

Mikrodenetleyiciler Üzerinde Sensör Sürücüsü Geliştirme ve Data-sheet Okuma

Ceyhun Şen

Sunum İçeriği

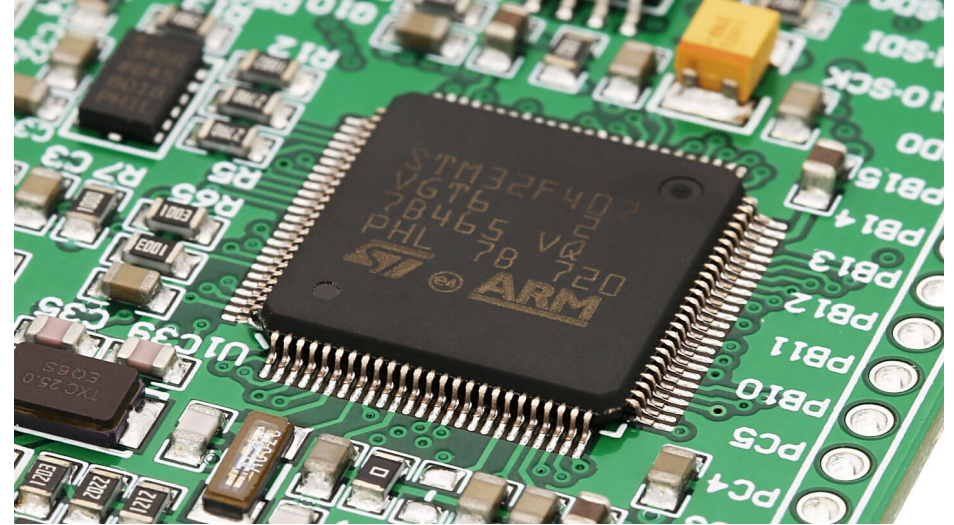
- Ön Bilgilendirme
 - Mikrodenetleyiciler ve Sensörler
 - Aygıt Sürücüsü Nedir?
 - Register Nedir?
- Data-sheet
 - Data-sheet Nedir?
 - Sensörün Özelliklerinin Anlaşılması
 - Sensör Register'ları
- Aygıt Sürücüsü Yazılımı
 - Genel Sürücü Yapısı
 - Sürücü Arayüzleri
 - Sürücünün Yazılması
- Kaynakça



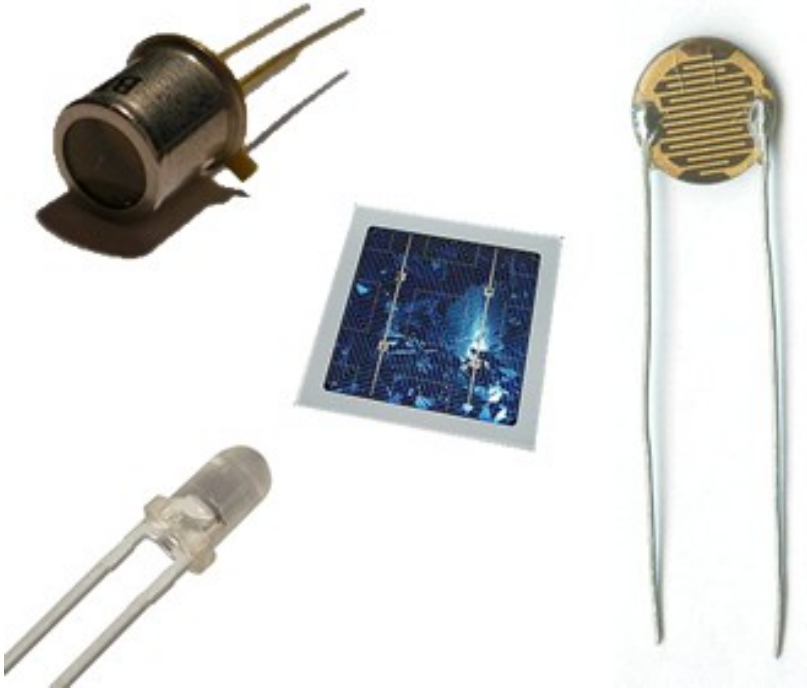
Ön Bilgilendirme

Mikrodenetleyiciler ve Sensörler (1/2)

Mikrodenetleyiciler içerisinde en az 1 tane CPU, hafıza ve programlanabilir giriş/çıkış birimleri bulunan entegre devrelerdir.



Mikrodenetleyiciler ve Sensörler (2/2)



Sensörler fiziksel bir değişimi algılayıp karşılığında sinyal üreten ve bunu başka bir elektronik cihaza aktaran cihazlardır.

Aygıt Sürücüsü Nedir?

Aygıt sürücülerini bir cihazın bilgisayar tarafından kontrol edilmesine imkan tanıyan bilgisayar programlarıdır. Sürücüler, bilgisayar veri yollarını kullanarak cihazlarla haberleşir.

Register Nedir?

- Register, işlemcinin üzerinde bulunan hızlı erişilebilir hafızadır.
- Donanım üzerinde değişiklik yapabilirler.
- Register'lar ana hafızanın başlangıcında olabilir.



Data-sheet

Data-sheet Nedir?

Data-sheet bir ürünün özelliklerini özetleyen dokümanlardır. Data-sheet'ler genellikle ürünün üreticisi tarafından yazılır ve ürünü kullanacak mühendise ürün hakkında teknik bilgiler sağlar.

Sensörün Özelliklerinin Anlaşılması (1/2)

- Bir data-sheet'in ilk bölümünde genellikle ürün ile alakalı teknik bilgiler bulunur.
- Ürünün özelliklerine bağlı olarak yazılacak sürücünün yapısı ve özellikleri planlanabilir.

Sensörün Özelliklerinin Anlaşılması (2/2)

MPU-9255 hakkında çıkarılan bazı sonuçlar:

- İvmeölçer, jiroskop, manyetometre ve termometreye sahip.
- Ölçeklenebilir ometreleri, ayarlanabilir kesmeleri ve başkaları seviye özellikleri var.
- I2C ve SPI arayüzleri ile haberleşebiliyor.

	MPU-9255 Product Specification	Document Number: DS-000007 Revision: 1.0 Release Date: 09/14/2014
---	---------------------------------------	---

1.2 Purpose and Scope

This document is a preliminary product specification, providing a description, specifications, and design related information on the MPU-9255™ MotionTracking device. The device is housed in a small 3x3x1mm QFN package.

Specifications are subject to change without notice. Final specifications will be updated based upon characterization of production silicon. For references to register map and descriptions of individual registers, please refer to the MPU-9255 Register Map and Register Descriptions document.

1.3 Product Overview

MPU-9255 is a multi-chip module (MCM) consisting of two dies integrated into a single QFN package. One die houses the 3-Axis gyroscope and the 3-Axis accelerometer. The other die houses the AK8963 3-Axis magnetometer from Asahi Kasei Microdevices Corporation. Hence, the MPU-9255 is a 9-axis MotionTracking device that combines a 3-axis gyroscope, 3-axis accelerometer, 3-axis magnetometer and a Digital Motion Processor™ (DMP) all in a small 3x3x1mm package available as a pin-compatible upgrade from the MPU-6515. With its dedicated I²C sensor bus, the MPU-9255 directly provides complete 9-axis MotionFusion™ output. The MPU-9255 MotionTracking device, with its 9-axis integration, on-chip MotionFusion™, and run-time calibration firmware, enables manufacturers to eliminate the costly and complex selection, qualification, and system level integration of discrete devices, guaranteeing optimal motion performance for consumers. MPU-9255 is also designed to interface with multiple non-inertial digital sensors, such as pressure sensors, on its auxiliary I²C port.

MPU-9255 features three 16-bit analog-to-digital converters (ADCs) for digitizing the gyroscope outputs, three 16-bit ADCs for digitizing the accelerometer outputs, and three 16-bit ADCs for digitizing the magnetometer outputs. For precision tracking of both fast and slow motions, the parts feature a user-programmable gyroscope full-scale range of ± 250 , ± 500 , ± 1000 , and ± 2000 /sec (dps), a user-programmable accelerometer full-scale range of $\pm 2g$, $\pm 4g$, $\pm 8g$, and $\pm 16g$, and a magnetometer full-scale range of $\pm 4800\mu T$.

MPU-9255 data-sheet, sayfa 5


Sensör Register'ları (1/2)

- Bazı sensörlerin üzerinde programlanabilir register'lar bulunur.
- Bu register'lara haberleşme arayüzleri ile erişilebilir.
- Sensör register'ları data-sheet'ler üzerinde tanımlanır.

Sensör Register'ları (2/2)

MPU-9255 hakkında çıkarılan bazı sonuçlar:

- 126 farklı register var.
- Bazı register'lar veri okumak için, geri kalanlar ise sensörü ayarlamak için var.
- Sensörün güç tüketimi, offset değerleri, filtre ve diğer ayarları değiştirilebilir.

	MPU-9255 Register Map and Descriptions	Document Number: RM-000008 Revision: 1.0 Release Date: 10/10/2014
---	--	---

3 Register Map for Gyroscope and Accelerometer

The following table lists the register map for the gyroscope and accelerometer in the MPU-9255 MotionTracking device.

Addr (Hex)	Addr (Dec.)	Register Name	Serial I/F	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0		
00	0	SELF_TEST_X_GYRO	R/W				xg_st_data [7:0]						
01	1	SELF_TEST_Y_GYRO	R/W				yg_st_data [7:0]						
02	2	SELF_TEST_Z_GYRO	R/W				zg_st_data [7:0]						
0D	13	SELF_TEST_X_ACCEL	R/W				XA_ST_DATA [7:0]						
0E	14	SELF_TEST_Y_ACCEL	R/W				YA_ST_DATA [7:0]						
0F	15	SELF_TEST_Z_ACCEL	R/W				ZA_ST_DATA [7:0]						
13	19	XG_OFFSET_H	R/W				X_OFFSET_USR [15:8]						
14	20	XG_OFFSET_L	R/W				X_OFFSET_USR [7:0]						
15	21	YG_OFFSET_H	R/W				Y_OFFSET_USR [15:8]						
16	22	YG_OFFSET_L	R/W				Y_OFFSET_USR [7:0]						
17	23	ZG_OFFSET_H	R/W				Z_OFFSET_USR [15:8]						
18	24	ZG_OFFSET_L	R/W				Z_OFFSET_USR [7:0]						
19	25	SMP_LRT_DIV	R/W				SMP_LRT_DIV [7:0]						
1A	26	CONFIG	R/W	-	FIFO MODE	EXT_SYNC_SET [2:0]			DLPF_CFG [2:0]				
1B	27	GYRO_CONFIG	R/W	XGYRO_Ct en	YGYRO_Ct en	ZGYRO_Ct en	GYRO_FS_SEL [1:0]		-	FCHOICE_B [1:0]			
1C	28	ACCEL_CONFIG	R/W	ax_st_en	ay_st_en	az_st_en	ACCEL_FS_SEL [1:0]		-	-			
1D	29	ACCEL_CONFIG 2	R/W	-				ACCEL_FCHOICE_B		A_DLPF_CFG			
1E	30	LP_ACCEL_ODR	R/W	-				-		Lpacc_ciksel [3:0]			
1F	31	WOM_THR	R/W	WOM_Threshold [7:0]									
23	35	FIFO_EN	R/W	TEMP_FIFO_EN	GYRO_XO_UT	GYRO_YO_UT	GYRO_ZO_UT	ACCEL	SLV2	SLV1	SLV0		

MPU-9255 register map
data-sheet, sayfa 7



Aygıt Sürücüsü Yazılımı

Genel Sürücü Yapısı

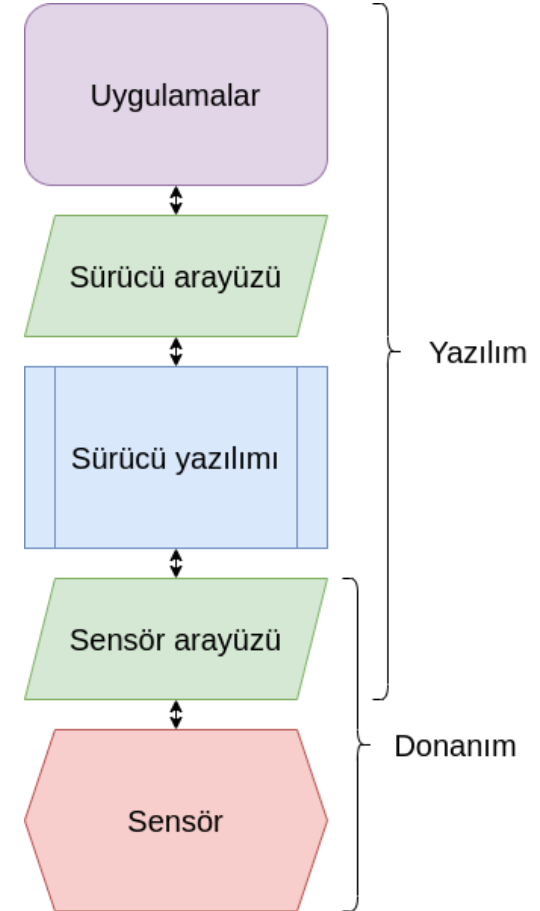
Bir sensör sürücüsünde:

- İlk ayarlamaları yapan,
- Daha sonrasında isteğe bağlı ayar değiştiren,
- Sensör verilerini okuyan

fonksiyonlar bulunabilir. Farklı sensörler ile bu maddelere yenileri eklenebilir veya bazı maddeler çıkartılabilir.

Sürücü Arayüzleri

- Sensör haberleşme arayüzü ile mikrodenetleyiciye veri gönderir/alır.
- Sensör haberleşme arayüzü donanımsaldır ve mikrodenetleyiciden mikrodenetleyiciye göre değişir.
- Sürücü kendi arayüzü ile kullanıcı uygulamasına veri gönderir/alır.
- Sürücü arayüzü yazılımsaldır ve mikrodenetleyiciden mikrodenetleyiciye değişmez.



Sürücünün Yazılması (1/4)

- Sürücüler genellikle düşük seviyeli dillerle yazılır. C, C++ veya Rust buna örnek olarak verilebilir.
- Donanımsal arayüzlere erişimi olan kütüphaneler ile yüksek seviyeli diller de kullanılabilir. Raspberry Pi platformu üzerinde Python buna örnek verilebilir.
- Ancak sensörlerin kullanıldığı sistemlerin ihtiyaçlarından dolayı düşük seviyeli diller daha çok tercih edilir.

Sürücünün Yazılması (2/4)

- Sürücü sensör ile haberleşebilmelidir.
- Sürücü sensörü istenilen şekilde ayarlayabilmelidir.
- Sürücü yapılan ayar değişikliklerini hatırlamalıdır.
- Sürücü sensörden veri okuyabilmelidir.
- Bir mikrodenetleyiciye birden fazla sensör takılabileceğinden sürücü bunu desteklemelidir.

Sürücünün Yazılması (3/4)

Sürücü ayarları sensörün özellikleri göz önünde bulundurularak, dile özgü veri yapıları ile saklanılabilir. Bu veri yapısına C dilinde struct örnek verilebilir.

```
138 /**
139  * @struct mpu925x_t mpu925x.h mpu925x.h
140  * @brief Main struct for MPU-925X driver.
141  * This struct includes sensor data, driver settings and master specific
142  * handle data, bus and delay function pointers.
143  */
144 typedef struct mpu925x_t {
145     /**
146      * @struct sensor_data
147      * @brief Holds sensor data.
148      */
149     struct sensor_data {
150         int16_t acceleration_raw[3], rotation_raw[3], magnet_raw[3], temperature_raw;
151         float acceleration[3], rotation[3], magnetic_field[3], temperature;
152     } sensor_data;
153
154     /**
155      * @struct settings
156      * @brief Holds sensor settings.
157      */
158     struct settings {
159         mpu925x_orientation orientation;
160         mpu925x_accelerometer_scale accelerometer_scale;
161         mpu925x_gyroscope_scale gyroscope_scale;
162         mpu925x_magnetometer_measurement_mode measurement_mode;
163         mpu925x_magnetometer_bit_mode bit_mode;
164         float acceleration_lsb, gyroscope_lsb, magnetometer_lsb, temperature_lsb;
165         float magnetometer_coefficient[3];
166         uint8_t address;
167     } settings;
168
169     /**
170      * @struct master_specific
171      * @brief Holds master specific pointers.
172      */
173     struct master_specific {
174         uint8_t (*bus_read)(struct mpu925x_t *mpu925x, uint8_t slave_address, uint8_t reg, uint8_t *buffer, uint8_t size);
175         uint8_t (*bus_write)(struct mpu925x_t *mpu925x, uint8_t slave_address, uint8_t reg, uint8_t *buffer, uint8_t size);
176         void (*delay_ms)(struct mpu925x_t *mpu925x, uint32_t delay);
177         void *bus_handle;
178     } master_specific;
179 } mpu925x_t;
```

Sürücünün Yazılması (4/4)

```
181 // Core
182 uint8_t mpu925x_init(mpu925x_t *mpu925x, uint8_t ad0);
183
184 // Sensor data
185 void mpu925x_get_all_raw(mpu925x_t *mpu925x);
186 void mpu925x_get_all(mpu925x_t *mpu925x);
187 void mpu925x_get_acceleration_raw(mpu925x_t *mpu925x);
188 void mpu925x_get_acceleration(mpu925x_t *mpu925x);
189 void mpu925x_get_rotation_raw(mpu925x_t *mpu925x);
190 void mpu925x_get_rotation(mpu925x_t *mpu925x);
191 void mpu925x_get_magnetic_field_raw(mpu925x_t *mpu925x);
192 void mpu925x_get_magnetic_field(mpu925x_t *mpu925x);
193 void mpu925x_get_temperature_raw(mpu925x_t *mpu925x);
194 void mpu925x_get_temperature(mpu925x_t *mpu925x);
195
196 // General settings
197 void mpu925x_set_sample_rate_divider(mpu925x_t *mpu925x, uint8_t sample_rate_divider);
198 void mpu925x_set_clock_source(mpu925x_t *mpu925x, mpu925x_clock clock);
```

MPU-925X sürücüsü, fonksiyon imzaları

- Sensörün ilk ayarını ve daha sonrasında da diğer ayarlarını değiştirebilen fonksiyonlar tanımlanabilir.
- Bu fonksiyonlara, sürücüye özel olarak oluşturulan veri yapısı parametre olarak verilerek sürücünün eski ayarları hatırlaması ve yeni ayarları kaydetmesi sağlanabilir.
- Bu fonksiyonlar, data-sheet'te belirtilen adreslere, yine data-sheet'te anlatılan mantık ile veri yazar veya okur.

Kaynakça

- <https://github.com/ceyhunsen/mpu925x-driver>
- <https://en.wikipedia.org/wiki/Microcontroller>
- <https://en.wikipedia.org/wiki/Sensor>
- https://en.wikipedia.org/wiki/Data_sheet
- https://en.wikipedia.org/wiki/Device_driver
- https://en.wikipedia.org/wiki/Processor_register
- https://en.wikipedia.org/wiki/Sensor_fusion
- <https://web.stanford.edu/class/ee267/misc/MPU-9255-Datasheet.pdf>
- <https://web.stanford.edu/class/ee267/misc/MPU-9255-Register-Map.pdf>