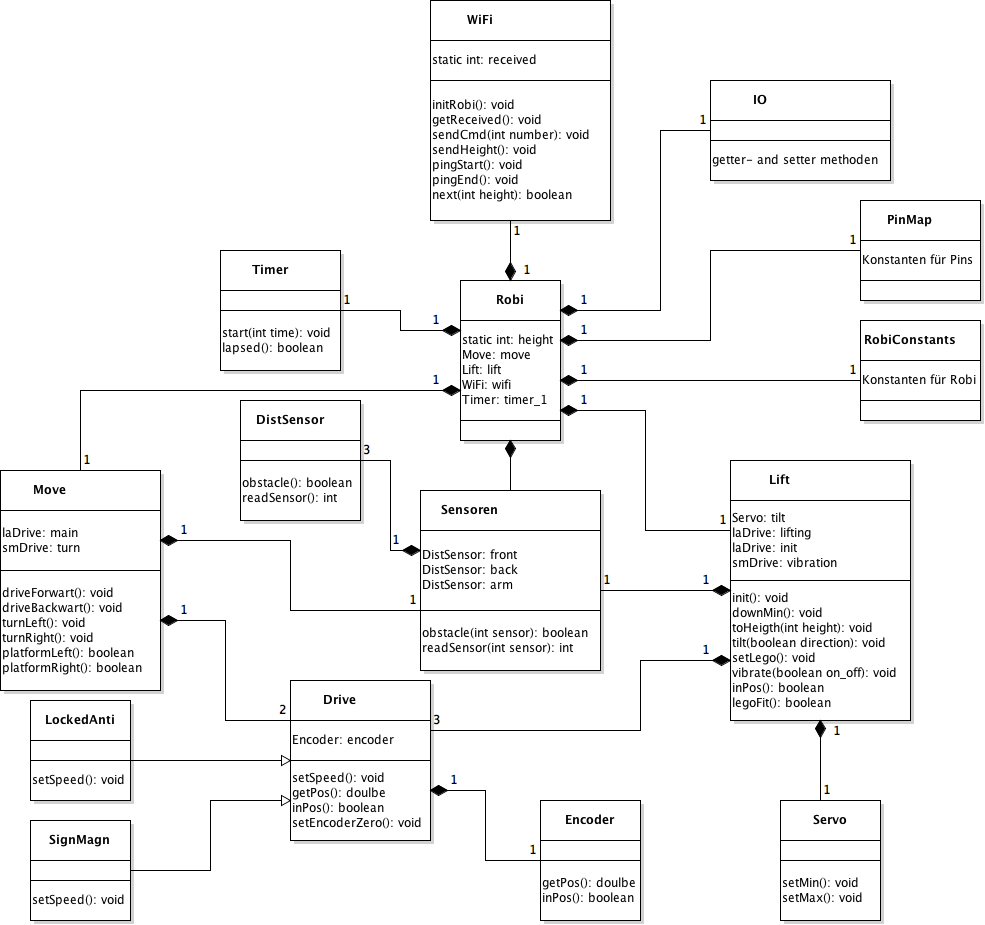
Klassendiagramm



Die Klasse Robi ist die Klasse, von der aus der Ablauf gesteuert wird. Alle Funktionen anderer Klassen werden von dieser Klasse aus gesteuert. Die verschiedenen Zustände werden im Zustandsdiagramm genauer erläutert.

Die Klasse Timer beinhaltet eine Stopuhr, durch welche man Zeitabstände, die nicht durch Tasks gesteuert werden, messen kann.

Die Kommunikation wird durch die Klasse Wifi gesteuert. Der genaue Ablauf der Kommunikation wird im Kommunikationsdiagramm erklärt.

Die Klasse IO instanziiert Ein- und Ausgänge, in der Klasse PinMap sind alle benutzten Pins aufgeführt. Diese Klassen helfen, Ordnung zu halten, da man immer weiss, wo ein Pin zu finden ist und man mit einem Klick den Wert für alle Aufrufe dieses Pins verändern kann. In RobiConstants werden Konstanten gespeichert, um diese mit einem Klick verändern zu können.

Sensoren beinhaltet eine Liste der DistSensoren, diese beiden Klassen verwalten die Distanzsensoren und beinhalten Methoden um Hindernisse zu erkennen oder zu testen, ob ein bestimmter Grenzwert erreicht wurde.

Servo steuert den Servomotor, für dieses Projekt wurde nur die Maximal- und Minimalposition gebraucht.

Die Klasse Encoder speichert die Position eines Motors. Ein Encoder kann mit einem Faktor initialisiert werden, damit alle Übersetzungen unterstütz werden.

Drive ist eine abstrakte Klasse, die entweder vom Typ LockedAnti (=Locked Antiphase) oder SignMagn (=Sign Magnitude) ist. Die verschiedenen Ausprägungen müssen die Methode setSpeed überschreiben, damit man beide Arten von Motoren gleich ansteuern kann. Ausserdem wird beim Instanziieren eines Motors festgelegt, ob dieser über einen Encoder gesteuert wird.

Die Klasse Move verwaltet die Bewegung des Roboters und steuert zwei Sensoren sowie ein Locked-Antiphase-Motor ohne Encoder für die Bewegung auf dem Spielfeld und ein Sign-Magnitude-Motor mit Encoder für die Drehung des Roboters.

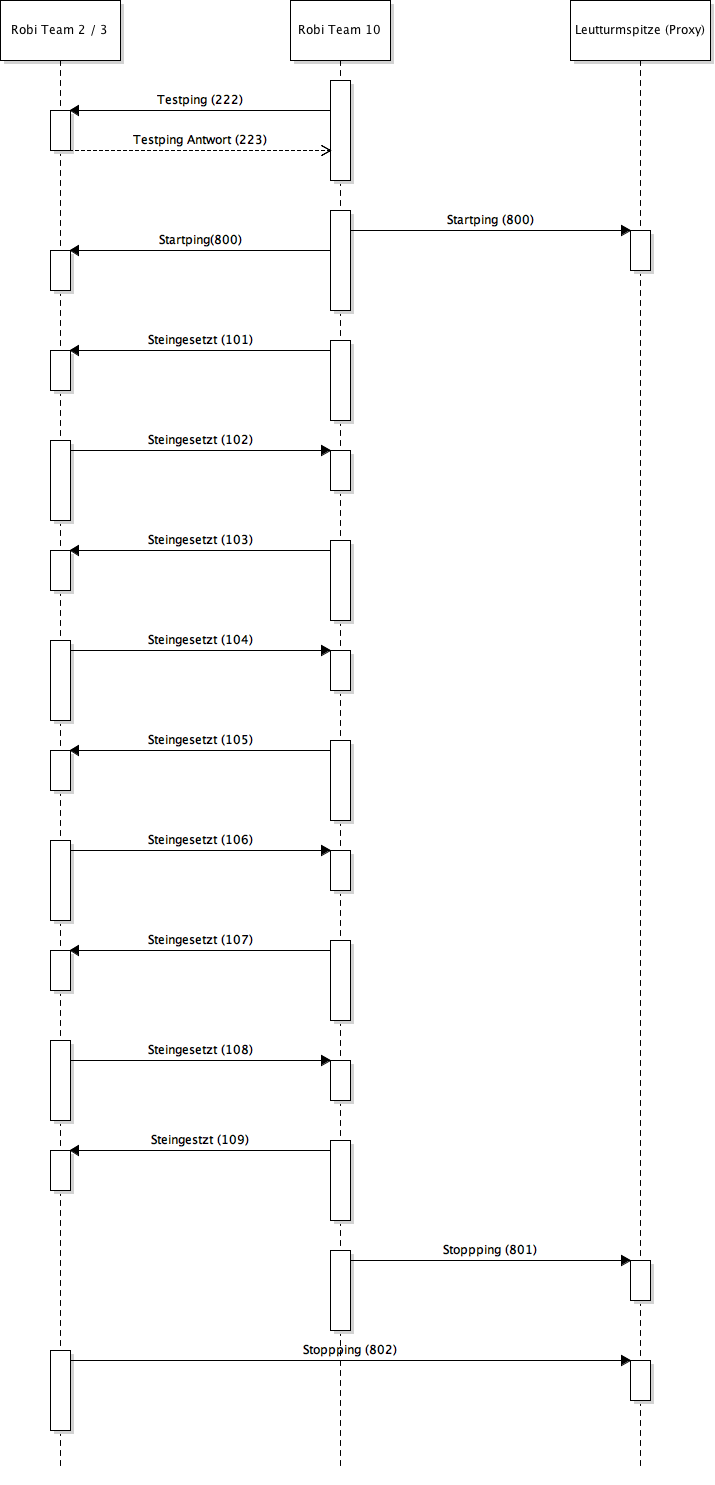
Die Klasse Lift ist ähnlich wie die Klasse Move, steuert aber den Greifmechanismus.

Beide Klassen stellen Methoden für die Klasse Robi zur Verfügung, um diese Klasse möglichst übersichtlich zu halten.

Kommunikation

Die Kommunikation wird über Fire&Forget-System gemacht, es wird keine Bestätigung für das erhalten einer Nachricht verschickt. Die einzige Ausnahme dazu ist das Signal 222, das zu Beginn verschickt wird, um zu Überprüfen, ob der Partnerroboter bereit ist und die Kommunikation steht. Falls dies nicht der Fall ist, wird das Signal nach kurzer Zeit wiederholt. Sobald die Bestätigung kommt, wird das Startsignal an den Leuchtturm und den Partnerroboter geschickt.

Während des Bauens wird jeweils das Setzen eines Steins mit einer Nachricht an den Partnerroboter bestätigt, um Komplikationen mit der Nummer 10, die beim Testen Probleme verursachte, zu vermeiden, wird jeweils die Höhe in Steinen+100 gesendet. Sobald unser Roboter den letzten Stein gesetzt hat, schickt er das Schlusssignal an den Leuchtturm.



Informatik Webseite

Die Software, die Klaus zum Laufen bringt, wurde mit Hilfe der Programmiersprache «Java» entwickelt. Um das Programm übersichtlicher zu machen, und das Testen zu vereinfachen, wurden Teilsysteme verwendet, die jeweils wenige Aufgaben übernehmen müssen.

Sowohl um sich bewegen und drehen zu können, also auch zum Bewegen des Armes sind Motoren nötig. Diese können sich zu einer beliebigen Position bewegen, und, falls nötig, diese Position auch halten.

Die Kommunikation mit dem Partnerroboter funktioniert über WLAN. Zu Beginn wird ein Signal gesendet und vom Partner bestätigt, um die Verbindung zu kontrollieren. Ausserdem wird ein Startsignal, eine Bestätigung für das Setzten eines Steines und das Endsignal verschickt.

Die Orientierung des Roboters findet über Distanzsensoren statt. Die letzten Werte werden gemittelt, falls ein neuer Wert ausschlägt, weiss der Roboter, dass ein Hindernis bei diesem Sensor ist.

Damit während des Durchgangs Probleme von aussen sichtbar sind, werden einige Status über LEDs sichtbar gemacht. Die wichtigste Information ist, ob die Verbindung mit dem Partnerroboter steht.

Um diese Teilsystem miteinander zu koordinieren, wird eine Hauptklasse gebraucht, die den Ablauf managet und den anderen Klassen ihre Aufgaben zuteilt.