

Embedded Systems Project

Heart Monitor

Spring 2020

Eslam Adel Soliman

900163257

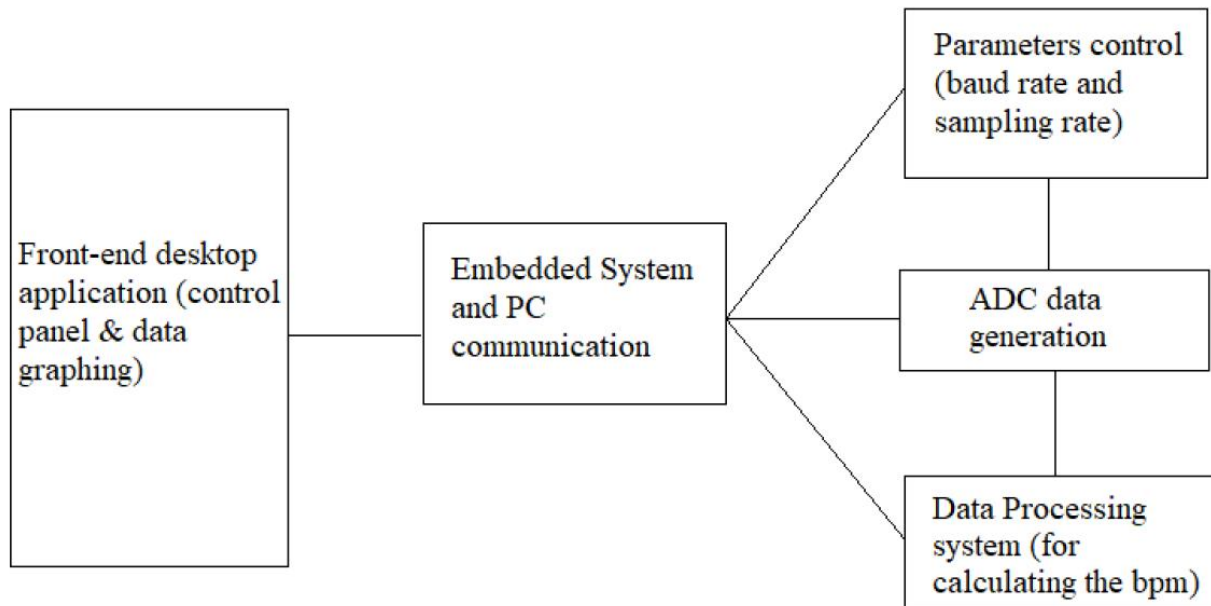
Project Description:

Real-time ECG plotter for AD8232 sensor, using the STM32F103C8T6 microcontroller (black pill)

Project Features:

- The ability to collect and plot 1 min worth of ECG data
- The ability to set the sampling rate from 4 values (25, 50, 100, 200) Hz, with the possibility of adding new values
- The ability to ask for calculating and displaying the BPM (beats per minute)

Project Architecture:



The project consists of two main components:

- C++ driver for the MCU and the ECG sensor, its main functionalities:
 - Analogs-to-Digital conversion of the sensor output
 - Transmitting Digital Data to PC using serial communication (UART)
 - Setting sampling rate for the ADC conversion

- Python GUI PC application for plotting transmitted data and convenient user interaction, main application functionalities:
 - Receiving and plotting data from the MCU through serial communication
 - Transmitting two kinds of commands to the MCU, request to send data, and setting the sampling rate
 - Moderating commands issuing synchronization

Communication Encoding:

Communication is based on UART with a baud rate of 115200bit/s

MCU-to-PC:

- Data are sent to PC via UART as 2-byte unsigned integers, in the range [0, 4,095], this range represent voltage output in the range [0, 3.3] volt.

PC-to-MCU:

- Commands are sent to the MCU as an 1-byte ASCII characters

‘0’: Request to send 1-minute worth of ECG data

‘1’: Set the sampling rate to 25Hz

‘2’: Set the sampling rate to 50Hz

‘4’: Set the sampling rate to 100Hz

‘8’: Set the sampling rate to 200Hz

Decisions / Assumptions:

- ECG plot shows last 500 ECG readings, the x-axis is not time based.
- As a result of code latency, plot refreshing/updating rate is less than sampling rate, as a result, to display 1-minute worth of data in high sampling rates ≥ 50 , the plot will take more than one minute. Plotting time increases linearly with the chosen sampling rate
- Beats per minutes (BPM) value is calculated using the most recent 1-minute set of data, to be able to calculate and get the bpm, plotting data request has to be issued first.
- All buttons are disabled while receiving ECG data, the user is not able to request new group of data, change the sampling rate, neither request calculating bpm.
- Beats per minute threshold (BPM) is 2300 in digital output, or 1.85V in analog. This limit is chosen based on several runs with testing and observation, while comparing with expected results.
- Available sampling rates are (25, 50, 100, 200). No higher sampling rates are supported as higher frequencies are not popular for regular use ECGs. Also, higher sampling rates will lead to an inconvenience in waiting for ≥ 10 minutes to collect results

MCU Driver Implementation Details (Program Flow):

- In case MCU is transmitting data, timer interrupt gets fired regularly, parametrized via the sampling rate chosen by the user.
- Timer interrupts triggers ADC start of conversion.
- ADC conversion complete callback function transmits conversion output (non-blocking) to PC via UART.
- MCU continuously listen for PC commands, commands sent from PC are either requests for sending data or requests for changing sampling rate

Desktop App Implementation Details:

- Application consists of two main threads:
 - Main thread for handling GUI events; user interaction and screen refreshing, checks for data updates, and updates the plot.
 - Listening thread that receives data from the MCU and pushes them to updates queue.
- Plot refresh rate depends on run-time performance; data are being displayed in the plot as-soon-as it is possible. As a result, 1 min worth of data could take more than 1 minute to be plotted in the case of high sampling rates
- BPM is calculated on the desktop app side via checking if signal passes certain threshold, in out case it is 1.85V.

System configuration (MCU):

- System clock is set to 8MHz
- Timer clock is set to 8MHz
- Timer prescaler is set to 799
- ADC clock is set to 4MHz
- UART baud rate = 115200bits/s

CubeMX timer configurations:

The screenshot displays the STM32CubeMX interface for configuring the STM32F103C8Tx microcontroller. The 'Pinout & Configuration' tab is active, showing the 'TIM2 Mode and Configuration' settings.

TIM2 Mode and Configuration:

- Mode: **Disable**
- Trigger Source: **Disable**
- Clock Source: **Internal Clock**
- Channel1: **Disable**
- Channel2: **Disable**
- Channel3: **Disable**
- Channel4: **Disable**
- Combined Channels: **Disable**
- ☐ Use ETR as Clearing Source
- ☐ XOR activation
- ☐ One Pulse Mode

Configuration:

- Reset Configuration
- Parameter Settings (selected)
- User Constants
- NVIC Settings
- DMA Settings

Configure the below parameters:

- Counter Settings
 - Prescaler (PSC - 16 bits value): **799**
 - Counter Mode: **Up**
 - Counter Period (AutoReload Register - 16 bits val...): **0**
 - Internal Clock Division (CKD): **No Division**
 - auto-reload preload: **Disable**
- Trigger Output (TRGO) Parameters
 - Master/Slave Mode (MSM bit): **Disable (Trigger input effect not delayed)**
 - Trigger Event Selection: **Reset (UG bit from TIMx_EGR)**

Pinout view:

The pinout view shows the STM32F103C8Tx LQFP48 package. The pins are labeled as follows:

- VBAT
- PC13
- PC14
- PC15
- PD0
- PD1
- NRST
- VSSA
- VDDA
- PA0
- PA1
- PA2
- PA3
- PA4
- PA5
- PA6
- PA7
- PB0
- PB1
- PB2
- PB10
- PB11
- VSS
- VDD
- PA12
- PA11
- PA10
- PA9
- PA8
- PB15
- PB14
- PB13
- PB12

Additional labels on the right side of the pinout view include:

- USART1_RX
- USART1_TX

Limitations:

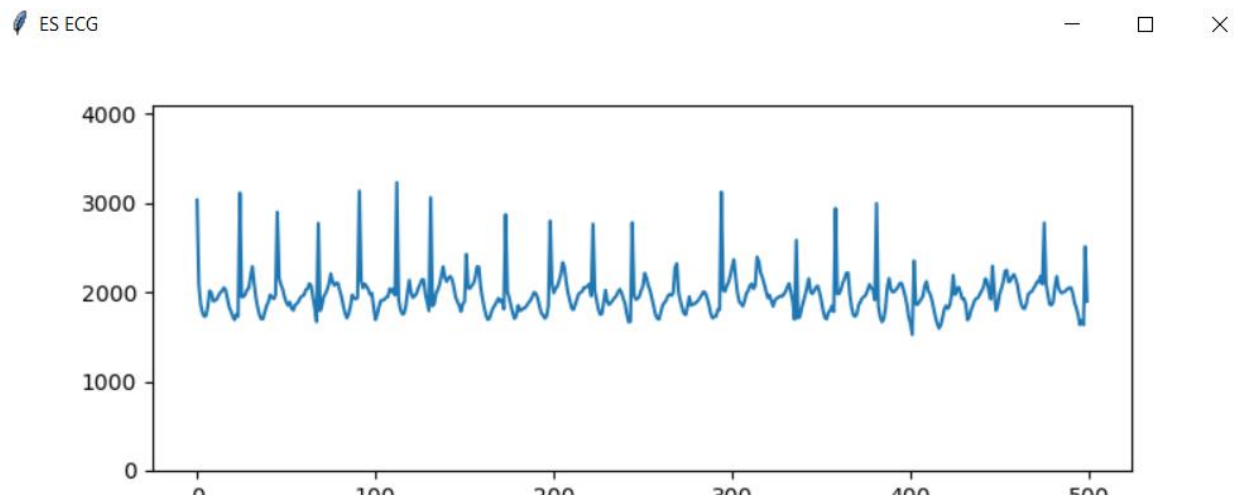
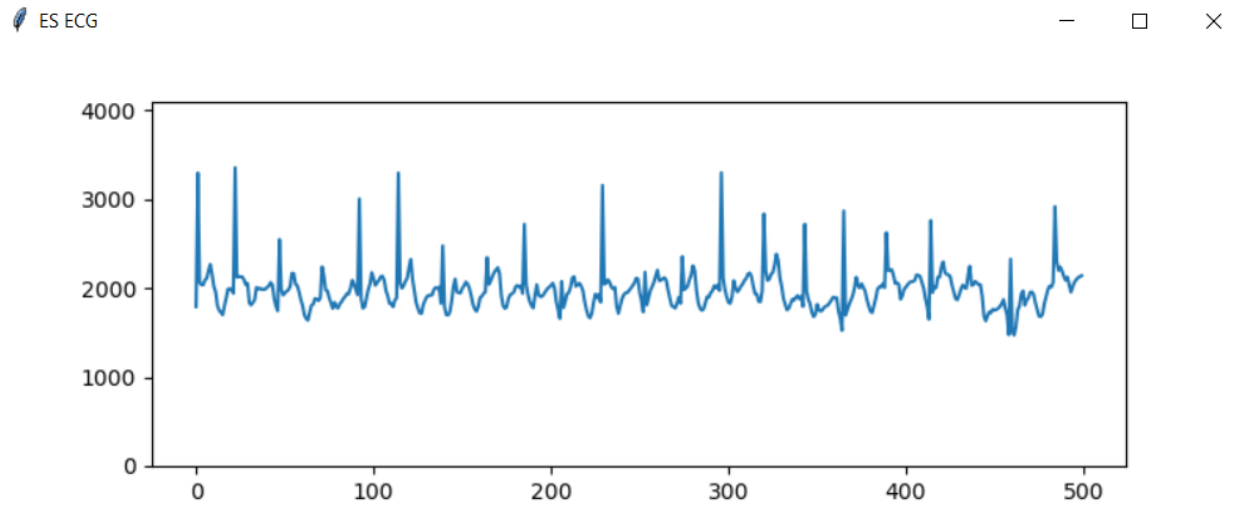
- Python latency made it impossible for the plot to be refreshed in the same frequency as the sampling rate. As a result, displaying time will be more in 1 minute for high sampling rates. However, 1-minute worth of data is still being transmitted and displayed without any loss.

Possible Future Increments:

- Pass the sensor signal through a low-pass filter to eliminate the high frequency noise.
- Transform the x-axis to be time based
- Add compatibility for multiple ports
- Add functionality to change the baud rate

ECG Plot Screenshots:

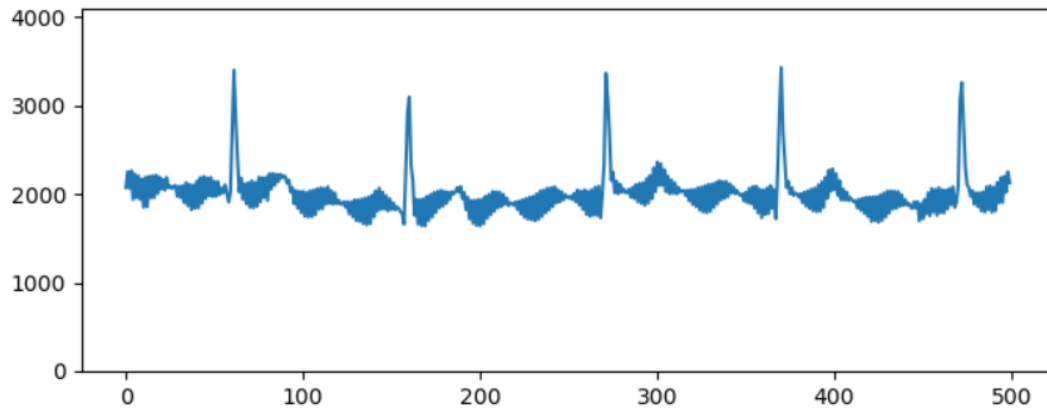
Sampling frequency set to 25 samples per second



Sampling frequency set to 100 samples per second

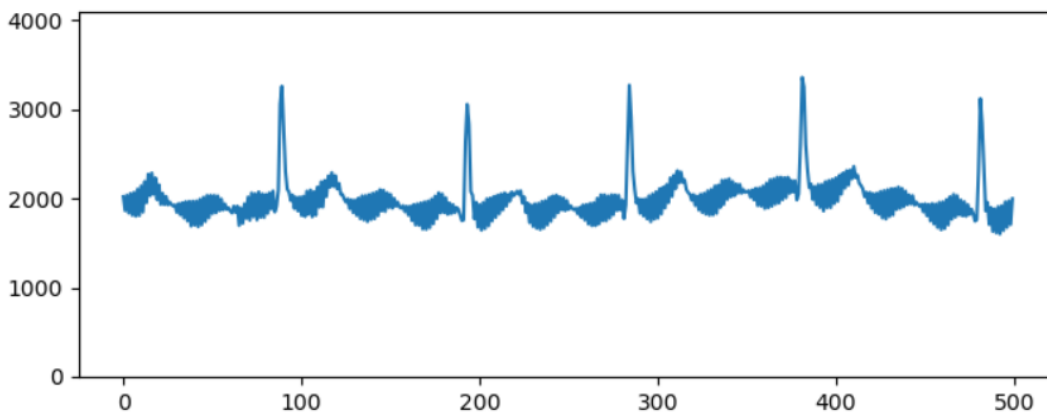
ES ECG

— □ ×



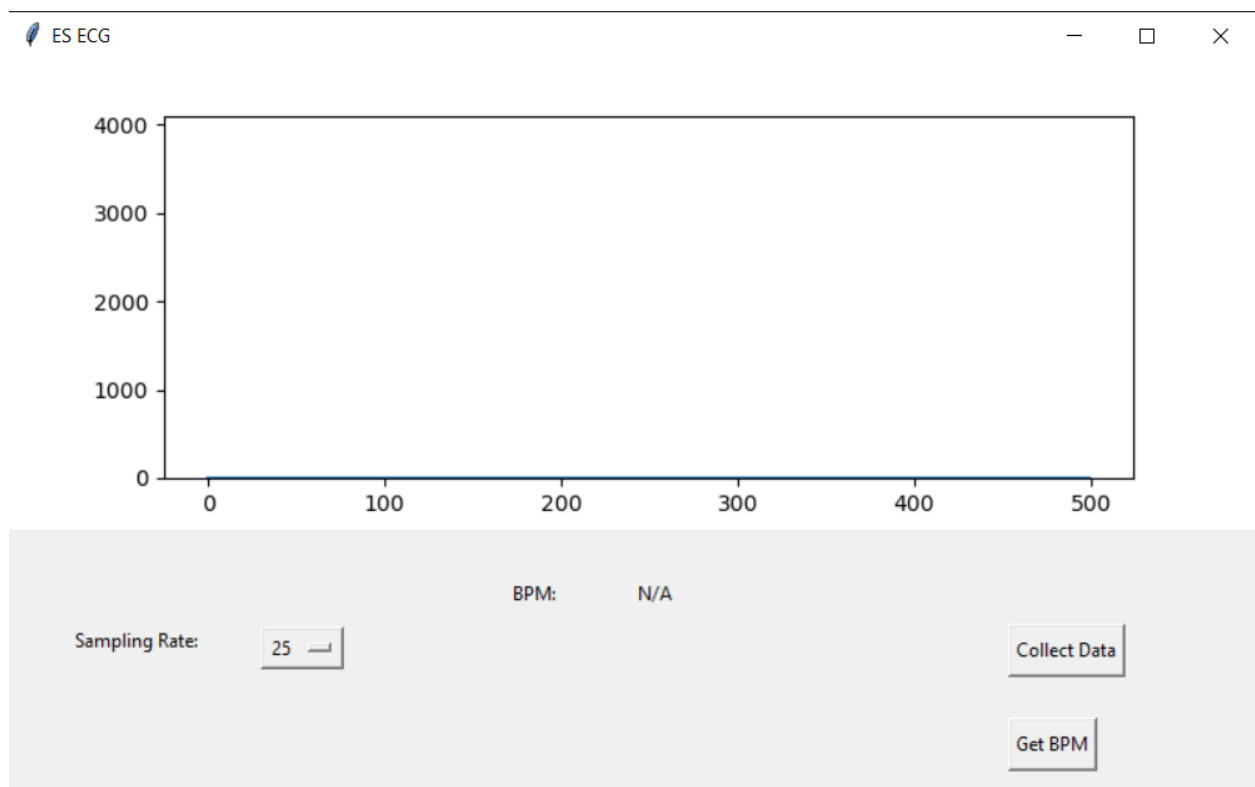
ES ECG

— □ ×

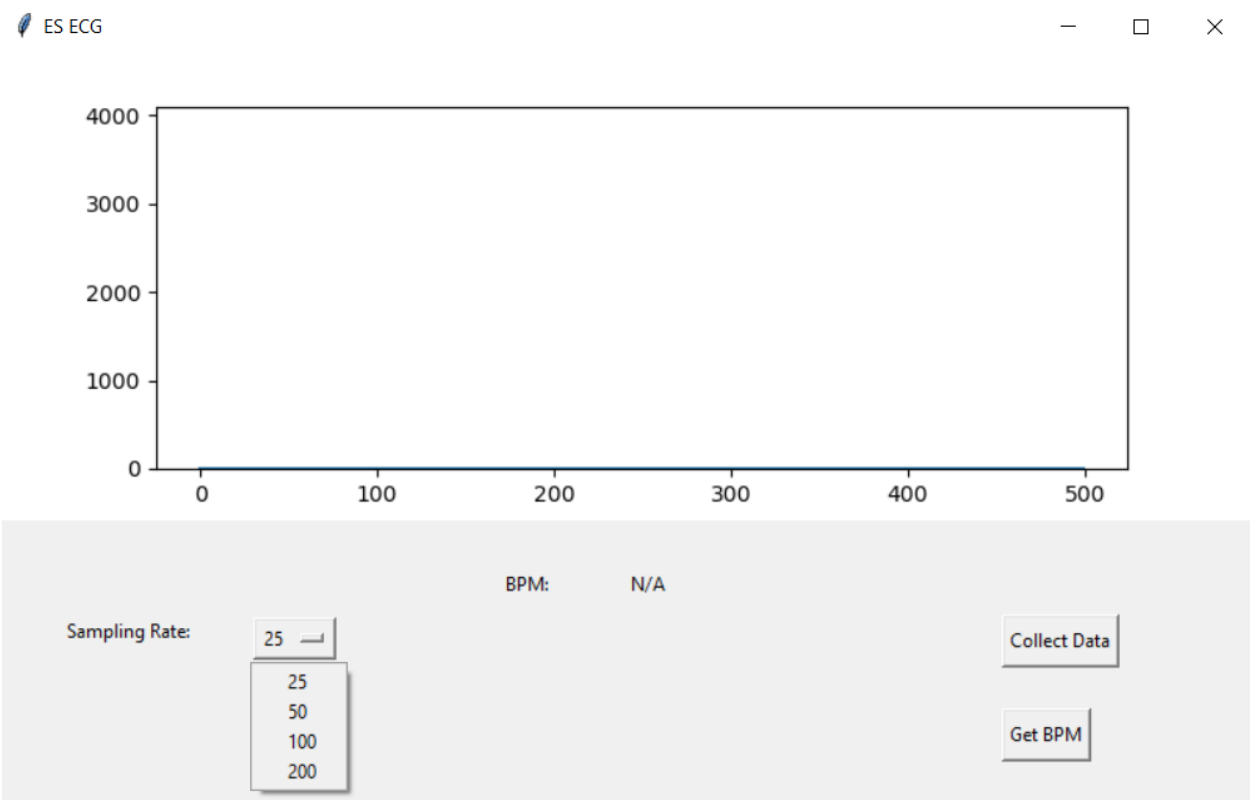


Desktop Application Screenshots:

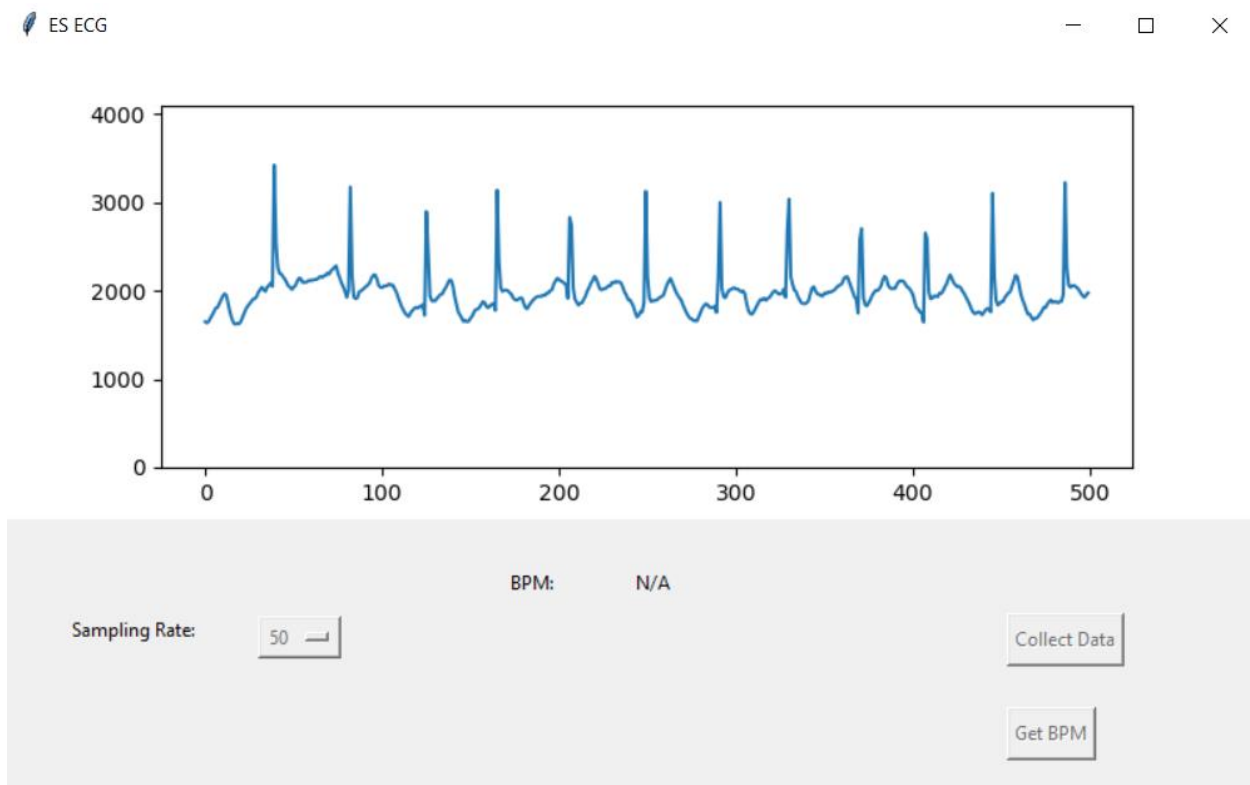
Idle state, immediately after start-up



Sampling rates list



While receiving and plotting data



After plotting ended and requested BPM

