Overview

KNX is a standard for wired home automation and building automation. KNX devices can manage lighting, blinds, and shutters, HVAC, security systems, energy management, audio-video, white goods, displays, remote control, etc. The KNX standard has been built on the OSI-based EIB (European Installation Bus, EIB, or Instabus) communication stack extended with the physical layers, configuration modes, and application experience of BatiBUS and EHS.

KNX systems are relatively inflexible because new wiring is needed for each new or relocated device. KNX incorporates tools for project engineering tasks, such as linking a series of individual devices into a functioning installation and integrating different media and configuration modes. This is made available in the Engineering Tool Software (ETS) suite.

EasyEdge KNX Engine connects to KNX networks via a KNX IP interface. EasyEdge KNX Engine supports importing the project descriptor files from the ETS (esf and knxproj) and Auto Binding process, allowing easy creation of the interface for the Read and Write functions on a KNX installation. Multiple IP interfaces are supported for distributed large systems.

Features

- Connects through KNXnet/IP tunneling;
- Supports KNXnet/IP server discovery;
- Supports KNXnet/IP routing (multicast);
- Supports importing ETS projects from esf and knxproj files;
- Supports Read, Write and Indication group services;
- Supports Communication (C), Read (R), Write (W), Transmit (T), Update (U) and Read on Init (RI) group flags;
- Support for data model automatic binding through group description;
- Besides events, it supports polling with adjustable request time;
- Adjustable polling request time per group address:
    - three-level (main[5-bit]/middle[3-bit]/sub[8-bit]);
    - two-level (main[5-bit]]/sub[11-bit]);
    - one-level (16-bit freestyle group address);
- Supported datapoint types:
    - 1.yyy = boolean, like switching, move up/down, step;
    - 2.yyy = 2 x boolean, e.g. switching + priority control;
    - 3.yyy = boolean + 3-bit unsigned value, e.g. dimming up/down;
    - 4.yyy = character (8-bit);
    - 5.yyy = 8-bit unsigned value, like dim value (0..100%), blinds position (0..100%);
    - 6.yyy = 8-bit 2's complement, e.g. %;
    - 7.yyy = 2 x 8-bit unsigned value, i.e. pulse counter;
    - 8.yyy = 2 x 8-bit 2's complement, e.g. %;
    - 9.yyy = 16-bit float, e.g. temperature;
    - 10.yyy = time;
    - 11.yyy = date;
    - 12.yyy = 4 x 8-bit unsigned value, i.e. pulse counter;
    - 13.yyy = 4 x 8-bit 2's complement, i.e. pulse counter;
    - 14.yyy = 32-bit float, e.g. temperature;
    - 15.yyy = access control;
    - 16.yyy = string -> 14 characters (14 x 8-bit);
    - 17.yyy = scene number;
    - 18.yyy = scene control;
    - 19.yyy = time + data;
    - 20.yyy = 8-bit enumeration, e.g. HVAC mode ('auto', 'comfort', 'standby', 'economy', 'protection').