

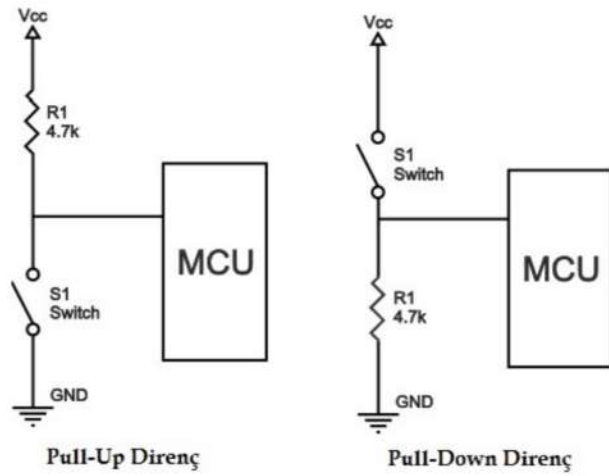
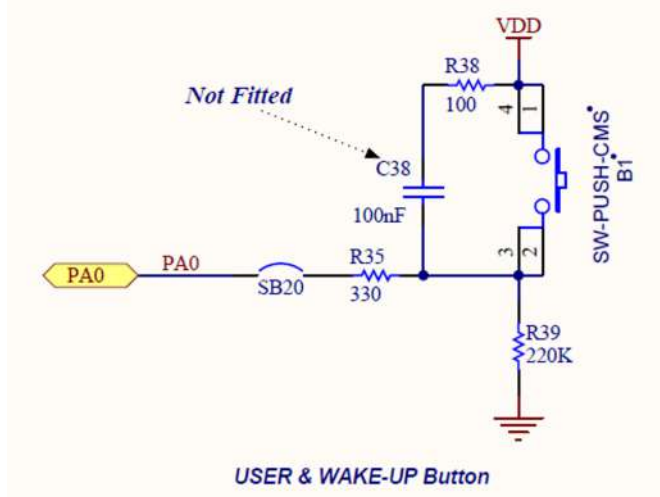
01 GPIO

5 Mayıs 2021 Çarşamba 08:02

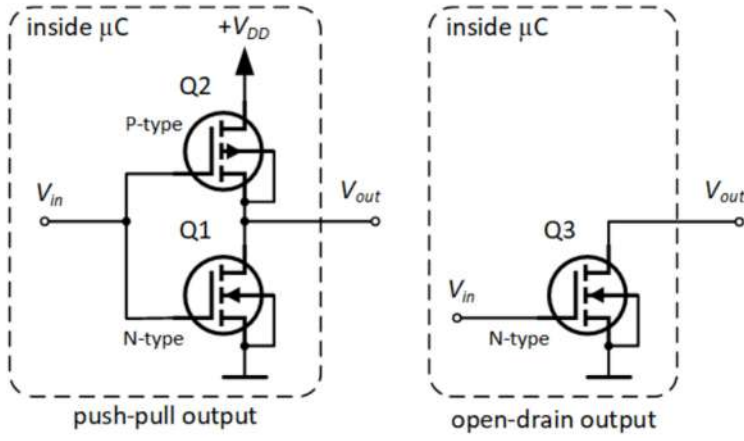
01 GPIO

Giriş

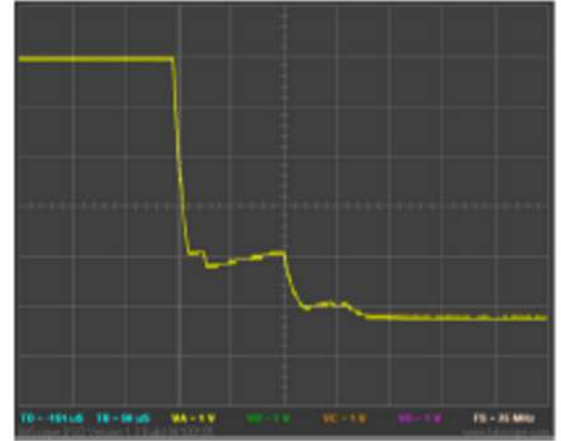
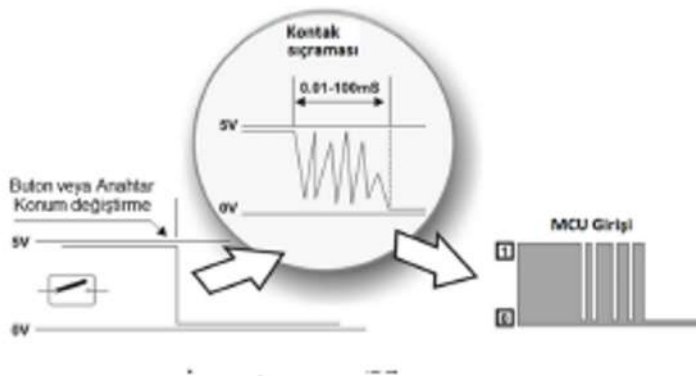
- Butonlar ve anahtarlar mikrodenetleyiciye giriş pini üzerinden lojik 1 ve lojik 0 olarak bilgi girişini sağlayan mekanik elemanlardır.
- Resimde görüldüğü gibi kullanıcı butonu A portunun 0. pinine bağlı ve pull down durumundadır.



- PullUp 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.
- PullDown bağlantıda, GPIO girişi direnç üzerinden GND ye bağlanır. Butona basılmadığı durumda girişte lojik 0 bulunur. Butona basıldığı durumda buton üzerinden lojik 1 uygulanmış olur.



- Buton ve anahtarlarda konum değiştiğinde arktan dolayı mikrodeneleyici girişinde çok sayıda istenmeyen lojik değeri oluşur. Bu duruma ark deniyor.

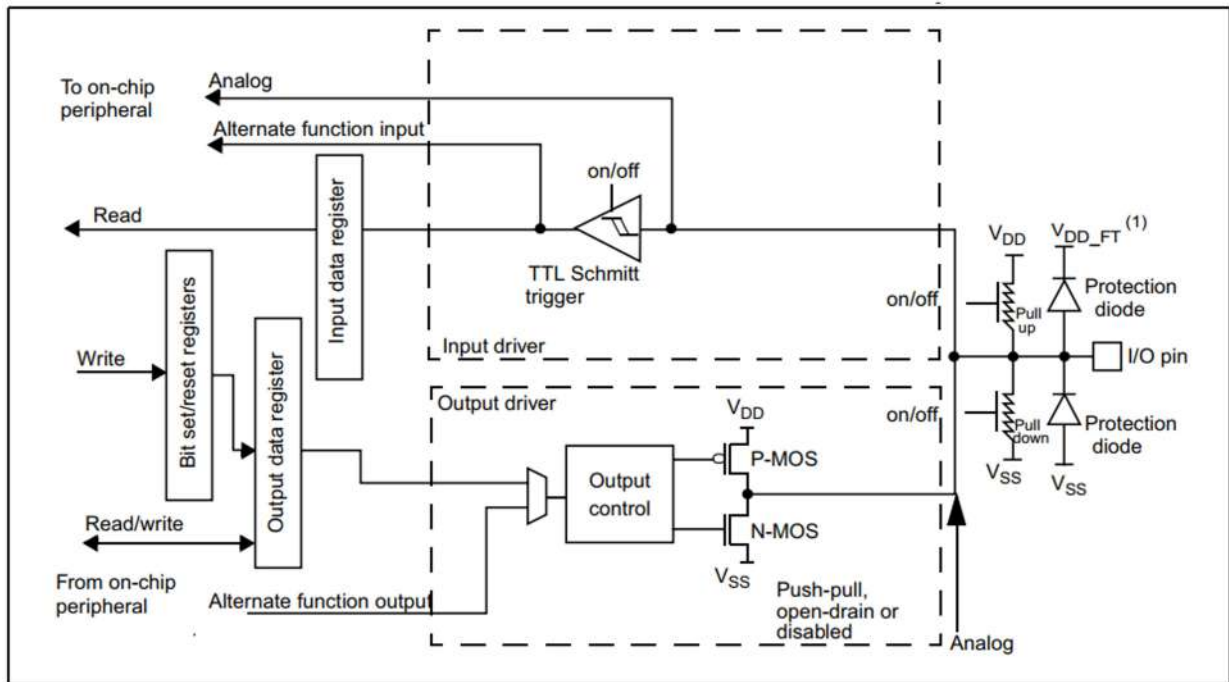


- Ark problemini <https://akademi.robotlinkmarket.com/buton-arki-nedir-nasil-cozulur/> linkten donanımsal ve yazılımsal olarak paylaşılan çözümleri inceleyip uygulayabiliriz.

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**, mikrodeneleyici 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. Kesmelerin işlemi engelleyeceği veya karmaşık hale getireceği durumlarda kullanışlıdır. Zamanlama veya hızlı tepki gerekli olmadığında kullanılabilir.
- **Interrupt yöntemi**, bir olay (örneğin, GPIO pininin durum değiştirmesi) gerçekleştiğinde normal programın çalışmasını kesip belirli bir kesme servisi 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 kesmeler 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 sorgulama kullanılabilir.

Birim Yapısı



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	
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		
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)	AFR17[3:0]				AFR16[3:0]				AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]				AFR11[3:0]				AFR10[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)**, her pin için iki bit kullanılır. Giriş, çıkış, alternatif fonksiyon veya analog modunu seçmek için kullanılır.
- **GPIOx_OTYPER (Output Type Register)**, her pin için bir bit kullanılır. Push-pull veya Open-drain çıkış tipini seçmek için kullanılır.
- **GPIOx_OSPEEDR (Output Speed Register)**, her pin için iki bit kullanılır. Çıkış hızını kontrol etmek için kullanılır.
- **GPIOx_PUPDR (Pull-up/Pull-down Register)**, her pin için iki bit kullanılır. Dahili pull-up veya pull-down direncini etkinleştirmek için kullanılır.
- **GPIOx_IDR (Input Data Register)**, her pin için bir bit kullanılır. Pinin mevcut durumunu okumak için kullanılır.
- **GPIOx_ODR (Output Data Register)**, her pin için bir bit kullanılır. Çıkış durumunu ayarlamak veya temizlemek için kullanılır.
- **GPIOx_BSRR (Bit Set/Reset Register)**, her pin için iki bit içerir. Bir GPIO pininin durumunu set etmek veya resetlemek için kullanılır.
- **GPIOx_LCKR (Lock Register)**, her pin için bir bit içerir. GPIO pin konfigürasyonunun kilitlenmesini sağlar.

- **GPIOx_AFRL ve GPIOx_AFRH (Alternate Function Low/High Register)**, her biri 32-bit uzunluğunda iki registerdır ve her pin için dört bit içerir. GPIO pinlerinin alternatif fonksiyonlarını belirlemek için kullanılır.