

Gömülü Sistem Rehberi

Gökhan DÖKMETAŞ

Gömülü Sistemler Üzerinde Çalışacak Herkes İçin Bir Rehber

© Bu elektronik kitabın her hakkı Gökhan DÖKMETAŞ'a aittir. Yazarın izni olmaksızın kısmen veya tamamen alıntı yapılamaz. Kaynak göstermek suretiyle yasal iktibas sınırları içerisinde alıntı yapılabilir. Elektronik kitap aslına sadık kalınmak suretiyle çoğaltılabilir, herhangi bir ticareti yapılmadan çıktısı alınabilir ve paylaşılabilir. Herhangi bir intihal durumunda yazar ile iletişime geçilmesi rica olunur. KİTAP ÜCRETSİZDİR.

Gökhan DÖKMETAŞ Kimdir?

Kendi firmamızda jeofizik ve arkeojeofizik ölçüm ve görüntüleme sistemlerinin ar-ge'sini yaparak işime devam etmekteyim. Aynı zamanda bu alanda eğitim içeriklerini hazırlamaktayım. Yazdığım basılı kitaplar Arduino Eğitim Kitabı ve Arduino ve Raspberry PI ile Nesnelerin İnterneti kitaplarıdır. Aynı zamanda www.lojikprob.com adlı internet sayfasında gömülü sistemler için eğitim içerikleri hazırlamaktayım. C ile AVR Programlama ve Mühendisler İçin Arduino Kaynak Kodu İncelemesi elektronik kitaplarından sonra üçüncü kitap olarak bu kitabı yayınlama kararı aldım.

Benimle iletişime geçmek için aşağıdaki e-posta adresini veya facebook profil adresini kullanabilirsiniz.

E-Posta : gokhandokmetas0@gmail.com

Facebook: <https://www.facebook.com/lojikprob>

Ayrıca bir facebook grubumu oluşturduk ve üst seviye bir ortamda gömülü sistemler hakkında bilgi alışverişi yapıyoruz. Sizi de bekleriz.

<https://www.facebook.com/groups/lojikprob/>

Gömülü Sistemlere Başlamak İçin Neler Gerek?

Lojikprob açıldığından beri belli konularda yazı yazdık fakat gömülü sistemlere ait unsurlar hakkında pek çok yazımız olsa da başlangıca yönelik gömülü sistemleri genel olarak ele alan yazıları yazmayı ihmal ettiğimizi söyleyebiliriz. Bı eksikliği gidermek için “Gömülü Sistem Rehberi” adı altında yol gösterici yazıları yazmaya başlıyoruz. İlk yazımızın konusu ise gömülü sistemler üzerinde çalışmak isteyen herkesin nasıl başlayacağı hususunda olacaktır.

Öncelikle mühendis olup olmamanız fark etmez. Ben şahsen mühendis değilim ve piyasanın ihtiyaçlarına göre kendimi geliştirmiyorum. Malumunuz elektrik-elektronik mühendislerinin neredeyse %90'ı elektriğe yönelmekte ve elektroniği öğrenmeye gerek bile duymamaktadır. Her ne kadar onbinlerce elektrik-elektronik ve bilgisayar mühendisliği okuyan öğrenci olsa da bunların pek azı gömülü sistemlere ilgi duymaktadır. Bunda iş imkanlarının kısıtlılığı, sektörün küçük olması ve en önemlisi zor bir alan olması sebepler arasında sayılabilir. Ayrıca gömülü sistemlere ilgi duyan tekniker, esnaf, teknisyen ve amatörleri de görmekteyiz. Gömülü sistemler üzerinde iş yapmak için illa mühendis olmanıza gerek yoktur. Ben jeofizik ölçüm sistemlerini tasarlama ve geliştirme alanında kendi işimde çalıştığım için bu alanda kendimi geliştirmekteyim. Bu konuda çalışabilmek için elektronik, fizik, gömülü sistemler ve yazılımda bir hayli ileri gitmek gerektiği için sürekli yeni birşeyler öğrenmek zorundayım. Bu yüzden bu rehber yazısını da dersleri geçmek, iş bulmak değil sürekli öğrenmek ve yeni teknolojileri üretmek zorunda olan birinin bakış açısıyla yazacağım.

Gereksinimler

Gömülü sistemlere başlamak isteyenler öncelikle belli gereksinimleri belli bir seviyede karşılamak zorundasınız. Eğer bu gereksinimleri başta veya başlangıçta karşılayamazsanız bu alanda ilerlemeniz yavaşlayabilir, durabilir veya mümkün olmayabilir. O yüzden büyük dikkatle öncelikle bu gereksinimleri öğrenmeniz gereklidir. Bütün gereksinimleri tek tek yazma imkanımız yok fakat bize göre ve aklımıza gelen belli başlı gereksinimleri size açıklayacağız.

Dirayet

Sokaktan geçen herkesin gömülü sistem geliştiricisi olması beklenemez. Belli bir dirayet ve alt yapı şarttır. Bu konuda kimseyi küçümsediğimizi iddia edemeyiz fakat gömülü sistemler ressamlık, futbol, bisiklet sürme gibi herkesin yapabileceği basit bir alan değildir. Hatta üniversitelerde 4 yıllık gömülü sistemler bölümlerinin bile olmadığını görmekteyiz. Bilgisayar ve elektronik mühendisliğinin ortak alt dallarından biri olan gömülü sistemlerin belli bir temel üzerine bina edilmesi şarttır. Bu temeller üzerine bina edilmiş bir konuda da temelsiz birinin ilerleyememesinden doğal bir şey yoktur. O yüzden gömülü sistemlere başlayacak herkesin başlamadan önce veya başlarken bu temeli edinmesi gereklidir. Temelin yanında bu konuya yatkınlık da şarttır. Eğer bu konuda yeteneğiniz olduğunu hissetmiyorsanız kendinizi zorlamanın bir anlamı yoktur çünkü sektör çok para kazanabileceğiniz bir sektör değildir. Yukarı dediğimiz gibi bu yüzden çoğu elektronik mühendisi elektriği seçmektedir. Örneğin, bazıları bir yılda Arduino fonksiyonlarını çözemez bazıları ise bir ayda Arduino'yu öğrenir. Burada kişinin dirayeti yani ince meseleleri kavrama yeteneği büyük fark ettirmektedir. Öncelikle bu alan için yeteneğinizin ve yatkınlığınızın olduğunu keşfetmeniz gereklidir.

Merak

Dirayetten daha önemlisi ise meraktır. Merakınız yoksa boş yere bu alana yönelmeyin. Zor, uğraştırıcı ve pek büyük sektörü olmayan bir alanda merakınız olmadan çalışmak size işkence gibi gelecektir. Bu alanda her zaman öğrenci gibi olmanız ve sürekli yeni şeyleri öğrenmekle beraber gelişen teknolojiyi takip edebilmeniz gereklidir. Çoğu zaman uykunuzdan, tatilinizden, televizyondan ve sair keyiflerden ödün vermeniz gereklidir. Ayrıca merakınız olmadan yazmaçları ve bitleri öğrenmek, işlemci komutlarını anlamak veya boolean aritmetiğinin öğrenmenin bir keyfini alamazsınız. Ancak merak sayesinde bunları öğrenme imkânınız vardır. Unutmayın, çoğu meseleyi dersi geçmek için öğrenmek zorunda kalmayacaksınız. Mühendislik fakültelerinde Arduino bile seçmeli ders olduğu için bu alan hakkında öğreneceğiniz bilgilerin çoğu sizin şahsi çalışmalarınızdan ibaret olacaktır. Mühendis olmanın avantajı ise temel teorik bilgilere ve matematiğe aşina olmaktan ibarettir. Bu alanda ciddi bir eğitimin olmaması aslında meraklı insanların yararına bir durumdur. Çünkü dersi geçmek zorunda olmak bütün merakınızı öldürür ve sizi müfredata hapseder. O yüzden başlamadan önce gerçekten meraklı mıyım yoksa bir anlık heves mi diye kendinizi sorgulayın.

İngilizce

Eğer basit İngilizce cümleleri kuramıyor, kelimeleri anlayamıyor iseniz gömülü sistemlere başlamayın. Lisede “almanız gereken” yani B1-B2 seviyesinde bir İngilizceniz varsa bile başlangıç için yeterli olacaktır. Sonra zamanla İngilizce kaynakları ve dokümanları okudukça bunu geliştirme imkanı bulursunuz. Fakat A1 seviyesinde ya da bunu bile karşılamayacak kadar İngilizce bilgisiyle mezun olan mühendisleri görmekteyiz. Bu mühendisler kendilerini nasıl geliştirebilecektir? Mühendisliği bırakıp birkaç yıl dil öğrenmeleri gerektir artık. Gömülü sistemlerde ilerlerken aynı zamanda İngilizcenizi de ilerletmeniz mümkündür. Fakat dediğim gibi temel bir İngilizce bilgisi şarttır. Bunu ölçmek için en iyi kriterlerden biri İngilizce Wikipedi makaleleridir. Eğer İngilizce Wikipedi makalelerinin çoğunu anlayabiliyorsanız yeterli bir İngilizceye sahipsiniz demektir. Eğer İngilizce blog veya haber yazılarını basit dilde olması kaydıyla anlıyorsanız İngilizceyi geliştirme üzerine düşerek bu alana giriş yapabilirsiniz. Belli bir seviyede İngilizceden sonra İngilizceniz kendiliğinden gelişmeye başlayacaktır fakat hiç bilmeyen birisine gömülü sistemler üzerinde çalışmak İngilizce’yi öğretmez. Türkçe kaynaklardan da faydalanabilirim diyen biri belki yazılmasını istediği bir blog yazısı için yıllarca beklemesi gerekecektir. O blog yazılarının çoğu da İngilizce blog yazılarının ucuz tercümelemeleri olmaktadır. Hiçbir zaman İngilizce kaynaklardaki ayrıntıyı ve derinliği Türkçe kaynaklarda göremezsiniz. Bu site bir istisna olsa da genel olarak Türkçe kitapların bile İngilizce kaynaklardan fersah fersah geride olduğunu görüyoruz. Biz başlangıçtakilerin zorluk çekmemesi için bu siteyi hazırlasak da bu süreçte İngilizcenizi ilerletmeniz şarttır.

Temel Bilgiler

Temel bilgiler dediğimizde analog elektronik, dijital elektronik ve bilgisayar mimarisi konuları başlıca konulardır. Sizin çok derin fizik ve matematik bilgisine başlangıçta ihtiyacınız olmasa da konuyu anlayabilmek için dijital elektronik ve bilgisayar mimarisini iyi bilmeniz gereklidir. Gömülü sistemleri öğrenmeye çalışırken karşınıza çıkan engellerin çoğu sizin temel bilgidен yoksun olmanızdan kaynaklanır. Örneğin ben STM32 öğrenirken o güne kadar idare eder seviyede olan dijital elektronik bilgimle datasheetteki terimleri ve kavramları rahat rahat anlayamadığımı farkına vardım. 8-bit mikrodenetleyicilerde çok sıkıntı olmasa da 32-bit mikrodenetleyicilerde DMA, CRC gibi kavramları görünce bunları rahatça anlamak için temel bilgiyi ilerletmek gerektiğini anladım ve Morris Mano, Anil K. Naimi, Mark Balch gibi yazarların dijital elektronik ve tasarım kitaplarını okudum. Bu kitapları okuma nedenim datasheet okurken bazı yerlerde anlamamın zorlaşmasıydı, anlayamamak bile değil. Bu kitapları okuduktan sonra tekrar

döndüğümde ise meselelerin ne kadar kolay olduğunun farkına vardım. Siz datasheet okuyacak derecede iyi bir İngilizce bilerseniz bile temel bilgi olmadan o datasheet size işkence gibi gelecektir. Ana dili İngilizce olan birinin önüne datasheet koysanız o da anlamayacaktır. Çok iyi İngilizce bilerseniz dahi temel bilgileri ya öğrenmeden önce ya da öğrenirken öğrenmeniz gereklidir.

C Programlama Dili

Ben burada tek bir programlama dilini öğrenin demiyorum fakat C dilini bilmezseniz neredeyse hiçbirşey yapamazsınız ve ilerleyemezsiniz. C dili olmadan Assembly diliyle program yazılabilir de ana akım C dilini kullanmaktadır ve bir noktadan sonra gelişmiş programları yazmak için C diline ihtiyaç duyacaksınız. C dilini temel olarak bilerseniz dahi gömülü sistemlerde kullanımına (Gömülü C) aşına olmanız da gereklidir. Örneğin bitwise operatörler, işaretçiler veya sabit tamsayı tiplerini temel C programlamada pek sık göremezsiniz. C programlama dilini iyi bilmek sadece dile ait döngü, karar, ön işlemci yönergesi, kütüphane fonksiyonları gibi yapıları bilmek değil aynı zamanda bunları etkin bir biçimde kullanabilmek demektir. Bunun için sizin algoritma ve programcılık yeteneğini de öğrenip kazanmanız gereklidir. C programlama dilini anlayarak kullanmadıkça gömülü sistemlerde program yazmanız söz konusu olamaz. Çünkü çoğu programı sıfırdan yazdığınız gibi kütüphaneleri de sıfırdan yazmanız gerekecektir. Bir yazmadığınız şeyin derleyici olması gereklidir. Hazır kodları kopyala-yapıştır ile baka baka yapmak ile gömülü sistem geliştireciliği olmaz.

Pratik Elektronik

Sizin Arduino kartını kullanıp breadboard üzerinde iki kablo bağlamakla kurduğunuz devre bile size bir pratik kazandıracaktır. Fakat pratik konusunda bununla yetinmemek gereklidir. Tam kapsamlı bir elektronik cihaz üretebilmek ve bunun bütün aşamalarındaki pratiği kavramak gerekir. Örneğin PCB tasarımı, PCB yapımı, lehim ve dizgi, test ve kontrol, kutulama, panel tasarımı, arayüz üretimi, etiket basımı gibi konularda bile pratiği kavramanız gereklidir. Siz bir cihaz yapmak istiyorsanız bilginiz ve yazılımınız ne kadar iyi olsa da işçilik ve malzeme kalitesi iyi olmadıkça o cihazdan pek hayır gelmeyecektir. Sizin bir atölye kurmaya yavaştan başlamanız gereklidir ve bu konuda kendi cebinizden ciddi bir meblağ ayırmanız gerekebilir. Osiloskop, lojik analizör, iyi bir havya, lehim teli, lehim pastası derken maliyet belki binlerce lirayı bulabilir. Arduino kullanıcıları bu konuda pratik yaptığını sanmakta ama yaptıkları pratik hep birbirini

tekrarlayan aşamalardan ibaret olmaktadır. Breadboard üzerine yeni bir eleman koyup aralara jumper kablosu takmak pratiğin ilk aşamasıdır. Sizin bu pratiği sürekli ilerletmeniz gereklidir.

Bilgisayar Okur-Yazarlığı

Pek çoğunun gözünden kaçan bir konu olan bilgisayar okuryazarlığı aslında en önemli konulardan biridir. Bilgisayar kullanmayı bilmemekten dolayı pek çok geliştirici aday ve mühendis zorluk çekmektedir. Bir mühendisin bilgisayar kullanmayı bilmemesi pek ala mümkündür. Örneğin Aygıt yöneticisini açıp bir cihazı kontrol edemiyorsa, sürücü yükleyemiyorsa, ekran görüntüsü almak yerine telefonun kamerasını yamuk şekilde tutup ekranın fotoğrafını çekiyorsa bilgisayar kullanmayı bilmiyor demektir. Ayrıca bilgisayar okuryazarlığı sadece bilgisayarı ve işletim sistemini kullanmayı değil interneti kullanmayı da kapsamaktadır. Google'dan arama yapmasını bilmeyen, yer imlerini kullanamayan, neyi nerede bulacağını bilmeyen birisi gömülü sistem geliştiriciliğinde büyük zorluk çekecektir. Bilgisayarla ilk defa tanışıyorsanız etkin bir biçimde bilgisayar nasıl kullanılır onu öğrenmekle işe başlayın. Bir derleyici programı yüklemek için bile "Next" tuşuna basılması gerektiğini öğrenmeye muhtaçsanız ciddi bir sorun vardır. Hemen hemen bütün işlerinizi bilgisayarda yaptığınız için bilgisayarı etkin bir biçimde kullanamamak kabul edilemez.

Bu yazıda gömülü sistemlere başlamak isteyenlerin karşıması gereken gereksinimlerden bahsettim. Bunların hepsini karşılamadan gömülü sistemlere bulaşmayın demiyorum ama bütün bunların başlangıç seviyesinde karşılanması şarttır. Arduino ile birkaç proje yapayım, çizgi izleyen robot yapayım bana yeter diyenleri muhatap almadığımızı bu noktaya kadar anlamış olmanız gereklidir. Biz bu işte gerçekten ilerlemek, yeni teknoloji ortaya koymak ve sayılı kişilerden biri olmak isteyenlere yönelik bir rehber hazırlamaktayız.

Gömülü Sistemlere Kimler Başlamalı?

Gömülü sistemlere nereden başlamalı ve kimlerin başlaması gerektiği bir önceki yazıda bahsettiğim gereksinimleri ne kadar karşıladığı ile doğrudan ilgilidir. Eğer bu gereksinimleri yeterince karşılamıyorsanız gömülü sistemlere başlarken bunların boşluğunu da eş zamanlı olarak doldurmanız gerekecek ve sizin için daha zor bir süreç olacaktır. Eğer saydığımız gereksinimlerde sıfıra yakın iseniz gömülü sistemlere başlamakla bırakma

arasında pek fazla zaman harcamazsınız. Çünkü ne kadar çalışsanız da bu gereksinimleri karşılayamadığınız için ilerleyemezsiniz ve bırakmaktan başka bir çareniz kalmaz. Biz öğrenebilecek kimsenin yanlış usul ve yönlendirmelerden dolayı öğrenemeyip bırakmasını hiç istemeyiz. Fakat öğrenmeyecek kişiye de boşuna emek harcamak istemiyoruz. Bu yazıları yazarken sarf ettiğimiz onca emek, öğrenebilecek bir avuç kişi içindir. Ne kadar on binlerce elektrik-elektronik ve bilgisayar okuyan olsa da çoğu bizim yazdıklarımızla ilgilenmemektedir. Hatta bu sektörde bile “konuyla alakalı” görünen fakat konuyla alakasız kitleler oldukça fazladır. O yüzden gömülü sistemlere gerçekten giriş yapabilen biri olursanız seçkin bir konumda olduğunuzu biliniz.

Önceki yazıda dediğimiz gibi çizgi izleyen robot yapmak için, bitirme projesini yapmak için veya internette gördüğü hazır projeyi yapmak amacıyla bu alana girip öğrenmek isteyen kişileri muhatap almadığımızı, bizim muhatap aldığımız kişilerin gerçekten öğrenmek isteyen, hevesli değil meraklı ve vizyonlu kitleler olduğunu hatırlatalım.

Gömülü sistemlere başlamadan önce başlama sebebinizi sorgulamanız gereklidir. Örneğin sizin bir ihtiyacınız var. Bu kuluçka makinesi veya buzdolabı ısı kontrolü, zaman ayarlı sulama sistemi gibi başka bir sektöre yönelik bir ihtiyaç olsun. Siz sırf bu alandaki geliştiricilere para verip yaptırmamak için kendiniz öğrenip sadece onu yapmaya çalışıyorsanız öğrenme sürecindeki maliyet ve zamanın bunun onlarca katı olacağını ve konuyla alakalı bir insan olmadıkça bir faydanızın olmayacağını biliniz. Aynı şekilde sırf bitirme projesini yapmak için istemediğiniz ve alakanızın olmayacağı bir alanda aylar boyu çalışmanız da sizi gömülü sistem geliştiricisi veya mühendisi yapmayacaktır. Çoğu da bu alanın zorluğunun farkına varıp bu işlere hiç bulaşmadan bitirme projesini yaptıracak birini aramaktadır. Biz burada gömülü sistemler hakkında yüzlerce makale yazmış olsak da bunları okuma zahmetini bırakın bizi tanıma zahmetine bile girmeyip hala bitirme projelerini yaptıрма konusunda bizi rahatsız edenleri görmekteyiz.

Eğer çocukluktan beri teknolojiye, bilgisayarlara ve elektronik cihazlara meraklıysanız, bir şey üretmek ya da icat ortaya koymak istiyorsanız gömülü sistemler tam size göre bir alandır. Çünkü bu alanda var olanın

üzerine bir şey koymadıkça, yeni bir şeyler üretmedikçe, aklınızı kullanmadıkça iş yapmanız söz konusu değildir. Hazır parçaları birbirine takıp hazır kodu yükleyerek örneğin bir 3 boyutlu yazıcı yapmak bu alanda “yapmak” sayılmamaktadır. Bu gömülü sistemlerde geliştiricilik değil sadece işçilikten ibarettir. Aynı şekilde hazır Arduino projelerini iki kablo bağlamak ve iki tuşa basmakla yapmanın gömülü sistem geliştiriciliği ile bir alakası yoktur. Siz böyle hazır projeleri yapmakla eğlenebilirsiniz, boş zamanınızı değerlendirirsiniz fakat gömülü sistemlere giriş yapmak isteyen birinin oldukça disiplinli ve planlı çalışması gereklidir. Bu öğrenme süreci sizin ikinci işiniz veya okulunuz olmalıdır. Bu alanda 1-2 hafta ara vermeniz bile alana yabancılaşmanıza sebep olacaktır. Diğer programlama ve geliştiricilik konularında olduğu gibi sürekli bir çalışma ve devam eden bir merak ihtiyacı vardır. Bir anlık hevesle başlayıp hevesten sonra devamını getirmek mümkün değildir. Eğer meraklı biriyseniz öğrendikçe merakınız artacak ve ne kadar az şey bildiğinizin farkına varacaksınız. Gömülü sistemler üzerinde çalışarak bilgisayar ve elektrik-elektronik mühendisliğinin konuları sizi ilgilendirirken örneğin bizim gibi jeofizik üzerinde çalışmak isterseniz jeofizik mühendisliğinin, arkeojeofizikte buna ek olarak arkeolojinin de konularını da öğrenmeniz gerekecektir.

Jeofizikten bahsetmişken önemli bir noktayı size başlarda anlatalım. Sadece gömülü sistemler alanında uzman olmanız bir ürün ortaya koymanız için her zaman yeterli değildir. Başlangıçta kontrol ve otomasyon uygulamalarında bu şekilde ilerleseniz de gömülü sistemler ülkemizde görüldüğü gibi kontrol ve otomasyondan ibaret değildir. Örneğin eşyaların interneti alanında çalışmak istiyorsanız ağ teorisini, protokolleri, sunucu-istemci ve en önemlisi HTML, PHP gibi Web ile alakalı dilleri bilmeniz gereklidir. Web programcılığı apayrı bir dal olsa da eşyaların interneti (IoT) bunun üzerine bina edilmiştir ve iyi bir web programcısı olmayı da gerektirir. Özellikle arka plan (back-end) yani PHP gibi programlama dillerine aşina olmanız gereklidir. Eğer bilgisayar kontrollü uygulamalar yapmak istiyorsanız seri iletişim alanınızla ilgili olsa da masaüstü programlamak gömülü sistemlerin dışındadır. Bu durumda Visual C#, Java, Visual C++ gibi programlama dilleri ve ortamlarını öğrenmeniz ve bunun üzerinde program yazmanız gereklidir. Bazı cihazlar bilgisayar programı olmadan hiçbir işe yaramamaktadır. Bir cihazın bilgisayar programını, analog devre tasarımını

bir yandan da üretimini hallettikten sonra gömülü yazılımına sıra gelebilir. Üstelik bu gömülü yazılımı yapabilmek için de bazen dijital sinyal işleme yöntemlerini öğrenmek bile gerekir. Kısacası gömülü sistemler gibi dar görünen fakat pek çok alanla irtibatlı bir alanda uzmanlaşmak yeterli olmamaktadır aynı zamanda çalıştığınız alanla ilgili farklı konularda uzmanlaşmanız gerekebilir.

Üzerinde çalışma yapacağınız alanı ilgilendiren dallarda uzmanlaşmak da bir ürün ortaya koymanız için yeterli olmamaktadır. Ar-ge adını verdiğimiz araştırma ve geliştirme sürecinde o konulara hâkim olsanız dahi yeniden birşeyler öğrenmek, araştırmak ve çalışmak gereklidir. Bu ARGE süreci birkaç ay olabileceği gibi 5-10 yıl da olabilir. Yaptığınız işin katma değerine ve zorluğuna göre değişir. Kısa yoldan bir şeyler yapayım diyorsanız bu alan size göre değildir. Arduino ile 5 dakikada breadboard üzerinde yapılan projeleri görüp de “az bir bilgiyle” kısa zamanda katma değerli iş yapabileceğiniz algısı oluşmasın. Kısa zamanda iş yapabilmek için bile kısa zamanda iş yapabilmek için gerekli olan bilgiyi epey uzun bir zaman harcayarak edinmeniz gereklidir. Çalışmadığınız sürelerin öğrenmeye gideceğini unutmayınız. İki arduino fonksiyonu yazmakla “usta” olup iş yaptığını sananlar veya 100-200 liraya çocukların ödevlerini yapanlar gibi olmamak için bu şarttır. Sizin bu alanda kaliteli bir insan olduğunuz yaptığınız işin kalitesiyle belli olur.

Gömülü sistemlere başlamadan önce bütün bunları göze almış ve göğüs gerebilecek biri olduğunuzu düşünüyorsanız gömülü sistemlere başlamanız lazımdır. Eğer ben uğraşamam, emek harcayamam ya da öğrenemem diyorsanız kendinizi zorlamanın bir anlamı yoktur. Bir sonraki yazıda ise nasıl başlanmalı sorusuna cevap vereceğiz.

Gömülü Sistemlere Nereden Başlamalı?

Gömülü sistemlere nereden başlanacağı sizin ilgi ve çalışma alanınıza bağlı olduğu gibi sizin ön gereksinimleri ne kadar karşıladığınız ile de doğrudan alaka gösterir. Örneğin analog ve dijital elektronikte çok ileri seviye olan, C/C++ dilinde ileri seviye, mikroişlemci ve bilgisayar mimarisine oldukça hâkim birisi doğrudan doğruya 32-bit mikrodenetleyicilerle gömülü sistemlere başlayabilir. Fark ettiyseniz söylediğimiz kriterler ilk yazıda

bahsettiğimiz gereksinimler dâhilindeydi. Bu gereksinimleri ileri derecede karşılayan birisi için biz nereden başlayacağı konusunda tavsiye veremeyiz çünkü kendisi nereden başlayacağını az çok bilir. Fakat bahsettiğimiz koşulları sağlayan mühendisleri bile günümüzde pek görememekteyiz. Hatta bazı yeni mezun mühendisler bu konularda sıfıra yakın olmaktadır. Örneğin İngilizce birkaç kelimeyi bile anlamayan, C dilinden bihaber, dersi geçmek için ezberleyip unutan, okuduğu okulu anlamamış pek çok mezun mühendisleri görmek mümkündür. Hatta bazı geliştirici olmaya isteyenlere Türkçe öğrenmekle işe başlaması yönünde tavsiye de vermekteyiz. Çünkü daha Türkçe'yi bilmeden İngilizce öğrenmesi veya diğer konuları öğrenmesi çok zor olmaktadır. Bunu okuduğum dilbilim kitapları da tasdik etmektedir. Örneğin Almanya'da yaşayan Türk ailelerin çocuklarına yönelik Alman hükümeti Türkçe dersleri vermektedir. Çünkü bu çocuklar aile çevresinde Türkçe'yi yeteri kadar öğrenememekte ve bu yeteri kadar bilmedikleri anadille Almanca veya öteki dersleri de öğrenememekte ve okulda başarısız olmaktadır. Türkçeyi öğrenme konusundaki tavsiyemizin sebebi budur.

Gömülü sistemlere nereden başlanacağı konusunda daha önce saydığımız gereksinimleri temel/orta seviye karşılayanlar için bu yazımızı kaleme alalım. Dikkat edin, sıfır veya çok ileri seviye olana bir tavsiyede bulunmuyoruz. Bu konularda sıfır olana datasheet oku demekle ileri seviye olana Arduino ile başla demenin zararı büyüktür. Nereden başlanacağı konusu tamamen seviye ile alakalı bir durumdur.

Gömülü sistemlere başlamak isteyenler için son yıllarda bir takım olumlu gelişmeler ortaya çıkmıştır. Bunlardan birincisi hobi elektronik kitlesinin büyümesidir. Hobi elektronik kitlesi büyüse de bu büyümenin olumlu yönde olduğunu söylememiz biraz zor olur. Çünkü hobi elektronik ile uğraşanların eskiden yaptığı işlerle şimdi yaptığı işleri kıyaslayınca arada büyük bir uçurum görmekteyiz. Fakat biz her durumu kendi yararımıza kullanmayı öğrenmemiz gereklidir. Hobi elektronik kitlesinin büyümesi bu konudaki pazarın büyümesine sebep olmuş ve eğitimde kullanılacak malzemelere erişmemiz kolaylaşmıştır. İkinci nokta ise bilgi noktasındadır. Türkiye'de Arduino'nun bile yavaş yavaş öldüğünü görsek de yabancı ülkelerde hobi elektronik toplulukları oldukça canlıdır. Bu canlılık bu alanda yıllarca çalışan insanların sesini duyurmasına ve bilgilerini paylaşmasına imkân

tanımaktadır. Youtube, blog ve forumlarda bu tarz eğitim içeriklerinin artması hem hobiciler hem de gömülü sistemler üzerinde çalışmak isteyen amatör ya da profesyoneller açısından büyük bir imkândır.

Gömülü sistemler önceleri şimdiki halinden daha sessiz ve küçük bir sektördü. Son yıllarda Eşyaların İnterneti, ucuz mikrodenetleyiciler, gömülü linux gibi sistemler ortaya çıkıp bunlar eğitimlerde ve uygulamalarda kullanmaya başlandıkça bu sektör iyice büyümüş ve firmalar iyice rekabet içerisine girmiştir. Mikrodenetleyici pazarı yıllar geçtikçe büyük bir yükselişe büyümektedir. Aynı zamanda talep arttıkça mikrodenetleyiciler ve diğer gömülü sistem elemanları ucuzlamakta ve erişilebilir hale gelmektedir. Aliexpress gibi Çin siteleri de bu olumlu gelişmenin başka bir boyutudur. Biz belki onlarca dolar harcıyıp temin edebileceğimiz bir entegreyi birkaç dolara alabilir hale geldik. Ayrıca Adafruit, Sparkfun gibi firmalar her ne kadar maker pazarına yönelik ürün üretir gibi görünseler de gömülü sistemlerde geliştiricilerin işini kolaylaştıracak pek çok modülü üretmektedir. Elbette onların modüllerine onlarca dolar vermek istemeyiz, birkaç dolara aynı modülleri Aliexpress'de bulmamız mümkündür. Çin malı ürünlere bakmasanız bile firmalar büyük rekabete girdiği, eğitim ve hobi elektronik pazarına yöneldiği için bazı ürünleri oldukça uygun fiyatlara bulabilirsiniz. Önceden bir programlayıcı, bir deneme kartı 50-100 dolarlardan başlarken şimdi 5-10 dolara programcısı ile beraber orjinal geliştirme kartları piyasaya sürülmektedir.

Gömülü sistemlere giriş yapmak için öncelikle programlanabilir mantıksal sistemlerden birini tercih etmeli ve bir markadan başlamak gereklidir. Programlanabilir mantık içerisinde CPLD/FPGA gibi mantık bloklarından oluşan ve donanım tanımlama dili ile tasarım yapılan sistemlerden daha önce bahsetmiştik. İsterseniz Assembly dilini de C dilini de bir köşeye bırakıp mikrodenetleyici/mikroişlemci sistemlerine hiç bulaşmadan programlanabilir mantık üniteleri üzerinde geliştirme yapabilirsiniz. Mikrodenetleyici ile yapabileceğinizden daha fazlasını bu sistemler ile yapmanız mümkündür. Fakat bu sistemlere doğrudan geçiş yapmak için ciddi bir dijital elektronik ve dijital tasarım temelinizin olması şarttır. Zorluk bakımından en zoru da CPLD/FPGA sistemleri olduğunu söyleyebiliriz. Mikroişlemciler ile tasarım yapmak istediğinizde 6502, Z80, 8088 gibi antika mikroişlemcilerin bile halen yan sanayi olarak üretilip kullanıldığını

görmekteyiz. Bunun yanında günümüzde üretilen mikroişlemciler (CPU değil microprocessor) yüksek işlem kabiliyeti ve işletim sistemi üzerinde çalışmaya yönelik üretilmektedir. Bu tarz cihazları basit kontrol ve otomasyon uygulamaları için kullanmanız biraz abes kaçsa da yazılım boyutu ağır basan ve yüksek seviye işlem gerektiren uygulamalarda mikroişlemci üzerinden çalışmayı tercih edebilirsiniz. Günümüzde Raspberry PI gibi mikroişlemci tabanlı ARM çekirdekli ve Linux sistemli tek kart bilgisayarlar oldukça yaygınlaşmıştır.

En son olarak en yaygın ve en klasik yöntem olarak mikrodenetleyiciler ile gömülü sistemlere başlanmasından bahsedebiliriz. Bizim de temeli çok oturmamış ve gereksinimleri fazla karşılamayan yeni başlayanlara tavsiyemiz mikrodenetleyiciler yönünde olacaktır. Mikrodenetleyiciler sayesinde dijital elektronik, pratik elektronik, analog elektronik, mikroişlemci mimarisi ve C programlama konusunda hem bilgi hem tecrübe edinme imkânınız olacaktır. Yalnız ilk yazıda bahsettiğimiz gereksinimleri karşılamayan yani sıfıra biraz daha yakın olan kişiler için son yıllarda çıkan yeni bir eğitim platformu vardır. Bu platformun adı Arduino'dur.

Biz mühendisler Arduino öğrenmemelerini söylesek de mühendislik fakültelerinde Arduino'nun bile seçmeli ders olduğunu görmekteyiz. Üstelik mühendisler temel bilgilerden o kadar yoksun ki pek çoğu Arduino öğrenirken bile büyük zorluk çekiyor. Bizim sıfıra yakın yani sokaktan geçen insana yakın olan durumu pek çok mühendislik öğrencisinde şimdiye kadar gözlemledik. Ne yazık ki eğitim sistemi bu kadar kötü olunca Arduino'yu hakkıyla bilebilen mühendislik öğrencisini bile tebrik ediyoruz. Biz orta seviyeye yakın olanlara Arduino'yu tavsiye etmesek de temel bilgilerde sıkıntı yaşayan, temeli yeterince oturtamamış veya temel oturturken zaman kazanmak isteyen öğrencilerin Arduino ile başlaması gerektiğini savunuyoruz. Arduino eğitim platformu olarak iyi yönde kullanılabildikçe oldukça olumlu neticeler elde edilmektedir. Bunun için yazdığım "Arduino Eğitim Kitabı" bir Arduino eğitiminin nasıl olması gerektiğini gösterir niteliktedir. O yüzden Arduino ile gömülü sistemlere nasıl başlanır sorusunu atlayıp okuyucularımızı kitaba yönlendiriyoruz. Bir sonraki yazıda mikrodenetleyiciler ile gömülü sistemlere nasıl başlanır sorusunu sizlere bütün ayrıntısı ile açıklayacağız.

Mikrodenetleyiciler ile Gömülü Sistemlere Başlamak

Gömülü sistem rehberi adını koyduğumuz yazı dizisinin bu yazısında mikrodenetleyiciler ile gömülü sistemlere başlamayı ve bu aşamada karşılaşacağınız önemli noktaları sizlere açıklayacağız. Daha önceki yazımızda mikrodenetleyiciler ile gömülü sistemlere giriş yapmayı tavsiye etsek de ülkemizde kavramlar yerine tam oturmadığı için bu gömülü sistemlere giriş eğitimi "PIC Programlama" gibi adlarla adlandırılmaktadır. Üstelik eğitimler öyle kötü olmaktadır ki bir firmanın, bir mikrodenetleyici ailesine ait sadece bir mikrodenetleyici programlamak öğretilmekte, ne mikrodenetleyicinin mimarisi ne de geliştiricilik öğretilmektedir. Ülkemizde yıllardır PIC 16F84 anlatılmakta ve sadece bunu programlama doğrultusunda bir eğitim verilmektedir. Gömülü sistemler şu veya bu mikrodenetleyiciyi programlamaktan ibaret görülmemelidir. Gömülü sistemleri bundan ibaret görenler de geliştirici vizyonuna sahip olamamış demektir.

Mikrodenetleyiciler ile gömülü sistemlere giriş yapmak istiyorsanız karşınıza onlarca firma ve yüzlerce mikrodenetleyici ailesi çıkmaktadır. Önce hangisini seçeceğiniz büyük oranla size bağlı olmaktadır. Mikrodenetleyiciler 8 ve 32-bit olarak iki ana kategoriye ayrılrsa da bunların arasında pek yaygın olmayan fakat kendine ait bir kulvarı olan 16-bit mikrodenetleyiciler de bulunmaktadır. Üstelik 32-bit mikrodenetleyiciler de tek çeşit değildir. Genel olarak ARM çekirdekli olsa da MIPS çekirdekli PIC32ler ve adını pek duymadığımız farklı çekirdeklere ve mimarilere sahip mikrodenetleyiciler bulunmaktadır. Üstelik ARM çekirdekli mikrodenetleyicileri üreten her firma kendine özgü mikrodenetleyici tasarlamaktadır ve bu mikrodenetleyicileri programlayabilmek için de temel bilgide olmasa da çevre birimlerinde ve mikrodenetleyici mimarisinde ayrı ayrı bilgilere ihtiyacımız olmaktadır. O yüzden "ARM Programlama" tabiri yerine ARM çekirdekli üretilen bir mikrodenetleyicilerden birini örneğin, "STM32 Programlama" tabirini kullanmamız daha doğru olacaktır. Görüldüğü üzere 32-bit kendi içinde mimarilere ayrılmakta ve bu mimariler de markalara ayrılmaktadır. Gömülü sistemlere giriş yapmak için tek bir modelle başlayacak olsanız da öncelikle 32-bit mi 8-bit mi, hangi mimari ve

hangi marka sorularına cevap bulmanız gereklidir. Sonrasında ise başlamak istediğiniz modeli kararlaştırmanız lazımdır.

Önceki yazıda doğrudan 32-bit mikrodeneleyicilere başlayacak kişilerin daha önceden belirttiğimiz gereksinimleri ileri derece karşılması gerektiğini söylemiştik. Pek çok kişi için doğrudan 32-bit mikrodeneleyiciler ile başlamak hüsrarla sonuçlanır. Fakat herkes için doğrudan 32-bit ile başlamayın da diyemeyiz. Eğer bu alanda kendinize gerçekten güveniyorsanız 32-bit ile başlayabilirsiniz. 32-bit mikrodeneleyiciler ile gömülü sistemlere başlamak isteyenler için başta saydığımız gereksinimler içindeki birkaç özel noktayı belirtelim. Bu aynı zamanda 8-bit üzerinde çalışıp 32-bite geçmek isteyenler için de bir rehber olacaktır.

Datasheet Okuma ve Anlama

Eğer 32-bit mikrodeneleyicileri programlayacaksanız amatörler için yazılmış eğitim makalelerinden, örnek kodlardan, kütüphanelerden ve eğitim kitaplarından çoğunlukla mahrum olacağınızı biliniz. Örneğin PIC veya AVR hakkında onlarca kitap, video ve eğitim makaleleri yer alırken STM32 bunlardan daha popüler olmasına karşın hakkında bu kadar eğitim meteryali yer almamaktadır. STM32 programlayacak geliştiriciler datasheet, referans kılavuzu ve uygulama notlarını okuyup anlayacak derecede İngilizce'ye sahip olmalıdır. PIC için CCS C, Mikro C gibi hazır kütüphaneleri bulunduran derleyiciler, eğitim meteryalleri ve yardımcı kaynaklar datasheet okumadan uzun bir süre program yazabilmeniz size yardımcı olacaktır. Aynı durum Arduino için de geçerlidir.

İleri Seviye C bilgisi, 8-bit PIC veya AVR mikrodeneleyicileri programlarken bu kadar ileri seviye bir programlama bilgisine ihtiyacınız olmasa da STM32 gibi 32-bit mikrodeneleyicilerin hazır kütüphanelerini (HAL) kullanabilmek için bile ileri seviye bir C programlama bilgisine ihtiyaç vardır. Aynı zamanda kütüphane referansını okurken tanımları, yapıları ve fonksiyon prototiplerini okumakla anlammanız gerekecektir. Her fonksiyon için veya konu için örnek kod veya eğitim içeriği bulmanız beklenemez. Datasheetinin de referans kılavuzunun da 1000 sayfadan fazla olduğunu söyleyelim.

Mikrodenetleyici Mimarisi ve Dijital Elektronik

8-bit mikrodenetleyiciler üzerinde çalışırken temel seviyede bir dijital elektronikle ilerleyebilir ve çalıştıkça bu elektronik bilginizin üzerine katabilirsiniz. Üstelik bir süre sonra Assembly ile mikrodenetleyicinin mimarisini de öğrenme şansınız vardır. 32-bit mikrodenetleyicilerde ise pek çok kavram bir araya getirilmiştir ve bunlar hakkında bilgi edindiğiniz varsayılarak ilerlenir. Yani 32-bit mikrodenetleyicilerin temeli olmayana toleransı yoktur. Örneğin datasheette “DMA, CRC, open-drain, event, vector” gibi terimleri anlayabilmek için önceden temel bilgi edinmeniz şarttır.

İleri Seviye Yazılım Bilgisi

32-bitte yazılan yazılımlar 8-bitteki gibi basit olmamaktadır. Örneğin RTOS veya başka bir Middleware kullanmak istediğinizde bunun yazılımını çözebilmeniz gereklidir. Üstelik kullandığınız donanım birimleri de ileri seviye yazılım bilgisi ve programcılık yeteneği ile ilişkilidir. Birimler karmaşık olunca bunları kullanabilmek için de karmaşık programlar yazmak gerekecektir.

8-bit üzerinde çalışmayıp bütün bu temeli oturtmadan doğrudan 32-bit mikrodenetleyicilere geçtiğimizde kafanız çok fazla karışacaktır. Bazı eğitim veren mühendisler 8-bit mikrodenetleyicilere zaman kaybı dese de biz 8-bit mikrodenetleyiciler üzerinde çalıştığımızda ADC, USART, SPI, Kesme, Zamanlayıcı, EEPROM, PWM gibi pek çok konuyu öğrenmiş oluyoruz. Bu bilgilerle 32-bit mikrodenetleyicilere geçtiğimizde onlarda daha karmaşık olan bu ortak birimleri basit seviyede de olsa ilk bakışta kullanabilir hale geliyoruz. Zamanımızı tamamını anlamaya değil yeni özellikleri ve yeni birimleri anlamaya ayırıyoruz. Önceki öğrendiğimiz konuları 32-bit mikrodenetleyici üzerinde öğrenmek de belki 2-3 kat daha fazla zamanımızı alacak ve hatta belki de bazıları öğreneceği varsa da hiç öğrenemeyecek. Bu durumda 8-bit mikrodenetleyicileri öğrenmeye nasıl zaman kaybı diyebiliriz? 8-bit mikrodenetleyicilerin bütün dünyadaki üniversiteler

tarafından eğitimde kullanılması ve bunların üzerinde durulmasının bir sebebi vardır.

Bu noktaya geldiysek yeni başlayan birinin 32-bit mikrodeneleyicilerle başlamasının riskli bir yol olduğunu ve sağlam bir temel oluşturarak temkinli bir şekilde ilerlenmesi gerektiğini anlamış olmalısınız. O halde çoğunluğun seçeceği mikrodeneleyiciler 8-bit mikrodeneleyici olmalıdır. 8-bit mikrodeneleyici üreten firmalara baktığımızda pek çok firmayı görmekteyiz. Sadece PIC veya AVR değil STM8, Renesas, 8051 tabanlı vb. mikrodeneleyici aileleri mevcuttur. Bizim yeni başlayan olarak ve üretimde değil eğitimde kullanılmak üzere hangi mikrodeneleyiciyi seçeceğimizin kriterleri bir geliştiricinin seçeceği mikrodeneleyici kriterlerinden farklı olacaktır. Eğitimde kullanılmak için uygun mikrodeneleyici seçerken şu kriterleri göz önünde bulundururuz.

C derleyicisinin ve geliştirme stüdyosunun olması

Bazı mikrodeneleyiciler için sadece Assembly derleyicisini bulunabilir. Yakın bir zamana kadar aslında firmaların alışkanlıkları bu yöndeydi. Hatta PIC mikrodeneleyiciler için bile resmi C derleyicisi yoktu. Artık bu konuda ilerlediğimizi düşünürsek günümüzde resmi C derleyicisi ve ücretsiz geliştirme stüdyosu olmayan bir markaya pek iyi gözle bakmayız. Eğitimde kullanılacak bir derleyicinin açık kaynak olması büyük bir artıdır. Geliştirme stüdyosunun olması en azından konsol komutlarıyla derleme yapmamamız da önemlidir. Bu yönüyle Atmel Studio oldukça hoşumuza gitmekte.

Fiyatı ucuz, geliştirme ve deneme kartlarının yaygın olması

Önemli noktalardan biri de bu mikrodeneleyiciyi kolaylıkla temin edebilmemizdir. Bir öğrencinin bütçesini sarsmayacak şekilde günümüz teknolojisine göre makul fiyatlarda olmalıdır. Geliştirme kartları büyük firmaların geliştiricilerine yönelik yüzlerce dolar fiyatta değil öğrenci harçlığı ile alınabilecek fiyatlarda olmalıdır. Bu yönüyle Arduino (AVR) ve STM32 geliştirme kartları ön plana çıkmaktadır. Çok ucuz fiyata Arduino ve modüllerini alıp eğitim uygulamalarını yapabiliriz. Arduino'yu burada

“Mikrodenetleyici Geliştirme Kartı” olarak kullandığımızı unutmayalım. Bir geliştirme platformu olarak Arduino Core (API) farklı bir boyuttur.

Hakkında yeterli kaynak bulunması

Mikrodenetleyici eğitiminde öğrenciler oturup datasheet okuyarak iş başlamamalıdır elbette. Datasheet geliştiricinin okuyacağı bir doküman olup başlangıç seviyesinde datasheeti anlama yolunda ikincil kaynaklara ihtiyaç vardır. Bu kaynakların başlıcası kitaplardır. Kitaplar datasheetin ucuz bir çevirisi değil tecrübe, eğitimcilik, anlatım ve uygulamaların bir araya geldiği bir eğitim kitabı olmalıdır. O yüzden piyasadaki datasheetin “kırpılmış” halinden ibaret olan ve yazara ait cümleleri bile göremediğimiz kitaplara itibar edilmemelidir. Kitap konusunda PIC mikrodenetleyicilerin önde olduğunu görüyoruz fakat benim “C ile AVR Programlama” kitabım ve İngilizce kitapları hesaba katarsak AVR’nin bu konuda eksik olduğunu iddia etmemiz güç olmaktadır. STM32 içinse İngilizce eğitim kitaplarını bile pek görememekteyiz. ARM mimarisini genel olarak ele alan akademik kitaplar olsa da onlar başlangıç için uygun değildir. Kaynakların bir diğeri de forum ve bloglarda yer alan makaleler ve uygulamalardır. Kütüphane ve örnek kodları da unutmamak gereklidir. PIC ve AVR mikrodenetleyicilerin iki konuda da oldukça zengin olduğunu söylemek mümkün. Ama gidip Renesas ya da STM8 mikrodenetleyiciler hakkında aynısını bulmak mümkün değildir. O yüzden yeni başlayacak birinin zorluk çekmemesi açısından PIC veya AVR mikrodenetleyiciler ile başlamasını tavsiye ederiz. Kendine güvenen varsa başka mikrodenetleyicileri de tercih edebilir. Bu yazıda mikrodenetleyiciler arasında ayrımı yaptık ve seçilmesi gereken mikrodenetleyicilerin kriterlerinden bahsettik. Bir sonraki yazıda ise bir mikrodenetleyiciyi öğrenme sürecini sizlere açıklayacağım ve bu konuda nasıl ilerlemeniz gerektiğinden bahsedeceğim.

Bir Mikrodenetleyiciyi Öğrenmeye Başlamak

Önceki yazıda mikrodenetleyiciler ile gömülü sistemlere giriş yapmayı ve uygun mikrodenetleyiciyi seçmeyi size anlatmıştık. Diyelim bir mikrodenetleyici ailesinde veya modelinde karar kıldık ve bunu öğrenerek gömülü sistemlere giriş yapacağız. Bu durumda hangi adımları uygulamalı ve hangi konularda nasıl hareket etmeli bütün bunları da bu yazıda size anlatacağız. Daha ilerisi için artık sitemizdeki diğer yazılara ve yazdığımız kitaplara bakabilirsiniz ve kaynak araştırması yapabilirsiniz.

Öncelikle bir ön araştırma yapıp piyasada hangi mikrodenetleyici ailelerinin eğitim için uygun olduğunu saptamanız gerekli. Bizim size söylememiz yerine sizin araştırıp kendi kanaatinizle seçmeniz geliştiricilik yönünde atılan ilk adım olacaktır. Hangi mikrodenetleyiciyi seçtiğiniz değil hangi mikrodenetleyiciyi seçebileceğiniz burada önemlidir. Çünkü bundan sonraki bütün işlerinizi kendiniz araştırıp kendi kanaatinize göre yapmanız gerekir. Herkesin sözüyle hareket etmeye kalkarsanız nasrettin hoca fıkrasındaki durumu yaşarsınız. Siz kendi araştırmanızda farklı bir mikrodenetleyiciyi öğrenmekte karar kılısanız da ben kendi açımdan bir ön araştırma yapıp AVR mikrodenetleyicilerde karar kılıyorum. Bu yazıyı da bu örnek üzerinden devam ettireceğiz. AVR mikrodenetleyicileri öğrenmeye karar vermemizin sebebinde daha önceden bahsettiğimiz kriterler yer almaktadır. Ayrıca benim AVR mikrodenetleyicileri öğrenmede karar kılmam ise hakkında Türkçe kaynak olmasa da İngilizce videoları, blog yazılarını ve forumları takip edecek İngilizce'yi bilmemden dolayıdır. Eğer İngilizce bilgim yeterli olmasaydı hakkında Türkçe kaynağın daha fazla olduğu PIC mikrodenetleyicilere yönelecektim.

Kaynak araştırmasını yaptıktan sonra sıra kaynak toplamaya gelmektedir. Kaynak toplamak mikrodenetleyici hakkındaki eğitim meteryallerini arayıp bulup bir araya getirmek olarak özetlenebilir. Örneğin ben AVR öğrenmeye başlamadan önce AVR hakkında yazılmış İngilizce kitapları topladım, pdf dosyalarını bir klasörde depoladım, datasheet ve uygulama notlarını indirdim ve birincil ve ikincil kaynaklarımı elde etmiş oldum. Kaynak toplamaktaki ilk adım üretici firmanın internet sayfasına girip

mikrodenetleyici hakkındaki datasheet, uygulama notu, programlama kılavuzu, derleyici kılavuzu gibi “birincil” kaynakları toplamaktır. İkinci adım ise bu birincil kaynaklara en yakın olan ve birincil kaynaklardan faydalanmak için gerekli olan kitapları toplamaktır. Bunu da Amazon veya Türk kitap sitelerinden anahtar kelime ile yaptığınız bir taramaya göre belirlersiniz. Sonrasında karar kıldığınız kitapları elde etmeniz gereklidir. Kitap alırken bir kitapla her şeyi öğrenmeyi beklemek bir geliştiricinin yolu değildir. Hele Türkçe kitap alıyorsanız bazı kitaplar hiç öğretici nitelikte olmayabilir. Bunları göz önünde bulundurup kitaplar hakkındaki yorumları ve içeriği tarayıp birkaç kitap almanız daha isabetli olacaktır. Eğer İngilizce kitap okuyabiliyorsanız elde etmeniz daha kolay olacaktır. İngilizce kitaplarda hayal kırıklığı yaşama imkânınız daha azdır. Burada datasheet gibi üreticinin yayınladığı notlar birincil kaynak olup kitap yazarları da bunlardan faydalanarak anlatım, uygulama, tecrübe ve eğitimciliği ilave ederek kitap yazmaktadır. Eğer datasheeti okuyup rahatça anlayıp uygulayabilecek seviyedeyseniz en iyi kitap bile size pek birşey katamaz. Fakat ben Muhammed Ali Mazidi'nin kitabını okurken dijital elektronik ve mikroişlemci mimarisine dair pek çok bilgiyi öğrendim ve Assembly dilinin mantığını anladım. Burada tecrübe, anlatım ve farklı konulara değinmek datasheetin size öğretemeyeceği şeyleri öğretebilir.

Kaynak olarak birinci ve ikinci derecede kaynakların ardından pek çok Youtube kanalı, blog sitesi, forum ve eğitim içeriği üçüncü derece kaynaklar içerisine girmektedir. Bu noktada Google'ı iyi kullanmak ve yer imleri olarak bu kaynakları kaydetmek gereklidir. Datasheet ve kitap gibi derli toplu halde olmayan bu kaynakları bazen oldukça dağınık halde bulursunuz ve aradığınız bilgi için epey bir araştırma yapmanız gerekebilir. Bu kaynakların önemli yanı datasheette ve kitapta yazmayan bazı pratik bilgileri ve püf noktalarını bulabilme imkanınızın oluşudur. Elbette İngilizce kaynaklardan bahsediyorum. Türkçe gruptaki saçma sapan sorulara gelen cevapları okumakla bilginize çok da şey katılmayacaktır. Ama örneğin Arduino forumunda Arduino ve AVR programlamaya dair derin bilgileri okuyabilir ya da avrfreaks forumunda datasheette yazmayan hususların tartışıldığı ve paylaşıldığını görebilirsiniz. Böyle bir topluluğun olması her seviyedeki geliştiricilerin faydasına olmaktadır. O yüzden forumdan öğrenmeye

çalışmayın fakat bir mesele olduğunda Google'dan aratıp sorunuzun cevabını bulmasını bilin.

Youtube videolarına gelince ben Youtube'dan video izleyerek öğrenen biri değilim. O yüzden Youtube meselesine pek sıcak bakmıyorum. Fakat bazen uzmanların yaptığı ileri seviyedeki çalışmalar Youtube'dan paylaşılmakta. Örneğin Ben Eater'in üniversitede AVR'yi anlattığı dersleri Youtube'dan izleyebilirsiniz. Kısacası şovmen olmayan gerçekten işinin uzmanı eğitimcilerin videolarını izlemenin bir faydası olacaktır. Ama genel olarak şunu söyleyebilirim. Ben izlediğim Youtube videolarının %90'ından fazlasında kitaplarda bulduğum bilgileri ve derinliği bulamamaktayım. Youtube videolarının seviyesi oldukça basit gelmekte ve hakkıyla bir geliştirici olmak isteyenler için zaman kaybı olmaktadır. Saatlerce Youtube videosu izlesem de sonrasında tekrar kitap ve datasheet okumam gerekecek. O yüzden saatlerce izleyip vakit kaybetmek istemiyorum. Siz de birincil ve ikincil kaynakları esas alarak çalışırsanız zor bir süreçten geçerseniz de yüzlerce saat Youtube videosu izleyenlerden kat kat ileride olursunuz.

Kaynak toplamanın başlarda zor bir süreç olması ve uzun zaman alması sizi yıldırmamalı. Çünkü başlangıç için en büyük sermayeniz topladığınız kaynaklardır. Kaynak toplamanın ardından geliştirme kartı almak, derleme stüdyosunu kurmak gibi işler birkaç tıka bakmaktadır.

Elinizde yeterli kaynak olduktan sonra hemen uygulama yapmaya, robot yapmaya, uçak yapmaya kalkmayın. Öncelikle bu kaynaklara göz gezdirmeniz ve ön fikir edinmeniz gereklidir. Sonrasında sağlam bir giriş için sindire sindire okumanız şarttır. Takıldığınız noktada yine Google'ı kullanarak araştırma yapmanız gerekir. Bu noktada İngilizce bilmek ile hiç İngilizce bilmemek arasında dağlar kadar fark vardır. Hiç İngilizce bilmeyen birinin 6 ayda öğreneceği bilgiyi İngilizce bilen biri bilgisayarı etkili bir şekilde kullanıyorsa bir ayda öğrenebilir. Öğrenme sürecinde boş zamanlarınızı ana kaynaklardan farklı olarak Youtube, Google gibi sitelerde araştırma yaparak, haberleri takip ederek ve farklı yazıları okuyarak geçirmeniz iyi olur. Bu alanda kültürü edinmek için datasheeti okumak yeterli değildir. Eğer İngilizceniz orta seviyede (B1-B2) ise bu alanda yazılmış basit blog yazılarını okumakla işe başlayabilirsiniz. Tutorial adı

verilen bu yazılar basit bir dille yazılmıştır ve datasheet ya da kitap dilinden çok daha basit noktadadır. İnternetteki makalelerin genel olarak böyle bir basit dil yapısı vardır. İş kitap okumaya gelince birsürü teknik terim, gramer ve anlatım tarzı girmektedir. Datasheetleri ise öğrencilere yönelik değil geliştiricilere yönelik yazmaktadırlar. O yüzden sizin o teknik dili okuyup anlayabilmeniz gereklidir. Fakat teknik terimleri bir kenara bırakırsak datasheetler çok üstün seviye bir İngilizcede yazılı değildir. Bazen kitapları okuyup anlamak datasheet okumaktan daha çok İngilizce gerektirmektedir.

Öğrenim sürecinde en çok şikayet edilen konulardan biri de bir gün öğrendiğinizi ertesi gün unutmadır. Konuyu ezberlemeyip anlasanız dahi tam sindiremediğinizden dolayı kolaylıkla unutursunuz. Mikrodenetleyici seviye seviye ilerleyen bir konudur ve eski konuları unutmakla yeni konuları öğrenemez hale gelirsiniz. Bu yüzden öğrenmenin en etkili yollarından biri olan öğretmek ve yazma yollarını tercih etmelisiniz. Örneğin ben yazdığım kitaplarda ve blog yazılarındaki bilgileri unutmuş değilim. Yazarken en önemli nokta bunu kendi cümlelerinizle anlatabilmeniz gerektiğidir. Türkçe öğrenme tavsiyemizin ne kadar önemli olduğu burada da karşımıza çıkmaktadır.

Gömülü sistemleri öğrenmeye başlarken en önemli gördüğüm noktaları sizinle paylaştım. Pek çok nokta kalsa da yazımız yeteri kadar uzadığı için burada noktalayacağız ve yeri geldiğinde anlatmaya devam edeceğiz.

Arduino ile İlerlemek

Gömülü sistemlere yeni başlayanların karşılaştığı en büyük sorun ilerleyememek ve bir noktada kalmaktır. Çoğu kimse kendi isteği ile bir noktada kalmayı tercih etse de biz sürekli ilerlemek, öğrenmek ve en ilerisi olmak isteyenlere göre bu rehberi hazırladığımızdan bu yazıda da başlıca olarak bir yerde takılıp kalma sorununa çözüm getirmeye çalışacağız.

Bir yerde takılıp kalmanın en bariz örneği “Arduino” konusunda olmaktadır. Arduino gömülü sistemlere giriş için iyi bir araç olsa da medyanın telkini ve firmaların bu yönde çalışması ile gömülü sistemlerden ayrı bir “Arduino sektörü” meydana getirilmeye çalışılmaktadır. Biz Arduino kitlesinin ilerlemesini, öğrenmesini arzularken onlar Arduino kitlesini ellerinde tutmaya çalışmaktadır. Bu yolla sürekli ürettikleri Arduino kartlarına, modüllerine ve yayınladıkları kitap ve dergilere kalabalık bir abone topluluğu oluşturma peşindedir. Biz bu yönüyle Arduino tarafını eleştirmektediriz. Yüzlerce liralık onlarca eğitime giden birisi bile Arduino’yu hakkıyla öğrenememekte ve bir ileri aşamaya geçememektedir. Elbette Arduino kitlesinin de bu konuda kabahati vardır. Arduino’yu amaç olarak görüp yeniliklere kapalı, öğrenme taraftarı olmayan ve kendi isteği ile cahil kalmak isteyenlere istersek biz binlerce makale ve kitap yazalım yine söz dinletemeyiz. Arduino’yu gömülü sistemlere giriş için araç olarak kullanıp Arduino’nun faydalarından yararlandıktan sonra Arduino’yu bir kenara koyup daha ilerisine bakacak birilerinin çıkmasının bu kadar zor olması da ilginç noktalardan biridir. İki Arduino fonksiyonu öğrenmekle gömülü sistem geliştiricisi olmayı arzulayanlar ve iş yapmaya çalışanlar zaten bizim yazdıklarımıza itibar edecek değildir. Hatta azıcık bilgileri ile bize laf yetiştirme cüretinde bile bulunmaktadır.

Yukarıda verdiğimiz örnekteki bir yerde takılıp kalma durumu kişilerin istediği ile olduğu için onları kendi hallerine bırakmaktan başka çaremiz yoktur. Biz ilerlemek istediği halde bir yerde takılıp kendilerine yol gösterecek bir rehber arayanlara yönelik yazımıza devam edelim.

Arduino ile İlerlemek

Arduino'ya herkes başlayabilir fakat herkes Arduino'da ilerleyemez. Bir noktadan sonra ise hiç kimse ilerleyemez. Çünkü Arduino'nun yapısında bu vardır. Dört işlem öğrenip matematikte ancak pazarcı veya bakkal olarak ilerleyebilirsiniz. Kimse dört işlem bilerek matematik öğretmeni, profesörü ya da bir mühendis bile olamaz. Mesele burada Arduino'nun dört işlem kadar basit fakat önemli bir noktada olmasıdır. Herkesin Arduino'da ilerleyememesi ise ondalıklı, çok haneli veya negatif sayılar üzerinde dört işlem yapmayı öğrenememe durumuna benzetilebilir. Yani bazılarının bu alanda yeterli dirayeti ve yeteneği olmadığından basit konularda ilerlemesi bile çok yavaş olmakta ve hatta uzun zaman çalışsalar da ilerleyememektedirler. Bizim gömülü sistemler için gereksinimlerin içerisinde merakın yanında dirayeti de yazmamızın sebebi budur. Çalışmakla, merakla herkesin öğrenebileceği bir alan değildir. Bu konuda kavrayışınız, bakış açınız ve şahsi yönlerinizin yeterli olması gereklidir. Bunu okuyan yeni başlayan bazılarının hevesi kırılrsa da gerçek olan budur. Beş parmağın beşi de bir olmayacağına göre olmuyorsa başka bir alana yönelmek gereklidir. Buradan karşınıza gelen ilk zorlukta pes etmeniz gerektiğini söylemesem de bir kartın dizgisini 2 yıl boyunca uğraşıp yapamamak ya da 1-2 yıl uğraşsa da temel Arduino fonksiyonlarını anlayamamak gibi uç durumlar da yaşanmaktadır. Hatta bazen mühendisler içinde bile böyle durumlar görülmektedir. Örneğin iyi bir elektrik mühendisi olsa da C programlamayı bir türlü öğrenemeyebilir. Üstelik üniversite sınavı ve okullar asla sizin gerçekten öğrenip öğrenmediğinizi ve yeteneğinizi ölçemezler. Gömülü sistemler üniversite sınavı veya okul sınavlarını geçmeye benzemez. Gerçekten öğrenmeniz gereklidir.

Arduino ile ilerlemeye çalışmak belli bir noktadan sonra zaman kaybı olacaktır. Biz daha önce Arduino'nun zaman kaybı olmadığını savunsak da makalemizi doğru okuyabilenler Arduino'nun zaman kaybı olup olmamasının tamamen kişinin seviyesi ile alakalı olduğunu anlamış olmalıdır. Eğer başta saydığımız gereksinimlerde eksikiniz varsa ve bu gereksinimleri yeterli seviyeye getirmeye çalışıyorsanız Arduino ile bunu eş zamanlı olarak yapabilirsiniz. Size hem pratik sağlayacağı gibi temel bilgilerin de öğrenilmesine aracı olacaktır. Dikkat etmeniz gereken nokta bunu Arduino'yu öğrenerek öğrenemeyeceğinizdir. Örneğin ben Arduino

kitabımda temel elektroniğe, c programlamaya ve proje yapmaya dair ayrıntılar versem de bunlar Arduino bilgileri içerisinde değildir. Arduino ile proje yapabilmek için bile temel bilgiye sahip olmak gereklidir. Temel bilgileri öğrenirken aynı zamanda mikrodenetleyici programlamayı ve gömülü sistemlere kısa sürede geçiş yapmayı Arduino vasıtasıyla yaparız. Arduino bizi kısa süre içerisinde gömülü sistemler dünyasına sokar. Fakat gömülü sistemler arasında Arduino deryada bir damla gibi kalmaktadır. Siz Arduino ile başladığınızda en kısa sürede Arduino'yu öğrenmeye çalışıp bir sonraki platforma geçiş yapmalısınız. Normalde zor ile başlayacak olsanız da Arduino sayesinde kolay ile başlamış olursunuz. Fakat bu sizi asla kolayla alıştırmamalı.

Arduino'ya başlayanların daha ileriye gidememesindeki en önemli sebeplerden biri Arduino'nun kullanıcıları kolayla alıştırmasıdır. Çoğu firma da bu kolaycılığa yönelik ürün üretmekte ve kolaycılık üzerinden para kazanmaktadır. Örneğin birkaç kablo bağlatmamak veya devre kurdurmamak için hazır modül üretmekte ve bu modülün entegresini kullanmayı öğretmemek için de kütüphane yazıp hazır ürün olarak vermektedir. Sizin bu modül veya teknoloji hakkında bilginiz üç fonksiyon yazmaktan ibaret olduğu gibi on binlerce arduinocunun bilgisi de sizin bilginizle eşit olacaktır. Bu durumda farklı bir uygulama yapmak gerektiğinde kimseden cevap gelememektedir.

Arduino ile ilerlerken kesinlikle hazır kodları kopyalayıp yapıştırmamalı ve iki kablo bağlamaktan ibaret bir pratiği yapmamalısınız. Yeri geldiğinde ütü ile baskı devre yapmayı, yeri geldiğinde delikli pertinanks kullanmayı, lehim yapmayı ve araçları doğru kullanmayı öğrenmelisiniz. Hatta Arduino öğrenirken osiloskop alıp Arduino'da ürettiğiniz PWM gibi sinyalleri ölçmeyi lojik analizör ile gönderilen sinyallerin grafiğine bakmayı öğrenmelisiniz. Örneğin bir modül I2C veya SPI ile iletişime geçiyorsa bu protokoller hakkında teorik bilgiyi de öğrenmelisiniz. Kütüphaneleri kullanırken kütüphanelerin içlerine sık sık bakmalı. Oyuncakvari projeleri bir kenara bırakıp faydalı projeleri yapmaya yönelmelisiniz. Eğer internetten öğrenecekseniz Arduino forumundaki mesajları takip etmeli ve pek çok kaynaktan faydalanmalısınız. Gerçekten kaliteli bilgi pek göz önünde bulunmamakta ve bazen samanlıkta iğne arama hissi vermektedir. Ben bir sorun olduğunda bazen arama sonuçları içinde 40-50. sıralarda bu sorunun

cevabını buluyordum. Çok stresli bir araştırma yaparak öğrenme sürecinden sonra kitap yazacak kadar bilginiz olacaktır. Benim Arduino Eğitim Kitabı'nı yazmam bu şekilde olmuştu. Eğer Youtube'daki medya çalışanlarının "Evet arkadaşlar bugün Arduino ile şunu yapıyoruz." diye başlayan videolarını izleseymdim ne kitap yazabilecektim ne de ortaya koyabildiğimiz ürünler olacaktır.

Arduino ile ilerlemek konusu bu kadar uzadığı için geri kalan kısmı başka bir başlıkta size aktaracağım.

Arduino'dan İlerisine Geçiş

Arduino ile başlayan birinin bir ömür boyu Arduino üzerinde çalışması ancak keyif amaçlı uğraşıyorsa kabul edilebilir bir durumdur. Türkiye'de ise pek kimse sırf keyfi için yüzlerce dolar elektronik malzemeye para ayırmadığı ve çoğunun bundan para kazanma hedefinde olduğu için biz son zamanlarda artık işin hobi tarafını görmezden geliyoruz. Hatta bu internet sayfamızda hobicilere yönelik bir makale bile yazılmamıştır diyebiliriz. Bazıları pahalı elektronik oyuncak olarak bile görse de bizim bu görüşte olanlarla işimiz yoktur. Amatör veya profesyonel olsun öğrenmek, çalışmak ve ortaya bir ürün koyma peşinde olanlara yönelik içerik hazırlamaktayız.

Arduino'dan daha ilerisine geçiş çoğu zaman çok sancılı bir süreç olmaktadır. Öncelikle Arduino kitlesinden kendinizi koparmanız ve yalnız kalmanız gereklidir. Arduino kitlesinden kopmasanız bile zamanla ilerledikçe kendiliğinden kopmaya başlarsınız. Bu durumda yeni kaynaklar, yeni platformlar ve yeni topluluklar ile tanışmanız gereklidir. Örneğin Arduino forumunda takılmayı bırakıp Avrfreaks'e geçiş yapmanız gerekecektir. Çünkü bir noktadan sonra Arduino kitlesinin büyük çoğunluğu sizin seviyenizde olmayacak, sizin dilinizde konuşmayacak ve sizin dediğinizi de anlamayacaktır. Örneğin ben burada gömülü sistemler alanında çoğu zaman çok kıymetli bilgilerden bahsetsem de Arduino kitlesinin çoğunun dikkatini bile çekmemektedir. Ayrıca Arduino'dan ayrılırken hazır fonksiyonları, kütüphaneler, desteği ve projeleri bir kenarda bırakmanız ve yapacağınız projelere daha fazla zaman ve emek ayırmanız gerekecektir. Bu durumda işin sıkıntılı tarafı ile tanışacaksınız ve yazdığınız çoğu kod Arduino'daki gibi bir basitlikte olmayacaktır. Bu durumda şevkiniz

kırılabilir ve tekrar Arduino'ya dönmek isteyebilirsiniz. Çünkü Arduino az bir bilgi ile kolayca proje yapma imkanı sunarken aynı işleri Arduino'suz çok daha zor yaptığınızın farkına varırsınız.

Bu durumda Arduino'yı bırakmak için Arduino'nun kötü yanlarını gerçekten görmemiz lazımdır. Örneğin Arduino ile bir frekans üretici ya da frekans sayıcı yapamamak ve bunun için AVR kodlarını kullanmak bu sebeplerden biri olabilir. Fakat siz işinizde frekans üretici ya da sayıcı kullanmak zorunda kalmıyorsanız, Arduino'nun kararsızlığını ve yavaşlığını hissedemiyorsanız veya Arduino kütüphaneleri yetersiz kalmıyorsa bunu yaşayarak tecrübe etmeniz de mümkün olmayacaktır. Arduino'dan ilerisini öğrenme zorunluluğunu hissettirecek en büyük sebep bu olduğu için Arduino ile yapılamayacak şeyleri yapmaya çalışmanız çok önemlidir. Örneğin PORT işlemlerine olan ihtiyacınız bile Arduino'nun eksik olduğunu ve ilerisine geçilmesi gerektiğine ikna etmeye yetecektir. Bu da temel bilgileri öğrenmeniz ve yaptığınız uygulamaların oyuncaklıktan çıkıp gerçek uygulamalara doğru ilerlemesi ile olur.

Bunu yaşayarak tecrübe etmeniz şart değildir çünkü uzmanlar bu konuda bizim dediklerimizden çok farklısını dememektedir. Hiçbir uzman size "Arduino ile yapabileceğiniz hayal gücünüzle sınırlı, başka bir şey öğrenmenize gerek yok." diyemez. Bunu diyen pazarlamacılıkla eğitimciliği, şovmenlikle uzmanlığı karıştırmış birisidir. Benim Arduino'yu terk edip daha ilerine yönelme sebepim öğrenme isteğinden çok ihtiyaçtan dolayı oldu. Şu an ihtiyaç hissetmediğim durumlarda kısa zamanda deneme yapmak için yine Arduino'yu kullanmaktayım. Fakat bilgim Arduino bilgisinden ibaret olmadığı için tekrar Arduino kullansam da kütüphaneler yazabilmekte, donanımsal özellikleri kullanabilmekte ve Arduino'yu tam anlamıyla kontrol edebilmekteyim. Arduino'nun fonksiyonları ve kütüphaneleri içinde hapsolmuş değilim. Bu şekilde Arduino kullanabilmek bile Arduino ile mümkün olamamaktadır.

Örneğin sürekli kartlarını ve modüllerini aldığınız birkaç firma var. Bunlar dünya geneline bu modülleri satıyor ve siz de bunların modülüne para vererek yazdıkları kütüphane üzerinden Arduino ile modül kullanabiliyorsunuz. Bunların yaptığı on binlerce modülü kullansanız bile kendinize ait bir modül yapabilir misiniz? Yüzlerce kütüphane kullanan birisi

kendi kütüphanesini yazabilir mi? Elbette bunları yapamaz o yüzden mesele kütüphane ya da modül kullanıp kullanmamak değil bir noktada takılıp kalmaktır. Daha ilerisine gidemediğiniz sürece eliniz kolunuz bağlı olur. O durumda “Keşke şunun modülünü yapsalar, kütüphanesini yazsalar da kullansam.” diye bekler durursunuz. Kendi kartınızı yapabildiğiniz, modülünüzü üretebildiğiniz ve kütüphanenizi yazabildiğiniz durumda çoğu zaman onların ürünlerine de ihtiyacınız kalmayacaktır. Siz bir üretici ve geliştirici olarak bu aşamaları aşmanız gereklidir. Eğer tüketici ve kullanıcı olmakta ısrar ediyorsanız lütfen kendinize üretici ve geliştirici demeyin.

Daha iyisini yapmak, ortaya ciddi bir ürün koymak, özgün olmak ve kendinizi ifade edebilmek için Arduino’dan kurtulmanız gerektiğini kabul ettiniz. Bu noktada öğrenci olup işin başında olsanız da Arduino ile zaman kaybederek bunu öğrenmek istemiyorsunuz ve hemen daha ilerisine geçmek istiyorsunuz. Eğer bu durumda gerçekten nereden başlayacağınızı bilemezseniz aylarca zaman kaybedebilirsiniz. Örneğin Youtube’da bir üniversite öğrencisinin videosundan öğrenmekle yeterli seviyeye gelemezsiniz. Aldığınız Türkçe kitapların ciddi bir kısmı sizi ileri seviyeyi getirmeyi bırakın ilerletecek nitelikte olmaktan bile yoksundur. Bilgisini saklayan insanların kitap yazdığını, Youtube videosu çektiğini görebilirsiniz. Hem bilgilerini saklayıp hem de itibar kazanmak için eğitimcilik yapmanın ironisini gözlemlemekteyiz. Eğer bilgisini saklamayan ve gerçekten öğretme arzusu ile içerik hazırlayan bir eğitimci bulabilirseniz derhal ürettiği içerikleri takip etmelisiniz. Lojikprob sitesinin tamamen bu niyetle, olan bilginin aktarıldığı ve kar amacı gütmeyen hazırlandığını çok rahat söyleyebiliriz.

Doğru kaynağı ve içeriği bulmak da sizin bakış açınızla mümkün olmaktadır. Örneğin ben yazar ve eğitimci aday olarak hangi kitabın ve içeriğin kaliteli ve hangisinin kalitesiz ve iş görmez olduğunu bir okumada anlayabiliyorum. Fakat sizin bunu başlarda tecrübe ederek öğrenmeniz gereklidir. Bu tecrübe etme sürecinde ben zaman kaybetsem de sizin zaman kaybetmemeniz açısından Facebook grubumuzda kaynak paylaşımlarını yapmaktayım. Burada bir örnek vermemiz gerekirse tecrübe ettiğimiz bir durumu size aktaralım. Ben iletişim protokollerinin (SPI, I2C, UART) temellerini bilmiyordum ve kaynak araştırması yaparak bunu öğrenmeye çalışıyordum. Çoğu Arduino kaynağında işin ayrıntısı ve mantığı anlatılmadığı için anlamış gibi oluyordum fakat anlamıyordum.

Açıkcası SPI iletişim demek üç kablo bağlayıp iki Arduino fonksiyonu yazmaktan ibaret görünüyordu çoğu kaynakta. Üstelik bunun mantığını anlatmak üzere teorik bilgiyi verenlerin çoğu da yeterli anlatımda bulunmadığı için karanlık kalan noktalar çok oluyordu ve yine anlamıyordum. Sonrasında Gammon Forum'daki makaleleri keşfettim ve çok kısa sürede protokolleri ve iletişimin mantığını anladım. Daha sonrasında kitabıma ve eğitimlere de kaynak olan bu makalelere denk gelmem bu kadar bilgi kirliliği arasında çok uzun sürmüştü. O yüzden farklı bir çalışma ortaya koyamadıkça bilgi kirliliğine katkıda bulunmaktan pek de öte gidilemiyor. SEO'yu ve tık almayı dert edenler için bu pek sorun değildir. Böylelerinden de öğrenilir mi bunu siz sorgulayın.

Bunlar benim öğrenim sürecinde karşılaştığım zorluklar olsa da kimsenin bir daha bunlarla uğraşmaması için eğitim alanında faaliyet göstermek zorunda kaldım. Çünkü bu alandaki eğitimlerin çoğu eğitmemeye yönelik yapılmaktadır. Yurtdışında durum farklı olsa da ülkemizde genel olarak böyledir.

AVR mikrodenetleyicileri öğrenmeye başlarken öncelikle İngilizcemi ilerletmek zorunda kaldım. Çünkü hakkında Türkçe kaynak olmayan bir alanda ilerlemenin tek yolu buydu. Bu konular hakkında Türkçe kaynak olsa bile kaynakların çoğununun üniversite öğrencilerinin ben biliyorum demek için çektiği videolar ve SEO amaçlı yazılan tık alma peşindeki ucuz makalelerden öte gidemeyeceğini biliyordum. Bu konuda İngilizce blog yazıları ve kaynakların bile birinin yeterli olmadığını gördüm. Örneğin bir tutorial yazısında PORT işlemleri bitwise operatörler ile anlatılmıyor ve BV_() makrosu kullanılıyor ötekinde ise bitwise operatörler ile yapılıyordu. Bazı noktalarda eksik kısımlar vardı ve mantığını kavramak için aynı konudaki 5-6 blog yazısını bir araya getirmek zorundaydınız. Üstelik buna rağmen datasheet okumuş kadar bilgiye sahip olamıyordunuz. AVR hakkında pek çok İngilizce kitap bulsam da bunların arasında 2-3 tanesini gerçekten faydalı buldum. Bazı kitaplar "Çöp kitap" olarak nitelendirilebilecek seviyede zaman kaybı olabiliyor. İngilizce kaynakların kalitesini öğrenmek istiyorsanız üniversitelerde ders kitabı olarak okutulup okutulmadığını araştırmanız gereklidir. Benim kaliteli bulduğum ve bana faydası olan kitapların yurtdışında mikrodenetleyiciler dersinde ders kitabı olarak kullanıldığını gördüm. AVR mikrodenetleyici derslerini yazarken bu

yüzden kullandığım kaynak datasheet olmuştur. Çünkü diğer kaynaklardaki eksikliklerin farkına varmış ve tam kapsamlı bilgi vermek için bunların kaynak olamayacağını görmüştüm. Bu konuda İngilizce öğrenmenin, araştırma yapmanın ve tek bir kaynağa bağlı kalmamanın önemini kavramanız gereklidir.

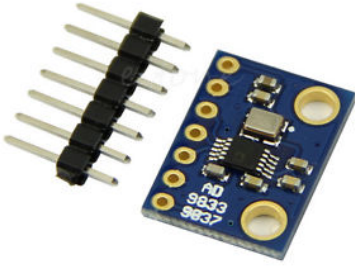
Biz Arduino ile başlayıp AVR ile devam etsek de bu süreç bizim için uzun ve zor bir zaman içerisinde oldu. Ben bunun çözümünü bu siteyi açıp kaynak oluşturarak giderdiğime inanıyorum. O yüzden Arduino'dan ilerisine geçmek isteyenlere AVR mikrodenetleyicileri tavsiye etmekte, sitemizden faydalanmasını tavsiye etmekte ve grupta paylaştığım AVR kitaplarını okumayı önermekteyiz.

8-Bit Mikrodenetleyiciler Üzerinden İlerlemek

Arduino ile ilgili tartışmalara bakınca bazen keşke Arduino hiç olmasaydı da bunlarla uğraşmak ve bunları tartışmak yerine herkes doğrudan 8-bit mikrodenetleyicileri programlamaya başlasaydı dediğim olmuştur. Günümüzde mühendislerin bile Arduino ile uğraştığını ve daha 8-bit mikrodenetleyicileri programlamayı öğrenemediklerini görmekteyiz. Arduino'nun gerçek geliştiricilik arasında nasıl bir perde görevi gördüğünü ve eğer doğru bakış açısıyla olaya bakılmadıkça bu perdenin kalkamayacağını tecrübe ederek öğrenmekteyiz. Benim gibi pek çok ileri seviye konuları anlatan insan bir konuyu anlatırken ilgi çekmesi için Arduino'yu bahane ederek konuyu anlatmaktadır. Örneğin normalde AVR'de zamanlayıcı, analog karşılaştırıcı kullanımını anlatacaksak Arduino'da zamanlayıcı ya da analog karşılaştırıcı kullanımı diye anlatmak zorundayız. Çünkü o sektör Arduino merkezli hale gelmiştir ve Arduino'ya rağmen biz bir şeyler anlatmak durumunda kalmaktayız. Bunu da yine Arduino'nun adını kullanarak yapmak durumundayız. Arduino merkezli düşünen ve konuya yukarıdan bakamayan kitlelerden ve Arduino'nun tekelleşmesinden aşırı derecede mustaribiz.

Ben mikrodenetleyiciler ve elektroniğe başladığım sıralarda Arduino'nun adı hiçbir yerde duyulmuş değildi. Fakat Arduino sektörü oluşmadığından

elektronik parça, modül ve geliştirme kartlarını da kolay kolay bulamıyorduk. Elimizde yeterli malzeme olmadığı için yapabileceğimiz oldukça kısıtlıydı. Şimdi ise çok güzel entegreleri bile modül haline getirip Arduino ile kullanılmak üzere pazarlamaktalar. Bu biz geliştiricilerin yararına olmuştur. Gerçi bu modülleri bir noktadan sonra kendimiz de yaparsak da işin başında kolay kullanılır ve erişebilir malzemeler oldukça artmıştır. Örneğin AD9833 dalga üretici modülünü birkaç tık ile alıp projemde kullanabilirim. Ben ilk başladığım zamanlarda bunu almak da kullanmak da hayal gibi birşeydi.



Bunun gibi ekranlar, konnektörler, entegreler ve çeşitli modüller bu Arduino ve büyüyen kitlesi sayesinde sürekli artmış ve geniş bir pazar haline gelmiştir. Bunu biz geliştiriciler lehimize kullanabiliriz. Aynı zamanda Arduino ile açık kaynak ve kod paylaşımı konusunda geliştiriciler motive olmuş ve pek çok kütüphane yazmıştır. Bu kütüphaneleri illa Arduino ile kullanmamız şart değildir. Pek çoğu bize yazılımsal açıdan fikir verecek ve yardımcı olacak kütüphanelerdir. Bir kütüphane nasıl yazılır sorusuna bir kütüphaneyi inceleyerek başlamak en iyi yollardan biridir. Ne kadar hazır yazılmış kodları ve kütüphaneleri incellerseniz o kadar ustalığı öğrenirsiniz. Bunu öğrendikten sonra Arduino kullanıcıları gibi bulduğunuz kütüphaneyi olduğu gibi kullanmazsınız. Hatta kodları incellerseniz ve beğenmezseniz düzeltebilirsiniz.

Ben 8-bit mikrodenetleyiciler üzerinde AVR ile ilerledim. Daha öncesinde PIC ile ilgili çalışmalarım olsa da onlar lise zamanlarında kaldığı için başlangıç seviyesinde kalmıştı. 8-bit mikrodenetleyiciler üzerinde ilerlemenin en iyi yolunun AVR öğrenmekle olduğunu iddia etmekteyim. Bunun en önemli sebebi ise mevcut olan kaynak kodlar, kütüphaneler ve Arduino'dur. Arduino'nun kaynak kodunu incelediğim ve size aktardığım

Arduino Kaynak Kodu İncelemesi adlı çalışmamda Arduino ile birşey yapmayı değil doğrudan Arduino yapmayı öğrendik aslında. Kaynak kodları inceledikten sonra Arduino'nun nasıl meydana geldiğini, eksik yönlerini ve bizim nasıl ideal bir kütüphane koleksiyonu (Framework) yapabileceğimizi anlamış oldum. Bunları inceledikten sonra Arduino'nun pek dikkate değmeyecek öğrenci projesi olmaktan öte de pek gidemediğini görebilme imkânına sahip oluruz. Zaten Arduino da bir öğrencinin master tezinden öte bir şey değildi. Bunun marka haline gelmesi, tekelleşmesi, büyük bir sektör olması medyanın manipülasyonu ile olmuştur. Arduino bize çok büyük bir icat gibi gösterilmekte ve eğitimlerde Arduino'dan ilerisine atıfta bile bulunulmamaktadır. Bu belki hobicilere yönelik böyle olsa kabul edilebilir fakat mühendislerin ve bu işte profesyonel olmak isteyenlerin bu şekilde zaman kaybetmesi kabul edilemez.

8-bit mikrodenetleyiciler üzerinde ilerlemenin en iyi yolunun AVR öğrenmek ve Arduino'nun kaynak koduyla kütüphanelerini incelemek olduğunu söylesek de işin PIC tarafını görmezden gelememek gereklidir. PIC programlarken yine AVR ve Arduino ikilisinde olduğu gibi işin basit ve karmaşık boyutu olmak üzere iki boyutu vardır. PIC Basic, Proton, CCS C, Mikro C gibi derleyiciler ile işin basit boyutunda kalsak da yazmaç tabanlı C dilinde program yazmakla ya da PIC Assembly ile işin ileri boyutuna gitme şansımız olur. 8-bit mikrodenetleyicilerde ileri gitmek için hazır fonksiyonları donanımdan bihaber kullandığımız derleyicilerden kurtulmamız gereklidir. Fakat başlangıç için Arduino'nun kötü olmadığını söylediğimiz gibi CCS C veya Mikro C'nin de kötü olmadığını söylüyoruz. Hazır fonksiyonların ve kütüphanelerin olması bize projelerimizde hız katacaktır ve yazılım uygulamalarını, pratiği yüzlerce sayfa datasheet okumadan edinmemizi sağlayacaktır. Fakat işin eninde sonunda ileri seviye olabilmek için muhakkak yazmaçlarla ve bitlerle muhatap olmak gereklidir. Assembly programcısı olmak ya da Assembly üzerinde program yazmak şart olmasa da bir platformun Assembly dilini öğrenip anlamak lazımdır. Bunu hiç kullanmasanız dahi günün birinde karşınıza çıktığında zorlanmadan öğrenebilmeniz için mantığını bilmeniz gereklidir. Örneğin ben AVR Assembly dilini ve mantığını kavrasam da Assembly dilinde bir satır kod yazmadım. Açıkcası kod yazma ihtiyacı duymadım. Fakat kod yazmak

gerektiğinde sıfır durumunda olmayacağım ve sadece teorik bilgimi pratiğe dökmek gerekecek.

Ben hobicilerin BASIC dili ile program yazmasına karışmasam da bir geliştiricinin C dilini öğrenmesi ve bir C derleyicisinde program yazmaya başlamasını kesin olarak savunmaktayım. Hatta Assembly öğrenilmeden önce C öğrenilmeli ve Assembly ile programlama işi sonraya bırakılmalıdır. AVR için Arduino -> AVR-GCC şeklinde bir ilerleme izlendiği gibi PIC için de CCS C/MikroC -> C dilinde yazmaç tabanlı programlama şeklinde bir yol izlenmelidir. Yani önce hazır fonksiyonların ve kütüphanelerin bulunduğu derleyicileri öğrensek de sonrasında datasheet okumaya başlamalı, temel bilgi ve donanım bilgisini ilerletmeli ve gömülü C programcılığını geliştirerek yazmaç tabanlı programlamaya geçmeliyiz. Durum böyle olduğu zaman işin çoğunu halletmiş sayılırız. Hatta pek çok geliştirici 8-bit mikrodenetleyicileri bu seviyede programlamayı öğrendikten sonra artık çok da ileri gitmemektedir. Aslında bu noktada durup bu konuda uzmanlaşmak gerekse de öğrencinin zamanı kısıtlı olduğu için diğer konuları öğrenmeyi çok geciktirmemesi lazımdır. Ben şu an kullanmasam dahi ileride ihtiyacım olacağından dolayı STM32'lerin mantığın çoktan kavramış ve programlanabilir mantık üzerinde ön araştırmalara başlamış durumdayım. Eğer sağlam bir temel ve yabancı dile sahip olmasaydım çok fazla tökezleyecektim. O yüzden kendine güvenmeyen öğrencilerin bu kadar hızlı gitmesini tavsiye etmiyorum.

Bazıları basamaklar arasında uzun süre durmayı zaman kaybı olarak nitelendirse de bu öğrenciye göre değişmektedir. Eğer öğrenci bir sonraki seviyeye geçmeye hazır olup da geçmiyorsa zaman kaybediyor demektir. Sizin bir sonraki seviyeye geçme zamanınızı yine siz bilmek zorundasınız. Bunun için en iyi ölçüt "Artık avcumun için gibi biliyorum. Öğreneceğim birşey kalmadı." diyebilmektir. AVR'yi avcumun içi gibi bilmeden STM32'ye başlasaydım ne AVR'yi öğrenebilecektim ne STM32'yi. O yüzden bir şeyi sağlam olarak öğrenin, o sizin garantiniz olsun ve sonrasında öğrenip öğrenemeyeceğiniz meçhul olan bir konuya giriş yapın. Bu bir programlama dilini tam olarak bilmeden bir diğer programlama dilini öğrenmeme konusunda da geçerlidir. Birini tam olarak bilmek 5-6 programlama dilini yarım yarım öğrenmekten daha iyidir. Yeni başlayanların tam öğrenmesi gereken iki temel konu AVR mikrodenetleyiciler ve C dili olmalıdır. C dilini

etkin bir şekilde kullanabilmeli, yazılı kodları anlayabilmeli, kütüphaneleri çözebilmeli, kütüphane yazabilmelisiniz. Aynı zamanda datasheeti de neredeyse ezberinizdeymiş gibi okuyup anlamalısınız. Burada ezberlemek değil anlamak ve kafanızda bir mikrodenetleyiciyi bina etmeniz gereklidir.

8-bit mikrodenetleyiciler üzerinden ilerlemeyi anlatırken pratiği, devre tasarımını, c programlama dilini konuya dahil etmiyorum. Çünkü bunlar üzerinde ayrı fakat eşzamanlı olarak ilerlemeniz gerekli ve hepsinin de ayrı ayrı püf noktaları bulunmaktadır. Bunları farklı bir yazıda ele alacağım için benim yazdığım yazıları da ayrı ayrı okumalı fakat eşzamanlı olarak takip etmelisiniz. İlerleyen yazılarda farklı yönlerden gömülü sistemler üzerinde ilerleme konusuna devam edeceğim.

Mikrodenetleyicileri Öğrenmek Yeterli mi ?

Gömülü sistem geliştiriciliğini donanım ve yazılım olmak üzere iki kısma ayırabiliriz. Yalnız ülkemizde kalifiye eleman eksikliği sebebiyle ayrı ayrı donanımcı ve yazılımcı olmanın getireceği pek çok sorun olacaktır. Üstelik kendi işinizi yapıyorsanız ister istemez bütün konularda bilgi sahibi olmanız gerekecektir. Çünkü donanım olmadan yazılımla, yazılım olmadan da donanımla ortaya bir ürün koymamız mümkün değildir. Bu yüzden sizin de donanım ve yazılım konusunda eş zamanlı olarak kendinizi geliştirmeniz ve bir noktada sabit kalmamanız gereklidir. Biz Arduino öğrenirken Arduino'yu bahane ederek gömülü sistemlere giriş yapıp en temel bilgileri öğrenmeye başladığımız gibi 8-bit mikrodenetleyicileri öğrenirken de aslında 8-bit mikrodenetleyicileri bahane ederek temel bilgilerde ilerlemiş olmaktadır. Burada başka bir bilgiye sahip olmadan datasheeti okuyup mikrodenetleyiciyi öğrenmemiz bize pek bir fayda getirmeyecektir. Ayrıca datasheeti anlamak için bile temel bilgi gerektiğini ana dili İngilizce olsa dahi temel bilgisi olmayan birinin datasheetten bir şey anlamayacağını daha öncesinde size söylemiştik.

8-bit mikrodeneleyici üzerinde alıřmak bize pek ok alanda bilgi ve deneyim katmaktadır. Bu alanları kesin olarak belirlemek istersek řunlardan bahsedebiliriz,

- Analog Elektronik
- Dijital Elektronik
- Devre Tasarımı ve Üretimi
- Programlama
- Mikroişlemci/Bilgisayar Mimarisi
- Pratik Elektronik

Bu yazdığımız konuların gömülü sistemler üzerinde alıřacak olanların karşılaması gereken gereksinimlerden ok farklı olmadığını görmekteyiz. Yalnız bu gereksinimler üzerinde ayrı ayrı ilerlemeye kalkarsak oldukça fazla zaman harcarız ve bu bilgileri birbirine bağlamakta sıkıntı yaşayabiliriz. Örneğın ok güzel analog veya dijital devre kurarız fakat bunları birbiriyle beraber kullanmayı bilemeyebiliriz. Mikrodeneleyiciler bütün bunları bir araya toplamakta ve gerçekten iş yapabileceğimiz bir alan olmaktadır. Bazı mühendislik öğrencileri o kadar teorik bilgi bilmesine rağmen bunları en basit derecede bile pratiğe dökememektedir. Biz bunu istemediğimizden dolayı bu ilerlemeyi başta değil pratik yaparak ve konuları beraber işleyerek yapmak isteriz.

Örneğın bir mikrodeneleyici üzerinde alıřmak istediğimizde analog elektroniğın basit konularına ihtiyaç duyarız. Bir led yakmak için gereken gerilim, diren kullanma, besleme devresi, osilatör devresi, koruma diyotu gibi basit bir mikrodeneleyici sistemi tasarlamak için gereken analog elemanları ve devreleri öğrenmek zorunda kalırız. Fakat işi ilerlettiğimizde röle, transistör, potansiyometre gibi elemanlar karşımıza çıkar ve bu röleleri sürerken diyot kullanmak gerektiğini, gerilim bölücü direnleri ve analog okumayı öğreniriz. Daha ilerisinde ise motor sürerken PWM sinyalleri, frekansları öğrenmek zorunda kalırız. Kısacası mikrodeneleyici ile yaptığımız kontrol uygulamalarının çoğı analog elektronik bilgisi gerektirmekte ve bizi bu konuda mecbur bırakmaktadır. Analog elektronik konuları uygulamalarda basitten zora gittiği için mikrodeneleyiciler üzerinde alıřırken analog elektronik konularını kademeli bir şekilde öğrenebiliriz. Bu örnekte mikrodeneleyiciyi öğrenmenin yeterli olmadığını aynı zamanda analog elektroniğe dair bilgilere sahip olmanız gerektiğini anlayabilirsiniz.

Dijital elektronik konusunda mikrodenetleyiciler analog konularda olduđu kadar fakir deđildir. Kendi i birimleri dijital sistem üzerine bina edildiđinden pek ok dijital elektronik uygulamasını ekstra bir dijital entegreye ihtiya duymadan yapabilirsiniz. Fakat mikrodenetleyicileri ğrenmek ve anlamanın dijital elektroniđi ğrenmekten getiđini bilmeniz gereklidir. Dijital elektronik sadece 74 serisi entegreler ile sınırlı deđildir ve ođu zaman kullandıđınız algılayıcı, modül ve entegre dijital sisteme entegre edilmiř halde bize gelir. Artık LM35 gibi analog sıcaklık algılayıcıları yerine dijital sistemi kullanan DHT22, DS18B20 gibi sıcaklık algılayıcıları kullanılmaktadır. Siz bu noktada seri iletiřim ve dijital kontrol seviyesinde ilerlemeniz gereklidir. Ayrıca 74 serisi dijital entegreleri kullanmak bazı noktalarda hayat kurtarıcı olabilir. Mikrodenetleyici uygulamalarında pek ok zaman temeli yazma ya da oklayıcı kullanmanız gerekecektir. Bazen sayalar vasıtasıyla frekans blc yapmanız gerekecek bazen de dijital-analog ya da analog-dijital evirici entegreleri kullanmanız gerekecektir. Ayrıca src entegrelerden de bahsedebiliriz. Bu konuda dijital elektroniđi ğrenmek mikrodenetleyici tabanlı uygulamalarda analog elektronikten bile nce gelmektedir. Ayrıca iřin mantıđını anlamak iin temelini ğrenmek gerektiđi konusunda hemfikir olmalıyız. Dijital elektronik analog elektronik gibi matematik ve fizik bilgisi gerektirmediđi ve mantık tabanlı olduđu iin ğrenmesi de kolaydır. O yzden en azından 1-2 sayısal elektronik kitabı alın ve dijital elektronik hakkında temeliniz olsun.

Devre tasarımı ve retimi mikrodenetleyicileri ğrenirken ister istemez ğrendiđimiz konulardan biridir. Fakat bunun artık eskiden bu řekilde olduđunu syleyebiliriz. Artık geliřtirme kartı, modl ve breadboard kullanım oranı ciddi bir biimde artmıřtır ve ođu projeyi artık breadboard zerinde kabloları birbirine bađlayarak yapmaktayız. Bu durumda devre kurmayı ve kalıcı devre tasarımıını ğrenememekte ve hatta yaptıđımız projelerde bile breadboard kullanmak zorunda kalmaktayız. rneđin ben pek ok đrencinin robot yapmak istediklerinde řasenin zerine breadboard koyduklarını ve devreyi bunun zerinde yaptıklarını grmekteyim. Bu bizim tasvip etmediđimiz bir durumdur. Proteus, EasyEDA, Sprint-Layout, Eagle gibi programları ğrenmek gerekli olup buradan izdiđimiz PCB'leri retmeyi veya rettirmeyi bilmemiz gereklidir. PCB kart tasarımı dijital elektronik gibi kitabını alıp okuyup ođu bilgiyi ğreneceđiniz bir alan

değildir. Pratik ve tecrübeye dayalı olduğu için zamanla onlarca PCB çizerek iyi bir PCB'nin nasıl yapılması gerektiğini öğrenmeniz gerekecektir. Aynı zamanda yapılmış PCB'leri incelemeniz de bu yönde size katkı sağlayacaktır.

PCB çizmek ve üretmek kolay olsa da sizin yaptığınız iş "kart üretmek" ile sınırlı kalmamalıdır. Bir kart kapalı bir kutuya girdiği zaman cihaz haline gelmektedir. Bu projenin son aşaması olup kartı cihaz haline getirmekte de bir el becerisi ve işçilik gereklidir. Bu yönde pratiğinizi geliştirmeniz ve gerekli ekipmanları almanız gereklidir. Örneğin panel tasarımı, etiket tasarımı, kutulama gibi konularda çalışmanız gereklidir. Her biri ayrı bir ustalık olduğu gibi mikrodenetleyici öğrenirken pek ala öğrenebileceğiniz konulardır. Örneğin LCD ekran ve düğmelerden ibaret bir devre tasarlasanız da bir elektronik cihaz kutusu alıp bunun panelinde LCD ekran ve düğmeler için yer açmalı, aradaki kablo bağlantılarını çoğu zaman konnektörler vasıtasıyla yapmalı, vidalarla elemanları sabitlemek için vida yuvası açmalısınız. Bu durumda alacağınız pek çok malzeme ve ekipman olacaktır. İş yapmak istiyorsanız bu konuda da uzmanlaşmanız gerekecektir.

Mikrodenetleyiciler üzerinde çalışırken aynı zamanda programlama yeteneğimizi geliştirme imkanı buluruz. Bu programlama yeteneği sadece mikrodenetleyiciyi programlama yönündeki Assembly/C dillerini kullanmadan ibaret değildir. Bir mikrodenetleyici projesini bilgisayar veya akıllı telefon ile beraber kullanmamız gerektiğinde bunlara ait yazılımı da öğrenmek zorunda kalırız. Biz hep zorunda kalmak tabirini kullansak da öğrenmenin anahtarı aslında bu tabirdedir. Örneğin size okulda matematik dersi anlatırlarken ne günlük hayatta kullandığınız ne de kullanmak zorunda kalacağınız bilgileri anlatmaktalardı. Pek çok öğrenci de bunlar bizim günlük hayatta ne işimize yarayacak diye sorup duruyordu haliyle. Öğretmenlerin buna pek bir cevabı olmasa da biz burada şunu öğrenin dediğimizde ne işinize yarayacağını da sizlere açıklayabiliyoruz. Burada zorunda olmadığı halde öğretilen bilgilerin yerine öğrenmek zorunda kalınan bilgileri görmekteyiz. Gömülü sistemler üzerinde çalışmanın en güzel noktalarından biri de budur. Siz birşeyler yapmak için öğrenmek zorunda kalıyorsunuz ve işinize yaramayacak bilgiyi öğrenmiyorsunuz. O yüzden ben gidip size şu dili veya şu platformu öğrenin demeyeceğim. Siz proje

yaptıkça ve bir alanda çalışmaya başladıkça zaten onu öğrenmek zorunda kalacaksınız.

Mikroişlemci ve bilgisayar mimarisini öğrenmek zorunda olduğunu hissetmek sizin ilerlediğinize en büyük delillerden biri olacaktır. Çünkü bu noktada artık siz datasheeti incelemeye başlamış ve diyagramları pek anlayamamış yada Assembly'e geçip komutların mantığını anlayamamış olmanız gerektir. Bu noktada sizin en ileri adıma geçebilmeniz için bunu öğrenmeniz şarttır. Bizim mikroişlemci ve mikrodenetleyici mimarisi üzerinde yoğunlaşmamız ve bu konuda eğitim makaleleri hazırlamamız sizi en ileri seviyeye götürmek içindir. Çoğu eğitim ve kitapların hedefi sizi giriş seviyesinde tutmak iyileri de orta seviyeye getirmek olsa da biz kimsenin orta seviyede kalmasını kabul etmiyoruz. Biz kendimiz ilerlediğimiz gibi diğerleri de bizimle beraber ilerlemelidir. Bu konuda en büyük adımlardan birinin mimari öğrenmek olduğunu tecrübe ederek öğrenmiş birisi olarak sizin de bir noktadan sonra mimari öğrenmeniz gerektiğini savunuyorum. Mikrodenetleyicileri öğrenirken ister istemez ileri seviye için mimari öğrenmeniz gerekecektir. Çünkü datasheetin büyük kısmı mimariden ibarettir ve mimariyi tam olarak anlamadıkça mikrodenetleyiciyi anlamış olmazsınız. Bu mimariyi öğrenmek için tek kaynak datasheet olmamaktadır. Mimari eğitiminde sizin için gerekli olan bilgiler sitemizde yazılıdır. Bu konuda Türkçe kaynak çok az olduğu için mimari alanına özellikle yoğunlaştım.

Pratik elektronik sürekli tecrübe ettiğiniz ve yaşayarak öğrendiğiniz bir konudur. Fakat pratiğe yönelik elektronik kitapları da mevcuttur. "Arduino ile şunu yapıyoruz." diye iki kablo bağlamanın pratik olmayacağını şimdiye kadar anlamış olmanız gereklidir. Pratikte size en büyük katkısı olacak şey tam kapsamlı bir elektronik cihazı baştan sona yapmaktır. Örneğin EPE dergisi gibi dergilerde bazı projeler bu niteliktedir. Hazır projeyi bakarak yapmanız bile size pratik açısından büyük fayda sağlayacaktır. İlla kendi projeniz olması gerek değildir. Hobi elektronikçiler için pratik yönüyle daha çok ilgilenip dergilerde ve internette hazır olan devreleri ve projeleri sıkça kullanmaktadır. Sizin **pratik kazanma gayesiyle** bu hazır projeleri yapmanızda bir sakınca yoktur. Sizin de pratik kazanmak için oyuncakvari projeleri değil yabancı elektronik dergilerini ve kaynaklarını tarayarak güzel projeleri yapmanız gereklidir.



Bir dergide yer alan amatör projenin amatörler tarafından cihaz haline getirilmiş halini yukarıda görmekteyiz. Aslında bu konu tamamen sizin ustalığınıza bağlıdır. Aynı projenin bir başka biri tarafından şu şekilde cihaz haline getirildiğini görmekteyiz.



Gördüğümüz gibi bu kutulamayı, paneli ve etiket tasarımını yapmadıktan sonra cihaz ticari bir değeri olmayan elektronik kart olmaktan öte gidememektedir. Bu konuda çeşitli yöntemler ve aletler bulunmaktadır. Örneğin bu panelde düğmeleri için açılan delik veya ekran için yapılan pencere CNC tezgahıyla yapılabilir.

Görüldüğü gibi bütün bunlar mikrodenetleyici merkezli ve mikrodenetleyiciler ile beraber yürütülmesi gereken birçok alan olmaktadır. Bunları öğrenmeden sırf bir mikrodenetleyici bilgisi ile pek bir şey yapmanız

mümkün değildir. O yüzden mikrodenetleyicileri öğrenirken bakış açınızı daraltmamalı ve öğrenmeye oldukça aç olmalısınız.

Datasheet Okumak

Datasheet okumanın önemini sık sık dile getirsek de gerçekten önemli bir konu olduğu için datasheet için ayrı bir makale yazacağız. Datasheet okumanın ne kadar önemli olduğunun idrakine varamamak yeni başlayanlar için yaygın bir sorundur. O yüzden datasheet okumanın ne kadar önemli olduğu konusunda sizi ikna etmeye çalışacağım.

Öncelikle bir elektronik malzemenin hakkında bütün bilgileri edinebileceğiniz bir doküman bulunması şarttır. Bu dokümanı da doğal olarak üretilen parçayı en iyi bilen üretici yazıp yayınlayacaktır. Üreticinin yayınladığı teknik veri kitapçığında (datasheet) ürüne ait teknik özellikler, karakteristikler, devre şemaları, kod örnekleri, algoritma blokları, karakteristikler ve standart bilgileri yer almaktadır. Elbette üretici bu bilgileri bizimle paylaşmak zorundadır fakat her zaman her bilgiyi bizimle paylaşmamaktadır. Yani ürünü nasıl ürettiğini, ürünü nasıl programlamanız gerektiğini, transistörün ne olduğunu, ALU'nun ne işe yaradığını size öğretmek zorunda değildir. O yüzden ana dili İngilizce olan birisi dahi temel bilgileri bilmedikçe ve teknik terimlere aşina olmadıkça teknik veri kitapçığından bir şey anlamayacaktır.

Kısacası üretici bize elektroniği öğretmek için teknik veri kitapçığı yazmamaktadır. Üretici, geliştiricilere ürettiği ürün hakkında bilgi vermek için bunları yazmaktadır. Daha ilerisi ticari sırlar olduğu için ürün hakkındaki en geniş bilgiyi teknik veri kitapçığı dışında bir yerden bulamazsınız. Konu hakkındaki veya teknoloji hakkındaki en geniş bilgiyi demiyorum. Ürün hakkındaki bilgi teknik veri kitapçığında olsa da örneğin transistörler, diyotlar, mikroişlemci mimarisi gibi konular hakkındaki en geniş bilgi akademik kitaplarda ve yayınlanan makalelerde bulunmaktadır. Yani işin teorik boyutu ve temel bilgiler okulda öğrenilmesi gerekmektedir. O yüzden kimse bir mikrodenetleyici öğrenirken veya elektroniğe başlarken açık teknik veri kitapçığını okuyarak işe başlamaz. Bu alanda yazılmış eğitim kitaplarını ve akademik yayınları takip eder. Yalnız bazen teknik veri kitapçığı da "okumasını bilene" pek çok şey öğretebilir. Örneğin dijital elektronik ve mikrodenetleyici mimarisi üzerinde temel bir bilginiz var fakat

işin ayrıntısını bir örnek üzerinden görmek istiyorsanız PIC veya AVR mikrodenetleyicilerin teknik veri kitapçığını açıp okuyarak bazen kitaplarda görmeyeceğiniz ayrıntıyı orada görüp öğrenebilirsiniz.

Bir geliştiricinin öğrenme kaynağı bir noktadan sonra teknik veri kitapçığı olmaktadır. Çünkü temel bilgileri bildikten sonra işin ayrıntısını, uygulamasını ve gelişen teknolojiyi teknik veri kitapçıklarını okuyarak takip edebiliriz. Biz yazımızın devamında bir teknik veri kitapçığını inceleyip ne yönden bunu okumanın önemli olduğunu sizlere anlatacağız. Eğer teknik terimlere aşina olmuşsanız ve temel bilgileriniz ileri düzeydeyse İngilizce konusunda ileri olmasanız dahi sizin işinize yarayacak bilgiyi bu kitapçıktan bulmanız mümkündür. Çünkü bu dokümanlar bir İngiliz edebiyat kitabı değildir ve çoğu bilgi tablo veya şema halinde verilmektedir.

Arduino eğitimlerinde üç ayaklı transistöre benzeyen bir entegreden gelen analog değeri bir formül ile okuyup sıcaklık değerini ölçüyorduk. O entegrenin adı ise LM35 idi. LM35 hakkındaki bilgilerimiz Arduino seviyesinde bir sıcaklık algılayıcısı olduğu, bundan gelen çıkışın analog olarak okunup sıcaklığa çevrildiği yönündedir. Bir üretici geliştiricilere bu kadar sığ bilgileri vermese gerektir. Fakat bu bilgileri de ne kitaba bu şekliyle koyabiliriz ne de LM35 eğitimini verebiliriz. O yüzden datasheetteki bilgiyi başka yerde bulmanız çok çok zor olmaktadır. Youtube videosu izleyerek elektronik öğrenen mühendislerin halini siz düşünün.

LM35'in teknik veri kitapçığını sizin de indirip bizimle beraber incelemeniz gereklidir. O yüzden aşağıdaki bağlantıdan bunu indirip hep beraber inceleyelim.

<http://www.ti.com/lit/ds/symlink/lm35.pdf>

Öncelikle datasheetin en baş kısmına baktığımızda şöyle bir başlık görürüz.



LM35

SNIS159H – AUGUST 1999 – REVISED DECEMBER 2017

LM35 Precision Centigrade Temperature Sensors

Görüldüğü gibi üreticinin logosu ile beraber bu elemanın ne olduğu bir başlıkta özetlenmiştir. LM35 Kararlı(Tekrarlanabilir) Santigrat Sıcaklık Algılayıcıları olarak başlıkta bize bildirilmiştir. Çoğu zaman elektronik

gruplarına parça üzerinde yazan adlar yazılır ya da fotoğrafı çekilerek bu ne diye sorulur. En basit olan bu sorunun cevabını bile başlıkta görmekteyiz. LM35'in sıcaklık algılayıcısı olduğunu Arduino'ya yeni başlayanlar bile bilse de binlerce çeşit algılayıcı ve entegre bulunmaktadır. Bunların her biri için bu dediğimiz geçerlidir. Burada LM35'den örnek vermemiz sizin yabancılamanız içindir.

Teknik veri kitapçıklarında ilk sayfada özet bir bilgi yer almaktadır. Siz geliştirici olarak sayfalarca okuyacak zamanınız olmayabilir ve işinize yaramayacak entegrenin de teknik veri kitapçığını sayfalarca okuyarak zaman kaybedebilirsiniz. Bu yüzden bu kitapçıkların ilk sayfasına üreticiler çok önem vermişler ve ürün ile ilgili hemen hemen bütün bilgiyi özet halinde burada vermişlerdir. Geri kalan ayrıntılı bilgiler, devre şemaları ve uygulamaya dair bilgiler kitapçığın devamında yer almaktadır. İlk sayfaya baktığımızda üç ana başlıkta bu bilgilerin verildiğini görmekteyiz.

1 Features

- Calibrated Directly in Celsius (Centigrade)
- Linear + 10-mV/°C Scale Factor
- 0.5°C Ensured Accuracy (at 25°C)
- Rated for Full -55°C to 150°C Range
- Suitable for Remote Applications
- Low-Cost Due to Wafer-Level Trimming
- Operates From 4 V to 30 V
- Less Than 60-µA Current Drain
- Low Self-Heating, 0.08°C in Still Air
- Non-Linearity Only ±¼°C Typical
- Low-Impedance Output, 0.1 Ω for 1-mA Load

2 Applications

- Power Supplies
- Battery Management
- HVAC
- Appliances

3 Description

The LM35 series are precision integrated-circuit temperature devices with an output voltage linearly-proportional to the Centigrade temperature. The LM35 device has an advantage over linear temperature sensors calibrated in Kelvin, as the user is not required to subtract a large constant voltage from the output to obtain convenient Centigrade scaling. The LM35 device does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55°C to 150°C temperature range. Lower cost is assured by trimming and calibration at the wafer level. The low-output impedance, linear output, and precise inherent calibration of the LM35 device makes interfacing to readout or control circuitry especially easy. The device is used with single power supplies, or with plus and minus supplies. As the LM35 device draws only 60 µA from the supply, it has very low self-heating of less than 0.1°C in still air. The LM35 device is rated to operate over a -55°C to 150°C temperature range, while the LM35C device is rated for a -40°C to 110°C range (-10° with improved accuracy). The LM35-series devices are available packaged in hermetic TO transistor packages, while the LM35C, LM35CA, and LM35D devices are available in the plastic TO-92 transistor package. The LM35D device is available in an 8-lead surface-mount small-outline package and a plastic TO-220 package.

Burada bize üç ana başlık verilse de ürünün tipine ve üreticiye göre bu başlıklar değişmektedir. Fakat belli başlı noktalarda aynı kalmaktadır. Bu aynı kalan şeyler teknik özellikler, uygulama alanları ve açıklamadır. Burada da özellikler kısmına baktığımızda bizim ihtiyacımız olan eleman olup olmadığını hemen öğrenme şansımız olacaktır. Hatta elemanı tercih

ederken geliřtiriciler genelde ilk sayfayı okumakla yetinebilirler. Bu yüzden üreticiler datasheetlerin ilk sayfasında biraz pazarlama ağızı ile yazmakta ve ürünlerini övmektedir. Ürünün dezavantajlarını ve sınırlarını saklamasalar dahi burada pek göz önüne getirmezler. O yüzden ilk sayfa hem bilgi verme konusunda çok önemli hem de olmayan bilgi konusunda dikkat etmemiz gereken bir sayfadır. Bunu STM32'yi anlatırken de başlarda size anlatmıştım.

Burada özelliklerin ne olduğunu inceleyelim ve önemli noktaları size anlatalım.

Calibrated Directly in Celsius, Burada üretici entegrenin santigrat dereceye göre kalibre edildiğini bize bildirmektedir. Örneğin NTC ve PTC'ler basit birer direnç olmakta ve kalibrasyon işlemleri zaman kaybına sebep olmaktadır. Burada kalibre edilmiş bir hazır entegre olması tercih edilesi sebeplerdendir.

Linear + 10-mV/C Scale Factor, Derece başına 10mV skalasının olduğunu bizlere bildirmektedir. Bu entegrenin nasıl çalıştığı konusunda bizi aydınlatacak ilk bilgidir. Linear bildiğiniz üzere çizgisel demektir ve bu terimi bilmeden fikir sahibi olamayacağımız gibi Scale Factor'ün de ölçek emsali yani ölçeklendirmede olan çarpan faktörü olduğunu bilmezsek bunu anlamamız mümkün olmaz. O yüzden bu tarz terimleri gerek önceden gerek de okurken araştırarak öğrenmemiz lazımdır. Böyle terimlerin çevirisini Türkçe olarak öğrenebilmeniz de İngilizce sözlüklerde açıklamasını da bulabilirsiniz. Kısacası konu yine araştırmaya gelmektedir.

0.5C ensured Accuracy (at 25c), Accuracy isabetlilik olup burada terim anlamında kullanılmaktadır. Pek çok algılayıcı için bu terimi kullanmaktayız. İsabetliliği hassasiyetle karıştırmamak gereklidir. Burada da 0.5C isabetliliğe yani +- 0.5C hata payına sahip olduğu anlamına gelmektedir.

Rated for Full -55 to +150C Range, ölçüm aralığı burada belirtilmiştir. Bir sıcaklık algılayıcısını tercih ederken en önemli iki etken isabetlilik ve ölçüm aralığı olduğu için bunlar burada yazılmıştır.

Sonraki bilgilerde düşük maliyetli olduğundan, çizgiselsizliğin düşük olduğundan, ısınmanın az olduğundan bahsedilmektedir. Burada örneğin pazarlama diline ait bir kelimeyi burada görebiliriz "Only" ifadesi sadece anlamı taşımaktadır ve malı burada övmektedir.

Applications başlığında ise bu algılayıcının kullanıldığı alanlardan bahsedilmektedir. Bu alanlar asla kullanım alanını sınırlandırıcı öğeler değildir. Sadece kullanımının uygun olduğu ve fikir vermesi açısından verilen birkaç örnektir. Description, yani açıklama kısmında ise LM35'in ne olduğu daha ayrıntılı olarak açıklanmaktadır. Bu açıklama kısmı genelde teknik dilden biraz daha uzak olup sizin anlayabileceğiniz yapıdadır. O yüzden teknik konuda zayıf olanların en rahat anlayacağı kısım burası olacaktır.

Datasheetin devamında bu entegre ile alakalı hemen herkes için bilgiler yer almakta ve bizim için örnek devre ve uygulama notları bulunmaktadır. Yazı yeterince uzadı için geri kalanını incelemeyi size bırakıyoruz ve internette olsun kitaplarda olsun bu sıcaklık algılayıcısı hakkında bu kadar ayrıntılı bilgiyi başka yerde bulamayacağınızı görmek istiyoruz.

Datasheet okumaları yapmak oldukça eğitsel bir özelliğe sahip olduğundan ilerleyen zamanlarda datasheet üzerinde okumalar yapacağız.

32-Bit Mikrodenetleyicilere Geçiş

Pek çok kişi 32-bit mikrodenetleyicileri nihai hedef olarak görse de ben günümüzde gerçek manada geliştiriciliğe atılan ilk adım olarak nitelendiriyorum. Eskiden olduğu gibi yazılımlar birkaç satır Assembly koduyla yazılmamakta, 8-bit mikrodenetleyicilerin çevrebirimleri çoğu zaman yeterli gelmemekte ve gömülü yazılımlar ile bilgisayar yazılımları arasındaki boşluk daralmaktadır. Bu noktada gelişen teknolojiye ayak uydurabilmek ve yeni teknolojiler ortaya koyabilmek için 32-bit mikrodenetleyicilere geçmek gereklidir. 32-bit mikrodenetleyicileri programlamayı öğrenmekle son bulan bir öğrenim süreci olacağını sanmamak gerekir. 32-bit mikrodenetleyicilere geçtikten sonra gelişmiş yazılım konuları karşımıza çıkacak, işletim sistemi ve ileri seviye teknolojileri öğrenmemiz gerekecektir. Örneğin benim karşımda şu an Dijital Sinyal İşleme konusu durmaktadır. İleride yapacağım arge çalışmaları için bunu öğrenmem gereklidir. Pek çok mühendislik öğrencisi ne yazık ki vizyonsuzluktan 32-bit mikrodenetleyici programlamayı "STM32F407" programlama ve ileri seviye yazılım konularını da OpenCV ile görüntü işlemeden ibaret görmektedir. Hatta bir insanın ileri seviye olup olmadığını

belirlemedeki kriter de OpenCV ile görüntü işlemeyi bilip bilmediği (!) olmaktadır. Bu tarz yönelimler asla bir geliştiricinin yolu olmamalıdır. Geliştirici bir sürüdeki koyun, bir fabrika hattındaki ürün değildir. Hepsi aynı kalıpta olmaz ve geliştirici kendini geliştirerek farklılaşır. Örneğin OpenCV ile görüntü işleme herkesin dilindeyken ondan çok daha önemlisi dijital sinyal işlemeye kimse meraklanmamaktadır.

32-bit mikrodenetleyiciler son yıllarda iyice ucuzladığı ve yazılımsal desteği arttığından dolayı mikrodenetleyici pazarında en büyük payı almıştır ve mikrodenetleyici pazarının gelişmesinde büyük rol almıştır. Günümüzde 400MHz'lere varan ve megabaytlarca belleği olan mikrodenetleyicilerden söz etmemiz mümkündür. Bir dönemin bilgisayarlarının yaptığı işleri artık mikrodenetleyiciler yapabilmektedir. Donanım kısıtlamasından dolayı yazılımsal olarak sığ olan 8-bit mikrodenetleyiciler yerine işletim sistemi, middleware ve platformları yükleyip kullanabileceğimiz aygıtlar haline gelmiştir. Örneğin .NET veya Java ortamlarını bile 32-bit mikrodenetleyicilerde kullanabiliriz. Bunun yanında pek çok işletim sistemi seçeneği, yazılım yönünden ağır basan protokoller örneğin (FAT, TCP/IP) gibi protokolleri kullanabiliriz. Bu yüzden gömülü sistemlerde bir noktadan sonra ilerlemenin tek yolu 32-bit mikrodenetleyicileri öğrenmektir. 8-bit mikrodenetleyicilerle ne kadar devam edilir bilinmez fakat şimdiden gelecek vaat etmedikleri ortadadır.

Biz burada 8-bit mikrodenetleyicileri kötülememekteyiz. Aksine kendileri oldukça iyi, basit ve etkili elemanlardır. Fakat bir noktadan sonra gerek kapasitesi, gerek hızı ve gerekse çevre birimleri ile ihtiyaçlarımıza cevap vermemektedir. Örneğin bir tarafta saniyede 15 bin ve 10-bit ADC ölçümü yapan AVR dururken öteki tarafta saniyede 5 milyon ölçüm yapan ve 16-bit çözünürlüklü modelleri bile olan STM32 durmaktadır. STM32 kullandığımız zaman yapabileceklerimizin sayısı kat kat artacaktır. Örneğin AVR'de genellikle 3 adet zamanlayıcı olduğundan dolayı frekans işlemlerinde yapabileceklerimiz kısıtlı olmaktadır. STM32'ye baktığımızda ise bazı modellerde 20'ye yakın zamanlayıcı bulunmaktadır. Dijital sinyal işleme komutları olduğu için dijital sinyal işlemeyi etkili bir şekilde yapıp pek çok işi tek bir çipe sığdırma imkanımız vardır. Üstelik bu çip ile yaptığımız devre 8-bit kullandığımız sistemden çok daha ucuza mal olmaktadır. O halde

donanım ve maliyet bakımından 32-bit mikrodnetleyicileri öğrenmemeyi gerektirecek bir sebep yoktur diyebiliriz.

32-bit mikrodnetleyicilerin öğrenilmemesinin en büyük sebebi karmaşık ve zor aygıtlar olmaları ve yeni başlayanlar için hakkında kaynak bulunmamasıdır. AVR için binlerce kütüphane, yüzlerce makale ve video yine yüzlerce örnek proje ve onlarca kitap bulunabilirken en popüler 32-bit mikrodnetleyici STM32 hakkında bunun onda birini bile göremeyiz. O yüzden 32-bit mikrodnetleyiciler üzerinde çalışmadan önce kendimizi buna hazırlamalıyız ve kendi kendimize öğrenebilir seviyeye gelmeliyiz. 8-bit ve 32-bit arasındaki fark bir yandan da Arduino ve AVR arasındaki fark gibidir. 8-bit ile az bilgi ve emekle çok iş yapabilirken 32-bitte çok bilgi ve emek gereklidir. Fakat 8-bitte çok iş yapsak da yapabileceğimiz işler sınırlıdır. Aynı AVR programlarken pek çok şey yaparken Arduino'da pek bir şey yapamamak gibidir.

32-bit mikrodnetleyicilere geçmeden önce 8-bit mikrodnetleyiciler, dijital elektronik, C programlama dili ve mikroşlemci mimarisi iyi öğrenilmek zorundadır. Eğer bunları yeteri kadar bilemezseniz çok zorlanırsınız hatta bazen hiçbir şey anlamayabilirsiniz. Üstelik 32-bit mikrodnetleyicilere nereden başlayacağını bilememek de büyük bir meseledir. Bu alanda bazı kitapları incelesen de datasheetin özetinden ibaret olduğunu gördüm. Bunu okuyup anlayabilen adam zaten datasheette çok daha ayrıntılı bilgiyi bulabilir o halde neden kitap okumaya ihtiyacı olsun? Biz STM32 makalelerini yazarken bunu dikkate aldık ve olabilecek en anlaşılır ve mantığı kavramaya yönelik eğitimi hazırlamaya gayet ettik. Şu an STM32 makalelerinin sayısı 45 olup sizin mantığı kavramanıza ve başlangıç yapmanıza yeterlidir. Konuların hepsini şu an anlatmadık ve fazla uygulama yapmadık. Buna işlerimden dolayı ara vermek zorunda kaldım ileride devam edeceğim.

32-bit mikrodnetleyicilere geçmek isteyenlerin geçip geçemeyeceğini anlaması için sitemizde yer alan STM32 kategorisindeki makaleleri sırasıyla okumasını tavsiye ediyoruz. Eğer mantığını öğrendiler ve datasheet de okuyabiliyorlarsa 32-bitte geçmekte gecikmemeleri gereklidir. 32-bit hakkında fazla kaynak ve örnek olmadığı için Datasheet okumak en önemli kriterlerden biri olmaktadır. Gömülü sistem rehberi yazı dizimizi burada

bitiriyoruz. Biz yeteri kadar konudan bahsettiğimize inanıyoruz. İlerleyen yazılarda daha farklı konularla karşınıza çıkacağız.