Lidar Menzil Verisinin Wifi Modülü ile Thingspeak'e Aktarılması

Ahmet Haydar ERDEM  
Elektronik ve Haberleşme Mühendisliği

Kocaeli Üniversitesi

Kocaeli, Türkiye  
205103040@kocaeli.edu.tr

***Abstract*— Bu çalışmada SoC miarisi kullanılarak farklı uart kanallarından sensörlerin kontrol edilmesi işlemleri gerçekleştirilmiştir. Uart kanalları SoC mimarisinin Programmable Logic kısmında bulunmaktadır. AXI arayüzünden Processing System'de bulunan uygulamalar ile kontrol edilmektedir.**

***Anahtar Kelimeler—AXI, PL, PS, SoC, Uart, AT Komutları, Lidar, Wifi, LVTTL, SDK, MQTT, HTTP, TCP/IP***

* Giriş

Günümüzdeki teknolojik gelişmeler dikkate alındığında haberleşme alanında çok hızlı gelişmeler olduğu görülmektedir. Bu gelişmeler genel olarak veri aktarım hızı, maliyet, boyut, erişim alanı gibi özelliklerde yoğunlaşmaktadır. Bu gelişmeler dikkate alındığında kablolu haberleşme yerine kablosuz haberleşme kullanımı hem erişim alanını arttırması hem de maliyeti açısından büyük önem arz etmektedir. Bu çalışmada belirtilen özellikler dikkate alınarak kablosuz veri transferi işleminin gerçekleştirilmesi amaçlanmıştır. Çalışmada oluşturulan sistemin genel yapısı şekilde gösterilmiştir.



Fig. 1. Genel Mimari

* Genel Bakış

Bu çalışmada wifi ve lidar sensörleri kullanılmıştır. Bu sensörlerin haberleşme kanalları uart arayüzüne sahiptir. Lidar sensörü olarak Benewake firmasının tf02-pro lidar sensörü, wifi sensörü olarak Espressif System firmasının ürettiği esp8266 seçilmiştir. Sensörlerle alakalı ayrıntılı bilgi kendi başlıkları altında verilmiştir.

* *Lidar Sensörü (TF02-PRO Sensörü)*

Benewake firmasının tf02-pro lidar ürünü orta menzilli sensör kategorisinde bulunmaktadır. Fiziksel arayüz olarak güç hattı ve uart haberleşme hattına sahiptir. Haberleşme seviyesi olarak LVTTL (3.3V) belirtilmiştir. 5-12 V güç beslemesi kabul eden sensörün ortalama akım değeri 200mA, ortalama gücü 1W civarında olduğu belirtilmiştir. TF02-Pro'nun algılama açısı (FOV) 3° 'dir. Bu sebeple farklı mesafelerde cisim algılama aralığının boyutu değişmektedir. Lazer sinyalinin teorikte doğrusal olarak yayıldığını varsayarsak hesaplama yapıldığı zaman 40m menzildeki bir cismin algılanabilmesi için 208cm boyutundan büyük olması gerekmektedir. [1]



Table 1. Menzil ve Algılanabilir Boyut Tablosu

* Haberleşme Protokolü

Haberleşme protokolü olarak seri haberleşme arayüzü olan uart kullanılmıştır. Sensörün iki farklı modu bulunmaktadır. Bu modlar konfigürasyon modu ve standart veri çıkış modudur. Veri çıkış modunda sensör menzil, sinyal gücü ve sıcaklık bilgilerini uart üzerinden göndermektedir. Paket yapısını ve veri gönderme hızını değiştirmek için konfigürasyon modunu kullanmak gerekmektedir. Bu modda ise sensör reset, frame rate, veri çıkış formatı (cm, m, mm), bud rate gibi ayarlar yapılmaktadır. Gerekli ayarlar yapıldıktan sonra kongigürasyon modundan çıkılmalıdır. Böylece sensör yeni ayarlara göre veri göndermeye başlamaktadır.

* *Wifi Sensörü (ESP8266 Wifi Modülü)*

Espressif System tarafından üretilen esp8266 wifi modülü çoğunluka hobi olarak veya IoT çalışmalarında kullnılmaktadır. Modül 2.4GHz-2.5GHz frekans aralığında haberleşme yapmaktadır. Operasyon voltajı 3.0-3.6V olarak belirtilmiştir. Wifi modlarından station, softAP ve station+softAP modları kullanılabilmektedir. Sensör güvenlik modlarından WPA/WPA2 'ye sahiptir. Şifreleme yapılarından WEP/TKIP/AES kullnılmaktadır. Sensörün kullandığı ağ protokolleri IPv4/ TCP/ UDP/ HTTP/ FTP' dir. Kullanıcı konfigürasyonu sağlamak için AT komut seti, Cloud Sunucuları veya Android/iOS uygulamaları kullnılabilir. Bu çalışmada AT komut seti kullanılmıştır.

* AT Komutları

AT komut seti (Hayes komut seti olarak da bilinir), başlangıçta Dennis Hayes tarafından 1981'de Hayes Smartmodem 300 baud modem için geliştirilen özel bir komut dilidir[]. Komut seti, çevirme, kapatma ve bağlantı parametrelerini değiştirme gibi işlemler için komutlar üretmek üzere birleştirilebilen bir dizi kısa metin dizisinden oluşur.

"AT", 'dikkat' anlamına gelir. Her bir komut dizesinin önünde "AT" bulunur ve bir dizi ayrı komut, "AT" harfinden sonra birleştirilebilir.[2]

AT komut seti kablosuz haberleşme kullanan gsm, gprs, wifi gibi sensörler tarafından sıkça kullanılmaktadır. Bu çalışmada wifi modülünü konfigüre etmek amacıyla çeşitli AT komutları kullanılmıştır. Kullanılan bazı komutlar şekilde gösterilmektedir.



Fig. 2. AT Komutları [3]

* *MQTT ve HTTP* [4]

MQTT(Message Queuing Telemetry Transport) protokolü, internette yaygın olarak kullanılan makinalar arası (M2M) mesaj tabanlı bir protokoldür. Lightweight oluşu ve düşük kaynak tüketmesiyle Internet of Things(IoT) ekosisteminde benimsenmiştir. Hemen hemen tüm IoT bulut platformları akıllı nesnelerden veri gönderip almak için MQTT protokolünü desteklemektedir.

Bu protokol, istek(request)-yanıt(response) yapısına dayalı HTTP’ye karşıt olarak yayın(publish)-abone(subscriber) yapısında TCP/IP bağlantısı kurulur. TCP/IP protokolünün bulunduğu Linux, Windows, Android, iOS, MacOS gibi işletim sistemlerinde çalışabilmektedir.

* MQTT Mesaj Yapısı

MQTT protokolü yayıncı-abone yapısında bir mesaj yayınlayan bir client (yayıncı) mesajı alan diğer clientlara ayıracaktır (aboneler). Ayrıca, MQTT asenkron protokoldür, bu da mesajı beklerken clientı engellemediği anlamına gelir. HTTP protokolünün aksine, esas olarak eşzamanlı bir protokoldür. MQTT protokolünün bir başka özelliği, istemcinin (abone) ve yayıncının aynı anda bağlı olmasını gerektirmemesidir.

* MQTT Yayıncı-Abone Mimarisi

MQTT’deki kilit unsur MQTT brokerıdir. MQTT brokerın asıl görevi, clientlara (abonelere) mesajlar göndermektir. Yani yayıncıdan mesajlar alır ve bu mesajları abonelere gönderir. Mesaj gönderirken, MQTT broker mesajı alacak olan clientları filtrelemek için konuyu(topic) kullanır. Konu bir dizedir ve konu seviyeleri yaratan konuları birleştirmek mümkündür.



Fig. 3. Örnek Ölçeklenmiş Topic Tanımlaması [5]

Konu(topic) bir yayıncıyı abonelerine bağlayan sanal bir kanala benzer. Bu konu MQTT brokerı tarafından yönetilmektedir. Bu sanal kanal sayesinde, yayıncı abonelerden ayrılmıştır ve istemcilerin(yayıncılar veya aboneler) birbirlerini tanıması gerekmemektedir. Bu yapısı gereği bu protokolü mesaj üreticisine(yayıncı) ve mesaj tüketicisine(abone) doğrudan bağımlılık olmadan çok ölçeklenebilir hale getirir.



Fig. 4. Örnek MQTT Broker-Client Bağlantısı [6]

* Hyper Text Transfer Protocol (HTTP)

World Wide Web (WWW) tarafından mesajlarının nasıl iletileceğini ve biçimlendirileceğini tanımlamak için kullanılır. Bu protokol, bir sunucunun ağ üzerinden bilgi gönderirken yapması gereken işlemden sorumludur. Tarayıcıya bir URL girilirken, bu protokol sunucuya bir HTTP isteği gönderir ve ardından tarayıcıya bir HTTP yanıtı gönderilir. Bu protokol ayrıca, biçimlendirmeleri ve temsilleri için World Wide Web üzerindeki web sayfalarının kontrolünden de sorumludur.

* MQTT ve HTTP Protokollerinin Karşılaştırılması

MQTT ve HTTP protokollerinin bazı parametrelere göre karşılaştırılmasını konu alan bir tablo hazırlanmıştır.



Table 2. MQTT VE HTTP Karşılaştırma Tablosu [7]

* Proje Geliştirme Aşamaları

Proje çalışması kapsamında öncelikli olarak uart arayüzleri oluşturuldu. Bu kapsamda PL kısmında Xilinx'in sunduğu AXI-Uart Lite IP'si kullanılmıştır. Her kanal için kullanılan bu IP AXI-Lite arayüzü üzerinden Zynq PS tarafına bağlanmıştır. Bu şekilde oluşturulan tasarım sentez aşamalarından sonra .bit dosyası SDK ortamına aktarılmıştır.

Tasarımın donanım platformu kullanılarak SDK üzerinde uygulama oluşturuldu. Bu uygulama içerisinde sensörler için header ve c driver dosyaları oluşturulmuştur. Header dosyaları içerisinde kullanılan kütüphaneler, genel ve sabit tanımlamalar, fonksiyon prototipleri ve veri tipi tanımlamaları eklenmiştir.

Uart haberleşme arayüzünü kullanabilmek için yine benzer çalışmalar yapılmış ve driver dosyaları oluşturulmuştur. Sonrasında uart driverları sensör driverları içerisine eklenmiştir.

Yazılım geliştirme aşaması tamamlandıktan sonra sensörler fiziksel olarak sisteme entegre edilerek donanım üzerinde testlere başlanmıştır. Bu aşamada yazılım doğrulama açısından ftdi ttl to uart dönüştürücü ile terminal üzerinden veri alış verişi çalışması etkili olmuştur.

Bakıldığı zaman ilk aşamada yazılım çalışmaları yapılmıştır. Sonraki aşamada sensörlerin bu yazılımlarla ayağa kaldırılması işlemi uygulanmış, son aşamada tüm sistemin test edilmesi hedeflenmiştir. Belirtilen çalışmalar blok diyagram olarak şekilde gösterilmiştir.



Fig. 3. Proje Aşamaları ve İş Paketleri

* Proje Çıktıları,

Lidar sensörü donanım üzerinde test edilmiş ve terminal üzerinden çıktılar doğrulanmıştır. Bu kapsamda lidar sensöründen uart kanalı üzerinden gelen paketler ayrıştırılarak menzil ve sinyal gücü bilgileri elde edilmiştir. Bu bilgiler kullanıcı için terminale basılmıştır. İlgili çalışmanın çıktıları şekildeki gibidir;



Fig. 4. Lidar Menzil ve Sinyal Gücü Terminal Çıktısı

Wifi sensörü içi oluşturulan yazılım çalışmaları ftdi ile terminal üzerinden test edilmiştir. İlgili paketlerin doğru aktarımı terminal üzerinde görülmesine karşın wifi sensörü donanım üzerinde ayağa kaldırılamamıştır. Wifi modülü donanımdan bağımsız terminal üzerinden test edildiğinde de benzer problemlerin yaşandığı tespit edilmiştir. Dolayısıyla bu çalışma yarıda bırakılmıştır.

* Sonuç

Bu çalışmada hedeflenen kablosuz veri aktarımı wifi sensöründe yaşanılan problemlerden dolayı istenildiği şekilde sonuçlanamamıştır. Dolayısıyla lidar sensöründen elde edilen veriler anlamlandırılarak cloud yerine yine uart üzerinden terminale basılmıştır. Bu sebeple çalışma tam olarak başarıya ulaşmamıştır.

References

1. SJ-PM-TF02-Pro A01 Product Manual, Benewake (Beijing) Co., Ltd.
2. <https://en.wikipedia.org/wiki/Hayes_command_set>
3. <https://github.com/esp8266/at-command-set/blob/master/commands.txt>
4. <https://devnot.com/2017/mqtt-nedir-nasil-bir-mimaride-calisir/>
5. <https://www.hivemq.com/blog/mqtt-essentials-part-5-mqtt-topics-best-practices/>
6. <https://mqtt.org/>
7. <https://www.geeksforgeeks.org/difference-between-mqtt-and-http-protocols/>