NYU TANDON SCHOOL OF ENGINEERING Name: _____

ECE4144 Intro to Embedded Systems ID: _____

Exam 2, Spring 2024

Real ECG Implementation

During our most recent recitation we worked with an ECG chip from Analog Devices (SparkFun Single Lead Heart Rate Monitor - AD8232 - SEN-12650 - SparkFun Electronics). Because we were focused on implementing a simple moving average filter, we somewhat "brute forced" the implementation. The purpose of this exam is to implement a more refined and efficient version of a heart rate ECG monitor, using interrupts, GPIO and serial communications.

Part 1-Setup the Moving average filter function….

We want to have a configurable N point moving average filter where N is a number from 1 to 9. Set up a global variable "N" to hold the configurable average count. Default N to 1. Write a function called "calculateNpointMovingAVG" that takes in one parameter, an array holding the stored values to average. The function should return an "int" representing the N point moving average.

Part 2 – Set up the sampling using timer interrupts….

Pick a timer on your device and configure it such that it fires one of the timer Interrupts (of your choosing) every 5ms. Write the interrupt service routine that runs when your interrupt occurs. You can use register names like TIMSK, OCR etc. Be sure to configure the timer properly using TCCRA, B and OCRA, B etc. Leave the interrupt service routine empty for now.

Part 3-Setup the ADC to read the output of the ECG sensor….

In the setup, write the code to setup the ADC. Choose an available Analog Input pin on your controller to connect to the analog output of the ECG sensor. Use a clock prescaler of 4 and setup the ADC, including ADMUX, ADCSRA etc. such that the ADC is ready to make a measurement, but don't start a conversion. Use a Right Justified result (ADLAR=0), and Vcc as a reference. Setup the Auto Trigger source to be the timer interrupt that you setup in Part 2. If you do Part 2 correctly, the timer event you setup will trigger the ADC to make a conversion every 5ms.

Now, also setup the ADC interrupt which runs automatically when the ADC conversion is complete. Implement the service routine such that when the ADC interrupt occurs, the ADC value is read and the data is stored in an array. In addition, you should call the "calculateNpointMovingAVG" function to get the moving average value which is outputted over serial using:

Serial.print(">HR:");

Serial.println(HRmovingAvg);

Part 4-Make the N configurable….

Using the built in Serial library in Arduino (Serial.read, Serial.available() etc.) , allow the developer to change N by typing a single digit in the serial terminal window to dynamically change N. So if the user types in the terminal "5", N now becomes 5 dynamically, in run-timea.

Part 5 (EXTRA CREDIT)-Use the Playground Classic resources to indicate the moving average value….

Write a function called “UpdateUI” which takes in a single number (which will be the moving average). The function should do “something” on the board that indicates the current value. Be creative…You can use the led, speaker, NeoPixels or anything else on the board to provide some real time feedback regarding the current ECG moving average. Call the UpdateUI function inside the calculateNPointMovingAVG function. You can use the Adafruit library we used in class if you wish (or simple GPIO etc.)