



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projekt

Department Fahrzeugtechnik und Flugzeugbau

Development of a System for Signal Analysis and Location in the Aircraft

Justin Müller

15. Februar 2011

Abstract

The aim of the overall project is to develop a device that is able to do an automatic location of components inside an aircraft cabin. This requires at least three measuring points. In this project work, an electronic system for range measuring only was developed. For the range detection, ultrasonic transducers were deployed, where the varying traveling time of sound over range was utilized to measure a range. The system consists of two devices: one to monitor and control measures, and one that answers the ultrasonic pulse from the first device.

The work includes the circuit design, electronic board design, manufacturing, firmware programming and testing of the devices. The development, design and manufacturing were accomplished at the Hamburg University of Applied Sciences, mostly in the Department of Electrical Engineering and Computer Science. The testing was done in the Maintenance Technologies Test Center (MTTC) at Airbus in Hamburg/Finkenwerder.

The system satisfies its development objective by reaching an operating distance of 6.5 meters. The accuracy of measurement at this range is about +/- 5 cm, which is not precise enough to do an exact locating within the inch-grid, where most of cabin interior parts are attached to. At smaller ranges, the accuracy is within 1 cm.

fW

7 ca a YfWU'i gY'ghf]Wmidfc\]V]hYX"

Mci f'fYei Yghia UmVY'X]f YWYX'hc.

**DfcZ'8 f"! =b["8]YhYf'GW c`nžA GA 9
9!A U]`gYY. \ ltd.#k k k 'DfcZGW c`n'XY**

**8 ck b`cUX'h]g'Z`Y`Zca .
\ ltd.#6]V]cH Y_ 'DfcZGW c`n'XY**



PAHMIR



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

DEPARTMENT FAHRZEUGTECHNIK UND FLUGZEUGBAU

Development of a system for signal analysis and location detection in the aircraft

Definition of a *project* according to examination regulations.

Background

The aim of the project PAHMIR (Preventive Aircraft Health Monitoring for Integrated Re-configuration) is to design a new approach for location and failure detection. Both tasks need a hardware platform to perform calculations, sensing and communication. The location detection and the sensing shall be implemented as a prototype system to evaluate the accuracy and to perform calibration of the algorithms. A prototype is needed to be able to verify the concepts.

Task

The task is to develop a device, which contains an RFID and an ultrasonic sound transmitter, interfaces for sensors and memory as well as a programmable processor for signal analysis. The design should use the available concepts and designs from PAHMIR and the EVB (Electronic Vibration Box). At the end a prototype system should be available that implements the concepts.

The specific tasks are:

- Evaluate the EVB design and the PAHMIR location detection design.
- Enhance the project with hardware components (processors, transmitter etc.) for a prototype
- Draw a circuit design.
- Built a prototype system.

The report has to be written in English or German based on German or international standards on report writing.

Content

Abstract	2
Content	4
List of Figures	6
List of Tables.....	7
Nomenclature	8
Subscript.....	8
List of Abbreviations.....	9
Terms and Definitions.....	10
1 Introduction	11
1.1 Motivation.....	11
1.2 Objectives	11
1.3 Literature.....	11
1.4 Report Structure	12
2 Technical Fundamentals	13
2.1 Ultrasound.....	13
2.2 Localization by Trilateration.....	13
2.3 Physical Sound Situation inside Aircraft Cabins	14
3 Project Concepts	15
3.1 Ultra Sound Signals	15
3.2 Concept of the electronic Circuit	16
3.3 User Interfaces	18
3.4 Software Concept.....	18
4 Design.....	20
4.1 Choice of essential Parts	20
4.2 Voltage Supply.....	22
4.3 Oscillator.....	23
4.4 Integrated Circuits (IC)	23
4.5 Transistors.....	23
4.6 Capacitors	23
4.7 Resistors.....	25
4.8 Diodes	26
4.9 Non-dialectical components.....	27
5 Circuit Layout.....	28
5.1 Circuit Schematic Overview	28
5.2 ICSP	30
5.3 Transmitting.....	30
5.4 Receiving	31
5.5 Digitizing	32
5.6 Status LED / LCD	33
5.7 Board Layout	33
6 Manufacturing	35

6.1	Etching	35
6.2	Soldering	38
6.3	Programming.....	40
6.4	Generating the Signal.....	41
7	Testing.....	41
7.1	Configuration of Timer0	41
7.2	Finding of Constants	42
7.3	Test Arrangement.....	43
	Conclusion.....	47
	Acknowledgements.....	48
	References	49
	Appendix A Main Source Code.....	50
	Appendix B Source Code of Sub Functions.....	55
	Source Code of Sub Functions	55
	Appendix C Quick Operation Manual.....	61

List of Figures

Figure 2.1	Trilateration (Principle overtaken from Gerdes 2010)	13
Figure 3.1	Illustration of overall ultrasound project concept	16
Figure 3.2	Electronic circuit concept - active device	17
Figure 3.3	Electronic circuit concept - passive device	17
Figure 3.4	Software concept - active device.....	19
Figure 3.4	Software concept - passive device	20
Figure 4.1	Essential Components	22
Figure 4.2	Oscillator image of ultrasonic pulse.....	24
Figure 4.3	Oscillator image of ultrasonic pulse received by receiver	25
Figure 5.1	Circuit schematic overview	29
Figure 5.2	Circuit schematic detail: ICSP	30
Figure 5.3	Circuit schematic detail: transmitting	31
Figure 5.4	Circuit schematic detail: receiving.....	31
Figure 5.5	Circuit schematic detail: digitizing	32
Figure 5.6	Circuit schematic detail: optics	33
Figure 5.7	Board layout	34
Figure 6.1	PCB exposure machine	35
Figure 6.2	PCB in photo developer liquid	36
Figure 6.3	Test exposure (the protect folia failed to release from the board here).....	36
Figure 6.4	PCB etching machine	37
Figure 6.5	Ready etched circuit board.....	37
Figure 6.6	Finished active device - top.....	38
Figure 6.7	Finished active device - bottom	39
Figure 6.8	Finished passive device - top	39
Figure 6.9	Receiver / transmitter combination in frame.....	40
Figure 7.1	Test arrangement	43
Figure 7.2	Result of test distance measurements shown in a diagram	45
Figure 7.3	Evaluating of the system in the MTTC	46
Figure C.1	Board configuration.....	61
Figure C.2	Connection of equipment	62

List of Tables

Table 3.1	Required components & techniques.....	15
Table 4.1	Item Design Criteria	21
Table 4.2	Choice of essential parts.....	21
Table 4.3	Choice of Integrated Circuits	23
Table 4.4	Choice of capacitors	25
Table 4.5	Choice of resistors	26
Table 4.6	Choice of diodes	26
Table 4.7	Choice of non-dialectical components	27
Table 7.1	Result of test distance measurements.....	44

Nomenclature

d	distance
f	frequency
t	time
v	velocity

Subscript

meas.	measured
s	sound
un_high	unwanted high (frequency)
un_low	unwanted low (frequency)

List of Abbreviations

μ C	Microcontroller
CAD	Computer Aided Design
CCP	Capture Compare PWM (see also PWM)
CPU	Central Processing Unit
DIP	Dual Inline Package
I/O	Input / Output
IC	Integrated Circuit
ICSP	In Circuit Serial Programming
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MCLR	Memory Clear
MTTC	Maintenance Technology Test Center
PCB	Printed Circuit Board
PWM	Pulse Width Modulation
RAM	Random Input Output Memory
RFID	Radio Frequency Identification
ROM	Read Only Memory
SMD	Surface Mounted Device
URL	Universal Resource Locator
USB	Universal Serial Bus

Terms and Definitions

Microcontroller

A single chip that contains the processor (the CPU), non-volatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock and an I/O control unit. Microcontrollers come in all sizes and architectures. (**Sanchez 2007**)

Ultrasound

Ultrasound is sound at frequencies with more than 16 kHz. (**Kuttruff 1988**)

PAHMIR

PAHMIR (Preventive Aircraft Health Monitoring for Integrated Reconfiguration) is a research project of the HAW, Airbus Deutschland GmbH and Philotech. Its main goals are to gain more revenue for airlines and decrease the costs for maintenance and repairs. This shall be achieved by, among other things, the usage of new diagnosis systems such as this project work (**Scholz 2011**).

Port

This is the term for in- and output at an electronic interface. A port contains a varying number of single connection pins.

1 Introduction

1.1 Motivation

One often occurring error during mounting of cabin components at maintenance or cabin re-configuration is the wrong placement by mechanical applicants. This can potentially result in high extra costs since the components need to be removed and correctly reinstalled. In order to prevent misplaced components, a cheap and room saving solution, that detects the cabin configuration, can be expedient.

1.2 Objectives

The task is to develop a device, which contains an ultrasonic sound transmitter and receiver, interfaces for sensors and memory as well as a programmable processor for signal analysis. This includes the drawing of an electronic circuit and the building of a prototype. The second objective is to evaluate the system, whether it is appropriate to be applied inside an aircraft cabin.

1.3 Literature

The literature, that was used to develop the ultrasonic system, were datasheets for programming and code compiling for the microcontroller. The datasheets were downloaded from the webpage of "Microchip Technologies".

Furthermore, the book "The Microchip PIC" (**Sanchez 2007**) was used for some ideas concerning the programming.

Kuttruff 1988 was used as a source to gather some information about technical application of ultrasound.

1.4 Report Structure

- Chapter 2** contains background information about the physics of ultrasound and a mathematical solution of localization.
- Chapter 3** includes the description of the overall concept, as well as the preliminary design solutions for the hard- and software.
- Chapter 4** concretely describes the design of the schematics, including the choosing of parts and their values.
- Chapter 5** contains the design of the circuit schematic and the board layout.
- Chapter 6** the manufacturing process is presented here, mostly illustrated by photos. The programming of the firmware is also a part of manufacturing.
- Chapter 7** contains testing and calibrating of the system. Since the calibrating already requires the testing of the system, the testing was actually done in the same course.

2 Technical Fundamentals

2.1 Ultrasound

Ultrasound is sound at frequencies with more than 16 kHz. Also, ultrasound is defined as frequencies that are higher than the human ear could recognize. The older a human, the less high frequency can be heard by her or him. There is also a frequency band, which is *lower* than the normal level to be heard: infrasound.

Ultrasound is used in several technical applications. Its good reflecting properties makes it suitable for the localization of various things (holes, flaws, a fetus in a uterus, etc.). In air at room temperature, the speed of sound is around $330 \frac{m}{s}$ (Kuttruff 1988).

In order to calculate a distance, the velocity of sound (v_s) needs to be known¹. The velocity depends on the medium and its temperature, in which a sound wave is moving.

2.2 Localization by Trilateration

Trilateration is an algorithm, which gives the position of an object in a two dimensional area. It requires three reference points (a, b, c) of which the arrangement coordinates ($a_x, a_y, b_x, b_y, c_x, c_y$) are known parameters. With this technique, the exact position of a point (x, y) between these references can be determined.

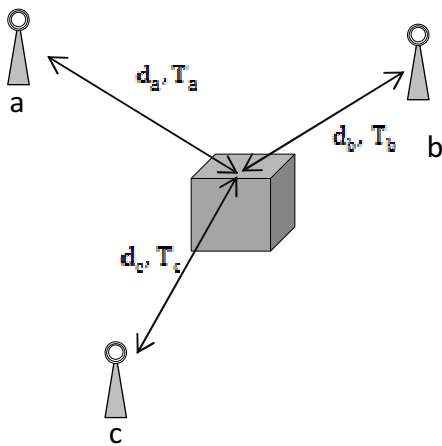


Figure 2.1 Trilateration (Principle overtaken from Gerdes 2010)

¹ As it is shown in this work in chapter 7, this is not absolutely necessary

For each of the distances the formula reads as:

$$\begin{aligned}d_a^2 &= (x - x_a)^2 + (y - y_a)^2 \\d_b^2 &= (x - x_b)^2 + (y - y_b)^2 \\d_c^2 &= (x - x_c)^2 + (y - y_c)^2\end{aligned}\tag{1.1}$$

If rearranged, the formulas can be solved for x and y :

$$\begin{aligned}d_a^2 - d_b^2 &= -2x \cdot (x_a - x_b) - 2y \cdot (y_a - y_b) + x_a^2 + y_a^2 - x_b^2 - y_b^2 \\d_a^2 - d_c^2 &= -2x \cdot (x_a - x_c) - 2y \cdot (y_a - y_c) + x_a^2 + y_a^2 - x_c^2 - y_c^2\end{aligned}\tag{1.2}$$

(Gerdes 2010)

For technical solutions, it is furthermore necessary to do an identification of the reference point. In other words, if the central device, whose location needs to be detected, sends a signal in order to calculate a range, only the corresponding reference point should be responding. This identification can, for example, be realized by sending data via the ultrasound signal, or the use of Radio Frequency Identification (RFID).

2.3 Physical Sound Situation inside Aircraft Cabins

Sound has the characteristic, to deflagrate on soft surfaces, or surfaces with small pores. This is why the application of ultrasonic based measure instruments can potentially cause problems in aircraft cabins, since there is mostly a carpet placed on the ground as well as many fabric upholstered seats. Also the wall panels are designed to deflagrate sound.

That is one reason why, for this project, instead of one passive device working by reflection, an active system with two or more devices is chosen.

3 Project Concepts

The conceptual design of the technical solution for the distance measuring device requires some reflections concerning the realization. Firstly, it is appropriate to write down (Table 3.1), which components are needed for which function. Also, it is a matter if this can be realized within the capabilities of a six month project that is being accomplished at the University of Applied Sciences in Hamburg.

Table 3.1 Required components & techniques

Function	Implemented by
Control electrical components	Microcontroller
Programming of microcontroller	USB serial programmer
Transmit ultra sound signal	Ultra Sound Transducer
Receive ultra sound signal	Ultra Sound Transducer
Carry electrical components	Circuit Board
Draw circuit schematic & board layout	CAD software (freeware)
Build Circuit Board	Etching Technique
Connect electrical components	Soldering
Sound proof test room	Present at the HAW

All of the listed components are common and easy to acquire. Indeed, there will be more than these, but those will most probably all be standard elements. The required techniques for testing and manufacturing are all in the scope of the HAW capabilities.

3.1 Ultra Sound Signals

Many technical solutions with ultrasound use the property of reflection. As already described in Chapter 2.3, an active system working with two devices is chosen due to too much interference in aircraft cabins.

At the overall concept of the arrangement, each of the devices has one ultrasonic transmitter and one ultrasonic receiver. One device, hereinafter named active device, or active Printed Circuit Board (PCB), will transmit an ultrasonic pulse. The pulse will reach the passive device after a certain amount of time. As a reaction of the input, it will answer with another pulse, which will need the same time to travel. The total traveling time plus hardware delays can be read out and recalculated into the corresponding distance by the active device. The traveling time depends on the distance between the two devices. The above described process is again illustrated by Figure 3.1.

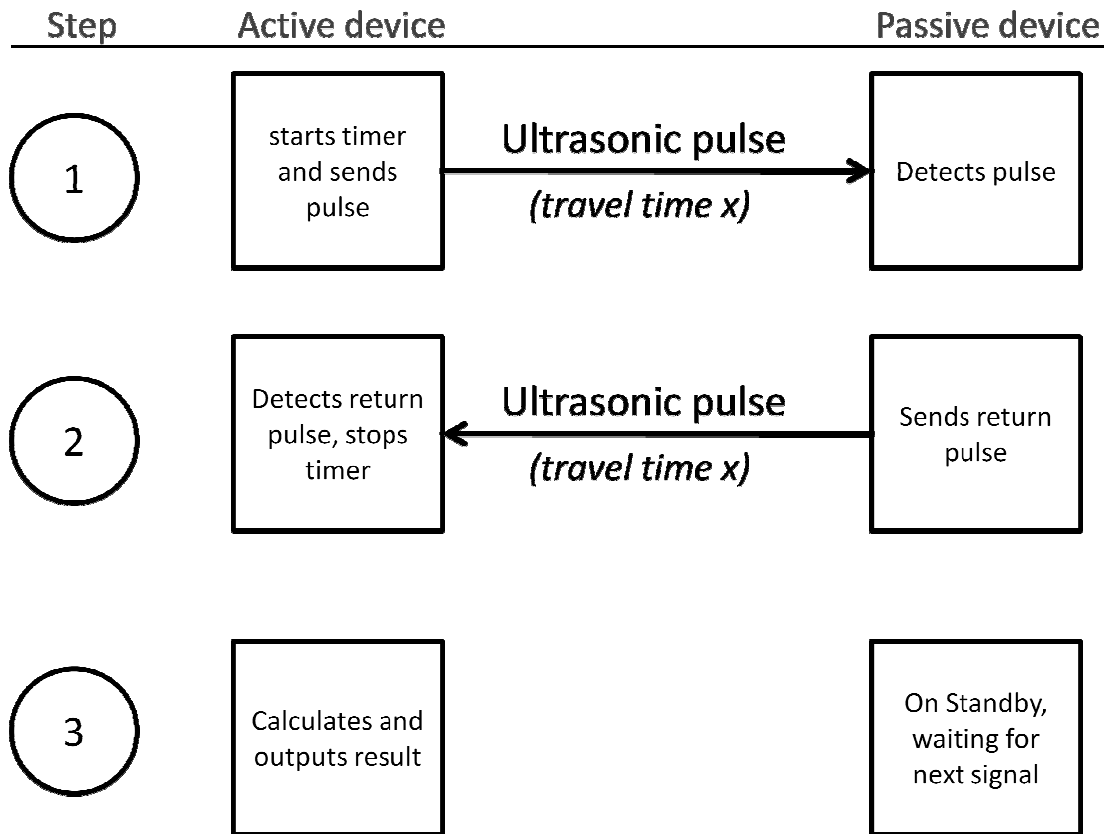


Figure 3.1 Illustration of overall ultrasound project concept

The generation of the signal, as well as all other necessary hardware controls will be realized by a microcontroller.

3.2 Concept of the electronic Circuit

Besides the main function to measure a distance, the circuit itself has generally three sub functions: transmitting, receiving and monitoring. Due to the project target costs, the fourth function, to measure the environment temperature, will be kept as an option.

The other functions can be explained like the following: The microcontroller generates the required 25 kHz signal via a Pulse Width Modulation (PWM) function. The output signal will be at one port only. Since the ultrasonic transmitter runs best with alternating voltage between both connections, the signal must be inverted and then sent in parallel to the second outlet of the transmitter. Secondly, these both signals must be amplified from 5V supply voltage from the microcontroller, to the desired operating voltage of the transmitter. Both functions can be realized by a "HEX inverting buffer". Since the inverting buffer needs a pre-amplified signal, in order to amplify from 5 V up to operating voltage, a transistor will be applied here.

The received ultrasonic pulse can possibly be partially very weak. For that reason, the pulse needs to be amplified in order to be detected by the microcontroller. This can be done by an

amplifier IC. In order to bring the amplified signal on a constant voltage level, it will be digitized afterwards. This function can, for example, be realized by a transistor or comparator. In order for the applicant to be able to see whether a signal reached the system, a yellow flash LED will be integrated behind the digitizer. This LED can be switched on and off by a jumper. This is to avoid voltage dropping once the system is running reliably.

The following scheme describes the concept of the electronic circuit for the active device:

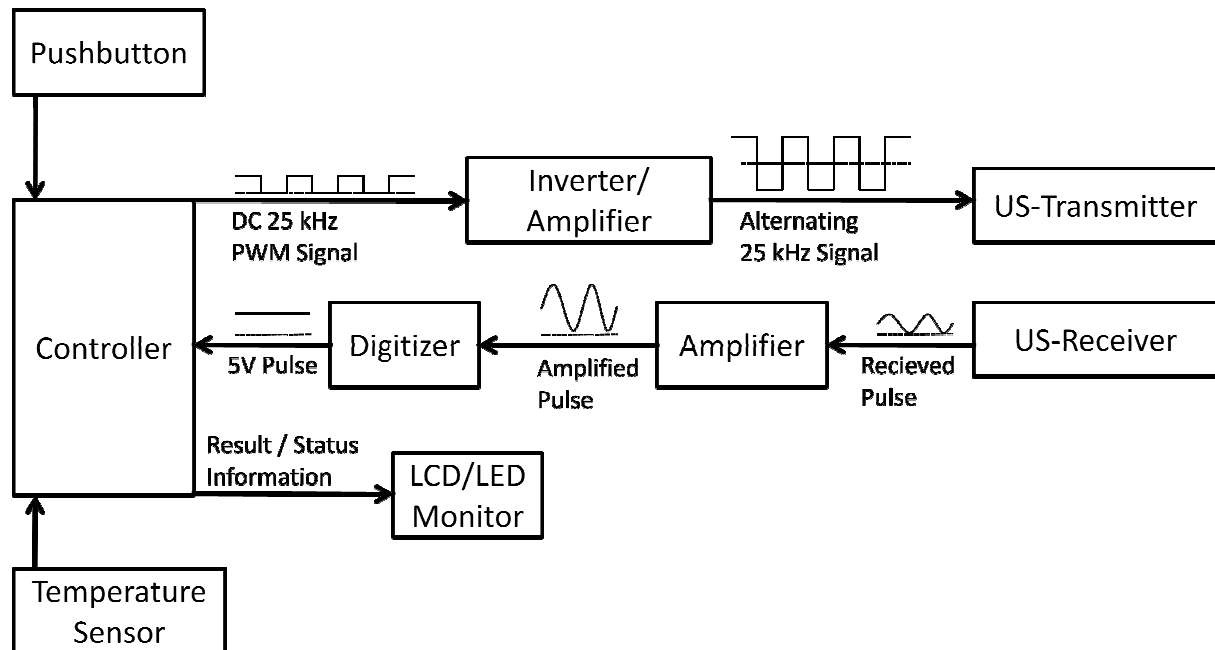


Figure 3.2 Electronic circuit concept - active device

Since the passive circuit board only needs to return the signal, it does not need a pushbutton to start the measurement or display information. A status LED will also be integrated, so the user can read out when the microcontroller has received a signal and returns it (although this process will not take more than approximately two seconds). The concept of the circuit would look like this (Figure 3.3):

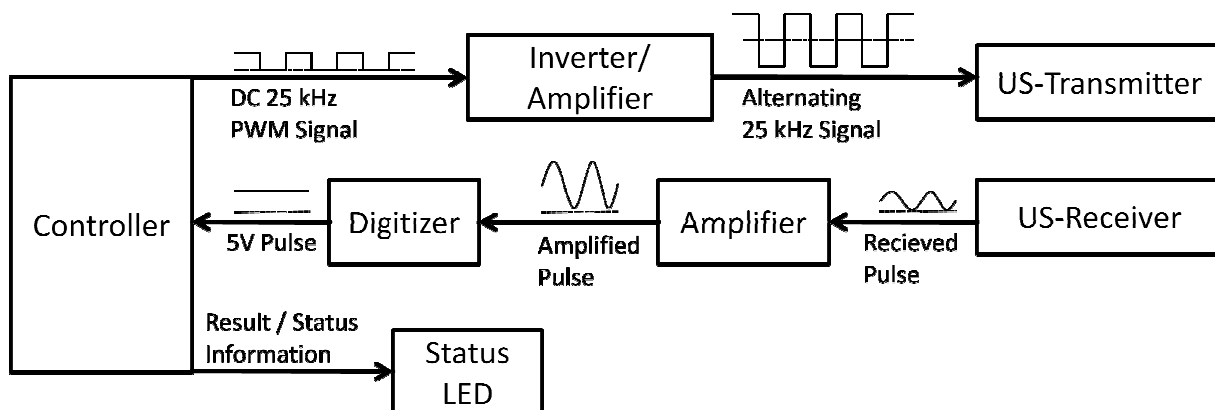


Figure 3.3 Electronic circuit concept - passive device

3.3 User Interfaces

It is necessary to be able to see the status of the system in order to let the applicant know what happens at the moment. Therefore, a two-color status-LED will be integrated. This LED can be used to show four statuses: off, red, green and orange². If this should be insufficient, the statuses can be extended by blinking colors. Furthermore, the user must be able to manually start a measurement. For this reason, a pushbutton will be added. Finally, the user must be able to read out the result of the measurement. The most common solution for this is to integrate an LCD Monitor.

3.4 Software Concept

The software code for the microcontroller, also called firmware, must handle all the procedures of sending, receiving, processing and displaying. Since the user can only press one button in order to continue, a user-diagram would be inappropriate here. Instead, a process diagram for the firmware is being drawn. This needs to be done for the active PCB, as well as for the passive PCB. The process of the active PCB is being illustrated in Figure 3.4. The diagram is separated by the columns: microcontroller process, status-LED, LCD Monitor, and user interface. The "interrupt flag", which is mentioned here, is a register on the microcontroller that indicates when an interrupt has taken place. In this case, the interrupt is the input signal reaching the microcontroller on the port pin where it is connected to the digitizer. The other interrupt is the overflow of the timer. The microcontroller needs to be told on which interrupts it shall react. This can be done by setting up the corresponding registers.

² Both, red and green active

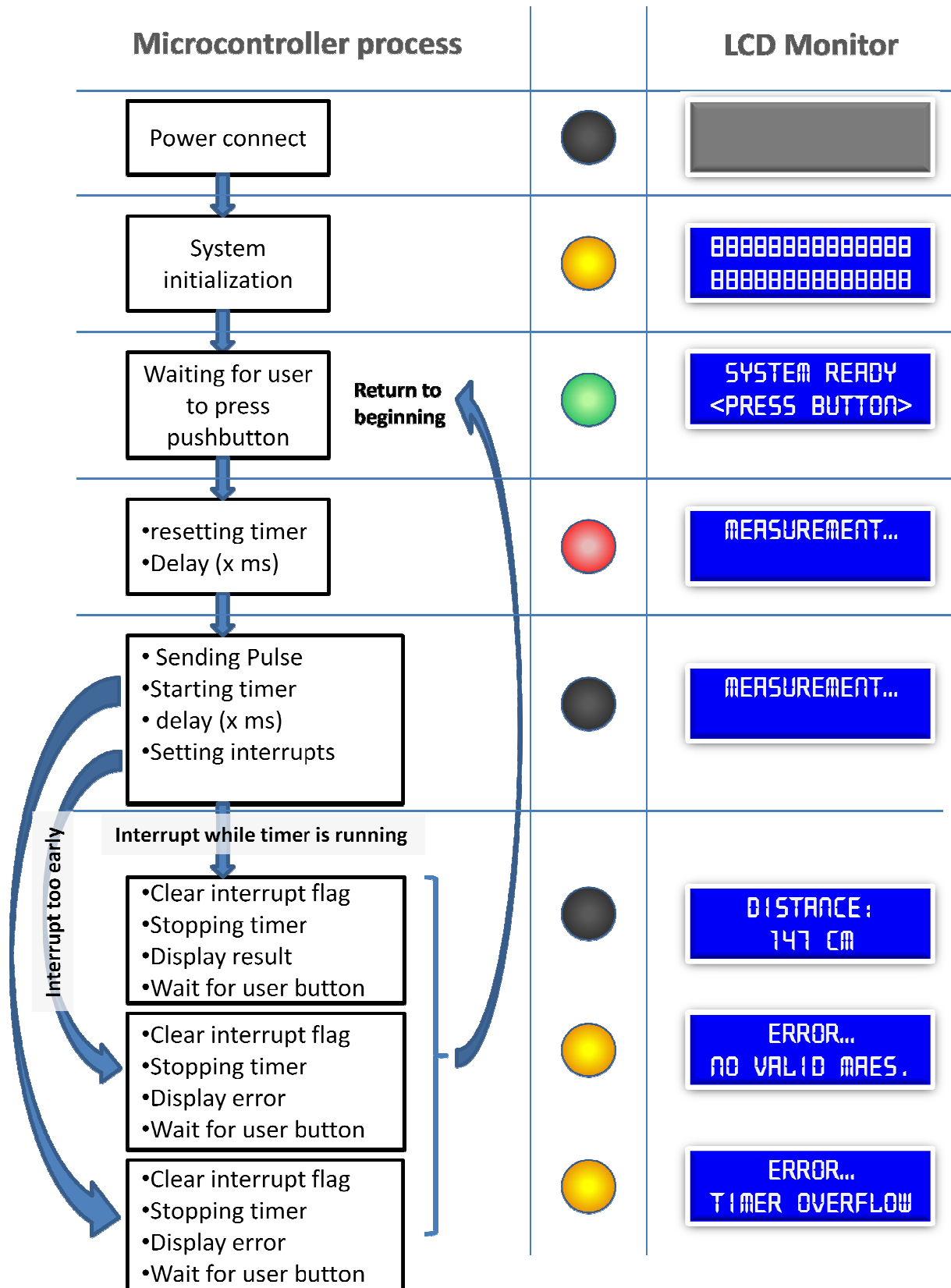


Figure 3.4 Software concept - active device

For the passive PCB, the flowchart is much simpler, since it does not have an LCD display or an active user input.

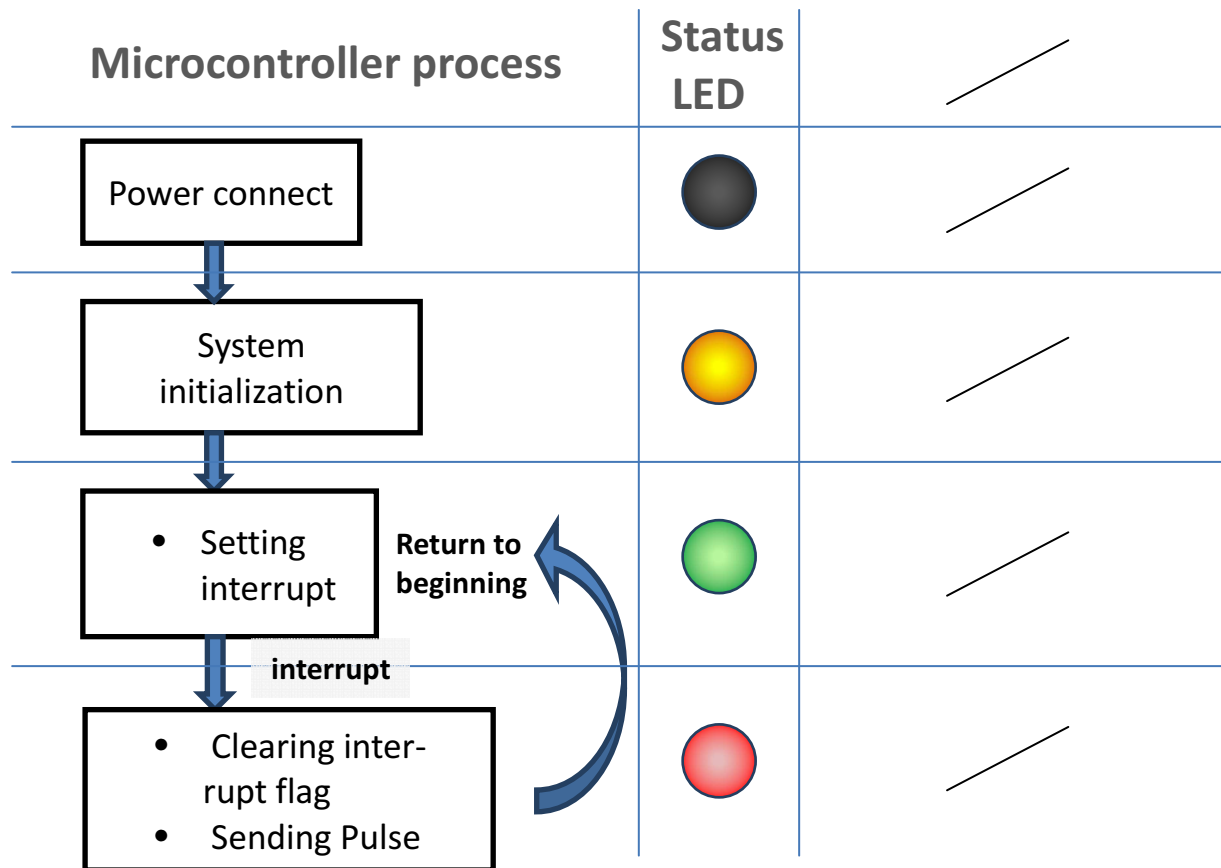


Figure 3.4 Software concept - passive device

4 Design

In this chapter, the design process is stated. It includes the dimensioning and choosing of the single parts. Based on this, the board circuit schematic and the board layout will be drawn.

4.1 Choice of essential Parts

For the most relevant parts, the microcontroller, transmitter, receiver, LCD monitor and the temperature sensor, there are several design rules for selecting the most appropriate one. These rules are listed in the following Table 4.1:

Table 4.1 Item Design Criteria

Item	Criterion
Microcontroller	<ul style="list-style-type: none"> - At least 8kb of program memory - ICSP - CCP Timer Module - One extra timer - At least 16 port pins to drive LED / LCD Transmitter etc. - Digital input port - DIP Package
Transmitter	<ul style="list-style-type: none"> - 25 kHz - 10V operating current
Receiver	<ul style="list-style-type: none"> - 25 kHz - 10V operating current
LCD Monitor	<ul style="list-style-type: none"> - 2 lines - 16 Characters per line - Ability of 4 bit driver mode
Temperature sensor	<ul style="list-style-type: none"> - Supporting -20 to 50 °C - 1-wire Digital technology

Following these rules, the parts displayed in Table 4.2 are chosen:

Table 4.2 Choice of essential parts

Item	Part
Microcontroller	Microchip PIC18F452 I/P
Transmitter	Prowave 250ST/160 (ceramic transducer)
Receiver	Prowave 250SR/160 (ceramic transducer)
LCD Monitor	Anag Vision AV1624
Temperature sensor	DS18S20 (1-Wire Digital Temp Sensor)

The parts mentioned in the table, are shown in the following figure. The temperature sensor has not been purchased.

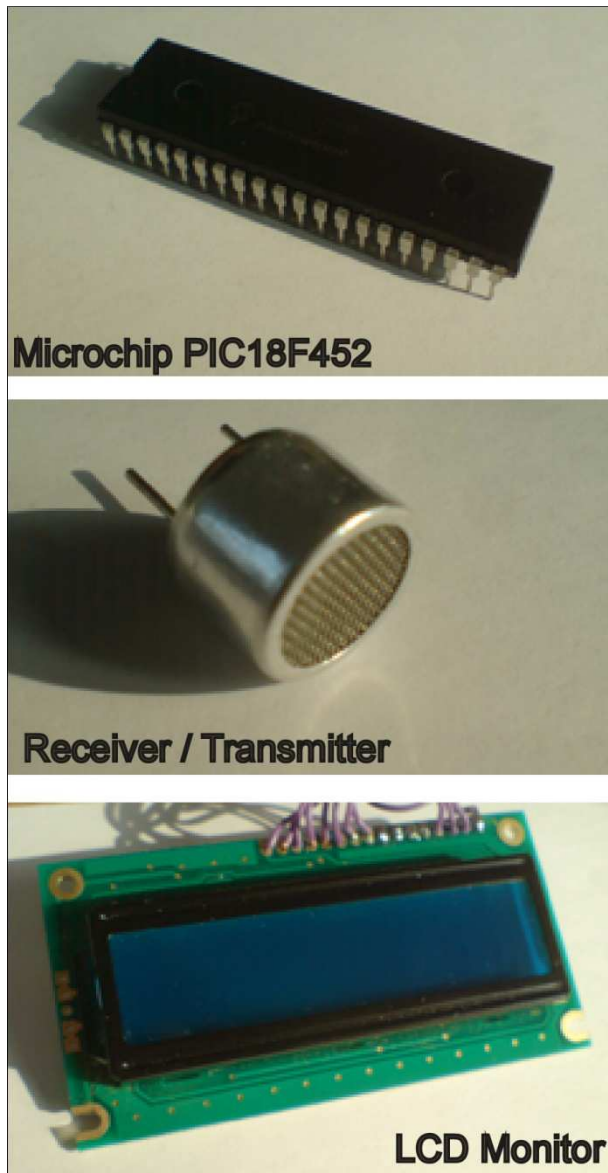


Figure 4.1 Essential Components

The rest of the relevant parts, as well as the designation in the board schematic are described in the following sub-chapters.

4.2 Voltage Supply

There are two voltage circuits needed: one low-voltage (5 V) and one higher voltage (10 V-12 V). The 5 V is needed for the microcontroller and to supply some of the ICs. The 10 V are needed to drive the receiver and the transmitter.

In order to realize the two voltages, a positive voltage regulator will be integrated. The specifications are the input voltage of 12 V, output voltage of 5 V, as well as a current of at least

500 mA. The current is only roughly estimated, but is, by experience, more than sufficient for this application. A 7805T positive voltage regulator, which satisfies these design rules, is chosen.

4.3 Oscillator

The PIC18F452 has no internal oscillator. Hence, an external crystal oscillator is needed. A standard device with 20.000 Hz (20 MHz) clock frequency is being chosen. This is the maximum operating clock speed for a PIC18F452 microcontroller. The controller needs 4 cycles for one instruction. This gives an instruction rate of 5 MHz.

4.4 Integrated Circuits (IC)

Although some of these parts were mentioned before, here is an overview (Table 4.3) of all Integrated Circuit parts of the board:

Table 4.3 Choice of Integrated Circuits

Function	Designation	Value
Microcontroller	IC1	Microchip PIC18F452 I/P
Hex inverting Buffer	IC2	4049N
Comparator	IC3	LM393N
Hex inverting Buffer	IC4	4049N
Operating amplifier	IC5	TL072P
Positive voltage regulator	IC6	7805T

4.5 Transistors

For both functions, the pre-amplifying for the inverting buffer as well as digitizing, a standard BC458 transistor can be applied.

4.6 Capacitors

The capacitors in the electrical circuit have two main functions: to debounce and to filter out unwanted frequencies. All values, that are necessary to debounce, can be found in the corres-

ponding datasheets. The values to filter out unwanted input frequencies were found in the electronic laboratory of the HAW. In order to find the right dimension for the capacitors, a switchable capacitor is taken to see the effect on the incoming signal. If the unwanted frequencies $f_{un_low} < 25 \text{ kHz} < f_{un_high}$ damps, but the target frequency of 25 kHz stays constant in its amplitude, the corresponding value can be overtaken for the final layout. Figure 4.1 and Figure 4.2 show the sending pulse as well as the same signal as input of the receiver both on the oscilloscope.

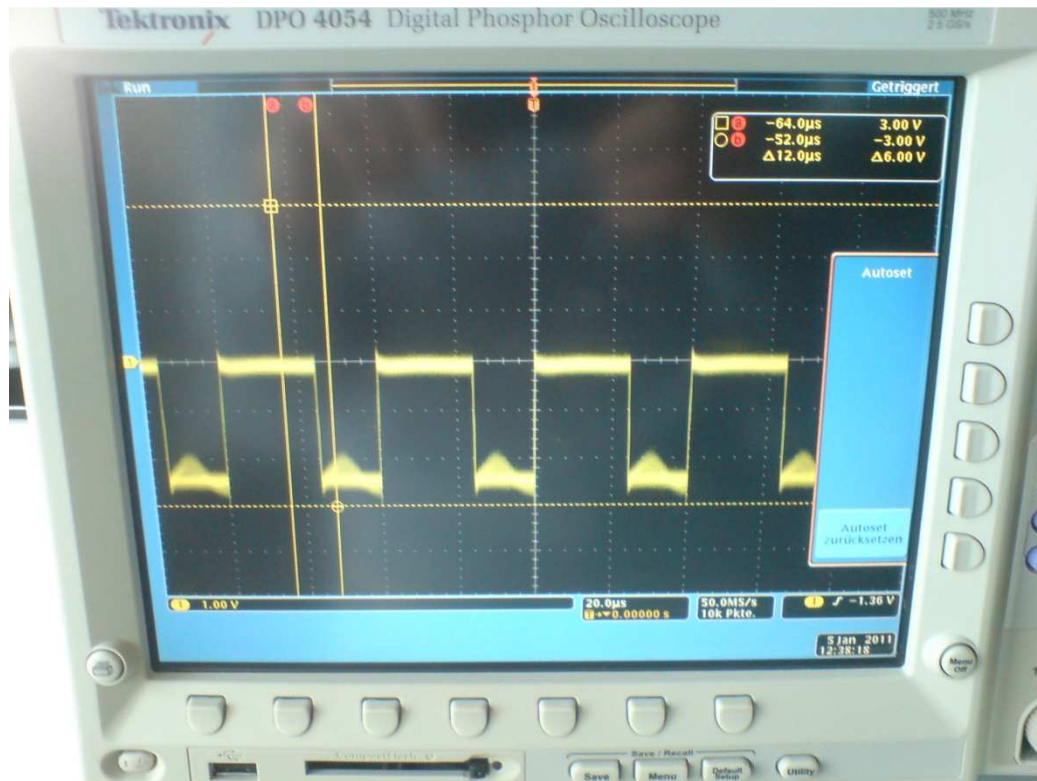


Figure 4.2 Oscillator image of ultrasonic pulse

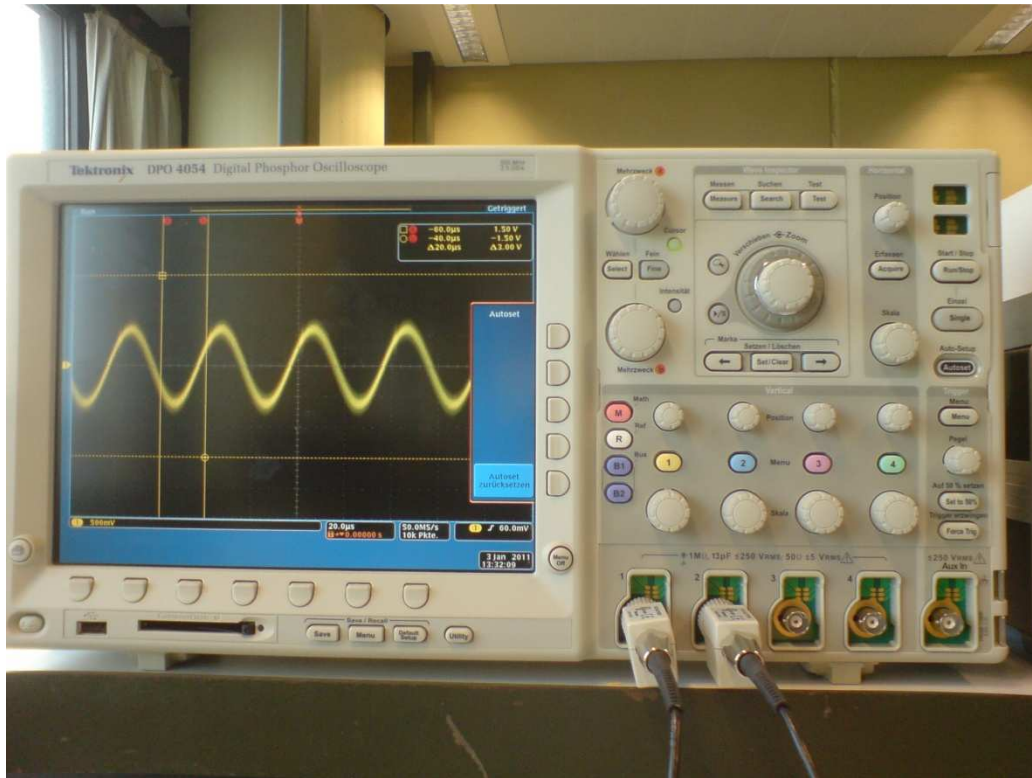


Figure 4.3 Oscillator image of ultrasonic pulse received by receiver

After figuring out the right values for the capacitors this way, the following values seen in Table 4.4 were chosen:

Table 4.4 Choice of capacitors

Function	Designation	Value
ICSP μ C protection	C1	0,1 μ F
Receiver frequency Filter 1	C2	1 μ F
Signal In debounce	C3	0,1 μ F
Pushbutton debounce	C4	0,1 μ F
Receiver frequency Filter 2	C5	470 pF
IC1 debounce	C6	4,7 nF
IC2 debounce	C7	4,7 nF
Receiver frequency Filter 3	C8	2,2 nF
μ C debounce	C9	10 μ F
Receiver frequency Filter 4	C10	0,1 μ F

4.7 Resistors

Most of the resistors in the board schematic are standard values. These can be found in the corresponding data-sheets. All other resistors concern the LED dropping resistor as well as the diode protection. These values are experience-based. Table 4.5 summarizes all resistor values.

Table 4.5 Choice of resistors

Funtion	Designation	Value
ICSP μ C protection	R1	10 k Ω
Amplifier standart R.	R2	100 k Ω
ICSP μ C protection	R3	470 Ω
Amplifier regulation	R4*	100 k Ω
Operating Point R. Q1	R5*	47 k Ω
Base R. Q2	R6	47 k Ω
Amplifier regulation	R7*	100 k Ω
Diode protection	R8	10 k Ω
Comparator standart R.	R9	10 k Ω
Base R. Q1	R10	5 k Ω
Collector R. Q2	R11	10 k Ω
Operating Point R. Comparator	R12*	100 k Ω
MCLR Button R.	R13	100 k Ω
LCD Contrast R.	R14*	10 k Ω
Status LED Dropping R.	R15	220 k Ω
Flash LED Dropping R.	R16	220 k Ω
Diode protection	R17	10 k Ω
Amplifier standart R.	R18	10 k Ω
LCD Backlight R.	R19	300 k Ω
Amplifier standart R.	R20	10 k Ω
Comparator standart R.	R21	4,7 k Ω
MCLR Pullup R.	R22	10 k Ω
Collector R. Q1	R23	4,7 k Ω
Base R. Q2	R24	100 k Ω

R. = Resistor

*Potentiometer

4.8 Diodes

There are two Light Emitting Diodes (LED) used for indicating and two normal diodes used for rectifying the input signal. The four diodes that are being used are listed in the following Table 4.6:

Table 4.6 Choice of diodes

Function	Designation	Value
Status LED	D1	R/G Duo LED
Signal rectifier	D2	1N4148
Signal rectifier	D3	1N4148
Flash LED	D4	LED yellow

4.9 Non-dialectical components

Non dialectical parts such as switches, bridges and pin headers are listed in Table 4.7:

Table 4.7 Choice of non-dialectical components

Function	Designation	Value
Bridge	J1	J15MM
USB for ICSP	J2	MTA06-100
Bridge	J3	J20MM
Bridge	J5	J7MM
Bridge	J6	J7MM
Bridge	J7	J10MM
Bridge	J8	J7MM
Bridge	J9	J7MM
Bridge	J10	J12MM
Bridge	J11	J5MM
Bridge	J12	J10MM
Bridge	J13	J5MM
Bridge	J14	J7MM
Bridge	J15	J5MM
Bridge	J16	J10MM
Bridge	J17	J7MM
Bridge	J18	J15MM
Bridge	J19	J15MM
Bridge	J20	J20MM
Flash LED Jumper	JP1	JP1E
Transmitter Pin Header	JP2	PINHD-1X2
Power Supply	JP3	PINHD-1X2
Receiver Pin Header	JP4	PINHD-1X2
Input Select Switch	S1	TL32PO
MCLR Switch	S2	RACON12
User-Switch	S5	RACON12
LCD Data Lines	SV2	MA08-1
LCD Power Supply	SV3	MA04-1

5 Circuit Layout

Based on the functionality of the two PCBs, a schematic was drawn with the help of a CAD software³ for electronic circuits and board layouts. With this software, a schematic can be drawn and valued. Out of this, the board layout is being created by placing the parts on the board and then routing their connections together. The schematic and the board layout stay the same for the passive PCB; simply the LCD will be left out when assembling. Thus, only one design that satisfies both boards was done.

5.1 Circuit Schematic Overview

The schematic provides the following sub functions:

- In Circuit Serial Programmable (ICSP) (red)
- Transmit ultra sound signal (yellow)
- Receive ultra sound signal (green)
- Digitize ultra sound signal (blue)
- User interfaces (purple)
- Voltage supply and regulation (orange)
- Controlling (left white)

The colors indicate the corresponding position on the schematic, shown in Figure 5. The functions are presented more detailed in Chapter 5.2 - Chapter 5.6.

³ EAGLE Cadsoft version 5.10.0

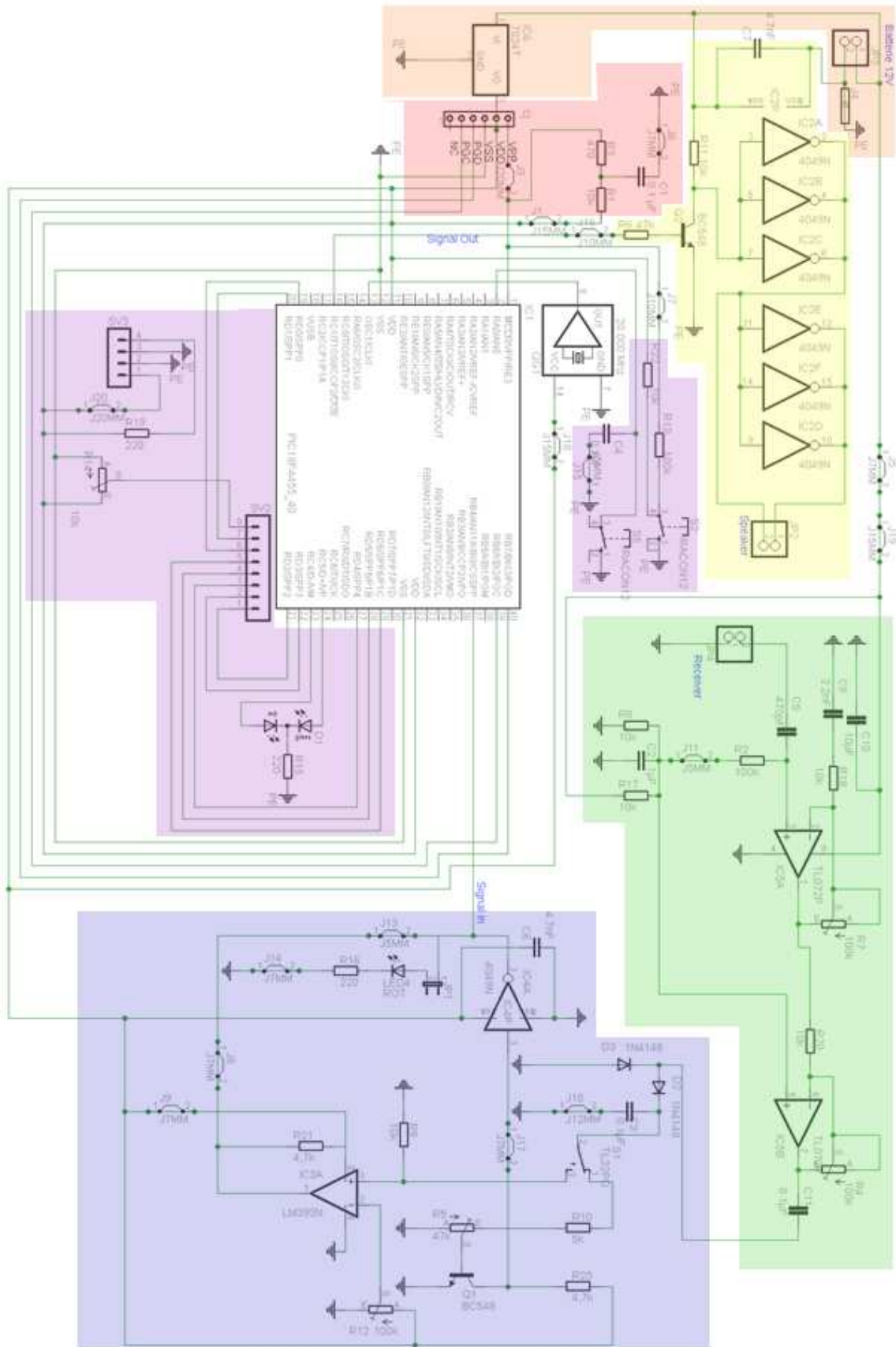
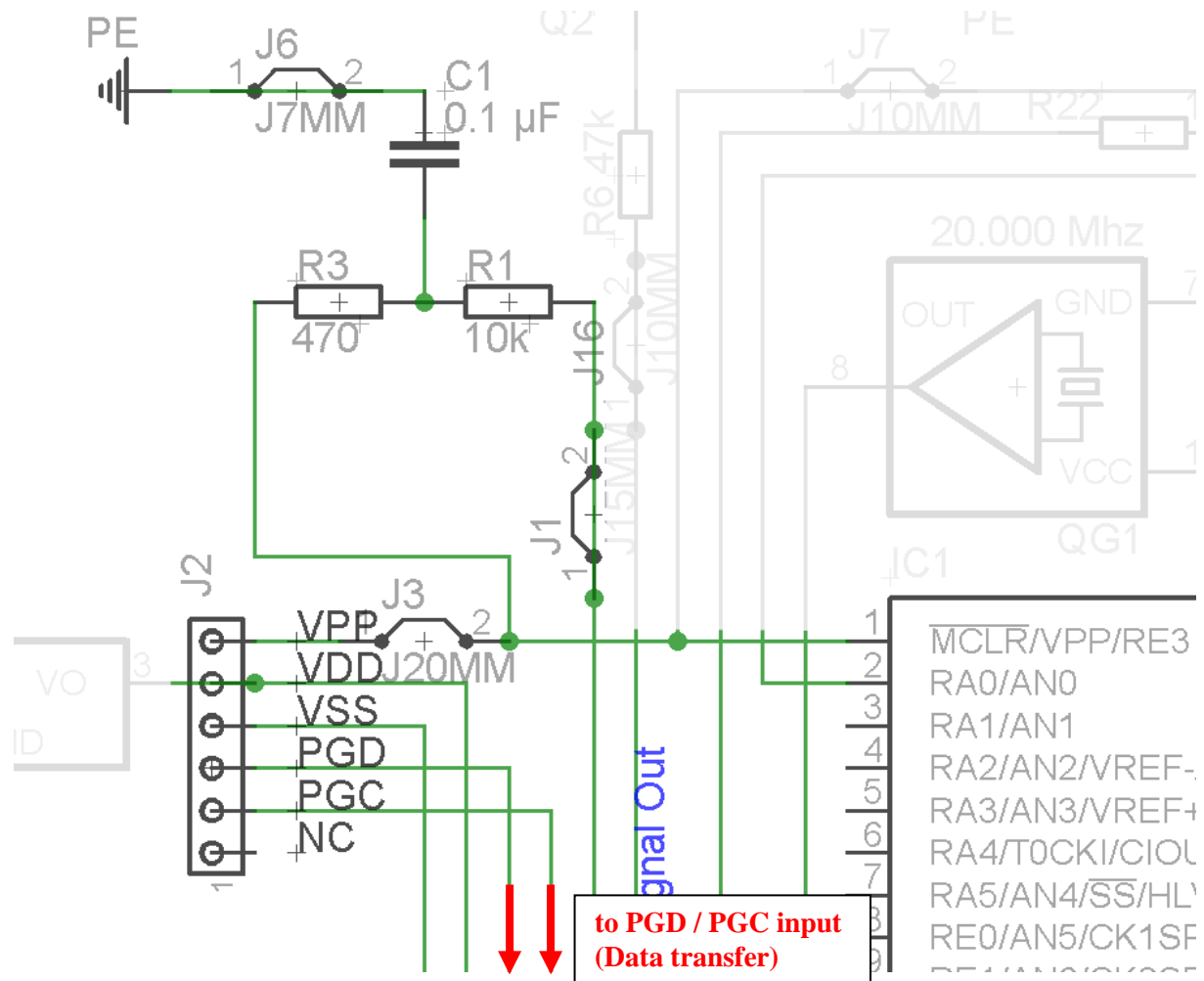


Figure 5.1 Circuit schematic overview

5.2 ICSP

This circuit is used for programming and reprogramming the microcontroller while it is built in the whole electronic circuit. Also, it can be used for debugging. The parts R1, R3 and C1 are a protective circuit for the microcontroller.

Figure 5.2 shows the “In Serial Programming Circuit” in detail:



5.3 Transmitting

In Figure 5.3, the transmitting circuit is shown. The signal pulse is being pre-amplified by the transistor Q2 and then amplified to 12 V battery power, as well as inverted by the IC2 (4049 Buffer). The transistor is necessary for the inverting buffer, because it cannot operate if the level between the operating voltage and the input voltage exceeds a specific value.

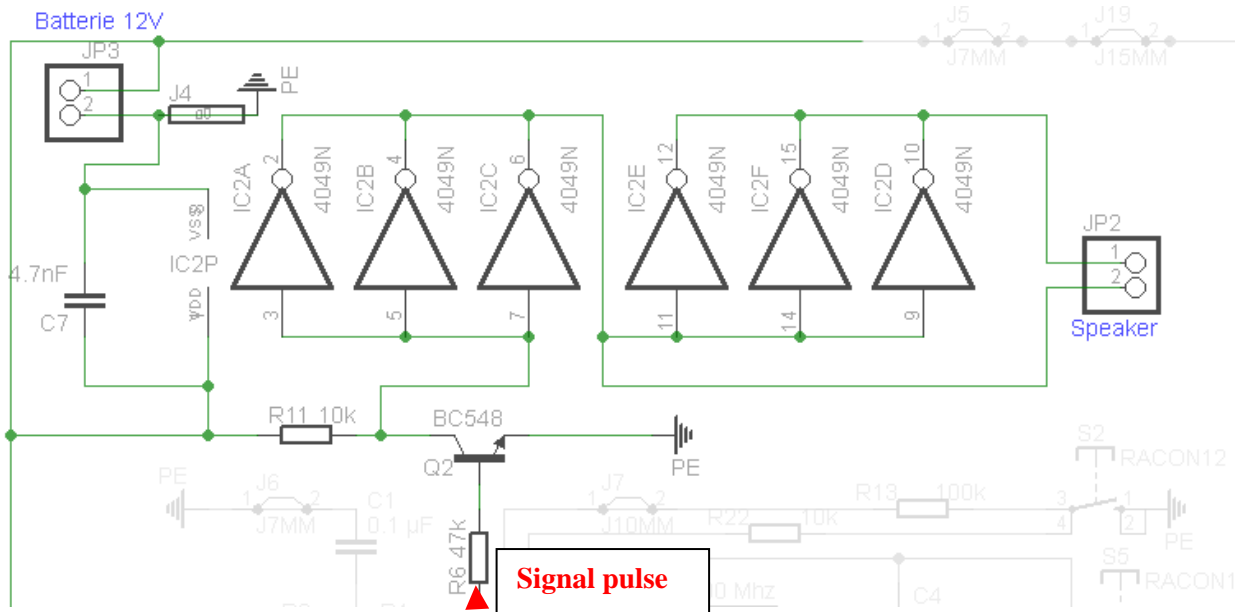


Figure 5.3 Circuit schematic detail: transmitting

5.4 Receiving

In the following Figure 5.4, the circuit of the receiver and amplifier is shown. The resistors R7 and R4 adjust the level of amplifying.

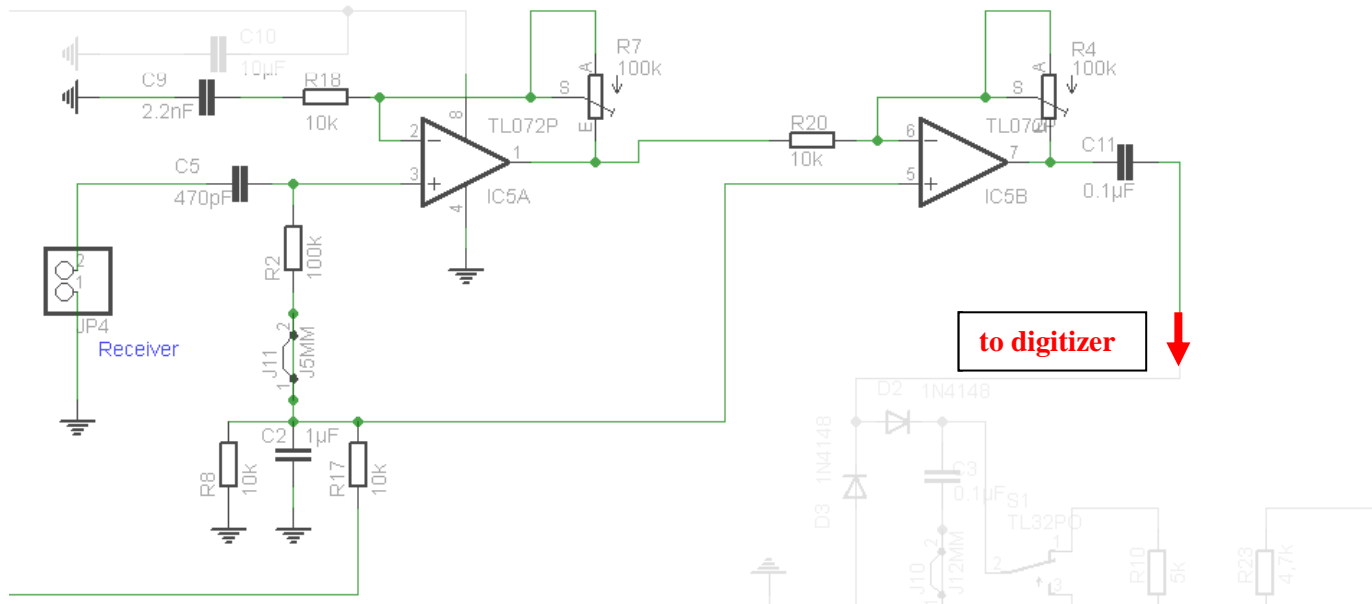


Figure 5.4 Circuit schematic detail: receiving

5.5 Digitizing

In Figure 5.5, the digitizer circuit is shown. A switch (S1) enables the applicant to choose between a comparator and a transistor to do this task. In order to make this possible, the corresponding IC, which is currently not being used, must be removed from its mounting. Furthermore, R21 is being built as a removable bridge and must only be inserted when the comparator is being used. The two diodes D2 and D3 rectify the signal. The resistor R12 regulates the operating point of the comparator, as R5 does this with the transistor Q1. The LED4 can be activated by jumper JP1 and flashes as soon as an amplified and digitized signal reaches the receiver.

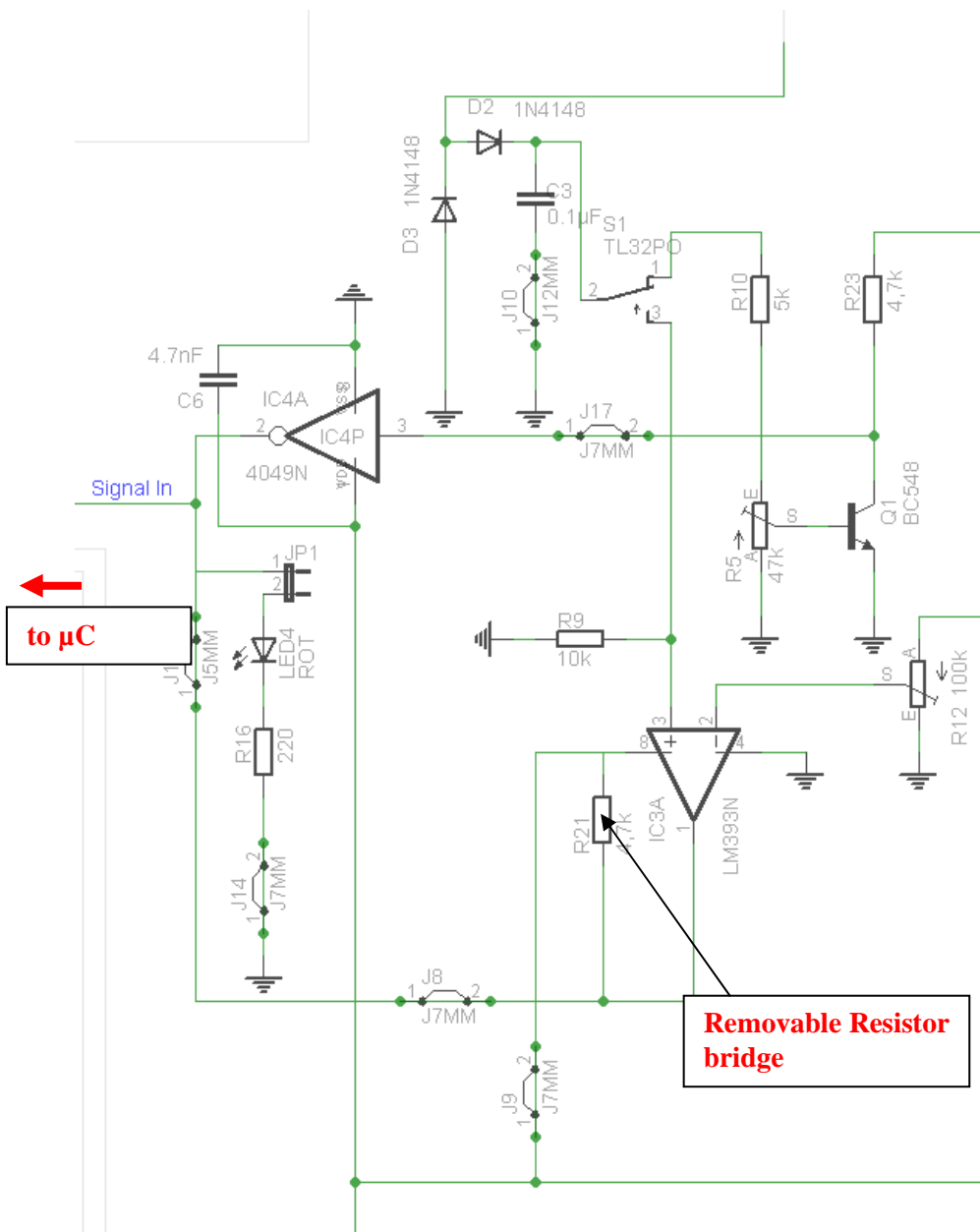


Figure 5.5 Circuit schematic detail: digitizing

5.6 Status LED / LCD

In the status display circuit (Figure 5.6), the R14 potentiometer is being used to regulate the contrast of the LCD display. R19 is a dropping resistor for the background LED for the LCD. Via the ports RC4 and RC5, the status LED (D1) can be controlled.

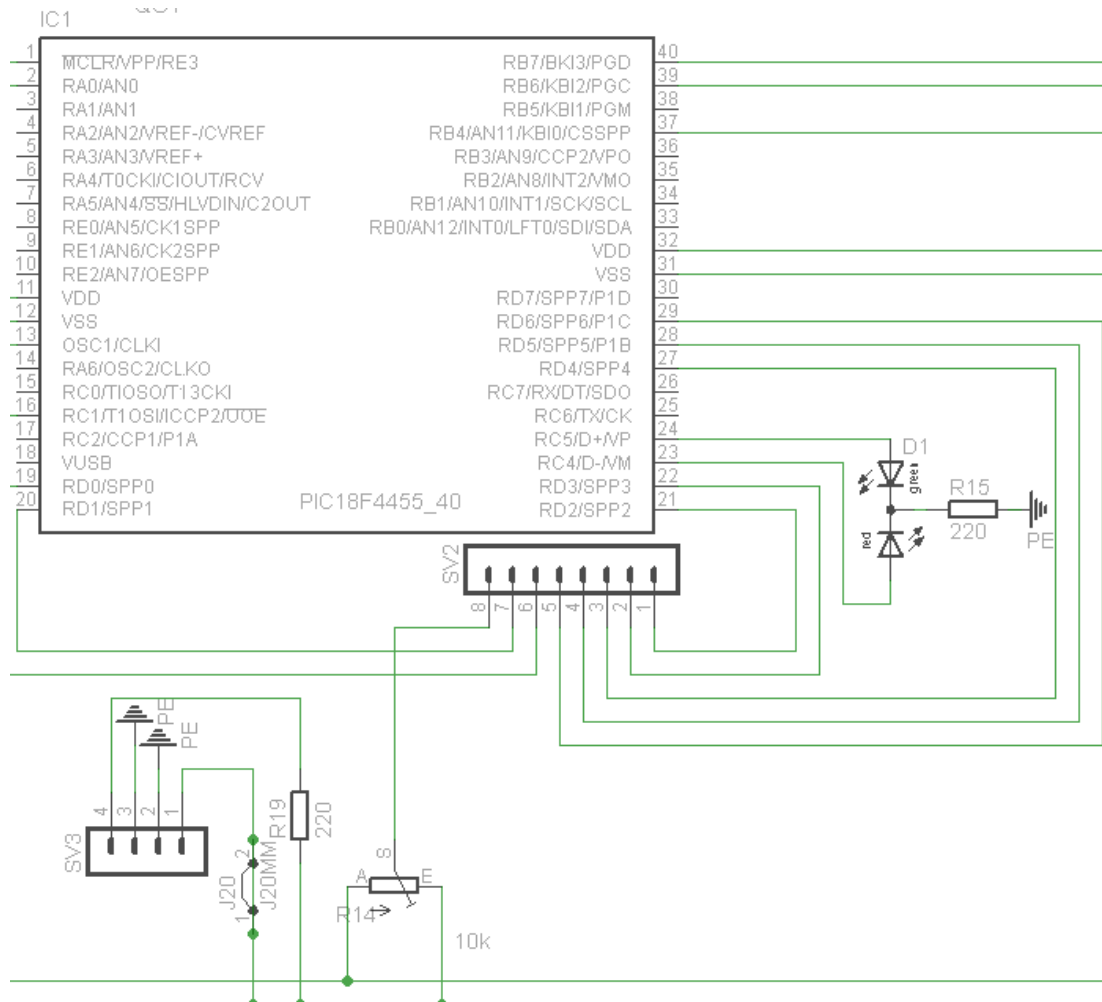


Figure 5.6 Circuit schematic detail: optics

5.7 Board Layout

Out of the schematic, the PCB design is being developed. The board may not be larger than 80mm x 100mm, since the license-free version of "Eagle Cadsoft" does not support larger dimensions. The final dimensions of the board are 75 mm x 100 mm. The following Figure 5.7 shows the ready developed board. All blue lines are strip conductors. The green parts are pads, where the parts are later being attached to by soldering. The white outlines adumbrate the hardware parts.

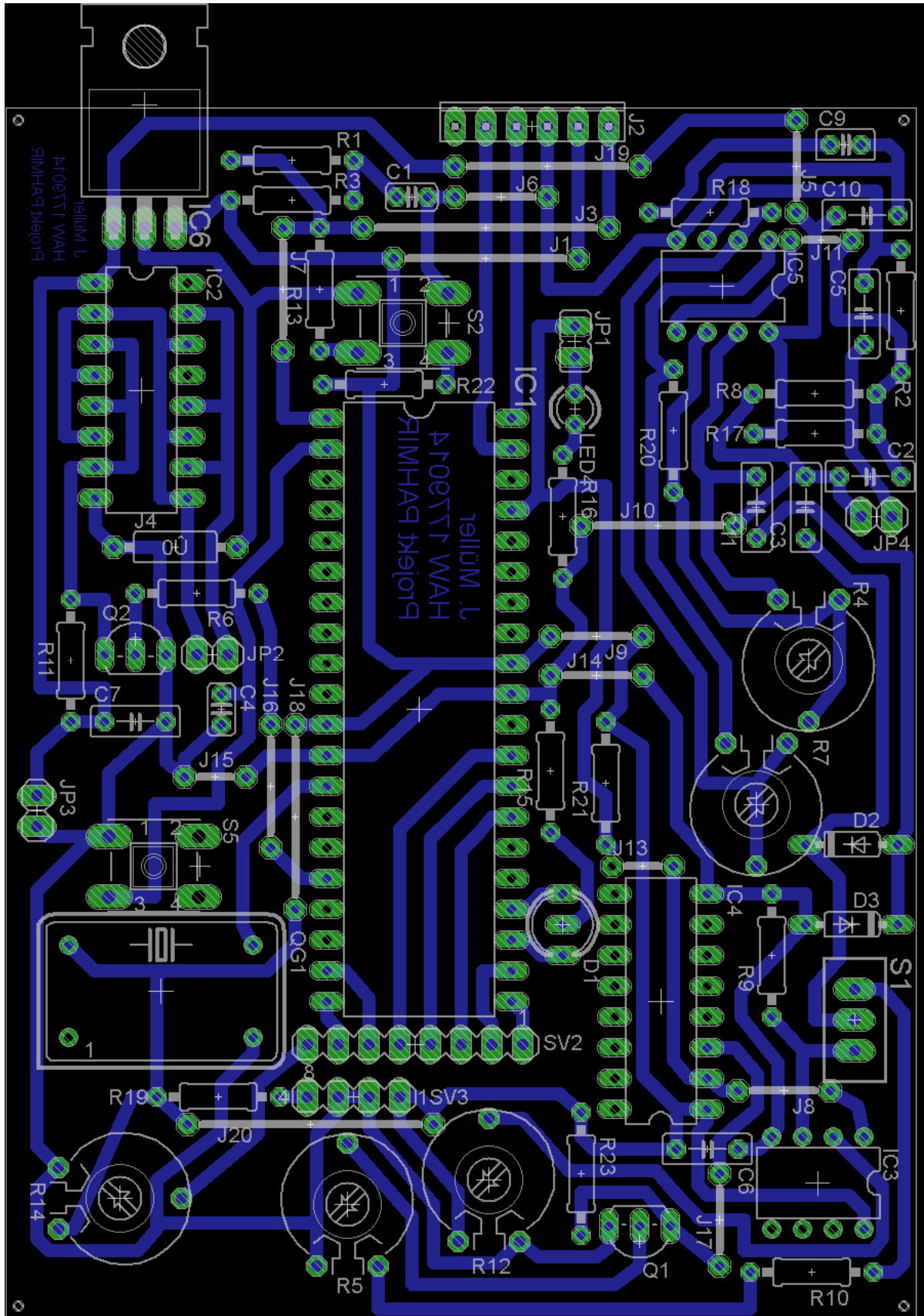


Figure 5.7 Board layout

6 Manufacturing

Out of the layout, two boards, the active and the passive one, are being manufactured. The single steps until the finished products are being described in the following two sub-chapters.

6.1 Etching

The manufacturing of the board requires the layout to be printed on transparent folia. Afterwards, a photosensitive blank circuit board is being exposed with an exposure machine (Figure 6.1). Therefore, the layout on the folia is being laid over the board and the machine power is switched on. This process takes about 3-5 minutes, depending on the age of the photosensitive material.



Figure 6.1 PCB exposure machine

When the layout is looming, the board is ready to be put into the developer liquid (Figure 6.2). This prepares the board for the essential etching process. When this is finished, the board looks as in Figure 6.3.



Figure 6.2 PCB in photo developer liquid

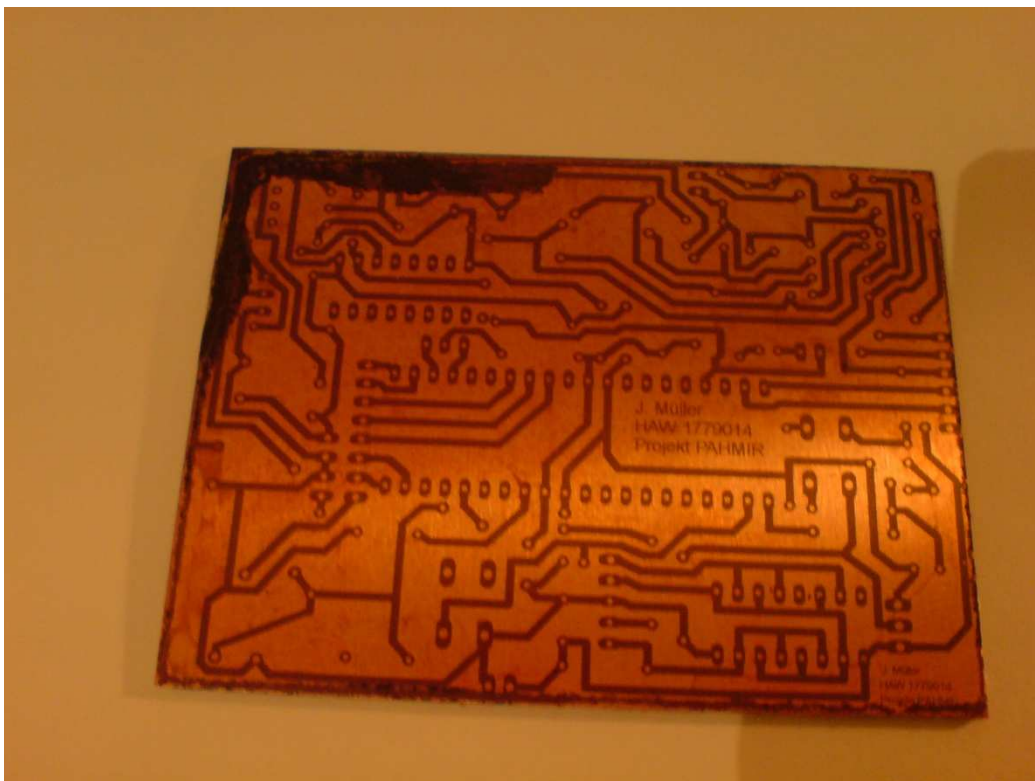


Figure 6.3 Test exposure (the protect folia failed to release from the board here)

The unexposed parts are being etched off by sodium sulfate with the help of an etching machine (Figure 6.4). The machine ensures a constant temperature of the liquid at about 50°C. The higher the temperature, the faster the process takes place.

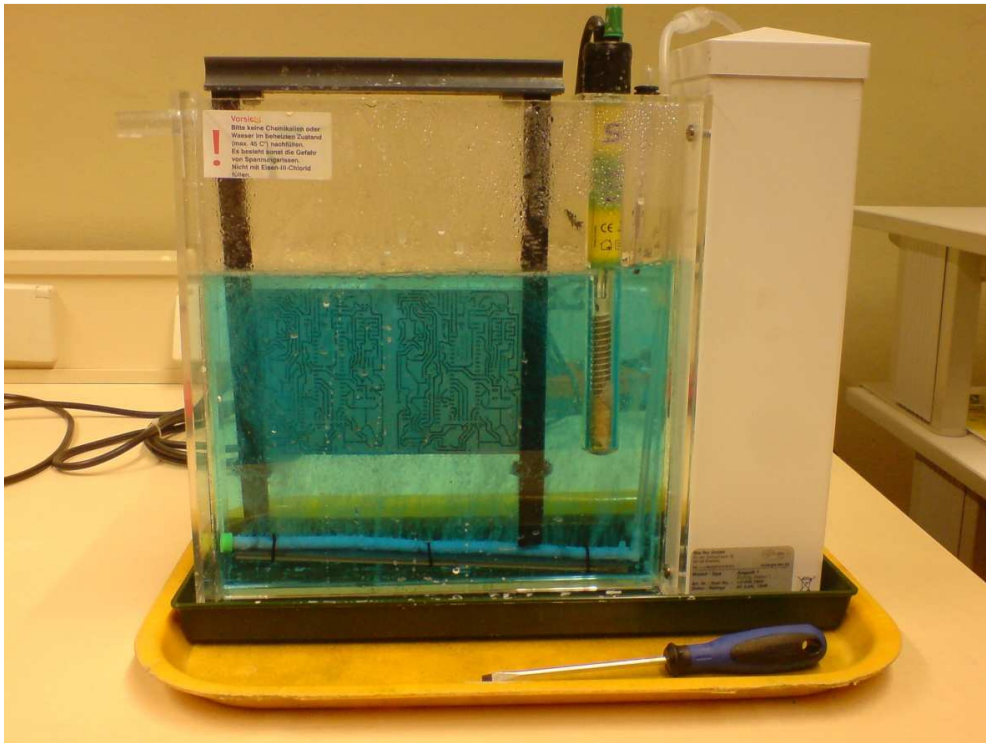


Figure 6.4 PCB etching machine

When this is finished, the board looks like in Figure 6.5

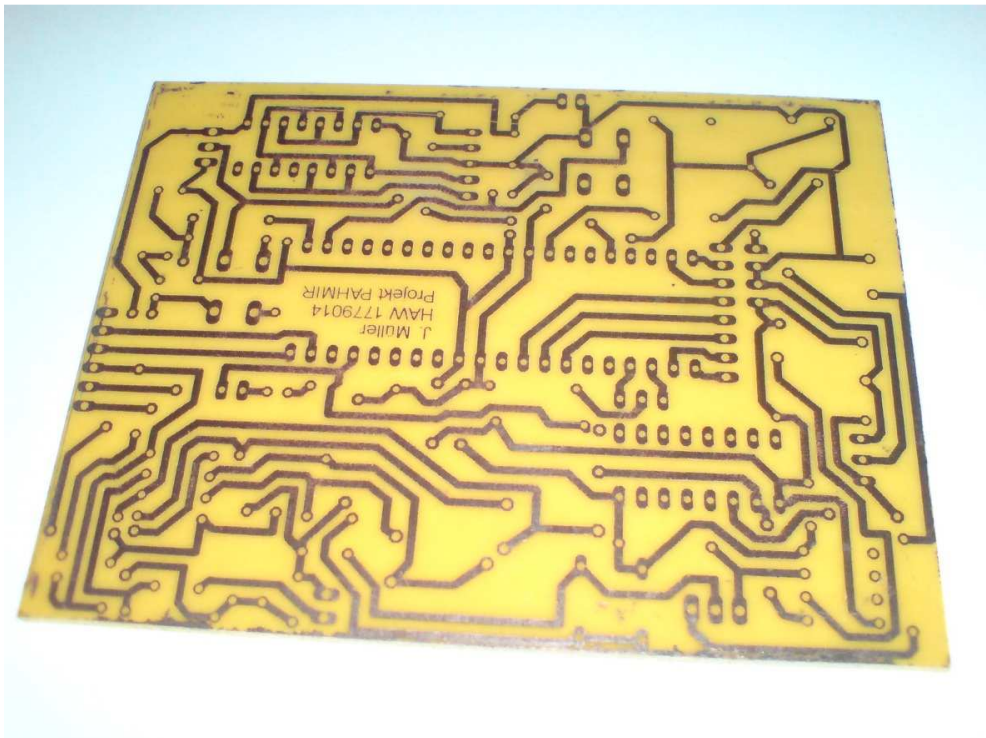


Figure 6.5 Ready etched circuit board

6.2 Soldering

After all necessary holes for the hardware components have been drilled they can be soldered onto the board. The soldering temperature must be about 350°C for leaded solder tin. On the active board, the user pushbutton as well as the LCD monitor and the status LED are being connected by wires to the board due to easier handling. Also, one wire for the temperature probe is being installed. Figure 6.6 shows the finished active device from the top, Figure 6.7 from the bottom.

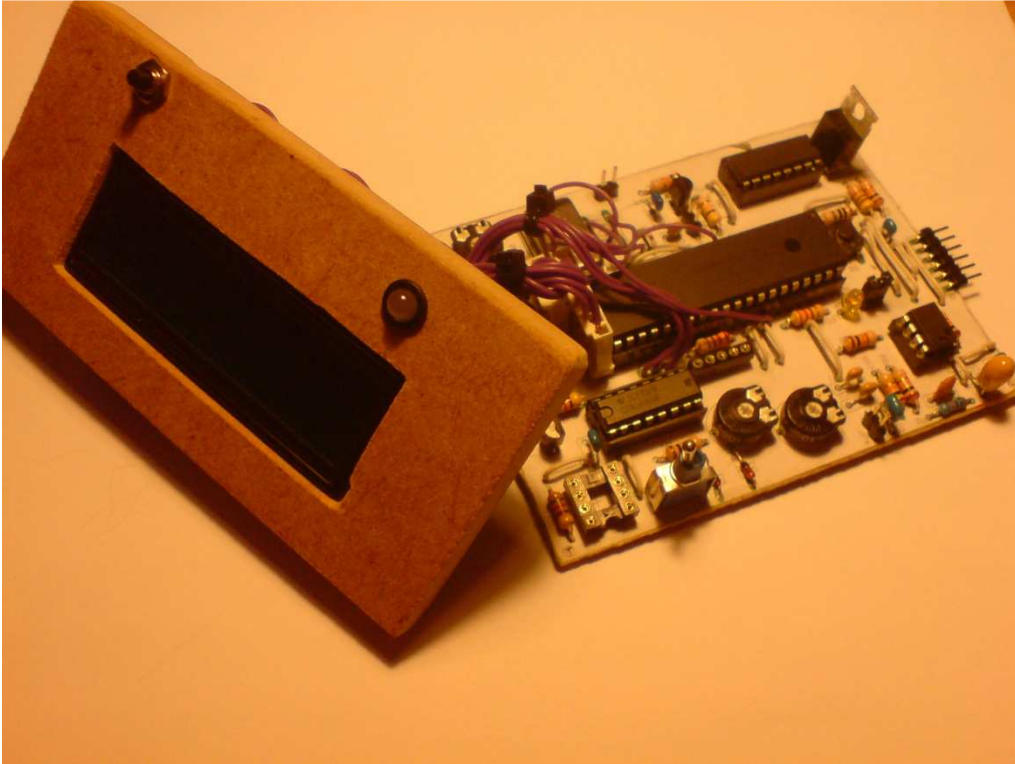


Figure 6.6 Finished active device - top

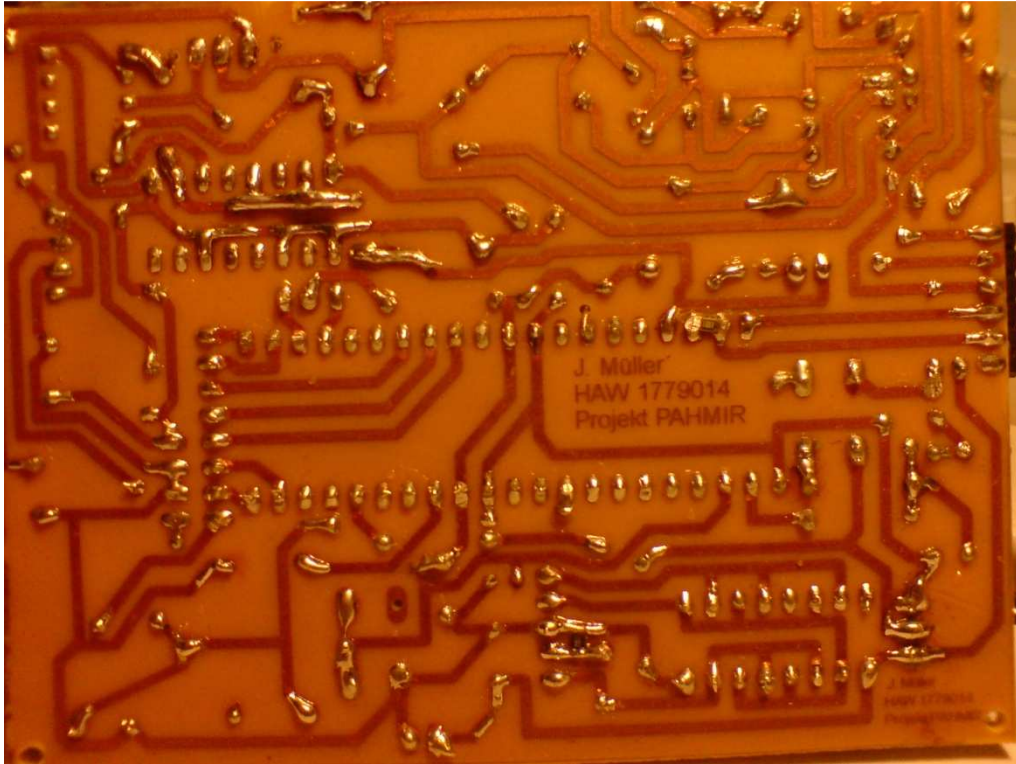


Figure 6.7 Finished active device - bottom

Figure 6.8 shows the finished passive device:

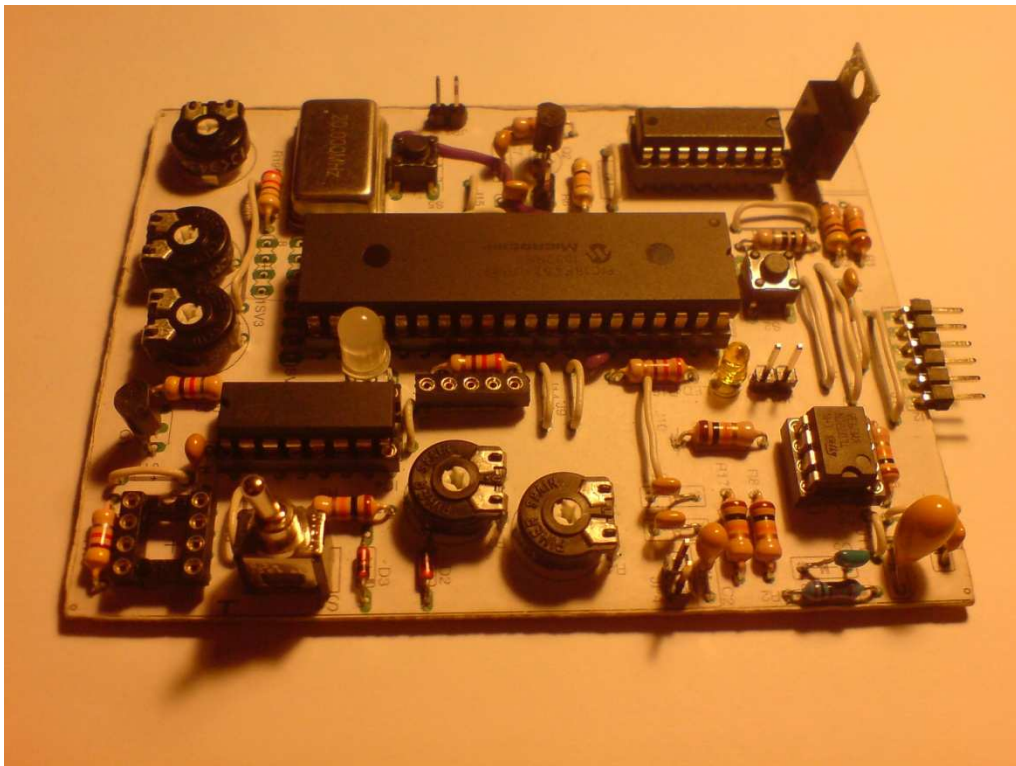


Figure 6.8 Finished passive device - top

The transmitter and the receiver are glued into a wooden frame and connected to wires and a plug. The plug enables the transducers to be removed from the board later. The transmitter

wires are marked by a red wire, wrapped around them. One of the two transmitter/receiver combination is shown in Figure 6.9.

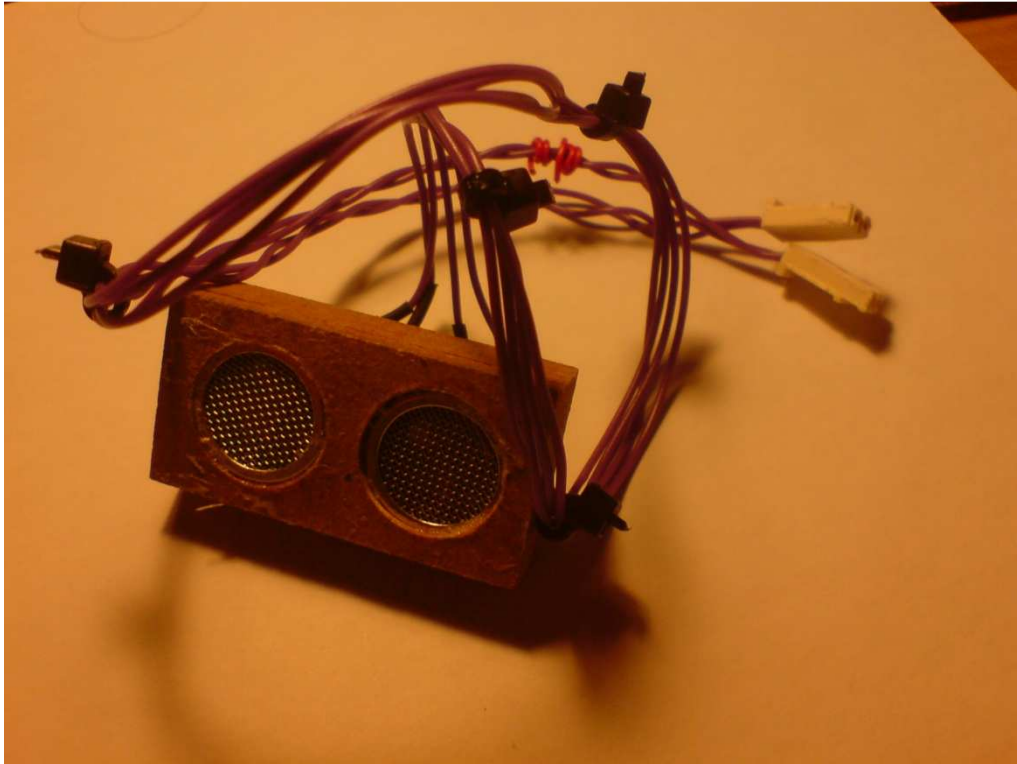


Figure 6.9 Receiver / transmitter combination in frame

6.3 Programming

The programming of the firmware to the microcontroller is being done with the help of the "Microchip PICkit2" programming device. This device provides serial programming and can be connected to the USB port of a PC. As the development environment, the software "MPLAB" from Microchip is being used, in which either the programming language "C" or "Assembler Code" can be used. The firmware for this project is being written in "C". As a compiler, which compiles the C-written code to a readable HEX (-adecimal) file for the microcontroller, the integrated Microchip C18 Compiler for the PIC18-series is being used.

The full source code can be found in Appendix A and Appendix B.

6.4 Generating the Signal

The Microchip PIC18F452 I/P has three timers: Timer0, Timer1 and Timer2. Timer2 is used for the generating of the PWM signal. The timer0 will be used for measuring the time between the signal release and the input of the return signal. Timer2 is being configured by the T2CON, PR2 and the CCPR2L registers according to the data sheet. Timer1 will stay unused for this application.

7 Testing

From the first power-up of the system, some more adjustments to the firmware are needed to be done. This mostly concerns the calibration of the measurement procedure to have exact results. Also, the minimum delay times are needed to be found, in order to not have unnecessary long delays but also without interferences of signal inputs, which come from the pulse of the same board. These other important processes of calibrating the timer and hardware delays are described in the following chapters.

7.1 Configuration of Timer0

"Timer0" of the PIC18F452 controller is selectable as an 8- or 16-bit counter. In 8-bit mode, the timer counts until it reached $8^2 = 64$ bits and then resets to zero. Since the timer is running at 1 MHz, it would overflow too fast; before the return signal reaches the receiver:

$$\frac{64}{1 \cdot 10^6 \text{ s}^{-1}} = 0,000256 \text{ s} = 0,256 \text{ ms}$$

Thus, the 16-bit selection with $16^2 = 65538$ bit values is chosen.

But still, the timer will overflow too fast: for 65583 bit values, only

$$\frac{65583}{1 \cdot 10^6 \text{ s}^{-1}} = 0,0655 \text{ s} \approx 65,5 \text{ ms}$$

will pass until the timer overflows. The solution is the prescaler function, which lets the timer run either with 1:2nd, 1:4th, 1:8th, 1:16th, 1:32nd, 1:64th, 1:128th, or 256th of the time per instruction. However, this function needs to be set carefully, since the timer must still run fast

enough to be able to detect the time difference while the sound signal travels 1 cm. The speed of sound is around:

$$v_s \approx 330 \frac{m}{s} \approx 3300 \frac{cm}{s}$$

Thus, the time, that sounds needs to travel 1 cm, is:

$$v = \frac{d}{t} \tag{7.1}$$

$$t = \frac{d}{v} = \frac{1 \text{ cm}}{3300 \text{ cm s}^{-1}} = 0,000303...s \approx 0,3 \text{ ms}$$

For a precise measuring, 1:64 will be sufficient. At this prescaler setting, the timer runs one instruction per

$$\frac{64}{1 \cdot 10^6 \text{ s}^{-1}} = 0,00006...s \approx 0,0064 \text{ ms}$$

This is finer than the required 0.3 ms. At this prescaler setting, the timer will overflow after

$$65,5...ms \cdot 64 \approx 4 \text{ s}$$

7.2 Finding of Constants

There are several delays in the software and also in the hardware that influence the measure result. In the software, for example, a clearing time is required to avoid the signal pulse to go directly into the receiver of the board, where it was sent from. The hardware components have individual delays, which are mainly unknown. When summing up all the occurring delays, one constant delay can be found. This constant can be found by doing measures from various distances. The delta values between the results correspond to the delta time that the ultrasonic signal needs to go travel. When creating a linear formula out of this, the constant value can be found when setting the distance to zero. In reality, a zero distance measure is being avoided because of unknown direct interferences between the two transmitter/receiver modules. The general linear equation is:

$$y = a \cdot x + b \tag{7.2}$$

Where the gradient a in this case is:

$$a = \frac{\Delta t}{\Delta d_{meas}} \quad (7.3)$$

7.3 Test Arrangement

In order to find a valid a , a calibration starting from 50 mm – 410 mm is being accomplished in 10 mm steps (see Figure 7.1). In order to have as little interference as possible as a reference, the test was conducted in the sound proof room of the HAW.

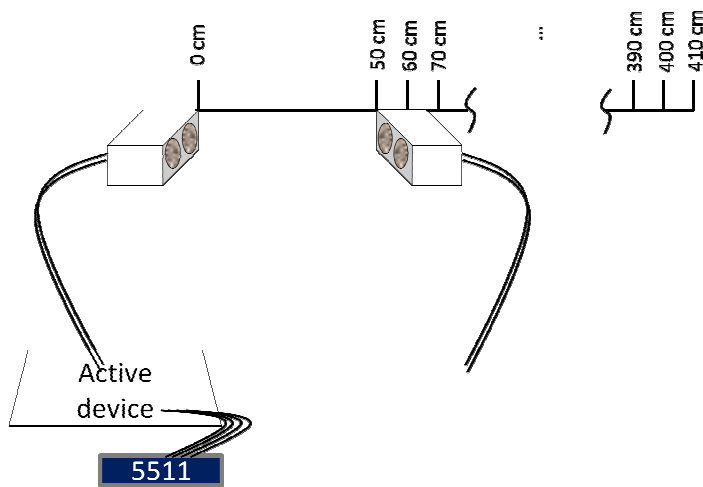


Figure 7.1 Test arrangement

For each distance, the timer value of timer0 is being noticed.

The test results are written down in the following Table 7.1:

Table 7.1 Result of test distance measurements

Timer1 Value @ 1:128 [timer counts]	Actual distance [mm]
5511	50 mm
5514	60 mm
5516	70 mm
5520	80 mm
5523	90 mm
5526	100 mm
5528	110 mm
5532	120 mm
5534	130 mm
5538	140 mm
5541	150 mm
5543	160 mm
5546	170 mm
5549	180 mm
5550	190 mm
5553	200 mm
5555	210 mm
5556	220 mm
5560	230 mm
5564	240 mm
5566	250 mm
5567	260 mm
5568	270 mm
5571	280 mm
5575	290 mm
5579	300 mm
5581	310 mm
5584	320 mm
5586	330 mm
5590	340 mm
5592	350 mm
5595	360 mm
5598	370 mm
5601	380 mm
5605	390 mm
5608	400 mm
5612	410 mm

When assuming that the coherence between the distance and the stopped timer value is linear, one can simply take the first and the last value to determine a . One way to find out if linearity is given is to do a regression analysis and then check for the standard variance. The simpler, but also more subjective, way to do this is to plot a diagram (Figure 7.2) of the test values. For orientation, a red line from the first to the last value is being drawn.

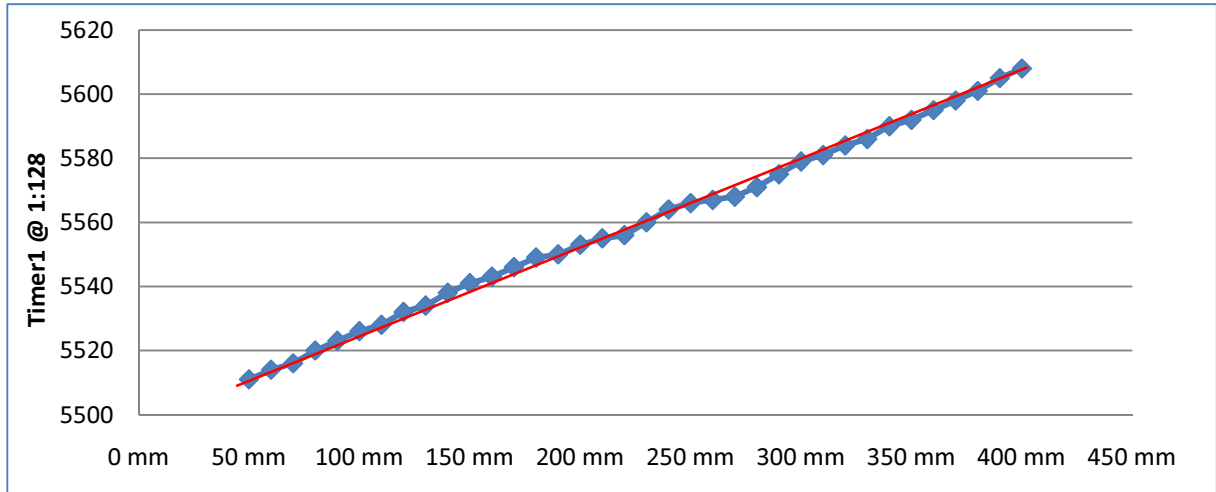


Figure 7.2 Result of test distance measurements shown in a diagram

Despite from little ups and downs, the overall connectedness can be treated as linear. The exact reason for this effect of inconsistency is basically unknown, but will most probably be originated by interferences of the aircraft cabin interior, where the measuring was done. With Equation (7.3), a can now be calculated by plugging the deltas of time and distance:

$$a = \frac{\Delta t}{\Delta d_{meas}} = \frac{(t_{37} - t_1)}{(d_{37} - d_1)} = \frac{(5612 \text{ tcnts} - 5511 \text{ tcnts})}{(410 \text{ mm} - 50 \text{ mm})} = 0,2805 \dots \frac{\text{tcnts}}{\text{mm}}$$

Now, the missing b of Equation 7.2 can be found by plugging a random value:

$$b = y - a \cdot x$$

$$b = 5511 \text{ tcnts} - 0,2805 \dots \frac{\text{tcnts}}{\text{mm}} \cdot 50 \text{ mm}$$

$$b \approx 5496,972 \text{ timercounts}$$

The two constants a and b can now be defined in the firmware to determine the measured distance by the timer value.

After this, the system maximum distance was evaluated in the MTTC at Airbus Hamburg. The furthest possible distance, restricted only by the arrangement of the MTTC cabin, is around 6.5 meters. The devices were still able to receive signals by each other at this distance.

Certainly, the results had ups and downs of up to 5 cm. In the following Figure 7.3, the test build up for a low distance evaluation can be seen.

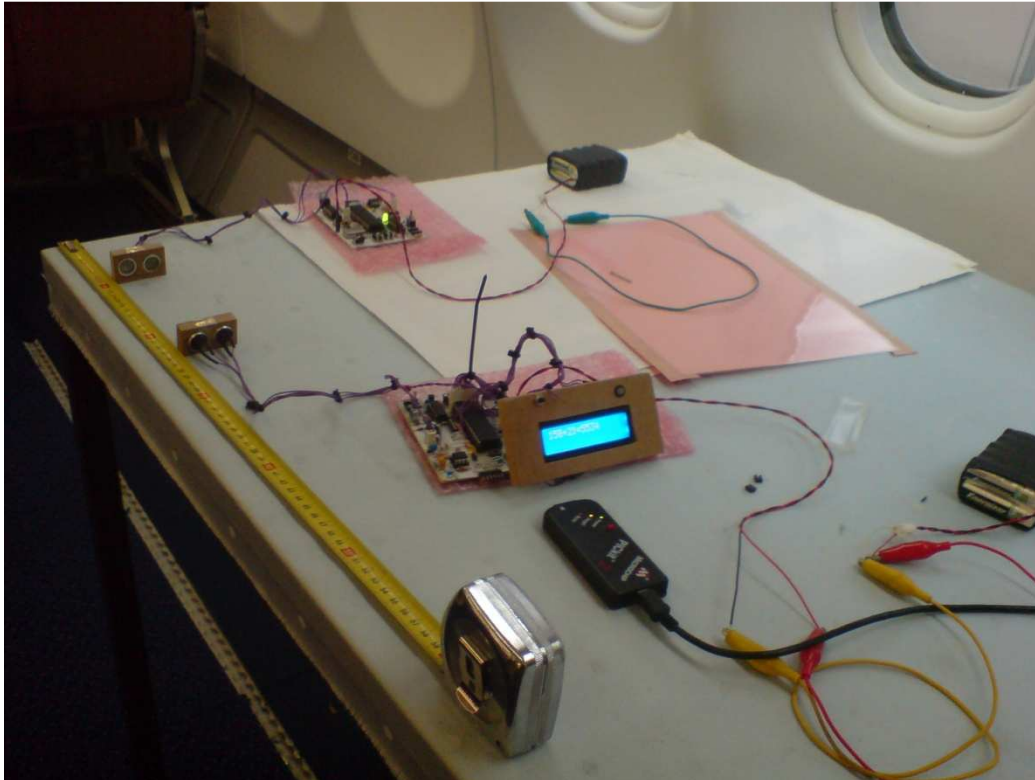


Figure 7.3 Evaluating of the system in the MTTC

Conclusion

In order to have a simple and reliable system, that detects locations in an aircraft cabin, the use of ultrasound can generally be rated as inappropriate.

Apart from the fact that this prototype can be improved in many aspects, there will always be insecurities in a cabin that could disturb the measures. The prototype could significantly be reduced in its size within a serial production. This can be achieved by the use of SMD parts, as well as multi layered PCBs. However, there is one significant problem that will occur, as soon as there is not only one device, whose distance need to be checked, but at least two more. For this purpose, either transducers that can transmit and receive ultra sound in/from any direction, or a rotary system needs to be evolved. In both cases, the accuracy will probably be even worse. The prototype showed measure results varying up to 5 cm without distance change. This at the highest possible range which could be arranged in the MTTC (about 6.5 m, using two boards both at each end of the cabin). Another problem is the required fixed installation. The ultrasonic transducers need free space in order to work properly. In aircraft cabins, that have various kinds of configurations, these positions need to be found and calibrated in every single case. If a removable detection system is chosen, there need to be fix points in the aircraft cabin where the system can be attached to.

Acknowledgements

Due to my inexperience in circuit design and especially the inevitable need to get into the topic of microcontroller programming, I would like to give special thanks to a couple of persons who gave their unsophisticated and much appreciated help in doing this project.

At first, thanks to Helmut Otte from the Department of Electronic Engineering at the HAW, who truly helped with the circuit design as well as some tests.

Thanks also to Dipl.-Ing Reinhard Breuer, who allocated his knowledge and equipment for etching the circuit board.

An indispensable help was the support by Mr. Watslavick from Microchip, who offered his time to help me setting up the microcontroller and getting started with programming.

Thanks also to Mike Gerdes for his logistical and financial support, as well as for proofreading the report.

References

- Gerdes 2010** M. GERDES & D. SCHOLZ: *Designing and Evaluating a Location Detection Solution for the Aircraft Cabin with Model based System Engineering*. Hamburg: Hamburg University of Applied Sciences Aero-Aircraft Design and Systems Group, 2010
- Kuttruff 1988** HEINRICH KUTTRUFF: *Physik und Technik des Ultraschalls*. Stuttgart: S. Hirzel Verlag, 1988
- Sanchez 2007** JULIO SANCHEZ & MARIA P. CANTON: *The Microchip PIC*. Boca Raton, Florida: Taylor & Francis Group, 2007
- Scholz 2011** URL: <http://www.fzt.haw-hamburg.de/pers/Scholz/PAHMIR.html>
(2011-02-04)

Appendix A

Main Source Code

```
/*----- I N C L U D E S -----*/

#include "p18cxxx.h"
#include "delays.h"
#include "stdio.h"
#include "stdlib.h"

/*----- D E C L A R A T I O N S ----- */

const char aaa[]="System Ready...";
const char bbb[]=" >Press button< ";
const char mess1[]="Measurement...";
const char mess2[]="Distance:";
const char error1[]="Error...";
const char error2[]="Timer overflow!";
const char error3[]="No valid meas.";

double f_b = 5496.97222;
double f_a = 0.2805556;
int disp1,disp2;

double range = 0;

const char IF[]="TMR1IF=";
const char value[]="TMR1=";
unsigned char stopflag = 0;
unsigned int timer1counts;
unsigned int timer1_h;
unsigned int timer1_l;
unsigned char XPORTB;
char Buffer1[20];
char Buffer2[20];
char Buffer3[20];
char Buffer4[20];
```

```

/*----- Function Prototypes -----*/
void result(void);
void high_isr(void);
void PWM_init(void);
void lcd_init(void);
void timer_init(void);
void LED(unsigned char a);
void delay_ms(unsigned int b);
void wait_for_signal(void);
void send_pulse(void);
void wait_for_pushbutton(void);
void LCD_WriteRegister(unsigned char c);
void LCD_WriteString(unsigned char *d);
void LCD_WriteZeichen(unsigned char e) ;
//char sprintf(char *str,const char *format, int x);

/*----- Interrupt Routine -----*/

#pragma code HIGH_INTERRUPT_VECTOR = 0x08 // High priority interrupt adress

void high_isr(void) // High priority interrupt routine
{
    _asm
        goto result // on interrupt goto void result
    _endasm
}

#pragma code
#pragma interrupt result

void result(void)
{
    T1CONbits.TMR1ON = 0 ; // Stop timer
    INTCONbits.TMR0IE = 0; // Disable timer 0 interrupt
    INTCONbits.RBIE = 0; // Disable Interrupt on Change PortB
    XPORTB = PORTB; // Read PortB to avoid another interrupt
    INTCONbits.RBIF = 0; // Clear PortB Interrupt on change flag bit
    stopflag = 1; // set flag for leaving loop in main()
    if(INTCONbits.TMR0IF) // If Timer overflow caused error
    {
        INTCONbits.TMR0IF = 0; // Clear Timer0 Interrupt flag
    }
}

```

```

TMR0H = 0 ; // clear timer0
TMR0L = 0 ;

LED('y'); // indicate Error
LCD_WriteRegister(0x01); // Clear Display
LCD_WriteString(&error1); // Write "Error..."
LCD_WriteRegister(0xC0); // Next Line
LCD_WriteString(&error2); // Write "Timer overflow!"
wait_for_pushbutton(); // Continue...

}
else // Signal caused interrupt
{
    INTCONbits.TMR0IF = 0; // Clear Timer0 Interrupt flag
    timer1_l = TMR0L; // Put timer0 value into variables
    timer1_h = TMR0H;
    TMR0H = 0 ; // clear timer0
    TMR0L = 0 ;

    timer1counts = timer1_h*256+timer1_l; // calculate timercounts

    range = (timer1counts - f_b)/f_a; //calculate actual range
    disp1=(int)range;
    disp2=(int)((range-disp1)*10);
    if (disp2 > 5) {disp1++;}

    if (range =0) // interrupt was released too early
    {
        LED('y'); // indicate Error
        LCD_WriteRegister(0x01); // Clear Display
        LCD_WriteString(&error1); // Write "Error..."
        LCD_WriteRegister(0xC0); // Next Line
        LCD_WriteString(&error3); // Write "No valid meas."
        wait_for_pushbutton(); // Continue...
    }
    else
    {

        LCD_WriteRegister(0x01); // Clear Display

```

```

// sprintf(Buffer1,"%i xxx",timer1counts);
// LCD_WriteString(Buffer1);

LCD_WriteString(mess2);
LCD_WriteRegister(0xC0);           // Next Line
sprintf(Buffer2,"%i cm",disp1);
LCD_WriteString(Buffer2);
wait_for_pushbutton();
    }
}

void main (void)
{
/*----- Configuring I/O Ports -----*/

TRISB = 1           // set PORTB as input
TRISC = 0 ;        // set PORTC as output
TRISD = 0 ;        // set PORTC as output

LED('y');         // Indicate Boot sequence

INTCON2bits.RBPU = 0; // enable PORTB internal pullups
INTCON2bits.RBIP = 0; // RB Port on Change high priority

TRISBbits.TRISB1 = 1; // PORTB bit 0 (connected to switch) is input
TRISBbits.TRISB4 = 1; // PORTB bit 0 (connected to signal in) is input

lcd_init();        // Initialize LCD
PWM_init();        // Initialize PWM
timer_init();      // Initialize Timer and interrupt

delay_ms(1000);   // 1s warmup

/*----- Endless loop from here -----*/

while (1)
{
timer_init();

INTCONbits.RBIE = 0; // RB Port Interrupt On Change Disabled

```

```

T0CONbits.TMR0ON = 0 ;           // Stop timer
TMR0H = 0 ;                       // clear timer1
TMR0L = 0 ;

stopflag = 0;                     // clear flag for waiting loop

LCD_WriteRegister(0x01);          // Clear Display
LCD_WriteString(aaa);             // Write "System Ready..."
LCD_WriteRegister(0xC0);          // Next Line
LCD_WriteString(bbb);            // Write " <Press button> "
LED('g');                         // Indicate system ready

wait_for_pushbutton();           // When button pressed, measurement starts

LED('r');                         // Indicate system busy
LCD_WriteRegister(0x01);          // Clear Display
LCD_WriteString(mess1);           // Write "Measurement..."
INTCONbits.TMR0IE = 1 ;          // Enable timer 1 interrupt on overflow
delay_ms(500);                   // Delay until Display finishes writing

send_pulse();                    // Sending Ultrasonic pulse
delay_ms(200);                   // Clearing time after sending pulse to avoid self-
interrupt

LED('o');                         // LED off to indicate system on standby and waiting
for reply

INTCONbits.RBIF = 0;             // Clear Interrupt on PortB Change Flag bit
INTCONbits.RBIE = 1;            // RB Port Interrupt On Change Enabled

T0CONbits.TMR0ON = 1;           // Start timer0

while(stopflag<1){}             // Loop ends when interrupt routine finished

    }
}

```

Appendix B

Source Code of Sub Functions

```

/* Includes */

#include "p18cxxx.h"

/*Definitions*/
#define PORTLCD LATD
#define TRIS_LCD TRISD
#define LCD_E PORTDbits.RD2
#define LCD_RS PORTDbits.RD0
#define LCD_RW PORTDbits.RD1
#define nop _asm nop _endasm           // verzögerung 0,25µs

/*Function Prototypes*/
void Delay1KTCYx(unsigned char x);
void Delay10TCYx(unsigned char x);
void delay100us(unsigned char x);

/*Delay functions*/

void delay_ms(int multiplikator)       //ms delay Funktion
{
while(multiplikator--) Delay1KTCYx(5);
}

void delay100us(unsigned char multiplikator)
{
while(multiplikator--) Delay10TCYx(50);
}

/* STATUS LED */

void LED(unsigned char toggle)
{
switch(toggle)
{
case 'o':                               //LED OFF

```

```

        LATCbits.LATC5 = 0;
        LATCbits.LATC4 = 0;
    break;

    case 'r':                //LED RED
        LATCbits.LATC4 = 0;
        LATCbits.LATC5 = 1;
    break;

    case 'g':                //LED GREEN
        LATCbits.LATC4 = 1;
        LATCbits.LATC5 = 0;
    break;

    case 'y':                //LED YELLOW
        LATCbits.LATC5 = 1;
        LATCbits.LATC4 = 1;
    break;
}
}

/*Timer*/

void timer_init(void)
{
    T0CONbits.T08BIT = 0;        // Timer0 16bit mode
    T0CONbits.T0CS = 0;         // Timer0 Internal OSC
    T0CONbits.T0PS0 = 0;        // Timer0 prescaler assigned
    T0CONbits.T0PS1 = 1;        // Timer0 prescaler set
    T0CONbits.T0PS2 = 1;
    TMR0L = 0x00;               // Reset Timer0
    TMR0H = 0x00;
    INTCONbits.TMR0IE = 0;      // Disable timer 1 interrupt
    INTCONbits.TMR0IF = 0;      // Clear timer 0 interrupt flag
    INTCONbits.GIE = 1;         // Enable global interrupts
    INTCONbits.PEIE = 1;        // Enable peripheral interrupts
}

/*
111 = 1:256 prescale value
110 = 1:128 prescale value
101 = 1:64 prescale value

```



```

100 = 1:32 prescale value
011 = 1:16 prescale value
010 = 1:8 prescale value
001 = 1:4 prescale value
000 = 1:2 prescale value
*/

```

```

/*PWM Signal*/

```

```

void PWM_init(void)
{
    PR2 = 0b00110001;
    T2CON = 0b00000101;
    CCP2CON = 0b00111100;
    T0CON = 0;
    CCPR2L = 0;
}

```

```

void send_pulse(void)
{
    CCPR2L = 25 ;           //PWM with 50% duty cycle
    delay_ms(50);          //50ms Pulselength
    CCPR2L = 0;           //PWM with 0% duty cycle
}

```

```

void wait_for_pushbutton(void)
{
    unsigned char Switch_Count = 10;    //Debounce counter initialization

    do
    {
        //Monitor switch input
        if (PORTBbits.RB1 == 0)
        {
            //Pressed state detected
            Switch_Count++;
        }
    }
    else
    {
        Switch_Count = 0;
    }

    delay_ms(10);
} while (Switch_Count < 5);

```

```

    }

/*
LCD DRIVER
-----
PortPin: Display
RD0      = E
RD2      = RS
RD3      = R/W
RD3..RD6 = D4..D7
*/

void enable(void)
{
    LCD_E = 1;
    delay100us(3);
    LCD_E = 0;
}

void LCD_WriteZeichen(unsigned char zeichen)
{
    unsigned int pointer;
    unsigned char x;
    delay100us(200);
    LCD_RW = 0;          // LCD Write Mode
    LCD_RS = 1;         //LCD Command Mode

    x = zeichen & 0xf0;
    pointer = (x>>=1);
    x = x | LCD_RS;
    PORTLCD = x;        // Higher nibble
    nop;
    enable();
    delay100us(50);     // Wait 5ms after 1st nibble

    x = zeichen;
    x = (x << 4)& 0xf0; //niederwertiges nibbel
    pointer = (x>>=1);
    x = x | LCD_RS;
    PORTLCD = x;

    enable();

```

```

delay100us(50);          //5ms warten nach den 2.nibble
LCD_RS = 0;
}

void LCD_WriteString(unsigned char *String)
{

unsigned char zeichen;

zeichen = *String;

while(zeichen !='\0')
{
LCD_WriteZeichen(zeichen);          //Zeichen am LC-Display ausgeben
String++;
zeichen = *String;
}

}

void LCD_WriteRegister(unsigned char data)
{
    unsigned char x;

delay100us(200);

x = data & 0xf0;

PORTLCD = (x >>= 1);

enable();

delay100us(50);          //5ms warten nach dem 1.nibble
x = data;
x = (x << 4)& 0xf0;          //niederwertiges nibbel um 3 schieben

PORTLCD = (x >>= 1);

enable();
delay100us(50);          //5ms warten nach dem 2.nibble
}

```

```

void lcd_init(void)          //LCD initialisieren:

{
unsigned char pointer;

TRIS_LCD = 0x00;           //LCD-PORT = Ausgänge
PORTLCD = 0x00;           //Daten und Steuerleitungen LOW
delay100us (150);
delay100us (150);         // warte über 30ms
delay100us (100);

pointer = 0x30;
PORTLCD = (pointer>>=1);

enable();

delay100us(50);           //warte 5ms

enable();                 //Enable (LCD)

delay100us(2);           //warte 200µs
enable();                 //Enable (LCD)

delay100us(2);           //warte 200µs
pointer = 0x20;           //Interface auf 4-Bit setzen
PORTLCD = (pointer>>=1); //High nibble runter setzen
enable();                 //Enable (LCD)
LCD_WriteRegister(0x28); //2-zeilig, 5x8-Punkt-Matrix
LCD_WriteRegister(0x08); //Display aus
LCD_WriteRegister(0x01); //Display löschen
LCD_WriteRegister(0x06); //Kursor nach rechts wandernd, kein Display-Shift
LCD_WriteRegister(0x0f); //Display ein [ 0x0C = ein, 0x08 = aus ]
LCD_WriteRegister(0x0C); //Cursor Off
}

```

Appendix C

Quick Operation Manual

To get started, the following steps need to be done:

1) Set up both boards like in Figure C.1:

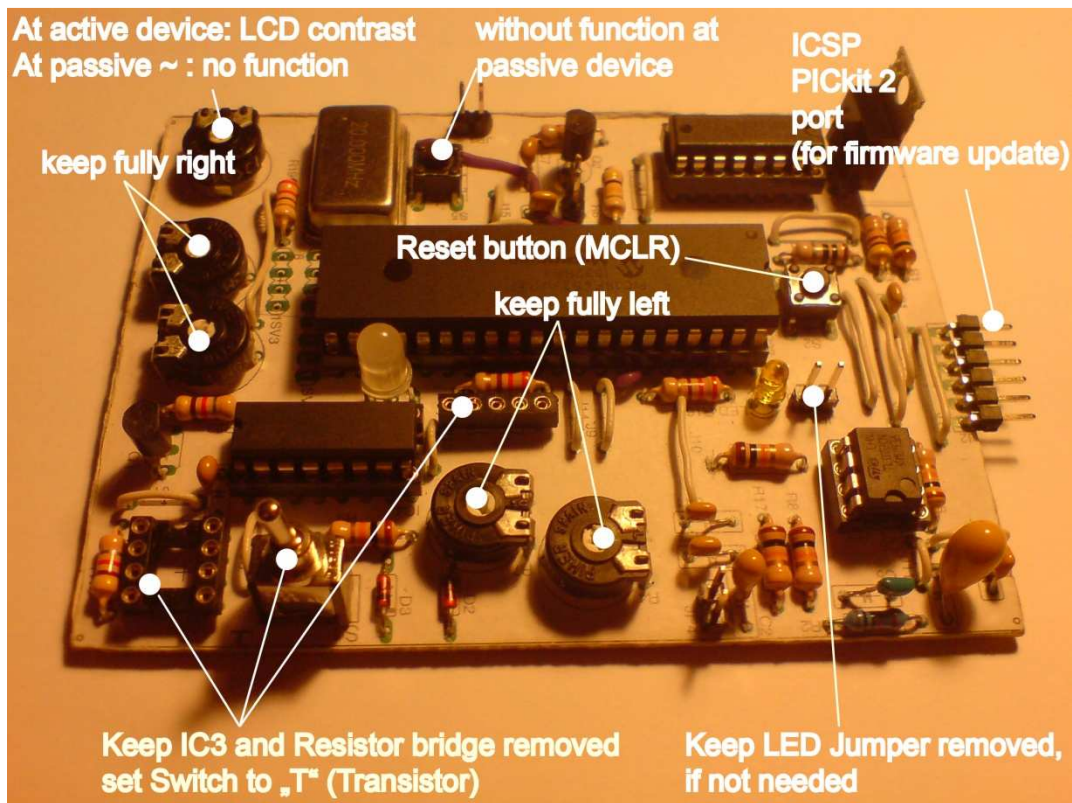


Figure C.1 Board configuration

2) Connect equipment like shown in Figure C.2:

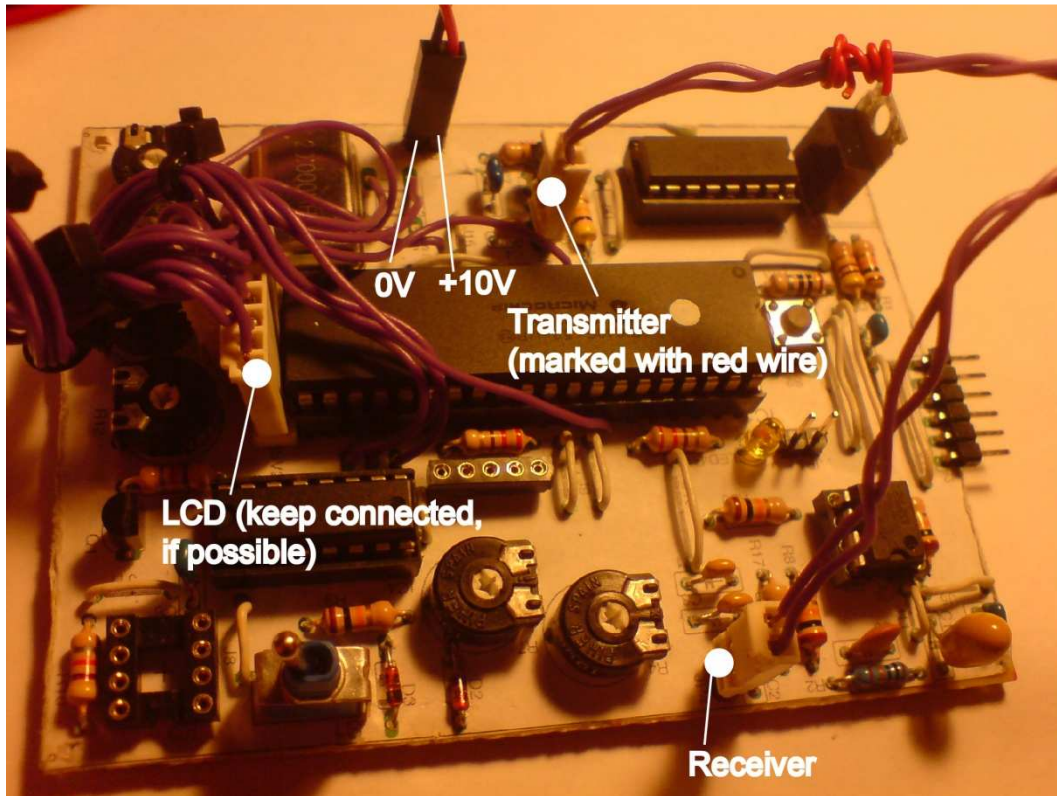


Figure C.2 Connection of equipment

3) Place the two transmitter/receiver devices so that they are facing each other.

4) Press button of active device to start measurement