

ALİ GENÇ

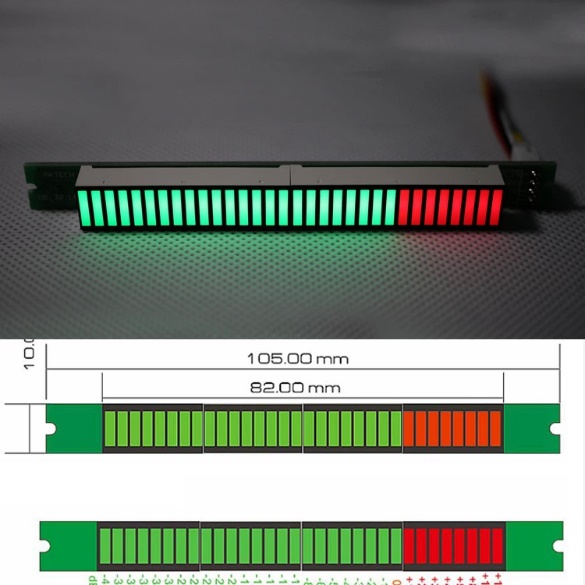
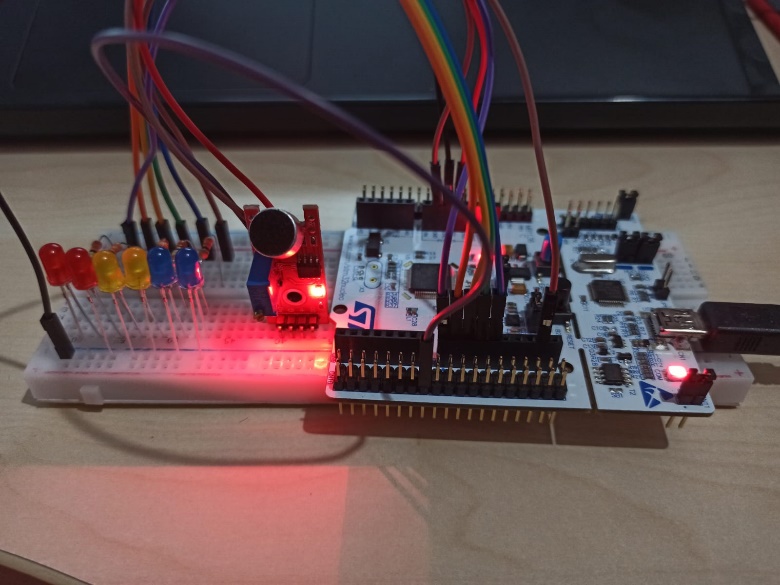
193302031

NORMAL ÖĞRETİM

GÖMÜLÜ SİSTEMLER FİNAL ÖDEVİ

**PROJE TANITIMI**

KY038 kullanarak ortamdan alınan ses şiddeti verileri kaydedilir. Bu veriler belli değer aralıklarına bölünür. Her led bir değer aralığını temsil eder. Ses şiddetine göre artan veri değeri, sistemdeki yanan led sayısını da artırır. Yani ortamdaki ses şiddeti ne kadar fazlaysa sistemde yanan led sayısı da o kadar artacaktır. Halk arasında bu sisteme “vumetre” denilmektedir. Piyasada hazır modül halinde de bulunmaktadır. Ancak bu projede bu sistemi STM32f103RBT6 kartı ile yapacağız.

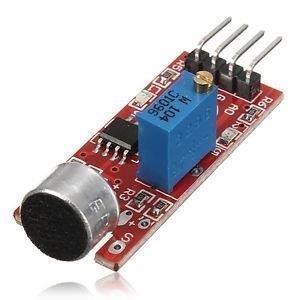
 

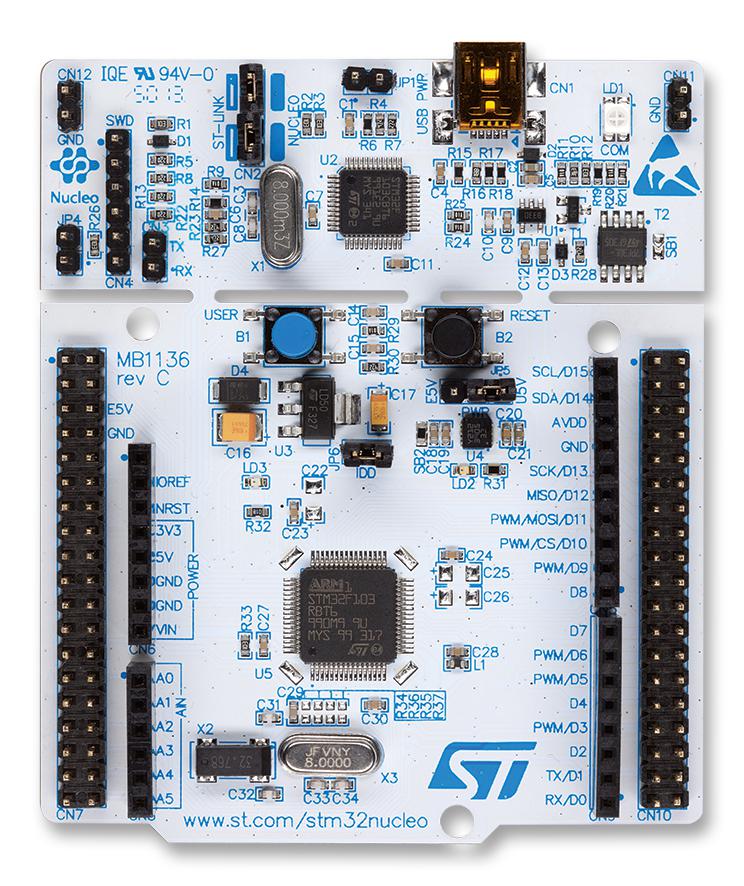
Projesini gerçekleştirdiğimiz Vumetre

Piyasada bulunan örnek bir Vumetre

**PROJEDE KULLANILAN MALZEMELER**

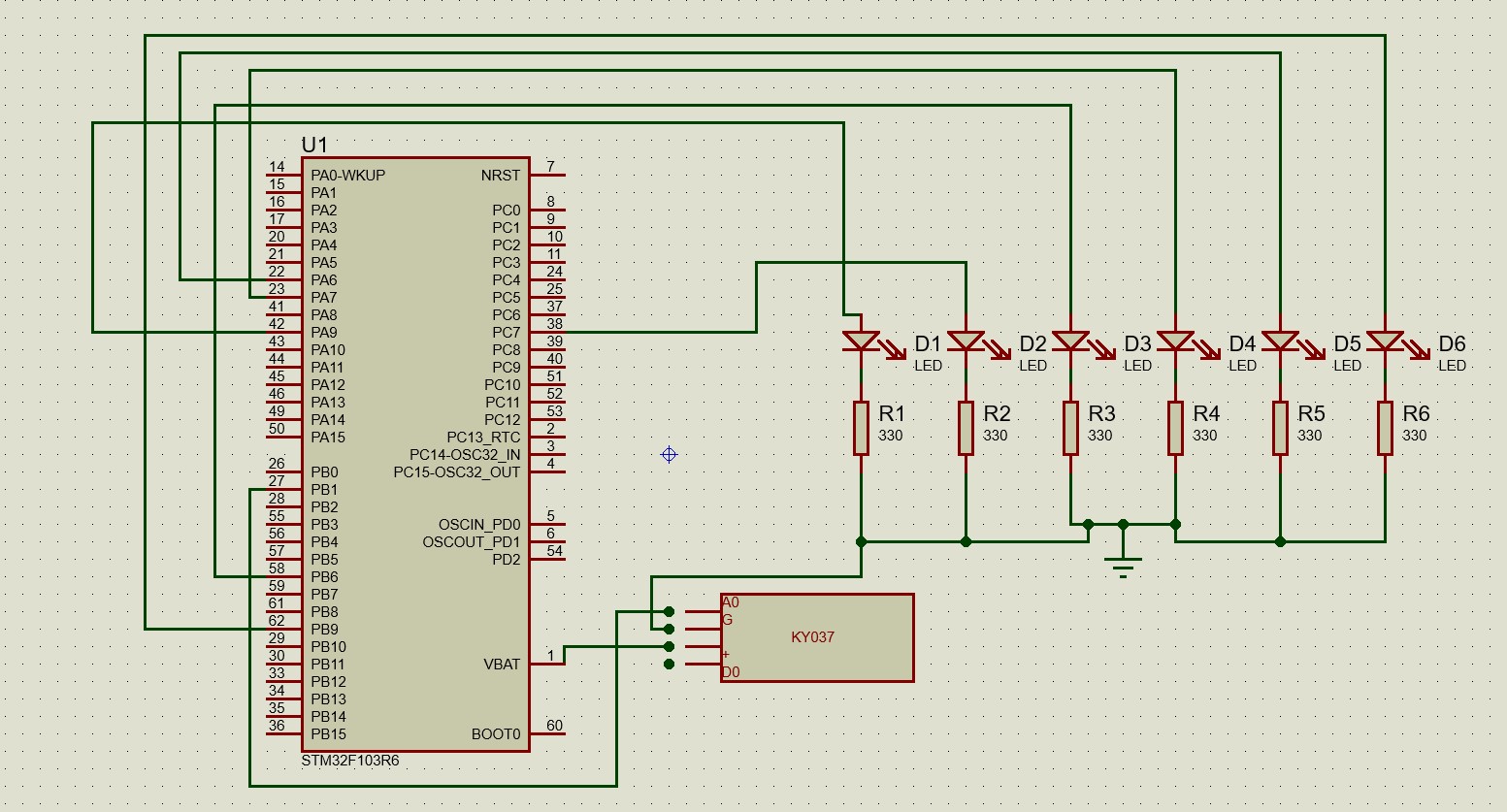
* STM32F103RBT6 Kartı
* KY 038 ses sensörü modülü





* 6 adet led
* 6 adet 330R direnç
* Jumper kablolar

**PROJE TASARIMI**



Projemizin Proteus çizimleri bu şekilde gözüküyor. Bağlantıları bu şekilde yapacağız.

**PROJE KODLARI VE AÇIKLAMALARI**

Öncelikle projemizde kullanacağımız KY037-KY038 modülümüzün bağlantılarını yapıyoruz. Modülümüzün A0 pinini Stm32 kartımızın PB1 pinine bağlıyoruz. Böylece modülümüzden aldığımız analog verileri kartımızda dijital verilere çevirebileceğiz. 6 adet ledimizi eksi bacakları gnd hattına gelecek şekilde breadboarda yerleştiriyoruz. Artı bacaklarının ucuna da 330R lik dirençler ekliyoruz. Dirençlerin ucuna da jumper kablolarımızı yerleştiriyoruz. Bu jumperları da kartımızda daha düzenli durması adına sıralı GPIO pinlerine bağlıyoruz. Bu pinler sırasıyla PA9, PC7, PB6, PA7, PA6, PB9 olacaktır. Tasarımımızı tamamladıktan sonra kodlama aşamamıza geçiyoruz.

Ioc sayfamızdan belirtmiş olduğum PA9, PC7, PB6, PA7, PA6, PB9 pinlerini seçip GPIO output olarak seçiyoruz. Soldaki Analog sekmesinden ADC1 i seçiyoruz. Ben IN9 u kullanmak için bu seçeneği işaretliyoruz. Alttaki configuration penceresinden parameter settings sekmesini seçiyoruz ve Continuous Conversion Mode’u aktif ediyoruz. NVIC sekmesindeki Global İnterrupts seçeneğini de işaretliyoruz. ADC sekmesiyle işimiz bitti.

Şimdi Timers sekmesinden TIM3 ü seçiyoruz ve trigger source u ITR1 olarak seçiyoruz. Clock source seçeneğini de Internal clock olarak değiştiriyoruz. Configuration penceresinden TIM3 global interrupt’ı aktif ediyoruz. Parameter settings sekmesindeki prescaler’ı 1000 olarak değiştiriyoruz. Counter periodu da 6553 yaptık. Bunları yapmamızın sebebi daha seri çevrimler yaparak istediğimiz değerleri anlık olarak almasını istememizdir.

Generate kod dediğimizde önümüze çıkan main.c sayfasına ADC değişkenimizi tanımlıyoruz.

/\* USER CODE BEGIN PV \*/

uint16\_t readValue=0;

/\* USER CODE END PV \*/

ADC ve Timerımızı çalıştırmak için de bu kodları yazıyoruz.

/\* USER CODE BEGIN 2 \*/

HAL\_ADC\_Start\_IT(&hadc1);

HAL\_TIM\_Base\_Start\_IT(&htim3);

/\* USER CODE END 2 \*/

Stm32f1xx sayfasını açıp ADC değerlerimizi aktardığımız değişkeni yani readValue değerini extern ediyoruz.

**extern** uint16\_t readValue;

**void** **ADC1\_2\_IRQHandler**(**void**)

İf ve if else komutlarıyla ADC den çektiğimiz değerleri belirli aralıklara bölüp istediğimiz sonuçları elde edene kadar sınır değerlerini değiştirdik. Çalışmasında sıkıntı yok ancak daha hassas bir ölçüm için sınır değerleriyle oynanabilir.

readValue = HAL\_ADC\_GetValue(&hadc1);

**if** (readValue<=2500)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 0);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=2500 && readValue<=3010)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=3010 && readValue<=3050)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=3090 && readValue<=3130)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 0);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=3130 && readValue<=3170)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 0);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=3170 && readValue<=3200)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 0);

}

**else** **if** (readValue>=3200)

{

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_9, 1);

HAL\_GPIO\_WritePin(GPIOC, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_6, 1);

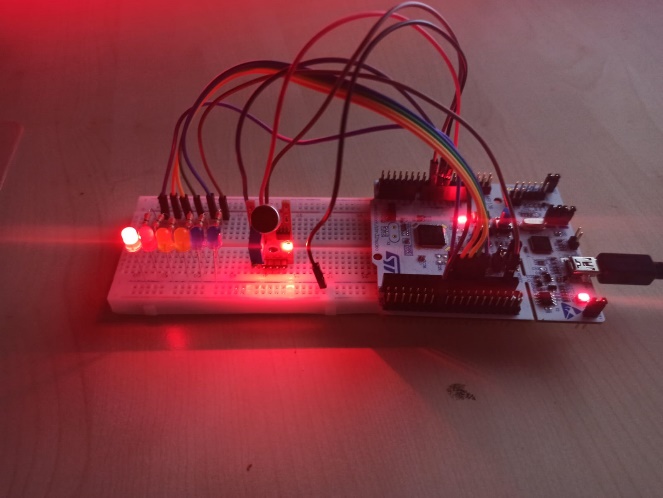
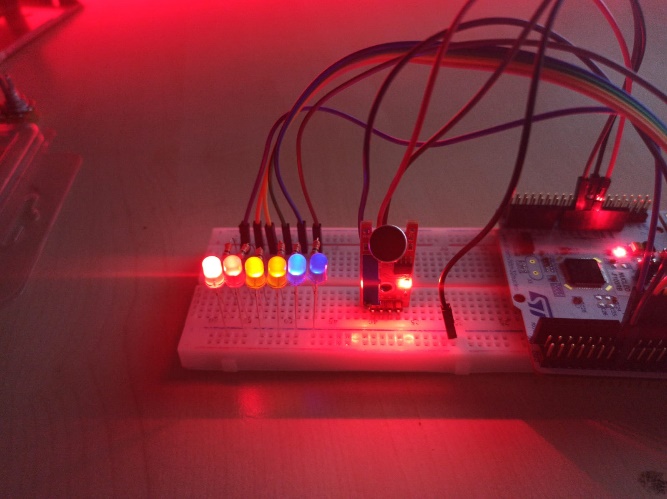
HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_7, 1);

HAL\_GPIO\_WritePin(GPIOA, GPIO\_PIN\_6, 1);

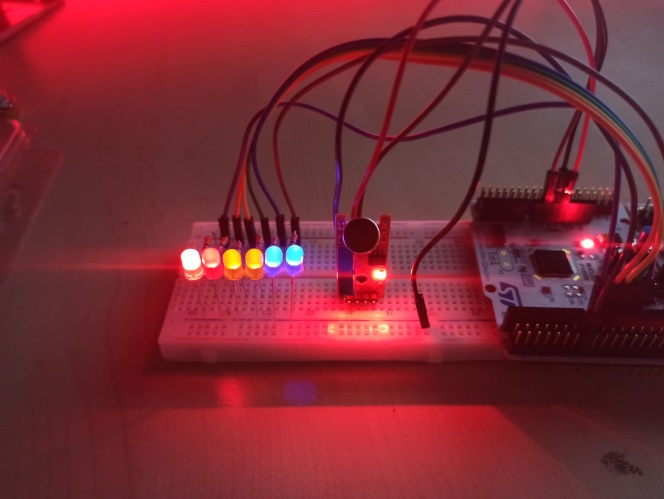
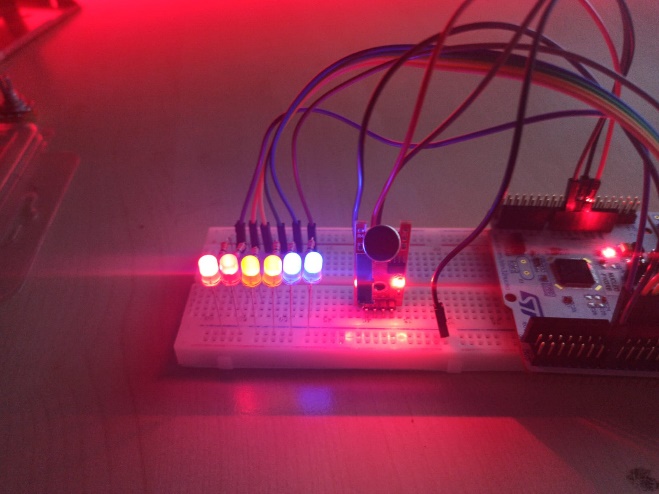
HAL\_GPIO\_WritePin(GPIOB, GPIO\_PIN\_9, 1);

}

/\* USER CODE END ADC1\_2\_IRQn 0 \*/

Belirlediğim aralıklarda ses şiddetine göre yanan ledler

Projenin video linki aşağıda yer alıyor.

https://drive.google.com/drive/folders/1sghqtlZJUihfH1tM6refw9zcTOrhCGFH?usp=sharing