

01 GPIO

5 Mayıs 2021 Çarşamba

08:02

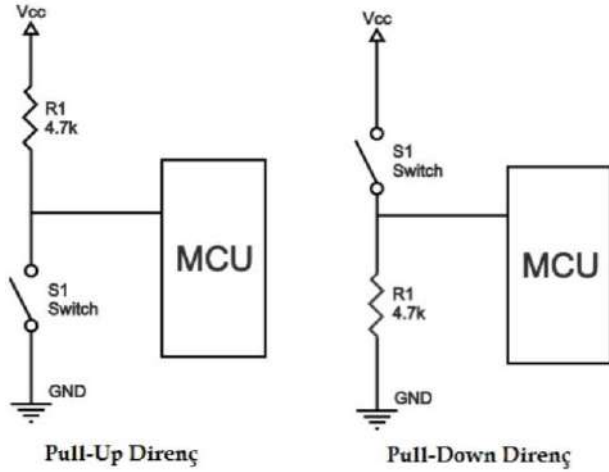
01 GPIO

Giriş

- Butonlar ve anahtarlar mikrodenetleyiciye giriş pini üzerinden lojik 1 ve lojik 0 olarak bilgi girişini sağlayan mekanik elemanlardır.

Bağlantılar

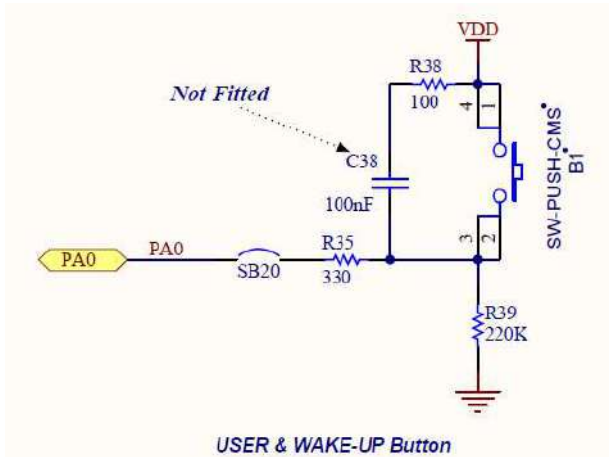
- Pull-Up** ve **Pull-Down** bağlantı tipleri olmak üzere iki giriş bağlantı tipi yer almaktadır. Bu bağlantılar, GPIO pinlerinin kararsız kalmasını önlemek ve doğru lojik seviyelerin **okunmasını** sağlamak amacıyla kullanılır.
 - Pull-Up** bağlantıda GPIO girişi direnç üzerinden + beslemeye (VCC/VDD) bağlanır.
 - Butona basılmadığı durumda GPIO girişinde lojik 1 vardır.
 - Butona basıldığı durumda girişe 0V (lojik 0) uygulanmış olur.
 - Pull-Down** bağlantıda, GPIO girişi direnç üzerinden GND ye bağlanır.
 - Butona basılmadığı durumda girişte lojik 0 bulunur.
 - Butona basıldığı durumda buton üzerinden lojik 1 uygulanmış olur.



Pull-Up Direnç

Pull-Down Direnç

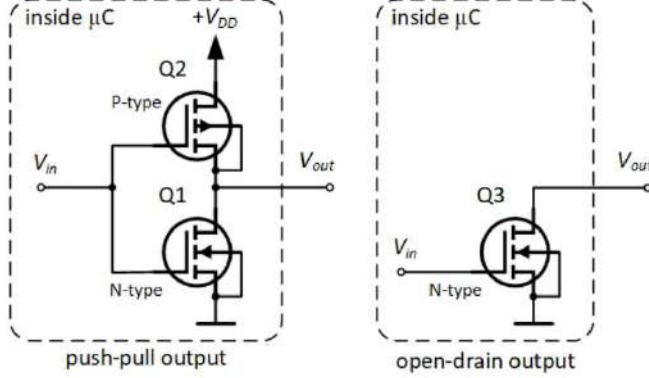
- Resimde görüldüğü gibi Nucleo bordunda kullanıcı butonu A portunun 0. pinine bağlı ve pull down durumundadır.



- Push-Pull** ve **Open-Drain** çıkış yapılandırmaları olmak üzere iki temel çıkış tipi açıklanmıştır. Bu çıkış türleri, GPIO pinlerinin harici devrelerle nasıl etkileşime gireceğini belirler ve her biri farklı uygulama senaryoları için avantajlar sunar.
 - Push-pull** çıkışlar, bir pini yüksek veya düşük seviyeye aktif bir şekilde çekebilir. Bu yapılandırmada, çıkış aşamasında genellikle iki transistör bulunur: biri çıkışı VCC'ye (yüksek voltaj) çekerken, diğeri çıkışı GND'ye düşük seviyeye çeker. Bu iki transistör birbirinin tamamlayıcısı olarak çalışır; **biri açıkken diğeri kapalıdır**. Bu sayede çıkış hızlı bir şekilde yüksekte düşüğe veya düşüğe yükseğe geçebilir. Push-pull çıkışlar genellikle LED'ler veya iç direnci olan diğer yüklerle kullanılır.
 - Daha hızlı geçiş hızları sağlar çünkü çıkış direk olarak hem yüksek hem de düşük voltaj

seviyelerine çekilebilir.

- Çıkışta bulunan iki transistör sayesinde daha yüksek akım taşıyabilir.
- Kısa devre riski daha yüksektir. Eğer her iki transistör de yanlışlıkla aynı anda aktif olursa, VCC ve GND arasında direkt bir bağlantı oluşur.
- **Open drain** veya **open collector** çıkışında, çıkış noktasında yalnızca bir transistör bulunur ve bu transistör çıkışı yalnızca GND'ye düşük seviyeye çekebilir. Yüksek seviyeye çıkması için harici bir çekme direncine pull-up resistor ihtiyaç duyar. Bu direnç, çıkışı VCC'ye çekerken transistör kapalıdır.
 - Birden fazla open drain çıkışı aynı hat üzerinde bağlanabilir (örneğin I²C gibi veri yollarında kullanılır). Bu, birbiri ile iletişim halinde olan cihazların çakışmadan veri alışverişinde bulunmasını sağlar.
 - Transistör yalnızca tek bir yönde çalıştığı için tasarım daha basit olabilir.
 - Harici bir çekme direncine ihtiyaç duyar.
 - Genellikle push-pull çıkışlara göre daha yavaş geçiş hızlarına sahiptir.

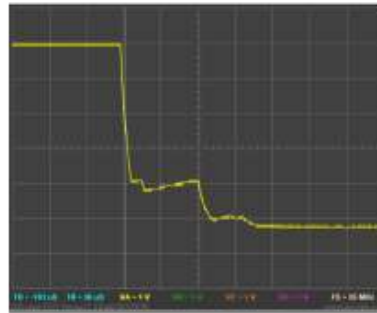
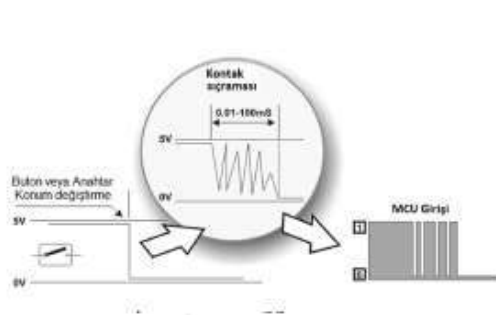


Durum	Input/Output	Pull	Drive	Örnek
Buton GND'ye bağlı	Input	Pull-Up	-	Kullanıcı butonları
Buton VCC'ye bağlı	Input	Pull-Down	-	Kullanıcı butonları
LED sürme	Output	-	Push-Pull	LED yakma
I2C veri hattı	Output	Pull-Up	Open-Drain	SDA/SCL
Çoklu cihaz aynı hattı kullanacaksa	Output	Pull-Up	Open-Drain	Wired-AND hatları

- **Pull-up kullanın** → Sinyalin varsayılan durumunun **yüksek** olmasını istediğinizde (Örnek: Aktif-düşük butonlar, I2C pull-up'lar).
- **Pull-down kullanın** → Sinyalin varsayılan durumunun **düşük** olmasını istediğinizde
- **Push-pull kullanın** → Standart dijital çıkış sinyalleri için. (Örnek: LED kontrolü, röle sürme, hızlı anahtarlama dijital sinyaller).
- **Open-drain kullanın** → Ortak bir veri yolu üzerinde birden fazla cihazın iletişim kurması gerektiğinde veya farklı voltaj seviyelerinde çalışacak cihazları bağlarken. (Örnek: I2C, tek kablolu protokoller).

Ark

Buton ve anahtarda konum değiştiğinde arkten dolayı mikrodenetleyici girişinde çok sayıda istenmeyen lojik değer oluşur. Bu duruma ark deniyor.



- Ark problemini <https://akademi.robotlinkmarket.com/buton-arki-nedir-nasil-cozulur/> linkten donanımsal ve yazılımsal olarak paylaşılan çözümleri inceleyip uygulayabiliriz.

Kontrol Yöntemleri

- GPIO pinlerini kontrol etmek için iki temel yöntem vardır. Bunlar interrupt ve polling.

İşlemcinin ve uygulamanın gereksinimlerine bağlı olarak her iki yöntem de tercih edilebilir.

- **Polling yöntemi**, mikrodenetleyici tarafından belirli bir durumun sürekli olarak kontrol edilmesine dayanır. Örneğin, bir GPIO pininin durumu sürekli bir döngü içinde kontrol edilebilir. Avantajları basit ve doğrudan bir yaklaşım ile donanım ve yazılım karmaşıklığı düşüktür. Dezavantajları sürekli olarak işlem yaparak sistem kaynaklarını tüketir. Anında tepki verme yeteneği sınırlıdır.

Basit uygulamalarda veya sürekli düşük güç tüketimi gerektiren durumlarda tercih edilebilir. Kescmelerin işlemi engelleyeceği veya karmaşık hale getireceği durumlarda kullanışlıdır. Zamanlama veya hızlı tepki gerekli olmadığında kullanılabilir.

- **Interrupt yöntemi**, bir olay (örneğin, GPIO pininin durum değıştirmesi) gerçekleştiğinde normal programın çalışmasını kesip belirli bir kesme servis rutinini çalıştırarak olaya tepki verir. Avantajları düşük enerji tüketimi, çünkü işlemci, beklenmeyen olaylar olana kadar bekler. Anında tepki verme yeteneği yüksektir.

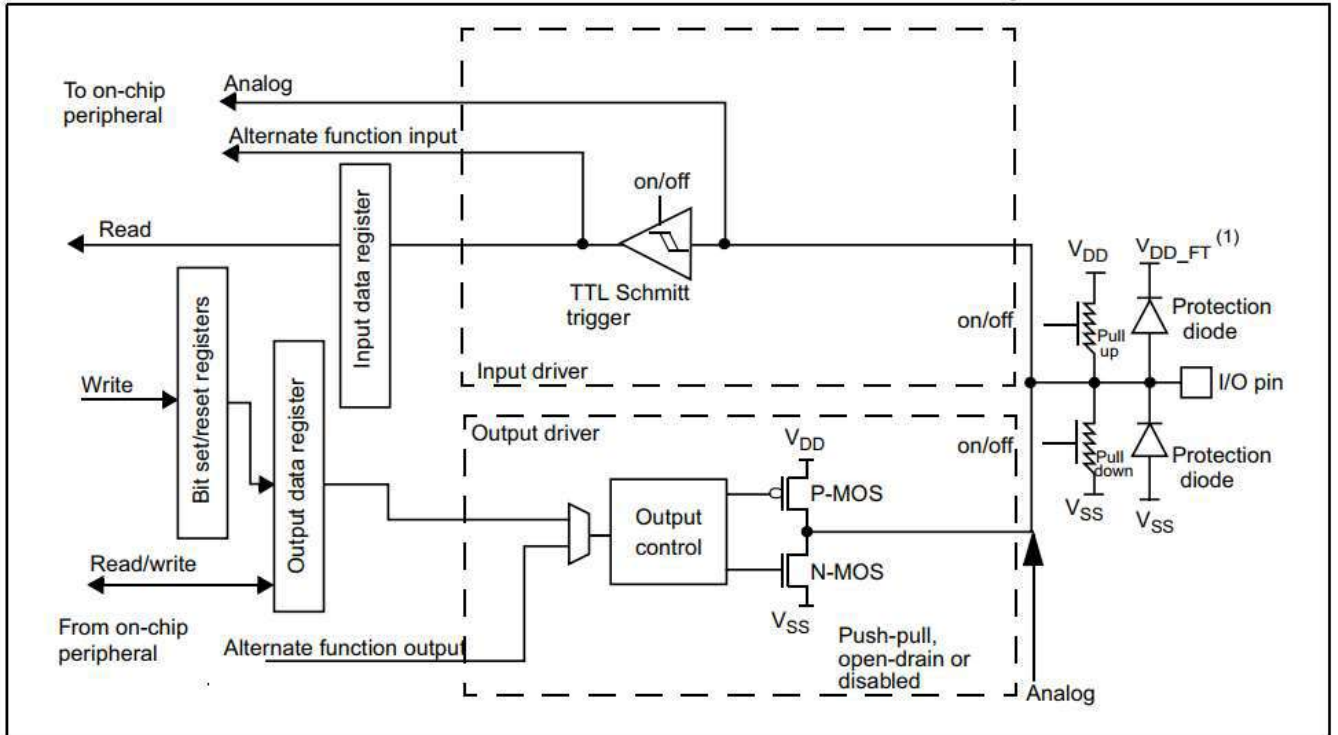
Dezavantajları, Kod karmaşıklığı ve debug işlemleri artabilir. Zamanlaması hassas olabilir ve bazı durumlarda kescmeler birbirini engelleyebilir.

Anında tepki gerektiren durumlarda (örneğin, düğme basıldığında). Enerji tüketiminin daha fazla toleranslı olduğu durumlarda. Sık sık kontrol etmenin pratik olmadığı durumlar için uygun bir seçenektir.

- Genel olarak, interrupt yöntemi, enerji tüketimi veya anında tepki gereksinimleri gibi durumlarda daha uygun olabilirken, sadece belirli durumlarda kontrol yapılması gereken basit uygulamalarda polling uygulama kullanılabilir.

Birim Yapısı

- **GPIO (General Purpose Input/Output)** pininin **iç yapısını** göstermektedir. Bu yapı, pinin hem giriş hem çıkış olarak çalışabilmesini, alternatif fonksiyonlara atanabilmesini ve analog olarak çalışmasını mümkün kılar.



- **I/O pin**, harici dünyaya bağlı olan fiziksel GPIO pinidir.
- **Protection Diode**, harici voltaj darbelerine karşı koruma sağlar böylelikle GPIO pinine aşırı voltaj uygulandığında bu diyotlar zarar görmesini engeller.
- **Schmitt trigger**, sinyali dijitalleştirerek stabil bir geçiş sağlar (histerezis)

Register

Offset	Register	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0x00	GPIOx_MODER (where x = C..I/J/K)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x04	GPIOx_OTYPER (where x = A..I/J/K)	Reserved																OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	GPIOx_OSPEEDR (where x = A..I/J/K except B)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x0C	GPIOx_PUPDR (where x = C..I/J/K)	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x10	GPIOx_IDR (where x = A..I/J/K)	Reserved																IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0	
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	GPIOx_ODR (where x = A..I/J/K)	Reserved																ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	GPIOx_BSRR (where x = A..I/J/K)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x1C	GPIOx_LCKR (where x = A..I/J/K)	Reserved																LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	GPIOx_AFR1 (where x = A..I/J/K)	AFRL7[3:0]			AFRL6[3:0]			AFRL5[3:0]			AFRL4[3:0]			AFRL3[3:0]			AFRL2[3:0]			AFRL1[3:0]			AFRL0[3:0]											
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x24	GPIOx_AFRH (where x = A..I/J)	AFRH15[3:0]			AFRH14[3:0]			AFRH13[3:0]			AFRH12[3:0]			AFRH11[3:0]			AFRH10[3:0]			AFRH9[3:0]			AFRH8[3:0]											
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		

- **GPIOx_MODER** (Mode Register) **giriş, çıkış, alternatif fonksiyon veya analog** modunu seçmek için kullanılır.
 - **MODERy[1:0]: GPIO Pin Mode Selection (y = 0..15)**
 - **00:** Input
 - **01:** Output
 - **10:** Alternate Function
 - **11:** Analog Mode
- **GPIOx_OTYPER** (Output Type Register), **Push-pull** veya **Open-drain** çıkış tipini seçmek için kullanılır.
 - **OTy: GPIO Output Type (y = 0..15)**
 - **0:** Push-Pull
 - **1:** Open-Drain
- **GPIOx_OSPEEDR** (Output Speed Register), çıkış hızını kontrol etmek için kullanılır.
 - **OSPEEDRy[1:0]: GPIO Output Speed (y = 0..15)**
 - **00:** Low Speed
 - **01:** Medium Speed
 - **10:** High Speed
 - **11:** Very High Speed
- **GPIOx_PUPDR** (Pull-up/Pull-down Register), dahili **pull-up** veya **pull-down** direncini etkinleştirmek için kullanılır.
 - **PUPDRy[1:0]: GPIO Pull-Up/Pull-Down (y = 0..15)**
 - **00:** No Pull
 - **01:** Pull-Up

- **10:** Pull-Down
- **11:** Rezerve
- **GPIOx_IDR** (Input Data Register), pinin mevcut durumunu **okumak** için kullanılır.
 - **IDRy:** GPIO Input Data (y = 0..15)
 - **0:** Low
 - **1:** High
- **GPIOx_ODR** (Output Data Register), **çıkış** durumunu ayarlamak veya temizlemek için kullanılır.
 - **ODRy:** GPIO Output Data (y = 0..15)
 - **0:** Low
 - **1:** High
- **GPIOx_BSRR** (Bit Set/Reset Register), pininin durumunu **set etmek** veya **resetlemek** için kullanılır.
 - **BSy (Bit Set):**
 - **1:** Pin Set
 - **0:** Etkisiz
 - **BRy (Bit Reset):**
 - **1:** Pin Reset
 - **0:** Etkisiz
- **GPIOx_LCKR** (Lock Register), GPIO pin konfigürasyonunun **kilitlenmesini** sağlar.
 - **LCKK (Lock Key):** Kilit komutunu etkinleştirir.
 - **1:** Kilidi etkinleştir
 - **0:** Kilidi devre dışı bırak
 - **LCKy:** GPIO Pin Lock (y = 0..15)
 - **1:** Pin kilitli
 - **0:** Pin kilitsiz
- **GPIOx_AFRH** ve **GPIOx_AFRH** (Alternate Function Low/High Register), GPIO pinlerinin **alternatif fonksiyonlarını belirlemek** için kullanılır.
 - **AFRLy[3:0]: GPIO Alternate Function Low (y = 0..7)**
 - **0000 - 1111:** Belirli alternatif fonksiyonlar (AF0 - AF15)
 - **AFRHy[3:0]: GPIO Alternate Function High (y = 8..15)**
 - **0000 - 1111:** Belirli alternatif fonksiyonlar (AF0 - AF15)

Detay

- **GPIO Output Speed** ayarı, bir çıkış pininin **yükselme (rise time)** ve **düşme (fall time)** hızını yani pinin lojik seviye değişim hızını belirler. Bu, temel olarak pinin çıkış sürücüsünün ne kadar hızlı çalışacağını kontrol eder.
 - GPIO pininde bir çıkış değişimi gerçekleştiğinde, seçilen **çıkış hızına bağlı** olarak pin üzerinden geçen **akım miktarı** değişir. Daha yüksek hız ayarlarında, transistörler daha hızlı anahtarlanır ve pin çıkışı çok daha kısa sürede seviyeler arasında geçiş yapar. Bu sayede hızlı dijital haberleşmeler mümkün olur. Ancak bu durum, artan **güç tüketimi**yle birlikte daha fazla **elektromanyetik girişim (EMI)** oluşmasına yol açabilir. Öte yandan, düşük hız ayarları daha temiz sinyaller üretir ve EMI'yi azaltır, fakat bu da veri iletim hızını sınırlayabilir. Dolayısıyla, çıkış hızı seçimi yapılırken uygulamanın ihtiyaçları ve çevresel koşullar dikkatle değerlendirilmelidir.

Seviye	Tipik Yükselme Süresi	Kullanım Senaryosu
Düşük hız	Yavaş	Güç tüketimi düşük, EMI kritik
Orta hız	Orta	Genel amaç
Yüksek hız	Hızlı	Daha hızlı haberleşmeler
En yüksek hız	Çok hızlı	Hızlı dijital veri (SPI, etc.)

Gerçek hız değerleri (örneğin ns cinsinden rise/fall time) kullanılan çekirdek (F0, F1, F4, G0 vb.), pin yapısı ve yük (kapasitans) gibi etmenlere bağlı olarak değişir.

Konfigürasyon Ayarları

1. Clock Yapılandırması

- GPIO portunu kullanmadan önce **RCC üzerinden portun saatini açmak** gerekir.

2. Pin Mode Seçimi

- Her pin için **MODER registerı** üzerinden mod seçilir:
 - 00 → Input
 - 01 → Output
 - 10 → Alternate Function
 - 11 → Analog

3. Output Type (Push-Pull / Open-Drain)

- **OTYPER registerı** ile ayarlanır:
 - 0 → Push-Pull
 - 1 → Open-Drain

4. Output Speed

- **OSPEEDR registerı** ile pin hızını ayarlarsın:
 - Low Speed
 - Medium Speed
 - High Speed
 - Very High Speed

(Hız seçimi devrenin ihtiyaçlarına göre yapılır. Örn. I2C için Medium/High Speed.)

5. Pull-up / Pull-down

- **PUPDR registerı** ile belirlenir:
 - 00 → No Pull
 - 01 → Pull-Up
 - 10 → Pull-Down

6. Alternate Function Seçimi

- Eğer pin **Alternate Function (AF)** olarak kullanılacaksa (örn. I2C, UART, SPI):

7. Veri Yazma / Okuma

- **Output pin** için:
 - ODR registerı üzerinden pin set/reset yapılır
 - veya daha hızlısı BSRR registerı kullanılabilir.
- **Input pin** için:
 - IDR registerından okunur.