

**Heart Beat Monitoring
System Using
Microcontroller
ATMEGA328**



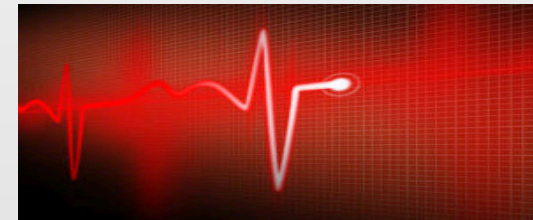
OBJECTIVE

- Heartbeat Sensor are designed to give digital output of a heartbeat rate when finger is placed on it
- The main objective of this project is to design a heartbeat rate monitoring system using Arduino UNO and heart beat sensor
- Arduino is a open source prototyping platform and is based on ATMEGA328 microcontroller
- Using LCD to display the heartbeat rate, Heartbeat sensor to measure speed of the heart rate
- It is cost effective and reliable

INTRODUCTION

Heartbeat Sensor

- Heartbeat Sensor is an electronic device that is used to measure the heartbeat rate
- Heartbeat rate can be monitored in two ways
 - Manually check the pulse
 - Use Heartbeat Sensor
- Most precise one is electrocardiography but the most easier way to monitor the heartbeat rate is to use the heartbeat sensor



- Lower than normal heart rates are usually an indication of condition known as bradycardia, while higher is known as tachycardia

Adult	Babies	Children	Males	Females
72 (bpm)	120 (bpm)	90 (bpm)	70 (bpm)	75 (bpm)

- Heartbeat rate can vary according to the demand of muscles to absorb oxygen and excrete carbon dioxide changes

HEARTBEAT SENSOR

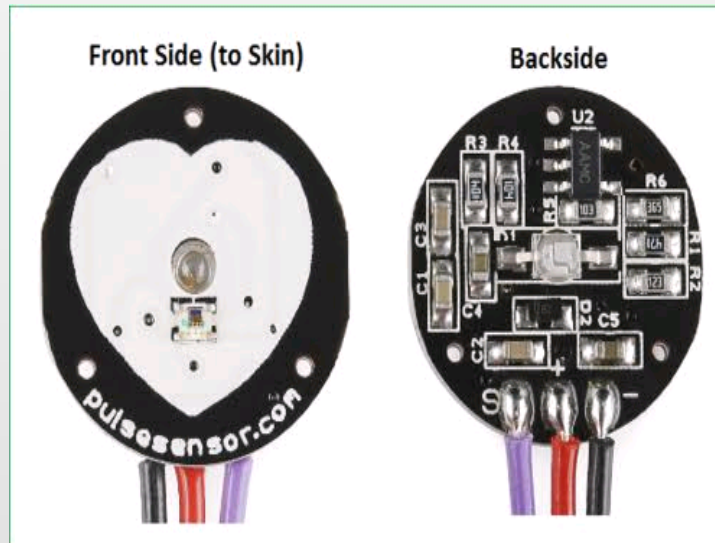


Fig 1: Heartbeat Sensor Front and Back view

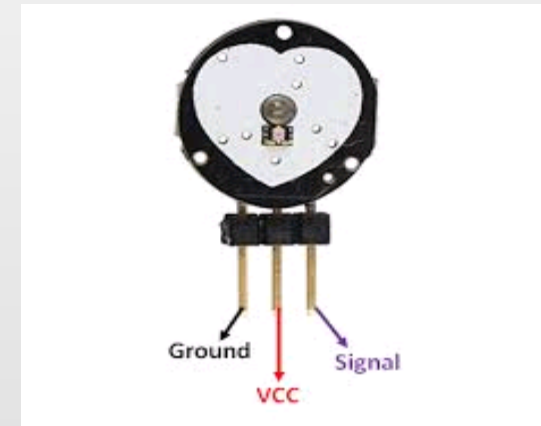


Fig 2: Heartbeat Sensor Pinout

ARDUINO UNO

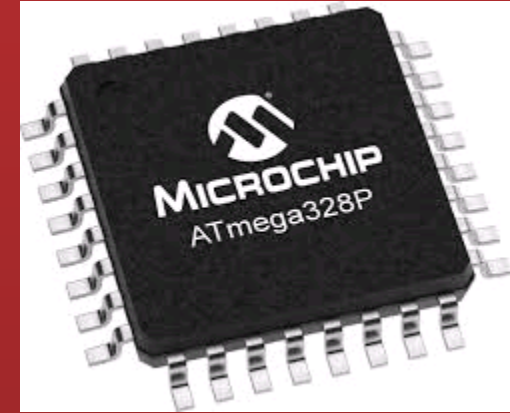
- The Arduino UNO is a board based on ATmega328 microcontroller
- It is an open source physical computing platform
- It has 14 digital input/output pins, 16MHz crystal oscillator, USB connection, 6 analog inputs, a power jack, ICSP header and reset button



Fig 3: Arduino UNO board

MICROCONTROLLER

- It is a small , low cost, and self – contained computer on a chip that can be used as an embedded system
- It consists of CPU, memory, system clock and peripherals
- It is used for control purposes and data analysis



**Fig 4: ATMega328
microcontroller**

HARDWARE REQUIREMENTS



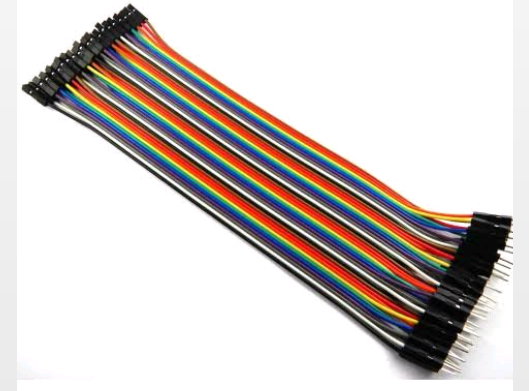
Arduino UNO



16 x 2 LCD



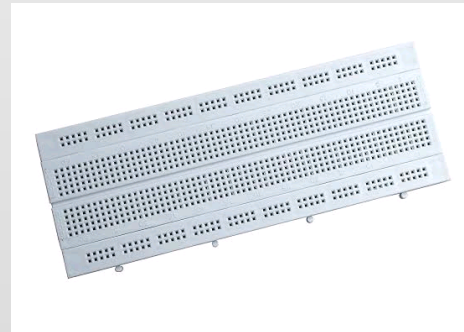
Pulse Sensor



Jumper wires



10 K Ω Potentiometer

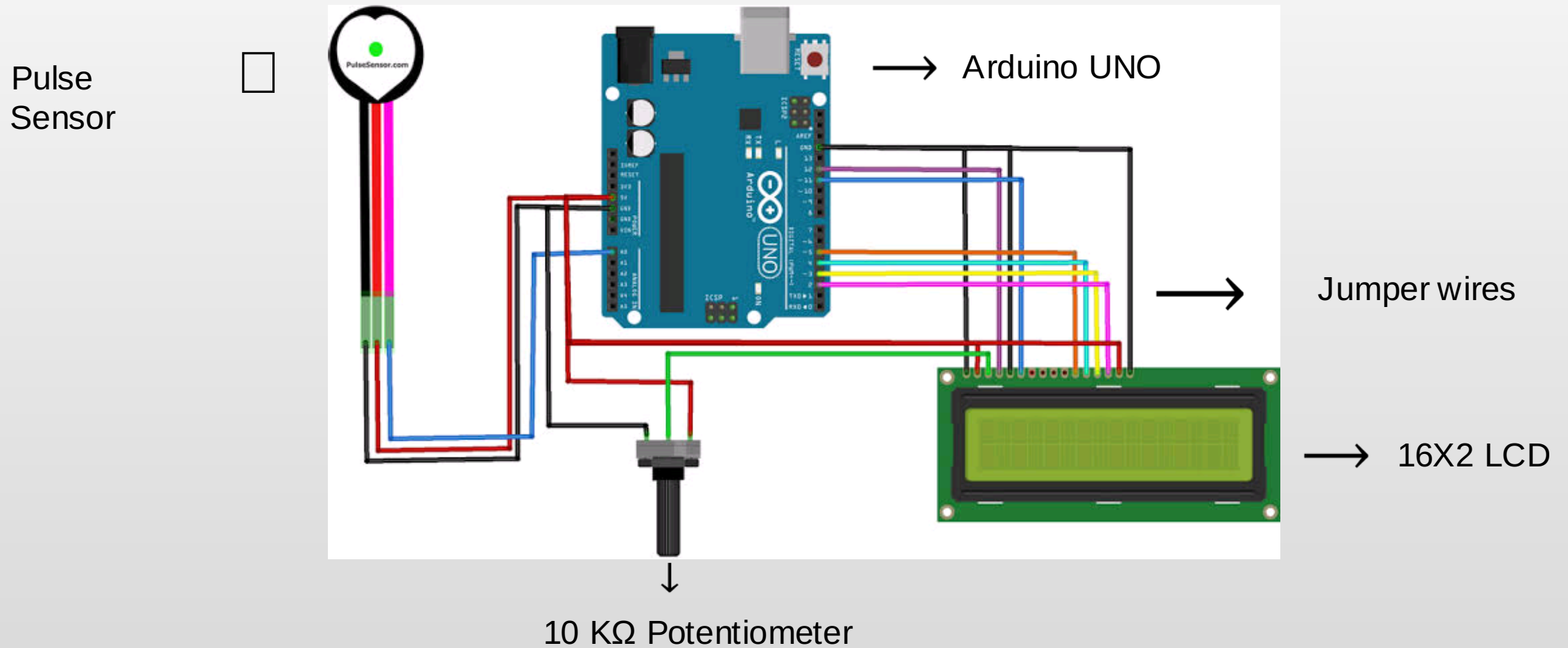


Bread board



USB Cable

CIRCUIT DIAGRAM

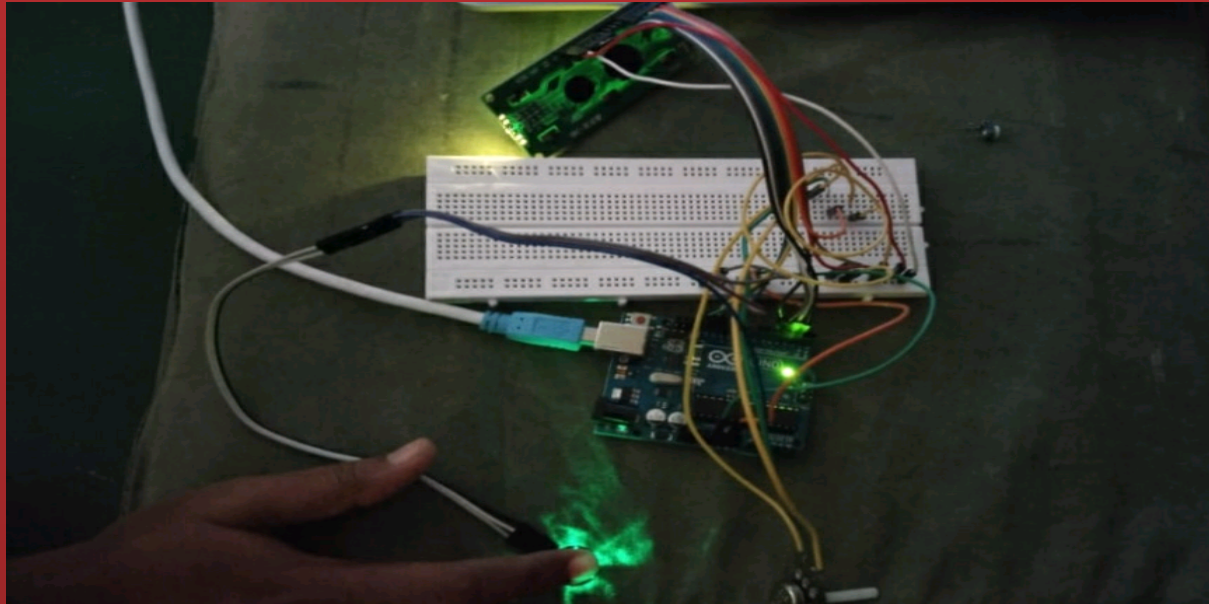


METHODOLOGY

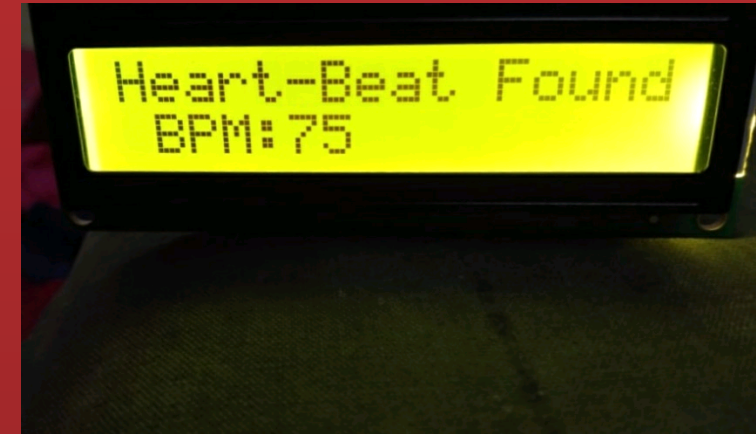
Steps involved in programming the Arduino

- The circuit was wired as shown in the circuit diagram
- Install the Arduino IDE software
- Open the IDE software
- Install drivers for Pulse Sensor, LCD (Liquid Crystal_12C)

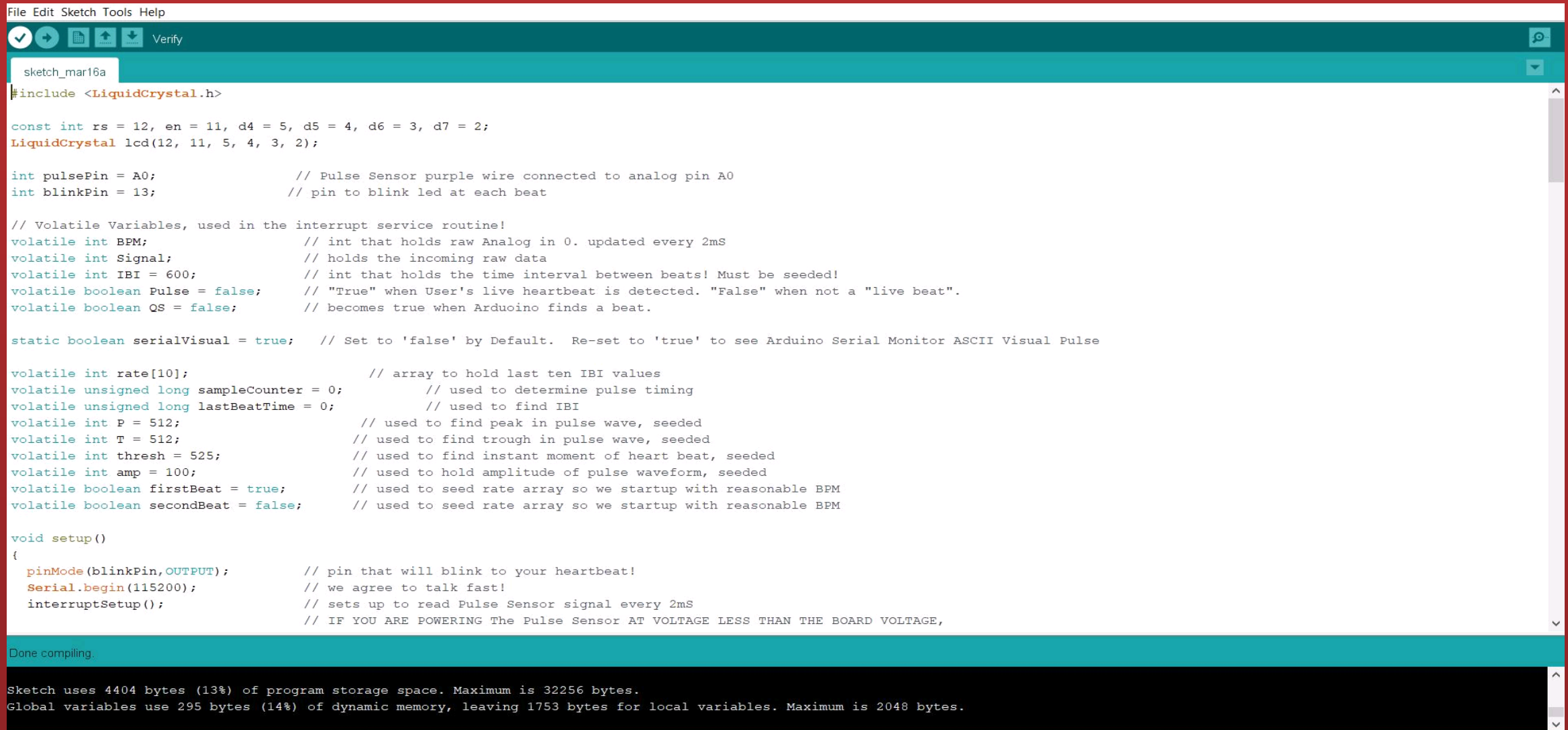
- Click the verify/compile button to check the program for errors
- Connect the Arduino board with laptop using USB
- Setup serial ports (COM6) being used
- Setup board (Arduino UNO) which we need to program
- Click upload code to send code to Arduino board



**Fig 5: Circuit connection of Arduino UNO
and Pulse Sensor**



**Fig 6: Heartbeat rate using
Pulse sensor**



```
File Edit Sketch Tools Help
Verify

sketch_mar16a
#include <LiquidCrystal.h>

const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int pulsePin = A0;           // Pulse Sensor purple wire connected to analog pin A0
int blinkPin = 13;           // pin to blink led at each beat

// Volatile Variables, used in the interrupt service routine!
volatile int BPM;             // int that holds raw Analog in 0. updated every 2mS
volatile int Signal;          // holds the incoming raw data
volatile int IBI = 600;       // int that holds the time interval between beats! Must be seeded!
volatile boolean Pulse = false; // "True" when User's live heartbeat is detected. "False" when not a "live beat".
volatile boolean QS = false;  // becomes true when Arduino finds a beat.

static boolean serialVisual = true; // Set to 'false' by Default. Re-set to 'true' to see Arduino Serial Monitor ASCII Visual Pulse

volatile int rate[10];        // array to hold last ten IBI values
volatile unsigned long sampleCounter = 0; // used to determine pulse timing
volatile unsigned long lastBeatTime = 0; // used to find IBI
volatile int P = 512;         // used to find peak in pulse wave, seeded
volatile int T = 512;         // used to find trough in pulse wave, seeded
volatile int thresh = 525;    // used to find instant moment of heart beat, seeded
volatile int amp = 100;       // used to hold amplitude of pulse waveform, seeded
volatile boolean firstBeat = true; // used to seed rate array so we startup with reasonable BPM
volatile boolean secondBeat = false; // used to seed rate array so we startup with reasonable BPM

void setup()
{
  pinMode(blinkPin, OUTPUT); // pin that will blink to your heartbeat!
  Serial.begin(115200);       // we agree to talk fast!
  interruptSetup();           // sets up to read Pulse Sensor signal every 2mS
  // IF YOU ARE POWERING The Pulse Sensor AT VOLTAGE LESS THAN THE BOARD VOLTAGE,

  Done compiling.

Sketch uses 4404 bytes (13%) of program storage space. Maximum is 32256 bytes.
Global variables use 295 bytes (14%) of dynamic memory, leaving 1753 bytes for local variables. Maximum is 2048 bytes.
```

Fig 7: Code into Arduino UNO

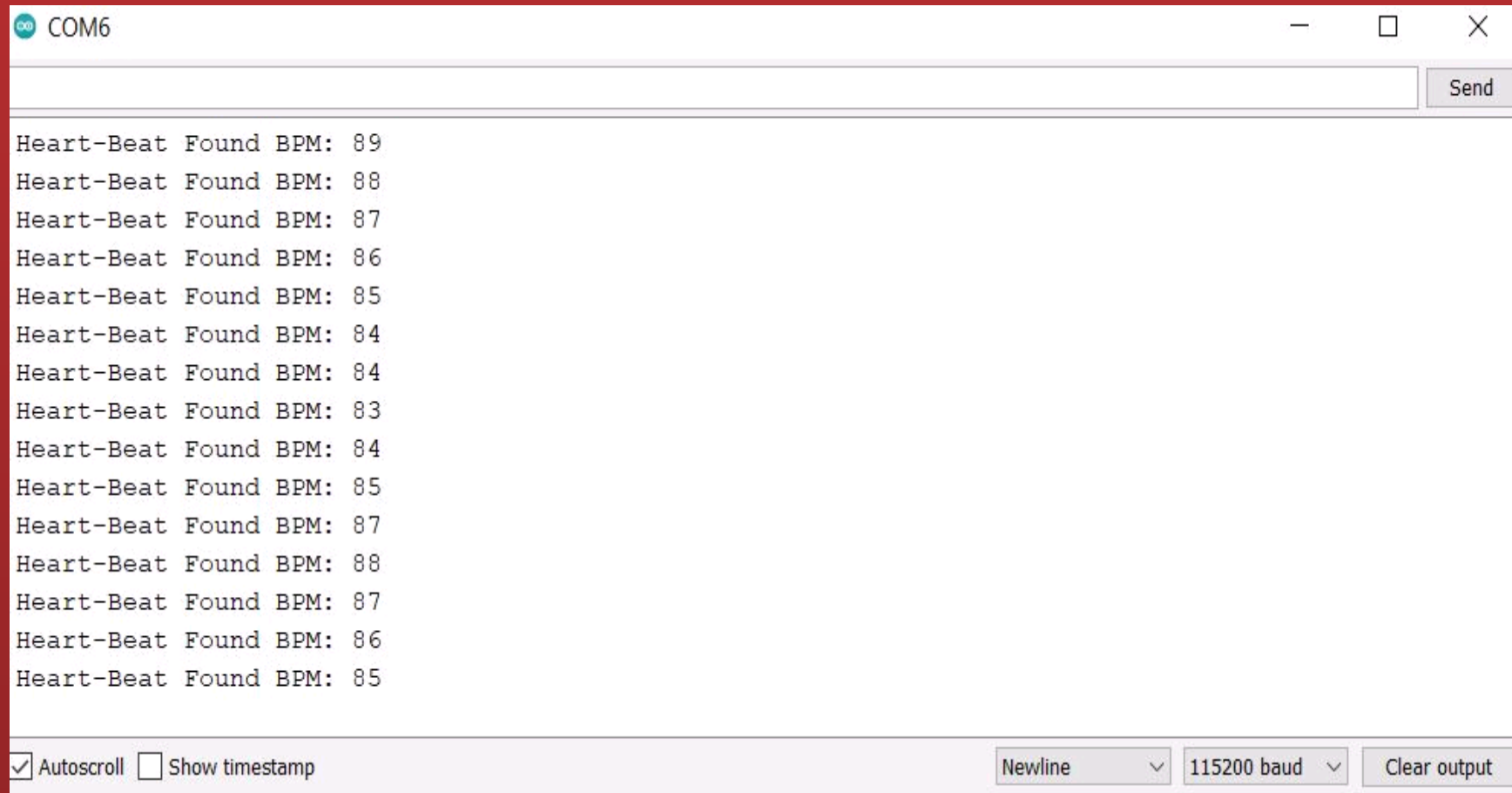


Fig 8: Serial monitor showing the output

WORK DONE SO FAR

- Heart beat monitoring system using Arduino will detect the heart beat using Hear beat sensor
- The Heart rate will be displayed on serial monitor with BPM readings and display the result on LCD

<https://drive.google.com/file/d/1vU8NbfCBR8MW8KtqylqdsaIXcE8qPA7A/view?usp=drivesdk>

REFERENCES

1. Muharrem Celebi . “ *Portable ECG Monitoring Device Design Based on ARDUINO* ”, Kartepe Vocational and Technical Anatolian High School, Kartepe, Kocaeli.
2. P. Srinivasan, A . Ayub Khan, T. Prabu, M. Manoj, M. Ranjan, K .Karthik , “ *Heart Beat Sensor Using Fingertip Through Arduino* ”, Journal of Critical Reviews . ISSN-2394-5125, Vol 7, Issue 7, 2020.
3. F M Yassin , N A Sani and S N Chin , 2019 J.Phys.: Conf.ser.1358 012041 , “ *Analysis of Heart Rate and Body Temperature from the wireless Monitoring System Using Arduino* ”.

4. . K.Sriram Datta , L. Mani Kumar, M. Kiran , M. Pavan Satya Arun ,
R.G V V D Pavan Kumar, “*Heart Beat Sensor Using Arduino*” Electrical and
Electronics Engineering, Pragati Engineering College, Surampalem , ISSN
NO: 2249-2976.

Website References:

1. <https://www.arduino.cc/en/guide/introduction>
2. <https://www.electronicshub.org/heartbeat-sensor-using-arduino-heart-rate-monitor/>
3. <https://images.app.goo.gl/CJ47CExywar9jspt5>

THANK
YOU