UP: Interface et acquisition

RUP: Raphael VIERA (raphael.viera@emse.fr)

GP : Projet FPGA sécurisé

RGP : Jean-Baptiste Rigaud (rigaud@emse.fr)

Document Title

FPGA Lab Plan: ASCON-Based Encryption, Decryption, and Analysis of ECG Signals

Document Description

This document lists the laboratory exercises that students are required to complete.

Last update

March 09, 2025

Activity 1:

Encrypt / decrypt ECG waveform data using the ASCON engine implemented on Python in order to emulate the FPGA.

1. Encrypt second waveform from the .csv file.
2. Decrypt the encrypted waveform in order to validate that the engine works correctly and that you are recovering the correct waveform.
3. Add these functionalities to your Python Class. Don't forget to comment, update the logging system and document the methods.

Activity 2:

Emulate the reception of encrypted data from the FPGA and decrypt it using the Python ASCON library.

1. Modify the provided Python ASCON script to:
    a. Read encrypted ECG signals via UART through your Python Class.
    b. Decrypt the ciphertext using the same key, nonce, and associated data.
    c. Convert the decrypted hexadecimal values into decimal amplitude values.
2. Verify correct decryption by comparing with the original waveform.
3. Plot the decrypted ECG waveform using matplotlib or similar. You can create a live plot that updates in "real time".
4. Merge multiple ECG waveforms and compute the BPM
    a. Apply FIR filters to identify peaks in the ECG signals.
    b. Measure time between consecutive peaks.
    c. Convert time intervals to BPM.
5. Identify the parts of ECG (PQRST) in order to identify heart problems like Arrhythmias, Ischemia and Infarction, Electrolyte Imbalances, Abnormalities in Conduction etc.
6. Noise Addition & Error Handling: Add artificial noise to ECG data before encryption and implement error detection/correction techniques.

Annex

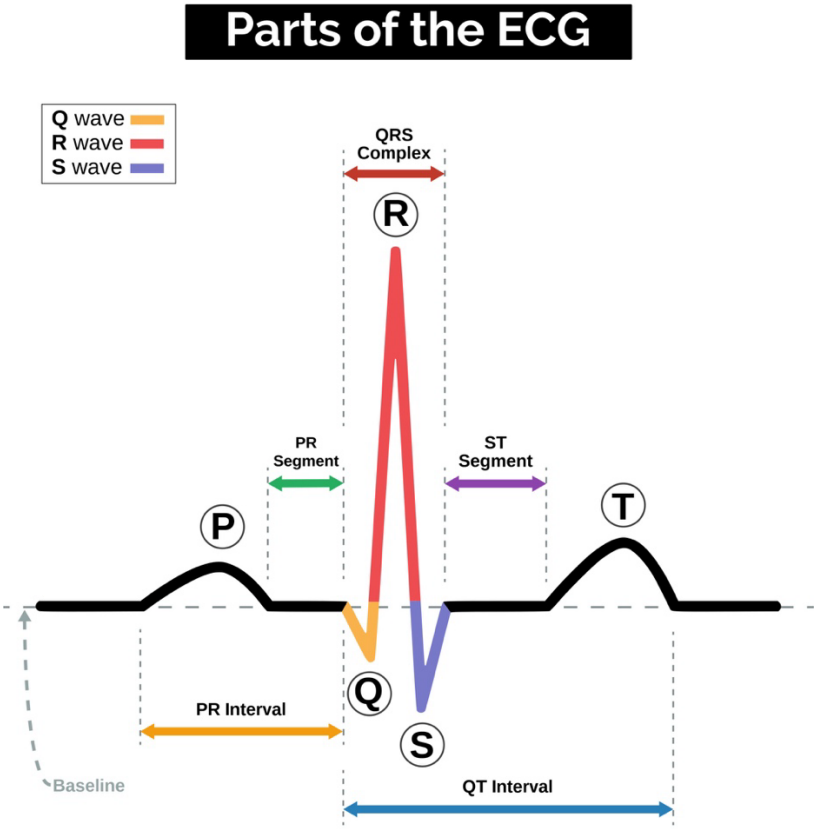

Image 1: Parts of the ECG.

Source: https://geekymedics.com/how-to-read-an-ecg/