



Bauvorschlag: Radarwagen

12

Bei diesem Modell haben wir einen Wagen gebaut, über dem sich eine Radarantenne ständig dreht. Sobald etwas in ihr Sichtfeld gelangt, ändert der Wagen seine Richtung, bis er wieder auf ein Hindernis stößt.



Probiere aus, wie lange der Radarwagen in Bewegung bleibt, ohne dass du eingreifen musst.

Du ahnst es sicherlich schon, dass die Antenne vor allem aus dem Entfernungssensor von LEGO Boost besteht. Damit sich das Kabel des Sensors nicht verwickelt, dreht er sich immer nur einmal um die eigene Achse und dann wieder zurück. Deshalb musst du aber auch darauf achten, dass du ihn beim Start des Programms immer so ausrichtest, dass der Sensor gerade nach vorne schaut. Das Kabel muss direkt zum Eingang vom Move-Hub gehen und sollte nicht um irgendetwas gewickelt sein.

Programmteil, der den Sensor dreht.



Das Programm besteht aus drei verschiedenen Abläufen. Der obere Programmteil ist nur dafür da, den Sensor immer hin und her zu bewegen. Außerdem haben wir hier zur Demonstration eine Variable eingebaut. Ganz am Anfang wird diese auf den Wert »15« gesetzt (also eine »15« in den Kasten »a« gelegt). Das wird bei den nächsten beiden Programmen noch wichtig.

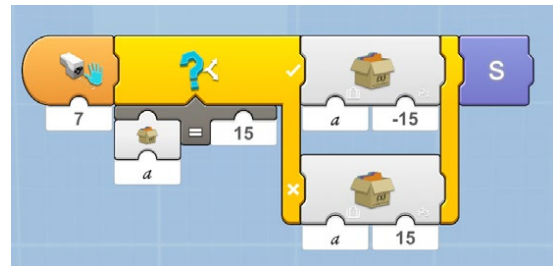


Programmteil zum Fahren des Wagens.

Denn in der nächsten Zeile befindet sich das Programm, das den Wagen fahren lässt. Dazu verwenden wir eine Schleife. In der ist ein Befehl programmiert, sodass die Motoren für den Antrieb immer wieder eine Sekunde lang bewegt werden. Interessant ist hier, dass wir nicht direkt einen Wert für die Geschwindigkeit eingegeben haben, sondern die Variable »a« verwenden. Der Wert für die Geschwindigkeit entspricht also

dem, was in Kiste »a« drin ist, und das ist zunächst die Zahl »15«. Das bedeutet also, dass der Wagen zunächst einfach mit geringer Geschwindigkeit vorwärts fährt. Wäre die Zahl hingegen negativ (also »-15«), so würde er rückwärtsfahren.

Das spannendste Programm ist die unterste Zeile. Dieses wartet darauf, dass etwas in die Nähe des Sensors gelangt (er also beim Herumschwenken ein Hindernis erkennt). Mit der Entfernungsangabe kannst du ein wenig rumprobieren. Wir fanden einen Wert von »6« oder »7« ausreichend, um zu verhindern, dass der Wagen gegen ein Hindernis fährt. Wenn du den Wagen schneller fahren lässt, dann kann auch ein größerer Wert notwendig sein.



Programmteil zum Auswerten des Sensors und Ändern der Fahrtrichtung.

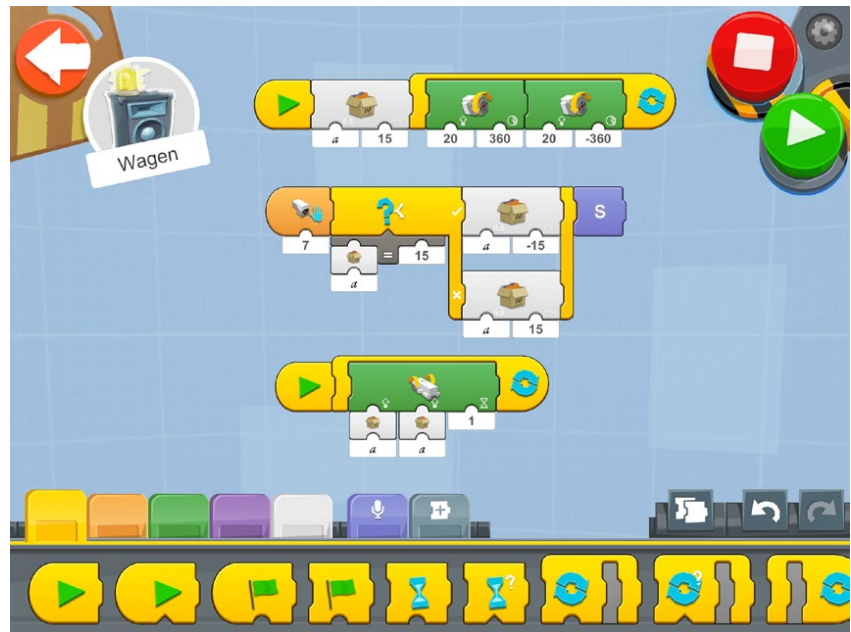
Erkennt also der Sensor, dass ihm etwas zu nahe kommt, dann soll nun der Wagen dazu veranlasst werden, die Richtung zu ändern. Deshalb haben wir hier eine Bedingung eingebaut, die abgefragt wird, wenn der Sensor etwas bemerkt. Ist dann der Wert der Variablen »a« gleich »15«, so wird der Wert in »-15« geändert. Ist er jedoch nicht »15« (er kann dann nur »-15« sein, mehr Möglichkeiten gibt es nicht), dann wird er wieder auf »15« gesetzt. Dadurch ändert der Wagen die Richtung (fährt also bei der »-15« rückwärts und später bei erneutem Wert »15« wieder vorwärts), denn ob die »15« positiv oder negativ ist, ist hierfür ausschlaggebend. Da wir hier die Variable ändern und der mittlere Programmteil immer aus dieser Variablen »a« den Wert für die nächste Sekunde Fahrt nimmt, ändert der Wagen auch tatsächlich die Richtung.

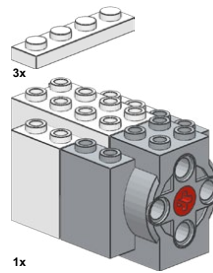
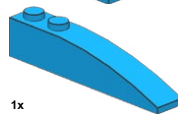
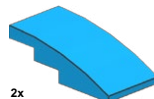
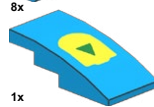
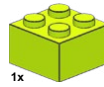
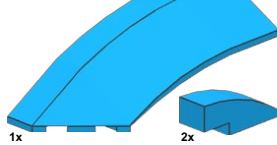
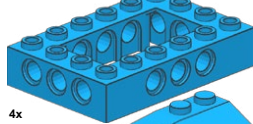
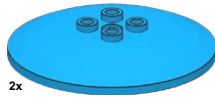
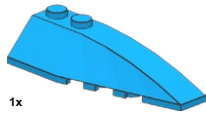
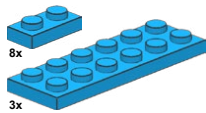
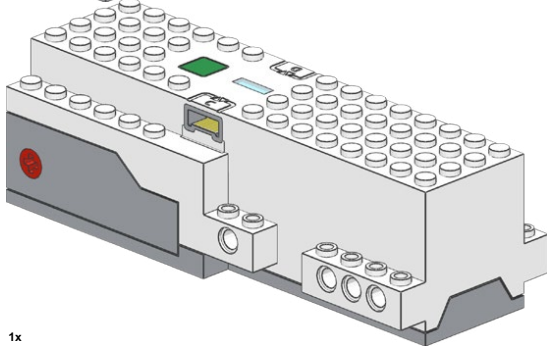
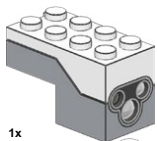
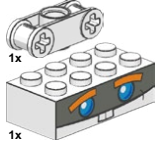
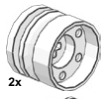
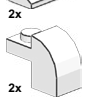
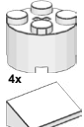
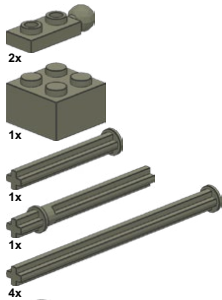


Wenn du möchtest, kannst du ihn bei der Richtungsänderung am Ende vom unteren Programmteil noch einen Ton oder einen selbst eingesprochenen Satz wie »Kollisionswarnung« oder »Auf Aufprall vorbereiten« sagen lassen.

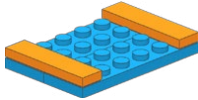
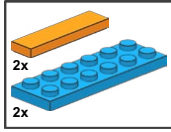
Das Modell und Programm sollen dazu einladen, selber ein wenig auszuprobieren. So könntest du programmieren, dass sich der Wagen nicht nur hin und her bewegt, sondern beim Erkennen des Hindernisses auch eine kleine Kurve fährt. So erreicht er auch andere Bereiche deines Zimmers. Oder du lässt ihn zusätzlich Farben erkennen, wie zum Beispiel von einer Ampel, die du aus Legosteinen baust. Vielleicht hast du auch andere Ideen. Das Schöne ist, dass du eigentlich nichts falsch machen kannst.

Alle drei Programmteile nochmal im Überblick.

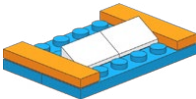




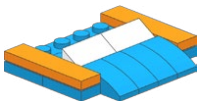
1



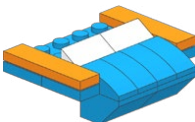
2



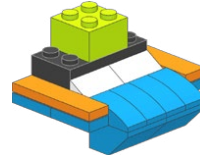
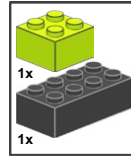
3



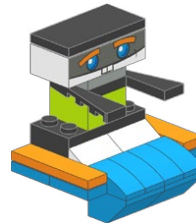
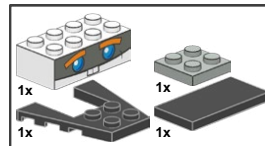
4



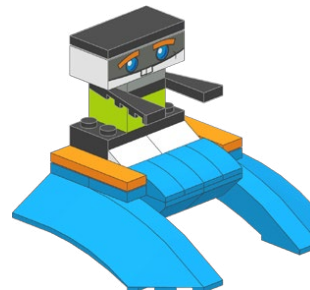
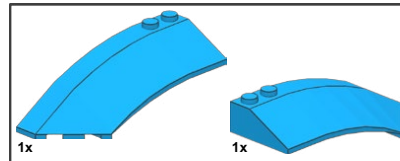
5



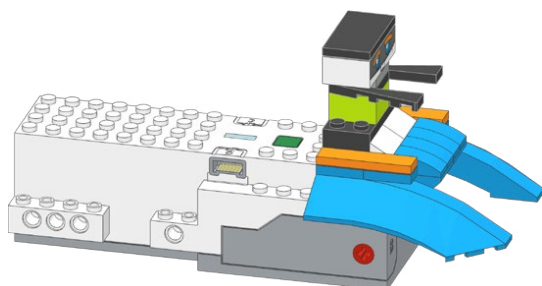
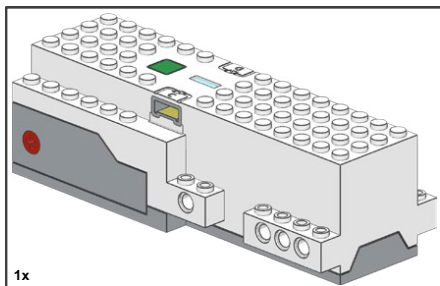
6



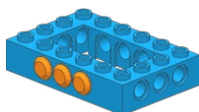
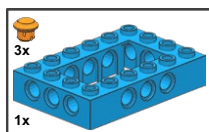
7



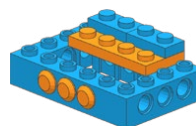
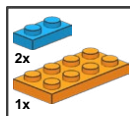
8



9



10



11

