



**T.C.
ERCIYES ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
MEKATRONİK MÜHENDİSLİĞİ BÖLÜMÜ**

MEKATRONİK LABORATUVARI – I

**STM32F407 Mikroişlemci Programlama
&
Matlab / Simulink Uygulamaları**

**DENEY SORUMLUSU
Arş. Gör. Ahmet Hüsrev AKDENİZ**

**EKİM 2026
KAYSERİ**

STM32F407 Discovery Kartının Matlab Simulink Ortamında Waijung Bloksetleri İle Programlanması Deneyi

https://drive.google.com/drive/folders/1vuKJO8i3MICdvzKNhaSbVc_xbiVWvvCu?usp=sharing



1. Giriş

Mekatronik sistemler; duyucular (sensörler) ile çevresel değişimleri algılayan, elde ettiği verileri işleyen ve eyleyiciler (motor sistemleri vb.) yardımıyla algoritmik denetim sağlayan tümleşik donanım ve yazılım yapılarıdır. Bu denetimi gerçekleştiren gömülü sistemler, makine veya cihazın içerisine entegre edilmiş, belirli bir görevi yerine getirmek üzere tasarlanmış özel amaçlı bilgisayarlardır. Günümüzde endüstriyel mekatronik uygulamalarının karmaşıklığı arttıkça, klasik denetleyiciler yerine yüksek hızlı ve çok yönlü donanımlara ihtiyaç duyulmaktadır. Bu laboratuvar kapsamında incelenecek olan STM32F407; 168 MHz saat frekansında çalışan 32-bit ARM Cortex-M4 çekirdeğine sahip, özellikle sayısal sinyal işleme (DSP) yeteneği ve donanımsal zenginliği sayesinde birden fazla alt sistem içeren karmaşık gömülü mimarilerin ana kontrol birimi olarak yaygın şekilde kullanılan endüstriyel bir

mikroişlemcidir. Sahip olduğu bu yüksek performans, esneklik ve güvenilirlik sayesinde STM32 tabanlı sistemler; başta savunma sanayi, insansız hava araçları (İHA), otomotiv elektroniği ve gelişmiş tıbbi cihaz teknolojileri olmak üzere günümüzün birçok kritik sektöründe etkin bir şekilde kullanılmaktadır.

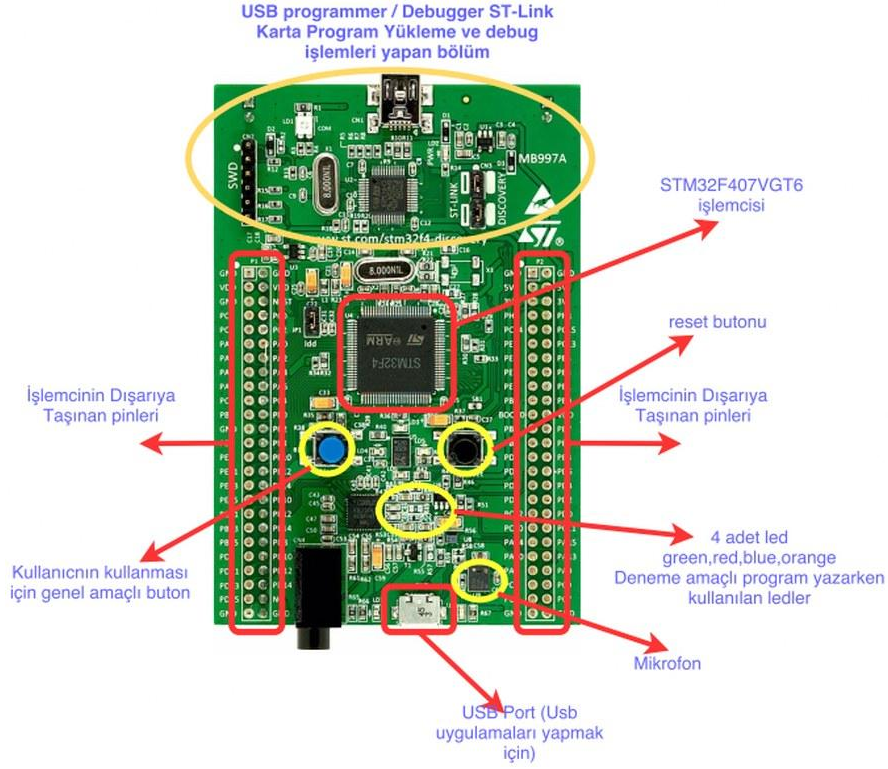
2. Bu Deneyin Amacı

Bu deneyin temel amacı, gelişmiş mekatronik ve gömülü sistem tasarımlarında sıkça tercih edilen yüksek işlem kapasiteli STM32F407 (ARM Cortex-M4) mikroişlemcisinin donanım mimarisini ve çevresel birimlerini tanıtmaktır. Laboratuvar çalışması kapsamında, geleneksel donanım programlama yöntemlerinin ötesine geçilerek, modern mühendislik projelerinde hız ve esneklik sağlayan model tabanlı (model-based) tasarım yaklaşımının kavratılması hedeflenmektedir. Bu doğrultuda; MATLAB Simulink ortamı ve Waijung bloksetleri kullanılarak gerçek zamanlı kontrol algoritmalarının nasıl modelleneceği ve doğrudan donanım üzerine nasıl gömüleceği uygulamalı olarak gösterilecektir. Deney sonucunda öğrencilerin; ADC (Analog-Dijital Çevirici), PWM, dijital giriş/çıkış ve çeşitli haberleşme protokollerini (UART, I2C vb.) kullanarak sensör okuma, motor kontrolü ve veri işleme gibi temel ayrık zamanlı gömülü sistem uygulamalarını pratik bir şekilde gerçekleştirebilme becerisi kazanması amaçlanmaktadır.

3. STM32F407 Nedir?

STM32F407, 168 MHz saat frekansında çalışan, 32-bit ARM tabanlı bir mikroişlemcisidir. Yüksek işlem gücü ve donanımsal özellikleri sayesinde özellikle sayısal sinyal işleme (DSP) uygulamalarında ve birden fazla alt sistem içeren karmaşık yapıların ana kontrol birimi olarak yaygın şekilde kullanılmaktadır. Bu “leyici, STM32F4 serisine ait Cortex-M4 çekirdeğini barındırmaktadır.

STM32F407 Discovery geliştirme kartı üzerinde; 3 eksenli ivmeölçer, dahili ses girişi, bu girişe bağlı dijital-analog dönüştürücü (DAC), 4 adet programlanabilir LED ve 2 adet kullanıcı butonu bulunmaktadır. Bu donanım bileşenleri, kartın farklı gömülü sistem uygulamalarında kullanılmasına olanak sağlamaktadır.



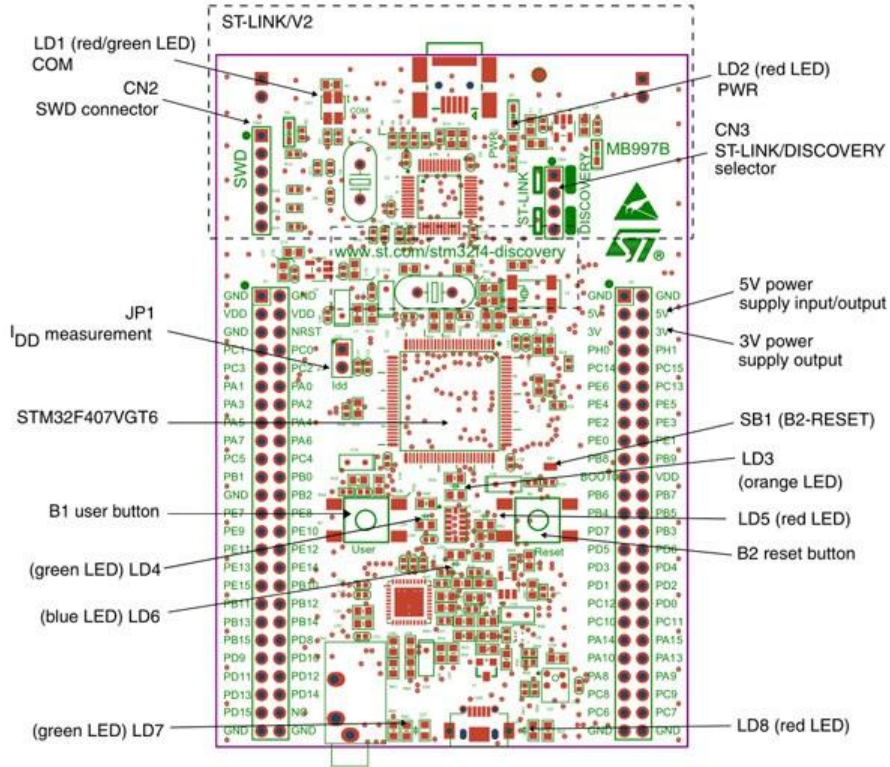
STM32F407 Genel Bakış

Kart üzerinde yer alan ve görselde kırmızı daire ile işaretlenen bölüm, geliştirme kartının kendi mikroişlemci çekirdeğini ve programlama/debug işlemleri için kullanılan ST-LINK/V2 birimini içermektedir. Bu bölümde bulunan SWD (Serial Wire Debug) pinleri sayesinde yalnızca kart üzerindeki STM32F407 mikroişlemcisi değil, aynı zamanda harici ARM mimarisine sahip diğer mikroişlemciler de programlanabilmekte ve hata ayıklama işlemleri gerçekleştirilebilmektedir.

STM32F407 Teknik Özellikleri

- 32-bit ARM Cortex-M4F çekirdeğine sahip STM32F407VGT6 mikroişlemcisi, 1 MB Flash ve 192 KB RAM
- Dahili ST-LINK/V2 JTAG Debugger
- USB veya harici kaynaktan doğrudan 5v ile çalışabilme
- 3 V ve 5 V luk çıkış pinleri
- 3-eksen dijital ivmeölçer ([LIS302DL](#))
- Omni-Directional Mikrofon ([MP45DT02](#))
- D sınıfı yükselteçli ses sürücü çipi (CS43L22)
- Sekiz adet LED:
 - LD1 (kırmızı/yeşil)(USB haberleşmesi için)
 - 3v3 power on/off LEDi

- Dört adet kullanıcı LEDi, LD3 (turuncu), LD4 (yeşil), LD5 (kırmızı) ve LD6 (mavi)
- 2 adet USB OTG LEDi, LD7 (yeşil) ve LD8 (kırmızı)
- Bir adet Reset ve bir adet kullanıcı tanımlı buton
- USB OTG için mikro-AB konnektör.
- 100 pin'in tamamını kullanabilmeye imkan tanıyan çıkışlar.



STM32F407 Şematığı

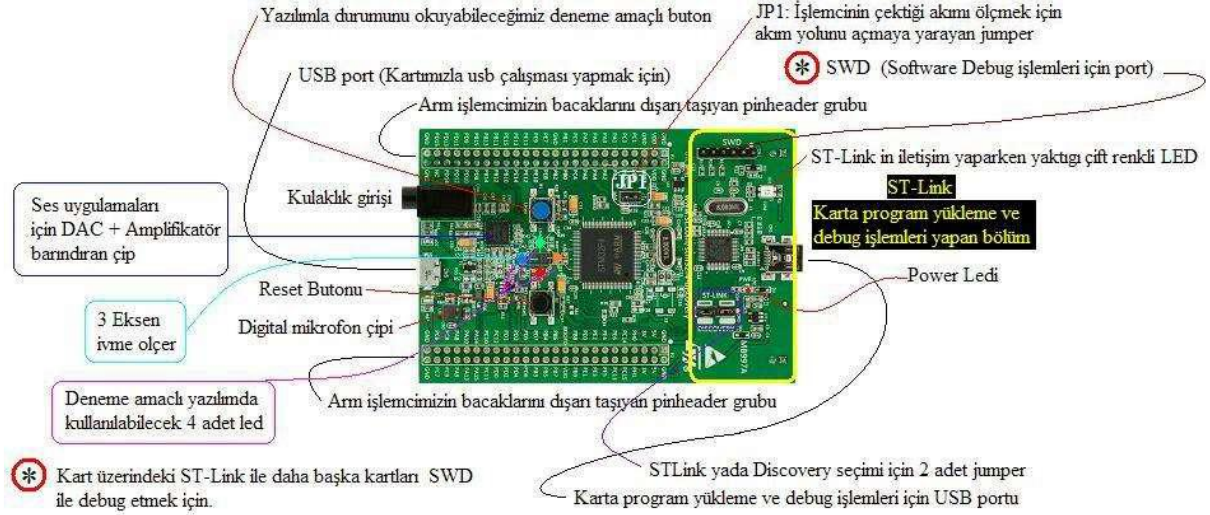
STM32F407 Discovery geliştirme kartı kullanılarak birçok farklı gömülü sistem projesi gerçekleştirilebilmektedir. Bu noktada, gömülü sistem kavramının ne anlama geldiğinin açıklanması gerekmektedir.

Gömülü sistem; belirli bir görevi veya görevler bütününe yerine getirmek üzere tasarlanmış, genellikle daha büyük bir elektronik ya da mekanik sistemin içerisine entegre edilmiş, özel amaçlı bir bilgisayar sistemidir. Gömülü sistemler; günlük yaşamda yaygın olarak kullanılan beyaz eşya, otomotiv elektroniği, endüstriyel kontrol sistemleri, tıbbi cihazlar ve tüketici elektroniği gibi pek çok elektronik ürünün içerisinde yer almaktadır. Bu sistemler, cihazların çalışma mantığını yöneten ve halk arasında “beyin” olarak ifade edilen kontrol ve karar verme işlevlerini üstlenerek, sistemlerin akıllı ve otomatik bir şekilde çalışmasını sağlar.

Gömülü sistemlere örnek olarak asansör kontrol sistemleri, klima ve güvenlik izleme sistemleri gibi ev otomasyonu ürünleri; hesap makineleri, mikrodalga fırınlar, çamaşır makineleri, televizyonlar ve medya oynatıcıları gibi ev elektroniği cihazları verilebilir. Temel olarak bir gömülü sistem, kendisine önceden tanımlanmış ve özel olarak belirlenmiş görevleri yerine getirmek üzere tasarlanır.

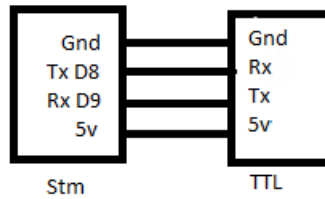
4. STM32F407 Discovery Kartını Programlama

STM32 mikroişlemcileri, **ARM tabanlı** bir mimariye sahiptir. ARM tabanlı mikroişlemcilerin programlanması sürecinde, geliştirme ortamlarının ve derleyicilerin kullanımı, diğer bazı mikroişlemci platformlarına kıyasla daha karmaşık olabilmektedir. Bu durum, kullanılan donanım mimarisi, çevre birimlerinin çeşitliliği ve yazılım yapılandırmalarının ayrıntılı olmasıyla ilişkilidir. StmCubeMX , Stm Studio ve Matlab Simulink gibi derleyicilerde programlanabilmektedir.



EK BİLGİ

STM32F4 kartının bağlantı ve debug için gerekli port üzerinde mevcuttur. Ancak bazı stm32 kartlarını programlarken ttl dönüştürücüye ihtiyaç olabilir. Bu noktada usb – ttl bağlantısını gösteren şema şu şekildedir:



USB – TTL bağlantısı

4.1 Matlab Simulink – Waijung Kurulum

İlk olarak Stm32F4 geliştirme kartını bilgisayarın tanıyabilmesi için ST-LINK kurulumu yapılmalıdır.

STM32F4 DISCOVERY driver kurulum linki;

<https://www.st.com/en/evaluation-tools/stm32f4discovery.html>

STLINK Kurulum Linki

<https://www.st.com/en/development-tools/stsw-link009.html>

STLINK Güncelleme

<https://www.st.com/en/development-tools/stsw-link007.html>

Matlab Simulink ortamında waijung bloksetlerini kullanabilmemiz için waijung-1 indirilip kurulumu yapılmalıdır. (Waijung-1 ücretsizdir, web sistesine üyelik gerekmektedir)

WAJUNG Kurulum;

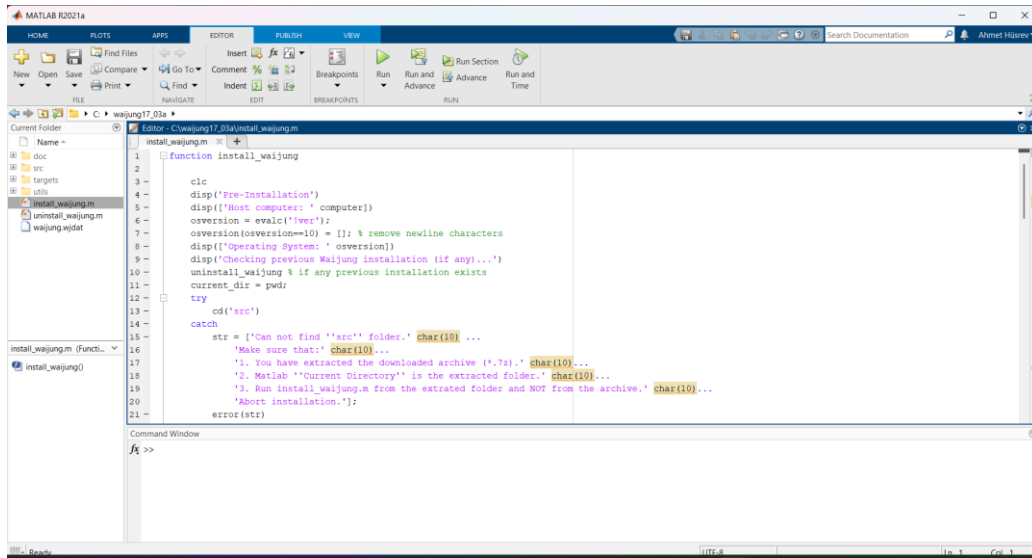
<https://www.aimagin.com/en/download>

NOT: İndirme linklerindeki setup ları indirebilmek için sisteme üye olup giriş yapılması gerekmektedir. Ardından indirme işlemini başlatabilirsiniz. Ayrıca matlab 'in yeni sürümlerinde simulink ile waijung hatası alınmakta. Bunun için R2021 ve öncesi matlab sürümler için uygun olabilir.

NOT'2: Bütün indirme linklerinde indirilen kurulum dosyaları içerisinde "README" dosyası olup kurulumun nasıl yapılacağı tarif edilmiştir. Linklerdeki setup dosyaları eski sürüm olabilir. Aynı sayfada bulunan güncel kurulum dosyalarını indirebilirsiniz.

4.2 Simulink Ortamında Waijung

Waijung kurulumunda dikkat edilmesi gereken husus, kurulum dosyasının uzantısı en kısa olacak şekilde bilgisayara kopyalanmasıdır. Şöyle ki;

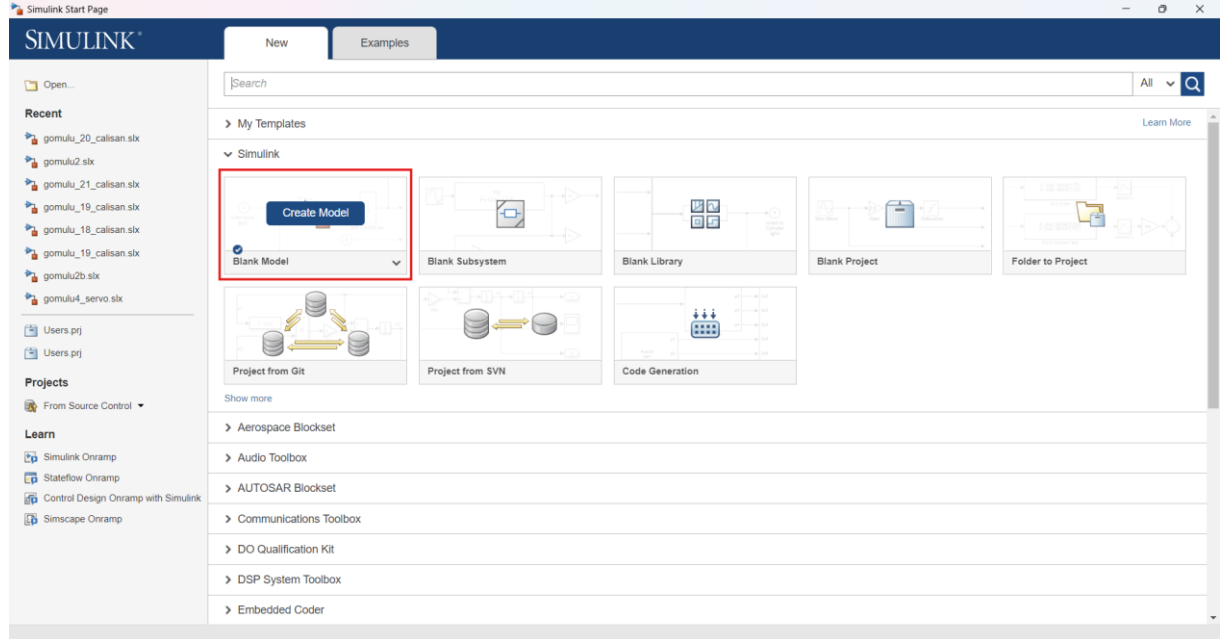


Yukarıda görülen resimde waijung17 kurulum dosyasının uzantısı "C:\waijung17_03a" şeklinde en kısa yol olacak haliyle ayarlanmıştır ve sol tarafta klasör ağacı görülmektedir. Bu aşamada **install_waijung.m** dosyasına sağ tıklayarak Run komutuna basmalıyız. Waijung kurulumu yapacaktır.

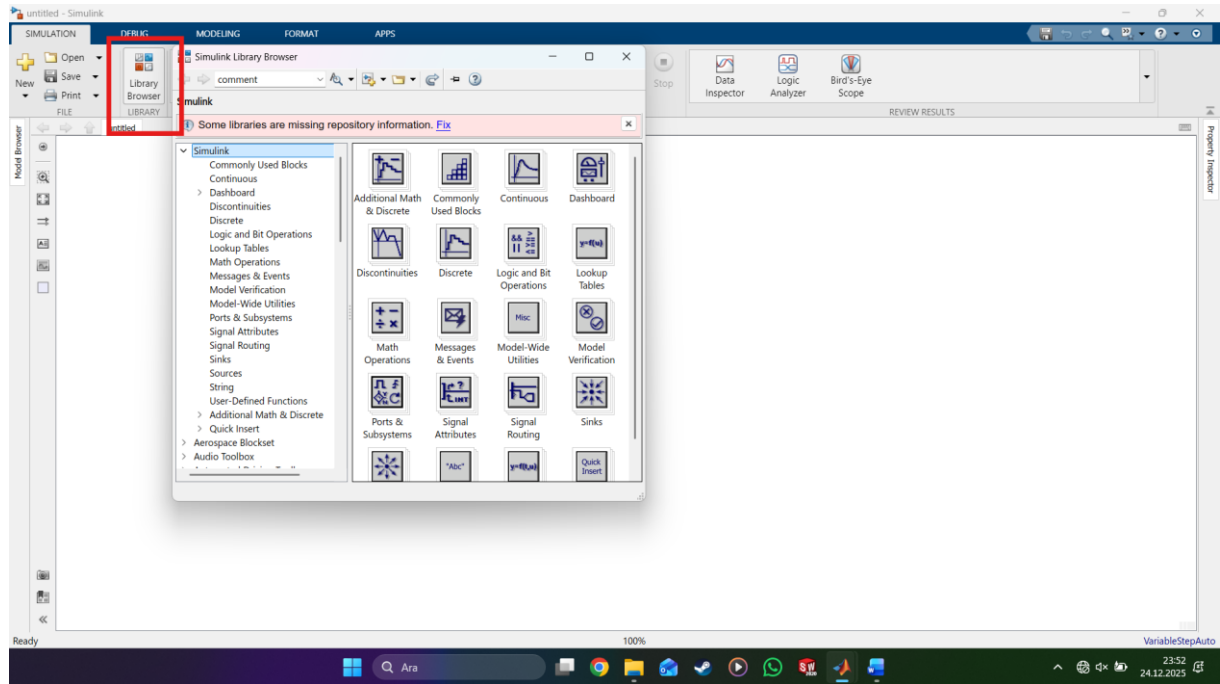
NOT: Matlab 'i yönetici olarak çalıştırınız.

NOT'2: Kullandığınız matlab sürümüne göre waijung18 de kurulumu yapılabilir.

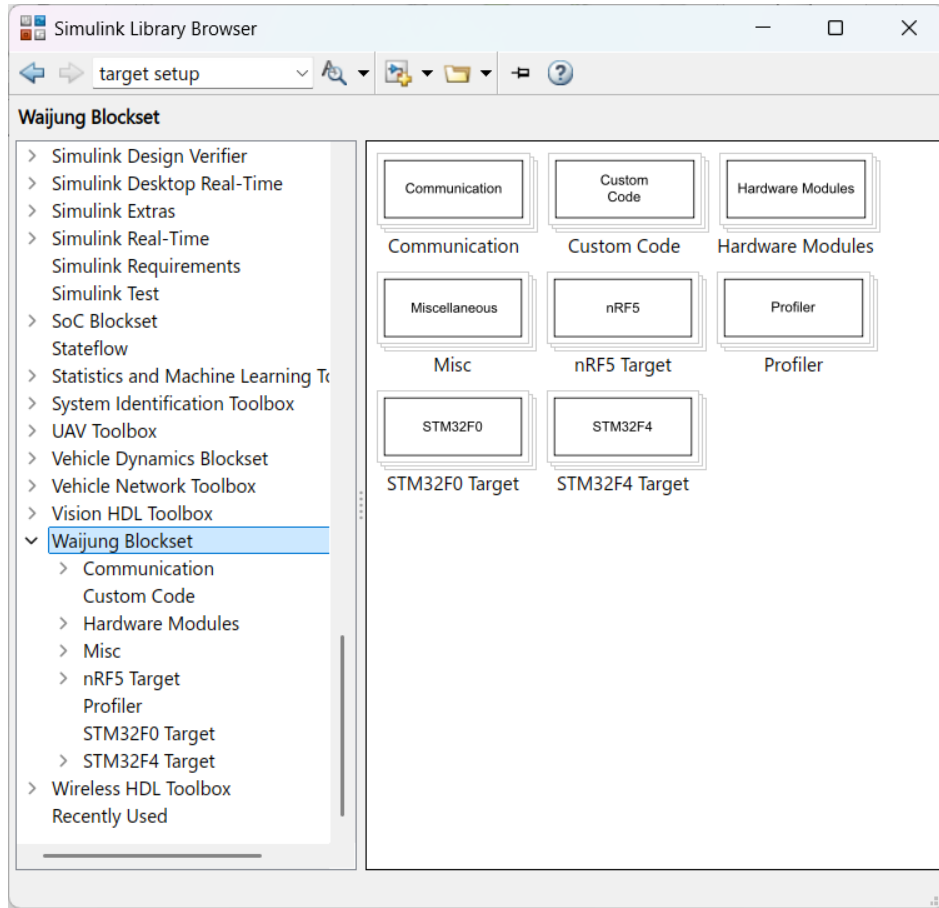
Bu aşamadan sonra Simulink tarafına geçiyoruz:



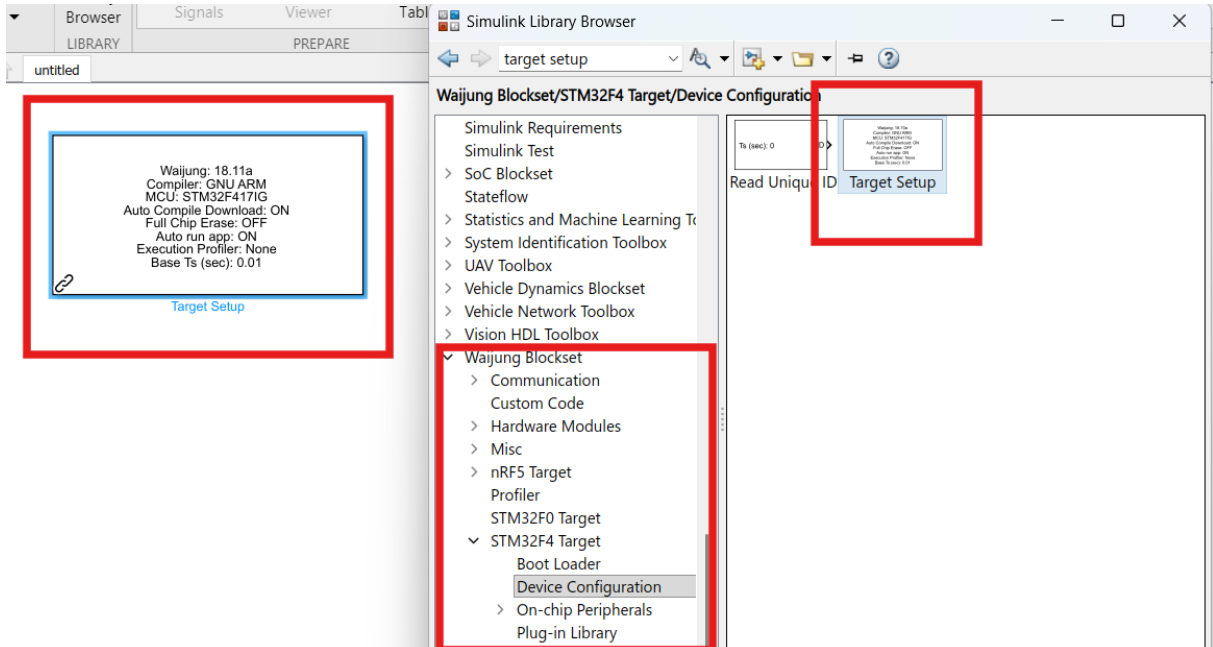
Yukarıda simulink menüsünün giriş sayfası görülmekte. Kırmızı kutuyla işaretli boş bir simulink modeli açalım.



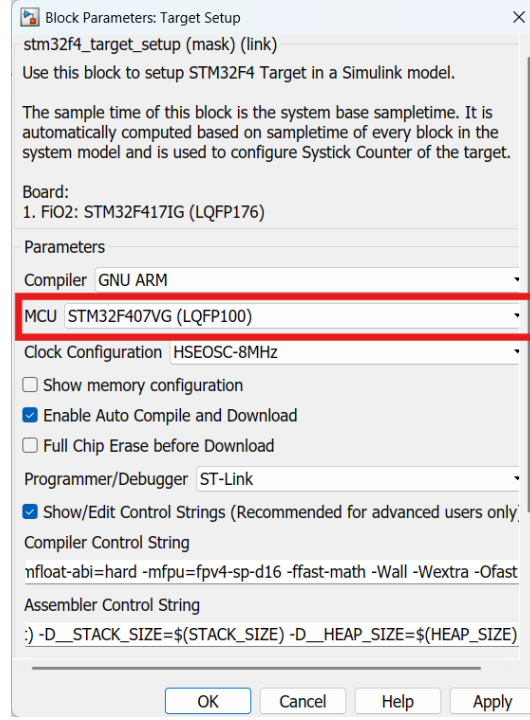
Açılan pencerede Library Browser adı verilen toolbox açalım (bir üstteki resim). Önümüze aşağıda görüldüğü gibi çeşitli uygulamalar için kütüphaneler mevcut. Waijung kurulumunun başarılı olduğunu anlamak için en alt tarafta waijung bloksetlerini görmeliyiz:



Bütün STM32F4 kartlarını simulink üzerinden waijung bloksetleri ile gömülü olarak prorgamlarken “TARGET SETUP” kullanılmalı ve kartın özelliklerine göre ayarlanmalıdır.



Burada target setup bloğunu boş simulink ekranına sürükleyip bırak yapıyoruz ardından target setup bloğuna çift tıklıyoruz:

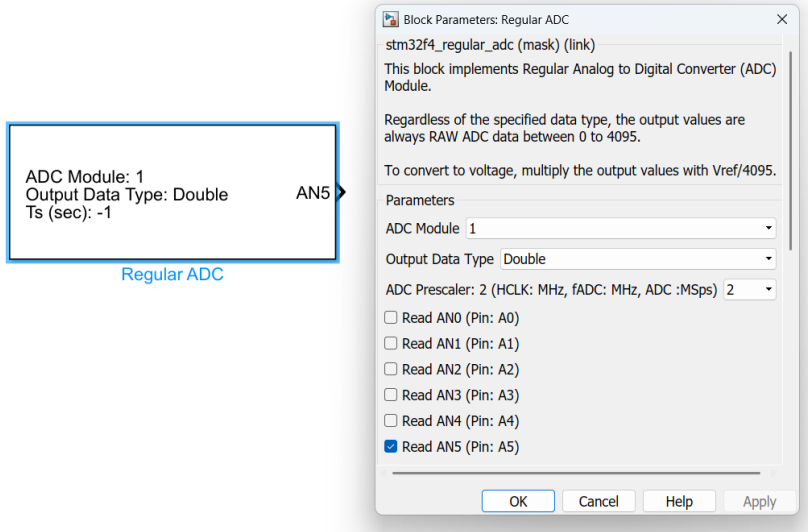


Bu blokta yalnızca program atacağımız kartın modelini seçiyoruz ve uygulayarak çıkıyoruz. (Stm32F4 serisi)

4.2.1 Bloksetler

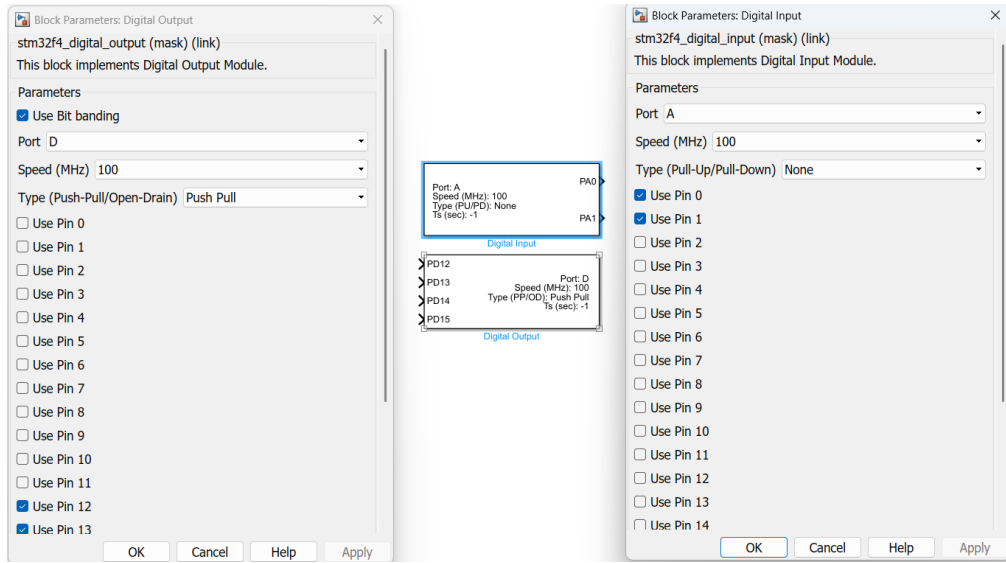
Simulink ortamında sıklıkla kullanılan bloklar şunlardır:

A. Regular ADC



Bu blok, dış dünyadan gelen analog voltaj sinyallerini (örneğin 0V - 3.3V arası) mikroişlemcinin işleyebileceği dijital değerlere dönüştürür. Seçilen pindeki voltajı okur ve bunu 0 ile 4095 arasında bir sayısal veriye çevirerek Simulink ortamına aktarır. Genellikle sıcaklık ve ışık gibi analog sensörlerden veri toplamak, potansiyometre aracılığıyla sistem parametrelerini manuel kontrol etmek veya bir devredeki voltaj seviyelerini anlık olarak izlemek amacıyla kullanılır.

B. Digital Input – Output



- Digital Input (Dijital Giriş) Bloğu

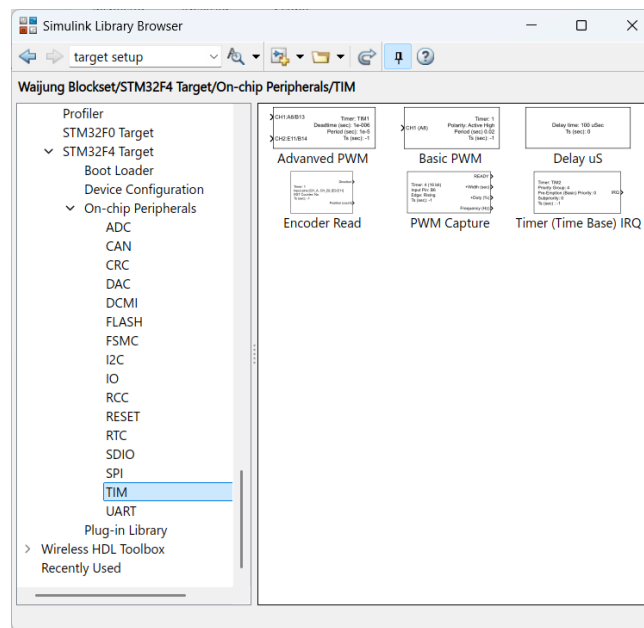
Belirlenen pinlerdeki voltaj seviyesini anlık olarak takip ederek, dışarıdan gelen dijital bilgilerin (Lojik 0 veya 1) mikroişlemci tarafından okunmasını sağlar. Bir butona basılıp basılmadığını algılamak, sınır anahtarlarından (limit switch) konum bilgisi almak veya dijital çıkış veren sensörlerin durumunu kontrol etmek için kullanılır.

- Digital Output (Dijital Çıkış) Bloğu

Mikroişlemcinin seçilen pinlerinden dış dünyaya dijital sinyal (Lojik 0 veya 1) gönderilmesini sağlar; yani pindeki voltajı tamamen keser veya tam seviyeye çıkarır. LED'leri yakıp söndürmek, röleleri tetiklemek veya diğer dijital cihazlara "aç/kapat" şeklinde kontrol sinyalleri göndermek amacıyla kullanılır.

C. Timer (zamanlayıcı)

Burada birden fazla çeşitte blok var. Uygulamaya göre birkaç çeşide ayrılır.



En çok kullanılan blok genellikle PWM bloklarıdır. Bunun yanında encoder read bloğuda motor hareket ve hızlarını okumada kullanılır. Tek bir blokta 4 kanaldan ayrı ayrı pwm üretilebilir.

- **Basic PWM Bloğu**

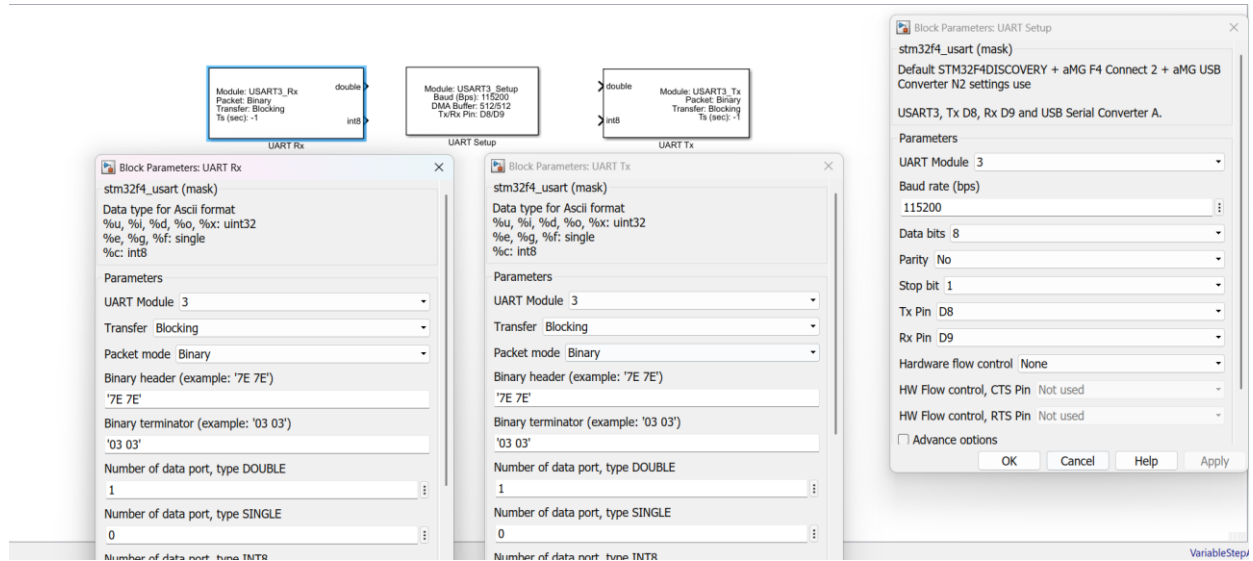
Mikroişlemcinin zamanlayıcılarını kullanarak belirli bir frekansta ve ayarlanabilir doluluk oranında (örneğin Duty Cycle) kare dalga üretir.

Parametre Özellikleri:

- **Timer:** PWM üretimi için kullanılacak donanımsal zamanlayıcıyı (Örn: Timer 1) seçer.
- **PWM Period (seconds):** Sinyalin periyodunu saniye cinsinden belirler (Örn: 0.02s periyot, 50Hz frekans sağlar).
- **Polarity:** Sinyalin "Active High" (aktifken yüksek) veya "Active Low" (aktifken düşük) olma durumunu ayarlar.
- **Channel:** PWM sinyalinin hangi kanaldan/pinden (Örn: A8) çıkacağını belirler.

DC motor hızı kontrol etmek, LED parlaklığı ayarlamak veya servo motor konumlandırmak için kullanılır.

D. UART Com.



Mikroişlemci ile bilgisayar arasında veri takibi yapmak, sensörlerden gelen seri verileri okumak veya iki farklı mikroişlemciyi birbiriyle konuşturmak amacıyla kullanılır.

- **UART Setup (Kurulum) Bloğu**

Mikroişlemcinin donanımsal UART birimini yapılandırarak haberleşmenin temel kurallarını (hız, pinler vb.) belirler.

- **UART Module:** Hangi fiziksel birimin (USART1, 2, 3 vb.) kullanılacağını seçer.
- **Baud rate (bps):** Haberleşme hızını belirler (Örn: 115200, 9600 vb.). Her iki cihazın da aynı hızda olması gerekir.

- **Data bits / Parity / Stop bit:** Veri paketinin formatını ve hata kontrol ayarlarını belirler.
- **Tx / Rx Pin:** Verinin gönderileceği (Tx) ve alınacağı (Rx) fiziksel pinleri (Örn: D8/D9) tanımlar.

UART Tx (Veri Gönderme) Bloğu

Mikroişlemci içerisindeki verileri, belirlenen formatta dışarıdaki bir cihaza (bilgisayar, Bluetooth modülü vb.) gönderir.

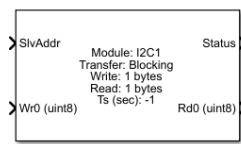
- **Transfer:** Veri iletiminin "Blocking" (işlem bitene kadar bekle) veya "Non-blocking" (arka planda gönder) olacağını belirler.
- **Packet mode:** Verinin "Binary" (saf veri) veya "Ascii" (okunabilir metin) olarak gönderilmesini sağlar.
- **Header / Terminator:** Veri paketinin başlangıç ('7E 7E') ve bitiş ('03 03') işaretlerini ekleyerek verinin karşı tarafta doğru tanınmasını sağlar.

UART Rx (Veri Alma) Bloğu

Dış dünyadan UART hattı üzerinden gelen verileri okuyarak Simulink ortamına aktarır.

- **Number of data port:** Alınacak verilerin tipini ve sayısını (Double, Single veya Int8) belirler. Bu sayede karmaşık veri paketleri parçalanarak okunabilir.
- **Packet mode:** Gelen verinin hangi formatta çözümleneceğini (Binary/Ascii) seçer.

E. I2C Bloğu



Block Parameters: I2C Master Setup

stm32f4_i2c (mask)

Parameters

I2C Module 1

Transfer mode Blocking

Duty cycle in fast mode TLow/THigh = 2

Clock speed (kHz) 400

Timeout (ms) 25

SDA pin B7

SCL pin B6

OK Cancel Help Apply

Block Parameters: I2C Master Read/Write

stm32f4_i2c (mask) (link)

SlvAddr:

- I2C device address, left justify (R/W bit append to bit 0).
- For I2C Read transaction, the block will adjust bit 0 of device address to '1' automatically.

Status:

- Output port represent the status of transaction.
- Status = 0, Success.
- Status = 1, Error NAK.
- Status = 2, Error Timeout.

Wait write time (ms):

- The block perform WRITE transaction then followed by READ transaction, some devices need delay time to update register.
- This parameter specify the number of ms to delay between WRITE and READ.
- Note: Set to 0 if not sure about this value.

Generate STOP before Re-Start for READ:

- To enable/disable STOP bit between WRITE and READ transaction (This option not apply when WRITE only or READ only).
- Option:

Checked: [S]<DevAddr+W>[A]<Wr0>[A]...<WrN>[A][P]<Wait write time>[S]<DevAddr+R>[A]<Rd0>[A]...<RdN>[NAK][P]

UnChecked: [S]<DevAddr+W>[A]<Wr0>[A]...<WrN>[A]<Wait write time>[S]<DevAddr+R>[A]<Rd0>[A]...<RdN>[NAK][P]

Parameters

I2C Module 1

Bytes write count

OK Cancel Help Apply

İvmeölçer, jiroskop, barometre gibi dijital sensörlerden veri okumak, harici EEPROM belleklerle çalışmak veya I2C destekli LCD ekranları kontrol etmek amacıyla kullanılır.

I2C Master Setup (Kurulum) Bloğu

Mikroişlemcinin I2C birimini "Master" (Efendi) modunda yapılandırarak, haberleşme hızı ve kullanılacak fiziksel pinler gibi temel protokol ayarlarını yapar.

- **I2C Module:** Kullanılacak donanım birimini seçer (Örn: I2C1).
- **Clock speed (kHz):** Haberleşme hızını belirler; görselde **400kHz** (Fast Mode) seçilmiştir.
- **SDA / SCL Pin:** Veri (SDA) ve saat (SCL) sinyalleri için kullanılacak fiziksel pinleri (Örn: B7/B6) tanımlar.

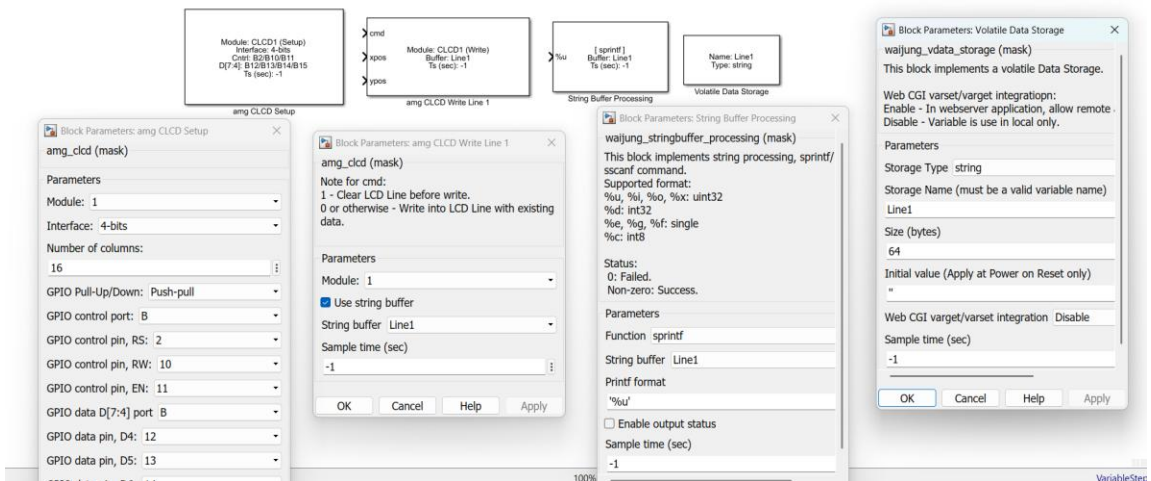
I2C Master Read/Write (Okuma/Yazma) Bloğu

Kurulumu yapılan I2C hattı üzerinden belirli bir adresteki cihazla veri alışverişini (veri gönderme veya alma) gerçekleştirir.

- **SlvAddr (Slave Address):** Haberleşilecek hedef cihazın benzersiz 7-bitlik donanım adresidir.
- **Status:** İşlemin sonucunu (Başarılı, Hata/NAK veya Zaman Aşımı) bir çıkış portu üzerinden bildirir.
- **Wait write time (ms):** Yazma işleminden sonra okuma işlemine geçmeden önce donanımın ihtiyaç duyduğu bekleme süresini ayarlar.
- **Generate STOP before Re-Start:** Okuma ve yazma işlemleri arasında "Dur" sinyali üretilip üretilmeyeceğini belirler.

F. LCD Blokları

Bu bloklar stm32f4 klasörü içerisinde değil. Waijung bloksetleri içerisinde hardware klasöründe bulunmaktadır.



Sensörlerden gelen verileri, sistem durumunu veya hata mesajlarını kullanıcıya anlık olarak fiziksel bir ekran üzerinden raporlamak amacıyla kullanılır.

amg CLCD Setup (Kurulum) Bloğu

Karakter tabanlı LCD ekranın fiziksel bağlantılarını, ekran boyutunu ve çalışma modunu tanımlayarak ekranı kullanıma hazırlar.

- **Interface:** Ekranın haberleşme modunu (örneğin **4-bit**) belirler.
 - **Number of columns:** Ekranın kaç sütunlu (örneğin **16** karakter) olduğunu ayarlar.
 - **GPIO control/data pins:** Ekranın kontrol (RS, RW, EN) ve veri (D4-D7) pinlerinin mikroişlemci üzerindeki hangi port ve pinlere bağlı olduğunu tanımlar.
-

amg CLCD Write Line (Satıra Yazma) Bloğu

Hazırlanan bir metin içeriğini LCD ekranın belirli bir satırına (Line 1 veya Line 2) yazdırmak için kullanılır.

- **Use string buffer:** Yazılacak verinin bir "String Buffer" (metin deposu) üzerinden alınıp alınmayacağını belirler.
 - **Sample time:** Bloğun veriyi ekrana ne kadar sürede bir güncelleyeceğini belirler.
-

String Buffer Processing (Metin İşleme) Bloğu

Mikroişlemciadaki sayısal verileri (örneğin bir sensör değeri), ekranın görüntüleyebileceği metin formatına (C dilindeki sprintf mantığıyla) dönüştürür.

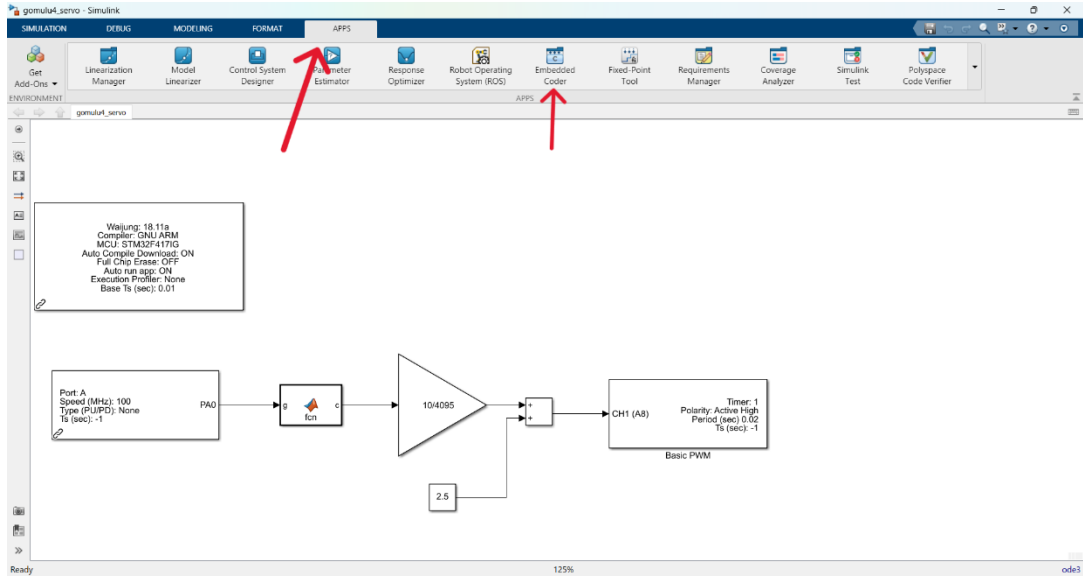
- **Function:** Yapılacak işlemi (örneğin **sprintf**) belirler.
 - **Printf format:** Verinin nasıl görüneceğini (örneğin %u ile tam sayı formatında) tanımlar.
-

Volatile Data Storage (Geçici Veri Deposu) Bloğu

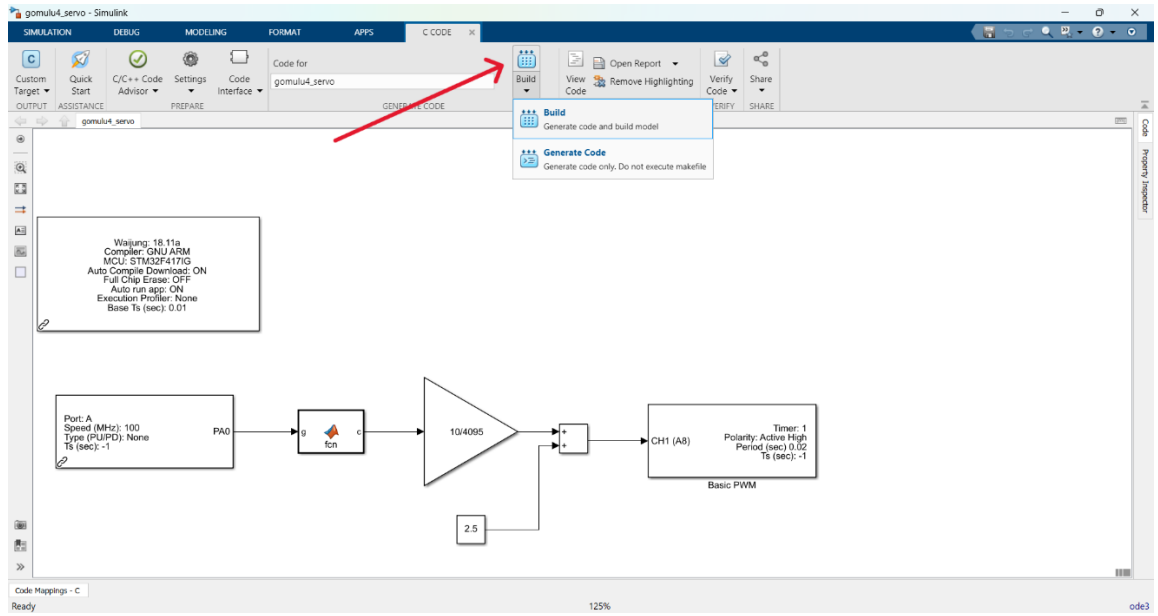
Ekrana yazdırılacak olan karakter dizilerinin (string) hafızada geçici olarak saklandığı bir tampon bölge oluşturur.

- **Storage Type / Name:** Saklanan verinin tipini (String) ve bu depo alanına verilen ismi (Örn: **Line1**) belirler.
- **Size (bytes):** Bu bellek alanının ne kadar büyük olacağını (örneğin **64 byte**) tanımlar.

Genel olarak bloklar ve özellikleri bu şekilde. Şimdi bu blokları kullanarak oluşturduğumuz uygulamayı stm32f4 kartına gömmek gerekiyor. Bunun için apps sekmesinden Embeded Coder uygulamasını açıyoruz.

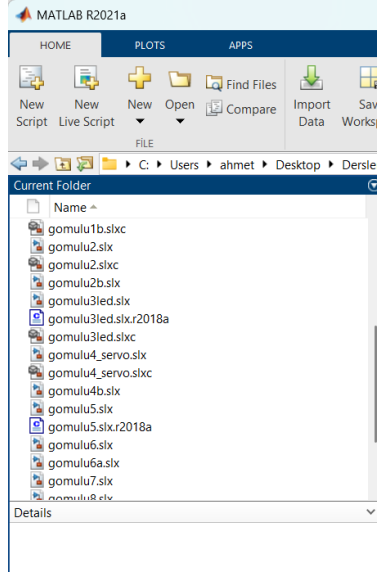


Daha sonra simulink modelini gömeceğimiz karta build edeceğiz.

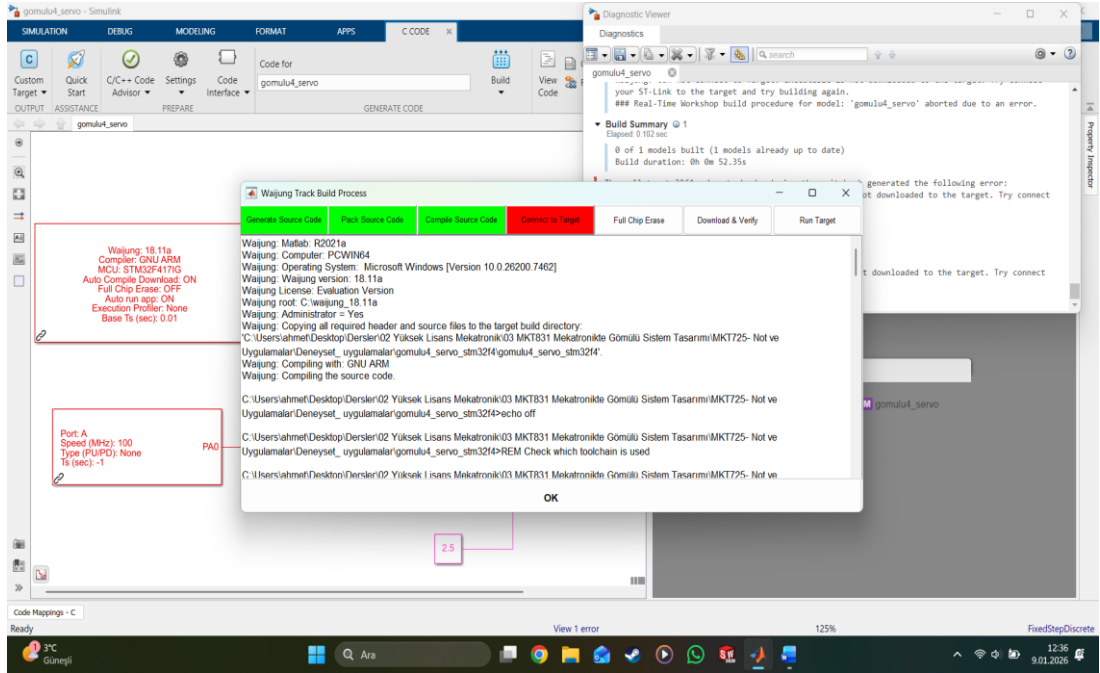


Bu nokta pratik olması bakımından “CTRL+B” tuşlarını kullanarakda build yapılabilir.

NOT: Burada dikkat edilmesi gereken nokta, build yapacağımız simulink modelinin matlab ekranında klasör ağacı veya uzantısının açık olması gerekmektedir. Yoksa program hata verir.



Aynen burada olduğu gibi çalışmanın uzantı ağacı matlab da açık olmalı. Bu kontrolü yaptıktan sonra build işlemi yapılır.

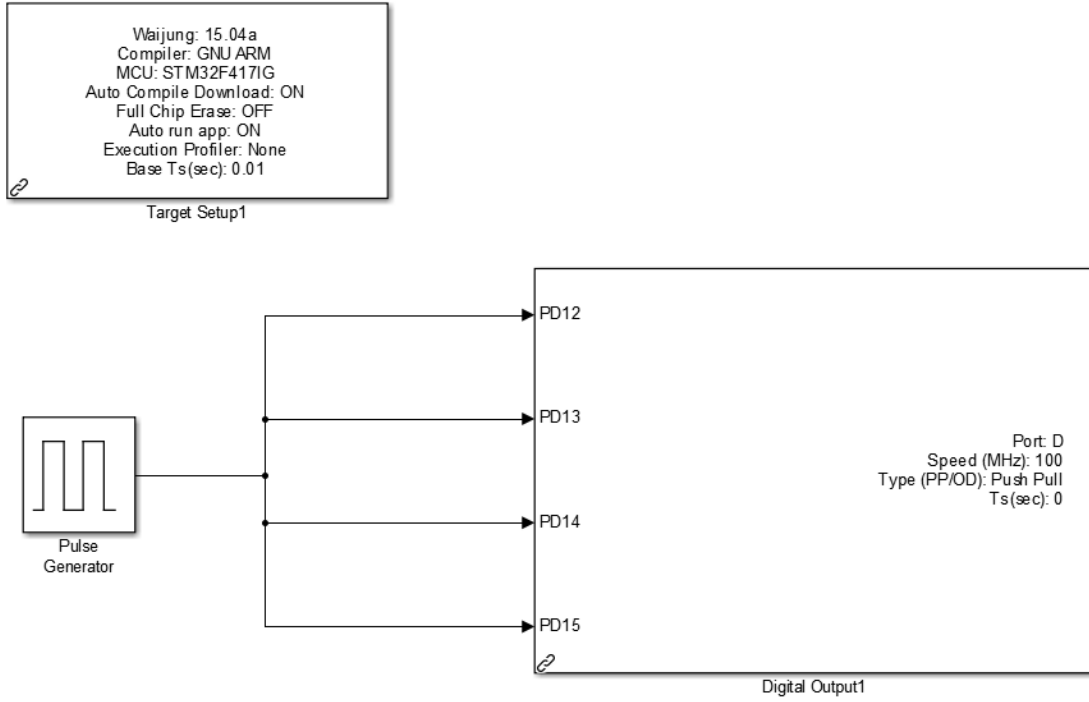


Yukarıdaki ekranda görünen “wajung Track build Process” penceresi yüklemenin durumunu gösterir. Soldan sağa doğru işlemler adım adım gerçekleşir ve tam gömülü kod yüklemesi için bütün işlemler yeşil olmalı. Sağ taraftaki “Diagnostic Viewer” penceresi; hata, uyarı veya bilgilendirme amaçlı olarak geribildiririm verir. Alt taraftaki bölümü ise oluşturulan simulink modelin C dilinde karşılık gelen hataları belirtir.

Bundan sonraki aşamalarda yapılacak uygulamalardan bahsedilecektir.

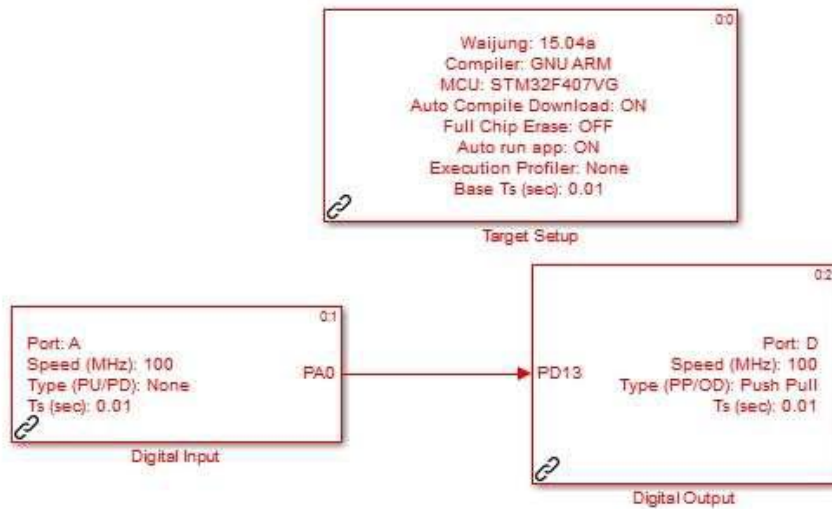
ÖRNEK DENEYSEL UYGULAMALAR

Uygulama 1.



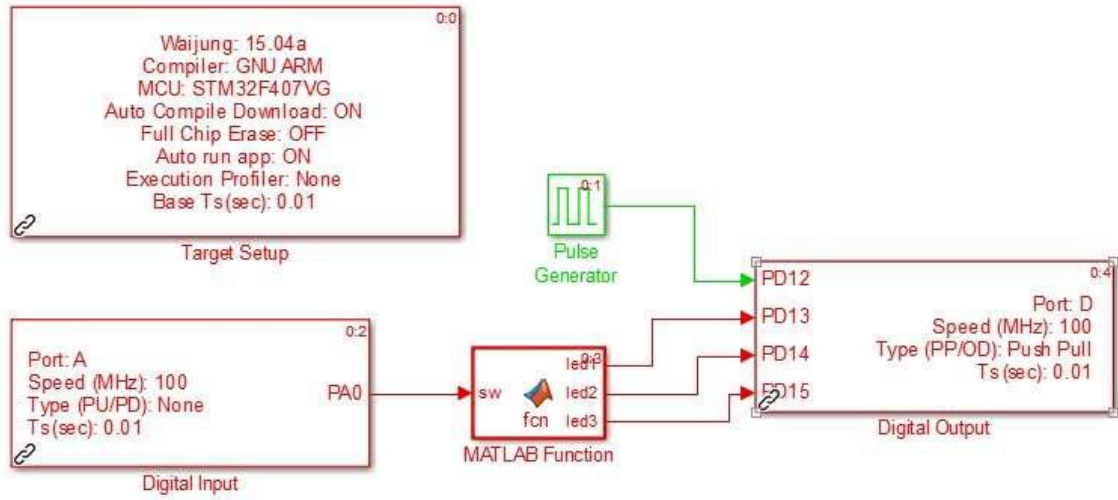
Bu uygulama ledi belirli bir periyotta yakıp söndürme uygulamasıdır. Pulse generator simülin kütüphanesinde mevcut arama çubuğundan aratılarak bulunabilir. Üzerine çift tıklayarak period ayarlanmaktadır.

Uygulama 2.



Buton ile led yakıp söndürme uygulamasıdır. Input ve output blokları kullanılarak 1 dijital giriş ile 1 dijital çıkış oluşturmak için kullanılır. (PA0, kart üzerinde kullanıcı tanımlı mavi renkli buton)

Uygulama 3.



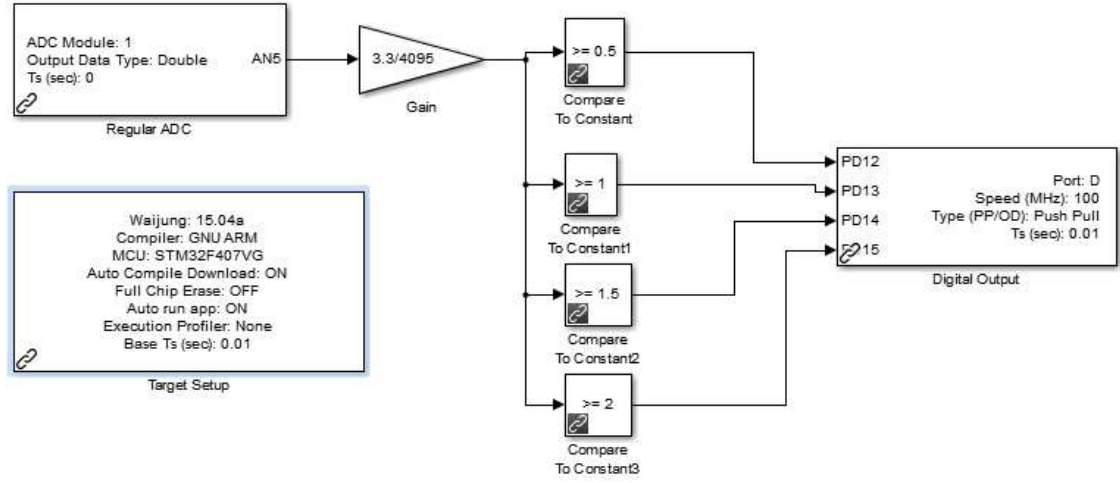
```

1 function [led1,led2,led3] = fcn(sw)
2 %#codegen
3 persistent count1;
4 persistent flag;
5 %persistent komutu deęişkenin hafızada tutulması
6 %saęlanıyor ve count1 deęişkenin bir bir artmasını
7 %saęlıyor
8 %persistent komutu sadece m-file ve m-function içinde
9 %kullanılır
10
11 if isempty(count1) %count1 boşsa 0 deęeri atanıyor
12     count1=0;
13 end
14 if isempty(flag) %flag boşsa 0 deęeri atanıyor
15     flag=0;
16 end
17 if (sw==1) && (flag==0)
18     count1 = count1 + 1;
19     flag = 1;
20     if count1 >= 3
21         count1 = 0;
22     end
23 end
24
25 end
26
27 if sw==0
28     flag=0;
29 end
30
31 if count1 == 0
32     led1 = 1;
33     led2 = 0;
34     led3 = 0;
35 elseif count1 == 1
36     led1 = 0;
37     led2 = 1;
38     led3 = 0;
39 elseif count1 == 2
40     led1 = 0;
41     led2 = 0;
42     led3 = 1;
43 else
44     led1 = 0;
45     led2 = 0;
46     led3 = 0;
47 end

```

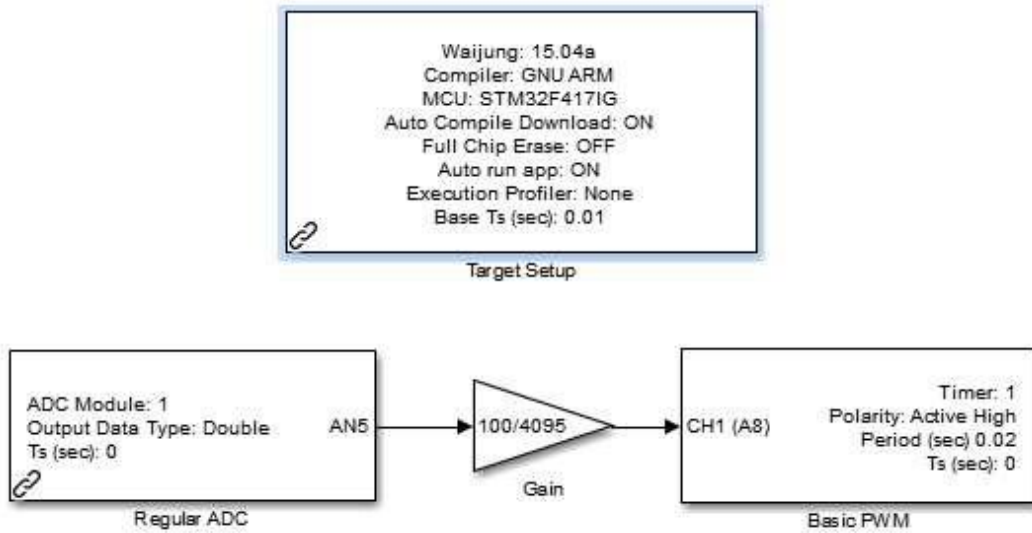
Puls generator kullanara bir adet dijital çıkış (led PD12) ve Matlab function blok kullanılarak buton ile diğer ledleri sırayla yakıp söndürme uygulamasıdır.

Uygulama 4.



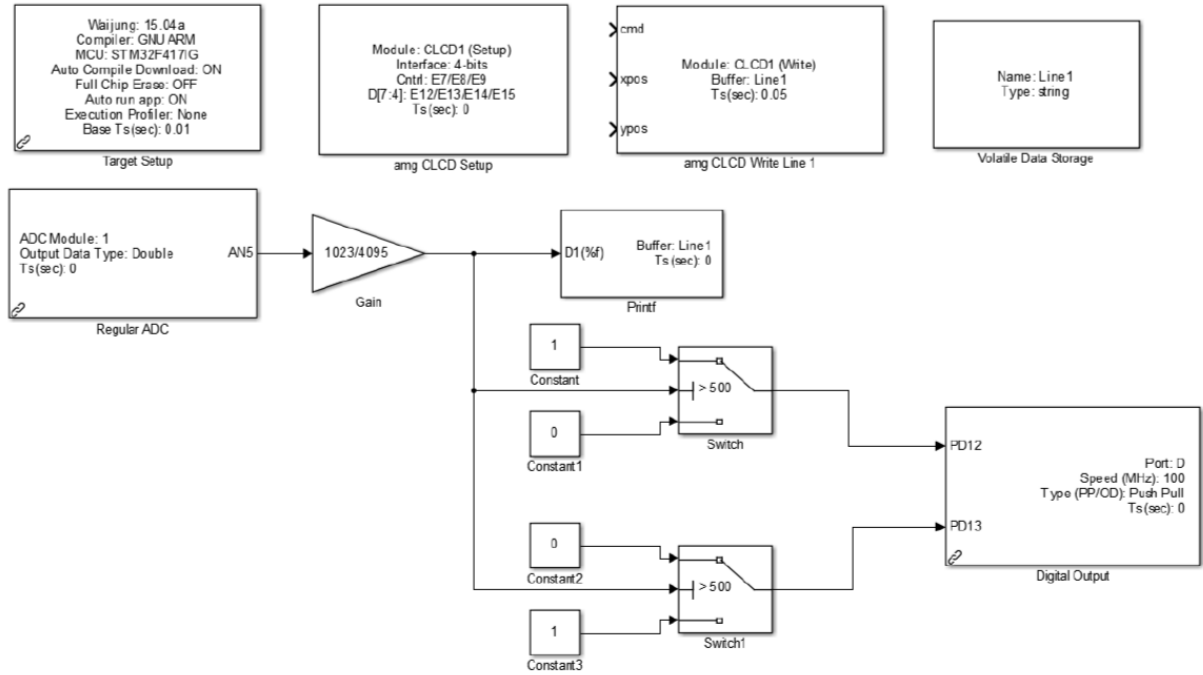
ADC ile okunan değerler voltaj değerine dönüştürülerek belirli değerlere göre (dijital çıkış) ledler yakılmıştır.

Uygulama 5.



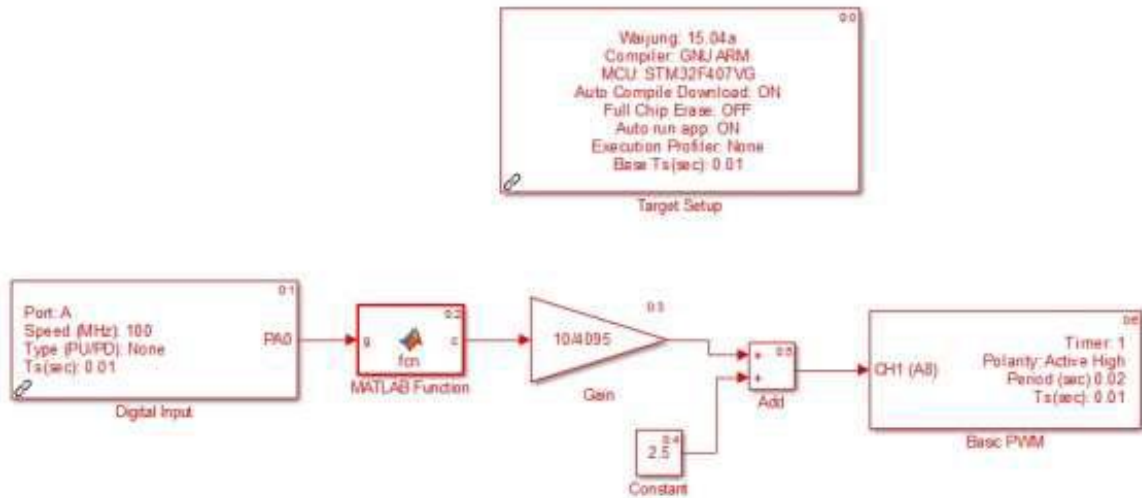
ADC portundan okunan değer ile PWM kontrolü yapılmıştır.

Uygulama 6.



A portunun 5. pinine bağlı olan potansiyometreden okunan değeri alfanümerik LCD’de yazdıran ve okunan değerin 500’den büyük veya küçük olma durumuna göre D portunun 12, 13. Pinine bağlı ledleri yakan STM32F4 modeli

Uygulama 7.



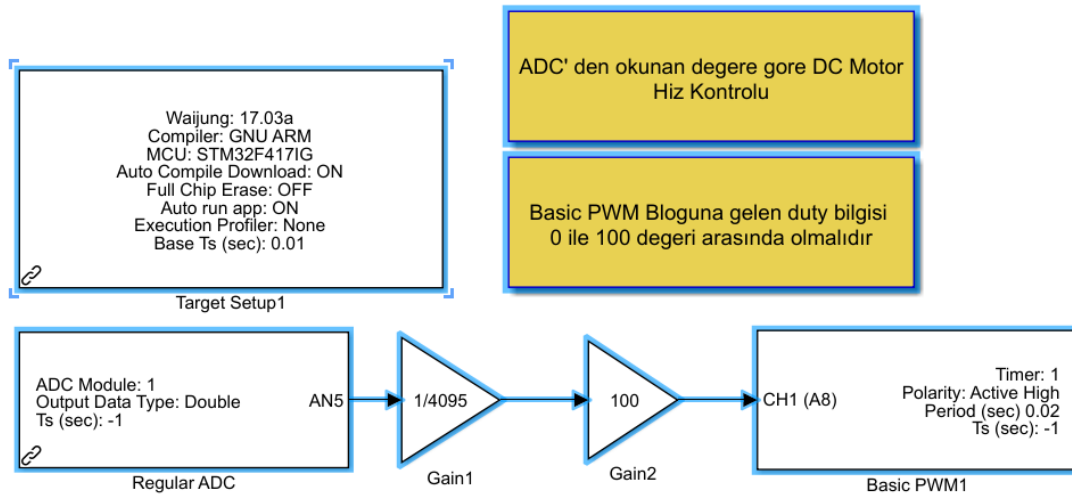
```

1 function c = fcn(g)
2     %#codegen
3     persistent buton;
4     persistent bayrak;
5
6     if isempty (buton)
7         buton=0;
8     end
9     if isempty (bayrak)
10        bayrak=0;
11    end
12
13    if (g==1) && (bayrak==0)
14        buton=buton+1;
15        bayrak=1;
16        if buton >=4
17            buton=0;
18        end
19    end
20
21    if g==0
22        bayrak=0;
23    end
24
25    if buton==0
26        c=1000;
27    elseif buton==1
28        c=2000;
29    elseif buton==2
30        c=3000;
31    else
32        c=4095;
33    end
34

```

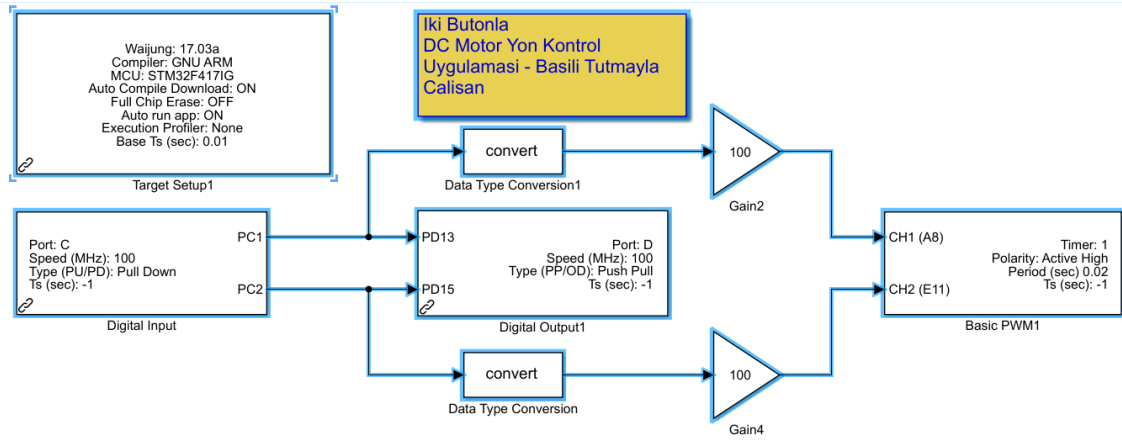
Buton ile servo hareket konumlandırma uygulaması. Butona 4 kez basılıp her seferinde farklı konumlarda servo hareket etmekte. 4. Basma da sıfırcı konuma döner.

Uygulama 8.



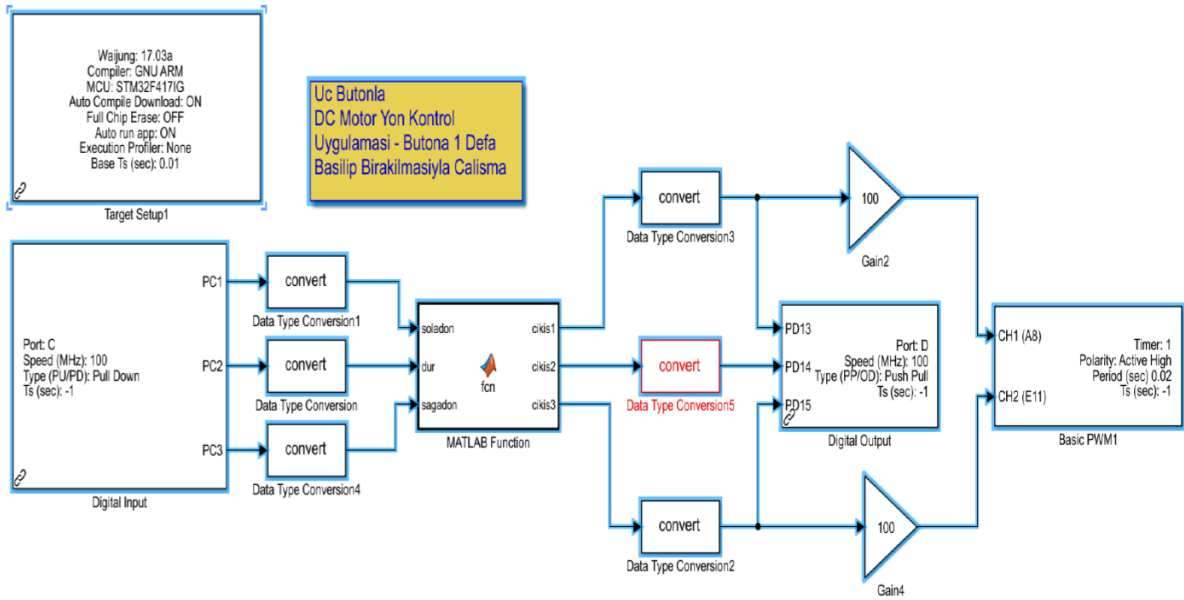
DC motor hız kontrol uygulaması

Uygulama 9.



DC motor yön kontrol uygulaması (butonla)

Uygulama 10.



%Matlab Function Kodu

```
function
[cikis1,cikis2,cikis3] =
fcn(soladon,dur,sagadon)
%#codegen
persistent sayac1;
persistent sayac2;

if isempty (sayac1)
sayac1=0;
end
if isempty (sayac2)
sayac2=0;
end

if (soladon==1)  &&
(sayac1==0)
%Motorun sola döndürülmesi
cikis1 = 1;
cikis2 = 0;
cikis3 = 0;
sayac1=1;
sayac2=0;

elseif (sagadon==1)  &&
(sayac2==0)
%Motorun sağa döndürülmesi
cikis1 = 0;
cikis2 = 0;
cikis3 = 1;
sayac1=0;
sayac2=1;

elseif (dur==1)
%Motorun Durdurulması
cikis1 = 0;
cikis2 = 1;
cikis3=0;
sayac1=0;
sayac2=0;

elseif (soladon==0)  &&
(sayac1==1)
%Solan dönen motorun dönmeye
devam ettirilmesi
cikis1 = 1;
cikis2 = 0;
cikis3 = 0;

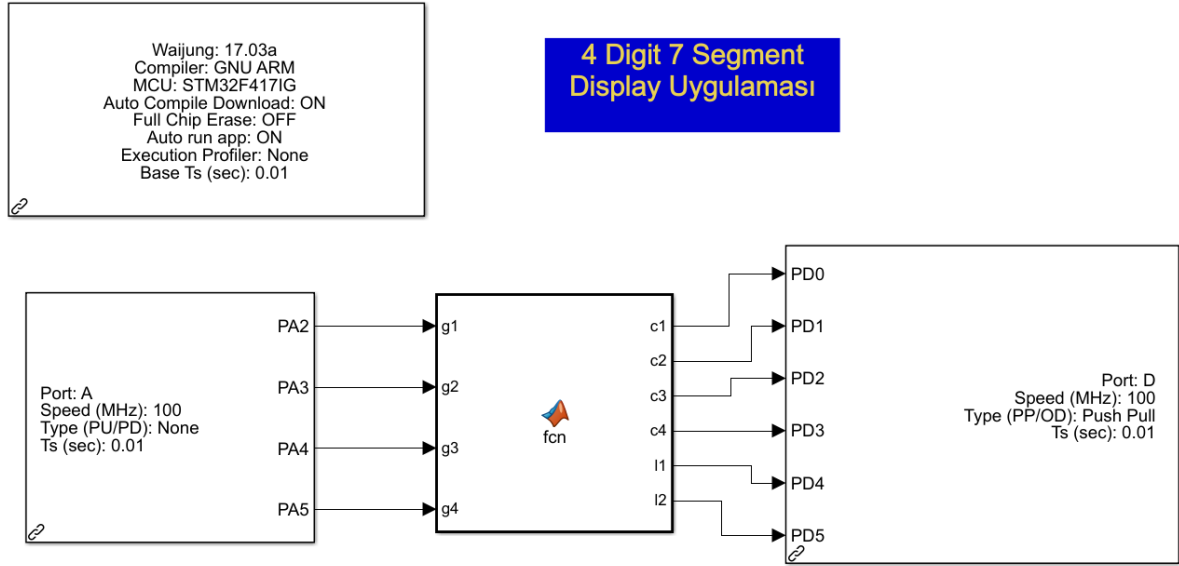
elseif (sagadon==0)  &&
(sayac2==1)
```

```
%Sağa dönen motorun dönmeye
devam ettirilmesi
cikis1 = 0;
cikis2 = 0;
cikis3 = 1;

else
%Başlangıçta motorun durması
cikis1 = 0;
cikis2 = 1;
cikis3 = 0;
end
end|
```

DC motor yön kontrol Matlab function blok kullanarak uygulanması

Uygulama 11.



```
function [c1, c2 ,c3, c4 ,l1
,l2] = fcn(g1,g2,g3,g4)
%#codegen
%b ve c parçalarını ifade
ediyor
l1=0;
l2=0;
% 4 digitten 1 tanesi giriş
butonlarıyla seçilecek
% Seçilen digitte 1 rakamı
oluşturulacak
% 1 rakamı içinde 7 segment
displayin b ve c parçaları
yakılmalı
% Kullandığımız 4 digit 7
parçalı gösterge ortak anotlu
bağlantıya sahiptir
% Anotlarının tümü birleşik,
katotlarına 0 uygulayarak
kontrolü sağlıyorum
    if(g1==1)
        % Hangi digitin ortak
anot kısmına 5V verileceğini
seçiyoruz
        c4=0;
    end
end
```

```
c1=1;
c2=0;
c3=0;
c4=0;
elseif(g2==1)
c1=0;
c2=1;
c3=0;
c4=0;
elseif(g3==1)
c1=0;
c2=0;
c3=1;
c4=0;
elseif(g4==1)
c1=0;
c2=0;
c3=0;
c4=1;
else
c1=0;
c2=0;
c3=0;
```