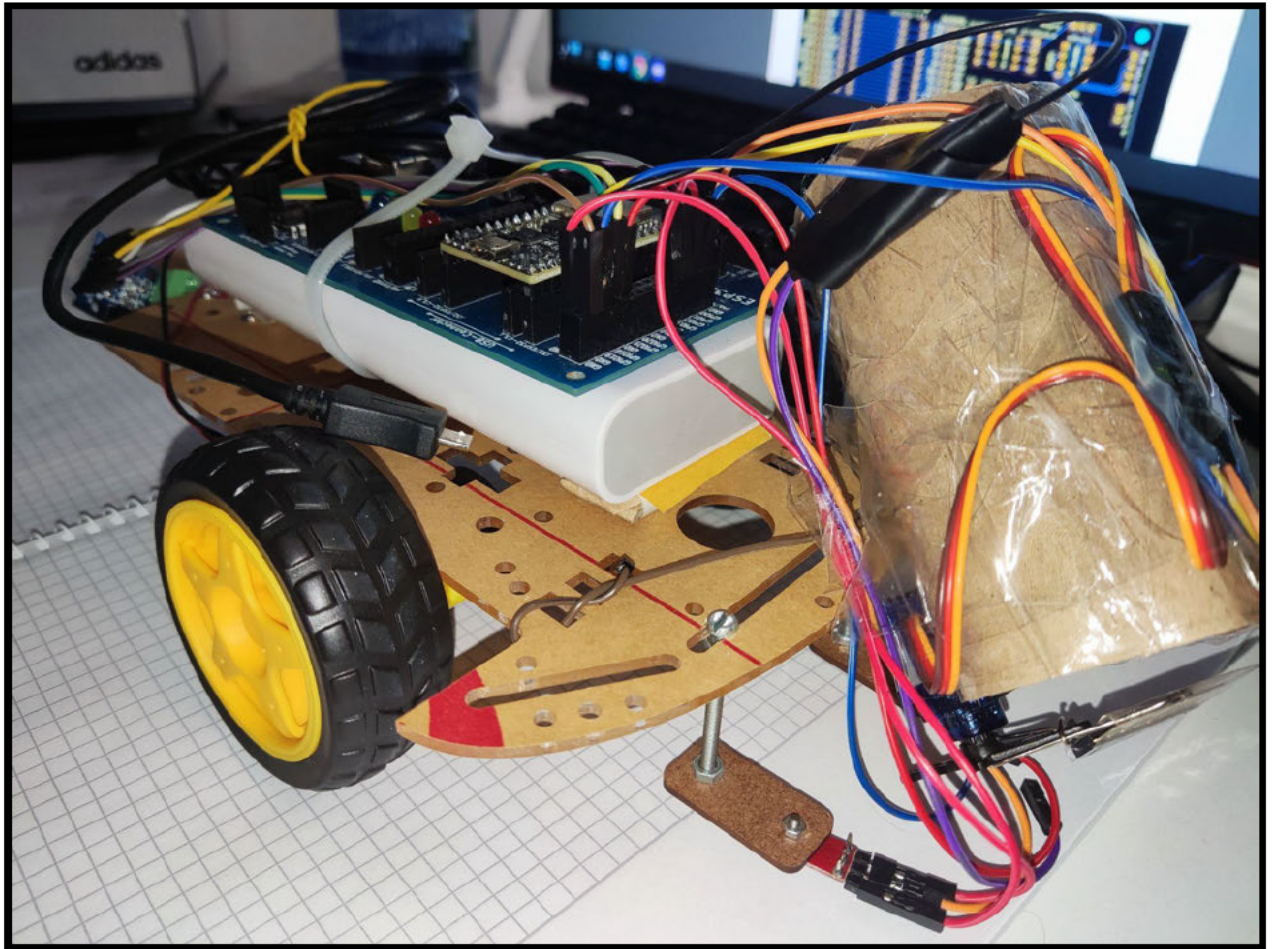


Entwicklung und Konstruktion eines Roboters zur Ballübertragung durch einen Tunnel



Hochschule Niederrhein Krefeld, Fachbereich 03 Informatik

████████ Celal Agit Köseoglu, ██████████ ██████████

05.01.2023

Einleitung

Das Ziel des Erstsemesterprojekts ist es, einen Roboter zu entwickeln und zu bauen, der in der Lage ist, einen aufgenommenen Ball durch einen Tunnel auf das Zielfeld zu transportieren. Der Roboter sollte ferngesteuert werden und im Tunnel einer auf dem Boden markierten Linie folgen.

Um diese Aufgabe zu lösen, wurden verschiedene Bauteile und elektrische Komponenten zur Verfügung gestellt, wie zum Beispiel Räder, DC-Motoren, eine Plattform, ein Schwenkrad, Mutter, Schrauben, Foto-Sensoren, eine Powerbank, eine H-Brücke und Dupont-Kabel.

Aufgabenverteilung und Arbeitsweise in der Gruppe

Die Gruppe hat gemeinsam die Stärken und Schwächen des Jeweiligen abgeschätzt. Wir sind zum Entschluss gekommen, dass es gravierende Überschneidungen unserer Kenntnisse gibt und wir zusammen am gesamten Projekt arbeiten müssen, da jedermanns Stärken und Schwächen sowohl im Softwaretechnischen als auch Hardwaretechnisch anders ausgeprägt sind. Somit haben wir uns gegenseitig bei den Aufgabenfeldern ergänzt.

Der Arbeitsfortschritt wurde in regelmäßigen Meetings besprochen und abgestimmt. Jedes Teammitglied hatte die Möglichkeit, seine Ideen und Vorschläge einzubringen und gemeinsam wurde entschieden, welche Lösung am besten geeignet war.

Beschreibung der Lösung

Strategie

Die Lösungsidee war es, den Roboter mit zwei Rädern und zwei DC-Motoren auszustatten, um ihm die nötige Beweglichkeit zu geben. Um den Ball aufnehmen und transportieren zu können, wurde eine Plattform mit einem Schwenkrad konstruiert, das durch einen Micro-Servo gesteuert wurde. Der Roboter konnte so den Ball aufnehmen und ihn zum Ausgang des Tunnels bringen, wo er ihn dann in den Zielbereich fallen lassen konnte.

Um im Tunnel der Linie folgen zu können, wurden zwei Foto-Sensoren an der Unterseite des Roboters befestigt. Diese Sensoren waren in der Lage, die Linie zu

erkennen und dementsprechend die Geschwindigkeit und Richtung der Räder anzupassen.

Der Schaltplan für die Hardware-Konstruktion ist im Anhang dieser Dokumentation als Datei "Schaltplan_HOLY_ESP.pdf" dargestellt. Der Schaltplan zeigt, wie die einzelnen Bauteile miteinander verbunden werden müssen, um den Roboter zu betreiben.

Die Software für den Roboter wurde in der IDE von Arduino entwickelt. Der Code ist im Anhang dieser Dokumentation als Datei "HOLY_ESP_CODE.ino" enthalten.

Vorbereitung

Um uns auf das Projekt vorzubereiten, haben wir als Team zunächst die vorgegebenen Materialien aus dem ESP-Moodle-Kurs sorgfältig durchgearbeitet. Dabei haben wir uns mit den Grundlagen der Arduino-Plattform vertraut gemacht und uns über die verschiedenen Funktionen und Befehle informiert, die für unser Vorhaben von Bedeutung sein werden.

Anschließend haben wir eine erste Skizze gezeichnet, in der wir unsere Ideen und Vorstellungen für das Projekt visualisiert haben. Dies hat uns dabei geholfen, unsere Gedanken zu ordnen und einen Überblick über die einzelnen Schritte und Arbeitsprozesse zu erhalten.

Um unsere Kenntnisse zu vertiefen und uns auf mögliche Herausforderungen vorzubereiten, haben wir außerdem den Arduino Crash Course auf YouTube angeschaut. Dieser Video-Kurs gibt einen umfassenden Überblick über die wichtigsten Konzepte und Funktionen von Arduino und bietet zahlreiche Beispiele und Übungen, die dabei helfen, die erlernten Inhalte zu verinnerlichen. Insgesamt haben wir uns also gründlich und umfassend auf das Projekt vorbereitet und fühlen uns gut gerüstet, um mit der Arbeit zu beginnen.

Motoren zum Laufen bringen

Um unseren Roboter für das Projekt zum Laufen zu bringen, haben wir zunächst nur die Motoren verbunden, um zu testen, wie diese funktionieren. Dabei haben wir gelernt, dass sich die Richtung der Motoren ändert, wenn man die Polarität der Stromversorgung umkehrt. Auf diese Weise können wir das Fahrzeug vorwärts oder rückwärts bewegen, indem wir die Polarität der Stromversorgung entsprechend umkehren.

Nachdem wir uns mit den Motoren vertraut gemacht hatten, haben wir uns mit der H-Brücke beschäftigt, die verwendet wird, um die Motoren zu steuern. Die H-Brücke ist ein elektronisches Schaltungsdesign, das es ermöglicht, den Stromfluss durch die Motoren zu kontrollieren und somit die Geschwindigkeit und Richtung des Fahrzeugs zu verändern.

Nachdem wir die H-Brücke verstanden hatten, haben wir das Fahrzeug zusammengebaut und die ersten Testfahrten im Makerspace durchgeführt. Dabei haben wir die Funktionsweise der Motoren und der H-Brücke in der Praxis ausprobiert und unseren Roboter erfolgreich zum Laufen gebracht.

Handysteuerung Testen

Beim ersten Versuch haben wir versucht, irgendeinen Output im seriellen Monitor zu bekommen, um sicherzustellen, dass die Verbindung zwischen dem Arduino-Mikrocontroller und dem Handy funktioniert. Dazu haben wir den Code so angepasst, dass alle Variablen, die wir für die Steuerung des Roboters verwenden würden, mit `Serial.print` ausgegeben werden. Auf diese Weise konnten wir uns alle Werte in Echtzeit ansehen und somit leichter Fehler finden und beheben.

Nachdem wir sichergestellt hatten, dass die Verbindung zwischen dem Arduino-Mikrocontroller und dem Handy funktionierte und alle Variablen mit `Serial.print` ausgegeben wurden, um sie in Echtzeit ansehen zu können, haben wir uns an die Implementierung der Panzersteuerung gemacht. Hierzu haben wir zunächst einige Tests durchgeführt, um sicherzustellen, dass jedes einzelne Element der Steuerung ordnungsgemäß funktioniert. Sobald wir sicher waren, dass jedes Element ordnungsgemäß funktionierte, haben wir die Steuerung in ihrer Gesamtheit getestet, um sicherzustellen, dass der Roboter tatsächlich wie gewünscht gesteuert werden konnte.

Linienfolger

Um die automatische Steuerung mit Fotosensoren zu entwickeln, haben wir zunächst die Sensoren an der Vorderseite des Autos befestigt. Dies war ein wichtiger Schritt, da die Sensoren das Fundament für unsere automatische Steuerung bilden würden. Anfangs haben wir jedoch einen Fehler gemacht: Wir haben versucht, die Werte mit der `digitalRead`-Funktion auszulesen. Das Problem hierbei war jedoch, dass `digitalRead` nur Werte zwischen 1 und 0 liefern kann, was uns in unserem Fall nicht weitergeholfen hat.

Da wir den Grenzwert, an dem digitalRead zwischen 0 und 1 gewechselt hat, nicht kannten, haben wir uns entschieden, stattdessen die analogRead()-Funktion zu verwenden.

Um herauszufinden, wann das Auto genügend Kraft hatte, um loszufahren, mussten wir durch Trial and Error vorgehen.

Zunächst haben wir versucht, if-Statements in der loop-Funktion zu verwenden, um die automatische Steuerung zu implementieren. Leider hat dies jedoch nicht so funktioniert, wie wir uns das vorgestellt hatten, und wir mussten stattdessen while-Schleifen verwenden. Das Problem bei der Verwendung von while-Schleifen ist, dass sie den REMOTEXYHandler() nicht aufrufen, was bedeutet, dass die Input-Variablen vom Handy nicht geupdatet werden und somit nicht eingesehen werden können. Das heißt, selbst wenn wir auf Handy-Steuerung wechseln, muss der Roboter erst alle while-Schleifen durchlaufen, bevor die Steuerung durch das Handy übernommen werden kann. Dies kann in manchen Fällen zu Verzögerungen bei der Steuerung führen und ist somit etwas, das wir in Zukunft verbessern möchten.

Servo

Um den Servo zu installieren, damit er den Ball aus der Rolle herausnehmen konnte, haben wir ihn zunächst mit dem Arduino-Mikrocontroller verbunden. Zu unserer Freude hat alles auf Anhieb funktioniert und der Servo konnte wie gewünscht gesteuert werden.

Nachdem wir einige Zeit damit verbracht hatten, den Servo über die Panzersteuerung zu steuern, haben wir beschlossen, auf Joystick-Steuerung umzusteigen. Leider war dies jedoch sehr fehlerhaft und wir mussten schließlich wieder auf Panzersteuerung umsteigen, da der Joystick zu inkonsistent war.

Plötzlich hat jedoch gar nichts mehr funktioniert und wir konnten keinen Code mehr auf das Board laden. Nachdem wir eine Weile herumgefummelt hatten, haben wir schließlich herausgefunden, dass die Platine teilweise defekt war. Glücklicherweise befanden wir uns im Raum F110, wo wir das Problem lösen konnten.

Reflexion

Was hat besonders gut funktioniert?

Als IT-Frischlinge waren wir besonders begeistert und haben uns als Gruppe zusammengeschlossen, um unsere Kreativität zum Vorschein zu bringen. Da alle Mitglieder aus der Gruppe verschiedene Stärken besitzen, hatten wir den Vorteil, uns gegenseitig zu ergänzen und haben voneinander besonders viel gelernt. Als Neulinge haben wir die Möglichkeit genutzt, uns als Gruppe zu integrieren und um einen erfolgreichen Start bzw. um ein neues Kapitel an der Hochschule Niederrhein zu beginnen. Da Zusammenarbeit an Hochschulen das A und O ist, um voneinander zu lernen und sich inspirieren zu lassen, haben wir die ersten Eindrücke dafür gesammelt und werden in Zukunft unsere Zusammenarbeit stärken und in neuen Projekten zum Vorschein bringen. Die Möglichkeit, sich mit den Professoren und Mitarbeitern auszutauschen, gefiel uns besonders stark und diente uns im ersten Semester als Willkommen. Als eine kreative und neugierige Gruppe haben wir alle Möglichkeiten und alle Plattformen genutzt, um die Kommunikation der Gruppe beizubehalten und uns auszubauen. Durch die Nutzung von Kommunikationsplattformen haben wir uns an geplanten Terminen gehalten und uns ausgetauscht. Möglichkeiten, wie z.B. Besuche des Makerspaces und Absolvieren der Lasercutter-Unterweisungen haben unsere Teamarbeit gestärkt und dienten als Willkommen. Außerdem haben wir uns für die nächsten Projekte ausgerüstet, da wir auch Erfahrungen mit Makerspace und Lasercutter besitzen.

Was hat nicht so gut funktioniert und wie könnte man es beim nächsten Mal besser machen?

Leider hatten wir auch Probleme und Hindernisse, die wir zusammen bewältigen mussten. Da dies unsere erste richtige Erfahrung mit Software und Hardware ist, wussten wir anfangs nicht, wie wir dieses Projekt angehen sollen. Und konnten uns dementsprechend nicht gut koordinieren. Das hat dafür gesorgt, dass uns der Anfang schwerer gefallen ist, als es hätte sein müssen.

Außerdem konnten wir kurz vor Abschließung des Projekts keinen Code mehr auf das Board laden. Da wir bis dato auch keine Erfahrungen mit Fehlersuche/Debugging eines Hardware Projektes hatten, hat uns das auch viel Zeit gekostet.

Zusammenfassung

Das Erstsemesterprojekt hat das Ziel verfolgt, einen Roboter zu entwickeln und zu bauen, der in der Lage ist, einen Ball durch einen Tunnel zu transportieren und ihn in den Zielbereich zu befördern. Diese Aufgabe wurde erfolgreich gelöst, indem ein Roboter konstruiert wurde, der über zwei Räder und zwei DC-Motoren verfügte und mit Hilfe von Foto-Sensoren die Linie im Tunnel verfolgen konnte. Die Software wurde in C++ programmiert und konnte den Roboter ferngesteuert betreiben. In Zukunft wäre es sinnvoll, früher mit der Hardware-Konstruktion zu beginnen, um genügend Zeit für die Feinschreibung der Software zu haben.