



## **KARABÜK ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ**

# **STAJ DEFTERİ**

Ad Soyad : **Mohammad AL MSALMA**

Öğrenci No : **2017710215004**

Bölüm : **Elektrik-Elektronik Mühendisliği**

E-mail: **Symohamed199@gmail.com**

Telefon NO: **05350284346**

Staj Türü : **Endüstri Stajı II**

**KARABÜK-2020**

## STAJ YAPAN ÖĞRENCİNİN

Adı Soyadı:	Mohammad Al MSALMA	
Öğrenci Numarası:	2017710215004	
Bölümü:	Elektrik-Elektronik Mühendisliği	
Staj Başlangıç Tarihi:	29.06.2020	
Staj Bitiş Tarihi:	24.07.2020	
Tarih:		
Öğrencinin İmzası:		

## STAJ YAPILAN İŞ YERİNİN

İletişim Bilgileri	Firma Adı: KBU Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü Adresi: Demir Çelik Kampüsü Mühendislik Fakültesi Kat : 4 Telefon ve Faks: 370 418 72 61 E-maili: ekyaylaci@karabuk.edu.tr
İş Yerinin Özellikleri	Çalışan eleman sayısı: 29 Lisans mezunu personel sayısı: 27 Staj yapabilecek öğrenci kontenjanı: 3 Servis, yemek, v.b. sosyal hizmetleri: - Üretim türü ve kapasitesi: Eğitim ve araştırma Makine Parkı: Test sistemler – Laboratuvar çalışması Diğer Bilgiler: -
Yetkili İmza	Firma Yetkilisinin Adı Soyadı: Ersagun Kürşat YAYLACI Görevi: Dr.Öğr.üyesi Tarih: İmza Kaşe:

## STAJ KOMİSYONU

Defter Kontrolü	Kontrol Eden Öğretim Elemanın Adı Soyadı:		Sonuç
	Tarih:		<input type="checkbox"/> Kabul <input type="checkbox"/> Red
	İmza:		Onay (Bölüm Başkanı Kaşe, İmza):

**KARABÜK ÜNİVERSİTESİ, MÜHENDİSLİK FAKÜLTESİ**  
**STAJ SİCİL FİŞİ**  
**STAJ YAPAN ÖĞRENCİ**

<b>Adı Soyadı:</b>	Mohammad Al MSALMA	
<b>Öğrenci Numarası:</b>	2017710215004	
<b>Bölümü:</b>	Elektrik-Elektronik Mühendisliği	
<b>Staj Başlangıç Tarihi:</b>	29.06.2020	
<b>Staj Bitiş Tarihi:</b>	24.07.2020	
<b>Tarih:</b>		
<b>Öğrencinin İmzası:</b>		

**STAJ YAPILAN İŞ YERİ**

<b>İletişim Bilgileri</b>	<b>Firma Adı:</b> KBU Mühendislik Fakültesi Elektrik-Elektronik Mühendisliği Bölümü <b>Adresi:</b> Demir Çelik Kampüsü Mühendislik Fakültesi Kat : 4 <b>Telefon ve Faks:</b> 370 418 72 61 <b>E-maili:</b> ekyaylaci@karabuk.edu.tr		
	<b>PUAN(%)</b>	<b>DÜŞÜNCE</b>	
<b>Amirin Öğrenci Hakkında Görüşleri</b>	<b>Öğrencinin Devam Durumu:</b> <b>Verilen işi vaktinde ve tam yapma:</b> <b>Amirlerine karşı tavır ve hareketi:</b> <b>İş Arkadaşlarına karşı tutum ve davranışısı:</b> <b>Öğrencinin Teorik Bilgisi:</b> <b>Öğrencinin Pratik Bilgisi:</b> <b>Öğrencinin Takım Çalışması Kabiliyeti:</b> <b>Öğrenci ile mezuniyeti sonrası çalışmayı düşünürmüştünüz:</b> <b>Ek Bilgiler:</b>		
<b>Yetkili İmza</b>	<b>Firma Yetkilisinin Adı Soyadı:</b> Ersagun Kürşat YAYLACI <b>Görevi:</b> Dr.Öğr.üyesi <b>Tarih:</b> <b>İmza Kaşe:</b>		

Not: Bu staj sicil fizi İş yerinin öğrenci hakkında görüşlerini ifade eder. Staj fışları kapalı, ağızı mühürlü/imzalı bir zarf içerisinde öğrenci vasıtasyyla Üniversiteye staj defterleri ile birlikte teslim edilecektir.

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	01
	<b>29.06.2020</b>	<b>Stajın İş Planı</b>	<b>Yetkili İmza</b>	

Stajın ilk gününde Ersagün Kurşat YAYLACI ile iletişime geçildi ve stajda neler yapılacağını belirlendi.

Yapılacak şeyler:

MATLAB, Visual Studio, Arduino ve Keil uvision programları kullanarak:

1. 8051 mikrodenetleyici programlama devresi,
2. Küçük bir dc motorun kapalı çevrim PI/PID kontrolör ile analog devresi,
3. Küçük bir dc motorun kapalı çevrim PI/PID kontrolör ile digital devresi gerçekleştirilecek.

Kullanılacak programları belirledikten sonra Getintopc web sitesi, her programın ana sitesi ve farklı referanslar kullanarak

- MATLAB,
- Visual Studio,
- Arduino,
- Keil uvision,
- Altium,
- LTspice,
- Proteus,
- EAGLE,

Programları kuruldu.



*Şekil 1 - Kullanılacak programlar*

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>30.06.2020</b>	<b>Yapılan İş:</b> <b>PID Kontrolcu Araştırmaları</b>	<b>Sayfa No:</b> <b>02</b>
		<b>Yetkili İmza</b>	

Stajın ikinci gününde Dropbox'ta hesap oluşturuldu ve PID kontrolcünün çalışma mantığını daha iyi kavrayabilmek için Ersagün hocanın Dropbox'a yüklediği makalelerle birlikte birkaç internet sitesi inceletti.

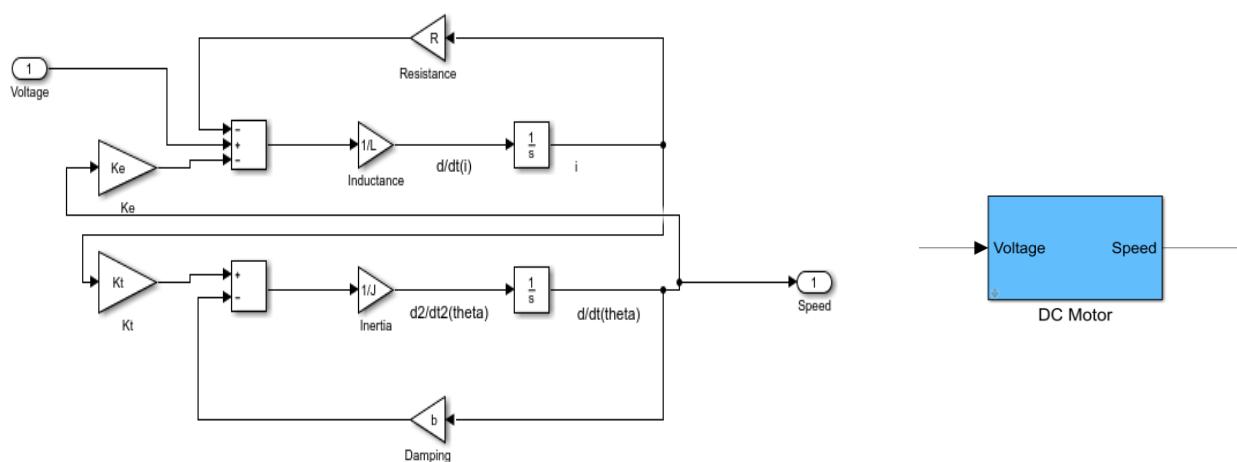
Incelemelerden sonra PID tasarımına başlayabilmek için bir motor modülü bulunması gereklidir, internette araştırdıktan sonra bir DC motorun MATLAB modeli elde edildi.

Kullanılan motorun Transfer Fonksiyonu:

$$G_m = \frac{0.72}{0.01855 s^2 + 0.0475 s + 0.5484}$$

- 1-s2.0-S2314717218300011-main
- 5\_Design
- an-cm-250\_position\_and\_speed\_control\_...
- JAETV7I102
- lab6\_closed\_loop\_velocity
- LAB8
- ME389\_OPA01\_TFSynthesis\_Guideline
- op\_amp\_pid\_paper\_MATEC\_FORMAT
- speed-control-of-dc-motor-using-analo...

*Şekil 2 - İncelenen makaleler*



*Şekil 3 - DC motorun Simulink modeli*

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	03
	<b>01.07.2020</b>	<b>PID Kontrolcu Tasarımı - 1-</b>	<b>Yetkili İmza</b>	

Bugün PID tasarımına başlandı, yükselme zamanı 50ms ve maksimum aşım %1 olacak şekilde Control Systems I dersinde görülen denklemleri kullanarak K<sub>p</sub>, K<sub>i</sub> ve K<sub>d</sub> değerleri bulup TF yazıldı.

$$k_i \approx \frac{1}{22} \text{ olsun.}$$

$$\zeta = \sqrt{\frac{\ln^2(\% Mp)}{\pi^2 + \ln^2(\% Mp)}} = 0.8261$$

$$\omega_n = \frac{5}{\zeta t_s} = 121.053$$

$$s_1 = -\zeta \omega_n + j\omega_n \sqrt{1 - \zeta^2} = -100 + j68.2162$$

$$\Rightarrow |s_1| = 121.01 , \quad \beta = 145.7^\circ$$

$$|G_m(s_1)| = 0.00269 , \quad \varphi = 69.1977^\circ$$

$$k_p = \frac{-\sin(\varphi + \beta)}{|G_m(s_1)| \sin \beta} + \frac{2k_i \cos \beta}{|s_1|}$$

$$\Rightarrow k_p = 377.412$$

$$k_d = \frac{\sin \varphi}{|G_m(s_1)| |s_1| \sin \beta} + \frac{k_i \sin \beta}{|s_1|^2}$$

$$\Rightarrow k_d = 5.096$$

*Şekil 4 - kullanılan denklemler*

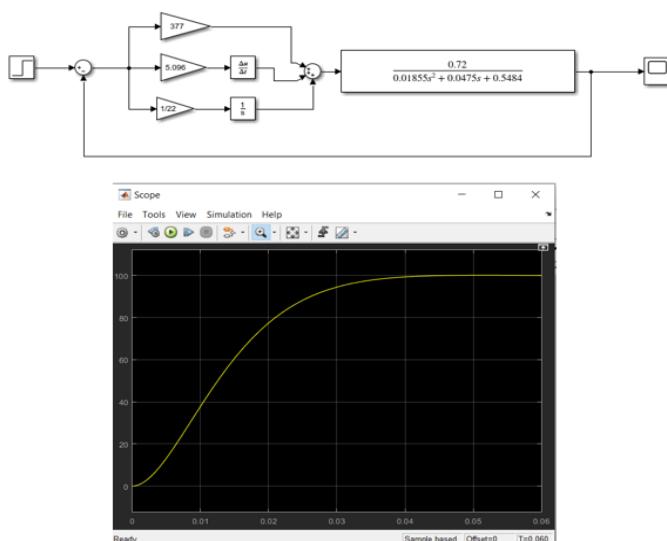
Açık çevrim transfer fonksiyonu:

$$G = \frac{411.4 s^2 + 5978 s + 0.72}{0.4081 s^3 + 1.045 s^2 + 12.0648 s}$$

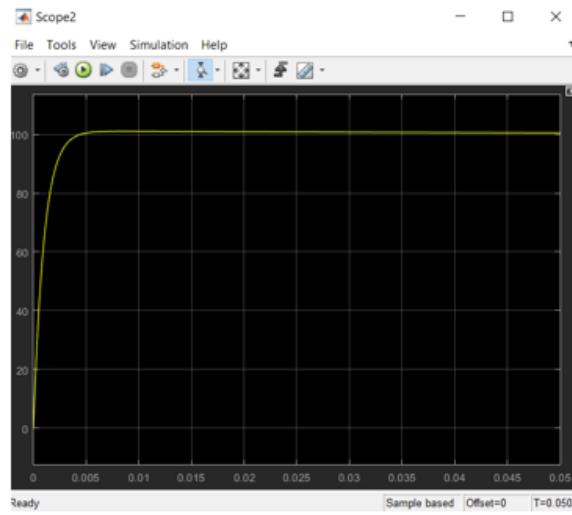
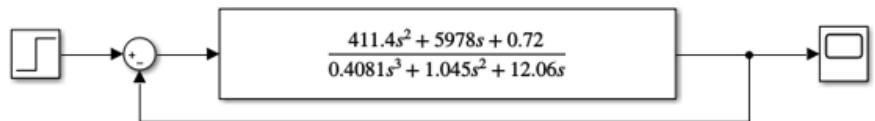
Kapalı çevrim transfer fonksiyonu:

$$M = \frac{G}{1+G} = \frac{411.4 s^2 + 5978 s + 0.72}{0.4081 s^3 + 412.445 s^2 + 5990.0648 s + 0.72}$$

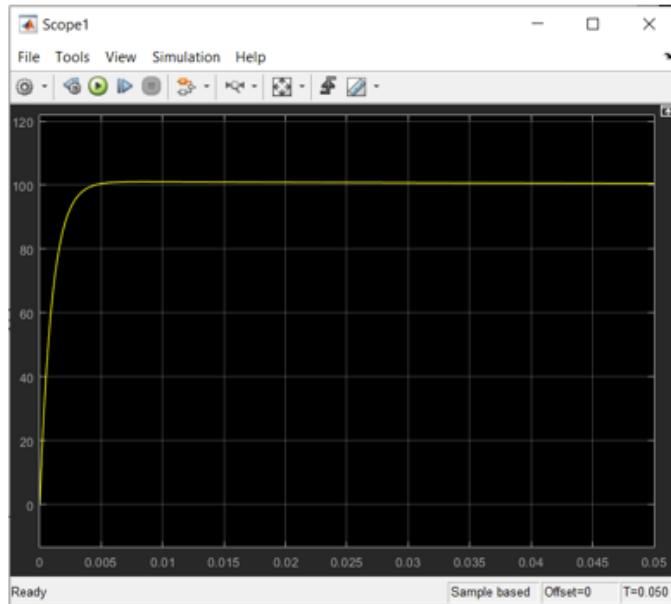
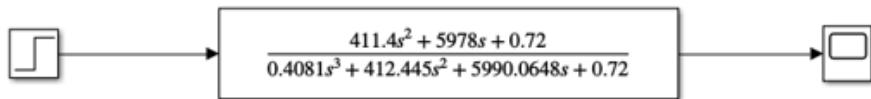
TF elde ettikten sonra MATLAB'da simülasyon gerçekleştirildi ve elde edilen sonuçlar incelendi.



*Şekil 5 - Simulink TF simülasyonu*



Sekil 6 - KÇTF simülasyonu



Sekil 7 - KÇTF simülasyonu

Bugün devreyi tasarlamağa başlandı. Tasarlanan devre Opamplardan oluşacak ve K<sub>p</sub>, K<sub>i</sub> ve K<sub>d</sub> parametrelerine göre integrator ve türev alıcı devreleri tasarlandı.

$$K_d = RC \approx 5 = 5K * 1000\mu F$$

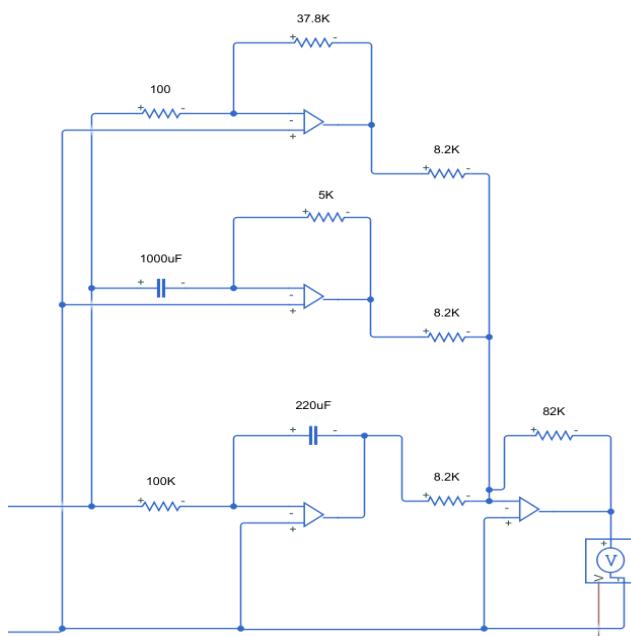
$$K_p = R_2 / R_1 \approx 378 = 37.8K / 100$$

$$K_i = 1/RC \approx 1 / 22 = 1 / (220\mu F * 100K)$$

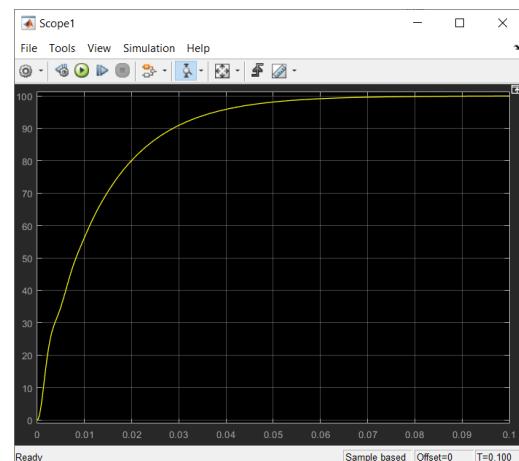
Elde edilen sonuç TF ve KÇTF simülasyonlarının sonuçları ile karşılaştırıldı. En son bütün gelişmeleri ile ilgili bir rapor hazırlanıp diğer dosyalar ile birlikte Dropbox'a yüklandı.

Incelemelerin sonucu:

Kurulan devrede sürekli hal hatası yaklaşık 0.02, yerleşme zamanı 50ms ye çok yakın yaklaşık 56ms, aşım ise hiç rastlanmadı. Elde edilen sonuçlar belirlenen kriterlere çok yakındır; ancak elde edilen sinyal, açık ve kapalı çevrim TF simülasyonlarından elde edilen sinyallerden farklıdır, bunun sebebi ise hala araştırılmaktadır.



Şekil 8 - Tasarlanan devre

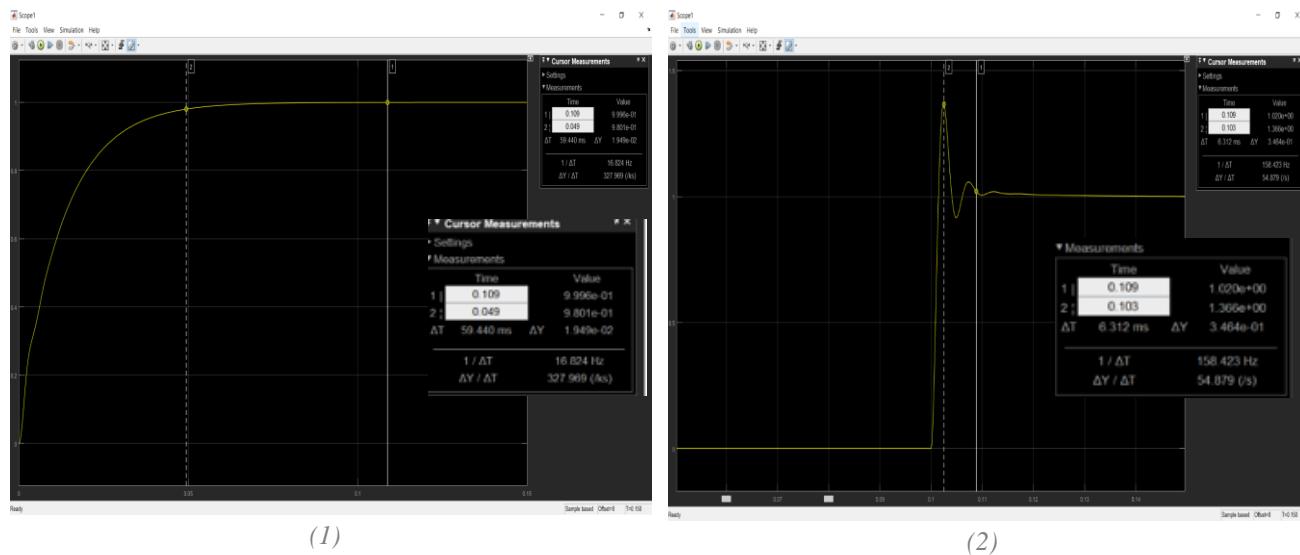


Şekil 9 - Elde edilen sonuç

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>03.07.2020</b>	<b>Yapılan İş:</b> <b>PID Kontrolcu Tasarımı -3-</b>	<b>Sayfa No:</b> <b>Yetkili İmza</b>	05
-------------------	--	---	---	----

Devreyi gerçekle kurmadan önce testler yapıldı, testlerin sonucunda bir sıkıntı tespit edildi:

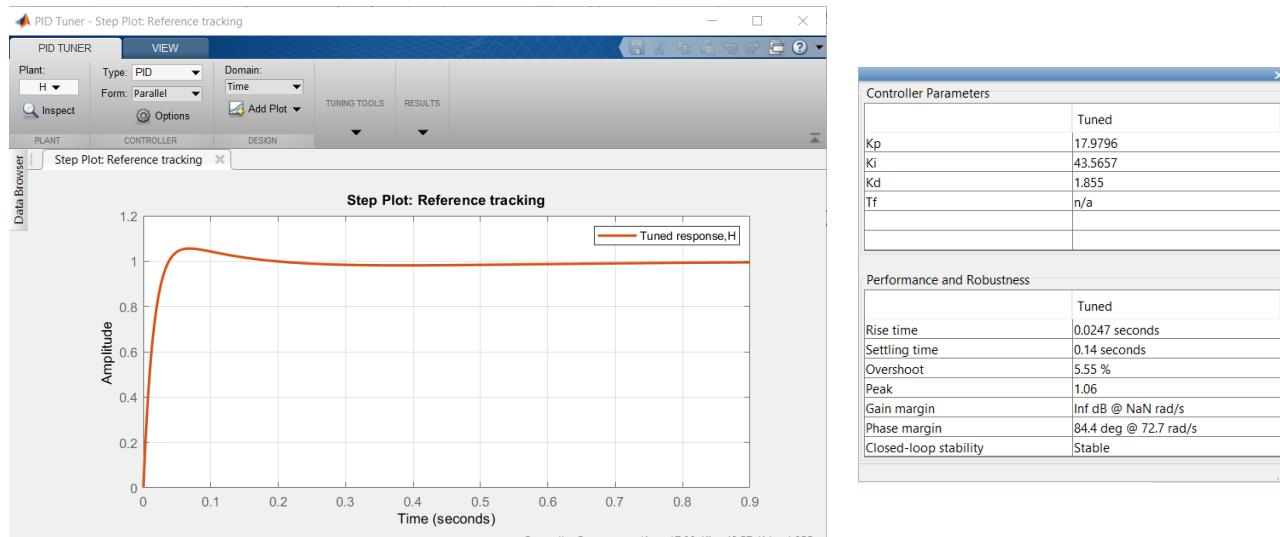
Step fonksiyonu 0.saniyede başlatıldığında “Şekil 10 – 1” daki sinyal elde edildi, burada aşım yok ve sistem yaklaşık 50ms’de yerleşiyor. Ancak step fonksiyonu 0’dan başka bir zamanda başlatıldığında (örneğin 0.1) “Şekil 10 – 2” deki sinyal elde edildi, burada ise sistem çok hızlı yaklaşık 10ms’de yerleşiyor ancak aşımı çok büyük yaklaşık %36.



Problemin sebebi araştırıldı ve sonuçta sistemin parametrelerini değiştirdip yeni bir sistem tasarlamaya karar verildi.

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>06.07.2020</b>	<b>Yapılan İş:</b> <b>PID Kontrolcu Tasarımı -4-</b>	<b>Sayfa No:</b> <b>Yetkili İmza</b>	06
-------------------	--	---	---	----

Bugün yeni sistemi tasarlamağa başlandı. Bu sefer Kp, Ki ve Kd parametreleri bulurken denklemler kullanılmadı, onun yerine MATLAB'daki "PID Tuner" özelliği kullanıldı. Yeni sistem, aşım %5.5 ve yerleşme zamanı 140ms olacak şekilde tasarlandı. Bu özellikleri sağlayacak parametreler "PID Tuner" kullanarak bulundu. Parametreler: Kp = 18, Ki = 43.56, Kd = 1.855 .



*Şekil 11 - PID Tuner*

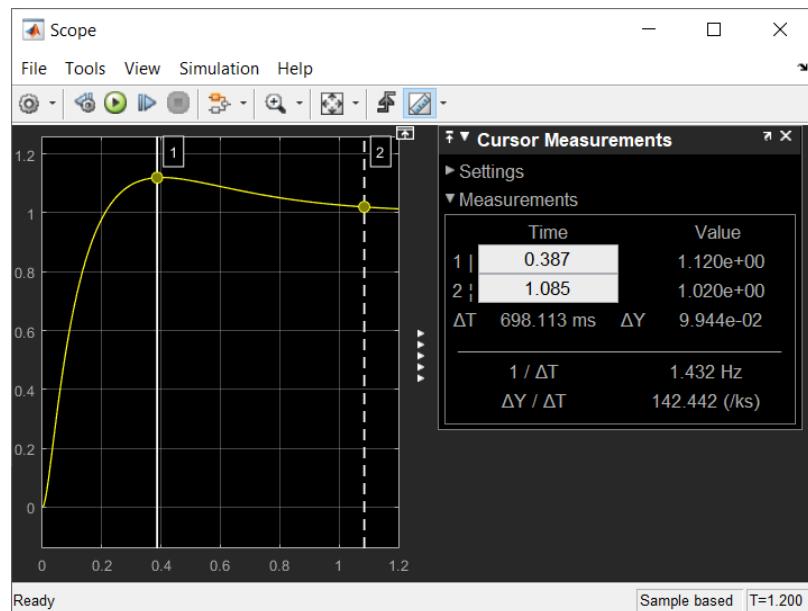
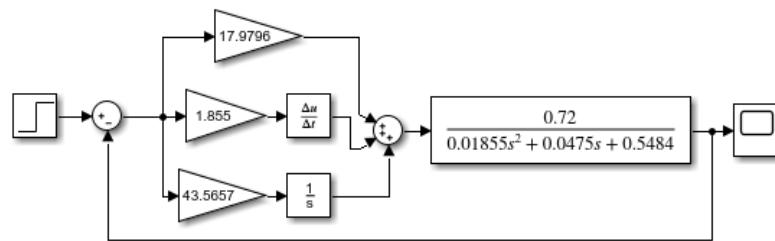
Parametreleri bulduktan sonra TF bulundu ve bir önceki sistemde karşılaşılan problem göz önünde bulundurarak Simulink, KÇTF simülasyonlarının sonuçları incelendi.

Sistemin açık çevrim transfer fonksiyonu:

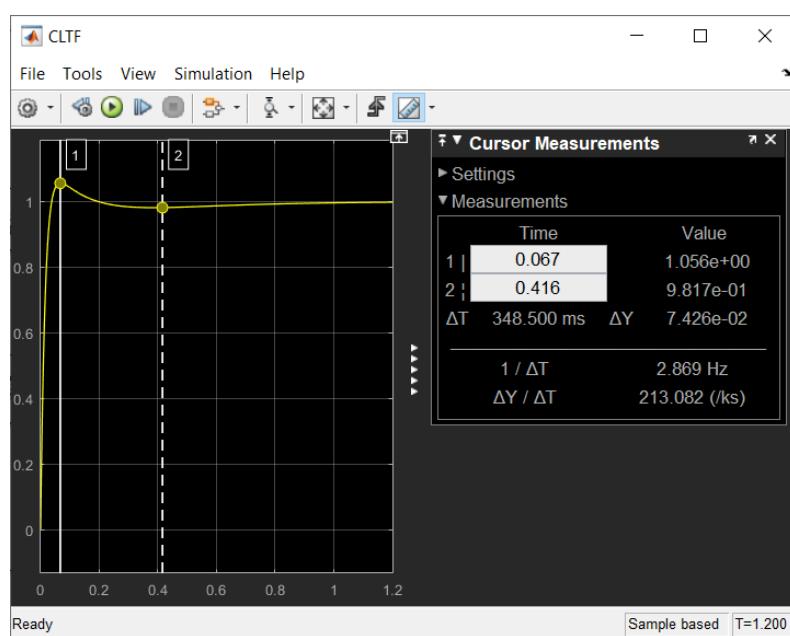
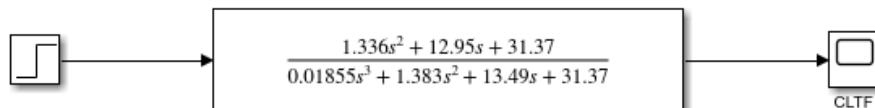
$$G = \frac{1.336 s^2 + 12.95 s + 31.37}{0.01855 s^3 + 0.0475 s^2 + 0.5484 s}$$

Sistemin kapalı çevrim transfer fonksiyonu:

$$M = \frac{G}{1+G} = \frac{1.336 s^2 + 12.81 s + 31.37}{0.01855 s^3 + 1.383 s^2 + 13.49 s + 31.37}$$



Sekil 12 - Simulink Simülasyonu



Sekil 13 - KCTF Simülasyonu

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>07.07.2020</b>	<b>Yapılan İş:</b> <b>PID Kontrolcu Tasarımı -5-</b>	<b>Sayfa No:</b> <b>07</b>
			<b>Yetkili İmza</b>

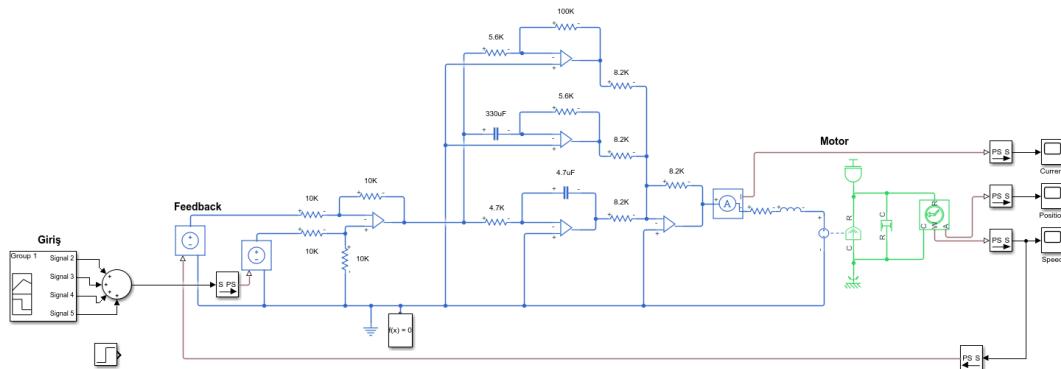
Simülasyonu gerçekleştirdikten sonra devre tasarlama geçildi. Kp, Ki ve Kd parametrelerini belirleyecek devre elemanları seçildi:

$$K_p = R_2 / R_1 \approx 18 \approx 100K / 5.6K$$

$$K_i = 1 / R * C \approx 43.5 \approx 1 / (4.7\mu F * 4,7K)$$

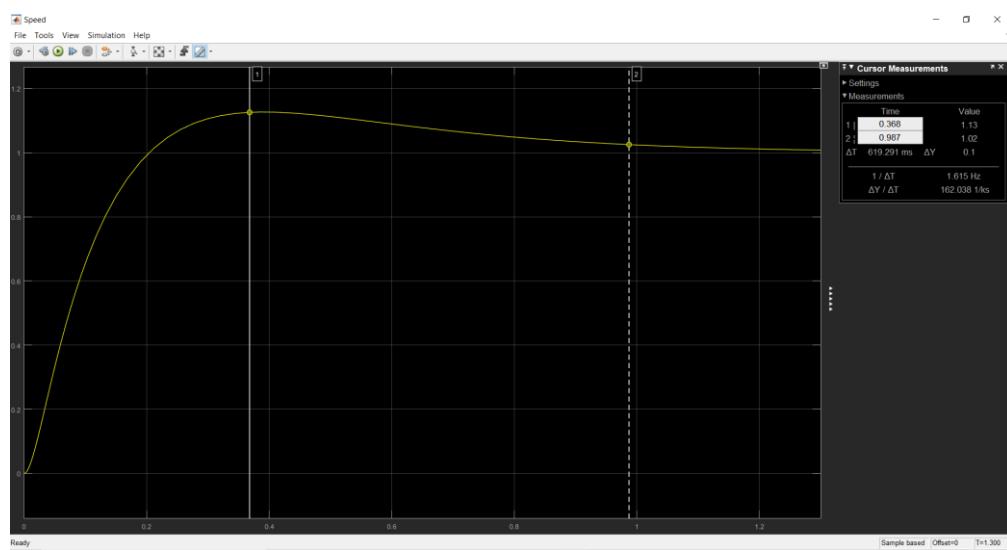
$$K_d = R * C \approx 1.86 \approx 5.6K * 330\mu F$$

Elemanları belirledikten sonra devre MATLAB'da çizildi ve elde edilen sonuçları incelemeye başlandı.

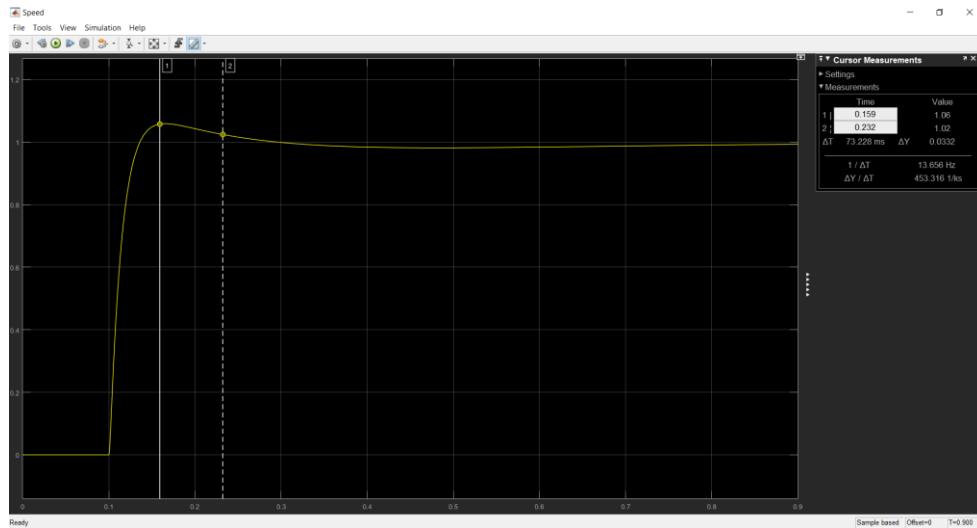


Sekil 14 - Devre şematiği

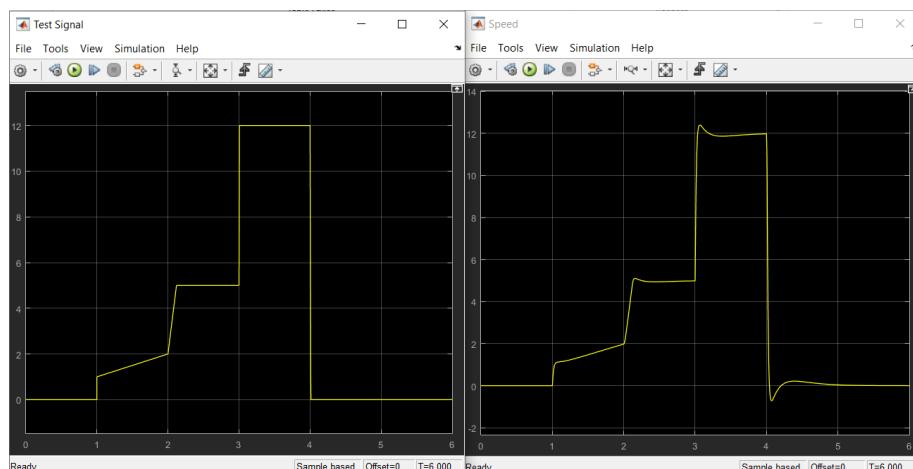
Simülasyonların sonuçları:



(1)



(2)



(3)

*Sekil 15 - Test Sinyaller ve sonuçlar*

Gözlemlerin sonucu:

Step fonksiyonu 0. saniyede başlatıldığından “Şekil 15-1” **aşım ve yerleşme zamanı** belirlenen değerden daha büyükü yaklaşık **%13** ve **1s**. Ancak step fonksiyonu 0,1. saniyede “Şekil 15-2” elde edilen sonuçlar belirlenen kriterlere çok yakın **aşım yaklaşık %6** ve **yerleşme zamanı yaklaşık 230ms** dir.

Bu iki sinyalin arasındaki farkın nedeni ise henüz belli değildi. “Şekil 15-2” de elde edilen sinyal, kapalı çevrim transfer fonksiyonunun simülasyonundan elde edilen sinyale oldukça benzıyor ancak “Şekil 15-1” deki sinyal onlara hiç benzemiyor ve Simulink simülasyonundan elde edilen sinyal ise hiç birine benzemiyor. Bu farklılık ya MATLAB’da çözme yöntemlerinden ya da tasarlanan devrede veya türetilen ifadelerde hatalardan kaynaklandığını tahmin ediliyor.

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>08.07.2020</b>	<b>Yapılan İş:</b> <b>PID Kontrolcu Tasarımı -6-</b>	<b>Sayfa No:</b> <b>Yetkili İmza</b>	<b>08</b>
-----------------------	--	---	---	-----------

Bugün bulunan sonuçları gösteren bir rapor hazırlandı ve Ersaün hocayla **ZOOM** programı ile online görüşme yapıldı. Bulunan sonuçlar tartışıldı ve karşılaşılan problemin MATLAB çözme yöntemlerinden kaynaklı olduğunu Ersagün hoca tarafından tespit edildi.

Geribesleme kısmı Hall Effect sensör ile alınacağına karar verildi, onun üzerine biraz araştırıldı ve en son gerekli olan malzemelerin Robotistan web sitesinden siparişi verildi. Malzemeler gelene kadar PID tasarlama duraklatıldı ve Programlayıcı tasarlamaya geçildi.

	<u>*L298N Voltaj Regulatörlü Çift Motor Sürücü Kartı(Kırmızı PCB)</u>	1 Adet	KDV Dahil Birim Fiyat: 13.48 TL
	<u>*680 uF 16 V Elektrolit Kondansatör</u>	2 Adet	KDV Dahil Birim Fiyat: 0.65 TL
	<u>*2.2 uF 50 V Elektrolit Kondansatör</u>	2 Adet	KDV Dahil Birim Fiyat: 0.16 TL
	<u>*2200 uF 50 V Elektrolit Kondansatör</u>	1 Adet	KDV Dahil Birim Fiyat: 3.27 TL
	<u>*330 uF 25 V Elektrolit Kondansatör</u>	4 Adet	KDV Dahil Birim Fiyat: 0.41 TL
	<u>*Erkek Kablo Tipi Power Soketi - DC Barrel Jack Male</u>	3 Adet	KDV Dahil Birim Fiyat: 1.23 TL
	<u>*1 mm Makaron - 1 Metre</u>	2 Adet	KDV Dahil Birim Fiyat: 1.63 TL
	<u>*4.7 uF 16 V Kondansatör</u>	2 Adet	KDV Dahil Birim Fiyat: 0.16 TL
	<u>*4700 nF 25 V Elektrolit Kondansatör</u>	1 Adet	KDV Dahil Birim

*Şekil 16 - sipariş verilen malzemelerin bazıları*

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	09
	<b>09.07.2020</b>	<b>Programlayıcı Araştırmaları</b>	<b>Yetkili İmza</b>	

Yapılacak devre 2 tane AT89S52 mikrodenetleyici, usb girişi ve USB – UART dönüştürücüden oluşacak. İlk AT89S52 Master mikrodenetleyici görevini, diğer ise Slave mikrodenetleyici (programlamak istediğimiz mikrodenetleyici) görevini sağlayacak. Bugün programlayıcı devrenin mantığını anlayabilmek için araştırmalar yapıldı. AT89S52'nın datasheet'ine bakıldı ve programlamada yapılması gereken bütün adımlar incelendi.

PC, master mikrodenetleyici ile UART protokolu kullanarak iletişim kuracak. Slave'i programlarken seri programlama modu ve SPI iletişim protokolu kullanılacak.

## 20. Serial Programming Algorithm

To program and verify the AT89S52 in the serial programming mode, the following sequence is recommended:

1. Power-up sequence:
  - a. Apply power between VCC and GND pins.
  - b. Set RST pin to "H".

If a crystal is not connected across pins XTAL1 and XTAL2, apply a 3 MHz to 33 MHz clock to XTAL1 pin and wait for at least 10 milliseconds.

2. Enable serial programming by sending the Programming Enable serial instruction to pin MOSI/P1.5. The frequency of the shift clock supplied at pin SCK/P1.7 needs to be less than the CPU clock at XTAL1 divided by 16.
3. The Code array is programmed one byte at a time in either the Byte or Page mode. The write cycle is self-timed and typically takes less than 0.5 ms at 5V.
4. Any memory location can be verified by using the Read instruction which returns the content at the selected address at serial output MISO/P1.6.
5. At the end of a programming session, RST can be set low to commence normal device operation.

Power-off sequence (if needed):

1. Set XTAL1 to "L" (if a crystal is not used).
2. Set RST to "L".
3. Turn  $V_{CC}$  power off.

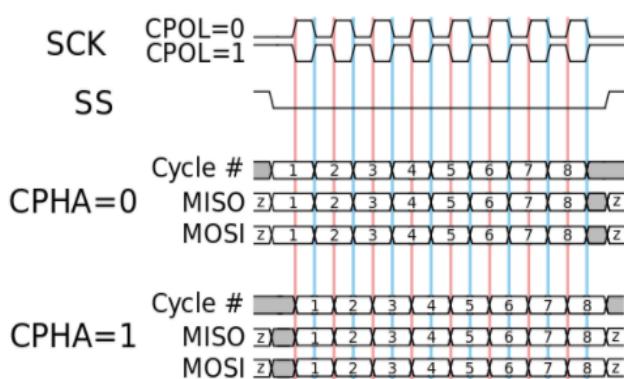
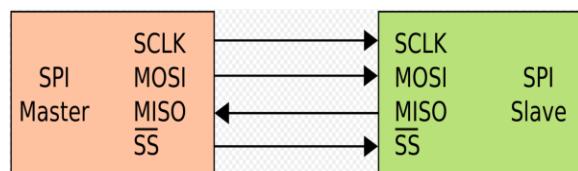
**Data Polling:** The Data Polling feature is also available in the serial mode. In this mode, during a write cycle an attempted read of the last byte written will result in the complement of the MSB of the serial output byte on MISO.

*Şekil 17 – Datasheetteki Programlama Algoritması*

Son olarak yapılan araştırmalardan faydalananarak projeyi kalan sürede bitirebilmek için bir plan kuruldu.

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	10
	<b>10.07.2020</b>	<b>İletişim Protokolların Araştırması</b>	<b>Yetkili İmza</b>	

Programlayıcıda iletişim çok önemli olduğu için bugün kullanılacak iletişim protokolleri daha detaylı bir şekilde incelendi ve onun üzerine araştırmalar yapıldı. Mantığını kavradıktan sonra C programa dilinde bu protokolların kodlanmasına bakıldı.



```

uint8_t SPI_transfer_byte(uint8_t byte_out)
{
    uint8_t byte_in = 0;
    uint8_t bit;

    for (bit = 0x80; bit; bit >>= 1) {
        /* Shift-out a bit to the MOSI Line */
        write_MOSI((byte_out & bit) ? HIGH : LOW);

        /* Delay for at Least the peer's setup time */
        delay(SPI_SCLK_LOW_TIME);

        /* Pull the clock line high */
        write_SCLK(HIGH);

        /* Shift-in a bit from the MISO Line */
        if (read_MISO() == HIGH)
            byte_in |= bit;

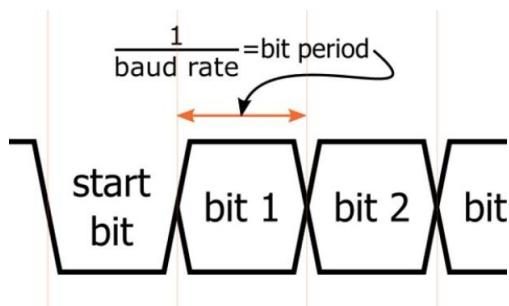
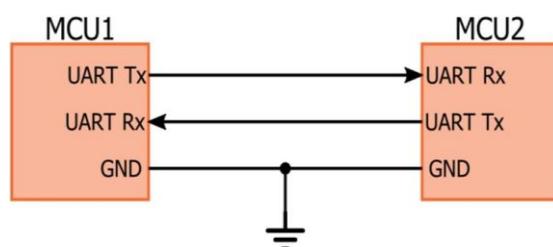
        /* Delay for at Least the peer's hold time */
        delay(SPI_SCLK_HIGH_TIME);

        /* Pull the clock line low */
        write_SCLK(LOW);
    }

    return byte_in;
}

```

Şekil 18 - SPI İletişim Protokolu



```

void UART_Init(int baudrate)
{
    SCON = 0x50; // Asynchronous mode, 8-bit data and 1-stop bit
    TMOD = 0x20; // Timer1 in Mode2.
    TH1 = 256 - (11059200UL)/(long)(32*12*baudrate); // Load timer value for baudrate generation
    TR1 = 1; // Turn ON the timer for Baud rate generation
}

void UART_TxChar(char ch)
{
    SBUF = ch; // Load the data to be transmitted
    while(TI==0); // Wait till the data is transmitted
    TI = 0; // Clear the Tx flag for next cycle.
}

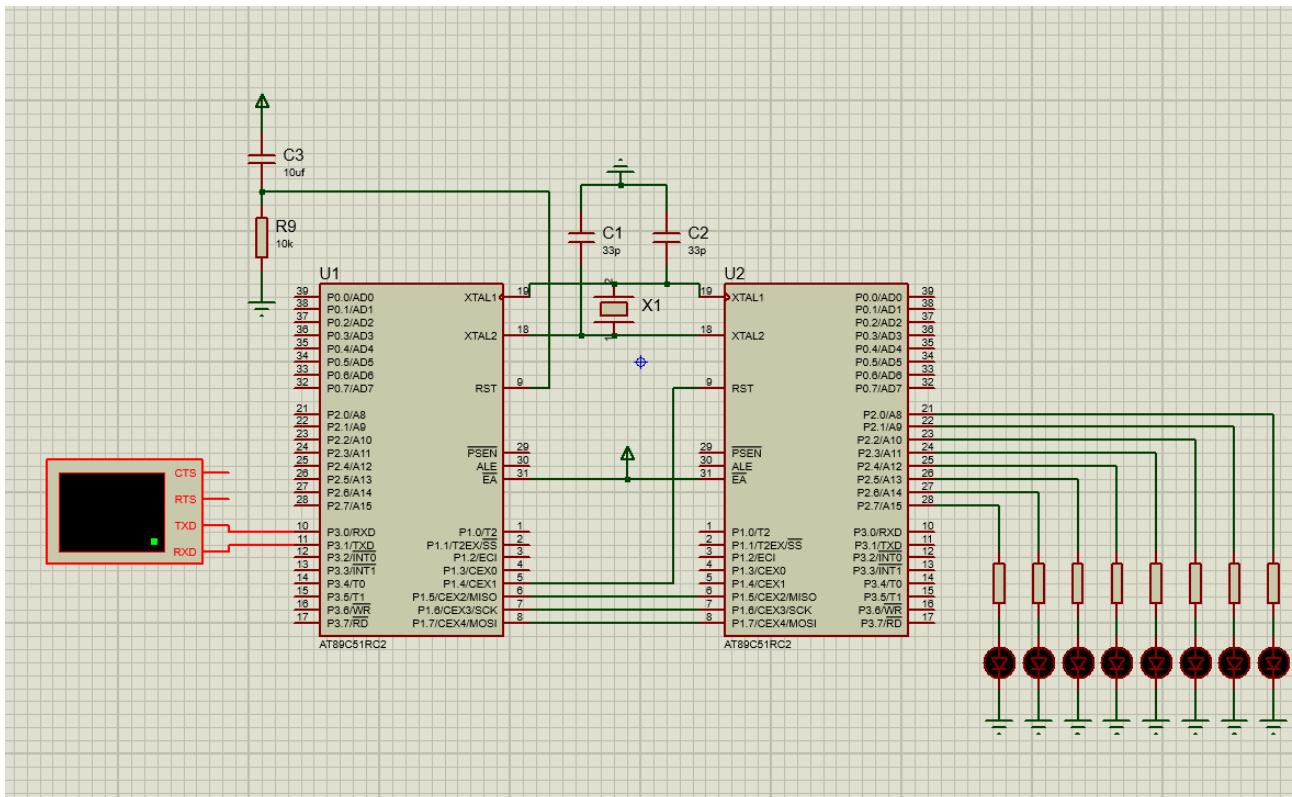
char UART_RxChar(void)
{
    while(RI==0); // Wait till the data is received
    RI=0; // Clear Receive Interrupt Flag for next cycle
    return(SBUF); // return the received char
}

```

Şekil 19 - UART İletişim Protokolu

Herhangi bir proje yapmadan önce simülasyonu yapılır ondan sonra gerçek hayatı uygulanır. Bugün programlayıcının simülasyonu gerçekleştirmeye çalışıldı. İlk olarak Proteus programı ile başlandı, ancak AT89S52'nin modeli bulunamadı. Internet araştırıldı ancak üreten firma bir simülasyon modeli üretmemiştir. Proteus'tan vazgeçip LTSpice ve Multisim programlarında denemeler yapıldı ancak aynı sonuca varıldı.

AT89S52 değil Proteus'ta modeli olan ve aynı aileden bir chip kullanarak simülasyon yapmaya çalışıldı, AT89C51RC2 kullanarak devre kuruldu ancak datasheet'i inceleyken bazı farklılıklar tespit edildi. Bu farklılıklar yüzünden simülasyon fikrinden vazgeçildi.



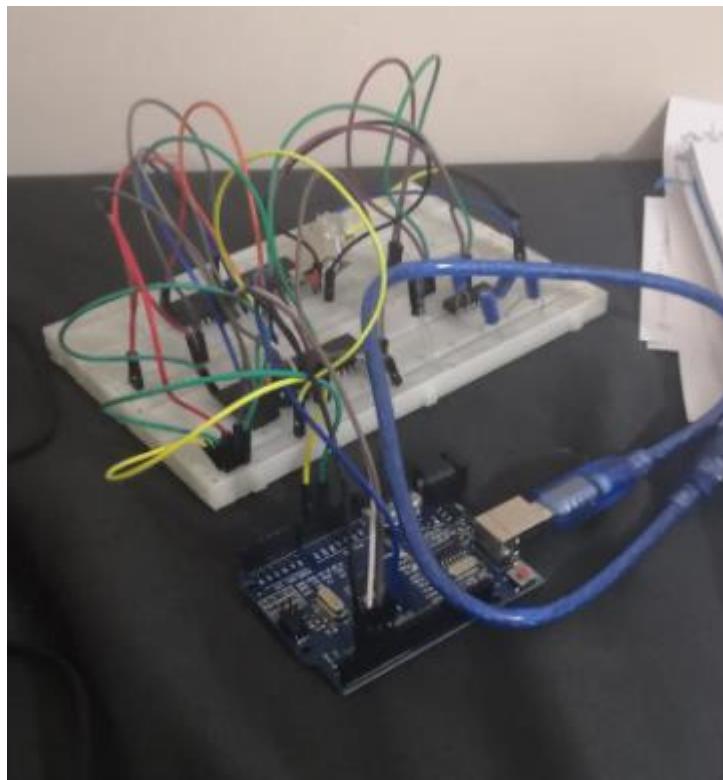
<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	12
	<b>14.07.2020</b>	<b>Programlayıcı Tasarımı -1-</b>	<b>Yetkili İmza</b>	

Simülasyon yapılamadığı için bütün testler gerçekte yapılması gereklidir, bu yüzden bugün kullanılabilecek malzemeler tespit edildi ve alındı.

Alınan malzemeler:

- 2 tane AT89S52
- 33pf kondansatörler
- 10k dirençler
- 2 tane 11.0592 Mhz kristal
- FT232 USB-UART seri çevirici modülü
- LED'ler

Malzemeleri aldıktan sonra devre Breadboard üzerinde kuruldu, böylece kodları yazmak ve test etmek aşamasına geçildi.

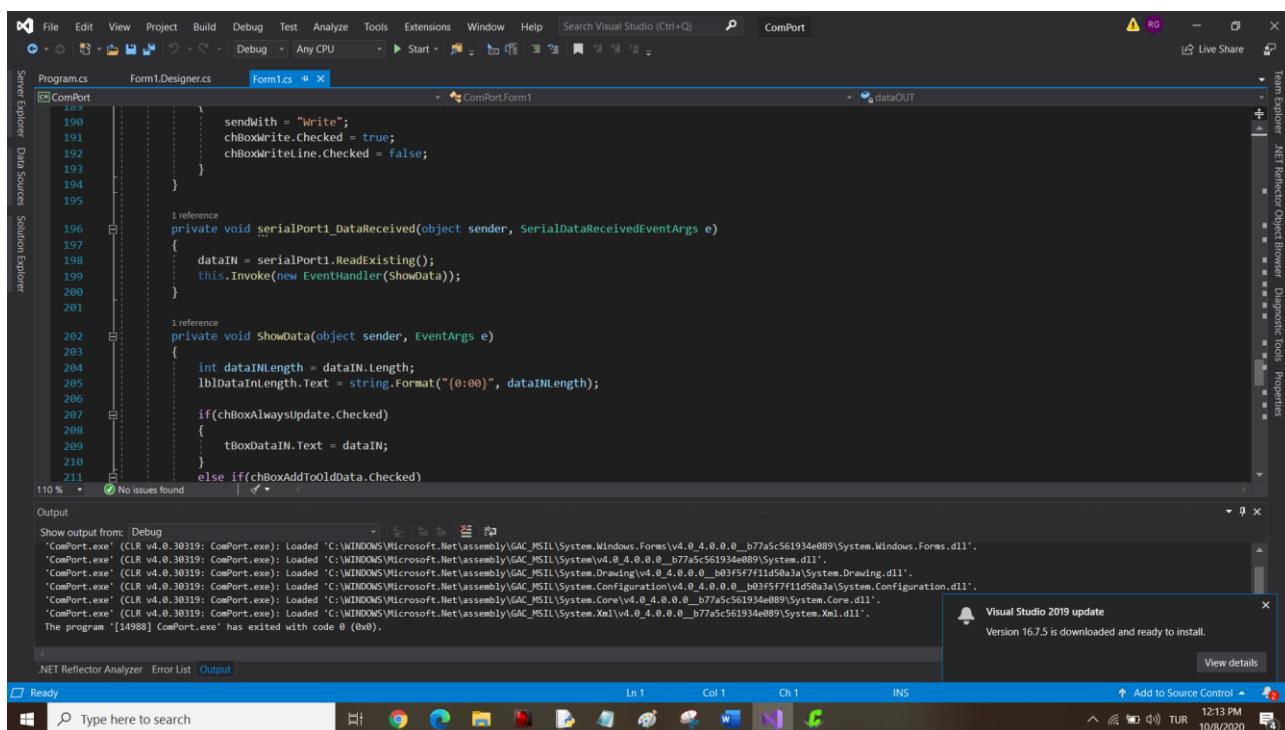
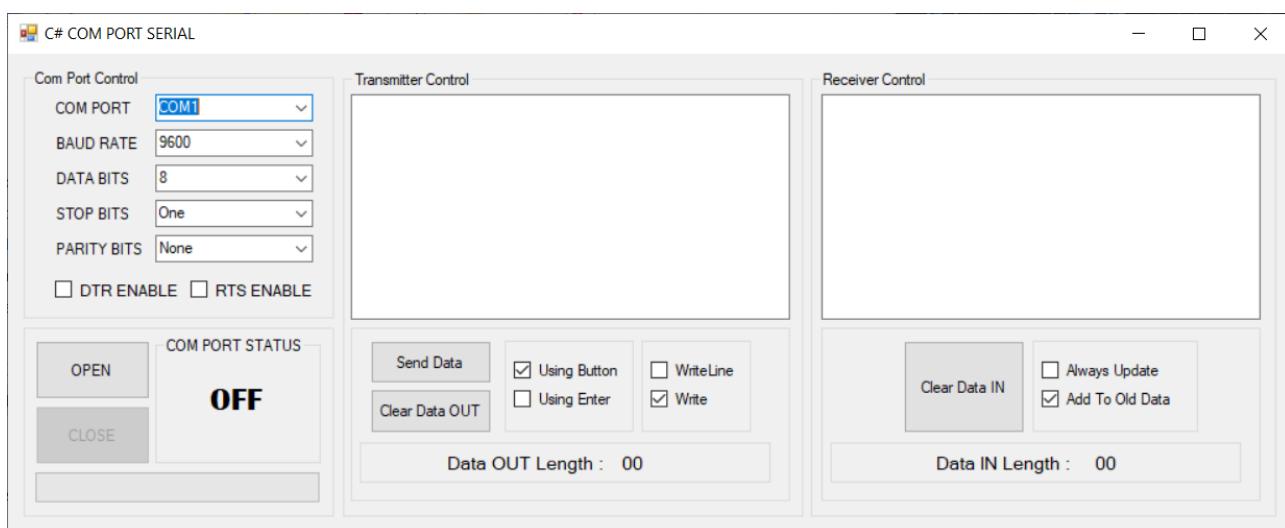


Şekil 21 - Kurulan devre

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>15.07.2020</b>	<b>Yapılan İş:</b> <b>Programlayıcı Tasarımı -2-</b>	<b>Sayfa No:</b> <b>13</b>
			<b>Yetkili İmza</b>

Devreyi kurduktan sonra iletişimi sağlayacak bir software bulunması gereklidir, bugün programlayıcının software’ı yazmaya başlandı. Yapılacak software Visual Studio programında C# programlama dili ile yazılmıştır. İlk olarak PC ile Master arasındaki iletişim sağlanmalıdır, Visual Studio’da yeni bir proje oluşturuldu, arayüz tasarlandı ve son olarak UART iletişimini sağlayacak kodlar yazıldı.

NOT: UART iletişim protokolünün kodları Master’da Arduino kullanarak yüklenmiştir.



Şekil 22 - UART iletişim software

## Master'e yüklenen kod

```
#include<reg51.h>
void uart_init();
void timer_init();
void uart_tx(unsigned char x);
unsigned char uart_rx();
void uart_msg(unsigned char *c);
void n_line();
void delay(unsigned int ms);
unsigned char tx_data;
void main()
{
    uart_init();
    uart_msg("Initializing
Serial Communication");
    n_line();
    delay(10);
}

n_line();
while(1)
{
    uart_msg("<<");
    tx_data =
    uart_rx();
    n_line();
    n_line();
    uart_msg(">>");
    uart_tx(tx_data);
    n_line();
    n_line();
}
void uart_init()
{
    SCON = 0x50;
    timer_init();
}
void timer_init()
{
    TMOD = 0x20;
    TH1 = 0xFD;
    TR1 = 1;
}

void uart_tx(unsigned char x)
{
    SBUF = x;
    while(!TI);
    TI = 0;
}

unsigned char uart_rx()
{
    unsigned char z;
    while(!RI);
    z = SBUF;
    RI = 0;
    return(z);
}

void delay(unsigned int ms)
{
    int i,j;
    for(i=0;i<=ms;i++)
        for(j=0;j<=120;j++);
}

void n_line()
{
    uart_tx(0xd);
}
```

C# diliyle çok az kod yazıldığı için arayüzü tasarlamak uzun süre almıştır, bundan dolayı planlar değişti. Herşeyi sıfırdan yazmak yerine Arduinoda kullanılan software'i kullanıp Arduino kodu modifiye edip 8051'e uygun şekilde çevirerek projeyi devam etmeye karar verildi.

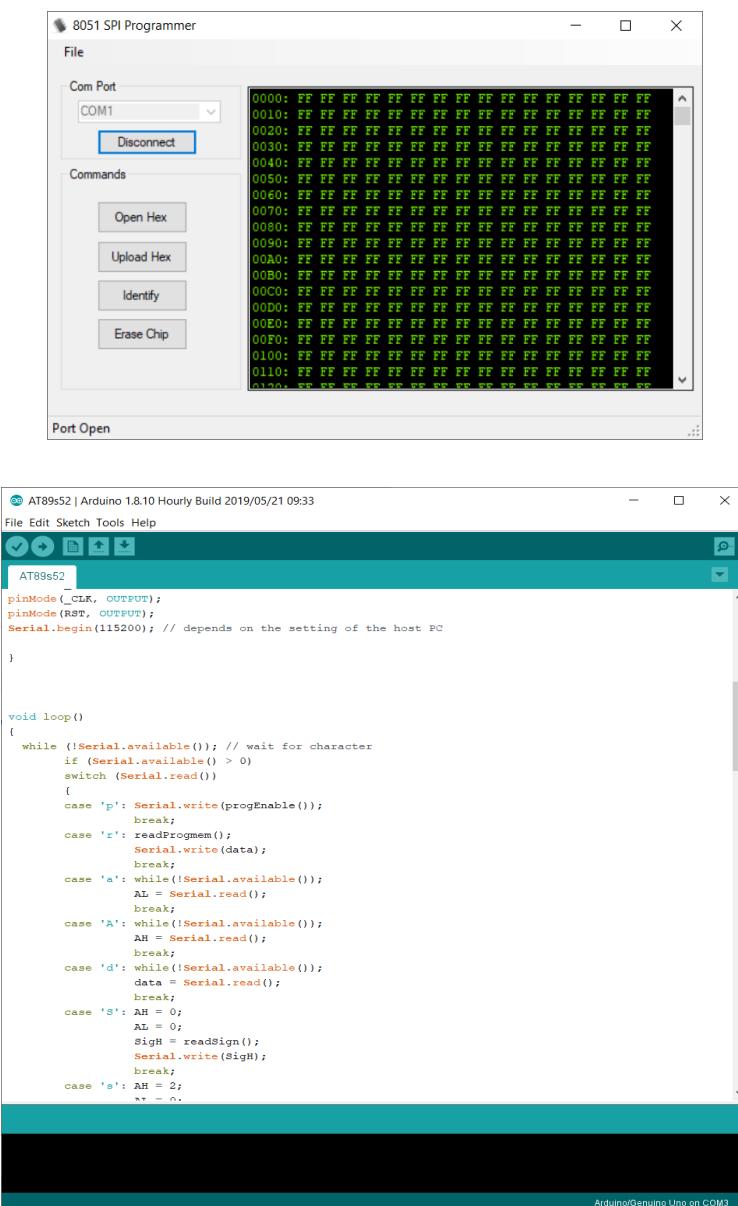
<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	14
	<b>16.07.2020</b>	<b>Programlayıcı Tasarımı -3-</b>	<b>Yetkili İmza</b>	

Arduino'yu 8051 programlayıcısı olarak çalıştmak için içine bir kod atılır, ama bu kodu AT89S52'ye atmak için birçok şeyin değiştirilmesi gereklidir. Bugün bu kod 8051'e uygun olacak şekilde çevirilmeye başlandı.

Arduino'da hazır fonksiyonlar vardır, onların yerine aynı görevi yapan fonksiyonlar yazıldı (örneğin; “Serial.Println()” fonksiyonun yerine “UART\_TxChar”). Kodu çevirdikten sonra Arduino'daki kullanılan software kullanıldı ancak proramlayıcı çalışmadı ve bilgi alışverişi sağlanamadı. Birkaç sebep tahmin edildi ve onları çözmeye çalışıldı.

Tahmin edilen problemlerlerin bazıları:

1. Arduino kodundaki “Byte, uint8\_t” datatypelerin Keil’de bulunmaması.
2. Baudrate’lerin uyuşmaması.



*Şekil 23 - Arduino kodu ve Softwar'i*

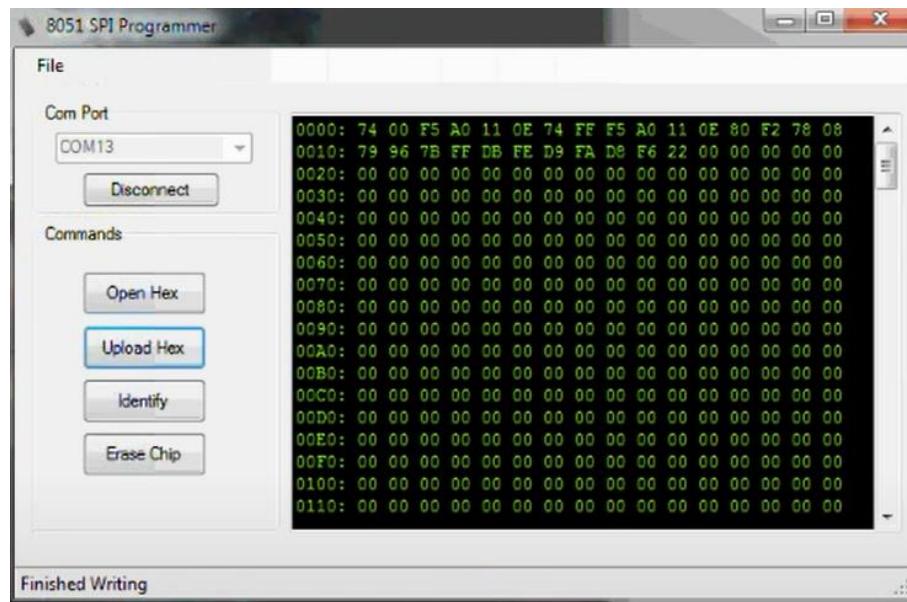
<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	15
	17.07.2020	Programlayıcı Tasarımı -4-	Yetkili İmza	

Bugün tahmin edilen 1. Problemi çözmeye başlandı. Internet araştırması yapıldı ve bu datatype'leri eklemek için bazı kütüphanelerin eklenmesi gerekti. Keil'de gerekli olan kütüphaneler yoktur, internetten indirip eklemeye çalışıldı ancak sonuç alınamadı. Keil'e bu datatype'leri ekleyemeyince başka programlar kullanmaya çalışıldı, DevC++, Code Blocks, Mplab X IDE, Code Composer ve MikroC programları kullanıldı ancak her seferinde bir sıkıntı ile karşılaşılıyordu ve problem çözülmemi.

Denemelerden sonra, tekrar biraz araştırmalar yapıldı ve bu datatype'lerin yerine “unsigned char” kullanılırsa problem kalır mı sorusu ortaya çıktı. Bu sorunun cevabını Arduino kodundaki “Byte, uint8\_t” datatype'lerin yerine “unsigned char” kullanarak bulundu. Kod normal çalıştı ve böylece 1. Problem çözüldü.

```
unsigned char SendSPI(unsigned char data)    unsigned char SendSPI(unsigned char data)
{
    uint8_t retval = 0;
    uint8_t intData = data;
    int t;

    önce
    {
        unsigned char retval = 0;
        unsigned char intData = data;
        int t;
        Sonra
}
```

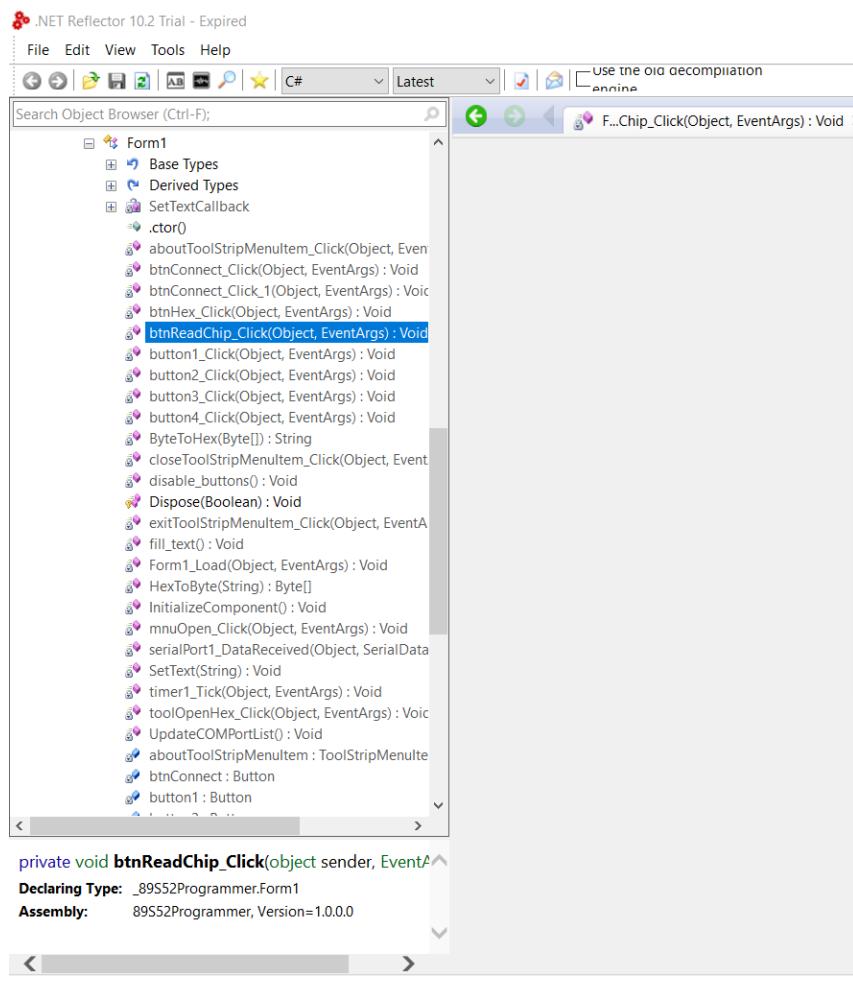


Şekil 24 - Kodun doğru çalışması

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	16
	<b>20.07.2020</b>	<b>Programlayıcı Tasarımı -5-</b>	<b>Yetkili İmza</b>	

Tahmin edilen 1. problem çözüldüğüne göre 2. probleme geçildi. Software'deki belirlenen baudrateler bilinmediği için uyum sağlamak biraz zordu, bu problemi çözmek için software'in ana kodunu incelemek gerekti ancak software'in yalnız ".exe" uzantisını internette bulunur.

.exe uzantisını normal koda çevirmek için araştırmalar yapıldı ve ".NET Reflector" programını kullanarak ters mühendislik yapıp ana kodu elde edileceği bulundu. Program kuruldu ve çalışmasıyla ilgili bir kaç tutorial izlendi ancak kodun tamamı değil sadece Software'in ana fonksiyonları elde edildi.

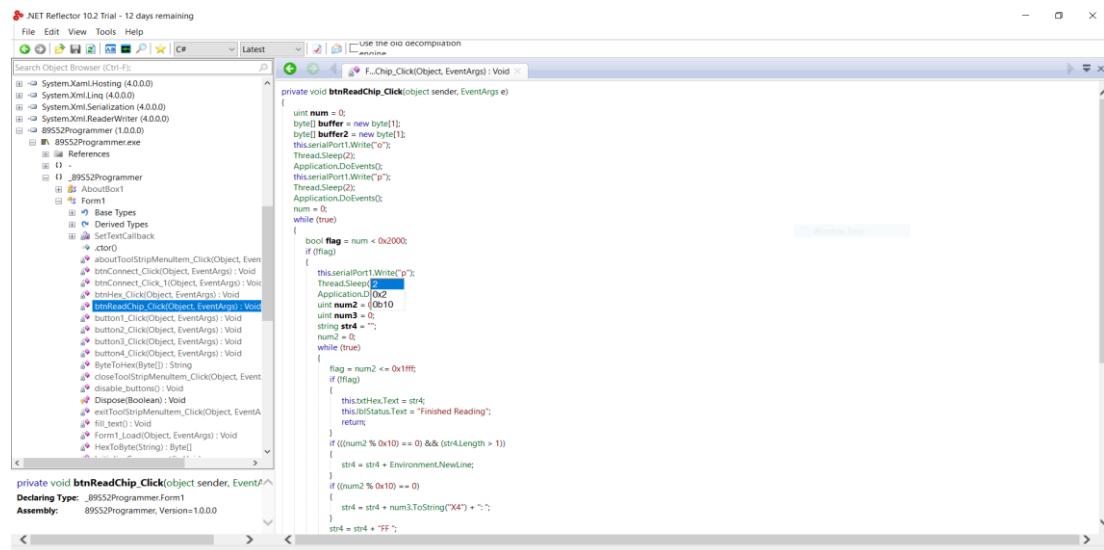


Şekil 25 - Software'in fonksiyonları

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>21.07.2020</b>	<b>Yapılan İş:</b> <b>Programlayıcı Tasarımı -6-</b>	<b>Sayfa No:</b> <b>17</b>
			<b>Yetkili İmza</b>

Ana kodu elde etme işlemine devam edildi ve daha derin araştırmalar yaparak kodu elde etmenin bir yolu bulundu. Github'dan, proramın bir uzantısını indirip programa yükleyerek kod elde edildi.

Elde edilen kodlar Visual Studio da açıldığında hata gösterip proje çalışmadı. Bu problemi gidermek için Visual Studio'da yeni bir proje açıp kodları tekrar oraya yazıp ters mühendislik işleminden yanlış ve gereksiz kodları silerek proje çalıştı, böylece ana kod modifiye etmeye ve incelenmeye hazır oldu.



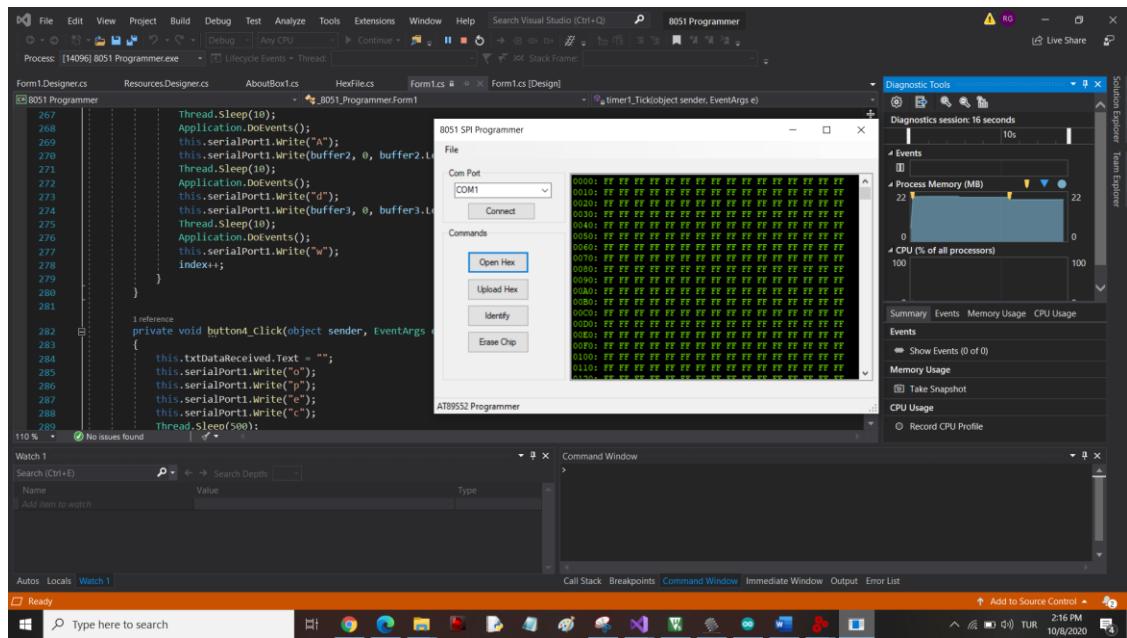
```

.NET Reflector 10.2 Trial - 12 days remaining
File Edit View Tools Help
Search Object Browser (Ctrl+F)
Latest
private void btnReadChip_Click(object sender, EventArgs e)
{
    uint num = 0;
    byte[] buffer = new byte[1];
    byte[] buffer2 = new byte[1];
    this.serialPort1.WriteLine("C");
    Thread.Sleep(2);
    Application.DoEvents();
    this.serialPort1.WriteLine("P");
    Thread.Sleep(2);
    Application.DoEvents();
    num = 0;
    while (true)
    {
        bool flag = num < 0x2000;
        if (!flag)
        {
            this.serialPort1.WriteLine("P");
            Thread.Sleep(2);
            Application.DoEvents();
            uint num2 = (0x10 * num) + 0;
            string str = "";
            string str2 = "";
            num2 = 0;
            while (true)
            {
                flag = num2 <= 0x1fff;
                if (!flag)
                {
                    this.serialPort1.WriteLine("P");
                    this.lblStatus.Text = "Finished Reading";
                    return;
                }
                if ((num2 % 0x10) == 0) && (str.Length > 1)
                {
                    str = str + Environment.NewLine;
                }
                if ((num2 % 0x10) == 0)
                {
                    str = str + num2.ToString("X4") + "\n";
                }
                str2 = str2 + "FF ";
            }
            str2 = str2 + "FF ";
        }
    }
}

private void btnReadChip_Click(object sender, EventArgs e)
{
    Declaring Type: 89552Programmer
    Assembly: 89552Programmer, Version=1.0.0.0
}

```

Şekil 26 - .NET Reflector programından elde edilen source kod

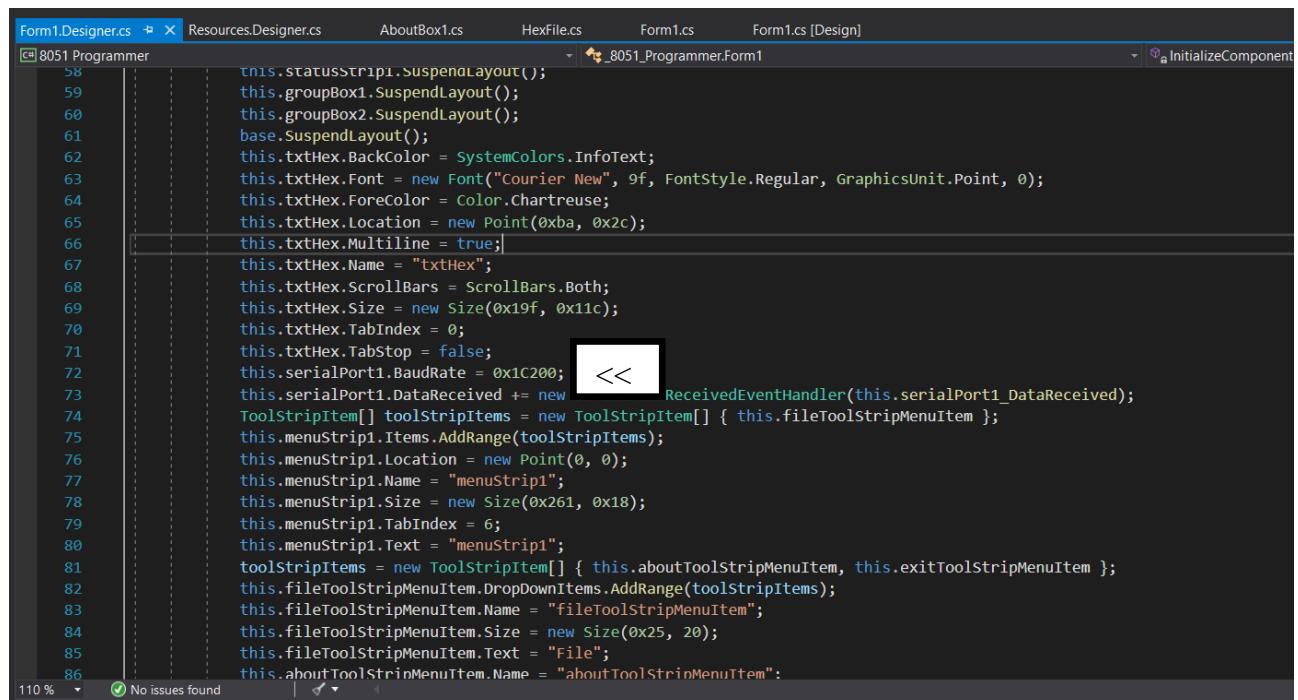


Şekil 27 - Elde edilen kodun Visual Studio'da çalışması

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b> <b>22.07.2020</b>	<b>Yapılan İş:</b> <b>Programlayıcı Tasarımı -7-</b>	<b>Sayfa No:</b> <b>Yetkili İmza</b>	18
-------------------	--	---	---	----

Ana kod elde edildiğine göre bugün incelemeler yapıldı ve kodun çalışma mantığını anlamaya çalışıldı. Kodu inceledikten sonra hangi baudrate’te çalıştığı anlaşıldı ve ona göre kodları modifiye etmeye başlandı. Software 115200 baudrate ile çalışıyor ancak yazılan Master kodda baudrate 9600 olarak belirlendi.

115200 baudrate’i, 11.0592 Mhz kristali ile Timer 1’i kullanarak üretilemediği için başka yöntemler bulabilmek için araştırmalar yapıldı. Timer 2 Baudrate üretici olarak kullanılırsa 115200 baudrate elde edilebilir, gereken ayarları yaparak istenen baudrate elde edildi. Böylece tahmin edilen 2. problem çözülmüş olur ve programlayıcı test etmeye tekrar hazır oldu.



```

Form1.Designer.cs X Resources.Designer.cs AboutBox1.cs HexFile.cs Form1.cs Form1.cs [Design]
C# 8051 Programmer _8051_Programmer.Form1 InitializeComponent
58     this.statusStrip1.SuspendLayout();
59     this.groupBox1.SuspendLayout();
60     this.groupBox2.SuspendLayout();
61     base.SuspendLayout();
62     this.txtHex.BackColor = SystemColors.InfoText;
63     this.txtHex.Font = new Font("Courier New", 9f, FontStyle.Regular, GraphicsUnit.Point, 0);
64     this.txtHex.ForeColor = Color.Chartreuse;
65     this.txtHex.Location = new Point(0x8a, 0x2c);
66     this.txtHex.Multiline = true;
67     this.txtHex.Name = "txtHex";
68     this.txtHex.ScrollBars = ScrollBars.Both;
69     this.txtHex.Size = new Size(0x19f, 0x11c);
70     this.txtHex.TabIndex = 0;
71     this.txtHex.TabStop = false;
72     this.serialPort1.BaudRate = 0x1c200; <<<
73     this.serialPort1.DataReceived += new ReceivedEventHandler(this.serialPort1_DataReceived);
74     ToolStripItem[] toolStripItems = new ToolStripItem[] { this.fileToolStripMenuItem };
75     this.menuStrip1.Items.AddRange(toolStripItems);
76     this.menuStrip1.Location = new Point(0, 0);
77     this.menuStrip1.Name = "menuStrip1";
78     this.menuStrip1.Size = new Size(0x261, 0x18);
79     this.menuStrip1.TabIndex = 6;
80     this.menuStrip1.Text = "menuStrip1";
81     toolStripItems = new ToolStripItem[] { this.aboutToolStripMenuItem, this.exitToolStripMenuItem };
82     this.fileToolStripMenuItem.DropDownItems.AddRange(toolStripItems);
83     this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
84     this.fileToolStripMenuItem.Size = new Size(0x25, 20);
85     this.fileToolStripMenuItem.Text = "File";
86     this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";

```

Şekil 28 - Software kodundaki baudrate belirten kod

```

// 115200 Baudrate üretmek için kullanılır

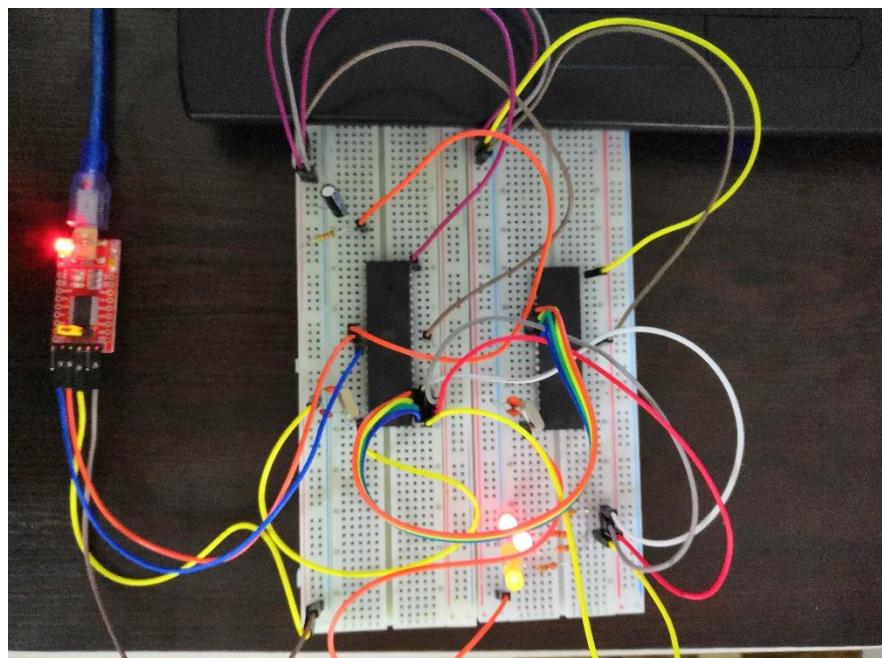
SCON = 0x50;
T2CON = 0x34;
RCAP2H = 0xFF;
RCAP2L = 0xFD;
TH2 = 0xFF;
TL2 = 0xFD;
TR2 = 1;

```

Şekil 29 - Timer 2 Baudrate üretici çalıştırmak için gerekli ayarlar

<b>İŞ YAPRAĞI</b>	<b>Staj Tarihi:</b>	<b>Yapılan İş:</b>	<b>Sayfa No:</b>	19
	23.07.2020	Programlayıcı Tasarımı -8-	<b>Yetkili İmza</b>	

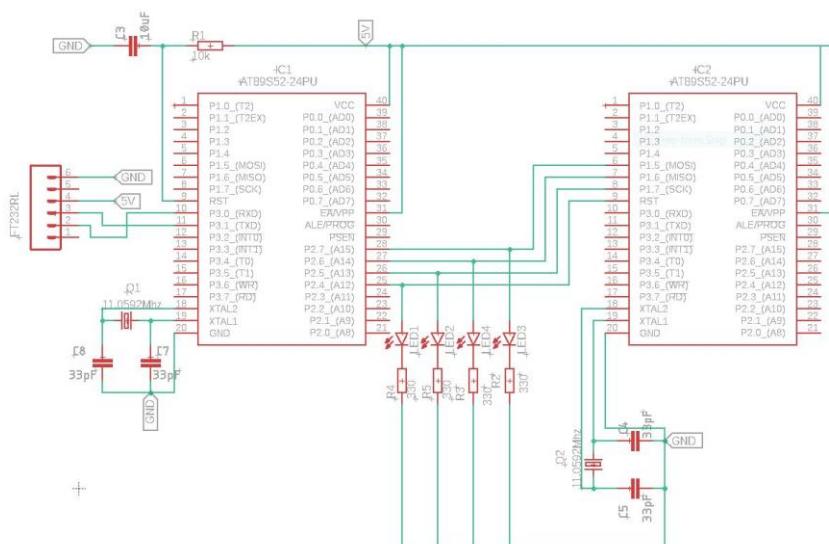
Bugün kodda değişiklikler yaparak denemeler yapıldı. Bilgi alışverişi yüksek hızla olur ve denemeler breadboard üzerinden yapıldığı için bazen bilgi alışverişinde sıkıntılar oluşuyordu. PCB tasarılayıp devre basıldığından bu sıkıntının çözülmesi beklandı ve çözülmediği halde kodlardaki gecikmeleri değiştirerek çözmeye planlandı.



Şekil 30 - Programın test seti

Bugün devrenin şematik tasarımı EAGLE programında, PCB tasarımı Altium programında gerçekleştirildi. Devreyi tasarladıkten sonra malzemeler alındı ve devre basıldı.

Bu araştırmalar ve projeleri Novruz MAMMADLI adlı arkadaşımla birlikte deneyip başarılı olarak stajımızı gayet verimli bir şekilde bitirdik.



## ❖ KAYNAKÇA:

- <https://embedronicx.com/tutorials/microcontrollers/8051/8051-uart-tutorial-serial-communication/>
- [https://exploreembedded.com/wiki/A4.8051\\_Communication\\_Protocols:UART,\\_RS232](https://exploreembedded.com/wiki/A4.8051_Communication_Protocols:UART,_RS232)
- <https://www.engineersgarage.com/tutorials/89s51-52-isp-programmer-basics/>
- <https://www.keil.com/products/c51/baudrate.asp>
- <https://www.engineersgarage.com/?s=isp+programmer>
- <https://www.youtube.com/watch?v=oIt-8XQxWKI>
- [https://www.youtube.com/results?search\\_query=communicate+pc+with+8051c%23](https://www.youtube.com/results?search_query=communicate+pc+with+8051c%23)
- <https://www.youtube.com/watch?v=RxDf6mj-vj4>
- <https://www.dropbox.com/s/i0f3jxlubpm4oxw/uart.c>
- <https://www.youtube.com/watch?v=WShhcGl3A6g>
- <https://www.youtube.com/watch?v=vHeG3Gt6STE>
- <https://educ8s.tv/c-application-arduino-communication/>
- <http://www.keil.com/support/docs/1584/>
- [https://www.youtube.com/results?search\\_query=read+hex+file+c%23](https://www.youtube.com/results?search_query=read+hex+file+c%23)
- <https://www.youtube.com/watch?v=wBVI1ym2VjE>
- <https://www.youtube.com/watch?v=aP4sgPEl7vc>
- <https://forum.arduino.cc/index.php?topic=100028.0>
- <https://www.youtube.com/watch?v=z4WZKaZoGao>
- <https://www.youtube.com/watch?v=9HivniieLvI>
- <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SimulinkModeling>