REMOTE CONTROL FOR DEVICES FROM INTERNAL ELECTRICAL NETWORK FOR HANDICAPPED

HAMK
UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Degree Programme in Electronic Engineering

Forssa 26.5.2012

Κωνσταντίνος Μακρίδης

Επιβλέπων καθηγητής Erasmus: Juha Sarkula

Επιβλέπων καθηγητής: Άγγελος Γιακουμής

4.11.2011 – 26.5.2012

11145ΕΣ

Forssa
Ηλεκτρονικός Μηχανικός

Συγγραφέας	Κωνσταντίνος Μακρίδης	Χρονιά 2012
Θέμα πτυχιακής	Έλεγχος Συσκευών Μέσω Της Εσωτερικής Ηλεκτρικής Εγκατάστασης Για Ανάπηρους	

ΠΕΡΙΛΗΨΗ

Ο σκοπός αυτής της πτυχιακής είναι να βοηθήσει μια μερίδα ανθρώπων με κινητικά προβλήματα στην καθημερινή τους ζωή μέσα στο σπίτι, ασφαλώς βέβαια θα βοηθηθούν και όλοι οι άλλοι καθώς δίνει μια ελευθερία κίνησης. Ο αντικειμενικός σκοπός αυτής της πτυχιακής είναι ο ασύρματος έλεγχος των συσκευών όπου είναι συνδεδεμένα με το εσωτερικό ηλεκτρικό δίκτυο ενός σπιτιού / μαγαζιού / εργοστασίου και ειδικότερα τον έλεγχο ανοίγματος /κλεισίματος των πριζών, των διακοπτών φωτός αλλά και τον άλλων ειδικών συσκευών που υπάρχουν σε ένα σπίτι.

Το εν λόγο εγχείρημα αυτής της πτυχιακής μπορεί να χωριστεί σε τρία ανεξάρτητα κυκλώματα όπου το καθένα έχει διαφορετικό ρόλο στο σύστημα. Το πρώτο κύκλωμα είναι ένα μέρος ασύρματου έλεγχου, ένα τηλεχειριστήριο. Θα δούμε σε θεωρητικό επίπεδο πως μπορούμε να στείλουμε ασύρματα της εντολές και την διαχείριση των συσκευών.

Το δεύτερο κύκλωμα είναι το πιο σημαντικό-κρίσιμο όλου του συστήματος επειδή είναι ο κύριος ελεγκτής των συσκευών. Εναπόκειται σε αυτό το κύκλωμα όλοι οι διακόπτες φωτός, οι πρίζες αλλά και οι άλλες ειδικές συσκευές να δουλεύουν εναρμονισμένα και να ακολουθούν σωστά τις εντολές για να είναι το σύστημα επιτυχημένο. Η λειτουργία του κυκλώματος είναι να δέχεται ασύρματα δεδομένα και κατόπιν επεξεργασίας να τα στείλει μέσα από το ηλεκτρικό δίκτυο.

Το τρίτο κύκλωμα μπορεί να χωριστεί σε τέσσερα ανεξάρτητα κυκλώματα με διαφορετικά χαρακτηριστικά. Υπάρχουν οι διακόπτες φωτός, οι πρίζες, οι ροοστάτες και άλλα ειδικά κυκλώματα. Όλα αυτά δέχονται τις εντολές μέσα από το εσωτερικό ηλεκτρικό δίκτυο (Ηλεκτρικό δίκτυο 110/220V) χρησιμοποιώντας ένα τηλεχειριστήριο για τον έλεγχο και την διαχείριση τους.

Όταν εξετάσουμε όλες τις πληροφορίες και της λεπτομέρειες τότε θα έχουμε μια ξεκάθαρη εικόνα στο μυαλό μας πως το εν λόγο σύστημα (σκοπός της πτυχιακής) μπορεί να βοηθήσει τον καθένα μας στο να κερδίσουμε χρόνο αλλά και στην αίσθηση της ελευθερίας μέσα στο σπίτι / μαγαζί / εργοστάσιο όπου ζούμε και εργαζόμαστε.

HAMK
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Forssa
Electronic Engineering

Author Konstantinos Makridis **Year** 2012

Subject of Bachelor's thesis Remote control for devices from internal electrical network for handicapped.

ABSTRACT

The purpose of this thesis is to help a special group of humans with handicap problems in day-to-day life inside the house, naturally it offer all the others because he give a free movement. The object of this thesis is to control by a remote control the devices that are connected to internal electrical network of a house / shop / factory and command to switch on/off lights, sockets and special devices inside a house.

The whole project can split into three independent circuits that have a different role in system. The first circuit is a part from remote controls and called telecontrol. We will see both theoretical and as an application of the theory how we can send wireless commands and manipulate the devices pressing buttons.

The second circuit is the most crucial part of the project because is the master devices controller. It is up to this circuit all light switches, sockets and special devices work smoothly and follow commands for a correct system. The feature of this circuit is to receive the wireless data and after processing sent it by the power supply.

The third circuit can split into four independent circuits with different circuits and different feature. We have light switch, socket, dimmer and special function circuits. All of them receiving the commands from the internal electrical network (Powers supply 110/220V) by a remote control and execute it.

When we examine all the information's and details then we will have a clear picture in our minds how this project can help each person saving time and release our hands inside our home / shop / factory that we live and work there.

Keywords Remote control, Power line modem, Microcontroller, Protocol, internal electric network

Pages 112 p. + appendices 6 p.

CONTENTS

1 INTRODUCTION

1.1 Problem description

Life of humans today is more fragile and more accident-prone. Many humans in our days they are in awkward predicament that cannot move as before, the result of trying have a normal life as before inside their own house make them be tired. Humans with difficulty in movement, paralyze humans or people of the third age all of them facing the same problem how to move inside their house with easement, for example those task: switch on/off the lights - open/close electrical heater, radiator, alarm and more other devices will bring relief to handicap people.

1.2 Problem solution

The purpose of science is to grant us more knowledge/information about our world, ourselves and how to improve our quality of life. As electronic engineer that is piece of science is my duty to bring to those people a small relief in their day-to-day battle for life. This small relief can come from my project I want to present you from now on. My project subject is to control devices (lights, electrical heater, alarm and more others) inside a house using as data line the electrical network (without new wire installation) by a remote control.

1.3 Motivation

When I was in Alexander Technological Educational Institution of Thessaloniki he came to my ear the idea to sent data over electrical network (power supply) I like it and want to learn more about it, how it work. After a search for that case I start thinking where I can use it, the solution came to mind very quick, of course for automation inside the house. I make thoughts about handicap people do they have any device that can make their life easier; The answer was that don't have anything like this but also most house and people haven't any cheap, easy to install home automation device to control all the house (It is estimated that X10 compatible products can be found in over 10 million homes, but not recommended for home automation applications). For that I want to build a cheap home automation system for the majority of people that will be also a relief-relaxation for most of people.

1.3 The Desire

My desire for the start is to build the base were the system work normally. Because this project is a prototype and not a complete version my desire after I have made the base is to update it in full scale of feature that have as idea for my project. And then prepare it for commercial use and then see it in shops.

2 HOME AUTOMATION

2.1 Terminology

Home automation is the residential extension of "building automation". It is automation of the home, housework or household activity. Home automation include control over all the systems we use every day and those is lighting, High Voltage AC including heating, ventilation and air conditioning, appliances, and other systems, to provide for people improved convenience, comfort, energy efficiency and security. Home automation for disabled people can provide increased quality of life and independence that raises self confidence.

Devices share the same channel (wired electrical power 110/220V or wireless) and can communicate each other, also may be connected through a computer network to allow control by a personal computer and for a new version can be controlled by Smartphone's, and last may allow remote access from the internet using a web server.

Controlling systems for a home automation is:

- Lighting
- Security systems & access control
- Home theater & entertainment
- Phone systems
- Heating (High Voltage AC)
- Irrigation
- Power sockets
- Cooling (High Voltage AC)

2.2 Known used home automation protocols

The most popular home automation protocols that running in markets is five protocols: X10, UPB, INSTEON, Z-Wave, ZigBee and from all five the most known is the X10 protocol. Each protocol has advantage and disadvantage and also a different way how it work.

2.2.1 X10

X10 protocol primarily uses power line wiring for signaling and control, where the signals involve brief radio frequency bursts representing digital information. A wireless radio based protocol transport is also defined.

X10 remains popular in the home environment with millions of units in use worldwide and inexpensive availability of new components.

To configure the system for the users it use four bit for house code as a letter from A through P while the four bit unit code is a number 1 through 16. When the system is installed, each controlled device is configured to respond to one of the 256 possible addresses (16 house codes × 16 unit codes); each device reacts to commands specifically addressed to it, or possibly to several broadcast commands.

In the 60 Hz AC current flow, a bit value of one is represented by a 1 millisecond burst of 120 kHz at the zero crossing point, immediately followed by the absence of a pulse. A zero value is represented by the absence of 120 kHz at the zero crossing point (pulse), immediately followed by the presence of a pulse. All messages are sent twice to reduce false signaling, data rates are around 20 bit/s.

Advantage of X10:

- Inexpensive
- No new wiring is required
- Simple to install
- 100's of compatible products
- Time proven -- it has been around for over 30 years

Disadvantage of X10:

- X10 communications are slow
- Collision problem, one messages every time
- Load-related unreliable X10 communications, reliability decreases and increases by load
- Attenuation of X10 signals because of EMI filters in electronic appliances
- X10 signals can be attenuated by Earth Leakage Detectors to the point where X10 communications become unreliable

For more details and knowledge about X10 protocol see: “What is X10 home automation;” and “X10 (industry standard)” from sources.

2.2.2 UPB

Universal powerline bus or UPB is a protocol that uses power line wiring for signaling and control. Based on the concept of the ubiquitous X10 standard, UPB has an improved transmission rate and higher reliability.

The UPB communication method consists of a series of precisely timed electrical pulses (called UPB Pulses). UPB Pulses are generated by charging a capacitor to a high voltage and then discharging that capacitor's voltage into the powerline at a precise time. This quick discharging of the capacitor creates a large "spike" (or pulse) on the powerline that is easily detectable.

While transmitting, one UPB Pulse is generated each half-cycle in one of four predefined positions in the half-cycle of the AC powerline. The position of each UPB Pulse determines its value as 0, 1, 2, or 3. This method of encoding is a well-known as Pulse-position modulation (PPM) in digital communications. Since each UPB Pulse can encode two bits of digital information and there are 120 AC half-cycles per second (at 60Hz), UPB communication has a raw speed of 240 bits per second.

Advantage of UPB:

- Highly Reliable, UPB 99.9% - X10 70%~80%
- No New Wires
- Two Way Communications, Hardware, software and protocol design allow for two-way communication in all products
- House Separation, UPB system incorporates over 64,000 total address space vs 256 for conventional X-10.
- Peer to Peer, No central controller necessary for single point-to-point control or group control
- Simplicity, UPB solution uses "off the shelf" components for transmission, receiving and control circuits, including standard microprocessors.

Disadvantage of UPB:

- High initial cost, Cheaper to operate in the long run, expensive to install
- Powerline issues, Susceptible to problems related to all forms of communication on a shared powerline
- Other minor issues, Shortcomings include product compatibility

For more details and knowledge about UPB protocol see: "UPB Technology Description" from sources.

2.2.3 INSTEON

INSTEON is a dual-band mesh home area networking topology employing AC-power lines and a radio-frequency (RF) protocol to communicate with devices; this is intended to improve reliability.

Automatic error detection and correction are included in all INSTEON compatible products. The power line protocol uses phase-shift keying and is designed so that the repetition is synchronized: All repeaters repeat the same message during precisely-defined time slots. The power line AC frequency is used as the synchronization source.

All INSTEON devices are peers, meaning each device can transmit, receive, and repeat any message of the INSTEON protocol. As a peer-to-peer network, devices do not require network supervision, thus dispensing with the need for controllers and routing tables. Data rates are Instantaneous 13,165 bit/s and Sustained 2,880 bit/s.

Advantage of INSTEON:

- Reliability, Both communication methods: power lines and Radio Frequency (RF), which means it transmits messages faster
- Affordable, It is relatively inexpensive to install this system
- X10 backwards compatible, Most Insteon devices can respond to X10 commands
- Single set-up, Set-up is achieved using a web-based interface, via Internet Explorer
- Peer networks, Every device acts as a repeater, every added node strengthens the home automation network

Disadvantage of INSTEON:

- High volume issues, Same bandwidth limitations as X10,UPB and Z-Wave all face
- PRICE, 4-6 times cost more than X10

For more details and knowledge about INSTEON protocol see: “Insteon the details” from sources.

2.2.4 ZigBee

ZigBee is a high level communication protocols using small, low-power digital radios based on an IEEE 802.15.4 standard for personal area networks. ZigBee is targeted at radio-frequency (RF) applications that require a low data rate, long battery life, and secure networking.

ZigBee network layer natively supports both star, tree and mesh networks. Every network must have one coordinator device, tasked with its creation, the control of its parameters and basic maintenance. Within star networks, the coordinator must be the central node. Both trees and meshes allow the use of ZigBee routers to extend communication at the network level. ZigBee is not intended to support powerline networking but to interface with it at least for smart metering and smart appliance purposes.

ZigBee it use Offset quadrature phase-shift keying (OQPSK) that transmits two bits per symbol, in the 2.4 GHz band there are 16 ZigBee channels, with each channel requiring 5 MHz of bandwidth. The 2.4 GHz band provides up to 250 kbit/s but data transmission rates vary from 20 to 900 kilobits/second. Transmission range is between 10 and 75 meters and up to 1500 meters for ZigBee Pro. The output power of the radios is generally 0 dBm (1 mW). ZigBee uses 128-bit keys to implement its security mechanisms.

Advantage of ZigBee:

- Power saving
- Reliability, Collision avoidance is adopted
- Low cost of the modules and ZigBee protocol patent fee free
- Short time delay, Typically 30 ms for device searching, 15 ms for standby to activation, and 15 ms for channel access of active devices
- Large network capacity, One ZigBee network contains one master device and maximum 254 slave devices.
- Safety, AES-128 is adopted

Disadvantage of ZigBee:

- Short range
- Low data speed

For more details and knowledge about ZigBee protocol see: “ZigBee” and “ZigBee Alliance” from sources.

2.2.4 Z-Wave

Z-Wave is a wireless communications protocol designed for home automation using a low-power RF radio. Z-Wave wireless protocol is optimized for reliable, low-latency communication of small data packets. Z-Wave operates in the sub-gigahertz frequency range, around 900 MHz the modulation is GFSK the range approximately 30 meters and data rate is 9,600 bit/s or 40 kbit/s, fully interoperable.

Z-Wave uses a source-routed mesh network topology and has one or more master controllers that control routing and security. A controllable device must be "included" to the Z-Wave network before it can be controlled via Z-Wave.

Advantage of Z-Wave:

- Reliability, Z-Wave is extremely reliable and well supported due to its wireless mesh networking system
- Two-way communication
- Limited interference, More devices installed in the network, the stronger the signal strength becomes
- Wide range, The 900 MHz range allows signals to be stronger and transmit through longer distances with more power
- Complex commands, Z-Wave products are wide-ranging and diverse

Disadvantage of Z-Wave:

- Cost, Z-Wave products are more expensive
- Network controller, At least one controller for a Z-system
- Proprietary, For own custom software programming need to be sign a Non-Disclosure Agreement (NDA) and buy a developers' kit

For more details and knowledge about Z-Wave protocol see: "Z-Wave" and "Z-Wave Protocol Overview "from sources.

3 O.P.R.A PROTOCOL

3.1 Protocol structure ‘O.P.R.A’ **O**ne_ **P**rotocol_to_**R**ule_them_**A**ll

A new protocol for home automation starts from here exploring all the prospects. Home automation it uses many kind of protocols from different companies and each protocol have positives and negatives. O.P.R.A protocol is not the best among the other but will do the difference from other angle of view. The main subject in O.P.R.A protocol is to control whole house or other place from one device that act as the master, the name of that device is Master Device Controller or MDC. One way to control or command the MDC is with remote control. One category from remote controls is “Telecontrol”, a device where user can send commands, check settings and watch the status of whole house. The whole theory, how O.P.R.A protocol for home automation works is been describe from this point and over.

O.P.R.A devices communicate with MDC by sending and receiving message particular length and structure. They are two types of message, a simple one and an extended. Each message composed by: **From:** address, **To:** address, **command 1**, **command 2** and **message status**. Extended message have the previous data plus the **user data map 1**, **user data map 2** and **Extended data 1-16** (variable length for each extended data, depending from what kind of function have) both type of message have integrity byte.

In O.P.R.A protocol will see:

- O.P.R.A message structure:
    - Message length
    - Message field
- O.P.R.A message repetition:
    - Message hopping
    - Message retrying
- O.P.R.A message summary
- O.P.R.A commands
- O.P.R.A device category
- O.P.R.A device linking
- O.P.R.A route map
- Data Table
- Signaling Details

3.1.1 O.P.R.A Message Structure

O.P.R.A devices communicate each other and with MDC by sending a fixed-length simple message. When extended messages is needed to be sent a simple-fixed-length message together with a variable-length-extended data 1-16 plus the two bytes from user data map is forming the message. O.P.R.A protocol starts from understanding the **message length** from both message type and the contents of **message field** within the message.

Message length

O.P.R.A protocol have two different kind of message, the first have 13 byte of data as the standard message, the second have as base the standard message (13 bytes) plus a variable data of 56 bytes total 71 bytes maximum message length and this is the extended message.

Table 1 Standard message structure

Standard Message 13 bytes				
4 bytes	4 bytes	1 bytes	2 bytes	2 bytes
FROM**: Address**	**TO**: Address	**Message Status**	**Command 1,2**	**CRC**

Table 2 Extended message structure

Extended Message 71 bytes max						
4 bytes	4 bytes	1 bytes	2 bytes	2 bytes	56 bytes	2 bytes
FROM**: Address**	**TO**: Address	**Message Status**	**Command 1,2**	**User Data Map 1,2**	**Data**	**CRC**

Table 3 Standard message details

<table>
<tr><th colspan="2">DATA</th><th>BITS</th><th>BYTES</th><th>CONTENTS</th></tr>
<tr><td colspan="2">FROM: Address</td><td>32</td><td>4</td><td>Message originator address</td></tr>
<tr><td colspan="2">TO: Address</td><td>32</td><td>4</td><td>Target device address</td></tr>
<tr><td rowspan="4">Message Status</td><td>Transmit type</td><td>3</td><td rowspan="4">1</td><td>Direct, Broadcast, Acknowledge, Rout map, Group, House region</td></tr>
<tr><td>Hop number</td><td>2</td><td>Max number of devices to hand the message</td></tr>
<tr><td>Retransmission number</td><td>2</td><td>Max retransmit number for message</td></tr>
<tr><td>Set Extended message</td><td>1</td><td>‘0’ for standard message</td></tr>
<tr><td colspan="2">Command 1</td><td>8</td><td>1</td><td rowspan="2">Executable command</td></tr>
<tr><td colspan="2">Command 2</td><td>8</td><td>1</td></tr>
<tr><td colspan="2">CRC</td><td>16</td><td>2</td><td>Cyclic Redundancy Check</td></tr>
</table>

Standard message have been design for direct transmit type as fast can become, payload here is only 2 bytes, command 1and 2.

Extended message have been design for transmitting data from user profile of “Telecontrol” remote control or other remote control to MDC and MDC transmit it in each device.

Route map process (connect dead link devices to turn in online) and routing message process (deliver message to a distant device that is connected with chain link device) using both extended message to deliver the packets.

Table 4 Extended message details

DATA		BITS	BYTES	CONTENTS
FROM: Address		32	4	Message originator address
TO: Address		32	4	Target device address
Message Status	Transmit type	3	1	Direct, Broadcast, Acknowledge, Rout map, Group, House region
	Hop number	2		Max number of devices to hand the message
	Retransmission number	2		Max retransmit number for message
	Set Extended message	1		‘0’ for standard message
Command 1		8	1	Executable command
Command 2		8	1	
User data map 1 (UDM 1)		8	1	Extended data content 1-16
User data map 2 (UDM 2)		8	1	
Extended Data 1		8	1	DBDN 1, 2, 3 (in/out) choose physical channel
Extended Data 2		8	1	Route table number
Extended Data 3		64	8	1, 2 Hop: Intermediate N.1address and X address
Extended Data 4		32	4	3 Hop: Intermediate N.3address
Extended Data 5		32	4	Final device address
Extended Data 6		8	1	Control state and status state of device
Extended Data 7		8	1	Category type
Extended Data 8		8	1	House region and group number
Extended Data 9		8	1	Group day period and device day period
Extended Data 10		16	2	Device post on
Extended Data 11		16	2	Device post off
Extended Data 12		16	2	Group post on
Extended Data 13		16	2	Group post off
Extended Data 14		96	12	Group name
Extended Data 15		16	2	Device name
Extended Data 16		96	12	nickname
CRC		16	2	Cyclic Redundancy Check

Message field

Device address

The first 2 fields in O.P.R.A protocol is source and destination address (**FROM - source**, **TO - destination**). Each address is a unique ID from 2^{32} = 4.292.967.296 (32bits - 4bytes) devices that are possible to exist. All devices have that unique ID written in nonvolatile memory from manufactory (flash memory) that can't be lost or erase it. The first field (**FROM** address) carrying always device address, **who sent** the message and second field (**TO** address) carrying always the destination, **who receive** the message, about the second field when many devices must listen the same message at the same time a broadcast message can be sent out without to need any destination address. Most messages are direct type which has always source and destination address. When someone use house region broadcast, group broadcast and broadcast there is no need for destination address.

Message status

The third field contains the transmit type, hop number, retransmission number and finally message data, 13bytes standard message or 71 bytes max extended message.

Table5 Message status details

Bit position	Status	Meaning
Bit 7 (MSB) /Broadcast	Transmit type	000= Direct message 001= ASK of Direct message 010= Routing + Direct message 011= Routing +ASK of direct message 100= Route Map 101= House Region Broadcast 110= Group Broadcast 111= Broadcast
Bit 6 /Routing + Direct - ASK message		
Bit 5 /Direct – ASK message		
Bit 4	Hope number	00=direct message to final device 01=1max connection for routing 10=2max connection for routing 11=3max connection for routing
Bit 3		
Bit 2	Retransmission number	00= Final Device here 01=1 device remaining 10=2 devices remaining 11=3 devices remaining
Bit 1		
Bit 0 (LSB)	Extended	0= Standard message 1= Extended message

Transmit type

- **Broadcast**: message without a specific destination, all device receive the message and execute the command. In broadcast transmission type MDC don’t receive any acknowledge from device. This type of transmission is used for specific work or for emergency action.
- **Group broadcast**: message without any specific destination, contrary the destination is a number of devices that create a group. MDC know the name and the number of that group and link a number of devices. There is not an acknowledging for group broadcast also. This type of transmission is ideal for people who want a deep configuration system for many cases, for example someone want all garden lights open at 8:00PM and close at 6:00AM, all the lights in garden can be in one group with group name: garden light [12 characters max] and link all the light devices that exist in garden work with schedule. The maximum number of groups that user can create is 15 and is very decent for an average user, also there is house region group.
- **House region broadcast**: message without any specific destination, contrary the destination here is a specific area inside a house (for industrial sector MDC is more suitable for factory). There is 16 places that a device can be part of a region, 9 is by default fill from manufactory and 7 spots left from the user to add more rooms and call them as user wish.
- **Route map**: this type of message is used only when a device is out of range from MDC and ‘link status’ is ‘offline’ then a route map procedure can began to find alternative way for making ‘link status’ of that device ‘online’. Maximum number of intermediate device that connect final device is three. When all devices is ‘link status’ ‘online’ then a route table have been created with their address and route table number that link all the device together as a chain. Maximum number of routed device is 255 in number.
- **Routing + direct +ASK message:** when MDC want to send a message in one device that is out of MDC range the message field have intermediate devices address that connect each other to reach final device. Maximum connected device to reach final device is three, this type of message have acknowledge from final device to MDC.
- **Direct + ASK message**: this type of message is the most common among all the other types. Connect direct MDC with the final destination device, that mean the connection is physical because the share the same network inside of MDC range, also that message have acknowledge from final device to MDC.

Hope number and retransmission number

Hope number bit inform if the message that following is for direct transmission (physical connection) or from indirect message (logical connection).
When hop number inform that connection is indirect (logical connection) then retransmission number inform how many device left until to reach final device (Maximum 3, next message decrease one, 2 and so on).

- **Bit4**= 0 **Bit3**= 0: That mean the message for delivery it point directly final device address, also **Bit2**= 0 and **Bit1**= 0 inform that only final device can be in destination address. When MDC didn’t receive any ASK from final device then retransmitting message again, if 3 times fail to receive any ASK message from final device MDC turn final device address as a link dead and route map procedure can start to find and turn that device ‘link status’ ‘online’ again.
- **Bit4, Bit3**= 01,10,11: that mean the message for delivery point to an indirect/ logical connected device that need to know all intermediate devices address to reach final device. **Bit4**, **Bit3** inform the number of connected device to reach final device, the maximum is three intermediate devices. The message load with intermediate address and load at **Bit2, Bit1** the maximum number of connected device that will take part. For example it need to be reach a final device and message must travel among three intermediate device **Bit4**= 1 **Bit3**=1, **Bit2**= 1 **Bit1**= 1 load intermediate 1, 2, 3 address and final device address. When the message move to the first device that device before sent the message to the next device decrease retransmission number to two make **Bit2**= 1 **Bit1**= 0 and then sent it to the next, that will occur until reach final device and **Bit2**= 0 **Bit1**= 0.

Extended message

Bit0= 0: that bit inform that the message is standard and only 13 bytes need to send.

Bit0= 1: that bit inform that the message is extended and maximum bytes for transfer is 71.

Command 1, 2

Command 1 is the main command that describe the job for execution and command 2 is the subcommand carrying details for that job. Command 1 have 254 different jobs for execution and each subcommand can take 254 parameters. Command 1 is split in different category’s that include each type of device and command for near future. For example a number of commands are used for light switch device, for direct message, for route map, for group broadcast and more.

User data map 1, 2

Both bytes make a single extended data table that reveal which extended data have been used in message and make device understand the data because they are not 8bit all, some have more than 8bit. Each bit inform if the message carrying those ED 1-16 (ED= Extended Data) , default state is 0 who reveal that EDx is off when EDx is on then bit have logic ‘1’.

Table 6 User Data Map 1 table

USER DATA MAP 1							
ED8	**ED7**	**ED6**	**ED5**	**ED4**	**ED3**	**ED2**	**ED1(LSB)**
House region and group number	Category type	Control state and status state of device	FD address	3 Hop	1, 2 Hop	Route table number	DBDN 1, 2, 3 (in/out) physical channel

Table 7 User Data Map 2 table

USER DATA MAP 2							
ED9	**ED10**	**ED11**	**ED12**	**ED13**	**ED14**	**ED15**	**ED16(MSB)**
Group day period and device day period	Device post on	Device post off	Group post on	Group post off	Group name	Device name	Nickname

Table 8 Extended data 1

Extended Data 1=DBDN							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
FD IN	FD OUT	DBDN3 IN	DBDN3 OUT	DBDN2 IN	DBDN2 OUT	DBDN1 IN	DBDN1 OUT

ED 1: this data inform each intermediate device which connection method to choose for transmitting the message. DBDN(dual band device network) have maximum three intermediate device that can inform a dual band device which way to choose for message delivery. This data is been used when dual band device exist for message delivey, DBDN 1, 2,3 have all ‘in’ and ‘out’. When bit have ‘0’ mean use wireless network for message delivery, when bit is ‘1’ mean use wired network (power line network 110/220 V). Each dual band device can receive/transmit wireless or wired, when it need to be sent message to the next device we know that device how to receive the message (wired/wireless) and know how to trasmit (wired/wireless). For example when final device receive message with Bit1= 0(IN wireless), Bit0= 1(OUT wired) that mean when final device must respond with ASK message final device change the Bit1= 1(IN wired), Bit0= 0(OUT wireless) and sent the message.

Table 9 Extended data 2

Extended Data 2=Route Table Number							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
X	X	X	X	X	X	X	X

ED2: this data carrying the number of logical connected final device, MDC have whole route table and knowing for each route table number final device address and intermediate devices address. Each number from 1-255 of route table number can connect 1 final device with 3 maximum intermediate devices.

Table 10 Extended data 3

Extended Data 3=HOP 1, 2							
BIT63	**BIT-**	**BIT-**	**BIT32**	**BIT31**	**BIT-**	**BIT-**	**BIT0(LSB)**
Final device address(hop=1)or Intermediate 2 address(hop=2)				Intermediate 1 address			

ED3: this data hold for **hop=1** two address, the first 32 bit have the intermediate 1 address and the other 32 bit of total 64 bit have final device address. When **hop=2** then the first 32 bit hold intermediate 1 address and the other 32 bit hold intermediate 2 address.

Table 11 Extended data 4

Extended Data 4=HOP 3							
BIT31	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT0(LSB)**
Intermediate 3 address							

ED4: this data hold intermediate 3 address when **hop=3**.

Table 12 Extended data 5

Extended Data 5=FD address							
BIT31	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT0(LSB)**
Final Device address							

ED5: this data contain final device address when **hop=2** and **hope=3**.

Table 13 Extended data 6

Extended Data 6= Control state and status state of device							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
TRANSPORT STATE		REGION CODE	GROUP STATE ON/OFF	GROUP SCHEDULE ON/OFF	GROUP DEFAULT STATE	DEVICE SCHEDULE ON/OFF	DEVICE DEFAULT STATE
DEVICE→MDC			MDC← →DEVICE				

ED6: this data contain two groups the first group have to do with read only data (DEVICE sent to MDC) and the second group have to do with read/write data (MDC read/write to DEVICE). Read only data is the bit 5 to 7. Region code inform if the **Bit5= 0** that work for 220V 50Hz and if **Bit5= 1**mean that work for 110V 60Hz. Transport state inform if **Bit7= 0, Bit 6= 1**that device work upon power line network (110/220V) if **Bit7= 1, Bit 6= 0** that device work upon wireless network and if **Bit7= 1, Bit 6= 1** device is dual band (wired/wireless).

Read/write data is the bit 0 to 4. If **Bit0= 0** device default state = normal close N.C, if **Bit0= 1** device default state = normal open N.O. If **bit1= 0** device schedule = off, If **bit1= 1** device schedule = on. If **bit2= 0** group default state = normal close N.C, If **bit2= 1** group default state = normal open N.O. If **bit3= 0** group schedule on/off = off, If **bit3= 1** group schedule on/off = on. If **bit4= 0** group state on/off = off, If **bit4= 1** group state on/off = on.

Table 14 Extended data 7

Extended Data 7=Category type							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
6 Main categories, [128, 64, 32, 16, 8, 4] subcategories							

ED7: this data is only readable from each device nonvolatile memory. They are 6 main categories and each of the main categories divide in subcategories. Main categories are socket, dimmers, light switch, special function, house protection, home entertainment. Some sub categories from main categories is [socket].1, 2, 3, 4, 6, 8 position, [light switch].1, 2, 3, 4, 6 switch.

Table 15 Extended data 8

Extended Data 8= House region and group number							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
Group number				House region number			

ED8: this data carrying number of region and group for group and house region broadcast message. User can create up to 15 different group configuration with multiple devices address as memory allows it. Also house region number allow up to 16 different regions inside a house, 7place free for user to configure.

Table 16 Extended data 9

Extended Data 9= Group day period and device day period							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
G.D.P Group day period				D.D.P Device day period			

ED9: this data contain for device and group schedule the day period that device turn on or off.

Table 17 Day period table

Bit code 0000 to 1111 0 to16	**Description**
0000 – 0	All days
0001 – 1	Monday - Friday
0010 – 2	Saturday- Sunday
0011 – 3	Monday
0100 – 4	Tuesday
0101 – 5	Wednesday
0110 – 6	Thursday
0111 – 7	Friday
1000 – 8	Saturday
1001 – 9	Sunday
1010 – 1111 11-16	[empty]

Table 18 Extended data 10

Extended Data 10= Device post on							
BIT31	**BIT-**	**BIT-**	**BIT16**	**BIT15**	**BIT-**	**BIT-**	**BIT0(LSB)**
Post(power on start time) hour **hh**:mm				Post(power on start time) minute hh:**mm**			

ED10: this data carrying power on start time for device, 16 bit for minute and 16 bit for hour.

Table 19 Extended data 11

Extended Data 11= Device post off							
BIT31	**BIT-**	**BIT-**	**BIT16**	**BIT15**	**BIT-**	**BIT-**	**BIT0(LSB)**
Post(power off stop time) hour **hh**:mm				Post(power off stop time) minute hh:**mm**			

ED11: this data carrying power off stop time for device, 16 bit for minute and 16 bit for hour.

Table 20 Extended data 12

Extended Data 12= Group post on							
BIT31	**BIT-**	**BIT-**	**BIT16**	**BIT15**	**BIT-**	**BIT-**	**BIT0(LSB)**
Post(power on start time) hour **hh**:mm				Post(power on start time) minute hh:**mm**			

ED12: this data carrying power on start time for group, 16 bit for minute and 16 bit for hour.

Table 21 Extended data 13

Extended Data 13= Group post off							
BIT31	**BIT-**	**BIT-**	**BIT16**	**BIT15**	**BIT-**	**BIT-**	**BIT0(LSB)**
Post(power off stop time) hour **hh**:mm				Post(power off stop time) minute hh:**mm**			

ED13: this data carrying power off stop time for group, 16 bit for minute and 16 bit for hour.

Table 22 Extended data 14

Extended Data 14= Group name							
BIT95	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT0(LSB)**
12 bytes of string							

ED14: this data contain 12 bytes string type for group name.

Table 23 Extended data 15

Extended Data 15= Device name							
BIT15	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT0(LSB)**
2 bytes of string							

ED15: this data is only for reading from device that enumerate MDC the name of this devices from the first letter of main category, for example socket name is = SO, for light switch name is = LS and so on.

Table 24 Extended data 16

Extended Data 16= Nickname							
BIT95	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT-**	**BIT0(LSB)**
12 bytes of string							

ED16: this data hold nickname that user gave for specific device, 12 bytes of string maximum.

CRC

Cyclic redundancy check is a 16 bit data that is the result of a polynomial algorithm showing if data have been corrupted or not after transmission. Cover standard and extended version of message (13bytes, 71bytes). If the message is corrupt, the receiver CRC will not match with the transmitted CRC and command for retransmission will need.

3.1.2 O.P.R.A Message Repetition

O.P.R.A protocol has two ways to retransmit a message that fail to reach final destination. The first way is for direct physical connected devices and the second for indirect logical connected device. Message retrying is for direct connected device and message hopping for indirect connected device.

Message hopping

Message hopping is a chain line from devices that each device leads to the final destination, because final device is out of MDC range. This type of message can connect up to three devices to connect the final device with MDC. Each message use some extended data to support the communication between MDC and final device.

Routing message structure for Socket device, 3 Hop, command: ”Power On”

From address:
To address:
Command 1, 2: (From command depend what data will be add for transmission)

Message status:
User data 1, 2:
ED1 **if** device support and use dual band (wired/wireless)
ED2 route table number
ED3 hop 1, 2
ED4 hop 3 **if** hop=3
ED5 final device address **if** hop>=2

Table 25 Extended routing message for Socket device

DATA		Variable	BYTES	CONTENTS
FROM: Address		[ffff,ffff]	4	MDC address
TO: Address		[ffff,ffff]	4	Intermediate 1 address
Message Status	Transmit type	010	1	Routing +direct message
	Hop number	11		3 hop - 3devices
	Retransmission number	11		Start price from MDC=3
	Set Extended message	1		extended message
Command 1		[ff]	1	"Power on"
Command 2		[empty]	1	
User data map 1 (UDM 1)		[00011111]	1	Extended data content 1-5 on
User data map 2 (UDM 2)		0	1	
Extended Data 1		[11,01,10,11]	1	DBDN 1, 2, 3 (in/out) and FD(in/out)
Extended Data 2		[ff]	1	Route table number
Extended Data 3		[ffff,ffff], [ffff,ffff]	8	1, 2 Hop: Intermediate N.1address and X address
Extended Data 4		[ffff,ffff]	4	3 Hop: Intermediate N.3address
Extended Data 5		[ffff,ffff]	4	Final device address
CRC		[ffff]	2	Cyclic Redundancy Check

In this example a far distant final device that is connected with other three devices wants to talk with MDC. Command is "Power On" to a socket device, is been used the routing + direct message structure to reach the final device and execute the command. Intermediate 1 is single band (wired) connection, intermediate 2 is dual band connection, intermediate 3 is dual band connection also and final device is single band connection (wired). MDC use extended message to send all the extended data to intermediates devices. Total bytes for routing + direct message are 33bytes.

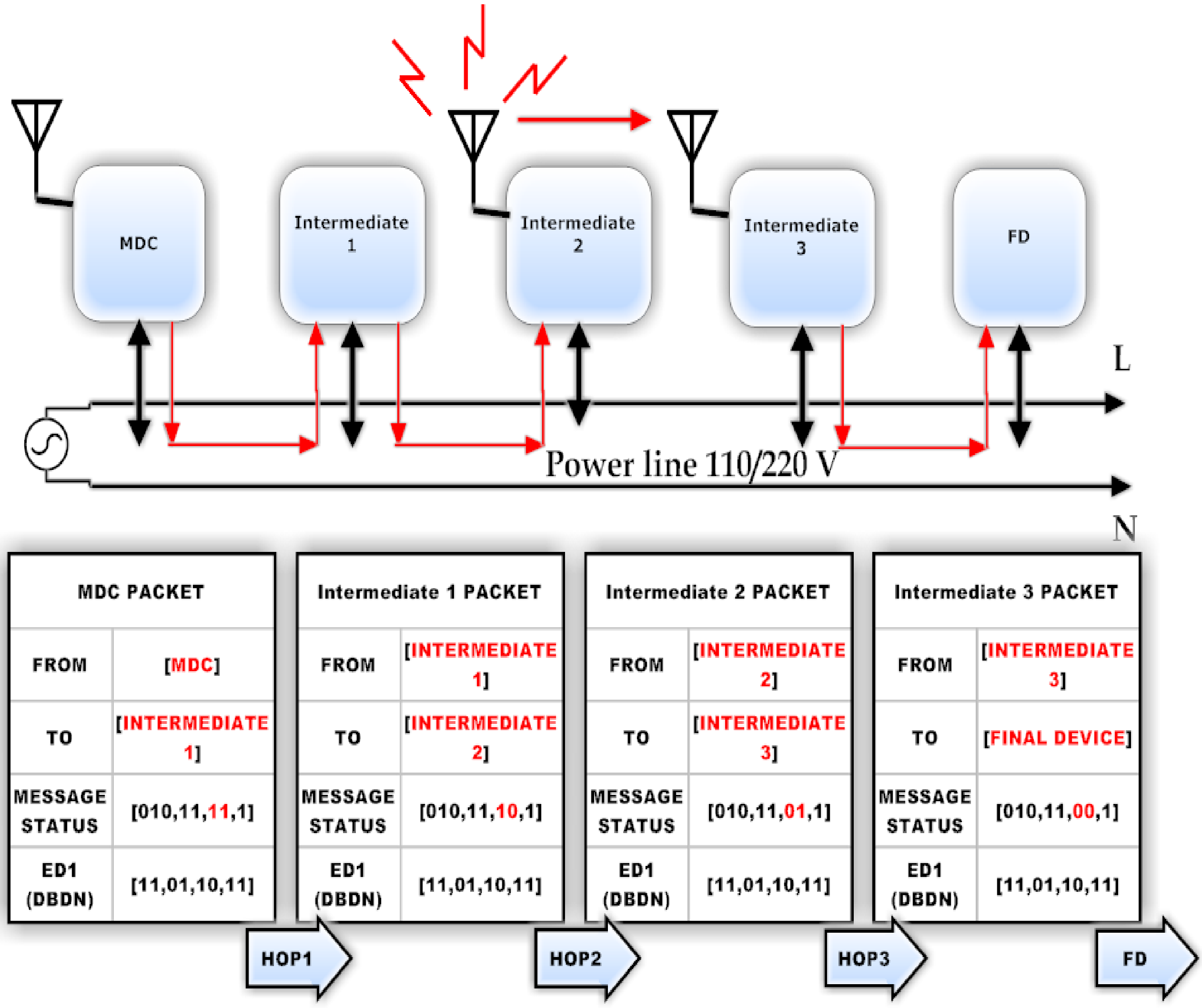


Figure 1 Routing + direct message procedure with 3 Hops

The message that MDC sent has all the information about whole routing + direct process. MDC data message have four addresses for each device that message will pass, also have data that inform device which physical channel to choose for transporting the message to the final destination. Message status show the transport type, how many device message must pass and how many device left to reach final destination. The data that change is **FROM address**, **TO address**, **message status** and **extended data 1** all the other data stay same. If command data was different then extended data will be much more or less. In figure 1 intermediate 2, 3 is dual band, that mean can receive/transmit wired or wireless. In this example final device is very far to reach MDC, from **ROUT MAP** procedure MDC found the quickest path that lead to final device, the path show that MDC sent/receive data via wired, intermediate 1 sent/receive via wired, intermediate 2 receive data wired and sent it wireless, intermediate 3 receive data wireless and sent wired and final device receive/transmit wired.

In message status retransmission number decrease each time that past form one device to the next, when retransmission number reach 0 that mean there is final device. DBDN extended data show the physical path of each device that takes part in this routing + direct message procedure.

Table 26 Extended data 1 in routing + direct prodesure as example [1=wired, 0=wireless]

Extended Data 1=DBDN							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
FD IN 1	FD OUT 1	DBDN3 IN 0	DBDN3 OUT 1	DBDN2 IN 1	DBDN2 OUT 0	DBDN1 IN 1	DBDN1 OUT 1

Each device when receive the message must change **TO address** field so can be the correct device for a smooth procedure. In example intermediate 1 receive message from MDC, when its own turn come to send the message to next device must change the field **TO**: intermediate 1 with intermediate 2 address and decrease by one the retransmission number.

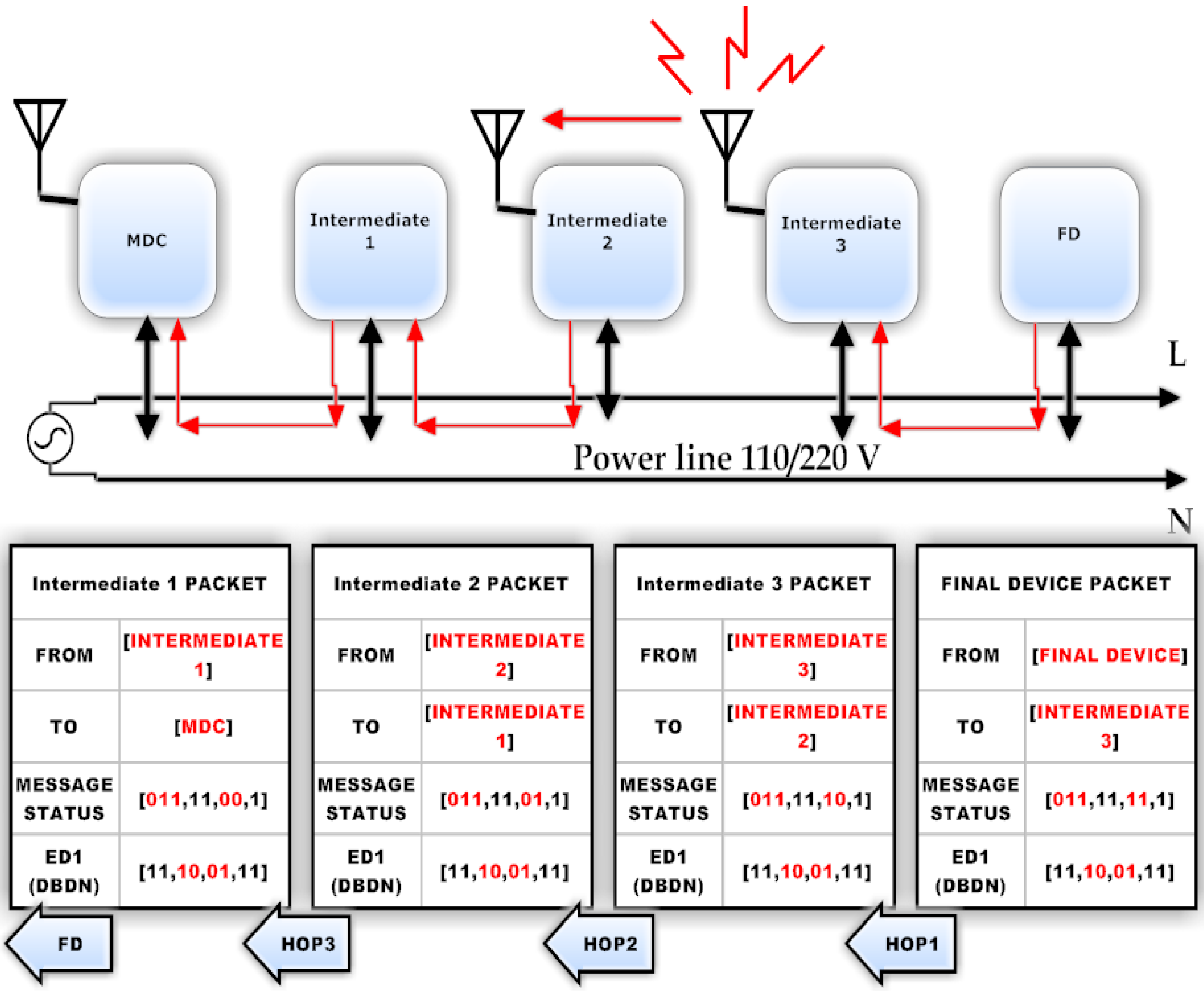


Figure 2 Routing + ASK message procedure with 3 Hops

In routing + ASK message final device must sent acknowledge back to MDC to confirm that command have been executed. In figure 2 final device have change message status in transmit type field to make it 011 from 010 and load retransmission number again with number three. Also have change DBDN data in dual band devices only to be correct from final device to MDC path. The return way from final device to MDC is the same as routing + direct message procedure but with different direction.

Table 27 Routing + direct message procedure address table

Routing + direct message, 3 Hop					
DATA	**MDC**	**Intermediate 1**	**Intermediate 2**	**Intermediate 3**	**FINAL DEVICE**
FROM	[MDC]	[Intermediate 1]	[Intermediate 2]	[Intermediate 3]	
TO	[Intermediate 1]	[Intermediate 2]	[Intermediate 3]	[FINAL DEVICE]	
ED3	[Int 1][Int 2]	[Int 1][Int 2]	[Int 1][Int 2]	[Int 1][Int 2]	
ED4	[Int 3]	[Int 3]	[Int 3]	[Int 3]	
ED5	[FD]	[FD]	[FD]	[FD]	
	HOP 1➔	HOP 2➔	HOP 3➔	FD ➔	

Table 28 Routing + ASK message procedure address table

Routing + ASK message, 3 Hop					
DATA	**MDC**	**Intermediate 1**	**Intermediate 2**	**Intermediate 3**	**FINAL DEVICE**
FROM		[Intermediate 1]	[Intermediate 2]	[Intermediate 3]	[FINAL DEVICE]
TO		[MDC]	[Intermediate 1]	[Intermediate 2]	[Intermediate 3]
ED3		[Int 1][Int 2]	[Int 1][Int 2]	[Int 1][Int 2]	[Int 1][Int 2]
ED4		[Int 3]	[Int 3]	[Int 3]	[Int 3]
ED5		[FD]	[FD]	[FD]	[FD]
		🡐 FD	🡐HOP 3	🡐 HOP 2	🡐 HOP 1

Message retrying

Message retrying is common procedure for direct connection and indirect connections. When devices is connected via direct link with MDC and if device didn't send any ASK message to MDC then MDC will try sent message and wait for response again. If the same action (MDC sent message /wait ASK) fails about 3 times then MDC will change status link to offline of that device and route map can start to discover the lost device. If from route map procedure discover lost device then route table number been add with the proper address from device. If the same problem occurs in indirect device connection, device status link will change to offline and route map procedure can start to solve the problem. If the problem remain and can't receive any ASK from device, status link change to dead link. Dead link mean the problem can't be solve need to check device electrical state or some other parameter.

In figure 3 there is the whole procedure for Retrying message repetition. When a message have been sent microprocessor wait for the right time until he receive reply, if that time end without receive any message from final device then he sent again a message and set to '1' retrying number, if receive an ASK message then refresh 'status link = online' if didn't receive any message then sent again a message and set to '2' retrying number. If retrying number reach number three then change 'status link= offline' and end. When a device have 'status link= offline' user have two options can start 'recheck' procedure to start again sent message and wait for ASK message, the second option is to start 'Route map' procedure, with that way a lost device can be found again. For indirect message the path going like this, same procedure sent message, wait for ASK message. The waiting time in this case is different because the message past from 1to 3 devices for that if hop=1 waiting time is different with hop=2. The difference in indirect procedure change after failure to receive ASK message and retrying number is equal with three. When 'status link= dead link' mean that a serious problem exist in that device. In 'status link=dead link' user have three options first to recheck and see if can reach that device, second can reset the 'status link' to default number that is '00'and when device have default state mean that can sent first direct message to discover final device, third case user can delete device if have some problem and be removed from all memory tables.

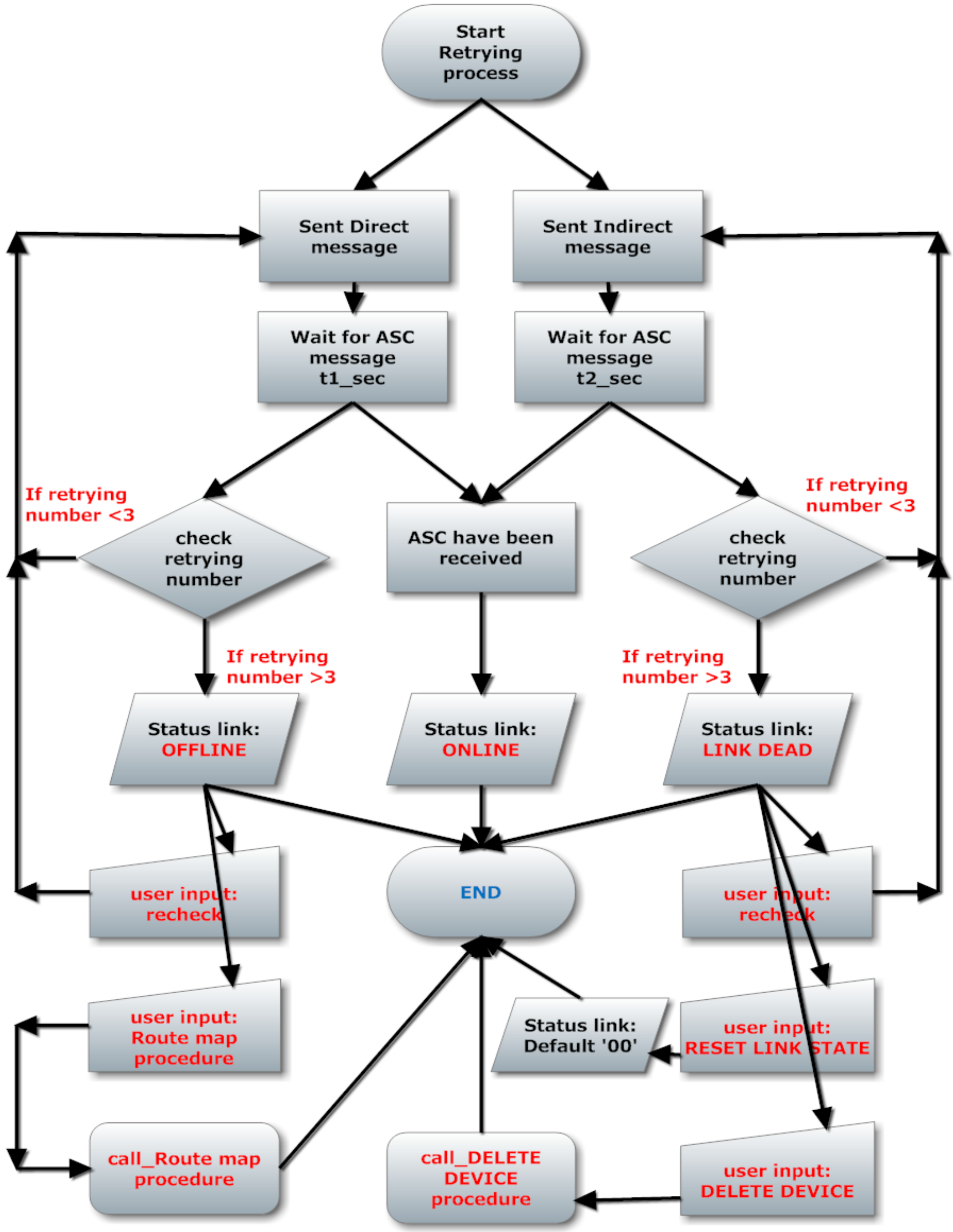


Figure 3 Retrying message procedures

3.1.3 O.P.R.A Message Summary

Table 29 Standard message summary

	Message	FROM address 4 bytes	TO address 4 bytes	Message Status 1 byte						Cmd 1 1 byte / Cmd 2 1 byte	CrC 2 byte
STANDARD				Type			Hop	Retry	Ext.		
	Broadcast	MDC	DEVICE	1	1	1	00	00	0	Broadcast command	CRC
	Group broadcast	MDC	Group number	1	1	0	00	00	0	Group command + param.	CRC
	House region broadcast	MDC	House region	1	0	1	00	00	0	House command + param.	CRC
	Route map	MDC	DEVICES	1	0	0	[X]	00	0	Route map commands	CRC
	Route + ASK	DEVICE, MDC	MDC, DEVICE	0	1	1	[X]	00	0	ASK commands	CRC
	Route + direct	MDC	DEVICE	0	1	0	[X]	[X]	0	Device commands	CRC
	ASK	DEVICE, MDC	MDC, DEVICE	0	0	1	[X]	00	0	ASK commands	CRC
	Direct	MDC	DEVICE	0	0	0	[X]	[X]	0	Device commands	CRC
				broadcast			3 max hop	device remaining	extended		

Table 30 Extended message summary

	Message	FROM address 4 bytes	TO address 4 bytes	Message Status 1 byte						Cmd 1 1 byte Cmd 2 1 byte	Udm 1 1 byte Udm 2 1 byte	Extended data 56 bytes	CRC 2 byte
EXTENDED				Type			Hop	Retry	Ext.				
	Broadcast	MDC	DEVICE	1	1	1	00	00	1	Broadcast command	[X]	ED1 - ED16	CRC
	Group broadcast	MDC	Group number	1	1	0	00	00	1	Group command + param.	[X]	ED1 - ED16	CRC
	House region broadcast	MDC	House region	1	0	1	00	00	1	House command + param.	[X]	ED1 - ED16	CRC
	Route map	MDC	DEVICES	1	0	0	[X]	00	1	Route map commands	[X]	ED1 - ED16	CRC
	Route + ASK	DEVICE, MDC	MDC, DEVICE	0	1	1	[X]	00	1	ASK commands	[X]	ED1 - ED16	CRC
	Route + direct	MDC	DEVICE	0	1	0	[X]	[X]	1	Device commands	[X]	ED1 - ED16	CRC
	ASK	DEVICE, MDC	MDC, DEVICE	0	0	1	[X]	00	1	ASK commands	[X]	ED1 - ED16	CRC
	Direct	MDC	DEVICE	0	0	0	[X]	[X]	1	Device commands	[X]	ED1 - ED16	CRC

3.1.4 O.P.R.A Commands

O.P.R.A messaging technology can work in many different devices either is wireless, wired or dual band. Also O.P.R.A protocol has been designed for the user so he can be informed and configure the devices in whole house. Remote controls like telecontrol, computer, and smart phones can be the right tool for the user so can be the master of the whole house/factory. To connect all that device and have the control of them need all devices obey in some rules and commands.

One criterion for a smart house automation is commands, type and diversity of commands make the system more open to be configurable by the user who need the maximum number of capability. O.P.R.A protocol has a wide number of commands as main and for each main command have also few for subcommand. The maximum number for main commands is 254 and for subcommands are also 254. Main commands are split in categories, for some devices they can use a shared number of command 1 and command 2. Each device have a unique category that mean for each device category have a repertoire of commands, a device can't execute command that is not in list of category that belong. Except the list for each category all device can execute commands that belong to protocol or MDC. Because total number of commands is too many to refer will be inform the most common and usable commands. O.P.R.A protocol gives the opportunity for companies to add their own commands to the system and be used for their devices. Main categories for commands are **Device, Protocol, MDC,** and **Special Function.**

Command 1

Command 1 holds an 8 bit number that inform device the type of execution it will make, command 1 is the primary command and command 2 can hold a parameter or be a subcommand but that depend from primary command type.

Command 2

When command 2 act as a parameter carrier the 8 bit data hold a number that says about the percent or level for brightness level, fade on/off. When command 2 act as subcommand the 8 bit data hold a second command for execution for example fast on/off.

Commands structure:

- Device commands
    - Socket [SO]
    - Light switch [LS]
    - Dimmer [DI]
    - Home entertainment [HE]
    - House protection [HP]
    - Special function [SF]
- Protocol commands
    - Group broadcast
    - House region
    - Broadcast
    - Direct message
    - ASK of direct message
    - Routing + direct message
    - Routing + ASK message
    - Route map
- MDC commands
- Special function commands

From the six main devices commands will focus in three, most usable – known categories. Light switch, socket and dimmer is the most used devices inside the house for that we focus to explore the capabilities for controlling these categories. For route map commands have a close look at Route map procedure, there describe the mechanism behind search and find and all the other details about the procedure and data. Special function commands is open for companies that wish to add their commands for their own devices, also space remain for feature firmware version and upgraded devices that can support new commands for use.

Table 31 Device-Socket commands

DEVICE COMMANDS – SOCKET [SO]		
Comments	**COMMAND 1**	**COMMAND 2**
Single socket	Power on	Fast on, soft on [x sec]
Single socket	Power off	Fast off
Multi socket	Power on (all)	Fast on, soft on [x sec]
Multi socket	Power off (all)	Fast off
Multi socket	Power on (.1)	Fast on, soft on [x sec]
Multi socket	Power on (.2)	Fast on, soft on [x sec]
Multi socket	Power on (.3)	Fast on, soft on [x sec]
Multi socket	Power on (.4)	Fast on, soft on [x sec]
Multi socket	Power on (.5)	Fast on, soft on [x sec]
Multi socket	Power on (.6)	Fast on, soft on [x sec]
Multi socket	Power on (.7)	Fast on, soft on [x sec]
Multi socket	Power on (.8)	Fast on, soft on [x sec]
Multi socket	Power off (.1)	Fast off
Multi socket	Power off (.2)	Fast off
Multi socket	Power off (.3)	Fast off
Multi socket	Power off (.4)	Fast off
Multi socket	Power off (.5)	Fast off
Multi socket	Power off (.6)	Fast off
Multi socket	Power off (.7)	Fast off
Multi socket	Power off (.8)	Fast off
Shared	Device time alarm on	Time alarm (all), (.1), (.2), (.3), (.4), (.5), (.6), (.7), (.8), time alarm (single)
Shared	Device time alarm off	Time alarm (all), (.1), (.2), (.3), (.4), (.5), (.6), (.7), (.8), time alarm (single)

Table 32 Device-Light Switch commands

DEVICE COMMANDS – LIGHT SWITCH [LS]		
Comments	**COMMAND 1**	**COMMAND 2**
Single light switch	Light on	Fast on, fade on [x steps], random bright
Single light switch	Light off	Fast off, fade off [x steps]
Single light switch	Dimming on	Brightness [x % 1-100]
Multi light switch	Light on (all)	Fast on, fade on [x steps], random bright
Multi light switch	Light off (all)	Fast off, fade off [x steps]
Multi light switch	Dimming on (all)	Brightness [x % 1-100]
Multi light switch	Light on (.1)	Fast on, fade on [x steps], random bright
Multi light switch	Light on (.2)	Fast on, fade on [x steps], random bright
Multi light switch	Light on (.3)	Fast on, fade on [x steps], random bright

Multi light switch	Light on (.4)	Fast on, fade on [x steps], random bright
Multi light switch	Light on (.5)	Fast on, fade on [x steps], random bright
Multi light switch	Light on (.6)	Fast on, fade on [x steps], random bright
Multi light switch	Light off (.1)	Fast off, fade off [x steps]
Multi light switch	Light off (.2)	Fast off, fade off [x steps]
Multi light switch	Light off (.3)	Fast off, fade off [x steps]
Multi light switch	Light off (.4)	Fast off, fade off [x steps]
Multi light switch	Light off (.5)	Fast off, fade off [x steps]
Multi light switch	Light off (.6)	Fast off, fade off [x steps]
Multi light switch	Dimming on (.1)	Brightness [x % 1-100]
Multi light switch	Dimming on (.2)	Brightness [x % 1-100]
Multi light switch	Dimming on (.3)	Brightness [x % 1-100]
Multi light switch	Dimming on (.4)	Brightness [x % 1-100]
Multi light switch	Dimming on (.5)	Brightness [x % 1-100]
Multi light switch	Dimming on (.6)	Brightness [x % 1-100]
Shared	Device time alarm on	Time alarm (all), (.1), (.2), (.3), (.4), (.5), (.6), time alarm (single)
Shared	Device time alarm off	Time alarm (all), (.1), (.2), (.3), (.4), (.5), (.6), time alarm (single)

Table 33 Device-Dimmer commands

DEVICE COMMANDS – DIMMER [DI]		
Comments	**COMMAND 1**	**COMMAND 2**
Single dimmer	Light on	Fast on, fade on [x steps], random bright
Single dimmer	Light off	Fast off, fade off [x steps]
Single dimmer	Dimming on	Brightness [x % 1-100]
Shared	Device time alarm on	[empty]
Shared	Device time alarm off	[empty]

From the above tables (socket, light switch and dimmer commands) 53 commands is used as primary command for execution and for each primary there is a subcommand or parameter. Parameter in command 2 is soft on [x sec], fade on/off [x steps] and brightness [x % 1-100], in soft on [x] the variable x is a range of time that parameter can take and make socket start slowing to protect circuit and electrical devices. In fade on/off [x] the variable x is a range of steps that light brightness will raise/fall until reach the full brightness or the zero brightness. For brightness [x %] the variable x is a number from 1 to 100 percent that make dimmable bulb light range from 1% to 100% of bulb brightness.

Table 33 Protocol-Group Broadcast commands

PROTOCOL COMMANDS – GROUP BROADCAST		
Comments	**COMMAND 1**	**COMMAND 2**
	Start (all)	Fast on
	Stop (all)	Fast off
	Start (all sockets)	Fast on, soft on [x sec]
	Stop (all sockets)	Fast off
	Start (all light switch)	Fast on, fade on [x steps], random bright, Brightness [x % 1-100]
	Stop (all light switch)	Fast off, fade off [x steps]
	Start (all dimmer)	Fast on, fade on [x steps], random bright, Brightness [x % 1-100]
	Stop (all dimmer)	Fast off, fade off [x steps]
	Start (all SF)	[special commands for each special device]
	Stop (all SF)	[special commands for each special device]
Terminate group from devices	Disjoin group	[empty] **user code required for this action**
MDC to Group EW=Eeprom write	Download group data	EW (all), EW(ED6), EW(ED9), EW(ED12), EW(ED13), EW(ED14)
Group to MDC ER=Eeprom read	Upload group data	ER (all), ER(ED6), ER(ED9), ER(ED12), ER(ED13), ER(ED14)
Shared	Group time alarm on	Time alarm (all), (all sockets), (all light switch), (all dimmer), (all SF)
Shared	Group time alarm off	Time alarm (all), (all sockets), (all light switch), (all dimmer), (all SF)
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 34 Command 2 for protocol-group commands details

Command 2							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
Not in use	Not in use	Not in use	1=ED14 ON EW/ER(ALL) =>**1**	1=ED13 ON **1**	1=ED12 ON **1**	1=ED9 ON **1**	1=ED6 ON **1**

Table 35 Protocol-House Region Broadcast commands

PROTOCOL COMMANDS – HOUSE REGION BROADCAST		
Comments	**COMMAND 1**	**COMMAND 2**
	Start (all)	Fast on
	Stop (all)	Fast off
	Start (all sockets)	Fast on, soft on [x sec]
	Stop (all sockets)	Fast off
	Start (all light switch)	Fast on, fade on [x steps], random bright, Brightness [x % 1-100]
	Stop (all light switch)	Fast off, fade off [x steps]
	Start (all dimmer)	Fast on, fade on [x steps], random bright, Brightness [x % 1-100]
	Stop (all dimmer)	Fast off, fade off [x steps]
	Start (all SF)	[special commands for each special device]
	Stop (all SF)	[special commands for each special device]
Terminate house region from devices	Disjoin house region	[empty] **user code required for this action**
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 36 Protocol- Broadcast commands

PROTOCOL COMMANDS – BROADCAST		
Comments	**COMMAND 1**	**COMMAND 2**
Panic button	Emergency	Shut-down all, power-up all, reset all
Add devices to group	Join group	[empty] **maximum 10 devices per message**
Add devices to house region	Join House region	[empty] **maximum 10 devices per message**
User code protection for device	User device password	EW (write), EW (erase)
MDC to device	Set devices clock time	**hh**:**mm =hour, minute for device with real time clock**
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 37 Protocol- Direct message commands

PROTOCOL COMMANDS – DIRECT MESSAGE		
Comments	**COMMAND 1**	**COMMAND 2**
Recheck link status	Direct status questing	Sent, reply
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 38 Protocol- ASK of Direct message commands

PROTOCOL COMMANDS – ASK OF DIRECT MESSAGE		
Comments	**COMMAND 1**	**COMMAND 2**
	Direct ASK	Success, failure
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 39 Protocol- Routing + Direct message commands

PROTOCOL COMMANDS – ROUTING + DIRECT MESSAGE		
Comments	**COMMAND 1**	**COMMAND 2**
Recheck link status	Routing + direct status questing	Sent, reply
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 40 Protocol- Routing + ASK message commands

PROTOCOL COMMANDS – ROUTING + ASK MESSAGE		
Comments	**COMMAND 1**	**COMMAND 2**
	Routing + ASK	Success, failure
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

From the above tables (protocol group broadcast, house region broadcast, broadcast, direct message, ASK of direct message, routing +direct message and routing + ASK message commands) 35 commands is used as primary command for execution and for each primary there is a subcommand or parameter.

Table 41 MDC commands

MDC COMMANDS		
Comments	**COMMAND 1**	**COMMAND 2**
MDC to device	Lock device	M/R (on/on), M/R (on/off), M/R (off/on) M=manual control, R=remote control
MDC to device	Reset device	[empty]
MDC to Telecontrol	Upload clock time	**hh:mm =hour, minute, if telecontrol lost real time clock**
Remote controls to MDC	Download clock time	**hh:mm =hour, minute, if MDC lost real time clock**
Between MDC, Telecontrol	Synchronize real time clocks	If MDC clock==Telecontrol clock =pass If MDC clock [not equal] Telecontrol clock =user choose correct time and fix time
MDC to device	Manufactory settings	Default states for device data
MDC to remote control	Upload MDC file	Upload all data that is store in MDC to a remote control like telecontrol or computer if have been lost/update
remote control to MDC	Download Telecontrol file	Download all data that is store in remote control like telecontrol or computer to MDC if have been lost/update
device to MDC ER=Eeprom read	Read device data	ERO, ER (all), ER(ED2), ER(ED6), ER(ED7), ER(ED9), ER(ED10), ER(ED11), ER(ED15), ER(ED16) ERO=read only data
MDC to device EW=Eeprom write	Write device data	ER (all), ER(ED2), ER(ED6), ER(ED9), ER(ED10), ER(ED11), ER(ED16)
MDC to device	Delete device	Erase all data
MDC to device	Add device	
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

Table 42 Command 2 for MDC read commands details

Command 2							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
1=ED16 on	**1=ED15 on**	**1=ED11 on**	**1=ED10 on**	**1=ED9 on**	**1=ED7 on**	**1=ED6 on**	**1=ED2 on**
ER(ALL) =>1	1	1	1	1	1	1	1

Table 43 Command 2 for MDC write commands details

Command 2							
BIT7	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0(LSB)**
not in use	**not in use**	**1=ED16 on**	**1=ED11 on**	**1=ED10 on**	**1=ED9 on**	**1=ED6 on**	**1=ED2 on**
EW(ALL) =>1	1	1	1	1	1		

Table 44 Special function commands

SPECIAL FUNCTION COMMANDS		
Comments	**COMMAND 1**	**COMMAND 2**
These types of commands are for special device that use different commands to control them, also companies can add here there commands for their products.		
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

3.1.5 O.P.R.A Device Category

From the six main categories that device can be a total of 252 different devices can exist in standard MDC version 1.0. The maximum number of different device can grow up to 65.532 different devices in standard MDC version 1.1 but also in new version 1.2 the maximum number of device can raise more. Device can be one of each category: socket, light switch, dimmer, house protection, home entertainment and special function. Each category supports a maximum number of different versions of that category. Each device has a register in nonvolatile memory (flash) that informs which category belongs and the version of that category. Register name is 'category' and for standard/extended MDC version 1.0 is an 8bit data with 252 different device categories.

Table 45 Category data register details

CATEGORY DATA REGISTER								
Device	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
DI	0	0	0	0	0	1	2bit [x] - 4 different devices	
LS	0	0	0	0	1	3 bit [x] - 8 different devices		
SO	0	0	0	1	4 bit [x] - 16 different devices			
HE	0	0	1	5 bit [x] - 32 different devices				
HP	0	1	6 bit [x] - 64 different devices					
SF	1	7 bit [x] - 128 different devices						

Table 46 Dimmer Devices category and subcategory details

DIMMER DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
DI	General use	0	0	0	0	0	1	0	0
DI	Led	0	0	0	0	0	1	0	1
DI	[empty]	0	0	0	0	0	1	1	0
DI	[empty]	0	0	0	0	0	1	1	1

Table 47 Light switch Devices category and subcategory details

LIGHT SWITCH DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
LS	Single	0	0	0	0	1	0	0	0
LS	Double	0	0	0	0	1	0	0	1
LS	Triple	0	0	0	0	1	0	1	0
LS	Fourfold	0	0	0	0	1	0	1	1
LS	Fivefold	0	0	0	0	1	1	0	0
LS	Six fold	0	0	0	0	1	1	0	1
LS	[empty]	0	0	0	0	1	1	1	0
LS	[empty]	0	0	0	0	1	1	1	1

Table 48 Socket Devices category and subcategory details

SOCKET DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
SO	Single	0	0	0	1	0	0	0	0
SO	Double	0	0	0	1	0	0	0	1
SO	Triple	0	0	0	1	0	0	1	0
SO	Fourfold	0	0	0	1	0	0	1	1
SO	Fivefold	0	0	0	1	0	1	0	0
SO	Six fold	0	0	0	1	0	1	0	1
SO	Seven fold	0	0	0	1	0	1	1	0
SO	Eight fold	0	0	0	1	0	1	1	1
SO	[empty]	0	0	0	1	1	0	0	0
SO	[empty]	0	0	0	1	1	0	0	1
SO	[empty]	0	0	0	1	1	0	1	0
SO	[empty]	0	0	0	1	1	0	1	1
SO	[empty]	0	0	0	1	1	1	0	0
SO	[empty]	0	0	0	1	1	1	0	1
SO	[empty]	0	0	0	1	1	1	1	0
SO	[empty]	0	0	0	1	1	1	1	1

Table 49 Home entertainment Devices category and subcategory details

HOME ENTERTAINMENT DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
HE	[empty]	0	0	1	0	0	0	0	0
HE	[empty]	0	0	1	0	0	0	0	1
HE	[empty]	-	-	-	-	-	-	-	-
HE	[empty]	0	0	1	1	1	1	1	1

Table 50 House protection Devices category and subcategory details

HOUSE PROTECTION DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
HP	[empty]	0	1	0	0	0	0	0	0
HP	[empty]	0	1	0	0	0	0	0	1
HP	[empty]	-	-	-	-	-	-	-	-
HP	[empty]	0	1	1	1	1	1	1	1

Table 51 Special function Devices category and subcategory details

SPECIAL FUNCTION DEVICES CATEGORY AND SUBCATEGORY									
Main category	**Subcategory**	**BIT7**	**BIT6**	**BIT5**	**BIT4**	**BIT3**	**BIT2**	**BIT1**	**BIT0 (LSB)**
SF	[empty]	1	0	0	0	0	0	0	0
SF	[empty]	1	0	0	0	0	0	0	1
SF	[empty]	-	-	-	-	-	-	-	-
SF	[empty]	1	1	1	1	1	1	1	1

Some subcategory devices from house protection category are: smoke sensor, fire sensor, gas sensor, water sensor and alarm panel. Some subcategory devices from special function category are: water heater, electric oven, air condition, thermostat, electric heater, electric irrigation, electric tent, electric shutter motor, electric curtains and ceiling fans.

3.1.6 O.P.R.A Device Linking

A way to connect device and control them as a team is to create group or house region. Maximum number of group that a user can create is 15 and for house region 16 also. There is an 8 bit data register that share it both group and house region and keep inside data register two number, one for group and one for house region. In new protocol version 1.1 the total number for group can raise to 1023 different groups and 63 for house region. User can add device as memory allow it, the same for house region, if someone has in his house 100 devices he can create one group to add many of them. From figure 4 there is a house with 20 devices inside, each room have few devices, from the example can see the procedure to create a house region team and control those devices inside a room. At beginning MDC knows every device that is connected via power line (wired, single band) or via RF waves (wireless, single band) or via power line/RF (dual band). User want to control some devices in specific room of the house but he don’t need something complicate like to create a group for that he will add these devices to a specific house region room like bedroom 1. User chooses which device will add to bedroom house region and when it finishes MDC start the procedure of ‘join house region’. MDC sent a broadcast message in whole house with device address, command= ‘join house region’ room and the correct data for transfer. Because in broadcast message there is not any acknowledge message from devices to MDC to confirm the success, MDC check each device if they execute broadcast command for ‘join house region’ and after response from each device it end the procedure, in that moment user can sent commands to control all device in bedroom 1.

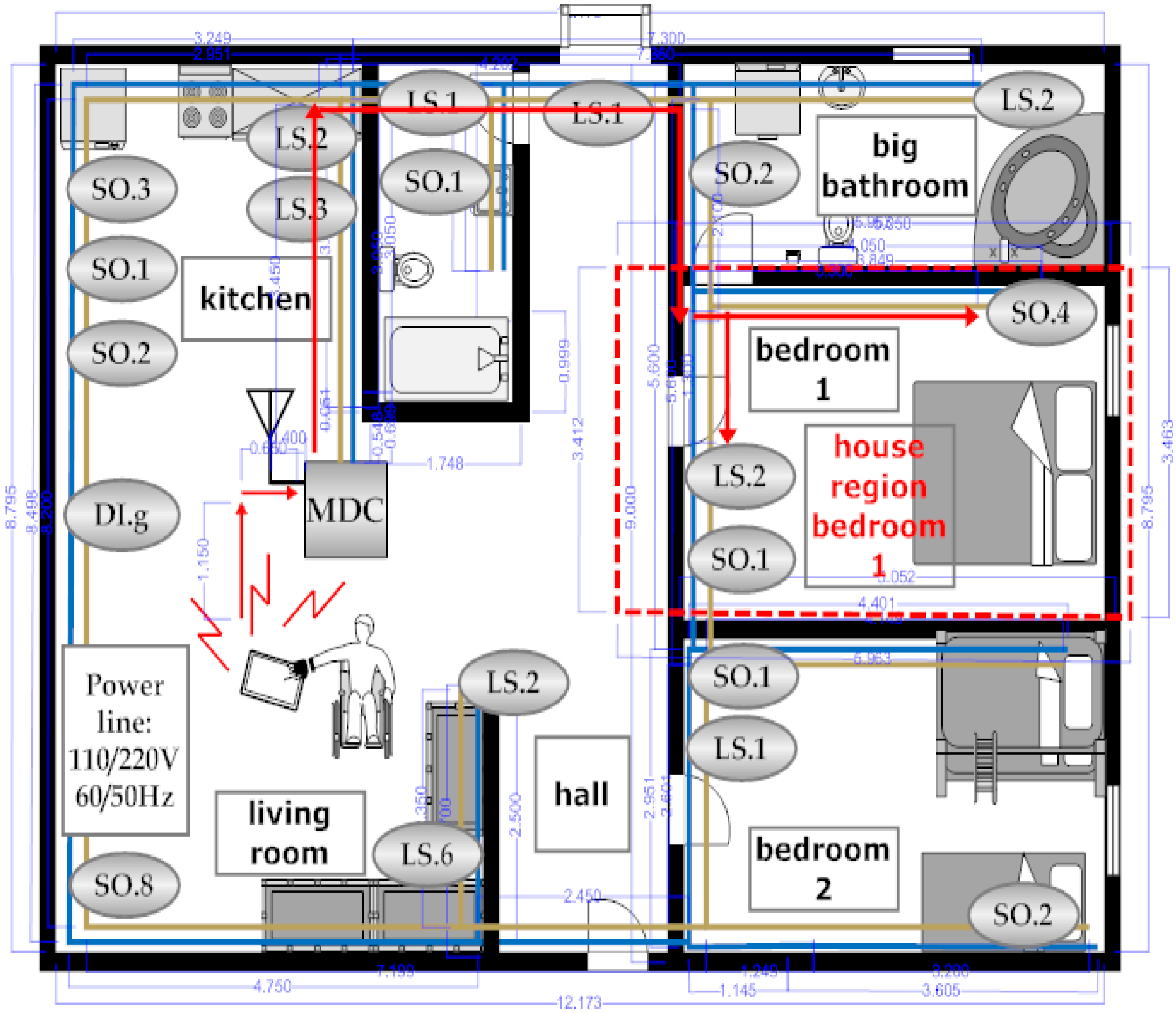


Figure 4 Example of a house region procedure

Table 52 Example of devices inside a house – house region [bedroom 1]

TOTL DEVICE = 20			TOTAL HOUSE ROOMS = 7						
LS	DI	SO	bedroom 1	bedroom 2	big bathroom	small bathroom	hall	kitchen	Living room
3xLS.1	1xDI.G	4xSO.1	LS.2	LS.1	LS.2	LS.1	LS.1	LS.2	DI.G
4xLS.2	-	3xSO.2	SO.1	SO.1	SO.2	SO.1	LS.2	LS.3	LS.6
1xLS.3	-	1xSO.3	SO.4	SO.2	-	-	-	SO.1	SO.8
1xLS.6	-	1xSO.4	-	-	-	-	-	SO.2	-
-	-	1xSO.8	-	-	-	-	-	SO.3	-

Table 53 Example - broadcast message for house region creation

DATA		Variable	CONTENTS
FROM: Address		[ffff,ffff]	MDC address
TO: category of each device and automatic MDC ascending number		[ascending number][category] [ascending number][category]	Device 1 identifier 16bit Device 2 identifier 16bit
Message Status	Transmit type	111	Broadcast
	Hop number	00	0 hop
	Retransmission number	00	0 retransmission
	Set Extended message	1	extended message
Command 1		[ff]	“join house region”
Command 2		[empty]	
User data map 1 (UDM 1)		[10001000]	Extended data content 4, 8 on
User data map 2 (UDM 2)		0	
Extended Data 4		[ascending number][category]	Device 3 identifier 16bit
Extended Data 8		[0, 0101]	House region number /bedroom 1
CRC		[ffff]	Cyclic Redundancy Check

Table 54 Inside **TO**: carrying identifier of devices

TO: 0-15 BIT DEVICE 1 INDENTIFIER							
BIT15	**BIT-**	**BIT-**	**BIT8**	**BIT7**	**BIT-**	**BIT-**	**BIT0(LSB)**
Device 1 ascending number created by MDC				Device 1 category data			
TO: 16-31 BIT DEVICE 2 INDENTIFIER							
BIT15	**BIT-**	**BIT-**	**BIT8**	**BIT7**	**BIT-**	**BIT-**	**BIT0(LSB)**
Device 2 ascending number created by MDC				Device 2 category data			

Table 55 Inside **ED4** carrying identifier of device

EXTENDED DATA 4: 0-15 BIT DEVICE 3 INDENTIFIER							
BIT15	**BIT-**	**BIT-**	**BIT8**	**BIT7**	**BIT-**	**BIT-**	**BIT0(LSB)**
Device 3 ascending number created by MDC				Device 3 category data			

Table 56 inside **ED8** house region and group number

BIT0-BIT3 Bit code 0000 to 1111 0 to16	Description
0000 – 0	All rooms
0001 – 1	Bathroom – default
0010 – 2	Kitchen – default
0011 – 3	Living room – default
0100 – 4	Hall – default
0101 – 5	Bedroom 1 – default
0110 – 6	Bedroom 2 - default
0111 – 7	Bedroom 3 - default
1000 – 8	User /add room – [empty]
1001 – 9	User /add room – [empty]
1010 – 1111 11-16	User /add room - [empty]

After house region procedure it left the procedure to create a group, connect different devices inside a house and have total control from all of them. Data for group is much more than house region for that we use same number and same devices as we used in above example. In that time we want more prospect from the devices like more variety from commands or maybe to use time alarm. At beginning MDC knows all device address, user choose which device to add and write the name of that group, in this point user can choose more details about the group he create but to form a group minimum variable is which device and the name of that group. After choosing device and writing group name MDC start the procedure to form a group with a broadcast message. Because broadcast message don't have any ASK message from device to MDC the role of MDC is to check if device belong to a group if is true it send direct message to each device and wait for response. When that procedure end user can configure the group options and sent data to device for execution. Now that device belong to a group user can control all of them by sending commands like 'start all light switch' in bedroom 1 group.

Table 57 Example of devices inside a house – group 1 [LS.2, SO.1, SO.4]

TOTL DEVICE = 20			TOTAL HOUSE ROOMS = 7						
LS	**DI**	**SO**	**bedroom 1**	**bedroom 2**	**big bathroom**	**small bathroom**	**hall**	**kitchen**	**Living room**
3xLS.1	1xDI.G	4xSO.1	LS.2	LS.1	LS.2	LS.1	LS.1	LS.2	DI.G
4xLS.2	-	3xSO.2	SO.1	SO.1	SO.2	SO.1	LS.2	LS.3	LS.6
1xLS.3	-	1xSO.3	SO.4	SO.2	-	-	-	SO.1	SO.8
1xLS.6	-	1xSO.4	-	-	-	-	-	SO.2	-
-	-	1xSO.8	-	-	-	-	-	SO.3	-

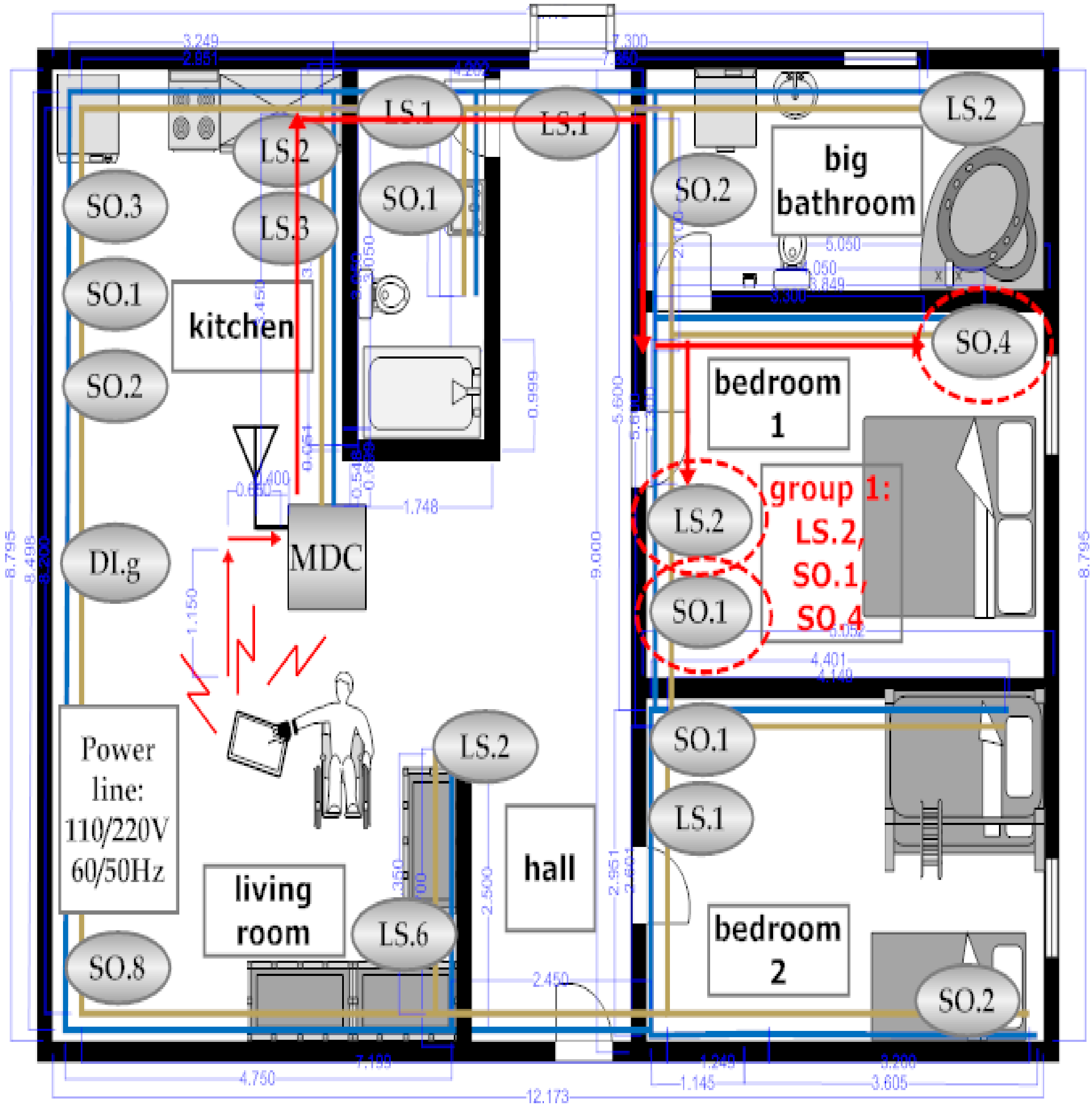


Figure 5 Example of a group procedure

When 'group state=on' device can accept commands for open/close only from group commands and from group time alarm. If 'group state=off' device can accept commands for open/close from group commands, group time alarm if 'group schedule=on', device commands and device time alarm if 'group schedule=off'.

Table 58 Example - broadcast message for group creation

DATA		Variable	CONTENTS
FROM: Address		[ffff,ffff]	MDC address
TO: category of each device and automatic MDC ascending number		[ascending number][category] [ascending number][category]	Device 1 identifier 16bit Device 2 identifier 16bit
Message Status	Transmit type	111	Routing +direct message
	Hop number	00	0 hop
	Retransmission number	00	0 retransmission
	Set Extended message	1	extended message
Command 1		[ff]	"join group"
Command 2		[empty]	
User data map 1 (UDM 1)		[10001000]	Extended data content 4, 8, 14 on
User data map 2 (UDM 2)		[00100000]	
Extended Data 4		[ascending number][category]	Device 3 identifier 16bit
Extended Data 8		[0001, 0]	Group number 1
Extended Data 14		[string]	Group name [12 bytes max]
CRC		[ffff]	Cyclic Redundancy Check

Table 59 Example - group message for downloading group data settings

DATA		Variable	CONTENTS
FROM: Address		[ffff,ffff]	MDC address
TO: not in use		[empty]	Only group number need
Message Status	Transmit type	110	Group message
	Hop number	00	0 hop
	Retransmission number	00	0 retransmission
	Set Extended message	1	extended message

Command 1	[ff]	"download group data" EW (6, 9, 12, 13)
Command 2	[---01111]	
User data map 1 (UDM 1)	[10100000]	Extended data content 13, 12, 9, 8, 6 on
User data map 2 (UDM 2)	[00011001]	
Extended Data 6	[---111--]	Group state=on, group schedule=on, group default state=on
Extended Data 8	[0001, 0]	Group number 1
Extended Data 9	[0001----]	Group day period Monday-Friday
Extended Data 12	[20:00]	Group post on
Extended Data 13	[01:00]	Group post off
CRC	[ffff]	Cyclic Redundancy Check

3.1.7 O.P.R.A Route Map

Route map is a procedure for MDC to search and discover devices that connection state is offline or link dead. Route map procedure always tries to find the shortest way to reach the lost device without any intermediate device between MDC and lost device. There is two ways to start route map procedure, first is manually by user and last automatically. Manually user start route map and wait for result, whereas automatically default time is 3:00 AM for starting 'automatic recovery lost connection", each day at 3:00 AM this procedure start for seek and repair all the lost connected device that exist, user can change the time that start. Route map procedure is split in levels; maximum level is three as maximum intermediate device can be connected with the final device. First level is for direct connection devices if fail to reach it then goes to level 2. In level 2 MDC check final device response using one intermediate device, if all combination between final device and some other device fail then goes to level 3. In that level combination is more complex because it uses two intermediate devices to connect final device, if all fail then goes to level 4 and last one. In final level combination is to complicate and need more time to finish because it uses three different intermediate devices as the way to reach final device. If level 4 fail to connect final device with MDC then the 'status state' goes 'link dead', end procedure for that device and moving to next device. In below examples we see 'automatic recovery lost connections', route map level 1, 2 and 3.

Automatic recovery lost connection

In MDC database all devices have a register that enumerate MDC the status of each devices connection if it is online, offline or link dead. When the procedure start to fix all broken connection the mechanism always check ‘status link’ bit of each device, when the state is ‘11=online’ that mean device is online and not need repair. Every time that MDC see ‘online’ state it moves to the next device for checking, increasing the number from total installed device in the system and when that number reaches overall number of installed device end the whole procedure. When MDC find ‘01=offline’ check the transmit type (direct or routing +direct) and from each case sent direct or routing + direct message and wait for ASK reply. When reply comes update the status link state to ‘11=online’ and move to next device. If wait time ends from MDC without any ASK then start route map procedure. When route map end and repair connection problem then move to next device or else update status link to ‘10=link dead’ after that move to next device, all this procedure repeat again until MDC reach end of devices. In figure 6 is the flow chart of ‘automatic recovery lost connection’ procedure.

Route map

When route map procedure is been called manually or automatically the pass some data to route map procedure to know what decision it must take when algorithm is running. At level 1 route map algorithm MDC sent a direct message to unreached device and after that wait for response, if response come write status link to ‘online’ and then depend from data that have been pass route map can end or move to next device. If response not come route map move to level 2 procedure, figure 7. Level 1 and level 2 the different between them is that MDC connect one by one final device with an intermediate one device and check if response have been received, figure 8. The maximum possibilities that MDC can do with the broken connection device is: maximum installed device (in that point that number is for general use but from some other reason possibilities can be decrease). When it reach maximum possibilities and haven’t received any response go to level 3. Level 3 with level 2 the only difference is that it use two different devices connected each time and try to receive response from final device, the number of possibilities is: maximum installed device * maximum installed device – 1 (is a general type for calculate maximum possibilities). When MDC didn’t receive any response from total possibilities then it go to level 4, figure 9. Final level adds one more device and with three device combination the possibilities raise much more, the possibility number is: maximum installed device * maximum installed device – 1 * maximum installed device – 2. Even final level can’t solve connection problem then MDC write status link as ‘link dead’ and end route map procedure, figure 10.

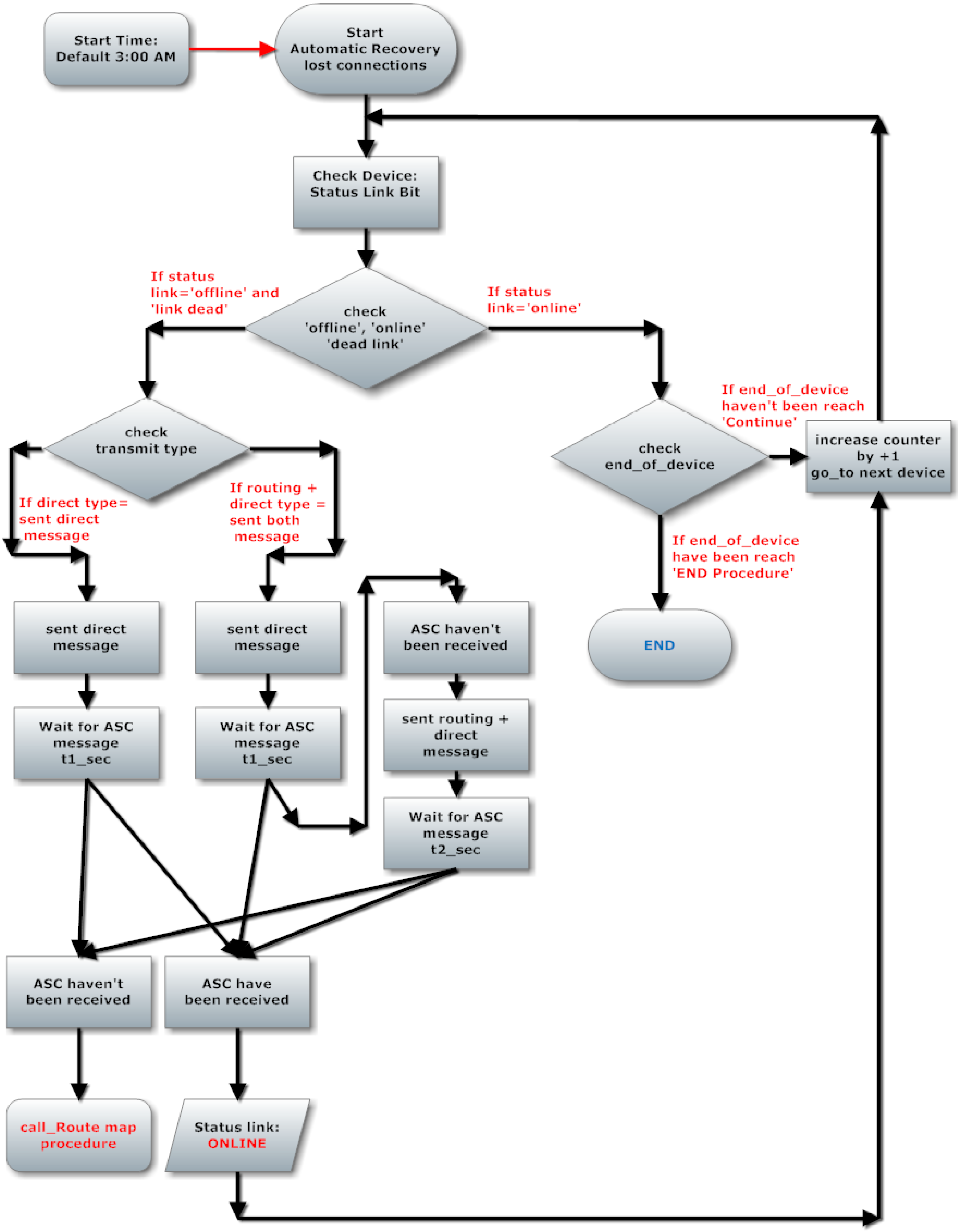


Figure 6 Automatic recovery lost connection procedures

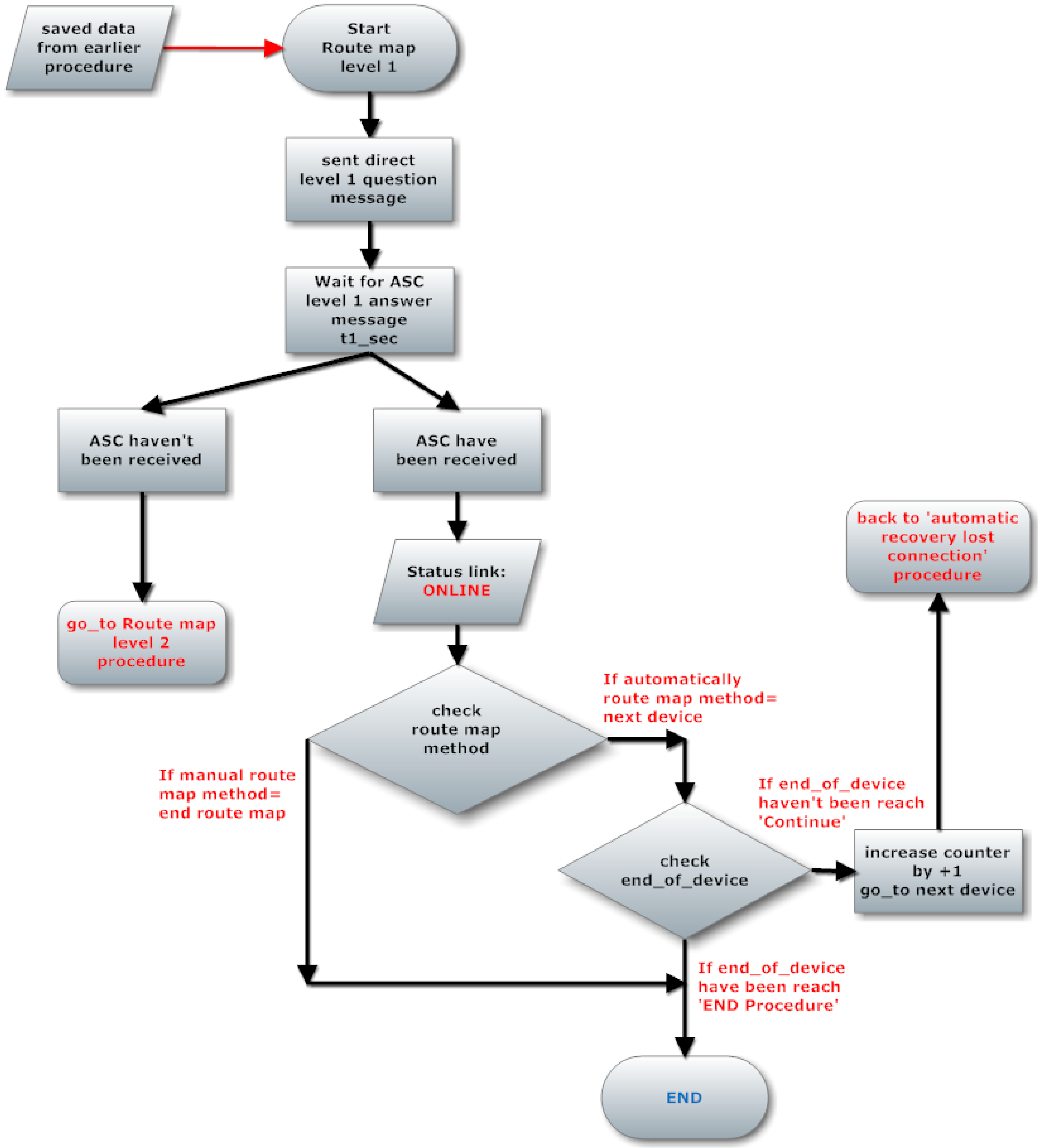


Figure 7 Route map level 1 procedures

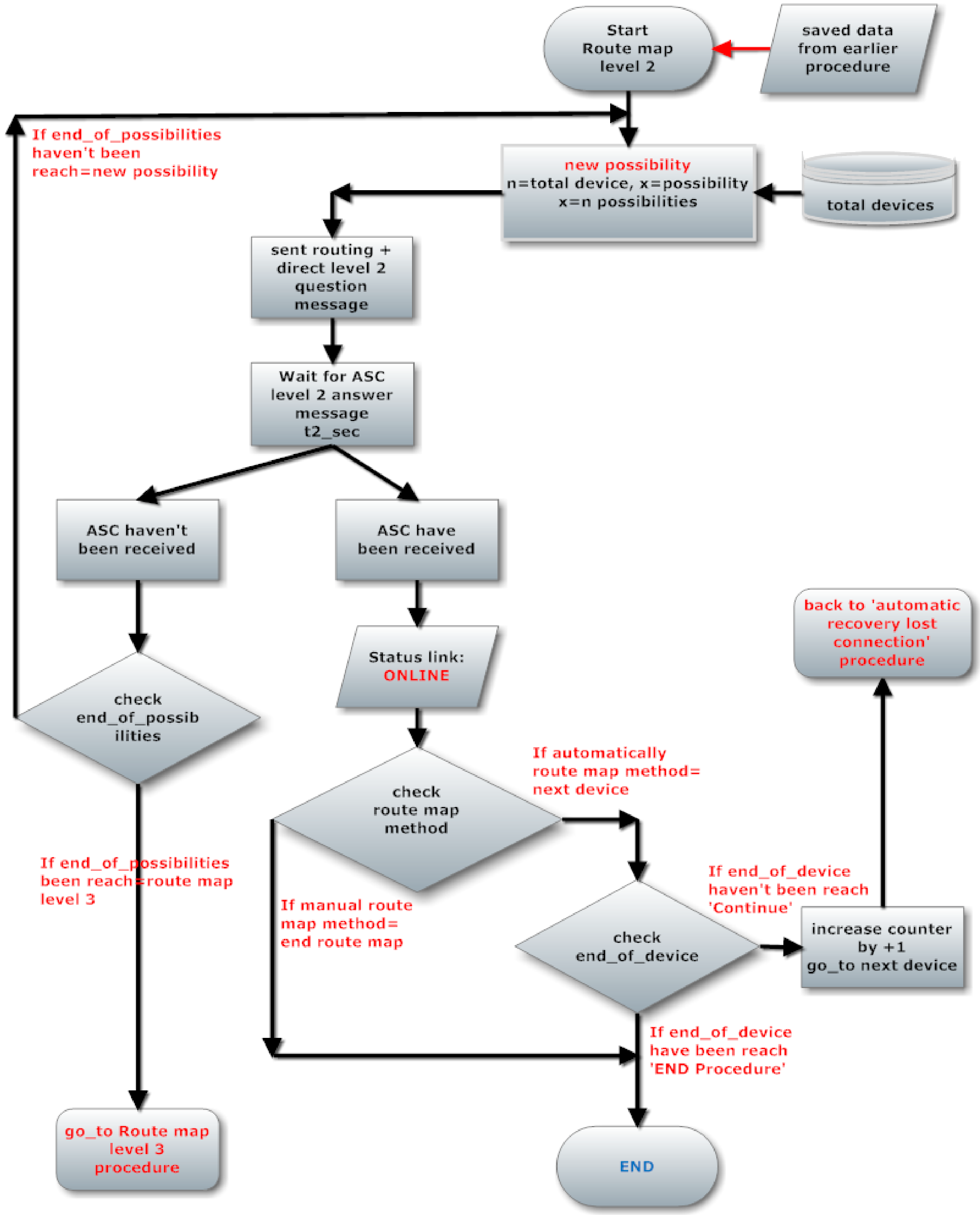


Figure 8 Route map level 2 procedures

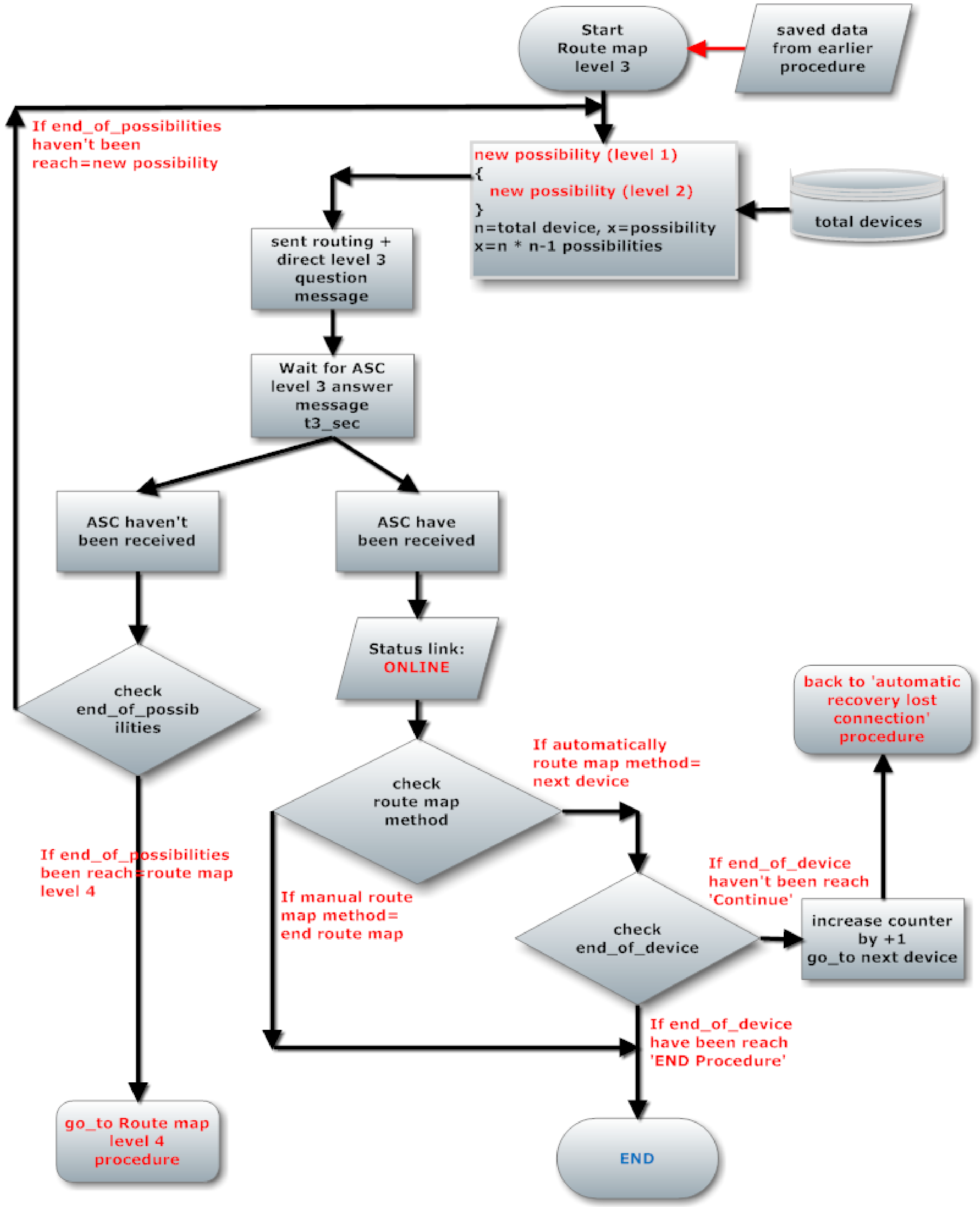


Figure 9 Route map level 3 procedures

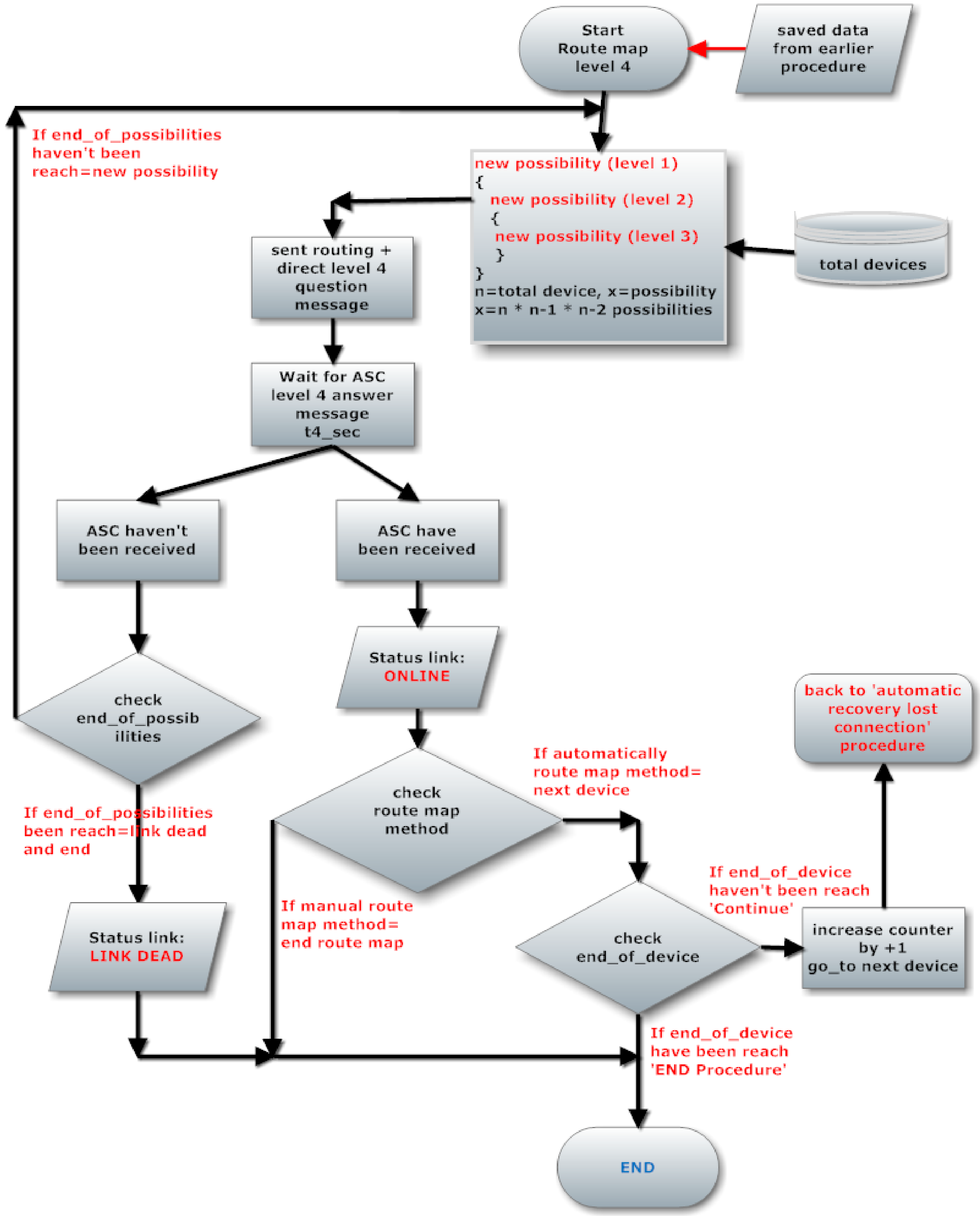


Figure 10 Route map level 4 procedures

Table 60 Route map commands

ROUTE MAP COMMANDS		
Comments	**COMMAND 1**	**COMMAND 2**
MDC➔FD	Route path level 1	L1(question), (answer), (erase)
MDC➔INT1➔FD	Route path level 2	L2(question), (answer), (write), (erase)
MDC➔INT1➔INT2➔FD	Route path level 3	L3(question), (answer), (write), (erase)
MDC➔INT1➔INT2➔INT3➔FD	Route path level 4	L4(question), (answer), (write), (erase)
New firmware version, upgraded device, companies commands	[empty]	[empty]
	[empty]	[empty]
	[empty]	[empty]

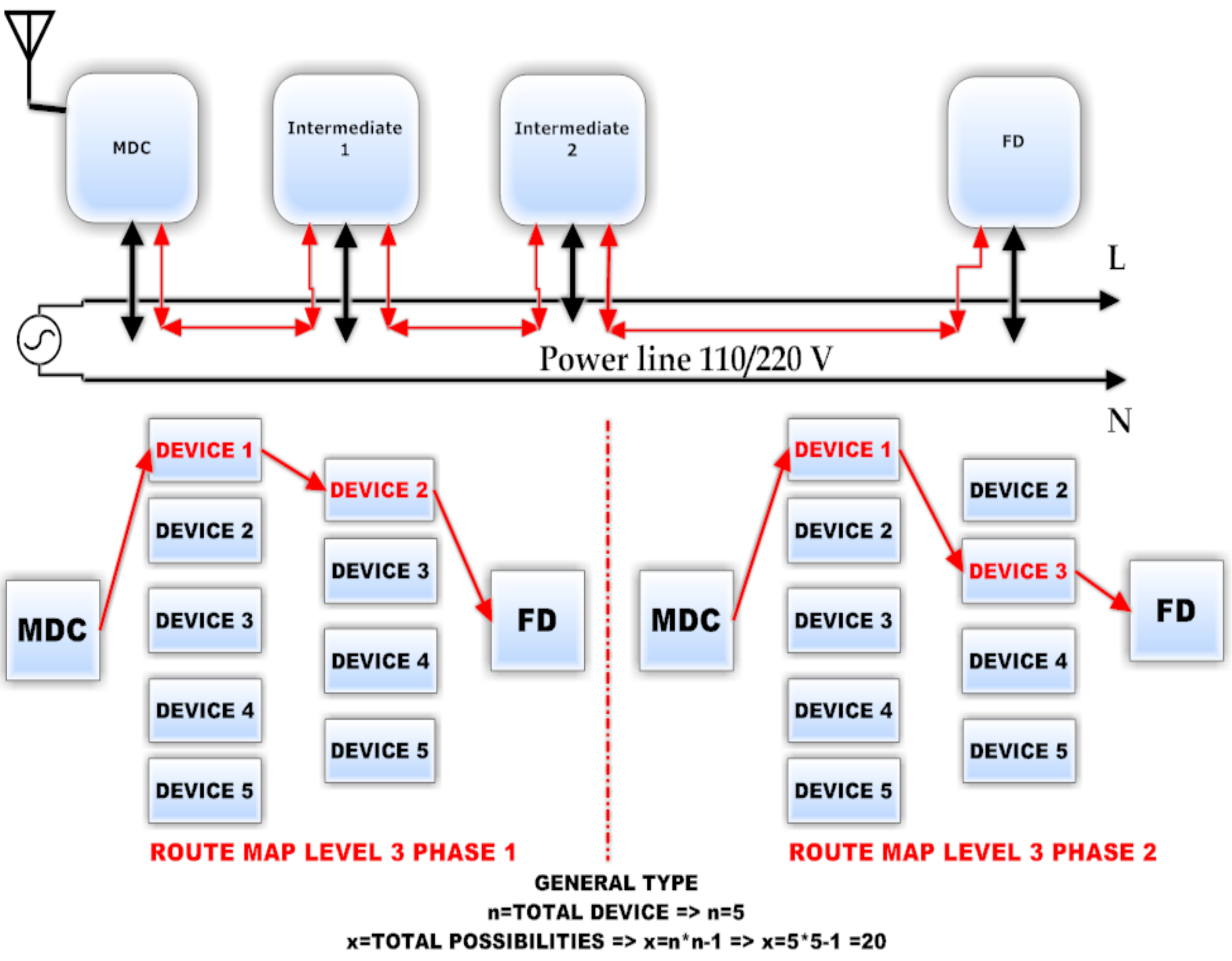


Figure 11 Example route map level 3 phase 1, 2 only

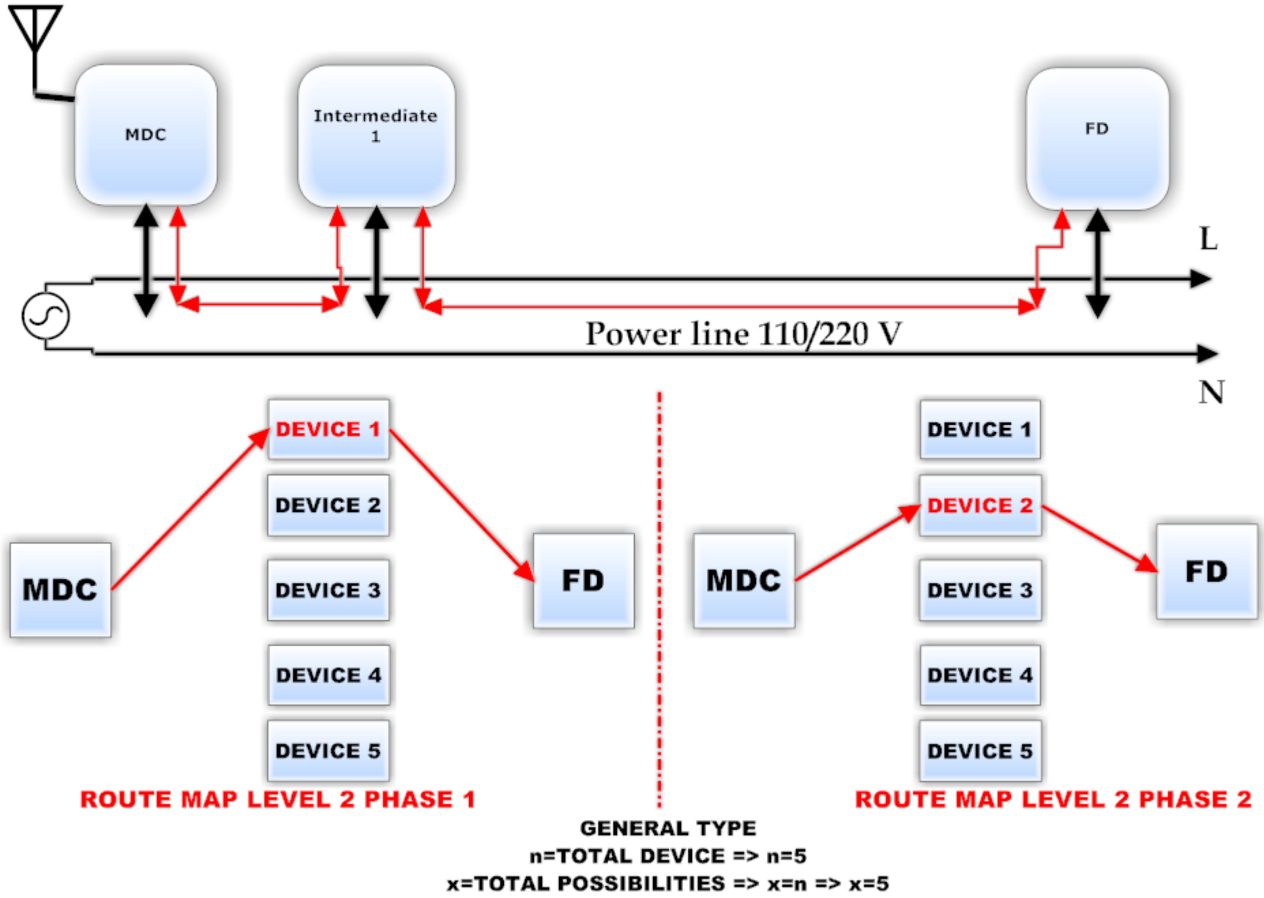


Figure 12 Example route map level 2 phase 1, 2 only

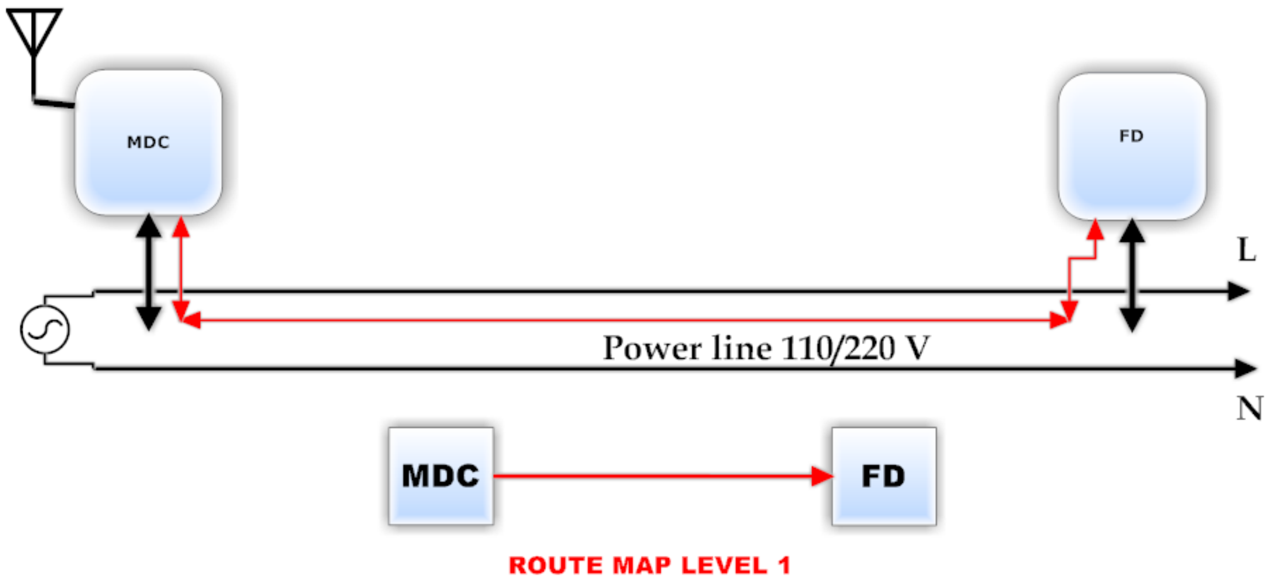


Figure 13 Example route map level 1

When route map procedure end writing status link with ‘online’ that mean the path to reach the problematic device have been discovered and that path can be add to problematic device in MDC memory with all details about sent/receive message to that device. When route path have been discovered MDC sent again a message with route map command writing in each intermediate device and final device the route table number, with that way when a device sent a message to device they will accept message if the route table number is correct. If automatic recovery lost connections procedure have establish direct connection with final device from a previous routing +direct state MDC will sent route map message with erase route table number from all the old path. In direct message between MDC and final device no need route table number or any other detail. From route map level 2until 4 MDC write the intermediate devices addresses, DBDN and route table number that is used to create a communicate channel between MDC and final device.

3.1.8 Data Table

For each device, group and some other data all of them consume memory space, for each device there is a maximum memory space that reserve and subtract from total free memory.

In this section will see:

- Device data table
- MDC data table
    - Device data
    - Group data
    - Route map data
    - Settings data

MDC Datastore	
Flash	permanent MDC data
Flash	manual MDC data (write/erase)
Eeprom	manual MDC data (write/erase)

DEVICE Datastore	
Flash	permanent device data
Eeprom	manual device data (write/erase)

Figure 14 Memory type inside MDC and device

Device data table

Table 61 Memory data inside device

DEVICE DATA TABLE					
Memory type	**Data name**	**Input / Output**	**Bits**	**Default state**	**comments**
FLASH	Device ID	Output	32	-	Device address
	Region code	Output	1	-	0= 220V 50Hz 1= 110V 60Hz
	Manufacture ID	Output	32	-	Constructed ID in line
	Protocol version	Output	24	'1.0'	Ascii character
	Category type	Output	8	-	LS, SO, DI, HP, SF, HE
	Device name	Output	16	-	Ascii character example 'LS'
	Transport state	Output	2	-	01=power line, 10= wireless, 11= dual band
Eeprom	Nickname	I/O	96	0	Ascii characters
	MDC ID	I/O	32	-	MDC address
	Route table number	I/O	8	0	A device can have multiple route table number to accept message only that is written in route table number, max 255
	House region / Group number	I/O	8	0	Max house region 16, max group 15
	Control state of device	I/O	5	[00000]	Bit4=group state on/off , bit3=group schedule on/off, bit 2=group default state, bit1=device schedule on/off, bit0=device default state
	Group name	I/O	96	0	Ascii characters
	Group post on	I/O	16	0	hh:mm
	Group post off	I/O	16	0	hh:mm
	Device post on	I/O	16	0	hh:mm
	Device post off	I/O	16	0	hh:mm
	Group day period / Device day period	I/O	8	0	
	Dimming brightness	I/O	8	0	1-100 %

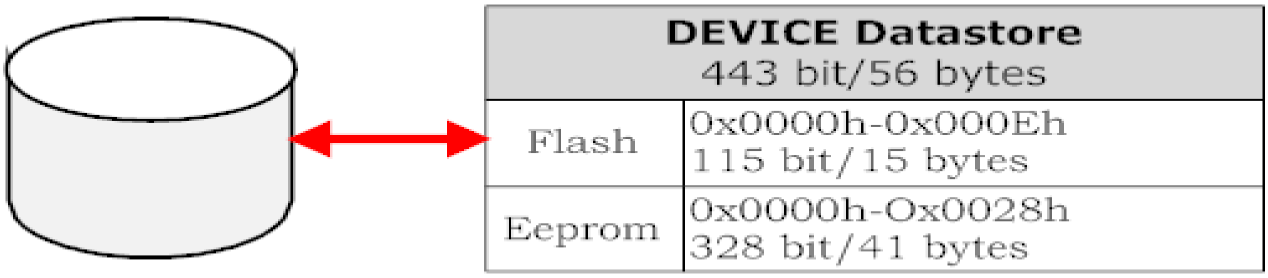


Figure 15 Memory space for device

MDC data table

Table 62 Flash memory data inside MDC

MDC-FLASH DATA TABLE					
Memory type	**Data name**	**Input / Output**	**Bits**	**Default state**	**comments**
FLASH	MDC ID	Output	32	-	MDC address
	Region code	Output	1	-	0= 220V 50Hz 1= 110V 60Hz
	Manufacture ID	Output	32	-	Constructed ID in line
	Protocol version	Output	24	'1.0'	Ascii character
	Name	Output	24	'MDC'	Ascii character
	More data is written in permanent flash memory like 2.45GHz IEEE 802.15.4 MAC details and data structures				
	SETTINGS DATA: HOUSE REGION ROOMS NAME				
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1000** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1001** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1010** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1011** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1100** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1101** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1110** house region number
	[empty]	Input	80	-	Ascii character, bit3-bit0 **1111** house region number
	More data is written in manual flash memory for user settings.				

Table 63 Eeprom memory data inside MDC

MDC-EEPROM DATA TABLE					
Memory type	**Data name**	**Input / Output**	**Bits**	**Default state**	**comments**
Eeprom	**DEVICE DATA:**				
	Nickname	I/O	96	0	Ascii characters
	Device ID	I/O	32	-	device address
	Route table number	I/O	8	0	Link to route map data
	House region / Group number	I/O	8	0	Group number link to group data, max group 15, max house region 15
	Control state / Status state of device	I/O	5	[00000]	Bit4,3= transport state, bit 2=region code, bit1=device schedule on/off, bit0=device default state
	Category type	I/O	8	-	
	Device name	Input	16	-	Ascii characters
	Ascending number	I/O	8	0	Each category have an ascending number, that number increase from 1 to 255 and added to each new device that been installed
	Device post on	I/O	16	0	hh:mm
	Device post off	I/O	16	0	hh:mm
	Device day period	I/O	4	0	
	Link state	I/O	2	[00]	11=online, 10=link dead, 01=offline
	ROUTE MAP DATA:				
	Route table number	I/O	8	0	Link with device, max 255
	Intermediate 1 ID	I/O	32	0	Device 1 address
	Intermediate 2 ID	I/O	32	0	Device 2 address
	Intermediate 3 ID	I/O	32	0	Device 3 address
	DBDN	I/O	8	-	Bit7,6=FD in/out, bit5,4= DBDN3 in/out , bit3,2= DBDN2 in/out , bit1,0=DBDN1 in/out

Table 64 Continue - Eeprom memory data inside MDC

MDC-EEPROM DATA TABLE					
Memory type	**Data name**	**Input / Output**	**Bits**	**Default state**	**comments**
Eeprom	**GROUP DATA:**				
	Device ID	I/O	32	-	For each device address add 4 bytes or 32 bit
	Group number	I/O	4	0	Group number link to device, max 15
	Control state of device	I/O	3	[000]	Bit2=group state on/off , bit1=group schedule on/off, bit 0=group default state
	Group name	I/O	96	-	Ascii characters
	Group post on	I/O	16	0	hh:mm
	Group post off	I/O	16	0	hh:mm
	Group day period	I/O	4	0	

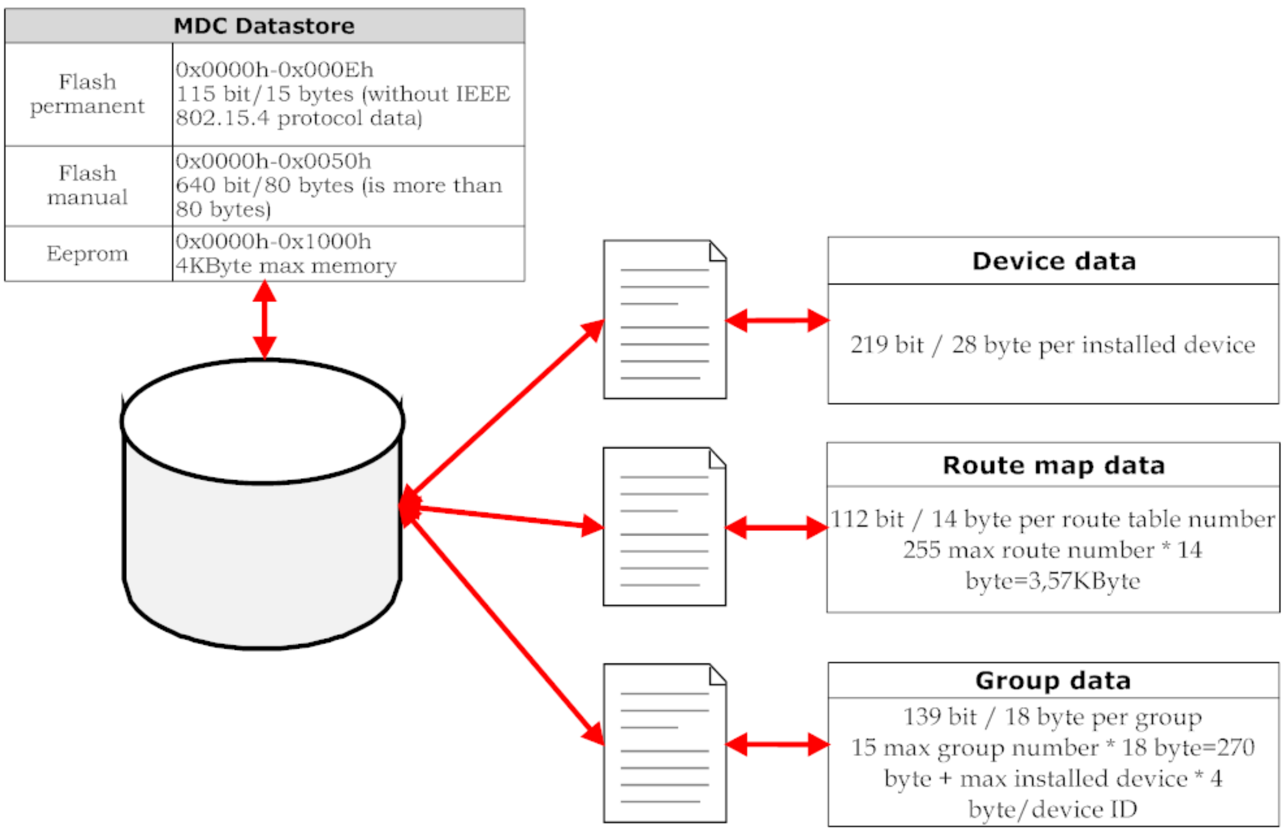


Figure 16 Memory space for MDC

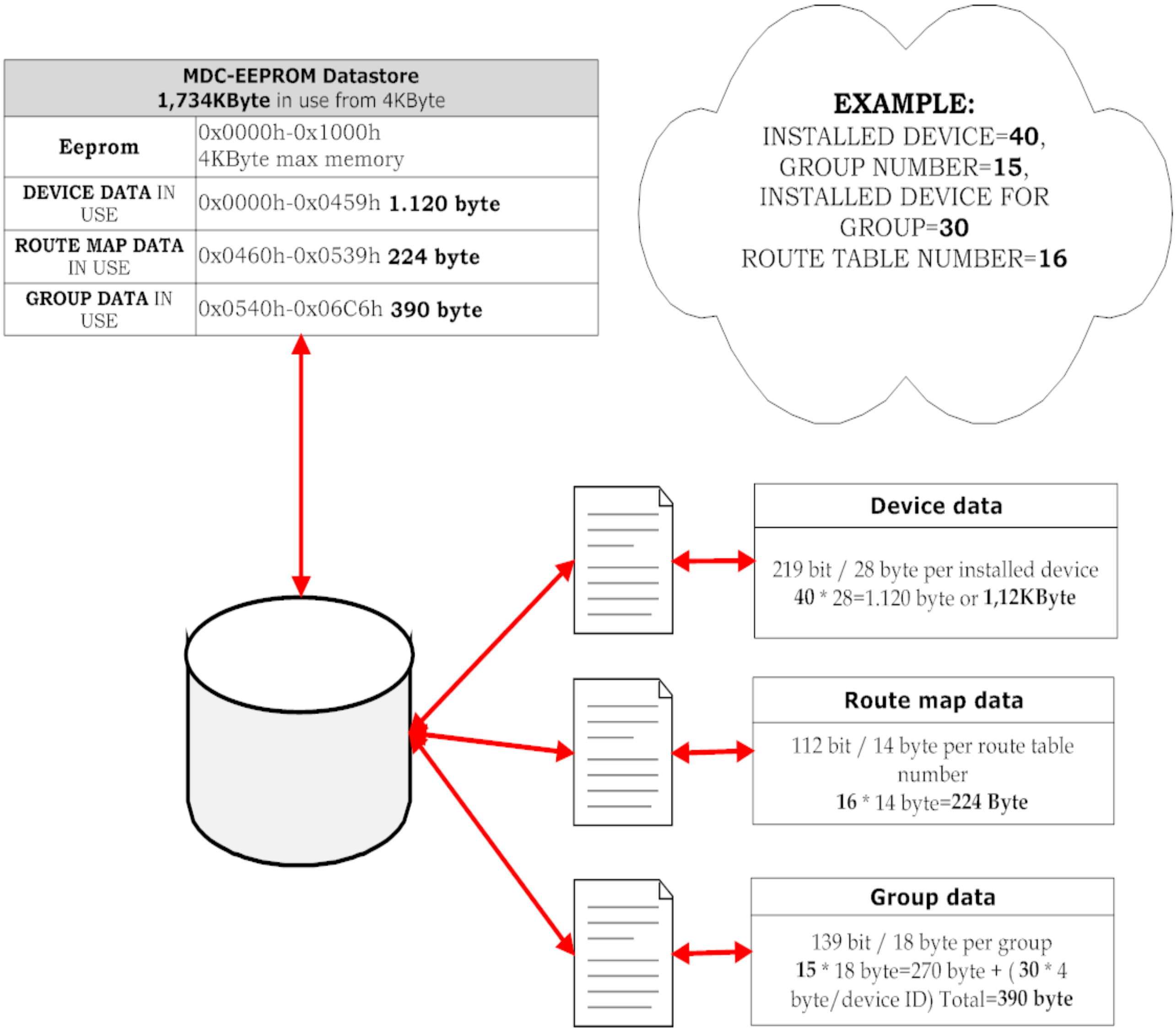


Figure 17 Example of MDC-Eeprom memory in use

3.1.9 Signaling Details

In this section will see the physical structure of data how the travel over the power line via wire or airline via electromagnetic waves. We focus to power line transmission because is the main subject of final thesis.

In this section will see:

- O.P.R.A packet structure
- O.P.R.A signaling

Packet structure

Message for power line network or wireless is split to packets each packet have a special function, the packet s over airline-wireless follow the rules of protocol IEEE 802.15.4 PHY, MAC.

Power line packet

STANDARD MESSAGE

Figure 18 Standard message packets

EXTENDED MESSAGE

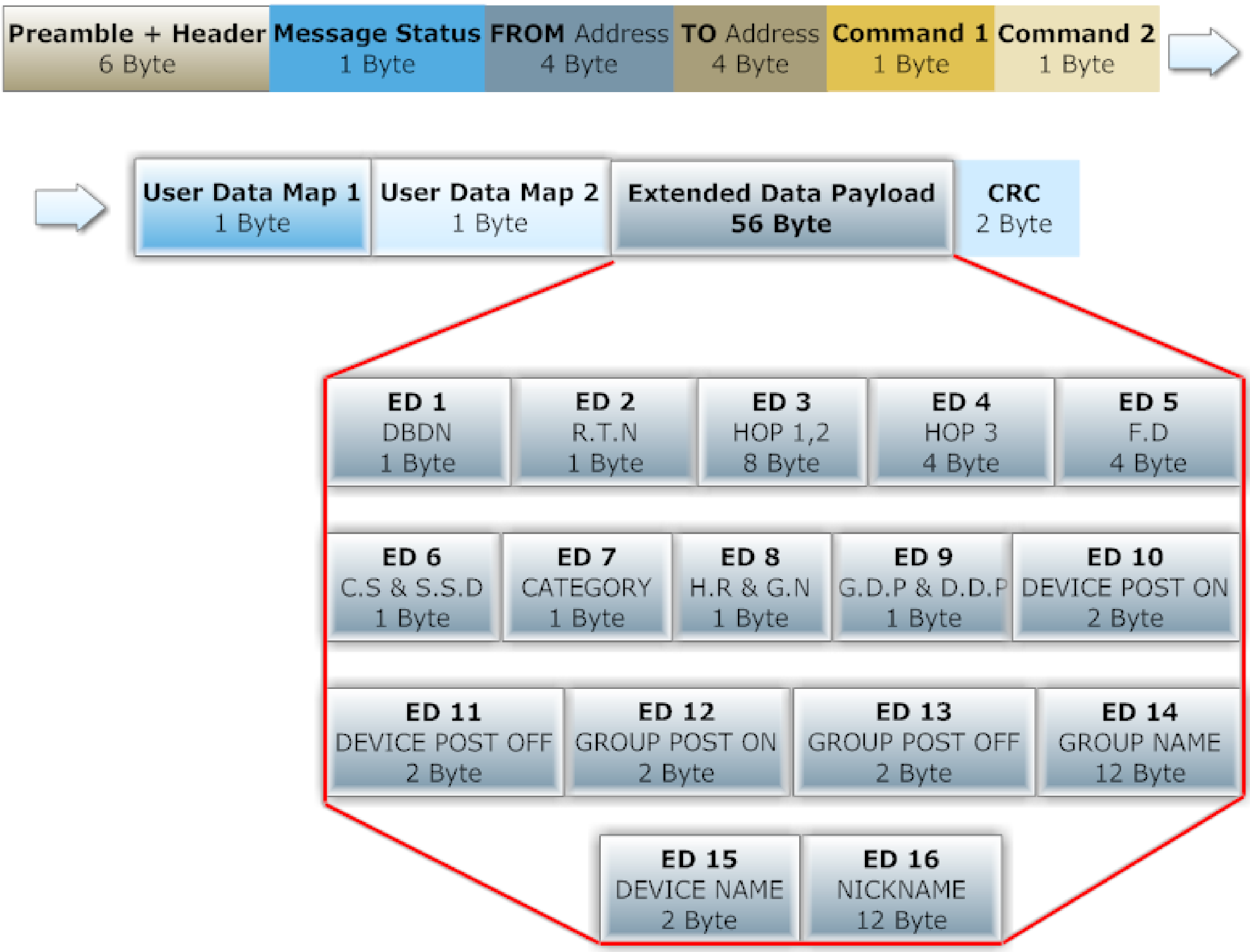


Figure 19 Extended message packets

- **Preamble** = 0xAAAAAAAA 4 bytes, A➔1010
- **Header** = [0x????] 2 bytes, header byte is written in ST7540 chip and is different for MDC, devices.

  Total bytes for standard message = **19**Bytes, total bytes for extended message = **77** Bytes.

IEEE 802.15.4 packet

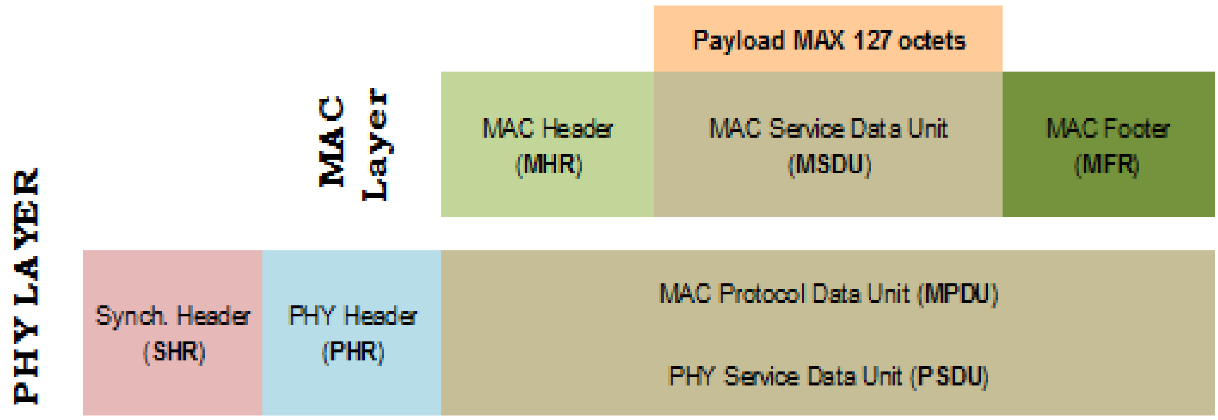


Figure 20 802.15.4 Frame structure (Jose Gutierrez 2003, IEEE 802.15.4 Tutorial, 21.)

PHY Protocol Data Unit (PPDU)			
Preamble Sequense	SFD	Frame length	PHY Payload
5 octets Synchronization Header (SHR)		1 octet (PHR)	max 127 octets PHY Service Data Unit (PSDU)
			MAC Protocol Data Unit (MPDU)

Figure 21 "IEEE 802.15.4 Frame Format – PHY Layer Frame Structure (PPDU)" (ATmega128RFA1 Datasheet, 62.)

For more details about the PHY Layer Frame Structure (PPDU) see in ATmega128RFA1 Datasheet, pages 62 to 63 from sources.

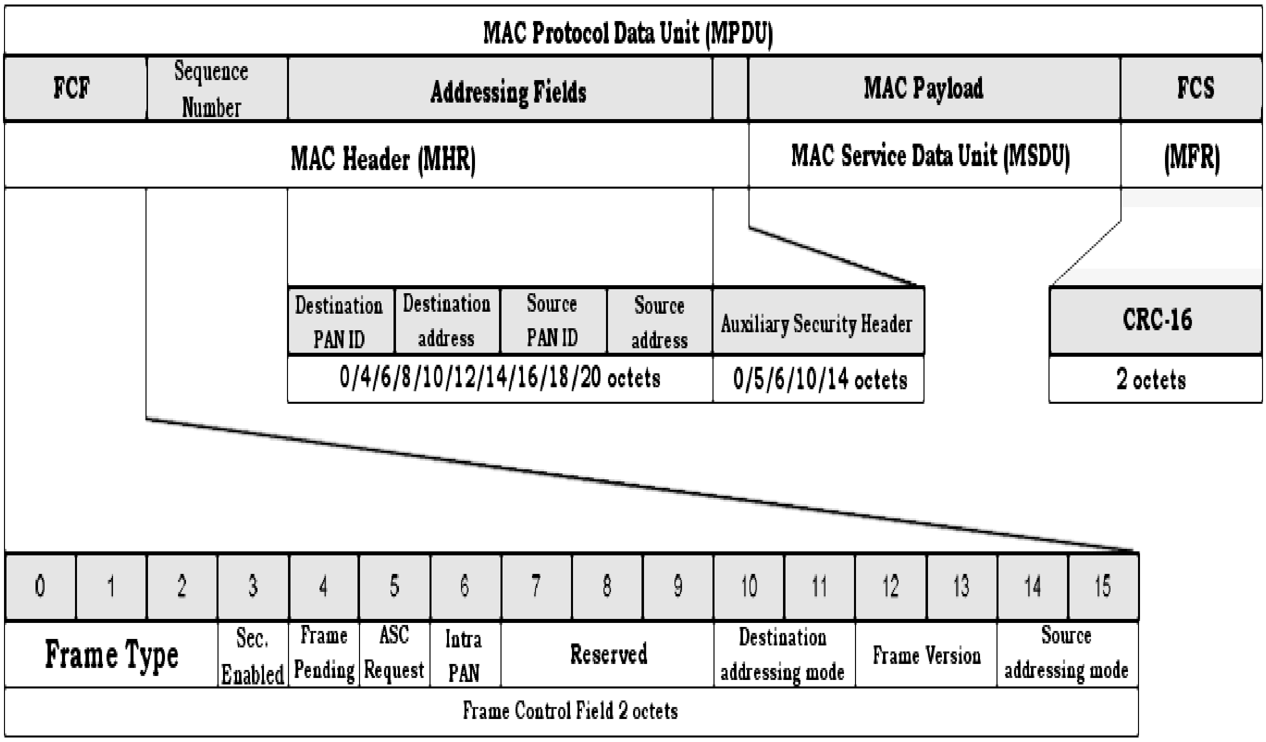
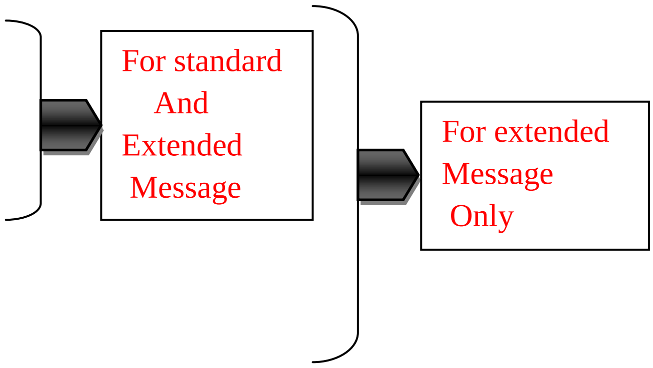


Figure 22 IEEE 802.15.4 Frame Format – MAC Layer Frame Structure (MPDU) (ATmega128RFA1 Datasheet, 63.)

For more details about the MAC Layer Frame Structure (MPDU) see in ATmega128RFA1 Datasheet, pages 64 to 66 from sources.

The **MAC payload** for standard and extended message has:

- Message Status
- FROM Address
- TO Address
- Command 1
- Command 2
- User Data Map 1
- User Data Map 2
- Extended Data 1-16



Signaling

Sending Signal on low voltage over electric network (power line 110/220 Volt) in the frequency range 3 kHz to 148,5KHz must follow some rules that some organization have set. One organization for setting rules is CENELEC (European Committee for Electro technical Standardizations) that have published a series of regulations about the communication on low voltage electrical installations and we can find that regulations in they own site (www.cenelec.eu). Some of CENELEC standards are: EN50065-1, EN50065-4-2, and EN50065-7.

For more details about noise, impedance, CENELEC standards check Giuseppe Cantone, AN1714 Application note, 2-4 from sources.

That we need is a way so we can send over electrical network a low-voltage signal to build house automation with O.P.R.A protocol in use.

FSK modulation by ST7540

The FSK (Frequency Shift Key) modulation convert a digital signal in to a sinusoidal signal that can take two values, one for logic ‘high’ and one for logic ‘low’. Sinusoidal signal take two different frequency values, fh for logic ‘high’ and fl for logic ‘low’. The average value for fh and fl is the carrier frequency fc. The difference between the two frequencies fh, fl is a function of the baud-rate (BAUD) (as we see from the equation below) of the digital signal (symbols transmitted in one second) multiply with the deviation.

The equations are (Giuseppe Cantone. AN1714 Application note, 7.) :

$$fh - fl = BAUD * dev, \qquad Fc = \frac{fh + fl}{2}$$

The function of the devices is to receive and transmit the message through power line connected with coupling circuit; electrical signals coded according a half duplex FSK modulation. Below figures show two examples of FSK modulation first show the spectrum analyzer of FSK modulation for Baud –rate=4800, deviation=1, Fc=132.5 KHz, f- low=134,928 KHz and f -high=130,046 KHz. The second figure shows a digital signal as input and the result after the FSK modulation, how frequency change when digital signal have logic ‘1’ and when ‘0’.

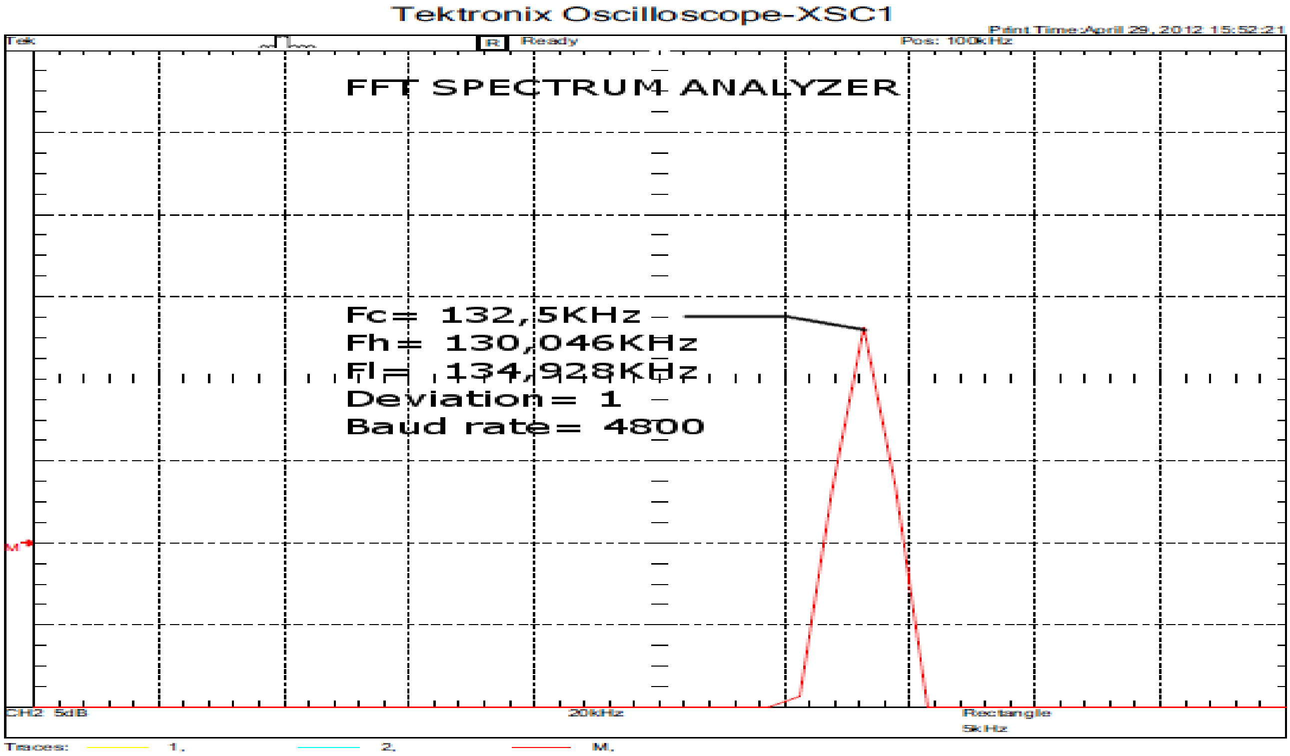


Figure 23 FSK spectrum analyzer of 132,5KHz Fcarrier

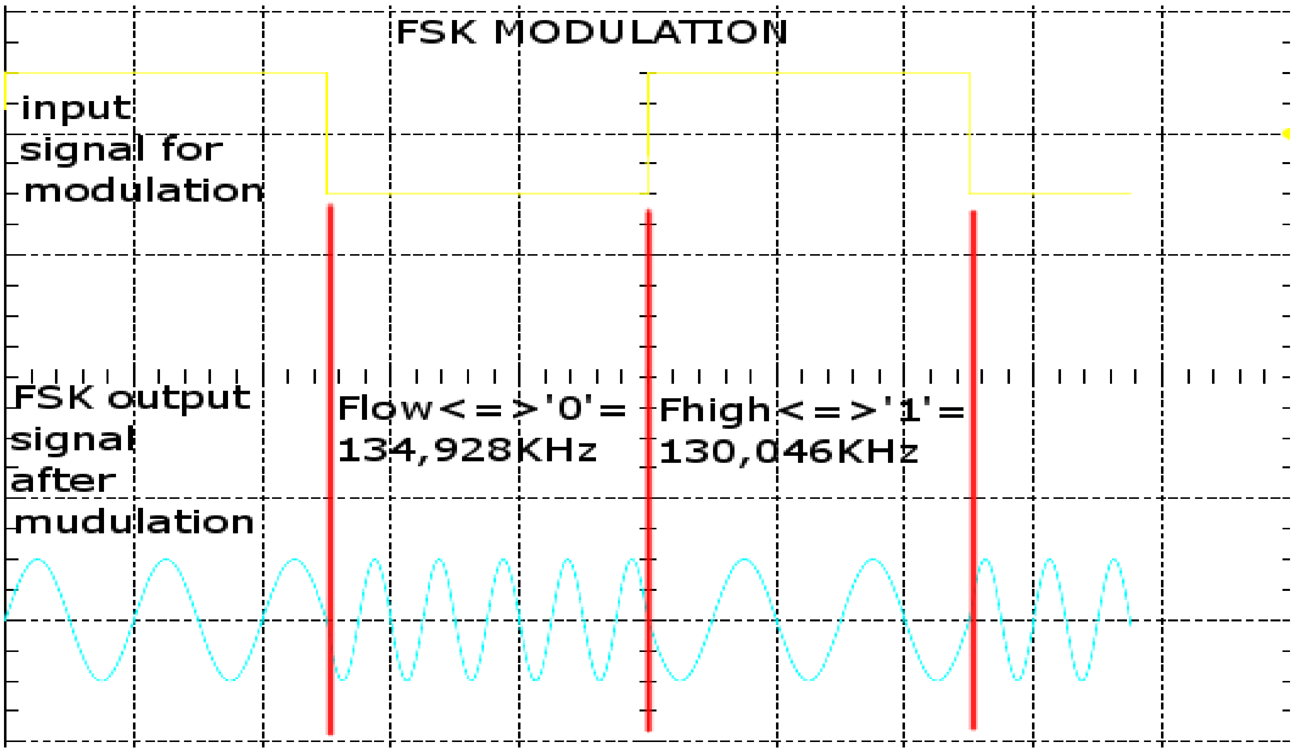


Figure 24 FSK modulation of 132,5KHz Fcarrier

4 REMOTE CONTROLS

4.1 Telecontrol

Telecontrol is one way to control devices inside a house or factory, have been designed for disable people with the main goal the help that we can offer. Simplicity is the main factor without to be complicate, is wireless using IEEE 802.15.4 protocol and that mean in any room inside the house user can control/check the whole house from alarm, oven, heat system, cool system, lights and more others. Have LCD screen with backlight for night and have touch buttons with programmable buttons for instant executable commands. Telecontrol is one device for home automation the other way is GUI (window program for computer) software for Windows xp-vista-7-8, Linux and Mac. One other way that the majority of population uses is cell-phone/ Smartphone (android, iphone, palm, windows and blackberry), with the right API software can work as remote control device.

4.1.1 Touch buttons (QMatrix) theory

For touch buttons have been used material from Atmel Corporation. Atmel Company has created controllers that support touch sensors; touch solution is split to **buttons**, **sliders** and **wheels**. For touch buttons is split to single button, up to 10 buttons and more than 10 buttons.

For **single buttons** they have: AT42QT1010 QTouch 1-Button Sensor IC with Max On, AT42QT1011 QTouch 1-Button Sensor IC with Touch On and AT42QT1012 QTouch 1-Button Sensor IC with Power Management Functions. (Atmel, single button.)

For **up to 10 buttons** they have: AT42QT1040 QTouch 4-Button Sensor IC, AT42QT1060 QTouch 6-Button Sensor IC, AT42QT1070 QTouch 7-Button Sensor IC with Optimized Architecture, AT42QT1085 QTouch 8-Button Sensor IC with Haptics. (Atmel, up to 10 buttons.)

For **more than 10 buttons** they have: AT42QT1110 Atmel QTouch 11-Button Sensor IC, AT42QT1111 QTouch 11-Button Sensor IC with 1.8 to 5.5 Operating Voltage, AT42QT1481 Atmel QMatrix 48-Button Sensor IC with EN/IEC 60730 certification & FMEA support, AT42QT2120 QTouchADC 12-channel Sensor IC with Long-Range Proximity Capability and Slider/Wheel, QT60160 Atmel QMatrix 16-Button Sensor IC, QT60168 QMatrix 16-Button Sensor IC, QT60240 QMatrix 24-Button Sensor IC, QT60248 QMatrix 24-Button Sensor IC, QT60326 QMatrix 32-Button Sensor IC, QT60486 QMatrix 48-Button Sensor IC.
(Atmel, more than 10 buttons.)

For **sliders and wheels** they have: AT42QT2100 QTouch 10-channel Sensor IC (up to 7 buttons and 3 for Slider or Wheel), AT42QT2120 QTouchADC 12-channel Sensor IC with Long-Range Proximity Capability and Slider/Wheel, AT42QT2160 QMatrix 16-Channel Sensor IC (Slider from 2 to 8 Channels). (Atmel, sliders and wheels.)

Atmel corporation use a technique for making all Atmel MCUs, AT91SAM, tinyAVR, megaAVR, XMEGA, UC3A, and UC3B be QTouch using Library (API software), with that way all MCUs can support up to 64 sense channels for maximum interface sensitivity; 256-level sliders and wheels require only three channels. QTouch Library supports three capacitive touch acquisition methods: QTouch, QTouchADC and QMatrix. The tools for using QTouch Library are QTouch Library 4.4, Atmel AVR studio 5 and QTouch Studio 4.3.1 from Atmel site that can be downloaded after a registration.

From that moment we will choose Atmel MCU and how many buttons we want to have for use this is the first step for knowing which port of MCU we use for QTouch Matrix and what electronic components need to use and connect. When the first part ends start the second part which is the theory behind QTouch Matrix which mean the tool for making in PCB board these buttons reality. Final step is to configure the QTouch Library write the software and programming the MCU.

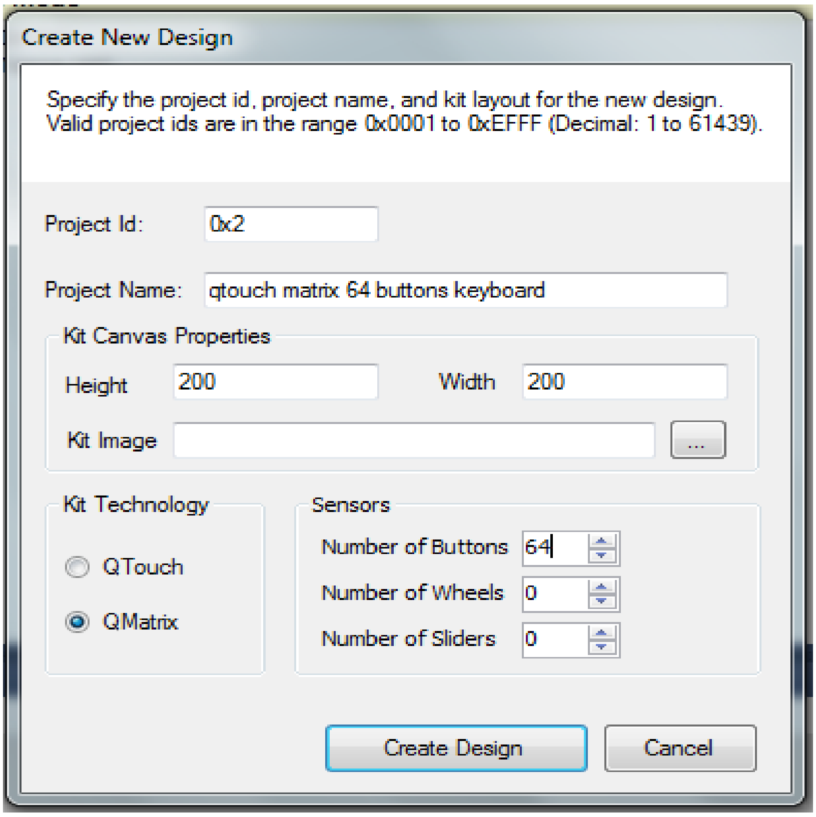


Figure 25 AVR QTouch Studio-New design

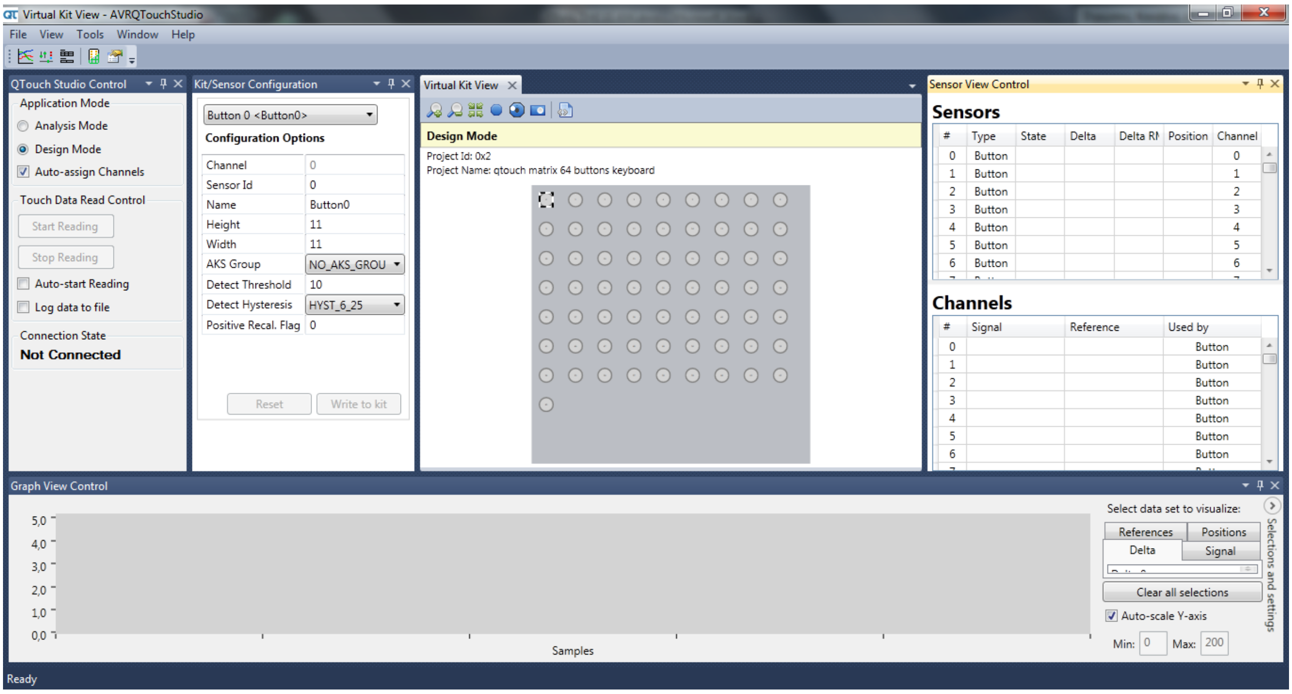


Figure 26 AVR QTouch Studio-Environment window

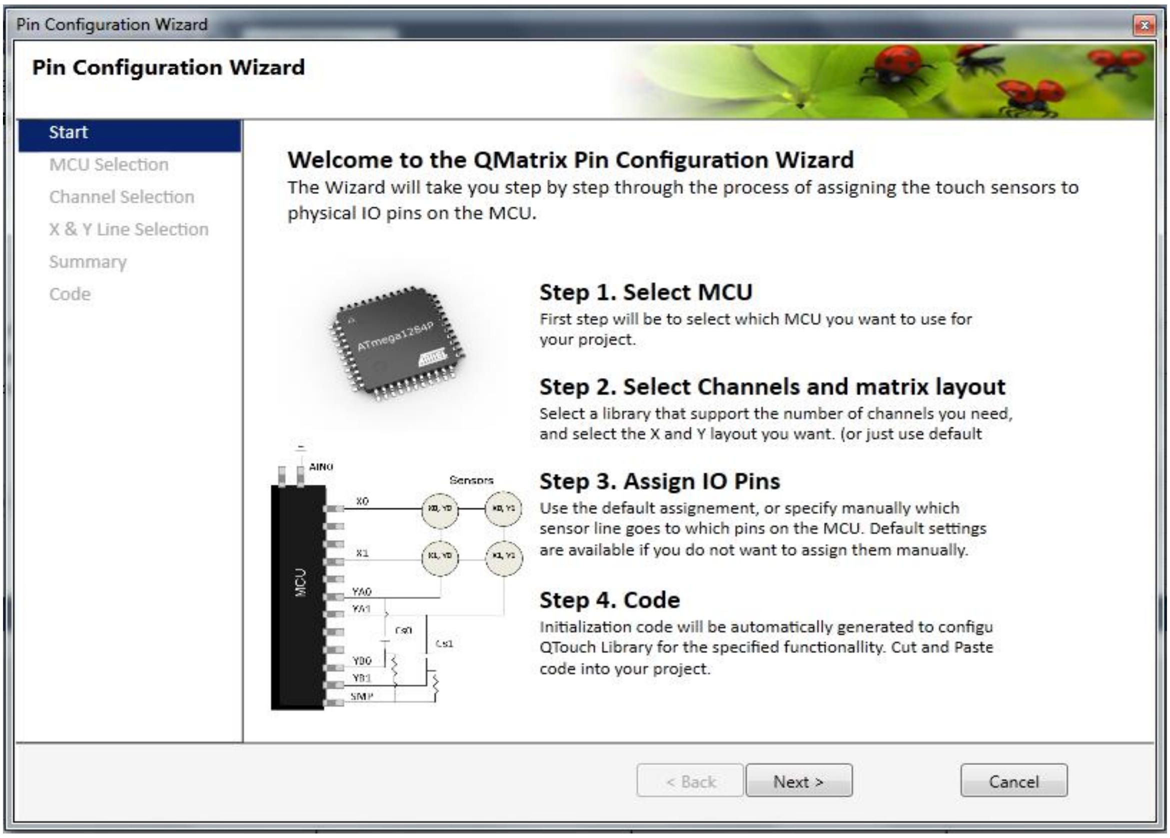


Figure 27 AVR QTouch Studio-Pin configuration wizard

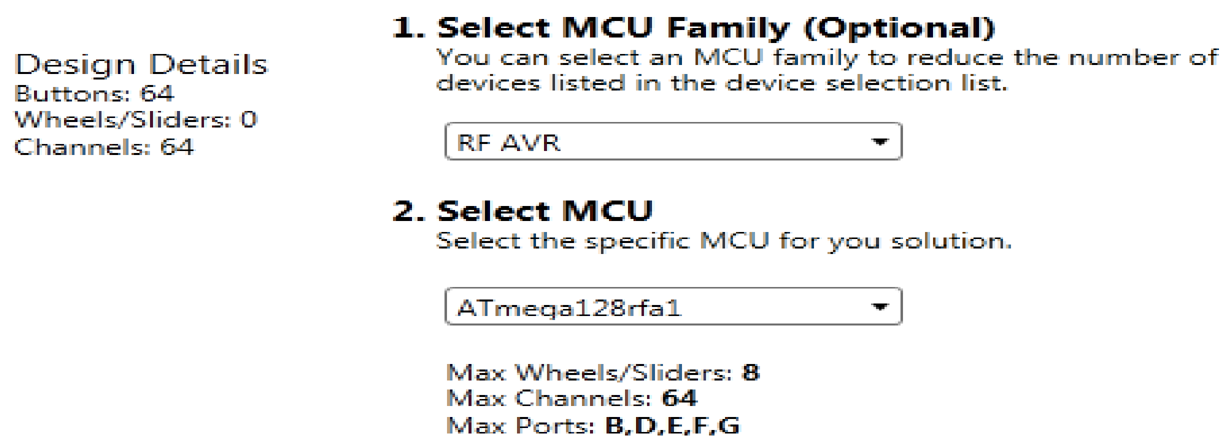


Figure 28 AVR QTouch Studio-MCU family

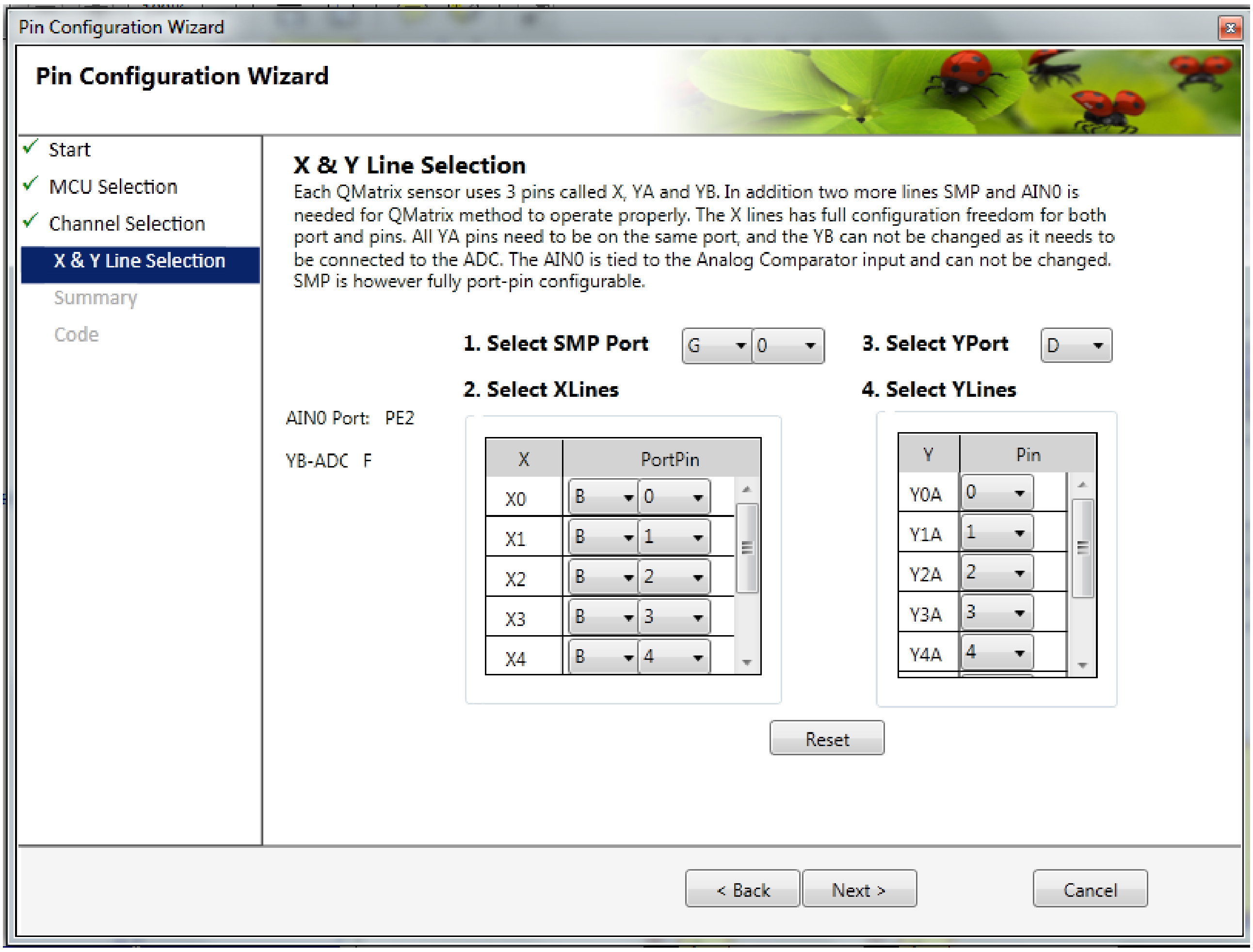


Figure 29 AVR QTouch Studio-MCU X & Y lines over MCU ports

MCU : **ATmega128rfa1**			Kit Technology : **QMatrix**			Max Channels Supported : **64**		
X & Y Lines Config : **8x8**			SMP : **PG0**			AI NO : **PE2**		
X0:PB0		Y0A:PD0		Y0B:PF0				
X1:PB1		Y1A:PD1		Y1B:PF1				
X2:PB2		Y2A:PD2		Y2B:PF2				
X3:PB3		Y3A:PD3		Y3B:PF3				
X4:PB4		Y4A:PD4		Y4B:PF4				
X5:PB5		Y5A:PD5		Y5B:PF5				
X6:PB6		Y6A:PD6		Y6B:PF6				
X7:PB7		Y7A:PD7		Y7B:PF7				
Channel	**X**	**YA**	**YB**		**Channel**	**X**	**YA**	**YB**
Channel0	PB0	PD0	PF0		Channel32	PB0	PD4	PF4
Channel1	PB1	PD0	PF0		Channel33	PB1	PD4	PF4
Channel2	PB2	PD0	PF0		Channel34	PB2	PD4	PF4
Channel3	PB3	PD0	PF0		Channel35	PB3	PD4	PF4
Channel4	PB4	PD0	PF0		Channel36	PB4	PD4	PF4
Channel5	PB5	PD0	PF0		Channel37	PB5	PD4	PF4
Channel6	PB6	PD0	PF0		Channel38	PB6	PD4	PF4
Channel7	PB7	PD0	PF0		Channel39	PB7	PD4	PF4
Channel8	PB0	PD1	PF1		Channel40	PB0	PD5	PF5
Channel9	PB1	PD1	PF1		Channel41	PB1	PD5	PF5
Channel10	PB2	PD1	PF1		Channel42	PB2	PD5	PF5
Channel11	PB3	PD1	PF1		Channel43	PB3	PD5	PF5
Channel12	PB4	PD1	PF1		Channel44	PB4	PD5	PF5
Channel13	PB5	PD1	PF1		Channel45	PB5	PD5	PF5
Channel14	PB6	PD1	PF1		Channel46	PB6	PD5	PF5
Channel15	PB7	PD1	PF1		Channel47	PB7	PD5	PF5
Channel16	PB0	PD2	PF2		Channel48	PB0	PD6	PF6
Channel17	PB1	PD2	PF2		Channel49	PB1	PD6	PF6
Channel18	PB2	PD2	PF2		Channel50	PB2	PD6	PF6
Channel19	PB3	PD2	PF2		Channel51	PB3	PD6	PF6
Channel20	PB4	PD2	PF2		Channel52	PB4	PD6	PF6
Channel21	PB5	PD2	PF2		Channel53	PB5	PD6	PF6
Channel22	PB6	PD2	PF2		Channel54	PB6	PD6	PF6
Channel23	PB7	PD2	PF2		Channel55	PB7	PD6	PF6
Channel24	PB0	PD3	PF3		Channel56	PB0	PD7	PF7
Channel25	PB1	PD3	PF3		Channel57	PB1	PD7	PF7
Channel26	PB2	PD3	PF3		Channel58	PB2	PD7	PF7
Channel27	PB3	PD3	PF3		Channel59	PB3	PD7	PF7
Channel28	PB4	PD3	PF3		Channel60	PB4	PD7	PF7
Channel29	PB5	PD3	PF3		Channel61	PB5	PD7	PF7
Channel30	PB6	PD3	PF3		Channel62	PB6	PD7	PF7
Channel31	PB7	PD3	PF3		Channel63	PB7	PD7	PF7
Recommended Library :			IAR : libm128rfa1_64qm_8x_8y_k_0rs.r90					
			GCC : libm128rfa1_64qm_8x_8y_k_0rs.a					

Figure 30 AVR QTouch Studio-Summary

Code of QTouch Matrix 64 buttons:

```
#define _QMATRIX_ 1
#define QT_NUM_CHANNELS       64
#define PORT_X_1    B
#define PORT_YA     D
#define PORT_YB     F

#define PORT_SMP   G
#define SMP_PIN      0

#define NUM_X_LINES       8
#define NUM_Y_LINES       8

#define NUM_X_PORTS      1
#define PORT_NUM_1         1

x_line_info_t x_line_info[NUM_X_LINES]= {
FILL_OUT_X_LINE_INFO(1,0);
FILL_OUT_X_LINE_INFO(1,1);
FILL_OUT_X_LINE_INFO(1,2);
FILL_OUT_X_LINE_INFO(1,3);
FILL_OUT_X_LINE_INFO(1,4);
FILL_OUT_X_LINE_INFO(1,5);
FILL_OUT_X_LINE_INFO(1,6);
FILL_OUT_X_LINE_INFO(1,7);
};

y_line_info_t y_line_info[NUM_Y_LINES]= {
FILL_OUT_Y_LINE_INFO(0);
FILL_OUT_Y_LINE_INFO(1);
FILL_OUT_Y_LINE_INFO(2);
FILL_OUT_Y_LINE_INFO(3);
FILL_OUT_Y_LINE_INFO(4);
FILL_OUT_Y_LINE_INFO(5);
FILL_OUT_Y_LINE_INFO(6);
FILL_OUT_Y_LINE_INFO(7);
};
```

Atmel QTouch controllers using one of two capacitive measurement types:

“Self-capacitance sensors” For more information about the sensor check QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, page 2.

“Mutual-capacitance sensors” For more information about the sensor check QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, page 2.)

Capacitor calculation (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 3.) :

- Cx: Capacitance of the electrode, it is separate from any other parasitic capacitance
- e_r: Is the relative dielectric constant of the panel material
- e_0: Is the free space capacitance that can exist and counting per meter, defined as 8.85 x 10-12 F/m
- T: Is the thickness of the panel in meters
- A: Is the area of the touched region in square meters
- Cp: Is the parasitic capacitance
- SNR: Is the signal to noise ratio

$$C = \frac{e_0 * e_r * A}{T}$$

For more details about thickness of panel and how capacitance can be affected from thickness of the panel check QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, pages 3, 11 from sources.

Sensitivity Factor (S) (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 11.) :

$$S = \frac{e_r}{t}$$

- t is the thickness of the layer in question
- e_r is the dielectric constant of the layer

Table 65 Front panel material dielectric constant (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 10-11.)

Material	Dielectric constant (e_r)
Air	1.00059
Glass	3.7 to 10
Plexiglas	3.4
Polycarbonate	2.9 to 3.0
Nylon	3
PET (Polyethylene Terephthalate)	3
Polystyrene	2.56
Polyethylene	2.2 to 2.4
FR4 (Glass Fiber + Epoxy)	4.2

Common shapes for electrode are filled discs, squares and rectangles. (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 16.)

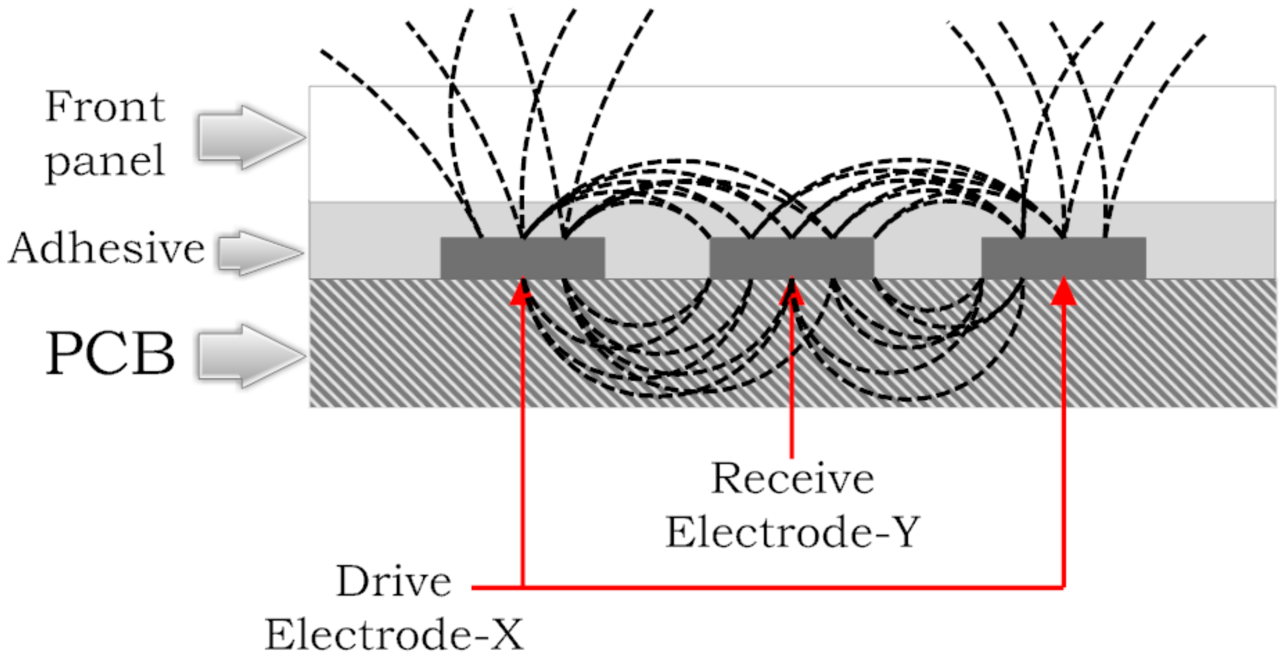


Figure 31 QMatrix touch button field (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 24.)

QMatrix contains two electrodes one for X electrode (as transmitter) and one for Y electrode (as receiver). The sensitive touch area is the gap between the X and Y electrodes.

How touch sensitivity can be affected by factors for QTouch/QMatrix:

- Electrode size and design
- Dielectric front panel thickness and material
- Ground loading and other signals
- Ground return
- Supply voltage
- Detection threshold (negative threshold – NTHR)
- Sampling capacitor (Cs)
- Burst length
- QMatrix sampling resistor (Rsmp)

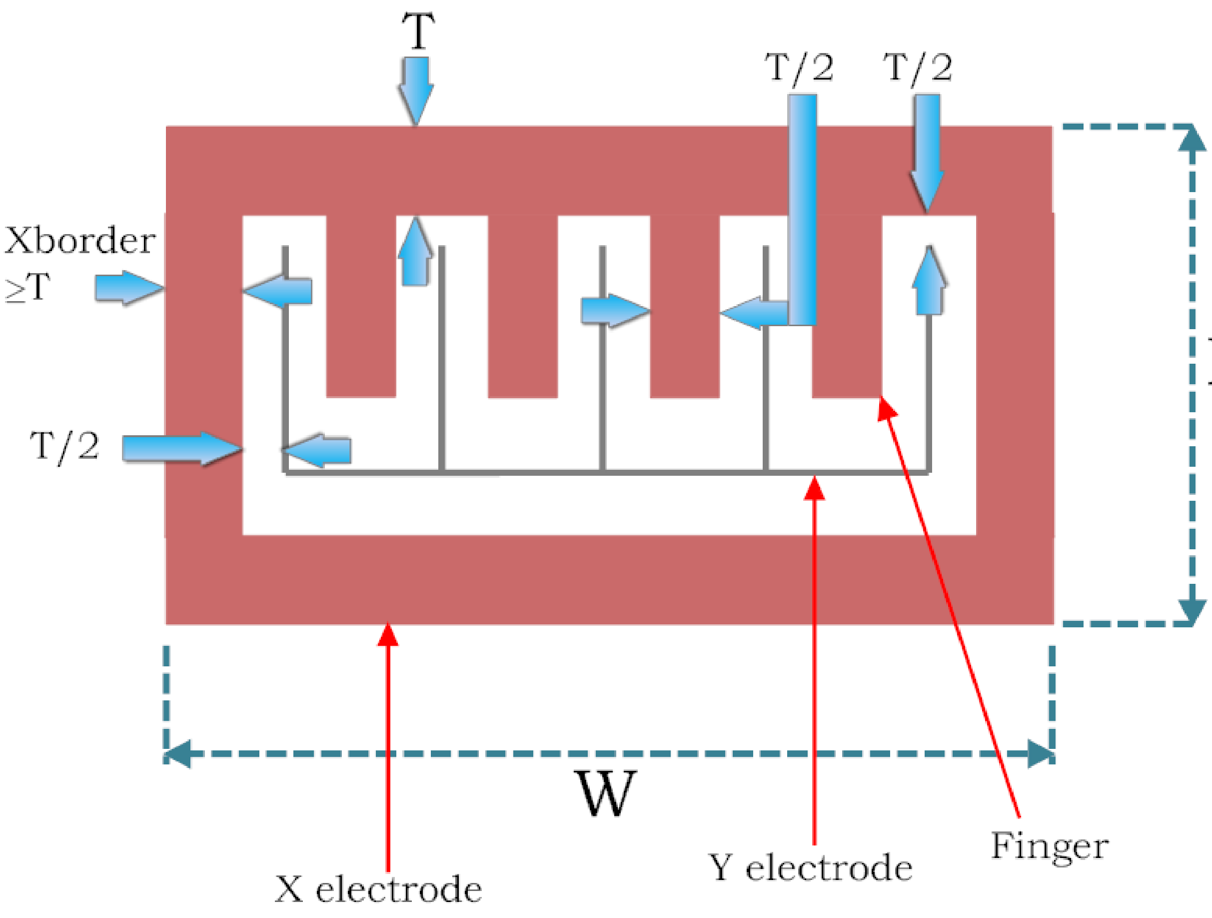


Figure 32 QMatrix touch button mechanical details (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 25.)

For more information about the X, Y electrodes width and details check QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, pages 25-28 from sources.

Calculation for Xfingers and Xborder (QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide, 25.) :

T = Front panel thickness

Xfingers = └ (W - 3T - Ywidth) / (1.5T + Ywidth) ┘

Xborder = (W - T - Ywidth - Xfingers (1.5T + Ywidth)) / 2

Example calculation:

- T = 2 mm
- W = 15 mm
- H = 15 mm
- Ywidth=0.3 mm

Xfingers = └ (15 - 6 – 0.3) / (3+ 0.3) ┘

= └ (8.7) / (3.3) ┘

= └ 2.6 ┘

= 2

Xborder = (15 - 2 – 0.3 - 2 (3 + 0.3)) / 2

= 6.1/2

= 3.05mm

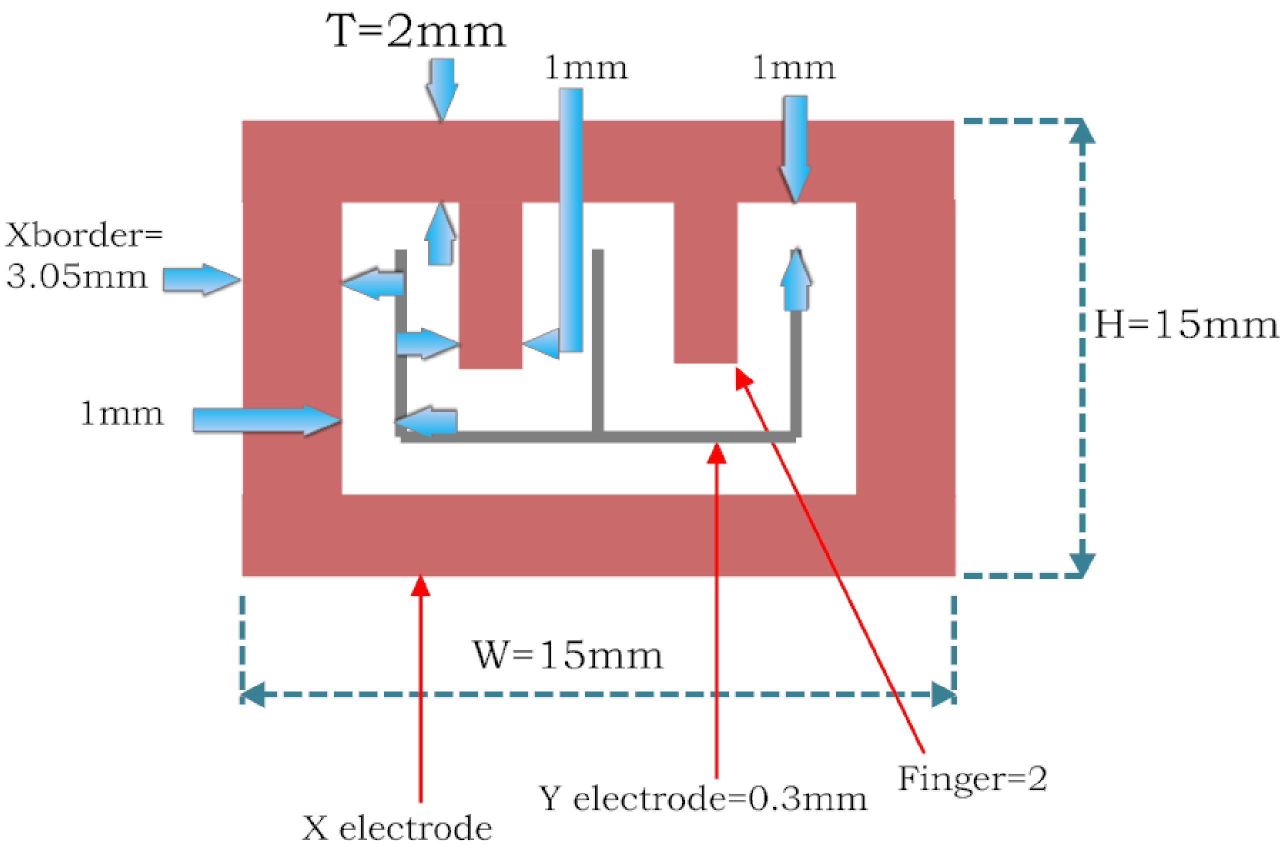


Figure 33 QMatrix touch button example

When from theory and papers must move to design and PCB creation there is a problem of wasting lot of time to add each mechanical detail for touch buttons and also some human mistake can be added upon PCB when the engineer design it. For that Atmel and Altium corporations have collaborate together to bring the solution for that problem. Atmel have created Library (QTouch Library) for all touch solutions (buttons, sliders, wheels) for Altium designer software (PCB design). For now on is a piece of cake when someone wants to use touch buttons, wheels, and sliders, draw it in to schematic and from schematic transferring it to PCB. Each touch solution is configurable for example the height, width, front panel thickness and more other, after configuration with one command can transfer the whole schematic to PCB board and all the details of touch solution find it in PCB board without any problem. Altium designer software is a powerful tool for all electronic engineers that design and build PCB boards, some steps for adding and building touch buttons, sliders and wheels can check it in figures below. That was the end how to calculate-design touches buttons/sliders/wheels for all the phases (from paper to PCB board).

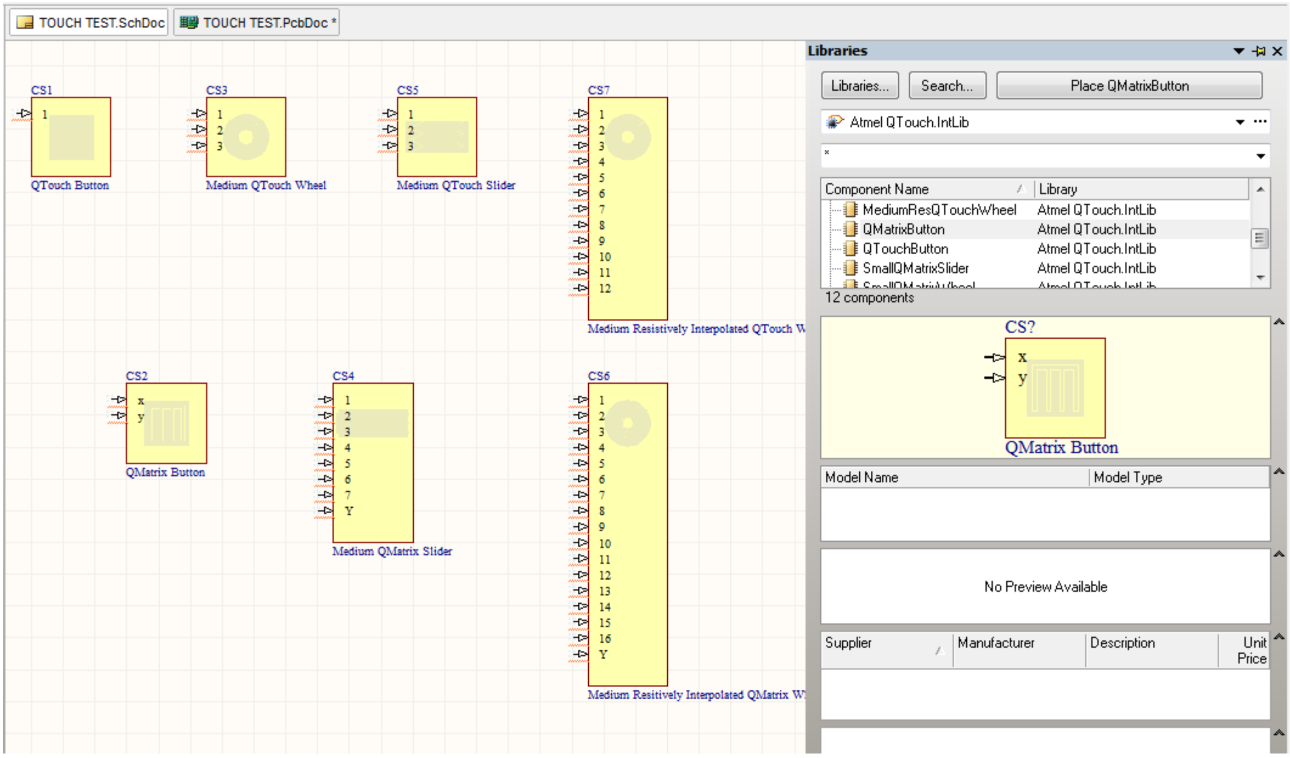


Figure 34 Altium Designer 10 – QTouch Library

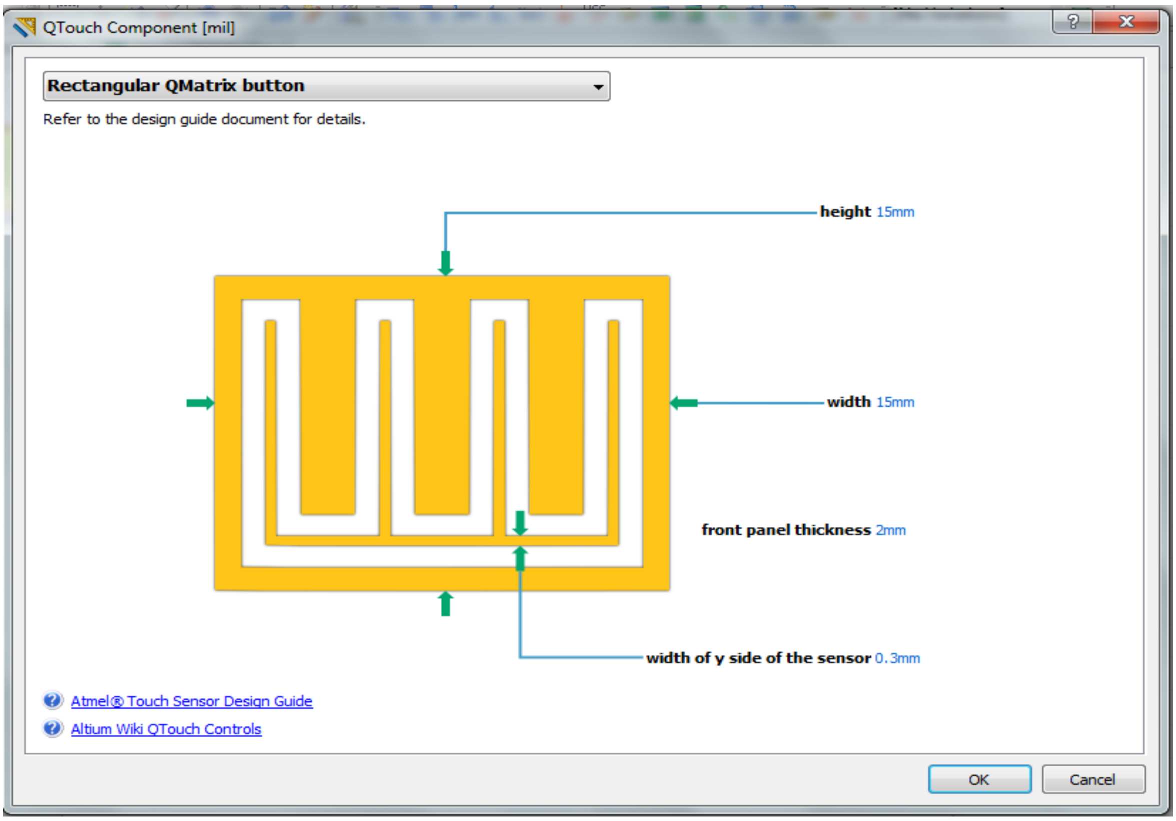


Figure 35 Altium Designer 10 – QTouch component configuration

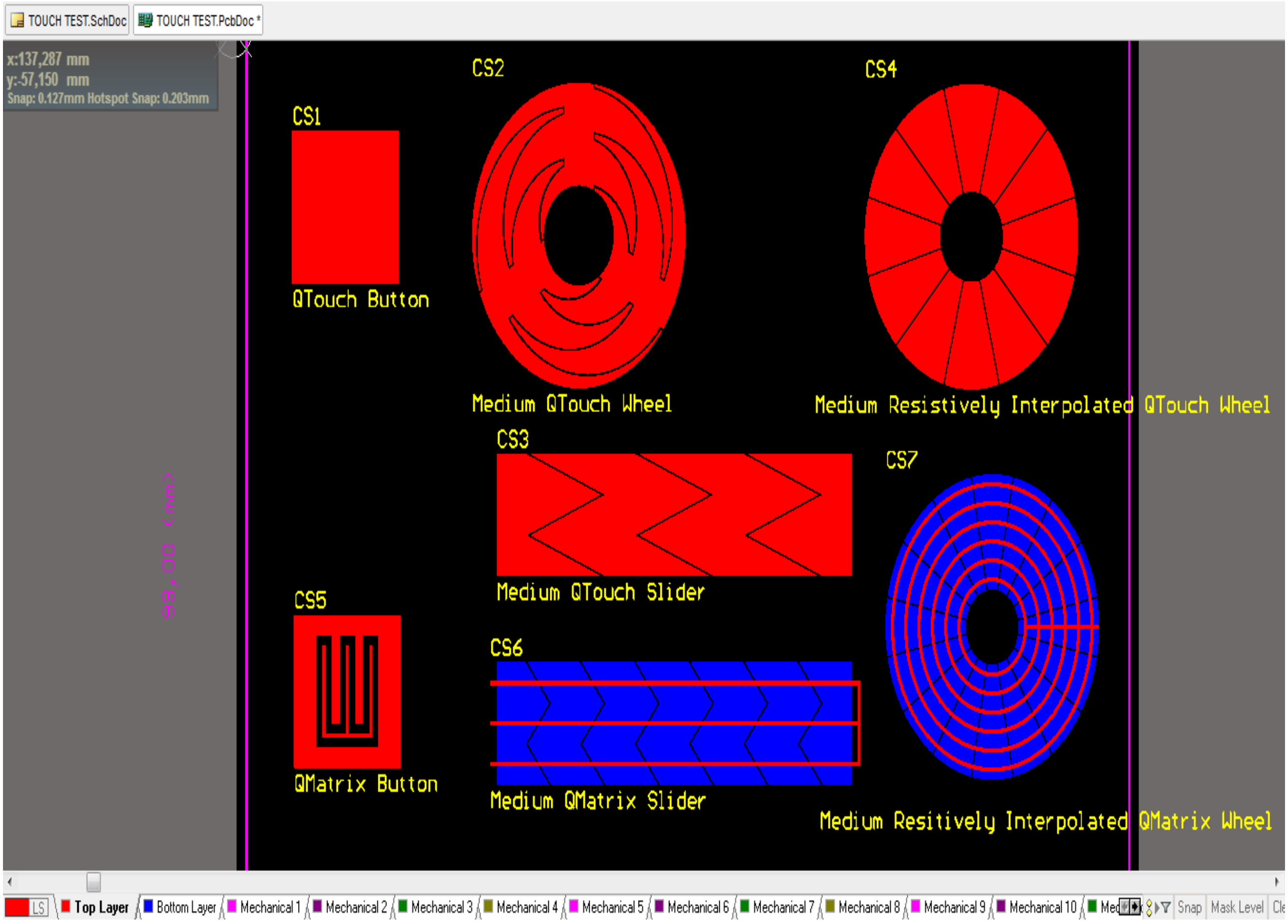


Figure 36 Altium Designer 10 – PCB board top and bottom layer components

4.1.2 Keyboard

PCB Keyboard use 64 buttons, 48 buttons for main functions and commands and 16 programmable buttons. The purpose of programmable buttons is for the user create macros and save it in each button, macros is commands that user can compose it and make the control of device more easy and faster without need to press much buttons to transmit the message. Programmed buttons is 16 and combination with num number 9 in number, user can save (16*9) = 144 programmable macros. 48 main function buttons is split to categories and each category has a number of buttons for a special function.

Main and sub- categories from 48 buttons are:

- **Alphanumeric Keypad [18 buttons]**
  - Enter
  - Backspace
  - Space
  - Caps lock on/off
  - Num on/off
  - Symbol on/off
  - Alphanumeric [num, ABC/abc, symbol] X 12 times
- **Control keys [7 buttons]**
  - Menu
  - Up
  - Down
  - Left
  - Right
  - Ok
  - Exit
- **Programming keys [3 buttons]**
  - Add shortcut key
  - Delete shortcut key
  - Shortcut on/off
- **Device manager [4 buttons]**
  - Add device
  - Delete device
  - Configure device
  - Repair ID
- **House area manager [2 buttons]**
  - Add new area
  - Remove house area
- **Group manager [3 buttons]**
  - Create group
  - Delete group
  - Setup group
- **Find by [5 buttons]**
  - Nickname
  - Category
  - Group
  - House area
  - Channel

- **Control devices [6 buttons]**
  - On
  - Off
  - Dimming on
  - Command
  - Panic button
  - Sent message

Alphanumeric Keypad:

When num button is on keypad work with these 1 2 3 4 5 6 7 8 9 0 * #
When symbol button is on keypad work with these $ @ ^ < > { } ~ % + = [ ]
When num and symbol is off keypad work with these ( ) & ; ? \ ! - _ " : , . a b c d e f g h j k l m n o p q r s t u v x y z (when caps lock is on letters is capital).

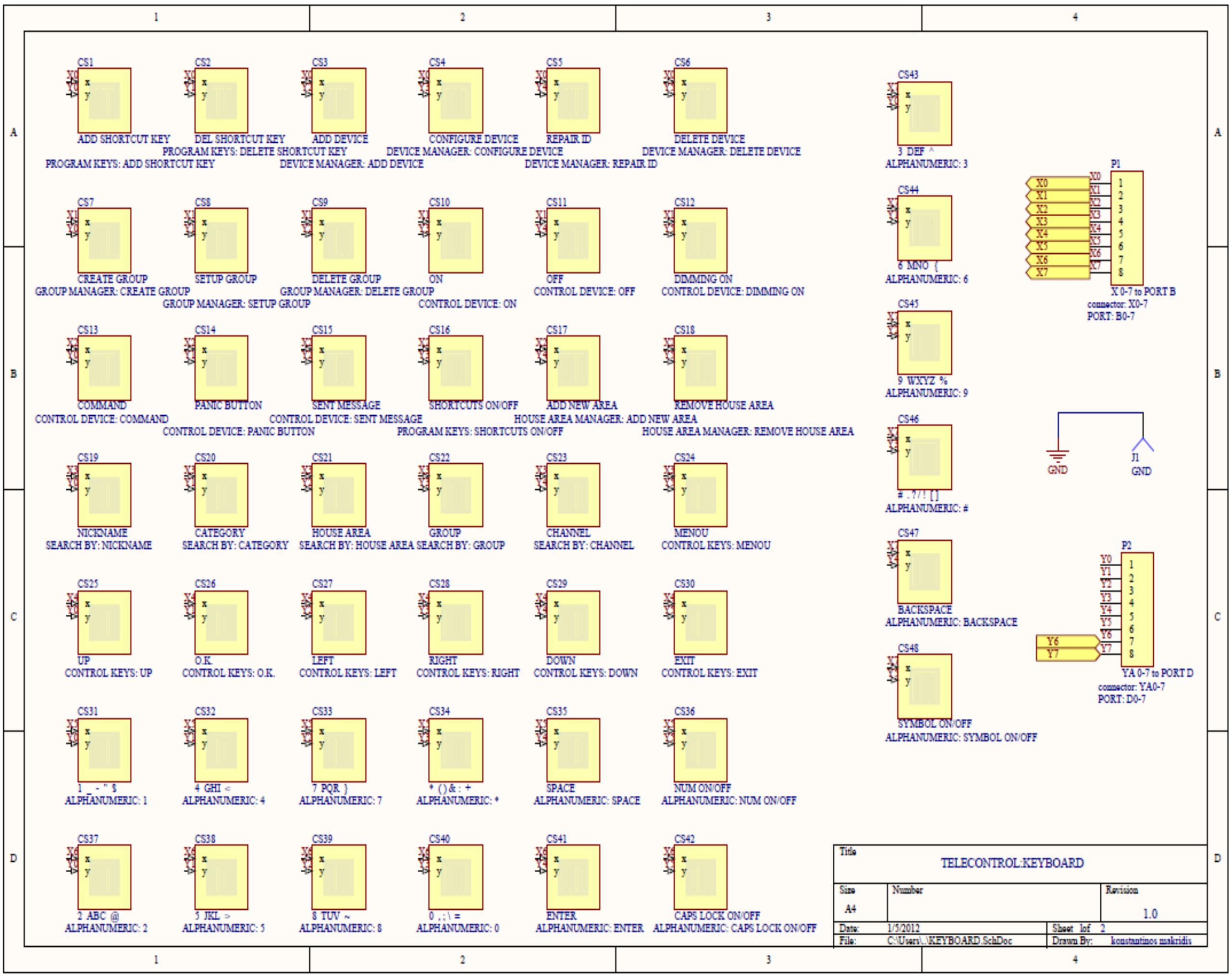


Figure 37 Telecontrol-keyboard 48 buttons schematic

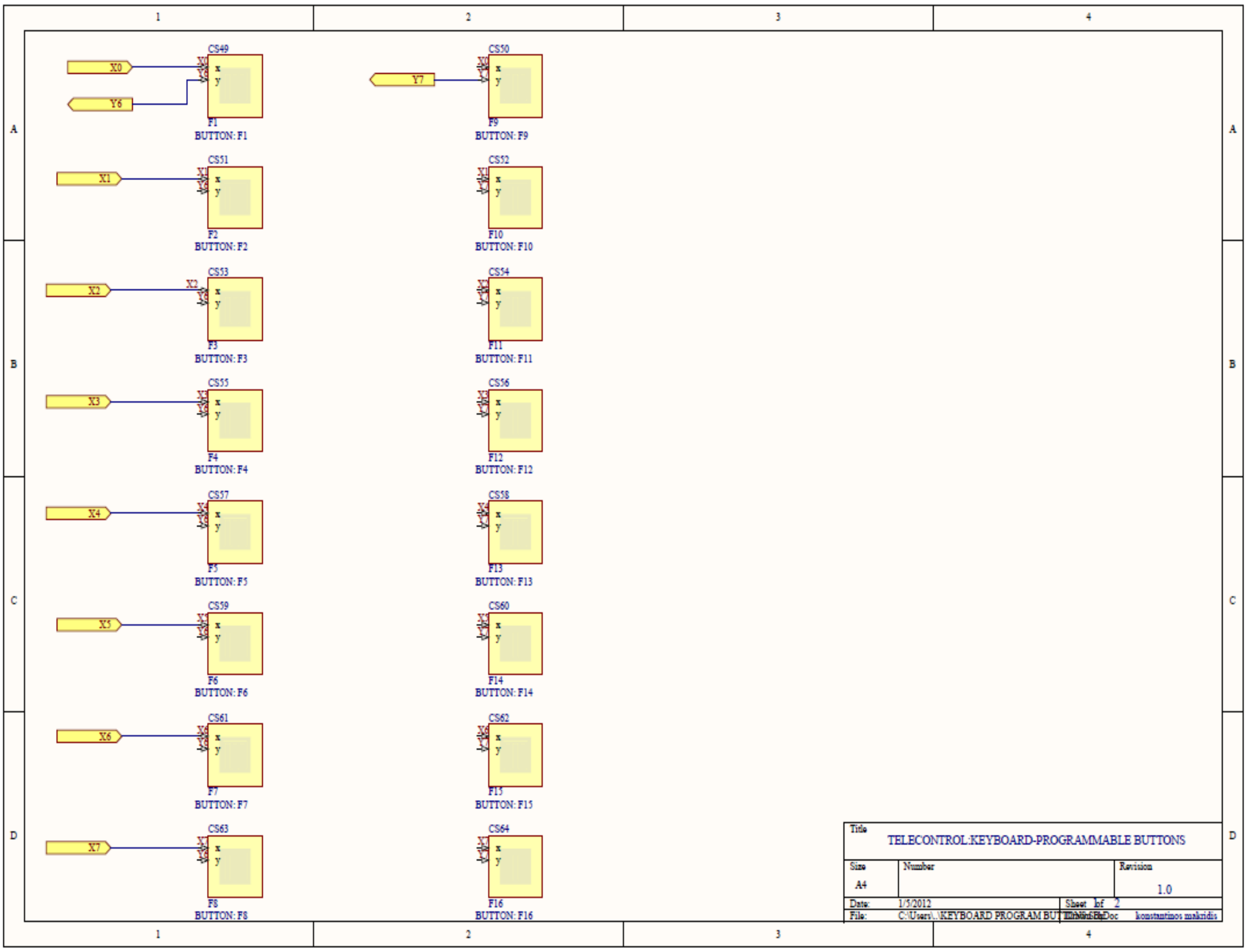


Figure 38 Telecontrol-keyboard 16 programmable buttons schematic

When user want to **create a macro** of a special command the steps to do it is: he must press 'add shortcut key', insert the programmable buttons from F1 to F16 and then insert from num keypad the number to make the combination, for example F1-5 after that insert the command he want and when have finish press 'O.K' to end the procedure. When user want to use the macro that already made must have the button 'shortcut =on' and choose the programmable button and then the number and press 'sent message'. **Example** for making macro: user wants to turn on from multiple socket (4 positions) only the 3 position. **First step** pressing 'add shortcut key=F1' **second step** adding number for combination num=5 then **third step** finding the device he want to control, for that it press 'category=SO' from all sockets that exist he choose the multiple socket 4 position then he press 'O.K', now that user knows the device he want to control is time for adding command, **fourth step** is to press 'command' , from all the commands that device support he choose 'power on (.3)' as main command and as subcommand he choose 'fast on' and press 'O.K', with that he finish the creation of a macro.

When user want to **add a device** the steps to do it is: pressing ’add device’ user must add the unique device ID, is a 8 character from 0 toF (0000-1111) in hex form, for example [024f9ebf] then pressing enter. After that it goes to configuration tab when user must fill data like nickname, device default state, schedule on/off and some other, when user end from insert details then pressing ‘O.K’ finishing the procedure.

When user want to **delete a device** the steps to do it is: pressing ‘delete device’ then user must find the device that is for delete, choosing device by category, nickname, and channel, group and house area. When user finds the device for delete it is time for pressing ’O.K’ and the procedure finish.

When user want to **create a group** the steps to do it is: pressing ‘create group’ then user must fill group number one from 15 numbers that exist for groups and write ‘group name’. After filling the details user press ‘O.K’ and it moves to setup tab to configure the group that user have create. Setup tab have group default state, group schedule on/off, group default state on/off and some others when write all of them user pressing ‘O.K’ and finish the procedure.

When user want to **delete a group** the steps to do it is: pressing ‘delete group’ from 15 groups that exist user choose the one for delete and then pressing ‘O.K’ and finish the procedure.

Menu has details about wireless connection, details for devices, configuration for telecontrol and for MDC. Some settings are: factory settings, touch buttons configuration, IEEE 802.15.4- 2.45GHz wireless connection status, real time clock setup and much more.

4.1.3 Mainboard

Mainboard of telecontrol is the base when all the electronics components will be installed and when keyboard PCB board be connected. Mainboard have ATMEL MCU ATmega128RFA1, capacitors/resistors for QTouch Library, headers/receptacles for connections like ISP programmer, Keyboard, power supply, LCD screen 16x4, Antenna, LED and push buttons. Most parts are smd except parts that is through hole like connectors. There is two push buttons one for reset MCU ATmega128RFA1 and second to lock and put in low power consumption mode telecontrol device. There is a trimmer to adjust contrast of LCD screen. Also there is a smd led for lighting keyboard PCB board.

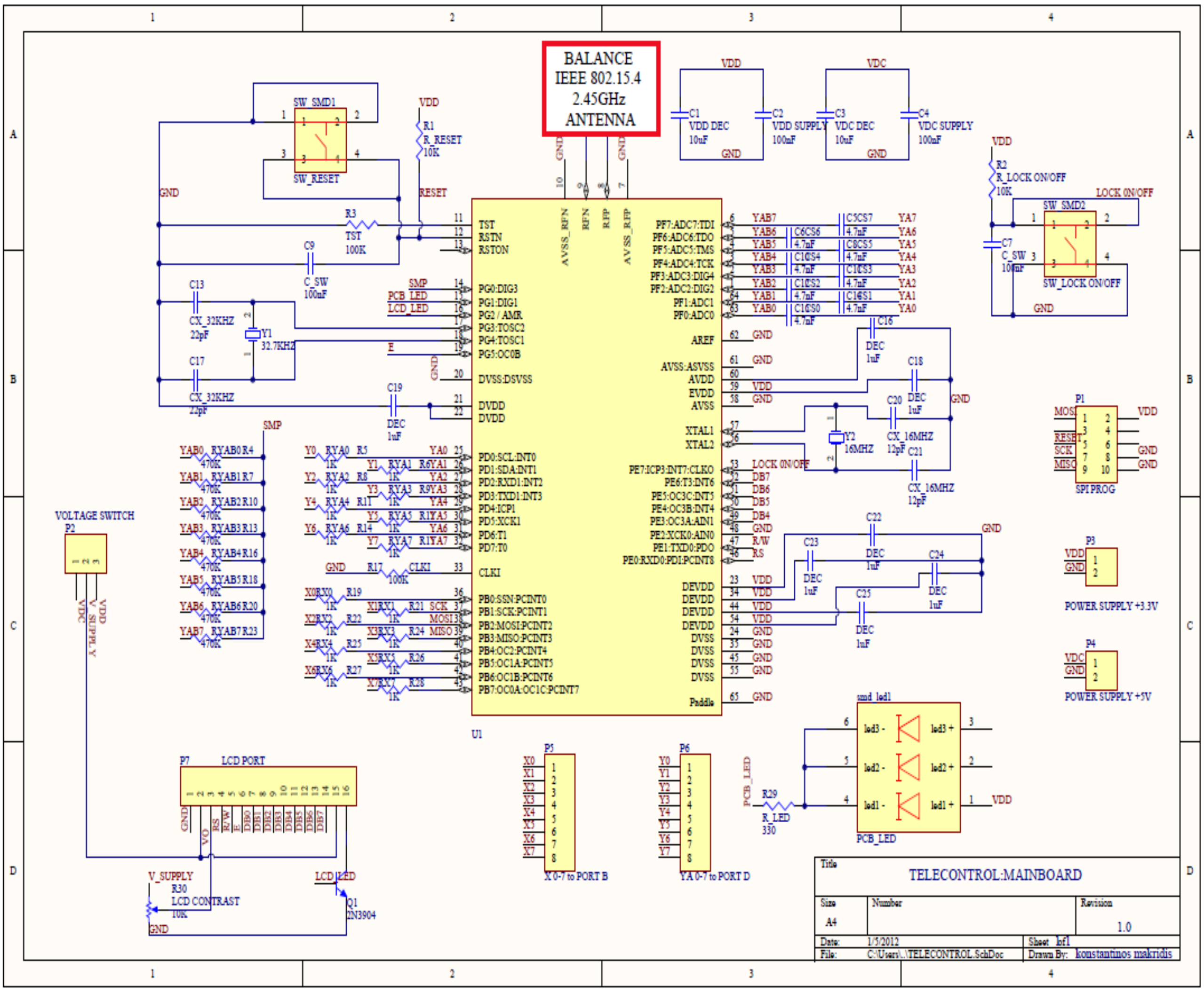


Figure 39 Telecontrol-Mainboard schematic

QTouch Library use Rx0-Rx7, Cs0-Cs7, RYAB0-RYAB7 and RYA0-RY7 to create 64 touch buttons of 8X8 matrix. It uses 3 ports from ATmega128RFA1 MCU and 2 pins from some ports that left free. Default value for Rx0-7 and RYA0-7 is 1KΩ, for RYAB0-7 is 470 KΩ and for Cs0-7 is 4,7 nF. Trimmer value for LCD contrast must be around 10 KΩ to 20 KΩ. For real time clock MCU it use 32,7KHz and for MUC function it use 16MHz crystal.

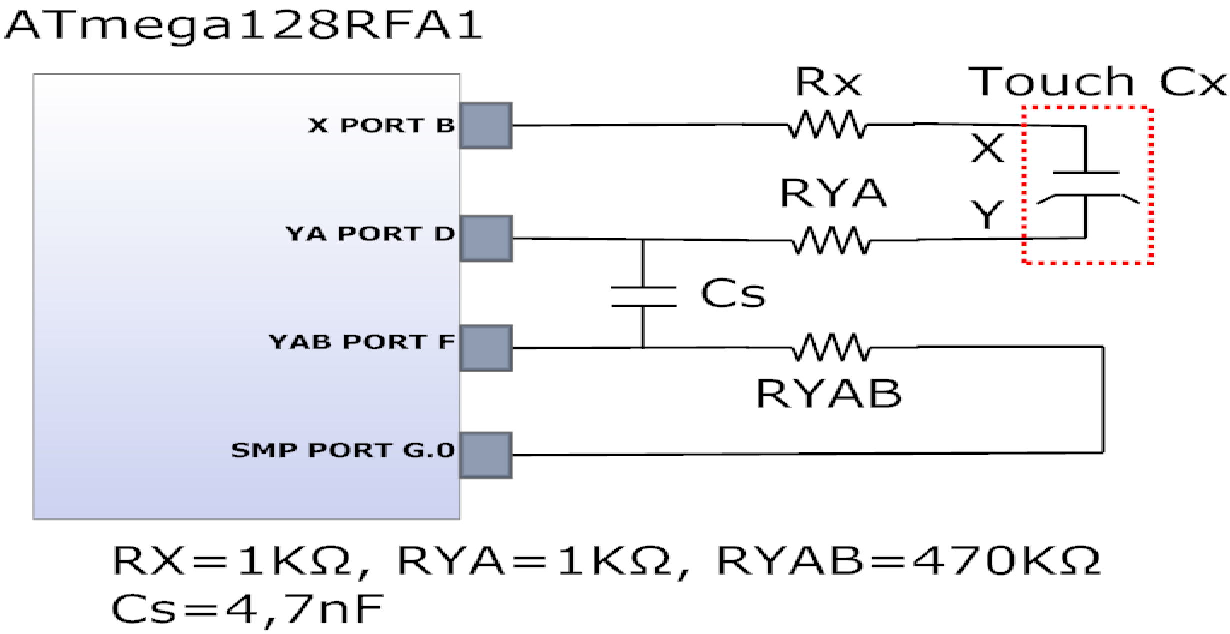


Figure 40 General QTouch Matrix Library schematic (Atmel QTouch Library, 16.)

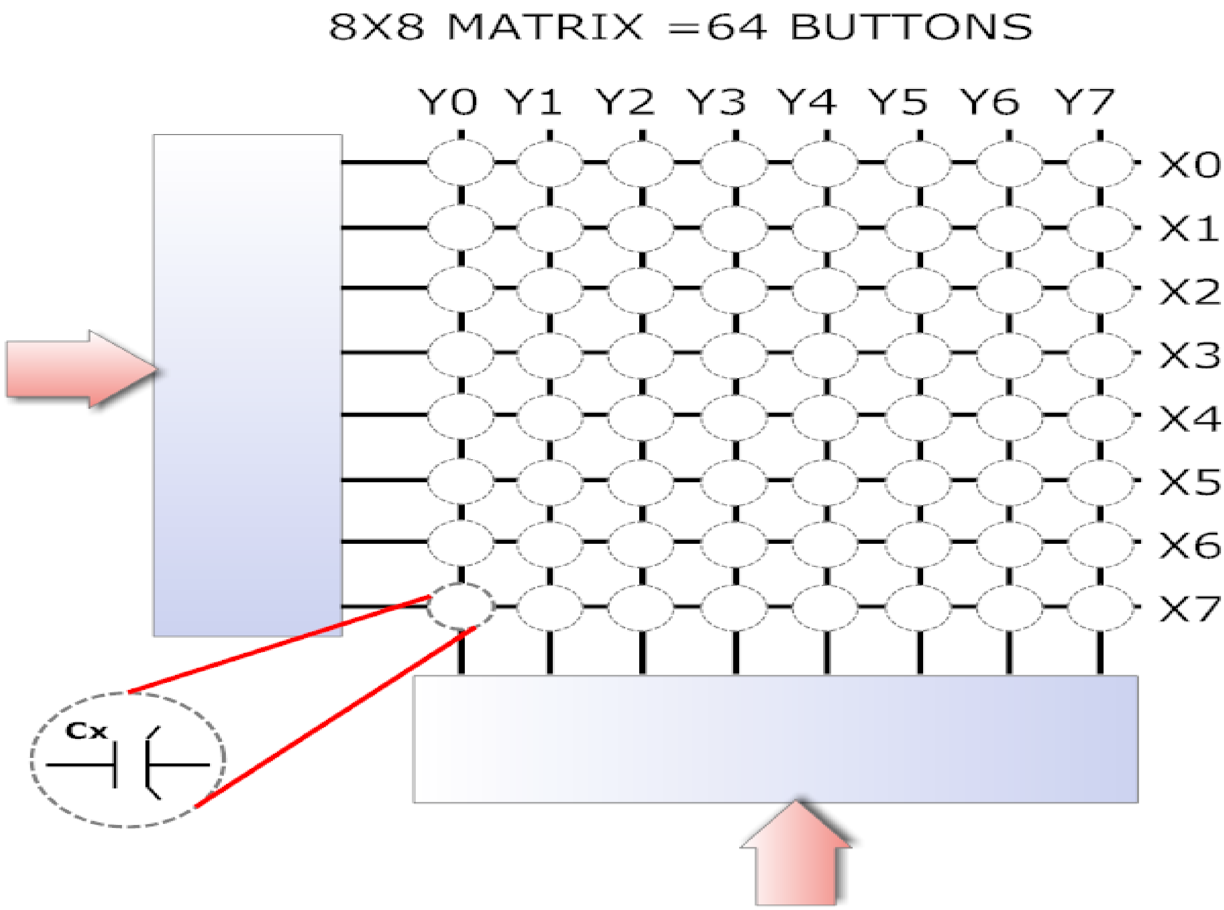


Figure 41 QTouch Matrix buttons schematic

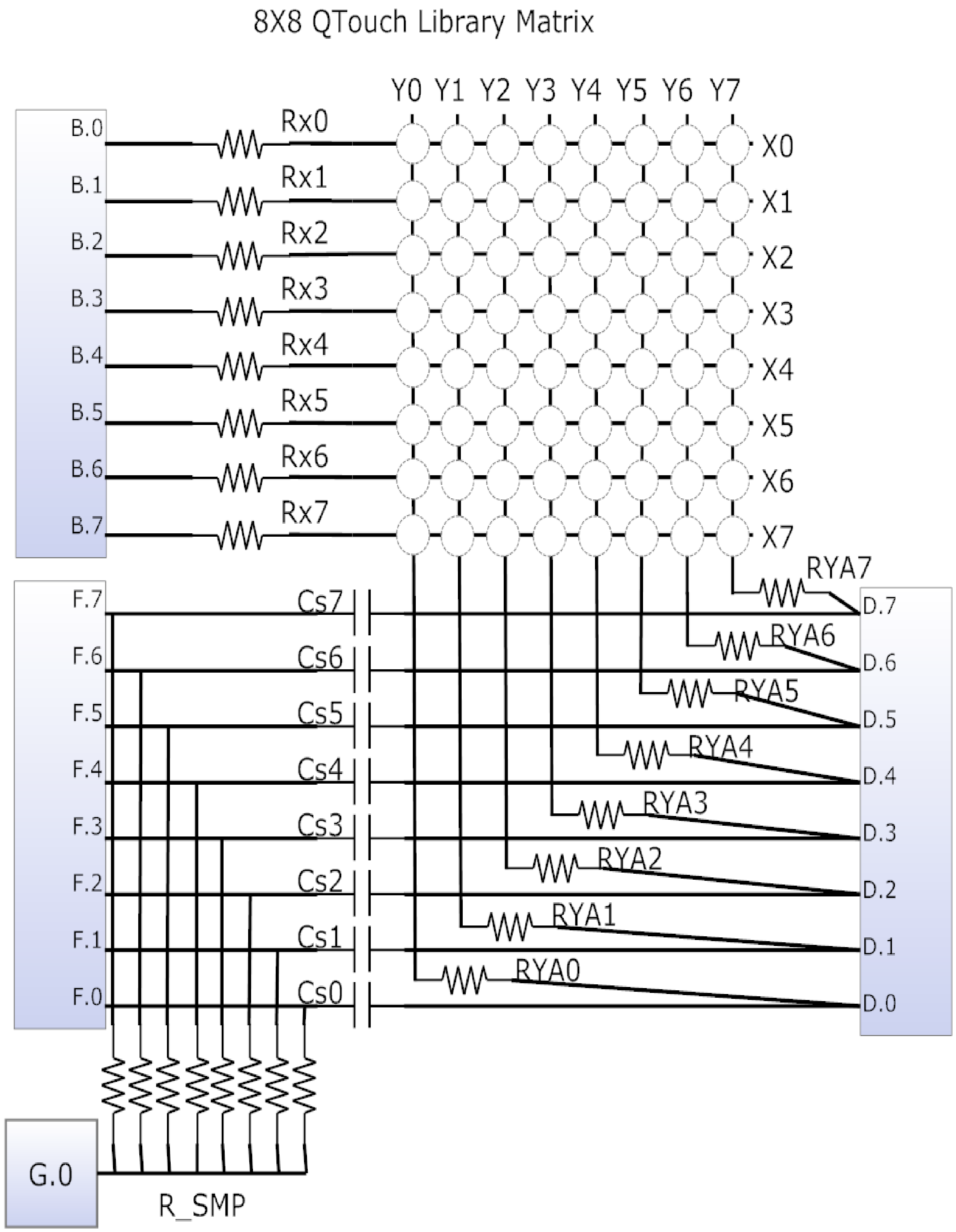


Figure 42 QTouch Matrix Mainboard schematic (Atmel QTouch Library, 16.)

4.2 On feature

We study telecontrol device one from remote control category, but is not the only one. In feature we can see **computer** and special how with software can control whole house/factory and also for **Smartphone's** a software for controlling whole house/factory.

O.P.R.A software will be design for Smartphone's and computers, it will be two different package one for houses and the other for small/big corporations/factories. For houses the GUI (window software) can connect with improvement MDC via USB cable and Ethernet cable. For small/big corporations/factories the GUI can connect with improvement MDC via USB cable, Ethernet cable and with USB to Wireless transmitter (this device is connected to USB port and transform the electrical signal from computer to wireless RF waves). For houses the GUI from Smartphone's device can connect with improvement MDC via Bluetooth and via cell phone network by SMS message.

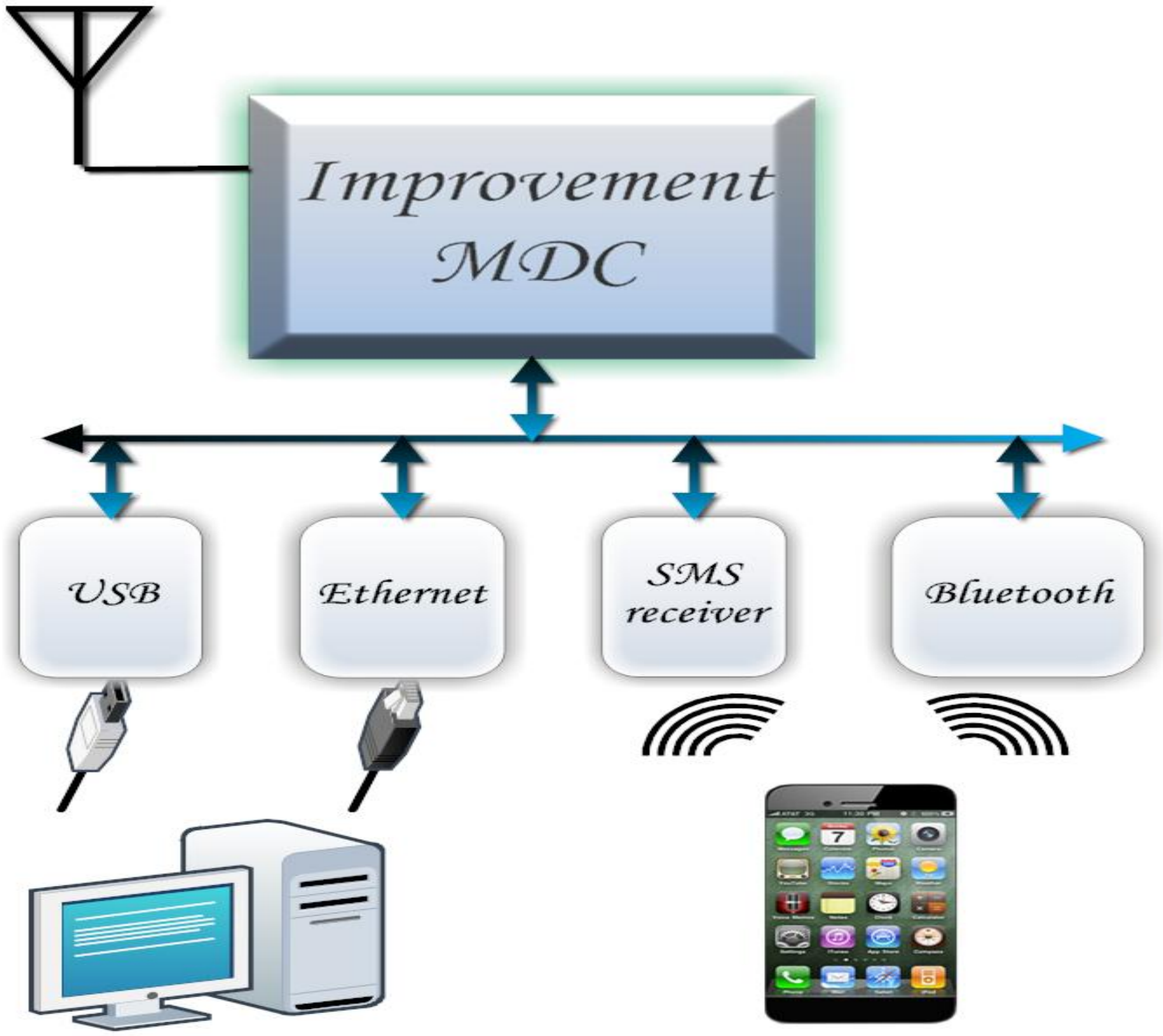


Figure 43 Feature remote controls and connectivity

5 MASTER DEVICES CONTROLLER 'MDC'

5.1 Standard MDC

MDC board is the leader in home automation system, is the coordinator of the communication. MDC receive commands from remotes controls let's say from telecontrol and after some processing communicate wireless or wired with devices to transfer commands for execution. When he sent message via wired it use power line modem that is ST7540 and when he sent message via wireless it use the same protocol like he receive message from telecontrol, the protocol it use is IEEE 802.15.4 - 2,45GHz. There are two different packages for MDC, standard version is simple without any add-on, only MCU, ST7540 and power supply. The improvement version have all the add-ons that standard version don't have, add-on is USB port, Ethernet port, SMS receiver and Bluetooth.

5.1.1 ATmega128RFA1

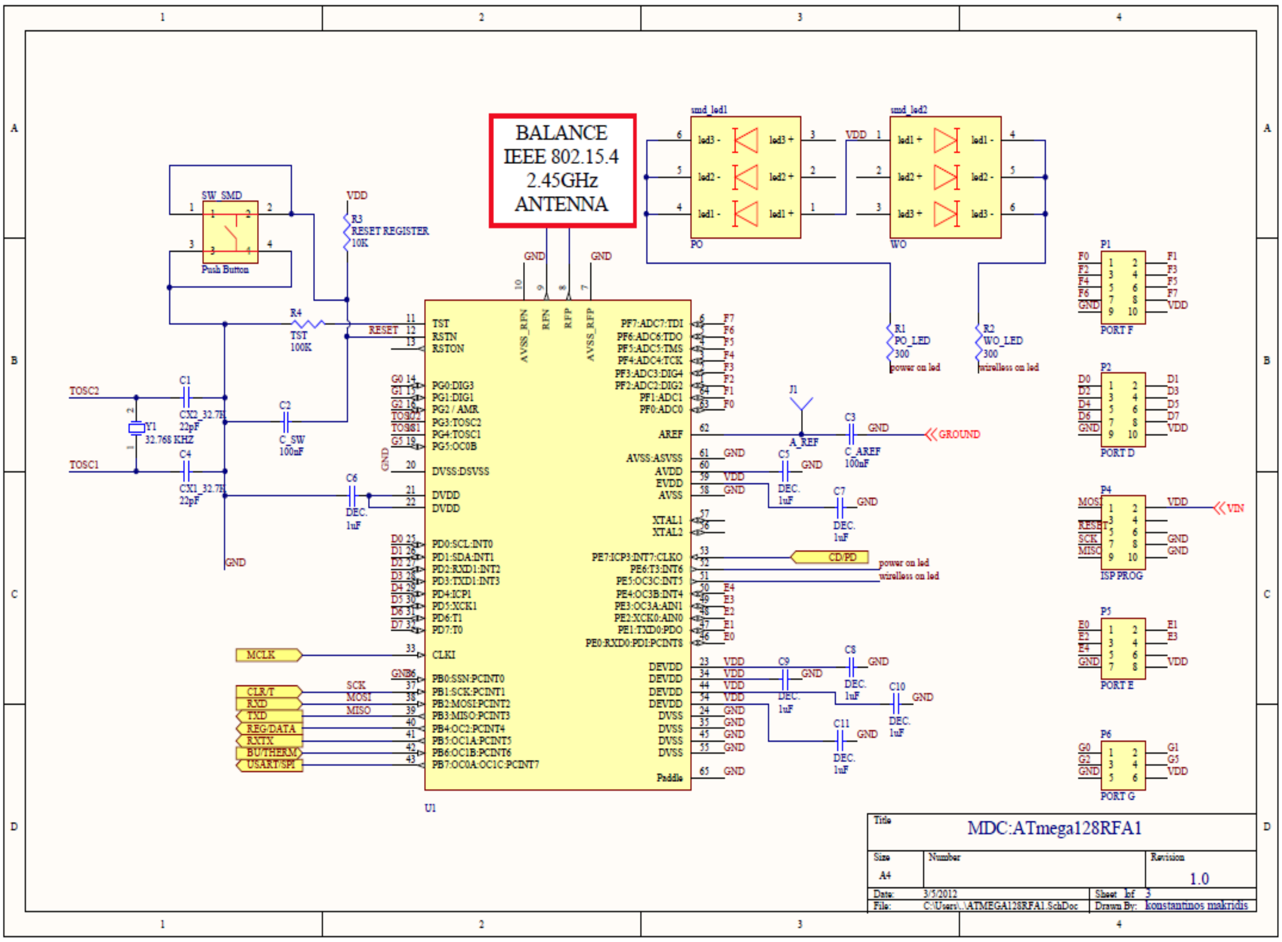


Figure 44 MDC-ATmega128RFA1 schematic

For standard MDC there is not much thing to say only that have two led for knowing the status of wireless connection and power supply. There is some receptacle for feature upgrade and final connectivity with ST7540 chip using some pins for receiving and transmitting data.

In this point we must know few things about Atmel MCU ATmega128RFA1 that is been used to telecontrol and MDC.

ATmega128RFA1 features (ATmega128RFA1 Datasheet, 1.) :

- RISC Architecture
- Most instructions is Single Clock Cycle Execution
- Up to 16 MIPS Throughput at 16 MHz and 1.8V
- 128K Bytes of In-System Self-Programmable Flash
- 4K Bytes EEPROM
- 16K Bytes Internal SRAM
- JTAG (IEEE std. 1149.1 compliant) Interface
- Multiple Timer/Counter & PWM channels
- Real Time Counter
- 10-bit, 330 ks/s A/D Converter
- Master/Slave SPI Serial Interface
- Two Programmable Serial USART
- Byte Oriented 2-wire Serial Interface
- Fully integrated Low Power Transceiver for 2.4 GHz ISM Band
    - Supported Data Rates: 250 kb/s and 500 kb/s, 1 Mb/s, 2 Mb/s
    - -100 dBm RX Sensitivity; TX Output Power up to 3.5 dBm
- Hardware Security (AES, True Random Generator)
- 38 Programmable I/O Lines
- 64-pad QFN (RoHS/Fully Green)
- Supply voltage range 1.8V to 3.6V

For more details about ATmega128RFA1 chip features check ATmega128RFA1 Datasheet, page 1.

For more details about ATmega128RFA1 radio transceiver that is a fully integrated ZigBee check ATmega128RFA1 Datasheet, chapter 9 pages 30 and on.

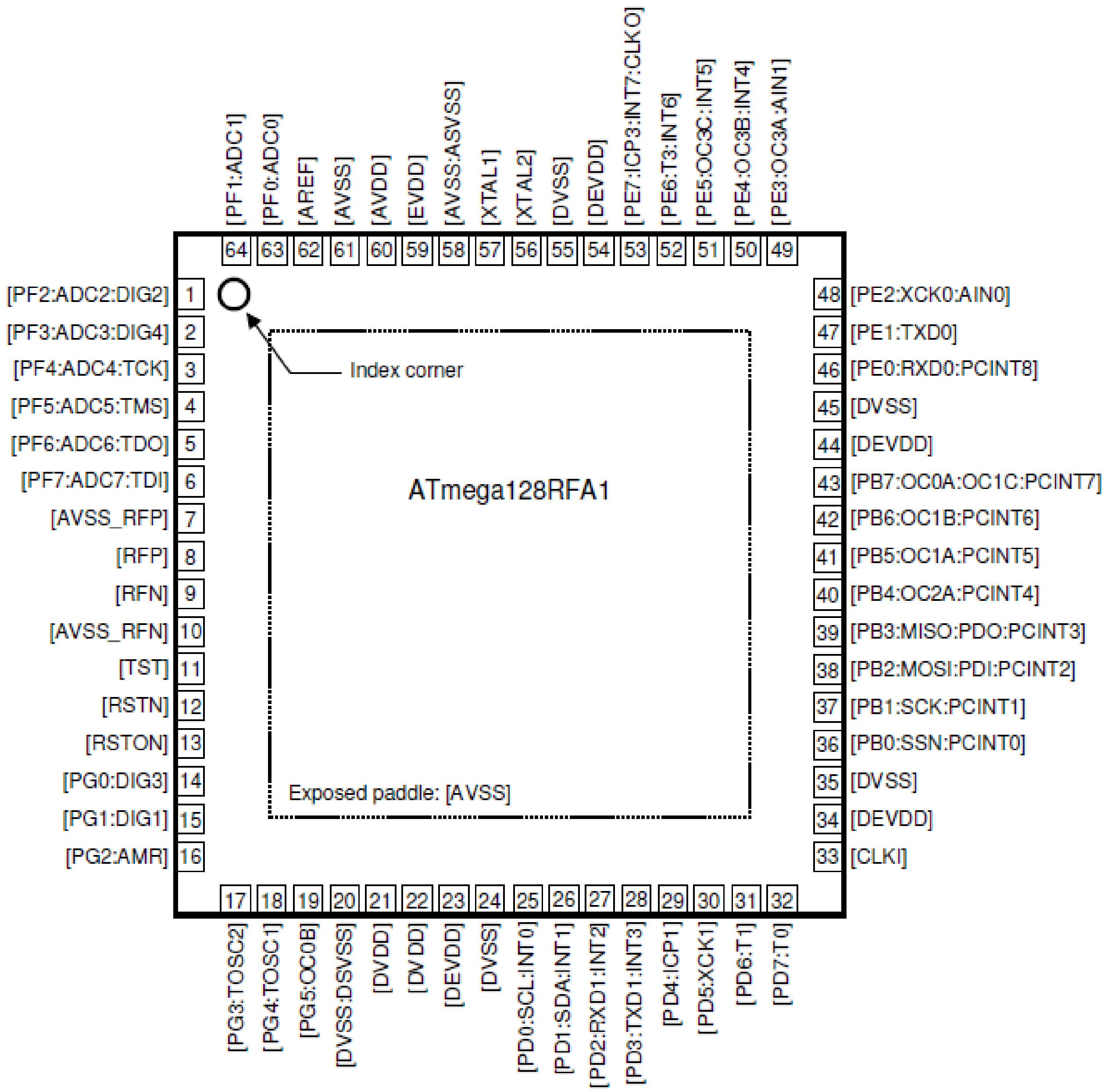


Figure 45 ATmega128RFA1 pinout (ATmega128RFA1 Datasheet, 2.)

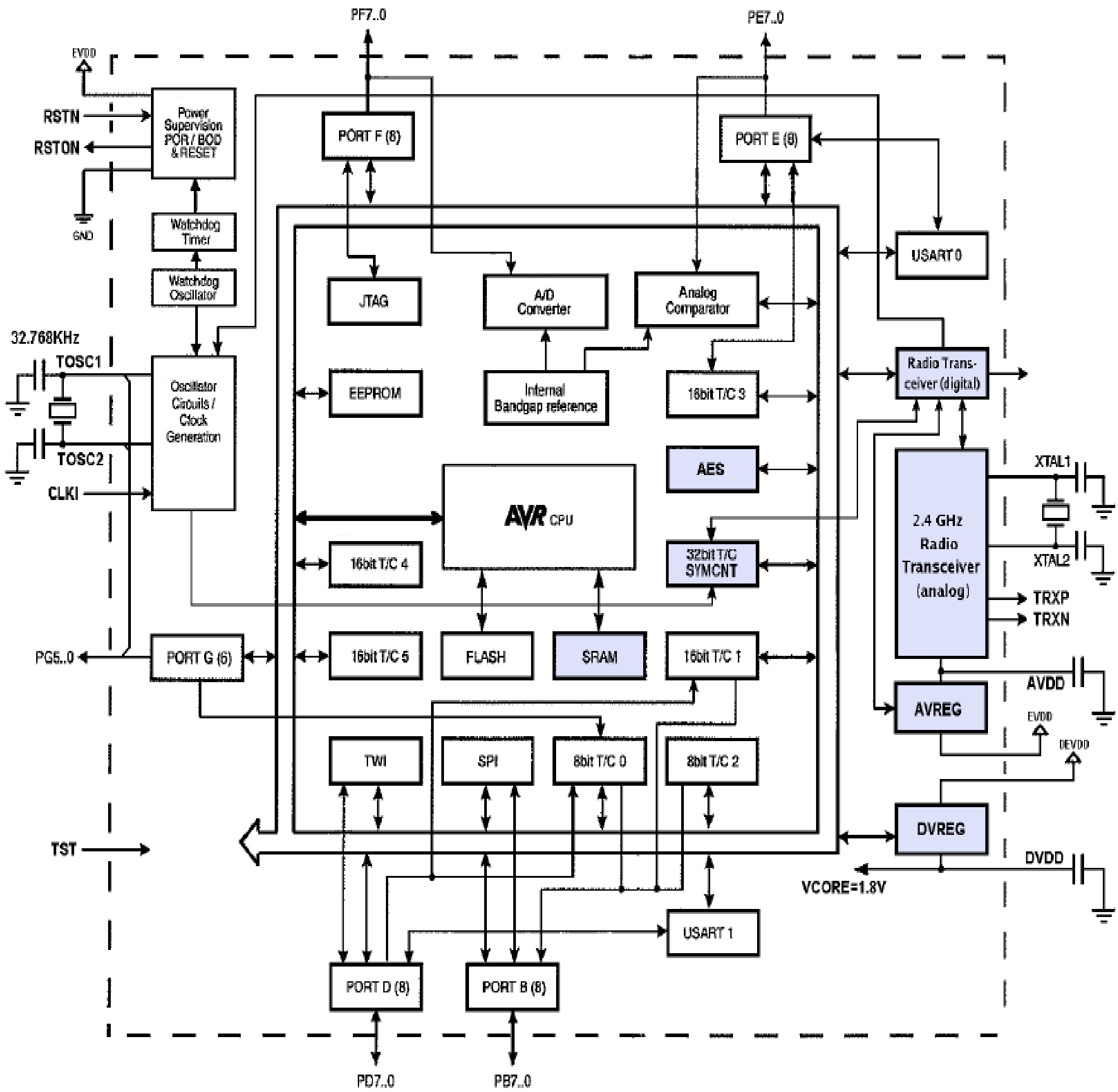


Figure 46 ATmega128RFA1 block diagram (ATmega128RFA1 Datasheet, 3.)

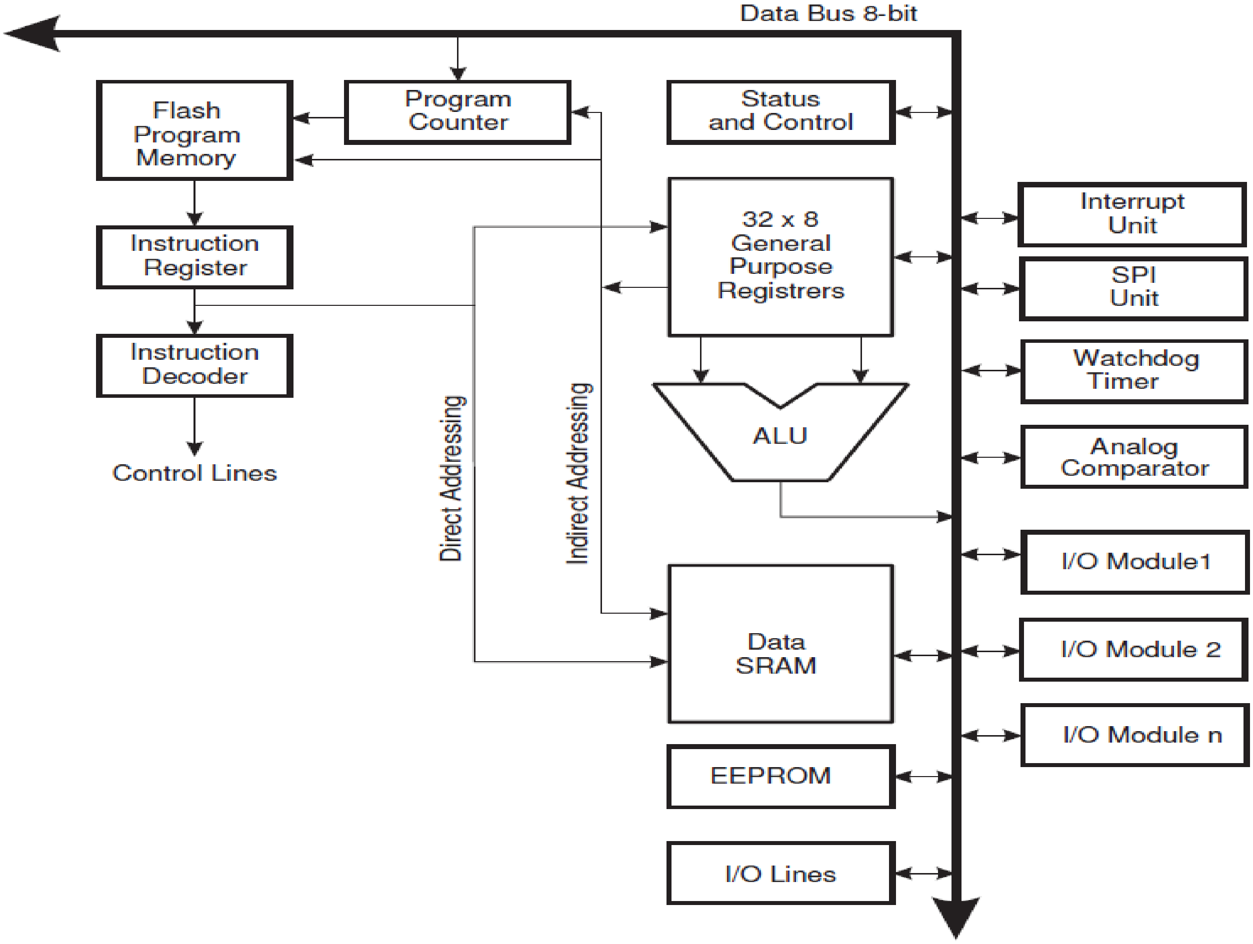


Figure 47 Atmel MCUs Harvard architecture (ATmega128RFA1 Datasheet, 9.)

5.1.2 ST7540 (PLM)

ST7540 chip uses frequency shift keying (FSK) modulation over powerline network (110/220V), this transceiver can perform a half duplex communication. ST7540 operates from a 7.5 to 13.5 Volt with single supply voltage (Vcc) and inside chip there is a power amplifier (PA), which is able to drive low line impedance with user can define the current and voltage of that signal from PA. Also the chip has inside two linear regulators providing 5 V (Vdc) and 3.3 V (Vdd). The ST7540 can communicate using eight different communication channels (60, 66, 72, 76, 82.05, 86, 110, 132.5 kHz), four baud rates (600, 1200, 2400, 4800) and two deviations (1, 0.5). Transceiver main operation is to support physical communication, that mean is needed the use of a microcontroller so he can manage the communication using O.P.R.A protocol. The host controller can transmit/receive data with the ST7540 transceiver through a serial interface, there is two operations of a serial interface first UART (CLR/T data clock not supported) and second SPI. Preferable communication over serial interface is SPI because message can be synchronize by the clock of ST7540. (AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 10.)

For more information about **transmission mode** in ST7540 chip check AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 10 from sources.

For more information about **receiving mode** in ST7540 chip check AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 10-11 from sources.

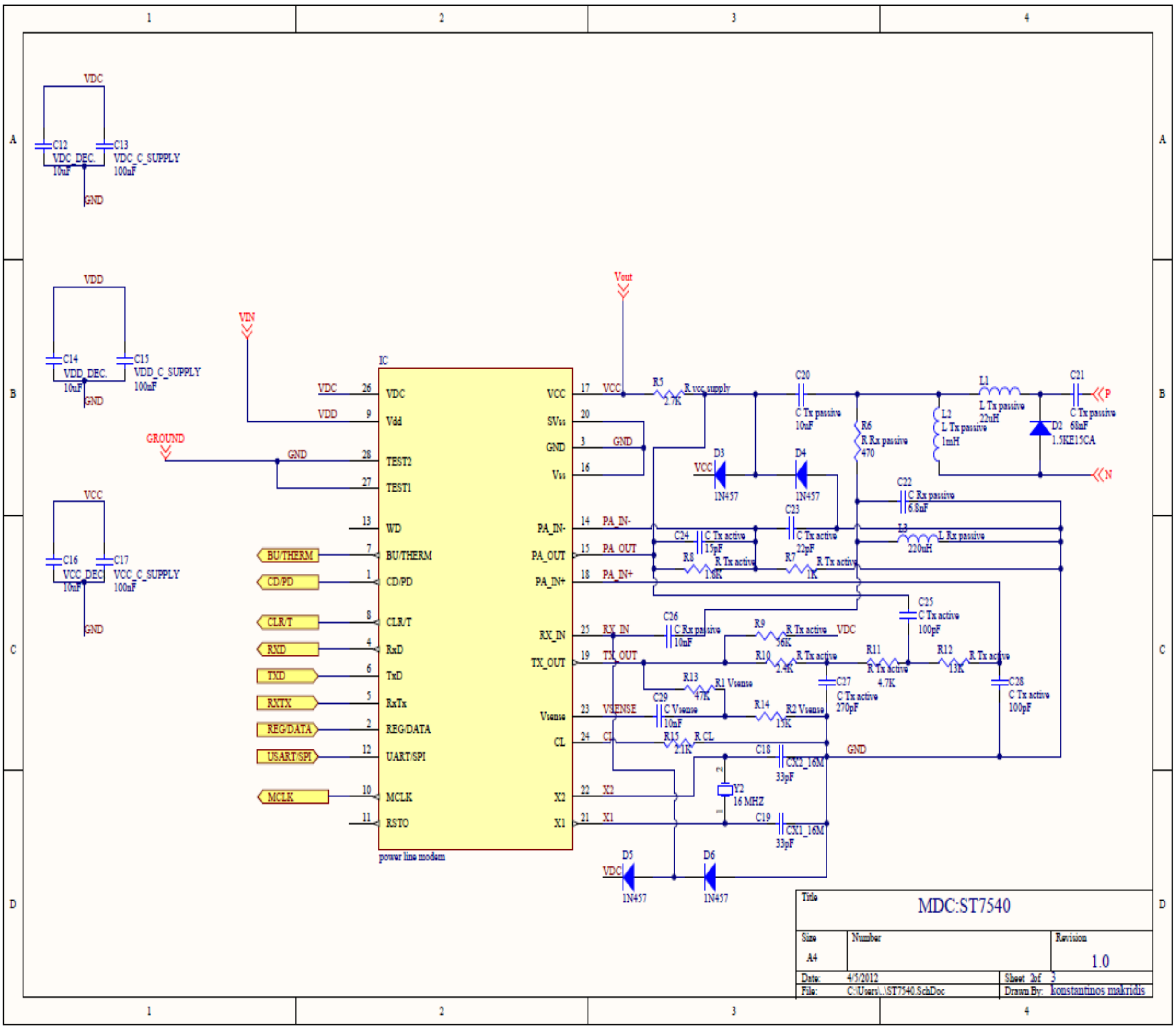


Figure 48 MDC-ST7540 schematic

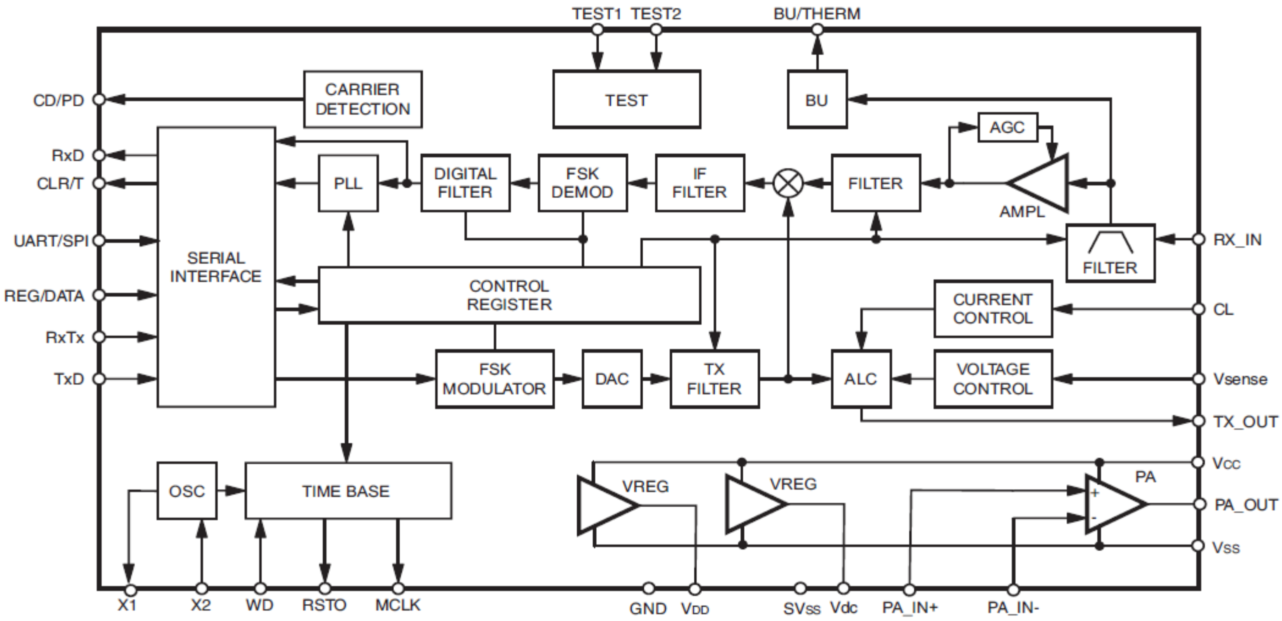


Figure 49 ST7540 block diagram (ST7540 Power Line Modem Datasheet, 4.)

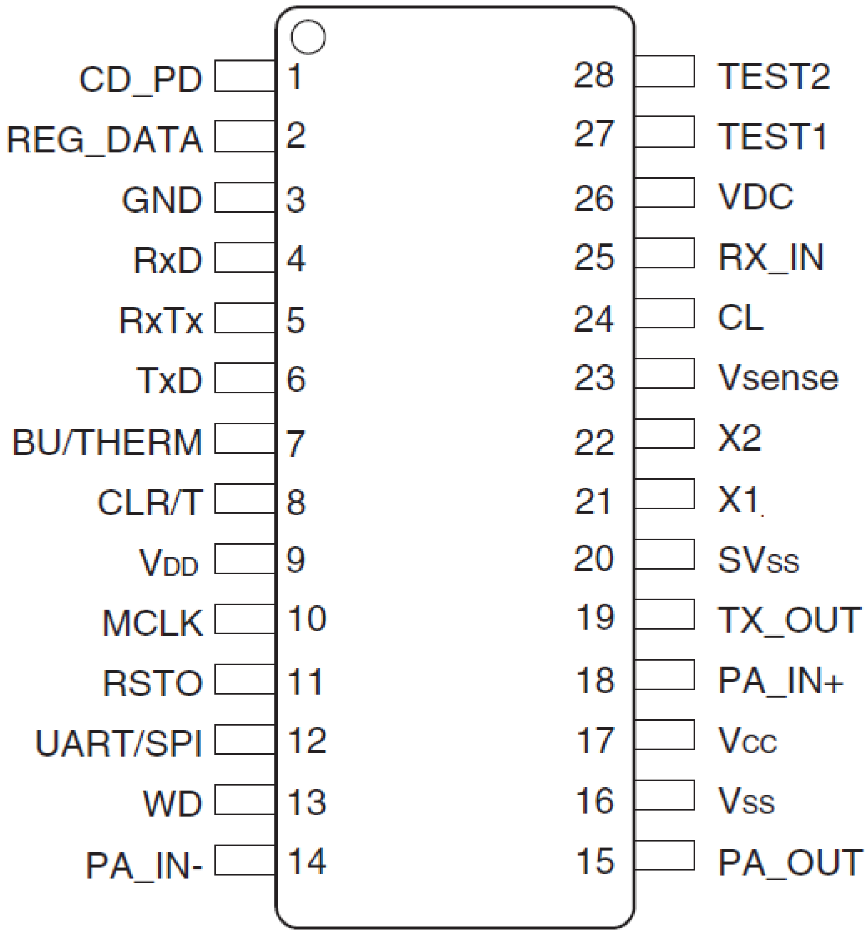


Figure 50 ST7540 pin connection (ST7540 Power Line Modem Datasheet, 5.)

Table 66 Pin description (ST7540 Power Line Modem Datasheet, 5-6.)

PIN N.	NAME	TYPE I/O	DESCRIPTION
1	CD_PD	Digital / Output	Carrier, preamble or frame header detects output. "1" No carrier, preamble or frame header detected "0" Carrier, preamble or frame header detected
2	REG_DATA	Digital / Input	Mains or control register access selector "1" - Control register access "0" - Mains access
3	GND	Supply	Digital ground
4	RxD	Digital / Output	RX data output.
5	RxTx	Digital / Input	Rx or Tx mode selection input. "1" - RX Session "0" - TX Session
6	TxD	Digital / Input	TX data input.
7	BU/THERM	Digital / Output	Band in use/Thermal Shutdown event detection output. In Rx mode: "1" Signal within the programmed band "0" No signal within the programmed band In Tx mode: "1" - Thermal Shutdown event occurred "0" - No Thermal Shutdown event occurred
8	CLR/T	Digital / Output	Synchronous mains access clock or control register access clock
9	Vdd	Power	+3.3V voltage regulator output
10	MCLK	Digital / Output	Master clock output
11	RSTO	Digital / Output	Power ON or watchdog reset output
12	UART/SPI	Digital / Input	Interface type: “0” - Serial peripheral interface “1” - UART interface
13	WD	Digital / Input	Watchdog input.
14	PA_IN-	Analog / Input	Power line amplifier inverting input
15	PA_OUT	Power / Output	Power line amplifier output
16	Vss	Supply	Power analog ground
17	Vcc	Supply	Power supply voltage 7.5 – 13.5V
18	PA_IN+	Analog / Input	Power line amplifier not inverting input
19	TX_OUT	Analog / Output	Small signal analog transmit output
20	SVss	Supply	Analog signal ground
21	X1	Analog / Output	Crystal oscillator output
22	X2	Analog / Input	Crystal oscillator input - or external clock input
23	Vsense	Analog / Input	Sensing input for the voltage control loop
24	CL	Analog / Input	Current limiting feedback.
25	RX_IN	Analog / Input	Receiving analog input
26	VDC	Power	+5V voltage regulator output
27	TEST1	Digital / Input	Test input. Must be connected to GND.
28	TEST2	Analog / Input	Test input. Must be connected SVSS

Control Register (ST7540 Power Line Modem Datasheet, 33-35.)

Table 67 Control Register parameter summary

BITS	PARAMETER	BITS	PARAMETER
0-2	Carrier Frequency	**17**	Output voltage freeze
3-4	Baud Rate	**18**	Header recognition
5	Deviation	**19**	Frame length count
6	Watchdog	**20**	Header length
7-8	Transmission timeout	**21**	Extended control register
9-10	Frequency detection time	**22**	Sensitive mode
11	Not used	**23**	Pre- filter
12-13	Detection method	**24-39**	Frame header
14	Mains interface mode	**40-47**	Frame length
15-16	Output clock		

Table 68 Carrier Frequency

BITS 0-2 (MSB to LSB)	VALUE (KHz)	BITS 0-2 (MSB to LSB)	VALUE (KHz)
000	60	**100**	82.05
001	66	**101**	86
010	72	**110**	110
011	76	**111**	**132.5 (default)**

Table 69 Baud Rate

BITS 3-4 (MSB to LSB)	VALUE	BITS 3-4 (MSB to LSB)	VALUE
00	600	**10**	**2400 (default)**
01	1200	**11**	4800

Table 70 Deviation

BIT 5	VALUE	BIT 5	VALUE
0	**0.5 (default)**	**1**	1

Table 71 Watchdog

BIT 6	VALUE	BIT 6	VALUE
0	Disabled	**1**	**Enabled 1.5sec (default)**

Table 72 Transmission timeout

BITS 7-8 (MSB to LSB)	VALUE (sec)	BITS 7-8 (MSB to LSB)	VALUE (sec)
00	Disabled	**10**	3 sec
01	**1 sec (default)**	**11**	Not used

Table 73 Frequency detection time

BITS 9-10 (MSB to LSB)	VALUE	BITS 9-10 (MSB to LSB)	VALUE
00	500 us	**10**	3 msec
01	**1 msec (default)**	**11**	5 msec

Table 74 Detection method

BITS 12-13 (MSB to LSB)	VALUE	BITS 12-13 (MSB to LSB)	VALUE
00	Preamble detection without conditioning	**10**	**Carrier detection without conditioning (default)**
01	Preamble detection with conditioning	**11**	Carrier detection with conditioning

Table 75 Mains interface mode

BIT 14	VALUE	BIT 14	VALUE
0	Synchronous	**1**	**Asynchronous (default)**

Table 76 Output clock

BITS 15-16 (MSB to LSB)	VALUE	BITS 15-16 (MSB to LSB)	VALUE
00	16 MHz	**10**	**4 MHz (default)**
01	8 MHz	**11**	Clock off

Table 77 Output voltage freeze

BIT 17	VALUE	BIT 17	VALUE
0	Enabled	**1**	**Disabled (default)**

Table 78 Header recognition

BIT 18	VALUE	BIT 18	VALUE
0	**Disabled (default)**	**1**	Enabled

Table 79 Frame length count

BIT 19	VALUE	BIT 19	VALUE
0	**Disabled (default)**	**1**	Enabled

Table 80 Header length

BIT 20	VALUE	BIT 20	VALUE
0	8 bit	**1**	**16 bit (default)**

Table 81 Extended control register

BIT 21	VALUE	BIT 21	VALUE
0	**Disabled 24bits (default)**	**1**	Enabled 48bits

Table 82 Sensitive mode

BIT 22	VALUE	BIT 22	VALUE
0	**Normal (default)**	**1**	High

Table 83 Pre- filter

BIT 23	VALUE	BIT 23	VALUE
0	**Disabled (default)**	**1**	Enabled

Table 84 Frame header

BIT 24-39	VALUE
0000h-FFFFh	**9B58h (default)**

Table 85 Frame length

BIT 40-47	VALUE
01h-FFh	**08h (default)**

Mark and Space Communication Frequencies:

- F ("0") = Fcarrier + [ΔF]/2
- F ("1") = Fcarrier - [ΔF]/2

ΔF = Frequency Deviation.

- "0.5" deviation ΔF=0.5*Baudrate
- "1" deviation ΔF= Baudrate

Table 86 Mark and Space for 132.5 KHz Fcarrier

Carrier Frequency (KHz)	Baudrate	Deviation	Exact frequency in Hz (clock = 16MHz)	
			"1"	"0"
132.5KHz	600	-		
		1	132161	132813
	1200	0.5	132161	132813
		1	131836	133138
	2400	0.5	131836	133138
		1	131348	133626
	4800	0.5	131348	133626
		1	130046	134928

(ST7540 Power Line Modem Datasheet, 17-18.)

Host MCU interface with ST7540

ST7540 receiving/transfer data with the AT128RFA1 through a serial interface and special with the SPI mode. (ST7540 Power Line Modem Datasheet, 19.)

Table 87 Data and Control register access bits configuration (ST7540 Power Line Modem Datasheet, 19.)

ST7540 WORKING MODE	REG_DATA	RxTx
Data Transmission	0	0
Data Reception	0	1
Control Register Read	1	1
Control Register Write	1	0

ST7540 use two types of Host Communication Interfaces:

- SPI
- UART

Table 88 Host MCU / ST7540 interface (ST7540 Power Line Modem Datasheet, 19.)

Host device interface type	**UART/SPI pin**	**Communication mode**	**Main access**	
			Asynchronous	**Synchronous**
USART	“1”	Transmission	X	
USART	“1”	Reception	X	
SPI	“0”	Transmission		X
SPI	“0”	Reception		X

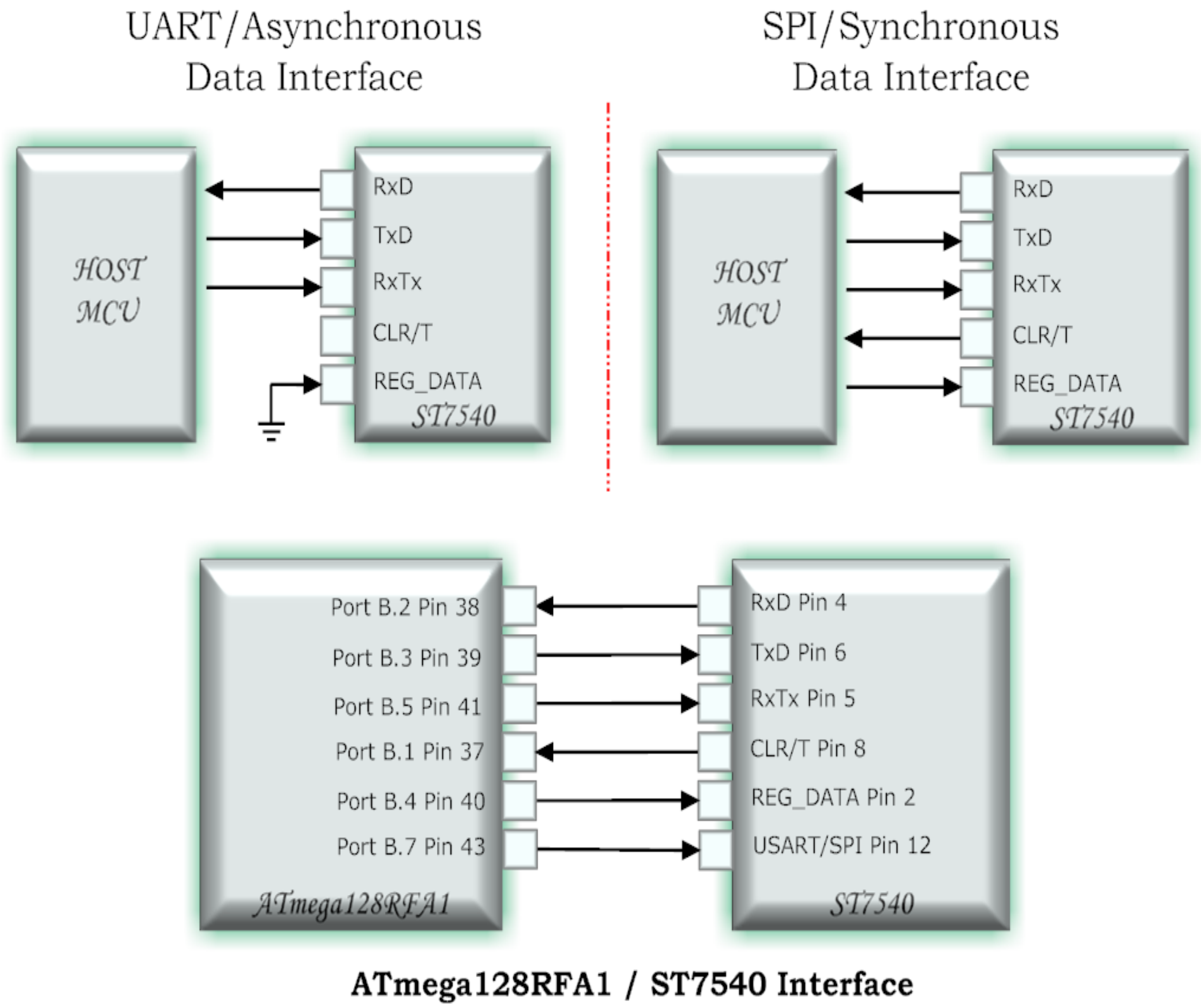


Figure 51 Synchronous and Asynchronous ST7540/Host MCU interfaces

ST7540 FILTERS

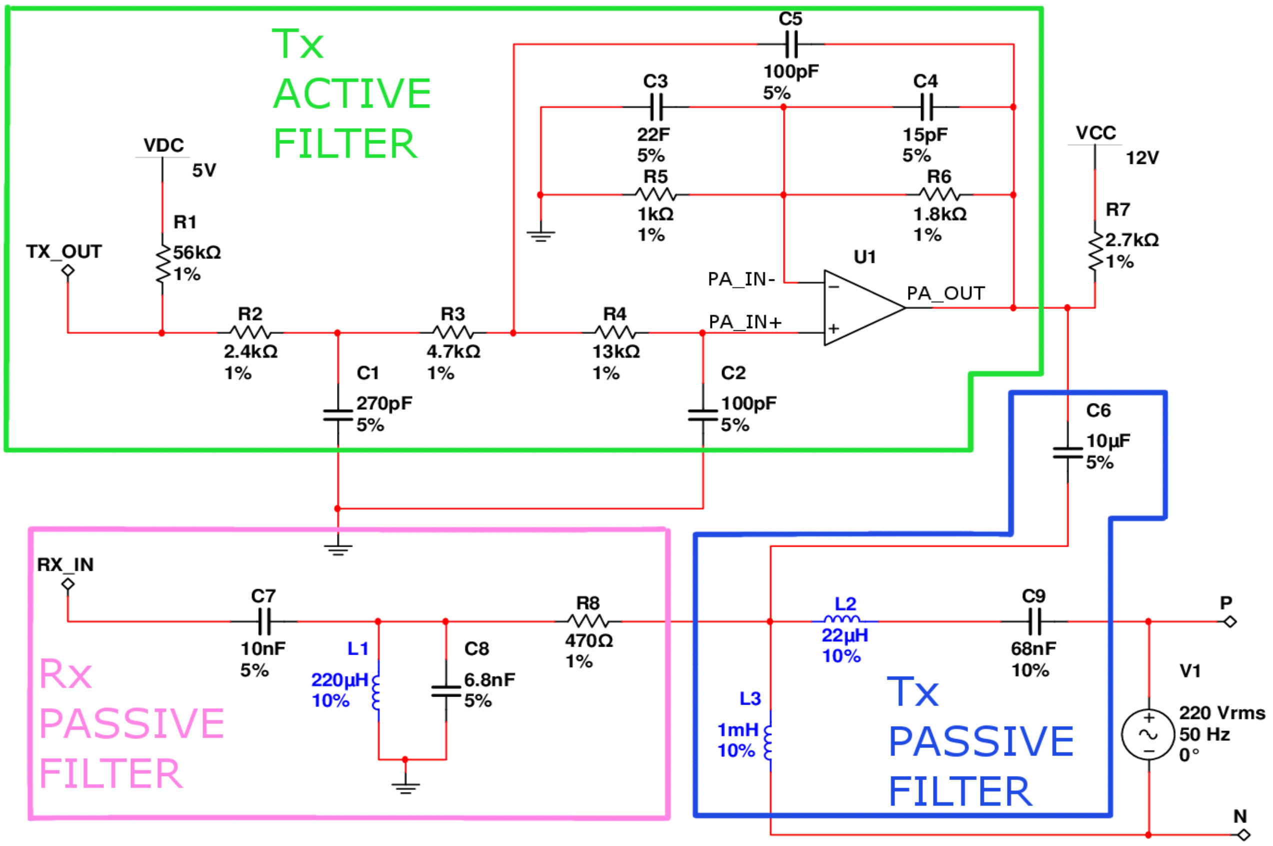


Figure 52 Rx and Tx filter of ST7540

From figure 52 Tx active filter is based on the ST7540 internal power amplifier (PA_in-, PA_IN+, PA_OUT) using external R-C components; Tx active filter is a 3-pole low-pass filter by cascading a simple R-C low-pass with a Sallen-Key 2-pole. In figure 52 at Tx active filter R2, C1 is first order low-pass filter and R3, R4, R5, R6, C2, C3, C4, C5 make a second order Sallen-key low-pass filter. (AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 19.) **Notice: the number over the components is different in figure 48; figure 52 is an illustration how can calculate filters.**

The transfer function of the 2nd order Sallen-Key is:

$$A(s) = \frac{A_0}{\frac{s^2}{\omega c^2} + \frac{s}{\omega c * Q} + 1}$$

$$A_0 = \left(1 + \frac{R6}{R5}\right), \qquad \omega c = \frac{1}{\sqrt{R3*R4*C2*C5}}$$

$$Q = \frac{\sqrt{R3*R4*C2*C5}}{R3*C2+R4*C2+R3*C5*(1-A_0)}$$

$$fc = \frac{1}{2 * \pi * \sqrt{R3 * R4 * C2 * C5}}$$

(AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 19.)

From above equations "As" is the transfer function of the filter and represent the output to input (Vout/Vin), "A0" is the gain of the op-amp, in our case is the internal power amplifier of ST7540, "Q" is the quality factor of the active filter and "fc" is the corner frequency of the active filter.

Using real number in equations we take:

$$A_0 = \left(1 + \frac{1.8K}{1K}\right) = \mathbf{2.8}$$

$$Q = \frac{\sqrt{4.7K * 13K * 100p * 100p}}{4.7K * 100p + 13K * 100p + 4.7K * 100p * (1 - 2.8)} = \mathbf{0.84}$$

$$fc = \frac{1}{2 * \pi * \sqrt{4.7K * 13K * 100p * 100p}} = \mathbf{203.6KHz}$$

For the first order R-C low-pass filter the equation is:

$$fc = \frac{1}{2*\pi*R2*C1}, \quad \omega c = \frac{1}{R2*C1}, \quad Q<1/2$$

Using real number in equations we take:

$$fc = \frac{1}{2 * \pi * 2.4K * 270p} = \mathbf{245.6KHz}$$

The Tx passive filter is made of the decoupling capacitor C6, inductor L2 and X2 safety capacitor C9. The equation and result for center frequency is:

$$fc = \frac{1}{2*\pi*\sqrt{L2*C9}} \Rightarrow fc = \frac{1}{2*\pi*\sqrt{22u*68n}} = \mathbf{130.1\ KHz}$$

The Rx filter is made up of a resistor in series R8 with a parallel L1-C8 resonant. (AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 20, 22.)

The transfer function of the filter is:

$$R(s) = \frac{\frac{s*L1+RL}{R8*L1*C8}}{S^2 + \frac{R8*RL*C8+L1}{R8*L1*C8} * S + \frac{R8+RL}{R8*L1*C8}}$$

$$fc = \frac{1}{2*\pi*\sqrt{L1*C8}}, \quad Q = \frac{R8*L1*C8}{R8*RL*C8+L1} * \omega c$$

“RL” is the DC series resistor of the inductor. Using real number in equations we take:

$$fc = \frac{1}{2 * \pi * \sqrt{220u * 6.8n}} = \mathbf{130.1\ KHz}$$

(AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR, 22-23.)

5.1.3 Power supply

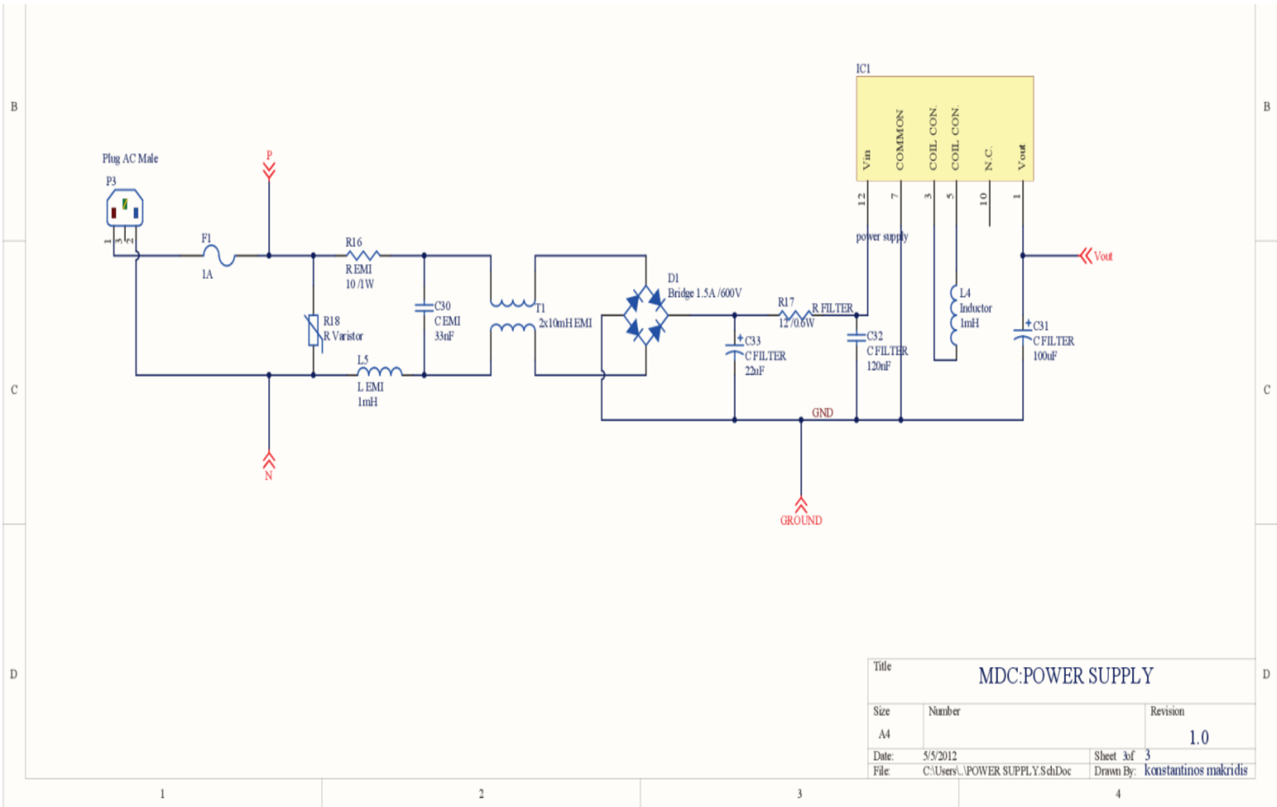


Figure 53 Power supply schematic of MDC

Power supply has been design for fewer components, small space and less complexity. L5, R16, C30 and T1 is the EMI filter and after T1 is the power supply, the main IC for transforming main power (110/220V) to 12 Volt is BP5048 of ROHM semiconductor. ROHM Company has AC/DC modules for 80 to 120V in AC conversion, 127 to 276V in AC conversion, 176 to 253V in AC conversion and 80 to 264V universal in AC conversion. Output voltage of the modules is: -12, -5, +5, +3.3, +12, +15 +24 and for current is: 0.03, 0.05, 0.08, 0.1, 0.12, 0.14, 0.15, 0.17, 0.2, 0.3, 0.35, 0.5, 0.6, 0.8, 1, 3 Ampere. For bigger current it need external transformer, because MDC board no need high current the BP5048 is suitable for the small current but that module work for 220V power network is not universal power supply. Power supply feed with 12 voltages the ST7540 chip and from ST7540 feed MCU with +5 and +3.3 voltage. (Rohm, Non-isolated AC/DC Converter)

Table 89 Electrical characteristically of BP5048 (BP5048 Non-Isolated AC/DC Converter Datasheet, 1.)

BP5048 AC/DC 176 to 253V/+12V 300mA						
Parameter	**Symbol**	**Min**	**Typ.**	**Max**	**Unit**	**Conditions**
Input voltage range	Vi	249	311	358	V	DC
Output voltage	Vo	11	12	13	V	Vi=311 Io=100mA
Output current	Io	0	-	300	mA	Vi=311V
Line regulation	Vr	-0.2	0.05	0.2	V	Vi=249 to 358 Io=100mA
Load regulation	VI	-0.2	0.05	0.2	V	Vi=311 Io=0 to 100mA
Output ripple voltage	Vp	-	0.07	0.15	Vp-p	Vi=311 Io=100mA
Power conversion efficiency	η	65	78	-	%	Vi=311 Io=300mA

5.2 On feature

For common use without extra operations people can use standard MDC, but for people who need extra prospects suitable for that case is Improvement MDC. Improvement MDC has extra modules for extra connectivity and control, via USB user can control whole house from computer using O.P.R.A software, via Ethernet web server module user can control whole house from local network also in that part if is connected to router user can control whole house from internet, just getting in the O.P.R.A web page.

For cell phones/Smartphone's Improvement MDC have SMS controller module, with that way user can send SMS with the right command and control whole house and last there is a Bluetooth module for controlling whole house from cell phones/Smartphone's.

6 DEVICES

6.1 Light switch

Light switch is one from two main categories and device that people use and have inside a house, without light and light switch house can't exist. Light switch comes in 1 to 6 switches, all light switch have both manual and remote control. Manual control comes with a touch button without any mechanical switch, with that way a light switch can have up to 6 buttons-switches that mean a light switch can control up to 6 loads (led's, Incandescent light bulb, Fluorescent lamp). The electrical connection of a single mechanical switch is one wire (phase) as input of the switch and one wire as the output that goes to the lamb. In touch button-switch it needs two wires as input phase and neutral and one wire for output if it is single switch, 6 position switches have 6 output wires. Every light switch have the same power supply circuit as the MDC and the same ST7540 power line modem circuit as the MDC, the difference comes to MCU chip and connectivity with whole components. Light switch use QTouch Library for up to 6 touch sensors-buttons with only two components Cs and Rs. For controlling up to 6 loads the circuit composed by optotriac and triac, both components they make a solid state relay. The role of optotriac is to protect-isolate the rest circuits from the high voltage 110/220 V. Using optotriac can control triac on/off using PWM method; lamb is connected to load 1-6 pin.

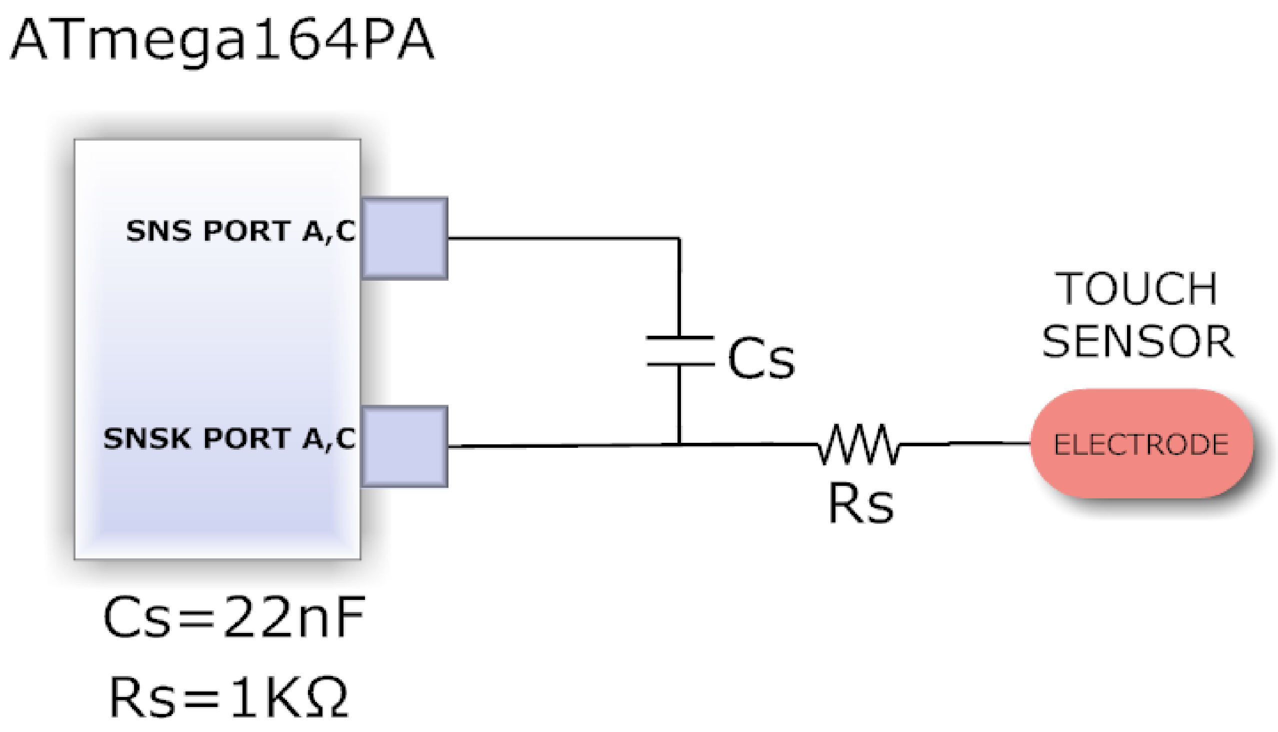


Figure 54 General QTouch sensor Library schematic (Atmel QTouch Library, User Guide, 15.)

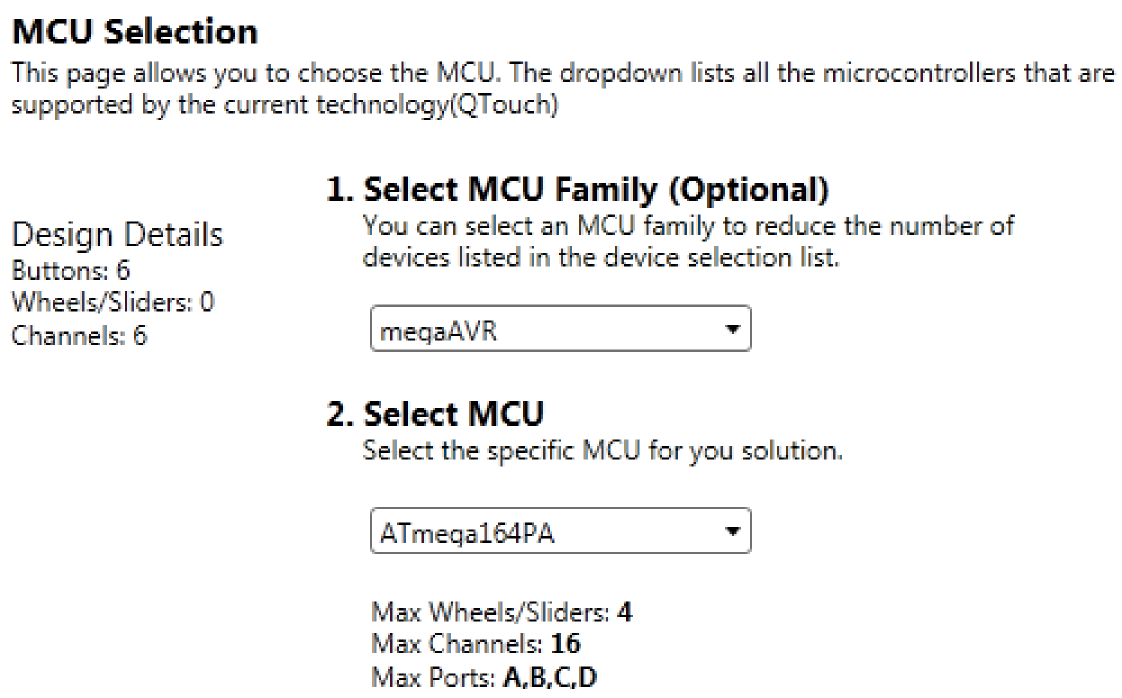


Figure 55 AVR QTouch Studio-MCU family

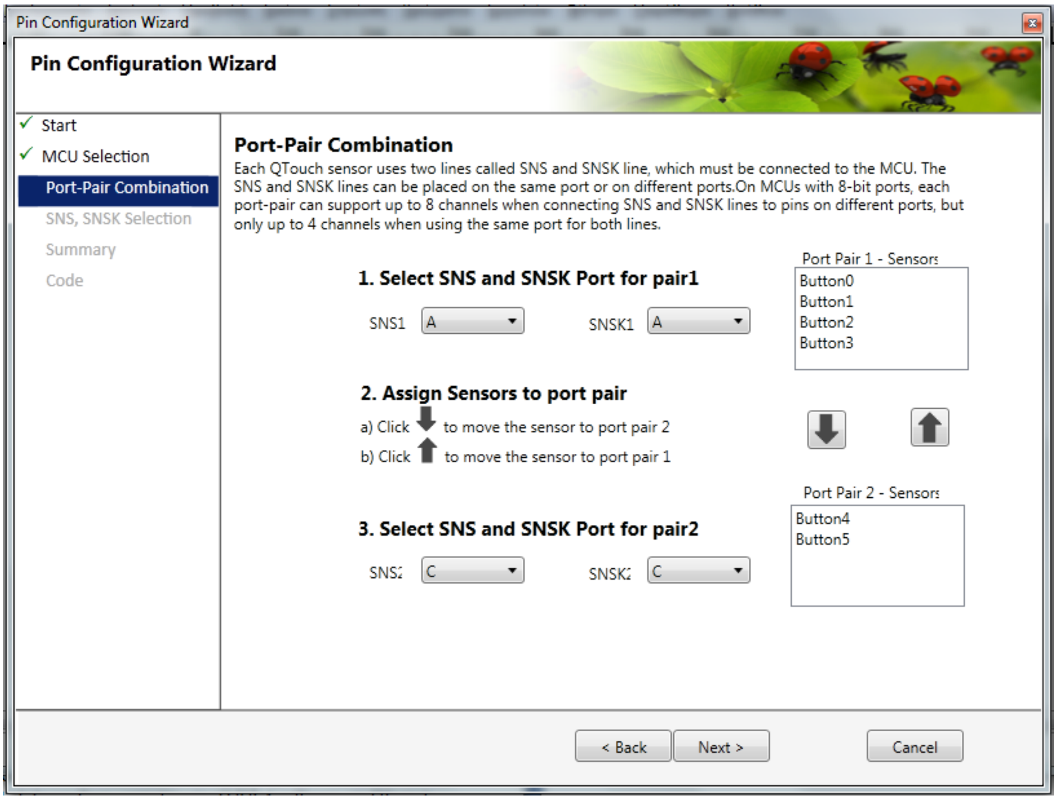


Figure 56 AVR QTouch Studio-MCU SNS, SNSK lines over MCU ports

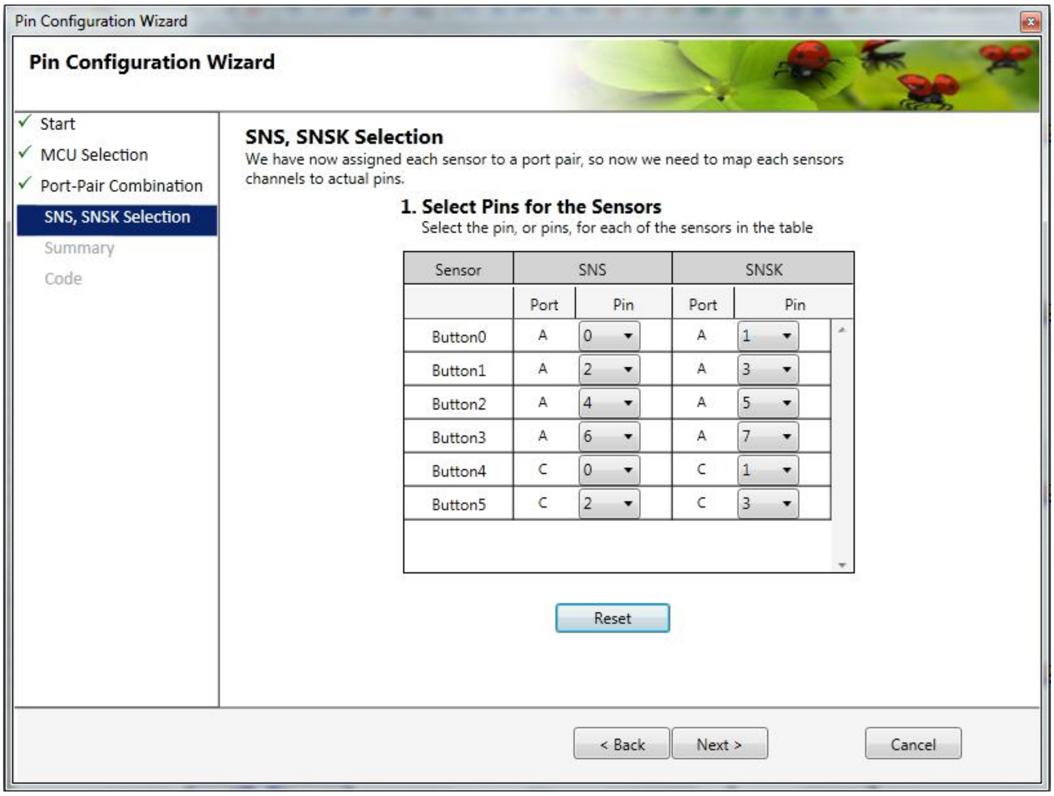


Figure 57 AVR QTouch Studio-Summary

Code of QTouch sensor 6 buttons:

```
#define QTOUCH_STUDIO_MASKS         1
#define NUMBER_OF_PORTS        2
#define SNS1  A
#define SNSK1          A
#define SNS2  C
#define SNSK2          C
#ifdef QTOUCH_STUDIO_MASKS
SNS_array[0][0]= 0x55
SNS_array[0][1]= 0x0
SNS_array[1][0]= 0x5
SNS_array[1][1]= 0x0
SNSK_array[0][0]= 0xaa
SNSK_array[0][1]= 0x0
SNSK_array[1][0]= 0xa
SNSK_array[1][1]= 0x0
#endif
```

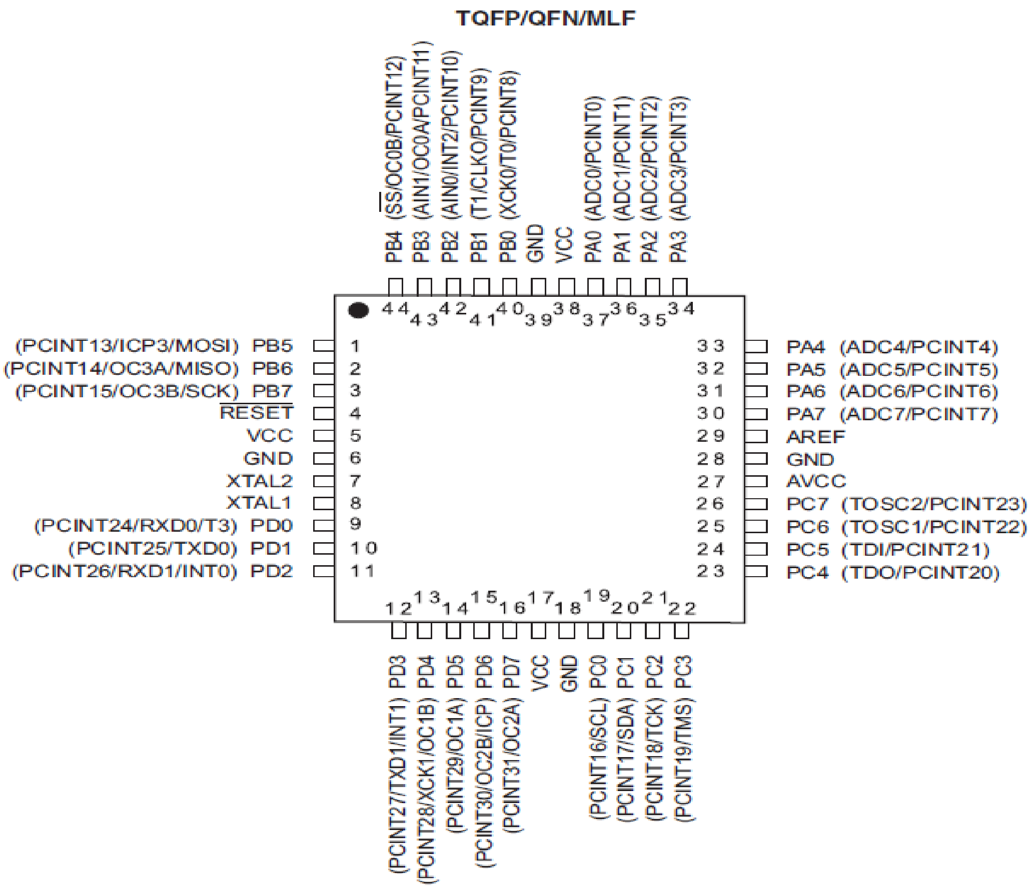


Figure 58   ATmega164PA-MU pinout (ATmega164PA Datasheet, 2.)

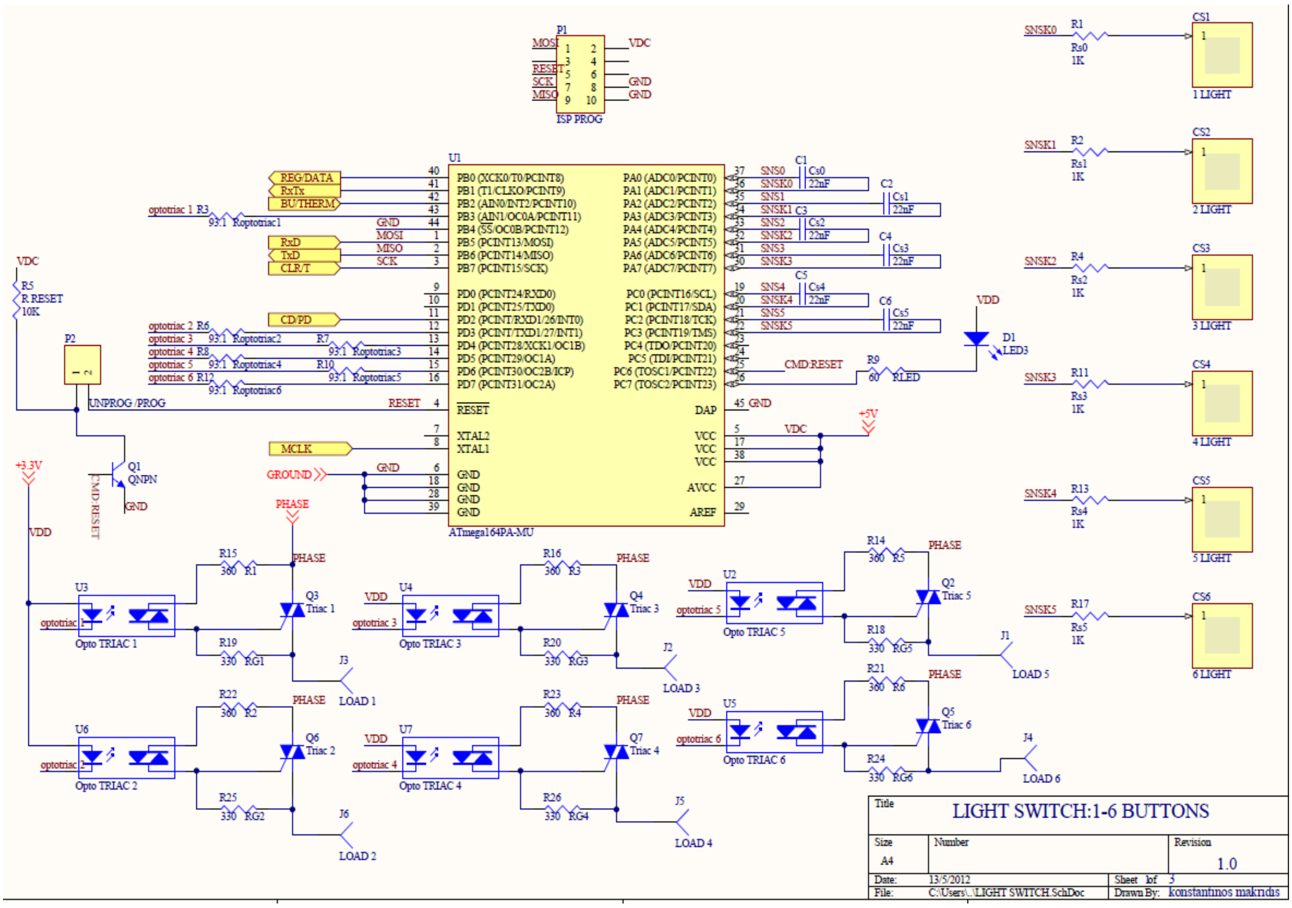


Figure 59 Light switch-ATmega164PA-MU schematic

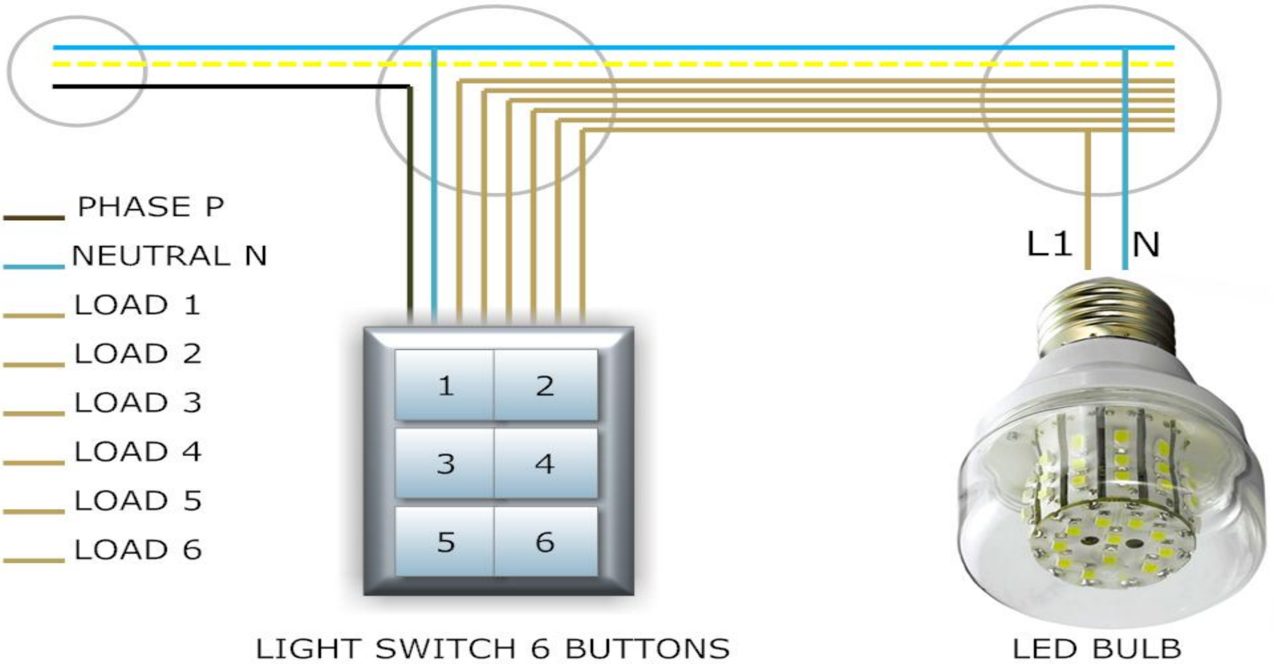


Figure 60 Electric connection of Light switch with led bulb

6.2 Socket

Socket is the second most usable device inside the house, socket device include the ST7540 circuit the power supply cicuit and the MCU who connect all the componets. There is nothing new in socket circuit because we have see them above in light switch and MDC. The maximum positions for multisocket is up to 8, the circuit for 6 positions is below.

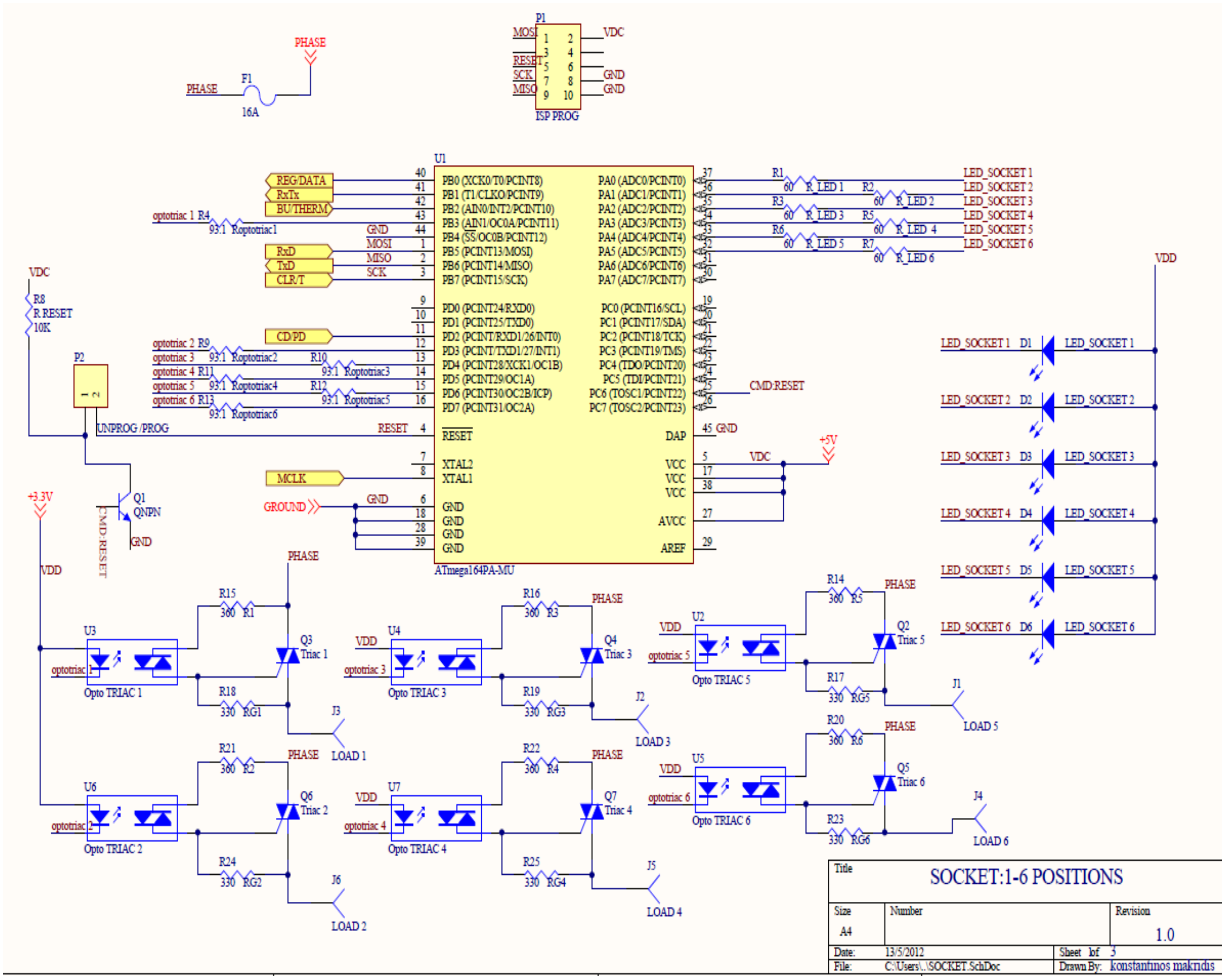


Figure 61 Multisocket-ATmega164PA-MU schematic

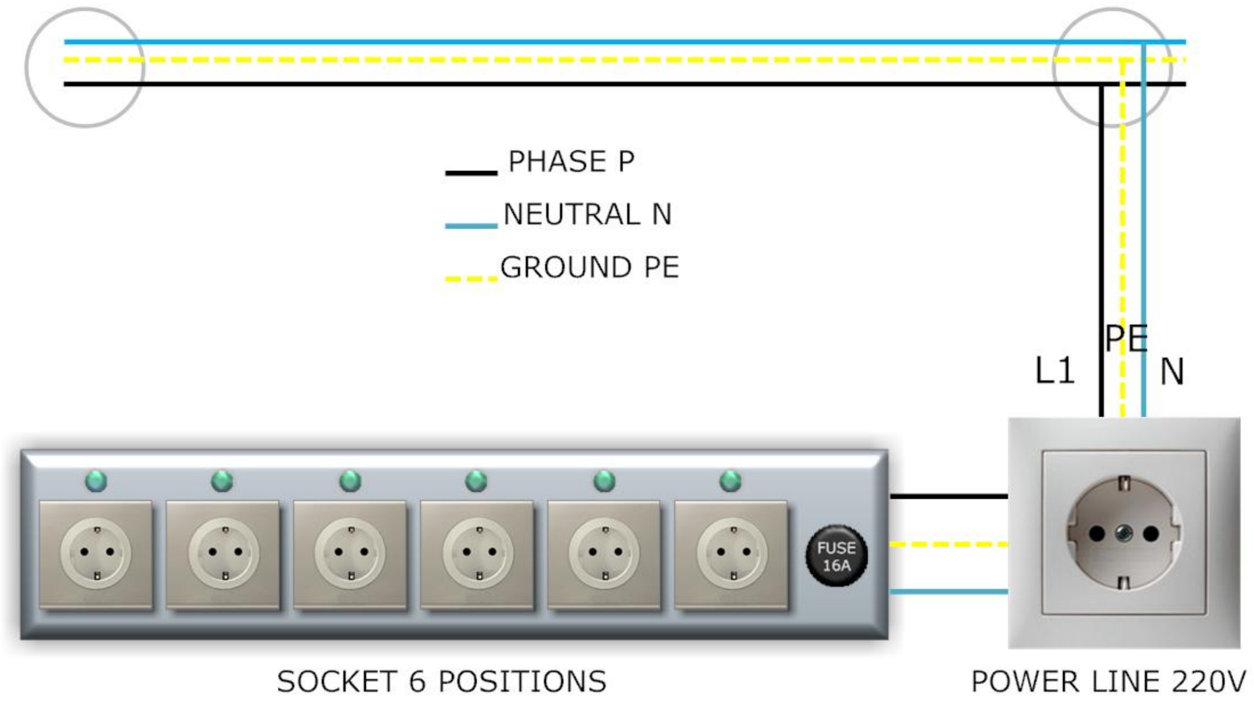


Figure 62 Electric connection of Multisocket

7 CONCLUSIONS

The time I had for final thesis was four months, inside that time I need to study and design Telecontrol, MDC, light switch and socket and after that create the O.P.R.A protocol for home automation. From design that circuits need also to create PCB boards of them, I had created both boards of Telecontrol in PCB and MDC board. Because the price for produce these PCB boards were too expensive I sent only two boards of Telecontrol in EuroCircuit Company for production. I left MDC PCB board because I have added some components more and now need to redesign the PCB board. I had problem finding company with low cost per PCB, I lost a week finding a company and after I find EuroCircuit Company I need to redesign the boards correctly to their settings. After a week from ordering day, PCB boards had arrived and then I start to solder the components. My big problem was the ATmega128RFA1 chip because was in QFN package (9mm X 9mm) and pins is so small that with common soldering station and microscope is very difficult to solder the chip. I used solder paste and with the microscope I try to put it in every pin after that I add the chip and heat it until the chip was stick to board.

When I connected AVR programmer to the ISP programmer port in Telecontrol board and try to program the chip programmer show me error message. I knew the problem was in bad soldering of the chip and to fix that problem needed a hot air station for solder smd components like that chip. Without good connectivity of the chip with the board Telecontrol device can‘t work normally as mention to work.

The conclusion in hardware side is that the first step (schematic circuit) have been done, from the second step (PCB boards) from all the boards only Telecontrol have been send for production and from Telecontrol boards the problem was the small package of the chip for that this device didn’t work.

The majority of time has been spent for design home automation protocol (O.P.R.A), because home automation protocol includes many details and allots material, the first and the main step for home automation protocol have been done in theoretical view. Is sad that haven’t achieve my goal to make an complete functional system ready to present it as a complete work, but I have prepare the ground so in near feature I will improve and complete the whole system in hardware and software part.

The conclusion in software side is that I create the base for a home automation protocol in theory. There is many details to add also in O.P.R.A protocol to be complete but I haven’t the time to present more material, so far is a good system with good possibilities for been in use worldwide. O.P.R.A system has been design for handicap people to offer them control over the whole house easy and quick and as O.P.R.A letters mean One Protocol Rule them All, may my final thesis be a good start for all the people who need a good house automation protocol.

Sources

AN2451 Application note, ST7540 FSK power line transceiver design guide for AMR. Accessed 21.2.2012
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/APPLICATION_NOTE/CD00143379.pdf

Atmel QTouch Library, User Guide. Accessed 12.3.2012
http://www.atmel.com/Images/doc8207.pdf

Atmel, single button. Accessed 30.1.2012
http://www.atmel.com/products/touchsolutions/bsw/single_touch_key.aspx

Atmel, up to 10 buttons. Accessed 30.1.2012
http://www.atmel.com/products/touchsolutions/bsw/buttons_or_10.aspx

Atmel, more than 10 buttons. Accessed 30.1.2012
http://www.atmel.com/products/touchsolutions/bsw/buttons_10.aspx

Atmel, sliders and wheels. Accessed 30.1.2012
http://www.atmel.com/products/touchsolutions/bsw/wheels_and_sliders.aspx

ATmega128RFA1 Datasheet. Accessed 27.1.2012
http://www.atmel.com/Images/doc8266.pdf

ATmega164PA MCU Datasheet. Accessed 1.5.2012
http://www.atmel.com/Images/doc8272.pdf

AVR2006, Design and characterization of the Radio Controller Board's 2.4GHz PCB Antenna. Accessed 13.3.2012
http://www.atmel.com/Images/doc8095.pdf

BP5048 Non-Isolated AC/DC Converter Datasheet. Accessed 4.2.2012
http://www.rohm.com/products/databook/pm/pdf/bp5048.pdf

Giuseppe Cantone. AN1714 Application note, ST7538 Transceiver demo-kit description. Accessed 27.4.2012
http://www.datasheetarchive.com/AN1714-datasheet.html

IS623XSM Optotriac Datasheet. Accessed 1.5.2012
http://www.farnell.com/datasheets/90959.pdf

Insteon the details, Accessed 16.2.2012
http://www.insteon.net/pdf/insteondetails.pdf

Jose Gutierrez 4.1.2003 IEEE 802.15.4 Tutorial 2003.
IEEE P802.15 Working Group for WPANs. Accessed 22.2.2012
http://odysseus.ieee.org/cs.html?url=http%3A//grouper.ieee.org/groups/802/15/pub/2003/Jan03/03036r0P802-15_WG-802-15-4-TG4-Tutorial.ppt&charset=iso-8859-1&qt=url%3Ahttp%3A//grouper.ieee.org/groups/+%7C%7C+p802.15.4+tg4+tutorial&col=sa&n=1&la=en

MC41605A6W-FPTLW 16X4 LCD Screen Datasheet. Accessed 30.1.2012
http://www.farnell.com/datasheets/1485452.pdf

QTAN0079 Buttons, Sliders and Wheels-Sensor Design Guide. Accessed 30.1.2012
http://www.atmel.com/Images/doc10752.pdf

Rohm, Non-isolated AC/DC Converter. Accessed 4.2.2012
http://www.rohm.com/products/module/power_module/ac-dc_converter/

ST7540 Power Line Modem Datasheet. Accessed 31.1.2012
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00096923.pdf

T845H-6G Triac Datasheet. Accessed 1.5.2012
http://www.st.com/internet/com/TECHNICAL_RESOURCES/TECHNICAL_LITERATURE/DATASHEET/CD00161183.pdf

UPB Technology Description, Accessed 14.2.2012
http://www.pulseworx.com/downloads/upb/UPBDescriptionv1.4.pdf

What is X10 home automation;. Accessed 17.2.2012
http://www.cleverhome.com.au/X10-home-automation.shtml

X10 (industry standard). Accessed 14.2.2012
http://en.wikipedia.org/wiki/X10_(industry_standard)

ZigBee, Accessed 19.2.2012
http://en.wikipedia.org/wiki/ZigBee

ZigBee Alliance, Accessed 19.2.2012
http://www.zigbee.org/

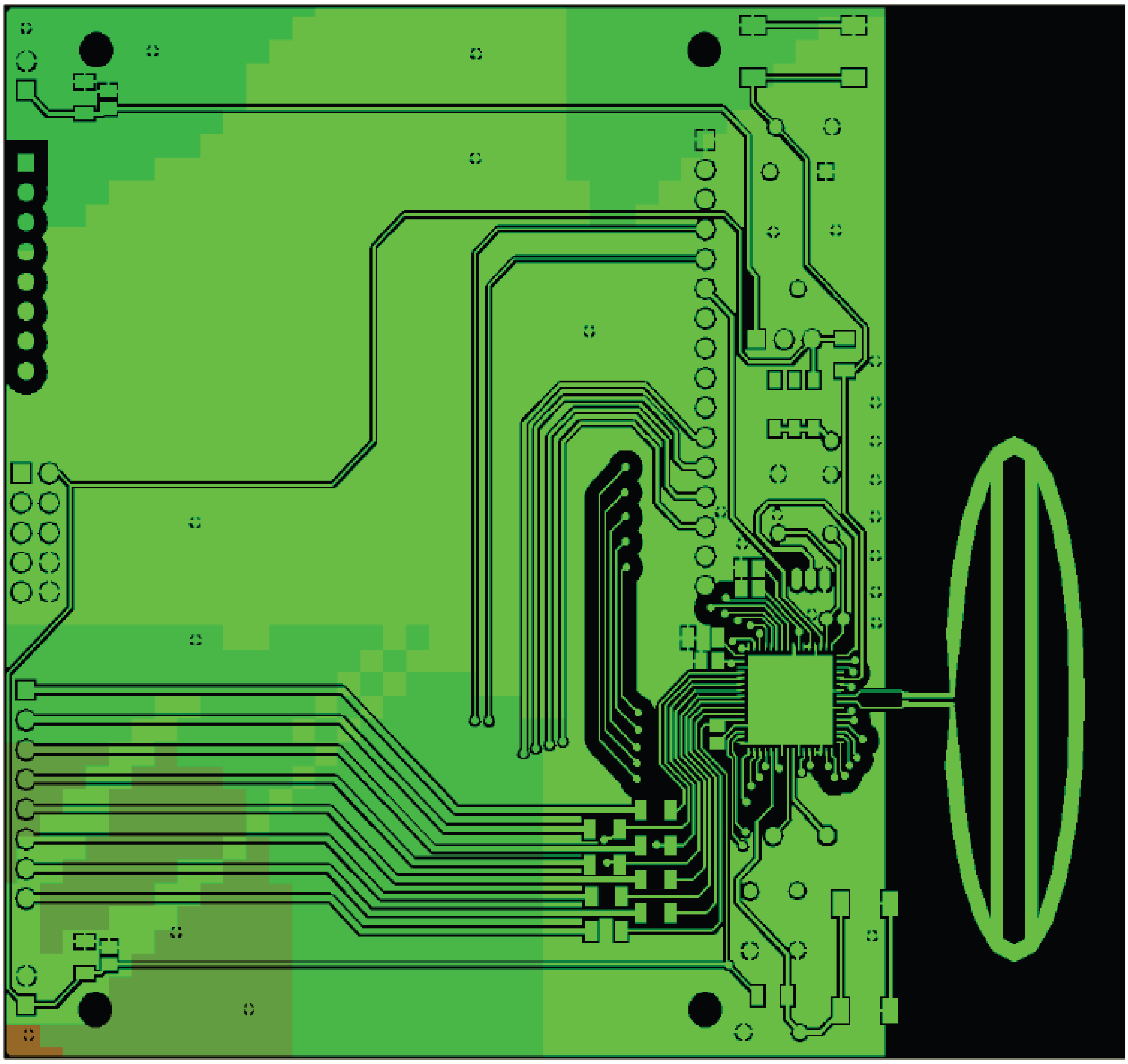
Z-Wave Protocol Overview, Accessed 20.2.2012
http://www.eilhk.com/en/product/Datasheet/Zensys/SDS10243-2%20-%20Z-Wave%20Protocol%20Overview.pdf

Z-Wave, Accessed 19.2.2012
http://en.wikipedia.org/wiki/Z-Wave

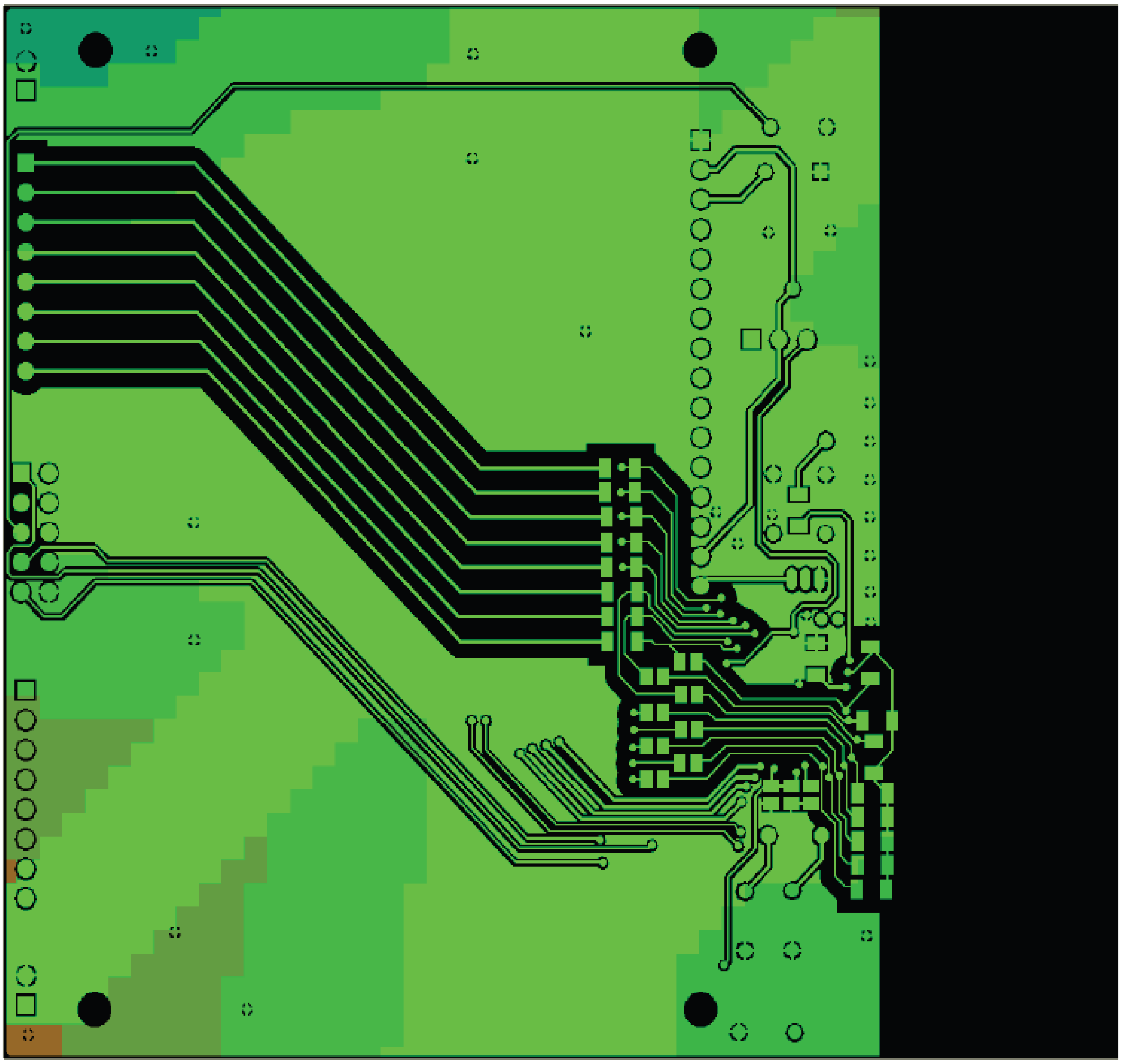
Appendix 1

Telecontrol PCB

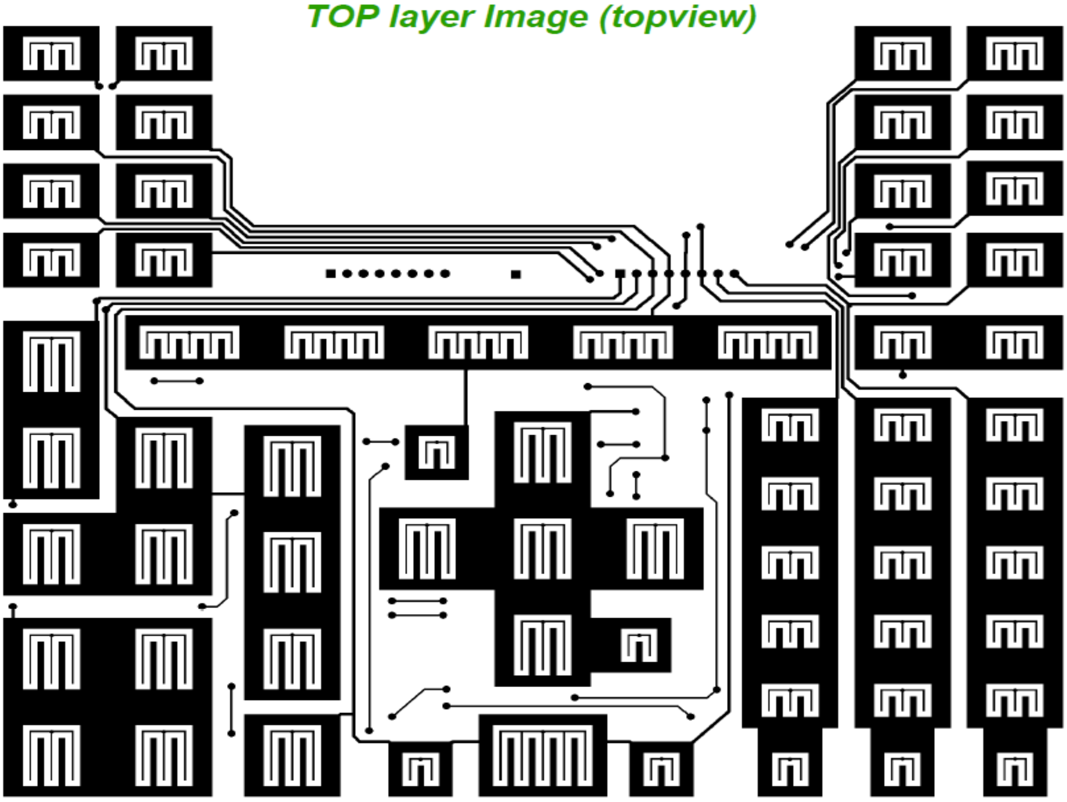
TOP layer Plating Image - PlatingIndex = 0.79 (topview)



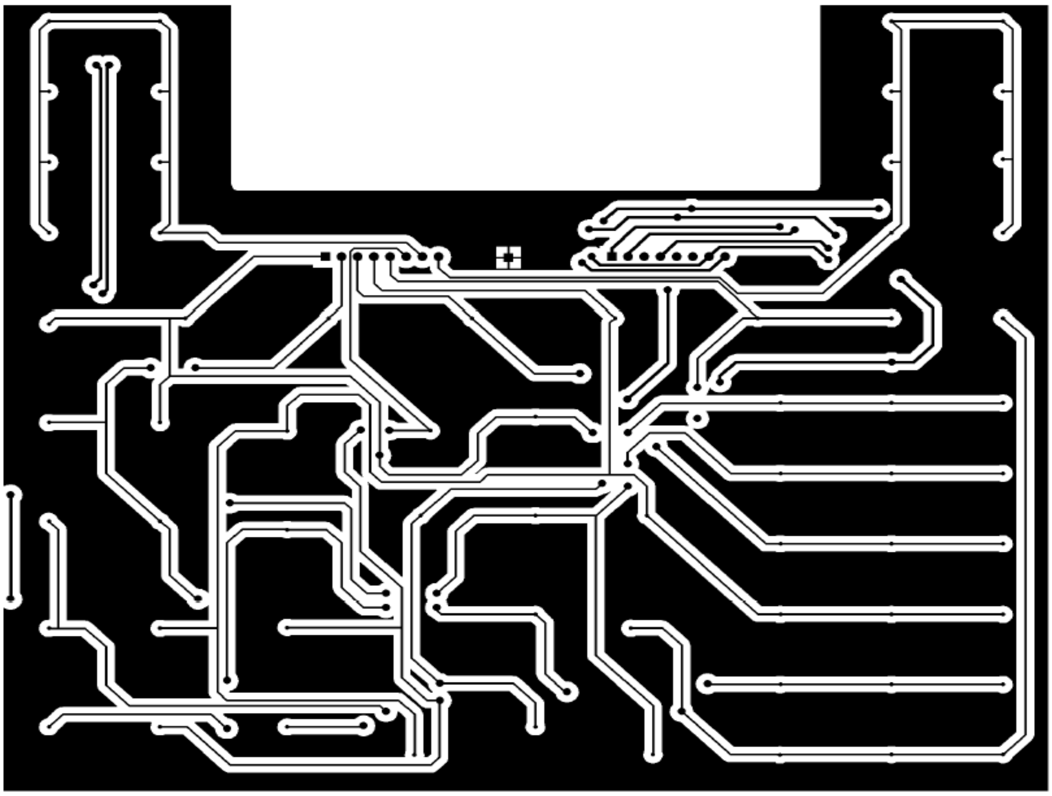
BOTTOM layer Plating Image - PlatingIndex = 0.77 (topview)

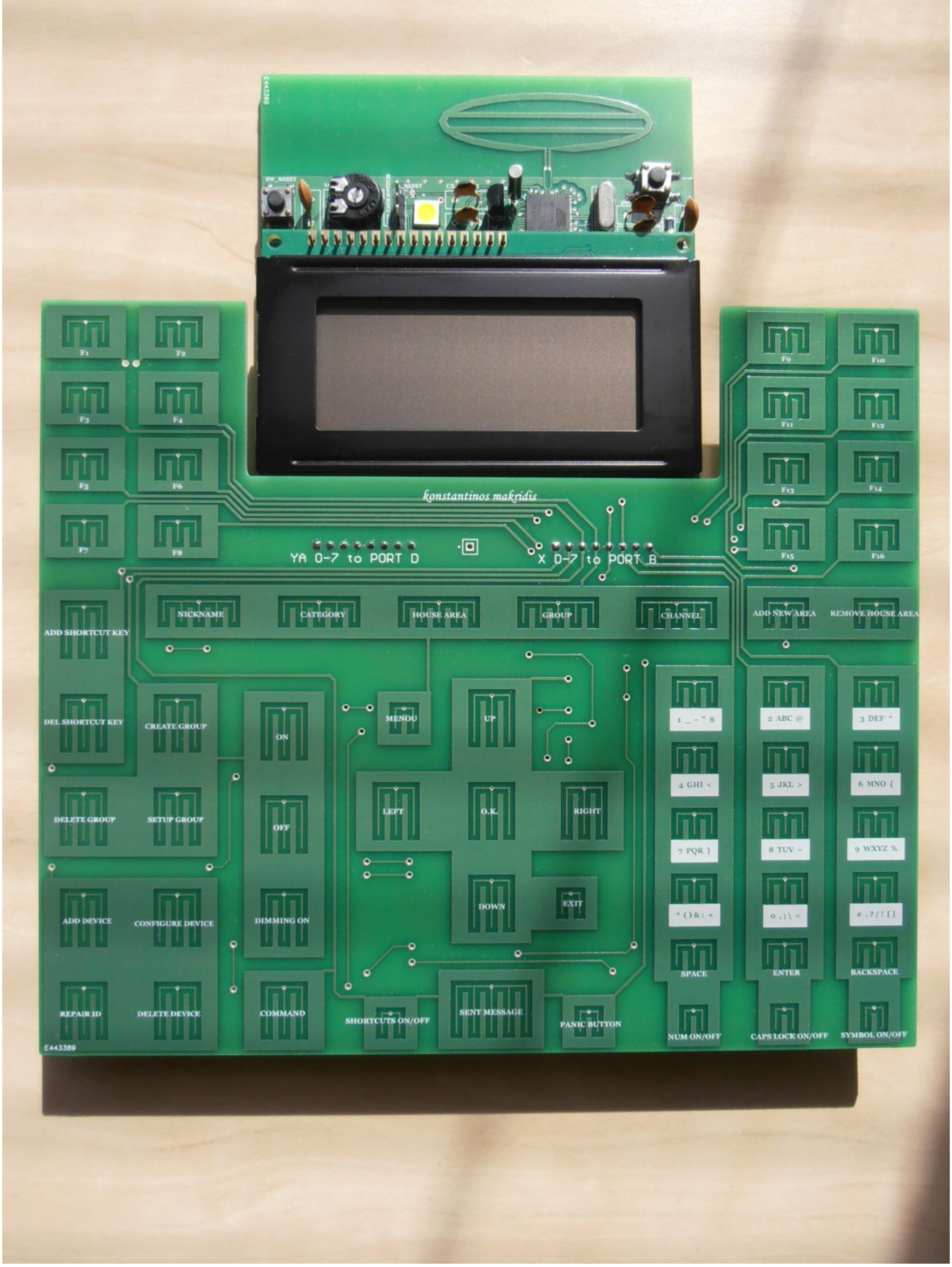


TOP layer Image (topview)



BOTTOM layer Image (topview)

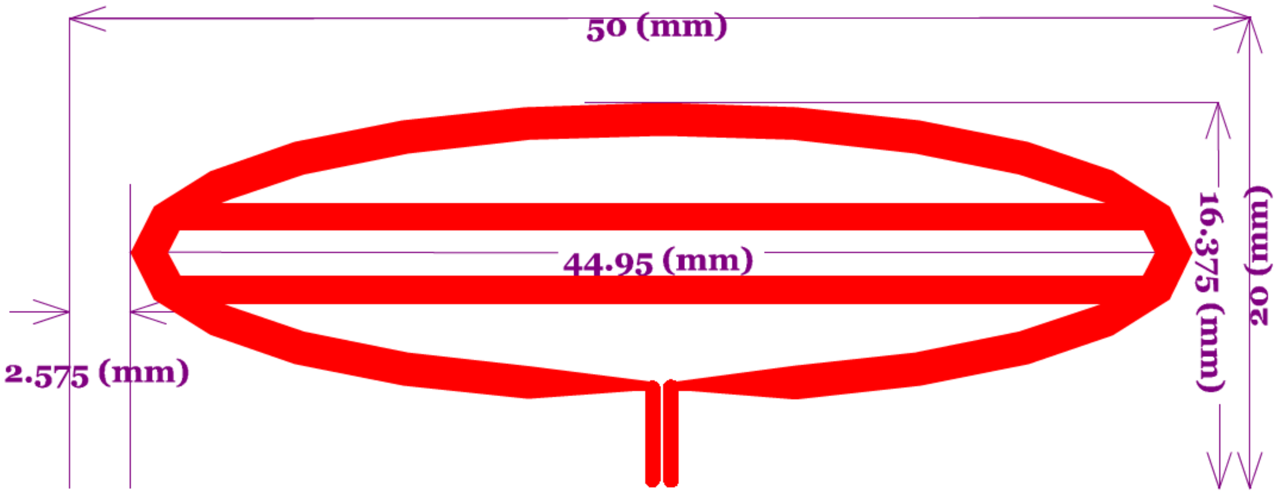




konstantinos makridis
YA 0-7 to PORT D
X 0-7 to PORT B
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12
F13
F14
F15
F16
NICKNAME
CATEGORY
HOUSE AREA
GROUP
CHANNEL
ADD NEW AREA
REMOVE HOUSE AREA
ADD SHORTCUT KEY
DEL SHORTCUT KEY
CREATE GROUP
DELETE GROUP
SETUP GROUP
ADD DEVICE
CONFIGURE DEVICE
REPAIR ID
DELETE DEVICE
ON
OFF
DIMMING ON
COMMAND
MENOU
UP
LEFT
O.K.
RIGHT
DOWN
EXIT
SHORTCUTS ON/OFF
SENT MESSAGE
PANIC BUTTON
1 _ - " $
2 ABC @
3 DEF ^
4 GHI <
5 JKL >
6 MNO {
7 PQR }
8 TUV ~
9 WXYZ %
* ( ) & : +
0 , ; \ =
. ? / ! [ ]
SPACE
ENTER
BACKSPACE
NUM ON/OFF
CAPS LOCK ON/OFF
SYMBOL ON/OFF
E443389

Appendix 2

ANTENNA PCB



For more information about how to design 2.45GHZ ANTENNA for ATmega128RFA1 and more technical details check AVR2006, Design and characterization of the Radio Controller Board's 2.4GHz PCB Antenna.

Appendix 3

Telecontrol BOM (Bill of material)

Creation Date:	14/5/2012	3:46:03 μμ
Print Date:	14-May-12	3:46:16 PM

#Column Nam	Footprint	Comment	LibRef	Designator	Description	Quantity
	C0805	VDD DEC	Cap Semi	C1	Capacitor (Semiconductor SIM Model)	1
	CAPC3216L	VDD SUPPLY	Cap Semi	C2	Capacitor (Semiconductor SIM Model)	1
	C0805	VDC DEC	Cap Semi	C3	Capacitor (Semiconductor SIM Model)	1
	CAPC3216L	VDC SUPPLY	Cap Semi	C4	Capacitor (Semiconductor SIM Model)	1
	C0805	CS7	Cap Semi	C5	Capacitor (Semiconductor SIM Model)	1
	C0805	CS6	Cap Semi	C6	Capacitor (Semiconductor SIM Model)	1
	RAD-0.2	C_SW	Cap Semi	C7, C9	Capacitor (Semiconductor SIM Model)	2
	C0805	CS5	Cap Semi	C8	Capacitor (Semiconductor SIM Model)	1
	C0805	CS4	Cap Semi	C10	Capacitor (Semiconductor SIM Model)	1
	C0805	CS3	Cap Semi	C11	Capacitor (Semiconductor SIM Model)	1
	C0805	CS2	Cap Semi	C12	Capacitor (Semiconductor SIM Model)	1
	RAD-0.2	CX_32KHZ	Cap Semi	C13, C17	Capacitor (Semiconductor SIM Model)	2
	C0805	CS1	Cap Semi	C14	Capacitor (Semiconductor SIM Model)	1
	C0805	CS0	Cap Semi	C15	Capacitor (Semiconductor SIM Model)	1
	C0805	DEC	Cap Semi	C16, C18, C19, C22, C23, C24, C25	Capacitor (Semiconductor SIM Model)	7
	RAD-0.2	CX_16MHZ	Cap Semi	C20, C21	Capacitor (Semiconductor SIM Model)	2
	HDR2X5H	SPI PROG	Header 5X2H	P1	Header, 5-Pin, Dual row, Right Angle	1
	HDR1X3	VOLTAGE SWITCH	Header 3	P2	Header, 3-Pin	1
	HDR1X2	POWER SUPPLY +3.3V	Header 2	P3	Header, 2-Pin	1
	HDR1X2	POWER SUPPLY +5V	Header 2	P4	Header, 2-Pin	1
	HDR1X8	X 0-7 to PORT B	Header 8	P5	Header, 8-Pin	1
	HDR1X8	YA 0-7 to PORT D	Header 8	P6	Header, 8-Pin	1
	HDR1X16H	LCD PORT	Header 16H	P7	Header, 16-Pin, Right Angle	1
	TO-92	2N3904	2N3904	Q1	NPN General Purpose Amplifier	1
	RESC3216L	R_RESET	Res3	R1	Resistor	1
	RESC3216L	R_LOCK ON/OFF	Res3	R2	Resistor	1
	RESC3216L	TST	Res3	R3	Resistor	1
	RESC3216L	RYAB0	Res3	R4	Resistor	1
	RESC3216L	RYA0	Res3	R5	Resistor	1
	RESC3216L	RYA1	Res3	R6	Resistor	1
	RESC3216L	RYAB1	Res3	R7	Resistor	1
	RESC3216L	RYA2	Res3	R8	Resistor	1
	RESC3216L	RYA3	Res3	R9	Resistor	1
	RESC3216L	RYAB2	Res3	R10	Resistor	1
	RESC3216L	RYA4	Res3	R11	Resistor	1
	RESC3216L	RYA5	Res3	R12	Resistor	1
	RESC3216L	RYAB3	Res3	R13	Resistor	1
	RESC3216L	RYA6	Res3	R14	Resistor	1
	RESC3216L	RYA7	Res3	R15	Resistor	1
	RESC3216L	RYAB4	Res3	R16	Resistor	1
	RESC3216L	CLKI	Res3	R17	Resistor	1
	RESC3216L	RYAB5	Res3	R18	Resistor	1
	RESC3216L	RX0	Res3	R19	Resistor	1
	RESC3216L	RYAB6	Res3	R20	Resistor	1
	RESC3216L	RX1	Res3	R21	Resistor	1
	RESC3216L	RX2	Res3	R22	Resistor	1
	RESC3216L	RYAB7	Res3	R23	Resistor	1
	RESC3216L	RX3	Res3	R24	Resistor	1
	RESC3216L	RX4	Res3	R25	Resistor	1
	RESC3216L	RX5	Res3	R26	Resistor	1
	RESC3216L	RX6	Res3	R27	Resistor	1
	RESC3216L	RX7	Res3	R28	Resistor	1
	RESC3216L	R_LED	Res3	R29	Resistor	1
	VR4	LCD CONTRAST	RPot	R30	Potentiometer	1
	PLCC-6 5050	PCB_LED	PLCC-6 5050 White Led	smd_led1	High power 0,5W led white color	1
	FSM2JSMA	SW_RESET	FSM2JSMA SMD PUSH BUTTON	SW_SMD1	PUSH BUTTON SMD	1
	FSM2JSMA	SW_LOCK ON/OFF	FSM2JSMA SMD PUSH BUTTON	SW_SMD2	PUSH BUTTON SMD	1
	MLF64-M2	ATmega128RFA1-DE	ATmega128RFA1-DE	U1	802.15.4 compliant AVR transceiver	1
	BCY-W2/D1.5	32.7KHZ	XTAL	Y1	Crystal Oscillator	1
	BCY-W2/E4.7	16MHZ	XTAL	Y2	Crystal Oscillator	1
						69