

MAKÜ|GUBYO

GÖLHİSAR UYGULAMALI BİLİMLER YÜKSEKOKULU



MAKÜ
GUBYO

MİKROİŞLEMCİLER UYGULAMA VİZE ÖDEV RAPORU

HAZIRLAYANLAR:

- ❖ **AD** : Ebubekir Kartal, Burhan Üstübi, Abdullah Furkan Aslan
- ❖ **No** : 2112903004, 2012903069, 2112903044
- ❖ **Bölüm** : Bilişim Sistemleri ve Teknolojileri 3 / B
- ❖ **Ders** : Mikroişlemciler Uygulama

Deneyin Adı

Deney: Ultrasonik Mesafe modülü uygulaması

Senaryo: Ultrasonik mesafe sensörü ile bir engele olan mesafe ölçülerek OLED ekranda yazdırılır. Engel mesafesi 10 cm' den daha az olması durumunda buzzer sesli uyarı verir.

Kazanımlar:

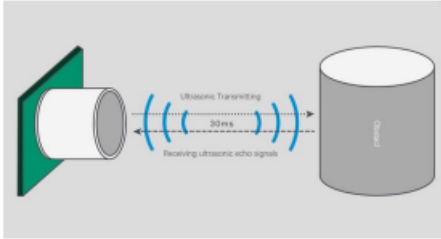
- ✓ Ultrasonik Mesafe Modülü Çalışma Mantığı
- ✓ Ultrasonik Mesafe Module Kullanımı

Kullanılan Malzemeler

- PinoLab-CodeBoard.Micro
- PinARM-STM32F103C8T6
- Ultrasonik Mesafe Module
- Bağlantı kablosu

Genel Bilgiler

Ultrasonik Mesafe Modülü (HC-SR04)

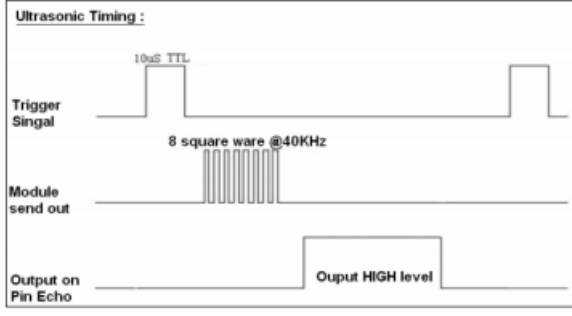


Şekil 1: HC-SR04

Şekil 1' de gösterilen HC- SR04 Ultrasonik Sensör sonar iletişim kullanarak karşısındaki nesneye olan mesafeyi hesaplayan bir kaynaktır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığı hesaplamamıza yardımcı olur. Sensörün çalışma prensibi basittir. TRIG pininden sinyal verildiğinde 40 Khz frekansında bir ses dalgası sensör tarafından üretilir ve bu ses dalgası bir cisme çarpıp geri döndüğünde ECHO pini aktif hale gelir. Sesin havada yayılma hızını bildiğimiz için TRIG pinine verilen sinyalden sonra ECHO pininin aktif olduğu zaman kadar ki süreyi ölçerek aradaki mesafe kolayca hesaplanabilir.

Ultrasonik Mesafe Modülü (HC-SR04) nasıl çalışır?

Hc-sr04 ultrasonik mesafe sensörünün zaman diyagramı aşağıda gösterilmektedir. Ölçümün başlaması için Trig pini en az 10 uS de +5V almalıdır. Böylelikle sensörden 40 KHZ'de 8 devir ses dalgası dışarıya iletilir ve yansıması beklenir. Hc-sr04 alıcıdan ses dalgasını aldığıında Echo pini 0V dan 5V a geçer ve mesafeyle orantılı bir süre bekler. Biz burada Echo pinindeki genişliği ölçerek , aradaki mesafeyi elde edebiliriz.



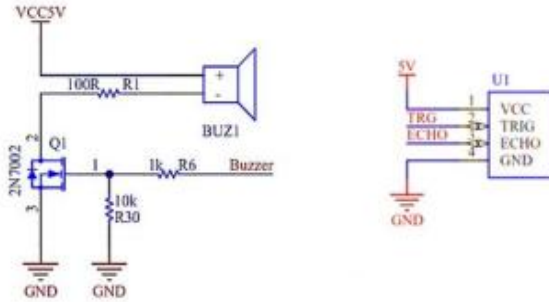
Trigger Sinyal = Trig pini aktif olduğunda gönderilen 1 adet sinyali ifade eder. Module send out = Trig pini aktif olduğunda tek seferde 40 kHz de 8 adet kare dalga sinyal gönderir. Output on pin 9b= Yansıyan dalgaların dönüşte 9b pininde oluşturduğu sinyali ifade eder

DENEY SETİ GÖRSELİ (AÇIK DEVRE)



Şekil 2:PinoLab-CodeBoard.Micro üzerindeki Ultrasonik Mesafe Module

Şekil 3:Ultrasonik Mesafe Modül açık devre şeması



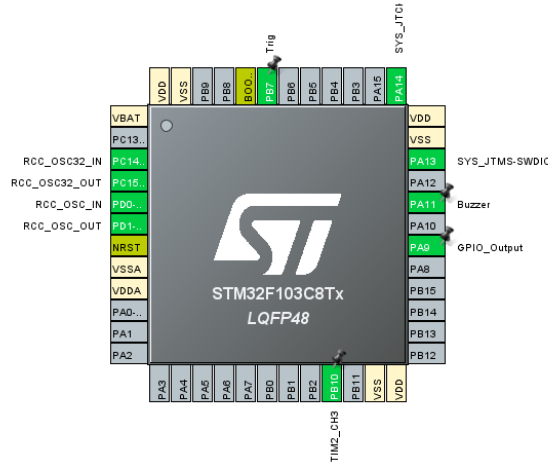
Ultrasonik Mesafe Sensörü Bağlantı Pinleri

- ULTRASONIC M. TRIG PB7
- ULTRASONIC M. ECHO PB10 Timer olarak ayarlanır.
- BUZZER

Donanım Bağlantıları

Şekil 6. PinoLab-CodeBoard.Micro üzerindeki Ultrasonik Mesafe Modülü (HC-SR04),switch konumları

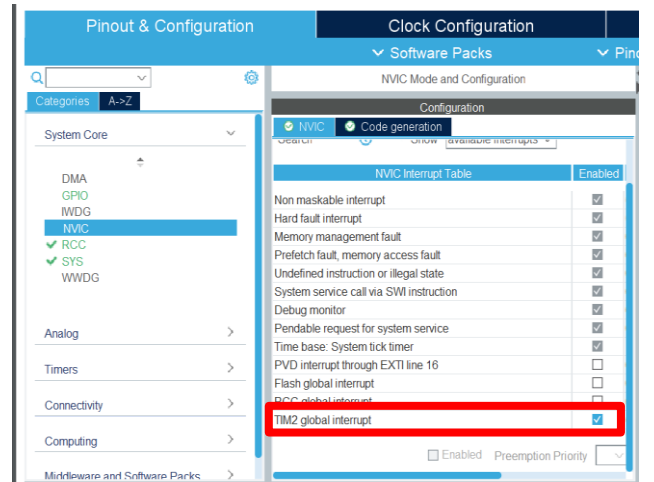
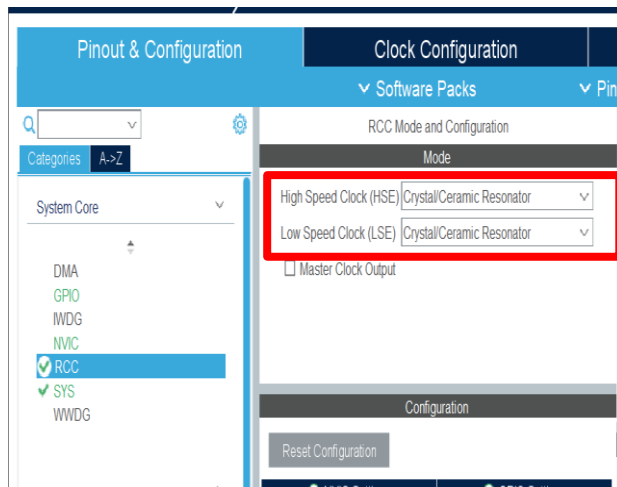
NOT: Uygulamaya ait algoritma oluşturulurken, ultrasonik mesafe sensörü ile ilgili verilen bilgiler mutlaka göz önünde bulundurulmalıdır. Echo ve Trig pinlerinin çalışma mantığına dayanarak, TIM birimi ile aradaki süre ölçülmelidir. Bunu için gönderme, okuma ve aradaki farkın hesaplanması işlemleri yapılır. Aradaki fark TIM ile süre cinsinden belirlendiğinden, bu fark (Difference) ilerleyen adımlarda gerekli matematiksel işlemler yapılarak mesafe cinsinden (Distance) elde edilir. TIM birimi ve I2C haberleşme protokolü kullanılırken, kanal aktivasyonu mutlaka yapılmalıdır. Kanal polaritesi gelen sinyal yükselen veya düşen kenara göre ayarlanmalıdır.



Şekil 7. Pin durumları

GENEL UYGULAMA KONFİRASYONLARI

NOT: Gösterim yapılan konfigrasyonlar dışında ayarlar defaulttur.



Mode

Debug

Serial Wire

☐ System Wake-Up

Timebase Source

SysTick

Categories

System Core

DMA

GPIO

IWDG

NVIC

✓ RCC

✓ SYS

WWDG

Analog

Configuration

Group By Peripherals

✓ GPIO

✓ RCC

✓ SYS

✓ TIM

Search Signals

Search (Ctrl+F)

☐ Show only Modified Pins

Pin...	Signal ...	GPIO ...	GPIO ...	GPIO ...	Maxim...	User L...	Modified
PA9	n/a	Low	Output ...	No pull...	Low		<input type="checkbox"/>
PA11	n/a	Low	Output ...	No pull...	Low	Buzzer	<input checked="" type="checkbox"/>
PB7	n/a	Low	Output ...	No pull...	Low	Trig	<input checked="" type="checkbox"/>

UYGULAMA KODLARI

Main.c

```
/* Includes -----*/
#include "main.h"
/* Private variables -----*/
TIM_HandleTypeDef htim2;
/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_TIM2_Init(void);
/* USER CODE BEGIN PFP */
/* USER CODE END PFP */
/* Private user code -----*/
/* USER CODE BEGIN 0 */
uint32_t IC_Val1 = 0;
uint32_t IC_Val2 = 0;
uint32_t Difference = 0;
uint8_t Is_First_Captured = 0;
uint8_t Distance = 0;
char mesafe[14];
void delayuS(uint16_t time)
{
    __HAL_TIM_SET_COUNTER(&htim2, 0);
    while (__HAL_TIM_GET_COUNTER(&htim2) < time);
}
void HAL_TIM_IC_CaptureCallback(TIM_HandleTypeDef *htim) {
    if (htim->Channel == HAL_TIM_ACTIVE_CHANNEL_3) {
        if (Is_First_Captured == 0) {
            IC_Val1 = HAL_TIM_ReadCapturedValue(&htim2, TIM_CHANNEL_3);
            Is_First_Captured = 1;
            __HAL_TIM_SET_CAPTUREPOLARITY(&htim2, TIM_CHANNEL_3, TIM_INPUTCHANNELPOLARITY_FALLING);
        } else if (Is_First_Captured == 1) {
            IC_Val2 = HAL_TIM_ReadCapturedValue(&htim2, TIM_CHANNEL_3);
            __HAL_TIM_SET_COUNTER(&htim2, 0);

            if (IC_Val2 > IC_Val1) {
                Difference = IC_Val2 - IC_Val1;
            } else if (IC_Val1 > IC_Val2) {
                Difference = (0xFFFF - IC_Val1) + IC_Val2;
            }

            Difference = Difference * 0.034 / 2;
            Distance = (uint8_t)Difference;
            Is_First_Captured = 0;

            __HAL_TIM_SET_CAPTUREPOLARITY(&htim2, TIM_CHANNEL_3, TIM_INPUTCHANNELPOLARITY_RISING);
        }
    }
}
```

```

        __HAL_TIM_DISABLE_IT(&htim2, TIM_IT_CC3);
    }
}
}

void HCSR04_Read(void) {
    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 0);
    delayuS(2);

    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 1);
    delayuS(10);

    HAL_GPIO_WritePin(GPIOB, GPIO_PIN_7, 0);

    __HAL_TIM_ENABLE_IT(&htim2, TIM_IT_CC3);
}
/* USER CODE END 0 */
int main(void)
{

    /* Initialize all configured peripherals */
    MX_GPIO_Init();
    MX_TIM2_Init();
    /* USER CODE BEGIN 2 */
    HAL_Delay(200);
    HAL_TIM_IC_Start_IT(&htim2, TIM_CHANNEL_3);

    /* USER CODE END 2 */

    /* Infinite loop */
    /* USER CODE BEGIN WHILE */
    while (1) {
        /* USER CODE END WHILE */

        /* USER CODE BEGIN 3 */
        HCSR04_Read();
        if (Distance <= 14) {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 1);
        } else {
            HAL_GPIO_WritePin(GPIOA, GPIO_PIN_9, 0);
        }
        HAL_Delay(300);
    }
    /* USER CODE END 3 */}

```


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. Tüm ayarlamalar yapıldıktan sonra “Kaydet” butonuna tıklanır ve kodlar program tarafından oluşturulur.
 5. Kodlar oluşturulduktan sonra kod penceresinde (main.c) kullanıcının kod yazmasına izin verilen alanlara gerekli kodlar yazılır.
- NOT: Kullanıcının yazacağı kodlar yalnızca izin verilen alanlara yazılmalıdır. Aksi takdirde yazılan bütün kodlar SİLİNECEKTİR.
6. Kodlar derlenir. Kodlarda herhangi bir hata var ise hatalar düzeltilir.
 7. PinARM-STM32F103C8T6 kartı PinoLab-CodeBoard.Micro üzerine takılır.
 8. Modül portuna ait switchler ON konumuna, bunun haricindeki tüm switchler OFF konumuna getirilir.
 9. Bağlantı kablosu takılır.
 10. Derlenen kodlar yüklenir.
 11. Kodlarda herhangi bir hata yok ise Ultrasonik mesafe sensörü, engele olan mesafeyi ölçecek ve engel 10 cm’ den daha yakın ise buzzer sesli uyarı verecektir.

Kaynakça

- 10-STM32_Hafta10(HC-SR04) Pdf’i
- İnternet