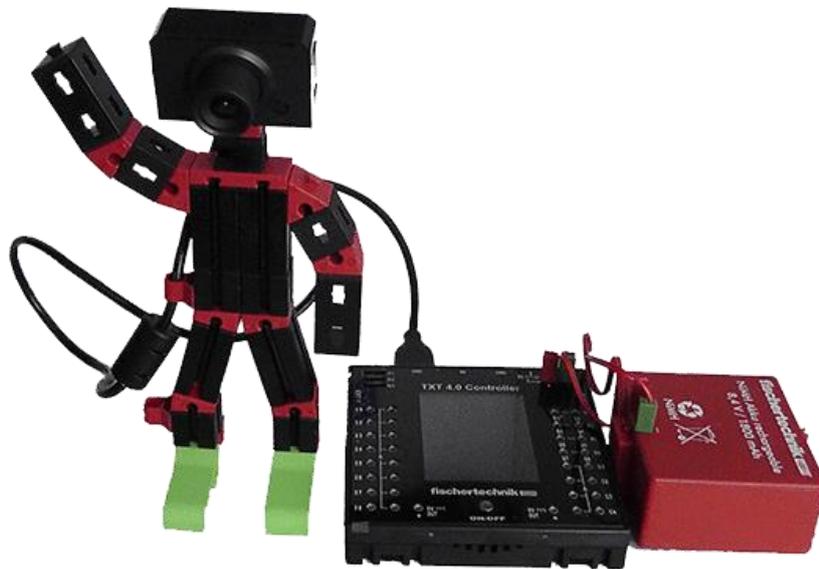


## Kameramann



Mit diesem Modell erarbeitest du dir die ersten Grundlagen der Programmierung mit der Kamera.



Wenn du das Modell anhand der Bauanleitung aufgebaut hast, wird dir sicher schon ein ganz neues Bauteil auffallen, nämlich - die Kamera.

Die Kamera ist ein besonders vielseitiger Sensor. Die Bildauflösung beträgt 1 Megapixel. Die Bilder, die die Kamera aufnimmt, können auf einen PC übertragen und am Bildschirm angezeigt werden. Außerdem kann der TXT 4.0 Controller die Bilder verarbeiten und dadurch Bewegungen, Farben und Spuren erkennen, so dass man Robotermodelle steuern kann. Es besteht aber auch die Möglichkeit, die Daten, die die Kamera liefert, mit der Software ROBO Pro Coding zu verarbeiten. Durch das Drehen des Kameraobjektivs kann die Aufnahme scharf gestellt werden.

Aufgabe 1	Programmierlevel 2 Bewegung im Kamerabild soll erkannt werden. Ist dies der Fall, wird ein akustisches und ein optisches Signal am TXT 4.0 Controller ausgegeben.	<a href="#">Hyperlink zur Aufgabe 1</a>
Aufgabe 2	Programmierlevel 2 Es sollen 3 verschiedene Farben (rot, gelb, grün) erkannt werden (Farbtafeln). Die Farbe, die erkannt wurde, soll im Display der TXT 4.0 Controllers angezeigt werden.	<a href="#">Hyperlink zur Aufgabe 2</a>

## Aufgabe1

### Starten von ROBO Pro Coding



Starte ROBO Pro Coding und wähle eine Vorlage für dein neues Projekt aus.



Da du die Kamera sowie die Anzeige benötigst, aktivierst du die Auswahl:



Kamera + Anzeige

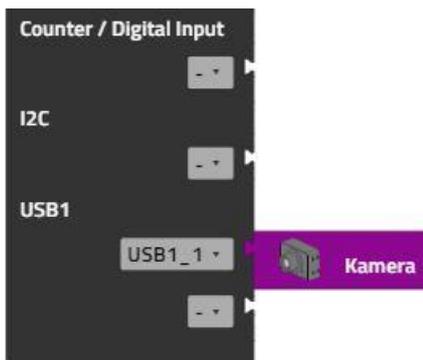
und bestätigst mit :



Du musst noch einen Projektnamen vergeben, z. B.: **cameraman\_alarm**

### Controllerkonfiguration

Auch hier musst du wie beim Funktionsmodell die angeschlossenen Aktoren und Sensoren definieren. Wähle das Fenster „**Controllerkonfiguration**“ aus und ziehe das Controllersymbol aus dem Block „**Controller**“ in den Arbeitsbereich.

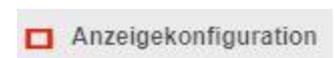
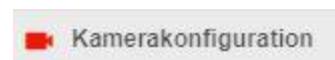


Am Controller muss aus der Gruppe „**USB**“ nur das Element „**Kamera**“ eingebunden werden.



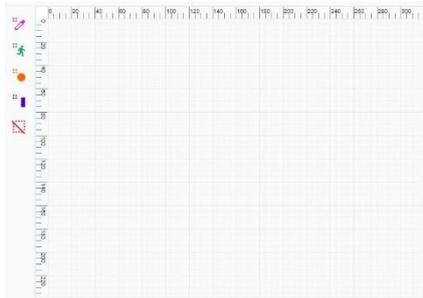
Somit ist der Controller konfiguriert.

Im Projektfenster findest du noch zwei weitere Einträge:



Auch hier musst du verschiedene Einträge bzw. Anpassungen vornehmen.

## Kamerakonfiguration



Klicke zuerst auf die Kamerakonfiguration.

Es erscheint nebenstehendes Fenster.

Rechts neben dem Fenster werden 5 Detektoren dargestellt. Diese sind:



Color- oder Farbdetektor



Motion- oder Bewegungsdetektor



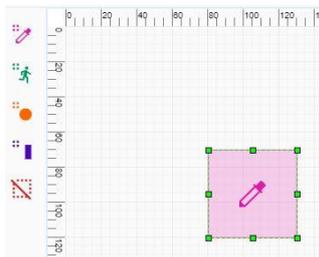
Balldetektor



Line- oder Liniendetektor



Blocked- oder Gesperrter Bereich

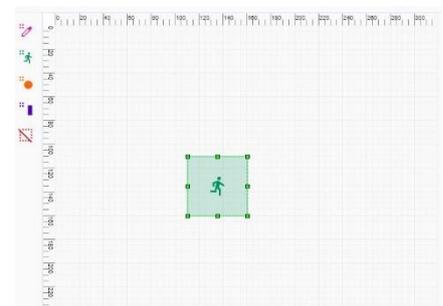


Benötigst du einen dieser Detektoren ziehst du diesen mit der Maus in das Koordinatenfenster (siehe Bild). In der rechten Ecke über dem Code-Fenster findest du den Eintrag „Inspektor“. Aktiviere den Eintrag und es erscheint ein Fenster, in dem du den Detektor noch genauer definieren kannst.

Hier lassen sich Größe, Lage, Name sowie Kontrast verändern. Möchtest du das Fenster löschen, musst du es markieren (grüne Rahmenpunkte), dann die rechte Maustaste und im Auswahlfenster „Löschen“ drücken.

Größe	
Breite :	<input type="text" value="50"/> <input type="button" value="-"/> <input type="button" value="+"/> px
Höhe :	<input type="text" value="50"/> <input type="button" value="-"/> <input type="button" value="+"/> px
Position	
x :	<input type="text" value="80"/> <input type="button" value="-"/> <input type="button" value="+"/> px
y :	<input type="text" value="70"/> <input type="button" value="-"/> <input type="button" value="+"/> px
Identität	
Name :	<input type="text" value="color_detector"/>
Farbe	
Kontrast :	<input type="text" value="1"/> <input type="button" value="-"/> <input type="button" value="+"/>

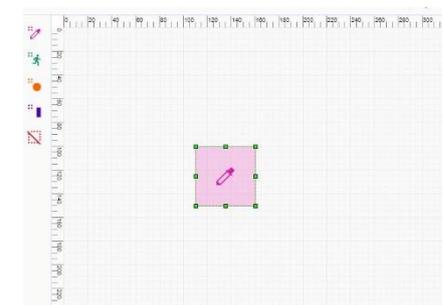
Für die erste Aufgabe benötigst du ein Fenster, in dem eine Bewegung erkannt werden soll. Dazu ziehst du „Bewegungsdetektor“ in das Koordinatenfenster.



Wird später in diesem Fenster eine Bewegung erkannt, kann durch weitere Befehle eine Aktion ausgelöst werden und z.B. ein Warnton ausgegeben werden.

In der zweiten Aufgabe soll eine Farbe erkannt werden. Dafür musst du einen Farbdetektor verwenden.

Bei beiden Aufgaben, wird das Display der TXT 4.0 Controllers benötigt.

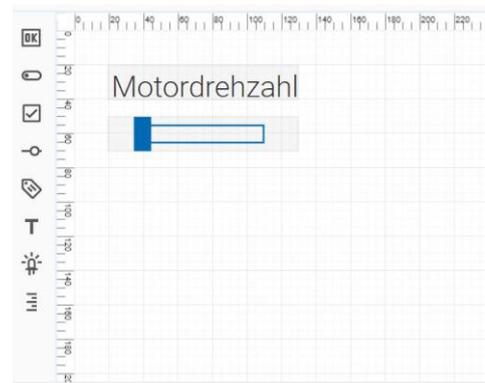


Um es zu konfigurieren, aktivierst du die Schaltfläche

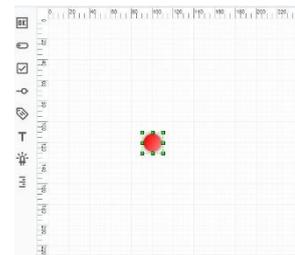
 Anzeigeconfiguration

Hier findest du auf der linken Seite verschiedene Elemente zur Gestaltung der Displayansicht. Diese sind:

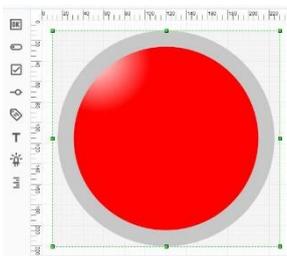
-  TXTButton - Bestätigungsknopf
-  TXTSwitch - Schiebeschalter
-  TXTCheckBox - Kontrollfeld
-  TXTSlider - Schieberegler
-  TXTLabel - Text
-  TXTInput - Texteingabe
-  Statusindikator- Statusanzeige
-  Gauge - Messgeräteanzeige



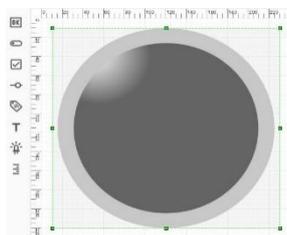
Du ziehst aus der Auswahl die Statusanzeige in den Bildschirm.



Mit den grünen Haltepunkten kannst du die Statusanzeige formatfüllend vergrößern.



In der linken oberen Ecke des Bildschirmes findest du den Eintrag „Inspektor“. Wenn du diesen aufrufst, kannst du auch hier weitere Einstellungen vornehmen.



Im Bereich „Statusindikator“ musst du von „Aktiv“ auf „nicht Aktiv“ umschalten indem du den Haken entfernst.



**Wichtig:** Du musst die Kamera nicht extra über einen Befehl einschalten. Wenn sie definiert wurde, ist sie automatisch aktiviert.

Jetzt kannst du mit dem Programmieren beginnen.



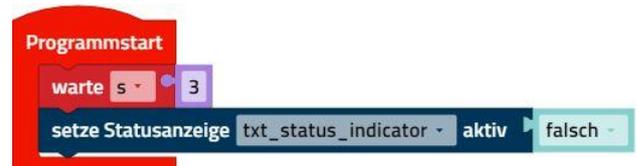
In den Programmstart fügst du zuerst einen Wartebefehl aus dem Block „Util“ mit einer Zeitvorgabe von „3“ Sekunden ein. Die Wartezeit dient nur dazu, nach der Übertragung zum Modell bis zur Ausführung der nächsten Befehle eine kleine Zeitspanne zu warten.



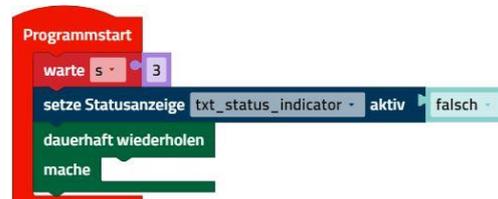
Es folgt ein Befehl aus der Gruppe „Anzeige“



Wurde eine Statusanzeige in der Anzeigekonfiguration erzeugt, erscheint sie in dem Befehl „setze Statusanzeige“ und kann dort ausgewählt werden. Ändere nur die Logikausgabe von „wahr“ auf „falsch“ um. Damit wird die Anzeige auf dem Display ausgeschaltet.



Damit das Programm dauerhaft ausgeführt wird, erweiterst du es mit dem Befehl „dauerhaft wiederholen – mache“ aus der Gruppe „Schleifen“



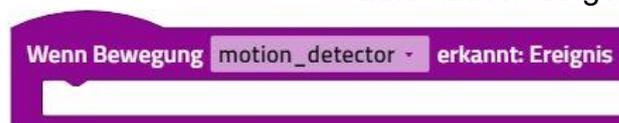
Somit ist der erste Programmabschnitt fertig und du kannst dich an das eigentliche Abfrage- und Auswerteprogramm wagen.

Abgefragt wird laut Kameradefinition der „Bewegungsdetektor – oder auch „motion\_detector“.



Hierfür findest du in der Gruppe „USB“ den Befehl „Wenn Bewegung --- erkannt“. Diesen Befehl ziehst du unter den Programmstart.

Auch hier wird der Eintrag „motion\_detector“ automatisch eingetragen.



Was muss geschehen, wenn die Kamera eine Bewegung erkannt hat?

Im Display soll die Statusanzeige aktiviert werden. Dies wird ermöglicht durch Setzen der Statusanzeige auf „wahr“. Baue diesen Befehl ein.

```

Anzeige
setze Statusanzeige - aktiv - wahr -
    
```

Im Programm sieht das dann wie folgt aus:

```

Wenn Bewegung motion_detector - erkannt: Ereignis
setze Statusanzeige txt_status_indicator - aktiv - wahr -
    
```

**Sound**

stoppe Tonwiedergabe

Stoppe als Nächstes die Tonwiedergabe. Den Befehl „stoppe Tonwiedergabe“ findest du in der Gruppe „Sound“. Füge den Befehl ins Programm ein.

```

Wenn Bewegung motion_detector - erkannt: Ereignis
setze Statusanzeige txt_status_indicator - aktiv - wahr -
stoppe Tonwiedergabe
    
```

**Sound**

01\_Airplane.wav - spiele Datei

Schalte danach einen Ton ein. Dazu fügst du den Befehl „xxx spiele Datei“ aus der Gruppe „Sound“ ein. Wenn du den Pfeil betätigst, erscheint eine Liste verschiedener Töne. Wähle einfach einen passenden aus.

```

Wenn Bewegung motion_detector - erkannt: Ereignis
setze Statusanzeige txt_status_indicator - aktiv - wahr -
stoppe Tonwiedergabe
02_Alarm.wav - spiele Datei
    
```

**Util**

warte bis

gibt Ton wieder

Die Tondauer kannst du über den Befehl „warte bis“ aus der Gruppe „Util“ und dem Befehl „gibt Ton wieder“ aus der Gruppe „Sound“ definieren.

```

Wenn Bewegung motion_detector - erkannt: Ereignis
setze Statusanzeige txt_status_indicator - aktiv - wahr -
stoppe Tonwiedergabe
02_Alarm.wav - spiele Datei
warte bis - gibt Ton wieder
    
```

Jetzt fehlt nur noch der Befehl „setze Statusanzeige – falsch“ der die Anzeige wieder ausschaltet. Kopiere ihn einfach mit Rechtsklick auf den schon definierten Befehl und schiebe ihn in die letzte Programmzeile. Ändere hier den Wert „wahr“ auf „falsch“ um.

```

Wenn Bewegung motion_detector - erkannt: Ereignis
setze Statusanzeige txt_status_indicator - aktiv - wahr -
stoppe Tonwiedergabe
02_Alarm.wav - spiele Datei
warte bis - gibt Ton wieder
setze Statusanzeige txt_status_indicator - aktiv - falsch -
    
```

Somit ist das Programm fertig und du kannst es testen.



Dazu klickst du mit der Maus in der rechten oberen blauen Leiste auf den Befehl „Programm starten“.



Das Programm wird zu deinem TXT 4.0 Controller übertragen und gestartet.

Wenn du auf das Display schaust, wird dir die nicht aktive Statusanzeige angezeigt.



Erkennt die Kamera eine Bewegung, schaltet die Anzeige auf den Status „wahr“ um und ein Ton wird über den Lautsprecher am TXT 4.0 Controller ausgegeben.



Das Programm läuft in einer Endlosschleife. Möchtest du es anhalten, klickst du in der blauen Leiste auf den Befehl „Stopp“.



## Aufgabe 2

Bei der zweiten Aufgabe sollen verschiedene Farben erkannt und auf dem Display angezeigt werden. Als Farbflächen kannst du die farbigen Bauplatten aus dem Baukasten verwenden.

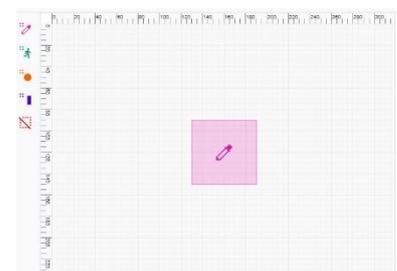
Die Controllerkonfiguration ist die Gleiche wie bei der ersten Aufgabe.

In der Kamera- wie auch in der Anzeigekonfiguration musst du ein paar Änderungen vornehmen.

Rufe zuerst die Kamerakonfiguration auf. Ersetze den Bewegungsdetektor mit dem Farbdetektor.

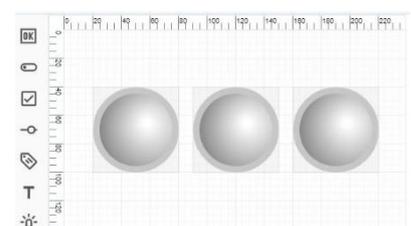
Im Inspektor kannst du noch zwei Einstellungen vornehmen. Ändere die Größe auf „60/60“

und die Position auf „130/90“. Du kannst aber auch eigene Werte verwenden und etwas experimentieren.



Schalte anschließend auf die Anzeigekonfiguration um.

Anstelle einer Anzeige brauchst du für diese Aufgabe 3 Anzeigen. Lösche die eine und füge 3 Statusindikatoren ein wie im Bild dargestellt.



Die jeweilige Größe kannst du im Inspektor einstellen:  
Größe: 60/60 und die jeweilige Position: 20/40, 90/40 und

160/40. Schalte den Statusindikator auf „nicht aktiv“.

Verwende für die jeweilige Farbe einen eindeutigen Namen

Identität: Für Grün: txt\_status\_indikator\_green  
Für Gelb: txt\_status\_indikator\_yellow  
Für Rot: txt\_status\_indikator\_red

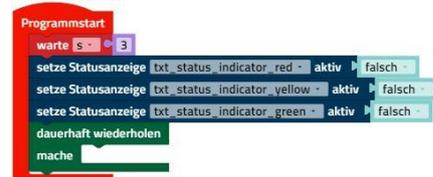
Ändere auch noch die jeweilige Farbe für die Darstellung der Statusanzeigen ab. Dazu klickst du auf die Farbfläche und wählst die entsprechende Farbe aus, entweder mit dem Slider oder du verwendest folgende Werte



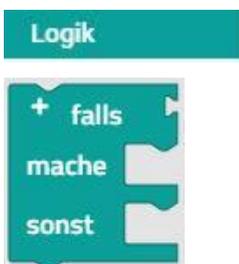
Grün: #61996a, Gelb: #b88124, Rot: #85131d

Die Eingabe bzw. Farbänderung schließt du mit „Save“ ab.

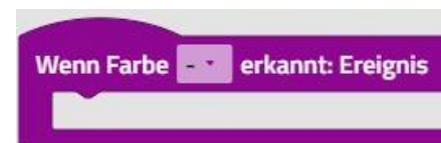
Den ersten Teil der Programmierung kannst du aus der Aufgabe 1 übernehmen. Füge nur zwei weitere „setze Statusanzeige“ ein und ändere „aktiv“ auf „falsch“.



Jetzt kannst du das Programm für die Ermittlung der Farbe aufbauen. Dazu benötigst du aus dem Block „USB“ den Befehl „Wenn Farbe --- erkannt“.

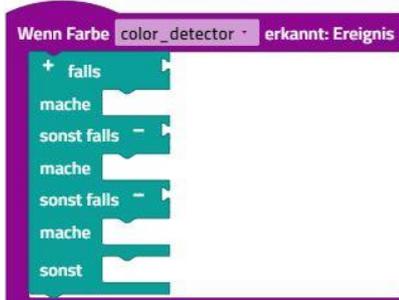


Dann baust du einen dir schon bekannten Befehl „falls --- mache – sonst“ aus der Gruppe „Logik“ ein.



Da du drei Farben abfragen musst, benötigst du ihn dreimal.

Klicke dazu zweimal auf das + Zeichen.



Die Abfrage wird wie im Bild dargestellt erweitert.

In die Andockstellen fügst du die Abfrage nach den einzelnen Farben ein. Den Befehl „ist Farbe – Ereignis = Farbe“ findest du in der Gruppe „USB“



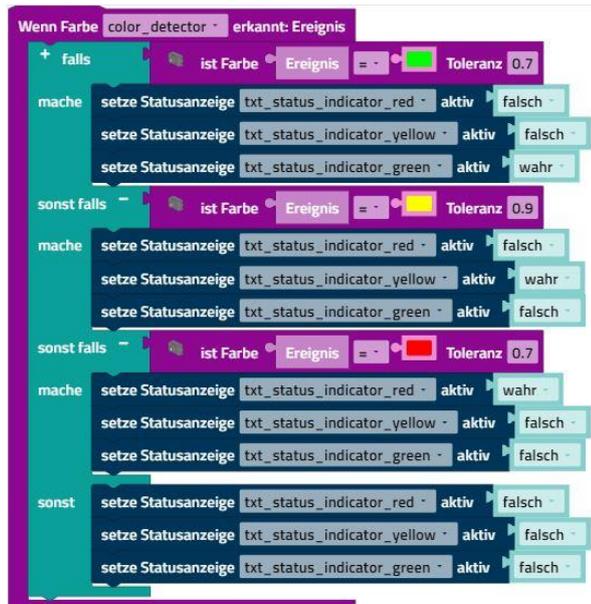
Den Befehl kopierst du an die beiden anderen Andockstellen. Ändere dort die Farbe auf Gelb und Rot um. Die Farbcodes kannst du aus der Anzeigekonfiguration entnehmen.

Jetzt müssen die Statusanzeigen für die einzelnen Farben definiert werden. Dazu benötigst du aus dem Block „Anzeige“ den Befehl „setze Statusanzeige ...“. Füge den Befehl jeweils an die freien Stellen im Programm.

Füge ihn einmal ein und kopiere ihn dann an die weiteren Stellen.

Ändere auch gleich hier die Farbtoleranz auf 0.7 bei grün und rot sowie 0.9 bei gelb. Damit funktioniert die Erkennung sehr zuverlässig.

Die Statusanzeigen musst du für jede erkannte Farbe ein oder ausschalten. Dazu werden in jeder Abfrage nochmals zwei „setze Statusanzeigebefehle“ benötigt. Kopiere diese einfach an die entsprechenden Stellen.

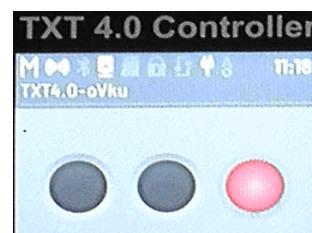


Ändere zuerst die Anzeige um (Pfeilchen und Auswahl) und dann die „aktiv-Einstellung“ für die jeweilige Farbe.

Grün: falsch, falsch, wahr  
Gelb: falsch, wahr, falsch  
Rot: wahr, falsch, falsch

Sonst: falsch, falsch, falsch

Somit ist das Programm fertig und du kannst es starten. Im Display des TXT 4.0 Contollers werden entsprechend der erkannten Farben die dazugehörige Anzeige aktiviert.



## Erweiterung des Programms mit der „Hexadezimalen Anzeige“ und der „RGB Anzeige“ der ermittelten Farbe.

Das System der **hexadezimalen Farbdefinition** findest du in vielen Bereichen des computergestützten Designs. Bei diesem System wird eine Farbe durch drei aufeinander folgende **Hexadezimalzahlen** dargestellt, die jeweils für eine Farbe des **RGB-Farbraums** stehen.

RGB steht dabei als Abkürzung für die Anteile der **Basisfarben** Rot, Grün und Blau an der **Mischfarbe**.

Grundfarben

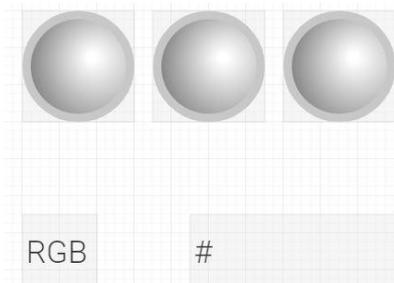
Farbe	Hex-Wert	R	G	B	Komplementär zu
Schwarz	#000000	0	0	0	Weiß
Grau	#888888	136	136	136	#777777
Weiß	#FFFFFF	255	255	255	Schwarz
Rot	#FF0000	255	0	0	Cyan
Dunkelrot	#880000	136	0	0	#77FFFF
Grün	#00FF00	0	255	0	Magenta
Dunkelgrün	#008800	0	136	0	#FF77FF
Blau	#0000FF	0	0	255	Gelb
Dunkelblau	#000088	0	0	136	#FFFF77

Üblich ist die hexadezimale Farbdefinition in sechsstelliger Form, das heißt, als eine Aneinanderreihung von drei jeweils zweistellig geschriebenen Hexadezimalzahlen, nach dem Schema: **#RRGGBB**, die es erlaubt, pro **Objekt** (= 256) verschiedene Zustände zu definieren – von 0 für eine komplett „ausgeschaltete“ Farbe und 255 (FF) für 100 % **Farbsättigung**. Es wird die **additive Farbmischung** zum Mischen der Farben benutzt; der Code **#FFFF00** ergibt die Farbe Gelb, weil Rot mit Grün gemischt wird.

Erweitere in der Anzeigekonfiguration das Fenster mit einem Textlabel und einem Textfenster, in dem etwas angezeigt wird.

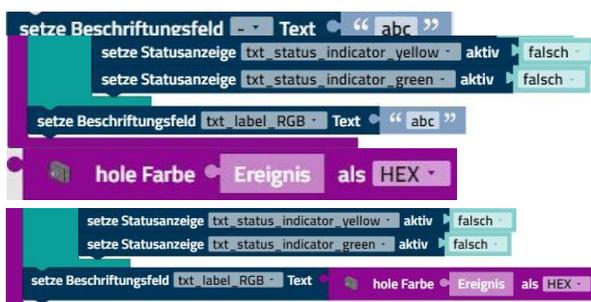


Positioniere beide Elemente unter der Farbdarstellung. Du kannst die folgenden Koordinaten verwenden: Öffne dazu den Inspector.



Größe: 40/40, Position: 20/150, Identität Name: text\_label, TXTLabel Text: RGB  
Größe 110/40, Position: 110/150, Identität Name: txt\_label\_RGB, TXTLabel Text: #

Somit ist die Erweiterung in der Anzeigekonfiguration fertig und du kannst dein Programm mit einer Programmzeile erweitern.



Füge aus dem Block „Anzeige“ den Befehl „setze Beschriftungsfeld --- Text – abc“ nach dem „falls sonst“ Befehl ein.

Anstelle von ABC fängst du aus dem Block „USB“ den Befehl „hole Farbe Ereignis als RGB“ ein. Im Display wird je nach erkannter Farbe der RGB-Wert angezeigt.

Somit ist das Programm fertig und du kannst es mit deinem Modell testen.

Demontiere danach das Modell und widme dich der nächsten Aufgabe.

