Projektarbeit

Embedded C++

Sommersemester 2024

Hochschule
Bonn-Rhein-Sieg
University of Applied Sciences

Farbsortierung mittels Kranarm

von

Marc Heimermann, Marvin Schönwälder und Fabian Weber

Inhaltsverzeichnis

I. Lasten- & Pflichtenheft

1. Anforderungen gemäß Lastenheft

Das Lastenheft fordert eine Steuerungs-Software für eine Fischertechnik-Sortieranlage, welche Holzsteine nach Farben sortiert und lagert. Das Lastenheft stellt folgende Anforderungen:

- Es sollen folgende Bedienelemente implementiert werden:
    - Start-/Stopp-Taste (bzw. Auswurftaste)
- Es soll ein LCD eingebunden werden und folgende Anzeigeelemente implementiert werden:
    - Aktueller Betriebszustand
    - Aktueller Luftdruckstatus
- Nach einem Systemstart erfolgt vorerst keine Aktion. Erst nach einem Tastendruck startet eine Verarbeitung der Objekte.
- Es soll eine Start-/Stopp-Taste implementiert werden, die die Verarbeitung startet, sofern ein Objekt im Eingangslager vorhanden ist. Ein erneutes Betätigen unterbricht die Verarbeitung, bis die Start-/Stopp-Taste ein weiteres Mal betätigt wird.
- Kompressor nur eingeschaltet, wenn nicht genügend Druck vorliegt.

2. Beschreibung der Sortieranlage

Die Fischertechnik-Sortieranlage besteht aus einer Lichtschranke, einem Farbsensor, einem Eingangslager und einem Ausgangslager mit der Kapazität für drei unterschiedliche Farben. Das Sortieren erfolgt durch einen Kran, der sich mittels eines Motors, im Uhrzeigersinn bis zu ca. 180 Grad bewegen und sich im Bedarfsfall absenken kann. Zusätzlich beinhaltet der Kran noch einen Ausgangspositionssensor und einen Flankenzähler, welcher der Positionserkennung dient.

Außerdem hat die Anlage einen Kompressor incl. mehrerer Pneumatik-Steuereinheiten die das Auf-/Absenken, sowie das Ansaugen von Werkstücken ermöglicht.

Das verwendete Board ist das STM32L100C-Discovery Kit. Es ist mit Tasten, LCD und einem Drehknopf ausgestattet. Wir steuern die Anlage mit dem Wannenstecker an, weitere Anschlüsse sind vorhanden, werden aber nicht genutzt. Die Belegung der Pins des Wannensteckers befindet sich im Anhang.

Die vier Knöpfe auf der rechten Seite des Board werden für die unterschiedlichen Funktionen der Sortieranlage genutzt.

3. Implementierte Funktionen

- Sortierung der Werkstücke nach Farbe
- Zwei verschiedene Modi – Eine automatische Verarbeitung und eine manuelle Verarbeitung
- LCD für unterschiedliche Informationen:
    - Betriebsmodus
    - Kranposition
    - Druckanzeige
    - Farbanzeige
    - Anzahl von Werkstücken im Ausgangslager
    - Lichtschrankenstatus
- Mehrere Bedienelemente
    - Rotary Encoder zum Scrollen und Bewegen des Krans
    - 4 Knöpfe
- Ein Verarbeitungsstart benötigt einen Tastendruck
- Start/Stopp-Taste
- Auswurftaste
- Kompressor ist nur an, wenn er benötigt wird.

4. Abweichungen zum Lastenheft

Zusätzlich zum Lastenheft, werden mehrere Informationselemente auf dem Display angezeigt. Außerdem ist eine manuelle Steuerung der Sortieranlage möglich. Die Bedienung des Krans ist intuitiv und über den Rotary Encoder einfach möglich.

Die Anzahl der erfolgreichen Werkstücken wird hochgezählt und auf dem Display angezeigt, ausgenommen sind Werkstücke, die zum Auswurf markiert wurden.

II. Dokumentation & Diagramme

1. Entwicklungsprozess

Die Sortieranlage wurde vom Team begutachtet und der Aufbau abstrahiert. Es wurden erste objektorientierte Einordnungen vorgenommen und Überlegungen angestellt, welches Bauteil mit welcher Rolle später im Source Code bedient werden soll.

Aufgrund fehlenden, bzw. abweichenden Lastenheft und Pinbelegung, haben wir begonnen in den jeweiligen Datasheets (bzgl. des Boards) und in der „EmbSysLib“ Informationen zu sammeln, um danach per Trial & Error die jeweilige Pin-Belegung herauszufinden und zu dokumentieren. So konnten wir erfolgreich eine erste „Pinbelegungsmap“ herstellen.

Daraufhin haben wir uns geeinigt ein GitHub-Repository zu nutzen, um unser Projekt zu sichern und zu verwalten. Ein UML-Klassendiagramm-Prototyp ist dabei bereits entstanden, welcher unser erster Ansatz war, um „GitHub-Issues“ untereinander zu verteilen.

Die ersten Grundstrukturen wurden erstellt und es kam zu einer fehlerhaften Ansteuerung von digitalen Elementen via Analog-Digital-Konverter, danach konnten wir jedoch unsere „Pinbelegungsmap“ finalisieren. Wir haben unsere UML-Struktur dramatisch erweitern müssen, da wir einen Großteil der Logik kapseln wollen. So sind unteranderem die Klassen Crane & Display entstanden. Erste Testläufe mit der Klasse Crane wurden durchgeführt.

Um einen reibungslosen Ablauf bzgl. des Timings gewährleisten zu können, haben wir weitere Klassen, Funktionen und Structs ergänzt, die als eine Zustandsmaschine fungieren sollen. Insgesamt soll das Benutzerinterface in einer App-Klasse gekapselt werden.

Es ist aufgefallen, dass der Kran die Werkstücke nicht präzise genug auf dem Farbsensor platziert, sodass es zu Fehlmessungen und daraufhin zur Fehlsortierung kam. Um dem entgegenzuwirken, wurde die Position des Farbsensors verlegt.

Nach dem Abschluss von unseren großzügigen Tests der Zustandsmaschine für die Bewegung der Sortieranlage haben wir mit den Displayfunktionen anfangen können. Wir implementieren ein Menüsystem, welche in einer Containerklasse „Display“ gesammelt wird. Es soll möglich sein in einem Hauptmenü aus mehreren Optionen wählen zu können.

Zum Beispiel soll der Benutzer die Wahl haben, ein automatisches oder ein manuelles Ausführungsprogramm zu wählen. Dabei kam uns die Idee, dass das Display automatisch alle wichtigen Informationen anzeigen soll. Die gewünschte Anforderung eines Auswurfknopfes wird dabei virtuell implementiert. Leider hat das umfangreiche Menü den Rahmen gesprengt. Unser Programm ist zu groß für den Programmspeicher des Microcomputers. Daher haben wir unsere zuvor eingeführten Klassen: Display, Screen, MenuEntry, MenuButton, MenuText, wieder gestrichen. Die Folge davon ist ein wesentlich simpleres Menü als zuvor. Wir implementieren lediglich eine Scroll Funktion für das Informationsdisplay, eine Ansteuerung der Anlage läuft nicht mehr über ein virtuelles Menü mit Cursor, sondern nur noch per Buttons.

Die Ordnerstruktur wurde von einer Frontend/Backend Unterteilung zu einer Unterteilung nach Funktion verändert. Die letzten Funktionen wurden implementiert. Die Testphase beginnt, mehr dazu in der Testdokumentation.

2. Diagramme (UML-Strukturen)

Die vollständige Steuerungssoftware für die Sortieranlage beinhaltet 10 Klassen. Ein ausführliches UML-Klassendiagramm incl. Objektbeziehungen und Methoden sind dem Paper beigelegt.

Außerdem wurde ein grobes Zustandsdiagramm angefertigt, um die Funktionsweise der Zustandsmaschine zu verdeutlichen. (Siehe Anhang)

3. Steuerungsablauf – automatisch

1. Die Anlage fährt hoch, Kompressor, Farbsensor und Lichtschranke werden eingeschaltet, der Kran fährt in seine Ausgangsposition und betätigt dabei den Endswitch.
2. Die Lichtschranke prüft, ob ein Werkstück zur Verarbeitung vorliegt. Falls dem so ist, bewegt sich der Kran herab und nimmt das Werkstück per Saugknopf auf. Der Kran steigt wieder hoch.
3. Der Kran bewegt sich nach rechts zum Lichtsensor. Das Werkstück wird abgesenkt und vom Farbsensor erfasst. Das System erkennt die Farbe. Der Kran steigt wieder auf.
4. Der Kran bewegt sich zum Ausgangslager. Je nach Farbe, wird eine andere Reihe gewählt. Das Werkstück wird abgelegt. Der Kran bewegt sich wieder zurück zur Ausgangsposition.

Es kann eine Auswurfoption gewählt werden, wenn diese Option gewählt wird, so wird unabhängig vom aktuellen Verarbeitungsstand, das Werkstück in einer gesonderten Position aussortiert. Der Kran bewegt sich wieder auf die Ausgangsposition. Die Verarbeitung wird fortgeführt.

4. Steuerungsablauf – manuell

Die Steuerung erfolgt manuell nach gegebener Bedienungsanleitung.

Bedienungsanleitung	
Rotary Encoder – Linksbewegung	Bewegt den Kran nach links
Rotary Encoder – Rechtsbewegung	Bewegt den Kran nach rechts
Rotary Encoder – Drücken	Senkt den Kran ab/auf und nimmt das Werkstück auf, bzw. legt es ab. (mindestens eine Sekunde warten)
Knopf 3	Betreten/Verlassen der manuellen Bedienung

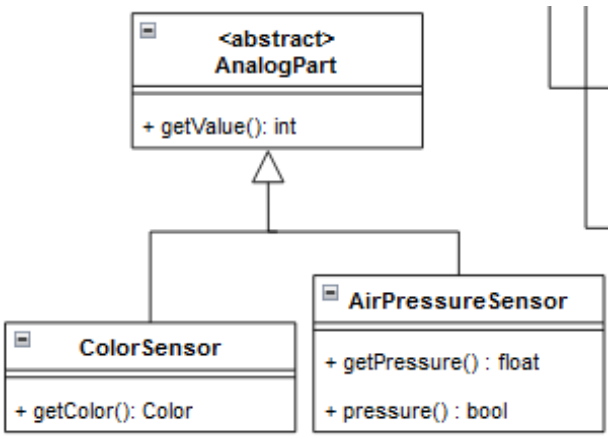
III. Testdokumentation

Anforderung	**Testablauf**	**Ergebnis**	**Bemerkung**
Sortierung nach Farbe	Steine wurden aus dem Eingangslager vom Kran, zum Farbscanner transportiert, gescannt und ins richtige Regal transportiert	Bestanden	Probleme wenn Sensorhalterung, oder Ausgangslager verstellt. Daher bei Abweichungen Anlage prüfen
Starttaste/Start erst ab Interaktion	Das Sortierprogramm startet erst ab Auswahl des gewünschten Programmes.	Bestanden	
Auswurftaste	Das Werkstück kann im Zeitraum von der Aufnahme bis zum Colorsensor zum Auswurf markiert werden. Markierte Werkstücke werden wie gewünscht entfernt. Danach geht die Verarbeitung weiter.	Bestanden	Wurde nur im automatischen Modus implementiert.
Stoptaste	Die Verarbeitung wurde nach Betätigung gestoppt. Eine erneute Betätigung sorgt für eine Fortführung.	Bestanden	Wurde nur im automatischen Modus implementiert
LCD-Display	Das Display zeigt gewünschte Informationen an.	Bestanden	/
Kompressor-Status	Kompressor aktiviert sich nur, wenn benötigt wird. Auf dem Display erscheint „PUMPING“, wenn aktiviert.	Bestanden	In zu langen Pausen kann es zum Druckabfall kommen. Das Werkstück fällt ab
Verarbeitungs-Stopp	Sobald alle Werkstücke verarbeitet wurden, bewegt sich der Kran auf die Ausgangsposition zurück. Der Kompressor schaltet sich aus. Das System ist jedoch noch bereit, weitere Werkstücke zu verarbeiten. Dies geschieht automatisch, sobald die Lichtschranke unterbrochen wird.	Bestanden	/

IV. Fazit

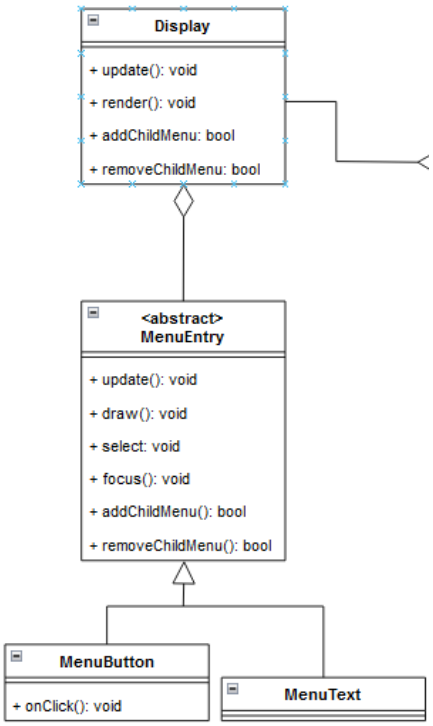
Das Arbeiten mit der Sortieranlage hat uns sehr Spaß gemacht. Es mussten Herausforderungen überwunden, Dokumentationen gewälzt und viel getestet werden.

Besonders war das Fehlen eines zugeschnittenen Lastenheftes eine Herausforderung, da wir unsere Ziele selbst stecken mussten und so in manchen Aspekten mehr versuchen wollten, als es eigentlich nötig war. Unsere Klassenstruktur ist mit der Entwicklungszeit schlecht gewachsen, sodass sich später einige Vererbungen und Klassen als redundant herausstellten. Zu Beginn war eine Generalisierung von allen analogen und digitalen Bauteilen geplant. So hätte der Farbsensor und der Luftdrucksensor von AnalogPart geerbt und eine getter Methode für die Rohwerte der Sensoren via ADC übernommen. Diese hätten diese Rohwerte interpretieren, umrechnen und in Pascal, bzw. RGB-Werten zurückliefern sollen.



Das war aber nur bedingt möglich und auch nicht gefordert. Ohne diese Funktionen erwiesen sich die Klassen als redundant, dasselbe gilt für die digitalen Bauteile. So wurde diese Idee dann im Verlauf der Entwicklung entfernt.

Ein weiterer Punkt ist die Entwicklung eines komplexen Menüs für das Display. Es war geplant jegliche Steuerung über den Rotary Encoder zu ermöglichen. Das hätte mit mehreren Menüs, einem Cursor und einer Fokuserkennung implementiert werden sollen.
Die Implementierung war bereits abgeschlossen aber bei der Definition der einzelnen Menüs haben wir dann den Speicher überschritten. So waren wir innerhalb von kürzester Zeit gezwungen, ein wesentlich simpleres Interaktionssystem zu implementieren.



Diese Rückschläge bzw. Anpassungen des Systems waren aber ein guter Ausblick in die spätere Softwareentwicklung. Nicht alle Pläne können immer umgesetzt werden und die Rahmenbedingungen sind oft ein wichtiger Punkt. (Insbesondere bei eingebetteten Systemen)

Außerdem waren wir gefordert zusammenzuarbeiten und aus drei verschiedenen Ansätzen und Wünschen einen gemeinsamen Lösungsansatz zu entwickeln, was nicht immer ganz einfach war. Jedoch können wir abschließen sagen, dass wir dennoch zufrieden mit unserer Implementierung einer Sortieranlage sind.

Das Arbeiten mit der Programmiersprache C++ war anfangs eine Herausforderung, doch wir haben erhebliche Fortschritte bei der Anwendung dieser Sprache gemacht. Während unserer Projektarbeit im Bereich Embedded Systems konnten wir wertvolle Erfahrungen sammeln und unser Verständnis für die Programmierung vertiefen. C++ erwies sich als schnell und flexibel, insbesondere im Umgang mit hardware-naher Programmierung. Trotz anfänglicher Schwierigkeiten haben wir durch kontinuierliches Lernen und Ausprobieren ein erfolgreiches Projekt abgeschlossen.