Touch Panel Design for KNX Home Automation System

AHMET BURAK GÖKBAYRAK[1], SEMAVI AKIN[2] and OĞUZHAN URHAN[1]

[1]Electronics & Telecom. Eng. Dept. Embedded and Vision Systems Lab. (KULE)
Kocaeli University
Umuttepe Campus, 41380, İzmit-Kocaeli
TURKEY
{burak.gokbayrak, urhano}@kocaeli.edu.tr - http://kule.kocaeli.edu.tr

[2]Viko by Panasonic R&D Center
İstanbul, TURKEY
sakin@viko.com.tr – http://www.viko.com.tr

Abstract: - KNX is a widely used standard for building and home automation solutions in recent years across the Europe. In many real-life scenario, KNX-enabled devices (i.e. sensors and actuators) are controlled by making use of a multi-functional controller or touch panel. In this paper, a touch panel design is presented to monitor and control KNX networks. The KNX touch panel is designed using ARM Cortex-A8 series processor and it can execute Android operation system (OS). We also developed a KNX application running on the Android OS using the Java Native Interface (JNI). Our experiments show that the designed hardware and Android application allow efficient control of KNX devices connected to the KNX network.

Key-Words: - Home/Building automation, KNX, Touch Panel.

1 Introduction

Today, developments in many technology changing the way we live. For example, automation systems started to find many applications in home/building area besides the factories. The idea of observing and controlling whole system by making use of a control station has increased the demand for home/building automation systems. There are many popular protocol in home automation area including X-10 [1], UPB [2], ZigBee [3], Z-wave [4] and KNX [5]. The X-10 is one of the oldest home automation protocol which employs the power lines already installed in the building. At the zero-crossing of the AC waveform one-bit is transmitted by encoding the signal at 120kHz for 1ms duration. The UPD (Universal Powerline Bus) utilizes pulse position modulation (PPM) to eliminate possible noise and interference effect. It superimposes the data signal into sinusoidal power signal which makes the UPD more robust compared X-10. ZigBee and Z-wave protocols enable wireless automation. The Konnex (simply KNX) is the leading home automation standard which supports both wired and wireless physical communication media.

KNX system was developed by leading companies working on home/building automation area [6] and administrated by the KNX Association. KNX Association was founded by EIBA (European Installation Bus Association), EHSA (European Home Systems Association) and BCI (BatiBUS Club International) in Brussels in 1999. KNX was approved international standard (ISO/IEC 14543-3) for home/building automation in 2006. Additionally, it was certified as America (ANSI /ASHRAE 135), European (CENELEC EN50090, CEN EN 13321-1 and 13321-2) and China (GB/T 20965) standard as well. KNX system becomes the most commonly used standard in Europe for home/building automation solutions by reaching about 75% [7] adaption rate. Today, more than 300 companies are producing KNX devices all over the world.

KNX is an open source standard. It is possible to develop KNX based home/building automation systems which include many sensors and actuators such as on-off switch, heating and curtain controls, etc. [7]. All the KNX devices can communicate with each other thanks the certification process handled by the KNX Association. On the other hand, the association provides a tool named as ETS (Engineering Tool Software) which is a manufacturer independent configuration software to design and configure intelligent home and building control installations with the KNX system. Each KNX certified device has specific library which is included into ETS tool for its usage in KNX system projects. It is important to note that each KNX installation requires some configurations in ETS tool.

ISBN: 978-1-61804-329-0

There are several physical environments that can be used for KNX. In Table 1, comparison of different KNX environments are given. The most common physical environment in KNX system is the twisted pair (KNX.TP). KNX also allows data transmission using power lines, RF and Ethernet interfaces.

Table 1. Comparison of different KNX environment

Environment	Transmission Line	Application Areas
Twisted pair	Different control cable	Areas where new installations or restorations on a large scale
Power Line	Existing power line	Areas where power lines are available and it is not possible to make new installation.
Radio Frequency	Wireless	Areas where the cable installation is not desired
Ethernet	Ethernet line	Fast communication needed in huge amount of devices.

A sample KNX environment is given in Fig.1. As it is seen from the figure, different end units like lighting, switch, sensors can be found in KNX line.

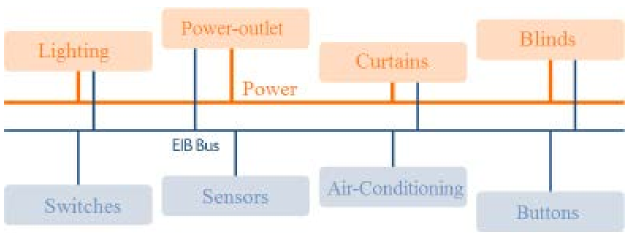

Fig.1. A sample KNX environment [7].

Devices in KNX system divided into 2 main groups. These are actuators (on/off devices, dimmers, curtain/blinds modules, input modules etc.) and drivers (switches, detectors, thermostats, touch panels, displays etc.). A KNX system may consist of more than 50.000 devices with an interconnected bus. Actuators and sensors communicate sending messages each other using the interconnected bus.

One of the most important feature of the KNX system is its scalability. The smallest unit of KNX system is called as Segment which may contain up to 63 devices. A KNX line is formed by the combination of 4 KNX segments. Each segment must contain 29 Volt DC power supply. When 15 lines are connected to each other, they constitute a KNX area. The KNX system supports up to 15 areas. Thus, by making use of 15 KNX areas, it is possible to address more than 50.000 KNX devices in a KNX network. The KNX connection topology is shown in Fig. 2.

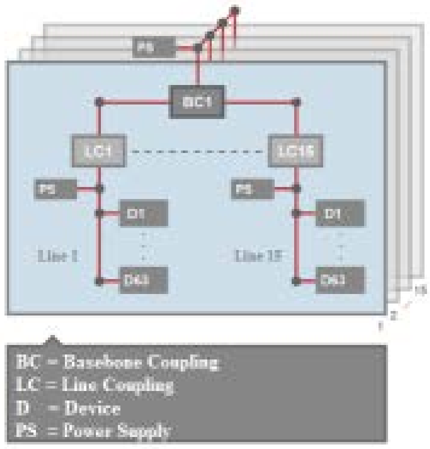

Fig. 2. KNX connection topology [8].

2 The System Developed

The system developed consists two main hardware components. These are circuit which consists touch panel and ARM based microprocessor board. The touch panel is manufactured by a supplier and has a parallel interface for data connection to ARM board. The ARM based microprocessor board is designed specifically for Android OS based touch panel application. The PCB of the ARM based processor board is shown in Fig. 3. This board is connected to touch screen via pin header connections.

The system developed in this word is connected a KNX test setup shown in Fig. 4 where KNX enabled on-off switches, dimmers and curtain contain modules are available.

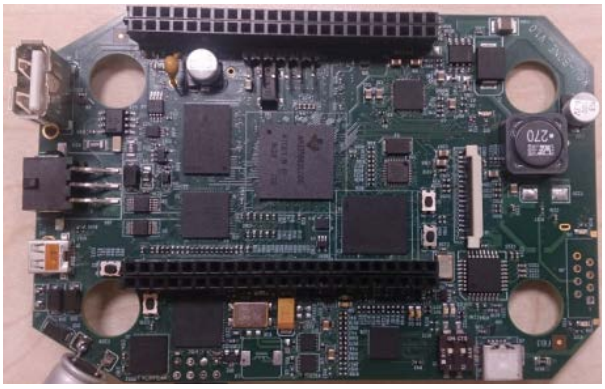

Fig. 3. The ARM processor based PCB.

Fig. 4. KNX test panel used in this system.

The developed system also consists a graphical user interface to provide interaction with the user. This graphical user interface is designed using Java language to run on Android OS.

2.1 System Hardware

In the developed system, ARM Cortex-A8 based microprocessors is employed to run Android OS and provide data communication with KNX network. The ARM based processor is to enable easy software development using Java. Additionally, it becomes possible to drive touch screen without extra effort.

The communication between the developed KNX touch panel and the KNX network is handled by a TI MSP430 based microcontroller. KNX stack is implemented on this MCU which means that this MCU acts as a KNX controller. Thus, the data coming from the KNX network is initially handled by this MCU and then transferred to ARM based processors. The connection between ARM based processor and MSP430 based KNX controller is performed using RS232 interface. We design a specific data structure including specific device IDs between these modules to data exchange between KNX touch panel and KNX network. At the physical layer of the KNX network, ELMOS E981.03 IC is utilized. The block diagram of the overall system is given in Fig. 5.

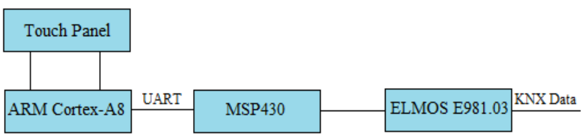

Fig. 5. System hardware diagram.

2.1 System Software

The user interface is one of the most important part in a typical KNX touch panel. The user interface need to be configured easily at the installation time. Additionally, it needs to be user friendly in daily usage.

We prefer to utilize Android OS infrastructure to develop intended application easily. The applications running on Android OS usually are develop by making use of Java programming language. However, when additional hardware units required to be integrated into the system, it is necessary to use low-level programming languages.

JNI (Java Native Interface) simply enables JVM (Java Virtual Machine) to execute the code written in low-level languages such as C/C++. Thus, we employ JNI to reach KNX interface.

In this work, graphical user interface (GUI) functions in the developed application is designed using Java programming language. The communication with KNX interface unit is a low level operation and it is implemented in C programming language. After that it is integrated application software using the JNI.

In the developed software there are 3 main screens. These are startup screen, configuration screen and user screen.

When the application program is launched, a startup screen shown in Fig 6 is displayed first. This start-up screen allows the user to choose configuration or user mode.

Fig. 6. Startup screen.

At the configuration mode of the KNX touch panel the user interface allows user to set the KNX devices in the rooms. It is important to note that the KNX system need to be configured via ETS software before this operation. Next, it becomes possible to configure devices the room. A typical configuration screen in the configuration mode of the KNX touch panel is shown in Fig. 7.

ISBN: 978-1-61804-329-0

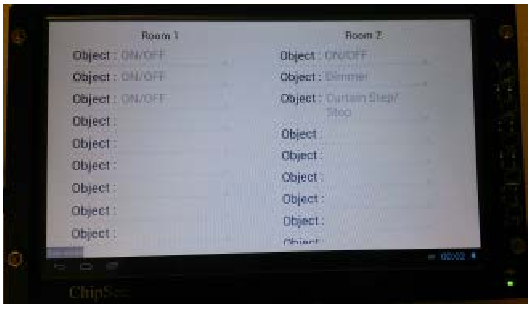

Fig. 7. A sample configuration screen.

The user mode of the developed KNX touch panel allows users to send and receive data from the KNX network. The developed application supports up to 10 rooms. The navigation between these rooms are carried out by simple swiping gesture. The users can change the status of the device on the KNX network with corresponding objects in the room. The contents of room screens in the user mode is adjusted dynamically according to changes at the configuration screen. A sample user screen is shown in Fig. 8.

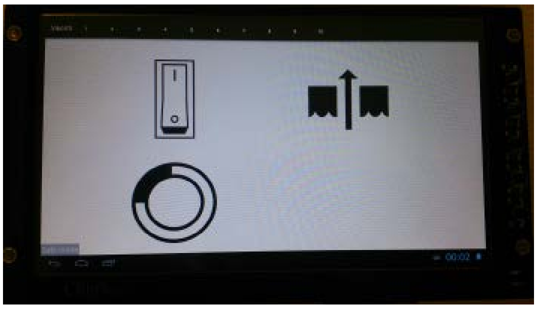

Fig. 8. A sample user screen.

At the used screen, it is required to transmit instructions to KNX network taking the user touch based inputs into account. These data is transmitted to KNX network via KNX controller and PHY. The data transfer between the ARM application processor and MSP430 based KNX controller is carried out according to the data packet structure is given in Table 2. In this packet structure SOF (Start of Frame) represents the packet start whereas the EOF (End of Frame) denotes end of the frame. Obj. ID defines the object numbers of devices in KNX network.

Table 2. Data packet for communication between the application processor and KNX controller.

SOF	Obj. Id	ACK	Data	CRC	EOF
1byte	1 byte	1byte	7 byte	1 byte	1 byte

ACK (Acknowledge) provides notification of the received data. The payload information is arranged as 7 byte and called Data in this table. The CRC is used to check the data integrity. A simple Ex-OR based approach is adopted for this purpose.

3 Experiments

The developed hardware and software modules are connected to the KNX shown in Fig. 4. Next, extensive test scenarios are executed to investigate effectiveness of the proposed KNX touch panel. Our observations show that the proposed architecture allow data transfer between the panel and KNX network. Additionally, all the functionalities of the test environment is controlled including switches, motion detectors, thermostat and also drivers for controlling systems like lighting, curtain and fan-coil.

4 Conclusions

A touch panel hardware and software is developed for KNX based home/building automation applications. The system enables efficient communication thanks to its compact hardware and user friendly GUI. The device is capable of running Android OS and KNX application developed. Experiments show the effectiveness of the proposed KNX touch panel.

Acknowledgement

This work was supported by the Turkish Ministry of Science, Industry and Technology under grant number 0315.STZ.2013-2.

References:

[1] Walko J., Home Control, IET Computing & Control Engineering, 2006, **5**, 16-19.

[2] Powerline Control Systems, “The UPB System Description” (2003).

[3] Gislason D., *Zigbee Wireless Networking*, 1st Ed., WA, 2008.

[4] Zensys A/S, “Z-Wave Protocol Overview” (2006).

[5] Demarest J., “Home & Building Automation with KNX – Benefits compared to competing systems”, KNX Association, September 2008.

[6] KNX Association. "KNX Standard." (2013).

[7] Gokbayrak A. B., Home Automation with Wireless Sensor Networks and KNX Integration, University of Kocaeli, MSc Thesis, 2015.

[8] Kyselytsya Y., Weinzierl T., “Implementation of the KNX Standard”, Weinzierl Engineering GMBH, 2006.

ISBN: 978-1-61804-329-0