EMBEDDED SYSTEM PROJECT

SUBJECT:-
A system to monitor heart beats of a human subject using 8051 microcontroller.

By

Sarthak Naik (122EE0155)

Gourav Rout (122EE0161)

Utkal Keshari Sahoo(122EE0163)

Under the supervision of
Prof. Supratim Gupta

NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

REAL-TIME HEARTBEAT MONITORING SYSTEM USING 8051 MICRO CONTROLLER

OBJECTIVE OF THE PROJECT :

The objective of the Heartbeat Sensor Project with the 8051 microcontroller is to build a compact and efficient system to monitor and display heart rate. The 8051 microcontroller reads the analog signal from a heart rate sensor, converts it into a digital value, and shows the heart rate on a display. This project demonstrates key concepts like sensor interfacing, analog-to-digital conversion, and display operations within an embedded system. It provides hands-on experience with microcontroller applications, particularly in the field of health monitoring, and serves as a foundation for developing future biomedical systems using microcontrollers.

APPARATUS :

1.8051(AT89S52) Micro Controller

2. ATMEL 8051 Starter Kit (Quick Development Board + ISP Programmer

3. Heartbeat Sensor

4. 7-Segment Display (Common Anode) - 3

5. Transistors(2N2222) - 3

6. Resistors - 0.1Kohm -10

7. Breadboard

8. Jumper Wires

CIRCUIT DESCRIPTION :

➢ The user places their finger on the heartbeat sensor, which detects pulse fluctuations and converts them into a digital signal. The test input is set to 1, with ground connected to the sensor.

This ensures the sensor detects heartbeats and converts the analog signal for processing by the microcontroller.

➢ The heartbeat sensor's output is sent to the 8051 microcontroller via the P3.5 port. The microcontroller processes the data, enabling pulse monitoring and heart rate calculation. The digital input allows easy processing and display of results.

➢ The microcontroller sends signals from P0.0 to P0.2 to the base terminals of three transistors, controlling the 7-segment displays. These signals turn the transistors on or off based on pulse data to display the correct heart rate.

➢ The collector terminals of the transistors are connected to a 5V supply, while the emitter terminals connect to the 7-segment displays. This allows transistors to control current to the displays, ensuring proper heart rate display.

➢ Three 7-segment displays are connected in series to the microcontroller via P0.0 to P0.7 pins. Each display shows one digit of the heart rate, with the microcontroller managing each segment.

➢ The common anode (COM) pin of each 7-segment display connects to transistors that regulate blinking. The microcontroller controls the transistor bases to create the blinking effect for readable heart rate display.

➢ Resistors of 100 ohms are used between the 7-segment displays and microcontroller to limit current. These resistors protect the displays from excess current and ensure proper functioning.

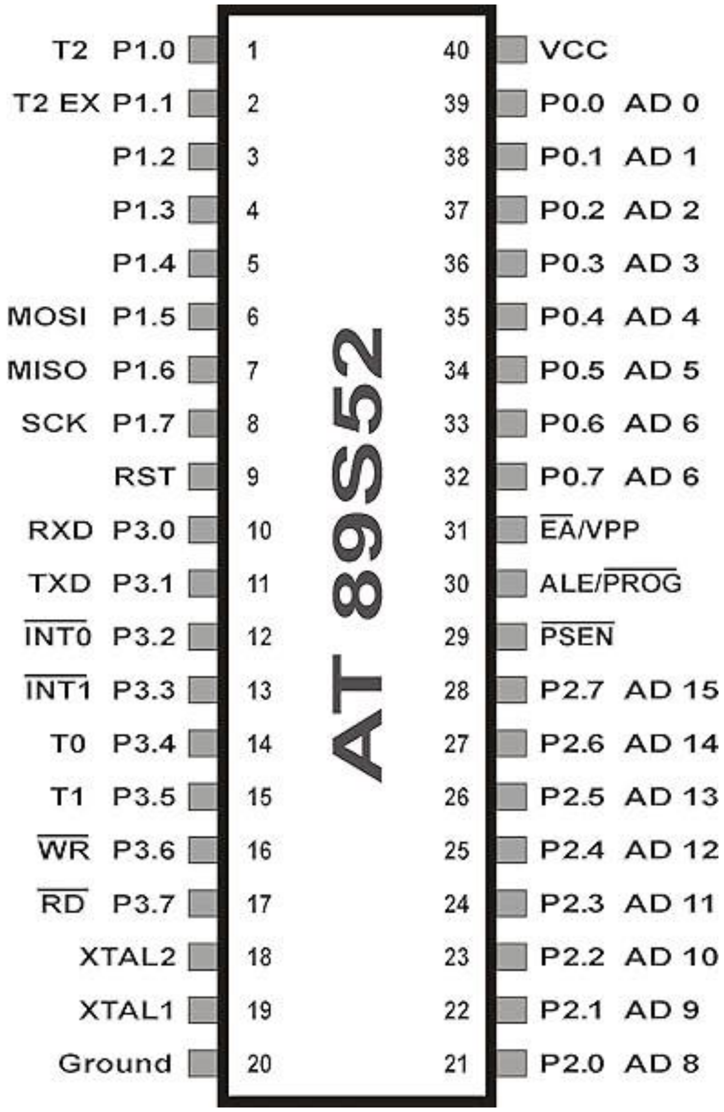
➢ The heartbeat sensor's signal is processed by the microcontroller, which sends it to transistors to control the 7-segment displays. The displays show heart rate in beats per minute, updating every 15 seconds.

➢ When powered on, LED D4 lights up. After placing the fingertip on the sensor, preset R14 adjusts LED D4 blinking, indicating sensor activation. After blinking, the power resets, and the display shows heart rate.

COMPONENTS DISCRIPTION:

1.8051 Micro Controller:

- **8-bit microcontroller**: Processes 8 bits of data at a time.
- **4KB ROM**: Built-in memory for program storage.
- **128 Bytes RAM**: For data storage during operation.
- **4 I/O ports**: Offers flexibility in interfacing with external devices.
- **Timers/Counters**: For time-based operations.
- **Serial Communication**: Supports UART for communication.
- **Low power consumption**: Suitable for embedded applications.



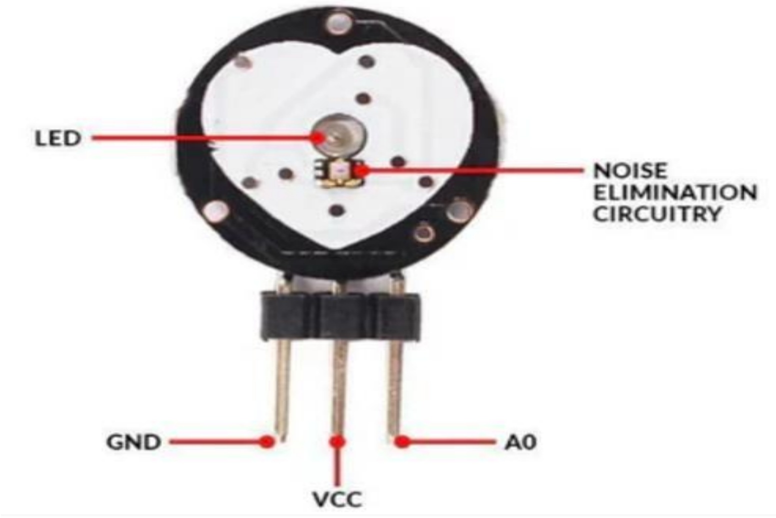
2. ATMEL 8051 Starter Kit:

- **Microcontroller**: Based on ATMEL 8051, an 8-bit microcontroller.
- **Development Platform**: Ideal for embedded system design and learning.
- **Memory**: Contains on-board flash memory for program storage.
- **I/O Ports**: Provides multiple I/O pins for interfacing with peripherals.
- **Debugging Tools**: Supports in-circuit debugging and programming.
- **Power Supply**: Operates with standard power inputs.
- **Versatile**: Suitable for various applications, including sensors and displays.

3. Heartbeat Sensor:

- **Function**: Detects heart rate by sensing pulse fluctuations.
- **Technology**: Utilizes optical or electrical sensors for pulse detection.
- **Output**: Converts analog signals to digital for microcontroller processing.
- **Interface**: Communicates with microcontrollers through analog or digital pins.

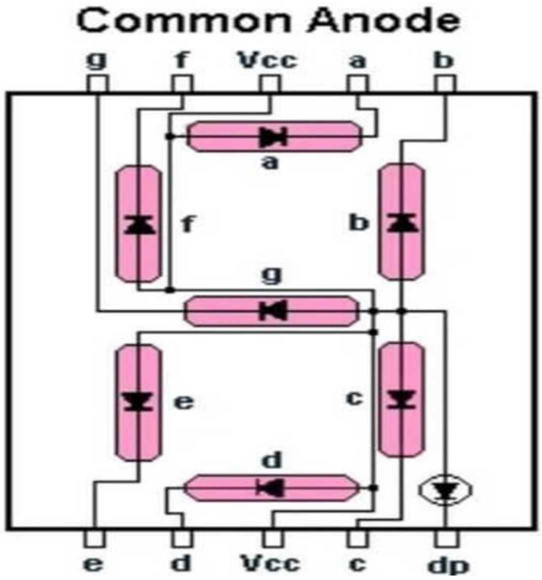
- **Applications**: Used in health monitoring and wearable devices.
- **Accuracy**: Provides real-time heart rate data for analysis.



Pinout of Heart Rate Pulse Sensor Module

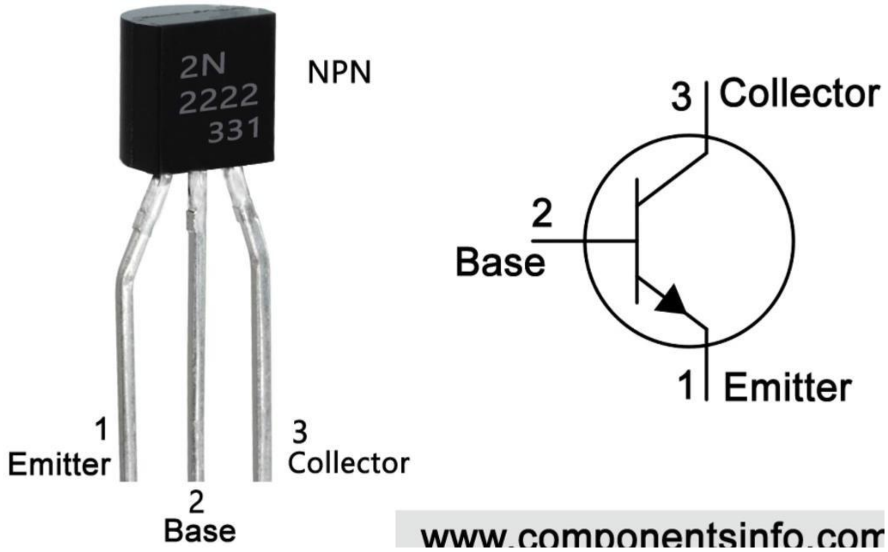
4. 7-Segment Display (Common Anode):

- **Common Anode**: All the anodes of the LED segments are connected to a common pin.
- **Operation**: Each segment lights up when its corresponding cathode is connected to ground.
- **Digits**: Displays numerical digits (0-9) by lighting up specific segments.
- **Power Supply**: Operates with a positive voltage supply.
- **Control**: Driven by microcontroller or external logic circuits.



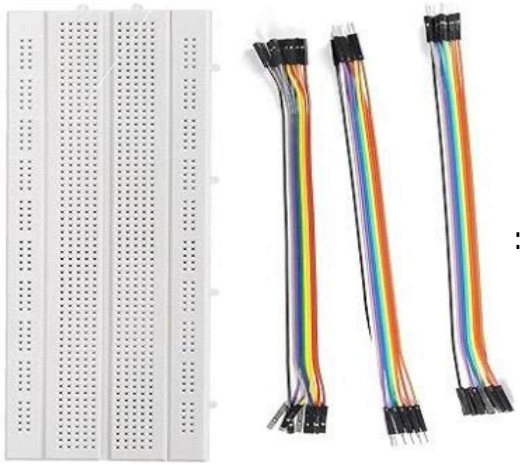
4. Transistors(2N2222):

- **Type**: NPN bipolar junction transistor (BJT).
- **Function**: Acts as a switch or amplifier in circuits.
- **Current Handling**: Can handle up to 800mA of collector current.
- **Voltage**: Maximum collector-emitter voltage is 40V.
- **Application**: Used in signal amplification and switching applications.
- **Package**: Available in TO-92 packaging.



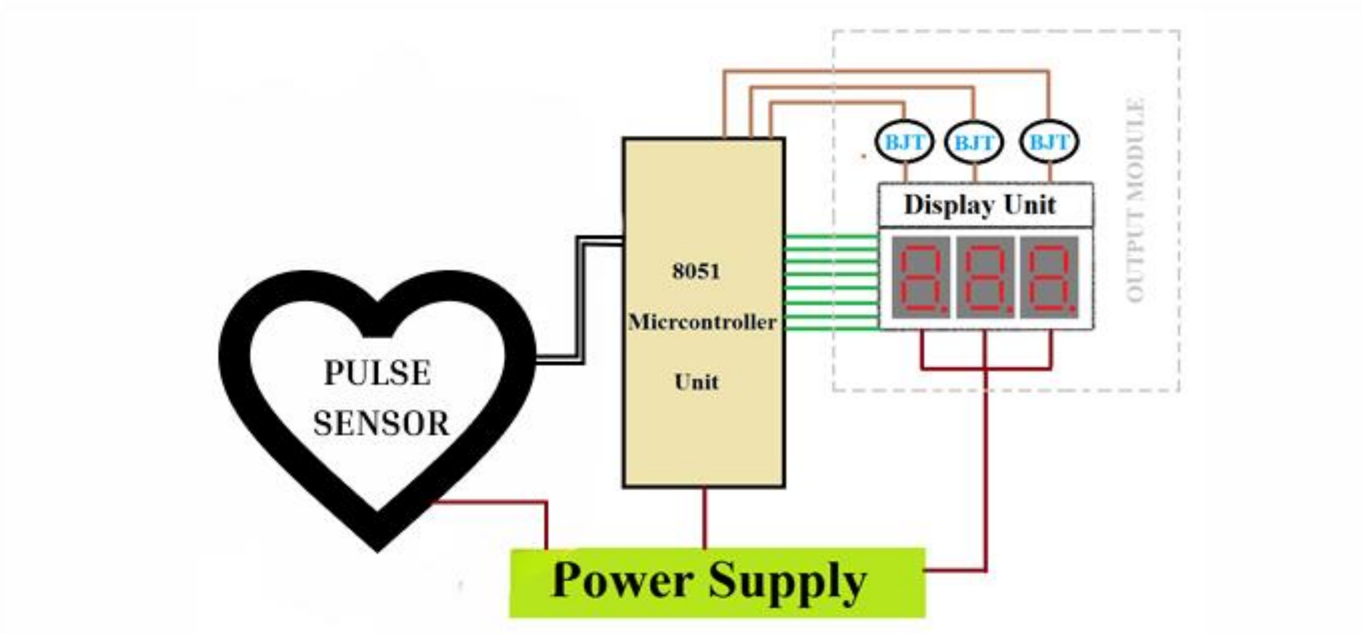
5.Breadboard & Jumper Wires:

- **Breadboard**: A reusable platform for prototyping electronic circuits without soldering.
- **Connections**: Contains rows and columns of interconnected holes for components and wires.
- **Jumper Wires**: Flexible wires used to make connections between components on the breadboard.
- **Application**: Ideal for testing and designing circuits quickly.
- **No Soldering**: Makes it easy to modify circuits.



:

Block Diagram Representation:



- **Heartbeat Sensor**: Detects pulse rate and generates an analog signal.
- **Analog-to-Digital Conversion**: The signal is sent to the 8051 microcontroller for conversion.
- **8051 Microcontroller**: Processes the digital signal and calculates heart rate.
- **7-Segment Display**: Displays the heart rate (beats per minute)
- **Power Supply**: Provides necessary voltage to components.

Algorithm used:

1. Initialization: The Look-Up Table (LUT) is initialized first, followed by loading the starting address of the LUT into the DPTR register.
2. Port Configuration: Ports P0 and P1 are set as output ports, while P3.5 is configured as an input port to receive data from the heartbeat sensor.
3. Timer Setup: Timer0 and Timer1 are used for counting and generating delays. TL1 and TH1 are loaded with initial values for Timer1.
4. Timer Functionality: Timer1, configured as an 8-bit auto-reload counter, counts incoming heartbeat pulses, while Timer0, set as a 16-bit timer, generates a delay of 1 second for Timer1's counting.
5. Pulse Counting: The program loops 230 times to create a 15-second interval. The number of pulse counts during this period is multiplied by 4 to calculate the heart rate in beats per minute.
6. Timer0 Configuration: Timer0, in Mode 1, counts up to 65536 and generates a 65536 µS delay, driven by a 12MHz crystal.
7. Digit Separation: The heart rate value is separated into hundreds, tens, and units places by dividing the number by 100, 10, and 1, respectively.
8. Display Mechanism: Transistors (Q1, Q2, Q3) connected to ports P1.0 to P1.2 control the display of the digits on three 7-segment displays connected to ports P0.0 to P0.7.
9. Subroutine Loop: The DLOOP subroutine is repeated 100 times to drive the display, fetching the 7-segment pattern from the LUT using the MOVC instruction.
10. LUT Usage: The LUT is used to get the corresponding 7-segment display pattern for each digit, stored at the address defined by DPTR.

Assembly Code:

```
ORG 0000H                          ; origin address of the program
MOV DPTR, #LUT   ; moves the starting address of the LUT (look-up table) to DPTR
MOV P1, #00H             ; sets port P1 as an output port (all bits as output)
MOV P0, #00H        ; sets port P0 as an output port (all bits as output)
MAIN:
   MOV R6, #230D            ; loads register R6 with value 230D (decimal 560)
   SETB P3.5                        ; sets P3.5 as input port (typically used for external interrupt or input)
   MOV TMOD, #61H           ; sets Timer1 as Mode2 (8-bit auto-reload) counter and Timer0 as Mode1 (16-bit timer)
   MOV TL1, #00H           ; loads the low byte of Timer1 with initial value 0
   MOV TH1, #00H           ; loads the high byte of Timer1 with initial value 0
   SETB TR1                       ; starts Timer1 (counter mode)
BACK:
   MOV TH0, #00H           ; loads the high byte of Timer0 with initial value 0
   MOV TL0, #00H           ; loads the low byte of Timer0 with initial value 0
   SETB TR0                        ; starts Timer0 (timer mode)
HERE:
   JNB TF0, HERE                  ; jumps to HERE label if Timer0 overflow flag (TF0) is not set (wait for timer overflow)
   CLR TR0                        ; stops Timer0
   CLR TF0                        ; clears Timer0 overflow flag
   DJNZ R6, BACK           ; decrements R6 and jumps to BACK if R6 is not zero (repeats the process)
   CLR TR1                        ; stops Timer1 (counter mode)
   CLR TF0                        ; clears Timer0 overflow flag
   CLR TF1                        ; clears Timer1 overflow flag
   ACALL DLOOP            ; calls the DLOOP subroutine to display the count
   SJMP MAIN            ; jumps back to the MAIN label to restart the process
```

```
DLOOP:
    MOV R5, #252D         ; loads register R5 with value 252D (decimal 594)
BACK1:
    MOV A, TL1                        ; loads the current count value from TL1
(Timer1 low byte) into the accumulator
    MOV B, #4D                        ; loads register B with value 4D (decimal 77)
    MUL AB           ; multiplies A and B (A = A * B, B is used as the multiplier)
    MOV B, #100D                ; loads register B with value 100D (decimal 256)
    DIV AB                            ; divides AX by B, A = quotient, B = remainder
(isolate first digit of the count)
    SETB P1.0            ;sets bit 0 of port P1 (turn on Q1 transistor for first digit
display)
    ACALL DISPLAY                    ; calls the DISPLAY subroutine to convert
first digit to 7-segment pattern
    MOV P0, A                        ; sends the pattern for the first digit to port P0
(7-segment display data)
    ACALL DELAY          ; calls the DELAY subroutine to create a small delay
    ACALL DELAY             ; additional delay to ensure the display is visible
    MOV A, B                          ; moves the second digit of the count (from B
register) to the accumulator
    MOV B, #10D                        ; loads register B with value 10D (decimal 16)
    DIV AB           ; divides AX by 10, A = quotient (second digit), B =
remainder (unused)
    CLR P1.0      ; clears bit 0 of port P1 (turn off Q1 transistor for first digit)
    SETB P1.1                         ; sets bit 1 of port P1 (turn on Q2 transistor for
second digit display)
    ACALL DISPLAY                    ; calls the DISPLAY subroutine to convert
second digit to 7-segment pattern
    MOV P0, A                     ; sends the pattern for the second digit to port P0
    ACALL DELAY                        ; calls the DELAY subroutine
    ACALL DELAY              ; additional delay to ensure the display is visible
    MOV A, B                          ; moves the third digit of the count (from B
register) to the accumulator
```

CLR P1.1 ; clears bit 1 of port P1 (turn off Q2 transistor for second digit)

SETB P1.2 ;sets bit 2 of port P1 (turn on Q3 transistor for third digit display)

ACALL DISPLAY ; calls the DISPLAY subroutine to convert third digit to 7-segment pattern

MOV P0, A ; sends the pattern for the third digit to port P0

ACALL DELAY ; calls the DELAY subroutine

ACALL DELAY ; additional delay to ensure the display is visible

CLR P1.2 ; clears bit 2 of port P1 (turn off Q3 transistor for third digit)

DJNZ R5, BACK1 ; decrements R5 and jumps back to BACK1 if R5 is not zero (loops 252 times)

MOV P0, #0FFH ; sets all bits of port P0 to 1 (turn off all 7-segment displays)

RET ; returns from DLOOP subroutine

DELAY:

MOV R7, #250D ; loads register R7 with value 250D (decimal 592)

DEL1:

DJNZ R7, DEL1 ; decrements R7 and repeats the loop until R7 reaches 0 (creates delay)

RET ; returns from DELAY subroutine

DISPLAY:

MOVC A, @A+DPTR ; fetches the 7-segment pattern from the look-up table (LUT) at address DPTR + A

CPL A ; complements the accumulator (inverts bits for common cathode display)

RET ; returns from DISPLAY subroutine

LUT:

DB 3FH ; 7-segment pattern for digit 0

DB 06H ; 7-segment pattern for digit 1

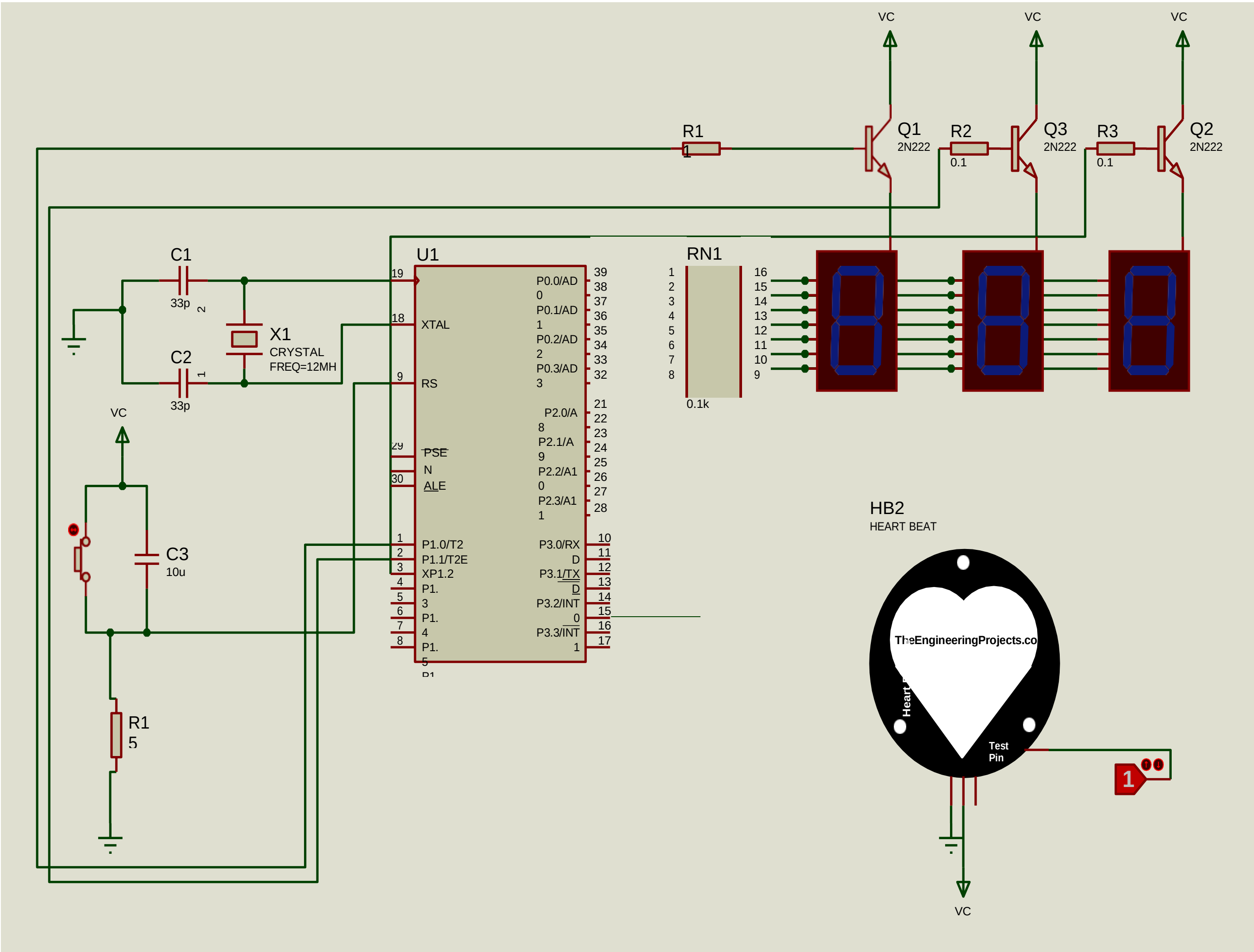
DB 5BH ; 7-segment pattern for digit 2

DB 4FH ; 7-segment pattern for digit 3

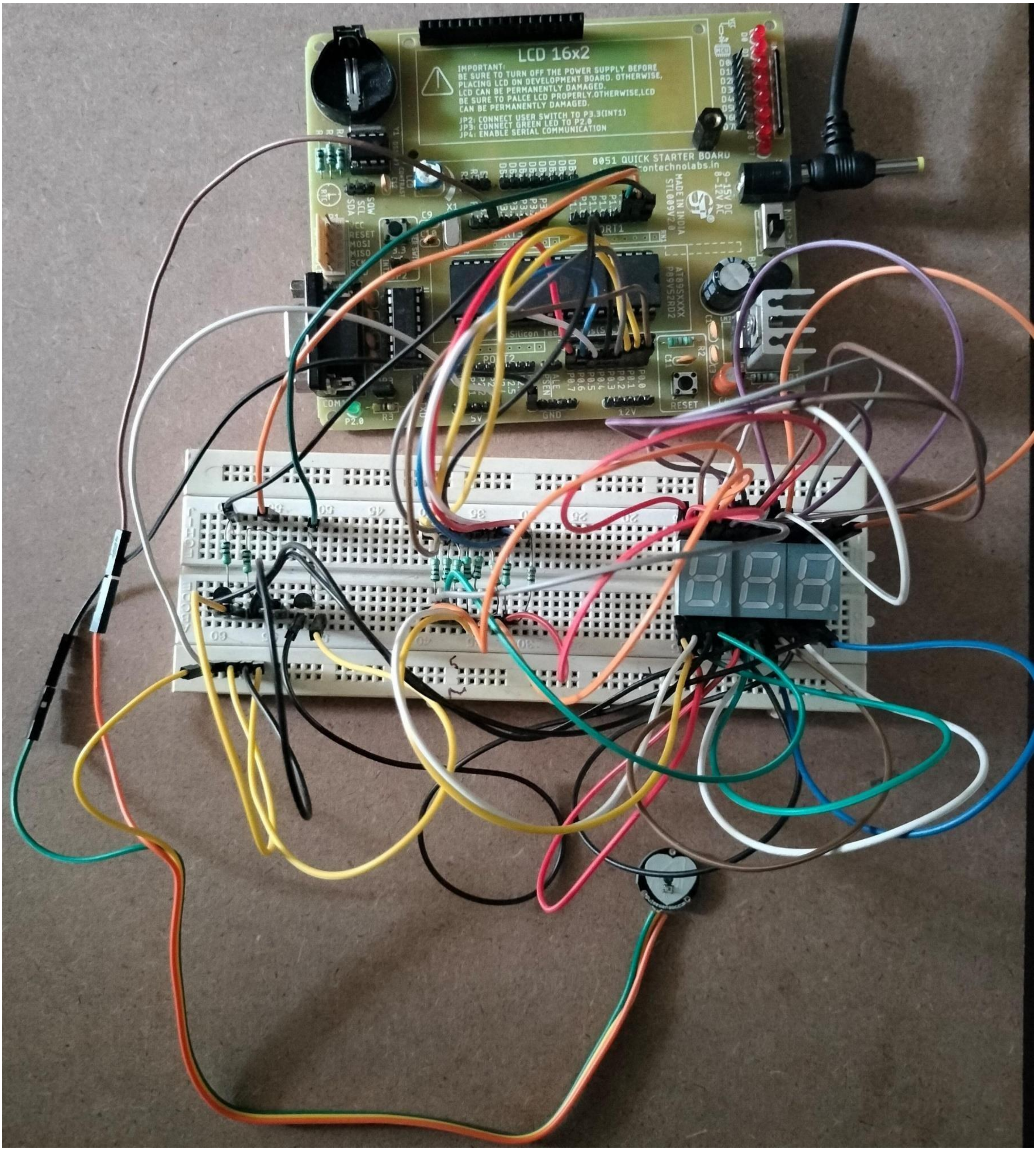
DB 66H ; 7-segment pattern for digit 4

```
    DB 6DH                    ; 7-segment pattern for digit 5
    DB 7DH                    ; 7-segment pattern for digit 6
    DB 07H                    ; 7-segment pattern for digit 7
    DB 7FH                    ; 7-segment pattern for digit 8
    DB 6FH                    ; 7-segment pattern for digit 9
END                           ; marks the end of the program
```

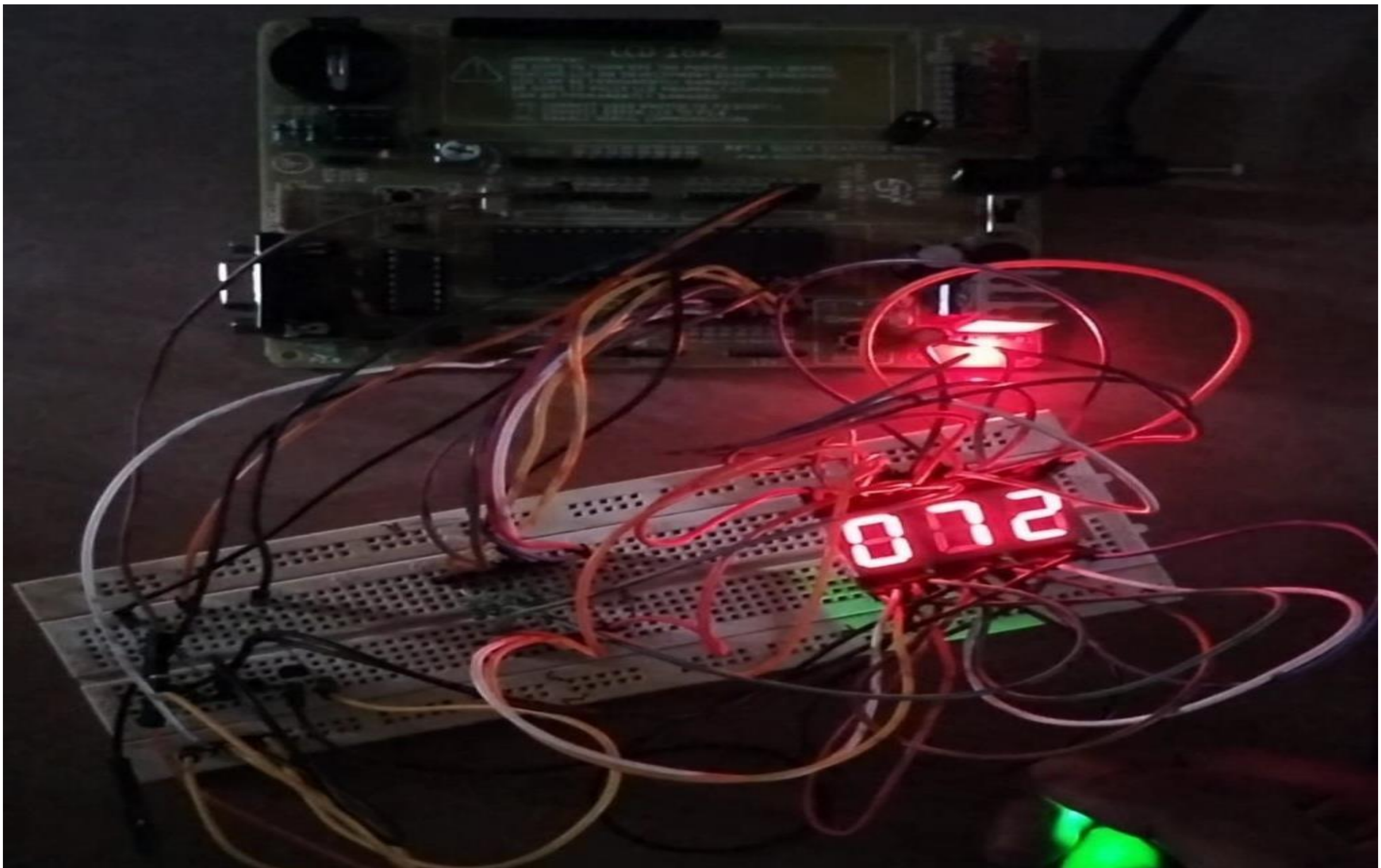
Simulation on Proteus:

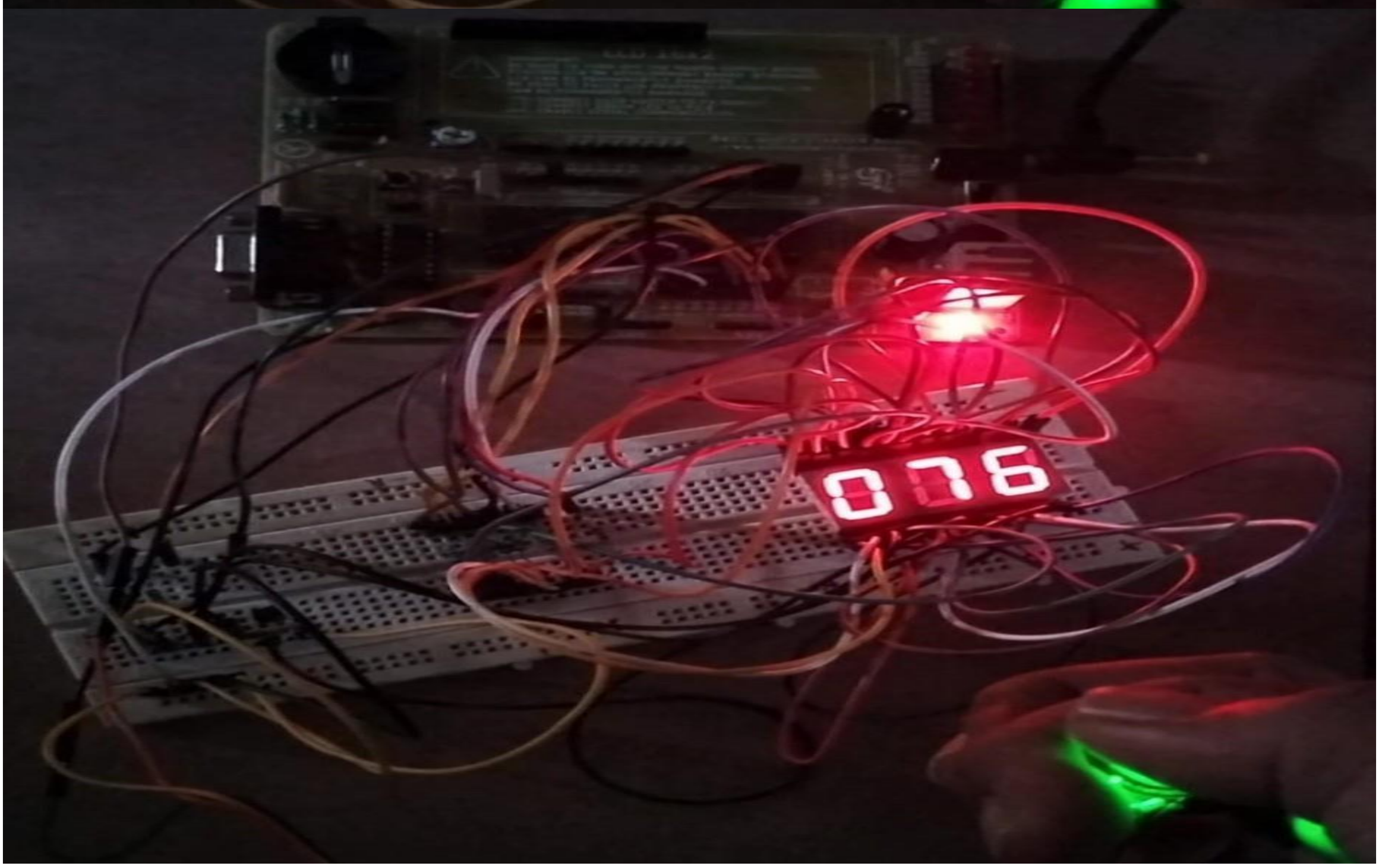


Circuit Representation:



Observation:





Practical Applications:

1. Health Monitoring: Used in hospitals and clinics to continuously monitor patients' heart rates for early detection of abnormalities.
2. Wearable Devices: Integrated into wearable health devices for personal fitness tracking and monitoring.
3. Emergency Response: Helps medical personnel track vital signs in emergency situations.
4. Sports Medicine: Monitors athletes' heart rates to optimize training and prevent overexertion.
5. Home Healthcare: Allows individuals to track heart rate at home for chronic disease management.

Advantages:

1. Cost-effective**:** Affordable components make it a budget-friendly solution for heart rate monitoring.
2. Real-time Monitoring: Provides continuous tracking of heart rate, ensuring timely intervention.
3. Compact Design**:** The system is lightweight and portable for ease of use in various settings.
4. Low Power Consumption**:** Efficient power usage, ideal for prolonged use.
5. Scalability**:** Can be expanded with additional sensors and features for more complex applications.

Conclusion:

The Real-Time Heartbeat Monitoring System using the 8051 microcontroller offers an efficient and cost-effective solution for continuous heart rate monitoring. By integrating sensors, a microcontroller, and display units, this system provides real-time, accurate heart rate data with minimal power consumption. It is highly portable, making it ideal for medical, sports, and home healthcare applications. The system's ability to process signals and display results quickly ensures timely detection of abnormal heart rates, improving patient care. Furthermore, its scalability and simplicity make it a versatile tool for various health monitoring needs.