

KONU 5:

HABERLEŞME – MORS TELGRAFI - ŞİFRELEME

HABERLEŞME – MORS TELGRAFI - ŞİFRELEME ÖĞRENME HEDEFİ

GÖREV

Bu görevler iletişim, kodlama ve şifreleme konularını ele almaktadır. Basit bir “ışık sinyali” birden çok adımda Mors kodu ve Vigenère şifrelemesi kullanılarak veri iletimi için güvenli bir kanala dönüştürülmektedir.

Konular

İletişim için veri kodlama, eşzamanlı verici/alıcı, Mors kodu, şifreleme (Sezar şifrelemesi, Vigenère şifrelemesi).

Öğrenme Hedefi

- Bir iletişim kanalının işlevini anlamak
- Bir iletişim kanalı üzerinden iletim için veriyi kodlamak
- Bir sinyal dizisini taramak
- Bir metin dizisini işlemek, fonksiyonlar

Yetkisiz erişime karşı korumak için verilerin şifrelenmesi ve şifresinin çözülmesi

Gerekli Süre

Öğrenciler Mors düğmesini yalnızca birkaç dakikada kurabilir.

Öğrencilerin programlama alanındaki önceki deneyimlerine bağlı olarak, programlama görevlerini çözmek için 90-180 dakikaya ihtiyaç duyacaklardır.

DeneySEL görevler, öğrencinin sınıf seviyesine bağlı olarak, ek 180-270 dakika gerektirecektir. Mors alıcısı, iki TXT kullanılarak grup halinde test edilmelidir.

Öğrenciler, derse başlamadan önce Mors kodu ile Sezar ve Vigenère şifreleri hakkında bilgilendirilmiş olmalıdır (bkz. ilgili ek materyaller).

Ek

Görev 5: İletişim – Mors Kodu – Şifreleme

Gerekli Malzemeler

- Program geliştirme için PC, yerel olarak veya web arayüzü üzerinden.
- Programı TXT4.0'a aktarmak için USB kablosu veya BLE ya da WiFi bağlantısı.
- Program şablonu (Mors kodu için): Morse_Code_Template.ft

Ek Bilgiler

[1] Albrecht Beutelspacher: Kryptologie: Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen. 10. Baskı, Springer Verlag, 2015.

[2] Simon Singh: Codes. Die Kunst der Verschlüsselung. Impian, 2021.

[3] Durum diyagramları oluşturmak için çevrimiçi diyagram düzenleyici (drawio formatında):

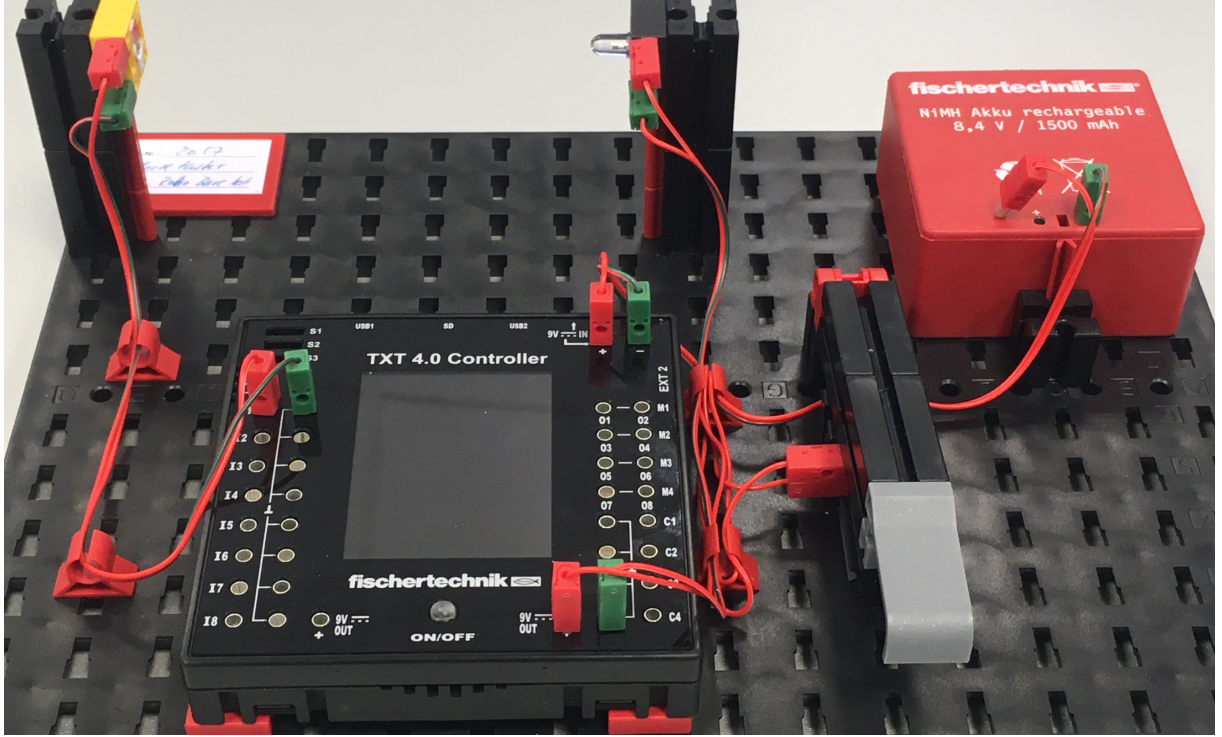
<https://www.diagrammeditor.de/>

HABERLEŞME – MORS TELGRAFI - ŞİFRELEME GÖREVİ

Yapı Görevi

Mors düğmesini inşa edin. Etkinleştirildiğinde, iletim LED'i yanmalıdır.

Taban plakasına, iletim LED'i ve bir fototransistörden oluşan bir ışık bariyeri kurun (bkz. yapım talimatları). Fototransistörü I1'e, alıcı gösterge LED'ini ise O2'ye bağlayın (bkz. bağlantı şeması).



Işık Yoluyla İletişim: Işık Bariyeri Bir İletişim Kanalı Olarak

Programlama Görevleri

1. Işık Sinyalleri

Görev 2’de LED ve fototransistörden oluşan bir ışık bariyeri ile çalıştınız. Şimdi bu tür bir ışık yolunu (light path) sinyalleri iletmek için kullanacağız. Gönderici Mors düğmesine bastığında, bir ışık sinyali iletilecek. Fototransistör bu ışık sinyalini aldığı anda, alıcı tarafında bir sinyal LED’ini yakmalıdır. Bu şekilde bilgi aktarmak için kullandığımız bağlantı türüne bir iletişim kanalı diyoruz.

1a. Alıcı için bir durum diyagramı (state diagram) çizin.

1b. Işık sinyalinin alıcısı için bir Blockly programı yazın.

1c. Programı, alınan sinyallerin sayısını ekranda gösteren bir sayaç ekleyerek genişletin.

Bu basit ışık iletişim kanalı başlangıçta sadece basit bir sinyal, yani tek bir bitlik durum bilgisi (açık/kapalı) iletmek için kullanılabilir.

Sayacı kullanarak alıcı tarafından yorumlanabilecek (tek) bir değer iletebiliriz.

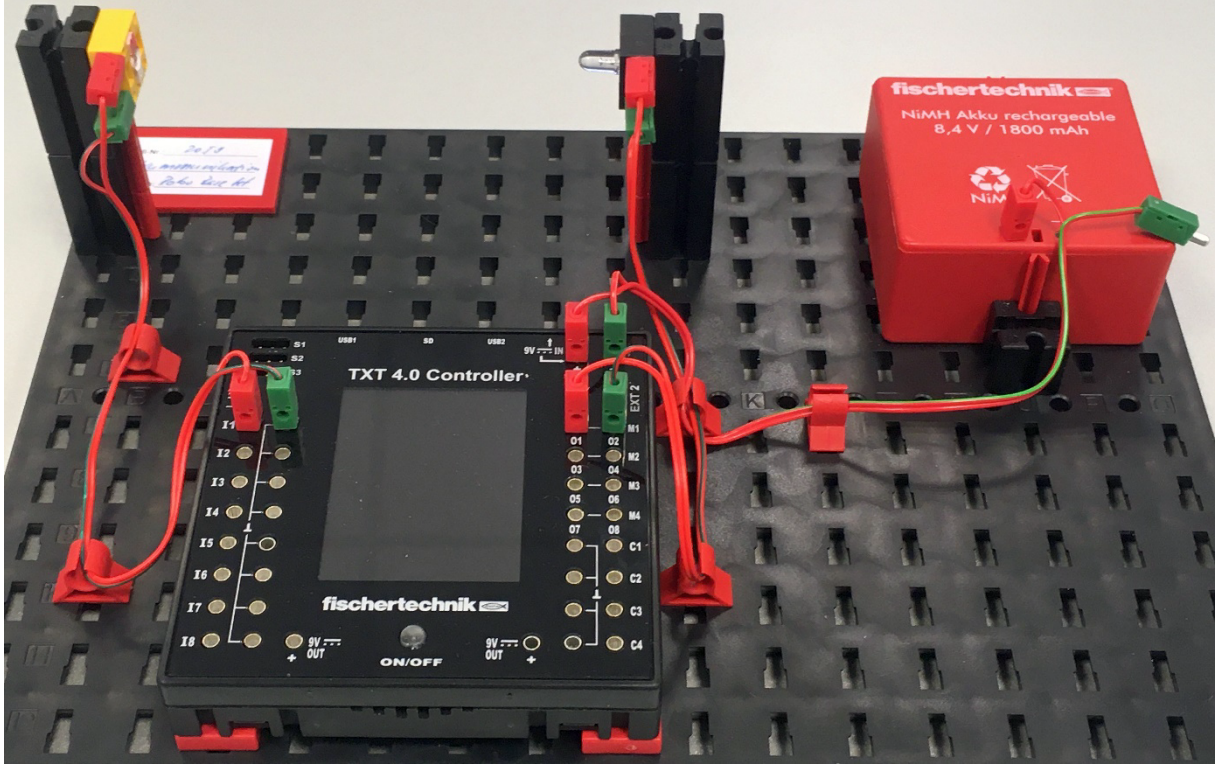
2. İkili Kod Gönderici

Daha fazla veri iletmek istiyorsak, iletilen veriyi temsil (kodlama) edebileceğimiz bir koda ihtiyacımız olacak. Bunun için kısa ve uzun sinyallerden oluşan ikili bir kod kullanacağız:

- Kısa bir sinyal (“dit”) “0” değerini temsil eder.
- Uzun bir sinyal (“dah”) “1” değerini temsil eder.

Tek tek sinyaller, aralarına konan bir duraklama (bekleme süresi) ile birbirinden ayrılır. Ayrıca alıcının kısa ve uzun sinyalleri ayırt edebilmesi için bir “ritim” gereklidir. Bu, sinyallerin ve duraklamaların uzunluğuyla belirlenir:

- Bir “dah” bir “dit”in üç katı uzunluğundadır.
- İki sinyal arasındaki duraklama süresi bir “dit” uzunluğundadır.



Programlama Görevleri

Şimdi, TXT'nin sinyalleri iletmesini sağlayalım. Bunun için butonsuz iletim LED'ini O1 çıkışına ve GND'ye doğrudan bağlayacağız (doğru polariteye dikkat edin!).

2a.

Belirli bir "0" / "1" dizisini, kısa ("dit") ve uzun ("dah") ikili sinyal dizisine (her iki sinyal arasında bir bekleme süresi olacak şekilde) dönüştüren bir program için bir durum diyagramı çizin.

2b.

İletişim kanalımız aracılığıyla belirli bir "0" / "1" dizisini (metin olarak) ileten bir Blockly programı yazın. Birim zaman olarak 100 ms (= bir "dit" süresi) kullanın.

İpucu: Temel zaman birimini bir değişkene atarsanız, daha sonra iletişim hızını bu değişkenin değerini değiştirerek kolayca ayarlayabilirsiniz.

2c.

Programa eşzamanlı (concurrent) bir işlem ekleyerek alınan sinyalleri alıcı LED ile görüntüleyin. Bu işlevi 1b alt-görevinde zaten programlamıştınız.

3. Mors Kodu Gönderici

Şimdi daha uzun bir metin mesajını Mors koduna çevirip göndermek istiyoruz. Mors kodunda her karakter (harfler ve rakamlar), Mors alfabesindeki belirli bir "0" / "1" (Mors karakterleri) dizisi ile kodlanır (ilgili materyallere bakınız). Karakterin sonu, üç "dit" uzunluğunda bir duraklama ile belirtilir.

3a.

2a alt-görevindeki durum diyagramına dayanarak basit bir durum diyagramı çizin.

3b.

2c alt-görevindeki programınızı genişleterek, belirli bir metni önce harf harf Mors karakterlerine çeviren (kodlayan), ardından bu Mors karakterlerini ışık sinyalleri olarak ileten bir program haline getirin. Bunun için "Morse_Code_Template.ft" adlı program parçasını

kullanın. Bu parça, alfabenin tüm harflerini ve Mors alfabetini içeren önceden tanımlanmış iki liste içerir.

Test etmek için “thequickbrownfoxjumpsoverthelazydog” metnini iletin. Bu yaygın kullanılan test metni, İngiliz alfabesindeki tüm harfleri içerir.

DeneySEL Görevler

1. İkili Kod Alıcısı

Bit dizisi göndericisi artık iletilen bit dizisini sadece LED ile göstermekle kalmayıp, tüm ikili diziyi de ekranda göstermelidir.

1a.

İkili kod gönderici programınızın alıcı sürecini genişleterek alınan sinyalleri bit olarak çözümleyin ve iletilen dizi yerine, alınan ikili diziyi TXT'nin ekranında görüntüleyin.

1b.

Programı, başka bir öğrencinin TXT'sine bit dizisi göndererek test edin. Fototransistör üzerinden diğer TXT'ye veri gönderin.

1c.

Bir “dit” ne kadar kısa olabilir ve yine de iletim hatasız gerçekleşir? İletişim kanalınız kaç bit/s hızında iletişim yapabilir?

2. Mors Kodu Alıcısı

Şimdi Mors Kodu göndericisine bir de Mors Kodu alıcısı eklemek istiyoruz. Alıcı süreci, alınan ikili karakter dizisini Mors karakterlerinde birleştirmeli, ardından bu Mors karakterlerini çözümleyerek alfabedeki harflere dönüştürmeli ve ekranda göstermelidir.

2a.

Programınızdaki ikili dizi için alıcı süreci genişleterek Mors karakterlerinin tespitini ekleyin. Bunları çözümleyin ve alınan metni ekranda görüntüleyin.

2b.

Programı, başka bir öğrencinin TXT'siyle test edin. Mesajınızı diğer TXT'nin fototransistörüne iletin.

2c.

“Dit” ne kadar kısa olabilir ve yine de hatasız iletim sağlanır?

2d.

Genellikle bir Mors vericinin hızı “dakikada kelime” (WpM) cinsinden ifade edilir. Standart olarak “Paris” kelimesi (kelime sonu işareti dahil) kullanılır. 100 ms'lik bir “dit” süresi ile Mors verici/alıcınızın hızı kaç WpM'dir?

3. Mors Eğitmeni

Mors Kodu alıcısıyla, Mors kodunu pratik yapmak için kullanabileceğiniz bir “Mors Eğitmeni” elde etmiş olursunuz. Bunun için iletim LED'ini tekrar Mors düğmesine bağlayın. Mors düğmesini manuel olarak kullanarak “SOS” (veya başka bir metni) iletmeyi deneyin. Mors Kodu alıcısının metni doğru algıladığından emin olun.

4. Şifreleme – Sezar Şifresi

Işık iletişim kanalımızla yapılan iletim korunmamakta ve yetkisiz kişiler ilettiğimiz mesajları okuyabilir. Mesajı iletmeden önce şifreleyerek bunu engellemek istiyoruz.

DeneySEL görev 2'deki Mors Kodu alıcısı programınıza bir Sezar şifrelemesi ekleyin (ilgili materyallere bakınız).

Not: Anahtarınızı (harflerin alfabedeki kaydırma miktarı) bir değişkende saklayın, böylece ileride kolayca değiştirebilirsiniz.

5. Şifreleme – Vigenère Şifresi

Vigenère şifrelemesi basit frekans analizine karşı koruma sağlar. Deneysel görev 4'teki Sezar şifrelemesini Vigenère şifrelemesiyle değiştirin (ilgili materyallere bakınız).

Not: Anahtarları programda belirtmek yerine, TXT'nin ekranı aracılığıyla girebilirsiniz.

Ek

Görev 5: İletişim – Mors Kodu – Şifreleme

Gerekli Malzemeler

- Program geliştirme için PC (yerel veya web arayüzü üzerinden).
- Programı TXT4.0'a aktarmak için USB kablosu, BLE veya WiFi bağlantısı.
- Program şablonu (Mors kodu için): **Morse_Code_Template.ft**

Ek Bilgiler

[1] Albrecht Beutelspacher: *Kryptologie: Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen*. 10. Baskı, Springer Verlag, 2015.

[2] Simon Singh: *Codes. Die Kunst der Verschlüsselung*. Impian, 2021.

[3] Durum diyagramları oluşturmak için çevrimiçi diyagram düzenleyici (drawio formatında): [URL eksik – uygun bir düzenleyici önerilebilir].

<https://www.diagrameditor.de/>

HABERLEŞME – MORS TELGRAFI - ŞİFRELEME ÇÖZÜMÜ

Görevleri çözerken, öğrenciler veri iletimi sırasında bir verici ve alıcının nasıl senkronize edildiğini ve verilerin bir iletişim kanalı üzerinden iletimi için nasıl kodlandığını ve şifrelendiğini öğrenir. Aynı zamanda, metin dizilerini ve dizinli listeleri nasıl kullanacaklarını da öğrenirler.

Öğrencilere yapım talimatlarına ek olarak alfabeleri içeren bir program parçası ("**Morse_Code_Template.ft**") sağlanır.

Yapım Görevi

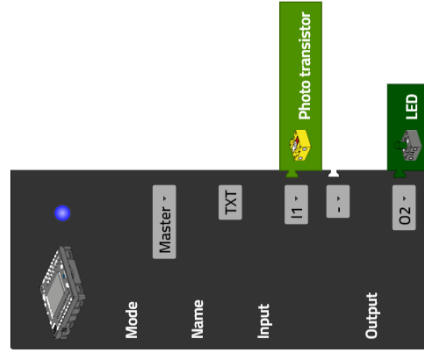
Yapım talimatlarına bakınız.

- **Verici LED:** 9V çıkışına ve GND bağlantısına Mors düğmesi aracılığıyla bağlanır. Mors düğmesi yalnızca TXT'den güç kaynağı olarak yararlanır.
- **Alıcı LED:** O2'ye bağlanır.
- **Fototransistör:** I1'e bağlanır.

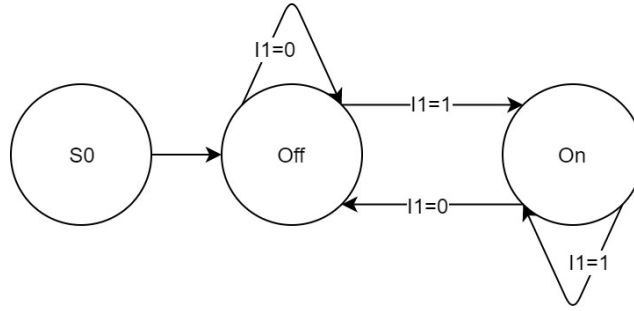
Programlama Görevleri

1. Işık Sinyalleri

Bağlantı Yapılandırması:

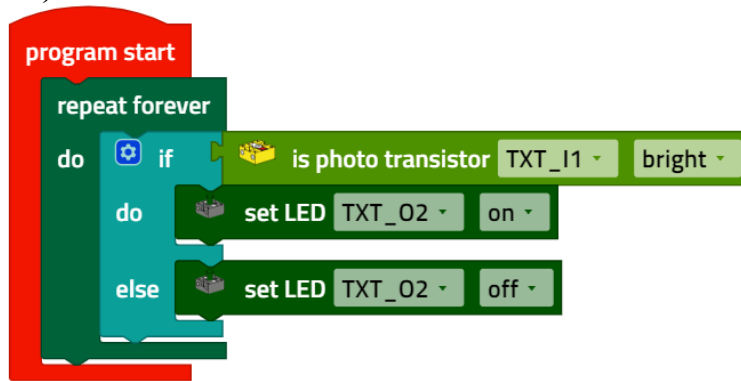


1a. Alıcı için Durum Diyagramı:



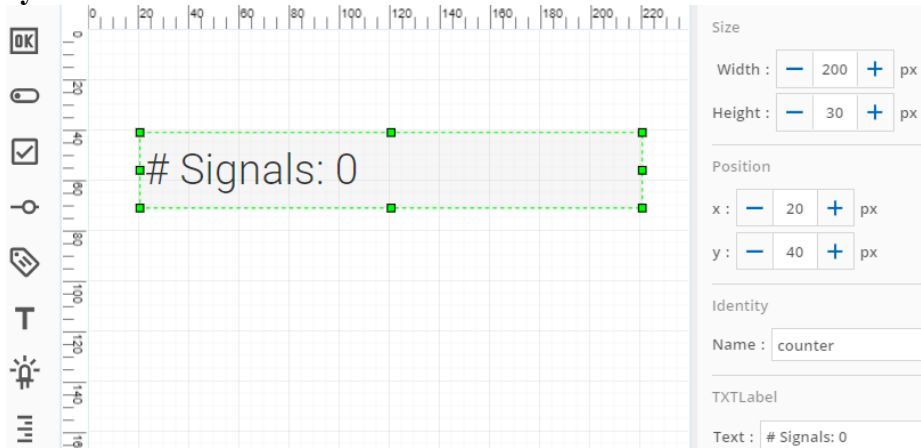
State-Transition_Diagram_Light_Signal.drawio

1b. Program (örnek) alıcı:

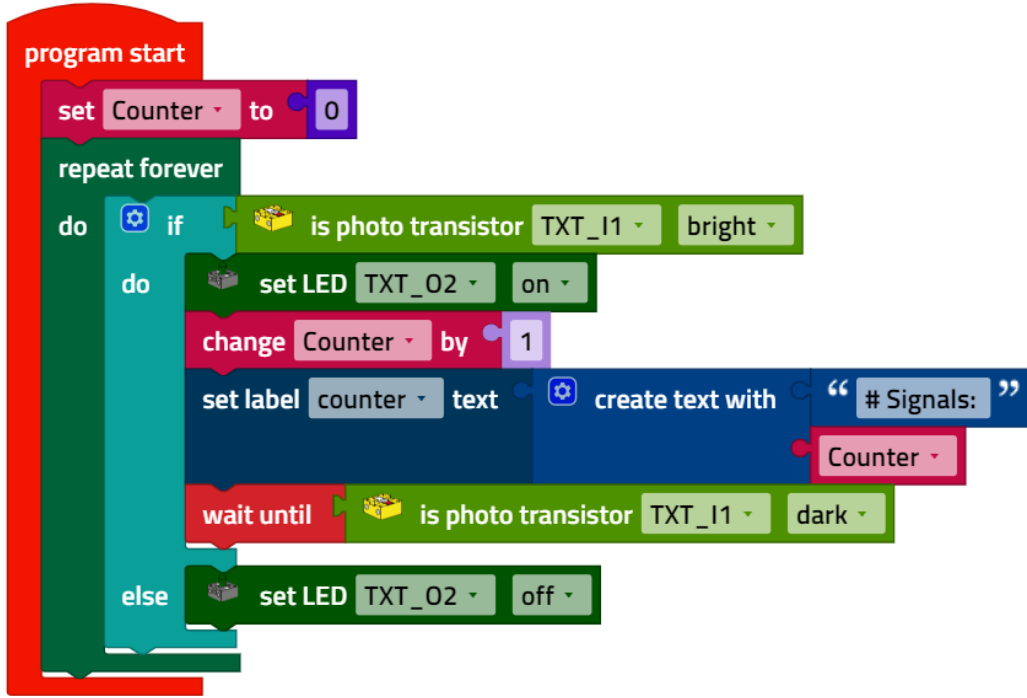


Light_Signal.ft

1c. Ekran Ayarları:



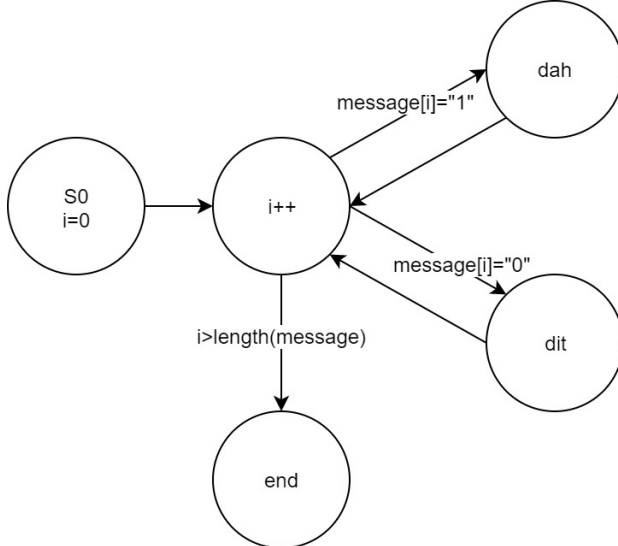
Program (örnek): Sayaçlı alıcı:



Light_Signal_Counter.ft

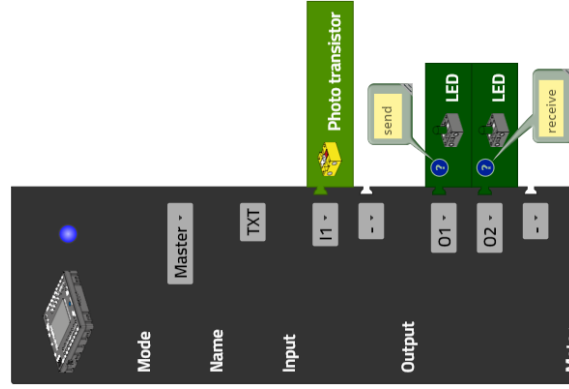
2. İkili Kod Gönderici

2a. Bit Dizisi Gönderici için Durum Diyagramı:



State-Transition_Diagram_Bit_Sequence_Sender.drawio

2b. Verici LED için Bağlantı Yapılandırması:



Ekran Ayarları:

Size

Width : 200 px

Height : 30 px

Position

x : 20 px

y : 60 px

Identity

Name : message

TXLabel

Text : M: -

İletilen bit dizisinin son 20 karakteri ekranda görüntülenir.

Bu görevi çözmek için, mesaj metinlerini değerlendirmek amacıyla metin fonksiyonunu kullanmak ve programın bir kısmını fonksiyonlara ayırmak iyi bir fikirdir. Böylece, genel program anlaşılır bir şekilde kalır.

Program (örnek): Bit Dizisi Gönderici

```

program start
  set dit to 100
  set dah to 3 x dit
  set bit_sequence to "11011100101001110100101110110"
  set send_bit_sequence to ""
  set send_bit_counter to 1
  repeat length of bit_sequence times
    do
      set send_bit to in text bit_sequence get letter # send_bit_counter
      send_bit with:
        bit send_bit
      to send_bit_sequence append text send_bit
      change send_bit_counter by 1
      set label message text create text with "M:"
      in text
      get substring from letter # from end 20
      to last letter
    end
  end
  wait s 10

```

```

define send_bit with: bit
  if bit = "1"
  do
    dah
  else
    dit
  end

```

```

define dah
  set LED TXT_01 on
  wait ms dah
  set LED TXT_01 off
  wait ms dit

```

```

define dit
  set LED TXT_01 on
  wait ms dit
  set LED TXT_01 off
  wait ms dit

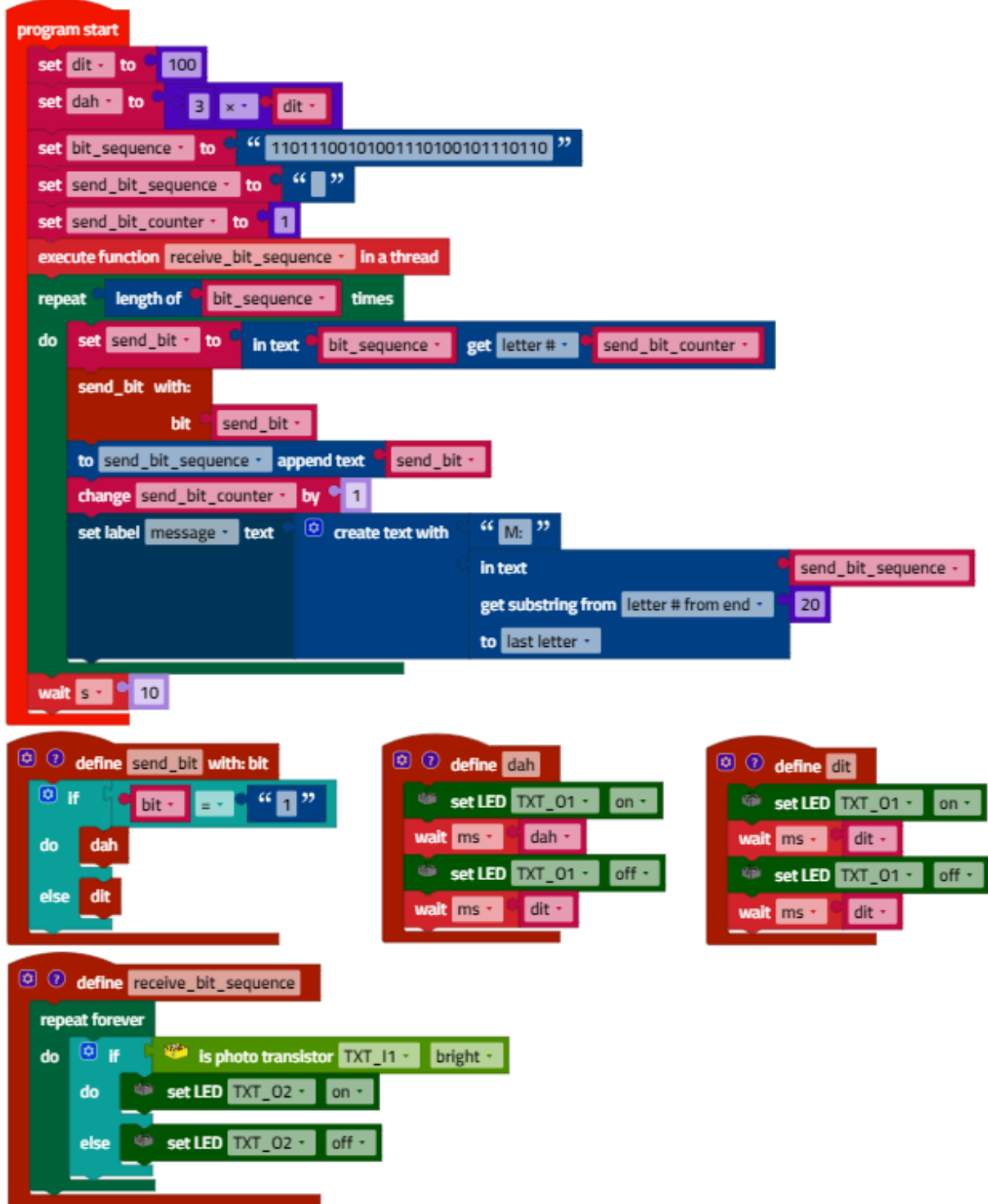
```

Bit_Sequence_Sender.ft

2c. Alıcı Rutini (“receive_bit_sequence”)

Alıcı rutini eşzamanlı bir işlem (thread) olarak başlatılır. Deneysel görevlerde bir kod çözücü (decoder) ile genişletilebilir.

Program (örnek): Alım Görüntülemeli (LED) Bit Dizisi Gönderici



The image displays a Scratch script for a bit sequence sender and receiver. The main script starts with setting variables: 'dit' to 100, 'dah' to 3 times 'dit', 'bit_sequence' to "11011100101001110100101110110", 'send_bit_sequence' to "", and 'send_bit_counter' to 1. It then executes the 'receive_bit_sequence' function in a thread. A loop repeats the length of 'bit_sequence' times, performing the following steps: setting 'send_bit' to the character at 'letter #' in 'bit_sequence', sending it with 'send_bit' as the bit, appending 'send_bit' to 'send_bit_sequence', and incrementing 'send_bit_counter'. After the loop, it creates a message 'M:' and extracts a substring of length 20 from the end of 'send_bit_sequence'. A 10-second wait follows. Below the main script are three defined functions: 'send_bit with: bit' which checks if the bit is '1' and sends 'dah' or 'dit'; 'dah' which turns LED TXT_01 on for 'dah' ms and off for 'dit' ms; and 'dit' which turns LED TXT_01 on for 'dit' ms and off for 'dit' ms. The 'receive_bit_sequence' function is a 'repeat forever' loop that checks if photo transistor TXT_I1 is bright, turning LED TXT_O2 on, otherwise turning it off.

```

program start
  set dit to 100
  set dah to 3 x dit
  set bit_sequence to "11011100101001110100101110110"
  set send_bit_sequence to ""
  set send_bit_counter to 1
  execute function receive_bit_sequence in a thread
  repeat length of bit_sequence times
    do set send_bit to in text bit_sequence get letter # send_bit_counter
      send_bit with:
        bit send_bit
      to send_bit_sequence append text send_bit
      change send_bit_counter by 1
      set label message text create text with "M:"
      in text
      get substring from letter # from end 20
      to last letter
  wait s 10

define send_bit with: bit
  if bit = "1"
  do dah
  else dit

define dah
  set LED TXT_01 on
  wait ms dah
  set LED TXT_01 off
  wait ms dit

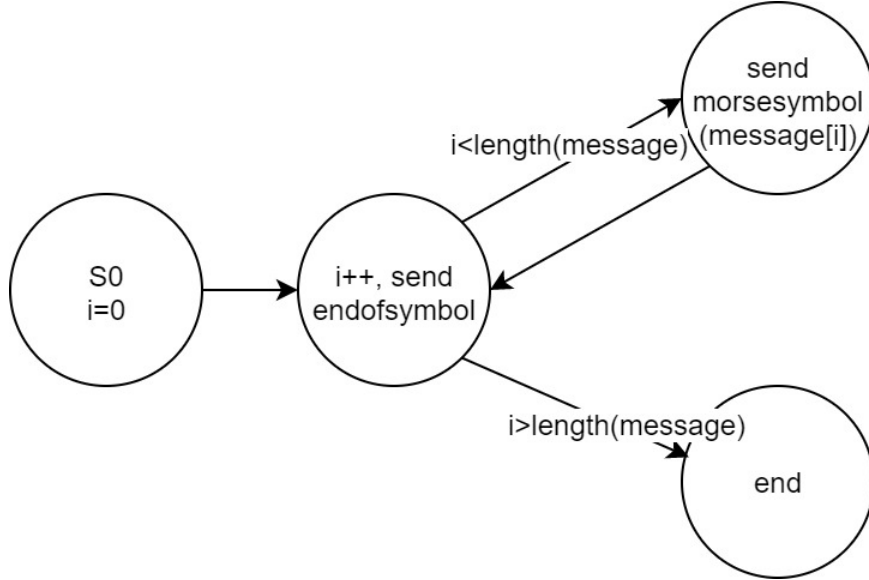
define dit
  set LED TXT_01 on
  wait ms dit
  set LED TXT_01 off
  wait ms dit

define receive_bit_sequence
  repeat forever
  do if is photo transistor TXT_I1 bright
    do set LED TXT_O2 on
    else set LED TXT_O2 off
  
```

Bit_Sequence_Sender_Receive_LED.ft

3. Mors Kodu Gönderici

3a. Mors Kodu Gönderici için Durum Diyagramı:



State-Transition_Diagram_Morse_Code_Sender.drawio

Bir mesajdaki bir karakterin iletilmesi, 2a alt-görevindeki bit dizisi göndericisinde olduğu gibi bir Mors sembolünün (bir dizi “0” ve “1”) iletilmesine karşılık gelir.

3b. Liste fonksiyonu, bir liste öğesinin dizinini belirlemek ve belirli bir dizine sahip öğeyi okumak için kullanılabilir. Bu, metni kodlamayı oldukça şık bir hale getirir.

Not: Mors alfabesi küçük ve büyük harfler arasında ayırım yapmaz. Bu nedenle, mesaj metni kodlanmadan önce tamamen büyük harfe dönüştürülmelidir.

Çözüm Varyantları: Mesaj, TXT ekranındaki giriş alanı aracılığıyla da yazılabilir. Aşağıdaki örnek çözümde mesaj program kodunda bir değişken olarak belirtilmiştir. (Her iki alfabenin başlatılması şekil üzerinde kısaltılmış bir biçimde gösterilmiştir.)

Program (example):

```

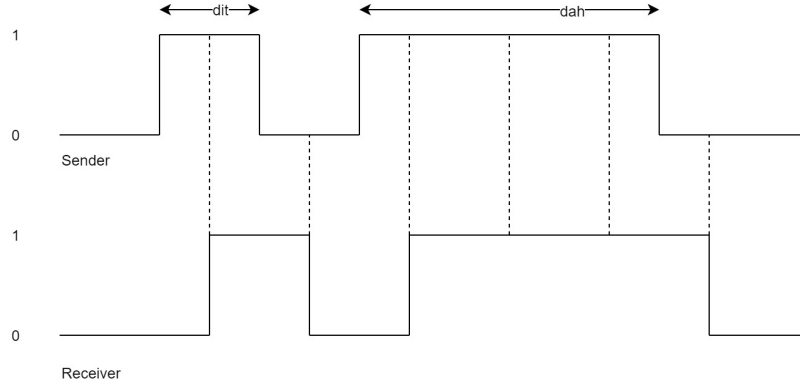
program start
  set dit to 100
  set dah to 3 x dit
  set alphabet to create list with
  "A"
  "B"
  "C"
  "D"
  "E"
  "F"
  "G"
  "H"
  "I"
  "J"
  "K"
  "L"
  "M"
  "N"
  "O"
  "P"
  "Q"
  "R"
  "S"
  "T"
  "U"
  "V"
  ...
  set message to to UPPER CASE "thequickbrownfoxjumpsoverthelazydog"
  set send_message to ""
  set send_character_counter to 1
  execute function receive_bit_sequence in a thread
  repeat length of message times
  do
    set send_character to in text message get letter # send_character_counter
    set send_index to in list alphabet find first occurrence of item send_character
    set send_morse_symbol to in list morse_alphabet get # send_index
    send_morse_symbol with:
      morse_symbol send_morse_symbol
    to send_message append text send_character
    change send_character_counter by 1
    set label message text create text with "M:"
    in text
    get substring from letter # from end 15
    to last letter
  wait s 10
  
```

Morse_Code_Sender.ft

DeneySEL Görevler

1. Bit Dizisi Alıcısı

Alıcı süreci, fototransistör tarafından “tetiklenir.” Ardından süreç, sinyal uzunluğunun yaklaşık yarısı kadar bir süre bekler ($0.5 \times \text{“dit”}$) ve sinyal süresinin ortasında yeniden sinyali örnekler (bkz. taslak).



Morse_Signal_Sampling.drawio

Alınan bit dizisi ("received_bit_sequence"), ana programda iletilen dizi yerine TXT'nin ekranında görüntülenir.

1a. Program Alıntısı (örnek):

```

define receive_bit_sequence
  repeat forever
    do if is photo transistor TXT_I1 - bright -
      do wait ms - 0.6 x - dit -
        set LED TXT_O2 - on -
        to received_bit_sequence - append text - receive_bit
  
```

```

define receive_bit
  wait ms - dit -
  if is photo transistor TXT_I1 - dark -
    do set received_bit - to - "0"
  else set received_bit - to - "1"
  wait ms - 2 x - dit -
  set LED TXT_O2 - off -
  return received_bit
  
```

Bit_Sequence_Sender_Receiver.ft

1b. İletim, “dit” uzunluğu = 90 ms olduğunda hâlâ hatasız bir şekilde tamamlanabilir. Bu, yaklaşık 11 bit/s hızına karşılık gelir.

Karşılaştırma için: Günümüzde bir DSL bağlantısı 100 Mbit/s hıza ulaşabilir, bu da yaklaşık olarak bu hızdan 9.000.000 kat daha hızlıdır.

2. Mors Kodu Alıcısı

Örnek programdaki alıcı işlevi üç bölüme ayrılmıştır:

- Tek bir bitin alınması (Deneysel Görev 1'deki gibi)
- Bir Mors kodunun alınması
- Bir mesajın alınması (thread olarak başlatılır)

2a. Program Alıntısı (örnek):

```

define receive_message
repeat forever
do if is photo transistor TXT_I1 bright
do wait ms 0.6 x dit
set receive_index to in list morse_alphabet find first occurrence of item receive_symbol
to received_message append text in list alphabet get # receive_index

```

```

define receive_symbol
set received_symbol to ""
repeat while is photo transistor TXT_I1 bright
do to received_symbol append text receive_bit
wait ms dit
return received_symbol

```

```

define receive_bit
wait ms dit
if is photo transistor TXT_I1 dark
do set received_bit to "0"
else set received_bit to "1"
wait ms 2 x dit
return received_bit

```

Morse_Code_Sender_Receiver.ft

2c. Örnek programda “dit” uzunluğu = 70 ms olduğunda alım hâlâ güvenilirdir. Bu nedenle, maksimum iletim hızı yaklaşık 14 “dit” saniye başındır.

2d. Referans kelime “Paris” 50 “dit”ten oluşur. Bu nedenle, “dit” = 100 ms olduğunda, Mors vericimiz 12 WpM hızına ulaşabilir. Bu, amatör radyo operatörlerinin testlerinde başarması gereken Mors hızına karşılık gelir. Rekor ise 88 WpM’dir.

4. Şifreleme – Sezar Şifresi

Sezar şifresini kullanarak şifreleme, listedeki “Mors alfabesi” öğelerindeki karakterleri uygun sayıda harf ile “kaydırarak” programlamak oldukça kolaydır:

- Şifreleme sırasında +3 karakter,
- Şifre çözme sırasında -3 karakter ile kaydırılır.

Not: Alfabenin ve Mors alfabesinin liste öğeleri için indeksler 1’den başlayarak toplam öğe sayısına (burada 26) kadar çalışır. Anahtarın eklenmesi veya çıkarılmasından sonra, bölümün kalan değeri 26 ile ayarlanmalıdır. Eğer indeks sonucu = 0 olursa, indeks maksimum değer (26) olarak ayarlanmalıdır.

Çözüm Varyantları: Anahtar (alfabenin “kaydırıldığı” harf sayısı), ekrandaki giriş üzerinden değişken olarak seçilebilir.

Program Alıntısı (örnek): Şifreleme:

```

...
set send_message to to UPPER CASE " thequickbrownfoxjumpsoverthelazydog "
set send_character_counter to 1
set key to 3
execute function receive_message in a thread
repeat length of send_message times
do
set send_character to in text send_message get letter # send_character_counter
set send_index to remainder of
in list alphabet
find first occurrence of item send_character
+ key
+ 26
+ if send_index = 0
do set send_index to 26
set send_morse_symbol to in list morse_alphabet get # send_index
send_morse_symbol with:
morse_symbol send_morse_symbol
change send_character_counter by 1
set label message text + create text with " M: "
in text received_message
get substring from letter # from end 15
to last letter
wait s 10

```

Şifre Çözme:


```

define receive_message
repeat forever
do
+ if is photo transistor TXT_11 bright
do
wait ms 0.6 x dit
set receive_index to remainder of remainder of in list morse_alphabet
find first occurrence of item receive_symbol
- - key
+ if receive_index = 0
do set receive_index to 26
to received_message append text in list alphabet get # receive_index

```

Morse_Code_Sender_Receiver_with_Caesar_Encryption.ft

5. Şifreleme – Vigenère Şifresi

Program Alıntısı (örnek):

Şifreleme:

```

...
set send_message to to UPPER CASE "thequickbrownfoxjumpsoverthelazydog"
set send_character_counter to 1
set key to to UPPER CASE "fischer"
execute function receive_message in a thread
repeat length of send_message times
do
set send_character to in text send_message get letter # send_character_counter
set encrypt_index to remainder of send_character_counter ÷ length of key
+ if encrypt_index = 0
do set encrypt_index to length of key
set encrypt_char to in text key get letter # encrypt_index
set encrypt_offset to in list alphabet
find first occurrence of item encrypt_char
set send_index to remainder of remainder of in list alphabet
find first occurrence of item send_character
+ + encrypt_offset
+ if send_index = 0
do set send_index to 26
set send_morse_symbol to in list morse_alphabet get # send_index
send_morse_symbol with:
morse_symbol send_morse_symbol
change send_character_counter by 1
set label message text + - create text with "M:"
in text
get substring from letter # from end 15
to last letter
wait s 10

```

Şifre çözme için geçerli anahtar sembolünü belirlemek amacıyla başka değişkenler kullanılmalıdır, çünkü şifre çözme ve şifreleme eşzamanlı olarak gerçekleşmektedir.

Şifre Çözme:

```

define receive_message
  set decrypt_index to 0
  repeat forever
    do
      if is photo transistor TXT_I1 bright
        do
          wait ms 0.6 x dit
          change decrypt_index by 1
          if decrypt_index > length of key
            do set decrypt_index to 1
          set decrypt_char to in text key get letter # decrypt_index
          set decrypt_offset to in list alphabet find first occurrence of item decrypt_char
          set receive_index to remainder of in list morse_alphabet find first occurrence of item receive_symbol - decrypt_offset 26
          if receive_index = 0
            do set receive_index to 26
          to received_message append text in list alphabet get # receive_index
  
```

Morse_Code_Sender_Receiver_with_Vigenere_Encryption.ft

Ek

Gerekli Malzemeler

- PC: Program geliştirme için, yerel veya web arayüzü üzerinden.
- USB Kablosu / BLE / WiFi: Programı TXT4.0'a aktarmak için.
- Program Şablonu: Mors kodu için **Morse_Code_Template.ft**.

Ek Bilgiler

[1] Albrecht Beutelspacher: *Kryptologie: Eine Einführung in die Wissenschaft vom Verschlüsseln, Verbergen und Verheimlichen*. 10. Baskı, Springer Verlag, 2015.

[2] Simon Singh: *Codes. Die Kunst der Verschlüsselung*. Impian, 2021.

[3] Durum diyagramları oluşturmak için çevrimiçi diyagram düzenleyici (drawio formatında).

<https://www.diagrammeditor.de/>