



MCB Mobil

Marco Stauber

Ben17A

Inhaltsverzeichnis

Abstract.....	3
Zusätzliche Infos.....	3
Änderungen.....	4
Funktionen	5
Probleme.....	6
Fazit.....	6

Abstract

Die erste Version des Controllers funktionierte nicht mehr und hatte viele Designfehler. Am einfachsten ging es mit einem neuem Design und einem neuem Print.

Das Ziel war es das bereits begonnene Projekt „MCB Mobil“ fertig zu stellen und ein zweites Fahrzeug für Herrn Malacarne zusammen zu bauen.

Zusätzliche Infos

Alle Dateien für das Projekt können auf dem folgenden GitHub gefunden werden:

<https://github.com/MXACE/MCB-Mobil>



Änderungen

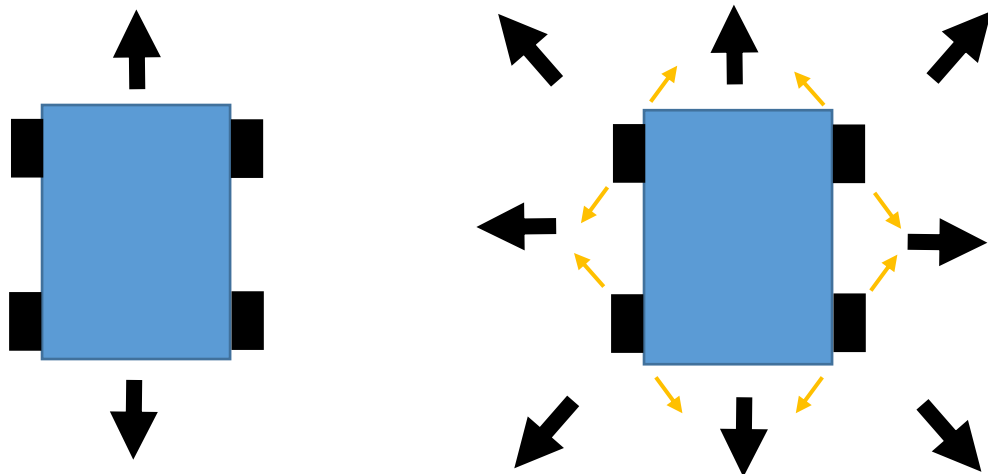
Anstelle eines RN2483-Breakoutboard das ich selbst gemacht habe, befindet sich direkt ein solches Modul auf dem Print. Dieses ist zuständig für die Kommunikation mit dem Fahrzeug. Zur Sicherheit habe ich den Footprint des Breakoutboards trotzdem hinzugefügt, falls das Modul auf dem PCB nicht funktioniert.

Da das mit dem Bootloader nicht geklappt hat, konnte ich ihn ganz weg lassen. Somit sparte ich nicht nur Platz, sondern auch noch Geld und viele Bauteile wie ein zweiter Mikrokontroller oder ein USB Anschluss.

Die Joysticks befinden sich nun auf der gleichen Höhe vom Print. Ein Fehler den ich gemacht habe war, dass ich den linken Joystick nicht um 45° gedreht habe. Dies würde es vereinfachen das Fahrzeug zu steuern, da es mit den Mecanum Rädern andere Fahrtrichtungen hat.

Diese Räder besitzen Rollen die im 45° Winkel montiert wurden. Somit verändert es die Fahrtrichtung.

In der linken Abbildung sieht man die Fahrtrichtungen mit herkömmlichen Rädern. In der rechten Abbildung sieht man die orangen Kräftevektoren der einzelnen Räder. Kombiniert man diese beliebig, kann man mit dem Fahrzeug in alle Richtungen fahren.



Neben den zusätzlichen Debug-LEDs gibt jetzt noch eine Schaltung für die Speisung. Ein PMOS stellt die Speisung der Batterie hochohmig, sobald sich eine Spannung am USB-Port befindet. Dies soll verhindern, dass die Batterie sich schnell entlädt.

Ein Achsensensor, der BMG250 von Bosch, sollte eine zweite Möglichkeit sein um das Fahrzeug zu steuern, indem man den Controller in die jeweilige Richtung kippt.

Funktionen

Um das eine Problem mit den Joystick zu beheben, musst ich nur die XY Koordinaten davon um 45° drehen. Durch eine Anleitung auf YouTube konnte ich einfach herausfinden wie es funktioniert.

$$\alpha = 45^\circ \rightarrow \cos(\alpha) = \sin(\alpha) = \frac{\sqrt{2}}{2} = \sqrt{0.5} \approx 0.707$$

$$x_{45} = x \times \cos(\alpha) - y \times \sin(\alpha) = x \times \sqrt{0.5} - y \times \sqrt{0.5}$$

$$y_{45} = x \times \sin(\alpha) + y \times \cos(\alpha) = x \times \sqrt{0.5} + y \times \sqrt{0.5}$$

Da der Sinus und der Cosinus aus der Math.h Library den ganzen Speicher füllen würden nahm ich anstelle von $\cos(45^\circ)$ und $\sin(45^\circ)$, 0.707 für die Rechnung.

Da mir die Zeit fehlte kam ich nicht weit beim Kommunikationsprotokoll zwischen dem Fahrzeug und dem Controller. Also war die Lösung: eine Kommunikation nur in eine Richtung. Der Controller schickt alle 0.5 Sekunden eine Nachricht mit allen Joystick Werten. Im Fall das eine Nachricht ausfallen sollte und das Fahrzeug 0.5 Sekunden lang keine Daten bekommt, bleibt es stehen. Der Controller muss daher einfach neugestartet werden. Das Fahrzeug wartet ein paar Sekunden und geht dann wieder in RX Modus. Gleichzeitig sieht man das an den LEDS um das Fahrzeug.

Der Bildschirm ist nun ein wenig überflüssig. Dieser zeigt nur noch an wann der Controller bereit ist für den Start und die Steuerung des Scheinwerfers.

Probleme

Der Achsensensor hat mich sehr viel Zeit gekostet. Das Hauptproblem war die Kommunikation mit dem Chip. Designt habe ich ihn für SPI. Nach nahezu endlosen neuprogrammieren und neu löten wurde es immer noch nichts. Als ich es mit einem STM Discovery und der Breakout Version des BMG250 über I2C versuchte ging es. Auf dem Kontroller wiederrum nicht, also hab ich das ganze sein gelassen.

Bei dem Aufsetzten der Unidirektionalen Kommunikation gab es auch ein Paar Fehler die aber auf der Priorität der Interrupts zurückzuführen war.

Fazit

Nun im letzten Semester konnte ich mein Fahrzeug endlich fertig stellen. Das mit dem Achsensensor fand ich eine coole Idee, wurde aber nur anstrengend und hat mich zu viel Zeit gekostet. Für das ich gesamthaft Eineinhalb Jahre an diesem Projekt dran war, hat es mir sehr viel Freude bereitet. Es war nicht immer einfach daran zu arbeiten, da viele Abschlussprüfungen und sonst noch welche Dinge dazwischen kamen.

Das Fahrzeug fährt jetzt immerhin und es funktioniert grösstenteils. Es gäbe noch ein paar Verbesserungen, wie zum Beispiel eine Akkuschutzschaltung oder einen kleineren Print, die man bei der nächsten Version machen könnte.