

Safety over EtherCAT, 확장된 IEC 61784-3:2021 요구사항에 충족됨을 TÜV로부터 승인받다

올해 2월 IEC 61784-3의 4판이 발행된 후, TÜV SÜD Rail GmbH는 EtherCAT 프로토콜 Safety over EtherCAT(FSoE)이 아무 변경 없이도, 기능적 안전성을 위해 확장된 IEC 표준의 요구사항에 충족됨을 공식적으로 승인했다.

Safety over EtherCAT은 2010년 IEC-61748-3-12:2010에서 FSCP 12(Functional Safety Communication Protocol)로 이미 국제 표준이 되었으며, 오늘날까지도 여전히 원본 1판에 의해 유효하다. TÜV SÜD는 FSoE 프로토콜이 아무 변경 없이 IEC 61784-3:2021 4판의 새로운 요구사항도 충족시키며, Safety Integrity Level (SIL) 3까지의 적용에 적합하다는 것을 확인했다. 따라서, 2010년 이후 디바이스와의 완전한 호환성은 그대로 보장된다.

FSoE 프로토콜은 전송 매체가 안전성 고려사항에 포함되지 않기 때문에, 통신 시스템 자체의 특성에 대한 어떠한 요구사항을 두지 않는 이른바 블랙 채널 접근방식을 기반으로 한다. 따라서 Safety over EtherCAT은 EtherCAT을 기반으로 할 뿐만 아니라, 표준 이더넷을 통한 기계 네트워킹 또는 무선 전송을 통한 모바일 기계 연결과 같은 다른 통신 프로토콜, 미디어를 통해 라우팅되는 경우가 많다.

IEC 61784-3, IEC 실무 그룹 WG12의 ETG 전문가 Guido Beckmann은 다음과 같이 설명했다. "우리는 안전한 통신을 위한 새로운 연구와 요구사항들을 논의하고 구체화하기 위해서, IEC에 적극적으로 동참합니다. 처음부터 FSoE 프로토콜의 보안 조치는 이미 IEC 61784-3:2021의 새로운 요구사항에도 충족될 수 있도록 설계되었습니다. 이를 통해 사용자와 많은 디바이스 공급업체는 안정성, 최대한의 투자 보호와 함께 개방적인 세이프티 관련 커뮤니케이션을 보장받을 수 있습니다."

IEC 61784-3은 산업 네트워크에서 기능적으로 안전한 통신을 위한 프로필에 대한 일반적인 요구사항을 정의한다. IEC 61784-3:2021의 신판에서는 기본적으로 두 가지 요구사항이 늘어났다. 첫 번째, 안전한 논리적 연결에서 적시성, 신뢰성, 데이터 무결성에 대한 잔여 오류 확률을 추정하는 모델이 도입되었다. 두 번째, 이제 표준은 세이프티

프로토콜의 명시적이고 암묵적인 메커니즘의 보호를 요구한다. 이 경우, 암시적인 메커니즘은 보호에 필요한 모든 데이터를 전송하지는 않지만, 송신자와 수신자 모두에게 알려진 데이터를 사용하여 전체 CRC 서명을 계산한다. 예를 들어 Safety over EtherCAT 의 경우, 증가 사이클 카운터에 해당한다.

EtherCAT 테크놀로지 그룹 (ETG):

EtherCAT 테크놀로지 그룹은 EtherCAT 기술을 지원, 홍보, 발전시키기 위해 다양한 산업의 주요 사용자 회사와 자동화 공급 업체가 힘을 모으는 협회이다. 69 개국의 6,300 곳 이상의 회원사를 보유하고 있는 EtherCAT 테크놀로지 그룹은 세계에서 가장 큰 필드버스 협회가 되었다. 2003 년 11 월에 설립된 이 협회는 필드버스 협회 중 가장 빠르게 성장하고 있다.

EtherCAT®:

EtherCAT 은 유연한 토폴로지를 가진 고성능, 저비용의, 사용하기 쉬운 산업용 이더넷 기술이다. 2003 년에 소개되었고, 2007 년부터 국제 IEC 표준 및 SEMI 표준이 되었다. EtherCAT 은 개방형 기술로, 누구나 EtherCAT 을 구현하고 사용하는 것이 허용된다.

→ 더 자세한 내용: www.ethercat.org

연락처:

EtherCAT Technology Group

Christiane Hammel
Ostendstraße 196
90482 Nuremberg
Germany

Tel.: +49 (911) 5 40 56 226

Fax: +49 (911) 5 40 56 29

press@ethercat.org

www.ethercat.org/press