Heart Monitor
Final Demo

By: Habiba Gamal (900151007)

GITHUB REPO:

https://github.com/habibagamal/EmbeddedProject_HeartMonitor

Achieved Progress in Embedded Application

- Fully developed Python application with receipt and transmission of commands, as well as, plotting of ECG signals
- 3 supported commands in the embedded application: collect 1 minute worth of data, report heart rate and set the sampling frequency

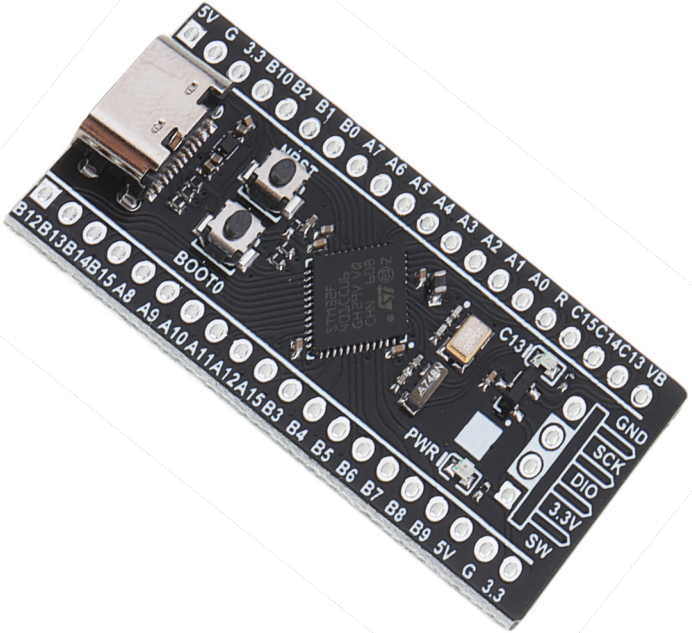
To-do

- In 1 minute, calculate several values for the bpm and take average.

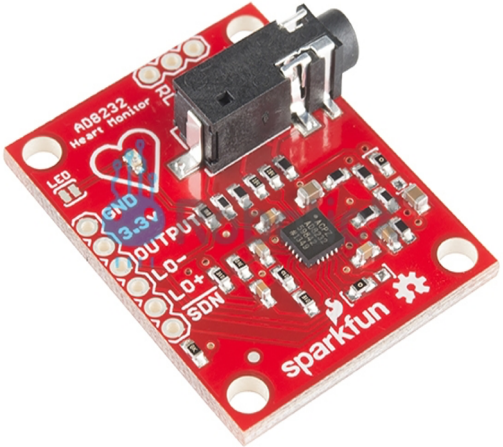
Progress from Demo 1

- Calculating bpm
- Concatenating white spaces to short commands to fill the UART buffer
- Bug fixes

Used Components

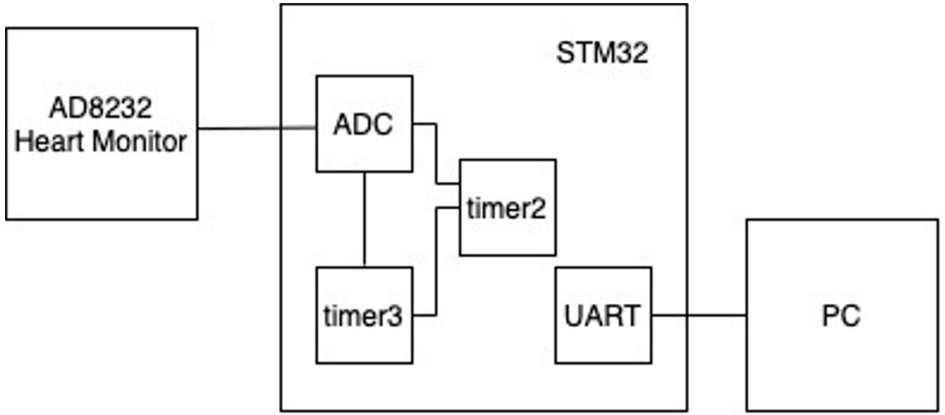


USB
TTL
232
485
Converter



Block Diagram

- When developing the application, I realized I need 2 timers. Timer 2 is used to count 1 minute, timer 3 is used to trigger ADC conversion.



Design Details of Embedded Application

- ADC is started as application starts
- UART receive generates interrupt.
- If one of the three supported commands is received, its operation starts
- If “start” command is sent, timer2 and timer3 are started. At the end of each ADC conversion, the ADC converted value sent over UART.
- The python application uses this value to plot the real-time graph

UART baud rate

- I increased the baud rate to 115200
- 1 kHz is practically the maximum sampling rate of the application
- To transmit 1k samples per second, each sample is made up of 4 digits and each digit is transmitted in 10 bits.
- In 1 second, 40000 bits need to be transmitted

Design Details of Embedded Application (Cont’d)

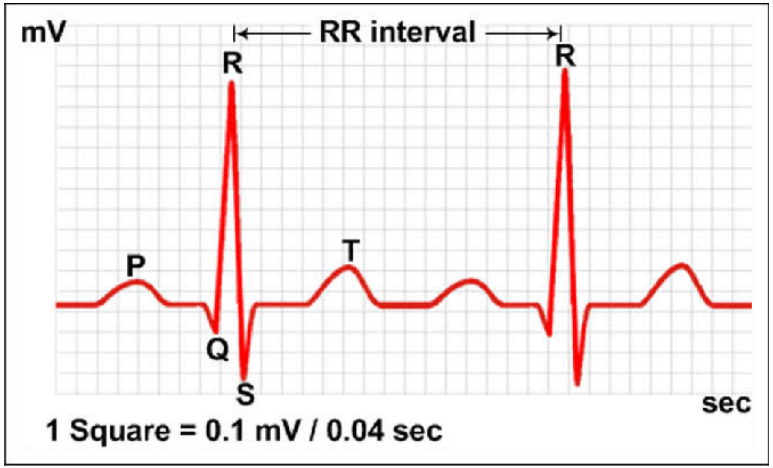
- If “f=xxxx” is sent, the program reads “xxxx”, converts it to integer, then computes the period that corresponds to this frequency.
- The period of timer3 (which triggers the start of conversion of the ADC) is changed to match the computed period through changing the autoreload register.
- The next “start” command will use this new frequency

Design Details of Embedded Application (Cont’d)

- If “bpm” command is sent, bpm is calculated and reported to the user
- If none of the 3 commands is received, the UART transmits “no command” to indicate that it received unsupported command

How is BPM calculated?

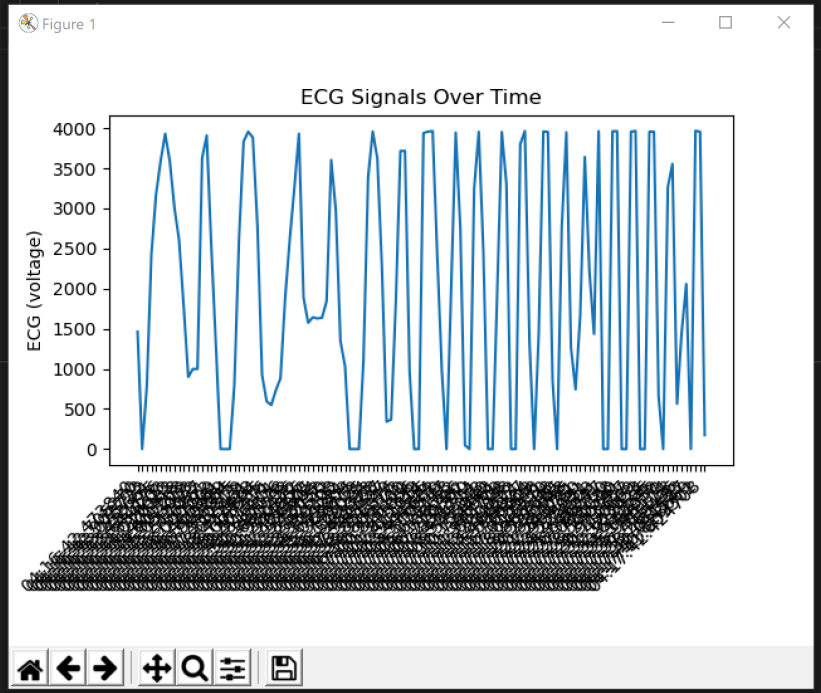
- When collecting 1 minute worth of data, the ECG signal peaks are detected
- The time for 2 consecutive peaks is recorded
- RR interval: time between 2 consecutive peaks
- Min RR interval = 0.6s, max RR interval = 1.2s
- Heartrate = 60 / RR interval (in seconds)



term.py

Home > Desktop > Spring2020 > EmbeddedSystems > Project > Project > KEIL_CubeMX_Projects > MDK-ARM > term.py >

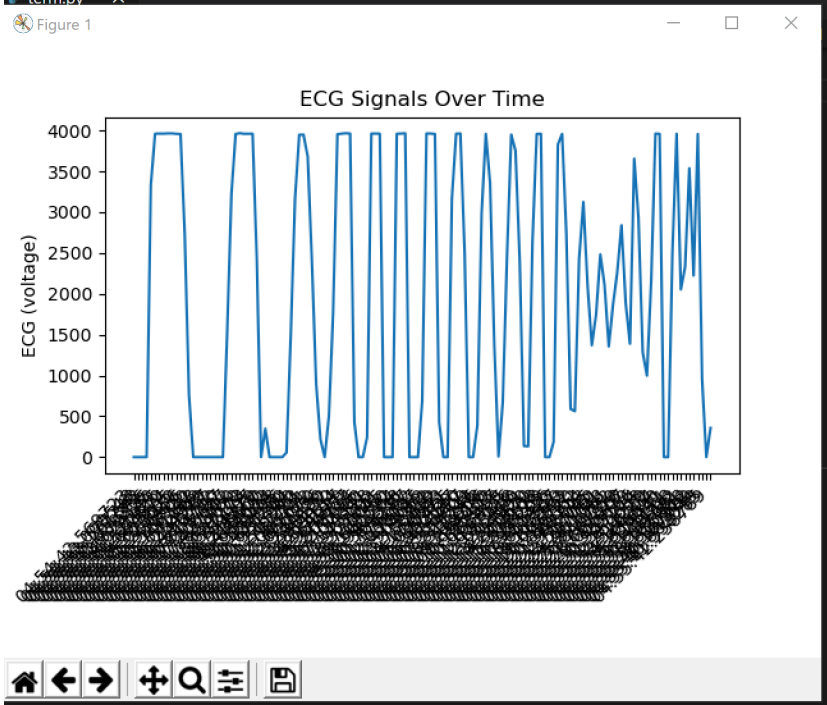
```
while (1):
    try:
```



PROBLEMS

```
3957
3956
656
0
3266
3555
565
1455
2058
0
3968
3956
173
bpm
bpm=66.964287
```

term.py




```
2224
3958
963
0
357
bpm
bpm=68.571426
```

Design Details of Python Application

- Multi-threaded application
- Thread 1
  - Reads command line input
  - Send this data serially to the microcontroller
- Thread 2
  - Reads data from the microcontroller
  - Prints received data to the user
  - Plots the real-time ECG signals