

MAKÜ|GUBYO

GÖLHİSAR UYGULAMALI BİLİMLER YÜKSEKOKULU



MİKROİŞLEMCİLER UYGULAMA FİNAL ÖDEV RAPORU

HAZIRLAYAN:

Öğrencinin Adı : Elif ALTINTAŞ

No : 1912901001

Bölüm : Bilişim Sistemleri ve Teknolojileri 4 / A

Ders : Mikroişlemciler Uygulama

1) DENEYİN ADI

Deney : OLED Modul ve Tilt sensörü uygulaması

Senaryo: Tilt sensörden okunan değer “% urun: int” cinsinden, OLED ekranına yazdırılır. Tilt sensörü hareket ettikçe OLED ekrana hareket sayısı yazdırılır aynı zamanda saymaya başlayınca yeşil led yanar. Sayaç 6 ya geldiğinde OLED ekranda “SINIR!!” şeklinde uyarı verir aynı zamanda kırmızı led yanar buzzer çalmaya başlar.

KAZANIMLAR

- ✓ OLED Module kullanımı
- ✓ OLED ekran kontrolü
- ✓ I2C Haberleşme Protokolü ve fonksiyonları
- ✓ Tilt Sensörü ve Kullanımı

2) DENEYDE KULLANILACAK MALZEMELER

- PinoLab-CodeBoard.Micro
- PinARM-STM32F103C8T6
- OLED Module
- Tilt Sensörü
- Bağlantı kablosu

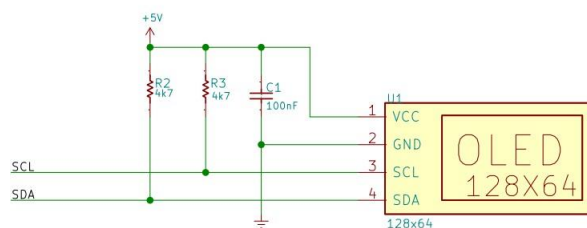
3) GENEL BİLGİ

OLED Module

Şekil 1’de OLED Module ve Şekil 2’de OLED Module açık devre şeması ve pin bağlantıları verilmiştir. Programda, Şekil 2’de gösterilen SCL (PB6) veya SDA (PB7) pini kullanılmalıdır.



Şekil 1. OLED Module



Şekil 2. OLED Module açık devre şeması ve pin bağlantıları

OLED Ekran

Şekil 3’te 128x64 0.96 inch OLED Display Ekran verilmiştir.



Şekil 3. 128x64 0.96 inch OLED Display Ekran

OLEDler 128x164, 128x64 piksel olmak üzere 2 boyut, tek renk ve iki renk olacak şekilde seçenekleri vardır. Bu uygulamada 128x64 piksel 0.96 inch OLED Display Ekran kullanılmıştır.

I2C Haberleşme Protokolü

I2C (Inter-Integrated Circuit) protokolü, seri iletişim için kullanılan bir protokoldür. Master-slave yapısına dayanır ve veri iletişimi için SDA ve SCL hatlarını kullanır.

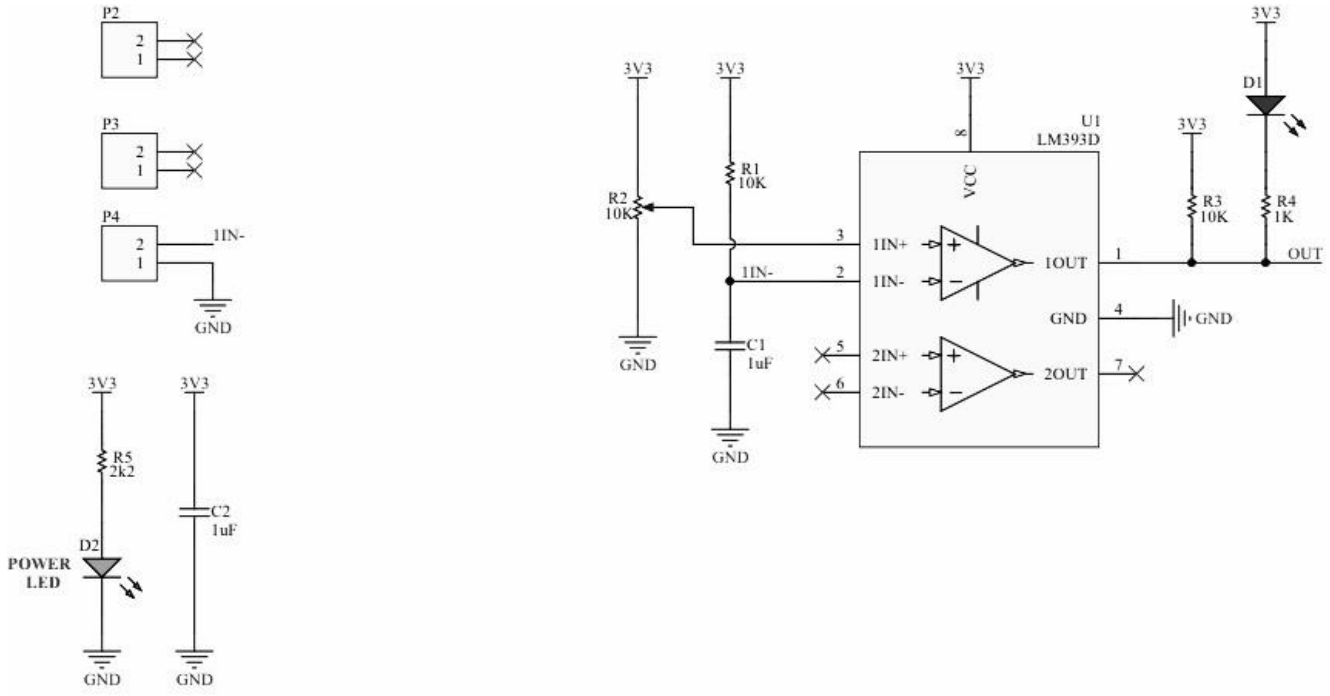
TILT Module

Oryantasyon veya eğimi tespit etmek amacıyla kullanılan sensörlere Tilt (Eğim) Sensörü denir. Tilt sensörleri düşük maliyetli bir elektronik devre bileşenidir. Nesnelerin yönünü belirlemede de kullanılabilir. Güç tüketimi düşük olan Tilt (Eğim) sensörler uzun süre bozulmadan kullanılabilir.

Şekil 1’de TILT Module ve Şekil 2’de TILT Module açık devre şeması ve pin bağlantıları verilmiştir. Programda **PB12 pini** kullanılmalıdır.

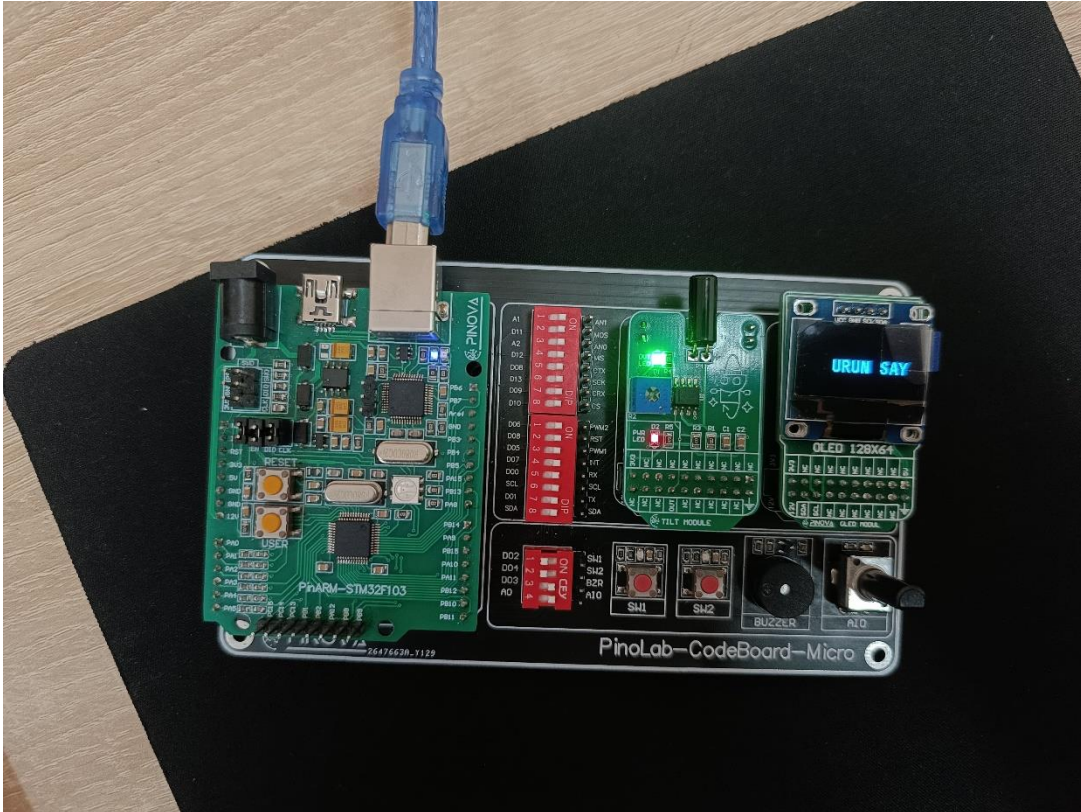


Şekil 1. TILT Module

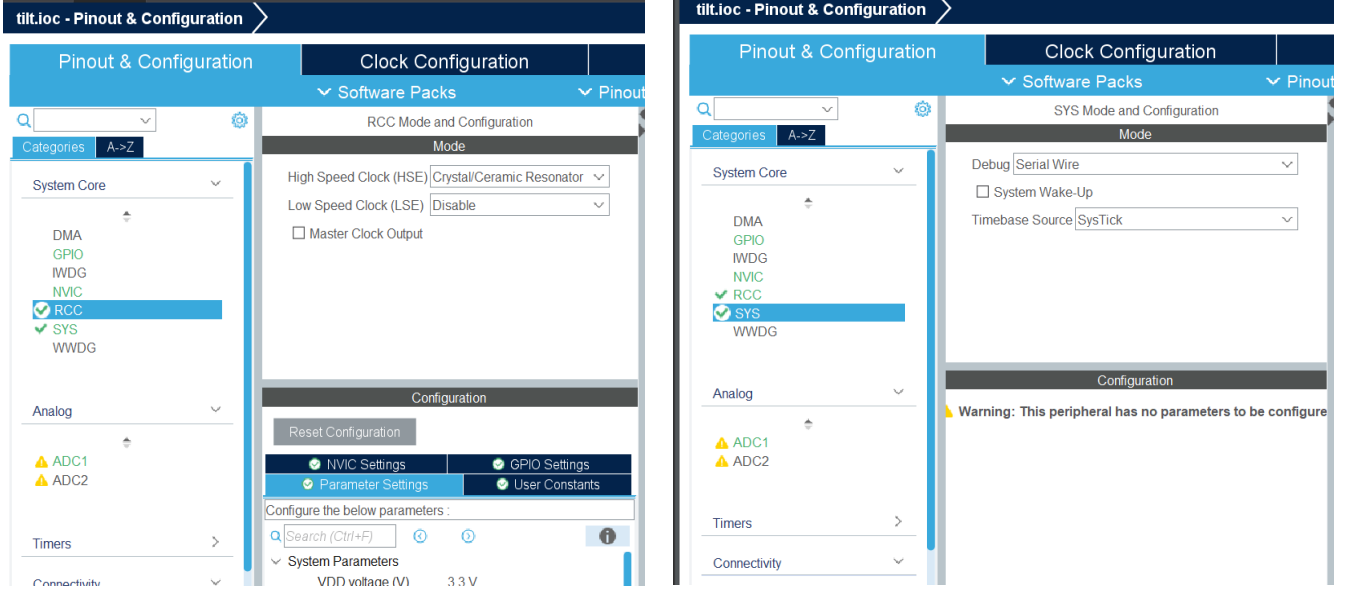


Şekil 2. TILT Module açık devre şeması ve pin bağlantıları

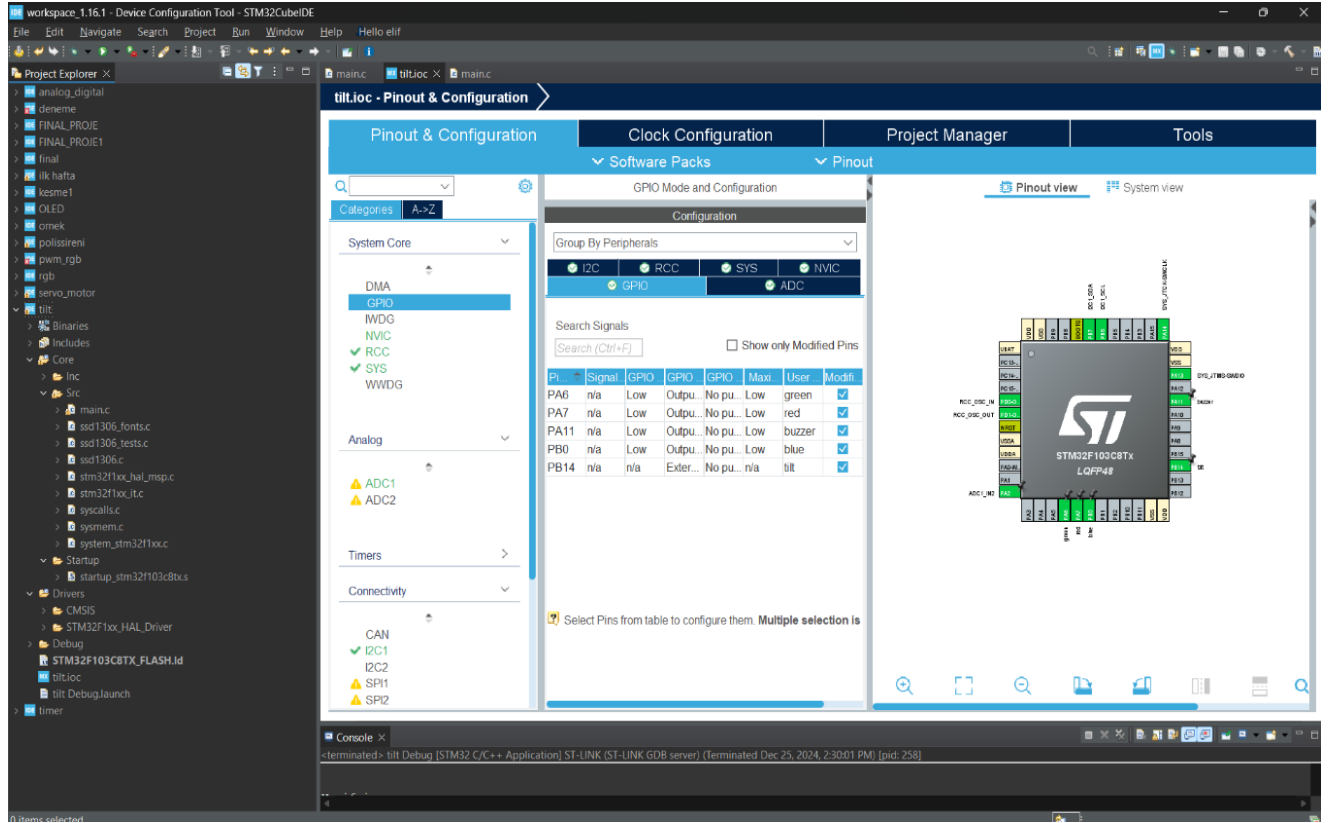
4- Donanım Bağlantıları

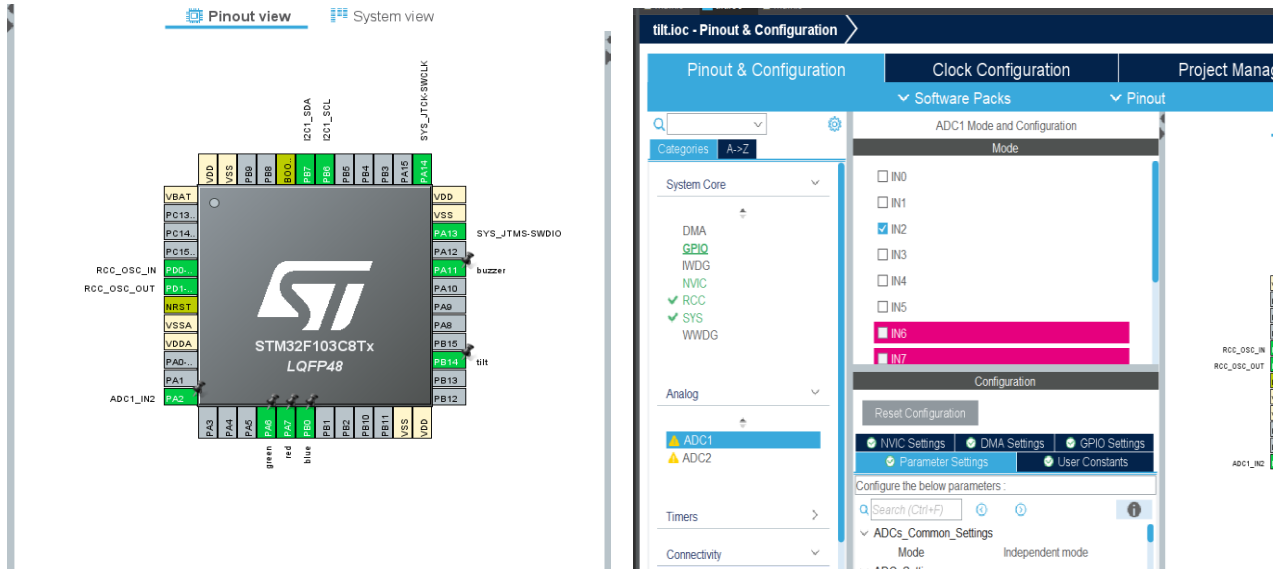


System Core -> SYS -> Debug : Serial Wire seçilir. System Core -> RCC -> High Speed Clock (HSE): Crystal/Ceramic Resonator seçilir.

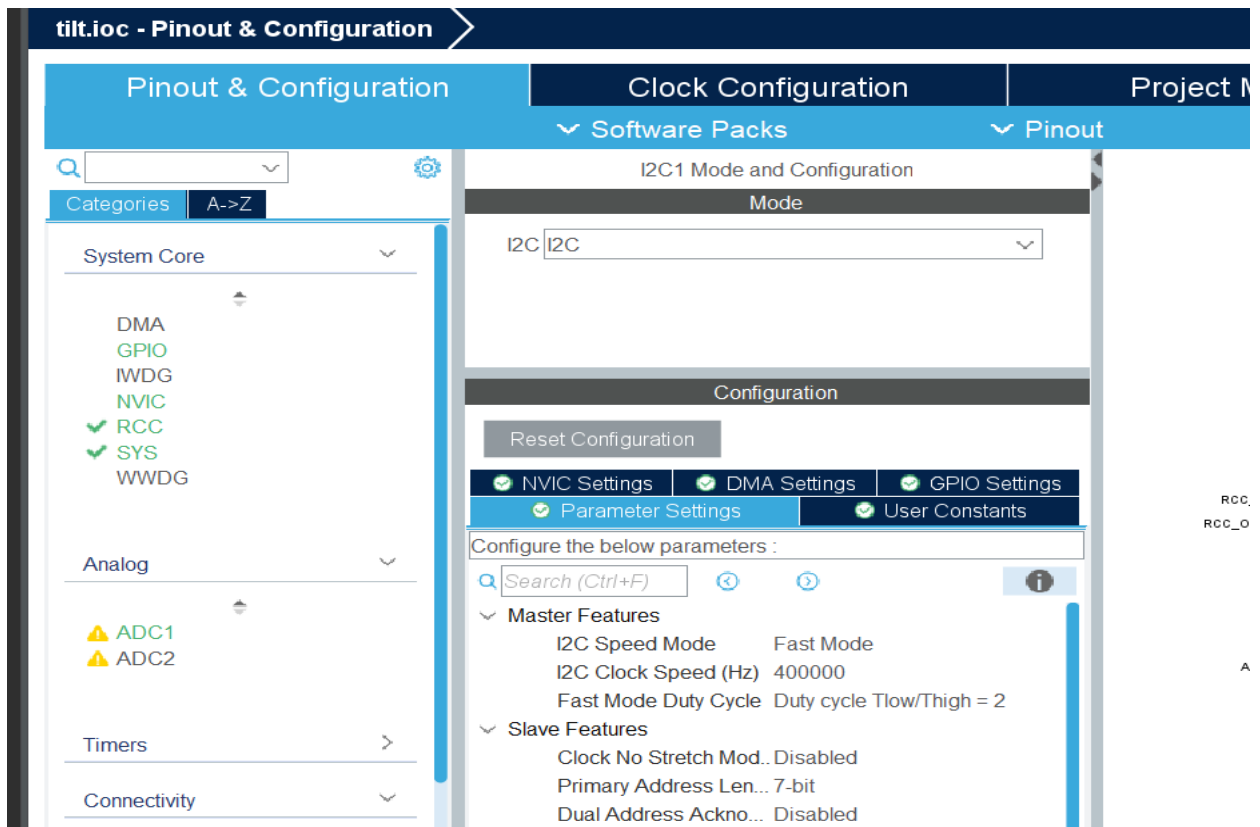


Genel ayarların tümü yapıldıktan sonra Pinout&Configuration -> System Core -> GPIO bölümünden istenilen pinlerin ayarlaması yapılabilir. PA2 pini ADC1_IN2 şeklinde seçilir ve parametre ayarları yapılır.

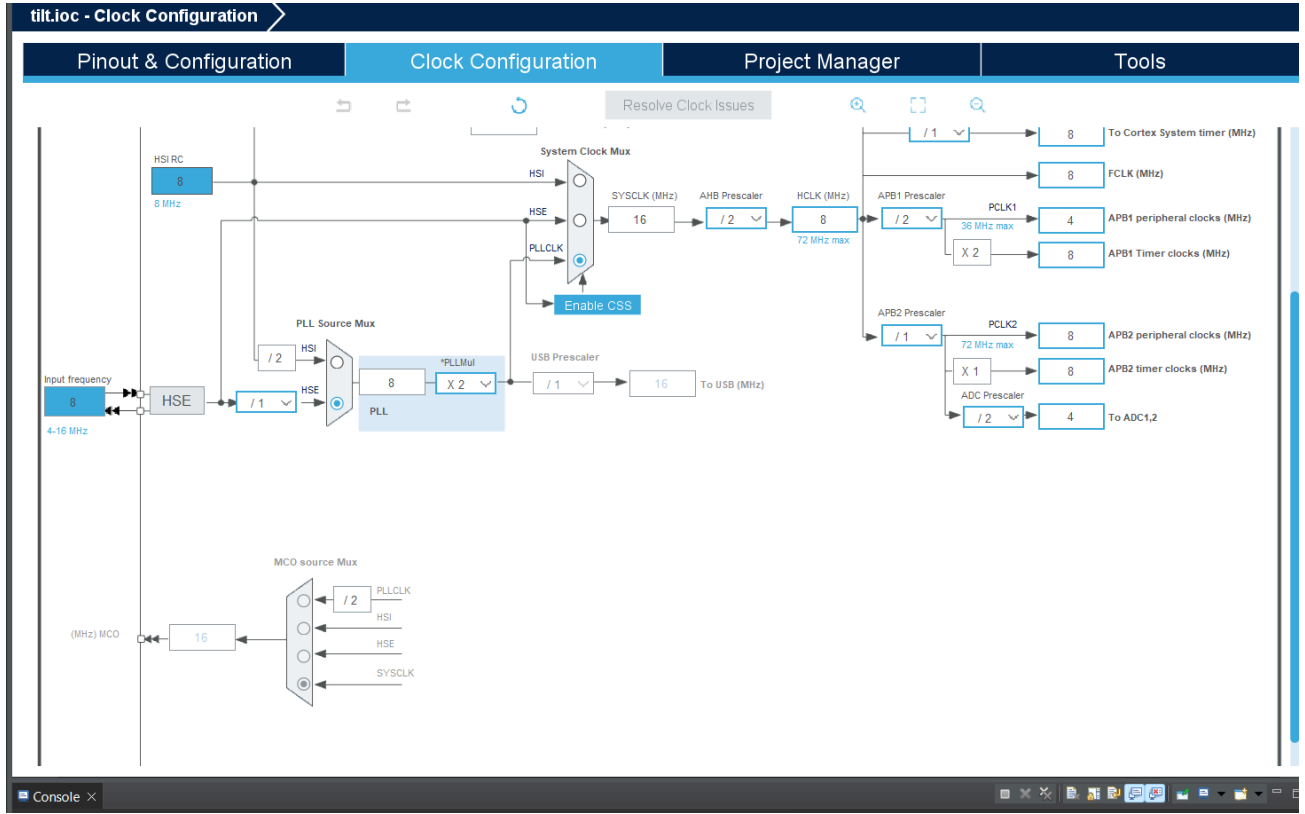




Connectivity bölümünde I2C kısmı seçilir ve PB7 pini I2C1_SDA; PB6 pini I2C1_SCL olarak seçilir. Parametre ayarları girilir. I2C Speed Mode seçeneği Fast Mode olarak seçilir.



Daha sonra Clock Configuration ayarları yapılır.



Pin seçimleri ve ayarlar yapıldıktan sonra, TILT sensörün bağlı olduğu pin harici kesme şeklinde belirlendiğinden, `stm32f1.._it.c` penceresinde `HAL_GPIO_EXTI_IRQHandler(GPIO_PIN_14)` komutu yazılmalıdır. Bu komut kullanıldıktan sonra, `main.c` dosyasında kodlar, `void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)` fonksiyonu oluşturularak bu fonksiyonun içerisine yazılmalıdır.

```
tilt
├── Binaries
├── Includes
├── Core
│   ├── Inc
│   └── Src
│       ├── main.c
│       ├── ssd1306_fonts.c
│       ├── ssd1306_tests.c
│       ├── ssd1306.c
│       ├── stm32f1xx_hal_msp.c
│       ├── stm32f1xx_it.c
│       ├── syscalls.c
│       ├── system_m.c
│       └── system_stm32f1xx.c
├── Startup
├── Drivers
├── Debug
├── STM32F103C8TX_FLASH.ld
├── tilt.ioc
└── tilt.Debug.launch
```

```
212 /* USER CODE END EXTI15_10_IRQn 1 */
213 }
214
215 /**
216  * @brief This function handles EXTI line[15:10] interrupt request via EXTI interrupt 10.
217  */
218 void EXTI15_10_IRQHandler(void)
219 {
220     /* USER CODE BEGIN EXTI15_10_IRQn 0 */
221
222     /* USER CODE END EXTI15_10_IRQn 0 */
223     HAL_GPIO_EXTI_IRQHandler(tilt_Pin);
224     /* USER CODE BEGIN EXTI15_10_IRQn 1 */
225
226     /* USER CODE END EXTI15_10_IRQn 1 */
227 }
228
229 /* USER CODE BEGIN 1 */
230
231 /* USER CODE END 1 */
232
```

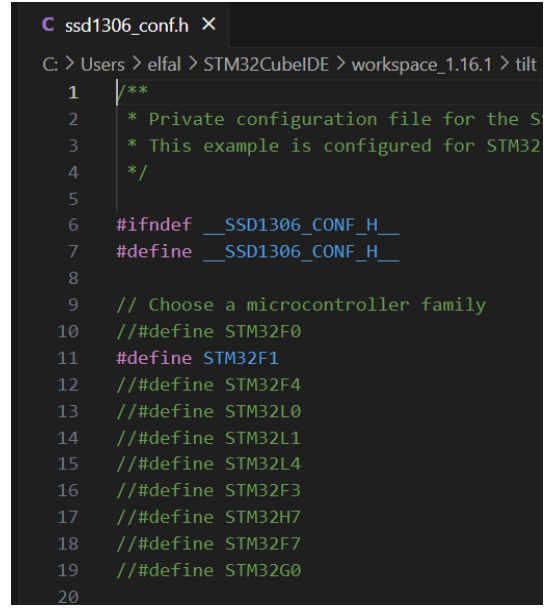
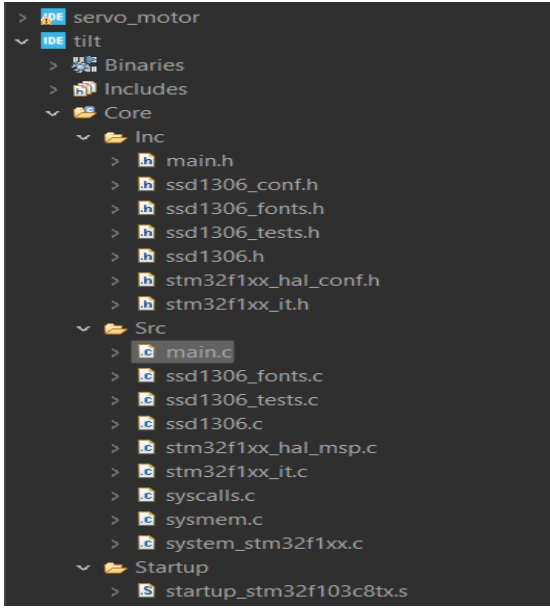
Console X
No consoles to display at this time

OLED Ekran Modülü için Kütüphane Kurulumu

SSD1306 denetleyicisi karmaşık sürücülere sahip olduğundan ekranı kontrol etmek adına bu uygulama için, daha basit komutların kullanımına olanak sağlayan , SSD1306 kütüphanesi kullanılacaktır.

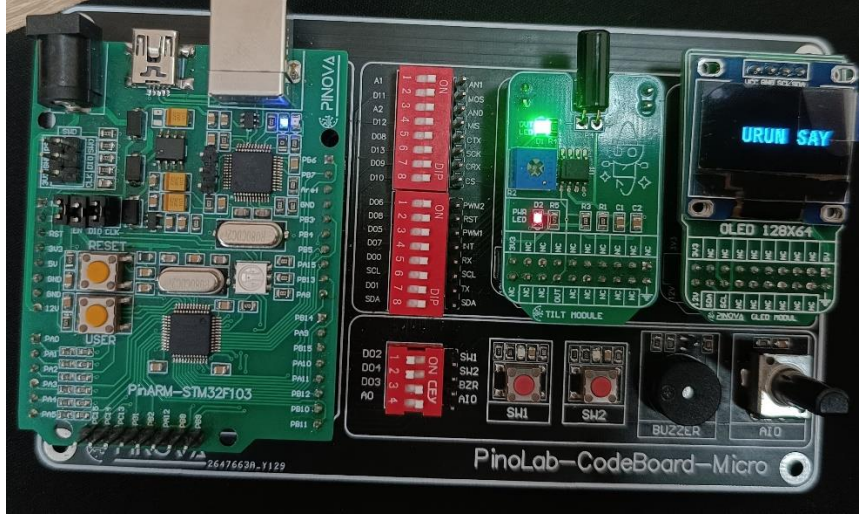
STMCubeIDE programında yaptığınız projelerin bulunduğu klasöre giderek .h uzantılı (**ssd1306.h** ve **ssd1306_conf.h**, **ssd1306_fonts.h**, **ssd1306_tests.h**) dosyaları proje dosyası içerisindeki Inc klasörünün içine kopyalayınız (Core-> Inc). Daha sonra .c uzantılı dosyaları (**ssd1306.c** ve **ssd1306_fonts.c**, **ssd1306_tests.c**), src klasörünün içine kopyalayınız. Bu işlem için bir örnek aşağıda gösterilmiştir.

ssd1306_conf.h dosyasında **#define STM32F0** kısmı yorum satırına çevrilip **#define STM32F1** kısmı yorum satırından çıkartılır.



Kütüphaneler klasörlere eklendikten sonra STMCubeIDE programında Src ve Inc klasörleri sağ tıklanarak Refresh edilmelidir. Bu aşamadan sonra .h ve .c uzantılı dosyalar Inc ve Src klasörleri içerisinde görülecektir.

4) DENEY SETİ GÖRSELİ



Şekil 6. PinoLab-CodeBoard.Micro üzerindeki Tilt ve OLED Module,

Gerekli ayarlar yapıldıktan sonra kodlar kaydedilir veya derlenir. Main.c dosyasında kodlamaya başlarken kütüphanelerin ekli olduğundan emin olunmalıdır. Aksi takdirde yazılacak kod satırlarında hatalar meydana gelecektir.

```
/*  
 * USER CODE END Header */  
/* Includes -----  
#include "main.h"  
#include "ssd1306.h"  
#include "ssd1306_fonts.h"  
/* Private includes -----  
/* USER CODE BEGIN Includes */  
  
/* USER CODE END Includes */
```

Main.c dosyasındaki ayarları kontrol ediniz.

```
42 /* Private variables -----  
43 ADC_HandleTypeDef hadc1;  
44  
45 I2C_HandleTypeDef hi2c1;  
46  
47 /* USER CODE BEGIN PV */  
48  
49 /* USER CODE END PV */  
50  
51 /* Private function prototypes -----  
52 void SystemClock_Config(void);  
53 static void MX_GPIO_Init(void);  
54 static void MX_ADC1_Init(void);  
55 static void MX_I2C1_Init(void);  
56 /* USER CODE BEGIN PFP */  
57  
58 /* USER CODE END PFP */  
59  
60 /* Private user code -----  
61 /* USER CODE BEGIN 0 */  
62  
63 uint32_t ADC_data;
```

5) DENEYİN YAPILIŞI VE SONUÇLAR

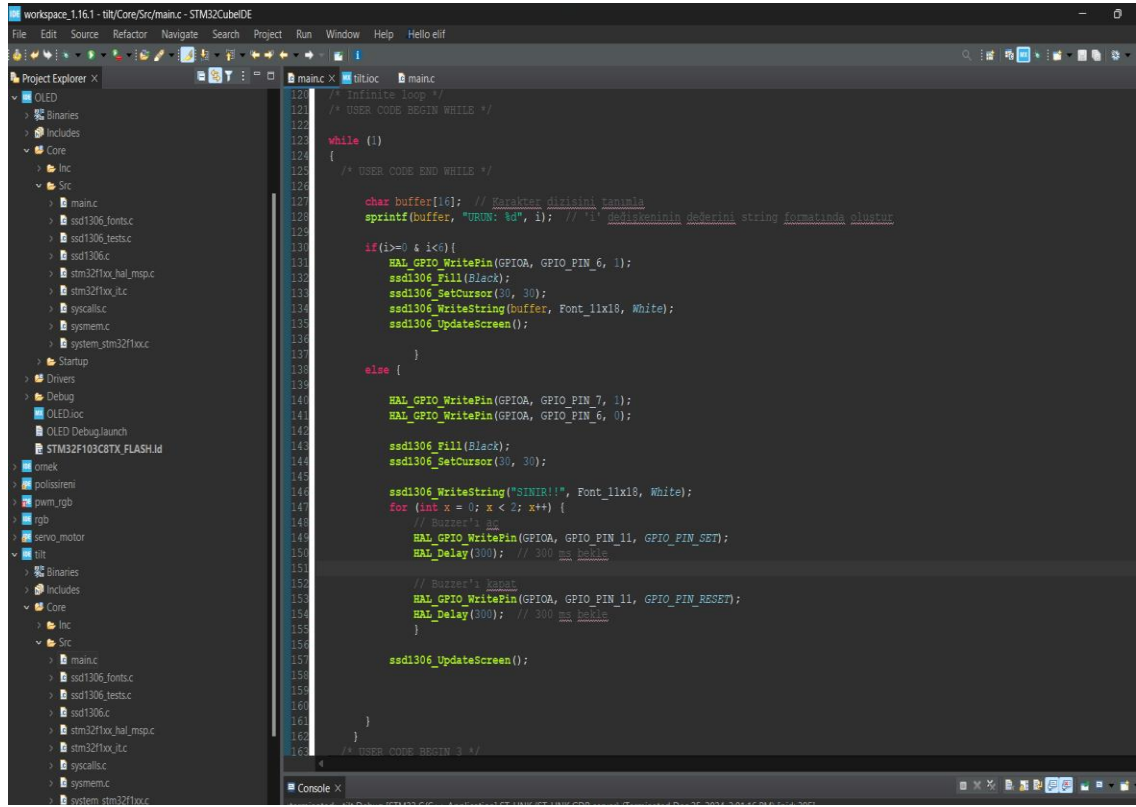
1. STM32CubeIDE programında yeni proje açılır.
2. STM32CubeIDE programında SYS (Debug: Serial Wire) ve RCC (Reset and Clock Controller) ayarları yapılır.
3. “Pinout&Configuration” ayarlarından sonra “Clock Configuration” ardından da “Project Manager” ayarları yapılır.
4. STM32CubeIDE GPIO ayarlarından ADC uygulaması için gerekli pinler anlatıldığı üzere ayarlanır.
5. Tüm ayarlamalar yapıldıktan sonra “**Kaydet**” butonuna tıklanır ve kodlar program tarafından oluşturulur.
6. Kodlar oluşturulduktan sonra kod penceresinde (main.c) kullanıcının kod yazmasına izin verilen alanlara gerekli kodlar yazılır.

1. NOT: Kullanıcının yazacağı kodlar yalnızca izin verilen alanlara yazılmalıdır. Aksi takdirde yazılan bütün kodlar SİLİNECEKTİR.

7. Kodlar derlenir. Kodlarda herhangi bir hata var ise hatalar düzeltilir.
8. PinARM-STM32F103C8T6 PinoLab-CodeBoard.Micro üzerine takılır.
9. LDR Module ve OLED Module Şekil 6’da gösterildiği üzere PinoLab-CodeBoard.Micro üzerine takılır.
10. Modül portlarına ait switchler ON konumuna getirilir. Diğer tüm switchler OFF konumunda olmalıdır.
11. Bağlantı kablosu takılır.
12. Derlenen kodlar yüklenir.
13. Kodlarda herhangi bir hata yok ise OLED ekranda okunan değerler gözlenir. Değerlerin doğru olup olmadığı Debug üzerinden, değişkenin aldığı değerlere göre kontrol edilir.

6-Kod Kısmı

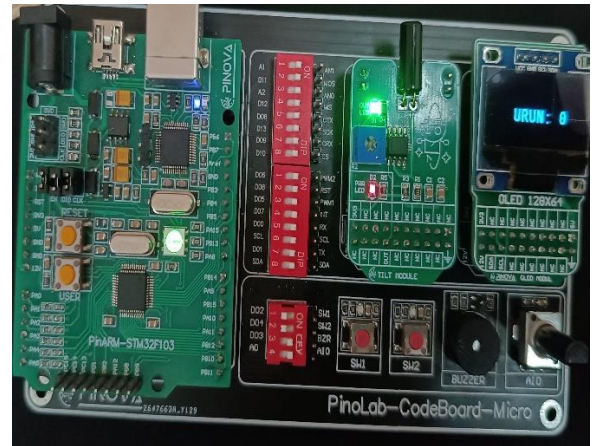
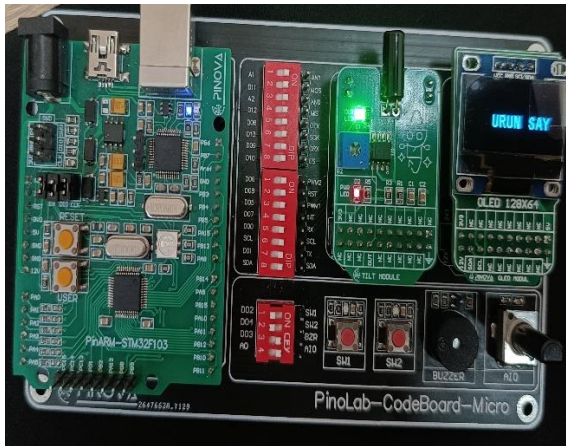
```
main.c x tiltioc main.c
99  /* USER CODE END SysInit */
100
101  /* Initialize all configured peripherals */
102  MX_GPIO_Init();
103  MX_ADC1_Init();
104  MX_I2C1_Init();
105  /* USER CODE BEGIN 2 */
106  ssd1306_Init();
107  ssd1306_Fill(Black); // Ekranı temizle
108
109  // OLED ekranı başlat
110  ssd1306_Init();
111  ssd1306_Fill(Black); // Ekranı temizle
112
113  // Yazıyı ekrana yaz
114  ssd1306_SetCursor(30, 30);
115  ssd1306_WriteString("URUN SAY", Font_11x18, White);
116  ssd1306_UpdateScreen();
117  HAL_Delay(3000);
118  /* USER CODE END 2 */
119
120  /* Infinite loop */
121  /* USER CODE BEGIN WHILE */
122
```

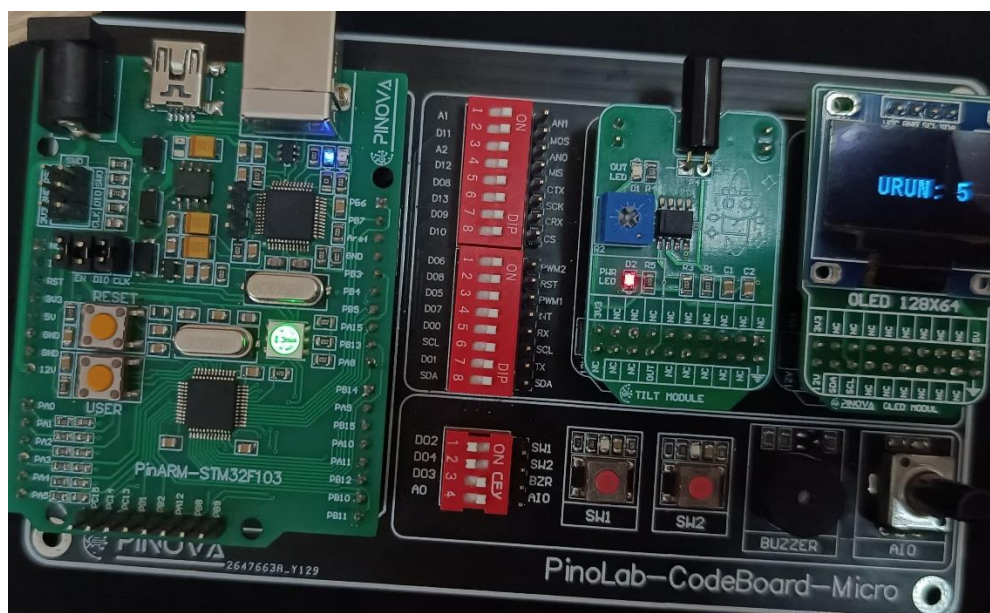
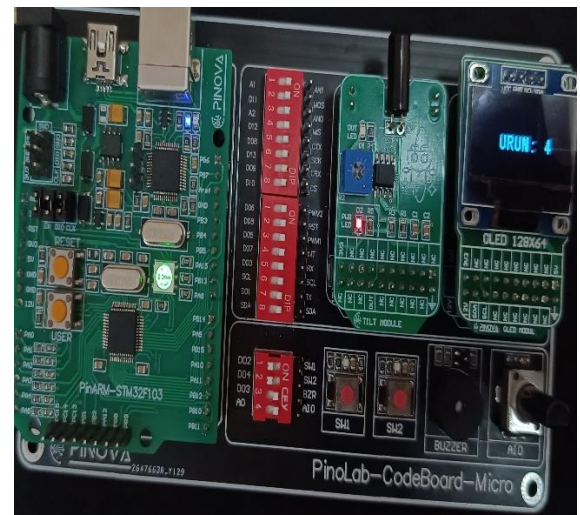
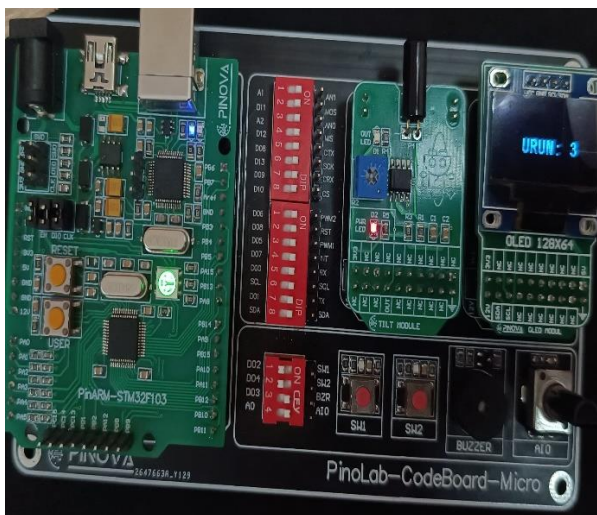
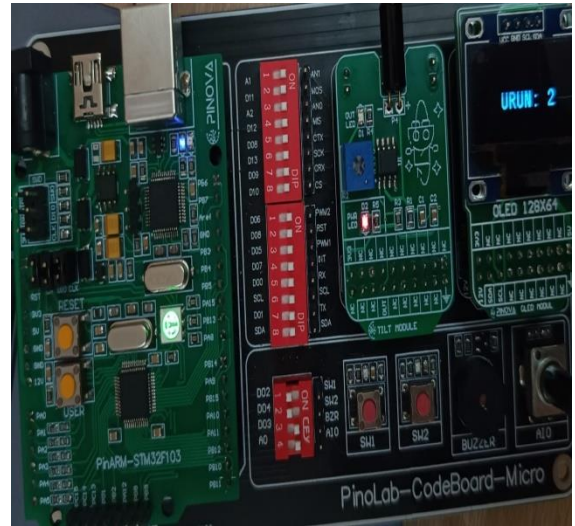
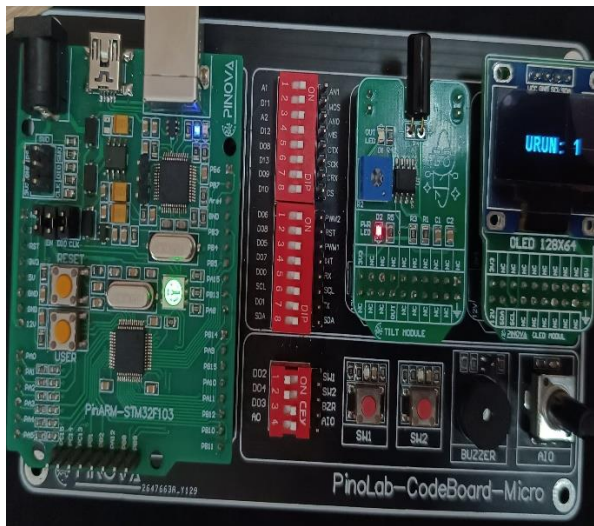


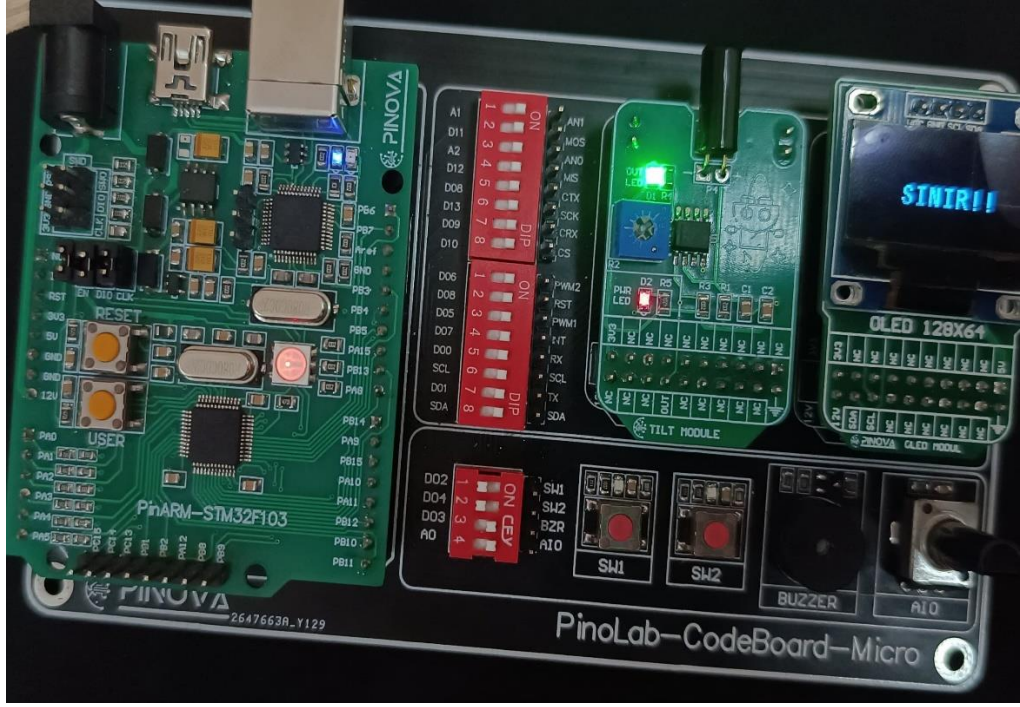
```
121  /* Infinite loop */
122  /* USER CODE BEGIN WHILE */
123
124  while (1)
125  {
126      /* USER CODE END WHILE */
127
128      char Buffer[16]; // Karakter dizisini tanımla
129      sprintf(Buffer, "URUN: %d", i); // 'i' değisinin değeri string formatında alıntır.
130
131      if (i >= 0 & i < 6) {
132          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
133          ssd1306_Fill(Black);
134          ssd1306_SetCursor(30, 30);
135          ssd1306_WriteString(Buffer, Font_11x18, White);
136          ssd1306_UpdateScreen();
137      }
138      else {
139          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
140          HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);
141
142          ssd1306_Fill(Black);
143          ssd1306_SetCursor(30, 30);
144
145          ssd1306_WriteString("SINIR!!!", Font_11x18, White);
146          for (int x = 0; x < 2; x++) {
147              // Buzzer'ın sesi
148              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
149              HAL_Delay(300); // 300 ms bekle
150
151              // Buzzer'ın sesi
152              HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
153              HAL_Delay(300); // 300 ms bekle
154          }
155
156          ssd1306_UpdateScreen();
157      }
158  }
159  /* USER CODE BEGIN 3 */
160  }
```

7-Sonuçlar

Program çalışmaya başladığında ekrana “urun say” mesajı geliyor ve ardından tilt sensörü hareket ettikçe “URUN: %d” şeklinde saymaya başlıyor. Aynı zamanda yeşil ışık yanmaya başlıyor.







Tilt sensörü 6 ya ulaştığında ise ekranda “SINIR!!” mesajı geliyor ve aynı zamanda hem kırmızı ışık yanıp hem de buzzer kesintili bir şekilde çalmaya başlıyor.