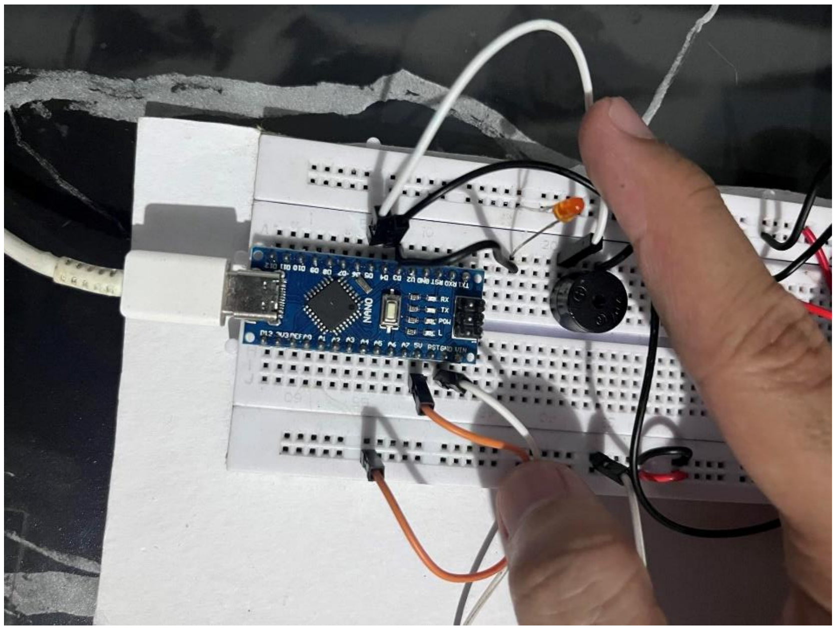


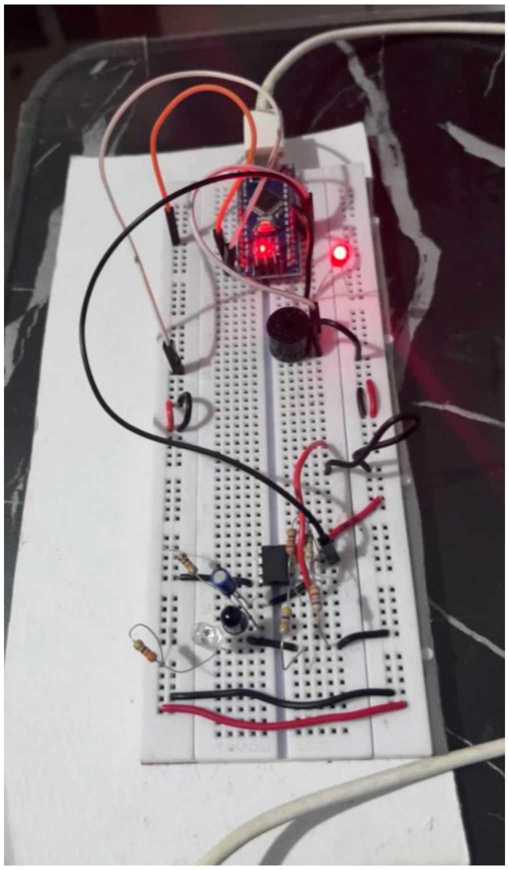
```
.equ HARTBITE_PIN, 4
.equ BUZZ_PIN, 5
.equ LED_PIN, 3

.global setup_
.global buzzmeup

setup_:
  CBI DDRD, HARTBITE_PIN   ; Set HARTBITE_PIN as input
  SBI DDRD, BUZZ_PIN       ; Set BUZZ_PIN as output
  SBI DDRD, LED_PIN        ; Set LED_PIN as output
  SBI PORTD, HARTBITE_PIN  ; Enable pull-up resistor on HARTBITE_PIN
  RET

buzzmeup:
  SBIS PIND, HARTBITE_PIN  ; Skip next instruction if HARTBITE_PIN is high (touched)
  RJMP buzzmenot
  SBI PORTD, BUZZ_PIN      ; Set BUZZ_PIN high (1) if touched
  CBI PORTD, LED_PIN       ; Set LED_PIN low (0) if touched
  RET

buzzmenot:
  CBI PORTD, BUZZ_PIN      ; Set BUZZ_PIN low (0) if not touched
  SBI PORTD, LED_PIN       ; Set LED_PIN high (1) if not touched
  RET
```

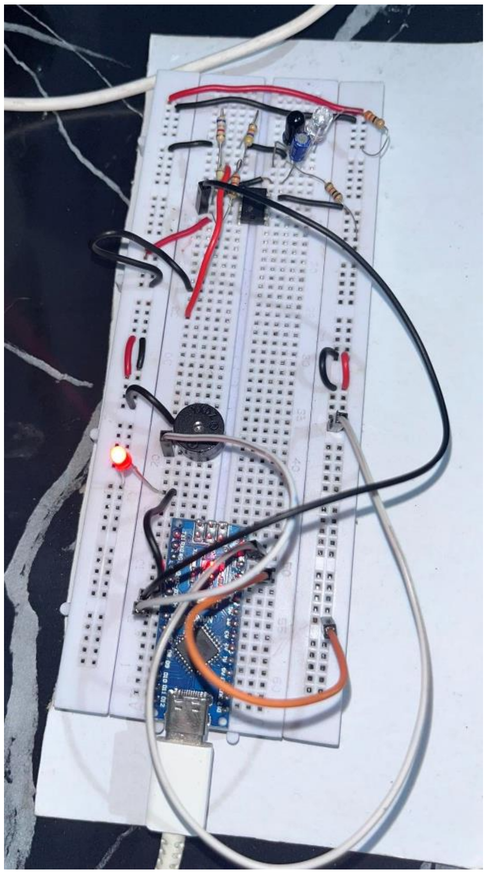
PIN CONNECTION

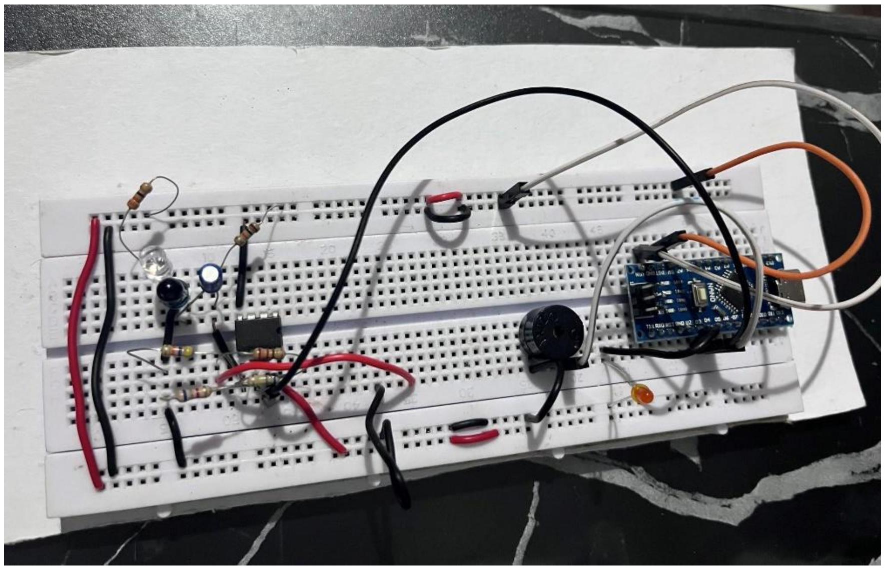


Simulation









Output = 1 (HIGH)

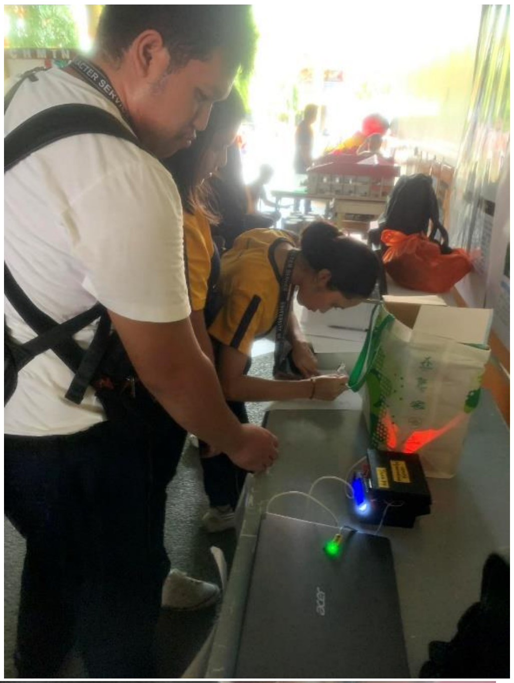
How it works?

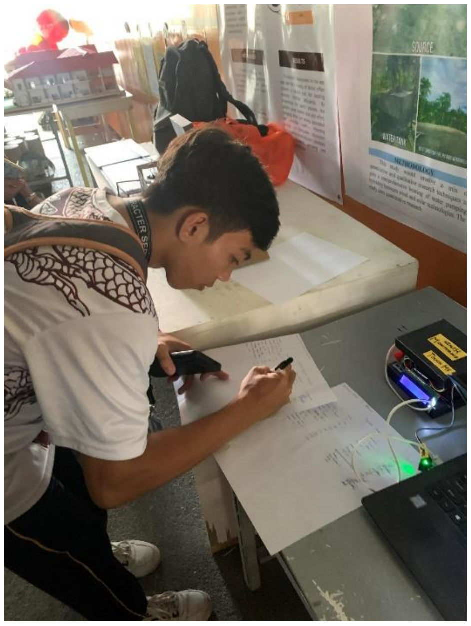
This project utilizes an ATMEGA328 microcontroller, programmed in assembly language, to effectively monitor and control heart rate variations. The core component of the system is a photoplethysmogram (PPG) sensor, which detects changes in blood volume that correlate with heartbeats. As the heart rate changes, the PPG sensor's output varies, leading to corresponding changes in voltage that are measured by the microcontroller's analog-to-digital converter (ADC).

The assembly code sets up the necessary I/O ports and configures the ADC for accurate voltage readings. The heartbeat sensor pin is defined as an input, while the LED and buzzer control pins are set as outputs. The program operates in a continuous loop, repeatedly checking the ADC value to determine the current heart rate. If a heartbeat is detected (high signal), the microcontroller activates the buzzer by setting the output pin high using the SBI (Set Bit in I/O Register) instruction and turns off the LED by clearing the output pin using the CBI (Clear Bit in I/O Register) instruction. Conversely, if no heartbeat is detected (low signal), the buzzer is turned off, and the LED is turned on.

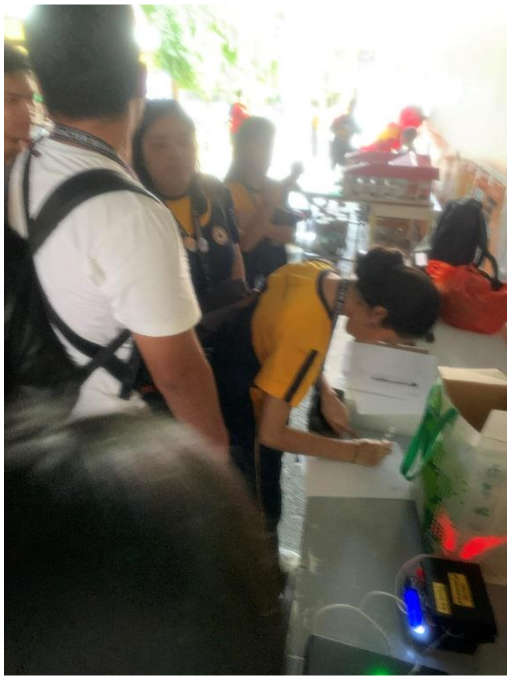
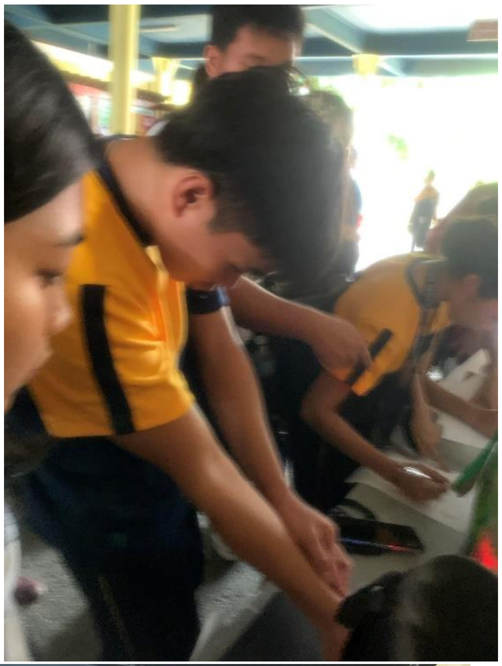
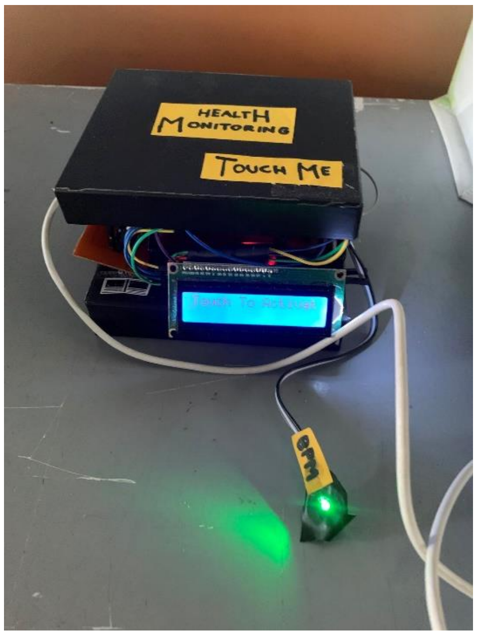
The conditional branching mechanism in the assembly code ensures that the buzzer and LED are only activated when necessary, providing efficient and responsive heart rate monitoring. This project highlights the advantages of using assembly language for precise hardware control, offering a solution for heart rate regulation in various applications. The system's design demonstrates the integration of analog sensor data with digital control logic, showcasing the potential of microcontroller-based systems in real-time health monitoring and management.

Photos on the simulation during Technocraft





line Medical
sultation




HEALTH MONITORING
TOUCH ME
BPM

BREAKTHROUGH TO THE
NO SMOKING

Project Summary

Working on the heartbeat sensor project has been quite challenging for me. I faced significant difficulties in understanding the concepts involved. Converting the Arduino code to Assembly language proved to be especially hard. However, through hard work and perseverance, I was able to successfully complete the task and defend the project.

This project demonstrates the potential of microcontroller-based systems for real-time health monitoring and management. It has been an exhilarating journey of growth and overcoming challenges, highlighting the importance of precise hardware control in achieving efficient and responsive heart rate monitoring.

Curriculum Vitae

Curriculum Vitae

Name: John Claidy Ken O. Taguran

Nickname: Ken-ken

Age: 20 years old

Birthday: May 15, 2003

Address: Poblacion Baclayon Bohol

Contact Number: 09497629625

Email: jckotaguran@universityofbohol.edu.ph

Mother's name: Marilyn O. Taguran

Father's name: John M. Taguran

Motto*: "EMBRACE THE NEW ITERATION OF YOURSELF; YOU DON'T ALWAYS HAVE TO BE A WARRIOR."*

Portfolio: https://claidytaguran.github.io/porttaguran.github.io/