Damian Cisneros

CPE301 – SPRING 2018

Final Project Heartrate Sensor (AD8232)

**DO NOT REMOVE THIS PAGE DURING SUBMISSION:**

The student understands that all required components should be submitted in complete for grading of this assignment.

|  |  |  |  |
| --- | --- | --- | --- |
| **NO** | **SUBMISSION ITEM** | **COMPLETED (Y/N)** | **MARKS**  **(/MAX)** |
| 1 | COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS |  |  |
| 2. | INITIAL CODE OF TASK 1/A |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 2/B |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 3/C |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 4/D |  |  |
| 3. | INCREMENTAL / DIFFERENTIAL CODE OF TASK 5/E |  |  |
| 4. | SCHEMATICS |  |  |
| 5. | SCREENSHOTS OF EACH TASK OUTPUT |  |  |
| 5. | SCREENSHOT OF EACH DEMO |  |  |
| 6. | VIDEO LINKS OF EACH DEMO |  |  |
| 7. | GOOGLECODE LINK OF THE DA |  |  |
|  |  |  |  |
|  |  |  |  |

Damian Cisneros

1. **COMPONENTS LIST AND CONNECTION BLOCK DIAGRAM w/ PINS**

List of Components used:

ATmega328p

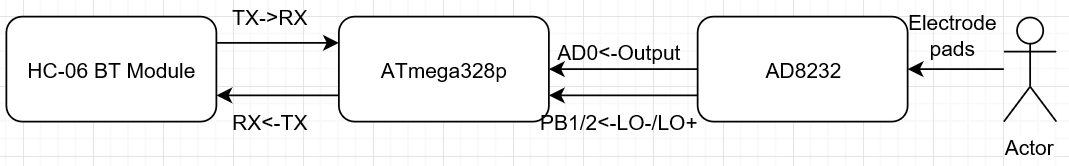
AD8232 with electrode pads

Breadboard

10kΩ resistor

Nodemcu(only to power 3.3V)

HC-06 (UART sending)



Block diagram for heartrate sensor circuit

1. **INITIAL/DEVELOPED CODE OF HEARTRATE SENSOR**

/\*

\* HeartRateSensor.c

\*

\* Created: 5/3/2018 11:50:08 AM

\* Author : Damian

\* Descrition : This program collects and outputs data from the AD8232 to UART

\* terminal. I used a bluetooth module to send data

\*/

#define BAUD 9600

#define F\_CPU 8000000UL

#include <stdlib.h>

#include <avr/io.h>

#include <avr/interrupt.h>

#include <stdio.h>

#include <util/delay.h>

void init\_USART**();**

void init\_ADC**();**

void USART\_tx\_string**(**char**\*);**

static int put\_char**(**char c**,** FILE **\***stream**);** //required for printf

static FILE mystdout **=** FDEV\_SETUP\_STREAM**(**put\_char**,** **NULL,** \_FDEV\_SETUP\_WRITE**);** //required for printf

int main**(**void**)**

**{**

DDRB **=** 0**;** //input for LO- and LO+ from AD8232

char c**[**9**];** //holds converted value in string

stdout **=** **&**mystdout**;** //set the output stream

init\_USART**();**

init\_ADC**();**

while(1){

uint8\_t low; //ADCL

uint16\_t all = 0; //ADC 10 bit

uint16\_t avg = 0; //average buffer

for(int i=0;i < 100 ; i++){ //grab 100 data points and sum then average.

ADCSRA |= (1 << ADSC); //start the conversion. while in free running mode it will

while((ADCSRA&(1 << ADIF))==0); //check if conversion done

ADCSRA |= (1 << ADIF); //reset flag

low = ADCL; //grab lower 8 adc bits

all = ADCH << 8 | low; //grab all 10 bits of ADC

avg += all; //add to buffer

\_delay\_us(1000);

}

all = avg/10; //average and scale down so its easier to see

dtostrf(all,3,1,c); //convert double to string

if(PINB & 0b00000011){ //check if all pads connected

printf("0"); //pads not connected so output 0

}

else

USART\_tx\_string(c); //print value to terminal

printf("\r\n");

}

}

void init\_USART(){

unsigned int BAUDrate;

//set BAUD rate: UBRR = [F\_CPU/(16\*BAUD)]-1

BAUDrate = ((F\_CPU/16)/BAUD) - 1;

UBRR0H = (unsigned char) (BAUDrate >> 8); //shift top 8 bits into UBRR0H

UBRR0L = (unsigned char) BAUDrate; //shift rest of 8 bits into UBRR0L

UCSR0B |= (1 << RXEN0) | (1 << TXEN0); //enable receiver and trasmitter

UCSR0C |= (1 << UCSZ01) | (1 << UCSZ00); //set data frame: 8 bit, 1 stop

}

void init\_ADC(){

ADMUX = 0; //use ADC0

ADMUX |= (1 << REFS0); //use AVcc as the reference (5V)

ADMUX |= (1 << ADLAR); //set to right adjust for 8-bit ADC

//ADCSRA |= (1 << ADIE); //ADC interrupt enable

ADCSRA |= (1 << ADEN); //enable ADC

//set pre-scale to 2 for input frequency

ADCSRA &= ~(1 << ADPS2) | ~(1 << ADPS1) | ~(1 << ADPS0);

ADCSRB = 0; //free running mode

}

static int put\_char(char c, FILE \*stream) //required for printf

{

while(!(UCSR0A &(1<<UDRE0))); // wait for UDR to be clear

UDR0 = c; //send the character

return 0;

}

void USART\_tx\_string(char\* data){

while((\*data!='\0')){ //print until null

while(!(UCSR0A &(1<<UDRE0))); //check if transmit buffer is ready for new data

UDR0=\*data; //print char at current pointer

data++; //iterate char pointer

}

}

1. **SCHEMATICS**

Heartrate Circuit Schematic

1. **VIDEO LINKS OF EACH DEMO**

playlist - <https://www.youtube.com/playlist?list=PL5RuXbzEXwesoMc_Fql0MObBBz-9phIft>

Heartrate circuit functioning - <https://youtu.be/g1nvmksnAwo>

Heartrate before filtering - <https://youtu.be/ZfCkhiwIAkg>

Heartrate after filtering - <https://youtu.be/UEHIkM5jgPI>

1. **GITHUB LINK OF THIS DA**

**Student Academic Misconduct Policy**

<http://studentconduct.unlv.edu/misconduct/policy.html>

“This assignment submission is my own, original work”.