Name: \_\_\_\_\_\_\_\_\_\_ Klasse: \_\_\_ Datum: \_\_\_\_\_\_\_\_

# Lösungsblatt Aufgabe 5

# Balllroboter

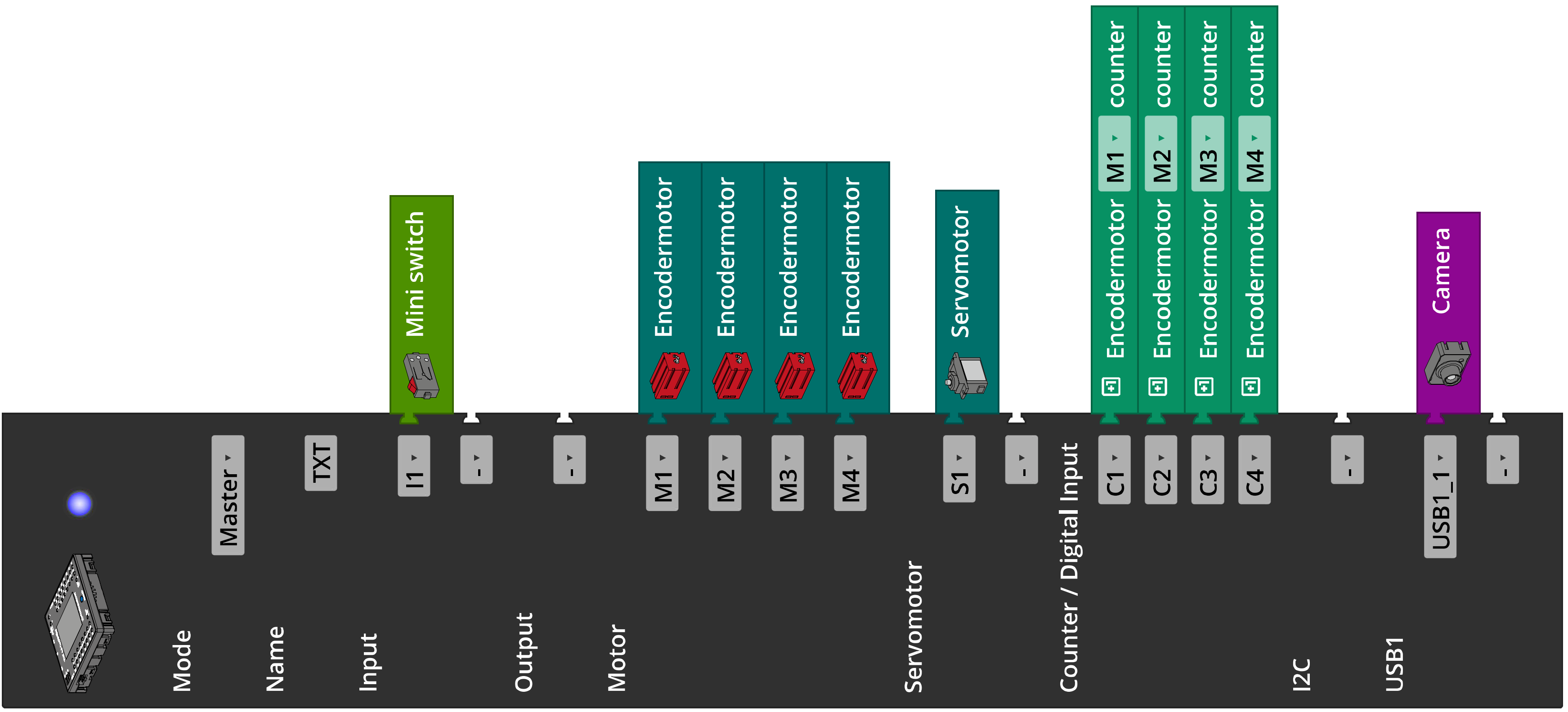
Die Aufgaben sind eine Übung zur Nutzung von Schleifen und Funktionen mit dem Ziel, ein übersichtliches, kompaktes und verständliches Programm zu erhalten. Die Ergebnisse der Schülerinnen und Schüler sollten diesbezüglich miteinander verglichen werden.

## Konstruktionsaufgabe

Siehe Bauanleitung.

## Programmieraufgaben

Konfiguration der Sensoren und Aktoren:



Für die Lösung von Programmieraufgabe 2 wird die „Voice Control“-App (für iOS oder Android) benötigt. Die App muss für die Spracherkennung mit dem Internet verbunden sein und (über Bluetooth oder WLAN) mit dem Controller verbunden werden.

**1. Auswurfmechanismus**

1a. Geeignete Werte für die Variablen „fire“ und „load“ (die gesuchten Positionen des Servo-Hebels) können leicht mit dem Interface-Test bestimmt werden. Sie sind modellabhängig und hängen insbesondere von der Position des aufgesteckten Servo-Hebels ab. Im hier verwendeten Modell liegen sie bei 120 (fire) und 380 (load).

1b. Funktion „fire“ (Beispiel):

Ein Bild, das Text, Screenshot, Schild, Verkauf enthält.

Automatisch generierte Beschreibung

*Mecanum\_Fire\_and\_Load.ft*

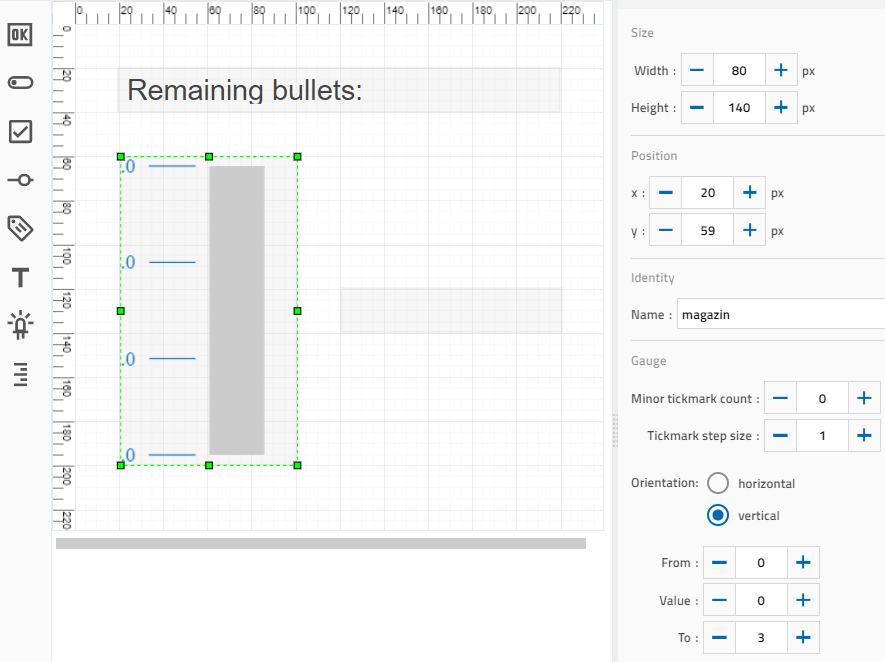
1c. Programmauszug (Beispiel):

Ein Bild, das Text, Schild, Screenshot enthält.

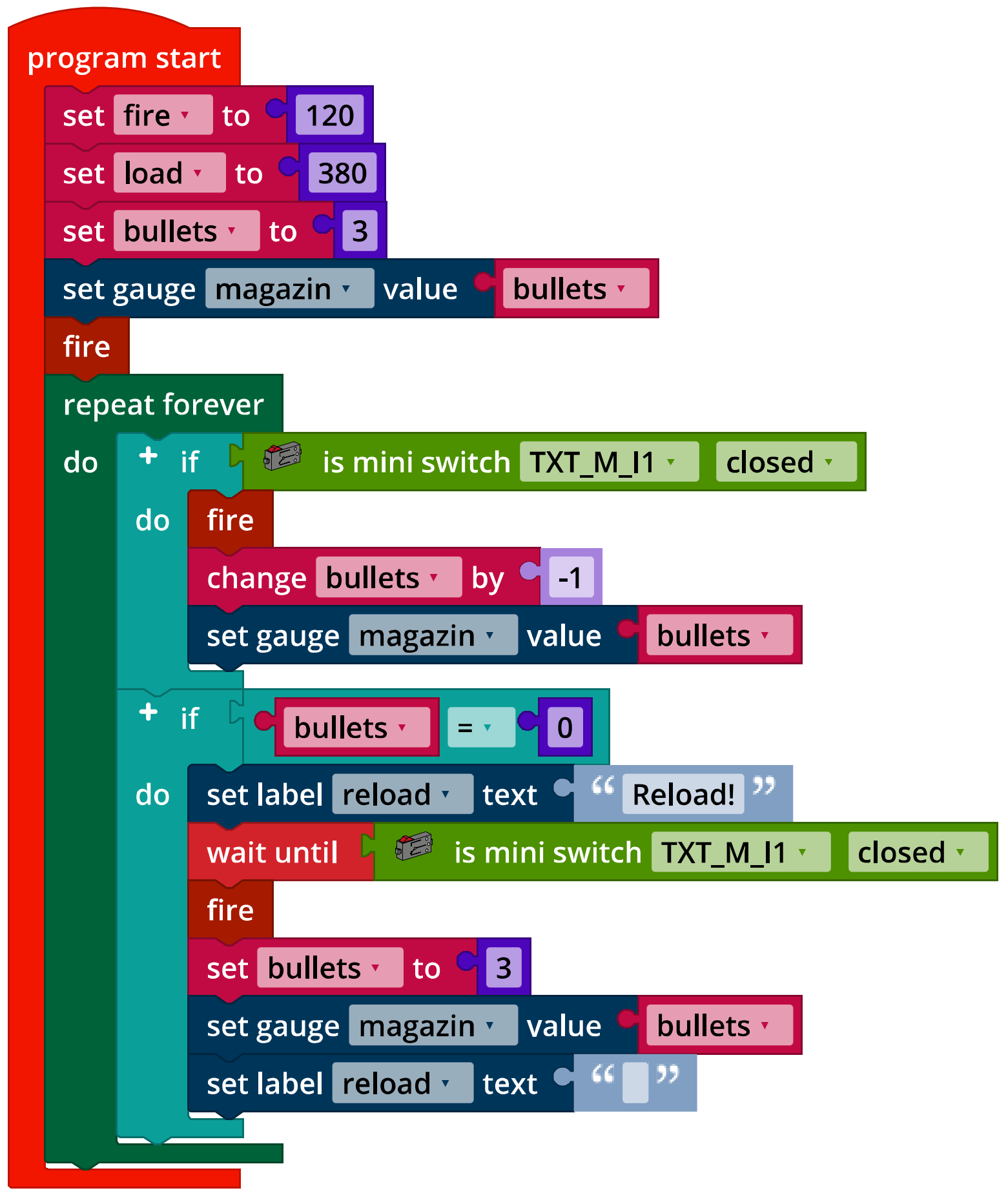
Automatisch generierte Beschreibung

*Mecanum\_Fire\_and\_Load.ft*

1d. Konfiguration des TXT-Displays (Beispiel):



Programm (Beispiel):



*Mecanum\_Fire\_and\_Load\_extended.ft*

**2. Sprachsteuerung**

Programm (Beispiel):

Ein Bild, das Text, Screenshot, drinnen enthält.

Automatisch generierte Beschreibung

*Mecanum\_Fire\_and\_Load\_Voice\_Command.ft*

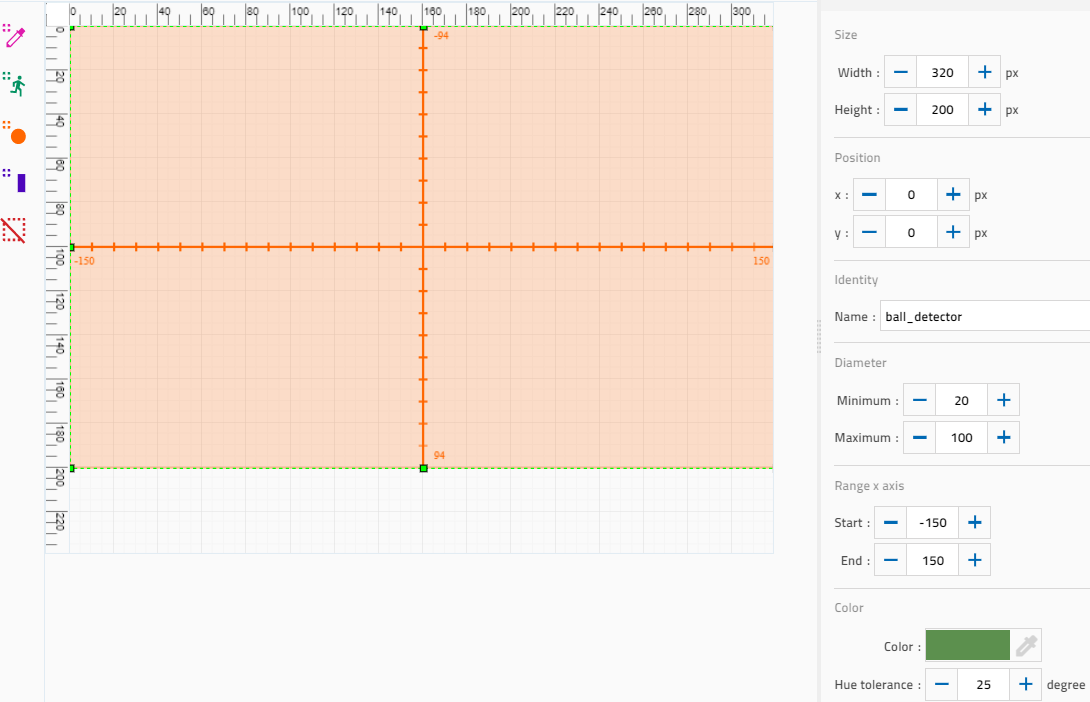
## Experimentieraufgaben

**1. Zielscheibe**

1a. Die Zielscheibe sollte idealerweise so weit entfernt stehen, dass die Kunststoffhohlkugel die Zielscheibe etwas oberhalb der Mitte treffen.

Im verwendeten Beispielmodell liegt dieser ideale Abstand bei etwa 35 cm. Dieser Wert kann von Modell zu Modell leicht variieren.

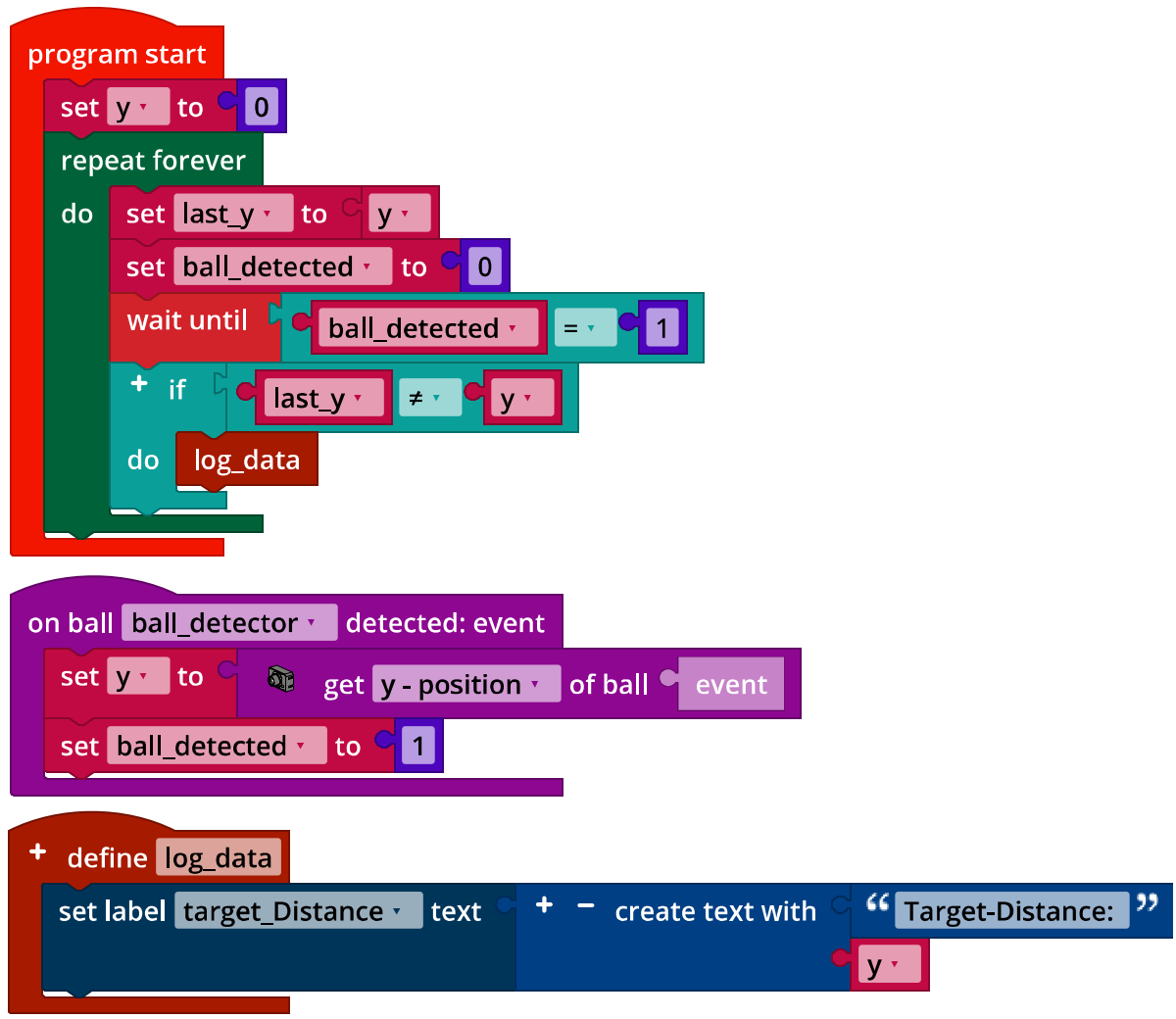
Konfiguration des Erkennungsfensters (Ball-Erkennung):



Je größer das Erkennungsfenster und der Bereich des zu erkennenden Ball-Durch­messers, desto größer die maximale Abweichung vom idealen Abstand, die erkannt werden kann. Als Range für die X-Achse ist das Intervall [-150, 150] zu empfehlen: Es nutzt fast die gesamte Auflösung (320 Pixel) und lässt sich leicht in einen Winkel umrechnen, da der Öffnungswinkel der Kamera 60° beträgt.

Die gewählte Farbe und die Hue-Toleranz müssen an die jeweiligen Lichtverhältnisse angepasst werden.

1b. Programm zur Abstandsmessung (Beispiel):



*Test\_Target\_Distance.ft*

Beispielmessung zum Verhältnis des Abstands der Zielscheibe zum y-Wert des Ballmittelpunkts, den die Ball-Erkennung liefert:

|  |  |
| --- | --- |
| Abstand von  idealer Distanz | y-Koordinate |
| +15 cm | 32 |
| +10 cm | 24 |
| +5 cm | 14 |
| 0 cm | 0 |
| -5 cm | -13 |
| -10 cm | -34 |
| -15 cm | -58 |

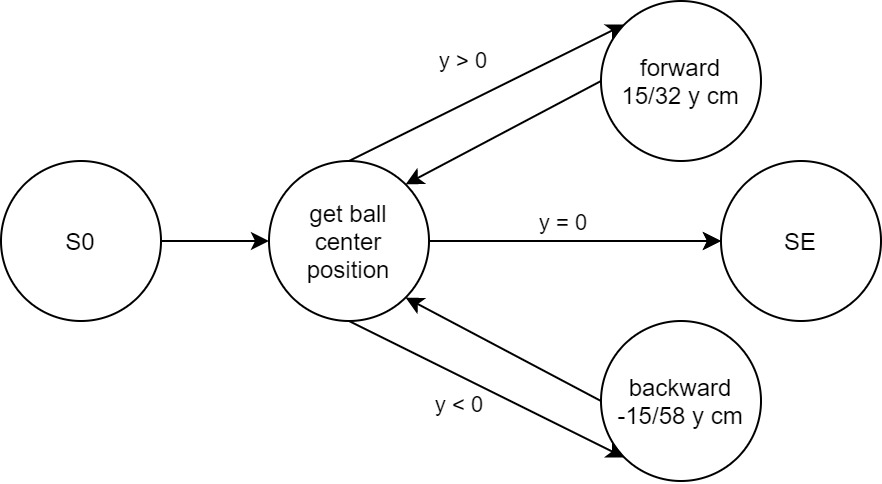
Die von der Ball-Erkennung zurückgelieferten Werte der y-Koordinate des Ball­mittelpunkts hängen von der gewählten Skalierung und der Höhe des Erken­nungsfensters ab. Bis auf einen Skalierungsfaktor sollten die Werte jedoch mit den in obiger Tabelle angegebenen übereinstimmen.

1c. Die Distanz , um die das Fahrzeug vom idealen Abstand zur Zielscheibe entfernt ist, lässt sich leicht mit zwei linearen Funktionen annähern:

Auch hier hängt der Steigungsfaktor der Geraden von der Höhe des Erkennungs­fensters und der in der Ball-Erkennung eingestellten Skalierung ab und kann um einen Faktor abweichen.

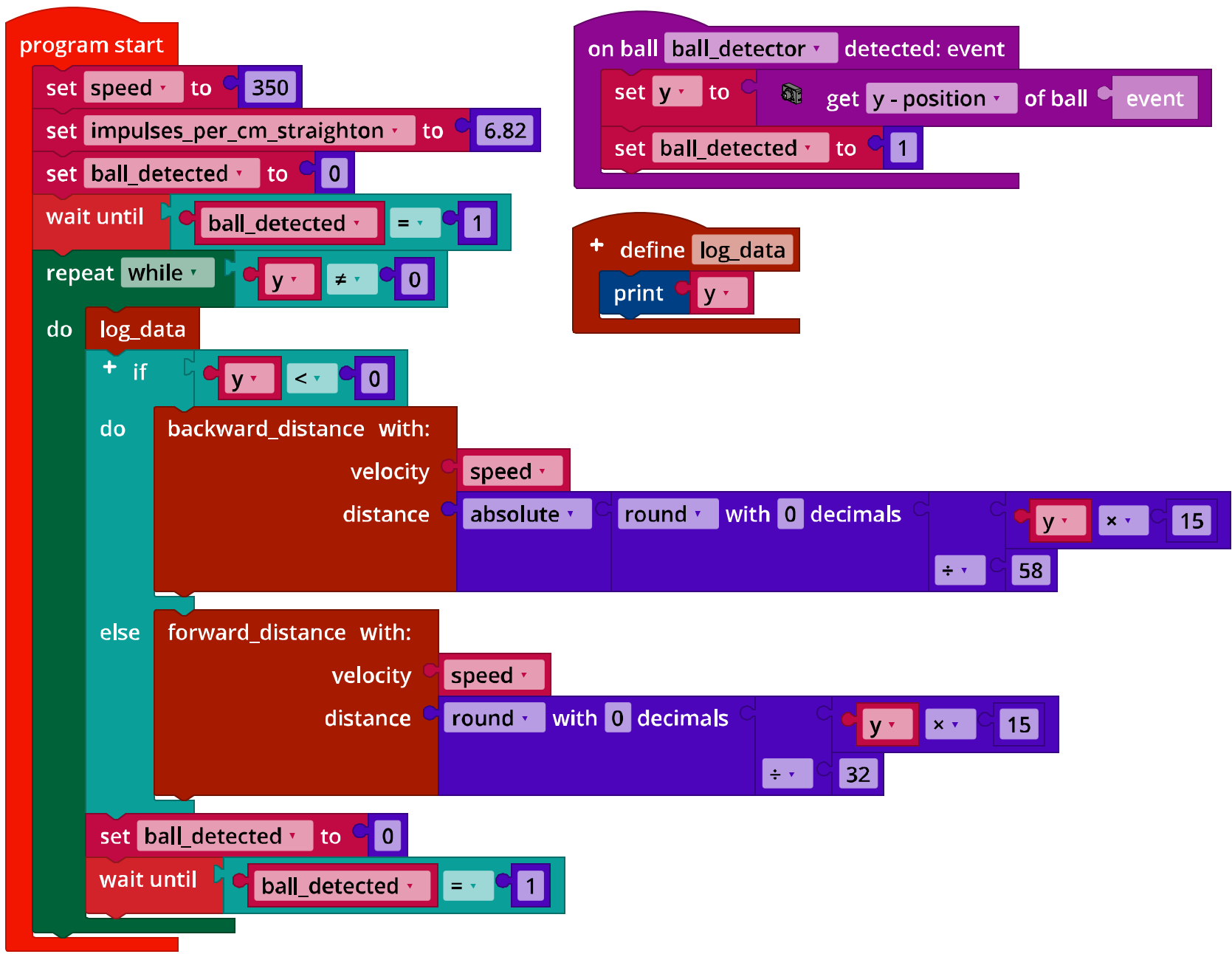
**2. Zielabstandskorrektur**

2a. Zustandsübergangsdiagramm:



*State\_Transition\_Diagram\_Correct\_Position.drawio*

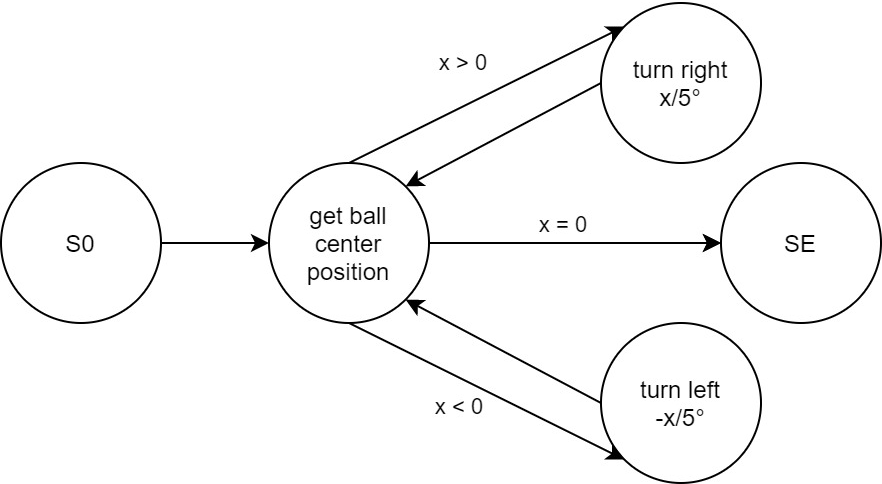
2b. Programmauszug (Beispiel):



*Mecanum\_Correct\_Distance.ft*

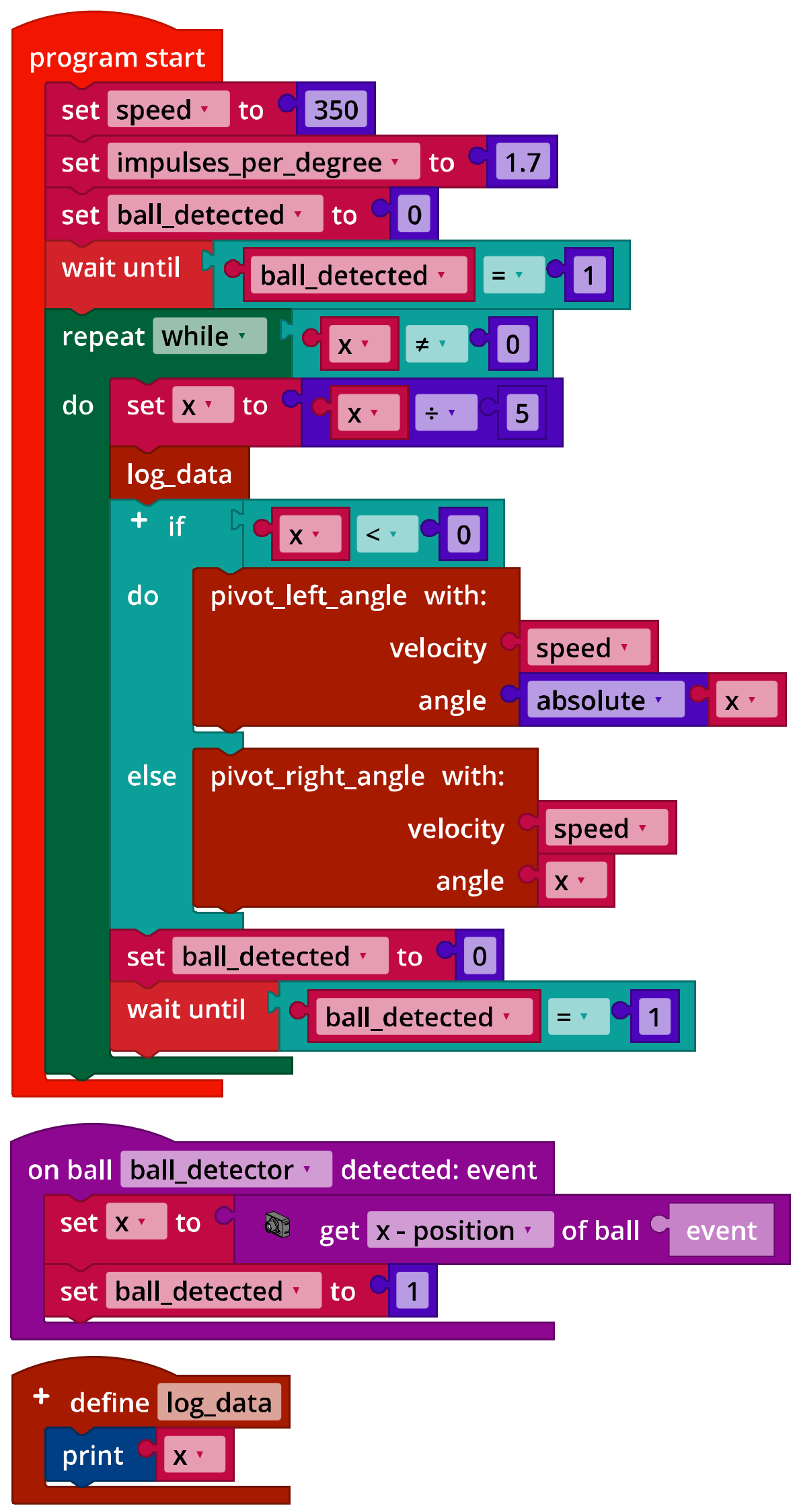
**3. Zielausrichtung**

3a. Zustandsübergangsdiagramm:



*State\_Transition\_Diagram\_Correct\_Position.drawio*

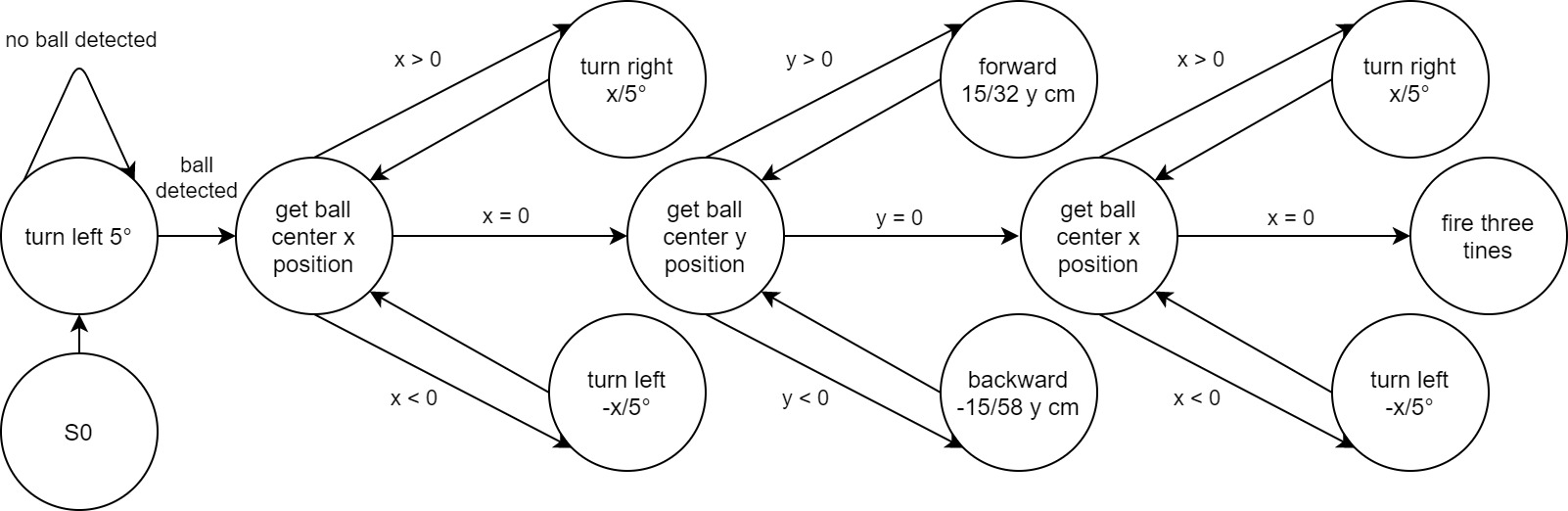
3b. Programmauszug (Beispiel):



*Mecanum\_Turn2Target.ft*

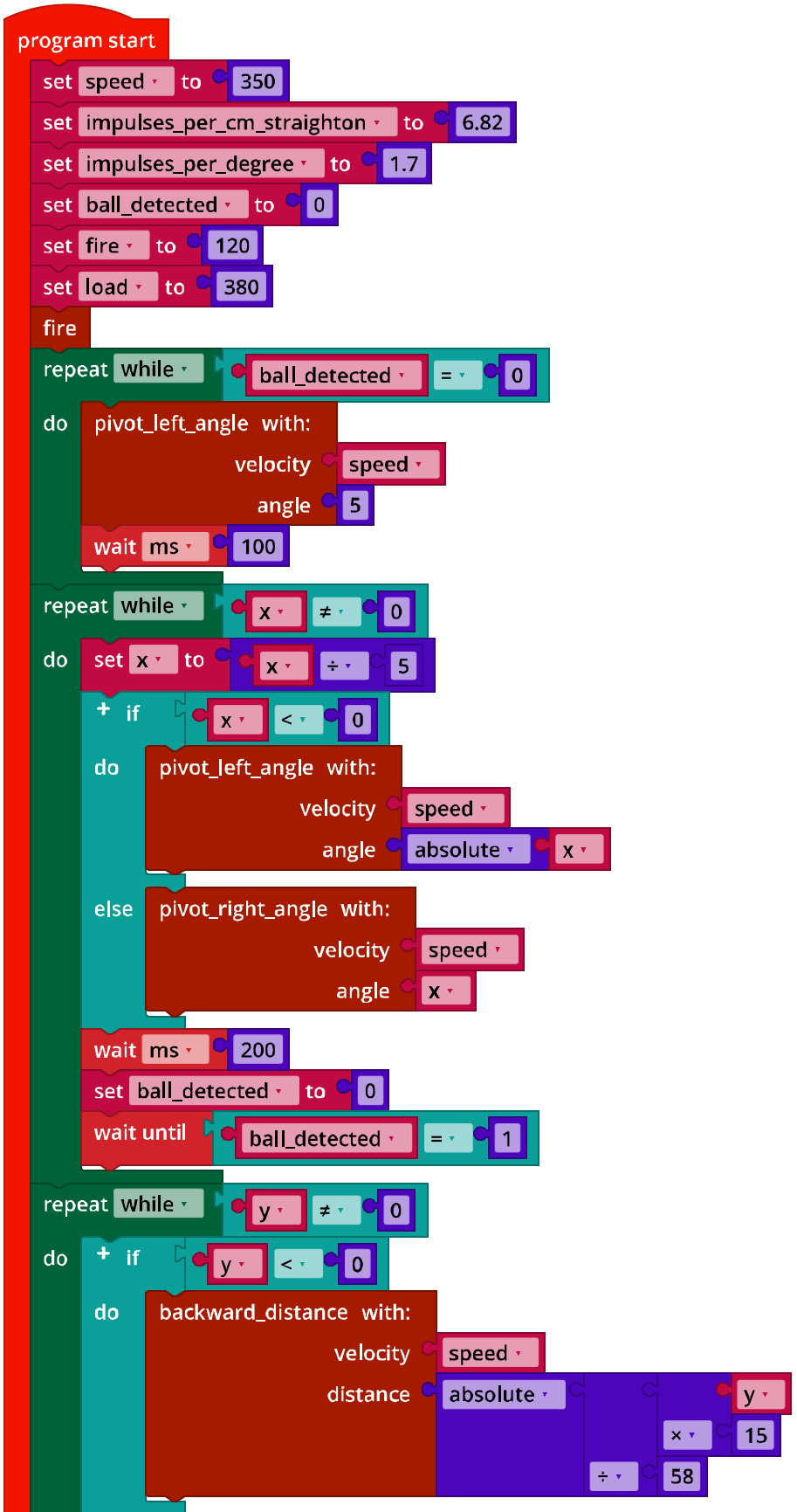
**4. Zielsuche**

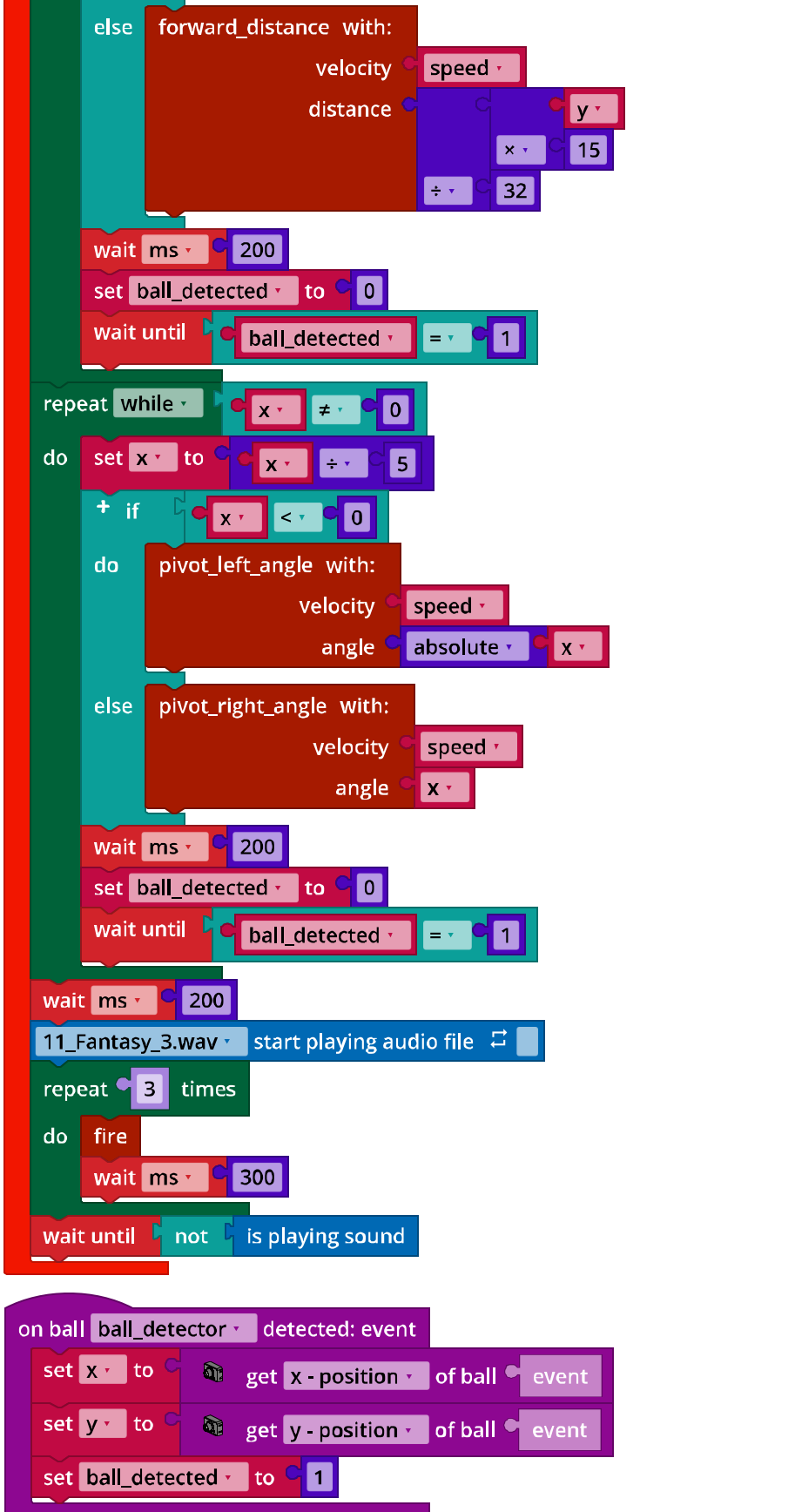
4a. Zustandsübergangsdiagramm:



*State-Transition\_Diagram\_Find\_Target.drawio*

4b. Programmauszug (Beispiel):





*Mecanum\_Find\_Target.ft*

Anlagen

# Aufgabe 5: Ballroboter

## Erforderliches Material

* PC für Programmentwicklung, lokal oder über Web-Schnittstelle.
* USB-Kabel oder BLE- bzw. WLAN-Verbindung für die Übertragung des Programms auf den TXT4.0.
* Styroporkugeln

## Weiterführende Informationen

[1] FRC Team 2605 (Bellingham, WA): [*How a Mecanum Drive Works*](https://seamonsters-2605.github.io/archive/mecanum/). github.io

[2] Wikipedia: [*Endlicher Automat (Zustandsautomat)*](https://de.wikipedia.org/wiki/Endlicher_Automat)

[3] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, Peter Wolstenholme: [*Modeling Software with Finite State Machines. A Practical Approach*](http://is.ifmo.ru/download/modelingsoftwarewithfinitestatemachinesapracticalapproach.pdf). Auerbach Publications, 2006.

[4] Online-Diagrammeditor zur Erstellung von Zustandsübergangsdiagrammen (Format drawio): <https://www.diagrammeditor.de/>