MICROCONTROLLER-BASED HEART RATE METER

Prof. K. Padmanabhan

Heart rate can be measured either by the ECG waveform or by the blood flow into the finger (pulse method). The pulse method is simple and convenient. When blood flows during the systolic stroke of the heart into the body parts, the finger gets its blood via the radial artery on the arm. The blood flow into the finger can be sensed photoelectrically.

To count the heart beats, here we use a small light source on one side of the finger (thumb) and observe the change in light intensity on the other side. The blood flow causes variation in light intensity reaching the light-dependent resistor (LDR), which results in change in signal strength due to change in the resistance of the LDR.

Circuit description

Fig. 1 shows the circuit of microcontroller-based heart-rate meter. The setup uses a 6V electric bulb for light illumination of flesh on the thumb behind the nail and the LDR as detector of change in the light intensity due to the flow of blood. The photo-current is converted into voltage and amplified by operational amplifier IC LM358 (IC1). The detected signal is given to the non-inverting input (pin 3) and its output is fed to another non-inverting input (pin 5) for squaring and amplification. Output pin 7 provides detected heartbeats to pin 12 of the microcontroller. Preset VR1 is used for sensitivity and preset VR2 for trigger-level settings.

Microcontroller IC AT89C2051 (IC2) is at the heart of the circuit. It is a 20-pin, 8-bit microcontroller with 2 kB of Flash programmable and erasable read-only memory (PEROM), 128 bytes of RAM, 15 input/output (I/O) lines, two 16-bit timer/counters, a five-vector two-level interrupt architecture, a full-duplex serial port, a precision analogue comparator, on-chip oscillator and clock circuitry.

Port-1 pins P1.7 through P1.2, and port-3 pin P3.7 are connected to input pins 1 through 7 of IC ULN2003 (IC3), respectively. These pins are pulled-up with 10-kilo-ohm resistor network RNW1. They drive all the segments of the 7-segment display with the help of inverting buffer IC3.

The displays are selected through port pins P3.0, P3.1 and P3.2 of the microcontroller (IC2). Port pins P3.0 down through P3.2 are connected to the base of transistors T3 through T1, respectively. Pin 6 of IC2 goes low to drive transistor T1 into saturation and provide supply to the common-anode pin (either pin 3 or pin 8) of DIS1. Similarly, transistors T2 and T3 drive common-anode pins 3 or 8 of 7-segment displays DIS2 and DIS3, respectively. Only three 7-segment displays are used.

IC2 provides segment-data and display-enable signals simultaneously in time-division-multiplexed mode for displaying a particular number on the 7-segment display unit. Segment-data and display-enable pulses for the display are refreshed every 5 ms. Thus the display appears to be continuous, even though it lights up one by one.

Switch S2 is used to manually reset the microcontroller, while the power-on reset signal for the microcontroller is derived from the combination of capacitor C4 and resistor R8. An 11.0592MHz crystal is used to generate the basic clock frequency for the microcontroller. The circuit is powered by a 6V battery.

Port pin P3.6 of the microcontroller is internally available for software checking. This pin is actually the output of the internal analogue comparator, which is available internally for comparing the two analogue levels at pins 12 and 13. As pins 12 and 13 of IC2 can work as an analogue comparator, these are used for sensing the rise and fall of the pulse waveform and thereby evaluate the time between two peaks and hence the beat rate.

The output of the pulse pick-up preamplifier is fed to pin 12 of the microcontroller. Pin 13 of the microcontroller is connected to the preset for reference-level setting of the comparator. Thus voltages at pins 12 and 13 are always compared. The signal rise and the fall at pin 12 are sensed...
The internal timer of the microcontroller is used to find the time taken for one wavelength. This time is converted into the heart beat rate in beats per minute by a pre-calculated look-up table. The program notes the time between the high-to-low and low-to-high transitions of the wave. This time in microseconds is converted in steps of 4 ms for comparison with the values already stored in the look-up table. This number is used to find (from the look-up table) the heart rate in beats per minute. The number so obtained is converted into a 3-digit number in binary-coded decimal (BCD) form. The same is output to the 7-segment LED displays in a multiplexed manner. The display shows the rate for a while and proceeds to another measurement. Thus beat rates obtained from time to time are visible on the display.

**Construction and testing**

The arrangement for heart beat rate detection is shown in Fig. 2. Purchase a plastic 'T' tube from an electrical parts shop. The tube should be about 5cm long and have a diameter of 1.5 cm. House the electric bulb into the left tube and the LDR (soldered on a small PCB) into the right tube. Fit shields on both sides of the tube to maintain darkness for better performance. Connect the 6V battery supply to the bulb and the LDR to the circuit board via a shielded cable.

For heart beat detection, which can be seen on a cathode ray oscilloscope (CRO), insert your thumb with the nail facing the LDR inside the T-tube. Shaking the thumb will change the level of signal from the previous
the levels of sensitivity, trigger and voltage reference for the comparator by using presets VR1, VR2 and VR3, respectively.

Hold the thumb steady and observe the heart beat rate on the display. The rate may vary and may not be exactly steady. For instance, normally, the rate can vary between 60 and 100.

Since this is a beat-to-beat measurement and not an average over a time period of one minute, variation is expected. However, when the reading shows high value at times, say, 140, it may be due to unusual mains hum picked up by the transducer. To suppress it, place a separate capacitor of 100 µF across the 5V supply.

An actual-size, single-side PCB for the microcontroller-based heart-rate meter is shown in Fig. 4 and its component layout in Fig. 5.

**Software**

The software is written in Assembly language and assembled using ASM51 cross-assembler. The Intel hex code is generated and burnt into the microcontroller chip by using a suitable programmer. The software is well commented and easy to understand.

Place the circuit components and IC bases on the PCB board. Check the pulse pick-up through the CRO at output pin 7 of IC1 (refer Fig. 3). Insert the programmed microcontroller and other ICs into the IC bases. Set

---

Fig. 3: Waveform of heartbeat detection

Fig. 4: A single-side, actual-size PCB layout for microcontroller-based heart-rate meter

Fig. 5: Component layout for the PCB
The timer does the job of finding the time between two successive pulse waveform points. Since the comparator within the microcontroller IC knows the point of crossing of the wave with the DC line determined by preset VR3, the three crossings follow one after another and at the end of the third crossing the time is read from the time-count register. This time is then converted in terms of the number of 4ms intervals. From the number of such 4ms units, the number of beats per minute is determined from the look-up table already stored in the same memory starting from the label ‘table’ in the program listing.

**HEARTRT.ASM**

```assembly
ORG 00H
AJMP 30H
ORG 0BH ; TIMER 0 INTERRUPT
VECTOR
AJMP TIM0ISR ; Timer 0 Interrupt service routine address
ORG 30H
MOV SP,#60H ; set stack pointer
MOV P3,#0FFH ; set all port 3 bits high to enable inputs also
MOV P1,#03H ; set port 1
MOV TMOD,#01100001B
MOV TL0,#0F0H
BEG: MOV TH0,#0F0H ; TIMER 1 - MODE 2 COUNTER, TIMR-0 TO MODE 1
REG.0 IS SET TO foo0, GIVES 4ms
MOV P3,#0FFH ; set all
MOV P1,#03H
SETB EA
SETB ET0
CLR 20H ; flag to burn for some time
; Timer 0 Interrupt Service Routine
AJMP TIM0ISR

ORG 0BH ; TIMER 0 INTERRUPT
SJMP BEG

ORG 00H
; CALL REFRESH1
DJNZ 50H,REFR
REFR: CALL REFRESH1
; so many times
CIRINT: CLR ET0
; more interrupts
JC TACHY
;jmp beg
TACHY: MOV R2,#10
; to show on LED pin 8 that rate is too high
JMP BEG
; Hex to BCD
; This routine is for 16 bit Hex to BCD Conversion for 8051 Microcontroller
; ; MOV A,R1,R2
; ; table looked up
; ; rate is now in r2
; ; high byte is zero
; ; make it in BCD format
; ; show the value on LED
; ; refresh a 100 times (.5 sec)
; ; DISP1:
; DISP1:
```

**CONSTRUCTION**

The source code and other relevant files of this article are included in this month’s EFY-CD.

**EFY note.**
; MOV A,#07H
; MOV P3,A ;
; setb p3.0
; prevent shadow
; setb p3.1
; to extinguish the digit after that time
; setb p3.2
s6: ; ADD 1 TO SECONDS
DA A
MOV R2,A
setb 20h
; seconds over
K1A:
; pop acc
pop psw
RETI ; INTERRUPT RETURN INSTRUCTION

;table: