//FINAL YEAR PROJECT

```
#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>
int led = 13; // define the LED pin
int digitalPin = A5; // KY-028 digital interface
int digitalVal; // digital readings
//  Variables
const int PulseWire = A2;       // 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.
PulseSensorPlayground pulseSensor;  // Creates an instance of the
PulseSensorPlayground object called "pulseSensor"

LiquidCrystal lcd(7, 12, A0, A1, 9, 8, 2, A3, 5, 4, 3);  /* For 8-bit mode */
// Otherwise leave the default "550" value.
void setup()
{
pinMode(led, OUTPUT);
pinMode(digitalPin, INPUT);
pinMode(10, INPUT); // Setup for leads off detection LO +
pinMode(11, INPUT); // Setup for leads off detection LO -

Serial.begin(9600);          // For Serial Monitor
// set up the LCD's number of columns and rows:
lcd.begin(16, 2);
lcd.setCursor(0,0);
lcd.print("Health Care");
// 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.clear();
lcd.print("BPM:"); // BEATS PER MINUTE
}
}

void loop()
{
while(1)
{
ECG();
delay(100);
temperature();
delay(100);
heart_rate();
delay(100);
}
}
void ECG()
{
if((digitalRead(10) == 1)||(digitalRead(11) == 1)){
Serial.println('!');
//lcd.print('-');
}
else{
unsigned int AnalogValue;
AnalogValue = analogRead(A4);
lcd.setCursor(0,1);
lcd.print(AnalogValue);
}
//Wait for a bit to keep serial data from saturating
delay(1);
}
void temperature()
{
// Read the digital interface
digitalVal = digitalRead(digitalPin);
if(digitalVal == HIGH) // if temperature threshold reached
{
digitalWrite(led, LOW); // turn ON Arduino's LED
}
else
{
digitalWrite(led, HIGH); // turn OFF Arduino's LED
}
}
void heart_rate()
{
```

```
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.clear();
lcd.setCursor(0,0);
lcd.print("BPM:");
lcd.setCursor(4,0);
lcd.print(myBPM);
}

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

}
```