

11.07.2025

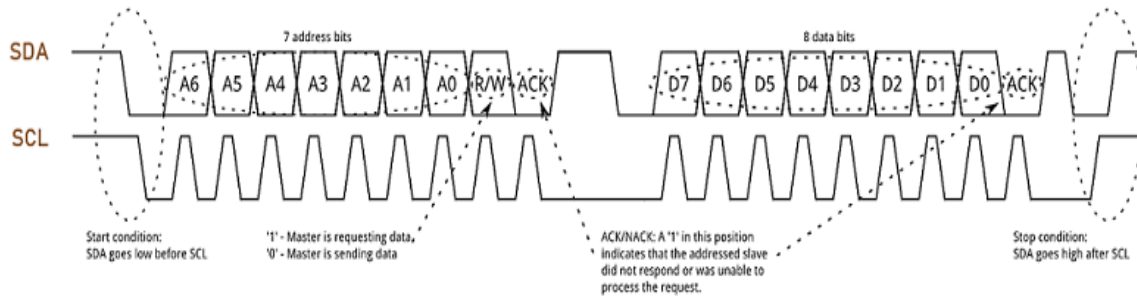
STAJ RAPORU

STM32G03x - BQ25672 I2C DRIVER

Yunus Emre DUNAY
TUBİTAK MEKTAM

1- I2C İletişim Protokolü

I2C, NXP tarafından geliştirilmiş ve birden fazla master cihazı destekleyebilen bir haberleşme protokolüdür. Bu protokolün öne çıkan özelliklerinden biri, master-slave yönetiminin tamamen donanım seviyesinde otomatik olarak gerçekleştirilmesidir. Bu durum, özellikle gömülü sistemlerde yazılım yükünü azaltır ve protokolün yönetimini kolaylaştırır. Örneğin, SPI gibi alternatif protokollerde bu yönetim yazılım aracılığıyla manuel şekilde yapılırken, I2C'de geliştiricinin bu sürece müdahil olmasına gerek kalmaz. Verinin güvenilir biçimde aktarılabilmesi için, her iletilen byte'tan sonra alıcı cihaz tarafından bir onay (ACK) sinyali gönderilmesi gerekir. Bu ACK sinyali, SDA hattında saat darbesiyle eş zamanlı olarak oluşturulmalıdır. SPI'da bu tür kontrol mekanizmaları kullanıcı tarafından yazılım ile denetlenirken, I2C'de tüm süreç donanım tarafından otomatik olarak yürütülür ve kullanıcıya inanılmaz büyük bir kolaylık sağlanır.

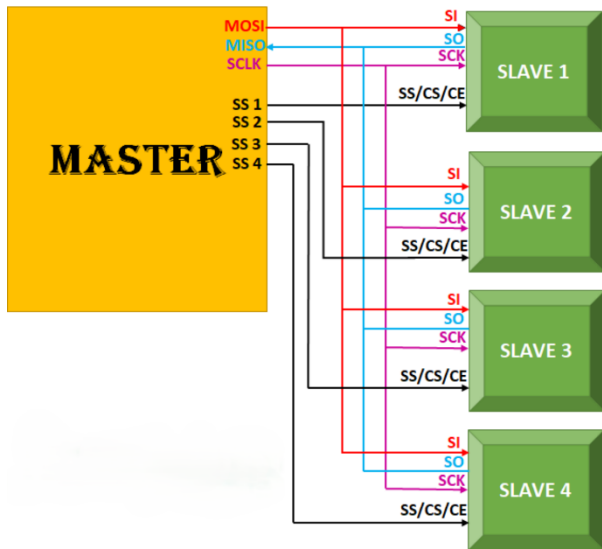


Şekil 1.1

Şekil 1.1’de, I²C iletişim protokolünde kullanılan SDA (Serial Data Line) ve SCL (Serial Clock Line) hatlarının zamanlama diyagramı gösterilmektedir. Bu iki hat, veri iletim sürecinin senkron ve güvenli bir şekilde gerçekleşmesini sağlar. İletişim, “start condition” olarak adlandırılan özel bir başlangıç durumu ile başlar. Bu durumda SDA hattı, SCL hâlâ yüksekken düşer (SDA goes low before SCL) ve bu, tüm cihazlar tarafından iletişimin başladığı anlamına gelir. Ardından master cihaz, hedef slave cihaza 7 bitlik adres (A6–A0) ve ardından bir R/W biti gönderir. R/W biti ‘0’ olduğunda veri gönderimi(write), ‘1’ olduğunda ise veri talebi(read) anlamına gelir. Slave cihaz, bu adresleme sonrasında iletişimi kabul ettiğini bir ACK sinyali ile bildirir. Veri iletim aşamasında, gönderilen her byte (D7–D0) sonrası bir ACK biti beklenir. Eğer bu sinyal alınamazsa, durum bir NACK olarak değerlendirilir ve bu, adresin tanınmadığını ya da yanıt verilemediğini gösterir. İletişim tamamlandığında master, SDA hattını SCL yüksekken tekrar yüksek seviyeye çekerek “stop

condition” üretir (SDA goes high after SCL) ve veri aktarımı sonlandırılır. Bu şekil, I2C protokolünde iletişim sırasının donanımsal düzeyde nasıl işlediğini zamana bağlı olarak açıkça ortaya koymaktadır.

I2C mimarisinin en önemli avantajlarından biri, yalnızca iki hat üzerinden çok sayıda cihazın birbirine bağlanabilmesidir. SDA (veri hattı) ve SCL (saat hattı) üzerinden gerçekleşen bu yapı, yalnızca iki pin kullanımı ile tüm haberleşme sürecini yöneterek sistem tasarımında tasarruf sağlar. Bu minimal donanım gereksinimi, özellikle çok sayıda çevresel birimin bulunduğu uygulamalarda büyük bir avantaj sağlar. Ancak bu avantajın beraberinde getirdiği önemli bir dezavantaj da söz konusudur: I2C iletişimi yarı-dupleks yapıda çalışır. Yani veri iletimi eş zamanlı olarak iki yönlü gerçekleştirilemez; gönderim ve alım işlemleri sırayla yapılmak zorundadır. Bu yapı, protokolün genel hızını yüksek oranda düşürür.

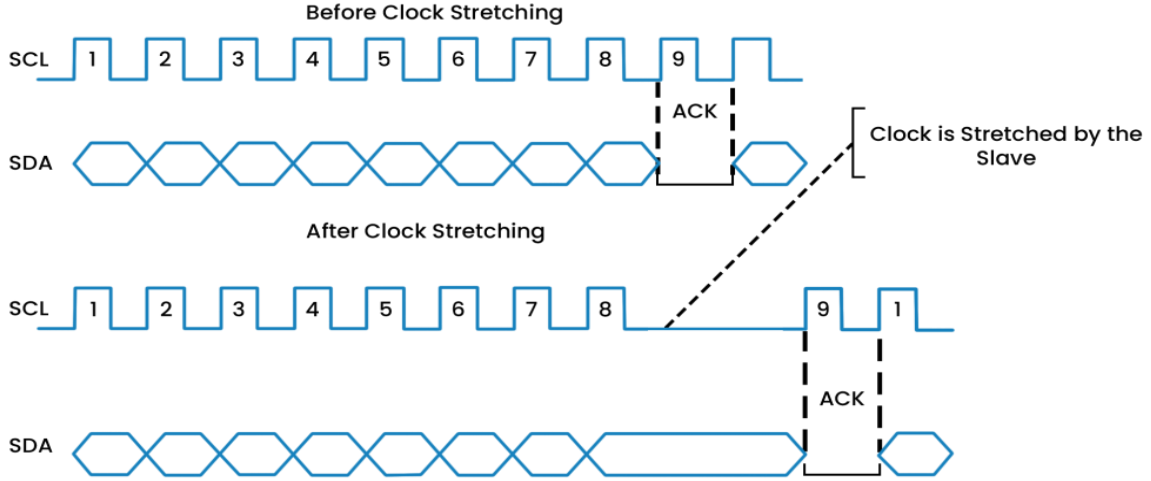


Şekil 1.2

Bu durum, tam-dupleks çalışan SPI protokolü (Şekil 1.2) ile karşılaştırıldığında daha belirgin hale gelir. SPI, aynı anda hem veri gönderip hem de alabilme yeteneği sayesinde I2C'ye kıyasla çok daha yüksek hızlarda çalışabilir. Örneğin STM32G03 serisi mikrodeneleyicilerde, I²C ile standart modda 100 kbps, hızlı modda 400 kbps ve hızlı mod plus (FM+) ile yaklaşık 1 Mbps seviyelerine kadar

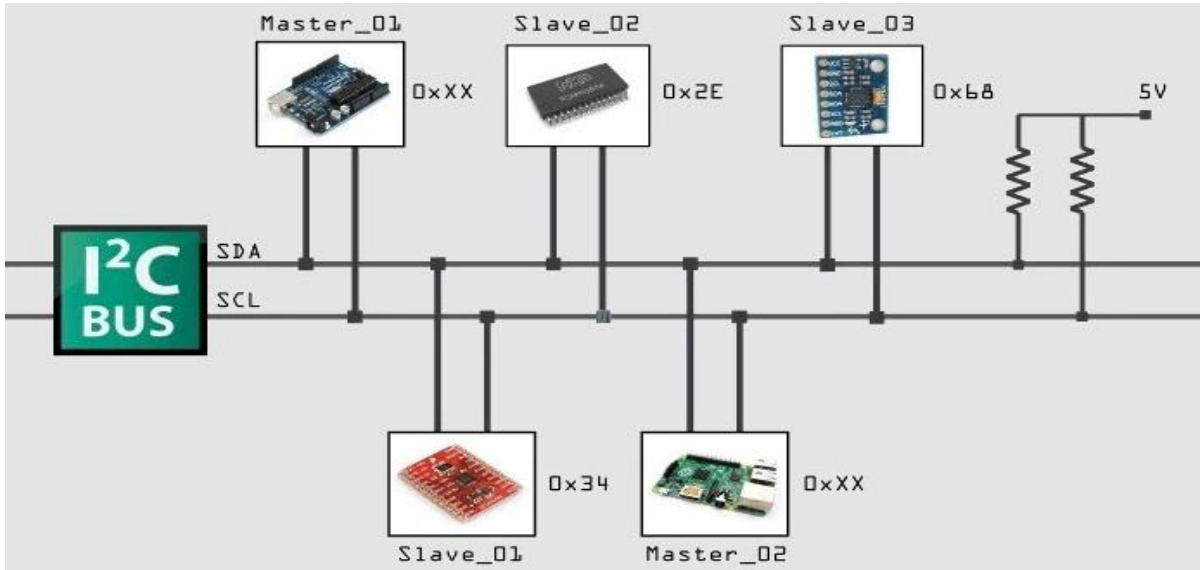
veri iletim hızı elde edilebilirken, SPI protokolü aynı seride 16 Mbps düzeylerine kadar çıkabilmektedir. Dolayısıyla, yüksek hızlı ve eşzamanlı veri aktarımı gerektiren uygulamalarda, I2C'nin bu yarı-dupleks yapısı ciddi bir dezavantaj oluşturur ve protokol tercihinde önemli bir rol oynar.

Veri iletişimi sırasında, bazı durumlarda alıcı cihazın işlem hızına göre senkronizasyonu kontrol altına alması gerekebilir. Bu tür senaryolarda “clock stretching” adı verilen bir mekanizma devreye girer (Şekil 1.3). Burada, slave cihaz saat sinyalinin low seviyede tutarak, veri işleme süresini uzatabilir. Bu özellik sayesinde protokol, farklı işlem gücüne sahip cihazlar arasında uyumlu ve sorunsuz bir şekilde çalışabilir.



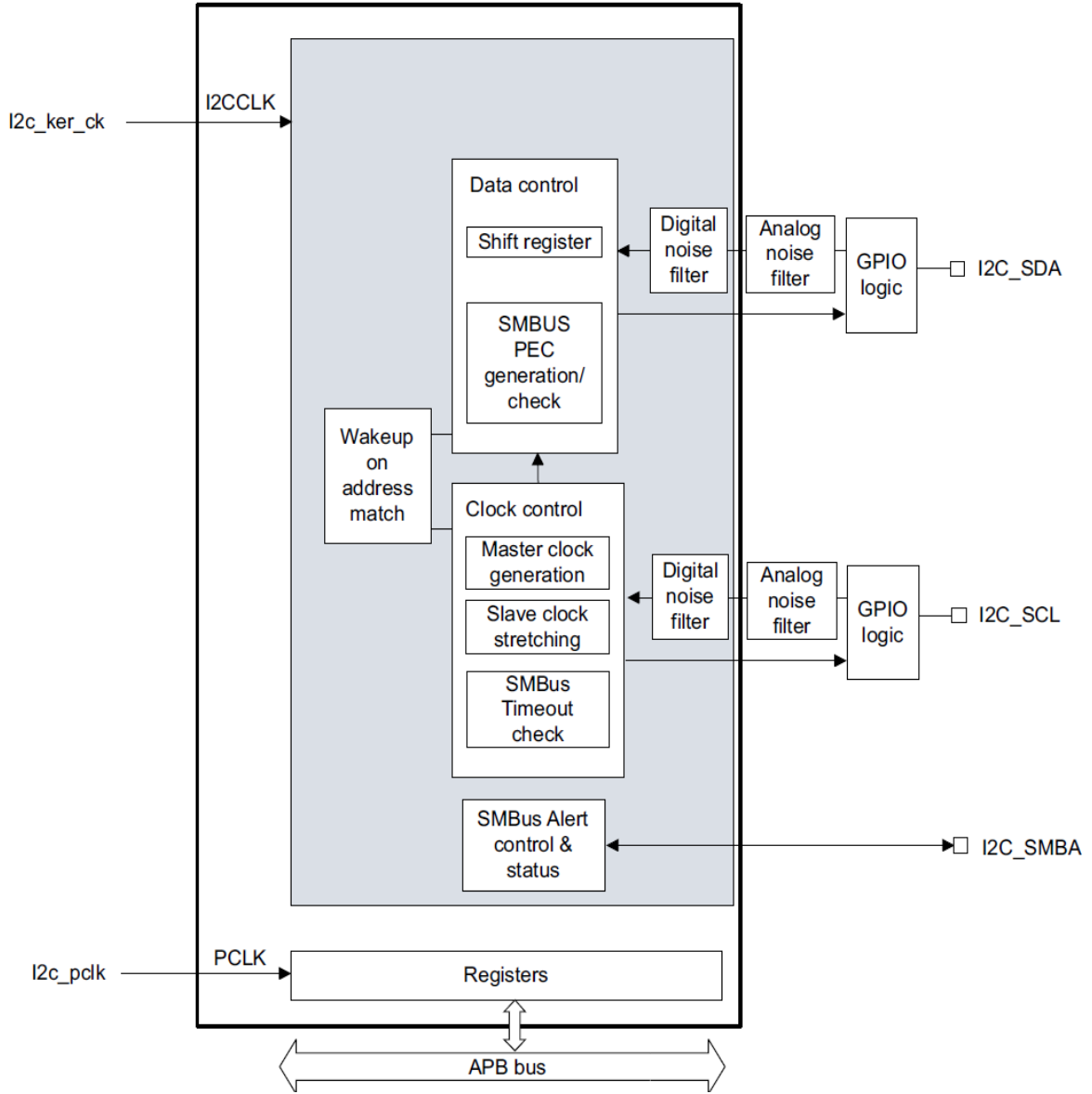
Şekil 1.3

I2C hatları, çift yönlü çalışacak şekilde tasarlanmıştır ve hatlar boşta kaldığında, harici pull-up dirençleri aracılığıyla Vcc seviyesinde tutulur. Bu yapı, veri hattında çakışma riskini azaltmak için genellikle open-drain veya open-collector tipi çıkışlarla birlikte kullanılır. Push-pull çıkışlar ise, aynı hatta birden fazla cihaz bağlandığında kısa devre riski taşıdığı için I2C’de tercih edilmez. Raporun ilerleyen kısımlarında çıkış yapıları ile alakalı çok daha detaylı açıklamalar mevcuttur.



Şekil 1.4

I2C’nin bir diğer güçlü yönü ise çoklu master desteğidir (Şekil 1.4). Bu tür durumlarda, bir master cihaz veri gönderimini bitirmeden doğrudan başka bir cihaza veri iletmek istediğinde “tekrar başlatma” (repeated start) sinyali kullanılır. Bu mekanizma, özellikle bus çakışmalarının önlenmesi açısından kritik bir rol oynar ve I2C’nin çoklu cihazlı yapısında sorunsuz iletişimi mümkün kılar.



Şekil 1.5

STM32G03 serisi mikrodnetleyicilerde yer alan I²C donanımı, Şekil 1.5'te gösterilen blok diyagramda görüldüğü gibi; veri kontrolü, saat yönetimi, gürültü filtreleme ve hata denetimi gibi birçok temel işlevi içinde barındıran bir yapıya sahiptir. SDA ve SCL hatları mikrodnetleyiciye ulaşmadan önce sırasıyla analog ve dijital filtrelerden geçirilerek parazitlere karşı korunur. Bu filtreleme sayesinde, hat üzerindeki kısa süreli sinyal bozulmalarının haberleşmeyi etkilemesinin önüne geçilmiş olur.

Veri iletimi, shift register aracılığıyla gerçekleştirilirken, aynı modül içinde SMBus (System Management Bus – I²C tabanlı, hata yönetimi ve güvenlik odaklı sistemler için geliştirilmiş iletişim protokolü) desteği kapsamında PEC (Packet Error Checking) işlemleri de donanım düzeyinde sağlanır. Clock

sinyalleri, master veya slave moda göre otomatik olarak üretilir. Slave modda, clock stretching özelliği sayesinde cihaz, işlem süresi gerektiğinde master'ı geçici olarak duraklatabilir. Sistem, APB veri yolu üzerinden işlemciyle haberleşir ve gerektiğinde DMA(Direct Memory Access) ile doğrudan bellek erişimi de desteklenir.

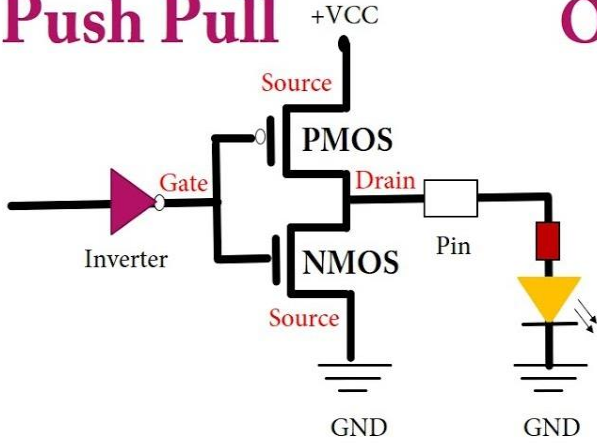
Sonuç olarak, I2C protokolü; donanım temelli yönetimi, az sayıda pin kullanımı ve çoklu cihaz desteği ile ön plana çıkmaktadır. Özellikle düşük ve orta hız aralığında çalışan, çok sayıda bileşenin haberleştiği sistemlerde I2C, güvenli bir tercih olarak kullanılmaya devam etmektedir.

2- Çıkış Yapıları: Push-Pull ve Open-Drain

Mikrodenetleyici sistemlerinde dijital çıkış yapılandırmaları, pinlerin dijital sinyal üretmesini sağlamakla birlikte sistemin donanım uyumu açısından önemli bir rol oynamaktadır. Bu yapıların en temel iki örneği push-pull ve open-drain çıkışlardır.

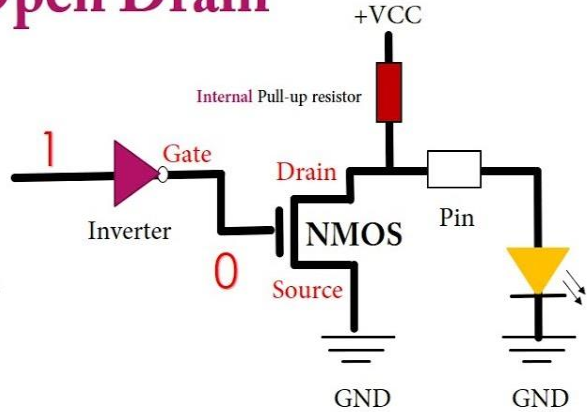
Push-pull çıkış tipi, bir mikrodenetleyici pininin hem lojik 1 (HIGH) hem de lojik 0 (LOW) seviyeye aktif olarak çekilebildiği bir yapıdır. Bu, pinin çıkışına bağlı bir sinyalin doğrudan +VCC (besleme) veya GND (toprak) seviyelerine sürülebilmesini sağlar. Donanım seviyesinde bu yapı, biri yukarı çeken (PMOS) ve diğeri aşağı çeken (NMOS) olmak üzere iki MOSFET’le gerçekleştirilir (Şekil 2.1). Bu yapı sayesinde çıkış sinyali hızlı bir şekilde 1 ve 0 olarak geçiş yapabilir. Push-pull çıkışlar; SPI gibi yüksek hızlı haberleşme protokollerinde, PWM sinyallerinin üretiminde veya LED, röle gibi doğrudan yüklerin sürülmesinde tercih edilir. Ayrıca harici bir pull-up direncine ihtiyaç duymadan çalışabilmeleri, devre tasarımını sadeleştiren önemli bir avantajdır.

Push Pull



Şekil 2.1

Open Drain



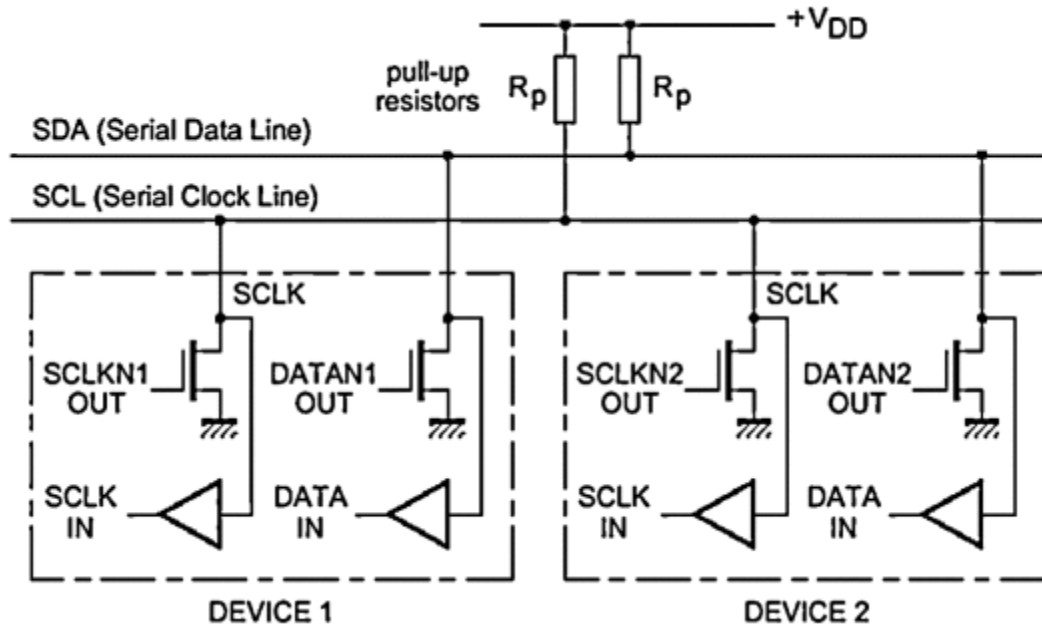
Şekil 2.2

Ancak push-pull yapılar, aynı veri hattını birden fazla cihazın paylaştığı durumlarda ciddi riskler barındırır. Eğer iki cihaz aynı anda aynı hatta zıt seviyeleri sürmeye çalışırsa, bu durum kısa devreye yol açar ve donanımda kalıcı hasarlar oluşturur. Bu nedenle, çoklu cihazların ortak bir hat üzerinden iletişim kurduğu sistemlerde push-pull yapıları kesinlikle kullanılmamalıdır.

Open-drain çıkış yapısı, mikrodenetleyici sistemlerinde özellikle birden fazla cihazın aynı veri hattını paylaştığı durumlarda tercih edilen güvenli bir haberleşme yöntemidir.(Şekil 2.2) Bu yapıda, pin yalnızca sinyali lojik 0’a, yani toprağa çekebilir; yüksek seviyeye geçiş ise doğrudan pin üzerinden değil,

harici bir pull-up direnci yardımıyla sağlanır. Donanımsal olarak bu çıkışlar, yalnızca bir NMOS içerir. MOSFET aktif olduğunda hat GND seviyesine çekilir, pasif hale geçtiğinde ise bağlı olduğu direnç tarafından stabil olarak lojik 1 seviyesinde tutulur.

Bu yaklaşımın temel avantajı, birden fazla cihazın aynı hat üzerinde çakışma yaşamadan haberleşebilmesidir. Çünkü hiçbir cihaz hattı aktif olarak yukarı çekmediği için, aynı anda birden fazla cihaz açık olsa dahi veri çakışması veya kısa devre gibi problemler oluşmaz. Özellikle I²C protokolü bu prensip üzerine kuruludur. Data (SDA) ve clock (SCL) hatları open-drain yapı ile çalışır ve hat üzerindeki seviyeler harici (pull-up) dirençlerle belirlenir (Şekil 2.3).



Şekil 2.3

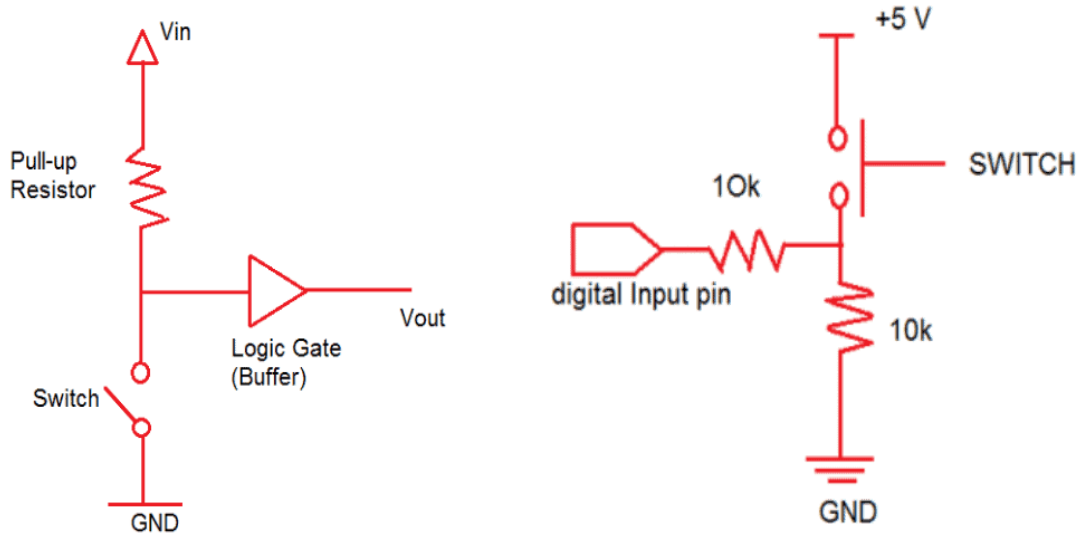
Open-drain yapının bir diğer önemli avantajı da farklı gerilim seviyelerinde çalışan bileşenler arasında doğrudan haberleşme imkânı sunmasıdır. Örneğin, 3.3V ile çalışan bir mikrodenetleyici, 5V seviyesinde çalışan bir cihazla doğrudan iletişim kurabilir. Bununla birlikte, bu yapı hız açısından bazı sınırlamalara sahiptir. Pull-up direncine bağlı olarak sinyalin yüksek seviyeye çıkışı pasif biçimde gerçekleştiğinden, geçiş süreleri uzayabilir. Bu nedenle open-drain çıkışlar, yüksek hız yerine güvenli haberleşmenin öncelikli olduğu sistemler için daha uygundur.

Open-drain ve benzeri çıkış yapılarında kullanılan pull-up dirençlerin değeri, hem haberleşme hızını hem de güç tüketimini doğrudan etkiler. Örneğin I2C iletişiminde tipik olarak 4.7 k Ω değerinde direnç kullanılırken, daha yüksek hız ve daha keskin sinyal geçişleri istenen durumlarda 2.2 k Ω veya daha düşük

direnç değeri tercih edilebilir. Direnç değerinin küçük olması, yükselme süresini kısaltırken, güç tüketimini artırabilir; bu nedenle seçim yapılırken uygulamanın gereksinimlerine göre seçim yapılmalıdır.

Open-collector çıkışlar ise, open-drain yapının BJT’li (bipolar junction transistor) versiyonudur. MOSFET yerine BJT kullanılarak tasarlanan bu yapı, çalışma prensibi bakımından open-drain ile tamamen aynıdır.

Mikrodenetleyici devrelerinde karşılaşılan diğer temel direnç konfigürasyonlarından biri de pull-down dirençleridir. Pull-up dirençlerin tam tersi olarak, hatları GND seviyesinde sabitlerler. Özellikle boşta kalan giriş pinlerinin istemsizce HIGH seviyeye geçmesini veya çevresel parazitlerden etkilenmesini önlemek ve kısa devre kaynaklı donanım bozulmalarının önüne geçmek için tercih edilirler (Şekil 2.4).



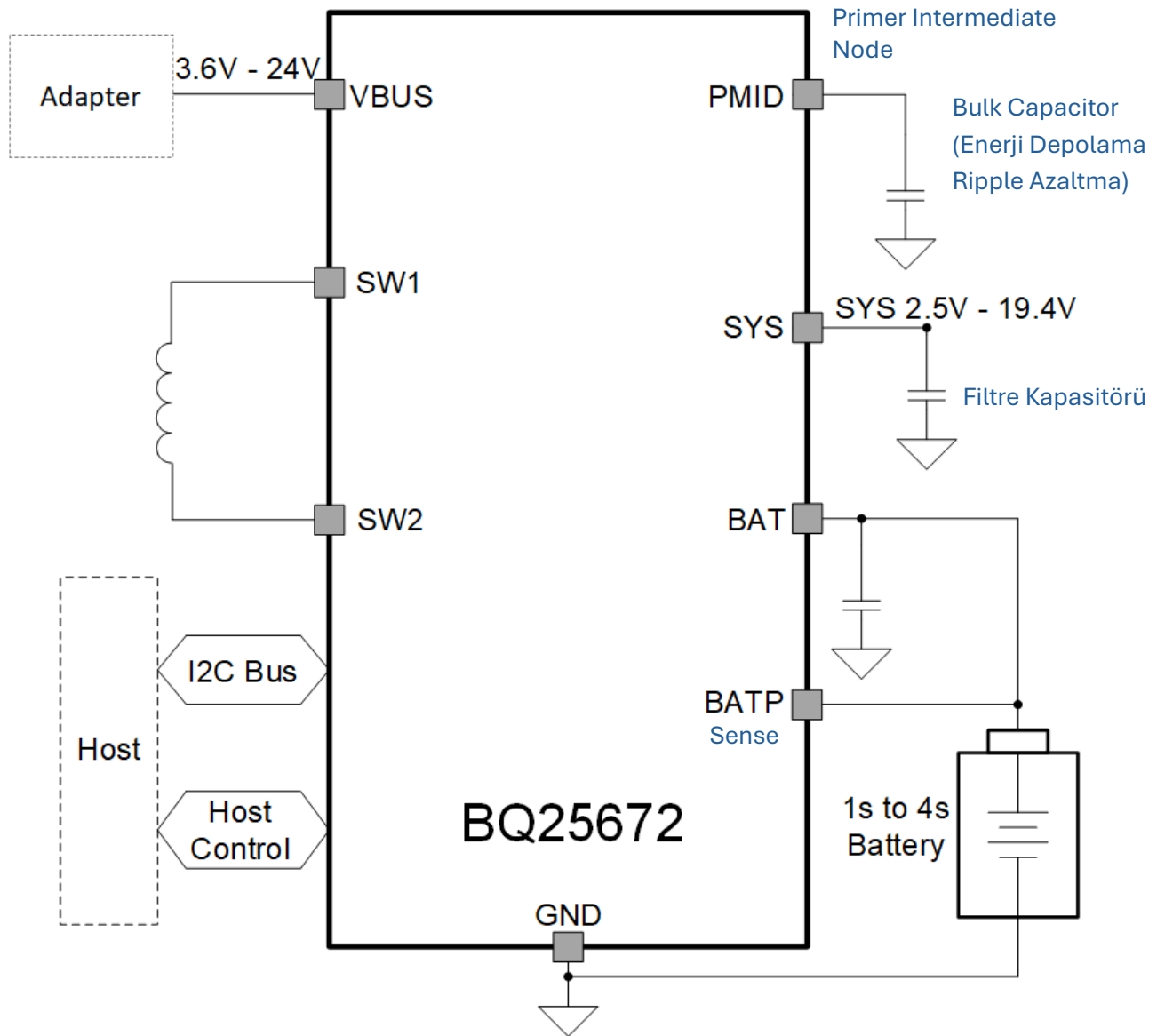
Şekil 2.4

Ayrıca bazı uygulamalarda mikrodenetleyici çıkışlarının “yüksek empedans” (Hi-Z) moduna alınması gerekebilir. Tri-state (üç durumlu) olarak da adlandırılan bu modda çıkış pininin hem HIGH hem de LOW seviyeye aktif olarak sürülmesi durdurulur ve pin tamamen pasif hale getirilir. Bu özellik özellikle çift yönlü veri hatlarında (örneğin I2C’deki SDA hattı gibi) farklı cihazlar arasında kontrol devri gerektiğinde kullanılır. Hi-Z modu, pinin devre dışı bırakılmış gibi davranmasını sağlayarak hattın başka bir cihaz tarafından kullanılmasına olanak tanır.

BQ25672 gibi entegreler, mikrodenetleyiciyle I2C protokolü üzerinden iletişim kurar. Bu protokolde kullanılan SDA ve SCL hatları open-drain yapısında

olduğundan, hatların güvenli ve sağlıklı çalışabilmesi için harici pull-up dirençler bağlanmalıdır. Bu yaklaşım, farklı seviyelerde çalışan cihazların uyumlu şekilde haberleşmesini sağlar.

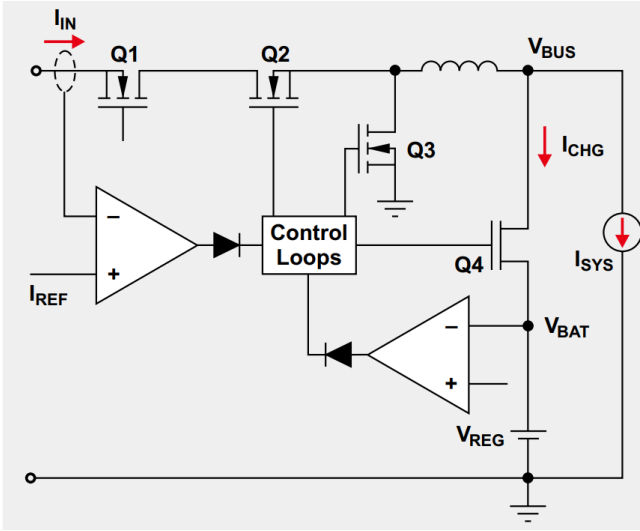
Konuya dair son bir özet olarak, mikrodenetleyici sistemlerinde çıkış yapı tiplerinin doğru seçimi, sistem performansı ve güvenliği açısından büyük bir öneme sahiptir. Push-pull çıkışlar hız ve kararlılık sağlarken, open-drain yapılar çoklu cihazlı ve gerilim seviyesi uyumsuzluklarının bulunduğu durumlar için ideal çözümler sunar. Bu doğrultuda sistem için gerekli olan hangisiyse o tercih edilmelidir.



Şekil 3.1

3- BQ25672 Genel Bakış

BQ25672 entegresi (Şekil 3.1), 1 ila 4 hücreli Li-ion ve Li-polymer bataryaları destekleyen gelişmiş bir şarj yönetim entegresidir. Seri bağlantı yapıldığında yaklaşık 4.2 V'tan 16.8 V'a kadar farklı bataryalarla uyumlu şekilde çalışabilir. Bu entegreyi öne çıkaran önemli bir avantaj, batarya ile sistem arasındaki bağlantıyı yöneten anahtarlama elemanının (BATFET) entegre içerisinde dahili olarak sunulmasıdır. Böylece harici MOSFET kullanımına ihtiyaç kalmaz; bu da yer ve maliyet tasarrufu sağlar. Entegre içindeki BATFET aynı zamanda batarya güvenliği, güç geçiş kontrolü ve sistem kararlılığı gibi konularda da rol oynar. Özellikle bataryanın sistemden izole edilmesi gereken aşırı akım, ters gerilim vb. durumlarda BATFET devreye girerek sistemin korunmasına katkı sağlar.



Şekil 3.2

Entegre giriş voltajı 3.6V ile 24V arasındaki bir giriş gerilimiyle çalışabilir ve anlık olarak 30 V'a kadar olan gerilimlere dayanabilir. Sistemde yer alan VINDPM (Input Voltage Dynamic Power Management) özelliği sayesinde, özellikle 22 V'a kadar olan giriş voltajı kontrol altında tutulabilir. Bu sayede giriş kaynağında ani gerilim değişimleri olsa bile entegre kendini ve bağlı

sistemleri koruyarak kararlı bir çalışma sağlar. Buna ek olarak, IINDPM (Input Current Limit) fonksiyonu sayesinde girişten çekilecek maksimum akım seviyesi 3.3 A'ya kadar sınırlandırılabilir. Bu da sistemin güç kaynağını aşırı yüklemekten çalışmasını ve daha güvenli bir enerji yönetimini mümkün kılar (Şekil 3.2).

BQ25672 entegresi, çoklu güç kaynaklarının yönetimini kolaylaştırmak ve sistem esnekliğini artırmak amacıyla entegre edilmiş bir Dual Input Power Mux Controller içerir. Bu yapı sayesinde, örneğin USB adaptörü ve harici bir DC kaynak gibi iki farklı girişten biri aktif olduğunda sistem bu kaynaklar arasında otomatik olarak geçiş yapabilir. Bu geçiş, tamamen donanım düzeyinde ve yazılım müdahalesine gerek olmadan gerçekleşebilir. Bu sayede, güç kaynağı erişilebilirliğinde yaşanabilecek dalgalanmalarda sistem çalışmaya kesintisiz şekilde devam eder.

BQ25672 aynı zamanda yazılım kontrollü sistemler için de büyük bir kolaylık sağlar. Entegre, hem otonom modda çalışabilir (default parametrelerle) hem de I²C arayüzü üzerinden yazılım ile yönetilebilir. I²C modda çalışırken, sistem geliştiricisi giriş akımı limiti, şarj voltajı, termal eşikler, batarya algılama gibi parametreleri değiştirebilir. Bu yapı sayesinde hem gerçek zamanlı kontrol hem de adaptif olarak enerji yönetimi sağlanmış olur.

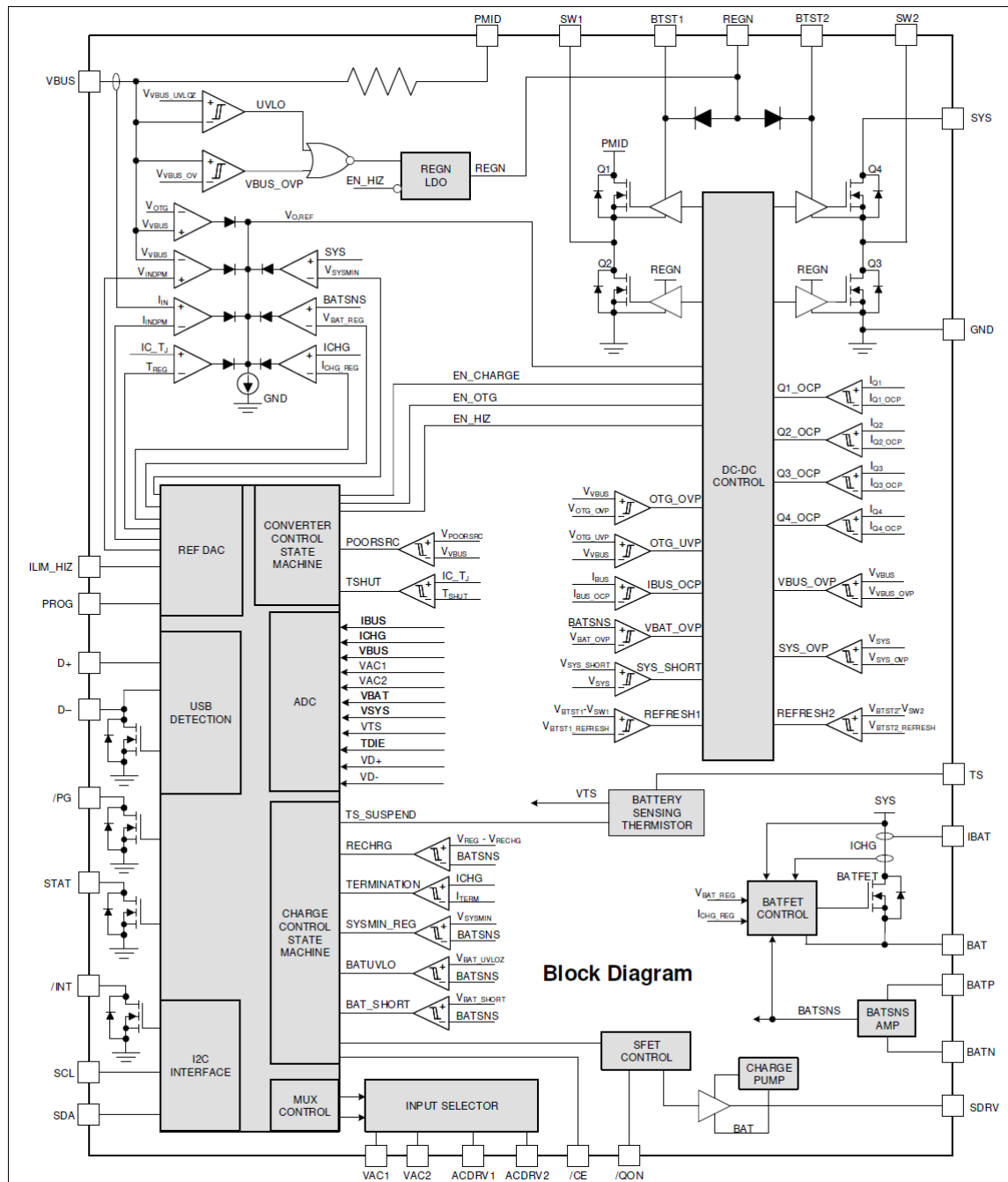
Entegre içerisinde dahili bir 16-bit ADC bulunur. Bu ADC sayesinde, sistem gerilimi, akımı ve sıcaklığı ölçülerek gerçek zamanlı izleme yapılabilir. Bu da enerji ve tehlike yönetimi açısından hassas kontrol imkânı sunar. Ayrıca, entegreye bağlanacak enerji kaynağı için geniş bir giriş voltajı aralığı desteklenmektedir. BQ25672, 5 V ila 12 V arasında çalışan farklı adaptör türleriyle uyumlu şekilde çalışabilir. Bu sayede hem düşük güçlü USB kaynakları hem de daha güçlü DC adaptörler tek bir devreyle desteklenebilir.

Entegrenin sunduğu önemli özelliklerden biri de MPPT (Maximum Power Point Tracking) desteğidir. Bu yapı değişken çıkış karakteristiğine sahip güç kaynaklarının kullanıldığı durumlarda sistem kararlılığı açısından fayda sağlar. MPPT algoritması, voltaj salınımı gösteren kaynaklardan sürekli ve mümkün olan en verimli gücü çekebilmek için giriş gerilimini ve akımını optimize eder. Yaygın olarak solar panellerde kullanılsa da yalnızca güneş panellerine özel bir teknoloji değil; dalgalanmalı veya düşük güçlü giriş kaynaklarının olduğu sistemlerde güvenilir enerji yönetimi sağlamak için oldukça değerli bir özelliktir.

Şekil 3.1’de görülen sistem diyagramı, BQ25672 entegresinin güç yönetim yapısına aittir. Adaptörden gelen 3.6 V – 24 V aralığındaki giriş gerilimi, VBUS pininden entegreye alınır ve dahili anahtarlama yapıları olan SW1 ve SW2 üzerinden işlenerek sistemin geri kalanına yönlendirilir. İlk etapta enerji, PMID adı verilen ara düğüm noktasına iletilir. Bu nokta, enerji aktarımında geçici depolama ve gerilim sabitleme görevleri üstlenir. Buraya bağlı bulk kapasitör, ani yük değişimlerinden kaynaklanan gerilim dalgalanmalarını (ripple) bastırır. Ardından güç, SYS pinine aktarılır; bu pin aracılığıyla sistem yükleri beslenir. SYS hattı, 2.5 V ila 19.4 V arasında çıkış sağlayabilir ve kararlı bir sistem gerilimi sunmak amacıyla burada ayrıca bir filtre kondansatörü kullanılır. Batarya şarjı içinse enerji, entegre içerisinden BAT pinine yönlendirilir. BAT terminaline bağlı şarj hattı, 1S ila 4S Li-ion bataryalarla doğrudan uyumludur. Entegre, bataryaya giden akımı kontrol altında tutmak amacıyla BATP pininden bir akım izleme (sense) hattı sağlar. Bu sayede hem şarj süreci yönetilir hem de

batarya güvenliği artırılmış olur. Öte yandan, mikrodenetleyici ile haberleşme I²C üzerinden sağlanır ve bu sayede sistemin parametreleri yazılımla da izlenip ayarlanabilir.

Sonuç olarak verilen tüm bilgiler BQ25672'nin nasıl çalıştığını genel hatlarıyla ortaya koysa da, entegre aslında çok daha karmaşık bir iç yapıya sahiptir. Bu yapının içinde bulunan tüm kontrol blokları, güvenlik devreleri ve güç yönetimi bileşenlerinin yer aldığı detaylı blok diyagramı Şekil 3.3'te gösterilmiştir. Bu diyagram sayesinde, entegrenin enerji akışını nasıl yönettiği ve sistemle nasıl etkileşim kurduğu daha net bir şekilde görülebilir.



Şekil 3.3

4- BQ25672 Driver Konfigürasyonları

BQ25672 entegresi ile haberleşme, I2C protokolü üzerinden gerçekleştirilmektedir ve entegre içindeki tüm kontrol (control), durum(status) ve yapılandırma(status) bilgileri ilgili register'lar ile yönetilmektedir. Bu kapsamda, datasheet esas alınarak BQ25672'nin sahip olduğu tüm register

Offset	Acronym	Register Name
0h	REG00_Minimal_System_Voltage	Minimal System Voltage
1h	REG01_Charge_Voltage_Limit	Charge Voltage Limit
3h	REG03_Charge_Current_Limit	Charge Current Limit
5h	REG05_Input_Voltage_Limit	Input Voltage Limit
6h	REG06_Input_Current_Limit	Input Current Limit
8h	REG08_Precharge_Control	Precharge Control
9h	REG09_Termination_Control	Termination Control
Ah	REG0A_Re-charge_Control	Re-charge Control
Bh	REG0B_VOTG_regulation	VOTG regulation
Dh	REG0D_IOTG_regulation	IOTG regulation
Eh	REG0E_Timer_Control	Timer Control
Fh	REG0F_Charger_Control_0	Charger Control 0
10h	REG10_Charger_Control_1	Charger Control 1
11h	REG11_Charger_Control_2	Charger Control 2
12h	REG12_Charger_Control_3	Charger Control 3
13h	REG13_Charger_Control_4	Charger Control 4
14h	REG14_Charger_Control_5	Charger Control 5
15h	REG15_MPPT_Control	MPPT Control
16h	REG16_Temperature_Control	Temperature Control
17h	REG17_NTC_Control_0	NTC Control 0
18h	REG18_NTC_Control_1	NTC Control 1
19h	REG19_ICO_Current_Limit	ICO Current Limit
1Bh	REG1B_Charger_Status_0	Charger Status 0
1Ch	REG1C_Charger_Status_1	Charger Status 1
1Dh	REG1D_Charger_Status_2	Charger Status 2
1Eh	REG1E_Charger_Status_3	Charger Status 3
1Fh	REG1F_Charger_Status_4	Charger Status 4
20h	REG20_FAULT_Status_0	FAULT Status 0
21h	REG21_FAULT_Status_1	FAULT Status 1
22h	REG22_Charger_Flag_0	Charger Flag 0
23h	REG23_Charger_Flag_1	Charger Flag 1
24h	REG24_Charger_Flag_2	Charger Flag 2
25h	REG25_Charger_Flag_3	Charger Flag 3
26h	REG26_FAULT_Flag_0	FAULT Flag 0
27h	REG27_FAULT_Flag_1	FAULT Flag 1
28h	REG28_Charger_Mask_0	Charger Mask 0
29h	REG29_Charger_Mask_1	Charger Mask 1
2Ah	REG2A_Charger_Mask_2	Charger Mask 2
2Bh	REG2B_Charger_Mask_3	Charger Mask 3
2Ch	REG2C_FAULT_Mask_0	FAULT Mask 0
2Dh	REG2D_FAULT_Mask_1	FAULT Mask 1
2Eh	REG2E_ADC_Control	ADC Control
2Fh	REG2F_ADC_Function_Disable_0	ADC Function Disable 0
30h	REG30_ADC_Function_Disable_1	ADC Function Disable 1
31h	REG31_IBUS_ADC	IBUS ADC
33h	REG33_IBAT_ADC	IBAT ADC
35h	REG35_VBUS_ADC	VBUS ADC
37h	REG37_VAC1_ADC	VAC1 ADC
39h	REG39_VAC2_ADC	VAC2 ADC
3Bh	REG3B_VBAT_ADC	VBAT ADC
3Dh	REG3D_VSYS_ADC	VSYS ADC
3Fh	REG3F_TS_ADC	TS ADC
41h	REG41_TDIE_ADC	TDIE_ADC
43h	REG43_D+_ADC	D+ ADC
45h	REG45_D-_ADC	D- ADC
47h	REG47_DPDM_Driver	DPDM Driver
48h	REG48_Part_Information	Part Information

Şekil 4.1

adresleri ve bu adreslere karşılık gelen offset değerleri detaylı şekilde çıkarılmıştır. Çalışmanın bu aşamasında, yazılım geliştirme sürecinde kullanılmak üzere tüm bu register adresleri ve offset değerleri, ilgili header dosyasında tanımlanarak sistematik bir yapıya kavuşturulmuştur.

Şekil 4.1'de yer alan tabloda , BQ25672 şarj entegresine ait tüm register adresleri ve offset değerleri yer almaktadır. Bu tablodan hareketle ilgili register'ların isimlendirilmesi ve işlevleri, doğrudan üretici datasheet'ine uygun şekilde düzenlenmiştir. Bu yapı sayesinde, sürücü geliştirme aşamasında donanım kontrolüne yönelik tüm erişimler yazılım içerisinde kolay ve anlaşılır biçimde sağlanabilmekte olup herhangi bir anlam karışıklığına sebebiyet verebilecek direkt adrese veri atama tarzında bir yaklaşımdan da uzak durulmuştur. Register alanlarının yönetimi yazılım tarafında daha okunabilir ve sürdürülebilir bir yapı oluşturmak adına bit bazında mask ve shift tanımlamaları kullanılarak gerçekleştirilmiştir. Header dosyasında her bir register alanı, ilgili bit konumlarını ve maske değerlerini açıkça tanımlayan makrolarla belirtilmiş ve yorum satırları eklenerek okunabilirlik artırılmıştır. Örneğin, REG05 register'ı için kullanılan Input Voltage Limit (VINDPM)

alanına ait mask ve shift yapısı Şekil 4.2’de yer alan datasheet’e uygun olarak aşağıdaki gibi tanımlanmıştır:

```
// REG05: Input Voltage Limit (VINDPM)
#define BQ25672_VINDPM_SHIFT 0U // Start bit of input V limit (VINDPM)
#define BQ25672_VINDPM_MASK (0xFFU << BQ25672_VINDPM_SHIFT)
// 8-bit mask for VINDPM (bits 7:0)
```

Figure 8-28. REG05_Input_Voltage_Limit Register

7	6	5	4	3	2	1	0
VINDPM_7:0							
R/W-24h							

Table 8-13. REG05_Input_Voltage_Limit Register Field Descriptions

Bit	Field	Type	Reset	Description
7-0	VINDPM_7:0	R/W	24h	<p>Absolute VINDPM Threshold</p> <p>VINDPM register is reset to 3600mV upon adapter unplugged and it is set to the value based on the VBUS measurement when the adapter plugs in. It is not reset by the REG_RST and the WATCHDOG</p> <p>Type : RW</p> <p>POR: 3600mV (24h)</p> <p>Range : 3600mV-22000mV</p> <p>Fixed Offset : 0mV</p> <p>Bit Step Size : 100mV</p> <p>Clamped Low</p>

Şekil 4.2

Bir diğer örnek olarak, Şekil 4.3’te yer alan REG10_Charger_Control_1 register’ı için de benzer şekilde bit alanları mask ve shift kullanılarak tanımlanmıştır. Bu register’a ait VAC_OVP, WD_RST ve WATCHDOG alanları için yapılan tanımlar aşağıdaki gibi yapılandırılmıştır:

Figure 8-37. REG10_Charger_Control_1 Register

7	6	5	4	3	2	1	0
R-0h		VAC_OVP_1:0		WD_RST	WATCHDOG_2:0		
R-0h		R/W-0h		R/W-0h	R/W-5h		

Table 8-22. REG10_Charger_Control_1 Register Field Descriptions

Bit	Field	Type	Reset	Notes	Description
7-6	RESERVED	R	0h		Reserved
5-4	VAC_OVP_1:0	R/W	0h	Reset by: REG_RST	<p>VAC_OVP thresholds</p> <p>Type : RW</p> <p>POR: 11b</p> <p>0h = 26V (default)</p> <p>1h = 22V</p> <p>2h = 12V</p> <p>3h = 7V</p>
3	WD_RST	R/W	0h	Reset by: WATCHDOG REG_RST	<p>I2C watch dog timer reset</p> <p>Type : RW</p> <p>POR: 0b</p> <p>0h = Normal (default)</p> <p>1h = Reset (this bit goes back to 0 after timer resets)</p>
2-0	WATCHDOG_2:0	R/W	5h	Reset by: REG_RST	<p>Watchdog timer settings</p> <p>Type : RW</p> <p>POR: 101b</p> <p>0h = Disable</p> <p>1h = 0.5s</p> <p>2h = 1s</p> <p>3h = 2s</p> <p>4h = 20s</p> <p>5h = 40s (default)</p> <p>6h = 80s</p> <p>7h = 160s</p>

Şekil 4.3

```
// REG10: Charger Control 1
#define BQ25672_VAC_OVP_SHIFT      4U
#define BQ25672_VAC_OVP_MASK      (0x3U << BQ25672_VAC_OVP_SHIFT) //bits 5:4

#define BQ25672_WD_RST_SHIFT      3U
#define BQ25672_WD_RST_MASK      (0x1U << BQ25672_WD_RST_SHIFT) //bit 3

#define BQ25672_WATCHDOG_SHIFT      0U
#define BQ25672_WATCHDOG_MASK      (0x7U << BQ25672_WATCHDOG_SHIFT) //bits 2:0
```

Bu yapı sayesinde, BQ25672 entegresi ile gerçekleştirilen register erişimlerinde doğrudan register'ın tamamı hex bir değer olarak yazılmaz veya okunmaz. Çünkü her bir register alanı bit bazında ayrı ayrı tanımlanan mask ve shift makroları aracılığıyla değer almaktadır. Yazılım geliştirme sürecinde her bir yapıya (charge_status) ait tüm parametreler, önceden oluşturulmuş config struct yapıları içerisinde saklanmakta ve her parametre ilgili bit pozisyonuna, tanımlı mask ve shift işlemleri kullanılarak yerleştirilmektedir. Bu işlemlerin ardından elde edilen register değeri, write_reg() fonksiyonu ile IC'ye yazılmaktadır.

Donanımdan bir register okunduğunda ise read_reg() fonksiyonu ile register içeriği bir değere atanır ve bu değerden ilgili alanlar, mask ve shift makroları ile ayrıştırılır. Ayrıştırılan bu değerler, ilgili struct değerlerine atanarak yazılım tarafında hem okunabilir hem de güvenilir bir biçimde kullanılabilir hale getirilir.

Bu yöntem sayesinde register erişimi sistematik ve hatasız bir yapıya oturtulmuş olur. Özellikle bir register içerisinde birden fazla bağımsız kontrol alanı (bit field) barındıran durumlarda, doğrudan hex değeri yazmak yerine her alanı ayrı ayrı struct üzerinden yöneterek yazılımın okunabilirliği, taşınabilirliği ve doğruluğu artırılır. İster datasheet'te tanımlı default konfigürasyonlar kullanılsın, ister kullanıcı tarafından yazılımla belirlenen özel konfigürasyonlar tercih edilsin, yapılandırma işlemlerinin tamamı struct tabanlı olarak gerçekleştirilmektedir.

Driver geliştirme sürecinde oluşturulan tüm fonksiyonlar temel olarak birer getter veya setter mantığıyla çalışmaktadır. Başlangıç aşamasında sistem konfigürasyonu, şarj parametreleri, USB-OTG ayarları, zamanlayıcılar, watchdog fonksiyonu, MPPT yapılandırması, NTC (JEITA) uyumluluğu, sıcaklık limitleri ve interrupt yönetimi alanlarına yönelik spesifik getter ve setter fonksiyonları oluşturulmuştur. Bu fonksiyonlar sayesinde, yazılım tarafında yapılan tüm yapılandırmalar donanıma doğru şekilde yazılmakta veya

donanımdan okunarak yazılım tarafında anlamlı ve kullanılabilir hâle getirilmektedir.

```
typedef struct
{
    bq25672_system_config_t    system;    ///< System behaviors: EN_HIZ, EN_TERM, ACDRV, PFM etc.
    bq25672_charge_config_t    charge;    ///< Charge settings: voltages, currents, input limits
    bq25672_usb_config_t       usb;       ///< USB OTG settings: IOTG, VOTG, EN_OTG
    bq25672_timer_config_t     timer;     ///< Charging timer enables: fast charge, precharge, top-off
    bq25672_watchdog_config_t  watchdog;  ///< I2C watchdog timer
    bq25672_mppt_config_t      mppt;      ///< Maximum Power Point Tracking parameters
    bq25672_ntc_config_t       ntc;       ///< JEITA NTC thresholds, TS comparator, OTG thresholds
    bq25672_temp_config_t      jeita;     ///< Temperature regulation (TREG), shutdown (TSHUT), pull-downs
    bq25672_interrupt_config_t interrupt;  ///< Charger and fault interrupt masks
}bq25672_config_t;
```

Sekil 4.4

Yazılım mimarisinde, bu yapıların en üst seviyesinde yer alan apply ve read fonksiyonları, kullanıcının oluşturduğu tüm yapılandırmaları (Şekil 4.4) donanıma topluca uygulamak veya donanımdan mevcut tüm yapılandırma parametrelerini bir seferde okumak amacıyla kullanılmaktadır. Bu yöntem sayesinde kullanıcı, sistemde tüm konfigürasyon değişikliklerini tek seferde düzenleyebilir veya mevcut yapılandırmayı kolayca okuyarak yazılım tarafında güncel hâle getirebilir.

Yazılımın sonunda konumlandırılan init fonksiyonu, kullanıcının yazılım tarafında tanımladığı yapılandırma değerlerini donanıma uygulamak için geliştirilmiştir. Yani, init fonksiyonu yazılım içerisinde oluşturulan config yapısının donanıma uygulanmasını sağlar. Bir diğer fonksiyon olan reset fonksiyonu ise, herhangi bir özel yapılandırma uygulanmadan önce entegreyi datasheet'te belirtilmiş olan default fabrika ayarlarına sıfırlamak amacıyla kullanılır. Soft ve hard olarak iki farklı çeşidi vardır. Reset fonksiyonu

```
bq25672_status_t bq25672_set_system_config(I2C_HandleTypeDef *hi2c, const bq25672_system_config_t *cfg);
bq25672_status_t bq25672_get_system_config(I2C_HandleTypeDef *hi2c, bq25672_system_config_t *cfg);

bq25672_status_t bq25672_set_charge_config(I2C_HandleTypeDef *hi2c, const bq25672_charge_config_t *cfg);
bq25672_status_t bq25672_get_charge_config(I2C_HandleTypeDef *hi2c, bq25672_charge_config_t *cfg);

bq25672_status_t bq25672_set_usb_config(I2C_HandleTypeDef *hi2c, const bq25672_usb_config_t *cfg);
bq25672_status_t bq25672_get_usb_config(I2C_HandleTypeDef *hi2c, bq25672_usb_config_t *cfg);

bq25672_status_t bq25672_set_timer_config(I2C_HandleTypeDef *hi2c, const bq25672_timer_config_t *cfg);
bq25672_status_t bq25672_get_timer_config(I2C_HandleTypeDef *hi2c, bq25672_timer_config_t *cfg);

bq25672_status_t bq25672_set_watchdog_config(I2C_HandleTypeDef *hi2c, const bq25672_watchdog_config_t *cfg);
bq25672_status_t bq25672_get_watchdog_config(I2C_HandleTypeDef *hi2c, bq25672_watchdog_config_t *cfg);

bq25672_status_t bq25672_set_mppt_config(I2C_HandleTypeDef *hi2c, const bq25672_mppt_config_t *cfg);
bq25672_status_t bq25672_get_mppt_config(I2C_HandleTypeDef *hi2c, bq25672_mppt_config_t *cfg);

bq25672_status_t bq25672_set_ntc_config(I2C_HandleTypeDef *hi2c, const bq25672_ntc_config_t *cfg);
bq25672_status_t bq25672_get_ntc_config(I2C_HandleTypeDef *hi2c, bq25672_ntc_config_t *cfg);

bq25672_status_t bq25672_set_temp_config(I2C_HandleTypeDef *hi2c, const bq25672_temp_config_t *cfg);
bq25672_status_t bq25672_get_temp_config(I2C_HandleTypeDef *hi2c, bq25672_temp_config_t *cfg);

bq25672_status_t bq25672_set_interrupt_config(I2C_HandleTypeDef *hi2c, const bq25672_interrupt_config_t *cfg);
bq25672_status_t bq25672_get_interrupt_config(I2C_HandleTypeDef *hi2c, bq25672_interrupt_config_t *cfg);

bq25672_status_t bq25672_read_ibus_current_ma(I2C_HandleTypeDef *hi2c, int16_t *ma);
bq25672_status_t bq25672_read_vbus_voltage_mv(I2C_HandleTypeDef *hi2c, uint16_t *mv);
bq25672_status_t bq25672_read_ibat_current_ma(I2C_HandleTypeDef *hi2c, int16_t *ma);
bq25672_status_t bq25672_read_vbat_voltage_mv(I2C_HandleTypeDef *hi2c, uint16_t *mv);

bq25672_status_t bq25672_get_device_status(I2C_HandleTypeDef *hi2c, bq25672_device_status_t *status);
bq25672_status_t bq25672_get_fault_status(I2C_HandleTypeDef *hi2c, bq25672_fault_status_t *status);

bq25672_status_t bq25672_set_charging_state(I2C_HandleTypeDef *hi2c, bool enabled);
bq25672_status_t bq25672_get_charging_state(I2C_HandleTypeDef *hi2c, bool *enabled);
bq25672_status_t bq25672_set_hiz_state(I2C_HandleTypeDef *hi2c, bool enabled);
bq25672_status_t bq25672_get_hiz_state(I2C_HandleTypeDef *hi2c, bool *enabled);

bq25672_status_t bq25672_apply_config(I2C_HandleTypeDef *hi2c, const bq25672_config_t *cfg);
bq25672_status_t bq25672_read_config(I2C_HandleTypeDef *hi2c, bq25672_config_t *out);

bq25672_status_t bq25672_init(I2C_HandleTypeDef *hi2c, const bq25672_config_t *cfg);
bq25672_status_t bq25672_soft_reset(I2C_HandleTypeDef *hi2c);
```

Sekil 4.5

kullanıldığında sistem, BQ25672'nin donanım default değerlerine geri döner; init fonksiyonu kullanıldığında ise yazılım tarafından belirlenen yeni değerler donanıma uygulanır.

Bu fonksiyonların tamamı Şekil 4.5'de verilmiştir, ilerleyen bölümlerde tüm fonksiyonlar detaylı açıklamalar ve örneklerle sunulacaktır.

İlk olarak geliştirilen bq25672_set_system_config ve bq25672_get_system_config fonksiyonları doğrudan BQ25672'nin REG0F (Charger Control 0), REG12 (Charger Control 3) ve REG13 (Charger Control 4) register alanlarına yönelik tasarlanmıştır. Özellikle sistem davranışını belirleyen temel parametrelerin bu üç register üzerinden yönetiliyor olması, sistem konfigürasyonunu ayrı bir yapıda ele almayı mantıklı kılmıştır.

bq25672_set_system_config fonksiyonu, yazılımla tanımlanan bq25672_system_config_t yapılandırmasını (Şekil 4.6) temel alarak ilgili değerleri donanıma aktarır. REG0F altında yer alan EN_TERM, EN_HIZ, FORCE_ICO, EN_ICO, REG12 altında PFM devre dışı bırakma, LDO devre dışı bırakma, OOA devre dışı bırakma, REG13 altında ise ACDRV sürücüleri ve PWM frekans seçimi gibi alanlar doğrudan bu yapı üzerinden kontrol edilmektedir.

```
typedef struct
{
    bool enable_termination; ///< Enable battery charge termination
    bool enable_hiz;          ///< Enable HIZ mode (zero input current)
    bool force_ico;           ///< Force input current optimization
    bool enable_ico;          ///< Enable ICO mode
    bool disable_pfm;         ///< Disable Pulse Frequency Modulation
    bool disable_ldo;         ///< Disable internal LDO regulator
    bool disable_ooa;         ///< Disable out-of-audio-mode (light load optimization)
    bool use_1500kHz;         ///< Use 1.5 MHz switching frequency instead of 750 kHz
    bool enable_acdrv1;       ///< Enable external ACDRV1 gate driver
    bool enable_acdrv2;       ///< Enable external ACDRV2 gate driver
} bq25672_system_config_t;
```

Şekil 4.6

Tersine işlem gerçekleştiren bq25672_get_system_config fonksiyonu ise, donanımdaki mevcut yapıyı okuyarak yazılıma aktarmak amacıyla kullanılmaktadır. Bu fonksiyon, öncelikle donanımdan ilgili register'ları okur, ardından her bit alanını tanımlanmış mask ve shift yapılarıyla ayrıştırır ve kullanıcının yazılım tarafındaki yapılandırma yapısına (bq25672_system_config_t) eksiksiz şekilde aktarır. Bu sayede sistemin anlık donanım durumu yazılım tarafına aktarılıp gerektiğinde güncellenebilir.

Sistem konfigürasyon fonksiyonlarının ardından, şarj parametrelerine yönelik yapılandırmalar için bq25672_set_charge_config ve bq25672_get_charge_config isimli fonksiyonlar geliştirilmiştir. Bu fonksiyonlar, şarj voltajı, şarj akımı, giriş voltajı ve akım sınırları gibi temel parametreleri yapılandırmak amacıyla kullanılmaktadır. Bu yapılandırma, Şekil 4.7'de gösterilen bq25672_charge_config_t isimli yapı üzerinden gerçekleştirilmiştir. İlgili yapı içerisinde şarj voltajı, şarj akımı, giriş tarafı için

maksimum/minimum gerilim ve akım limitleri, bitiş (termination) akımı ve precharge yüzdesi gibi temel parametreler tanımlanabilmektedir.

```
typedef struct
{
    uint16_t charge_voltage_max_mv;    // Charge voltage limit in millivolts (e.g., 8400 = 8.4V)
    uint16_t charge_current_max_ma;    // Charge current limit in milliamperes (e.g., 1000 = 1A)

    struct
    {
        uint16_t input_voltage_max_mv;    // VINDPM: maximum input voltage (mV), e.g., 9000
        uint16_t input_voltage_min_mv;    // VBUS UVLO: minimum input voltage (mV), usually 3200-3600
        uint16_t input_current_max_ma;    // IINDPM: maximum input current (mA), e.g., 2000
        uint8_t termination_current_ma;    // ITERM: termination current threshold (mA), e.g., 50
        bq25672_precharge_pct_t precharge_pct; // VRECHG: recharge threshold as percentage of full voltage
    } advanced;
} bq25672_charge_config_t;
```

Sekil 4.7

bq25672_set_charge_config fonksiyonu, kullanıcı tarafından yapılandırılan bq25672_charge_config_t struct yapısına ilgili parametreleri datasheet'te belirtilen register alanlarına yazar. Bu kapsamda REG01 ve REG02 register'ları şarj voltaj limiti için kullanılırken, REG03 ve REG04 ise şarj akım limiti değerlerini taşımaktadır. Her iki parametre de LSB ve MSB olmak üzere iki ayrı register'a high ve low kısımları için bölünmüş şekilde yazılmaktadır. Şarj voltajı için 10 mV, şarj akımı için ise 10 mA adımlı bir yapı kullanılmıştır.

Giriş gerilimi ve akım limitleri için sırasıyla REG05 ve REG06 register'ları kullanılmıştır. REG05 üzerinden VINDPM (Input Voltage Dynamic Power Management) limiti yapılandırılırken, default olarak 3600 mV'den başlayan 100 mV adımlı bir yapı kullanılmıştır. Benzer şekilde REG06, giriş akım limitini 10 mA adımlar ile tanımlayan IINDPM alanını barındırmaktadır.

Şarj sonlandırma akımı (ITERM) ve precharge eşik yüzdesi (VRECHG) gibi parametreler REG09 register'ı üzerinden yapılandırılır. Termination akımı 40 mA adımlarla belirlenirken, precharge yüzdesi ise enum tipi (bq25672_precharge_pct_t) ile %15-%71.4 arası değerler üzerinden seçilebilmektedir (Şekil 4.8). Ek olarak, minimum sistem gerilimi için REG00 kullanılır ve default olarak 2500 mV'den başlayan 250 mV adımlı bir yapı mevcuttur.

```
typedef enum
{
    BQ25672_PRECHARGE_15PCT = 0x0,    // Pre-charge to 15% of charge voltage
    BQ25672_PRECHARGE_62P2 = 0x1,    // Pre-charge to 62.2% of charge voltage
    BQ25672_PRECHARGE_66P7 = 0x2,    // Pre-charge to 66.7% of charge voltage
    BQ25672_PRECHARGE_71P4 = 0x3,    // Pre-charge to 71.4% of charge voltage
} bq25672_precharge_pct_t;
```

Sekil 4.8

Tersine çalışan bq25672_get_charge_config fonksiyonu ise, donanımdaki mevcut yapıyı yazılıma almak için kullanılır. İlgili register'lerden okunan değerler, doğru biçimde bq25672_charge_config_t yapısına atanır. Bu fonksiyon sayesinde donanımın anlık şarj konfigürasyonu yazılım tarafından doğru şekilde okunabilir ve kontrol edilebilir hale gelir.

İlgili sistem ve şarj yapılandırmalarının ardından USB-OTG (On-The-Go) fonksiyonlarına yönelik yapılandırma fonksiyonları geliştirilmiştir. Bu kapsamda bq25672_set_usb_config ve bq25672_get_usb_config isimli fonksiyonlar, OTG çıkış gerilimi, çıkış akımı ve OTG çalışma modları gibi parametrelerin değerlerinin değiştirilmesini ve değerlerinin alınmasını sağlar. Yapılandırma işlemleri Şekil 4.9'da gösterilen bq25672_usb_config_t yapısı üzerinden gerçekleştirilmiştir.

```
typedef struct
{
    bool enable_otg;          ///< Enable OTG
    uint16_t voltage_mv;      ///< VOTG target (mV)
    uint16_t current_max_ma;  ///< IOTG limit (mA)
    struct
    {
        bool disable_pfm;    ///< Disable PFM in OTG
        bool disable_oaa;    ///< Disable OOA in OTG
    } advanced;
} bq25672_usb_config_t;
```

Şekil 4.9

bq25672_set_usb_config fonksiyonu, OTG çıkış gerilimini REG0B üzerinden yapılandırır. Bu register 2800 mV başlangıçlı ve 10 mV adımlı şekilde çalışmaktadır. Kod içerisinde milivolt cinsinden olan değer, uygun bit alana dönüştürülerek yazılmaktadır. Benzer şekilde çıkış akımı limiti REG0D register'ı üzerinden yapılandırılmakta olup 40 mA adımlı bir yapı kullanılmaktadır. OTG modunun PFM (Pulse Frequency Modulation) ve OOA (Out-of-Audible) modları REG12 register alanları üzerinden devre dışı bırakılabilmekte, ilgili yapılandırma da struct içerisindeki advanced alanı ile kontrol edilmektedir. OTG çıkış fonksiyonunun etkinleştirilmesi veya devre dışı bırakılması ise REG0F register'ı üzerinden sağlanmaktadır. bq25672_get_usb_config fonksiyonu da yine aynı mantıkla donanımdan mevcut yapılandırmaları okur ve struct içerisine aktarır.

USB-OTG yapılandırmalarının ardından, şarj sürecinde zamanlayıcı konfigürasyonları için bq25672_timer_config_t yapısı oluşturulmuştur (Şekil 4.10).

```
typedef struct
{
    bq25672_topoff_time_t    top_off_time;        ///< Top-off timer duration
    bool                     trickle_enable;        ///< Enable trickle timer (1h)
    bool                     fast_charge_enable;    ///< Enable fast-charge timer
    bq25672_fastchg_time_t   fast_charge_time;     ///< Fast-charge timer duration
    struct
    {
        bool precharge_enable; ///< Enable pre-charge timer
        bool timer_slow;       ///< Enable timer slowdown (2x mode under DPM / Thermal)
    } advanced;
} bq25672_timer_config_t;
```

Şekil 4.10

```
*typedef enum
{
    BQ25672_TOPOFF_DISABLED = 0,
    BQ25672_TOPOFF_15MIN   = 1,
    BQ25672_TOPOFF_30MIN   = 2,
    BQ25672_TOPOFF_45MIN   = 3
} bq25672_topoff_time_t;

*typedef enum
{
    BQ25672_FASTCHG_5H  = 0,
    BQ25672_FASTCHG_8H  = 1,
    BQ25672_FASTCHG_12H = 2,
    BQ25672_FASTCHG_24H = 3
} bq25672_fastchg_time_t;
```

Şekil 4.11

Bu yapı ile trickle, pre-charge, fast-charge ve top-off gibi şarj süreleri detaylı şekilde kontrol edilebilmektedir. Top-off zamanlayıcısı için bq25672_topoff_time_t enum yapısı kullanılmış olup, 15, 30, 45 dakika gibi süre seçenekleri sunulmuştur. Benzer şekilde, fast-charge süresi için bq25672_fastchg_time_t enum yapısı kullanılmış ve 5, 8, 12, 24 saat seçenekleri sunulmuştur (Şekil 4.11).

Timer konfigürasyonlarında ayrıca DPM (Dynamic Power Management) ve termal koşullar altında zamanlayıcıların 2x yavaşlatılması gibi gelişmiş özellikler advanced alanı altında sunulmuştur. Pre-charge ve trickle modları için enable/disable seçimleri yapılabilmekte, böylece sistemin tüm zamanlayıcı davranışları kod ile kontrol edilebilmektedir. Tüm bu işlemler REG0A (Timer Control) register'ı üzerinden gerçekleştirilmektedir.

Timer yapılandırmalarının ardından sistemde watchdog işlevlerine yönelik yapılandırmalar gerçekleştirilmiştir. Bu yapı, özellikle I²C haberleşmesinde yaşanabilecek yazılım hatalarına veya sistem donmalarına karşı bir tür koruma mekanizması görevi görür. Bu amaçla geliştirilen bq25672_watchdog_config_t isimli yapı sayesinde watchdog timeout süresi, anlık WD reset ihtiyacı ve timeout sonrası şarjın kesilmesi gibi sistem davranışları kullanıcı tarafından kontrol edilebilmektedir (Şekil 4.12).

```
typedef struct
{
    bq25672_watchdog_timeout_t timeout;    ///< Watchdog timeout period
    bool reset_watchdog_now;              ///< Force immediate watchdog reset (REG10)
    bool disable_on_expiry;               ///< Stop charging when WDT expires (REG09)
} bq25672_watchdog_config_t;
```

Şekil 4.12


```
typedef enum
{
    BQ25672_WATCHDOG_DISABLE = 0x0, ///< Disable
    BQ25672_WATCHDOG_0_5S    = 0x1, ///< 0.5 seconds
    BQ25672_WATCHDOG_1S      = 0x2, ///< 1 second
    BQ25672_WATCHDOG_2S      = 0x3, ///< 2 seconds
    BQ25672_WATCHDOG_20S     = 0x4, ///< 20 seconds
    BQ25672_WATCHDOG_40S     = 0x5, ///< 40 seconds
    BQ25672_WATCHDOG_80S     = 0x6, ///< 80 seconds
    BQ25672_WATCHDOG_160S    = 0x7, ///< 160 seconds
} bq25672_watchdog_timeout_t;
```

Şekil 4.13

Watchdog timeout süreleri, bq25672_watchdog_timeout_t enum yapısı ile değer alır (Şekil 4.13). Yazılım tarafında 0.5 saniyeden başlayarak 160 saniyeye kadar çeşitli süre seçenekleri sunulmuştur. Bu süreler, sistemin

watchdog yenilenmeden ne kadar süre boyunca beklemesine müsaade edileceğini tanımlar. Timeout süresi dolduğunda ve watchdog resetlenmemişse, cihaz dahili register yapılandırmalarını sıfırlamakta veya opsiyonel olarak şarj işlemini durdurmaktadır.

Bu yapılandırma donanım tarafında REG10 (Charger Control 1) ve REG09 (Termination Control) register'ları üzerinden gerçekleştirilmektedir. Timeout süresi ve anlık reset kontrolü REG10 konfigürasyonu ile sağlanırken, timeout sonrası şarjın durdurulup durdurulmayacağı REG09 konfigürasyonu ile belirlenmektedir.

Yazılım tarafında bq25672_set_watchdog_config() fonksiyonu, kod ile belirlenen yapılandırma parametrelerini IC'ye iletirken, bq25672_get_watchdog_config() fonksiyonu ise mevcut yapılandırmaları okuyarak yazılıma aktarır. Böylece sistemin watchdog davranışı dinamik bir şekilde yönetilebilir ve tüm kontrol parametreleri tek bir yapı üzerinden düzenlenebilir.

Watchdog konfigürasyonunun ardından, BQ25672 entegresinin Maximum Power Point Tracking (MPPT) yapısı için fonksiyonlar geliştirilmiştir. MPPT fonksiyonu, özellikle değişken enerji kaynaklarından (örneğin solar paneller veya zayıf adaptörler) mümkün olan en verimli şekilde enerji çekebilmek için kullanılmaktadır. Yazılım tarafında bu yapı bq25672_mppt_config_t isimli yapı ile temsil edilmiştir (Şekil 4.14).

```
typedef struct {
    bool enable_mppt;           ///< Enable MPPT VOC measurement
    bq25672_mppt_pct_t ratio;   ///< VOC target ratio

    struct {
        uint32_t delay_ms;      ///< Delay after switching (0: 50ms, 1: 300ms, 2: 2s, 3: 5s)
        uint32_t interval_s;    ///< VOC measurement interval (0: 30s, 1: 2min, 2: 10min, 3: 30min)
    } advanced;
} bq25672_mppt_config_t;
```

Şekil 4.14

BQ25672'nin MPPT fonksiyonunun tüm ayarlamaları REG15 register'ı üzerinden sağlanmaktadır. Bu yapı üzerinden VBUS hattının açık devre

gerilimine göre hedef oranı belirlenebilir, VOC ölçümü öncesi kısa bir bekleme süresi tanımlanabilir ve bu ölçümlerin ne sıklıkla yapılacağı zamanlanabilir. Ayrıca, MPPT fonksiyonunun kod tarafından aktif ya da pasif edilmesi mümkündür.

Yazılım fonksiyonları, donanımın izin verdiği tüm bu parametreleri kullanıcı tarafından belirlenen değerlere göre konfigüre eder. MPPT oranı, bekleme süresi ve ölçüm sıklığı gibi alanlar, donanım tarafında belirli enum değerleri ile (saniye bazlı süreler veya yüzde oranları) seçilebilmektedir. Yapılan okuma ve yazma işlemleri sonucunda sistem, belirlenen yapılandırmaya göre enerji kaynağından en verimli şekilde güç çekeabilmekte ve dinamik koşullara uyum sağlayabilmektedir. BQ25672'nin sunduğu bu MPPT yapısı sayesinde, yalnızca sabit adaptör kaynaklarında değil, değişken karakterdeki güç kaynaklarında da sistem verimliliği ve stabilitesi korunabilmektedir.

Ardından geliştirilen NTC (Negative Temperature Coefficient) ve sıcaklık kontrol yapısı, BQ25672'nin sıcaklık odaklı koruma ve davranışlarını kod ile kontrol etmeyi hedeflemektedir. Bu amaçla oluşturulan bq25672_ntc_config_t yapısı (Şekil 4.15), batarya şarj sürecinde sıcaklığa bağlı parametrelerin ayarlanmasına imkân tanır. Söz konusu yapı donanım tarafında REG17 (NTC Control 0) ve REG18 (NTC Control 1) register alanlarına karşılık gelmektedir.

```
typedef struct
{
    bq25672_ntc_vreduce_t      vreg_reduce_mv; ///< JEITA High Temp VREG reduction
    bq25672_ntc_ichg_reduce_t  ichg_hot_reduce; ///< JEITA High Temp current reduction
    bq25672_ntc_cold_current_t ichg_cold_reduce; ///< JEITA Low Temp current reduction

    uint8_t ts_cool_deg; ///< TS cool threshold (rising)
    uint8_t ts_warm_deg; ///< TS warm threshold (falling)
    uint8_t bhot_deg;    ///< OTG BHOT threshold
    bool    bcold_20c;   ///< OTG BCOLD threshold: 0 = -10°C, 1 = -20°C
    bool    ignore_ts;   ///< Ignore TS comparators entirely
} bq25672_ntc_config_t;
```

Şekil 4.15

Bu yapı üzerinden yüksek sıcaklıklarda şarj gerilimi düşürme (VREG reduction), yüksek ve düşük sıcaklıklarda şarj akımı azaltma oranları gibi JEITA standardına uygun davranışlar datasheet'e uygun enumlar ile yapılandırılmaktadır. Ayrıca, soğuk ve sıcak eşik değerleri (TS Cool/Warm), OTG modunda BHOT ve BCOLD eşikleri gibi detaylar da bu yapı üzerinden tanımlanmıştır. Buradaki seçimler yazılım tarafında sıcaklık derecesi değeri olarak girilip kod içerisinde yine datasheet'e uygun bir şekilde set edilmektedir.

Donanım tarafında REG17 ve REG18 register'larına yazılan bu parametreler ile BQ25672 entegresi sıcaklığa bağlı olarak sistem koruması ve şarj verimliliğini optimize edebilmektedir. Örneğin, sıcaklık yükseldiğinde şarj akımı veya gerilimi otomatik olarak düşürülerek bataryanın güvenliği sağlanır ve bu çok önemli bir kontrol sağlar.

Buna ek olarak, sistemin genel sıcaklık davranışını kontrol etmek amacıyla bq25672_temp_config_t isimli bir yapı daha oluşturulmuştur (Şekil 4.16). Bu yapı, REG16 (Temperature Control) register'ını konfigüre etmektedir. İlgili yapı üzerinden termal regülasyon eşiği (default 120°C) ve termal kapanma eşiği (default 150°C) tanımlanabilmektedir. Ayrıca, VBUS, VAC1 ve VAC2 giriş hatlarında pull-down dirençlerinin aktif edip edilmeyeceği yine bu yapı ile belirlenmiştir.

```
typedef struct
{
    bq25672_temp_regulation_t treg;    ///< Thermal regulation threshold (°C)
    bq25672_temp_shutdown_t tshut;    ///< Thermal shutdown threshold (°C)
    bool vbus_pd_en;                  ///< Enable VBUS pull-down resistor
    bool vac1_pd_en;                  ///< Enable VAC1 pull-down resistor
    bool vac2_pd_en;                  ///< Enable VAC2 pull-down resistor
} bq25672_temp_config_t;
```

Şekil 4.16

Bu iki yapı sayesinde BQ25672'nin sıcaklık odaklı tüm fonksiyonları, kullanıcıya yazılım üzerinden sade, okunabilir ve sürdürülebilir bir şekilde sunulmuş; donanım katmanı ile yazılım katmanı arasındaki register kısmında soyutlama başarıyla sağlanmıştır.

BQ25672 entegresinde yer alan kesme (interrupt) yönetimi, şarj ve hata durumlarının yazılım ile kolayca kontrolü amacıyla kapsamlı şekilde yapılandırılmıştır. Bu kapsamda geliştirilen bq25672_interrupt_config_t yapısı (Şekil 4.17), donanımdaki kesme kaynaklarının yazılım üzerinden kontrolünü sağlar. İlgili yapı, REG28 (Charger Event Masks) ve REG2C (Fault Event Masks) register alanlarını temel alır.

Bu yapı üzerinden iki ana kesme grubu kontrol edilmektedir. İlk grup şarj olayları (charger events) ile ilgilidir ve REG28 register'ı üzerinden yönetilir. REG28'de yeralanlar; giriş akım limiti (IINDPM), giriş voltaj limiti (VINDPM), watchdog zamanlayıcısı, zayıf kaynak algılama, güç iyi (power good), VAC1 ve VAC2 hatlarının varlığı ve VBUS algılama gibi sistemin çalışma durumlarını izleyen kesmelerdir. Her bir bit, 0 olduğunda ilgili

kesmenin aktif olduğunu, 1 olduğunda ise maskelendiğini (devre dışı olduğunu) belirtmektedir.

```
typedef struct
{
    /* Charger Event Interrupt Masks (REG28) */
    bool iindpm;        ///< Enable IINDPM / IOTG interrupt
    bool vindpm;        ///< Enable VINDPM / VOTG interrupt
    bool watchdog;      ///< Enable I2C watchdog timer interrupt
    bool poor_source;   ///< Enable Poor Source interrupt
    bool power_good;    ///< Enable Power Good interrupt
    bool ac2_present;   ///< Enable VAC2 Present interrupt
    bool ac1_present;   ///< Enable VAC1 Present interrupt
    bool vbus_present;  ///< Enable VBUS Present interrupt

    /* Fault Event Interrupt Masks (REG2C) */
    bool ibat_reg;      ///< Enable IBAT regulation fault interrupt
    bool vbus_ovp;      ///< Enable VBUS over-voltage fault interrupt
    bool vbat_ovp;      ///< Enable VBAT over-voltage fault interrupt
    bool ibus_ocp;      ///< Enable IBUS over-current fault interrupt
    bool ibat_ocp;      ///< Enable IBAT over-current fault interrupt
    bool conv_ocp;      ///< Enable Converter over-current fault interrupt
    bool vac2_ovp;      ///< Enable VAC2 over-voltage fault interrupt
    bool vac1_ovp;      ///< Enable VAC1 over-voltage fault interrupt
} bq25672_interrupt_config_t;
```

Şekil 4.17

İkinci grup ise hata olayları (fault events) ile ilgilidir ve REG2C register'ı üzerinden yönetilir. Batarya akımı regülasyon hatası (IBAT regulation fault), VBUS ve VBAT aşırı gerilim (OVP), IBUS ve IBAT aşırı akım (OCP), konvertör aşırı akım hatası ve VAC1/VAC2 hatlarının aşırı gerilim durumları gibi donanımsal koruma mekanizmalarına yönelik kesme kaynakları yer alır.

bq25672_set_interrupt_config() fonksiyonu ile kullanıcının yapılandığı struct içeriği donanıma yazılırken; bq25672_get_interrupt_config() fonksiyonu ile donanımdaki mevcut yapı okunarak struct yapısı güncellenmektedir. Bu yapı sayesinde hangi kesmelerin aktif olduğunun ve hangilerinin devre dışı bırakıldığının kontrolü sağlanmaktadır. BQ25672'nin sağladığı kesme yönetimi sayesinde, hem sistem olayları hem de güvenlik odaklı hatalar detaylı şekilde takip edilmekte ve gerekli müdahaleler yazılım seviyesinde hızlıca gerçekleştirilebilmektedir.

Sonuç olarak tüm bu yapıların ve fonksiyonların geliştirilmesinin temel amacı, daha önce Şekil 4.4'te gösterilen bq25672_config_t ana yapısı altında, BQ25672 entegresinin tüm konfigürasyon alanlarının yazılım seviyesinde tek bir yapı içerisinde tutulmasını sağlamaktır. Bu yapı içerisinde, sistem, şarj,

USB-OTG, zamanlayıcılar, watchdog, MPPT, NTC (JEITA), sıcaklık kontrolleri ve interrupt maskeleri gibi her fonksiyonel blok için ayrı ayrı tanımlanmış alt yapılar yer almaktadır. Bu sayede tüm donanım konfigürasyonu tek bir yapı üzerinden yönetilebilmektedir.

bq25672_read_config() fonksiyonu ile entegre üzerindeki tüm register alanları donanımdan okunarak ana yapı içerisine aktarılır. Kullanıcı, yazılım üzerinden tüm sistemin mevcut durumunu görebilir ve bu yapı üzerinde değişiklik yaparak yeni bir konfigürasyon oluşturabilir. Benzer şekilde bq25672_apply_config() fonksiyonu sayesinde, yazılımda tanımlanan tüm parametreler, donanım register'larına kolayca yazılır. Böylece sistem başlatılırken veya konfigürasyon değişikliği gerektiğinde, tüm yapı tek bir fonksiyonla hızlı ve güvenli şekilde uygulanabilir.

Bu yapılandırma yaklaşımı, hem okunabilirliği hem de bakım kolaylığını artırmakta; yazılım tarafında yapılan değişikliklerin donanım üzerindeki etkisini rahat anlaşılabilir ve kontrol edilebilir hâle getirmektedir. Ayrıca, yapılandırma ve okuma işlemlerinin modüler şekilde ayrılması sayesinde her bir struct gerektiğinde bağımsız olarak güncellenebilmekte veya izlenebilmektedir. Bu noktada ana yapı, sistemin yönetilebilirliği açısından merkezî bir kontrol noktası niteliği kazanmakta, tasarımın sürdürülebilirliğine katkıda bulunmaktadır.