

Masterarbeit

TRC²

TAPE AND REEL CONTROLLER CUSTOMIZATION

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

Marcell Bergner, BSc.

1910322017

betreut und begutachtet von
FH-Prof. DI Dieter Lutzmayr

Graz, im November 2020

A handwritten signature in blue ink, reading "Marcell Bergner", written over a horizontal dotted line.

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.



Unterschrift

DANKSAGUNG

An dieser Stelle möchte ich mich bei meiner Familie und meiner Partnerin bedanken, die mich während der Anfertigung dieser Arbeit unterstützt und motiviert haben. Ein besonderer Dank gilt auch meinem Arbeitgeber Flex, im Speziellen meinem Vorgesetzten und Flex-Projektbetreuer Herrn Christian Mayer. Um mich bestmöglich auf mein Studium konzentrieren zu können, wurde eine Reduktion meiner Arbeitszeit durch Inanspruchnahme der Bildungsteilzeit ermöglicht. Außerdem wurden mir zeitliche als auch finanzielle Mittel für die Umsetzung des Projektes eingeräumt. Ein weiterer Dank gilt meinem FH-Projektbetreuer Herrn FH-Prof. DI Dieter Lutzmayr, der mich tatkräftig unterstützt hat.

KURZFASSUNG

Flex produziert kleine elektronische Module, die am Ende des Produktionsflusses getestet und verpackt werden. Dazu sind selbst entwickelte Testlinien im Einsatz, bei welchen zugekaufte Verpackungsmaschinen integriert sind. Neu entwickelte Produktionsmaschinen führen zu veränderten Anforderungen an die integrierte Verpackungsmaschine. Fehlende Kommunikation schränkt die Möglichkeiten bei der Entwicklung ein. Zusätzlich führt eine unpraktische Ablaufsteuerung beim Ansprechen von Sensoren zu deren Deaktivierung. Ziel des Projekts ist es die bestehende Steuereinheit durch einen neuen Mikrocontroller und ein speziell entwickeltes Controller-Board zu ersetzen.

Zunächst werden bekannte Mikrocontroller recherchiert und deren Eigenschaften verglichen. Für die Evaluierung werden die grundlegenden Anforderungen an den Mikrocontroller durch eine Analyse der Verpackungsmaschine ermittelt. Mit den Ergebnissen wird ein Mikrocontroller ausgewählt und ein Prototyp des Controller-Boards angefertigt. Auf dessen Basis wird ein verbesserter Schaltplan erstellt und ein Controller-Board designet und angefertigt. Zur Überprüfung der neuen Steuereinheit werden die Funktionen zur Ansteuerung der Komponenten ausprogrammiert.

Die modifizierte Steuereinheit ermöglicht individuelle Verpackungsabläufe mit unterschiedlichen Möglichkeiten der Steuerung des Verpackungsprozesses, sowie der Materialüberwachung. Zukünftige Anforderungen können aufgrund der gewonnenen Flexibilität einfach umgesetzt werden.

ABSTRACT

The company Flex produces small electronic modules that are tested and packaged at the end of the production flow. For this purpose, developed in-house test lines are utilized, in which purchased packaging machines are integrated. Newly developed production machines lead to changed requirements on the integrated packaging machine. A lack of communication restricts possibilities of new development. Additionally, impractical sequence control of irregularity detection by sensors results in their deactivation. The aim of the master's thesis is to replace the existing control unit by a new microcontroller and a specifically developed controller board.

First, known microcontrollers are investigated and their characteristics are compared. For the evaluation, the basic requirements for the microcontroller are examined by analysing the packaging machine. With the obtained results a microcontroller is selected and a prototype for the controller board is constructed. Based on the prototype, an improved circuit diagram is created and a new controller board is designed and manufactured. The functions are programmed in order to control the components of the packaging machine for their verification of the new control unit.

The modified control unit enables individually customized packaging processes with different possibilities of controlling or material monitoring. Future requirements can be easily implemented due to the gained flexibility.

INHALTSVERZEICHNIS

1	Einleitung.....	1
1.1	Flex Global.....	1
1.2	Flex Althofen.....	1
1.3	Projekteinführung.....	2
1.4	Projektumfang.....	2
2	Analyse des Gerätes.....	3
2.1	Ausgangssituation.....	4
2.2	Optimierungspotenzial.....	7
3	Reverse Engineering.....	8
3.1	Einsatzbereiche.....	9
3.2	Rechtliche Aspekte.....	10
4	Controllertypen.....	12
5	Controller Evaluierung.....	29
5.1	Hardwareanalyse Verpackungseinheit.....	29
5.2	Anforderungen an den Controller.....	35
6	Hardware-Aufbau.....	38
6.1	Versuchsaufbau.....	38
6.2	Platinendesign.....	45
7	Software-Implementierung.....	51
7.1	Display Anzeige und Tastenfeld.....	52
7.2	Ansteuerung Schrittmotor.....	55
7.3	Serielle Kommunikation.....	57
7.4	I/O Steuerung.....	58
8	Ergebnisse und Ausblick.....	61
	Literaturverzeichnis.....	62
	Abbildungsverzeichnis.....	64
	Tabellenverzeichnis.....	67
	Abkürzungsverzeichnis.....	68
	Anhang 1: Schaltplan des Controller-Boards.....	69
	Anhang 2: Auszüge Datenblätter.....	70

1 EINLEITUNG

1.1 Flex Global

Flex ist einer der weltweit führenden Anbieter von Electronic Manufacturing Services (EMS) mit über 100 Standorten in 30 verschiedenen Ländern. Der Firmensitz befindet sich in Singapur und eine weitere Verwaltungszentrale in San José, Kalifornien. Weltweit werden um die 200.000 Mitarbeiter beschäftigt und ein Jahresumsatz von 24 Milliarden Dollar generiert. Flex beliefert 1.000 Kunden und führt über 12.500 neue Produkte pro Jahr ein.

Der Firmenleitzatz lautet „Sketch to Scale“ und beschreibt das Angebot für Services von Konzeptentwicklung, Design, Prototyping und Produktion bis hin zu Vertrieb und End-of-Life-Management.

1.2 Flex Althofen

Der Standort Althofen wurde 1970 als Philips Niederlassung gegründet und ist seit 1997 Teil des Flex Konzerns. Heute liegt der Fokus des Betriebs auf Elektronikproduktion im Medical, Industrial & Emerging Industries und Automotive Bereiche. Um hier tätig sein zu können, ist Flex Althofen nicht nur ISO 9001 zertifiziert, sondern deckt zusätzlich den Automotive-Bereich mit Zertifizierungen im ISO IATF 16949 und den Medizinbereich mit ISO 13485 ab.

Mit heutigem Stand beschäftigt der Betrieb 950 Mitarbeiter, die sich auf die Bereiche Produktion, Engineering Services, Customer Services und Entwicklung (mechanisch, elektrisch) aufteilen.

Flex Althofen ist auch eines von drei Product Innovation Center (PIC) im globalen Flex Konzern. Das PIC Althofen arbeitet mit Unternehmen jeder Größe zusammen, von kleinen Start-ups bis hin zu großen etablierten Branchenführern.



Abb. 1.1: Flex Standort Althofen, Quelle: Eigene Darstellung

Das PIC Althofen verfügt über umfangreiche Fähigkeiten, die über die traditionellen EMS-Dienstleistungen hinausgehen. Dazu gehören fortschrittliches Engineering, Design für Fertigung und Lieferkettenoptimierung, Prototyping, Fehleranalyse, Tests, Produktion und Transfer. PICs unterstützen Kundenprodukte in einem frühen Stadium des Lebenszyklus vom Konzept bis zur Realität und mit End-to-End-Lösungen über die gesamte Lieferkette hinweg.

1.3 Projekteinführung

Im Produktionsbereich der Firma Flex in Althofen sind automatisierte Prozesse für den Aufbau und Test elektrischer Module rund um die Uhr (24/7) im Einsatz. In der Abteilung „Test Development“ werden die Produktionsmaschinen für den Testbereich entwickelt und gebaut. Dafür sind 42 Mitarbeiter mit unterschiedlichen Kompetenzen im Bereich Mechanik, Elektronik und Software tätig. Bei einer großvolumigen Produktparte (elektronische Kommunikations-Module), werden die Produkte am Ende der Fertigungskette vollautomatisiert funktionsgetestet und verpackt.

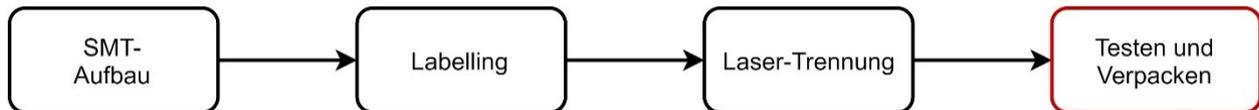


Abb. 1.2: Process Flow Flex, Quelle: Eigene Darstellung

Für den Prozess „Testen und Verpacken“ ist die Verpackungseinheit OEM TM-50 des amerikanischen Unternehmens V-TEK in Verwendung, die seit mehr als 30 Jahren nahezu unverändert am Markt erhältlich ist. Der Hersteller, mit Standorten in Minnesota, Kalifornien und Mexiko, hat sich durch die hohe Zuverlässigkeit der Produkte über drei Jahrzehnte als führender Hersteller für Tape and Reel-Systeme herauskristallisiert.

Gesteigerte Möglichkeiten im Produktionsbereich von Flex werfen Wünsche über erweiterte Funktionen und Modifikationen des OEM TM-50 auf. Anfragen nach Modifikation des Funktionsumfangs werden vom Hersteller, mit Verweis auf anhaltenden Absatz des bestehenden Systems, zurückgewiesen. Die Durchführung der Änderungen durch externe Firmen ist aufgrund schwer vorhersehbarer zeitlicher und finanzieller Ressourcen riskant, daher wird das Projekt firmenintern durchgeführt.

1.4 Projektumfang

Diese Arbeit umfasst die Modifikation der Steuerung der OEM TM-50 Verpackungstation. Darin werden die Möglichkeiten zur Umsetzung beleuchtet. Dafür wird die Verpackungseinheit analysiert, um die Anforderungen an die Steuereinheit zu evaluieren. Mit den Erkenntnissen wird im Anschluss darauf ein Prototyp mit der gewählten Hardware aufgebaut, auf dessen Basis Bauteilwerte ermittelt und erste Ansteuerung eines LC-Displays durchgeführt werden. Auftretende Probleme werden kontinuierlich behoben, wodurch ein Schaltplan entsteht. Dadurch kann ein Controller-Board für die Verbindung der Verpackungskomponenten und der Beschaltung der Ein- und Ausgänge gelayoutet werden. Die Arbeit wird durch die Programmierung zur Ansteuerung einzelner Komponenten abgerundet und bereitet den Grundstein für flexible Programmierung der Abläufe. Durch das Projekt erhofft sich das Unternehmen Flex, ein für die speziellen Anforderungen verbessertes Verpackungssystem zu erschaffen. Es ist eine veränderte Ablaufsteuerung beim Ansprechen von Sensoren gewünscht. Des Weiteren kann durch Modifikation der Software verbesserter Informationsfluss zwischen dem Robotersystem und dem Verpackungssystem resultieren. Durch die Umsetzung des Projektes, soll das Know-How über die Verpackungstation in die eigene Firma geholt werden. Das ermöglicht einfache Modifikationen zur Anpassung an neu entwickelte Testsysteme.

2 ANALYSE DES GERÄTES

Das OEM TM-50 ist ein schmales Tape and Reel-Verpackungssystem für Pick and Place Anwendungen von kleinen Elektronikkomponenten bei Industrieanlagen.



Abb. 2.1: OEM TM-50, Quelle: V-TEK, Inc., Online-Quelle [14.September.2020]

Das Verpackungssystem führt das Gurtband von einer Einzugsrolle durch eine breitenverstellbare Schiene. Durch einen Schrittmotor sind die Taschen fixiert und werden nach dem Einlegen von elektrischen Modulen oder Bauteilen um einen voreingestellten Abstand weitergezogen.

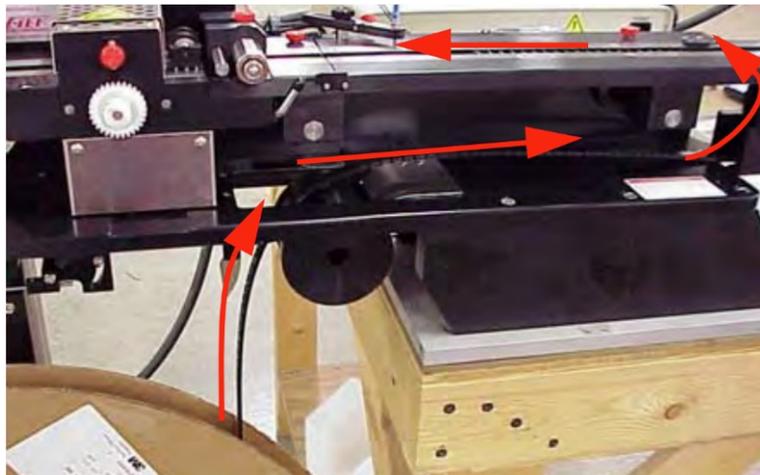
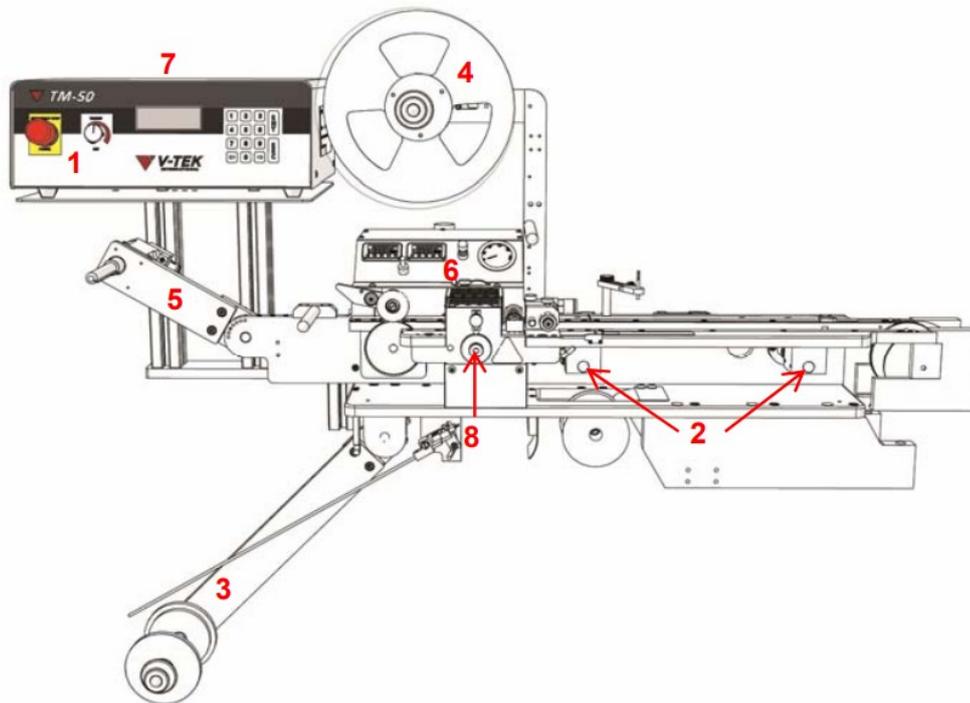


Abb. 2.2: OEM TM-50 Führung des Gurtbandes, Quelle: James (2012)

Im Bereich der Verschweißungseinheit wird das Deckband auf das Gurtband mittels erhitzter Kufen aufgeschweißt. Das Verpackungsmaterial mit den verschweißten Modulen wird auf einer passenden Rolle aufgewickelt. Wenn die Anzahl auf einer vollen Rolle erreicht ist, wird zusätzlich ein vordefinierter Abstand an leeren Taschen verschweißt, um eine Gurtspleißung beim Kunden zu ermöglichen. Danach kann das leer verpackte Material abgeschnitten werden, um mit einer neuen Rolle beginnen zu können.

In Abb. 2.3 ist die Verpackungseinheit mit den einzelnen Systemkomponenten dargestellt:



- 1 ... Hauptschalter
- 2 ... Führungsbolzen für Breitenverstellung
- 3 ... Halterung Gurtband
- 4 ... Halterung Deckband
- 5 ... Halterung fertig verpackte Rolle
- 6 ... Verschweißungssteuerung
- 7 ... Steuerkasten
- 8 ... Schweißkufen-Spurverstellung

Abb. 2.3: OEM TM-50 Systemkomponenten, Quelle: James (2012)

2.1 Ausgangssituation

Das OEM TM-50 wird seit mehr als 20 Jahren im Unternehmen Flex verwendet. Damit werden produzierte elektrische Module nach einem Funktionstest mit optionaler produktspezifischer Firmwareprogrammierung verpackt. Das Handling erfolgt über zwei hängend montierte Roboter, die mit Vakuum-Saugnapfen ausgestattet sind. Vor dem Test werden die chaotisch liegenden Module über Kamerasysteme erkannt und in einer Vorzentrierung gerichtet sortiert. Im Anschluss werden die gerichteten Module in den produktspezifischen Tester gehoben und getestet.

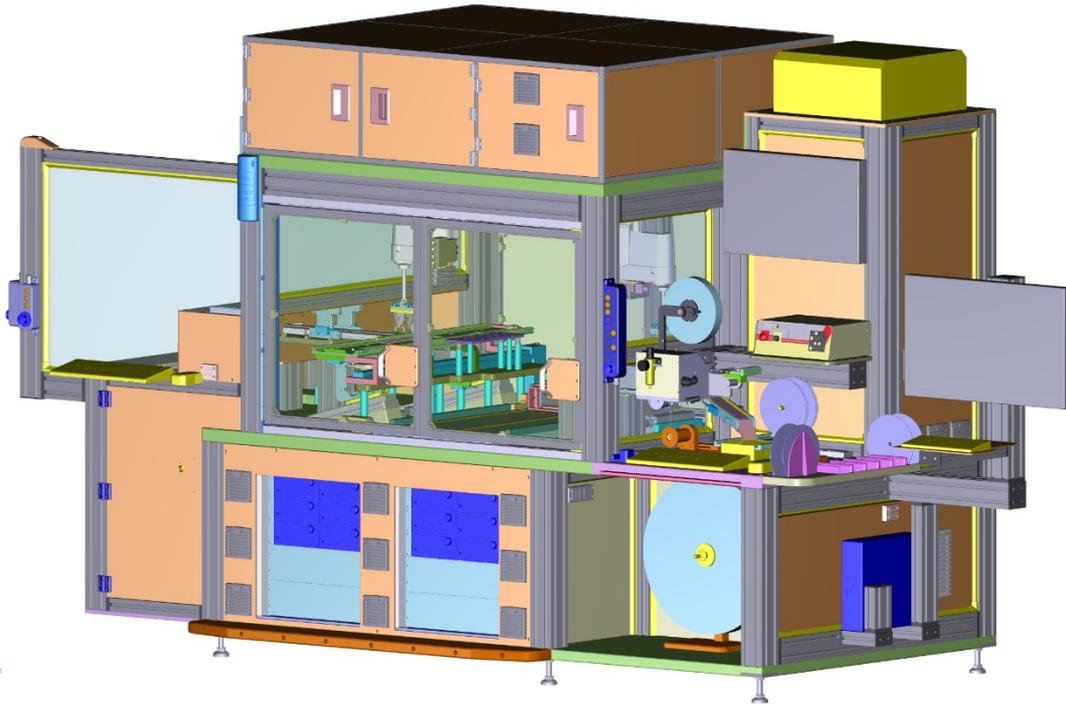


Abb. 2.4: Testlinie bei Flex, Quelle: Eigene Darstellung

Module, die den Test bestehen, werden vom Roboter gerichtet in leere Taschen des Gurtbandes eingelegt. Ein paar Taschen weiter wird über eine Kamera die Einlege-Richtung überprüft und der Status des Moduls im Shop Floor System aktualisiert. Zwischen jedem Einlegevorgang wird das Gurtband um einen modulgrößen-abhängigen Pitch (Abstand) weitergezogen. Der gesamte Vorgang wird über ein Controller-Board im Steuerkasten der Verpackungseinheit gesteuert.

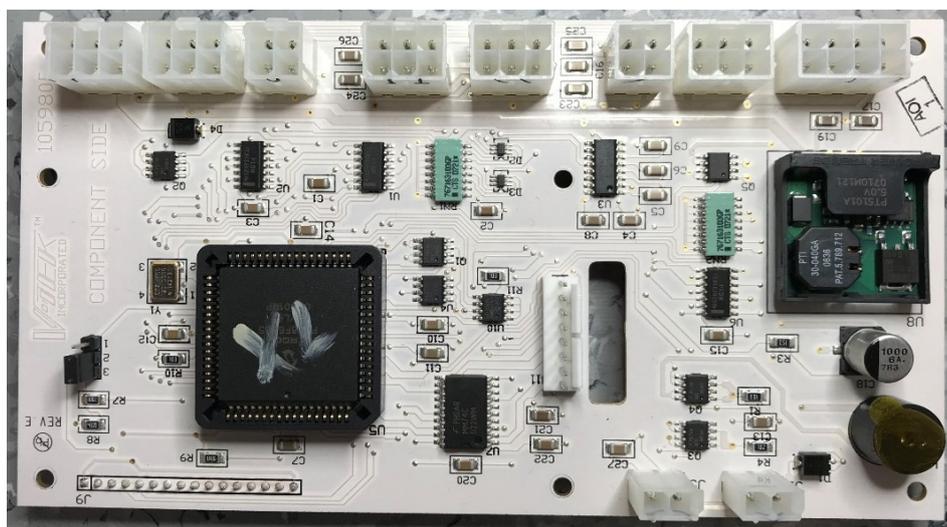


Abb. 2.5: Originales Controller-Board, Quelle: Eigene Darstellung

Der Trigger für den Vorschub kommt von einem digitalen Ausgang der Robotersteuerung, welcher über ein Relais angesteuert wird. Nach jedem Verpackungszyklus wird der Zählerstand um 1 erhöht. Die Anzeige des aktuellen Zählerstandes sowie weiterer Parameter erfolgt über ein LC-Display durch einen verglasten Ausschnitt im Steuerkasten.

Das Controller-Board verfügt über eine serielle Schnittstelle, über die Kommandos von der Robotersteuerung empfangen werden können. Bei den Kommandos handelt es sich um Job-Konfigurationen, die auf einem EEPROM abgespeichert sind. Davon können am bestehenden Mikrocontroller bis zu 64 verschiedene abgespeichert werden. Wird ein Kommando empfangen, werden die zuvor abgespeicherten Parameter eingestellt und am Display angezeigt.

Sensoren

Die OEM TM-50 hat 4 Sensoren (Lichtschranken) eingebaut, welche softwaremäßig deaktiviert werden können. Sie dienen der Überwachung des Verpackungsablaufs und der Materialüberwachung von Gurtband und Deckband.



Abb. 2.6: Sensor FS-V21, Quelle: Keyence, Online-Quelle [14.September.2020]

Wird eine Lichtschranke unterbrochen, verändert sich der Zustand eines digitalen Eingangs am Mikrocontroller und die Hardware stoppt sofort. Am Display wird eine spezifische Fehlermeldung angezeigt, die auf das Problem hinweist. Nach dem Beheben ebendieses kann der Fehler über die ESC-Taste am Tastenfeld quittiert werden.

Piezo-Signalgeber

Neben der visuellen Anzeige am Display ist ein Piezo-Signalgeber am Controller-Board verbaut. Beim Ansprechen eines Sensors, sowie als Feedback bei Eingaben über das Tastenfeld wird durch den Buzzer ein zusätzliches akustisches Signal gegeben.



Abb. 2.7: Piezo-Signalgeber, Quelle: RS Components, Online-Quelle [7.November.2020]

2.2 Optimierungspotenzial

In Abb. 2.8 ist die Kommunikation zwischen Robotersteuerung (PC) und Verpackungssteuerung (Mikrocontroller) dargestellt. Der Informationsaustausch reduziert sich auf Kommandos bei Produktumstellung für aktualisierte Job-Konfigurationen und Empfangsbestätigungen.



Abb. 2.8: Erweiterbarer Informationsaustausch, Quelle: Eigene Darstellung

Aus diesem Grund sind die auf der Verpackungseinheit befindlichen Informationen für die Robotersteuerung und das Shop Floor System unzugänglich. Ein konkretes Beispiel ist der Zählerstand der verpackten Module auf einer Rolle, der mit den eingelesenen Modulen im Shop Floor System übereinstimmen muss. Bei Verfügbarkeit durch Datenkommunikation kann der Zählerstand systematisch verglichen und die Fehlerquelle Mensch ausgeschlossen werden.

Für die Kontrolle des Verpackungsprozesses sind Einweg- und Reflexions-Lichtschranken in der OEM TM-50 verbaut. Spricht ein Sensor infolge einer Unregelmäßigkeit an, stoppt die Hardware sofort. Der Controller der Verpackungssteuerung schaltet daraufhin einen Ausgang, der von der Robotersteuerung als Indikator für ein Problem erkannt wird. Welcher Sensor angesprochen hat, beziehungsweise warum die Verpackungseinheit einen Fehler meldet, ist der Robotersteuerung unbekannt. Am Bildschirm wird eine allgemeine Fehlermeldung angezeigt und ein Alarmsignal der Testlinie macht darauf aufmerksam.

Für die Fortsetzung der Produktion ist ein Techniker notwendig, der im Bedarfsfall manuelle Schritte und Kontrollen durchführen muss, um die Testlinie in den Normalbetrieb zu bringen.

Wegen des unpraktischen Stoppens der Verpackungseinheit wurden Sensoren deaktiviert, die für die Sicherstellung der Qualität der Verpackung nicht zwingend notwendig sind. Das hat zur Folge, dass die Materialüberwachung vom Operator der Testlinie durchgeführt werden muss. Es kommt daher zu menschlichen Fehlern, die verhindert werden können. Geht ein Material zur Neige und wird dieser Umstand vom Operator nicht frühzeitig erkannt, muss die angefangene Rolle wieder ausgepackt und die Module erneut getestet werden. Die Folge sind Kapazitätsverluste in Bezug auf die Auslastung der Testlinie, die sich in Form von Lieferverzögerungen auswirken können.

Eine modifizierte Software des Controllers kann die Verpackungseinheit beim Ansprechen eines Sensors den Zyklus fertig machen lassen und erst danach stoppen, wenn es überhaupt notwendig ist. Ein sensorabhängiger Prozess mit Fehlerbeschreibung am Bildschirm kann die Einbindung der Sensoren praktikabel machen.

Neben den genannten Vorteilen bietet eine eigens programmierte Steuerung Flexibilität für zukünftige Anforderungen mit zahlreichen Möglichkeiten.

3 REVERSE ENGINEERING

Reverse Engineering ist der Prozess des Extrahierens des Wissens über Designentwürfe aus allem, was von Menschenhand geschaffen wurde. Es wird in der Regel angewendet, um fehlendes Wissen, Ideen und Designphilosophie zu erhalten, wenn diese Informationen nicht verfügbar sind. In einigen Fällen sind die Informationen im Besitz von jemandem, der nicht bereit ist sie weiterzugeben. In anderen Fällen sind sie verloren gegangen oder zerstört worden.

Es gibt zwei Arten von Engineering, Forward Engineering und Reverse Engineering. Forward Engineering ist der traditionelle Prozess des Übergangs von Abstraktionen und logischen Entwürfen auf hoher Ebene zur physischen Implementierung eines Systems. In manchen Situationen kann es ein physisches Teil/Produkt ohne technische Details wie Zeichnungen, Materiallisten oder ohne technische Daten geben. Der Prozess des Duplizierens eines bestehenden Teils oder eines Produkts ohne Zeichnungen, Dokumentation oder ein Computermodell wird als Reverse Engineering bezeichnet. Es ist auch definiert als der Prozess der Gewinnung eines geometrischen CAD-Modells, die durch Scannen/Digitalisieren vorhandener Teile/Produkte erfasst werden. Reverse Engineering ist heute in zahlreichen Anwendungen weit verbreitet, wie in der Fertigung, im Industriedesign und in der Schmuckgestaltung und -reproduktion. Ein weiterer Grund für Reverse Engineering ist die Komprimierung der Produktentwicklungs-Zykluszeiten. Auf dem hart umkämpften globalen Markt suchen Hersteller ständig nach neuen Wegen, um die Vorlaufzeiten bis zur Markteinführung eines neuen Produkts zu verkürzen. Rapid Product Development (RPD) bezieht sich auf neu entwickelte Technologien und Techniken, die Hersteller und Konstrukteure dabei unterstützen, die Anforderungen an eine verkürzte Produktentwicklungszeit zu erfüllen.

Behauptungen zufolge wird in den meisten Branchen das Reverse Engineering zum Zweck der Entwicklung von Konkurrenzprodukten eingesetzt. Bemerkenswert dabei ist, dass die Popularität von Reverse Engineering in der Softwareindustrie nicht hoch ist. Dafür gibt es mehrere Gründe, aber vor allem weil Software so undurchsichtig ist, wird das Reverse Engineering für Wettbewerbszwecke in vielen Fällen als ein so komplexer Prozess angesehen, dass sich der finanzielle Aufwand nicht lohnt.¹

Dekompilierung

Ein Decompiler nimmt eine ausführbare Binärdatei und versucht daraus einen lesbaren Hochsprachen-Code zu erzeugen. Es wird die Idee verfolgt den Kompilierungsprozess umzukehren, um die ursprüngliche Quelldatei oder etwas ähnliches zu erhalten. Meist ist die Wiederherstellung des ursprünglichen Quellcodes nicht möglich, da in den meisten Sprachen wichtige Elemente während des Kompilierungsprozesses ausgelassen werden und unmöglich wiederherzustellen sind. Trotzdem sind Decompiler mächtige Werkzeuge, die in manchen Situationen und Umgebungen einen gut lesbaren Quellcode aus einem Programm rekonstruieren können.²

¹ Vgl. Eilam (2011), S. 1 f

² Vgl. Eilam (2011), S. 10

3.1 Einsatzbereiche

Für den Einsatz von Reverse Engineering kann es unter anderem folgende Gründe geben:

- Der ursprüngliche Hersteller existiert nicht mehr, aber ein Kunde benötigt das Produkt, zum Beispiel Flugzeugersatzteile, die typischerweise benötigt werden, nachdem ein Flugzeug mehrere Jahre in Betrieb war
- Der ursprüngliche Hersteller eines Produkts stellt das Produkt nicht mehr her, da dieses Produkt veraltet ist
- Die ursprüngliche Dokumentation zum Produktdesign ist verloren gegangen oder hat nie existiert
- Erstellen von Daten zur Aufarbeitung oder Herstellung eines Teils, für das es keine CAD-Daten gibt oder für das die Daten veraltet oder verloren gegangen sind
- Inspektion und/oder Qualitätskontrolle — Vergleich eines gefertigten Teils mit seiner CAD-Beschreibung oder zu einem Standardartikel
- Einige schlechte Eigenschaften eines Produkts müssen beseitigt werden. Dabei könnte übermäßiger Verschleiß darauf hinweisen, wo ein Produkt verbessert werden sollte
- Stärkung der guten Eigenschaften eines Produkts auf der Grundlage einer langfristigen Nutzung
- Analyse der guten und schlechten Eigenschaften von Konkurrenzprodukten. Erkundung neuer Wege zur Verbesserung der Produktleistung und -merkmale
- Erstellen von 3D-Daten aus einem Modell oder einer Skulptur für Animationen in Spielen und Filmen
- Erstellen von 3D-Daten von einer Person, einem Modell oder einer Skulptur zum Erstellen, Skalieren oder Reproduzieren von Kunstwerken
- Architektur- und Konstruktionsdokumentation mit Vermessung
- Anpassen von Kleidung oder Schuhwerk an Personen und Bestimmen der Anthropometrie einer Bevölkerung
- Generierung von Daten zur Erstellung von zahnmedizinischer oder chirurgischer Prothetik, von Körperteilen aus Gewebezüchtungen oder für die chirurgische Planung
- Dokumentation und Reproduktion von Tatorten
- ...³

Reversing Prozess

Um zielgerichtetes Reverse Engineering zu betreiben, soll das Projekt in zwei Bereiche aufgeteilt werden. Der erste Bereich ist eine Art groß angelegte Beobachtung/Analyse des Systems/Geräts und wird als System-Level Reversing bezeichnet. Dieses Reversing auf Systemebene hilft dabei, die allgemeine Struktur der Abläufe eines Prozesses oder eines Programmes zu ermitteln. Nachdem ein generelles Verständnis der Abläufe erreicht ist, kann tiefer in die Materie eingetaucht werden. Der zweite Bereich wird

³ Vgl. Raja/Fernandes (2007), S. 3 f

als Code-Level Reversing bezeichnet und beschreibt den komplexen Prozess der Umsetzung von Reversing-Techniken.⁴

3.2 Rechtliche Aspekte

Wenn eine neue Technologie entwickelt wird, stehen den Entwicklern in der Regel zwei Hauptoptionen zum Schutz der einzigartigen Aspekte dieser Technologie zur Verfügung. In einigen Fällen ist die Anmeldung eines Patents die richtige Wahl. Der Vorteil der Patentierung besteht darin, dass sie dem Erfinder oder Patentinhaber die Möglichkeit einräumt, bis zu 20 Jahre lang die Kontrolle über die Erfindung zu behalten. Der Nachteil für den Erfinder ist, dass die Einzelheiten der Erfindung veröffentlicht werden müssen und dass sie nach Ablauf des Patents im Wesentlichen allgemein verwendbar wird. Natürlich macht das Reverse Engineering von patentierten Technologien keinen Sinn, da die Informationen ohnehin öffentlich zugänglich sind.

Eine neu entwickelte Technologie, die nicht patentiert ist, erhält automatisch den rechtlichen Schutz eines Geschäftsgeheimnisses, wenn erhebliche Anstrengungen in ihre Entwicklung und Geheimhaltung gesteckt werden. Ein Geschäftsgeheimnis schützt den Entwickler rechtlich vor Fällen von "Geschäftsgeheimnis-Missbrauch", wie beispielsweise dem Verkauf des Geheimnisses durch einen abtrünnigen Mitarbeiter an einen Konkurrenten. Die Tatsache, dass es sich bei einem Produkt um ein Geschäftsgeheimnis handelt, schützt den Eigentümer jedoch nicht, wenn ein Wettbewerber bei diesem Produkt Reverse Engineering anwendet, vorausgesetzt, das Produkt ist auf dem freien Markt erhältlich und wird rechtmäßig erworben. Ein Geschäftsgeheimnis bietet auch keinen Schutz im Falle eines Konkurrenten, der selbständig erfindet.

In Österreich ist die Rechtslage im §§ 26a bis §§ 26j des Bundesgesetzes gegen den unlauteren Wettbewerb 1984 — UWG definiert. Bei zivilrechtlichen Sonderbestimmungen zum Schutz von Geschäftsgeheimnissen im §26d sind ‚rechtswidriger Erwerb, rechtswidrige Nutzung und rechtswidrige Offenlegung von Geschäftsgeheimnissen sowie Ausnahmen‘ wie folgt geregelt:

„Der Erwerb eines Geschäftsgeheimnisses ist rechtmäßig, wenn das Geschäftsgeheimnis

1. durch unabhängige Entdeckung oder Schöpfung,
2. durch Beobachtung, Untersuchung, Rückbau oder Testen eines Produkts oder Gegenstands, das bzw. der öffentlich verfügbar gemacht wurde oder sich im rechtmäßigen Besitz des Erwerbers der Information befindet, der keiner rechtsgültigen Pflicht zur Beschränkung des Erwerbs des Geschäftsgeheimnisses unterliegt,
3. durch Inanspruchnahme des Rechts der Arbeitnehmer oder Arbeitnehmervertreter auf Information und Anhörung gemäß den bestehenden Vorschriften oder
4. durch jede andere Vorgehensweise, die unter den gegebenen Umständen mit einer seriösen Geschäftspraktik vereinbar ist, bekannt wird.“⁵

⁴ Vgl. Eilam (2011), S. 8 f

⁵ UWG (1984), S. 13

Während in Österreich die Umsetzung der EU-Richtlinie (Nr. 2016/943 zum Schutz von Geschäftsgeheimnissen vor rechtswidrigem Erwerb sowie rechtswidriger Nutzung und Offenlegung) im UWG ergänzt wurde, ist in Deutschland mit 26.4.2019 das Gesetz zum Schutz von Geschäftsgeheimnissen (GeschGehG) in Kraft getreten.

Der § 3 GeschGehG besagt:

„Ein Geschäftsgeheimnis darf insbesondere erlangt werden durch

1. Eine eigenständige Entdeckung oder Schöpfung;
 2. Ein Beobachten, Untersuchen, Rückbauen oder Testen eines Produkts oder Gegenstands, das oder der
 - a) öffentlich verfügbar gemacht wurde oder
 - b) sich im rechtmäßigen Besitz des Beobachtenden, Untersuchenden, Rückbauenden oder Testenden befindet und dieser keiner Pflicht zur Beschränkung der Erlangung des Geschäftsgeheimnisses unterliegt;
 3. ein Ausüben von Informations- und Anhörungsrechten der Arbeitnehmer oder Mitwirkungs- und Mitbestimmungsrechte der Arbeitnehmervertretung.
- (2) Ein Geschäftsgeheimnis darf erlangt, genutzt oder offengelegt werden, wenn dies durch Gesetz, aufgrund eines Gesetzes oder durch Rechtsgeschäft gestattet ist.“⁶

Auffällig dabei ist, dass sich österreichisches UWG und deutsches GeschGehG kaum unterscheiden und die Rechtslage im deutschsprachigen Raum gleich ist.

Da Reverse Engineering auf Punkt 1) und 2) von §26d UWG zutrifft, ist es in Österreich rechtlich erlaubt dies an Produkten oder Gegenständen anzuwenden, sofern sich diese im eigenen Besitz befinden und kein Patentschutz vorliegt.

Im Bereich Software gibt es jedoch unterschiedliche Auffassungen der Gesetzestexte. Urheberrechtsgesetze dienen dazu, Software und anderes geistiges Eigentum vor jeder Art von unautorisierter Vervielfältigung zu schützen. Die rechtliche Debatte um Reverse Engineering dreht sich gewöhnlich um die Frage, welche sozialen und wirtschaftlichen Auswirkungen Reverse Engineering mit sich bringt und wofür es eingesetzt wird. Prinzipiell ist es nicht möglich vorherzusagen, ob ein gewisses Reverse Engineering-Szenario als legal angesehen wird oder nicht, da es von vielen Faktoren abhängt. Der drastischste Fall ist der direkte Diebstahl von Code-Segmenten von einem Konkurrenzprodukt und deren Einbettung in ihr eigenes. Dies ist ein klarer Verstoß gegen das Urheberrecht und in der Regel sehr leicht zu beweisen. Ein komplizierterer Fall ist die Anwendung eines Dekompilierungsprozesses auf ein Programm und die Rekompilierung seiner Ausgabe auf eine Weise mit identischer Funktionalität.

Insgesamt gestaltet sich die Rechtslage im Bereich Software als komplex und bedarf einer sorgfältigen rechtlichen Prüfung im Einzelfall.

⁶ GeschGehG (2019), § 3

4 CONTROLLERTYPEN

In Abb. 4.1 ist ein Blockdiagramm eines generischen Mikrocontrollers abgebildet. Er besteht aus drei Blöcken:

- Prozessor (CPU)
- Speicher (RAM, ROM)
- Ein-/Ausgabesystem (I/O)

Zwischen diesen Einheiten werden drei verschiedene Arten der Datenkommunikation, der Adressierung von Informationen und der Steuerbefehle verwendet. Im Zentrum befindet sich der Mikroprozessor (CPU), der den Ablauf des abgespeicherten Programmes steuert. Die Aufgabe des Mikroprozessors besteht darin, die im Speicher befindlichen Befehle auszulesen, diese zu interpretieren und auszuführen. Der Prozessor beinhaltet die Schaltkreise, die zur Ausführung arithmetischer und logischer Operationen mit binären Daten erforderlich sind. Diese spezielle Schaltungstechnik wird als arithmetische und logische Einheit (ALU) bezeichnet.

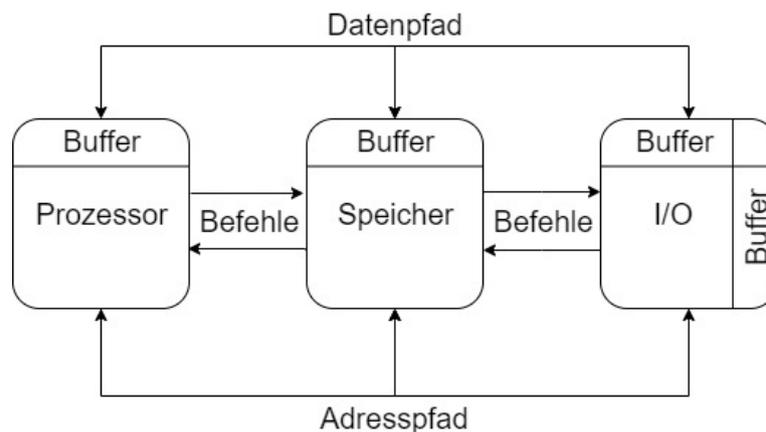


Abb. 4.1: Blockdiagramm Mikrocontroller, Quelle: Eigene Darstellung

Mikrocontroller werden für unterschiedlichste Anwendungen in den Bereichen Automobilindustrie, Kommunikationssysteme, elektronischer Instrumentierung, Krankenhausausrüstung, industrielle Geräte, Haushaltsgeräte, Spielzeug und vielen mehr verwendet. Sie sind für den Einsatz in Anwendungen konzipiert, in denen sie eine kleine Anzahl von Aufgaben zu möglichst geringen wirtschaftlichen Kosten ausführen. Realisiert wird es dadurch, dass ein Programm dauerhaft ausgeführt wird, während die Ein-/Ausgabe-Pins des Mikrocontrollers für die Interaktion mit der Außenwelt genutzt werden.

Die CPU des Mikrocontrollers hat verschiedene Register. Einige dieser Register sind für den allgemeinen Gebrauch bestimmt, während andere einen spezifischen Zweck wie Befehlsregister, Akkumulator, Statusregister, Programmzähler, Datenadressregister und Stack-Pointer haben.⁷

Ein wichtiger Faktor für Mikrocontroller ist die Größe des internen Speicherregisters. Damit ist festgelegt, wie viele Bits gleichzeitig verarbeitet werden können.

⁷ Vgl. Valdes-Perez/Pallas-Areny (2017), S.1 f

Es wird zwischen vier Mikrocontroller-Kategorien unterschieden:

- 4-Bit-Mikrocontroller
- 8-Bit-Mikrocontroller
- 16-Bit-Mikrocontroller
- 32-Bit-Mikrocontroller

Dabei ist anzumerken, dass 4-Bit-Mikrocontroller für einfache Anwendungen wie Fernbedienungen eingesetzt werden und die Verwendung rückläufig ist.

Leistungsfähigkeit

Für die Einstufung der Leistungsfähigkeit von Mikrocontrollern wird die Maßeinheit MIPS (Million instructions per second) angegeben. Für eine aussagekräftige Beurteilung der Performance müssen neben MIPS weitere Faktoren berücksichtigt werden. Der Unterschied liegt im Anwendungsgebiet, bei dem sich die Anforderungen stark unterscheiden können. Applikationen können aufgrund einer Systemintegration viele Taktgeber und Schnittstellen für Kommunikation besitzen, während bei anderen Applikationen der Fokus auf hoher Genauigkeit und die Analog-Technik gelegt wird. Bei Applikationen mit Echtzeitfähigkeit darf es keinen Abbruch der Datenkommunikation geben, was einen geordneten Programmablauf mit vorherbestimmten Zykluszeiten bedingt.

Unabhängig von den spezifischen Systemanforderungen sind Anwendungsfreundlichkeit der Entwicklungstools, Beispiele und gute Dokumentation inklusive ausreichendem Online-Support ein wesentlicher Bestandteil der Performance-Beurteilung.⁸

8-/16-Bit-Mikrocontroller

8-Bit Mikrocontroller wurden aufgrund der meist linear mit der Bitbreite steigenden Kosten am häufigsten verwendet, wobei mittlerweile öfter leistungsfähigere Mikrocontroller gewählt werden. Die Bitbreite hat sich als sinnvolle Word-Größe für Steuer- und Überwachungsanwendungen erwiesen. Vorteile von alten 8-/16-Bit-Mikrocontroller sind hohe I/O-Stromstärken, viele interne und externe Oszillatoroptionen und keine Notwendigkeit für externe Komponenten zur Aufnahme von On-Chip-Spannungsreglern. Der Energiebedarf bei 8-/16-Bit-Mikrocontrollern ist im Vergleich zu 32-Bit-Mikrocontrollern meist geringer (obwohl das stark von der Technologie abhängt) und der zur Verfügung stehende Speicher ist deutlich kleiner, was die Größe des darauf laufenden Programmes limitiert.

32-Bit-Mikrocontroller

Um die Probleme alter CPU-Architekturen zu lösen, die den Marktanforderungen nicht mehr gerecht werden, bieten viele Hersteller 32-Bit-Lösungen an. Dies schafft Abhilfe für Anwendungen, die 32-Bit-Rechenleistung benötigen, wobei viele Anwender aus den falschen Gründen zu 32-Bit wechseln.

In Bezug auf die Energieeffizienz sind 32-Bit-Architekturen durch die hohe Rechengeschwindigkeit den 8-/16-Bit-Architekturen deutlich überlegen. Daher kommen höhere Bandbreiten für Applikationen mit hoher Rechenleistung zur Anwendung.

⁸ Vgl. Seather/Fredriksen (2008), S 3

Eine 32-Bit-CPU enthält wesentlich mehr digitale Logik als jede 8-/16-Bit-CPU, was die Kosten erhöht. Ein wesentlicher Nachteil ist ein viel höherer Leckstrom und statische Stromaufnahme. Anwendungen wie beispielsweise Wasserzähler, Gaszähler, Thermostate, Schlüsselanhänger, Mautstraßenanhänger und Alarmsysteme befinden sich zu einem Großteil ihrer Zeit im Schlafmodus, indem die CPU angehalten wird. Da die Batterielebensdauer solcher Anwendungen 5-10 Jahre betragen muss, sind diese Geräte mit CPU-Architekturen niedrigerer Leistungsfähigkeit realisiert.

Seit 2008 bietet Atmel die AVR XMEGA Mikrocontroller-Familie an, die ein Technologie-Upgrade liefert, das die Anwendungsgebiete von 8-/16-Bit Mikrocontrollern für neue Embedded-Designs erweitert.⁹

Führende Hersteller mit zugehörigen Mikrocontroller-Familien

- ARM-Mikrocontroller (meist ARM Cortex-M Kerne im Mikrocontroller-Bereich angewendet)
- Microchip Technology Atmel AVR (8-bit), AVR32 (32-bit) und AT91SAM (32-bit)
- Microchip Technology PIC (8-bit PIC16, PIC18, 16-bit dsPIC33 / PIC24), (32-bit PIC32)
- Freescale ColdFire (32-bit), S08 (8-bit)
- Intel 8051 und abgeleitete Typen wie Silicon Laboratories Pipelined 8-bit 8051
- PowerPC ISE
- Renesas Electronics: RL78 16-bit MCU, RX 32-bit MCU, SuperH, V850 32-bit MCU, H8, R8C 16-bit MCU
- Texas Instruments TI MSP430 (16-bit), MSP432 (32-bit) und C2000 (32-bit)
- Toshiba TLCS-870 (8-bit/16-bit)¹⁰

Interessante Mikroprozessoren

Für die Evaluierung der interessantesten Mikrocontroller wird eine Studie aus dem Jahr 2019 zum Thema „Integrating IoT and Advanced Technology Designs, Application Development & Processing Environments“ herangezogen. Dafür wurden verschiedene Entwickler über deren aktuell verwendete Hardware für die Umsetzung der Projekte befragt. Dabei geht hervor, dass der wichtigste Faktor bei der Wahl des Mikroprozessors ist, welche Software-Entwicklungstools zur Verfügung stehen, gefolgt von der Leistungsfähigkeit des Chips.

In Abb. 4.2 ist das Ergebnis der Befragung dargestellt, von welchem Hersteller der aktuell verwendete Prozessor gekauft wurde. Daraus gehen fünf Hersteller für Mikroprozessoren hervor, die von mehr als 20 % der Befragten in Verwendung sind.

⁹ Vgl. Seather/Fredriksen (2008), S 5

¹⁰ Vgl. RS Components, Architekturen

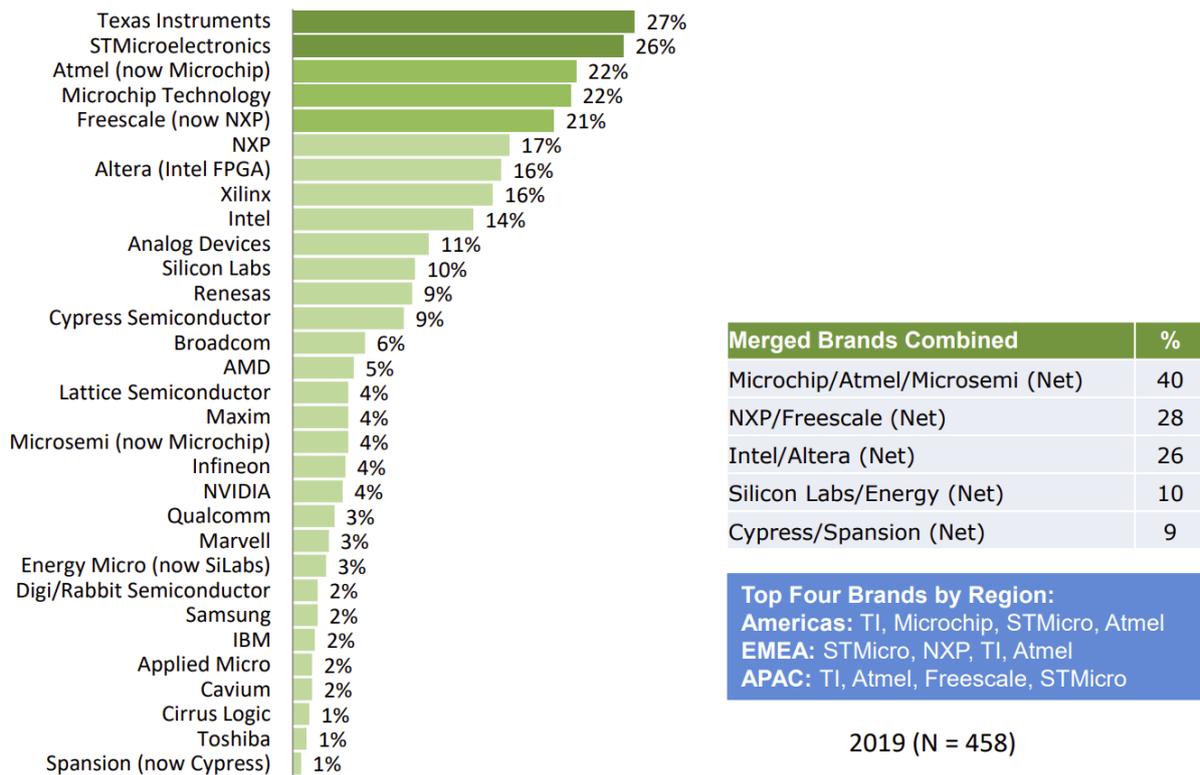


Abb. 4.2: Ergebnis der Befragung über den Hersteller des aktuell verwendeten Prozessors, Quelle: Studie Aspencore (2019) https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf

In Abb. 4.3 ist das Ergebnis der Befragung dargestellt, von welchem Hersteller der verwendete Prozessor des nächsten Projektes sein soll.

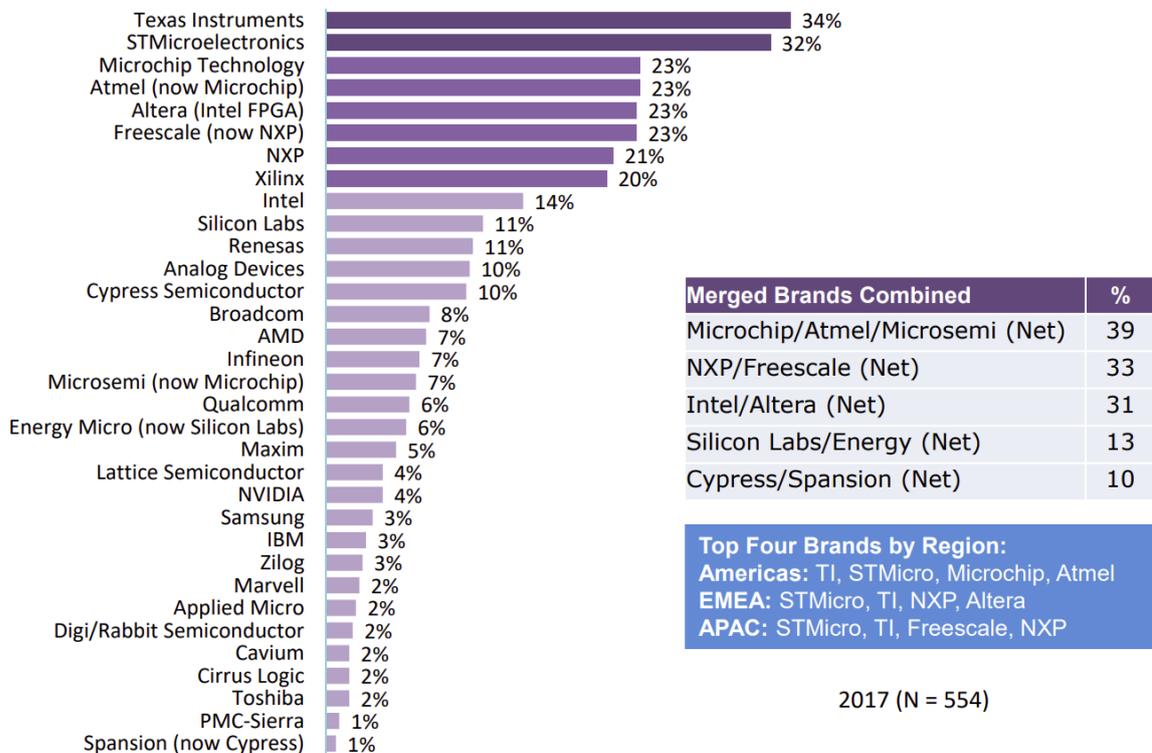


Abb. 4.3: Ergebnis der Befragung des gewünschten Herstellers des Prozessors für zukünftige Projekte, Quelle: Studie Aspencore (2019) https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf

Aus der Studie ergeben sich folgende wichtige Prozessoren und Hersteller:

8-Bit:

Microchip Technology Atmel AVR (8-bit)

Microchip Technology PIC16, PIC18

Freescale S08

Intel 8051 kompatibel

16-Bit:

Texas Instruments TI MSP430

Microchip Technology dsPIC33 / PIC24

Renesas Electronics RL78

32-Bit:

STMicroelectronics ARM STM32

Microchip Technology Atmel AVR32 und AT91SAM

Microchip Technology PIC32

Freescale ColdFire

Speicher

Wie in Abb. 4.1 dargestellt, ist der Speicher mit der CPU durch Busse für Daten, Adresse und Steuerung verbunden. Die Busbreite muss mit der Menge an Daten übereinstimmen, die von der CPU verarbeitet werden können. Daher hat beispielsweise ein 8-Bit Prozessor einen Datenbus mit derselben Breite. Der Adressbus hat meist eine höhere Busbreite, da sonst der hexadezimale Adressbereich zu gering wäre.

In seltenen Fällen werden die Busse in einem Mikroprozessor für den Zugriff auf den externen Speicher auf Pins herausgeführt.

Speicherklassifizierung

Flüchtiger Speicher verliert seinen Inhalt, wenn die Stromversorgung unterbrochen wird. Dieser wird als Random-Access-Memory (RAM) bezeichnet. Der Vorteil des flüchtigen Speichers ist ein einfacher Schreib-/Lese-Vorgang, mit damit verbundenen kurzen Zugriffszeiten. Bei flüchtigen Speichern wird zusätzlich zwischen static RAM und dynamic RAM unterschieden. Der bei Mikrocontrollern meist verwendete static RAM behält die Daten, wenn der Taktgeber stoppt, vorausgesetzt die Stromversorgung wird nicht unterbrochen. Aufgrund des komplexen Aufbaues des daraus resultierenden größeren Platzbedarfes an Chipfläche und den damit verbundenen Kosten ist meist wenig Speicher vorgesehen. Beim platzsparenden dynamic RAM muss der Speicherinhalt regelmäßig aktualisiert werden, um den Inhalt nicht zu verlieren.

Nicht flüchtiger Speicher behält seinen Inhalt auch wenn die Stromversorgung unterbrochen wird und findet seine Anwendung daher als Programmspeicher und Speicher für dauerhafte Informationen. Dieser wird als

„Read-Only-Memory“ (ROM) bezeichnet, obwohl die meisten modernen Mikrocontroller auf den Speicher auch schreiben können. Es ist aber zu beachten, dass Dauer und Komplexität beim Schreiben auf nicht flüchtigen Speicher bedeutend höher ist als auf flüchtigen Speicher. Neben dem „masked ROM“ dem „one-time programmable Memory“ (OTP) und dem „electrically programmable ROM“ (EPROM) gibt es den „electrically erasable programmable ROM“ (EEPROM). Dieser ist der am weitesten verbreitete nicht flüchtige Speicher, bei dem einzelne Bytes gelöscht werden können während bei anderen Speichertypen nur blockweise gelöscht werden kann.¹¹

Speicherarchitektur:

Für den flüchtigen und nicht flüchtigen Speicher gibt es zwei unterschiedliche Konzepte zur Steuerung des Zugriffes darauf.

Harvard Architektur: Bei diesem Konzept werden flüchtige und nicht flüchtige Speicher in zwei separate Systeme aufgeteilt, indem beide eine eigene Adresse und einen eigenen Datenbus besitzen.

Der Vorteil der Harvard Architektur ist, die Möglichkeit eines parallelen Zugriffs auf Programm und Datenspeicher. Dadurch kann beispielsweise das Lesen einer Variable aus dem Datenspeicher gleichzeitig mit dem Auslesen des nächsten Programmablaufes aus dem Programmspeicher erfolgen. Zusätzlich können die beiden separaten Systeme in Bezug auf Busbreiten speziell optimiert werden.

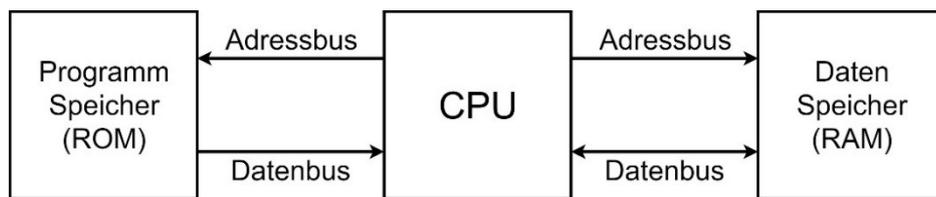


Abb. 4.4: Harvard Architektur, Quelle: Eigene Darstellung

Von Neumann Architektur: Hier gibt es nur ein Speichersystem, bei dem ein Adressbereich sowohl auf flüchtigen als auch nicht flüchtigen Speicher Zugriff hat.

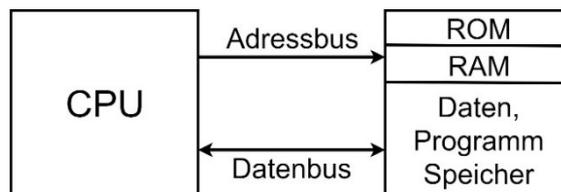


Abb. 4.5: Von Neumann Architektur, Quelle: Eigene Darstellung

Diese Architektur ist intrinsisch wenig effizient, da oft mehrere Zyklen für einen vollständigen Lesevorgang von Instruktionen aus dem Speicher notwendig sind. Der Vorteil aber ist, dass dieses System einfach zu steuern ist und es keinen Unterschied beim Zugriff auf flüchtigen und nicht flüchtigen Speicher gibt.¹²

¹¹ Vgl. Davies (2008), S. 11 f

¹² Vgl. Davies (2008), S. 13

Ein weiterer Vorteil dabei ist, dass die Größe der Programme und die Menge an Daten flexibler sein kann. Der Nachteil ist, dass das Überschreiben des Programmes möglich ist.

Programmierung

Der größte Unterschied zwischen dem Schreiben eines Programmes für einen Computer und einem Mikrocontroller ist das Fehlen eines Betriebssystems (in den meisten Fällen). Computer verbringen viel Zeit mit Berechnungen zur Analyse von Daten oder für grafische Anzeigen. Der Prozessor eines Mikrocontrollers im Gegensatz verbringt einen Großteil der Zeit mit Interaktion der Peripheriegeräte und muss diese interpretieren. Für die Programmierung von Mikrocontrollern gibt es mehrere Sprachen.

Maschinencode:

Der Maschinencode besteht aus binären Daten, die der Prozessor selbst versteht. Jede Anweisung hat einen binären Wert, der Opcode genannt wird. Er ist nur für Programmierer mit Erfahrungen im Bereich Low-Level Debugging verständlich. Manche sehr frühe Computer verlangten eine Programmierung in Maschinencode.

Assemblersprache:

Die Assemblersprache ist im Grunde eine Übersetzung des Maschinencodes in die englische Sprache. Die Anweisungen werden als Wörter und nicht als binäre Werte geschrieben, die Mnemonik genannt werden und vom Assembler-Programm in Maschinencode übersetzt werden. Ein Nachteil der Assemblersprache ist, dass sie sehr spezifisch auf den verwendeten Prozessor und auf die gewählte Entwicklungsumgebung zugeschnitten ist und daher für jede Architektur anders sein kann. Frühere Mikrocontroller mussten mit Assemblersprache programmiert werden, weil Compiler für C noch einen wenig effizienten Code produzierten. Mittlerweile sind die Compiler besser und moderne Prozessoren werden mit Blick auf die Compiler entworfen, sodass die Assemblersprache an den Rand gedrängt wurde. Heute findet die Assemblersprache Anwendungen für wenige Operationen, insbesondere bitweise Rotationen, die nicht direkt in C geschrieben werden können. Dekompilierung ist der entgegengesetzte Prozess zu Assembler und bedeutet die Übersetzung von Maschinencode in Assemblersprache.

C:

Die Programmierung in C war über lange Zeit die gebräuchlichste Wahl bei Mikrocontrollern. Ein Compiler übersetzt C in Maschinencode, den die CPU verarbeiten kann. Dies bringt die Vorteile einer Hochsprache wie Datenstrukturen, Funktionen, Typüberprüfung und vieles mehr mit sich; in Verbindung mit effizientem Code beim Übersetzen. Über den Compiler wird direkt Maschinensprache erzeugt, wodurch Assembler nicht mehr benötigt wird.

C++:

Dies ist eine objektorientierte Sprache, die meist für größere Anwendungen verwendet wird. Die Programmiersprache hat den Nachteil, dass manche Funktionen einen ineffizienten Maschinencode generieren. Embedded C++ ist eine abgeleitete Programmiersprache mit eingeschränktem Funktionsumfang, die für eingebettete Systeme mit geringen Ressourcen entwickelt wurde. Java ist eine weitere objektorientierte Sprache, die mehr interpretiert als kompiliert wird und einen wesentlich leistungsfähigeren Prozessor benötigt.

Basic:

Diese Programmiersprache ist für einige wenige Prozessoren verfügbar und besitzt spezielle Anweisungen zur Ansteuerung der Peripheriegeräte. Dadurch können Programme sehr schnell entwickelt werden, ohne dass ein detailliertes Verständnis der Peripheriegeräte erforderlich ist. Nachteile sind, dass der Code oft sehr langsam läuft und die Hardware teuer ist, wenn sie einen Interpreter enthält.¹³

Shields

Als Shields werden vorgefertigte Schaltungsboards bezeichnet, die meist mechanisch auf dessen Hersteller zugehöriges Mikrocontroller-Board angesteckt werden können. Dadurch ist es möglich, Funktionalitäten des bestehenden Mikrocontrollers wie beispielsweise Internetverbindung, Wireless Kommunikation oder LCD-Steuerung zu erweitern.



Abb. 4.6: Arduino MKR CAN Shield, Quelle: Arduino, Online-Quelle [22.August.2020]

In Abb. 4.6 ist das Arduino Shield abgebildet, das auf ein Arduino-Mikrocontroller-Board gesteckt werden kann und dieses dadurch über CAN (Controller Area Network) in ein Bussystem integriert werden kann.

Andere Shields wie in Abb. 4.7 abgebildet, dienen zur Erstellung von Schaltungen als Prototypen für bestehende Pinouts von Mikrocontrollern.

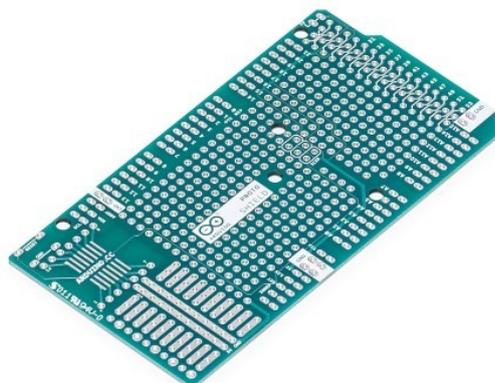


Abb. 4.7: Arduino MEGA Prototyping Shield, Quelle: Arduino, Online-Quelle [22.August.2020]

¹³ Vgl. Davies (2008), S. 15 f

Raspberry Pi:

Der Raspberry Pi ist ein Einplatinen-Rechner im Scheckkartenformat. Er verfügt über vorinstallierte Software, wie „Scratch“, womit Benutzer Animationen, Spiele oder Videos programmieren und gestalten können. Über die Programmiersprache Python können Skripts und Programme für beispielsweise Client/Server Kommunikation entwickelt werden. Er hat sich als bemerkenswert populär erwiesen, wodurch es zahlreiche Beispielprogramme und andere Hilfestellungen gibt.



Abb. 4.8: Raspberry Pi 4 Model B, Quelle: Raspberry Pi, Online-Quelle [24.Juli.2020]

Der Mikroprozessor des Raspberry Pi besitzt eine ARM Architektur, worauf ein komplettes Betriebssystem laufen kann. Durch die Anschlussmöglichkeiten wie HDMI und USB kann ein Bildschirm mit Tastatur und Maus angeschlossen und der Mikrokontroller so als Computer verwendet werden. Zusätzlich besitzt der Raspberry Pi einen Standard 40-Pin GPIO Header, was das Auslesen von Sensoren, beziehungsweise das Steuern der Ausgänge und/oder Kommunikation mit anderen Geräten ermöglicht. Beim Raspberry Pi ist es aufgrund der hohen Rechenleistung und verfügbarem Speicher möglich, verschiedene Applikationen parallel laufen zu lassen. Ein Nachteil davon ist, dass der Zeitpunkt der Abarbeitung bestimmter Programmteile dadurch nicht exakt bestimmt werden kann.

Eigenschaften:

Prozessor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-Bit SoC @ 1.5 GHz
Speicher:	2 GB, 4 GB oder 8 GB LPDDR4 (abhängig vom Modell)
Anschlüsse:	2.4 GHz und 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet

	2 × USB 3.0 Ports 2 × USB 2.0 Ports
GPIO (Spannung):	Standard 40-Pin GPIO Header 3,3 Volt I/O
Video & Sound:	2 × Micro HDMI Ports (bis 4Kp60 unterstützt) 2-lane MIPI DSI Display Port 2-lane MIPI CSI Kamera Port 4-poliger Stereo Audio und Composite Video Port
Multimedia:	H.265 (4Kp60 decode); H.264 (1080p60 decode, 1080p30 encode); OpenGL ES, 3.0 Grafik
SD-Kartenunterstützung:	Micro SD
Energieversorgung:	5V DC via USB-C Connector (min. 3A) 5V DC via GPIO Header (min. 3A) Power over Ethernet (PoE) (benötigt separaten PoE HAT)

Tab. 4.1: Raspberry Pi 4 Eigenschaften, Quelle: Raspberry Pi, Online-Quelle [24.Juli.2020]

Arduino:

Im Unterschied zum Raspberry Pi kann auf dem Arduino Mikrocontroller kein komplettes Betriebssystem laufen, sondern es wird ein Programm zyklisch in einer Schleife abgearbeitet. Der Arduino Mega besitzt viele Hardware Pins, wobei die meisten davon digitale und analoge Ein-/Ausgänge sind.

Im Gegensatz zum Raspberry Pi hat der Arduino deutlich weniger Rechenleistung und Speicher zur Verfügung. Das muss nicht zwangsläufig ein Nachteil sein und hängt stark von der Anwendung ab. Durch die häufige Verwendung des Arduino gibt es verhältnismäßig viele Beispiele und online Support in diversen Arduino-Foren, die bei Problemstellungen weiterhelfen können. Außerdem wird die Programmierung durch vorgefertigte Funktionen aus Bibliotheken beschleunigt. Durch die zyklische Abarbeitung des Programmes, kann ein Determinismus erreicht werden, da die Dauer einzelner Funktionen beim Schreiben der Software vom Programmierer fix definiert werden kann.

Arduino Mega 2560 Rev3:

Der Arduino Mega 2560 Rev3 ist ein Mikrocontroller-Board, das auf dem Prozessor ATmega2560 basiert. Er verfügt über 54 digitale Ein-/Ausgangspins (von denen 15 als PWM-Ausgänge verwendet werden

können), 16 analoge Eingänge, 4 UARTs (serielle Hardware-Ports), einen 16-MHz-Quarzoszillator, einen USB-Anschluss, eine Netzbuchse, einen ICSP-Header und einen Reset-Knopf.¹⁴



Abb. 4.9: Arduino Mega 2560 Rev3, Quelle: Arduino, Online-Quelle [22.August.2020]

Eigenschaften:

Prozessor:	AVR® 8-Bit Microcontroller 16 MHz
Speicher:	Flash Speicher 256 KB SRAM 8 KB EEPROM 4 KB
Anschlüsse:	1 x USB 2.0
GPIO (Spannung):	70 digitale I/O Pins (wovon 15 für PWM und 16 für analoge I/Os geeignet sind) 5 Volt I/O
Video & Sound:	N/A
Multimedia:	N/A
SD-Kartenunterstützung:	N/A
Energieversorgung:	7 – 12 V

Tab. 4.2: Arduino Mega 2560 Rev3 Eigenschaften, Quelle: Arduino, Online-Quelle [22.August.2020]

¹⁴ Arduino Store, Arduino Mega 2560 Rev3

Odroid C4:

Der Odroid C4 ist die Weiterentwicklung des Odroid C2 und ist ähnlich dem Raspberry Pi ein Einplatinencomputer auf Basis von 64 Bit. Die Programmierung kann über die von Arduino zur Verfügung gestellte Programmierumgebung „Arduino IDE“ erfolgen. Dafür wird von Odroid die Bibliothek „odroid_go.h“ zur Verfügung gestellt, die inkludiert werden kann. Diese Bibliothek beinhaltet Grundfunktionen wie bei Arduino, die dem Programmierer den Einstieg erleichtern.¹⁵



Abb. 4.10: Odroid C4, Quelle: Odroid, Online-Quelle [22.Juli.2020]

Eigenschaften:

Prozessor:	Amlogic S905X3 (ARM® Cortex®-A55 (ARMv8a) 2.0 GHz Quad Core CPU)
Speicher:	4 GB DDR4 SDRAM eMMC5.0 HS400 Flash Storage Slot / UHS-1 SDR50 MicroSD Slot
Anschlüsse:	Gigabit Ethernet USB 3.0 Host x 4, USB OTG x 1 Infrared(IR) Receiver HDMI 2.0
GPIO (Spannung):	Standard 40-Pin GPIO-Header 3,3 Volt I/O
Video & Sound:	1x HDMI bis 4K/60 Hz mit HDR, CEC, EDID

¹⁵ Vgl. Lee (2018), S. 1 f

Multimedia:	ARM Mali™-G31 MP2 GPU (OpenGL ES 2.0/1.1 für Linux und Android)
SD-Kartenunterstützung:	MicroSD Slot
Energieversorgung:	5,5 – 15,5 V DC

Tab. 4.3: Odroid C4 Eigenschaften, Quelle: Odroid, Online-Quelle [22.Juli.2020]

BeagleBone:

Der Beaglebone Black (BBB) ist ein Einplatinen-Rechner, der von Texas Instruments hergestellt wird. Wie der Raspberry Pi wurde auch der BBB seit seiner Erstveröffentlichung im Jahr 2013 einer Reihe von Änderungen unterzogen. Der BBB kombiniert viele der Vorteile des Raspberry Pi mit denen eines Arduino-Mikrocontrollers. Er ist in der Lage, ein Linux-System wie der Raspberry Pi zu betreiben und wie der Arduino hat er einen eingebauten Analog-to-Digital-Converter (ADC). Der ADC des Beaglebone Black hat jedoch 12 Bit statt 10 Bit und kann mit über 100 ksp/s betrieben werden. Wie die anderen Mikrocontroller ist der Beaglebone Black im Scheckkartenformat, läuft mit etwa 3 Watt und ist von der Größe her dem Raspberry Pi sehr ähnlich. Der Prozessor des Beaglebone Black ist ein ARM Sitara, der mit 1 GHz läuft. Der Hauptvorteil des BBB gegenüber anderen Mikrocontrollern ist das Vorhandensein von zwei programmierbaren Echtzeit-Einheiten (programmable real-time units PRUs). Jeder dieser 32-Bit-Prozessoren läuft unabhängig vom Hauptprozessor und ist dabei in der Lage, durch die Verwendung von gemeinsam genutztem RAM und einer Reihe von Interrupts, Informationen mit dem Hauptprozessor zu kommunizieren. Die PRUs laufen mit einer Taktfrequenz von 200 MHz, deutlich schneller als die 16 MHz des Arduino Mega2560. Der Nachteil dabei ist, dass PRUs vom Hersteller nicht unterstützt werden, sodass keine zuverlässigen Informationen über die Programmierung vorhanden sind. Die PRUs können über die GPIO-Pins auf dem Beaglebone Black direkt auf externe Sensoren zugreifen. Allerdings ist die Konfiguration dieser Pins kompliziert. Die PRUs müssen in Assemblersprache programmiert werden, was die Anwendungsentwicklung sehr zeitaufwendig macht.¹⁶

¹⁶ Vgl. Travaglione/Munyard/Matthews (2014), S. 4 f

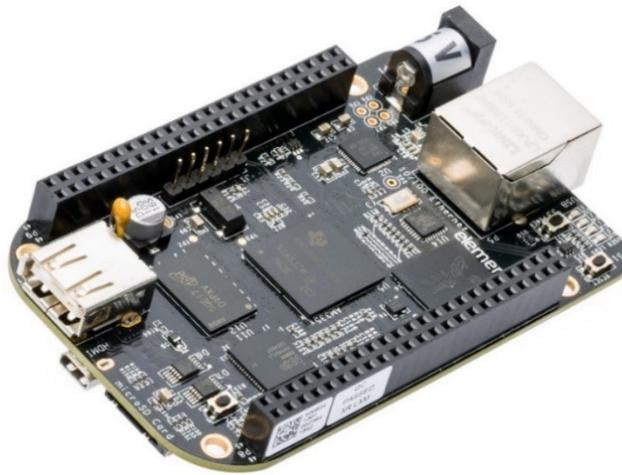


Abb. 4.11: Beagle Bone Black, Quelle: BeagleBoard, Online-Quelle [24.Juli.2020]

Eigenschaften:

Prozessor:	AM335x 1 GHz ARM® Cortex-A8
Speicher:	512 MB DDR3 RAM 4 GB 8-Bit eMMC on-Board Flash Speicher
Anschlüsse:	USB Client für Power & Kommunikation USB Host Ethernet HDMI
GPIO(Spannung):	2 x 46 Pin Headers 3,3 Volt I/O
Video & Sound:	N/A
Multimedia:	3D graphics accelerator
SD-Kartenunterstützung:	N/A
Energieversorgung:	5 V
Software Kompatibilität:	Debian Android Ubuntu Cloud9 IDE on Node.js w/ BoneScript library

Tab. 4.1: Beagle Bone Black Eigenschaften, Quelle: BeagleBoard, Online-Quelle [24.Juli.2020]

Banana Pi:

Der Banana Pi BPI F2S verbindet die Leistung eines Linux-Systems mit der einfachen Integrierfähigkeit eines Mikrocontrollers mit Cortex-A7 Quad-Core, ARM926 Echtzeit-Core und einem 8051 Low-Power-Core Prozessor. Der F2S bietet umfangreiche GPIOs, Speicher und eine USB-Schnittstelle. Zusätzlich ist eine MIPI-CSI-Schnittstelle für den Videoeingang und eine HDMI-Schnittstelle für den Videoausgang vorgesehen. Außerdem bietet es eine FPGA-Schnittstelle zur Funktionserweiterung und 4 Ethernet-Ports.



Abb. 4.12: Banana Pi BPI F2S, Quelle: Banana Pi, Online-Quelle [24.Juli.2020]

Eigenschaften:

Prozessor:	Quad-core 1 GHz Cortex-A7 CPU, plus A926 and 8051 cores
Speicher:	Built-in 128 MB oder 512 MB DDR3 DRAM 8 GB eMMC Flash onboard MicroSD Slot
Anschlüsse:	2 x 10/100M Ethernet RS-485 RS-232 HDMI 1.4 Output MIPI CSI Connector 2 x USB 2.0 Ports 1 x USB 2.0 OTG
GPIO(Spannung):	Standard 40 Pin GPIO-Header 2 x 50 Pin FPGA Interface Power (+5 V, +3,3 V und GND)

	UART, I2C, SPI oder PWM 1,8 – 3,3 Volt I/O
Video & Sound:	HDMI 1.4 Output
Multimedia:	N/A
SD-Kartenunterstützung:	MicroSD Slot
Energieversorgung:	12 VDC @2 A PoE (benötigt separates PoE Modul)
Software Kompatibilität:	Linux, Android

Tab. 4.2: Banana Pi BPI F2S Eigenschaften, Quelle: Banana Pi, Online-Quelle [24.Juli.2020]

ASUS Tinker Board:

Das Unternehmen ASUS hat mit dem Tinker Board ein dem Raspberry Pi ähnliches Board entwickelt, das einen 32-Bit RK3288 Rock-Chip-Prozessor besitzt. Im Gegensatz zur Raspberry hat das Tinker Board im Moment nur zwei offizielle Betriebssysteme, TinkerboardOS von Debian und TinkerOS Kodi. Es verfügt über 2 GB LPDDR3 Dual-Channel-Speicher. Zusätzlich ist das Tinker Board mit einer SDIO 3.0 Schnittstelle ausgestattet, die für erhöhte Lese- und Schreibgeschwindigkeit für MicroSD-Erweiterungskarten sorgen soll. Diese werden sowohl für das Betriebssystem als auch zum Speichern von Anwendungen und Dateien verwendet.

Das Tinker Board unterstützt die gängigen Verbindungsmöglichkeiten, einschließlich einer 40-Pin GPIO-Schnittstelle.

Im Vergleich zu anderen Mikrocontrollern wird das Tinker Board nicht häufig verwendet und bietet dadurch weniger Unterstützung für Einsteiger und bei Problemstellungen.



Abb. 4.13: ASUS Tinker Board, Quelle: Asus, Online-Quelle [24.Juli.2020]

Eigenschaften:

Prozessor:	Rockchip Quad-Core RK3288 Prozessor
Speicher:	2 GB Dual Channel DDR3 Micro SD
Anschlüsse:	4 x USB 2.0 Ethernet 802.11 b/g/n WiFi, Bluetooth V4.0 + EDR HDMI MIPI DSI MIPI CSI
GPIO:	40 Pin GPIO-Header
Video & Sound:	RTL ALC4040 CODEC
Multimedia:	Integrated Graphics Processor ARM® Mali™-T764 GPU*1
SD-Kartenunterstützung:	MicroSD Slot
Energieversorgung:	5 V 2~2,5 A
Software Kompatibilität:	Android Debian

Tab. 4.3: ASUS Tinker Board Eigenschaften, Quelle: Asus, Online-Quelle [24.Juli.2020]

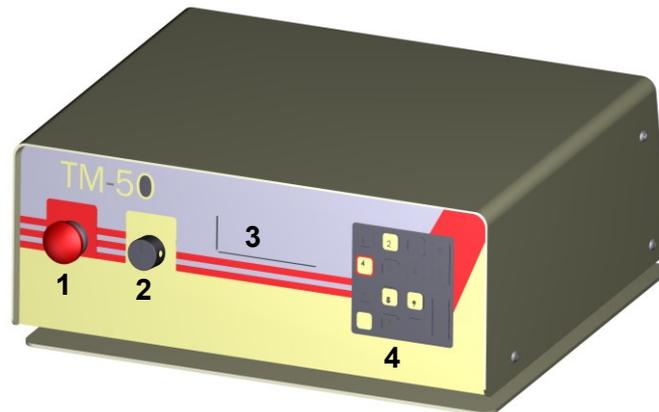
Wahl des Mikrocontrollers

Die Recherche zeigt, dass die derzeitigen Mikrocontroller für eine Vielzahl an Datenerfassungs- und Signalverarbeitungsaufgaben geeignet sind. Die kleine Baugröße und der geringe Energiebedarf können bei der Entwicklung von tragbaren und leichten Geräten hilfreich sein. Die niedrigen Kosten sind ein klarer Vorteil im Vergleich zu den kommerziellen Alternativen. Jedes der recherchierten Mikrocontroller-Boards besitzt Stärken und Schwächen in unterschiedlichen Bereichen. Die Entscheidung, welcher Controller für die Umsetzung eines Projektes gewählt wird, hängt stark von den unterschiedlichen Anforderungen ab. Dafür muss geklärt werden, worauf im Projekt Wert gelegt wird und welche physikalischen Voraussetzungen für das Projekt erforderlich sind.

5 CONTROLLER EVALUIERUNG

5.1 Hardwareanalyse Verpackungseinheit

Für die Wahl eines Mikrocontrollers für die Umsetzung des Projektes muss das Verpackungssystem OEM TM-50 analysiert werden. Eine Komponente ist der Steuerkasten, der das originale Controller-Board beinhaltet.

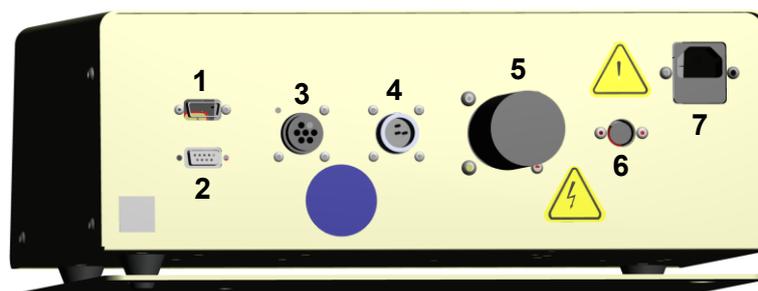


- 1 ... Hauptschalter
- 2 ... Drehpotentiometer für die Aufrollstraffung
- 3 ... LC-Display
- 4 ... Tastenfeld

Abb. 5.1: Vorderseite TM50, Quelle: Eigene Darstellung

Auf der Vorderseite (Bedienseite) des Steuerkastens befinden sich Bedien- und Anzeigeelemente. Über den Hauptschalter kann die gesamte Verpackungseinheit stromlos gemacht werden. Über das Drehpotentiometer kann das Drehmoment für das Aufrollen der verpackten Module eingestellt werden. Am LC-Display werden die wichtigsten Informationen der aktuellen Einstellungen angezeigt. Mit Hilfe des Tastenfeldes können diese Einstellungen über Navigation in der Menüführung auf dem Display geändert werden.

Wie in Abb. 5.2 dargestellt, sind an der Rückseite des Steuerkastens verschiedene Verbindungsmöglichkeiten für die Verbindung der einzelnen Komponenten ausgeführt.



- 1 ... Vision Port
- 2 ... COM Port
- 3 ... Sensor Through
- 4 ... I/O
- 5 ... Peripheral
- 6 ... Foot Switch
- 7 ... Power

Abb. 5.2: Rückseite TM50, Quelle: Eigene Darstellung

Für den Betrieb der Verpackungseinheit bei Flex werden nicht alle Verbindungsmöglichkeiten benötigt. Die systemrelevanten Verbindungen sind:

- Power für die Stromversorgung
- Peripheral für die Überwachung des Prozesses über Sensoren
- I/O für die Statusinformationen zur Ansteuerung
- Com Port für die serielle Kommunikation

In Abb. 5.3 ist der geöffnete Steuerkasten dargestellt, in dem sich das originale Controller-Board befindet. Aufgrund des nicht verfügbaren Schaltplans der Verpackungseinheit muss die Funktionsweise ermittelt werden. Für die Versorgung des Boards ist ein AC/DC-Converter eingebaut, worauf abzulesen ist, dass die Spannung von 230 Volt Wechselfspannung auf 24 Volt Gleichspannung bei maximal 1,3 Ampere konvertiert wird. Für die Heizelemente der Schweißkufen ist ein weiteres Regler-Board inklusive Transformator für höhere Heizleistung unabhängig verbaut.

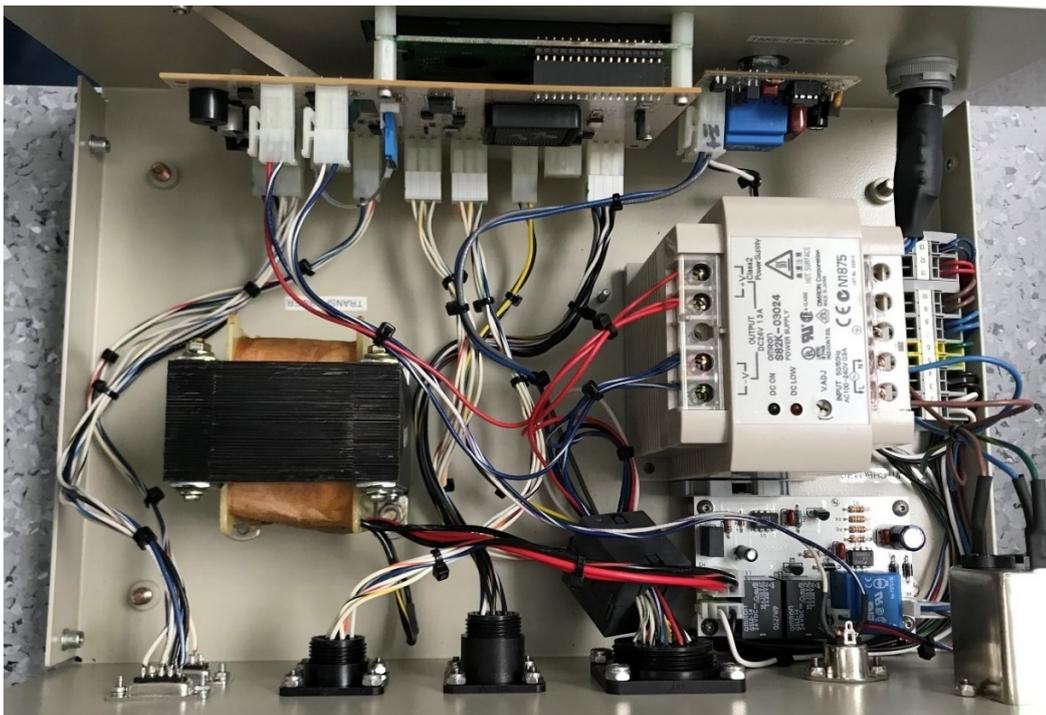


Abb. 5.3: Steuerkasten im Inneren, Quelle: Eigene Darstellung

Die Anschlussmöglichkeiten des Steuerkastens werden im Inneren auf das Controller-Board geführt und über Molex-Stecker kontaktiert. Durch Nachverfolgung der Kabelführung können die farblich gekennzeichneten Kabel den einzelnen Funktionen zugeordnet werden. Ein Stecker mit dem gelbfarbigen Kabel führt beispielsweise zu einem Schrittmotor in der Verschweißungssteuerung.

Für die Messung der anliegenden Spannungen wird ein Multimeter verwendet mit dem alle Verbindungsmöglichkeiten des Controller-Boards geprüft werden. Die Messung der Spannungen an den Pins zeigt, dass der originale Controller mit 5 Volt an den Ein- und Ausgängen arbeitet.

Um die Anforderungen an den Mikrocontroller evaluieren zu können, wird das Signal an den Pins für die Ansteuerung des Schrittmotors gemessen. Dafür wird das Universalmessgerät VirtualBench VB-8012 von National Instruments verwendet. In Abb. 5.4 ist ersichtlich, dass Tastköpfe für die Messung mittels Oszilloskop verwendet werden. Davon wird erwartet, die benötigte Schaltfrequenz des Mikrocontrollers für die Ansteuerung des Schrittmotors zu ermitteln. Aus analytischen Überlegungen wird davon ausgegangen, dass die Schaltfrequenz hier am höchsten ist und dieser Wert als Referenz genommen werden kann.



Abb. 5.4: Virtual Bench VB-8012, Quelle: Eigene Darstellung

Für einen geeigneten Messvorgang wird das originale Controller-Board ausgebaut und mit einem mobilen Schraubstock festgehalten. Außerdem werden die Kabel angesteckt und über den Hauptschalter Spannung angelegt. Für die Messung wird der Ground-Anschluss des Tastkopfes an einen Pin mit Ground-Potenzial geklemmt, während die Prüfspitze des Tastkopfes auf den Ausgangspin gedrückt wird. In Abb. 5.5 ist der Messvorgang ersichtlich, bei dem zusätzlich das Signal für die umgekehrte Drehrichtung des Schrittmotors gemessen wird.

In Abb. 5.7 ist das Signal bei fortgeschrittener Zeit respektive bei ungefähr einem halben Vorschub dargestellt. Die gemessene Frequenz von aufgerundet 12 kHz dient der Evaluierung des Mikrocontrollers.

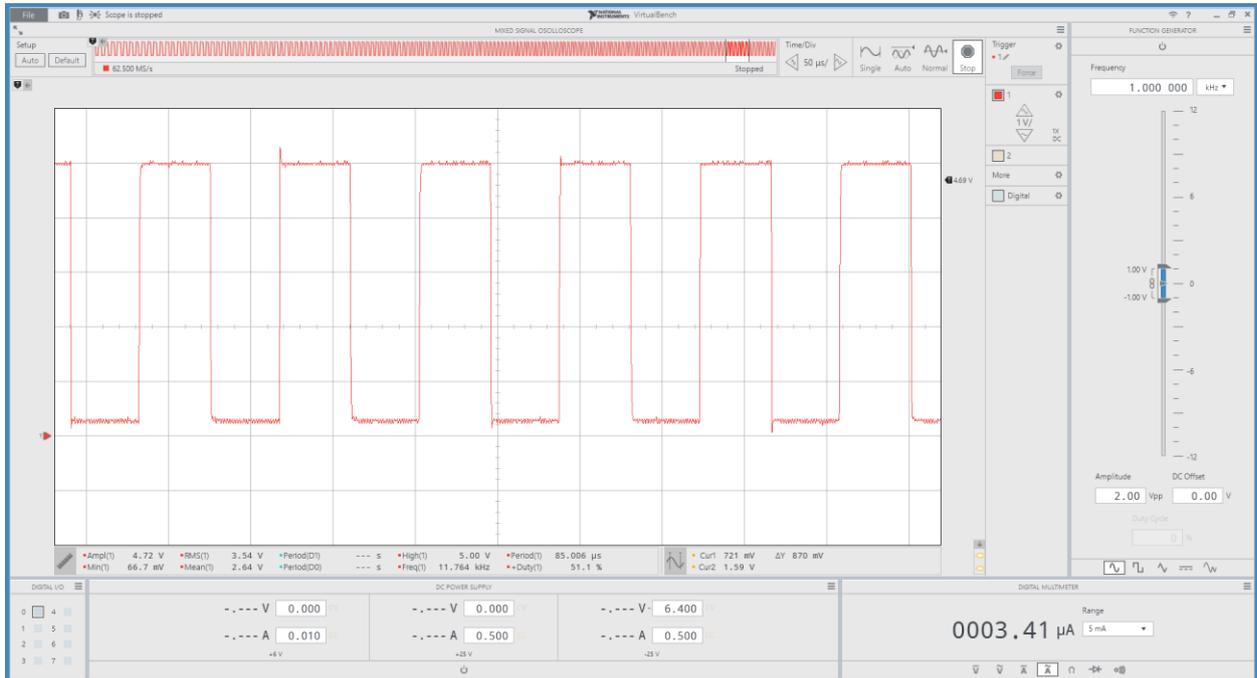


Abb. 5.7: Signal am Schrittmotor mit höchster Frequenz, Quelle: Eigene Darstellung

Da nach der Mitte eines Vorschubvorgangs die Frequenz des Rechtecksignals abnimmt, ist von einem geschmeidigen Geschwindigkeitsverlauf des Schrittmotors auszugehen. Durch die höher werdende Frequenz wird auch die Geschwindigkeit des Schrittmotors schneller, daher ist sie bei halber Zeit am höchsten.

Das Signal für die Richtung wird am zweiten Tastkopf gemessen und ist in Abb. 5.8 dargestellt. Im Produktionsprozess ist eine verkehrte Drehrichtung einzig für die Positionierung des Gurtbandes für die Einlegeposition des Roboters notwendig. Für den Messvorgang wird auf diesem Weg der Treiber des Schrittmotors für eine verkehrte Drehrichtung angesteuert und der Ausgang gemessen. Es zeigt, dass das Signal bei standardmäßiger Vorwärts-Richtung auf 0 Volt geschaltet ist. Bei verkehrter Drehrichtung wird das Signal auf logisch 1 gesetzt.

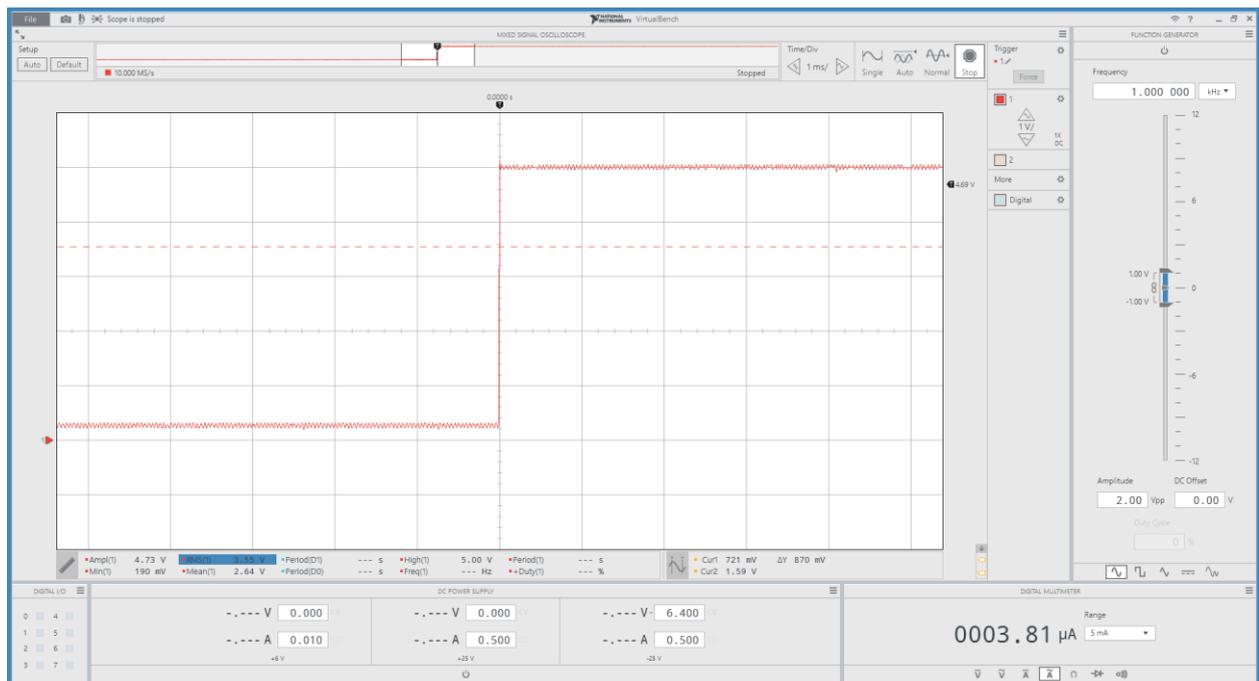


Abb. 5.8: Signal Drehrichtung Schrittmotor, Quelle: Eigene Darstellung

Lichtschranken und Lichttaster zur Prozessüberwachung

Bei der Messung mit einem Multimeter ist auffällig, dass die Spannung bei nicht angesteckten Kabeln ebenfalls 5 Volt beträgt. Das Verhalten lässt darauf schließen, dass der Ausgang des Sensors nicht wie erwartet den Eingang auf logisch 1 setzt, sondern eine andere Beschaltung am Controller-Board aufweist. Für eine Erklärung wird das Datenblatt des Sensors mit der darin enthaltenen Abbildung zur Beschaltung herangezogen. In Abb. 5.9 ist ersichtlich, dass der geschaltete Pin extern mit einer Last auf einen Spannungspegel zwischen 5 und 40 Volt Gleichspannung gelegt werden muss. Das bedeutet, der Ausgang des Sensors ist als „Open Collector“ ausgeführt. Beim Ansprechen wird der Pin über eine interne Transistorschaltung auf Ground-Potenzial gezogen. Um den Strom zu begrenzen ist ein vorgeschalteter Lastwiderstand notwendig.

Beim FS-V21 kann aus der Abb. 5.9 entnommen werden, dass für die Beschaltung des Sensors ein Lastwiderstand benötigt wird. Da der Pin beim Ansprechen des Sensors auf Ground gezogen wird, würde ohne eine Last dazwischen ein hoher Strom fließen und möglicherweise die Spannung einbrechen lassen. Daher wird ähnlich wie beim Advance Pin eine Schaltung mit einem Pull-Up-Widerstand realisiert, um den Stromfluss zu begrenzen.

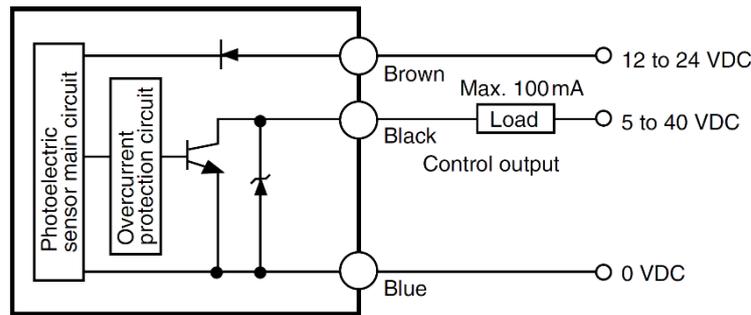


Abb. 5.9: FS-V21-Beschaltung, Quelle: Keyence, Online-Quelle [26.September.2020]

Am Eingang des Mikrocontrollers muss daher standardmäßig eine Spannung von 5 Volt anliegen, die beim Ansprechen eines Sensors auf Ground gezogen wird. Bei der Softwareprogrammierung muss am Eingang auf ein logisches 0 gewartet werden.

5.2 Anforderungen an den Controller

Für die Wahl eines geeigneten Mikrocontrollers sind die analytischen Erkenntnisse aus Kapitel 5.1 einzubinden.

Am bestehenden Controller-Board ist ein PIC18F6585 Mikrocontroller des Herstellers Microchip verbaut.

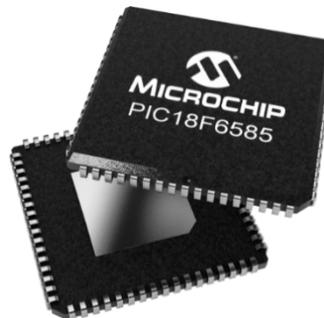


Abb. 5.10: Microchip PIC18F6585, Quelle: Microchip, Online-Quelle [29.Juni.2020]

Der Mikrocontroller funktioniert für die Anwendung in der Verpackungssteuerung zuverlässig und ist auch für die gewünschten Modifikationen prinzipiell geeignet. Durch das Fehlen des Programm-Codes ist der Schaltplan für eine neue Programmierung der gesamten Verpackungseinheit zwingend notwendig. Aufgrund des fehlenden Schaltplanes des Controller-Boards ist das Risiko hoch beim Ausmessen viel Zeit zu verlieren oder Fehler zu machen, wodurch es bei der Programmierung zu unerwarteten Problemen kommen kann. Außerdem ist ein Debugging des PIC18F6585 praktisch nicht anwendbar, da eine spezielle Schaltung für die Programmierung mit einem Sockel zur Kontaktierung aufgebaut werden müsste, um die Programmierung des Chips durchführen zu können. Daher wäre bei jedem Programm-Update ein umständlicher Flash-Vorgang anzuwenden.

Aus diesen Gründen wird die Verwendung eines neuen Mikrocontrollers für die Umsetzung des Projektes angestrebt. Durch Zählung der Pins der Molex-Stecker mit Abzug der Ground- und Versorgungspins sind 47 Ein-/Ausgänge erforderlich. Für ein erleichtertes Debugging bei der Programmierung sind weitere freie Ein-/Ausgänge wünschenswert.

Die eingebauten Sensoren (Lichtschranken) benötigen laut Datenblatt mindestens 5 Volt am Collector. Um zusätzlich benötigte Schaltungen am Controller-Board zu vermeiden, wird ein Controller benötigt, der mit mindestens 5 Volt an den Ein-/Ausgängen arbeitet.

Für die Versorgung des Mikrocontrollers stehen vom AC/DC-Converter im Steuerkasten 24 Volt DC zur Verfügung.

Aus der Analyse der Ausgangsfrequenz des Schrittmotor-Treibers im Kapitel 5.1 geht hervor, dass der Mikrocontroller die Ausgänge mit mindestens 12 kHz schalten können muss.

Wahl Arduino Mega Rev3

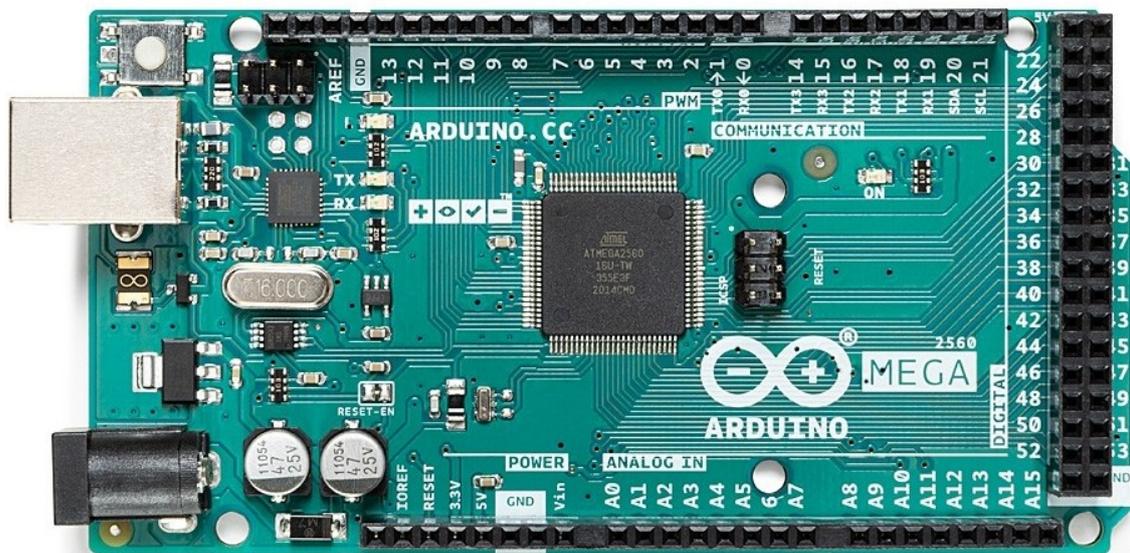


Abb. 5.11: Arduino Mega Rev3, Quelle: Arduino, Online-Quelle [22.August.2020]

Für die Wahl eines Arduino Mega Rev3 (aktuelle Version) sprechen mehrere Gründe, die aus den Analysen hervorgehen.

Von den recherchierten Mikrocontroller-Boards ist der Arduino der Einzige, der an den I/O Pins mit 5 Volt arbeiten kann. Zusätzlich kann der Arduino an diesen Pins bis zu 40 mA für saubere Schaltvorgänge und Störungsresistenz liefern. Die daraus resultierende Erleichterung für die Beschaltung und Ansteuerung der Komponenten bietet große Vorteile.

Der Arduino Mega besitzt serienmäßig 70 digitale I/O Pins und benötigt keine zusätzlichen Erweiterungen. Daher sind für die Ansteuerung von beispielsweise LEDs für ein erleichtertes Debugging oder visuelle Information im Störfall, ausreichend viele Reserve-Pins verfügbar. Andere Mikrocontroller-Boards, wie der Beagle Bone Black können ähnlich viele I/O Pins aufweisen, arbeiten aber mit maximal 3,3 Volt Spannung und niedrigeren Stromstärken.

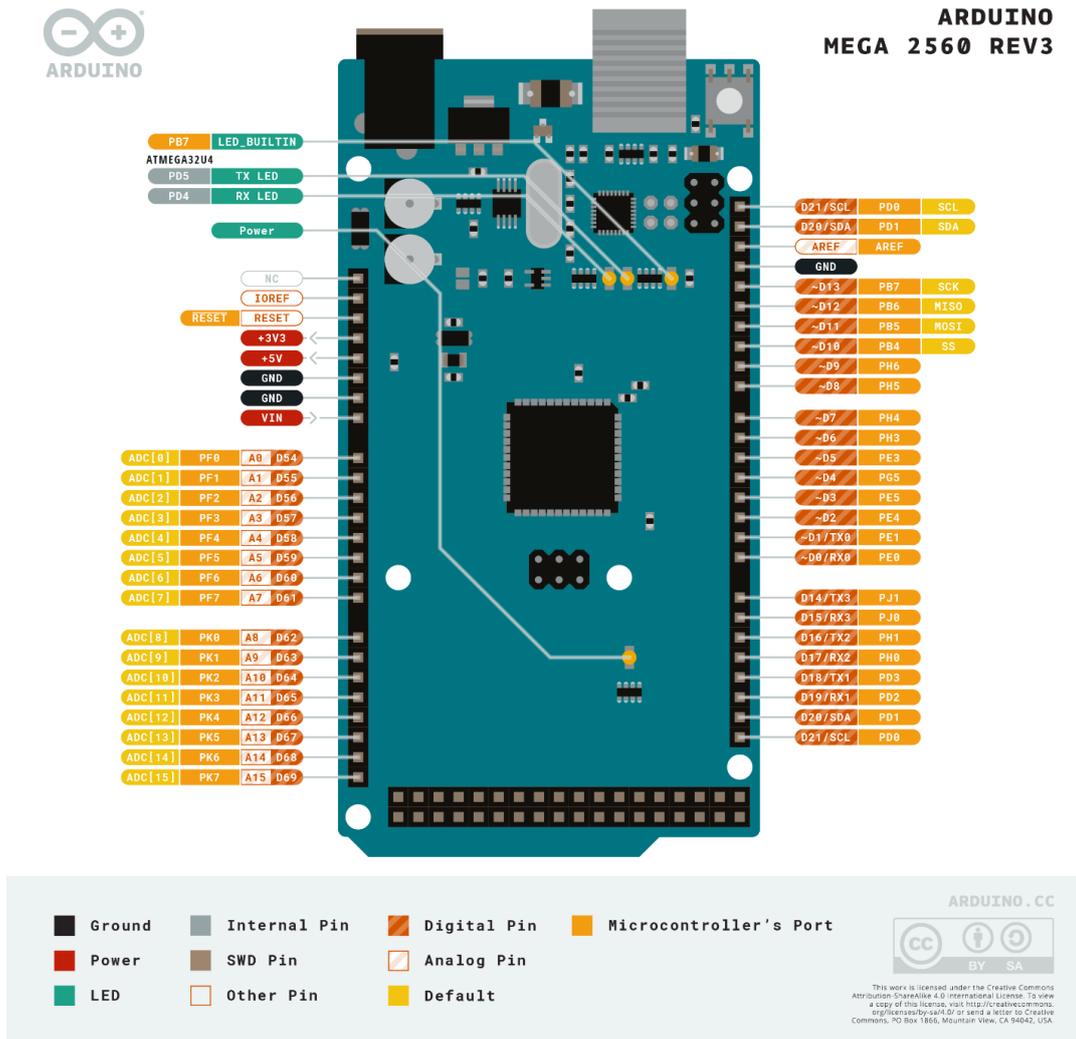


Abb. 5.12: Pinout Arduino Mega Rev3, Quelle: Arduino, Online-Quelle [22.August.2020]

Der Arduino wird weltweit neben dem Raspberry Pi am häufigsten verwendet und daraus resultiert ein breites Spektrum an Online-Support und Beispielprojekten. Das beschleunigt und erleichtert die Umsetzung der Projekte, da zusätzlich Funktionen über Libraries zur Einbindung zur Verfügung stehen.

Die Programmierung erfolgt über eine zur Verfügung gestellte Entwicklungsumgebung Namens ‚Arduino Integrated Development Environment‘ (IDE). Die Software kann auf verschiedenen Betriebssystemen, wie Windows, MacOS oder Linux betrieben werden. Die Programmierung erfolgt in einer C- beziehungsweise C++ orientierten Programmiersprache.

Der Arduino Mega arbeitet mit 16 MHz und einer Performance von bis zu 16 MIPS, welche für die Ansteuerung des Schrittmotoren-Treibers ausreichend ist, wobei die Konkurrenz dabei erheblich mehr Ressourcen zur Verfügung hat. Im spezifischen Anwendungsgebiet sind höhere Rechenleistungen nicht relevant.

Die im Steuerkasten verfügbaren 24 Volt sind für alle recherchierten Mikrocontroller zu viel und müssen beim Arduino optimalerweise auf eine Spannung zwischen 7 und 12 Volt reduziert werden.

6 HARDWARE-AUFBAU

6.1 Versuchsaufbau

Ein Versuchsaufbau ist sinnvoll, wenn theoretisch durchdachte Konzepte und Ideen überprüft werden sollen. Das ist speziell bei neuen Projekten vorteilhaft, wo keine Erfahrungen vorhanden sind und nicht von einem funktionierenden Konzept ausgegangen werden kann. Ein Ziel von Versuchsaufbauten ist es, Kosten für Spezialanfertigungen zu sparen, die sich zu einem späteren Zeitpunkt als fehlerhaft erweisen könnten.

Ein häufig angewendeter Weg zur praktischen Umsetzung ist der Aufbau der Schaltung auf einer Lochrasterplatine. Für eine vereinfachte Verlötlung sind meist Bauteile für eine Durchsteckmontage in Verwendung.

Displayansteuerung

Für die Anzeige und die Menüführung der Verpackungssteuerung wird das im Steuerkasten bestehende LC-Display „LCD 20x4 DMC-20481NY-LY“ verwendet.



Abb. 6.1: Originales LC-Display, Quelle: Digi-Key, Online-Quelle [16.Juli.2020]

Ein Grund dafür ist, dass für die Änderungen einzig das Controller-Board ausgetauscht werden soll. Das führt einerseits dazu, dass es für den Operator keine Änderungen bei der Bedienung gibt. Zum anderen können dadurch Änderungen bei schriftlichen Unterlagen wie Arbeitsanweisungen, Bedienungsanleitungen und anderen freigegebenen Dokumenten vermieden werden.

Da das Display das gesamte Feld des verglasten Bereiches abdeckt und die angezeigten Informationen am Display notwendig sind, wäre ein Umstieg auf ein anderes Display nur aus Designgründen angedacht, aber nicht funktionsrelevant. In Abb. 6.2 ist ersichtlich, dass der Anzeigebereich auf dem LC-Display mit einer Größe von 20x4 Zeichen für 6 benötigte Parameter der Steuerung ausgenutzt ist.



Abb. 6.2: Verglaster Anzeigebereich, Quelle: Eigene Darstellung

Pinbelegung:

Pin	Symbol	Level	Function
1	VSS	—	Power Supply (0V, GND)
2	VCC	—	Power Supply for Logic
3	VEE	—	Power Supply for LCD Drive
4	RS	H / L	Register Select Signal
5	R/W	H / L	Read/Write Select Signal H: Read L: Write
6	E	H, H→L	Enable Signal
7	DB0	H / L	Data Bus Line / Non-connection at 4-bit operation
8	DB1	H / L	Data Bus Line / Non-connection at 4-bit operation
9	DB2	H / L	Data Bus Line / Non-connection at 4-bit operation
10	DB3	H / L	Data Bus Line / Non-connection at 4-bit operation
11	DB4	H / L	Data Bus Line
12	DB5	H / L	Data Bus Line
13	DB6	H / L	Data Bus Line
14	DB7	H / L	Data Bus Line
15	LED: Anode	—	LED Anode Terminal
16	LED: Cathode	—	LED Cathode Terminal

Tab. 6.1: Pinbelegung LCD, Quelle: Digi-Key, Online-Quelle [16.Juli.2020]

Die Ansteuerung des Displays wird anhand der in Tab. 6.1 dargestellten Pinbelegung im 4-Bit Modus implementiert. Die Versorgung des Displays wird über 5 Volt Pins des Arduino realisiert und die Intensität der Hintergrundbeleuchtung sowie Kontrast mittels Widerstände über Versuche ermittelt.

Laut Datenblatt benötigt das Display eine Spannungsversorgung von 5 Volt mit einer Toleranz von ± 10 Prozent. Im Versuchsaufbau wird beobachtet, dass die Ansteuerung des Displays sporadisch nicht funktioniert. Eine umfangreiche Analyse der Ursache ergibt, dass zwischen den Versorgungspins 1 und 2 ein Kriechstrom fließt, durch den die Spannung um 0,15 auf 4,85 Volt, gemessen am kalibrierten Multimeter, absinkt. Die Versorgungsspannung liegt damit noch in der Toleranz, die im Datenblatt mit ± 10 Prozent angegeben wird. Nachdem der Kriechstrom durch Nachlöten unterbunden wird und die Versorgungsspannung bei 5 Volt liegt, tritt das Problem nicht mehr auf.

Versuchsaufbau LCD:

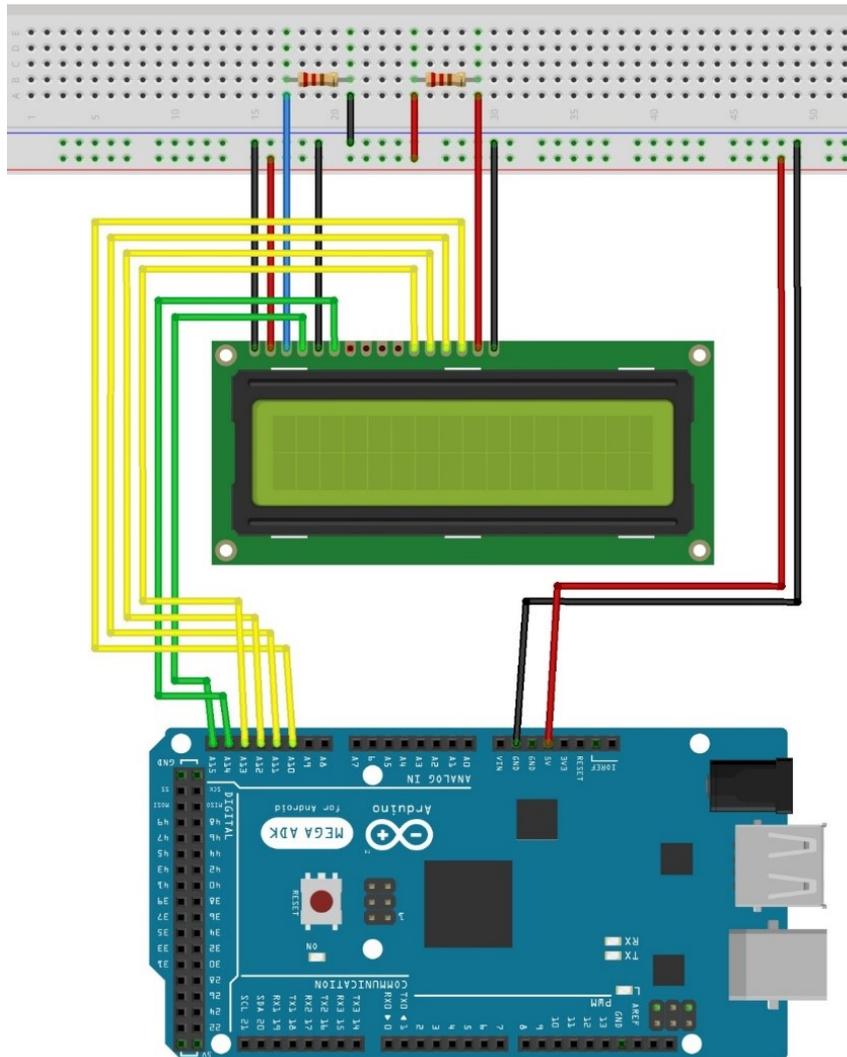


Abb. 6.3: Versuchsaufbau LCD, Quelle: Eigene Darstellung

Reale Anzeige:



Abb. 6.4: Hello World! auf dem Versuchsaufbau, Quelle: Eigene Darstellung

In Abb. 6.3 sind auf dem dargestellten Steckbrett zwei Widerstände für die Anzeigeeinstellungen des Displays abgebildet. Der linke Widerstand dient zur Einstellung des Kontrastes und der rechte Widerstand zur Einstellung der Intensität der Hintergrundbeleuchtung.

Der Versuchsaufbau zeigt, dass der optimal dimensionierte Widerstand für die Einstellung des Kontrastes, bei einem Wert von 470Ω liegt. Die Hintergrundbeleuchtung kann über einen niederohmigen Widerstand im Bereich von 22Ω optimal eingestellt werden. Bei anders dimensionierten Einstellungen kann die Anzeige unlesbar werden.

Nach der erfolgreichen Ansteuerung und Anzeige des Displays kann der Versuchsaufbau um weitere Funktionen erweitert werden. Mithilfe des funktionierenden Displays ist das Debugging durch Anzeige von Parametern oder Statusinformationen erleichtert.

Serielle Kommunikation

Ein wesentliches Ziel des Projektes ist es, die Kommunikation zwischen Roboter-Steuerung und Verpackungs-Steuerung zu erweitern. Dafür wird die bestehende serielle Schnittstelle RS-232 verwendet und mit den dafür vorgesehenen Pins des Arduinos verbunden.

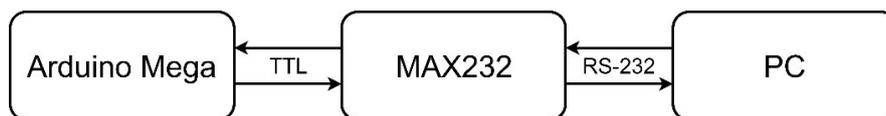


Abb. 6.5: Schematische Darstellung der Kommunikation, Quelle: Eigene Darstellung

Am Arduino stehen jeweils vier Pin-Paare für jeweils Senden (TX) und Empfangen (RX) zur Verfügung. Dabei ist zu beachten, dass die Verbindung vom PC zum Mikrocontroller nicht direkt erfolgen kann, da bei RS-232 Schnittstellen bis zu ± 15 Volt auftreten können. Da der Arduino mit 5 Volt an den Ein-/Ausgängen arbeitet, würde eine direkte Verbindung den Mikrocontroller beschädigen. Zusätzlich muss das Signal invertiert werden.

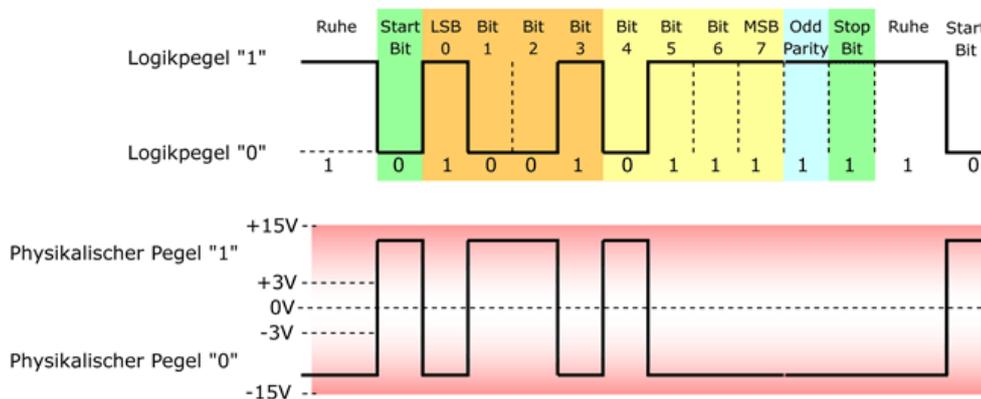


Abb. 6.6: RS232/TTL Datenübertragung, Quelle Kompodium, Online-Quelle [21.Oktober.2020]

Aus diesem Grund wird eine Schaltung für die benötigte Pegelanpassung auf der Lochrasterplatine aufgebaut.

Schaltungsaufbau:

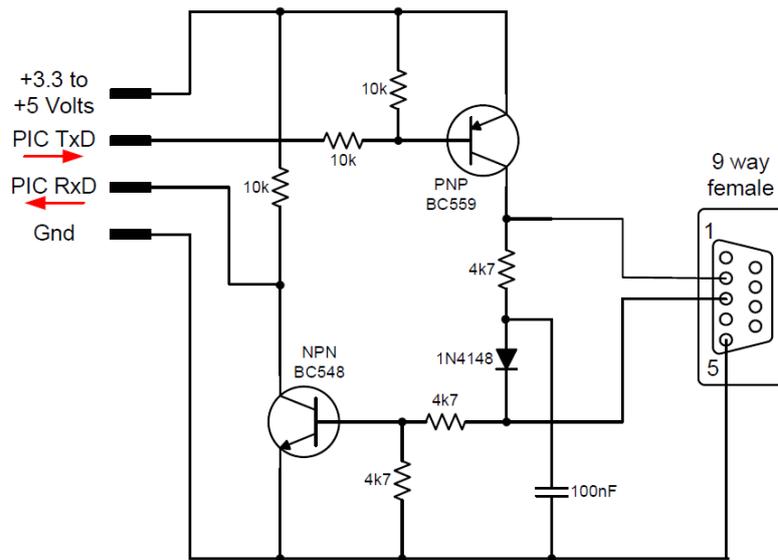


Abb. 6.7: Schaltung RS-232/TTL, Quelle: Picprojects, Online-Quelle [29.September.2020]

Durch Messung der Signale am Oszilloskop kann verifiziert werden, ob die Schaltung ihren Zweck erfüllt.

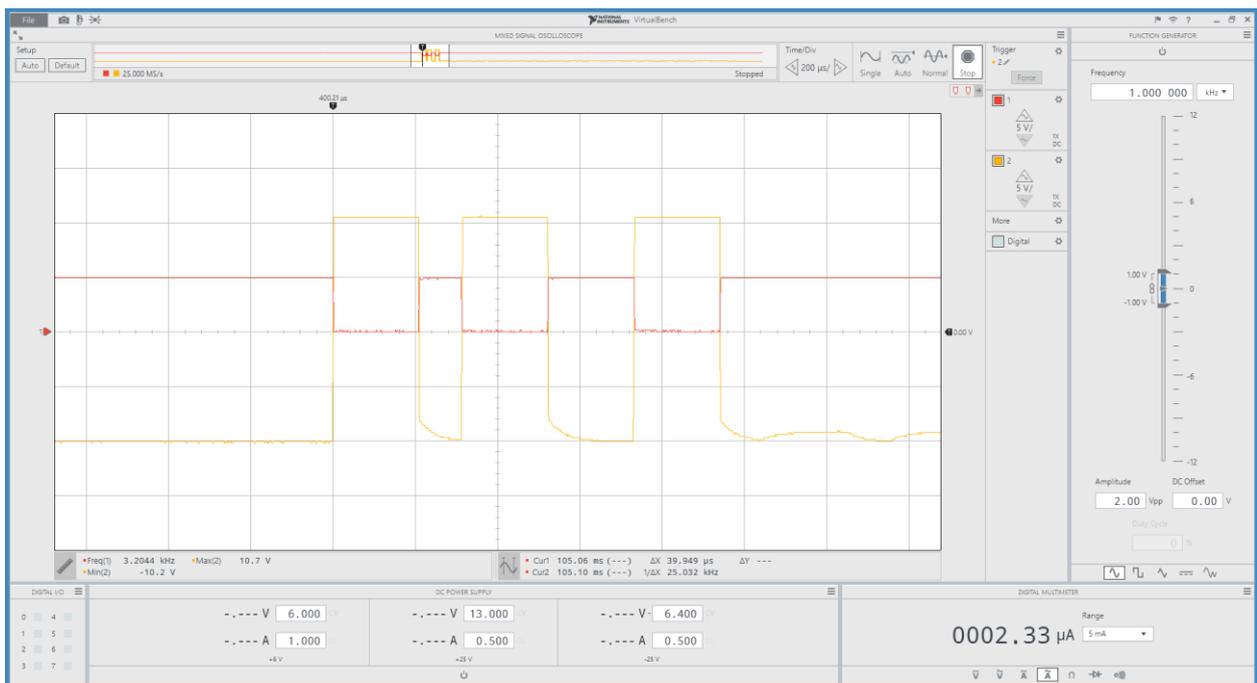


Abb. 6.8: RS232/TTL Signalmessung, Quelle: Eigene Darstellung

Am Kanal 1 (rot) wird das TTL Signal des Mikrocontrollers und am Kanal 2 (gelb) das RS232 Signal des PCs dargestellt. Durch Erweiterung der Funktionen in der Software des Arduinos und Verwendung des Programmes „Tera Term“ für eine serielle Verbindung am angeschlossenen Port, kann die Kommunikation ausprobiert werden.

Aus Performance-Gründen wie beispielsweise Flankensteilheit und daraus resultierenden höheren Übertragungsgeschwindigkeiten wird die Schaltung durch einen optimierten Pegelkonverter ersetzt. Der „Maxim MAX232“ ist einer der bekanntesten Sendeempfänger dafür.

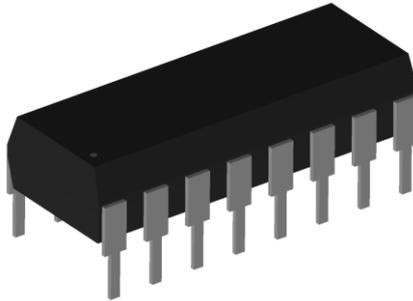


Abb. 6.9: Maxim MAX232, Quelle: Eigene Darstellung

Der MAX232 verfügt über eine proprietäre Transmitter-Ausgangsstufe mit niedrigem Verlustfaktor, was gute RS232-Performance mit 3 – 5,5 V ermöglicht. Der IC benötigt fünf Kondensatoren mit einer Kapazität von 1µF. Über den MAX232 kann mit einer Geschwindigkeit von bis zu 120 kbps kommuniziert werden. Außerdem gibt es zwei Empfänger-Pins und zwei Treiber-Pins, worüber parallel zwei unterschiedliche Datenübertragungen stattfinden können.

Beschaltung MAX232:

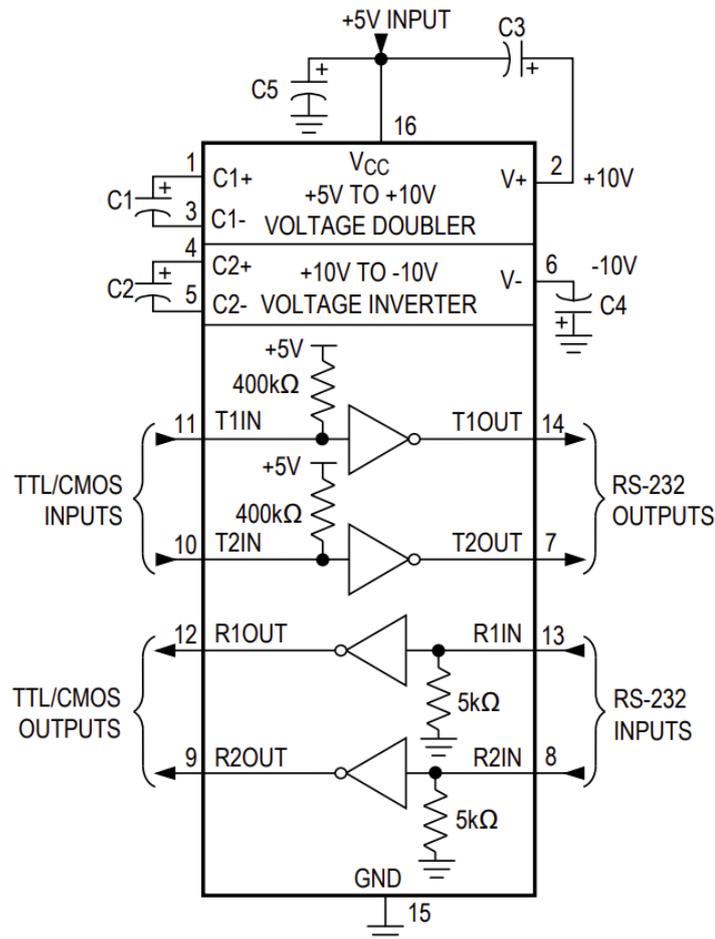


Abb. 6.10: MAX232 Beschaltung, Quelle: Maxim Integrated (2019), Online-Quelle [28.September.2020]

Nach stetigen Erweiterungen am Versuchsaufbau befindet sich der Arduino in einer Reboot-Schleife. Außerdem ist eine hohe Wärmeentwicklung beim Arduino festzustellen. Es zeigt sich, dass die Stromaufnahme des Versuchsaufbaus über die Versorgungspins des Arduinos zu hoch ist und die Spannung einbricht. Abhilfe dafür schafft ein zusätzlicher DC/DC-Converter auf dem Controller-Board, der die Spannung von 24 V auf 5 V bei einem Dauerstrom von bis zu 1 A regelt. Dadurch wird das Controller-Board ausreichend versorgt, der Spannungsregler auf dem Arduino-Board entlastet und die Wärmeentwicklung deutlich reduziert.

Versuchsaufbau:

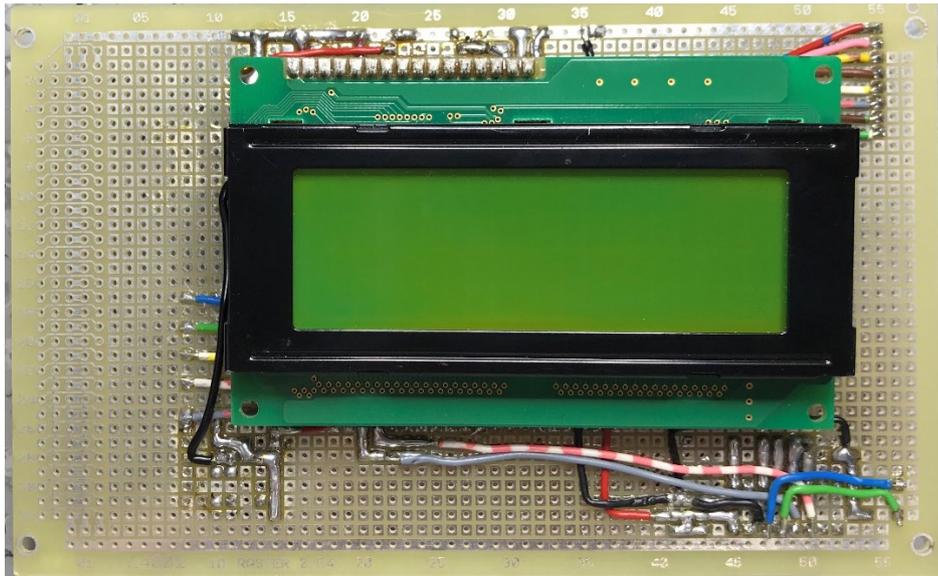


Abb. 6.11: Vorderseite Versuchsaufbau, Quelle: Eigene Darstellung

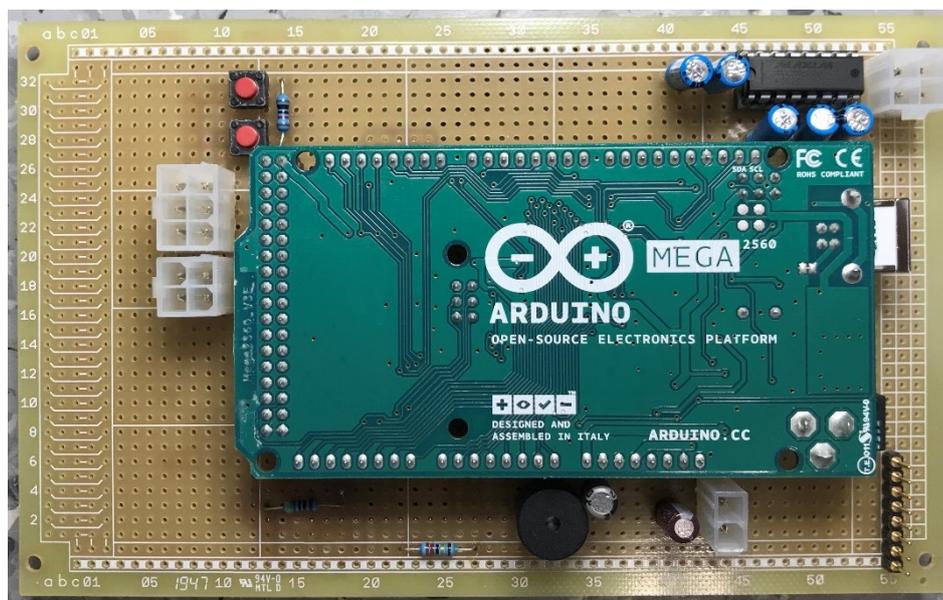


Abb. 6.12: Rückseite Versuchsaufbau, Quelle: Eigene Darstellung

6.2 Platinendesign

Das Controller-Board dient zur mechanischen Befestigung des Displays und des Mikrocontrollers. Außerdem dient es zur Verbindung der einzelnen Komponenten der Verpackungseinheit mit dem Arduino über Molex-Steckverbindungen. Dafür sind zusätzliche Schaltungen für Versorgung, Beschaltung (beispielsweise Pull-Ups) der Ein- und Ausgänge notwendig. Dementsprechend werden die Erkenntnisse aus dem Versuchsaufbau für die Schaltungsentwicklung angewendet. Des Weiteren sind LEDs für visuelle und ein Piezo-Lautsprecher für akustische Signalgebung verbaut.

Versorgung

Im Steuerkasten stehen 24 Volt vom AC/DC-Converter zur Verfügung. Der technischen Spezifikation des Arduino Mega Rev3 ist zu entnehmen, dass die Limits für die Versorgungsspannung bei 6 und 20 Volt liegen und eine Spannung zwischen 7 und 12 Volt vorgeschlagen wird. Dafür wird der DC/DC-Converter „TRACOPOWER TSR 2“ verwendet, der bei anliegender Eingangsspannung zwischen 15 und 36 Volt eine Ausgangsspannung von 12 Volt mit bis zu 2 Ampere liefert.

Für die Versorgung des Controller-Boards wird aufgrund der Erkenntnisse im Versuchsaufbau der DC/DC-Converter „RS Pro K7805M-1000R3“ verwendet, der eine Eingangsspannung zwischen 8 und 36 Volt auf 5 Volt regelt.

Der Arduino kann an I/O Pins maximal 40 mA und an den Versorgungspins maximal 50 mA Strom liefern. Damit ist der Energiebedarf am Limit und muss von außen unterstützt werden. Der Grund, warum ein DC/DC-Konverter zur Anwendung kommt ist, dass im Gegensatz zu Spannungsteilern mit Widerständen ein hoher Wirkungsgrad von 96 % bei der Spannungsregulierung erreicht werden kann, während bei Spannungsteilern die überschüssige Energie in Wärme umgewandelt wird.

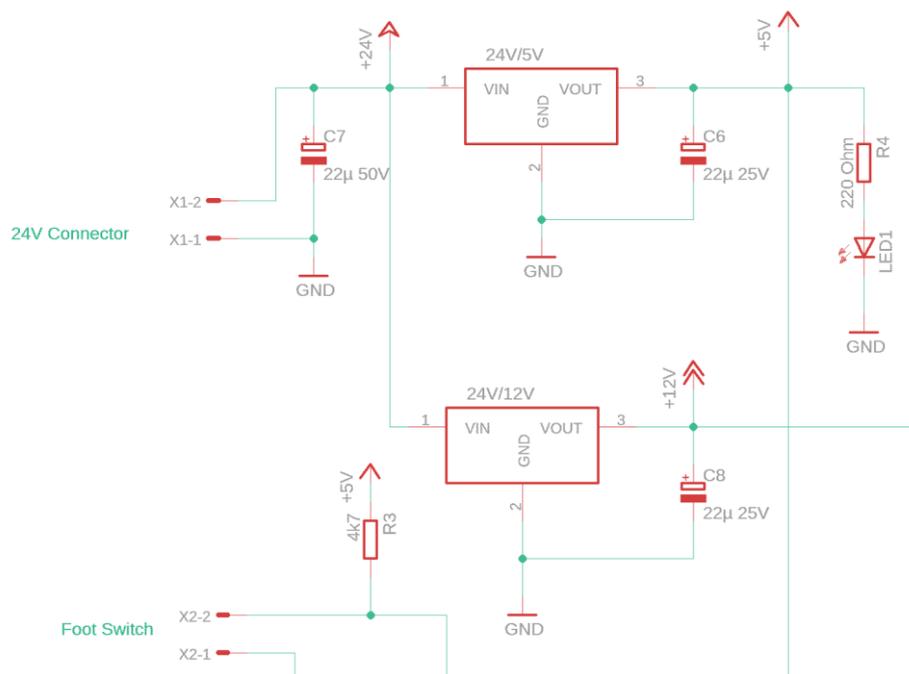


Abb. 6.13: DC/DC-Konverter-Schaltung, Quelle: Eigene Darstellung

Für die Spannungsstabilisierung ist die dem Datenblatt zu entnehmende Schaltung, wie in Abb. 6.14 dargestellt, realisiert.

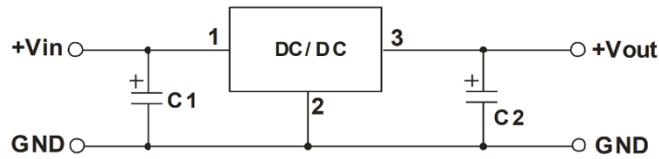


Abb. 6.14: Beschaltung DC/DC-Converter, Quelle: RS Components, Online-Quelle [15.Oktober.2020]

Arduino

Für die Erstellung des Schaltplanes mit anschließendem Layouten ist ein passendes Symbol mit Anschlüssen, sowie ein korrekter Footprint vom Arduino Mega Rev3 notwendig. Dafür wird ein Package einer früheren Version des Arduino Mega abgeändert, um ein geeignetes Package der aktuellen Revision 3 zu erhalten, wie in Abb. 6.15 und Abb. 6.16 abgebildet.

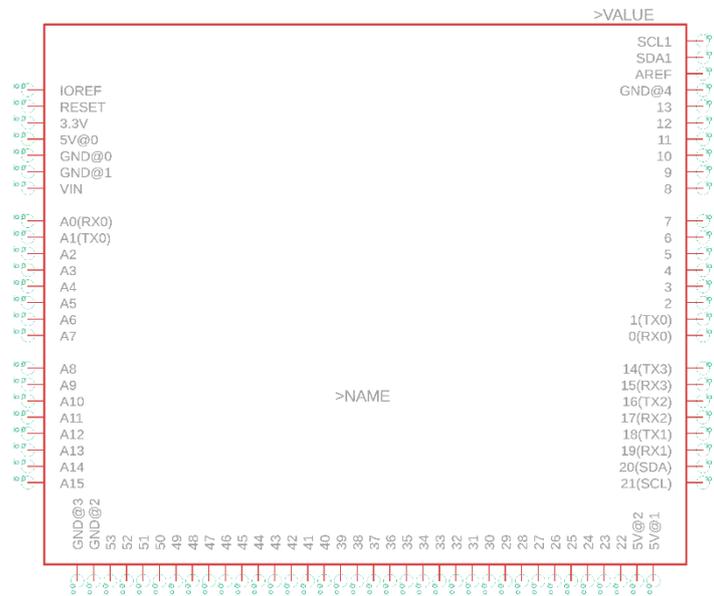


Abb. 6.15: Arduino Mega Rev3 Symbol, Quelle: Eigene Darstellung

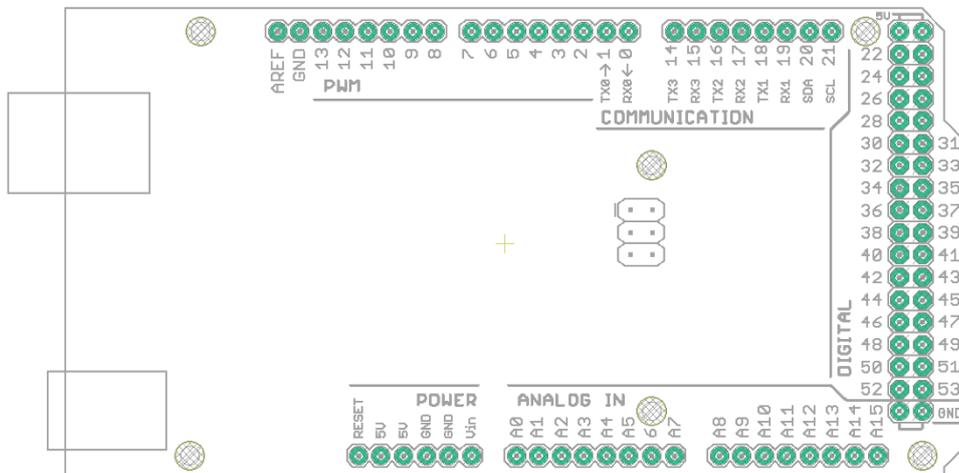


Abb. 6.16: Arduino Mega Rev3 Footprint, Quelle: Eigene Darstellung

Für die Anordnung der Molex-Steckverbindungen für die Ein-/Ausgänge muss auf die Kabellängen und den zur Verfügung stehenden Platz im Steuerkasten geachtet werden.

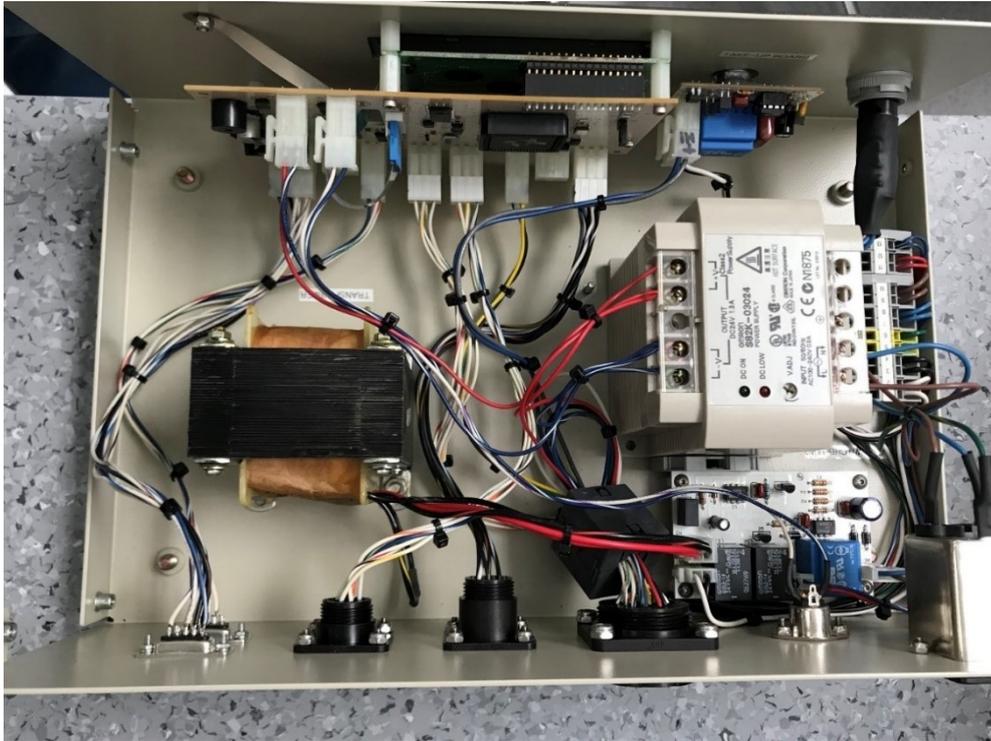


Abb. 6.17: Steuerbox von innen, Quelle: Eigene Darstellung

Ein besonderes Augenmerk ist dabei auf den Anschluss des Tastenfeldes zu legen, da dieser 8-polige Stecker ein kurzes Kabel besitzt und im bestehenden Board mittig in einem dafür vorgesehenen Ausschnitt durchgeführt ist. Aufgrund der Höhe des 8-poligen 1-reihigen Anschlusses für das Tastenfeld ist eine Platzierung zwischen Adapterplatine und dem aufgesteckten Arduino nicht möglich. Durch die Höhe des USB-Anschlusses und des Arduino-Versorgungssteckers „DC21MMX“ ist ein Ausschnitt in der Adapterplatine ebenfalls nicht zielführend. Aus diesem Grund ist am Seitenrand eine Aussparung in die Adapterplatine designet, wodurch das unter der Adapterplatine herausgeführte Kabel einen kürzeren Weg hat und angesteckt werden kann.

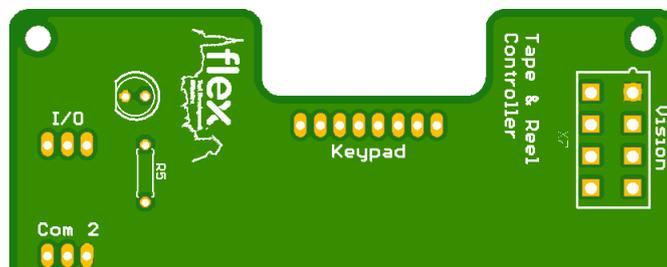


Abb. 6.18: Aussparung am Platinenrand, Quelle: Eigene Darstellung

Montage

Für die Montage des Displays ist die Position des Displays durch das zugeschnittene Schauglas im Gehäuse der TM50 vorgegeben. Durch bestehende Gewindestifte des bestehenden Controller-Boards am

Gehäuse, sind die Befestigungspunkte der Adapterplatine ebenfalls festgelegt. Zusätzlich müssen die Befestigungspunkte des Arduino beim Design der Adapterplatine berücksichtigt werden. In Abb. 6.19 ist die designte Adapterplatine mit den beim Routing berücksichtigten Bohrungen für die Befestigungen ersichtlich.

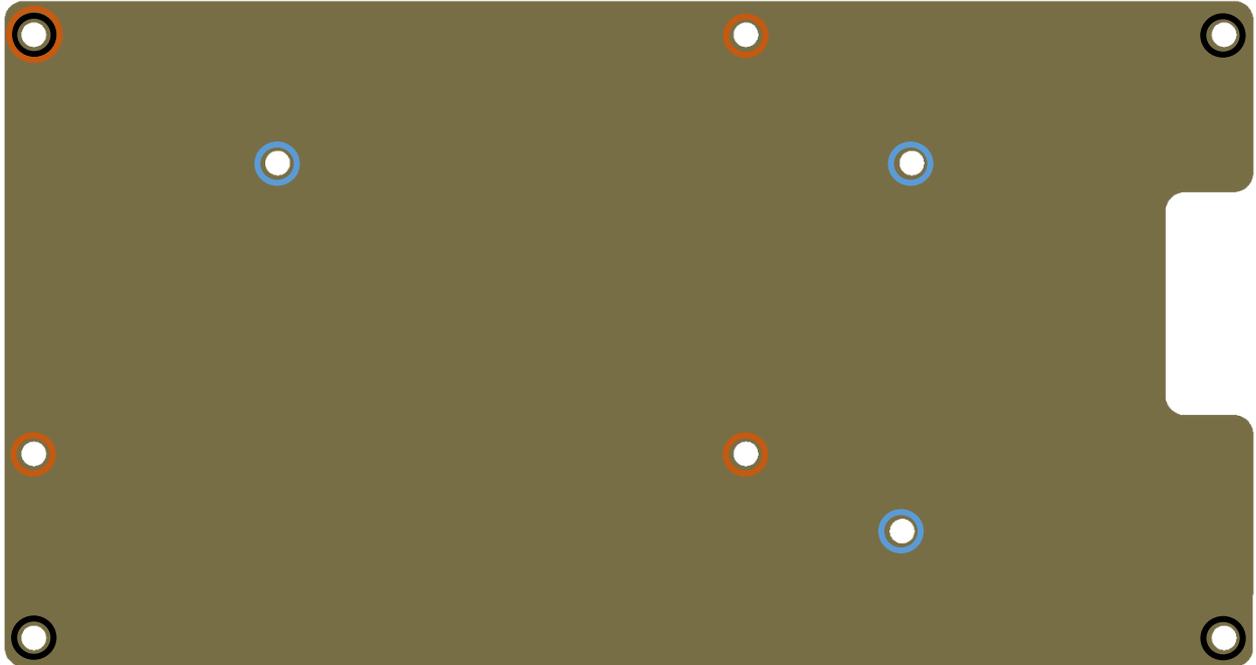


Abb. 6.19: Befestigungspunkte Controller-Board, Quelle: Eigene Darstellung

- Befestigungspunkte für Controller-Board
- Befestigungspunkte für Arduino Mega Rev3
- Befestigungspunkte für LC-Display

Anordnung:

Wie in Abb. 6.17 ersichtlich, müssen die Steckverbindungen des Controller-Boards auf derselben Seite wie am Steuerkasten platziert sein. Sonst kann es passieren, dass die Kabel aufgrund des dazwischen liegenden Transformators zu kurz sind.

In Abb. 6.20 ist das designte Platinenlayout mit einer passenden Anordnung der Molex-Steckverbindungen dargestellt. Da beim Arduino die meisten Pins für digitale Ein- / Ausgänge verwendbar sind, ist es möglich, die den Steckern in der Nähe befindlichen I/Os zu verwenden. Bei den restlichen vorgegebenen Pins, wie beispielsweise Kommunikation oder Tastenfeld, ist die Platzierung mit der Kabellänge abzustimmen. Um das Anstecken der Stecker am dafür vorgesehenen Anschluss zu erleichtern, sind am Controller-Board zahlreiche Beschriftungen vorgesehen. Für die Stecker ist die Benennung der Anschlüsse auf der Außenseite des Steuerkastens auf das Board gedruckt. Zusätzlich ist neben der Beschriftung des Boards und dem Flex-Logo der Footprint des Arduinos aufgedruckt, um das Debugging bei anschließender Softwareprogrammierung zu erleichtern.

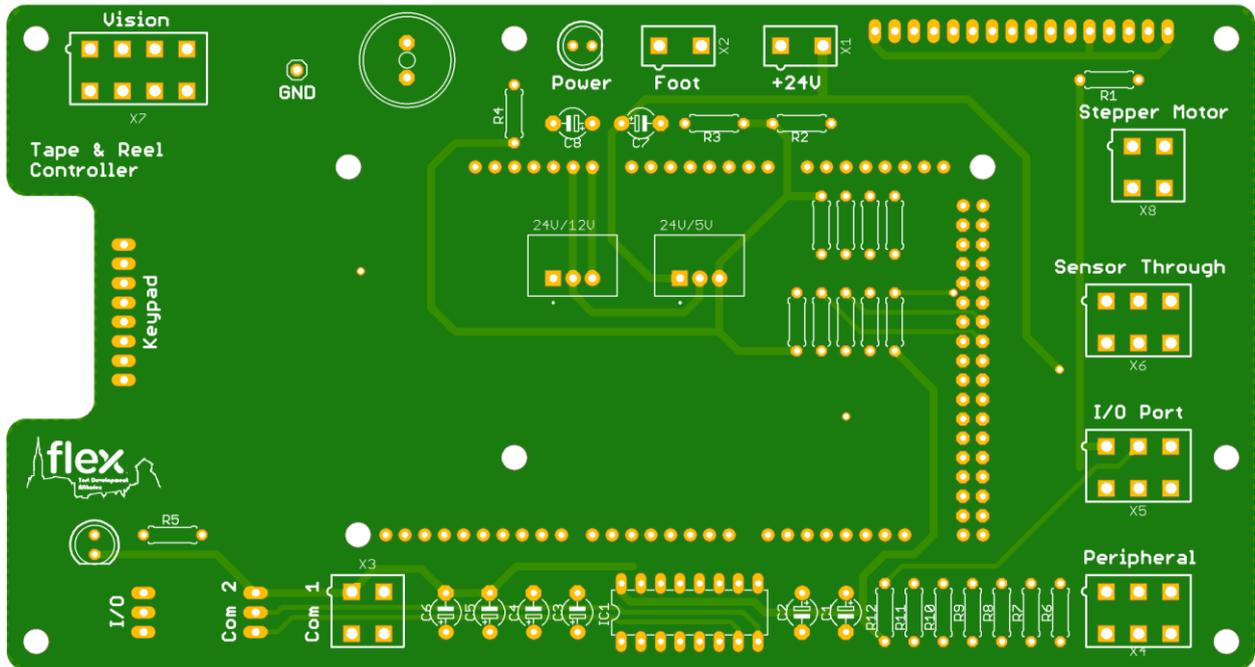


Abb. 6.20: Platinendesign Bottom Layer, Quelle: Eigene Darstellung

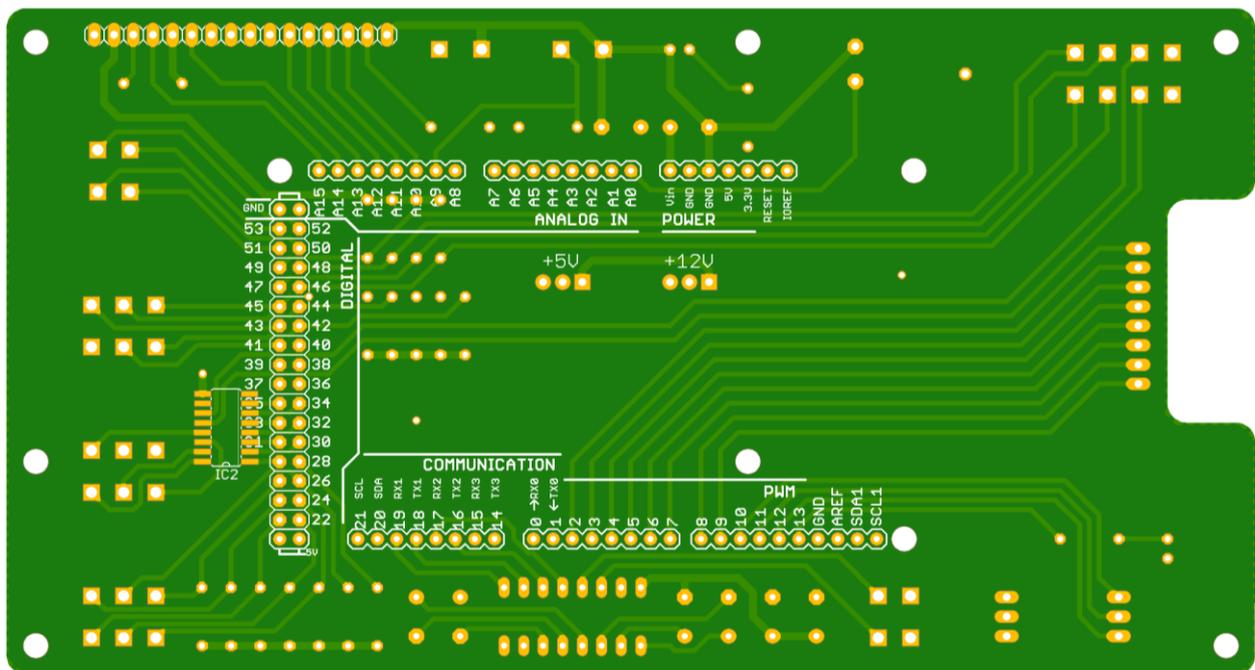


Abb. 6.21: Platinendesign Top Layer, Quelle: Eigene Darstellung

Bestücktes Controller-Board

In Abb. 6.22 ist das voll bestückte Controller-Board dargestellt. Das Board ist als Dual-Layer ausgeführt, wobei an beiden Layern (Bottom und Top) die Kontaktpads angebracht und durchkontaktiert sind. Dadurch ist es möglich an beiden Seiten Bauteile zu platzieren und diese zu verlöten. Über das gesamte Board ist auf beiden Layern eine Massefläche gezogen, womit das Layouten durch Reduktion von Leitungen erleichtert und der Querschnitt der Massefläche vergrößert wird. Zusätzlich werden Störungen durch

elektrische Felder verringert. Der MAX232-Chip ist auf einem Sockel steckbar montiert, um bei Bedarf leicht ausgetauscht zu werden. Die verwendeten Kondensatoren sind für Spannungen bis zu 50 Volt ausgelegt und daher überdimensioniert und störungsresistent. Die grüne Power-LED leuchtet sobald das Controller-Board mit Spannung versorgt ist, um beim Öffnen des Steuerkastens gegebenenfalls auf anliegende Spannung aufmerksam zu machen.

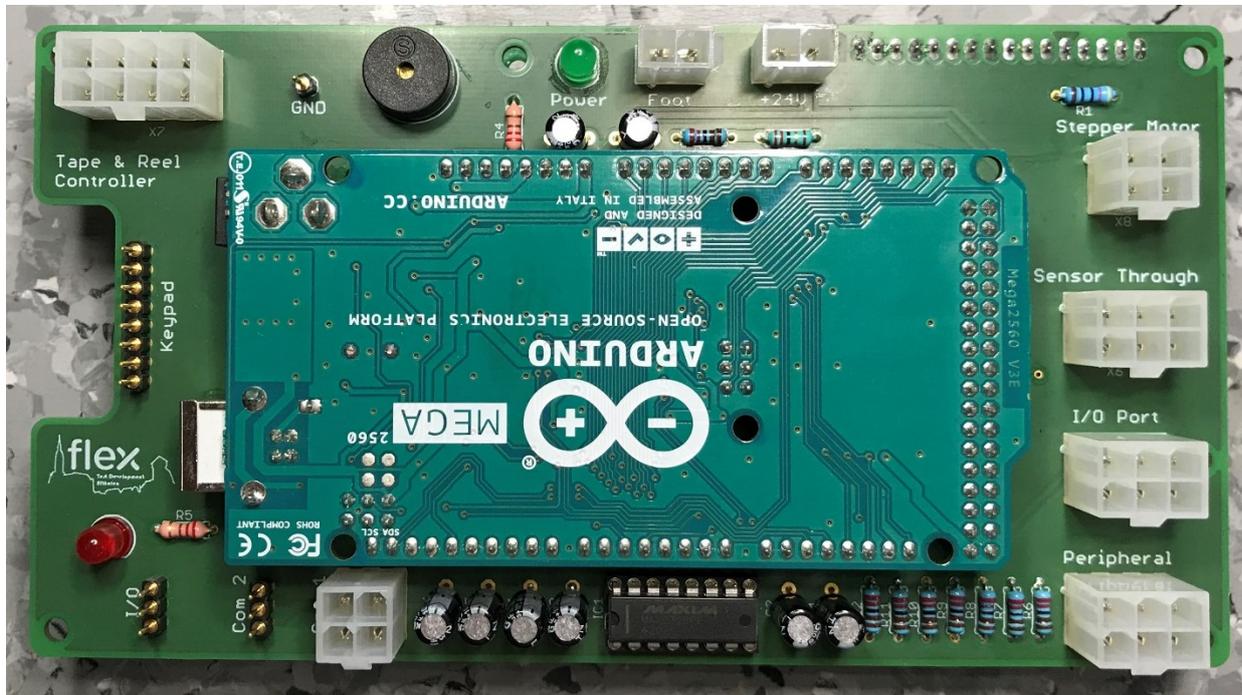


Abb. 6.22: Bestückte Adapterplatine Vorderseite, Quelle: Eigene Darstellung

An der Rückseite des Controller-Boards kann das Display über die Kontaktleiste angesteckt werden.

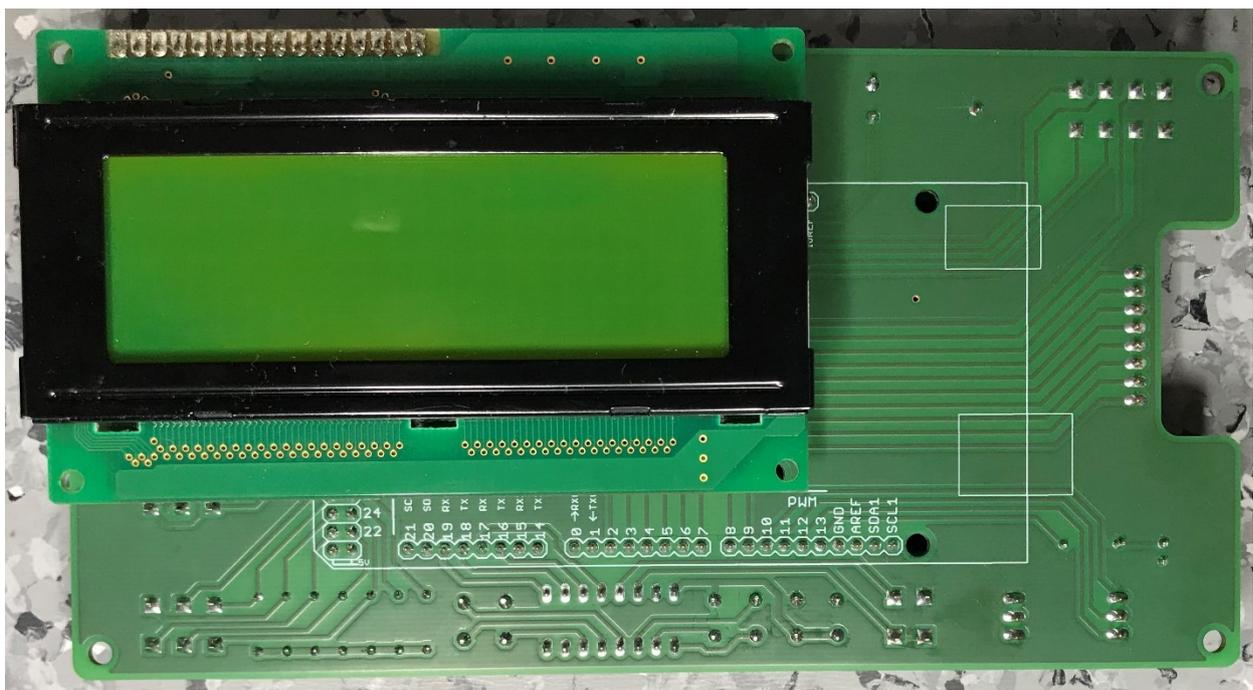


Abb. 6.23: Bestückte Adapterplatine Rückseite, Quelle: Eigene Darstellung

7 SOFTWARE-IMPLEMENTIERUNG

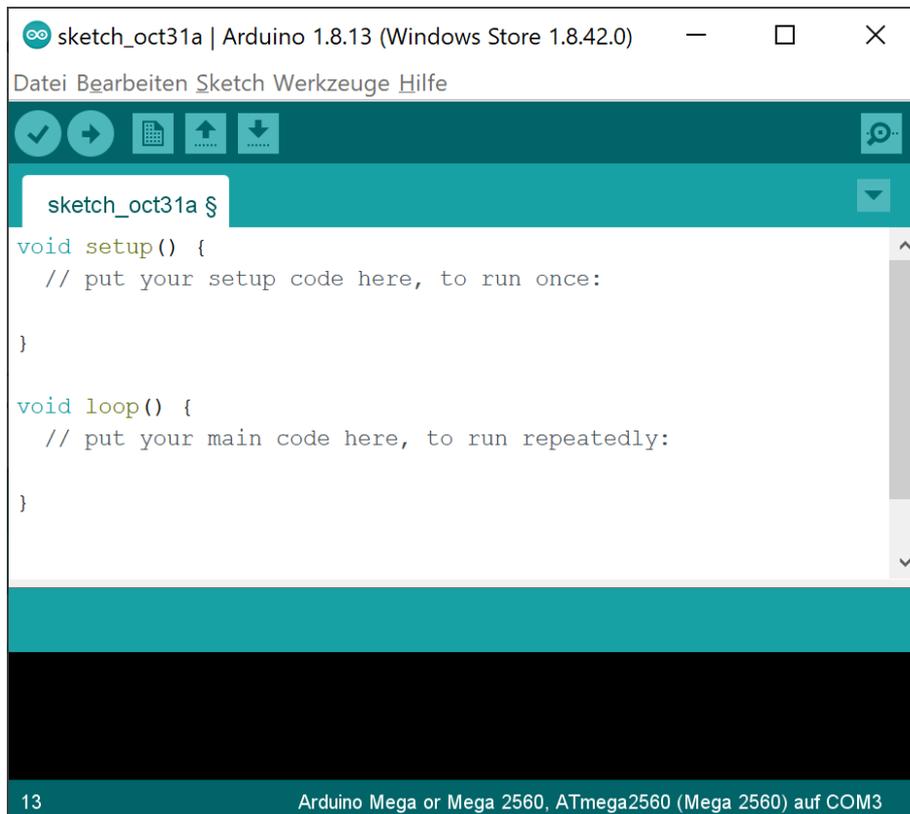
Die Programmierung erfolgt mittels der von Arduino zur Verfügung gestellten Software Arduino IDE 1.8.13.

Die dafür notwendige Programmiersprache ist C beziehungsweise C++ basiert und beinhaltet fertige Bibliotheken für Funktionen.

Arduino IDE beinhaltet einen Code-Editor, in dem der Programmcode geschrieben wird.

Für ein lauffähiges Programm werden mindestens zwei Funktionen benötigt:

- `setup()` – Diese Funktion wird nur einmal beim Start des Programms (Einschalten des Arduino) abgearbeitet. Hier sind alle Einstellungen wie die Deklaration der Ein- und Ausgänge und Initialisierungen vorzunehmen.
- `loop()` – Diese Schleife wird nach Abarbeitung der `Setup()`-Funktion zyklisch abgearbeitet bis der Arduino wieder von der Spannung getrennt wird. Hier werden alle Funktionen ausprogrammiert und je nach Zustand abgearbeitet.



The image shows a screenshot of the Arduino IDE code editor. The window title is "sketch_oct31a | Arduino 1.8.13 (Windows Store 1.8.42.0)". The menu bar includes "Datei", "Bearbeiten", "Sketch", "Werkzeuge", and "Hilfe". The toolbar contains icons for check, run, upload, and download. The code editor shows the following code:

```
sketch_oct31a §
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

The status bar at the bottom indicates "13" and "Arduino Mega or Mega 2560, ATmega2560 (Mega 2560) auf COM3".

Abb. 7.1: Arduino IDE Schleifen, Quelle: Eigene Darstellung

Programmaufbau:

Für die Programmierung sind vier inkludierbare Bibliotheken mit vorgefertigten Funktionen hilfreich:

- `<LiquidCrystal.h>` für die Display-Ansteuerung
- `<Keypad.h>` für das Tastenfeld
- `<EEPROM.h>` für den nicht-flüchtigen Speicher
- `<Stepper.h>` für den Schrittmotor

In der originalen Steuerung gibt es fünf Parameter, die für die Einstellungen der Verpackung benötigt werden:

- Pitch für den modulgrößen-abhängigen Abstand zwischen den Taschen
- Advance für die Einstellung, um wie viele Taschen vorgeschoben wird
- Dwell für die Dauer der Verschweißung
- Speed für die Geschwindigkeit des Vorschubs
- Stop für die Anzahl einer fertigen Rolle, bei der gestoppt werden soll

Die fünf Parameter ergeben einen ‚Job‘, der vom Bediener an der Robotersteuerung eingestellt und über die serielle Schnittstelle an den Controller der Verpackungseinheit gesendet werden kann. Die Jobs sind auf dem Controller auf einem nicht flüchtigen Speicher vorgespeichert. Wird ein Job mittels Kommandos an den Controller gesendet, werden die Parameter aus dem Speicher ausgelesen und für den weiteren Verpackungsablauf eingestellt. Die angezeigten Parameter am Display werden zur Kontrolle mit den neu ausgelesenen Parametern überschrieben und angezeigt.

Diese Grundstruktur mit dem Aufbau des Programmes soll beim neuen Mikrocontroller ähnlich umgesetzt werden. Dafür wird ein Objekt einer Struct-Variable erzeugt, das diese Parameter als Integer-Variablen beinhaltet. Für die dauerhafte Abspeicherung am EEPROM des Arduinos muss eine Speicherstruktur überlegt werden. Der Speicher am Arduino wird mit einer Lebensdauer von mindestens 100.000 Schreibzyklen angegeben. Dieser Wert kann bei einem ungewollten Schreibvorgang in einer Schleife schnell erreicht werden.

Der Speicher für die Job-Konfigurationen wird so organisiert, dass an der ersten Adresse eine Integer-Variable mit dem aktuell geladenen Programm abgespeichert ist. Wird bei Produktumstellung ein Kommando vom PC für eine Jobkonfiguration empfangen, so wird die Variable überschrieben. Für die Überschreibung der Variable wird im Programm die Funktion „EEPROM.update(Adresse, Wert)“ verwendet. Das hat den Vorteil, dass bei Empfang der bereits geladenen Job-Konfiguration die Variable nicht upgedatet, sprich überschrieben werden muss und dadurch Schreibzyklen eingespart werden können. Der gesamte restliche Speicher (4 KB) steht den verschiedenen Job-Konfigurationen für unterschiedliche Produkt/Material-Kombinationen zur Verfügung.

Für die Speicherung der Job-Konfigurationen sind die Parameter über Menüführung durch das Tastenfeld einzustellen. Ist die Job-Konfiguration eingestellt, muss im vorgesehenen Menü die gewählte Job-Adresse eingegeben und bestätigt werden.

7.1 Display Anzeige und Tastenfeld

Für die Entwicklung der Software soll es keine ersichtlichen Veränderungen in der Menüführung am Display geben. Daher wird die bestehende Struktur als Referenz für die Programmierung der Menüführung verwendet und gleich aufgebaut.

In Abb. 7.2 ist die Menüstruktur abgebildet, durch die über das Tastenfeld navigiert wird. Die grauen Felder stellen die Tasten des Tastenfeldes dar, die in ein spezifisches Menü für Einstellungen der Parameter führen. Im Programm kommen für die Programmierung Switch-Case Strukturen zur Anwendung. Dabei

werden die unterschiedlichen Menüs mit einem Integer-Wert kategorisiert, der während der Navigation geändert wird. Dadurch kann zwischen den Menüpositionen unterschieden werden, um eine Menüführung mit mehreren Eingaben zu ermöglichen. Um eine Eingabe von Parameter-Werten zu ermöglichen, wird eine zusätzliche Boolean-Variable verwendet. Sie wird geändert, wenn der Eingabemodus aktiviert werden soll. Dadurch wird beim Drücken der nummerierten Tasten nicht in die Switch-Case-Anweisung gesprungen, sondern in den Eingabemodus. Darin werden die eingegebenen Parameter einzeln am Display angezeigt und bei Bestätigung durch die Enter-Taste weiterverarbeitet.

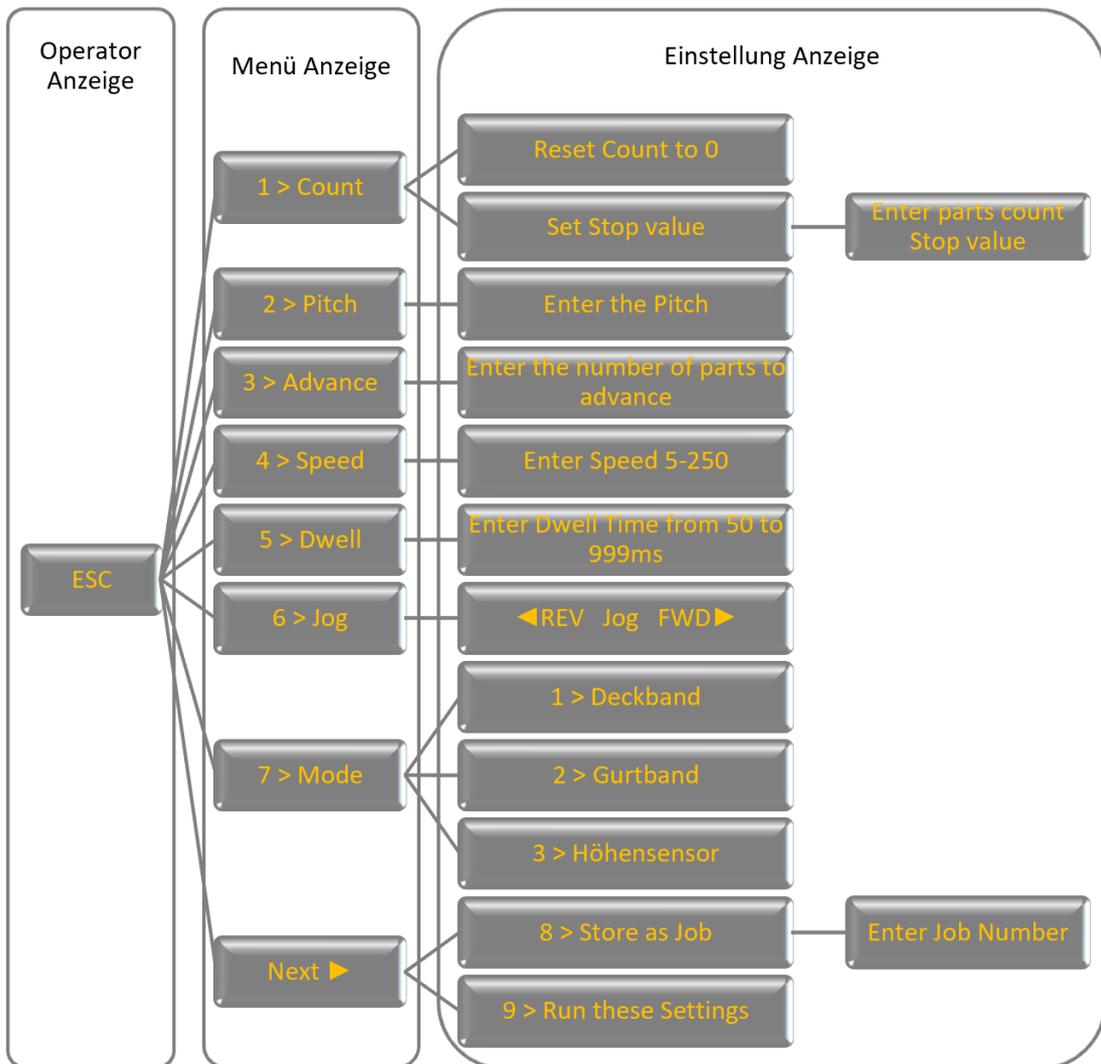


Abb. 7.2: Menüstruktur, Quelle: Eigene Darstellung

In Abb. 7.3 sind als Beispiel die Operator und die Menü Anzeige dargestellt, zwischen denen durch die Integer-Variable unterschieden wird.



Abb. 7.3: Operator/Menü Anzeige, Quelle: Eigene Darstellung

Um die Bibliothek zur Ansteuerung des Displays verwenden zu können, muss ein Objekt der Klasse „LiquidCrystal“ erstellt werden.

```
LiquidCrystal lcd(69, 68, 67, 66, 65, 64);
```

In die Klammer müssen die für das Display verwendeten Pins des Arduinos zur Übergabe an das Objekt geschrieben werden. Aufgrund der Nähe der analogen Pins des Arduinos zur vorgegebenen Kontaktleiste des Displays, werden diese als digitale Pins verwendet. Die fortlaufende Bezeichnung zur digitalen Verwendung der I/Os ist in Abb. 5.12 dargestellt.

Nach Erstellung des Objektes können die Funktionen der Bibliothek verwendet werden. In der Setup-Schleife muss über die Funktion `lcd.begin(20,4)` die Größe des Displays eingestellt werden. Zusätzlich wird die Anzeige gelöscht und die Schreib-/Leserichtung eingestellt.

```
lcd.clear();
```

```
lcd.leftToRight();
```

Dadurch ist sichergestellt, dass die Anzeige leer ist und richtig dargestellt wird. Beim Schreiben muss darauf geachtet werden, dass bei Verwendung des Löschbefehls ein Delay benötigt wird. Versuche zeigen, dass 2 Millisekunden dafür ausreichen. Im Programm ist zur Sicherheit ein Delay von 5 Millisekunden nach jedem Lösch-Befehl einprogrammiert.

Das verwendete Display DMC20481 NY–LY wird über Multiplexing angesteuert, bei dem mehrere Signale zusammengefasst und simultan übertragen werden. Durch diese Eigenschaft muss nicht die gesamte Anzeige überschrieben werden, sondern es können Teile beziehungsweise einzelne Digits verändert werden. Das führt einerseits zum Vorteil, dass weniger Kabel (daher auch weniger Ausgänge) benötigt werden und der Energiebedarf geringer ist.

Für die Navigation zum Schreiben am Display wird der Befehl `lcd.setCursor(Spalte, Zeile)` verwendet. Damit kann der Cursor auf jedes Digit des Displays navigiert werden, um mit Schreibfunktionen wie `lcd.write()` oder `lcd.print()` etwas darzustellen. Der Unterschied zwischen den Funktionen ist, dass mit `lcd.print()` auch Inhalte von Variablen angezeigt werden können, während bei `lcd.write()` nur hartcodierte Texte angezeigt werden können.

Tastenfeld

Für eine erleichterte Programmierung des Tastenfeldes ist die Bibliothek „Keypad.h“ im Programmeditor zu inkludieren. Für die Interpretation der Tasten muss ein zweidimensionales Character-Array initialisiert werden.

```
char keys[ROWS][ COLS] = {  
    {'1', '2', '3', 'A'},  
    {'4', '5', '6'},  
    {'7', '8', '9', 'B'},  
    {'<', '0', '>'}  
};
```

Zusätzlich sind für das nachfolgende Anlegen des Objektes der Klasse „Keypad“, die verwendeten Pins des Arduino für Reihen und Spalten in eindimensionalen Arrays zu initialisieren, die dem Objekt übergeben werden.

```
Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );
```

Dadurch wird das Tastenfeld bei Änderungen an den I/Os des Arduinos richtig interpretiert. Für die Erkennung einer Tastenbetätigung wird die Funktion „keypad.getKey();“ verwendet. Die Rückgabe der Funktion wird in einer Character Variable gespeichert und für die Menüführung in einer Switch-Case-Anweisung interpretiert. Da die Tastaturabfrage für die Verpackungssteuerung zeitlich unkritisch ist, ist eine Kontrolle bei jedem Schleifendurchlauf nicht notwendig. Um Ressourcen für andere Komponenten einzusparen, kann über die Funktion „millis()“, die die Laufzeit des Arduinos zurückgibt, eine Abfrage erfolgen.

```
if(millis())>=(Timesafer+Timer)){  
    Timesafer=millis();  
    char Taste = keypad.getKey();  
}
```

In der Integer-Variable „Timesafer“ kann die Zeit zwischen den Abfrageintervallen in Millisekunden eingestellt werden.

Hochstarten

Ein Feature der neuen Steuerung ist ein beweglicher Text „flex“, der von links nach rechts über das Display wandert. Die Unterfunktion dafür wird in der Setup-Schleife aufgerufen und daher zu Beginn einmal aufgerufen. Für die Positionierung des Cursors wird eine for-Schleife durchlaufen und mittels der Funktion „lcd.scrollDisplayRight();“ der Text bewegt.

7.2 Ansteuerung Schrittmotor

Für den Vorschub des Gurtbandes ist ein Motor der Type ‚MDrive 23 Plus‘ eingebaut. Dieser Antrieb verfügt über einen Schrittmotor, der von einem Mikrostepping-Treiber gesteuert wird. Durch den am Motor vorgesehenen Treiber müssen keine zusätzlichen Treiber am Controller-Board eingeplant werden.



Abb. 7.4: MDrive 23 Plus, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020]

Dem Datenblatt ist zu entnehmen, dass für die Ansteuerung des Treibers vier I/O Ports für Signale zwischen +5 und +24 Volt am Motor vorgesehen sind. Die Auflösung der Schritte pro Umdrehung kann bis zu 51.200 Schritte pro Umdrehung betragen. Dadurch ist es möglich die Länge des Vorschubes exakt zu definieren, sodass die Position der leeren Tasche nicht von der Einlegeposition des Roboters abzuweichen beginnt.

Pin	Funktion	Beschreibung
1	I/O 1	0 to +24 VDC Programmable I/O Point 1
2	I/O 2	0 to +24 VDC Programmable I/O Point 2
3	I/O 3	0 to +24 VDC Programmable I/O Point 3
4	I/O 4	0 to +24 VDC Programmable I/O Point 4
5	AIN	0 to 10 V / 4 to 20 mA / 0 to 20 mA Analog Input.
6	GND	Power and Auxiliary Ground
7	+V	+12 to +75 VDC Motor Power Supply input.

Tab. 7.1: Pinbelegung Schrittmotor, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020]

Auf der Antriebswelle ist eine Scheibe mit einrastenden Bolzen an der Außenseite angebracht, die in den Löchern des Gurtbandes einrasten. Durch das mit Federkraft anliegende Rad darüber, wird das Lösen des Gurtbandes aus den Bolzen verhindert.

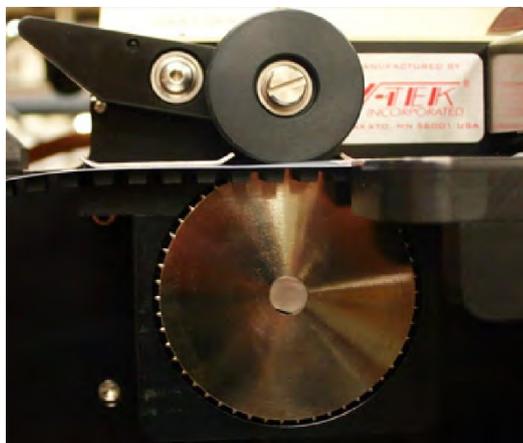


Abb. 7.5: Antriebsrad Schrittmotor, Quelle: Eigene Darstellung

Wie in Abb. 7.6 ersichtlich, sind die I/O Pins 1, 3, 4 und die Versorgungspins 6 und 7 am Treiber angeschlossen. Pin 3 (gelb) und Pin 4 (grün) sind direkt mit den 5-Volt-Ausgängen des Mikrocontrollers verbunden. Pin 1 (orange) führt zum Pneumatik-Ventil der Verschweißungseinheit.

Funktionen der Pins:

- Pin 1: Blockade des Vorschubes während der Verschweißung
- Pin 3: Anzahl der Schritte des Vorschubes
- Pin 4: Drehrichtung des Antriebes

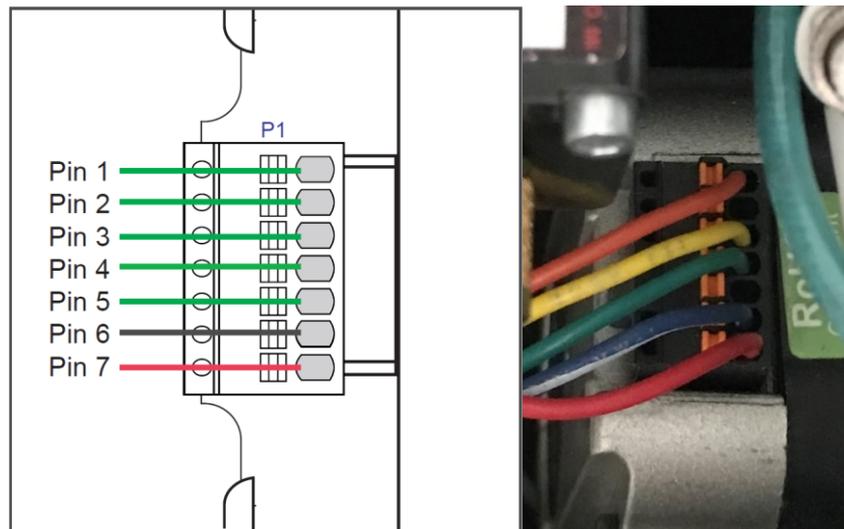


Abb. 7.6: Pinbelegung Schrittmotor, Quelle: Eigene Darstellung

Am Pin 3 werden die Schritte über ein TTL-Signal vom Arduino übertragen. Wie in Abb. 5.7 ersichtlich, wird das Signal mit einer Geschwindigkeit von 12 kHz übertragen. Am Pin 4 wird der Ausgang bei standardmäßigem Vorschub nicht geschaltet und bei gewünschter Rückwärtsdrehung des Motors, wie in Abb. 5.8 ersichtlich, auf +5 Volt gesetzt.

Bei dem 4-poligen Molex-Stecker am Controller-Board ist neben den Pins für Schritte und Drehrichtung ein weiterer Pin (schwarz) ausgeführt. Dieser Pin ist für die Verschweißung im Anschluss des Vorschubes notwendig, um ein pneumatisches Ventil für die Bewegung der Schweißkufen zu schalten. Daher wird dieser Ausgang für die eingestellte Dauer über den Parameter „Dwell“ nach dem Vorschub auf HIGH gesetzt, um in dieser Zeit das Deckband auf das Gurtband zu schweißen.

7.3 Serielle Kommunikation

Da der Arduino speziell für die Kommunikation vorgesehene Pins besitzt, ist wie in Abb. 7.7 ersichtlich, der RS-232-Chip in deren Nähe platziert. Bei den Pins ist zu beachten, dass die Pins 0 und 1 (RX0, TX0) mit dem USB-to-Serial-Adapter des Arduinos verbunden sind. Eine parallele Nutzung der beiden Verbindungen über dieselben Pin-Paare funktioniert nicht. Da der USB-Anschluss für zukünftige Updates des Arduino Programmes verwendet werden wird, ist diese Verbindungsmöglichkeit unverzichtbar. Für die Kommunikation von USB und RS-232 über Pins 0 und 1 am Arduino müsste eine Stiftleiste eingebaut werden, worauf ein Jumper für die Verbindung und Trennung der RS-232-Verbindung benötigt werden würde. Da der Arduino drei weitere Pin-Paare für externe Kommunikation zur Verfügung hat, werden diese anstatt der Pins 0 und 1 für RS-232 verwendet. Beim Arduino sind die dafür vorgesehenen Kommunikations-Pins 16 – 19 in Verwendung. Eines der zwei Pin-Paare ist für eventuelle zukünftige Anwendungen auf eine Stiftleiste (Com 2) redundant herausgeführt und im Programm nicht berücksichtigt. Am Molex-Stecker ‚Com 1‘ ist das Kabel für die RS-232-Verbindung zum PC vorgesehen. Im Programm ist es nicht notwendig für die Kommunikation eine zusätzliche Bibliothek zu inkludieren. Für die Funktionen sind je nach verwendeten Pins, die zugehörigen Objekte wie in diesem Fall „Serial2“ zu verwenden. Mit dem Befehl `Serial2.begin(Baudrate)` werden in der Setup-Schleife die Kommunikations-Pins und die Baudrate definiert. Über die Funktion `Serial2.available()` kann erkannt werden, ob vom PC ein Datenpaket

erhalten werden soll. Über eine if-Anweisung wird in die Schleife gesprungen, um die Daten weiterzuverarbeiten.

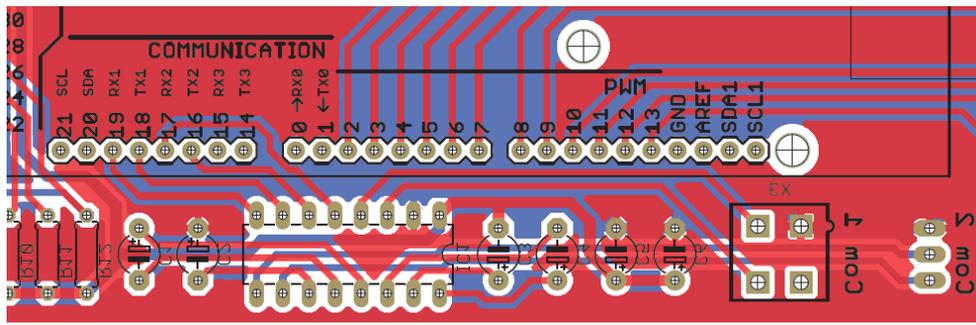


Abb. 7.7: Pins für externe Kommunikation, Quelle: Eigene Darstellung

Vom PC wird das Kommando für die zu verwendenden Parameter der Verpackungseinheit gesendet. Dies ist ein Integer-Wert, der von den Technikern am PC abgespeichert wird und vom produktspezifischen Kameraprogramm ausgelesen und an den Controller gesendet wird. Der Sendevorgang wird beim Umrüsten der Produktionslinie manuell vom Linientechniker ausgelöst. Im Programm des Mikrocontrollers wird regelmäßig überprüft, ob ein Kommando empfangen werden soll. Für die Kommandos zur Einstellung der Job-Konfiguration wird nach dem Einlesevorgang der Parameter an der ersten Stelle des EEPROM upgedatet und die zugehörigen Parameter von deren Speicherort ausgelesen und am Display angezeigt.

7.4 I/O Steuerung

An der I/O-Buchse an der Rückseite des Steuerkastens sind die Anschlüsse mit den Relais der Robotersteuerung verbunden. Sie dienen zur Ansteuerung einer Beckhoff-SPS zum Start eines Verpackungszyklus und zur Zustandsinformation über den aktuellen Status der Verpackungseinheit.

Pinbelegung I/O Buchse:

- Pin 1: Advance Trigger zum Start eines Verpackungszyklus
- Pin 2: Fault Ansprechen eines Sensors
- Pin 3: Ready/Busy Information ob Verpackung bereit oder in Arbeit
- Pin 4: Ground GND
- Pin 5: End of Job Anzahl einer vollen Rolle erreicht

Über die Pins 2, 3 und 5 wird die Information über den aktuellen Zustand der Verpackung an den Roboter weitergegeben. Daher fungieren diese Pins als Ausgänge des Mikrocontrollers und müssen für die Einbindung in die Robotersteuerung Relais schalten können. Die verbauten Relais arbeiten mit +24 Volt und sind auf Eingänge einer Beckhoff-SPS verkabelt. Über den Pin 1 wird der Verpackung der Befehl erteilt, einen Verpackungszyklus durchzuführen. Daher ist dieser Pin als Eingang am Mikrocontroller ausgeführt und muss der Verdrahtung der Steuerung angepasst werden. Aus dem Schaltplan ist ersichtlich, dass der Schließer, des von der Beckhoff-SPS geschalteten Relais, als „Open Collector“ ausgeführt ist und die angelegte Spannung bei Betätigung gegen Ground schaltet. Daher wird der Eingang am Controller-Board über einen Pull-Up-Widerstand auf einen Spannungspegel von +5 Volt gelegt.

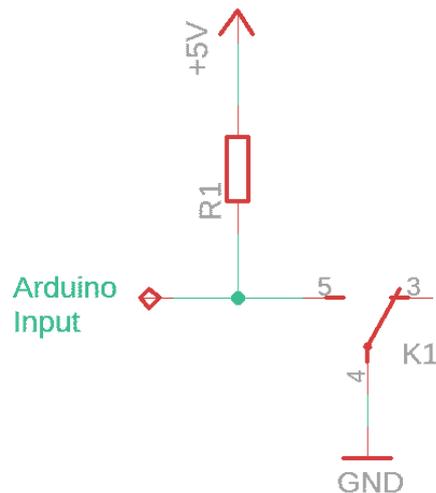


Abb. 7.8: Beschaltung Relais, Quelle: Eigene Darstellung

Wird das Relais von der Robotersteuerung über eine Beckhoff-SPS geschaltet, so wird das Potenzial des Eingangs auf Ground gezogen. Im Programm läuft ein Timer, der überprüft ob sich das Signal länger als 100 Millisekunden auf logisch 0 befindet, um eventuelle Störungen oder Prellungen zu ignorieren. Ist das Signal danach weiterhin auf LOW wird ein Verpackungszyklus gestartet.

Bei den Ausgängen an den Pins 2, 3 und 5 sind +24-Volt-Relais zu schalten und erfordern eine andere Schaltung. Zusätzlich ist zu beachten, dass durch die gespeicherte Energie in der Spule beim Schaltvorgang ein negativer Spannungs-Peak entsteht, sofern wie in diesem Fall keine Freilaufdiode eingebaut ist. Die Höhe der Spannung ist abhängig von der Größe der Spule und der proportional dazu gespeicherten Energie. Die in der Testlinie verbauten Relais können einen negativen Spannungs-Peak im Bereich von 40 Volt erreichen. Auch wenn die I/Os des Arduino mit +24 Volt arbeiten könnten, würde diese Spannung zu Zerstörungen am Mikrocontroller und dem Controller-Board führen. Daher wird auf der Adapterplatine mit einem Ausgangstreiber zu diesem Problem Abhilfe geschaffen. Der Ausgangstreiber ULN2003, wie in Abb. 7.9 dargestellt, dient dazu, hohe Ströme und Spannungen mit einer geringen Spannung eines Mikrocontrollers gegen Ground zu schalten. Er besteht aus 7 Darlington-Arrays über die dauerhaft ein Strom von bis zu 500 mA fließen und eine Spannung von 50 Volt anliegen kann.

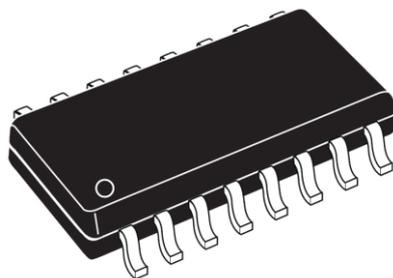


Abb. 7.9: ULN2003, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]

In der Testlinie sind die Relais so verkabelt, dass an einem Anschluss (A1) immer +24 Volt anliegen und der zweite Anschluss (A2) geschaltet wird. Im standardmäßigen Zustand liegt an beiden Anschlüssen +24 Volt an. Wird der digitale Ausgang des Arduino geschaltet, so wird der Ausgang des ULN2003 gegen

Masse geschaltet. Dadurch liegt am Anschluss A2 ein Ground-Potenzial an und das Relais mit einem verbundenen Schließerkontakt wird geschaltet. Durch den Schließer verändert sich die Spannung an dem Eingang der Beckhoff-SPS und stellt dadurch die Statusinformation der Verpackungsstation dem Roboter zur Verfügung. Der hohe Strom des Ausgangstreibers ermöglicht einen sauberen Schaltvorgang der Relais. Prinzipiell wären auch Pegelwandler ohne Relais möglich, aber der Informationsaustausch über die Statusinformationen der Verpackung findet über die Relais in den bestehenden Systemen der Testlinien statt und soll nicht umgebaut werden müssen.

Ersatzschaltbild:

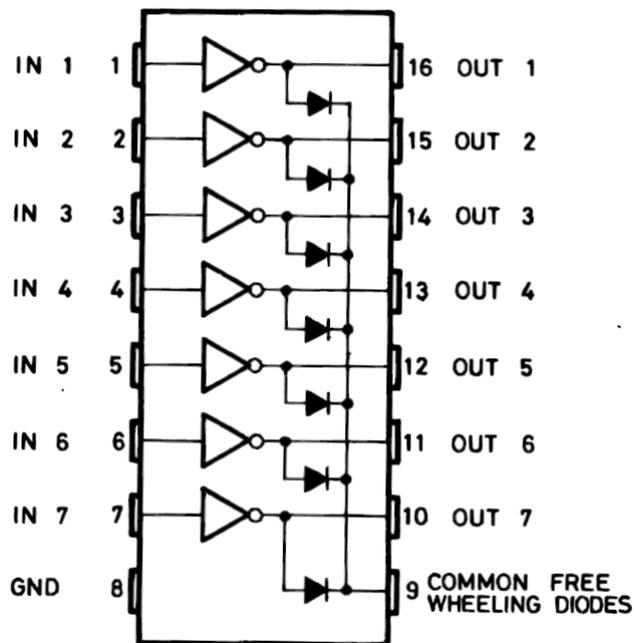


Abb. 7.10: ULN2003 Ersatzschaltbild, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]

In Abb. 7.10 ist ersichtlich, dass der ULN2003 an jedem der 7 Ausgänge eine Freilaufdiode implementiert hat. Dadurch kann der negative Spannungs-Peak der zu schaltenden Relais abgeleitet und damit für die I/Os des Arduino unschädlich gemacht werden. Somit ermöglicht die Verwendung eines Ausgangstreibers die Arbeit mit höheren Spannungen als bei Arduino-Ausgängen.

Der Ausgang auf Pin 2 für ‚Fault‘ wird geschaltet, sobald einer der aktivierten, in Abb. 2.6 dargestellten Sensoren anspricht.

Pin 3 für den ‚Ready/Busy‘-Status wird geschaltet, wenn ein Verpackungszyklus gestartet wird. Dadurch weiß die Robotersteuerung, dass die Verpackungseinheit beschäftigt ist und der Roboter mit dem Einlegen eines weiteren Modules in eine Tasche des Gurtbandes warten muss. Nach der Fertigstellung eines Verpackungszyklus wird der Ausgang wieder auf ‚Ready‘ gesetzt.

Der verbleibende Pin 5 für den ‚End of Job‘-Status wird geschaltet, sobald der zuvor eingestellte Wert der Anzahl von Modulen auf einer Rolle erreicht ist, respektive eine Rolle voll ist. Danach muss über das Drücken der ESC-Taste am Tastenfeld die Eingabe vom Operator bestätigt werden und die Verpackungsmaschine führt eine voreingestellte Anzahl an Verpackungszyklen durch, um die Rolle abzuschließen.

8 ERGEBNISSE UND AUSBLICK

Durch die Umsetzung des Projektes kann das Wissen über die Hard- und Software der Verpackungsstation in die eigene Firma geholt werden. Beim Reverse Engineering ist es erlaubt, durch eigene Beobachtung und Nachbau, Wissen über Bestehendes zu erhalten. In diesem Fall ist es notwendig, da vom Hersteller zum eigenen Vorteil keine detaillierten Informationen über Schaltungen und Programmierung zur Verfügung gestellt werden. Aufgrund der genauen Analyse der Verpackungseinheit, kann die Hardware vom Mikrocontroller über das neu entwickelte Controller-Board angesteuert werden. Durch die Berücksichtigung der Befestigungspunkte des originalen Controller-Boards, Displays und Arduinos beim Design des neuen Controller-Boards, kann die neue Steuereinheit ohne Zusatzaufwand im Steuerkasten eingebaut werden. Die Ausprogrammierung der Funktionen mit anschließenden Tests zeigt, dass die Ansteuerung der Komponenten funktioniert. Für die Kommunikation dient eine Funktion, die die gesendeten Daten vom PC in einer Variable zwischenspeichern lässt und wieder zurücksendet. Somit ist sichergestellt, dass sowohl das Empfangen als auch das Senden von Daten zwischen den Teilnehmern funktioniert.

Ein Vorteil bei der Umsetzung des Projektes ist, dass verbaute Komponenten wie die Verschweißungssteuerung eine eigene Intelligenz besitzen und somit nicht alle Abläufe ausschließlich vom Mikrocontroller gesteuert werden müssen. Im konkreten Fall kann an der Verschweißungssteuerung selbst eine Soll-Temperatur inklusive Schalthysterese eingestellt werden. Dadurch muss neben der Ablaufsteuerung nicht zusätzlich die Temperatursteuerung der Schweißkufen vom Mikrocontroller übernommen werden. Die Verschweißungssteuerung liefert über einen Ausgang die Information, ob sich die Temperatur in der Toleranz befindet.

Durch die eigene Entwicklung des Controller-Boards und Programmierung des Mikrocontrollers ist es möglich, die Funktionen von zukünftigen Ideen flexibel umzusetzen. Ein Ansatz dafür wäre im Programm die serielle Schnittstelle für Datenkommunikation miteinzubeziehen. Dadurch sind neben den Job-Kommandos auch Steuerbefehle von beiden Teilnehmern bei zukünftigen Anwendungen machbar. Es wäre theoretisch möglich, die I/O-Verbindung, die für die Ein- und Ausgänge der Beckhoff-SPS benötigt wird, durch die serielle Kommunikation zu ersetzen.

Neben den neu entwickelten Testlinien ist das veraltete Design des Steuerkastens wenig ansprechend und könnte für zusätzlich verbesserte Visualisierung-Möglichkeiten selbst entworfen werden.

Der nächste Schritt im Unternehmen ist die vollständige Ausprogrammierung der Abläufe mit anschließenden Tests im laufenden Betrieb. Des Weiteren muss der benötigte Multiplikator für die Steps per Revolution des Schrittmotors für den Pitch eruiert werden. Dafür sollen mehrere Meter eines Gurtbandes über Vorschübe durchgelassen und im Anschluss kontrolliert werden, ob sich die Position der Tasche zum Einlegen verschiebt. Dadurch soll ein passender Multiplikator für die Programmierung und Einstellung der Variable Pitch gewonnen werden.

Im Anschluss kann die neu entwickelte Steuereinheit auf alle Testlinien ausgerollt werden. Für Updates der Software des Mikrocontrollers ist die Abdeckung des Steuerkastens zu entfernen und über die USB-Schnittstelle hochzuladen.

LITERATURVERZEICHNIS

Gedruckte Werke (5)

Davies, John (2008): *MSP430 Microcontroller Basics*, Elsevier

Eilam, Eldad (2011): *Reversing: Secrets of Reverse Engineering*, John Wiley & Sons

James, Christina (2012): *OEM TM-50 SMD Taping Module User's Guide*

Raja, Vinesh; Fernandes, Kiran (2007): *Reverse Engineering: An Industrial Perspective*, Springer Science & Business Media

Valdes-Perez, Fernando; Pallas-Areny, Ramon (2017): *Microcontrollers: Fundamentals and Applications with PIC*, CRC Press

Wissenschaftliche Artikel (1)

Lee, Justin (2018): *Coding Camp Part 2: How to Display "Hello, ODROID-GO" on an LCD Screen*, in: Odroid Magazine

Online-Quellen (23)

Maxim Integrated (2019): *+5V-Powered, Multichannel RS-232 Drivers/Receivers*

<https://datasheets.maximintegrated.com/en/ds/MAX220-MAX249.pdf> [Stand: 28.September.2020]

Schneider Electric (2008): *MDrive 17 & 23 Plus Motion Control*

https://motion.schneider-electric.com/downloads/manuals/MDI17_23_Plus.pdf [Stand: 30.September.2020]

STMicroelectronics (2020): *Datasheet ULN2001, ULN2002, ULN2003, ULN2004*

<https://www.st.com/resource/en/datasheet/uln2001.pdf> [Stand: 30.September.2020]

Arduino *Arduino Store*

<https://www.arduino.cc> [Stand: 22.August.2020]

Asus *Asus Tinker Board*

<https://www.asus.com> [Stand: 24.Juli.2020]

Banana Pi *Banana Pi F2P*

www.banana-pi.org [Stand: 24.Juli.2020]

BeagleBoard *Beagle Bone Black*

<https://beagleboard.org> [Stand: 24.Juli.2020]

Keyence *Datasheetspdf.com*

<https://datasheetspdf.com> [Stand: 26.September.2020]

Digi-Key *DMC-20481NY-LY-AZE-BJN*

<https://www.digikey.at/product-detail/de/kyocera-international-inc/DMC-20481NY-LY-AZE-BJN/73-1300-ND/1004561> [Stand: 16.Juli.2020]

Keyence *Digitale Lichtleitersensoren*

<https://www.keyence.eu> [Stand: 14. September.2020]

Microchip *PIC18F6585*

<https://www.microchip.com> [Stand: 29. Juni.2020]

Odroid *Odroid C4*

<https://www.odroid.co.uk> [Stand: 22. Juli.2020]

Kompendium *RS-232 - Die serielle Schnittstelle*

<https://kompendium.infotip.de/rs-232-die-serielle-schnittstelle.html> [Stand: 21. Oktober.2020]

Picprojects *Simple RS-232 to logic level converter*

<http://picprojects.org.uk/projects/simpleSIO/ssio.htm> [Stand: 29. September.2020]

Raspberry Pi *Raspberry Pi 4 Model B*

<https://www.raspberrypi.org> [Stand: 24. Juli.2020]

V-TEK, Inc. *OEM TM-50*

<http://www.vtekusa.com> [Stand: 14. September.2020]

RS Components *RS Pro K78xxM-1000R3 DC-DC-Converter*

<https://docs.rs-online.com/dc16/A700000006631882.pdf> [Stand: 15. Oktober.2020]

RS Components *Mikrocontroller-Leitfaden*

<https://at.rs-online.com/web/generalDisplay.html?id=ideen-und-tipps/mikrocontroller-leitfaden> [Stand: 24. Oktober.2020]

Rechtsinformationssystem des Bundes *Bundesgesetz gegen den unlauteren Wettbewerb 1984 - UWG*

<https://www.ris.bka.gv.at/GeltendeFassung.wxe?Abfrage=Bundesnormen&Gesetzesnummer=10002665&ShowPrintPreview=True> [Stand: 8. Oktober.2020]

RS Components *RS-Online*

<https://at.rs-online.com> [Stand: 7. November.2020]

Ben TRAVAGLIONE, Andrew (2014): *Using low cost single-board microcontrollers to record underwater*

https://www.acoustics.asn.au/conference_proceedings/INTERNOISE2014/papers/p236.pdf [Stand: 20. Juli.2020]

Justiz, Bundesamt (2019): *Gesetz zum Schutz von Geschäftsgeheimnissen (GeschGehG)*

<http://www.gesetze-im-internet.de/geschgeh/BjNR046610019.html> [Stand: 8. Oktober.2020]

Kristian Saether, Ingar (2008): *Introducing a New Breed of Microcontrollers for 8/16-bit*

<http://ugweb.cs.ualberta.ca/~c274/resources/tools/doc/atmel/Xmega-overview.pdf> [Stand: 9. Oktober.2020]

ABBILDUNGSVERZEICHNIS

Abb. 1.1: Flex Standort Althofen, Quelle: Eigene Darstellung	1
Abb. 1.2: Process Flow Flex, Quelle: Eigene Darstellung.....	2
Abb. 2.1: OEM TM-50, Quelle: V-TEK, Inc., Online-Quelle [14.September.2020].....	3
Abb. 2.2: OEM TM-50 Führung des Gurtbandes, Quelle: James (2012).....	3
Abb. 2.3: OEM TM-50 Systemkomponenten, Quelle: James (2012).....	4
Abb. 2.4: Testlinie bei Flex, Quelle: Eigene Darstellung	5
Abb. 2.5: Originales Controller-Board, Quelle: Eigene Darstellung	5
Abb. 2.6: Sensor FS-V21, Quelle: Keyence, Online-Quelle [14.September.2020].....	6
Abb. 2.7: Piezo-Signalgeber, Quelle: RS Components, Online-Quelle [7.November.2020].....	6
Abb. 2.8: Erweiterbarer Informationsaustausch, Quelle: Eigene Darstellung	7
Abb. 4.1: Blockdiagramm Mikrocontroller, Quelle: Eigene Darstellung	12
Abb. 4.2: Ergebnis der Befragung über den Hersteller des aktuell verwendeten Prozessors, Quelle: Studie Aspencore (2019) https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf	15
Abb. 4.3: Ergebnis der Befragung des gewünschten Herstellers des Prozessors für zukünftige Projekte, Quelle: Studie Aspencore (2019) https://www.embedded.com/wp-content/uploads/2019/11/EETimes_Embedded_2019_Embedded_Markets_Study.pdf	15
Abb. 4.4: Harvard Architektur, Quelle: Eigene Darstellung	17
Abb. 4.5: Von Neumann Architektur, Quelle: Eigene Darstellung	17
Abb. 4.6: Arduino MKR CAN Shield, Quelle: Arduino, Online-Quelle [22.August.2020]	19
Abb. 4.7: Arduino MEGA Prototyping Shield, Quelle: Arduino, Online-Quelle [22.August.2020]	19
Abb. 4.8: Raspberry Pi 4 Model B, Quelle: Raspberry Pi, Online-Quelle [24.Juli.2020].....	20
Abb. 4.9: Arduino Mega 2560 Rev3, Quelle: Arduino, Online-Quelle [22.August.2020].....	22
Abb. 4.10: Odroid C4, Quelle: Odroid, Online-Quelle [22.Juli.2020].....	23
Abb. 4.11: Beagle Bone Black, Quelle: BeagleBoard, Online-Quelle [24.Juli.2020].....	25
Abb. 4.12: Banana Pi BPI F2S, Quelle: Banana Pi, Online-Quelle [24.Juli.2020]	26
Abb. 4.13: ASUS Tinker Board, Quelle: Asus, Online-Quelle [24.Juli.2020]	27
Abb. 5.1: Vorderseite TM50, Quelle: Eigene Darstellung	29
Abb. 5.2: Rückseite TM50, Quelle: Eigene Darstellung	30
Abb. 5.3: Steuerkasten im Inneren, Quelle: Eigene Darstellung	30
Abb. 5.4: Virtual Bench VB-8012, Quelle: Eigene Darstellung.....	31

Abb. 5.5: Messung der Frequenz am Schrittmotor, Quelle: Eigene Darstellung	32
Abb. 5.6: Signal am Schrittmotor-Treiber, Quelle: Eigene Darstellung	32
Abb. 5.7: Signal am Schrittmotor mit höchster Frequenz, Quelle: Eigene Darstellung	33
Abb. 5.8: Signal Drehrichtung Schrittmotor, Quelle: Eigene Darstellung	34
Abb. 5.9: FS-V21-Beschaltung, Quelle: Keyence, Online-Quelle [26.September.2020]	35
Abb. 5.10: Microchip PIC18F6585, Quelle: Microchip, Online-Quelle [29.Juni.2020]	35
Abb. 5.11: Arduino Mega Rev3, Quelle: Arduino, Online-Quelle [22.August.2020]	36
Abb. 5.12: Pinout Arduino Mega Rev3, Quelle: Arduino, Online-Quelle [22.August.2020]	37
Abb. 6.1: Originales LC-Display, Quelle: Digi-Key, Online-Quelle [16.Juli.2020]	38
Abb. 6.2: Verglaster Anzeigebereich, Quelle: Eigene Darstellung	38
Abb. 6.3: Versuchsaufbau LCD, Quelle: Eigene Darstellung	40
Abb. 6.4: Hello World! auf dem Versuchsaufbau, Quelle: Eigene Darstellung	40
Abb. 6.5: Schematische Darstellung der Kommunikation, Quelle: Eigene Darstellung	41
Abb. 6.6: RS232/TTL Datenübertragung, Quelle Kompendium, Online-Quelle [21.Oktober.2020]	41
Abb. 6.7: Schaltung RS-232/TTL, Quelle: Picprojects, Online-Quelle [29.September.2020]	42
Abb. 6.8: RS232/TTL Signalmessung, Quelle: Eigene Darstellung	42
Abb. 6.9: Maxim MAX232, Quelle: Eigene Darstellung	43
Abb. 6.10: MAX232 Beschaltung, Quelle: Maxim Integrated (2019), Online-Quelle [28.September.2020]	43
Abb. 6.11: Vorderseite Versuchsaufbau, Quelle: Eigene Darstellung	44
Abb. 6.12: Rückseite Versuchsaufbau, Quelle: Eigene Darstellung	44
Abb. 6.13: DC/DC-Konverter-Schaltung, Quelle: Eigene Darstellung	45
Abb. 6.14: Beschaltung DC/DC-Converter, Quelle: RS Components, Online-Quelle [15.Oktober.2020]	46
Abb. 6.15: Arduino Mega Rev3 Symbol, Quelle: Eigene Darstellung	46
Abb. 6.16: Arduino Mega Rev3 Footprint, Quelle: Eigene Darstellung	46
Abb. 6.17: Steuerbox von innen, Quelle: Eigene Darstellung	47
Abb. 6.18: Aussparung am Platinenrand, Quelle: Eigene Darstellung	47
Abb. 6.19: Befestigungspunkte Controller-Board, Quelle: Eigene Darstellung	48
Abb. 6.20: Platinendesign Bottom Layer, Quelle: Eigene Darstellung	49
Abb. 6.21: Platinendesign Top Layer, Quelle: Eigene Darstellung	49
Abb. 6.22: Bestückte Adapterplatine Vorderseite, Quelle: Eigene Darstellung	50
Abb. 6.23: Bestückte Adapterplatine Rückseite, Quelle: Eigene Darstellung	50

Abb. 7.1: Arduino IDE Schleifen, Quelle: Eigene Darstellung	51
Abb. 7.2: Menüstruktur, Quelle: Eigene Darstellung	53
Abb. 7.3: Operator/Menü Anzeige, Quelle: Eigene Darstellung	53
Abb. 7.4: MDrive 23 Plus, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020].....	55
Abb. 7.5: Antriebsrad Schrittmotor, Quelle: Eigene Darstellung	56
Abb. 7.6: Pinbelegung Schrittmotor, Quelle: Eigene Darstellung	57
Abb. 7.7: Pins für externe Kommunikation, Quelle: Eigene Darstellung	58
Abb. 7.8: Beschaltung Relais, Quelle: Eigene Darstellung	59
Abb. 7.9: ULN2003, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]	59
Abb. 7.10: ULN2003 Ersatzschaltbild, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]	60
Abb. 8.1: Schaltplan Controller-Board, Quelle: Eigene Darstellung	69
Abb. 8.2: Beschaltung MAX232, Quelle: Maxim Integrated (2019), Online-Quelle [28.September.2020]	70
Abb. 8.3: ULN2003 Maximum-Werte, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]	70
Abb. 8.4: MDrive23Plus Spezifikationen, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020]	71

TABELLENVERZEICHNIS

Tab. 4.1: Raspberry Pi 4 Eigenschaften, Quelle: Raspberry Pi, Online-Quelle [24.Juli.2020]	21
Tab. 4.2: Arduino Mega 2560 Rev3 Eigenschaften, Quelle: Arduino, Online-Quelle [22.August.2020]	22
Tab. 4.3: Odriod C4 Eigenschaften, Quelle: Odroid, Online-Quelle [22.Juli.2020]	24
Tab. 7.1: Pinbelegung Schrittmotor, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020]	56

ABKÜRZUNGSVERZEICHNIS

ADC	Analog-to-Digital-Converter
ALU	Arithmetisch-logische Einheit
CAD	Computer Aided Design
CPU	Central Processing Unit
EMV	Elektromagnetische Verträglichkeit
ESD	Electro Static Discharge
GPIO	General-Purpose Input/Output
ICSP	In Circuit/System Programmer
I/O	Input/Output
IoT	Internet of Things
LCD	Liquid Crystal Display
MIPI	Mobile Industry Processor Interface
MIPS	Million Instructions Per Second
PRU	Programmable Real-Time Unit
PWM	Pulsweitenmodulation
RAM	Random Access Memory
RE	Reverse Engineering
ROM	Read Only Memory
RPD	Rapid Product Development
SPS	Speicherprogrammierbare Steuerung
T&R	Tape and Reel
UART	Universal Asynchronous Receiver Transmitter

ANHANG 2: AUSZÜGE DATENBLÄTTER

Maxim MAX232

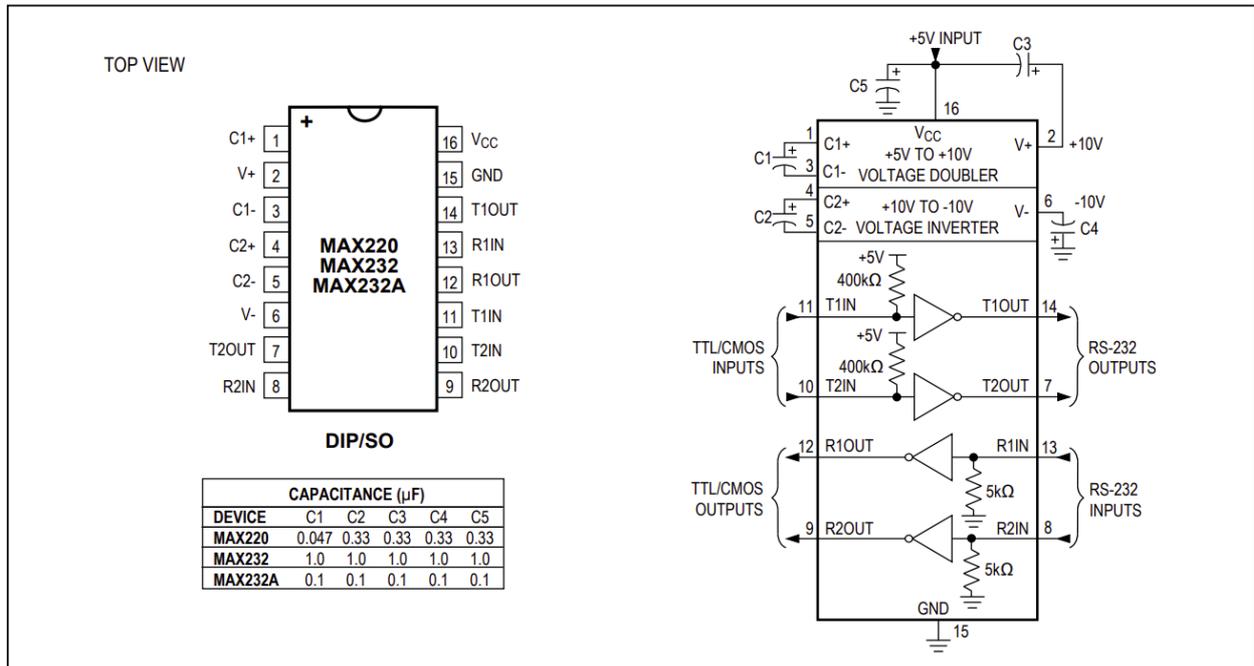


Abb. 8.2: Beschaltung MAX232, Quelle: Maxim Integrated (2019), Online-Quelle [28.September.2020]

ULN2003:

Symbol	Parameter	Value	Unit
V_O	Output voltage	50	V
V_I	Input voltage (for ULN2002A/D - 2003A/D - 2004A/D)	30	V
I_C	Continuous collector current	500	mA
I_B	Continuous base current	25	mA
I_F	Clamping diode continuous current	350	mA
V_R	Clamping diode reverse voltage	50	V
T_A	Operating ambient temperature range	- 40 to 85	°C
T_{STG}	Storage temperature range	- 55 to 150	°C
T_J	Junction temperature	150	°C
ESD	Electrostatic discharge rating - HBM	2	kV

Abb. 8.3: ULN2003 Maximum-Werte, Quelle: STMicroelectronics (2020), Online-Quelle [30.September.2020]

MDrive23Plus

Motion Specifications									
Microstep Resolution - Open Loop									
Number of Resolutions								20	
Available Microsteps Per Revolution									
200	400	800	1000	1600	2000	3200	5000	6400	10000
12800	20000	25000	25600	40000	50000	51200	36000 ¹	21600 ²	25400 ³
1=0.01 deg/ μ step 2=1 arc minute/ μ step 3=0.001 mm/ μ step									
Microstep Resolution - Closed Loop Configuration (Optional)									
Steps Per Revolution (Fixed)								51200	
Position Resolution								2048	
Optional Differential Encoder (Internally Mounted)									
Type								Internal, Magnetic	
Resolution (Lines)								512	
Resolution (Edges)								2048	
Counters									
Counter 1 (C1) Type								Position	
Counter 2 (C2) Type								Encoder	
Resolution								32 Bits	
Maximum Edge Rate								5 MHz	
Velocity									
Range								$\pm 5,000,000$ Steps/Sec.	
Resolution								0.5961 Steps/Sec.	
Acceleration/Deceleration									
Range								1.5×10^9 Steps/Sec. ²	
Resolution								90.9 Steps/Sec. ²	

Abb. 8.4: MDrive23Plus Spezifikationen, Quelle: Schneider Electric (2008), Online-Quelle [30.September.2020]