MAKÜ | GUBYO

GÖLHİSAR UYGULAMALI BİLİMLER YÜKSEKOKULU

MİKROİŞLEMCİLER UYGULAMA
FİNAL ÖDEV RAPORU

HAZIRLAYAN:

Öğrencinin Adı : Elif ALTINTAŞ BURHAN ÜSTÜBİ

No : 1912901001 2012903069

Bölüm : Bilişim Sistemleri ve Teknolojileri 4 / A

Ders : Mikroişlemciler Uygulama

1) DENEYİN ADI

Deney : OLED Modul ve Tılt 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

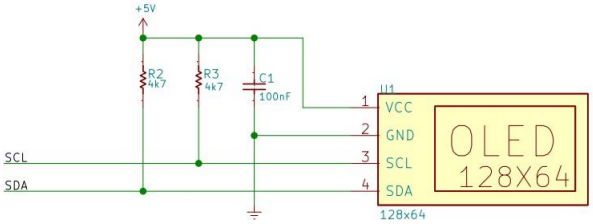
-PinoLab-CodeBoard.Micro
-PinARM-STM32F103C8T6
-OLED Module
Tilt Sensörü
-Bağlantı kablosu

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

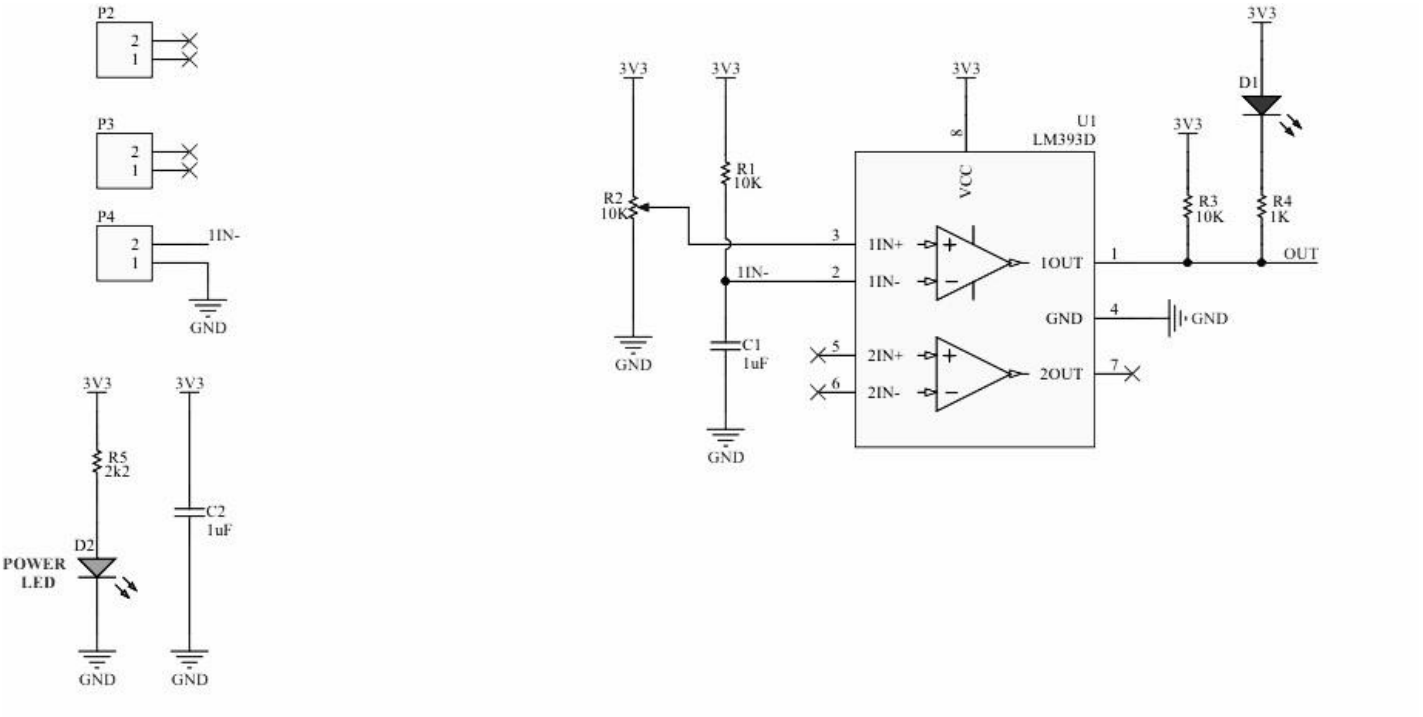
I2C (Inter-Integrated Circuit) protokolü, seri iletişim için kullanılan bir protokoldür. Master-slave yapısına dayanır ve veri iletişimi için SDA ve SCL hatlarını kullanır.

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.

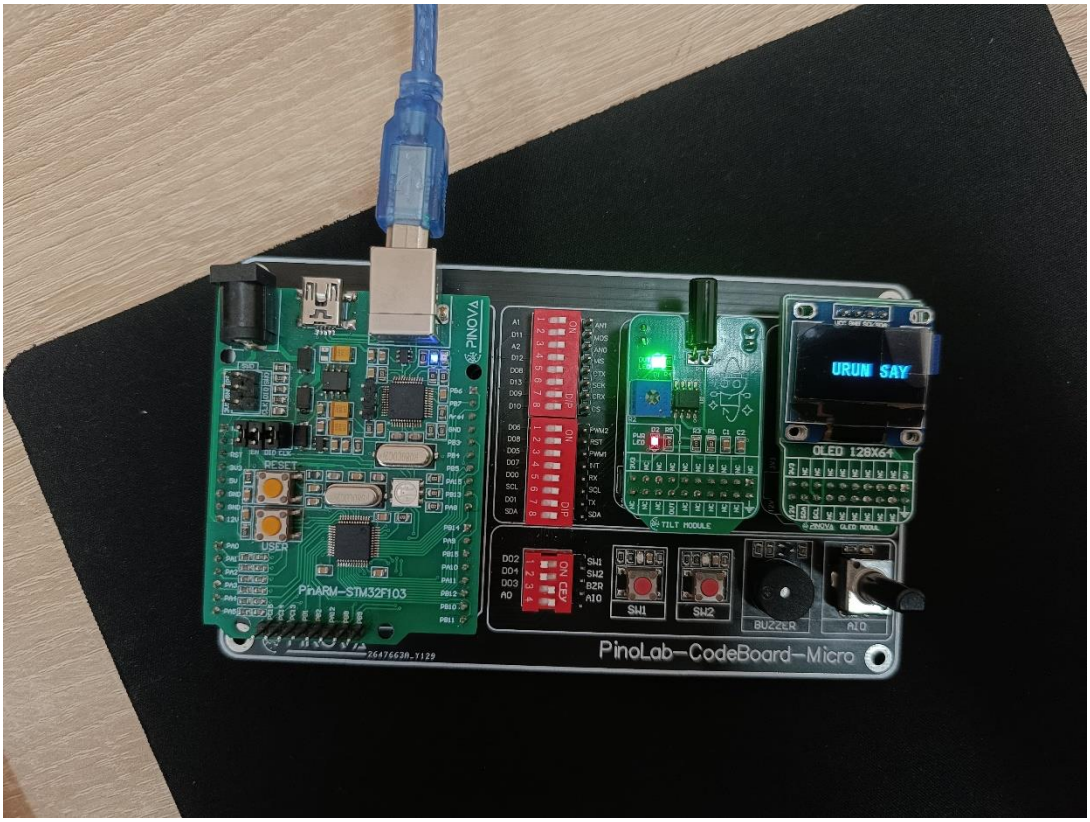
Şekil 1'de TILT Module ve Şekil 2'de TILT Module açık devre şeması ve pin bağlantıları verilmiştir. Programda **PB12 pini** kullanılmalıdır.

Şekil 1. TILT Module

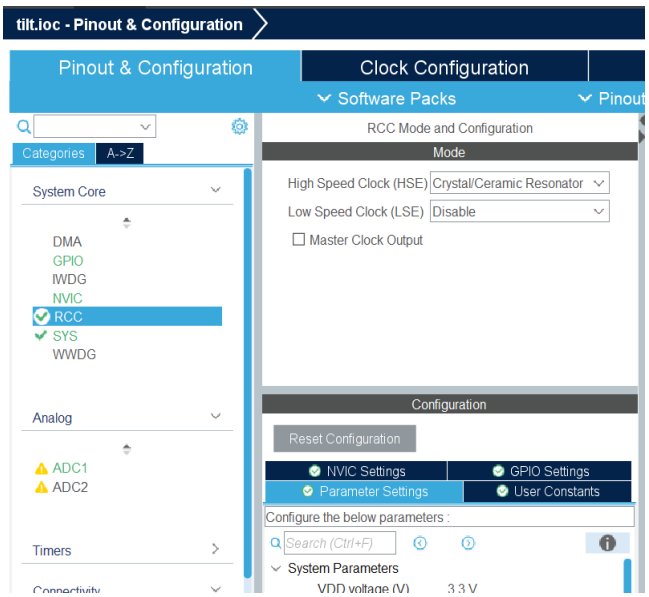


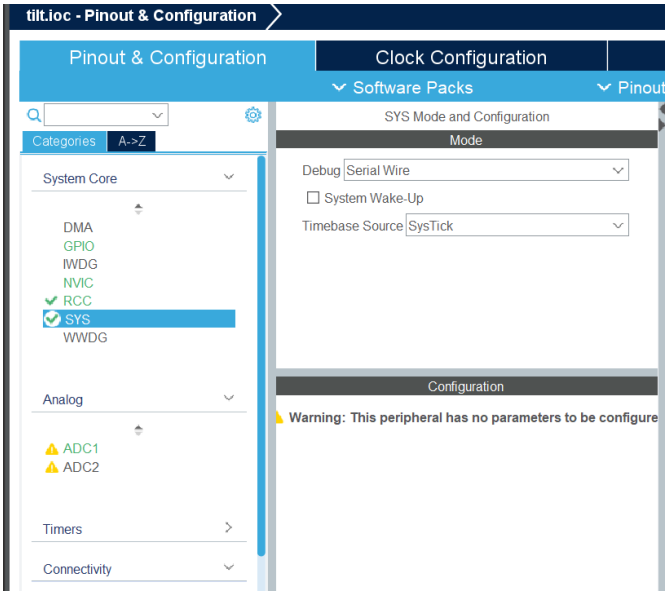
Şekil 2. TILT Module açık devre şeması ve pin bağlantıları

4- Donanım Bağlantıları

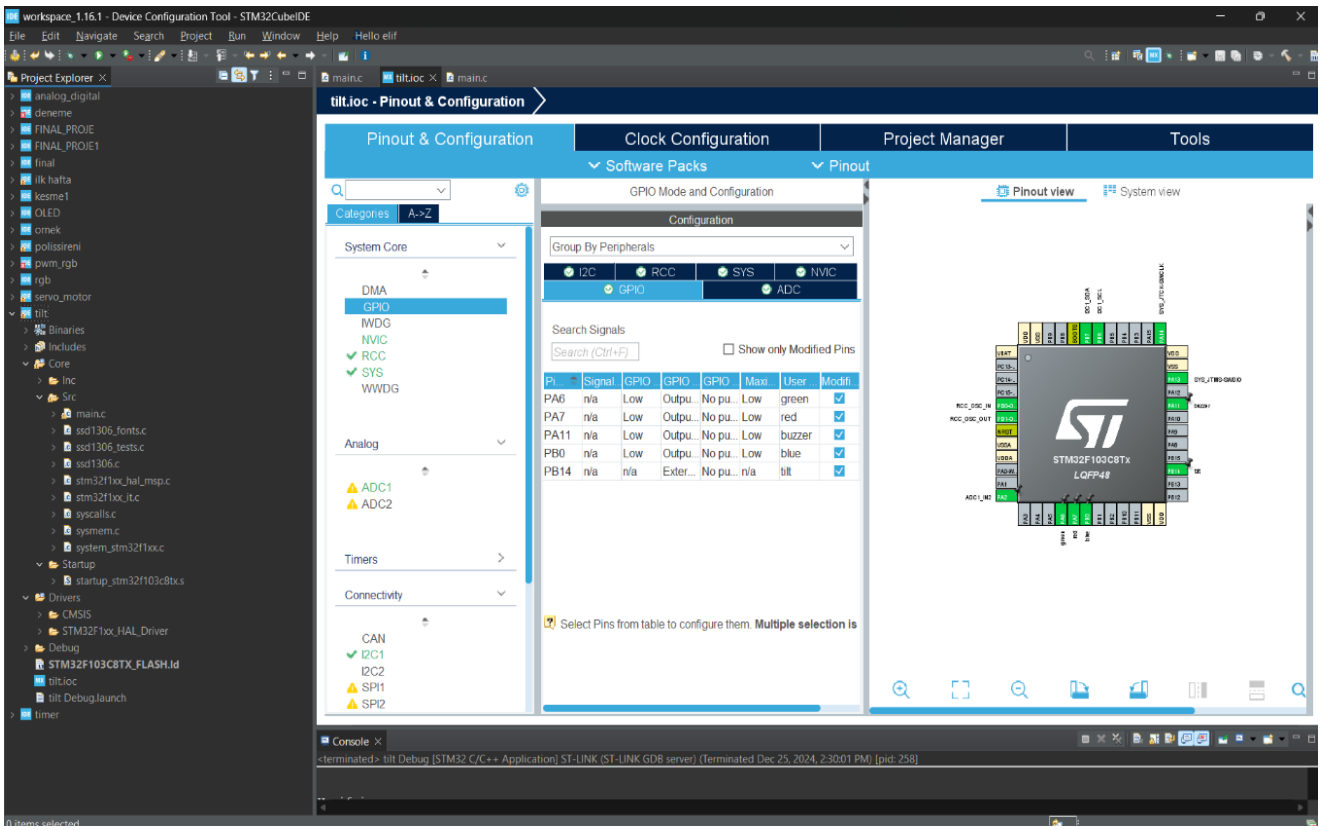


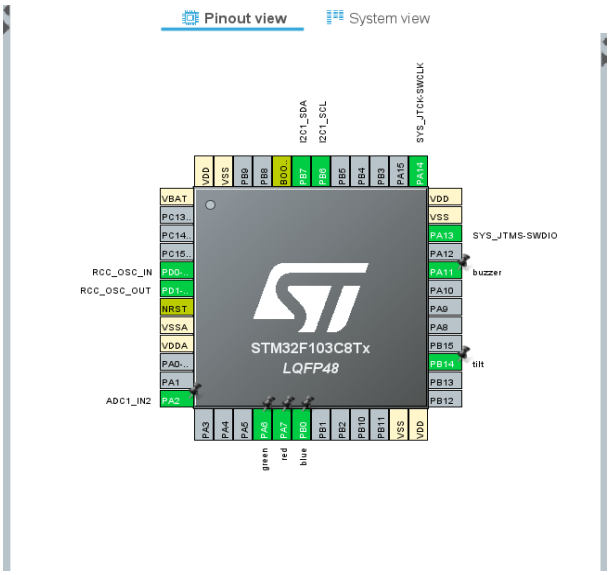
System Core -> SYS -> Debug : Serial Wire seçilir. System Core -> RCC -> High Speed Clock (HSE): Crystal/Ceramic Resonator seçilir.

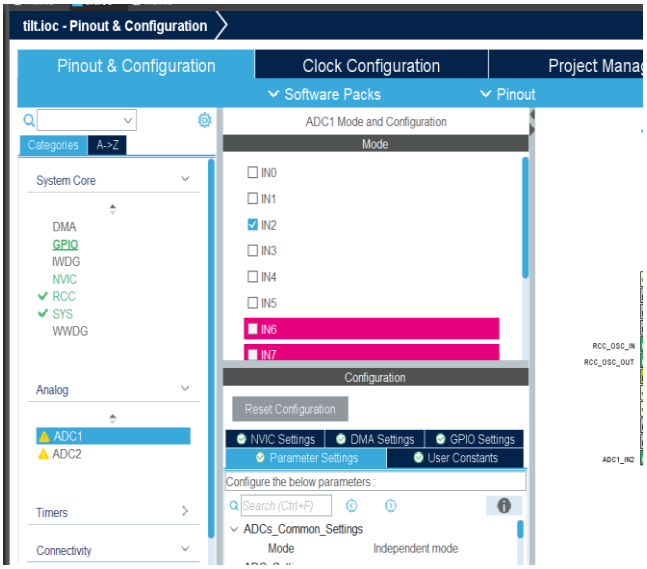




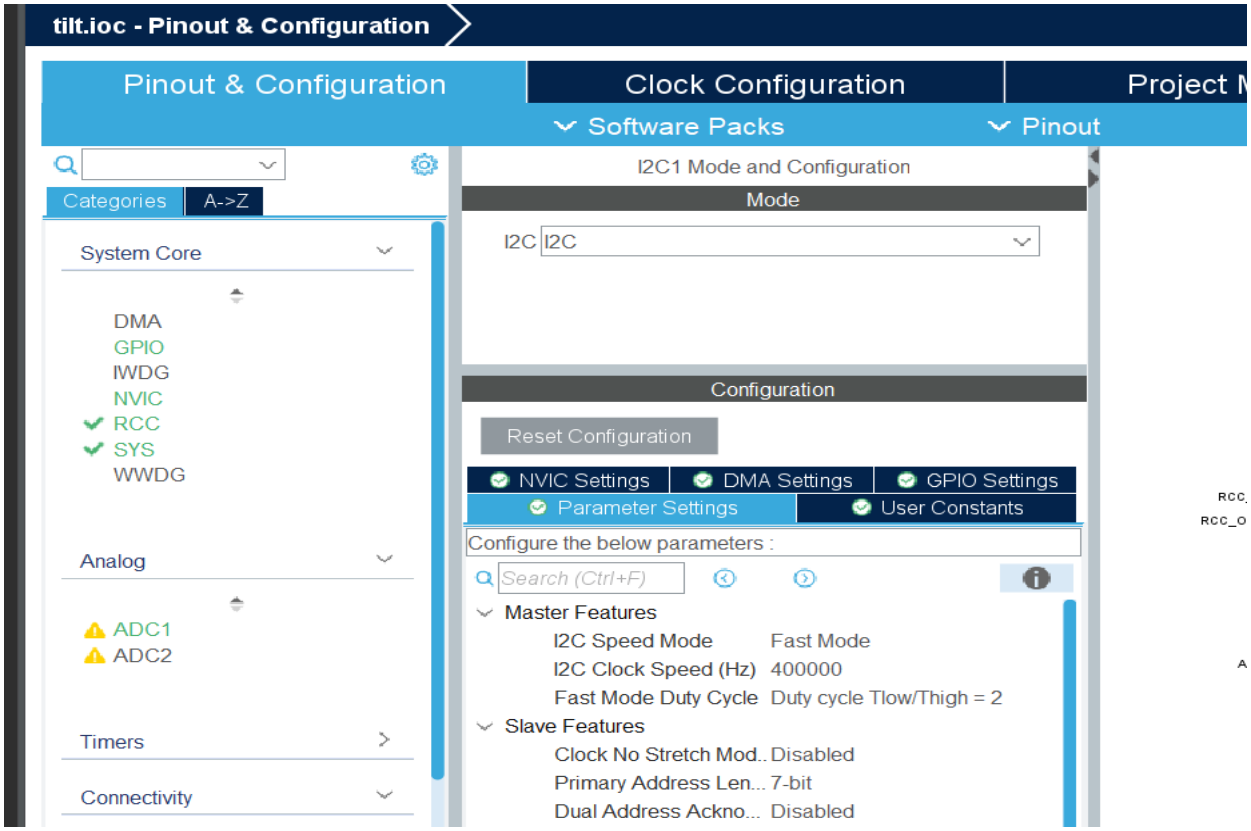
Genel ayarların tümü yapıldıktan sonra Pinout&Configuration -> System Core -> GPIO bölümünden istenilen pinlerin ayarlaması yapılabilir. PA2 pini ADC1_IN2 şeklinde seçilir ve parametre ayarları yapılır.



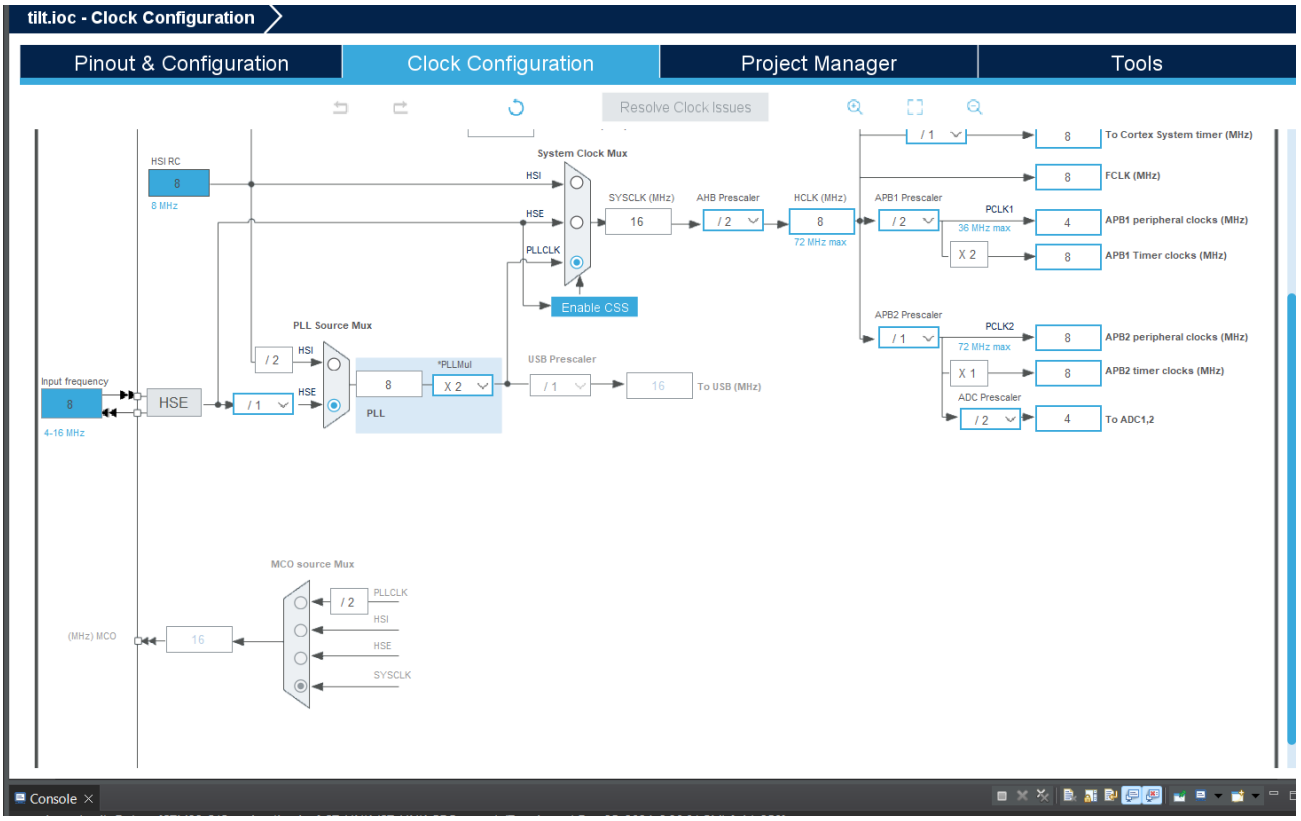




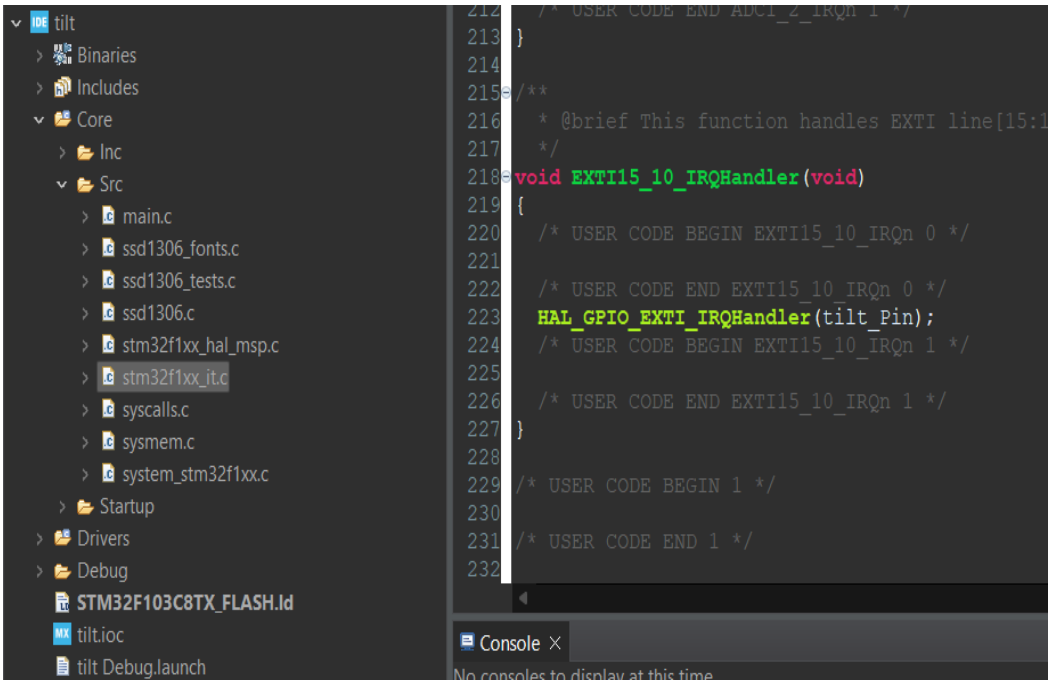
Connetivity bölümünde I2C kısmı seçilir ve PB7 pini I2C1_SDA; PB6 pini I2C1_SCL olarak seçilir. Parametre ayarları girilir. I2C Speed Mode seçeneği Fast Mode olarak seçilir.



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.

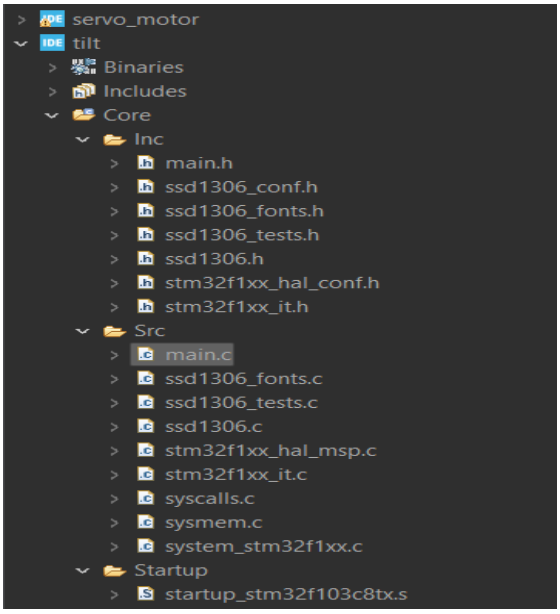


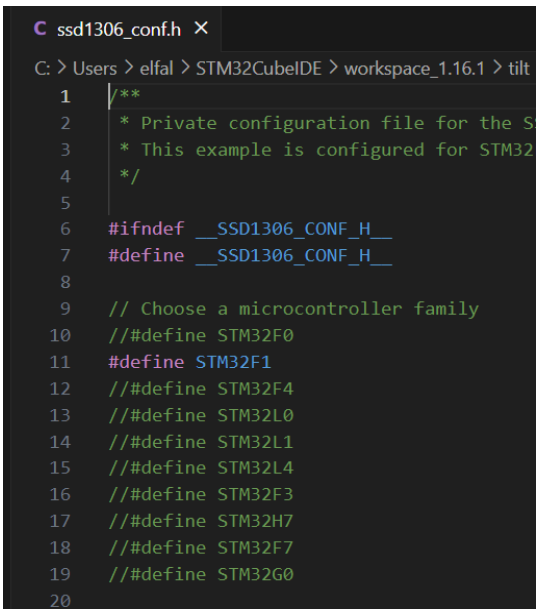
OLED Ekran Modülü için Kütüphane Kurulumu

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.

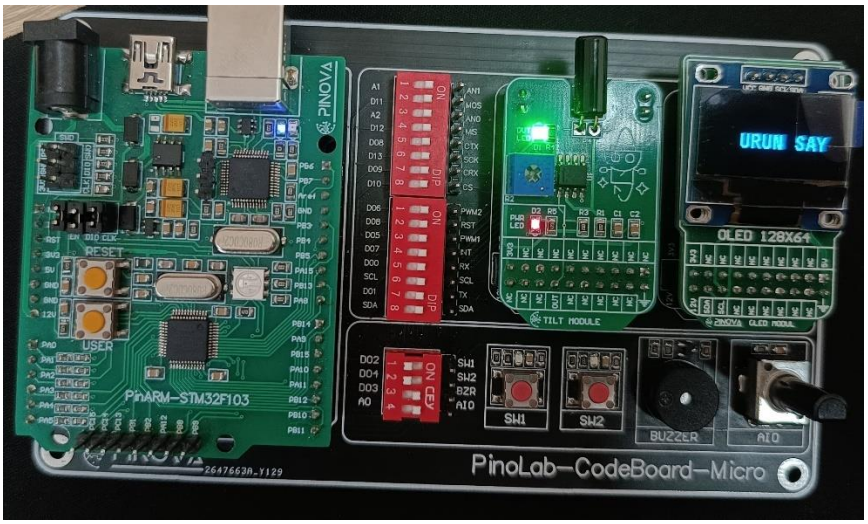
ssd1306_conf.h dosyasında **#define STM32F0** kısmı yorum satırına çevrilip **#define STM32F1** kısmı yorum satırından çıkartılır.





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

```
/* Private variables ---------------------------------------------------------*/
 ADC_HandleTypeDef hadc1;

I2C_HandleTypeDef hi2c1;

/* USER CODE BEGIN PV */

/* USER CODE END PV */

/* Private function prototypes -----------------------------------------------*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);
static void MX_I2C1_Init(void);
/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code ---------------------------------------------------------*/
/* USER CODE BEGIN 0 */

uint32_t ADC_deger;
```

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.

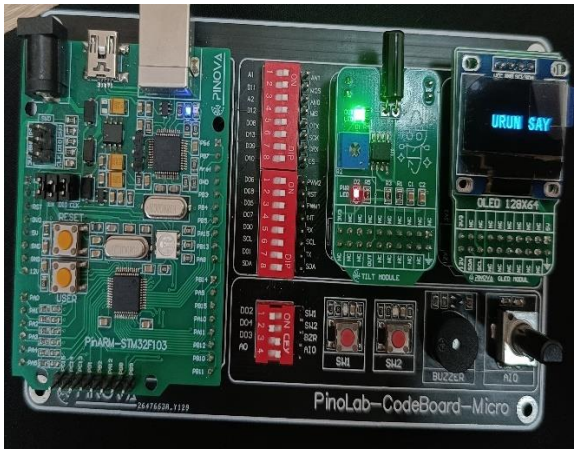
6-Kod Kısmı

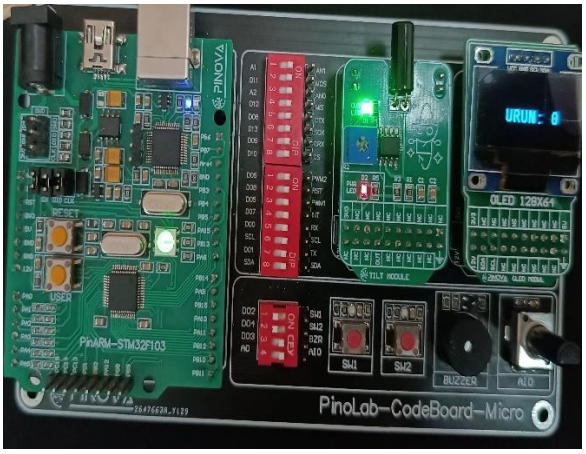
```
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_ADC1_Init();
  MX_I2C1_Init();
  /* USER CODE BEGIN 2 */
  ssd1306_Init();
  ssd1306_Fill(Black); // Ekranı temizle

  // OLED ekranı başlat
  ssd1306_Init();
  ssd1306_Fill(Black); // Ekranı temizle

  // Yazıyı ekrana yaz
  ssd1306_SetCursor(30, 30);
  ssd1306_WriteString("URUN SAY", Font_11x18, White);
  ssd1306_UpdateScreen();
  HAL_Delay(3000);
//  /* USER CODE END 2 */
```

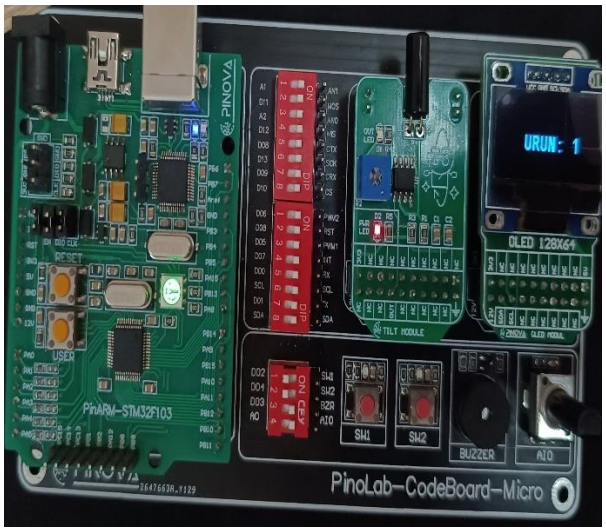
```c
  while (1)
  {
    /* USER CODE END WHILE */
        char buffer[16];  // Karakter dizisini tanımla
        sprintf(buffer, "URUN: %d", i);  // 'i' değişkeninin değerini string formatında oluştur

        if(i>=0 & i<6){
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 1);
            ssd1306_Fill(Black);
            ssd1306_SetCursor(30, 30);
            ssd1306_WriteString(buffer, Font_11x18, White);
            ssd1306_UpdateScreen();
}
        else {

            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, 1);
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, 0);

            ssd1306_Fill(Black);
            ssd1306_SetCursor(30, 30);

            ssd1306_WriteString("SINIR!!", Font_11x18, White);
            for (int x = 0; x < 2; x++) {
                // Buzzer'ı aç
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_SET);
                HAL_Delay(300);  // 300 ms bekle

                // Buzzer'ı kapat
                HAL_GPIO_WritePin(GPIOA, GPIO_PIN_11, GPIO_PIN_RESET);
                HAL_Delay(300);  // 300 ms bekle
                }
            ssd1306_UpdateScreen();
        }
      }
    /* USER CODE BEGIN 3 */
```

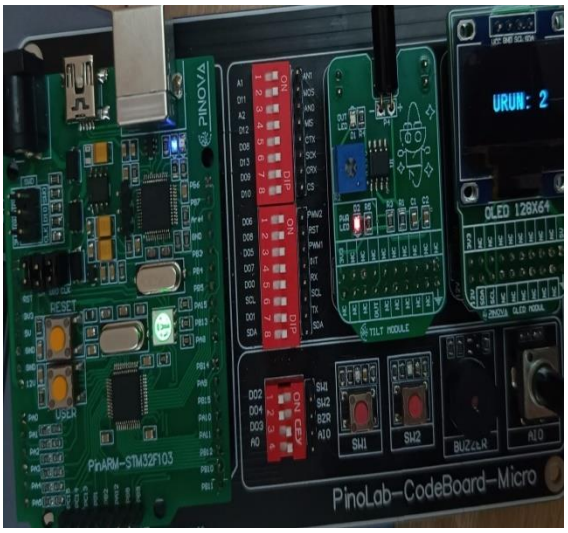
7-Sonuçlar

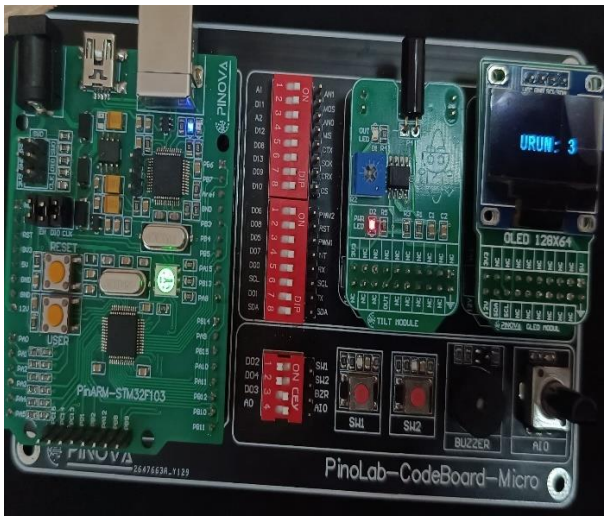
Program çalışmaya başladığında ekrana “urun say” mesajı geliyor ve ardından tilt sensörü hareket ettikçe “URUN: %d” şeklinde saymaya başlıyor. Aynı zamanda yeşil ışık yanmaya başlıyor.

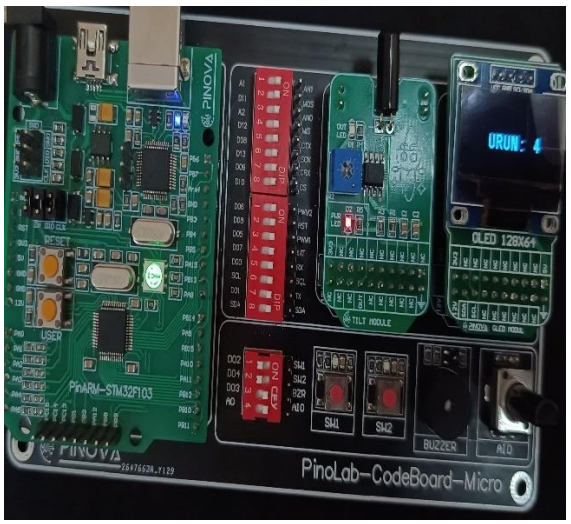


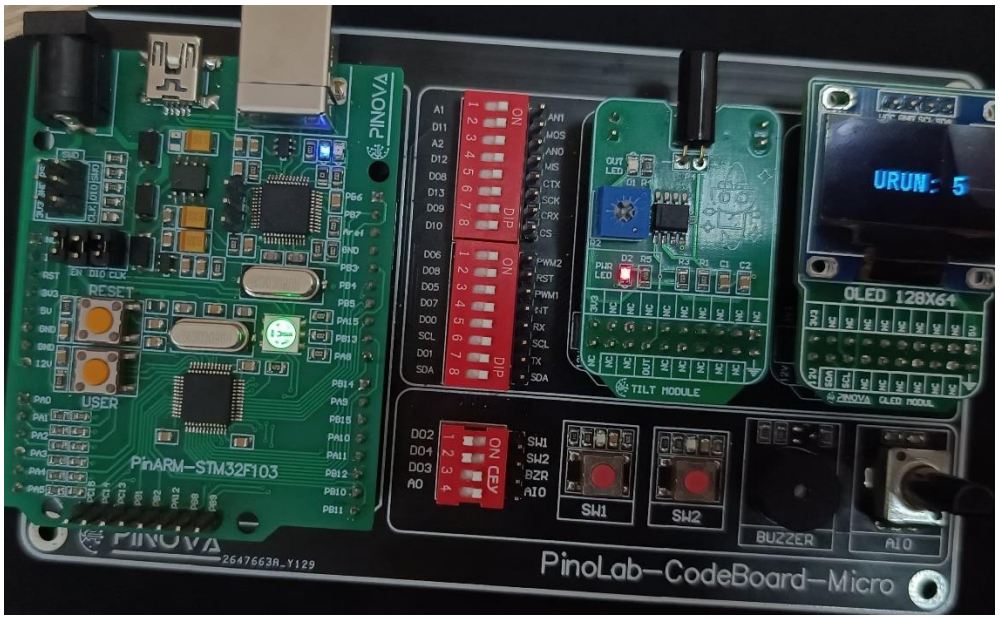


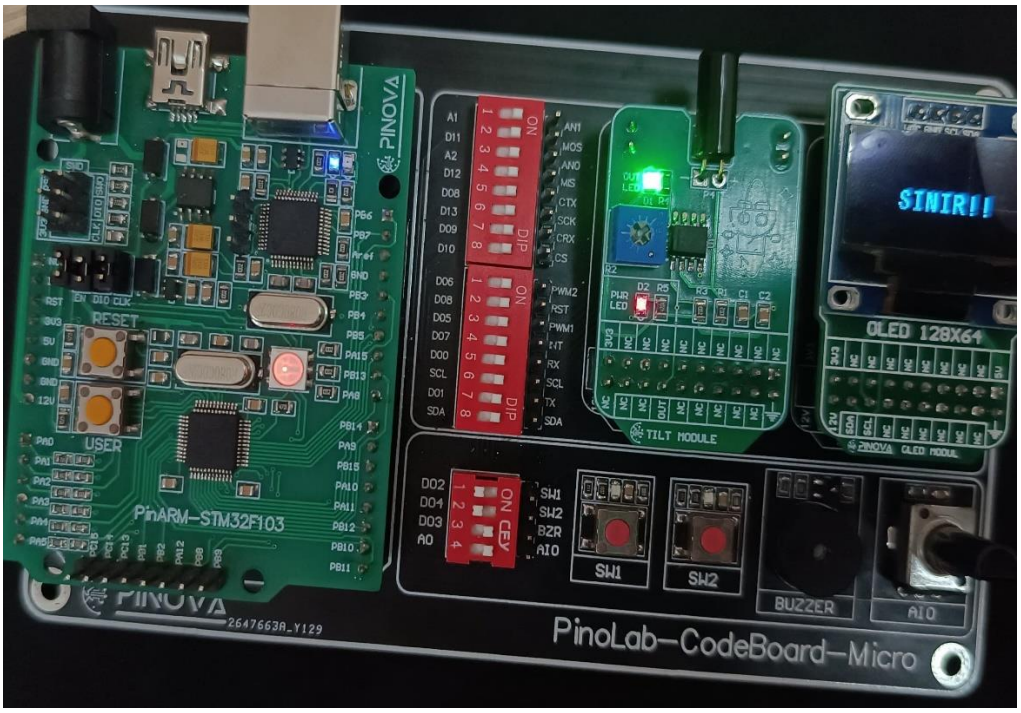

URUN: 1
OLED 128X64
PinARM-STM32F103
SW1
SW2
BUZZER
AIO
PinoLab-CodeBoard-Micro


URUN: 2
OLED 128X64
PinARM-STM32F103
SW1
SW2
AIO
PinoLab-CodeBoard-Micro


URUN: 3
OLED 128X64
PinARM-STM32F103
SW1
SW2
BUZZER
AIO
PinoLab-CodeBoard-Micro


URUN: 4
OLED 128X64
PinARM-STM32F103
SW1
SW2
BUZZER
AIO
PinoLab-CodeBoard-Micro


URUN: 5
OLED 128X64
PinARM-STM32F103
SW1
SW2
BUZZER
AIO
PinoLab-CodeBoard-Micro



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.