**GÖMÜLÜ SİSTEM TASARIMI**

**ÖDEV 2-Rapor**

## Osman ŞEYBAN

Video adresi: https://youtu.be/Gw6DJ3aHVI4

Projede yapılacaklar:

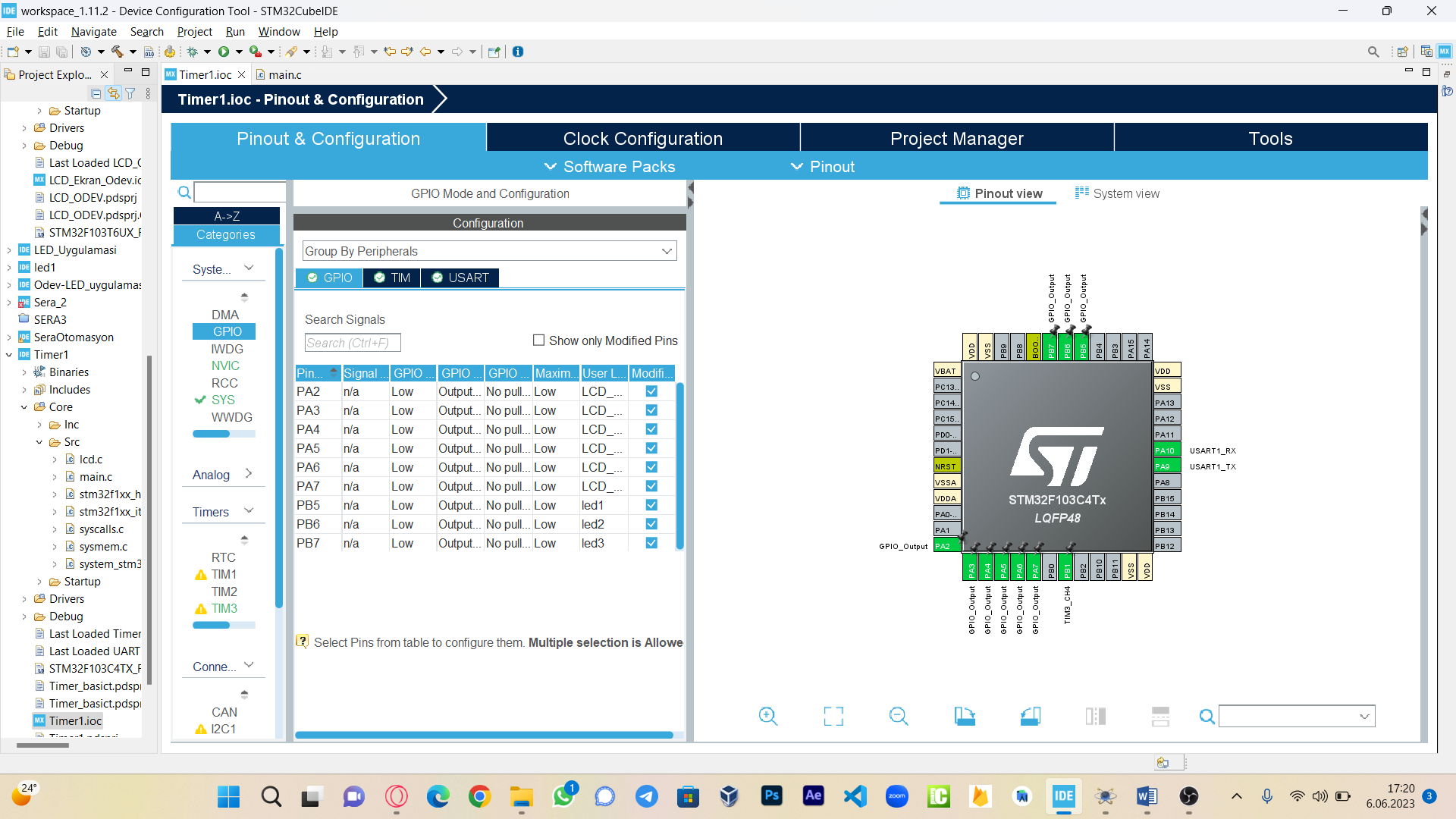
STM32CubeIDE kullanılarak bir proje oluşturacağız. Projede yapılacaklar şöyle:

1. Uygulamamızda 1 adet buton, 3 adet led, UART modülü ve timer modülü kullanılacaktır.
2. Başlangıçta UART modülü üzerinden önce “Osman” yazdırılacak, 2 sn sonra “ODEV 3” yazdırılacak.
3. Buton pull-up olarak koşullanacaktır. Buton kesme fonsiyonu ile okunacaktır.
4. Buton’a 1 kere basıldığında UART modülüne “buton’a basıldı” mesajı gönderilecek ve **timer3 koşullamaları yapılarak timer3 aktif edilecek. Butona tekrar basıldığında sayaç temizlenecek yani 0 yapılacak**
5. Timer ile her saniyede 1 kesmeye gidilecek ve 3 adet led sıra ile yakılacaktır. Aynı anda sadece 1 adet led yanacak ve sürekli olarak led yanma bilgisi ve kesmeye gitme sayısı yazdılacaktır.
6. UART üzerinden timer her kesmeye gittiğinde “ LED1:ON LED2:OFF”, “LED3:OFF, sayac:1” gibi bir formatta ledlerin durumu ve sayaç bilgisi yazdırılacaktır.

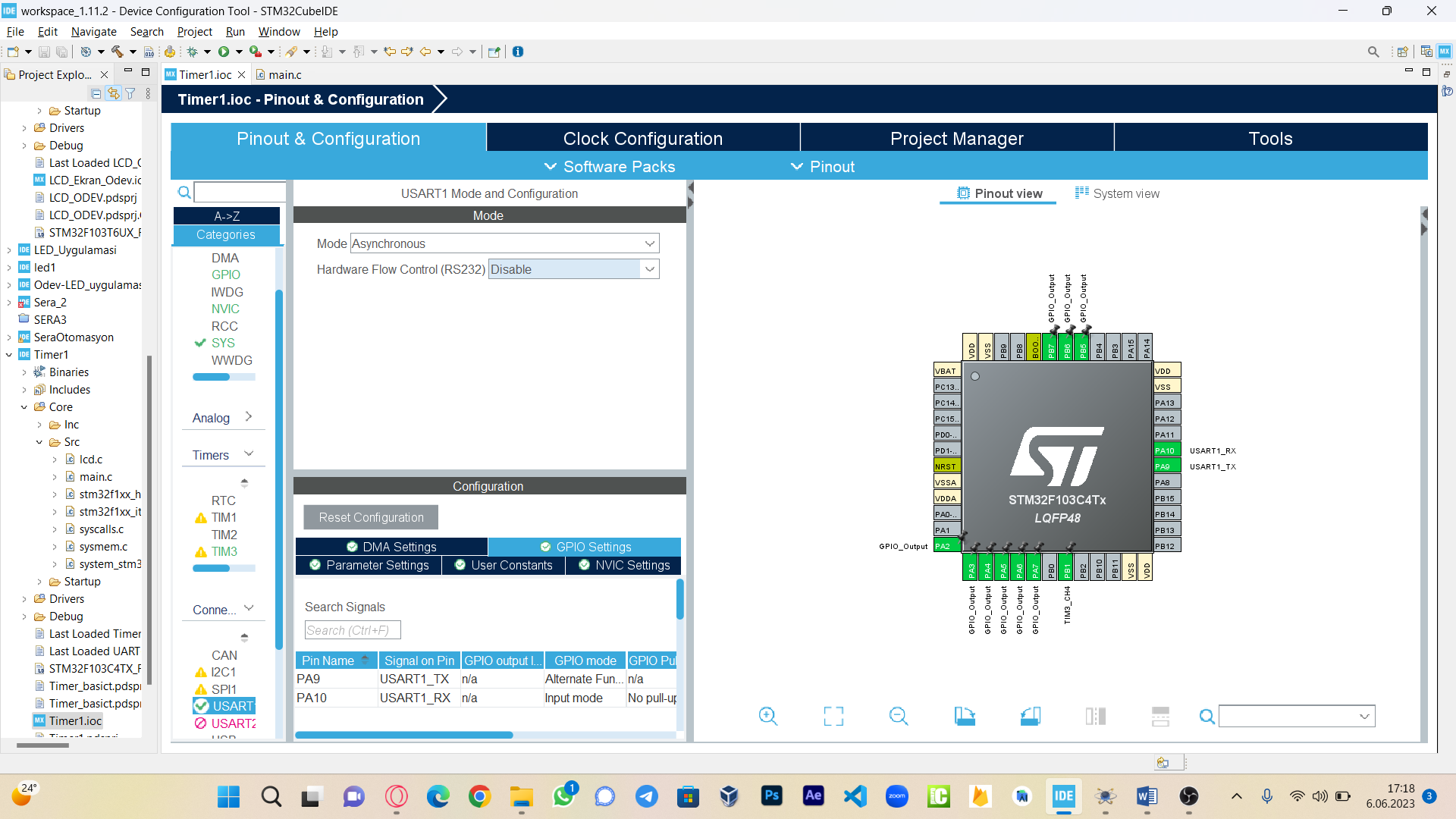
Kullanılacak bilgiler : lcd bağlantıları : d4-d7 arası : PA4-PA7 olacak, RS=PA2 ve E=PA3 olacak. Buton PA1 e bağlı olacak button PB1 portuna bağlı olacak ve bu buton ile timer 3 kullanılacak. Ledler PB5-PB6-PB7 portlarına bağlı olacak, Uart: PA9(TXD) ve PA10(RXD) a bağlı olacak.

**Proje Oluşturma Aşamaları :**

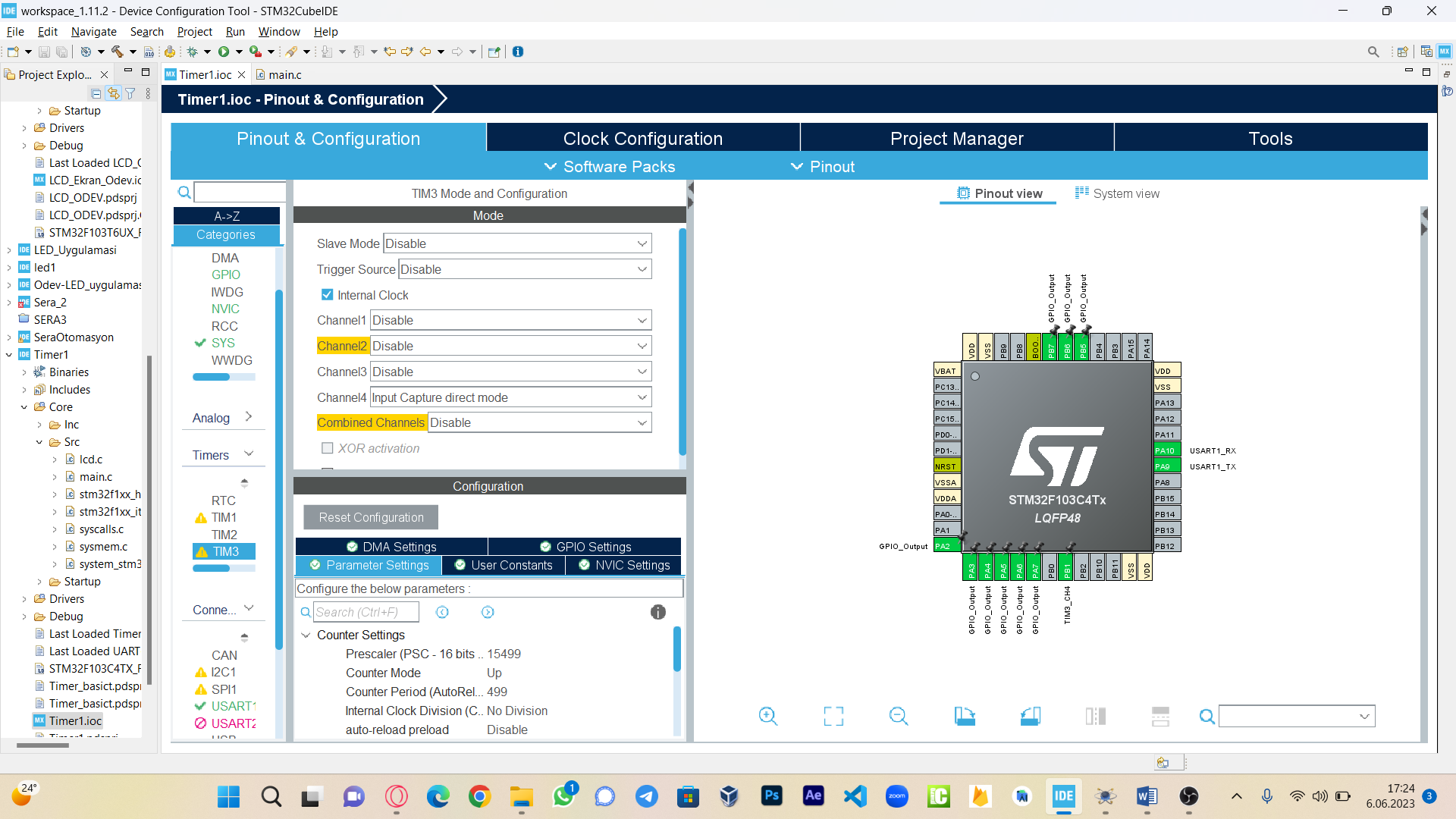
1. Pinler Ayarlandı: LCD pin bağlantıları



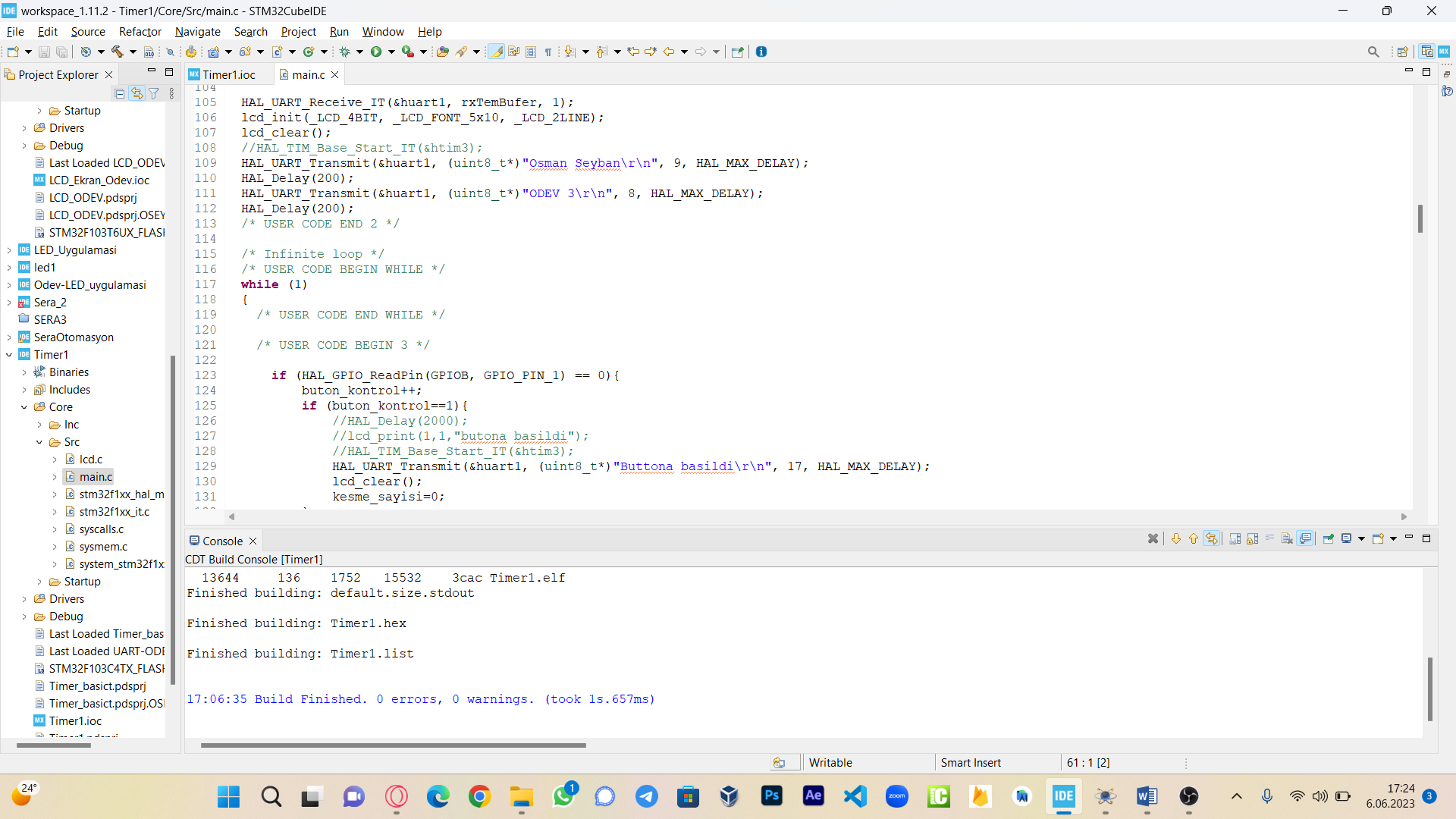
1. Usart aktifleştirildi.



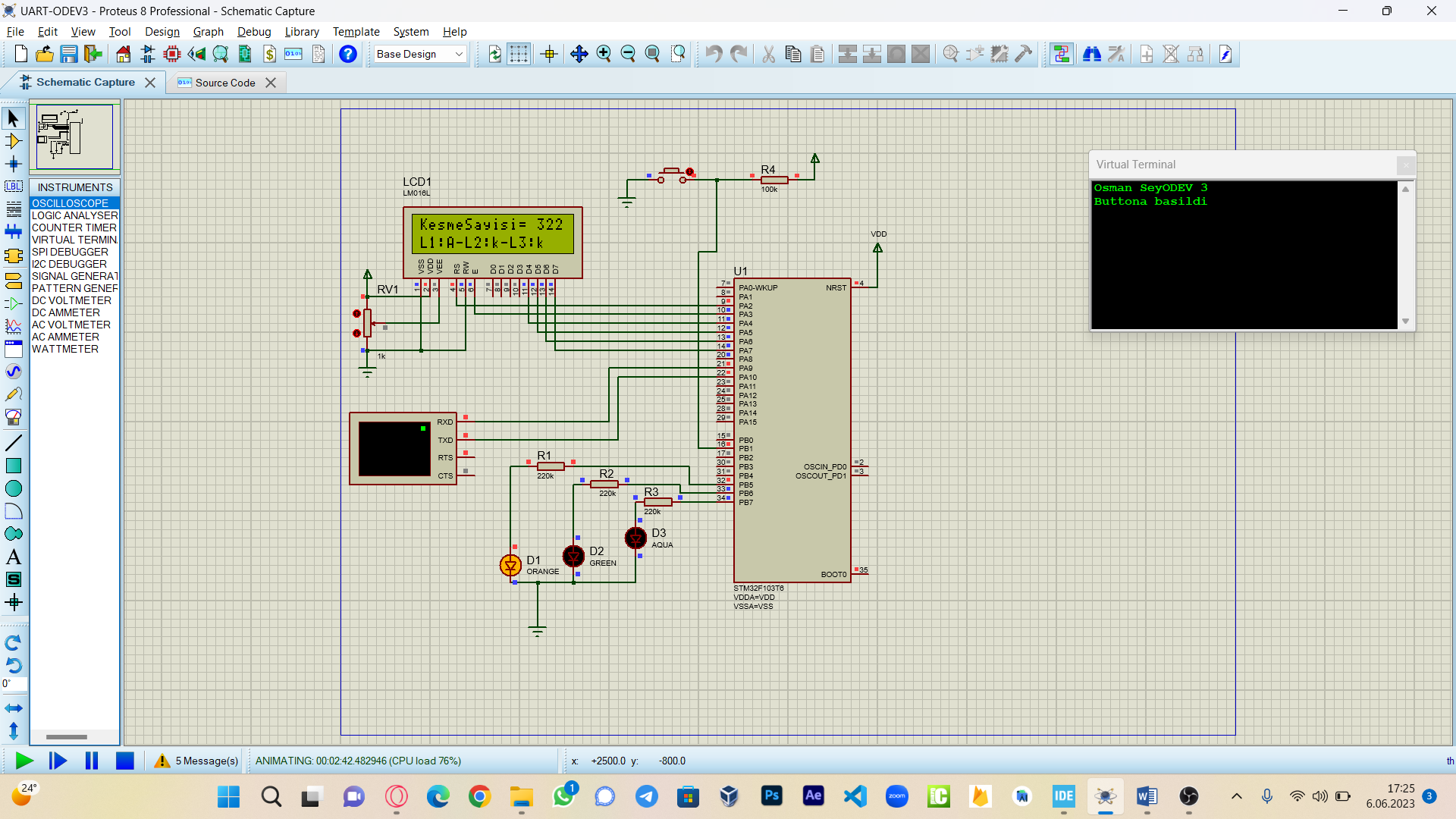
1. Timer aktifleştirildi ayarları yapıldı



1. Gerekli kodlar yazıldı



1. Proteus bağlantıları kuruldu



1. İlgili kod blokları :

**char** yazi[16]=" ";

**int** kesme\_sayisi=0;

**int** buton\_kontrol=0;

**int** **main**(**void**)MX\_GPIO\_Init();

MX\_TIM3\_Init();

MX\_USART1\_UART\_Init();

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

HAL\_UART\_Receive\_IT(&huart1, rxTemBufer, 1);

lcd\_init(\_LCD\_4BIT, \_LCD\_FONT\_5x10, \_LCD\_2LINE);

lcd\_clear();

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)"Osman Seyban\r\n", 9, HAL\_MAX\_DELAY);

HAL\_Delay(200);

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)"ODEV 3\r\n", 8, HAL\_MAX\_DELAY);

HAL\_Delay(200);

**while** (1)

{

**if** (HAL\_GPIO\_ReadPin(GPIOB, GPIO\_PIN\_1) == 0){

buton\_kontrol++;

**if** (buton\_kontrol==1){

HAL\_UART\_Transmit(&huart1, (uint8\_t\*)"Buttona basildi\r\n", 17, HAL\_MAX\_DELAY);

lcd\_clear();

kesme\_sayisi=0;

}

**else** {

lcd\_clear();

kesme\_sayisi=0;

}

}

}

}

**void** **HAL\_TIM\_PeriodElapsedCallback**(TIM\_HandleTypeDef \*htim)

{

**if** (htim == &htim3)

{

kesme\_sayisi++;

**sprintf**(yazi,"KesmeSayisi= %d",kesme\_sayisi);

lcd\_print(1,1,yazi);

**if** (kesme\_sayisi % 3 == 1)

{

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(led3\_GPIO\_Port, led3\_Pin, *GPIO\_PIN\_RESET*);

lcd\_print(2,1,"L1:A-L2:k-L3:k");

}

**else** **if** (kesme\_sayisi % 3 == 2)

{

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_SET*);

HAL\_GPIO\_WritePin(led3\_GPIO\_Port, led3\_Pin, *GPIO\_PIN\_RESET*);

lcd\_print(2,1,"L1:k-L2:A-L3:k");

}

**else** **if** (kesme\_sayisi % 3 == 0)

{

HAL\_GPIO\_WritePin(led1\_GPIO\_Port, led1\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(led2\_GPIO\_Port, led2\_Pin, *GPIO\_PIN\_RESET*);

HAL\_GPIO\_WritePin(led3\_GPIO\_Port, led3\_Pin, *GPIO\_PIN\_SET*);

lcd\_print(2,1,"L1:k-L2:k-L3:A");

}

//sprintf(yazi2,"Nem= %d",nem);

//lcd\_print(1,1,yazi1);

}

}

**void** **HAL\_UART\_RxCpltCallback**(UART\_HandleTypeDef \*huart)

{

**if** (huart == &huart1)

{

rxBufer[rxIndex++] = rxTemBufer[0];

HAL\_UART\_Receive\_IT(&huart1, rxTemBufer, 1);

}

}

**static** **void** **MX\_USART1\_UART\_Init**(**void**)

{

/\* USER CODE BEGIN USART1\_Init 0 \*/

/\* USER CODE END USART1\_Init 0 \*/

/\* USER CODE BEGIN USART1\_Init 1 \*/

/\* USER CODE END USART1\_Init 1 \*/

huart1.Instance = USART1;

huart1.Init.BaudRate = 9600;

huart1.Init.WordLength = UART\_WORDLENGTH\_8B;

huart1.Init.StopBits = UART\_STOPBITS\_1;

huart1.Init.Parity = UART\_PARITY\_NONE;

huart1.Init.Mode = UART\_MODE\_TX\_RX;

huart1.Init.HwFlowCtl = UART\_HWCONTROL\_NONE;

huart1.Init.OverSampling = UART\_OVERSAMPLING\_16;

**if** (HAL\_UART\_Init(&huart1) != *HAL\_OK*)

{

Error\_Handler();

}

/\* USER CODE BEGIN USART1\_Init 2 \*/

/\* USER CODE END USART1\_Init 2 \*/

}