

Projektdokumentation: Kameragesteuerter Aufräumroboter

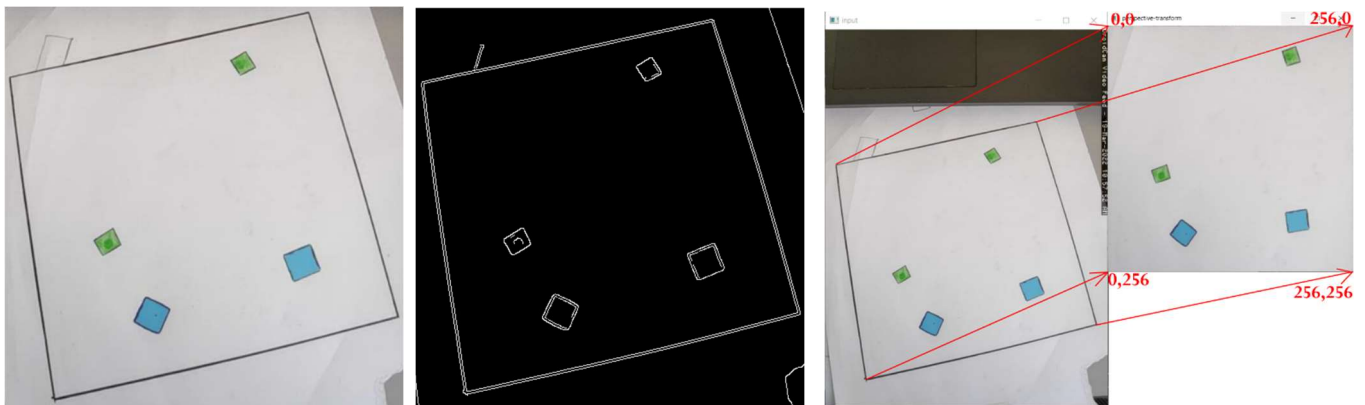
Benjamin Schmid Ties, Ruben Chizzali

Einleitung:

Wie werden in der Industrie der Zukunft Maschinen flexibler und kosteneffizienter gesteuert? Die Antwort lautet nicht selten: Bilderkennung. Auch die Firma Progress Maschinen & Automation AG, bei der ich mein Schulpraktikum absolviert habe, nutzt diese Technik intensiv. Bei meinem Projekt geht es darum, in einem Bild die Koordinaten von metallischen Objekten zu erkennen und diese automatisch wieder an ihren Ursprungsort zuzückzubringen, ohne von einem Menschen gesteuert zu werden, oder wahllos den gesamten Bereich, in dem die Maschine operiert, abzufahren. Die Maschine besteht aus mehreren Bestandteilen, die Hand in Hand arbeiten.

PC-Software:

Bilderkennung ist zwar für einen Menschen sehr einfach, für einen Computer erfordert sie jedoch viel Rechenleistung. Da ein Microcontroller die nötige Rechenleistung nicht hat, habe ich die Software auf einen PC ausgelagert. In mehreren Schritten wird das aufgenommene Bild verarbeitet, um daraus Informationen zu erlangen. Hier sind nur



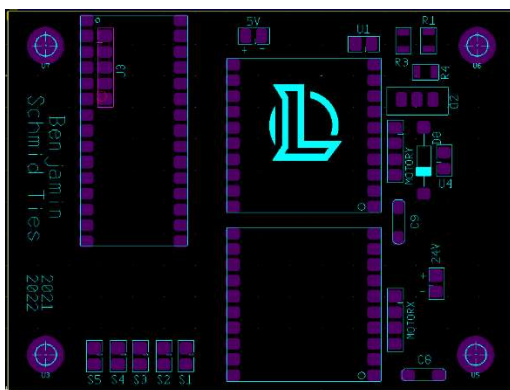
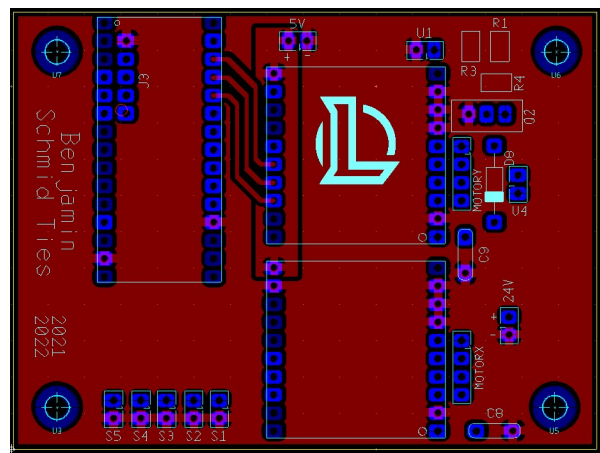
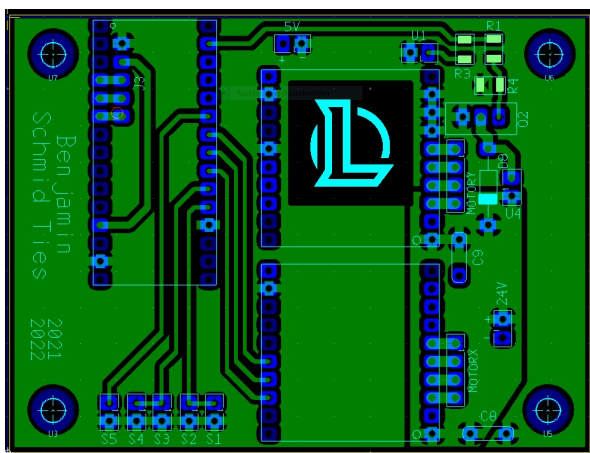
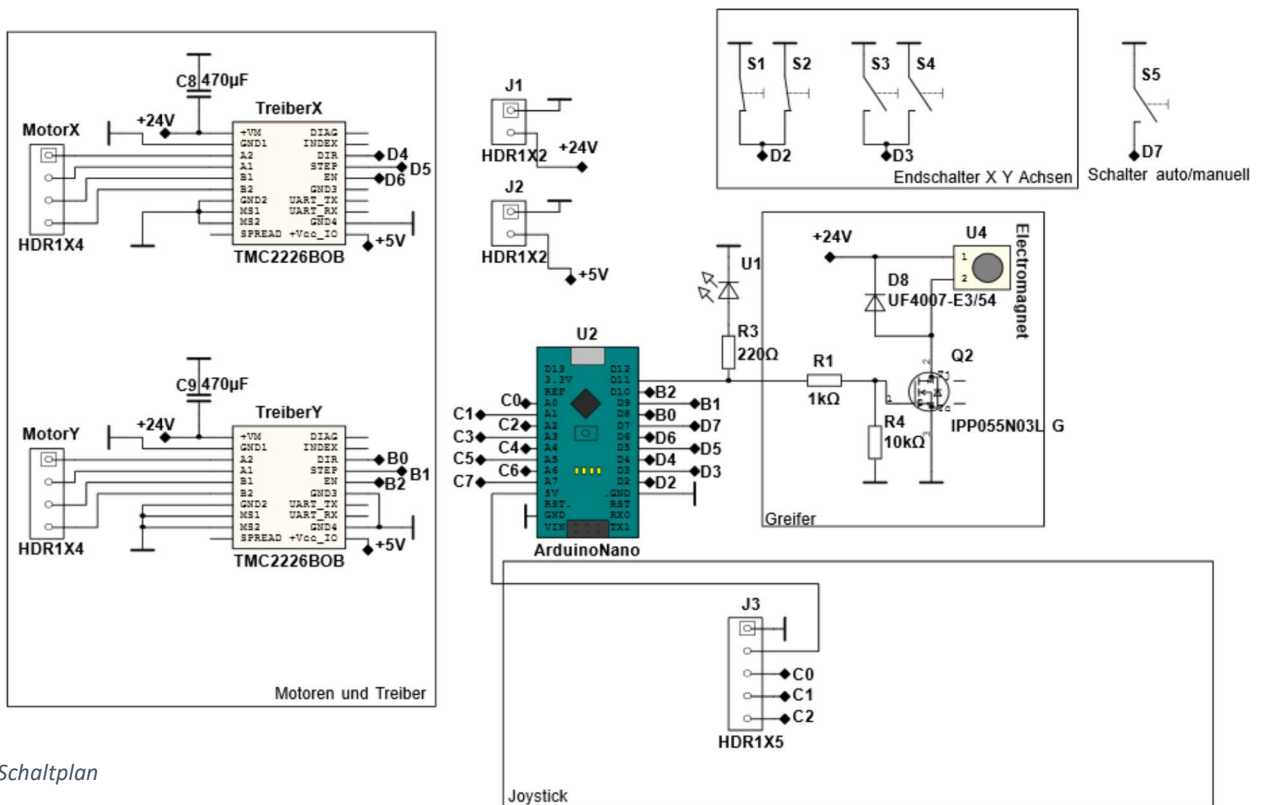
die wichtigsten Schritte abgebildet. Zuerst werden auf dem Bild Konturen (abrupte Farbänderungen zwischen benachbarten Pixeln) erkannt. Es ergeben sich sehr viele Punkte (hunderte), die mit dem Ramer-Douglas-Peucker-Algorithmus zu Eckpunkten mehrerer Vierecke zusammengefasst werden. Anhand der Eckpunkte aller Vierecke im Bild wird deren Fläche berechnet. Das größte Viereck stellt den Bereich dar, in dem die Maschine operiert. Es wird eine Matrix berechnet, die jeden Punkt im eventuell gedreht und perspektivisch verzerrt fotografierten, größten Viereck des anfänglichen Bildes so transformiert, dass er nun innerhalb eines senkrecht abgebildeten und nicht rotierten Quadrats liegt. Die Mittelpunkte der metallischen Teile, die ebenfalls als Vierecke erkannt werden, werden mit dieser Matrix transformiert und die Koordinaten, die sich ergeben, sind nun die Koordinaten der metallischen Teile relativ zum größten Viereck im Bild. Die Seitenlänge des Quadrats, in dem sie liegen, ist 256, ein Byte. Somit können sie sehr einfach an den Microcontroller über die serielle Schnittstelle geschickt werden.

MCU-Software:

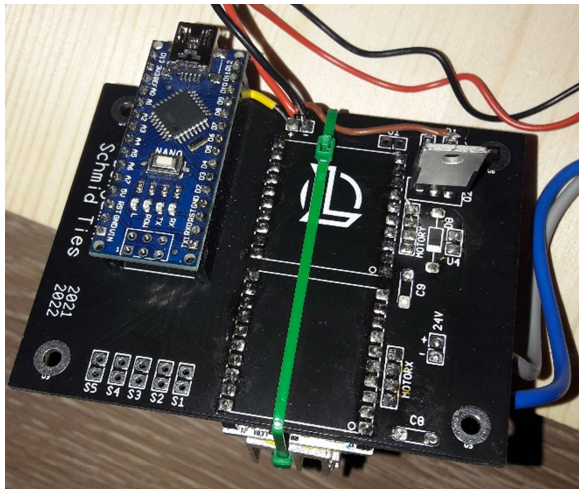
Die Software des Microcontrollers besteht im Wesentlichen daraus, die gesendeten Koordinaten zu empfangen, daraus eine Anzahl an Schritten zu berechnen, die von Schrittmotoren gemacht werden muss, um die Metallteile zu erreichen, und die Schrittmotoren und einen Elektromagneten richtig anzusteuern. Dazu müssen folgende Größen manuell gemessen und im Programm eingetragen werden: der Umfang des Zahnrads der Schrittmotoren, die Größe des Quadrats, in dem sich die Metallteile befinden, und den Abstand des linken unteren Punkts des Quadrats zum 0-Punkt der Maschine. Ein Teil des Programms, bei dem ich lange gebraucht habe, den richtigen Lösungsweg zu finden, war die lineare Beschleunigung der Schrittmotoren. Ich bin auf folgenden Lösungsansatz gekommen: wenn man eine Beschleunigung a wählt (Einheit: Schritte/Sekunde²) und zu jedem Zeitpunkt im Programm die vergangene Zeit t berechnet (über den Zählwert eines vom Systemtakt abhängigen Zählers) kann man die Schritte s , die der Motor bis dahin gemacht haben sollte, berechnen mit $s = \frac{a \cdot t^2}{2}$. Wenn diese Anzahl an Schritten größer ist, als die Schritte die bereits gemacht wurden, wird ein Schritt gemacht. Die Anzahl an Schritten, die zur Beschleunigung verwendet werden, ist konstant. Zum gleichmäßigen Bremsen der Motoren wird der gleiche Ansatz verwendet. Nun wird, um ein Metallteil zu erreichen, das beispielsweise 10000 Schritte in X- und 8000 Schritte in Y-Richtung entfernt liegt, zuerst 2000 Schritte in X Richtung beschleunigt, dann 6000 Schritte mit konstanter Geschwindigkeit zurückgelegt, und anschließend wieder über 2000 Schritte gebremst. In Y-Richtung passiert das Selbe, nur dass hier 4000 Schritte mit konstanter Geschwindigkeit zurückgelegt werden. Danach wird der Elektromagnet eingeschaltet und der Weg wird rückwärts gefahren. Am Ende wird der Elektromagnet wieder ausgeschaltet. Es gibt auch einen manuellen Modus, in dem man die Motoren mit einem Joystick manuell steuern und durch Klicken des Joysticks den Elektromagneten an- bzw. ausschalten kann.

Elektronik:

Grundlegend besteht der Schaltplan meines Projekts aus einem Microcontroller (Atmega 328p), 2 Schrittmotoren und den dazugehörigen Treibern, und einem Elektromagneten. Da alternativ zum automatischen Ansteuern der Schrittmotoren auch ein manueller Modus praktisch ist, habe ich einen Joystick vorgesehen, mit dem der Benutzer die Motoren steuern kann. Der Elektromagnet wird durch ein Logik-Mosfet geschaltet, und die induzierten Spannungen über eine schnelle Diode kurzgeschlossen. An beiden Achsen sind jeweils Endschalter vorgesehen, um eventuelle Schäden durch Fehler zu vermeiden. Versorgt wird die Schaltung von einem 24V Netzgerät mit 120 Watt und die Logikspannung der Treiber wird durch ein 5V Netzgerät zur Verfügung gestellt. Die elektrische Beschaltung der Schrittmotor-Treiber wurde von Ruben Chizzali geplant und wir haben die Platinen der Treiber professionell anfertigen lassen (JLC-PCB).

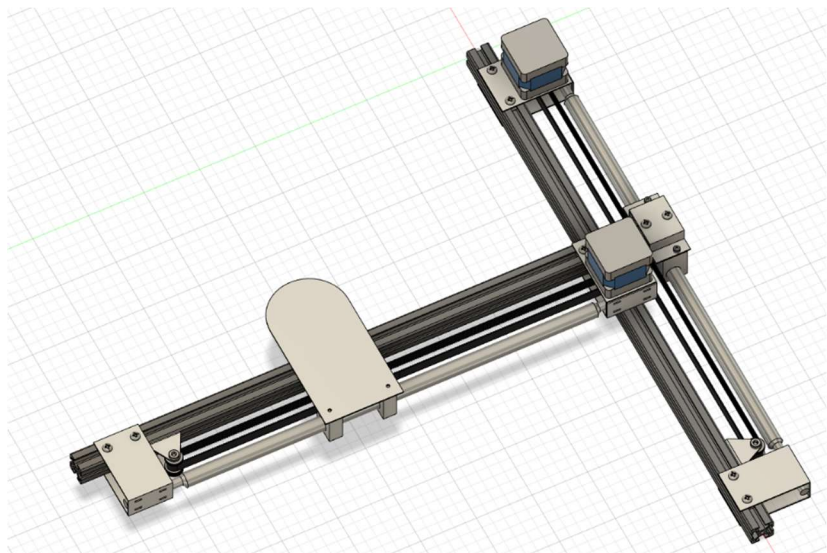


Für die Schaltung habe ich ein Platinen-Layout mit Ultiboard entworfen und anfertigen lassen. Die Platine ist zweilagig und beinhaltet weibliche Header-Pins zum Verbinden des Arduino Nano und der Schrittmotor-Treib/er Platinen.



Mechanischer Aufbau:

Um die Maschine zu bauen haben wir Führungen, Motoren, Gleitlager und Riemen aus einem kaputten Drucker recycelt. Ruben Chizzali hat einen 3D-Plan in Fusion-360 konstruiert, und wir haben uns beim Aufbau danach gerichtet. Das Grundgerüst besteht aus Aluminium-Profilen,



die über Inox Bleche miteinander verbunden sind und an denen mit Blechen die Führungen und Riemen angebracht sind. Am Ende der X-Achse befindet sich ein Rad. Auch die Platte, an der der Elektromagnet angebracht ist, hat ein Rad, damit sie auf dem Aluprofil rollt.