STUDY AND EVALUATION OF BIOMETRIC TECHNIQUES: ECG RECOGNITION

Anna Vittoria Damato
Antonio Nardone
Carolina Sbiroli
Raffaele Di Benedetto

Academic Year 2024/2025

Electronic systems for Biometrics and biosensing

Table Of Contents

Table Of Figures

Table Of Tables

Acronyms

BPM – Beats Per Minute
STM – STMicroelectronics
LED – Light Emitting Diode
USB – Universal Serial Bus
COM – Communication
DAC – Digital-to-Analog Converter
ADC – Analog-to-Digital Converter
ST – STmicroelectronics
IDE – Integrated Development Environment
HAL – Hardware Abstraction Layer
TDS –Tektronix Digital Storage
DC – Direct Current
Wi-Fi – Wireless Fidelity
ECG - ElectroCardioGram
RLD - Right Leg Drive
LA – Left Arm
RA – Right Arm
RL – Right Leg
LOD - Leads Off Detection
LFCSP - Lead Frame Chip Scale Package
DSO – Digital Storage Oscilloscope
TFT – Thin-Film Transistor
WVGA – Wide Video Graphics Array
PC – Personal Computer
FFT - Fast Fourier Transform
MSO – Mixed Signal Oscilloscope
DPO – Digital Phosphor Oscilloscope
OS – Operating System
RISC - Reduced Instruction Set Computer
Arm - Advanced RISC Machines
MATLAB - MATrix LABoratory
RTOS - RealTime Operating System
IoT - Internet of Things
IPv6 – Internet Protocol version 6
TLS – Transport Layer Security
DTLS – Datagram Transport Layer Security

Electronic systems for Biometrics and biosensing

MQTT – Message Queuing Telemetry Transport

CoAP – Constrained Application Protocol

FAT – File Allocation Table

LittleFS – Little Flash File System

CMSIS - Cortex Microcontroller Software Interface Standard

RTX - RealTime Ray Tracing

Definitions

I. **Biometrics**: the discipline that studies the measurement of the unique characteristics of the human being, both physical and behavioral.

II. **Biometric technologies**: the set of all techniques that reliably exploit measurable physiological or behavioral characteristics to identify and distinguish an individual from others.

III. **Biometric recognition**: a process that allows the identity of a person to be identified or verified based on unique physical or behavioral characteristics. The general process involves several phases, starting with the acquisition of the biometric data, which is then subjected to pre-processing to improve its quality. Next, distinctive characteristics are extracted that can be stored in a database (enrollment) for later comparison. When a new biometric sample is captured, it is compared to those already stored to determine the user's identity.

IV. **ECG Biometrics**: A method that uses the electrocardiogram to identify a person based on the unique characteristics of their heart signal, such as the shape of P, T, and QRS waves. After the signal is acquired, it is processed and compared with the stored data to determine if there is a match.

V. **Identification Process**: a process that allows an individual to be recognized by comparing their biometric characteristics with those stored in a database. The process begins with the acquisition phase, during which biometric data is collected, which can be taken from images, audio or physiological signals. Subsequently, distinctive characteristics are extracted from this data, a fundamental step in obtaining a biometric template that can be compared with others already present in the database.

At this point, the "1-to-many" comparison takes place, in which the new sample is compared with all the biometric information stored to identify an individual among a series of registered subjects. If the system finds a match, it returns the user's identity; otherwise, identification fails.

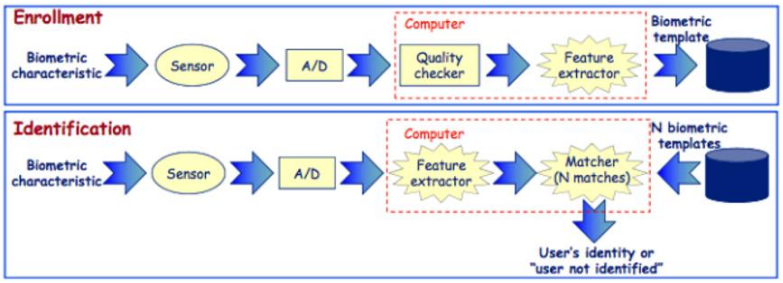


Figure 1: Outline of the identification process in a biometric system

VI. **Authentication Process**: process in which the system checks whether the biometric data provided by the user matches those previously registered. In this process, the user declares their identity (for example, an ID or a username) and proceeds with the acquisition and extraction of characteristics, similar to the identification process. A "1 to 1" comparison is made between the extracted biometric data and the template saved in the database to verify

if they correspond to the declared identity, determining whether authentication has succeeded or failed.

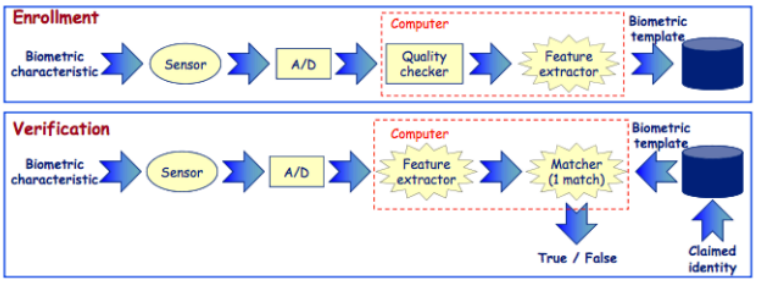


Figure 2: Outline of the authentication process in a biometric system

VII. **Biometric Template**: digital representation of distinctive characteristics extracted from a biometric sample.

VIII. **Fixed Threshold**: A default value used in decision-making systems to determine whether or not a piece of data exceeds a certain limit. In biometrics and other fields, it is employed to establish the critical point beyond which a measure, such as a biometric similarity or security level, is accepted or rejected. Unlike an adaptive threshold, a fixed threshold does not vary based on specific conditions or data, making it simple to implement but potentially less flexible in dynamic environments.

IX. **Performance Evaluation**: In biometric systems, it refers to the process of measuring the accuracy and reliability of the system in correctly recognizing users. This assessment is based on a matching score, which represents the similarity between two fingerprints, and is compared to a fixed threshold to determine whether the comparison is considered a match (matching) or not (not-matching).

X. **Interclass Similarity**: degree of similarity between biometric samples of different individuals. High interclass similarity can reduce the effectiveness of the biometric system, increasing the risk of false positives (False Acceptance Rate, FAR).

XI. **Intraclass Variability**: Variations in biometric samples of the same individual, which may be due to factors such as environmental conditions, physiological changes, or modes of acquisition. High intraclass variability can increase the False Rejection Rate (FRR) and reduce the reliability of biometric recognition.

XII. **Cardiac cycle:** The set of events that occur in the heart during a single beat, with an average duration of 0.8 seconds. It is divided into two main phases:

- Diastole (0.49 sec): the heart muscle relaxes, allowing the heart to fill with blood.
- Systole (0.31 sec): The heart muscle contracts, pumping blood into the blood vessels.

This cycle ensures blood circulation and the proper functioning of the cardiovascular system.

XIII. **ECG:** recording of the electrical changes that occur in the heart muscle during the cardiac cycle. This test shows a series of waves, P-QRS-T, of which the P wave identifies atrial systole, the QRS wave ventricular systole and the T wave the end of the systolic phase. The set of waves provides a trace that represents the electrical activity of the heart, allowing you to monitor the heart rhythm, identify abnormalities and diagnose heart diseases.

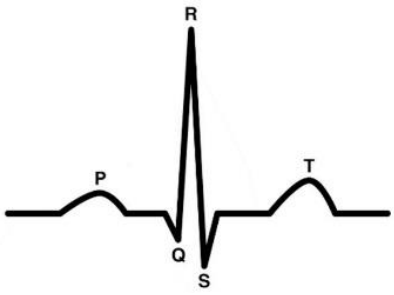


Figure 3: Schematic Representation Of An ECG Showing A Cardiac Cycle

1 OBJECTIVES OF THE REPORT

The aim of this report is to document the activities carried out during the laboratory sessions, in which different recognition techniques based on the processing of electronic signals and biometric data were tested.

The main objectives were structured as follows:

- **ECG signal acquisition:** acquire signals measured with an ECG sensor with the support of an oscilloscope for real-time visualization;
- **Application of pre-processing techniques:** use ECG signals to extract individual biometric parameters;
- **Feature extraction:** the analysis of P, QRS and T heart waves it allowed the identification of peculiar characteristics, such as the duration and amplitude of QRS complexes and heart rate variability.

2 HARDWARE

2.1 Components

2.1.1 STM32 Nucleo F091RC

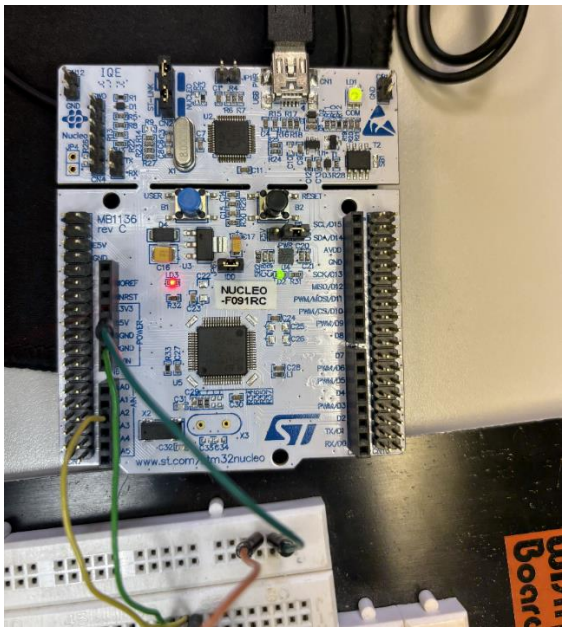


Figure 4: STM32 NUCLEO F091RC

The STM32 Nucleo-64 board provides users with a cost-effective and flexible way to experiment with new concepts and build prototypes by choosing from the various combinations of performance and power consumption offered by the STM32 microcontroller.

LEDs:

- LD1 (USB communication): Indicates the USB communication status, which is useful for viewing data activity or connections when debugging or using the Virtual COM port.
- LD2 (User LED): a customizable LED available to the user. It can be managed via software to report particular states of the application (e.g. error reporting, calculation activity, diagnostic blink, etc.).
- LD3 (Power LED): provides an indication of the power supply of the board, allowing you to quickly understand if the board is correctly powered.

BUTTONs:

- USER: a programmable button that the user can read via software as input (e.g. to implement start/stop functions, mode selection or other events).
- RESET: allows you to reset the microcontroller returning it to its initial state. It is useful both in the development and debugging phases (quick reboot) and in situations where the software needs a hard reset.

BOARD CONNECTORs:

- Arduino Uno V3 connectivity:
    - Layout compatible with standard Arduino shields, simplifying the addition of existing modules or extensions (e.g. motor shields, Wi-Fi, various sensors...).

- ST morpho extension pin headers:
  - Provides full access to all microcontroller I/O pins.
  - Useful for those who need advanced features and want to take advantage of every single peripheral of the chip (e.g. DACs, multiple ADCs, serial interfaces, etc.).
  - Allows integration with ST-specific extension boards (e.g. shields or expansion boards).

FLEXIBLE BOARD POWER SUPPLY:

- ST-LINK USB V_{BUS}: using the USB port connected to the PC, the microcontroller is powered directly by the 5 V provided by the USB cable.
- External source (3.3 V, 5 V, 7 - 12 V): If your design requires more power or a different power supply, you can connect external sources with different voltage ranges.
- Power management access point: there is a test point dedicated to monitoring and measuring consumption.

USB INTERFACEs:

- Virtual COM port: allows you to communicate with the microcontroller via a virtual serial port, useful for logging, debugging, data exchange with the application on PC (e.g. serial terminals, PuTTy).
- Mass storage: the board is recognized as a storage unit (drive). It's a quick way to load firmware onto the microcontroller: simply drag and drop the binary or .hex file into the dedicated folder.
- Debug port: using the integrated ST-LINK programmer/debugger (on board), it is possible to program and debug the microcontroller directly from the IDE (e.g. KeilStudioCloud, or others) without the need for external programmers.

HAL SOFTWARE:

- HAL library: Simplifies access to peripherals through high-level functions, reducing programming complexity. Ready-to-use starting software is available to test functionality (e.g. reading sensors, managing displays, etc.).

2.1.2 Sparkfun Single Lead Heart Rate Monitor - AD8232

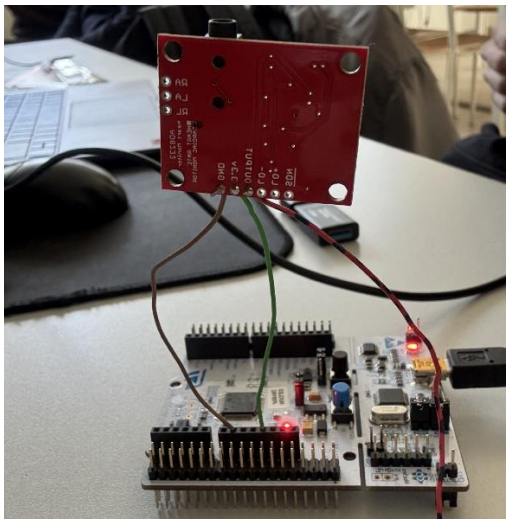


Figure 5: Heart Rate Monitor AD8232

The AD8232 is an integrated circuit designed for the acquisition and conditioning of biopotential signals, with particular reference to single-lead electrocardiogram (ECG) applications. This device is optimized to extract, amplify, and filter small biopotential signals in environments characterized by high noise floor, such as those generated by movement or remote electrode placement. Its architecture allows for easy signal acquisition by low-power analog-to-digital converters (ADCs) or embedded microcontrollers, making it easy to integrate into portable devices and monitoring systems.

At the heart of the device is an instrument amplifier with a fixed gain of 100 dB, designed to provide high common-mode rejection of up to 80 dB from DC to 60 Hz, which is essential for reducing line noise.

Coupled to this amplifier, a two-pole high-pass filter to eliminate motion artifacts and electrode half-cell potentials. In addition, the device integrates an op-amp that can be configured as a three-pole low-pass filter with adjustable gain, allowing the frequency response of the system to be tailored to the specific needs of the application, thus improving signal quality. The user can select the cut-off frequency of all filters to suit different types of applications.

To ensure stable acquisition, the AD8232 includes a dedicated amplifier to guide the reference electrode, commonly used as a "right leg drive" (RLD). This amplifier inverts the common-mode signal present at the inputs of the instrument amplifier and reinjects it into the subject through the reference electrode. This improves the system's common-mode rejection and reduces common-mode voltage variations, helping to achieve a cleaner, interference-free ECG signal.

An important feature built into the device is the Fast Restore mode, which allows fast recovery of the ECG signal after saturation events, such as sudden electrode disconnection. When the device detects a sudden change that causes the amplifier to saturation, the filter cutoff frequency is automatically increased, speeding up the restoration of the valid signal once the electrodes are reattached to the subject and significantly reducing the system's settling time.

The AD8232 supports both two-electrode and three-electrode configurations. In the three-electrode configuration, the instrument amplifier inputs are connected to the electrodes on the left arm (A) and right arm (RA), while the RLD output is connected to the electrode on the right leg (RL).

The device also integrates a "Leads Off Detection" system in both AC and DC mode. In particular, the LOD+ and LOD− pins signal the connection status of the electrodes, allowing the system to detect any disconnections and take appropriate countermeasures.

Electrically, the AD8232 operates on a single supply ranging from 2.0V to 3.5V and has a typical current consumption of 170µA, making it well-suited for portable applications and low-power devices. The output is rail-to-rail, which allows maximum excursion of the output signal, optimizing the use of the available voltage. In addition, the device includes a reference buffer that generates a virtual ground point, simplifying the design of single-power systems.

The AD8232 is available in a 4 mm × 4 mm, 20-lead LFCSP package. Performance is specified from 0°C to 70°C and is operational from -40°C to +85°C.

2.1.3 Tektronix Digital Storage Oscilloscope - TBS1052B-EDU

Figure 6: Tektronix Oscilloscope TBS1052B

The Tektronix TBS1052B is a digital memory oscilloscope (DSO) designed to deliver reliable performance in a compact format, making it ideal for educational, design, and maintenance applications. This instrument is available in different models with bandwidth of 200 MHz, 150 MHz, 100 MHz, 70 MHz and 50 MHz, all of which have two channels. The sampling rate is up to 2 GS/s on all channels, ensuring an accurate representation of the acquired signals. In addition, the record length is 2.5k points per channel, allowing detailed temporal analysis of the signals.

The 7-inch color TFT display, with WVGA (800x480) resolution, offers a clear display of waveforms and critical information. Thanks to the 34 built-in automatic measurements, the analysis of signal parameters is fast and intuitive.

The TBS11052B also includes a dual-window FFT function that allows for simultaneous analysis in the time and frequency domains. Advanced features include waveform limit testing, TrendPlot™ for long-term monitoring, and a dual-channel frequency counter.

Connectivity-wise, the device has a USB 2.0 host port on the front panel, which makes it easy to save data, and a USB 2.0 device port on the back for connecting to a PC. Tektronix's proprietary sampling technology ensures that the claimed sampling rate is maintained across all channels without compromise, ensuring at least ten times oversampling. This allows a faithful representation of the signal characteristics, even when adjusting horizontal settings or using multiple channels simultaneously.

Advanced triggering capabilities, including pulse triggering and line-selectable video triggering, allow you to quickly isolate signals of interest. Once the data is acquired, the built-in mathematical functions allow waveforms to be added, subtracted, and multiplied, while the dedicated FFT function allows waveforms to be displayed in the time and frequency domains simultaneously. The dual-channel frequency counter provides accurate measurements, with independent control of the trigger level for each channel, making it easy to monitor two different frequencies simultaneously.

The TBS1000B's user interface is inspired by Tektronix's MSO/DPO series, providing quick access to all oscilloscope functions. The high-resolution "Pan & Zoom" function allows you to examine signal details up to ten times the normal resolution.

2.2 Circuits

2.2.1 ECG sensor

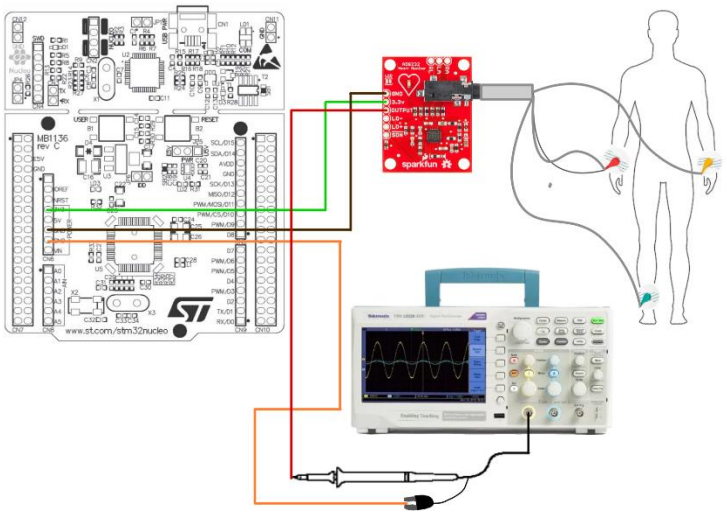


Figure 7: Circuit Diagram For ECG Acquisition

To acquire the electrocardiographic (ECG) signal via the microcontroller, an ECG sensor with the following connections was used:

- **Power supply**: The sensor was powered with 3.3V, connecting the power line of the microcontroller to the power terminal of the ECG sensor. The GND terminal of the sensor has been connected to the ground of the microcontroller.
- **Signal monitoring:** The output of the ECG sensor was connected to the oscilloscope probe, while a second GND of the microcontroller was connected to the oscilloscope reference terminal. This setup made it possible to view the ECG signal on a screen in real time, allowing a preliminary analysis of the trace and the verification of the correct acquisition of the signal.

3 FIRMWARE

3.1 MBED OS

Mbed OS is an RTOS operating system developed by Arm and designed for embedded devices based on Cortex-M microcontrollers. Provides a complete, IoT-optimized development environment, with support for connectivity, security, and efficient energy management.

This operating system offers an RTOS kernel that allows for multitasking task management, thread synchronization, and efficient use of system resources. It adopts an event-driven programming model, which is especially suitable for low-power devices. It also integrates network stacks and communication protocols such as IPv6, TLS/DTLS, MQTT, and CoAP, making it easy to connect with cloud services and edge devices. The native file system manager supports several formats, including FAT and LittleFS, which are optimized for use with flash memory. Communication security is ensured by the Mbed TLS encryption library, which is designed to operate on resource-constrained devices.

Mbed OS is compatible with a wide range of development boards based on Arm Cortex-M microcontrollers and offers an ecosystem of ready-to-use drivers for different peripherals and sensors. Integration with the Mbed Cloud framework facilitates remote deployment and management of IoT devices.

Keil Studio is the IDE provided by Arm for programming embedded applications on Mbed OS. This platform includes advanced tools for writing, compiling, and debugging code, with support for CMSIS (Cortex Microcontroller Software Interface Standard) and the Arm libraries for Cortex-M microcontrollers. The system supports programming in C/C++ and offers advanced real-time debugging, simulation, and hardware emulation capabilities. With the integration of Arm Compiler 6, it enables the generation of optimized code to reduce power consumption and improve performance. In addition, the implementation of Keil RTX RTOS ensures efficient thread and resource management, improving the reliability of embedded applications. The debugging and tracing system allows a detailed analysis of the software's behavior, allowing the optimization of critical applications.

3.1.1 ECG Recognition

The ECG data was saved directly via the oscilloscope, without the need to implement specific firmware on the microcontroller. The oscilloscope acquired the ECG signal in real time, and thanks to its data export function, it was possible to store the recordings directly on a USB stick. This approach simplified the data collection process, avoiding the need for an additional communication interface between the microcontroller and the storage system.

3.2 Matlab

MATLAB is a programming environment and software developed by MathWorks, widely used for numerical computing, simulation, and data visualization.

This software is optimized for matrix and vector operations and supports advanced mathematical functions such as linear algebra, interpolation, Fourier transforms, differential equations, and optimization. The interactive programming environment offers an intuitive user interface with a script editor and command window, allowing for both structured programming and interactive experimentation.

One of the distinguishing features of MATLAB is the presence of numerous specialized toolboxes that expand its capabilities in different application areas. These include Signal Processing Toolbox, for signal analysis and filtering, Fourier transforms (FFT), and FIR/IIR filter design, Statistics and Machine Learning Toolbox, statistical analysis, predictive modeling, clustering, and machine learning.

In addition, MATLAB allows you to create advanced 2D and 3D graphs with extensive customization possibilities, making data representation and analysis particularly effective.

3.2.1 ECG Analysis

```
clc;
clear all;
close all;

% Set Default Parameters for Figures
%set(0, 'DefaultFigureColor', 'w'); % White background
set(0, 'DefaultAxesFontSize', 14); % Axis Font Size
%set(0, 'DefaultAxesFontWeight', 'Bold'); % Bold text
%set(0, 'DefaultLineLineWidth', 2); % Line thickness
set(0, 'DefaultAxesLineWidth', 1); % Axle thickness
%set(0, 'DefaultAxesGridLineStyle', '--'); % Dotted grid style
%set(0, 'DefaultAxesXGrid', 'on', 'DefaultAxesYGrid', 'on'); % Enable the grid
%set(0, 'DefaultFigurePosition', [200 200 800 600]); % Window size and position
set(0, 'DefaultLegendFontSize', 14);

% Name of the CSV file
file_name = 'ecg_raffaele.CSV';

% Reading the data from the CSV file
raw_data = readmatrix(file_name, 'NumHeaderLines', 19); % Skip the first 19 header lines

fs = 1000; % Sampling frequency in Hz (1 kHz)

% Extracting the time and amplitude columns
time = raw_data(:, 4);          % Time column (seconds)
ecg_signal = raw_data(:, 5);    % Amplitude column (Volt)
ecg_signal = ecg_signal -1.8;

% Low-pass filtering to reduce noise
fc = 30; % Cutoff frequency (Hz)
[b, a] = butter(4, fc / (fs / 2), 'low'); % Design a low-pass Butterworth filter
filtered_ecg = filtfilt(b, a, ecg_signal); % Apply the filter to the ECG signal

% R-peak detection using the filtered ECG signal
[peaks_R, locs_R] = findpeaks(filtered_ecg, time, 'MinPeakHeight', 0.05, 'MinPeakDistance',
0.6);

% Initialize empty arrays for P, Q, S, and T peaks
locs_Q = []; locs_S = []; locs_P = []; locs_T = [];

% Loop through each detected R-peak to find the corresponding Q, S, P, and T peaks
for i = 1:length(locs_R)

    % Region around the R-peak to define the search window for Q, S, P, and T
    region_start = round((locs_R(i) - 0.3) * fs); % Start of the window (0.3 seconds before R-
peak)
    region_end = round((locs_R(i) + 0.3) * fs);   % End of the window (0.3 seconds after R-
peak)

    % Ensure that the indices are valid within the signal length
    region_start = max(1, region_start); % At least 1 (avoid negative indices)
    region_end = min(length(filtered_ecg), region_end); % Do not exceed the signal length

    % Index of the R-peak
    R_idx = round(locs_R(i) * fs);

    % Define the window for the Q-peak (e.g., 60 ms before the R-peak)
    region_start_Q = round((locs_R(i) - 0.06) * fs);  % 60 ms before the R-peak
    region_end_Q = round((locs_R(i) - 0.01) * fs);   % Up to the R-peak

    % Ensure that the indices are valid
    region_start_Q = max(1, region_start_Q);  % At least 1
    region_end_Q = min(length(filtered_ecg), region_end_Q);  % Do not exceed the signal length
```

```
    % Find the minimum value (Q) in the window
    [Q_val, Q_rel_idx] = min(filtered_ecg(region_start_Q:region_end_Q));

    % Convert the relative index within the window to a global index
    Q_global_idx = region_start_Q + Q_rel_idx - 1;

    % Calculate the time corresponding to the Q-peak
    locs_Q(i) = time(Q_global_idx);  % Time of the Q-peak

    % Define the window for the S-peak (e.g., 5 ms after R-peak to 30 ms after)
    region_start_S = round((locs_R(i) + 0.005) * fs);  % 5 ms after R-peak
    region_end_S = round((locs_R(i) + 0.03) * fs);    % Up to 30 ms after R-peak

    % Ensure that the indices are valid
    region_start_S = max(1, region_start_S);  % At least 1
    region_end_S = min(length(filtered_ecg), region_end_S);  % Do not exceed the signal length

    % Find the minimum value (S) in the window
    [S_val, S_rel_idx] = min(filtered_ecg(region_start_S:region_end_S));

    % Convert the relative index within the window to a global index
    S_global_idx = region_start_S + S_rel_idx - 1;

    % Calculate the time corresponding to the S-peak
    locs_S(i) = time(S_global_idx);  % Time of the S-peak

    % Define the window for the P-peak (e.g., 200 ms before R-peak)
    region_start_P = round((locs_R(i) - 0.2) * fs);  % 200 ms before R-peak
    region_end_P = round((locs_R(i) - 0.1) * fs);   % Up to 100 ms before R-peak

    % Ensure that the indices are valid
    region_start_P = max(1, region_start_P);  % At least 1
    region_end_P = min(length(filtered_ecg), region_end_P);  % Do not exceed the signal length

    % Find the maximum value (P) in the window
    [P_val, P_rel_idx] = max(filtered_ecg(region_start_P:region_end_P));

    % Convert the relative index within the window to a global index
    P_global_idx = region_start_P + P_rel_idx - 1;

    % Calculate the time corresponding to the P-peak
    locs_P(i) = time(P_global_idx);  % Time of the P-peak

    % Define the window for the T-peak (e.g., 100 ms after R-peak)
    region_start_T = round((locs_R(i) + 0.1) * fs);  % 100 ms after R-peak
    region_end_T = round((locs_R(i) + 0.3) * fs);   % Up to 300 ms after R-peak

    % Ensure that the indices are valid
    region_start_T = max(1, region_start_T);  % At least 1
    region_end_T = min(length(filtered_ecg), region_end_T);  % Do not exceed the signal length

    % Find the maximum value (T) in the window
    [T_val, T_rel_idx] = max(filtered_ecg(region_start_T:region_end_T));

    % Convert the relative index within the window to a global index
    T_global_idx = region_start_T + T_rel_idx - 1;

    % Calculate the time corresponding to the T-peak
    locs_T(i) = time(T_global_idx);  % Time of the T-peak

end
```

```
% Plot the ECG signal and the detected peaks
figure;
plot(time, filtered_ecg, 'k', 'DisplayName', 'ECG Signal', 'LineWidth', 1); % Plot the filtered
ECG signal
hold on;
plot(locs_R, peaks_R, 'ro', 'DisplayName', 'R Peaks', 'LineWidth', 2, 'MarkerSize', 8);  % R
Peaks con marker più grandi
plot(locs_P, interp1(time, filtered_ecg, locs_P), 'go', 'DisplayName', 'P Peaks', 'LineWidth',
2, 'MarkerSize', 8);  % P Peaks con marker più grandi
plot(locs_Q, interp1(time, filtered_ecg, locs_Q), 'bo', 'DisplayName', 'Q Peaks', 'LineWidth',
2, 'MarkerSize', 8);  % Q Peaks con marker più grandi
plot(locs_S, interp1(time, filtered_ecg, locs_S), 'o', 'DisplayName', 'S Peaks', 'LineWidth',
2, 'MarkerSize', 8,"MarkerEdgeColor","[0.9290 0.6940 0.1250]");  % S Peaks con marker più grandi
plot(locs_T, interp1(time, filtered_ecg, locs_T), 'mo', 'DisplayName', 'T Peaks', 'LineWidth',
2, 'MarkerSize', 8);  % T Peaks con marker più grandi

legend();  % Display the legend
xlabel('Time [s]');  % Label the x-axis
ylabel('Amplitude [mV]');  % Label the y-axis
title('ECG Signal with detected peaks');  % Title of the plot
grid on;  % Enable grid for better visualization

% Feature calculations
R_heights = max(peaks_R - interp1(time, filtered_ecg, locs_S));
S_heights = interp1(time, filtered_ecg, locs_S);

max_R = max(R_heights);
min_R = min(R_heights);
mean_R = mean(R_heights);
std_R = std(R_heights);

QRS_durations = locs_S - locs_Q;
mean_QRS_duration = mean(QRS_durations);

RR_intervals = diff(locs_R);
mean_RR = mean(RR_intervals);
heart_rate = 60 / mean_RR;

% Calculation of the average interval PQ
PQ_intervals = locs_P - locs_Q; % Time difference between P and Q
mean_PQ_interval = mean(PQ_intervals);

% ST Mean Interval Calculation
ST_intervals = locs_T - locs_S; % Time difference between S and T
mean_ST_interval = mean(ST_intervals);

% Displaying results
fprintf('Max R: %f\n', max_R);
fprintf('Min R: %f\n', min_R);
fprintf('Mean R: %f\n', mean_R);
fprintf('Std R: %f\n', std_R);
fprintf('Mean QRS duration: %f s\n', mean_QRS_duration);
fprintf('Mean RR interval: %f s\n', mean_RR);
fprintf('Mean PQ interval: %f s\n', mean_PQ_interval);
fprintf('Mean ST interval: %f s\n', mean_ST_interval);
fprintf('Heart rate: %f bpm\n', heart_rate);
```

Figure 8: ECG Signal Processing And Analysis Code With Peak Detection

The Code starts by reading data from a CSV file, specifying the file name, and using the *readmatrix* function to load the data into the *raw_data* vector. The *fs* sampling rate of 1000 Hz is defined, and the ECG signal time and amplitude columns are extracted into the *time* and *ecg_signal* variables, respectively.

The ECG signal displayed on the oscilloscope had an upward shift of 1.8 mV, due to an offset introduced by the oscilloscope. This shift, while not altering the morphology of the signal, could

have made both its visual interpretation and the accurate extraction of the features necessary for the analysis more complex. To correct this problem and return the signal to a more appropriate reference level, a downward translation of 1.8 mV was applied, thus eliminating the offset. In this way, the ECG trace was clearer and better aligned with the physiological baseline, facilitating both the reading and the automated analysis of cardiac parameters.

To reduce noise, a low-pass filter is applied to the ECG signal using a Butterworth filter with a cut-off frequency *fc* set at 30 Hz, parameter set after spectral analysis since the spectral components are below 30 Hz. The filter is implemented with the *butter* function, and the *filter* function applies the filter to the signal, producing the *filtered_ecg signal*, which is the result of the filtering.

The code continues with the detection of the R peaks in the filtered signal. Using the *findpeaks* function, R peaks are identified, specifying that the minimum height of a peak must be 0.05 mV and that the minimum distance between peaks must be 0.6 seconds. The R peaks and their time positions are stored in the *variables peaks_R* and *locs_R*.

Next, the code identifies the peaks of the Q, S, P, and T waves. For each R peak found, the program defines time windows around the R peak to look for peaks in nearby regions. The windows for the Q, S, P, and T waves are defined to cover specific portions of the signal around the R peaks.
The `min` or `max` function is used to find the minimum or maximum values in the selected windows, depending on the wave you are looking for (for example, Q is a minimum and P is a maximum). The relative indices within the windows are then transformed into global indices in the full signal, and the time corresponding to each peak is saved in the respective variables (locs_Q*, locs_S, locs_P, locs_T*).

Finally, the code generates a graph to display the filtered ECG signal and the peaks detected. The filtered ECG signal and the R, P, Q, S, and T peaks are plotted, using distinctive colors for each type of peak.

Once the peaks have been plotted, the code proceeds to calculate different characteristics of the signal. Features such as the minimum and maximum amplitude of the R wave, the mean amplitude of R, the standard deviation of the amplitudes of R, the mean duration of QRS complex, PQ, ST, RR are analyzed.
Another key aspect is heart rate analysis, which is calculated through the averaging of RR intervals, i.e. the time between two consecutive R-peaks in BPM.
At the end of the process, all information is printed to allow quantitative analysis of the ECG signal.

4 RESULTS

4.1 ECG Analysis

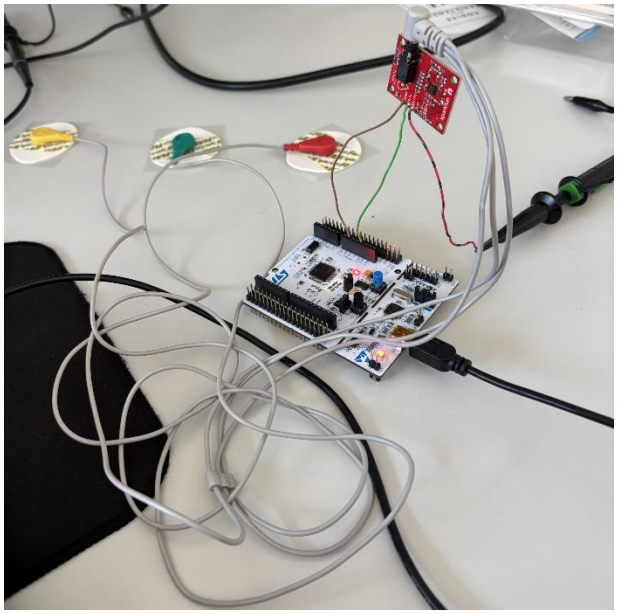


Figure 9: Microcontroller, Electrodes, Heart Rate Monitor and Oscilloscope

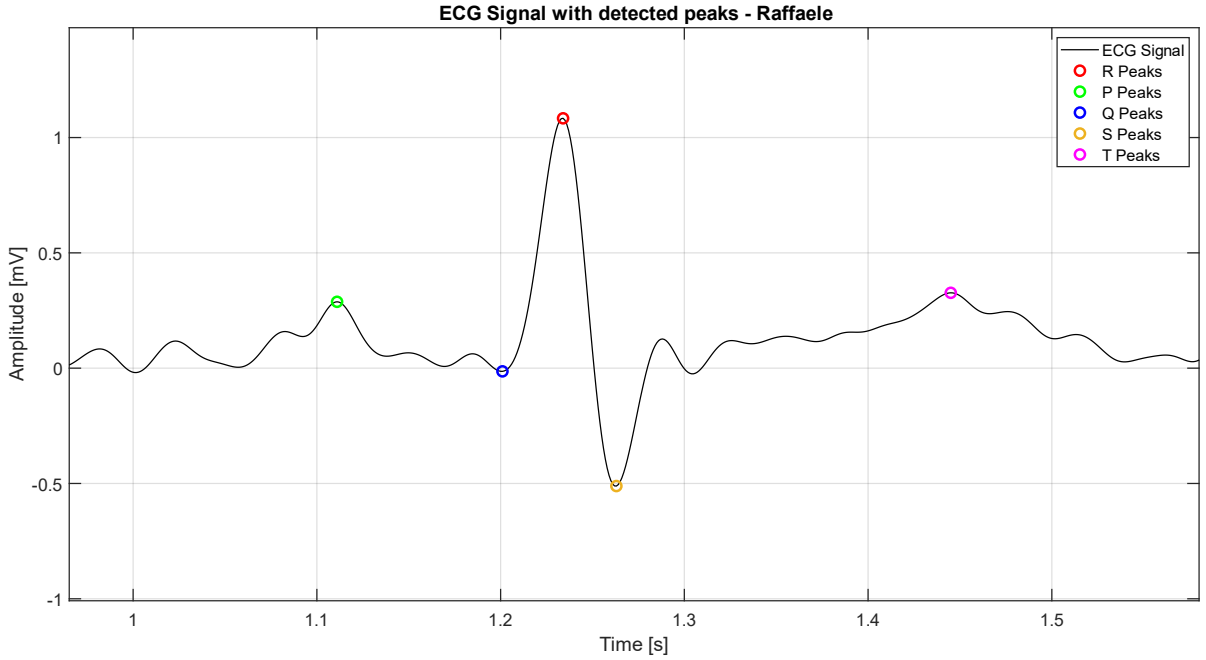
The goal of this exercise was to process the acquired cardiac signals and try to extract significant features (Figure 8). The ECG is the recording of the electrical changes that occur in the heart muscle during the cardiac cycle (Def. XII) and shows a series of waves that are marked with P-QRS-T (Def. XIII).

For each user, the signal has been sampled at 1 kHz and each sample contains information about the instant and relative amplitude of the ECG in mV.

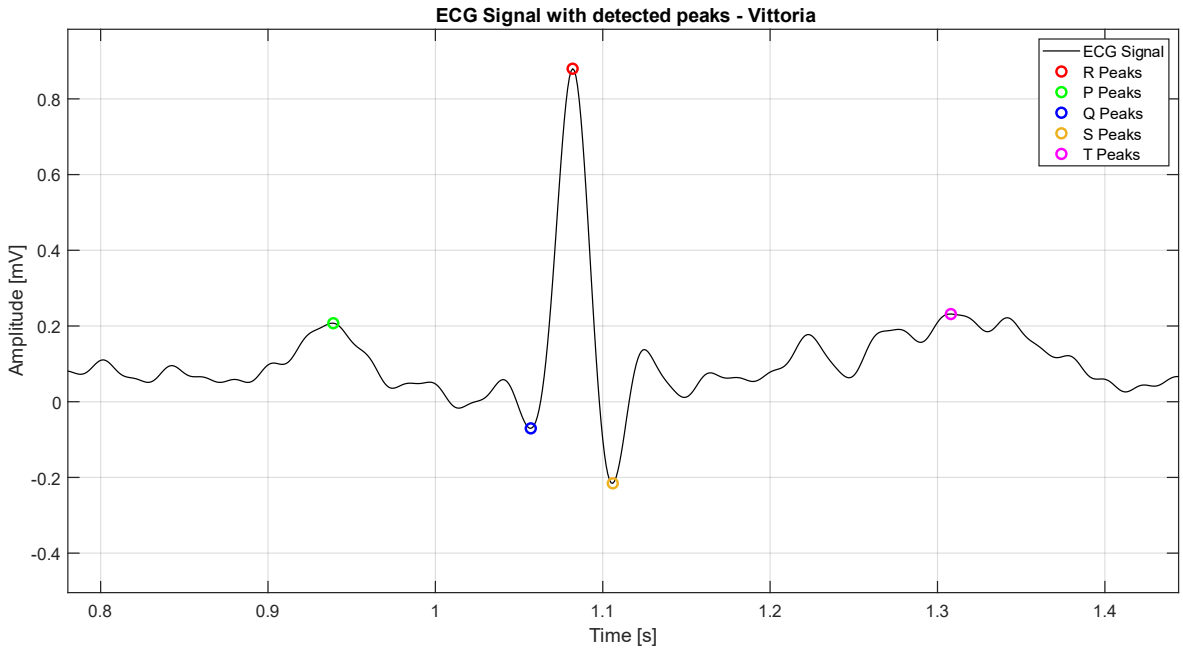
It was chosen to filter ECG signals with a 30 Hz low-pass filter because the main waves (P, QRS and T) are typically located below this frequency, while the high-frequency components are often linked to e.g. muscle noise and electromagnetic interference. The use of a filter made it possible to improve the quality of the signal, making the fundamental details of cardiac activity clearer.

Trying to adopt a filter with a lower cut-off frequency, for example 20 Hz, it was found that the QRS complex, which contains significant components at those frequencies, was found to be excessively smoothed, compromising the accuracy in the analysis of the morphology of the heartbeat. On the other hand, a filter with a higher cut-off, such as 50 Hz, would have allowed the shape of the QRS to be better preserved, but would have been less effective in eliminating noise.

An attempt was also made with a bandpass filter with a range of 0.5-40 Hz, with the aim of eliminating low-frequency noise, such as that due to breathing, and also high-frequency noise, such as that due to muscle contractions near the signal detection probes. However, the results were not what was hoped for because the filter greatly reduced the amplitude of the QRS wave and made the P and T points indistinguishable from the noisy component that shares the same frequencies as the ECG signal, making subsequent signal processing ineffective.



(a)



(b)

Figure 10: Focus On Single ECG Cardiac Cycle Identification For Each User

Subsequently, the identification of the R peaks was carried out, from which the identification of the Q-S-P-T waves was carried out. First of all, the P and Q waves were identified, respectively: atrial systole, detectable in a range between 100 - 200 ms, and the first negative deflection, in a range between 60 - 10 ms, both before R. The analogous operation was performed for S, which is the first negative deflection in the range 5 - 30 ms and T, ventricular repolarization between 100 - 300 ms, both after R.

The result of the search for the various peaks can be observed in *Figure 10*(a) e (b) relating respectively to Raffaele and Vittoria: the filtered ECG signal was traced in black while each of the previously mentioned points was circled with a different color.

Analyzing the two signals, it can be seen that in the case of Raffaele the R peaks reach values close to 1.8 mV, while in Vittoria they stop at around 1.1 mV. This difference can be affected by several factors, including the location of the electrodes, the impedance of the skin, and the level of filtering applied to the data.

From the point of view of regularity, Raffaele's ECG signal appears relatively constant, with uniform distances between the R peaks and a homogeneous waveform. The Vittoria's signal, on the other hand, has a slight variability between the QRS complexes and in the characteristics of the P and T waves, which could result from a suboptimal acquisition.

In *Table 1* the main calculated parameters are shown, which allow the comparison of the values obtained with reference values and the identification of any deviations.

User	Features	Extracted Values	Expected Values
Raffaele	Max Amplitude of R peak	1.781224 mV	0.5 – 2.5 mV
	Min Amplitude of R peak	1.704511 mV	0.5 – 2.5 mV
	Mean Amplitude of R peak	1.749575 mV	1.2 – 1.8 mV
	Standard deviation of the peak amplitude of R	0.040077 mV	Not Standard
	Mean QRS duration	0.058667 s	0.06 – 0.10 s
	Mean RR duration	0.717000 s	0.6 – 1.0 s
	Mean PQ interval	0.093667 s	0.12 – 0.20 s
	Mean ST interval	0.197333 s	0.08 – 0.12 s
	Heart rate (HR)	83.6 bpm	60 – 100 bpm
Vittoria	Max Amplitude of R peak	1.095267 mV	0.5 – 2.5 mV
	Min Amplitude of R peak	1.016879 mV	0.5 – 2.5 mV
	Mean Amplitude of R peak	1.050583 ms	1.2 – 1.8 mV
	Standard deviation of the peak amplitude of R	0.040331 mV	Not Standard
	Mean QRS duration	0.05166 s	0.06 – 0.10 s
	Mean RR duration	0.683500 s	0.6 – 1.0 s
	Mean PQ interval	0.114333 s	0.12 – 0.20 s
	Mean ST interval	0.201667 s	0.08 – 0.12 s
	Heart rate (HR)	87.7 bpm	60 – 100 bpm

Table 1: ECG Extracted Features

In general, the data obtained are largely consistent with the reference ranges, but have some discrepancies.

As for the amplitude of the peak R, the measured values are within the limits set for both users, with the exception of the average for Vittoria, which is slightly lower than the expected range. This could indicate a lower amplitude of the acquired ECG signal, potentially due to physiological variables or factors related to the acquisition itself. The lifetime of the QRS complex is also slightly below the reference values for both users, while remaining close to the lower limit. This

aspect could be influenced by signal segmentation or the presence of residual noise in the acquisition.

The PQ and ST intervals show some more marked variations than expected. Notably, Raffaele's PQ interval is shorter than expected, while Vittoria's is near the lower limit. The ST range, on the other hand, is longer than the expected range for both users, which may indicate that the waves found are not extremely accurate due to the filtering applied to eliminate noise. On the other hand, RR interval duration and heart rate are in line with baseline values, suggesting that heart rhythm is regular for both users.

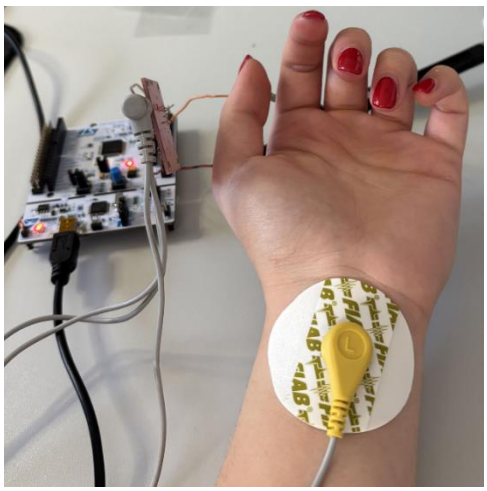


Figure 11: Electrode For ECG Acquisition

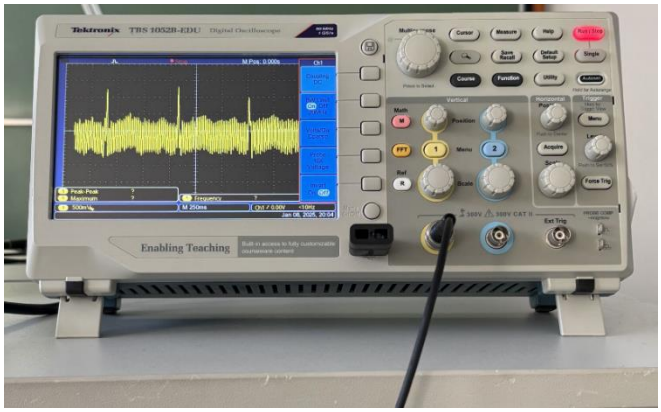


Figure 12: ECG Visualization On The Oscilloscope

5 CONCLUDING NOTES AND POSSIBLE OBSERVATIONS

5.1 Observations

Due to experimental conditions, it was not possible to place the ECG suction cups in the optimal configuration. This affected the quality of the acquired signal, introducing possible distortion and unwanted noise. To mitigate these effects and improve the analysis, targeted filtering was applied, aimed at reducing the disturbing components without altering the fundamental characteristics of the cardiac tracing. While this allowed for a cleaner and more interpretable signal, limitations related to the arrangement of the electrodes may still have affected the overall accuracy of the data.

5.2 Conclusion

The laboratory experience has allowed us to deepen the process of acquisition, processing and extraction of features from ECG signals. The activity highlighted the importance of proper signal acquisition, as data quality directly affects the subsequent processing and analysis steps.

The experimental activities carried out allowed us to analyze some biometric methodologies for ECG recognition, highlighting the main challenges related to their implementation and data acquisition.

ECG-based recognition confirmed the presence of unique biometric features in cardiac signals, making it a promising method for individual identification. However, the quality of the acquired signal was affected by the instrumentation used, highlighting the need for accurate calibration and advanced pre-processing techniques to reduce noise and artifacts.

In general, the results obtained confirm that biometric recognition is a technology with high application potential, but that it requires special attention in the signal acquisition phase and in data processing to obtain reliable performance. Improved filtering techniques, optimization of acquisition parameters and integration with machine learning models could be future developments to make these systems more robust and efficient.