

AN41908 Driver API

Driver AN41908.c ve AN41908.h dosyalarından oluşuyor. Üst seviye yazılımı ilgilendiren fonksiyonlar ve açıklamaları şunlar:

int initDriver(char* opt) ;

Tek bir opsiyon stringi alan bu fonksiyon öncelikle SPI konfigüre eder. Başarılı olursa, sırasıyla iris ve focus&zoom register'larını konfigüre eder. Bu register tekrar okunarak değerleri teyit edilir.

Tek argüman olarak "init_pos" girilirse, iris 200 konumuna, focus ve zoom ise 0 konumlarına çekilir. Bunun dışında herhangi string girilirse, konumlar değiştirilmez.

Başarılı işlem sonrası 0, aksi takdirde -1 döner.

int setIris(int VAL) ;

Girilen mutlak değere diyafram açıklığı ayarlanır. Bu değer 0 (en açık, aydınlık) – 1023 (en kapalı, karanlık) arasında olmalıdır. Bu değer kontrol edilir, aralık dışında ise uyarı verilir ancak program durmaz. Başarılı işlem sonrası 0, aksi takdirde -1 döner.

int getIris() ;

En güncel iris değerini verir. Ancak donanımdan direkt olarak değer okuma imkanı olmadığından, açılışta her zaman 0 değeri döner. Yazılım tarafından ilk setIris sonrası doğru değer alır.

int setFZ(int SPD, int POS, char* mode, int CCWCW, int VD, int numVD) ;

Girilen değerler doğrultusunda focus veya zoom motorları, belirlenen hız, yön ve periyotlarda istenilen pozisyona getirilir. Pozisyon okuma imkanı olmadığından açılışta initDriver("init_pos") veya setFZ(-,-,"i",-,-) komutları ile sıfır konumuna getirme işlemi yapılmalıdır.

Kısaca çalışma prensibi şöyle, VD_FZ sinyali ile hareketi periyotlara ayırıyoruz. Önerilen; bu VD'lerin frame rate ile senkron olması böylece her karede bir değişim görebiliriz. Bunu sağlamak için normal modda 4.argüman FPS girilmelidir. Her ardışık VD_FZ sinyali arasında PSUM kadar adım atılıyor ve yön CCWCW ile belirleniyor. Ben yaptığım testlerde zoom için yaklaşık 15000, focus için yaklaşık 10000 adım olduğu sonucuna vardım. Diğer önemli register ise INTCT. Bu da tek bir adımın süresini belirliyor. Normal modda elimizde VD ve PSUM değerleri olacağı için INTCT otomatik hesaplanır.

Ancak test modlarında PSUM ve INTCT yazılım girer, bu kez de VD frekansı içeride hesaplanır. Son argüman #periyot ise kaç defa VD aralığı yollayacağımız belirler. Diğer bir deyişle kaç tane PSUM kadar adım atacağımızı. PSUM 0 ile 255 arası değer alır, uzun mesafeler gitmek için birden fazla VD sinyali yollamak gerekir. **Test modlarının amacı aradaki formülasyonları atlayıp direkt register'lara değer yazabilmektir.**

Mod	ARG1	ARG2	ARG3	ARG4	ARG5	ARG6
Focus (Normal)	Hız	Pozisyon	"f" veya "F"	Frame rate	-	-
Zoom (Normal)	Hız	Pozisyon	"z" veya "Z"	Frame rate	-	-
Focus (Test)	PSUM	INTCT	"tf" veya "TF"	-	CCWCW	#Periyot
Zoom (Test)	PSUM	INTCT	"tz" veya "TZ"	-	CCWCW	#Periyot
Init	-	-	"i" veya "I"	-	-	-

Normal modlarda 1-7 arasında bir hız belirteci alınır. Aşağıdaki tabloya göre bir PSUM değeri belirlenir. İlk 6 hız için bir PSUM değeri belirlenir. Girilen frame rate'e göre INTCT içeride hesaplanır. Girilen pozisyona göre de kaç frame gerektiği (#periyot) ve yön de içeride belirlenir.

Son olarak, girilen hedef pozisyona, girilen hız ile gitmeye çalışırız. Daha sonra hız kademe kademe düşürülerek tam hedefe ulaşılır. Örneğin hızımız 4 ve hedef 100 adım ileride olsun. Hareket için 1 periyot 64 adım, 1 periyot 32 adım ve 4 periyot 1'er adım atılır.

Hız	Adım Sayısı (PSUM)	Adım Süresi (INTCT)	Frekans
1	1	Girilen FPS ile hesaplanır	FPS ile senkron
2	8	Girilen FPS ile hesaplanır	FPS ile senkron
3	32	Girilen FPS ile hesaplanır	FPS ile senkron
4	64	Girilen FPS ile hesaplanır	FPS ile senkron
5	128	Girilen FPS ile hesaplanır	FPS ile senkron
6	255	Girilen FPS ile hesaplanır	FPS ile senkron
7	255	110	40 Hz

```
int getFZ(char* mode) ;
```

Girilen moda göre ("f","F","z","Z") ilgili güncel değeri verir. Ancak donanımdan direkt olarak değer okuma imkanı olmadığından, açılışta her zaman 0 değeri döner. Bu nedenle, yazılım tarafından açılışta initDriver("init_pos") veya setFZ(-,-,"i",-,-,-) komutları ile sıfır konumuna getirme işlemi yapılmalıdır.

Örnek Kullanım

```
initDriver("no_init_pos"); // Her şey konfigüre edilir (sıfırlama yok)
setIris(350); // Iris 350 değerine çekildi.
setFZ(-1,-1,"i",-1,-1,-1); // Focus&zoom sıfır konumlarına çekildi.
setFZ(4,1000,"f",30,-1,-1); // Focus normal modda 1000 konumuna gitti
setFZ(7,2500,"z",30,-1,-1); // Zoom asenkron şekilde 2500'e gitti
printf("I:%d F:%d Z:%d\n", getIris(), getFZ("f"),getFZ("z"));
setFZ(128,200,"Z",-1,0,15); // Zoom ileri yönde, 15 adet 128 lik
adımlar attı. Bir adımın periyodu 200 değerinden hesaplandı.
printf("I:%d F:%d Z:%d\n", getIris(), getFZ("f"),getFZ("z"));
```

Ayrıca terminalde direkt "motor_driver" binary dosyasını Hisilicon üzerinde çalıştırabilirsiniz. Bir adet initDriver("no_init_pos") ve bir setFZ fonksiyonu çalıştırır. Argümanları terminalde girmeniz gerekir:

```
>>chmod +x ./motor_driver
>>./motor_driver ARG1 ARG2 ARG3 ARG4 ARG5 ARG6
```