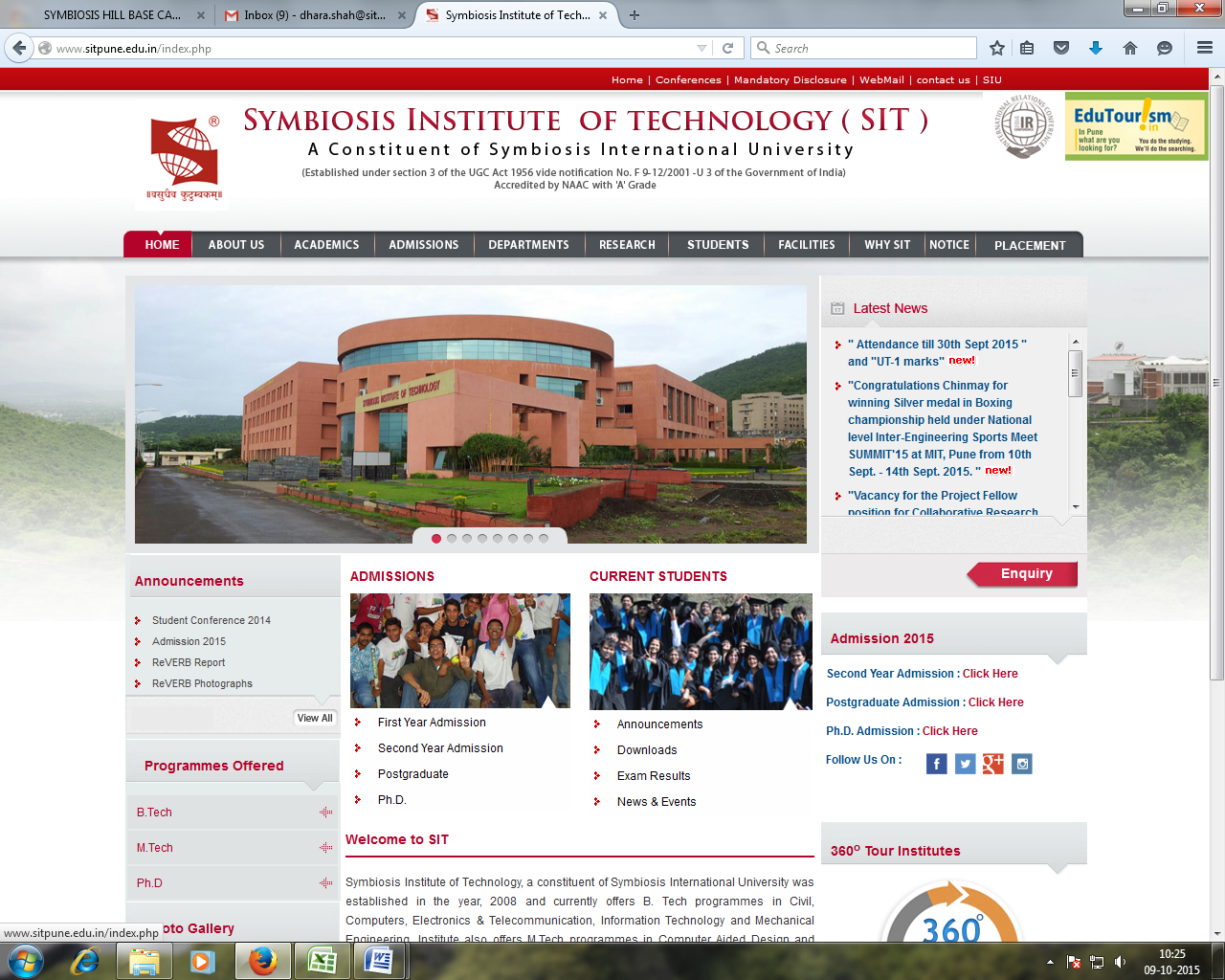
**A PROJECT REPORT ON**

**TEMPERATURE AND HEART RATE**

**MONITORING SYSTEM**

**Electronics & Telecommunication**



**Under the Guidance of**

**DR. BHASKAR THAKKER**

**SYMBIOSIS INSTITUTE OF TECHNOLOGY, PUNE – 412115**

**(A CONSTITUENT OF SYMBIOSIS INTERNATIONAL UNIVERSITY)**

**2018-19**

ACKNOWLEDGEMENT

We would like to express our special thanks to our subject teacher, **Dr. Bhaskar Thakker** for his constant guidance as well as for providing necessary information and direction to approach to our project **“Temperature And Heart Rate Monitoring System**” which helped us to

do a lot of useful research.

Also, a special thanks to the lab assistant officials for their cooperation and allowing us to use the lab for implementation of our project.

CERTIFICATE

This is to certify that the project entitled, “ **Temperature And Heart Rate Monitoring System**” has been done by : **Akshay Bhagwani** , **Amisha Kuwarbi** and **Vennela Bandla** of Electronics & Telecommunication Department as prescribed by the Symbiosis Institute of Technology, Pune in the academic year 2018-19 under the supervision of **Dr. Bhaskar Thakker.**

**Date: Sign of Guide:**

ABSTRACT

In this project, we made a circuit that basically measures the heart rate using heart rate sensor and monitors the surrounding temperature using temperature sensor (LM35) respectively and displays the value on a 16x2 LCD. In this, we interfaced an LCD, LM35 and heart rate sensor with LPC2148. The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The sensed temperature is in analog form so an analog to digital conversion function is used in the program to display it on an LCD.

For heart pulse rate sensing we used a sensor that gives digital output so the sensor is directly configured with LPC2148 and displayed the output on the second line on the LCD.

**Embedded System Design**

TY B. Tech. E&TC | Batch 2016-20

**Mini Project**

|  |  |  |
| --- | --- | --- |
| **Name of the Student:** Akshay Bhagwani | | **PRN:** 16070123013 |
| **Name of the Student:** Amisha Kuwarbi | | **PRN:** 16070123015 |
| **Name of the Student:** Vennela Bandla | | **PRN:** 16070123030 |
| **Academic Year :** 2018-19 | **Division:** A |  |

**Title:** Temperature and heart pulse rate monitoring using LPC2148 and displaying the same on LCD.

**Objective:** To detect surrounding temperature and monitor heart pulse rate by interfacing LM35 and Heart Rate sensor (TCRT1000) with LPC2148 and display their values on a 16x2 LCD.

**Equipment Needed:**

1. LPC2148 Module
2. Temperature Sensor – LM35DT
3. Heart Rate Sensor
4. 16x2 LCD
5. Jumper Wires
6. Breadboard
7. Software used – Keilµvision4
8. Software used – Flash Magic

**ARM Peripherals used:**

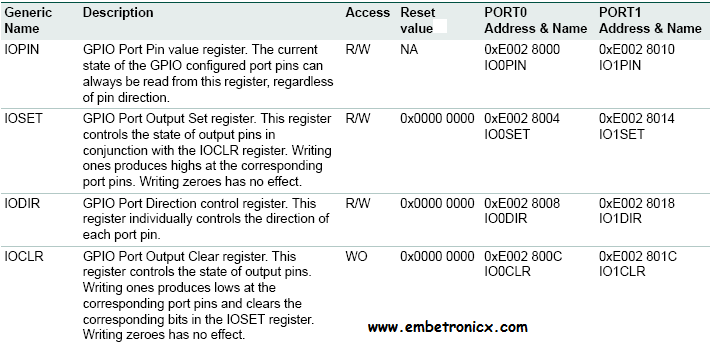
* Analog to Digital Converter.
* General Purpose I/O
* Timer/Counter

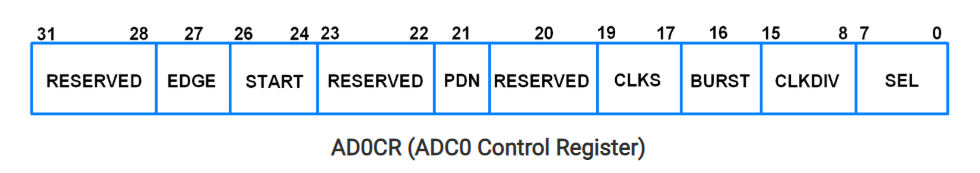
**Peripherals Description:**

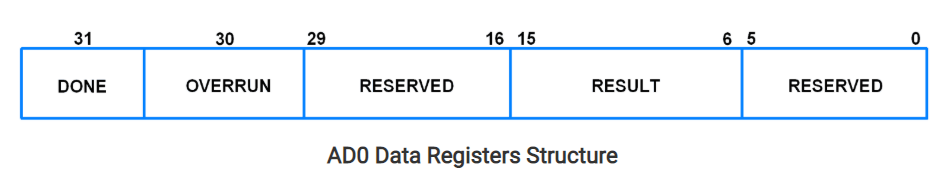
* **Analog to Digital Converter:**
* The A/D converter present in LPC2148 is 10-bit successive approximation converter, with a conversion time of 2.44µsec.
* The A/D converter in LPC2148 has 8 multiplexed inputs.
* **General Purpose I/O:**
* The GPIO pins are controlled by four registers, namely- IOSET, IOCLR, IODIR and IOPIN.
* Each GPIO pin is controlled by a bit in each of the four GPIO registers. These bits are data direction(IODIR), set(IOSET), clear(IOCLR) and pin status(IOPIN).
* **Timer/Counter:**
* The LPC2148 has two general purpose timers TIMER0 and TIMER1.
* The timers are based around a 32-bit timer counter with a 32-bit prescaler.
* The clock source for all the timers is VLSI peripheral clock PCLK.

**Registers Description:**

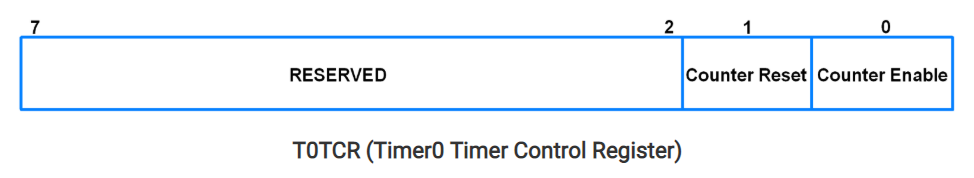
* **IOxPIN:** GPIO Port Pin value register. The current state of the GPIO configured port pins can always be read from this register, regardless of pin direction.
* **IOxDIR:** GPIO port direction control register. This register individually controls the direction of each port pin.
* **IOxSET:** GPIO Port Output Set register. This register controls the state of output pins in conjunction with the IOCLR register. Writing ones produces highs at the corresponding port pins. Writing zeroes has no effect.
* **IOxCLR:** GPIO Port Output Clear register. This register controls the state of output pins. Writing ones produces lows at the corresponding port pins and clears the corresponding bits in the IOSET register. Writing zeroes has no effect.

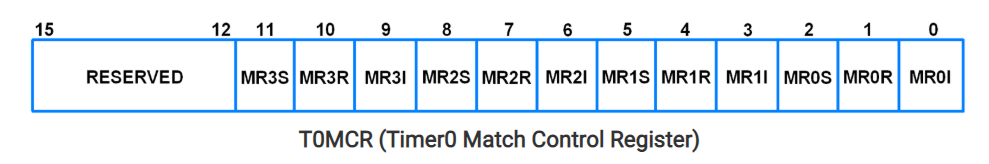


* **PINSEL:** The configuration register is called PINSEL and is classified in to three registers: PINSEL0, PINSEL1 and PINSEL2. These configuration registers are of 32-bit wide. Any pin on the LPC2148 can have a maximum of 4 functions. Hence in order to select one of the four functions, two corresponding bits of the PINSEL register are needed. So, a 32-bit PINSEL register can control 16 pins with 2-bits to control each pin.PINSEL0 controls PORT0 pins P0.0 to P0.15, PINSEL1 controls PORT0 pins P0.16 to P0.31 and PINSEL2 controls PORT1 pins P1.16 to P1.31.
* **AD0CR:** AD0CR is a 32-bit register. This register must be written to select the operating mode before A/D conversion can occur. It is used for selecting channel of ADC, clock frequency for ADC, number of clocks or number of bits in result, start of conversion and few other parameters. 
* **AD0DR0:** It is is a 32-bit register. It holds the result when A/D conversion is completed. It also includes flag that indicate when a conversion has been completed and when a conversion overrun has occurred.



* **T0TCR (Timer0 Timer Control Register):** It is an 8-bit read-write register. It is used to control the operation of the timer counter.



* **T0MR0 (Timer0 Match Register):** It is a 32-bit register. The values stored in this register are continuously compared with the Timer Counter value. When the two values are equal, the timer can be reset or stop or an interrupt may be generated.  The T0MCR controls what action should be taken on a match.
* **T0MCR (Timer0 Match Control Register):** It is a 16-bit register. It controls what action is to be taken on a match between the Match Registers and Timer Counter.
* **T0PR (Timer0 Prescale Register): It i**s a 32-bit register. It holds the maximum value of the Prescale Counter.

**Program:**

#include<lpc21xx.h>

#define RS 0x00020000 /\* RS - P1.17 \*/

#define RW 0X00040000 /\* R/W - P1.18 \*/

#define EN 0X00080000 /\* E - P1.19 \*/

#define CLR 0X00FE0000

unsigned int adc\_value=0;

int count;

void PORT\_Initial(void)

{

IO1DIR = 0x00FE0000; /\* LCD pins set as o/p \*/

IO0DIR = 0x00000000;

PINSEL0 = 0x00000000;

PINSEL1 = 0x05000000;

PINSEL2 = 0x00000000;

}

int Delay(unsigned int x)

{

x=x\*8000;

while(x>0)

{

x--;

}

return 0;

}

void LCD\_Command(char command)

{

int Temp;

IO1CLR = CLR; /\* Clearing the port pins \*/

IO1SET = EN; /\* Enable pin high \*/

IO1CLR = RS; /\* RS=0 for command register \*/

IO1CLR = RW; /\* R/W=0 for write \*/

Temp = (command & 0xF0) << 16; /\* Taking the first nibble of command \*/

IO1SET = IO1SET | Temp; /\* Writing it to data line \*/

Delay(2);

IO1CLR = EN; /\* Enable pin low to give H-L pulse \*/

}

void LCD\_Command1(char command1)

{

int Temp;

IO1CLR = CLR; /\* Clearing the port pins \*/

IO1SET = EN; /\* Enable pin high \*/

IO1CLR = RS; /\* RS=0 for command register \*/

IO1CLR = RW; /\* R/W=0 for write \*/

Temp = (command1 & 0xF0); /\* Taking the first nibble of command \*/

Temp = Temp << 16; /\* Shift it 16 bits to left \*/

IO1SET = IO1SET | Temp; /\* Writing it to data line(P1.20-P1.23) \*/

Delay(2);

IO1CLR = EN; /\* Enable pin low to give H-L pulse \*/

IO1CLR = CLR; /\* Clearing the port pins \*/

IO1SET = EN; /\* Enable pin high \*/

IO1CLR = RS; /\* RS=0 for command register \*/

IO1CLR = RW; /\* R/W=0 for write \*/

Temp = (command1 & 0x0F); /\* Taking the second nibble of command \*/

Temp = Temp << 20; /\* Shift it 20 bits to left \*/

IO1SET = IO1SET | Temp; /\* Writing it to data line \*/

Delay(2);

IO1CLR = EN; /\* Enable pin low to give H-L pulse \*/

}

void LCD\_Data(char data)

{

int Temp;

IO1CLR = CLR; /\* Clearing the port pins \*/

IO1SET = EN; /\* Enable pin high \*/

IO1SET = RS; /\* RS=1 for data register \*/

IO1CLR = RW; /\* R/W=0 for write \*/

Temp = (data & 0xF0); /\* Taking the first nibble of data \*/

Temp = Temp << 16; /\* Shift it 16 bits to left \*/

IO1SET = IO1SET | Temp; /\* Writing it to data line \*/

Delay(2);

IO1CLR = EN; /\* Enable pin low to give H-L pulse \*/

IO1CLR = CLR; /\* Clearing the port pins \*/

IO1SET = EN; /\* Enable pin high \*/

IO1SET = RS; /\* RS=1 for data register \*/

IO1CLR = RW; /\* R/W=0 for write \*/

Temp = (data & 0x0F); /\* Taking the second nibble of data \*/

Temp = Temp << 20; /\* Shift it 20 bits to left \*/

IO1SET = IO1SET | Temp; /\* Writing it to data line \*/

Delay(2);

IO1CLR = EN; /\* Enable pin low to give H-L pulse \*/

}

void LCD\_value( int count)

{

count=count+1;

LCD\_Data(count+0x30);

}

void LCD\_String( char \*dat)

{

while(\*dat!='\0') /\* Check for termination character \*/

{

LCD\_Data(\*dat); /\* Display the character on LCD \*/

dat++; /\* Increment the pointer \*/

}

}

void LCD\_Init(void)

{

Delay(15);

LCD\_Command(0x30);

Delay(10);

LCD\_Command(0x30);

Delay(5);

LCD\_Command(0x30);

Delay(5);

LCD\_Command(0x20);

Delay(5);

LCD\_Command1(0x28);

Delay(5);

LCD\_Command1(0x01); /\* Clear display \*/

Delay(5);

LCD\_Command1(0x06); /\* Auto increment \*/

Delay(5);

LCD\_Command1(0x0C); /\* Cursor off \*/

}

int ADC\_Conversion()

{

int ab; /\* Variable to store ADC value \*/

Delay(1);

ADCR = ADCR|0x01000000; /\* Start conversion \*/

while((ADDR&0x80000000)!=0x80000000); /\* Wait here till conversion is over\*/

ab = (ADDR&0x0000FFC0); /\* Extracting the result \*/

ab = (ab>>6); /\* Shift 6 bits right \*/

return ab; /\* Return the result \*/

}

void Int\_ASCII(int value,char cnt)

{

int i = 0; /\* Local variables \*/

char array[7];

int values;

values= value;

for(i=1;i<=cnt;i++) /\* Store the received value in array\*/

{

array[i] = values%10;

values = values/10;

}

for(i=cnt;i>=1;i--) /\* Display it on LCD\*/

{

LCD\_Data(array[i]+'0');

}

}

void Sensor\_Check()

{

ADCR=0x00200602; /\* PDN=1,CLKDIV=6,channel=AD0.2\*/

LCD\_Command1(0x80);

LCD\_String("Temp:");

adc\_value=ADC\_Conversion(); /\* Get the result of conversion\*/

// Delay(500000);

adc\_value=((adc\_value\*0.48828125));

LCD\_Command1(0x86); /\* 2nd row, 5th location \*/

Int\_ASCII(adc\_value,2); /\* Display the result on LCD \*/

}

\_\_irq void T0\_ISR (void)

{

int i;

for(i=0;i>=100;i++)

{

if((IO0PIN&0x00000100)==1)

{

count=count+1;

}

else

{

LCD\_Command1(0xC0);

LCD\_String(" ABnorm ");

}

}

LCD\_Command1(0xC0);

LCD\_value(count);

T0IR = ( T0IR | (0x01) );

VICVectAddr = 0x00;

}

int main()

{

PORT\_Initial(); /\* Initialize port \*/

LCD\_Init(); /\* Initialize LCD \*/

// LCD\_String("Temperature");

while(1)

{

Sensor\_Check(); /\* Take ADC reading \*/

Delay(50);

break;

}

LCD\_Command1(0xC0);

LCD\_String(" Heart Rate: ")

VPBDIV = 0x00000002; /\* For Pclk = 30MHz \*/

/\* We have configured Cclk=60MHz. Above instruction makes Pclk = Cclk/2 = 30MHz \*/ PINSEL0 = PINSEL0 | 0x00000020; /\* Configure P0.2 as Capture 0.0 \*/

IO0DIR = ( IO0DIR | (0x00000100) ); /\* 8 P0.8-P0.15 as output \*/

//IO0PIN = IO0PIN | 0x00000100; /\* Writing 1 to pin P0.8 \*/

VICVectAddr0 = (unsigned) T0\_ISR; /\* T0 ISR Address \*/

VICVectCntl0 = 0x00000024; /\* Enable T0 IRQ slot \*/

VICIntEnable = 0x00000010; /\* Enable T0 interrupt \*/

VICIntSelect = 0x00000000; /\* T0 configured as IRQ \*/

T0TCR = 0x02; /\* Reset TC and PR \*/

T0TCR = 0x00; /\* Timer mode, increment on every rising edge \*/

T0PR = 0x1D; /\* Load Pre-Scalar counter with 29 (0 to 29 = 30), so that timer counts every 1usec \*/

T0MR0 = 10000; /\* Load timer counter for 10\sec delay, 1usec\*1000\*100\*100\*/

T0MCR = 0x0003; /\* Interrupt generate on match and reset timer \*/

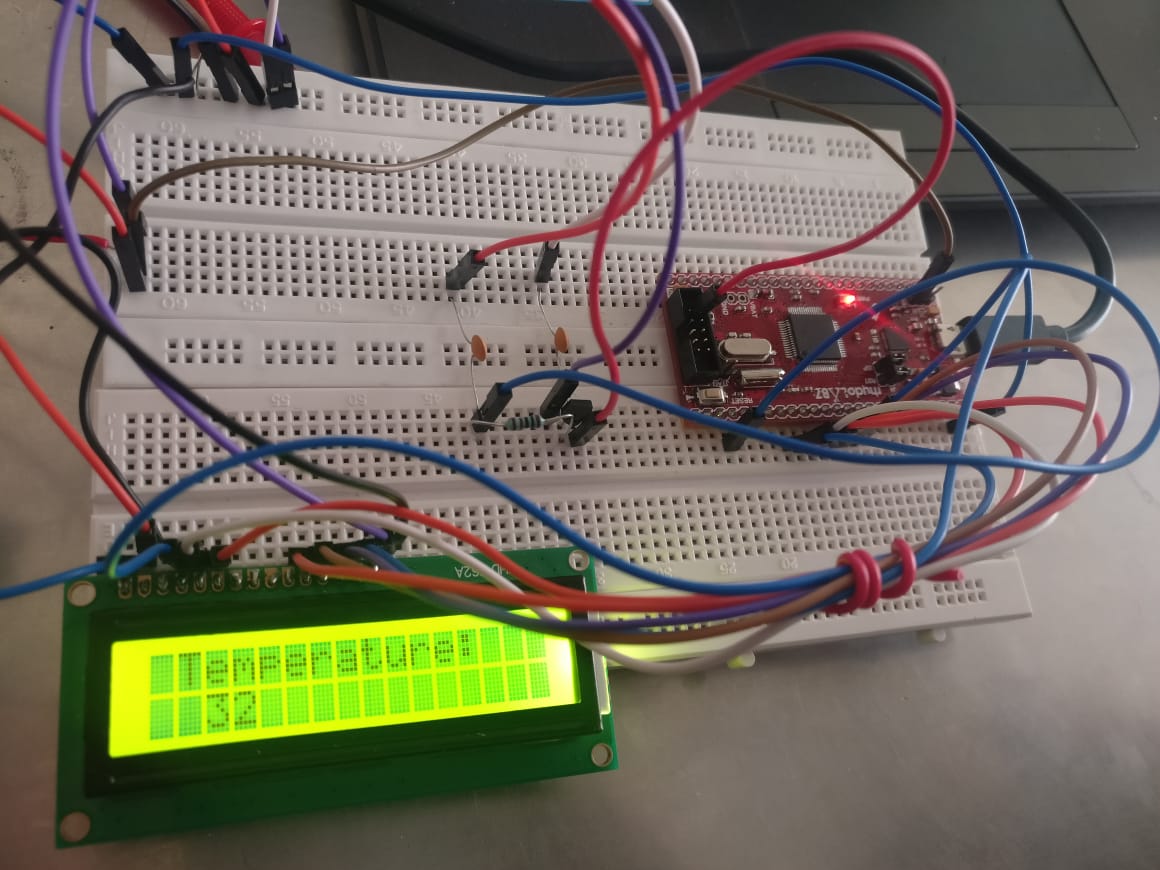
T0TCR = 0x01; /\* Enable timer \*/

while(1);

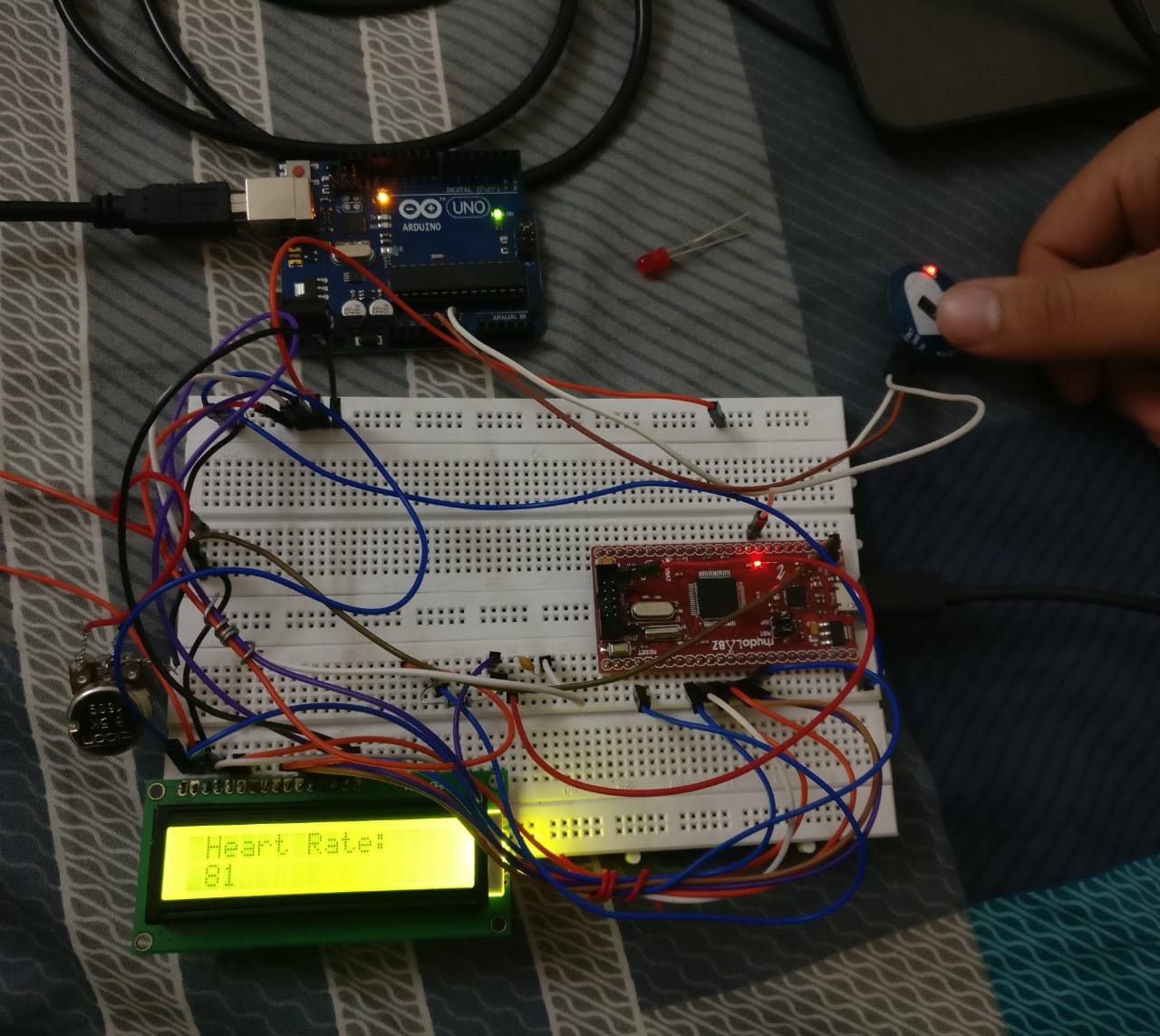
}

**Results :**

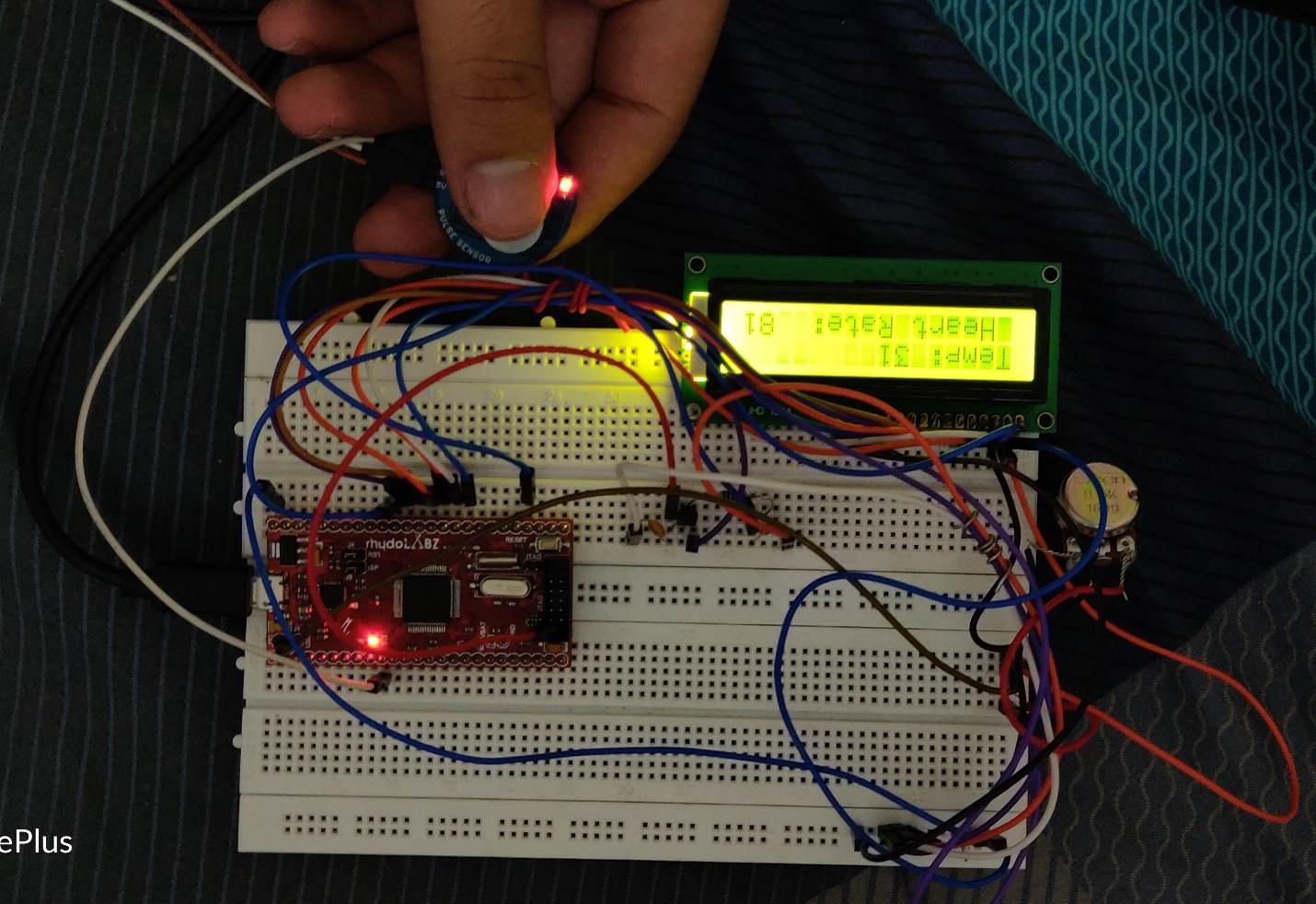
* First we have generated the code for Temperature sensing and interfaced temperature sensor with LPC2148 and displayed LCD to see the temperature sensed. The output of the same is as shown below.

****

* Then we have generated the code for Heart pulse rate sensing and interfaced hear rate sensor with LPC2148 and displayed on LCD to see the heart rate sensed. The output of the same is as shown below.

****

* We have merged both the codes of temperature sensing and heart pulse rate sensing and also interfaced both the sensors with LPC2148 and final output of the same is as shown below.



**Conclusion:**