Day 9: Design 2

Wednesday, July 26, 2023 10:07 AM

Goals:

- Work on the Design (pseudo code for App software and define variables needed for project)
- Try to work on making the SD card module function
- Begin testing some integration between Arduino and App.
- Do the team assessment
- Figure out needed items and if we do a PCB. (Due at 1pm!!)

Accomplishments:

- SD card module is now working, but finicky
- Requirements signed off!
- Improved pseudo code detail for Arduino DAQ

Arduino DAQ:

External Libraries:

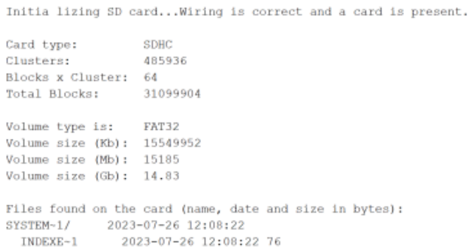
```
    SPI
    SD
    PulseSensorPlayground
    uRTCLib
    ArduinoBLE
    UnixTime
```

Global Variables and Constants:

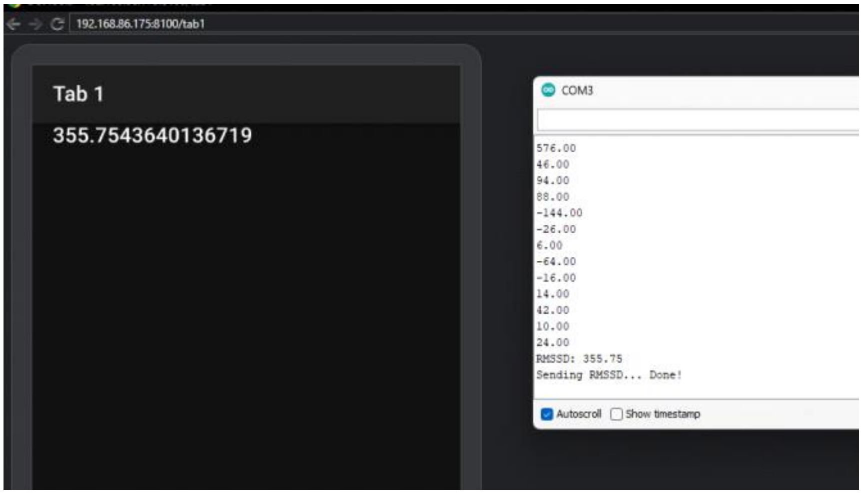
```
    Bluetooth:
        hrvService: ArduinoBLE::BLEService
            Bluetooth Low Energy HRV Metric Service
        hrvChar: ArduinoBLE::BLECharacteristic
            Bluetooth Low Energy Characteristic to send HRV records
        errorChar: ArduinoBLE::BLEByteCharacteristic
            Bluetooth Low Energy Characteristic to send error codes to application
        requestChar: ArduinoBLE::BLECharacteristic
            Bluetooth Low Energy Characteristic to receive data requests from application
        transferInProgress: bool
            Whether a transfer is currently in progress
        transferFile: SD::File
            Current file used in transfer
        transferFilename: string
            Current file name (see reqs)
        transferFilePosition: uint32_t
            Current read position from the file
        connectedDevice: ArduinoBLE::BLEDevice
            Current connected device

    SD:
        CHIP_SELECT: const int
            Digital pin for the SD card's chip select

    RTC:
        rtc: uRTCLib::uRTCLib
            uRTCLib library object

    Pulse Sensor:
        PULSE_INPUT: const int
            Analog pin for pulse sensor
        THRESHOLD: const int
            Threshold for pulse sensor signal for the PulseSensorPlayground library
        pulseSensor: PulseSensorPlayground::PulseSensorPlayground
            PulseSensorPlayground library object

    HRV calculation:
        rmssd: float
            The calculated RMSSD in ms (the HRV metric)
        numRRDetected: int
            Counter for the number of RR intervals found in the 5 minute measurement period
        hrvStartTime: unsigned long
            Millisecond where the HRV measurement began
        lastPeakTime: unsigned long
            Millisecond where the last peak was found
        lastRRInterval: unsigned long
            Duration of the last RR interval
        rrDiffSquaredTotal: float
            Intermediate value for calculating RMSSD (the numerator under radical).
        heartbeatsTilMeasure: int
            Number of heartbeats to detect before measurement (initial value is 5).
```

Arduino Functions:

```
setup: () => void

    Start BLE library

    If BLE fails to start:
        Hang execution

    Set the BLE local name to Tranquil+

    Set the BLE advertised service to hrvService

    Add hrvChar, errorChar, requestChar to hrvService

    Add hrvService to BLE

    Write null values to each BLECharacteristic.

    Advertise the BLE device

    Start SD library using CHIP_SELECT as the chip select pin

    If SD card fails to initialize:
        Hang execution

    Begin RTC module

    Set up all the pulseSensor information (PULSE_INPUT and THRESHOLD)

    Begin pulseSensor
```



```
Initia lizing SD card...Wiring is correct and a card is present.

Card type:         SDHC
Clusters:          485936
Blocks x Cluster:  64
Total Blocks:      31099904

Volume type is:    FAT32
Volume size (Kb):  15549952
Volume size (Mb):  15185
Volume size (Gb):  14.83

Files found on the card (name, date and size in bytes):
SYSTEM~1/     2023-07-26 12:08:22
  INDEXE~1      2023-07-26 12:08:22 76
```



192.168.86.175:8100/tab1

Tab 1

355.7543640136719

COM3

```
576.00
46.00
94.00
88.00
-144.00
-26.00
6.00
-64.00
-16.00
14.00
42.00
10.00
24.00
RMSSD: 355.75
Sending RMSSD... Done!
```

```
    If pulseSensor fails to start
        Hang execution

    Initialize all HRV variables

loop: () => void:

    If requestChar is written to from another device and not transferInProgress:
        Set transferInProgress to true
        Set transferFilename and transferFilePosition according to the requestChar value

    If transferInProgress:
        sendRecords

    If hrvStartTime is set and 5 minutes have passed since hrvStartTime:
        Let beatsPerMinute: float be (numRRDetected+1)/ 5 mins

        If  40 bpm < beatsPerMinute < 240 bpm
            Set rmssd to be sqrt(rrDiffSquaredTotal/(numRRDetected+1))
            storeRecord
        Else:
            Write 1 to errorChar

        resetHrv

    If  pulseSenor detects a new beat:
        If hrvStart is not set:
            Decrement heartbeatsTilMeasure
            If heartbeatsTilMeasure <= 0:
                Set hrvStartTime to current millisecond
                updateHrv
        Else:
            updateHrv


Defined Functions:
storeRecord: () => int

    If transferInProgress:
        Close transferFile

    Let unix: unsigned long be the current unix millisecond (obtained from RTC)
    Let date: string be the current ISO 8601 UTC date
    Let datafile: string be "HRV-{date}.txt" (according to requirement doc)

    Open datafile from SD card for writing

    Write a record using unix and rmssd (according to requirement doc)

    Close datafile

    If transferInProgress:
        Open transferFilename from SD card for reading and store the file object in transferFile


sendRecords: () => int
    If transferFile is not set:
        Let filename: string be "HRV-{transferDate}.txt"

        If filename exist:
            Open "HRV-{transferDate}.txt" from SD card for reading and store the file object in transferFile
        else:
            Write 2 to errorChar
            Return 1


    Let data:string be the next line of data from transferFile (starting from transferFilePosition and ending at the
    position of the next newline character)

    Write data to hrvChar

    Set transferFilePosition to the start of the next line (next newline character + 1)

    If transferFilePosition is greater than the size of the file:
        Close transferFile
        Set transferDate to next day
        Let filename: string be "HRV-{transferDate}.txt"

        If filename exist:
            Open "HRV-{transferDate}.txt" from SD card for reading and store the file object in transferFile
            Set transferFilePosition to 0
        else:
            Write current file and position to transferChar
            Set transferInProgress to false

    Return 0


updateHrv: () => void
    Let currentPeakTime: unsigned long be current millisecond

    If lastPeakTime is not set:
        Set lastPeakTime to currentPeakTime

    Else
        Let currentRRInterval: unsigned long be currentPeakTime - lastPeakTime
        Increment numRRDetected

        if lastRRInterval is not set:
            Set lastRRInterval to currentRRInterval

        Else:
            Let rrDiff: float be currentRRInterval - LastRRInterval
            Add rrDiff^2 to rrDiffSquaredTotal


resetHrv: () => void
    Reset lastRRInterval, lastPeakTime, rmssd, rrDiffSquaredTotal, numRRDetected, hrvStartTime,
    heartbeatsTilMeasure to initial value
```

RTC initialization:

TODO (one time run before hand)

App Code:
Framework: Ionic/React
Using "Tabs" template as a baseline

(TODO)

Day 10: Design 3

Sunday, July 30, 2023 3:21 PM

Goals:

- Continue working on software design
  - Pseudocode for app and RTC init still need to be written
  - Revising DAQ pseudocode based on preliminary testing
- Begin developing some of the software for the DAQ and App
- Aid with other parts of the design, time permitting