

SOLAR PANEL TRACKER

Osman Turgut
Sercan Akın
Ali Can
Hüseyin Sertkaya

İÇİNDEKİLER

| | |
|--|----|
| I – Amaç | 3 |
| I.1 Önceki Çalışmalar..... | 3 |
| II – Mikroişlemci Tabanlı Sistem Tasarımı, Donanım..... | 4 |
| II.1 – Proje Birimleri ve Kullanılan Elektronik Donanımlar..... | 4 |
| II.1.1 – Merkezi İşlem Birimi..... | 4 |
| II.1.2 – Giriş Birimi..... | 5 |
| II.1.3 – Çıkış Birimi | 5 |
| II.1.4 – Harici Bellek:..... | 5 |
| II.1.5 – Yollar | 6 |
| II.2 – Elektiksel Blok Diyagramı ve Güç Bütçesinin Çıkarılması | 7 |
| III – Proje Yazılım..... | 8 |
| III.1 – Akış Diyagramı | 8 |
| III.2 – Algoritma | 9 |
| III.3 – Sistemde Kesmeler | 10 |
| III.3.1 – Buton Kesmesi | 10 |
| III.3.2 – Gecikme (Delay) Kesmesi..... | 10 |
| III.3.3 – Sistem Çalışma Mantiğı | 10 |
| III.4 – Akım ve Gerilim Hesabı..... | 10 |
| III.5 – Simülasyon | 11 |
| IV – Proje Uygulama Sonuçları | 12 |
| V – Yorumlar | 13 |
| VI - Kaynakça | 13 |
| VII – Ekler..... | 14 |
| VII.1 – Proje Öneri Formu | 14 |
| VII.2 – Kullanılan Kodlar | 15 |

I – Amaç

Sabit güneş panelleri günün çok azında maksimum verimde çalışmaktadırlar. Ayrıca sabit olmaları sürekli hareket halinde olan gemi uydusu, uzay istasyonu gibi araçlarda kullanımı sınırlamaktadır. Bunun yanında metrekare başına düşen güneş ışını miktarı gibi bilimsel ölçümlerin analizini zorlaştırmaktadır. Bu proje ile hem kendi eksenini hem de horizontal açı hareketi ile günün her saati güneş takibi ile paneli güneşe dik konumlandırılacak panelin maksimum verimi sağlanır. Bu projede üretilen enerjiyi pillerde depolamayı ve akım gerilim sensörü ile üretilen enerjiyi gerçek zamanlı ölçmeyi hedeflemekteyiz. Ayrıca akım gerilim sensöründen gelen bilginin bir SD karta yazılması ile gün saat verim gibi bilgilere ulaşılması sağlanacaktır.

Projemiz enerji tasarrufu için gün içinde çalışacak olup güneşin batmasıyla referans kabul edilen konuma gelecek ve bir sonraki çalışma saatlerinin başlamasını bekleyecektir. Bu beklentinin sağlanması için RTC kesme kullanılmıştır. Ayrıca beklenmesik durumlarda da başlangıç konumuna gelerek durması için acil durum kesmesi kullanılmıştır.

I.1 Önceki Çalışmalar

Bilgin, (2006) yaptığı araştırma sonucunda takip sisteminin sabit sisteme oranla %30 daha fazla verim ürettiği ve özellikle gemi veya karavan gibi araçlarda daha az panel kullanılmasını sağlayarak yer tasarrufu sağlayacağını belirtmiştir. Özellikle mevsimsel değişimlerde dikey tilt açısını değiştirerek %40'lara varan bir enerji verimi elde edilebileceği sonucuna varmıştır. ^[1]

Beyoğlu, (2011) Balıkesir ilinde yaptığı çalışma ile iki eksenli takip sistemi ile sabit fotovoltaik sistemin eşzamanlı olarak verim analizini yapmıştır. Yaptığı çalışma ile 1 yıl boyunca farklı zamanlarda elde ettiği verileri datalogger ile kaydetmiştir. Çalışma 4 süresi sonunda güneş takip sisteminin, sabit sisteme göre %39 daha fazla verim sağladığını tespit etmiştir. Özellikle Karadeniz Bölgesi gibi bulutlu bölgelerde yağışlı günlerin arka arkaya çok olmasından dolayı akülerin boşalabileceği ve bu sebeple daha yüksek kapasiteli akülerin kullanılması gerektiğini belirtmiştir. ^[2]

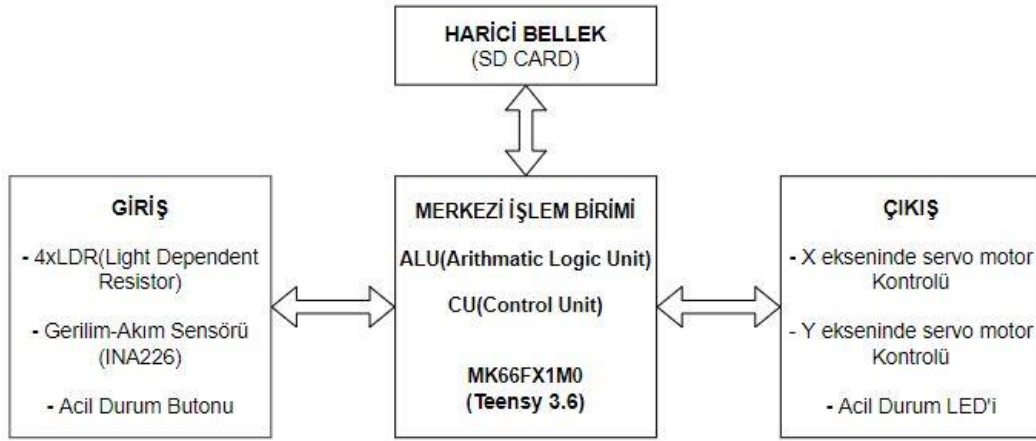
Yılmaz, (2013) Türkiye'nin en çok güneş alan illerinden olan Diyarbakır'da iki eksenli takip sistemi ile sabit sistemi eşzamanlı olarak kurarak 1 yıl boyunca verimlerini incelemiştir. Takip sistemi içerisinde hareketi sağlaması için damperli motor kullanmış böylece daha az kayıp enerji sağlamıştır. Çalışması süresince, kış aylarında %70, yaz aylarında ise %11 oranında iki eksenli takip sisteminin verimli olduğunu belirtmiştir. Özellikle kapladığı alan bakımından takip sistemlerinin kullanılmasının daha avantajlı olacağını vurgulamıştır. ^[3]

Varış, (2017) yaptığı araştırmasında İstanbul Büyükşehir Belediyesi'ne ait sosyal bir tesisin konum bilgilerini kullanarak farklı simulasyon programlarıyla 5,2 kWh değerinde güneş enerji sistemini, farklı günlerde sabit ve takip sistemi olarak simüle etmiştir. Simülasyon sonucunda takip sisteminin sabit sisteme oranla %32 daha verimli olduğu ve takip sisteminin yapılan yatırımı 9,8 yılda amorti ettiğini ortaya koymuştur. ^[4]

II – Mikroişlemci Tabanlı Sistem Tasarımı, Donanım

II.1 – Proje Birimleri ve Kullanılan Elektronik Donanımlar

Yıllar içinde insanlar ihtiyaçlarını gidermek, zorlu işleri kolaylaştırmak ve zamandan tasarruf sağlamak için yeni şeyler keşfetti. El aletleri kullanarak yeni el aletleri üretti, tekerlek ile mesafeleri kolayca aştı ve ateşle ısınma/beslenme ihtiyaçlarını giderdi. Elektrik kullanılması ve telgraf/telefonla haberleşmeyi sağlayan insanlar çok daha hızlı gelişmeye başladı. Bilgiyi aktarabilen insanlar, artık çok karmaşık işlemleri kısa sürede çözmeye ihtiyaçları vardı. Bu noktadan sonra, önce transistörün keşfi ardından da mikro işlemcilerin tasarlanması günümüze kadar yapılan tüm icatların önüne geçti. Bu proje kapsamında gelişmiş bir mikro işlemci kullanılarak dış dünyadan bilgi alınacak ve yine dış dünyaya bilgi aktarılacaktır. Tüm bu işlemler için gerekli birimlere yakından bakalım.



Şekil 1: Kullanılan Birimler ve Aralarındaki Bağlantılar

II.1.1 – Merkezi İşlem Birimi

Mikro işlemci aritmetik ve mantıksal işlemler yapabilen ve bu işlemlerin sonucuna göre çalışmasını yönlendirebilen tümleşik bir devre elemanıdır. Mikro işlemciyi oluşturan temel birimlerden olan ‘Merkezi İşleme Birimi’ dijital veriyi işleyen ve yazılım komutlarını gerçekleştiren birimdir. Projemiz için üzerinde MK66FX1M0 mikrodenetleyicisi bulunan Teensy 3.6 kullanılmaktadır. Şekil 1 de merkezi işlem birimize bağlı giriş-çıkış birimleri ve harici bellek bağlantısı gösterilmektedir. Bu bağlantılar sayesinde mikro işlemcimiz: LDR’lerden gelen verileri karşılaştırarak güneşin konumunu hesaplamakta kullanıldı. Bu veriler doğrultusunda servo motorlar ile güneş panelinin yönlendirilmesi sağlandı. Teensy 3.6(MK66FX1M0)’nın temel bilgileri Tablo 1’de verilmektedir.

Tablo 1: MK66FX1M0 DataSheet Verileri

| İşlemci Hızı | RAM | Flaş Bellek | Çalışma Voltajı | Çalışma Akımı |
|--------------|--------|-------------|-----------------|---------------|
| 180 MHz | 256 kB | 1024 kB | 3.3 V | 79 mA |

II.1.2 – Giriş Birimi

Mikroişlemcimize dış dünyadan aktarılan bilgilerin elde edildiği birimdir. Şekil 1 de giriş birimlerinde kullanılan elemanlar belirtilmektedir. Dış dünyadan ışık şiddeti almak için 4 adet LDR(Light Dependent Resistor) kullanılmaktadır. Bu LDR'ler güneş panelimizin, ölçülen değere göre sağ-sol ve yukarı-aşağı pozisyon almasını sağlamaktadır. Sistemimizin ürettiği ve tükettiği güç, akım-gerilim sensörü olan INA226 ile hesaplanamsı planlandı ve INA226'nın temel özellikleri Tablo 2 de gösterilmektedir. Fakat INA226 kullanılamadığı için gerilim bölücü devresi yapılarakTeensy 3.6'nın analog pininden gerilim hesabı yapıldı. Kullanılan dirençlere göre koldan geçen akım hesaplandı. Bu sayede verim hesabı yapılarak projenin beklenen çoklukları ve sonuçları ilerleyen bölümlerde karşılaştırıldı. Ayrıca sistemimizin acil durumda kesme alt programında girmesini sağlayan bir 'Acil Durum Butonu' kullanıldı.

Tablo 2: INA226 Datasheet verileri

| Ölçülebilen Gerilim Aralığı | Çalışma Gerilimi | Çalışma Akımı |
|-----------------------------|------------------|---------------|
| 0-36 V | 2.7 – 5.5 V | 0.3 mA |

II.1.3 – Çıkış Birimi

Merkezi işlem birimimiz dış dünyadan gelen bilgileri bazı işlemlere tabi tutar. Projemiz de bu işlemler yazılım bölümünde detaylı olarak gösterilmektedir. Giriş biriminden alınan veriler merkezi işlem biriminde işlendikten sonra dış dünyaya (servo motorlara) aktarılırarak panelin x ve y eksenlerinde hareket etmesi için kullanılmıştır. Servo motor olarak kullandığımız 'Tower Pro SG90' temel özellikleri Tablo 3 de verilmiştir. Girişteki acil durum butonuna basıldığında çıkışta uyarı LED'inin yanması sağlandı.

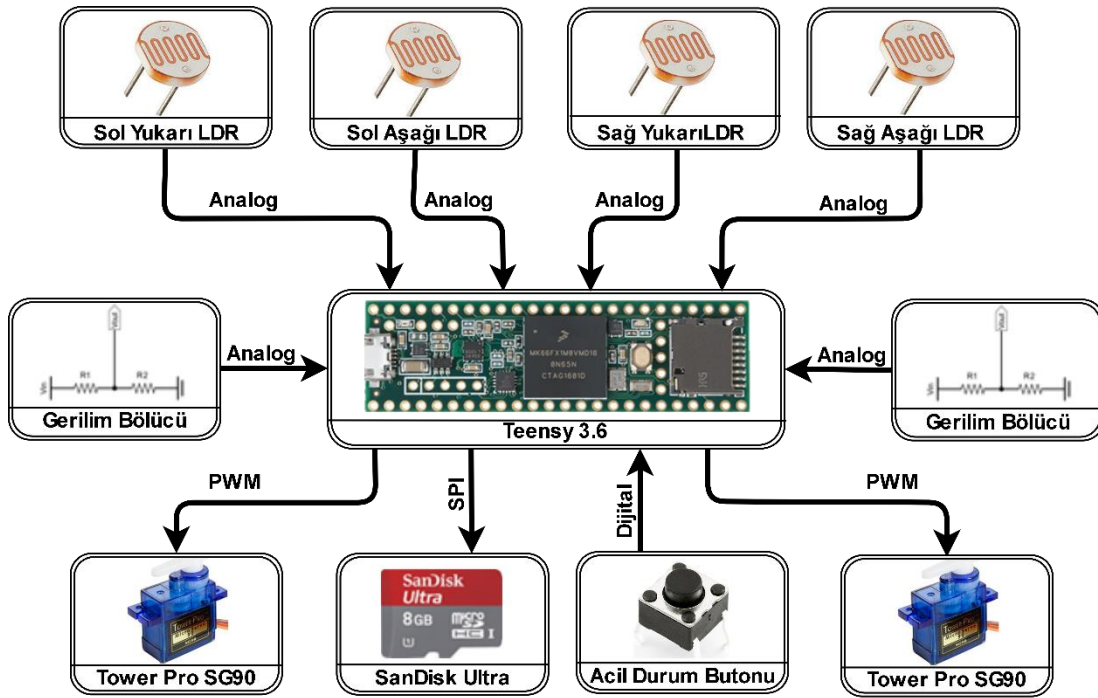
Tablo 3 : Tower Pro SG90 Datasheet Verileri

| Çalışma Gerilimi | Yüklü Tork | Hız (Yüksüz) |
|------------------|----------------|---------------------|
| 4.8 - 7 V | 4,8V 1,3 Kg.cm | 4,8V 0,12/60 Derece |

II.1.4 – Harici Bellek:

Mikroişlemcimiz de akım-gerilim verileri yardımıyla verim hesabı yapıldı. Bu sayede projemizin başarısı kanıtlandı. Fakat bu verilerin kaybolmaması ve istenildiği zaman incelenebilmesi için harici belleğe kayıt ihtiyacı duyarız. Bu sebeple SanDisk Ultra Micro SD kart kullanarak verilerin dış dünyaya aktarılması ve dijital hafıza kartında kaydedilmesi sağlandı. Mikroişlemcimize yazılım yüklenmesi için yeterli dahili kapasiteye sahip olduğu için (Tablo 1) başka bir depolamaya ihtiyaç duyulması.

II.1.5 – Yollar:

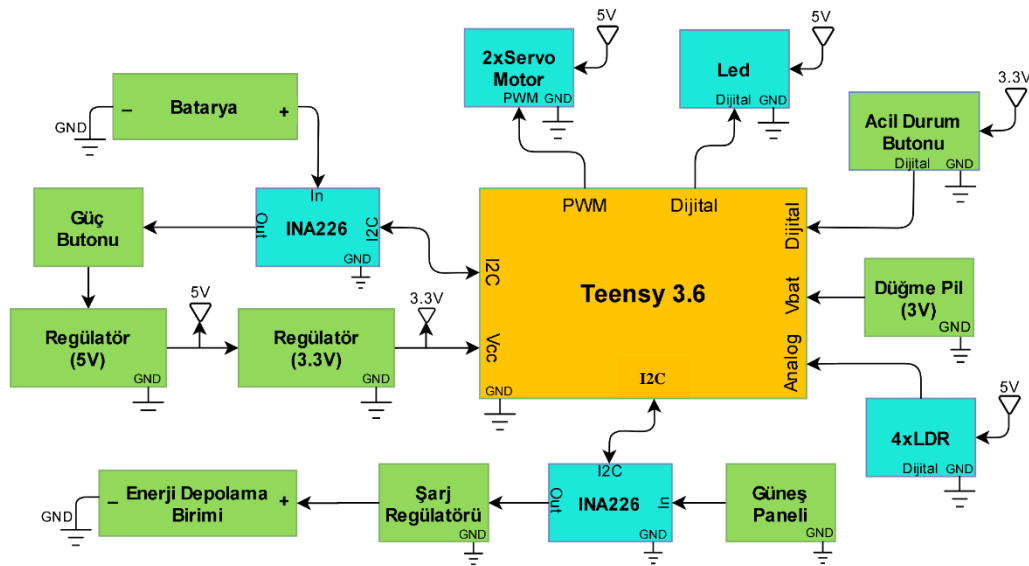


Şekil 2: Önemli Elektronik Komponentlerin Mikroişlemciye Bağlantı Yolları

Dış dünya ile bilgi alış verişi yapabilmek için bazı iletişim yollarına ihtiyaç duyarız. Bu yollar bildiğimiz anlamda Analogi Digital ve PWM sinyalleri olabileceği gibi I2C ve SPI gibi haberleşme protokolleri de olabilir. Projemiz kapsamında belirtilen bağlantıların birçoğu kullanılmış olup Şekil 2 de verilen diyagramda kullanılan önemli komponentlerin bağlantı yolları gösterilmektedir. Burada:

- LDR sürekli zamanlı ölçüm yaptığında (gerilim değişimini dışarı aktarmakta) analog sinyal üretmektedir ve mikroşlemcimizin analog pinine bağlanır.
- Acil durum butonumuz mikroşlemcimize dijital giriş olarak bağlanır. Bu sayede butona basılmadığında 0 volt(Low), basıldığında ise 3.3 volt (High) okuyarak sistemin kesmeye girmesi sağlanacaktır. Benzer işlemi girdi olarak değil de çıktı olarak yaparak, çıkışına bağladığımız LED'in yanmasını sağladık.
- Servo motorlar belli açılarda ve hızlarda konum değişimi yapması beklenmektedir. Bu sebeple dijital ve ya analog sinyal yerine PWM(Pulse Width Modulation) pinlerini kullandık. Bu sayede dijital bir çıkış modüle edilerek belli zaman aralıklarında veri gönderimi sağladı. Gönderilen bu sinyaller kare dalga oluşumunu sağlayarak servo motorumuzun, gecikme kesmesi de kullanılarak, konum alması sağlandı.
- Topladığımız verileri bir SD karta yazmaya ihtiyaç duyarız. Bunun içinde SPI bağlantı yollarını kullanmamız gerekti. SPI bir çeşit senkron seri veri bağlantı standartıdır. Dört pin dizini kullanarak bağlantı sağlanır. Burada bulunan MOSI ve MISO veri aktarımını gerçekleştirdiğimiz pinlerdir. CS pini ile hatta bulunan herhangi bir slave aygıtı seçmemiz sağlanıyor. Clock pini ise seri haberleşme gereği olarak verilerin iletilmesini sağlıyor.

II.2 – Elektiksel Blok Diyagramı ve Güç Bütçesinin Çıkarılması



Şekil 3: Sistemiimizin Elektriksel Blok Diyagramı

Şekil 3 de kurulan blok diyagramın da projemizde kullanılan tüm komponentler, dirençi kapasitör vb.hariç, gösterilemektedir. Burada elemanların bağlantı yolları ve besleme voltajları gösterilmektedir. Burada eğer INA226 akım gerilim sensörü kullanılsaydı, nasıl bağanacağı gösterilmektedir. İki farklı batarya ile sistemin tüketti enerji ve üretilen enerjinin depolanması sağlandı. Gerçek zamanlı saat verisini güç kesinisinde dahi elde edebilmek için harici bir düğme pil kullanıldı. Devre de bulunan 3.3 V ve 5 V regülatörler elektronik elemanların çalışması için gerekli voltajı sağlarken, şarj regülatörü en yüksek enerji deolama voltajı sağlayacak şekilde ayarlanmıştır.

Tablo 4: Güç Bütçesi

| Komponent | Çalışma Gerilimi (V) | Çalışma Akımı (mA) | Harcanılan Güç (W) | Kaynak |
|-----------------------|----------------------|--------------------|--------------------|-------------------------|
| Teensy 3.6 | 3.3 | 79 | 0.260 | Datasheet |
| Tower Pro SG90 | 5 | 2 x 200 | 2 | Datasheet |
| LDR | 5 | 4 x 0.05 | 0.001 | Datasheet |
| Diğer Basif Elemanlar | 3.3 | 10 | 0.033 | Tahmini En Yüksek Değer |

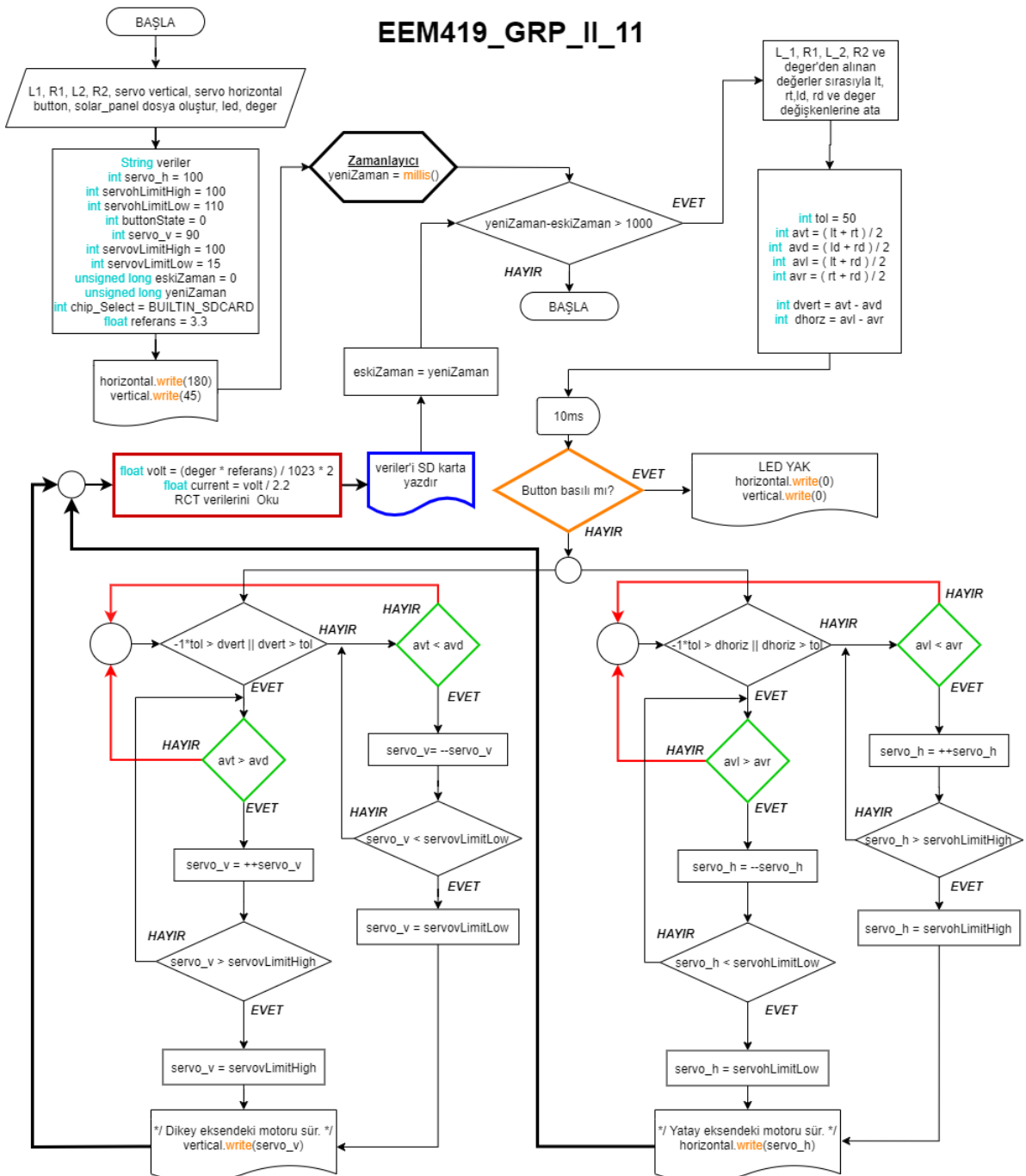
Elektronik komponentlerimizin güç tüketimi Tablo 4 de verilmiştir. Tüketimlerimizi referans alarak sistemimizin toplam ihtiyaç duyduğu güç 2.294 W diyebiliriz. Bataryamız seri bağlanmış 3000 mAh iki adet pilden oluştuğundan dolayı yaklaşık 23 W'lık güç sağlar. Bu da hiç üretim olmadan çok uzun süreler çalışabileceği anlamına gelir. Burada tüm hesaplar olabilecek en yüksek tüketim göz önüne alınarak yapılmıştır. Testlerimizde bu değerlerin altında tüketimle karşılaşır.

III – Proje Yazılım

III.1 – Akış Diyagramı

Şekil 4: Tasarlanan sistemin akış diyagramı.

İlk olarak sistem gerekliliklerini sağlayacak değişkenler tanımlandıktan sonra tasarlanan projenin çalışacağı karar mekanizmalarıyla, sistem tamamlanmıştır. Sistem kesmeler haricinde sonsuz döngüden çıkmayacağı için Şekil 4 de “SON” bloğu kullanılmamıştır. Şekilde renklendirilmiş bloklar bu sistemin özgün dokunuşlarıdır. **EK 2** de uygulama yazılımı verilmektedir.



Şekil 4: Sistemin Akış Diyagramı

III.2 – Algoritma

1. Başla
2. L1, R1, L2, R2, servo vertical, servo horizontal, button, solar_panel dosya oluştur, led, değer
3. Veriler, servo_h = 100, servohLimitHigh = 100, servohLimitLow = 110, buttonState= 0, servo_v = 90, servovLimitHigh = 100, servovLimitLow = 15, eskiZaman = 0, yeniZaman, chip_Select = BUILTIN_SDCARD, referans = 3.3
4. horizontal servosunu 180° ye sür
vertical servosunu 45° ye sür.
5. yeniZaman = millis()
6. Eğer (yeniZaman – eskiZaman > 1000) değil ise git 1
7. L_1, R1, L_2, R2 ve deger'den alınan değerler sırasıyla lt, rt, ld, rd ve deger değişkenlerine ata, tol = 50, avt = (lt + rt) / 2, avd = (ld + rd) / 2, avl = (lt + rd) / 2, avr = (rt + rd) / 2, dvert = avt – avd, dhorz = avl – avr
8. delay(10ms)
9. Eğer butona basılı ise git 36
10. Eğer (-1*tol > dvert || dvert > tol) değil ise git 16
11. Eğer (avt > avd) değil ise git 10
12. servo_v = ++servo_v
13. Eğer (servo_v > servovLimitHigh) değil ise git 11
14. servo_v = servovLimitHigh
15. vertical servosuna servo_v değerini sür ve git 21
16. Eğer (avt < avd) değil ise git 10
17. servo_v = --servo_v
18. Eğer (servo_v < servovLimitLow) değil ise git 16
19. servo_v = servovLimitLow
20. vertical servosuna servo_v değerini sür ve git 21
21. eskiZaman = yeniZaman ve git 6
22. Eğer (-1*tol > dhoriz || dhoriz > tol) değil ise git 28
23. Eğer (avl > avr) değil ise git 22
24. servo_h = --servo_h
25. Eğer (servo_h < servohLimitLow) değil ise git 11
26. servo_h = servohLimitLow
27. horizontal servosuna servo_h değerini sür ve git 33
28. Eğer (avl < avr) değil ise git 22
29. servo_h = ++servo_h
30. Eğer (servo_h > servohLimitHigh) değil ise git 28
31. servo_h = servohLimitHigh
32. horizontal servosuna servo_h değerini sür ve git 33
33. volt = (deger * referans) / 1023 * 2, current = volt / 2.2, RCT verilerini Oku
34. veriler'i SD karta yazdır.
35. eskiZaman = yeniZaman ve git 6
36. LED Yak ve vertical ve horizontal servolarına 0 değerini sür.

Şekil 4'te verilen akış diyagramı gibi algoritmada sistemin ilerleyişini, nasıl bir yol izlendiğini göstermektedir. Algoritma içeriğinde de görüleceği üzere, iki farklı kesme kullanılarak sistemin içinde bulunduğu sonsuz döngüden uzaklaşması sağlanmıştır.

III.3 – Sistemde Kesmeler

III.3.1 – Buton Keskmesi

Şekil 4’te turuncu ile renklendirilmiş karar mekanizmasında ve algoritmada görüldüğü üzere sisteme bir buton entegre edilmiştir. Entegre edilen bu buton “Acil Durum Butonu” olarak belirtilmektedir. Oluşturacak olduğu kesme; butona basıldığında oluşacak sinyal, sistemde bulunan servo motorlarına sıfır değeri gönderilerek sistemin devamlılığı kesilecektir ve butonun aktif olduğunu gösteren bir LED yanacaktır.

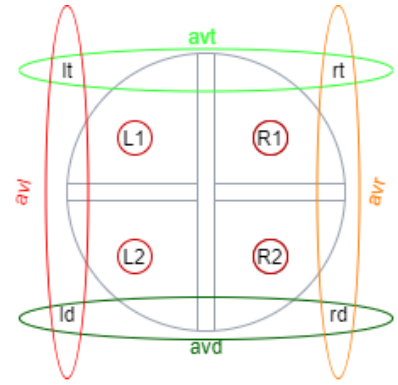
III.3.2 – Gecikme (Delay) Keskmesi

Bu kesmede basit bir şekilde, sistem döngüsünü tamamlayınca 10ms saniye gecikmeye uğrayarak çalışmaya devam ediyor. Bu kesmenin sebebi; Sonsuz döngü içerisindeki sistemin yorulması ile performans düşüklüğünün önüne geçilmesi sağlanmıştır.

III.3.3 – Sistem Çalışma Mantığı

Tasarlanan sistemde LDR’ler Şekil 5’de ki gibi yerleştirilmiştir. Bu sistemde iki eksen kullanılmıştır. Bunlar dikey eksen ve yatay eksen olarak belirtilmiştir.

Yapılan işlem, dikeyde ve yatayda ışık alan noktaları hesaplayıp servo motorları hareket ettirmektir. Şekil 5’te görüldüğü gibi LDRler karşılaştırılabilmek için gruplandırılmış ve bu grupların kendi aralarında ortalaması alınmıştır. Yanlarında görülen avt, avd, avl ve avr bu ortalamaları simgelemektedir.



Şekil 5: Ana Çalışma Mantığı

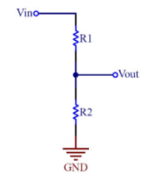
Aynı anda yapılan iki eksenideki hareketler tek tek anlatılacak olunursa; Dikeyde ki hareket için "avt" büyük "avd" ise dikey eksen servo motorunun açısı bir derece artırılır ve işlem başa döner. Eğer bu işlemin tam tersi ise dikey eksen servo motorunun açısı bir derece azaltılır. Aynı işlem yatay eksen için de geçerlidir. "avl" büyük "avr" ise yatay eksen servo motor Şekil 6 da gerilim bölücü devresi gösterilmektedir. bir derece azaltılır. Tam tersi durumda ise bir derece artırılır. Şekil 4’te yeşil ile renklendirilen karar mekanizmaları sistem çalışma mantığını program akışında görülmesini sağlamaktadır.

III.4 – Akım ve Gerilim Hesabı

Proje fikir aşamasında iken projede Akım, gerilim ve güç hesabını yappması için tasarlanacak sisteme INA226 sensörü entegre edilecekti. Ancak INA226 sensörünü temin edilemediği için sistemde alternatif bir yola gidilmiştir. Şekil 4’te kırmızı ile renklendirilen işlem bloğunda;

$$\text{volt} = (\text{deger} * \text{referans}) / 1023 * 2 \quad (1)$$

$$\text{current} = \text{volt} / 2.2 \quad (2)$$



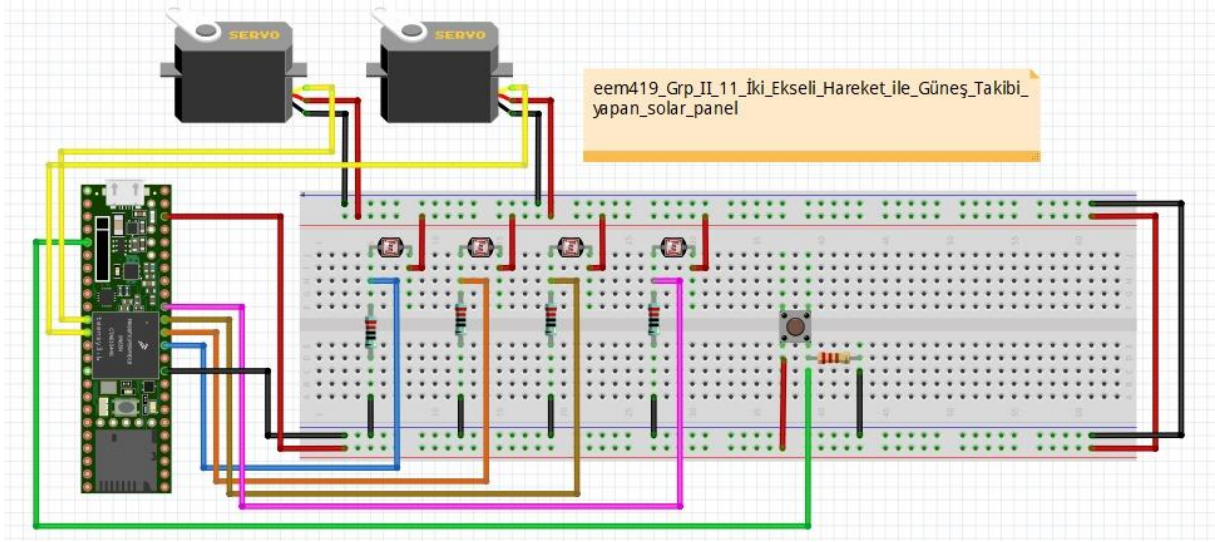
Şekil 6: Gerilim Bölücü Devresi

formülleri ile gerilim ve akım hesaplanarak Şekil 4’te mavi ile renklendirilmiş blok ile SD karta yazdırılma işlemi gerçekleştirilmiştir. Bu işlem, devreye bağlanan iki direnç elemanı üzerinde, gerilim bölücü uygulanarak elde edilmiştir. Şekil 6 da gerilim bölücü devresi gösterilmektedir.

Gerilim bölücü devresinde iki eş direnç kullanıldığı için Teensy 3.6'nın analog pininden okunan voltaj değeri 2 ile çarpılarak gerçek voltaj değeri elde edilmiştir.

III.5 – Simülasyon

İçinde bulunan Covid-19 pandemi süreceği gereğince grup üyeleri olarak, projeyi bireysel olarak da geliştirebilmek için simülasyon ortamlarından yararlanılmıştır. Şekil 7 da görüldüğü üzere TinkerCAD simülasyon programı üzerinden tasarlanan sistem test edilerek sonuca ulaşılmıştır. Bu sayede yazılımızda test edilmiş oldu. Sistemin simülasyon videosu proje dosyasıyla teslim edilmiştir.



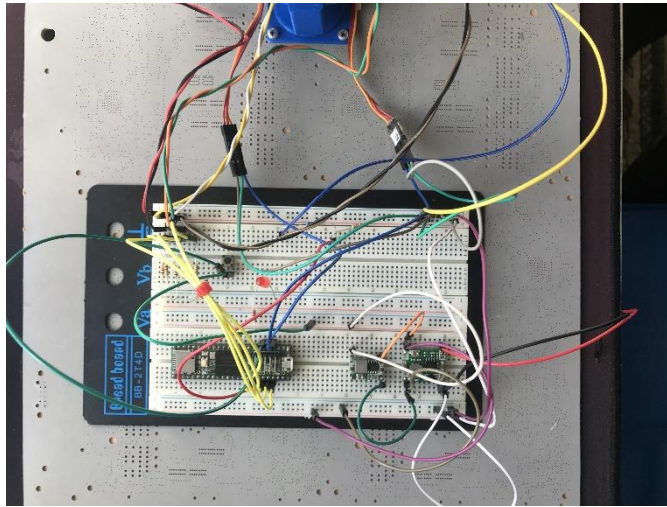
Şekil 7: Simülasyon Ortamında Kurulan Devre

IV – Proje Uygulama Sonuçları

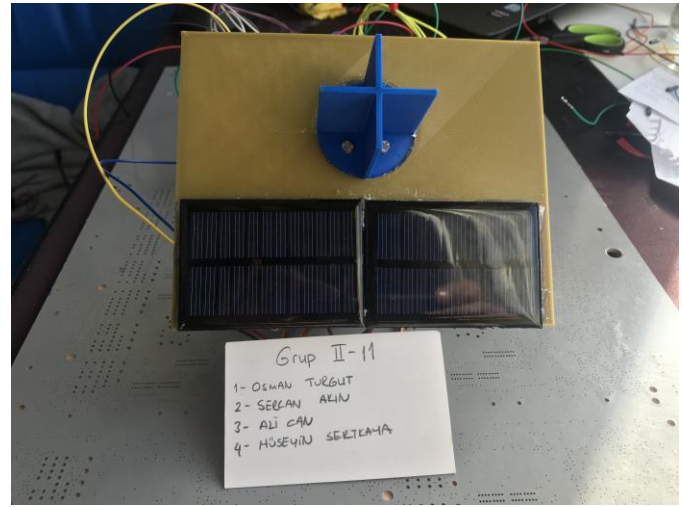
Yapmış olduğumuz testler sonucunda eşit panel yüzeyine sahip sabit güneş paneli ile güneş takibi yapan projemiz arasında 25 Türk Lirası kadar(Servo ve LDR kaynaklı) kadarlık bir fiyat farkı olduğu gözlemlenmiştir. Enerji verimlerini kıyaslayacak olursak %5-%7'lik verim farkı ile güneş takip sistemli projemizin öne çıktığı gözlemlenmiştir. Bu verim artışını arttırmak için yüksek çözünürlüklü güneş sensörleri, daha kararlı servo motorlar ve yüksek verimli güneş panelleri kullanılarak verimin artması sağlanır.

Projemizde kullanılan komponentler yaklaşık 600 Türk Lirası değerindedir. Mikroişlemci gibi pahalı komponentler zaten elimizde mevcuttu. Bu sebeple kişi başı düşen maliyet ciddi anlamda düşürülerek sistemimiz toplandı.

Sistemimiz 'BroadBoard' üzerinde kurularak denenmiştir. Şekil 8.a da kurulan devre, Şekil 8.b de güneş panellerimiz ve güneş ışığının geldiği yönü bulmak için kullandığımız LDR'lerin görseli mevcuttur.



a.



b.

Şekil 8: Kurduğumuz Sistemin Görselleri

Verilerimizin kaybolmaması için SD karta yazdırıldığını belirtmiştik. Bunun yanı sıra toplanan veriler 'Arduino ID'si kullanılarak seri port ekranında yazdırılmıştır. Yazdırılan bu verilerin ekran görüntüleri Şekil 9 da gösterilmektedir. Burada zaman verileri Teensy 3.6'nın dahili RTC'den alınmıştır. Bunun yanı sıra gerilim-akım değerlerimiz ve panellerimizin baktığı yön(motor konumları olarak) resimde görülmektedir.

| COM3 (Teensy) Serial | | | | | | | | | | | | | | |
|----------------------|------|----------|---------|-----------|-----------|------------------|---------------|-------------|----------------|-----------------|--|--|--|--|
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:9 | Analog Deger:714 | Voltaj: 4.61V | Akım:2.09mA | Yatay konum:90 | Dikey Konum:100 | | | | |
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:10 | Analog Deger:623 | Voltaj: 4.02V | Akım:1.83mA | Yatay konum:90 | Dikey Konum:100 | | | | |
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:11 | Analog Deger:519 | Voltaj: 3.35V | Akım:1.52mA | Yatay konum:90 | Dikey Konum:100 | | | | |
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:12 | Analog Deger:578 | Voltaj: 3.73V | Akım:1.70mA | Yatay konum:90 | Dikey Konum:100 | | | | |
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:13 | Analog Deger:500 | Voltaj: 3.23V | Akım:1.47mA | Yatay konum:90 | Dikey Konum:100 | | | | |
| Gün:12 | Ay:1 | Yıl:2021 | Saat:17 | Dakika:10 | Saniye:14 | Analog Deger:490 | Voltaj: 3.16V | Akım:1.44mA | Yatay konum:90 | Dikey Konum:100 | | | | |

Şekil 9: Arduino Seri Monitör Ekranına Alınan Çıktılar

V – Yorumlar

Projemiz sabit eksenli panellere göre daha pahalı olmasına rağmen uzun vade de enerji veriminin fazla olması sayesinde daha avantajlıdır.

Hazırlanmış projede belirlenen ilk hedefler doğrultusunda INA226 gerilim-akım sensörü kullanılması düşünülmekteydi. Fakat pandemi sürecinde karşılaşılan sorunlar neticesiyle INA226 sensörü tedarik edilemedi. Bu sebeple sistemimizde gerilim bölücü metodu kullanılarak panelin gerilimi-akımı ve pilin doluluk seviyesi ölçülmüştür. Bu da maliyetin düşmesini sağladı.

SD kartta yazdırılan bilgiler ile projemizin çalışma saatleri içinde gün saat verim analizi yapılmıştır. Ayrıca sisteme eklenebilecek bir haberleşme modülü ile pillerin doluluk durumları gerçek zamanlı olarak kontrol edilebilir. Bu sayede bir çok hareketli takip sistemi kurularak veriler merkezi bir yerde depolanabilir ve uzun vade de güneş tarlaları kurulabilir.

VI - Kaynakça

[1] Bilgin, Z. (2006). Güneştakip sistemi tasarımı ve gerçekleştirilmesi. Yüksek Lisans Tezi, Gazi Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Ankara.

[2] Beyoğlu M. (2011). Balıkesir ilinde çift eksenli güneş takip sistemi ile sabit eksenli pv sistemin verimlerinin karşılaştırılması. Balıkesir Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik-Elektronik Mühendisliği Anabilim Dalı, Balıkesir

[3] Yılmaz, M. (2013). Güneştakip sistemi ile güneş enerjisinden elektrik enerjisi elde etme yöntemleri ve optimum verimin belirlenmesi. Yüksek Lisans Tezi, Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Eğitimi Anabilim Dalı, İstanbul.


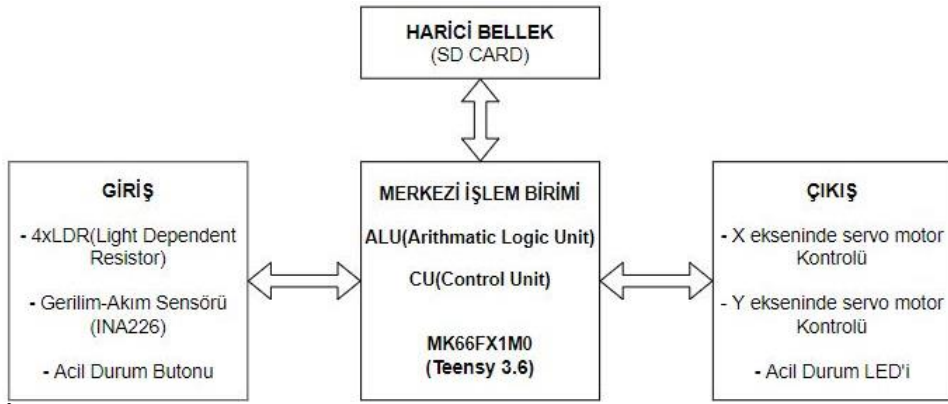
[4] Varış, Ç. (2017). Çift eksenli güneş takip sisteminin pilot uygulaması, üretim değerlendirmesi ve ekonomik analizi. Yüksek Lisans Tezi, Yıldız Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Elektrik Mühendisliği Anabilim Dalı, İstanbul.

VII – Ekler

VII.1 – Proje Öneri Formu

EEM 419 Mikroislemciler

Güz 2020

| | | |
|---|---|---|
| Proje Adı: İKİ EKSENİ HAREKET İLE GÜNEŞ TAKİBİ YAPAN SOLAR PANEL TASARIMI | | |
| Grup Üyeleri: 1. 150106203056 Osman TURGUT (No,Ad-Soyad) 2. 150106203066 Sercan AKIN 3. 170106203017 Ali CAN 4. 170106203065 Hüseyin SERTKAYA |  | I. <input type="checkbox"/> II. <input checked="" type="checkbox"/> Grup No: II-11 |
| Proje Amacı: Güneş enerjisinin popülaritesi dünya çapında artmaktadır. Ülkemizde güneş enerjisi için mükemmel bir potansiyel bulunmaktadır. Güneş takip cihazları güneş panellerinden elde edilen enerjinin artırılması için kullanılan sistemlerdir. Dünyanın güneşe göre dönüşü tüm yıl boyunca aynı olmadığından iki eksenli bir izleme sistemi güneşi doğrudan takip ederek elde edilen enerjinin artırılması hedeflenmektedir. | | |
| Proje Birimleri:  | | |
| Merkezi İşlem Birimi: LDR'lerden gelen verileri karşılaştırarak ışık şiddetinden güneşin konumunu hesaplayacaktır. Bu veriler doğrultusunda servo motorlar ile güneş panelinin yönlendirilmesi sağlanacaktır. | | |
| Giriş Birimi: Dış dünyadan ışık şiddeti 4 adet LDR kullanılarak ölçülecektir. Sistemin ürettiği ve tükettiği güç, gerilim-akım sensörü ile hesaplanacaktır. Acil durumlarda sistemin durdurulması için bir buton kullanılacaktır. | | |
| Çıkış Birimi: Giriş biriminden alınan verilerin merkezi işlem biriminde işlendikten sonra dış dünyaya servo motorlar ile x ve y eksenlerinde hareket sağlayacaktır. Girişteki acil durum butonuna basıldığında çıkışta uyarı LED'i yanacaktır. | | |
| Harici Bellek: Gerilim ve akım sensöründen alınan verilerin SD karta depolandığı birimdir. | | |
| Yollar: I2C, SPI, Analog, Digital ve PWM ile birimler arasındaki haberleşmeyi sağlayan bağlantılardır. | | |
| Beklenen Çıktılar: <ul style="list-style-type: none">✓ Güneşten gelen ışınlarının güneş paneline dik açı ile ulaşabilmesi için servo motorların doğru konumda bulunması.✓ SD karta depolanan veriler doğrultusunda enerji üretimi için uygun saatlerin kullanıcıya sunulması.✓ Güneş takip sisteminin beklenmeyen bir durumda sisteme ve çevresine zarar vermemesi için acil durum butonu ile güvenlik sağlanması.✓ Güç analizi yapılarak güneş panelinin veriminin artırılması. | | |

VII.2 – Kullanılan Kodlar

```
#include <Servo.h>
#include <SD.h>
#include <SPI.h>
#include <TimeLib.h>

String veriler;
File solar_panel;
int chip_Select = BUILTIN_SDCARD;

Servo horizontal; // Yatay Servo Motor
int servo_h = 100; // Yatayda ki Servo motorun Harekete Başlaayacağı konum
int servohLimitHigh = 100; // Yatayda ki Servo motorun ulaşabileceğ en ust aci degeri
int servohLimitLow = 110; // Yatayda ki Servo motorun ulaşabileceğ en alt aci degeri
int tol = 50;
int dvert; // Alt ve Üst Sensörlerin Farkı
int dhoriz; // Sağ ve Sol Sensörlerin Farkı
int avt;
int avr;
int avd;
int avl;

const int buttonPin = 4; // Button Pin
const int ledPin = 5; // LED Buton
int buttonState = 0; // Buton durumu belirlendi

Servo vertical; // Dikey Servo

int servo_v = 90; // Dikeyde ki Servo motorun Harekete Başlaayacağı konum
int servovLimitHigh = 100; // Dikeyde ki Servo motorun ulaşabileceğ en ust aci degeri
int servovLimitLow = 15; // Yatayda ki Servo motorun ulaşabileceğ en alt aci degeri

int L_1 = A8; // Sol Üst LDR Sensör Pini
int R1 = A7; // Sağ Üst LDR Sensör Pini
int L_2 = A9; // Sol Alt Sensör Pini
int R2 = A6; // Sağ Alt Sensör Pini
float deger = A5; // Güneş paneli gerilim ölçümü
float referans = 3.3;
unsigned long eskiZaman = 0;
unsigned long yeniZaman;

float current;
void setup()
{
    Serial.begin(9600);

    SD.begin(BUILTIN_SDCARD);
    horizontal.attach(3); // Yatay Ekseni Servo Motor Pini
    vertical.attach(2); // Dikey Ekseni Servo Motor Pini
```

```

horizontal.write(180);           // Yatay Eksen 180 dereceye gönderildi.
vertical.write(45);

pinMode(ledPin, OUTPUT);         // LED Çıkış olarak ayarlandı.
pinMode(buttonPin, INPUT);       // Buton giriş olarak ayarlandı.

setSyncProvider(getTeensy3Time);

}

void loop()
{
    if (Serial.available()) {
        time_t t = processSyncMessage();
        if (t != 0) {
            Teensy3Clock.set(t); // set the RTC
            setTime(t);
        }
    }
}

yeniZaman = millis();
if (yeniZaman - eskiZaman > 1000) {

    int lt = analogRead(L_1);      // Sol Üst LDR Sensörün Degerini "lt" değişkenine atandı.
    int rt = analogRead(R1);      // Sağ Üst LDR Sensörün Degerini "rt" değişkenine atandı.
    int ld = analogRead(L_2);      // Sol Alt LDR Sensörün Degerini "ld" değişkenine atandı.
    int rd = analogRead(R2);      // Sağ Alt LDR Sensörün Degerini "rd" değişkenine atandı.
    int deger = analogRead(deger); // Güneş paneli gerilimini oku.

    int dtime = 10;
    int tol = 50;

    int avt = (lt + rt) / 2;        // Üst Sensör Verilerinin Ortalaması
    int avd = (ld + rd) / 2;        // Alt Sensör Verilerinin Ortalaması
    int avl = (lt + ld) / 2;        // Sol Sensör Verilerinin Ortalaması
    int avr = (rt + rd) / 2;        // Sağ Sensör Verilerinin Ortalaması

    int dvert = avt - avd;          // Alt ve Üst Sensörlerin Farkı
    int dhoriz = avl - avr;         // Sağ ve Sol Sensörlerin Farkı

    //Serial.print(avt);
    //Serial.print(" ");
    //Serial.print(avd);
    //Serial.print(" ");
    //Serial.print(avl);
    //Serial.print(" ");
    //Serial.print(avr);

```



```

//Serial.print(" ");
//Serial.print(dtime);
//Serial.print(" ");
//Serial.print(tol);
//Serial.println(" ");
buttonState = digitalRead(buttonPin);
if (buttonState == HIGH) {
    warning();
} else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
}

if (-1 * tol > dvert || dvert > tol)
{
    if (avt > avd)
    {
        servo_v = ++servo_v;
        if (servo_v > servovLimitHigh)
        {
            servo_v = servovLimitHigh;
        }
    }
    else if (avt < avd)
    {
        servo_v = --servo_v;
        if (servo_v < servovLimitLow)
        {
            servo_v = servovLimitLow;
        }
    }
    vertical.write(servo_v);
}

if (-1 * tol > dhoriz || dhoriz > tol)
{
    if (avl > avr)
    {
        servo_h = --servo_h;
        if (servo_h < servohLimitLow)
        {
            servo_h = servohLimitLow;
        }
    }
    else if (avl < avr)
    {
        servo_h = ++servo_h;
        if (servo_h > servohLimitHigh)
        {
            servo_h = servohLimitHigh;
        }
    }
}

```

```

    }
}
else if (avl = avr)
{

}
horizontal.write(servo_h);
}
float volt = (deger * referans) / 1023 * 2;
digitalClockDisplay();
veriler += " Analog Deger:";
veriler += deger;
veriler += " Voltaj: ";
veriler += volt;
veriler += "V";
veriler += " Akım:";
current = volt / 2.2;
veriler += current;
veriler += "mA";
veriler += " Yatay konum:";
veriler += servo_v;
veriler += " Dikey Konum:";
veriler += servo_h;

Serial.println(veriler);
sd_card();
eskiZaman = yeniZaman;
veriler = "";
}

}
void warning() {
    horizontal.write(servo_h);           // Yatay Eksen 180 dereceye gönderildi.
    vertical.write(servo_v);
    digitalWrite(ledPin, HIGH);
}

void sd_card()
{
    solar_panel = SD.open("Veriler.txt", FILE_WRITE);
    if (solar_panel)
    {
        solar_panel.println(veriler);
        solar_panel.close();
    }
}

void digitalClockDisplay() {
    // RTC verileri veriler stringinin içerisine atılıyor.

```

```

veriler += " Gün:";
veriler += day();
veriler += " Ay:";
veriler += month();
veriler += " Yıl:";
veriler += year();
veriler += " Saat:";
veriler += hour();
veriler += " Dakika:";
veriler += minute();
veriler += " Saniye:";
veriler += second();
}

time_t getTeensy3Time()
{
    return Teensy3Clock.get();
}

#define TIME_HEADER "T"

unsigned long processSyncMessage() {
    unsigned long pctime = 0L;
    const unsigned long DEFAULT_TIME = 1357041600; // Jan 1 2013

    if(Serial.find(TIME_HEADER)) {
        pctime = Serial.parseInt();
        return pctime;
        if( pctime < DEFAULT_TIME) { // check the value is a valid time (greater than Jan 1 2013)
            pctime = 0L; // return 0 to indicate that the time is not valid
        }
    }
    return pctime;
}

void printDigits(int digits){
    // utility function for digital clock display: prints preceding colon and leading 0
    Serial.print(":");
    if(digits < 10)
        Serial.print('0');
    Serial.print(digits);
}

```