



UPPSALA
UNIVERSITET

UPTEC F16 021

Examensarbete 30 hp
Juni 2016

Implementation of Sub-GHz Real Time Radio

Lisa Åstrand



UPPSALA
UNIVERSITET

**Teknisk- naturvetenskaplig fakultet
UTH-enheten**

Besöksadress:
Ångströmlaboratoriet
Lägerhyddsvägen 1
Hus 4, Plan 0

Postadress:
Box 536
751 21 Uppsala

Telefon:
018 – 471 30 03

Telefax:
018 – 471 30 00

Hemsida:
<http://www.teknat.uu.se/student>

Abstract

Implementation of Sub-GHz Real Time Radio

Lisa Åstrand

The field of automation and smart devices is currently expanding. As most devices require wireless communication, the market for low budget, low power radios is growing rapidly. Many applications, such as the remote control of machines, have real time requirements with minimal latency. In this report, the market of wireless chips in the sub-GHz region has been investigated in order to give Syntronic AB an insight of the current market. Several key features such as range, data rate and output power were ranked among the available chips. To provide Syntronic with a marketing tool when reaching out to customers in the remote control field, a demonstration prototype (demo) was implemented using one of the radio chips from the list. The demo shows real time requirements with low latency between two wireless nodes. The first node takes movement data from a computer mouse and transmits it to the other node controlling an array of light emitting diodes (LEDs). This report contains the selection of radio configurations, design of a new wireless communications protocol, and implementation of the system in hardware and software. Measurements show a latency of 3.6 ms and a range of approximately 450 m which is regarded satisfactory.

Handledare: Lars Johansson
Ämnesgranskare: Anders Rydberg
Examinator: Tomas Nyberg
ISSN: 1401-5757, UPTec F16 021

Sammanfattning

Dagens utveckling av automationstekniker och smarta enheter ökar. I och med att många produkter använder trådlös kommunikation ökar efterfrågan av billiga, energieffektiva radiochip snabbt. Många av användningsområdena, exempelvis fjärrstyrning av maskiner, har realtidskrav som ställer villkor på låg fördröjning. I detta examensarbete har marknaden för trådlösa chip i <1 GHz frekvensområdet undersökts med syfte att ge Syntronic AB en inblick i dagens marknadsutbud. Ett antal nyckelparametrar såsom räckvidd, datahastighet, och uteffekt rankades mellan de tillgängliga chippen. För att tillhandahålla Syntronic ett marknadsföringsverktyg för nya kunder i fjärrstyrningsbranschen har en demonstrationsprototyp (demo) implementerats med ett av chippen från marknadsundersökningen. Demons syfte är att visa att chippet klarar av hårda realtidskrav med låg fördröjning mellan två trådlösa noder. Den första noden tar emot positionsdata från en datormus och sänder det till den andra noden som kontrollerar en rad av LED-lampor. Denna rapport innehåller urval av radioinställningar, design av ett nytt trådlöst kommunikationsprotokoll och implementering av systemet i hårdvara och mjukvara. Mätningar visar en fördröjning på 3,6 ms och en ungefärlig räckvidd på 450 m vilket anses vara tillräckligt.

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 1.1 | Background and Purpose | 1 |
| 1.2 | Thesis Specifications | 1 |
| 2 | Technical Background | 3 |
| 2.1 | RF Theory | 3 |
| 2.1.1 | Digital modulation | 3 |
| 2.1.2 | PN9 | 4 |
| 2.1.3 | Range | 5 |
| 2.1.4 | dBm | 5 |
| 2.2 | Spectral Access and Regulatory Standards | 5 |
| 2.2.1 | Spectral Access | 5 |
| 2.2.2 | Standards | 5 |
| 2.2.3 | RF parameters | 6 |
| 2.3 | Wireless Communication | 6 |
| 2.3.1 | Packets | 6 |
| 2.3.2 | Preamble and Synchronization word | 7 |
| 2.3.3 | Cyclic Redundancy Check | 7 |
| 2.3.4 | Wireless topologies | 8 |
| 2.4 | PS/2 Mouse Interfacing | 8 |
| 2.4.1 | Communication interface | 8 |
| 2.4.2 | Mouse movement packets | 9 |
| 2.4.3 | Mouse commands | 9 |
| 3 | Design Choices | 11 |
| 3.1 | Selection of RFIC | 11 |
| 3.2 | Modulation scheme | 12 |
| 3.3 | ETSI EN 300 220 compliance considerations | 12 |
| 3.3.1 | Data rate | 13 |
| 3.3.2 | Duty cycle | 14 |
| 3.4 | Frequency and Power | 14 |
| 4 | Demo Implementation | 15 |
| 4.1 | Hardware | 15 |
| 4.1.1 | C8051F930 MCU | 15 |
| 4.1.2 | Si4463 | 16 |
| 4.1.3 | PS/2 Mouse | 16 |
| 4.1.4 | LED array | 16 |
| 4.2 | RF Communication protocol | 17 |
| 4.2.1 | Packet structure | 17 |
| 4.3 | Development tools | 18 |
| 4.4 | Software | 19 |
| 4.4.1 | PS/2 implementation | 19 |
| 4.4.2 | RFIC interfacing | 21 |
| 4.4.3 | Main program on master | 22 |

| | | |
|----------|--|-----------|
| 4.4.4 | Main program on slave | 24 |
| 5 | Results | 26 |
| 5.1 | Verification of RF parameters | 26 |
| 5.2 | PS2 mouse interface verification | 27 |
| 5.3 | Protocol verification | 28 |
| 5.4 | Demo verification | 28 |
| 6 | Discussion | 29 |
| 6.1 | Goals met | 29 |
| 6.2 | Future work | 29 |
| | References | 30 |

Abbreviations

| | |
|-------------|---|
| API | Application Programming Interface |
| bps | Bits per second |
| Bps | Bytes per second |
| ETSI | European Telecommunications Standards Institute |
| FCC | Federal Communications Commission |
| FIFO | First In First Out |
| FSK | Frequency Shift Keying |
| GFSK | Gaussian Frequency Shift Keying |
| IC | Integrated Circuit |
| ISR | Interrupt Service Routine |
| LED | Light Emitting Diode |
| LSB | Least Significant Bit |
| MCU | Micro Controller Unit |
| MSB | Most Significant Bit |
| OOK | On-Off Keying |
| PER | Packet Error Rate |
| RF | Radio Frequency |
| RFIC | Radio Frequency Integrated Circuit |
| Rx | Receiver |
| SPI | Serial Peripheral Interface |
| sps | Symbols per second |
| Tx | Transmitter |

Notation

0x1234 Numbers starting with 0x are hexadecimal

0b1011 Numbers starting with 0b are binary

1 Introduction

1.1 Background and Purpose

Due to the home automation, remote control and smart metering trends, the market for low budget, low power devices operating in sub-GHz frequency bands is expanding [1]. There are many chip vendors offering solutions with varying performance and cost. As a result, the design process of wireless applications includes a time consuming market scan in order to select the appropriate hardware. In order to speed up the decision time, Syntronic AB has requested a market research which will identify key performance parameters common to most applications and a presentation of the currently available chips on the market.

The purpose of this thesis is to provide a market research and use the results to develop a demonstration prototype (demo) of a point-to-point radio with real time constraints. The demo will be used as a marketing tool for Syntronic AB when acquiring new customers in the remote control field.

1.2 Thesis Specifications

The demo showcases a direct link between two nodes, called *master* and *slave*. Each node consists of a micro controller unit (MCU) and a radio frequency integrated circuit (RFIC). The demo should provide the user with direct feedback of user interaction in order to demonstrate real time functionality. As user interface, a standard computer mouse is connected to the master. When moved in x-direction it sends data over the wireless link and illuminates an array of light emitting diodes (LEDs) on the slave node. Figure 1.1 shows a conceptual overview of the setup.

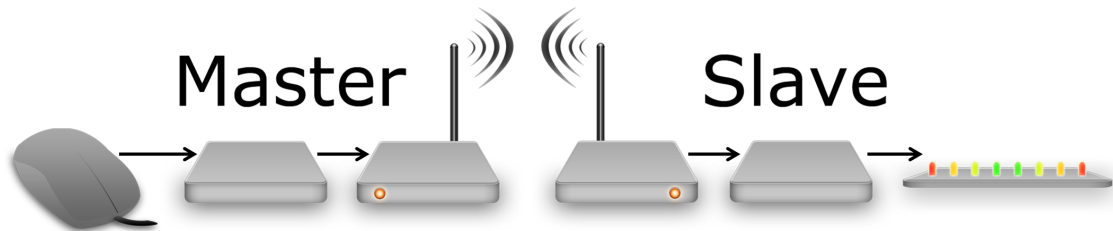


Figure 1.1: User input from the computer mouse is sent from the master MCU via the RFIC, through air to the slave which controls the LEDs accordingly.

The thesis work includes the selection of an appropriate RFIC, hardware setup, design of the wireless protocol used by the nodes, and implementing the following software on the MCUs:

- Drivers for interfacing the mouse

- Controller for LED array handling
- Drivers for interfacing the RFICs
- Main programs on both nodes

As part of the system design preceeding the implementation, a knowledge database will be set up in order to ease the design of further projects within the field. This includes an RFIC market research, and studies of regulatory standards and design considerations.

2 Technical Background

2.1 RF Theory

Radio communication is the transmission and reception of information by electromagnetic waves. The amplitude, phase and frequency of the wave is altered in order to represent the information. The central idea of a transmitter is to generate the information bearing signal, impose it onto a higher frequency carrier wave with fixed amplitude, phase and frequency, and transmit it over air. The receiver works in the opposite way, with its basic blocks shown in figure 2.1. The signal received at the antenna contains unwanted information from frequencies other than the carrier as antennas often have wide bandwidth. Therefore, a band filter damps interferers before the signal is amplified. The information signal is extracted from the RF-signal by removing of the carrier in the mixer. A narrower channel filter is applied to remove noise introduced by the mixer. At last, the information bearing signal is demodulated in the modulator/demodulator (MODEM). For digital signals, the modem includes an analog to digital converter (ADC). The transmitter consists of the same blocks with the exception of a digital to analog (DAC) converter replacing the ADC [2].

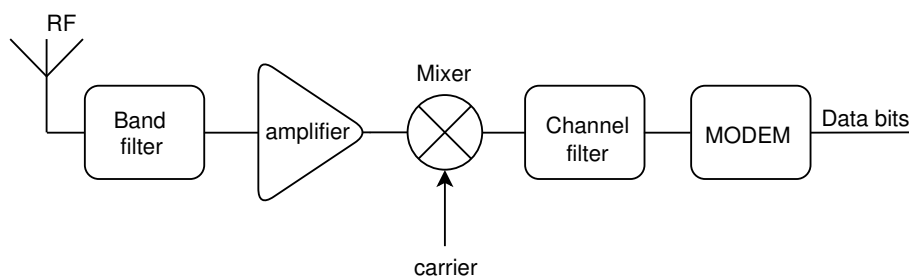


Figure 2.1: Basic blocks of a simple receiver.

2.1.1 Digital modulation

The imposing of a digital data bit stream into the carrier is known as modulation. There are different modulation schemes which change the frequency, amplitude or phase of the carrier depending on the data. There are also schemes which utilize combinations of the three. In this section, two of the simpler schemes known as on off keying (OOK) and frequency shift keying (FSK) will be explained. Figure 2.2 shows both schemes in time domain.

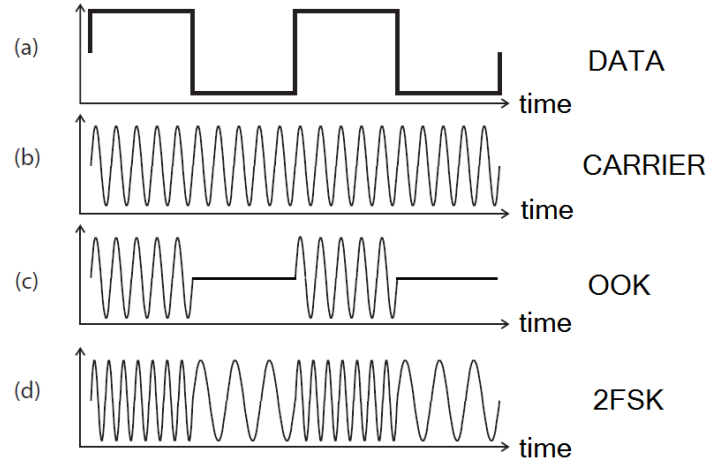


Figure 2.2: Stages of digital modulation: (a) data bit stream, (b) carrier wave, (c) carrier modulated using OOK, (d) carrier modulated using 2FSK.

On Off Keying

OOK is the simplest form of modulation where the carrier is turned on in order to transmit a 1, and turned off in order to send a 0. During on-times, the amplitude is constant.

Frequency Shift Keying

In FSK, each symbol is mapped to a unique frequency which the carrier is modulated into. During transmission, the amplitude of the modulated signal is kept constant [2]. The simplest form of FSK is binary FSK (2FSK), which uses only two symbols, 1 and 0. Hence, the carrier takes one of the two frequencies

$$f_1 = f_c + \Delta f \quad \text{and} \quad f_2 = f_c - \Delta f$$

where f_c is the carrier frequency and Δf is the frequency deviation [3].

If higher data rates are required, 4FSK can be used. Each symbol is two bits long resulting in four possible symbols and frequencies. This way, twice the data can be transmitted within the same time frame using 4FSK as opposed to 2FSK.

In order to reduce the side lobes created by FSK in the frequency spectrum, the digital bit stream can be Gaussian filtered before modulation. This way, the carrier's transition between frequencies is smoothed out, resulting in lower interference at adjacent channels.

2.1.2 PN9

As the phase shifts of the carrier will be different when transitioning from 0 to 1 and 1 to 0 etc., it is important to have an evenly distributed bit pattern when measuring the spectral properties of the transmitter. To achieve this, pseudo-random bit sequences are used as data. One commonly used sequence is the PN9 sequence which generates the values 1 through 511 in a pseudo-random order. [4]

2.1.3 Range

When propagating through free-space without any reflecting objects, a wave lose energy according to Frii's equation

$$P_{Rx} \propto \frac{P_{Tx}}{d^2 f^2}$$

where P_{Rx} is the power at the receiving node, P_{Tx} the power at the transmitting node, f the carrier frequency and d the distance between the nodes [5].

2.1.4 dBm

In RF measurements, power is often measured in dBm [6]. It is defined as the ratio between the measured signal and a reference of 1 mW,

$$P_{dBm} = 10 \log \frac{P}{0.001}$$

2.2 Spectral Access and Regulatory Standards

2.2.1 Spectral Access

As the use of wireless devices increase rapidly, there is a need for organizing how the RF-spectrum is shared. In most countries, the allocation of spectrum is controlled by governmental agencies. The spectrum is shared among commercial and military use, where the commercial frequencies is further divided into licensed and unlicensed frequency bands. In Sweden, the licensed bands are auctioned to the highest bidder [7]. The license owner is then free to plan the network and standards in the band by themselves. By comparison, the unlicensed bands are free to use by everyone. However, certain regulations applies, limiting the maximum allowed power etc. Additionally, as the bands are free to use by anyone, there is a high risk of interference from other users [8].

The combined group of license free bands in the sub-GHz region is called the Industrial, Scientific and Medical (ISM) bands. In Europe, they include the 169, 433, and 868 MHz bands [9].

2.2.2 Standards

In Europe, the use of devices radiating in the license free ISM bands is regulated by the European Telecommunications Standards Institute (ETSI) whereas the US is regulated by the Federal Communications Commission (FCC). In this thesis, focus has been on the European market, hence only ETSI standards are considered. For sub-GHz devices the applicable regulation document is ETSI EN 300 220. The document state limits for

transmitted power, allowed operational frequencies, duty cycles, receiver performance etc [10].

In addition to regulating transmitter parameters, ETSI also provides three receiver categories which devices can fulfill. The categories state how robust a communications link is. For applications having human life implications, category one is required.

2.2.3 RF parameters

In order to measure the performance of transmitters and receivers, several properties are used.[11]

Receiver

Sensitivity

The minimum power required by a signal to be detected by the receiver. Higher sensitivity results in longer range.

Adjacent channel selectivity

The receiver's capability to receive a wanted signal in the presence of a strong unwanted signal in a neighboring channel.

Blocking

The receiver's capability to receive a wanted signal in the presence of a strong unwanted signal at a distant frequency.

Transmitter

Frequency offset

The difference between the desired carrier frequency and the actual transmitted carrier frequency.

Output power

The maximal power of the signal sent by the transmitter. Higher output power gives longer range.

Adjacent channel power

The amount of power sipping out into adjacent channels.

Duty cycle

The ratio between transmitter on- and off-time.

2.3 Wireless Communication

2.3.1 Packets

Upon transmission, data is often put together in *packets*. Other from the data itself, the packets contain information about the data. This information is sent right before the data and is called the *header* of the packet. The amount of data and type of information in each packet varies for different applications. Common header content is information about where the data comes from, whom is supposed to receive the data, the length and type of data, etc. At the end of the packet it is common to include some sort of error checking functionality. All the content which is not actual data is called *data overhead*.

2.3.2 Preamble and Synchronization word

In order to synchronize the transmission of a data packet between two nodes, a transmitter can send a preamble. A preamble is a known sequence of bits which prepares the receiver for incoming data. At the end of the preamble, right before the packet header bits, a synchronization word different from the preamble pattern is sent.

2.3.3 Cyclic Redundancy Check

In order to confirm that data sent over unreliable communication channels is received properly, error detection techniques are used. One of the more common error detection methods is the cyclic redundancy check (CRC) as it can be implemented in hardware using only shift registers and XOR-gates [12]. CRC is based on long division of the data bit stream by a generator code. However, instead of using subtraction, the remainder is generated with the use of XOR.

A common notation for generator and data bits is on polynomial form. For instance, the data bit stream 10011 would be described by the polynomial $x^4 + x^1 + x^0$ where a one indicates a representation in the polynomial. The numeric value is given by substituting x by 2: $2^4 + 2^1 + 2^0 = 19$ [13].

As an example, consider the data bit stream $D = \{0b1101\}$ and a generator code $G = \{0b101\}$. Before division, D is multiplied with the highest order of G , $x^2 = 4$, in order to extend the bit stream to give room for the CRC. The new data stream, 0b110100 is divided by the generator code using long XOR-division, see table 2.1a. The remainder, 0b10, is therefore the CRC which will be transmitted at the end of the data. Upon reception, the data and CRC bits, 0b110110, are XOR-divided by the same generator code, see table 2.1b. If the remainder is non-zero, the data contains errors.

Table 2.1: (a) CRC generator using XOR-division and
(b) CRC error detection using XOR-division.

| (a) | (b) |
|--|---|
| $ \begin{array}{r} 110 \\ 101 \overline{)110100} \\ \underline{101} \\ 111 \\ \underline{101} \\ 10 \leftarrow \text{CRC} \end{array} $ | $ \begin{array}{r} 111 \\ 101 \overline{)110110} \\ \underline{101} \\ 0111 \\ \underline{101} \\ 0101 \\ \underline{101} \\ 0 \text{ No errors} \end{array} $ |

If many bits are inverted, some combinations of data will give the same CRC result as the error-free data. The number of bit errors the CRC can detect depend on the generator code's Hamming distance [14]. A Hamming distance of 3 will detect all combinations of 1- or 2-bit errors whereas at least one combination of 3-bit errors will be undetected.

2.3.4 Wireless topologies

Wireless communication can take place in many different topology constellations. Figure 2.3 shows three examples; *star* systems where one transmitter sends information to several receivers, *fully connected* systems where each node can communicate with every other node in the system and *point-to-point* systems with only two nodes.

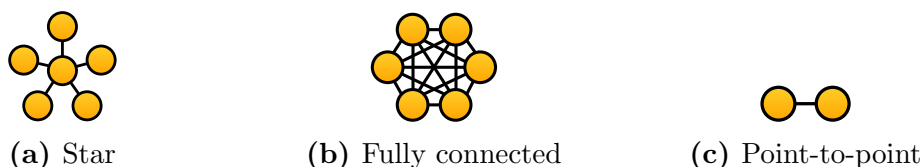


Figure 2.3: Examples of three different network topologies.

2.4 PS/2 Mouse Interfacing

The PS/2 computer mouse uses the PS/2 communications protocol for sending data and receiving commands. In the following sections, the PS/2 protocol will be explained as well as the operation of the mouse.

2.4.1 Communication interface

The PS/2 protocol is a synchronous bidirectional means of communication between two nodes. It uses a host-device configuration with four wires; V_{cc} , GND, CLK and DATA. Both CLK and DATA are tied to V_{cc} through pullup resistors.

The device generates the clock regardless of the direction of data transmission. However, the host has ultimate control and can inhibit communication at any time by pulling CLK low.

Transfer from device to host differs slightly from host to device. Common for both directions is the 11 bit frame structure containing a start bit, one data byte, a parity bit and a stop bit, see figure 2.4. Data bytes are sent LSB first. The host reads and writes data on the falling edge of CLK whereas the device uses the rising edge.

Bit errors are detected through a parity checksum. All data bytes are summed and if the result is even, the parity bit is set to 1, otherwise it is set to 0. At the receiving side, the same summation is performed and if the result is even, the frame has been corrupted during transmission.

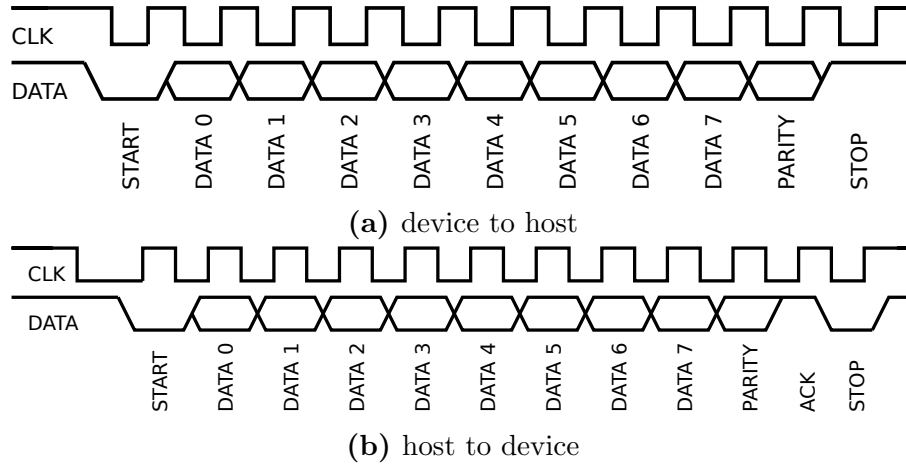


Figure 2.4: PS/2 protocol

2.4.2 Mouse movement packets

The mouse samples its position using two counters; one for horizontal (x) and one for vertical (y) movement. The counters contain the distance the mouse has moved since the last movement sample. At each sample, the status of its three buttons is checked as well. x- and y-movements are stored as 9 bit signed values resulting in the interval -256 to 255. Overflow flags are set if the mouse moves too fast for the counters [15].

Each time the mouse is either moved or one of its buttons is pressed, it sends a 3 byte movement packet as seen in figure 2.5 to the host via PS/2 protocol. Note that the MSB of the x- and y-movement counters are stored in the first byte [16].

| Bit no: 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|------------|------------|--------|--------|---|---------------|--------------|-------------|
| y overflow | x overflow | y sign | x sign | 1 | middle button | right button | left button |
| x-movement | | | | | | | |
| y-movement | | | | | | | |

Figure 2.5: Three byte movement packets sent from the mouse whenever it is moved. Note that bits 5 and 4 in the first byte make up the 9th bit in the x- and y-movement counters.

2.4.3 Mouse commands

Apart from sending movement data, the mouse can receive commands from the host. Table 2.2 shows a list of the most commonly used commands. The mouse responds to every command by sending an acknowledgment (ACK) followed by other optional data.

Table 2.2: List of commands sent from the host and the mouse's responses.

| Hex Code | Command | Response |
|----------|------------------|--------------|
| 0xFF | Reset | ACK and 0xAA |
| 0xF2 | Get ID | ACK and ID |
| 0xF3 | Set sample rate | ACK |
| 0xE8 | Set resolution | ACK |
| 0xF4 | Enable streaming | ACK |

3 Design Choices

This chapter will describe the various choices in both hardware and software made before the implementation of the demo.

The starting point of the systems design was the following parameters:

- 10 - 100 kbps data rate
- Long range
- Robustness
- Operation in the European region's license free sub-GHz ISM bands

3.1 Selection of RFIC

A thorough market research was done before the selection of the RFIC. In order to ease the selection for this and future projects, a list of over one hundred RFICs operating in the sub-GHz frequency region was put together. As the demo's application will involve bidirectional communication, only transceivers were considered.

The highest robustness is achieved by selecting an RFIC capable of reaching ETSI category 1. The need of European operation limits the selection to RFICs operating in either 433 or 868 MHz license free bands. Long range is possible with high transmitter output power and high receiver sensitivity.

Using the list of RFICs, the group of possible choices was narrowed down to six different chips from two manufacturers; Texas Instruments and Silicon Laboratories. All six can reach ETSI category 1 and supports operation in the desired frequency bands. See table 3.1 for a comparison of the RFICs.

Table 3.1: Comparison of possible RFICs from Silicon Laboratories (Sixxxx) and Texas Instruments (CCxxxx).

| Chip ID | Max. Tx power | Rx sensitivity | Max. data rate | Price |
|---------|---------------|----------------|----------------|--------|
| CC1125 | 16 dBm | -129 dBm | 200 kbps | \$3.50 |
| CC1120 | 16 dBm | -127 dBm | 200 kbps | \$3.00 |
| Si4468 | 20 dBm | -133 dBm | 1 Mbps | \$2.62 |
| Si4467 | 13 dBm | -133 dBm | 1 Mbps | \$2.63 |
| Si4464 | 20 dBm | -126 dBm | 1 Mbps | \$2.91 |
| Si4463 | 20 dBm | -126 dBm | 1 Mbps | \$2.42 |

CC1125 can reach ETSI category 1 in all frequency bands and a document providing instructions on how to reach it is provided for the 868 MHz frequency band. However, CC1125 is inferior to Si4463/4/8 in both Tx output power and data rate.

CC1120 can reach ETSI category 1 in 433 MHz frequency band but there is no documentation of board design available.

Si4468 has the highest performance in the considered areas. However, it can only reach ETSI category 1 under certain conditions and there are no available evaluation kits in the 433- nor 868 MHz frequency bands.

Si4467 has the lowest output power and might not provide sufficient range.

Si4464 perform equally as Si4463 in the considered areas. However, Si4464 is mainly intended for operation in unusual frequency bands and the manufacturer recommends using Si4463 for operation in 433- and 868 MHz bands.

Si4463 is superior to both chips from Texas Instruments in output power, data rate and price. A development kit with ETSI category 1 compliance in the 868 MHz frequency band is available.

Si4463 is chosen as it fulfills the technical requirements and is easily accessible as a development kit with extensive documentation and custom software development tools.

3.2 Modulation scheme

The chosen RFIC supports 2(G)FSK, 4(G)FSK and OOK. As OOK requires a wider bandwidth for the same data rate as FSK, it needs a wider Rx filter, resulting in an increased noise floor and decreased sensitivity. For the same data rate, 4FSK needs only half the bandwidth of 2FSK, resulting in +3 dB increased sensitivity as the Rx filter's bandwidth is halved. However, the inner deviation is significantly less in 4FSK having even greater impact on the sensitivity, finally resulting in an overall decrease. To ease compliance with ETSI category 1 regarding spurious emissions and adjacent channel power Gaussian filtering is used even though it slightly reduces sensitivity.

As high sensitivity was the main focus, 2GFSK was considered the best modulation.

3.3 ETSI EN 300 220 compliance considerations

When developing a new device operating in the 433 and 868 MHz bands, much effort must be taken to make sure the design complies with the regulatory standards. A part of this thesis has been dedicated to produce documents simplifying ETSI EN 300 220. As the designer often knows which frequency is desired, visual representations such as the one in figure 3.1 provides fast information on the most important transmitter parameters. This way, regulatory compliance gets involved at an earlier stage in the design process, potentially saving money and time.

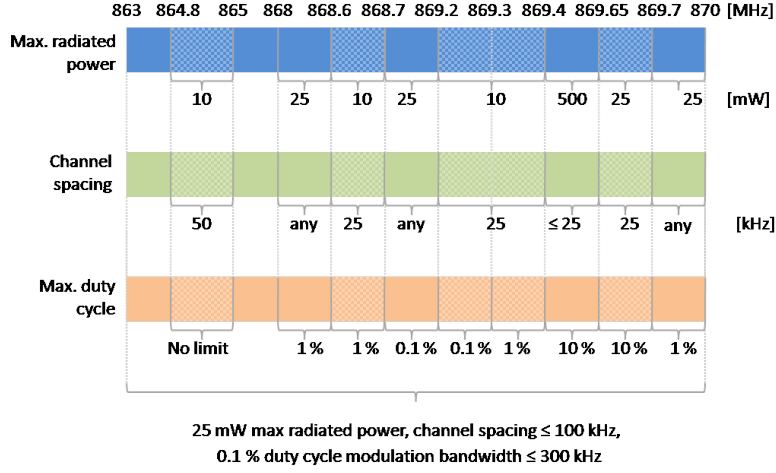


Figure 3.1: A visual representation of the transmitter regulations in ETSI EN 300 220 simplifies the design process at an early stage.

3.3.1 Data rate

The chosen RFIC can reach ETSI category 1. However, according to the manufacturer the data rate must be kept below 4 kbps in order to comply with ETSI category 1 if used in the 868 MHz ISM band. With the radio DAT packets being 14 bytes and ACK packets being 11 bytes in total (see section 4.2.1), the theoretical time for transmission of a single packet at 4 kbps is

$$\frac{(14 + 11) \cdot 8 \text{ bits}}{4000 \text{ bps}} \approx 50 \text{ ms}$$

Given a sample rate of 60 Hz, the mouse sends a new packet every 16.7 ms. The transmission of each byte takes approximately 0.88 ms at a PS/2 clock frequency of 12.5 kHz, see figure 3.2. A total mouse packet takes 2.9 ms leaving 13.8 ms for the MCU to handle other tasks. Obviously, this time is far lower than the 50 ms required by the radio to forward the packet at 4 kbps. Counting backwards, the mouse sample rate would need to be below 20 Hz to allow enough time for the radio to finish a transmission. Another solution would be to shorten the length of packets. The maximum combined length of DAT and ACK packets when the data rate is 4 kbps and mouse sample rate 60 Hz would be

$$\frac{x \cdot 8 \text{ bits}}{4000 \text{ bps}} = \frac{1}{60 \text{ Hz}} - 2.9 \text{ ms} \rightarrow x = 7 \text{ bytes}$$

A sample rate below 20 Hz would introduce noticeable delays in the demo which is undesirable. Furthermore, the Si4463 needs at least 20 bits preamble per packet which leaves two bytes for headers and payload [17]. As the mouse data itself is three bytes, this reduction of packet size is impossible.

The most important feature of the demo is the smoothness experienced by the user, therefore it was decided to use a data rate of 40 kbps.

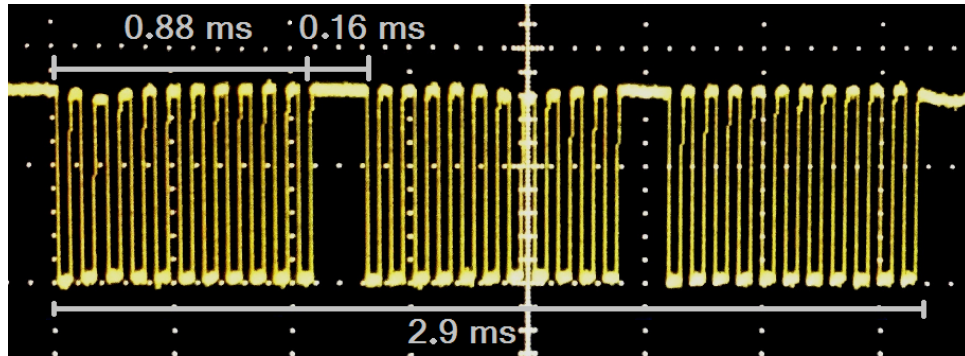


Figure 3.2: Timing of the PS/2 clock when the mouse sends a movement packet containing three eleven bit frames. Each frame takes 0.88 ms which, including 0.16 ms pauses between frames, sums up to a total of 2.9 ms.

3.3.2 Duty cycle

If moved continuously, the mouse sends movement packets at 60 Hz frequency. At the chosen 40 kbps data rate, each packet including its acknowledgment takes approximately 5 ms of transmission time. Over an hour, the total transmitter on time would be 1080 s resulting in a 30 % duty cycle. Despite exceeding the regulatory limit of 1 %, no listen before talk or duty cycle limiter is implemented in the demo. As the mouse will not be used continuously for one hour long periods, the actual duty cycle is expected to be much lower.

3.4 Frequency and Power

After the Si4463 was chosen, the choices of operational frequency was limited to the 868 and 433 MHz bands. Despite 433 MHz giving a longer range, 868 MHz was chosen as there was a development kit available operating at 869.3 MHz. Considering ETSI EN 300 220, the output power was set at the maximum allowed 10 dBm.

4 Demo Implementation

The demo consists of two wireless nodes; the *master* and *slave*. As both are based on the same development kit, the RFIC-MCU interface is similar on both devices and will be presented together. The main programs and the surrounded peripherals do however differ and will be presented in separate chapters.

4.1 Hardware

For both the master and slave, Silicon Laboratories' Wireless Motherboard (WMB) development kit is used, see figure 4.1. Each board includes a C8051F930 MCU, Si4463 RFIC, on board debugger, programmer, 132×24 pixel LCD, 4 LEDs, 4 push buttons and an 868 MHz antenna. The board is powered via USB or two AA batteries located on the back [18].

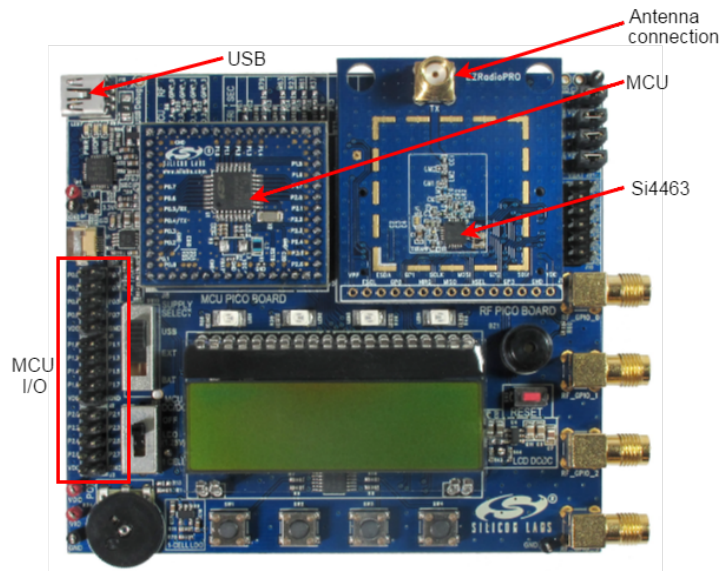


Figure 4.1: Wireless motherboard platform from Silicon laboratories. [19]

4.1.1 C8051F930 MCU

The development kit comes mounted with a C8051F930 MCU. It holds a 8-bit 8051-architecture microprocessor. Other features relevant for the demo are:

- Two external interrupts
- Two serial peripheral interface (SPI) controllers
- Two 8-bit timers
- Two 16-bit timers
- 24 general purpose input/output (GPIO) pins

- 64 kB flash program memory
- 256/4096 B internal/external data RAM

4.1.2 Si4463

The exchange of information between the MCU and Si4463 is mainly done through the four wire SPI bus controlled by the MCU. Four GPIO pins can be used to access special functions. Additionally, Si4463 can issue interrupts to the MCU on the active low NIRQ pin and the MCU can force Si443 into shutdown by pulling SDN high. See figure 4.2 for the entire connection between the MCU and Si4463. The SPI bus is used for both commands and radio data.

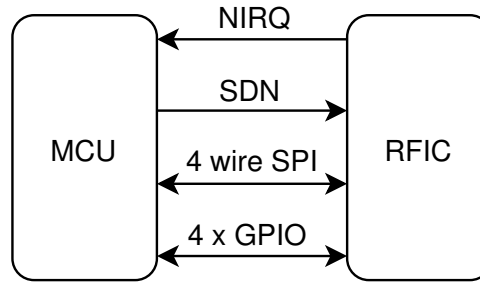


Figure 4.2: Hardware connection between the MCU and Si4463 RFIC.

Si4463 contains two 64 byte long first in first out (FIFO) shift registers; one for received data and one for data waiting to be transmitted. Furthermore, it contains a packet handler which automatically adds preamble, synchronization word, etc. to data before transmission. The properties of Si4463 are held in the application programming interface (API) and are accessed by the MCU by sending a set/get property command over SPI.

4.1.3 PS/2 Mouse

The mouse used is a Logitech RX250. The cord's PS/2 connector is cut off and replaced by jumper wires. Vcc and GND are connected to the WMB's +3.3 V power breakout pins. The clock and data wires are connected to GPIO pins on the MCU, pulled to Vcc through internal pullup resistors. As the mouse's controller chip is very sensitive to wrong supply voltage polarization, the mouse's chassis was removed and a multimeter was used to verify all connections.

4.1.4 LED array

Due to limitations in available GPIO pins on the MCU, the array consists of eight LEDs. Each LED is simply connected between ground and an GPIO pin in series with a 220 Ω resistor.

4.2 RF Communication protocol

To reduce the load on the MCUs and maintain the simplicity of the data transfer, a new RF communication protocol was designed instead of using any of the existing standards. The protocol is designed as a point-to-point system with a master-slave configuration, see figure 4.3. The master sends mouse movement data in DAT radio packets whereas the slave's only function is to respond with ACK radio packets. If no ACK is received within a set time, the packet is considered lost. The protocol does not implement re-transmissions as it would require newer data to be buffered, going against the real time requirements.

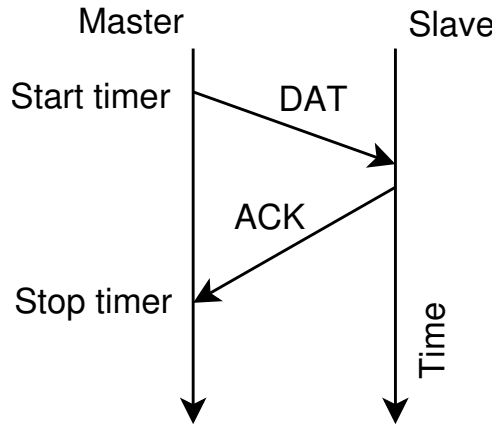


Figure 4.3: Time diagram of data transmission with ACK response. A timeout timer is started at transmission and stopped at reception of an ACK.

Each radio data packet contains exactly one mouse movement packet. To reduce data overhead, several movement packets could be buffered in the host MCU before being sent in a combined radio data packet. However, this approach would interfere with the real time requirements where the wireless link should keep the impact on the flow of movement data to a minimum.

4.2.1 Packet structure

The packets sent between master and slave are inspired by the IEEE 802.15.4g standard's packet structure [20]. However, the packets have a smaller data overhead in order to lower the latency. Both DAT and ACK radio packets use the same packet structure, only the content field differs, see figure 4.4. The synchronization header consists of a four byte standard 0101... preamble pattern and a two byte long synchronization word, 0xB42B. The length of the preamble is chosen from the manufacturer's recommendations. The physical header contains one identification byte in which the four first bits represents the destination ID and the four last represent the source ID. The master has ID 0x4 and the slave has ID 0x2. The length byte specifies the length of the following packet. Furthermore, the next byte contains two bits to specify whether the packet contains DAT (0b00) or ACK (0b10). The remaining six bits are used for sequence number counting. For

DAT packets, the content field consists of the three byte long mouse movement packets whereas it is empty for ACK packets. Lastly, there is a two byte long CRC field for error detection.

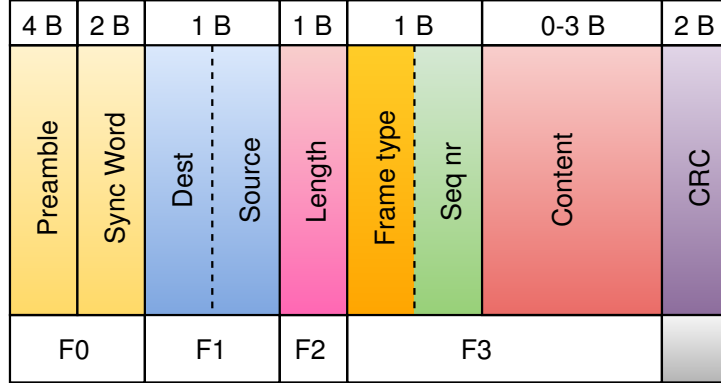


Figure 4.4: Structure of packets sent between the master and slave over the wireless link. Lengths vary from 11 - 14 bytes. Packets are sent from left to right.

The packets are divided into four fields. In order to reduce the load on the MCU, the following tasks are performed by Si4463's internal packet handler:

- Field 0 is constructed and added to the beginning of each packet without any involvement from the MCU.
- Upon reception of a packet, the packet handler compares field 1 with a stored match byte to make sure the source and destination addresses are correct. If not, the packet is thrown away without the MCU taking notice.
- As packets can be of varying length, the packet handler uses the value in field 2 to know how long the remaining packet is when receiving. Field two is not stored in the Rx FIFO as it is of no interest to the MCU. However, the length byte must be loaded into the Tx FIFO by the MCU in transmit mode.
- A CRC is computed and checked over fields 1 through 3 and transmitted at the end of the packet without any involvement from the MCU. The CRC engine have four different 16 bit CRC polynomials to chose from; IEC-16 Bachieva-16, CRC-16 and CCIT-16. Baicheva-16, $x^{16} + x^{15} + x^{12} + x^7 + x^6 + x^4 + x^3 + x^0$, was chosen as it has the best Hamming distance for packets of lengths up to 135 bits [21].

4.3 Development tools

For code development, Keil μ Vision 4 integrated development environment is used. It features the C51 compiler together with the LX51 linker and A51 assembler [22].

For Si4463 settings, Silicon Laboratories wireless development suite is used to auto-generate a source file containing all API property settings [23].

4.4 Software

The software is divided into five functional blocks:

- PS/2 mouse interfacing on the master node
- Radio interfacing on master node
- Radio interfacing on slave node
- Main program on master node receiving packets from the PS/2 mouse and sending them to the RFIC
- Main program on the slave node which receives packets from the RFIC and handles the LED array accordingly

Each block will be described in detail in the following sections. As the radio interface works similarly for master and slave, it will be presented under a combined section.

4.4.1 PS/2 implementation

As the MCU does not include a PS/2 controller, drivers for initialization and streaming of the mouse have been constructed.

Initialization

During the initialization of the PS/2 mouse, the MCU must both write commands and read responses, while only read is needed during the following main program execution.

The proceedings of initializing the PS/2 mouse is as follows:

| | | |
|--------------|------------------------|----------------------------------|
| Write | 0xFF | // reset command |
| Read | until response is 0xAA | // device self check test passed |
| Read | until response is 0x00 | // device ID |
| Write | 0xF3 0x3C | // set sample rate 60 Hz |
| Read | 0xFA | // acknowledge |
| Write | 0xE8 0x00 | // set resolution 1 mm/count |
| Read | 0xFA | // acknowledge |
| Write | 0xF4 | // enable packet streaming |
| Read | 0xFA | // acknowledge |

Streaming mode

During normal operation the PS/2 mouse streams data to the MCU. That is, every time the mouse is moved it sends a three byte movement packet. As opposed to initialization mode, the MCU does not know exactly when to expect data in streaming mode. Therefore, the falling edge of CLK is designed to trigger an interrupt in the MCU. The interrupt service routine (ISR) is designed as a state machine with four states, see figure 4.5. The state machine goes through all states during reception of a single byte, checking for errors when three bytes (a full packet) are received. Each state does the following:

- idle** Initializes bit counter, buffer, and parity check bit. Reads the start bit on the PS/2 data pin.
- rx_data** Shifts in eight bits from the PS/2 data pin into a buffer and updates the parity check.
- parity** Reads the parity bit from the PS/2 data pin and updates the parity check.
- stop** Reads the stop bit on the PS/2 data pin. Stores the buffered data into the PS/2 FIFO. Every time a byte has wrong parity or an invalid stop bit, an error flag is set. The flag is only cleared after the reception of an entire three byte packet. Even if only one byte is invalid, the whole packet is considered invalid. Valid packets are stored by increasing the write pointer in the FIFO. For invalid packets, the pointer is not increased and the packet will be overwritten by the next received packet.

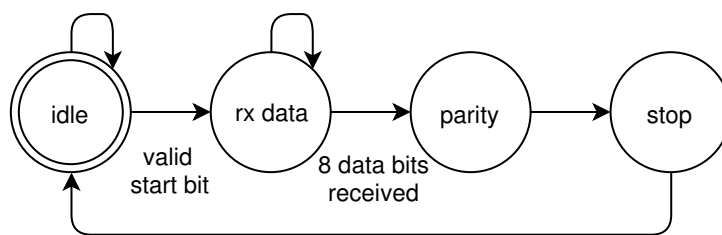


Figure 4.5: State machine for the PS/2 data reception in streaming mode. Each transition is triggered by an interrupt on the PS/2 clock pin.

Packets are stored in a 30×3 byte circular FIFO, where the first index holds the packet number and the last index is the packet-byte number. For instance, the x-position of the fifth received packet is stored at `FIFO[4][1]` using indices starting from 0. Two pointers keep track of which elements in the FIFO are filled. The write pointer points at the slot where the PS/2 ISR shall store the next received packet and the read pointer points to the next packet to be read by the main program. Figure 4.6 shows an example with a FIFO containing up to eight packets.

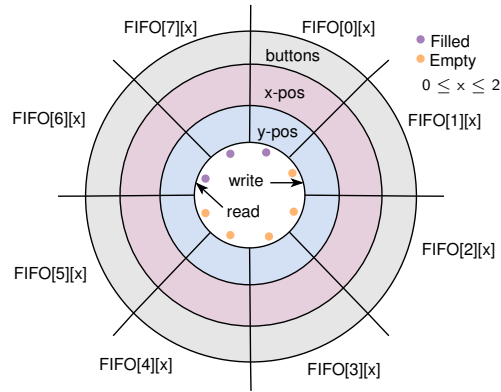


Figure 4.6: Example of a circular 8×3 byte circular FIFO. The write and read pointers point to complete mouse packets containing three bytes. The distance between the write and read pointer is the length of the packet queue. In this case, three packets have been received from the mouse but have not yet been read by the main program.

4.4.2 RFIC interfacing

Using Silicon Laboratories' drivers for configuring the radio API, functions to be used by the main programs on master/slave for radio interfacing was designed:

radio_init

- Set RF specific parameters such as modulation, output power, and data rates.
- Configure packet handler settings.
- Configure radio GPIO pin 1 as a *ready to receive new command* signal.
- Allow radio to issue interrupts on NIRQ whenever it is done transmitting or receiving a packet.

radio_start_TX

- Read and clear radio interrupts.
- Command radio to reset it's Tx FIFO.
- Write the specified number of bytes to the radio's Tx FIFO.
- Command the radio to start transmitting the data in the FIFO using the packet handler settings. Radio shall return to RX_TUNE state upon completion of transmission.

radio_start_RX

- Read and clear radio interrupts.
- Command radio to reset it's Rx FIFO.

- Command the radio to start receiving data using packet handler settings. Radio shall return to TX_TUNE if a valid packet is received, otherwise it shall re-enter RX_STATE.

radio_getPacket

- Check the length of radio Rx FIFO.
- Read an equal amount of bytes from the radio Rx FIFO.

4.4.3 Main program on master

The master's main program handles the PS/2 mouse interface, the Si4463 interface, LCD interface and computes the packet error rate (PER) of the radio link. As the PS/2 mouse controls whenever the movement packets are generated, it determines the timing of the entire system. Each data packet should be forwarded to the wireless link as soon as possible to minimize the delays. A state machine is used to step through the program, see figure 4.7. Following the figure is a detailed description of the actions of each state. The PS/2 mouse is not included in the state machine as it is handled in the background of the program by its ISR described earlier in this chapter. To secure the timings of all components the state transitions are controlled by these six event flags which are set by ISRs:

| | |
|---|--|
| fTx Set by the external interrupt issued by Si4463 when it is done transmitting a packet. | fPS2 Set by the PS/2 ISR when a valid three byte movement packet has been received. |
| fRx Set by the external interrupt issued by Si4463 when it is done receiving a valid packet. | fLCD Set at even time intervals by a timer overflow. |
| fTO Set when the ACK time out timer overflows. | fPER Set at even time intervals by a timer overflow. |

As the timing of the PS/2 mouse is not controlled by the MCU itself, it is the most timing critical and has the highest priority interrupt.

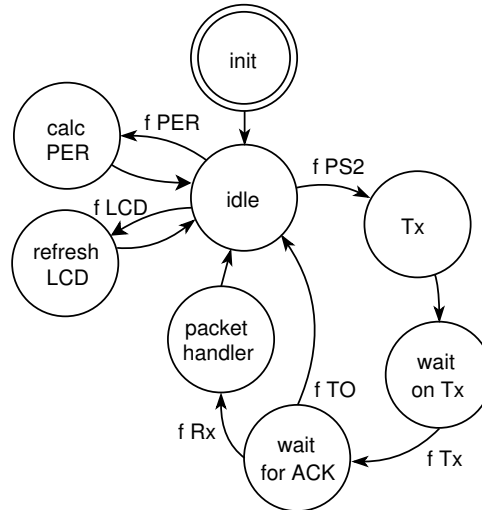


Figure 4.7: State machine of the master’s main program. Conditional state transitions are triggered by event flags. If the flags are unset, the program stays in the current state unless there exist a unconditional transition. Upon a hardware reset, the system returns to the init state.

| | |
|-----------------------|--|
| init | Initialize the MCU’s system clock, GPIO pin configuration and enable the two SPI controllers. Initialize the Si4463 through the <code>radio_init</code> function. Initialize LCD and PS/2 mouse. Clear the event flags, mouse FIFO, and sequence number counter. Start LCD/PER refresh timers. |
| idle | Waits for a mouse packet or timeouts of LCD/PER refresh timers. If more than one of the event flags is set, the order of priority is fPS2, fPER, fLCD. |
| Tx | Clear fPS2. Generate packet header containing the slave and master IDs, packet length, DAT frame type and sequence number. Increase the sequence number for the next packet. Retrieve the oldest packet from the mouse FIFO and decrease the mouse FIFO’s packet counter. Update the mouse FIFO’s read pointer. Change the length property in the Si4463 packet handler. Transmit the radio packet using the <code>radio_startTX</code> function. Increase the counter for sent packets. |
| wait on Tx | Waits for Si4463 to finish transmission. When complete, fTx is cleared and Si4463 is set in receive mode using <code>radio_startRX</code> . Start ACK timeout timer. |
| wait for ACK | Waits for the fRx or fTO flags. If fRx is set, the ACK timeout timer is stopped. Else, if fTO is set, the counter for lost packets is increased. |
| packet handler | Retrieve the radio packet using <code>radio_getPacket</code> . Check if the packet type field contains ACK (0b10) and the sequence number is correct. If not, increase the counter for lost packets. Clear fRx. |
| calc PER | Calculate the PER as the ratio between lost packets and sent packets. Clear fPER. |

refresh LCD Update the display with current values of sent packets, lost packet, PER and data rate. Clear fLCD.

4.4.4 Main program on slave

The slave's main program handles reception of DAT packets and transmission of ACK packets through the Si4463 interface. Additionally, it controls the LED array. Similarly to the main program of the master, the program implements a state machine, see figure 4.8. Only two event flags are used, fTx and fRx which work equally as in the master's main program. Following the figure is detailed description of the actions of each state.

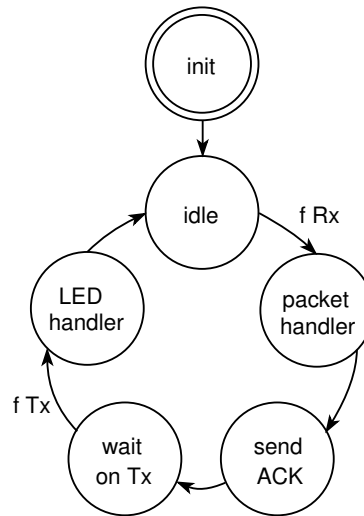


Figure 4.8: State machine of the slave's main program. Conditional state transitions are triggered by event flags. If the flags are unset, the program stays in the current state unless there is exist a unconditional transition. Upon a hardware reset, the system returns to the init state.

| | |
|-----------------------|---|
| init | Initialize the MCU's system clock, GPIO pin configuration and enable one SPI controller for Si4463 interfacing. Initialize the Si4463 through the radio_init function. Clear the event flags, turn off all LEDs, set the x-position to 0. Command radio to start receiving using radio_startRX. |
| idle | Wait for Si4463 to receive a valid packet. |
| packet handler | Retrieve packet from Si4463 using radio_getPacket. Check if the frame type bits contains DAT frame type (0b00). If yes, the three bytes of mouse data are extracted from the received packet. Extract the sequence number bits. |
| send ACK | Generate ACK packet header containing the master and slave IDs, packet length, ACK frame type, and previously retrieved sequence number. Change the length property in the Si4463 packet handler. Transmit the ACK packet using radio_startTX. |

- wait on Tx** Wait for Si4463 to finish transmission of the ACK packet. When complete, configure the length property of the Si4463 packet handler to receive DAT packets. Start receiving using `radio_startRX`.
- LED handler** Check the overflow bits of the mouse packet. If unset, update the current x-position with the counter in the mousepacket. If the new position is out of bounds, set it to min or max. Determine how many LEDs should be illuminated. Write a 1 to the output pins connected to the corresponding LEDs.

5 Results

5.1 Verification of RF parameters

A spectrum analyzer was connected to the antenna's RP-SMA connection to verify the device's RF parameters. The output power and frequency offset was measured by setting the Si4463 into continuous transmission of an unmodulated carrier wave at 869.3 MHz. Figure 5.1 shows the output power to be 9.85 dBm. The peak power is measured at 869.2978 MHz, giving a frequency offset of -2.2 kHz which stays within the ± 12.5 kHz limit set by ETSI-EN 300 220.

Measurements on four different modulation schemes were conducted in order to verify that the 2GFSK used in the demo has the highest spectral efficiency. Figure 5.2 shows the results of putting the Si4463 into continuous transmission of a 40 kbps PN9 sequence. OOK provides the least spectral efficiency far away from the carrier whereas 2GFSK is the most efficient. At frequencies close to the carrier 4FSK has the highest damping. As both spectral efficiency and sensitivity is of interest for the demo, 2GFSK is confirmed to be the best choice.

A basic range test was performed by configuring the master to send a DAT packet upon a button press on the WBM. The slave responds with an ACK. The distance between the nodes was increased until the master no longer received ACKs on its data packets. The test was performed in line of sight at different places in Uppsala resulting in an approximate range of 450 m using the demo's radio configurations of 10 dBm, 40 kbps, 2GFSK, 20 kHz deviation.

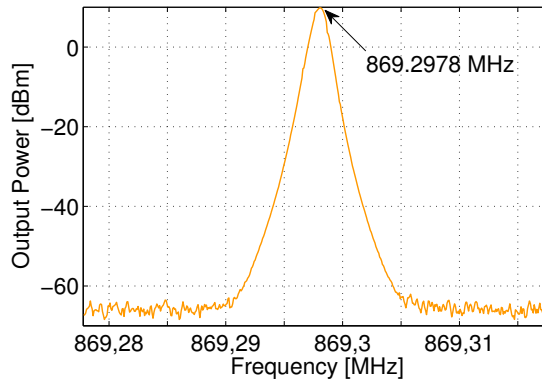


Figure 5.1: RF-spectra of an unmodulated carrier shows a center frequency of 869.2978 MHz and a maximum output power at 9.85 dBm.

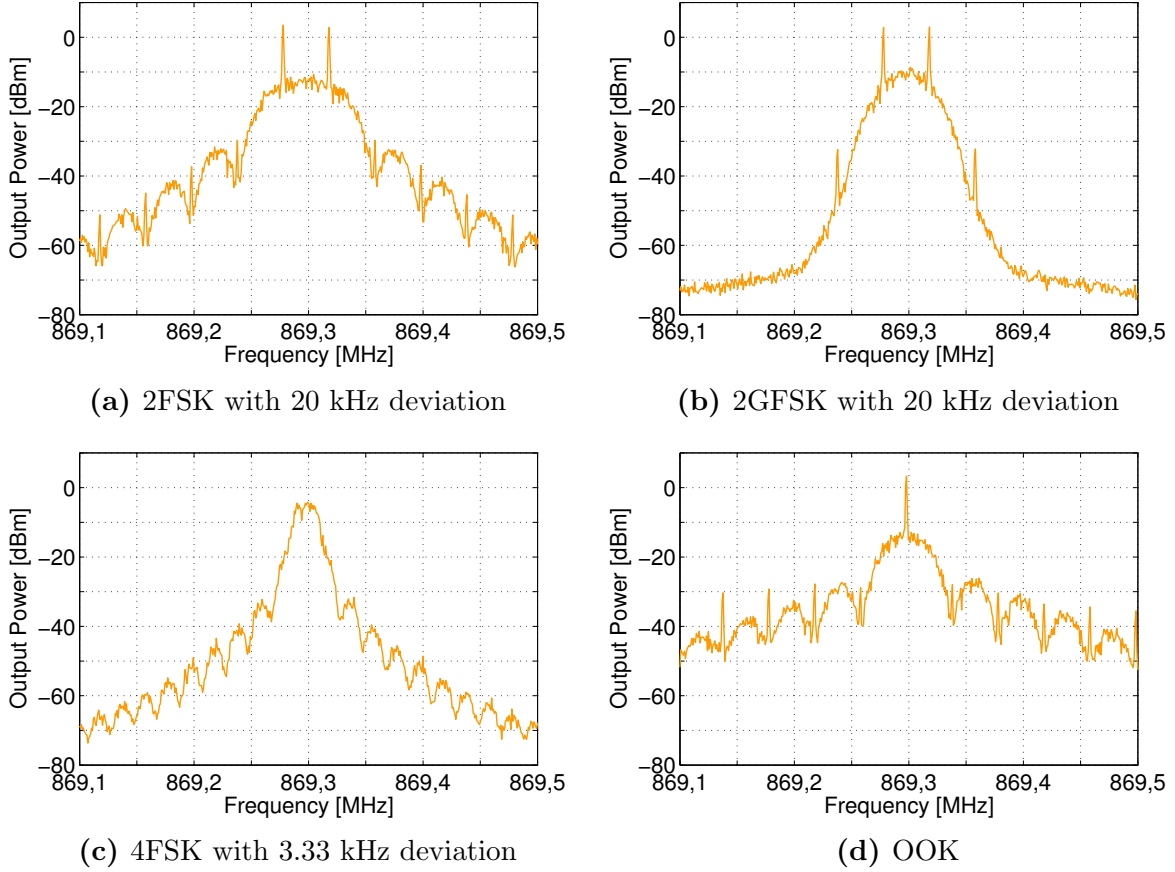


Figure 5.2: RF spectra of a continuous PN9 wave at 40 kbps data rate and 10 dBm output power for different modulation schemes.

5.2 PS2 mouse interface verification

The PS/2 mouse interface was verified by moving and clicking the mouse while monitoring the values in the mouse FIFO using the debugger. Additionally, the signal was viewed on an oscilloscope, see figure 5.3 for an example where the mouse is held still and the left button is pressed. Each of the three bytes can easily be identified. Considering the first byte, there is a start bit, followed by the data 0b10010000 and a high parity and stop bit. Note that the data is sent LSB and must be reversed before compared with the movement packets in section 2.4.2.

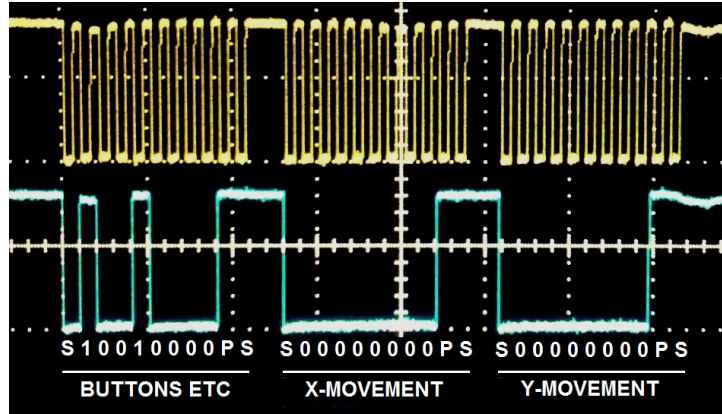


Figure 5.3: Clock (above) and data (below) of the PS/2 mouse while sending a three byte packet generated by a left click. Note that data bits are LSB.

5.3 Protocol verification

The matching functionality in the Si4463 was verified by configuring the slave MCU to illuminate an LED on the WMB whenever the Si4463 issued a mismatch interrupt on NIRQ. Thereafter the master was configured to send packets with correct or incorrect header bytes upon depression of two different push buttons on the WMB. The same approach was used for testing the variable length functionality.

5.4 Demo verification

The content of received DAT packets was verified by monitoring the data buffer through the slave's debugger while moving the mouse. The function of the LED array was verified visually while moving the mouse back and forth. No noticeable delays were experienced. Additionally, the latency was measured by configuring the master to pull a GPIO pin high upon start of transmission while the slave pulled a GPIO pin high upon reception of a packet. The signals were monitored on an oscilloscope showing a pulse width of 3.6 ms. Lastly, the entire communications link was verified by monitoring the LCD on the master while clicking the mouse, see figure 5.4. As one click generates two packets (press and release), the sent packets count was simply verified. The ACK timeout functionality and PER counter were checked by shutting off the slave's power while continuing clicking of the mouse.

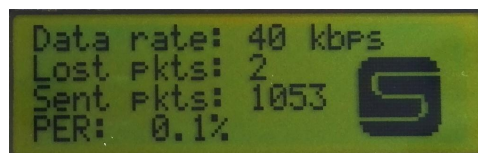


Figure 5.4: Example of master output during normal program execution.

6 Discussion

6.1 Goals met

- A list of over one hundred RFICs has been put together. The RFICs can be sorted based on sensitivity, output power, operational frequency, modulation type, cost, supported data rates, and ability to comply with different regulation standards.
- The ETSI EN 300 220 regulations have been studied and a document providing guidance during the design process has been produced.
- A demo for real time remote control in the 868 MHz frequency band has been successfully constructed.

6.2 Future work

- As of the current implementation, RF communication only happens when the mouse is moved. To increase robustness, the RF communication protocol should be extended to include a function for the master to ping the slave at even time intervals. Doing so, the quality of the communications link is evaluated at all times.
- To save power, the nodes should go into sleep mode between pings. The master would then wake up upon reception of mouse packets and/or at even time intervals to send pings. Similarly, the slave would wake up on even time intervals to receive pings or mouse data.
- The current one channel system should be extended to operate on several channels and a frequency hopping scheme should be implemented to make sure the demo complies with the regulatory standards requirements on duty cycle limits.
- A function for dynamically altering the data rate based on the PER was originally considered. Both nodes should have look-up tables containing Si4463 API properties for changing the data rate to some set values. A new type of radio packet, *command* containing the new data rate should be sent from the master to the slave whenever the PER got too high. Both nodes would then change data rates according to the look up table. As time ran out the function was never fully implemented in this thesis.
- In the current hardware setup the LED array is simply mounted on a breadboard. In order to make the demo more portable and less fragile a PCB should be designed with the LED's soldered onto it. Also, the LEDs' current size is too small to see at longer distances. As the partial purpose of the demo is to show real time functionality at long range a more visible alternative to the LEDs should be considered.
- In order to give a better evaluation of the current channel quality, the PER should be expressed as a moving average.

References

- [1] Silicon Laboratories, “Key Priorities for sub-GHz Wireless Deployment.” <http://www.silabs.com/Support%20Documents/TechnicalDocs/Key-Priorities-for-Sub-GHz-Wireless-Deployments.pdf>. [Accessed May 2016].
- [2] M. B. Steer, *Microwave and RF Design - A Systems Approach*. SciTech Publishing, 2010.
- [3] U. Dalal, *Wireless Communication and Networks*. Oxford University Press, 2015.
- [4] Texas Instruments, *DN509 - Data Whitening and Random TX Mode*, 2010.
- [5] C. G. Christodoulou and P. F. Wahid, *Fundamentals of Antennas - Concepts and Applications*. SPIE, 2001.
- [6] R. A. Witte, *Spectrum and Network Measurements (2nd Edition)*. Institution of Engineering and Technology, 2014.
- [7] The Swedish Post and Telecom Authority, “Auctions.” <http://www.pts.se/en-GB/Industry/Radio/Autctions/>. [Accessed May 2016].
- [8] A. Goldsmith, *Wireless Communications*. Cambridge University Press, 2005.
- [9] Texas Instruments, “ISM-Band and Short Range Device Regulatory Compliance.” <http://www.ti.com/lit/an/swra048/swra048.pdf>. [Accessed May 2016].
- [10] European Telecommunications Standards Institute, *ETSI EN 300 220-1*, 2012. v2.4.1.
- [11] Dag Grini, Texas Instruments, “RF basics, RF for non-RF engineers.” <http://www.ti.com/lit/ml/slap127/slap127.pdf>. [Accessed May 2016].
- [12] Microchip Technologies, *AN730 - CRC Generating and Checking*, 2000. Rev 0.6.
- [13] S. R. Bullock, *Transceiver and System Design for Digital Communications (4th Edition)*. Institution of Engineering and Technology, 2014.
- [14] S. Mukherjee, *Architecture Design for Soft Errors*. Elsevier, 2008.
- [15] V. A. Pedroni, *Circuit Design and Simulation with VHDL (2nd Edition)*. MIT Press, 2010.
- [16] OSDev, “Mouse input.” http://wiki.osdev.org/Mouse_Input, 2014. [Accessed March 2016].
- [17] Silicon Laboratories, *Si4463/63/60-C - High-Performance, Low Current Transceiver*, 2014. Rev. 1.0.
- [18] Silicon Laboratories, “Si4461 868 MHz Wireless Development kit.” <https://www.silabs.com/products/wireless/EZRadioPR0/Pages/Si4461-868-PDK.aspx>. [Accessed April 2016].

- [19] Silicon Laboratories, *AN633 - Programming Guide for EZRadioPRO Si4x6x Devices*, 2014. Rev 0.6.
- [20] IEEE Computer Society, *IEEE Std 802.15.4g-2012*, 2012.
- [21] P. Koopman and T. Chakravarty, “Cyclic Redundancy Checks (CRC) Polynomial Selection For Embedded Networks.” https://users.ece.cmu.edu/~koopman/roses/dsn04/koopman04_crc_poly_embedded.pdf. [Accessed May 2016].
- [22] ARMKeil, “uVision IDE.” <http://www2.keil.com/mdk5/uvision/>. [Accessed May 2016].
- [23] Silicon Laboratories, “Wireless development suite.” <https://www.silabs.com/products/wireless/EZRadio/Pages/WirelessDevelopmentSuite.aspx>. [Accessed April 2016].