Heart rate 3 click

From MikroElektonika Documentation

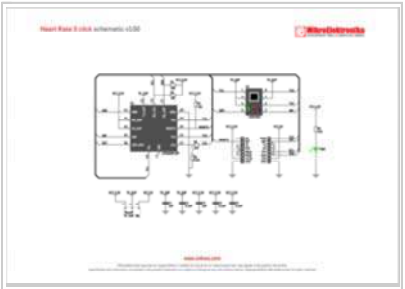
Heart Rate 3 click is a mikroBUS™ add-on board whose functionality is determined by two components: an OSRAM's SFH7050 pulse oximetry and heart rate monitoring module, and a TI AFE4404 (analong-front-end) IC specialized for bio-sensing.

Heart rate 3 click

Heart rate 3 click

IC/Module	AFE4404 (http://www.ti.com/lit/ds/symlink/afe4404.pdf) SFH7050 (http://www.osram-os.com/media/resource/HIRES/541656/246267/light-is-wearable---flysheet-biomon-sensor-sfh-7050-gb.pdf)
Interface	I2C
Power supply	3.3V
Website	www.mikroe.com/click/heart-rate-3 (http://www.mikroe.com/click/heart-rate-3)

Features and usage notes



Schematic also available in PDF (http://cdn-docs.mikroe.com/images/d/d1/Heart_Rate_3_click_s

Heart Rate 3 click is a mikroBUS™ add-on board whose functionality is determined by two components: an OSRAM's SFH7050 pulse oximetry and heart rate monitoring module, and a TI AFE4404 (analong-front-end) IC specialized for bio-sensing.

The SFH7050 multichip package contains 3 LEDs and one photodiode separated with a light barrier to prevent optical crosstalk. When the three LEDs shine through a subject's finger, some of the light is absorbed by the pulsating blood.

The analog reading from the SFH7050 is forwarded to the AFE chip that is able to derive pulse readings from the intensity of the reflected light.

AFE4404 is highly-configurable and adaptable for different usage scenarios (different lighting conditions or skin tones) making Heart Rate 3 click a robust heart rate monitoring solution.

The board communicates with the target MCU through the mikroBUS™ I2C interface, with additional functionality provided by RST, CLK and RDY pins.

Heart Rate 3 click works on a 3.3V power supply, but an onboard jumper allows you to set the voltage for driving the SFH7050 LEDs at either 3.3V or 5V.

Programming

Setting up of Heartrate 3 click and external interrupt to read values at 100hz and using an algorithm to find a heartrate.

```
#include <stdint.h>
#include "heartrate_3.h"
#include "resources.h"


// HeartRate 3 GPIO
sbit RST at GPIOC_ODR.B2;

void system_setup( void );
void setup_interrupt();

char  uart_text[20] = {0};
uint64_t int_count = 0;      //Used by timer to calibrate sampling freq.

void main()
{
     //Local Declarations
     uint16_t rate     = 0;
     char txt[15]      = {0};

     system_setup();     // GPIO / HeartRate 3 / UART / I2C Setups
     Delay_ms(200);
     initStatHRM();      // Initializes values to 0
     setup_interrupt();  // Setup interrupt handler

     while(1)
     {
         rate = hr3_get_heartrate();

         IntToStr( rate, uart_text );
         UART1_Write_Text( uart_text );
         UART1_Write_Text( "\r\n" );
     }

}

void system_setup( void )
{
       //Local Declarations
     char            text[40] = { 0 };
```

```
    dynamic_modes_t dynamic_modes;
    uint8_t         address = 0x58;

       //Set up dynamic modes for Heart Rate 3 Initialization
    dynamic_modes.transmit = trans_dis;                     //Transmitter disabled
    dynamic_modes.curr_range = led_double;                  //LED range 0 - 100
    dynamic_modes.adc_power = adc_on;                       //ADC on
    dynamic_modes.clk_mode = osc_mode;                      //Use internal Oscillator
    dynamic_modes.tia_power = tia_off;                      //TIA off
    dynamic_modes.rest_of_adc = rest_of_adc_off;            //Rest of ADC off
    dynamic_modes.afe_rx_mode = afe_rx_normal;              //Normal Receiving on AFE
    dynamic_modes.afe_mode = afe_normal;                    //Normal AFE functionality

      //GPIO setup
    GPIO_Digital_Output( &GPIOC_BASE, _GPIO_PINMASK_2 );
    GPIO_Digital_Input( &GPIOA_BASE, _GPIO_PINMASK_0 );
    GPIO_Digital_Input( &GPIOD_BASE, _GPIO_PINMASK_10 );

      //UART Initialize
    UART1_Init( 9600 );
    UART1_Write_Text( "UART is Initialized\r\n" );

      //Toggle Reset pin
    RST = 0;
    Delay_us(50);
    RST = 1;

      //I2C Initialize
    I2C1_Init_Advanced( 400000, &_GPIO_MODULE_I2C1_PB67 );
    UART1_Write_Text( "I2C Initialized\r\n" );

      //Heart Rate 3 Initialize
    hr3_init( address, &dynamic_modes );


}

void setup_interrupt()
{
  GPIO_Digital_Output(&GPIOE_BASE, _GPIO_PINMASK_HIGH);  // Enable digital output on PORTD
  GPIOE_ODR = 0xAAAA;
  GPIO_Digital_Input(&GPIOD_BASE, _GPIO_PINMASK_10);

  RCC_APB2ENR.AFIOEN = 1;              // Enable clock for alternate pin functions
  AFIO_EXTICR3 = 0x0300;               // PD10 as External interrupt
  EXTI_RTSR = 0x00000400;              // Set interrupt on Rising edge
  EXTI_IMR |= 0x00000400;              // Set mask
  NVIC_IntEnable(IVT_INT_EXTI15_10);   // Enable External interrupt
  EnableInterrupts();                  // Enables the processor interrupt.
}

void ExtInt() iv IVT_INT_EXTI15_10 ics ICS_AUTO {
   EXTI_PR.B10 = 1;                            // clear flag
   int_count++;
   statHRMAlgo( hr3_get_led1_amb1_val() );      // Give led1 ambient value to heartrate function. ( 100 times a second )

}
```

Code examples that demonstrate the usage of Heart rate 3 click with MikroElektronika hardware, written for mikroC for ARM, AVR, dsPIC, FT90x, PIC and PIC32 are available on Libstock (http://libstock.mikroe.com/projects/view/1908/heart-rate-3-click).

Resources

- Learn article explaining Heart rate 3 click library (http://learn.mikroe.com/microcontrollers-have-a-heart-too/)

- Libstock Heart rate 3 click example (http://libstock.mikroe.com/projects/view/1908/heart-rate-3-click)

Retrieved from "http://docs.mikroe.com/index.php?title=Heart_rate_3_click&oldid=764"
