

ÖZET

Bu projede ARM tabanlı mikrodenetleyici kullanılarak bir güneş enerjisi ölçüm cihazı yapılmaya çalışılmıştır. Elde edilen ölçüm sonuçları anlık olarak bir LCD ekranda gösterilmiştir. Güneş enerjisi ölçüm cihazı ile belirli bir bölgedeki güneş enerjisinden elde edilebilecek elektrik enerjisi potansiyeli hakkında fikir sahibi olunabilir.



Güneş Enerjisi Ölçüm Cihazı

1-KULLANILAN DONANIMLAR

1.1-STM32F407VG DISCOVERY

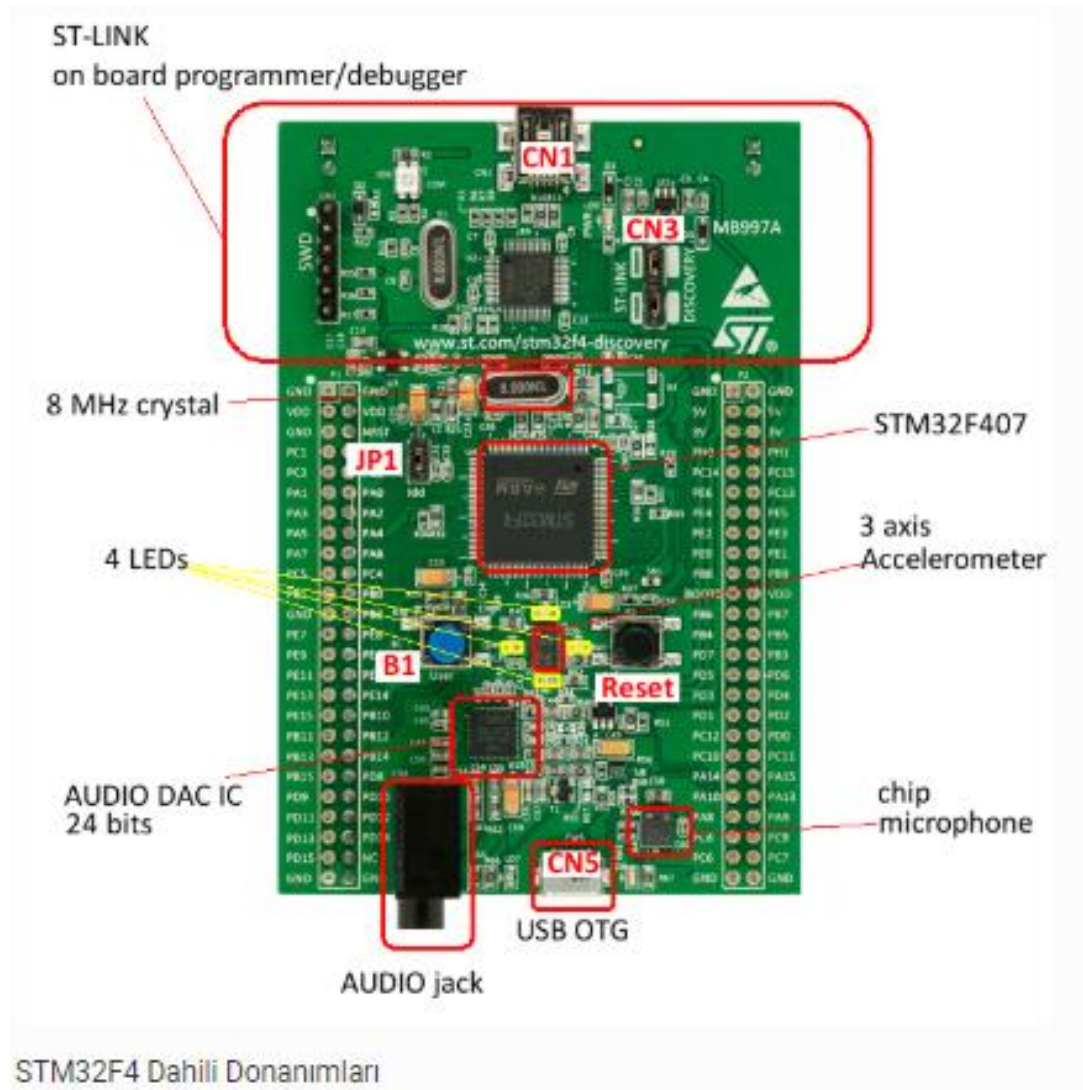
STM32F4 Discovery Kit kullanıcıların 32 bit ARM Cortex-M4F tabanlı STM32F406 ve STM32F417 serisi mikrokontrolcüler üzerinde geliştirme yapabilmesi için tasarlanmış basit ve düşük maliyetli bir kittir.

Bütün geliştirme kartı kitlerinde olduğu gibi STM32F4 geliştirme kartı da başlangıç düzeyindeki ve ileri seviyedeki uygulama geliştiricilerin kullanımına sunulmuştur.

Bu geliştirme kartı;

- 32-bit ARM Cortex-M4F Core
- 1Mb Flash Memory
- 192 KB RAM(LQFP100 Kılıf)
- Dahili seçim moduna sahip ST-LINK/V2 (SWD konektör ile programlama ve debug)
- USB den Bus Power sağlayabilir ya da harici 5V beslemesi yapabilir.
- Uygulamalarınız için harici 3V ve 5V besleme
- ST elektroniğin MEMS (mikro elektromekanik sensör) tabanlı dijital ivmeölçeri olan LIS302DL ya da LIS3DSH
- ST elektroniğin MEMS tabanlı dijital mikrofonu olan MP45DT02
- CS43L22 audio DAC D class speak sürücüsünü

- 4 Adet kullanıcı kontrolünde olan LED
- Biri Reset olmak üzere 2 adet pushbutton
- STM32F405 , STM32F407, STM32F415, STM32F417 mikrokontrolcileri 32 bit ARM Cortex-M4 RISC core (168 MHz e kadar) olarak yapılandırılmıştır.
- 3 adet 12 -bit ADC
- 2 DAC
- 1 low -power RTC
- 12 adet genel kullanım amaçlı 16-bit timer bunların ikisi motor kontrolü için PWM timerdır.
- 2 adet genel kullanım amaçlı 32-bit timer
- Gerçek rastgele sayıcı üretici (RNG)
- Standart gelişmiş haberleşme arayüzü içerir.



1.2- 16X2 LCD EKRAN

Açılımı Liquid Crystal Display olan LCD bir ekran türüdür.

Cep telefonlarından televizyonlara, uçuş kontrol panellerinden kol saatlerine kadar çok geniş bir sahada kullanılmaktadır.

LCD ekranlar üzerindeki her bir piksel, iki şeffaf elektrod ve iki polarize filtre arasında bir molekül katmanından oluşur.

Şeffaf elektrodlerden giren ışık polarize filtrede tutularak piksele görüntü verilmiş olur.

Renkli LCD'lerde ise her bir piksel, üç ana rengi temsil eden hücreye bölünmüştür. Milyonlarca farklı renk görüntüleyebilmek için her hücre ayrı bir elektronik devreyle kontrol edilir.

16x2 Karakter LCD Ekran Display 1602 Teknik Özellikleri:

- +5V ile çalışmaktadır.
- Back Lighting özelliğine sahiptir.
- LCD arka fon ışığı olmadan 4mA akım çekmektedir.
- Boyutları 80x36x9.4mm'dir.
- Çalışma sıcaklığı -20 ile +70 derece arasındadır.

Pin Bağlantıları:

Vss-> GND

Vdd-> +5V

Vo-> Kontrast ayarı

RS-> Register seçme pini

R/W-> Data read/write pini.

E-> Enable pini

DB0-> Data Bus 0

DB1-> Data Bus 1

DB2-> Data Bus 2

DB3-> Data Bus 3

DB4-> Data Bus 4

DB5-> Data Bus 5

DB6-> Data Bus 6

DB7-> Data Bus 7

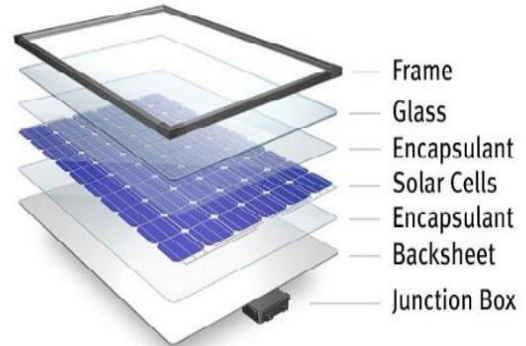
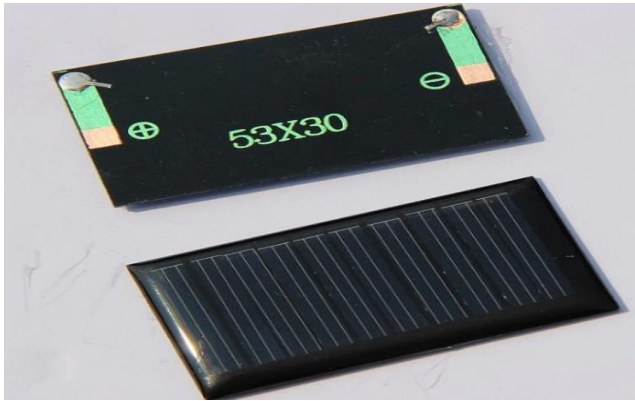
A-> LED arka ışık anot (+5V)

K-> LED arka ışık katot (GND)



1.3-Güneş Paneli

Güneş panelleri dünyaya gelen güneş ışıklarını elektriğe çevirirler. Güneş paneli üzerine gelen güneş ışığını emmeye yarayan solar hücrelerden oluşur, bu hücrelerin birleşmesi ile Güneş Paneli oluşur. Bir güneş panelinin verimi, kullanıldığı hücrenin güneş ışığını elektriğe dönüştürme oranı ile belirlenir. Devremizde kullanılan güneş paneli hesap makinelerinde kullanılan mini güneş paneli (fotovoltatik pil) dir.



2- KULLANILAN PROGRAMLAR

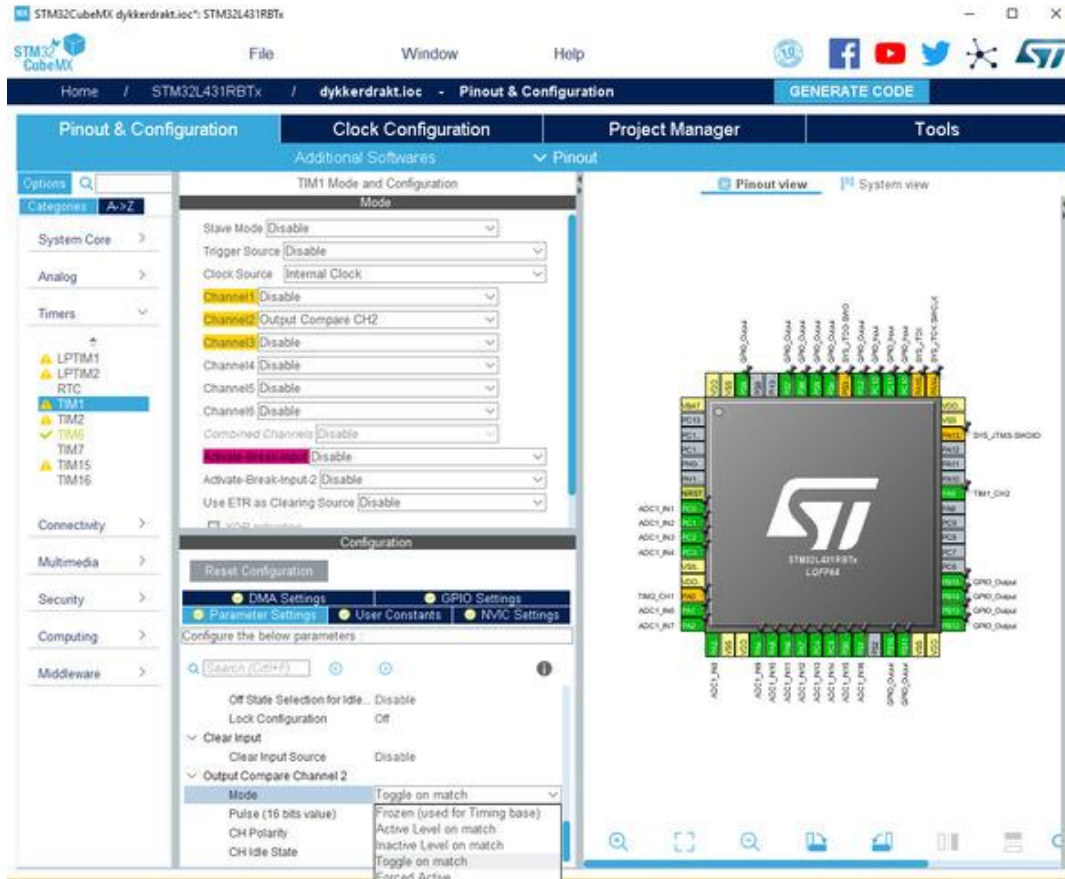
2.1- ATOLLIC TRUE STUDIO

True Studio, yüksek kaliteli gömülü yazılımların geliştirilmesine yardımcı olmak için STM32 MCU geliştiricilerine yönelik esnek bir geliştirme ve hata ayıklama IDE'sidir. True Studio açık standartlara (ECLIPSE ve GNU) dayanmaktadır.



2.2- STM32 CubeMX

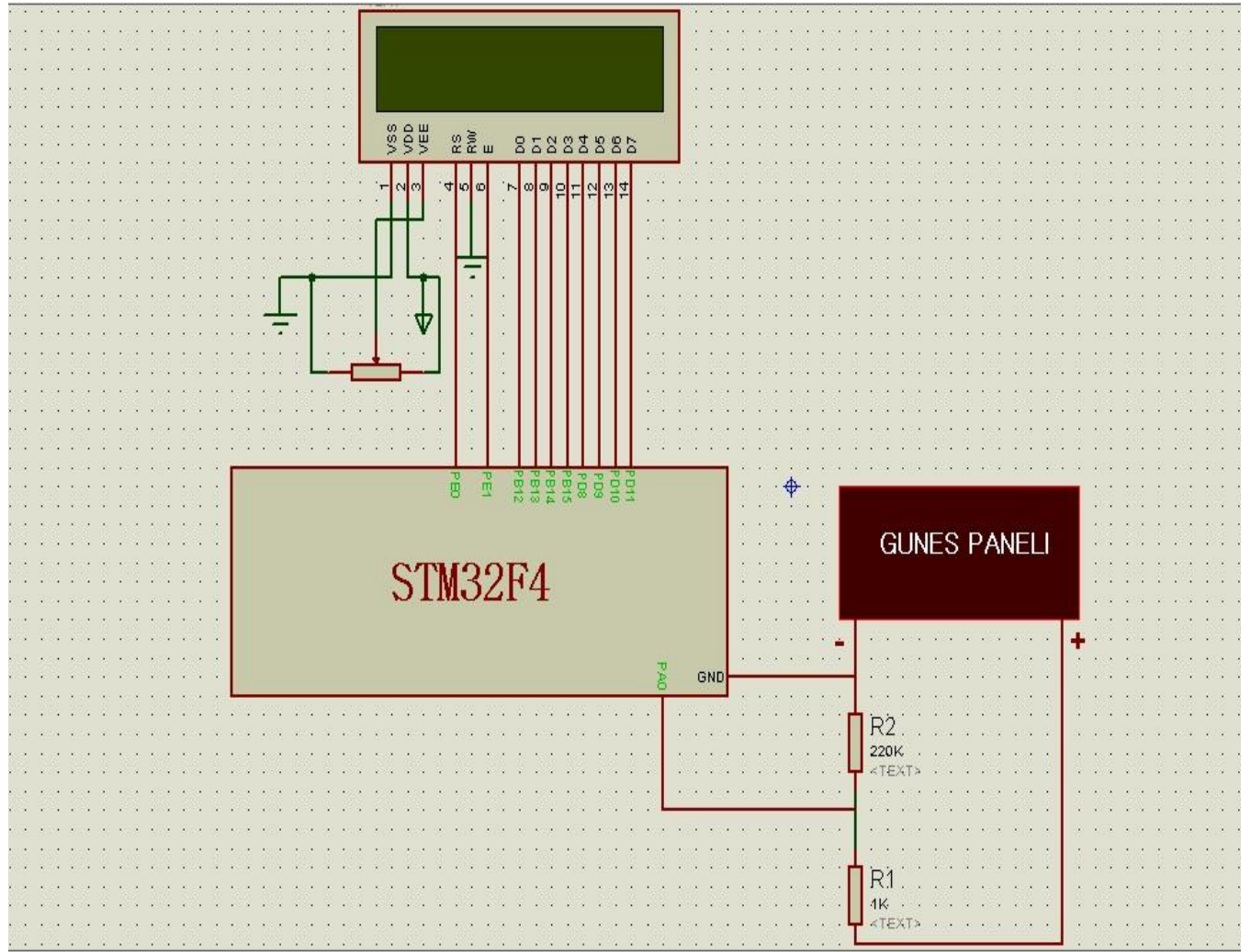
STM32CubeMX STM32 mikroişlemcisi ve mikrodenetleyicisi için gerekli olan konfigürasyon ayarlarını görsel olarak yapmayı ve ARM için karşılık gelen başlatma C kodu üretilmesini sağlayan bir grafik araçtır.



3- DEVRENİN KURULUMUMU

Devre bir breadboard üzerine aşağıdaki adımlar çerçevesinde kurulmuştur.

- I. STM32f4 karta 8 bit gönderim modunda olmak üzere 16*2 LCD ekranın bağlanması.
- II. Mini güneş paneline uygun gerilim bölücü devrenin kurulması.
- III. Güneş paneli ile analog bir pin üzerinden STM32f4 kartın bağlantısının yapılması.
- IV. Gerekli yazılımların yüklenmesinin ardından sisteme güç verilip çalıştırılması.



DEVRE ŞEMASI

4- KONFİGÜRASYON AYARLARI

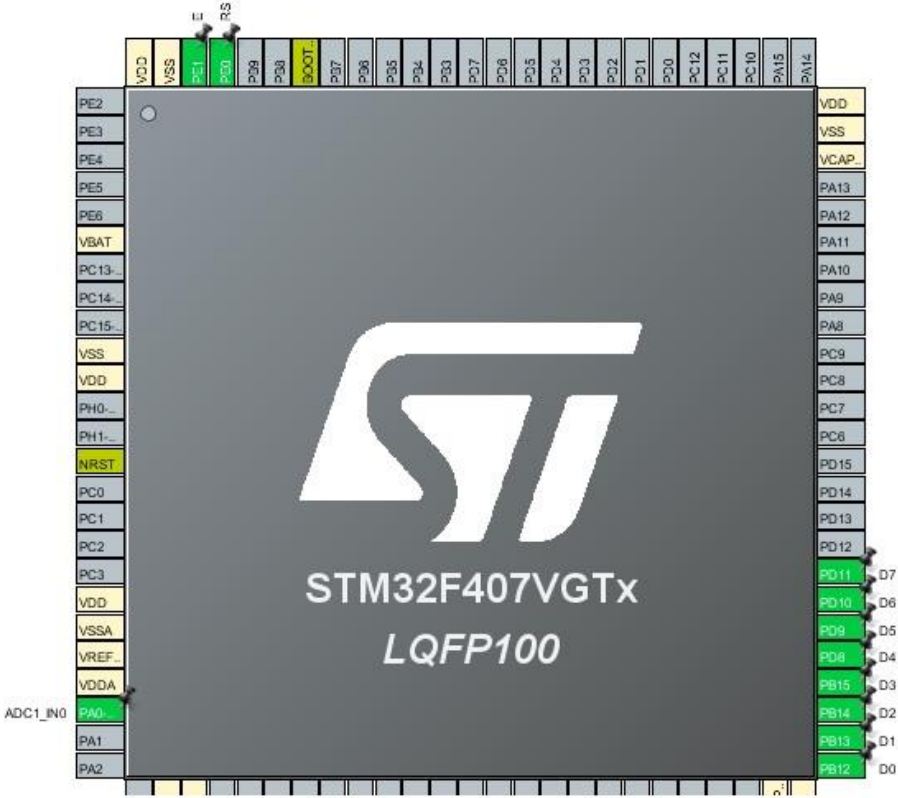
LCD ve güneş panelinin bağlantısında kullanılan pinlerin ve STM32F4 kartı için gerekli olan tüm Clock ayarlarının yapılabilmesi için bir dizi konfigürasyon işleminin yapılması gerekmektedir. Bu konfigürasyon işlemi için CubeMX programı kullanılmıştır.

4.1-PİN KONFİGÜRASYONLARI

LCD pinleri stm32f4 kartın pinleriyle aşağıdaki şekilde eşleştirilmiştir. Tüm pinler için **GPIO_OUT** modu seçilmiştir.

LCD	STM32F4
RS	PE1
E	PE0
D0	PB12
D1	PB13
D2	PB14
D3	PB15
D4	PD8
D5	PD9
D6	PD10
D7	PD11

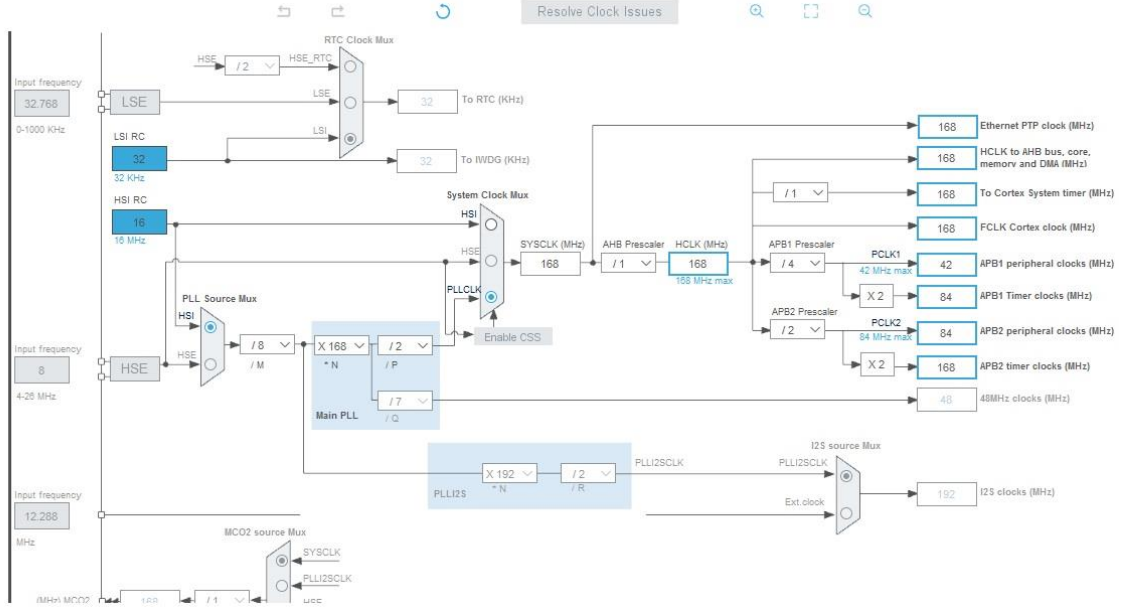
Bu pinlerin dışında Güneş panelinden analog ölçüm almak için PA0 pinini analog input olarak ayarlanmıştır.



CubeMX ÜZERİNDEN GÖRÜNEN PIN SEÇİMLERİ

4.2- CLOCK AYARLARI

Yine CubeMX üzerinden tercih edilen clock konfigürasyonu görsel olarak yapılabilmektedir. Bu projede aşağıdaki görselde görüleceği gibi dahili osilatör seçilmiştir.

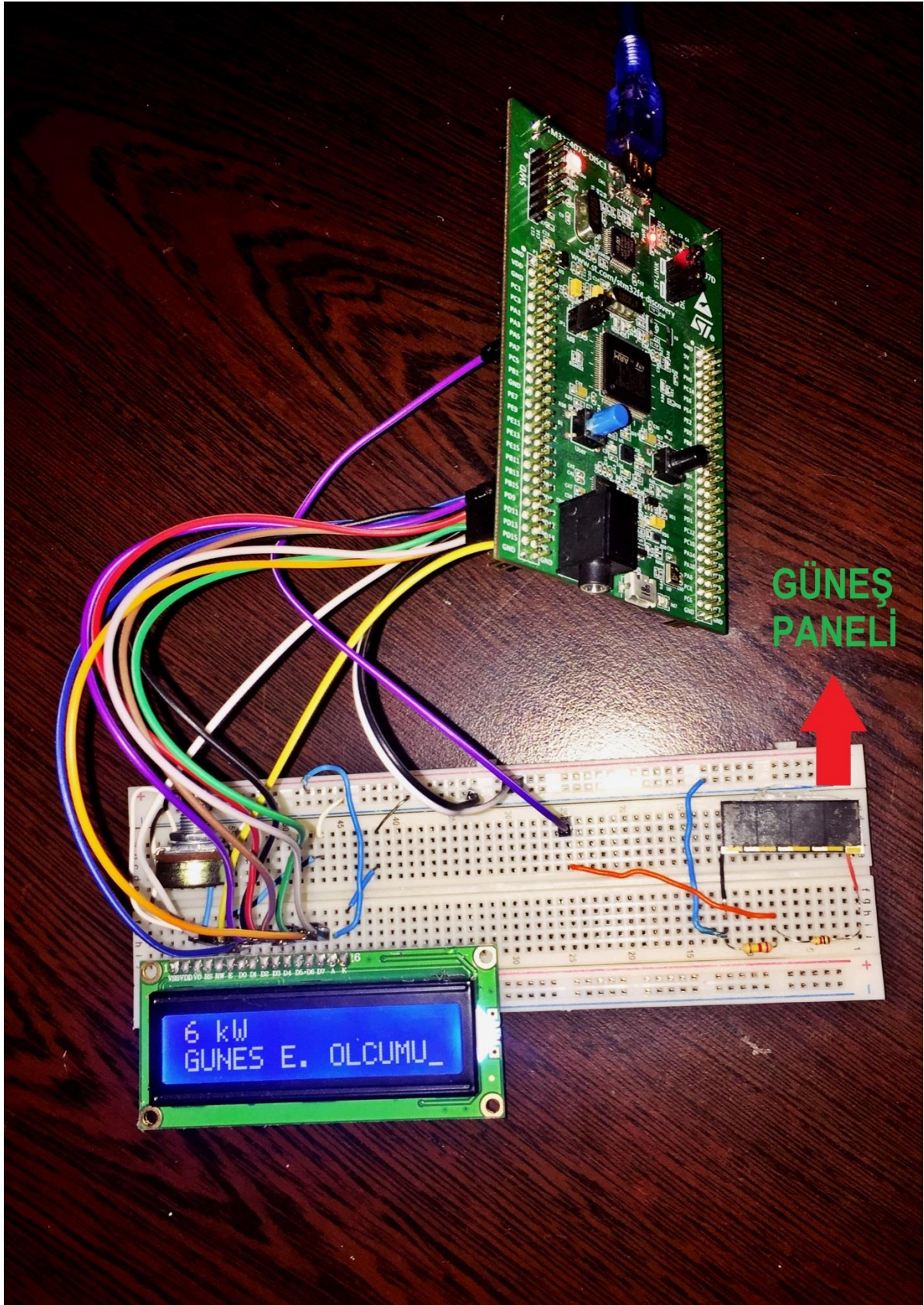


CubeMX ÜZERİNDEN YAPILAN CLOCK KONFIGÜRASYONLARI

5- DEVRENİN UYGULANMASI

Devre breadboard üzerine aşağıdaki gözüktüğü haliyle kurulmuş ve farklı ışık yoğunluklarında farklı güç değerleri verdiği gözlemlenmiştir. Devrenin çalışmasını ait video burada paylaşılmıştır:

https://drive.google.com/file/d/1vhO-qbpTlhB8xLiqyK4fAHGgeI3mm_H/view



DEVRENİN ÇALIŞMASI

6- PROGRAMIN YAZILMASI

Program C dilinde Atollic True Studio IDE'si ile yazılmıştır. Arm için donanım soyutlama kütüphanelerinden olan HAL kütüphanesi tercih edilmiştir. Ayrıca LCD için “STM_MY_LCD16x2” kütüphanesinden yararlanılmıştır. Programın kaynak kodları aşağıda paylaşılmıştır.

```
#include "main.h"
#include "stm32f4xx_hal.h"
#include "STM_MY_LCD16X2.h"
#include <stdlib.h>
#include <stdio.h>

ADC_HandleTypeDef hadc1;

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);

int gerilim; // güneş panelinden okunan analog deger
int enerji; // hesaplanan enerji (Watt)

int main(void)
{

    HAL_Init();

    SystemClock_Config();

    MX_GPIO_Init();
    MX_ADC1_Init();

    LCD1602_Begin8BIT(RS_GPIO_Port, RS_Pin, E_Pin, D0_GPIO_Port, D0_Pin, D1_Pin,
D2_Pin, D3_Pin, D4_GPIO_Port, D4_Pin, D5_Pin, D6_Pin, D7_Pin);
    LCD1602_noBlink(); // blink gizleme

    LCD1602_setCursor(1,1);
    LCD1602_print("Muhammed Deniz");
    LCD1602_setCursor(2,1);
    LCD1602_print("161005034");
    HAL_Delay(2000);
    LCD1602_clear();

    LCD1602_setCursor(1,1);
```

```
LCD1602_print("Ilayda Metin");
LCD1602_setCursor(2,1);
LCD1602_print("161006207");
HAL_Delay(2000);
LCD1602_clear();
```

```
LCD1602_setCursor(1,1);
LCD1602_print("Mehmet Bozkurt");
LCD1602_setCursor(2,1);
LCD1602_print("141005032");
HAL_Delay(2000);
LCD1602_clear();
```

```
LCD1602_1stLine();
LCD1602_print("DUZCE UNI. 2020");
LCD1602_2ndLine();
LCD1602_print("Microprocessor 2");
HAL_Delay(3000);
LCD1602_clear();
```

```
while (1) //loop
{
    HAL_ADC_Start(&hadc1);
    gerilim=HAL_ADC_GetValue(&hadc1);
    enerji=(gerilim/100);
    enerji*=2;
    LCD1602_setCursor(1,1);
    LCD1602_PrintInt(enerji);
    LCD1602_print(" kW");
    LCD1602_setCursor(2,1);
    LCD1602_print("GUNES E. OLCUMU");
    HAL_Delay(2000);
    LCD1602_clear();
}
```

```
void SystemClock_Config(void) //CubeMX tarafından olusturulan clock ayarlari
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    __HAL_RCC_PWR_CLK_ENABLE();
```



```

__HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);
/** Initializes the CPU, AHB and APB busses clocks
 */
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
RCC_OscInitStruct.HSISState = RCC_HSI_ON;
RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
RCC_OscInitStruct.PLL.PLLM = 8;
RCC_OscInitStruct.PLL.PLLN = 168;
RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
RCC_OscInitStruct.PLL.PLLQ = 7;
if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
{
    Error_Handler();
}
/** Initializes the CPU, AHB and APB busses clocks
 */
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                               |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) != HAL_OK)
{
    Error_Handler();
}
}

static void MX_ADC1_Init(void) //CubeMX tarafından olusturulan ADC ayarları
{

ADC_ChannelConfTypeDef sConfig = {0};
hadc1.Instance = ADC1;
hadc1.Init.ClockPrescaler = ADC_CLOCK_SYNC_PCLK_DIV4;
hadc1.Init.Resolution = ADC_RESOLUTION_12B;
hadc1.Init.ScanConvMode = DISABLE;
hadc1.Init.ContinuousConvMode = ENABLE;
hadc1.Init.DiscontinuousConvMode = DISABLE;
hadc1.Init.ExternalTrigConvEdge = ADC_EXTERNALTRIGCONVEDGE_NONE;

```

```

hadc1.Init.ExternalTrigConv = ADC_SOFTWARE_START;
hadc1.Init.DataAlign = ADC_DATAALIGN_RIGHT;
hadc1.Init.NbrOfConversion = 1;
hadc1.Init.DMAContinuousRequests = DISABLE;
hadc1.Init.EOCSelection = ADC_EOC_SINGLE_CONV;
if (HAL_ADC_Init(&hadc1) != HAL_OK)
{
    Error_Handler();
}

```

```

sConfig.Channel = ADC_CHANNEL_0;
sConfig.Rank = 1;
sConfig.SamplingTime = ADC_SAMPLETIME_3CYCLES;
if (HAL_ADC_ConfigChannel(&hadc1, &sConfig) != HAL_OK)
{
    Error_Handler();
}

}

```

static void MX_GPIO_Init(void) //CubeMX tarafından olusturulan LCD pin(GPIO_PIN) ayarları

```

{
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOE_CLK_ENABLE();
    HAL_GPIO_WritePin(GPIOB, D0_Pin|D1_Pin|D2_Pin|D3_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOD, D4_Pin|D5_Pin|D6_Pin|D7_Pin, GPIO_PIN_RESET);
    HAL_GPIO_WritePin(GPIOE, RS_Pin|E_Pin, GPIO_PIN_RESET);
    GPIO_InitStructure.Pin = D0_Pin|D1_Pin|D2_Pin|D3_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOB, &GPIO_InitStructure);
    GPIO_InitStructure.Pin = D4_Pin|D5_Pin|D6_Pin|D7_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOD, &GPIO_InitStructure);
    GPIO_InitStructure.Pin = RS_Pin|E_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;

```



```

GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

}

void Error_Handler(void)
{

}

#ifdef USE_FULL_ASSERT
void assert_failed(uint8_t *file, uint32_t line)
{

}

#endif

```

7- GELİŞTİRME VE ÖNERİLER

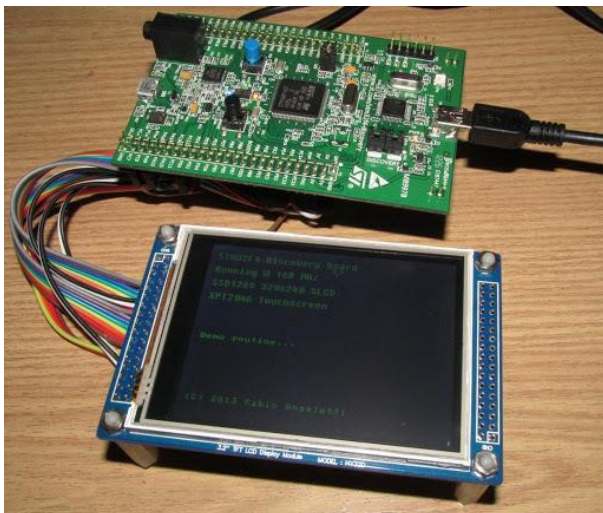
Fotovoltaik pilden okunan analog değerin tam olarak ne kadarlık bir elektirik enerjisi üretme potansiyeli olduğu çeşitli deneylerle doğru olarak hesaplanabilir.(Bu çalışmada temsili katsayılar kullanılmıştır).

Fotovoltaik panelden elde edilen analog değere karşılık gelen potansiyel elektrik enerjisinin **kW/m²** olarak LCD ekrana yazdırılması daha uygun ve gerçekçidir.

Sistem bir bölgede 24 saat süre ile sürekli çalıştırılarak belirli zaman aralıklarında alınan ölçüm değerlerinin (örneğin 5 dakikada 1) kaydedilmesi sağlanabilir. Daha sonra bu kayıtlar üzerinden çeşitli hesaplamalar yapılarak gün içersinideki ışık ve buna bağlı enerji potansiyeli yoğunluğu grafiksel ekranlarda gösterilebilir. Ekran olarak Discovery karta bağlanabilecek aşağıdaki gibi bir ekran kullanılabilir.

Gün boyu yapılan ölçümler aynı bölgenin farklı lokasyonlarında tekrarlanarak o bölgede güneş enerji paneli kurmak için en uygun yer otomatik olarak tespit edilebilir.

Güneş enerjisinden verimli elektrik elde etmek için bir diğer önemli faktör **sıcaklıktır**. Bu sebepten dolayı sisteme bir sıcaklık sensörü bağlanarak panelin sıcaklığı da gözönüne alınıp daha gerçekçi değerlendirmeler yapılabilir.



DISCOVERY İÇİN EKRAN ÖRNEĞİ

8- KAYNAKLAR

- 1- <http://akademik.duzce.edu.tr/fatihevran/Profil/DersGoruntule?dersID=113000>
- 2- https://www.st.com/resource/en/reference_manual/dm00031020-stm32f405415-stm32f407417-stm32f427437-and-stm32f429439-advanced-armbased-32bit-mcus-stmicroelectronics.pdf
- 3- <https://www.st.com/en/development-tools/truestudio.html>
- 4- <https://www.st.com/en/development-tools/stm32cubemx.html>
- 5- <https://www.youtube.com/watch?v=zfn5YqFlqbc>
- 6- <https://www.youtube.com/watch?v=z32WRmykjk>
- 7- ARM Cortex-M4 İle Mikrodenetleyici Programlama [kitap]