ftrobotik

fischertechnik

ROBOTİK LABORATUVARI

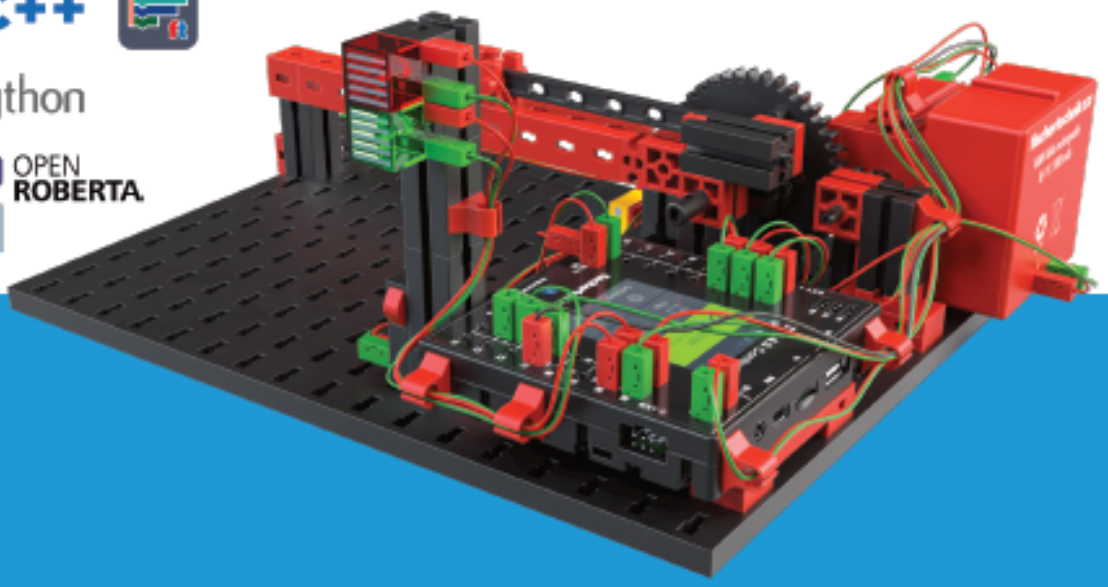
LİSE

Programlama Dilleri:

C, C++

python

OPEN ROBERTA



ETKİNLİK KİTABI 1

Temel Robotik

STEM

MATHEMATIK
INFORMATIK
NATURWISSENSCHAFT
TECHNIK

İçindekiler

GİRİŞ:

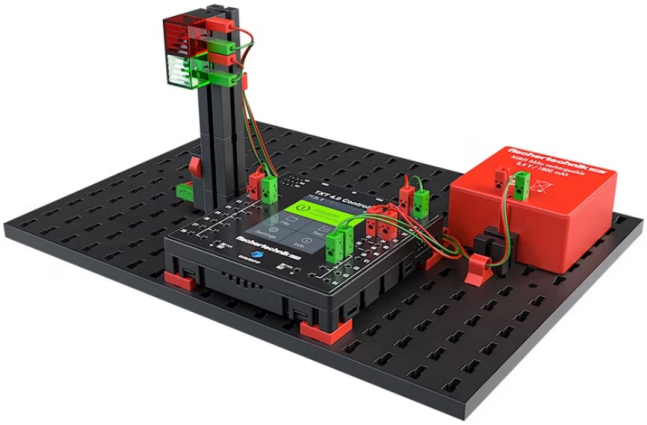
TXT 4.0 Kontrol Ünitesi

Sayfa: 6

Bölüm 1:

Yaya Işıkları

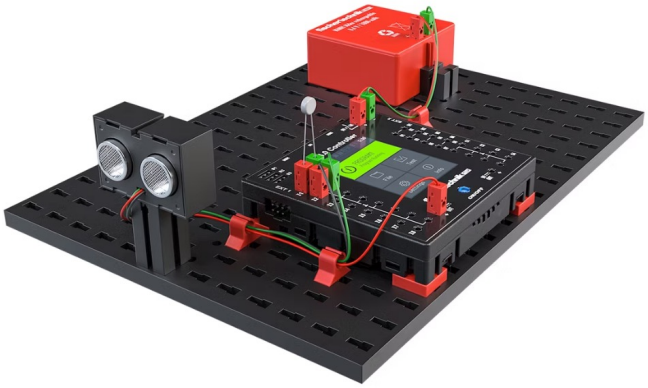
Sayfa: 18



Bölüm 2:

Analog Sensörler

Sayfa: 25



İçindekiler

Bölüm 3:

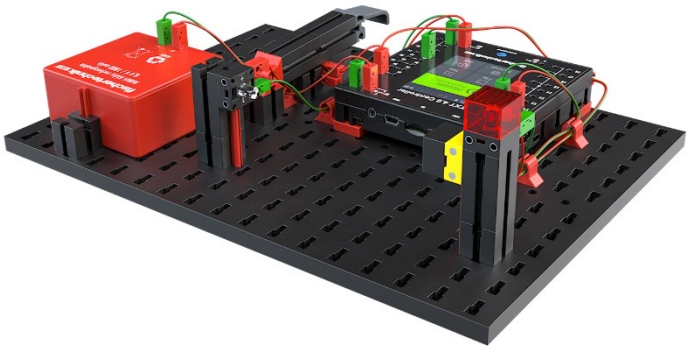
Barkod Okuyucu

Sayfa: **42**

Bölüm 4:

Mors Telgrafı

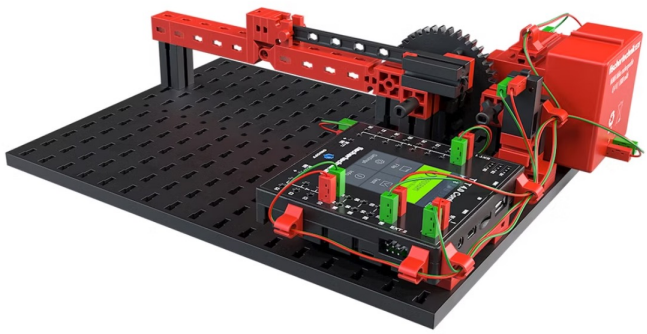
Sayfa: **62**



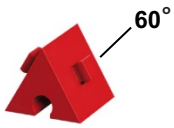
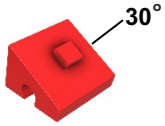
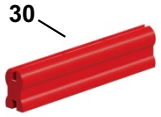
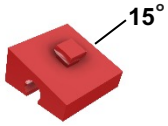
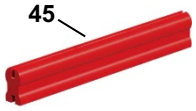
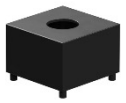
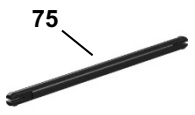
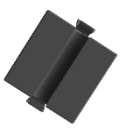
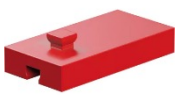
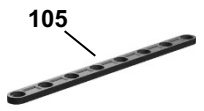
Bölüm 5:

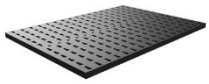
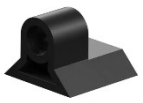
Otopark Bariyer

Sayfa: **83**

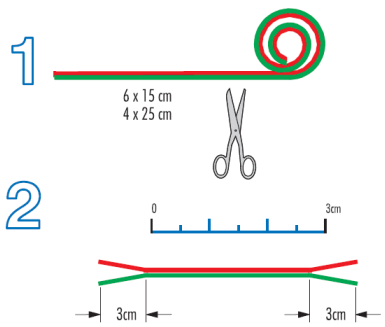


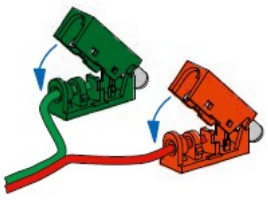
Parça Listesi

	31 010 2x		31 848 1x		31 060 4x		163 438 2x
	31 011 3x		32 064 4x		31 061 7x		35 079 1x
	31 981 5x		32 321 1x		31 330 1x		35 084 1x
	37 636 4x		31 597 2x		35 069 1x		31 124 1x
	37 237 14x		37 679 6x		163 518 1x		36 532 1x
	37 238 7x		31 022 1x		32 085 1x		162 134 2x
	32 879 8x		35 969 13x		31 058 2x		162 135 1x
	32 881 5x		35 945 2x		31 021 2x		35 087 1x
	32 882 1x		31 436 2x		35 031 2x		35 065 1x
	35 049 4x		31 426 2x		37 157 1x		176 775 1x
	32 330 2x		38 246 2x		35 050 2x		38 240 2x
	38 428 3x		38 241 2x		38 423 2x		35 073 2x

	163 206 2x		163 205 2x		32 880 1x		153 422 2x
	31 982 10x		35 033 2x		32 985 1x		137 125 2x
	185 360 2x		36 573 1x		36 134 1x		152 522 1x
	31 690 1x		36 950 4x		37 783 2x		172 804 1x
	37 468 7x		78 728 2x		127 421 1x		172 805 1x
	185 789 1x		35 608 1x		31 360 1x		36 437 1x
	133 009 1x		128 598 1x		181 583 21x		173 067 1x
	35 537 1x		522 460 1x		181 584 28x		134 867 1x

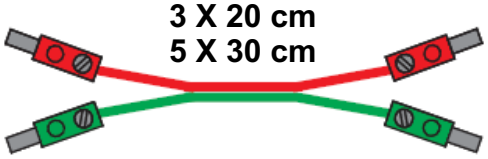
Kablo Bağlantıları





Hazır kablolar

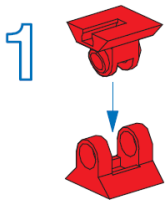
3 X 20 cm
5 X 30 cm



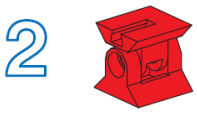
1 X 30 cm

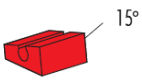
Kurulum Hazırlık

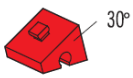
2x Teker

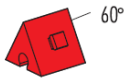


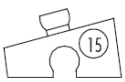
2x

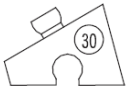


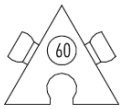












Açılı Bloklar

1x

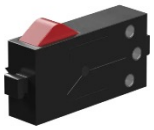
INPUT - Sensörler

Ultrasonik Mesafe Sensörü

IR İz Sensörü

USB Kamera

NTC Isı Sensörü



Touch (anantar) Sensör

Foto-transistör Sensör

OUTPUT – Motor / Lamba

Enkoder Motor

Lamba

Lens Uçlu Lamba

ftrobotik

fischertechnik

GİRİŞ

TXT 4.0 Kontrol Ünitesi

Öğrenme hedefi

- ✓ Mikrokontrolcüleri tanıyalım
- ✓ TXT 4.0 Kontrol Ünitesi
- ✓ Robo Pro Coding'i bilgisayarımıza kuralım
- ✓ TXT 4.0 Kontrol Ünitesini Robo Pro Coding programına bağlayalım

Robot Kontrol Ünitelerini Anlamak

Konuya Giriş

Bilgisayarlar günümüz dünyasının ayrılmaz bir parçası haline gelmiştir. Artık yalnızca masaüstü bilgisayarlar, dizüstü bilgisayarlar veya tabletler şeklinde değil; cebimizde taşıdığımız akıllı telefonlarda, araçlarda, bulaşık ve çamaşır makinelerinde, ısıtma sistemlerinde, elektrikli bisikletlerde ve kahve makinelerinde de yer almaktadırlar.

Her yıl daha güçlü, daha küçük ve daha kompakt hale gelen bilgisayarlar, bu sayede birçok alanda yaygın olarak kullanılabilmektedir. Gömülü sistemler olarak adlandırılan bu yapılar, belirli uygulamalara entegre edilerek özel görevleri yerine getirirler. Sensörler aracılığıyla çevrelerini algılayabilir, motor, LED veya diğer aktüatörleri kontrol edebilirler. WLAN veya Bluetooth gibi kablosuz iletişim teknolojileri sayesinde birbirleriyle haberleşebilir, veri paylaşımı yapabilir veya merkezi sistemlere bilgi aktarabilirler.

Peki bu gömülü sistemler ve onların "kalbi" olan mikroişlemciler nasıl çalışır? Mikrokontrolcüler sensör verilerini nasıl işler? Robotlar ve araçlar gibi taşınabilir sistemler nasıl kontrol edilir? Bu sistemler nasıl haberleşir?

Bu kitap, "Robot Kontrol Ünitesi" olarak da adlandırdığımız mikrokontrolcülerin temel prensiplerini ve kullanım alanlarını tanıtarak, konuyu hem teorik hem de uygulamalı yönleriyle ele alacaktır. Ölçüm, iletişim, kodlama ve şifreleme, kontrol ve düzenleme, robotik uygulamalar ve otonom sürüş gibi birçok önemli alan pratik görevlerle desteklenecektir.

Mikrokontrolcü Nedir?

Bir mikrokontrolcü, bir mikroişlemci ile çeşitli giriş ve çıkışlardan oluşan bir kontrol sistemidir. Giriş birimlerine sensörler veya bellekler, çıkış birimlerine ise motor, LED veya ekran gibi aktüatörler bağlanabilir. Mikrokontrolcülerde, cihaz kapatıldığında silinen uçucu bir bellek (RAM) bulunur. Ayrıca programların ve verilerin kalıcı olarak saklanabildiği bir EEPROM da genellikle yer alır. PC ve dizüstü bilgisayarlarda olduğu gibi standart klavye ve ekran birimleri bulunmaz; bu bileşenler harici olarak kontrol edilir.

Mikrokontrolcülere örnek olarak Arduino, BBC micro:bit, Calliope Mini ve fischertechnik'in TXT 4.0 kontrolcüsü verilebilir.

Mikroişlemci Nasıl Çalışır?

Mikrokontrolcünün kalbinde bir mikroişlemci yer alır. Mikroişlemci, başlı başına bir entegre devre (IC) olup, talimatları ve verileri geçici bellekte (RAM) tutar. Veriler, işlem birimine aktarılır, burada işlenir ve sonuç tekrar belleğe yazılır.

Mikroişlemciler, esas olarak iki konuma (açık/kapalı) sahip transistörlerden oluşur. Bu transistörler, "ve", "veya", "değil" gibi mantıksal devreleri oluşturmak için kullanılır. Sonuçlar, "flip-flop" adı verilen kararlı bir anahtarlama elemanında saklanır. Bu kitap içinde yapacağınız uygulamalarla burada bahsettiklerimizi daha iyi anlayacaksınız.

Mikroişlemcinin çalışma hızı, bir kuartz kristali ile belirlenir. Kristalin titreşim frekansı, mikroişlemcinin saniyede kaç işlem yapabileceğini belirler. İşlem yapılacak talimatlar, mikroişlemcinin mimarisine uygun makine kodu ile ifade edilir. Bu kodlar bir komut (örneğin “toplama”) ve bu komuta bağlı parametrelerden (veri değerleri, bellek adresleri vb.) oluşur.

Mikroişlemcinin performansı;

- Saat hızı (Hz cinsinden işlem sayısı),
- Veri yolu genişliği (örneğin 32-bit ya da 64-bit),
- Makine kodunun karmaşıklığı

gibi faktörlere bağlıdır. İşleyebileceği komutlar ne kadar gelişmişse, mikroişlemci genellikle o kadar verimli çalışır.

Modern mikroişlemciler, talimatları ve verileri daha hızlı işleyebilmek için önbellek (cache) kullanır. Bu önbellek, sık kullanılan bilgilere daha hızlı erişim sağlar.

Günümüzde Mikroişlemciler

Günümüz dizüstü ve masaüstü bilgisayarlarda kullanılan mikroişlemciler:

- 64-bit veri işleyebilir,
- Birkaç milyar transistör içerir,
- Saniyede milyarlarca işlem gerçekleştirebilir (GHz düzeyinde saat hızları),
- Birden fazla çekirdekle çoklu görevleri eşzamanlı yürütebilir,
- Megabaytlarca önbellek belleği ile çalışır.

Buna karşın ”robot kontrol ünitesi” olarak da ele alacağımız mikrokontrolcülerde genellikle daha az güçlü işlemcilere ihtiyaç duyulur. Çoğunlukla 32-bit veri yolu genişliğine sahip, 1 GHz altında çalışan işlemciler bu görevler için fazlasıyla yeterlidir.

Mikrokontrolcülerin Tarihçesi

Otomatlar ve mekanik kontrol sistemleri, binlerce yıldır insanları büyülemiştir. Antik çağlardan itibaren, su veya hava gücüyle çalışan ve belirli hareketleri tekrar edebilen makineler tasarlanmıştır.

🔶 12. yüzyılda yaşamış olan **El Cezeri (1136–1206)**, zamanının ötesinde mühendislik çalışmalarıyla bilinir. Tasarladığı su gücüyle çalışan otomatlar, robotik sistemlerin ilk örnekleri arasında sayılır. Eserlerinde, zamanlayıcılar, mekanik müzik aletleri ve otomatik hizmetçiler gibi sofistike düzenekler yer alır. "Olağanüstü Mekanik Araçlar Kitabı" adlı eseri, bugünkü mekatronik sistemlerin temellerine ışık tutmaktadır.

🔶 **Leonardo da Vinci (1452–1519)**, tiyatro sahnelerinde veya saraylarda sergilenen mekanik makineler tasarlamıştır.

🔶 **Wolfgang von Kempelen (1734–1804)**, 1791 yılında bir konuşma sentezleyici geliştirmiştir. Ayrıca 1770 yılında meşhur satranç otomatı “The Turk”ü tasarlamış, bu otomat **Joseph von Racknitz** tarafından 1789 yılında detaylı biçimde çizilmiş ve yayımlanmıştır.

🔶 **1805 yılında Joseph-Marie Jacquard** tarafından geliştirilen delikli kartlı dokuma tezgâhı ise, ilk "programlanabilir otomat" olarak kabul edilir.

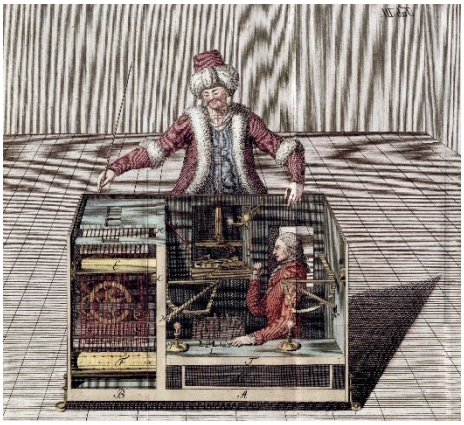
Ancak programlanabilir ve esnek kontrol sistemlerinin gelişimi esas olarak mikroişlemcilerin ortaya çıkmasıyla mümkün olmuştur.

🔷 **1958/1959’da** Jack Kilby ve Robert Noyce tarafından entegre devre (IC) icat edilmiş,
🔷 **1971’de** ise Texas Instruments tarafından ilk mikroişlemci olan **TMS1000** geliştirilmiştir. Bu çip yalnızca 8.000 transistör içeriyordu.

O günden bu yana, mikroişlemcilerdeki transistör sayısı her 18 ayda bir ikiye katlanmıştır – bu ilişki, **Moore Yasası**olarak bilinir.

Bugünün mikroişlemcileri milyarlarca transistör içerir, ancak fiziksel boyutları hâlâ oldukça küçüktür.

El Cezeri’nin *Saz Çalan Robot* Çalışması



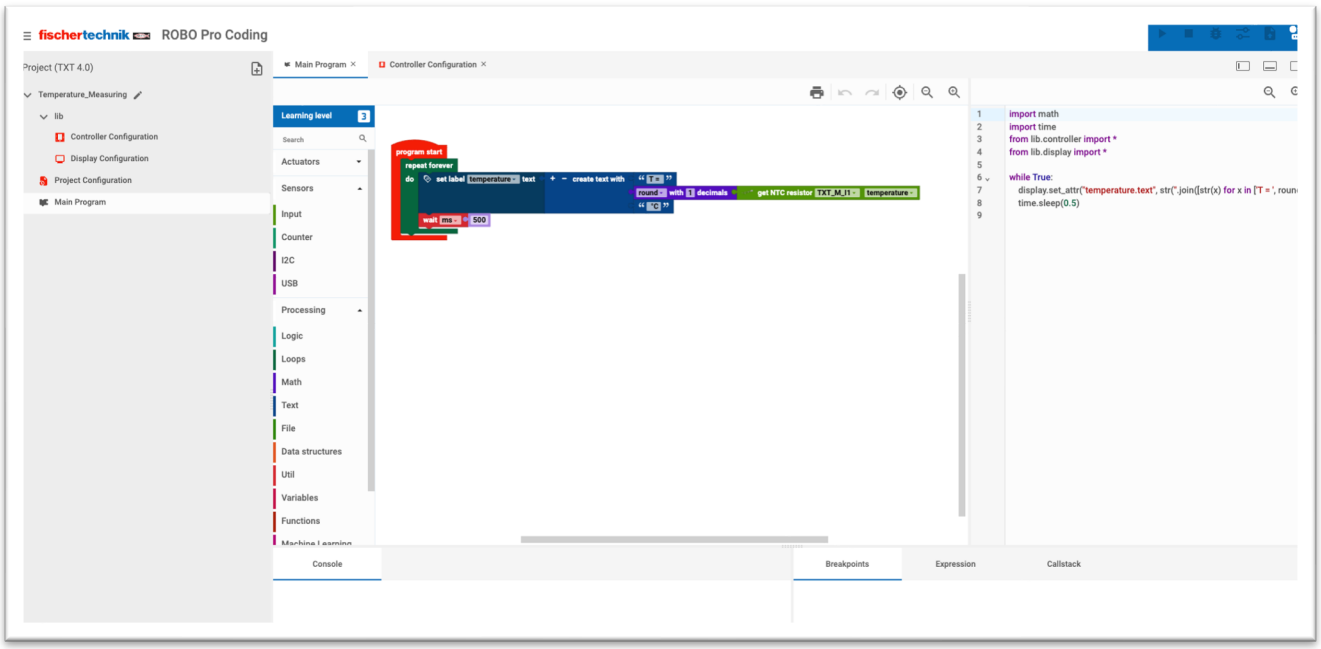
Racknitz’in *The Turk* illüstrasyonu

Mikrokontrolcü Nasıl Programlanır?

Mikrokontrolcüler, genellikle doğrudan üzerlerinden değil, bir bilgisayar, tablet veya akıllı telefon üzerinde çalışan bir **entegre geliştirme ortamı (IDE)** aracılığıyla programlanır.

Hazırlanan programlar USB, WLAN veya Bluetooth gibi bir iletişim arayüzüyle mikrokontrolcüye yüklenir ve burada çalıştırılır.

Örneğin **TXT 4.0**, ROBO Pro Coding üzerinden **Blok Tabanlı Görseller** ile veya **Python** ile programlanabilir. Program, geliştirilen cihaz üzerinden mikrokontrolcüye aktarılır ve orada çalıştırılır.



Robo Pro Coding ile TXT 4.0 Kontrol Ünitesi için örnek bir program

Robo Pro Coding;

- çok dilli programlama ortamı,
- grafik tabanlı programlama (Blockly ile blok tabanlı) veya Python ile metin tabanlı programlama,
- farklı öğrenme seviyeleri seçilebilir (başlangıç, orta, uzman),
- oluşturulan programları yerel olarak veya fischertechnik bulut depolamasında kaydetme imkânı,

sağlıyor.

Bu kitaptaki örnek uygulamalarda TXT 4.0 Kontrol Ünitesi'ni kullanacak ve Robo Pro Coding üzerinden yapacağınız programlarla robotlarınızı kontrol edebileceksiniz. Böylece bir mikrokontrolcünün nasıl programlandığını deneyimleyecek ve burada anlattıklarımızı daha iyi anlayacaksınız.

TXT 4.0 Kontrol Ünitesi

TXT 4.0 Kontrol ünitesi, gelişmiş özellikleriyle robotik projelerde güçlü bir kontrol ünitesi sunar! 512 MB RAM, 4 GB eMMC hafıza, 3 servo çıkışı ve kapasitif dokunmatik ekran gibi yenilikçi özelliklerle donatılmıştır. Dokunmatik ekran, kaydırma (swipe) hareketlerini destekleyerek sezgisel bir kullanım deneyimi sağlar.

Gelişmiş Bağlantı Seçenekleri

- WLAN & Bluetooth Modülü – Kablosuz bağlantı seçenekleri ile esnek kullanım.
- USB Host Portu – USB kamera veya USB bellek gibi cihazları bağlamak için kullanılabilir.
- Genişletilebilirlik – Tek bir kontrol birimine 8 ek denetleyici bağlanabilir, böylece projeler daha büyük ve kapsamlı hale getirilebilir.
- Düz Tasarım – İnce gövdesi sayesinde modellerin içine mükemmel şekilde entegre edilebilir.

Otomatik Güncellemeler & Programlama Esnekliği

TXT 4.0 Kontrol Ünitesi, otomatik bulut tabanlı firmware güncellemeleri alır, böylece her zaman en güncel özelliklere sahip olurken, kendi programlarınız korunur. ROBO Pro Coding yazılımı, hem grafiksel hem de Python kodlama desteği sunar. Bu yazılım, platform bağımsızdır ve mobil cihazlarda da kullanılabilir.

Sesli Kontrol Desteği

Bir mobil uygulama (Android / iOS) ile sesli komutlarla kontrol edilebilir, böylece projeler daha interaktif hale gelir!

TXT 4.0 Kontrol Ünitesi, yenilikçi özellikleriyle öne çıkan bir kontrol ünitesidir. 512 MB RAM, 4 GB eMMC bellek, üç servo çıkışı ve dokunmatik ekranıyla gelişmiş bir deneyim sunar.

Öne Çıkan Diğer Özellikler:

- Kablosuz Bağlantı: Geliştirilmiş WLAN ve Bluetooth modülü.
- USB Host Portu: USB kamera veya USB bellek gibi cihazları bağlama imkanı.
- Genişleme İmkanı: Sekize kadar ek kontrolcü bağlanabilir.
- Entegre Tasarım: İnce gövde yapısı modellerle mükemmel uyum sağlar.
- Otomatik Güncellemeler: Bulut üzerinden otomatik firmware güncellemeleri.
- Programlama Seçenekleri: ROBO Pro Coding ile grafiksel veya Python tabanlı programlama.
- Sesle Kontrol: Android/iOS uyumlu uygulama ile sesli komut desteği.

Robo Pro Coding Yazılımını Bilgisayara Kurmak

Adım 1

Öncelikle https://www.ftrobotik.com/programlar sayfasına gidin ve Robo Pro Coding başlığına ulaşın. Dilerseniz yandaki kare kod ile de bu sayfaya ulaşabilirsiniz.

Adım 2

Robo Pro Coding başlığı altında farklı işletim sistemleri için yazılım dosyalarının olduğu adresler belirtilmiştir. Bunlardan size uygun olanı seçin. Örnek olarak Windows işletim sistemine sahip bir PC ya da Notebook kullanıyorsanız "Robo Pro Coding – Windows" butonuna tıklayın. Eğer bir Mac kullanıcısıysanız "Robo Pro Coding – MAC/IOS" seçeneğini seçin.

Robo Pro Coding Program İndirme Sayfası Ekran Görüntüsü

Adım 3

İlgili link sizi programı yükleyeceğiniz Windows Store ya da Mac App Store'a götürecek. Oradan herhangi bir program yükler gibi bu programları yükleyeceksiniz.

TXT 4.0 Kontrol Ünitesini İlk Defa Kullanmak

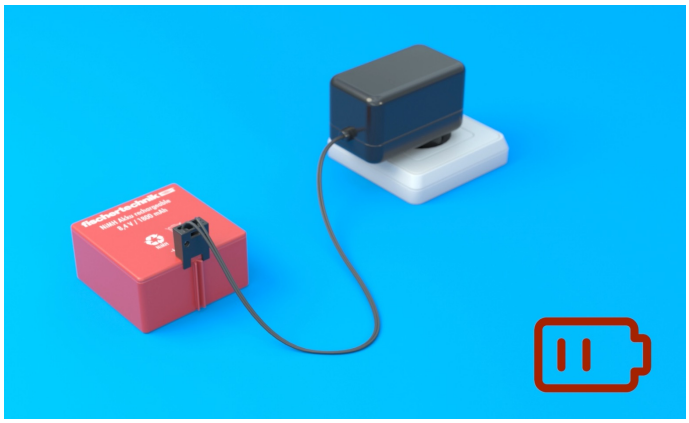
Video Anlatım İçin:

Bu bölümde anlatılan aşamaların videosuna ulaşmak için karekodu kullanabilirsiniz. İsterseniz video anlatımın devamında gösterilen ilk programı da yazıp test edebilirsiniz.

Adım Adım Görsel Anlatım:

Adım 1

Öncelikle set içerisinde çıkan aküyü, yine set içerisinde çıkan şary aletiyle şarj edin.

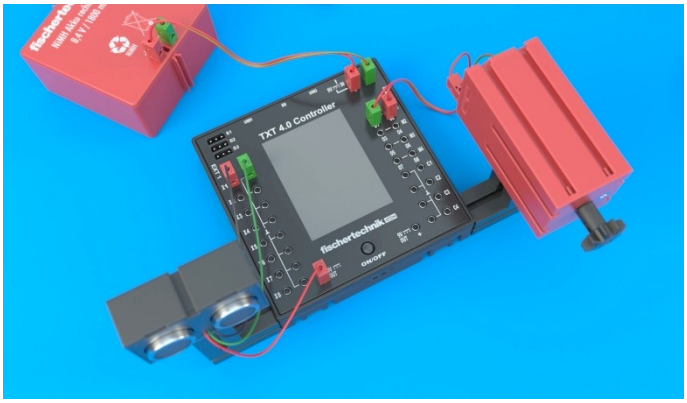


Adım 2

Modelinizi inşa edin. Her bölümde ilgili modeli nasıl inşa edeceğiniz anlatılıyor.

Adım 3

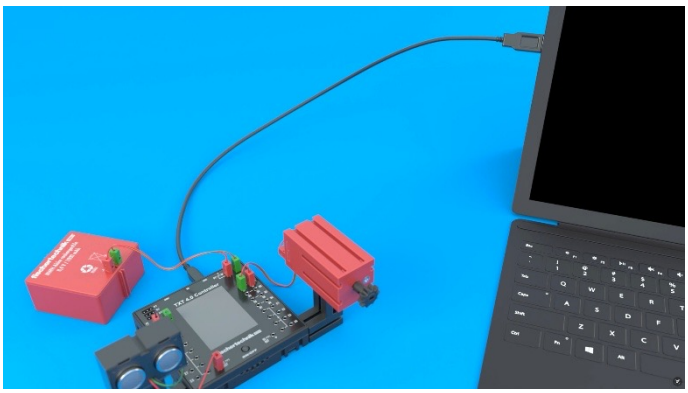
Modelinizin kablo bağlantılarını yapın. Pilin + ve – kutuplarına dikkat edin. Her bölümde ilgili modelin kablo bağlantılarını nasıl yapacağınız model yapımı sonrasında anlatılıyor.



TXT 4.0 Kontrol Ünitesini İlk Defa Kullanmak

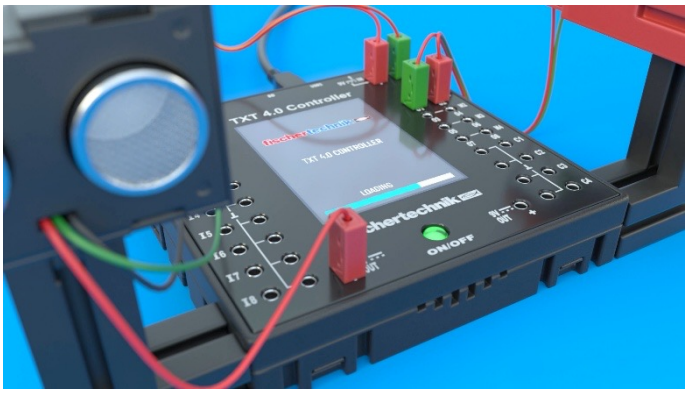
Adım 4

Kontrol ünitesini bilgisayara bağlayın. Resimde USB kablo ile nasıl bağlayacağınız gösterilmiş durumda. İsterseniz kablosuz bağlantı da kurabilirsiniz.



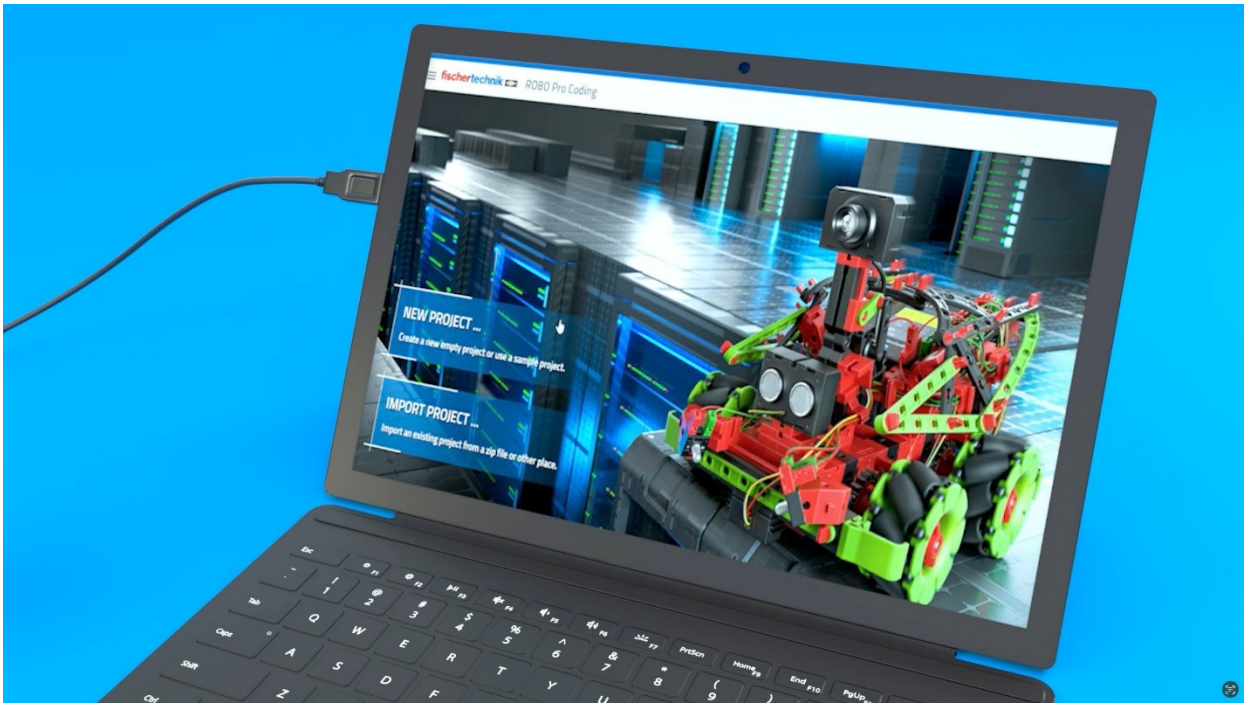
Adım 5

TXT 4.0 Kontrol Ünitesinin ON/OFF yani AÇ/KAPA tuşuna basılı tutarak açılmasını sağlayın. Kontrol Üniteniz kısa bir süre içinde açılacaktır.



Adım 6

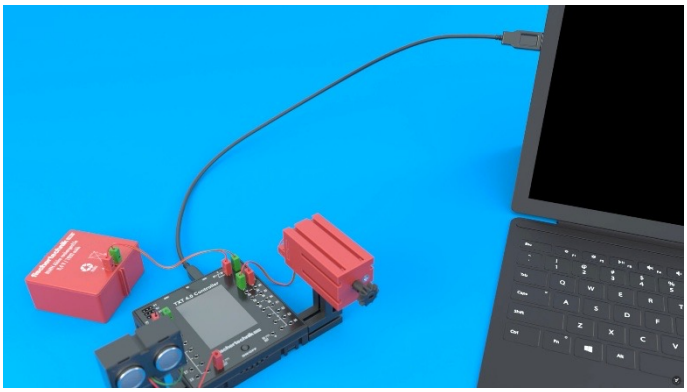
Bilgisayarınızda “Robo Pro Coding” uygulamasını açın ve “NEW PROJECT” yani “YENİ PROJE” seçeneğini seçin.



TXT 4.0 Kontrol Ünitesini İlk Defa Kullanmak

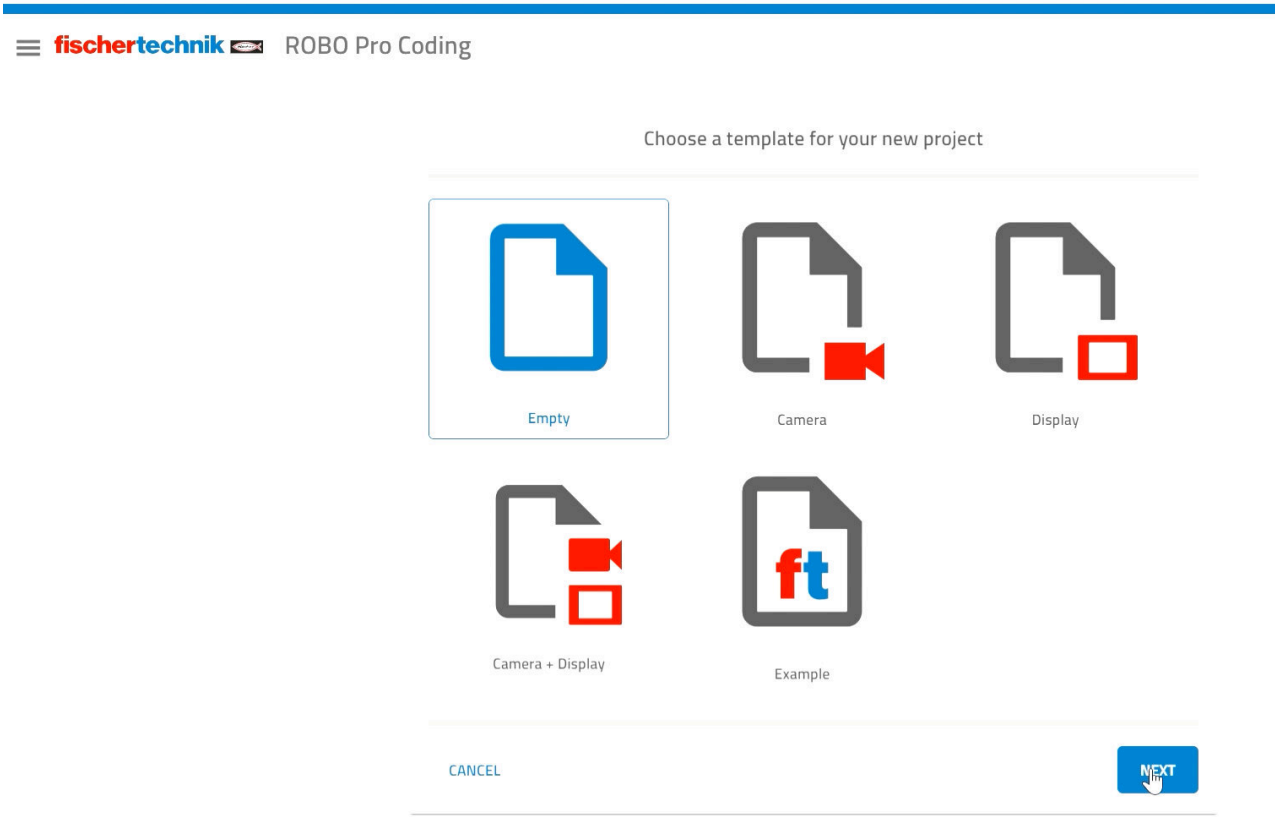
Adım 7

Kontrol ünitesini bilgisayara bağlayın. Resimde USB kablo ile nasıl bağlayacağınız gösterilmiş durumda. İsterseniz kablosuz bağlantı da kurabilirsiniz.



Adım 8

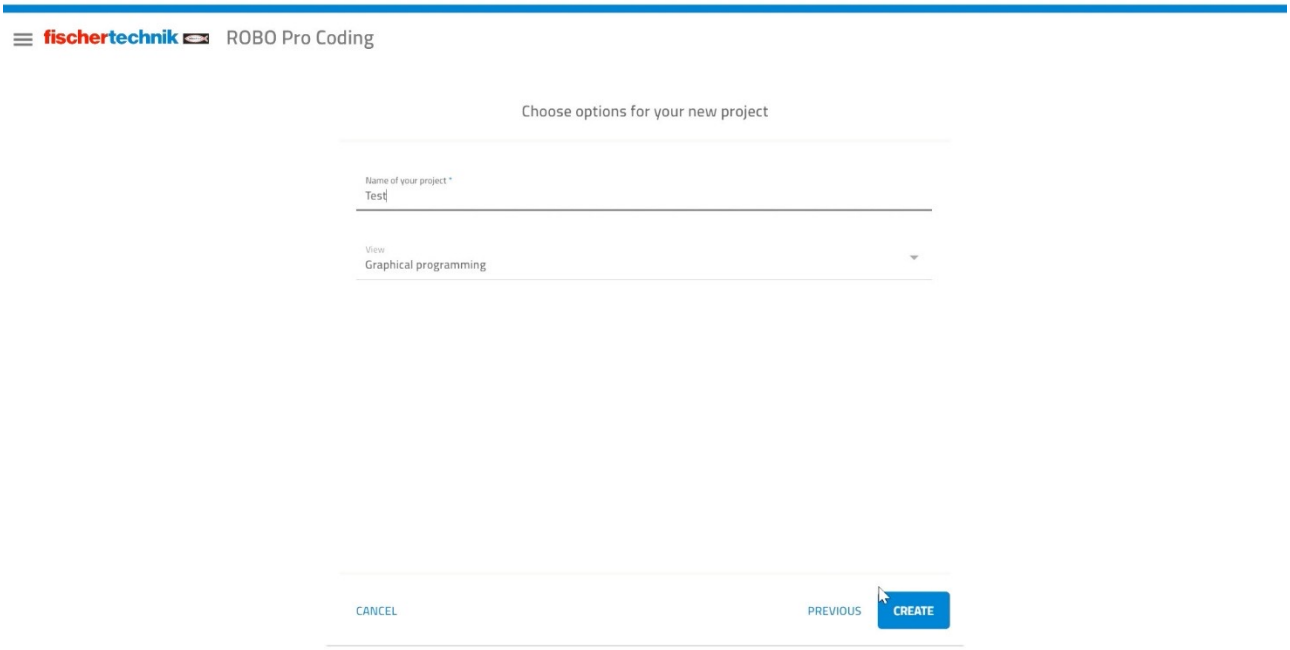
Bu adımda boş bir programlama sayfası açmak için “Empty” yani “Boş” seçeneğini seçip “Next” yani “İleri” tuşu ile ilerliyoruz.



TXT 4.0 Kontrol Ünitesini İlk Defa Kullanmak

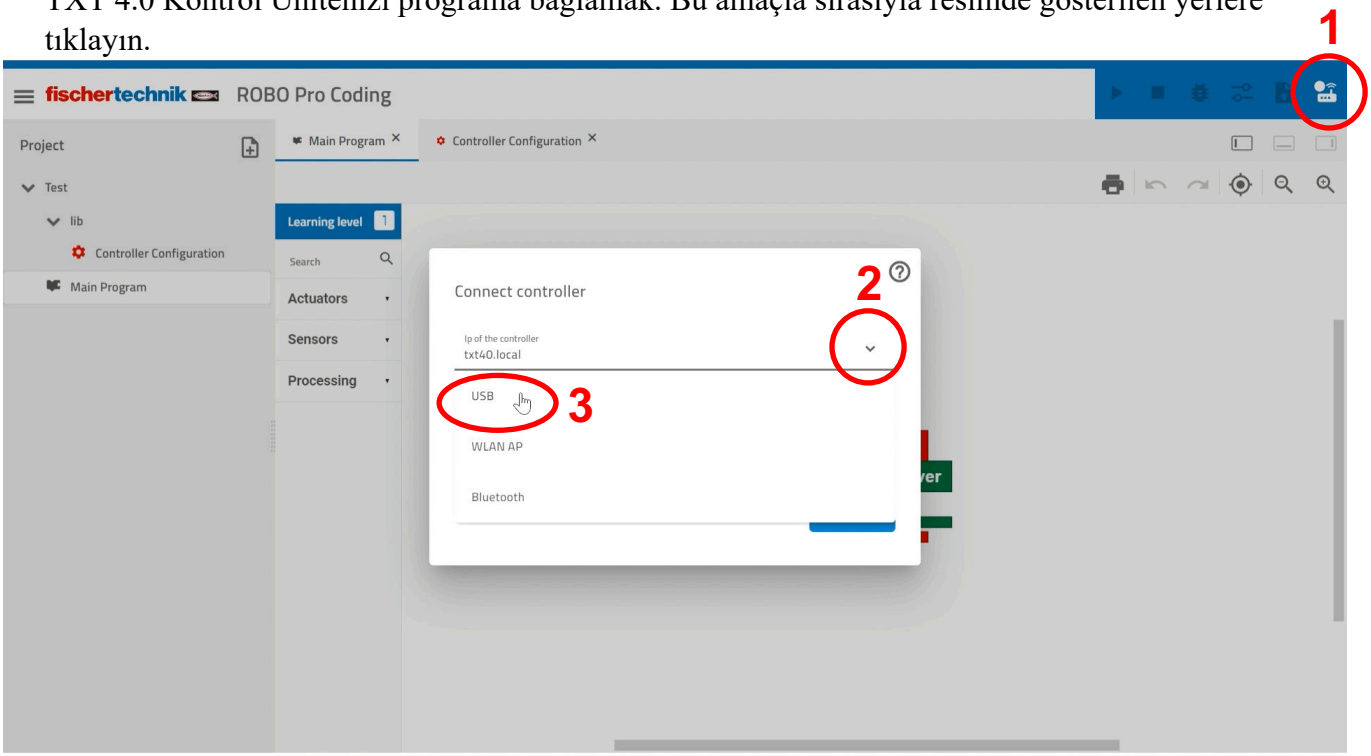
Adım 9

Bu adımda sayfamıza bir isim vermemiz lazım. Biz “Test” ismini verdik. Siz isterseniz farklı bir isim verebilirsiniz. Ardından “Create” yani “Oluştur” tuşu ile ilerliyoruz.



Adım 10

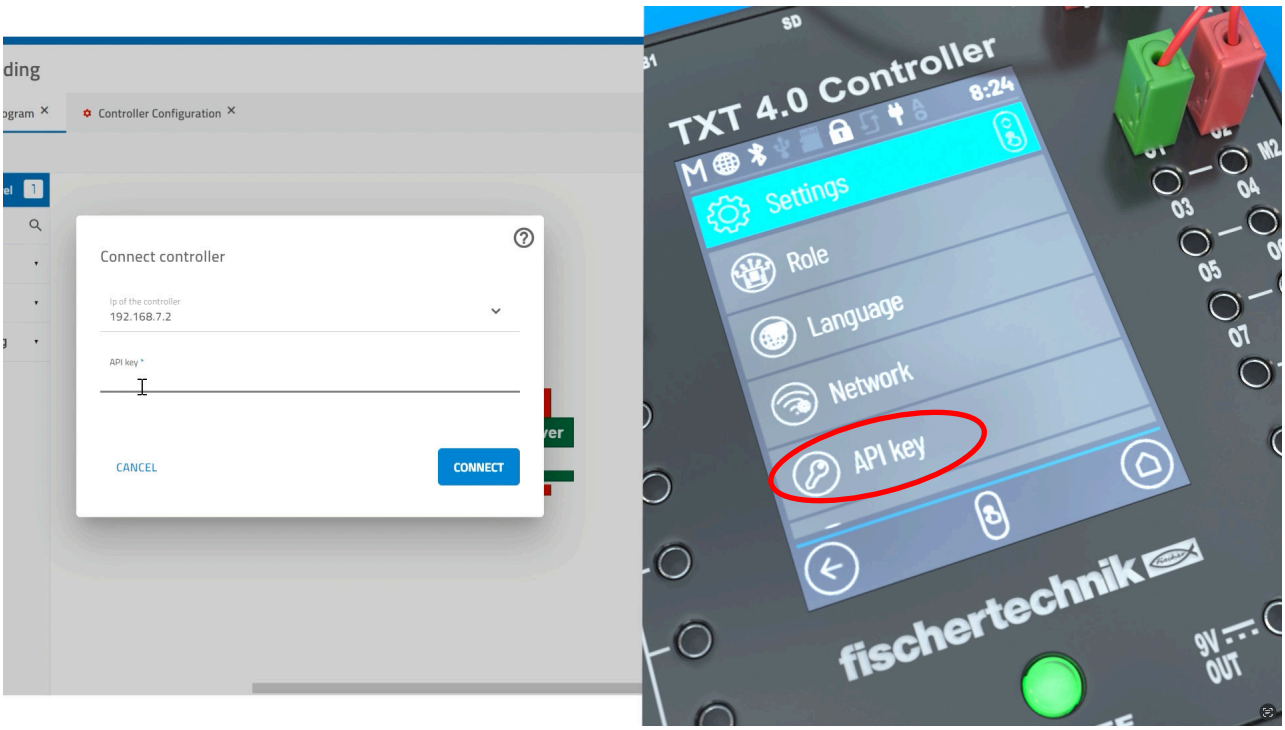
Tebrikler. Programlama ortamınız karşınızda. Şimdi yapmanız gereken bir adım daha kaldı. O da TXT 4.0 Kontrol Ünitenizi programa bağlamak. Bu amaçla sırasıyla resimde gösterilen yerlere tıklayın.



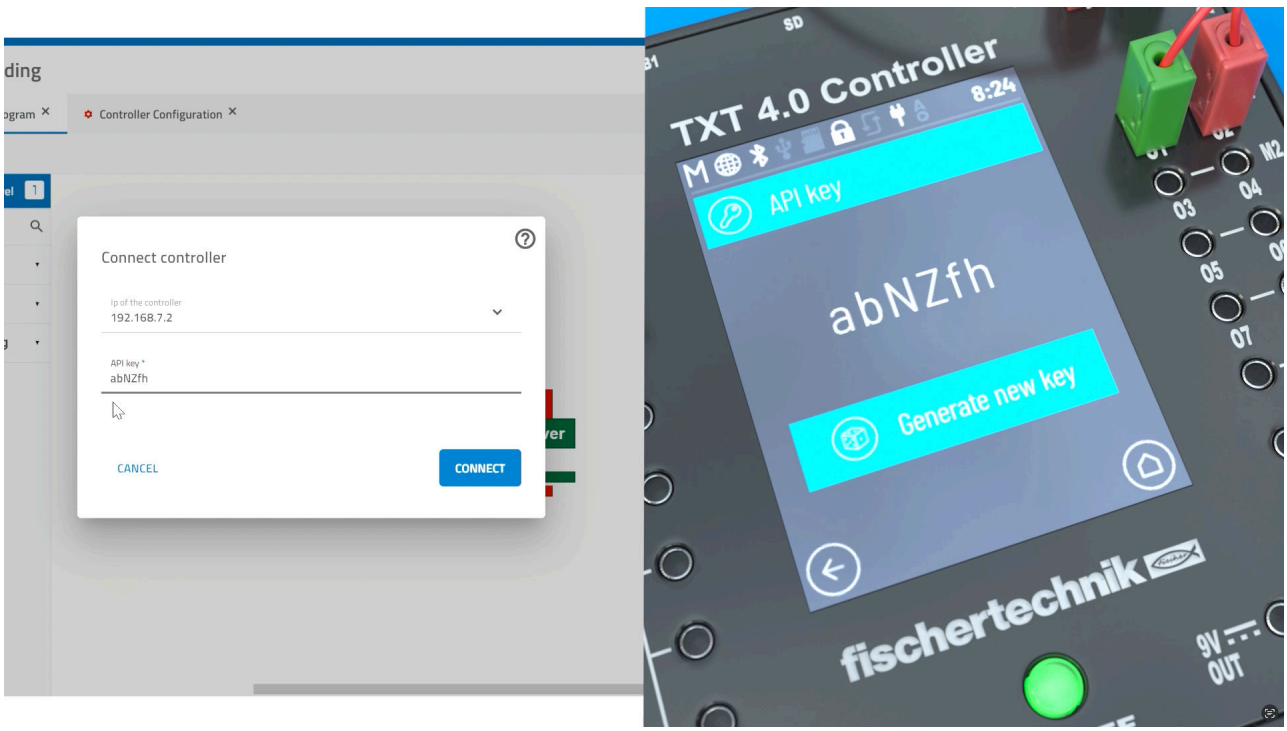
TXT 4.0 Kontrol Ünitesini İlk Defa Kullanmak

Adım 11

Şimdi tek bir işlemimiz kaldı. Kontrol Ünitemiz içerisinde Settings menüsü altında “API key” seçeneğini bulmak.



Buradaki kodu API key alanına yazıp “CONNECT” yani “BAĞLAN” tuşuna basıyoruz. Böylece bağlantımız kurulmuş oluyor. Artık program yazabilirsiniz.



ftrobotik

fischertechnik

BÖLÜM 03

Barkod Okuyucu

Öğrenme hedefi

- ✓ Kodların pratik önemini anlamak.
- ✓ Hata tespiti ile görsel 1D kodunun (barkod) çözülmesi
- ✓ RGB renk kodu ve renk tonunun (renk çarkındaki derece) belirlenmesi
- ✓ Öğrencilerin kendi çok boyutlu kodlarını geliştirmesi

Barkod Okuyucu Nasıl Çalışır?

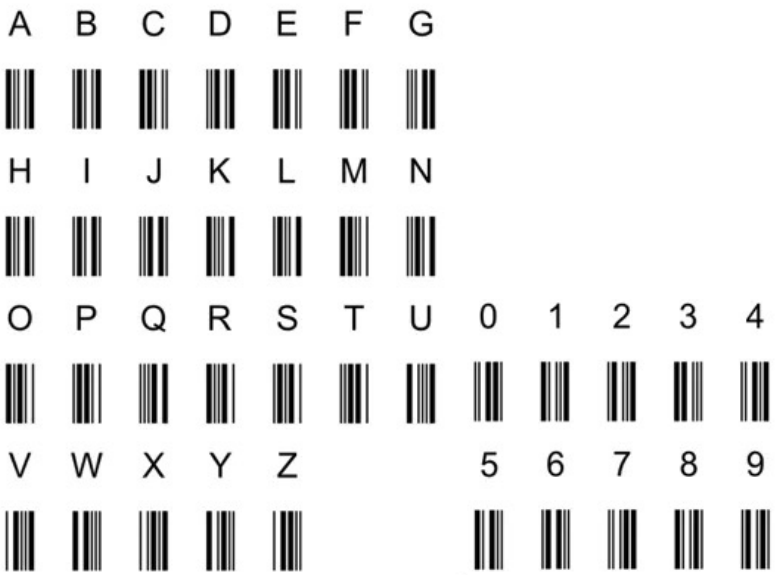
Barkodlar Nasıl Çalışır?

Bir barkod, farklı genişlikte siyah ve beyaz dikey çubuklardan oluşan bir “çubuk kod” veya “çizgi kod”dur. Barkodlar, bilgiyi yalnızca çubukların genişliğinde barındırdıkları için bir boyutlu kodlar sınıfına girer (örneğin dar çubuk = “0”, geniş çubuk = “1”). Çubukların siyah/beyaz renkte olması ise sadece onların ayırt edilebilmesini sağlar. Barkodlar “optik kodlar”dır; yani “açık” ve “koyu” ayrımını yapabilen bir sensör veya görüntü işleme yapabilen bir kamera ile okunabilirler. Barkodlar ağırlıklı olarak nesnelerin işaretlenmesinde kullanılır; kitaplarda ve dergilerde (ISBN/ISSN), ürün etiketlerinde veya paket postalarının üzerindeki çıkartmalarda bulunurlar. Çok farklı barkod tipleri vardır ve bunlar bilgi yoğunluğu ve kullanılan kontrol basamakları (check digit) bakımından birbirlerinden ayrılırlar. Barkodların avantajı, çok dayanıklı olmaları ve kolayca okunabilmeleridir; ayrıca okuyucular da çok ucuzdur.

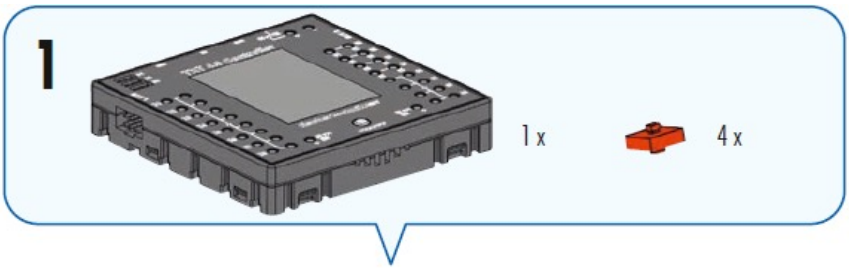
Tipik bir barkod örneği Code 39’dur. Bu kod ilk olarak 1973’te Intermec firmasında geliştirilmiş ve o zamandan beri ANSI, ISO gibi çeşitli standart kuruluşları tarafından farklı spesifikasyonlar veya standartlarda tanımlanmıştır (örneğin ISO/IEC 16388). Bir Code 39 karakteri, ikisi siyah ve biri beyaz olacak şekilde toplam dokuz çubuktan – beşi siyah ve dördü beyaz (aralarındaki boşluklar) – oluşur ve her biri iki farklı genişlikte olabilir. Geniş bir çubuk “1”i, dar bir çubuk “0”ı temsil eder; bu nedenle bir Code 39 karakteri dokuz haneli bir ikili sayıya denktir.

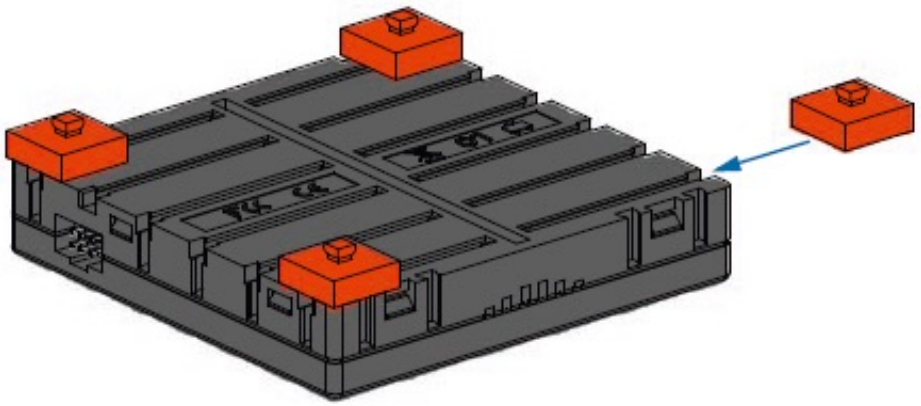
Her karakterin önünde ve arkasında dar bir beyaz boşluk bulunur, böylece karakterler birbirinden kolayca ayırt edilebilir. Code-39 barkodlarının genişliği her zaman aynıdır, çünkü her karakter tam üç geniş çubuk içerir: iki siyah ve bir beyaz. Bu nedenle her Code-39 karakteri, altı dar ve üç geniş çubuktan veya altı “0” ve üç “1”den oluşur. Code-39 standardı toplamda 44 karakter tanımlar: alfabenin 26 harfi, “0”dan “9”a kadar olan on rakam ve sekiz özel karakter: “*” (başlangıç/durma işareti), boşluk, “-“, “+”, “.”, “/”, “%” ve “$”.

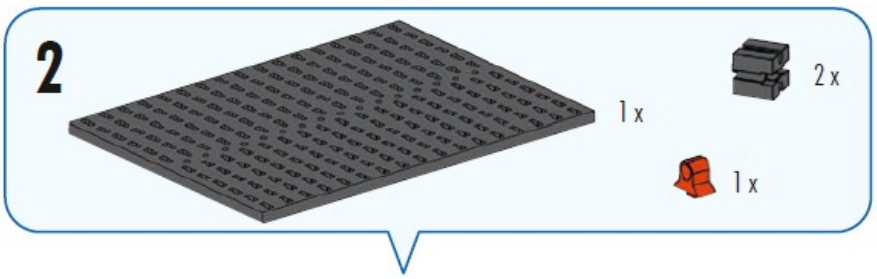
Barkodların bir gelişmiş sürümü ise QR kodlarıdır. QR kodları, kare bir yüzey üzerinde noktalardan oluşur. Bunlar iki boyutlu kodlardır ve okunup kodlarının çözülmesi için bir kamera gerektirir.

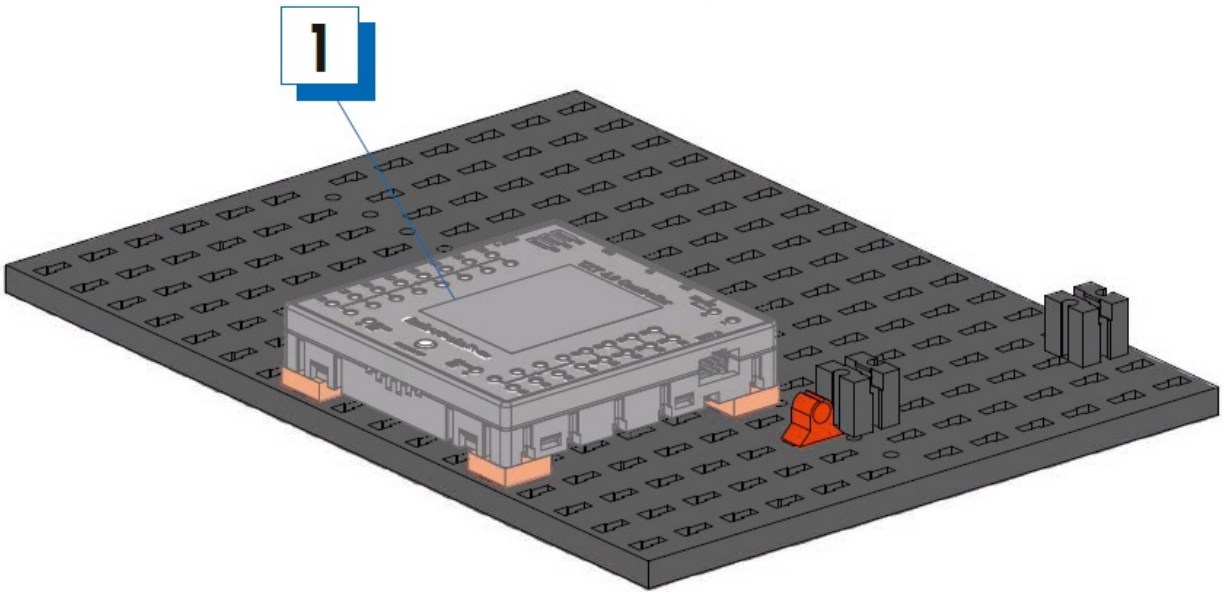


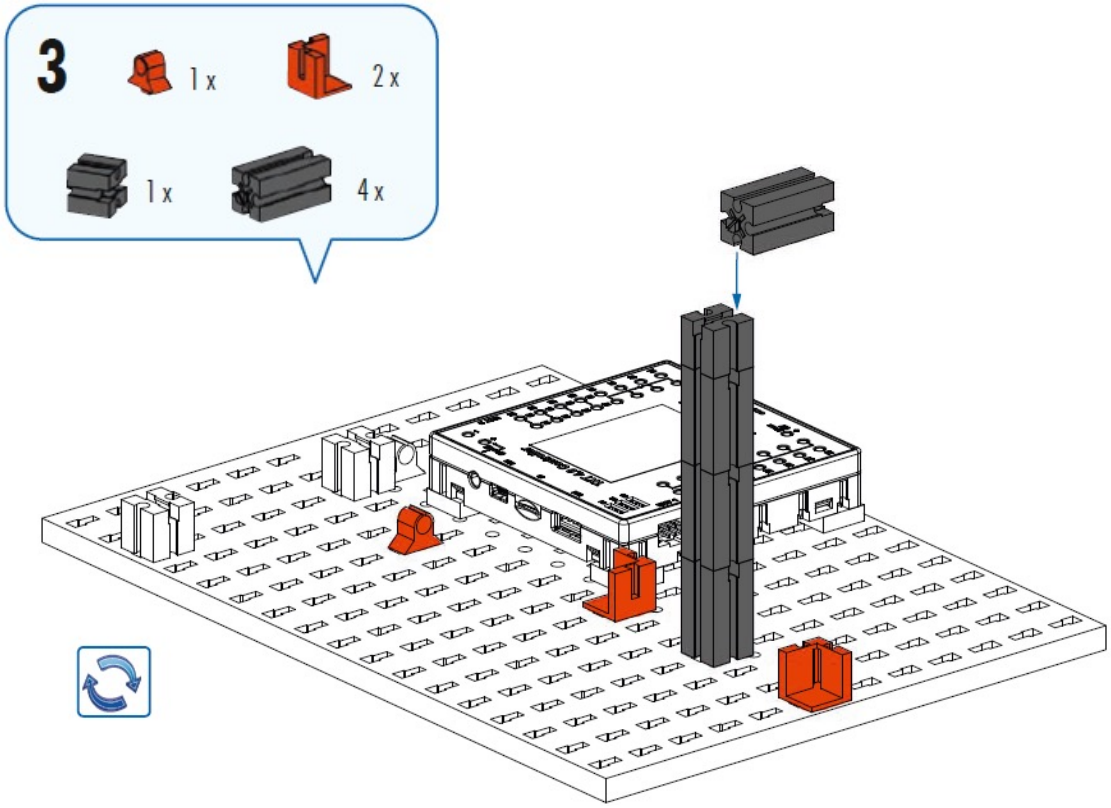
MODEL 3 Barkod Okuyucu

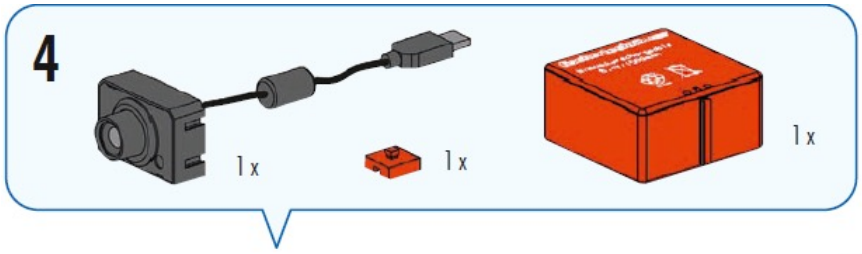


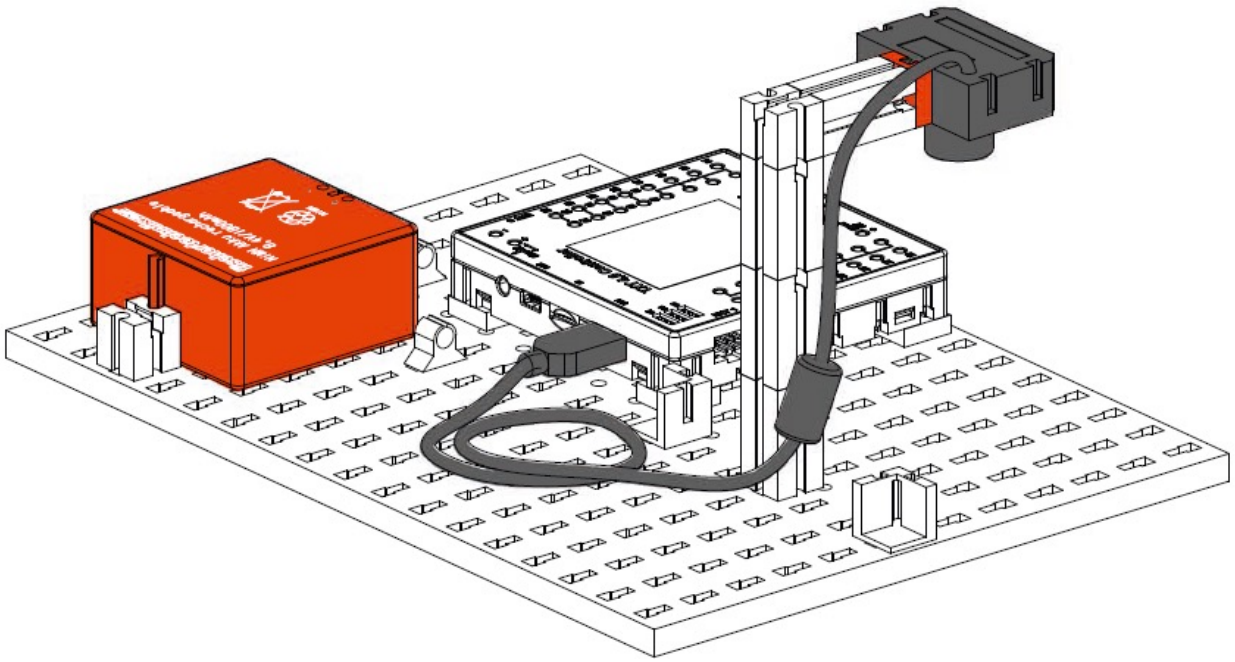




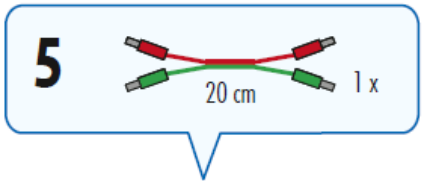


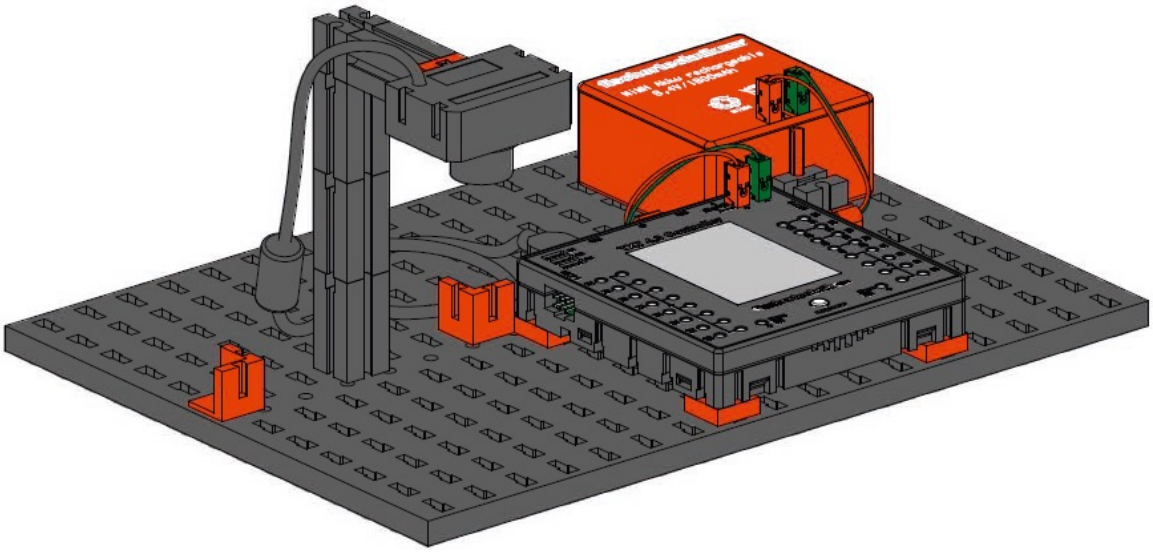

3
1 x
2 x
1 x
4 x


4
1 x
1 x
1 x

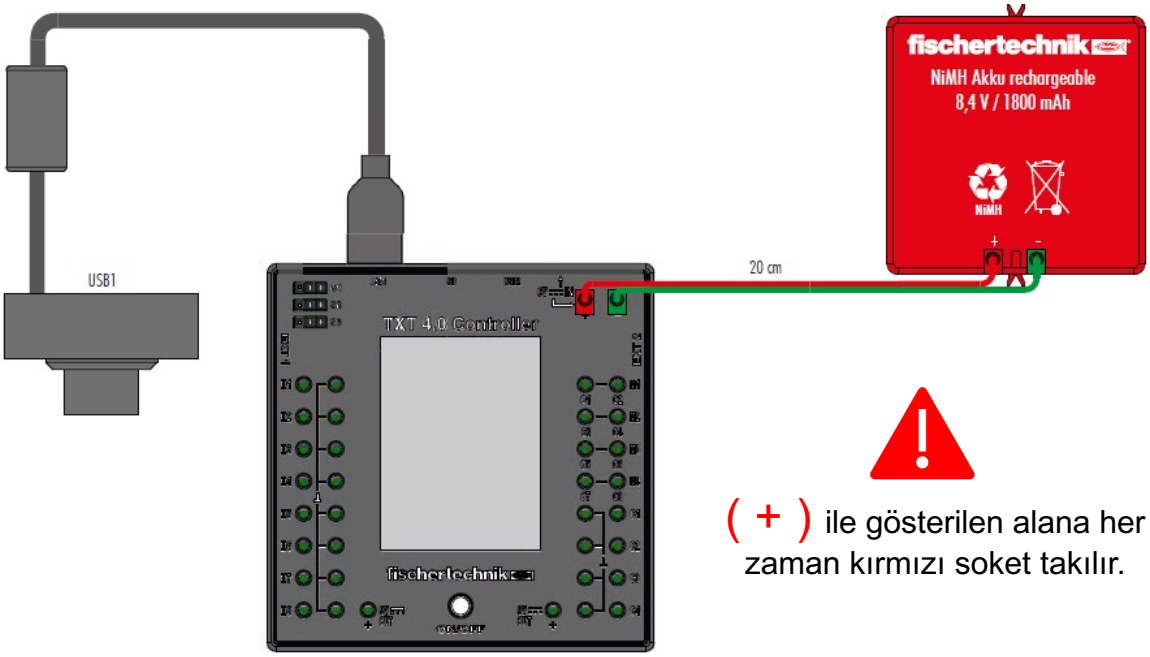


Model - 3
Barkod Okuyucu





Devre Şeması



(+) ile gösterilen alana her zaman kırmızı soket takılır.

Barkod Okuyucu Öğrenme Hedefleri

Konu

Bu görev kodlamayla ilgilidir. Öğrenciler önce bir kamera kullanarak bir barkodu (Kod 39) çözecek, ardından RGB renk kodlarını tonlara dönüştürecekler. Öğrenciler deneysel görev sırasında kendi kodlarını geliştirecekler.

Kodlama, hata tespiti ve bilgilendirme içeriği.

Öğrenme hedefi

- Kodların pratik önemini anlamak.
- Hata tespiti ile görsel 1D kodunun (barkod) çözülmesi
- RGB renk kodu ve renk tonunun (renk çarkındaki derece) belirlenmesi
- Öğrencilerin kendi çok boyutlu kodlarını geliştirmesi

Gerekli Malzemeler

- Programlama yapmak için bilgisayar veya tablet.
- Programı TXT4.0 iletmek için USB kablo, BLE veya WiFi bağlantısı.
- Program şablonu (Kod 39): Kod 39 Şablonu.ft
- Çözülmesi gereken kod 39 örnek (genişlik: 6 cm). Renk çemberi ve renk yıldızı
- Öğrencilerin kendi kodlarını tasarlamaları için kağıt ve renkli kalemler

Daha Fazla Bilgi

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

[2] Wikipedia: *Strichcode*, *Code 39*.

[3] RapidTables: *RGB Color Codes Chart*.

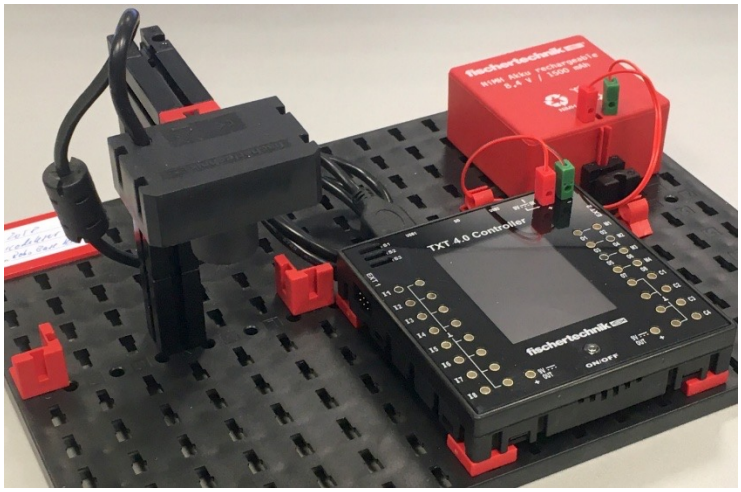
Barkod Okuyucu

TASARIM GÖREV ÇÖZÜM

Barkod Okuyucu İçin Verilen Görevler

Modelin kurulumu:

Yapım talimatlarına göre modelini inşa edin. USB kamerayı TXT'ye (USB1) bağlayın. Kesme çizgilerindeki Kod 39 örneklerini ve renk çemberini kesin.



Barkod Okuyucu modelin kurulumu

Programlama görevleri (GÖREV)

1. Çizgi tespiti:

Kamera çizgileri (çubukları) algılayabilir. Bunu yapmak için önce kamera penceresinde çizgi algılamayı yapılandırmalı, ardından lensi çevirerek kamerayı odaklamalısınız.

1a. Siyah dikey bir çubuğu algılayan ve TXT'nin ekranında ne kadar geniş olduğunu gösteren bir program yazın.

1b. Programı, yan yana duran iki dikey siyah çubuğu algılayabilecek ve aralarındaki (beyaz) boşluğun ne kadar geniş olduğunu belirleyebilecek şekilde genişletin. Boşluğun genişliğini TXT'nin ekranında çıktı olarak verin.

2. 9- Rakam barkodu:

Kamera aynı anda beş çubuğa kadar algılayabilir. Aralarındaki dört beyaz boşluğu eklerseniz, dokuz basamaklı bir barkodu çözmek için kullanılabilir.

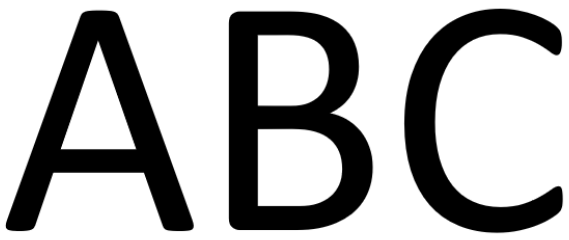
"0" ve "1" için iki farklı çubuk genişliği kullanarak dokuz basamaklı ikili bir barkodla kodlanmış karakterleri algılayan ve ikili kodun (sayısal) değerini TXT'nin ekranında çıkaran bir program yazın.

Test etmek için dahil edilen Code 39 barkodunu kullanın.

3. Kod 39 kod çözücü

Programınızı, dahil edilen kod 39 barkodunu çözecek ve algılanan karakterleri TXT'nin ekranında gösterecek şekilde genişletin. Bunu yapmak için, iki liste içeren "Code39_Template.ft" program parçasını kullanın: alfabedeki tüm rakamlar ve harfler ve bunların kod 39 kodlaması.

TASARIM GÖREV ÇÖZÜM



Kod 39 "ABC" gösterimi (başlangıç/bitiş karakterleri olmadan)

4. Hata tespiti

Kod 39'da, dokuz çubukla temsil edilebilen tüm kodlar geçerli değildir.

4a. Kod 39 kullanılarak kaç farklı sinyal kodlanabilir?

Kodun yedekliliğini veya herhangi bir bilgi kaybı olmadan dışarıda bırakılabilen bilgi birimlerini (bitleri) belirleyin.

4b. Kod 39'un yüksek düzeydeki yedekliliği hataları tespit etmeyi mümkün kılar. Geçersiz kod sözcüklerini tespit etmenin en kolay yolu nedir?

4c. Bir sinyal yanlış kodlanmışsa tespit edilen hatanın ekranda gösterilmesi için programınızı bir kod 39'u kod çözmek üzere genişletin.

Deneysel görevler (GÖREV)

1. RGB kodu:

Renkler bilgisayarda bir kod kullanılarak da temsil edilir. Örneğin, yaygın olarak kullanılan RGB kodu, bir renk tonunda Kırmızı, Yeşil ve Mavi renklerinin yüzdelerini gösterir. Kamera tarafından algılanan bir alanın RGB kodunu döndürmek için Blockly'yi kullanabilirsiniz.

1a. TXT ekranında kameranın altındaki (renkli) bir alan için RGB kodunu çıktı olarak veren bir Blockly programı yazın.

Renk tonunu doğrudan RGB kodundan türetemezsiniz. Ancak, RGB kodu renk çemberindeki bir konuma dönüştürülebilir. Bunu yapmak için ihtiyaç duyduğunuz hesaplama kuralı, ilgili renk tonu değerini (0 ile 1 arasında bir sayı) verir ve eşlik eden materyallerde verilmiştir. Elde edilen değeri 360° ile çarparsanız, renk çemberindeki açıyı elde edersiniz.

1b. Programınızı, kameranın altına yerleştirdiğiniz renkli bir yüzey için RGB kodundan ilişkili renk tonu değerini hesaplayacak şekilde genişletin. Program, RGB kodunu, renk tonunu ve hesaplanan açıyı TXT ekranındaki renk tekerleğinde göstermelidir.

Programı dahil edilen renk tekerleğini kullanarak test edin.

2. Renk tanıma

Dahil edilen renk yıldızındaki sekiz renkli alanın sınırlarını belirleyin (derece değerleri olarak). Tespit edilen rengin TXT ekranında çıktı olarak verilmesi için programınızı genişletin.

3. Çok boyutlu kod

Kod 39 yalnızca çubukların genişliklerini değerlendirir; renk değişiklikleri (beyaz/siyah) yalnızca bitişik çubukları ayırt etmeye yarar. Bu, her çubuğun yalnızca 1 bit bilgi içerdiği anlamına gelir. Bu nedenle, kod "tek boyutlu" veya "ID" kodu olarak kabul edilir.

3a. Altı renk ve altı çubuk genişliği arasında ayrım yapan çok boyutlu bir kod geliştirin, böylece yalnızca bir çubukla 36 karakteri (26 harf ve 10 rakam) kodlayabilirsiniz. Kodu bir tabloda not edin.

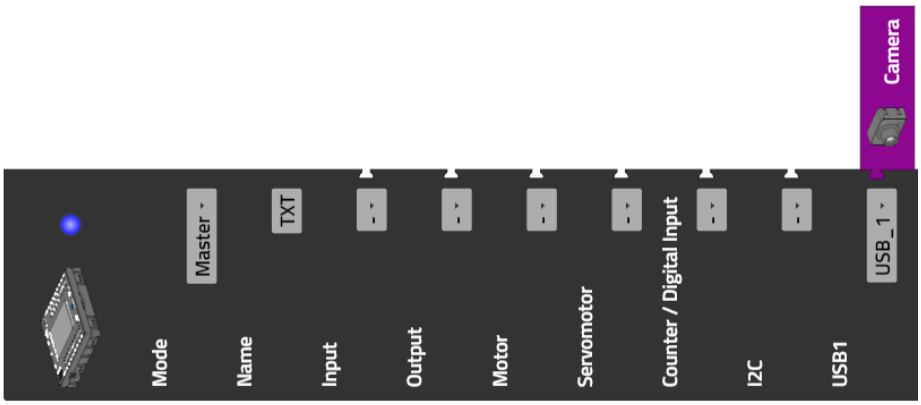
3b. Bu kod için bir kod çözücü programlayın. Kod çözülen karakter TXT'nin ekranında çıktı olarak verilmelidir. Programı kendiniz çizdiğiniz renkli çubuklarla test edin.

Barkod okuyucu İçin Verilen Çözümler

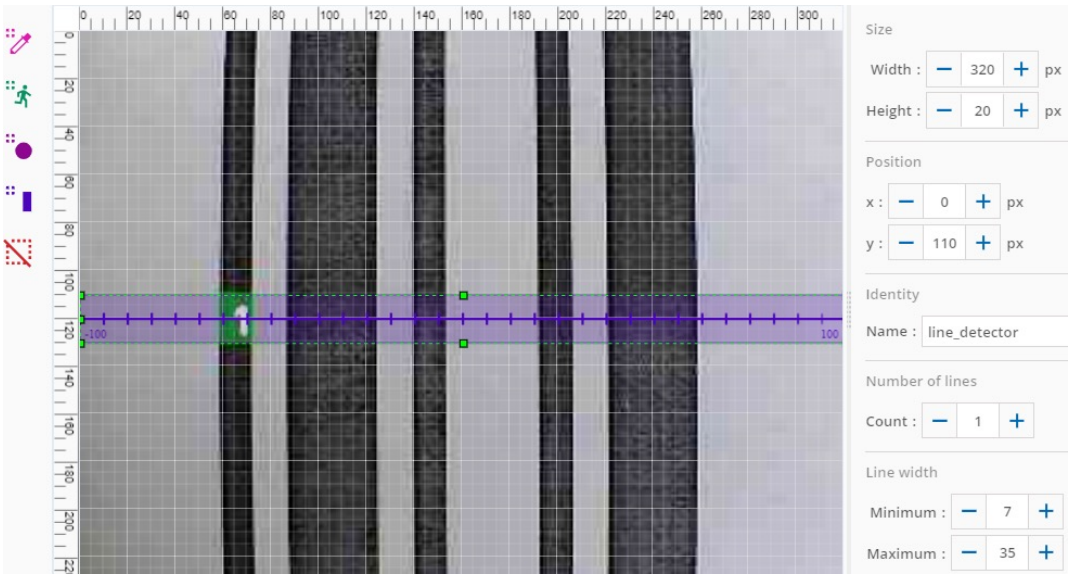
Programlama görevleri (ÇÖZÜM)

1. Çizgi tespiti:

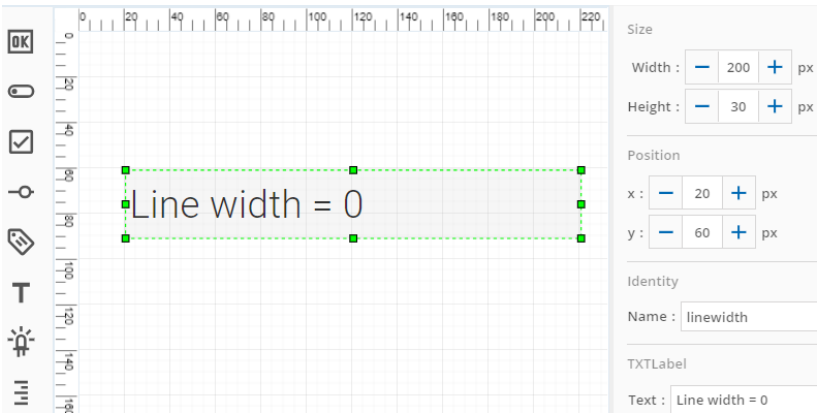
USB kameranın yapılandırılması:



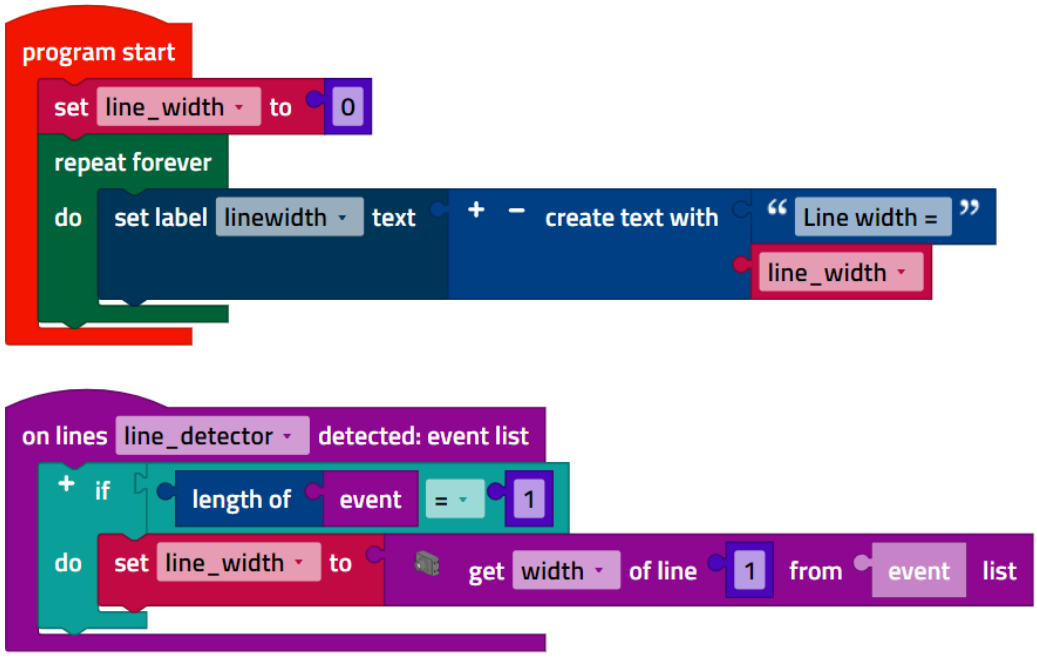
1a. Hat algılamanın yapılandırılması:



Ekran yapılandırması:

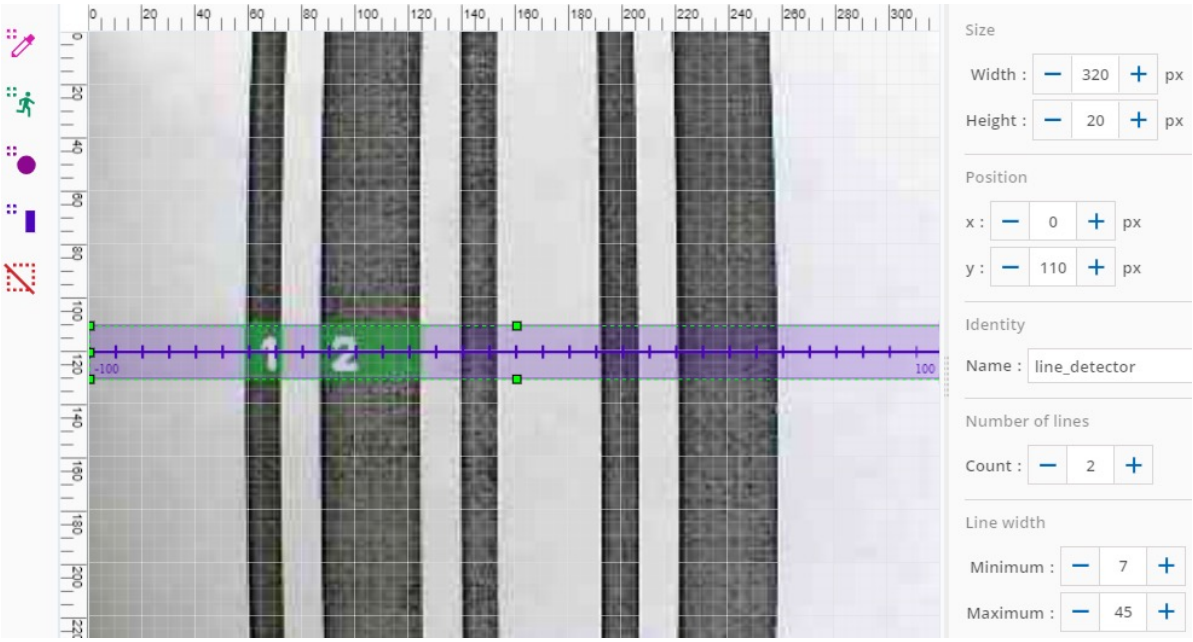


Program (örnek): Dikey siyah çizginin genişliğini belirleme

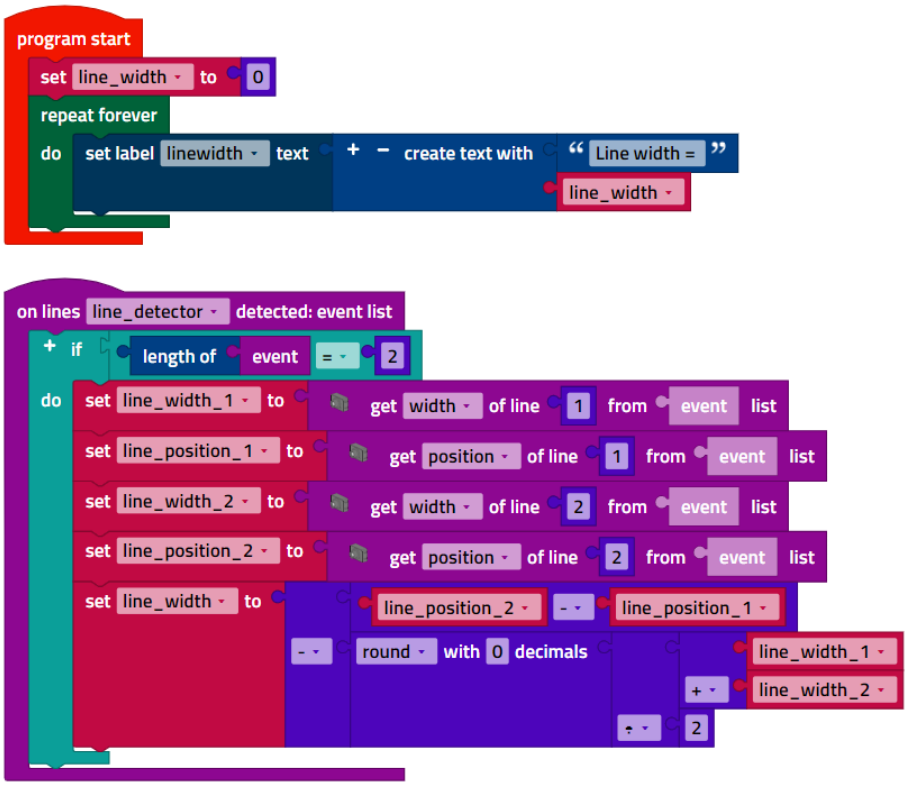


Line_Detection.ft

1b. Hat algılamanın yapılandırılması:



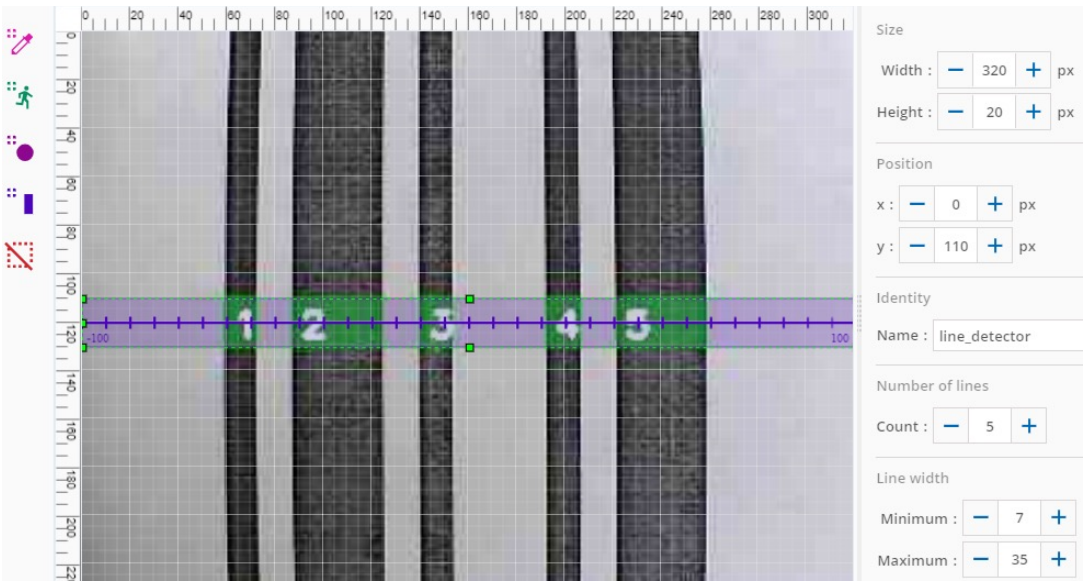
Program (örnek): İlk boşluğun genişliğinin belirlenmesi:



Bar_Width_Recognition.ft

2. 9- Rakam barkodu

Beş satırın algılanmasının yapılandırılması::

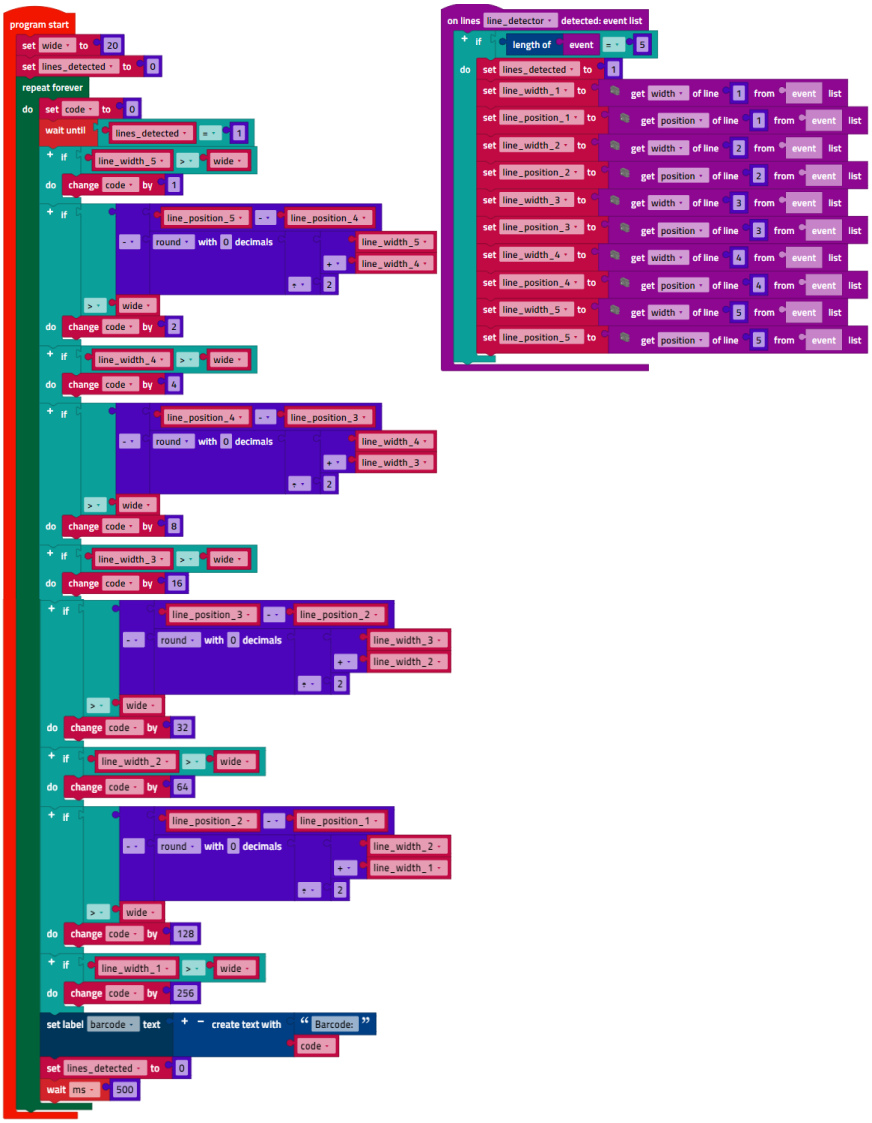


Kodun algılanan (siyah) çubukları soldan sağa doğru sayılır. Bunu değerlendirmek için kod önce ikili kod olarak yorumlanır: Geniş (siyah veya beyaz) bir çizgi 1'i, dar bir çizgi ise 0'ı temsil eder.

Şekilde gösterilen kod dokuz basamaklı ikili sayıya karşılık gelir 001001001b = 73.

Ekran yapılandırması:

Program (örnek): Dokuz basamaklı bir barkodu çöz
“lines_detected” değişkeni burada bir semafor görevi görür: Ana programa, satır algılamanın beş satır algıladığını bildirir:



Barcode_Decoder.ft

3. Kod 39 kod çözücü

"Code39_Template.ft" program şablonundaki listeler, kodun ikili değerini ek bir program satırı ve değiştirilmiş bir çıkış komutuyla kodlanmış sinyale "çevirmek" için kullanılabilir.
Program özeti (örnek):

Code39_Decoder.ft

4. Hata tespiti:

39 kodu 40 farklı sinyali kodlamak için kullanılabilir:

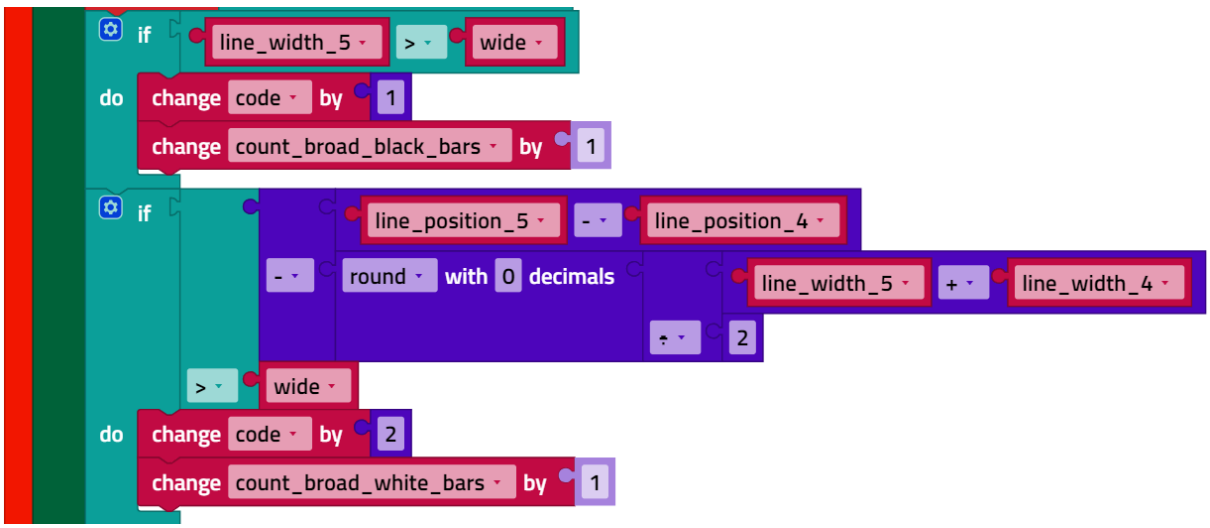
- İki geniş ve üç dar siyah çizgi 10 farklı kombinasyona olanak tanır ve bu kombinasyonların her biri için
- Geniş beyaz çizgi dört konumdan birinde olabilir.

Dolayısıyla kodun bilgi içeriği (= 40 sinyali kodlamak için gereken minimum bit sayısı) 6 bittir (26 = 64).
9 - bitlik bir kodla (29 = 512) bu kadar çok sinyali 8 kat daha fazla temsil edebilirsiniz. Bu nedenle, kod nispeten düşük bir bilgi içeriğine ve yüksek bir yedekliliğe sahiptir.

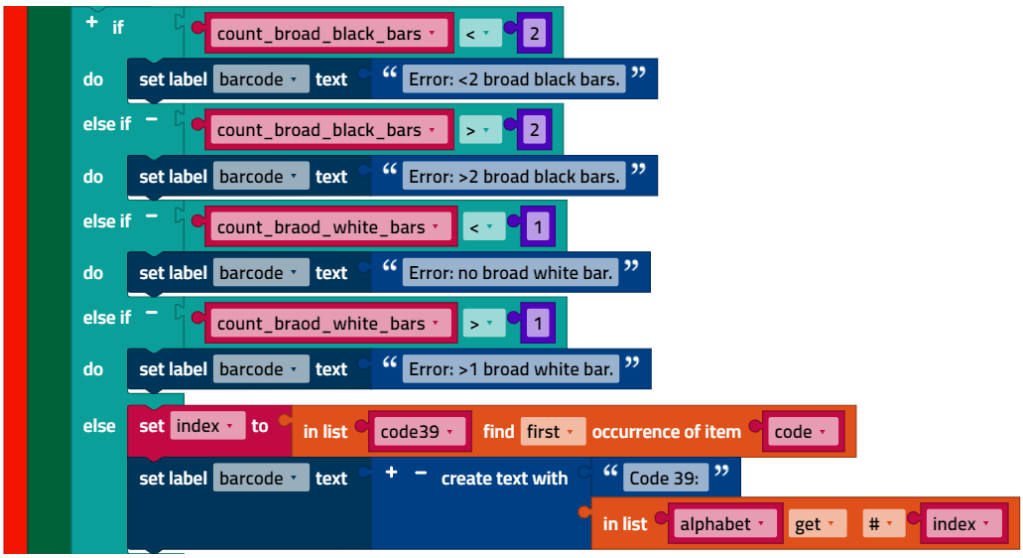
Kodun yüksek yedekliliği onu çok sağlam hale getirir. Geçersiz kod kelimelerini geniş siyah ve geniş beyaz çubukları sayarak tespit etmek kolaydır.

Program özeti (örnek):

…



TASARIM GÖREV ÇÖZÜM

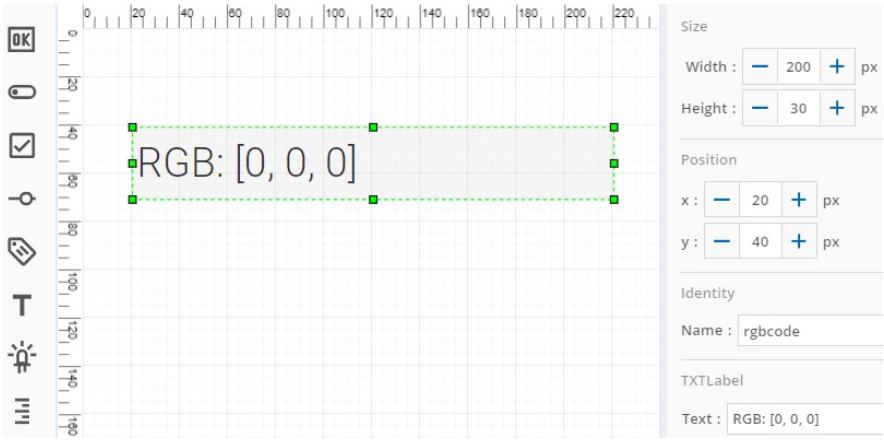


Code39_Decoder_with_Error_Detection.ft

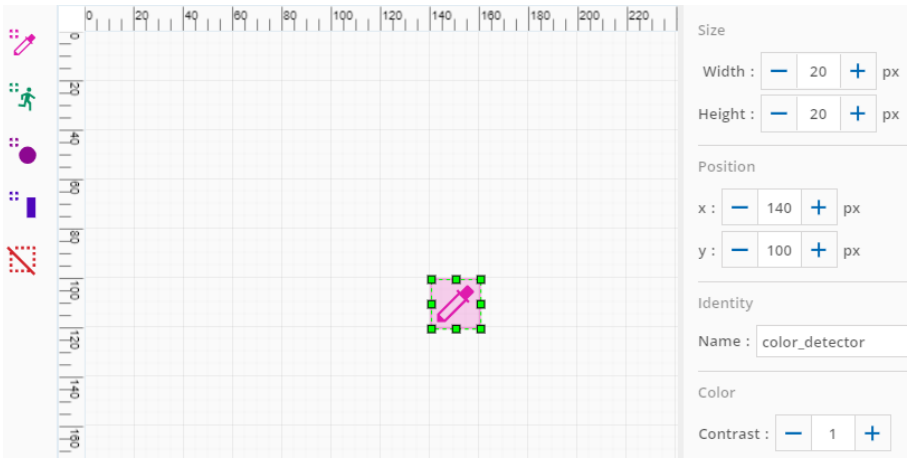
Deneysel görevler (ÇÖZÜM)

1. RGB kodu:

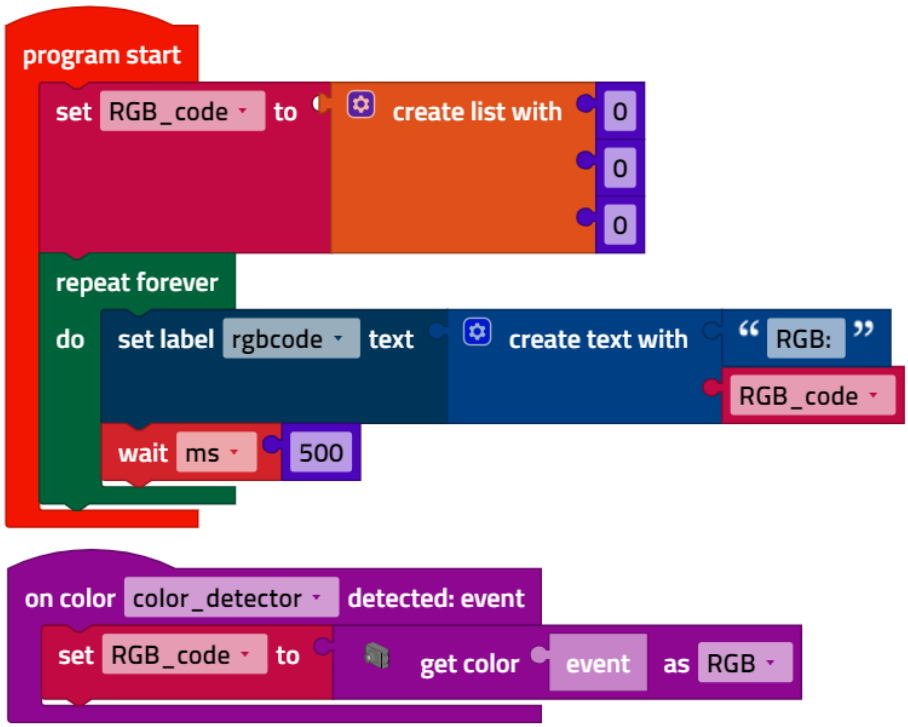
1a. Ekran yapılandırması:



Renk tanıma yapılandırması:

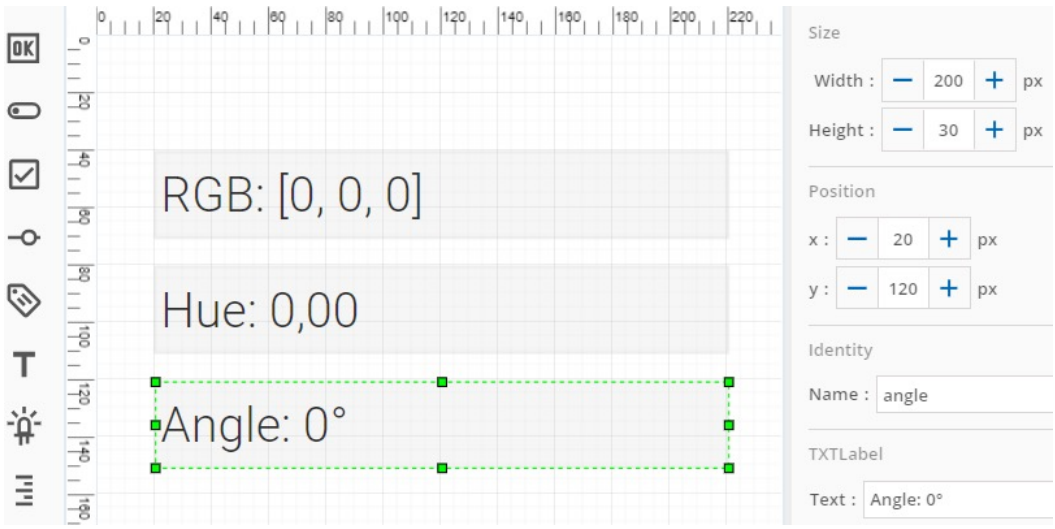


Program (örnek):

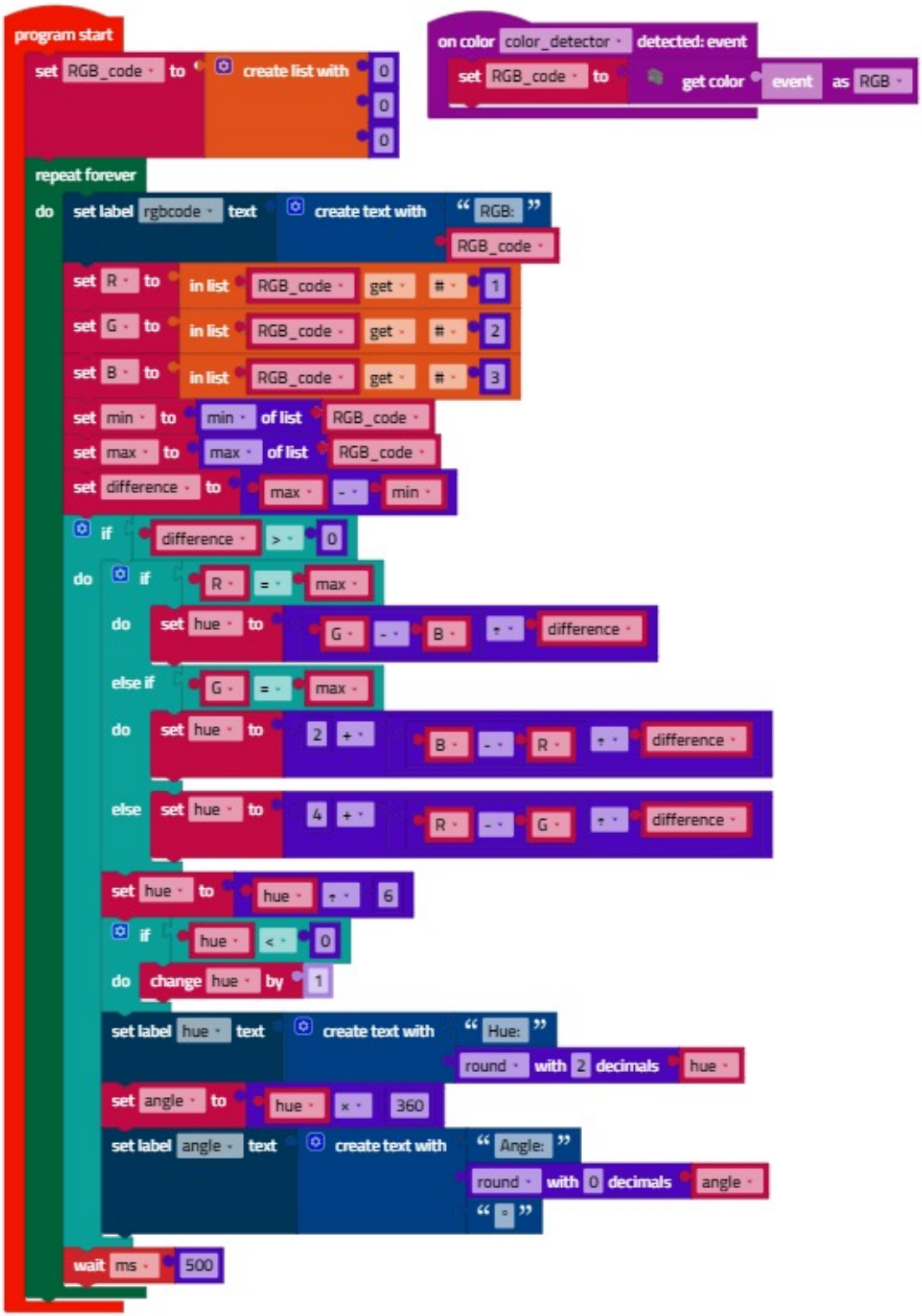


RGB_Colorcode.ft

1b. Ekran yapılandırması:



Program (örnek):



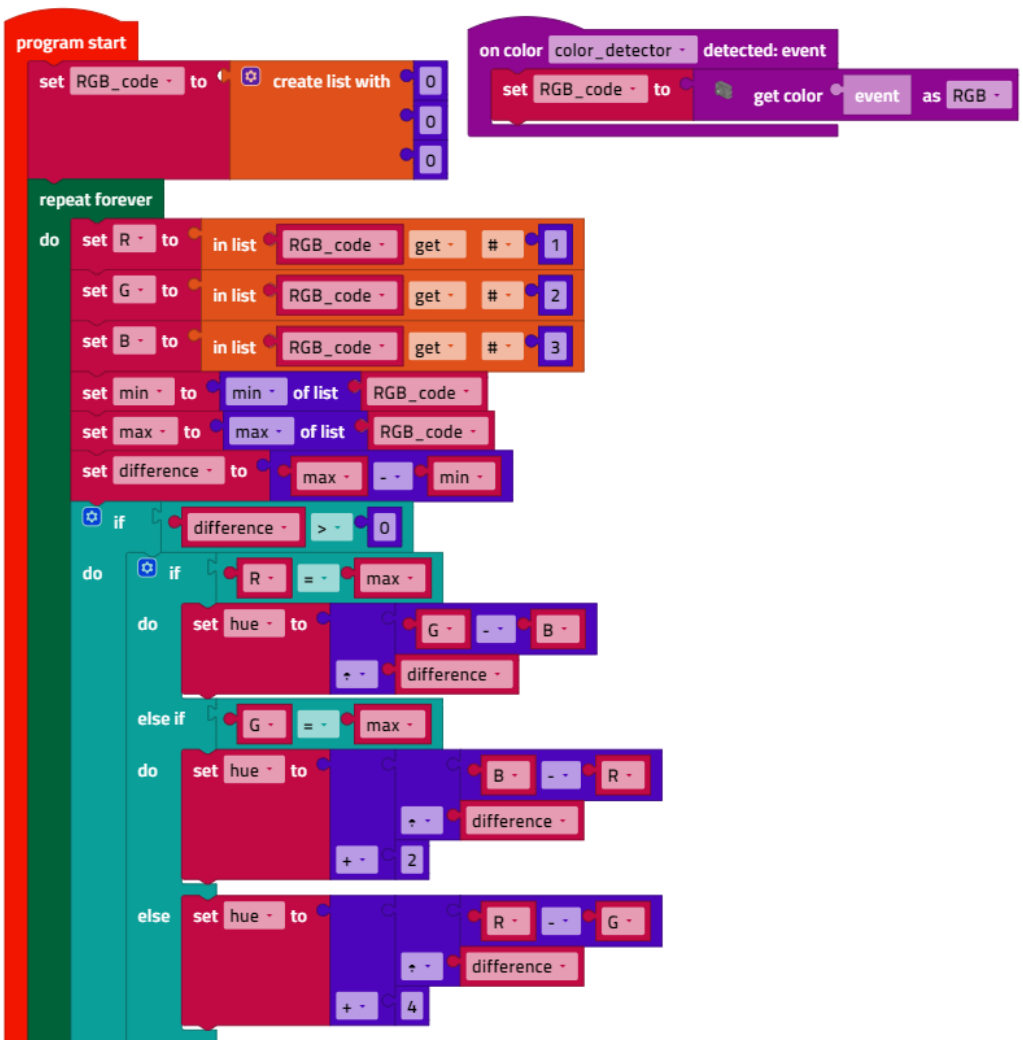
RGB_Decoder.ft

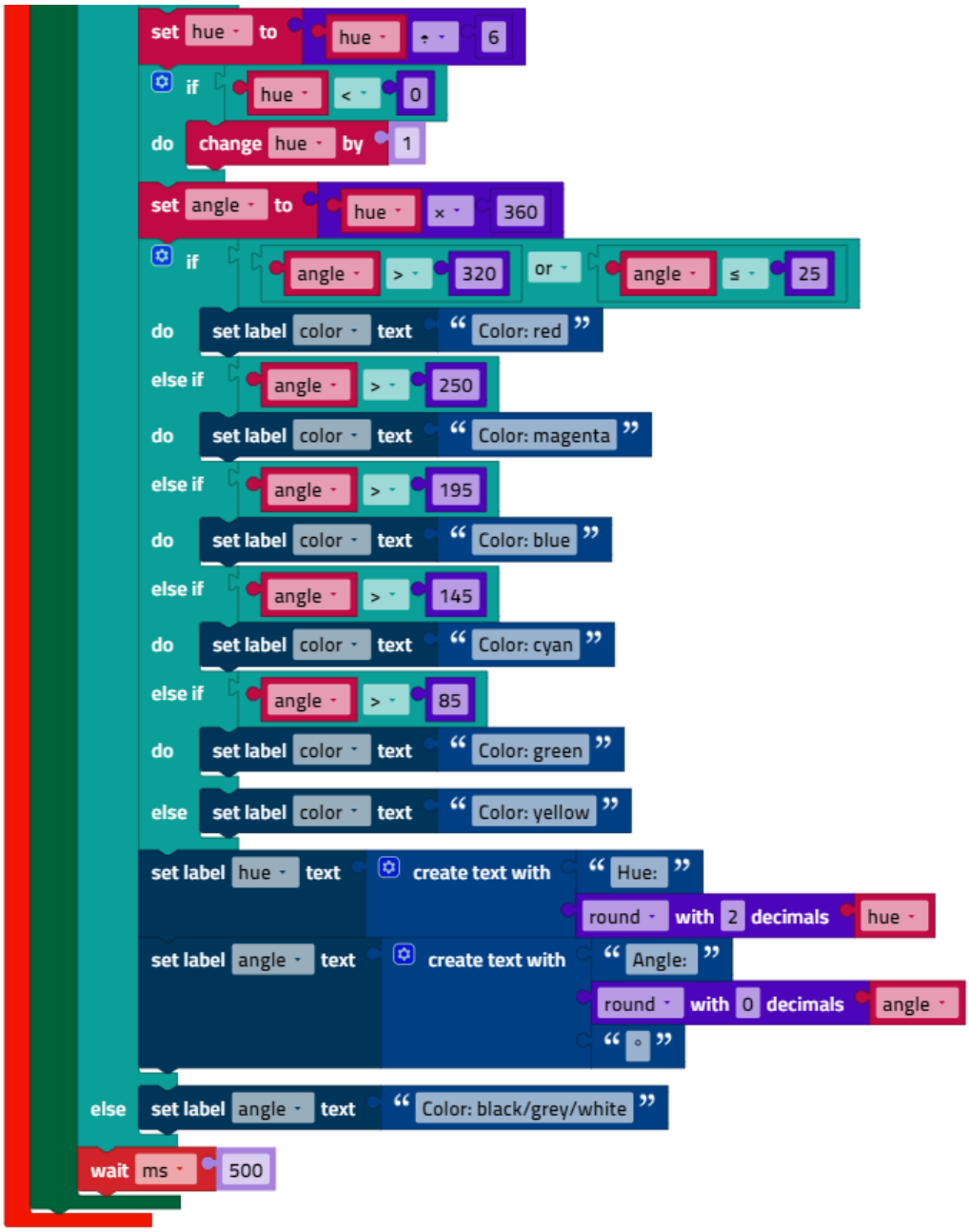
2. Renk tanıma

Ekran yapılandırması:

Program (örnek): Renkli yıldızın renklerini tanıma

…





Camera_Color_Decoder.ft

3. Çok boyutlu kod

Örnek kod (renkli barkod - renk/genişlik, CW)

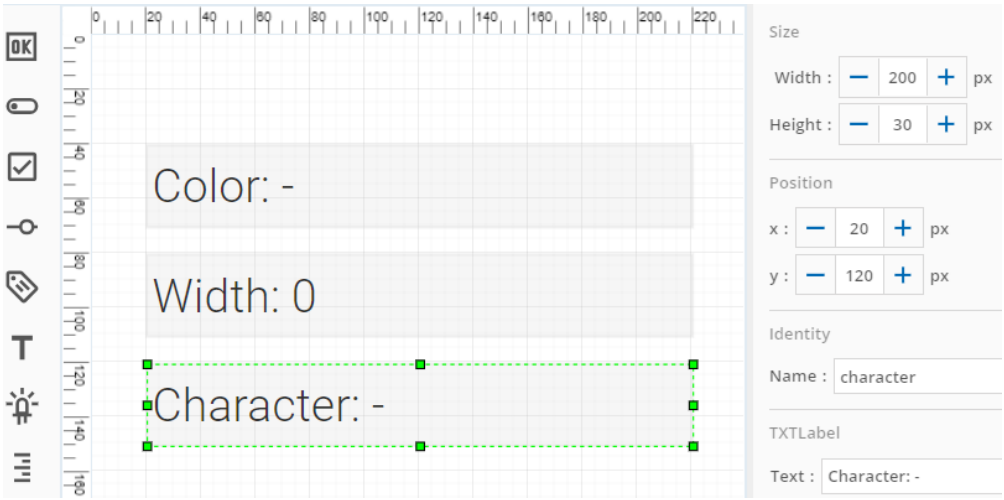
CW kodu altı renk (kırmızı, sarı, yeşil, camgöbeği, mavi, macenta) ve altı çubuk genişliği (1-6) arasında ayrım yapar ve bu nedenle 36 sinyali kodlayabilir:

	0	1	2	3	4	5	6	7	8	9	A	B
Genişlik	1	1	1	1	1	1	2	2	2	2	2	2
Renk	1	2	3	4	5	6	1	2	3	4	5	6
	C	D	E	F	G	H	I	J	K	L	M	N
Genişlik	3	3	3	3	3	3	4	4	4	4	4	4
Renk	1	2	3	4	5	6	1	2	3	4	5	6
	O	P	Q	R	S	T	U	V	W	X	Y	Z
Genişlik	5	5	5	5	5	5	6	6	6	6	6	6
Renk	1	2	3	4	5	6	1	2	3	4	5	6

Altı çubuk genişliği 5 mm'lik artışlarla seçilir: 0,5 cm (= 1), 1 cm (= 2), …, 3 cm (= 6). Barkod okuyucu modelindeki kameranın mesafesinde, 0,5 cm yaklaşık 15 genişlik noktasına karşılık gelir.

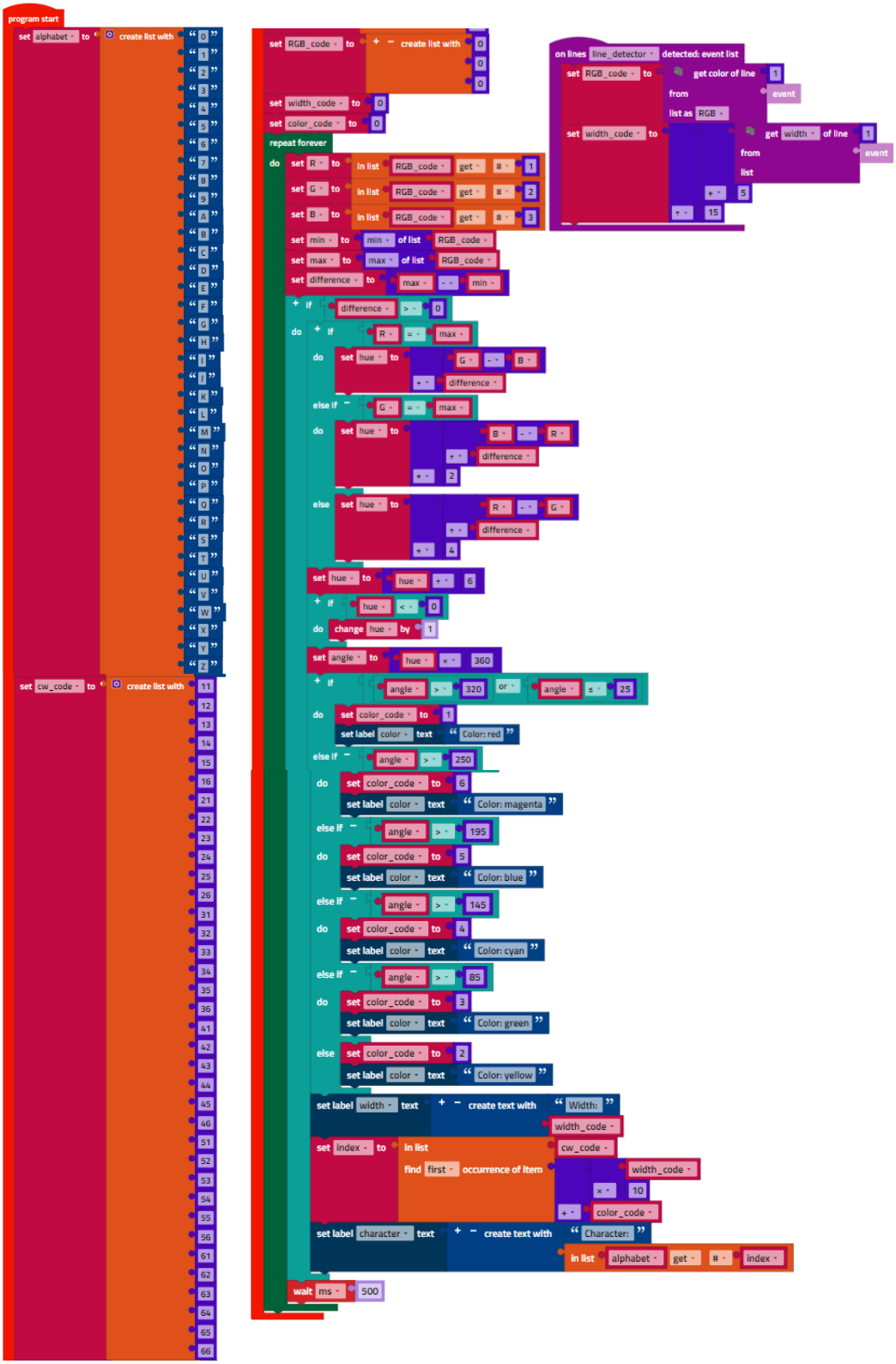
Genişlik noktalarını 15'e bölerek kodlama değerine dönüştürmek kolaydır. Algılamadaki dalgalanmaları telafi etmek için önce beş genişlik noktası eklenir.

Görüntüleme yapılandırması:



Kodlanmış değer, ondalık sayı olarak basitleştirilmiş bir şekilde görüntülenir:
Çubuk genişliği birinciye, renk ise ikinci basamağa karşılık gelir.

Döndürülen değerlerden biri "0" ise, çubuklar tanınamaz.



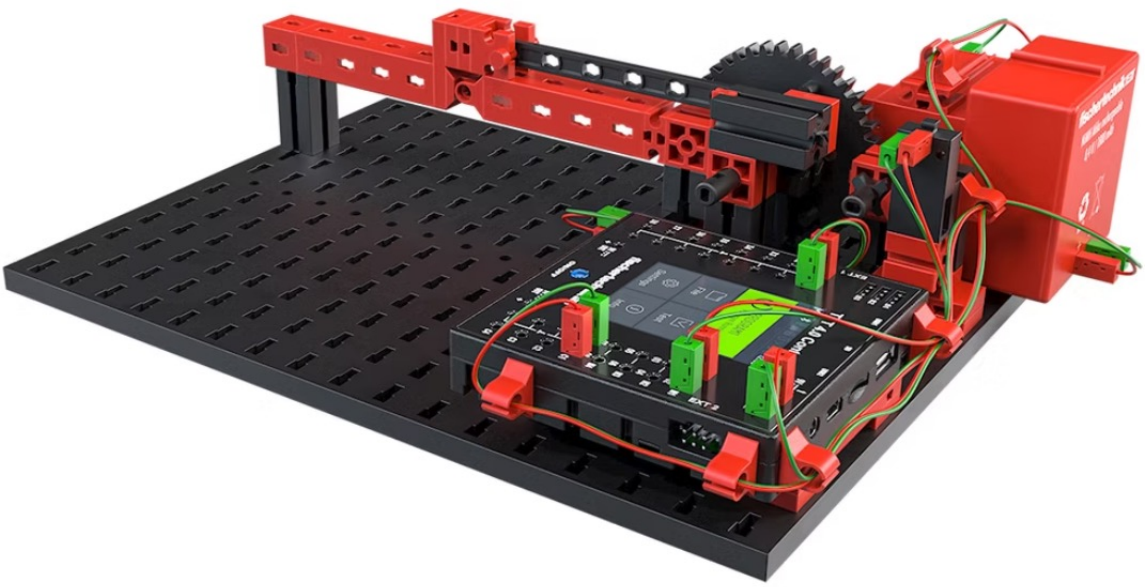
Colored_Barcode_Decoder.ft

ftrobotik

fischertechnik

BÖLÜM 05

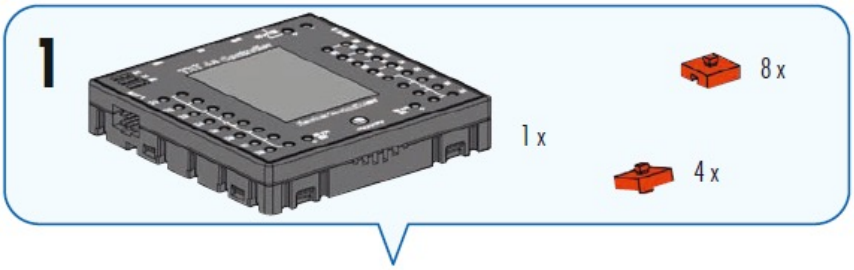
Otopark Bariyer

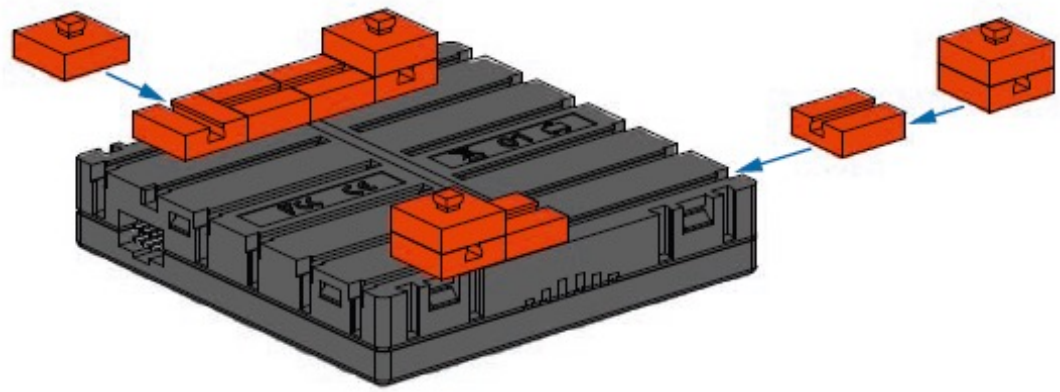


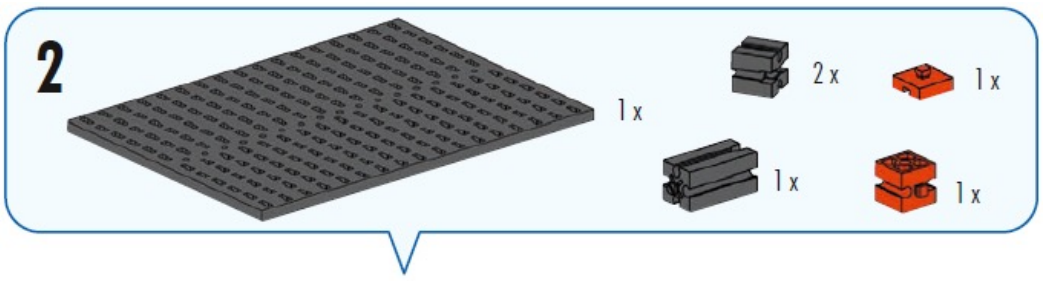
Öğrenme hedefi

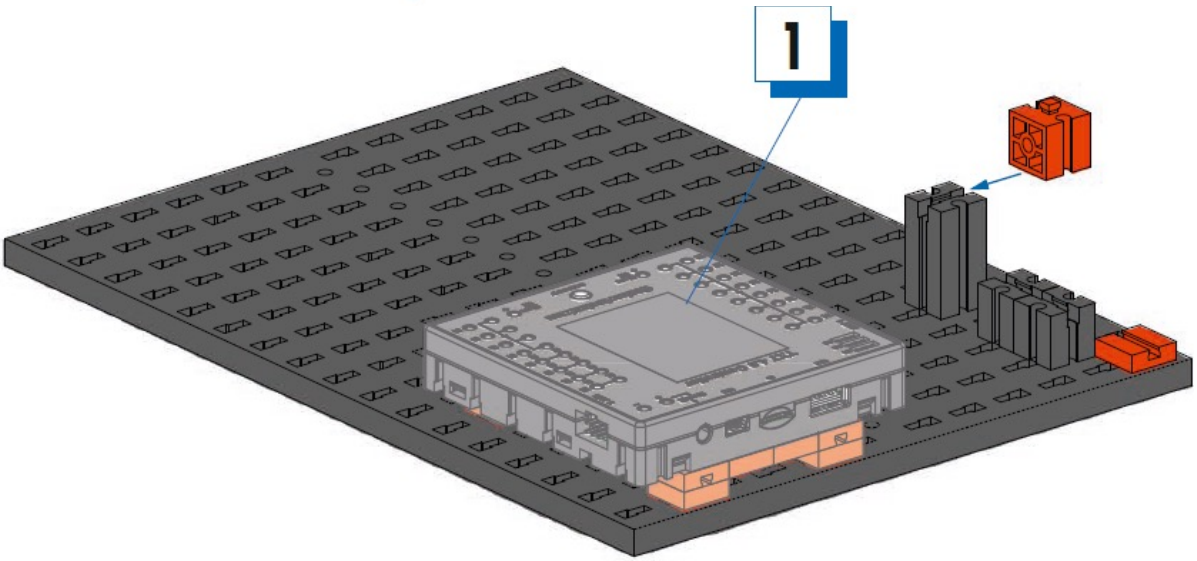
- ✓ Durum diyagramlarının kullanımı (sonlu otomasyonlar)
- ✓ Dijital sensörler ile basit kontrol
- ✓ Farklı dijital sensörlerin ve enkoder (kodlayıcı) motorun işlevi
- ✓ Açısal momentumu kullanarak bir aksı kontrol etmek
- ✓ Metre değişkenleri
- ✓ Paralel süreçler (iş parçacıkları; tamamlayıcı görev)

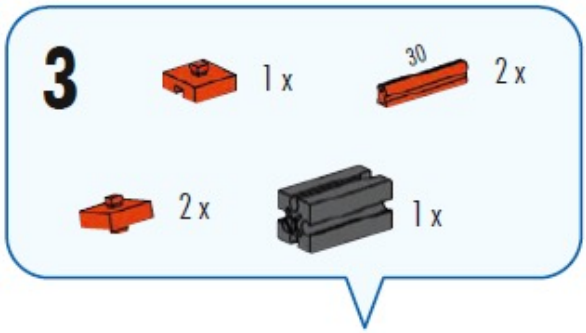
MODEL 5 Otomatik Bariyer

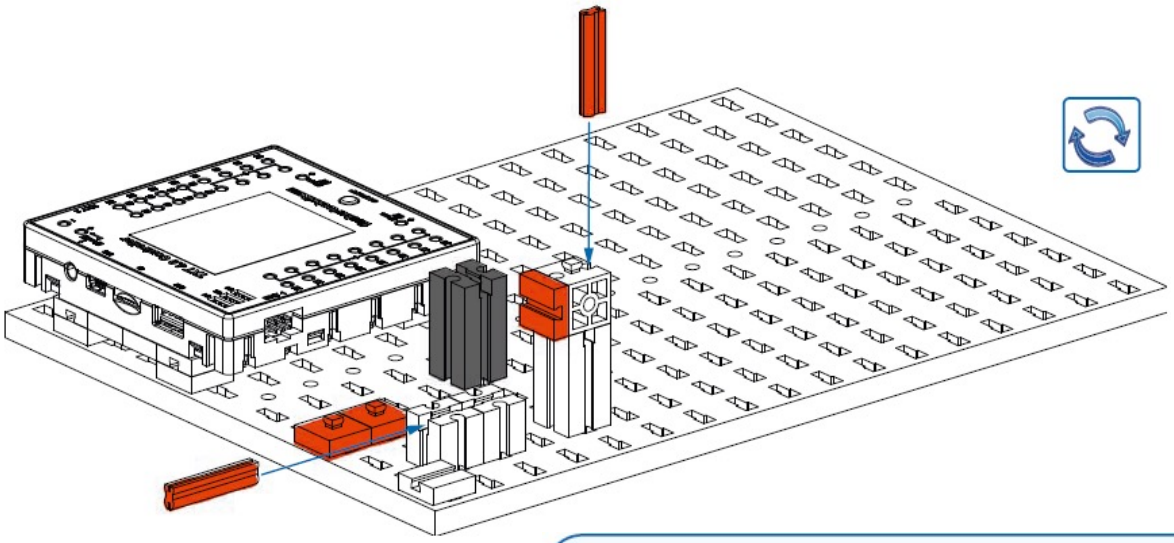


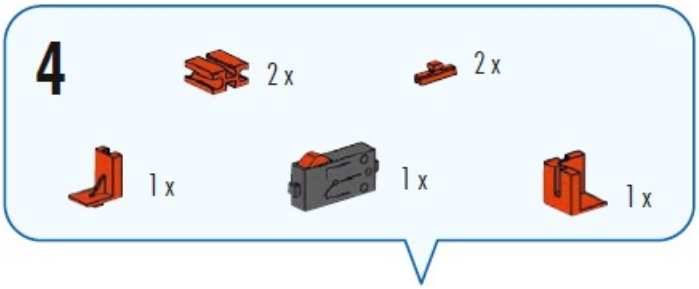


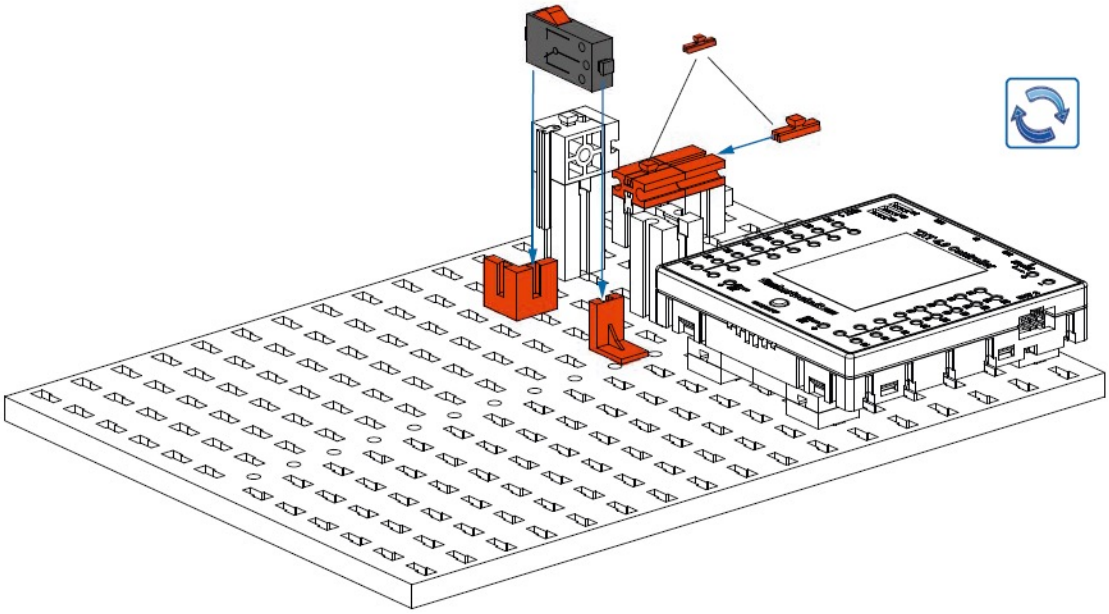


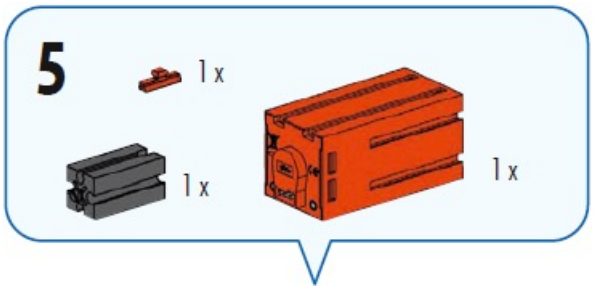


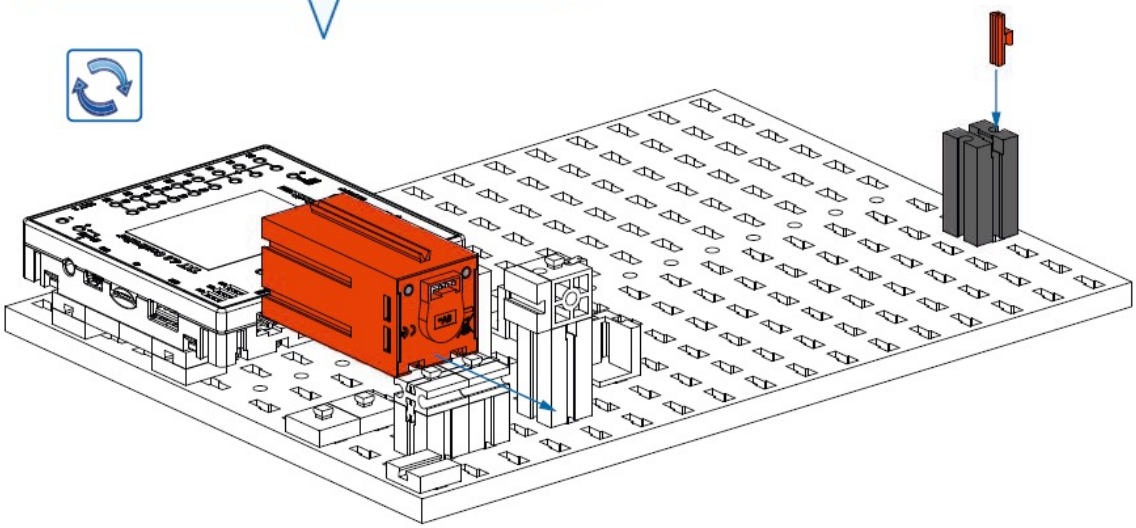

3
1 x
30
2 x
2 x
1 x

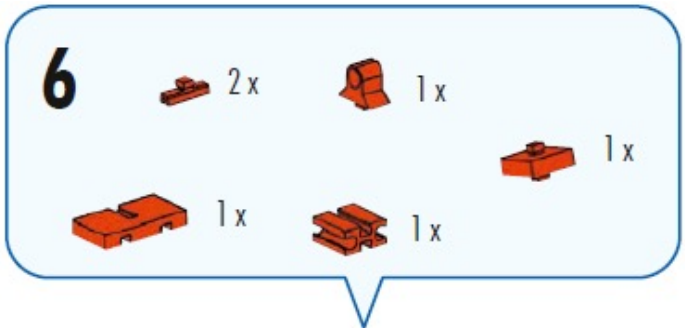


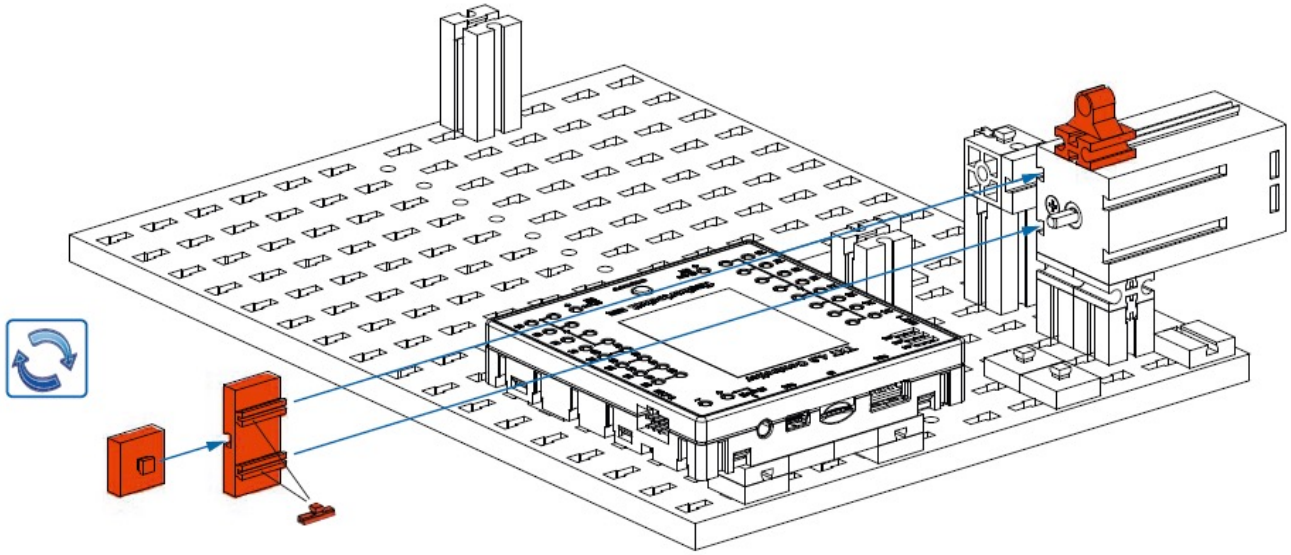

4
2 x
2 x
1 x
1 x
1 x

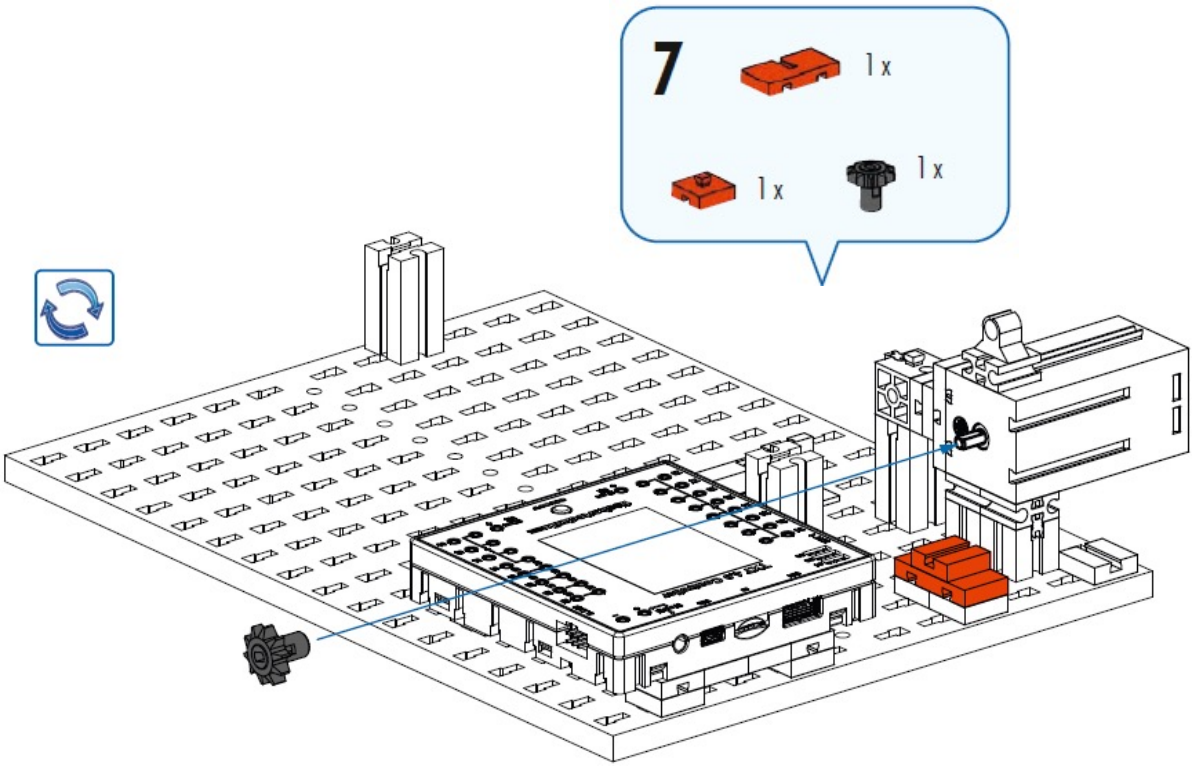


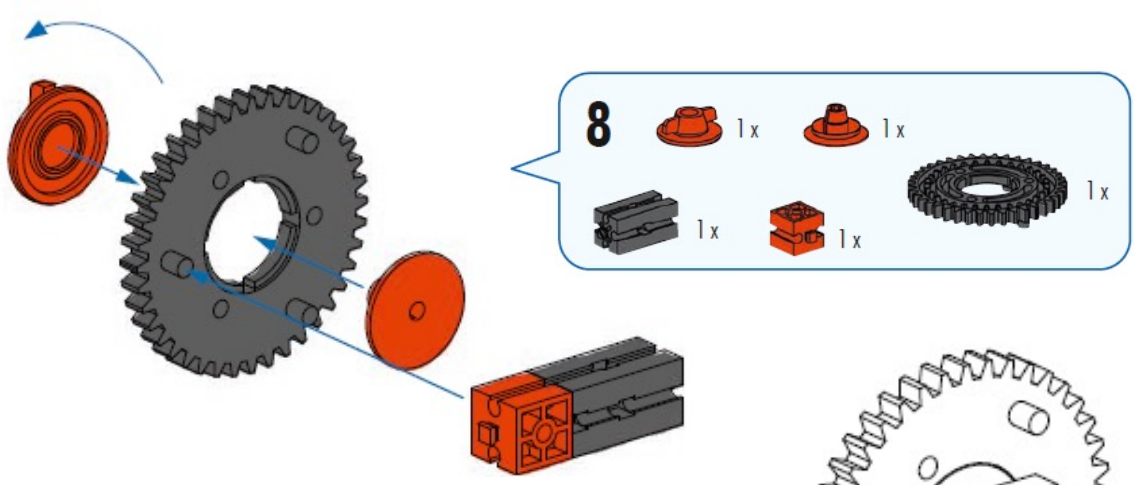

5
1 x
1 x
1 x

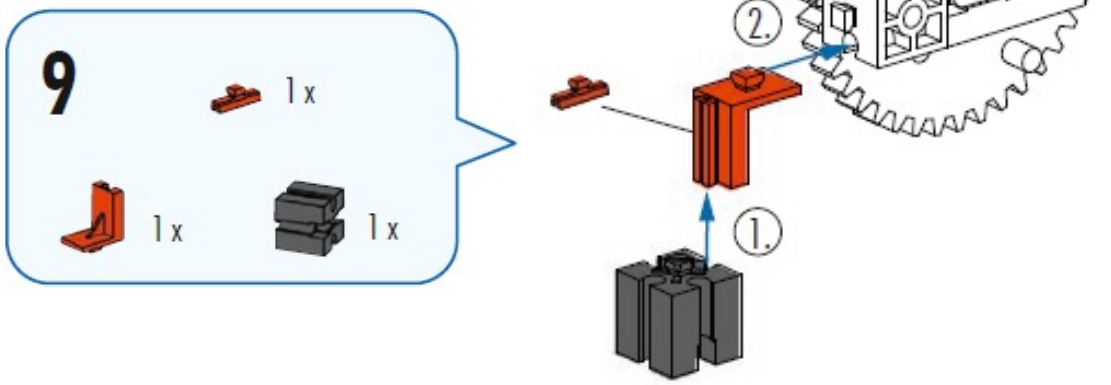


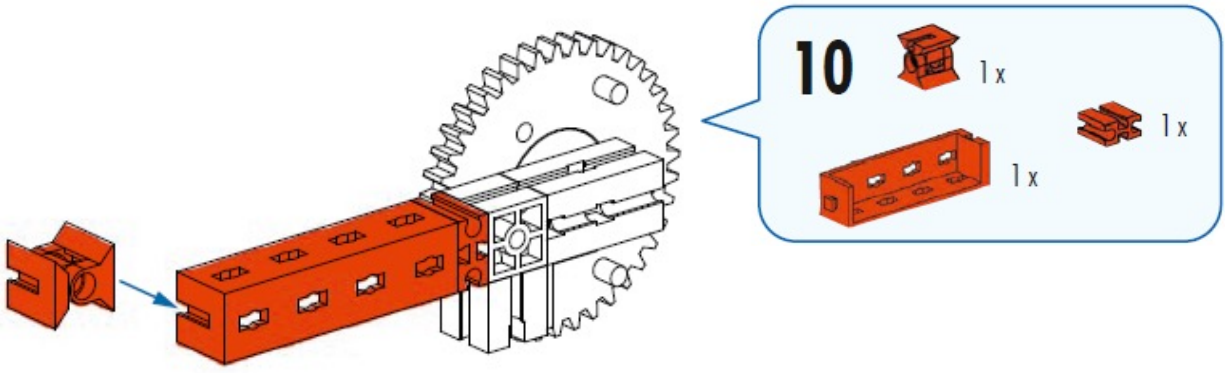

6
2 x
1 x
1 x
1 x
1 x

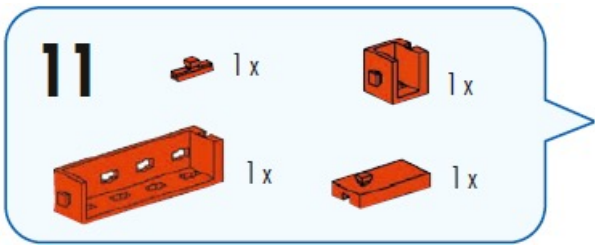


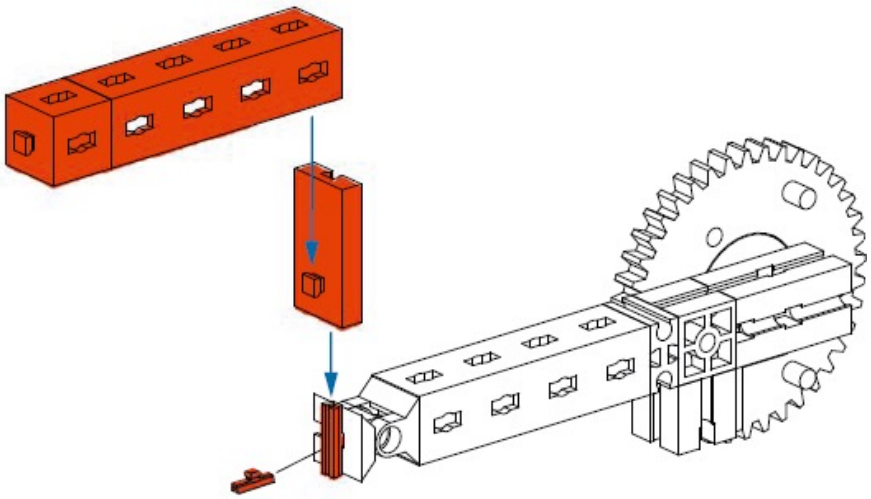

7
1 x
1 x
1 x

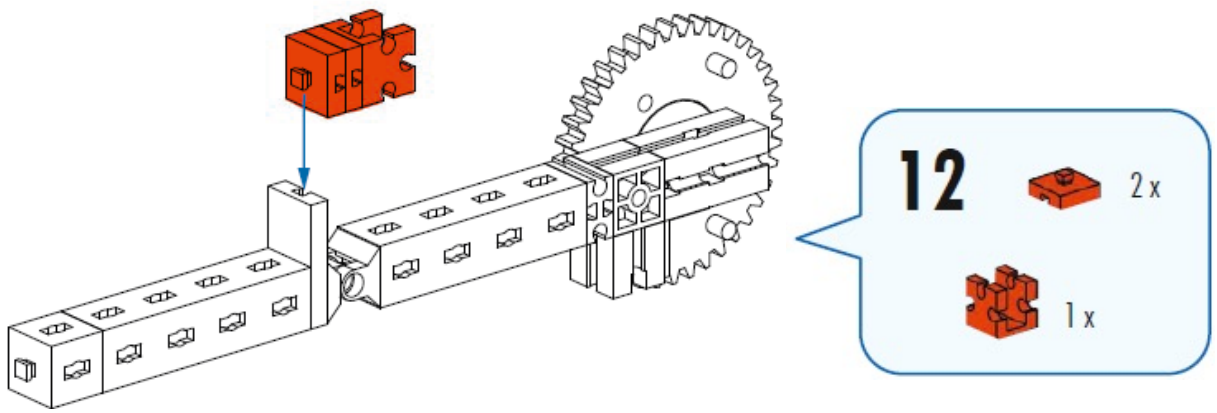

8
1 x
1 x
1 x
1 x
1 x

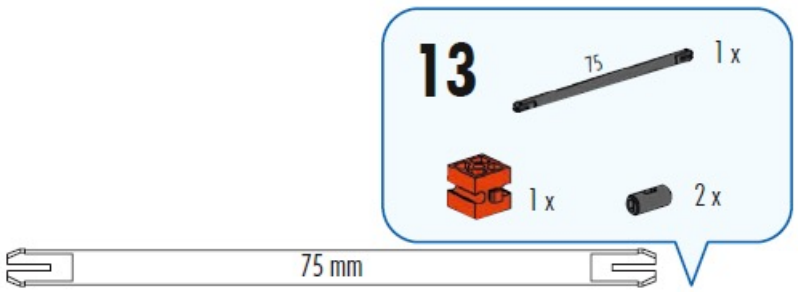

9
1 x
1 x
1 x
2.
1.

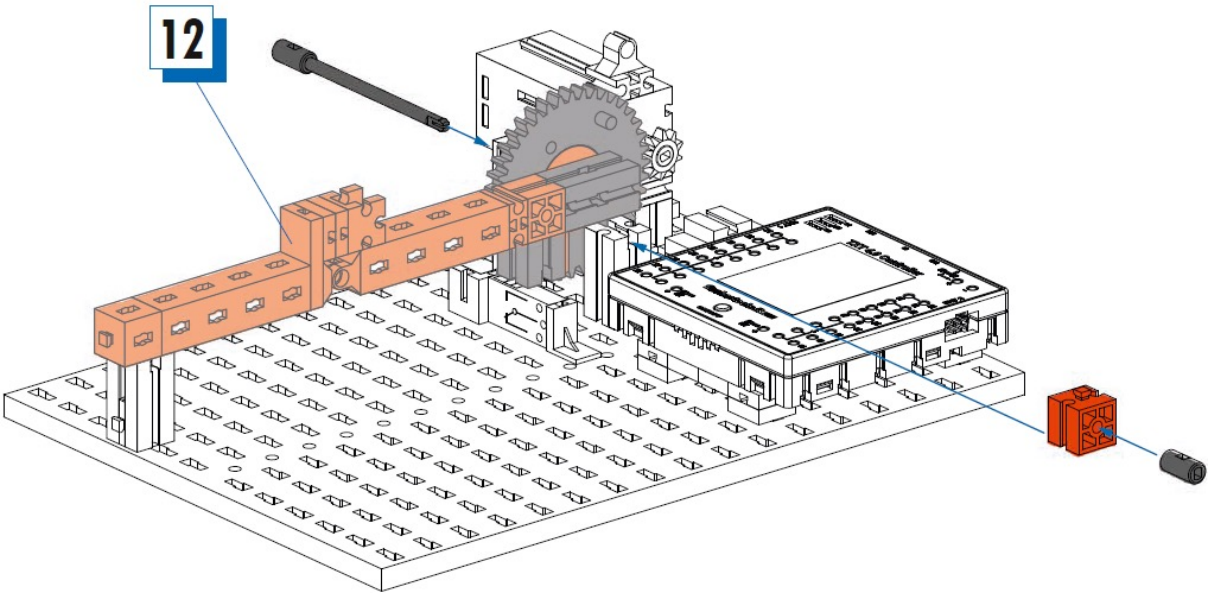

10
1 x
1 x
1 x

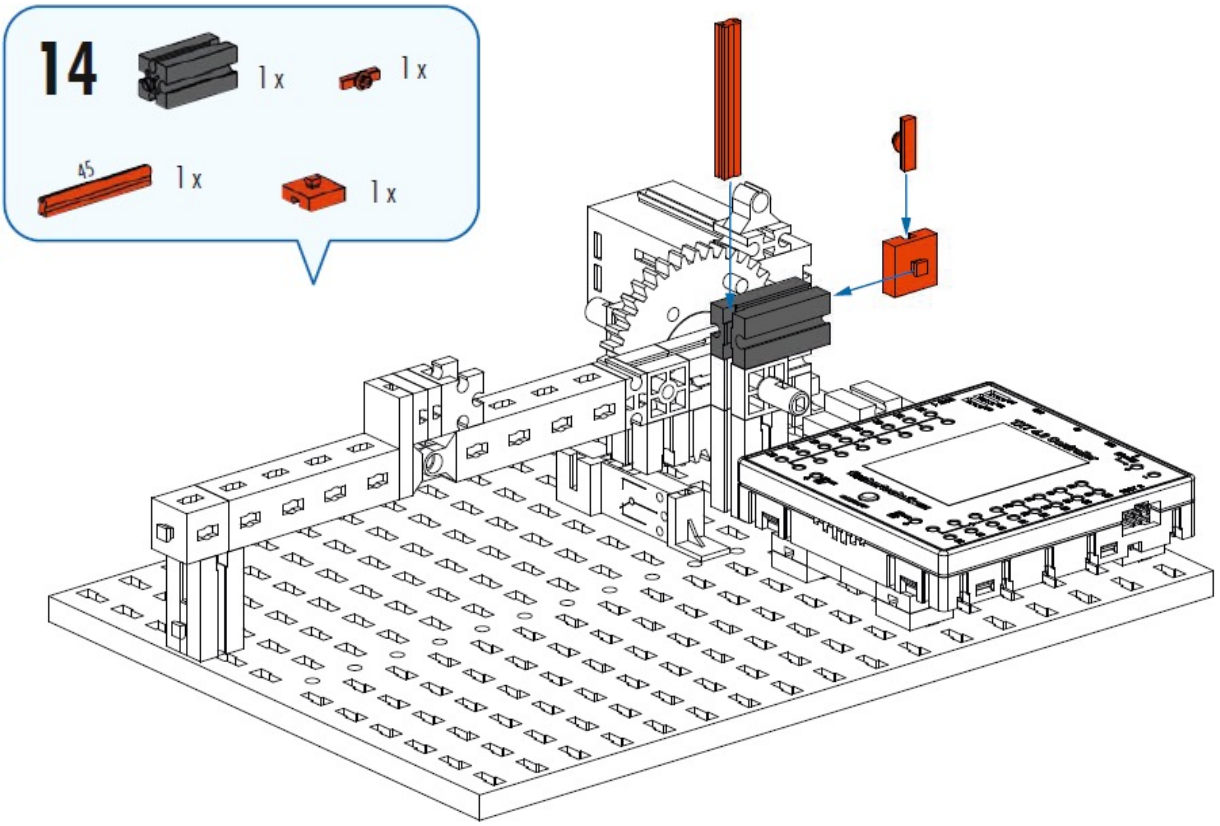

11
1 x
1 x
1 x
1 x

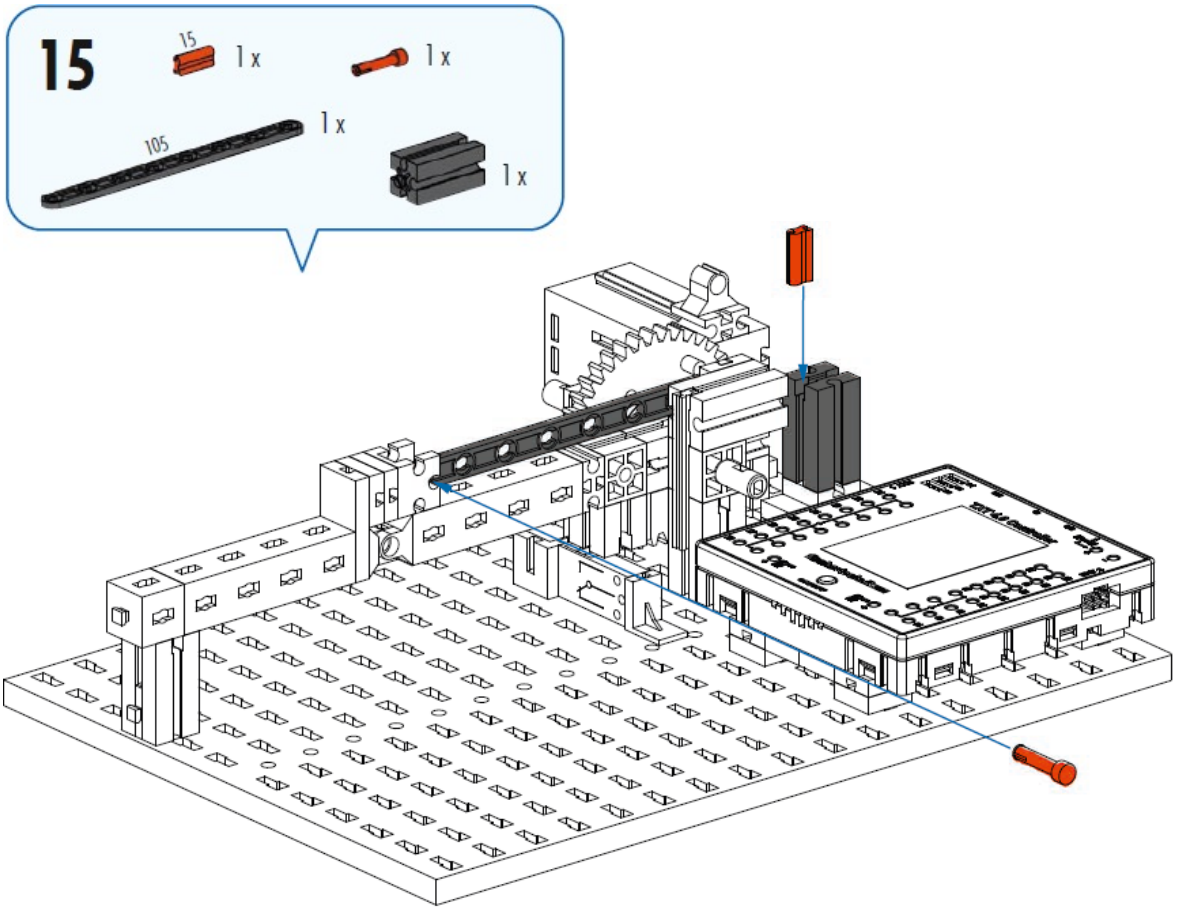


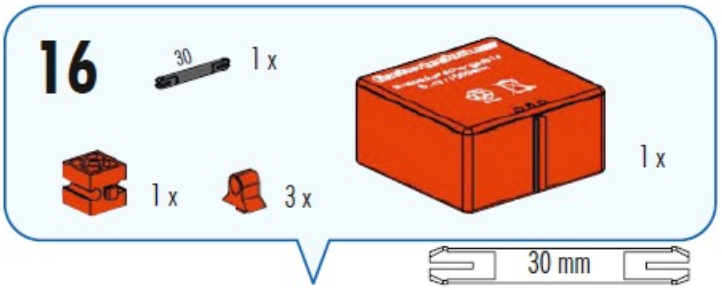

12
2 x
1 x

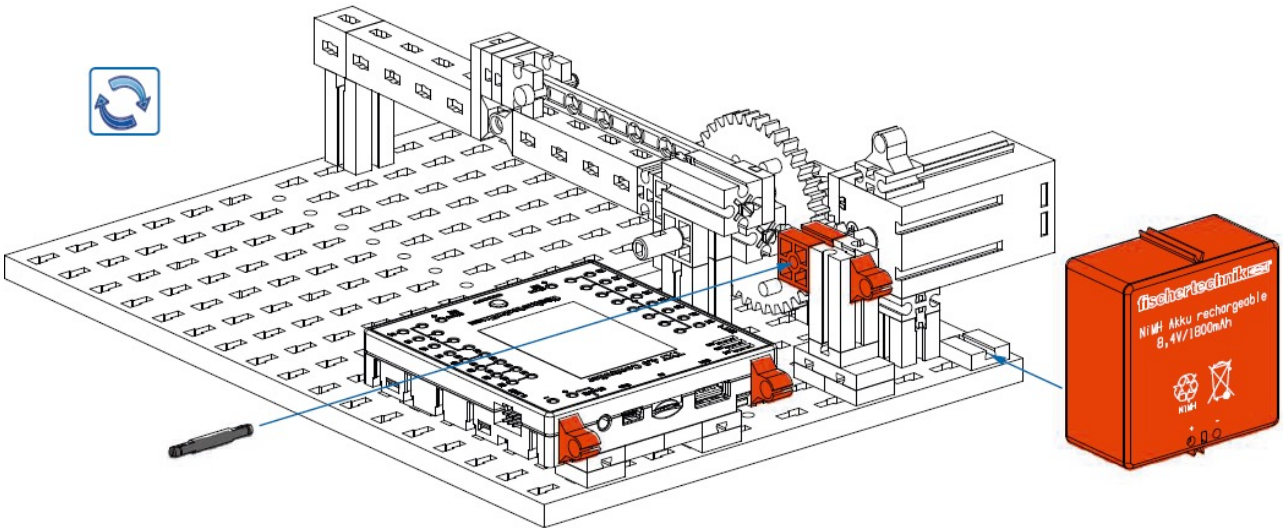

13
75
1 x
1 x
2 x
75 mm

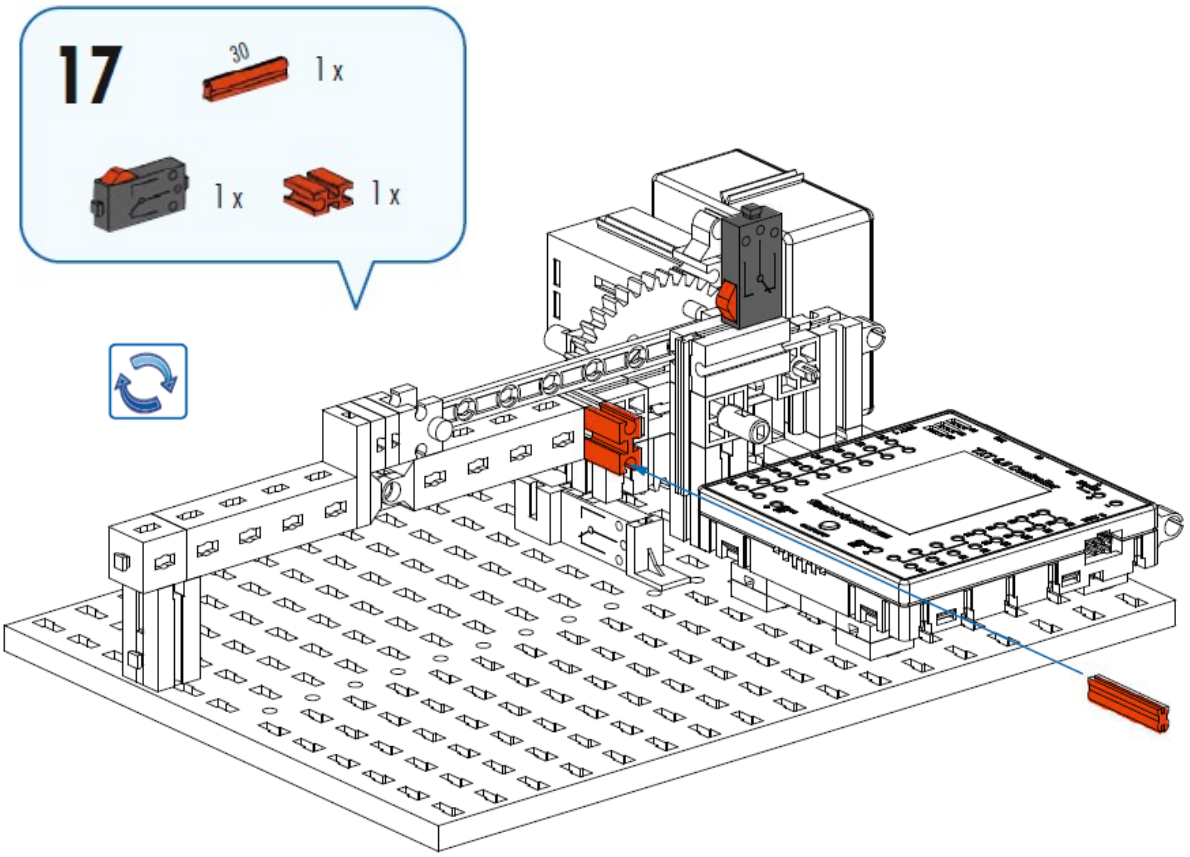

12

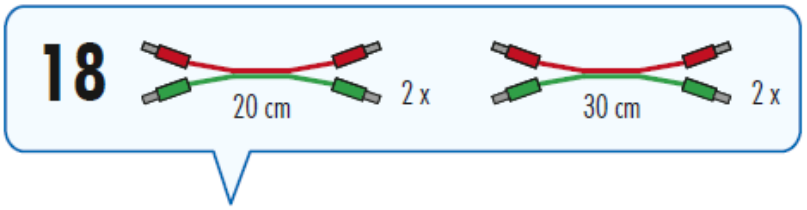

14
1 x
1 x
45
1 x
1 x

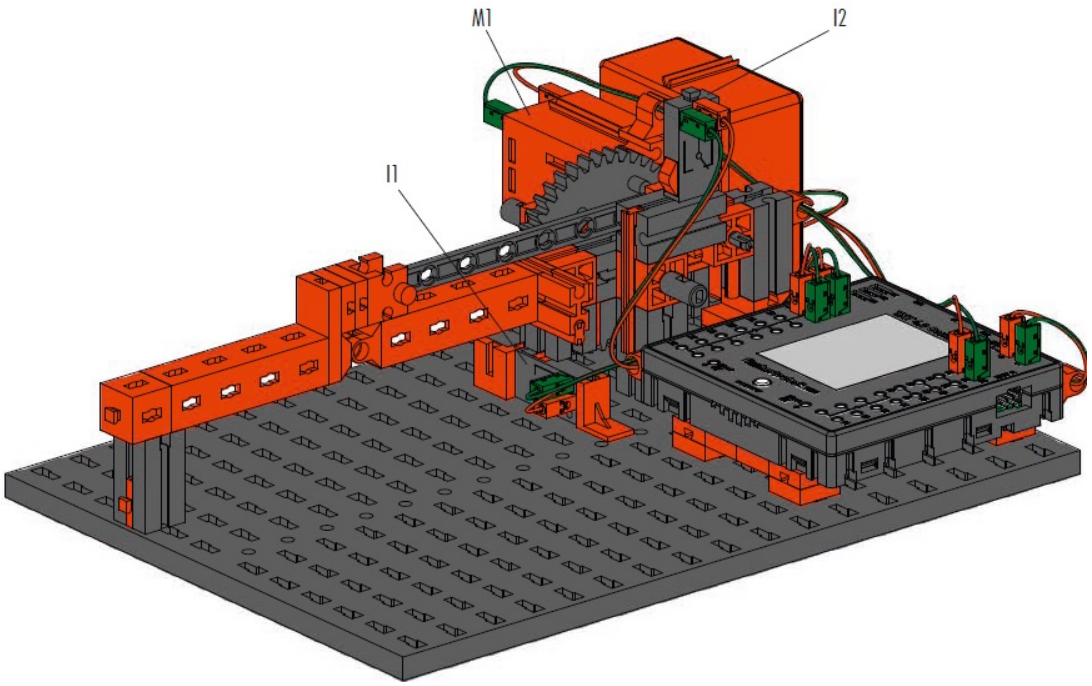

15
15
1 x
1 x
105
1 x
1 x


16
30
1 x
1 x
3 x
1 x
30 mm

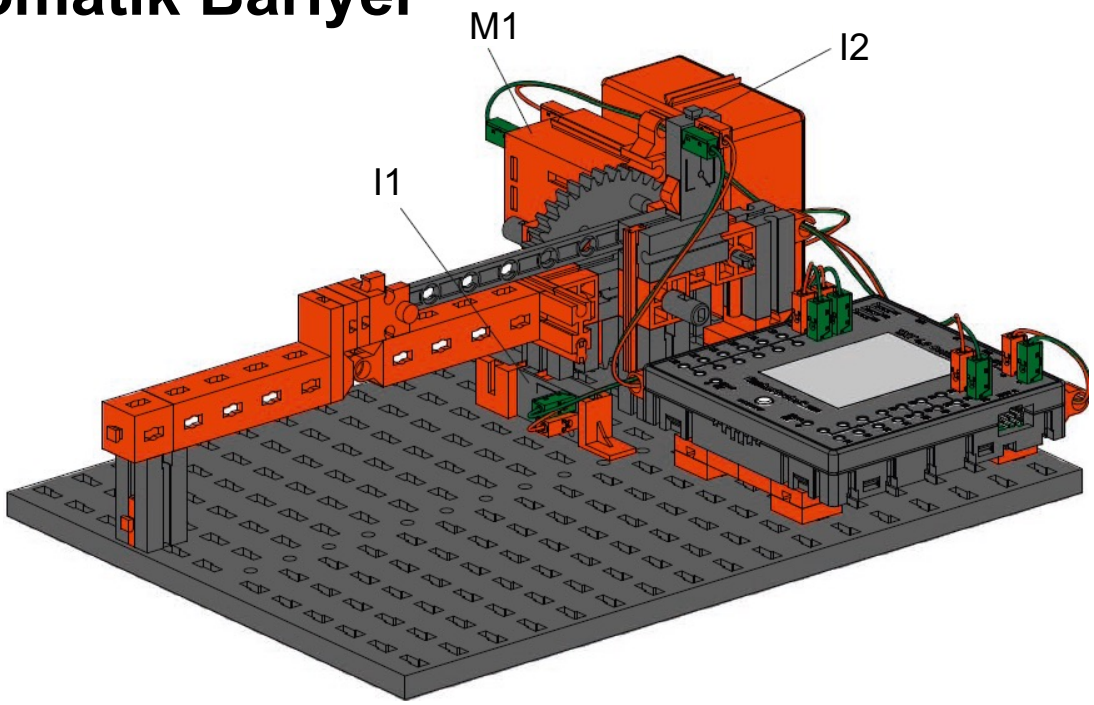

fischertechnik
NiMH Akku rechargeable
8,4V/1900mAh


17
30
1 x
1 x
1 x

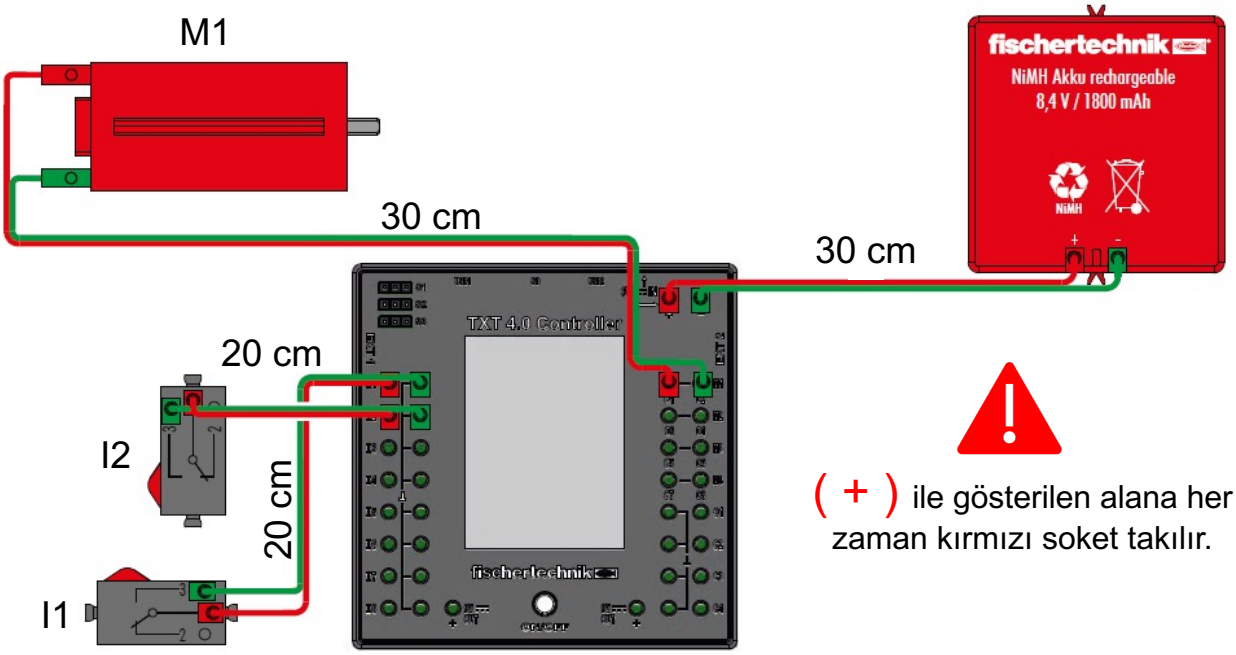

18
20 cm
2 x
30 cm
2 x


M1
I2
I1

Model - 5
Otomatik Bariyer



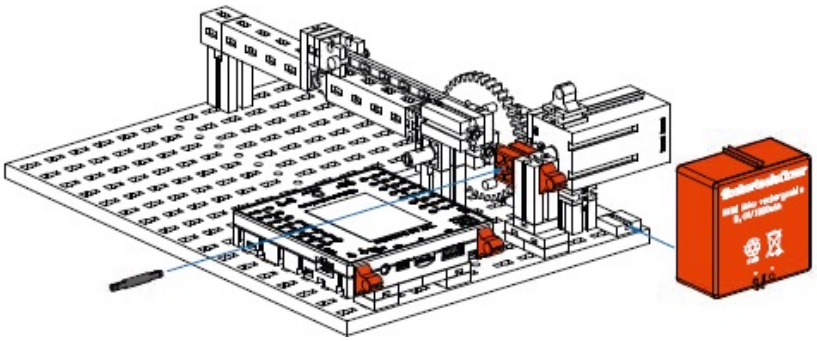
Devre Şeması

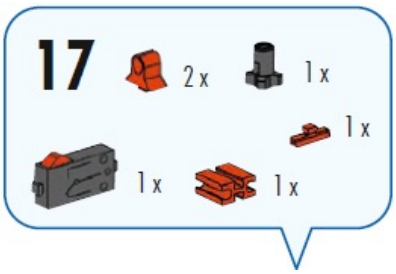


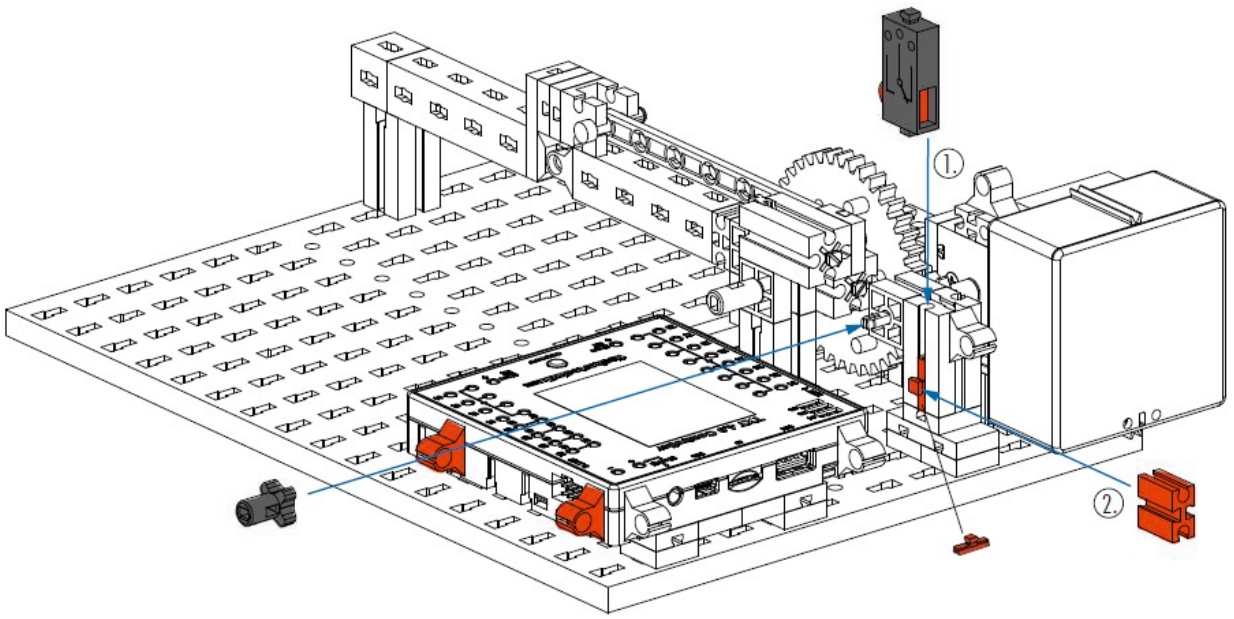
(+) ile gösterilen alana her zaman kırmızı soket takılır.

Model - 5
Otomatik Bariyer Versiyon 2

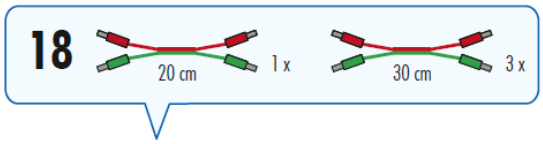
16. Kurulum adımı üzerinden devam edelim

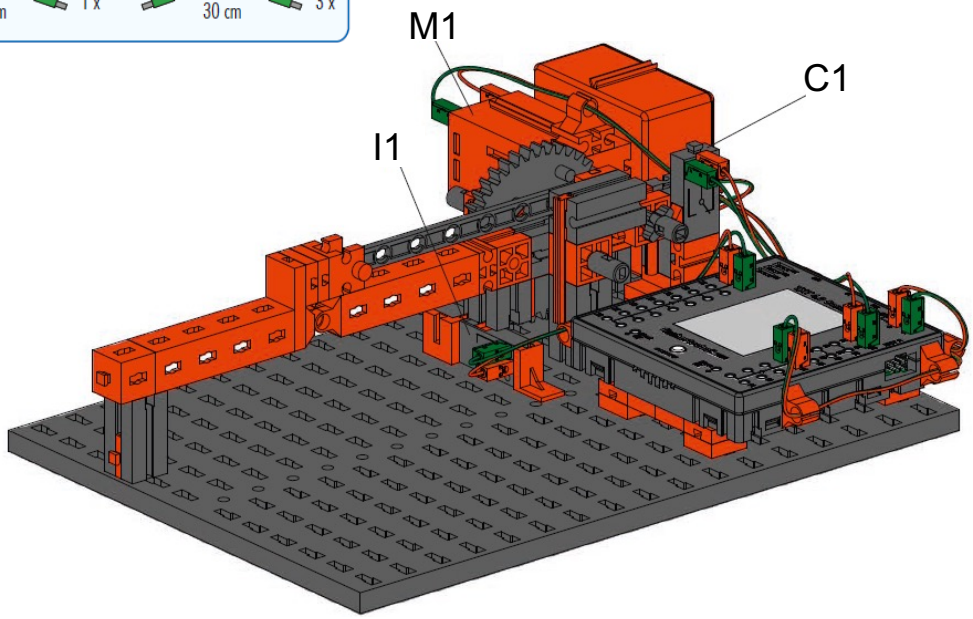




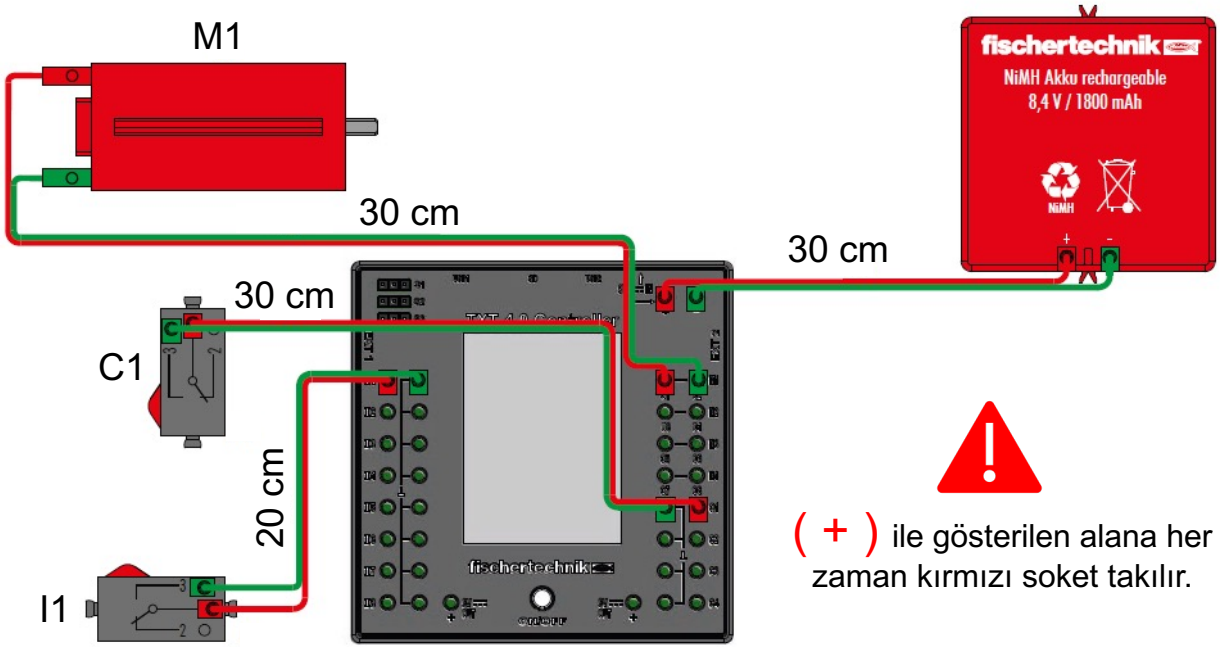


Model - 5
Otomatik Bariyer Versiyon 2



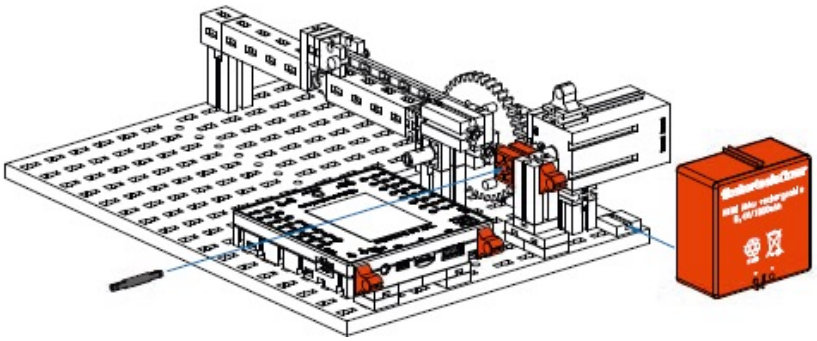


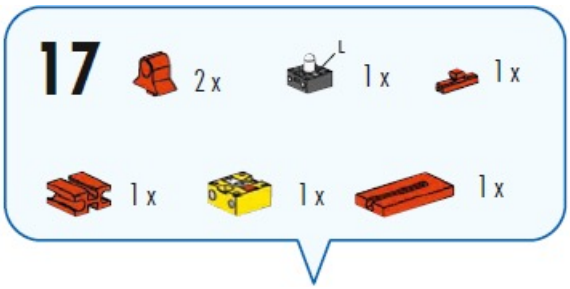
Devre Şeması

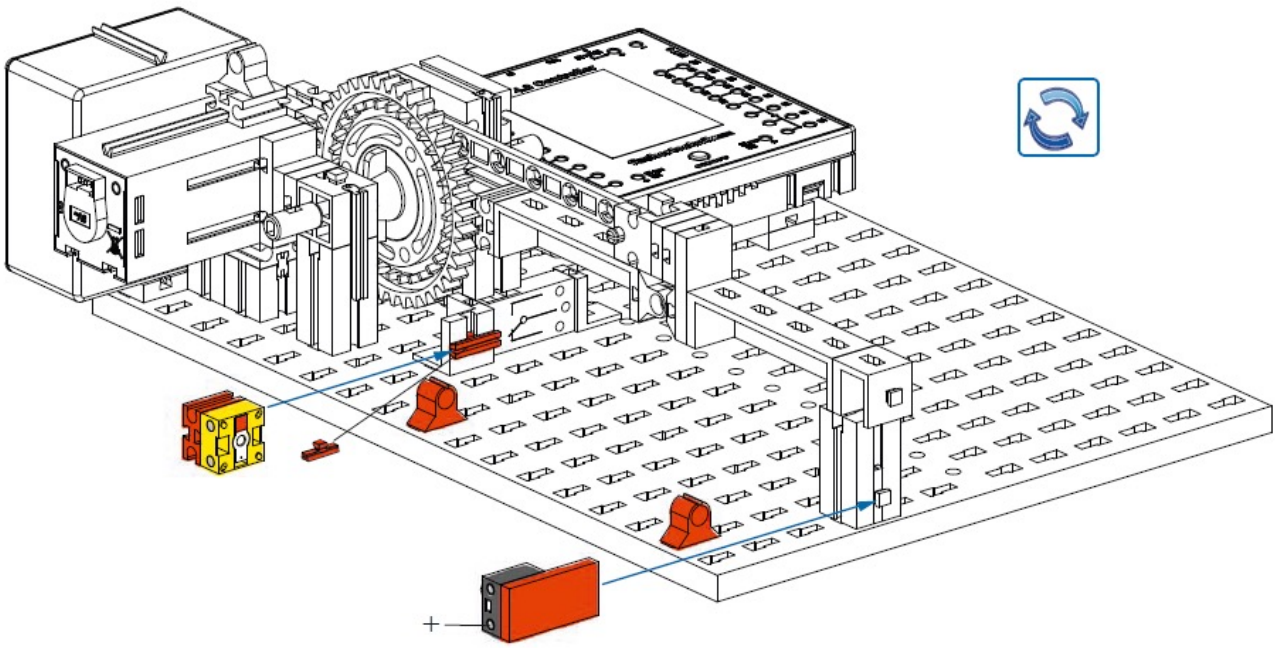


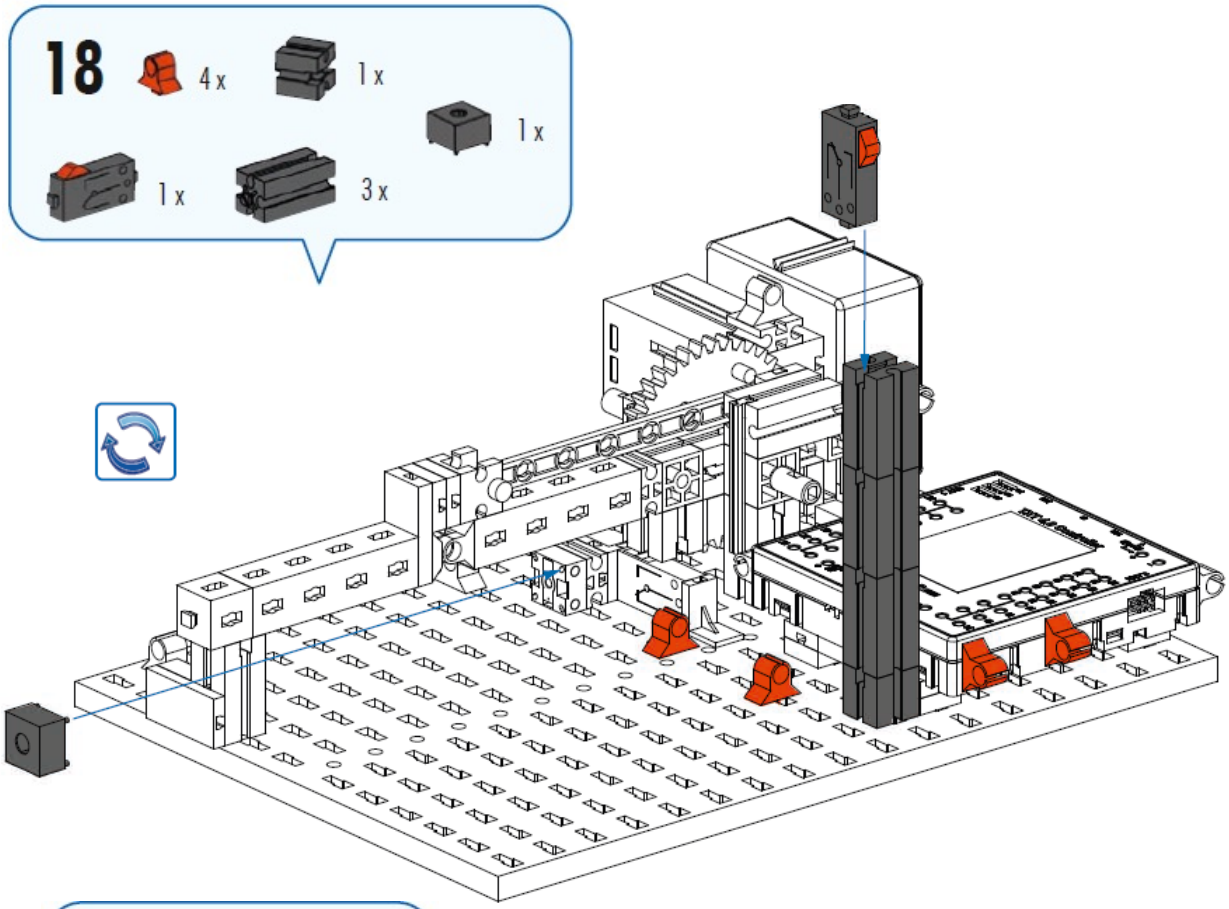
Model - 5
Otomatik Bariyer Versiyon 3

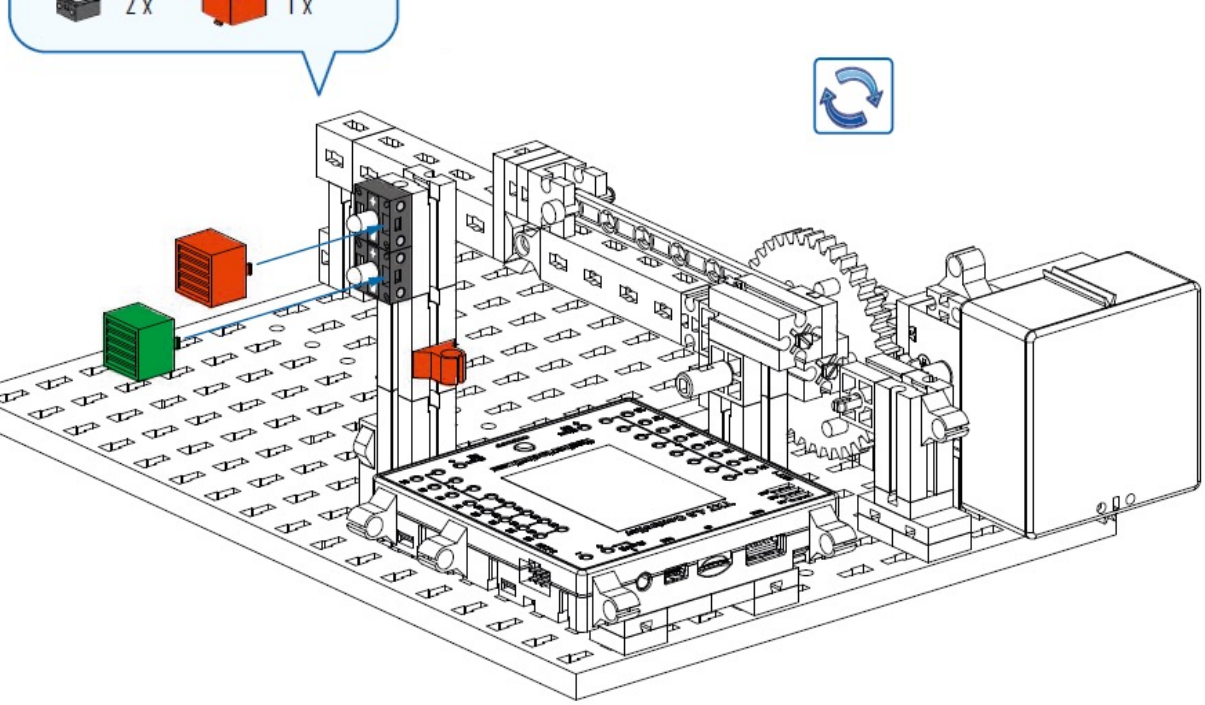
16. Kurulum adımı üzerinden devam edelim



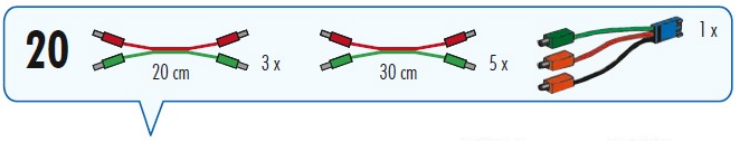


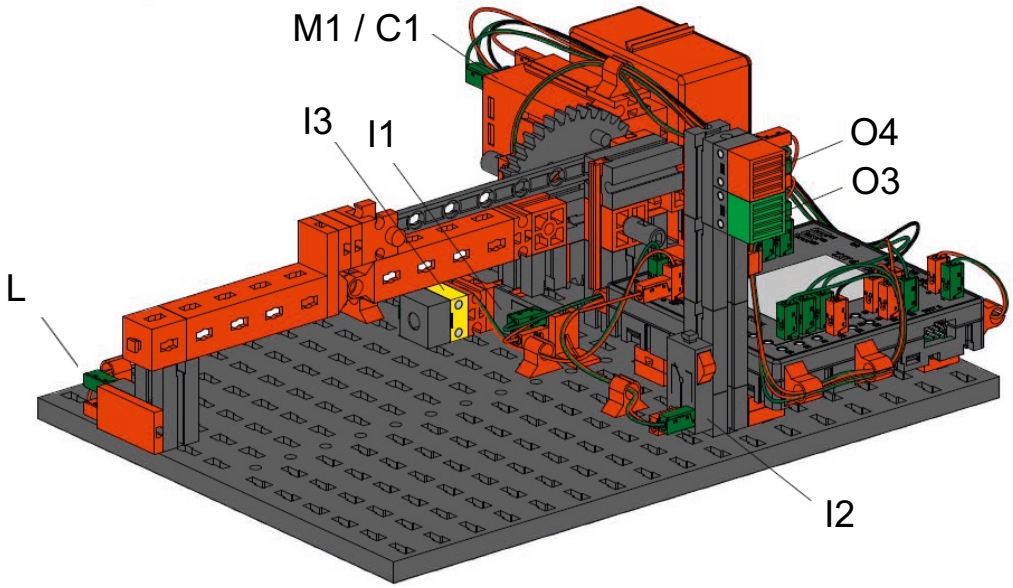



18
4 x
1 x
1 x
1 x
3 x

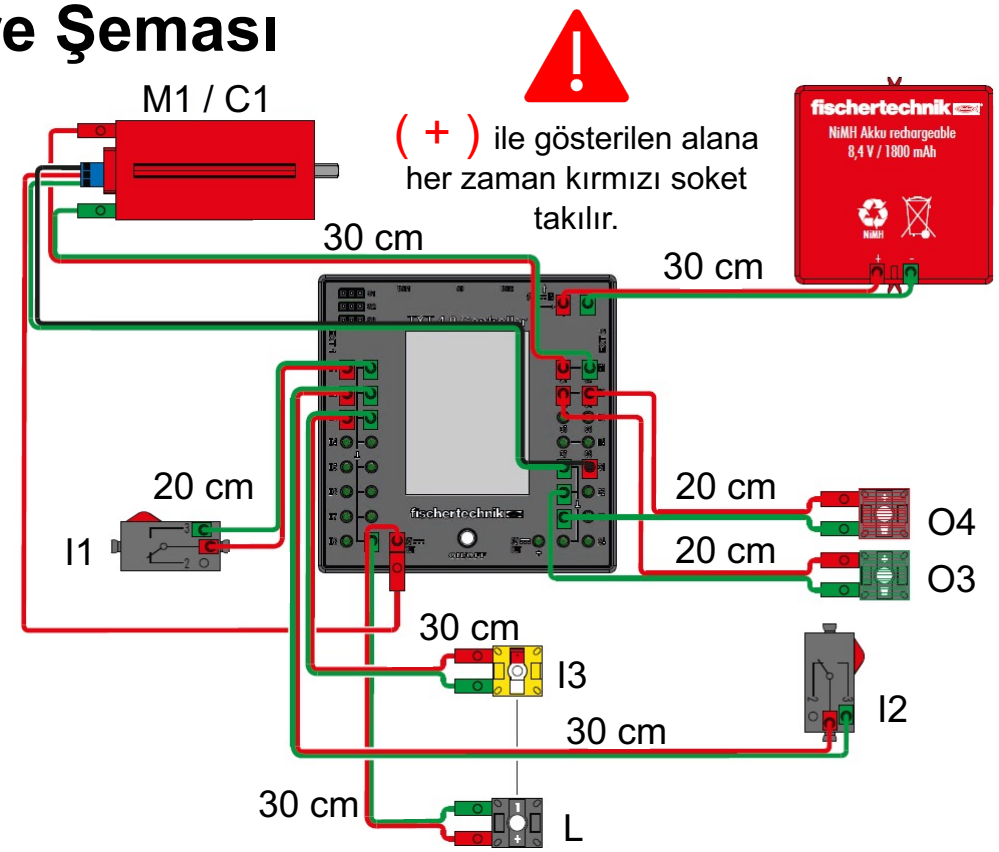

19
1 x
1 x
2 x
1 x

Model - 5
Otomatik Bariyer Versiyon 3





Devre Şeması



Otomatik Bariyer Öğrenme Hedefleri

Konu

Kapalı otopark bariyeri üç aşamalı olarak inşa edilmiştir; konum sensörleri, bir itme tekerleği ve bariyer konumunu kontrol eden bir kodlayıcı motor ile Ayrıca bir ışık bariyeri ve bir sayaç vardır.

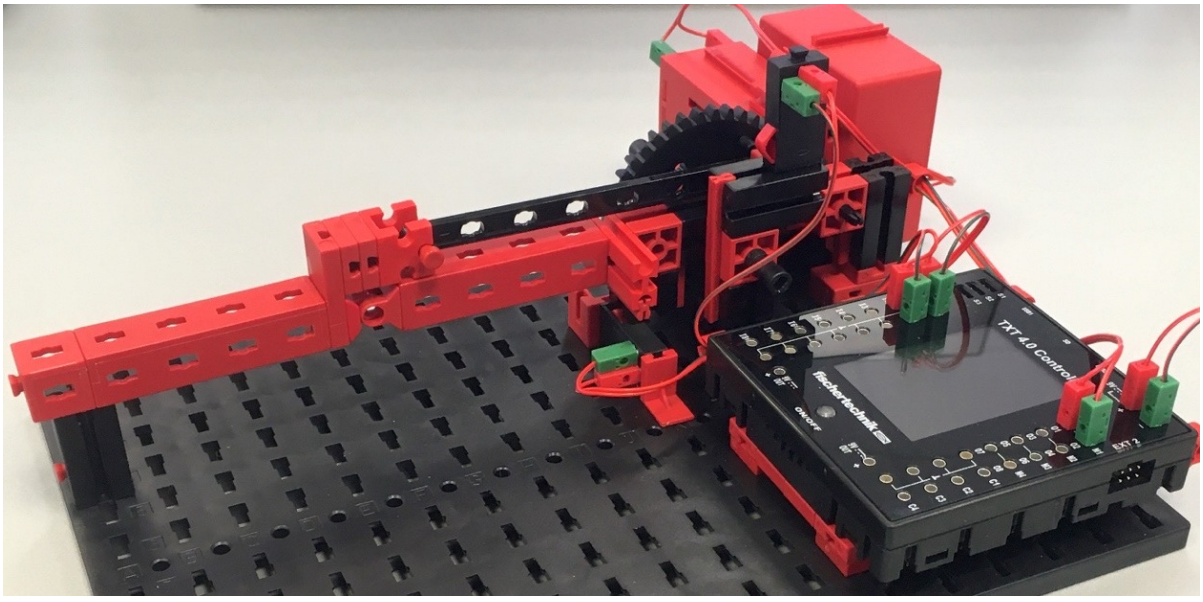
Bir otopark bariyerinin kontrol edilmesi, sayaç.

Öğrenme hedefi

- Durum diyagramlarının kullanımı (sonlu otomasyonlar)
- Dijital sensörler ile basit kontrol
- Farklı dijital sensörlerin ve enkoder (kodlayıcı) motorun işlevi
- Açısal momentumu kullanarak bir aksı kontrol etmek
- Metre değişkenleri
- Paralel süreçler (iş parçacıkları; tamamlayıcı görev)

Otomatik Bariyer İçin Verilen Görevler

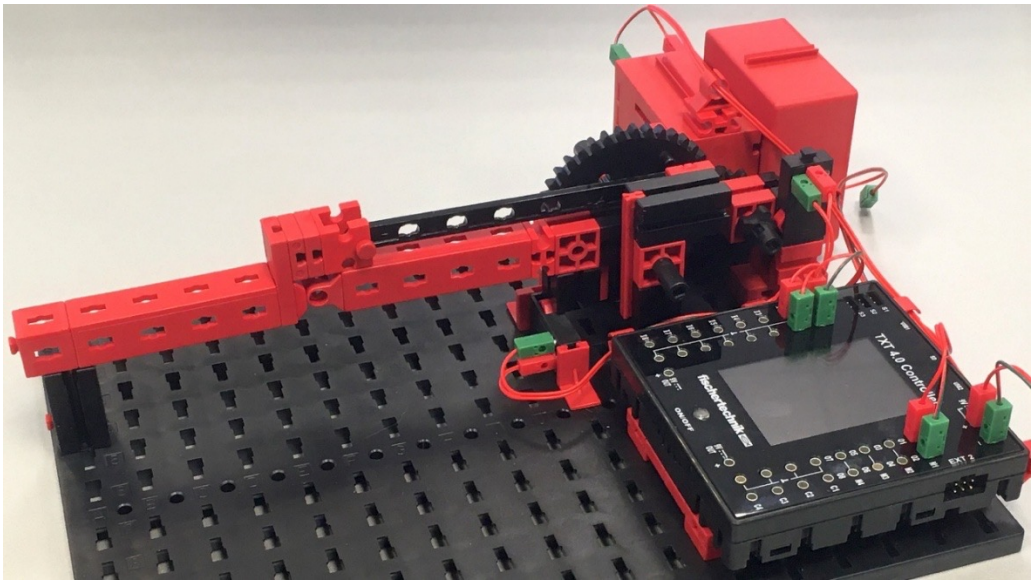
Modelin kurulumu, otomatik bariyer ilk versiyonunu (Otopark bariyer V1) yapım talimatlarına göre modelin kurulumunu yapın. Motoru ve düğmeyi TXT'ye bağlayın (kablo şemasına bakın). Motorun hangi yönde döndüğünü kontrol etmek için arayüzü kullanın (not: düşük bir hız seçin). İki durdurma konumu düğmesi, basıldığında (dijital) "1" değerini verecek şekilde I1 (alt uç konumu) ve I2 (üst uç konumu) girişlerine bağlanmalıdır.



Otomatik Bariyer Versiyon 1

Programlama görevleri

1. Durdurma konum anahtarlı otomatik bariyer:
Otomatik bariyer her 10 saniyede bir otomatik olarak açılmalı, beş saniye açık pozisyonda kalmalı ve ardından tekrar kapanmalıdır.



Otomatik Bariyer Versiyon 2

2a. İlk olarak, bariyeri kapatırken veya açarken düğmeyle ölçülebilen darbe sayısını hesaplayın.

2b. Program durumunu kapının değiştirilmiş tasarımına göre (V2) ayarlayın. Ayrıca burada bir son konum anahtarına (kapının aşağı konumunu algılaması için) ihtiyacınız olacak, böylece kapı herhangi bir durumda başlatılabilir ve kapatılabilir.

2c. Ardından, programı görev 1'den buna göre değiştirin, böylece bariyer kapısı, durdurma konum anahtarının etkinleştirilmesiyle açılırken durmaz, bunun yerine darbe sayacı belirli sayıda darbeden sonra durur.

3. Durdurma için manyetik kodlayıcılı bariyer:
Motor, eksenin dönüşü başına 63,9 darbe veren bir kodlayıcıya sahiptir. Darbe sayacı olmadan bile kapıyı hassas bir şekilde açmak için kullanabilirsiniz. Darbe sayacını ve düğmeyi çıkarın ve motorun kodlayıcı kablosunu (devre şemasında gösterildiği gibi) TXT'den C1'e bağlayın.

3a. Önce, kapınızın kaç darbe için açık veya kapalı olduğunu hesaplayın. Kapıyı manuel olarak kapatarak, sayacı sıfırlayarak ve ardından kapıyı açarak arayüz testini kullanın ve değeri kontrol edin.

3b. Programda darbe sayısını buna göre ayarlayın.

Deneysel görevler

1. Basmalı düğmeli garaj kapısı:
Otomatik bariyer her 10 saniyede bir otomatik olarak açılmalı, beş saniye açık pozisyonda kalmalı ve ardından tekrar kapanmalıdır.

Önceden, kapı otomatik olarak açılıyordu. Şimdi, bunu bir basmalı düğme anahtarı kullanarak değiştireceğiz.
İki düğmeden biri artık boş olduğundan, birini basmalı düğme anahtarı olarak kullanabilirsiniz. Sürücünün erişebilmesi için kapının önündeki bir modül 30'a monte edin. TXT'deki I2 girişine bağlayın. Garaj kapısı, basıldıktan üç saniye sonra açılmalıdır.

2a. Garaj kapısı basmalı düğme anahtarına basıldıktan üç saniye sonra açılacak şekilde durum programını ayarlayın. Programın başlangıcında, kapı önce otomatik olarak kapanmalıdır tabi eğer zaten kapalı değilse.

2b. Programınızı buna göre değiştirin.

2. Erişim kontrollü bariyer kapısı:
Ancak bariyer kapısı, sabit bir süre (5 saniye) geçtikten sonra otomatik olarak kapanmaya devam eder. İki araç hızlı hareket ettiğinde, ikisi de geçebilir - ve bir araç çok yavaşsa, kapanan kapı tarafından araç hasar görebilir. Bu senaryoların ikisini de önlemek istiyoruz.
Bariyer kapısı, yalnızca araç (veya bir yaya) geçtikten sonra (1 saniyelik gecikmeyle) kapanmalıdır. Bu nedenle, bariyer kapısına bir ışık bariyeri ekleyin (otomatik bariyer V3 modelini inşa edin, fototransistörü I3'e, LED'i 9V voltaj çıkışına bağlayın, devre şemasına bakın).

2a. Durum diyagramınızı, bariyer kapısı yalnızca ışık bariyeri artık kesintiye uğramadığında kapanacak şekilde genişletin.
2b. Bu işlevi programınıza ekleyin.

2. Erişim kontrollü bariyer kapısı:
Şimdi, kapıdan kaç aracın geçtiğini de bilmek istiyoruz.
2a. Durum diyagramınıza bir sayaç ekleyin.
2b. Programınıza, geçen arabaları sayan ve bu sayıyı TXT'nin ekranında gösteren bir değişken ekleyin.

Daha fazla bilgi

[1] Wikipedia: *Finite automaton (state machine)*

[2] Ferdinand Wagner, Ruedi Schmuki, Thomas Wagner, Peter Wolstenholme: *Modeling Software with Finite State Machines. A Practical Approach. Auerbach Publications, 2006.*

[3] Online diagram editor for creating state diagrams (Format drawio): *https://www.diagrammeditor.de/*

Otomatik Bariyer İçin Verilen Çözümler

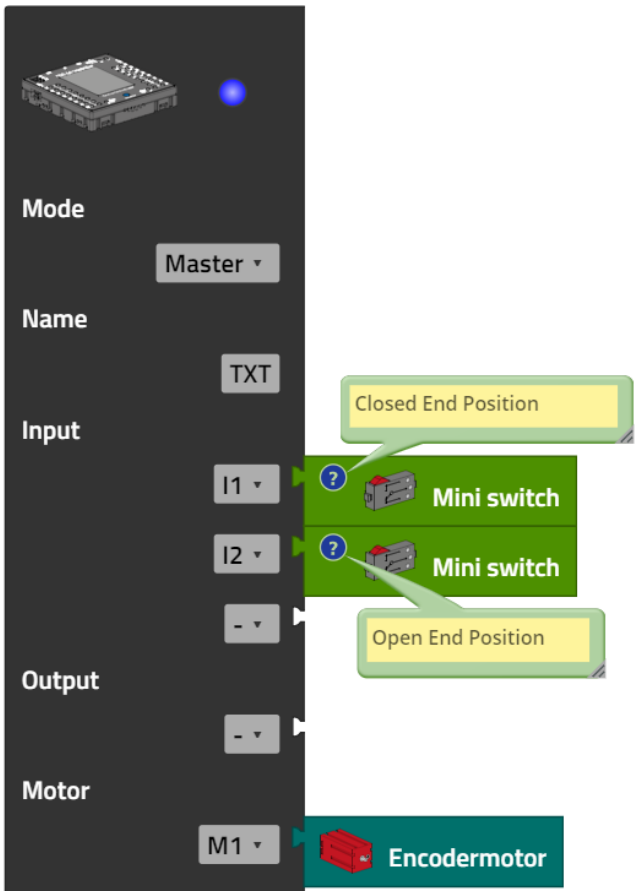
Kapıyı açarken ve kapatırken motoru durdurma işlemi üç farklı versiyona (V1, V2, V3) uygulanır: durdurma konum anahtarları, bir darbe sayacı ve bir kodlayıcı ile sistemin nasıl çalıştığını anlatılır. Bu sistem üzerinden sonlu otomasyonların önemini ve işlevini anlatılır. bu nedenle, programlamaya başlamadan önce çözüm ilk önce bir durum diyagramı olarak gösterilmelidir.

Modelin kurulumu

Yapım talimatlarına bakınız (Otopark bariyeri V1, versiyon 1).

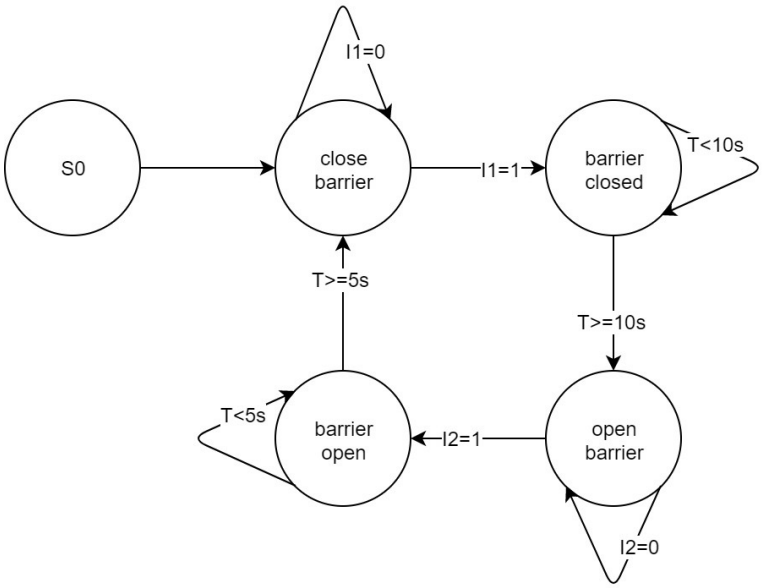
Programlama görevleri

1. Durdurma konum anahtarlı garaj kapısı:
Sensör ve aktüatörlerin yapılandırılması:



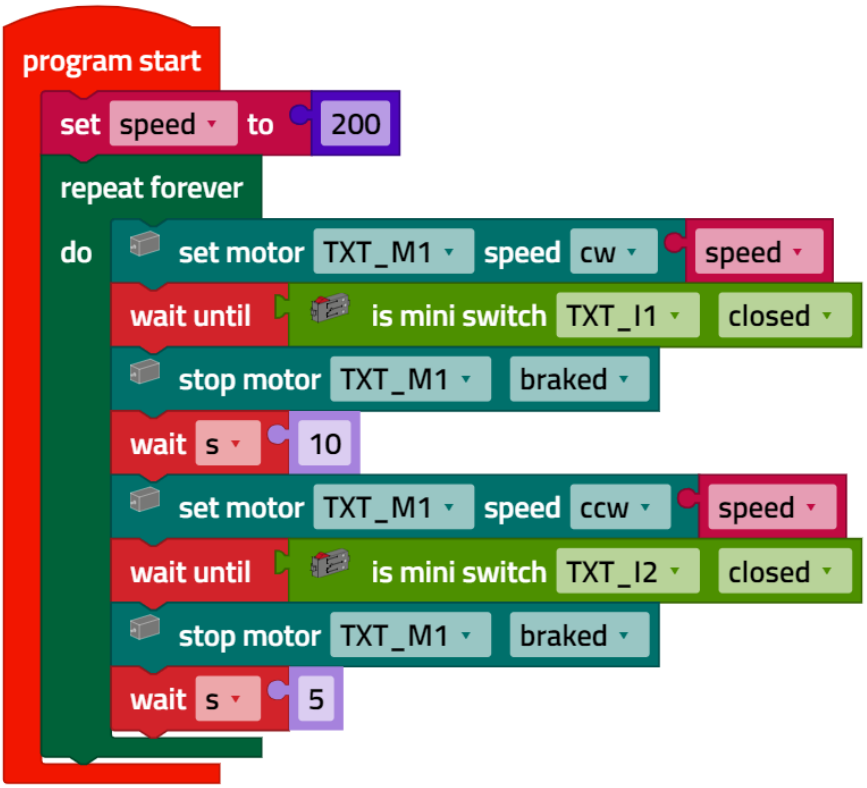
2b. Ekran yapılandırması:

1a. Durum diyagramı:



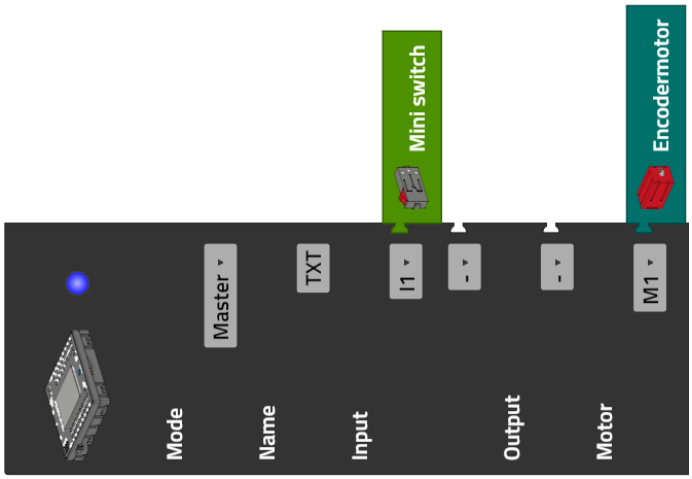
Durum geçiş diyagramı durdurma pozisyon anahtarı ile otomatik bariyer.

1b. Program (örnek):

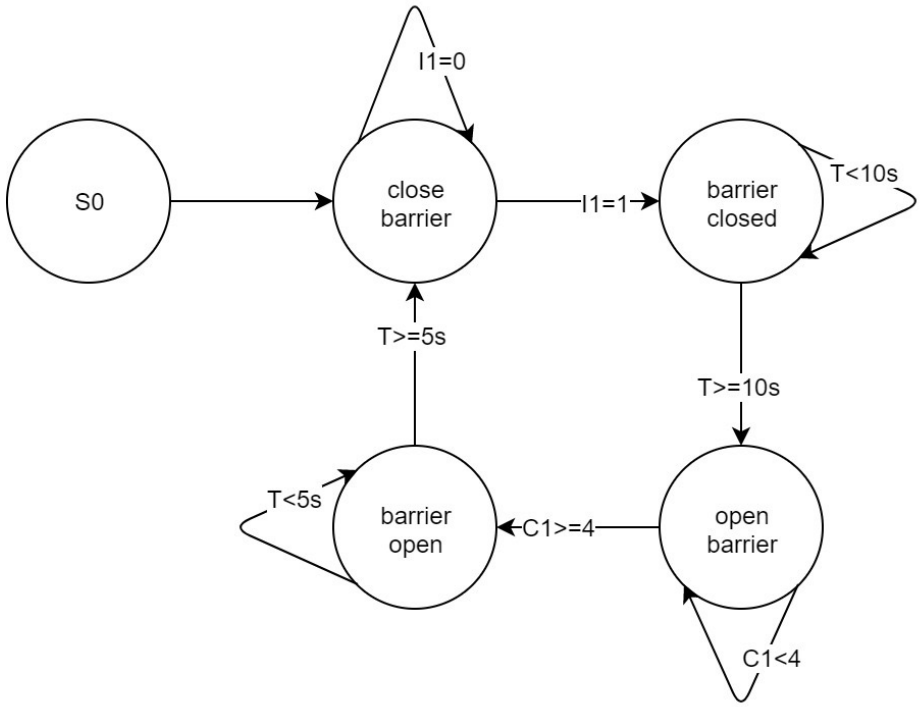


Boom_Gate_with_Stop_Position_Switch.ft
(Otomatik bariyerin stop konum pozisyonu)

2. Darbe sayacı ile garaj kapısı: Üst uç konum anahtarı C1'e bağlanır ve artık darbe sayacı tarafından etkinleştirilir (Otomatik bariyer V2 yapım talimatlarına bakın, versiyon 2).



2a. Dört darbe gereklidir (motor ekseninin bir dönüşü = 4 darbe, dişli redüksiyonu 1:4 veya kapı ekseninin dönüş başına 16 darbe veya bir çeyrek daire için dört darbe).
2b. Ayarlanmış durum diyagramı:

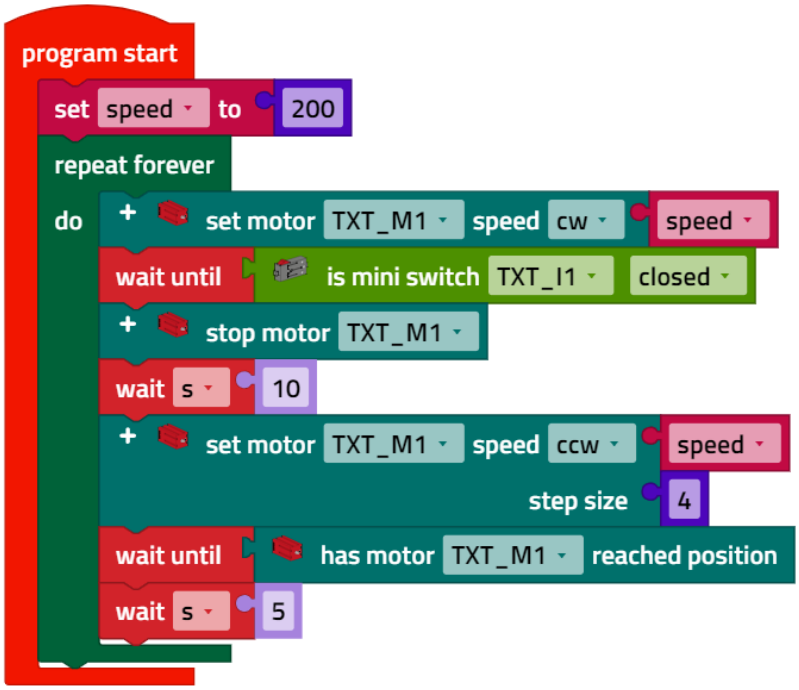


Durum geçiş diyagramı darbe sayacı ile otomatik bariyer. drawio

2c. Darbe sayacı tarafından etkinleştirilen düğme, M1 motoru için bir "kodlayıcı sayacı" olarak kullanılır, böylece motor, bir artış komutuyla dört darbe ile kontrol edilebilir.

Programlama yaparken, motor aktivasyonu için yalnızca kodlayıcı-motor komutlarının (kırmızı motor simgesi) kullanıldığından emin olun.

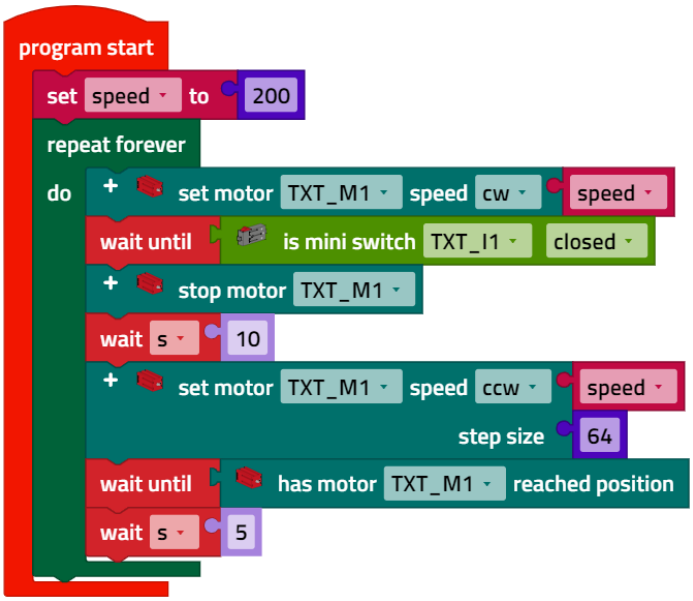
Program (örnek):



Boom_Gate_with_Pulse_Wheel.ft
(Darbe sayacı ile otomatik bariyer)

3. Manyetik kodlayıcı bariyer:
Darbe sayacı yerine, kodlayıcı motorun veri çıkışı C1'e bağlanır (Otomatik bariyer V3 yapım talimatlarına bakın, versiyon 3).
Kontrolörün yapılandırması değiştirilmek zorunda değildir.
3a. Hesaplama basit: $63.9 \cdot \frac{4}{4} \approx 64$ darbe.
3b. Program (örnek):

.

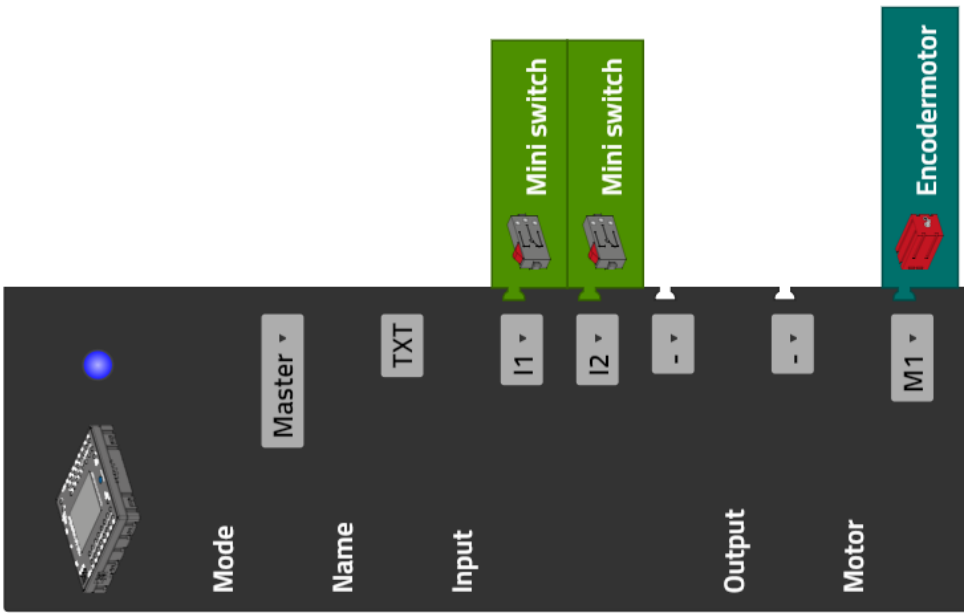


Boom_Gate_with_Encoder.ft
(Enkoder ile otomatik bariyer)

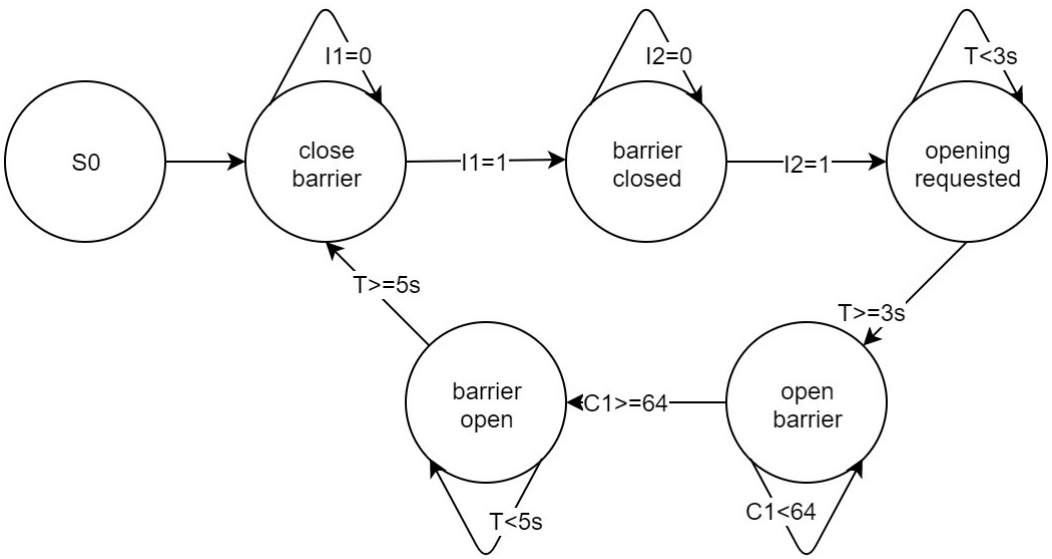
Deneysel görevler

1. Basmalı düğmeli bariyer:

Basmalı buton anahtarı I2'ye bağlanır:

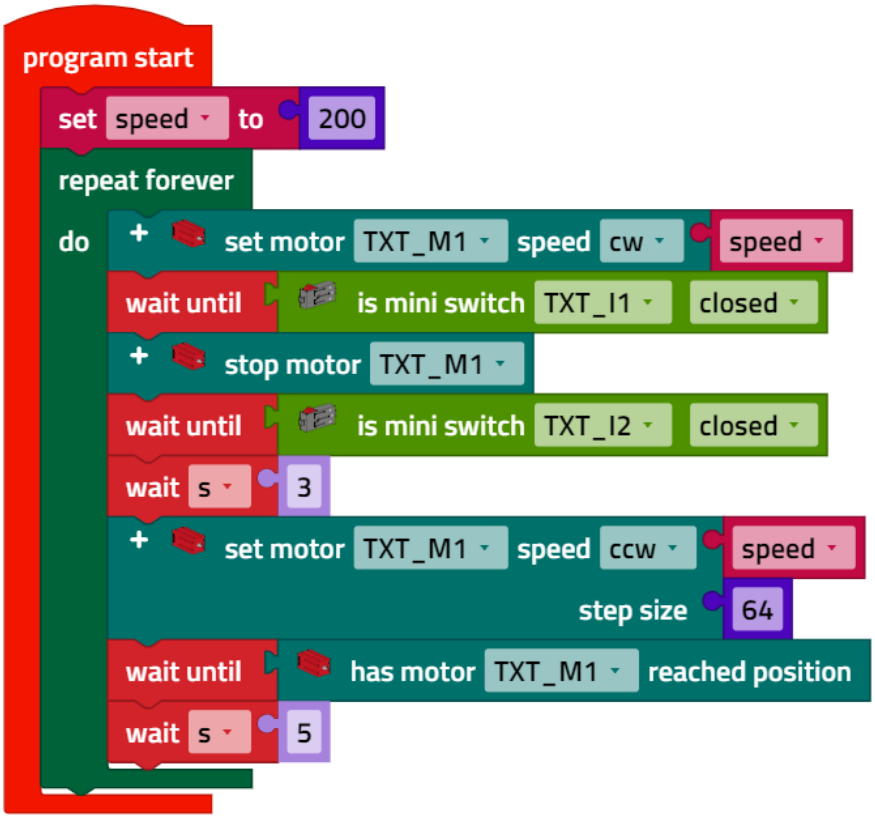


1a. Ayarlanmış durum diyagramı:



Durum geçiş diyagramı basmalı anahtar ile otomatik bariyer.

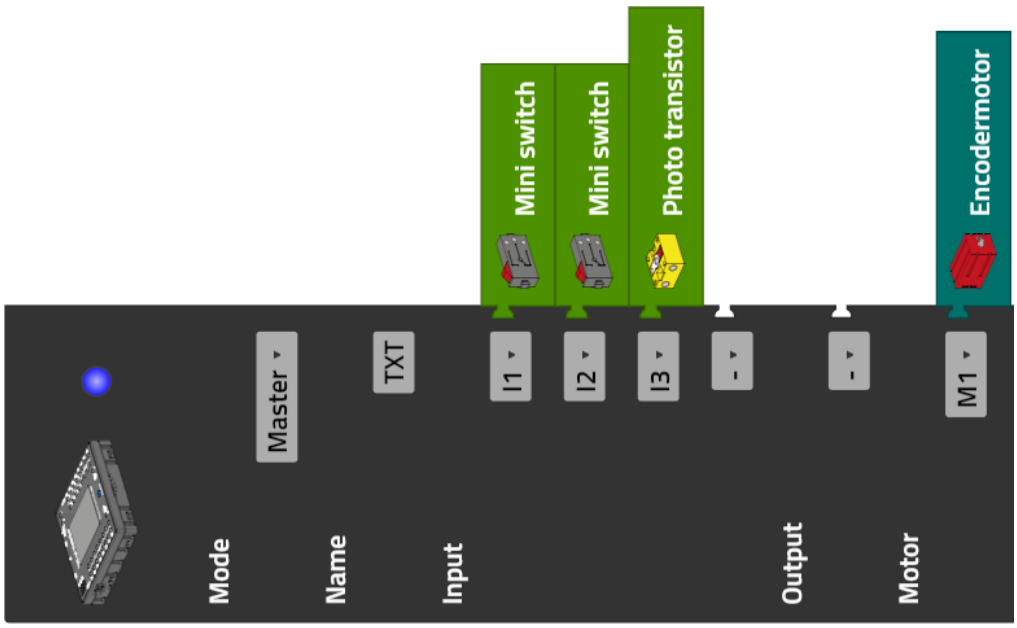
1b. Program (örnek):



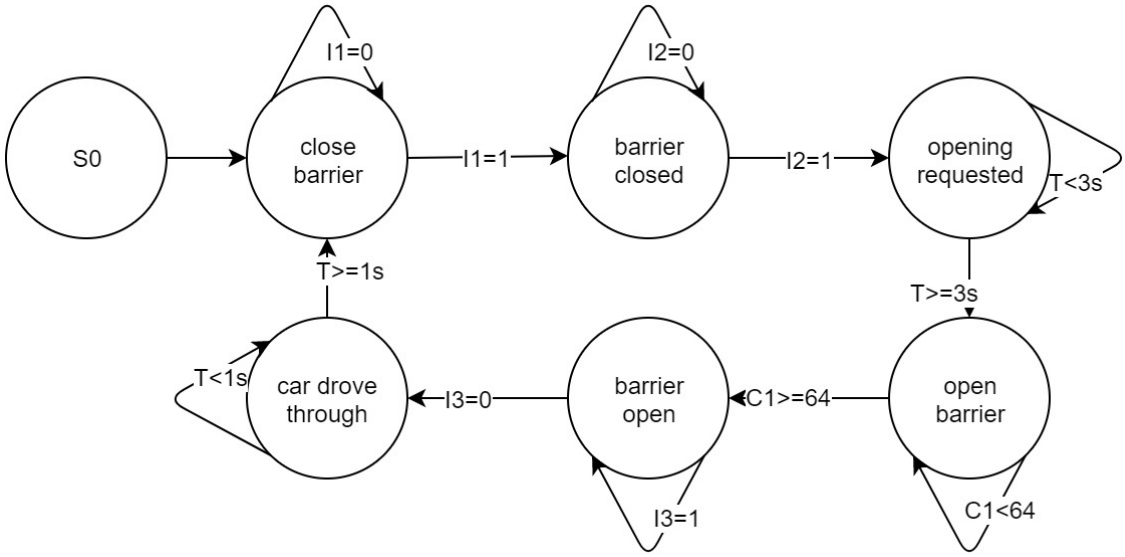
Boom_Gate_with_Pushbutton_Switch.ft
(Basmalı buton ile otomatik bariyer)

2. Erişim kontrolüne sahip bariyer kapısı

Foto transistörün yapılandırılması:

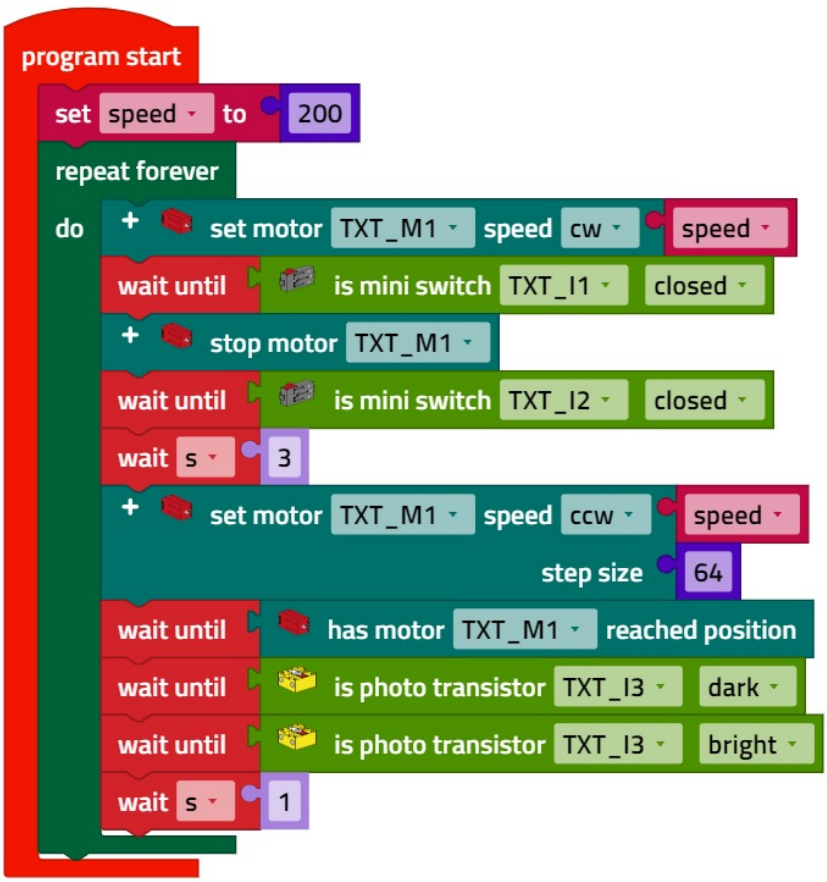


2a. Ayarlanmış durum diyagramı:



Durum geçiş diyagramı basmalı anahtar ve foto transistör ile otomatik bariyer.

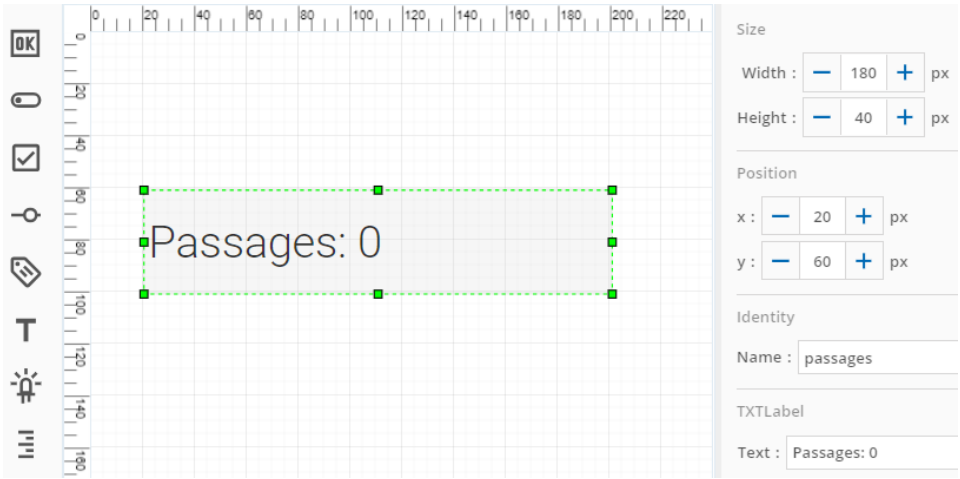
2b. Program (örnek):



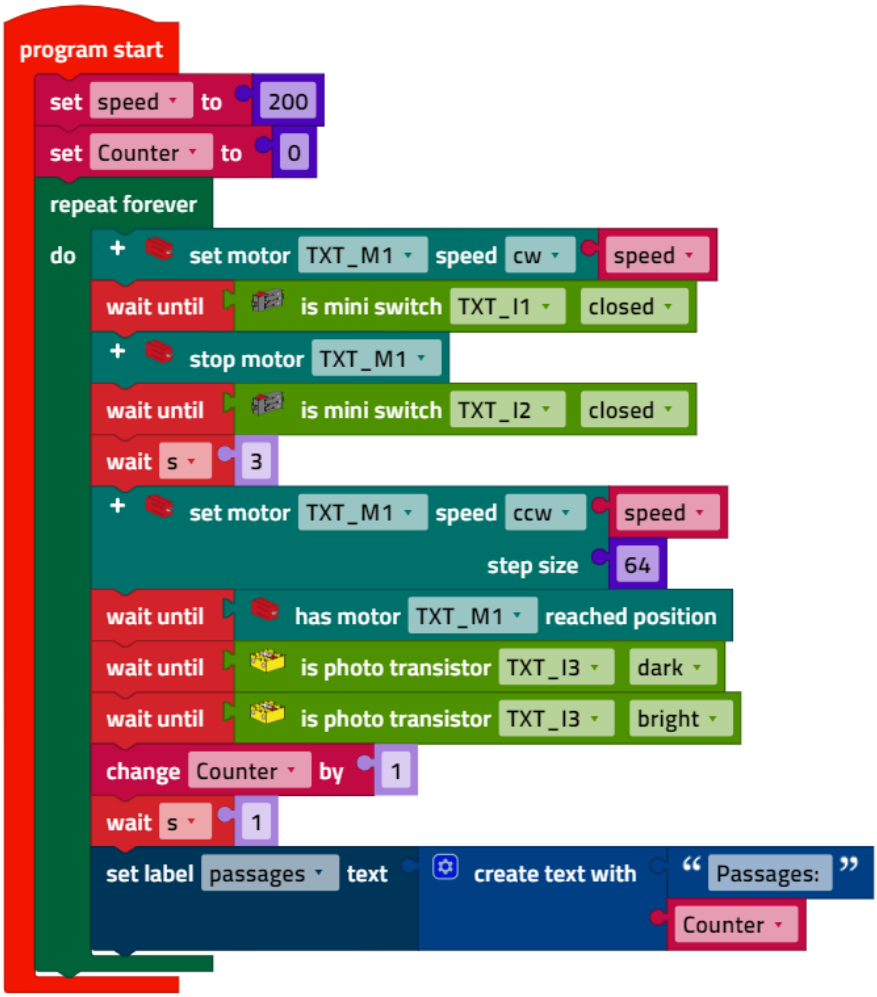
Boom_Gate_with_Pushbutton_Switch_and_Lightbeam.ft
(Basmalı anahtar ve foto transistör ile otomatik bariyer)

3. Araç sayacı olan bariyer

3a. Görüntü çıktısını yapılandırma:



3b. Program (örnek):



Boom_Gate_with_Pushbutton_Switch_Lightbeam_and_Counter.ft
(Darbe sayacı, Basmalı anahtar ve foto transistör ile otomatik bariyer)