Bu projede kapı kilit uygulaması yapılmıştır. Aynı zamanda odaya giren ve çıkan kişilerin sayıları bulunup grafiği çıkartılmıştır. Proje, kapısını kilitleyip güvende hissetme ihtiyacından yola çıkılıp aynı zamanda odaya kaç kişinin girip çıktığını ortaya koymak için gerçekleştirilmiştir. Sunucu üzerinden Wİ-Fİ ile bağlantı sağlanıp kapı kontrolü yapılabiliyor. Projede, ESP-8266 kullanılmış olup kapı kilidini kontrol etmek için servo motor, odaya giren çıkan kişi sayısını bulmak için ise HC-SR501 PIR adlı hareket sensörleri kullanılıp yazılımsal kısmı C++ programlama dili ile gerçekleştirilmiştir.

* ESP-8266

**ESP8266** Wifi modülü sayesinde internete kolay bir şekilde bağlanabiliyor.

Bu özelliği sayesinde uzaktan kontrol ve  **IOT** projelerinde çok fazla kullanılır. Ayrıca düşük güç tükettiği için, güç tüketimi önemli olan projelerde de çok tercih edilir.

Projemizde servo motor kontrolü ve hareket sensörlerin çalışması için kullanılmıştır.

* HC-SR501 PIR SENSÖRÜ

PIR sensörü, bir ortamda oluşan hareketleri algılamak için kullanılmaktadır. PIR sensörü görüş alanındaki enerji seviyelerinde meydana gelen fark edilebilir artışları ölçer. PIR hareket sensörü içerisinde bulunduğu çevreyi ölçen kızılötesi duyarlı materyaller bulunur. Sensörün algı alanından sıcak bir cisim geçtiğinde pozitif ve negatif farklılıklar sensör tarafından algılanır. Çevredeki objelerin ısı haritasında kayda değer bir değişiklik meydana geldiğinde sensör kullanıldığı sisteme bir uyarı göndererek ortamda değişiklik olduğunu belirtir.

Projemizde pır sensörü odaya giren çıkan kişi sayısını bulmak için kullanılmıştır.

* SERVO MOTOR

Servo Motor, mekanizmalardaki açısal-doğrusal pozisyon, hız ve ivme kontrolünü hatasız bir şekilde yapan tahrik sistemi olarak tanımlanır. Yani hareket kontrolü yapılan bir düzenektir.

Projemizde web sunucu üzerinden kontrolü sağlanan kapı kilit düzeneği  olarak kullanılmıştır.

* Breadboard

Breadboard elektronik devrelerin yarı kalıcı prototiplerini oluşturmak için kullanılan bir yapı temelidir. Breadboard iç yapısı dik ve yatay şekilde birbirlerine bağlı halde konumlanmış metal kıskaçlardan oluşur.

Projemizde servo motor, pır sensörü ve harici güç bağlantısı yapılmak için kullanılmıştır.

* FTDI FT232RL

FTDI (Future Technology Devices International) bir İskoç şirketidir. Şirketin FT232RL adlı ürünü, üzerinde 3.3V veya 5V güç veren VCC pini, GND pini ve seri haberleşmeyi sağlayan RX ve TX bulundurur.

Proje kapsamında FTDI F232RL, pir sensörüne güç vermek için kullanılmıştır.

* Jumper Kabloları

Jumper kabloları devre tasarlamak için kullanılan, tasarım aşamasında lehim gerektirmeyen elektrik kablolarıdır. Giriş ve çıkış başlıklarına göre erkek/erkek, erkek/dişi ve dişi/dişi şeklinde isimlendirilir.

Proje kapsamında Jumper kabloları erkek/dişi olarak haberleşme, güç ve toprak bağlantıları için kullanılmıştır.

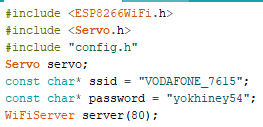
PROJENİN GERÇEKLEŞTİRİLMESİ

Projeye ilk olarak kablo bağlantıları ile başlanmıştır. Servo Motorun GND ucu ESP8266 ‘ nın GND ucuna, VCC ucu 3.3V ucuna ve son olarak dijital pin olarak D1 ucuna bağlanmıştır. Sonrasında ise hareket sensörünün kapıya giriş olarak kodlanacak olanının OUTPUT çıkışı D4 dijital pinine, GND ucu ESP8266’ nın GND ucuna ve VCC ucu ise 3.3V ucuna bağlanmıştır. En son olarak çıkış olarak kodlanacak olan diğer hareket sensörümüzün OUTPUT çıkışını D7 dijital pinine ve güç yetersizliğinden dolayı GND ve VCC çıkışlarını FTDI FT232RL isimli harici güç kartına bağlanmıştır.

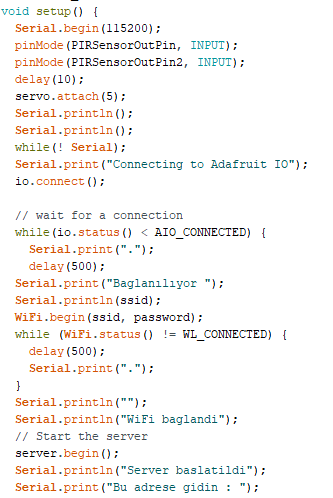
Bu bağlantılar beardboard ile sağlanmıştır.

YAZILIM KISMI

Kodlamaya önce tanımlamalarımızdan ve servo motor kontrölünden başlıyoruz. Servo motor kontrölü için #include <Servo.h> kütüphanesini projemize dahil edip Servo servo adında bir nesne oluşturuyoruz. İşlemlerimizi bu nesne ile sağlayacağımız. Sonrasında setup fonksiyonu içine servo.attach(5); olarak tanımlıyoruz çünkü D1 çıkışı 5. Pine denk gelmektedir. Sonrasında servo motorumuzu yazacağımız html sayfası ile kontrol ediceğimizden dolayı gerekli internet bağlantıları için kütüphanelerimizi dahil edip gerekli ayarlamaları aşağıda görüldüğü üzere yapıyoruz. Ssid kısmına Wifi adımızı password kısmına isme bağlı olduğumuz Wifi şifresini giriyoruz. En son olarak 80. Port üzerinden bağlantı sağlayacağımızı tanımlıyoruz.

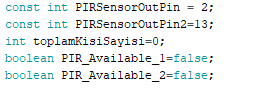


Setup Kısmı :

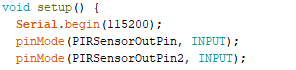




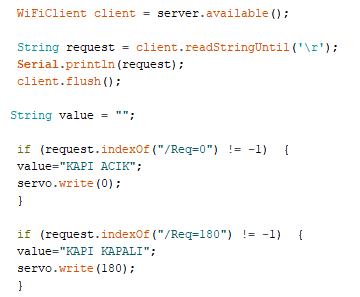
Burada ise hangi IP adresiyle bağlantı sağlanacağını kullanıcıya iletiyoruz. Aynı zamanda loop() fonksiyonumuza başlamadan hareket sensörlerimiz için de ayarlamalarımızı yapıyoruz.



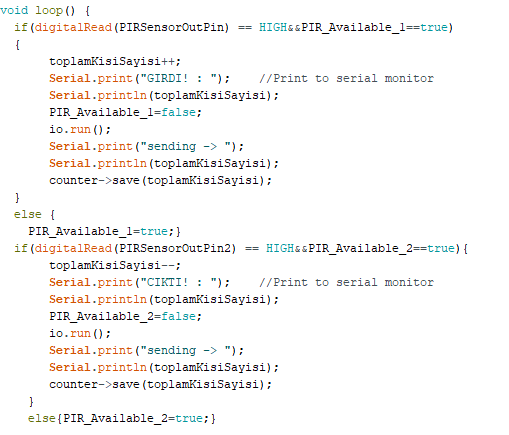
Yukarıda göründüğü üzere hareket senörlerimizi sırasıyla D4 ve D7 çıkışlara bağladığımızdan tanımlamasını yaparken 2 ve 13. Pinler olarak ayarlıyoruz. Odadaki toplam kişi sayısını atamak için toplamKisiSayisi adında bir değişken oluşturup 0 değerini atıyoruz. Ve ileride kullanmak üzere iki adet boolean değişkeni oluşturuyoruz.



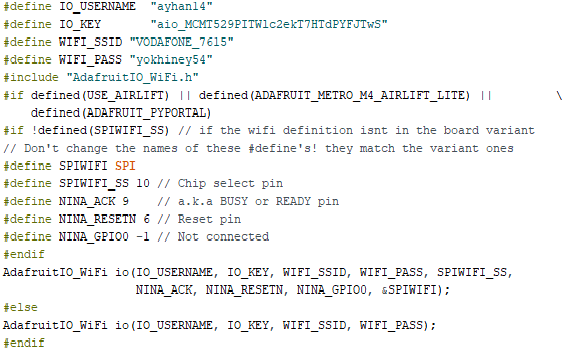
Setup kısmında pinMode() fonksiyonunu kullanarak INPUT olarak hareket sensörlerimizi tanımlıyoruz.



Bir client nesnesi oluşturup tanımladığımız serverın kullanılabilir olduğunu belirtiyoruz sonrasında ise sayfamızdan gelen isteğin baş kısmını atıp sondaki req=0 veya req=180 kısmını bir değişken oluşturup atıyoruz. Bunu yapma sebebimiz sadece bir request gelince servo motorun çalışmasını istememizdir. Mesela bir slider bar oluşturup programa sürekli slider bar değerini kontrol et deseydik aşırı yükleme yapacaktık gelen isteklere göre hareket et demek daha performanslı bir yazılım oluşturmamıza olanak sağlamıştır. Sonrasında ise oluşturduğumuz İF blokları ile isteğimiz /Req=0 ise servo motoru 0 derecesine getir tanımladığımız value değerini kapı açık olarak atayıp kullanıcı bilgilendir, /Req=180 ise servo motoru 180 derecesine getir tanımladığımız value değerini ise kapı kapalı olarak atayıp kullanıcıyı bilgilendir şeklinde işlemektedir.



Programımızın bu kısmında ise hareket sensörlerimizi digitalRead fonksiyonu ile tanımlayıp bir hareket algıladıysa if bloğuna girmesini tanımlamışızdır. Eğer giriş olduysa toplamKisiSayisi değişkenini 1 arttırıp kullanıcıyı bilgilendirmek için ekrana ‘GİRDİ! : kişi sayısı’ şeklinde yazdırmışızdır. Ve en başta tanımladığımız PIR\_AVAİLABLE\_1 değişkenini false olarak tanımlıyoruz bunun nedeni ise hareket sensörü bir kişi gördüğünde bunu daha fazla kişi olarak algılamasın diyedir yani sensöre eğer hareket algıladıysan bir say ve hareketin bitmesini bekle demiş oluyoruz. Aynı şekilde çıkış sensörümüz için de bu işlemleri ayıp ayrı olarak kişi sayımızı arttırmak yerine çıkış olduğu için azaltıyoruz. Son olarak html sayfamızdan önce adafruit ile grafik oluşturma işlemlerimizi yapıyoruz.

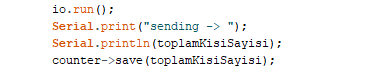




Config.h isimli bir proje oluşturup gerekli kodları yazıyoruz. Sonrasında bu projeyi ana projemize include ediyoruz.

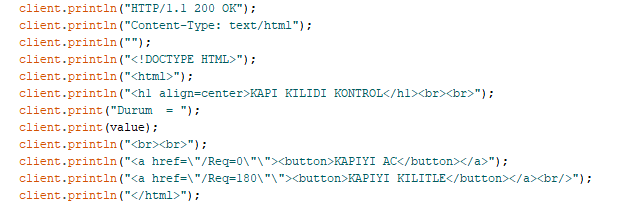


setup() fonksiyonumuz içine bağlantı için kodlarımızı yazıyoruz.

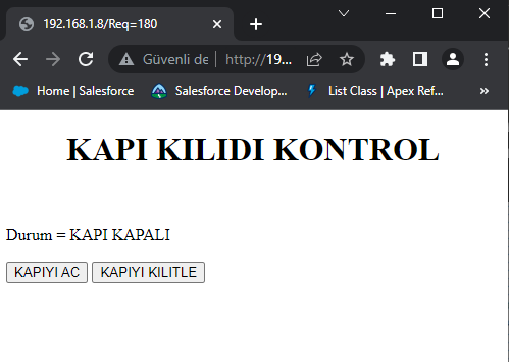


Hızlı olması açısından oluşturduğumuz IF blokları içerisine grafik fonksiyonumuzu çağırıp veri gönderimi işlemini kullanıcıyı bilgilendirme için ekrana yazdırarak gönderiyoruz. Böylelikle her giriş ve çıkışta IF bloklarımız çalışacak grafiğe veri gönderilecektir.

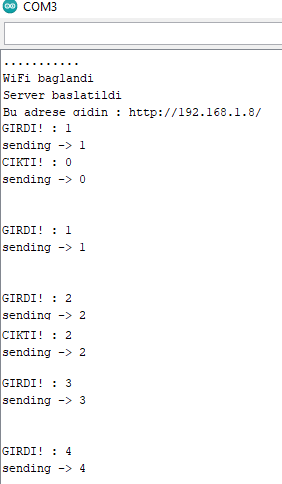
En son olarak html sayfamızı kodluyoruz. Bunun için oluşturduğumuz client nesnemizin .println() fonksiyonunu kullanacağız.



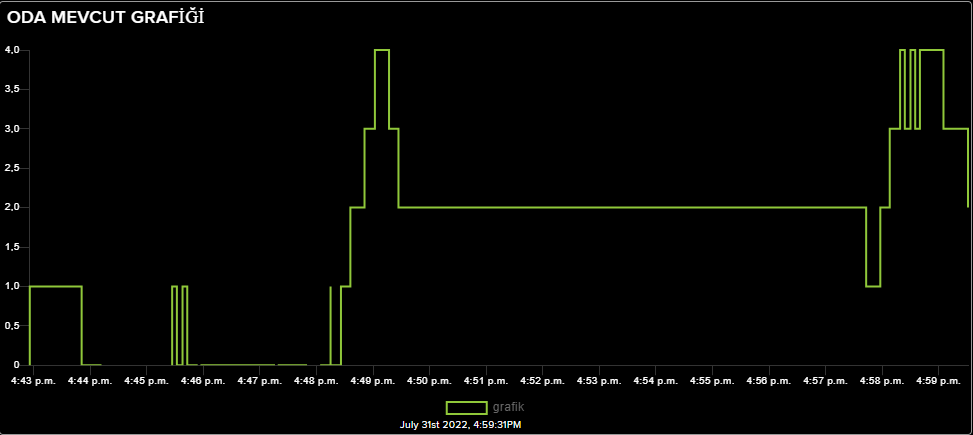
Html sayfamız şu şekilde görüntülenecektir.











SONUÇ

Projemizde Servo Motor Kontrolü ile kapı kilitleme, hareket sensörü ile odaya giren çıkan kişi sayısını bulma işlemleri gerçekleştirilmiştir. Bu projenin amacı insanaların daha güvende hissetmesini sağlamak ve ortama giren kişi sayısını bulmaktır. Projemizin bir çok kullanım alanı olabilir. Örnek vermek gerekirse yakın zamanda geçirdiğimiz pandemi döneminde bazı ortamlardaki kişi sayısı sınırlandırılmıştı. Bu kişi sayısının sınırını korumak ve sağlamak için kolaylıkla kullanılabilir. LCD bir ekran bağlanıp kişi sayısı bu ekrana yazdıralarak proje daha da geliştirilebilir.

KAYNAKÇA

* <https://www.motorobit.com/blog/icerik/servo-motor-nasil-calisir-ve-arduino-ile-servo-motor-kontrolu-nasil-yapilir>
* <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>
* <https://diyi0t.com/hc-sr501-pir-motion-sensor-arduino-esp8266-esp32/>
* <https://io.adafruit.com/ayhan14/wippersnapper>