

ASANSÖR KUMANDA SİSTEMLERİNDE CANOPEN HABERLEŞME PROTOKOLÜ

Serhat Ayaz

Mik-El Elektronik
serhat.ayaz@mik-el.com

ÖZET

CAN protokolü asansör uygulamalarında çok uzun yıllardır kullanılmaktadır. 2002 yılı ile birlikte orta ölçekli firmalar, cihazlarına tak ve çalıştır uyumluluğunu kazandıracak daha yüksek katmanlı bir protokol olan CanOpen (Açık CanBus Protokolü) standardına geçmeyi tartışmaya başladılar. CanOpen protokolünün cihazlara kazandırdığı tak ve çalıştır yeteneği sayesinde piyasalardaki farklı firmaların ürünleri birbirleri ile haberleşebilir hale geldi. Bu standartlaştırma çalışmaları sonucunda ikiyüzlü duraklı ve sekiz asansörlü sistemlere kadar CanOpen uygulamaları gerçekleştirilebilir oldu. CiA (Can in Automation) olarak adlandırılan üreticiler ve kullanıcılar birliği CanOpen protokolünün sorumluluğunu üstlenmiştir. CiA ile birlikte kurulan SIG (Special Interest Group Lift Control Systems), CiA profillerini kontrol edip asansörler için kullanılacak özellikleri bulup, bunların genişletilmelerini veya modifiye edilmelerini sağladılar. 2002 yılında da CiA 417 isimli yeni profil ile asansör kontrol sistemlerine uygunluk oluşturulmuş oldu.

1. CAN VE CANOPEN TARİHÇESİ

Robert Bosch, 1986 yılının Şubat ayında, CAN (Controller Area Network) seri veriyolu sistemini SAE (Society of Automotive Engineers) Detroit kongresinde tanıtmıştı. 1987 ortaları itibarı ile de Intel, Philips gibi yarıiletken üreticileri CAN çiplerini piyasaya sunmaya başladılar. Böylelikle başta otomobiller olmak üzere her türlü taşıtlardan sanayi ürünlerine kadar birçok alanda CAN en yaygın kullanılan veriyolu protokolü haline geldi.

CAN otomobiller için geliştirilmiş bir veriyolu protokolü olsa da, ilk uygulamalar kullanıcılar tarafından daha farklı alanlarda yapıldı. Özellikle Kuzey Avrupa'da ilk günden beri çok popüler bir hale geldi ve 1992 başlarında kullanıcılar ve üreticiler bir araya gelerek CiA (CAN in Automation) Uluslararası Kullanıcılar ve Üreticiler Birliğini kurdular. CiA'nın ilk işi CAL (CAN Application Layer) sınıflandırması oldu. Bu her ne kadar endüstriyel uygulamaları mümkün kılan bir yaklaşım olsa da, her bir kullanıcıya yeni bir profil oluşturma zorunluluğu getiriyordu. 1993 te, daha sonra adı CANopen olacak olan CAL tabanlı, ürün hücreleri arasında haberleşme sistemi kurmaya imkan tanıyan bir prototip geliştirildi. 1995 te CiA tamamen revize edilmiş CANopen profilini duyurdu. CANopen, mümkün olduğunca farklı aygıt, arayüz ve uygulama profillerinden oluşan programlanabilir sistemler için bir veriyolu protokolü çerçevesi tanımlar. Bu sayede, 1990 sonlarında tüm endüstri dallarında CANopen kullanımı düşünülür hale gelmişti.

CiA katılımcıları birkaç farklı uygulama profili tanımladılar. Her bir uygulama profili, belirli uygulama alanı içerisinde kullanılan tüm cihazların arayüzlerini belirler. Asansör kontrol sistemleri de ilk hayata geçen CANopen uygulama profillerinden biri olmuştur. CiA ile birlikte kurulan SIG (Special Interest Group Lift Control Systems), CiA profillerini kontrol edip asansörler için kullanılacak özellikleri bulup, bunların genişletilmelerini veya modifiye edilmelerini sağladılar. 2002 yılında da CiA 417 isimli yeni profil ile asansör kontrol sistemlerine uygunluk oluşturulmuş oldu.

2. GÖMÜLÜ AĞLAR

Bilgisayarların yaygınlaşmaya başlamasından beri yarı iletkenler konusunda her geçen gün daha gelişmiş teknolojiler ortaya çıkmıştır ve en büyük ilgide kuşkusuz mikroişlemci ve hafıza çiplerine yöneliktir. Her teknolojik gelişim sonucu daha küçük çiplerin içine daha hızlı işlemciler ve daha yüksek hafıza kapasitesi sığmaya başlamıştır. Kuşkusuz, bunun sonucunda da daha ucuz maliyetlere daha yüksek performans elde edilebilir oldu.

Teknolojinin hızla ilerlemesi ve mikro işlemcilerin ucuzlaması ile birlikte akıllı elektronik devreler her geçen gün birçok ürünün içerisinde yer almaya başladı. Ucuz mikro işlemciler sayesinde üreticiler ürün sistemlerini akıllı elektronik devrelerle donatmaya başladılar ve bu da mikro işlemciler arasında haberleşme ihtiyacını doğurdu. Böylece mikro işlemcilerin gömüldüğü sistemler içerisinde gömülü haberleşme ağları kuruldu. Asansör kontrol sistemleri de bu gelişimden nasibini almış önemli bir sektörel gelişime tanıklık ettiler.

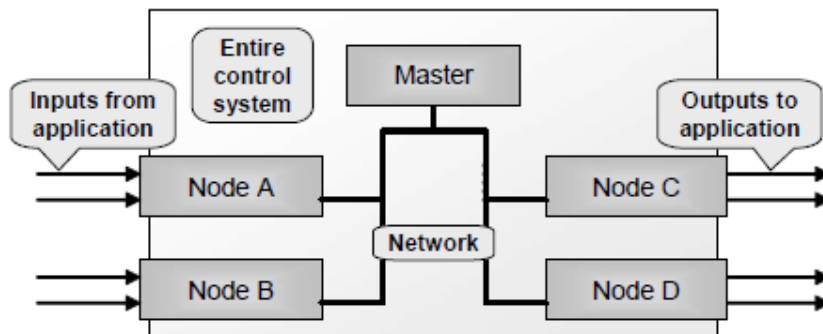
Endüstriyel otomasyon uygulamalarında ve de bizim sektörümüz olan asansör sistemlerinde, teknolojinin ilerlemesi, güvenlik ihtiyaçlarının artması, daha kolay kurulum-devreye alma-kullanım ihtiyaçlarının gelişmesi, daha konforlu ve kararlı çalışma ihtiyaçlarının doğması ve benzeri sebepler sonucunda, birçok akıllı kartlardan oluşan gömülü sistemler uygulamaya geçti. Bunun doğal sonucu olarak ta gömülü ağlar sistem içerisinde kurulmuş oldu. Gömülü ağlar kurulurken sistemin gereksinimleri ve büyüklüğüne göre haberleşme ağının oluşturulmasında kullanılacak teknik de belirlenmelidir. Güncel haberleşme ağı tekniklerine Ethernet, Profibus, Modbus, CAN ve benzeri teknikler örnek verilebilir. Asansör sektöründe kabul gören teknik kuşkusuz CAN olmuştur. Zaten CAN, teknik yapısı gereği 8-16-32 bit işlemcilerden oluşan sistemlerde kullanılmaya daha uygun düşük seviye bir haberleşme tekniğidir.

3. CAN VEYA CANOPEN KULLANILAN GÖMÜLÜ AĞLARDA BİLİNMESİ GEREKENLER

3.1. Giriş ve Çıkış

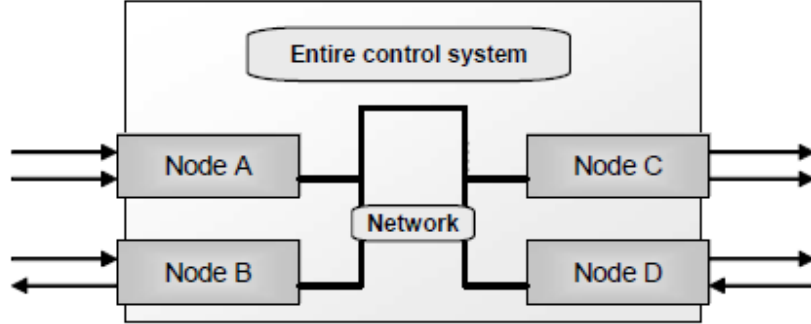
Tüm kontrol sistemleri ilgili oldukları uygulama ile bağlantı kurabilmek için giriş ve çıkış bilgilerine ihtiyaç duyarlar. Bu giriş ve çıkış noktaları sistem gömülü ağında bulunan haberleşme devreleri ile bağlantılı olarak çalışabilir. Böylece giriş birimleri okudukları sinyalleri seri haberleşme yolunu kullanarak varsa kontrol birimine veya direk olarak çıkış birimine gönderebilirler. Çıkış birimleri de kontrol biriminden veya giriş biriminden seri haberleşme yolu ile gelen bilgileri uygulama birimlerine iletmiş olur.

Şekil 1’de gösterilen geleneksel kontrol sisteminde iki adet giriş ve iki adet çıkış birimlerine sahip bir kontrol sistemi vardır. Mümkün olduğunca uzakta konumlandırılmış ana kontrol birimi giriş birimlerinden gelen bilgilere göre hesapladığı çıkış bilgilerini, çıkış birimine gönderir.



Şekil 1. Geleneksel Kontrol Yönteminde Giriş ve Çıkış Birimleri

Birçok gömülü ağlarda ise uygulama çok daha karmaşık hale gelmiştir. Bu da daha akıllı giriş ve çıkış birimleri barındıran ve ana kontrol sistemi kullanmayan gömülü ağların oluşturulmasına sebep olmuştur. Ana kontrol biriminin yaptığı görevler diğer birimler arasında dağıtılmış olur. Şekil 2 bu sistem oluşumunu ifade eder.



Şekil 2. Yenilikçi Kontrol Yönteminde Giriş Ve Çıkış Birimleri

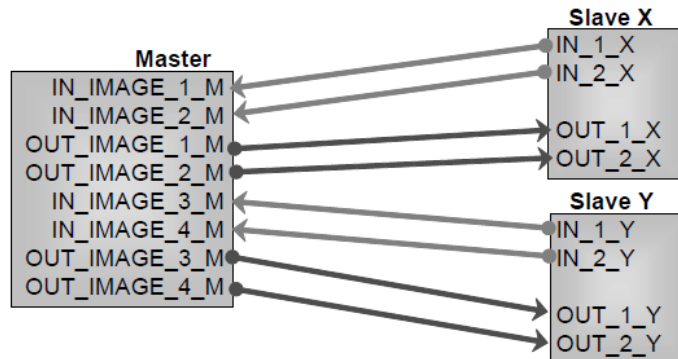
Bu yöntemde giriş birimi okuduğu giriş sinyallerine göre, hangi çıkış biriminin çıkışlarını ne zaman ve nasıl süreceği bilgisini oluşturur ve bu verileri direk olarak ilgili çıkış birimine gönderir. Bu sistemlerde bazı birimlerin hem giriş hem de çıkış olarak kullanılması zorunluluğu karışıklığa sebep olabilir.

Genel olarak giriş ve çıkışların uygulamaya ait giriş ve çıkışlar olduğunu anlamak önemlidir. Haberleşme ağına ait giriş ve çıkışlar değildir. Arada mutlaka akıllı veya yetisiz haberleşme birimleri mevcuttur.

3.2. Master/Slave

Bir ağda master/slave cihazların olması, master olan cihazın slave olan cihaz üstünde çeşitli kontrol fonksiyonlarına sahip olması anlamını taşır. Tipik olarak, master birim sürekli olarak ağı tarar ve ağı katılmış veya ayrılmış slave birimleri belirler, birimler için gerekli konfigürasyonları yapar. Belki ek olarak, tüm ağı veya bazı slave birimleri kapatma, resetleme işlemlerini sağlar.

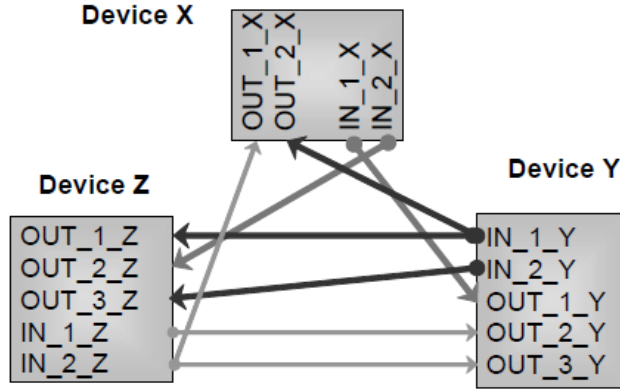
Şekil 3'de master birimin ağ haberleşmesini sürdürdüğü bir master/slave haberleşme ağı gösterilmektedir.



Şekil 3. Master/Slave Haberleşme Modeli

Bu modelde slave birimler birbirleri ile direkt olarak haberleşemezler. Sadece master birim haberleşmeyi başlatabilir ve slave birimler master birimin onlardan bir veri isteği olduğunda sadece master birime veri gönderirler.

Şekil 4 ise master birim olmaksızın oluşturulmuş bir haberleşme modelini ifade eder. Bu modelde tüm birimler birbirleri ile veri alışverişinde bulunabilir. Direk haberleşme modeli, veri yolundaki yoğunluk mümkün olduğunca az tutulabildiği sürece daha etkin bir yöntemdir. Tek bir mesaj paketi ile verinin bir cihazdan diğerine gönderimini mümkün kılmaktadır. Oysa master/slave haberleşme modelinde slave birimden giriş bilgileri master birime iletilir ve master birim çıkış birimine ilgili veriyi gönderir. Bu da en az iki mesaj paketinin hattı meşgul etmesi anlamına gelmektedir.

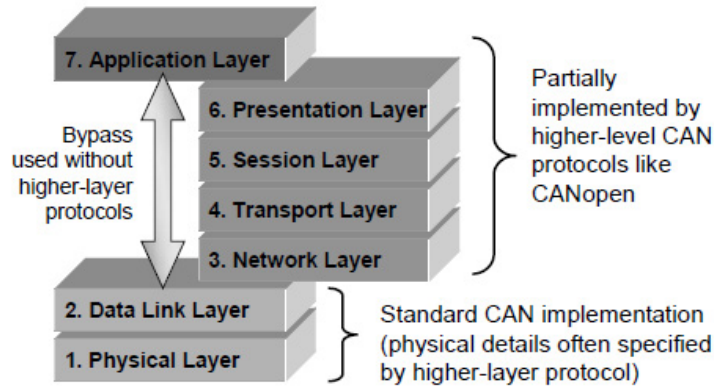


Şekil 4. Direkt Haberleşme Modeli

3.3. ISO 7-Katman Kaynak Modeli

Standart haberleşme ağı modeli, fiziksel ortamdan uygulama alanına kadar 7 katmandan oluşan bir ISO katman modelidir. Genelde çipler arası haberleşme uygulamaları sadece tek katmanlı (fiziksel katman) olsada, CAN uygulamalarında ilk iki katman kullanılır (fiziksel+veri gönderim). Daha üst katmanların kullanımı yazılım ile uygulanabilir.

Geleneksel yedi katmanlı haberleşme ağına sahip uygulamalarda, gömülü uygulamalar için kabul edilemez olan hat yüklenmelerini engelleyecek, herhangi iki katman arasından her birine geçişte kullanılan bir ara yüze ihtiyaç duyulur. Bu nedenle, haberleşme hattında oluşacak taşmaları en aza indirmek için, yüksek katmanlı CAN protokolleri sadece yüksek katmanlardan seçilmiş uygulamaları gerçekler.



Şekil 5. ISO 7-Katman Referans Modeli

CAN ile fiziksel ve veri hattı katmanlarının birçok parçası yerine getirilir ve bu durumda standartlaştırılmış ortak bir yazılımsal ara yüze gerek duyulmaz. Bazı uygulamalar son katman olan uygulama katmanını ikinci katmanın üzerine direk olarak oturtur. CANopen ile her katmanın parçaları uygulamaya konulur.

4. CANOPEN'A GİRİŞ

4.1. Nesnel Sözlük Kavramı

Her bir CANopen biriminin çekirdeği, onaltı bit dizin ve sekiz bit alt dizinli bir arama tablosundan oluşan nesnel sözlükten (Object Dictionary (OD)) meydana gelir. Böylece her dizinde iki yüz elliye kadar alt girdiye izin verilmiş olur. Her girdi herhangi tür ve uzunlukta bir değişken olabilir.

Tüm işlem ve haberleşme ilişkili veriler girdiler olarak önceden tanımlanmış nesnel sözlük alanlarında depolanır. Kullanım dışı girdiler gerçekleştirilmemiş olur.

Nesnel sözlük sadece, değişkenleri bir dizin veya alt dizin değeri ile ilişkilendirmek için kullanılan bir yol değildir, aynı zamanda tanım tablosunda bir veri türünü de belirler. CANopen desteklenen veri türleri listesine eklenecek uygulama özelindeki veri türlerini de destekler.

Belirtildiği gibi, nesnel sözlük temel ağ gereksinimleri olan ağ birimleri içerisindeki değişkenlerin yerleşimini ve tür tanımlamalarını mümkün kılmayı sağlar.

4.2. Aygıt Profilleri

Nesnel sözlük kavramı iletişimin ihtiyaç duyduğu veri yapılandırmasını sağlasa da, sözlükteki girdilerin ne amaçla kullanılacağı bilgisi eksiktir. Sözlük, master aygıtın bir tahmin yürütmesine izin vermek ve desteklenip desteklenmediğini görmek adına sözlüğün belirli alanlarına basitçe bağlanmasını denemek için çok büyüktür.

Çözüm basittir. İlk olarak, tüm CANopen birimlerinin desteklemek zorunda olduğu birkaç zorunlu girdi vardır. Bunlar birimin kendisini tanıtmayı sağlayan bir nesne kimliği ve muhtemel hata durumlarını raporlayan bir hata kodu içerirler. Aygıt profilleri, tüm haberleşme parametrelerini tanımlayan ve desteklenen belirli CANopen modüllerinin nesne sözlüğü girdilerinden oluşmuş ek özelliklerdir.

Master veya konfigüre aracı, bir servis veri nesnesi (Service Data Object (SDO)) kullanan slave birimin kimlik nesnesini okuyabilir veya ona bağlanabilir. Cevap olarak, hangi aygıt profiline uygun bir modül olduğu hakkında bilgi içeren bir SDO alır. Özel bir aygıt profili için, master aygıtın hangi nesne girdilerinin tanımlanmış olduğunu bildiğini varsayarsak, bu durumda master hangi nesnel sözlüğün desteklendiğini bilmiş olur ve onlara direkt olarak bağlanır.

4.3. Elektronik Bilgi Sayfaları

Elektronik bilgi sayfaları (Electronic Data Sheets(EDS)) desteklenen nesnel sözlük girdilerinin belirlemenin standartlaştırılmış yolunu sunar. Herhangi bir CANopen modül üreticisi modül ile birlikte bu sayfalardan gönderir. Sisteme bağlı bir CANopen master veya konfigüre aleti geçerli aygıtların ayarlarına EDS yi direkt olarak yükleyebilir. Ağda bağlı bir cihaz bulunduğunda, master veya konfigüre aleti ilk önce eşleşen bir EDS bulmaya çalışır. Bir kez bulunca, tüm nesnel sözlük girdileri master veya konfigüre aleti tarafından bilinmiş olur.

4.4. Nesnel Sözlüğe Erişim

Doğrudan bir iletişim kanalı kurmak master ve slave cihazlar açısından kaçınılmaz bir gerekliliktir. Master veya konfigüratör aygıt, ağa bağlı bulunan tüm birimlerin nesnel sözlük girdilerini yazabilir ve okuyabilir olmak durumundadır.

CANopen, basit bir istemci-sunucu ilişkili iletişim methodunu destekler. Böylece birimlerin nesnel sözlüklerine okuma veya yazma istemlerini iletmesine izin verecek bir doğrudan bağlantı sağlanabilir. Nesnel sözlüğe giden istemler veya nesnel sözlükten gelen cevaplar içeren mesaj paketleri servis veri nesneleri (SDO) olarak adlandırılır.

Şunu not etmek gerekir ki, sistemde, SDO iletişim modunu aktif olarak başlatacak hakka sahip olacak bir birim tanımlanmış olmalıdır. Bu da tipik olarak master birimin bir çeşidi olur.

Tanımlı SDO haberleşmesi master birim kontrollü bir haberleşme istemcisi gibi çalışır. Master birim tüm SDO iletişim kanallarını kurar ve sistemdeki her bir birime ulaşabilecek bir kanala sahip olur.

SDO lar bölünmüş transfer olarak adlandırılabilir transfer biçimlerini destekler. Böylece herhangi büyüklükteki nesnel sözlük girdileri transfer edilebilir olmuş olur. İçerik tek bir mesaj paketine sığacak kadar değilse, bu durumda içerik otomatik olarak birden fazla paketlere bölünerek ayrı ayrı mesaj paketleri halinde gönderilir.

SDO metodolojisi, ana sürücü biriminin ağa bağlanmış olan tüm birimlerin tüm nesnel sözlük girdilerine okuma veya yazma maksatlı bağlanmasına izin verir. İşletim ve konfigürasyon verileri nesnel sözlüklerin bir parçasıdır ve işletim dataları SDO transferi kullanılarak güncellenebilir hale gelmiş olur.

Bununla birlikte, bu yöntem çok verimli bir yöntem değildir. İlk olarak, tüm giriş ve çıkışların master birim tarafından idare edildiği bir sıralama şeması uygulanmak zorundadır. İkinci olarak ağa bir çok mesaj yüklemesi olacaktır. Bir girişin okunup çıkışa yazılabilmesi için en azından dört mesaj paketi yollanacaktır.

1. Master birim giriş birimine bir okuma istek SDO paketi yollar.
2. Giriş birimi bir SDO ile birlikte istenen veriyi master birime geri yollar.
3. Master birimi çıkış birimine bir yazma istek SDO paketi yollar.
4. Çıkış birimi bir SDO ile birlikte çıkış yazma işlemini doğrular.

Üçüncü olarak ise, sadece bir byte veri yükü olmasına rağmen bile her bir SDO mesaj paketi sekiz byte lık veriden oluşur.

Özet olarak, SDO herhangi bir birimin nesnel sözlük girdilerine erişmeyi sağlayan temel bir bağlantı yöntemi sağlar. Ancak, daha sağlam bir veri iletişimi için daha etkin bir yöntem gereklilik vardır.

Konfigürasyon verileri SDO bağlantısı ile elde edilebileceğinden dolayı, SDO lar tak-çalıştır gereksinimlerini yerine getirebilir. Enkoder gibi özel bir I/O birimine ihtiyaç duyan bir sistem kurucusu enkoder aygıt profiline uygun herhangi bir ürünü seçebilir. Sonrasında uygulamacının kendisi CANopen konfigürasyonunu veya master cihaz yazılımını kullanarak, uygulamanın özelliğine uygun bir şekilde iletişim aksiyonlarını gerçekleştirilmesi için bu birimi konfigüre eder.

4.5. PDO İle Performansın Artırılması

Bir çok uygulamada gerçek işlem verilerinin değişimini başarabilmek için SDO verimli bir metod olarak kullanılamaz. Hat yüklenmesi çok fazladır ve mesaj tetikleme metodu son derece kısıtlıdır.

CAN çoklu master birimi iletişim konseptini desteklese de, yüksek öncelikli veri işleme bağlantısı ve daha fazla etkinliği sağlayabilecek bir doğrudan iletişim metoduna ihtiyaç vardır.

İşlem veri nesnesi (Process Data Object), tek bir sekiz byte lık CAN mesaj paketi içerisine nesne sözlüğünden gelen birden fazla işlem veri değişkenini yerleştirebilmek için optimize edilmiş bir çözüm ortaya koyar.

4.6. Ağ Yönetimi (NMT)

CANopen bir master ağ yönetim birimine, ağda bulunan tüm aygıtları parametre değerlerine bağlı olarak çalıştıklarını izlemesi için izin verir. Herhangi bir birim hataya düşmüşse veya belirli bir alarm mesajı vermişse uygun bir kurtarma veya çalışmayı durdurma prosedürünü başlatabilir.

Burada bu birimler yönetiminin nasıl uygulamaya konulacağı noktasında birçok alternatif vardır. En yaygın olanı kalp vuruşu (heartbeat) mesajının kullanılması olmuştur. Bu da bir ağ yönetimi olmasa bile her bir birimi yönetmeye izin verir.

Her bir birim işletimde olduğu sürece belirli aralıklarla ağa bir kalp vuruşu mesajı atar. Eğer tüm birimler kalp vuruşunu işletirse, her bir birim iletişim partneri olduğu birimin tüm kalp vuruşlarını izleyebilir. Kalp vuruşu sisteminin kullanılması sayesinde, birimlerden bir PDO mesajı gelmediği durumlarda bile tüm birimler en azından iletişim partnerlerinin hala çalışır vaziyette olduğunu bilmiş olurlar.

5. SONUÇ

CANopen haberleşme modeli, cihazlara tak-çalıştır uyumluluğunu getirmiştir. Dolayısı asansör sistemlerinde de ürün bağımlılığını kırmak adına kendisine önemli bir yer edinmiştir. Kat butonyer ve gösterge kartlarından enkoderlere kadar pek çok gömülü sisteme sahip ürünler bu seri haberleşme standardı sayesinde aynı dili konuşabilir hale gelmiştir. Bu da firmalardan bağımsız olarak her firmanın farklı ürünlerinin bir arada kullanılması ile toplam bir sistemin oluşturulabilmesine olanak sağlar. Ülkemizde şu anda kendisine yer bulamasa da bu sistem özellikle Avrupa ülkeleri arasında yaygınlaşmıştır ve yakın bir gelecekte şüphesiz ülkemizde de kendisine yer bulacaktır.

KAYNAKLAR

- [1] **Olaf Pfeiffer, Andrew Ayre and Christian Keydel**, Embedded Networking With CAN And CANopen, Copperhill Technologies Corporation, 2003.
- [2] Application Profile For Lift Control Systems, CAN in Automation (CiA) e. V., 2011.
- [3] CiA, www.can-cia.org