TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutitehnika instituut

IAY70LT

Elary Saagim 111402

SMART LED ILLUMINATION CONTROL SYSTEM BASED ON AN IEEE 802.15.4 WIRELESS PERSONAL AREA NETWORK

Master thesis

Supervisor: Thomas Hollstein

Professor

Dependable Embedded Systems

Co-supervisor: Priit Ruberg

Early stage researcher

Tallinn 2014

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Arvutitehnika instituut

IAY70LT

Elary Saagim 111402

IEEE 802.15.4 RAADIOPERSONAALVÕRGUL PÕHINEV TARK LED VALGUSTUSE JUHTIMISSÜSTEEM

Magistritöö

Juhendaja: Thomas Hollstein

Professor

Dependable Embedded Systems

Kaasjuhendaja: Priit Ruberg

Nooremteadur

Tallinn 2014

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Elary Saagim

27.08.2014

Abstract

Wireless communication is preferred in most smart embedded applications. Wireless network enables creation of flexible system with quick installation and easily changeable structure. The objective of this thesis is to create a wireless personal area network based extendible and scalable platform.

Thesis contains comparison of radio communication protocols based on IEEE 802.15.4 standard, the most suitable is chosen. Embedded operating system is additionally selected to ease development and enable better code portability.

Thesis includes hardware design. Particular aim is taken to enable possibility to use the created platform as a development platform and make it usable for further expansions, allowing additional devices to be connected.

An illumination control application is developed. The main emphasis is set on embedded system software development, a communication protocol is created, DMX protocol is implemented. Simple PC utility is designed to enable control of devices.

The designed system is implemented, hardware platform was assembled, software was written. Resulting system is usable for both small illumination solutions and also for larger installations.

The thesis is in English and contains 47 pages of text, 7 chapters, 6 figures, 5 tables.

Annotatsioon

IEEE 802.15.4 Raadiopersonaalvõrgul põhinev Tark LED Valgustuse Juhtimissüsteem

Traadita juhtimine on eelistatud enamustes tarkades sardrakendustes. Traadita võrk võimaldab kiirelt paigaldatava ja lihtsalt muudetava struktuuriga süsteemi loomise. Käesoleva lõputöö eesmärgiks on luua raadiopersonaalvõrgul põhinev laiendatav ja skaleeritav platvorm.

Töös on võrreldud erinevaid IEEE 802.15.4 standardil põhinevaid raadiosideprotokolle, sobivaim antud lõputöö projekti tarbeks on valitud. Lihtsustamaks arendustööd ning võimaldamaks parem koodi üleviidavus teisele riistvaraplatvormile on edasiselt võrreldud ka erinevaid sardsüsteemides kasutatavaid operatsioonisüsteeme.

Projekti raames on teostatud vajaliku riistvara disain. Disaini puhul on eelkõige eesmärgiks võetud selle kasutamise võimalikkus arendusplatvormina ja valmistatava riistvara edasine laiendatavus, võimalus kasutada lisanduvat riistvara ja seadmeid.

Valitud on operatsioonisüsteemi poolt toetatud raadioside moodul, mis võimaldab vajaliku raadioside kasutamise ning sisaldab ka mikrokontrollerit. Mugavamaks arenduseks on kättesaadavaks tehtud enamus mooduli kontaktidest ning võimalikuks on tehtud platvormi kasutamine maketeerimislaual.

Lõputöö projekti tarkvara disain keskendub valgustuse juhtimissüsteemi loomisele. Põhiline rõhk on seatud sardsüsteemi tarkvara arendusele, loodud on suhtlusprotokoll juhtseadmetele, võimaldatud on suhtlus juhtseadmetega üle kohtvõrgu. Implementeeritud on DMX protokoll.

Disainitu sai realiseeritud ning on praktikas kasutatav. Loodud süsteem on rakendatav nii väikeste valgustuslahenduste tarbeks kui ka suurematel installatsioonidel, kus osutub vajalik kasutada suuremal hulgal juhtseadmeid.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 47 leheküljel, 7 peatükki, 6 joonist, 5 tabelit.

Table of abbreviations and terms

6LoWPAN	IPv6 over low power wireless personal area networks
AES	Advanced Encryption Standard
BPSK	Binary Phase-Shift Keying, digital modulation scheme
BSD	Berkley Software Distribution
CRC	Cyclic redundancy check
DMX512-A	Digital Multiplex, asynchronous serial digital data transmission standard for controlling lighting equipment and accessories
EEPROM	Electrically Erasable Programmable Read-Only Memory
FAT	File Allocation Table, file system architecture
FFD	Full-Function Device
FIFO	First In, First Out, data buffer
GFSK	Gaussian Frequency-Shift Keying, digital modulation scheme
GNU GPL	GNU General Public License
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
ICMP	Internet Control Message Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IoT	Internet of Things

IP	Internet Protocol
IPv6	Internet Protocol version 6
ISP	In-System Programming
JTAG	Joint Test Action Group, (IEEE) standard 1149.1: Standard Test Access Port and Boundary Scan Architecture.
LAN	Local Area Network
LED	Light Emitting Diode
MAC	Media Access Control
MCU	Micro-controller Unit
MIC	Message Integrity Code
MPSK	Multiple Phase-Shift Keying, digital modulation scheme
O-QPSK	Offset Quadrature Phase-Shift Keying, digital modulation scheme
OSI model	Open Systems Interconnection model, a layered model of network architecture
PAN	Personal Area Network
PC	Personal Computer
PCB	Printed Circuit Board
PHY	Physical layer of OSI model
RF	Radio Frequency
RFD	Reduced-Function Device
RGB	Red-Green-Blue colour model

RNDIS	Remote Network Driver Interface Specification, protocol for ethernet link over USB
RTOS	Real-Time Operating System
SLIP	Serial Line Internet Protocol
TCP	Transmission Control Protocol
TIA-485-A	Telecommunications Industry Association standard for multi-drop differential network bus
UART	Universal Asynchronous Receiver/Transmitter
UDP	User Datagram Protocol
USB	Universal Serial Bus
WPAN	Wireless Personal Area Network

Table of contents

List of figures

List of tables

1. Introduction

Wireless networks are becoming the backbone of most smart systems. This is expected, as necessary technology has cheapen enough to make such advancements possible, price for chips with integrated radio and processor has came down to single digit. Going wireless seems to be a tempting alternative as well as wired systems are expensive to install and maintain.

There are multiple choices needed to be done to create a wireless platform. There is a wide choice of different networking options, ranging from point-to-point communication to full mesh networks, different embedded operating systems are available, enabling simpler and faster application development, wide selection of hardware is available. All choices must be done carefully to create a scalable and easy to use platform.

This has led to the initiation of this master's thesis. The project aims at creating a wireless platform which provides scalability and extendibility. As a proof-of-concept, a LED illumination application is created.

1.1. Purpose

The purpose of this thesis is to investigate options for and create an expandable wireless platform for smart control applications. Project aims to create a proof-of-concept solution which allows central control for LED controllers.

1.2. Objectives

This master's thesis work has three main objectives, these are given below.

1. Study different networking options and embedded operating systems fit for the project, select the most suitable.
2. Design an extendible and scalable wireless platform.
3. Develop software for created platform which allows the control of LED lights from a central device.

2. Requirement specification

Initial requirements are set to create an overview of the system and define the scope for the project. Wireless node and gateway are necessary components for the network. These must be universal and must allow extendibility and scalability. Platform specific requirements and illumination control system requirements are given separately.

2.1. Wireless node

Wireless node is an embedded system which contains 802.15.4 radio, MCU and supporting devices. Wireless node is responsible for sensor and actuator control, data processing and communication with gateway.

Various applications need different node behaviour. Some offload most of computations to computers and expect it to be a simple input-output link which removes the necessity for data wires in the system, others expect the node to do most of processing needed from application. To provide necessary flexibility, the system should divide functionality to separate different problems and solve those separately. Using an embedded operating system is beneficial as it allows to deal with necessary resource management, provides common services and abstracts hardware. Communication should be done using already existing protocols, ensuring more sustainable design.

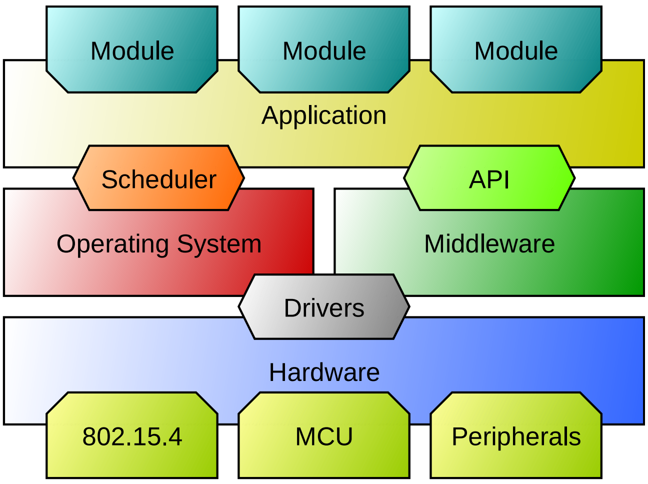


Figure 1: Wireless node structure

Requirements set for the wireless node:

1. Wireless communication using IEEE 802.15.4 WPAN with scalable protocol.
2. Operating system shall be used.
3. Portability to different architecture, vendor shall be retained.

2.2. Gateway

Gateway device is used to create a link between wireless network and external applications. It provides needed functionality to translate and exchange messages between different networks, make necessary protocol conversions, address translations and routing.

Requirements for the gateway:

1. Shall be able to communicate with IEEE 802.15.4 WPAN devices.
2. Shall provide means to make WPAN devices controllable from LAN.

2.3. Illumination control system

Requirements are specified for wireless DMX controller which drives illumination devices, and a network gateway which enables wireless communication between WPAN and LAN. Additionally, a PC utility is specified to verify system functionality and to provide control software for created application. The structure of LED illumination control application network is given on figure 2.

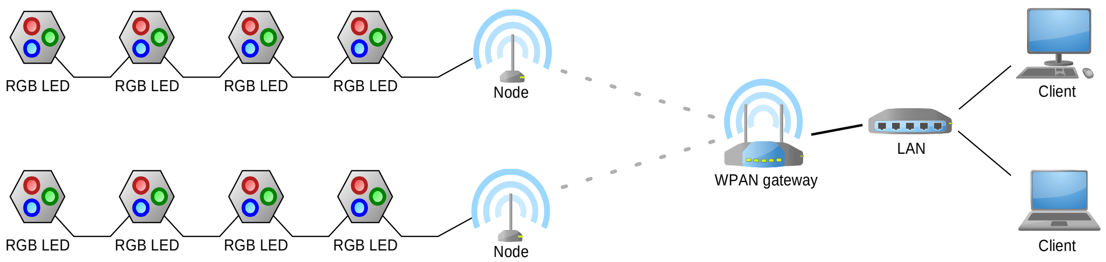


Figure 2: Planned network structure.

2.3.1. Wireless DMX controller

Wireless DMX controller must be controllable by client applications and be able to control DMX RGB LED lights. It must contain everything necessary to drive dimmable lights supported by the application.

Requirements set for the DMX controller:

1. RGB illumination shall be supported.
2. At least 8 RGB lights must be controllable with one node.
3. Multiple clients may connect and set colour of RGB LEDs.
4. DMX512 protocol shall be supported.
5. Node shall store basic information about itself:
    1. Following fields must exist, be client readable and writeable:
        1. name of the node, at least 16 bytes
        2. count of RGB lights to be controlled
        3. name for each RGB light
        4. network channel of the node.
    2. Following data must be readable from device:
        1. current colour of RGB LEDs

2.3.2. PC utility

For development, verification and demonstration purposes, simple PC utility shall be created. It shall have a graphical user interface which enables controlling LEDs colour and changing settings stored in nodes.

Requirements for the application:

1. Shall be able do detect wireless nodes connected to the network
2. Shall provide user interface to read and write wireless node parameters
3. Shall provide interface to set colour of any RGB LED connected to the wireless node

3. Background

3.1. Overview of IEEE 802.15.4 standard

IEEE 802.15.4 is a standard by the Institute of Electrical and Electronics Engineers which defines PHY and MAC layers for WPANs. WPANs are simple and low cost networks used for short distance communication. They are designed to work with minimal or no infrastructure to allow power-efficient and inexpensive solutions to be implemented for a various applications [1].

The standard supports multiple different frequencies. Some variations exist in PHY for different frequency bands, those include chip rate, modulation and symbol style and rate. Medium bit rates vary accordingly. Though variety of frequencies and modulation types are supported, 2.4 GHz radios with O-QPSK modulation are dominant in the market. Main frequencies and corresponding parameters are given in table 1 [1].

Table 1. Main frequency bands and data rates for radios defined by IEEE 802.15.4.

Frequency band (MHz)	Modulation	Bit rate (kb/s)	Region
779 – 787	O-QPSK	250	China
779 – 787	MPSK	250	China
868 – 868.6	BPSK	20	Europe
902 – 928	BPSK	40	USA
950 – 956	GFSK	100	Japan
950 – 956	BPSK	20	Japan
2400 – 2483.5	O-QPSK	250	Worldwide

Two types of devices are defined: a full-function device (FFD) and a reduced-function device (RFD). FFD is capable of serving as a PAN coordinator or a network coordinator. At least one FFD in the network must operate as a PAN coordinator. Device lacking the functionality to be a coordinator is a RFD. RFDs are usually highly constrained devices lacking computational power, memory, or energy for routing [1].

Each device in the network has an unique 64-bit identifier called extended address. PAN coordinator allocates 16-bit identifier unique within the network to every device associated to the network. Both of those addresses can be used for direct communication [1].

The standard defines two network layers: Physical layer and media access control (MAC) sub-layer. Physical layer controls the transmission and reception of data over physical radio unit. The most important tasks of MAC sub-layer are beacon management, channel access, frame delivery and device association [1].

As majority of applications are power constrained, limiting power consumption by radio duty-cycling is done. Radio is set to listen to RF channel periodically to receive pending messages. Interval between two sequential listening periods can be defined by the application. This enables designer to decide on power optimization strategies [1].

Symmetric-key cryptography is used for frame protection. Security is optional, it is up to the developer to decide which security elements are applied if any. There are total of 8 different security levels defined by the standard. Those are defined by combining two security options, message integrity code (MIC) and payload encryption option. MIC option defines encrypted authentication tag usage and its length. 4 different tag lengths are defined: 0 bits (tag not used), 32 bits, 64 bits and 128 bits. All lengths are allowed with and without payload encryption option. Standard requires implementation of at least 0 bits and 64 bits tag length and both payload encryption options. Key maintenance is done by application [1].

3.2. Overview of DMX512-A

DMX512-A is an asynchronous serial digital data transmission standard for controlling lighting equipment and accessories. It is universal, manufacturer independent standard developed by the Engineering Commission of United States Institute for Theatre Technology (USITT) and approved by the American National Standards Institute [2].

TIA-485-A standard electrical specifications are used. TIA-485-A standard defines balanced differential multi-drop communication link for digital data transmission. Usage of balanced differential data link increases immunity to noise. Additionally for two wires used for differential data, a common wire is used to limit common mode voltage. This adds additional noise immunity holding differential link voltage levels within receiver input limits [2].

DMX512-A physical layer defines two data links, a primary and an optional secondary. Primary link is normally used as an unidirectional link to transmit packets to network nodes. While enhanced modes are defined in a standard which does allow the use of primary link in bidirectional mode and usage of secondary link as either uni- or bidirectional link, no interoperable data structures or protocols for those modes are defined, usage of those modes is manufacturer specific [2].

The standard uses UART for data transmission. Data Communication is done using 1 start bit, 8 bit data and 2 stop bits format with symbol speed of 250kBd. Data packet consists of line reset (break and mark conditions), start code and payload [2].

Line is reset with low signal for at least 92us (break condition) and subsequent high signal (mark condition) for at least 12us and at most 1s. Start code byte must follow to mark the protocol used. For default 8-bit untyped information this must have value zero, alternate values are defined but are beyond standard use cases. After default start code, up to 512 data slots may be transmitted. Slots are transmitted sequentially, starting from the slot 0 and ending with the last implemented slot, mark conditions with less than 1s may be used between slots when necessary [2].

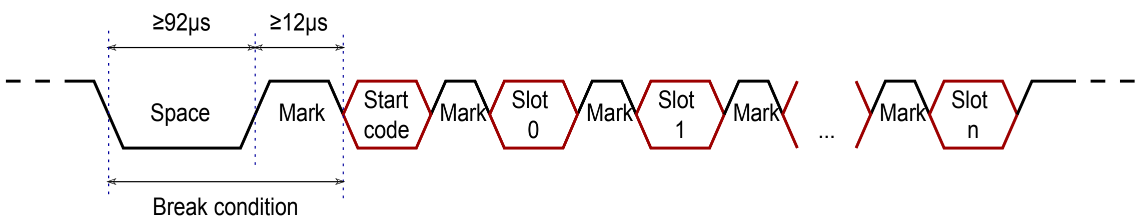


Figure 3: DMX packet.

4. Selection of supporting software

4.1. Network protocol selection

IEEE 802.15.4 specifies PHY and MAC layers. These are not sufficient for direct use in networking applications as only basic communication framework is provided. For additional network functionality, faster and easier development, a network protocol is selected.

There is a wide selection of protocols for IEEE 802.15.4 radios. Many protocols exist but are often with limited support, poor documentation or buggy implementations. To limit selection candidates, some initial requirements are set. Limitations are set to avoid using obsolete, unverified or badly defined protocols. Limitations for initial selection are given below.

1. Protocol must support self-organizing mesh networking
2. Network protocol must have stable implementation released
3. Implementation must have active development, support
4. Source code of the implementation should be available
5. Protocol should be in wide use
6. Manufacturer specific protocols should be avoided

Total of 4 protocols were found which satisfy initial requirements: ZigBee, 6LoWPAN, WirelessHART and MiWi PRO. Selected protocols are further examined below, supported network structure, security options, legal and availability are investigated.

4.1.1. ZigBee

ZigBee Alliance offers a set of standards designed to provide low-power decentralized wireless sensor and control networks [3]. Two feature sets are available, ZigBee and ZigBee PRO. ZigBee PRO is the most widely used specification, mainly for its better support for low power and large mesh networks [4]. Main advantages of ZigBee protocols are stated to be product interoperability and vendor independence [5].

Network consists of three type of devices: ZigBee coordinator, ZigBee router and ZigBee end device. ZigBee coordinator is responsible for network parameter selection, network initialization and device association. ZigBee router is a network node with routing capabilities and device association support. ZigBee end device is a network node without any coordinator or router capabilities. Supported network topologies are star, tree and mesh. [6]

ZigBee specification defines network up to application framework layer. Application layer provides functionality for service discovery, device binding and data transmission between application entities. One device can have up to 240 distinct application objects. Multiple application profiles are defined to enable interoperable device development. RGB dimming is defined in ZigBee Home Automation profile, further simplifying control application development. [6]

Data encryption is done using symmetric keys. Basic reply protection is provided, messages include counter for messages to ensure uniqueness. Counter is only reset if new network key is used. Unencrypted network keys are only sent over the air on unsecured join. Unsecured join is usually disabled to limit vulnerability, enabling must be explicit user interaction when new devices are associated with the network. [6]

Standards provided by ZigBee Alliance are free to use for non-commercial purposes. For commercial purposes, membership of the ZigBee Alliance is required. Membership is not required if ZigBee brand is not used, certification to get ZigBee Certified status is not requested and intellectual property of ZigBee is not used for commercial gain. [5]

Most vendors of IEEE 802.15.4 radio chips provide ZigBee stack for their hardware. Those include Atmel with its BitCloud Stack, Silicon Labs with Ember Stack, Texas Instruments with Z-Stack. These stacks are using IEEE 802.15.4 radio directly, no additional intermediary software is needed.

4.1.2. 6LoWPAN

6LoWPAN is an acronym for IPv6 over Low power Wireless Personal Area Networks. This protocol is defined by the Internet Engineering Task Force (IETF) 6LoWPAN working group to standardize transmission of IPv6 packets over 802.15.4 networks. Extending IP network to low power networks is done to add simple interconnectivity with other IP networks. [7]

IETF uses IEEE 802.15.4 defined device names. Network consists of FFDs and RFDs. FFDs do the mesh forwarding, RFDs only search for FFDs which would do forwarding for them. FFDs can be extended to route data to external network, acting as a gateway for the network. [8]

Protocol defines network up to transport layer. Standards include UDP header compression to minimize packet size. Direct end-to-end Internet integration is possible, through usage of IPv6, UDP and ICMP. Since IPv6 requires support of packet sizes much larger than the largest IEEE 802.15.4 frame size, an adaptation layer is defined, which allows IPv6 packet fragmentation [8].

6LoWPAN uses security defined by IEEE 802.15.4, additional security is not defined. Multiple possibilities are named in [9] but any specific recommendations are not done. Security measures and implementation is left to application.

6LoWPAN protocol is defined in open standards, rights are granted for both private and commercial use. IETF encourages the use of their standards describing protocols. [10]

Multiple embedded operating systems have 6LoWPAN stack integrated. Some have additional stacks available by third party developers.

4.1.3. WirelessHART

WirelessHART is a protocol defined by HART Communication Foundation. It was designed to enable secure industrial wireless sensor network communications. Emphasis is set on security of the network. Protocol is time-synchronized, time accuracy to 1 millisecond is available across the network. Latency and jitter problems are addressed and minimized. [11]

Network consists of three device types: wireless field device, gateway and network manager. Wireless field devices are carrying sensor and actuator functionality. Gateways enable communication with external devices through different networks. Network manager deals with network configuration, communication scheduling and route management. Frequency hopping is used to provide better noise immunity. [12]

WirelessHART protocol defines network up to application layer. Application layer handles communication between gateway and devices. Commands and responses are main message types used as data carriers, payload content can vary by implementation. Communication medium is not visible at application level, there is no difference between wired and wireless versions of HART protocol. [13]

Security is built in and cannot be disabled. It is implemented with end-to-end sessions utilizing AES-128 bit encryption. Separate keys are used for both broadcast and authentication process for additional protection. Methods for key exchange, key updates are defined. [14]

WirelessHART protocol is defined in IEC 62591 standard. Standard is closed. Membership is required to gain access to protocol specifications, development and test tools, support.

Nivis has WirelessHART stack available under the GNU GPL-3.0 license. Linear Technology provides SmartMesh WirelessHART stack.

4.1.4. MiWi PRO

MiWi PRO is Microchip's proprietary wireless network protocol. It targets applications that require large network support. MiWi PRO is backward compatible with MiWi networks. [15]

The MiWi protocol defines three types of MiWi protocol devices, based on their functions in the network: PAN Coordinator, Coordinator and End Device. MiWi PRO is configured to support wireless networks with up to 64 Coordinators. It provides three network configurations: star, cluster and mesh. [15]

The protocol provides the features to find, form and join a network, as well as discovering nodes on the network and route to them. It does not cover any application-specific issues, such as how to select which network to join to, how to decided when a link is broken or how often devices should communicate. Joining procedure is done by application. [15]

As with 6LoWPAN, IEEE 802.15.4 security features can be used. Stack supports all 7 security modes defined by IEEE 802.15.4. Any additional security measures are left to application to handle [16]

MiWi is a proprietary protocol. Microchip provides stack free for both commercial and non-commercial use. License limits provided stack usage to devices provided by Microchip. [17]

4.1.5. Fitness for application, selection

ZigBee Home Automation profile covers most of requirements set to the application. Multiple profiles defined cover wide range of actuator and sensor functionality a smart device could need, ensuring extendibility and interoperability. Any additional functionality not implemented can be added as manufacturer specific endpoints. Security is considered, methods for securing network have been developed to provide enough security particular application would need. Overall interoperability with wide range of ZigBee device providers further increases its favourability. The only negative aspect of ZigBee is its license. Requirements were set to allow usage in commercial applications, using ZigBee intellectual property is forbidden for that objective without becoming a member of ZigBee Alliance, and membership is not free. Usage of intellectual property of ZigBee Alliance can be avoided while still using network based by ZigBee stack but is unfavoured as deeper understanding of the protocol is necessary for optimal usage.

6LoWPAN provides extension to already existing IP networks, allowing end-to-end connections with devices, making gateway transparent for network. Standard routing can be used making WPAN devices visible and accessible directly from LAN or even from Internet making it a sound candidate for current project. Network security options are limited but sufficient for this application. Usage of standards defining 6LoWPAN is not limited for any purpose. Wide usage is expected in the future as IP networks are already the backbone for almost all communication networks, almost all homes already have the required infrastructure.

WirelessHART provides good solutions to time-critical control applications. Goal of this project would profit from those as timing of illumination switching needs to be precise. High security provided by protocol is another supporting argument for this protocol. As with ZigBee though, usage of intellectual property is restricted. Contrary to ZigBee, WirelessHART specification is not open, making it difficult to understand its methods and using it efficiently even if investigated for academic and educational purposes. Selection of this stack is therefore unfavoured.

MiWi PRO is provided to enable fast software development for Microchip devices without worrying too much about network management functionality. It provides everything necessary for WPAN. Simplicity is its main strength, emphasis is set to simplify every aspect of usage, starting from network security settings to packet transmission logic. As it is locked to specific vendor, no real advantage over other protocols exist.

Initial requirements were enough to filter out unfit protocols for current project. All protocols selected for further investigation are all well usable for required task. They provide necessary functionality for network forming, managing, and routing, leaving only application specific problems to solve. Legal constraints are influential though, being enough to provide good reasons to avoid 3 protocols investigated leaving 6LoWPAN in favoured position. ZigBee has existing support for illumination control suitable for this project, extendibility and availability does make it a better alternative for current application. This is alleviated by the fact that there is no stack available that does provide support for multiple vendors, making application portability hard to achieve. Taken this into consideration, 6LoWPAN protocol is selected for this project.

4.2. Embedded operating system selection

Software development for embedded platform should be easy and straightforward. Bare metal programming should be avoided – it is time consuming and may cause different problems when multiple different software components are separately developed. Additionally, written code will not be easily portable – different devices have different peripheral and methods to use those.

To achieve better code portability, avoid locking to one vendor or MCU and enable easier application development, an operating system will be used. This will remove most of low level dependencies as hardware abstraction layer will cover usage for most typical tasks.

There are multiple embedded operating systems available. As with network protocol selection, in-depth evaluation for given application for all available operating systems is not practical, some initial requirements were set.

Criteria set for initial selection is given below.

1. Support for multi-threading is required.
2. Basic hardware abstraction must be provided.
3. Selected operating system must not be defunct.
4. Operating system must have stable releases, active development, support.
5. Documentation must be available.
6. License must allow free use both commercial and non-commercial applications.
7. Operating system must be open source.
8. Must be able to run on resource constrained platforms.

Total of 4 operating systems were found which satisfy initial requirements: FreeRTOS, TinyOS, Nano-RK and Contiki. Selected operating systems are further examined, functionality, availability and legal considerations are viewed.

4.2.1. FreeRTOS

FreeRTOS is a real time operating system (RTOS) provided by Real Time Engineers Ltd. It has support for wide range of processor architectures. Official releases have no community contributions used to enable strict quality control and remove IP ownership ambiguity. FreeRTOS is a mature operating system with wide user and developer base. It is heavily documented, multiple books and manuals are for sale, many guides for learning are available for free. [18]

FreeRTOS is licensed under modified GNU GPL. Operating system can be used for both commercial and non-commercial applications but additional terms apply. Permission is given to link FreeRTOS with independent modules and distribute result under terms of user provided that communication with modules and FreeRTOS happens only through FreeRTOS API. Limitations on usage are also set, FreeRTOS may not be used for any competitive or comparative purpose. [18]

NanoStack is 6LoWPAN stack by Sensinode Ltd. built on top of FreeRTOS. [19] NanoStack is proprietary and must be licensed, multiple chip vendors have licenses to use this stack with their chips.

Main features provided by FreeRTOS [20]:

1. Priority based scheduler
2. Support for real time tasks
3. Preemptive and cooperative scheduling supported
4. Tick-less kernel
5. Stack overflow detection
6. Task priority assignment
7. Memory management
8. FAT file system support possible
9. GUI support using µGFX

4.2.2. TinyOS

TinyOS is an operating system designed for power-constrained wireless systems. It is managed by TinyOS Alliance. Documentation is available on the web, search from documentation and mailing lists is possible, books have been released. Multiple wireless sensor network platforms are officially supported. [21]

TinyOS is written in nesC programming language, a language built as an extension to C. It is defined and used to prevent overuse of macros and maximize optimization. Programming on the platform can be done in pure C if necessary or preferred. [21]

TinyOS is licensed under a 3-clause BSD license. Usage of developer names to promote or endorse products is prohibited, redistributing source code is not required, inclusion of copyright messages in documentation is.

Main features provided by TinyOS [21]:

1. Event driven architecture
2. Single shared stack
3. 2-level FCFS scheduler – events and tasks
4. Support for preemption
5. 6LoWPAN support with multi-hop routing
6. Flooding time synchronization protocol support

7. Collecting tree protocol support
8. Simple file system support (TinyFS)

4.2.3. Nano-RK

Nano-RK is a real-time operating system from Carniegie Mellon University. It is specialized for usage in wireless sensor network applications. Energy consumption is considered in controller power handling, kernel tasking and provided wireless communication protocols. FireFly sensor networking platform and MicaZ modes are supported.

Features:

1. Cooperative scheduling
2. Preemptive scheduling
3. Real time priority based scheduling
4. Built-in fault handling with task timing checks
5. Tickless kernel
6. Multiple wireless link layer protocols supported
7. 6LoWPAN support
8. Serial line SLIP interface for IP connectivity

4.2.4. Contiki

Contiki is an open source operating system designed to be used for low-cost, low-power systems. Design considers operating system to be mainly used for IoT. Contiki OS supports IP networking natively. Full networking stack is integrated to the operating system, both IPv4 and IPv6 support is available, TCP and UDP protocols are supported, additional header compression defined by 6LoWPAN is implemented. [22]

Contiki is made to enable rapid development. It comes with extensive set of examples for various different applications and a network simulator. Virtual machine with development environment set up is provided. [22]

Contiki source is licensed under a 3-clause BSD license. Usage of developer names to promote or endorse products is prohibited, redistributing source code is not required, inclusion of copyright messages in documentation is. [22]

Important features of Contiki [22]:

1. Multithreading support using protothreads
2. Earliest deadline first scheduling support
3. Preemptive task support
4. Support for IP Networking with UDP and TCP support
5. Additional light-weight network stack for extremely constrained devices
6. 6LoWPAN support with multi-hop routing
7. Managed memory allocation support
8. Lightweight flash file system support
9. Power consumption estimation
10. Radio duty cycling on network routers
11. Dynamic module loading

4.2.5. Fitness for application, selection

FreeRTOS is most widely used embedded operating system. It is mature and has good support, various tools for development are released. Licensing allows usage for both commercial and private use. Support for 6LoWPAN is provided by third party and is closed source which limits hardware to selected devices.

Nano-RK has support for 6LoWPAN but it is not yet usable in real-world applications as-is, many parts of 6LoWPAN is still under development. Finishing implementation would be time consuming.

Both TinyOS and Contiki are both good candidates for this project, they have finished implementations of 6LoWPAN stack, they are widely used and have good support. TinyOS is expected to have a steeper learning curve as it uses programming language specific to only this operating system, therefore Contiki is favoured.

5. Hardware

Hardware used for WPAN devices shall be based on devices which are supported by selected operating system. As initial design sets emphasis to create a platform for both development and prototyping the application, quantity of produced nodes will remain low, when larger quantities are needed application specific design should be used. Design already incorporates necessary components for DMX controlling.

5.1. Gateway

Gateway is a standard component of 6LoWPAN network, special application specific implementation is not necessary. Contiki software supports border router functionality with either using SLIP which uses serial line connection with special drivers or RNDIS which uses native drivers. RNDIS is preferred as it allows any device supporting RNDIS network devices and routing to act as a gateway. RNDIS is currently supported on RzUSBStick only. As this module is not exceedingly expensive and only one per network is necessary, RzUSBStick will be used as border router.

Firmware upload and debugging has to be made on selected gateway. For that, Atmel debugger is necessary. Lowest cost option is to select Atmel AVR Dragon as programmer and debugger.

5.2. Wireless node module selection

Contiki lists supported hardware on its web page. This list includes different nodes and modules provided by various manufactures. To simplify development, using already supported module or node is preferred, most suitable platform needs to be selected. It is therefore reasonable to investigate suitability of listed platforms for current application.

Modules that have no radio are excluded from the selection. Additionally, Memsic Iris, Redbee-Dev, Redbee-Econotag, Sentilla JCreate, Sentilla-USB, MSB430 and VirtualSense ESB are excluded from the selection as they are listed in the Contiki lost platforms [23], AVR-Raven and AVR-RCB are excluded as they are no longer supported by Atmel; MB851, MBxxx as STMicroelectronics does not recommend these for new designs; Zolertia z1, MICAZ, TMote Sky, Seed-eye, Econotag, Sensinode as they are not available in Estonia. This leaves a total of 6 modules to choose from: EVAL-ADF7023DB1, cc2538dk, exp5438, AVR ZigBit, ATAVR128RFA1-EK1 and CC2530DK.

Main criteria for module selection are available interfaces, size and cost of the module. UART is necessary to communicate over TIA-485. Additional GPIOs are necessary to control TIA-485 mode and external devices. As RF design is complicated, time-consuming and expensive, it is not reasonable to design a platform from scratch for small quantities. This means that selected module will be used in future applications and prototypes. Smaller size is favoured enabling a more compact design. Cost of the module should be low as well allowing cheaper prototypes.

Atmel ZigBit is selected for current application as it is smaller and cheaper than others, has necessary interfaces and plenty of GPIOs.

5.3. Printed Circuit Board

Typical wireless node contains 4 main components: power supply, MCU, radio and input-output. Current project has additional need for TIA-485-A driver.

Current project focus is set on controlling lighting devices which are powered by mains. It can be concluded that power is not a limited resource and low-power design is not necessary. Cheap mains-isolated 5V USB power supplies can be used, any additional voltage regulation can be done with linear regulators. This simplifies node design significantly and saves PCB space.

Board will mainly be designed for development purposes and for prototype application. It is reasonable to make the circuit board as universal as possible, avoiding multiple design iterations and also allowing additional future development. Component count should be minimal to save PCB space, design and assembly time, keep design simple and manageable.

Circuit board shall be designed to fit to solder-less breadboard so that temporary prototyping and experiments can be easily done on the breadboard without any loose parts stressing interconnecting wires. Furthermore, it is reasonable to make as many pins available to external devices as possible.

Basic interaction should be possible with module by means of buttons and LEDs. While it is easy to add buttons and LEDs on a breadboard, it is reasonable to include those already on PCB as they are beneficial in development of application, they would minimize development setup time, ease testing with multiple devices and may serve as possible interfacing option for application. Additionally, reset button should be put on the board. Resetting the device is frequently necessary in development and testing of application.

5.3.1. PCB components

Main criteria for component selection are availability and size. When possible, generic components are used to avoid locking to specific vendor. E12 series resistors and capacitors are used as these are widely available. They can be replaced by more accurate versions when necessary without the need to do new calculations, given values are present in series which provide values with more precision.

Board will be designed as small as possible while still containing everything necessary for development. Smaller size components should be selected if reasonable. Through-hole components should be avoided, they are bulky and usually more prone to physical damage.

Central component of designed PCB is the ATZB-24-A2 ZigBit module. It is a 13.5mm x 24mm module which contains ATmega1281 MCU, AT86RF230 RF transceiver and antenna. Multiple interfaces are made available on the module pins allowing usage on various different designs. Module contains all necessary necessary active and passive components, no additional components are required.

ZigBit module has allowed supply voltage range between 1.8V and 3.6V with additional limitation noting that if supply voltage is below 2.7V, MCU must be operating at 2MHz speed. As this design is not power-constrained, MCU under-clocking will not be necessary. 3.3V linear regulator will be used to provide power to ZigBit module.

Programming pins must be routed to accessible connectors to allow program upload to MCU. Two options are available for ATmega1281, ISP and JTAG. ISP connectors are more compact than JTAG connectors, therefore preferred on this design. Atmel standard 6-pin ISP pin-out is used and a header connector with 0.1 inch spacing will be added for that purpose.

Decoupling capacitor is added for ZigBit module. While module itself contains necessary capacitors and inductors for operation, it is reasonable to have additional decoupling capacitor at power supply lines to minimize noise spreading. [24]

To avoid accidental resets due to external noise an additional external pull-up resistor is added reset pin. To allow DebugWIRE functionality of the device to work, the pull-up must not be smaller than 10kΩ. [24] Selecting from E12 series resistor values, minimum resistor value of 15kΩ is allowed.

Communication with DMX devices is done using TIA-485-A transceiver. There is wide selection of devices with different properties available for that purpose. As current project aims to drive DMX chains, chip with good transmit properties is preferred. Internal protection is necessary to keep external component count low. Galvanic isolation of TIA-485 bus is not necessary as device has no unisolated connections to other devices, power source is isolated, communication is done over radio. Higher voltage devices are preferred as they enable higher driver output voltage swing, therefore increasing noise immunity.

Wide selection of TIA-485 transceivers is available, multiple vendors have their own versions of transceiver with different properties. At the moment of selection, total of 4 vendors with suitable transceiver are available: Analog Devices, Texas Instruments, Maxim, Linear Technology. All listed vendors provide SOIC-8 packed transceivers with 75176 compatible pin-out which cover set requirements. Required external components do not depend on used chip but are generic, only necessary external components are decoupling capacitor and optional bus termination resistors. Component is therefore generic, it is possible to change used chip for any suitable alternative making design independent of selected manufacturer.

For current application, a 5V powered 60V fault protected TIA-485-A transceiver will be used. Interfacing it with MCU does need some additional attention as MCU works on 3.3V levels. Selected chip does have 3.3V logic compatible inputs, those lines may be connected directly. Connecting it to ZigBit module is done through resistor to limit current, ATmega1281 internal clamping diodes are used to hold input at correct voltage level. Maximum allowed continuous current for clamping diodes is limited to 1mA [25]. Maximum voltage difference between input pin and receiver output is difference between minimum allowed supply voltage of ZigBit module, 2.7V and maximum allowed supply voltage of TIA-485-A transceiver, 5.5V. Minimal resistance is therefore 2.8kΩ. Selecting from E12 series resistor values and considering allowed margin, minimum resistor value of 3.3kΩ is allowed.

TIA-485-A bus needs termination resistors on DMX512 speeds to minimize signal reflection. On current application this resistor is not necessary on board as DMX512 transceiver will only be used for sending and therefore reflection on the board side will not be a problem but it will be added regardless to provide possibility to use designed hardware on other projects. Value for this resistor should be equal to cable characteristic impedance, which usually is around 120Ω.

Four tactile switches are included on platform as basic input devices. Three switches are used to provide generic input, one switch is used for resetting the device using reset pin. Debouncing of switches is done in software to keep component count minimal.

To avoid damage to device and external components, switches will be connected with series resistors. Value for resistors shall be high enough not to cause over-current on MCU pins when MCU pins are set as output but switch is accidentally pressed and low enough to ensure correct reading when switch is pressed. Calculations for allowed resistance value range must be made.

Minimum resistance should be selected to avoid over-current on pins when they are accidentally configured as outputs. Maximum current of 40mA per pin is allowed. Maximum voltage difference between pin and ground is equal to maximum allowed voltage on ZigBit module, 3.6V. Therefore, minimum allowed resistance is 90Ω. Selecting from E12 series resistor values and considering allowed margin, minimum resistor value of 100Ω is allowed.

Internal pull-up resistors are used for buttons to minimize component count. Pull-up resistors have values ranging from 20kΩ to 50kΩ on GPIO pins[26]. On reset pin, internal pull-up has value in range from 30kΩ to 60kΩ, with additional 10kΩ external pull-up resistor, minimum total resistance is 7.5kΩ. Maximum voltage on pin guaranteed to be read as low is 0.1Vcc. System can be viewed as resistor based voltage divider, maximum resistance between ground and input pin is 750Ω when switch is shorted. Selecting from E12 series resistor values and considering allowed margin, maximum resistor value of 680Ω is allowed.

Range of resistor values are allowed. To select specific one, mean value between minimum and maximum is taken to avoid extremes. With minimum allowed resistance value of 90Ω and maximum allowed value of 750Ω, mean value would be 420Ω. Selecting from E12 series resistor values closest available resistor value is 470Ω.

Four LEDs are included on platform. A green LED is connected to power rail which is used by ZigBit module to indicate its power state. Three additional LEDs are included and connected to ZigBit GPIOs and used as software controlled indicators. To make software indicators easily distinguishable, red, green and blue colors are used. Given LEDs are for indicative purposes, high and exact luminous intensity is not necessary. As with TIA-485, specific model and manufacturer of the LEDs is not important and it can be changed to suitable alternative later if necessary.

Current on LEDs must be limited, series resistor is used. Minimum resistor value allowed for current limiting resistor is calculated. Minimal forward voltage for red LED is 1.75V, for green LED is 2.7V, for blue is 2.90V. All LEDs have maximum forward current of 20mA. Maximum voltage difference is equal to maximum allowed voltage on ZigBit module, 3.6V. To keep unique component count low, usage of same value resistor for all LEDs in platform is preferred. As red LED has smallest forward voltage it requires highest value current limiting resistor. Therefore resistor value calculated for red LED will be suitable for all other LEDs as well.

Voltage difference on current limiting resistor pins in case of red LED is 1.85V. Minimum current limiting resistor value for red LED is therefore 92.5Ω. Using E12 series resistors and considering allowed margin, minimum resistor value of 150Ω is allowed.

Maximum resistor value is not precisely definable, with higher resistor values LED luminous intensity lowers. To keep unique component count low, same resistor value as used in switches can be used for LEDs. It is possible to select different value which would still be suitable for both switches and LEDs but there are no practical reasons for selecting any other value.

3.3V linear voltage regulator is needed to supply voltage to ZigBit module. Absolute maximum allowed current through module power pins is 200mA. This is selected as minimum output current requirement for the regulator. Additional current may be required when experimenting with additional hardware as prototype platform doubles as development board.

BA033T linear voltage regulator will be used for current project due to immediate availability. Its typical regulated output voltage is 3.3V. Minimum and maximum voltages are 3.13V and 3.47V staying inside MCU supply voltage limits. Low ESR input and output bypass capacitors are recommended with values at least 0.33uF for input capacitor and 22uF for output capacitor.

Support for additional linear voltage regulator will be added to design which would be used if input voltage is higher than 5V. This may be necessary on situations where higher supply voltage is already available removing necessity for any additional components. BA050T linear voltage regulator can be used. As with 3.3V linear regulator, input bypass capacitor and output bypass capacitor are required. Footprints will stay unpopulated in initial design.

5.4. PCB design

Circuit schematic design is rather simple as component count is kept at minimal and as ZigBit module already includes everything necessary for the MCU and radio communication. ZigBit GPIO pins are connected to 0.1” spacing pin rows if possible, to allow extending the platform. To allow using designed device on prototyping boards, power input pins are also connected to 0.1” spacing pins.

Standard 2 layer FR-4 PCB with 0.0347mm copper thickness is used on PCB layout. Highest theoretical current allowed on power supply traces is 1A, 0,25A on signal traces. If allowing maximum 10°C temperature rise on traces, power supply trace width should be more than 0.150mm and 0.022mm for signal lines. This project does not have necessity for thin signal lines and manufacturing designs with high-precision, ultra-thin

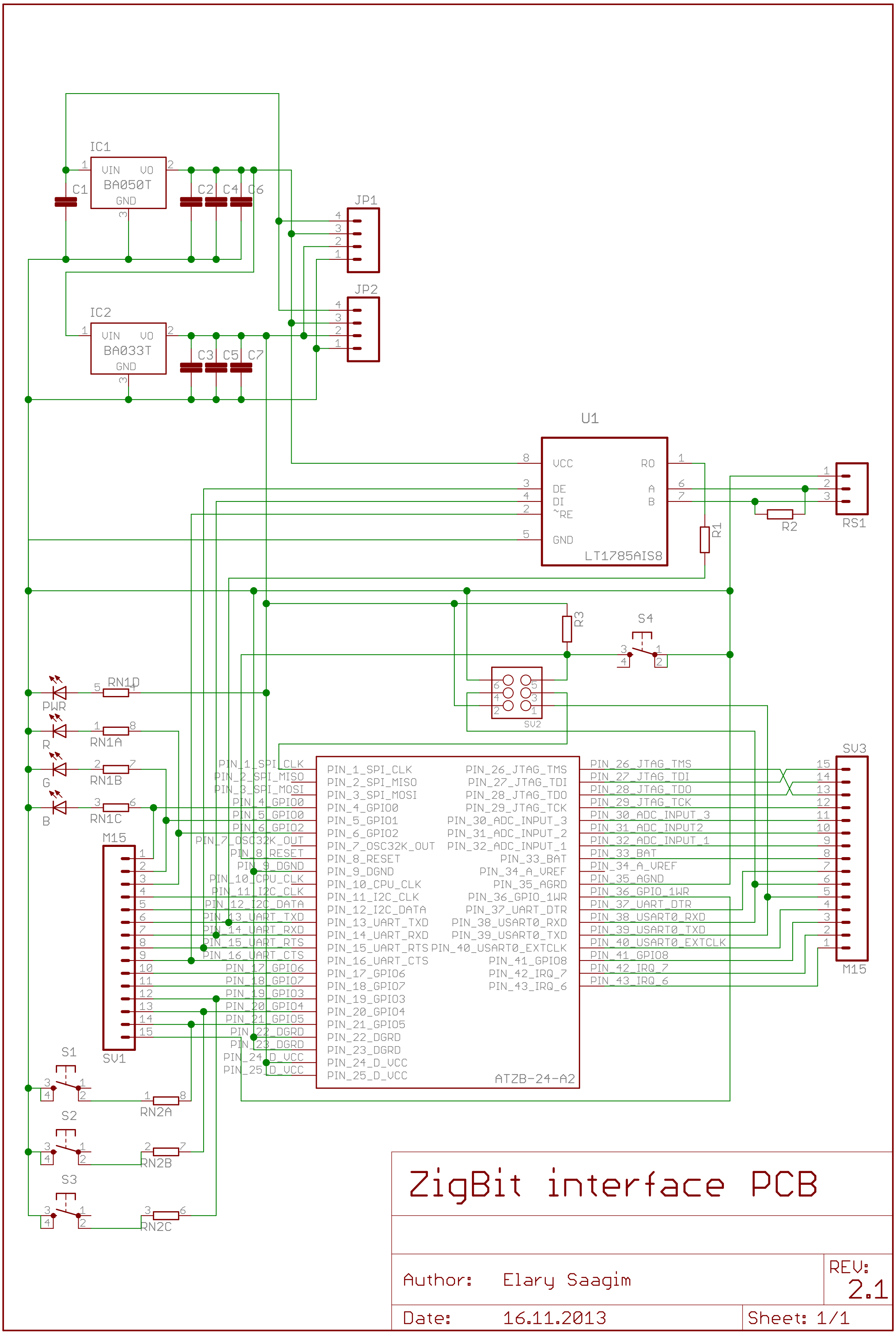


Figure 4: Created schematic.

is expensive. Therefore, thicker traces shall be used. At least 0.508mm (20mil) traces will be used for power supply traces, and 0.254mm (10mil) for other traces.

Components will be placed on both sides of the board. Large footprint SMD components will be used to allow hand assembly when necessary. Designed PCB layout is given on Figure 5. Assembled board is shown on figure 6. Bill of materials is given on appendix 1. Unpopulated PCB is shown on appendix 2.

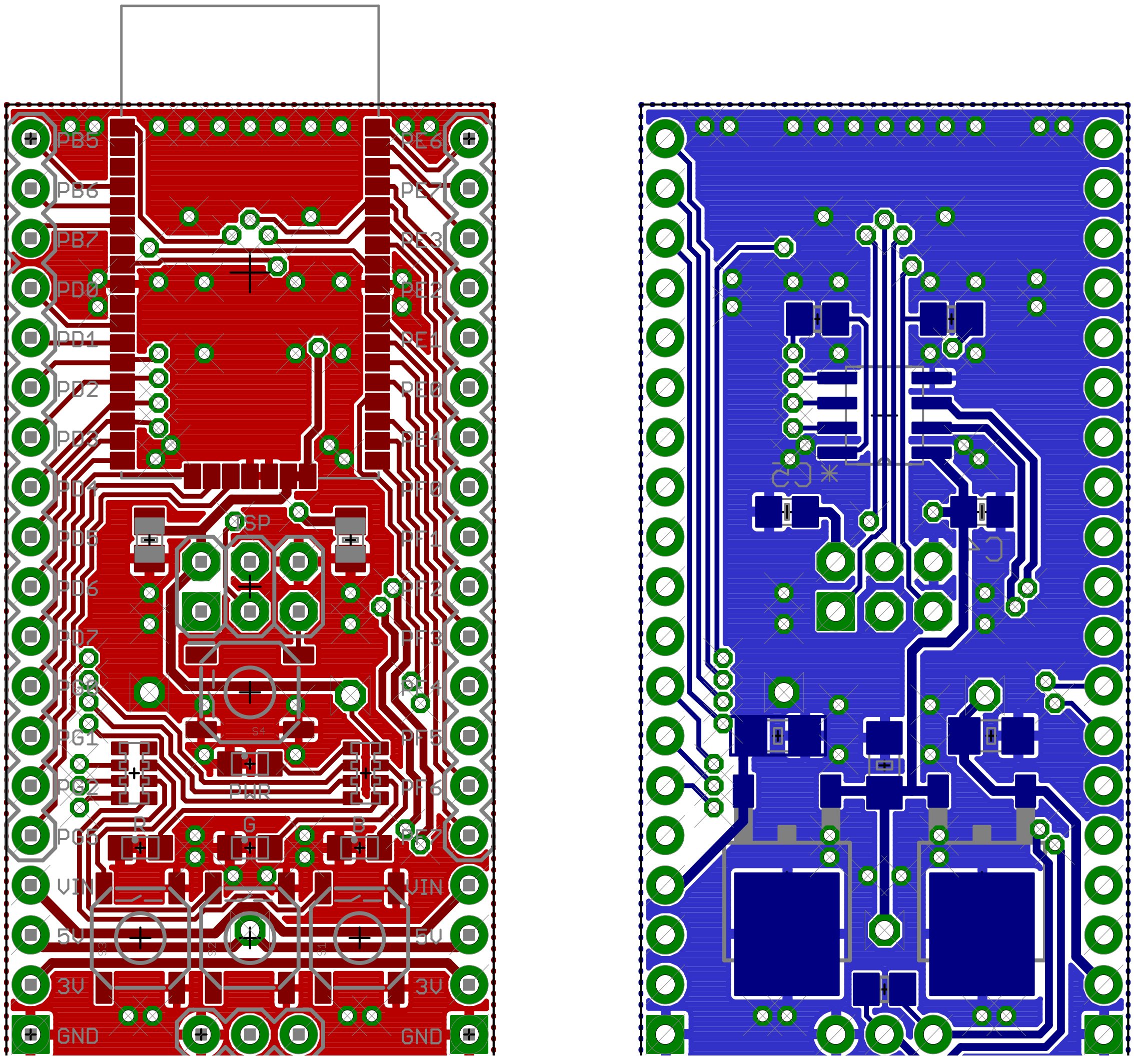


Figure 5: PCB Layout, top and bottom layers.

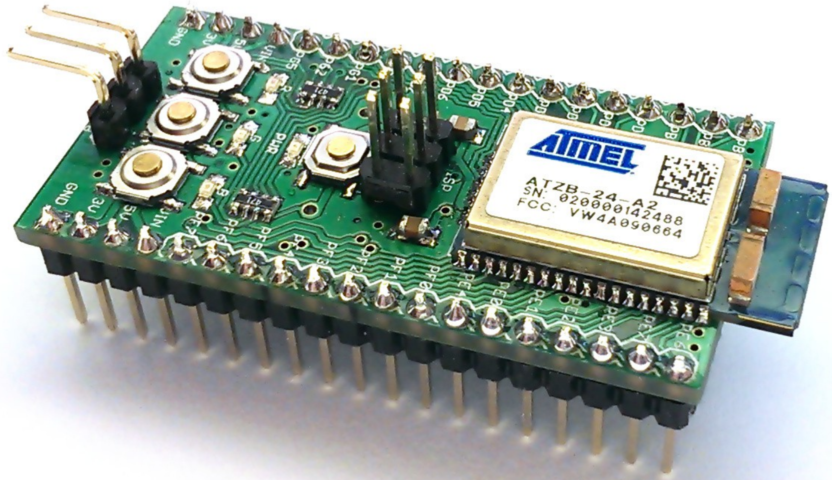


Figure 6: Assembled board.

6. Software

System has two embedded components, wireless node and wireless gateway. Both will use Contiki operating system and 6LoWPAN. Software development for the border router is not necessary, Contiki Jackdaw RNDIS border router with RPL is found to be suitable for the current project. Therefore, of embedded components, only wireless node needs software development for its application-specific functionality.

For development and verification purposes, PC utility will be developed. It will behave as a client application for the system and allow controlling of wireless nodes. Necessary functionality to demonstrate system operation shall be developed.

6.1. Wireless node software

Wireless node software can be divided into two distinct parts. Illumination control is the main task for this application. Second part consists of node configuration and information which can be expected to be seldom used. Communication protocols for these parts are defined separately to provide optimal performance and design.

6.1.1. Illumination control protocol

For illumination control, data is expected to be send to the node rather frequently when continuous illumination effects are used, infrequently when used as simple network controllable light switch. If fixed or frequent data update interval is used by client, lost packet may contain already stale information and be replaced by following packet at about the same time retransmission could have been done, therefore retransmission of lost packets should be optional.

WPAN throughput is limited and unreliable, it may not be possible to send colour values frequently enough to create smooth illumination effects. To compensate, some countermeasures should be taken. Initial delay value shall be introduced to compensate for latency in network. It allows the next control packet to be sent early and as soon as possible, making it possible to prevent peak network usages in cases when multiple values should change. Additionally, an optional fade effect with variable transition time is implemented. This reduces network load and allows guaranteed smooth transitions from one colour to other.

LED illumination control protocol is defined. First octet lower 7 bits define controlled RGB LED index, 8^{th} bit marks acknowledge request. Next three octets define RGB colour value. 5^{th} octet contains delay value, 6^{th} octet contains transition time. 5^{th} and 6^{th} octets are optional, with default values zero for initial delay and transition time. When acknowledge request is selected, received packet is echoed back to the sender. LED illumination control protocol packet structure is shown on table 2.

Table 2. LED illumination control protocol packet.

Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Ack req	RGB LED number						
2	Red colour component							
3	Green colour component							
4	Blue colour component							
[7]	Initial delay, unit size 1/32s							
[8]	Transition time, unit size 1/32s							

Packets with invalid length or RGB LED number are silently discarded. As an exception, empty packet will be acknowledged with last valid packet received. This does provide possibility to query last set values, thus enabling possibilities for cooperative controlling with multiple devices.

6.1.2. Node configuration and information protocol

Requesting information and configuring of a node is done using node configuration and information protocol. This protocol defines how to transmit function calls to node and receive function response. Functions are identified by numerical values to keep the packet length short. Additional function parameters are specific to particular function. These parameters are concatenated, forming a function payload.

Packet structure for node configuration and information protocol is defined. First octet contains command number, second octet contains octet count of command data. These octets are followed by optional and variable length command data. As default, multioctet numerical values are transmitted little-endian, text data is transmitted big-endian. Node configuration and information protocol packet structure is shown on table 3.

Table 3. Node configuration and information protocol.

Octet	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
1	Function number							
2	Function payload length							
[..]	Function payload							

Device and LED names must be readable and writeable. These values can be treated equally as generic string values, enabling use of one function to read and one function to write string values. LEDs are already indexed 0 to 7, same values are used for names. Device name is given index value 8.

Count of connected LEDs and network channel can be interpreted as 8 bit unsigned integers. As with strings, generic function can be used to read and write those values. For this purpose, count of connected LEDs is given index value 0, network channel index value 1.

Defined functions for this application are given on table 4. Respective responses are given on table 5

Table 4. Node configuration and information functions.

Function number	Payload length	Function name	Function parameters
0	1	Get string value	String index, 1 octet
1	1 + strlen	Set string value	String index, 1 octet String value, variable
2	1	Get 8 bit integer value	Integer index, 1 octet
3	2	Set 8 bit integer value	Integer index, 1 octet Integer value, 1 octet

Table 5. Node configuration and information function responses.

Function number	Payload length	Function name	Function parameters
0	1 + strlen	Get string value	String index, 1 octet String value, variable
1	1	Set string value	String index, 1 octet
2	2	Get 8 bit integer value	Integer index, 1 octet Integer value, 1 octet
3	1	Set 8 bit integer value	Integer index, 1 octet

6.1.3. DMX-512 driver

TIA-485-A driver controlling pins are driven with MCU GPIO pins. Current application needs the driver to be always transmitting, therefore pin values can be set when system is started, no additional changing of the values is necessary.

DMX-512 needs to send bytes over UART and control pins of TIA-485-A driver. Controlling UART is abstracted by Contiki but it needs some additional control logic to implement DMX-512 standard. Start condition of the DMX packet has break and mark conditions which are not transmittable using UART at 250kBd. To overcome that, UART shall be switched to slower speed.

DMX-512 protocol UART configuration is 1 start bit, 8 character bits, no parity bit and 2 stop bits. For fastest speed, over 9 out of 11 symbols should be for break condition. As maximum of 9 symbols can be used, breaking condition will require slower baud-rate than mark condition than mark condition. Baud-rate must not transmit more than 9 symbols in 92us, may not be faster than 97826kBd. Closest rate supported by Contiki is 57600Bd.

6.1.4. Implementation

Two main threads are used for the application: network communication thread and DMX control thread. Some initialisation is done before the threads are started, this includes setting default values to various variables and setting TIA-485 transmit and receive pins to enable transmitting DMX packets.

Network communication thread receives the packets, stores received information for further use and sends response if necessary. Received LED controlling information is stored in global variables as it is used by DMX control thread as well. Node configuration is stored on EEPROM, reading and writing is done when necessary.

Contiki supports TCP and UDP. Both TCP and UDP provide means for preventing packet data corruption by employing CRC. TCP provides reliable connection-based transmission, ensuring data packets are received in same order as sent, retransmitting lost packets if necessary. UDP provides more lightweight, smaller overhead, stateless message transport. For current application, UDP is more suitable, allowing smaller packets, faster transmission times, stateless connections and application-specific implementations for reliable communication.

6LoWPAN UDP provides header compression. Among others, short ports can be used. Header compression defined flag which sets limits port number to 4 bits. If the flag is selected, port value is offset by value 61616. This compression is used when possible, making 8 port values, from 61616 to 61623, highly preferable. Two different protocols are defined, these are controlled on separate ports. Port 616121 is used for illumination control, port 616122 is used for node configuration and information.

DMX control thread runs once every 1/32 seconds. Every time task runs, it transmits DMX start condition and iterates through illumination control packets, calculating RGB LED colour values and outputting them using the DMX controller.

6.2. PC utility

PC utility provides a simple interface to control connected nodes. Utility is designed to provide a necessary tool for testing and using functionality provided by the node.

Interface has possibility to set colour of any RGB LED connected to the system using RGB or HSV colour model. Node can be configured using the utility, enabling setting the name of the node or LEDs connected to the node. Screen capture of the utility interface is given on appendix 3.

7. Conclusions

Thesis gives overview of creating an expandable wireless platform for illumination control. Different networking options were studied to find an optimal communication option. Embedded operating system was selected to provide multitasking, hardware abstraction and portability of applications. Hardware platform was created for the project, it enables extendibility, is useful for further development and provides functionality necessary for the project. Software was developed for created platform to control DMX RGB lights. PC Utility was created as proof-of-concept control application.

References

1. IEEE Standard for Local and metropolitan area networks – Part 15.4: Low-Rate Wireless Personal Area : IEEE 802.15.4-2011. New York : Institute of Electrical and Electronics Engineers, 2011
2. Asynchronous Serial Digital Data Transmission Standard for Controlling Lighting Equipment and Accessories : ANSI E1.11-2008. New York : PLASA North America, 2008.
3. ZigBee Technology [WWW] http://www.zigbee.org/About/AboutTechnology/ZigBeeTechnology.aspx (27.08.2014)
4. ZigBee specifications [WWW] http://www.zigbee.org/Specifications.aspx (27.08.2014)
5. ZigBee FAQ [WWW] http://www.zigbee.org/About/FAQ.aspx (27.08.2014)
6. ZigBee specification : ZigBee 2007. San Ramon : ZigBee Standards Organization, 2008
7. IPv6 over Low power WPAN (6lowpan) [WWW] http://datatracker.ietf.org/wg/6LowPan/charter/ (27.08.2014)
8. Transmission of IPv6 Packets over IEEE 802.15.4 Networks : RFC4944. Montenegro, et al. : Internet Engineering Task Force, 2007
9. IPv6 over Low Power WPAN Security Analysis [WWW] https://tools.ietf.org/html/draft-daniel-6lowpan-security-analysis-04 (27.08.2014)
10. Advice to the Trustees of the IETF Trust on Rights to Be Granted in IETF Documents [WWW] http://tools.ietf.org/html/rfc5377 (27.08.2014)
11. Control with WirelessHART [Online] http://www.hcf-files.com/webasyst/published/DD/2.0/file_link.php?sl=a9d3da14df068c4e93ef2af9598a39ed (27.08.2014)
12. WirelessHART Device Types – Gateways [Online] http://www.hcf-files.com/webasyst/published/DD/2.0/file_link.php?sl=ec318f885c6771a8cd72d3e62e8595ce (27.08.2014)
13. IEC 62591 WirelessHART System Engineering Guide [Online] http://www.hcf-files.com/webasyst/published/DD/2.0/file_link.php?sl=a7cef97803a279607b7da3fae7648230 (27.08.2014)

14. WirelessHART Security Overview [Online]
http://www.hcf-files.com/webasyst/published/DD/2.0/file_link.php?sl=8fc5017b478acccd8f2806675669d68c (27.08.2014)

15. Microchip MiWi PRO Wireless Networking Protocol [Online]
http://ww1.microchip.com/downloads/cn/AppNotes/01371A_CN.pdf (27.08.2014)

16. Microchip MiWi P2P Wireless Protocol [Online]
http://ww1.microchip.com/downloads/cn/AppNotes/01204A_CN.pdf (27.08.2014)

17. Microchip Application Solutions Users Agreement [Online]
http://ww1.microchip.com/downloads/en/DeviceDoc/Microchip%20Application%20Solutions%20Users%20Agreement.pdf (27.08.2014)

18. About FreeRTOS [WWW]
http://www.freertos.org/RTOS.html (27.08.2014)

19. NanoStack manual [WWW]
http://x-meter.googlecode.com/svn/trunk/nanostack/nanostack%20manual.pdf (27.08.2014)

20. Features Overview [WWW]
http://www.freertos.org/FreeRTOS_Features.html (27.08.2014)

21. TinyOS Documentation Wiki [WWW]
http://tinyos.stanford.edu/tinyos-wiki/index.php/Main_Page (27.08.2014)

22. Contiki OS website [WWW] http://www.contiki-os.org/ (27.08.2014)

23. Contiki lost platforms [WWW]
https://github.com/contiki-os/contiki/wiki/Lost-platforms (27.08.2014)

24. AVR Hardware Design Considerations (Application Note) [WWW]
http://www.atmel.com/images/atmel-2521-avr-hardware-design-considerations_application-note_avr042.pdf (27.08.2014)

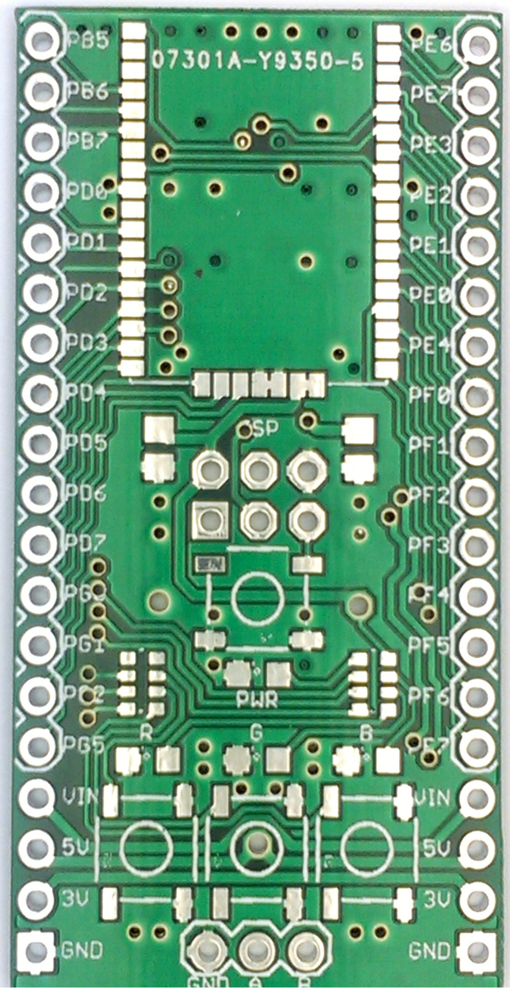
25. AVR182: Zero Cross Detector [WWW]
http://www.atmel.com/images/doc2508.pdf (27.08.2014)

26. ATmega640/1280/1281/2560/2561 Datasheet [WWW]
http://www.atmel.com/images/atmel-2549-8-bit-avr-microcontroller-atmega640-1280-1281-2560-2561_datasheet.pdf (27.08.2014)

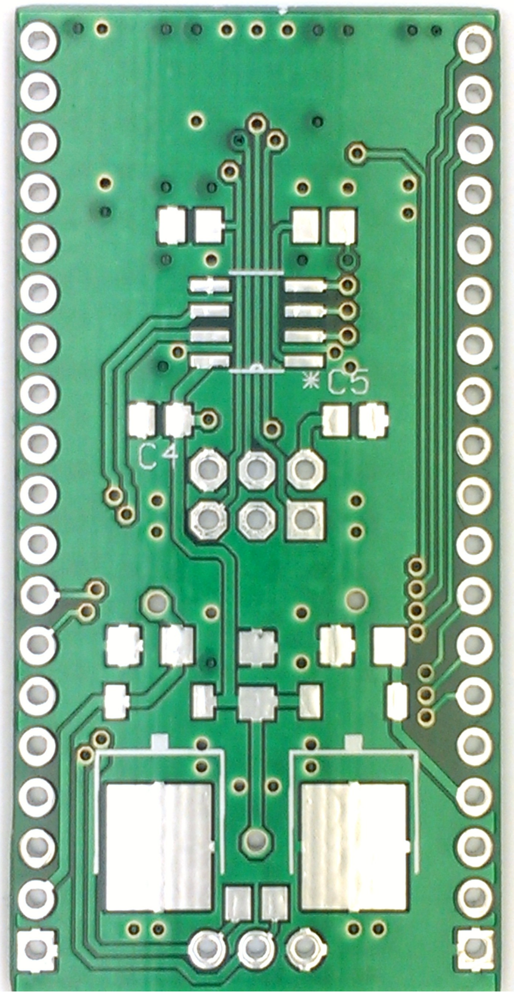
Appendix 1 – Bill of Materials

Part	Value	Device	Package	Description
U$1	-	ATMEL_ATZB_24_A2	ATZB-24-A2	**Atmel ZigBit Module**
U1	-	LT1785	SOIC8	**60V Fault Protected RS485/RS422 Transceiver**
IC1	-	BA050T	TO252	**Voltage Regulator**
IC2	-	BA033T	TO252	**Voltage Regulator**
C1	22µF	C-EUC1206	1206	**Capacitor**
C2	22µF	C-EUC1206	1206	**Capacitor**
C3	22µF	C-EUC1206	1206	**Capacitor**
C4	100nF	C-EUC0805	0805	**Capacitor**
C5	100nF	C-EUC0805	0805	**Capacitor**
C6	100nF	C-EUC0805	0805	**Capacitor**
C7	100nF	C-EUC0805	0805	**Capacitor**
SV1	-	M15	1X15	**Header 1x15**
SV3	-	M15	1X15	**Header 1x15**
SV2	-	M3X2	2X3	**Header 2x3**
RS1	-	M03PTH	1X03	**Header 3**
JP1	-	M041X04_NO_SILK	1X04	**Header 4**
JP2	-	M041X04_NO_SILK	1X04	**Header 4**
S1	-	SWCH-08247	4*4*1.5	**Momentary Switch**
S2	-	SWCH-08247	4*4*1.5	**Momentary Switch**
S3	-	SWCH-08247	4*4*1.5	**Momentary Switch**
S4	-	SWCH-08247	4*4*1.5	**Momentary Switch**
R1	3.3kΩ	R-EU_M0805	0805	**Resistor**
R2	120Ω	R-EU_M0805	0805	**Resistor**
R3	15kΩ	R-EU_M0805	0805	**Resistor**
RN1	470Ω	4R-NEXB38V	0603x4	**Array Chip Resistor**
RN2	470Ω	4R-NEXB38V	0603x4	**Array Chip Resistor**
PWR	-	19-213/GHC-XS1T1N/3T	0805	**LED**
R	-	19-213/R6C-AQ1R2B/3T	0805	**LED**
G	-	19-213/GHC-XS1T1N/3T	0805	**LED**

Appendix 2 – Unpopulated PCB



Top view



Bottom view

Appendix 3 – PC Utility main window