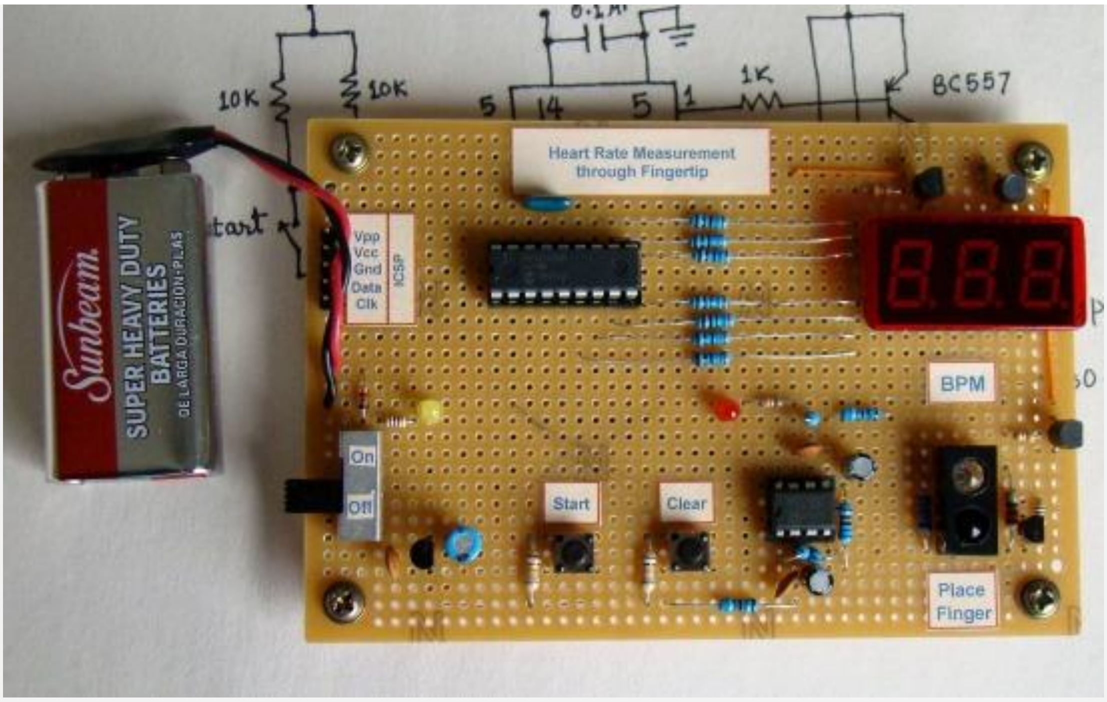
Introduction

Heart rate measurement indicates the soundness of the human cardiovascular system. This project demonstrates a technique to measure the heart rate by sensing the change in blood volume in a finger artery while the heart is pumping the blood. It consists of an infrared LED that transmits an IR signal through the fingertip of the subject, a part of which is reflected by the blood cells. The reflected signal is detected by a photo diode sensor. The changing blood volume with heartbeat results in a train of pulses at the output of the photo diode, the magnitude of which is too small to be detected directly by a microcontroller. Therefore, a high gain, active low pass filter is designed using Operational Amplifier (OpAmp) to filter and amplify the signal to appropriate voltage level so that the pulses can be counted by a microcontroller. The heart rate is displayed on a 3 digit seven segment display. The microcontroller used in this project is PIC16F628A.

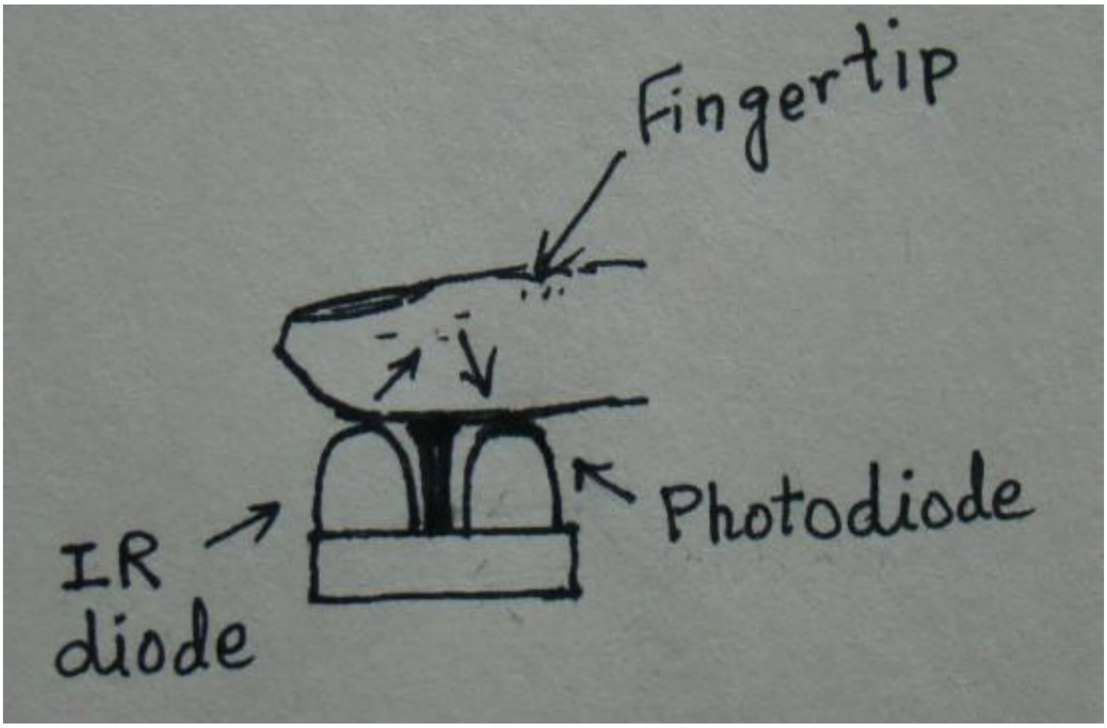


Heart rate measuring device using PIC16F628A

Theory

Heart rate is the number of heartbeats per unit of time and is usually expressed in beats per minute (bpm). In adults, a normal heart beats about 60 to 100 times a minute during resting condition. The resting heart rate is directly related to the health and fitness of a person and hence is important to know. You can measure heart rate at any spot on the body where you can feel a pulse with your fingers. The most common places are wrist and neck. You can count the number of pulses within a certain interval (say 15 sec), and easily determine the heart rate in bpm.

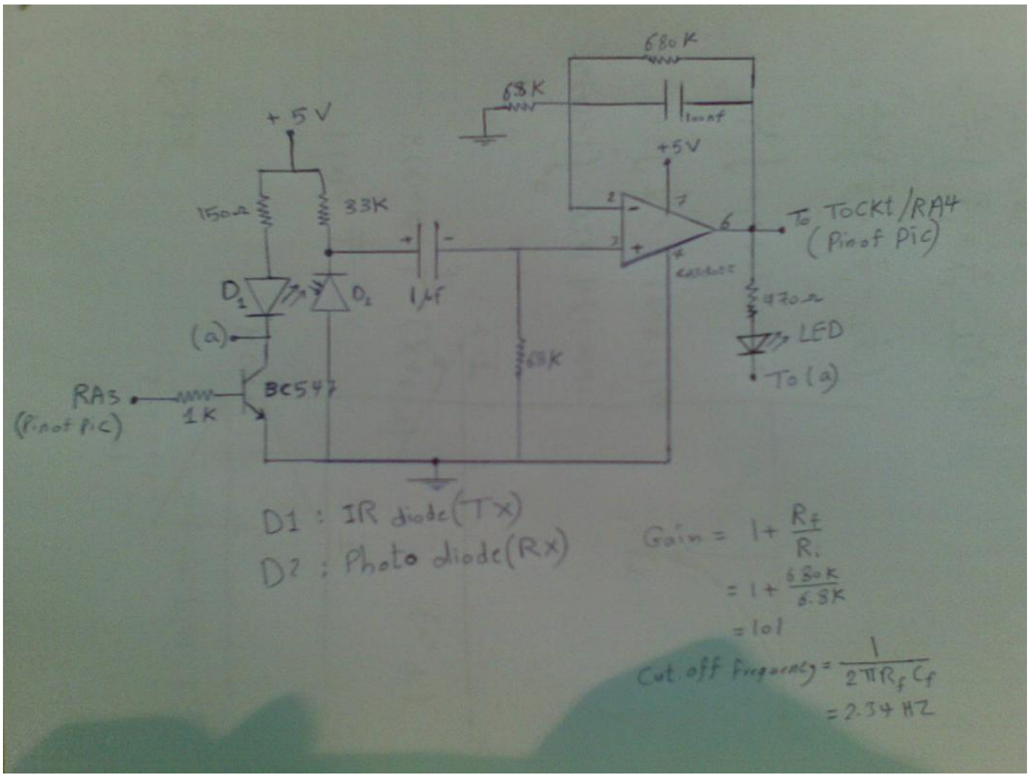
This project describes a microcontroller based heart rate measuement system that uses optical sensors to measure the alteration in blood volume at fingertip with each heart beat. The sensor unit consists of an infrared light-emitting-diode (IR LED) and a photodiode, placed side by side as shown below. The IR diode transmits an infrared light into the fingertip (placed over the sensor unit), and the photodiode senses the portion of the light that is reflected back. The intensity of reflected light depends upon the blood volume inside the fingertip. So, each heart beat slightly alters the amount of reflected infrared light that can be detected by the photodiode. With a proper signal conditioning, this little change in the amplitude of the reflected light can be converted into a pulse. The pulses can be later counted by the microcontroller to determine the heart rate.



Fingertip placement over the sensor unit

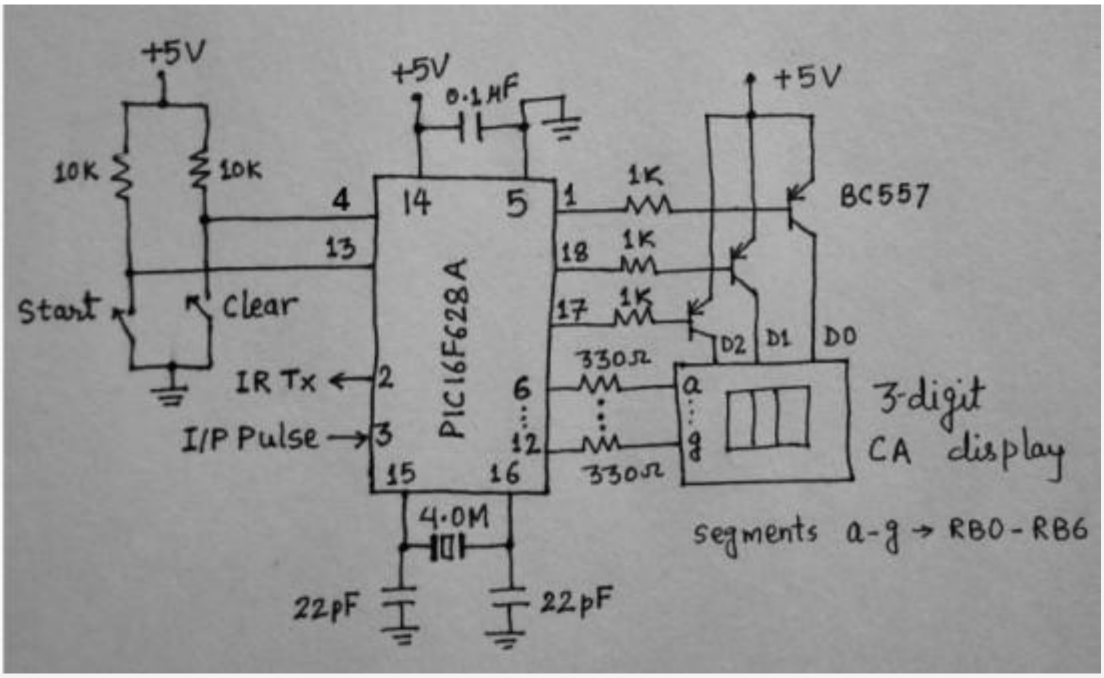
Circuit Diagram

The signal conditioning circuit consists of two identical active low pass filters with a cut-off frequency of about 2.5 Hz. This means the maximum measurable heart rate is about 150 bpm. The operational amplifier IC used in this circuit is CA3130EZ. The filtering is necessary to block any higher frequency noises present in the signal. The gain of filter is set to 101. A 1 uF capacitor at the input of each stage is required to block the dc component in the signal. The equations for calculating gain and cut-off frequency of the active low pass filter are shown in the circuit diagram. The amplifier/filter provides sufficient gain to boost the weak signal coming from the photo sensor unit and convert it into a pulse. An LED connected at the output blinks every time a heart beat is detected. The output from the signal conditioner goes to the T0CKI input of PIC16F628A.



IR sensors and signal conditioning circuit

The control and display part of the circuit is shown below. The display unit comprises of a 3-digit, common anode, seven segment module that is driven using multiplexing technique. The segments a-g are driven through PORTB pins RB0-RB6, respectively. The unit's, ten's and hundred's digits are multiplexed with RA2, RA1, and RA0 port pins. A tact switch input is connected to RB7 pin. This is to start the heart rate measurement. Once the start button is pressed, the microcontroller activates the IR transmission in the sensor unit for 15 sec. During this interval, the number of pulses arriving at the T0CKI input is counted. The actual heart rate would be 4 times the count value, and the resolution of measurement would be 4. You can see the IR transmission is controlled through RA3 pin of PIC16F628A. The microcontroller runs at 4.0 MHz using an external crystal. A regulated +5V power supply is derived from an external 9 V battery or adapter using an LM7805 regulator IC.



Microcontroller and Display Circuit

Software

The firmware does all the control and computation operation. In order to save the power, the sensor module is not activated continuously. Instead, it is turned on for 15 sec only once the start button is pressed. The pulses arriving at T0CKI are counted through Timer0 module operated in counter mode without prescaler. The complete program written for MikroC compiler is provided below.

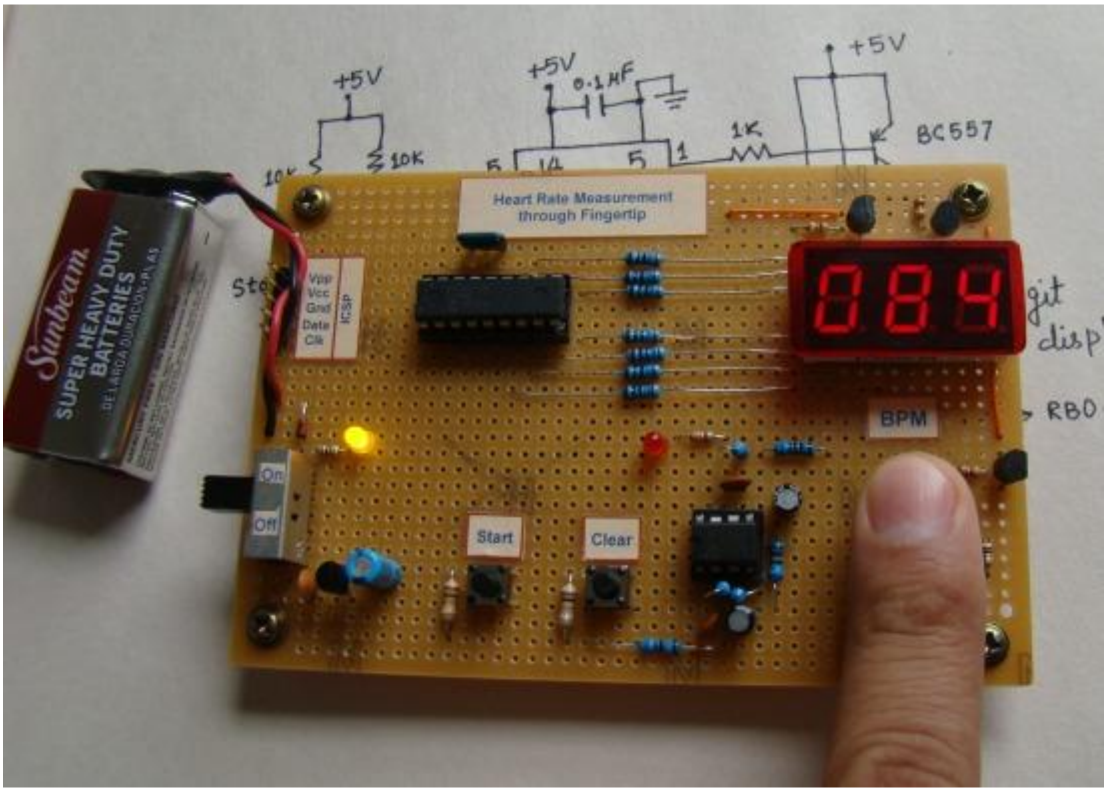
```
/*
  Project: Measuring heart rate through fingertip
  Copyright @ Rajendra Bhatt
  January 18, 2011
  PIC16F628A at 4.0 MHz external clock, MCLR enabled
*/

sbit IR_Tx at RA3_bit;
sbit DD0_Set at RA2_bit;
sbit DD1_Set at RA1_bit;
sbit DD2_Set at RA0_bit;
sbit start at RB7_bit;
unsigned short j, DD0, DD1, DD2, DD3;
unsigned short pulserate, pulsecount;
unsigned int i;
//-------------- Function to Return mask for common anode 7-seg.
display
unsigned short mask(unsigned short num) {
 switch (num) {
 case 0 : return 0xC0;
 case 1 : return 0xF9;
 case 2 : return 0xA4;
 case 3 : return 0xB0;
```

```
 case 4 : return 0x99;
 case 5 : return 0x92;
 case 6 : return 0x82;
 case 7 : return 0xF8;
 case 8 : return 0x80;
 case 9 : return 0x90;
 } //case end
}

void delay_debounce(){
 Delay_ms(300);
}

void delay_refresh(){
 Delay_ms(5);
}

void countpulse(){
 IR_Tx = 1;
 delay_debounce();
 delay_debounce();
 TMR0=0;
 Delay_ms(15000);  // Delay 15 Sec
 IR_Tx = 0;
 pulsecount = TMR0;
 pulserate = pulsecount*4;
}

void display(){
  DD0 = pulserate%10;
  DD0 = mask(DD0);
  DD1 = (pulserate/10)%10;
  DD1 = mask(DD1);
  DD2 = pulserate/100;
  DD2 = mask(DD2);
  for (i = 0; i<=180*j; i++) {
    DD0_Set = 0;
    DD1_Set = 1;
    DD2_Set = 1;
    PORTB = DD0;
    delay_refresh();
    DD0_Set = 1;
    DD1_Set = 0;
    DD2_Set = 1;
    PORTB = DD1;
    delay_refresh();
    DD0_Set = 1;
    DD1_Set = 1;
    DD2_Set = 0;
    PORTB = DD2;
    delay_refresh();
    }
  DD2_Set = 1;
}
```

```
void main() {
 CMCON = 0x07;     // Disable Comparators
 TRISA = 0b00110000; // RA4/T0CKI input, RA5 is I/P only
 TRISB = 0b10000000; // RB7 input, rest output
 OPTION_REG = 0b00101000; // Prescaler (1:1), TOCS =1 for counter
mode
 pulserate = 0;
 j = 1;
 display();
 do {
  if(!start){
   delay_debounce();
   countpulse();
   j= 3;
   display();
  }
 } while(1);  // Infinite loop
}
```

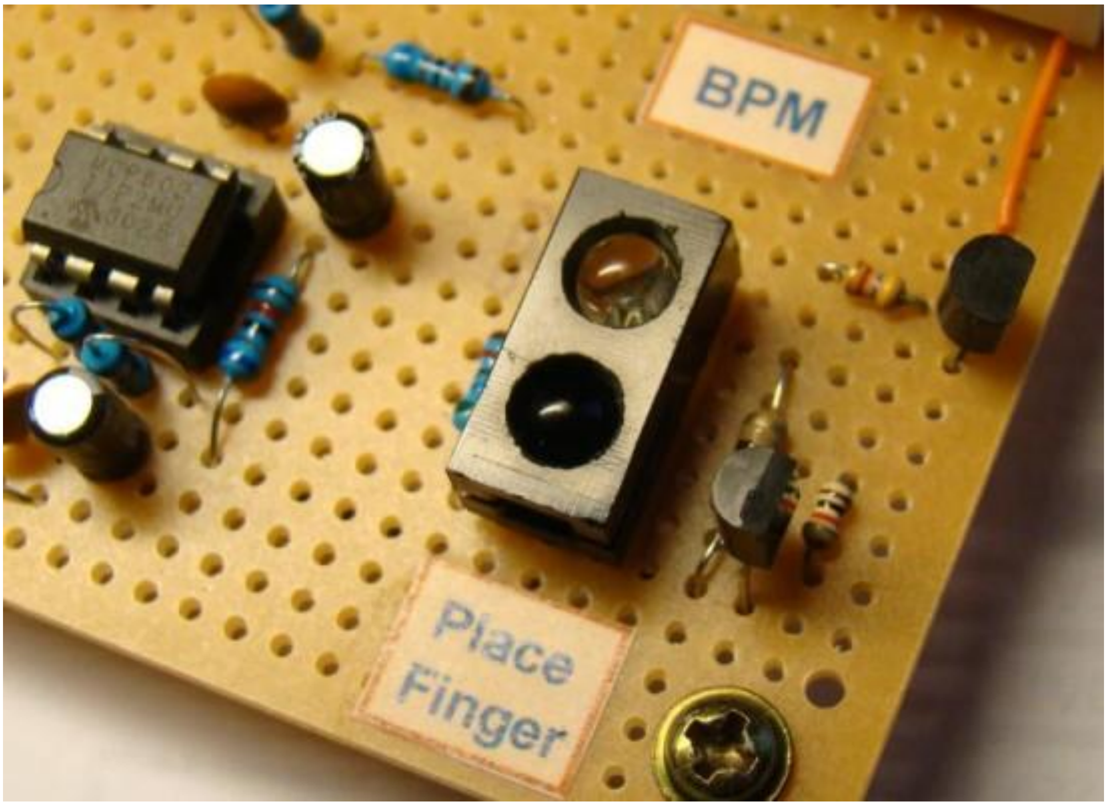
Output

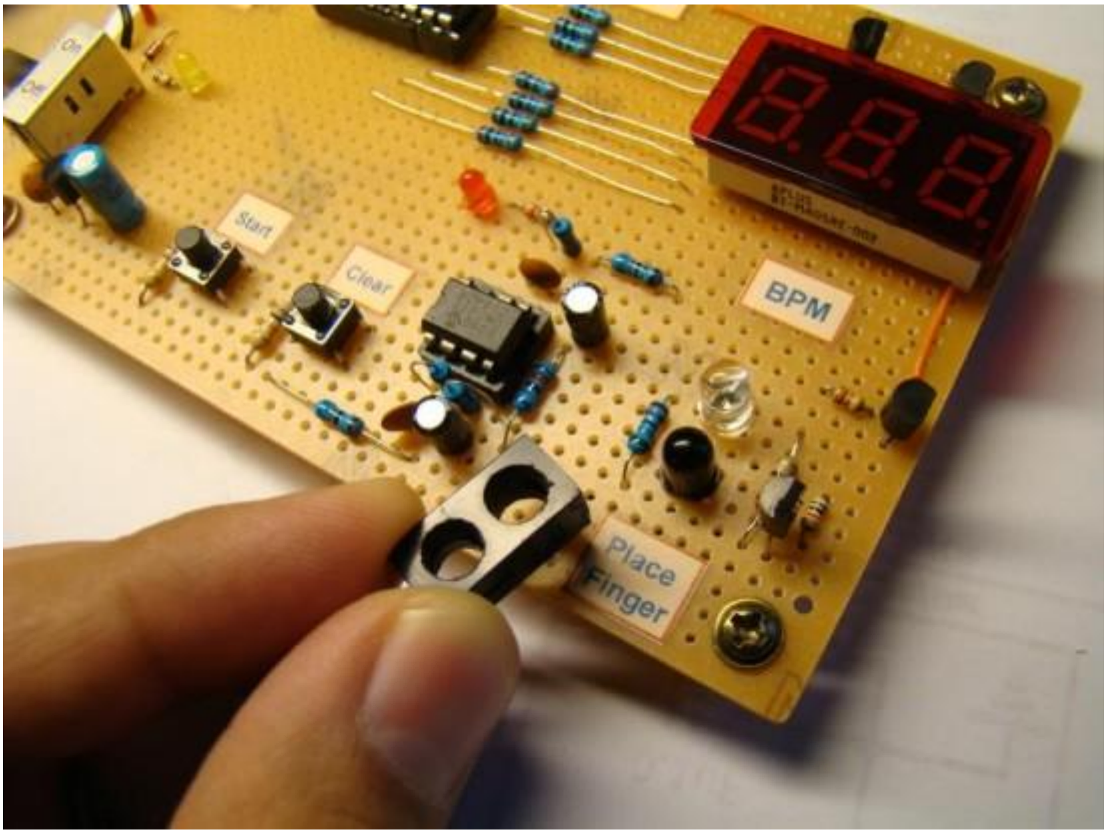
The use of this device is very simple. Turn the power on, and you will see all zeros on display for few seconds. Wait till the display goes off. Now place your forefinger tip on the sensor assembly, and press the start button. Just relaxed and don't move your finger. You will see the LED blinking with heart beats, and after 15 sec, the result will be displayed.



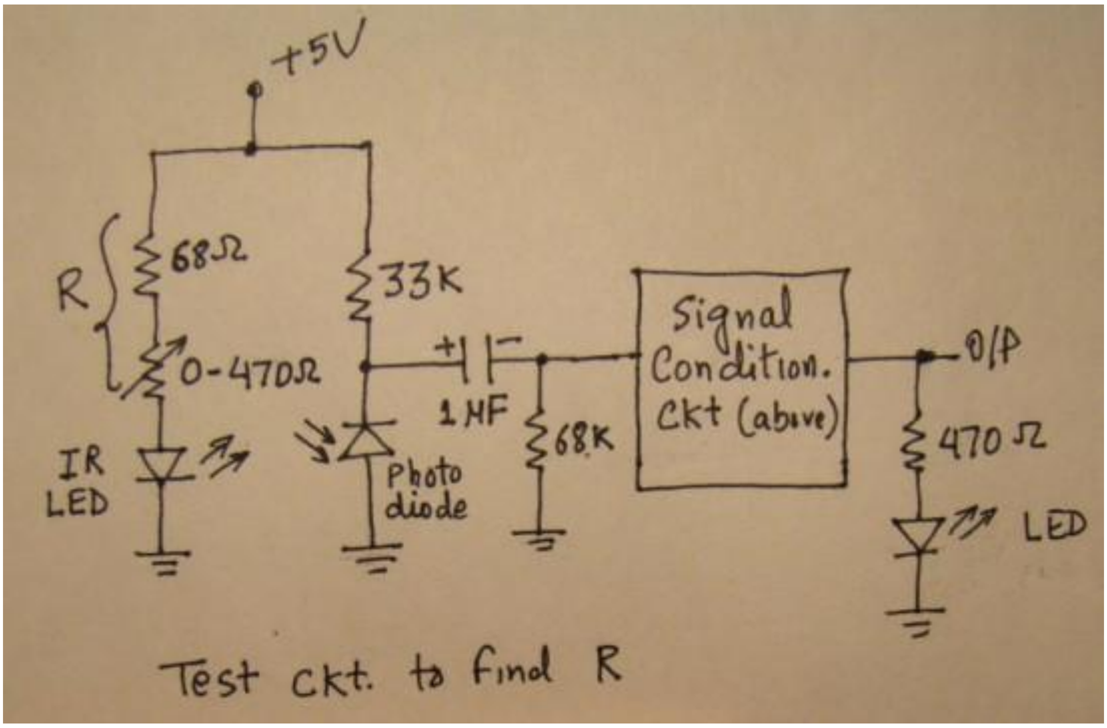
Important note:

The harder part in this project is the signal conditioning circuit that uses active low pass filters using OpAmps to boost the weak reflected light signal detected by the photo diode. The IR transmitting diode and the photo diode are placed closely but any direct crosstalk between the two are avoided. Look at the following pictures to see how I have blocked the direct infrared light from falling into the adjacent photo diode. Besides, surrounding the sensor with an opaque material makes the sensor system more robust to changing ambient light condition.





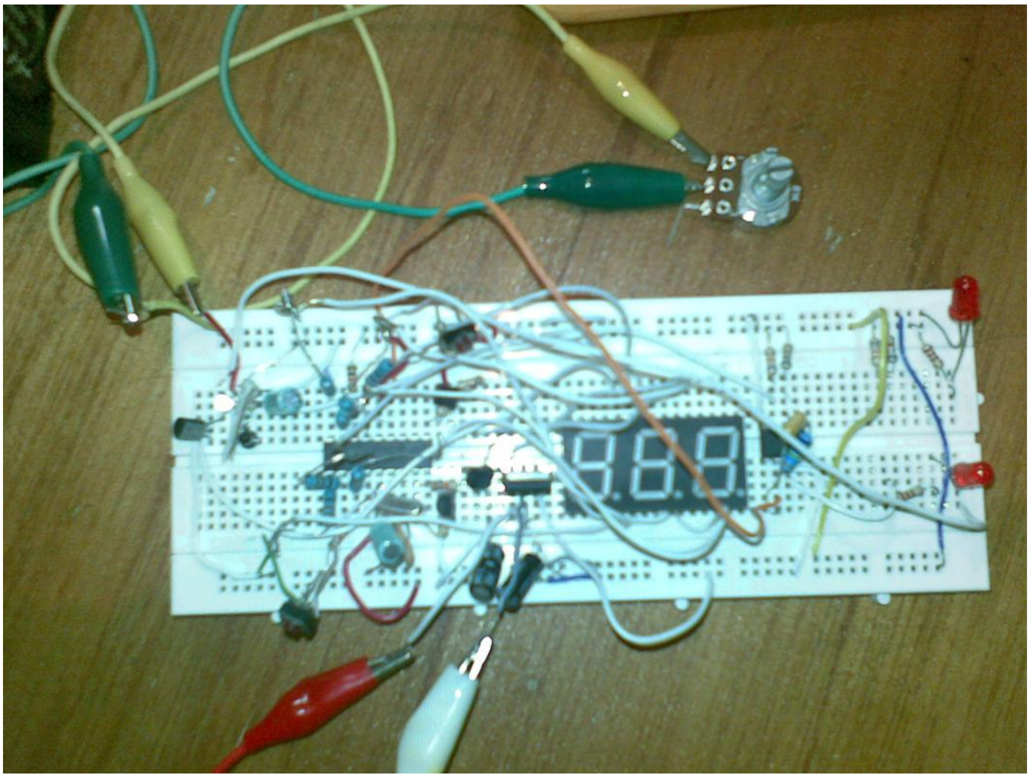
The 150 Ω resistance in series with the IR diode is to limit the current and hence the intensity of the transmitted infrared light. The intensity of IR light should not be too high otherwise the reflected light will be sufficient enough to saturate the photo detecting diode all the time and no signal will exist. The value of this current limiting resistor could be different for different IR diodes, depending upon their specifications. Here’s my practical test circuit that I used to find the appropriate value of the series resistor for the IR diode I used.



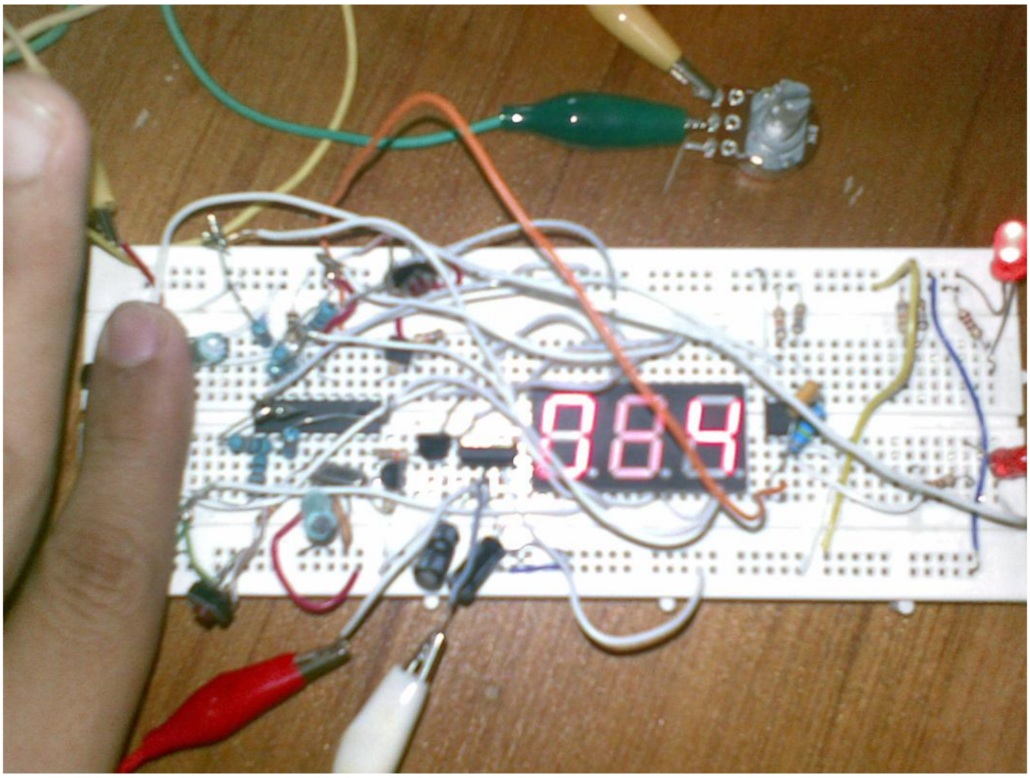
First I used a 68 Ω resistor with a 470 Ω potentiometer in series with the IR diode. Placing a fingertip over the sensor assembly, I slowly varied the potentiometer till I found the output LED blinking with heartbeat. Then I measured the equivalent resistance R and replaced the 68 Ω and the potentiometer with a single resistor closest to R. But I also keep the potentiometer in my circuit so that i can always adjust it when needed. You should keep your fingertip very still over the sensor while testing. Once you see the pulses at the output of the signal conditioning circuit, you can feed them to a microcontroller to count and display.

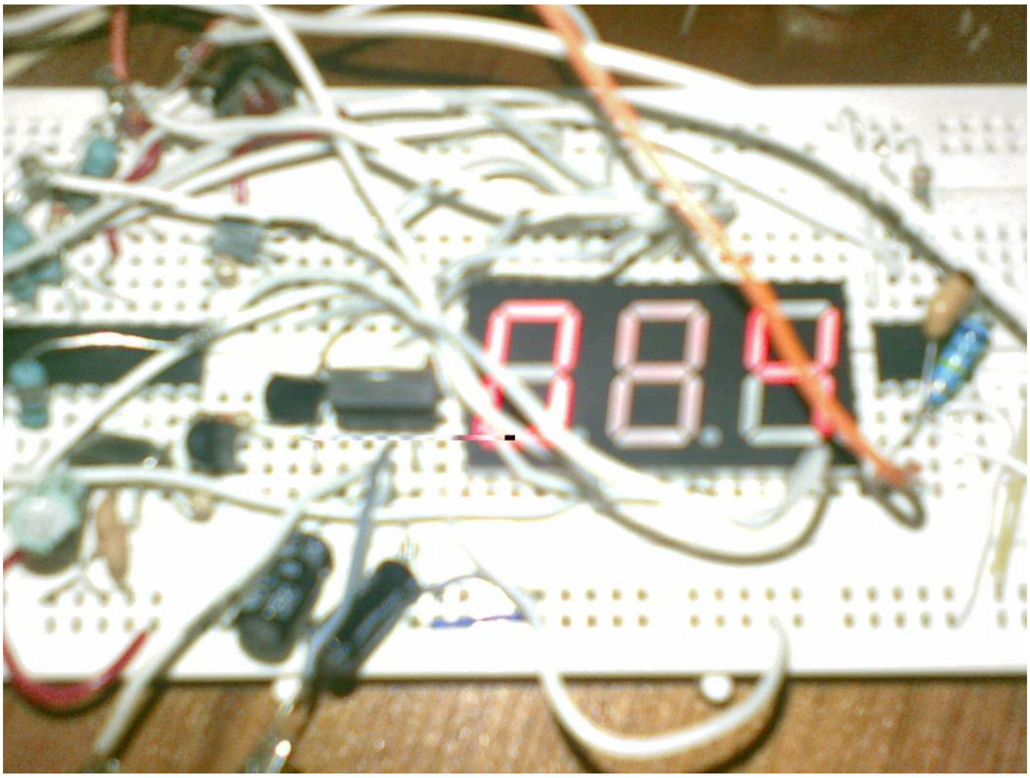
Circuit on breadboard

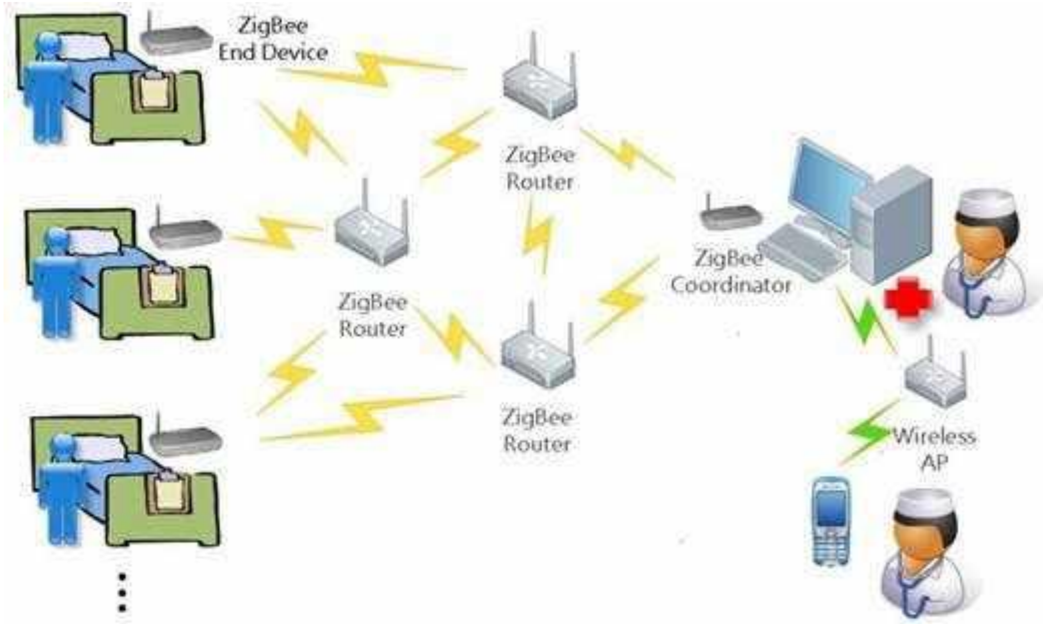
1-before power on:



2-after power on:


084


084


ZigBee
End Device
ZigBee
Router
ZigBee
Router
ZigBee
Router
ZigBee
Coordinator
Wireless
AP