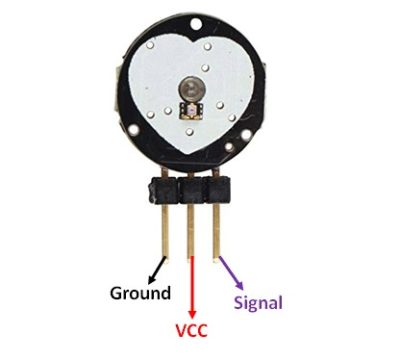
**Heart monitoring system using arduino**

Introduction

Heartbeat Sensor is an electronic device that is used to measure the heart rate i.e. speed of the heartbeat. Monitoring body temperature, heart rate and blood pressure are the basic things that we do in order to keep us healthy.

When a heartbeat occurs, blood is pumped through the human body and gets squeezed into the capillary tissues. Consequently, the volume of these capillary tissues increases. But in between the two consecutive heartbeats, this volume inside capillary tissues decreases. This change in volume between the heartbeats affects the amount of light that will transmit through these tissues. The pulse sensor module has a light that helps in measuring the pulse rate. When we place the finger on the pulse sensor, the light reflected will change based on the volume of blood inside the capillary blood vessels. This variation in light transmission and reflection can be obtained as a pulse from the output of the pulse sensor. This pulse can be then conditioned to measure heartbeat and then programmed accordingly to read as heartbeat count using Arduino.



**Components**

Arduino uno board

Pulse sensor

Breadboard

Connecting wires

**Application**

* Sleep Tracking
* Anxiety monitoring
* Remote patient monitoring/alarm system
* Health bands
* Advanced gaming consoles

Objective

During this activity ,you will help students to achieve following objectives

1. Understanding the principle and operation of pulse sensor
2. Design algorithm and flowchart for pulse sensor
3. Programming pulse Sensor using Arduino uno
4. Interfacing pulse sensor and LCD withArduino uno

Programming steps

1. Include pulse sensor library file
2. Include LCD libray file
3. Initialize pulse sensor input variable
4. Initialize threshld value
5. Read pulse sensor analog input continuosly
6. Display sensor values on LCD

Program

**#define USE\_ARDUINO\_INTERRUPTS true // Set-up low-level interrupts for most acurate BPM math.**

**#include <PulseSensorPlayground.h> // Includes the PulseSensorPlayground Library.**

**#include<LiquidCrystal.h>**

**LiquidCrystal lcd(7, 6, 5, 4, 3, 2);**

**// Variables**

**const int PulseWire = 0; // PulseSensor PURPLE WIRE connected to ANALOG PIN 0**

**const int LED13 = 13; // The on-board Arduino LED, close to PIN 13.**

**int Threshold = 550; // Determine which Signal to "count as a beat" and which to ignore.**

**// Use the "Gettting Started Project" to fine-tune Threshold Value beyond default setting.**

**// Otherwise leave the default "550" value.**

**PulseSensorPlayground pulseSensor; // Creates an instance of the PulseSensorPlayground object called "pulseSensor"**

**void setup() {**

**Serial.begin(9600); // For Serial Monitor**

**lcd.begin(20,4);**

**// Configure the PulseSensor object, by assigning our variables to it.**

**pulseSensor.analogInput(PulseWire);**

**pulseSensor.blinkOnPulse(LED13); //auto-magically blink Arduino's LED with heartbeat.**

**pulseSensor.setThreshold(Threshold);**

**// Double-check the "pulseSensor" object was created and "began" seeing a signal.**

**if (pulseSensor.begin()) {**

**Serial.println("We created a pulseSensor Object !"); //This prints one time at Arduino power-up, or on Arduino reset.**

**lcd.setCursor(0,0);**

**lcd.print(" Heart Rate Monitor");**

**}**

**}**

**void loop() {**

**int myBPM = pulseSensor.getBeatsPerMinute(); // Calls function on our pulseSensor object that returns BPM as an "int".**

**// "myBPM" hold this BPM value now.**

**if (pulseSensor.sawStartOfBeat()) { // Constantly test to see if "a beat happened".**

**Serial.println("♥ A HeartBeat Happened ! "); // If test is "true", print a message "a heartbeat happened".**

**Serial.print("BPM: "); // Print phrase "BPM: "**

**Serial.println(myBPM); // Print the value inside of myBPM.**

**lcd.setCursor(0,2);**

**lcd.print("HeartBeat Happened !"); // If test is "true", print a message "a heartbeat happened".**

**lcd.setCursor(5,3);**

**lcd.print("BPM: "); // Print phrase "BPM: "**

**lcd.print(myBPM);**

**}**

**delay(20); // considered best practice in a simple sketch.**

**}**

Hardware

Instruction

1.Connect the VCC pin of the Sensor to Arduino 5V Pin & GND to GND. 2.Connect the Analog output pin of the sensor to the A0 pin of the Arduino.

3.connect data pins of LCD (D7,D6,D5,D4) to digital pin 2,3,4,5 OF Arduino

4. connect VSS ,VEE and R/W pin to ground and enable and RS pin to VCC of arduino

