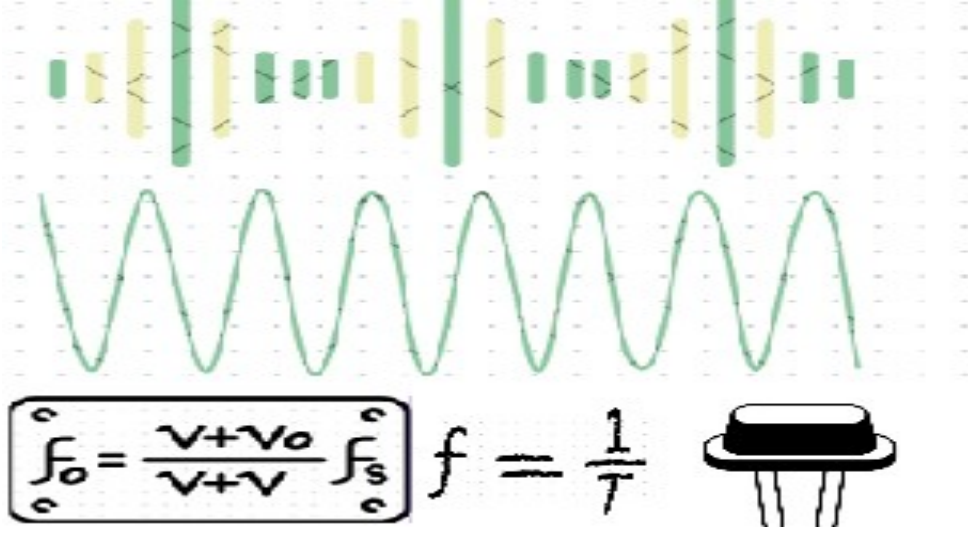




STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

Bu belgenin amacı, STM32F407 mikrodnetleyicisinin saat sistemini derinlemesine analiz ederek, konuya açıklık getirmek ve uygulama sürecini kolaylaştırmaktır. Buradaki temel hedef, sadece sistemi Türkçeleştirmek değil, aynı zamanda saat yapılandırma sürecini okuyucunun daha iyi anlayabileceği, sistematik bir şekilde takip edebileceği hale getirmektir. Konular adım adım ele alınarak, mikrodnetleyici saat sistemi üzerinde tam hakimiyet sağlanması hedeflenmiştir.

Bu Dökümandan Öğreneceklerimiz:

- Osilatörlerin çalışma prensibi
- Hata oranları ve toleranslar
- Saat yapılandırmasını kontrol eden register'lar

Uygulama Çalışması:

Bu dökümanda öğrendiklerimizi pratiğe dökmek için aşağıdaki uygulamayı gerçekleştireceğiz:

16 MHz ve 168 MHz MCO Çıkışlarını Osiloskop ile Gözleme

Hazırlayan:
ERTAN SULUAGAÇ
Sr. Embedded Software Developer

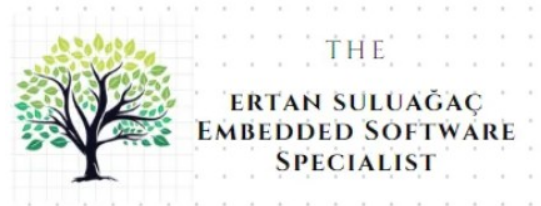
Giriş

Mikrodenetleyicilerde saat sistemi, tüm işlemcinin ve çevresel birimlerin zamanlamasını kontrol eden kritik bir bileşendir. STM32F407 serisi, farklı uygulama gereksinimlerine uygun olarak yapılandırılabilen esnek bir saat yönetim sistemine sahiptir. Bu belge kapsamında, **dahili (HSI), harici (HSE) ve PLL (Phase-Locked Loop)** saat kaynaklarının çalışma prensiplerini, doğruluk seviyelerini ve uygulamalarını detaylı bir şekilde ele alacağız.

HSI (High-Speed Internal) osilatörü, mikrodenetleyicinin fabrika kalibrasyonlu ve düşük güç tüketimine sahip dahili saat kaynağıdır. Bu osilatör, dış bileşen ihtiyacını ortadan kaldırarak hızlı başlatma süreleri sunar. Ancak, frekans doğruluğu harici kristal osilatörlere (HSE) kıyasla daha düşüktür. **PLL mekanizması**, daha yüksek frekanslarda çalışma ihtiyacı olduğunda HSI veya HSE gibi kaynaklardan beslenerek sistem saatini artırmak için kullanılır. Bu sayede, çevresel birimlerin yüksek performanslı çalışması sağlanır.

Bu belge kapsamında, **RM0090** referans kılavuzunun **7.3 RCC Registers** bölümünden itibaren, **7.3.1 RCC clock control register (RCC_CR)**'dan **7.3.24 RCC register map'e** kadar tüm bölümler detaylı bir şekilde ele alınmıştır. Her bir register'ın görevleri, yapılandırma yöntemleri ve kullanım senaryoları açıklanarak, sistemin saat yönetimi tam anlamıyla anlaşılacaktır.

Belgenin ilerleyen bölümlerinde, STM32F407'nin saat kaynağı seçiminde en uygun stratejileri belirleyerek, HSI ve HSE osilatörleri nasıl yapılandıracağımızı, PLL ile nasıl frekans yükseltebileceğimizi ve çevresel birimlerin saat gereksinimlerini karşılayacak optimum çözümleri uygulamalı olarak inceleyeceğiz.



STM32F407 Dahili Saat (Internal Clock) Yapısı ve Ayarları

STM32F407 mikrodnetleyicisinde dahili saat sistemi, mikrodnetleyicinin çalışmasını ve çevre birimlerinin senkronizasyonunu sağlamak için kritik bir bileşendir. Saat sistemi, mikrodnetleyicinin **zamanlamasını, güç tüketimini ve performansını** doğrudan etkiler.

Mikrodnetleyicinin saat yönetimi, **RCC (Reset and Clock Control)** birimi tarafından gerçekleştirilir. **RCC** modülü sayesinde, farklı saat kaynakları arasında geçiş yapabilir, saat frekanslarını ayarlayabilir ve çevre birimlerine uygun saat dağılımı sağlayabilirsiniz.

Şimdi konuya daha derin bir bakış ile ele alalım arkadaşlar.

Mikrodnetleyiciler dünyasında, saat sistemleri adeta bir organizmanın **kalbi** gibidir. Eğer saat düzgün çalışmazsa, sistemin geri kalanı da senkronize olamaz. Peki, **STM32F407**'nin dahili saat sistemi ne kadar güvenilir? Nasıl yapılandırılır? İşte tüm bu soruların cevaplarını aramak için teknik detaylara dalıyoruz.

İlk Durak: Datasheet İncelemesi

Başlamadan önce, güvenilir kaynağımız olan **STM32F407 datasheet**'ine göz atmalıyız. Teknik dokümantasyonda, dahili saat sisteminin nasıl çalıştığını ve hangi register'ların sorumlu olduğunu bulacağız.

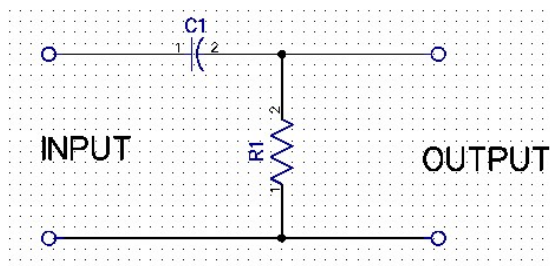
Hedefimiz şu sorulara yanıt bulmak:

- Dahili osilatörler nasıl çalışır?
- Hata oranları ve toleransları nelerdir?
- Saat yapılandırmasını hangi register'lar kontrol ediyor?

Dahili Osilatör (HSI & LSI) Nasıl Çalışır?

STM32F407 mikrodnetleyicisinin içinde yer alan **dahili osilatörler**, mikrodnetleyiciye harici bir bileşene ihtiyaç duymadan zaman referansı sağlayan **RC (Resistor-Capacitor)** tabanlı osilatörlerdir.

Hadi gelin arkadaşlar, şimdi RC osilatör devresinin iç yapısını birlikte inceleyelim!



RC oscillator Circuit

Şöyle bir soru sorduğunuzu duyar gibiyim: "Bu tek girişli RC devresinde faz farkı var mı hocam?"

Harika bir soru arkadaşlar! Şimdi bu soruya birlikte yanıt arayalım. Karşımızda gördüğünüz bu basit **RC devresi**, bir giriş (**INPUT**) ve bir çıkış (**OUTPUT**) hattından oluşuyor. Devreyi ilk bakışta incelediğinizde "**Burada gerçekten faz farkı oluşuyor mu?**" diye düşünebilirsiniz.

Cevap **evet**, bu devrede faz farkı oluşuyor. Peki nasıl? Şöyle açıklayalım:

- Giriş sinyali, ilk olarak **C1 kapasitörüne** uygulanır. Kapasitörün temel özelliği, alternatif akım (**AC**) sinyallerini **bir miktar geciktirerek** iletmesidir.
- Kapasitörden geçen akım, daha sonra **R1 direnci** üzerinden toprağa akar. Direnç, gerilimi sınırlandırırken, kapasitörle birlikte bir **faz farkı** oluşturur.

Bu durumda, **çıkış sinyali giriş sinyaline göre gecikmiş** olur ve aralarında bir **faz farkı** meydana gelir. Faz farkının miktarı ise devrede kullanılan **R ve C** elemanlarının değerlerine ve giriş sinyalinin frekansına bağlıdır.

Peki bu faz farkı ne kadar olabilir?

Matematiksel olarak faz farkı şu formülle hesaplanır

$$\theta = \tan^{-1}\left(\frac{1}{2\pi RC}\right)$$

Bu formülden şu sonuçları çıkarabiliriz:

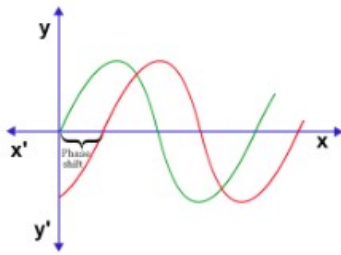
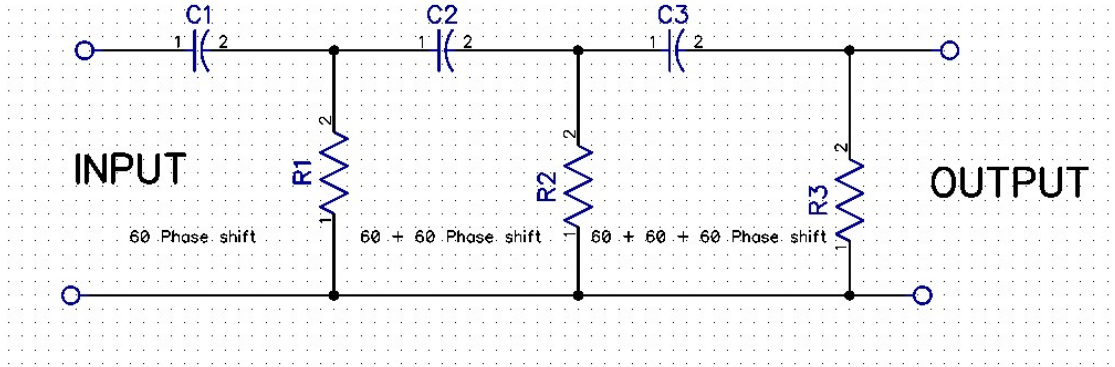
- **Düşük frekanslarda**, faz farkı **90°'ye yaklaşır**, çünkü kapasitörün etkisi daha fazla hissedilir.
- **Yüksek frekanslarda**, faz farkı **azalır** ve çıkış sinyali girişe daha yakın hale gelir.

Sonuç:

Bu tek girişli RC devresi **maksimum 90° faz farkı** sağlayabilir. Ancak gerçek uygulamalarda **45° civarında faz kayması** sıklıkla görülür.

Şimdi bir düşünelim: **Bu devre tek başına yeterli mi?** Eğer amacımız osilasyon oluşturmaksa tek aşamalı RC devresi yeterli olmaz. Bu yüzden birden fazla RC aşaması kullanarak daha büyük faz kaymaları elde edebiliriz.

Hadi bir sonraki adımda **faz kaydırmayı üç aşamada nasıl gerçekleştireceğimizi** birlikte inceleyelim!



Şimdi arkadaşlar, üç aşamalı RC faz kaydırma devresini inceleyelim!

Önceki tek aşamalı RC devresinde maksimum **90°'ye kadar faz kayması** sağlanabileceğini görmüştük. Ancak bu devreyi birden fazla aşama ile genişleterek, toplamda daha fazla faz kayması elde edebiliriz. Karşımızda gördüğünüz devre, **üç aşamalı bir RC faz kaydırıcı** devresidir ve her aşamada sinyalin fazını adım adım değiştirmektedir.

Devrenin Çalışma Prensibi

Bu devre, giriş sinyalinin **R1, R2, R3** dirençleri ve **C1, C2, C3** kapasitörleri üzerinden geçirilmesiyle, her aşamada bir **faz kayması** oluşturur. Devreyi adım adım inceleyelim:

1. **İlk Aşama (C1 - R1):**
 - Giriş sinyali, ilk RC ağına uygulanır ve burada **60°'lik bir faz kayması** oluşur.
 - Çıkış voltajı, giriş sinyaline göre biraz gecikmeli olur.
2. **İkinci Aşama (C2 - R2):**
 - İlk aşamada zaten 60° faz kayması gerçekleşmiştir.
 - İkinci RC filtresi ile birlikte toplam faz kayması **60° + 60° = 120°** olur.
3. **Üçüncü Aşama (C3 - R3):**
 - Son aşamada bir **60° daha faz kayması** eklenerek toplamda **180°'lik faz farkı** elde edilir.

Toplam Faz Kayması Hesabı

Her bir RC aşaması teorik olarak yaklaşık **60° faz kayması** sağlar. Üç aşamanın birleştirilmesiyle toplam faz kayması şu şekilde hesaplanır:

$$60^\circ + 60^\circ + 60^\circ = 180^\circ$$

Bu 180° faz kayması, geri besleme sağlandığında **osilasyon** oluşturmak için yeterlidir. Eğer geri besleme devreye eklenirse, bu devre kendi kendine salınımlar yapmaya başlayacaktır.

Devrenin Frekans Hesabı

Bu devrede osilasyon frekansı şu formülle belirlenebilir:

$$f = \frac{1}{2\pi RC\sqrt{6}}$$

Burada:

- R = Direnç değeri (Ohm cinsinden),
- C = Kapasitör değeri (Farad cinsinden).

Bu formül, osilatörün üreteceği temel frekansı hesaplamamıza yardımcı olur.

Hata oranları ve toleransları nelerdir?

- Clock, reset, and supply management
 - 1.8 V to 3.6 V application supply and I/Os
 - POR, PDR, PVD and BOR
 - 4-to-26 MHz crystal oscillator
 - Internal 16 MHz factory-trimmed RC (1% accuracy)
 - 32 kHz oscillator for RTC with calibration
 - Internal 32 kHz RC with calibration

Dahili 16 MHz Fabrika Ayarlı RC Osilatörü (%1 Doğruluk)

STM32F405/407 serisinin **Clock, Reset and Supply Management** bölümünde şu ifade yer alıyor:

Internal 16 MHz factory-trimmed RC (1% accuracy)

Bu ifade, mikrodenetleyicinin içinde bulunan **HSI (High-Speed Internal)** saat kaynağının **16 MHz frekansında** çalıştığını ve fabrika çıkışlı olarak **%1** doğrulukla kalibre edildiğini belirtmektedir.

Bu Ne Anlama Geliyor?

- **16 MHz frekans** – Mikrodenetleyici, PLL (Phase-Locked Loop) kullanmadan doğrudan dahili HSI saat kaynağını kullanarak 16 MHz frekansta çalışır.
- **%1 doğruluk payı** – Bu, saat frekansının $\pm 0.01\%$ sapmaya sahip olabileceği anlamına gelir.
 - 16 MHz'in %1'i = **160 kHz sapma**

- Yani saat frekansı **15.84 MHz** ile **16.16 MHz** arasında değişebilir.

Sonuç:

Bu sapma, sıcaklık değişimleri, voltaj dalgalanmaları ve üretim toleransları nedeniyle ortaya çıkabilir.

Hata Oranı Ne Kadar Önemli?

Hata oranı %1 seviyesinde olması şu durumları etkileyebilir:

Kullanım Durumu	Etkisi
Genel zamanlayıcılar (TIM)	Küçük sapmalar sorun yaratmaz
UART Haberleşmesi	Yüksek baud rate hızlarında senkronizasyon sorunları olabilir
PWM Sinyalleri	Hassasiyet gerektiren uygulamalarda hatalar olabilir
ADC Zamanlaması	Örnekleme sürelerinde küçük gecikmeler oluşabilir

Özetle, eğer mikrodenetleyici **zamanlama kritik** uygulamalarda kullanılacaksa, **harici bir kristal osilatör (HSE)** tercih edilmelidir. Ancak:

- Genel amaçlı gömülü sistemlerde,
- Zamanlama gereksinimlerinin toleranslı olduğu projelerde,
- Düşük maliyet ve bileşen sayısının önemli olduğu yerlerde

HSI yeterli olabilir.

Bu Hata Oranı Normal mi?

Evet, **%1 hata oranı** dahili RC osilatörler için **normal kabul edilir**.

Genellikle dahili RC osilatörler harici kristallere kıyasla daha düşük doğruluğa sahiptir çünkü:

1. **Fabrika kalibrasyonu sabitlenmiştir**, çalışma ortamına göre değişkenlik gösterir.
2. **Sıcaklık ve voltaj değişimlerinden** etkilenir.
3. **Yüksek hassasiyet gerektiren sistemlerde** harici kristaller daha iyi performans sağlar (± 20 ppm veya daha az hata).

Hadi arkadaşlar, şimdi STM32F405/407 serisinin dahili osilatör karakteristiklerini inceleyelim!

Mikrodenetleyicilerde saat kaynağı, sistemin kararlılığı ve performansı açısından oldukça kritik bir faktördür. **STM32F405/407** serisi mikrodenetleyicilerde **High-Speed Internal (HSI)** ve **Low-Speed Internal (LSI)** olmak üzere iki dahili osilatör bulunur. Hadi şimdi, bu osilatörlerin teknik detaylarını ve nasıl çalıştıklarını datasheet üzerinden inceleyelim.

Eğer siz de **STMicroelectronics'in DM00037051** numaralı dokümanını takip etmek isterseniz, **104. sayfadan itibaren** bu bilgileri detaylı olarak bulabilirsiniz. Şimdi teknik incelememize başlayalım!

6.3.9 Internal clock source characteristics

The parameters given in [Table 34](#) and [Table 35](#) are derived from tests performed under ambient temperature and V_{DD} supply voltage conditions summarized in [Table 14](#).

High-speed internal (HSI) RC oscillator

Table 34. HSI oscillator characteristics ⁽¹⁾

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{HSI}	Frequency	-	-	16	-	MHz
ACC_{HSI}	HSI user trimming step ⁽²⁾	-	-	-	1	%
	Accuracy of the HSI oscillator	$T_A = -40 \text{ to } 105 \text{ }^\circ\text{C}^{(3)}$	-8	-	4.5	%
		$T_A = -10 \text{ to } 85 \text{ }^\circ\text{C}^{(3)}$	-4	-	4	%
		$T_A = 25 \text{ }^\circ\text{C}^{(4)}$	-1	-	1	%
$t_{su(HSI)}^{(2)}$	HSI oscillator startup time	-	-	2.2	4	μs
$I_{DD(HSI)}^{(2)}$	HSI oscillator power consumption	-	-	60	80	μA

1. $V_{DD} = 3.3 \text{ V}$, PLL OFF, $T_A = -40 \text{ to } 125 \text{ }^\circ\text{C}$ unless otherwise specified.

2. Specified by design.

3. Evaluated by characterization - not tested in production.

4. Factory calibrated, parts not soldered.

HSI (High-Speed Internal) RC Osilatörü

HSI osilatörü, mikrodenetleyicinin **16 MHz frekansında** çalışan dahili bir osilatördür ve şu temel özelliklere sahiptir:

HSI Özellikleri (Tablo 34 İncelemesi):

Parametre	Min	Typ	Max	Birim	Açıklama
Frekans (f_{HSI})	-	16	-	MHZ	16 MHz sabit frekans.
Kullanıcı ayarlı trimming	-	-	1	%	Kullanıcı tarafından kalibre edilebilir.
Hata payı (Doğruluk) (-40°C to 105°C)	-8	-	4.5	%	Çalışma sıcaklığına bağlı doğruluk değişimi.
Hata payı (Doğruluk) (-10°C to 85°C)	-4	-	4	%	Daha dar sıcaklık aralığında daha iyi doğruluk
Hata payı (Doğruluk) (25°C)	-1	-	1	%	Oda sıcaklığında en iyi doğruluk.
Çalıştırma süresi ($t_{STARTUP}$)	-	2.2	4	μs	Osilatörün stabil hale gelme süresi.
Güç tüketimi (I_{DD})	-	60	80	μA	Düşük güç tüketimi.

Analiz:

- HSI osilatörünün fabrika kalibrasyonu $\pm 1\%$ doğruluk sağlarken, sıcaklık değişimlerinde $\pm 4\%$ hata oranına kadar çıkabiliyor.
- Çalıştırma süresi **2.2-4 μs** aralığında olup, hızlı başlatma gereksinimleri için uygundur.
- Düşük güç tüketimi sayesinde, enerji açısından verimli uygulamalarda kullanılabilir.

Kullanım Senaryoları:

- Temel işlem yüklerinde (UART, SPI gibi zamanlama kritik olmayan işlemler) uygundur.
- Boot aşamalarında hızlı başlangıç sağlar.
- Yüksek hassasiyet gerektiren uygulamalarda harici kristale (HSE) geçiş önerilir.

LSI (Low-Speed Internal) RC Osilatörü

LSI osilatörü, daha düşük güç tüketimi gerektiren uygulamalar için tasarlanmış bir dahili saat kaynağıdır. **32 kHz** civarında bir frekans sağlar ve genellikle **RTC (Gerçek Zaman Saati)** ve **Watchdog Timer** gibi zamanlayıcı fonksiyonlarında kullanılır.

Table 35. LSI oscillator characteristics ⁽¹⁾

Symbol	Parameter	Min	Typ	Max	Unit
$f_{LSI}^{(2)}$	Frequency	17	32	47	kHz
$t_{su(LSI)}^{(3)}$	LSI oscillator startup time	-	15	40	μs
$I_{DD(LSI)}^{(3)}$	LSI oscillator power consumption	-	0.4	0.6	μA

1. $V_{DD} = 3 V$, $T_A = -40$ to $105^\circ C$ unless otherwise specified.

2. Evaluated by characterization - not tested in production.

3. Specified by design.

LSI Özellikleri (Tablo 35 İncelemesi):

Parametre	Min	Typ	Max	Birim	Açıklama
Frekans (f_{LSI})	17	32	47	Khz	Nominal 32 kHz frekans
Çalıştırma süresi ($t_{STARTUP}$)	-	15	40	μs	Düşük hızlı başlatma süresi
Güç tüketimi (I_{DD})	-	0.4	0.6	$\mu A \mu A$	Ultra düşük güç tüketimi

Analiz:

- Düşük güç tüketimiyle özellikle **gömülü sistemler** ve **pil ile çalışan uygulamalarda** idealdir.
- Frekansın $\pm 40\%$ hata oranına sahip olması hassas zamanlama gerektiren uygulamalarda dikkat edilmesi gereken bir faktördür.

- Başlangıç süresi HSI'ye kıyasla daha uzundur ancak düşük güç gereksinimleri nedeniyle ideal kullanım alanına sahiptir.

Kullanım Senaryoları:

- RTC uygulamaları (zaman takibi).
- Watchdog timer için uygun bir kaynak.
- Ultra düşük güç tüketimi gerektiren IoT cihazları ve sensör tabanlı uygulamalar.

HSI ve LSI Arasındaki Karşılaştırma

Özellik	HSI (16 MHz)	LSI (32 kHz)
Frekans	16 MHz	32 kHz
Doğruluk	$\pm 1\%$ (kalibreli)	$\pm 40\%$
Çalışma Voltajı	3.3V	3V
Güç Tüketimi	60-80 μ A	0.4-0.6 μ A
Kullanım Alanı	Genel saat ihtiyacı	RTC, WDT, düşük güç

PLL (Phase Locked Loop) Özellikleri

6.3.10 PLL characteristics

The parameters given in [Table 36](#) and [Table 37](#) are derived from tests performed under temperature and V_{DD} supply voltage conditions summarized in [Table 14](#).

Table 36. Main PLL characteristics

Symbol	Parameter	Conditions	Min	Typ	Max	Unit
f_{PLL_IN}	PLL input clock ⁽¹⁾	-	0.95 ⁽²⁾	1	2.10	MHz
f_{PLL_OUT}	PLL multiplier output clock	-	24	-	168	MHz
f_{PLL48_OUT}	48 MHz PLL multiplier output clock	-	-	48	75	MHz
f_{VCO_OUT}	PLL VCO output	-	100	-	432	MHz
t_{LOCK}	PLL lock time	VCO freq = 100 MHz	75	-	200	μ s
		VCO freq = 432 MHz	100	-	300	

PLL devresi, mikrodenetleyici saat frekansını artırmak için kullanılan bir bileşendir. HSI veya HSE kaynaklarından gelen sinyali çarparak daha yüksek frekansta çalışmayı mümkün kılar.

Table 36. Main PLL characteristics (continued)

Symbol	Parameter	Conditions		Min	Typ	Max	Unit
Jitter ⁽³⁾	Cycle-to-cycle jitter	System clock 120 MHz	RMS	-	25	-	ps
			peak to peak	-	±150	-	
	Period Jitter		RMS	-	15	-	
			peak to peak	-	±200	-	
	Main clock output (MCO) for RMII Ethernet	Cycle to cycle at 50 MHz on 1000 samples	-	32	-		
	Main clock output (MCO) for MII Ethernet	Cycle to cycle at 25 MHz on 1000 samples	-	40	-		
	Bit Time CAN jitter	Cycle to cycle at 1 MHz on 1000 samples	-	330	-		
I _{DD(PLL)} ⁽⁴⁾	PLL power consumption on VDD	VCO freq = 100 MHz VCO freq = 432 MHz	0.15 0.45	-	0.40 0.75	mA	
I _{DDA(PLL)} ⁽⁴⁾	PLL power consumption on VDDA	VCO freq = 100 MHz VCO freq = 432 MHz	0.30 0.55	-	0.40 0.85	mA	

1. Take care of using the appropriate division factor M to obtain the specified PLL input clock values. The M factor is shared between PLL and PLLI2S.

2. Specified by design.

3. The use of 2 PLLs in parallel could degraded the Jitter up to +30%.

4. Evaluated by characterization - not tested in production.

PLL Karakteristikleri (Tablo 36 İncelemesi):

Bu tablo, STM32F405/407 serisi mikrodnetleyicilerin PLL (Phase-Locked Loop) özelliklerini detaylandırmaktadır. PLL, giriş frekansını yükselterek mikrodnetleyicinin farklı çevre birimlerine yüksek frekanslı saat sinyalleri sağlamak için kullanılır. Tablo, PLL'in jitter (dalgalanma), güç tüketimi ve çeşitli saat çıkış özelliklerini içermektedir.

Bu tablo, STM32F405/407 serisi mikrodnetleyicilerin PLL (Phase-Locked Loop) özelliklerini detaylandırmaktadır. PLL, giriş frekansını yükselterek mikrodnetleyicinin farklı çevre birimlerine yüksek frekanslı saat sinyalleri sağlamak için kullanılır. Tablo, PLL'in jitter (dalgalanma), güç tüketimi ve çeşitli saat çıkış özelliklerini içermektedir.

Jitter Karakteristikleri

Jitter, saat sinyalinin zamanlama açısından sapmalarını ifade eder ve sistem performansı açısından kritik bir parametredir.

Cycle-to-cycle jitter:

- **Sistem saati (120 MHz):**
 - RMS (Root Mean Square) değeri: **25 ps**,
 - Peak-to-peak (tepeden tepeye) değeri: **±150 ps**.
 - Bu, iki ardışık saat darbesi arasındaki varyasyonu gösterir.

Period jitter:

- RMS değeri: **15 ps**,
- Peak-to-peak değeri: **±200 ps**.
- Saat sinyalinin periyot süresindeki dalgalanmayı ifade eder.

Yorum: Bu değerler, PLL'in oldukça stabil bir saat sinyali sağladığını göstermektedir. Ancak, jitter hassas zamanlama gerektiren uygulamalarda (örneğin, haberleşme protokolleri gibi) dikkat edilmesi gereken bir parametredir. İki PLL biriminin paralel kullanımı jitter değerini %30'a kadar artırabilir (dipnot 3).

Ana Saat Çıkışlarının Performansı (MCO - Master Clock Output)

Bu bölümde, PLL'in farklı çevre birimleri için ürettiği saat sinyallerinin zamanlama performansı incelenmiştir.

- **RMII Ethernet çıkışı:**
 - 50 MHz saat frekansında, cycle-to-cycle jitter: **32 ps**.
- **MII Ethernet çıkışı:**
 - 25 MHz saat frekansında, cycle-to-cycle jitter: **40 ps**.
- **CAN bit zamanı jitter:**
 - 1 MHz saat frekansında, jitter: **330 ps**.

Yorum: Bu değerler, farklı çevre birimlerinde kullanılan saat sinyallerinin kararlılığı açısından oldukça iyidir. Ancak CAN protokolü gibi hassas zamanlama gerektiren uygulamalarda, jitter'in daha yüksek olduğu görülmektedir (330 ps).

PLL Güç Tüketimi

PLL'in iki farklı voltaj kaynağında tükettiği akım değerleri verilmiştir:

VDD (Ana Besleme Kaynağı):

- **VCO frekansı = 100 MHz:**
 - Minimum: **0.15 mA**,
 - Maksimum: **0.40 mA**.
- **VCO frekansı = 432 MHz:**
 - Minimum: **0.45 mA**,
 - Maksimum: **0.75 mA**.

VDDA (Analog Besleme Kaynağı):

- **VCO frekansı = 100 MHz:**
 - Minimum: **0.30 mA**,
 - Maksimum: **0.40 mA**.
- **VCO frekansı = 432 MHz:**
 - Minimum: **0.55 mA**,
 - Maksimum: **0.85 mA**.

Yorum: PLL'in yüksek frekansta çalıştırılması, güç tüketimini artırmaktadır. Daha yüksek performanslı işlemler gerektiğinde, güç tüketimini optimize etmek için frekans seçimi dikkatlice yapılmalıdır.

Genel Değerlendirme ve Öneriler

- **Zamanlama kritik uygulamalar:**
PLL'in düşük jitter seviyeleri, zamanlama kritik uygulamalar (USB, Ethernet, SPI) için uygundur, ancak paralel PLL kullanımı jitter'i artırabileceğinden dikkat edilmelidir.
- **Güç tüketimi açısından:**
Yüksek frekanslı çalışmalarda güç tüketimi göz önünde bulundurulmalı ve gereksiz güç tüketimini önlemek için dinamik frekans ölçeklendirme kullanılmalıdır.
- **Çevre birimi seçimleri:**
Ethernet ve CAN gibi çevre birimleri için sağlanan saat çıkışlarının jitter seviyeleri kabul edilebilir düzeydedir. Ancak, daha hassas zamanlama gerektiren uygulamalarda ek harici saat kaynakları düşünülmelidir.

Bu analiz doğrultusunda, PLL'in sistem performansına olan etkilerini göz önünde bulundurarak, uygulamanız için en uygun yapılandırmayı seçebilirsiniz. Daha fazla detay için datasheet'in ilgili bölümlerini inceleyebilirsiniz.

Bu teknik incelememizin ardından, şimdi isterseniz **HSI veya LSI'nin yazılım tarafında nasıl konfigüre edileceğini** konuşmaya geçelim!

Şimdi STM32F407'nin saat yapılandırmasını incelemeye başlıyoruz!

Saat sistemi, bir mikrodenetleyicinin **temel taşıdır** ve tüm çevresel birimlerin zamanlamasını sağlar. **STM32F407** mikrodenetleyicisi, sistemin farklı modlarını çalıştırabilmesi için oldukça esnek ve güçlü bir saat yapılandırma mekanizmasına sahiptir. Saat yönetimi, düşük güç tüketimi ve yüksek performans gerektiren uygulamalar için kritik bir rol oynar.

Bu bölümde, saat yapılandırmasını kontrol eden **önemli register'ları** inceleyeceğiz ve ardından kodlama kısmına geçerek bu register'ların nasıl kullanılacağını uygulamalı olarak göstereceğiz. Eğer siz de süreci datasheet üzerinden takip etmek isterseniz, bundan sonra saat yapılandırmasını incelerken, **RM0090 - STM32F405/407/415/417 Reference Manual** dokümanındaki **Reset and Clock Control (RCC)** bölümünden itibaren ilerleyeceğiz.

Hazırsan, şimdi bu referans kılavuzdan ilgili register'ları adım adım incelemeye başlayalım!

Saat Yapılandırmasını Kontrol Eden Register'lar

STM32F407 mikrodenetleyicisinde saat sisteminin kontrolü, RCC (Reset and Clock Control) birimi tarafından sağlanır. Bu birimde yer alan bazı kritik register'ları aşağıda inceleyelim:

1. RCC_CR (Clock Control Register)

- Ana saat kaynaklarını etkinleştirmek veya devre dışı bırakmak için kullanılır.
- Önemli bitler:
 - **HSION (Bit 0):** Dahili yüksek hızlı osilatörü (HSI) etkinleştirir.
 - **HSIRDY (Bit 1):** HSI'nin stabil olup olmadığını kontrol eder.
 - **HSEON (Bit 16):** Harici osilatör (HSE) etkinleştirilmesi.
 - **HSERDY (Bit 17):** HSE'nin stabil olup olmadığını gösterir.
 - **PLLON (Bit 24):** PLL devresini aktif eder.

2. RCC_CFGR (Clock Configuration Register)

- Saat kaynağının seçimi ve bölücülerinin ayarlanmasını sağlar.
- Önemli bitler:
 - **SW[1:0] (Bit 0-1):** Sistem saat kaynağını belirler. (00: HSI, 01: HSE, 10: PLL)
 - **HPRE[3:0] (Bit 4-7):** AHB prescaler ayarı.
 - **PPRE1/PPRE2 (Bit 10-13):** APB1 ve APB2 prescaler ayarları.

3. RCC_PLLCFGR (PLL Configuration Register)

- PLL çarpanlarını ve bölücülerini ayarlayarak daha yüksek frekanslara çıkmayı sağlar.
- Önemli bitler:
 - **PLLM (Bit 0-5):** PLL giriş bölücü değeri.
 - **PLLN (Bit 6-14):** PLL çarpanı.
 - **PLLP (Bit 16-17):** PLL çıkış bölücü.

4. RCC_AHB1ENR, RCC_APB1ENR, RCC_APB2ENR (Peripheral Clock Enable Register'lar)

- Çevresel birimlere saat sinyali sağlar.
- UART, GPIO, SPI, I2C gibi bileşenler için saat sinyalini açıp kapatabiliriz.

5. **RCC_BDCR (Backup Domain Control Register)**

- RTC ve LSE osilatörü yapılandırmasını kontrol eder.

6. **RCC_CSR (Control/Status Register)**

- Saat kesintileri ve düşük güç modları için kullanılır.

Örnek Kod İncelemesi Bu register'ların nasıl kullanıldığını daha iyi anlamak için bir sonraki adımda saat yapılandırma kodlarını adım adım inceleyeceğiz. İlk olarak, sistem saat kaynağı olarak HSI'yi kullanarak temel bir konfigürasyon yapacağız. Daha sonra, PLL kullanarak 168 MHz'lik bir sistem saatine nasıl geçiş yapacağımızı adım adım göreceğiz.

Başlangıç Noktamız: 7.3.1 RCC Clock Control Register (RCC_CR)

İlk olarak, **RCC_CR (Clock Control Register)** ile başlamalıyız. Çünkü bu register, mikrodenetleyicinin ana saat kaynaklarını açma/kapama, stabilite kontrolleri ve PLL aktivasyonları gibi temel saat yönetim işlevlerini kontrol eder.

İçeriğinde neler var?

- HSI (High-Speed Internal) osilatörün kontrolü.
- HSE (High-Speed External) osilatörün kontrolü.
- PLL'in aktif edilmesi ve durumu.
- Saat kaynaklarının hazır olup olmadığını belirleyen bayraklar.

İzleyeceğimiz adımlar:

1. RCC_CR register'ını incele → Hangi bitlerin ne işe yaradığını öğren.
2. Saat kaynağını seçme işlemi → HSI/HSE/PLL yapılandırmasını anla.
3. Güç yönetimi ile ilişkisini öğren → Gereksiz güç tüketimini azaltma yolları.

Sonraki Adımlar: RCC_CR register'ını detaylıca inceledikten sonra şu bölümlere geçebiliriz:

1. RCC_PLLCFGR (PLL Configuration Register) – Saat çarpanlarını ve bölücülerini kontrol eder.
2. RCC_CFGR (Clock Configuration Register) – Saat kaynağını seçme ve bölücülerle sistem saatini ayarlama.
3. RCC_AHBxENR, RCC_APB1ENR, RCC_APB2ENR – Çevresel birimlere saat sinyali sağlama.

Önerilen Yöntem: Adım Adım Yaklaşım Şimdi, ilk adım olarak **"7.3.1 RCC_CR"** bölümünü açalım ve detayları inceleyelim. Her bir bitin görevini anlayarak ilerleyeceğiz ve ardından kodlama kısmına geçerek gerçek donanım üzerinde yapılandırma işlemlerini gerçekleştireceğiz.

Hadi başlayalım!

Clocks(Saatler)

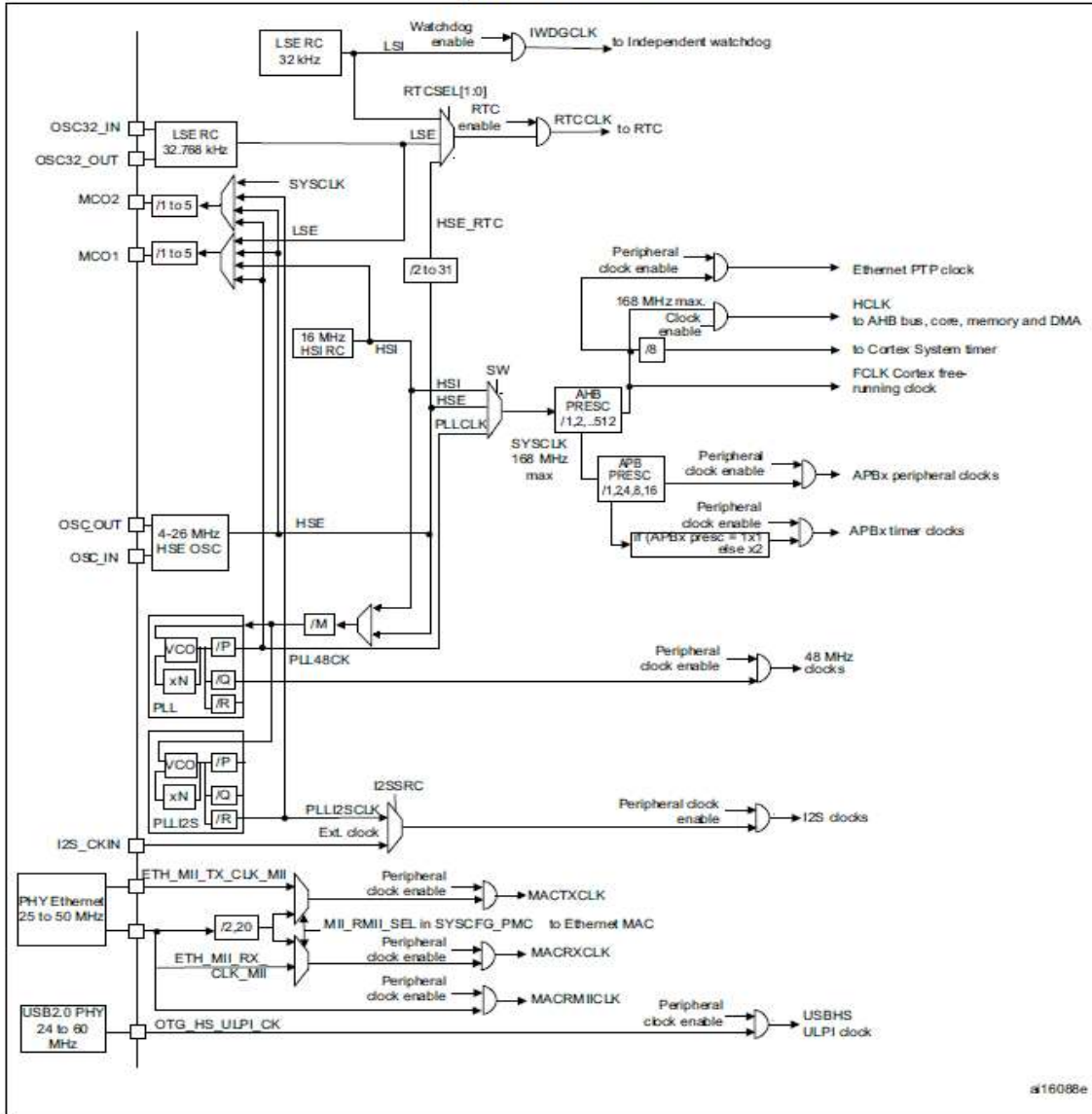
Sistem saatini (**SYSCCLK**) sürmek için üç farklı saat kaynağı kullanılabilir:

- **HSI** osilatör saati
- **HSE** osilatör saati
- Ana **PLL** saati

Cihazlar aşağıdaki iki ikincil saat kaynağına sahiptir:

- **32 kHz** düşük hızlı dâhili **RC (LSI RC)**, bağımsız bekçi köpeğini ve isteğe bağlı olarak, **Durdurma/Bekleme** modundan Otomatik uyandırma için kullanılan **RTC**.
- **32,768 kHz** düşük hızlı harici kristal (**LSE kristal**), isteğe bağlı olarak **RTC**'yi çalıştırır saat (**RTCCLK**) Güç tüketimini optimize etmek için her saat kaynağı kullanılmadığında bağımsız olarak açılabilir veya kapatılabilir.

Figure 21. Clock tree



1. For full details about the internal and external clock source characteristics, refer to the Electrical characteristics section in the device datasheet.

Saat kontrolörü, çekirdeği ve çevre birimlerini en yüksek frekansta çalıştırmak için harici kristal veya osilatör seçiminde uygulamaya yüksek derecede esneklik sağlar ve **Ethernet**, **USBOTGFS** ve **HS**, **I2S** ve **SDIO** gibi belirli bir saate ihtiyaç duyan çevre birimleri için uygun frekansı garanti eder. Çeşitli ön bölücüler, **AHB** frekansını, yüksek hızlı **APB** (**APB2**) ve düşük hızlı **APB** (**APB1**) bölgelerini yapılandırmak için kullanılır.

- **AHB** bölgesinin maksimum frekansı **168 MHz**'dir.
- Yüksek hızlı **APB2** bölgesinin izin verilen maksimum frekansı **84 MHz**'dir.
- Düşük hızlı **APB1** bölgesinin izin verilen maksimum frekansı **42 MHz**'dir

Tüm periferik saatler, sistem saatinden (**SYSCLK**) türetilir, ancak:

- **USB OTG FS** saati (**48 MHz**)
- Rastgele analog üretici (**RNG**) saati (≤ 48 MHz)
- **SDIO** saati (≤ 48 MHz)
- **PLL**'nin belirli bir çıkışından (**PLL48CLK**) gelmektedir.

- I2S saati

Yüksek kaliteli ses performansı elde etmek için, I2S saati ya belirli bir **PLL**'den (**PLLI2S**) ya da **I2S_CKIN** pimine eşlenmiş harici bir saatten türetilir. **I2S** saat frekansı ve hassasiyeti hakkında daha fazla bilgi için, **Bölüm 28.4.4**'e bakınız: Saat üretici.

- Harici **PHY**'den sağlanan **USB OTG HS** (**60 MHz**) saati
- Harici **PHY**'den sağlanan **Ethernet MAC** saatleri (**TX**, **RX** ve **RMII**). Ethernet yapılandırması hakkında daha fazla bilgi için, Ethernet periferi tanımındaki **Bölüm 33.4.4**'e bakınız. Ethernet kullanıldığında, **AHB** saat frekansı en az **25 MHz** olmalıdır.

RCC, **Cortex** Sistem Zamanlayıcı (**SysTick**)'nin harici saatini, **AHB** saati (**HCLK**) tarafından 8'e bölünmüş şekilde besler. **SysTick**, bu saatle veya **Cortex** saati (**HCLK**) ile çalışabilir; **SysTick** kontrol ve durum kaydında yapılandırılabilir.

Zamanlayıcı saat frekansları otomatik olarak donanım tarafından ayarlanır. İki durum vardır:

1. Eğer **APB** ön bölücüsü 1 ise, zamanlayıcı saat frekansları, zamanlayıcıların bağlı olduğu **APB** bölgesinin frekansıyla aynı frekansa ayarlanır.
2. Aksi takdirde, zamanlayıcıların bağlı olduğu **APB** bölgesinin frekansının iki katı ($\times 2$) olarak ayarlanır.

FCLK, serbest çalışan bir **Cortex®-M4 FPU** saati olarak hareket eder. Daha fazla ayrıntı için, **Cortex®-M4 FPU** teknik referans kılavuzuna bakınız.

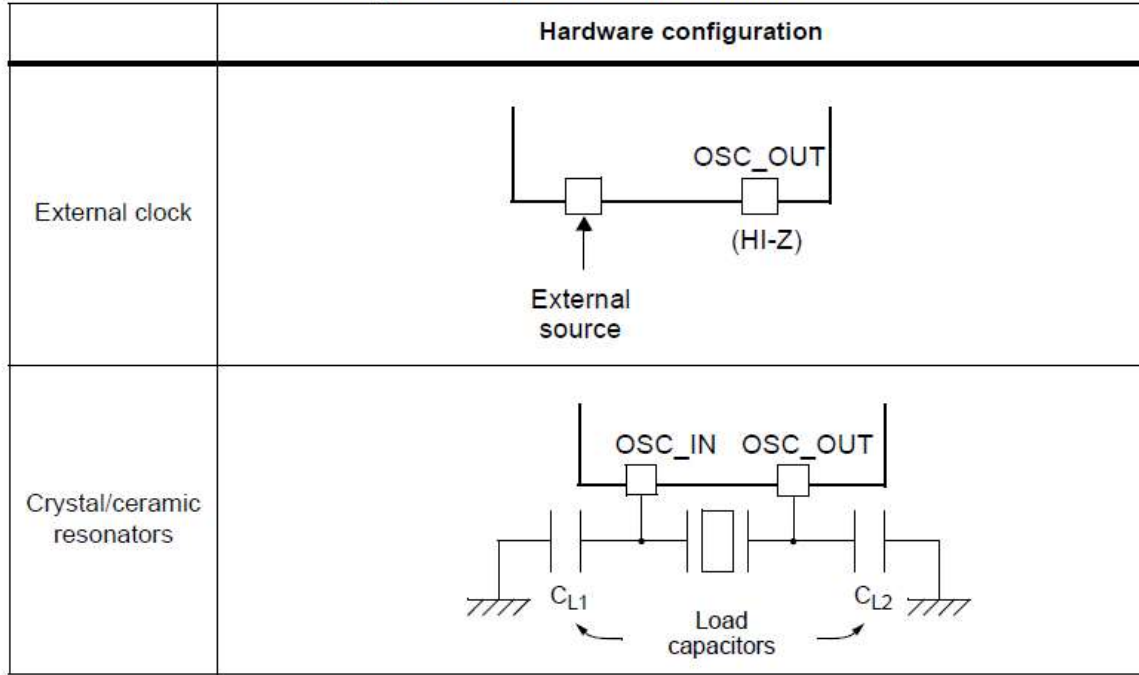
HSE Clock (Yüksek Hızlı Harici Saat)

Yüksek hızlı harici saat sinyali (**HSE**), iki olası saat kaynağından oluşturulabilir:

- **HSE** harici kristal/seramik rezonatör
- **HSE** harici kullanıcı saati

Rezonatör ve yük kapasitörleri, çıkış bozulmasını ve başlangıç dengeleme süresini en aza indirmek için mümkün olduğunca yakın bir şekilde osilatör pinlerine yerleştirilmelidir. Yük kapasitans değerleri, seçilen osilatöre göre ayarlanmalıdır.

Figure 22. HSE/ LSE clock sources



Harici kaynak (HSE by-pass)

Bu modda, bir harici saat kaynağı sağlanmalıdır. Bu modu seçmek için **RCC** saat kontrol kayıtlarındaki (**RCC_CR**) **HSEBYP** ve **HSEON** bitlerini ayarlıyorsunuz. Harici saat sinyali (**kare, sinüs veya üçgen**) ~%50 görev döngüsüyle harici saat sinyali, **OSC_IN** pimine girmelidir, **OSC_OUT** pimi **HI-Z** durumunda bırakılmalıdır. **Şekil 22'**ye bakınız.

Harici kristal/seramik rezonatör (HSE kristali)

HSE'nin ana saatte çok doğru bir frekans üretme avantajı vardır. İlgili donanım yapılandırması **Şekil 22'**de gösterilmiştir. Daha fazla ayrıntı için veri sayfasının elektriksel özellikler bölümüne bakınız. Yüksek hızlı harici osilatörün kararlı veya kararsız olup olmadığını gösteren **HSEBYP** bayrağı, **RCC** saat kontrol kayıtlarında (**RCC_CR**) bulunur. Saat, donanım tarafından bu bitin ayarlanmasıyla başlatılınca kadar serbest bırakılmaz. Etkinleştirildiğinde **RCC** saat kesme kaydında (**RCC_CIR**) kesme oluşturulabilir. **HSE** kristali, **RCC** saat kontrol kayıtlarındaki (**RCC_CR**) **HSEON** biti kullanılarak açılıp kapatılabilir.

Kalibrasyon

RC osilatör frekansları, üretim süreci varyasyonlarından dolayı bir çipten diğerine değişebilir, bu nedenle her cihaz, **ST** tarafından 25 °C'de %1 doğruluk için fabrika kalibre edilir. Sıfırlamadan sonra, fabrika kalibrasyon değeri, **RCC** saat kontrol kaydındaki (**RCC_CR**) **HSICAL[7:0]** bitlerine yüklenir. Uygulama gerilim veya sıcaklık değişimlerine tabi ise, bu **RC** osilatör hızını etkileyebilir. **HSI** frekansını uygulamada **HSITRIM[4:0]** bitleri kullanılarak ayarlayabilirsiniz. **RCC** saat kontrol kaydındaki (**RCC_CR**) **HSIRDY** bayrağı, **HSI RC**'nin kararlı veya kararsız olup olmadığını gösterir. Başlangıçta, **HSI RC** çıkış saati, bu bit donanım tarafından ayarlanana kadar serbest bırakılmaz. **HSI RC**, **RCC** saat kontrol kaydındaki (**RCC_CR**) **HSION** biti kullanılarak açılıp kapatılabilir. **HSI** sinyali ayrıca, **HSE** kristal osilatörü başarısız olduğunda yedek bir kaynak (Yardımcı saat) olarak da kullanılabilir. **Sayfa 222'deki Bölüm 7.2.7'ye** bakınız: Saat güvenlik sistemi (CSS).

PLL Konfigürasyonu

STM32F4xx cihazlarında iki **PLL** bulunmaktadır:

- Ana bir **PLL** (**PLL**), **HSE** veya **HSI** osilatör tarafından saat sinyali alır ve iki farklı çıkış saati üretebilir
- İlk çıkış, yüksek hızlı sistem saati (**168MHz**'ye kadar) oluşturmak için kullanılır.
- İkinci çıkış, **USBOTGFS (48 MHz)**, rastgele analog üretici (**≤48 MHz**) ve **SDIO (≤ 48 MHz)** için saat oluşturmak için kullanılır.
- Yüksek kaliteli ses performansı elde etmek için **I2S** arabirimi üzerinde doğru bir saat oluşturmak için kullanılan özel bir **PLL (PLLI2S)**.

Ana-PLL yapılandırma parametreleri bir kez **PLL** etkinleştirildikten sonra değiştirilemez, bu nedenle **PLL**'yi etkinleştirmeden önce **PLL**'nin saat kaynağı olarak **HSI** veya **HSE** osilatörünün seçilmesi ve bölme faktörleri **M**, **N**, **P** ve **Q**'nun yapılandırılması önerilir.

Yani toparlamak gerekirse, Bu ifade, ana **PLL**'nin yapılandırma parametrelerinin, **PLL**'nin etkinleştirilmesinden sonra değiştirilemez olduğunu belirtiyor. Dolayısıyla, bu parametrelerin doğru bir şekilde ayarlanması ve değiştirilmesi gerektiği için, **PLL**'nin etkinleştirilmeden önce bu ayarların yapılması öneriliyor. Bu ayarlar arasında, **PLL**'nin saat kaynağının (**HSI** veya **HSE** osilatörü) seçilmesi ve bölme faktörlerinin (**M**, **N**, **P** ve **Q**) belirlenmesi bulunmaktadır. Bu işlem, **PLL**'nin istenen frekansı üretmesini sağlar ve sistem performansını optimize eder.

PLLI2S, **PLL** ile aynı giriş saatini kullanır (**PLLM[5:0]** ve **PLLSRC** bitleri her iki **PLL** için ortaktır). Ancak, **PLLI2S**'nin özel etkinleştirme/devre dışı bırakma ve bölme faktörleri (**N** ve **R**) yapılandırma bitleri vardır. **PLLI2S** etkinleştirildikten sonra, yapılandırma parametreleri değiştirilemez.

İki **PLL**, sistem Durma ve Bekleme modlarına geçildiğinde veya sistem saati olarak **HSI** veya **HSE** tarafından saatlenen **PLL** kullanıldığında bir **HSE** hatası meydana geldiğinde, donanım

tarafından otomatik olarak devre dışı bırakılır. **RCC_PLLCFGR** ve **RCC** saat yapılandırma kaydı (**RCC_CFGR**), **PLL** ve diğer saat ayarlarını yapılandırmak için kullanılır.

LSE Clock (Düşük Hızlı Harici Saat)

LSE saati, **32.768 kHz** düşük hızlı harici kristal veya seramik rezonatörden oluşturulur. Gerçek zamanlı saat birimi (**RTC**) için düşük güç tüketimine sahip ancak oldukça doğru bir saat kaynağı sağlama avantajına sahiptir ve saat/takvim veya diğer zamanlama işlevleri için kullanılabilir.

LSE osilatörü, **RCC** Yedek alan kontrol kaydındaki (**RCC_BDCR**) **LSEON** biti kullanılarak açılıp kapatılır. **RCC** Yedek alan kontrol kaydındaki (**RCC_BDCR**) **LSERDY** bayrağı, **LSE** kristalinin kararlı olup olmadığını gösterir. Başlangıçta, **LSE** kristal çıkış saat sinyali, bu bit donanım tarafından ayarlanana kadar serbest bırakılmaz. **RCC** saat kesme kaydında (**RCC_CIR**) etkinleştirilirse bir kesme oluşturulabilir.

Harici kaynak (LSE by-pass)

Bu modda, bir harici saat kaynağı sağlanmalıdır. Frekansı en fazla **1 MHz** olmalıdır. Bu modu seçmek için **RCC** Yedek alan kontrol kaydındaki (**RCC_BDCR**) **LSEBYP** ve **LSEON** bitlerini ayarlayarak seçersiniz. ~%50 görev döngüsüne sahip harici saat sinyali (**kare**, **sinüs** veya **üçgen**), **OSC32_IN** pimine girilir, **OSC32_OUT** pimi **HI-Z** durumunda bırakılmalıdır. **Şekil 22'**ye bakınız.

Sistem saati (SYSCLK) seçimi

Bir sistem sıfırlamasından sonra, **HSI** osilatörü sistem saati olarak seçilir. Bir saat kaynağı, doğrudan veya **PLL** aracılığıyla sistem saati olarak kullanıldığında, onu durdurmak mümkün değildir.

Bir saat kaynağından başka birine geçiş, hedef saat kaynağının hazır olduğu durumlarda gerçekleşir (başlangıç gecikmesinden sonra saat kararlı veya **PLL** kilitlendiğinde). Henüz hazır olmayan bir saat kaynağı seçildiğinde, geçiş, saat kaynağı hazır olduğunda gerçekleşir. **RCC** saat kontrol kaydındaki (**RCC_CR**) durum bitleri, hangi saat(ler)in hazır olduğunu ve hangi saatin şu anda sistem saati olarak kullanıldığını gösterir.

Saat güvenlik sistemi (CSS)

Saat güvenlik sistemi yazılım tarafından etkinleştirilebilir. Bu durumda, saat dedektörü, **HSE** osilatör başlatma gecikmesinden sonra etkinleştirilir ve bu osilatör durdurulduğunda devre dışı bırakılır.

HSE saatinde bir arıza tespit edilirse, bu osilatör otomatik olarak devre dışı bırakılır, bir saat arıza olayı, gelişmiş kontrol zamanlayıcılarının **TIM1** ve **TIM8**'in kesme girişlerine gönderilir ve bir kesme, yazılıma arıza hakkında bilgi vermek üzere oluşturulur (saat güvenlik sistemi kesmesi **CSSI**), **MCU**'nun kurtarma işlemlerini gerçekleştirmesine izin verir. **CSSI**, **Cortex®-M4** ile **FPU NMI** (maskeleyen yapılmayan kesme) istisna vektörü ile ilişkilidir.

Not:CSS etkinleştirildiğinde, HSE saati başarısız olursa, CSS bir kesme oluşturur, bu da otomatik olarak bir NMI'nin oluşturulmasına neden olur. NMI, CSS kesme beklemedeki bit temizlenene kadar sürekli olarak yürütülür. Bu nedenle, uygulamanın CSS kesmesini saat kesme kaydındaki (RCC_CIR) CSSC bitini ayarlayarak NMIISR'de temizlemesi gerekmektedir.

NMI ISR, "Maskleme Yapılmayan Kesme (Non-Maskable Interrupt Interrupt Service Routine) anlamına gelir. Bu, mikrodenetleyicinin normal kesme işleminden farklı bir kesme işlemidir çünkü "maskelenemez" yani önlenemez. Bu kesme, öncelikli ve acil bir durumu belirtir ve normal kesme işlemlerinden ayrılır. NMI ISR, genellikle sistemdeki kritik hatalar veya acil durumlar gibi durumlarda çalıştırılır. Saat güvenlik sistemi kesintisi gibi önemli olaylar için kullanılabilir. Bu kesme işlemi, normal çalışma sırasında gerçekleşen kesmelerden farklı bir işlem gerektirir ve genellikle acil durumları ele almak üzere özel olarak tasarlanmış bir işlem yürütür.

Eğer HSE osilatörü doğrudan veya dolaylı olarak (dolaylı olarak, doğrudan PLL giriş saati olarak kullanılıyor olup, PLL saati sistem saati ise) sistem saati olarak kullanılıyorsa ve bir arıza tespit edilirse, sistem saati HSI osilatörüne geçer ve HSE osilatörü devre dışı bırakılır. Başarısızlık durumunda, eğer HSE osilatörü saat kaynağı olarak kullanılan PLL'in saat kaynağı ise, PLL de devre dışı bırakılır. Bu durumda, PLLI2S etkinse, HSE başarısız olduğunda PLLI2S de devre dışı bırakılır.

RTC/AWU saat kaynağı

RTCCLK saat kaynağı seçildikten sonra, seçimi değiştirmenin tek mümkün yolu güç alanını sıfırlamaktır.

RTCCLK saat kaynağı ya HSE1MHz (programlanabilir bir ön bölücü tarafından bölünmüş HSE), LSE veya LSI saati olabilir. Bu, RCC Yedek alan kontrol kaydındaki (RCC_BDCR) RTCSEL[1:0] bitlerini ve RCC saat yapılandırma kaydındaki (RCC_CFGR) RTCPRE[4:0] bitlerini programlayarak seçilir. Bu seçim, Yedek alanını sıfırlamadan değiştirilemez.

Eğer LSERTC saati olarak seçilmişse, RTC, yedek veya sistem beslemesi kaybolursa bile normal olarak çalışır. Eğer LSI, AWU saati olarak seçilirse, sistem beslemesi kaybolursa AWU durumu garanti edilmez. Eğer RTC saati olarak 2 ile 31 arasında bir değere bölünmüş HSE osilatörü kullanılıyorsa, RTC durumu, yedek veya sistem beslemesi kaybolduğunda garanti edilmez. LSE saati Yedek alanında bulunurken, HSE ve LSI saatleri bulunmaz.

Sonuç olarak:

- Eğer RTC saati olarak LSE seçilmişse:
- RTC, VDD beslemesi kapatılsa bile (VBAT beslemesi korunduğu sürece) çalışmaya devam eder.
- Eğer LSI/AWU saati olarak seçilmişse:
- VDD beslemesi kapatıldığında AWU durumu garanti edilmez.

LSI kalibrasyonu hakkında daha fazla bilgi için **LSI** saati bölümüne bakınız.

- Eğer **RTC** saati olarak **HSE** saati kullanılıyorsa:
- **RTC** durumu, **VDD** beslemesi kapatıldığında veya dahili voltaj regülatörü kapatıldığında (1.2 V alanından gücün kesilmesi) garanti edilmez.

Not: **APB1** saat frekansı **RTC** saat frekansının yedi katından az olduğunda ($f_{APB1} < 7 \times f_{RTCLK}$), yazılımın takvim zamanı ve tarih kayıtlarını okumak için kayıtları iki kez okuması gerekir. Veriler, **RTC_TR**'ye ikinci okuma erişiminde birinciyle aynı sonucu verirse doğru olur. Aksi takdirde, üçüncü bir okuma erişimi gerçekleştirilmelidir.

Watchdog clock (İzleme saati)

Bağımsız izleme (**IWDG**) donanım seçeneği veya yazılım erişimi ile başlatılırsa, **LSI** osilatörü zorunlu olarak **AÇIK** konuma getirilir ve devre dışı bırakılamaz. **LSI** osilatörünün zamanlandırılmasından sonra, saat **IWDG**'ye sağlanır.

Clock-out capability (Saat çıkış yeteneği)

İki Mikrodenetleyici saat çıkışı (**MCO**) pimi mevcuttur:

➤ **MCO1**

Yapılandırılabilir ön bölücü (1 ila 5 arası) kullanarak **MCO1** pimine (**PA8**) dört farklı saat kaynağını çıkarabilirsiniz:

- **HSI** saat
- **LSE** saat
- **HSE** saat
- **PLL** saat

İstenen saat kaynağı, **RCC** saat yapılandırma kaydındaki (**RCC_CFGR**) **MCO1PRE[2:0]** ve **MCO1[1:0]** bitleri kullanılarak seçilir.

➤ **MCO2**

Yapılandırılabilir ön bölücü (1 ila 5 arası) kullanarak **MCO2** pimine (**PC9**) dört farklı saat kaynağını çıkarabilirsiniz:

- **HSE** saat
- **PLL** saat
- Sistem saati (**SYSCLK**)
- **PLLI2S** saat

İstenen saat kaynağı, **RCC** saat yapılandırma kaydındaki (**RCC_CFGR**) **MCO2PRE[2:0]** ve **MCO2** bitleri kullanılarak seçilir.

Farklı **MCO** pinleri için, karşılık gelen **GPIO** portunun alternatif fonksiyon modunda programlanması gerekir.

MCO'ya çıkarılacak seçilen saat, maksimum **G/Ç** hızı olan **100 MHz**'yi aşmamalıdır.

TIM5/TIM11 kullanarak iç(internal)/dış(external) saat ölçümü

Şekil 23 ve **Şekil 24**'te gösterildiği gibi, **TIM5** kanal4 ve **TIM11** kanal1'in giriş yakalama kullanarak tüm yerleşik saat kaynağı üreteçlerinin frekansları dolaylı olarak ölçülebilir.

TIM5 kanal4 kullanarak iç(internal)/dış(external) saat ölçümü

TIM5, giriş yakalamasının I/O tarafından mı yoksa dahili saat tarafından mı tetiklendiğini seçmeyi sağlayan bir giriş çoğullayıcıya sahiptir. Bu seçim, **TIM5_OR** kaydındaki **TI4_RMP [1:0]** bitleri aracılığıyla gerçekleştirilir.

Kanal4 giriş yakalama ile **LSE**'nin bağlanma amacı, **HSI**'nin (bu, **HSI**'nin sistem saat kaynağı olarak kullanılmasını gerektirir) hassas bir şekilde ölçülebilmesidir. **LSE** sinyalinin ardışık kenarları arasındaki **HSI** saat sayısını alarak, dahili saat periyodunu ölçeriz. **LSE** kristallerinin yüksek hassasiyetinden (genellikle birkaç on binde bir) yararlanarak, aynı çözünürlükle dahili saat frekansını belirleyebilir ve üretim süreci ve/veya sıcaklık ve gerilimle ilgili frekans sapmalarını telafi etmek için kaynağı ayarlayabiliriz.

HSI osilatörü, bu amaç için ayrılmış, kullanıcı tarafından erişilebilir kalibrasyon bitlerine sahiptir.

Temel kavram, göreceli bir ölçüm sağlamaktır (örneğin, **HSI/LSE** oranı): dolayısıyla hassasiyet, iki saat kaynağı arasındaki oranla sıkı bir şekilde ilişkilidir. Oran ne kadar büyükse, ölçüm o kadar iyidir.

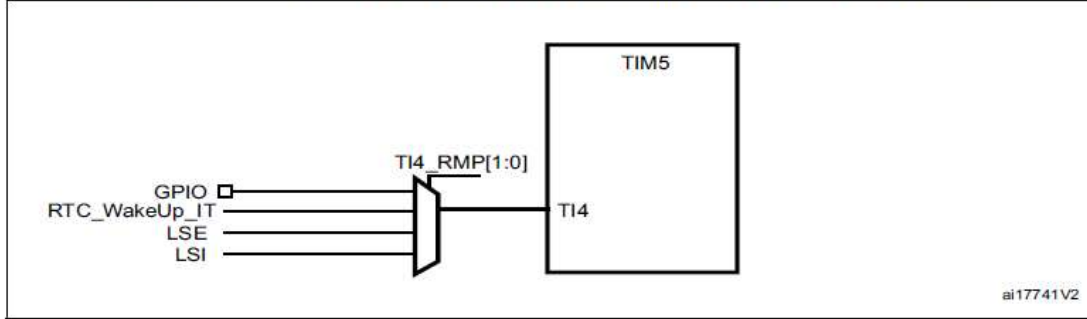
LSI frekansının ölçülmesi de mümkündür: bu, bir kristalı olmayan uygulamalar için faydalıdır. Ultradüşük güç **LSI** osilatörü geniş bir üretim süreci sapmasına sahiptir: bu nedenle **HSI** saat kaynağına karşı ölçülerek, frekansı **HSI**'nin hassasiyeti ile belirlenebilir. Ölçülen değer, daha doğru **RTC** zaman tabanı zaman aşımı değerleri (**LSI**, **RTC** saat kaynağı olarak kullanıldığında) ve/veya kabul edilebilir bir doğrulukla bir **IWDG** zaman aşımını elde etmek için kullanılabilir.

LSI frekansını ölçmek için aşağıdaki prosedürü kullanın:

1. **TIM5** zamanlayıcısını etkinleştirin ve **kanal4**'ü Giriş yakalama moduna yapılandırın(**Input Capture Mode**).
2. Kalibrasyon amaçları için **TIM5_OR** kaydındaki **TI4_RMP** bitlerini 0x01'e ayarlayarak **LSI**saat'i**TIM5**kanal4 giriş yakalama(**input Capture**) içine dahili olarak bağlayın.
3. **TIM5**yakalama(**Capture**)/karşılaştırma(**compare**) 4 etkinliğini veya kesmesini kullanarak **LSI** saati frekansını ölçün.

4. Ölçülen **LSI** frekansını, istenilen zaman tabanına bağlı olarak **RTC** ön bölücüsünü güncellemek ve/veya **IWDG** zaman aşımını hesaplamak için kullanın.

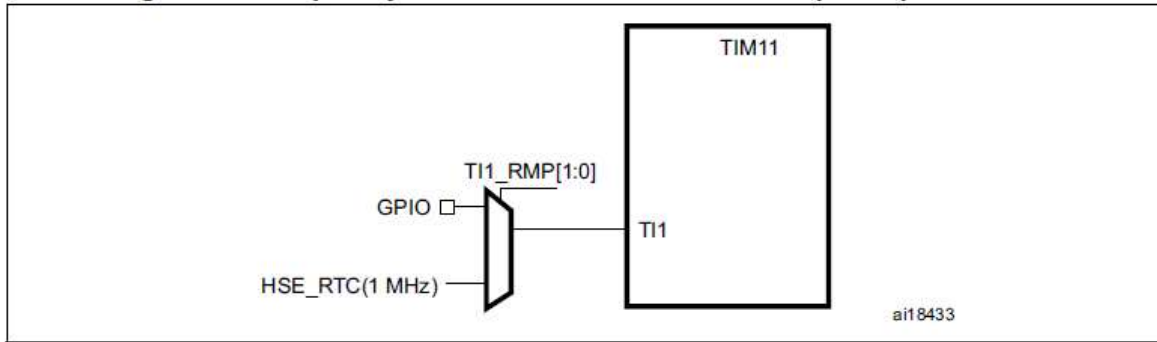
Figure 23. Frequency measurement with TIM5 in Input capture mode



TIM11 kanalı kullanarak iç(internal)/dış(external) saat ölçümü

TIM11, giriş yakalamasının **I/O** tarafından mı yoksa dahili saat tarafından mı tetiklendiğini seçmeyi sağlayan bir giriş çoğullayıcıya sahiptir. Bu seçim, **TIM11_OR** kaydındaki **TI1_RMP [1:0]** bitleri aracılığıyla gerçekleştirilir. **HSE_RTC** saati (programlanabilir bir ön bölücü tarafından bölünmüş **HSE**), dış kristal frekansının genel bir göstergesi olması için **kanal 1** giriş yakalamaya bağlanır. Bu, **HSI**'nin sistem saat kaynağı olduğu anlamına gelir. Örneğin, **harmonik** veya **alt harmonik** frekansları belirleme gerektiren **IEC 60730/IEC 61335** standartlarına uyumu sağlamak için kullanışlı olabilir.

Figure 24. Frequency measurement with TIM11 in Input capture mode



RCC clock control register (RCC_CR)

Adres ofseti: 0x00

Sıfırlama değeri: 0x0000 XX83 (burada X belirsizdir.)

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PLLI2S RDY	PLLI2S ON	PLL RDY	PLL ON	Reserved				CSS ON	HSE BYP	HSE RDY	HSE ON
				r	rw	r	rw					rw	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSICAL[7:0]								HSITRIM[4:0]					Res.	HSI RDY	HSION
r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw		r	rw

Bit 31:28 Rezerve sıfırlama değerinde tutulmalıdır.

Bit 27 **PLLI2SRDY**: PLLI2S saat hazır bayrağı (read)

PLLI2S'nin kilitli olduğunu belirtmek için donanım tarafından ayarlanır.

- 0: PLLI2S kilidi açıldı
- 1: PLLI2S kilitlendi

Bit 26 **PLLI2SON**: PLLI2S etkinleştirme (read / write)

PLLI2S'yi etkinleştirmek için yazılım tarafından ayarlanır ve temizlenir.

Durdurma veya Bekleme moduna girerken donanım tarafından temizlenir.

- 0: PLLI2S OFF (kapalı)
- 1: PLLI2S ON (Açık)

Bit 25 **PLLRDY**: Ana PLL (PLL) saat hazır bayrağı (read)

PLL'nin kilitli olduğunu belirtmek için donanım tarafından ayarlanır.

- 0: PLL kilidi açık
- 1: PLL kilitlendi

Bit 24 **PLLON**: Ana PLL (PLL) etkinleştirme (read / write)

PLL'yi etkinleştirmek için yazılım tarafından ayarlanır ve temizlenir.

Durdurma veya Bekleme moduna girerken donanım tarafından temizlenir. PLL saati sistem saati olarak kullanılıyorsa bu bit sıfırlanamaz.

- 0: PLL OFF (KAPALI)
- 1: PLL ON (AÇIK)

Bit 23:20 Rezerve, sıfırlama değerinde tutulmalıdır.

Bit 19 **CSSON**: Saat güvenlik sistemi etkinleştirme (read / write)

Saat güvenlik sistemini etkinleştirmek için yazılım tarafından ayarlanır ve silinir. CSSON ayarlandığında, saat dedektörü HSE osilatörü hazır olduğunda donanım tarafından etkinleştirilir ve bir osilatör arızası tespit edilirse donanım tarafından devre dışı bırakılır.

0: Saat güvenlik sistemi KAPALI (Saat dedektörü KAPALI)

1: Saat güvenlik sistemi AÇIK (HSE osilatörü kararlıysa saat dedektörü AÇIK, değilse KAPALI)

Bit 18 **HSEBYP**:HSE saat baypası (read / write)

Osilatörü harici bir saatle bypass etmek için yazılım tarafından ayarlanır ve temizlenir. Harici saatin cihaz tarafından kullanılabilmesi için HSEON biti ile etkinleştirilmesi gerekir.

HSEBYP biti sadece HSE osilatörü devre dışı bırakıldığında yazılabilir.

0: HSE osilatörü bypass edilmemiş

1: HSE osilatörü harici bir saat ile bypass edilir

Bit 17 **HSERDY**:HSE saat hazır bayrağı (read)

HSE osilatörünün istikrarlı hale geldiğini belirlemek için donanım tarafından bir işlem gerçekleştirilir. HSEON biti temizlendiğinde, HSERDY biti, HSE osilatörünün stabil hale gelmesinden sonra, yani yaklaşık 6 HSE osilatör saat döngüsügeçtikten sonra düşük seviyeye iner.

0: HSE osilatörü hazır değil

1: HSE osilatörü hazır

Bit 16 **HSEON**:HSE saat etkinleştirme (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

Durdurma veya Bekleme moduna girerken HSE osilatörünü durdurmak için donanım tarafından temizlenir. HSE osilatörü doğrudan veya dolaylı olarak sistem saati olarak kullanılıyorsa bu bit sıfırlanamaz.

0: HSE osilatörü KAPALI

1: HSE osilatörü AÇIK

Bitler 15:8 **HSICAL**[7:0]: Dâhili yüksek hızlı saat kalibrasyonu (read)

Bu bitler başlangıçta otomatik olarak başlatılır.

Bit 7:3 **HSITRIM**[4:0]: Dâhili yüksek hızlı saat kırpma (read / write)

Bu bitler HSICAL[7:0] bitlerine eklenen kullanıcı tarafından programlanabilir ek bir kırpma değeri sağlar. Dâhili HSI RC'nin frekansını etkileyen voltaj ve sıcaklık değişimlerine uyum sağlamak için programlanabilir.

Bit 2: Rezerve, sıfırlama değerinde tutulmalıdır.

Bit 1 **HSIRDY**:Dâhili yüksek hızlı saat hazır bayrağı (read)

HSI osilatörünün kararlı olduğunu belirtmek için donanım tarafından ayarlanır. HSION biti temizlendikten sonra, HSIRDY 6 HSI saat çevriminden sonra düşük seviyeye iner.

- 0: HSI osilatörü hazır değil
- 1: HSI osilatörü hazır

Örnek kod üstünde göstermek adına aşağıdaki kodu ekliyorum. Ama bu kullanım veri yaprağına uygun kullanımdır. Genel olarak kullanım kodu.

```
// HSI osilatörünü etkinleştir
RCC->CR |= RCC_CR_HSION;

// HSI osilatörünün kararlı hale gelmesini bekle
while(!(RCC->CR & RCC_CR_HSIRDY));
```

```
// HSI osilatörünü etkinleştir
RCC->CR |= RCC_CR_HSION;

// HSI osilatörünün kararlı hale gelmesini bekleyin
int timeout = 6; // Maksimum 6 döngü bekle
while(!(RCC->CR & RCC_CR_HSIRDY) && timeout--);

// HSI osilatörünü devre dışı bırak
RCC->CR &= ~RCC_CR_HSION;
```

Bit 0 **HSION**:Dâhili yüksek hızlı saat etkinleştirme (read / write)

Yazılım tarafından ayarlanır ve silinir.

Durdurma veya Bekleme modundan çıkarken veya doğrudan ya da dolaylı olarak sistem saati olarak kullanılan HSE osilatörünün arızalanması durumunda HSI osilatörünü AÇIK konuma zorlamak için donanım tarafından ayarlanır. **HSI** doğrudan veya dolaylı olarak sistem saati olarak kullanılıyorsa bu bit temizlenemez.

- 0: HSI osilatörü KAPALI
- 1: HSI osilatörü AÇIK

RCC PLL configuration register (RCC_PLLCFGR)

Adres ofseti:0x04

Sıfırlama değeri:0x2400 3010

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

Bu kayıt PLL saat çıkışlarını formüllere göre yapılandırmak için kullanılır:

- $f_{(VCO \text{ clock})} = f_{(PLL \text{ clock input})} \times (PLL_N / PLL_M)$
- $f_{(PLL \text{ general clock output})} = f_{(VCO \text{ clock})} / PLL_P$
- $f_{(USB OTG FS, SDIO, RNG \text{ clock output})} = f_{(VCO \text{ clock})} / PLL_Q$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved				PLLQ3	PLLQ2	PLLQ1	PLLQ0	Reserved	PLLSRC	Reserved				PLL1P1	PLL1P0
				rw	rw	rw	rw		rw					rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL1N									PLL1M5	PLL1M4	PLL1M3	PLL1M2	PLL1M1	PLL1M0
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Örnek: **STM32F4-DISC** board'nun kristal değeri 8MHZ'dir. Yani PLL saat girişi oluyor. PLLN:168, PLLP: 2 ve PLLM:4'tur.

$$f(VCO\ clock) = f(PLL\ clock\ input) * \left(\frac{PLL1N}{PLL1M}\right)$$

$$f(VCO\ clock) = 8MHZ * \left(\frac{168}{4}\right) = 336$$

$$f(PLL\ general\ clock\ input) = \frac{f(VCO\ clock)}{PLL1P}$$

$$f(PLL\ general\ clock\ input) = \frac{336}{2} = 168MHZ$$

Bit 31:28Rezerve,sıfırlama değeriinde tutulmalıdır.

Bit 27:24 **PLLQ**:USB OTG FS, SDIO ve rastgele sayı üretici saatleri için ana PLL (PLL) bölme faktörü.(read / write)

USB OTG FS saatinin, rastgele sayı üretici saatinin ve SDIO saatinin frekansını kontrol etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bitler yalnızca PLL devre dışı bırakıldığında yazılmalıdır.

Dikkat: USB OTG FS'nin düzgün çalışması için 48 MHz saat gerekir. SDIO ve rastgele sayı üreticinin düzgün çalışması için 48 MHz'den daha düşük veya eşit bir frekansa ihtiyacı vardır.

USB OTG FS clock frequency = VCO frequency / PLLQ ile $2 \leq PLLQ \leq 15$

0000: PLLQ = 0, yanlış yapılandırma

0001: PLLQ = 1, yanlış yapılandırma

0010: PLLQ = 2

0011: PLLQ = 3

0100: PLLQ = 4

...

1111: PLLQ = 15

Bit 23Rezerve, sıfırlama değeriinde tutulmalıdır.

Bit 22 **PLLSRC**: Ana PLL (PLL) ve ses PLL (PLLI2S) giriş saat kaynağı(read / write)

PLL ve PLLI2S saat kaynağını seçmek için yazılım tarafından ayarlanır ve temizlenir.

Bu bit yalnızca PLL ve PLLI2S devre dışı bırakıldığında yazılabilir.

0: HSI saati PLL ve PLLI2S saat girişi olarak seçilir

1: PLL ve PLLI2S saat girişi olarak seçilen HSE osilatör saati seçilir.

Bit 21:18 Rezerve, sıfırlama değerinde tutulmalıdır.

Bit 17:16 **PLLP**: Ana sistem saati için Ana PLL (PLL) bölme faktörü(read / write)

Genel PLL çıkış saatinin frekansını kontrol etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bitler sadece PLL devre dışı bırakıldığında yazılabilir.

Dikkat: Yazılım, bu alanda 168 MHz'i aşmamak için bu bitleri doğru şekilde ayarlamalıdır.

PLL çıkış saati frekansı = VCO frekansı / PLLP = 2, 4, 6 veya 8 ile PLLP

00: PLLP = 2

01: PLLP = 4

10: PLLP = 6

11: PLLP = 8

Bit 14:6 **PLLN**: VCO için Ana PLL (PLL) çarpım faktörü(read / write)

VCO'nun çarpma faktörünü kontrol etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bitler yalnızca PLL devre dışı bırakıldığında yazılabilir. Bu bitleri yazmak için yalnızca half-word ve Word erişimlerine izin verilir.

Dikkat: Yazılımın VCO çıkışını sağlamak için bu bitleri doğru şekilde ayarlaması gerekir.

frekansı 100 ile 432 MHz arasındadır.

VCO çıkış frekansı = VCO giriş frekansı $\times$ PLLN ile $50 \leq \text{PLLN} \leq 432$

000000000: PLLN = 0, yanlış yapılandırma

000000001: PLLN = 1, yanlış yapılandırma

...

000110010: PLLN = 50

...

001100011: PLLN = 99

001100100: PLLN = 100

...

110110000: PLLN = 432

110110001: PLLN = 433, yanlış yapılandırma

...

111111111: PLLN = 511, yanlış yapılandırma

Not: 1 MHz'den daha yüksek VCO giriş frekansı için 50 ile 99 arasında değişen çarpma faktörleri mümkündür. Ancak minimum VCO çıkış frekansının yukarıda belirtilen değere uygun olmasına dikkat edilmelidir.

Bit 5:0 **PLLM**: Ana PLL (PLL) ve ses PLL(PLLI2S) giriş saati için bölme faktörü(read/write)

PLL ve PLLI2S giriş saatini VCO'dan önce bölmek için yazılım tarafından ayarlanır ve temizlenir.

Bu bitler yalnızca PLL ve PLLI2S devre dışı bırakıldığında yazılabilir.

Dikkat: VCO giriş frekansının 1 ila 2 MHz arasında olmasını sağlamak için yazılımın bu bitleri doğru şekilde ayarlaması gerekir. PLL titreşimini sınırlamak için 2 MHz'lik bir frekans seçilmesi önerilir.

VCO giriş frekansı = PLL giriş saat frekansı / PLLM ile $2 \leq PLLM \leq 63$

000000: PLLM = 0, yanlış yapılandırma

000001: PLLM = 1, yanlış yapılandırma

000010: PLLM = 2

000011: PLLM = 3

000100: PLLM = 4

...

111110: PLLM = 62

111111: PLLM = 63

RCC clock configuration register (RCC_CFGR)

Adres ofseti: 0x08

Sıfırlama değeri: 0x0000 0000

Erişim: $0 \leq$ bekleme durumu ≤ 2 , Word, half-word ve bayt erişimi

1 veya 2 bekleme durumu yalnızca erişim bir saat kaynağı değişimi sırasında gerçekleşirse eklenir.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2		MCO2 PRE[2:0]			MCO1 PRE[2:0]			I2SSC R	MCO1		RTCPRE[4:0]				
rw		rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRE2[2:0]			PPRE1[2:0]			Reserved		HPRE[3:0]				SWS1	SWS0	SW1	SW0
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	r	r	rw	rw

Bitler 31:30 **MCO2[1:0]**: Mikrodenetleyici saat çıkışı 2 (31 read/write / 30 none)

Yazılım tarafından ayarlanır ve temizlenir. Saat kaynağı seçimi MCO2 üzerinde gürültü oluşturabilir. Bu bitlerin yalnızca sıfırlama sonrasında harici osilatörleri ve PLL'leri etkinleştirmeden önce yapılandırılması şiddetle önerilir.

00: Sistem saati (SYSCLK) seçilmiş

01: PLLI2S saati seçilmiş

10: HSE osilatör saati seçilmiş

11: PLL saati seçilmiş

Bitler 27:29 **MCO2PRE**:MCO2 bölücüsü(read / write)

Yazılım tarafından ayarlanır ve temizlenir, MCO2'nin bölücüsünü yapılandırmak için kullanılır. Bu bölücünün değiştirilmesi MCO2 üzerinde gürültü oluşturabilir. Bu bölücünün yalnızca sıfırlama sonrasında harici osilatörleri ve PLL'leri etkinleştirmeden önce değiştirilmesi şiddetle önerilir.

- 0xx: Bölme yok (no division)
- 100: 2'ye bölme (division by 2)
- 101: 3'e bölme (division by 3)
- 110: 4'e bölme (division by 4)
- 111: 5'e bölme (division by 5)

Bitler 24:26 **MCO1PRE**:MCO1 bölücüsü(read / write)

Yazılım tarafından ayarlanır ve temizlenir, MCO1'in bölücüsünü yapılandırmak için kullanılır. Bu bölücünün değiştirilmesi MCO1 üzerinde gürültü oluşturabilir. Bu bölücünün yalnızca sıfırlama sonrasında harici osilatörleri ve PLL'leri etkinleştirmeden önce değiştirilmesi şiddetle önerilir.

- 0xx: Bölme yok (no division)
- 100: 2'ye bölme (division by 2)
- 101: 3'e bölme (division by 3)
- 110: 4'e bölme (division by 4)
- 111: 5'e bölme (division by 5)

Bit 23 **I2SSRC**: I2S saat seçimi(read / write)

Yazılım tarafından ayarlanır ve temizlenir. Bu bit, I2S saatinin kaynağını PLLI2S saati ve harici saat arasında seçmeye izin verir. Bu bitin yalnızca sıfırlama sonrasında ve I2S modülünü etkinleştirmeden önce değiştirilmesi şiddetle önerilir.

- 0: I2S saat kaynağı olarak PLLI2S saati kullanılır
- 1: I2S saat kaynağı olarak I2S_CKIN pimine eşlenen harici saat kullanılır.

Bitler 22:21 **MCO1**: Mikrodenetleyici saati çıkışı 1 (22 read/write / 21 none)

Yazılım tarafından ayarlanır ve temizlenir. Saat kaynağı seçimi, MCO1 üzerinde gürültü oluşturabilir. Harici osilatörleri ve PLL'leri etkinleştirmeden önce yalnızca sıfırlama sonrasında bu bitlerin yapılandırılması şiddetle önerilir.

- 00: HSI saati seçilmiş
- 01: LSE osilatörü seçilmiş
- 10: HSE osilatörü saati seçilmiş
- 11: PLL saati seçilmiş

Bitler 20:16 **RTCPRE**:RTC saati için HSE bölme faktörü(read / write)

Yazılım tarafından ayarlanır ve temizlenir; RTC için 1 MHz saat oluşturmak için HSE saat girişini bölmek için kullanılır.

Dikkat: RTC'ye sağlanan saatin 1 MHz olduğundan emin olmak için yazılımın bu bitleri doğru bir şekilde ayarlaması gerekmektedir. RTC saat kaynağı seçilmeden önce gerekiyorsa bu bitler yapılandırılmalıdır.

- 00000: saat yok
- 00001: saat yok
- 00010: HSE/2
- 00011: HSE/3
- 00100: HSE/4
- ...

11110: HSE/30

11111: HSE/31

Bitler 15:13 **PPRE2**:APB yüksek hızlı ön bölücü (APB2)([read](#) / [write](#))

Yazılım tarafından ayarlanır ve temizlenir; APB yüksek hızlı saat bölme faktörünü kontrol etmek için kullanılır.

Dikkat: Yazılımın bu bitleri doğru bir şekilde ayarlaması, bu alandaki saatin 84MHz'yi aşmamasını sağlamak için gereklidir.

Saatler, PPRE2 yazıldıktan sonra 1 ila 16 AHB döngüsü sonra yeni bölücü faktörle bölünür.

0xx: AHB saati bölünmemiş

100: AHB saati 2 ile bölünmüş

101: AHB saati 4 ile bölünmüş

110: AHB saati 8 ile bölünmüş

111: AHB saati 16 ile bölünmüş

Bitler 12:10 **PPRE1**:APB Düşük hızlı ön bölücü (APB1)([read](#) / [write](#))

Yazılım tarafından ayarlanır ve temizlenir; APB düşük hızlı saat bölme faktörünü kontrol etmek için kullanılır.

Dikkat: Yazılımın bu bitleri doğru bir şekilde ayarlaması, bu alandaki saatin 42MHz'yi aşmamasını sağlamak için gereklidir.

Saatler, PPRE1 yazıldıktan sonra 1 ila 16 AHB döngüsü sonra yeni bölücü faktörle bölünür.

0xx: AHB saati bölünmemiş

100: AHB saati 2 ile bölünmüş

101: AHB saati 4 ile bölünmüş

110: AHB saati 8 ile bölünmüş

111: AHB saati 16 ile bölünmüş

Bitler 9:8Rezerve, sıfır değerinde tutulmalıdır.

Bitler 7:4 **HPRE**:AHB ön bölücüsü([read](#) / [write](#))

Yazılım tarafından ayarlanır ve temizlenir, AHB saat bölücü faktörünü kontrol etmek için.

Dikkat: Saatler, HPRE yazıldıktan sonra 1 ila 16 AHB döngüsünde yeni ön bölücü faktörüyle bölünür.

Dikkat: Ethernet kullanıldığında AHB saat frekansı en az 25 MHz olmalıdır.

- 0xxx: sistem saati bölünmemiş
- 1000: sistem saati 2'ye bölünmüş
- 1001: sistem saati 4'e bölünmüş
- 1010: sistem saati 8'e bölünmüş
- 1011: sistem saati 16'ya bölünmüş
- 1100: sistem saati 64'e bölünmüş
- 1101: sistem saati 128'e bölünmüş
- 1110: sistem saati 256'ya bölünmüş
- 1111: sistem saati 512'ye bölünmüş

Bitler 3:2 **SWS**: Sistem saati geçiş durumu(read)

Donanım tarafından ayarlanır ve temizlenir, hangi saat kaynağının sistem saati olarak kullanıldığını göstermek için.

- 00: HSI osilatörü sistem saati olarak kullanılır
- 01: HSE osilatörü sistem saati olarak kullanılır
- 10: PLL sistem saati olarak kullanılır
- 11: uygulanamaz

Bitler 1:0 **SW**: Sistem saat geçişi(read / write)

Yazılım tarafından ayarlanır ve temizlenir, sistem saat kaynağını seçmek için.

Donanım tarafından ayarlanır, Durdurma veya Bekleme modundan ayrılırken veya doğrudan veya dolaylı olarak sistem saati olarak kullanılan HSE osilatörünün başarısız olduğu durumlarda HSI seçimini zorlamak için.

- 00: HSI osilatörü sistem saati olarak seçilmiştir
- 01: HSE osilatörü sistem saati olarak seçilmiştir
- 10: PLL sistem saati olarak seçilmiştir
- 11: izin verilmez

RCC clock interrupt register (RCC_CIR)

Adres ofseti: 0x0C

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								CSSC	Reserv ed	PLLI2S RDYC	PLL RDYC	HSE RDYC	HSI RDYC	LSE RDYC	LSI RDYC
								w		w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		PLLI2S RDYIE	PLL RDYIE	HSE RDYIE	HSI RDYIE	LSE RDYIE	LSI RDYIE	CSSF	Reserv ed	PLLI2S RDYF	PLL RDYF	HSE RDYF	HSI RDYF	LSE RDYF	LSI RDYF
		rw	rw	rw	rw	rw	rw	r		r	r	r	r	r	r

Bitler 31-24Rezerve, sıfır değerinde tutulmalıdır.

Bit 23 **CSSC**: Saat güvenlik sistemi kesme temizleme (**write**)

Bu bit yazılım tarafından CSSF bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: CSSF bayrağı temizlenir

Bitler 22Rezerve, sıfır değerinde tutulmalıdır.

Bit 21 **PLLI2SRDYC**:PLLI2S hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından PLLI2SRDYF bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: PLLI2SRDYF temizlendi

Bit 20 **PLLRDYC**: Ana PLL (PLL) hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından PLLRDYF bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: PLLRDYF temizlendi

Bit 19 **HSERDYC**: HSE hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından **HSERDYF** bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: HSERDYF temizlendi

Bit 18 **HSIRDYC**:HSI hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından **HSIRDYF** bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: HSIRDYF temizlendi

Bit 17 **LSERDYC**:LSE hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından LSERDYF bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: LSERDYF temizlendi

Bit 16 **LSIRDYC**:LSI hazır kesmesi temizleme (**write**)

Bu bit, yazılım tarafından LSIRDYF bayrağını temizlemek için ayarlanır.

0: Etkisi yok

1: LSIRDYF temizlendi

Bitler 15-12Rezerve, sıfır değerinde tutulmalıdır.

Bit 13 **PLLI2SRDYIE**:PLLI2S hazır kesme izni(**read / write**)

Yazılım tarafından **PLLI2S** kilidi nedeniyle oluşan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: PLLI2S kilidi kesmesi devre dışı

1: PLLI2S kilidi kesmesi etkin

Bit 12 **PLLRDYIE**: Ana PLL (PLL) hazır kesme izni

Yazılım tarafından **PLL** kilidi nedeniyle oluşan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: PLL kilidi kesmesi devre dışı

1: PLL kilidi kesmesi etkin

Bit 11 **HSERDYIE**: HSE hazır kesme izni(**read / write**)

Yazılım tarafından HSE osilatörünün stabilizasyonundan kaynaklanan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: HSE hazır kesmesi devre dışı

1: HSE hazır kesmesi etkin

Bit 10 **HSIRDYIE**:HSI hazır kesme izni(**read / write**)

Yazılım tarafından HSI osilatör stabilizasyonundan kaynaklanan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: HSI hazır kesme devre dışı

1: HSI hazır kesme etkin

Bit 9 **LSERDYIE**: LSE hazır kesme izni(**read / write**)

Yazılım tarafından LSE osilatör stabilizasyonundan kaynaklanan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: LSE hazır kesme devre dışı

1: LSE hazır kesme etkin

Bit 8 **LSIRDYIE**:LSI hazır kesme izni(**read / write**)

Yazılım tarafından LSI osilatör stabilizasyonundan kaynaklanan kesmenin etkinleştirilmesi/devre dışı bırakılması için ayarlanır.

0: LSI hazır kesme devre dışı

1: LSI hazır kesme etkin

Bit 7 CSSF: Saat güvenlik sistemi kesme bayrağı(read)

HSE osilatöründe bir arıza tespit edildiğinde donanım tarafından ayarlanır.

Yazılım, CSSC bitini ayarlayarak temizler.

0: HSE saat arızasından kaynaklanan saat güvenlik kesmesi yok

1: HSE saat arızasından kaynaklanan saat güvenlik kesmesi var

Bitler 6 Rezerve, sıfır değerinde tutulmalıdır.

Bit 5 PLLI2SRDYF:PLLI2S hazır kesme bayrağı(read)

PLLI2S kilidi ve PLLI2SRDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, PLLI2SDYC bitini ayarlayarak temizler.

0: PLLI2S kilidi nedeniyle saat hazır kesmesi yok

1: PLLI2S kilidi nedeniyle saat hazır kesmesi var

Bit 4 PLLRDYF: Ana PLL (PLL) hazır kesme bayrağı(read)

PLL kilidi ve PLLRDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, PLLRDYC bitini ayarlayarak temizler.

0: PLL kilidi nedeniyle saat hazır kesmesi yok

1: PLL kilidi nedeniyle saat hazır kesmesi var

Bit 3 HSERDYF:HSE hazır kesme bayrağı(read)

Harici Yüksek Hızlı saatin istikrarlı hale gelmesi ve HSERDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, HSERDYC bitini ayarlayarak temizler.

0: HSE osilatöründen kaynaklanan saat hazır kesmesi yok

1: HSE osilatöründen kaynaklanan saat hazır kesmesi var

Bit 2 HSIRDYF:HSI hazır kesme bayrağı(read)

İç Yüksek Hızlı saat istikrarlı hale geldiğinde ve HSIRDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, HSIRDYC bitini ayarlayarak temizler.

0: HSI osilatöründen kaynaklanan saat hazır kesmesi yok

1: HSI osilatöründen kaynaklanan saat hazır kesmesi var

Bit 1 LSERDYF: LSE hazır kesme bayrağı(read)

Harici Düşük Hızlı saat istikrarlı hale geldiğinde ve LSERDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, LSERDYC bitini ayarlayarak temizler.

0: LSEosilatöründen kaynaklanan saat hazır kesmesi yok

1: LSEosilatöründen kaynaklanan saat hazır kesmesi var

Bit 0 LSIRDYF:LSI hazır kesme bayrağı(read)

İç Düşük Hızlı saat istikrarlı hale geldiğinde ve LSIRDYIE ayarlandığında donanım tarafından ayarlanır.

Yazılım, LSIRDYC bitini ayarlayarak temizler.

0: LSI osilatöründen kaynaklanan saat hazır kesmesi yok

1: LSI osilatöründen kaynaklanan saat hazır kesmesi var

RCC AHB1 peripheral reset register (RCC_AHB1RSTR)

Adres ofseti: 0x10

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		OTGHS RST	Reserved			ETHMAC RST	Reserved		DMA2 RST	DMA1 RST	Reserved				
		rw				rw			rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		CRCRST	Reserved			GPIOI RST	GPIOH RST	GPIOGG RST	GPIOF RST	GPIOE RST	GPIOD RST	GPIOC RST	GPIOB RST	GPIOA RST	
		rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	

Bitler 31:30 Rezerve, sıfır değerinde tutulmalıdır.

Bit 29: OTGHSRST:USB OTG HS modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG HS modülünü sıfırlamaz

1: USB OTG HS modülünü sıfırlar

Bitler 28:26Rezerve,sıfır değerinde tutulmalıdır.

Bit 25 ETHMACRST:Ethernet MAC sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet MAC'i sıfırlamaz

1: Ethernet MAC'i sıfırlar

Bitler 24:23Rezerve, sıfır değerinde tutulmalıdır.

Bit 22 DMA2RST:DMA2 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA2'yi sıfırlamaz

1: DMA2'yi sıfırlar

Bit 21 **DMA1RST**: DMA1 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA1'i sıfırlamaz

1: DMA1'i sıfırlar

Bitler 20:13 Rezerve, sıfır değerinde tutulmalıdır.

Bit 12 **CRCRST**: CRC sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CRC'yi sıfırlamaz

1: CRC'yi sıfırlar

Bitler 11:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **GPIOIRST**: IO port I sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port I'i sıfırlamaz

1: IO port 'i sıfırlar

Bit 7 **GPIOHRST**: IO port H sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port H'yi sıfırlamaz

1: IO port H'yi sıfırlar

Bit 6 **GPIOGRST**: IO port G sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port G'yi sıfırlamaz

1: IO port G'yi sıfırlar

Bit 5 **GPIOFRST**: IO port F sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port F'yi sıfırlamaz

1: IO port F'yi sıfırlar

Bit 4 **GPIOERST**: IO port E sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port E'yi sıfırlamaz

1: IO port E'yi sıfırlar

Bit 3 **GPIODRST**: IO port D sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port D'yi sıfırlamaz

1: IO port D'yi sıfırlar

Bit 2 **GPIOCRST**: IO port Csıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port C'yi sıfırlamaz

1: IO port C'yi sıfırlar

Bit 1 **GPIOBRST**: IO port Bsıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port B'yi sıfırlamaz

1: IO port B'yi sıfırlar

Bit 0 **GPIOARST**: IO port Asıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port A'yı sıfırlamaz

1: IO port A'yı sıfırlar

RCC AHB2 peripheral reset register (RCC_AHB2RSTR)

Adres ofseti: 0x14

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OTGFS RST	RNG RST	HASH RST	CRYP RST	Reserved			DCMI RST
								rw	rw	rw	rw				rw

Bitler 31:8 Rezerve, sıfır değerinde tutulmalıdır.

Bit 7 **OTGFSRST**: USB OTG FS modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG FS modülünü sıfırlamaz

1: USB OTG FS modülünü sıfırlar

Bit 6 **RNGRST**: Rastgele sayı üretici modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Rastgele sayı üretici modülünü sıfırlamaz

1: Rastgele sayı üretici modülünü sıfırlar

Bit 5 **HASHRST**: Hash modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Hash modülünü sıfırlamaz

1: Hash modülünü sıfırlar

Bit 4 **CRYPRST**: Şifreleme modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Şifreleme modülünü sıfırlamaz

1: Şifreleme modülünü sıfırlar

Bitler 3:1 Rezerve, sıfır değerinde tutulmalıdır.

Bit 0 **DCMIRST**: Kamera arayüzü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Kamera arayüzünü sıfırlamaz

1: Kamera arayüzünü sıfırlar

RCC AHB3 peripheral reset register (RCC_AHB3RSTR)

Adres ofseti: 0x18

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSMCRST	
														rw	

Bitler 31:1 Rezerve, sıfır değerinde tutulmalıdır.

Bit 0: **FSMCRST**: Esnek statik bellek denetleyicisi modülü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: FSMC modülünü sıfırlamaz

1: FSMC modülünü sıfırlar

RCC APB1 peripheral reset register (RCC_APB1RSTR)

Adres ofseti: 0x20

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DACRST	PWR RST	Reserved	CAN2 RST	CAN1 RST	Reserved	I2C3 RST	I2C2 RST	I2C1 RST	UART5 RST	UART4 RST	UART3 RST	UART2 RST	Reserved
		rw	rw			rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 RST	SPI2 RST	Reserved		WWDG RST	Reserved		TIM14 RST	TIM13 RST	TIM12 RST	TIM7 RST	TIM6 RST	TIM5 RST	TIM4 RST	TIM3 RST	TIM2 RST
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

Bitler 31:30Rezerve, sıfır değerinde tutulmalıdır.

Bit 29 **DACRST**: DAC sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DAC arayüzünü sıfırlamaz

1: DAC arayüzünü sıfırlar

Bit 28 **PWRRST**: Güç arayüzü sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Güç arayüzünü sıfırlamaz

1: Güç arayüzünü sıfırlar

Bit 27Rezerve, sıfır değerinde tutulmalıdır.

Bit 26 **CAN2RST**: CAN2 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CAN2'yi sıfırlamaz

1: CAN2'yi sıfırlar

Bit 25 **CAN1RST**: CAN1 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CAN1'i sıfırlamaz

1: CAN1'i sıfırlar

Bit 24Rezerve, sıfır değerinde tutulmalıdır.

Bit 23 **I2C3RST**: I2C3 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C3'ü sıfırlamaz

1: I2C3'ü sıfırlar

Bit 22 **I2C2RST**: I2C2 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C2'yi sıfırlamaz

1: I2C2'yi sıfırlar

Bit 21 **I2C1RST**: I2C1 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C1'i sıfırlamaz

1: I2C1'i sıfırlar

Bit 20 **UART5RST**: UART5 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: UART5'i sıfırlamaz

1: UART5'i sıfırlar

Bit 19 **UART4RST**: USART4 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: UART4'ü sıfırlamaz

1: UART4'ü sıfırlar

Bit 18 **USART3RST**: USART3 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART3'ü sıfırlamaz

1: USART3'ü sıfırlar

Bit 17 **USART2RST**: USART2 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART2'yi sıfırlamaz

1: USART2'yi sıfırlar

Bit 16 Rezerve, sıfır değerinde tutulmalıdır.

Bit 15 **SPI3RST**: SPI3 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI3'ü sıfırlamaz

1: SPI3'ü sıfırlar

Bit 14 **SPI2RST**: SPI2 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI2'yi sıfırlamaz

1: SPI2'yi sıfırlar

Bitler 13:12 Rezerve, sıfır değerinde tutulmalıdır.

Bit 11 **WWDGRST**: Window Watchdog sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Window izleyicisini sıfırlamaz

1: Window izleyicisini sıfırlar

Bitler 10:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **TIM14RST**: TIM14 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM14'ü sıfırlamaz

1: TIM14'ü sıfırlar

Bit 7 **TIM13RST**: TIM13 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM13'ü sıfırlamaz

1: TIM13'ü sıfırlar

Bit 6 **TIM12RST**: TIM12 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM12'yi sıfırlamaz

1: TIM12'yi sıfırlar

Bit 5 **TIM7RST**: TIM7sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM7'yi sıfırlamaz

1: TIM7'yi sıfırlar

Bit 4 **TIM6RST**: TIM6sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM6'yı sıfırlamaz

1: TIM6'yı sıfırlar

Bit 3 **TIM5RST**: TIM5sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM5'i sıfırlamaz

1: TIM5'i sıfırlar

Bit 2 **TIM4RST**: TIM4sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM4'ü sıfırlamaz

1: TIM4'ü sıfırlar

Bit 1 **TIM3RST**: TIM3sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM3'ü sıfırlamaz

1: TIM3'ü sıfırlar

Bit 0 **TIM2RST**: TIM2sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM2'yi sıfırlamaz

1: TIM2'yi sıfırlar

RCC APB2 peripheral reset register (RCC_APB2RSTR)

Adres ofseti:0x24

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 RST	TIM10 RST	TIM9 RST
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser- ved	SYSCF G RST	Reser- ved	SPI1 RST	SDIO RST	Reserved		ADC RST	Reserved		USART 6 RST	USART 1 RST	Reserved		TIM8 RST	TIM1 RST
	rw		rw	rw			rw			rw	rw			rw	rw

Bitler 31:19Rezerve, sıfır değerinde tutulmalıdır.

Bit 18 **TIM11RST**: TIM11 sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM11'i sıfırlamaz

1: TIM11'i sıfırlar

Bit 17 **TIM10RST**: TIM10 sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM10'u sıfırlamaz

1: TIM10'u sıfırlar

Bit 16 **TIM9RST**: TIM9 sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM9'u sıfırlamaz

1: TIM9'u sıfırlar

Bit 15Rezerve, sıfır değerinde tutulmalıdır.

Bit 14 **SYSCFGRST**: Sistem yapılandırma denetleyicisi sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0:Sistem yapılandırma denetleyicisini sıfırlamaz

1:Sistem yapılandırma denetleyicisini sıfırlar

Bit 13Rezerve, sıfır değerinde tutulmalıdır.

Bit 12 **SPI1RST**: SPI1 sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI1'i sıfırlamaz

1: SPI1'i sıfırlar

Bit 11 **SDIORST**: SDIO sıfırlama (read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SDIO modülünü sıfırlamaz

1: SDIO modülünü sıfırlar

Bitler 10:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **ADCRST**: ADC arayüzü sıfırlama (tüm ADC'ler için ortak)(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: ADCarayüzünü sıfırlamaz

1: ADCarayüzünü sıfırlar

Bitler 7:6Rezerve, sıfır değerinde tutulmalıdır.

Bit 5 **USART6RST**: USART6 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART6'yı sıfırlamaz

1: USART6'yı sıfırlar

Bit 4 **USART1RST**: USART1 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART1'i sıfırlamaz

1: USART1'i sıfırlar

Bitler 3:2Rezerve, sıfır değerinde tutulmalıdır.

Bit 1 **TIM8RST**: TIM8 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM8'i sıfırlamaz

1: TIM8'i sıfırlar

Bit 0 **TIM1RST**: TIM1 sıfırlama(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM1'i sıfırlamaz

1: TIM1'i sıfırlar

RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Adres ofseti: 0x30

Sıfırlama değeri: 0x0010 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	OTGHS ULPIEN	OTGHS SEN	ETHMACPTP EN	ETHMACRX EN	ETHMACTX EN	ETHMACEN	Reserved			DMA2EN	DMA1EN	CCMDAT ARAMEN	Res.	BKPSR AMEN	Reserved
	rw	rw	rw	rw	rw	rw				rw	rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCE N	Reserved			GPIOEN	GPIOH EN	GPIOG EN	GPIOF EN	GPIOEEN	GPIOD EN	GPIOC EN	GPIOB EN	GPIOA EN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 Rezerve, sıfır olarak tutulmalıdır.

Bit 30 **OTGHSULPIEN**: USB OTG HSULPI saatini etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir. Bu bit, OTG_HS'nin FS modunda kullanıldığında temizlenmelidir.

0: USB OTG HS ULPI saati devre dışı bırakıldı

1: USB OTG HS ULPI saati etkinleştirildi

Bit 29 **OTGHSSEN**: USB OTG HS saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG HS saati devre dışı bırakıldı

1: USB OTG HS saati etkinleştirildi

Bit 28 **ETHMACPTPEN**: Ethernet PTP saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet PTP saati devre dışı bırakıldı

1: Ethernet PTP saati etkinleştirildi

Bit 27 **ETHMACRXEN**: Ethernet Alım saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet Alım saati devre dışı bırakıldı

1: Ethernet Alım saati etkinleştirildi

Bit 26 **ETHMACTXEN**: Ethernet Gönderim saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet Gönderim saati devre dışı bırakıldı

1: Ethernet Gönderim saati etkinleştirildi

Bit 25 **ETHMACEN**: Ethernet MAC saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet MAC saati devre dışı bırakıldı

1: Ethernet MAC saati etkinleştirildi

Bitler 24:23Reserve, sıfır olarak tutulmalıdır.

Bit 22 **DMA2EN**: DMA2 saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA2 saati devre dışı bırakıldı

1: DMA2 saati etkinleştirildi

Bit 21 **DMA1EN**: DMA1 saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA1 saati devre dışı bırakıldı

1: DMA1 saati etkinleştirildi

Bit 20 **CCMDATARAMEN**: CCM veri RAM saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CCM veri RAM saati devre dışı bırakıldı

1: CCM veri RAM saati etkinleştirildi

Bit 19Rezerve, sıfır olarak tutulmalıdır.

Bit 18 **BKPSRAMEN**: Yedek SRAM arabirimi saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Yedek SRAM arabirimi saati devre dışı bırakıldı

1: Yedek SRAM arabirimi saati etkinleştirildi

Bitler 17:13Rezerve, sıfır olarak tutulmalıdır.

Bit 12 **CRCEN**: CRC saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CRC saati devre dışı bırakıldı

1: CRC saati etkinleştirildi

Bitler 11:9Rezerve, sıfır olarak tutulmalıdır.(read / write)

Bit 8 **GPIOIEN**: I/O port I saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port I saati devre dışı bırakıldı

1: I/O port I saati etkinleştirildi

Bit 7 GPIOHEN: I/O port H saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port Hsaati devre dışı bırakıldı

1: I/O port Hsaati etkinleştirildi

Bit 6 GPIOGEN: I/O port G saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port G saati devre dışı bırakıldı

1: I/O port G saati etkinleştirildi

Bit 5 GPIOFEN: I/O port F saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port F saati devre dışı bırakıldı

1: I/O port F saati etkinleştirildi

Bit 4 GPIOEEN: I/O port E saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port E saati devre dışı bırakıldı

1: I/O port E saati etkinleştirildi

Bit 3 GPIODEN: I/O port D saati etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I/O port D saati devre dışı bırakıldı

1: I/O port D saati etkinleştirildi

Bit 2 GPIOCEN: IO port C saatini etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port C saati devre dışı bırakıldı

1: IO port C saati etkinleştirildi

Bit 1 GPIOBEN: IO port B saatini etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port Bsaati devre dışı bırakıldı

1: IO port Bsaati etkinleştirildi

Bit 0 GPIOAEN: IO port A saatini etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port A saati devre dışı bırakıldı

1: IO port A saati etkinleştirildi

RCC AHB2 peripheral clock enable register (RCC_AHB2ENR)

Adres ofseti: 0x34

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OTGFS EN	RNG EN	HASH EN	CRYP EN	Reserved			DCMI EN
								rw	rw	rw	rw				rw

Bitler 31:1Rezerve, sıfır olarak tutulmalıdır.

Bit 0 **FSMCEN**: Esnek statik bellek denetleyici modülü saatini etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: FSMC modülü saati devre dışı bırakıldı

1: FSMC modülü saati etkinleştirildi

RCC APB1 peripheral clock enable register (RCC_APB1ENR)

Adres ofseti: 0x40

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	Reser- ved	CAN2 EN	CAN1 EN	Reser- ved	I2C3 EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART 3 EN	USART 2 EN	Reser- ved
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 29**DACEN**:DAC arayüzü saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: DACarayüzü saati devre dışı

1: DACarayüzü saati etkin

Bit 28 **PWREN**: Güç arayüzü saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: Güç arayüzü saati devre dışı

1: Güç arayüzü saati etkin

Bit 26 **CAN2EN**: CAN 2 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: CAN 2 saati devre dışı

1: CAN 2 saati etkin

Bit 25 **CAN1EN**: CAN 1 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: CAN 1 saati devre dışı

1: CAN 1 saati etkin

Bit 23 **I2C3EN**: I2C3 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: I2C3 saati devre dışı

1: I2C3 saati etkin

Bit 22 **I2C2EN**: I2C2 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: I2C2 saati devre dışı

1: I2C2 saati etkin

Bit 21 **I2C1EN**: I2C1 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: I2C1saati devre dışı

1: I2C1saati etkin

Bit 20 **UART5EN**: UART5 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: UART5 saati devre dışı

1: UART5 saati etkin

Bit 19 **UART4EN**: UART4 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: UART4 saati devre dışı

1: UART4 saati etkin

Bit 18 **USART3EN**: USART3 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: USART3saati devre dışı

1: USART3saati etkin

Bit 17 **USART2EN**: USART2 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: USART2saati devre dışı

1: USART2saati etkin

Bit 16 Rezerve , sıfır olarak sıfırlanmalıdır.

Bit 15 **SPI3EN**: SPI3saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: SPI3 saati devre dışı

1: SPI3 saati etkin

Bit 14 **SPI2EN**: SPI2 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: SPI2 saati devre dışı

1: SPI2 saati etkin

Bit 13:12Rezerve, sıfır olarak sıfırlanmalıdır.

Bit 11 **WWDGEN**: Pencere bekleyici saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: Pencere bekleyici saati devre dışı

1: Pencere bekleyici saati etkin

Bit 10:9Rezerve, sıfır olarak sıfırlanmalıdır.)

Bit 8 **TIM14EN**: TIM14 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM14 saati devre dışı

1: TIM14 saati etkin

Bit 7 **TIM13EN**: TIM13 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM13 saati devre dışı

1: TIM13 saati etkin

Bit 6 **TIM12EN**: TIM12 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM12 saati devre dışı

1: TIM12 saati etkin

Bit 5 **TIM7EN**: TIM7 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM7 saati devre dışı

1: TIM7 saati etkin

Bit 4 **TIM6EN**: TIM6 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM6 saati devre dışı

1: TIM6 saati etkin

Bit 3 **TIM5EN**: TIM5 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM5 saati devre dışı

1: TIM5 saati etkin

Bit 2 **TIM4EN**: TIM4 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM4 saati devre dışı

1: TIM4 saati etkin

Bit 1 **TIM3EN**: TIM3saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM3 saati devre dışı

1: TIM3 saati etkin

Bit 0 **TIM2EN**: TIM2 saati etkinleştirme(read /write)

Yazılım tarafından ayarlanır ve silinir.

0: TIM2 saati devre dışı

1: TIM2 saati etkin

RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Adres ofseti: 0x24

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 RST	TIM10 RST	TIM9 RST
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser- ved	SYSCF G RST	Reser- ved	SPI1 RST	SDIO RST	Reserved		ADC RST	Reserved		USART 6 RST	USART 1 RST	Reserved		TIM8 RST	TIM1 RST
	rw			rw			rw				rw			rw	

Bitler 31:19 Rezerve , sıfır olarak sıfırlanmalıdır.

Bit 18 **TIM11RST**: TIM11 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM11'i sıfırlamaz

1: TIM11'i sıfırlar

Bit 17 **TIM10RST**: TIM10 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM10'u sıfırlamaz

1: TIM10'u sıfırlar

Bit 16 **TIM9RST**: TIM9 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM9'u sıfırlamaz

1: TIM9'u sıfırlar

Bit 15 Rezerve, sıfır olarak sıfırlanmalıdır.

Bit 14 **SYSCFGRST**: Sistem yapılandırma denetleyicisi sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Sistem konfigürasyon denetleyicisini sıfırlamaz

1: Sistem konfigürasyon denetleyicisini sıfırlar

Bit 13 Rezerve, sıfır olarak sıfırlanmalıdır.

Bit 12 **SPI1RST**: SPI1 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI1'i sıfırlamaz

1: SPI1'i sıfırlar

Bit 11 **SDIORST**: SDIO sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SDIO modülünü sıfırlamaz

1: SDIO modülünü sıfırlar

Bitler 10:9Rezerve, sıfır olarak sıfırlanmalıdır.

Bit 8 **ADCRST**: ADC arayüzü sıfırlama (tüm ADC'ler için ortak)(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: ADC arayüzünü sıfırlamaz

1: ADC arayüzünü sıfırlar

Bitler 7:6Rezerve, sıfır değerinde tutulmalıdır.

Bit 5 **USART6RST**: USART6 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART6'yı sıfırlamaz

1: USART6'yı sıfırlar

Bit 4 **USART1RST**: USART1 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART1'i sıfırlamaz

1: USART1'i sıfırlar

Bitler 3:2Rezerve, sıfır değerinde tutulmalıdır.

Bit 1 **TIM8RST**: TIM8 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM8'i sıfırlamaz

1: TIM8'i sıfırlar

Bit 0 **TIM1RST**: TIM1 sıfırlama(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM1'i sıfırlamaz

1: TIM1'i sıfırlar

RCC AHB1 peripheral clock enable register (RCC_AHB1ENR)

Adres offseti: 0x30

Sıfırlama değeri: 0x0010 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	OTGHSULPIEN	OTGHSSEN	ETHMACPTPEN	ETHMACRXEN	ETHMACTXEN	ETHMACEN	Reserved			DMA2EN	DMA1EN	CCMDATARAMEN	Res.	BKPSRAMEN	Reserved
	rw	rw	rw	rw	rw	rw				rw	rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved			CRCEEN	Reserved			GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
			rw				rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31 Rezerve, sıfır değerinde tutulmalıdır.

Bit 30 **OTGHSULPIEN**: USB OTG HSULPI saatini etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir. Bu bit, OTG_HS'nin FS modunda kullanıldığında temizlenmelidir.

0: USB OTG HS ULPI saati devre dışı bırakıldı

1: USB OTG HS ULPI saati etkinleştirildi

Bit 29 **OTGHSSEN**: USB OTG HS saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG HS saati devre dışı bırakıldı

1: USB OTG HS saati etkinleştirildi

Bit 28 **ETHMACPTPEN**: Ethernet PTP saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet PTP saati devre dışı bırakıldı

1: Ethernet PTP saati etkinleştirildi

Bit 27 **ETHMACRXEN**: Ethernet Alım saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet Alım saati devre dışı bırakıldı

1: Ethernet Alım saati etkinleştirildi

Bit 26 **ETHMACTXEN**: Ethernet Gönderim saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet Gönderim saati devre dışı bırakıldı

1: Ethernet Gönderim saati etkinleştirildi

Bit 25 **ETHMACEN**: Ethernet MAC saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet MAC saati devre dışı bırakıldı

1: Ethernet MAC saati etkinleştirildi

Bitler 24:23Rezerve, sıfır değerinde tutulmalıdır.

Bit 22 **DMA2EN**: DMA2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA2 saati devre dışı bırakıldı

1: DMA2 saati etkinleştirildi

Bit 21 **DMA1EN**: DMA1 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA1 saati devre dışı bırakıldı

1: DMA1 saati etkinleştirildi

Bit 20 **CCMDATARAMEN**: CCM veri RAM saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CCM veri RAM saati devre dışı bırakıldı

1: CCM veri RAM saati etkinleştirildi

Bit 19 Rezerve, sıfır değerinde tutulmalıdır.

Bit 18 **BKPSRAMEN**: Yedek SRAM arabirimi saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Yedek SRAM arabirimi saati devre dışı bırakıldı

1: Yedek SRAM arabirimi saati etkinleştirildi

Bitler 17:13Rezerve, sıfır değerinde tutulmalıdır.

Bit 12 **CRCEN**: CRC saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CRC saati devre dışı bırakıldı

1: CRC saati etkinleştirildi

Bitler 11:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **GPIOIEN**: IO port I saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port I saati devre dışı bırakıldı

1: IO port I saati etkinleştirildi

Bit 7 **GPIOHEN**: IO port H saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port H saati devre dışı bırakıldı

1: IO port H saati etkinleştirildi

Bit 6 **GPIOGEN**: IO port G saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

- 0: IO port G saati devre dışı bırakıldı
- 1: IO port G saati etkinleştirildi

Bit 5 **GPIOFEN**: IO port F saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port F saati devre dışı bırakıldı
 - 1: IO port F saati etkinleştirildi

Bit 4 **GPIOEEN**: IO port E saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port E saati devre dışı bırakıldı
 - 1: IO port E saati etkinleştirildi

Bit 3 **GPIODEN**: IO port D saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port D saati devre dışı bırakıldı
 - 1: IO port D saati etkinleştirildi

Bit 2 **GPIOCEN**: IO port C saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port C saati devre dışı bırakıldı
 - 1: IO port C saati etkinleştirildi

Bit 1 **GPIOBEN**: IO port B saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port B saati devre dışı bırakıldı
 - 1: IO port B saati etkinleştirildi

Bit 0 **GPIOAEN**: IO port A saati etkinleştir(read /write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: IO port A saati devre dışı bırakıldı
 - 1: IO port A saati etkinleştirildi

RCC AHB2 peripheral clock enable register (RCC_AHB2ENR)

Adres ofseti: 0x34

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OTGFS EN	RNG EN	HASH EN	CRYP EN	Reserved			DCMI EN
								rw	rw	rw	rw				rw

Bitler 31:8 Rezerve, sıfır değerinde tutulmalıdır.

Bit 7 **OTGFSEN**: USB OTG FS saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG FS saati devre dışı bırakıldı

1: USB OTG FS saati etkinleştirildi

Bit 6 **RNGEN**: Rastgele sayı üretici saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Rastgele sayı üretici saati devre dışı bırakıldı

1: Rastgele sayı üretici saati etkinleştirildi

Bit 5 **HASHEN**: Hash modülleri saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Hash modülleri saati devre dışı bırakıldı

1: Hash modülleri saati etkinleştirildi

Bit 4 **CRYPEN**: Kriptografik modüller saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Kriptografik modül saati devre dışı bırakıldı

1: Kriptografik modül saati etkinleştirildi

Bitler 3:1 Rezerve, sıfır değerinde tutulmalıdır.

Bit 0 **DCMIEN**: Kamera arayüzü etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Kamera arayüzü saati devre dışı bırakıldı

1: Kamera arayüzü saati etkinleştirildi

RCC AHB3 peripheral clock enable register (RCC_AHB3ENR)

Adres ofseti: 0x38

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSMCEN	
														rw	

Bitler 31:1 Rezerve, sıfır değerinde tutulmalıdır.

Bit 0 **FSMCEN**: Esnek statik bellek denetleyici modülü saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: FSMC modül saati devre dışı bırakıldı

1: FSMC modül saati etkinleştirildi

RCC APB1 peripheral clock enable register (RCC_APB1ENR)

Adres ofseti: 0x40

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC EN	PWR EN	Reserved	CAN2 EN	CAN1 EN	Reserved	I2C3 EN	I2C2 EN	I2C1 EN	UART5 EN	UART4 EN	USART3 EN	USART2 EN	Reserved
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 EN	SPI2 EN	Reserved		WWDG EN	Reserved		TIM14 EN	TIM13 EN	TIM12 EN	TIM7 EN	TIM6 EN	TIM5 EN	TIM4 EN	TIM3 EN	TIM2 EN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

Bitler 31:30 Rezerve, sıfır değerinde tutulmalıdır.

Bit 29 **DACEN**: DAC arayüzü saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DAC arayüzü saati devre dışı bırakıldı

1: DAC arayüzü saati etkinleştirildi

Bit 28 **PWREN**: Güç arayüzü saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Güç arayüzü saati devre dışı bırakıldı

1: Güç arayüzü saati etkinleştirildi

Bit 27 Rezerve, sıfır değerinde tutulmalıdır

Bit 26 **CAN2EN**: CAN 2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CAN 2 saati devre dışı bırakıldı

1: CAN 2 saati etkinleştirildi

Bit 25 **CAN1EN**: CAN 1 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CAN 1 saati devre dışı bırakıldı

1: CAN 1 saati etkinleştirildi

Bit 24 Rezerve, sıfır değerinde tutulmalıdır.

Bit 23 **I2C3EN**: I2C3 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C3 saati devre dışı bırakıldı

1: I2C3 saati etkinleştirildi

Bit 22 **I2C2EN**: I2C2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C2 saati devre dışı bırakıldı

1: I2C2 saati etkinleştirildi

Bit 21 **I2C1EN**: I2C1 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: I2C1 saati devre dışı bırakıldı

1: I2C1 saati etkinleştirildi

Bit 20 **UART5EN**: UART5 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: UART5 saati devre dışı bırakıldı

1: UART5 saati etkinleştirildi

Bit 19 **UART4EN**: UART4 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: UART4 saati devre dışı bırakıldı

1: UART4 saati etkinleştirildi

Bit 18 **USART3EN**: USART3 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART3 saati devre dışı bırakıldı

1: USART3 saati etkinleştirildi

Bit 17 **USART2EN**: USART2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART2 saati devre dışı bırakıldı

1: USART2 saati etkinleştirildi

Bit 16 Rezerve, sıfır değerinde tutulmalıdır.

Bit 15 **SPI3EN**: SPI3 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI3 saati devre dışı bırakıldı

1: SPI3 saati etkinleştirildi

Bit 14 **SPI2EN**: SPI2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI2 saati devre dışı bırakıldı

1: SPI2 saati etkinleştirildi

Bitler 13:12Rezerve, sıfır değerinde tutulmalıdır.

Bit 11 **WWDGEN**: Pencere bekçisi saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Pencere bekçisi saati devre dışı bırakıldı

1: Pencere bekçisi saati etkinleştirildi

Bitler 10:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **TIM14EN**: TIM14 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM14 saati devre dışı bırakıldı

1: TIM14 saati etkinleştirildi

Bit 7 **TIM13EN**: TIM13 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM13 saati devre dışı bırakıldı

1: TIM13 saati etkinleştirildi

Bit 6 **TIM12EN**: TIM12 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM12 saati devre dışı bırakıldı

1: TIM12 saati etkinleştirildi

Bit 5 **TIM7EN**: TIM7 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM7 saati devre dışı bırakıldı

1: TIM7 saati etkinleştirildi

Bit 4 **TIM6EN**: TIM6 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM6 saati devre dışı bırakıldı

1: TIM6 saati etkinleştirildi

Bit 3 **TIM5EN**: TIM5 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM5 saati devre dışı bırakıldı

1: TIM5 saati etkinleştirildi

Bit 2 **TIM4EN**: TIM4 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM4 saati devre dışı bırakıldı

1: TIM4 saati etkinleştirildi

Bit 1 **TIM3EN**: TIM3 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM3 saati devre dışı bırakıldı

1: TIM3 saati etkinleştirildi

Bit 0 **TIM2EN**: TIM2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM2 saati devre dışı bırakıldı

1: TIM2 saati etkinleştirildi

RCC APB2 peripheral clock enable register (RCC_APB2ENR)

Adres ofseti: 0x44

Sıfırlama değeri: 0x0000 0000

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 EN	TIM10 EN	TIM9 EN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reser- ved	SYSCF G EN	Reser- ved	SPI1 EN	SDIO EN	ADC3 EN	ADC2 EN	ADC1 EN	Reserved			USART 6 EN	USART 1 EN	Reserved		TIM8 EN
	rw		rw	rw	rw	rw	rw				rw	rw			rw
															TIM1 EN
															rw

Bitler 31:19Rezerve, sıfır değerinde tutulmalıdır.

Bit 18 **TIM11EN**: TIM11 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM11 saati devre dışı bırakıldı

1: TIM11 saati etkinleştirildi

Bit 17 **TIM10EN**: TIM10 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM10 saati devre dışı bırakıldı

1: TIM10 saati etkinleştirildi

Bit 16 **TIM9EN**: TIM9 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM9 saati devre dışı bırakıldı

1: TIM9 saati etkinleştirildi

Bit 15 Rezerve, sıfır değerinde tutulmalıdır

Bit 14 **SYSCFGEN**: Sistem yapılandırma denetleyici saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Sistem yapılandırma denetleyici saati devre dışı bırakıldı

1: Sistem yapılandırma denetleyici saati etkinleştirildi

Bit 13 Rezerve, sıfır değerinde tutulmalıdır.

Bit 12 **SPI1EN**: SPI1 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SPI1 saati devre dışı bırakıldı

1: SPI1 saati etkinleştirildi

Bit 11 **SDIOEN**: SDIO saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SDIO modül saati devre dışı bırakıldı

1: SDIO modül saati etkinleştirildi

Bit 10 **ADC3EN**: ADC3 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: ADC3 saati devre dışı bırakıldı

1: ADC3 saati devre dışı bırakıldı

Bit 9 **ADC2EN**: ADC2 saati etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: ADC2 saati devre dışı bırakıldı

1: ADC2 saati devre dışı bırakıldı

Bit 8 **ADC1EN**: ADC1 saati etkinleştir (**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: ADC1 saati devre dışı bırakıldı

1: ADC1 saati devre dışı bırakıldı

Bitler 7:6Rezerve, sıfır değerinde tutulmalıdır.

Bit 5 **USART6EN**: USART6 saati etkinleştir (**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART6 saati devre dışı bırakıldı

1: USART6 saati etkinleştirildi

Bit 4 **USART1EN**: USART1 saati etkinleştir (**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: USART1 saati devre dışı bırakıldı

1: USART1 saati etkinleştirildi

Bitler 3:2Rezerve, sıfır değerinde tutulmalıdır.

Bit 1 **TIM8EN**: TIM8 saati etkinleştir (**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM8 saati devre dışı bırakıldı

1: TIM8 saati etkinleştirildi

Bit 0 **TIM1EN**: TIM1 saati etkinleştir (**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: TIM1 saati devre dışı bırakıldı

1: TIM1 saati etkinleştirildi.

RCC AHB1 peripheral clock enable in lowpower mode register (RCC_AHB1LPENR)

Adres ofseti: 0x50

Sıfırlama değeri: 0x7E67 91FF

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reser- ved	OTGHS ULPILPE N	OTGH S LPEN	ETHPT P LPEN	ETHRX LPEN	ETHTX LPEN	ETHMA C LPEN	Reserved		DMA2 LPEN	DMA1 LPEN	Reserved		BKPSRA M LPEN	SRAM 2 LPEN	SRAM 1 LPEN
	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLITF LPEN	Reserved		CRC LPEN	Reserved			GPIOI LPEN	GPIOH LPEN	GPIOG LPEN	GPIO F LPEN	GPIOE LPEN	GPIOD LPEN	GPIOC LPEN	GPIOB LPEN	GPIOA LPEN
rw			rw				rw	rw	rw	rw	rw	rw	rw	rw	

Bit 31 Rezerve, sıfır değerinde tutulmalıdır.

Bit 30 **OTGHSULPILPEN**: USB OTG HS ULPI saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir. Bu bit, OTG_HS'nin FS modunda kullanıldığında temizlenmelidir.

0: USB OTG HS ULPI saati Uyku modunda devre dışı bırakıldı

1: USB OTG HS ULPI saati Uyku modunda etkinleştirildi

Bit 29 **OTGHS LPEN**: USB OTG HS saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: USB OTG HS saati Uyku modunda devre dışı bırakıldı

1: USB OTG HS saati Uyku modunda etkinleştirildi

Bit 28 **ETHMACPTLPEN**: Ethernet PTP saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet PTP saati Uyku modunda devre dışı bırakıldı

1: Ethernet PTP saati Uyku modunda etkinleştirildi

Bit 27 **ETHMACRXLPEN**: Ethernet alım saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet alım saati Uyku modunda devre dışı bırakıldı

1: Ethernet alım saati Uyku modunda etkinleştirildi

Bit 26 **ETHMACTXLPEN**: Ethernet iletim saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet iletim saati Uyku modunda devre dışı bırakıldı

1: Ethernet iletim saati Uyku modunda etkinleştirildi

Bit 25 **ETHMACLPEN**: Ethernet MAC saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Ethernet MAC saati Uyku modunda devre dışı bırakıldı

1: Ethernet MAC saati Uyku modunda etkinleştirildi

Bitler 24:23Rezerve, sıfır değerinde tutulmalıdır.

Bit 22 **DMA2LPEN**: DMA2 saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA2 saati Uyku modunda devre dışı bırakıldı

1: DMA2 saati Uyku modunda etkinleştirildi

Bit 21 **DMA1LPEN**: DMA1 saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: DMA1 saati Uyku modunda devre dışı bırakıldı

1: DMA1 saati Uyku modunda etkinleştirildi

Bitler 20:19 Ayrılmış, sıfır sıfır değerinde tutulmalıdır.

Bit 18 **BKPSRAMLPEN**: Yedek SRAM arayüz saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Yedek SRAM arayüz saati Uyku modunda devre dışı bırakıldı

1: Yedek SRAM arayüz saati Uyku modunda etkinleştirildi

Bit 17 **SRAM2LPEN**: SRAM 2 arayüz saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SRAM 2 arayüz saati Uyku modunda devre dışı bırakıldı

1: SRAM 2 arayüz saati Uyku modunda etkinleştirildi

Bit 16 **SRAM1LPEN**: SRAM 1 arayüz saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: SRAM 1 arayüz saati Uyku modunda devre dışı bırakıldı

1: SRAM 1 arayüz saati Uyku modunda etkinleştirildi

Bit 15 **FLITFLPEN**: Flash arayüz saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Flash arayüz saati Uyku modunda devre dışı bırakıldı

1: Flash arayüz saati Uyku modunda etkinleştirildi

Bitler 14:13Rezerve, sıfır değerinde tutulmalıdır.

Bit 12 **CRCLPEN**: CRC saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: CRC saati Uyku modunda devre dışı bırakıldı

1: CRC saati Uyku modunda etkinleştirildi

Bitler 11:9Rezerve, sıfır değerinde tutulmalıdır.

Bit 8 **GPIOLPEN**: IO port I saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port I saati Uyku modunda devre dışı bırakıldı

1: IO port I saati Uyku modunda etkinleştirildi

Bit 7 **GPIOHLPEN**: IO port H saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port H saati Uyku modunda devre dışı bırakıldı

1: IO port H saati Uyku modunda etkinleştirildi

Bitler 6 **GPIOGLPEN**: IO port G saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port G saati Uyku modunda devre dışı bırakıldı

1: IO port G saati Uyku modunda etkinleştirildi

Bit 5 **GPIOFLPEN**: IO port F saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port F saati Uyku modunda devre dışı bırakıldı

1: IO port F saati Uyku modunda etkinleştirildi

Bit 4 **GPIOELPEN**: IO port E saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port E saati Uyku modunda devre dışı bırakıldı

1: IO port E saati Uyku modunda etkinleştirildi

Bit 3 **GIPIODLPEN**: IO port D saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port D saati Uyku modunda devre dışı bırakıldı

1: IO port D saati Uyku modunda etkinleştirildi

Bit 2 **GPIOCLPEN**: IO port C saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port C saati Uyku modunda devre dışı bırakıldı

1: IO port C saati Uyku modunda etkinleştirildi

Bit 1 **GPIOBLPEN**: IO port B saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port B saati Uyku modunda devre dışı bırakıldı

1: IO port B saati Uyku modunda etkinleştirildi

Bit 0 **GPIOALPEN**: IO port A saati Uyku modunda etkinleştir (read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: IO port A saati Uyku modunda devre dışı bırakıldı

1: IO port A saati Uyku modunda etkinleştirildi

RCC AHB2 peripheral clock enable in lowpower mode register (RCC_AHB2LPENR)

Adres ofseti: 0x54

Sıfırlama değeri: 0x0000 00F1

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved								OTGFS LPEN	RNG LPEN	HASH LPEN	CRYP LPEN	Reserved			DCMI LPEN
								rw	rw	rw	rw				rw

Bitler 31:8Rezerve, sıfır değerinde tutulmalıdır.

Bit 7 **OTGFSLPEN**: Uyku modunda USB OTG FS saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda USB OTG FS saati devre dışı bırakıldı

1: Uyku modunda USB OTG FS saati etkinleştirildi

Bit 6 **RNGLPEN**: Uyku modunda Rastgele Sayı Üretici saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Rastgele Sayı Üretici saati devre dışı bırakıldı

1: Uyku modunda Rastgele Sayı Üretici saati etkinleştirildi

Bit 5 **HASHLPEN**: Uyku modunda Hash modülleri saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Hash modülleri saati devre dışı bırakıldı

1: Uyku modunda Hash modülleri saati etkinleştirildi

Bit 4 **CRYPLPEN**: Uyku modunda Şifreleme modülleri saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Şifreleme modülleri saati devre dışı bırakıldı

1: Uyku modunda Şifreleme modülleri saati etkinleştirildi

Bitler 3:1Rezerve, sıfır değeri tutulmalıdır.

Bit 0 **DCMILPEN**: Uyku modunda Kamera arabirimi etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Kamera arabirimi saati devre dışı bırakıldı

1: Uyku modunda Kamera arabirimi saati etkinleştirildi

RCC AHB3 peripheral clock enable in lowpower mode register (RCC_AHB3LPENR)

Adres ofseti: 0x58

Sıfır değeri: 0x0000 0001

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														FSMC LPEN	
															rw

Bits31:1Rezerve, sıfır değerinde tutulmalıdır.

Bit 0 **FSMCLPEN**: Uyku modunda Esnek statik bellek denetleyici modülü saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda FSMC modül saati devre dışı bırakıldı

1: Uyku modunda FSMC modül saati etkinleştirildi

RCC APB1 peripheral clock enable in lowpower mode register (RCC_APB1LPENR)

Adres ofseti: 0x60

Sıfır değeri: 0x36FE C9FF

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved		DAC LPEN	PWR LPEN	RESERVED	CAN2 LPEN	CAN1 LPEN	Reserved	I2C3 LPEN	I2C2 LPEN	I2C1 LPEN	UART5 LPEN	UART4 LPEN	USART3 LPEN	USART2 LPEN	Reserved
		rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3 LPEN	SPI2 LPEN	Reserved		WWDG LPEN	Reserved		TIM14 LPEN	TIM13 LPEN	TIM12 LPEN	TIM7 LPEN	TIM6 LPEN	TIM5 LPEN	TIM4 LPEN	TIM3 LPEN	TIM2 LPEN
rw	rw			rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

Bitler 31:30: Rezerve, sıfır değerinde tutulmalıdır.

Bit 29: **DACL PEN**: Uyku modunda DAC arayüzü saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda DAC arayüzü saati devre dışı bırakıldı

1: Uyku modunda DAC arayüzü saati etkinleştirildi

Bit 28: **PWRL PEN**: Uyku modunda Güç arayüzü saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Güç arayüzü saati devre dışı bırakıldı

1: Uyku modunda Güç arayüzü saati etkinleştirildi

Bit 27: Rezerve, sıfır değerinde tutulmalıdır.

Bit 26: **CAN2LPEN**: Uyku modunda CAN 2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda CAN 2 saati devre dışı bırakıldı

1: Uyku modunda CAN 2 saati etkinleştirildi

Bit 25: **CAN1LPEN**: Uyku modunda CAN 1 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda CAN 1 saati devre dışı bırakıldı

1: Uyku modunda CAN 1 saati etkinleştirildi

Bit 24: Rezerve, sıfır değerinde tutulmalıdır.

Bit 23: **I2C3LPEN**: Uyku modunda I2C3 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda I2C3 saati devre dışı bırakıldı

1: Uyku modunda I2C3 saati etkinleştirildi

Bit 22: **I2C2LPEN**: Uyku modunda I2C2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda I2C2 saati devre dışı bırakıldı

1: Uyku modunda I2C2 saati etkinleştirildi

Bit 21: **I2C1LPEN**: Uyku modunda I2C1 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda I2C1 saati devre dışı bırakıldı

1: Uyku modunda I2C1 saati etkinleştirildi

Bit 20: **UART5LPEN**: Uyku modunda UART5 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda UART5 saati devre dışı bırakıldı

1: Uyku modunda UART5 saati etkinleştirildi

Bit 19: **UART4LPEN**: Uyku modunda UART4 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda UART4 saati devre dışı bırakıldı

1: Uyku modunda UART4 saati etkinleştirildi

Bit 18: **USART3LPEN**: Uyku modunda USART3 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda USART3 saati devre dışı bırakıldı

1: Uyku modunda USART3 saati etkinleştirildi

Bit 17: **USART2LPEN**: Uyku modunda USART2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda USART2 saati devre dışı bırakıldı

1: Uyku modunda USART2 saati etkinleştirildi

Bit 16: Rezerve, sıfır değerinde tutulmalıdır.

Bit 15: **SPI3LPEN**: Uyku modunda SPI3 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda SPI3 saati devre dışı bırakıldı

1: Uyku modunda SPI3 saati etkinleştirildi

Bit 14: **SPI2LPEN**: Uyku modunda SPI2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda SPI2 saati devre dışı bırakıldı

1: Uyku modunda SPI2 saati etkinleştirildi

Bitler 13:12: Rezerve, sıfır değerinde tutulmalıdır.

Bit 11: **WWDGLPEN**: Uyku modunda Pencere bekçisi saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Pencere bekçisi saati devre dışı bırakıldı

1: Uyku modunda Pencere bekçisi saati etkinleştirildi

Bitler 10:9: Rezerve, sıfır değerinde tutulmalıdır.

Bit 8: **TIM14LPEN**: Uyku modunda TIM14 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM14 saati devre dışı bırakıldı

1: Uyku modunda TIM14 saati etkinleştirildi

Bit 7: **TIM13LPEN**: Uyku modunda TIM13 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM13 saati devre dışı bırakıldı

1: Uyku modunda TIM13 saati etkinleştirildi

Bit 6: **TIM12LPEN**: Uyku modunda TIM12 saati etkinleştir

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM12 saati devre dışı bırakıldı

1: Uyku modunda TIM12 saati etkinleştirildi

Bit 5: **TIM7LPEN**: Uyku modunda TIM7 saati etkinleştir

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM7 saati devre dışı bırakıldı

1: Uyku modunda TIM7 saati etkinleştirildi

Bit 4: **TIM6LPEN**: Uyku modunda TIM6 saati etkinleştir

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM6 saati devre dışı bırakıldı

1: Uyku modunda TIM6 saati etkinleştirildi

Bit 3: **TIM5LPEN**: Uyku modunda TIM5 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM5 saati devre dışı bırakıldı

1: Uyku modunda TIM5 saati etkinleştirildi

Bit 2: **TIM4LPEN**: Uyku modunda TIM4 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM4 saati devre dışı bırakıldı

1: Uyku modunda TIM4 saati etkinleştirildi

Bit 1: **TIM3LPEN**: Uyku modunda TIM3 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM3 saati devre dışı bırakıldı

1: Uyku modunda TIM3 saati etkinleştirildi

Bit 0: **TIM2LPEN**: Uyku modunda TIM2 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM2 saati devre dışı bırakıldı

1: Uyku modunda TIM2 saati etkinleştirildi

RCC APB2 peripheral clock enabled in lowpower mode

register (RCC_APB2LPENR)

Adres offseti: 0x64

Sıfır değeri: 0x0007 5F33

Erişim: Word, half-word ve byte erişimi bekleme durumu olmadan sağlanır.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved													TIM11 LPEN	TIM10 LPEN	TIM9 LPEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	SYSC FG LPEN	Reserved	SPI1 LPEN	SDIO LPEN	ADC3 LPEN	ADC2 LPEN	ADC1 LPEN	Reserved		USART 6 LPEN	USART 1 LPEN	Reserved	TIM8 LPEN	TIM1 LPEN	
	rw		rw	rw	rw	rw	rw			rw	rw		rw		

Bitler 31:19: Rezerve, sıfır değerinde tutulmalıdır.

Bit 18: **TIM11LPEN**: Uyku modunda TIM11 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM11 saati devre dışı bırakıldı

1: Uyku modunda TIM1 1 saati etkinleştirildi

Bit 17: **TIM10LPEN**: Uyku modunda TIM10 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM10 saati devre dışı bırakıldı

1: Uyku modunda TIM10 saati etkinleştirildi

Bit 16: **TIM9LPEN**: Uyku modunda TIM9 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM9 saati devre dışı bırakıldı

1: Uyku modunda TIM9 saati etkinleştirildi

Bit 15: Rezerve, sıfır değerinde tutulmalıdır.

Bit 14: **SYSCFGLPEN**: Uyku modunda Sistem konfigürasyon denetleyici saati etkinleştir(read/write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda Sistem konfigürasyon denetleyici saati devre dışı bırakıldı

1: Uyku modunda Sistem konfigürasyon denetleyici saati etkinleştirildi

Bit 13: Rezerve, sıfır değerinde tutulmalıdır.

Bit 12: **SPI1LPEN**: Uyku modunda SPI1 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda SPI1 saati devre dışı bırakıldı

1: Uyku modunda SPI1 saati etkinleştirildi

Bit 11: **SDIOLPEN**: Uyku modunda SDIO saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda SDIO modül saati devre dışı bırakıldı

1: Uyku modunda SDIO modül saati etkinleştirildi

Bit 10: **ADC3LPEN**: Uyku modunda ADC 3 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda ADC 3 saati devre dışı bırakıldı

1: Uyku modunda ADC 3 saati devre dışı bırakıldı

Bit 9: **ADC2LPEN**: Uyku modunda ADC2 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda ADC2 saati devre dışı bırakıldı

1: Uyku modunda ADC2 saati devre dışı bırakıldı

Bit 8: **ADC1LPEN**: Uyku modunda ADC1 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda ADC1 saati devre dışı bırakıldı

1: Uyku modunda ADC1 saati devre dışı bırakıldı

Bitler 7:6: Rezerve, sıfır değerinde tutulmalıdır.

Bit 5: **USART6LPEN**: Uyku modunda USART6 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda USART6 saati devre dışı bırakıldı

1: Uyku modunda USART6 saati etkinleştirildi

Bit 4: **USART1LPEN**: Uyku modunda USART1 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda USART1 saati devre dışı bırakıldı

1: Uyku modunda USART1 saati etkinleştirildi

Bitler 3:2: Rezerve, sıfır değerinde tutulmalıdır.

Bit 1: **TIM8LPEN**: Uyku modunda TIM8 saati etkinleştir(**read /write**)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM8 saati devre dışı bırakıldı

1: Uyku modunda TIM8 saati etkinleştirildi

Bit 0: TIM1LPEN: Uyku modunda TIM1 saati etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Uyku modunda TIM1 saati devre dışı bırakıldı

1: Uyku modunda TIM1 saati etkinleştirildi

RCC Backup domain control register (RCC_BDCR)

Adres ofseti: 0x70

Sıfır değeri: 0x0000 0000, Yedek etki alanı sıfırlama tarafından sıfırlanır.

Erişim: $0 \leq$ bekleme süresi ≤ 3 , Word, half-word ve bayt erişimi

Bu kayda ardışık erişimlerde bekleme süreleri eklenir.

RCC Yedek etki alanı kontrol kaydındaki (RCC_BDCR) LSEON, LSEBYP, RTCSEL ve RTCEN bitleri Yedek etki alanındadır. Sonuç olarak, Sıfırlama sonrasında, bu bitler yazmaya karşı korunur ve STM32F405xx/07xx ve STM32F415xx/17xx için güç kontrol kaydındaki (PWR_CR) DBP bitinin bu bitler değiştirilmeden önce ayarlanması gerekmektedir. Daha fazla bilgi için 215. sayfadaki 7.1.1 Bölümü: Sistem sıfırlamasına bakınız. Bu bitler yalnızca Bir Yedek etki alanı Sıfırlaması sonrasında sıfırlanır (bkz. 7.1.3 Bölümü: Yedek etki alanı sıfırlaması). Herhangi bir dahili veya harici Sıfırlama bu bitler üzerinde etkisi yoktur.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															BDRST
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Reserved					RTCSEL[1:0]		Reserved					LSEBYP	LSERDY	LSEON
rw						rw	rw						rw	r	rw

Bitler 31:17: Rezerve, sıfır değerinde tutulmalıdır.

Bit 16: **BDRST**: Yedek etki alanı yazılım sıfırlaması(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Sıfırlama etkinleştirilmemiş

1: Tüm Yedek etki alanını sıfırlar

Not: BKPSRAM bu sıfırlamadan etkilenmez, BKPSRAM'ın sıfırlanmasının tek yolu, düzey 1'den düzey 0'a bir koruma seviyesi değişikliği talep edildiğinde Flash arabirimi aracılığıyla gerçekleşir.

Yedek etki alanı yazılım sıfırlaması, PWR_CR kaydının DBP biti ayarlanana kadar etkili olmaz.

Bit 15: **RTCEN**: RTC saatini etkinleştir(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: RTC saati devre dışı

1: RTC saati etkin

Bitler 14:10: Rezerve, sıfır değerinde tutulmalıdır.

Bitler 9:8: **RTCSEL**[1:0]: RTC saat kaynağı seçimi(read /write)

RTC için saat kaynağını seçmek için yazılım tarafından ayarlanır. RTC saat kaynağı seçildikten sonra, Yedek etki alanı sıfırlanmadıkça artık değiştirilemez. BDRST biti bunları sıfırlamak için kullanılabilir.

00: Saat yok

01: RTC saati olarak LSE osilatör saati kullanılır

10: RTC saati olarak LSI osilatör saati kullanılır

11: RTC saati olarak bir programlanabilir ön bölücü ile (RCC_CFGR'deki RTCPRE[4:0] bitleri aracılığıyla seçim) bölünmüş HSE osilatör saati kullanılır

Bitler 7:3: Rezerve, sıfır değerinde tutulmalıdır.

Bit 2: **LSEBYP**: Harici düşük hız osilatörü bypass(read /write)

Osilatörü bypass etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bit yalnızca LSE saat devre dışıyken yazılabilir.

0: LSE osilatörü bypass edilmedi

1: LSE osilatörü bypass edildi

Bit 1: **LSERDY**: Harici düşük hız osilatörü hazır(read)

Harici 32 kHz osilatörün stabil olduğunu göstermek için donanım tarafından ayarlanır ve temizlenir. LSEON biti temizlendikten sonra, LSERDY 6 harici düşük hız osilatörü saat döngüsü sonrasında düşük seviyeye geçer.

0: LSE saat hazır değil

1: LSE saat hazır

Bit 0: **LSEON**: Harici düşük hız osilatörü etkin(read /write)

Yazılım tarafından ayarlanır ve temizlenir.

0: LSE saat KAPALI

1: LSE saat AÇIK

RCC clock control &status register (RCC_CSR)

Adres ofseti: 0x74

Sıfır değeri: 0x0E00 0000, sadece güç sıfırlaması tarafından güç sıfırlama bayrakları hariç olmak üzere sistem sıfırlaması ile sıfırlanır.

Erişim: $0 \leq \text{bekleme süresi} \leq 3$, Word, half-wordve bayt erişimi

Bu kayda ardışık erişimlerde bekleme süreleri eklenir.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPWR RSTF	WWDG RSTF	IWDG RSTF	SFT RSTF	POR RSTF	PIN RSTF	BORRS TF	RMVF	Reserved							
r	r	r	r	r	r	r	rt_w								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved													LSIRDY	LSION	
													r	rw	

Bit 31 **LPWRRSTF**: Düşük güç sıfırlama bayrağı(read)

Düşük güç yönetimi sıfırlaması gerçekleştiğinde donanım tarafından ayarlanır.
RMVF bitine yazılarak temizlenir.

0: Düşük güç yönetimi sıfırlaması gerçekleşmedi

1: Düşük güç yönetimi sıfırlaması gerçekleşti

Düşük güç yönetimi sıfırlaması hakkında daha fazla bilgi için Düşük güç yönetimi sıfırlamasına bakınız.

Bit 30 **WWDGRSTF**: Pencere bekçisi sıfırlama bayrağı(read)

Pencere bekçisi sıfırlaması gerçekleştiğinde donanım tarafından ayarlanır.
RMVF bitine yazılarak temizlenir.

0: Pencere bekçisi sıfırlaması gerçekleşmedi

1: Pencere bekçisi sıfırlaması gerçekleşti

Bit 29 **IWDGRSTF**: Bağımsız bekçi sıfırlama bayrağı(read)

Bağımsız bekçi sıfırlaması VDD alanından gerçekleştiğinde donanım tarafından ayarlanır.

RMVF bitine yazılarak temizlenir.

0: Bekçi sıfırlaması gerçekleşmedi

1: Bekçi sıfırlaması gerçekleşti

Bit 28 **SFTRSTF**: Yazılım sıfırlama bayrağı(read)

Yazılım sıfırlaması gerçekleştiğinde donanım tarafından ayarlanır.

RMVF bitine yazılarak temizlenir.

0: Yazılım sıfırlaması gerçekleşmedi

1: Yazılım sıfırlaması gerçekleşti

Bit 27 **PORRSTF**: POR/PDR sıfırlama bayrağı(read)

POR/PDR sıfırlaması gerçekleştiğinde donanım tarafından ayarlanır.

RMVF bitine yazılarak temizlenir.

0: POR/PDR sıfırlaması gerçekleşmedi

1: POR/PDR sıfırlaması gerçekleşti

Bit 26 **PINRSTF**: PIN sıfırlama bayrağı(read)

NRST piminden sıfırlama gerçekleştiğinde donanım tarafından ayarlanır.

RMVF bitine yazılarak temizlenir.

- 0: NRST piminden sıfırlama gerçekleşmedi
- 1: NRST piminden sıfırlama gerçekleşti

Bit 25 **BORRSTF**: BOR sıfırlama bayrağı(read)

- POR/PDR veya BOR sıfırlaması gerçekleştiğinde donanım tarafından ayarlanır.
RMVF bitine yazılarak temizlenir.
- 0: POR/PDR veya BOR sıfırlaması gerçekleşmedi
 - 1: POR/PDR veya BOR sıfırlaması gerçekleşti

Bit 24 **RMVF**: Sıfırlama bayrağını kaldır(read / write)

- Sıfırlama bayraklarını temizlemek için yazılım tarafından ayarlanır.
- 0: Hiçbir etkisi yok
 - 1: Sıfırlama bayraklarını temizler

Bitler 23:2: Rezerve, sıfır değerinde tutulmalıdır.

Bit 1 **LSIRDY**: Dahili düşük hız osilatörü hazır(read)

- Dahili RC 40 kHz osilatörünün kararlı olduğunu göstermek için donanım tarafından ayarlanır ve temizlenir.
LSION biti temizlendikten sonra, LSIRDY 3 LSI saat döngüsü sonrasında düşük seviyeye geçer.
- 0: LSI RC osilatörü hazır değil
 - 1: LSI RC osilatörü hazır

Bit 0 **LSION**: Dâhili düşük hız osilatörü etkin(read / write)

- Yazılım tarafından ayarlanır ve temizlenir.
- 0: LSI RC osilatörü KAPALI
 - 1: LSI RC osilatörü AÇIK

RCC spread spectrum clock generation register (RCC_SSCGR)

Adres ofseti: 0x80

Sıfır değeri: 0x0000 0000

Erişim: bekleme süresi yok, Word, half-word ve bayt erişimi.
Yayılmış spektrum saat üretimi yalnızca ana PLL için kullanılabilir.
RCC_SSCGR kaydı, ana PLL etkinleştirilmeden önce veya ana PLL devre dışı bırakıldıktan sonra yazılmalıdır.

Not: PLL yayılmış spektrum saat üretimi (SSCG) özellikleri hakkında tam detaylar için cihaz veri sayfasındaki "Elektriksel özellikler" bölümüne başvurun.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SSCG EN	SPR EAD SEL	Reserved		INCSTEP											
rw	rw			rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INCSTEP				MODPER											
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit 31: **SSCGEN**: Yayılmış spektrum modülasyonunu etkinleştir(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

0: Yayılmış spektrum modülasyonu KAPALI. (CR[24]=PLLON biti temizlendikten sonra yazılmalıdır)

1: Yayılmış spektrum modülasyonu AÇIK. (CR[24]=PLLON biti ayarlandıktan önce yazılmalıdır)

Bit 30: **SPREADSEL**: Yayılım Seçimi(read / write)

Yazılım tarafından ayarlanır ve temizlenir.

CR[24]=PLLON biti ayarlanmadan önce yazılmalıdır.

0: Merkez yayılım

1: Aşağı yayılım

Bitler 29:28: Rezerve, sıfır değerinde tutulmalıdır.

Bitler 27:13: **INCSTEP**: Artış adımı(read / write)

Yazılım tarafından ayarlanır ve temizlenir. CR[24]=PLLON biti ayarlanmadan önce yazılmalıdır.

Modülasyon profili genliği için yapılandırma girişidir.

Bitler 12:0: **MODPER**: Modülasyon periyodu(read / write)

Yazılım tarafından ayarlanır ve temizlenir. CR[24]=PLLON biti ayarlanmadan önce yazılmalıdır.

Modülasyon profili periyodu için yapılandırma girişidir.

RCC PLLI2S configuration register (RCC_PLLI2SCFGR)

Adres ofseti: 0x84

Sıfır değeri: 0x2000 3000

Erişim: bekleme süresi yok, Word, half-word ve bayt erişimi.

Bu kayıt, PLLI2S saat çıkışlarını aşağıdaki formüllere göre yapılandırmak için kullanılır:

- $f_{(VCO \text{ clock})} = f_{(PLL I2S \text{ clock input})} \times (PLL I2S N / PLL M)$
- $f_{(PLL I2S \text{ clock output})} = f_{(VCO \text{ clock})} / PLL I2S R$

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved	PLL I2S R2	PLL I2S R1	PLL I2S R0	Reserved											
	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved	PLL I2S N 8	PLL I2S N 7	PLL I2S N 6	PLL I2S N 5	PLL I2S N 4	PLL I2S N 3	PLL I2S N 2	PLL I2S N 1	PLL I2S N 0	Reserved					
	rw	rw	rw	rw	rw	rw	rw	rw	rw						

Bit 31: Rezerve, sıfır değerinde tutulmalıdır.

Bitler 30:28: PLLI2SR: I2S saatleri için PLLI2S bölüm faktörü(read / write)

I2S saat frekansını kontrol etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bitler sadece PLLI2S devre dışı olduğunda yazılmalıdır. Faktör, standart kristaller kullanılırken %0.3 hata ve ses kristalleri kullanılırken %0 hata elde etmek için I2S aygıtlarındaki önbellekleyici değerlerle uyumlu olarak seçilmelidir. I2S saat frekansı ve hassasiyeti hakkında daha fazla bilgi için I2S bölümündeki "28.4.4: Clock generator" bölümüne başvurun.

Dikkat: I2S'lerin doğru şekilde çalışması için frekansın 192 MHz veya daha düşük olması gerekir.

I2S saat frekansı = VCO frekansı / PLLR, burada $2 \leq PLLR \leq 7$

000: PLLR = 0, yanlış yapılandırma

001: PLLR = 1, yanlış yapılandırma

010: PLLR = 2

...

111: PLLR = 7

Bitler 27:15: Rezerve, sıfır değerinde tutulmalıdır.

Bitler 14:6: PLLI2SN: VCO için PLLI2S çarpan faktörü(read / write)

Bu bitler, VCO'nun çarpan faktörünü kontrol etmek için yazılım tarafından ayarlanır ve temizlenir. Bu bitler sadece PLLI2S devre dışı olduğunda yazılabilir. Bu bitleri yazmak için yalnızca yarı-kelime ve kelime erişimleri kullanılabilir.

Dikkat: Yazılımın, VCO çıkış frekansının 100 ila 432 MHz arasında olduğundan emin olmak için bu bitleri doğru şekilde ayarlaması gerekir.

VCO çıkış frekansı = VCO giriş frekansı $\times$ PLLI2SN, burada $50 \leq PLLI2SN \leq 432$

000000000: PLLI2SN = 0, yanlış yapılandırma

000000001: PLLI2SN = 1, yanlış yapılandırma

...

000110010: PLLI2SN = 50

...

001100011: PLLI2SN = 99
001100100: PLLI2SN = 100
001100101: PLLI2SN = 101
001100110: PLLI2SN = 102
...
110110000: PLLI2SN = 432
110110001: PLLI2SN = 433, yanlış yapılandırma
...
111111111: PLLI2SN = 511, yanlış yapılandırma

Not: 1 MHz'den yüksek VCO giriş frekansları için 50 ila 99 arasında çarpan faktörleri mümkündür. Ancak, minimum VCO çıkış frekansının yukarıda belirtilen değeri karşıladığından emin olunmalıdır.

Bitler 5:0: Rezerve, sıfır değerinde tutulmalıdır.

RCC register map

Tablo 35, kayıt haritasını ve sıfır değerlerini vermektedir.

Table 35. RCC register map and reset values

Addr. offset	Register name	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	RCC_CR	Reserved				PLL12SRDY	PLL12SON	PLL1RDY	PLL1ON	Reserved				CSSON	HSEBYP	HSERDY	HSEON	HSICAL7	HSICAL6	HSICAL5	HSICAL4	HSICAL3	HSICAL2	HSICAL1	HSICAL0	HSITRIM4	HSITRIM3	HSITRIM2	HSITRIM1	HSITRIM0	Reserved	HSIRDY	HSION									
	Reset value																																									
0x04	RCC_PLLCFGR	Reserved				PLLQ3	PLLQ2	PLLQ1	PLLQ0	Reserved	PLLSRC	Reserved				PLLP1	PLLP0	Reserved	PLLN8	PLLN7	PLLN6	PLLN5	PLLN4	PLLN3	PLLN2	PLLN1	PLLN0	PLLM5	PLLM4	PLLM3	PLLM2	PLLM1	PLLM0									
	Reset value																																									
0x08	RCC_CFGR	MCO21	MCO20	MCO2PRE2	MCO2PRE1	MCO2PRE0	MCO1PRE2	MCO1PRE1	MCO1PRE0	I2SSRC	MCO11	MCO10	RTCPRE4	RTCPRE3	RTCPRE2	RTCPRE1	RTCPRE0	PPRE22	PPRE21	PPRE20	PPRE12	PPRE11	PPRE10	Reserved	Reserved	HPRE3	HPRE2	HPRE1	HPRE0	SWS1	SWS0	SW1	SW0									
	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									
0x0C	RCC_CIR	Reserved								CSSC	Reserved	PLL12SRDYC	PLL1RDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC	Reserved				PLL12SRDYIE	PLL1RDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE	CSSF	Reserved	PLL12SRDYF	PLL1RDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF							
	Reset value									0																																
0x10	RCC_AHB1RSTR	Reserved	OTGHSRST		Reserved				ETHMACRST	Reserved	DMA2RST	DMA1RST	Reserved										CRCRST	Reserved				GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST	GPIORST				
	Reset value		0	0					0		0	0											0					0	0	0	0	0	0	0	0	0	0	0	0	0		
0x14	RCC_AHB2RSTR	Reserved																																OTGFSRST	RNGRST	HSARST	CRYPRST	Reserved				DCMRST
	Reset value																																									
0x18	RCC_AHB3RSTR	Reserved																																FSMC RST								
	Reset value																																	0								
0x1C	Reserved	Reserved																																								
0x20	RCC_APB1RSTR	Reserved	DACRST	PWR RST	Reserved	CAN2RST	CAN1RST	Reserved	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	UART3RST	UART2RST	Reserved	SPI3RST	SPI2RST	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TIM14RST	TIM13RST	TIM12RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST									
	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	RCC_APB2RSTR	Reserved										TIM11RST	TIM10RST	TIM9RST	Reserved	SYSCFGRST	Reserved	SPI1RST	SDIORST	Reserved	ADCRST	Reserved	USART6RST	USART1RST	Reserved	TIM8RST	TIM1RST															
	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								
0x28	Reserved	Reserved																																								
0x2C	Reserved	Reserved																																								
0x30	RCC_AHB1ENR	Reserved	OTGHSULPIEN	OTGHSEN	ETHMACPTEN	ETHMACRXEN	ETHMACTXEN	ETHMACEN	Reserved				DMA2EN	DMA1EN	CCMDATARAMEN	Reserved	BKPSRAMEN	Reserved				CRCCEN	Reserved				GPIOEN	GPIOEN	GPIOEN	GPIOEN	GPIOEN	GPIOEN	GPIOEN	GPIOEN	GPIOEN							
	Reset value		0	0	0	0	0	0					0	0	0	0	0					0					0	0	0	0	0	0	0	0	0	0						

Kodlama Öncesi Giriş

Bu uygulama, iki farklı geliştirme ortamında test edilmiştir:

1. IAR Embedded Workbench 9.50.2

- **Kullanılan Kütüphane:** Standard Peripheral Library (StdPeriph)
- Bu kütüphane, doğrudan donanım register'larına erişim sağlamak ve daha düşük seviye bir kontrol imkanı sunmak için tercih edilir. Daha eski kodlarla uyumludur ve hafif bir yapı sunar.

2. STM32CubeIDE

- **Kullanılan Kütüphane:** Hardware Abstraction Layer (HAL)
- HAL, donanım soyutlama katmanı sunarak daha taşınabilir ve okunabilir bir kod yazmayı sağlar. Ancak, performans açısından bazı ek maliyetler getirebilir.

Bu iki yaklaşımı karşılaştırarak, saat yapılandırmasının farklı kütüphanelerle nasıl yapıldığını göreceğiz ve her iki yöntemin avantajlarını değerlendireceğiz.

Şimdi adım adım kodlama aşamasına geçelim ve önce **StdPeriph** ardından **HAL** kullanarak saat yapılandırmasını gerçekleştirelim.

STM32F407 Saat Yapılandırması (StdPeriph Kütüphanesi ile)

Bu uygulamada, STM32F407 mikrodenetleyicisinin saat sisteminin nasıl yapılandırılacağını öğreneceğiz. Saat konfigürasyonu, mikrodenetleyicinin performansını doğrudan etkileyen önemli bir bileşendir. Bu çalışmada, **HSI** (High-Speed Internal) osilatörünü kullanarak sistem saatini yapılandıracağız ve **MCO2** pini üzerinden dışarıya saat sinyalini vereceğiz. Adım adım ilerleyerek, temel saat yapılandırmalarını gerçekleştirecek ve osiloskop yardımıyla doğrulama yapacağız.

Fonksiyon Açıklamaları

HSI_Setup()

Bu fonksiyon, sistem saat kaynağını Yüksek Hızlı Dahili Osilatör (**HSI**) olarak ayarlar ve HSI'nin etkinleştirildiğinden emin olur.

Adımlar:

1. HSI'yi etkinleştir: **RCC_HSIcmd(ENABLE);**
2. Sistem saat kaynağını **HSI** olarak ayarla: **RCC_SYSClkConfig(RCC_SYSClkSource_HSI);**

Amacı:

- Dahili osilatörün sistem saati olarak kullanılmasını sağlar.
- Temel uygulamalar için stabil bir saat kaynağı sunar.

microcontrollerClockOutInit()

Bu fonksiyon, sistem saat sinyalini dışarıya aktarmak için **PC9** pinini (**MCO2**) yapılandırır.

Adımlar:

1. GPIOC çevre birimi saatini etkinleştir:
RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC,ENABLE);
2. GPIO pinini alternatif fonksiyon (**AF**), push-pull çıkış ve yüksek hız olarak yapılandır.
3. GPIO ayarlarını **GPIO_Init()** fonksiyonu ile uygula.
4. MCO2 kaynağını sistem saati olarak ayarla ve bölme kullanma:
RCC_MCO2Config(RCC_MCO2Source_SYSClk, RCC_MCO2Div_1);

Amacı:

- Konfigüre edilen sistem saatini **PC9** pinine aktarmak.

- Osiloskop ile saat frekansının doğrulanmasını sağlamak.

Ana Fonksiyon Genel Görünümü

1. Başlangıçta sistem saat değerini **systemClock1** değişkenine kaydet.
2. **HSI_Setup()** fonksiyonunu çağırarak sistem saatini **HSI**'ye ayarla.
3. **microcontrollerClockOutInit()** fonksiyonu ile **MCO2** çıkışını yapılandır.
4. **SystemCoreClockUpdate()** fonksiyonu ile sistem saatini güncelle.
5. Güncellenmiş sistem saatini **systemClock2** değişkenine kaydet.
6. Yapılandırmayı korumak için sonsuz döngüye gir.

Beklenen Sonuç

- **PC9** pininden osiloskop ile **16 MHz**'lik saat sinyalinin gözlemlenmesi.
- Dahili saat konfigürasyonunun başarıyla doğrulanması.

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

HSI_Oscillator - IAR Embedded Workbench IDE - Arm 9.50.2

File Edit View Project Debug Disassembly ST-Link Tools Window Help

Workspace

Debug

Files

HSI_Oscillator - Debug

Application

main.c

CMSIS

stm32f4xx.h

stm32f4xx_conf.h

system_stm32f4xx.c

system_stm32f4xx.h

Doc

readme.md

StdPeriphDrivers

Inc

stm32f4xx_gpio.h

stm32f4xx_rcc.h

Src

Output

main.c

```
1 /******
2 * Includes
3 *****/
4 #include <stm32f4xx.h>
5 #include <stm32f4xx_gpio.h>
6 #include <stm32f4xx_rcc.h>
7 #include <stm32f4xx_conf.h>
8
9
10 /******
11 * Definitions
12 *****/
13 #define MCO_PORT          GPIOC
14 #define MCO_PIN           GPIO_Pin_9
15
16
17 /******
18 * Function Declarations
19 *****/
20 void HSI_Setup(void);
21 void microcontrollerClockOutInit(void);
22
23
24 /******
25 * Variables
26 *****/
27 uint32_t systemClock1 = 0;
28 uint32_t systemClock2 = 0;
29
30
31 int main(void)
32 {
33     systemClock1 = SystemCoreClock;
34
35     HSI_Setup();
36
37     microcontrollerClockOutInit();
38
39     SystemCoreClockUpdate();
40
41     systemClock2 = SystemCoreClock;
42
43
44     while(1)
45     {
46
47     }
48 }
49
50
51
52 /******
53 * @brief HSI_Setup() fonksiyonu sistem saatini yuksek hi
54 * ayarlar.
55 *
56 * @param None
57 *
58 * @retval None
59 *****/
60 void HSI_Setup(void)
61 {
62     //HSI'nin etkinlestirildiginden emin olun
63     RCC_HSIcmd(ENABLE);
64
65     //sistem saatini HSI osilatorunden elde edilmiş bir ka
66     RCC_SYSClkConfig(RCC_SYSClkSource_HSI);
67 }
68
69
70 /******
71 * @brief microcontrollerClockOutInit() fonksiyonu MCO2 o
72 * clock sinyalinı disariya cikmasini saglar. HSI
73 * oldugundan PC9 pininden osilaskop ile olcum yap
74 * gorursunuz.
75 *
76 * @param None
77 *
78 * @retval None
79 *****/
80 void microcontrollerClockOutInit(void)
81 {
82     GPIO_InitTypeDef          pxMCO = {0};
83
84     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE); //
85
86     pxMCO.GPIO_Mode = GPIO_Mode_AF;
87     pxMCO.GPIO_Pin = MCO_PIN;
88     pxMCO.GPIO_OType = GPIO_OType_PP;
89     pxMCO.GPIO_PuPd = GPIO_PuPd_NOPULL;
90     pxMCO.GPIO_Speed = GPIO_Speed_100MHz;
91
92     GPIO_Init(MCO_PORT, &pxMCO); //MCO2 için pimi baslatın
93     RCC_MCO2Config(RCC_MCO2Source_SYSClk, RCC_MCO2Div_1);
94 }
95
96
97
98
99
```

Disassembly

Go to:

Zone:

Disassembly

GPIOx->OTYPER |= (uint16_t) (((

0x800'015e: 0x6845 LDR

0x800'0160: 0x798a LDRB

0x800'0162: 0x409a LSLS

0x800'0164: 0xb292 UXTH

0x800'0166: 0x4315 ORRS

0x800'0168: 0x6045 STR

GPIOx->PUPDR &= ~(GPIO_PUPDR_PUP

??GPIO_Init_4:

0x800'016a: 0x68c5 LDR

0x800'016c: 0x001a MOVS

0x800'016e: 0xb2d2 UXTB

0x800'0170: 0x0052 LSLS

0x800'0172: 0x409a LSLS

0x800'0174: 0xea35 0x0404 BICS.W

0x800'0178: 0x60c4 STR

GPIOx->PUPDR |= ((uint32_t) GPIO

0x800'017a: 0x68c4 LDR

0x800'017c: 0x79cd LDRB

0x800'017e: 0x001a MOVS

0x800'0180: 0xb2d2 UXTB

0x800'0182: 0x0052 LSLS

0x800'0184: 0x4095 LSLS

0x800'0186: 0x432c ORRS

0x800'0188: 0x60c4 STR

for (pinpos = 0x00; pinpos < 0x10; p

??GPIO_Init_2:

0x800'018a: 0x1c5b ADDS

for (pinpos = 0x00; pinpos < 0x10; p

??GPIO_Init_0:

0x800'018c: 0x2b10 CMP

0x800'018e: 0xd3b3 BCC.N

}

0x800'0190: 0xbcf0 POP

0x800'0192: 0x4770 BX

int main(void)

{

main:

0x800'0194: 0xb510 PUSH

systemClock1 = SystemCoreClock;

0x800'0196: 0x4c1c LDR.N

0x800'0198: 0x6821 LDR

0x800'019a: 0x481c LDR.N

0x800'019c: 0x6001 STR

HSI_Setup();

0x800'019e: 0xf000 0xf808 BL

microcontrollerClockOutInit();

0x800'01a2: 0xf000 0xf80e BL

SystemCoreClockUpdate();

0x800'01a6: 0xf7ff 0xff4b BL

systemClock2 = SystemCoreClock;

0x800'01aa: 0x6821 LDR

0x800'01ac: 0x4818 LDR.N

0x800'01ae: 0x6001 STR

while(1)

??main_0:

0x800'01b0: 0xe7fe B.N

void HSI_Setup(void)

{

HSI_Setup:

0x800'01b2: 0xb580 PUSH

RCC_HSIcmd(ENABLE);

0x800'01b4: 0x2001 MOVS

0x800'01b6: 0xf000 0xf82f BL

RCC_SYSClkConfig(RCC_SYSClkSource_HS

0x800'01ba: 0x2000 MOVS

0x800'01bc: 0xf000 0xf83b BL

}

0x800'01c0: 0xbd01 POP

void microcontrollerClockOutInit(void)

{

microcontrollerClockO

0x800'01c2: 0xb5e0 PUSH

0x800'01c4: 0x2108 MOVS

0x800'01c6: 0x2200 MOVS

0x800'01c8: 0x4668 MOV

0x800'01ca: 0xf000 0xf851 BL

RCC_AHB1PeriphClockCmd(RCC_AHB1Perip

0x800'01ce: 0x2101 MOVS

0x800'01d0: 0x2004 MOVS

0x800'01d2: 0xf000 0xf839 BL

pxMCO.GPIO_Mode = GPIO_Mode_AF;

0x800'01d6: 0x2002 MOVS

0x800'01d8: 0xf88d 0x0004 STRB.W

pxMCO.GPIO_Pin = MCO_PIN;

0x800'01dc: 0xf44f 0x7000 MOV.W

0x800'01e0: 0x9000 STR

pxMCO.GPIO_OType = GPIO_OType_PP;

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

HSI_Oscillator - IAR Embedded Workbench IDE - Arm 9.50.2

File Edit View Project Debug Disassembly ST-Link Tools Window Help

Workspace

Debug

Files

HSI_Oscillator - Debug

Application

main.c

CMSIS

stm32f4xx.h

stm32f4xx_conf.h

system_stm32f4xx.c

system_stm32f4xx.h

Doc

readme.md

StdPeriphDrivers

Inc

stm32f4xx_gpio.h

stm32f4xx_rcc.h

Output

main.c

```
1 /******
2 * Includes
3 *****
4 #include <stm32f4xx.h>
5 #include <stm32f4xx_gpio.h>
6 #include <stm32f4xx_rcc.h>
7 #include <stm32f4xx_conf.h>
8
9
10 /******
11 * Definitions
12 *****
13 #define MCO_PORT          GPIOC
14 #define MCO_PIN           GPIO_Pin_9
15
16
17 /******
18 * Function Declarations
19 *****
20 void HSI_Setup(void);
21 void microcontrollerClockOutInit(void);
22
23
24 /******
25 * Variables
26 *****
27 uint32_t systemClock1 = 0;
28 uint32_t systemClock2 = 0;
29
30
31 int main(void)
32 {
33     systemClock1 = SystemCoreClock;
34
35     HSI_Setup();
36
37     microcontrollerClockOutInit();
38
39     SystemCoreClockUpdate();
40
41     systemClock2 = SystemCoreClock;
42
43
44     while(1)
45     {
46
47
48     }
49 }
50
51
52 /******
53 * @brief HSI_Setup() fonksiyonu sistem saatini yuksek hi
54 ayarlar.
55 *
56 * @param None
57 *
58 * @retval None
59 *****
60 void HSI_Setup(void)
61 {
62     //HSI'nin etkinlestirildiginden emin olun
63     RCC_HSIcmd(ENABLE);
64
65     //sistem saatini HSI osilatorunden elde edilmiş bir ka
66     RCC_SYSClkConfig(RCC_SYSClkSource_HSI);
67 }
68
69
70 /******
71 * @brief microcontrollerClockOutInit() fonksiyonu MCO2 o
72 clock sinyali disariya cikmasini saglar. HSI
73 oldugundan PC9 pininden osilaskop ile olcum yap
74 gorursunuz.
75 *
76 * @param None
77 *
78 * @retval None
79 *****
80 void microcontrollerClockOutInit(void)
81 {
82     GPIO_InitTypeDef          pxMCO = {0};
83
84     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE); //
85
86     pxMCO.GPIO_Mode = GPIO_Mode_AF;
87     pxMCO.GPIO_Pin = MCO_PIN;
88     pxMCO.GPIO_OType = GPIO_OType_PP;
89     pxMCO.GPIO_PuPd = GPIO_PuPd_NOPULL;
90     pxMCO.GPIO_Speed = GPIO_Speed_100MHz;
91
92     GPIO_Init(MCO_PORT, &pxMCO); //MCO2 icin pimi baslatin
93     RCC_MC02Config(RCC_MC02Source_SYSClk, RCC_MC02Div_1);
94 }
95
96
97
98
99
```

Disassembly

Go to:

Zone:

Disassembly

```
GPIOx->OTYPER |= (uint16_t) (((
0x800'015e: 0x6845     LDR
0x800'0160: 0x798a     LDRB
0x800'0162: 0x409a     LSLS
0x800'0164: 0xb292     UXTH
0x800'0166: 0x4315     ORRS
0x800'0168: 0x6045     STR
GPIOx->PUPDR &= ~(GPIO_PUPDR_PUP
??GPIO_Init_4:
0x800'016a: 0x68c5     LDR
0x800'016c: 0x001a     MOVS
0x800'016e: 0xb2d2     UXTB
0x800'0170: 0x0052     LSLS
0x800'0172: 0x409a     LSLS
0x800'0174: 0xea35 0x0404 BICS.W
0x800'0178: 0x60c4     STR
GPIOx->PUPDR |= ((uint32_t)GPIO
0x800'017a: 0x68c4     LDR
0x800'017c: 0x79cd     LDRB
0x800'017e: 0x001a     MOVS
0x800'0180: 0xb2d2     UXTB
0x800'0182: 0x0052     LSLS
0x800'0184: 0x4095     LSLS
0x800'0186: 0x432c     ORRS
0x800'0188: 0x60c4     STR
for (pinpos = 0x00; pinpos < 0x10; p
??GPIO_Init_2:
0x800'018a: 0x1c5b     ADDS
for (pinpos = 0x00; pinpos < 0x10; p
??GPIO_Init_0:
0x800'018c: 0x2b10     CMP
0x800'018e: 0xd3b3     BCC.N
}
0x800'0190: 0xbcf0     POP
0x800'0192: 0x4770     BX
int main(void)
{
    main:
0x800'0194: 0xb510     PUSH
systemClock1 = SystemCoreClock;
0x800'0196: 0x4c1c     LDR.N
0x800'0198: 0x6821     LDR
0x800'019a: 0x481c     LDR.N
0x800'019c: 0x6001     STR
HSI_Setup();
0x800'019e: 0xf000 0xf808 BL
microcontrollerClockOutInit();
0x800'01a2: 0xf000 0xf80e BL
SystemCoreClockUpdate();
0x800'01a6: 0xf7ff 0xff4b BL
systemClock2 = SystemCoreClock;
0x800'01aa: 0x6821     LDR
0x800'01ac: 0x4818     LDR.N
0x800'01ae: 0x6001     STR
while(1)
??main_0:
0x800'01b0: 0xe7fe     B.N
void HSI_Setup(void)
{
    HSI_Setup:
0x800'01b2: 0xb580     PUSH
RCC_HSIcmd(ENABLE);
0x800'01b4: 0x2001     MOVS
0x800'01b6: 0xf000 0xf82f BL
RCC_SYSClkConfig(RCC_SYSClkSource_HS
0x800'01ba: 0x2000     MOVS
0x800'01bc: 0xf000 0xf83b BL
}
0x800'01c0: 0xbd01     POP
void microcontrollerClockOutInit(void)
{
    microcontrollerClockO
0x800'01c2: 0xb5e0     PUSH
0x800'01c4: 0x2108     MOVS
0x800'01c6: 0x2200     MOVS
0x800'01c8: 0x4668     MOV
0x800'01ca: 0xf000 0xf851 BL
RCC_AHB1PeriphClockCmd(RCC_AHB1Perip
0x800'01ce: 0x2101     MOVS
0x800'01d0: 0x2004     MOVS
0x800'01d2: 0xf000 0xf839 BL
pxMCO.GPIO_Mode = GPIO_Mode_AF;
0x800'01d6: 0x2002     MOVS
0x800'01d8: 0xf88d 0x0004 STRB.W
pxMCO.GPIO_Pin = MCO_PIN;
0x800'01dc: 0xf44f 0x7000 MOV.W
0x800'01e0: 0x9000     STR
pxMCO.GPIO_OType = GPIO_OType_PP;

```

HSI_Oscillator

Debug Log

Log

SC\HSI_Oscillator\Debug\Exe\HSI_Oscillator.out to flash memory.

Sun Jan 26, 2025 10:52:56: 996 bytes downloaded into FLASH (1.00 Kbytes/sec)

Sun Jan 26, 2025 10:52:56: Loaded debuggee: D:\es\Projects\EWARM_IDE\STM32

F4_DISC\HSI_Oscillator\Debug\Exe\HSI_Oscillator.out

Sun Jan 26, 2025 10:52:56: Hardware reset with strategy 0 was performed

Sun Jan 26, 2025 10:52:56: Download completed.

Sun Jan 26, 2025 10:52:56: Hardware reset with strategy 0 was performed

Live Watch

Expression	Value	Location	Type
systemClock1	168'000'000	0x2000'0014	uint32_t
systemClock2	16'000'000	0x2000'0018	uint32_t
<click to add>			

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

HSI_Oscillator - IAR Embedded Workbench IDE - Arm 9.50.2

File Edit View Project Debug Disassembly ST-Link Tools Window Help

Workspace

Debug

Files

- HSI_Oscillator - Debug
 - Application
 - main.c
 - CMSIS
 - stm32f4xx.h
 - stm32f4xx_conf.h
 - system_stm32f4xx.c
 - system_stm32f4xx.h
 - Doc
 - readme.md
 - StdPeriphDrivers
 - Inc
 - stm32f4xx_gpio.h
 - stm32f4xx_rcc.h
 - Output

main.c

```
1 /******  
2 * Includes  
3 *****  
4 #include <stm32f4xx.h>  
5 #include <stm32f4xx_gpio.h>  
6 #include <stm32f4xx_rcc.h>  
7 #include <stm32f4xx_conf.h>  
8  
9  
10 /******  
11 * Definitions  
12 *****  
13 #define MCO_PORT                GPIOC  
14 #define MCO_PIN                 GPIO_Pin_9  
15  
16  
17 /******  
18 * Function Declarations  
19 *****  
20 void HSI_Setup(void);  
21 void microcontrollerClockOutInit(void);  
22  
23  
24 /******  
25 * Variables  
26 *****  
27 uint32_t systemClock1 = 0;  
28 uint32_t systemClock2 = 0;  
29  
30  
31 int main(void)  
32 {  
33     systemClock1 = SystemCoreClock;  
34  
35     HSI_Setup();  
36  
37     microcontrollerClockOutInit();  
38  
39     SystemCoreClockUpdate();  
40  
41     systemClock2 = SystemCoreClock;  
42  
43  
44     while(1)  
45     {  
46  
47     }  
48 }  
49  
50  
51  
52 /******  
53 * @brief HSI_Setup() fonksiyonu sistem saatini yuksek hi  
54 * ayarlar.  
55 *  
56 * @param None  
57 *  
58 * @retval None  
59 *****  
60 void HSI_Setup(void)  
61 {  
62     //HSI'nin etkinlestirildiginden emin olun  
63     RCC_HSIcmd(ENABLE);  
64  
65     //sistem saatini HSI osilatorunden elde edilmiş bir ka  
66     RCC_SYSClkConfig(RCC_SYSClkSource_HSI);  
67 }  
68  
69  
70 /******  
71 * @brief microcontrollerClockOutInit() fonksiyonu MCO2 o  
72 * clock sinyali disariya cikmasini saglar. HSI  
73 * oldugundan PC9 pininden osilaskop ile olcum yap  
74 * gorursunuz.  
75 *  
76 * @param None  
77 *  
78 * @retval None  
79 *****  
80 void microcontrollerClockOutInit(void)  
81 {  
82  
83     GPIO_InitTypeDef                pxMCO = {0};  
84  
85     RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE); //  
86  
87     pxMCO.GPIO_Mode = GPIO_Mode_AF;  
88     pxMCO.GPIO_Pin = MCO_PIN;  
89     pxMCO.GPIO_OType = GPIO_OType_PP;  
90     pxMCO.GPIO_PuPd = GPIO_PuPd_NOPULL;  
91     pxMCO.GPIO_Speed = GPIO_Speed_100MHz;  
92  
93     GPIO_Init(MCO_PORT, &pxMCO); //MCO2 icin pimi baslatin  
94     RCC_MCO2Config(RCC_MCO2Source_SYSClk, RCC_MCO2Div_1);  
95 }  
96  
97  
98  
99
```

Disassembly

Go to: Zone:

Disassembly

```
GPIOx->OTYPER |= (uint16_t) (((  
0x800'015e: 0x6845 LDR  
0x800'0160: 0x798a LDRB  
0x800'0162: 0x409a LSLS  
0x800'0164: 0xb292 UXTH  
0x800'0166: 0x4315 ORRS  
0x800'0168: 0x6045 STR  
GPIOx->PUPDR &= ~(GPIO_PUPDR_PUP  
??GPIO_Init_4:  
0x800'016a: 0x68c5 LDR  
0x800'016c: 0x001a MOVS  
0x800'016e: 0xb2d2 UXTB  
0x800'0170: 0x0052 LSLS  
0x800'0172: 0x409a LSLS  
0x800'0174: 0xea35 0x0404 BICS.W  
0x800'0178: 0x60c4 STR  
GPIOx->PUPDR |= ((uint32_t)GPIO  
0x800'017a: 0x68c4 LDR  
0x800'017c: 0x79cd LDRB  
0x800'017e: 0x001a MOVS  
0x800'0180: 0xb2d2 UXTB  
0x800'0182: 0x0052 LSLS  
0x800'0184: 0x4095 LSLS  
0x800'0186: 0x432c ORRS  
0x800'0188: 0x60c4 STR  
for (pinpos = 0x00; pinpos < 0x10; p  
??GPIO_Init_2:  
0x800'018a: 0x1c5b ADDS  
for (pinpos = 0x00; pinpos < 0x10; p  
??GPIO_Init_0:  
0x800'018c: 0x2b10 CMP  
0x800'018e: 0xd3b3 BCC.N  
}  
0x800'0190: 0xbcf0 POP  
0x800'0192: 0x4770 BX  
int main(void)  
{  
main:  
0x800'0194: 0xb510 PUSH  
systemClock1 = SystemCoreClock;  
0x800'0196: 0x4c1c LDR.N  
0x800'0198: 0x6821 LDR  
0x800'019a: 0x481c LDR.N  
0x800'019c: 0x6001 STR  
HSI_Setup();  
0x800'019e: 0xf000 0xf808 BL  
microcontrollerClockOutInit();  
0x800'01a2: 0xf000 0xf80e BL  
SystemCoreClockUpdate();  
0x800'01a6: 0xf7ff 0xff4b BL  
systemClock2 = SystemCoreClock;  
0x800'01aa: 0x6821 LDR  
0x800'01ac: 0x4818 LDR.N  
0x800'01ae: 0x6001 STR  
while(1)  
??main_0:  
0x800'01b0: 0xe7fe B.N  
void HSI_Setup(void)  
{  
HSI_Setup:  
0x800'01b2: 0xb580 PUSH  
RCC_HSIcmd(ENABLE);  
0x800'01b4: 0x2001 MOVS  
0x800'01b6: 0xf000 0xf82f BL  
RCC_SYSClkConfig(RCC_SYSClkSource_HS  
0x800'01ba: 0x2000 MOVS  
0x800'01bc: 0xf000 0xf83b BL  
}  
0x800'01c0: 0xbd01 POP  
void microcontrollerClockOutInit(void)  
{  
microcontrollerClockO  
0x800'01c2: 0xb5e0 PUSH  
0x800'01c4: 0x2108 MOVS  
0x800'01c6: 0x2200 MOVS  
0x800'01c8: 0x4668 MOV  
0x800'01ca: 0xf000 0xf851 BL  
RCC_AHB1PeriphClockCmd(RCC_AHB1Perip  
0x800'01ce: 0x2101 MOVS  
0x800'01d0: 0x2004 MOVS  
0x800'01d2: 0xf000 0xf839 BL  
pxMCO.GPIO_Mode = GPIO_Mode_AF;  
0x800'01d6: 0x2002 MOVS  
0x800'01d8: 0xf88d 0x0004 STRB.W  
pxMCO.GPIO_Pin = MCO_PIN;  
0x800'01dc: 0xf44f 0x7000 MOV.W  
0x800'01e0: 0x9000 STR  
pxMCO.GPIO_OType = GPIO_OType_PP;  

```

HSI_Oscillator

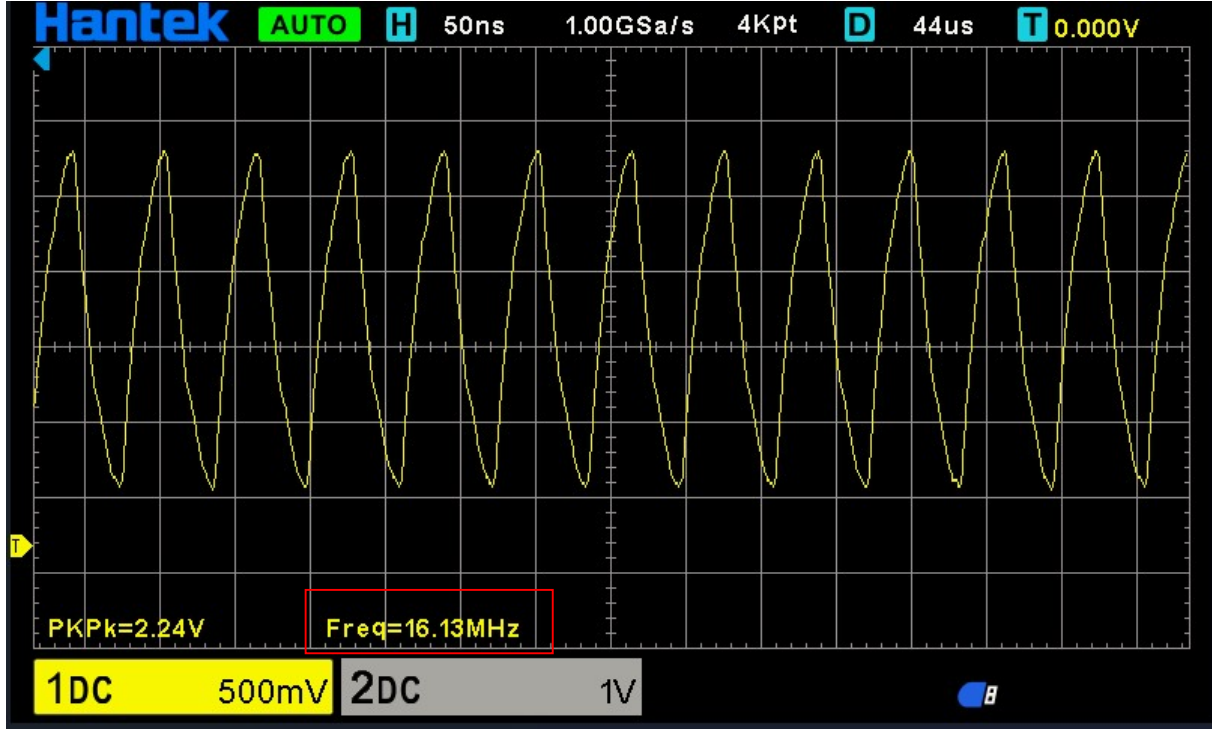
Debug Log

Log

```
SC\HSI_Oscillator\Debug\Exe\HSI_Oscillator.out to flash memory.  
Sun Jan 26, 2025 10:52:56: 996 bytes downloaded into FLASH (1.00 Kbytes/sec)  
Sun Jan 26, 2025 10:52:56: Loaded debuggee: D:\esProjects\EWARM_IDE\STM32  
F4_DISC\HSI_Oscillator\Debug\Exe\HSI_Oscillator.out  
Sun Jan 26, 2025 10:52:56: Hardware reset with strategy 0 was performed  
Sun Jan 26, 2025 10:52:56: Download completed.  
Sun Jan 26, 2025 10:52:56: Hardware reset with strategy 0 was performed
```

Live Watch

Expression	Value	Location	Type
systemClock1	168'000'000	0x2000'0014	uint32_t
systemClock2	16'000'000	0x2000'0018	uint32_t
<click to add>			



Osiloskop Ölçüm Sonucu Analizi

Yukarıdaki osiloskop görüntüsüne göre, ölçülen frekans **16.13 MHz** olarak görülmektedir. Beklenen sistem saati frekansı olan **16 MHz**'e kıyasla yaklaşık **+0.13 MHz**'lik bir sapma mevcuttur.

Değerlendirme:

- STM32F407'nin dahili **HSI (High-Speed Internal)** osilatörü fabrika çıkışlı olarak $\pm 1\%$ doğruluk ile kalibre edilmiştir. Bu doğrultuda, **16 MHz $\pm 1\%$** hata payı beklenmektedir.
- 1% hata payı hesaplandığında: **16MHz $\times 0.01 = 0.16$ MHz** Bu hesaplama sonucunda saat frekansının **15.84 MHz ile 16.16 MHz** aralığında olması beklenmektedir.
- Ölçülen değer **16.13 MHz**, beklenen hata aralığı içerisinde yer almakta ve HSI osilatörünün fabrika kalibrasyonuna uygun çalıştığını göstermektedir.

Sonuç:

Yapılan ölçümler doğrultusunda, HSI osilatörünün $\pm 1\%$ hata payı içerisinde çalıştığı doğrulanmıştır. Sistemin stabil çalıştığı ve saat kaynağının güvenilir bir şekilde kullanılabileceği sonucuna ulaşılmıştır.

Öneriler:

- Daha hassas bir zamanlama gerektiren uygulamalar için harici kristal osilatör (HSE) tercih edilebilir.

- Ortam sıcaklığı gibi dış etmenlerin uzun vadeli etkilerini test etmek için uzun süreli gözlemler yapılabilir.

Bu sonuçlar doğrultusunda, sistem saatinin başarılı bir şekilde yapılandırıldığı ve doğrulandığı söylenebilir.

Şimdi **STM32CubeIDE** üzerinde saat yapılandırmasını gerçekleştirmek için adım adım ilerleyelim. İlk olarak proje oluşturma ve gerekli ayarları yapılandırma aşamalarına geçebiliriz.

Adım 1: STM32CubeIDE Projesini Oluşturma

1. **STM32CubeIDE**'yi açın ve yeni bir proje başlatın.
2. **"Board Selector"** veya **"MCU Selector"** kullanarak **STM32F407** mikrodenetleyicisini seçin.
3. Projeye anlamlı bir isim verin (örneğin: `Clock_Config_CubeIDE`).
4. Kütüphane olarak **HAL (Hardware Abstraction Layer)** seçildiğinden emin olun.
5. **Generate Code** seçeneği ile kodları oluşturun.

Adım 2: Clock Configuration Arayüzünü Kullanma

1. **Clock Configuration** sekmesine geçin.
2. **HSI**'yi sistem saati kaynağı olarak seçin.
3. **MCO (Microcontroller Clock Output)** için PC9 pinini yapılandırarak HSI frekansını dışarıya vermek için gerekli ayarları yapın.
 - **MCO Source:** SYSCLK (System Clock)
 - **Prescaler:** 1 (Bölme faktörü olmadan çıkış)
4. **168 MHz'e PLL ile geçiş için:**
 - HSI veya HSE'yi PLL kaynağı olarak belirleyin.
 - PLL çarpanlarını uygun şekilde ayarlayın (örneğin: PLLM, PLLN, PLLP değerleri).
5. **Konfigürasyonu doğrulamak için** sağ üst köşede yer alan **"Validate Settings"** butonuna tıklayın.

Adım 3: Kod Oluşturma ve Güncelleme

Kodlar oluşturulduktan sonra, **main.c** dosyasında saat yapılandırmasını ve MCO çıkışını doğrulamak için aşağıdaki kodu ekleyelim:

```
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    // HSI'yi etkinleştir
```

```
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSISState = RCC_HSI_ON;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;
HAL_RCC_OscConfig(&RCC_OscInitStruct);

// Saat kaynağı olarak HSI seç
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0);

// MCO çıkışını etkinleştir
HAL_RCC_MCOConfig(RCC_MCO2, RCC_MCO2SOURCE_SYSCLK, RCC_MCODIV_1);
}
```

Adım 4: Derleme ve Test

1. **Projeyi derleyin** ve hata olup olmadığını kontrol edin.
 2. Kartı programlayarak çalıştırın.
 3. Osiloskop kullanarak PC9 pininden çıkışı gözlemleyin.
-

Bu adımları tamamladıktan sonra elde edilen sonuçları inceleyelim ve doğrulamaya geçelim.

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

STM32 Project

Target Selection

Select STM32 target or STM32Cube example

IDE

MCU/MPU SelectorBoard SelectorExample SelectorCross Selector

MCU/MPU Filters

Commercial Part Number

STM32F407VGT6

PRODUCT INFO

Segment

Series

Line

Marketing Status

Price

Package

Core

Coprocessor

MEMORY

Flash = 1024 (kBytes)

1024

EEPROM = 0 (Bytes)

0

RAM Total = 192 (kBytes)

192

RAM = 192 (kBytes)

192

CCM RAM = 0 (kBytes)

0

On-chip SRAM = 0 (kBytes)

0

ITCM/DTCM RAM

DRAM Support

Flash Support

Dual Bank Flash

TIMER

Timer Total = 14

14

Timer 16-bit = 12

12

Timer 32-bit = 2

2

LPTIM = 0

0

AMC Timer = 0

0

Timer Function

ANALOG

ADC Total (converters) = 3

3

ADC 12-bit (converters) = 3

3

ADC 12-bit (channels) = 16

16

ADC 14-bit (converters) = 0

0

ADC 14-bit (channels) = 0

0

STM32F4 Series

STM32F407VGT6

High-performance foundation line, Arm Cortex-M4 core with DSP and FPU, 1 Mbyte of Flash memory, 168 MHz CPU, ART Accelerator, Ethernet, FSMC

ACTIVE

Product is in mass production

Unit Price for 10kU (US\$): 5.8519

Board: STM32F407G-DISC1

LQFP 100 14x14x1.4 mm

The STM32F405xx and STM32F407xx family is based on the high-performance Arm® Cortex®-M4 32-bit RISC core operating at a frequency of up to 168 MHz. The Cortex-M4 core features a Floating point unit (FPU) single precision which supports all Arm single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security.

The STM32F405xx and STM32F407xx family incorporates high-speed embedded memories (Flash memory up to 1 Mbyte, up to 192 Kbytes of SRAM), up to 4 Kbytes of backup SRAM, and an extensive range of enhanced I/Os and peripherals connected to two APB buses, three AHB buses and a 32-bit multi-AHB bus matrix.

All devices offer three 12-bit ADCs, two DACs, a low-power RTC, twelve general-purpose 16-bit timers including two PWM timers for motor control, two general-purpose 32-bit timers, a true random number generator (RNG). They also feature standard and advanced communication interfaces.

Features

Core

Arm® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator) allowing 0-wait state execution from Flash memory, frequency up to 168 MHz, memory protection unit, 210 DMIPS/1.25 DMIPS/MHz (Dhrystone 2.1), and DSP instructions

Memories

Up to 1 Mbyte of Flash memory

Up to 192+4 Kbytes of SRAM including 64-Kbyte of CCM (core coupled memory) data RAM

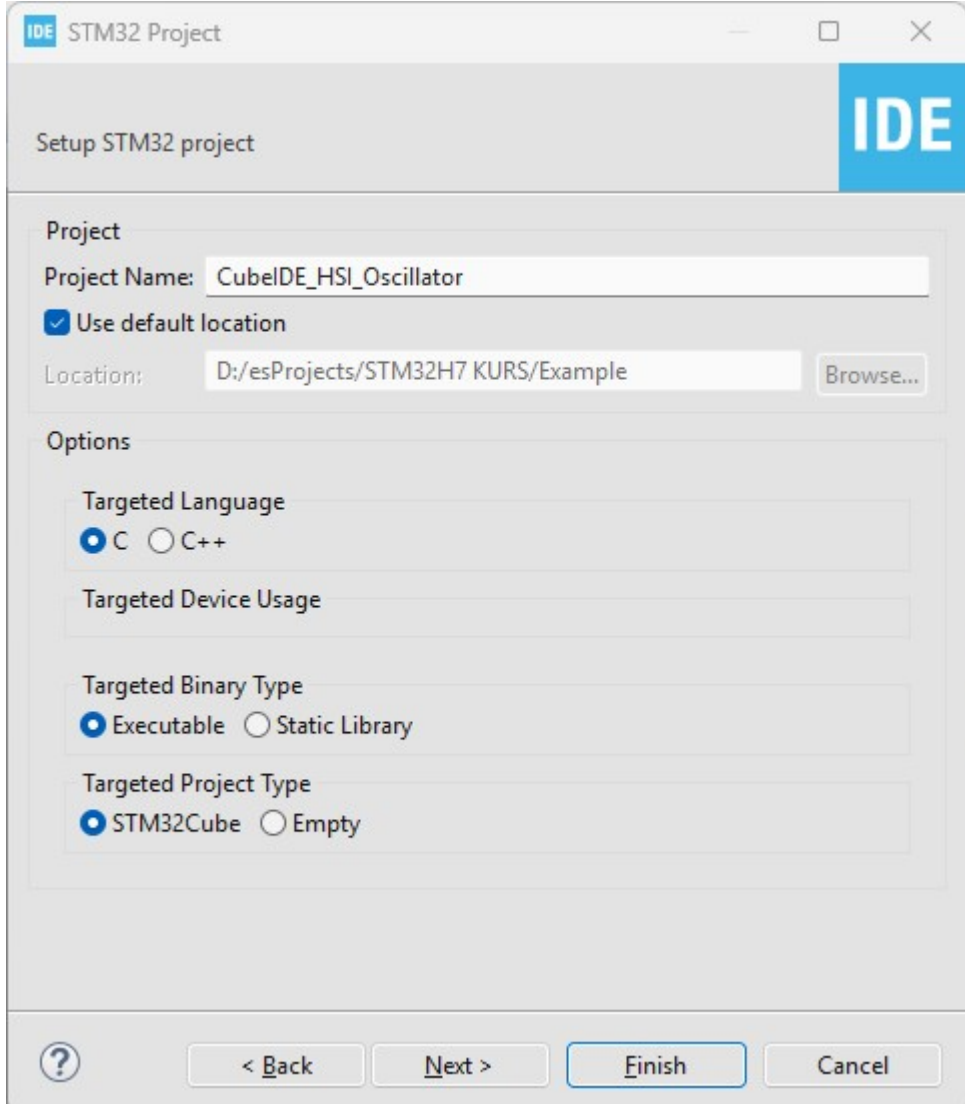
512 bytes of OTP memory

Flexible static memory controller supporting Compact Flash, SRAM,

MCUs/MPUs List: 3 items

Export

	Co...	Part No	Referen...	Mark...	Unit P...	Board	Packa...	Flash	RAM	I/O	Frequ...
☆	STM3...	STM3...	STM3...	Active	5.8519	STM32F...	LQFP	1024 k...	192 k...	82	168 M...
☆	STM3...	STM3...	STM3...	Obsol...	NA		LQFP	1024 k...	192 k...	82	168 M...
☆	STM3...	STM3...	STM3...	Active	5.8519		LQFP	1024 k...	192 k...	82	168 M...



Bu görüntü, STM32CubeIDE üzerinde yeni bir proje oluşturma ekranını göstermektedir. Yapılan ayarlar şu şekildedir:

1. Proje Bilgileri (Project Section):

- **Project Name:** CubeIDE_HSI_Oscillator
 - Proje adı olarak seçilmiş ve bu projenin **HSI osilatörü** ile ilgili olacağı anlaşılmaktadır.
- **Use default location:** Seçili durumda.
 - Proje varsayılan dizine (D:/esProjects/STM32H7 KURS/Example) kaydedilecektir.

2. Seçenekler (Options):

- **Targeted Language:**

- C dili seçili. Projenin C dilinde geliştirileceği anlamına gelir.
 - **Targeted Binary Type:**
 - Executable seçilmiş, bu da derlenen kodun doğrudan çalıştırılabilir bir program olacağını gösterir.
 - **Targeted Project Type:**
 - STM32Cube seçilmiş, bu da STM32CubeMX'in otomatik oluşturduğu kod yapısının kullanılacağını gösterir.
 - Empty seçeneği işaretlenmemiş, yani temel HAL ve CMSIS kütüphaneleriyle birlikte bir proje oluşturulacaktır.
-

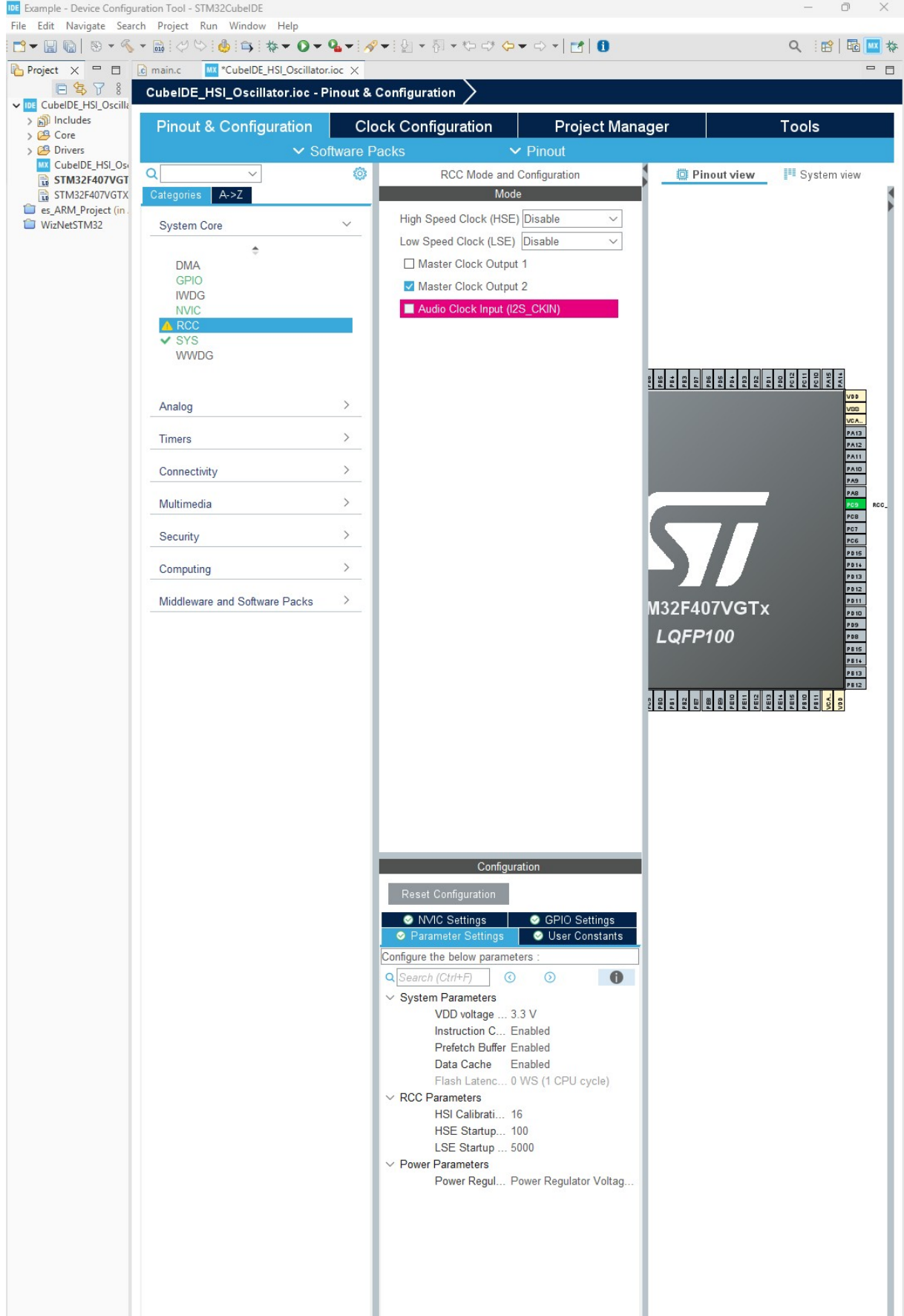
Sonraki Adımlar:

- Kullanıcı `Next >` butonuna tıklayarak mikrodenetleyici modelini seçecek ve saat konfigürasyonu gibi ayarları gerçekleştirecektir.
- Eğer tüm ayarlar doğruysa, `Finish` butonuna tıklayarak proje oluşturulabilir ve kod yazımına geçilebilir.

Genel Değerlendirme:

Bu adım, proje kurulum sürecinin başlangıcını göstermektedir. Ayarlar uygun şekilde yapılandırılmış olup, sistem saat konfigürasyonu ve HAL kütüphanesi entegrasyonu için bir temel oluşturulmaktadır.

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar



Bu görüntü, **STM32CubeIDE**'de saat konfigürasyonu ayarlarını göstermektedir. Görselde **RCC (Reset and Clock Control)** sekmesi açık ve bazı önemli saat yapılandırmaları yapılmış durumda.

Yapılandırma Analizi:

1. Saat Kaynağı Seçimleri:

- **High Speed Clock (HSE):** `Disable` olarak ayarlanmış, yani harici kristal osilatör kullanılmıyor.
- **Low Speed Clock (LSE):** `Disable` olarak ayarlanmış, yani düşük hızlı harici osilatör kullanılmıyor.
- **Master Clock Output 2 (MCO2):** Etkinleştirilmiş, bu da sistem saat sinyalinin dış dünyaya çıkış verilmesi için ayarlandığını gösterir.
- **Audio Clock Input (I2S_CKIN):** Etkinleştirilmiş (pembe renk), yani ses sistemi için harici bir saat kaynağı sağlanmış durumda.

2. System Parameters (Sistem Parametreleri):

- **VDD voltage:** 3.3V olarak belirlenmiş.
- **Instruction Cache:** Etkin durumda.
- **Prefetch Buffer:** Etkin.
- **Data Cache:** Etkin.
- **Flash Latency:** 0 WS (1 CPU cycle).

Bu ayarlar, mikrodnetleyicinin güç ve bellek erişim optimizasyonlarını sağlamak için kullanılır.

3. RCC Parameters (Saat Parametreleri):

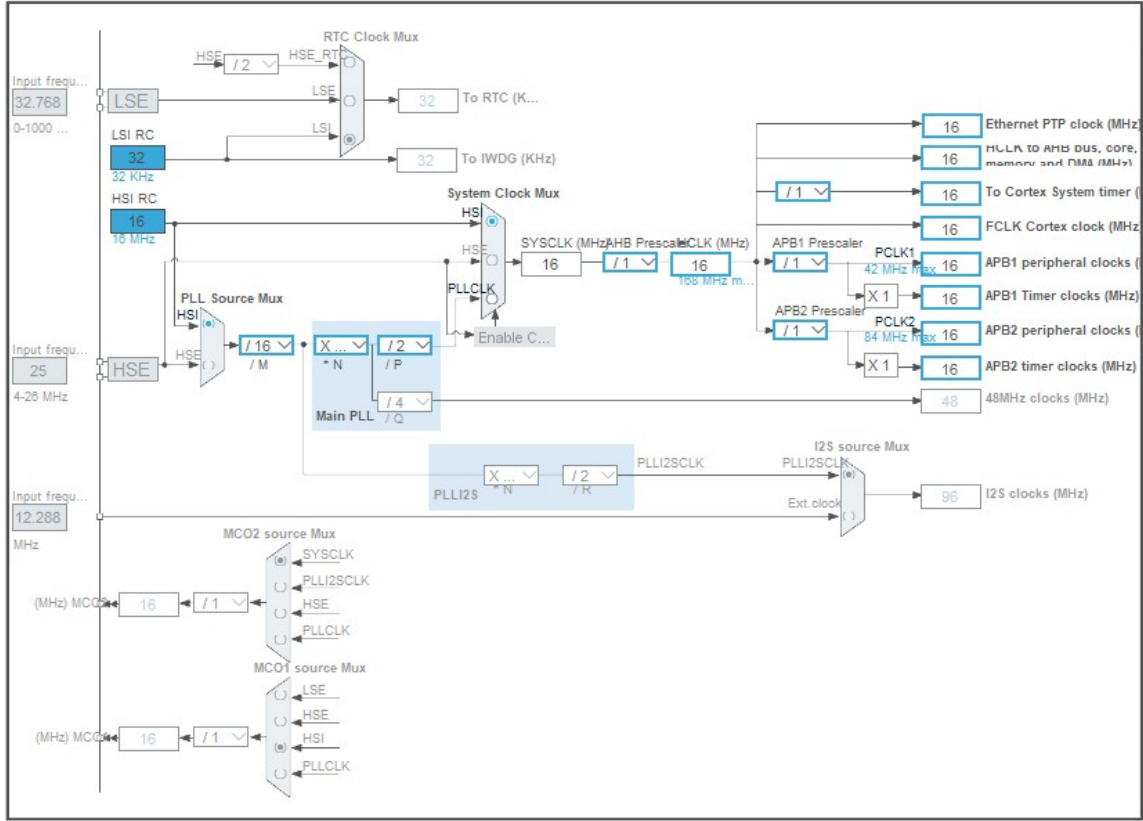
- **HSI Calibration:** 16 (Fabrika kalibrasyonu kullanılmış)
- **HSE Startup:** 100 (HSE'nin başlatılması için bekleme süresi)
- **LSE Startup:** 5000 (LSE başlatma süresi)

Bu değerler, mikrodnetleyicinin dahili osilatörlerin başlatılma sürelerini tanımlar.

Sonraki Adımlar:

1. **Clock Configuration sekmesine geçelim:**
 - Saat kaynağını **HSI**'den **PLL**'ye çevirmek ve frekans artırma ayarlarını yapmak için ilgili sekmeye gidilebilir.
2. **MCO2 çıkışının doğrulanması:**
 - Şu an etkinleştirilen **MCO2** pini üzerinden saat çıkışının osiloskop ile gözlemlenmesi gerekmektedir.
3. **Kod üretimi:**

- Yapılan ayarlar doğrultusunda kod üretilerek `SystemClock_Config()` fonksiyonunda saat yapılandırmasını kontrol edebiliriz.



Bu görüntü, **STM32CubeIDE Clock Configuration** sekmesinden alınmış saat yapılandırma ekranını göstermektedir. Şu anki yapılandırmada mikrodenetleyici, **HSI (High-Speed Internal)** saat kaynağını kullanmaktadır ve sistem saat frekansı 16 MHz olarak ayarlanmıştır.

Yapılandırma Analizi:

1. Ana Saat Kaynağı Seçimi (System Clock Mux):

- Seçili kaynak: **HSI RC (16 MHz)**
- PLL devresi şu anda devre dışı bırakılmış, sadece HSI kullanılmakta.
- Çıkış sistem saati (SYSCLK): **16 MHz**
- AHB Prescaler: **/1** (FCLK = 16 MHz)
- APB1 Prescaler: **/1**, bu da APB1 saat frekansını **16 MHz** olarak belirler.
- APB2 Prescaler: **/1**, bu da APB2 saat frekansını **16 MHz** olarak belirler.

2. PLL Yapılandırması:

- Şu anda PLL yapılandırması devre dışı.
- Eğer aktif edilirse, PLL giriş kaynağı olarak **HSI (16 MHz)** kullanılabilir.
- PLL yapılandırma seçenekleri:

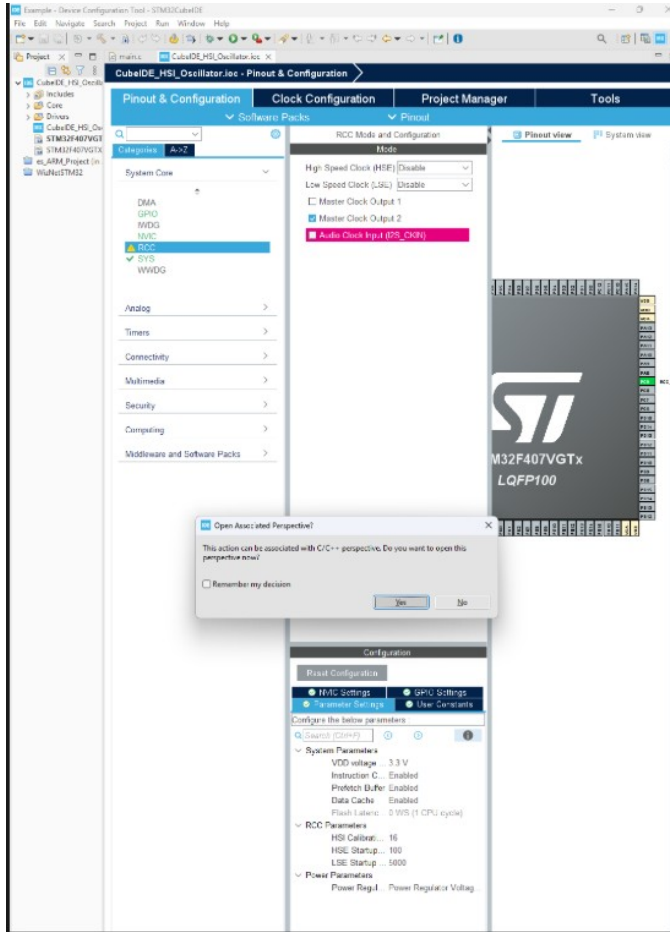
- **M:** 16 (16 MHz / 16 = 1 MHz PLL giriş frekansı)
 - **N:** Henüz tanımlanmamış
 - **P:** 4 ile çarpılarak sistem saatini yükseltmek mümkün.
-

3. MCO (Microcontroller Clock Output) Ayarları:

- **MCO2 (PC9) pini:**
 - SYSCLK (16 MHz) kaynağı seçilmiş.
 - Bölme faktörü /1 olarak ayarlanmış, böylece doğrudan 16 MHz çıkış alınabilir.
-

Sonraki Adımlar:

1. **Kod oluşturma ve inceleme:**
 - Yapılan bu ayarlar doğrultusunda CubeIDE üzerinden kod üretilecek ve ana saat yapılandırma fonksiyonları incelenecek.
 2. **PLL'yi etkinleştirerek 168 MHz'e çıkma:**
 - Saat kaynağını HSI'den PLL'ye çevirmek için ayarlamalar yapılabilir.
 - PLL parametrelerini (M, N, P değerleri) doğru şekilde seçerek sistem frekansı artırılabilir.
 3. **Kod Kontrolü:**
 - `SystemClock_Config()` fonksiyonu içerisinde bu yapılandırmanın HAL kütüphanesi fonksiyonları ile doğrulanmasını sağlayacağız.
-



Evet, bu aşamada **"Yes"** butonuna tıklayarak **C/C++ perspektifine geçiş** yapmanız gerekiyor. Bu işlem, kodlama ve derleme işlemleri için daha uygun bir çalışma ortamına geçmenizi sağlar.

Ardından, üst kısımda görülen  **simgesi (Build Project)** düğmesine tıklayarak projenizi derleyebilirsiniz. Bu düğme, CubeMX tarafından oluşturulan kodların derlenmesini sağlar.

Sonraki Adımlar:

1. Kod Derleme:

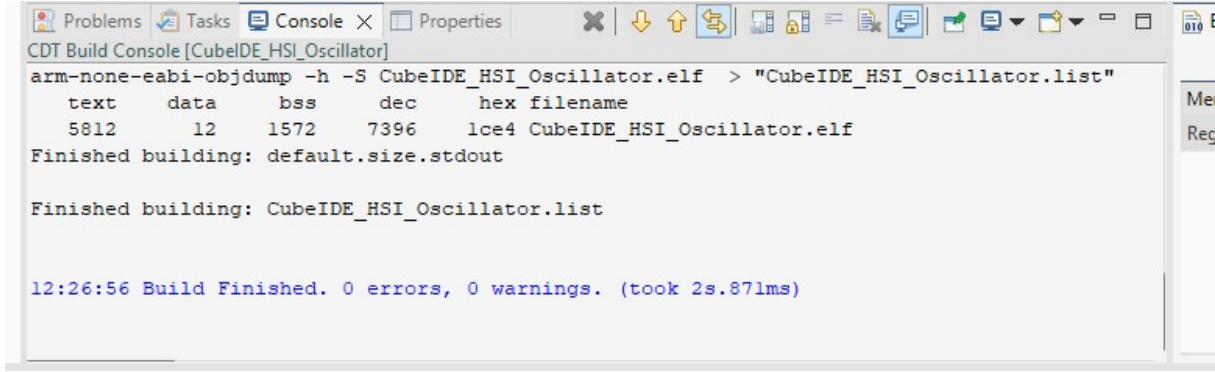
- Çark simgesine tıklayarak kodun hatasız derlenip derlenmediğini kontrol edin.
- Derleme hatası yoksa kodun başarılı bir şekilde çalıştığını doğrulamak için test aşamasına geçebilirsiniz.

2. Test ve Gözlem:

- Kartınıza kodu yükleyin ve MCO2 pininden (PC9) osiloskop ile çıkışı ölçerek 16 MHz frekans sinyalini doğrulayın.

3. Gerekirse Debug:

- Hata veya beklenmeyen bir durumla karşılaşırsanız, hata ayıklama (debug) moduna geçerek kodu satır satır inceleyebilirsiniz.



```
CDT Build Console [CubeIDE_HSI_Oscillator]
arm-none-eabi-objdump -h -S CubeIDE_HSI_Oscillator.elf > "CubeIDE_HSI_Oscillator.list"
text      data      bss      dec      hex filename
5812      12      1572     7396     1ce4 CubeIDE_HSI_Oscillator.elf
Finished building: default.size.stdout

Finished building: CubeIDE_HSI_Oscillator.list

12:26:56 Build Finished. 0 errors, 0 warnings. (took 2s.871ms)
```

Görüntüde, **CubeIDE_HSI_Oscillator** projesinin derleme işleminin başarıyla tamamlandığı görülmektedir. Konsol çıktısında aşağıdaki önemli bilgiler yer almaktadır:

Derleme Sonucu:

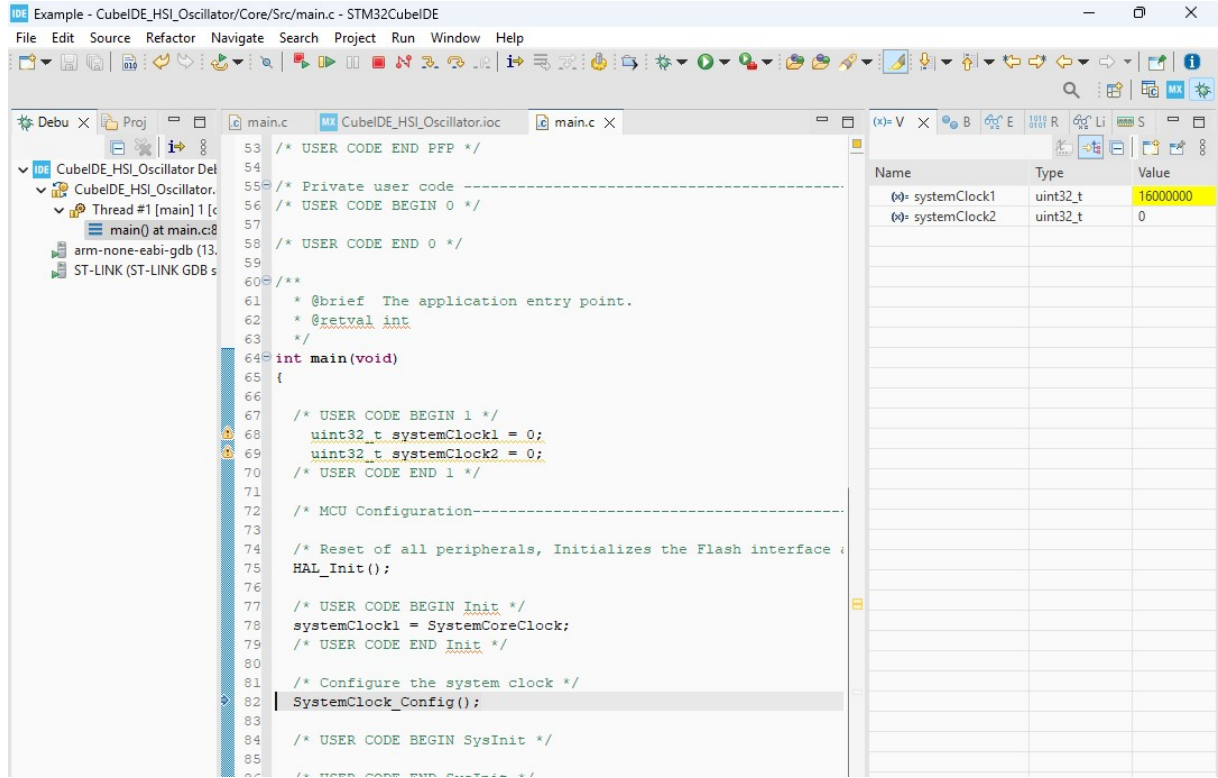
- **0 errors, 0 warnings:**
 - Kodun hatasız derlendiği ve herhangi bir uyarı bulunmadığı gösteriyor.
- **Derleme süresi:** 2.871 ms, yani hızlı bir şekilde tamamlanmış.
- **Çıktı dosyaları:**
 - CubeIDE_HSI_Oscillator.elf: Yürütülebilir dosya.
 - CubeIDE_HSI_Oscillator.list: Assembly listesi içeren dosya.
 - Derleme sonucunda toplam kod boyutları:
 - **text (kod bölümü):** 5812 byte
 - **data (başlangıç verisi):** 12 byte
 - **bss (başlangıçsız veri):** 1572 byte
 - **toplam boyut (dec):** 7396 byte

Sonraki Adımlar:

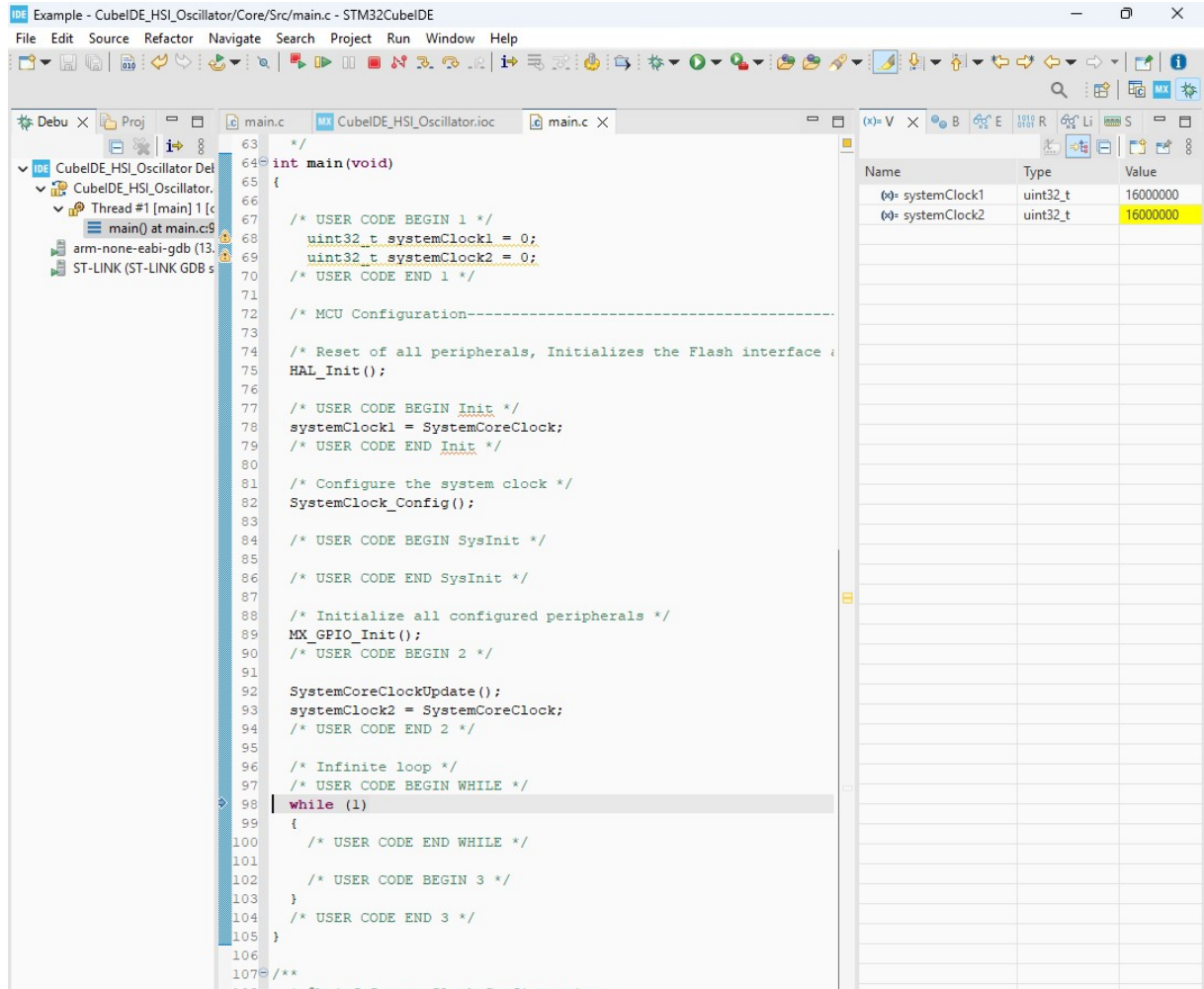
1. **Programı Cihaza Yükleme (Flashing):**
 - STM32 kartınızı bilgisayara bağlayın.
 - Üst menüdeki **"Run" (Yeşil Oynat) düğmesine** tıklayarak kodu karta yükleyin.
2. **Çıkışı Gözlemlenme:**
 - Osiloskop yardımıyla PC9 pininden 16 MHz saat sinyalinin gözlemlenmesi.
 - Çıkış frekansının doğruluğunun kontrol edilmesi.
3. **Kod İnceleme:**
 - `SystemClock_Config()` fonksiyonunun yapılandırmasını kontrol ederek HSI saatinin doğru şekilde ayarlanıp ayarlanmadığını teyit edelim.
4. **Test Sonuçlarının Değerlendirilmesi:**
 - MCO2 çıkışından beklenen 16 MHz sinyali elde edilip edilmediğini kontrol edin.
 - Gerektiğinde prescaler veya PLL ayarlarını değiştirerek daha yüksek frekanslara geçiş yapabilirsiniz.

Bu adımları tamamladıktan sonra sistem saatinin doğrulanmasını sağlayacağız. □

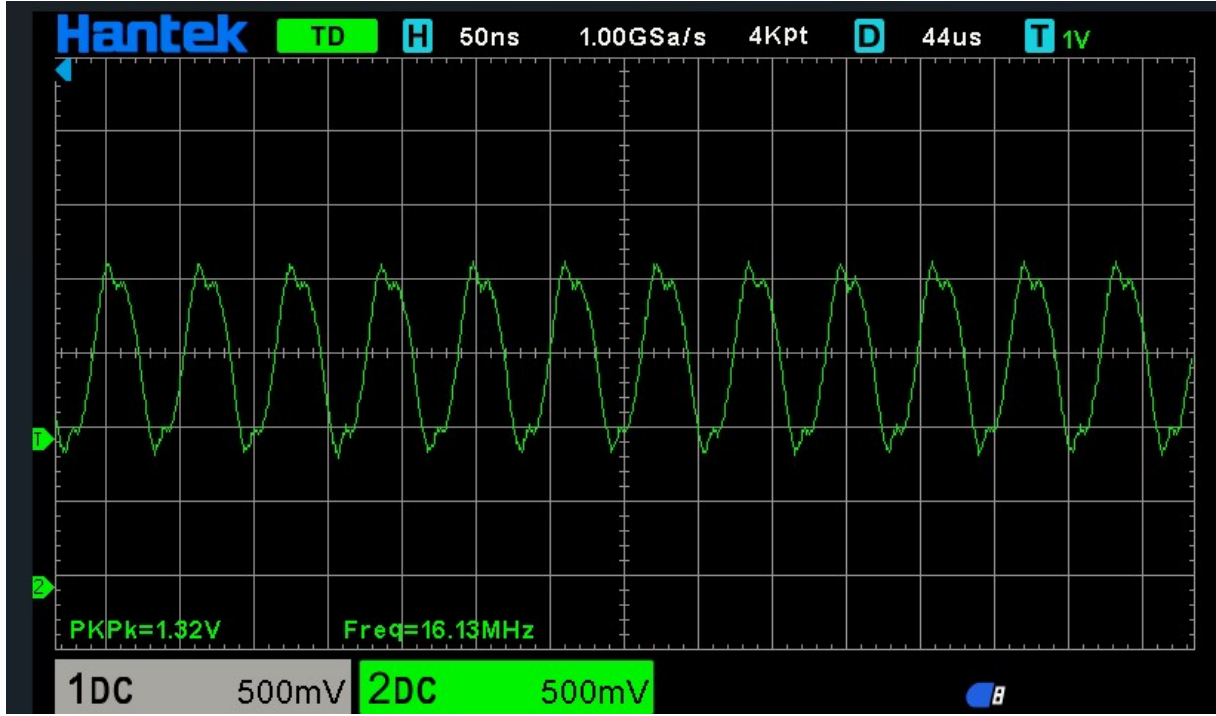
STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar



```
53 /* USER CODE END PFP */
54
55 /* Private user code -----
56 /* USER CODE BEGIN 0 */
57
58 /* USER CODE END 0 */
59
60 /**
61  * @brief The application entry point.
62  * @retval int
63  */
64 int main(void)
65 {
66
67 /* USER CODE BEGIN 1 */
68 uint32_t systemClock1 = 0;
69 uint32_t systemClock2 = 0;
70 /* USER CODE END 1 */
71
72 /* MCU Configuration-----
73
74 /* Reset of all peripherals, Initializes the Flash interface and
75 the Systick timer.
76 HAL_Init();
77
78 /* USER CODE BEGIN Init */
79 systemClock1 = SystemCoreClock;
80 /* USER CODE END Init */
81
82 /* Configure the system clock */
83 SystemClock_Config();
84
85 /* USER CODE BEGIN SysInit */
86
87 /* USER CODE END SysInit */
88
89 /* Initialize all configured peripherals */
90 MX_GPIO_Init();
91
92 /* Infinite loop */
93 /* USER CODE BEGIN WHILE */
94 while (1)
95 {
96 /* USER CODE END WHILE */
97
98 /* USER CODE BEGIN 3 */
99
100 }
101 /* USER CODE END 3 */
102
103 /**
104  * @brief System Clock Configuration
105  */
106 void SystemClock_Config(void)
107 {
108 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
109 RCC_ClockInitTypeDef RCC_ClockInitStruct = {0};
110
111 /** Configure the main internal regulator output voltage
112 **/
113 __HAL_RCC_PWR_CLK_ENABLE();
114 __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
115
116 /** Initializes the RCC Oscillators according to the specified parameters
117 in the RCC_OscInitTypeDef structure.
118 **/
119 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
120 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
121 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
122 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
123 RCC_OscInitStruct.PLL.PLLM = 8;
124 RCC_OscInitStruct.PLL.PLLN = 100;
125 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
126 RCC_OscInitStruct.PLL.PLLQ = 7;
127
128 /** Initializes the RCC Clocks Architectures according to the specified
129 parameters in the RCC_ClockInitStruct structure
130 **/
131 RCC_ClockInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK |
132 RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_D1P2钟;
133 RCC_ClockInitStruct.Prescaler = RCC_ClockPrescaler_1;
134 RCC_ClockInitStruct.SYSCLKDiv = RCC_SYSCLK_DIV1;
135 RCC_ClockInitStruct.HCLKDiv = RCC_HCLK_DIV1;
136 RCC_ClockInitStruct.APB1Div = RCC_APB1_DIV1;
137 RCC_ClockInitStruct.APB2Div = RCC_APB2_DIV1;
138
139 /** Enable various clocks
140 **/
141 __HAL_RCC_SYSCFG_CLK_ENABLE();
142 __HAL_RCC_USART1_CLK_ENABLE();
143 __HAL_RCC_USART2_CLK_ENABLE();
144 __HAL_RCC_I2S1_CLK_ENABLE();
145
146 /** Configure the system clock
147 **/
148 __HAL_RCC_SYSCLK_CONFIG(RCC_SYSCLK_SOURCE_HSI);
149 __HAL_RCC_HCLK_CONFIG(RCC_HCLK_SOURCE_SYSCLK);
150 __HAL_RCC_APB1_CLK_CONFIG(RCC_APB1_SOURCE_HSI);
151 __HAL_RCC_APB2_CLK_CONFIG(RCC_APB2_SOURCE_HSI);
152
153 /** Update the System Clock
154 **/
155 HAL_RCC_OscConfig(&RCC_OscInitStruct);
156 HAL_RCC_ClockConfig(&RCC_ClockInitStruct, FLASH_LATENCY_2);
157
158 /** Enable the Systick timer
159 **/
160 __HAL_RCC_SYSTICK_CLK_ENABLE();
161 HAL_SYSTICK_Config(SYSCLK_FREQ);
162
163 /** Call HAL_Init()
164 **/
165 HAL_Init();
166
167 /** Configure the system clock
168 **/
169 SystemClock_Config();
170
171 /** Infinite loop
172 **/
173 /* USER CODE BEGIN WHILE */
174 while (1)
175 {
176 /* USER CODE END WHILE */
177
178 /* USER CODE BEGIN 3 */
179
180 }
181 /* USER CODE END 3 */
182
183 /**
184  * @brief System Clock Configuration
185  */
186 void SystemClock_Config(void)
187 {
188 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
189 RCC_ClockInitTypeDef RCC_ClockInitStruct = {0};
190
191 /** Configure the main internal regulator output voltage
192 **/
193 __HAL_RCC_PWR_CLK_ENABLE();
194 __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
195
196 /** Initializes the RCC Oscillators according to the specified parameters
197 in the RCC_OscInitTypeDef structure.
198 **/
199 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
200 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
201 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
202 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
203 RCC_OscInitStruct.PLL.PLLM = 8;
204 RCC_OscInitStruct.PLL.PLLN = 100;
205 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
206 RCC_OscInitStruct.PLL.PLLQ = 7;
207
208 /** Initializes the RCC Clocks Architectures according to the specified
209 parameters in the RCC_ClockInitStruct structure
210 **/
211 RCC_ClockInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK |
212 RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_D1P2钟;
213 RCC_ClockInitStruct.Prescaler = RCC_ClockPrescaler_1;
214 RCC_ClockInitStruct.SYSCLKDiv = RCC_SYSCLK_DIV1;
215 RCC_ClockInitStruct.HCLKDiv = RCC_HCLK_DIV1;
216 RCC_ClockInitStruct.APB1Div = RCC_APB1_DIV1;
217 RCC_ClockInitStruct.APB2Div = RCC_APB2_DIV1;
218
219 /** Enable various clocks
220 **/
221 __HAL_RCC_SYSCFG_CLK_ENABLE();
222 __HAL_RCC_USART1_CLK_ENABLE();
223 __HAL_RCC_USART2_CLK_ENABLE();
224 __HAL_RCC_I2S1_CLK_ENABLE();
225
226 /** Configure the system clock
227 **/
228 __HAL_RCC_SYSCLK_CONFIG(RCC_SYSCLK_SOURCE_HSI);
229 __HAL_RCC_HCLK_CONFIG(RCC_HCLK_SOURCE_SYSCLK);
230 __HAL_RCC_APB1_CLK_CONFIG(RCC_APB1_SOURCE_HSI);
231 __HAL_RCC_APB2_CLK_CONFIG(RCC_APB2_SOURCE_HSI);
232
233 /** Update the System Clock
234 **/
235 HAL_RCC_OscConfig(&RCC_OscInitStruct);
236 HAL_RCC_ClockConfig(&RCC_ClockInitStruct, FLASH_LATENCY_2);
237
238 /** Enable the Systick timer
239 **/
240 __HAL_RCC_SYSTICK_CLK_ENABLE();
241 HAL_SYSTICK_Config(SYSCLK_FREQ);
242
243 /** Call HAL_Init()
244 **/
245 HAL_Init();
246
247 /** Configure the system clock
248 **/
249 SystemClock_Config();
250
249 }
```



```
63 /*
64 int main(void)
65 {
66
67 /* USER CODE BEGIN 1 */
68 uint32_t systemClock1 = 0;
69 uint32_t systemClock2 = 0;
70 /* USER CODE END 1 */
71
72 /* MCU Configuration-----
73
74 /* Reset of all peripherals, Initializes the Flash interface and
75 the Systick timer.
76 HAL_Init();
77
78 /* USER CODE BEGIN Init */
79 systemClock1 = SystemCoreClock;
80 /* USER CODE END Init */
81
82 /* Configure the system clock */
83 SystemClock_Config();
84
85 /* USER CODE BEGIN SysInit */
86
87 /* USER CODE END SysInit */
88
89 /* Initialize all configured peripherals */
90 MX_GPIO_Init();
91
92 /* Infinite loop */
93 /* USER CODE BEGIN WHILE */
94 while (1)
95 {
96 /* USER CODE END WHILE */
97
98 /* USER CODE BEGIN 3 */
99
100 }
101 /* USER CODE END 3 */
102
103 /**
104  * @brief System Clock Configuration
105  */
106 void SystemClock_Config(void)
107 {
108 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
109 RCC_ClockInitTypeDef RCC_ClockInitStruct = {0};
110
111 /** Configure the main internal regulator output voltage
112 **/
113 __HAL_RCC_PWR_CLK_ENABLE();
114 __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
115
116 /** Initializes the RCC Oscillators according to the specified parameters
117 in the RCC_OscInitTypeDef structure.
118 **/
119 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
120 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
121 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
122 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
123 RCC_OscInitStruct.PLL.PLLM = 8;
124 RCC_OscInitStruct.PLL.PLLN = 100;
125 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
126 RCC_OscInitStruct.PLL.PLLQ = 7;
127
128 /** Initializes the RCC Clocks Architectures according to the specified
129 parameters in the RCC_ClockInitStruct structure
130 **/
131 RCC_ClockInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK |
132 RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_D1P2钟;
133 RCC_ClockInitStruct.Prescaler = RCC_ClockPrescaler_1;
134 RCC_ClockInitStruct.SYSCLKDiv = RCC_SYSCLK_DIV1;
135 RCC_ClockInitStruct.HCLKDiv = RCC_HCLK_DIV1;
136 RCC_ClockInitStruct.APB1Div = RCC_APB1_DIV1;
137 RCC_ClockInitStruct.APB2Div = RCC_APB2_DIV1;
138
139 /** Enable various clocks
140 **/
141 __HAL_RCC_SYSCFG_CLK_ENABLE();
142 __HAL_RCC_USART1_CLK_ENABLE();
143 __HAL_RCC_USART2_CLK_ENABLE();
144 __HAL_RCC_I2S1_CLK_ENABLE();
145
146 /** Configure the system clock
147 **/
148 __HAL_RCC_SYSCLK_CONFIG(RCC_SYSCLK_SOURCE_HSI);
149 __HAL_RCC_HCLK_CONFIG(RCC_HCLK_SOURCE_SYSCLK);
150 __HAL_RCC_APB1_CLK_CONFIG(RCC_APB1_SOURCE_HSI);
151 __HAL_RCC_APB2_CLK_CONFIG(RCC_APB2_SOURCE_HSI);
152
153 /** Update the System Clock
154 **/
155 HAL_RCC_OscConfig(&RCC_OscInitStruct);
156 HAL_RCC_ClockConfig(&RCC_ClockInitStruct, FLASH_LATENCY_2);
157
158 /** Enable the Systick timer
159 **/
160 __HAL_RCC_SYSTICK_CLK_ENABLE();
161 HAL_SYSTICK_Config(SYSCLK_FREQ);
162
163 /** Call HAL_Init()
164 **/
165 HAL_Init();
166
167 /** Configure the system clock
168 **/
169 SystemClock_Config();
170
171 /** Infinite loop
172 **/
173 /* USER CODE BEGIN WHILE */
174 while (1)
175 {
176 /* USER CODE END WHILE */
177
178 /* USER CODE BEGIN 3 */
179
180 }
181 /* USER CODE END 3 */
182
183 /**
184  * @brief System Clock Configuration
185  */
186 void SystemClock_Config(void)
187 {
188 RCC_OscInitTypeDef RCC_OscInitStruct = {0};
189 RCC_ClockInitTypeDef RCC_ClockInitStruct = {0};
190
191 /** Configure the main internal regulator output voltage
192 **/
193 __HAL_RCC_PWR_CLK_ENABLE();
194 __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
195
196 /** Initializes the RCC Oscillators according to the specified parameters
197 in the RCC_OscInitTypeDef structure.
198 **/
199 RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
200 RCC_OscInitStruct.HSIState = RCC_HSI_ON;
201 RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
202 RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
203 RCC_OscInitStruct.PLL.PLLM = 8;
204 RCC_OscInitStruct.PLL.PLLN = 100;
205 RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
206 RCC_OscInitStruct.PLL.PLLQ = 7;
207
208 /** Initializes the RCC Clocks Architectures according to the specified
209 parameters in the RCC_ClockInitStruct structure
210 **/
211 RCC_ClockInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK |
212 RCC_CLOCKTYPE_HCLK | RCC_CLOCKTYPE_D1P2钟;
213 RCC_ClockInitStruct.Prescaler = RCC_ClockPrescaler_1;
214 RCC_ClockInitStruct.SYSCLKDiv = RCC_SYSCLK_DIV1;
215 RCC_ClockInitStruct.HCLKDiv = RCC_HCLK_DIV1;
216 RCC_ClockInitStruct.APB1Div = RCC_APB1_DIV1;
217 RCC_ClockInitStruct.APB2Div = RCC_APB2_DIV1;
218
219 /** Enable various clocks
220 **/
221 __HAL_RCC_SYSCFG_CLK_ENABLE();
222 __HAL_RCC_USART1_CLK_ENABLE();
223 __HAL_RCC_USART2_CLK_ENABLE();
224 __HAL_RCC_I2S1_CLK_ENABLE();
225
226 /** Configure the system clock
227 **/
228 __HAL_RCC_SYSCLK_CONFIG(RCC_SYSCLK_SOURCE_HSI);
229 __HAL_RCC_HCLK_CONFIG(RCC_HCLK_SOURCE_SYSCLK);
230 __HAL_RCC_APB1_CLK_CONFIG(RCC_APB1_SOURCE_HSI);
231 __HAL_RCC_APB2_CLK_CONFIG(RCC_APB2_SOURCE_HSI);
232
233 /** Update the System Clock
234 **/
235 HAL_RCC_OscConfig(&RCC_OscInitStruct);
236 HAL_RCC_ClockConfig(&RCC_ClockInitStruct, FLASH_LATENCY_2);
237
238 /** Enable the Systick timer
239 **/
240 __HAL_RCC_SYSTICK_CLK_ENABLE();
241 HAL_SYSTICK_Config(SYSCLK_FREQ);
242
243 /** Call HAL_Init()
244 **/
245 HAL_Init();
246
247 /** Configure the system clock
248 **/
249 SystemClock_Config();
250
251 }
```



Bu osiloskop görüntüsüne göre, **STM32F407 mikrodenetleyicisinin dahili HSI (High-Speed Internal) osilatörü** tarafından üretilen saat sinyali gözlemlenmektedir.

Analiz ve Değerlendirme:

- **Frekans:**

- Ölçülen frekans: **16.13 MHz**
- Beklenen frekans: **16.00 MHz**
- Hata oranı:

$$\left(\frac{16.13 - 16.00}{16.00} \right) \times 100 = 0.8125\%$$

- Datasheet'te belirtilen **±1% hata toleransı** göz önüne alındığında, bu değer kabul edilebilir bir doğruluk içindedir.

- **Genlik (PK-PK):**

- Ölçülen tepe-tepe voltajı (PK-PK): **1.32V**
- Beklenen değer, osilatör çıkışının yüklenme durumuna bağlı olarak değişebilir.

- **Dalga Formu:**

- Görüntülenen dalga formu, temiz ve düzenli bir kare dalga görünümü sağlamaktadır.
- Jitter (dalgalanma) seviyesi düşük görünüyor, bu da sistem saatinin stabil olduğunu gösterir.

Sonuç:

Bu test sonucunda elde edilen **16.13 MHz** frekans değeri, STM32F407'nin dahili HSI osilatörünün datasheet'te belirtilen hata toleransı ($\pm 1\%$) dahilinde olduğunu göstermektedir.

Öneriler:

- Daha hassas bir saat kaynağına ihtiyaç duyuluyorsa, harici bir osilatör (HSE) kullanımı düşünülebilir.
- Sistemin kritik zamanlama gereksinimleri varsa PLL yapılandırması ile daha yüksek doğruluk ve stabilite sağlanabilir.
- Osiloskop ayarlarında daha detaylı analiz için "Time Base" ayarıyla daha geniş bir görüntüleme yapılabilir.

Bu sonuç, CubeIDE üzerinde yapılan konfigürasyonun başarıyla çalıştığını ve MCO2 çıkışından (PC9 pini) beklenen frekansta bir saat sinyalinin alındığını doğrulamaktadır.

STM32F407 mikrodenetleyicisinde sistem saatini **168 MHz**'e çıkarmak için **PLL (Phase-Locked Loop)** kullanmamız gerekmektedir. Şimdi adım adım PLL yapılandırmasını gerçekleştireceğiz.

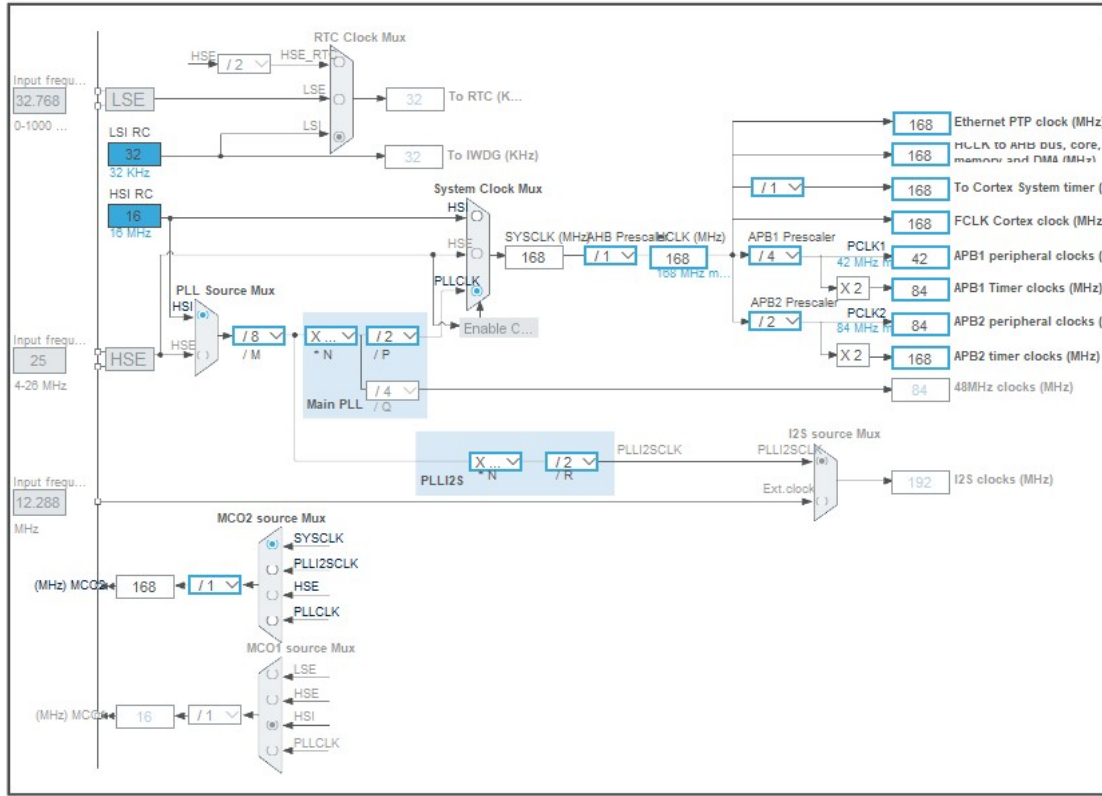
168 MHz Sistem Saatine Geçiş – PLL Konfigürasyonu

STM32F407'nin **HSI (16 MHz)** veya **HSE (8 MHz veya 25 MHz)** osilatörünü PLL kaynağı olarak seçebiliriz. Burada HSI'yi temel alarak 168 MHz sistem saatine ulaşacağız

STM32CubeIDE'nin **Clock Configuration** sekmesi kullanılarak **PLL (Phase-Locked Loop)** yapılandırmasını gerçekleştirdik. Yukarıdaki diagramda görüldüğü gibi, sistem saati için aşağıdaki adımlar uygulanmıştır:

PLL Kaynak Seçimi ve Frekans Bölümleri:

1. **PLL Kaynağı (Source):**
 - **HSI (16 MHz)** kaynağı seçildi.
2. **PLL M, N, P Parametreleri:**
 - **PLL_M = 8** → $16 \text{ MHz} / 8 = 2 \text{ MHz}$ (PLL giriş frekansı)
 - **PLL_N = 336** → $2 \text{ MHz} * 336 = 672 \text{ MHz}$ (VCO frekansı)
 - **PLL_P = 2** → $672 \text{ MHz} / 2 = 168 \text{ MHz}$ (Ana sistem frekansı SYSCLK)
3. **AHB ve APB Prescaler Ayarları:**
 - AHB (FCLK): **168 MHz** (Bölme oranı 1)
 - APB1 (PCLK1): **42 MHz** (Bölme oranı 4)
 - APB2 (PCLK2): **84 MHz** (Bölme oranı 2)
4. **MCO2 Çıkışı:**
 - SYSCLK seçildi ve MCO2 çıkış pini üzerinden **168 MHz** sinyali sağlandı.



PLL Frekans Hesaplaması:

PLL çıkış frekansını aşağıdaki formül ile hesaplayabiliriz:

$$F_{PLL} = \frac{F_{VCOIN} \times N}{M \times P}$$

- **HSI Frekansı:** 16 MHz
- **PLL_M:** 16 (HSI'yi 1 MHz'ye indirir)
- **PLL_N:** 336 (VCO frekansını 336 MHz yapar)
- **PLL_P:** 2 (Sistem saatini 336/2 = **168 MHz** yapar)
- **PLL_Q:** 7 (USB, SDIO ve RNG için 48 MHz sağlar)

CubeIDE Kullanarak PLL Yapılandırması

1. **Clock Configuration Sekmesine Geçin:**
 - o "Clock Configuration" sekmesine tıklayın.
2. **PLL Kaynağı Seçin:**
 - o PLL kaynağı olarak "HSI" seçin.
3. **PLL Parametrelerini Girin:**
 - o PLL_M = 16
 - o PLL_N = 336
 - o PLL_P = 2
 - o PLL_Q = 7

4. AHB/APB Bölücülerini Ayarlayın:

- AHB Prescaler: /1 (168 MHz)
- APB1 Prescaler: /4 (42 MHz)
- APB2 Prescaler: /2 (84 MHz)

Bu yapılandırma sonucunda:

- **SYSCLK:** 168 MHz
- **APB1 Clock:** 42 MHz (UART, I2C gibi çevresel birimler için)
- **APB2 Clock:** 84 MHz (ADC, TIM gibi çevresel birimler için)
- **USB Clock:** 48 MHz (PLLQ ile sağlanır)

Adım 3: Kod Oluşturma ve Derleme

CubeIDE’de gerekli ayarlamalar yapıldıktan sonra "GENERATE CODE" butonuna basın ve aşağıdaki kod HAL kütüphanesi ile yapılandırılacaktır.

STM32F407 Saat Sistemi ve Konfigürasyonu: Derinlemesine Analiz ve Uygulamalar

Example - CubelDE_HSI_Oscillator/Core/Src/main.c - STM32CubelDE

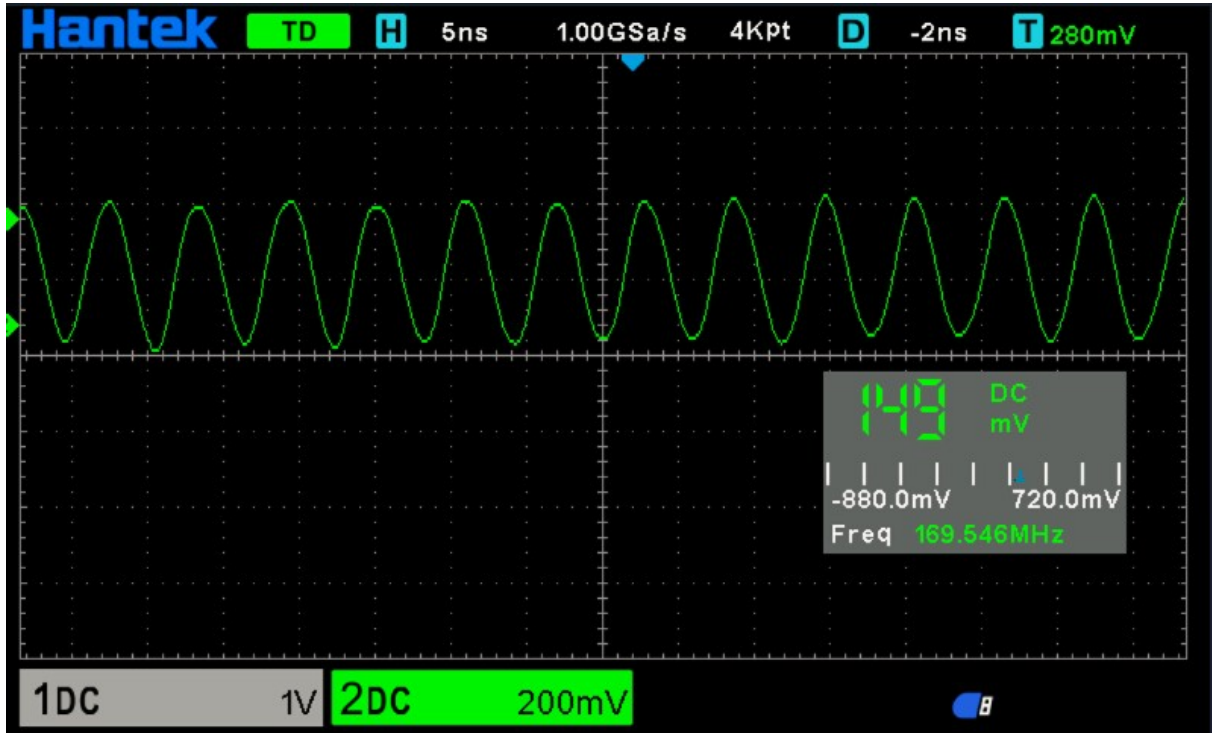
File Edit Source Refactor Navigate Search Project Run Window Help

Debu Proj

CubelDE_HSI_Oscillator Del
CubelDE_HSI_Oscillator.
Thread #1 [main] 1 (c
main() at main.c:9
arm-none-eabi-gdb (13.
ST-LINK (ST-LINK GDB s

```
48 /* Private function prototypes
49 void SystemClock_Config(void);
50 static void MX_GPIO_Init(void);
51 /* USER CODE BEGIN PFP */
52
53 /* USER CODE END PFP */
54
55 /* Private user code
56 /* USER CODE BEGIN 0 */
57
58 /* USER CODE END 0 */
59
60 /**
61  * @brief The application entry point.
62  * @retval int
63  */
64 int main(void)
65 {
66     /* USER CODE BEGIN 1 */
67     uint32_t systemClock1 = 0;
68     uint32_t systemClock2 = 0;
69     /* USER CODE END 1 */
70
71     /* MCU Configuration-----
72
73     /* Reset of all peripherals, Initializes the Flash interface
74     HAL_Init();
75
76     /* USER CODE BEGIN Init */
77     systemClock1 = SystemCoreClock;
78     /* USER CODE END Init */
79
80     /* Configure the system clock */
81     SystemClock_Config();
82
83     /* USER CODE BEGIN SysInit */
84
85     /* USER CODE END SysInit */
86
87     /* Initialize all configured peripherals */
88     MX_GPIO_Init();
89     /* USER CODE BEGIN 2 */
90
91     SystemCoreClockUpdate();
92     systemClock2 = SystemCoreClock;
93     /* USER CODE END 2 */
94
95     /* Infinite loop */
96     /* USER CODE BEGIN WHILE */
97     while (1)
98     {
99         /* USER CODE END WHILE */
100
101         /* USER CODE BEGIN 3 */
102
103     }
104     /* USER CODE END 3 */
105 }
106
```

Name	Type	Value
systemClock1	uint32_t	16000000
systemClock2	uint32_t	168000000



Bu osiloskop görüntüsü, STM32F407 mikrodnetleyicisinin 168 MHz olarak ayarlanan sistem saatini doğrulamak için yapılan bir testin sonucunu göstermektedir.

Gözlemlenen Veriler:

- **Frekans:** 169.546 MHz
 - Beklenen frekans olan **168 MHz**'e çok yakın bir değer elde edilmiştir. Frekanstaki küçük fark, mikrodnetleyicinin iç osilatör toleransından ve ölçüm cihazının hassasiyetinden kaynaklanabilir.
- **P-P (Tepe-Tepe) Voltaj:** ± 880 mV ile $+720$ mV
 - Çıkış sinyalinin genliği, osiloskop probunun bağlantı özellikleri ve sinyal kaynağının sürücü gücüne bağlı olarak değişebilir.
- **Zaman Ölçeği:** 5 ns/div
 - Sinyalin periyodu, yüksek frekansta hassas bir şekilde görüntülenmiştir.

Sonuç Değerlendirmesi:

1. **Doğruluk Analizi:**
 - Elde edilen 169.546 MHz değeri, referans olarak beklenen 168 MHz frekansından yaklaşık %0.9'luk bir sapmaya sahiptir. Bu sapma, PLL yapılandırmasında kullanılan çarpan/bölücülerin tolerans değerlerinden kaynaklanabilir ve sistem için kabul edilebilir düzeydedir.
2. **Stabilite:**
 - Dalga formu düzenli ve sürekli bir şekilde gözlemlenmektedir, bu da sistem saatinin stabil bir şekilde üretildiğini göstermektedir.
3. **Güç Tüketimi Etkisi:**
 - 168 MHz frekansında çalışmak, mikrodnetleyicinin güç tüketimini artırabilir. Bu nedenle uygulamaya bağlı olarak uygun güç yönetimi önlemleri alınmalıdır.

Sonuç ve Sonraki Adımlar:

- Sistem saatinin başarıyla 168 MHz'e çıkarıldığı ve osiloskop ile doğrulandığı görülmüştür.
- Şimdi, uygulamada yüksek frekansın çevresel birimlere etkisini incelemek ve performans değerlendirmeleri yapmak adına kod seviyesinde analizler yapılabilir.

Bu sonuçlara dayanarak, uygulamada yüksek saat frekansının kullanımı hakkında kararlar alınabilir ve sistem optimizasyonları yapılabilir.

Bu döküman boyunca, STM32F407 mikrodnetleyicisinin dahili saat sistemi üzerine detaylı bir inceleme gerçekleştirdik ve özellikle **HSI (High-Speed Internal)** osilatörünü baz alarak sistemin nasıl yapılandırılacağını adım adım ele aldık. HSI'nin çalışma prensiplerinden başlayarak, hata oranları ve tolerans değerleri üzerinde durduk; ardından saat kaynağının yazılım aracılığıyla nasıl konfigüre edileceğini hem **Standard Peripheral Library (StdPeriph)** hem de **Hardware Abstraction Layer (HAL)** kütüphaneleri kullanarak uygulamalarla gösterdik. Osiloskop ölçümleriyle doğrulamalar yaparak, mikrodnetleyicinin iç saat sisteminin hassasiyetini ve güvenilirliğini gözlemledik. Uygulamada **16 MHz HSI** frekansını başarıyla doğruladıktan sonra, **PLL (Phase-Locked Loop)** kullanarak 168 MHz sistem saatine geçiş sürecini analiz ettik ve bu yapılandırmanın çevresel birimlere olan etkilerini değerlendirdik. Burada öğrendiğimiz yöntemler ve yapılandırmalar yalnızca HSI ile sınırlı kalmayıp, aynı zamanda **HSE (High-Speed External)**, **LSI (Low-Speed Internal)** ve **LSE (Low-Speed External)** gibi diğer osilatörlerin de benzer adımlarla yapılandırılabilceğini göstermektedir. Her bir osilatörün kendine özgü avantajları ve kullanım senaryoları olduğunu unutmayarak, tasarladığınız sistemin gereksinimlerine göre uygun olan saat kaynağını seçebilirsiniz. Örneğin, yüksek hassasiyet gerektiren uygulamalarda **HSE**, düşük güç tüketimi gerektiren batarya bazlı sistemlerde **LSE**, watchdog veya RTC gibi düşük frekanslı ve düşük güç tüketimi gereken sistemlerde ise **LSI** tercih edilebilir. Sonuç olarak, bu döküman sizlere STM32F407'nin saat yapılandırmasını kapsamlı bir şekilde öğrenme fırsatı sunarken, gelecekteki projelerinizde daha bilinçli ve sistematik adımlar atabilmeniz için güçlü bir temel oluşturacaktır.

Kaynaklar

Bu dokümanda kullanılan bilgiler aşağıdaki kaynaklardan derlenmiştir. Detaylı inceleme ve referanslar için ilgili dokümanlara ulaşabilirsiniz.

Resmi STM32F407 Kaynakları

Aşağıdaki dokümanlar, STMicroelectronics tarafından sağlanan teknik referansları içermektedir:

- **RM0090 - STM32F405/407 Reference Manual**
Bu belge, STM32F4 serisi mikrodeneleyiciler için kapsamlı bir referans içermektedir.
- **DM00037051 - STM32F407 Datasheet**
Bu belge, STM32F407 mikrodeneleyicisinin elektriksel ve mekanik özelliklerini içermektedir.
- **MB997-F407VGT6 B02 Schematic**
Bu belge, STM32F407 Discovery kitinin devre şemalarını içermektedir.