



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**OTONOM ARAÇ SİLECEK KONTROLÜ**

**Zeynep KOTAN**

**FPGA UYGULAMALARI**

**DANIŞMAN :Dr. Öğr. Üyesi Süleyman UZUN**

**BİLECİK**

**21 Mayıs 2019**



**T.C.**  
**BİLECİK ŞEYH EDEBALI ÜNİVERSİTESİ**  
**MÜHENDİSLİK FAKÜLTESİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**OTONOM ARAÇ SİLECEK KONTROLÜ**

**Zeynep KOTAN**

**FPGA UYGULAMALARI**

**DANIŞMAN :Dr. Öğr. Üyesi Süleyman UZUN**

**BİLECİK**

**21 Mayıs 2019**

## **ÖZET**

### **Ödevin Amacı**

Teknolojinin gelişmesi hayatı tamamen etkilerken aynı zamanda kolaylaştırır. Bu yeniliklerden birtanesi de araçlardaki yağmur sensörüdür. 2006 sonrası çıkan C ve D segment araçların bir çoğunda bulunan yağmur sensörleri yağmurun hızına göre silecek hızını ayarlıyor ve koldan ayar yapma zahmetini ortadan kaldırıyor.

### **Projenin Kapsamı**

Bu proje için yağmur sensörü,servo motor,lcd kullanılmıştır. Yağmur sensöründen alınan veriyle servo motor ve lcd çalıştırılmaktadır.

## **ABSTRACT**

### **Project Objective**

The development of technology makes life easier at the same time. One of these innovations is the rain sensor in vehicles. Most of the C and D segment vehicles coming after 2006, the rain sensors adjust the windscreen speed according to the speed of the rain and eliminate the trouble of adjusting the arm.

### **Scope of Project**

Rain sensor, servo motor, lcd are used for this project. The servo motor and lcd are operated with the data from the rain sensor.

## **TEŞEKKÜR**

Bu projenin başından sonuna kadar hazırlanmasında emeği bulunan ve beni bu konuya yönlendiren saygıdeğer hocam ve danışmanım Sayın Dr. Öğr. Üyesi Süleyman UZUN'na tüm katkılarından ve hiç eksiltmediği desteğinden dolayı teşekkür ederim.

**Zeynep KOTAN**

21 Mayıs 2019

# İÇİNDEKİLER

<b>ÖZET</b>	<b>ii</b>
<b>ABSTRACT</b>	<b>iii</b>
<b>TEŞEKKÜR</b>	<b>iv</b>
<b>ŞEKİL LİSTESİ</b>	<b>vii</b>
<b>1 GİRİŞ</b>	<b>1</b>
<b>2 MATERYAL VE METODLAR</b>	<b>3</b>
2.1 SG90 Servo Motor . . . . .	3
2.1.1 Servo Motor Çalışma Prensibi . . . . .	3
2.2 Yağmur Sensörü . . . . .	5
<b>3 VHDL İLE SERVO MOTOR ÇALIŞTIRMA</b>	<b>6</b>
3.1 Duty Cycle Kontrolü . . . . .	7
3.2 Servo Motor Dönüş Hızı . . . . .	7
<b>4 VHDL İLE LCD ÇALIŞTIRMA</b>	<b>8</b>
4.1 Lcd Pinleri . . . . .	8
4.2 Kodda kullanılan Komutlar için ASCI Değerleri . . . . .	8
<b>5 BLOCK DIAGRAM İLE RAIN SENSOR ÇALIŞTIRMA</b>	<b>9</b>
<b>6 SENSÖR BİRLEŞTİRME</b>	<b>9</b>
<b>7 SONUÇLAR VE ÖNERİLER</b>	<b>13</b>
<b>8 EKLER</b>	<b>14</b>
8.1 Servo Motor Kodları . . . . .	14
8.2 Lcd Kodları . . . . .	16
<b>KAYNAKLAR</b>	<b>20</b>



## ŞEKİL LİSTESİ

1	Servo Motorun Yapısı . . . . .	4
2	Puls Genişlikleri . . . . .	4
3	Servo Motor Kodu . . . . .	6
4	FPGA ve LCD Arasındaki Pin Bağlantıları . . . . .	9
5	Schematic File Kodu . . . . .	9
6	Schematic File Kodu . . . . .	10
7	Yağmur Sensörü Diyagramı . . . . .	10
8	Servo Motor Diyagramı . . . . .	11
9	Lcd Diyagramı . . . . .	11
10	Schematic File Kodu . . . . .	12
11	Pin Planner . . . . .	12



# 1 GİRİŞ

Teknolojinin gelişmesi hayatı tamamen etkilerken aynı zamanda kolaylaştırır. Bu yeniliklerden birtanesi de araçlardaki yağmur sensörüdür. 2006 sonrası çıkan C ve D segment araçların bir çoğunda bulunan yağmur sensörleri yağmurun hızına göre silecek hızını ayarlıyor ve koldan ayar yapma zahmetini ortadan kaldırıyor. Peki bu olay nasıl gerçekleşiyor.

Öncelikle bilinmesi gereken en önemli bilgilerden biri her marka ve modelin sensör mimarisinin farklı olabileceğidir. X markanın kullandığı bir sistemi y marka kullanmayabilir. Daha gelişmiş ve daha sade yada daha karmaşık sistemler ile karşımıza çıkabilir ama genel olarak belli başlı iki standarda dayanarak çalışan bu sistem aracı "vayy be ne akıllı araba" tepkileri ile hit yapabilecek özelliklerden birtanesidir.

Önceleri, araçlara her özellik eklendiğinde büyük heyecan ile televizyonlarda dergilerde yayınlanırdı. Şimdilerde bu özellikler seçilen pakete göre belirleniyor ve kataloglarda araç seçimine göre yer bulmaktadır. Yıllar önce bir markanın "yeni modelinde geri vites lambası da var " sloganı unutulmuş değildir.

Yağmur sensörü bulunan araçların dikiz aynasının monte edildiği noktada veya dikiz ayansının üzerinde infrared sensörler bulunur. Bu sensörler aracın ön camı ile öyle bir açıda yerleştirilmişlerdir ki gönderilen ışın önce yağmur tanesine sonra aracın ön camına ve ardından sensöre toplanır. Araçla yola çıkıldığında yağmur yağmıyorsa gönderilen ışınlar herhangi bir engel ile karşılaşmayacağı için ön cama yansımaz ve dikiz aynasında bulunan sensörler işlem yapmaz. Yağmur başladığında ise yağmur tanelerine çarpan kızılötesi ışınlar damlalardan sekerek ön cama yansır oradan da aynada bulunan algılayıcıya düşerler. Yağmur damlaları ne kadar sık yağarsa kızılötesi ışıkların yoğunluğu okadar artar ve aynada bulunan sensör silecek hızını okadar artırır. Hız artırma işlemi bir decoder sayesinde gerçekleştirilir. Araçlarda ayrıca sensörü off konuma getirmek için kol ve ya buton da mevcuttur.

Yağmur sensörleri için ikinci prensip ise cam sıcaklığına dayalı sistemlerdir. Yeni model araçlarda tampon bölgesinde dış sıcaklığı ölçen termometreler mevcuttur aynı zamanda

araç içerisinde bir display de sıcaklık değeri gösterilir. Önceden programlanmış bir sistem tarafından aracınızın dış sıcaklığı ile iç sıcaklığı orantılanır elde edilen değer ile cam sıcaklığı denkleştirilmeye çalışılır eğer denkleştirilemiyorsa cam üzerinde sıvı akışının yaşandığı tahmin edilir ve aradaki fark silecek hızını verir. Çok sık kullanılmamakla beraber bazı araçların ön camlarında bulunan bakır çubuklar sayesinde cam direncindeki potansiyel değişimle de yağmur tespiti yapılabilir ve silecek sistemi kontrol edilebilir.

## 2 MATERYAL VE METODLAR

Bu bölümde kullanılan materyal ve metodlardan bahsedilecektir. Projede kullanılan materyaller:

- SG90 SERVO MOTOR
- YAĞMUR SENSÖRÜ
- ALTERA

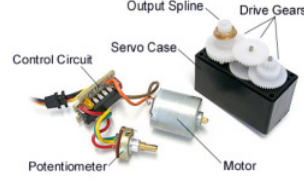
### 2.1 SG90 Servo Motor

Servo, herhangi bir mekanizmada oluşabilecek olan bir hatayı kısa sürelerde algılayan, bu hatayı denetleyen ve hatayı ortadan kaldıran otomatik cihazlardır. İçeriğinde kompanzasyon sargısı yer alan, kuvvetli bir manyetik alana sahip olan, uzun doğru akım motoruna servo motor denir. Servo motorların yapılış şekilleri DC motorlar gibidir. 1 devir- dakika ile hız bölgesinin daha altında çalışabilen ve hız kontrolünü sağlayabilen yardımcılardır teknolojik anlamda sık şekilde kullanılır. Servo motor sistemi pnömatik, hidrolik, elektronik ve mekanik şekilde kullanılabilir. Servo motor düzenekleri, sürücüler tarafından; hız ya da mekaniksel açıdan kontrol edilmektedir. Ayrıca kontrol devrelerini ve motor sürücüsünü de içinde barındırmaktadır[1].

#### 2.1.1 Servo Motor Çalışma Prensipleri

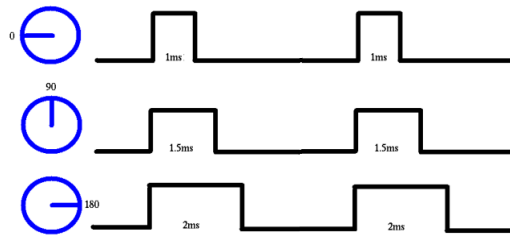
Servo motorların içerisinde motorun hareketini sağlayan bir DC motor bulunmaktadır. Bu motorun dışında bir dişli mekanizması, potansiyometre ve bir motor sürücü devresi bulunmaktadır. Potansiyometre, motor milinin dönüş miktarını ölçmektedir. Servo içerisindeki DC motor hareket ettikçe potansiyometre döner ve kontrol devresi motorun bulunduğu pozisyon ile istenilen pozisyonu karşılaştırarak motor sürme işlemi yapar. Yani, servolar diğer motorlar gibi harici bir motor sürücüye ihtiyaç duymadan çalışmaktadırlar.

Genellikle çalışma açıları 180 derece ile sınırlıdır fakat 360 derece çalışma açısına sahip özel amaçlı servo motorlar da vardır. Servolar genellikle 4.8-6V gerilim ile çalışmaktadırlar. 7.4V ve daha yüksek gerilimle çalışan servolar da bulunmaktadır.



Şekil 1: Servo Motorun Yapısı

Servo motorlar PWM (Sinyal Genişlik Modülasyonu) sinyal ile çalışmaktadırlar. Bu PWM sinyaller bir mikrokontrolciden veya uzaktan kumandadan sağlanabilmektedir. Servo, her 20 ms içerisinde bir pals değeri okumaktadır. Pals uzunluğu motorun dönüşünü belirlemektedir. Örnek olarak 1.5 ms'lik bir pals, motorun 90 derece pozisyonunu almasını sağlayacaktır (Nötr Pozisyon). Servolar hareket etmeleri için bir komut aldıklarında önce istenilen pozisyona hareket ederler, sonrasında ise o pozisyonda kalırlar. Servolar bulundukları pozisyonu korurken kendilerine dışarıdan bir güç uygulandığında bu güce direnirler. Bulundukları konumu sonsuza kadar koruyamazlar, pozisyonlarını koruyabilmeleri için palsin tekrar edilmesi gerekebilir. Hareket etmeleri için gereken pals genişliklerinin minimumları ve maksimumları vardır ve bu değerler değişkendir. Fakat genellikle minimum pals genişliği 1 ms, maksimum pals genişliği ise 2 ms'dir. Aşağıdaki şekilde genellikle sahip oldukları PWM değerleri vardır. 1 ms duty cycle değerinde  $0^\circ$  , 2 ms değerinde ise  $180^\circ$  pozisyonunu almış olur[2]. (Daha önce belirtildiği gibi bu limit bazen  $360^\circ$  de olabilmektedir.)



Şekil 2: Pals Genişlikleri

## 2.2 Yağmur Sensörü

Bu sensör yağmur ve su damlası sensörü olarak kullanılabilen bir üründür. Birbirine paralel olarak çekilmiş iletken hatların su ile teması sonucu sensör çıkış pininde analog bir değer okunabilmektedir. Arduino başta olmak üzere bir çok mikrodenetleyeci platformu ile beraber kullanılabilir. Sensör kullanımı oldukça basittir. Besleme voltajı ve toprak bağlantısı yapılarak, sensör çıkış bacağından okuma yapılabilir. Hem dijital hem analog çıkış verdiğinden dolayı, farklı sistemlere rahatlıkla uyarlanabilir. Sensör üzerindeki pot yardımı ile de sensör hassasiyeti ayarlanabilmektedir[3].

### Kullanımı:

- Vcc pinini 5V bağlayın. Gnd pinini sisteminizin toprağına bağlayın. Güç gösterge ledi yanmaya başlayacaktır. D0 gösterge ledi dijital çıkışı temsil eder. Sensör üzerine yağmur yağmaya başlayınca D0 gösterge ledi söner. Kuru iken D0 gösterge ledi yanık durumdadır. Kart üzerindeki potansiyometreyi minik bir yıldız tornavida ile çevirerek hangi yağış miktarında dijital çıkışın değişebileceğı ayarlayabilir.
- Analog çıkış pinini mikrokontrolcüye bağlayarak ufak bir program yazarak yağışın olup olmadığını ve yağış miktarı tespit edilebilir.

### Özellikler:

- Arduino başta olmak üzere bir çok mikrodenetleyeci platformu ile beraber kullanılabilir.
- Sensör kullanımı oldukça basittir.
- Besleme voltajı ve toprak bağlantısı yapılarak, sensör çıkış bacağından okuma yapılabilir.
- Hem dijital hem analog çıkış verdiğinden dolayı, farklı sistemlere rahatlıkla uyarlanabilir.
- Karşılaştırmacı modül üzerindeki potansiyometre ile hassasiyeti ayarlanabilir.

Bağlantılar:

- VCC:Güç kaynağı (3-5V),
- GND:Güç kaynağı toprak,
- DO: TTL dijital çıkış,
- AO: Analog sinyal çıkışı.

Teknik Özellikler:

- Çalışma Voltajı: 3.3V - 5V,
- Çalışma Akımı: 20mA,
- Sensör Çıkışı: Dijital ve Analog,
- Sensör hassasiyet ayarı,
- Sensör Alanı: 50 x 38 x 10mm[4].

### 3 VHDL İLE SERVO MOTOR ÇALIŞTIRMA

Vhdl kodları Quartus II 13.0sp1 (64-bit) Web Edition adlı program ile yazıldı.

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3
4 entity servo is
5     Port (
6         clk      : in  STD_LOGIC;
7         reset    : in  STD_LOGIC;
8         button_1 : in  STD_LOGIC;
9         button_r : in  STD_LOGIC;
10        pwm      : out STD_LOGIC
11    );
12 end servo;
13
14 architecture Behavioral of servo is
15     constant period:integer:=1000000;
16     constant dcycle_max:integer:=1000000;
17     constant dcycle_min:integer:=50000;
18     constant duty_in:integer:=50000;
19     signal pwm_req,pwm_next:std_logic;
20     signal duty_cycle,duty_cycle_next:integer:=0;
21     signal counter,counter_next:integer:=0;
22     signal tick:std_logic;
23 begin
24     --register
25     process(clk,reset)
26     begin
27         if reset = '1' then
28             pwm_req <= '0';
29             duty_cycle <= 0;
30             counter <= 0;
31             tick <= '0';
32         end if;
33     end process;
34 end Behavioral;
```

Şekil 3: Servo Motor Kodu

Programda, periyodu 20 ms olan bir adet PWM sinyali üretmek gerekmektedir.

Geliştirme boardunda 50 MHz'lik (20 ns) bir clk sinyali kullanıldı. 20 ms periyodunda bir sinyal üretmek için, programda 1 adet counter değişkeni tanımlandı ve Counter'ın üst sınırı (20ms/20ns) 1000 000 olarak belirlendi.

Counter üst sınırı periyod adında bir sabitle tanımlandı.

### 3.1 Duty Cycle Kontrolü

Kontrol sinyalinin Duty Cycle'ı 1ms ile 2 ms olması gerekiyor. Bundan dolayı programda dcycle\_max ile dcycle\_min adında iki sabit tanımlandı.

- dcycle\_max=Pulse genişliğinin alacağı maksimum değeri gösterecektir.  
(2ms/clock period=2 ms/ 20 ns=100000)
- dcycle\_min=Pulse genişliğinin alacağı minimum değeri gösterecektir.  
(2ms/clock period=1 ms/ 20 ns=50000)

Pulse genişliği PWM'in periyod zamanında değiştirilecektir. Bunun içinde programda tick adında bir değişken tanımlandı.

```
signal tick:std_logic;
```

```
tick<= '1' when counter= 0 else '0';
```

Pulse'nin her anahtar basımında değişimi, programında tanımlanan Duty\_in adında bir sabit kadar olacaktır. Aynı zamanda Duty\_in motorun dönüş hızını belirleyecektir.

### 3.2 Servo Motor Dönüş Hızı

Biz programımızı servo motorun bir tam devrini 5 ms'de tamamlayacak şekilde ayarladık. Programımızda

Motor hız(s) = ((dcycle\_max- dcycle\_min)\*period) / duty\_in denklemini kullanırsak,

Duty\_in= ((100000-50000)\*5 ms)/5 s= 50000 olarak hesaplarız.

## 4 VHDL İLE LCD ÇALIŞTIRMA

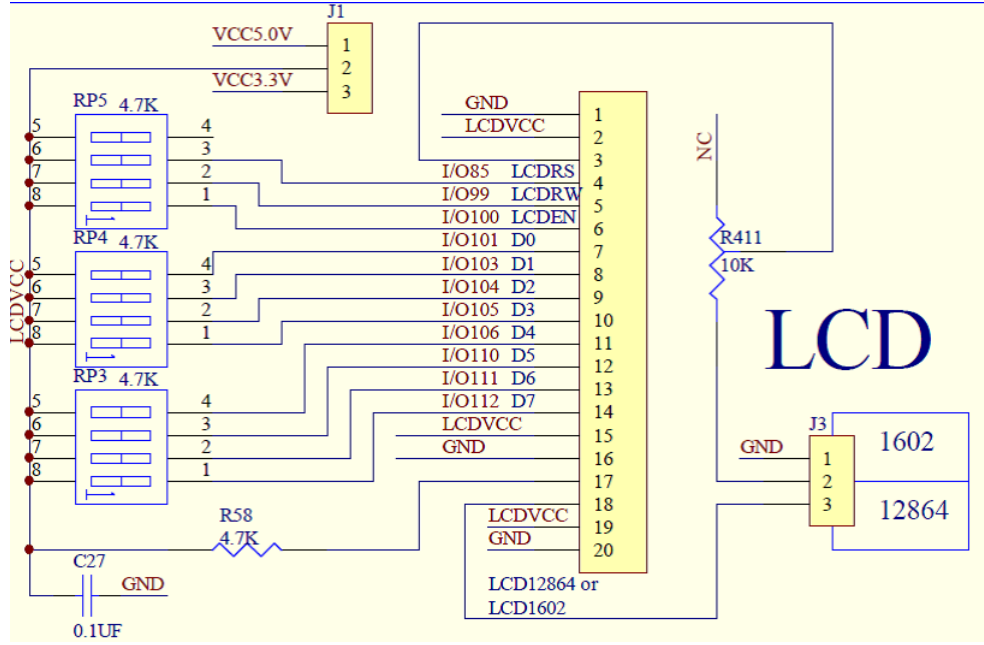
### 4.1 Lcd Pinleri

- RS(register select) pini:LCD'ye gelen bilginin komut yada veri bilgisi olduğunu tayin eder. RS '0' olursa gelen bilgi komut olarak algılanır. RS '1' olduğunda gelen bilgi veri bilgisi olduğu algılanır.
- R/W (read/write ) pini:LCD ye bilgi göndermek ve LCD den bilgi almak için kullanılır. R/W='1' ise LCD den bilgi okuma, R/W='0' ise LCD' ye bilgi gönderme işlemi yapılır.
- E(enable) pini: Yetki verme ucudur. Lcd işlemlerin gerçekleşmesi, bu ucun 1 den 0 düştüğü anda olur. (düşen kenarda tetiklenmesi)
- D0-D7 pinleri: Veri uçlarıdır.
- VDD,VSS, VEE(0) Lcd besleme uçlarıdır.

### 4.2 Kodda kullanılan Komutlar için ASCI Değerleri

- 38 = İşlev Seti: 8 bit, 2 Satır, 5x7 Nokta
- 0c = İmleç açık gösterme
- 06 = Giriş Modu
- 01 = Ekranı Temizle
- C0 = İmleç ögesini 2. satıra yerleştirin.

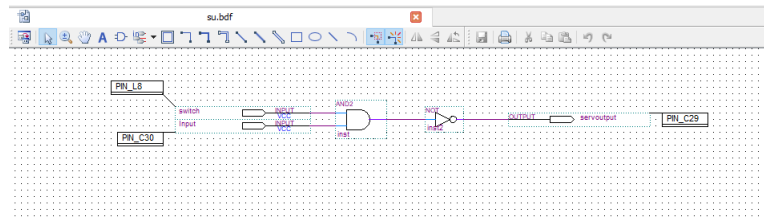




Şekil 4: FPGA ve LCD Arasındaki Pin Bağlantıları

## 5 BLOCK DIAGRAM İLE RAIN SENSOR ÇALIŞTIRMA

Switch , and kapısı ve değil kapısı kullanarak 0-1 mantığında yapıldı. Sensörden gelen veri ile Switch açık olduğunda bir üretildi. Ve elde edilen veri servo motora gönderildi.

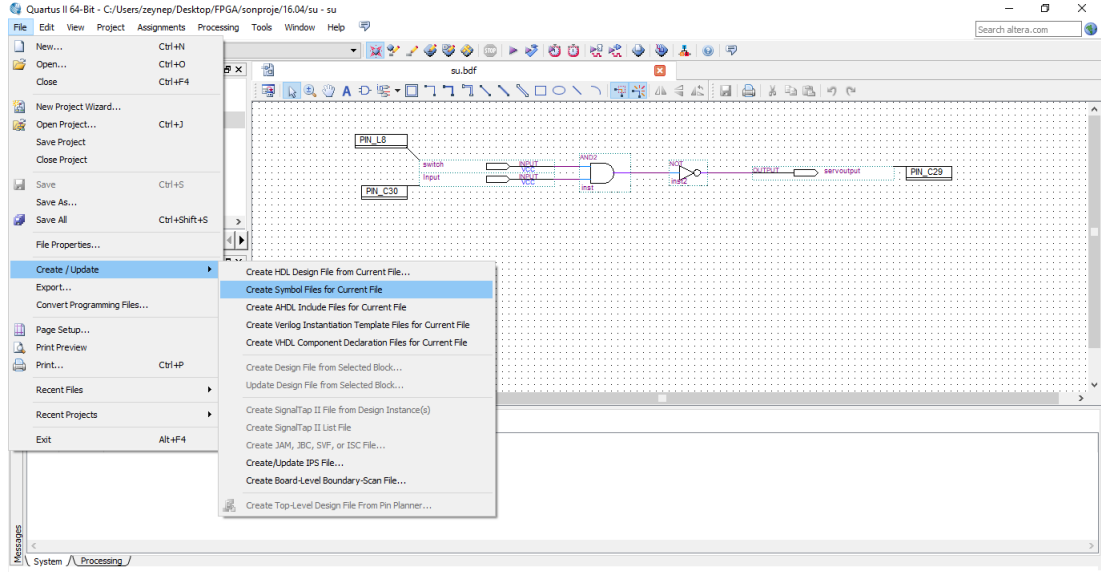


Şekil 5: Schematic File Tasarımı

## 6 SENSÖR BİRLEŞTİRME

Yağmur sensörü ile servo motor da aşağıda belirtilen şekilde diyagram oluşturulur.

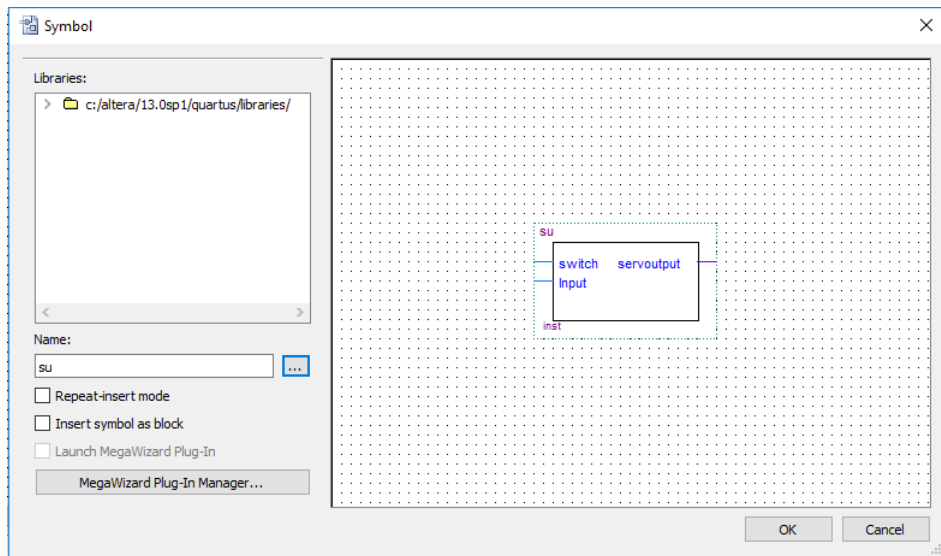
File >Create>Update>Create Symbol Files for Current File tıklanılır.



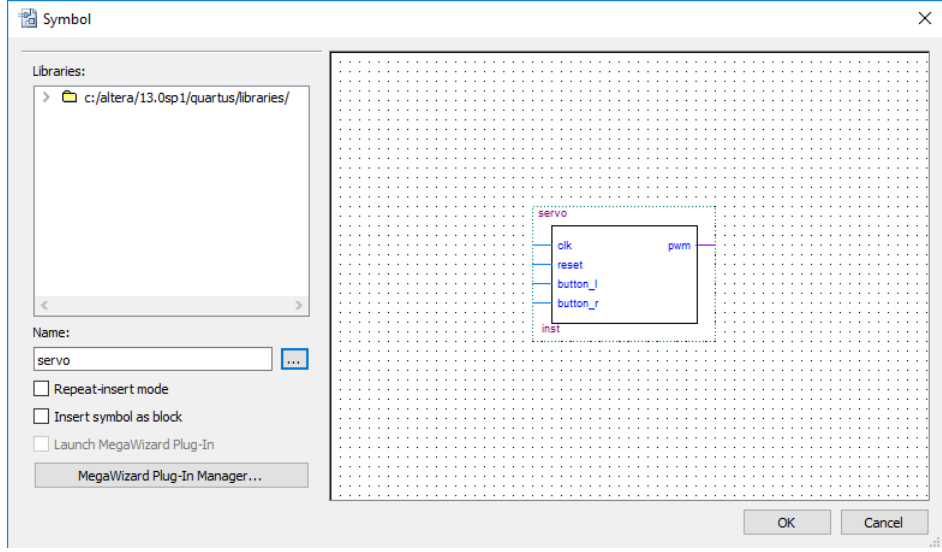
Şekil 6: Diyagram Oluşturma

Yeni proje açılır ve bu projede diğer projelerde oluşturulan dosyalar çağırılır.

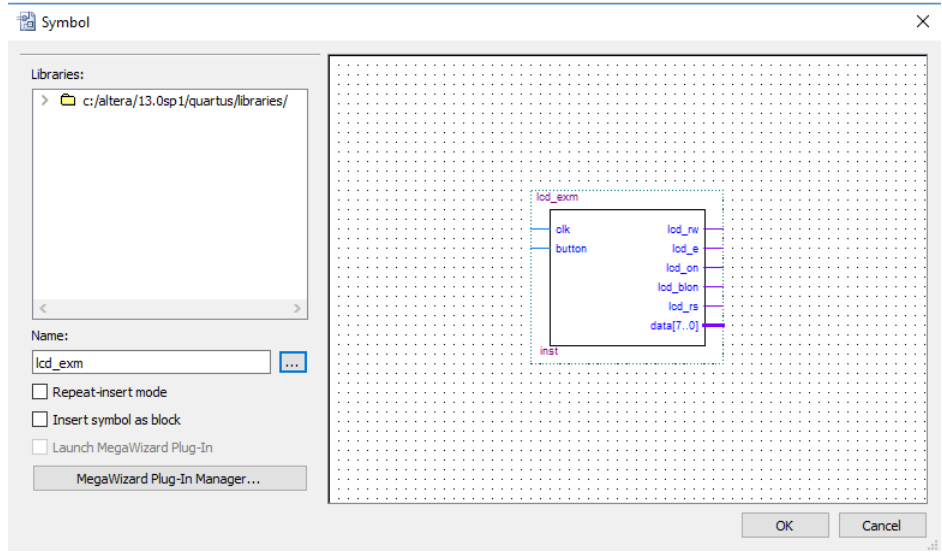
Block Diagram /Schematic File oluşturulur. Bu pencerenin içerisinde mouse'ın sağına tıklayıp **İnsert>symbol** kısmı seçilir. Ve burada servo,yağmur ve lcd'nin diyagramları seçilir.



Şekil 7: Yağmur Sensörü Diyagramı

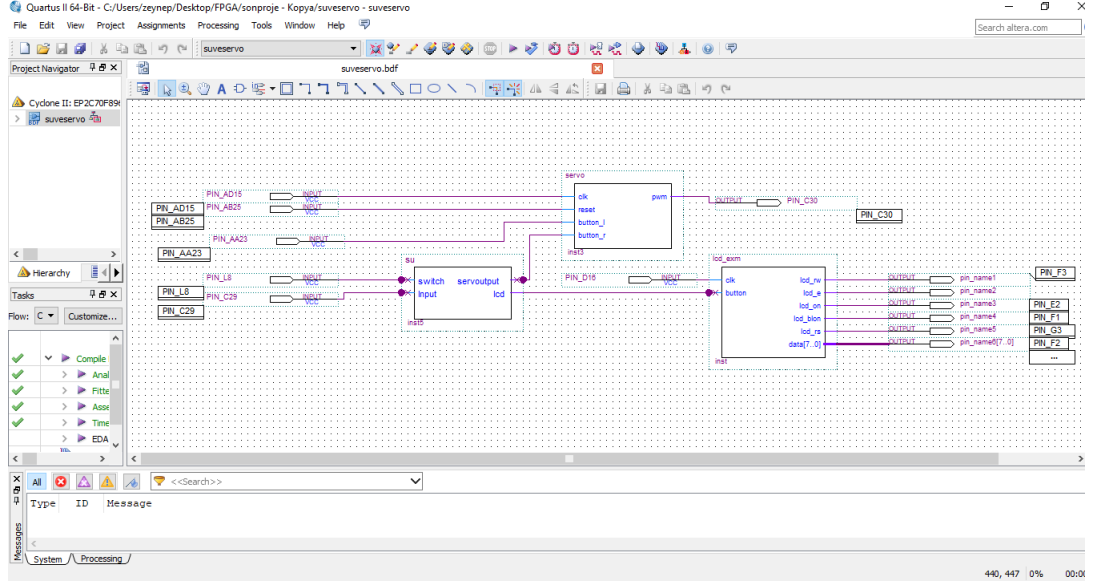


Şekil 8: Servo Motor Diyagramı



Şekil 9: Lcd Diyagramı

Yağmur sensöründen çıkan veri servo motor ve lcd'nin girişine bağlanır. Bu şekilde mekanizma oluşturularak sistemin çıkışı servonun çıkışı olarak ayarlanır.



Şekil 10: Schematic File Tasarımı

Proje derlendikten sonra son aşama olarak Pin Planner'a giriş çıkışlar verilir.

Node Name	Direction	Location	I/O Bank	VREF Group	Fitter Location	I/O Standard	Reserved	Current Strength	Differential Pair
PIN_AA23	Input	PIN_AA23	6	B6_N2	PIN_AA23	3.3-V LV .default		24mA (default)	
PIN_AB25	Input	PIN_AB25	6	B6_N2	PIN_AB25	3.3-V LV .default		24mA (default)	
PIN_AD15	Input	PIN_AD15	7	B7_N3	PIN_AD15	3.3-V LV .default		24mA (default)	
PIN_C29	Input	PIN_C29	5	B5_N0	PIN_C29	3.3-V LV .default		24mA (default)	
PIN_C30	Output	PIN_C30	5	B5_N0	PIN_C30	3.3-V LV .default		24mA (default)	
PIN_D16	Input	PIN_D16	4	B4_N3	PIN_D16	3.3-V LV .default		24mA (default)	
PIN_L8	Input	PIN_L8	2	B2_N1	PIN_L8	3.3-V LV .default		24mA (default)	
pin_name1	Output	PIN_F3	2	B2_N0	PIN_F3	3.3-V LV .default		24mA (default)	
pin_name2	Output	PIN_E2	2	B2_N1	PIN_E2	3.3-V LV .default		24mA (default)	
pin_name3	Output	PIN_F1	2	B2_N2	PIN_F1	3.3-V LV .default		24mA (default)	
pin_name4	Output	PIN_G3	2	B2_N0	PIN_G3	3.3-V LV .default		24mA (default)	
pin_name5	Output	PIN_F2	2	B2_N2	PIN_F2	3.3-V LV .default		24mA (default)	
pin_name6[7]	Output	PIN_B2	2	B2_N0	PIN_B2	3.3-V LV .default		24mA (default)	
pin_name6[6]	Output	PIN_C3	2	B2_N0	PIN_C3	3.3-V LV .default		24mA (default)	
pin_name6[5]	Output	PIN_C2	2	B2_N0	PIN_C2	3.3-V LV .default		24mA (default)	
pin_name6[4]	Output	PIN_C1	2	B2_N0	PIN_C1	3.3-V LV .default		24mA (default)	
pin_name6[3]	Output	PIN_D3	2	B2_N1	PIN_D3	3.3-V LV .default		24mA (default)	
pin_name6[2]	Output	PIN_D2	2	B2_N1	PIN_D2	3.3-V LV .default		24mA (default)	
pin_name6[1]	Output	PIN_E3	2	B2_N0	PIN_E3	3.3-V LV .default		24mA (default)	
pin_name6[0]	Output	PIN_E1	2	B2_N1	PIN_E1	3.3-V LV .default		24mA (default)	

Şekil 11: Pin Planner

## **7 SONUÇLAR VE ÖNERİLER**

Bu projede yağmur sensöründen alınan veriyle servo motor ve lcd çalıştırılması gerçekleştirilmiştir.

## 8 EKLER

### 8.1 Servo Motor Kodları

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity servo is
    Port (
        clk      : in  STD_LOGIC;
        reset    : in  STD_LOGIC;
        button_l : in  STD_LOGIC;
        button_r : in  STD_LOGIC;
        pwm      : out STD_LOGIC
    );
end servo;

architecture Behavioral of servo is
    constant period:integer:=1000000;
    constant dcycle_max:integer:=100000;
    constant dcycle_min:integer:=50000;
    constant duty_in:integer:=50000;
    signal pwm_reg,pwm_next:std_logic;
    signal duty_cycle ,duty_cycle_next:integer:=0;
    signal counter ,counter_next:integer:=0;
    signal tick:std_logic;
begin
    --register
    process (clk , reset)
    begin
        if reset = '1' then
```

```

        pwm_reg<='0';
        counter <=0;
        duty_cycle <=0;
    elsif clk='1' and clk'event then
        pwm_reg<=pwm_next;
        counter<=counter_next;
        duty_cycle <=duty_cycle_next;
    end if;
end process;

counter_next<= 0 when counter = period else
    counter+1;
tick<= '1' when counter= 0 else
    '0';

--Duty Cycle degisimi
process(button_l ,button_r ,tick ,duty_cycle)
begin
    duty_cycle_next<=duty_cycle;
    if tick='1' then
        if button_l ='1' and duty_cycle >dcycle_min then
            duty_cycle_next<=duty_cycle-duty_in;
        elsif button_r ='1' and duty_cycle < dcycle_max
            duty_cycle_next<=duty_cycle+duty_in;
        end if;
    end if;
end process;

--Buffer
pwm<=pwm_reg;
pwm_next<= '1' when counter < duty_cycle else
    '0';

```

```
end Behavioral;
```

## 8.2 Lcd Kodları

```
library IEEE;
```

```
use IEEE.STD_LOGIC_1164.ALL;
```

```
use IEEE.STD_LOGIC_ARITH.ALL;
```

```
use IEEE.STD_LOGIC_UNSIGNED.ALL;
```

```
entity lcd_exm is
```

```
port ( clk      : in  std_logic;
```

```
       button   : in  STD_LOGIC;
```

```
       lcd_rw   : out std_logic;
```

```
       lcd_e    : out std_logic;
```

```
       lcd_on   : out std_logic;
```

```
       lcd_blon : out std_logic;
```

```
       lcd_rs   : out std_logic;
```

```
       data     : out std_logic_vector(7 downto 0));
```



```
end lcd_exm;
```

```
architecture Behavioral of lcd_exm is
```

```
constant N: integer :=21;
```

```
type arr is array (1 to N) of std_logic_vector(7 downto 0);
```

```
constant datas : arr := (X"38",X"0c",X"06",X"01",X"80",x"48",  
x"41",x"56",x"41",x"20",x"59",x"41",x"47",x"4d",x"55",x"52",  
x"4c",x"55",x"20",x"20",x"20");
```

```
constant datas1 : arr := (X"38",X"0c",X"06",X"01",X"80",x"48",  
x"41",x"56",x"41",x"20",x"47",x"55",x"4e",x"45",x"53",x"4c",  
x"49",x"20",x"20",x"20",x"20");
```

```
begin
```

```
lcd_rw <= '0'; --lcd write
```

```
lcd_on <= '1';
```

```
lcd_blon <= '1';
```

```
process (clk)
```

```
variable i : integer := 0;
```

```
variable j : integer := 1;
```

```

begin

if clk'event and clk = '1' then

if i <= 1000000 then

i := i + 1;

lcd_e <= '1';

        if (button='1') then
            data <= datas(j)(7 downto 0);

        elsif (button='0') then
            data <= datas1(j)(7 downto 0);
        end if;

elsif i > 1000000 and i < 2000000 then

i := i + 1;

lcd_e <= '0';

elsif i = 2000000 then

```

```

j := j + 1;

i := 0;

end if;

if j <= 5 then

lcd_rs <= '0';

elsif j > 5 then

lcd_rs <= '1';

end if;

if j = 22 then

j := 5;

end if;

end if;

end process;

end Behavioral;

```

## KAYNAKLAR

- [1] <http://www.elektrikrehberiniz.com/elektrik-motorlari/servo-motor-nedir-526/>
- [2] <https://maker.robotistan.com/rc-servo-motor-nedir/>
- [3] [https://www.robotistan.com/yagmur-sensoru-rain-sensor?\\_sgm\\_campaign=scn\\_8071acc38e6a000&\\_sgm\\_source=1773&\\_sgm\\_action=click](https://www.robotistan.com/yagmur-sensoru-rain-sensor?_sgm_campaign=scn_8071acc38e6a000&_sgm_source=1773&_sgm_action=click)
- [4] <https://www.robotekno.com/arduino-yagmur-\algilama-sensoru-sivi-sensoru-hava-sensoru>

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Zeynep KOTAN  
**Uyruğu** : T.C.  
**Doğum Yeri ve Tarihi:** Aşkale /04.08.1997  
**Adres** : Güneşli 15 Temmuz mah. 1402.sokak no:19/2 Bağcılar/İST  
  
**Telefon** : 539 565 2406  
**e-mail** : zeynepkotan1@gmail.com

### EĞİTİM DURUMU

**Lisans Öğrenimi** : BŞEÜ Bilgisayar Mühendisliği Bölümü  
**Bitirme Yılı** :  
**Lise** : Ahi Evren Anadolu İHL

### İŞ DENEYİMLERİ

**Yıl** :  
**Kurum** :  
**Stajlar** : Staj1

### İLGİ ALANLARI:

**YABANCI DİLLER:** İngilizce:Orta Arapça:Başlangıç

**BELİRTMEK İSTEDİĞİNİZ DİĞER ÖZELLİKLER:**