

# GPS auf Rädern – Gruppe B

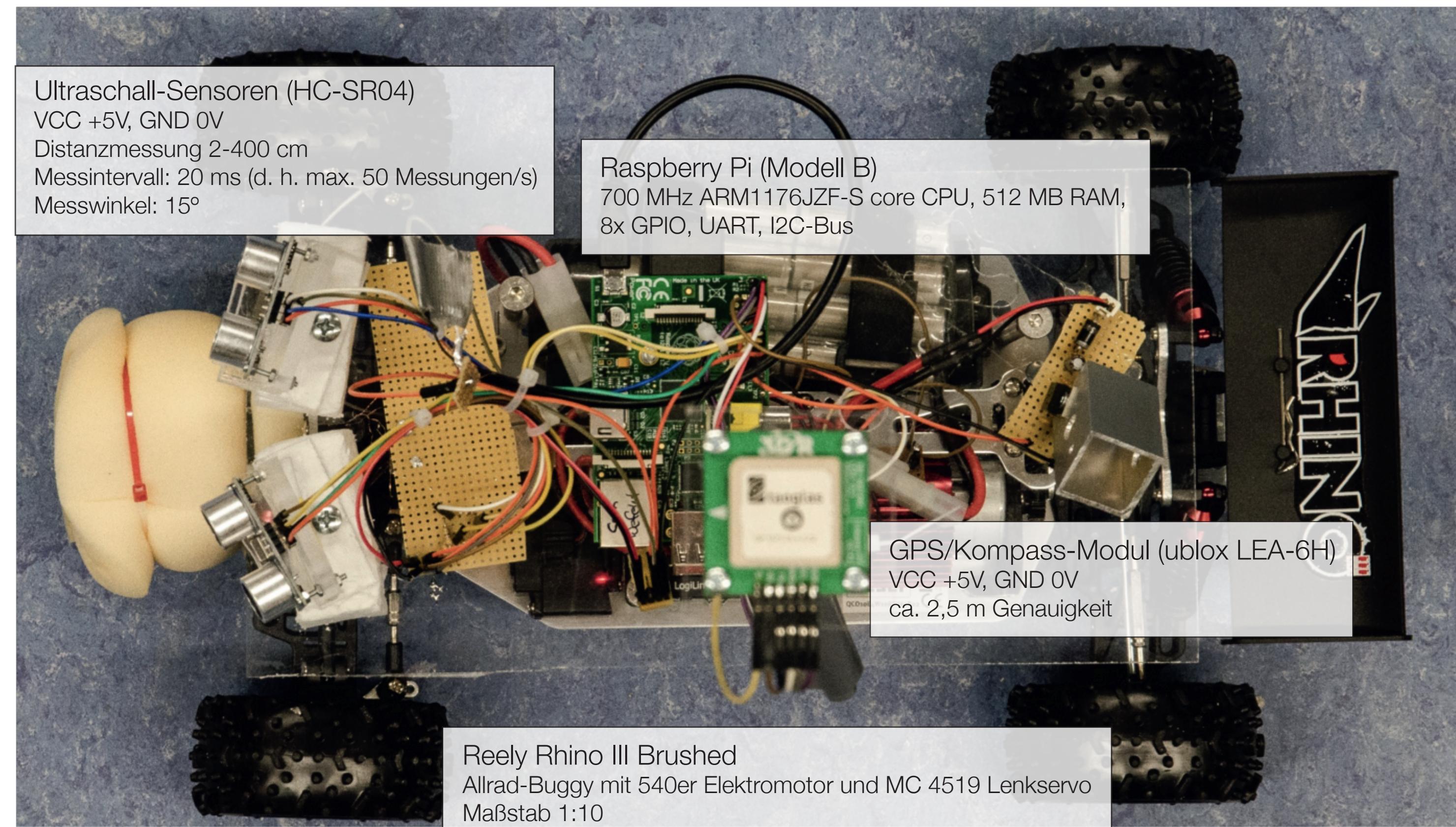
Gloria Feher, Jonathan Förste, Jannis Andrija Schnitzer

Betreut von Gero Plettenberg und Thomas Kloepfer

## Aufgabenstellung des Projekts

Das Ziel des Projektes ist es, ein Modellauto mittels eines Raspberry Pis und GPS-Moduls zu autonomer Geländefahrt zu befähigen. Nach Erhalt von entsprechenden Zielkoordinaten soll der Roboter mithilfe von Modul und externen Sensoren seinen Weg auch durch schwieriges Gelände finden und Hindernissen ausweichen können.

## Ergebnis



## Software

- Raspbian
- gpsd library
- RPI.GPIO library
- Selbstentwickelter Code für:
- GPS-Datenverarbeitung und Routenplanung
- Ansteuerung des Motorreglers und Lenkservos des Autos
- Distanzmessung
- Hinderniserkennung und -verarbeitung

## Funktionsweise

Die Funktionalität des Roboters lässt sich in vier größere Abschnitte gliedern: Die Hardware-Ansteuerung, welche Schnittstellen für die Verwendung der einzelnen Komponenten bietet, die Navigationsroutine, welche berechnet, wie der Roboter fahren muss um an die Zielkoordinaten zu gelangen, die Hinderniserkennung, welche die Daten der Ultraschallsensoren verarbeitet, Hindernisse erkennt und ihnen ausweicht und die Hauptroutine, welche alle Software-Komponenten miteinander verbindet.

## Hauptroutine

Das Programm wird mit GPS-Zielkoordinaten als Parametern aufgerufen.

Es wird auf ein GPS-Signal gewartet. Sobald dieses vorliegt, wird in eine Endlosschleife eingetreten und zunächst überprüft, ob das Auto gerade stillsteht.

Ist dies der Fall, so steht es noch am Anfang seiner Reise. Daher muss zunächst die Orientierung bestimmt werden, damit der Navigator den Kurs berechnen kann.

Dann wird überprüft ob das Auto bereits an seinem Ziel angekommen

ist. Falls ja, hält es an und das Programm bricht aus der Schleife aus. Ist es nicht an seinem Ziel, wird überprüft ob sich das Auto auf dem richtigen Kurs befindet und gegebenenfalls der Kurs korrigiert. Die letzte if-Anweisung prüft, ob das Auto gerade fährt – und wenn dem so ist, speichert es seine Position in die Liste der letzten Positionen des GPSTrackers und prüft während der Fahrt auf Hindernisse.

## Hardware-Ansteuerung

Lenkservo und Fahrtmotor werden über den Raspberry Pi mithilfe von PWM-Signalen (Pulse-Width-Modulation) angesteuert. Hierbei gibt die Länge des gesendeten Signals an, wie die Motoren reagieren (siehe Exemplarische PWM-Signale und Reaktionen der Motoren).

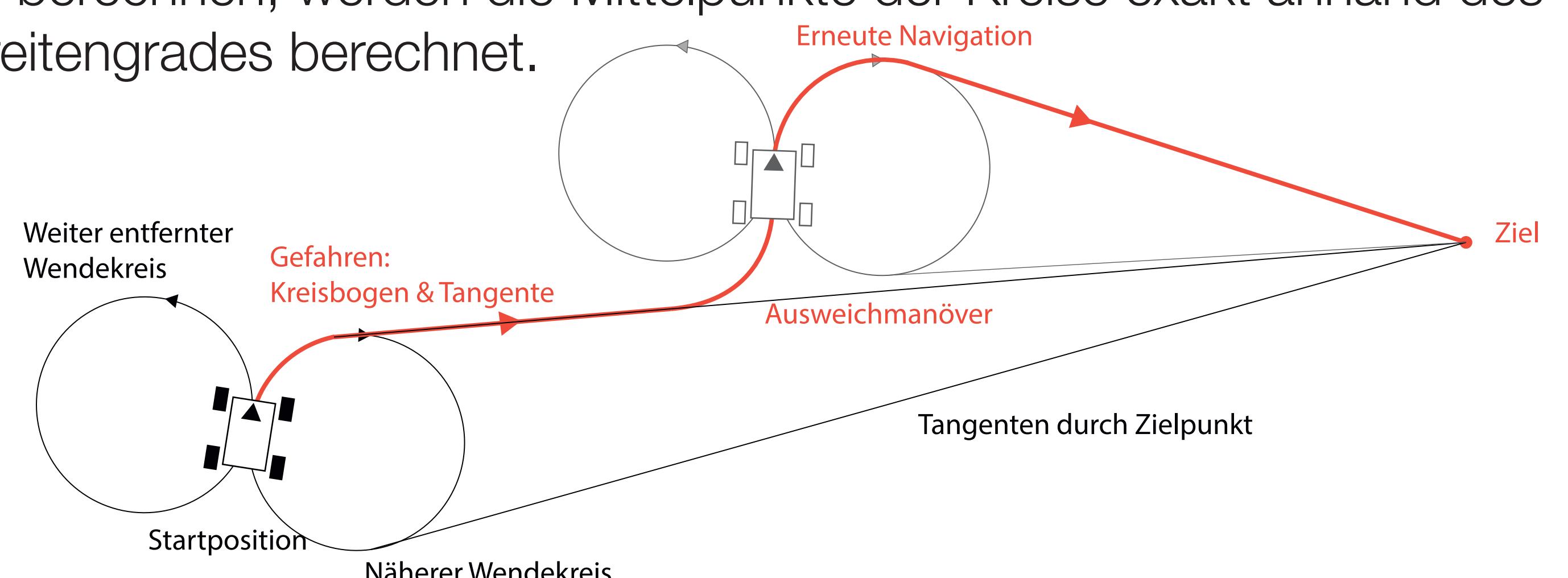
Das GPS-Modul ist über UART (Universal Asynchronous Receiver Transmitter; serielle Schnittstelle) mit dem Pi verbunden. Der eingebaute Kompass wird über I2C (Inter-Integrated Circuit; serieller Datenbus) angesteuert. Hierfür werden jeweils die designierten UART und I2C Pins des Raspberry Pis sowie die gpsd Library verwendet.

Die Ultraschallsensoren werden über eine kleine Schaltung, die garantieren soll, dass der Pi keine Signale über 3.3V erhält, mit dem Minicomputer verbunden (siehe Schaltplan: Verbindung der Ultraschallsensoren mit dem Raspberry Pi). Um die Distanz zu messen, wird ein Ultraschall-signal gesendet (trig). Dann wird die Zeit gemessen, die vergangen ist bis das Signal zurückgeworfen wurde (echo). Die vergangene Zeit wird mit der Schallgeschwindigkeit (343 m/s) multipliziert und durch zwei geteilt, da die Strecke zwei mal zurückgelegt wurde. Somit erhält man die Distanz zum nächsten Objekt.

## Navigation

Die Navigation funktioniert prinzipiell folgendermaßen: Ausgehend von der Position und Orientierung des Roboters und eines Kurvenradius, der im Voraus eingestellt wird, werden zwei Kreise betrachtet, die der Roboter bei entsprechendem Lenkeinschlag nach links oder rechts fahren kann. Es wird dann der Kreis ausgewählt, dessen Mittelpunkt näher am angesteuerten Zielpunkt liegt, und Tangenten an diesen Kreis durch den Zielpunkt gelegt. Diejenige Tangente, die auf dem Kreisbogen näher am Roboter liegt, wird dann angestrebt.

Zu beachten ist bei der Berechnung, daß wir direkt auf GPS-Daten, d. h. Längen- und Breitengradangaben arbeiten. Die geometrische Berechnung erfolgt im Wesentlichen kartesisch, d. h. die Genauigkeit ist nur bei vergleichsweise geringen Distanzen (einige zehn Kilometer sollten kein Problem sein) gewährleistet. Um trotzdem keinen Unsinn zu berechnen, werden die Mittelpunkte der Kreise exakt anhand des Breitengrades berechnet.



## Hinderniserkennung

Die Hinderniserkennung basiert auf Ultraschallsensoren und lässt sich mit der elektronischen Einparkhilfe vieler Moderner Kfz vergleichen. Die Ultraschallsensoren senden jeweils einen kurzen Schallpuls und fangen ihn nach Reflexion an Hindernissen wieder auf. Aus der verstrichenen Zeit kann dann der Abstand berechnet werden.

Mithilfe von zwei nebeneinander angeordneten Sensoren kann das Auto entscheiden, in welcher Richtung sich das Hindernis befindet und entsprechend ausweichen. Dabei passt es die Stärke der Lenkung der Entfernung zum zu umfahrenden Objekt an und kann so flüssige Manöver ausführen.

