10 I2C

5 Mayıs 2021 Çarşamba 08:03

10 I2C

Giriş

- https://www.ercankoclar.com/2018/01/i2c-iletisim-protokolu-ve-mikroc-kutuphanesi/ ve https://ozdenercin.com/2019/01/25/i2c-seri-haberlesme-protokolu/ linklerinden ayrıntılı bilgilere ulaşabiliriz.
- https://www.ti.com/lit/an/slva704/slva704.pdf?ts=1702118996423&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253DUnderstanding%2Bthe%2BI%2B2C%2BBus%2526nr%253D1136 linkten Texas Instruments'in I2C protokü hakkında yazdığı makaleyi okuyabiliriz.
- I2C prokolünün geliştirlme amacı, düşük hızlı çevre birimlerinin ana kartları, cep telefonları, gömülü sistemler gibi elektronik cihazlara daha az kablo ihtiyacı ile bağlanabilmesini sağlamaktadır.

Avantajları ve Dezavantajları

Avantajlar;

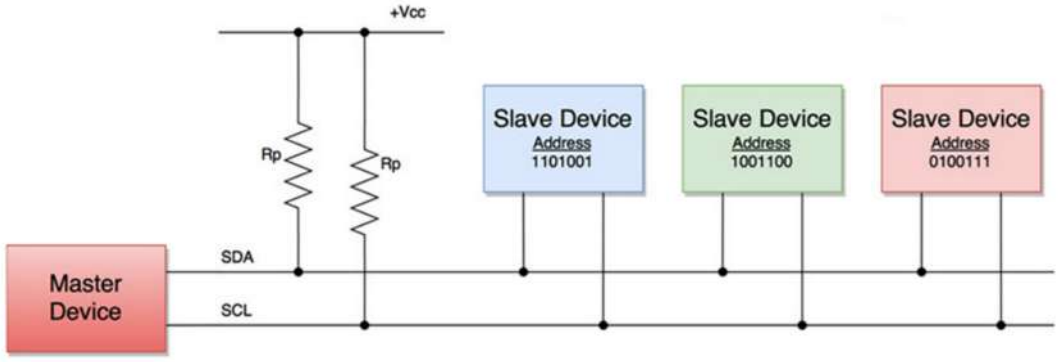
- Sadece iki telli bir yapıya sahiptir (SCL ve SDA), bu nedenle **az pin** kullanımı ile birçok cihazın bağlanmasını sağlar.
- Aynı hat üzerinden **çift yönlü iletişim** sağlar. Hem master hem de slave cihazlar veri gönderebilir ve alabilir.
- Bir dizi cihazın (EEPROM, sensörler, ekranlar vb.) bağlanmasını destekler, bu da çok **çeşitli uygulamalara** olanak tanır.
- Master ve slave cihazlar arasındaki bağlantıları daha **esnek** hale getirir. Çoklu master bağlantılarına izin verir.
- **Open-drain çıkışı**, güç tüketimini azaltır ve daha güvenilir bir iletişim sağlar.
- Y**üksek veri iletim hızlarına** izin verecek şekilde tasarlanmıştır.

Dezavantajlar;

- I2C'nin **uzun hat mesafelerinde** performansı düşük olabilir. Bu durum, iletişim hızını düşürmek veya ek güçlendirme önlemleri almak gerektirebilir.
- Birden çok masterın bulunduğu sistemlerde **çatallanma** sorunları ortaya çıkabilir. Bu durum, çakışmaları önlemek için dikkatlice senkronize edilmiş bir sistem gerektirir.
- Başlangıçta **karmaşık** olabilir ve doğru yapılandırma ve senkronizasyon gerektirebilir.
- Önceden belirlenmiş bir adres yapısı kullanır, bu nedenle **güvenlik** açısından zayıf olabilir.
- Uzun hat mesafelerinde veya yüksek hızlarda iletişimde, **elektromanyetik girişim** sorunları ortaya çıkabilir.

Bağlantılar

- I2C iletişiminde sadece iki hat vardır. Bunlar SDA (Serial Data Line) ve SCL (Serial Clock Line) hatlarıdır.
- Genellikle +5V ve +3.3V voltajlarda çalışmakla beraber, I2C protokolü daha pratik voltaj seviyelerine de izin vermektedir.
- SDA hattı harberleşmeyi başlatıp, sonlandırır. SCL ise veri hattı kondigurasyonunu sağlar.

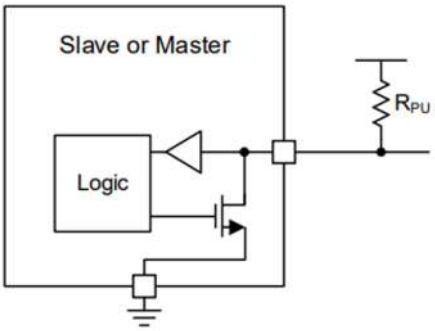


Çift Yönlü Haberleşme

- I2C, aynı hat üzerinde open-drain/open-collector ile bir giriş buffer kullanır, bu da tek bir veri hattının çift yönlü veri akışı için kullanılmasına olanak tanır.
- Bu hatlar ayrıca pull-up direncine ihtiyaç duyarlar.
- Yalnızca bir cihaz veri yolu hattını toprağa çekebilir veya veri yolu hattını serbest bırakır ve böylece pull-up

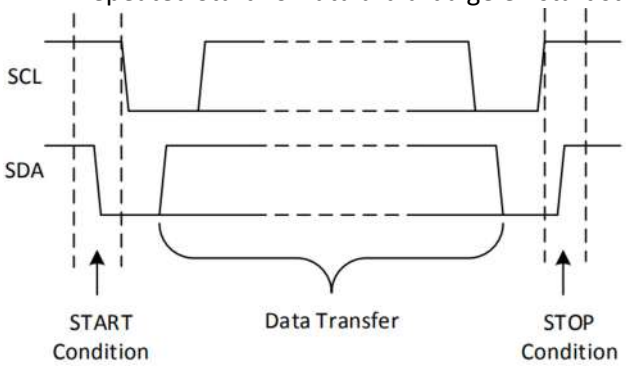
direncinin voltajı yükseltmesine izin verir.

- Hattı LOW seviyeye çekmek için FET transistör **tetiklenir** böylece hat toprak ile kısa devre olur.
- Hattı HIGH seviyeye çekmek için FET transistör **kapatılır** böylece hat pull-up direnci vasıtasıyla voltaj seviyesine çekilir.

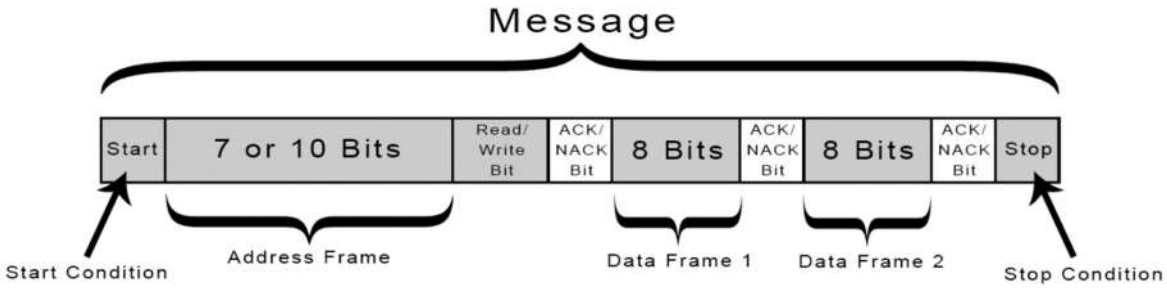


Veri İletimi

- I2C hattı SDA hattının lojik high seviyesinden lojik low seviyeye düşmesi ile başlar. Aynı şekilde lojik low seviyesinden lojik high seviyeye çıkması ile sonlanır. SDA hattının haberleşmeyi başlatabilmesi için SCL hattı da high olmalıdır.
- SCL hattı lojk high seviyesinde iken SDA hattı high seviyesine çekilirse haberleşme sonlanır.
- I2C veri gönderiminde start biti "0" stop biti "1" verilerinden oluşmaktadır.
- I2C veriyolu multimaster bir yapıdadır. Bu sayede letişim hattında birden fazla cihaz olabilir. Master cihazlarda bir saat sinyali ve data gönderildiği anda diğer cihazların tamamı slave moduna geçerler.
- Multimaster I2C haberleşmesinde Repeated Start komutu vardır ve sıklıkla kullanılır. I2C haberleşmesinde 2 adet cihaz olduğunu varsayalım.
  Birinci master cihaz start komutu gönderdi ve start komutundan sonra gerekli adres bilgilerini gönderdi. Tüm bu işlemler sürecinde I2C hattı birinci master cihaz tarafından kullanıldığından dolayı I2C hattı idle durumda olmayacaktır. Birinci cihaz stop durumu göndermeden önce haberleşmede bir değişiklik yapmak isterse Repeated Start komutunu gönderir ve böylece 1.Master cihazın slave cihaz ile I2C haberleşmesi kopmamış olur. Multimaster olmayan durumlarda Repeated Start komutunu kullanmaya gerek yoktur. Repeated Start komutu ard arda gelen start stop komutlarından oluşur.

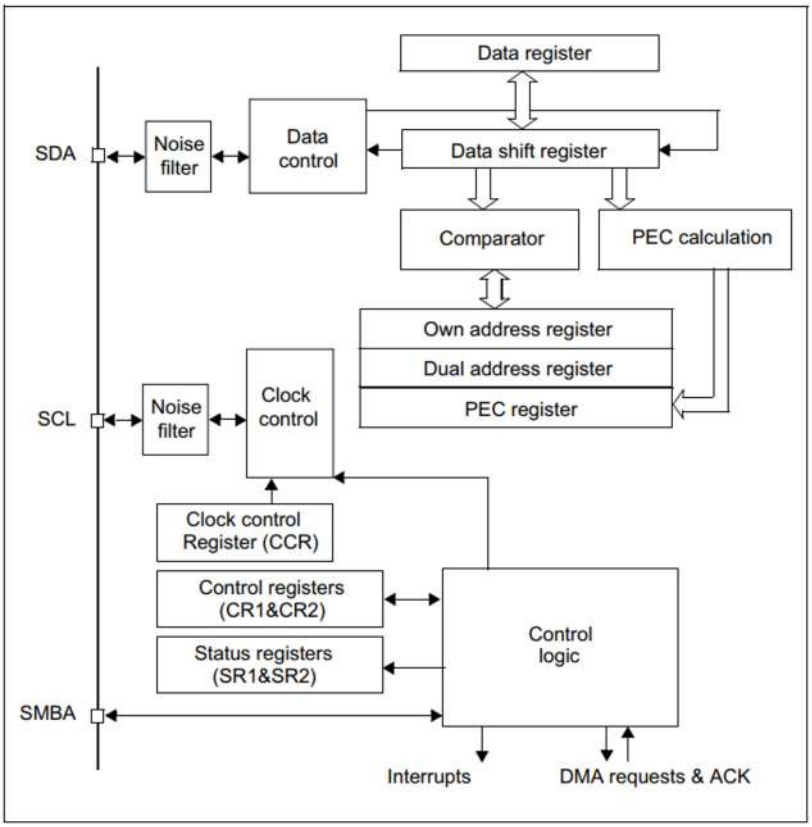


- İlk olarak SDA ve SCL hatları HIGH konumdadırlar. Daha sonra SDA hattı master tarafından **LOW** seviyeye çekilerek **iletişimin başlayacağı**, slave cihazlara bildirilir.
- Bu bildirimi alan slave cihazlar, adres bilgisini beklemeye başlarlar.
  Adres bilgisi slave cihazların yapısına göre 7 bit, 10 bit veya 16 bit olabilirler.
- Master cihaz hangi slave cihaz ile haberleşmek istiyorsa onun adres bilgisini gönderdikten sonra, **okuma** mi yoksa **yazma** mı yapacağını belirtir. Adres hangi slave cihazın ise o cihaz master ile iletişim kurmaya başlar.
- Adres kendisine ait olan slave cihaz, master cihaza verinin gönderildiği veya verinin alındığını doğrulamak için bir **ACK** (Acknowledge) kabul biti gönderir.
- Veri transferi işlemi gerçekleşir. Bu transfer iki yönlü de olabilir. (Slave'den Master'a veya Master'dan Slave'e)



- I2C haberleşmesinde 1 master cihaz ve birden fazla slave cihaz olduğunu varsayalım. Master cihaz herhangi bir slave cihaza erişmek için start komutundan sonra ilgili slave cihazın adresini gönderir. Aynı hatta bağlı olan slave cihazların tamamı bu mesajları alır ancak sadece bu mesaja sahip olan slave cihaz Ack mesajını göndererek iletişimin kurulduğu master cihaza bildirir ve Ack mesajını alan master cihaz adres bilgisinden hemen sonra veri göndermeye başlar.

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	I2C_CR1	Reserved																SWRST	Reserved	ALERT	PEC	POS	ACK	STOP	START	NOSTRETCH	ENGC	ENPEC	ENARP	SMBTYPE	Reserved	SMBUS	PE
	Reset value																	0		0	0	0	0	0	0	0	0	0	0	0		0	0
0x04	I2C_CR2	Reserved																			LAST	DMAEN	ITBUFEN	ITEVTEN	ITERREN	Reserved	FREQ[5:0]						
	Reset value																				0	0	0	0	0		0	0	0	0	0	0	0
0x08	I2C_OAR1	Reserved																ADDMODE	Reserved					ADD[9:8]		ADD[7:1]							ADD0
	Reset value																	0						0	0	0	0	0	0	0	0	0	0
0x0C	I2C_OAR2	Reserved																								ADD2[7:1]							ENDUAL
	Reset value																									0	0	0	0	0	0	0	0
0x10	I2C_DR	Reserved																								DR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x14	I2C_SR1	Reserved																SMBALERT	TIMEOUT	Reserved	PECERR	OVR	AF	ARLO	BERR	TxE	RxNE	Reserved	STOPF	ADD10	BTF	ADDR	SB
	Reset value																	0	0		0	0	0	0	0	0	0		0	0	0	0	0
0x18	I2C_SR2	Reserved																PEC[7:0]								DUALF	SMBHOST	SMBDEFAUL	GENCALL	Reserved	TRA	BUSY	MSL
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x1C	I2C_CCR	Reserved																F/S	DUTY	Reserved	CCR[11:0]												
	Reset value																	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	I2C_TRISE	Reserved																										TRISE[5:0]					
	Reset value																											0	0	0	0	1	0
0x24	I2C_FLTR	Reserved																											ANOFF	DNF[3:0]			
	Reset value																												0	0	0	0	0

- **I2C_CR1 (Control Register 1)**, Ana kontrol registerıdır. I2C Peripherals'ı etkinleştirir veya devre dışı bırakır. Gönderme tamamlandığında kesme (interrupt) etkinleştirir. Acknowledge kontrol biti ile Yazılım sıfırlama biti bulunmaktadır.
- **I2C_CR2 (Control Register 2)**, ikinci kontrol registerıdır. Saat frekansını belirler.
- **I2C_OAR1 (Own Address Register 1)**, I2C'nin kendi adresini ayarlamak için kullanılır.
- **I2C_DR (Data Register)**, veri göndermek veya almak için kullanılır.
- **I2C_SR1** ve **I2C_SR2 (Status Register 1 ve 2)**, I2C'nin durumu hakkında bilgi sağlar. Birçok farklı durumu içerir, örneğin, START biti durumu, adres gönderme durumu, veri alım durumu vb.
- **I2C_CCR (Clock Control Register)**, I2C saat frekansını kontrol eder.
- **I2C_TRISE (Rise Time Register)**, yükselme süresini ayarlamak için kullanılır.
- **I2C_FLTR (Filter Register)**, I2C hatlarında gürültüyü azaltmaya yönelik bir filtreleme mekanizması sağlar.

Haberleşme Metotları

- SPI üzerinden Polling, Interrupt ve DMA olmak üzere üç farklı haberleşme yapılabilir.
- https://deepbluembedded.com/stm32-i2c-tutorial-hal-examples-slave-dma/ linkinden konu hakkındaki bilgileri inceleyebiliriz.