PROJEKTDOKUMENTATION – CYBERPHYSISCHE SYSTEME

VERSION 1.0

Magnus Drolshagen

Inhaltsverzeichnis

Revisionsstände:

0.1	20.12.2023	Erste Version	Magnus Drolshagen	Erledigt

1. Einleitung

In diesem Dokument wird die Funktionsweise des Programms und dessen technische Umsetzung erläutert. Zur Voraussetzung, um das Programm ausführen zu können, bedarf es einem ESP32-Mikrocontroller, der auf das Carrier-Board gesteckt wird. Hier wird das Carrier-Board der Firma ASE-Schlierbach verwendet

Das Programm bietet die Funktion, Morsecode in Lateinische Buchstaben zu übersetzen.

Eine Morsecodetabelle, ist in den Anlagen zu finden.

2. Funktion und Bedienung

Es können Morsezeichen mithilfe von zwei Tastern auf dem Carrier-Board eingegeben werden. Mit dem Taster S4 wird ein Strich eingegeben (dargestellt als Minuszeichen) und mit der Taster S3 werden Punkte, bzw. kurze Signale eingegeben (dargestellt als Punkt). Die eingegebenen Zeichen werden in der unteren Zeile des LCDs angezeigt und in Echtzeit in der oberen Zeile in einen lateinischen Buchstaben übersetzt. Dabei entspricht die komplette untere Zeile dem hintersten Buchstaben in der obersten Zeile.

Eine Veranschaulichung hierzu ist in der Abbildung 2 zu sehen.

Ist ein Buchstabe fertig eingegeben, drückt man auf den Taster S2. Damit springt der Cursor in der oberen Zeile eine Stelle weiter und die untere Zeil wird geleert. Nun kann der nächste Buchstabe eingegeben werden. Ein Leerzeichen wird folglich durch ein erneutes Drücken des Tasters S2 eingegeben. Falls es für eine eingegebene Zeichenfolge keinen entsprechenden lateinischen Buchstaben gibt, wird an dieser Stelle eine Raute eingesetzt.

Um das eingegebene Wort zu löschen, wird der Taster „Reset“ betätigt, welcher recht unten auf dem Board zu finden ist. Das Wort und die eingegebenen Morsezeichen werden dann gelöscht.

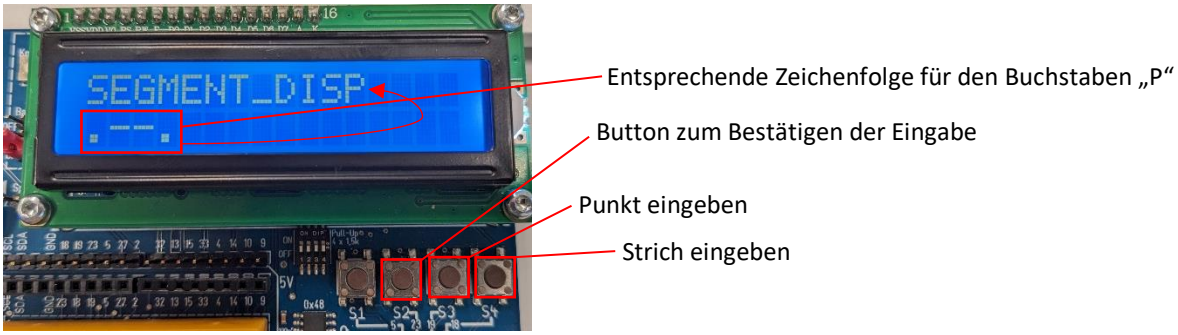

Abbildung 1

3. Technische Umsetzung

Folgende Tabelle gibt einen Überblick über die Variablen und dessen Funktion:

Name der Variable	Datentyp	Funktion/Verwendung der Variable
buttonPin	Integer Konstante	Pin des Tasters S4
linePin	Integer Konstante	Pin des Tasters S3
setPin	Integer Konstante	Pin des Tasters S2
morseType	Integer	Stelle die sagt, wie viele Morsezeichen eingegeben wurden
textType	Integer	Stelle die sagt, wie viele Buchstaben eingegeben wurden
BtnStart	Integer	Zu welcher Zeit (nach millis()) ein Button gedrückt wurde
BtnEnd	Integer	Zu welcher Zeit (nach millis()) ein Button losgelassen wurde

BtnDauer	Integer	Wie lange ein Button gedrückt wurde (in Millisekunden)
morseCode	String	Eingegebene Morsezeichen
currentCharacter	String	Der entsprechende Buchstabe für morseCode
plainText	String	Das vollständig eingegebene Wort
pressedB	Boolean	Wird in dem Durchlauf des Programms true, nachdem der Button buttonPin losgelassen wurde (wird verwendet, um Prellungen zu vermeiden)
pressedA	Boolean	Wird in dem Durchlauf des Programms true, nachdem der Button linePin losgelassen wurde (wird verwendet, um Prellungen zu vermeiden)
pressedC	Boolean	Wird in dem Durchlauf des Programms true, nachdem der Button setPin losgelassen wurde (wird verwendet, um Prellungen zu vermeiden)
translateArray	String Array	Dieses Array enthält die Übersetzung von Morsezeichen in Buchstaben

Folgend werden die zusätzlichen Methoden (abgesehen von setup und loop) erläutert:

handleInput()	Hier werden die Eingaben der Buttons S2 – S4 verarbeitet. Für die Verarbeitung von Eingaben werden pro Button zwei if-Schleifen verwendet. Die erste Schleife wird ausgeführt, wenn der entsprechende Button gedrückt wurde und der buttonspezifische Boolean pressed… false ist. Diese Überprüfung nach dem Boolean ist nötig, um Fehleingaben zu vermeiden, welche auftreten könnten, wenn der Button zu lange gehalten oder durch Prellungen des Buttons. Bei Ausführen der ersten Schleife, wird BtnStart auf millis() gesetzt. Pressed... wird auf true gesetzt um festzulegen, dass sich der Button gerade gedrückt wird. Danach wird eine Pause gemacht, um Prellungen zu vermeiden. Danach wird die zweite Schleife pro Button ausgeführt. Jedoch nur wenn der Button wieder losgelassen wurde und Pressed... auf true ist – der Button also im letzten Durchlauf des Programms gedrückt wurde. BtnEnd wird auf millis() gesetzt und mithilfe von, dem in der ersten Schleife gesetzten

	BtnStart, die Dauer ausgerechnet, die der Button gedrückt wurde. Wenn die Dauer größer als 50ms ist, wird die Verarbeitung weiter ausgeführt. Je nachdem welcher Button gedrückt wurde, wird entsprechend fortgefahren.
Ausgabe()	Hier werden die Daten auf dem LCD passend ausgegeben.
Translate()	Hier wird die eingegebene Morse-Zeichenfolge in Buchstaben übersetzt. Dies wird durch ein Array, realisiert, welches durchlaufen wird, bis die passende Zeichenfolge gefunden wurde. Dies ist einfacher erweiterbar als eine große if-Abfrage oder eine switch-Schleifen. Wenn keine passende Zeichenfolge gefunden wurde, wird eine Raute ausgegebeb.

4. Anlagen