

Programming

Programming the PIC16F877A microcontroller was done in the C programming language. This section will highlight the main programming methods used in order to make this system functional. (Note: The full code can be found in the Appendix.)

Setting up ports	
ADCON1 = 0b10001110;	//setup analogue input for heartbeat detection
TRISA = 0b00000011;	//setup porta as input
TRISB = 0b00000000;	//portb is set as output
TRISC = 0b00000000;	//portc and d is set as output

Table 12: Port set up code

Setting up the LCD display	
LCD_Config(&PORTB,4,5,6,3,2,1,0);	//sets up the LCD port
Lcd_Init(&PORTB);	// Initialize LCD connected to PORTB
Lcd_Cmd(LCD_CLEAR);	// Clears the display
Lcd_Cmd(LCD_CURSOR_OFF);	// Turns the LCD cursor off
Lcd_Cmd(LCD_SECOND_ROW);	
Lcd_Cmd(LCD_CLEAR);	//Clears the display
lcd_out(1,1,"Heart Rate Monitor");	//Displays the introduction to the LCD
delay_ms(2000);	//wait 2 seconds
Lcd_Cmd(LCD_CLEAR);	//Clears the display again

Table 13: LCD set up code

Measuring the heart rate and temperature
--

<pre> input1 = adc_read(0)/4; if(input1>=128) { led=1; buzzer=1; pulse=pulse+1; while(1) { if(sec_flag) { sec_flag=0; count=count+1; } if(count>15) break; input1=adc_read(0)/4; if(input1<=45) { led=0; buzzer=0; break; } } </pre>	<pre> //reads the input value as analogue and convert to digital //if the input value > 128 == 2.5Volts, identify as a high peak //turn LED on to indicate pulse //beep the buzzer to indicate pulse //add 1 to the pulse count //wait for the low peak from the signal //wait for the 1 second flag //keep counting seconds //if the counter reaches 15 seconds, then timeout //convert the input value (ADC) //if the input value is < 45 == 0.88 volts, identify as low peak //turn off the LED //turn off the buzzer //end from the loop </pre>
<pre> if(count>=15) { pulse=pulse*4; lcd_out(2,1,"BPM:"); inttostr(pulse,txt); </pre>	<pre> //if the count > 15 sec, execute the following //multiply pulse (15 sec) by 4 to get beats per minute //display pulse to the LCD </pre>
<pre> temp = adc_read(0x01)*2/4; </pre>	<pre> //read temperature value by converting ADC </pre>
<pre> lcd_out(1,1,"Temp:"); </pre>	<pre> //display temperature value on the LCD </pre>

Table 14: Heart rate and temperature sensing code

Bluetooth communication	
<pre> Usart_init(9600); </pre>	<pre> //initialize serial Bluetooth communication with a baud rate of 9600 bps //USART: Universal Synchronous Asynchronous Receiver </pre>

usart_write(pulse);	//sends the pulse rate data to Bluetooth
Usart_write(temp);	//send temperature data to Bluetooth

Table 15: Bluetooth code