Co-funded by the
Erasmus+ Programme
of the European Union

ITASDI PROJECT

Innovative Teaching Approaches in development of Software Designed Instrumentation and its application in real-time systems

Erasmus+ KA2 2018-1-RS01-KA203-000432

Course „Advanced LabVIEW Applications“

Laboratory no. 5 - Queued Message Handler

Leader Partner: [1]Warsaw University of Technology
Authors: Dariusz Tefelski [1]

Circulation: Public
Version: 02
Stage: Final
Date: 10.09.2019

Funding Disclaimer
This project has been funded with support from the European Commission. This communication reflects the views only of the authors, and the Commission cannot be held responsible for any use which may be made of the information contained therein.

1 DNA analyzer

1.1 Goal

Create an application, which simulate the biomedical laboratory working. The application consists of the blood donation points and laboratory, in which DNA is analyzed. The DNA consists of two chains, which are build based on: adenine (A), cytosine (C), guanine (G) and thymine (T). The proper DNA chains are connected accoring with scheme: A with T and G with C. If the A will be connected with G or C the DNA chains have anomalies.

1.2 General requirements

- The application should be hierarchical and scalable. Remember to use subVIs.
- Use the Queued Message Handler (QMH).
- **Do not use the state machine, flat sequence inside the QMH.**
- **Do not use the local, global or shared variables.**
- Close all opened references and handles.
- Application shouldn't crash. Inform the user about the errors using the error cluster or a dialog box.
- Remember to prepare well documented code. Especially remember about: labels on long wires, description showing in context help, tip strips of controls and labels of constant values.
- **All subVI should have intuitive icon and description, which will be shown in contex help.**

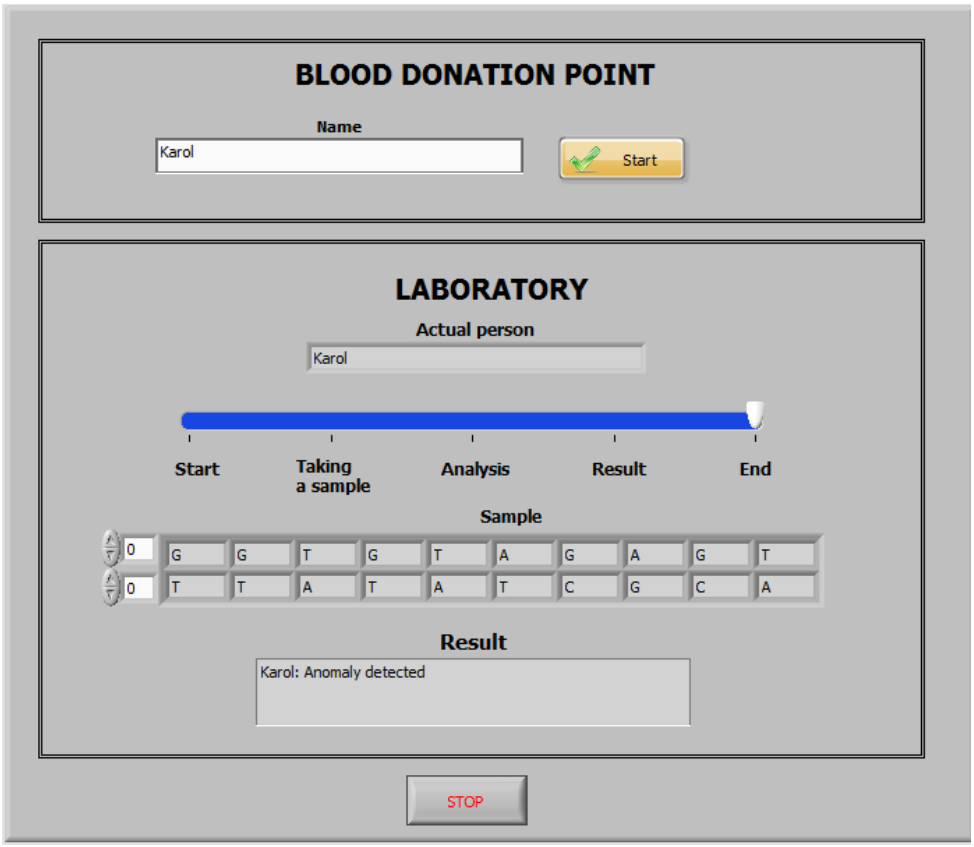
1.3 Description

- Front panel contains two sections:
- User fills the *Name* control. When the user clicks on *Start* button, a task will be added to queue.
- *Actual person* indicator shows, whose sample is analyzed.
- *Progress* indicator shows actual status of work.
- *Sample* indicator is an array of the bases of DNA. The first raw is the one DNA chain and the second raw is the second DNA chain.
- *Result* indicator is used to display the result of analysing the DNA.

1.4 Realization

- Use the Queued Message Handler Design Pattern, in which Event Handling Loop (EHL) will add a task and a name to the queue after clicking the *Start* button. Generation and analysis of the sample will be realized in the Message Handling Loop (MHL).
- Each stage of DNA analysis should take 2000 ms of time.
- The DNA analysis consists of following steps:

    1. *Taking a sample* - create subVI, in which the DNA sample will be randomly fill. Remember that sometimes DNA should be generated with anomalies e.g. A connected with G. Additionally sometimes the blood sample is too small to receive the DNA chains. The good sample should have lenght equal to 10. The wrong sample will have smaller lenght. Remember to add this functionality to subVI. If the sample is with appropiate leght, the next step will be *Analysis*. In other case, the next step will be *Result*.

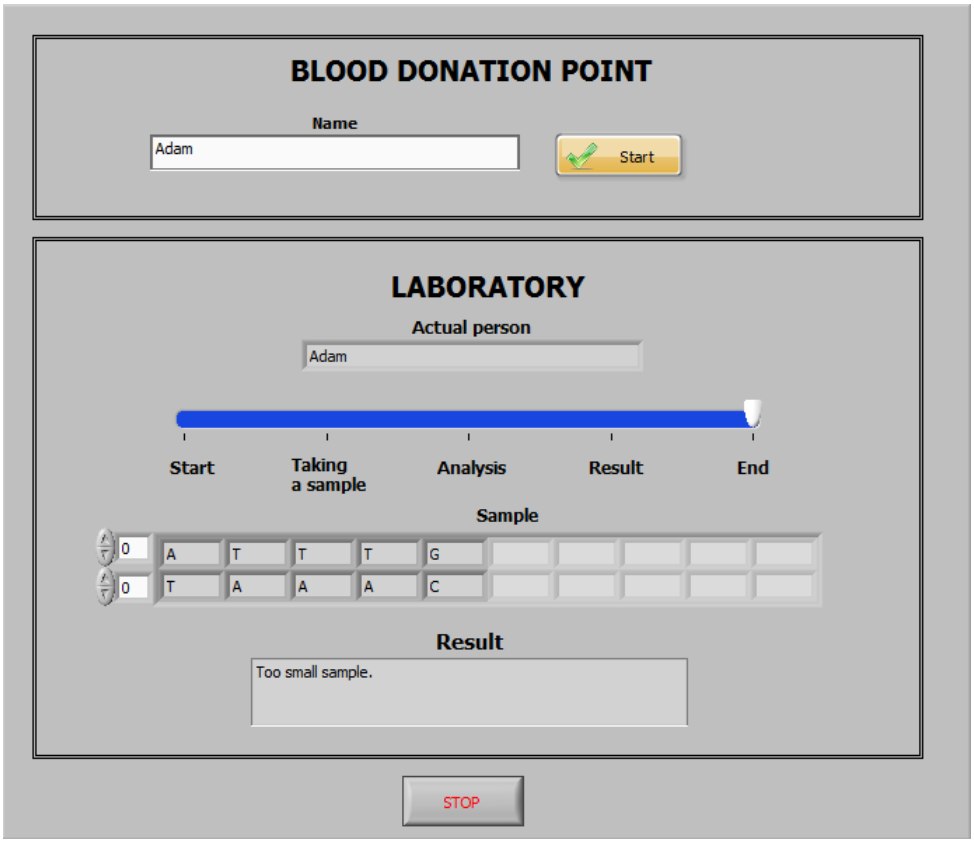


Rysunek 1: The *front panel* of DNA analyzer.

2. *Analysis* - the DNA chains are checked if the all pairs are connected according with scheme A-T and G-C.
3. *Result* - the appropiate comments are displayed e.g. "Too small sample", "Anomally detected", "DNA chains are correct".

- **Additional functionality:** Add the third MHL, in which commands about the status of analysis of the DNA sample will be saved e.g:
  Karol Nowak: sample has been taken.
  Karol Nowak: sample has been analysed.
  Karol Nowak: DNA chains are correct.
  Karol Nowak: analysis has been finished.
  The first MHL should send instructions to second MHL using separate queue.

Please use the front panel from public folder.



Rysunek 2: The sample output of DNA analyzer.