Vision and Scope Document

for

ENR-261 Final Project

Arduino Morse Code Translator

Version 1.1 approved

Prepared by Geoff Berl

May 11, 2023

Revision History

Name	Date	Reason For Changes	Version
Initial Creation	4/17/2018	Initial creation	1.0
2023 Update	4/6/2023	Updated dates and fixed typos	1.1
Typo in pauses	5/11/2023	Fixing typo in pause requirements	1.2

Table of Contents

1. Introduction

The goal of the ENR-261 Final Project is to provide a graphical user interface that displays messages that are received in morse code. Given an arduino that is receiving a digital signal input on one of its pins, a MATLAB program will be developed to read the state of that Arduino pin and translate morse code transmissions into readable text.

This software requirements specification details a brief overview of the product, high priority features and requirements with associated use cases, non-functional requirements, external interface interactions, and design and implementation constraints. As such, this document is intended for developers of the product and their product managers, as well as potential documentation writers for the ENR-261 Final Project.

1.1 Document Conventions

The first requirement is of the highest importance, with the rest having decreasing importance. Unless otherwise noted, no other typographical conventions within this document should imply any priority or other meaning.

1.2 Project Scope

For the initial release of the ENR-261 Final Project, the main focus of the scope is to handle the main use case of a user connecting to an Arduino, receiving a morse code transmission via an Arduino pin and reading that transmission in character form on the graphical user interface. Later releases will add more dynamic capabilities for robustness. These dynamic capabilities may be to be able to analyze the text and display information such as the number of vowels and consonants and the number of words.

1.3 References

Reference ID	Reference Name	Title	Description
1	Help Reference	“ENR-261 Project Help Reference”	Explains details and contains examples using MATLAB instructions to communicate with Arduino. Also contains MATLAB examples and instructions for manipulating GUIs.

2. Overall Description

2.1 Product Perspective

The system is a new, self-contained product which shall translate a digital signal received by an Arduino pin into a character representation of the message transmitted.

2.2 User Classes and Characteristics

End User
The end user will have full rights to the data, there are no distinguished roles or limitations based on privilege levels.

2.3 Operating Environment

The system will be in a localized environment on a PC running Windows operating system (OS) with MATLAB R2017b or greater. There is currently no expectation of concurrent use.

2.4 Design and Implementation Constraints

The system shall
- Run on a PC with Windows 10 or later
- Utilize MATLAB R2022b or greater
    - Accompanied with the MATLAB support package for Arduino hardware add-on

2.5 Assumptions and Dependencies

It is assumed that there is a single message repeatedly transmitted on an infinite loop. It is also assumed that the message will lead with a special character and end with a special character, each distinct, so as to identify the beginning and end of a message transmission.

3. System Features

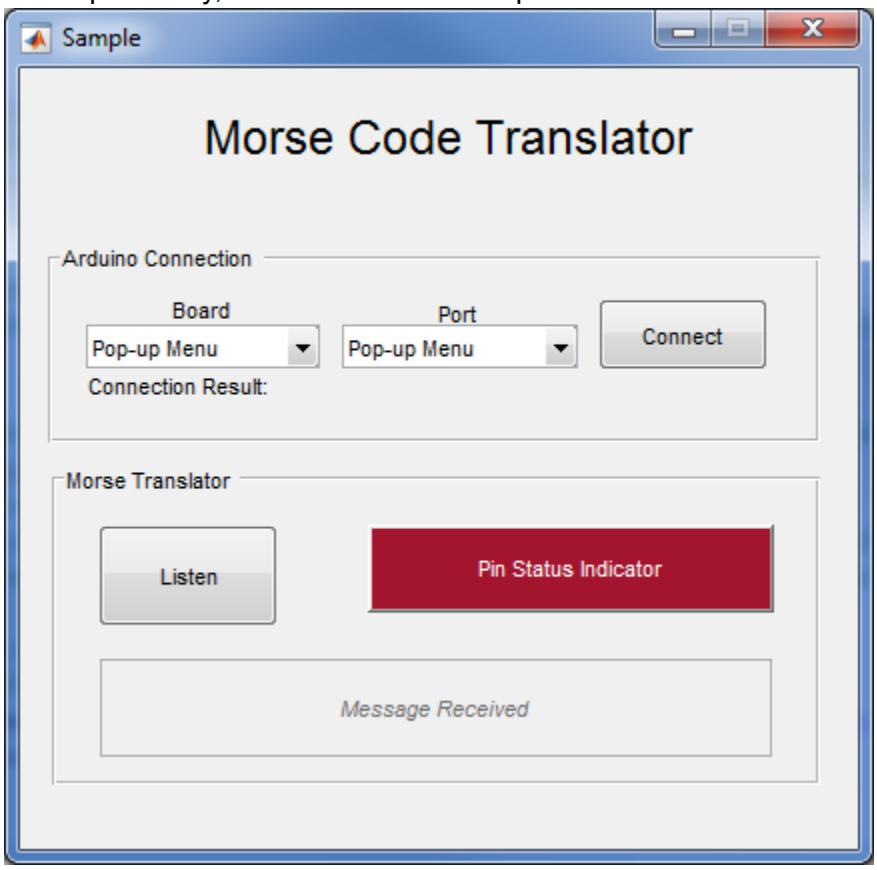
- The system shall successfully translate morse code messages
    - All morse transmissions will start with ‘KA’ (-.-.-) and end with ‘AR’ (.-.-.)
    - A dot is defined by a high signal received for a period of 100 milliseconds
    - A dash is defined by a high signal received for a period of a dot length times three
    - A pause between tones is defined by a lo signal received for a period of a dot length
    - A pause between characters is defined by a high signal received for a period of a dot length times three
    - A pause between words is defined by a high signal received for a period of a dot length times seven
- The system shall listen for pin high and low levels on digital pin nine
- The system shall import board types from a text file delimited by new lines
- The system shall use a reference of morse code characters based on a supplied comma separated values text file
- The system shall start listening when a GUI button is pressed
    - The system shall only display characters received after the character identifying the start of a transmission (KA)
    - The system shall end the listening event upon receiving the character identifying the end of a transmission (AR)
- The system shall handle failures in transmission
    - Further attempts to translate the transmission may need to be made, see *Other Software Quality Requirements* for details.

- If further attempts to translate are to be made, the system shall continue to attempt translating a transmission either until a message has encountered zero errors in transmission or until the attempt limit has been reached as met by the quality requirements.

4. External Interface Requirements

4.1 User Interfaces

The user interface will be a graphical user interface developed in the MATLAB environment. More specifically, the GUI will be developed with the aid of MATLAB's APPDESIGNER tool.



The Graphical User Interface shall

- Have a Drop Down List to select the type of Arduino board
  - A list of available boards will be available in line delimited form
- Have a Drop Down List to select from a list of available ports
  - The available ports shall be dependent on the available ports for a particular PC
- Have a button to connect to the arduino with the given board and port selections
  - The button shall handle the following cases
    - A board is not selected and a port is selected
    - A port is not selected and a board is selected
    - Neither a board or port is selected
    - A connection cannot be made due to error
    - A connection cannot be made due to warning
- Have a button to initiate "listening" for a morse code transmission

    - Upon activating the button, the previous message received shall be cleared from any display.
- Have a disabled text field for programmatically displaying a message that has been received.
- Handle all errors in an elegant manner, if an error occurs, it should not output to the Command Window, some indicator of the error (specific or generic) shall be displayed on the GUI for the user to take appropriate action to resolve the issue.

The Graphical User Interface may

- Have a component to indicate the pin status
- Have additional components as seen fit as an aid for the developer or to improve the user experience.
- Disable or Enable components, other than those previously specified, as seen fit to improve the user experience.

4.2 Hardware Interfaces

As this is an application in need of communicating with an Arduino microcontroller, there is a need for a USB hardware interface utilizing a cable with a USB A to a USB Mini B connection.

4.3 Software Interfaces

The client for this system should run on all non-deprecated major editions of Microsoft Windows (ie. Windows 7 or later). The software will be implemented in MATLAB R2022b or later.

4.4 Communications Interfaces

The tool shall communicate with an Arduino microcontroller via the hardware interface as mentioned in *Hardware Interfaces 3.2*.

5. Quality Attributes

5.1 Performance

The application shall take no longer than the length of time for one message transmission, times three. This shall give the tool enough time to translate the message successfully at least one out of three times allowing for up to 66% failure in message transmission.

5.2 Security Requirements

There is no assumed security with this application.

5.3 Safety Requirements

TBD

5.4 Usability Requirements

One of the main goals of this tool is to allow for translating morse code into a readable text output for the end user. Provided there are no software environmental issues outside of the tool itself, there should be no interaction by the user until the transmission has successfully completed.

5.5 Other Software Quality Requirements

The system shall function properly >=66% successful translations after initial release with future iterations improving this metric.

The system shall handle errors and attempt, to the best of the ability of the developer, to recover from such errors. Errors such as supplying improper information regarding the board or port for a connection.

Errors in transmission are not required to be identified and may be silently handled however it would be beneficial to include some type of indicator to the user to identify that the transmission has failed.

6. Other Requirements

- TBD

Appendix A: Glossary

TERM:	**DEFINITION:**
GUI	Graphical User Interface
OS	Operating System
MATLAB	Matrix Laboratory, an application for developing software
PC	Personal Computer
Algorithm	A mathematical software solution to a problem
USB	Universal Serial Bus

Appendix B: Analysis Models

TBD

Appendix C: Issues List

TBD