TAMPEREEN TEKNILLINEN YLIOPISTO

JUSSI HANNULA

MICROPHONE LINE USED FOR DATA TRANSFER

Master of Science Thesis

Examiner: Professor Karri Palovuori
The subject approved by the
department council 08.06.2011

ABSTRACT

TAMPERE UNIVERSITY OF TECHNOLOGY
Master’s Degree Programme in Electrical Engineering
HANNULA, JUSSI: Microphone Line Used for Data Transfer
Master of Science Thesis, 91 pages, 2 Appendix pages
August 2011
Major: Electronics
Examiner: Professor Karri Palovuori
Keywords: ECI, I^{2}C, microcontroller, accessory, design, measurement, processor

The accessory microphone line of Nokia mobile phones can be utilized to provide a channel for data transmission in addition to the traditional microphone use. The transferred data can be, e.g. button press information in an accessory, such as a headset. The connection between the phone terminal and the outside device is implemented using a standard 3.5 mm jack receptacle, which includes left and right audio channels, a microphone line and a ground connection.

The switching of the microphone line to the data transmission purposes and the integration of the actual data transmission interface is a product-specific variable. The goal of this thesis is to design an electrical circuitry, which allows the microphone line to be separated from the phone terminal’s microphone input and to be used to relay data between the phone and the connected accessory. The implementation of the electrical circuitry is designed so that it is suitable for mass production.

The most important tasks for implementation outside of the microphone line, are the ability to convert the data protocol used by the accessories to the data protocol supported by the phone terminal and vice versa, and also to store the received data in case the phone terminal is busy doing other tasks.

The used data protocols and electronics behind them are covered in order to create a sufficient level of understanding to the matters at hand. The design process of the electrical circuitry will be covered from the beginning; starting from the introduction of the interface under design, advancing to the selection of the base components and finally, designing of the schematic.

The process differs to some extent from the usual design flow where everything can be started from scratch. Some of the already placed components and functionalities of the interface under design are parallel or common with other interfaces inside the phone terminal and therefore, cannot be altered. The requirements and limitations set by the specification documents and the parallel functionalities concerning the design are fully covered and explained.

The functionality of the designed interface was verified and the operation of the electrical circuitry was tested in order to find possible errors from the design and provide the information for making the necessary corrections or modifications. Signal integrity and timing measurements were carried out with the help of an oscilloscope.

TIIVISTELMÄ

TAMPEREEN TEKNILLINEN YLIOPISTO
Sähkötekniikan koulutusohjelma
HANNULA, JUSSI: Mikrofonilinjan hyödyntäminen tiedonsiirtoon
Diplomityö, 91 sivua, 2 liitesivua
Elokuu 2011
Pääaine: Elektroniikka
Tarkastaja: professori Karri Palovuori
Avainsanat: ECI, rajapinta, I^2C, mikrokontrolleri, lisälaite, suunnittelu, mittaus

Nokian matkapuhelimien mikrofonilinjaa on mahdollista hyödyntää tavallisen mikrofonikäytön lisäksi datan siirtoon. Välikappaleena puhelimen mikrofonilinjan ja ulkoisen laitteen välillä toimii tavallinen 3,5 mm audioliitin, joka sisältää mikrofonisisääntulon lisäksi vasemman ja oikean audioulostulon ja maakytkennän.

Mikrofonilinjan erottaminen datansiirtoon ja itse datan välittäminen puhelimen käsiteltäväksi vaatii tuotekohtaista suunnittelua. Tämän diplomityön päällimäisenä tarkoituksena oli suunnitella elektroninen piiri, joka mahdollistaa mikrofonilinjan erottamisen puhelimen sisäisestä mikrofonisisääntulosta ja linjan käyttämisen tiedonsiirtoon puhelimen ja kytketyn lisälaitteen välillä. Piiri tuli suunnitella siten että sitä on mahdollista hyödyntää massatuotannossa sellaisenaan.

Suunnitellun piirin tulee pystyä dataväylän erottamisen lisäksi muuntamaan lisälaitteen käyttämä tiedonsiirtoprotokolla puhelimen käyttämän tiedonsiirtoprotokollan mukaiseksi. Muunto täytyy pystyä suorittamaan myös toiseen suuntaan. Piirin täytyy myös kyetä säilyttämään lisälaitteen lähettämää tietoa kunnes pääprosessori lukee tiedon, mikäli prosessori juuri kyseisellä hetkellä on kiireinen tehdessään muita toimenpiteitä.

Käytettyihin tiedonsiirtoprotokolliin liittyvä, teoria etenkin elektroniikan kannalta katsottuna, on käsitelty teoriaosuudessa, jotta suunnitellun piirin toiminta voitaisiin ymmärtää syvällisesti. Elektronisen piirin suunnitteluprosessi on käsitelty alusta saakka; aloittaen rajapinnan esittelystä edeten komponenttien valintaan ja aina piirin kytkentäkaavion suunnitteluun saakka.

Suunnitteluprosessi eroaa jonkin verran tavanomaisesta. Piirin suunnittelussa ei voida lähteä liikkeelle tyhjältä pöydältä, sillä suuri osa tärkeimmistä komponenteista ja mikropiireistä on jo asetettuina suunniteltuun piirin ja ne asettavat tiettyjä ehtoja ja rajoituksia suunnittelulle. Ympäröivistä piireistä ja olemassa olevista lisälaitteista johtuvat vaatimukset on käsitelty ainoastaan oleellisilta ja vaadituilta osin riittävän taustatiedon tarjoamiseksi.

Suunnitellun rajapinnan ja elektronisen piirin toiminnallisuus testattiin ja todennettiin, jotta mahdolliset suunnitteluvirheet ja piirin toiminnassa mahdollisesti esiintyvät virheet löydettäisiin ja voitaisiin korjata. Sähköisten signaalien laatu ja ajoitukset mitattiin oskilloskooppia hyödyntäen.

PREFACE

This Master of Science thesis, **Microphone Line Used for Data Transfer**, was written for Nokia Corporation in Tampere, Finland.

I want to express my greatest gratitude to my supervisor and mentor M.Sc. Jarmo Kangasaho, who made this project possible in the first place. Also a huge word of thanks for reading through my thesis and for being so excited about it all the time.

I also want to thank my colleagues at Nokia, especially M.Sc. Ville Rantala, for the excellent working environment, all their support and encouraging comments.

I also would like to thank Prof. Karri Palovuori for examining my thesis.

Finally, big thanks to all the colleagues, who had anything to do with my thesis work and to my family and friends for supporting me during my studies.

TABLE OF CONTENTS

TERMS AND SYMBOLS

ADC	Analog-to-digital converter
AHJ	American Headset Jack
ASIC	Application-specific Integrated Circuit
AV	Audio-video
BJT	Bipolar Junction Transistor
CMOS	Complementary Metal–Oxide–Semiconductor
CPU	Central Processing Unit
DC	Direct Current
DCS	Digital Cellular Service
ECI	Enhancement Control Interface
EMC	Electromagnetic Compatibility
EMI	Electromagnetic Interference
ESD	Electrostatic Discharge
GPS	Global Positioning System
GND	Ground
GSM	Global System for Mobile Communications
IC	Integrated Circuit
I^2C	Inter-Integrated Circuit
I/O	Input/Output
MOSFET	Metal–Oxide–Semiconductor Field-Effect Transistor
MSIC	Mixed Signal Integrated Circuit
NFC	Near Field Communication
NMOS	N-channel MOSFET
PCS	Personal Communications Service
PWB	Printed Wiring Board
RF	Radio Frequency
SCL	Serial Clock Line
SDA	Serial Data Line
TPI	Tiny Programming Interface
TVS	Transient Voltage Suppression Diode
TWI	Two Wire Interface
Vcc	Supply voltage
Vdd	Supply voltage
C	Capacitance
τ	RC time constant
R	Resistance
V	Voltage
t	Time
e	Euler's number
T	Signal rise time

1 INTRODUCTION

By just browsing the Nokia on-line store, one would notice that the product range includes not only mobile phones, but many other products as well. Accessories are a wide group of products designed to be used with Nokia mobile phones in order to add additional features, while providing more extensive user experience. Later in this thesis Nokia mobile phone is often referred to as phone terminal. [1]

Accessories are generally devices which are not possible or practical to integrate to the phone itself, e.g. memory cards, headsets, cables, chargers. A battery is also considered an accessory as it is actually not part of the mobile phone itself and in most cases can be replaced by the user. The word accessory also covers items which have nothing to do with the functionality of the phone or electricity. Such items include carrying cases or phone holders for example. An accessory in general is an item designed for use with a base item. [1]

When speaking of accessories, the meaning of the word in the context of this paper is limited. It is used as a common name for all the accessories that can be connected to the phone terminal’s 3.5 mm jack receptacle. An accessory supported by the phone terminal could be for instance, a video cable, basic headphones, a car kit, headset with a microphone and so on. When such an accessory is being connected to the receptacle, it gets connected to the Nokia AV interface.

Nokia AV interface is a specific analog/digital communication system between an accessory and the phone. The interface is used to create an electrical audio and video interface between phone terminal and wired accessories. AV interface consists of the audio/video interface and the Enhancement Control Interface (ECI). The focus of this thesis is mainly on the ECI, but understanding some parts of the audio interface is a necessity in order to understand the activities within the ECI.

Some advanced accessories developed by Nokia may have added value features such as headphones with a volume control. Such an accessory is called an ECI-accessory. An example of an ECI-accessory is presented in Figure 1.1. In order to enable the volume controlling, the information of volume button presses must be somehow delivered to the phone terminal. The actual function of controlling the volume is done in the phone terminal. Only the command to lower the volume level is being sent from an ECI-accessory in this instance. [1]

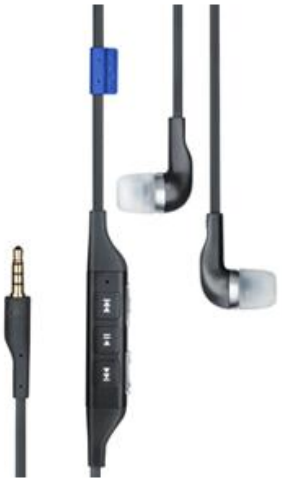


***Figure 1.1.** An ECI-accessory [1]*

The processing of the data generated by the button press utilizing the ECI is done by the main processor of the phone terminal. The data protocols used by an accessory and the main processor are different and therefore a block where the data conversion is done must be included. ECI is a data transmission interface with the purpose of making data transmissions between the phone and the accessory possible.

Without the ECI, the AV interface through the 3.5 mm jack receptacle is only capable of providing an audio and a video output and a microphone input. Video output and microphone input cannot be utilized simultaneously because both are using the same microphone line for transferring the data. ECI-protocol is a confidential Nokia proprietary data communication protocol, and therefore the ECI-accessory functionalities work only on a device having an ECI implemented, i.e. most of the Nokia mobile phones. An ECI-accessory works normally when connected, for example, to an mp3 player. In this case, only the ECI-functionalities are inoperative.

1.1 The purpose of this thesis

Nokia mobile phones have usually had a designated ASIC to handle all the data communications through the Nokia AV connector. The design has usually been implemented as a part of some greater functionality ASIC, normally mixed signal IC that includes most of the powering and analog audio circuitries. In this case however, the implementation with a mixed-signal ASIC was not possible, because the basic design is implemented with a commercial chipset and Nokia has very little room for affecting the specifications of the chipset.

The task was to design and test the electrical circuitry for the needed data communication paths between the Nokia AV connector and the phone. The main problem is the fact that the AV connector does not have a designated data line. The audio channels are solely reserved for audio use and the microphone line is mainly, as the name suggests, meant to use with a microphone. However, it is possible to utilize the microphone line for other purposes by muting it. The ECI takes advantage of this opportunity and uses the line for ECI-communication.

The users expect full time functionality from the control buttons, which means that the volume controlling, for example, must work even when the microphone is in use. A button press automatically mutes the microphone and the phone terminal switches the microphone line to ECI-use for as long as the data communication lasts. A button press causes only a little pause to the microphone audio. The break should not be longer than a few milliseconds.

The functionality of the ECI-protocol and the basic principles of the ECI are always the same, but the implementation is a product-specific variable. The design, however, is intended to be used with other chipsets in the future as well. The future aspect must be taken into consideration during the selection of the supply voltages, signal logic levels and so on.

The design of the ECI must follow the predetermined specifications to be compatible with the accessories already in the field. Also the understanding of the different protocols is important not just in designing of the electrical circuitry, but also in order to understand the functionality of the entire AV interface. The understanding provides knowledge for testing and measuring the functionality of the interface. It is easy to do the measurements with minimal information regarding the electrical circuitry, but deep understanding provides the tools to make corrections if something does not go according to the plan.

1.2 The structure of this thesis

Most of the theory required for understanding the functionality of the ECI is discussed in chapters 2 and 3. In these chapters, a foundation for understanding the functionality of the ECI and the designed circuit is created. Chapter 2 introduces the parts of the Nokia AV interface, which are important from the ECI point of view. The chapter is an introduction to the point and the interface is not discussed in its entirety. This is because many things are context-specific and therefore complemented on the following chapters.

Chapter 3 concentrates on offering an extensive overview of the data communication protocols used. Chapter 4 concentrates on the electronics concerning the protocols discussed in chapter 3. All the used electrical output stages for data transmissions are covered.

The entire design process is covered in detail from beginning to end in chapters 5 and 6. All the electrical and other specifications needed for designing the electrical circuitry are presented and the decisions are rationalized. The process of detecting an accessory and the functionality of the ECI are explained.

The measurement methods used in this thesis are described in Chapter 7. The measurement results are analyzed and the root causes for the possible functional failures or abnormities in measurement results are explained. Chapter 8 includes all the conclusions of the thesis. Possible future recommendations, proposals for improvement, evaluation of the work done, usefulness of the design and measurement results are discussed.

2 NOKIA AV INTERFACE

This chapter provides an introduction on the Nokia AV interface. The Nokia AV connector and related failure situations are reviewed. ECI is introduced and the basics of the accessory detection and powering of the accessory are also discussed. Items concerning the ECI will be specified in the following chapters. Thorough description of the accessory detection and the powering of the accessory will be given when the final structure of the designed ECI-block is known.

The Nokia AV interface is a wired mixed signal interface based on a 3.5 mm, or 2.5 mm in some cases, 4-pole miniature concentric audio plugs and jacks, which is used to connect a master device to a slave device. The master device is the phone terminal and the slave device is the accessory connected to the phone terminal. Connecting the two devices using the AV connector will enable transmission of power, stereo audio out and enhancement control data from master to slave, and transmission of microphone audio and accessory identification, configuration and status data from slave to master. In addition to this, the AV connector may also be used for transmission of a composite video signal from master to a display device.

The AV interface consists of the audio interface and the ECI. The interfaces are connected to the outside world via the AV connector presented in Figure 2.1. Nokia has specified the electrical interface between the master and the slave using the AV connector. The specification document concerning the ECI is confidential and only given to manufacturers under a non-disclosure agreement. All the electrical or other needed specifications concerning the ECI are given in the following chapters.

2.1 AV connector

The Nokia AV connector is the electrical and physical interface used to connect the phone and an accessory. The pin order of the AV connector is presented in Figure 2.1. The connector plug has four lines. The PLUGDET-functionality connects pins one and six when the lug is inserted to the counterpart, indicating to the phone terminal that a plug has been connected. HSEARL is the left audio channel and HSEARR is the right audio channel. HSMIC is the microphone line and according to its name it is normally used to connect a microphone or alternatively used as a composite video output, but the line can also be switched by the phone terminal to be used as a data path for the ECI-accessories. In addition to data transfer properties, the line is also used to provide microphone bias voltage and a supply voltage to accessories. Signals of the Nokia AV interface are presented in Table 2.1.

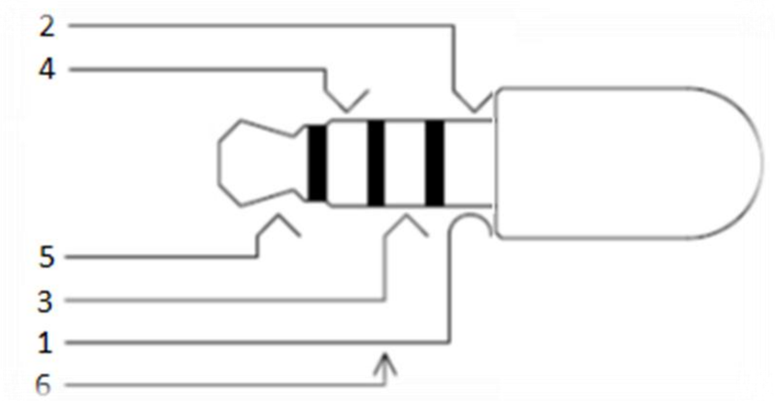


Figure 2.1. *Nokia AV connector pin layout*

Table 2.1. *Signals of the Nokia AV interface*

Pin Number	Signal Name	Direction	Description
6	PLUGDET	In	Host internal connection, plug detection
5	HSEARL	In / out	Audio output
4	HSEARR	In / out	Audio output
3	HSMIC	In / out	Multiplexed microphone audio and control data, C-video out, microphone bias supply voltage (MICBIAS)
1, 2	GND	-	Ground contacts (or other reference level)

Some compatibility issues may occur, if a Nokia accessory with the Nokia AV connector is used in normal, meaning not a Nokia specified, 3.5 mm receptacle. There is a possibility that the ground connection of the receptacle does not get connected to the plug's ground connection. The reason for the bad or wrong connection is usually an undersized receptacle or accordingly can be said, that the plug is oversized. As a result, the plug does not fit in its entirety inside the receptacle and the grounds do not have a proper connection. Clarifying pictures are presented in Figure 2.2.

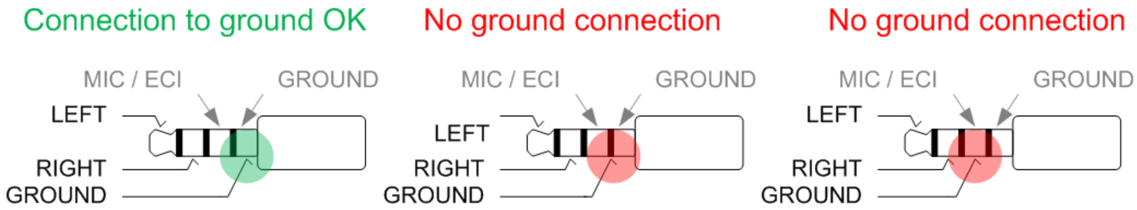


Figure 2.2. *Nokia AV interface compatibility*

According to Nokia's specification, the pin order of an AV connector should follow the pin order presented in Figure 2.1 and Table 2.1, although that is not always the case on the connected accessory. For example, headphones do not have support for a microphone, nor for ECI-communication. The HSMIC-contact surface is usually grounded in headphone plugs. A Basic headphone plug is presented in Figure 2.3.

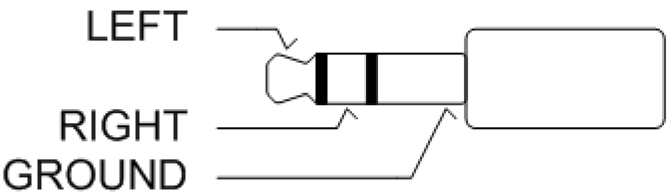


Figure 2.3. *A basic headphone connector*

The AV connector as well as the AV interface is fully compliant with the OMTP (Open Mobile Terminal Platform) specification. OMTP is an operator-sponsored non-profit forum serving the needs of each and every link in the mobile phone value chain by gathering and driving phone terminal requirements. This means that all accessories manufactured following the OMTP specification function with Nokia mobile phones. Accordingly, all Nokia-manufactured accessories, except the ECI-functionalities, function if connected to any mobile phone manufactured following the OMTP specification. [1], [2], [4]

2.1.1 AHJ

Headphones do not cause problems when connected to Nokia AV connector as the missing HSMIC-connection is taken into account in specification. Much more problematic is a plug where HSMIC- and GND-connections have been reversed. Such a plug is called American Headset Jack (AHJ) and it is used by some American companies. If an AHJ plug is being connected to a Nokia specified receptacle, HSMIC-line gets connected to GND and vice versa.

In case of an AHJ plug and a grounded HSMIC-connection, the accessory gets detected as headphones. The audio output channels would work as usual, but only in theory because the audio channels would not have a proper ground connection. The microphone line would be grounded and the possible microphone would not work at all.

To provide support for AHJ-plugs, some extra switching components can be placed on the circuit board in order to switch the HSMIC- and GND-lines. The drawback is that it causes additional resistance to ground. Good ground at the phone terminal side is extremely important for achieving a good EMC performance. Low common ground resistance is important to avoid an electrical echo via common ground from the audio outputs to microphone input. The bigger the ground resistance, the poorer the EMC protection will be because of the unwanted extra series resistance relative to ground. The common ground between the audio output and the microphone line needs to be designed the way that the path resistance is minimized. [29]

Undesired capacitive, inductive and conductive coupling, i.e. crosstalk is caused by two channels using the same shared electrical lead to ground. The common ground impedance defines the crosstalk performance of the system including crosstalk from ECI communication to the earpieces. Therefore, the microphone line and the audio outputs must have separate ground lines on the phone terminal side. The separation of the grounds would cause some extra switches alongside some alternative signal routing to be placed on host side implementation. [29], [11]

2.2 ECI

The ECI controls the data traffic between the phone and the accessory through a bi-directional data line. It is a slow digital interface decoupled from a microphone line. The ECI-block communicates to the accessory using the AV connector's HSMIC I/O-port.

The ECI-accessory supports only the Nokia proprietary ECI-data transfer protocol and the main processor does not have support for it, therefore, two different data transfer protocols have to be used to transfer information. The ECI-block must also be able to handle the data conversion between the used protocols.

The communication between accessories and the main processor via the ECI occurs using the I^2C-protocol. Communication between the interface and the accessory occurs using the Nokia proprietary ECI-protocol. Both protocols are bi-directional, but I^2C uses two lines for communication, and ECI uses only one. There are several different data transfer protocols available for use with the main processor, but the I^2C-protocol is the most adequate solution, because it is simple to implement and faster transfer protocol is not needed.

Generally speaking, the logic high voltage level on both protocols is not fixed. The main processor used in this design, however, uses fixed internal pull-up voltage of 1.8 V to power the line. ECI-protocol uses bit higher logic high voltage level. For that reason, a voltage level adaptation must be done at some point along the electrical circuitry.

The system block diagram illustrating the functionality of the intended solution for the ECI is presented in Figure 2.4. The audio interface and the ECI have a strong mutual dependence and have some overlapping functionalities. It is not possible to determine exactly which part of the AV interface or the block diagram belongs to the audio interface and which parts to the ECI. The dash line in the figure is there to outline the part of the overall block diagram where the actual ECI-communication and the data conversion occur.

The MSIC is a circuit, which includes most of the powering and analog audio circuitries of the phone terminal. It includes the electronics to detect plug insertion and removal. After insertion, the MSIC starts an accessory detection to differentiate between possible accessories. If the HSMIC-line is needed for the ECI-communication during the detection phase, the ECI-controller must be woken up. The wake up command is sent by the main processor. Before the main processor can wake up the ECI-controller, the MSIC must inform the main processor about the need for the ECI-communication.

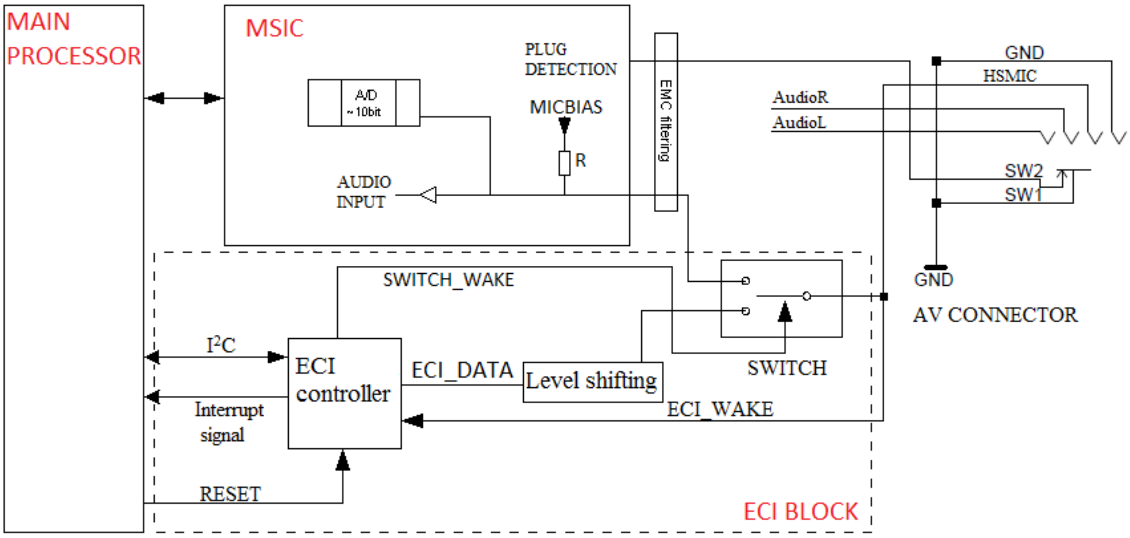


Figure 2.4. *The intended block diagram of the ECI*

The ECI-controller takes care of the data transfer to accessory, be it data reading or writing. For reading cases, the ECI-controller will activate the interrupt signal to the main processor when the ECI-data is ready to be read. For writing, the main processor must activate the ECI-controller using I^2C to transfer the data to the ECI-controller. The activation occurs only during the detection phase, because at all other times, the need for communication comes from the accessory.

To enable the communication between the ECI-controller and the accessory, the ECI-controller must activate the SWITCH_WAKE-signal to switch the HSMIC-line to ECI-communication use allowing the ECI-controller to start data transfer with the accessory. Immediately after the ECI-communication, the ECI-controller switches the line back to audio use.

After the detection phase the ECI-controller may be in idle mode for a long time, until the user presses a button. The ECI-controller must be woken up from the idle mode before it can handle the ECI-communication. The ECI_WAKE-line is routed directly from the HSMIC-line to the ECI-controller. A button press pulls the HSMIC-line and the ECI_WAKE-line to the ground and the falling edge of the HSMIC-signal immediately wakes up the ECI-controller, which activates the SWITCH_WAKE-signal and the line is once again ready to transfer ECI-data.

2.3 ECI-accessory

ECI-accessories are usually powered by the microphone bias supply provided by the MSIC. Later in this thesis the microphone bias supply is often referred to as MICBIAS. The HSMIC-line is used to provide enough power to charge a capacitor inside the ECI-accessory presented in Figure 6.4. As the MICBIAS-line must be switched off from the ECI-data path during the ECI-communication, the charged capacitor as well as the pull-up voltage of the ECI_DATA-line is used for powering the accessory. The HSMIC-line can be switched to the ECI-communication use only when the communication is needed in order to be able to charge the powering capacitor effectively and so that the microphone audio could be utilized, i.e. the HSMIC-line can be switched to the ECI-communication use only when the ECI-data is actually being transferred.

An ECI-accessory may also have its own internal power supply. In these devices, the bias supply from the phone terminal is usually used only for microphone biasing. The detection mechanism for these devices is exactly the same as for accessories powered by the phone terminal. Although, it is also possible to use the phone terminal's bias supply to power the ECI-accessory's control logic even if a local supply is available. A car kit accessory is a prime example of an application which uses its own supply; a car battery. Also some home stereo products such as external speakers may have their own supply.

The detection between different kinds of accessories is based on the voltage measurement in HSMIC-line. The same microphone bias supply voltage, which is used to power accessories, is also used during the detection phase. In addition to the microphone bias voltage, the detection phase also uses a pull-up voltage mainly for separating an open cable from other accessories.

Based on the measured voltage level, accessories are basically being divided into ECI-accessories and non-ECI-accessories, the latter of which are being sorted out individually depending on the measured voltage level. If the measured voltage suggests that the accessory is an ECI-accessory, the phone terminal tries to start the ECI-communication with the accessory.

A non-ECI-accessory is an accessory having audio inputs, a microphone output or both. A non ECI-accessory could also be a video cable presented in Figure 2.5. An example of a non-ECI accessory could be basic headphones, presented in Figure 2.5, which consist merely of one or two earpieces. It does not include any active parts, and the phone terminal cannot differentiate them from each other. For example, the phone terminal cannot detect whether the headset is mono or stereo.

Figure 2.5. *Examples of non-ECI-accessories; basic headphones and a video cable [1]*

Some phone terminals which do not have the ECI at all, may have a support for long and short button press detection. Such a feature could be used, for example, when answering and ending a call. The button is connected in parallel with the microphone. A button press draws the microphone line to ground, which is detected and handled by the MSIC. An example of such an accessory having an answer/end call button could be a basic headset, presented in Figure 2.6, which consists of a microphone and an earpiece.

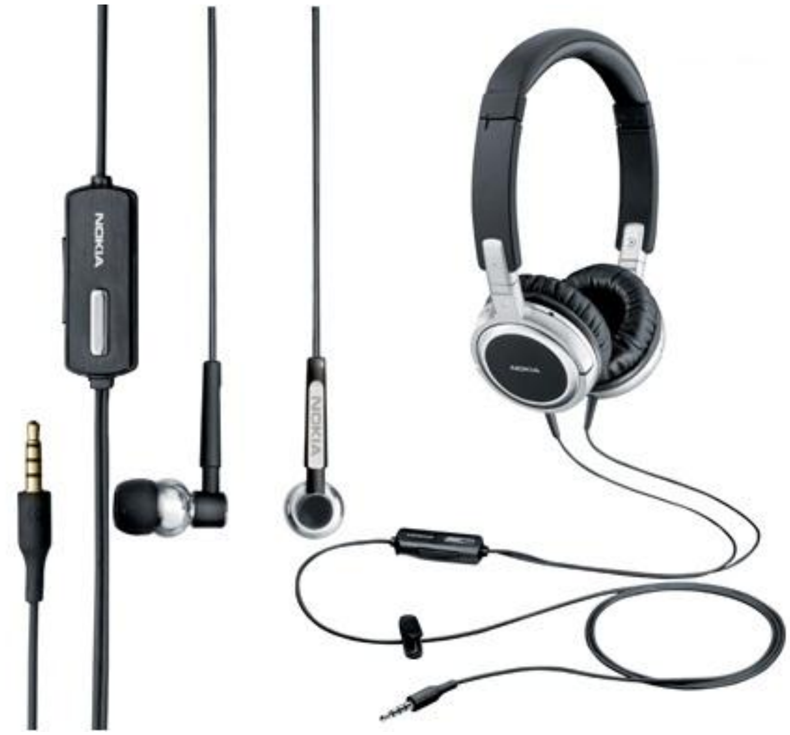


Figure 2.6. *Two headsets with a microphone, an earpiece and an answer/end call button [1]*

ECI is needed for some advanced accessories that enable added value features, which are implemented by using an internal ECI-ASIC. The ECI provides the data communication channel between the ECI-ASIC and the phone terminal. Figure 2.7 presents two examples of ECI-accessories. Examples of the ECI-accessories already in the field include:

- headset with earpiece and microphone with multiple buttons (e.g. volume +/-, play, stop, rewind, skip)
- induction loop for hearing impaired people
- speaker phone for car environment
- loudspeaker accessory without a microphone
- a car kit.

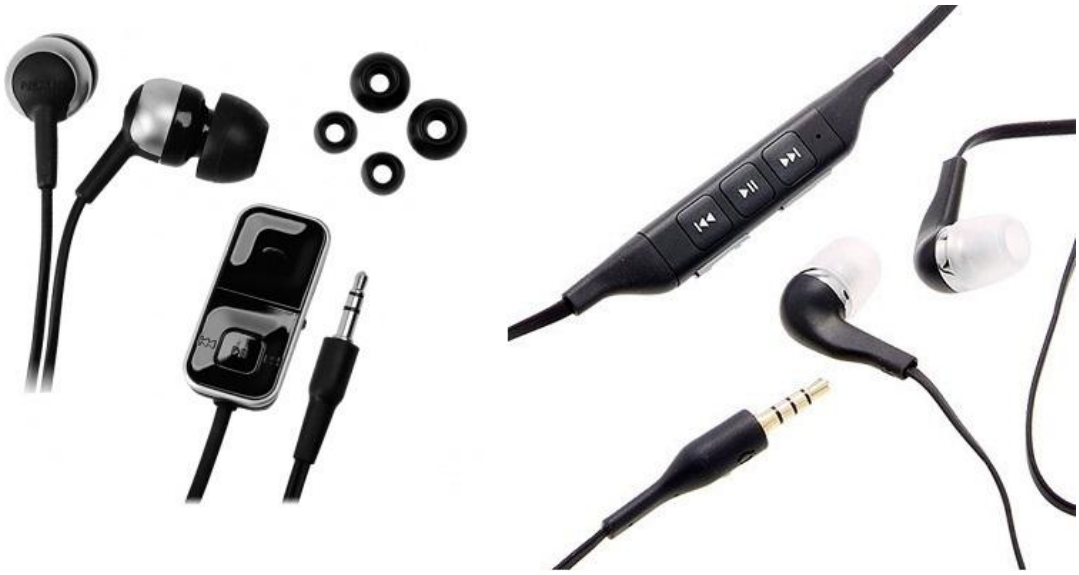


Figure 2.7. *Two headsets with some integrated basic audio control functions [1]*

An ECI-accessory with an electret microphone, such as an ECI-headset, typically contains an ECI-compliant ASIC or a microcontroller, which includes memory and I/O-ports for identification and control purposes. Accessory detection by the master device is made by reading accessory information from the memory of the accessory. Every ECI-accessory has its own parameters and these are stored in ECI-accessory memory. The stored parameters include the needed configuration and identification data.

3 DATA COMMUNICATION PROTOCOLS

In order to make the data communication possible, there must be a common set of rules and instructions that each device follows. A specific set of communication rules is called a protocol. The purpose of this chapter is to introduce the data communication protocols used to relay information between the concerning interfaces. [30]

All data communications concerning the main processor occur using the I^{2}C-protocol. Its integrated addressing and data-transfer protocol allow systems to be almost completely software-defined. All things concerning the I^{2}C-protocol are covered only on the basis of making the electronics design and measurements concerning this thesis possible, i.e. many features concerning the I^{2}C-protocol are deliberately left out of the content. For example, only the used speed-mode, i.e. data transfer rate, is being covered while the other possibilities are deliberately being left out of the content. [3]

Data communications concerning externally connected accessories are being handled using the ECI-protocol. Theory behind the I^{2}C-protocol is covered in more detail than the theory concerning the ECI-protocol as the ECI-protocol is currently determined to be classified by Nokia. In this thesis the ECI-protocol is defined on a need-to-know basis. Only a brief introduction concerning the communication protocol is given, but the things affecting the electronics design and the required measurements are being defined in detail.

3.1 I^{2}C-bus specification

I^{2}C-bus is an invention developed by Philips Semiconductors over 20 years ago. It is mainly used to attach low-speed peripherals to a motherboard, embedded systems or, like in this case, to a mobile phone main processor. For example, Nokia uses I^{2}C-protocol in its phones to attach sensors and other peripherals such as touch control, camera control, backlight control, GPS, NFC, ECI and many others. [2]

Several competitors such as Atmel have brought their own I^{2}C-products on the market or even developed their own I^{2}C-compatible communication interfaces. Atmel's version of I^{2}C Interface is called the Two Wire Interface (TWI) and is fully I^{2}C-compatible except for some particularities. As far as this thesis is concerned, both interfaces are identical, and all items referring to the I^{2}C interface are equally valid for the TWI. [5], [6]

3.1.1 I^{2}C-bus features

The I^{2}C-bus is a bi-directional, bus communication interface, which uses only two signal lines for communication. The two I²C-lines are called serial data line (SDA) and serial clock line (SCL). Virtually any number of slaves and any number of masters can be connected onto these 2 signal lines. The number is limited only by a maximum bus capacitance. I^{2}C-devices can be added or removed from a system without affecting the functionality of any other circuits on the bus. This makes the debugging relatively easy and may be the biggest benefit compared to many other bus protocols as the source of the problem is easy to locate and malfunctions can be quickly traced. [2]

The master device initiates and terminates the data transfer on the bus and generates the clock signals to permit the transfer. The addressed device is considered to be the slave. Devices are distinguished from each other by unique software configurable addresses. [3]

The master is always the device driving the SCL-line. The slaves are devices that respond to the master and cannot initiate a transfer over the I^{2}C-bus; only a master can do that. I^{2}C is a multi-master bus, i.e. it is possible to connect more than one master to the bus. More than one master could try to initiate a data transfer at the same time. Therefore, the bus includes data collision detection and arbitration to prevent data corruption if masters simultaneously initiate data transfer. Such features are not necessary in case of this design. The main processor is the master and all other I^{2}C-devices sharing the same bus are slaves, including the ECI-controller. [3], [2]

Data on the I^{2}C-bus can be transferred at rates of up to 100 kbit/s in the Standard-mode, up to 400 kbit/s in the Fast-mode and up to 3,4 Mbit/s in the High-speed mode. ECI-implementation uses the Fast-mode; therefore the maximum clock frequency is 400 kHz. [2]

Due to the variety of different transistor technologies, such as CMOS, NMOS and BJT which can be connected to the I^{2}C-bus, the levels of the logical ‘0’ (low) and ‘1’ (high) are not fixed and depend on the associated level of the supply voltage. Devices with a variety of supply voltages can share the same bus. According to the I^{2}C specification, the logical high and low levels are set as 70% and 30% of the pull-up voltage. By keeping any waveform distortions below 30% of V_{CC}, that portion of the rising edge will not be counted as part of the formal rise time. [2]

3.1.2 I^{2}C-communication procedure

Figure 3.1 presents the different conditions related to a complete data transfer on the I^{2}C-bus. All data transfers follow the format presented in the Figure 3.1. The basic ground rules for the I^{2}C-communication procedure are:

- the data on the SDA line must be stable during the HIGH period of the clock
- the HIGH or LOW state of the data line can only change when the clock signal on the SCL line is LOW

- one clock pulse is generated for each data bit transferred
- all transactions begin with a START (S) condition
- all transactions are terminated by a STOP (P) condition
- every byte put on the SDA line must be 8-bits long
- each byte must be followed by an acknowledge bit
- data is transferred with the most significant bit first. [2], [3]

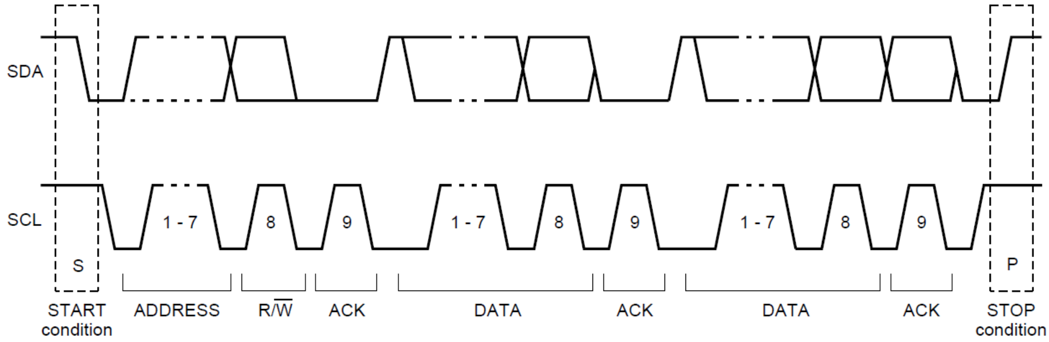


Figure 3.1. *A complete data transfer [2]*

A START condition (S) marks the beginning of a data transaction. The master issues the START condition by indicating a high to low transition on the SDA line while the SCL line is kept high. After the start condition a 7-bit long slave address is being sent by the master. The 8th bit is a data direction bit – LOW state indicates a transmission (WRITE), HIGH state indicates a request for data (READ). The 9th bit is the Acknowledge (ACK) bit or the Not Acknowledge (NACK) bit. At that moment, the master-transmitter becomes a master-receiver and the slave-receiver becomes a slave-transmitter. The first acknowledge is generated by the slave. Subsequent acknowledges are generated by the master. [2], [5]

The acknowledge takes place after every byte and data direction bit. The acknowledge bit allows the receiver to report the transmitter that the byte was successfully received and another byte may be sent. The number of bytes that can be transmitted per transfer is unrestricted. If the NACK is received, the data transfer is immediately terminated by a STOP condition, always generated by the master. STOP condition is indicated by a low to high transition on the SDA line while SCL line is kept high. There are five conditions that lead to the generation of a NACK:

- No receiver is present on the bus with the transmitted address
- The receiver is unable to receive or transmit because it’s performing some other tasks and is not ready to start the communication with the master
- During the transfer the receiver gets data or commands that it does not understand
- During the transfer the receiver cannot receive any more data bytes
- The master signals the slave of the end of the transfer. [2], [5]

However, instead of a STOP condition, if the master still wants to communicate on the bus, it can generate a repeated START (Sr) condition to speed up the communication process. The START and repeated START conditions are functionally identical. By generating the repeated START, the master can address another slave faster as it does not have to generate the START and STOP conditions separately. The bus also needs to be free for a period of time between the START and STOP conditions. Figure 3.2 presents the data transfer on the I^{2}C-bus. Repeated START condition is illustrated in the figure. [2]

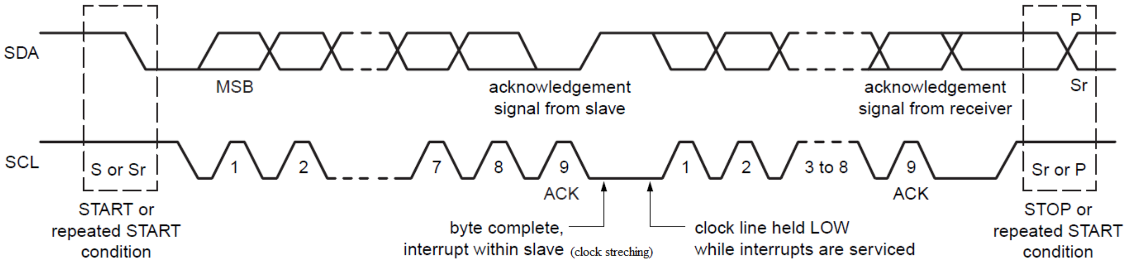

SDA
SCL
P
MSB
acknowledgement signal from slave
acknowledgement signal from receiver
Sr
S or Sr
1
2
7
8
9
ACK
1
2
3 to 8
9
ACK
Sr or P
START or repeated START condition
byte complete, interrupt within slave (clock streching)
clock line held LOW while interrupts are serviced
STOP or repeated START condition

Figure 3.2. *Data transfer on the I^{2}C-bus [2]*

Slave devices are not always fast enough in processing the data, so if a slave cannot receive or transmit another complete byte of data until it has performed some other function, it can pause a transaction and hold the clock line SCL low, in which case the master goes into a wait state. The master generates the clock, monitors the voltage on the SCL bus, immediately knows if there is a problem, and then waits until the slave releases the SCL line. The transaction cannot continue until the line is released HIGH again. Data transfers then continue when the slave is ready for another byte of data and release clock line SCL. [3]

For example, a device may be able to receive the data fast enough but needs more time to store the received byte or prepare another byte to be transmitted. Such a feature is illustrated in Figure 3.2 and is called clock stretching. As an exception to a normal slave device behavior, the slave device having the clock stretching capability must be able to drive the clock line in order to be able to control the clock. For I^{2}C-buses shared by multiple devices, it is important to estimate the impacts of clock stretching as the total bandwidth of the shared bus might be decreased. Causing the clock to stretch, the slowest I^{2}C device dominates the bus performance. Clock stretching also causes some extra current consumption. [2], [3], [5]

3.2 ECI-bus protocol

From the electronics point of view, the ECI-bus is very much like the I^{2}C-bus protocol. Both protocols use the same output stage technique for communication. Basically, a single I^{2}C-bus protocol's signal line is excluded from the design.

High and low levels of the ECI-bus protocol are not fixed, as they are not in the I^{2}C-bus protocol either. The biggest difference between the I^{2}C-protocol and the ECI-protocol is the fact that the I^2C is a synchronous communication protocol having a separate clock line, and the ECI is an asynchronous communication protocol without a separate clock line. [3]

The ECI controller must be able to tolerate large clock tolerances, therefore the ECI controller must be able to adapt to the current speed of the slave. The master adapts to the speed on the basis of the learning sequence sent by the slave. After the learning sequence, the data bit width is T. The sampling at the receiving end is done at T/2, i.e. in the middle of the bit, regarding to the starting rising edge of the START-pulse.

4 INPUT/OUTPUT STAGE ELECTRONICS

The goal of this chapter is to introduce the electronics concerning the implementation of the data buses, as well as provide the information needed for understanding the electrical measurements. Many items concerning the electronics are deliberately left out of the content and only the relevant items concerning the ECI implementation are being covered.

The open-drain output circuit technique, used by I^{2}C-protocol and the ECI-protocol, is covered in depth as it plays a major role in designing the circuit. Whereas the input part just scratches the surface as its significance towards the designed circuit is relatively small, because the input stages are already implemented inside the ICs. In addition, the basics of the push-pull output circuits are covered. [3]

4.1 Open-drain output

An open-drain output is a driver stage that can transmit data by pulling the bus low. An input stage, meant for receiving the data sent by the open-drain output, consists of high impedance sense amplifiers that monitor the bus voltage. The circuit technique allows multiple devices to communicate bi-directionally on a single wire. I^{2}C-protocol requires two wires to work. It consists of two active wires and a ground connection, while the ECI has only one active wire used for communication. Electronic designs behind both bus protocols are basically the same. The only physical difference is the number of wires. [2], [7, p. 487-489]

All I^{2}C-bus outputs are implemented using the open-drain or the open-collector circuit design presented in Figure 4.1. The output is called open-collector when using bipolar transistor and open-drain, if the used device is implemented using FET-technology. [2]

Instead of outputting a signal of a specific voltage or current, the output is not capable of supplying current to the signal line. The needed current is drawn from a positive supply voltage, named as V_{DD} in Figure 4.1, via pull-up resistor, i.e. a resistor tied to V_{DD}. A pull-up resistor provides the simplest means of tying off unused inputs. A great number of inputs can be connected with a pull-up resistor without violating minimum high or low input level requirements, since input currents are very low, usually as low as 1…10 µA for most CMOS devices. [12, p. 205]

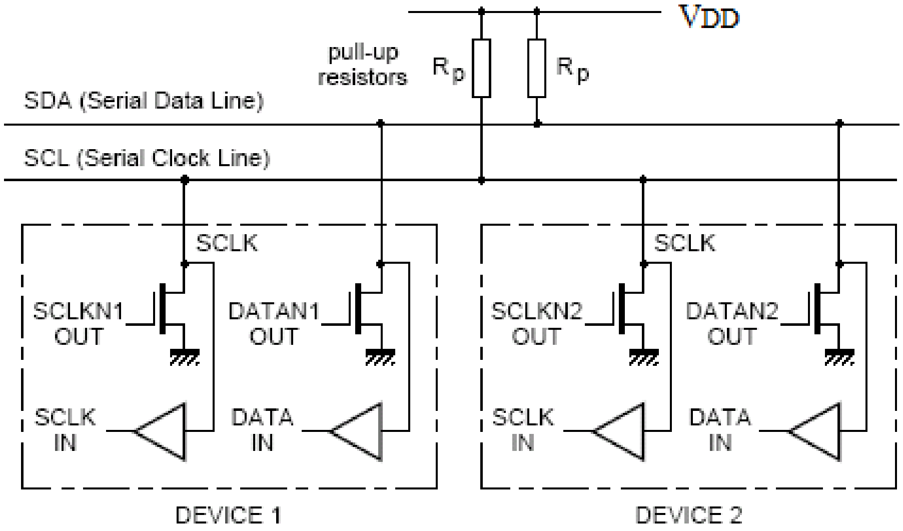


Figure 4.1. *I²C Hardware architecture [2]*

It is generally not wise or practical to connect too many devices together. The longer the line is, the greater the chance for coupled noise and bigger the bus capacitance will be. Also, the troubleshooting of a short-circuited line tied to a great number of points can be very difficult because the exact cause for a problem may be hard to find. One device not working as the way it should, may cause the whole bus to malfunction. [12, p. 205]

The operation of the open-drain output is simple. When the bus is free and the transistors are closed, the signal is on high state. The pull-up voltage determines the value of the high state. The output signal of an I^2C-device is basically applied to the base/gate of an internal transistor. To initiate communication, an I^2C-device pulls the SDA line low by turning on the output stage transistor. Generation of clock signals on the I^2C-bus is always the responsibility of the master, so it is not necessary for the slave device to have an output stage on the SCL bus line. Only the master can enable its pull-up circuit. In order for the slave device to obtain permission for pulling down the SDA line, the master must first send the READ command to the slave device. In other words, the master is controlling the on-time of the slave device’s output transistor. [2]

A series resistor is sometimes needed for protection against high-voltage spikes, or in some cases against overcurrent on the SDA and SCL lines. If series resistors are used, the additional resistance must be taken into consideration when calculating the value of the pull-up resistor and allowable bus capacitance. A series resistor slows down the output fall time and together with the pull-up resistor affects the static low level of the I^2C-lines. The use of a series resistor leads to different voltage levels at master’s and slave’s input during the LOW phase, depending on which device currently pulls SDA or SCL low. Thus, it is possible to recognize which I^2C device is currently active by analyzing the low level on the I^2C lines. [2], [5]

4.1.1 Defining the pull-up resistor

Defining the pull-up resistor is a compromise between the current consumption and the rise time of the signal. Rise time is determined to be the time period between the logical low and logical high levels of the rising edge of the CLK/SDA-signal. [3], [2]

The bus capacitance caused by the parasitic capacitances and the pull-up resistor together form an RC-circuit. According to the Equation 4.1, RC time constant τ is equal to the product of the circuit resistance and the circuit capacitance. It is determined as the time required to charge the capacitor through the resistor to about 63 percent of full charge. Calculating the time constant is not usable as such for defining the resistor, but taking the used logical high and low levels into consideration the equation becomes much more useful. [3], [2]

Capacitor's charging curve is presented in Figure 4.2. The used time unit, presented in the horizontal axis, is the time constant τ. The pull-up voltage with percentages of its final value is presented in the vertical axis. After a period equivalent to five time constants, the capacitor in this RC charging circuit is virtually fully charged and the voltage across the capacitor is now approximately 99% of its maximum value. [3], [2]

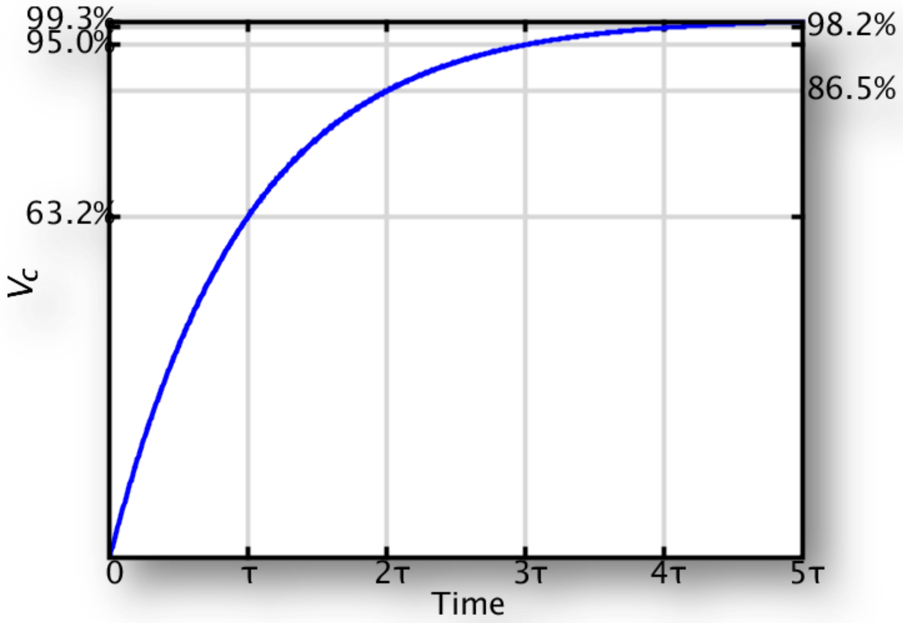


Figure 4.2. *Capacitor's charging curve with percentages of final value [15]*

General arithmetic operation for determining the voltage Vc(t) across the bus capacitance is presented in Equation 4.2, where t is the time since the charging started. The voltage Vc(t) is also the voltage that the input stage of the data bus sees. The voltage $V_{\text{pull-up}}$ is the supply voltage or the pull-up voltage of the bus line presented in Figure 4.1. Equations 4.3 and 4.5 are the same as Equation 4.2, except that they take the related input thresholds into consideration. The time t_1 spent in charging the capacitor to logic low level is presented in Equation 4.4 and the time t_2 spent in charging the capacitor to logic high level is presented in Equation 4.6. Rise time of the signal is the time spent between the logic low and high levels. Calculating the value for the signal rise time T is presented in Equation 4.7. [2], [7, p. 23-24]

According to the I^2C-specification the maximum rise time for a signal working in Fast-mode is 300 ns. The minimum value is determined to be 20 ns and a sum of the actual rise time, which can be calculated using Equation 4.7, multiplied by 0.1. For bus loads up to 200 pF, the pull-up device for each bus line can be a resistor and for bus loads between 200 pF and 400 pF, the pull-up device can be a current source or a switched resistor circuit. [2]

The discharge of an RC circuit occurs following the same principles as the charging of the circuit. According to Equation 4.1 the time constant τ is determined as the time required to discharge the capacitor, through the total resistance of the discharge path, to about 37% percent of full charge. As the charging turns into discharging, Equation 4.2 is adjusted to form presented in Equation 4.8. [2], [7, p. 23-24]

$$\tau = RC \tag{4.1}$$

$$Vc(t) = V_{pull-up} \times (1 - e^{-\frac{t}{RC}}) \tag{4.2}$$

$$Vc(t1) = 0{,}3 \times V_{pull-up} = V_{pull-up} \times (1 - e^{-\frac{t1}{RC}}) \tag{4.3}$$

$$t_1 = 0{,}3566749 \times RC \tag{4.4}$$

$$Vc(t2) = 0{,}7 \times V_{pull-up} = V_{pull-up} \times (1 - e^{-\frac{t2}{RC}}) \tag{4.5}$$

$$t_2 = 1{,}2039729 \times RC \tag{4.6}$$

$$T = t_2 - t_1 = 0{,}8473 \times RC \tag{4.7}$$

$$Vc(t) = V_{pull-up} \times e^{-\frac{t}{RC}} \tag{4.8}$$

The bus capacitance is always an estimate during the schematic design phase as the most significant part of the formation of the total bus capacitance is a sum of many different components, e.g. traces of the PWB, connections and pins. This capacitance limits the maximum value of the pull-up resistor due to the specified maximum rise time. A high bus capacitance can be compensated with a low pull-up resistor and vice versa. [2], [3]

Maximum value for the pull-up resistor can be determined using the Equation 4.9. The size of the final bus capacitance is impossible to know before the final layout design. The value can be simulated with reasonable accuracy, but usually the best resort is to use previously proven estimates and just estimate a suitable resistor value. If the operational performance of the bus does not meet the specifications, the value of the pull-up resistor can always be fine-tuned afterwards. [2]

$$R_{pull-up,max} = \frac{T}{0{,}8473 \times C} \tag{4.9}$$

The bus capacitance in this design must be as low as possible, because more capacitance means bigger value for time constant, which means more reduction to signal rise times. The signal rise times and the current consumption go hand in hand, so the bigger the bus capacitance, the smaller the pull-up resistor and higher the current consumption will be. An educated guess for the bus capacitance in this case is no more than 50 pF at maximum. The main reason for such a low value is the very short wires. Maximum and minimum values for the pull-up resistor in a bus having a bus capacitance of 50 pF are calculated based on Equation 4.9 in Equations 4.10 and 4.11. [2]

$$R_{pull-up,max} = \frac{300\,ns}{0{,}8473 \times 50\,pF} \approx 7\,k\Omega \tag{4.10}$$

$$R_{pull-up,min} = \frac{20\,ns}{0{,}8473 \times 50\,pF} \times 1.1 \approx 519\,\Omega \tag{4.11}$$

When the signal is pulled down, the bus capacitance is slowing down the falling edge just the way it is affecting the rising edge of the signal. The value of the time constant τ can be determined using the Equation 4.1. The only difference during the discharge period is that the capacitor is not discharging through the pull-up resistor, but instead it discharges through the transistor straight to ground. The path to ground is almost non-resistive and the value of the time constant is very small and the effect of the bus time constant can be completely ignored, unless a serial resistor is placed to the data path. [2], [5]

The two possible placements for $R_{series1}$ and $R_{series2}$ are presented in Figure 4.3. If the placed resistor is $R_{series1}$, the voltage low level V_{low} at slave's input when the master-transistor is in a conducting state could be calculated using the Equation 4.12. When the slave-transistor is in conducting state, the master's input voltage low level V_{low} is hardly rising at all, because the pull-up resistor $R_{pull-up}$ is not affecting the formation of the voltage level. [2]

Correspondingly, if the placed resistor is $R_{series2}$, the low voltage level at master's input when slave-transistor is pulling the line to ground could be calculated using the same equation. When the master-transistor is in a conducting state, the slave's input voltage low level is hardly rising at all, because the pull-up resistor is also not affecting the formation of the voltage level. [2]

$$V_{low} = \frac{R_{series}}{R_{pull-up}+R_{series}} \times V_{pull-up} \tag{4.12}$$

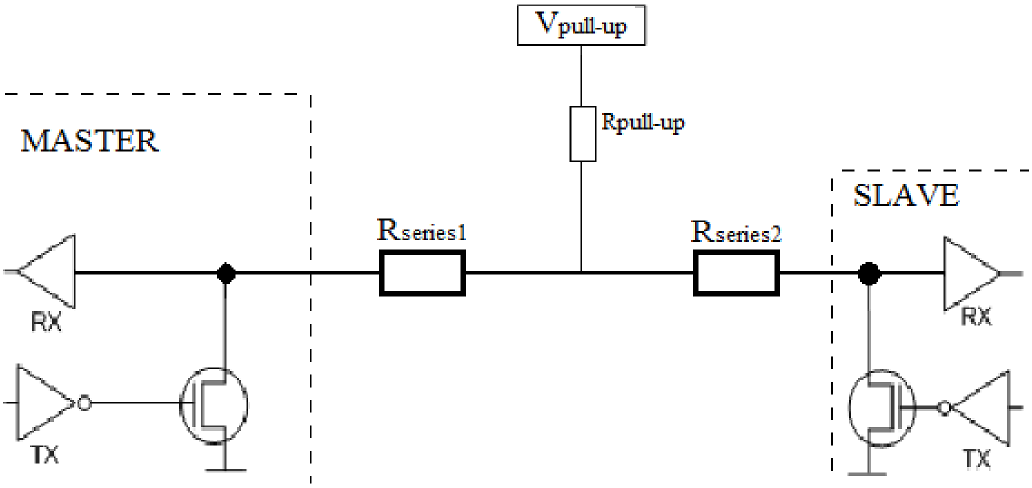


Figure 4.3. *Series resistor in open-drain data path*

The maximum values for the fall time of the signal are determined to be the same as the limits for the signal rise time. According to I^{2}C specification, the maximum fall time for a signal working in Fast-mode is 300 nanoseconds. The limit for the minimum fall time is slightly smaller than the limit for the minimum rise time, because the value of the time constant is smaller during the capacitor discharge than during capacitor charge period. [2]

If a series resistor is inserted, it slows down the falling edges of the signal. The fall time cannot be calculated using the normal RC circuit fall time determining Equations 3.1-3.7 because another determinant factor in the formation of the fall time is the switching speed of the transistor. The maximum value of the series resistor is generally limited by the maximum input voltage low level at one input. Which one, depends of the placement of the series resistor. This all is of course dependent on the switching speed of the transistor. If the switching speed is already near the maximum fall time limit of 300 ns, even a small series resistor would cause the fall time to be too slow and rule out the placement of the resistor completely. [3], [12, p.181-182]

Sometimes a series resistor could also be useful. If a transistor has a switching speed under the minimum fall time limit, a small series resistor could slow down the falling edges enough and also decrease the voltage-spikes for the line to fulfill the I^{2}C specifications and ensure the correct operation of the devices connected to I^{2}C-bus. [3], [12, p.181-182]

4.1.2 Current consumption

Current is required to establish a voltage potential across the capacitor. If a small resistance causing large current, is in series with the capacitor, the charge time will be short. If a large resistor causing small current is in series, the charge time will be long. Longer rise time means lower current consumption. Therefore, the bigger the pull-up resistor, the lower the current consumption. [2]

Input current and off-state output current are extremely small for the used CMOS type I/O-devices. Basically, the bus is drawing current only when a transistor is in a conducting state, but because of the current needed to charge the parasitic capacitances the current consumption slightly increases. The transient current flowing through the selected pull-up resistor when one of the output transistors is open can be determined using the Equation 4.13. The current is also the maximum current possible in the circuit. The small bus capacitance does not have any meaningful significance to the current consumption, except for the slight increase in time needed for the charging of the capacitor after the transistor switch has been closed. The current decreases exponentially during the short charging event, therefore the extra current drawn during the charging period is relatively small. [2]

$$I_{pull-up} = \frac{V_{pull-up}}{R_{pull-up}} \tag{4.13}$$

When the pull-up resistor selections in the electrical design in Appendix A are not predetermined, they are selected to be as big as possible within the limits of the I^{2}C- or any other specification in question in order to achieve as low current consumption as possible.

Two circuit elements placed nearby will always interact capacitively. The faster the rising edge, the bigger the chance of formation of crosstalk due to the mutual capacitance. Sudden changes in current cause greater problems with inductive coupling. Thus, fast edges are also more probable in causing EMC-problems and have to be avoided as well as possible. [10, p. 60-61]

4.2 Schmitt-trigger input

I^{2}C-bus inputs are implemented using high impedance sense amplifiers like Schmitt-triggers. The Schmitt-trigger is a comparator application which switches the output HIGH when the input passes upward through a positive reference voltage. Parallel positive feedback creates the needed hysteresis that is controlled by the proportion between the resistances of R1 and R2 and prevents switching back to the other state until the input passes through a lower threshold voltage. [5], [7, p. 231-232], [9, p. 679-682]

When the circuit input voltage is above the high threshold or below the low threshold, the output follows the input value, i.e. the output voltage has the same sign as the circuit input voltage. Figure 4.4 presents a schematic of a basic non-inverting Schmitt-trigger. [5], [7, p. 231-232], [9, p. 679-682]

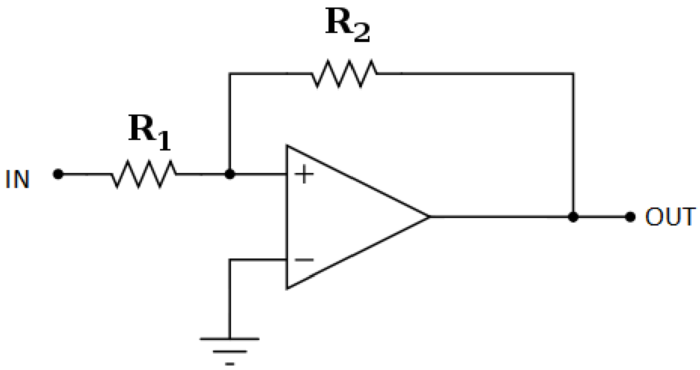


Figure 4.4. *A non-inverting Schmitt-trigger*

Schmitt-trigger has a built-in hysteresis effect, which is very effective in switching against rapid triggering and makes it very effective in noise reduction. Schmitt-trigger action at all inputs also makes the circuit tolerant for slower input rise and fall times. [5]

4.3 Push-pull output

The most common used digital logic output circuit is called push-pull or active pull-up. The output is held either HIGH or LOW through a conducting transistor or MOSFET. Output is generally realized as a complementary pair of transistors or MOSFETs, one sinking current from the load to ground or a negative power supply, and the other supplying or sourcing current to the load from a positive power supply. A basic push-pull output circuit is presented in Figure 4.5 [7, p. 487-488]

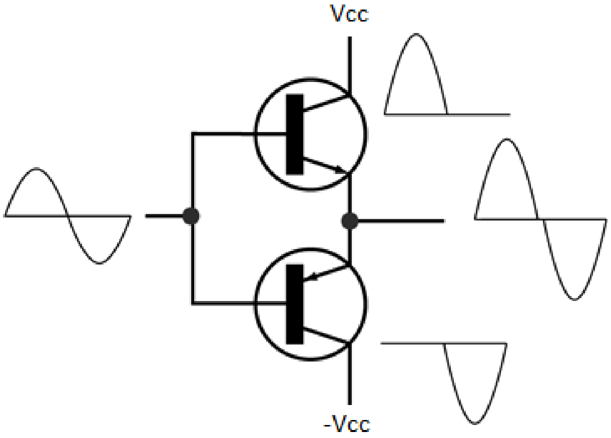


Figure 4.5. *Push-pull output circuit*

Push-pull output provides low output impedance in both states, giving faster switching time and better noise immunity, as compared with an open-drain output with passive drain resistor. In the case of CMOS, it also results in lower power dissipation. Significant power is only drawn when the transistors in the CMOS device are switching between on and off states, i.e. when the gate capacitance is being charged or discharged. Possibly the greatest disadvantage of the push-pull output is the fact that it cannot be used to drive a bus, since the output cannot be disconnected from the shared data lines. The output holds the data line either HIGH or LOW at all times. [7, p. 487-488], [15, p. 25]

Another output stage type is a gate whose output can be floating, i.e. an output port with a high impedance state in addition to the 0 and 1 logic levels. The high impedance state effectively removes the influence of the output from the circuit. Such devices come in two varieties; three-state devices and open-drain devices. Open-drain devices are mainly used in situations, where the speed is negligible or there are no high speed requirements. Figure 6.4 presents a realization of a three-state push-pull output with BJTs. Logic HIGH at the ENABLE-pin enables the output and logic LOW disables the output by putting the output pin in high-impedance mode, i.e. both output transistors in a non-conducting state. [7, p. 487-488]

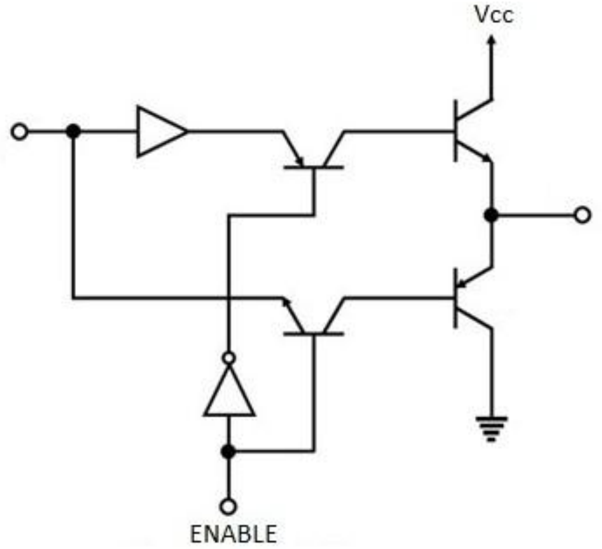


Figure 4.6. *Three-state output circuit [8]*

A separate enable-input determines whether the three-state output behaves like an ordinary push-pull output or goes into the high impedance state, regardless of the logic levels present at the other inputs. A device with three-state output behaves exactly like ordinary push-pull logic when enabled, always driving its output either HIGH or LOW. If the output is disabled, it effectively disconnects its output, so another logic device can drive the same line. So, basically it is a push-pull with a disable feature. [7, p. 487-488]

5 ECI IMPLEMENTATION

This chapter includes the justifications for the selection of the ECI-controller and introduces the ground rules and general requirements concerning the ECI implementation, focusing on the design of the electrical circuitry. Individual component characteristics and the requirements set by them will be introduced in later chapters, although most of the requirements set by the ECI-controller are covered in this chapter.

Unlike many other design issues, EMC is not an area where it is possible to list a set of rules that work with certainty or the rules are considered more as guidelines. EMC must always be considered as a system-level issue. EMC compliance cannot be guaranteed by design; it must be tested. [17], [11]

A good EMC design requires more knowledge than what can put into this thesis and the EMC design itself is not a subject of this thesis, any in-depth review of the subject is left out. Yet, the problems caused by the EMC are essential on the design point of view. Therefore, the problems caused by the EMC and the effects it has on the design of the electrical circuitry are covered as well as the most crucial EMC components, but only on a need-to-know basis. All the capacitor values and the final selections of the components affecting to the certain frequency bandwidths are determined based on the EMC testing.

The first step in the design was to choose the method of ECI-to-I^2C implementation. Many functionalities and electrical parameters were mandatory. ECI must be able to handle the data conversion from I^2C-protocol to ECI-protocol and the other way round, so a block where the data conversion is done must be included. This block is called ECI-controller.

5.1 ECI-controller selection

The first and the least demanding option from the electronics point of view, is to use the main processor to handle the tasks demanded by an ECI-controller. That could be possible in theory, but there is one decisive limitation to be taken into consideration. An ECI-ASIC inside an accessory does not have a buffer memory for button presses, so the button data must be read during the button is pressed. That means there is only limited time to read the data. The main processor might be busy with other tasks and will be unable to carry out the data processing in time. Therefore, the use of the main processor as an ECI-controller is not a feasible option. The controller must be able to act as a time buffer and store the data for the main processor. All in all, the ECI-controller is needed mainly for two reasons:

- to convert the ECI-protocol data into I^{2}C-protocol
- to act as a time buffer.

At the beginning of the actual decision making process there are three viable options from which to choose. The first is to implement the entire ECI-block using an FPGA. The second option is to use an ASIC for implementation. The last option is a microcontroller accompanied by some discrete components. The decision between the implementation possibilities is principally made on the basis of:

- cost
- size versus available space
- schedule.

The cheapest of the three is an ASIC when considering mass production. If the manufacturing lot is small, the cost per ASIC is relatively high. On the other hand, the bigger the amount of manufactured ASICs, the cheaper would the price be per individual ASIC. The expected size of the production lot is substantial, so the expected price is relatively low. [31]

The implementation using a microcontroller as a base of the ECI-block is the considerably more expensive implementation option. The cost of an FPGA based implementation would probably be even higher than the cost of a microcontroller based implementation and the manufacturing costs are not as production-lot-size-dependent as they are on the case of an ASIC. FPGA was basically rejected because it loses the price comparison to ASIC-option and Nokia has much more experience in using ASICs than FPGAs. Both would be made by vendors on the grounds of Nokia's requirements. [31]

An ASIC takes the smallest amount of space. This is true even when compared to an FPGA, which takes significantly more space from the circuit board. The size of a microcontroller is roughly the size of an FPGA, but it needs some discrete components in addition to the actual controller circuit. The spacing and routing would also take extra space if discrete components are used. [31]

A huge advantage of the microcontroller is the maturity of the technology. Microcontrollers in general are widely used and Nokia has much experience using them. If a microcontroller is selected, an off-the-shelf device could be used and the in-house-development could be started right away.

ASIC is a common technology, but its greatest disadvantage is the lack of flexibility and the schedule in general, i.e. the difficulty of making changes to the silicon. If the designed circuit inside the ASIC contains any errors, the fixing process takes a lot of time and money because an ASIC can no longer be altered after production. That is why the design must be perfect, especially when making large quantities of the same ASIC. [31]

The biggest advantage of an FPGA over an ASIC is the flexibility and speed of changes. This is partially because it is not as hardware dependent as the ASIC, and changes do not require as much money. An FPGA is basically a reprogrammable integrated circuit. Because of its programmable nature, it is possible to correct mistakes and to send out patches or updates after the manufacturing. [31]

In case of a microcontroller, the biggest advantage is its superior flexibility to changes. Minor changes or even addition of new features are possible even during the manufacturing due to the re-programming possibility of the microcontroller. There are also several other factors besides the cost, size and timetable issues to be taken into consideration during the decision making process.

A microcontroller is the best solution from the software point of view because the development could be started right away, the development cost would be the smallest and the development could be done in the same location as the hardware designing and testing, which allows face to face communication between the software and hardware personnel. The information between the two parties would travel in real-time, which is a huge advantage if any unexpected errors occur during the overall development process.

Requirements for the ECI were not 100% clear and they tend to change during the development process. It could also be possible that the ECI-controller must be able to handle some additional features, which were still speculations during the planning process. A microcontroller was chosen to be the base of the design, because the overall flexibility is considered to be more important than a few cents of extra cost per unit. Minor additions or changes at a later date would also be possible to implement with a tight schedule, without major effort or resources.

5.1.1 ATtiny20 microcontroller

The decision of which microcontroller to use is a pre-determined decision. Nokia has certain reference components and preferred suppliers from which to choose. Atmel’s ATtiny20 has been proven to be a strong performer and was chosen to be the base of the ECI. The simplified block diagram of the ECI after the decision of using a microcontroller looks the same as the block diagram in Figure 2.4.

ATtiny20 is a low-power CMOS 8-bit microcontroller. It can run either on internal oscillator or external clock. It includes two kilobytes of in-system programmable flash memory, and 128 bytes of internal SRAM, which should be more than enough for the task at hand. [6]

The communication bus to the main processor is I^2C-compliant TWI, which uses two dedicated serial ports with open-drain outputs; SDA and SCL. The microcontroller also has twelve general purpose I/O-ports with all port pins having individually selectable pull-up resistors with a supply-voltage invariant resistance, i.e. open-drain outputs with internal pull-up resistors, which can be used or alternatively, external pull-up resistors can be used. All of those general I/O-ports also have push-pull output stages and the system designer can determine which option to use. [6]

Most port pins have alternate functions in addition to being general digital I/O-ports. Port A pin PA1 has an alternate function of being a positive input of an analog comparator and Port A pin PA2 has an alternate function of being a negative input of the same analog comparator. The comparator can be used to provide an external level triggered interrupt to the microcontroller. [6]

ATtiny20 also provides various sleep modes to tailor the power consumption to the application’s requirements; power-down, standby, ADC noise reduction and idle. The utilized modes are standby and idle. [6]

Idle mode stops the CPU while allowing the SRAM, timer/counter, ADC, analog comparator and interrupt system to continue functioning. In Standby mode, the oscillator is running while the rest of the device is sleeping, allowing very fast start-up combined with low power consumption. [6]

A low level on the reset pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. An external reset pulse will wake-up the microcontroller from any of the sleep modes. [6]

There must be a reprogramming possibility available during the development phase, i.e. after the microcontroller has been placed to the circuit board. The voltage during the programming is 5 V, which is much more than its normal operating voltage. The microcontroller supports operating voltages between 1.8…5.5 V, but the needed and also available supply voltage is 2.5 V at maximum. Thus, the high programming voltage sets some demands to the electrical circuit around the controller. [6]

The electrical characteristics, or more precisely, the I/O-voltage levels, are determined according to the used supply voltage. I^{2}C voltage high level at the main processor end is fixed to 1.8 volts. The ECI-data voltage high level is 2.1 V at minimum, so the supply voltage for the microcontroller must be 1.8 V, 2.1 V or more, depending on the used ECI-data high voltage level. On the electrical circuitry point of view, it does not matter which one of the possible supply voltage values gets chosen, because a voltage level shifting must be done either way.

The lower the supply voltage is, the lower the current consumption will also be. Current consumption is a function of several factors such as operating voltage, operating frequency, loading of I/O-pins, switching rate of I/O-pins, code executed and ambient temperature. The dominating factors are operating voltage and frequency. Current consumption for the ATtiny20 in relation to the used frequency in active mode is presented in Figure 5.1. The figure also presents the dependency between the current consumption and processing speed. If the fastest processing speed is not needed, the system designer is allowed to optimize the system by lowering the processing speed. On the basis of the current consumption and the needed voltage levels, the supply voltage of the microcontroller was decided to be 1.8 volts. [6], [18]

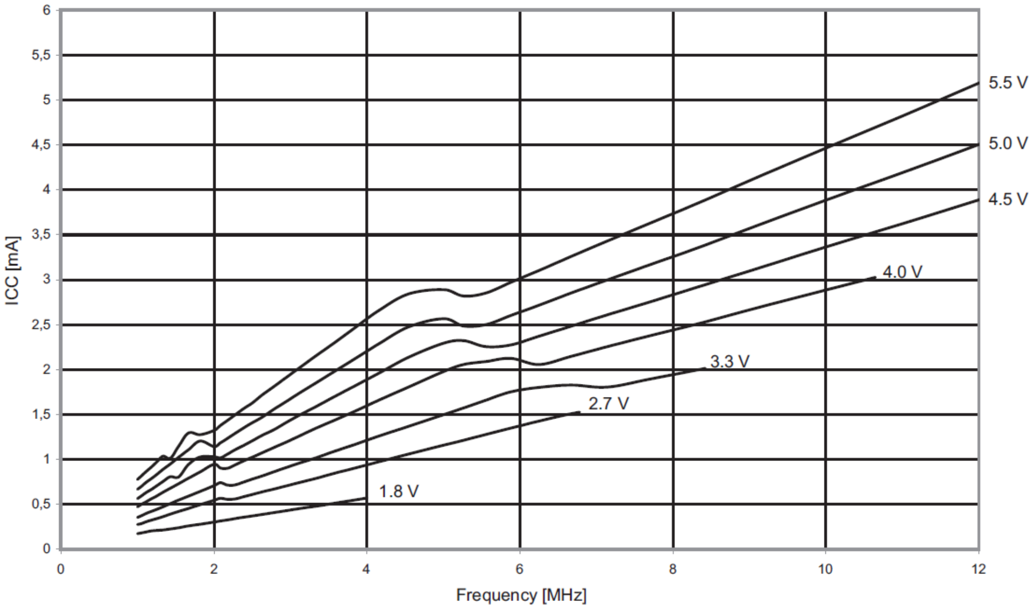


Figure 5.1. *ATtiny20 active supply current versus Frequency (1-12 MHz)[6]*

Looking at the microcontroller datasheet, one may believe that power supply is not critical. The device has a very wide voltage range, and draws a little supply current. But as with all digital circuits, the supply current is an average value. The current is drawn in very short spikes on the clock edges, and if I/O-lines are switching, the spikes will be even higher. [6], [17]

All in all, the filtering of the supply voltage affects the decision of which supply to use and is one of the selection criteria. Two main aspects should be considered when designing the power supply for the microcontroller; ESD protection and noise emission. All the available power supplies already include quality filtering and protection against ESD spikes, but the power lines still need the correct and sufficient decoupling. It is crucial for stable microcontroller behavior, and for minimizing the emitted noise from the device. The decoupling capacitor is placed next to the microcontroller, preventing the formation of a high current loop. [6], [17]

The microcontroller tolerates quite a large fluctuation at the input ports. Output voltage level is fixed to 1.8 V, but the output driver strength has its limitations. The bigger the source current, the lower the output voltage will be. The output high voltage in relation to source current is presented in Figure 6.15. [6]

It is possible to disable the internal pull-up resistor and switch to use an external pull-up resistor. In that case the driver strength would be irrelevant and the output voltage level would be determined by the external pull-up voltage. Electrical characteristics for the I/O-pins are presented in Table 5.1

Table 5.1. *Electrical characteristics for the Microcontroller's I/O ports*

Parameter	Min	Max	Unit	Comments
Input Low-voltage	-0.5	$0.2V_{cc} \approx 0.36$	V	
Input High-voltage Except RESET pin	$0.7V_{cc} \approx 1.26$	$V_{cc}+0.5 \approx 2.3$	V	
Input High-voltage RESET pin	$0.9V_{cc} \approx 1.62$	$V_{cc}+0.5 \approx 2.3$	V	
Output Low-voltage Except RESET pin		≈ 0.45	V	Source current dependent
Output High-voltage Except RESET pin		≈ 1.9	V	Source current dependent

5.2 ECI requirements

The hardware of the ECI, presented in Figure 2.4, consists of the MSIC, the main processor, the ECI-block and the audio block. A connected accessory can also be counted as part of the interface. The known and desired functionalities or characteristics for the circuit at the beginning of the designing process are:

- the mandatory ability for being able to handle the data conversion with all the accessories already in the market
- plug in/out detection
- voltage measurement for detecting and identifying an accessory
- possibility to use the microphone line for the audio and the ECI-communication purposes
- data buffering
- ECI-data voltage level shifting.

Ideally the ECI-block would have been implemented inside the MSIC or the main processor, but that was not possible for various reasons. The audio block, on the contrary, is implemented on the most part inside the MSIC. The MSIC includes features for the plug detection mechanism and it is also capable of doing the accessory detection with the help of its ADC and internal comparators.

In addition to the needed functionalities, the accessory detection protocol, I^2C-protocol, ECI-protocol and the audio block also need to function within the predetermined boundaries. The boundaries are mainly electrical specifications. The electrical specifications concerning the electrical designing of the ECI are specified in the ECI specification documents or in other Nokia specific documents.

The general electrical characteristics, i.e. the requirements given in ECI specification documents, concerning the designing of the electrical circuitry are presented in Table 5.2. The table also includes the requirements set by the product-specific components, which are the MSIC and the main processor. Component-specific characteristics are presented in their own chapters.

Table 5.2. *General electrical characteristics of the ECI*

HSMIC Audio input					
Function	**Min**	**Typ.**	**Max**	**Unit**	**Comment**
MICBIAS	2,00	2,1	2,25	V	
MICBIAS current capability	1,2	-	-	mA	
Vp	2.65	2.775	2.85	V	used during the accessory detection
HSMIC path resistance	-	-	2,4	kΩ	Including resistor R_{detect} and any other EMC-resistances in microphone path
ECI-controller data bus capacitance	-	-	2,5	nF	
AV-connector ECI-data input/output					
Output High-voltage	2.4	-	2.8	-	Open-drain
Output Low-voltage	0	-	0.42	V	≤ 2 mA
Input High-voltage	1.7	-	2.6	V	
Input Low-voltage	0	-	0.7	V	

The tolerances of the power supplies have to be considered at all cases, when determining the characteristics of the input and the output ports of the different components. The typical value of the supply voltage is not very useful in design as the worst-case scenario must always be taken into consideration. All the minimum and maximum values of the logic voltage levels presented in Table 5.3 are taken into consideration when determining the values or properties of different components.

Table 5.3. *Tolerances of the supply voltages*

Schematic power net name	**V_{out} min [V]**	**V_{out} typ [V]**	**V_{out} max [V]**	**I_{out} max [mA]**	**Notes**
VAUX2	2.375	2.5	2.625	200	= Vcc_SW
VIO_1V8	1.71	1.836	1.906	800	= Vcc_ECI
V_2P85_EMMC2	2.708	2.85	2.993	200	
VTG		5.0			

5.3 EMC problems

The mobile phone environment sets some special demands for EMC designing. There are several antennas designed to work in multiple frequency bandwidths. The frequency bandwidths that have the highest transmit power in the phone in question are the GSM850- and GSM900-band, which transmit in frequency bands of 824.0…849.0 MHz and 880.0…914.8 MHz respectively. The transmit power of the GSM-bands is two times higher than in any other used frequency bandwidths. DCS1800- and PCS1900-bands also have high transmit power. The frequency bands for DCS1800- and PCS1900-band are 1710.2…1784.8 MHz and 1850.0…1910.0 MHz.

The wire of a connected accessory behaves like a receiving antenna. Some coupling effect may occur between the transmitted signal and the wire. The signal couples to the connected wire causing interference to the signal. The GSM carrier wave has the highest transmission power, thus it is the most probable cause of interference to the signals in the wire. The lower the energy of the transmission, the more insignificant the coupling effect is and the lesser the impact of interference will be. [11], [17]

The ECI and the audio interface operate side by side partially using the same AV connector ports, which causes some extra challenges for the electrical designing. The requirements of the ECI-data line and the microphone audio line differ significantly; the microphone audio line is very sensitive to all interferences on the frequencies of the hearing range, but the ECI-data, on the contrary, is not as sensitive as the microphone input. Yet, some ESD, EMC and EMI protection must be implemented as near the AV controller as possible. For example, protection against ESD-spikes is a necessity to have.

A capacitor is potentially the easiest and cheapest way to solve many EMC problems, which implies that the microphone line needs much capacitance to ground but the big capacitance does not work for the ECI-data as the signal rise times would increase. The frequency of the microphone audio signal is relatively low and the low value bypass capacitors filter only very high frequencies. The higher the filtered frequency, the smaller the bypass capacitor. The microphone audio line needs several bypass capacitors of different capacitance values in parallel, because there is also a wide frequency bandwidth to be filtered which cannot be filtered using low value capacitors. Each parallel bypass capacitor of different capacitance values will respond better to different frequencies. [12, p. 138], [9, p. 297-298]

According to Table 5.2, the maximum ECI-data bus capacitance is 2.5 nF, which is nowhere near enough for microphone use. This means that the isolating switch must be placed in a way that most of the bus capacitance is isolated from the signal path during the ECI-communication. Also the protection against ESD spikes and the filtering against interference have to be implemented by using some other method than increasing the bus capacitance.

6 COMMUNICATION CIRCUITRY

The purpose of this chapter is to present the designed solution for the original problem and explain the reasoning. The accessory detection is explained in detail with the exception of the AHJ-plug detection, because it was dropped out of the content of the phone terminal before the design was even ready. This chapter, however, includes a draft version of the schematic having the required electronics designed in order to make the AHJ-plug detection possible.

The final version of the designed schematic without the components needed for the AHJ support can be found in Appendix A. The schematic version, which includes the support for AHJ, can be found in Appendix B. The section concentrating on the AHJ is not covered as thoroughly as the other parts of the schematic, due to the cancellation of the support for it. Testing and verifying the AHJ-accessories are impossible to perform even if the electrical design could be implemented to a circuit board, because of the lack of software support. All the explanations of the electrical circuitry and the presented simplified schematic refer to the schematic version in Appendix A, which does not have support for AHJ. The schematic including the AHJ support is covered in its own chapter.

The basics of the EMC and ESD components directly on the ECI-data path are covered. Other EMC components inside the AV interface, i.e. the EMC components not directly affecting the functionality of ECI, are illustrated as “black boxes” instead of being left out of Appendix A. The “black boxes” are presented in order for the reader to understand the possible effect they may have from a design perspective.

6.1 Common parts of the ECI and the audio interface

The common parts for the audio and ECI of the schematic are presented in Figure 6.1. The microphone line is called the HSMIC-line between the AV connector and the EMI filter and is in direct contact with the MSIC through the switch N4060 and the EMI filter N4061. Voltage measurements for the accessory detection purposes are done based on the voltage levels on the HSMIC-line. The microphone bias voltage is provided through the MICBIAS-line and the detection voltage is measured from the ACCDET-line.

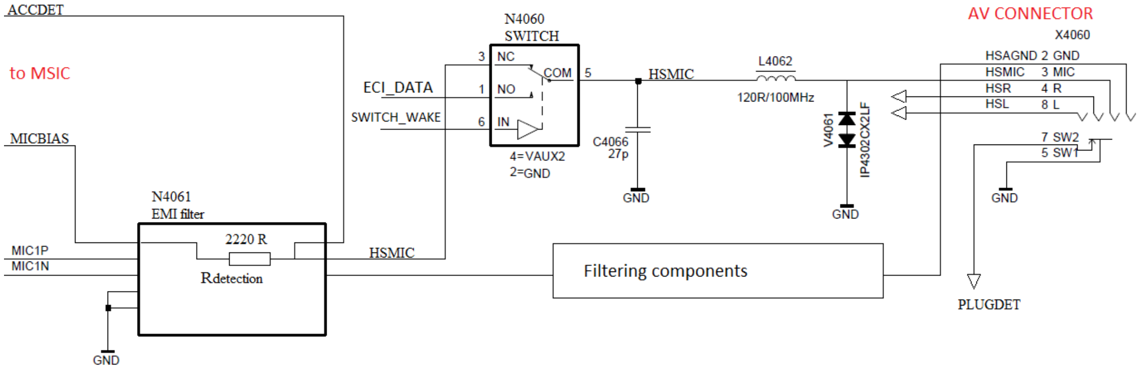


Figure 6.1. *The common components for the ECI and Audio interface*

Most of the noise cancellation against high frequency interference for the microphone audio is done by the EMI filter N4061. The filter is an IC and has many integrated components, but only the components and the routed signal lines affecting the functionality of the ECI are presented. The resistor $R_{detection}$ inside the EMI filter is relevant for the accessory detection and is disconnected from the ECI during the ECI-communication. The microphone bias voltage, used to power accessories, is also provided through the resistor $R_{detection}$.

To guarantee large enough supply voltage to an ECI-accessory during the boot-up, the pin path resistance from microphone bias voltage output through the HSMIC-line to the AV connector can be 2,4 kΩ at maximum. The value includes the bias resistor $R_{detection}$ and any other resistances in the microphone path.

During the ECI-communication the MSIC microphone input and MICBIAS must be disconnected from the HSMIC-line. The switch N4060 isolates the HSMIC-line for the audio or the ECI-data use. The microcontroller D4060 controls the switch with the help of the SWITCH_WAKE-line. Most of the EMC filtering and some ESD filtering is done after the switch to ensure the correct operations in the ECI block. Four filtering components L4062, C4066, V4060 and V4061 are placed between the AV connector and the isolating switch mainly for protection against ESD spikes and high frequency filtering.

6.1.1 EMC & ESD filtering

The surface mount ferrite bead L4062 performs the function of removing RF energy that exists within the HSMIC-line. Ferrite beads are used as high frequency resistors also known as attenuators that allow low frequency signals to pass while absorbing the RF energy and dissipating that energy in the form of heat. [10], [11, p. 218-219]

Since the impedance of the ferrite bead is essentially resistive to high frequency circuits, the problem of resonance experienced by other possible filtering choices like capacitors and inductors is eliminated. The ferrite bead presents minimal series impedance to the lower frequency signals or direct currents of the circuit. [10], [11, p. 218-219]

The main purpose of the ferrite bead L4062 is to block the high-frequency interferences from the HSMIC-line. The ferrite bead L4062 was selected, because it is very effective in eliminating the coupled high-frequency interferences from the signal line and has negligible effect to low-frequency- or DC-signals. The maximum DC resistance of the selected ferrite bead is 100 mΩ. The impedance characteristics of the ferrite bead are shown in Figure 6.2. The figure clearly proves that all the coupled interferences caused by the phone itself or the environment will be eliminated or attenuated effectively from the HSMIC-line using the selected ferrite bead. [11, p. 218-219]

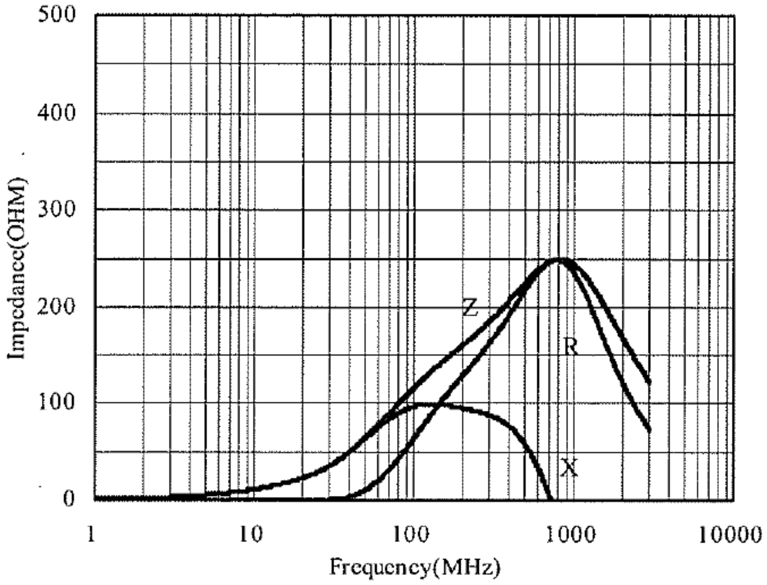


Figure 6.2. *Impedance characteristics of the ferrite bead [20]*

A transient voltage suppression diode (TVS) D1 protects the circuit from voltage spikes induced on AV-connector. TVS is basically a bi-directional diode, which is designed to suppress all voltages above its breakdown voltage. The TVS D1 provides protection to downstream components from ESD voltages as high as ±15 V. The breakdown voltage of the TVS is about 14 V and the maximum capacitance is 10 pF. In the circuit board layout the device is placed as close to the AV connector as possible in order to block all the ESD-spikes before they have made any damage to the circuit. [11], [22]

By placing the capacitor C4066 to circuit at least some high frequency filtering may be achieved. The value of the capacitor is so insignificant that even the bus capacitance of the ECI_DATA-line has a bigger capacitance value. The value cannot be larger, because a larger value would lengthen the ECI-data rise time. Extra capacitance would also increase the probability of causing the rise time to break the limits and ultimately lead to a malfunction of the ECI-communication. The value is not very important during the schematic design phase because a place for a capacitor will be reserved from the layout and the value of the capacitor may be altered at a later phase of the design. More accurate EMC studies will be made after a layout design of the final product is complete and the value of the capacitor can be determined more accurately. [2], [7, p. 20]

6.1.2 HSMIC-line switched to data transfer

During the ECI communication, the MSIC Microphone input and the MICBIAS output must be disconnected from the HSMIC-line. HSMIC-signal is routed to the microcontroller’s data input/output using a toggle switch N4060, to isolate the signal from the MSIC microphone input. The Function table of the switch N4060 is presented in Table 6.1 and the logic diagram in Figure 6.3.

Table 6.1. *Function table for the switch N4060*

CONTROL INPUT (SWITCH_WAKE)	ON CHANNEL
Low	NC/Mic2p
High	NO/ECI_DATA

The ECI-data path and the microphone line must be isolated from each other, as the microphone line needs enough capacitance for EMC-protection, and the ECI-communication cannot tolerate this big capacitance, because it slows down the rising edges of the data signal. Therefore, as presented in the schematic found in Appendix A, most of the EMC protection is implemented after the switch, so that it does not have an impact on the signal integrity of the ECI-data.

The most critical parameter in selecting the switch is the ON-state resistance. The accessory detection is based on the resistance measurement. The switch is in series in the HSMIC-line, so the ON-state resistance must be small enough so that the path resistance stays within the limits. Another important characteristic in selecting the switch is current consumption. The switch must also be compatible with the 1.8 V control signal.

Texas Instrument’s TS5A6442 switch was selected as its ON-state resistance is no more than 0.75 Ω, which is virtually meaningless. Turn-on or turn-off times are not important, because the delays in SWITCH_WAKE-signal are quite long and they are also adjustable. Even the slowest switch in the market would probably be fast enough. ECI-data can tolerate relatively high bus capacitances as the data is being transferred with a slow data rate. The ON-state capacitance of the switch is 116 pF at maximum, so it does not have any significant impact to the ECI-data signal integrity. According to Table 5.2 the maximum bus capacitance for the ECI-data is 2500 pF. [34]

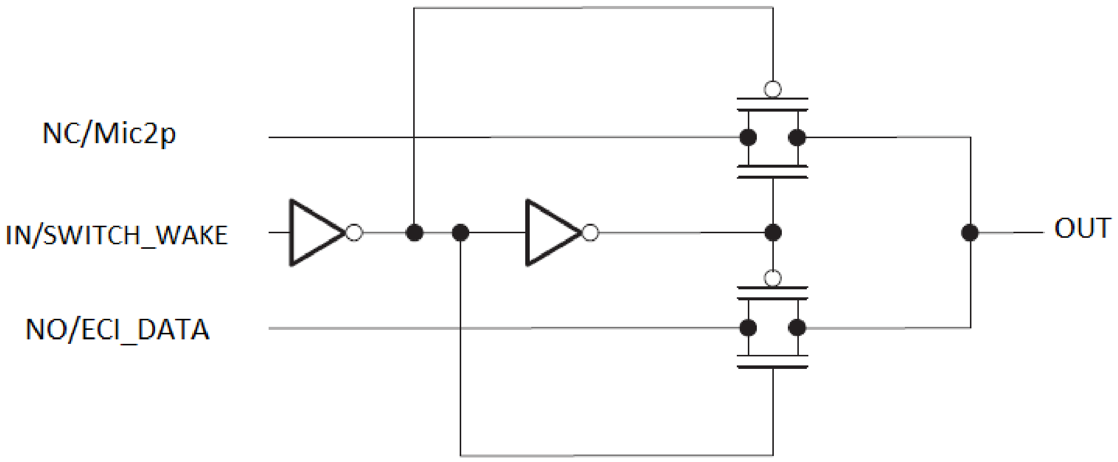


Figure 6.3. *Positive logic diagram for the switch N4060 [33]*

As there are many cheap and small switches available, the final choice was made mainly on the basis of the ON-state resistance. Other affecting factors were availability, size, current consumption and the price. The decision to use switches manufactured by Texas Instruments was mainly based on the price and the quality of the components. The use of switches manufactured by other companies, were also studied. [34]

6.1.3 Powering the accessory

Most ECI-accessories do not have their own power supply and need a small amount of current from the phone terminal to operate. The microphone line must be kept active when an accessory is connected. During the accessory identification and all other cases when the audio is active, the accessory is powered from the MICBIAS. According to Table 4.1 the MICBIAS current capability is 1.2 mA.

The ECI-accessory is powered from the MICBIAS until the HSMIC-line is switched from audio block to ECI-block. The pull-up resistor R4063 in the ECI-data path replaces the powering path, and provides a required amount of power for the data transfer. Meanwhile the MICBIAS-voltage inside the MSIC may stay active as the ECI communication occurs in short bursts. The microphone in the accessory is switched off during the data transfer to get the current consumption down and logic levels favorable.

ECI accessory includes a supply storage capacitor C1, presented in Figure 6.4, that keeps the supply voltage for the ECI-ASIC above 1.2 V during a data burst while it is gradually discharged during the data transfer. The 1.2 V is the minimum supply voltage for the ECI-ASIC. The matter concerning the voltage levels of the ECI-data is discussed more precisely when the measurement results are analyzed.

The capacitor is also charged during the ECI-communication through the pull-up resistor R4063 if the open-drain transistor of the accessory is not pulling the line to ground. Without the charging during the ECI-communication, the voltage over the capacitor C1 may fall too much to cause the ECI-communication to fail. Charging during the ECI-communication helps the capacitor to sustain high enough supply voltage in order to the ECI_ASIC to stay operational.

After an ECI-communication, the charge level of the capacitor might have substantially lowered. Right at the moment when the HSMIC-line is switched back to audio use, the capacitor is once again being charged from the MICBIAS. The capacitor is charged from the MICBIAS during the detection phase and after the detection phase at all other times when the HSMIC-line is switched to audio use.

The MICBIAS must be active when the HSMIC-line is switched to audio use, even if the microphone is not in use. The MICBIAS is then used to charge the capacitor C1. For example, during the listening of the music, the microphone is not in use, but the control buttons may be used.

The capacitor C1 and the serial resistance on the powering line, i.e. along the path between MICBIAS and the capacitor, form an RC circuit, which affects the rise time of the HSMIC-voltage. The capacitor must be big enough to allow the ECI-ASIC to keep working during the ECI-communication, but it also must be able to be charged to a sufficient voltage level fast enough in case of a series of frequent button presses. The sufficient value for the capacitor is usually small as the operating time of the ECI-data sending is not long, though even a big capacitor charges quite fast and slows down only a bit the voltage rise time on the HSMIC-line. The delays during the detection phase are well-defined, which sets the boundaries for defining the maximum size for the capacitor. [7, p. 23-24]

Ideally the MICBIAS or the pull-up supply VAUX2 powers the ECI-ASIC and charges the capacitor C1 when the switch VddSw is in a conducting state. A possibility of an error situation where the MICBIAS is not active or not routed to accessory and the switch is open exists. In that case, the resistor R_a prevents the capacitor from discharging as the ECI-ASIC provides a discharge route of much lower resistance.

The main purpose of the resistor R_a is to be part of the voltage measurement circuit during the accessory detection phase and to limit the amount of current drawn by the capacitor C1 during the ECI-communication. During the data transfer LOW state between the microcontroller and the ECI-ASIC, the resistor is floating and the capacitor is disconnected from the ECI-bus as the switch VddSw is open. At all other times, the switch VddSw is in a conductive state.

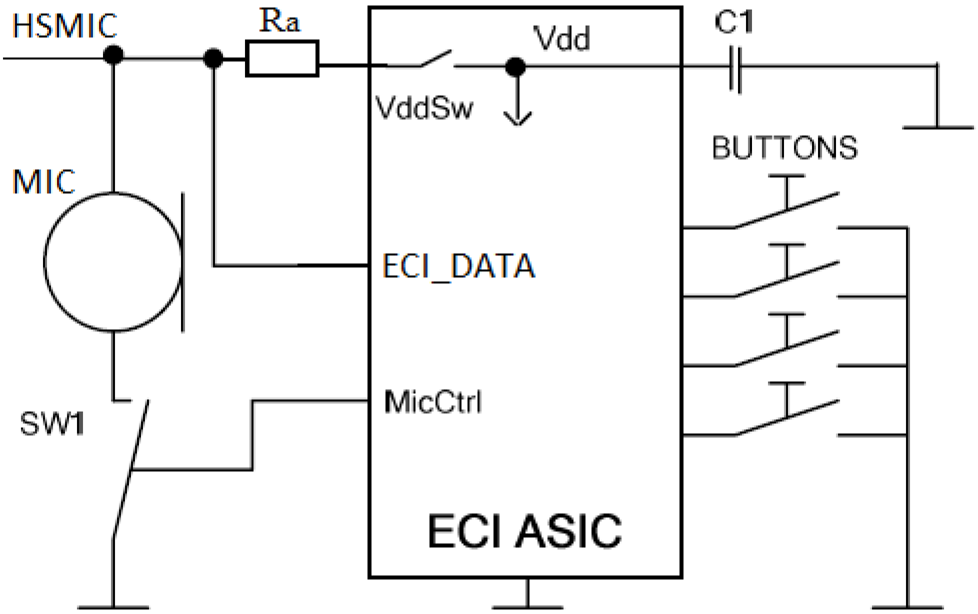


Figure 6.4. *Block diagram of an ECI- accessory*

The charged capacitor and the load, i.e. ECI-ASIC, form an RC-circuit. The size of the capacitor C1 can be determined using the Equation 6.1, where Vc(t) is the voltage over the capacitor C1 at a certain time t, V_0 is the voltage over a fully charged capacitor, R_{asic} is the resistance caused by the ECI-ASIC and t is the discharge time. The selection of the capacitor is done on the basis of the needed power to the ECI-ASIC. The needed power is determined on the basis of the operating time of the ECI-communication and the amount of current drawn by the ECI-ASIC.

$$\begin{aligned} Vc(t) &= Vo * e^{\frac{-t}{Rasic*C1}} \\ \Rightarrow t &= ln\frac{Vc}{Vo} * (-Rasic * C1) \\ \Rightarrow t &= \ln\frac{2.1\,V}{1.2\,V} * (-Rasic * C1) \qquad (6.1) \end{aligned}$$

The amount of data sent by the accessory is relatively short, so the capacitor does not need to provide the power for long and the discharge can occur quickly. With the help of the equation, the capacitor discharge time can be determined. ECI-ASIC needs a supply voltage of 1.2 V at minimum and the voltage over the fully charged capacitor is about 2.1 V. The capacitor must be fully loaded before the ECI-communication can be started. The discharge curve of the capacitor is presented in Figure 6.5.

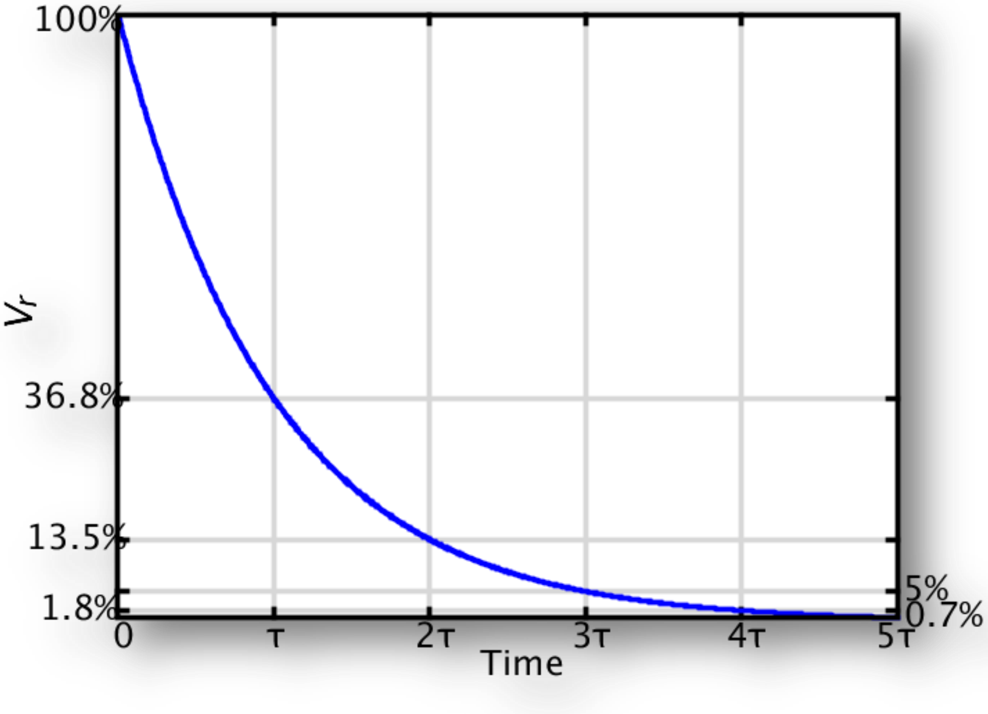


Figure 6.5. *Capacitor's discharging curve with percentages of initial value [27]*

Typical ECI-accessory implementations include a power supply buffer capacitor, but some ECI-accessories may include their own power supply. For example, a car kit accessory could be supplied from the car battery. The MSIC has a needed functionality to recognize an external power source during the accessory detection and is capable of adjusting to a situation by deactivating the MICBIAS in order to avoid unnecessary power consumption.

6.1.4 Accessory detection & identification

Accessory identification flow begins with a plug detection, which is done by the MSIC. Plug detection is based on a low-to-high voltage transition in the PLUGDET-line. With no accessory plugged in, the line is pulled down. Plug insertion disconnects the PLUGDET-signal from the other terminal connected to ground inside the jack and the PLUGDET-line is now pulled up to 1.8 V with 100 kΩ resistor. 1.8 V is a required voltage level defined by the MSIC characteristics. Accessory removal is also detected from the PLUGDET-signal transition when the plug is removed. The falling edge of the PLUGDET-signal acts as a mark of removal action.

After a successful plug insertion, the MSIC activates the HSMIC-line and sets the MICBIAS-voltage to 2.1 volts. The HSMIC-voltage is lower than the MICBIAS-voltage, due to the serial resistor $R_{detection}$ and the resistor $R_{accessory}$ inside the accessory if the connected accessory is not an open cable. The identification is done based on different impedance states of the HSMIC-line and further actions are done based on a voltage level on the HSMIC-line.

The voltage level of the HSMIC-line is defined using Equation 6.2. Appendix A presents the entire electrical circuitry related to the ECI communication, but does not include the components inside the accessory. Figure 6.6 (a) presents the voltage measurement part of the entire ECI schematic including the accessory. To clarify the actual voltage divider, only the components that have notable effect to the HSMIC-voltage division are presented. The HSMIC-line is in connection with the ACCDET-line, which is in the same potential as the HSMIC-line. The actual voltage measurement is being done from the ACCDET-line. The series resistor $R_{detection}$ is inside the phone terminal and $R_{accessory}$ is a series resistor inside the accessory.

$$V_{HSMIC} = \frac{Raccessory}{Rdetection + Raccessory} \times V_{MICBIAS} = \frac{Raccessory}{2220\ \Omega + Raccessory} \times 2{,}1\ V \qquad (6.2)$$

During the accessory identification, not only the series resistor $R_{accessory,}$ but all other serial resistances, affect the measured voltage. Tolerance levels for the different detection thresholds are quite loose, but even so, some designing guidelines must still be followed. According to Table 5.2 the HSMIC-path resistance in the phone terminal can be 2.4 kΩ at maximum.

On the phone design point of view, the actual size of the resistor $R_{accessory}$ is irrelevant and unknown and only the total path resistance between the measuring branch and the ground point of the accessory is relevant. The series resistor $R_{accessory}$ in Equation 6.2 actually includes all the resistance, e.g. the load resistance caused by the ECI-ASIC and the actual series resistor R_a presented in Figure 6.4, along the voltage divider path.

Simplified schematics of the supported detection scenarios using the MICBIAS are presented in Figure 6.6. Figure 6.6 (a) illustrates a normal ECI-accessory or the video cable detection situation. The complete structure of an ECI-accessory is presented more precisely in Figure 6.4. The detection scenario in Figure 6.6 (b) illustrates a short circuit between the HSMIC- and GND-signals, i.e. headphones or an AHJ-plug.

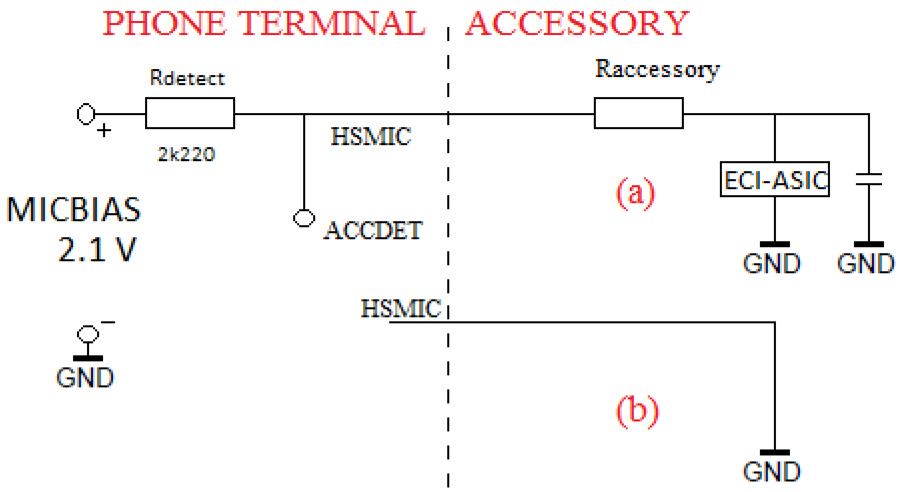


Figure 6.6. *Schematics of the two possible detection circuit scenarios when the MICBIAS-voltage is activated and the comparator COMP1 is used for the voltage measurement*

The MSIC starts the accessory identification to differentiate between the possible accessories. The circuit arrangement inside the MSIC, which is relevant during the accessory detection phase, is presented in Figure 6.7. The figure, however, does not include audio outputs, because they are used only for detecting a short circuit. The first step in identification flow is to identify the unsupported accessories. There are, for example, some headsets using differential signaling and having a different pin order than the AV connector and therefore are not be supported. The detection for unsupported accessories is done by:

- first enabling the microphone bias voltage
- then measuring the voltage of the ACCDET-line with the ADC
- then enabling the headset amplifier
- and finally measuring the ACCDET-line voltage again with the ADC.

If the two voltage level measurements differ, the connected plug has caused a short circuit between the microphone line and the audio output. Therefore, the accessory is not supported, the detection is terminated and all lines are de-activated. The short circuit detection is the only case during the detection flow or the ECI-communication when the audio outputs are used for identification purposes.

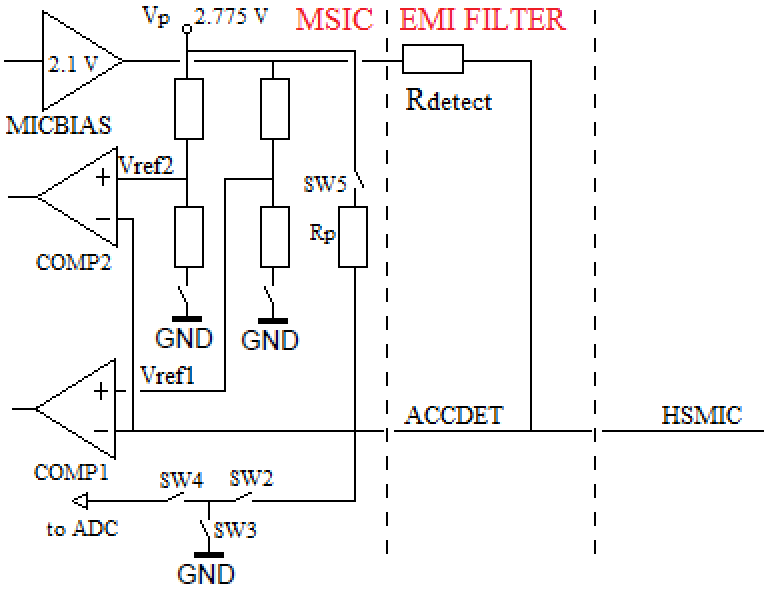


Figure 6.7. *The MSIC basic circuit arrangement used for accessory detection*

The MSIC has two internal comparators, which are used to differentiate the accessories. Comparator COMP1 has a reference voltage of 0.5 volts. The reference voltage for the comparator COMP2 is 1.85 volts. The main purpose of the comparators is to differentiate the ECI-accessories from the non-ECI-accessories and also the non-ECI-accessories from each other. Electrical characteristics, i.e. the detection threshold voltage levels, of the MSIC comparators are presented in Table 6.2.

Table 6.2. *Detection threshold voltages for different non-ECI-accessories*

Detection Threshold [V]					**Detection threshold [Ω]**		
Parameter	**Min**	**Typ.**	**Max**		**$R_{accessory}$ (typical)**	**Device**	**Comment**
MSIC (Vref1)	0.4	0.5	0.6	<	694 Ω	Headphones (grounded) Video cable (75 Ω)	MICBIAS active
MSIC (Vref2)	1.75	1.85	1.95	>	30 kΩ	Open cable	MICBIAS deactivated

If the measured voltage level of the ACCDET-line is lower than 0.6 V, the MSIC activates its internal ADC again by turning the switches SW2 and SW4. This phase occurs only if the detection flow was not already cancelled during the active time of the headset amplifier. The only possible accessory options, which may have a low enough serial resistance to generate such a low voltage level, are headphones or a video cable.

Regardless, neither of them are an ECI-accessory and no ECI-communication are required.

If the measured voltage is higher than 0.6 V, the main processor tells the microcontroller to enable the ECI-data path and try the ECI-communication, although a delay is needed in order to reach sufficient supply level on ECI-accessory's storage capacitor for matching logic levels with the master side. The SWITCH_WAKE-signal sets the toggle switch N4060 that routes the HSMIC-signal to the microcontroller. Thus the SWITCH_WAKE-signal must be kept active as long as the ECI-data communication continues, and it must be deactivated as soon as the communication is done in order to have the channel available for audio again.

The SWITCH_WAKE-line is also routed to the main processor. That is done to provide the information of the switch turn to main processor. In this design the information is not needed, but in future the option may be useful. The main processor I/O-pin that it is connected is in three-state mode.

With the enabled SWITCH_WAKE-signal, the microcontroller starts the ECI-identification protocol. The microcontroller sends a response request to the accessory and in order to start the ECI-communication protocol, it needs to get a valid ECI-identification response from the accessory. After the microcontroller has received a valid response, the accessory will be verified as an ECI-accessory. Based on the ECI-communication, the exact type of the accessory is also known at this point. As a result, the ECI-accessories are not identified based on the measured voltage level but rather based on the success of the ECI-communication.

The SWITCH_WAKE-signal stays active for only a short period of time. If the microcontroller does not get a valid response in time, the microcontroller switches the HSMIC-line back to the audio use and tells the MSIC via the main processor that the connected accessory was not ECI-accessory and the MSIC must continue accessory identification. After a successful ECI communication, the microcontroller signals the main processor that the data communication has been finished, which responds, and the microcontroller automatically goes back to the power-down mode. The MICBIAS-voltage may be activated or deactivated during the ECI-communication, but indubitably it must be disconnected from the ECI-data path.

The detection mechanism for the ECI-accessories with an external power source is exactly the same as the ECI-accessories powered by the MICBIAS. Such an ECI-accessory is capable of providing the type-information to the phone terminal in the same manner as all the other ECI-accessories. From the basis of the type-information, the phone terminal knows to keep the MICBIAS continuously deactivated, which lowers the current consumption in the phone terminal. The ECI-communication protocol occurs just like it does with the ECI-accessories not having their own power source.

Open cables, headsets or non-ECI-accessories, which have an external power source, do not send a response to identification attempt sent by the microcontroller. In case of a failed identification, the next steps for the MSIC are:

- keep the MICBIAS-voltage deactivated or deactivate it, if it already was not
- activate the comparator COMP2
- activate the pull-up resistor R_p by turning the switch SW5.

It is important to notice that the MICBIAS-voltage is now deactivated and the voltage divider circuit differs from the one used with the comparator COMP1. Figure 6.8 (a) illustrates an open cable situation and Figure 6.8 (b) illustrates a situation, where the load is a microphone or a headset. Figure 6.8 (c) illustrates a situation, where an accessory is having its own power source. It is not necessary to know the exact structure of accessories. The schematics are not meant to be specific or even right for that matter, but just intended to give a simplified model of the detection event.

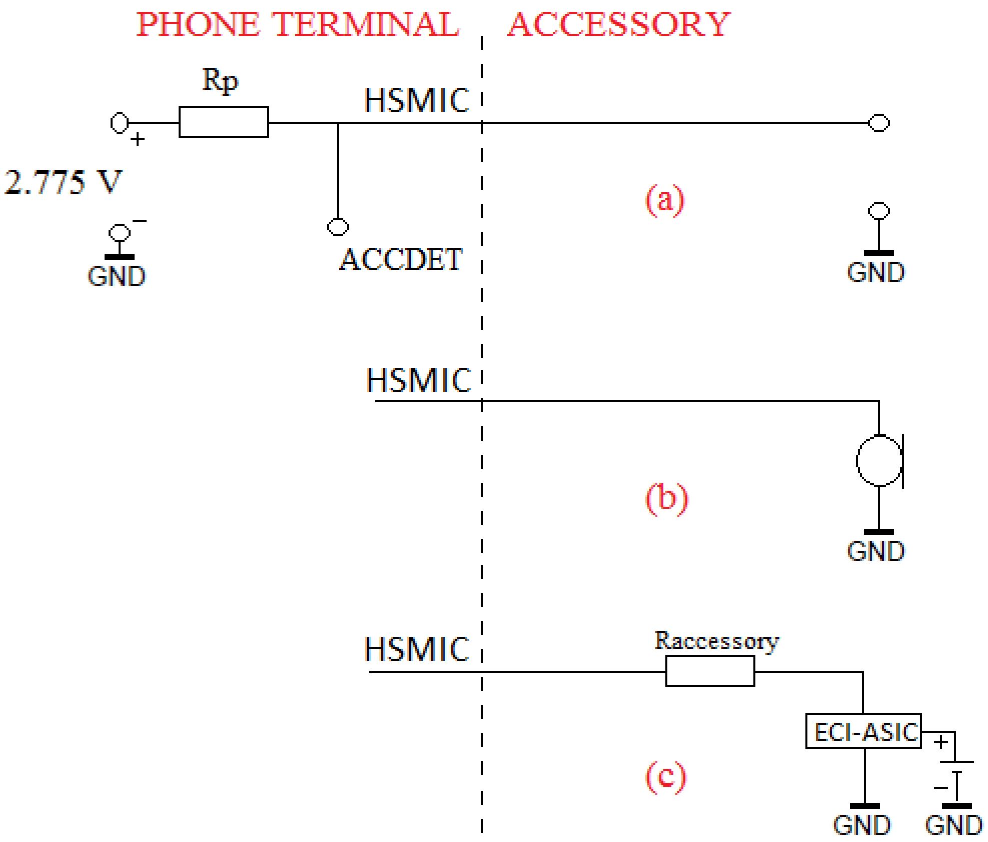


Figure 6.8. *Schematics of the three possible detection circuit scenarios when the MICBIAS-voltage is deactivated and comparator COMP2 used for the voltage measurement*

The reference voltage 1.85 V of COMP2 separates the open cable from other accessories. An open cable is equivalent to an infinite resistance, in which case the measured voltage does not ideally drop at all. A situation where the measured ACCDET-voltage is under 1.85 V can only occur after a previously unsuccessful ECI-communication attempt. The voltage level of the HSMIC-line is this time defined using Equation 6.3. A load over about 30 kΩ causes the comparator COMP2 to change its state.

$$V_{HSMIC} = \frac{Raccessory}{Rp+Raccessory} \times V_p = \frac{Raccessory}{15\ k\Omega+Raccessory} \times 2{,}775\ V \qquad (6.3)$$

The load caused by a microphone is reactive and relatively small, so the voltage measured or compared with the COMP1 was over 0.6 V. The measurement result with the COMP2 would also be under 1.85 V, which indicates that the accessory could be an ECI-accessory. At this point the ECI-communication has already been tried and failed, so the only thing to do is to analyze the nature of the accessory further. If the comparison done with the COMP2 produces a result under 1.85 V threshold level:

- the HSMIC-line is once again switched to audio use
- the MICBIAS stays grounded
- the switch SW5 is turned in conducting state
- and the ACCDET-voltage is measured with the ADC.

There is only one supported detection scenario which can cause the positive detection result at this point; a headset or just a microphone. In the event of a valid measurement result, the MSIC activates the short and long button press detection and the detection stays active for as long as the accessory is connected to the phone terminal. The MICBIAS-voltage stays also activated.

The short and long button press detection, i.e. answer- and end-call button detection, can be made with comparator COMP1 presented in Figure 6.7. When the MICBIAS-voltage is activated, a button press draws the ACCDET-line below the COMP1 threshold level and this generates an interrupt. The button press detection is based on the time difference measurement of the falling and rising edge of the ACCDET-line.

If the measurement done with the ADC does not return a valid result, the MSIC turns the switch SW5 to off-state and de-activates the MICBIAS. The accessory detection has led to an unsuccessful end result and the accessory is not supported.

6.2 Waking up the microcontroller

The microcontroller must be woken up from the idle mode before it can handle the ECI-communication. During the detection phase, the MSIC and the main processor communicate with each other, so that the main processor knows when to activate the microcontroller. The ECI-communication always occurs in short bursts, so the microcontroller does not need to stay continuously active. Immediately after the ECI-communication, the microcontroller switches the HSMIC-line back to audio use and goes to the idle mode. Figure 6.9 presents the schematic part of the ECI-block, which is needed for waking up the microcontroller from the idle mode.

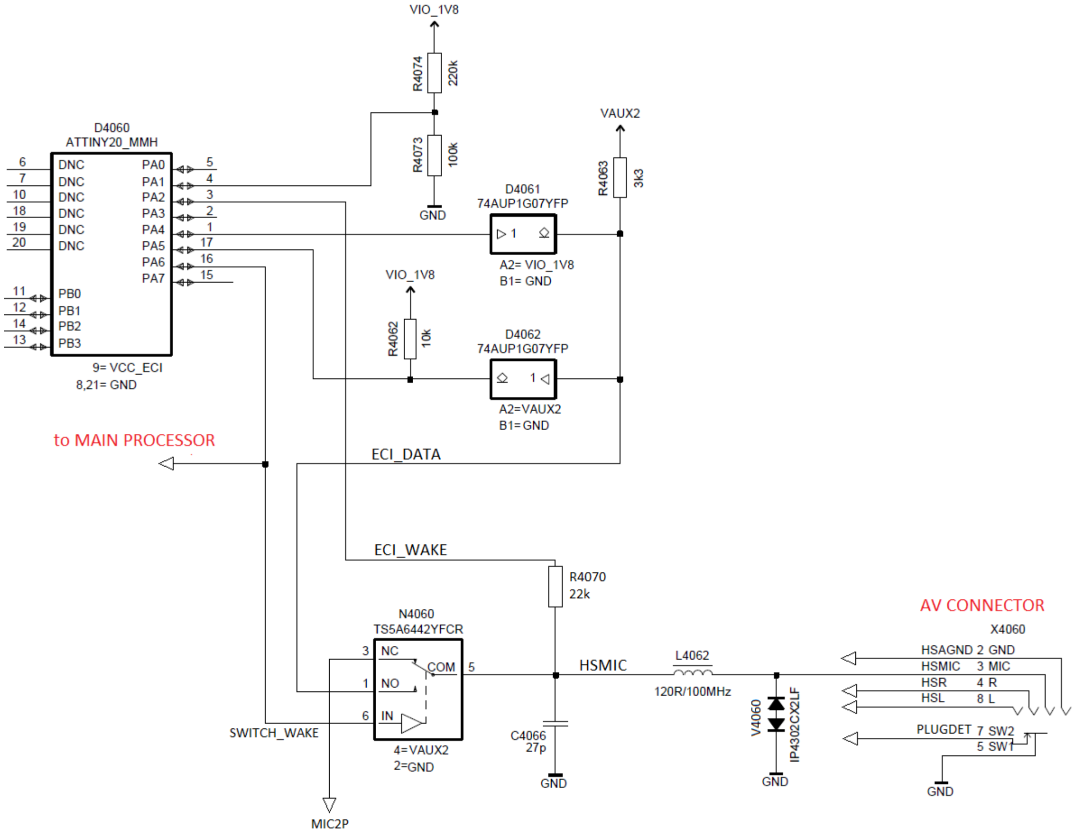


Figure 6.9. *The schematic part needed for waking up the microprocessor*

After the detection phase, the microcontroller may be in the idle mode for a long time, until a user presses a button. The ECI_WAKE-line is routed directly through the resistor R4070 from the HSMIC-line to the comparator input of the microcontroller. The value of the resistor is chosen to be large enough to separate the ECI_WAKE-line from the HSMIC-line during the data communication. On the other the hand, the value is chosen to be small enough to assure the generation of a falling edge when a button is pressed.

The reference voltage for the internal comparator is done with the voltage divider including the resistors R4073 and R4074. The reference level defined by the voltage divider is about 0.6 volts. A button press pulls the HSMIC-line and the ECI_WAKE-line to the ground for a certain period of time. When the voltage level of the ECI_WAKE-line drops below the reference voltage, the microcontroller wakes up from the idle mode. This is, of course, if it already was not in the operational mode. [6]

After activation, the microcontroller immediately enables the SWITCH_WAKE-signal and reads the information of which button is pressed. The reading operation must be done during the button press, because ECI-accessories do not save the information anywhere. If the information was not read in time, the information would be lost.

The microphone audio is biased, so that the voltage level never goes below the reference level of the comparator in microphone use. This is done to assure that the microcontroller stays in the idle mode when the microphone of an ECI-accessory is used, but no buttons are pressed. If buttons are not pressed, the microcontroller is not needed and it can stay in the idle mode.

If the accessory was detected for example, as a non-ECI-accessory with call/end call buttons, during the detection phase, the microcontroller would be in the standby mode as well as if no device is connected to the AV connector. In that case, the falling edge of the ECI_WAKE-signal would not have any effect as the analog comparator inside the microcontroller would be disabled. The grounding of the HSMIC-line wakes up the microcontroller only if the accessory was detected as an ECI-accessory. The first wake-up command or interrupt is always delivered by the main processor via the ECI_RESETn-line during the detection phase.

6.3 Level shifting electronics

Level shifting electronics are not required between the microcontroller and the main processor as the I/O-ports at the both ends, i.e. the microcontroller and the main processor, are using the same logic levels. The ECI-data logical voltage levels have to fulfill the Nokia requirements for the ECI. The output voltage level of the microcontroller is 1.8 volts. According to Table 5.2, the value is inside the limits of the input logic high voltage level of the ECI-accessory. Basically the voltage level shifting is not needed, but there are several things to be taken into consideration in mobile phone environment while making the decision of what voltage levels to use during the ECI-communication.

Several reasons speak on behalf of using a higher voltage level than 1.8 volts. The tolerance level of the 1.8 V supply voltage is ± 5%. Also a situation may occur where:

- the output voltage of the microcontroller might momentarily drop too much
- the voltage level decoding in the input stage of the ECI-ASIC may be off slightly
- the switch N4060 or any other component, that may later have to be placed along the ECI-data route, causes some unwanted voltage drop
- a series resistor may have to be placed to protect against voltage spikes.

The first implementation possibility is to use the same logic levels at both ends. The logic level for the ECI data can be either 2.5 V or 2.1 V, depending on a used bias scenario. According to Table 5.1 the logic level of 2.5 V is not within the specifications of the microcontroller using 1.8 V supply voltage, but 2.1 V meets the required characteristics. Thus, the feasible common voltage level is 2.1 volts.

The simplified electrical communication circuit using 2.1 V logic level is presented in Figure 6.10. All port pins at the microcontroller’s open-drain output have individually selectable internal pull-up resistors with a supply-voltage invariant resistance. The pull-up line inside the microcontroller can be disabled using the switching transistor presented in Figure 6.10 and Figure 6.11, in which case an external pull-up resistor must be used.

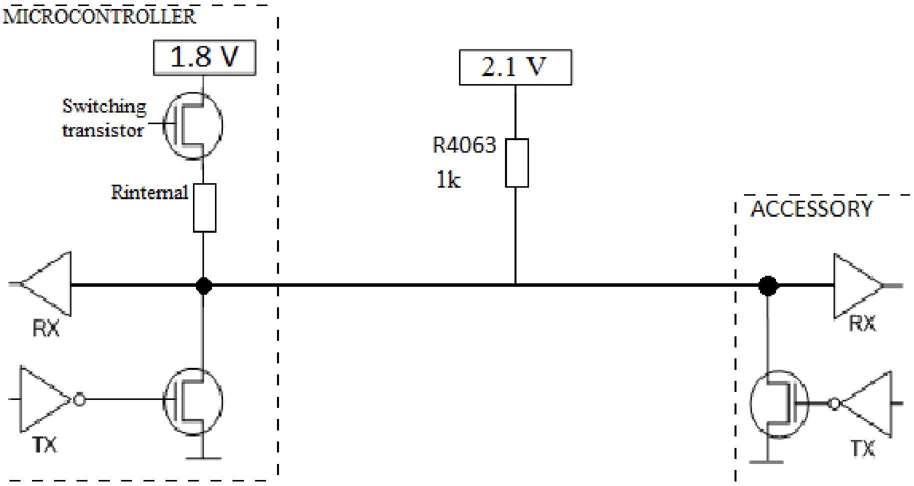


Figure 6.10. *Two different logic levels using the same pull-up voltage*

Most I/O-ports of the integrated-circuit logic devices, have extremely large gate-to-drain and gate-to-source impedances – 100 MΩ or more. Unless bypassed, such high impedances would allow significant electrostatic buildup. Thus, there is usually some form of low-impedance bypass circuitry to exposed external device terminals to prevent damage from electrostatic buildup. For example, diodes are used to implement low-impedance discharge paths for I/O-ports. [12, p. 64]

The difference between the supply voltages of the microcontroller and the data bus may cause some undesired and excessive leakage current through the microcontroller's internal protection diode to ground or supply voltage during the idle or the standby mode of the microcontroller. An equivalent schematic for the microcontroller's I/O-pin is presented in Figure 6.11. [6]

The current may flow if the voltage level of the external pull-up resistor exceeds a diode drop above the supply voltage of the microcontroller. Excessive input current is also a possibility at either end if either supply lags the other supply power turn-on or turn-off, or if one supply fails. The solution to these overcurrent issues is to use devices without input diode clamps to the supply voltage. Because that is not a possibility, some other implementation method must be considered. [12, p. 27-28]

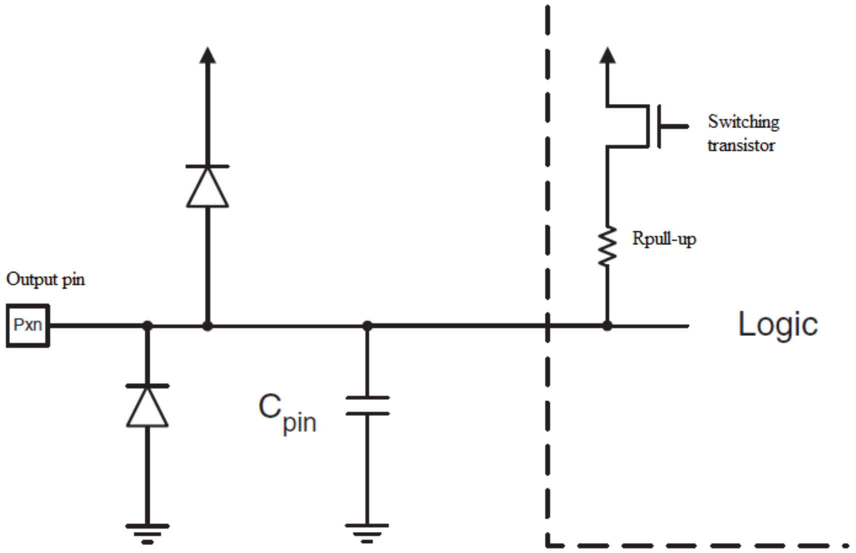


Figure 6.11. *ATtiny20 I/O-pin equivalent schematic [6]*

After the idea of using the same logic voltage levels at the both ends was rejected, the decision of using 2.5 V as the voltage logic level for accessories and 1.8 V for the microcontroller was made. By using the voltage level of 2.5 V the bus is not as sensitive for failure situations caused by the momentary voltage level drops as it would be if the used voltage level was any lower.

Another feasible option would have been to use 2.5 V as a supply voltage to the microcontroller, in which case the voltage high level for the ECI-communication would have been 2.5 volts. Thus, the level shifting would have been implemented to interrupt-line, reset-line and also to I^2C-lines between the main processor and the microcontroller. The option was rejected in the beginning, because of the higher current consumption caused by the higher supply voltage and also because the I^2C-protocol is much more sensitive to distractions than the ECI-protocol.

Ideally the best solution would be to use a microcontroller, which I/O-ports can handle 2.5 V voltage levels while its supply voltage is 1.8 V. The internal pull-up resistor would be disabled and the external pull-up voltage would be 2.5 volts. On the other hand, that was never an option considered, because of the pre-determined choice of using the ATtiny20 microcontroller. In addition, most of the I/O-ports of any microcontroller include the protection diodes, so a component not having those must have been found in order to prevent the leakage current to take place.

By using a bi-directional level shifter presented in Figure 6.12 it is possible to interconnect the two devices using the same bus protocol, with each device having different logic levels. Thus, it is possible to avoid the problems caused by the different supply voltages as the different voltage sections are separated from each other. The solution would also provide protection for the phone terminal against possible voltage spikes happening in the accessory. The level shifter circuit also needs just one wire allowing the interconnection of two different logic levels.

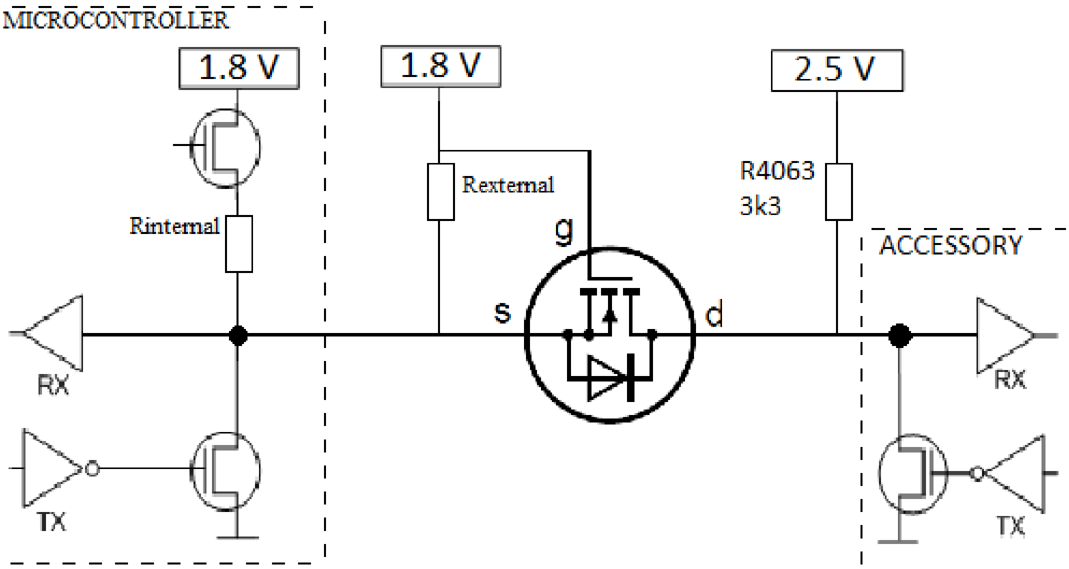


Figure 6.12. *Bi-directional level shifter circuit connecting two different voltage sections in an open-drain bus system*

The level shifter presented in Figure 6.12 consists of one discrete N-channel enhancement MOS-FET. The gate is connected to the lower pull-up voltage, the source to the bus line of the lower pull-up voltage, and the drain to the bus line of the higher pull-up voltage. If no device is pulling down the bus line on either side, the MOS-FET is not conducting and both sections are being pulled up through their individual pull-up resistors. So, the bus lines of both sections are at high-state, but at a different voltage level. [35]

If the microcontroller pulls the line down, the source of the MOS-FET also becomes low, while the gate stays at 1.8 V. The gate-source voltage rises above the threshold and the MOS-FET starts to conduct. The accessory side bus line is then also pulled down via the conducting MOS-FET. So, the bus lines of both sections are at the same low level, although a small insignificant difference may occur, due to the drain-source resistance of the MOS-FET. [35]

If an accessory pulls the bus line low, the integral drain-substrate diode of the MOS-FET begins to conduct and pulls the source voltage of the MOS-FET down enough that the gate-source voltage rises above the threshold and the MOS-FET starts to conduct. The bus line at the microcontroller side is then also pulled down via the conducting MOS-FET. Therefore, the bus lines of both sections are at the same low level. The three states show that the logic voltage levels are transferred in both directions of the bus system, independent of the driving section. [35]

In this case the decision was made to use buffers instead of the solution presented in Figure 6.12. It would also do the task at hand, but due to the several reasons, the FET-solution was rejected; the buffers are cheaper, easier to implement and they are at least as reliable solution as the single MOS-FET implementation. All the reasons for rejecting are heavily related to mass-production. It would, for example, have to be ensured that the noise margin of the FET-based solution would be undoubtedly good enough. [36]

There is always a small voltage over the parasitic drain-substrate diode of the MOSFET and it varies a little between individual components and is also temperature dependent. It affects especially the voltage levels seen in the microcontroller's input along with the other voltage lowering components, e.g. serial resistance of the data path. If the FET-based solution was used, it would somewhat increase the uncertainty of the reliable operation of the ECI-communication. The gate-source threshold is temperature dependent factor and the variations in threshold level should also be studied carefully. [13, p.87-88], [36]

There are no bi-directional buffers capable of doing the level shifting available. Therefore, the microcontroller implementation must include separate input and output ports for the ECI-communication and both lines have to be buffered individually. The communication still occurs on one wire as the accessory has only one I/O-port. The writing of the firmware is also easier if separate input and output ports are used instead of one common port.

The voltage levels have to be adapted for both directions separately and also isolate the microcontroller's input and output signal from each other. Buffers are suitable for the job, although the difference in logic levels cause some brain teasers as they set some conditions for the selection and implementation process. Matters to be taken into account in selecting the buffer:

- the current consumption and the price must be low
- size
- the difference in logic levels
- ECI-data rise and fall times are reasonably low and the selected buffer must tolerate this
- single versus dual
- on-resistance
- used output driver technique.

The electrical specifications between the different open-drain buffers are very much alike. The most limiting factors are the open-drain output and the tolerance for low rise and fall time transitions. The implementation was chosen to be carried out either with two SN74AUP1G07 single buffers or with a one SN74AUP2G07 dual buffer. The electrical specifications for the dual buffer are the same as the specifications for the single buffer; the dual buffer just includes two integrated single buffers. The decision to use buffers manufactured by Texas Instruments was mainly based on the price and the quality of the components. Other beneficial factors were the size and transition rate tolerances. A logic diagram of a buffer with an open-drain output is presented in Figure 6.13. [24], [25], [26]

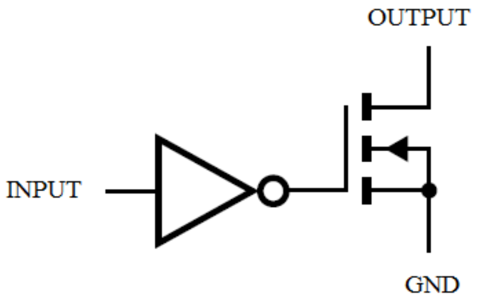


Figure 6.13. *Logic diagram of a buffer with an open-drain output*

The electrical characteristics for the two alternatives are presented in Table 6.3. SN74LVC1G07 is presented for comparison to demonstrate the difference in transition rates between available buffers. There are also two different supply voltage scenarios presented for the single buffer SN74AUP1G07. Corresponding electrical characteristics for the microcontroller are presented in Table 5.1. The input high-voltage level of the microcontroller using 1.8 V supply voltage can vary between 1.26…2.3 V. Figure 6.15 presents the “output high-voltage versus source current”-curve for the microcontroller output if an internal pull-up resistor is used. [24], [25], [26]

Table 6.3. *Electrical characteristics for three different buffers*

	SN74AUP1G07 D4061		**SN74AUP1G07 D4062**		**SN74AUP2G07**		**SN74LVC1G07**		
	Min	**Max**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Unit**
Supply voltage	1.8		2.5		2.5		2.5		V
Input low-voltage	-	0.63	-	0.7	-	0.7	-	0.7	V
Input high-voltage	1.17	-	1.6	-	1.6	-	1.7	-	V
Input transition rise/fall rate	-	200	-	200	-	200	-	20	ns/V

Figure 6.14 presents a basic buffer circuit with an open-collector output. The schematic is obviously not the same as the actual schematic of any of the mentioned buffers, but gives the idea of the design of the buffers in general. Because of the possible leakage current through the diode D1, presented in Figure 6.14, the supply voltage of the buffer must be at least as high as the value of the pull-up voltage affecting to buffer’s input. The buffer may also have internal protection diodes, which must also

be taken into account. The first option is to use a dual buffer using a 2.5 V as a supply voltage. The buffer needs external pull-up resistors to work. The pull-up voltage toward the accessory is 2.5 V and towards the microcontroller 1.8 volts.

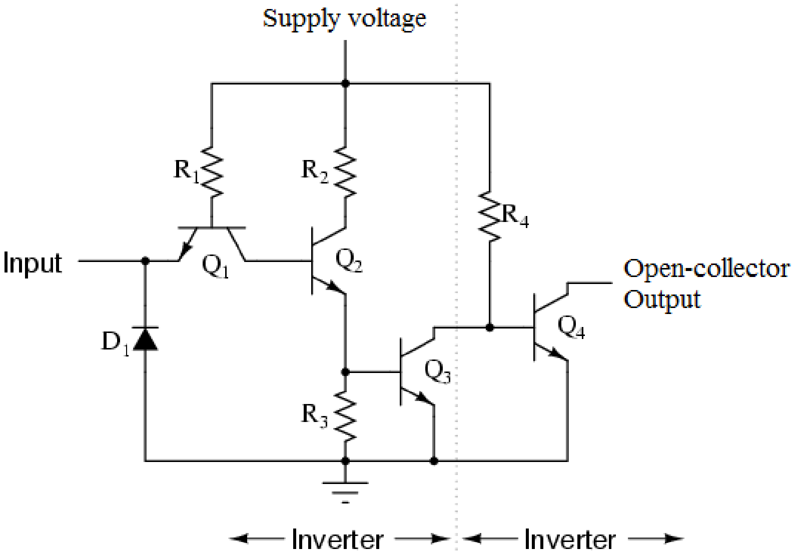


Figure 6.14. *Buffer circuit with an open-collector output [21]*

The electrical characteristics in Table 6.3 appear favorable for using a dual buffer, but the derating factor, i.e. margins of safety operation according to Nokia specifications, is not good enough. The input high voltage level of the buffer must be 1.6 V at minimum and the output voltage of the microcontroller presented in Figure 6.15 is source current dependent, although the amount current drawn is not much and does not really test the driver strength.

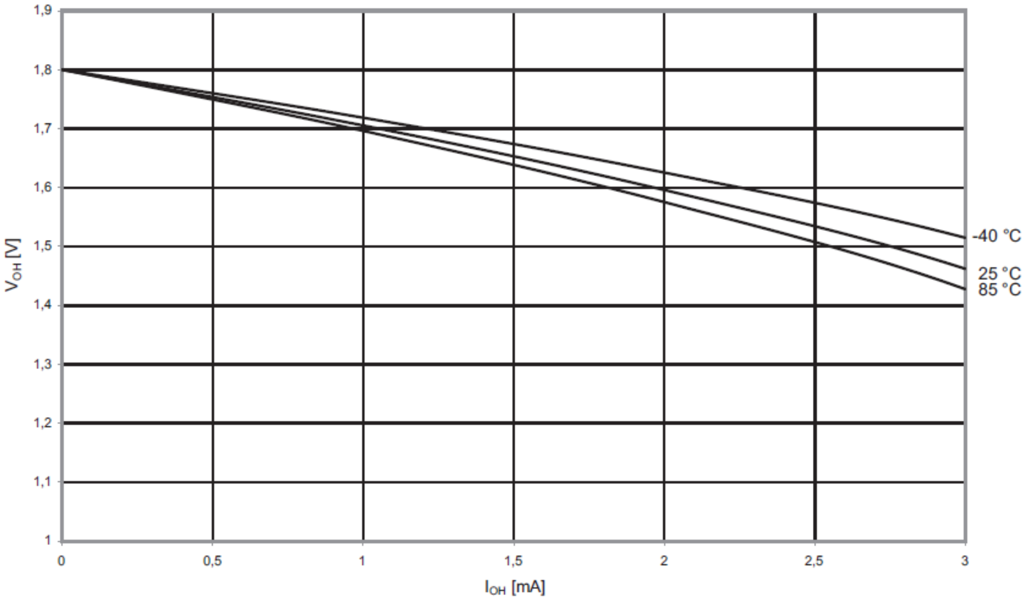


Figure 6.15. *ATtiny20 I/O-pin output high voltage vs. source current in general digital I/O-pin using 1.8 V supply voltage [6]*

Basically everything looks good, but the worst-case scenario must always be taken into account. The correct operation of every single component and functionality must be guaranteed in the temperature range of -35…+85°C. There are always some fluctuations in supply voltages and an existing possibility for voltage spikes. A bad scenario may occur, for example, when the supply voltage of the microcontroller drops and the supply voltage of the dual buffer simultaneously raises enough causing the data transmission to fail. Therefore a dual buffer cannot be used and two separate buffers have to be implemented.

The operational logic of the two separate buffers is exactly the same as the operational logic level of the dual buffer. The only difference is the possibility to use individual supply voltages. By taking the essential current leakage and worst case scenario issues into consideration, the supply voltages for the buffers are determined to be 1.8 V and 2.5 V. Buffer D4061 uses a supply voltage of 1.8 V and the buffer D4060 a supply voltage of 2.5 V. R4063 is the shared pull-up resistor for the accessory and the buffer D4061. It raises the voltage level of the signal coming from the microcontroller to 2.5 V, which is in accordance with the ECI-protocol. [6], [24], [25], [26]

Microcontroller D4060 is being driven in push-pull-mode, with both high sink and source capability. That is the reason why there is no pull-up resistor in the schematic between the microcontroller D4060 and the buffer D4061. The pull-up resistor for the buffer D4060 is R4062 and the used pull-up voltage is 1.8 V to prevent any leakage current. [6]

6.4 Flashing the microcontroller

ATtiny20 programming interface could be enabled on the electronics point of view by:

- keeping the ECI_RESETn-pin at 12 V for the entire programming session
- applying 5 V between the supply voltage and the ground

The Tiny Programming Interface (TPI) is a 3-pin interface, which uses the ECI-RESETn-pin as enable, the TPI_CLK-pin as the clock input, and the TPI_DATA-pin as data input and output. The handling of the communication occurs with an external programmer. [6]

The extra components needed for the design in order to make the flashing possible are presented in Figure 6.16. The VTG-voltage controls the switch N4069. Functional table of the switch N4069 is presented in Table 6.1. In normal operational mode the VTG-voltage is floating and the IN-port of the switch is pulled down through the resistor R4069, in which case the COM-port is in connection with the NC-port and the line in connection with ATtiny20 is used for I^2C-data transfers. During the programming the switch is turned by applying a voltage of 1.8 V to its IN-port, which creates a data path for the TPI_DATA between the flash connector, presented in Figure

6.17, and the microcontroller. The resistance-value of the resistor R4069 must be large in order to minimize the current flowing through it during the flashing operation.

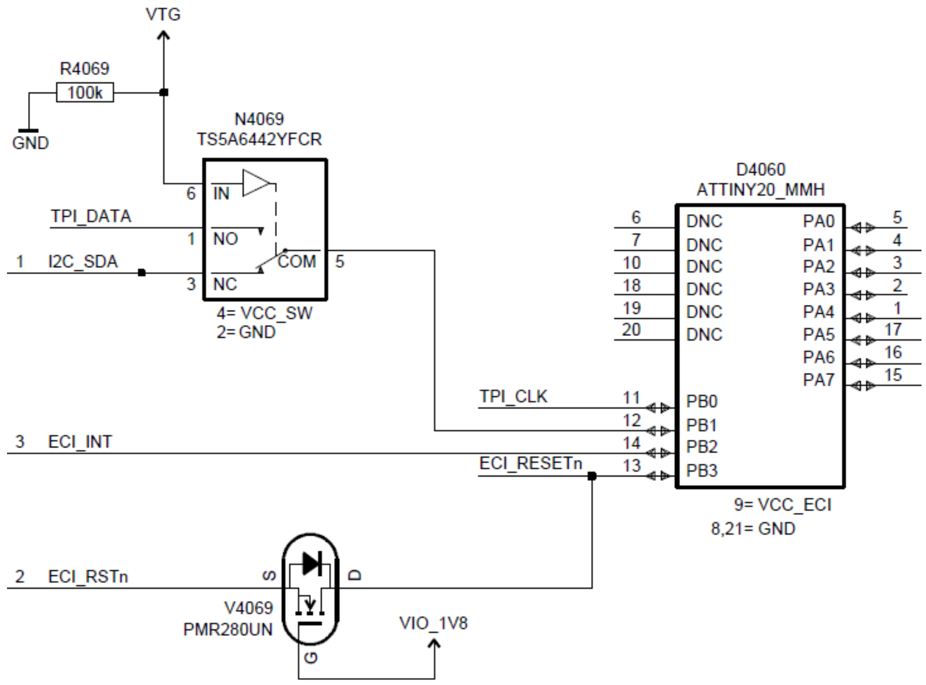


Figure 6.16. *The schematic part concerning the programming of ATtiny20*

ECI_RSTn-line is the RESET-line coming from the main processor and connected to the microcontroller RESET-pin. It is utilized during the normal operational use. The normal voltage logic high level of the reset-pin is the same as the supply voltage of the microcontroller, but during the programming the RESET-pin must be raised as high as 12 volts. ECI_RESETn is in direct contact with the flash connector, but also with the ECI_RSTn-line, so the two lines have to be separated during the programming. The NFET V4069 is used to separate the two lines. The voltage VIO_1V8 is applied to its gate during the normal operational mode, but during the programming the voltage is switched off and the NFET is not conducting. The RESET-pin of the main processor can be set to high-impedance-state during the programming so the high voltage level during the programming is not an issue, but the NFET is placed on the circuit in order to prevent leakage current to take place. [6]

All the components related to TPI, except the microcontroller, will be removed from the final design used with the commercial product. The final product does not include the possibility of programming the microcontroller after the production, therefore the design is only used during the development phase of the final product. For that reason, the selection criteria of the components are much looser than it usually is. For example, the price or the physical size of the components is not as important as it usually is. [6]

The flash connector X4069 is an ordinary pin header and the mechanical switch S4069 is a basic manual switch. The switch TS5A6442, which is used for the HSMIC-line switching, is also suitable for the programming purposes. Even if the price is not a major issue, the switch N4069 and the NFET V4069 must have as minimum of an impact to normal ECI operation as possible. These two components affect the testing and verification of the ECI, and therefore have to be selected carefully. The same performance related selection criteria apply to the selection of the switch N4060 as did during the selection of the switch N4060 in Chapter 5.1. The most important parameters in selecting the NFET are:

- drain-source breakdown voltage
- gate-source threshold voltage
- drain-source leakage current
- gate-source leakage current
- drain-source on-state resistance. [19]

TPI_DATA, TPI_CLK and ECI-RESETn are in direct contact with the ECI-flash connector presented in Figure 6.17. The connector is used for connecting an external programming device to the ECI. The mechanical switch S4069, presented in Figure 6.17, is used for enabling the TPI. The switch has three positions; normal, no connection and programming.

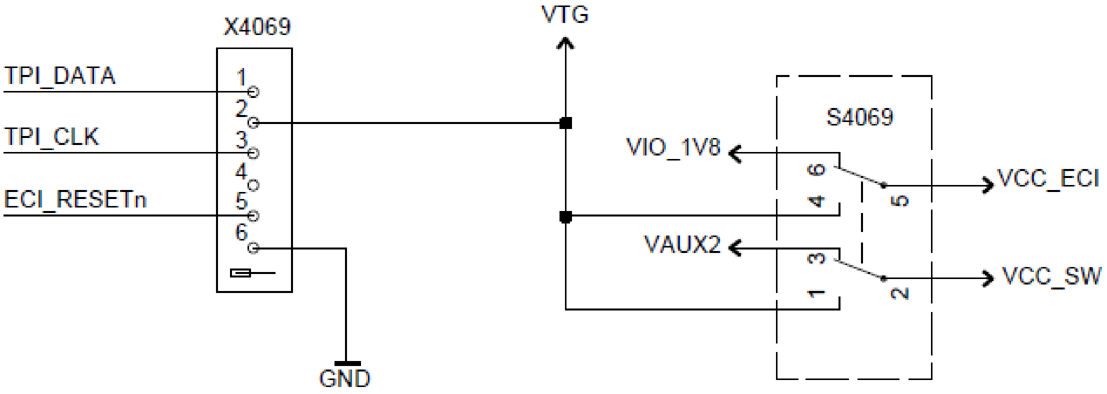


Figure 6.17. *The ECI-flash connector and the power selection switch*

Table 6.4 presents the values of the supply voltages applied to the components in the electrical circuitry with the different switch positions. In normal operational mode the switch S4069 is in “normal” position, i.e. in position presented in Table 6.4. During the programming the switch is in “program” position, i.e. VCC_ECI and VCC_SW are connected to VTG. By turning the switch to “no connection” position the whole ECI is being switched to non-operating mode. In this case, the microphone line and the AV connector are solely reserved for the audio use.

All the supply voltages related to programming and the ECI_RESETn-voltage are provided by the external programming device. As the Figure 6.17 presents, the supply voltage for the microcontroller and to the switch N4069 is the same VTG-supply voltage.

Table 6.4. *The typical supply voltage values applied to the components with different switch positions*

		Supply voltage [V]		
Pin number	**Supply name**	**Program**	**No connection**	**Normal**
1	VTG	5	0	0
2	Vcc_SW	5	0	2.5
3	VAUX2	0	0	2.5
4	VTG	5	0	0
5	Vcc_ECI	5	0	1.8
6	VIO_1V8	0	0	1.8

6.5 Support for the AHJ-headset

As stated in Chapter 2.1.1, the HSMIC- and the HSAGND-line have to be interchanged in order to provide support for AHJ-headsets, The MSIC has a control line AHJ_SW_CTRL dedicated to control the two switches N4062 and N4063 intended to do the switching operation. The part of the schematic allowing the microphone line of the AHJ-headset to be identified and used is presented in Figure 6.18. The interchange operation is done with a help of:

- control signal AHJ_SW_CTRL
- two MOSFETs V4064
- an inverter D4063
- two switches N4062 and N4063

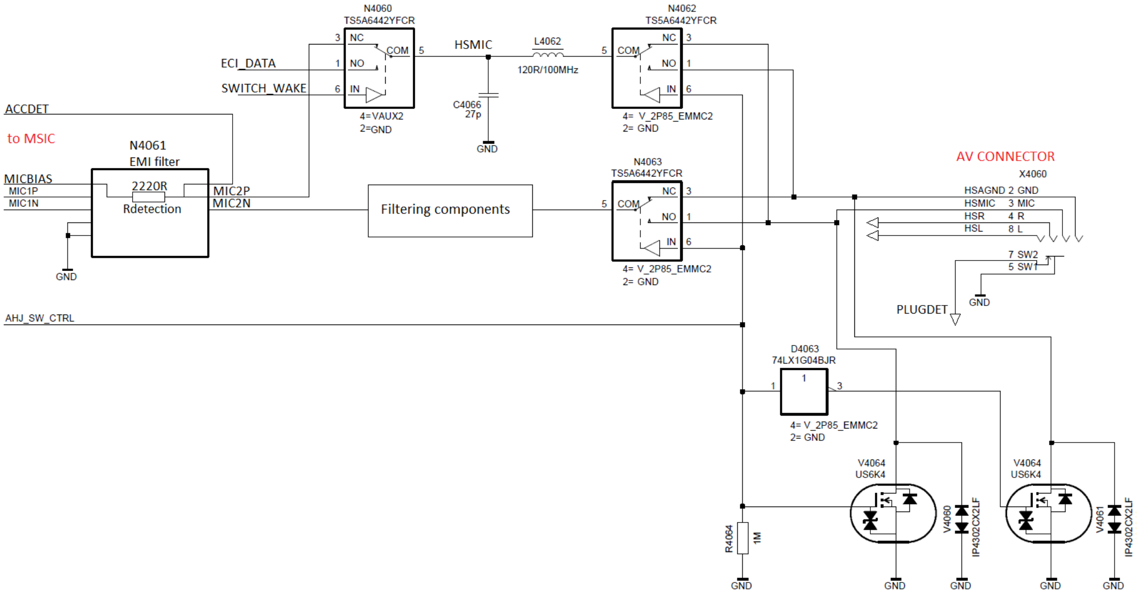


Figure 6.18. *Schematic design providing the AHJ support*

The two MOSFETs share the same package. That’s why both have the same name on the schematic; V4064. When referring to the MOSFET V4064 on the left of Figure 6.18, the expression left MOSFET is used. The same expression is used when talking about the other MOSFET, but the word left is being replaced with the word right. [37]

Normally the control signal AHJ_SW_CTRL is not activated, i.e. it is being kept LOW, and the switches N4062 and N4063 are in position presented in Figure 6.18. The left MOSFET V4064 is not conducting and the HSMIC-line is in connection with the switch N4060. Because of the inverter D4063 the right MOSFET V4064 is conducting and the HSAGND-line is grounded. So, as long as the AHJ_SW_CTRL-line is not activated, everything works as nothing has changed from the basic solution presented in Appendix A. Of course, the switches cause some extra resistance to line, but the added value is insignificant. Above all, everything still works as it should.

If the detection protocol supported AHJ-headsets, the AHJ_SW_CTRL-line would be activated when needed. In that case the switches N4062 and N4063 change their active inputs and the left MOSFET V4064 starts to conduct. At the same time, due to the inverter D4063 the right MOSFET V4064 stops to conduct. AV connector’s HSMIC-port is now grounded and is in connection with the MIC2N-input of the EMI-filter N4061 and the HSAGND-port is in connection of the switch N4060. Thus, the two AV connector ports are interchanged. [37], [38]

The MOSFET V4064 has a gate threshold level of 1.0 V at maximum and the voltage high level of the AHJ_SW_CTRL-line is about 1.8 V and the high level output of the inverter D4063 is over 2 V at minimum, so the correct operation of the MOSFETs can be guaranteed. The resistor R4064 is placed to schematic to ensure the grounding of the AHJ_SW_CTRL-line when the line is not activated. [37]

7 TESTING AND VERIFICATION

This chapter intends to give an extensive explanation of what measurement methods are used, the ideas behind them and how they are realized. Equipment used in test is described as accurately as is needed concerning the measurements in this thesis. The purpose of the tests carried out is justified and the most significant findings and abnormities are being highlighted.

The operation of the phone is always a sum of all variables, e.g. electrical components, connections between the components and software. Most of the components that are somehow connected to the design and to its verification are commercially available and have accurate datasheets, which are trusted to be accurate. There are, however, some new components, of which electrical performance is not fully known and has never been tested. The most important failures to be found are the ones that take the longest to be repaired, e.g. any failures in the silicon of the main processor.

The most important reason for testing altogether is to be sure that the electrical design really works. The first step is to know all the functionalities of which the ECI should be capable. With the help of the electrical measurements all abnormities in electrical behavior, functional errors and software issues are being researched. The electrical testing and verification includes measurements of:

- signal integrity
- signal timing
- performance test.

The meaning of the performance tests is to test the device’s temperature behavior. The basic signal integrity and timing tests are done in room temperature of +25°C. In performance part of the test, the same tests are carried out, but at temperatures of -30°C and +85°C. In these cases however, performance testing had to be left out, because the main components of the phone terminal, which are not part of the ECI, were not yet functioning properly in such low or high temperatures at the time of writing this thesis.

The measurements overall cover much more than just the visible electrical design. There are lots of functionalities, for example inside the MSIC, which are connected to the ECI. If the MSIC did not do the job it was supposed to, the accessory detection would lead to a failure causing the ECI-communication never to take place.

As important as it is to verify the functionality of the electrical circuitry, it is also important to test the functionality of the software. Functionality of the software is hardware dependent and vice versa. Many parameters directly affecting the electrical behavior of the ECI are software configurable. The main processor, for example has three internal pull-up resistors per I/O-port. The selection of which internal pull-up resistor the main processor activates is done with the help of software configurations.

Current consumption is always very important when considering mobile phones. Current consumption testing is not possible to be carried out because all the used supplies are also connected to other parts or blocks of the phone as well. It would be possible to electrically disconnect some of those blocks, but not all. That is why it is not possible to get reliable results of current consumption. All the components are selected, so that the information of their datasheet can be trusted and an accurate estimation can be given even without actual measurements.

7.1 Signal integrity

Primarily signal integrity involves the electrical performance of the traces and other affecting components within an electronic product. It is a measure of the quality of an electrical signal. A digital signal with strong integrity would have clean and fast transitions, stable and valid logic levels, accurate placement in time and it would be free of transients. Direct signal measurements are the only way to discover the causes of signal integrity-related problems. The parameters that are being measured include:

- voltage levels
- rise and fall times
- interference
- over/undershoot. [39]

Transmission line effects can have a significant effect on the data being sent, although at low speeds, the frequency response has little influence on the signal. The most likely issues to be found during the testing are probably layout design dependent variables, like ringing and crosstalk. Examples of signal integrity related issues that can be corrected by improving the schematic design are the following:

- signal rise times, which can be on most cases altered by lowering or raising the value of the RC time constant
- signal fall times, which are in some cases possible to be adjusted by altering the switching speed of the output transistor or by changing the value of the RC time constant
- series resistor against voltage spikes. [39]

7.2 Signal timings

The signal timing measurements are done in order to find any abnormities or errors from the transferred signals. Things affecting data transfer protocols and signal timings are mostly software configurable or at least cannot be fixed by changing the basic electrical design as the signals are being generated inside the ICs. The used data transfer protocols are also well known. Thus, the timings measurements can be done following the electrical characteristics of the I^2C- and ECI-protocol. If some of the measurements do not pass the set criteria, the only thing that can be generally done is to notify the person or a team responsible of the software. [28]

A clocked bus protocol such as I^2C requires the data to be stable at its input for a specified time before the clock arrives. This is known as setup time. Similarly, the input data must remain valid for a specified time after the leading edge of the clock. This is known as hold time. In case of the ECI- and I^2C-protocol, the leading edge is the rising edge. Setup and hold violations can occur in digital systems. Violating setup or hold requirements can cause unpredictable glitches on the output, or no output transition at all. Setup and hold time requirements are decreasing as device speeds increase, making the timing relationships harder to troubleshoot. [28], [39]

The actual information transferred is not important on the electrical design point of view, because it is completely software dependent. All data transfer protocols have specific start and stop commands as well as some other protocol specific commands. The target of the timing measurements is to verify the right operation of the protocol specific commands. The timing measurements are executed with the help of an oscilloscope by measuring whether the signal timings meet the given criteria or not, i.e. are the measured timing values within the limits specified by the protocol specification. [2], [28]

7.3 Measurement setup

The used measurement setup includes:

- Tektronix DSA71604B oscilloscope
- two Tektronix P6245 probes
- power supply
- phone terminal with an appropriate test software
- different phone accessories.

The issue of matching a probe to an oscilloscope has been tremendously simplified by oscilloscope manufacturers. Nowadays the probes and the oscilloscopes are designed as complete systems. As a result, the best probe-to-oscilloscope match is always obtained by using the standard probe specified by the oscilloscope manufacturer. [28]

The selection of the oscilloscope is not important, because the measured signals are relatively slow as well as the rising and falling edges of the signals. There are not many from which to choose anyway and they all meet the criteria set by the measuring environment perfectly. The selection of the used probes, on the contrary, is an essential factor. There are several other crucial factors besides the matching to be taken into consideration. [28]

7.3.1 Probe selection

A role of the probe is to deliver the representation of the signal to the oscilloscope. An oscilloscope can only display and measure the signal that the probe delivers to its input. Most probes are designed to match the inputs of specific oscilloscope models. The probes have to be selected so they match with the oscilloscope and the used measurement application. [28]

It is imperative that the probe has minimum impact on the probed circuit and that it maintains adequate signal fidelity for the desired measurements. If the probe does not maintain signal fidelity, if it changes the signal in any way or changes the way a circuit operates, the oscilloscope sees a distorted version of the actual signal. The result can be wrong or misleading measurements. Therefore, the selection of the probes is critical to measurement quality and a great amount of discretion must be taken into consideration while making the decision of which probe to use. [10, p. 95]

The ideal probe would offer the following key attributes:

- connection ease and convenience
- absolute signal fidelity
- zero signal source loading
- complete noise immunity.

In regards to the measurements related to this thesis, the most critical probe-parameter is the signal source loading, because of the high termination resistance values in open-drain data lines. Many active probes offering a high-bandwidth measurement range also tend to have relatively low input impedance values. For example, a widely used Tektronix P7240 probe works in relatively high frequency band, but has an input impedance value of 20 kΩ. If the probe is used to measure, for example, an I^2C-line having a pull-up voltage of 1.8 V and a pull-up resistor of 2 kΩ, the measured voltage high logic level would be a lot lower than it actually is. The probe creates a connection having an impedance of 20 kΩ to ground forming a voltage divider circuit. The observed voltage high level on oscilloscope’s screen would be about 1.64 volts. The loading of the probe always descends the voltage level while the probe is in connection to the line. The impedance value of the probe is frequency-dependent; generally speaking, the bigger the frequency, the lower the impedance and bigger the loading effect will be. [28], [40]

In practice, a probe with zero signal source loading cannot be achieved. This is because a probe must draw some small amount of signal current in order to develop a signal voltage at the oscilloscope input. Consequently, some signal source loading is to be expected when using a probe. The goal, however, should always be to minimize the amount of loading through appropriate probe selection. [10, p. 95]

Another critical parameter is the capacitance at the probe tip, which is the capacitance that the probe adds to the circuit test point. The capacitance stretches a signal’s rise time by increasing the value of the RC time constant in the line under measurement. Thus, a low tip capacitance minimizes errors in making rise time measurements. Also, if a pulse’s duration is less than five times the probe’s RC time constant, the amplitude of the pulse is affected, i.e. for low frequencies, the tip capacitance has a reactance that is very high, and there’s little or no effect. [28]

High-impedance, low-capacitance probes are the best choice for minimizing probe loading of the signal source. The most important properties of the selected Tektronix P6245 active probe:

- ≥ 1.5 GHz bandwidth (typical, probe only)
- ≤ 1 pF input/tip capacitance
- 1 MΩ input resistance. [32]

Another important thing to be taken into consideration is the added inductance to ground from the probe ground lead. The longer the ground lead, the greater the inductance and the greater the likelihood of seeing ringing on fast pulses. To avoid grounding problems, as short ground leads should be used as possible. Substituting other means of grounding can cause ringing to appear on measured pulses. On the other hand, the low capacitance reduces ground lead effects and compensates the possible effects of longer ground leads. [28]

7.4 Functional testing

The first step is to test the basic functionality of the accessories with the phone terminal. Different non-ECI- and ECI-accessories are being connected to the phone terminal and tested, whether they work with the phone terminal as expected. Everything went smoothly enough even though the software used for testing was unstable at times. All the functionalities of all the tested accessories functioned as they were supposed to.

Functionality of the audio outputs, microphone input and ECI-functionalities were tested including all imaginable combinations to verify the correct operation of the ECI and the audio interface. The functionality of the ECI-functionalities and the microphone audio were tested for example, during a phone call and audio playback. The other tests included for example, detection of plug insertion and removal and accessory identification.

The analog-digital-converter inside the MSIC is usually used for some tasks during the accessory detection phase. The test software, however, includes a feature, which allows the analog-digital-converter to be used for measuring the ACCDET-line voltage with the MICBIAS enabled and the measured value to be printed on screen. Table 7.1

includes the measured voltage values from all the tested accessories. The values are comparable to detection phase, which uses the threshold voltage level of 0.6 volts.

According to the accessory detection protocol, the ADC is used in normal operational mode only to measure voltages that are under the 0.6 V threshold level. Values above the 0.6 V level are not actually measured, but only compared with the comparators to the threshold level.

Table 7.1. *Measured voltage values from the ACCDET-line*

Accessory	ADC-voltage	Unit	Comment
MICBIAS/open cable	2097	mV	empty jack
WH-601	1539	mV	ECI-accessory
WH-701	1684	mV	ECI-accessory
HS-117	1637	mV	ECI-accessory
HS-43	1684	mV	ECI-accessory
HS-54	1656	mV	ECI-accessory
Video cable	85	mV	limit 40-90 mV
Headphones	21	mV	limit 0-40 mV
Headset	1433	mV	non-ECI

The measured voltage values are correct. All the measured voltage values of the ECI-accessories and the headset with a microphone are safely over the 0.6 V detection threshold level. The voltage value for the headset is also over 0.6 V as expected. The video cable and the headphones meet the set detection criteria perfectly.

7.5 Electrical measurements

All the signals between the microcontroller and the main processor have separate test points, but the signals between the microcontroller and the AV-connector can only be measured directly from the AV-connector. Signal integrity measurements are done for the I^{2}C-bus lines as well as to ECI-data line, and also to all the command lines. The only line that cannot be measured is the ECI_WAKE-line.

The timing measurements are done only for the I^{2}C-bus lines. Because of the confidentiality issues, the measurements of the ECI are limited to signal integrity tests. The selection of which ECI-accessory to use during the I^{2}C-measurements does not matter, because the individual accessories does not have any effect to the I^{2}C-protocol. Only the transferred data information would change if different ECI-accessories were used.

The I^{2}C measurements are repeatable only with the same software configurations used during the testing, because most of the timings and even integrity related measurements are software configurable. The components involved to signal generation, i.e. the main processor and the microcontroller, would have to stay unchanged. The same goes also with the ECI-communication measurements, because the ECI-controller and the software it uses would have to be exactly the same. Adding the fact that the voltage levels of the both data protocols are also not fixed makes the measurements completely product dependent. [2]

All the oscilloscope measurements are done using the persistence mode if possible. Persistence mode is useful for spotting glitches, when it is necessary to spot a rare fault event hidden in a series of repeated normal events. In normal scope mode, repetitive waveforms may appear on the display for a fraction of a second, too quickly for pressing the stop button in time freeze the abnormity on the screen. Persistence mode keeps the waveforms on the display for a predetermined time, allowing the trigger options to be set up to capture waveforms more reliably. If the waveform is not repetitive, the persistence mode cannot be used. If the mode was used, the screen would be full of waveforms and the separation or interpretation of the different waveforms is impossible. [28]

Each figure including a capture from the oscilloscope screen has a label which clarifies the signal on the screen. All the figures have a sign with a percentage value in it indicating the two spots between which the measurement marked with an asterisk is done. In the signal integrity measurements the signs have no significance other than to show the used reference levels. On the other hand, in the timing measurements the spots marked with the signs show the measurement mentioned on the label is done. [28], [39]

7.5.1 Signal Integrity Tests

The reference levels for the I^{2}C-protocol are 30% and 70% of the VIO_1V8 supply voltage. The used supply voltage value for determining the minimum and maximum limits is the typical value in order to get the measurement results easy to read. If the measured value is very close to the limit, the worst-case scenario values are being taken into consideration. Definitions for the electrical characteristics are presented in Figure 7.15. [2]

Table 7.2 presents the electrical characteristics of the I^{2}C-bus signals. The voltage levels and the rise and fall times of the clock and data line are measured. The I^{2}C_SCL-signal is generated by the main processor. The I^{2}C_SDA-signal is generated by the main processor or the microprocessor depending on the direction of the sent data. [2]

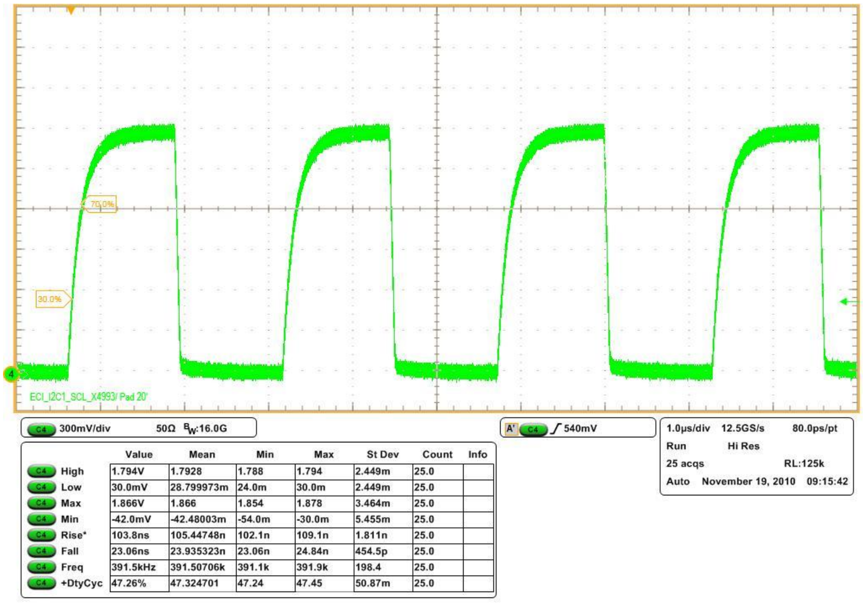
The minimum limit of the fall times should also include the impact of the RC time constant. The combined effect of the bus capacitance and the line resistance is so insignificant that it is left out of the limit values. The effect is at all events under one nano seconds. Even if a series resistor with a low resistance value was used, the falling edge would be significantly slower and new limits would have to be determined.

Table 7.2. *Electrical characteristics of the SDA and SCL I/O-stages [2]*

Measured parameter	Min	Typ.	Max	Unit	Measured Values	Comments
I^{2}C_SCL						
Voltage high	1.3	1.8	2.1	V	**1.79**	PASSED
Voltage low	-0.3	0	0.65	V	**0.021**	PASSED
Rise time	20		300	ns	**105**	PASSED
Fall time	20		300	ns	**24**	PASSED
I^{2}C_SDA, the main processor is sending						
Voltage high	1.3	1.8	2.1	V	**1.79**	PASSED
Voltage low	-0.3	0	0.65	V	**0.021**	PASSED
Rise time	20		300	ns	**299**	PASSED
Fall time	20		300	ns	**25**	PASSED
I^{2}C_SDA, ATtiny20 is sending						
Voltage high	1.3	1.8	2.1	V	**1.79**	PASSED
Voltage low	-0.3	0	0.65	V	**0.060**	PASSED
Rise time	20		300	ns	**264**	PASSED
Fall time	20		300	ns	**91**	PASSED

Figure 7.1 presents the I^{2}C_SCL-signal. Everything with regards to the signal integrity looks good. The fall time could not be any faster or it would violate the limits. Usually the problems caused by too rapidly falling edges are related to EMC-issues and must be avoided.

The rising edge of the clock signal is very near the upper limit and must be investigated further. As stated, the design worked well in practice but the test is done only in room temperatures and only with a single phone terminal. There are always differences between components due to the tolerances in component values. The supply voltages and pull-up voltages can also vary causing variation in limits. All in all, every single phone coming from the production must operate properly. With certainty can be said, that the rise time value does not meet the demands set by the worst-case scenario.



C4 300mV/div 50Ω B_W:16.0G

A' C4 540mV

1.0µs/div 12.5GS/s 80.0ps/pt
Run Hi Res
25 acqs RL:125k
Auto November 19, 2010 09:15:42

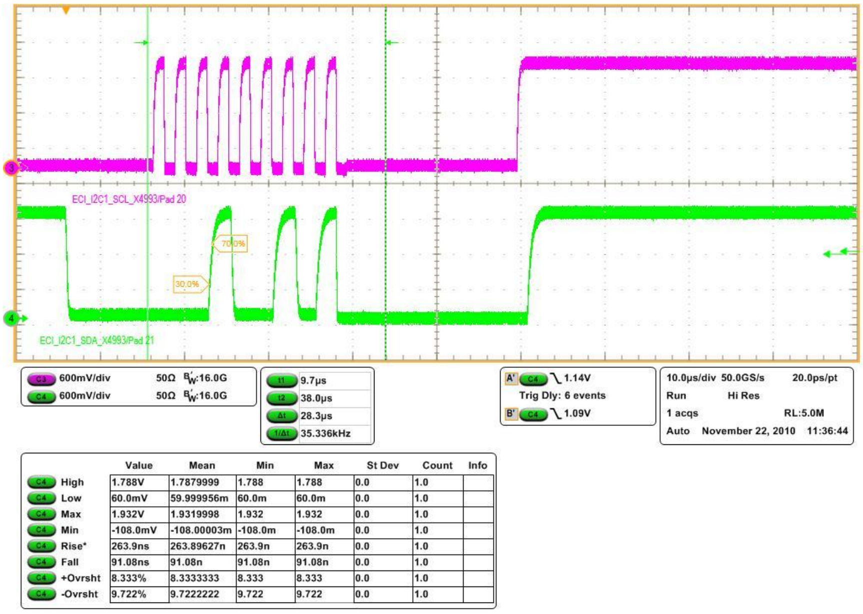
	Value	Mean	Min	Max	St Dev	Count	Info
C4 High	1.794V	1.7928	1.788	1.794	2.449m	25.0	
C4 Low	30.0mV	28.799973m	24.0m	30.0m	2.449m	25.0	
C4 Max	1.866V	1.866	1.854	1.878	3.464m	25.0	
C4 Min	-42.0mV	-42.48003m	-54.0m	-30.0m	5.455m	25.0	
C4 Rise*	103.8ns	105.44748n	102.1n	109.1n	1.811n	25.0	
C4 Fall	23.06ns	23.935323n	23.06n	24.84n	454.5p	25.0	
C4 Freq	391.5kHz	391.50706k	391.1k	391.9k	198.4	25.0	
C4 +DtyCyc	47.26%	47.324701	47.24	47.45	50.87m	25.0	

Figure 7.1. *I^{2}C_SCL integrity*

The only possible thing to do in electrical design is to modify the layout design due to the main processor's internal pull-up resistors and the lack of any components in the signal line. In practice, that means informing the person who is responsible of the layout design. The choices are to shorten the traces or use different routing between the main processor and the microcontroller in order to reduce the bus capacitance.

Another possibility is to inform the person who is responsible of the main processor software to change the used pull-up resistor to another one which has a lower resistance value. This was done, but the changes take time and did not make in time so, that they could be used in this thesis.

Figure 7.2 presents the I^{2}C-data signal while the microcontroller is pulling the line down and Figure 7.3 presents a situation when then the main processor is driving the data line. The green vertical lines in Figure 7.2 indicate that the measured area has been gated. When using the gated measurement, the oscilloscope's measurement results visible on the screen are only from within the gated area. In this manner the measurements can be limited exactly where wanted.



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

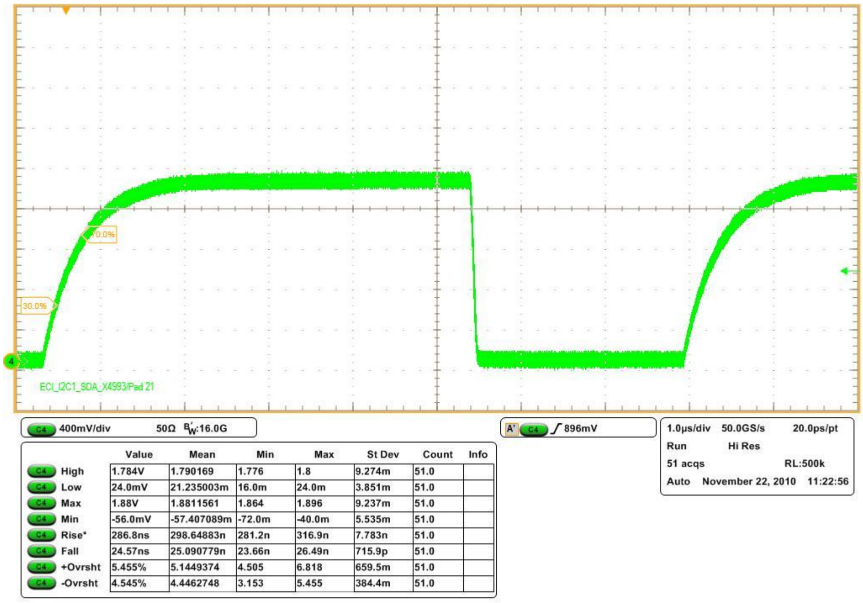
t1 9.7μs
t2 38.0μs
Δt 28.3μs
1/Δt 35.336kHz

A' C4 1.14V
Trig Dly: 6 events
B' C4 1.09V

10.0μs/div 50.0GS/s 20.0ps/pt
Run Hi Res
1 acqs RL:5.0M
Auto November 22, 2010 11:36:44

	Value	Mean	Min	Max	St Dev	Count	Info
C4 High	1.788V	1.7879999	1.788	1.788	0.0	1.0	
C4 Low	60.0mV	59.999956m	60.0m	60.0m	0.0	1.0	
C4 Max	1.932V	1.9319998	1.932	1.932	0.0	1.0	
C4 Min	-108.0mV	-108.00003m	-108.0m	-108.0m	0.0	1.0	
C4 Rise*	263.9ns	263.89627n	263.9n	263.9n	0.0	1.0	
C4 Fall	91.08ns	91.08n	91.08n	91.08n	0.0	1.0	
C4 +Ovrsht	8.333%	8.3333333	8.333	8.333	0.0	1.0	
C4 -Ovrsht	9.722%	9.7222222	9.722	9.722	0.0	1.0	

Figure 7.2. *SDA integrity (read)*



C4 400mV/div 50Ω BW:16.0G

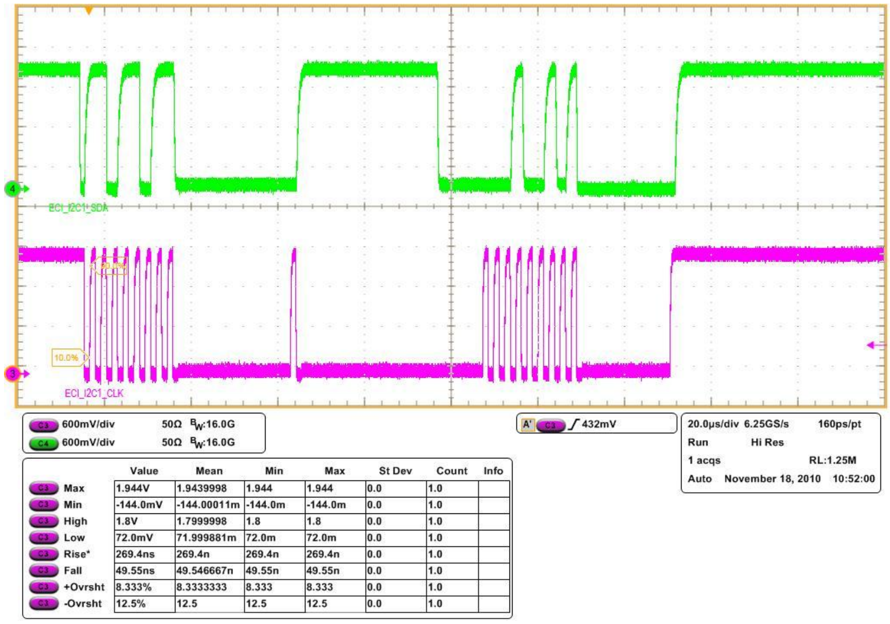
A' C4 896mV

1.0μs/div 50.0GS/s 20.0ps/pt
Run Hi Res
51 acqs RL:500k
Auto November 22, 2010 11:22:56

	Value	Mean	Min	Max	St Dev	Count	Info
C4 High	1.784V	1.790169	1.776	1.8	9.274m	51.0	
C4 Low	24.0mV	21.235003m	16.0m	24.0m	3.851m	51.0	
C4 Max	1.88V	1.8811561	1.864	1.896	9.237m	51.0	
C4 Min	-56.0mV	-57.407089m	-72.0m	-40.0m	5.535m	51.0	
C4 Rise*	286.8ns	298.64883n	281.2n	316.9n	7.783n	51.0	
C4 Fall	24.57ns	25.090779n	23.66n	26.49n	715.9p	51.0	
C4 +Ovrsht	5.455%	5.1449374	4.505	6.818	659.5m	51.0	
C4 -Ovrsht	4.545%	4.4462748	3.153	5.455	384.4m	51.0	

Figure 7.3. *SDA integrity (write)*

Because the software of the ATtiny20 microcontroller is not yet optimized, the microcontroller must stretch the clock in order to be able to handle the data. Figure 7.4 presents a situation where the microcontroller is stretching the clock line. The operation still occurs according to specifications and all the measured values are within the limits. The operation of the clock and the data line are fine hardware-wise, but the measurements indicate that the software must be developed further to achieve the required performance level. All the possible improvements regarding the clock stretching are software configurable variables.



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

A' C3 ∫ 432mV

20.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
1 acqs RL:1.25M
Auto November 18, 2010 10:52:00

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	1.944V	1.9439998	1.944	1.944	0.0	1.0	
C3 Min	-144.0mV	-144.00011m	-144.0m	-144.0m	0.0	1.0	
C3 High	1.8V	1.7999998	1.8	1.8	0.0	1.0	
C3 Low	72.0mV	71.999881m	72.0m	72.0m	0.0	1.0	
C3 Rise*	269.4ns	269.4n	269.4n	269.4n	0.0	1.0	
C3 Fall	49.55ns	49.546667n	49.55n	49.55n	0.0	1.0	
C3 +Ovrsht	8.333%	8.3333333	8.333	8.333	0.0	1.0	
C3 -Ovrsht	12.5%	12.5	12.5	12.5	0.0	1.0	

***Figure 7.4.** CLK stretching in the I^{2}C-bus*

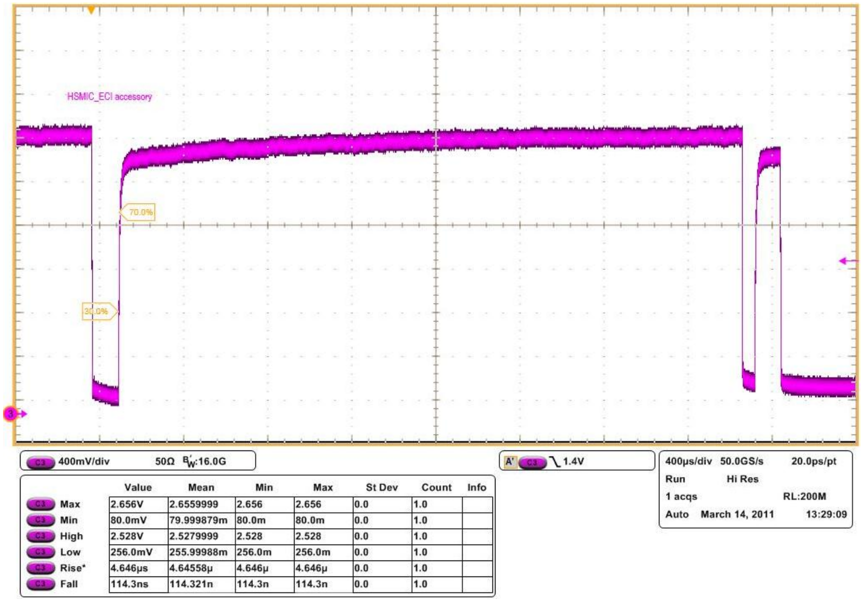
Table 7.3 presents the electrical characteristics of the ECI_RESETn-signal, the SWITCH_CTRL-signal and the ECI_INT-signal. The reference levels for the control signals are 10% and 90% of the VIO_1V8 supply voltage. Only the voltage levels of the control signals are measured, because the rise and fall times are not relevant as their functionality is based on signal state changes. The integrity of the ECI-data signal is measured and the figures attached, but only the voltage levels are analyzed. The reference levels for the ECI-data signal are 30% and 70% of the VAUX2 supply voltage.

The measured values concerning the ECI_DATA-signal are merely examples and can vary significantly between different accessories or buttons. The marked ECI_DATA-signal voltage high level values are the lowest measured voltage high values. Accordingly, the marked voltage low level values are the highest measured voltage low values.

Table 7.3. *Electrical characteristics of the ECI-bus and the control-signals*

Measured parameter	**Min**	**Typ.**	**Max**	**Unit**	**Measured Values**	**Comments**
SWITCH_CTRL						
Voltage high	1.05		Vcc ≈ 2.5	V	**1.77**	PASSED
Voltage low	-0.5	0	0.63	V	**-0.05**	PASSED
ECI_INT						
Voltage high	0.9Vcc ≈ 1.62		Vcc + 0.5 V ≈ 2.3	V	**1.80**	PASSED
Voltage low	-0.3	0	0.65	V	**-0.01**	PASSED
ECI_RESETn						
Voltage high	0.9Vcc ≈ 1.62		Vcc + 0.5 V ≈ 2.3	V	**1.78**	PASSED
Voltage low	-0.5		0.2Vcc ≈ 0.36	V	**0.02**	PASSED
ECI_DATA						
Voltage high						
-Accessory sending	1.6			V	**2.23**	PASSED
-Attiny20 sending	1.7		2.6	V	**2.28**	PASSED
Voltage low						
-Accessory sending			0.7	V	**0.15**	PASSED
-Attiny20 sending			0.7	V	**0.27**	PASSED

Figure 7.5 presents the “start”-command of the ECI-communication. The line is drawn to ground by the accessory. Figure 7.6 presents the falling edges of the ECI-data signal. The edge falling more sharply is drawn to ground by the accessory. The slower falling edge is drawn to ground by the buffer D4062. This can also be seen from the small differences in ground levels presented in Figure 7.6. The measurement is done from the AV connector, so when the accessory is pulling the line to ground the voltage level seen by the oscilloscope is the actual ground level. The only meaningful series resistance between the measuring point and the ground is then caused by the open-drain transistor and the contact resistance of the AV connector. When the buffer is pulling the line to ground, the voltage level seen in AV connector’s end is a bit higher than the actual ground level due to the series resistance in the signal path.



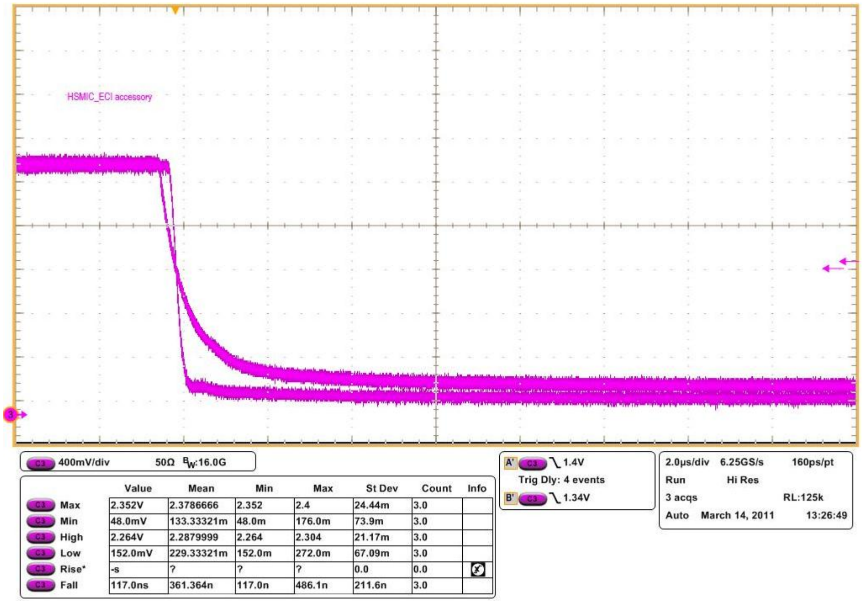
C3 400mV/div 50Ω BW:16.0G

A' C3 ⎲ 1.4V

400µs/div 50.0GS/s 20.0ps/pt
Run Hi Res
1 acqs RL:200M
Auto March 14, 2011 13:29:09

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	2.656V	2.6559999	2.656	2.656	0.0	1.0	
C3 Min	80.0mV	79.999879m	80.0m	80.0m	0.0	1.0	
C3 High	2.528V	2.5279999	2.528	2.528	0.0	1.0	
C3 Low	256.0mV	255.99988m	256.0m	256.0m	0.0	1.0	
C3 Rise*	4.646µs	4.64558µ	4.646µ	4.646µ	0.0	1.0	
C3 Fall	114.3ns	114.321n	114.3n	114.3n	0.0	1.0	

Figure 7.5. *ECI-communication START-command*



C3 400mV/div 50Ω BW:16.0G

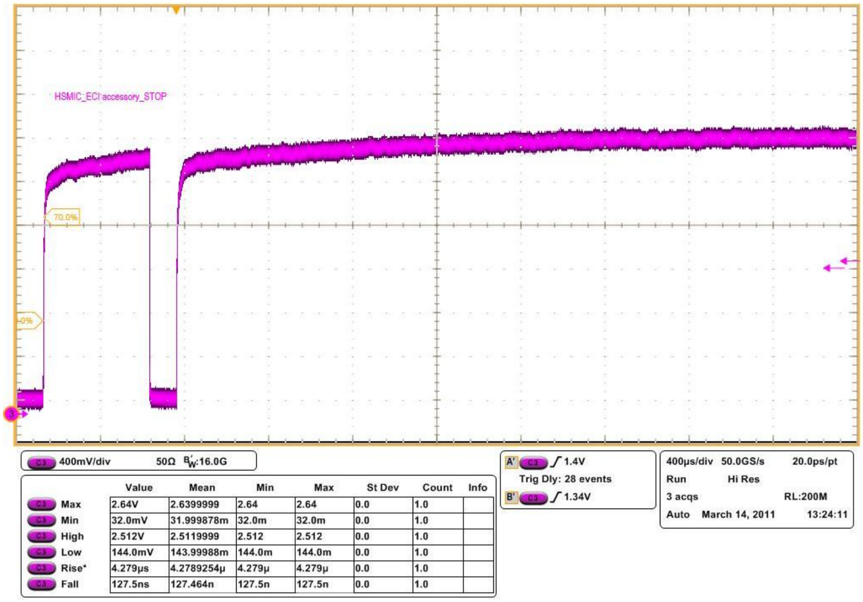
A' C3 ⎲ 1.4V
Trig Dly: 4 events
B' C3 ⎲ 1.34V

2.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
3 acqs RL:125k
Auto March 14, 2011 13:26:49

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	2.352V	2.3786666	2.352	2.4	24.44m	3.0	
C3 Min	48.0mV	133.33321m	48.0m	176.0m	73.9m	3.0	
C3 High	2.264V	2.2879999	2.264	2.304	21.17m	3.0	
C3 Low	152.0mV	229.33321m	152.0m	272.0m	67.09m	3.0	
C3 Rise*	-s	?	?	?	0.0	0.0	☒
C3 Fall	117.0ns	361.364n	117.0n	486.1n	211.6n	3.0	

Figure 7.6. *Falling edges of the ECI-communication. The edge falling more sharply is drawn down by the accessory and the other one by the buffer D4062*

Figure 7.7 presents the “stop”-command of the ECI-communication. Figure 7.8 presents a short data burst sent by the accessory. All the figures related to ECI-communication illustrate the large value of the RC time constant. The rising edges of the signal are very slow compared to rising edges of the I^{2}C-bus signals. The edges are also much slower compared to falling edges of the ECI-data signal.

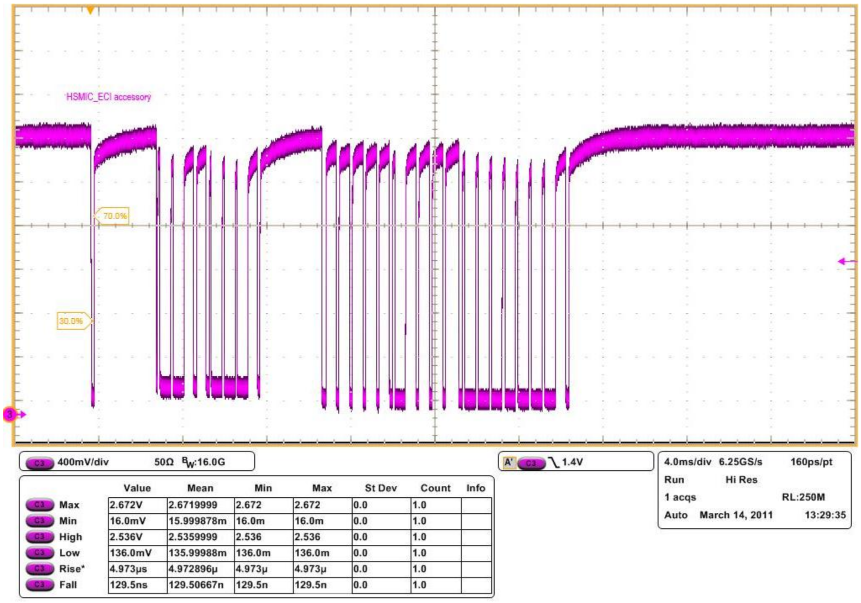


	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	2.64V	2.6399999	2.64	2.64	0.0	1.0	
C3 Min	32.0mV	31.999878m	32.0m	32.0m	0.0	1.0	
C3 High	2.512V	2.5119999	2.512	2.512	0.0	1.0	
C3 Low	144.0mV	143.99988m	144.0m	144.0m	0.0	1.0	
C3 Rise*	4.279µs	4.2789254µ	4.279µ	4.279µ	0.0	1.0	
C3 Fall	127.5ns	127.464n	127.5n	127.5n	0.0	1.0	

Figure 7.7. *Rising edge and the STOP-command of the ECI-communication*

In addition to the slow rising edges, especially Figure 7.7 shows the impact of the capacitor C1, presented in Figure 6.4, used for powering the ECI-ASIC. The high level of the ECI-data signal still rises after the signal has reached its top value. The phenomenon is easy to notice by comparing the figure to any of the I^{2}C-measurements. It can also be noticed that the capacitor is not charged if the signal is pulled down. In case of long series of logical zeros, the charging state of the capacitor drops significantly affecting to the signal voltage high level.

The voltage high level at the input of the accessory cannot be much higher than the voltage over the capacitor C1. There is a slight voltage drop over the resistor R_a, so the charging state of the capacitor cannot be spotted exactly from the Figure 7.8. Even if the rising edges are slow, the charging of the capacitor C1 occurs much slower. The point on every rising edge, when the rising substantially slows down is also the combined voltage difference over the capacitor C1 and the resistor R_a. The components significantly affecting the flattening of the signal high level are the capacitor C1, the pull-up resistor R4063 and the resistor R_a.



C3 400mV/div 50Ω BW:16.0G

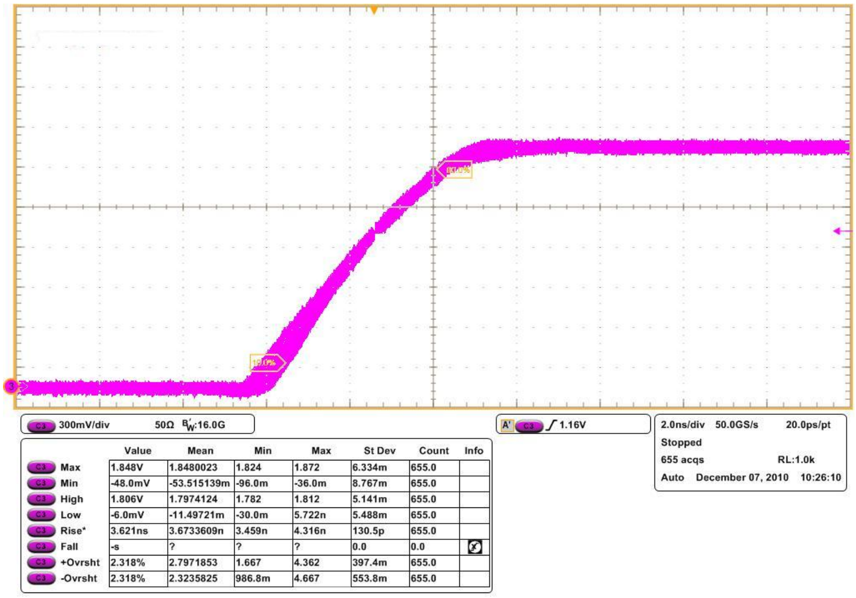
A' C3 1.4V

4.0ms/div 6.25GS/s 160ps/pt
Run Hi Res
1 acqs RL:250M
Auto March 14, 2011 13:29:35

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	2.672V	2.6719999	2.672	2.672	0.0	1.0	
C3 Min	16.0mV	15.999878m	16.0m	16.0m	0.0	1.0	
C3 High	2.536V	2.5359999	2.536	2.536	0.0	1.0	
C3 Low	136.0mV	135.99988m	136.0m	136.0m	0.0	1.0	
C3 Rise*	4.973µs	4.972896µ	4.973µ	4.973µ	0.0	1.0	
C3 Fall	129.5ns	129.50667n	129.5n	129.5n	0.0	1.0	

Figure 7.8. *ECI-data communication traffic*

Figure 7.9 presents the rising edge of the ECI_INT-signal. Figure 7.10 presents the falling edge of the ECI_INT-signal. The signal is controlled by the microcontroller. The waveforms seen on the screen of the oscilloscope are not as sharp as the waveforms in the other figures. This is because of the used persistence mode and the small fluctuations on the waveform. This is also nothing to be worried because there are not any noticeable abnormities in the waveform or notable variations in the rise or the fall times. [39]



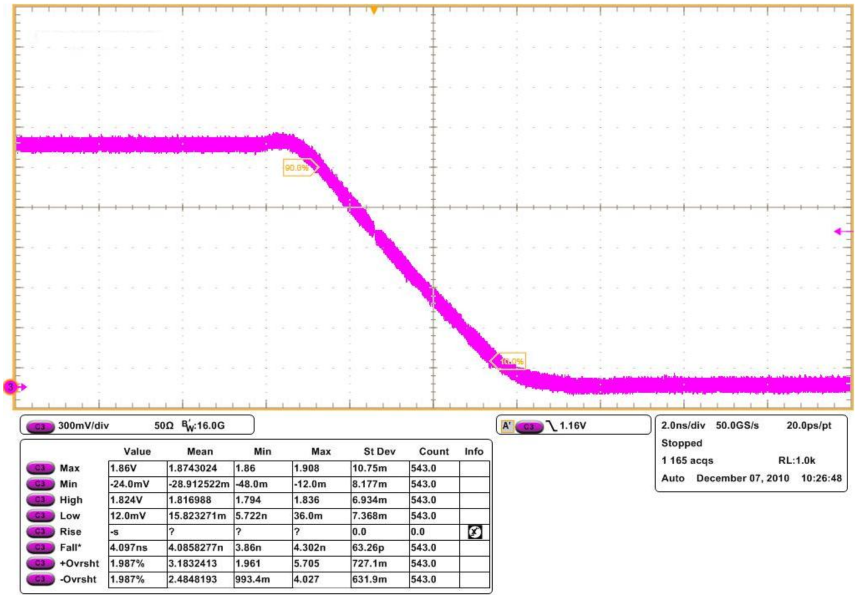
C3 300mV/div 50Ω BW:16.0G

A' C3 1.16V

2.0ns/div 50.0GS/s 20.0ps/pt
Stopped
655 acqs RL:1.0k
Auto December 07, 2010 10:26:10

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	1.848V	1.8480023	1.824	1.872	6.334m	655.0	
C3 Min	-48.0mV	-53.515139m	-96.0m	-36.0m	8.767m	655.0	
C3 High	1.806V	1.7974124	1.782	1.812	5.141m	655.0	
C3 Low	-6.0mV	-11.49721m	-30.0m	5.722n	5.488m	655.0	
C3 Rise*	3.621ns	3.6733609n	3.459n	4.316n	130.5p	655.0	
C3 Fall	-s	?	?	?	0.0	0.0	☒
C3 +Ovrsht	2.318%	2.7971853	1.667	4.362	397.4m	655.0	
C3 -Ovrsht	2.318%	2.3235825	986.8m	4.667	553.8m	655.0	

Figure 7.9. *Rising edge of the ECI_INT-signal*



C3 300mV/div 50Ω B'W:16.0G

A' C3 ⌄ 1.16V

2.0ns/div 50.0GS/s 20.0ps/pt
Stopped
1 165 acqs RL:1.0k
Auto December 07, 2010 10:26:48

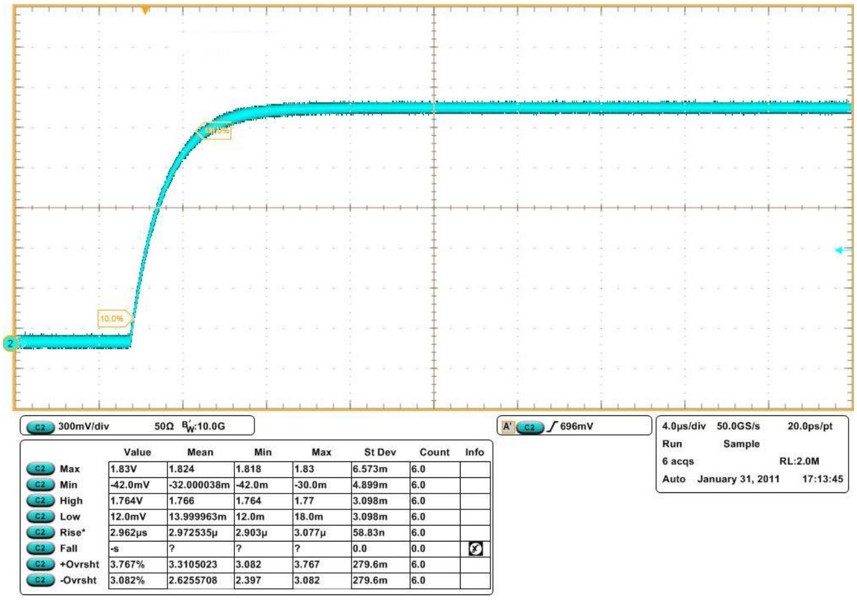
	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	1.86V	1.8743024	1.86	1.908	10.75m	543.0	
C3 Min	-24.0mV	-28.912522m	-48.0m	-12.0m	8.177m	543.0	
C3 High	1.824V	1.816988	1.794	1.836	6.934m	543.0	
C3 Low	12.0mV	15.823271m	5.722n	36.0m	7.368m	543.0	
C3 Rise	-s	?	?	?	0.0	0.0	☒
C3 Fall*	4.097ns	4.0858277n	3.86n	4.302n	63.26p	543.0	
C3 +Ovrsht	1.987%	3.1832413	1.961	5.705	727.1m	543.0	
C3 -Ovrsht	1.987%	2.4848193	993.4m	4.027	631.9m	543.0	

Figure 7.10. *Falling edge of the ECI_INT-signal*

Figure 7.11 presents the rising edge of the ECI_RESETn-signal and the Figure 7.12 presents the falling edge of the ECI_RESETn-signal. The signal has noticeable undershoot, which is caused by the open-drain transistor inside the main processor. The ideal thing to do is to tell the processor-manufacturer to use slower switching transistors. Making changes to silicon of the main processor is expensive and it takes time. The more practical solution, however, is to use some other method. For example, using a series resistor to reduce the phenomenon or ignoring it completely is a better option, because the minimum value is still within the limits of the microcontroller. [6]

According to the datasheet of the microcontroller, the input negative-voltage of the buffer may be exceeded if the input current ratings are observed. The control signals always use very small amount of current to operate, so the undershoot is not really a problem. However, if the undershoot is found to be any bigger during the performance tests, some filtering components must be added to design. [6]

The undershoot seen by the microcontroller is actually not as substantial as the measurement results suggest, because the measuring point is between the MOSFET V4069 and the main processor. The MOSFET dampens the undershoot a bit and the waveform would not look as bad if measured from the microcontroller’s input. The final product will not include the MOSFET, so the positive effect it has in tested design can be ignored and analyze the figure as it is. [7, p. 153-154]



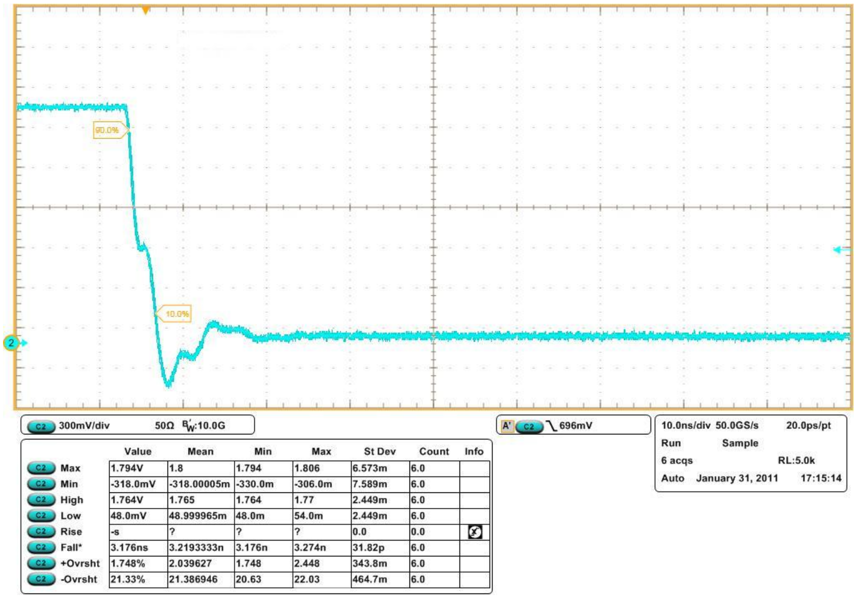
C2 300mV/div 50Ω BW:10.0G

A' C2 ∫ 696mV

4.0µs/div 50.0GS/s 20.0ps/pt
Run Sample
6 acqs RL:2.0M
Auto January 31, 2011 17:13:45

	Value	Mean	Min	Max	St Dev	Count	Info
C2 Max	1.83V	1.824	1.818	1.83	6.573m	6.0	
C2 Min	-42.0mV	-32.000038m	-42.0m	-30.0m	4.899m	6.0	
C2 High	1.764V	1.766	1.764	1.77	3.098m	6.0	
C2 Low	12.0mV	13.999963m	12.0m	18.0m	3.098m	6.0	
C2 Rise*	2.962µs	2.972535µ	2.903µ	3.077µ	58.83n	6.0	
C2 Fall	-s	?	?	?	0.0	0.0	☒
C2 +Ovrsht	3.767%	3.3105023	3.082	3.767	279.6m	6.0	
C2 -Ovrsht	3.082%	2.6255708	2.397	3.082	279.6m	6.0	

Figure 7.11. *Rising edge of the ECI_RESETn-signal*



C2 300mV/div 50Ω BW:10.0G

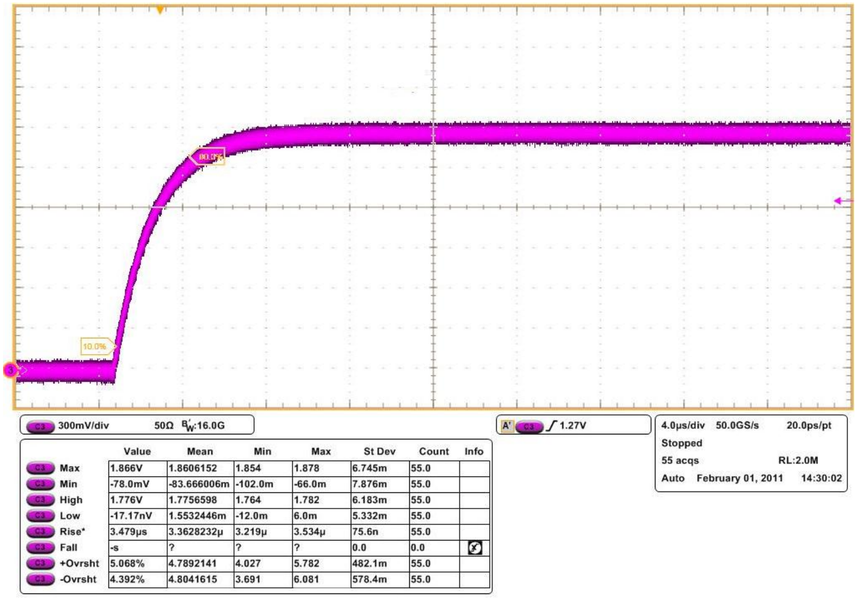
A' C2 ⟍ 696mV

10.0ns/div 50.0GS/s 20.0ps/pt
Run Sample
6 acqs RL:5.0k
Auto January 31, 2011 17:15:14

	Value	Mean	Min	Max	St Dev	Count	Info
C2 Max	1.794V	1.8	1.794	1.806	6.573m	6.0	
C2 Min	-318.0mV	-318.00005m	-330.0m	-306.0m	7.589m	6.0	
C2 High	1.764V	1.765	1.764	1.77	2.449m	6.0	
C2 Low	48.0mV	48.999965m	48.0m	54.0m	2.449m	6.0	
C2 Rise	-s	?	?	?	0.0	0.0	☒
C2 Fall*	3.176ns	3.2193333n	3.176n	3.274n	31.82p	6.0	
C2 +Ovrsht	1.748%	2.039627	1.748	2.448	343.8m	6.0	
C2 -Ovrsht	21.33%	21.386946	20.63	22.03	464.7m	6.0	

Figure 7.12. *Falling edge of the ECI_RESETn-signal*

Figure 7.13 presents the rising edge of the SWITCH_CTRL-signal and Figure 7.14 presents the falling edge of the SWITCH_CTRL-signal. The signal is controlled by the microcontroller. The signal integrity of the edges looks better than the edges driven by the microcontroller as expected. The main processor is a new component and still under development as the microprocessor is an old component with well tested outputs.



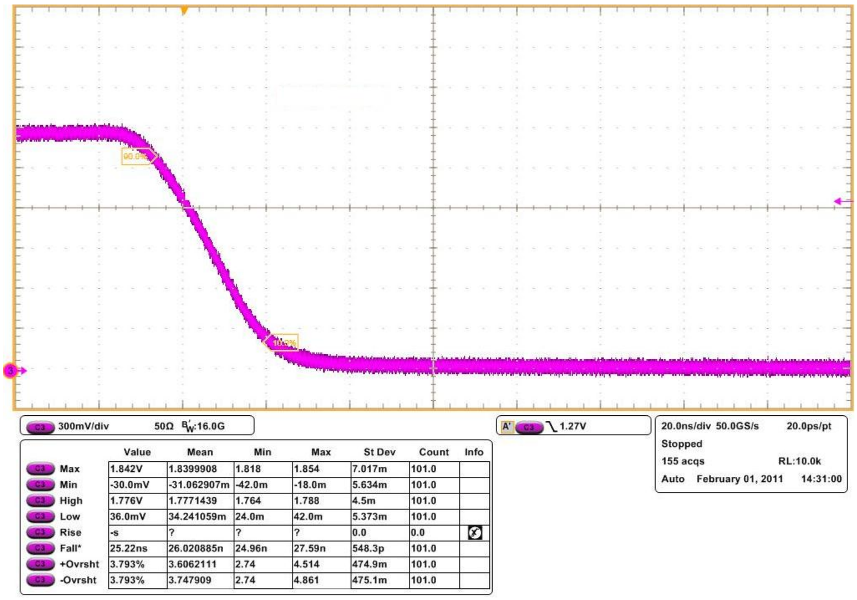
C3 300mV/div 50Ω BW:16.0G

A' C3 1.27V

4.0μs/div 50.0GS/s 20.0ps/pt
Stopped
55 acqs RL:2.0M
Auto February 01, 2011 14:30:02

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	1.866V	1.8606152	1.854	1.878	6.745m	55.0	
C3 Min	-78.0mV	-83.666006m	-102.0m	-66.0m	7.876m	55.0	
C3 High	1.776V	1.7756598	1.764	1.782	6.183m	55.0	
C3 Low	-17.17nV	1.5532446m	-12.0m	6.0m	5.332m	55.0	
C3 Rise*	3.479μs	3.3628232μ	3.219μ	3.534μ	75.6n	55.0	
C3 Fall	-s	?	?	?	0.0	0.0	☒
C3 +Ovrsht	5.068%	4.7892141	4.027	5.782	482.1m	55.0	
C3 -Ovrsht	4.392%	4.8041615	3.691	6.081	578.4m	55.0	

Figure 7.13. *Rising edge of the SWITCH_CTRL-signal*



C3 300mV/div 50Ω BW:16.0G

A' C3 1.27V

20.0ns/div 50.0GS/s 20.0ps/pt
Stopped
155 acqs RL:10.0k
Auto February 01, 2011 14:31:00

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	1.842V	1.8399908	1.818	1.854	7.017m	101.0	
C3 Min	-30.0mV	-31.062907m	-42.0m	-18.0m	5.634m	101.0	
C3 High	1.776V	1.7771439	1.764	1.788	4.5m	101.0	
C3 Low	36.0mV	34.241059m	24.0m	42.0m	5.373m	101.0	
C3 Rise	-s	?	?	?	0.0	0.0	☒
C3 Fall*	25.22ns	26.020885n	24.96n	27.59n	548.3p	101.0	
C3 +Ovrsht	3.793%	3.6062111	2.74	4.514	474.9m	101.0	
C3 -Ovrsht	3.793%	3.747909	2.74	4.861	475.1m	101.0	

Figure 7.14. *Falling edge of the SWITCH_CTRL-signal*

All the measurements related to signal integrity are within the limits set by the components or the data transfer protocols. The only potential cause for an error is the undershoot in the ECI_RESETn-line, which can be fixed in the future if the performance tests fail. The clock stretching is also recommended to be eliminated in order to achieve the optimal operation of the ECI-block.

7.5.2 Timing Tests

The measured timing values of the I^{2}C-bus include:

- clock frequency

- clock low time
- clock high time
- data setup and hold time for the master and slave individually
- start condition hold time
- repeated start condition setup time
- stop condition setup time

The timing tests are done only to I^{2}C-protocol. Figure 7.15 presents the definitions for the timing tests and the measurement results are presented in Table 7.4. Instead of “repeated start”-commands, the programmer has used separate “start”- and “stop”-commands. Both methods are equally good and the selection of which one to use does not affect the functionality of the data bus. With the exception of the “repeated start” all the defined timings are measured. [2]

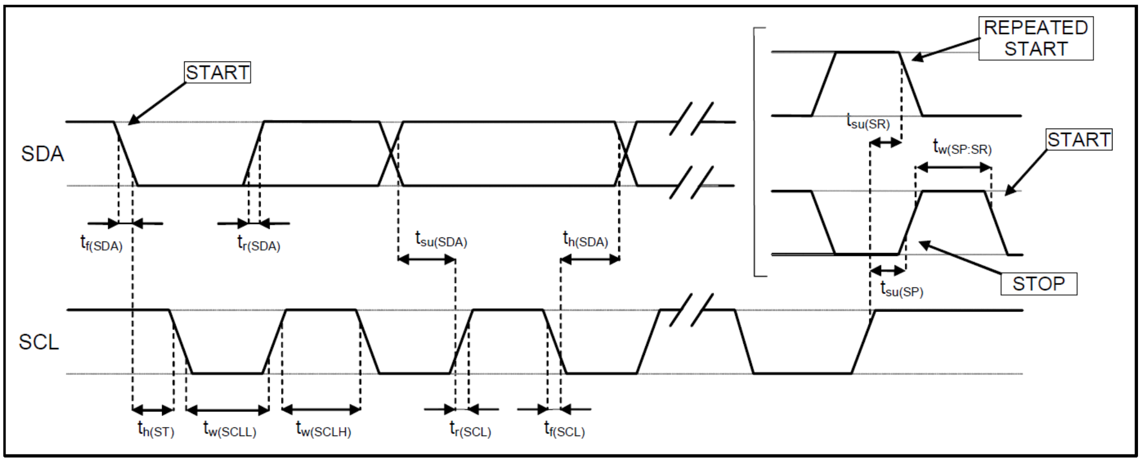
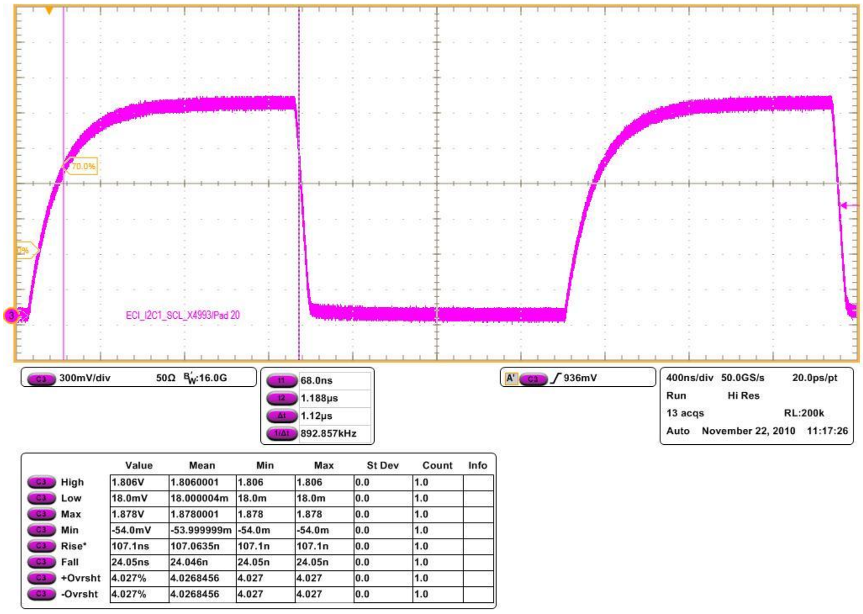


Figure 7.15. *Definition of timing for High-speeds-mode devices on the I^{2}C-bus [2]*

Table 7.4. *Electrical characteristics of the I^2C-timing measurements [2]*

Measured parameter	Min	Typical	Max	Unit	Measured Values	Comments
Timings						
SCL clock frequency, $f_{(SCL)}$			400	kHz	**391.5**	PASSED
SCL clock low time, $t_{w(SCLL)}$	1.3			µs	**1.3**	PASSED
SCL clock high time, $t_{w(SCLH)}$	0.6			µs	**1.12**	PASSED
SDA setup time, $t_{su(SDA)}$ - The main processor is sending - ATtiny20 is sending	100			ns	 **833** **673**	 PASSED PASSED
SDA data hold time, $t_{h(SDA)}$ - The main processor is sending - ATtiny20 is sending	0		900	ns	 **36** **329**	 PASSED PASSED
START condition hold time, $t_{h(ST)}$	0.6			µs	**0.910**	PASSED
Repeated START condition setup time, $t_{su(SR)}$	0.6			µs	**NA**	NA
STOP condition setup time, $t_{su(SP)}$	0.6			µs	**1.19**	PASSED

Figure 7.16 presents the I^2C_SCL-signal high time and the Figure 7.17 presents the clock low time. The low time is shorter than the high time due to the RC time constant. The bigger the time constant, the shorter the high time. The high and low times are connected to rise and fall time definitions, but also to a duty cycle. If the duty cycle was off enough from the normal 50%, the high and low time tests would also fail. [7, p. 28]



C3 300mV/div 50Ω BW:16.0G

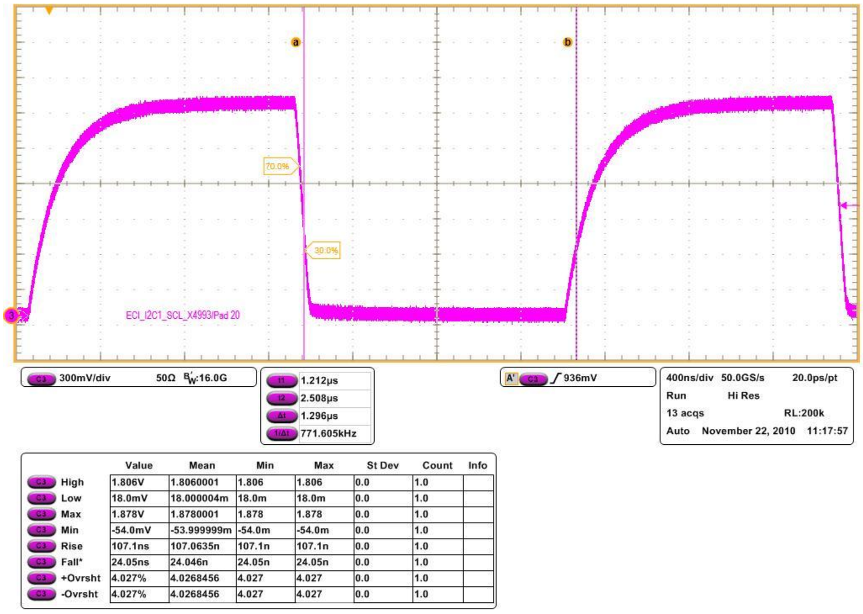
t1 68.0ns
t2 1.188µs
Δt 1.12µs
1/Δt 892.857kHz

A' C3 ∫ 936mV

400ns/div 50.0GS/s 20.0ps/pt
Run Hi Res
13 acqs RL:200k
Auto November 22, 2010 11:17:26

	Value	Mean	Min	Max	St Dev	Count	Info
C3 High	1.806V	1.8060001	1.806	1.806	0.0	1.0	
C3 Low	18.0mV	18.000004m	18.0m	18.0m	0.0	1.0	
C3 Max	1.878V	1.8780001	1.878	1.878	0.0	1.0	
C3 Min	-54.0mV	-53.999999m	-54.0m	-54.0m	0.0	1.0	
C3 Rise*	107.1ns	107.0635n	107.1n	107.1n	0.0	1.0	
C3 Fall	24.05ns	24.046n	24.05n	24.05n	0.0	1.0	
C3 +Ovrsht	4.027%	4.0268456	4.027	4.027	0.0	1.0	
C3 -Ovrsht	4.027%	4.0268456	4.027	4.027	0.0	1.0	

Figure 7.16. *I^{2}C_SCL-signal high time*



C3 300mV/div 50Ω BW:16.0G

t1 1.212µs
t2 2.508µs
Δt 1.296µs
1/Δt 771.605kHz

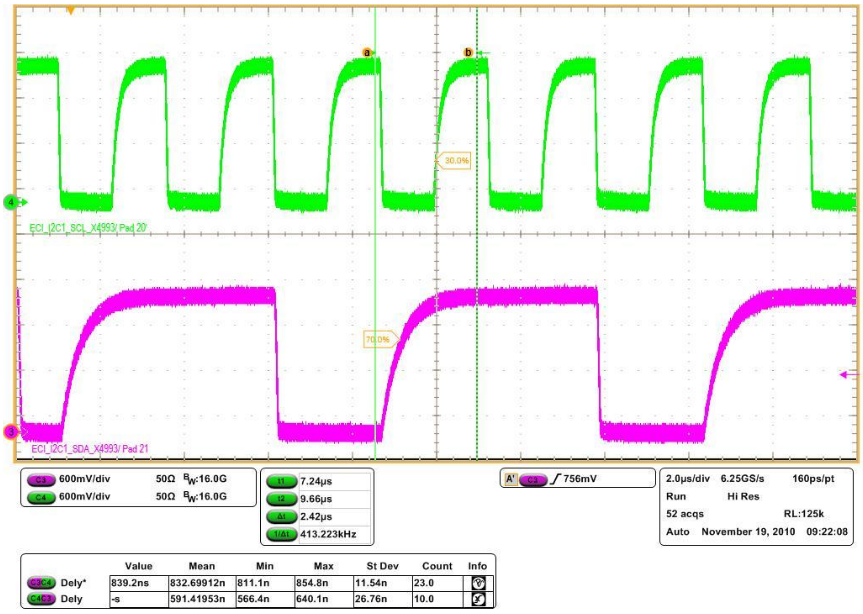
A' C3 ∫ 936mV

400ns/div 50.0GS/s 20.0ps/pt
Run Hi Res
13 acqs RL:200k
Auto November 22, 2010 11:17:57

	Value	Mean	Min	Max	St Dev	Count	Info
C3 High	1.806V	1.8060001	1.806	1.806	0.0	1.0	
C3 Low	18.0mV	18.000004m	18.0m	18.0m	0.0	1.0	
C3 Max	1.878V	1.8780001	1.878	1.878	0.0	1.0	
C3 Min	-54.0mV	-53.999999m	-54.0m	-54.0m	0.0	1.0	
C3 Rise	107.1ns	107.0635n	107.1n	107.1n	0.0	1.0	
C3 Fall*	24.05ns	24.046n	24.05n	24.05n	0.0	1.0	
C3 +Ovrsht	4.027%	4.0268456	4.027	4.027	0.0	1.0	
C3 -Ovrsht	4.027%	4.0268456	4.027	4.027	0.0	1.0	

Figure 7.17. *I^{2}C_SCL-signal low time*

Figure 7.18 presents the setup time of the I^{2}C_SDA-signal when the main processor is driving the bus. Setup time describes the point in time the data must be at a valid logic level before the clock arrives. The state of the data signal is read on the leading edge of the clock, so the data signal must have changed the state before that in order to be read correctly. Figure 7.19 presents the setup time of the I^{2}C_SDA-signal when the microcontroller is driving the bus.



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

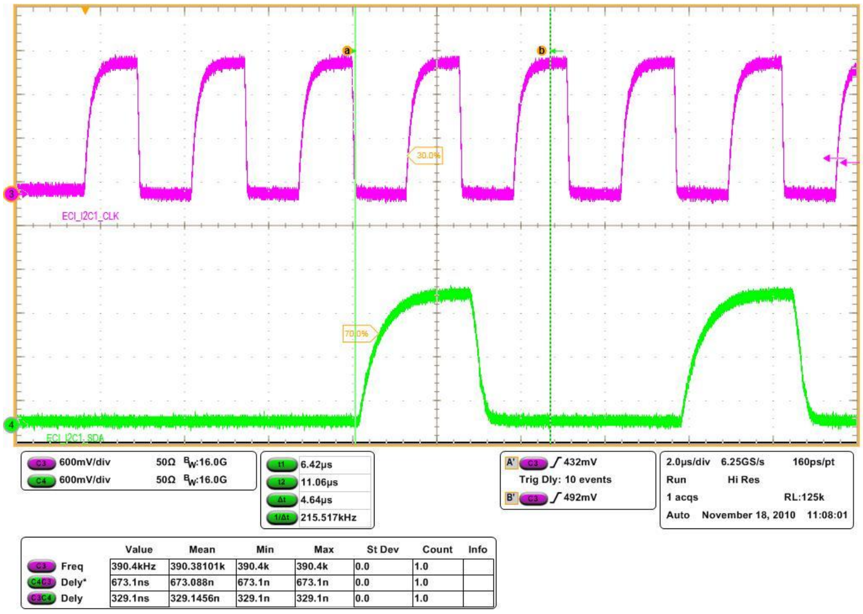
t1 7.24µs
t2 9.66µs
Δt 2.42µs
1/Δt 413.223kHz

A' C3 756mV

2.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
52 acqs RL:125k
Auto November 19, 2010 09:22:08

	Value	Mean	Min	Max	St Dev	Count	Info
C3C4 Dely*	839.2ns	832.69912n	811.1n	854.8n	11.54n	23.0	
C4C3 Dely	-s	591.41953n	566.4n	640.1n	26.76n	10.0	

Figure 7.18. *I^{2}C_SDA-signal setup time (write)*



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

t1 6.42µs
t2 11.06µs
Δt 4.64µs
1/Δt 215.517kHz

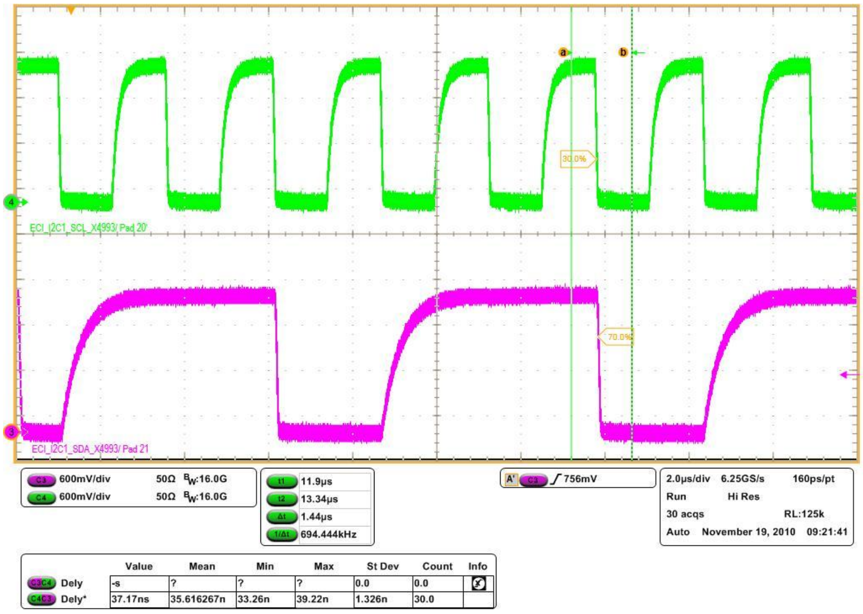
A' C3 432mV
Trig Dly: 10 events
B' C3 492mV

2.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
1 acqs RL:125k
Auto November 18, 2010 11:08:01

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Freq	390.4kHz	390.38101k	390.4k	390.4k	0.0	1.0	
C4C3 Dely*	673.1ns	673.088n	673.1n	673.1n	0.0	1.0	
C3C4 Dely	329.1ns	329.1456n	329.1n	329.1n	0.0	1.0	

Figure 7.19. *I^{2}C_SDA-signal setup time (read)*

Figure 7.20 presents the hold time of the I^{2}C_SDA-signal when the main processor is driving the bus. Hold time specifies the moment when the data can change after it has been sampled by the device. Figure 7.21 presents the hold time of the I^{2}C_SDA-signal when the microcontroller is driving the bus.



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

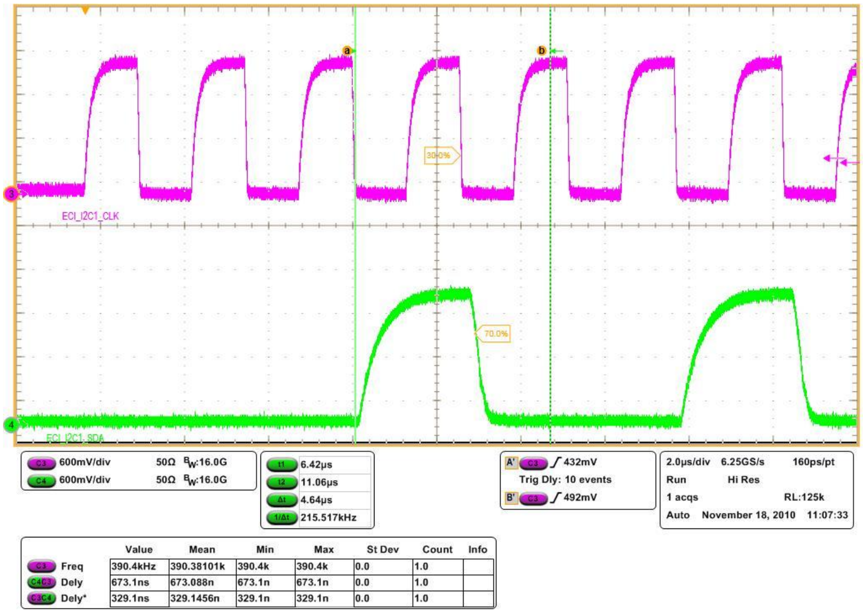
t1 11.9µs
t2 13.34µs
Δt 1.44µs
1/Δt 694.444kHz

A' C3 756mV

2.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
30 acqs RL:125k
Auto November 19, 2010 09:21:41

	Value	Mean	Min	Max	St Dev	Count	Info
C3C4 Dely	-s	?	?	?	0.0	0.0	
C4C3 Dely*	37.17ns	35.616267n	33.26n	39.22n	1.326n	30.0	

Figure 7.20. *I^{2}C_SDA-signal hold time (write)*



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

t1 6.42µs
t2 11.06µs
Δt 4.64µs
1/Δt 215.517kHz

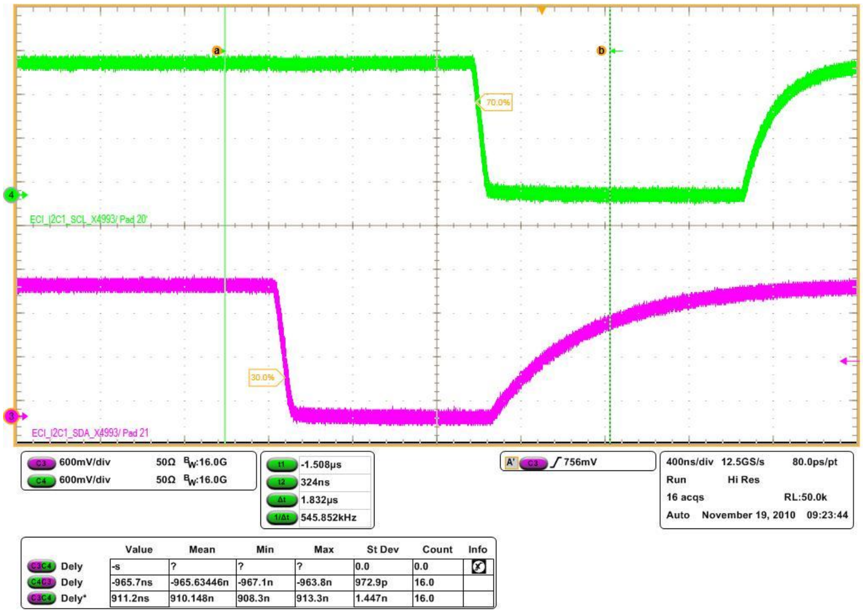
A' C3 432mV
Trig Dly: 10 events
B' C3 492mV

2.0µs/div 6.25GS/s 160ps/pt
Run Hi Res
1 acqs RL:125k
Auto November 18, 2010 11:07:33

	Value	Mean	Min	Max	St Dev	Count	Info
C3 Freq	390.4kHz	390.38101k	390.4k	390.4k	0.0	1.0	
C4C3 Dely	673.1ns	673.088n	673.1n	673.1n	0.0	1.0	
C3C4 Dely*	329.1ns	329.1456n	329.1n	329.1n	0.0	1.0	

Figure 7.21. *I^{2}C_SDA-signal hold time (read)*

Figure 7.22 presents the “start”-condition and Figure 7.23 presents the “stop”-condition for the I^{2}C-bus. The START and STOP bits are defined as rising or falling edges on the data line while the clock line is kept high.



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

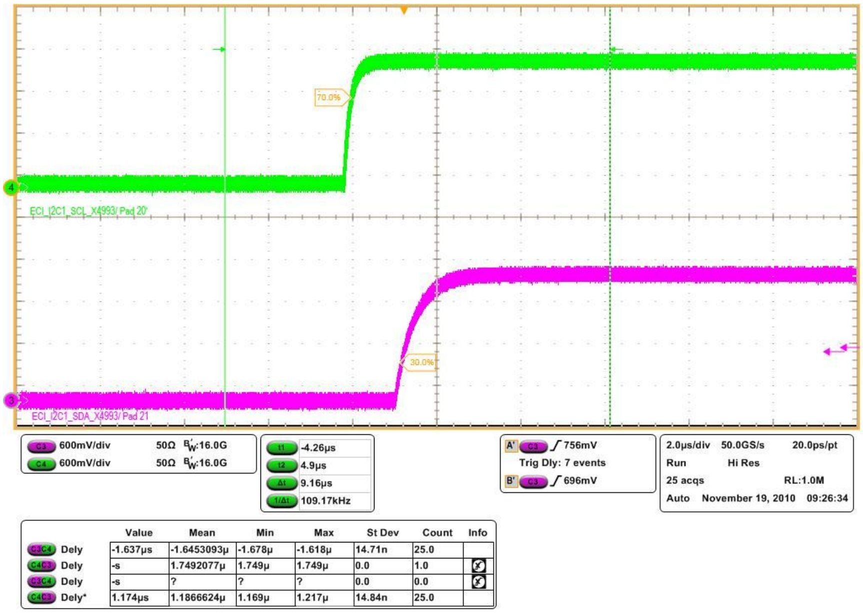
t1 -1.508µs
t2 324ns
Δt 1.832µs
1/Δt 545.852kHz

A' C3 756mV

400ns/div 12.5GS/s 80.0ps/pt
Run Hi Res
16 acqs RL:50.0k
Auto November 19, 2010 09:23:44

	Value	Mean	Min	Max	St Dev	Count	Info
C3C4 Dely	-s	?	?	?	0.0	0.0	☒
C4C3 Dely	-965.7ns	-965.63446n	-967.1n	-963.8n	972.9p	16.0	
C3C4 Dely*	911.2ns	910.148n	908.3n	913.3n	1.447n	16.0	

Figure 7.22. *START condition hold time*



C3 600mV/div 50Ω BW:16.0G
C4 600mV/div 50Ω BW:16.0G

t1 -4.26µs
t2 4.9µs
Δt 9.16µs
1/Δt 109.17kHz

A' C3 756mV
Trig Dly: 7 events
B' C3 696mV

2.0µs/div 50.0GS/s 20.0ps/pt
Run Hi Res
25 acqs RL:1.0M
Auto November 19, 2010 09:26:34

	Value	Mean	Min	Max	St Dev	Count	Info
C3C4 Dely	-1.637µs	-1.6453093µ	-1.678µ	-1.618µ	14.71n	25.0	
C4C3 Dely	-s	1.7492077µ	1.749µ	1.749µ	0.0	1.0	☒
C3C4 Dely	-s	?	?	?	0.0	0.0	☒
C4C3 Dely*	1.174µs	1.1866624µ	1.169µ	1.217µ	14.84n	25.0	

Figure 7.23. *STOP condition setup time*

All the timing measurements are well within the limits set by the I^2C specification. There is no need for making any adjustments or modifications to the schematic design. The software will be developed further and optimized. Hence, all the tests must be repeated when the software is finished. Changes in the software are not expected to cause any significant changes to signal integrity. The signal timing measurements, however, may and probably will produce totally different results. That is why the results of the timing measurements should be treated with caution.

8 CONCLUSIONS

The goal of this thesis was to design the schematic for the ECI-block. The entrustment also included verification and testing of the design. The target remained unchanged throughout the writing process with one exception; the support for the AHJ was cancelled. The solution did not have much significance regarding the outcome as it was an improvement or even a relief especially from the EMC filtering point of view. The solution also simplified the design and reduced the need for testing.

The most demanding and time spending tasks were to take the mass production and the future aspect into consideration. It was never an option to make a prototype which probably works after some adjustments. The designed schematic must be functional when it comes off the production line and stay that way even in challenging temperatures. The worst-case scenario for every imaginable operating condition must be taken into consideration. Component properties, combined effects of different components and temperature variations and so on, must always be taken into consideration.

All the tests were done with only one phone terminal as there were no more available and because of the minimal quality variances between different phone terminals. Also only one ECI-accessory went through the electrical testing as the point was not to test the accessories, but the phone terminal. On the other hand, the functionality of several non-ECI-accessories and ECI-accessories were tested in order to verify the correct functionality of the designed circuit and the software.

The software of the microcontroller or the main processor, for that matter, was not ready when the tests were carried out. All the tested accessories functioned as they were supposed to, even though the software used for testing was unstable at times. The instability slowed the testing and made performing recurrent measurements complicated. Eventually, the measurements showed that everything concerning the electrical design functioned as desired.

The performance tests were skipped due to the problems not concerning the electrical design. The ECI cannot be guaranteed to work with absolute certainty before all the necessary tests have been performed. On the basis of the gathered experiences from the similar designs done in the past and the measurements done with current design in room temperature, it is highly unlikely that high or low temperatures would cause any problems for the functionality of the ECI.

The ECI-block itself can be considered as an addition or a patch to the audio interface. The designed circuit was meant to be useful not only in this very design, but also in the similar environments in the future. The designed circuit does not have a significant negative effect to the functionality of the audio interface, where the ECI-block is attached to. The ECI-block can easily be removed from the design without affecting to the functionality of the audio interface or added to any design not having the ECI implemented yet. Therefore, it can be stated that the objective of future usefulness was fulfilled perfectly.

If the same concept of using commercial chipsets as a base of the phone terminal is used in the future, the design provides a great starting point for the ECI implementation in future products. Now that the design has once been done and found to be working, further development is much easier and less risky because it is always possible to look back and use the principals of the already designed circuit. The most probable future scenario is to implement the design into an ASIC. Hence, the price and the needed space from the circuit board would be even lower.

REFERENCES

[1] Nokia, “Nokia Online Store finder”. http://europe.nokia.com/buy-online, (referred 11.03.2011)

[2] NXP, “UM10204, I^{2}C-bus specification and user manual, Rev. 03 – 19.07.2007”. http://www.nxp.com/documents/user_manual/UM10204.pdf, (referred 11.03.2011)

[3] NXP, “AN10216-01 I^{2}C manual”. http://www.nxp.com/documents/application_note/AN10216.pdf, (referred 11.03.2011)

[4] Open Mobile Terminal Platform. http://www.omtp.org/, (referred 11.03.2011)

[5] Telos, “I^{2}C-Bus: What's that?”. http://www.i2c-bus.org/, (referred 11.03.2011)

[6] Atmel, “ATtiny20 microcontroller datasheet”. http://www.atmel.com/dyn/resources/prod_documents/doc8235.pdf, (referred 11.03.2011)

[7] Paul Horowitz, Winfield Hill. The Art of Electronics, 2nd edition. Cambridge University Press, 1989

[8] ECE lab, “Tri-State Output Circuit” http://www.ecelab.com/tri-state-output.htm, (referred 06.04.2011)

[9] Adel S. Sedra, Kenneth C. Smith. Microelectronic Circuits. CBS College Publishing, 1987

[10] Howard W. Johnson, Martin Graham. High-speed Digital Design. Prentice Hall PTR, 1993

[11] Tim Williams. EMC for Product Designers, 2nd edition. Reed Educational and Professional Publishing Ltd, 1996

[12] James E. Buchanan. Signal and Power Integrity in Digital Systems: TTL, CMOS, and BiCMOS. McGraw-Hill, 1996

[13] Jan M. Rabaey, Anantha Chandrakasan, Borivoje Nikolić. Digital Integrated Circuits; A Design perspective,2nd edition. Pearson Education, Inc. 2003

[14] Martin O´Hara. EMC at Component and PCB Level, Newnes, 1998

[15] Ned Mohan, Tore. Undeland, William P. Robbins. Power Electronics; Converters, Applications and Design, 3rd edition. John Wiley & Sons, Inc. 2003

[16] Jacob Millman, Arvin Grabel, Microelectronics, 2nd edition. McGraw-Hill Book Company, 1988

[17] Atmel, "AVR040: EMC Design Considerations". http://www.atmel.com/dyn/resources/prod_documents/doc1619.pdf, (referred 15.03.2011)

[18] Atmel, "AVR042: AVR Hardware Design Considerations". http://www.atmel.com/dyn/resources/prod_documents/doc2521.pdf, (referred 15.03.2011)

[19] PMR280UN datasheet. http://www.nxp.com/documents/data_sheet/PMR280UN.pdf, (referred 15.03.2011)

[20] BKP1005TM121-T datasheet. http://www.datasheetdir.com/BKP1005TM121-T+download, (referred 15.03.2011)

[21] All About Circuits. http://www.allaboutcircuits.com/vol_4/chpt_3/3.html, (referred 15.03.2011)

[22] Vishay, "ILB, ILBB Ferrite Beads". http://www.vishay.com/docs/ilb_ilbb_enote.pdf, (referred 15.03.2011)

[23] Neil Storey. Electronics; A System Approach, 2nd edition. Addison Wesley Longman Ltd, 1998

[24] SN74AUP2G07 datasheet http://focus.ti.com/docs/prod/folders/print/sn74aup2g07.html, (referred 15.03.2011)

[25] SN74AUP1G07 datasheet http://focus.ti.com/lit/ds/symlink/sn74aup1g07.pdf, (referred 15.03.2011)

[26] SN74LVC1G07 datasheet http://focus.ti.com/lit/ds/symlink/sn74lvc1g07-ep.pdf, (referred 15.03.2011)

[27] Wikipedia, "RC circuit". http://en.wikipedia.org/wiki/RC_circuit, (referred 15.03.2011)

[28] Tektronix, "ABCs of Probes". http://www.tek.com/Measurement/programs/301892X312611/?lc=EN&PRODUCT=&returnUrl=ct=TI&cs=pri&ci=2329&lc=EN, (available 8.05.2011)

[29] NXP, "A guide to designing for ESD and EMC". http://www.nxp.com/documents/application_note/AN10897.pdf, (referred 15.03.2011)

[30] Techterms.com.
http://www.techterms.com/definition/protocol,
(referred 15.03.2011)

[31] Xilinx, "FPGA vs. ASIC".
http://www.xilinx.com/company/gettingstarted/fpgavsasic.htm,
(referred 15.03.2011)

[32] Tektronix, "P6245 data sheet".
http://www2.tek.com/cmsreplive/psrep/13504/60W_13989_4_2011.01.11.15.14.12_13504_EN.pdf, (referred 15.03.2011)

[33] SN74LVC1G3157 datasheet,
http://focus.ti.com/lit/ds/symlink/sn74lvc1g3157.pdf,
(referred 15.03.2011)

[34] Texas Instruments, "Logic Guide".
http://focus.ti.com/lit/sg/slyb125a/slyb125a.pdf,
(referred 15.03.2011)

[35] NXP, "Level shifting techniques in I2C-bus design".
http://www.nxp.com/documents/application_note/AN10441.pdf,
(referred 08.05.2011)

[36] NXP, "Interface protection products selection guide".
http://www.nxp.com/documents/selection_guide/75016055.pdf,
(referred 08.05.2011)

[37] US6K4 datasheet.
http://www.rohm.com/products/discrete/transistor/complex_mos-mos/us6k4/,
(referred 08.05.2011)

[38] 74LX1G04 datasheet.
http://www.st.com/stonline/books/pdf/docs/8341.pdf,
(referred 08.05.2011)

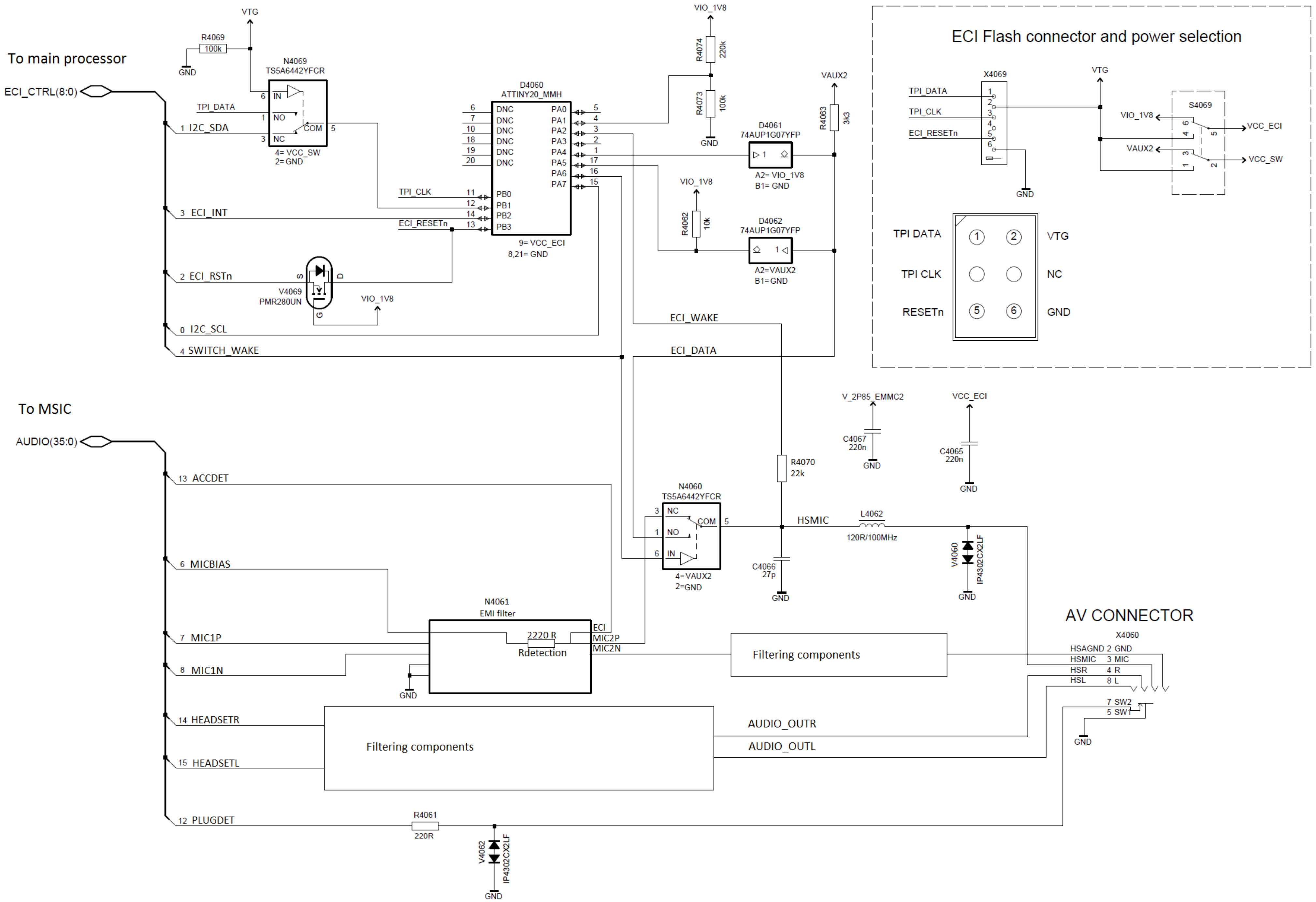
[39] Tektronix ”Fundamentals of Signal Integrity".
http://www.tek.com/Measurement/App_Notes/55_15465/55W_15465_4.pdf,
(referred 08.05.2011)

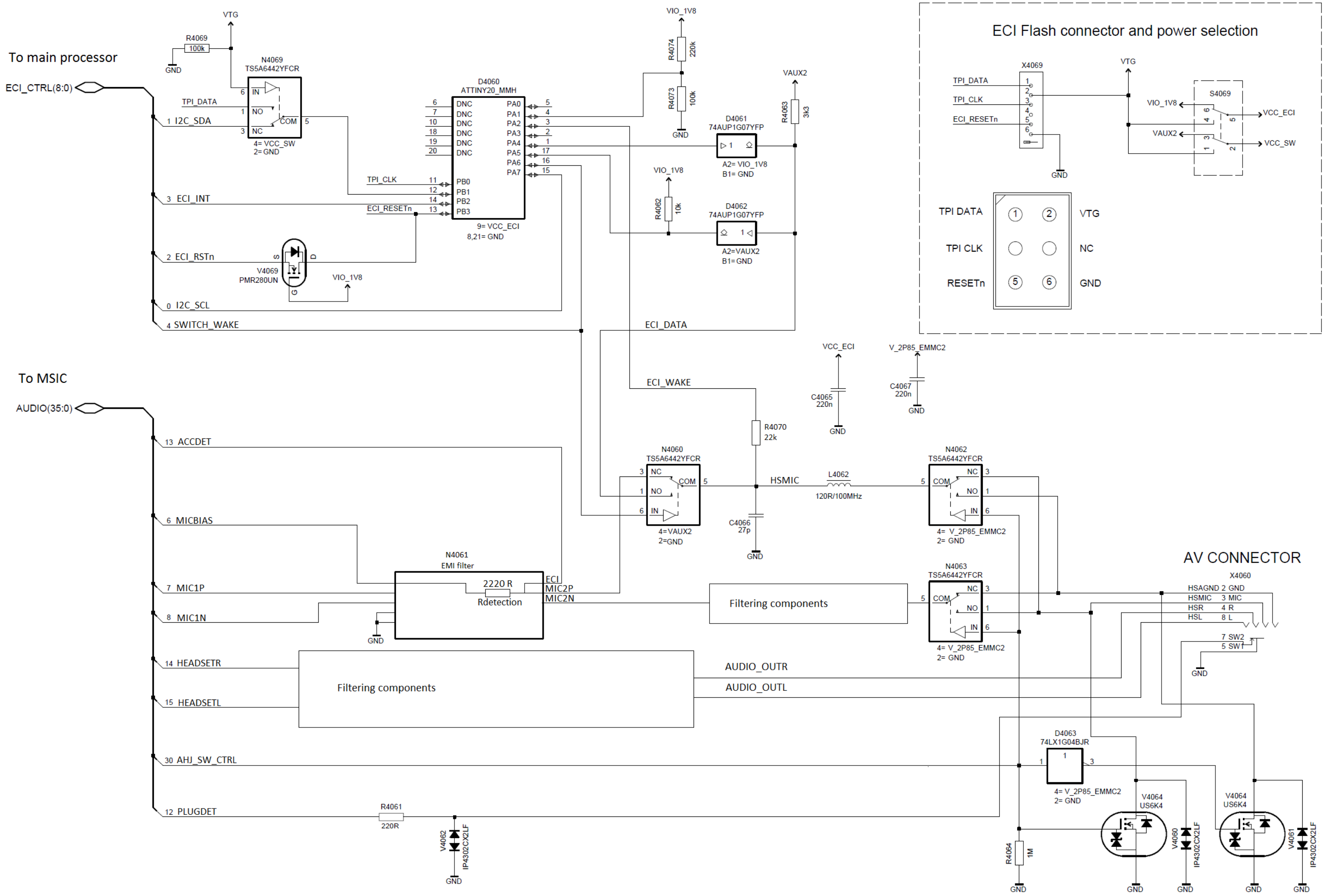
[40] P7240 "User Manual".
http://www2.tek.com/cmswpt/madetails.lotr?ct=MA&cs=mur&ci=8980&lc=EN,
(referred 08.05.2011)

APPENDICES

APPENDIX A: Schematic without the AHJ support, 1 page

APPENDIX B: Schematic including the AHJ support, 1 page


To main processor
ECI_CTRL(8:0)
1 I2C_SDA
3 ECI_INT
2 ECI_RSTn
0 I2C_SCL
4 SWITCH_WAKE
R4069
100k
GND
VTG
N4069
TS5A6442YFCR
IN
NO
NC
COM
4= VCC_SW
2= GND
TPI_DATA
V4069
PMR280UN
S
D
G
VIO_1V8
D4060
ATTINY20_MMH
DNC
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
9= VCC_ECI
8,21= GND
TPI_CLK
ECI_RESETn
R4074
220k
R4073
100k
VIO_1V8
GND
R4062
10k
D4061
74AUP1G07YFP
A2= VIO_1V8
B1= GND
D4062
74AUP1G07YFP
A2=VAUX2
B1=GND
VAUX2
R4063
3k3
ECI_WAKE
ECI_DATA
ECI Flash connector and power selection
X4069
TPI_DATA
TPI_CLK
ECI_RESETn
VTG
S4069
VIO_1V8
VAUX2
VCC_ECI
VCC_SW
GND
TPI DATA
TPI CLK
RESETn
VTG
NC
GND
1
2
5
6
To MSIC
AUDIO(35:0)
13 ACCDET
6 MICBIAS
7 MIC1P
8 MIC1N
14 HEADSETR
15 HEADSETL
12 PLUGDET
V_2P85_EMMC2
C4067
220n
VCC_ECI
C4065
220n
R4070
22k
N4060
TS5A6442YFCR
NC
NO
IN
COM
4=VAUX2
2=GND
HSMIC
L4062
120R/100MHz
C4066
27p
V4060
IP4302CX2LF
N4061
EMI filter
ECI
MIC2P
MIC2N
2220 R
Rdetection
Filtering components
AV CONNECTOR
X4060
HSAGND 2 GND
HSMIC 3 MIC
HSR 4 R
HSL 8 L
7 SW2
5 SW1
AUDIO_OUTR
AUDIO_OUTL
Filtering components
R4061
220R
V4062
IP4302CX2LF
GND


To main processor
ECI_CTRL(8:0)
1 I2C_SDA
3 ECI_INT
2 ECI_RSTn
0 I2C_SCL
4 SWITCH_WAKE
R4069
100k
GND
VTG
N4069
TS5A6442YFCR
IN
NO
NC
COM
TPI_DATA
4= VCC_SW
2= GND
V4069
PMR280UN
VIO_1V8
D4060
ATTINY20_MMH
DNC
PA0
PA1
PA2
PA3
PA4
PA5
PA6
PA7
PB0
PB1
PB2
PB3
TPI_CLK
ECI_RESETn
9= VCC_ECI
8,21= GND
VIO_1V8
R4074
220k
R4073
100k
GND
VAUX2
R4063
3k3
D4061
74AUP1G07YFP
A2= VIO_1V8
B1= GND
R4062
10k
D4062
74AUP1G07YFP
A2=VAUX2
B1=GND
ECI_DATA
ECI_WAKE
ECI Flash connector and power selection
X4069
TPI_DATA
TPI_CLK
ECI_RESETn
GND
VTG
S4069
VIO_1V8
VAUX2
VCC_ECI
VCC_SW
TPI DATA
TPI CLK
RESETn
VTG
NC
GND
VCC_ECI
C4065
220n
V_2P85_EMMC2
C4067
220n
To MSIC
AUDIO(35:0)
13 ACCDET
6 MICBIAS
7 MIC1P
8 MIC1N
14 HEADSETR
15 HEADSETL
30 AHJ_SW_CTRL
12 PLUGDET
R4070
22k
N4060
TS5A6442YFCR
4=VAUX2
2=GND
HSMIC
L4062
120R/100MHz
C4066
27p
N4062
TS5A6442YFCR
4= V_2P85_EMMC2
2= GND
N4061
EMI filter
2220 R
Rdetection
ECI
MIC2P
MIC2N
Filtering components
N4063
TS5A6442YFCR
4= V_2P85_EMMC2
2= GND
AV CONNECTOR
X4060
HSAGND 2 GND
HSMIC 3 MIC
HSR 4 R
HSL 8 L
7 SW2
5 SW1
Filtering components
AUDIO_OUTR
AUDIO_OUTL
D4063
74LX1G04BJR
4= V_2P85_EMMC2
2= GND
V4064
US6K4
V4064
US6K4
V4060
IP4302CX2LF
V4061
IP4302CX2LF
R4064
1M
R4061
220R
V4062
IP4302CX2LF