Bluetooth Communication for Remote Controlling Purpose

Examensarbete utfört i Elektroniksystem
av
Johan Florén
Daniel Lindberg
LiTH-ISY-EX-ET-0287-2004
Linköping 2004
Bluetooth Communication for Remote Controlling Purpose

Examensarbete utfört i Elektroniksystem
vid Linköpings tekniska högskola
av
Johan Florén och Daniel Lindberg
LiTH-ISY-EX-ET-0287-2004

Handledare: Jonas Carlsson och Mattias Olsson
Examinator: Kent Palmkvist
Linköping 2004-06-02
Bluetooth-kommunikation för fjärrstyrningsändamål
Bluetooth Communication For Remote Controlling Purpose

Author
Johan Florén Daniel Lindberg

Abstract
The concept of remote controlling electrical devices is not new; people of today are used to being able to control their household electronic items without making an effort to walk up to the device.

In this thesis the aim is to use Bluetooth radio signals instead of the traditional infrared light to achieve the remote control functions. As a test application a Sony Compact Disc player is modified in order to be radio controlled with a standard Ericsson T39 mobile phone used as a remote. Since the test application is rather simple without any special requirements the development of a specific radio communication system has not been considered. Instead available pre-manufactured devices were examined. A Bluetooth evaluation kit was purchased and the test project consisted of the goal of linking the evaluation board to the CD player with an AVR microcontroller and getting the whole system to communicate with the mobile phone.

Keyword
Bluetooth, AVR, Ericsson T39, Bluegiga Wrap Thor, RS-232, remote controlling, mobile phone
Preface

This report is the culmination of our Bachelor of Science thesis in Electrical Engineering at Linköping University.

It is recommended reading for an audience curious about the theory of Bluetooth and AVR programming.

We would like to thank all the people in the Department of Electrical Engineering at Linköping University for their support. We would also like to thank Derek and Lotta Cook who have spent a lot of time reading this report, helping us to correct spelling and grammatical errors.

Linköping 2 June, 2004

Johan Florén and Daniel Lindberg
Abstract

The concept of remote controlling electrical devices is not new; people of today are used to being able to control their household electronic items without making an effort to walk up to the device.

In this thesis the aim is to use Bluetooth radio signals instead of the traditional infrared light to achieve the remote control functions. As a test application a Sony Compact Disc player is modified in order to be radio controlled with a standard Ericsson T39 mobile phone used as a remote. Since the test application is rather simple without any special requirements the development of a specific radio communication system has not been considered. Instead available pre-manufactured devices were examined. A Bluetooth evaluation kit was purchased and the test project consisted of the goal of linking the evaluation board to the CD player with an AVR microcontroller and getting the whole system to communicate with the mobile phone.
Sammanfattning

Fjärrstyrning som koncept är inte ett nytt påfund, dagens människor är vana vid att kunna fjärrmanövrera sina hemelektronikprodukter.

Contents

1 INTRODUCTION ............................................................................................................ 1
   1.1 BACKGROUND OF WIRELESS REMOTE CONTROLLING .................................... 1
   1.2 PROBLEM DESCRIPTION .................................................................................. 1
   1.3 DEFINED REQUIREMENTS AND LIMITATIONS ............................................ 1

2 PREPARATIONS ......................................................................................................... 3
   2.1 FINDING APPROPRIATE HARDWARE .......................................................... 3
       2.1.1 Bluetooth device ............................................................................... 3
       2.1.2 Microcontroller ............................................................................... 4
   2.2 THE COMPLETE SYSTEM ............................................................................ 4

3 BLUETOOTH ............................................................................................................. 5
   3.1 INTRODUCTION .............................................................................................. 5
       3.1.1 History ............................................................................................... 5
       3.1.2 The SIG ............................................................................................. 5
       3.1.3 Harald Blåtand ............................................................................... 6
       3.1.4 Network Topology ............................................................................ 6
           3.1.4.1 Master and slaves ................................................................. 6
   3.2 BLUETOOTH PROTOCOL ARCHITECTURE ............................................. 7
       3.2.1 Radio .................................................................................................. 7
       3.2.2 Baseband .......................................................................................... 8
           3.2.2.1 Physical links ........................................................................... 8
           3.2.2.2 Logical Channels ................................................................. 9
       3.2.3 Link Manager Protocol .................................................................... 9
       3.2.4 Host Controller Interface .................................................................. 9
           3.2.4.1 HCI Firmware ......................................................................... 10
           3.2.4.2 HCI Driver ............................................................................... 10
           3.2.4.3 Host Controller Transport Layer ............................................ 10
       3.2.5 L2CAP ............................................................................................... 10
       3.2.6 Service Discovery Protocol ............................................................... 11
       3.2.7 RFCOMM .......................................................................................... 11
   3.3 BLUETOOTH PROFILES ............................................................................ 11
       3.3.1 Generic Access Profile ................................................................. 12
       3.3.2 Service Discovery Application Profile ...................................... 12
       3.3.3 Serial Port Profile ............................................................................ 12

4 THE HARDWARE .................................................................................................. 13
   4.1 ERICSSON MOBILE PHONE .................................................................... 13
       4.1.1 Basic Ericsson AT commands .................................................... 13
   4.2 WRAP THOR BLUETOOTH MODULE ................................................... 15
       4.2.1 The ASCII interface ..................................................................... 16
VI Bluetooth communication for remote controlling purpose

4.3 ATMEL AVR MICROCONTROLLER ......................................................... 17
  4.3.1 Programming techniques .......................................................... 17
    4.3.1.1 Assembler .............................................................................. 18
    4.3.1.2 BASIC ................................................................................... 18
    4.3.1.3 C ......................................................................................... 18
  4.4 SONY COMPACT DISC PLAYER ....................................................... 18

5 PRACTICAL EXPERIMENT ................................................................... 19
  5.1 PROGRAMMING THE MICROCONTROLLER ..................................... 19
    5.1.1 Finding out the phone commands ........................................... 20
    5.1.2 Testing the UART ................................................................. 21
  5.2 PREPARING THE CD PLAYER .......................................................... 21
  5.3 CONNECTING THE WRAP THOR AND FINISHING THE CODE ......... 23

6 DISCUSSION AND RESULTS .............................................................. 25
  6.1 RESULTS ..................................................................................... 25
  6.2 FURTHER IMPROVEMENTS ............................................................ 25

7 TERMS & ABBREVIATIONS ............................................................... 27

8 LIST OF REFERENCES ...................................................................... 29
1 Introduction

The aim of this thesis is to look into, and experiment with, Bluetooth\textsuperscript{1} as a wireless remote controlling channel for consumer electronics.

1.1 Background of wireless remote controlling

The most commonly used method of achieving remote control functions is to use infrared light to transmit the control signals through the air. This method has a lot of advantages such as low power consumption and easy implementation but there is one major drawback, the device being controlled must be in close visible proximity to the user. Another way of transmitting the control signals is through the use of radio signals.

In this thesis Bluetooth is examined as a means of transmitting the control signals. Bluetooth is a predefined radio communication protocol constantly growing in popularity. Normally you would require a license for transmitting radio signals through the air, but since Bluetooth operates in the unlicensed industrial, scientific and medical (ISM) band this is not necessary. This makes Bluetooth suitable for this project.

1.2 Problem description

The goal stated for this project is to make an Ericsson mobile phone act as a remote control for a Sony CD player. The hardware needed to accomplish this is, apart from the CD player and the phone, some kind of device that can communicate with the phone via Bluetooth. A link between this Bluetooth device and the CD player is also needed.

1.3 Defined requirements and limitations

The user should be able to easily link the CD player and the mobile phone via Bluetooth. When connected, the following functions should be accessible from the connected phone:

- Play
- Pause
- Stop
- Next
- Previous

These functions should be available as long as the mobile phone stays within range of the CD player.

\textsuperscript{1} Bluetooth is a registered trademark of Bluetooth SIG, Inc.
2 Preparations
The starting point of the project is the Ericsson T39 mobile phone. Due to the fact that one cannot alter the phone’s hardware or software, the rest of the system must be constructed to work with the phone as it is.

The T39 has a built in capability to handle Bluetooth accessories. One example is Sony Ericsson’s own remote controlled car, the Bluetooth CAR-100\(^2\). All communication with this kind of accessory is done over a Bluetooth serial link, later discussed in the Theory chapter.

2.1 Finding appropriate hardware
The next step was to determine what other hardware would be needed to complete the task. In order to access the mobile phone’s Bluetooth communication channel some kind of Bluetooth device is required. Such a unit is not capable of direct communication with the CD player; a microcontroller is needed to facilitate the communication.

2.1.1 Bluetooth device
The following demands have to be met by the Bluetooth device:

- Device must be able to connect to the phone via a Bluetooth serial link.
- Device should have a relatively small form-factor to enable integration into the CD player.

After searching the market, two devices with satisfying specifications were found.

- Wrap Thor from Bluegiga.
- F2M03C2 from free2move.

These modules have a lot of properties in common. They both support the serial connection and have the approximate size of a regular stamp. Both are also affordable for this project (500 to 1000 SEK per unit). The choice finally fell on the Wrap Thor, mainly because Bluegiga offered an evaluation kit to their module thus saving some construction time. Internet addresses to their respective datasheets are available on the reference page at the end of this report [12][13].

The Wrap Thor module, after a serial link has been established, acts as a transparent link to whatever it is connected to.

\(^2\) Visit www.sonyericsson.com for more info about CAR-100.
This evaluation kit has a RS-232 serial connector mounted directly on the board and the opportunity to control it by ASCII commands.

### 2.1.2 Microcontroller

To make the system complete some kind of microcontroller is needed to communicate with the phone through the Bluetooth module and upon interaction give the CD player the proper signals. To fit the system the microcontroller should be able to handle serial communication and be equipped with enough I/O pins to manage to control the CD player. One output pin per implemented function is required. Five functions, Play, Pause, Stop, Next and Previous, necessitate five output pins.

Today there are many manufacturers of microcontrollers, although two brands are more commonly used when it comes to small scale manufacturing and experimental work - Atmel and Microchip. They have gained great popularity amongst hobbyists and therefore a lot of information and program code is freely available.

For this project an Atmel AVR AT90S8515 microcontroller was used, with its STK500 development board. This choice of microcontroller was convenient because of the hardware already being available on site.

This microcontroller met all the demands regarding serial communication capabilities and had enough I/O pins.

### 2.2 The complete system

Knowing which components are supposed to go in the final system, a figure of required connections can be drawn (see figure 2.1).

![Figure 2.1 Description of the connection between CD/AVR/Wrap Thor/T39.](image)

Starting from the right, the phone connects via Bluetooth to the Wrap Thor module. The Bluetooth module is connected to the AVR microcontroller with a standard RS-232 serial cable. The transfer of the control signals to the CD player is accomplished by connecting the player to one of the microcontroller’s available I/O pins. However this connection is not elementary and requires some extra preparations later described in the Practical experiment part of the report.
3 Bluetooth

This section overviews the history, organisation and technology of Bluetooth. The information has mainly been gathered from books [1][2] and the official Bluetooth websites [7][8]. In the technology chapters the theory (based on Bluetooth specification 1.1 [3][4]) is simplified, bringing up only those areas relevant to this thesis.

3.1 Introduction

The concept behind Bluetooth is to provide a universal short-range wireless capability. There are already a couple of ways to avoid using wires. One is to use beams of infrared light to carry information. This technique is used in the remote control systems of most electronic household items, and with the standard Infrared Data Association (IrDA) it’s used to connect computers with peripheral devices. IrDA has a faster maximum data rate than Bluetooth - 4 Mbps compared to 721 kbps - but it also has some major drawbacks. It is limited to point-to-point connections and above all it requires a clear line of sight.

3.1.1 History

In 1994 Ericsson Mobile Communications decided to investigate the possibility of making a laptop computer connect wirelessly with a mobile phone. The idea was to make a low-power, low-cost radio, small enough to fit into mobile phones and their accessories.

The engineering work began a year later and soon the true potential of the technology became apparent. Instead of just replacing cables between laptops and mobile phones Bluetooth showed possibilities of becoming a universal wireless standard for a wide range of devices.

3.1.2 The SIG

The Bluetooth Special Interest Group (SIG) was formed in 1998 by Ericsson together with IBM, Intel, Nokia and Toshiba. Today the number of promoter companies has increased to also include Agere, Microsoft and Motorola along with thousands of Associate and Adopter member companies.

The assignment was to develop the concept towards an open global technology standard, thus preventing Bluetooth from becoming the property of a single company. In July 1999 this resulted in the release the first Bluetooth specification (1.0). The specification documentation contains the information necessary to ensure full operability between
different devices from various manufacturers. In December 2000 the SIG published specification 1.0b and the current version 1.1 was released in February 2001.

3.1.3 Harald Blåtand

The name Bluetooth comes from King Harald Gormsson of Denmark, who lived in the tenth century. Harald, sometimes called Harald Blåtand (eng. Bluetooth), is credited for bringing Christianity to Scandinavia and uniting Denmark and Norway during his regency. In the Danish town of Jelling, Harald Blåtand erected an enormous rune stone which still stands in its original position. The Bluetooth Logotype is derived from the runes of his initials. [11]

3.1.4 Network Topology

When Bluetooth devices come within range of each other they can set up ad hoc point-to-point or point-to-multipoint connections. Two or more Bluetooth devices when connected form a piconet. Several piconets can be linked together in ad hoc scatternets.

3.1.4.1 Master and slaves

To control the traffic in a piconet, one of the Bluetooth devices becomes the master and the rest serves as slaves. Up to seven slaves can actively communicate with one master in Bluetooth specification 1.1. Scatternets are formed if the master of a piconet becomes a slave of another piconet or if a slave serves more than one master. Up to ten piconets within range can form a scatternet. Figure 3.1 illustrates the different kinds of relationship between master and slaves.

![Figure 3.1 Piconet with single slave(A), piconet with multiple slaves(B), scatternet(C).](image-url)
3.2 Bluetooth Protocol Architecture

Bluetooth is defined as a layered protocol architecture consisting of core protocols, cable replacement and telephony control protocols, and adopted protocols [10]. Figure 3.2 illustrates a simplified model of the protocol architecture only showing the parts relevant to this thesis.

![Figure 3.2 Simplified Bluetooth protocol architecture.](image)

In the following chapters the architecture will be described, starting with the core protocols and ending with the cable replacement protocol.

3.2.1 Radio

The lowest defined layer of the specification is the Bluetooth radio. It defines the requirements of the Bluetooth transceiver. Bluetooth operates in the unlicensed ISM band at 2.40 to 2.48 GHz. There are three classes of transmitters defined where the difference between them is the output power and thus also the transmission range. These classes are presented in table 3.1.

<table>
<thead>
<tr>
<th>Power Class</th>
<th>Max Output Power</th>
<th>Min Output Power</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100 mW (+20 dBm)</td>
<td>1 mW (0 dBm)</td>
<td>~100 m</td>
</tr>
<tr>
<td>2</td>
<td>2.5 mW (+4 dBm)</td>
<td>0.25 mW (-6 dBm)</td>
<td>~10 m</td>
</tr>
<tr>
<td>3</td>
<td>1 mW (0 dBm)</td>
<td>N/A</td>
<td>~1 m</td>
</tr>
</tbody>
</table>

*Table 3.1 Bluetooth Power Classes.*

The Bluetooth radio is using Gaussian Frequency Shift Keying (GFSK) for modulation, with a positive frequency deviation representing a binary one and a negative frequency deviation representing a binary zero.
3.2.2 Baseband
This is the physical layer and lies on top of the Bluetooth radio. It manages physical channels and links. To avoid interference from other signals in the 2.4 GHz band, such as WLAN, Bluetooth makes use of frequency hopping. At 1600 times per second it hops between 79 different frequencies (or channels) displaced by 1 MHz (2.402-2.480 GHz). In a few countries a smaller band is available so only 23 different channels are used, spaced 1 MHz apart.

Every Bluetooth device has a unique hardware address and a Bluetooth clock. An algorithm is used to derive the hopping frequency based on the master’s hardware address (sequence) and clock (phase). For full duplex communication a Time Division Duplex (TDD) scheme is used which alternates transmitting and receiving, the master starts its transmission in even-numbered time slots, and the slave starts its transmission in odd-numbered time slots.

3.2.2.1 Physical links
Two types of links can be established between a master and a slave:

**Synchronous Connection Oriented**
The Synchronous Connection Oriented (SCO) link is a point-to-point connection between the master and a single slave with a fixed bandwidth. The master maintains the SCO link by using reserved slots at regular intervals. This makes the SCO link ideal for connections where a guaranteed data rate is important but since SCO packets are never retransmitted the Bluetooth profile needs to have built-in tolerance for lost data. Examples of such profiles are Headset, Hands-Free and other profiles handling streaming data. The master can support up to three simultaneous SCO links.

**Asynchronous Connectionless**
The Asynchronous Connectionless (ACL) link is a point-to-multipoint connection between the master and all the slaves that are paired with the master. It uses the slots not reserved for SCO links to send packets of data. No bandwidth reservation is possible but data delivery can be guaranteed through error detection and retransmission. Data is sent in 1-slot, 3-slot or 5-slot packets. The maximum data rate is 721 kbps in forward direction which is achieved using asymmetric unprotected 5-slot packets. The master device can only handle one single ACL link at a time.
3.2.2.2 Logical Channels

Bluetooth defines five types of logical channels which can be used to carry different types of information. The LC (Link Controller) channel is used in low level link control. The LM (Link Manager) channel carries control info exchanged between the devices. The UA (User Asynchronous), UI (User Isochronous) and US (User Synchronous) channels carry user data.

3.2.3 Link Manager Protocol

The Link Manager carries out link setup, authentication, link configuration and other aspects of the radio link between a master and a slave. It discovers other remote Link Managers and communicates with them via the Link Manager Protocol (LMP).

3.2.4 Host Controller Interface

In order to make different hardware implementations compatible, Bluetooth devices use the Host Controller Interface (HCI) as an interface between the Bluetooth host (e.g. a PC) and the Baseband and Link Manager. The HCI is functionally broken up into three parts, where each one has a different roll to play. As presented in figure 3.3, these parts are the HCI firmware, the HCI driver and the channel between them.

*Figure 3.3 Illustration of the data transfer path between two hosts.*
3.2.4.1 HCI Firmware
The Host Controller is the actual Bluetooth hardware. This is where the HCI Firmware is located. The firmware implements the HCI commands for the Bluetooth hardware by accessing Baseband commands, Link Manager commands, hardware status registers, control registers, and event registers. The term Host Controller means the HCI-enabled Bluetooth device.

3.2.4.2 HCI Driver
The HCI Driver is located on the Host (e.g. software unit). The host receives notifications of HCI events when something occurs. It will then analyze the received event packet to determine what event occurred. The term Host means the HCI-enabled software unit.

3.2.4.3 Host Controller Transport Layer
Interactions between the HCI driver and firmware occur over the Host Controller Transport Layer which is a definition of several layers that may exist between the HCI driver and firmware. These layers should be able to transfer data without intimate knowledge of the data being transferred. USB, UART and RS-232 are the layers that have initially been defined for Bluetooth.

3.2.5 L2CAP
The Logical Link Control and Adaptation Protocol (L2CAP) is layered over the Baseband protocol and provides services to upper layer protocols with protocol multiplexing capability and packet segmentation and reassembly (SAR). The L2CAP specification is defined for ACL links only.

The L2CAP must support protocol multiplexing because of the separation of the higher level protocols. Since the Baseband protocol cannot distinguish between upper layer protocols the data packets have to go through the L2CAP.

The SAR functionality is absolutely necessary to support protocols using data packets larger than those supported by the Baseband. Large L2CAP packets are segmented into multiple smaller Baseband packets prior to their transmission over the air. In the same way, multiple received Baseband packets may be reassembled into a larger L2CAP packet.
3.2.6 Service Discovery Protocol

The Service Discovery Protocol (SDP) basically provides a tool for applications to discover information from other Bluetooth devices, which services are available and to determine the characteristics of those available services.

3.2.7 RFCOMM

Serial ports are one of the most common types of communication services. The RFCOMM is a cable replacement protocol included in the Bluetooth specification. It emulates a RS-232 serial port over the L2CAP protocol and replaces serial cables with minimum modification of the devices. The RFCOMM supports up to 60 open emulated ports between two devices, however the number of ports that can be used simultaneously in a device is implementation-specific.

3.3 Bluetooth Profiles

The profiles have been developed in order to describe how implementations of user models are to be accomplished. This decreases the risk of interoperability problems between different manufacturers’ products. A profile defines options that are mandatory, optional and conditional in each protocol for the profile to implement its features. Figure 3.4 shows all the profiles mentioned in the following theory and how they are dependent on each other.

Profiles contained in another profile illustrate dependency. For example, the Headset profile is dependent on Serial Port and Generic Access profiles.
3.3.1 Generic Access Profile

The most essential of the Bluetooth profiles is the Generic Access Profile (GAP). All the other profiles are based upon this and dependent on parts of it. Basically the GAP describes how the lower layers (LMP and Baseband) are used, along with some higher layers. The GAP defines general procedures for how to discover, pair and set up security between Bluetooth devices. This profile also specifies user interface parameters such as Bluetooth Device Name (User Friendly Name), Bluetooth PIN and Bluetooth Class of Device.

3.3.2 Service Discovery Application Profile

The Service Discovery Application Profile (SDAP) defines the needed features and procedures for an application in a Bluetooth device to be able to locate services in other Bluetooth devices. The device should also be able to retrieve any desired available information related to these services using the Service Discovery Protocol.

3.3.3 Serial Port Profile

The Serial Port Profile (SPP) defines the required protocol options and procedures that shall be used when setting up an emulated RS-232 serial cable connection using RFCOMM between two Bluetooth devices.

In the SPP profile only one connection at a time is dealt with, but it is possible for several simultaneous connections to exist since multiple executions of the profile should be able to run concurrently in the same device.

In this type of connection there are no fixed master and slave roles. Other profiles like Headset, Hands-Free and Fax profile are built upon the SPP.
4 The hardware

This chapter contains facts and specifications about the hardware used in this project.

4.1 Ericsson mobile phone

The mobile phone used in this experiment is the Ericsson T39 as shown in figure 4.1. This was one of the first commercial phones equipped with Bluetooth technology.

![Figure 4.1 The Ericsson T39 mobile phone.](image)

This phone can handle many of the Bluetooth profiles, although only the Serial Port Profile is used when handling accessories. When the Bluetooth serial connection is established with an accessory the next step is to add its control menu to the phone’s built-in accessories menu. The T39 communicates with the *AT Command Set* over the serial connection. AT is a contraction of attention, a command originally used to program modems from Hayes Microcomputer Products. The *AT Command Set* was later adopted by other modem manufacturers as a standard protocol to interface modems via ASCII commands. Ericsson’s version of the command set is extended with Ericsson specific commands. This allows personal computers and other devices not only to communicate with the phone’s built-in modem but also affect the phone’s settings and user interface. It was introduced with the phone R320 and its IR link and has later been adapted for use with Bluetooth communication.

4.1.1 Basic Ericsson AT commands

The following part explains the most important commands used when initiating an accessory. The syntax is fully described in the *R320 AT Command Online Reference* [5]. When the phone understands a command it answers with an ‘OK’. If it doesn’t understand a command or the command’s parameters are in a bad form the answer would be ‘ERROR’. 
AT*EAM

AT*EAM="accessory menu item name"

This command adds an accessory menu item with given name to the phone’s accessory menu. A new menu item overwrites any existing menu item for the accessory. If the accessory disconnects, the menu item is deleted. When the menu is accessed the phone will send the following command:

*EAAI

The command tells the accessory to take action.

ATE

ATE=0

As a default the phone echoes every character being sent to it, ATE=0 turns off this option.

AT+CSCS

AT+CSCS="UTF-8"

This command defines the character set to be used.

AT+CMER

AT+CMER=0,0,0,0,0

The CMER command tells the phone how to communicate with the accessory. One can for example define how the phone shall handle keypad and display events. When all the parameters are set to zero the phone is in its most passive mode and reports events only on request.

AT*EASM

AT*EASM="menutitle",1,1,3,"item1","item2","item3"

To send a multiple choice menu to the phone the command above is used. By the parameters one can change the number of entries and their labels, menu title and, by the second digit, the default choice. When the user picks an item the following answer is sent:

*EAMI: 0

The digit in the answer tells the accessory which menu item was picked.
**AT*EAID**

\[ \text{AT*EAID}=1,2,"\text{message text}\" \]

When the accessory needs to send the user a message, this command can be used. It is possible, with the parameters, to make the message into, for example, an input prompt but the digits ‘1,2’ define it as a plain message.

### 4.2 Wrap Thor Bluetooth module

The Wrap Thor is a configurable Bluetooth module optimized for embedded applications. It is based on the current Bluetooth specification 1.1. The Wrap Thor is a Class 1 device which means it is capable of transmitting and receiving data over a distance of up to 100 meters. The module needs a 3.3 volt power supply to operate; however the evaluation kit has a power regulator allowing the board to be fed with 5 volts.

When the evaluation kit is purchased the surface mounted module comes soldered to a circuit board containing power regulator, antenna and serial communication connectors (see figure 4.2). The actual module is the metal-shielded circuit in the middle of the board.

![Figure 4.2 The Wrap Thor Evaluation Kit.](image)

This module is designed to be fully usable and configurable via UART. Available connectors are RS-232 and USB. When sending commands to the module, such as ‘search for available Bluetooth devices’, one may use a RS-232 serial cable and a terminal program on a PC to access the built-in ASCII interface.
4.2.1 The ASCII interface

The Wrap Thor’s ASCII interface includes a few commands that handle the modules Bluetooth functions. Below is a short description of the three commands used in this project. The full description is available in the \textit{WRAP THOR ASCII Interface Manual} [6].

\textbf{Inquiry}

\begin{verbatim}
INQUIRY <time> NAME
\end{verbatim}

This command tells the Wrap Thor to search for other Bluetooth enabled devices in range of the module. The answer consists of the found devices’ hardware addresses, type of devices and their user friendly names. The syntax is:

\begin{verbatim}
INQUIRY_PARTIAL 00:07:80:80:00:1f 520204 Name1
INQUIRY_PARTIAL 00:02:34:12:03:12 320104 Name2
INQUIRY 2
INQUIRY 00:07:80:80:00:1f 520204
INQUIRY 00:02:34:12:03:12 320104
NAME 0
\end{verbatim}

The 12 digit strings contains the found devices’ hardware addresses. The next number represents the type of device, 520204 is a mobile phone and 320104 is a personal computer. The last of the entries represents the friendly name of the device. INQUIRY 2 means two devices have been found.

\textbf{Call}

\begin{verbatim}
CALL <Bluetooth hw address> 1101 rfcomm
\end{verbatim}

The CALL command is used when a connection is going to be established. Numbers 1101 tells the Wrap Thor to create a Serial Port Connection. The pairing is automatically done in the same step. In the case of a successful call the Wrap Thor answers with:

\begin{verbatim}
CONNECT 0 RFCOMM 0
\end{verbatim}

The digit following CONNECT represents the internal link ID of the connection.

\textbf{Set}

\begin{verbatim}
SET BT AUTH * <pin>
\end{verbatim}

This command changes the Wrap Thor’s PIN code via the ASCII interface. This PIN must be entered every time a device is connected by the Wrap Thor to complete the pairing process.
It is also possible to change the Wrap Thor’s friendly name, although this cannot be done in the ASCII mode. This, and many other settings, requires the use of the special configuration software included in the evaluation package.

4.3 Atmel AVR microcontroller

The AT90S8515 [14] is an 8-bit RISC microprocessor. RISC is short for Reduced Instruction Set Computing. The technique of using a few powerful instructions makes this type of processor efficient as every instruction uses only one or two clock cycles. With 8 KB of in-system programmable flash program memory and 544 bytes of internal SRAM this microcontroller is very versatile. It has 32 programmable I/O pins where two of them can be used as RxD and TxD and linked to a RS-232 connector on the development board. The Atmel AVR STK500 Starter Kit development board can be seen in figure 4.3.

![Figure 4.3 The Atmel AVR STK500 Starter Kit board.](image)

There are two RS-232 connectors present on the board, one for communicating with the microcontroller and one for programming. All of the processor’s pins are grouped into I/O ports that are easily available on the board.

The microcontroller is supporting clock speeds of up to 8 MHz. In this project it will operate at 3.69 MHz since this is the standard speed when mounted on the development board.

Internet addresses to the datasheet of the processor are available on the reference page.

4.3.1 Programming techniques

There are a great variety of techniques for writing the actual instruction code for this microcontroller. Development tools for using Assembler, BASIC and C are available.
4.3.1.1 Assembler

Assembler is the most commonly used language when programming microcontrollers. With assembler the instructions are written as they should appear in the finished program. This is a powerful language when writing small effective programs. When designing larger programs this low-level programming technique tends to become somewhat difficult to survey. Atmel themselves have developed a free software package called *AVR studio*. This program is very useful with its own Assembler editor, programmer and debugger.

4.3.1.2 BASIC

BASIC is a high-level language developed in 1964. It is easy to learn due to its simple syntax. For a long time it had a reputation of producing slow applications but the compilers have been improved over the years to make it a competitive programming language. The most commonly used compiler for BASIC and AVR is *BASCOM-AVR* from MCS Electronics.

4.3.1.3 C

The C programming language was originally developed in parallel with the UNIX operating system in the early 1970s. It is a powerful high-level language due to its close-to-hardware syntax. When compiling C-code for an AVR microcontroller the *AVR-GCC* software package may be used. AVR-GCC is released as free software. It is a compiler based on the *GNU Compiler Collection (GCC)* belonging to the GNU project. Because of its Public Domain nature there is a lot of third party source-code available with written functions for handling UART connections, displays etc.

4.4 Sony Compact Disc player

The CD-player used for this project is a Sony CDP-XE220. This is a standard stationary CD-player intended for home use. It has a traditional design and comes with the basic playback functions (see figure 4.4).

![Figure 4.4 The Sony Compact Disc player](image)

The Play, Pause and Stop functions are controlled with push-buttons while the Next and Previous functions are controlled with a jog wheel.
5 Practical experiment

This project began with ordering a Wrap Thor Bluetooth Evaluation Kit from Bluegiga Technologies. The shipment was supposed to take about three weeks and therefore the work started with other parts of the system.

The system can be apportioned as two major units, the Bluetooth transmitter/receiver and the microcontroller. The intercommunication is handled by a RS-232 serial bus. This gives the opportunity to develop each part separately by using a standard PC with a terminal program to emulate one side of the system while developing the other. The split system also makes it easier to find and correct possible errors.

5.1 Programming the microcontroller

To learn the techniques of programming the microcontroller a few test programs were written. Among these were a few to determine which programming language was the most suitable when using UART communication. It was discovered quite quickly that it would be rather difficult to write the program code directly in Assembler since the AVR needed to be able to handle long text strings. The BASIC language was also set aside due to the fact that the BASCOM-AVR software couldn’t be fully used without purchasing a license. This leaves the C language, so some research was made to find information regarding implementing UART on a AVR microcontroller.

One very useful library was found called *Interrupt controlled UART library*, written by Peter Fleury [9]. It uses a ring buffer (circular buffer) to receive and transmit data via UART. The package consists of two files, uart.c and uart.h. Here is a description of some of the library’s functions.

```c
void uart_init(unsigned char baudrate)

Initializes UART and sets baud rate.

unsigned int uart_getc(void)

Obtains received byte from ring buffer.

void uart_putchar(unsigned char data)

Puts byte to ring buffer for transmitting via UART.

void uart_putstr(const char *s)

Puts string to ring buffer for transmitting via UART.
```
void uart_puts_p(const char *s)

Puts string from program memory to ringbuffer for transmitting via UART.

These functions make serial communication much easier and therefore they were chosen to be included in the project.

5.1.1 Finding out the phone commands

Before starting to write the program code that would send the proper commands to the phone it was necessary to find out what to send.

First of all, a menu amongst the accessories on the phone had to be created. It was achieved with the AT*EAM command:

AT*EAM="Sony XE-220"

The menu is named after the CD-player. After this, the character set to be used must be defined, as must how the phone reacts on user input:

AT+CSCS="UTF-8"
AT+CMER=0,0,0,0,0

‘UTF-8’ is the character set to be used. AT+CMER tells the phone to stay passive until receiving a request from the connected accessory or until user interaction occurs. When a user accesses the new CD-player menu the phone sends the command *EAAI and tells the microcontroller to transmit the CD-player control menu. It looks like this:

AT*EASM="Sony XE-220",1,1,6, "Play", "Pause", "Stop", "Next", "Previous", "About"

The menu consists of six choices, one for each function to be implemented and one ‘About’ choice. When the user selects an item the phone will answer with *EAMI: # where the # describes which choice was made, e.g. digits 1-6. If a user presses ‘NO’ on the phone and thereby exits the menu, the answer will be *EAMI: 0. In the case of the answer being *EAMI: 6 the ‘About’ is sent with the following command:

AT*EAID=1,2,"Bluetooth Sony CD Control 2004"

Later, when knowing how the AVR’s UART worked, it was considered much easier to interpret the phone’s answers if its default ‘command line echoing’ function were turned off. This was accomplished with the command:

ATE=0

When knowing the detailed task of the microcontroller it was possible to write a program to test the UART.
5.1.2 Testing the UART
While waiting for the arrival of the Wrap Thor Bluetooth module a Bluetooth USB dongle was used with a PC to emulate the Wrap Thor’s function. This setup was possible thanks to a PC application called PassThru by Kevin Millican. PassThru enabled a transparent connection between one of the computer com-ports and the Bluetooth dongle. If it were possible to get the AVR to communicate with the mobile phone through the PC it would work when connected to the Wrap Thor, although some extra Bluetooth initiation commands would have to be added in the latter case.

A program for the AVR was written and tests began. This first test-code had a lot of similarities with the \textit{ctrlmode}-function included in the finished code. It also included the \textit{getstring}-function used to read and place characters into a string from the UART. The end of a received string is symbolized by a new-line-character by the phone.

As well as PassThru another PC application was found very useful to this project - \textit{Serial Monitor} from HHD software. This program is capable of spying on one of its host computer’s serial ports. The code debugging became much easier when using this application as, amongst other things, the exact answers from the mobile phone could be seen.

After some successful testing it was time to get the AVR to communicate with the CD-player.

5.2 Preparing the CD player
To be able to control the CD player from the AVR microcontroller some additional components were needed. The microcontroller’s output ports work with a voltage of 5 volts to symbolize a logical ‘one’ and 0 volts for a ‘zero’. Somehow these voltages had to be used to emulate the CD player buttons being pushed. The original control button configuration of the CD player can be seen in figure 5.1 below.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{figure5.1.png}
\caption{The Sony CD players original button configuration.}
\end{figure}

Sony is using a switch that is usually open to lower the potential of the control wire when pressed. To make the AVR microcontroller able to push the buttons, a regular NPN bipolar transistor was added in parallel with each button (see figure 5.2).
The base of the transistor is connected via a resistor to a unique output pin on the AVR microcontroller. The extra resistor is used to limit the base current of the NPN transistor.

This technique worked for the functions Play, Pause and Stop. The Next and Previous functions are controlled by a jog wheel on the CD player. Such a device is often called a digital potentiometer, or pulse switch. It has two output pins and one ground pin. In its normal state the first of the output pins is connected to ground by the switch and the other one has a floating potential. When you turn the knob on the switch the second output pin gets connected to ground and the first enters the floating state for a short period. Depending on in which direction the knob is turned the two events just mentioned occur in a different order. You can then by decoding the pulses determine the direction in which the knob was turned. Figure 5.3 shows how it is normally connected to the CD player.

To make the AVR microcontroller able to change track on the CD player the changes seen in figure 5.4 had to be made.

By this configuration the AVR can both break the normally grounded wire and ground the normally floating wire. In order to keep the CD player’s original controls working, a logical ‘one’ must normally be in place on the first AVR control wire to ensure that the topmost transistor is turned on.
The complete schematic describing how the CD player was connected to the AVR microcontroller is available in appendix A.

### 5.3 Connecting the Wrap Thor and finishing the code

Once the communication part of the code was finished and the CD player was successfully connected to the microcontroller, the functions of the Wrap Thor could be looked into. Just like with the AVR, it was possible to connect the evaluation kit to a PC by a RS-232 cable. To make the Wrap Thor compatible with the rest of the system, settings regarding the baud rate were adjusted using the included BlueSuite software. The module’s User Friendly Name was also changed with this software.

The Wrap Thor is powered by 5 volts, available on the STK500 board. The reset pins of the two boards were connected allowing them to be reset at the same time. One of the less important buttons on the CD player was later modified to control this reset.

Initially the idea was to let the mobile phone establish the connection. Tests were made to see how the Wrap Thor and the phone responded when either one of them tried to connect with the other. But soon problems occurred as it became clear that the mobile phone was unable to initiate and set up the serial link. So contrary to the first idea, the Wrap Thor had to establish the connection instead. Another problem that occurred was that the Wrap Thor echoed back everything sent to it in command mode. Unlike the T39 mobile phone there was no option for turning this off, so a special function, named echosend, was written for the AVR, only used during link establishment. This function sends one character of a string at a time and waits for each character to return before sending the next one. This was necessary since otherwise the ring buffer would have been clogged by the echoed characters preventing a correct interpretation of the answer.

The part of the code establishing the serial link could now be written. On reset the AVR first waits for the Wrap Thor to start up. This is done by the function startup which is a basic delay loop. This function also tells the Wrap Thor which Bluetooth PIN to use when pairing with other devices.

After this the main program enters the connect-function. Here the Wrap Thor first sends an inquiry to search for Bluetooth devices set as discoverable. The returning answer is then saved into a buffer. From this the Bluetooth hardware address of the first discovered mobile phone is read out and saved into a variable. Later on, the CALL command is used with the hardware address in the variable as a parameter. The Wrap Thor
makes the call whilst the AVR waits for the confirmation of a successful link establishment before moving on in the main program.

In the next step the microcontroller enters the `ctrlmode`-function where all the communication with the phone and the CD-player is carried out. The execution flow is shown in figure 5.5.

![Diagram showing program flow](image)

*Figure 5.5 Program flow.*

To show the function of the system a log was kept during a test where the system was started and play followed by stop was pressed on the phone. The log is available as appendix B.
6 Discussion and results

In this concluding chapter the results of this thesis will be presented, together with the problems encountered and possible further improvements.

6.1 Results

In the present configuration the user has to press a button on the CD-player to reset the system, not making it an entirely remote system after all. However, once established, the connection between the mobile phone and the CD-player will stay up as long as they are within range of each other. Since the Bluetooth unit implemented in this project is a power class 1 device, the range is restricted by the mobile phone. Today practically all Bluetooth enabled mobile phones use a class 2 device, so the working range should be about 10 meters.

![Figure 6.1 The control menu.](image)

When connected, the user is able to access the control menu containing the implemented functions, as seen in Figure 6.1. If the connection is lost the system must be reset to recreate the link.

When selecting between different Bluetooth devices for this project the main parameters considered were price and availability. But there are more things to be considered when making this choice. One is to make sure that there is extensive product documentation available. Another is the support options. In Wrap Thor’s case neither was sufficient.

6.2 Further improvements

There are a few things one may improve to make this system ready for implementation in a consumer product. First of all one may find a smaller
and more suitable microcontroller to control the system; the AT90S8515 features a lot of functions not relevant to this project. Secondly the system lacks error handling, for example it would be better if the system could reconnect should the connection be broken due to interference. One additional drawback is that the system only works with Ericsson compatible mobile phones. Due to the fact that there isn’t a set standard for how this type of Bluetooth accessories should behave, there is probably a lot more work to be done to determine its suitability as a generic accessory.

As mentioned earlier in chapter 5.3 the goal was not completely met regarding the connection setup procedure. The best solution would undoubtedly be to establish the connection from the mobile phone as intended from the beginning of this thesis. When connecting from the CD-player, a problem occurs where multiple phones are present and in discoverable mode. This could be solved by choosing between all discovered Bluetooth devices from a display connected to the microcontroller.

Another improvement would be to make the remote system more advanced. The user may want to be able to control more functions of the CD-player from the mobile phone. If the CD-player is customized to better fit the remote control system, it would also be possible to implement 2-way communication between the two devices, e.g. show current track number on the phone’s display.
### Terms & Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACL</td>
<td>Asynchronous Connectionless</td>
</tr>
<tr>
<td>ASCII</td>
<td>American Standard Code for Information Interchange</td>
</tr>
<tr>
<td>AVR</td>
<td>Advanced Virtual RISC</td>
</tr>
<tr>
<td>BASIC</td>
<td>Beginner's All-Purpose Symbolic Instruction Code</td>
</tr>
<tr>
<td>C</td>
<td>C Programming Language</td>
</tr>
<tr>
<td>CD</td>
<td>Compact Disc</td>
</tr>
<tr>
<td>GAP</td>
<td>Generic Access Profile</td>
</tr>
<tr>
<td>GFSK</td>
<td>Gaussian Frequency Shift Keying</td>
</tr>
<tr>
<td>GNU</td>
<td>Free operation system project</td>
</tr>
<tr>
<td>HCI</td>
<td>Host Controller Interface</td>
</tr>
<tr>
<td>I/O</td>
<td>Input/Output</td>
</tr>
<tr>
<td>IrDA</td>
<td>Infrared Data Association</td>
</tr>
<tr>
<td>ISM</td>
<td>Industrial Scientific Medical</td>
</tr>
<tr>
<td>L2CAP</td>
<td>Logical Link Control and Adaptation Protocol</td>
</tr>
<tr>
<td>LC</td>
<td>Link Controller</td>
</tr>
<tr>
<td>LM</td>
<td>Link Manager</td>
</tr>
<tr>
<td>LMP</td>
<td>Link Manager Protocol</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDU</td>
<td>Protocol Data Units</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number</td>
</tr>
<tr>
<td>RFCOMM</td>
<td>Radio Frequency Communication</td>
</tr>
<tr>
<td>RISC</td>
<td>Reduced Instruction Set Computing</td>
</tr>
<tr>
<td>RS-232</td>
<td>Recommended Standard 232</td>
</tr>
<tr>
<td>RxD</td>
<td>Receive Data</td>
</tr>
<tr>
<td>SAR</td>
<td>Segmentation And Reassembly</td>
</tr>
<tr>
<td>SCO</td>
<td>Synchronous Connection Oriented</td>
</tr>
<tr>
<td>SDAP</td>
<td>Service Discovery Application Profile</td>
</tr>
<tr>
<td>SDP</td>
<td>Service Discovery Protocol</td>
</tr>
<tr>
<td>SIG</td>
<td>Special Interest Group</td>
</tr>
<tr>
<td>SPP</td>
<td>Serial Port Profile</td>
</tr>
<tr>
<td>TDD</td>
<td>Time Division Duplex</td>
</tr>
<tr>
<td>TxD</td>
<td>Transmit Data</td>
</tr>
<tr>
<td>UA</td>
<td>User Asynchronous</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver-Transmitter</td>
</tr>
<tr>
<td>UI</td>
<td>User Isochronous</td>
</tr>
<tr>
<td>UNIX</td>
<td>Computer operating system (org. UNICS)</td>
</tr>
<tr>
<td>US</td>
<td>User Synchronous</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
</tbody>
</table>
8 List of references

Books

Specification documents
http://linuxbrit.co.uk/downloads/R320AT_R1A.pdf  
2004-05-03

URLs
2004-05-10
http://www.bluetooth.com  
2004-05-03
[8] Bluetooth SIG, *The Official Bluetooth Wireless Membership Site*  
http://www.bluetooth.org  
2004-05-03
[9] Peter Fleury, *Interrupt controlled UART library*  
http://homepage.sunrise.ch/mysunrise/pfleury/group__pfleury__uart.html  
2004-05-03
[10] Palowireless Wireless Resource Center, *Bluetooth Tutorial*  
http://www.palowireless.com/infotooth/tutorial.asp  
2004-05-20
http://www.ericsson.com/bluetooth/aboutbluet/  
2004-05-20
Datasheets

2004-05-12

http://www.free2move.se/pdf/F2M03C2.pdf
2004-05-12

[14] Atmel, *Datasheet for Atmel AVR AT90S8515 Microcontroller*
http://www.atmel.com/dyn/resources/prod_documents/DOC0841.PDF
2004-05-12
To CD control logic

- Play push-button switch
- Pause push-button switch
- Stop push-button switch
- Next/previous pulse switch

4.7 kΩ

To AVR Port A

2
1
0
5
4
Appendix B – Log of communication between AVR and T39

WRAP THOR: WRAP THOR AI ($Name: AI-0-0-2 $ Jun 2 2003)
WRAP THOR: Copyright (c) 2003 Bluegiga Technologies Inc.
WRAP THOR: READY.
AVR      : SET BT AUTH * 220
WRAP THOR: SET BT AUTH * 220
AVR      : INQUIRY 6 NAME
WRAP THOR: INQUIRY 6 NAME
WRAP THOR: INQUIRY_PARTIAL 00:02:c7:20:8e:8d 320104 P70
WRAP THOR: INQUIRY_PARTIAL 00:80:37:ba:52:95 520204 Ericsson T39
WRAP THOR: INQUIRY 2
WRAP THOR: INQUIRY 00:02:c7:20:8e:8d 320104
WRAP THOR: INQUIRY 00:80:37:ba:52:95 520204
WRAP THOR: NAME 0
AVR      : CALL 00:80:37:ba:52:95 1101 rfcomm
WRAP THOR: CALL 00:80:37:ba:52:95 1101 rfcomm
WRAP THOR: CALL 0
WRAP THOR: CONNECT 0 RFCOMM 0
// AVR starts to communicate with T39 through the Wrap Thor //
AVR      : ATE=0
T39      : ATE=0
AVR      : AT*EAM="Sony XE-220"
T39      : OK
AVR      : AT+cscs="8859-1"
T39      : OK
AVR      : AT+cscs="UTF-8"
T39      : OK
AVR      : AT+CMER=0,0,0,0,0
T39      : OK
T39      : *EAAI
AVR      : AT*EASM="Sony XE-220",1,1,6, "Play", "Pause", "Stop", "Next", "Previous", "About"
T39      : OK
T39      : *EAMI: 1
AVR      : AT*EASM="Sony XE-220",1,1,6, "Play", "Pause", "Stop", "Next", "Previous", "About"
T39      : OK
T39      : *EAMI: 3
AVR      : AT*EASM="Sony XE-220",1,3,6, "Play", "Pause", "Stop", "Next", "Previous", "About"
T39      : OK
På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Uphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: http://www.ep.liu.se/

© Johan Florén, Daniel Lindberg