Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

DOKUMENTATION
AUTOMATISIERTE MESSTECHNIK

Morse-Code-Detektor mit Hardware-Tasten-Abfrage

Informations- und Kommunikationstechnik
Sommersemester 2020

Carlo Biermann
S0544023

Christian Böttcher
S0555692

Rainer Sitz
S0555614

Datum: 23.07.2020

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die Semesteraufgabe im Modul Automatisierte Messtechnik ist die Entwicklung eines Morse-Code Detektor Entwurf und dessen Umsetzung im Programm LabView. Morsecode ist eine Telegraphietechnik zur Übermittlung von Buchstaben, Zeichen und Ziffern. Theoretisch kann der Code als Ton, Funk oder Pulssignal übertragen werden.

Zur Darstellung des Morsecode werden Punkt (.) als kurzes Signal und der Strich (-) als langes Signal verwendet. Zusätzlich gibt es einen längeren Ton um die Pause zwischen zwei verschiedenen Wörtern darzustellen.

Die Grundidee unseres Projekts besteht darin, Morsezeichnen via Taster einzugeben und die Umsetzung in Buchstaben und Ziffern mittels der Software zu realisieren. Als einleitende Grafik ist nebenstehend unser Frontpanel dargestellt, welches die verschiedenen Buchstaben und Ziffern beinhaltet visualisiert.

Im folgenden Kapitel wird die Funktionsweise und der Aufbau unseres Programms erläutert und ein Screenshot des kompletten Entwurfs befindet sich im Anhang.

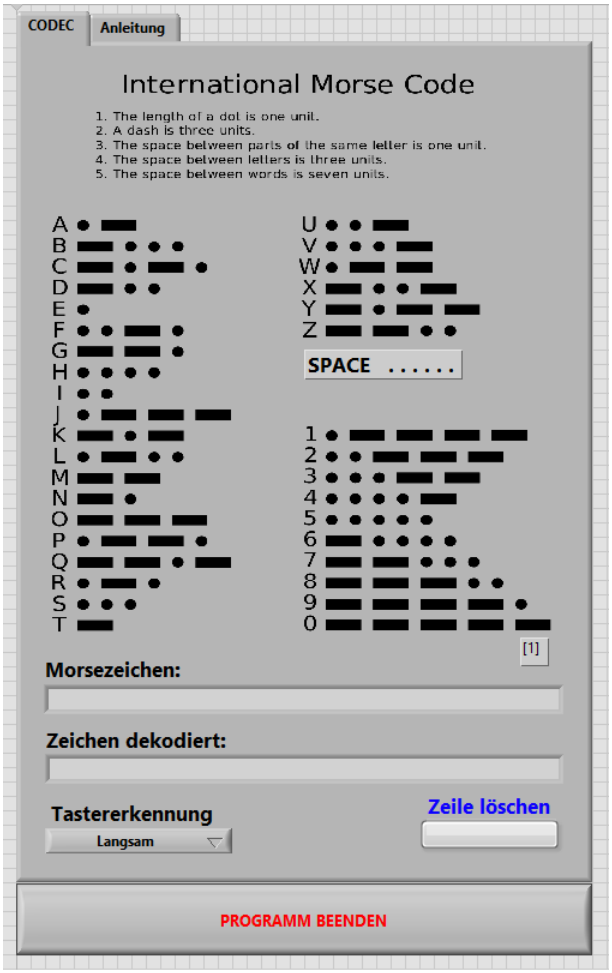

Abbildung 1.1: Frontpanel

2 Funktionalität

Der Morse-Code-Detektor wird, wie im Anhang dargestellt, durch zwei Hauptschleifen implementiert. In der oberen Schleife werden die Eingabesignale initialisiert, in *kurze* oder *lange* Signale eingeteilt und die verketteten Strings in die zweite Hauptschleife weitergeleitet. Dort werden die Zeichenblöcke mit den dort vordefinierten Zeichen abgeglichen und zugeordnet um den Morsecode auszugeben. Für die genauere Beschreibung werden folgend einzelne Teilabschnitte des Programms erläutert.

Bei einem Tastendruck wird zuerst eine True/False Abfrage durchgeführt. Das ist mittels eines Point-by-Point boolschen Übergang (als SubVI) realisiert und setzt die Zählerschleife für die Tastendruck ingang. Folgend werden die einzelnen Schleifendurchgänge aufsummiert und die Signale an die Tastenerkennung weitergeleitet.

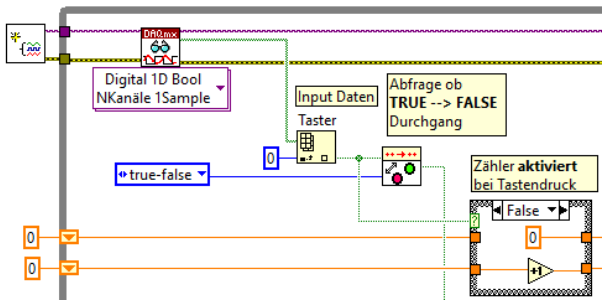

Abbildung 2.1: Tastenerkennung

Bei der Tastenerkennung haben wir drei verschiedene Zustände, für die Schnelligkeit der verschiedenen Signalvarianten vordefiniert.

Signal	schnell	mittel	langsam
kurz	25 ms	33 ms	38 ms
lang	63 ms	82 ms	95 ms
space	150 ms	195 ms	225 ms

Tabelle 2.1: Wertigkeiten der Tastenerkennung

Wie in folgender Abbildung zu sehen, wird die Initialisierung der Tastenerkennung über eine Case-Abfrage realisiert. Hierbei geht es um die Bestimmung der Genauigkeit der Tastererknnung. Nachdem nun die True/False Abfrage und die Tastererkennung durchlaufen ist, wird mit der Kombination der beiden Werte und unter Verwendung verschiedenen Vergleichsoperationen entschieden, ob das eingegebene Signal ein kurzes oder langes Signal ist und dann in die Zwischenspeicher geladen.

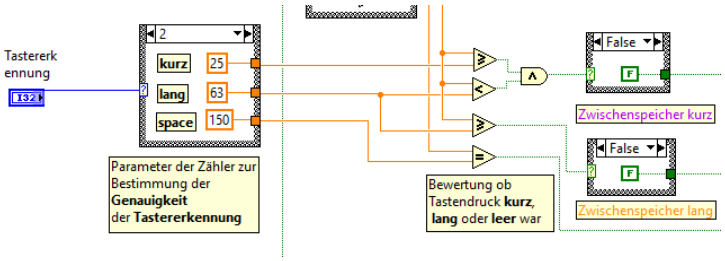

Abbildung 2.2: Genauigkeit der Tastererkennung

Nachfolgend werden wie in Abbildung 2.3 zu sehen, die verschiedenen Morsezeichen als Zeichenblöcke in eine Schieberegister gespeichert und an die zweite Hauptschleife übergeben. Es wird hierbei die auf True/False Wertigkeit der Durchgänge und die Zwischenspeicher für kurz/lang zurückgegriffen.

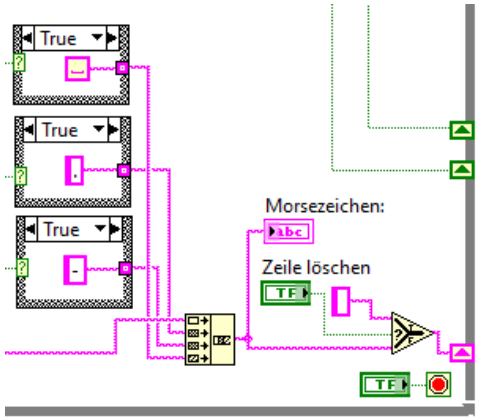

Abbildung 2.3: Verkettung der Morsezeichen

In der zweiten Hauptschleife wird aus den übergebenen Zeichenketten das Leerzeichen wieder entfernt, welches vorher für die Abtrennung der Signale benötigt wurde bevor die Zeichen mittels Case-Abfrage den jeweiligen Buchstaben/Zahlen zugewiesen werden. Danach folgt zuletzt die Übersetzte Ausgabe des Morse-Codes.

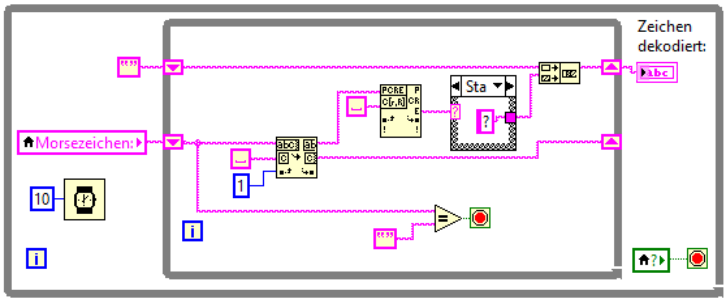

Abbildung 2.4: Übersetzung der Morsezeichen

2.1 Funktionalitätstest

Folgendes Video veranschaulicht die Funktinalität des Morse-Code Detektor:

3 Fazit

Grundsätzlich sind wir mit dem Ergebnis unseres Semesterprojekts zufrieden. Es hat sich gezeigt, dass die vorangegangenen Überlegungen für die Umsetzung bzw. Herangehensweise und Initialisierung der Grundidee mit wenigen Problemen durchführbar waren.

Zwischenzeitlich gab es kleinere Rückschläge mit Implementierungen von Teilabschnitten die zuerst nicht korrekt funktionierten. Wir mussten uns gegenseitig anpassen, da jeder Student verschiedene Hardwarekomponenten zuhause zur Verfügung hat. Jedoch wurden diese Problemstellungen durch gutes Zeitmanagement und kommunikativer Teamarbeit frühzeitig erkannt und behoben.

Anhang

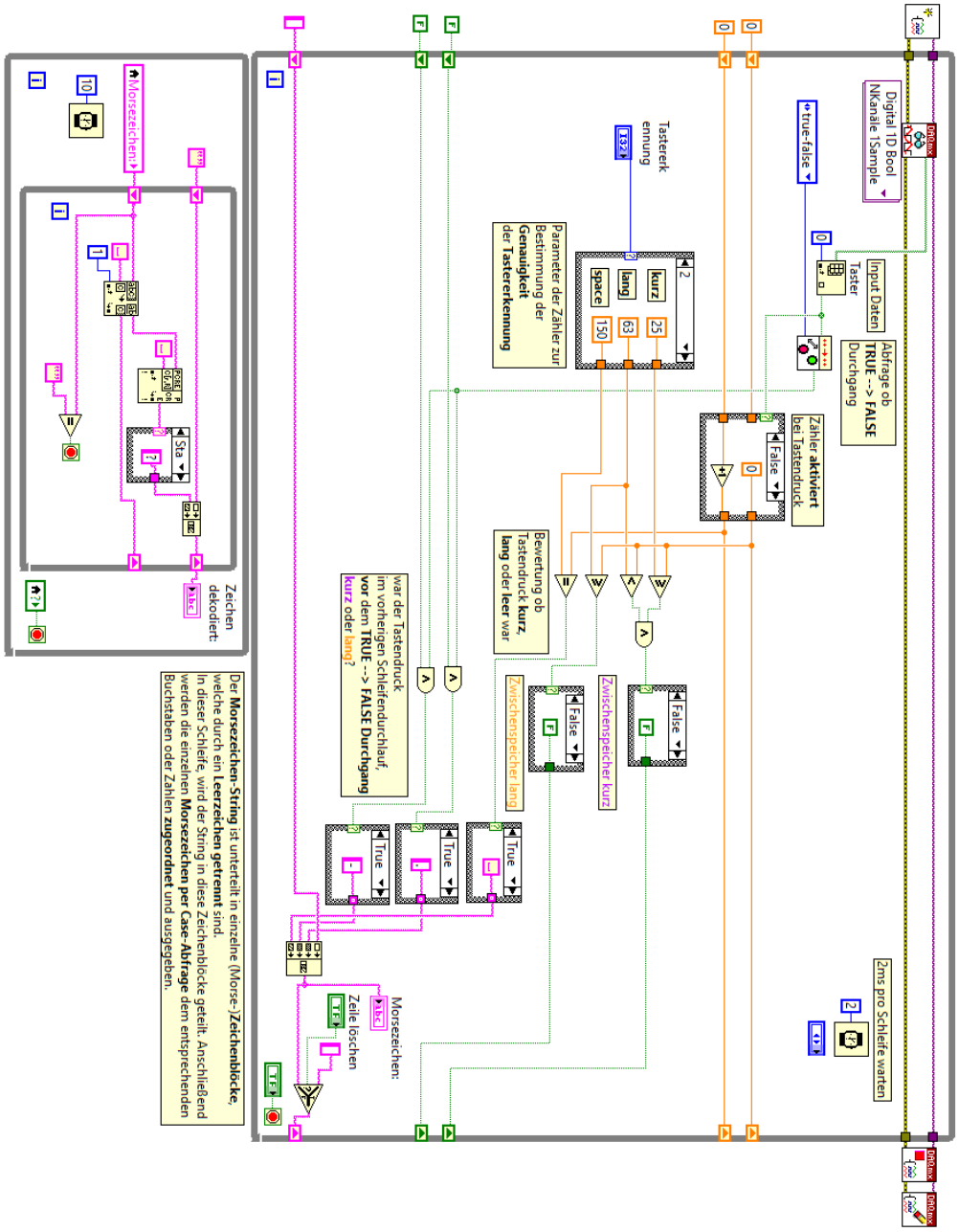

Abbildung 3.1: Blockschaltbild