

CAN Analyser의 필요성

최근 몇 년 동안 전 세계적으로 CAN 기반의 자동화 시스템들은 여러 형태의 산업 응용에 널리 보급되기 시작하였습니다. 이러한 CAN 디바이스와 시스템의 개발을 위해 필수적인 것은, 이것들의 설치, 테스트, 그리고 서비스와 더불어, 적절한 도구가 필요합니다. 분류된 시스템 life-cycle의 실제 단계에 따라, 보통은 시스템의 개발, 설치 그리고 서비스를 위한 특정 도구들이 사용됩니다.

다음은 시스템 설계와 개발에서부터 시스템 설치와 운영에 이르기 까지 CAN 기반 시스템의 전체 life-cycle 을 도와주는 매우 다용도인 CAN Analyser의 개념을 IXXAT사 제품을 예로하여 소개합니다.

통신 서버

IXXAT사의 CAN Analyser인 canAnalyser/32의 기본 기술 개념은 통신과 분석 작업의 엄격한 분리입니다. 통신 작업은 적용된 하드웨어 플랫폼 (PC/CAN 인터페이스)의 관리 뿐만 아니라 식별자들의 관리, CAN 메시지들의 실시간 중요한 전송과 수신을 포함합니다. 분석 작업은 사용자 인터페이스 그리고 canAnalyser/32에 의해 제공되는 CAN 버스 트래픽의 모니터링, 시뮬레이션, 또는 분석을 위한 여러 기능들을 포함합니다.

PC/CAN 인터페이스 접근을 위하여, "Control Panel"은 범용 프로그래밍 인터페이스인 VCI (Virtual-CAN-Interface)를 사용합니다. VCI는 PCMCIA, ISA, PCI, PC 카드 또는 병렬 포트와 USB 같은 여러 가지 인터페이스를 사용할 수 있으며, 11-과 29-비트 식별자들 처리, 신호 오류 프레임을 지원하며, 각각의 수신 오브젝트에 대해서 최대 25msec의 분석 시간을 허용합니다.

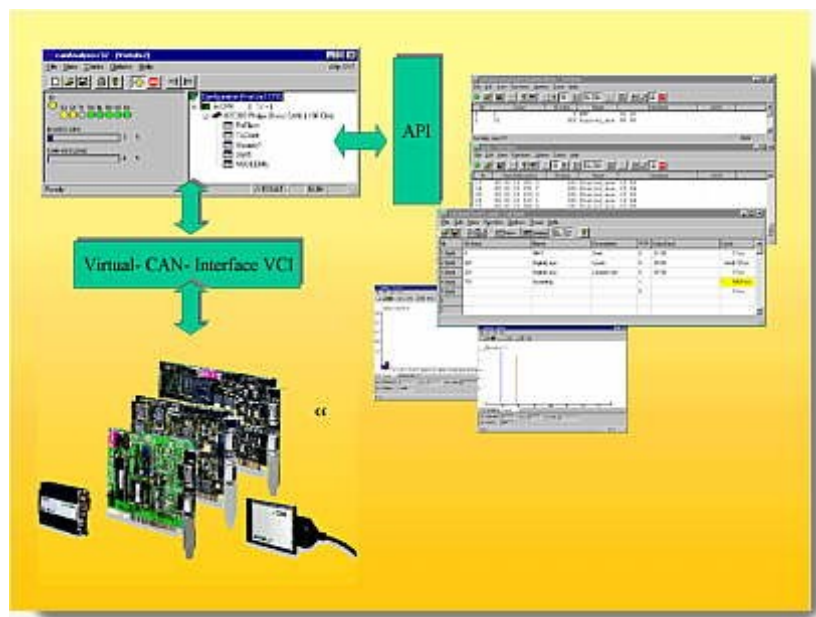


그림 1 : 통신과 분석 작업의 분리에 바탕을 둔 기술 개념

클라이언트 애플리케이션이 사용자 기능을 제공합니다

canAnalyser/32의 실제 기능은 "Client"에 의해 제공됩니다. 이러한 독립적인 "Client"는 Control Panel의 서버 기능에 바탕을 두고 있으며 오픈 응용 프로그래밍 인터페이스 (Control-Panel-API)를 통해 서버와 상호작용을 합니다. 이 API는 추상적 CAN 컨트롤러의 기능, 개체 (오브젝트) 버퍼들과 송수신 queue들의 구성, CAN 메시지들의 송신과 수신 그리고 실시간 메시지 trace 제어를 위한 기능들을 제공하며, 통신 Server에 통합되어 있습니다.

canAnalyser/32의 기본 버전에서는 CAN 메시지의 전송과 수신 (Transmit Client, Receive Client), 통계적 분석 (Statistic Client) 그리고 전송 메시지들의 순차적인 처리를 위한 (Batch Client) 여러 개의 기본 클라이언트들이 제공됩니다. 통신과 분석 작업을 구분하는 중요한 장점으로, 각 클라이언트를 여러 번 실행할 수 있습니다 (예를 들면 다중 receive window를 여러 개 표시). Receive Client에서 수신될 CAN 메시지들은 메시지 필터를 이용해서 구성될 수 있으므로, Receive Client는 시스템에서 단지 특정 디바이스 또는 기능의 데이터만을 나타내는데 사용될 수 있습니다. 따라서 각 디바이스 또는 기능은 각각의 receive window에서 표시될 수 있습니다. 수신된 데이터는 심볼 이름들과 수신시간으로 연속적으로 표시되거나, 또는 메시지 계수와 각 메시지의 최종 수신된 데이터를 통계적으로 표시될 수 있습니다. Transmit Client는 단일 메시지들의 전송을 수행합니다. 한 개의 메시지는 사용자의 요구시 또는 주기적으로 전송될 수 있습니다.

버스 소통량(traffic)의 통계적 분석을 위해 Statistic Client가 제공됩니다. Statistic Client는 전송된 CAN 메시지의 전체 수, 오류 프레임과 각 메시지의 초당 전송된 실제 수를 나타냅니다. 또한 그 때까지의 시스템에서의 비트율을 보여 줍니다.

전체 메시지의 순차적 전송 (예, 디바이스 또는 시스템의 시뮬레이션을 위해)은 Batch Client에 의해 지원됩니다. 이 클라이언트는 또한 저장된 트레이스 데이터의 온라인 재생을 위해서도 사용될 수 있습니다.

이러한 기본 클라이언트들 외에도, Control Panel 그 자체는 실시간의, 고성능, 트레이스 기능을 제공합니다. 메시지 트레이스는 하드 디스크에 직접 기록되므로 트레이스의 최대 길이는 이용할 수 있는 저장 용량에 따라 결정됩니다. 메시지 트레이스는 설정이 가능한 트리거 이벤트들을 사용하여 트리거될 수 있습니다. 트리거는 메시지 식별자, 메시지 데이터 비트 또는 원격이나 오류 프레임들의 발생을 통해서 가능합니다.

표준 클라이언트가 부가 기능을 제공합니다.

기본 클라이언트의 기능에 특수한 기능을 추가함으로써, Data Interpretation Module (DIM-Client), CANopen 또는 Programming Client와 같은 보다 더 표준적인 클라이언트 애플리케이션들의 형태로 이용할 수 있습니다. DIM Client를 이용하면 전송 또는 수신되는 데이터의 해석이 거의 모든 종류의 데이터와 변환 규칙들에 의해 지원됩니다. 그러므로 처리 변수들, 상태와 제어 정보가 사용자에게 이해할 수 있는 형태로 표시됩니다. 이 클라이언트는 그러므로 분산된, CAN 기반 자동화 시스템의 설치와, 테스트 그리고 유지보수에 특히 적합한데, 왜냐하면 서비스 직원이 통신 프로토콜의 세부사항을 알면 안 되기 때문입니다.

유사한 방식으로, 특수한 CANopen Client가 시스템의 개발 동안 CANopen 기반 시스템의 개발자를 지원합니다. 이 경우 CANopen Client는 CANopen 표준 수행과 데이터 포맷에 따라 사용자 인터페이스를 제공합니다.

앞에서 언급된 표준 클라이언트들 외에도, 디바이스나 시스템 에뮬레이션을 위한 특정용도 애플리케이션들의 구현이 개방형 Control-Panel-API를 매개로 하여 지원됩니다.

시스템 개발의 전통적인 방식에는 어떤 특정 CAN 도구가 디바이스의 개발 동안만 이거나 또는 시스템의 설치시에만 사용됩니다. 나중에 다른 도구들이 시스템의 수행을 위해 필요합니다. 개방형 API를 갖고 있는 canAnalyser/32는 설계 단계에서 사용될 뿐만 아니라 나중에 수행과 진단을 위해서도 사용될 수 있습니다.

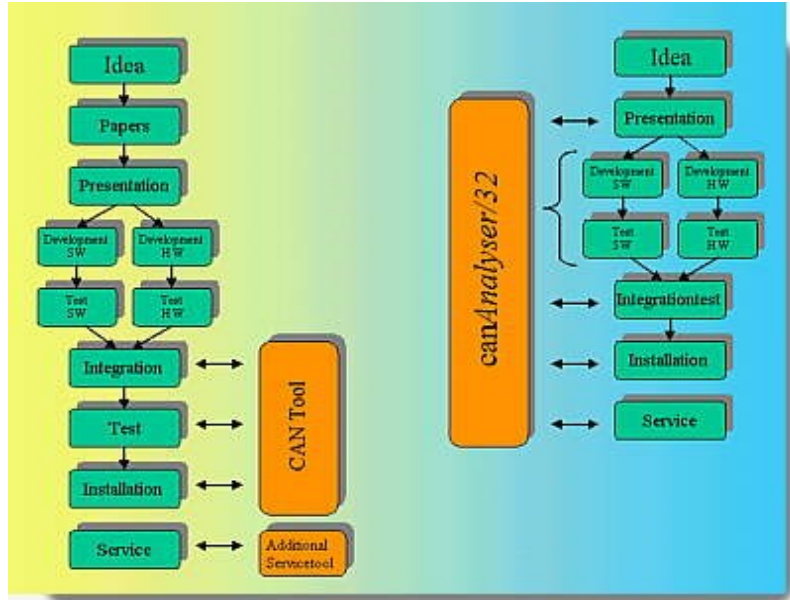


그림 2 : canAnalyser/32가 없을 때와 있을 때의 시스템 개발 과정

시스템 에뮬레이션과 시험을 위한 canAnalyser/32의 사용은 이제 간단한 엘리베이터 시스템의 설계와 개발을 예로 들어 설명합니다.

간단한 엘리베이터 모델에서는, 단지 엘리베이터 구동 제어 장치, 엘리베이터(몸체) 패널(car panel), 그리고 여러 바닥 패널(floor panel) 장치들이 검토되어집니다. 사용자 요청이 엘리베이터 패널 또는 바닥 패널들로부터, 그 요청을 처리하고 엘리베이터를 원하는 층으로 이동시키는 것을 제어하는 엘리베이터 컨트롤러 장치로 전송됩니다. 이 엘리베이터 컨트롤러는 또한 실제 엘리베이터 위치를 엘리베이터와 바닥 패널에 표시하도록 분산시킵니다.

시스템의 최초 프리젠테이션과 평가, 시스템의 완전한 기능적 시제품이 canAnalyser/32를 바탕으로 개발될 수 있습니다. 개방형 Control-Panel-API와 델파이, 비주얼 C++, 또는 비주얼 베이직 같은 일반 윈도우 프로그래밍 환경을 함께 이용하여 특정 클라이언트를 작성할 수 있습니다. 여기 예에서는, 엘리베이터 컨트롤러, 몸체와 바닥 패널 장치들의 에뮬레이션을 위한 구체적인 클라이언트들이 작성되어야합니다. 클라이언트들은 CAN 메시지들을 매개로 하여 이미 서로간에 통신합니다. 한 클라이언트의 한 개 이상의 인스턴스가 시스템에서 존재할 수 있기 때문에 단지 한 개의 바닥 패널 클라이언트만이 작성되어야만 합니다.

엘리베이터 컨트롤러와 패널 소프트웨어의 개발은 우선 PC 환경에서 수행될 수 있으며 향후에 타겟 하드웨어로 개발된 코드의 이식이 이루어집니다. 이러한 처리의 가장 큰 장점은 PC 환경으로 제공되는 편리한 디버깅의 사용이 가능하다는 것입니다. 여기에 추가하여 하드웨어가 동시에 개발될 수 있습니다.

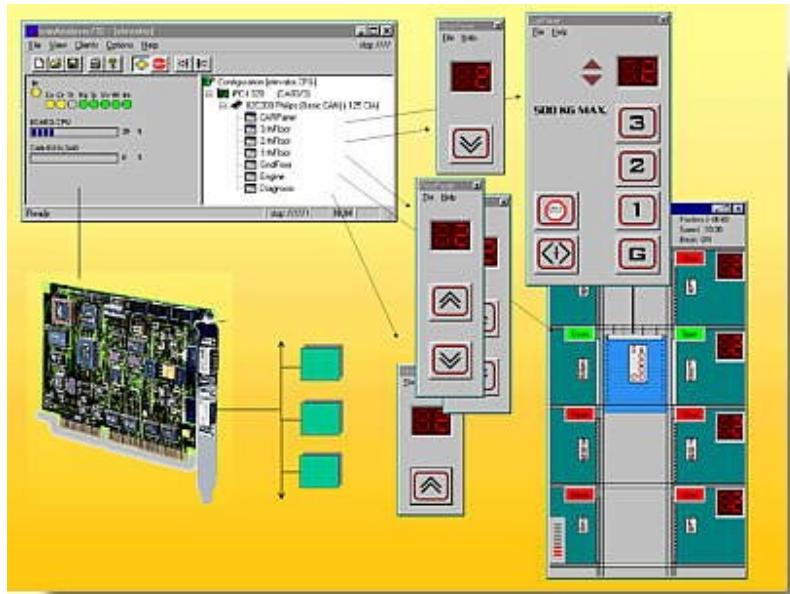


그림 3 : 엘리베이터 시스템을 위한 개발과 테스트 환경으로서의 canAnalyser/32

장치 (예를 들어 엘리베이터 컨트롤러)의 개발과 테스트 동안, 다른 장치들 (몸체와 바닥 패널들)의 기능성에 대한 시뮬레이션이 요구되어집니다. 이 경우, 전체 시스템이 우선 canAnalyser/32를 바탕으로 에뮬레이트된 경우, 전체적인 시뮬레이션과 시스템 테스트가 이미 이용 가능합니다. 하나의 장치 하드웨어의 이용이 가능한 후에는, PC 소프트웨어가 타겟 하드웨어로 전송되고 나머지 클라이언트들은 에뮬레이션 시스템의 환경 내에서 테스트될 수 있습니다. 그러므로 타겟 하드웨어와 소프트웨어의 단계별 통합과 테스트가 가능합니다. 개발 사이클의 초기에 이미, CAN 메시지 형태로 특정 테스트 패턴들을 사용함으로써 테스트를 수행할 수 있습니다. 테스트 메시지들은 시스템을 시뮬레이트 하기 위해 클라이언트들에 의해 사용됩니다. 클라이언트들의 반응들이 평가되고 기록됩니다.

시스템의 설치 후에 동일한 테스트 패턴들과 결과들이 시스템의 테스트를 위해 다시 사용될 수 있습니다. CANopen과 DeviceNet 기반 시스템들의 설계와 개발은 이러한 상위 계층 시스템 접근들의 적절한 프로토콜 DLL 모듈들을 이용하여 지원됩니다.

요약

canAnalyser/32를 이용하면, CAN 기반 자동화 시스템의 설계, 개발, 설치와 유지보수에 관한 매우 유용한 조력을 받을 수 있습니다. 클라이언트-서버 구조에 기초하여, 이 도구는 CAN 메시지들의 모니터링, 발생과 분석을 위한 기본 기능들 뿐만 아니라, 애플리케이션 또는 프로토콜 지향 사용자 인터페이스들을 위한 수단들도 제공합니다. 매우 편리한 이벤트 지향 프로그래밍 언어를 이용하여, 특수한 CAN 테스트 또는 프로토콜 처리 함수들의 개발이 지원됩니다.

또한 디바이스의 시제품 제작, 에뮬레이션과 테스트를 위한 플랫폼과 완벽한 시스템 솔루션으로 매우 유익하게 사용될 수 있습니다. 이 기술의 활용도는 복잡한 선박 자동화 시스템, 난방 제어 또는 측정 시스템들과 같은 다양한 애플리케이션들에 사용될 수 있습니다.