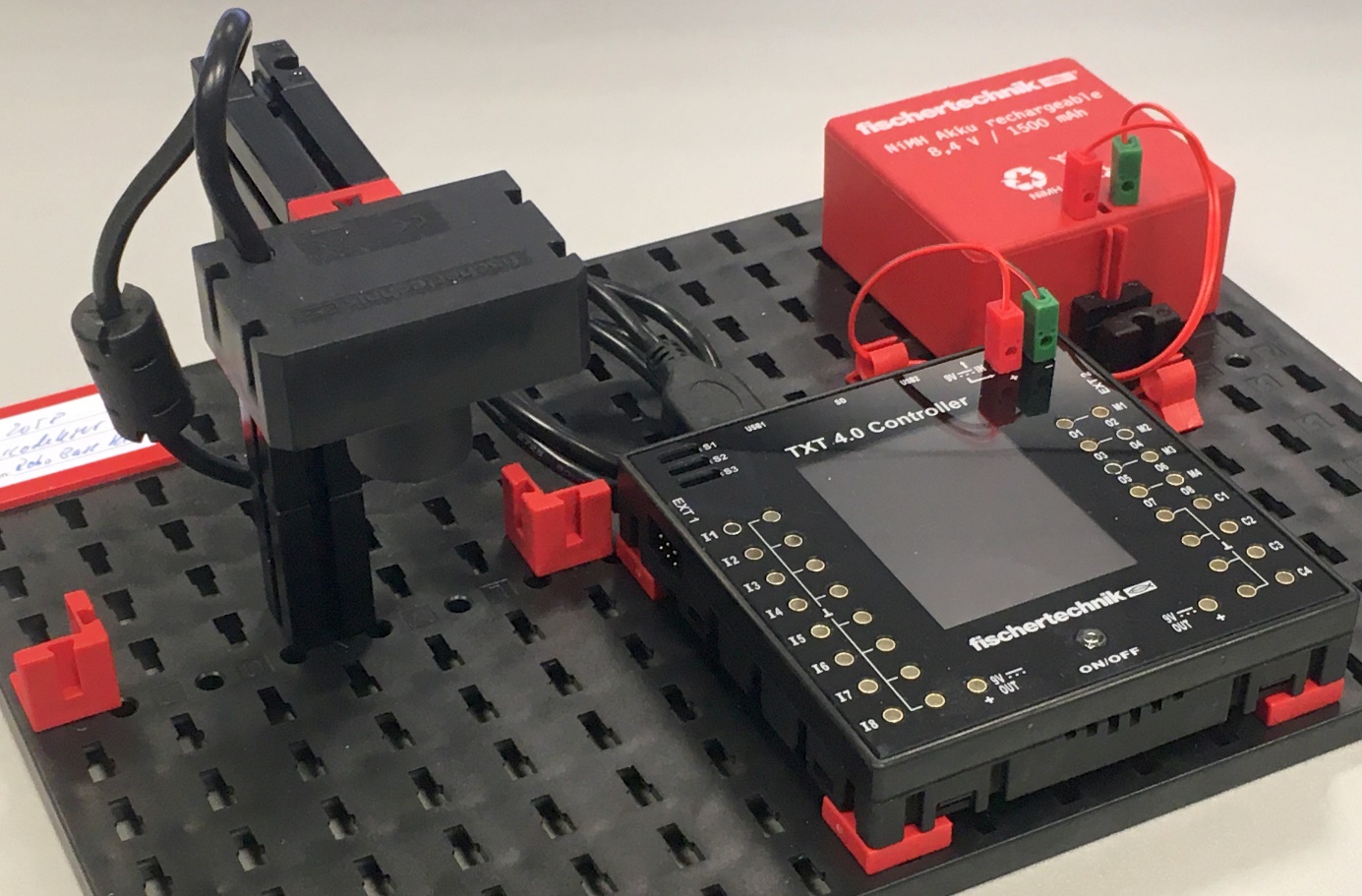
# Aufgabe 4: Kodierung

## Konstruktionsaufgabe

Baue das einfache Lesegerät nach der Bauanleitung. Schließe die USB-Kamera an den TXT an (USB1). Schneide die Code-39-Beispiele und den Farbkreis an den Schnitt­linien aus.



## Programmieraufgaben

**1. Balkenerkennung**

Die Kamera kann Linien (Balken) erkennen. Im Kamera-Fenster musst du dafür zunächst die Linienerkennung konfigurieren und die Kamera durch Drehen des Objektivs scharf stellen.

1a. Schreibe ein Programm, das einen schwarzen senkrechten Balken erkennt und auf dem Display des TXT anzeigt, wie breit er ist.

1b. Erweitere das Programm so, dass es zwei nebeneinander liegende senkrechte schwarze Balken erkennt und bestimmt, wie breit der dazwischen liegende (weiße) Zwischenraum ist. Gib die Breite des Zwischenraums auf dem Display des TXT aus.

**2. 9-stelliger Barcode**

Die Kamera kann bis zu fünf Balken gleichzeitig erkennen. Nimmt man die vier weißen Zwischenräume dazu, kann so ein neunstelliger Barcode dekodiert werden.

Schreibe ein Programm, das ein mit einem neunstelligen binären Barcode, also mit zwei verschiedenen Balkenbreiten für „0“ und „1“ kodiertes Zeichen erkennt und den (numerischen) Wert des Binärcodes auf dem Display des TXT ausgibt.

Verwende zum Testen die beiliegenden Code-39-Barcodes.

**3. Code-39-Dekodierer**

Erweitere dein Programm so, dass es die beiliegenden Code-39-Barcodes dekodiert und das jeweils erkannte Zeichen auf dem Display des TXT ausgibt. Verwende dafür das Programm-Fragment „*Code39\_Template.ft*“, das zwei Listen enthält: alle Ziffern und die Buchstaben des Alphabets sowie deren Code-39-Kodierung.

ABC

*Code-39-Darstellung von „ABC“ (ohne Start-/Stopp-Zeichen)*

**4. Fehlererkennung**

Beim Code-39 sind nicht alle mit neun Balken darstellbaren Codes gültig.

4a. Wie viele verschiedene Zeichen können mit dem Code-39 kodiert werden? Bestim­me die Redundanz des Codes, also die Informationseinheiten (Bits), die ohne Informa­tionsverlust weggelassen werden können.

4b. Die große Redundanz des Code-39 ermöglicht die Erkennung von Fehlern. Wie kannst du am einfachsten ungültige Codeworte erkennen?

4c. Erweitere dein Programm zur Dekodierung eines Code-39 so, dass bei einem fehlerhaft kodierten Zeichen der erkannte Fehler auf dem Display angezeigt wird.

## Experimentieraufgaben

**1. RGB-Code**

Auch die Darstellung von Farben im Computer erfolgt durch eine Kodierung. Der verbreitete RGB-Code beispielsweise gibt den Anteil der Farben Rot, Grün und Blau in einem Farbton an. In Blockly kannst du dir von der Kamera den RGB-Code einer erkannten Fläche (als Liste aus drei Werten) zurückgeben lassen.

1a. Schreibe ein Blockly-Programm, das dir den RGB-Code einer (farbigen) Fläche unter der Kamera auf dem Display des TXT ausgibt.

Aus dem RGB-Code kann man den Farbton nicht direkt ableiten. Jedoch lässt sich der RGB-Code in eine Position im Farbkreises umrechnen. Die dafür erforderliche Berechnungsvorschrift, die dir den passenden Farbton-Wert (eine Zahl zwischen 0 und 1) liefert, findest du im Begleitmaterial. Wenn du den resultierenden Wert mit 360° multiplizierst, erhältst du den Winkel im Farbkreis.

1b. Erweitere dein Programm so, dass es dir zu einer Farbfläche, die du unter die Kamera legst, den zugehörigen Farbton-Wert aus dem RGB-Code berechnet. Lass‘ dir den RGB-Code, den Farbton und den daraus berechneten Farbkreis-Winkel auf dem TXT Display anzeigen.

Teste das Programm mit dem beiliegenden Farbkreis.

**2. Farberkennung**

Bestimme die Grenzen der acht Farb­bereiche des beiliegenden Farbsterns (als Grad-Werte). Erweitere dein Programm so, dass die jeweils erkannte Farbe auf dem Display des TXT ausgegeben wird.

**3. Mehrdimensionaler Code**

Beim Code-39 wird nur die Breite der Balken ausgewertet; die Farbwechsel (weiß/schwarz) dienen lediglich zur Unterscheidung nebeneinander liegender Balken. Jeder Balken enthält also nur 1 bit Information. Der Code zählt daher zu den „eindimensionalen“ oder „1D“-Codes.

3a. Überlege dir einen mehrdimensionalen Code, der sechs Farben und sechs Balkenbreiten unterscheidet, sodass du mit nur einem Balken 36 Zeichen (26 Buchstaben und 10 Ziffern) kodieren kannst. Notiere den Code in einer Tabelle.

3b. Programmiere einen Dekodierer für diesen Code. Das dekodierte Zeichen soll auf dem Display des TXT angezeigt werden. Teste das Programm mit selbst gezeichneten farbigen Balken.

Anlagen

# Aufgabe 4: Kodierung

## 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.
* Programm-Template (Code 39): Code39\_Template.ft
* Code-39-Beispiele zum Dekodieren (Breite: 6 cm)
* Farbkreis und Farbstern
* Papier und Farbstifte zum Design eines eigenen Codes

## Weiterführende Informationen

[1] Dominic Welsh: *Codes und Kryptographie*. VCH 1991.

[2] Wikipedia: [*Strichcode*](https://de.wikipedia.org/wiki/Strichcode), [*Code 39*](https://de.wikipedia.org/wiki/Code_39).

[3] RapidTables:[*RGB Color Codes Chart*](https://www.rapidtables.com/web/color/RGB_Color.html)*.*