

Computing

Final Matlab Project, Winter Semester 18/19

Option 3: Morse Sender and Receiver GUI

29 & 30.11.2018

Please read through this Project description thoroughly and follow the formal and substantial requirements faithfully. It is relevant to your grade.

Create a Graphical User Interface (GUI) using either Matlab's GUI designer or App designer which is used to send and receive Morse audio signals.

Your task is to design and realize a GUI with which you can both enter a message which is converted into Morse code (international standard) and played back as well as detect a Morse audio signal and convert it back into a text message.

In Morse code, the Morse characters "Dit" and "Dah" exist. Each upper-case letter of the English alphabet, the digits 0 to 9, and a range of extra characters can be encoded (as shown in [this link](#)) using one or a combination of these two Morse characters. A "Dit" has the length of one unit, where the unit corresponds to a length of time which is left up to the operators. A "Dah" is three units, the space between characters within a single letter is one unit, the spaces between letters within a word are three units, and the space between a word is seven units.

The Morse sender and receiver GUI should at the very least have the following functionalities:

- Contain a function to translate text messages into Morse code and vice-versa.
- Within the GUI, be able to switch between sender and receiver mode.
- Each message which is sent, should begin with the word "DRILL".
- The default transmission rate of your Morse sender and receiver should be 20 *Morse words* per minute, where one *Morse word* is defined as 5 characters (e.g. "DRILL" = dah-di-dit di-dah-dit di-dit di-dah-di-dit di-dah-di-dit).
- Display the sent and received Morse code in a convenient representation (to test this you may need two computers).
- Display the sent and recorded signals in respective plots.

- The GUI should be able to save received messages in a `.txt` file and load the saved or custom preset messages (at least one test message should be uploaded with the rest of the project, for testing).

Once you have finished the project, play around with the transmission rate and document the resulting error-rate. Can you think of any ways to reduce the error-rate?

Some suggestions for further features are:

- The transmission rate could be variable for both the sender and receiver.
- Within the plot where the recorded signal is shown, also display the detected Morse message.
- Message encoding and decoding.
- The receiver could be configured so that it is able to detect the length of "Dits", "Dahs", and spaces to be able to automatically adjust to the transmission rate of the Morse code.

Formal Requirements

- *The deadline to choose your group project will be Thursday the 20th of December 2018, 4 pm. If you do not choose a project by this day, you will drop out of the course. You will not get a grade for the lecture Introduction to Matlab, but also not a failed. If you select a project by this deadline, it is equivalent to registering for the exam in Matlab. If you then do not submit a programming project or do not participate in the colloquium, the course will be considered as failed.*
- *But you can change to some other project until the 7th of January 2019, 4 pm. After this deadline changing of the project is no longer possible.*
- *The deadline to upload your group project to StudIP will be the 17th of January 2019, 8 pm! **Absolutely no upload after the deadline will be accepted, regardless of the reason.***
- *Hint: You can upload earlier versions of the project work to StudIP. Your latest upload will be taken as your contribution. In this way, you have handed in your project work even if there is any problem at the end of the upload deadline.*
- *Please make sure that your Matlab code is executable under the Matlab version used in the course and that it works correctly. This course uses Matlab version 2017b.*
- *Upload all files which are relevant to the project (.m file, .fig file, .mat file, data files, example files, the exported GUI file if a GUI was created etc.) and the published code to your group's Stud.IP folder. The compression of the data as *.zip file is desired.*
- *Follow the naming convention as they are provided in the Lecture Slides for the very first lecture.*
- *A good documentation (**published code and comments**) are essential for a good grade.*
- *Include a reference to all sources used in this project, where 3 or more lines of code were used (e.g. in the published code)*

Lecturer:
Markus Schellenberg

Matlab Course
WiSe18/19

Tutor:
Klaus Brümnn
Wiebke Middelberg

- *Plagiarism will be taken very seriously and appropriate action will be taken if it is discovered that code was copied without referencing or if code was copied from another group.*
- *There will be a oral colloquium about your programming work and MATLAB essentials at the end of the lecture period. Four dates have been reserved for the colloquium. Please check your emails to find out when your colloquium will take place.*
- *The best grades will be provided for creative solutions, elegant programming, fast code, and extra features which go beyond the listed minimum project specifications.*

Have fun and good luck!