

**İSTANBUL TEKNİK ÜNİVERSİTESİ
ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

**PROFINET CBA PROTOKOLÜ ÜZERİNDE
ENDÜSTRİYEL HABERLEŞME UYGULAMASI**

Bitirme Ödevi

**Dođan Fennibay
040010404**

**Bölüm : Bilgisayar Mühendisliđi
Anabilim Dalı: Bilgisayar Bilimleri**

Danışman : Yrd. Doç. Dr. Feza BUZLUCA

Ocak 2006

Özgünlük Bildirisi

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

İstanbul, Ocak 2006

Doğan Fennibay

Bu alıřmada bana yol gstermekten geri durmayan deęerli hocam Yrd. Do. Dr. Feza Buzluca'ya, alıřma sırasında maddi ve manevi destek sunan Siemens PSE TR İstanbul grubuna, daima yanıma olan arkadaşlarım ve aileme teřekkürü bir bor bilirim.

Bu alıřma Ziya Fennibay'a ithaf edilmiřtir.

Doęan Fennibay
Ocak 2006

PROFINET CBA PROTOKOLÜ ÜZERİNDE ENDÜSTRİYEL HABERLEŞME UYGULAMASI

(ÖZET)

Bu projede Profinet CBA üzerinde Profinet CBA (Component Based Automation, Bileşen Temelli Otomasyon) ve RS-232 (eski bir seri iletişim standardı) arasında bir köprü ve tanı bilgisi vermek üzere standart bir web sunucuyu içeren bir endüstriyel haberleşme uygulaması yapılmıştır. RS-232 – Profinet CBA köprüsünün geliştirilmesinde hedef; Profinet'in dezavantajı olan yeni olması ve dolayısıyla yaygın olmamasından kaynaklanan eski cihazlarla beraber çalışabilme zorunluluğuna yöneliktir. Web sunucu uygulaması ise Profinet'in tek bağlantı üzerinden birçok işlev sağlayabilme avantajını öne çıkarma amaçlıdır.

Profinet CBA, Profibus ve ardından Profinet IO'nun (Input/Output, Giriş/Çıkış) temsil ettiği DP (Decentralized Periphery, Dağıtık Çevreirimleri) sistemlerinin ötesinde, bir dağıtık işleme (distributed processing) sistemidir. Profinet CBA'de bir endüstriyel otomasyon sisteminin bileşenleri parça parça yaratılır. Her bileşen kendine ait elektrik, mekanik aksam ile kontrol yazılımını içerir. Böylece yeni otomasyon sistemlerinin yapılandırılmasında program yazma ihtiyacı büyük oranda elimine edilir.

Kullanılan diğer teknolojilere gelirsek. RS-232 oldukça eski ve yaygın bir asenkron seri iletişim protokolüdür. CGI dinamik web programlamanın temelini oluşturmuş, web sunucusuyla dinamik içerik sağlayan proses arasında haberleşmeyi tanımlayan bir standarttır. UNIX Etki Alanı Soketleri ise yerel, prosesler arası haberleşmede kullanılan standart soket kütüphanesiyle erişilebilen bir sistemdir.

Projede kullanılan çalışma ortamı PowerPC mimarili Kontron EB8245 endüstriyel bilgisayar üzerinde koşan Linux 2.4.18 işletim sistemidir. Geliştirme ortamı ise i386 mimarisinde çalışan Linux 2.6.4 çekirdekli SuSe 9.1 işletim sistemidir. Program i386 üzerinde de denendiğinden "little-endian" sistemlerde de sorunsuz çalışacağı söylenebilir.

Probleme çözüm olarak üç prosesli – Profinet CBA, seri port sunucusu ve web uygulaması – bir yapı önerilmiştir. Bu prosesler aralarında UNIX mesajlarıyla, istek-cevap esasına göre haberleşmektedirler.

Bu altyapı kurulduktan sonra, üzerinde PLC ile Profinet üzerinden haberleşmeyi içeren örnek bir senaryo uygulanmıştır. Bu senaryoda PLC; RS-232 arayüzlü barkod okuyucudan birden fazla defa aynı barkodun gelmesi durumunda işaret vermektedir.

Bu rapor; probleme bir giriş yaptıktan sonra proje tanımı ve kapsamını incelemektedir. Arından konuyla ilgili kuramsal bilgiler verilmekte, özellikle Profinet CBA ayrıntılı olarak incelenmektedir. Daha sonra problemin analizi yapılmakta ve çözüm olarak yukarıda bahsi geçen üç prosesli yapı modellenmektedir. İlerleyen sayfalarda tasarım ayrıntılarını tanımlanıp gerçekleştirilmede neler yapıldığından bahsedilmektedir. Son olarak, sistemden elde edilen deneysel sonuçlar yorumlandıktan sonra sistemin yeni teknolojiler, maliyet ve pazar açısından bir değerlendirmesi sunulup geliştirme önerileri verilmeye çalışılmaktadır.

AN INDUSTRIAL COMMUNICATION APPLICATION ON PROFINET CBA PROTOCOL

(SUMMARY)

In this project, an industrial application of Profinet CBA (Component Based Automation) was developed which consists of a bridge between Profinet CBA and RS-232 (a legacy serial communications standard) and a web server serving diagnostic information about the system. There are two objectives of this requirement. First of all, unlike Profibus, Profinet is not widely used yet, which means that it should be interoperable with other communication methods. The development of the RS-232 – Profinet CBA addresses this problem. The other objective of the project is to show the advantage of enjoying more than one function over exactly one connection. Diagnostic web server is meant to address this.

Profinet CBA is a descendant of the widely common Profibus communication protocol. Profibus, which has over 10 million nodes over the world, is the most widely used DP (decentralized periphery) system. As an industrial communication bus system, it works deterministically and data refresh periods of 1 ms can be reached with Profibus. This means that Profibus can be used for motion control, too. Profibus works on RS-485 or less commonly MBP (Manchester Bit Encoding).

Profinet IO (Input Output) is Profibus's successor in the DP area. However Profinet CBA which employs Microsoft's DCOM technology in implementing remote objects stands for something farther in the field of distributed processing. It envisions components in an industrial automation system. The mechanical and electrical parts plus the control program are encapsulated in a component. Each component can be built separately, tested separately and shipped separately. So these components can be reused in different configurations and programming during plant commissioning is largely eliminated.

Profinet (IO and CBA) uses industrial Ethernet as transmission medium in contrast to Profibus. Ethernet is a non-deterministic bus because of the randomness of its exponential back-off algorithm in CSMA/CD. Profinet overcomes this non-determinism in three phases. First phase, NRT (Non Real Time) uses only Ethernet switches instead of hubs to avoid collisions. Second phase, SRT (Soft Real Time) uses smaller Profinet Ethernet frames (EtherType = 0x8892) which do not pass through standard TCP/IP layers and thereby gain a performance increase. The most deterministic way of Profinet communication is IRT (Isochronous Real Time) which involves Ethernet interface cards integrated with switches to closely control the communication. With IRT, data refresh periods of 1 ms can be reached, which is useful for motion control. However, IRT is only available in Profinet IO. Profinet CBA can use NRT and SRT, in this project only NRT side was implemented.

RS-232 is a very old communication protocol defining a specific type of asynchronous serial data communication. It defines mainly the electrical characteristics of the signal. Other parameters of communication (number of data and stop bits, parity, flow control, speed) are chosen freely. However, there are some standards about the values these

parameters can take. In Linux, access to serial port is provided through `termios` library and `/dev/ttySx` device files, where `x` is the number of the serial port.

Common Gateway Interface (CGI) is a standard developed by National Center of Supercomputing Applications (NCSA) in 1993. It defines a way of web servers to communicate with other processes in order to get dynamic data to serve to the clients. CGI employs standard input and output channels of a process to accomplish this, namely: command line parameters, environment variables and `stdin` are used to send data to the process where process uses `stdout` to send data back to the web server.

In this study GoAhead's embedded web server was used which provides traditional CGI support. On the CGI application's side, GNU's object-oriented `cgicc` library was employed.

The solution developed in this project consists of three processes working together: serial port server, Profinet CBA and web application. For synchronization among these, UNIX Domain Sockets were employed. In this choice, the facts that UNIX Domain Sockets use the standard socket library, provide little overhead and guarantee ordered message-oriented delivery were taken into account.

Execution environment is a Linux 2.4.18 system running on Kontron's evaluation board EB8245, which contains a PowerPC MC8540 CPU. Linux 2.6.4 on i386 architecture with `gcc 2.95` cross-compiler for PowerPC was used as the development environment. Code was tested on i386 platform as well as PowerPC, it is safe to say that program can run on little-endian systems, too.

As a solution to the problem proposed above, a structure with three processes synchronizing with each other by messages were developed. Each message transaction is designed as a request (beginning with "REQ...") and a response ("RSP...") to the request. Requests are further categorized into control ("REQ CTRL...") and data ("REQ DATA...") commands. Responses then include the type of their content, such as "RSP DATA...", "RSP CONF..." or "RSP OUTPUT...". A common request for all processes is "RSP CTRL STATUS" which checks for process's availability. Response to this request should be "RSP OK" in normal conditions.

To implement this messaging infrastructure two classes were designed: one for representing a UNIX messaging endpoint (`unix_socket`) and one for representing a message (`unix_message`) which also holds sender's information in itself.

Profinet CBA application's development consisted of three main steps. First of these was the porting of Profinet CBA code to the Linux environment. This was done according to the Profinet CBA Implementation Guide [6] which divides the code stack into modules for easy porting: base module, remote procedure call module, DCOM module and auto marshaler module.

Second step was the development of the Profinet CBA application on top of auto marshaler module. This was done by modifying Windows port of the Profinet CBA according to Linux's needs and application specifications.

Third step was the integration of IPC mechanism with Profinet CBA code. To accomplish this, standard output which received all error and information messages was redirected to an in-process pipe which was read on output requests coming through UNIX Domain Sockets.

To implement serial port server, namely SerialServer, two classes were developed. serial is a class to wrap termios functions and kernel's standard I/O functions in an object-oriented way. serialbuf deriving from serial is another class which holds last read/written data from/to the port in order to send it to the requester without interfering with the port. SerialServer, is then merely a user of this serialbuf class, which it uses on requests coming through UNIX Domain Sockets.

This structure, namely processes executing on requests as UNIX messages is common in Profinet CBA and SerialServer. Additionally, Profinet CBA runs also on requests from Profinet peers and sends requests to SerialServer when needed.

Web application processes however, do not serve to these requests. They are only requesters in this schema who are invoked by the web server on HTTP requests.

Finally, after this infrastructure for Profinet CBA and HTTP server is developed, a simple example scenario employing the infrastructure is designed and implemented. In this scenario; a barcode reader connected to Kontron's board over RS-232 sends read barcodes to the Profinet CBA interface. Then, a Profinet CBA capable PLC reads this data and checks its internal database for a match. It informs the user about whether it has found given barcode data in the database and adds the barcode to the database in case of not found. Thereby a behavior of informing the user on repeated barcode readings is implemented. Naturally, this is only an example, any example requiring an RS-232 interface can be implemented.

This graduation assignment report you are reading starts with an introduction to the problem to be solved. It continues with the project description and a plan about timing and resources is given.

Afterwards, detailed theoretical information gained about particularly Profinet, serial communication over RS-232, web programming over CGI and UNIX domain sockets are given to the reader.

Later on, a detailed analysis of the problem is made and the proposed solution mentioned above is modeled. Then, this model's detailed design is made and information about the implementation is given.

Finally, experimental results of the system are commented on and a general evaluation of developed system concerning new technologies, cost and market is made. Thoughts about the project and its execution and improvement suggestions are given.

İÇİNDEKİLER

Profinet CBA Protokolü üzerinde Endüstriyel Haberleşme Uygulaması.....	I
An Industrial Communication Application on Profinet CBA Protocol.....	III
İçindekiler	VI
1. Giriş	1
2. Projenin Tanımı ve Plan	2
1. Proje Tanımı	2
2. İş Planı	2
3. Gerekli kaynaklar	3
3. Kuramsal Bilgiler.....	4
1. Profinet CBA	4
1.1. Profinet CBA Nesne Modeli.....	4
1.1.1. İşletme Modeli	5
1.1.2. Mühendislik Sistemi Modeli.....	5
1.2. Yazılım Mimarisi.....	6
1.3. İletişim Yapısı.....	7
1.4. DCOM	9
2. Seri İletişim ve RS-232.....	10
2.1. Elektriksel Karakteristikler	11
2.3. Çerçeve Yapısı.....	11
2.4. Linux Arayüzü	12
3. Dinamik Web Programlama ve CGI.....	13
3.1. GNU cgicc	14
3.2. GoAhead Web Sunucusu	15
4. UNIX Etki Alanı Soketleri	15
4. Analiz ve Modelleme.....	17
1. Analiz.....	17
2. Modelleme	18
5. Tasarım, Gerçekleme ve Test	20
1. Tasarım	20
1.1. IPC Mekanizması.....	20
1.2. Seri port sunucusu (SerialServer)	21
1.3. Web Sunucu.....	22
1.4. Profinet CBA	23
1.4.1. Profinet CBA’ın Port Edilmesi.....	23
1.4.2. CBA Uygulaması.....	24
1.4.3. Mesaj Yorumlayıcısı.....	24
1.5. PLC Programı	25
2. Gerçekleme	25
2.1. IPC Mekanizması.....	25
2.1.1. unix_message.....	25
2.1.2. unix_socket	26
2.1.3. C Erişim Fonksiyonları.....	27
2.2. Seri Port Sunucusu (SerialServer)	28
2.2.1. serialbuf	28
2.3. Web Sunucu.....	29
2.3.1. Genel Görünüm: all.cgi	29
2.3.2. Profinet CBA: pncba.cgi, pncba_action.cgi	30
2.3.3. SerialServer: serial.cgi, serial_action.cgi.....	31
2.4. Profinet CBA	32

2.4.1. Alt Katmanların Port Edilmesi	32
2.4.2. Uygulama Kodu.....	34
2.4.3. UNIX Soketi Haberleşmesiyle Entegrasyon	37
2.5. PLC Programı	37
6. Deneysel Sonuçlar	39
7. Sonuç ve Öneriler	40
8. Kaynaklar.....	41
Ek A: Proje Ağ Şeması.....	42
Ek B: Proje Gantt Şeması	43

1. GİRİŞ

Bu projede *Profinet CBA* protokolünün temel sorunu olan eski veri yolu sistemleriyle uyumsuzluğu dikkate alan ve Profinet'in en büyük avantajı olan standart Ethernet bağlantısını kullanabilme yeteneğinden faydalanacak bir uygulama geliştirilmesi amaçlanmıştır.

Mevcut veri yolu sistemlerinin Profinet ile beraber çalışmasının sağlanması Profinet'in başarısı için oldukça kritiktir. Bu işlevi gerçekleştiren aygıtlar mevcut yatırımları koruyarak geçişi kolaylaştıracaklarından tüm yeni veri yolu protokolleri, bilhassa endüstriyel Ethernet'e dayananlar böylesi aygıtları geliştirmektedirler. Bu alandaki çalışmalardan en tipik Profinet'i geliştiren *Profibus Kullanıcıları Örgütü*'nün (PNO) tasarladığı Profibus Proxy'dir [1]. Bu cihaz 2004 itibarıyla 10 milyon düğümü bulunduğu tahmin edilen Profibus aygıtlarının [2] yeni kurulan Profinet sistemlerine sorunsuzca bağlanmasını amaçlamaktadır. Bir diğer çalışma Profinet dokümantasyonunda da değinilen Profinet Webintegration ürünüdür, bu ürün Profinet ağına bağlı aygıtlara HTTP protokolü üzerinden web hizmeti ile erişilmesini sağlamaktadır [3]. Üçüncü bir örnek de Comtrol firmasının geliştirdiği DeviceMaster UP ürünüdür. Bu ürün, çeşitli seri iletişim protokolleri ile (RS-232, RS-422, RS-485) çeşitli endüstriyel Ethernet protokolleri arasında (Profinet, Ethernet/IP, Modbus/TCP) köprü görevi görmektedir [4]. Projede de RS-232 protokolüyle iletişim kuran aygıtları Profinet'e bağlayacak bir sistem geliştirilmiştir. Bunu yaparken RS-232'ye özgü kod ayrı bir modüle konarak, ileride diğer protokollere – RS-485, CAN-Bus vb. – özgü modüllerin eklenmesi kolaylaştırılmıştır.

Profinet'in – ve diğer endüstriyel Ethernet sistemlerinin – sağladığı en önemli avantaj tek bir veri yolunun hem üretim sahası hem de ofiste kullanılabilmesi ve saha ile ofis arasındaki bilgi akışının altyapısının kendiliğinden hazır olmasıdır. Projede bu yetenekten faydalanmak için Profinet CBA ile aynı Ethernet yolu üzerinde çalışacak bir web sunucusu kapsama eklenmiştir. Böylece aynı hat üzerindeki eşzamanlı Profinet erişimi ve web erişiminin sistem başarımı üzerindeki etkisini gözlemek de projenin amaçları arasındadır.

Elinizdeki raporun ikinci bölümünde proje tanımı ve başlangıçta yapılan planlama anlatılmaktadır. Üçüncü bölüm, yukarıda ismi geçen teknolojilerin kısa bir tanıtımına ve projedeki işlevlerinin anlatılmasına ayrılmıştır. Dördüncü bölümde problem tanımı derinleştirilmiş ve çözümün genel modellenmesi yapılmıştır. Bir sonraki bölümde modellenen çözümün ayrıntılı tasarımı, gerçekleştirilmesi ve test tasarımı aktarılmıştır. Altıncı bölüm test sonuçları ve bunların değerlendirilmesini içermektedir. Yedinci bölüm ise projenin bir değerlendirmesini ve projeyi geliştirme önerilerini sunmaktadır.