FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Transmitting video data over narrow bandwidth control networks

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Medizinische Informatik

eingereicht von

Felix Schuster
Matrikelnummer 0625068

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung: Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Wolfgang Kastner
Mitwirkung: Dipl.-Ing. Mag.rer.soc.oec. Dr.techn. Wolfgang Granzer

Wien, 28.11.2011

(Unterschrift Verfasser) (Unterschrift Betreuung)

Erklärung zur Verfassung der Arbeit

Felix Schuster
Gestettengasse 17/6/5, 1030 Wien

Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

(Ort, Datum) (Unterschrift Verfasser)

Abstract

Nowadays Closed Circuit Television (CCTV) systems are transmitting video signals to a dedicated control center where human beings are observing the video streams continuously – usually huge amounts of data. Most of the time there are no abnormal activities which can be detected – moreover video frames do not even change. However, due to the continuous video transmission, resources like network bandwidth and processing power are wasted. In addition concentration of security personnel decreases rapidly.

The proposed approach aims at performing on-the-spot image analysis and image transmission over Building Automation Networks (BAN). The desired solution is to transmit video data just in case. If an abnormal situation is detected by the camera, it shall send the related video sequence over a BAN to the control center. Video transmission should be stopped during idle times.

Although current BAN technologies are not capable of handling huge amounts of data, this master thesis focuses on solutions in transmitting video sequences over narrow bandwidth control networks that are typically used for the exchange of process data in building automation systems.

Network technologies used in the building automation domain (e.g., twisted pair, powerline, technologies based on radio frequency) provide only limited bandwidth. Even though on-the-spot image analysis decreases the amount of data to be transmitted, further improvements and special developments are necessary. Therefore, this master thesis has the objective to provide mechanisms that support a transmission of on-demand video sequences over low bandwidth network media.

In a first step, theoretical fundamentals about the transmission properties of different protocol standards will be analyzed. Based on this, a comprehensive analysis of appropriate network technologies that fulfill the requirements of on-demand video transmission will be further investigated. Afterwards, a protocol extension that supports the transmission of (compressed) video data shall be developed. To evaluate the proposed concepts, a prototype implementation will be realized.

Kurzfassung

Zur Überwachung eingesetzte Kameranetzwerke (CCTV Systeme) sind eigenständige Netze. Die gemeinsame Nutzung von bestehenden Netzen aus der Gebäudeautomation ist kaum zu finden. Das beruht zum einen darauf, da Hersteller von Heizung, Lüftung und Klimatechnik (HLK) Systemen das notwendige Know-How nicht mit sich bringen, um sicherheitsrelevante Infrastruktur einfügen zu können. Außerdem sind solche schmalbandigen Netzwerke nicht zur Übertragung großer (Video-)Datenmengen ausgelegt. Das zentrale Design dieser Kameranetzwerke und die kontinuierliche Datenübertragung erfordern große Bandbreiten.

CCTV Systeme werden immer größer und umfangreicher. Das ist auf die kleiner werdenden Hardware-Preise zurückzuführen. Die eingesetzten Kameras übertragen jedoch ständig Bildsequenzen, ohne jegliche Analyse vorzunehmen. Diese Tätigkeit verbleibt beim Personal eingerichteter Sicherheitszentralen. Durch diese enormen Informationsmengen kommt es jedoch frühzeitig zu Ermüdung des Personals. Ein konzentriertes Arbeiten ist nur zeitlich eingeschränkt möglich.

In dieser Diplomarbeit wird die Integration von intelligenten on-the-spot Minikameras in bestehende Netze der Gebäudeautomation vorbereitet. Solche Kameras bringen den Vorteil mit sich, dass sie ständig den Bildbereich analysieren und bei definierten Ereignissen eine Alarmmeldung und einen Mitschnitt an die Sicherheitszentrale senden. Weiters können erste Gegenmaßnahmen direkt veranlasst werden, während das Personal der Sicherheitszentrale weitere Aktionen erlässt. Zudem sind solche Kameras äußerst kostengünstig zu erwerben, kommt es doch in erster Linie nicht auf die Qualität der zu übertragenden Daten an. Da Gebäudeautomationsnetzwerke nicht für solche Zwecke konzipiert wurden, müssen die Rahmenbedingungen hierfür erst geschaffen werden. Übertragung über unterschiedliche Medien (Zwei-Drahtleitungen, Stromleitungen, Funkverbindungen) werden durch Protokolle und Standards der Gebäudeautomation definiert. Keine gängiger offener Standard ist bis dato für die Datenübertragung von CCTV Kameras geeignet.

Diese Diplomarbeit prüft die Möglichkeit einer solchen Umsetzung anhand von beschriebenen Anwendungsfällen aus verschiedenen Bereichen. Dazu wird auch der generelle Netzaufbau herangezogen und analysiert. Für zwei ausgewählte Standards werden die Protokolle entsprechend erweitert und eine Proof-of-Concept Realisierung vorgestellt.

Acknowledgment

This work was funded by FFG (Austrian Research Promotion Agency) under the Kiras project “Networked miniSpot” P824777.

Contents

1 Motivation **1**
- 1.1 History of CCTV systems . . . 2
- 1.2 CCTV and BAS . . . 6
- 1.3 Increasing sensor efficiency . . . 7
- 1.4 Contribution and Outlook . . . 9

2 Use Cases **11**
- 2.1 Ambient Assisted Living . . . 11
- 2.2 Safety domain – Fire alarm . . . 13
- 2.3 Security domain – Protection . . . 14
- 2.4 Advanced HVAC control . . . 15

3 Technologies and their media in Building Automation Systems **18**
- 3.1 KNX . . . 18
- 3.2 BACnet . . . 21
- 3.3 LonWorks . . . 23
- 3.4 ZigBee . . . 24

4 Integrating CCTV systems into Building Automation Systems **26**
- 4.1 System architecture . . . 26
- 4.2 Suitable building automation network protocols . . . 29

5 Wired integration – case study KNX TP1 **41**
- 5.1 Profiles . . . 41
- 5.2 Datapoints . . . 42
- 5.3 Functional Blocks . . . 44
- 5.4 Proof-of-concept . . . 49

6 Wireless integration – case study ZigBee 2.4 GHz **53**
- 6.1 Application framework . . . 54
- 6.2 Improving the home automation profile . . . 56
- 6.3 Proof-of-concept . . . 58

7 Conclusion **64**

Bibliography **66**

Acronyms

AAL	Ambient Assisted Living
ACE	Ancillary Control Equipment
APDU	Application Protocol Data Unit
API	Application Programming Interface
APL	Application Layer
APS	Application Support Sub-Layer
ASG	Automation Systems Groups
ASHRAE	American Society of Heating, Refrigerating and Air-Conditioning Engineers
BAS	Building Automation System
BPSK	Binary Phase-Shift Keying
CCA	Clear Channel Assessment
CCD	Charge-Coupled Device
CCTV	Closed-Circuit Television
CFB	Camera Functional Block
CIE	Control and Indicating Equipment
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CSMA	Carrier Sense Multiple Access
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense Multiple Access with Collision Detection

CVL	Computer Vision Lab
DBG	Debug Module
DSN	Data Sequence Number
DSSS	Direct Sequence Spread Spectrum
EIB	European Installation Bus
ETS	Engineering Tool Software
FB	Functional Block
FCS	Frame Check Sequence
FFD	Full-Function Device
FIFO	First In First Out
FSK	Frequency Shift Keying
FT-10	Free Topology 10
IAS	Intruder Alarm System
IP	Internet Protocol
IR	Infrared
ISM	Industrial, Scientific and Medical
ISO	International Organization for Standardization
HVAC	Heating, Ventilating and Air Conditioning
LIFS	Long Interframe Spacing
LNS	LonWorks Network Services
LP-10	Link Power 10
LSAB	Light Switch Actuator Block
LSB	Least Significant Bit
MAC	Medium Access Control Layer
MFR	MAC Footer
MHR	MAC Header
MSB	Most Significant Bit

NTSC	National Television Systems Committee
NWK	Network Layer
O-QPSK	Offset Quadrature Phase-Shift Keying
OSI	Open Systems Interconnection
PAL	Phase Alternating Line
PAN	Personal Area Network
PDM	Persistent Data Manager
PDUM	Protocol Data Unit Manager
PHR	PHY header
PHY	Physical Layer
PL	Power Line
PL110	Power Line 110
PWRM	Power Manager
PSSS	Parallel Sequence Spread Spectrum
PSU	Power Supply Unit
RAM	Random Access Memory
RF	Radio Frequency
RFD	Reduced-Function Device
RISC	Reduced Instruction Set Computing
RTOS	Real-time Operating System
SFB	Surveillance Functional Block
SFD	Start-of-Frame Delimiter
SFSK	Spread Frequency Shift Keying
SHR	Synchronization Header
SRD	Short Range Device
TP	Twisted Pair
TP1	Twisted Pair 1

TTL	Time To Live
UART	Universal Asynchronous Receiver Transmitter
UDP	User Datagram Protocol
UML	Unified Modeling Language
WD	Warning Device
XML	Extensible Markup Language
ZC	ZigBee Coordinator
ZCL	ZigBee Cluster Library
ZDO	ZigBee Device Object
ZDP	ZigBee Device Profile
ZED	ZigBee End Device
ZR	ZigBee Router

CHAPTER 1

Motivation

> "CCTV denotes a closed circuit television system. The term CCTV is commonly used for monitoring systems using television."[1] [1]

Wege's definition [1] of Closed-Circuit Television (CCTV) is about 15 years old, but still remains valid in most cases. Typically, a CCTV system is isolated from other systems to reduce the risk of influences or faults. CCTV systems exist in many different application domains with varying reasons and importance. For instance there is no other way to safely observe the nuclear fission in a reactor than using camera systems. In contrast, an entry to a building can be observed by a doorman sitting next to it. Using CCTV systems usually has a positive effect: It can reduce man-power, if it is used properly.

CCTV systems were introduced 1947 when Diamond Electronics installed the first one in Ohio (USA). In the 1960's they found a wider distribution. Meanwhile the National Television Systems Committee (NTSC) and the Phase Alternating Line (PAL) standards were specified. After the development of Charge-Coupled Devices (CCDs) around 1980 cameras were getting smaller and cheaper. The area of CCTV systems had definitely begun.

Besides the units of CCDs inside a camera, the resolution and color depth of cameras increased strongly.

According to [2], a CCTV system consists of four major pillars. They are illustrated in Figure 1.1. *Compression* of video data is important whenever parameters like bandwidth, storage size or transmission time come into mind. Compression quality can be reduced only at the expense of transmission time. Both extremes are not usable and therefore a compromise has to be found.

The *transmission* of video or image data is the main subject within this thesis. In general it depends on the used protocol and media. Furthermore it often depends on existing installations and the communication possibilities and features between parts of CCTV systems of different

[1] original text: Das CCTV bezeichnet das in sich geschlossenes Fernseh-System. Der Begriff wird häufig als Kurzbezeichnung für eine Fernseh-Überwachungsanlage verwendet.

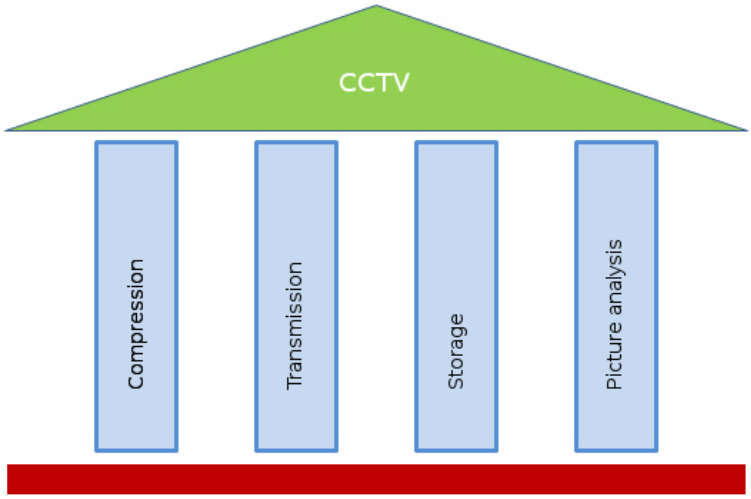


Figure 1.1: Four pillars of CCTV

vendors. The trend is towards open standards but there are still a number of products with proprietary protocols.

Especially *storage* is a positive aspect of newer digital systems where Random Access Memory (RAM) or non-volatile storage (e.g., hard disks) are used. It allows direct access to needed information and superseded reversing and forwarding video tapes which took a long time and allows direct access to needed information. Moreover, the storage capacity per unit is significantly higher thus leading to smaller storage devices. Anyway, the structure behind data storage systems has to be well considered in particular if the data volume shall be high.

Finally the *picture analysis* part is growing with digital technology. It can be assumed that current technologies are just the beginning [3] (e.g., face recognition, license tag parsing or manipulation of video data in real time). Picture analysis primarily depends on processing power and the algorithm speed. In the meantime even simpler tasks like motion detection are available to commercial cameras nowadays.

1.1 History of CCTV systems

When the very first camera systems for monitoring reasons were installed they were expensive and consisted of tubes. This made them vulnerable to different kinds of problems. Hence they were installed only where a camera was absolutely necessary. At this time camera systems were analog ones. Each camera was connected to a single monitor and there was no need to use crossbars or combining units. At this time it was not conceivable to use several cameras to observe non-critical areas.

In analog systems the transmission of video data and control data was separated. Control of cameras was often done by serial transfer using RS-232 or similar standards. Video data was transferred using coaxial cables or two wire transmission. Sometimes additional cables were

needed, if for example external signals were considered. This resulted in high installation costs. Since a central design of the CCTV system was popular one end of every cable was connected inside a single operator room. This was often not a trivial task. Due to the point-to-point cabling, installation and maintenance costs increased. Besides a single operator station being a single point of failure, it was also the main target of security attacks.

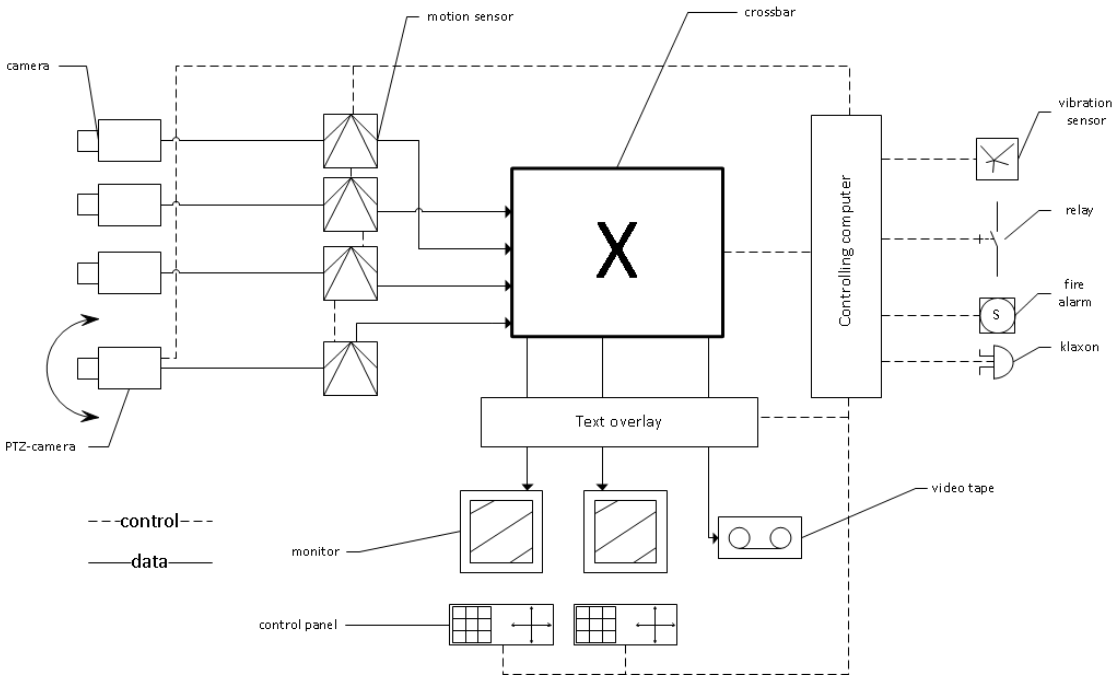


Figure 1.2: Analog CCTV system

A typical CCTV system with primarily analog components is shown in Figure 1.2. The main device is the video crossbar (marked with the X). It can handle n inputs (cameras) and m outputs (monitors) and is able to dynamically connect each camera to every single monitor. Cameras are connected through coaxial or two wire cables for video transmission. PTZ (pan, tilt, zoom)-cameras are connected with an additional serial data line. In front of the crossbar motion detectors are installed. They observe the video stream and notify the operator if motion is detected. Before the video is shown on one of the monitors a device can generate a text overlay. This is useful to display the camera's name or position. In a more advanced system an alarm message can even be displayed this way. The information needed is retrieved from the controlling computer using a serial connection. On the right side of the figure the controlling computer is installed. It handles simple actions. In case of an alarm, the video of the affected camera is shown on a predefined monitor and the event is videotaped. This connection is locked until the alarm is cleared. With operator panels next to the monitors the control of the PTZ cameras and the crossbar is made possible. Furthermore, devices connected to the controlling unit can be activated. In the figure a simple relay and a klaxon are shown. The central station

can use such devices too. For example a vibration sensor can detect glass breaking and a smoke sensor can trigger fire alarms. Both events could be displayed on the monitor. Of course, this is just an example and there are more possibilities (e.g., a video multiplexer for multiple cameras on one monitor), but for every device which is installed afterwards a high installation effort is needed.

In the course of time digital devices were developed. Thus CCTV systems were improved too. While cameras were still analog devices the network behind became digital. The most negative aspect of analog systems was reduced by using one medium which was capable of transmitting video and control data together. Multiple servers in a network retrieved the analog signals from cameras connected to them. They converted, compressed and provided them over the network. The video crossbar was replaced by a video management software. One of the greatest benefits of this digitalization was the storage possibility. Digital storage allows fast searching for video sequences, tagging with date and time or other keywords, and an easier archiving. If an alarm was detected the last few seconds before the alarm could be seen instantly. This should help better understanding the situation leading to an alarm.

> "Digital CCTV cameras capture motion pictures, compress them and provide them to an IP network. Terms like CCTV camera, CCTV network camera and CCTV IP camera can be used equally."[2] [3]

Nowadays cameras are working digital. They can even do simple analyzing tasks like motion detection on the spot. The difference to hybrid systems (which consist of analog cameras and a digital network in the background) is the decentralization. Against the definition of Döring [3] digital cameras do not need to communicate using the Internet Protocol (IP). Even though the cables were reduced, the benefit is, that any information is available in the network. If a connection is cut due to an attack, the connection can be established from another entry point. This point is needed for the operator's computer or tablet (or even his smart phone).

Different standards do not make life easier and the risk of incompatibility is existent, too. Every translation between different protocols entails loss of information and needs additional computational effort. However, there is no global standard. This is a thrilling challenge for standardization organizations.

In Figure 1.3 a modern CCTV system is presented. It consists of digital components capable of using the same network protocol. The sensors and actuators are connected via a dedicated controller to the network. On the left side of the figure (PTZ-) cameras are located. Moreover there is one server responsible for the connection to a Building Automation System (BAS) which is represented with a few sensors (presence detection and fire alarm) and actuators (a simple relay and a klaxon). Another server is responsible for the automated tasks in case of an alarm. Therefore it analyzes the situation continuously taking all information from sensors and cameras into consideration. A data storage server provides access to archived sequences. Since the camera is an embedded device the resources are typically limited. The best system would not

[2] original text: Digitale CCTV-Kameras sind Kameras, die Bewegtbilder in für CCTV-Zwecke geeigneter Bildqualität komprimieren und auf einer Netzwerkschnittstelle mittels IP-basierter Bildübertragungsprotokolle zu Netzwerkempfänger übertragen. Dabei sollen die Bezeichnungen digitale CCTV-Kamera, CCTV-Netzwerkkamera und CCTV-IP-Kamera als synonym angesehen werden.

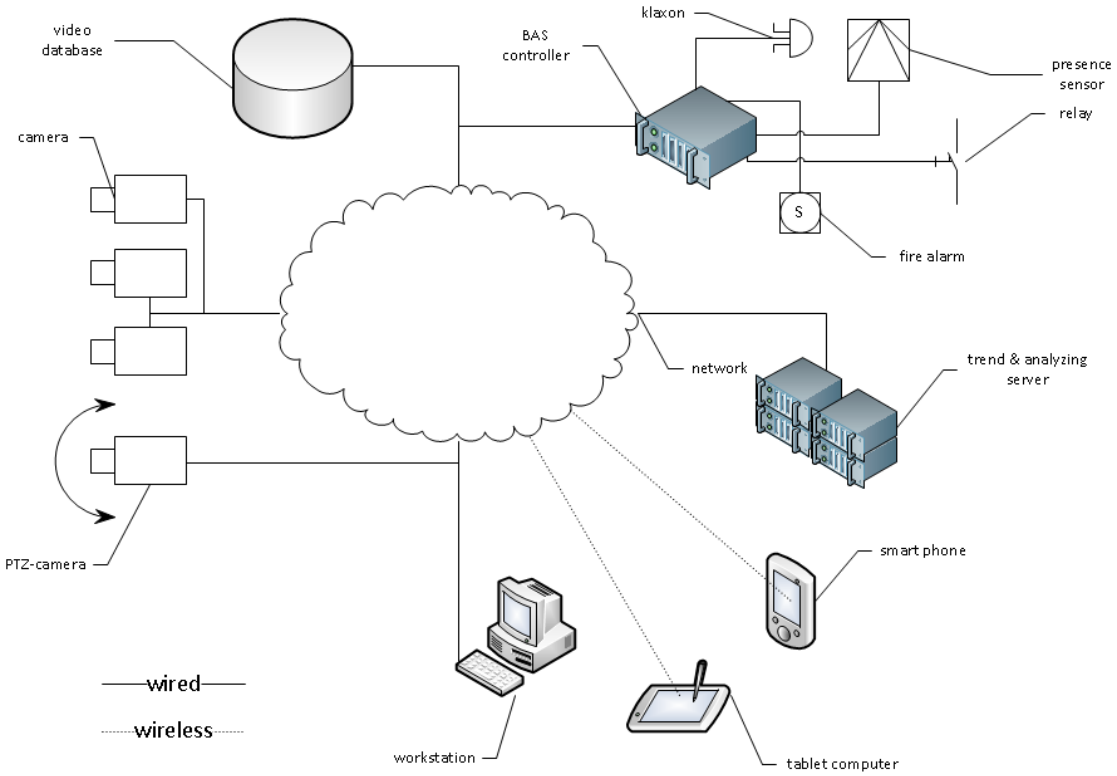


Figure 1.3: Digital CCTV system

work properly if there was no operator who controls the system, checks if alarms are valid and executes further tasks. Therefore the operator can connect to the system. Using this connection he is able to control the PTZ-cameras, to have a look at (past) video sequences and to control components through the use of the server connected to the BAS. Integrating wireless devices is also possible (illustrated through the dashed line in the figure).

Digital systems can relieve the operator in his daily work. In the past the monitoring of areas was exhausting since most of the time no alarm condition happened. This period of time was wasted and could be used more efficiently through the security staff. It is sufficient if the security staff is alerted just in case of an event. In that case it is required that the operator works fully concentrated. Newer systems help reducing the time of watching the screens. Alarm bells will inform the operator of an event and the monitor will display the corresponding scene. Additionally further actions can be recommended (or even executed) by the system autonomously.

Therefore cameras have to fulfill special requirements (cf. [4] [5] [6] [7]). They need to be smart. In future suspicious scene detection can be done automatically using smart CCTV systems. Image and scene recognition can be implemented within the smart cameras and so recognition can be done directly on the spot. The network will be assembled with low-cost but intelligent cameras. If existing infrastructure will be used, a smart CCTV system can be installed

easily. Buildings are typically equipped with a BAS. Obviously, such on-the-spot cameras could be integrated into existing BAS infrastructure. Since the image analyzing part is done inside the camera, there is no need for video streaming over the network. Image transmission seams to be sufficient.

Concerning Figure 1.3 once again video surveillance could work autonomously without interaction of human beings. Then there would be no need for transmission of video streams. Reaction to events is done by the system automatically if intelligent cameras are able to detect them. Operators then are needed for control of the proper work of the system.

1.2 CCTV and BAS

If a BAS exists it is reasonable to integrate cameras and necessary components for CCTV systems into the existing infrastructure. Therefore a basic knowledge of building automation systems is needed. A BAS is a system with the following key aspects:

- Reduce costs by making the system working more efficient
- Save energy and time
- Protection of the environment
- Increase comfort

The major application domains are Heating, Ventilating and Air Conditioning (HVAC) and lighting/shading. Systems from the security domain (e.g., intrusion alarms) and safety domain (e.g., safety alarms) are integrated as dedicated subsystems. BASs are arranged in a three level functional hierarchy which is illustrated in Figure 1.4.

The sensors and actuators are placed in the field level. They, for instance, have a minimal data volume to transmit but a quick reaction time can be necessary therefore a simple two wire communication is sufficient. The next higher level is called automation level where control functionality is performed. Regulation of temperature and reaction to an input event (e.g., light switch) is processed. The zones are connected to each other and communication effort is higher than in the field level. The top level is called management level and is for supervision, monitoring and logging of the whole network. High bandwidth is available therefore a lot of data can be transferred and saved. It is used to create trends, too. They are in turn used for optimizing the system.

From an implementation point of view a two-tier model is the result. In the upper level the backbone is located where Ethernet as medium and IP as protocol is the de facto standard. In the lower area the nodes are sensors and actuators. They are working with minimal energy, simple layouts and are cheap in production and therefore need a simpler connection with power over wire when possible. In this area typically field bus systems are used which shall provide the following features:

- Support for many nodes (sensors and actuators)
- Low reaction times

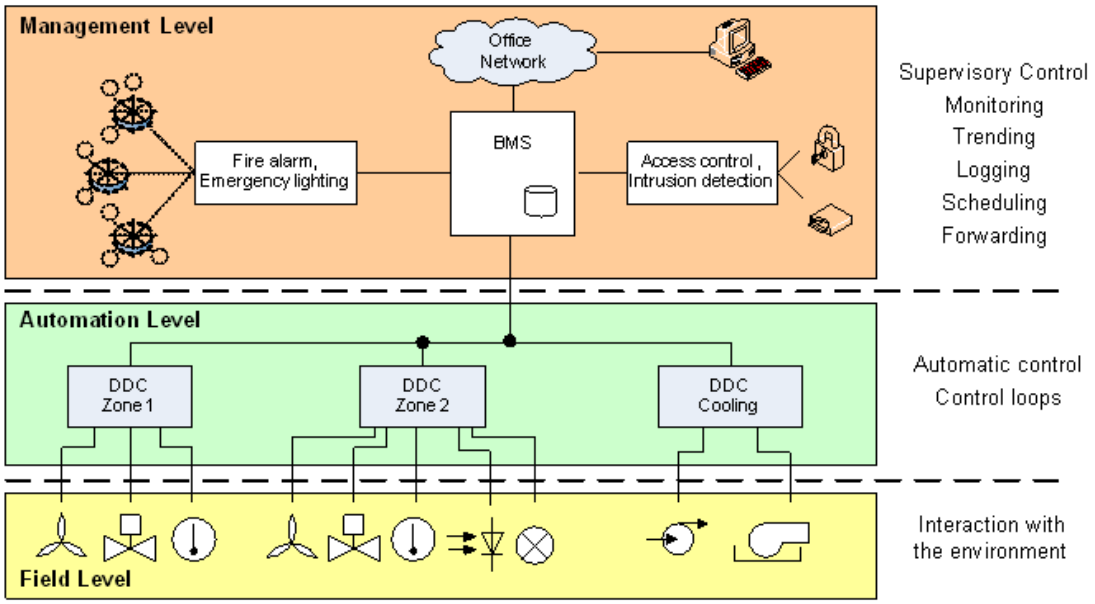


Figure 1.4: Conventional distributed three level model in BAS

- Robustness
- Power over wire
- Free topology
- Easy installation
- Low price

If CCTV systems should be integrated into such BASs these barriers have to be bypassed. However, the main advantages of integrating CCTV systems into BAS is that cameras can replace or support sensors.

1.3 Increasing sensor efficiency

In a BAS a lot of devices, nodes and in particular sensors exist. Nowadays they are basically used in one application only. Newer ideas facilitate connecting information of applications from different domains. Knowledge is joined together and results in a better overview of the current system status. This procedure is called sensor fusion and is explained in the following paragraph. Subsequently sensor sharing is announced, a possibility of using one sensor in different application domains at the same time.

Using both improvements at the same time a greater benefit is achieved. This includes reduced energy consumption, a wider field of application, lower risk of failures and lower installation costs.

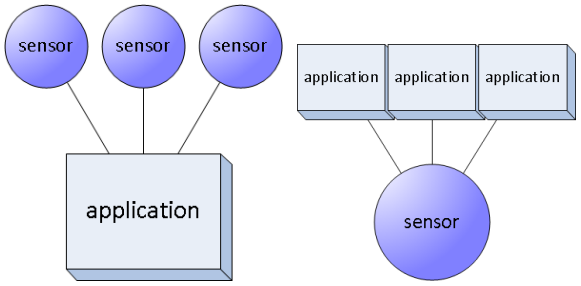
Sensor fusion

> "Sensor Fusion is the combining of sensory data or data derived from sensory data such that the resulting information is in some sense better than would be possible when these sources were used individually." [8]

Using several sensors, the output can be improved. So using different sensors (homogeneous or heterogeneous) can make the output more reliable. This means, if a camera is used for detection of a movement, an infrared sensor can be used, too. It will strengthen the result. Additionally, the robustness is improved. For example, a camera may be glared if sun is shining on a snow mantle. An additional infrared sensor can help avoiding this impairment. Aside from this the resolution of the result is higher because of different frequencies (a camera works in the frequency of visible light whereas the infrared sensor works in frequencies below) getting merged, thus covering a wider frequency range. Another aspect are outliers, they can be identified easier if more than one sensor is monitoring the same thing. Even - or especially - if they were heterogeneous. This reduces the risk of false alarms.

Another typical example for sensor fusion is the park distance control system where several ultrasonic distance sensors with partial overlapping coverage are used (cf. Figure 1.5a). Using cameras as additional sensors brings along the positive aspect to get a picture of the situation, too.

There are little differences in the terms *data fusion*, *sensor fusion*, *information fusion* and *multisensor data fusion* which for the matter of this theses will still be used equally.



(a) Sensor fusion (b) Sensor sharing

Figure 1.5: Increasing sensor efficiency

Sensor sharing

Different sensor devices can be combined to enlarge the fields of application. In Chapter 2 examples of use cases in different domains are described. They give an impression of the possibilities

of such applications. Using just one device (the camera) has different positive aspects. Less energy is consumed in total, less installation costs and maintenance costs are dedicated and there is no need for different protocols, which are just a few of these positive aspects.

Sensor sharing indeed means one sensor is serving more than one application domain as illustrated in Figure 1.5b. Therefore a sensor needs the ability to communicate with potentially different BAS protocols. Otherwise all connected nodes must use the same one. This on the other hand increases the protocol stack size and the required processing power which results in higher energy consumption. The easiest way is obviously using the same protocol for the whole network. That means, this protocol has to support every device type and transmission protocol used in such a network (or the other way round: each sensor uses the same kind of technology).

1.4 Contribution and Outlook

Currently CCTV systems are implemented as separated sub-systems. An integration into BAS is done if at all at the management level. For example, operator workstations that are used in the BAS for supervisory and visualization tasks can be equipped with an interface to the CCTV system. This way the operator has the opportunity to monitor the camera streams. However an interaction with the BAS is not supported. This way of integration has several drawbacks. Since the communication networks are physically separated, multiple communication lines and additional infrastructure are required. This results in higher installation and maintenance costs as well as an increased engineering effort. Since the integration is usually done using a central operator workstation a single point of failure exists. To overcome these problems the idea of this thesis is to integrate both systems into a single all-in-one solution. The benefits of the aimed approach are various. First the installation and management costs as well as the engineering effort can be reduced since only a single network infrastructure is needed. Second, instead of using a central operator workstation the control and management functionality can be distributed over devices that are even dedicated to different application domains. Therefore using sensor sharing and sensor fusion more complex and advanced applications are possible. Furthermore a single point of failure is avoided thanks to the use of a distributed approach.

Due to the contrary requirements of these domains such an integration is a challenging task. The main problem is that devices in CCTV systems need to exchange high amount of data. At the field level of BAS narrow bandwidth field bus systems are commonly used since robustness and efficiency are most important there. To be able to deal with the high amounts of CCTV data within field networks special mechanisms are necessary.

This thesis starts with the presentation of different use cases that benefit from such an integrated system (cf. Chapter 2). Afterwards an introduction to the most important open BAS technologies is given (cf. Chapter 3). Since the main problem of integrating CCTV systems is the required bandwidth the description is focused on the supported network media and their features. In the second part of this thesis the main contribution is presented, i.e. one possible approach to integrate CCTV systems into BASs. In Chapter 4 the general system architecture is shown. It also presents an analysis of the existing BAS technologies regarding their suitability for CCTV. The result shows that an integration is only possible in a limited way due to the low bandwidth of these networks. However transmitting single images is possible in a reasonable

time. Due to this limitation smart cameras are needed which perform event processing directly on-the-spot and transmit snapshots of the detected event only when necessary. To show the feasibility of this approach KNX as a representative of the wired domain and ZigBee as an important wireless standard are chosen for evaluation (cf. Chapter 5 & 6). To be able to integrate cameras within these two technologies the application models of both standards were extended since they provide no native support for this kind CCTV application. For KNX new Functional Blocks (FBs) and datapoint types were introduced. For ZigBee the Intruder Alarm System (IAS) device types were extended since they were insufficient for camera devices. To evaluate both extensions a proof-of-concept implementation has been realized.

CHAPTER 2

Use Cases

This chapter describes different application domains and their interaction with intelligent cameras. Therefore a specific use case out of every domain is chosen and illustrated using a Unified Modeling Language (UML) diagram. Additionally the interaction between the devices from different domains is shown. Obviously, these diagrams are not able to cover every possible behavior. Therefore the most useful ones are shown as examples here.

The purposes of the integration are various. It is possible to optimize the energy consumption or to increase the comfort. Other use cases tend to primarily increase the system's security or safety. As it will be shown use cases can also combine different positive aspects, too.

2.1 Ambient Assisted Living (AAL)

Ambient assisted living means monitoring, supervision or support of resident activities. In more detail it deals with the challenges in the context of ambient intelligence and applies its technology to enable people to live in their environment longer. Therefore the level of independence is improved and the psychological and physical state is encouraged [9] [10]. In Figure 2.1 a typical AAL use case is modeled. Actors are human beings and devices and both are placed outside the border. Inside the figure the activities are shown. Additionally, the domains are labeled, which actors are typical members of.

In this use case home and building automation devices are used to detect suspicious behavior. For example, an alarm is triggered if devices detect a longer absence. In addition, other sensors may also be used to monitor the behavior of individuals. If for a longer time no water is used or the refrigerator detects no opening of its door in a specified time period it could be assumed that an accident happened (e.g., a collapse of a person). An unused TV for an unusual long time duration may also be an indicator for a suspicious behavior. This may also be true for windows that are not closed during winter nights or lights that stay continuously switched on. Pressure sensors and intelligent cameras can be used to detect if the person accidentally fell in the flat leading to an informational alarm. The next step is the validation of this alarm and to

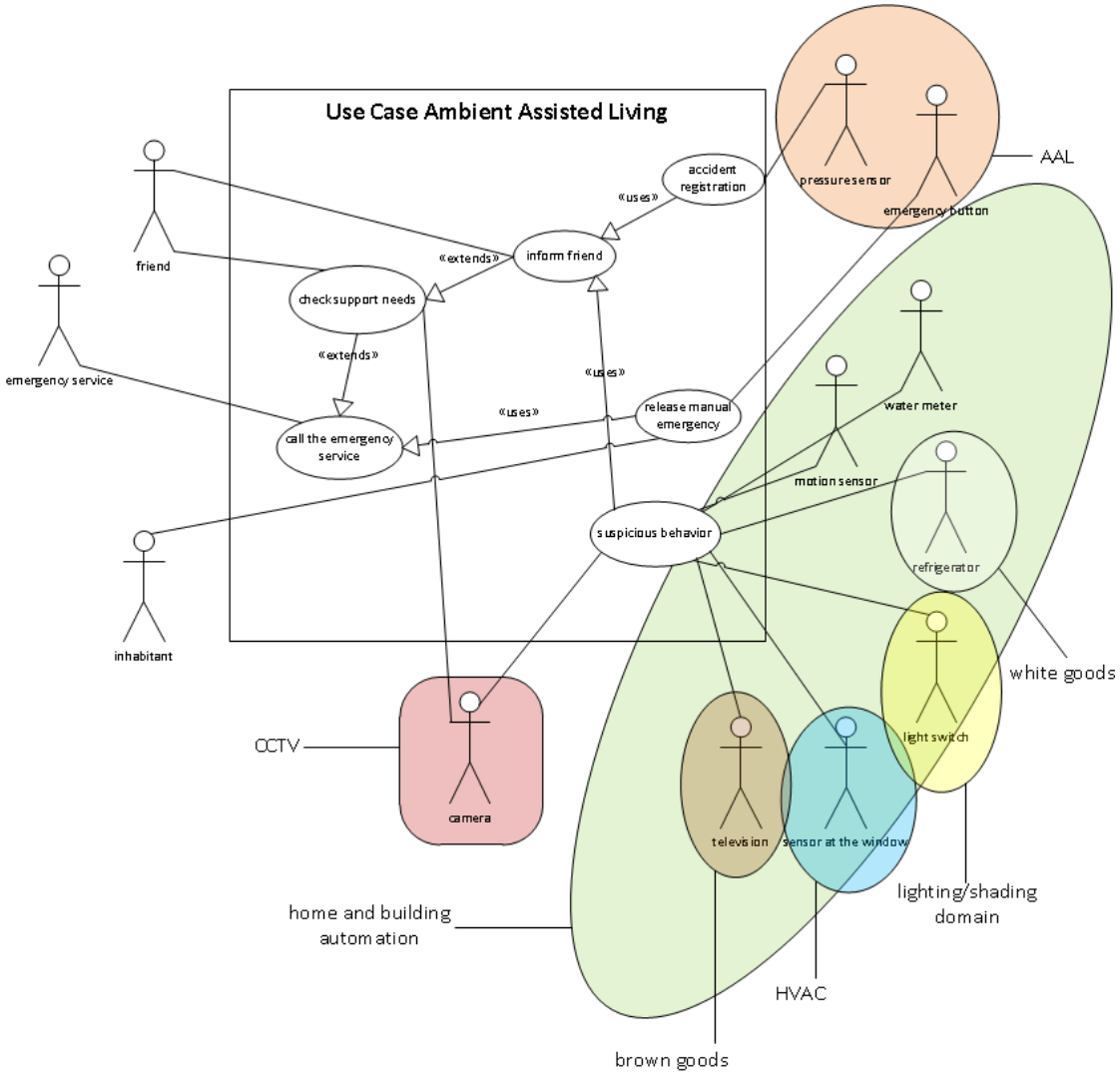


Figure 2.1: UML diagram – Ambient Assisted Living

take further measures. A confident may then contact the person. To speed up things the camera provides a snapshot. A collapse or any other emergency situation is recognized faster this way. Finally, the emergency service is informed. A different way to inform the emergency service bypassing the check routines is a manual emergency alarm. Therefore inhabitants may wear a wristlet with an emergency button.

2.2 Safety domain – Fire alarm

In the safety domain, incidents are effecting the risk of one's life directly and unintentionally. Fire alarm systems [11] [12] or social alarms are typical examples for that one. In general *safety* applications are regulated very strictly.

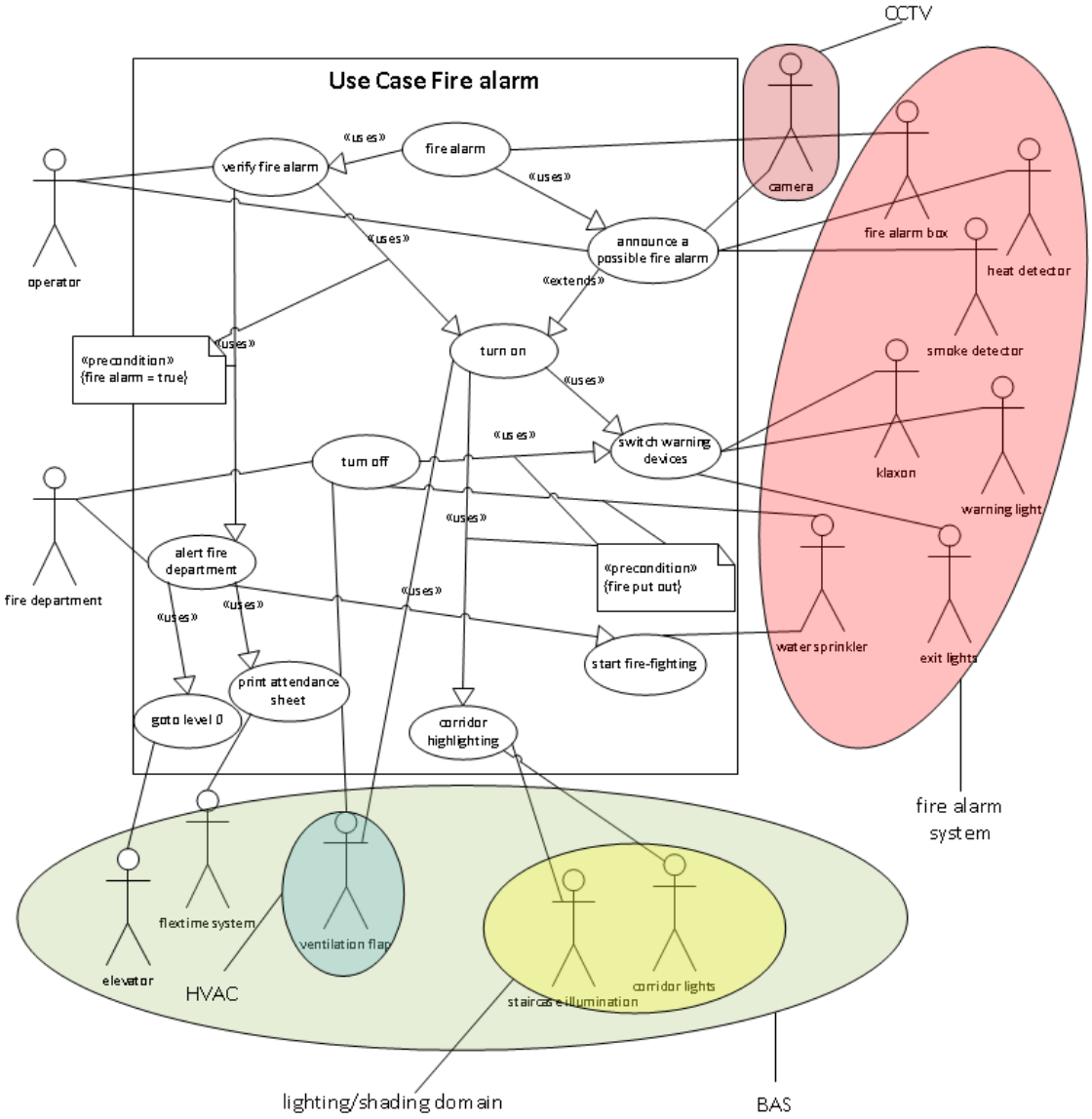


Figure 2.2: UML diagram – Fire alarm

The use case in Figure 2.2 shows a fire alarm system which is supported by the BAS. Traditional fire alarm systems operate with heat and smoke detectors. Manual fire alarm boxes

(pull stations) are provided, too. To decrease the amount of false alarms that are signaled to the fire department the security staff has the opportunity to cancel alarms as long as they did not originate from a manual fire alarm box in a predefined time. Therefore the staff has to appear personally at the alarm place and verify the correctness (or incorrectness) of it. If cameras are included in the fire alarm system, a snapshot can be transmitted to the operator panel where a decision can be taken.

In a distributed approach an alarm will be transmitted to the fire department and simultaneously different local actions can be executed automatically. After the activation of the fire alarm control panel, the klaxons and strobe warning lights will be turned on. People are prompted to leave the building (e.g., via announcements). Only when a fire is verified (through the operator) the water sprinklers or gaseous fire suppression are activated as these are expensive measures.

In case of a fire alarm the flextime system may print out a list of people having checked in, the HVAC system changes its state to a special fire mode which primarily focuses on smoke and heat extracting. The elevators move to the ground floor and shut down. Pursuing this idea further cameras will detect overcrowded emergency exits and therefore could re-route using dynamic exit lights. Also corridor lights can be switched on in a way that they lead to the next emergency exit. Therefore parts of the lighting and shading domain could be included, too.

2.3 Security domain – Protection

Security matters are incidents which occur intentionally aiming at destroying (or at least damaging) a system [13]. Human beings could get hurt consequently though. Examples for security applications are access control or intruder alarm systems.

The use case in Figure 2.3 shows a protection scenario. The focus of this scenario is an incident caused by intruders. CCTV systems can be used to improve motion detection since specific motion sequences could indicate crime scenes.

Whenever a person is discovered by the system its path is tracked by the camera. The trajectory is calculated and simultaneously analyzed. In collaboration with the BAS better results are possible. For example, the light can be adjusted to meet the brightness requirements of the camera. Sensor fusion enables the use of heterogeneous detectors. A sound sensor or a glass breaking sensor can be included in such a scenario.

In Figure 2.3 such interactions are illustrated. It also shows that third-party systems can be involved. For example, a car alarm is not part of an installed system in the building but its alarm can be detected by sound sensors from the BAS. On-the-spot cameras in parallel can detect the continuously flashing of the turn lights.

After alarm verification through the operator the warning devices get activated, the lights are turned fully on and the police is alerted. Sometimes the activation of klaxons and stroboscopic signal lamps already puts the burglar to flight. However, CCTV systems may be used to monitor all exits. If an alarm is verified, cameras can indicate a warning if persons are leaving the building and can take snapshots. Thus CCTV systems may help to solve such crimes.

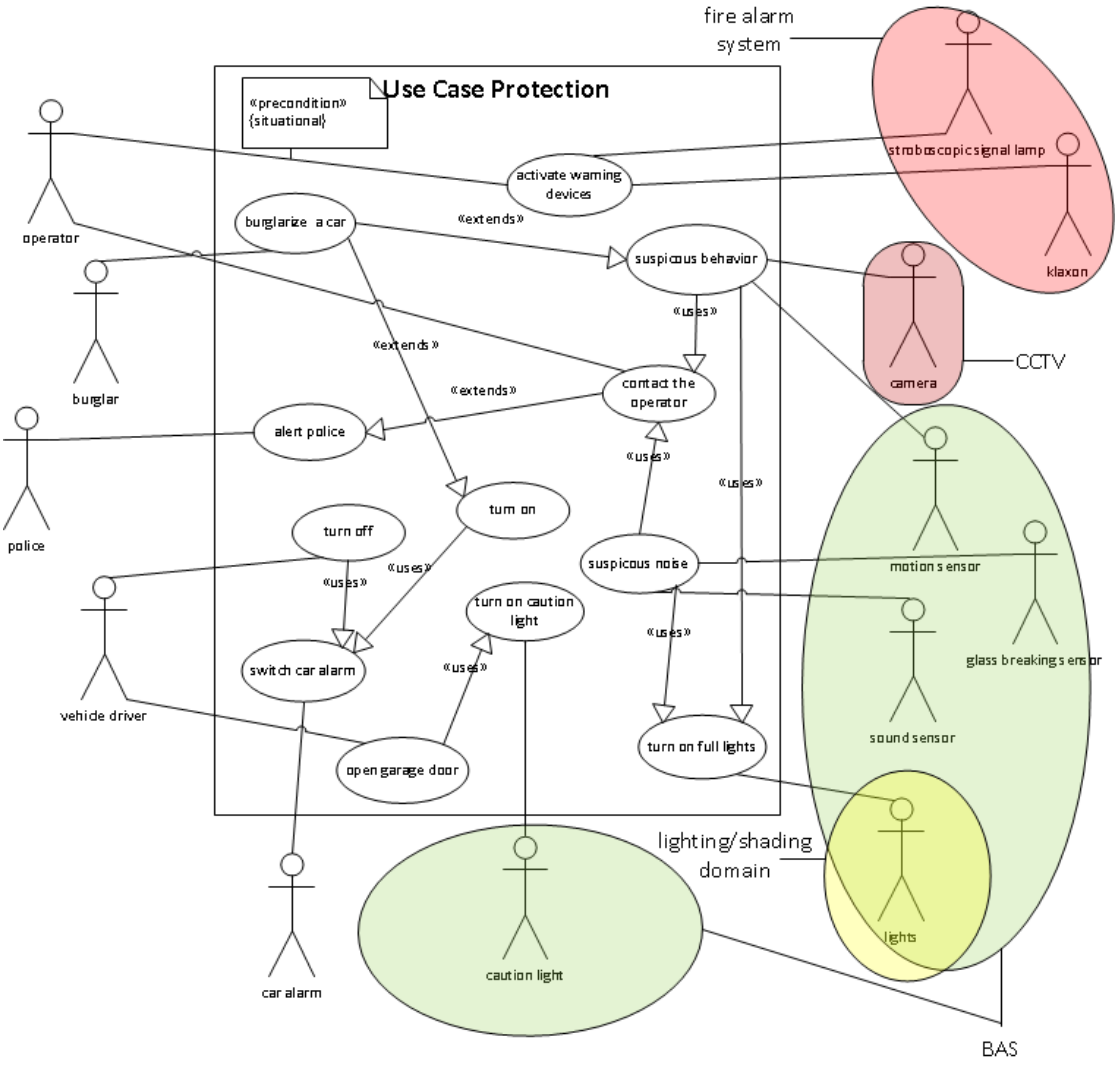


Figure 2.3: UML diagram – Protection

2.4 Advanced HVAC control

This use case utilizes the home and building domain to support the HVAC. This results in a more energy efficient use and a more comfortable environment.

Typical sensors out of the HVAC domain are the basis for regulating the heating, ventilation and air condition. A smoke sensor detecting the smoke after cooking will increase the light intensity. If more lights are turned on, more heat is emitted through the light bulbs. A smart control considers this effect and reduces heating power. Sensors report opened windows to a control unit. Heating is stopped for this time. This is the information a modern HVAC system

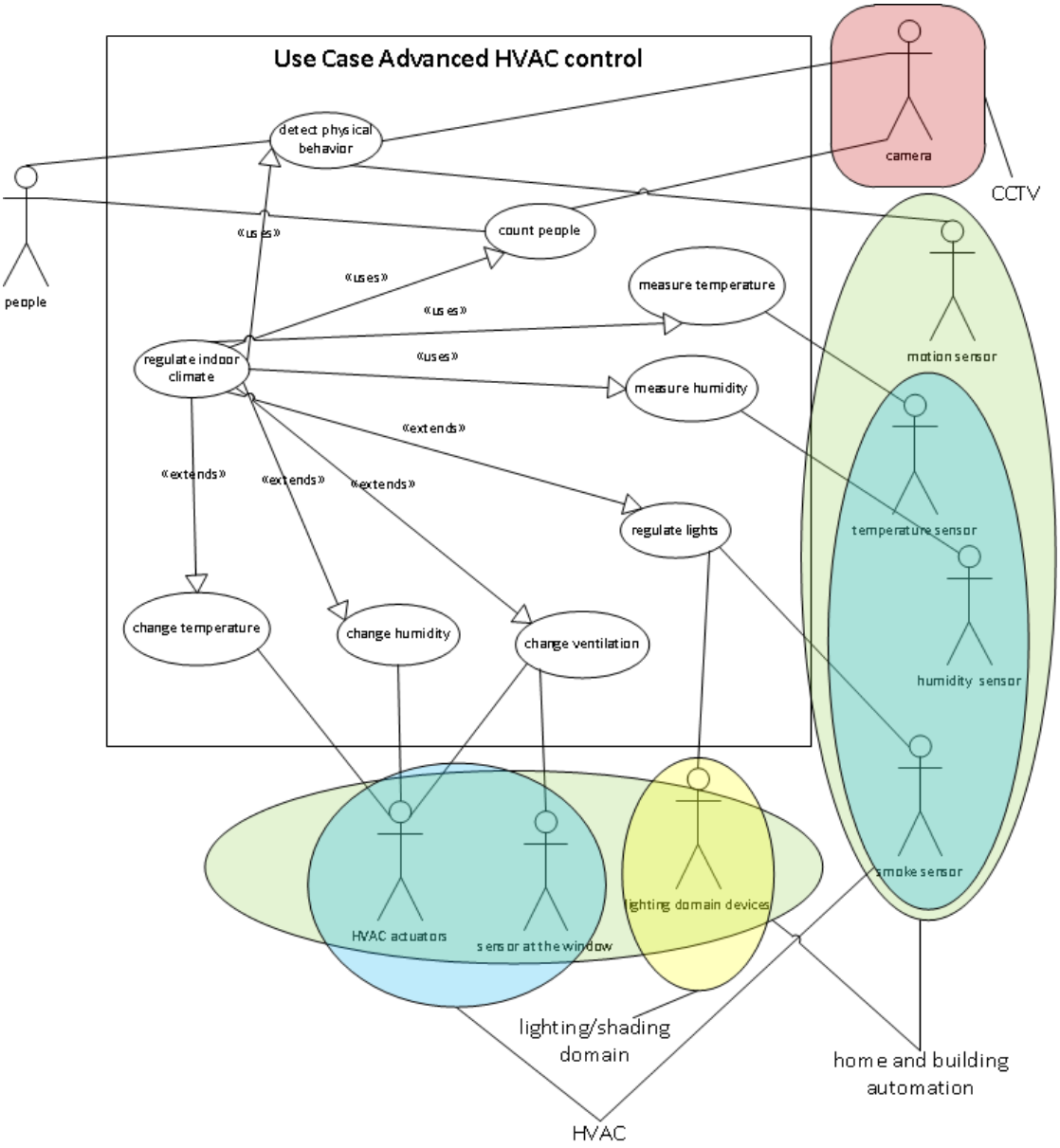


Figure 2.4: UML diagram – Advanced HVAC control

relies on.

In future applications CCTV systems can be used to improve the opportunities of the HVAC system. Installing on-the-spot cameras inside the building let the HVAC system easily detect the amount of people. Therefore, predictive measures can be taken (e.g., reducing the set point of the heating system). If their physical behavior is noticed, additional measures for the comfort

can be met (e.g., if the family is watching television the ventilation can be reduced to avoid noise). At the same time the information from the CCTV system can be passed to the lighting and shading system (e.g., a special TV mode is activated).

Including all this information the HVAC system is able to react to the inhabitant's needs in a more precise way. Figure 2.4 demonstrates the relationships between the participating devices and shows which information is needed for all sorts of actions.

CHAPTER 3

Technologies and their media in Building Automation Systems

In the following chapter the basics of different building automation system networks are explained. For the discussed protocols and standards an overview about their facilities as well as the provided network media will be given. Furthermore their relationship to the three level model (s. Figure 1.4) is announced. Moreover the involved organizations responsible for development and product certification are presented.

3.1 KNX

Based on the European Installation Bus (EIB) the first KNX standard was released in 2001 and updated in the following years. 2009 the current revision of the standard *KNX Standard v2.0* was published [14].

KNX is an open standard with particular focus on the field level. It includes a communication system which is compliant to five layers out of the seven layer International Organization for Standardization (ISO)/Open Systems Interconnection (OSI) model. For developing, programming and bus monitoring a software called Engineering Tool Software (ETS) is available. The KNX Association is a Belgian profit organization which certifies KNX products (and therefore guarantees the interoperability) and pushes the improvement of the KNX standard.

Devices are arranged in lines. Lines are coupled by a router to a main line resulting in an area. In Figure 3.1 multiple areas are shown. Devices labeled with *B* are bridges. The main use case of bridges is to extend network ranges. Each device in a KNX network can be addressed by two ways. One way is the individual address following the topology. Every node gets a unique device number. Devices with zero at the last digit are routers. Individual addresses are mostly used for configuration and management purposes. The second addressing scheme is called group addressing. Multiple devices may get the same group address representing communication relations between these nodes.

The standard specifies different transmission media, like Twisted Pair 1 (TP1), Power Line 110 (PL110), KNX Radio Frequency (RF), KNXnet/IP or KNX/IP. These media can be used homogeneously or combined in a cross-media network.

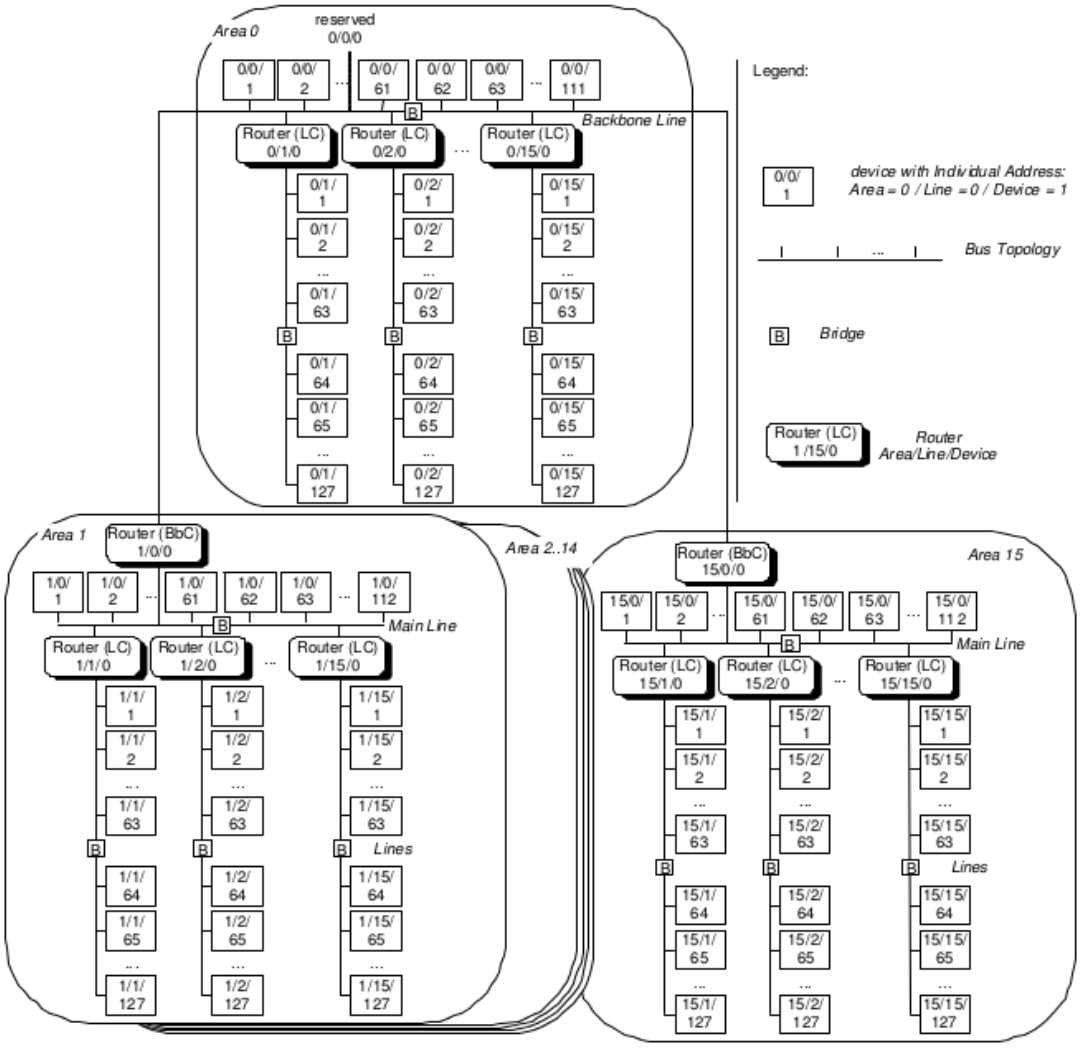


Figure 3.1: KNX topology [14]

Twisted Pair 1 (TP1)

A shielded twisted pair cable is typically used as medium. A benefit of this medium is the information and power transport over one wire pair at the same time. Maximum 12 mA per device are available at a voltage level of 30 V DC. The total amount of devices is 256 per physical segment, summing up to more than 65,000 devices per KNX network. Power Supply

Units (PSUs) are providing enough power for each device. The twisted pair medium can be arranged in a linear, a tree, a star or a mixed topology. A transmission rate of 9.6 kbps is available.

Bus arbitration is based on Carrier Sense Multiple Access with Collision Avoidance.

Power Line 110 (PL110)

With a bit rate of 1.2 kbps and a topology dependent on the electrical installation, power line is an alternative to TP1. Its benefit is the re-use of existent infrastructure, since no new cables have to be installed. At most 32,767 devices can be addressed.

Bus arbitration is handled via Spread Frequency Shift Keying (SFSK). For logical zeros a frequency of 105.6 kHz is used, for logical ones the frequency of 115.2 kHz is taken. Compared to Twisted Pair (TP) a negative aspect is the missing collision avoidance principle. If at any time more than one device is transmitting this results in a collision. In general data transfer over this medium is slow and prone to interference.

KNX RF

KNX RF uses the Industrial, Scientific and Medical (ISM) band for Short Range Devices (SRDs) which is located at 868 MHz. Telecommunication regulations for this band allow transmission in Europe only. Moreover transmission power is regulated. Thus, KNX RF devices send with a transmission power of 1-25 mW. Data is modulated using a Frequency Shift Keying (FSK) modulation. This results in a data rate of 16.4 kbps (chip rate = 32,768 cps).

For addressing in KNX RF a domain address is added to separate different networks inside the transmission range from each other. In general half-duplex unidirectional and bidirectional communication exists.

KNXnet/IP

Regarding the two-tier model introduced in Chapter 1, as common backbone in KNX IP networks can be used. Since the field level and its components are connected by conventional two wire communication or similar media, KNXnet/IP devices are encapsulating the KNX data frames. Two methods are available: KNXnet/IP routing for interconnecting KNX field networks using IP multicast and KNXnet/IP tunnelling for opening a management connection to KNXnet/IP routers/gateways using a unicast connection.

KNX/IP

Meantime, there exist even native KNX IP devices, using an IP network as communication medium. For communication with other KNX IP devices as well as KNX field networks via KNXnet/IP routers, KNXnet/IP routing is used. Additionally, special measures inside this protocol take care of flow control [14].

3.2 BACnet

BACnet is a vendor independent protocol standard for Building Automation and Control Networks [15]. It is mostly used at the management and automation level. It specifies services for accessing and manipulating data. BACnet is developed by the American Society of Heating, Refrigerating and Air-Conditioning Engineers (ASHRAE). 2003, it became an ISO standard. Since then, continuously refinements and extensions are added via addenda which are included from time to time in the BACnet standard.

BACnet devices which are coupled together in various topologies (cycles have to be avoided) form a so called BACnet network. Inside this network transmission is medium dependent. To enlarge network ranges repeaters and bridges can be used. Multiple networks are usually coupled together using routers (Figure 3.2). Messages can be transmitted using any network technology by encapsulating the BACnet messages. However, to increase the compatibility between devices of different vendors seven so called network options are specified.

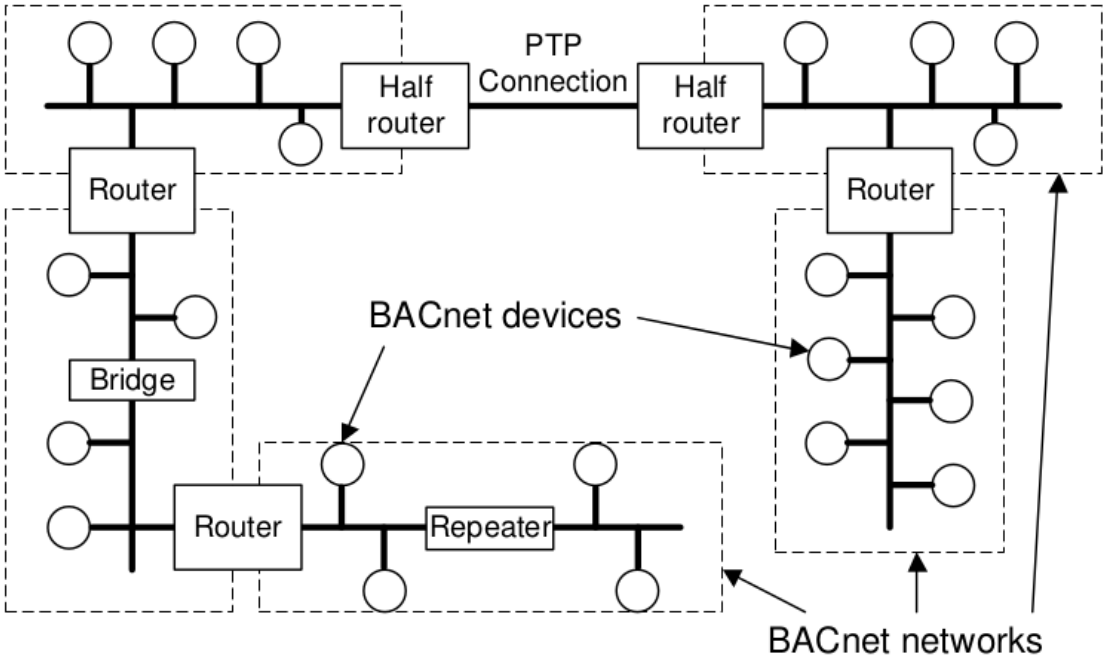


Figure 3.2: BACnet topology

Concerning the ISO/OSI model, BACnet just defines level three and seven. Therefore every medium (level one and two) can be used. Currently there exist definitions for the following network options[1]:

[1] Also for KNX a prototypical mapping is available.

Ethernet/ISO 8802-3

With Ethernet high data rates are possible. At the same time a Carrier Sense Multiple Access with Collision Detection (CSMA/CD) principle is implemented.

ARCNET

ARCNET provides no priorities, but is deterministic which results in a better performance in relation to the network load. It is based on a token passing mechanism. However, ARCNET was mostly displaced by Ethernet.

MS/TP

With a Master-Slave/Token Passing mechanism the topology connects each master in a logical ring. The token is then passed from master to master. Once the token has been received, every master can communicate with its slaves. As physical layer, EIA 485 is used. It is a multidrop BACnet network which uses differential signal encoding.

PTP

It is used for connections between half-routers. Since it enables a point-to-point connection, EIA 232 is used as physical connection.

LonTalk

Using a foreign frame as Application Protocol Data Unit (APDU) type LonWorks' LonTalk protocol can be used. Therefore every LonWorks medium is possibles as physical/data link layer. Nevertheless, integrating native LonWorks devices into this BACnet network is not possible.

ZigBee

ZigBee can be used as a wireless network option for BACnet. It is part of the BACnet standard since 2010. BACnet unicasts are encapsulated into ZigBee unicasts (only the addressing scheme is different). BACnet broadcasts are translated to ZigBee multicasts having all devices in the same group.

BACnet/IP

BACnet also specifies the use of User Datagram Protocol (UDP)/IP as native data link layer protocol (BACnet/IP). Using this scheme, an IP network can be used as native BACnet medium which may host BACnet IP only devices. Broadcasts are done using a special BACnet device or using IP multicast instead.

3.3 LonWorks

LonWorks was published in 1996 by Echelon Corporation. It is a universal and vendor independent field bus system. Like KNX, it is mostly used at the field and automation level. The LonMark Association was founded in 1994, consists of about 500 members and is among other tasks responsible for certification of products. The technology comprises of the LonTalk protocol (layer one to seven) and LonWorks nodes. Additionally a management software (LonWorks Network Services (LNS)) and guidelines for the interoperability are provided.

The LonTalk protocol includes all seven layers and provides services for unicast, multicast and broadcast. By observing the addresses the routers subsequently learn about the location of subnets. This leads to a minimal effort in organizing the network.

LonWorks nodes consist of a neuron chip which implements the LonTalk protocol stack. To extend the functionality a host chip might be added. It supports larger applications by providing more performance and more memory for the applications which are written in Neuron-C. Neuron-C is with some restrictions similar to ANSI C.

Devices are arranged physically in subnetworks. They are connected using routers and bridges. In a logical view communication groups are created. An illustration of the LonWorks network structure is given in Figure 3.3. LonWorks enables broadcasting to reach all devices in a domain, multicasting for transmitting messages to all nodes in a group and unicasting to address single nodes.

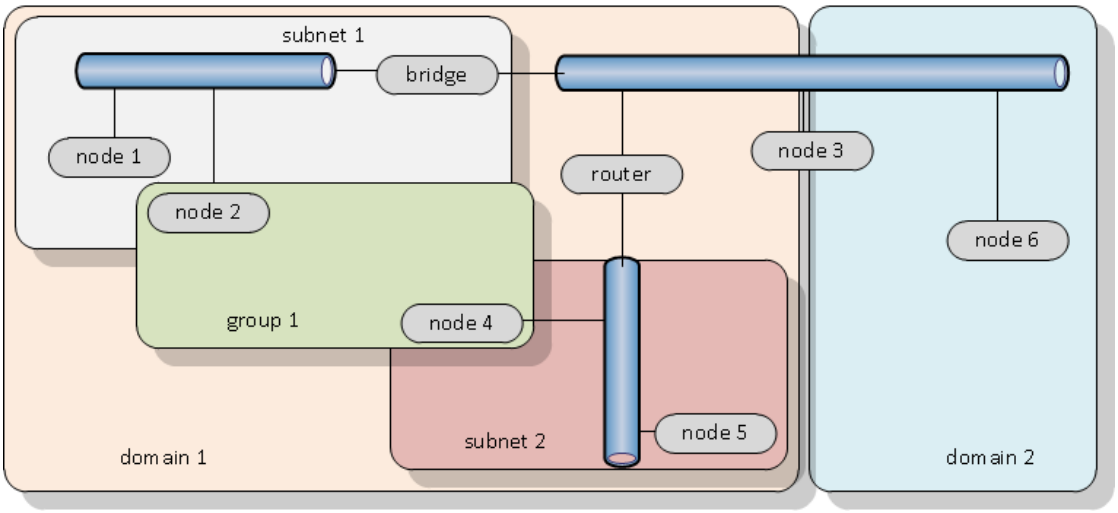


Figure 3.3: LonWorks topology

Beside the following physical layers, EIA-485, coax cable, Infrared (IR) and optic fiber are specified, too.

Twisted Pair

Basically, the data rate is 78.125 kbps in Free Topology 10 (FT-10) with a maximum bus length of about 600 m. If power is supplied, it is called Link Power 10 (LP-10). In both cases every topology is possible. Each segment needs a termination impedance. TP uses a predictive p-persistent Carrier Sense Multiple Access (CSMA) mechanism.

Power Line

Similar to KNX PL110 the Power Line (PL) topology is related to the electricity network. PL exists in different channel types. Again, as arbitration scheme predictive p-persistent CSMA is used [16]. Due to its interference liability a maximum data transfer rate of 10 kbps is possible.

3.4 ZigBee

ZigBee is a two-way wireless communication standard [17] based on the IEEE 802.15.4-2003 [18] standard[2]. It was first introduced in 2005 (ZigBee 2004 Specification). Since then two further versions were published: ZigBee 2006 Specification and the enhanced ZigBee PRO 2007 Specification. ZigBee specifies the network and the application layer and provides security services (cf. Figure 6.1). It is a standard mostly used at the field or automation level with the goal to support very low-cost and low-power devices.

The ZigBee development is organized by the ZigBee Alliance which was founded in 2002. The ZigBee Alliance describes itself as an open, non-profit association of members.

A ZigBee (IEEE 802.15.4-2003) Personal Area Network (PAN) consists of Reduced-Function Devices (RFDs) and Full-Function Devices (FFDs). An RFD has the capability of being a ZigBee End Device (ZED) whereas an FFD can be employed as ZED, as ZigBee Router (ZR) or even as ZigBee Coordinator (ZC). In Figure 3.4 different topologies of a ZigBee network are illustrated.

In a mesh topology (also called peer-to-peer network) every FFD can communicate with every other device inside the network and inside the device specific range. Additionally, a ZC is needed for organizing the PAN. If networks are connected together a so called (cluster) tree network can be built. Inside such a network multiple ZCs coexist. The very first ZC is still responsible for the whole tree.

In contrast, in a star topology communication is handled exclusively over the ZC.

In the ZigBee specification three RF bands predefined by the IEEE 802.15.4 standard are in use.

868/915 MHz

Due to different ISM RF bands in the world, ZigBee (IEEE 802.15.4) uses two sub-gigahertz RF bands (not at the same time). In America and Australia frequencies between 902 and 928

[2]Though in meantime IEEE 802.15.4 standard evolved to IEEE 802.15.4-2006, ZigBee in its current version is based on the older version of this specification.

[3]http://www.icpdas.com/products/GSM_GPRS/zigbee/images/zigbee_topology.jpg [10.11.2011]

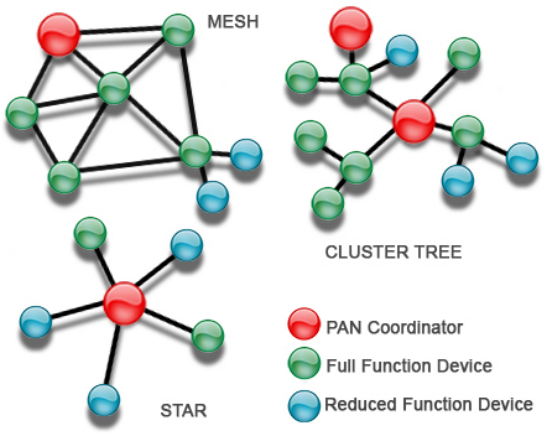


Figure 3.4: ZigBee topologies[3]

MHz are used, in Europe they are between 868.0 and 868.6 MHz.

In 2003 the use of Direct Sequence Spread Spectrum (DSSS) with Binary Phase-Shift Keying (BPSK) modulation was specified for these frequency ranges. Three years later the standard was extended and the use of DSSS Physical Layer (PHY) employing Offset Quadrature Phase-Shift Keying (O-QPSK) modulation and again optional Parallel Sequence Spread Spectrum (PSSS) PHY employing BPSK modulation were added.

The bit rates range from 20 kbps up to 250 kbps. Since ZigBee is based on the IEEE 802.15.4-2003 standard the two newer modulation modes are not available. This results in a limited bit rate (20 kbps in Europe; 40 kbps in America and Australia).

The differential encoder receives the binary data which should be sent and applies a logical XOR function. Then the bit-to-chip function converts the bit into a 15-chip value. This is a pseudo random noise. The last step is the BPSK modulation on the carrier.

2.4 GHz

In the 2.4 GHz band ZigBee uses a DSSS PHY employing O-QPSK modulation.

The binary data of each octet get divided into two symbols using a bit-to-symbol function. First, the four Least Significant Bits (LSBs) are mapped into a symbol, then the remaining bits (the four Most Significant Bits (MSBs)) are mapped into a second symbol. Using a symbol-to-chip spreading function, each symbol gets converted into a 32-chip pseudo random noise value. There are 16 (2^4) different chip values, defined by the IEEE 802.15.4-2003 standard. Afterwards an O-QPSK modulation modulates the even-indexed chips onto the in-phase carrier and the odd-indexed chips onto the quadrature-phase carrier with the LSB first. To generate an offset between I-phase and Q-phase, the Q-phase is delayed for the inverse of the chip rate.

This way a data rate of 250 kbps is possible.

CHAPTER 4

Integrating CCTV systems into Building Automation Systems

4.1 System architecture

When integrating CCTV systems into BAS, an adequate system architecture has to be chosen. There are three different communication models for BASs. A communication model describes the way how devices within the BAS communicate with each other. Typically BASs are arranged in a tree topology although different other topologies are also possible.

Centralized approach

In the *centralized approach* different parts of the network are connected via a gateway to a higher instance (e.g., control center). This communication model often follows the *client-server-model* [19]. If this instance is, for example, a control center the security staff is able to take over control of every action (e.g., turning on the water sprinklers in case of fire). False alarms from malfunctioning sensors can be easily detected and eliminated by manual intervention. However, if the central instance is defect the whole system stops working. In Figure 4.1 an application scenario is given. It shows a typical two-tier BAS. Subnets are separated by their application domain. So one is responsible for the HVAC and lighting domain, another one for access control and the third one is capable of motion detection devices. As mentioned later every subnet may use its own protocol. This way the benefits of the different protocols can be used where they are needed. In this example a camera (acting as an integrated CCTV system) detects motion in an area. It sends this information to the operator workstation which can, for example, open a door automatically. As another reaction the lights could be turned on using the lighting and shading domain.

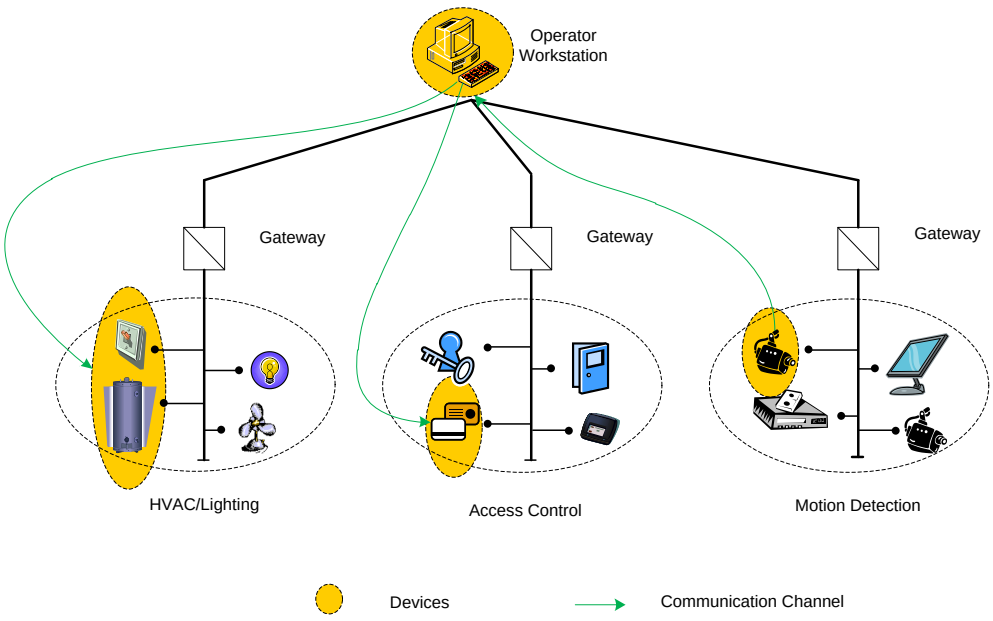


Figure 4.1: A sample centralized approach

Single point of failure

One drawback is that the whole communication happens via the central instance. On the one hand it is a benefit that all information is available at a single point, but if this instance stops working the communication in the network is interrupted. A highly reliable system is needed for this main instance, however this may increase the installation costs (e.g., redundant energy supply). Furthermore central instances are prone to security attacks.

Loss of information

The central instance may also act as a gateway. This allows the use of different media and protocols. So it is possible to have one subnetwork working with ZigBee 2.4 GHz, while another one operates on KNX TP1. The devices within the specific subnetwork just need to be able to process their own network protocol and do not have to worry about protocols in other subnetworks. Gateways are responsible for connecting them. Therefore they translate packages from one communication protocol to another one. Unfortunately this mapping leads to an information loss. The amount of lost information depends on the translation quality. Also extra time for translation is needed which may result in communication delays.

Bottleneck

Actually the higher layer (*backbone*) within this two-tier architecture contains links with major bandwidth and the central instance is usually not an embedded device. Nevertheless, this central instance is some kind of a bottleneck in this network, in particular if a large number of devices communicate with each other. If the central device needs to perform complex tasks which require high computational time, the hazard of a bottleneck is even higher. This slows down response time for the whole network. So, every domain or subnetwork is affected by that. In the worst case this affects the proper execution of even simple tasks like switching on some lights.

Engineering effort

In large systems the network structure and the available data points of devices have to be known by the gateways. This means there is an engineering and configuration effort for every gateway to ensure it can communicate with each device of interest used in this network. This effort escalates if the structure of the network is modified at some time which may cause a disproportional rising of the error probability, too.

Distributed approach

In a *distributed approach* a failure of an individual component involves no blackout. At most only a part of the network stops working. Since routers connect the network together, each device can communicate with every other device in the network. For example a fire alarm can start the klaxons immediately. But the drawback is that there is no possibility to check if an event is identified correctly. Configured initial actions will start immediately. Consider a smoke detector which detects a wrong fire. If this smoke detector is configured to turn on the water sprinklers instantly and send an alarm message to the operator at the same time, the water sprinklers will start before the operator is able to stop them. Therefore only measurements without destructive effect shall be applied as such initial actions. It depends on the situation, but it is conceivable that this sensor starts strobe lights and klaxons directly. Figure 4.2 shows a decentralized approach again with the same domains like in the example of the centralized approach. Note that the subnets are connected by routers. This means that all subnets must use the same protocol because routers do not translate messages above the network layer.

System monitoring

To monitor the activities a device can be connected to the backbone. It does not influence the transfer times at all. However, this device can become active and for example play the roll of a management station. Trend statistics can help to identify frequent sent messages, to observe the workload at different daytimes or to help to detect failures.

Reaction time

By activating devices directly through the alarming device time can be saved. In a centralized approach this is not possible since alarms are verified by the operator (or the single instance

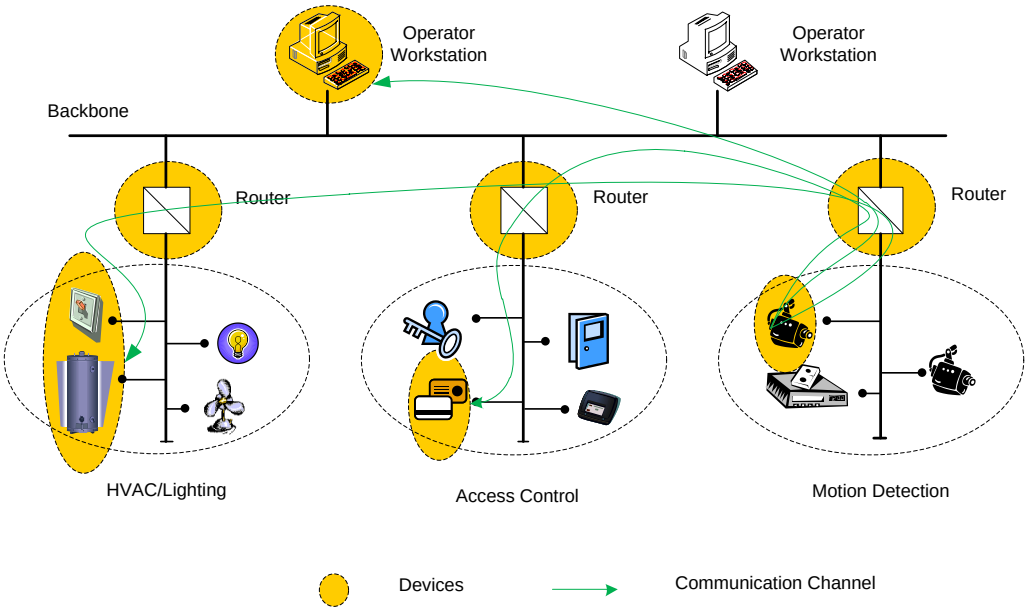


Figure 4.2: A sample decentralized approach

automatically) before any action will be taken. In case of events not belonging to the safety or security domain, this is negligible. But if a critical event is detected useful time is wasted.

Hybrid approach

To get the best out of the two different architectures a hybrid approach is chosen. In addition to the distributed approach a top element (e.g., operator panel) is added. With this setting the positive aspects of both approaches are unified. For example this means that the sensors activate the klaxons immediately while the water sprinklers are turned on after verification through the security staff. The warning system is turned on without loosing time while the need of systems which produce high costs or having destructive effects are verified before being activated. Moreover bus monitoring (for trend analysis) is still possible.

Another positive aspect is the chance of mixing routers and gateways where necessary. So every functional domain can use its protocol. Since the protocols are historically developed on a domain basis every standard has its advantages for an application domain. Mixing them together retrieves the best for all, although this means a high effort in installation and maintenance.

4.2 Suitable building automation network protocols

Since a hybrid approach as general system architecture has been chosen, heterogeneous existing network protocols may be used. To be suitable for the use within CCTV systems, existing BAS

protocols have to be analyzed. The most important difference between CCTV systems and BASs is the amount of data that can be exchanged. While in the building automation domain a few bytes or at most a few kilobytes are usually transmitted, the amount of data within CCTV is in the order of megabytes. Therefore, since the data rate and transmission duration of BAS protocols is the most limiting fact, they will act as basis for selecting suitable protocols. Thus, an analysis of the transmission duration of available BAS protocols is presented within this section.

Table 4.1 gives a first impression of the transmission duration of different protocols and communication media. The basis for the calculation is an image with a size of 30 kB. An overhead of 30% was added, so 39 kB is the total amount of data. This is just a first rough calculation because the real overhead is protocol dependent. Using the specifications of different BASs, the bit rates were identified and the times needed for transmission were calculated using Equation 4.1 with t_G as the overall time in seconds, d as the amount of data in bits and r as the data rate of the protocol in bits per seconds. Equation 4.2 shows the calculation for KNX TP1.

$$t_G = \frac{d + 30\%}{r} \tag{4.1}$$

$$32.5 = \frac{312,000^{1}}{9,600} \tag{4.2}$$

Protocol	Medium	Rate	Duration
KNX	TP1	9.6 kbps	32.5 s
KNX	PL110	1.2 kbps	4 m 20 s
BACnet	EIA-232	9.6-56 kbps	32.5-5.6 s
BACnet	EIA-485	9.6-76.8 kbps	32.5-4.1 s
BACnet	ARCNET	2.5 Mbps	0.2 s
BACnet	ISO8802-3 [2]	10 Mbps-10 Gbps	$\leq$0.04 s
LonWork	TP	78.125 kbps	4 s
ZigBee	868/915 MHz	20-40 kbps	15.6-7.8 s
ZigBee	2.4 GHz	250 kbps	1.3 s

Table 4.1: Bit rates and transmission times roughly calculated

The highlighted entries are calculated in detail. KNX TP1 is chosen to be the representative for wired networks. A prototypical implementation for BACnet is already available [20]. ZigBee enables high transmission rates using the 2.4 GHz band and is therefore the chosen representative for wireless BAS networks.

An overview of reasonable limits on the amount of data is given. Furthermore, the exact duration which is needed to transfer a given amount of data using the chosen standards is presented.

[1] In this calculation a conversion factor of 1,000 between a byte an its bits is chosen.

[2] Ethernet

Using KNX TP1 as transmission protocol

KNX TP1 provides the standard frame format (cf. Figure 4.3) and the extended data frame format (cf. Figure 4.4). Depending on the length of the payload the corresponding frame type is chosen dynamically.

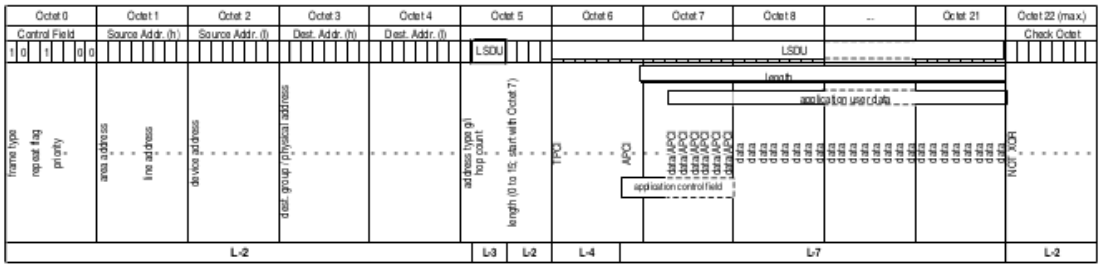


Figure 4.3: KNX TP1 standard data frame [14]

The first octets of the frames contain header data which is explained in this paragraph. Differences between the standard frame and the extended frame are denoted. The very first octet is the control field. Two bits specify the frame type: standard data frame, extended data frame, poll data frame or acknowledgment frame. If the data frame is repeated, the next bit is set. The following two bits define the priority of the frame. In KNX TP1 four priorities are available (low, normal, system and urgent). If the control frame indicates an extended data frame the next octet is the extended control field. It includes the address type, the hop count and the extended frame format type. Next the address fields for the source and destination addresses follow. Due to the 16-bit addressing scheme in KNX TP1 two octets for the source and two octets for the destination address are necessary. In each case the first octet contains the higher bits. In an extended data frame the next octet specifies the length of the frame. In a standard data frame, the next octet contains address type and the hop count. The following 10 bits describe the transport protocol control information and application protocol control information. The very last octet is a check sum. It is a logical NOT XOR function over the preceding bits of the frame.

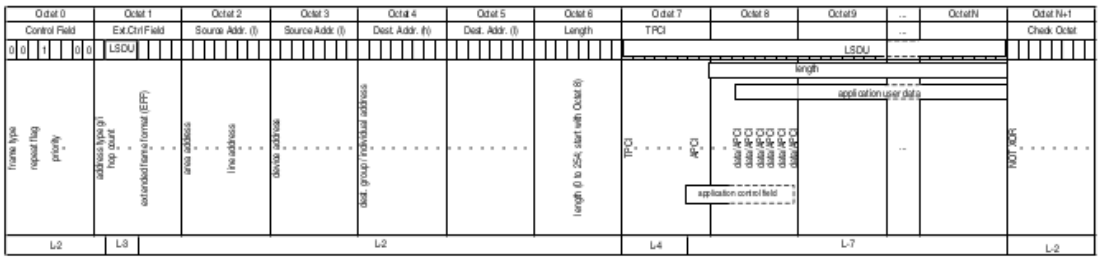


Figure 4.4: The KNX TP1 extended data frame [14]

With this information the payload size can be calculated. Table 4.2 gives an overview. A few more notable things are explained in the following paragraph.

description	octet	UART real bits	KNX TP1 bits	time [μs]	user data [bit]
single data octet	1	13	8	1352	8
short break (before ack)		15	0	1560	0
long break (low, normal)		53	0	5512	0
long break (system, urgency)		50	0	5200	0
extended data frame	263	3419	2104	355576	2024
standard data frame	23	299	184	31096	112
acknowledgment frame	1	13	8	1352	0

Table 4.2: Overview of KNX TP1 data frames

The communication data in a KNX TP1 packet are surrounded by one start bit in front of the eight data bits and one parity bit followed by one stop bit at the end. After the stop bit a duration of two bits is required until the next start bit can be sent (cf. Table 4.2). A bit in KNX TP1 takes 104 μs to be transmitted. Between a data frame and the corresponding acknowledgment frame a break of 15 bit times is required. According to the priority the next transmission has to wait at least 50 bit times. In the above calculations, a free medium has been assumed. In case of a line busy detection a device has to wait until the line is free again and before retrying to send its data. With the chosen settings just normal priority is used therefore the longer break is necessary.

The effective data rate in KNX TP1 is 5.56 kbps if extended frames are used.

Using ZigBee as transmission protocol

The challenge of calculating the transmission time for the ZigBee protocol is the interaction of the ZigBee specification with the IEEE 802.15.4-2003 specification. Also a few things are just random factors, which will be denoted in the following paragraphs. The basic settings are a transmission on a free band, with both, the transmitter and the receiver, inside their ranges in the 2.4 GHz band. Data frames are used with a size of 25 KiB for data.[3] The transmission is in a beacon-disabled PAN and a ZED transmits data to a bound ZC or ZR.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA)

First thing to do is to check if the band is free. Therefore an unslotted CSMA/CA mechanism is used (Figure 4.5). The algorithm uses time units called backoff periods, where one is equal to *aUnitBackoffPeriod*. Because of the settings of the devices it may happen that devices in a PAN use different values for *aUnitBackoffPeriod*. In the following this is not considered. At the beginning of the algorithm *NB* is set to zero and *BE* is set to *macMinBE*. *NB* is the counter of how

[3] Of course more data can be transmitted, too, but this means bits of the payload field are needed for an additional user fragmentation which results in a decreasing data rate.

often it was tried to access the medium. If it exceeds *macMaxCSMABackoffs* the transmission fails. *BE* is the backoff exponent, it defines how many backoff periods a device has to wait until it tries to assess the channel. In our example *macMinBE* is set to three. If it is zero, collision avoidance is disabled in the first run of this algorithm.

With this information the algorithm can start. After setting the variables the MAC sublayer delays a random period from zero to $2^{BE} - 1$.

$$\begin{aligned} random(2^{BE} - 1) \\ random(2^3 - 1) \\ random(8 - 1) \\ random(7) \end{aligned} \tag{4.3}$$

In our example, the value is from zero to seven. The following calculation is based on a delay of three backoff periods.

Afterwards a Clear Channel Assessment (CCA) is performed, which is equal to eight symbol periods by definition. If the channel is idle (which was an assumption) the first frame is going to be transmitted. The whole CSMA/CA length is 68 symbols or 34 octets. The duration is calculated with the predefined symbol rate (62.5 ksymbol/s from the ZigBee specification) and is 1088 μs.

Data frame

ZigBee provides two different data frame types. On one hand there is the single data frame with a maximum of 102 octets of payload for the application layer. On the other hand there exists a fragmented data frame with a maximum of 100 octets for the application layer in one such frame. These values are based on the *aMaxPHYPacketSize*[4] value which is 127 octets.

In a bottom-up view, the effective payload lengths for each layer are calculated (Table 4.3 and Table 4.4 show the summary):

[4]"The maximum PSDU size (in octets) the PHY shall be able to receive." [18] PSDU is the PHY service data unit, a different labeling for the PHY payload. So this is not the maximum frame size on the physical layer as the Synchronization Header (SHR) and PHY header (PHR) are added.

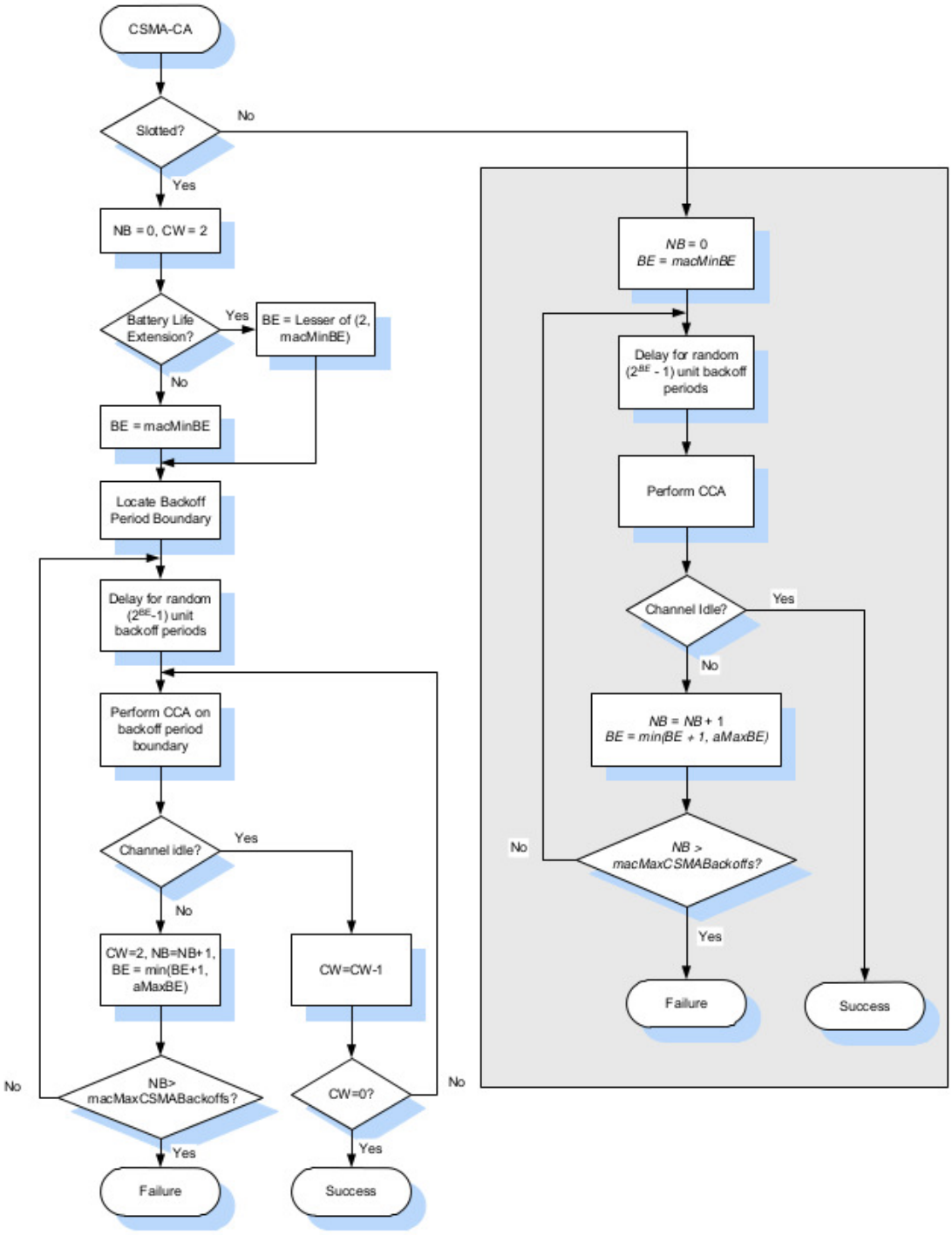


Figure 4.5: CSMA/CA mechanism (with the highlighted unslotted part) [21]

Frame	Part	Field	Data [octet]	Data_fragmented [octet]	ACK [octet]	ACK_fragmented [octet]
APDU	APS header	Frame Control	1	1	1	1
		Destination Endpoint	1	1	1	1
		Group Address	0	0	0	0
		Cluster Identifier	2	2	2	2
		Profile Identifier	2	2	2	2
		Source Endpoint	1	1	1	1
		APS Counter	1	1	1	1
		Extended Header	0	2	0	3
	APS payload	Frame Payload	102	100	0	0
NPDU	NWK header	Frame Control	2	2		
		Destination Address	2	2		
		Source Address	2	2		
		Radius	1	1		
		Sequence Number	1	1		
		Destination IEEE Address	0	0		
		Source IEEE Address	0	0		
		Multicast Control	0	0		
		Source Route Subframe	0	0		
	NWK payload	Frame Payload	110	110		

Table 4.3: Different frame lengths for the ZigBee layers

Frame	Part	Field	Data [octet]	Data_fragmented [octet]	ACK [octet]	ACK_fragmented [octet]
MPDU	MHR	Frame Control	2	2	2	2
		Sequence Number	1	1	1	1
		Destination PAN Identifier	0	0	0	0
		Destination Address	0	0	0	0
		Source PAN Identifier	2	2	0	0
		Source Address	2	2	0	0
		Auxiliary Security Header	0	0	0	0
	MAC payload	Frame Payload	118	118	0	0
	MFR	FCS	2	2	2	2
PPDU	SHR	Preamble	4	4	4	4
		SFD	1	1	1	1
	PHR	Frame length	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$	$\frac{7}{8}$
		Reserved	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
	PHY payload	PSDU	127	127	5	5

Table 4.4: Different frame lengths for the IEEE 802.15.4 layers

Physical Layer – PPDU Main goal of this layer is the synchronization with the preamble sequence and the Start-of-Frame Delimiter (SFD) fields. Moreover it contains the length of the payload field. The SHR contains a preamble with four octets and the SFD is one octet in length. This is specified in the IEEE 802.15.4 standard. The SHR is followed by the PHR which consists of the frame length field and a reserved bit – together one octet. The length of the remaining field is specified by *aMaxPHYPacketSize* and is the payload at the same time. The payload is available for the next higher layer. At the PHY this frame has a length of 133 octets.

Medium Access Control Layer – MPDU The Medium Access Control Layer (MAC) is specified by the IEEE 802.15.4 standard, too. The maximum length of a MAC frame is 127 octets long. In the MAC Header (MHR) a two octet Frame Control field contains information about the frame type, the security status, if there is a pending frame, if an acknowledgment is requested, the desired PAN and the addressing mode (short or extended address). A few bits are reserved. The next octet contains the Data Sequence Number (DSN) which identifies the frame. The next fields are the addressing fields. They are between four and 20 octets in length. Under the given circumstances they are exactly eight octets long and consist of the PAN identifier fields and address fields for source and destination. Because short addressing mode is used just two octets are needed for one specific address. Furthermore in each case the PAN identifier needs also two octets. The auxiliary security field is empty because no security is enabled in this setup. The remaining part of the octets is reduced by the MAC Footer (MFR) and is available as MAC payload. The MFR contains the Frame Check Sequence (FCS) and is two octets long. It contains a 16-bit Cyclic Redundancy Check (CRC) (cd. Equation 4.4) which is calculated over the MHR and MAC payload field.

$$G_{16}(x) = x^{16} + x^{12} + x^5 + 1 \tag{4.4}$$

118 octets are remaining for the next higher level.

Network Layer – NPDU The Network Layer (NWK) is specified by the ZigBee standard. The general frame structure is given in Figure 4.6. It consists of a NWK header and the NWK payload. The maximum length is 118 octets. Eight octets are used for the header. Therefore 110 octets are available for the payload which is filled up with data from the next higher layer.

Octets: 2	**2**	**2**	**1**	**1**	**0/8**	**0/8**	**0/1**	**Variable**	**Variable**
Frame control	Destination address	Source address	Radius	Sequence number	Destination IEEE Address	Source IEEE Address	Multicast control	Source route subframe	Frame payload
NWK Header									Payload

Figure 4.6: ZigBee NWK general frame format [17]

The NWK header includes the frame control field, a destination and a source address field, a radius field and a sequence number field. The frame control field is two octets and includes the frame type (data or NWK command), the protocol version (*nwkcProtocolVersion*), a discover route (suppress or enable route discovery), a multicast flag, the security settings, a source route flag and two flags for the source and destination IEEE addresses. Three further bits are reserved. The destination and source address fields are in each case two octets and contain the 16-bit short network addresses. The necessary radius field is one octet long and is used as hop counter. It is decremented every time the packet passes a device (cf. Time To Live (TTL) in IP). The last NWK header field is called sequence number, is one octet in length and helps to relate a response to its request. With each new frame this number is incremented.

Application Support Sub-Layer – APDU Beneath the application objects the Application Support Sub-Layer (APS) provides services for transmitting application data. The frame has a maximum length of 110 octets and contains an APS header and an APS payload.

<table>
<tr><td>Octets:
1</td><td>0/1</td><td>0/2</td><td>0/2</td><td>0/2</td><td>0/1</td><td>1</td><td>0/
Variable</td><td>Variable</td></tr>
<tr><td rowspan="2">Frame control</td><td>Destination endpoint</td><td>Group address</td><td>Cluster identifier</td><td>Profile identifier</td><td>Source endpoint</td><td rowspan="2">APS counter</td><td rowspan="2">Extended header</td><td rowspan="2">Frame payload</td></tr>
<tr><td colspan="5">Addressing fields</td></tr>
<tr><td colspan="8">APS header</td><td>APS payload</td></tr>
</table>

Figure 4.7: ZigBee APS general frame format [17]

The APS header lengths depends also on using fragmented frames and is illustrated in Figure 4.7. However, most things are similar. The frame control field is one octet in length and specifies if this frame is a data frame, a command frame or an acknowledgment frame. Then it provides information about the delivery mode (normal unicast delivery, indirect delivery, broadcast or group addressing), the acknowledgment frame format it expects, the security settings, if an acknowledgment is requested and if the extended header fields should be included (which depends on using fragmentation). The destination endpoint field is one octet and specifies the endpoint number whereupon Table 4.5 applies.

The cluster identifier is two octets and informs about the identifier of the cluster inside the application for which the frame is intended. Similarly the profile identifier field (also two octets) holds information about the profile identifier the frame relates to. The source endpoint field specifies the endpoint the frame originates from. In the APS counter field an eight-bit number helps detecting the reception of duplicate frames. With every transmission this value is incremented by one.

As mentioned before the extended header fields are different for fragmented and single data frames. Whereas single data frames have no extended header at all, fragmented frames need a header extended by two octets. Fragmented frames consist of multiple blocks. The first octet is

Address	Recipient
0x00	ZigBee Device Object (ZDO)
0x01-0xF0	application object at this endpoint
0xF1-0xFE	reserved
0xFF	all active endpoints except the ZDO

Table 4.5: Destination endpoints

used as an extended frame control field which indicates by two bits the fragmentation information (cf. Table 4.6). The remaining five bits are reserved.

Fragmentation Value	Description
00	Transmission not fragmented.
01	Frame is first fragment of a fragmented transmission.
10	Frame is part of a fragmented transmission but not the first part.
11	Reserved

Table 4.6: Fragmentation sub-field [18]

The second octet is used as block number. If the value of the fragmentation sub-field is 01, the block number indicates the entire number of blocks. Otherwise it is the number of the transmitted block (with 0x01 for the second block). Hence, the maximum number of blocks is $2^8 = 256$. If fragmented frames are used, the payload field has a size of 100 octets (102 octets for a single frame because of the missing extended header frame) which results in $256 * 100$ octets $= 25,600$ octets ($\approx$ 25 KiB).

Long Interframe Spacing (LIFS)

The LIFS is 20 octets and gives the MAC some time to process received data by the PHY. If the data would be smaller than *aMaxSIFSFrameSize* (which is actually 18 octets) a shorter interframe spacing is possible.

Acknowledgment frame

The acknowledgment frame contains the frame control field, the delivery mode value, the acknowledgment format field (which is not set if it is an acknowledgment for a data frame), an optional security field, an acknowledgment request field (which is obviously not set) and the extended header present field (which is set). Since the acknowledgment request field is not set, the destination endpoint field, the cluster identifier field, the profile identifier field and the source endpoint field are set. The APS counter field is one octet and contains the same value as the incoming frame. Since it is a response to a fragmented frame, the extended header field is set.

The extended header field itself is three octets long. The first one is the extended frame control field and its value is the same as in the fragmented frame. The second one is the block number, which is zero for the first frame and contains the number of the received fragment otherwise. The third octet acknowledges the subsequent blocks. Since *apsMaxWindowSize* is between one and eight, every block is covered. A set bit indicates a successful transmission, if the bit is not set the transmission failed. If the transmission window is smaller than eight (in this setup it is three), the remaining bits are set to one.

To summarize the acknowledgment frame for fragmented frames is transmitted in ZigBee being encapsulated in the payload. The resulting APDU is eleven octets long. Adding the lengths of the headers from the layers below (NWK + MAC + PHY) it results in a length of 34 octets.

Calculated transmission time

Table 4.7 shows the transmission of one fragmented data frame and the corresponding acknowledgment frame. In the last row the transmission time is calculated (18,592 μs). A transmission of 25 KiB will take about 85 times so long. With this information an effective data rate of 126.06 kbps is calculated.

		symbols	octets	time [μs]	data [octet]
transmission window	CSMA/CA	68	34	1088	
	Frame (block)	266	133	4256	100
	LIFS	40	20	640	
	Frame (block)	266	133	4256	100
	LIFS	40	20	640	
	Frame (block)	266	133	4256	100
	LIFS	40	20	640	
ACK	CSMA/CA	68	34	1088	
	ACK	68	34	1088	
	LIFS	40	20	640	
		1162	581	18592	300

Table 4.7: Transmission window

CHAPTER 5

Wired integration case study KNX TP1

In this chapter *KNX* as a representative for wired networks is extended. This extension provides the opportunity to integrate CCTV systems into KNX. To show the feasibility a proof-of-concept based on KNX TP1 is also presented.

5.1 Profiles

KNX profiles regulate interworking between devices for system configuration and during normal operation.

In KNX applications are distributed over the network, so every participating device implements one or more FBs. FBs together are building a specific application (*distributed application*). Between these FBs, data are exchanged over the network. Each FB consists of one or more datapoints where four different types are distinguished:

- inputs,
- outputs,
- parameters or
- diagnostic data.

Inputs are values that are received by the FB. They can simply be retrieved from a sensor. Outputs are values which originate from the FB. They are data values that are processed by the application and act as value for actuators. Parameters are used to change the behavior of processing the inputs or outputs. Diagnostic data show information of an internal or local status. Parameters usually are set by management functions while diagnostic data are not to be used for runtime communication.

A FB with its datapoints is represented in a diagram typically. There are several datapoint types standardized in the KNX Standard.

5.2 Datapoints

In KNX, no dedicated datapoint types for the use within CCTV systems are available. Therefore, the KNX interworking model has to be extended for using CCTV systems. For connecting cameras to the KNX network the following new datapoint types are defined.

DPT_ImageIndication

If a smart camera (*on-the-spot*) detects an event a *DPT_ImageIndication* datapoint is used to inform all subscribers about the event. This datapoint type is specified in Figure 5.1. If an event is detected an indication is sent to the group address of the datapoint and all subscribers listening to it are notified. This datapoint includes six octets. The EventNumber subsequently identifies the detected event in a range between zero and 255. The next octet is called ImagePriority and defines the priority of the detected event. The higher the number the higher is the priority. The accurate assignment of priorities is not defined since it strongly depends on the surveillance application. A general idea is to treat safety problems with a higher priority than security issues. Octet three is the FrameSize. The FrameSize is the payload of an image frame and is in the range from zero to 249, if extended frames are used. As explained later the datapoint *DPT_ImageTransmission* has a header size of three octets. Therefore the payload size is either eleven octets (standard data frame) or 249 octets (extended data frame). Theoretically smaller payload fields are possible, especially if low-latency applications in the same network coexist.

The RemainderSize is one octet in length and is between zero and FrameSize - 1. The RemainderSize is necessary for reassembling the image. Because the picture size is needed before the data transmission has finished, the RemainderSize is used to calculate the picture size with Equation 5.1. This means that the RemainderSize is at most as large as FrameSize - 1 and therefore fits into a field with one octet length.

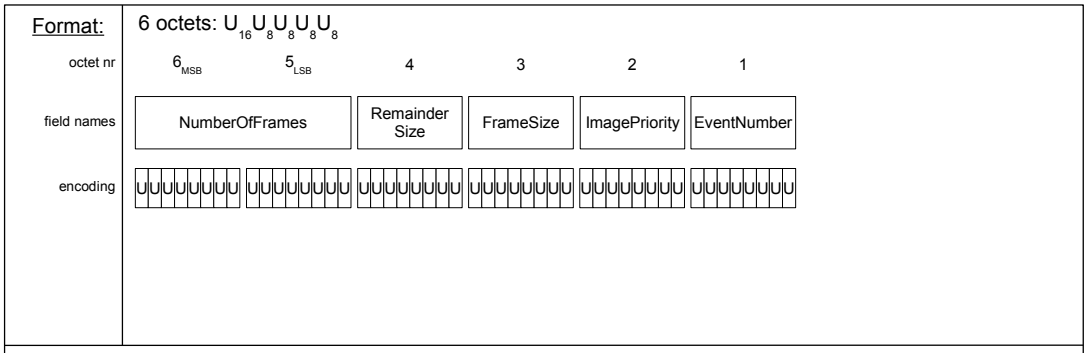
A different approach is the transmission of the actual size of the image as an absolute value. Using the RemainderSize a maximum value of one octet is used to specify this property. The maximum size of pictures using the KNX TP1 medium is $65,536 * 249 = 16,318,464$ octets ($\approx$ 15,936 KiB). Transmitting a number of this size is less efficient than the method introduced before, because more octets would be needed.

$$\text{ImageSize} = \text{FrameSize} * (\text{NumberOfFrames} - 1) + \text{RemainderSize} \tag{5.1}$$

The NumberOfFrames is necessary for calculation of the picture size. The values are between zero and 65,535 since two octets are available. This means 65,536 frames are possible for one picture which is enough for images in medium quality.

DPT_ImageTransmission

To actually transmit image data a datapoint called *DPT_ImageTransmission* is introduced. This datapoint transmits one single data frame of the image. The basic structure is parametrized



Datapoint Types

ID:	Name:	Use:
	DPT_ImageIndication	

Data fields	Description		Unit/Range
EventNumber	Subsequent event number	U_8	0...255
ImagePriority	Priority value of an event	U_8	0=lowest … 255=highest
FrameSize	Payload of an image frame	U_8	0...249
RemainderSize	Remainder octets of the image size	U_8	0...(FrameSize - 1)
NumberOfFrames	Number of frames of the according image	U_{16}	0...65535

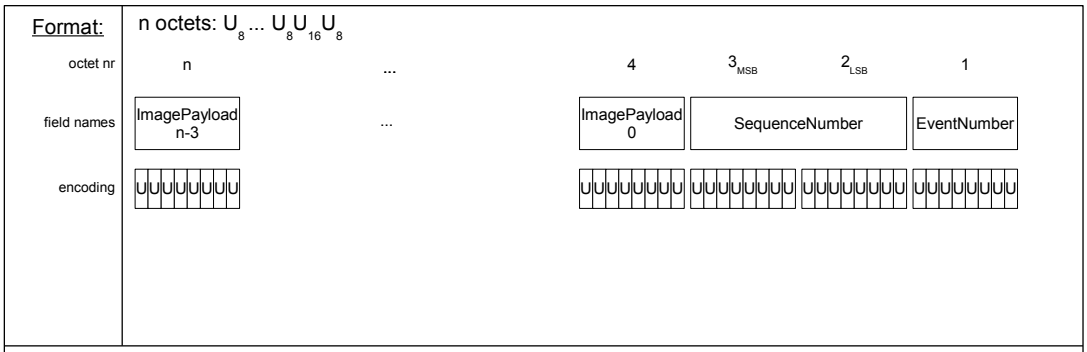
Figure 5.1: Datapoint type specification of *DPT_ImageIndication* in a simplified form

by the *DPT_ImageIndication* frame mentioned before. In detail the datapoint consists of three fields:

- EventNumber: To assign the image data to a specific event (especially if several events occur in short time) the field EventNumber includes the current number which was initially announced in the *DPT_ImageIndication* datapoint.
- SequenceNumber: The content of this field consecutively numbers the payload parts of the image. With this information it is possible to detect missing frames. It is used to reassemble the image data at the recipient. The SequenceNumber is between zero and NumberOfFrames - 1.
- ImagePayload: This field actually holds a part of the data of an image. Within a single extended data frame at most 249 octets are available for the payload data. The ongoing length is specified in the preceding *DPT_ImageIndictation* datapoint.

Figure 5.2 gives an overview of this datapoint type.

Since the image is transmitted using group addressing there is no need to transmit it more than one time, even if different devices have requested the image transmission.



Datapoint Types		
ID:	Name:	Use:
	DPT_ImageTransmission	

Data fields	Description	Encoding	Unit/Range
EventNumber	Subsequent event number	U_8	0...255
SequenceNumber	Identifier for a frame within an image	U_{16}	0...(NumberOfFrames - 1)
ImagePayload	Payload of an image frame	$(U_8)^{FrameSize}$	binary octets

Figure 5.2: Datapoint type specification of *DPT_ImageTransmission* in a simplified form

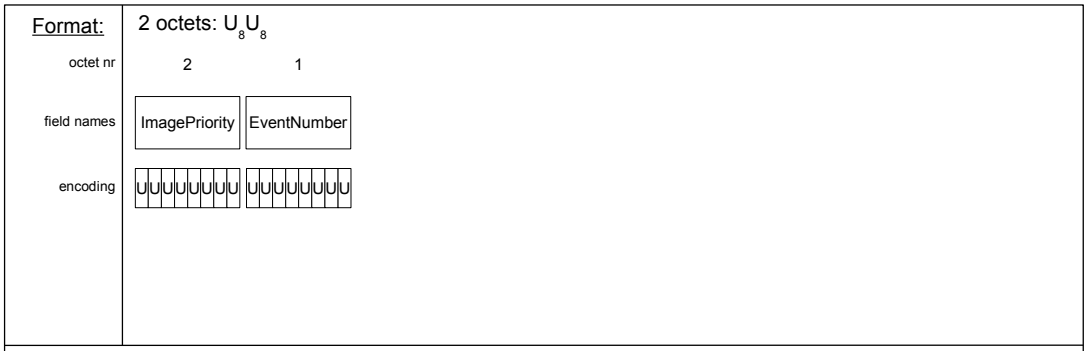
DPT_ImageRequest

After a *DPT_ImageIndication* is sent out each member of the group is able to request the transmission of the image. The decision whether an image is requested is within the responsibility of the device and depends on the system setup. It may be sufficient to disable image transmission in general and to request it if an operator needs it. In another system setup it may be priority dependent.

Anyway, if a picture shall be transmitted, a device has to request it using the *DPT_ImageRequest* datapoint. This datapoint contains the event number and the priority of the requested event and is outlined in Figure 5.3.

5.3 Functional Blocks

To permit image transmission inside a KNX network two FBs are introduced. On the one hand a Camera Functional Block (CFB) is necessary and on the other hand a so called Surveillance Functional Block (SFB) is needed. The on-the-spot image processing software detects events using the data of a camera. Accordingly, the client software interacts with the KNX network through the CFB. The CFB controls the KNX device, is able to inform SFBs about the event



Datapoint Types		
ID:	Name:	Use:
	DPT_ImageRequest	

Data fields	Description	Encoding	Unit/Range
EventNumber	Subsequent event number	U_8	0...255
ImagePriority	Priority value of an event	U_8	0=lowest … 255=highest

Figure 5.3: Datapoint type specification of *DPT_ImageRequest* in a simplified form

and transmits image data. Moreover it handles conflicts of different event requests from the SFBs and has control over its storage for a defined amount of saved event pictures.

Camera Functional Block

A CFB communicates using group objects. For the main tasks three different group objects are necessary. In addition more can be added. This way operations can be performed directly through the CFB. The interaction between the different FBs is shown in Figure 5.4.

In a chronological overview an event is detected by the mechanism of the on-the-spot camera. In a next step the CFB sends out a *DPT_ImageIndication* to a group address. Receiver of this group are all predefined SFBs. If at least one of these SFBs requests an image, the CFB transmits the image data using a *DPT_ImageTransmission*, which consists of several data frames in general. If a SFB requests an image may depend on the priority of the detected event. Using priorities it is possible to select an image in case different events occur in short time. Basically transmission of higher prioritized events must not be interrupted by transmissions of events with lower priorities.

If a *DPT_ImageRequest* datapoint is received and there is currently no image transmission in progress the desired image is transmitted. If in the meantime a different request is received the priorities of the event in transmission and the priority of the event which is required are compared. The result is one out of the following three possibilities: If the priority of the newer requested event is higher, the transmission under progress is aborted. Differentiation of the two images is feasible by the EventNumber field. If the priority of the newer requested event is

lower, it is neglected. No further communication happens automatically regarding this event. Nevertheless, the SFB is allowed to try it again after a specified amount of time. If the priorities are equal, the current transmission will continue. The newer request is queued and if the transmission of the current image has ended, the next picture in the queued event transmission list will be sent. This event transmission queue is arranged and acts as a First In First Out (FIFO) buffer. If more events occur than the transmission queue can temporarily save, the newer event will be saved, and the oldest one is deleted.

In case two similar requests are received the last request will be dropped. As mentioned before group communication is used, which allows the transmission of images only once independent of the amount of recipients, since every one inside the group can listen to the transmission. A SFB which wants to request an image and perceives that a different SFB requests the same picture has no need to send out a *DPT_ImageRequest*. The CFB does not pay attention to the amount of requests.

Surveillance Functional Block

The SFB is used by clients interested in the events and images. Examples are a security panel or an image archiving unit. The SFB has two main tasks. The first one is the decision on receiving the image, the second one is the reception of it. Through group communication several SFBs are listening to one (or more) CFB (cf. Figure 5.4). This has the big benefit that an image has to be transmitted only once. After a CFB sent out a *DPT_ImageIndication* all SFBs are allowed to request the image. A request of an image is initiated by using *DPT_ImageRequest* and group communication. As a result all SFBs retrieve the corresponding messages and so do not need to send a request, if another SFB has already sent one. Then the CFB starts transmitting (if there is no further transmission with same or higher priorities). As mentioned before, the SFB now has to collect all image data frames and passes them to the surveillance interface service (software running on the KNX device) which is able to reassemble the image and proceed further actions. A SFB always saves transmitted images and reassembles them even if no image was requested. If the transmission is complete and no image was requested by the user or software, the image is deleted.

If a transmission is continuing and a frame with a different event number is received, it is assumed, that the previous transmission is aborted because of a higher prioritized event. Normally, the communication messages prior to this are already received by the SFB. Therefore a request of a lower prioritized image is simply ignored by the CFB. This can happen if a transmission is in progress and a new SFB is turned on. Since this SFB has no information about the priority of the ongoing transmission (e.g., the KNX device with the SFB is turned on while a transmission takes place).

In Figure 5.4 the overall communication concept is shown. It presents an example application scenario and the interworking between a CFB, two SFBs and an optional Light Switch Actuator Block (LSAB). A camera is used to detect events (e.g., an intruder) which are treated by the CFB. On one hand it sends out a *DPT_ImageIndication* message using group addressing to all SFBs. On the other hand measures can be taken directly (e.g., turning on warning lights using the LSAB). If a SFB is interested in getting the image, it replies with a *DPT_ImageRequest*

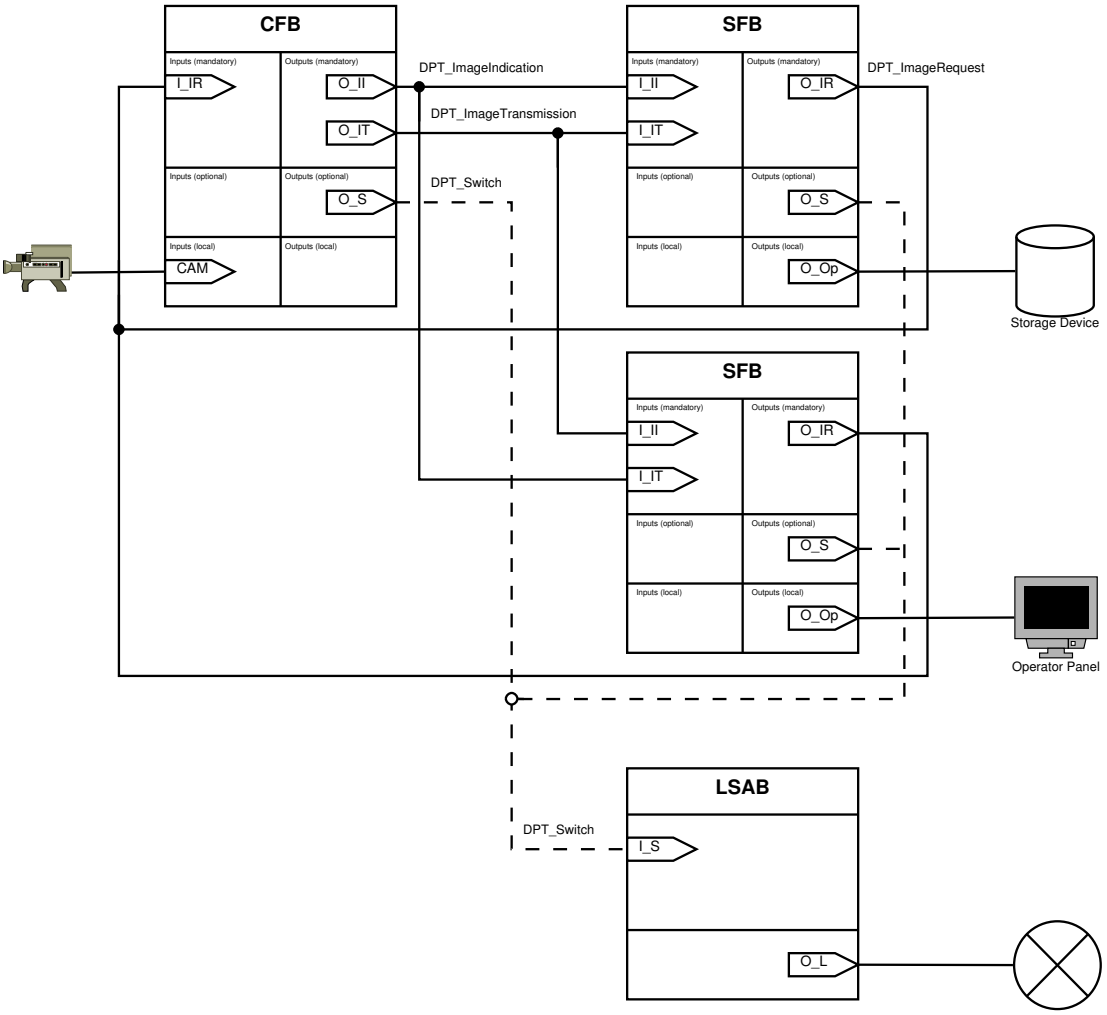


Figure 5.4: Communication between different FBs

group address. Then, the CFB starts transmitting image data using *DPT_ImageTransmission* messages.

All communication is done by using a group address. If individual addresses are used and two SFBs are interested in the image, it would have to be transmitted two times. This is not only time expensive and occupies the bus twice the time but also needs more energy. Furthermore this helps in holding back the messages which are not necessary. Consider two SFBs which are interested in two different prioritized events. If the *DPT_ImageIndication* for the higher prioritized image was sent, the SFB demanding the lower one can postpone its request because a properly working CFB would ignore it anyway. The same holds if an image is transmitted currently and a SFB decides to request an image with lower priority. If it knows that the priority is lower than the actual transmission, it can safely discard this request.

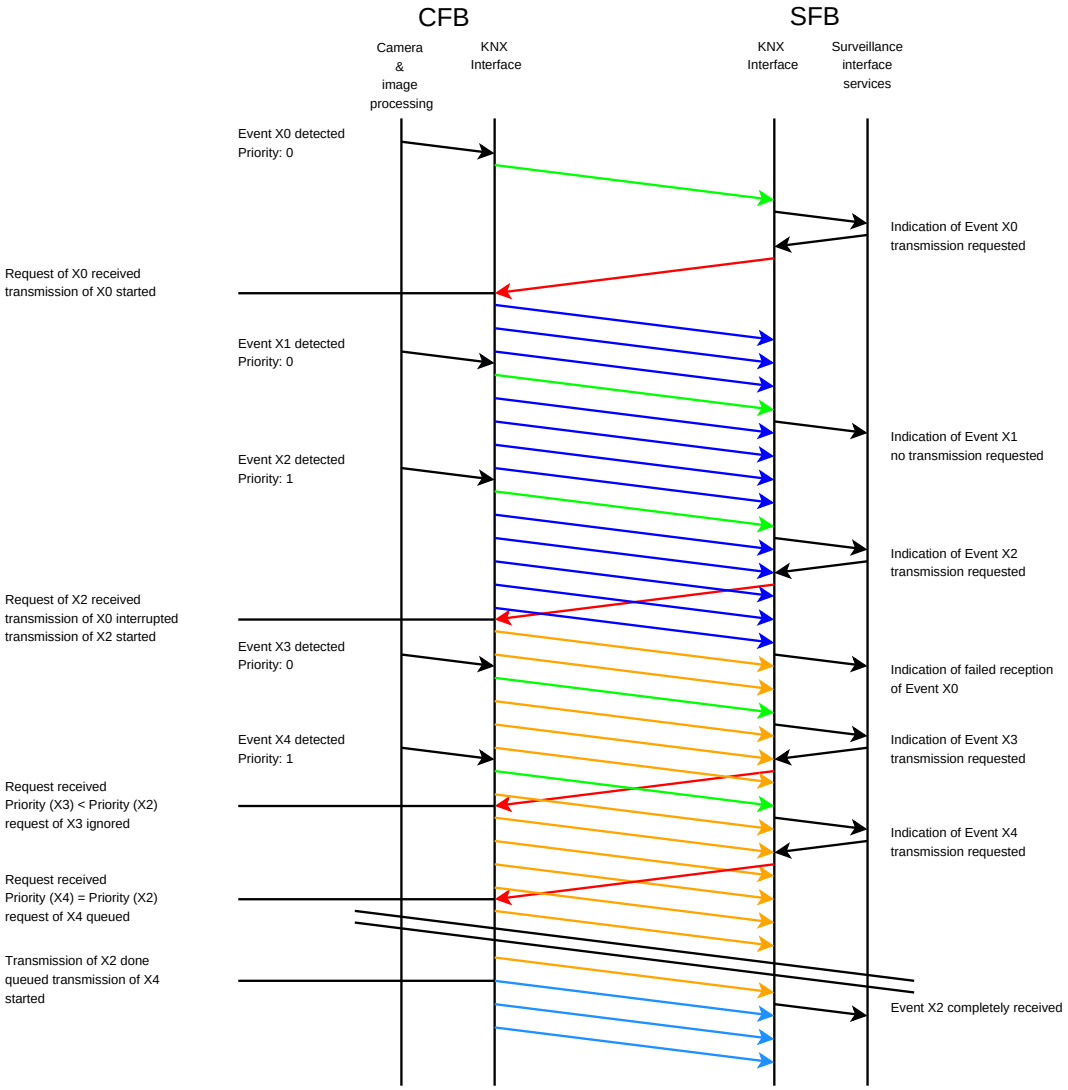


Figure 5.5: Interworking between CFB and SFB shown in a timeline diagram

In Figure 5.5 a time-line diagram of a typical application is shown. On the very left side the on-the-spot camera instance is represented by a vertical line. Next to it is the KNX Interface. Both are part of the CFB. On the very right side the (local) interface services are represented. Left to it there is the KNX interface. Both are part of the SFB. On the left and right side of this diagram the actions are described.

5.4 Proof-of-concept

The proposed approach was evaluated through a proof of concept. In Figure 5.6 the block diagram of a laboratory surveillance setting is shown. A standard web-camera was used and a TP-UART board [22] as connection to the KNX network. Additionally, a PC was used implementing the event detection and the CFB. To interface with the TP-UART board the KNX library daemon software eibd [23] was used. Event detection was reduced to motion detection by using the open-source library motion.

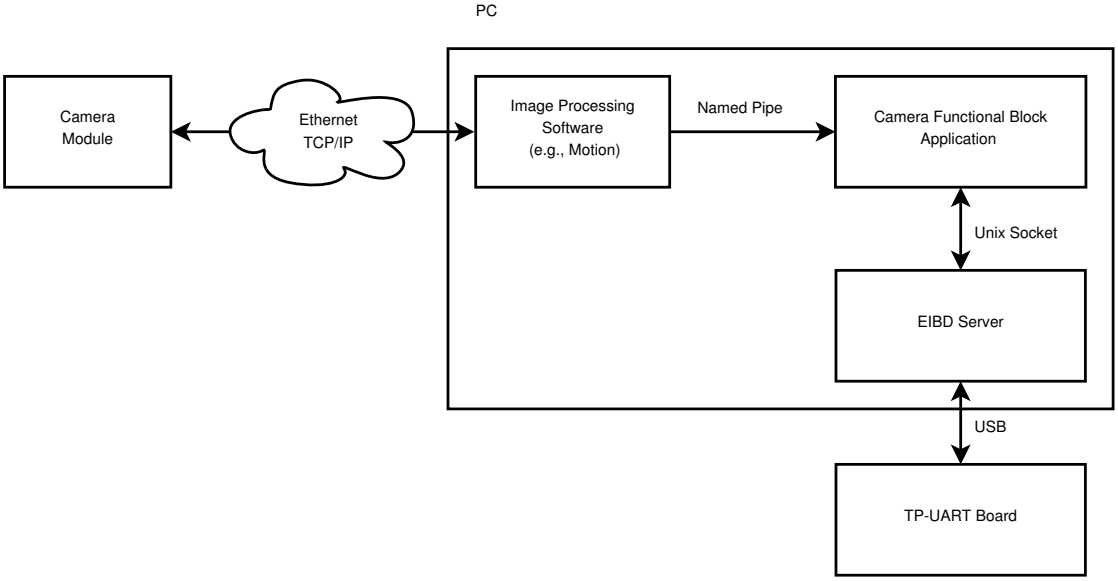


Figure 5.6: A block diagram of the laboratory setting

Motion is running as daemon and is configured using textual configuration files. Motion is powerful if all possibilities are exhausted. In this proof-of-concept an area-detection algorithm was used. Motion handles different connection types of (one or more) cameras. Connectivity is done through IP or over Unix sockets. Moreover interaction of multiple cameras was defined. Whenever an event was detected configured actions were taken. Additionally, the observed area was parted into several parts reflecting different priorities. Other options of Motion allow to choose not the first picture of an event but the best one. Consider an entrance where the first picture shows just the door opening. However, the best picture is one with the person identified. Also the further processing is configurable: Besides uploading to ftp, saving somewhere else or sending an email having the picture attached or running a command is possible. The latter was chosen to push a picture to the named pipe of the CFB.

The used axis camera included a rudimentary on-board motion detection which is adequate for simple needs. An experimental try pointed out the limitations of this system. The event detection works with just one camera but it can handle different events at the same time. Therefore events have to be defined in advance. They are identified by a name and an event description.

Events can be easily enabled or disabled. These events are associated with one out of three priorities which are used for processing through the Central Processing Unit (CPU) only. An Axis event is the description of a more general event than Motion does. Events can be triggered by starting-up the camera or by input ports, too. Less powerful are the possibilities of notifications of an event. To sum it up independent event detection libraries are more capable than on-board (and proprietary) tools. Therefore, Motion was used instead of the on-board event detection of Axis.

If an event was detected through the event detection library the CFB starts to work. The image is piped to the eibd process. eibd sends the *DPT_ImageIndication* message using the TP-UART. The following operation is illustrated in Figure 5.7 and is implemented as a state machine. This state machine regularly polls the pipes for new messages. There are different pipes for the different priorities. After start-up the state machine is in state *idle*. It remains in this state even after sending an indication of a new detected event. If a SFB requests an image by sending a *DPT_ImageRequest* message the state is changed to *transmit*. There are multiple *transmit* states. Each identified by the event number and the sequence number. After having finished the transmission of an image the state machine changes back to *idle* state if there is no further image in the queue which needs to be processed.

The SFB was created for an operator panel. The implemented SFB is indicating all events detected by the CFB with its event number to the user console. If an image should be requested, the operator has to enter the event number. If the prerequisites (priority, image still available) are met, the image is transmitted through the CFB and is displayed with the standard image image viewer of the operating system after it was reassembled.

Like the CFB before the SFB was realized as state machine, too. Figure 5.8 displays the possible states. Initially the SFB is in *idle* state. Every incoming event indication is displayed on the user console. If the user requests an image by entering the event number, the number and the priority are checked. If such an event exists (e.g., an indication was received before) and the priority is higher than the current priority (of images requested by other SFBs or transmission messages from the CFB), an image request is sent and the state is changed to *idle_rq*. If the CFB transmits an image, the state is changed to one of the *transmit* states. Here the image frames are buffered and saved to file, if the complete image was received. A new image is detected by a different event number and a sequence number equal to zero. A transmission is aborted if an image with a higher priority is requested (by some SFB). Note that the internal current priority is updated to compare it with image requests by the local user or application.

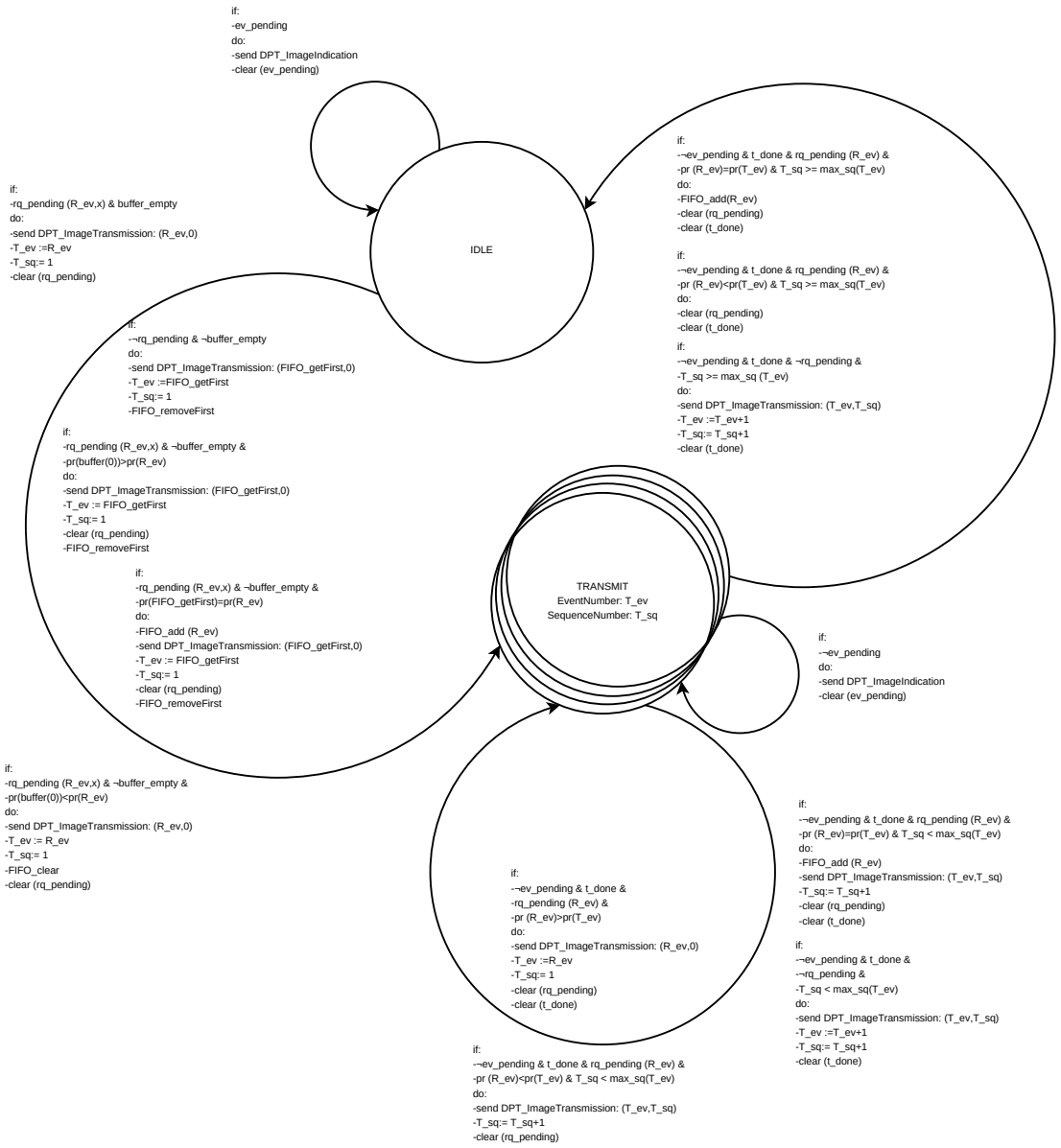


Figure 5.7: State machine of the CFB

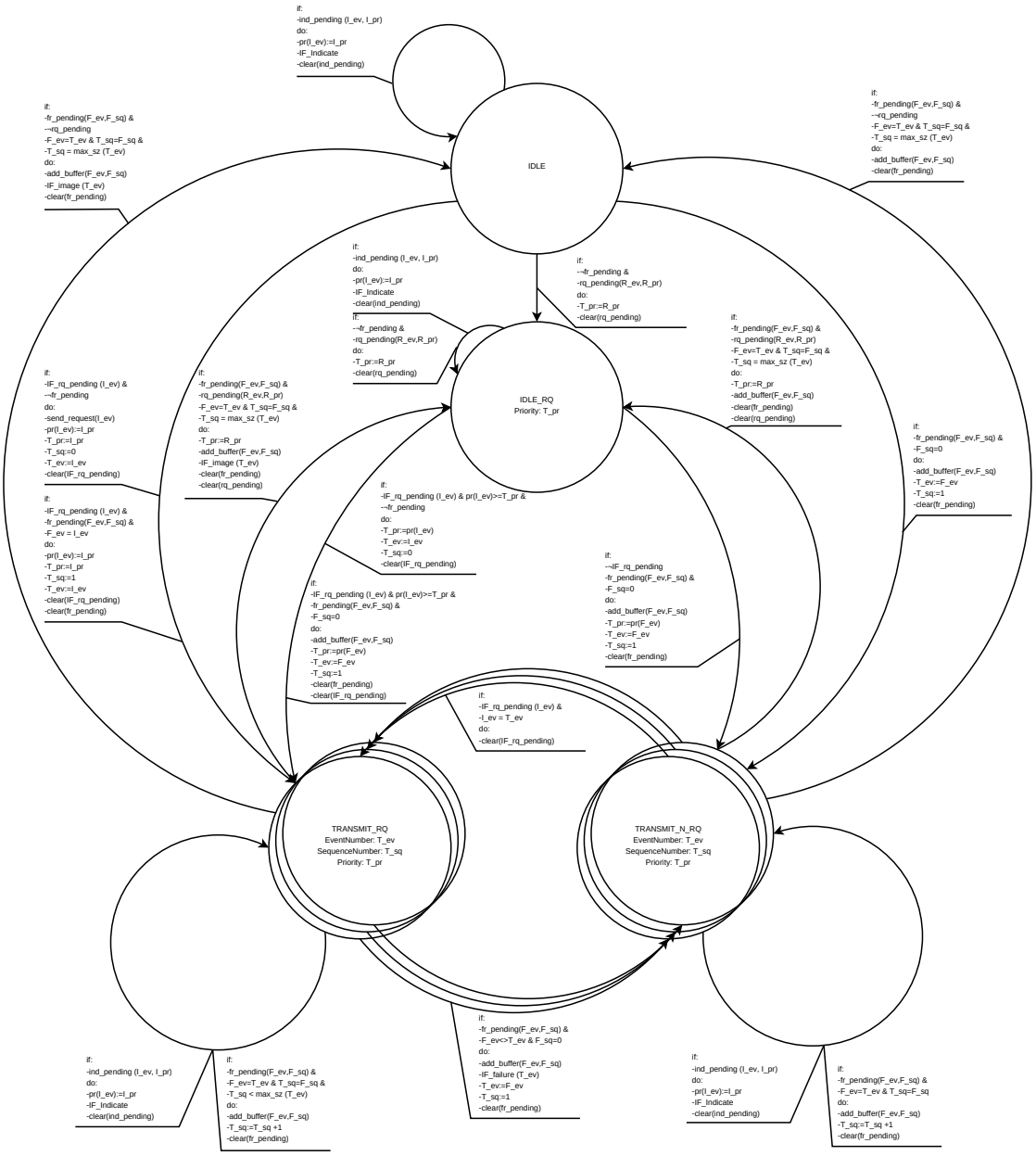


Figure 5.8: State machine of the SFB

CHAPTER 6

Wireless integration case study ZigBee 2.4 GHz

In this chapter a closer look at the Application Layer (APL) of ZigBee is given. Inside this layer the Application Support Sub-Layer, the application framework and the ZigBee Device Object resist. Improvements on this layer are necessary to prepare ZigBee for transmitting CCTV data. The general ZigBee stack structure is given in Figure 6.1. Note that the lower layers (PHY and MAC) are specified by the IEEE 802.15.4 standard.

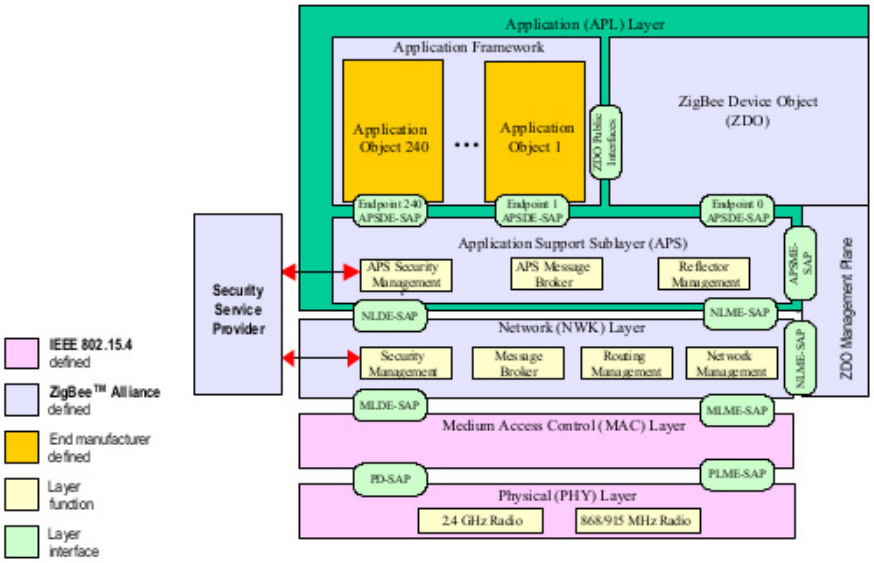


Figure 6.1: ZigBee stack architecture [17]

6.1 Application framework

If ZigBee devices want to exchange data they need to agree on a profile. A profile defines the way how data is stored within the ZigBee devices (e.g., encoding, ranges, semantic) and how data is accessed and exchanged. An overview of endpoints used by ZigBee devices is given in Table 4.5 in Chapter 4.

A profile consists of rules of general settings for the devices of a dedicated application domain. A profile defines different device types.

A device is able to support at least one profile. To find out details about a device descriptors are available. The following descriptors exist:

- Node Descriptor
- Node Power Descriptor
- Simple Descriptor
- Complex Descriptor
- User Descriptor

The node descriptor is mandatory and includes the type and capabilities of the node. Details about the node power are contained in the node power descriptor which is mandatory, too. A mandatory simple descriptor is available for each application object. Finally, an optional complex descriptor is used for further description of the device. The user descriptor contains a character string describing the device. It is optional, too.

For each device type, so called clusters are specified. Clusters can be mandatory or optional. In Figure 6.2 the basic architecture of a device with its clusters is illustrated. Every device supports common clusters and device specific clusters. Each cluster again consists of commands and attribute sets which finally hold the attributes that represent the process data.

Additionally, devices can implement clusters, server or client side. In the ZigBee Cluster Library (ZCL) [24] the server is defined as the device receiving read or write attribute requests. The report attribute command is typically sent out from the server. As a pre-requisite for communication between two devices, they have to support the same profile and are bound together. Binding functions are provided by the ZigBee Device Profile (ZDP).

ZigBee Device Object

The ZigBee Device Profile supports the following functions:

- Device and Service Discovery
- End Device Binding
- Bind and Unbinding
- Binding Table Management

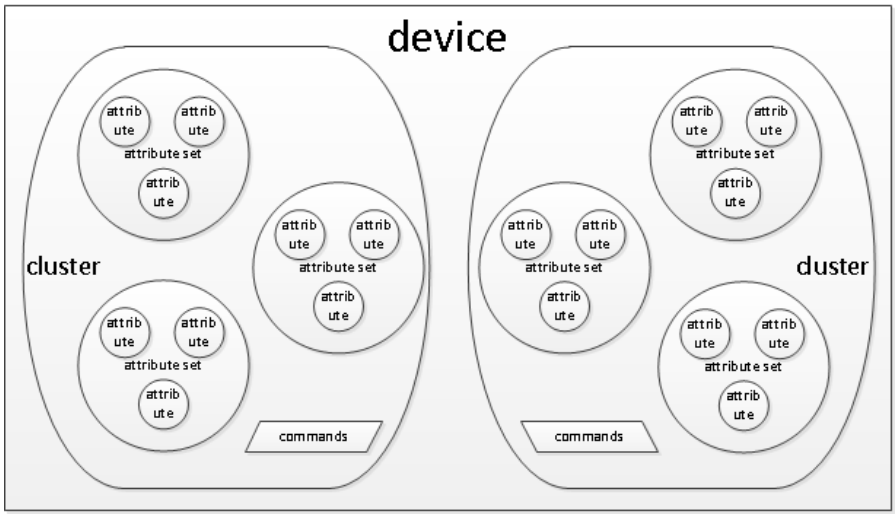


Figure 6.2: ZigBee device architecture

- Network Management

Device discovery is used to find other devices inside the PAN and obtain their identifier (64-bit or 16-bit network address). Having the network address of another device, the service discovery procedure allows to request information about services running on that device. In detail, there are different request types (e.g., to list active endpoints or to match the simple descriptor). End device binding is used whenever a user intervention is necessary to bind a device (e.g., at installation time buttons of two devices shall be pressed). Common binding of control messages to their destination is done with the bind and unbind functions. Further functions allow to modify or list the binding table which contains required binding information.

Home automation profile

The home automation profile is an example of a ZigBee profile that specifies device types common to the home automation domain. Its identifier is 0x0104. It contains device descriptors and standard practices for up to 500 different home automation devices [25]. HVAC devices, devices for lighting and shading and security devices are included in this profile. As mentioned before, if a device supports the home automation profile it is guaranteed that a vendor independence is given. This profile supports the following device groups:

Generic devices This group defines device types that implement simple on/off-switch clusters, mains power outlet clusters and the scene selector cluster.

Lighting devices In this group devices for lights and switches are described including ordinary lights/switches as well as color dimmable lights or color dimmer switches.

Closure devices Closure devices are shades and window covering devices.

HVAC devices Typical HVAC devices are specified here (e.g., heating/cooling units, thermostats, pumps, flow sensors).

IAS devices This group contains devices for control of IAS.

6.2 Improving the home automation profile

To be able to integrate CCTV devices the existing device specifications with the home automation profile need to be analyzed regarding their suitability. The interaction between the IAS devices is as follows:

The central point of an IAS is the *Control and Indicating Equipment (CIE)* device. It manages the connected devices. If an alarm is detected it is responsible for activating the Warning Devices (WDs). A CIE device supports the *IAS Ancillary Control Equipment (ACE)* cluster on its server side. On the client side *IAS WD*, *Identify* and *IAS Zone* cluster are mandatory. Optional the *Scenes* cluster and the *Groups* cluster can be implemented.

To remotely control the CIE the *IAS ACE* device is available. It can manipulate settings in the CIE. Additionally, it can act as a Zone device (e.g., personal alarm). It supports *IAS Zone* cluster on its server side and *IAS ACE* cluster and *Identify* cluster on its client side.

IAS Zone devices are generic sensors in the network. They can detect various alarm conditions. It is within the responsibility of the device which type of alarm is detected. Basically they support two different alarm types and they report if their batteries are low. An IAS zone device just needs an *IAS Zone* cluster on its server side.

A *IAS WD* is a generic output device for an IAS. It can be, for example, a klaxon or a warning light. The warning output is activated if the CIE initiates it. It supports the *IAS WD* cluster and *IAS Zone* cluster on its server side. Optional are the *Scene* cluster and the *Groups* cluster.

Having a closer look at the IAS devices, the *IAS Zone* cluster seems to support most of the required features.

Additional to the mentioned clusters in the last paragraphs every device in the home automation profile has common clusters such as *Basic* and *Identify* on the server side. Optional, there are more clusters possible.

To extend standard profiles three different opportunities exist: [24]

- Add manufacturer specific clusters to a standard profile.
- Add manufacturer specific commands to a standard cluster.
- Add manufacturer specific attributes to a standard cluster.

If the home automation profile is used for the given problem, then just an attribute for the transmission of the image data is missing. Therefore the third option is chosen to extend the home automation profile. The *IAS_Zone* cluster needs an additional attribute. In Table 6.1 the

attribute with the identifier 0x0003 is added. It is an array consisting of as many 64-bit data blocks as needed to represent the image data. The *ZoneType* field already includes information about the sensor type (e.g., motion sensor, contact switch) and describes the meaning of the two alarm bits included in the *ZoneStatus* attribute.

Identifier	Name	Type	Range	Access	Default	M/O[1]
0x0000	ZoneState	8-bit Enumeration	All	Read only	0x00	M
0x0001	ZoneType	16-bit Enumeration	All	Read only	-	M
0x0002	ZoneStatus	16-bit bitmap	All	Read only	0x00	M
0x0003	ZoneData	Array of 64-bit Data	-	Read only	-	O

Table 6.1: Improved attributes of the Zone Information Attribute Set

Figure 6.3 shows how to define clusters using the development environment that has been chosen for the proof-of-concept. The *IAS Zone* cluster is highlighted.

Next the descriptors of the ZED are defined. In Table 6.2 the simple descriptor of the on-the-spot camera ZED is shown. Finally, a complex descriptor has to be created (Table 6.3). The user descriptor is 16 characters long and was defined to be “onthespot camera” for such ZEDs.

Field name	Length [bits]	Value
Endpoint	8	1-240
Application profile identifier	16	0x0104
Application device identifier	16	
Application device version	4	0000
Reserved	4	0000
Application input cluster count	8	0
Application output cluster count	8	3
Application output cluster list	$16 * o$	0x0000 (Basic), 0x0003 (Identify), 0x0500 (IAS Zone)

Table 6.2: Fields of the Simple Descriptor of an on-the-spot camera ZED

If an on-the-spot camera ZED detects an event, an image transmission may be initiated. Whether an image shall be transmitted can be decided by the image analyzing algorithm or by the system engineering during configuration.[2]. If an image is transmitted the ZED indicates that using a *Zone Status Change Notification Command*. Next, the ZC will request the image (cf. Figure 6.4) and a fragmented transmission starts.

[1] Mandatory/Optional

[2] In later applications configuration at runtime can be implemented using specific clusters, which may need to be defined.

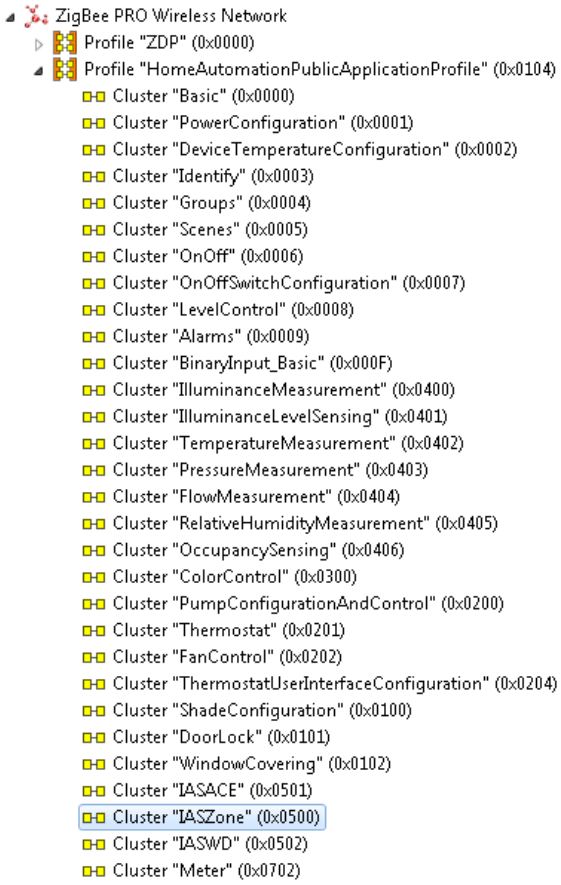


Figure 6.3: Define clusters using eclipse

6.3 Proof-of-concept

On the market multiple 802.15.4 chips are available which are suitable for implementing the ZigBee stack. For this proof-of-concept the system-on-chip solution Jennic JN5148 was chosen. These devices are low-cost and ultra low power devices. They consist of a 32-bit Reduced Instruction Set Computing (RISC) processor. To develop the application, an Eclipse based development environment is available.

Hardware

Jennic's development kit consists of a controller board and four sensor boards. Five ZigBee modules are delivered with the kit that can be connected to the boards. The difference of the modules is the transmission power and the antenna. The boards are equipped with some simple peripherals:

- *UART*: Two UART interfaces are for programming and debugging purposes. They can

Field name	XML Tag	Compressed XML Tag Value x_1x_0	Value
Reserved	-	00	0
Language and character set	<languageChar>	01	EN 0x00
Manufacturer Name	<manufacturerName>	02	Jennic
Model name	<modelName>	03	JN5148
Serial number	<serialNumber>	04	09476XXXXX
Device URL	<deviceURL>	05	www.jennic.com
Icon	<icon>	06	
Icon URL	<outliner>	07	
Reserved	-	08-FF	0

Table 6.3: Fields of the Complex Descriptor of an on-the-spot camera ZED

also be used to connect external peripherals.

- *Temperature and Humidity sensor*: The boards are equipped with a Sensirion SHT11 multi-sensor module that provides the current temperature and humidity. Both micros-sensors are coupled to a 14-bit analog-to-digital converter. The ranges are 0 – 100% humidity and a temperature between -40 °C – 85 °C.

- *Light sensor*: Containing a TAOS TSL2550 the board provides a digital output light sensor and a two-wire SMBus serial interface. For measuring the light conditions two photodiodes are included. The value is processed by a single CMOS IC over a dynamic range of 12 bits. The response should be similar to that of a human eye.

- *Switches*: Two push buttons (four on the controller board) are available for human interaction. Two additional buttons are available for resetting and programming the boards.

- *LEDs*: Two LEDs are placed on the board. Using them, feedback to the user can be given. Especially during the joining procedure to a ZigBee network they can be used to inform the user about the joining process.

- *LCD panel*: The controller board is equipped with a 128 x 64 px LCD screen. The readability can be improved through using its backlight. Due to high power consumption, it is just available in external power supply mode.

As mentioned before, power supply can be provided by on-board power with two AAA batteries or an external power supply. The boards can easily be used with external power since they can be powered with alternating or direct current. Additionally, they are protected against reverse polarity. Nevertheless, a fuse is provided. Every board has a 24AA01 chip which is the 128-byte x 8-bit EEPROM.

In Figure 6.5 the design of the proof-of-concept is shown for the transmitter's side. A Jennic ZigBee board is connected to the SBC-i.MX51 board from Bluetechnix [26] which again

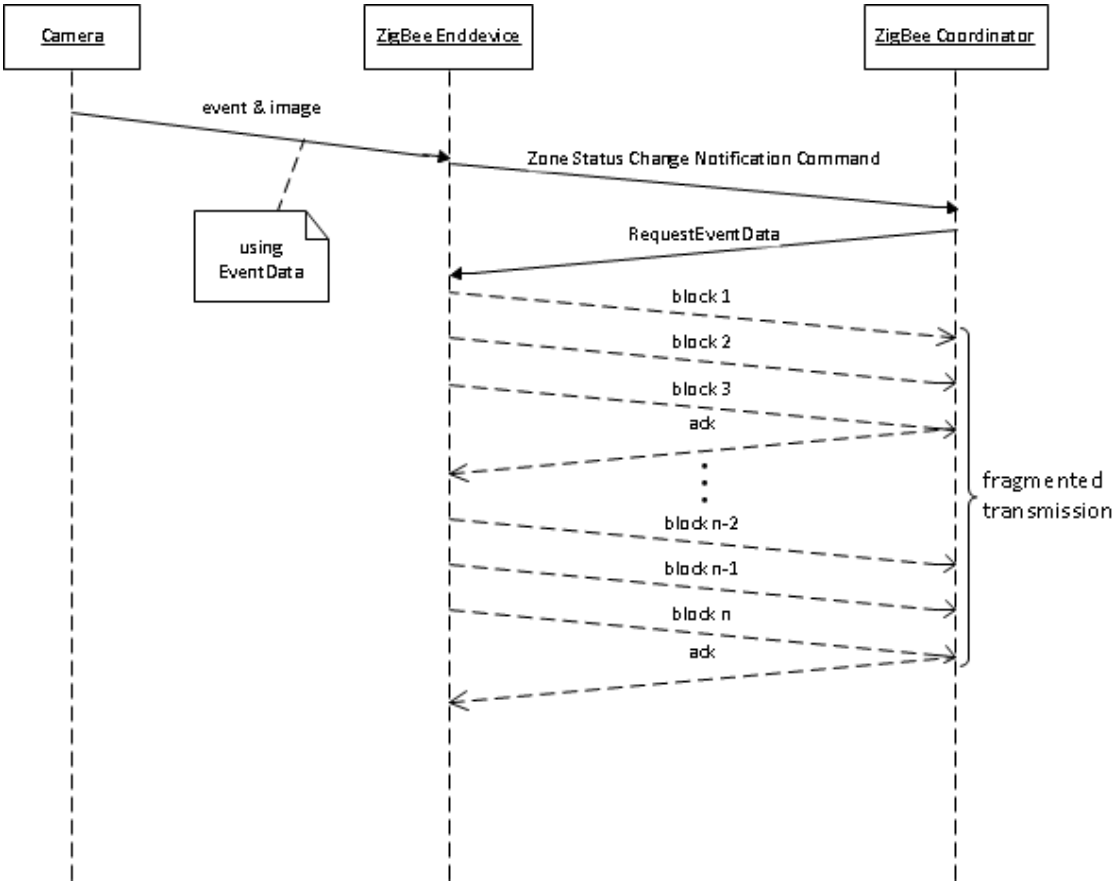


Figure 6.4: ZigBee transmission example

connects a USB camera. Running OpenCV [27] as pattern recognition software the event data is transmitted to the ZigBee board using a serial connection. Since the JN5148 supports Universal Asynchronous Receiver Transmitter (UART) transmission rates of up to 1 Mbps and the Bluetechnix board is capable of an ARM Cortex-A8 CPU, data can be transmitted in sufficient time.

On the receiver's side the transmitted data will be received by a ZC and will be handed over to a personal computer using a serial data connection.

Software

Jennic ZigBee devices come with the ZigBee PRO software on board. It supports the ZigBee PRO 2007 stack (cf. Figure 6.1) and includes the Jennic operating system JenOS. Through the use of Application Programming Interfaces (APIs) access to ZigBee PRO and JenOS functions is given.

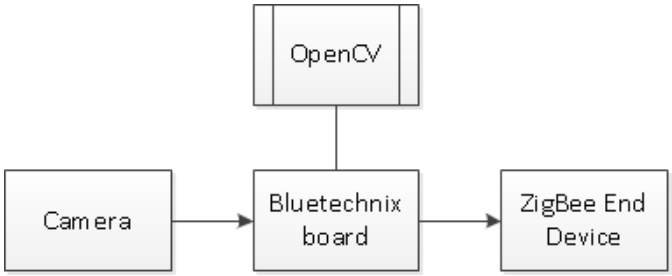


Figure 6.5: Block diagram illustrating the proof-of-concept solution

ZigBee PRO APIs

In more detail the ZigBee PRO APIs are listed and explained here.

- ZigBee Device Object (ZDO) API: This API provides functions for building a network, functions for setting the network parameters and functions for joining or leaving ZigBee networks.
- ZigBee Device Profile (ZDP) API: Using the ZDP functions interaction with remote devices is possible. Examples are the device and service discovery or binding.
- Application Framework API: This API is used for creating and modifying device descriptors or to work with data frames.

JenOS APIs

The JenOS API provides access for non-network-specific operations. Since the JenOS is divided into different modules, they are explained separately. JennOS interacts with the user application, the ZigBee PRO stack and the peripherals. For setting and modifying general options Jennic provides a configuration tool for Eclipse.

- Real-time Operating System (RTOS): The RTOS is responsible for assigning CPU time to tasks. To achieve this priorities are used. Additionally, the RTOS has to react to controlled interrupts. Since the Jennic RTOS is a pre-emptive operating system, it allows to hold tasks in favor of a task with higher priority. Though it supports mutexes for execution of critical sections. If there is no task to be processed, an idle task is executed. Developing user applications is done by implementing user tasks. They consist of a main function and sub-functions. The main function is executed by the operating system only. In Figure 6.6 a screenshot of the Eclipse plug-in for the system design is shown.
- Persistent Data Manager (PDM): The storage of data in non-volatile memory is handled with the PDM. It provides operations for initialization, for saving, recovering and deleting data and for using mutexes.

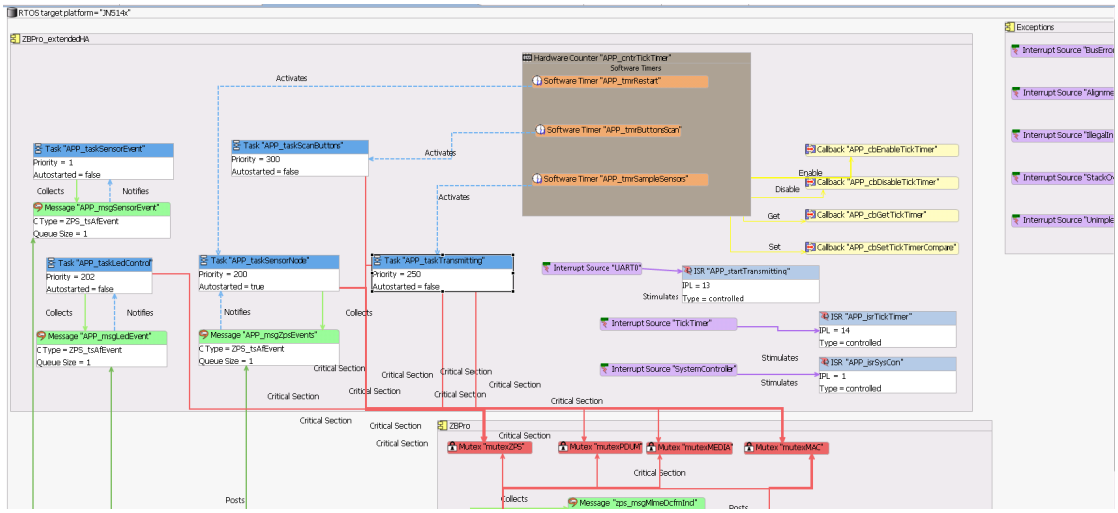


Figure 6.6: A JenOS eclipse plug-in

- Power Manager (PWRM): ZigBee is known for its low-energy consumption. To enable this, devices enter a sleep mode. They regularly awake for very short times. JenOS knows different sleep modes resulting in varying power consumption and re-awakening times. In the CCTV example no use of sleep modes was made.
- Protocol Data Unit Manager (PDUM): Data in APDUs are handled by the PDUM. Whenever a message between devices is sent the PDUM is involved. It is responsible for the APDU management (inserting data in and extracting data from APDUs). So, it appends headers and trailers and uses fragmentation for large messages if needed.
- Debug Module (DBG): Debug statements can be inserted which will be enabled only, if the DBG is activated. The output is provided on the serial interface. Additionally, logical test condition can be executed this way.

State machine

The state machine of the software running on the ZED is given in Figure 6.7. The state *off* represents the deactivated ZED. After power-on the initialization phase is started which automatically leads to the next state *Scanning*. When a ZC is found the state is changed to *Associating* where association and binding are done. If it succeeds the *Idle* state becomes active. In this state system tasks are executed while the RTOS waits for an interrupt indicating a new event. In that case, the state is changed to *Transmitting* and the event data is sent to the ZC.

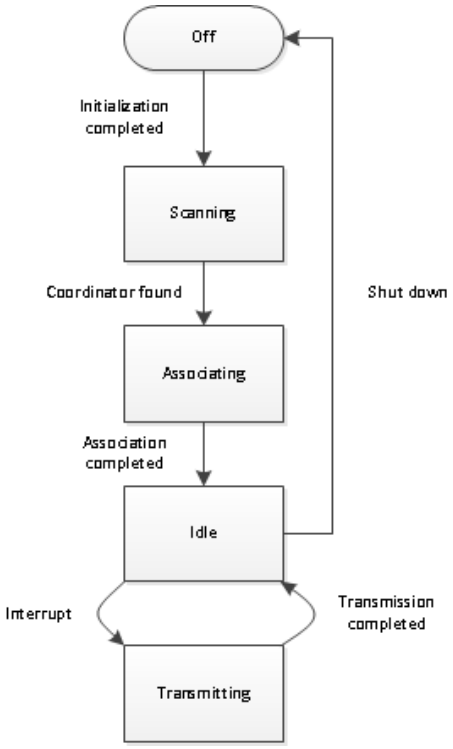


Figure 6.7: State machine of a ZED

CHAPTER 7

Conclusion

The project *miniSPOT.net* focuses on the use of simple camera modules as security cameras and the transmission of video data over home and building automation networks. Therefore several project partners are working together: While the *Computer Vision Lab (CVL)* (Vienna University of Technology) and *Cogvis* are improving the ability of the camera modules, the *Institute for Advanced Studies* takes care about the social effects and consequences of the observation. Furthermore *Helwacht* is the industrial partner with practical experiences. Last but not least the *Automation Systems Groups (ASG)* (Vienna University of Technology) investigates the communication aspects within this project. The work presented in this thesis is part of the latter activities.

CVL aims at improving motion detection algorithms suitable for embedded environments. The algorithms need to work with cameras like typically used in modern cell phones taking into account their limited resources. In future work further limitations of the employed hardware have to be considered. Calculations need to get faster and leaner to be executed under restricted resources to meet all requirements.

This thesis put the focus on transmitting video data over control networks that typically provide only low bandwidth. It aimed at integrating smart on-the-spot cameras in existing communication technologies. After a brief overview of existing BAS technologies and the possibilities of transmitting video data via two representatives. The aim was the integration of smart on-the-spot cameras in existing systems.

After a brief description of scenarios from different application domains, the sensor fusion and sensor sharing possibilities were explained. A short overview of current BAS technologies including the supported media was the basis for the calculation of transmission times. For precise results the KNX and the ZigBee protocol were analyzed in detail. This led to the calculation of the effective bit rate. Both protocols have a high protocol overhead in common. Therefore the most suitable frame types that fulfill the transmission requirements were chosen. As a result it was pointed out that video streaming is generally not feasible but the transmission of snapshots is practically possible. Thus, to put the underlying concept into practice, significant support

from image and picture analyzing algorithms is necessary. Following this idea, the basics of a transmission over two building automation networks was shown in a proof-of-concept.

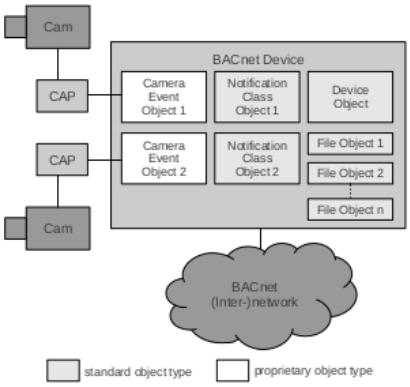


Figure 7.1: BACnet camera object model [20]

In the meantime [20] described how to implement smart camera systems into BACnet based networks. Therefore a BACnet object model including its properties for CCTV systems was developed which is shown in Figure 7.1. In a proof-of-concept an open source BACnet protocol stack and the BACnet/IP network option was used.

In a next step the possibilities of the data transmission in ZigBee can be extended. Configuration of parameters at runtime can be implemented. Therefore the ZigBee clusters need to be extended by appending (new) parameters to the attribute sets and a transmission of messages from the ZC to the ZED has to be installed. This bidirectional messaging increases the collision avoidance effort, especially if time considerations are an issue. If a standardization is desired, contact to the ZigBee alliance has to be made to get an approval of the extended home automation profile. Another possibility is to create a new manufacturer specific extension of this profile. On success, an official profile identification number is assigned.

In KNX TP1 transmission is possible but not useful since the transmission rate is too low. Future work in this direction is the change to another medium. Here IP based communication with KNXnet/IP and native KNX IP devices will come into play.

Bibliography

[1] A. Wege, *Video-Überwachungstechnik*. Hüthig, 2 ed., 1997.

[2] M. Gilge, "Networked Video for CCTV Applications. The Network is the Multiplexer.," *Euro Security International*, no. 6, pp. 8–14, 2001.

[3] M. G. Döring, *Digitale CCTV-Systeme*. Economica, 2004.

[4] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 34, no. 3, pp. 334–352, 2004.

[5] B. Rinner and W. Wolf, "An introduction to distributed smart cameras," *Proceedings of the IEEE*, vol. 96, no. 10, pp. 1565 –1575, 2008.

[6] M. Valera and S. Velastin, "Intelligent distributed surveillance systems: a review," *Vision, Image and Signal Processing, IEE Proceedings*, vol. 152, no. 2, pp. 192–204, 2005.

[7] M. Bramberger, A. Doblander, A. Maier, B. Rinner, and H. Schwabach, "Distributed embedded smart cameras for surveillance applications," *Computer*, vol. 39, no. 2, pp. 68–75, 2006.

[8] W. Elmenreich, *Sensor Fusion in Time-Triggered Systems*. PhD thesis, Vienna University of Technology, Oct. 2002.

[9] J.-C. Naranjo, C. Fernandez, P. Sala, and M. Hellenschmidt, "A Modelling Framework for Ambient Assisted Living Validation," in *Universal Access in HCI*, pp. 228–237, 2009.

[10] T. Kleinberger, M. Becker, E. Ras, A. Holzinger, and P. Müller, "Ambient Intelligence in Assisted Living: Enable Elderly People to Handle Future Interfaces," in *Universal Access in HCI*, pp. 103–112, 2007.

[11] "Gefahrenmeldeanlagen für Brand, Einbruch und Überfall." VDE 0833, 1989.

[12] "Fire detection and fire alarm systems." Oesterreichisches Normungsinstitut, EN54-13, 2004.

[13] W. Granzer, *Secure Communication in Home and Building Automation Systems*. PhD thesis, Vienna University of Technology, Feb. 2010.

[14] "KNX Specification Version 2.0." Konnex Association, 2009.

[15] "BACnet – A Data Communication Protocol for Building Automation and Control Networks." ANSI/ASHRAE 135-2010, 2010.

[16] J. Maier, "Powerline in Building Automation." Bachelor Thesis, Automation Systems Group, TU Vienna.

[17] "ZigBee Specification." ZigBee Alliance, 2007.

[18] "IEEE 802.15.4 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WANs)." IEEE Computer Society, 2003.

[19] A. S. Tanenbaum, *Distributed Systems: Principles and Paradigms*. Prentice Hall, 2 ed., 2007.

[20] C. Mauser, W. Granzer, and W. Kastner, "Integrating CCTV Systems into BACnet," in *Proc. of 16th IEEE Conference on Emerging Technologies and Factory Automation (ETFA '11)*, Sept. 2011.

[21] "Calculating 802.15.4 Data Rates." Jennic, 2006.

[22] G. Neugschwandtner and A. Fernbach, "Design of an enhanced TP-UART based KNX PC interface," in *Proc. KNX Scientific Conference 2008*, Nov. 2008.

[23] M. Kögler, "Free Development Environment for Bus Coupling Units of the European Installation Bus," Master's thesis, Vienna University of Technology, 2005.

[24] "ZigBee Cluster Library Specification." ZigBee Alliance, 2008.

[25] "ZigBee Home Automation Public Application Profile." ZigBee Alliance, 2010.

[26] Bluetechnix GmbH. `http://www.bluetechnix.com`.

[27] OpenCV. `http://opencv.willowgarage.com`.