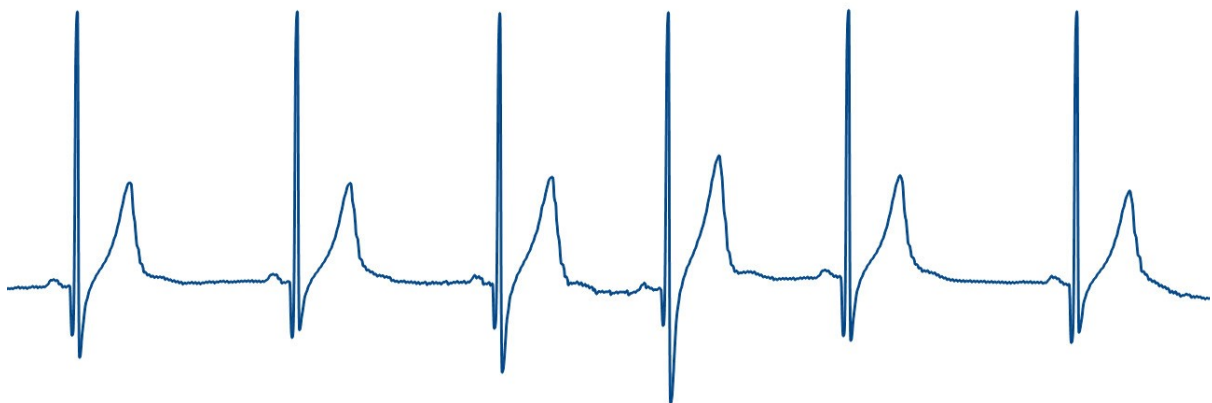# Final thesis

as state-certified technical engineer

2014 / 2015

„ECG"

Construction and analysis of a multi-stage, µC-controlled ECG measuring circuit for measuring, logging and evaluating the so-called "cardiac voltage curve"

Preface

Within the framework of the 2-year technical college, each student is required to complete a so-called technician's thesis as part of the final examination. In addition to the normal school lessons and the vacations, the student has to do this work independently.

Due to the fact, that I am personally very interested in several parts of electronics, and I have been employed at an EMS provider before, the following project was considered by me and was worked on by me from September 2013 to July 2014:

Construction and analysis of a multi-stage, µC-controlled ECG measuring circuit for measuring, logging and evaluating the so-called "cardiac voltage curve"

The topics of this thesis include, among others, circuit design based on the above-mentioned problem, its analysis, the technical design of printed circuit boards with CAD systems, software programming for Windows and an ARM microcontroller, as well as the familiarization with CAD systems, parts databases and the production process.

Note:

This document has been abridged for publication in several points.

Topic

1 Project

1.1 Short project description

Since I already had to do with various assemblies, such as medical technology, through my full-time job, and I am also interested in measurement technology, the project "ECG" caught my eye on a hobby website in the field of electronics. Quickly my interest was aroused and the project was set as a technician's work. The scope of the project includes to display the heart rate of a person as a line diagram, as it is often seen at the doctor's office or on TV. Since this project is connected to the human body to measure the signals, it includes a safe galvanic isolation of the devices that operate above the maximum permissible contact voltages. A sensor is designed that measures the signals as a compact, battery-powered assembly and sends the data by radio to a main unit. This unit stores the received data on an SD card or sends them via interface to a PC.

1.2 ECG Basics

The so-called ECG (electrocardiogram means translated "heart tension curve") is used in medical technology to diagnose problems that affect the human heart. It can also be used to measure the human pulse. Each contraction of the heart muscle is caused by an electrical "excitation" that originates from the so-called sinus node. This causative electrical voltage can be measured on the surface of the skin. By analyzing the heart voltage curve, conclusions can be drawn about heart rate, rhythm, position, and the activity of the atria and ventricles can be determined. To record the electrical signals, contact surfaces on the body (electrodes) are used to transmit the electrical signals to the measuring system.

The first possible measurment method is a derivation model according to "Wilson", in which up to 6 sensors are attached horizontally to the chest, which measure the most diverse occurring tensions. As a reference, a sensor on the extremities serves here as a zero point. This type of measurement, is used in a simpler way in this technician work.In addition, there are derivation models that use purely extremities (such as arms and legs) as measurement points, these were named after a Mr. "Goldberger".

The sequence of a heartbeat looks like this:

1 **P-Wave**

This shows the first excitation of the atria, which is triggered by the occurring voltage from the sinus node. Duration: 50ms – 100ms

2 **PQ-Interval**

Time between excitations. Duration: 130ms – 200ms

3 **QRS-Interval**

This section corresponds to the

Excitation propagation in the heart chambers, the so-called "ventricular excitation". Duration: 50ms – 100ms

4 **ST-Interval**

Indicates the onset of excitation regression of the ventricles.

5 **T-Wave**

Excitation regression of the ventricles

6 **U-Wave**

Post-variation of chamber regression, very slight to not seen.

2 Planning

2.1 Needed circuits

- **Microcontroller:**

  A frequently used standard type should be used as controller, especially said, the STM32 from ST Microelectronics (ARM Cortex M3). The reason for this selected controller is, that it supports a large number of programming adapters and debuggers. Required interfaces of this STM32 are several UART interfaces for communication, ADC channels for voltage measurements, an SPI interface for controlling an SD card, as well as the "Single Wire Debug" interface (SWD) required for programming.

- **Amplifier for the „ECG Body-Voltage"**

  For this purpose, an amplifier circuit is used which consists of an instrumentation amplifier and several filter stages (low-pass filters).
  These filter stages are important to filter interference signals of the environment, which are measurable on the human skin and falsify the measurement result.
  (50Hz interference from fluorescent tubes for example).
  The instrumentation amplifier has a higher input resistance than "normal" operational amplifiers and is therefore ideally suited as an input circuit.

- **Radio Interface**

  A radio interface is useful to ensure safe, galvanic isolation. This should offer a sufficient data rate and range to be able to move further away from the receiving unit.

- **ESD Protection**

  As ESD protection of the input circuitry of the instrumentation amplifier (sensor board) as well as the USB interface (interface board), a protective circuit is required that quickly and efficiently dissipates voltage peaks.

  Especially since the human body can quickly "charge" with several kilovolts of static voltage, such a protective circuit is very important.

- **Power Source**

  In order to be able to supply all circuit parts with voltage, a power supply unit must be designed that provides sufficient current and is also sufficiently dimensioned in terms of performance.

  Since the MCU and all other analog and digital components require the current "3.3V" standard, but do not need an excessive amount of current, the dimensioning is very manageable.

- **SD Card**

  To realize a "data logger" function on a SD card, it must be controlled accordingly. Since all SD cards support the so-called "SPI" interface, which is already integrated in the MCU, this control can be implemented very easily in terms of hardware.

- **USB Interface**

  An "RS232 to USB" interface module is provided on the interface unit as an interface to the PC. This enables a connection via USB to a PC with a small external circuit. There this hardware is then "simulated" as a virtual COM port and can be used in many ways.

2.2 Part selection

- **MCU selection:**
  Since the STM32 as the main unit for the control is already determined in advance, it only had to be determined which exact model should be considered for the two circuit parts. After comparing several types, I selected the STM32F103C8T6, which provides sufficient memory with 64KByte flash memory as well as 20KByte RAM memory and has all required interfaces such as SPI and USART already integrated.

- **Programming adapter**
  As programming interface the STM32 uses a so-called "Single Wire Debug" interface (SWD for short), which is part of most JTAG programmers on the market. The full JTAG interface makes no sense here, because only one controller has to be programmed.
  Here, the extremely inexpensive STLink V2 programmer from ST Microelectronics was chosen, which is easily supported by the IDE for programming and debugging.

- **Selection of the amplifier circuit**
  It was very important to pay attention to the exact tuning and the interaction of the components. The "INA326" was selected as the instrumentation amplifier, since it allows an externally adjustable gain setting. Furthermore, the "OPA2336" component was selected for the low-pass filters and other op-amp circuits. This already contains two operational amplifiers, which reduces the space required.

- **Selection of the radio modules:**

  This choice was more difficult than with the other circuit parts, since I personally have little experience with this technology. At the beginning, I chose the inexpensive "RFM12" radio modules, which transmit on the 868 MHz band and are controlled via SPI interface. In the subsequent test, however, it became apparent that the transmission stability left a lot to be desired. There were repeatedly incomplete and delayed transmissions, which are not acceptable for the planned operation of the project. Furthermore, the complete "overhead" (transmission frame, checksum formation) had to be taken care of by myself.

  Then I tested the somewhat more expensive AMB2520 modules from Amber Wireless, which operate on the 2.4 GHz band. The advantage of these is firstly the simpler management by means of UART interface (present in the MCU) in contrast to the SPI interface and secondly the control of the wireless module takes care of all the processing (overhead) by itself, so that only the raw data to be sent must be sent by interface. After a quick and extremely satisfactory test, this module has now been specified by me for use. Overall, this product clearly makes a more professional impression than the inexpensive wireless modules.

- **Selection of the USB Interface**

  As interface module a frequently used component "FT232RL" was used, which can also be found in almost every serial to USB converter on the market. A simple circuit and no programming effort speak for this choice.

- **Selection Body sensor**

  To measure the body signals, at least 3 points are needed, two on the chest and one as a reference on one leg. For this purpose, there were already belts from a famous fitness accessories manufacturer, where the contacts are led to the outside by button attachments. This allows the belt to be used for technician work without any problems. Originally, this belt was intended to be used with the heart rate monitor available as an accessory, but it could be used for this project without any problems.

- **Selection Power source**

  Various methods have been implemented to power both assemblies.

  A battery pack is to be installed in the housing of the interface circuit, which operates the circuit without an external power supply.

  (standalone operation with function data logger on SD card).

  However, when operating in conjunction with the PC, this circuit is to receive power via USB. For this purpose a "Power Switch" device is used, which always uses the higher available voltage as supply voltage.

  In the housing of the sensor circuit, the same battery pack was chosen.

  Since the same housing is used twice for the modules, space problems are eliminated. The required 3.3V power supplies on both circuit parts consist of simple 3.3V linear regulators with low voltage drop, so-called

  LDO or "low dropout" regulators.

- **Selection ESD Protection**

  Here a diode array consisting of several diodes connected in series and a suppressor diode was selected as ESD protection circuit for the USB input and the 3 ECG measurement inputs. This component is capable of dissipating peak voltages on the supply as well as on the measurement lines.

- **Housing selection**

  As housings two plastic housings were selected, which do not need much place, are inexpensive, in which the selected circuit parts can be well accommodated, and from plastic besides do not shield the radio signals.

  The case also specifies 4 mounting holes, which must be provided later in the PCB layout. The board is later screwed into the housing with these holes.

2.3 First test

As a test of the basic OP circuit as well as the power supplies and interfaces, the planned circuit was built up in advance on an electronics plug-in board.
This offers the advantage of being able to test all components quickly and largely without soldering in the assembly, and to be able to reuse them afterwards.
This was done once for the circuit part sensor and once for the circuit part interface. However, in order to be able to attach SMD components, such as the STM32 or the radio module, to the plug-in board, these had to be soldered onto small adapters.
This was unavoidable for this.
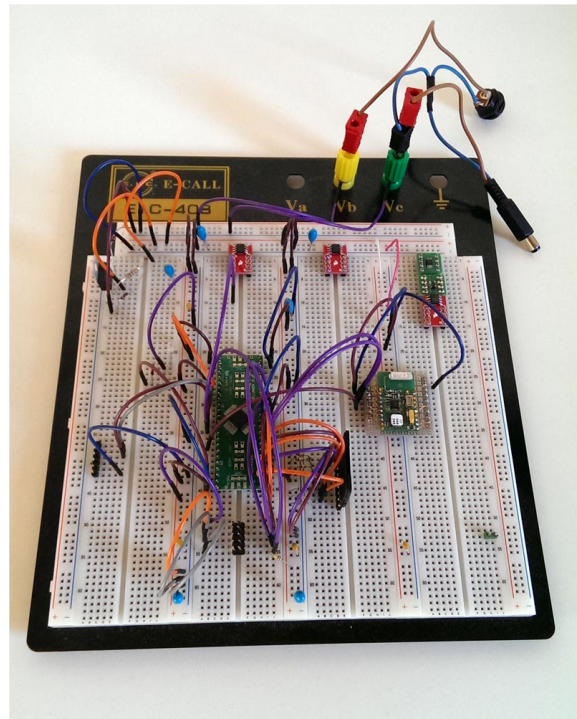
The biggest problem with this approach was that all electromagnetic interference from the environment, such as from fluorescent tubes or other electronic devices, was "well" captured. This occurred due to the "free" wiring, which was realized on the plug-in boards by means of wire bridges.
As a finished board, this behavior should already show significant improvement without special care.
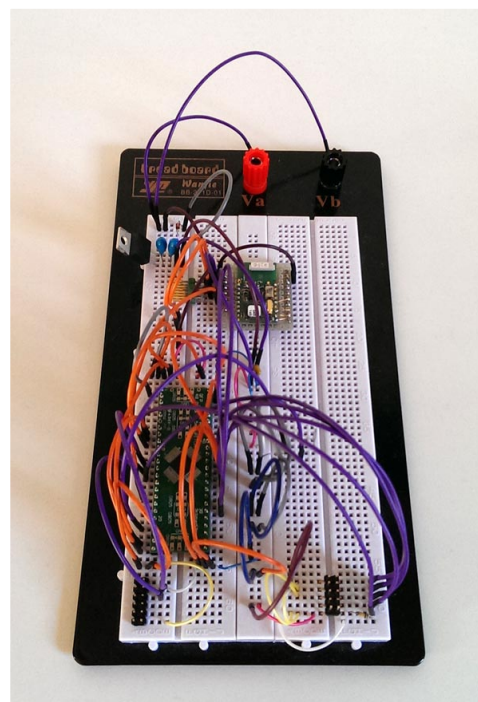
The pure OP circuit was tested with a calibrated digital multimeter from Agilent Technologies (34410A). This showed how promising the measurement circuit is and how accurately the individual heartbeats including the individual sections of the heartbeats can be seen.
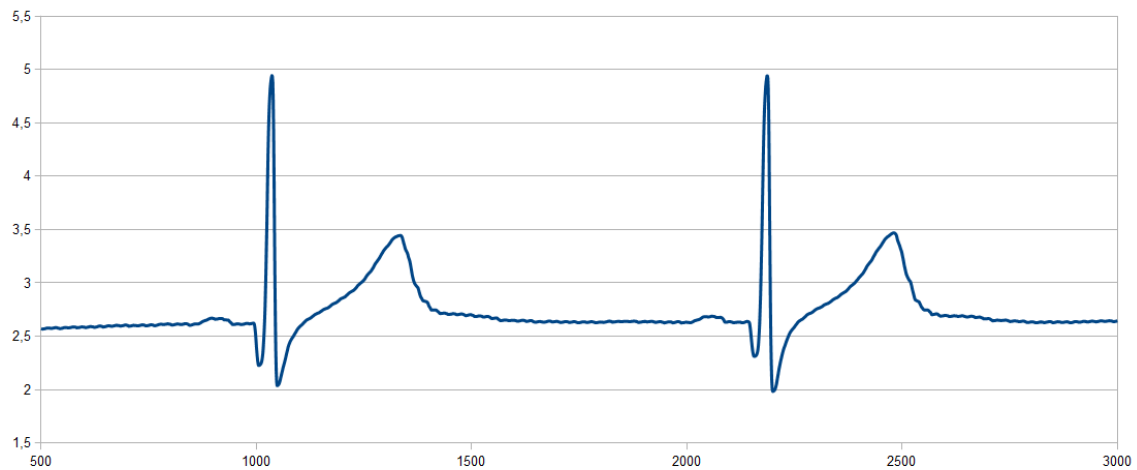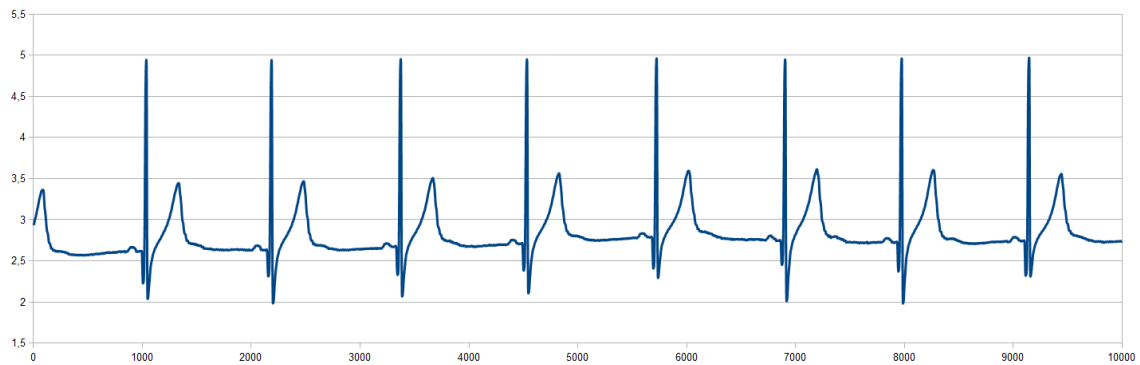
An inexpensive, commercially available "evaluation board" from the company Olimex was also used for the test, on which an STM32F103R8T6 is used, which is very similar to the STM32F103C8T6 used in this engineering work. Only the RAM and the flash memory are larger on the Eval board, and the latter has more pins (larger package).

Because of the used ST code libraries there is no difference necessary in programming, only the correct MCU type has to be selected once in the development environment.

This board includes already many interfaces like SPI, UART, CAN, USB, and more available, so that this can be used for program test very fast. In the circuit diagram of this board also one or the other own circuit could be checked or taken over.

3 Device production

3.1 Schematics creation

3.1.1 SWD & Serial interface

The main interface to program the selected microcontroller is the already selected "Single-Wire-Debug" interface on both circuit parts.

Here a 10pin connector is used, which provides the necessary lines for programming and debugging. This was provided as a 2x5pin header in the standard 2.54mm grid to offer fast and easy to handle connection. This was also provided for the required communication (USART) interface to be able to communicate with the board serially in case of emergency or for test purposes. Here a 3pin header is sufficient.

For both interfaces a separate adapter will be needed later.

In order not to "build in" both circuit parts unnecessarily, these pin headers were defined as layout variants and not assembled. The goal is to be able to plug in the board with a self-built adapter without these parts taking up space when not in use.
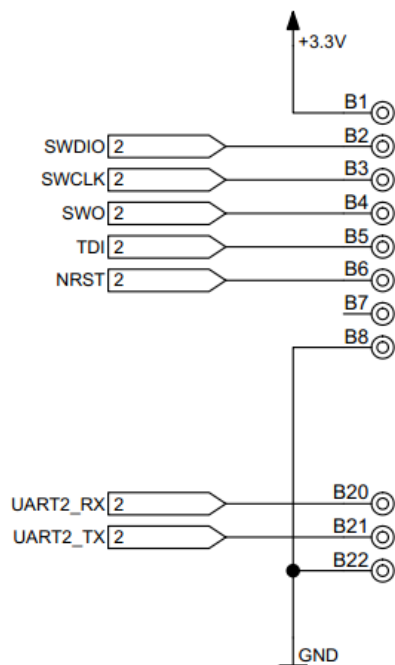
3.1.2 Mikrocontroller

The "STM32F103C8T6" microcontroller used belongs to the ARM Cortex M3 series. Independent of the exact type, which differs in flash and RAM size, as well as in interface options, this essentially consists of a Harvard architecture, which has 2 internal bus systems. This feature allows this type, compared to the smaller types realized with a von Neumann architecture, to load data and commands in one instead of two data cycles.

For this project, the "LQFP44" package variant was selected, which offers all the required functions with very small external dimensions. Used here are the SPI interface (only interface for SD card), UART1 (radio module) and UART2 (monitor PC) for communication, several IO ports for configuration and LED display, several analog inputs for voltage measurement, the connection for the external 16 MHz quartz, reset connection, and the necessary connections for the power supply.
The basic structure was taken from already used layouts and discussed again with my supervisor.

3.1.3 Power source

Here the choice fell on modern "LDO" voltage regulators, so-called "low dropout" regulators. These "low dropout" regulators, which provide the 3.3V voltage required by the MCU with little voltage drop and thus with little power loss.

It is important to have a well dimensioned stabilization of the voltages on the input and output sides of the linear regulators to compensate for fluctuations.

On the interface board there is an additional voltage switch in front of the linear regulator, which supplies either the primary USB voltage or the secondary battery voltage to the regulator. The switch decides for the higher of the two voltages. The reason for this is that the battery is not unnecessarily stressed when the module is connected to the PC.
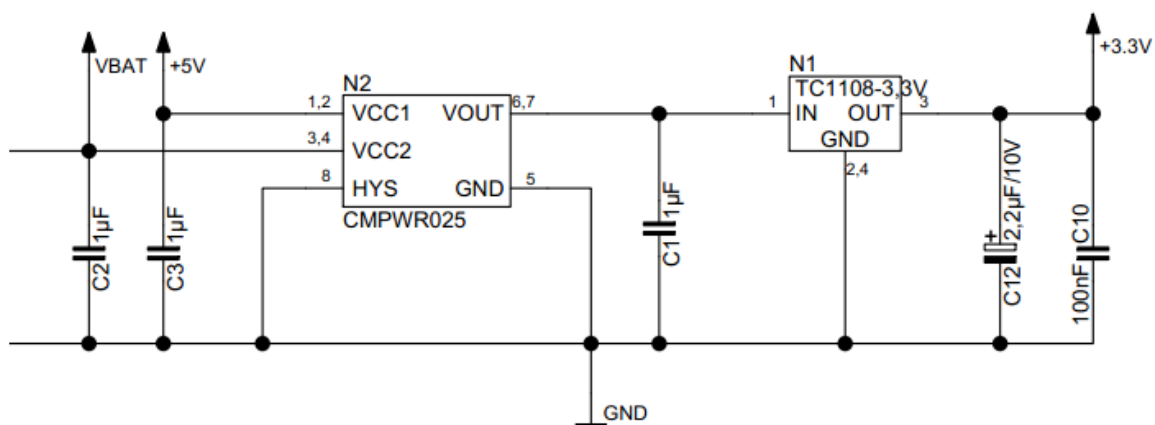
3.1.4 Radio module

For the radio module, a CAD symbol was designed especially for this project, which has all the required connections. Only the voltage connection for supplying the component with 3.3V and the UART interface for communication are required.

The unused interfaces are wired internally in the component as "default" and can therefore remain completely unconnected from the outside.
This wiring is the same on both circuit parts.
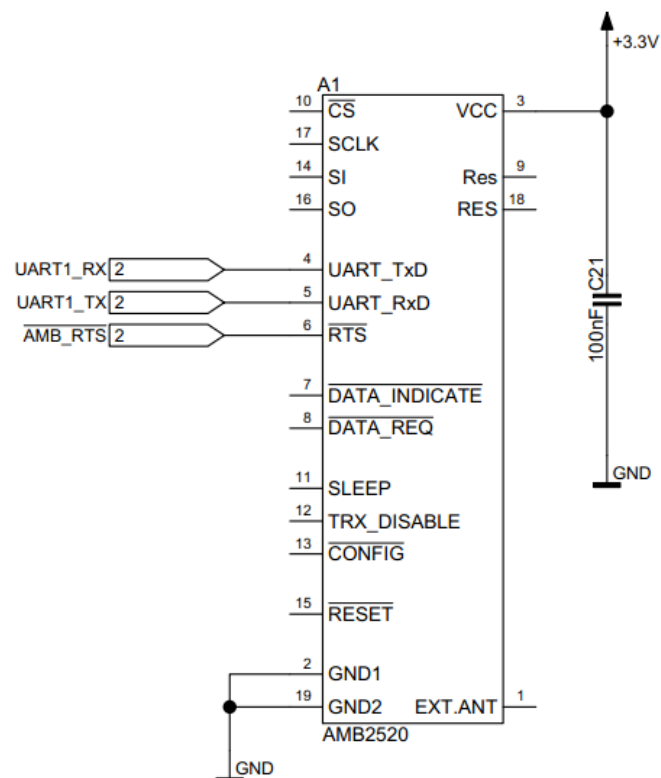
3.1.5 SD Card

An SD card is provided on the interface board for storing measured voltage values. This SD card is controlled via an SPI interface, which is already permanently integrated in the STM32 and therefore requires no additional external circuitry (except for resistors and capacitors).

This is realized in the form of a microSD card, which can be plugged into a soldered socket. This socket has the additional advantage that a switching output can be used to determine whether a memory card has been inserted or not.

The circuitry requires some "pull-up" resistors between the MCU and the socket, as well as a capacitor to protect the rather sensitive SD card from voltage fluctuations.
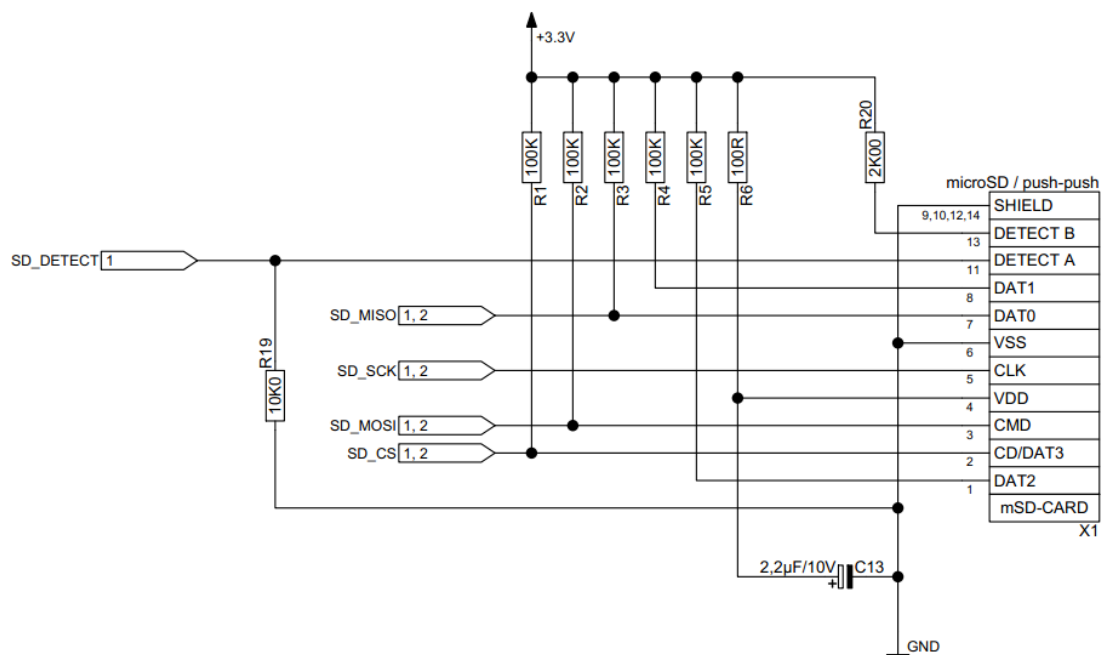
3.1.6 ECG circuit

In the run-up to all measuring circuits, the analog input signal from the human body must be protected against the module. Since several kilovolts of voltage can quickly build up on the human body, it is possible to permanently damage sensitive measurement inputs of electronic components. For this purpose, a diode array is used, which was originally intended for the protection of a USB interface.
Internally in this component, there are several diodes which can reduce voltage peaks of the measurement inputs compared to the supply voltage.
In addition, a TVS (suppressor) diode is integrated, which can dissipate peaks on the supply voltages itself at lightning speed.
The input of the measuring signals is done via miniature BNC sockets, which pass directly to the protection circuit as output.
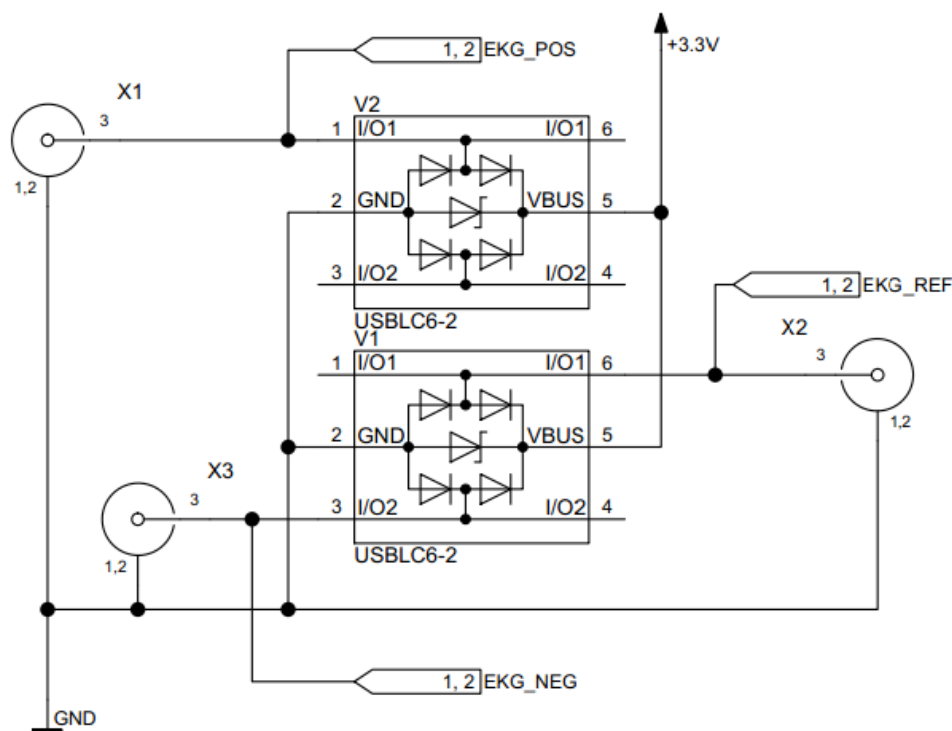
The actual measuring circuit of the ECG signal follows at the output of this protective circuit. This is by far the most complex circuit part on the sensor board. This consists of an INA326 instrumentation amplifier, which is set to a gain of 5 by means of an external resistor and capacitor circuit, and 2 low-pass filters, which are implemented using OPA2336 operational amplifiers. Once the measured input voltage is amplified by the instrumentation amplifier with very high input impedance, and this output signal is freed from noise via the low-pass filters. A stable voltage is required as a reference on which the basic signal is based, which in this case is provided by a precision voltage divider.

The output of this circuit goes directly to one of the analog measurement inputs of the STM32 MCU to be measured.

3.1.7 USB – RS232 Converter

An interface is required for communication with the PC system. For this, the serial interface (COM port) was chosen, because it is easy to handle.

A modern variant is available in the form of a converter, which converts a UART interface to a USB port with a small external circuit.

On the PC, this port appears as a so-called "virtual COM port" and behaves exactly the same as a real, physically existing COM port on the PC.

This procedure connects the serial interface, which is very easy to handle on the one hand, with the universal USB interface, which can no longer be imagined without.

Since the power supply can also be ensured to a certain extent via USB, the solution is perfect for connection to a PC.
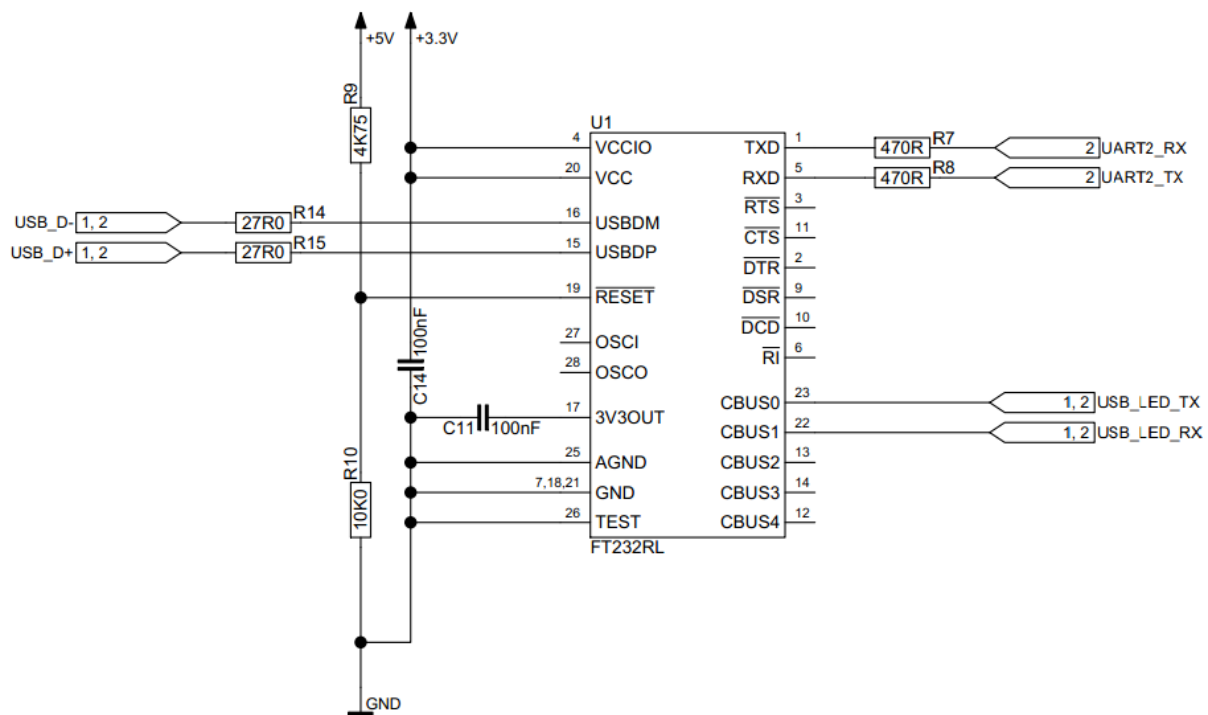
3.1.8 USB Input / Protection

The same protective circuitry is used for the required USB interface as for the ECG circuit. The only difference here is the connected supply voltage. Since 5V are provided by the USB port, the data lines are protected against this. As in the measurement part of the sensor, the protection circuit can quickly filter voltage peaks and thus protect the USB input.

A USB mini-jack in format B is used as the input, as this connection is now largely standardized in digital cameras, mobile hard drives and other devices. The connector includes the supply voltage with +5V, ground, data - and data +, as well as an ID connector, which, however, is no longer used today. In addition, the housing of the USB connector is sufficiently shielded via the connection to ground.
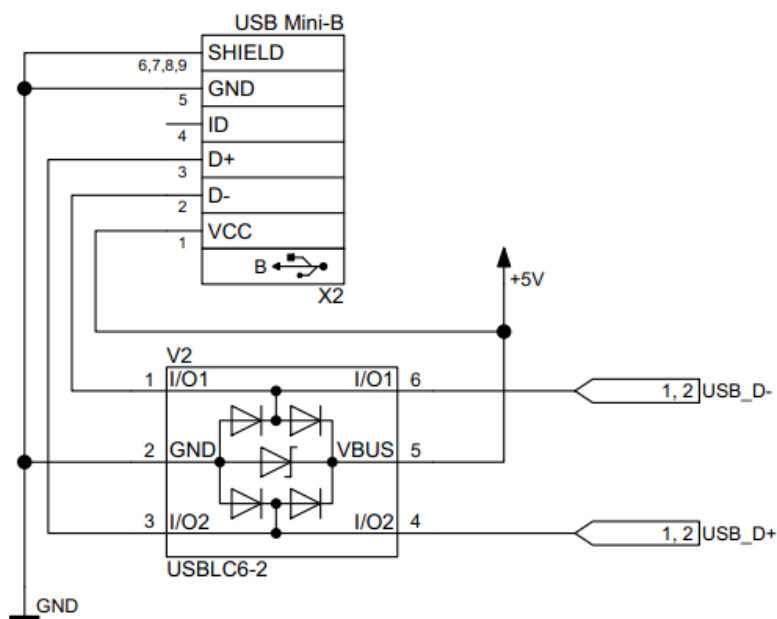
3.2 Layout creation

3.2.1 Requirements

Special layout defaults are important for the design of the PCB in the layout.
These are various settings with which the layout editor supports the user and checks for possible errors.
If these settings are chosen correctly, errors can be avoided in advance and the design can be improved in terms of EMC behavior. On the one hand, these defaults are applied from the default settings for the schematic, and in addition, further specifications are required here.

These specifications include trace width, trace spacing, diameter and size of vias, distance from objects to traces, distance to the edge of the board to be planned, and many other settings.

These default settings are called "Spacing Rules" in the layout and are set once for standard traces and once for traces of the power supply.
The latter should be dimensioned larger than standard traces because they carry current.

Design	Net Class - StdClass						
	Track	Pad	Via	Testpoint	Mounting Hole	Copper	Text
Track	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Pad	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Via	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Testpoint	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Mounting Hole	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Copper	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Text	0.200	0.200	0.200	0.200	0.200	0.200	0.000
Board	0.200	0.200	0.200	0.200	0.200	0.200	0.000

In addition, the selection of the BNC sockets on the sensor board resulted in the requirement to adapt the contour of the board to them. Since these are plugged onto the side of the layout and soldered there, this "edge mount" attachment requires suitable contour processing. This adaptation is easily possible and quickly done in the layout part of the CAD program. In the data sheet of the sockets, the exact specification of the manufacturer was found in order to incorporate these into a layout.

RECOMMENDED PCB LAYOUT
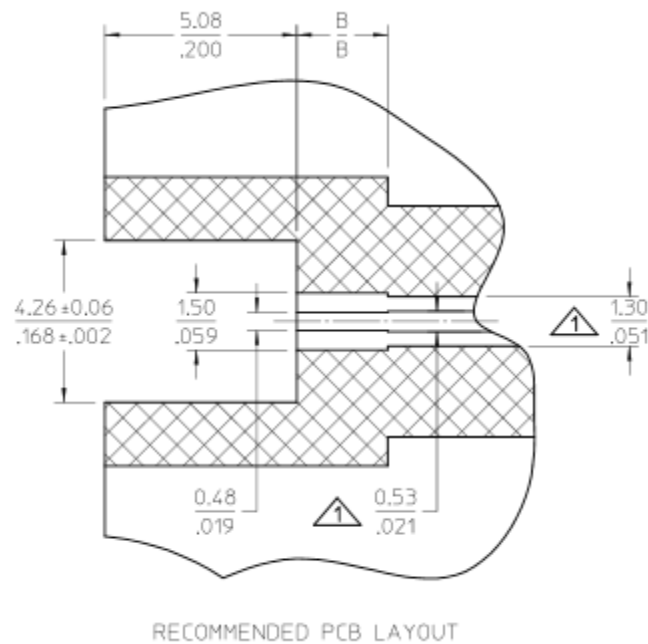
Furthermore, the housing, which was selected in advance for this work, determined the complete outer contour of the printed circuit boards.

3.2.2 Part placing

Before you can start laying the tracks, you must first place the components on the board in the CAD system. It is important to ensure that the components are not arranged in a jumbled fashion, but in a logical order. Then it is equally important in which environment of other components they are placed. For example, radio transmitters and receivers must not be disturbed in their function if other components are placed under them, on the underside of the board. The optical component should also play a role, albeit a subordinate one.

The top and bottom sides of the board were available for placement, but without any effort it was possible to place all components purely on the top side.
This clearly benefits manufacturing, as no additional manufacturing step is required for machine placement on the second side.

In both designs, the radio transmitter/receiver was provided in the direction of the outer side of the board to also enable lateral radiation of the radio signals.
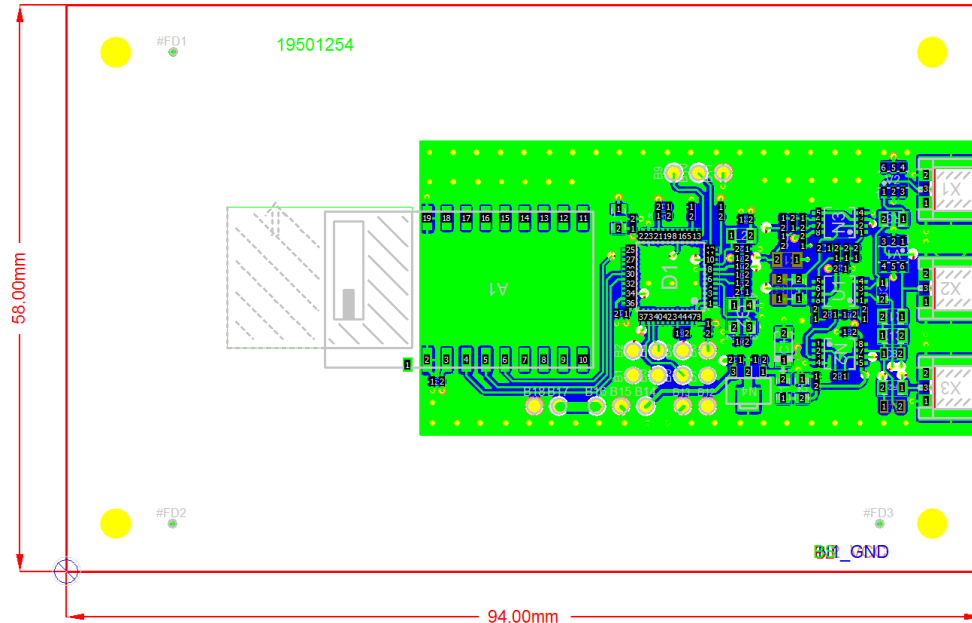
Likewise it had to be considered with the placement that the board has 4 fastening holes, with which it is screwed in the housing. So there should be no components around these holes to avoid complications with the plastic in the case.

3.2.3 Part connection

Once all the components are in place, you can start routing the tracks. In this step, the CAD program already takes a lot of steps from the user, because based on the previously defined specifications, the program now decides whether the current trace to be routed is only a simple signal line or a power supply line. Based on this decision, the set width of the trace is automatically selected. This allows the user to concentrate on the essentials. Afterwards, each trace can be manually changed in width, class and design. If components have already been connected to each other, it is also possible to move them again manually. In this case, the trace follows the component until it is moved again.

When clicking on a connection, the CAD system visually indicates the target to the user, so that the user can quickly track from where to where he has to lay the trace.

So-called "vias" are required for many traces. These are vias that penetrate through several layers in the layout to get signals from one layer to another. With this measure it is possible to cross signals or voltages. This is not possible with a single-sided layout.

3.2.4 Signal plane creation

Once all components are placed and connected, the remaining space on the board can be filled with signal areas. These have the function to spread supply voltages over a wide area on the board on the one hand, in order to avoid long traces. Furthermore, the "cross-section" of these areas is much larger than that of a single trace, so the resistance is reduced. This procedure was followed on both boards with the supply voltages. The inner layers are mainly used to route supply voltage and ground, the outer layers as signal layers. After routing, the free space on the outer layers, which were used purely for signals, can now be "flooded" with ground. This significantly improves the EMC behavior of electronic circuits. In addition, the resulting multiple ground planes were connected via many vias across multiple planes, again to increase the cross section and get the smallest possible resistance.
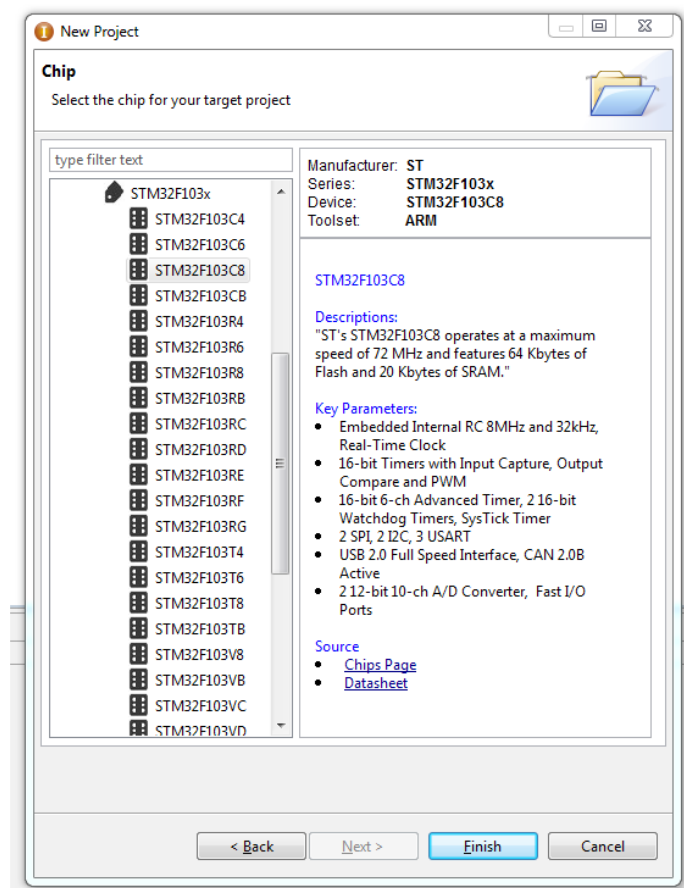
4 Software

4.1 Microcontroller

4.1.1 STM32 Development Environment

To develop a program for the microcontroller used, a system is required that can translate instructions in the form of source code into usable machine code using a compiler and linker. For this purpose, the easy-to-use and freely available open source development environment "CooCox" is used. This software is an advanced environment for ARM Cortex M4, M3, M0 and M0+ based microcontrollers. All tools are already included in it to develop, debug and test software for this hardware. The main advantages of this IDE is the free availability in contrast to the IDEs of other manufacturers, the associated, not limited development scope, Internet-based extensions of software libraries, the integrated "CooOS" Realtime Operation System and the wide support of debuggers and usable microcontrollers. The "STM32F103C8T6" chip to be used is already supported by the development environment out of the box. The also to be used debugger "STLink V2" is supported immediately without additional software.

4.1.2 Software structure

The software for this project consists of the following modules:

- **„CMSIS" – Cortex Mikrocontroller Software Interface Standard**
  The program for this technician work consists of the following program
  modules: The CMSIS library is a so-called hardware abstraction layer, which
  is to decouple the user as far as possible from pure hardware access. This
  functions as a driver library for the user, in the form of easy to handle, callable
  functions, with which the hardware can be addressed. Thus hardware
  accesses are simplified.

- **„Standard Peripheral Library" of ST Microelectronics**
  In addition to the CMSIS library, the Standard Peripheral Library supports the
  user even further by bundling processes for many of the functions provided
  and relieving the user of work. In conjunction with CMSIS and this library, for
  example, communication via SPI or memory access is possible with very few
  commands.

4.1.3 Configuration

To operate the designed board, the microcontroller must be configured for use. It is important to note that all pins of the MCU have different settings. As a basic configuration most of the pins have the state I/O, which defines this as input/ or output. This can be set more precisely in the software. Many connections also offer so-called alternative functions, which can be activated by means of "Remap". For example, 2 pins can become an SPI interface, or a USART interface.
The following connections (except power supply) are assigned as follows:

Pin	Sensor	Interface	Used as	Description
5	X	X	OSC_IN	Clock input
6	X	X	OSC_OUT	
7	X	X	RESET	N_RESET
44	X	X	BOOT0	Bootloader selection
10	X		ADC12_IN0	„ECG" voltage measurement
11	X	X	ADC12_IN1	3,3V measurement
12		X	USART2_TX	Serial connection PC
13		X	USART2_RX	
14	X	X	Input	Hardware selection Sensor /
17	X	X	Output	N_RTS for radio module
30	X	X	USART1_TX	Serial connection radio module
31	X	X	USART1_RX	
34	X	X	SWDIO	Programming interface
37	X	X	SWCLK	
38	X	X	TDI	
39	X	X	SWO	
21	X	X	Out	Status LED

4.1.4 Clock

To operate the MCU, a clock is needed. This can be generated with an internal 8MHz R/C oscillator (2% accuracy), or an external crystal can be connected. The latter improves the stability in case of temperature fluctuations, therefore this was used in the technician's work. A 16 MHz crystal is connected, with whose help the STM32 works internally over a multiplier with 72 MHz on the "SYSCLK".
From this "main clock" then various clocks for the different buses are derived.
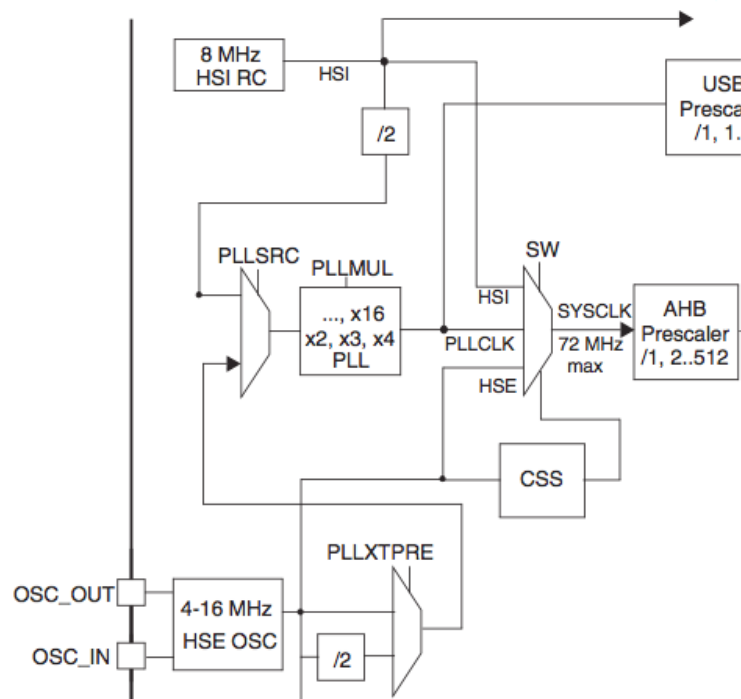
4.1.5 I/O-Ports

To activate the I/O ports in the STM32, several steps are required.
First of all the so called "Advanced Peripheral Bus" APB2 must be supplied with a clock (PCLK2), on which the IO ports hang. Then a "switching speed" can be defined for the ports, if one of these ports is set as output. The last step defines the exact ports including the direction in which they should work.

4.1.6 USART-Interface

The USART interface is the so-called serial interface of the microcontroller. With the help of this interface, both boards communicate with the PC via USB converter and with the radio module. These interfaces do not work like the UART interface to the PC with levels of up to +/- 15V, but with TTL levels of 3.3V, which are compatible for the MCU. In the case of the interface board, this 3.3V TTL level is converted to USB signals, and made available on the PC as a "virtual" COM port. For configuration, this interface requires an agreed speed between the end devices, which in this technician's work is set at a baud rate of 115200 (for serial interfaces, this corresponds to bits per second). To allow a fast and clean execution in the program, the USART interface with interrupt is used. The initialization is done like with the I/O ports first with clock and pin configuration, but now the exact interface settings (baud rate, parity, number of data bits, number of stop bits) and the interrupt are added.

```
// Clocks einschalten
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIO_AMB,   ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO,       ENABLE);
RCC_APB2PeriphClockCmd(RCC_APBPeriph_USART_AMB,   ENABLE);

// IO Konfiguration
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_2MHz;
GPIO_InitStructure.GPIO_Mode  = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Pin   = GPIO_Pin_AMB_Tx;
GPIO_Init(GPIO_AMB, &GPIO_InitStructure);
GPIO_InitStructure.GPIO_Mode  = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Pin   = GPIO_Pin_AMB_Rx | GPIO_Pin_AMB_Rts;
GPIO_Init(GPIO_AMB, &GPIO_InitStructure);

// Monitor USART Konfiguration
USART_InitStructure.USART_BaudRate          = USART_BaudRate_AMB;
USART_InitStructure.USART_WordLength         = USART_WordLength_8b;  // 8
USART_InitStructure.USART_Parity            = USART_Parity_No;       // N
USART_InitStructure.USART_StopBits          = USART_StopBits_1;      // 1
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode              = USART_Mode_Rx | USART_Mode_Tx;

// Enable the USARTy Interrupt
NVIC_InitStructure.NVIC_IRQChannel = USART_AMB_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 2;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
```

4.1.7 Analog / Digital Converter

To be able to measure analog voltage the STM32F103 has several analog measuring channels. A total of two 12 bit resolving ADCs are responsible for 16 channels, which are switched internally if required. These can be operated either in "single shot" mode (measurement once per command) or in "scan mode". In the latter, automatic measurements are performed at time intervals.

In this work, the ADC is operated in "single shot" mode, calibrated independently by the MCU during initialization to increase accuracy, and controlled by operating system. So with a small configuration immediately after the measurement a conversion of digital values (0-4095) into real voltage values (0V-3,3V) can take place. The Real Time OS already performs these measurements independently in the background and stores the current value in a variable. The user only needs to read out the finished value.

Since measurements of voltage happen often and with a lot of information, processing via interrupt would be possible here, but would not make sense since this would utilize the MCU to the greatest possible extent. To avoid this disadvantage, the STM32 has the so-called DMA ("Direct Memory Access") controller, with the help of which data can be exchanged between periphery and memory. The actual program processing is continued in parallel by the controller without any time delay of the main program.

This DMA method is available to the STM32 for interfaces like ADC, SPI, I²C, USART interface as well as several timers.

Since the ADC measurements should be as accurate as possible, the supply reference voltage of the STM32 has its own inductance on the board, in addition to the normal voltage supply, to remove possible interference. If this supply would be parallel to the normal supply of the device, without its own inductance, voltage fluctuations could result in measurement inaccuracies.

4.1.8 Sensor Software parts

For the execution of the task, the operating system still needs the actual task as a command program. With the sensor board this consists of the following points:

- **Status LED blink (optional)**
  In order to check the operability of the module, it is intended to flash an LED on the board. Optional is this point for reasons to save power. The LED should only signal the runnability of the board during commissioning until the software runs "smoothly".

- **Measurement of the „ECG" Voltage**
  The voltage, which is output by the instrumentation amplifier circuit, and which represents the course of the heartbeat, is interrogated and processed once per millisecond. This voltage is analyzed and converted once as a pulse value, and also output in the form of hexadecimal values together with the pulse frequency on the serial interface to the radio module and thus sent immediately. For the pulse evaluation, a "trigger point" is defined on the rising edge of the signal, and the number of measurements is counted until this appears again on the next rising edge. The conversion from this results in the pulse frequency.

- **Board type selection**
  When the program is started, an I/O pin is used to determine whether the board should run as a sensor or interface. On the sensor board this port pin is connected to ground with a pull-down resistor and thus signals to the program the state as sensor.

These points describe in outline the specific program of the sensor board.

4.1.9 Sensor ECG evaluation

In order to provide additional information for the optional point, the PC program, a pulse evaluation was also realized. Since it is very difficult to draw conclusions about the speed of the pulse by the pure, visual curve, an additional function for pulse display is useful. This evaluation is performed in the microcontroller when reading and processing the measured value. The following figure illustrates the function:
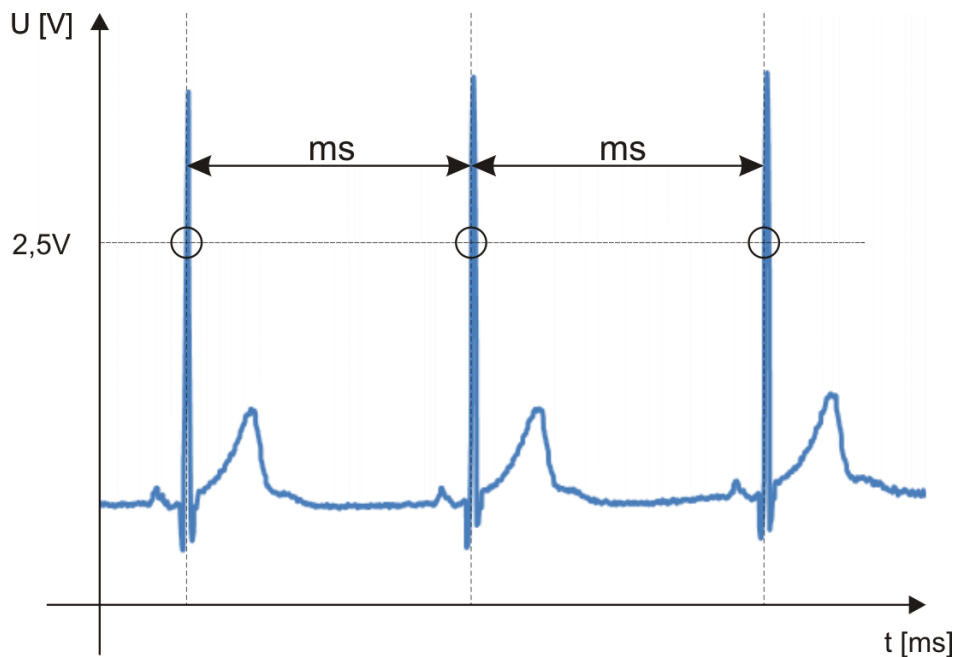
If the microcontroller detects a voltage value above 2.5V, the time until the next time this voltage is exceeded is counted in milliseconds.
If this occurs, the pulse is calculated using the following equation:

$$PulseVal = \frac{60 * 1000}{[ms\,between\,Heartbeats]}$$

Subsequently, the pulse value is transmitted together with the measured value as a hexadecimal number via USART to the PC or stored on the SD card.

4.1.10 Interface   Software parts

Das Programm des Interfaceboards besteht aus den folgenden Punkten:

- **Status LED blink (optional)**
  As on the sensor board, in order to check the runnability of the module, it is intended to flash an LED on the board. Optional is this point for power saving reasons. The LED should only signal the runnability of the board during commissioning until the software runs "round".

- **Radio / PC Interface**
  If the program detects a PC when switched on, characters received via radio module (read via USART1 interface) are sent to the PC (USART2 interface). This is done as a pure interconnection, or as a conversion from radio to USB. If no PC has been detected during power-on, the data is stored on the connected microSD card via SPI interface.
  These data can be evaluated later by the computer program.
  For saving on the SD card the free library "FatFs" is used. This is a library available for many microcontrollers, which acts as an interface between the user program and the "low level" functions for read and write actions. Several FAT standards are supported, it is platform independent and easy to use.

- **Board type selection**
  When starting the program, an I/O pin is used to determine whether the board should run as sensor or interface. On the interface board this port pin is connected with a pullup resistor to +3.3V and thus signals the state as interface to the program.

These points describe in outline the specific program of the interface board.

4.2 User software

For the use on the PC, the computer software is important. This is necessary, on the one hand, to convert the recorded values, and on the other hand, the optional point of the tproject was realized in it, to follow the measurement data "live". This computer program is a program written for Windows, which is equipped with a graphic user interface, in order to make the necessary operations as simple as possible for the user. This was created with the programming environment "Visual Studio 2008" available from Microsoft in the programming language Visual Basic.
As with Visual C++ or Visual C#, this is a language adapted by Microsoft to create graphical user interfaces for Windows. In addition, the so-called ".NET" framework from Microsoft is used in this project, which already allows many functions and hardware accesses to the user very easily via various libraries.

To be able to display the transmitted data, the freely available visualization library "ZedGraph" is used as visual data output.
"ZedGraph" was written as a class library by John Champion, Chris Champion and Ronan O Sullivan in the C# programming language. The development started in 2004 via the free source code management "Sourceforge.net" and grew in the past years to a very stable, and versatile software.
With the help of this diagram library, various types of diagrams can be displayed, such as line diagrams with single or multiple axes, so-called "pivot" tables, bar charts, pie charts and much more.

In addition to the optional live mode of the technician work, a file converter has been implemented that converts data from the microSD card to CSV tables. Correct formatting is also taken into account, so that in the end a double-click on the file is sufficient to open it in Excel and create a chart in just a few steps. The conversion progress is visualized in the window with the help of a progress bar and a percentage display.

Likewise, in live mode, the program has the option to write the data as a CSV file in the background as well. As in the pure data converter, this takes into account the correct formatting for Excel to quickly create charts from the data. This item can be selected in the first program window. For the live mode, there is also the possibility of a "debug" mode in the opening window, in which the program generates data as if they were stored by the microcontroller on SD card. This was helpful for the development of the optional data converter. Since the PC program communicates with the interface unit via a serial port, the COM port is important for the live mode. Since a PC can have several of these COM ports, the exact interface must be selected. For this purpose, a list of the currently available interfaces is retrieved in the background when the program is started and displayed as a drop-down menu, so that the user can simply select the port and start the live mode.
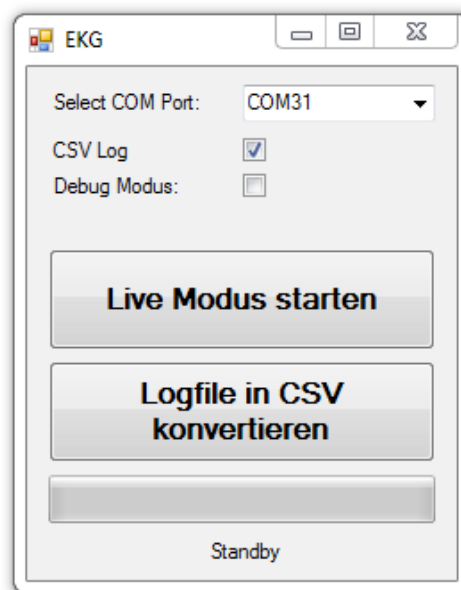
5 Circuit analysis

5.1 Test

5.1.1 Voltage and current consumption

The first step in testing the correct function of the modules is to check that the current consumption remains within a certain range and that the battery/supply voltage remains stable. Voltmeters as well as ammeters are used for this purpose. These measurements are checked once in the "virgin" state without and then with the user program.
If the current consumption is below an estimated 80 mA, the module is ready for operation.

Both modules showed the following measured values in each of the 2 states:

Interface unprogrammed: 24 mA
Interface programmed: 59 mA

Sensor unprogrammed: 24 mA
Sensor programmed: 18 mA

These values are within an absolutely acceptable range and the assemblies are fine as far as they go.
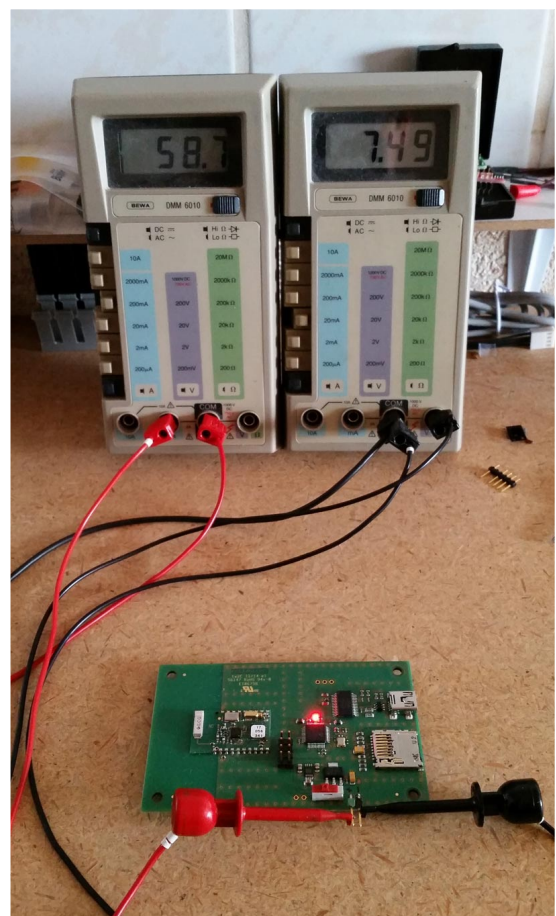
5.1.2 Interface

- **Radio Modules**

First of all, the radio modules must be parameterized. These operate in the factory state with 38400 baud and must first be set to 115200 baud for the application to enable a higher data rate. The "Amber Config Center" (ACC) tool provided by the manufacturer is used for this purpose. To parameterize the radio modules, the modules should still be without a program to avoid communication problems. The parameterization can be started with the help of two measuring lines, which are connected to the module at the pins RX and TX, as well as a ground line, which is connected to GND.
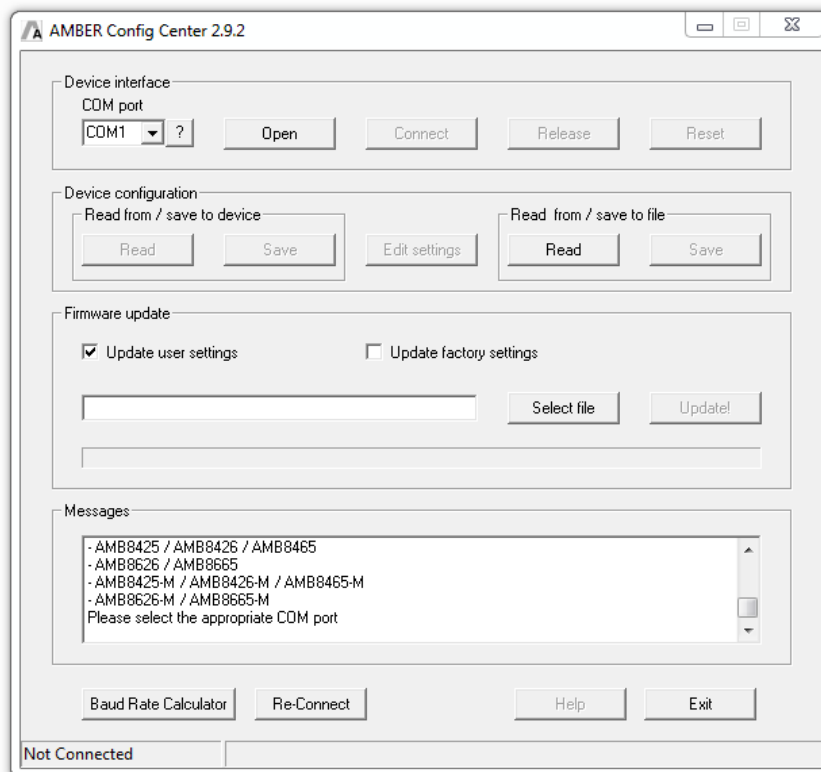
- **Serielle Schnittstelle**

The serial interfaces of the modules, which communicate with the radio modules and on the interface board with the PC, can be checked by their actual function. For this purpose the actual user program is loaded by the programming interface. For testing, the radio modules are used with a remote station in the form of a purchased USB stick from AMBER Wireless, which uses the same module as soldered on the boards. On the one hand you see the function on the sensor board in the form of the measured value output as hex values after switching on, and on the interface board in the form of the output on the actual serial interface. Optionally you can also work with the debugger in the source code and analyze it.

On the PC, if the interfaces work correctly, the module communication can be traced in the terminal program on the respective COM ports.
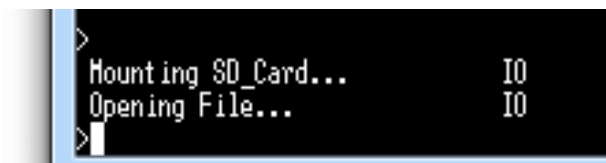
- **SD Card**

To test the SD card slot, a microSD card can be inserted into the slot when the module is in the switched-off state. If the voltage is then applied / the module is switched on, the SD card is tested immediately after the operating system has initialized. If the test is successful, the state is stored in the system and the received data is logged on the card.

If no SD card is found during initialization, this is indicated accordingly.

If an SD card is found, this is also indicated.

6 Conclusion

6.1 Results analysis

At the end of this project all required mandatory points and optional points are fulfilled. On both boards, all required interfaces are operational and they work error free.
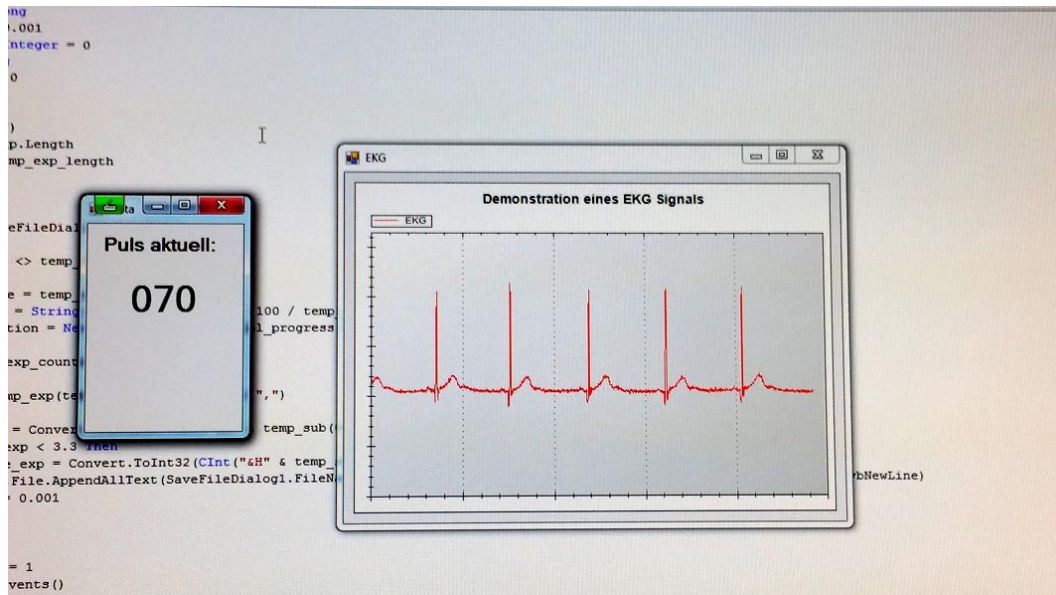

Finally, one interface and one sensor unit each are integrated into the plastic housings to ensure trouble-free handling.

The stability of the wireless interface is sufficiently stable over a few meters to send / receive correct values. If both devices are moved further away from each other, it becomes apparent that the signal slowly starts to stutter (connection interruptions) and then breaks off completely. This happens because of the integrated antenna of the AMB2520 module, an external antenna would guarantee a higher connection stability. Likewise, environmental disturbances play an important role.

Just about every section of the individual heartbeats can be traced in the voltage curve or, if necessary, viewed in retrospect using Excel.

In the end, the functioning modules were installed in the corresponding plastic housings. These housings now had to be adapted and labeled for the required BNC sockets, the SD card slot, the battery and the USB connector.
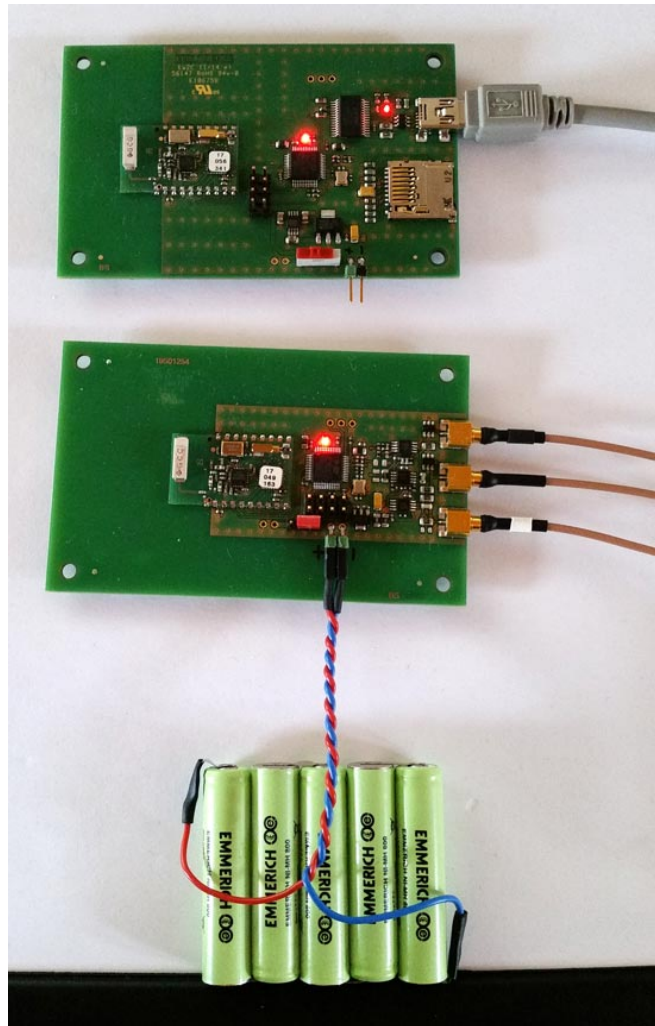
In conclusion, it remains to say that this analysis of the heart tension curve with the help of this work is not reliable enough to ensure medical relevance. In addition, the muscles produce clearly visible disturbances on the measurement line during movements, through which an accurate evaluation is then no longer possible. For this reason, the planned long-term test of the assemblies was not performed.
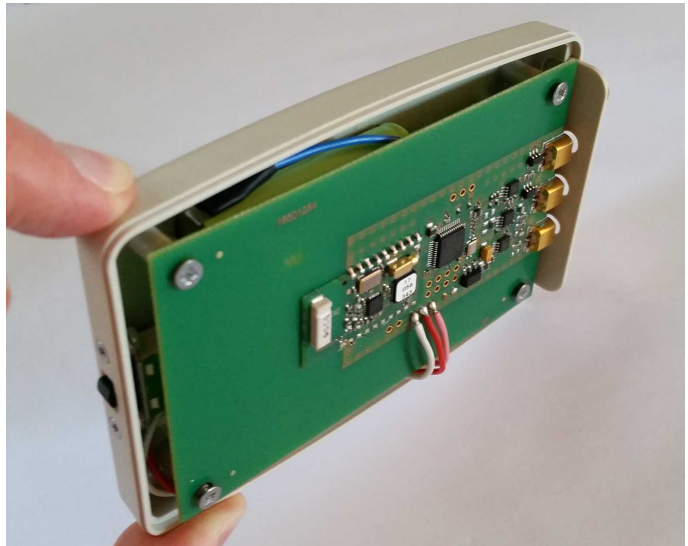
The professional ECG devices in the hospital or with physicians have several measuring points at the human body, and record also more than one measured voltage, these are thus more exact than this simple measuring method. This project is only to show the possibility that with little effort the voltage curve can be visualized for simple observation purposes.

6.2 Conclusion

In addition to the actual realization of the assembly, I was also interested in being able to better understand the individual steps in the development processes of such assemblies. This insight showed me important steps and processes that I could not really perceive as a skilled worker. Due to my previous work experience, it was not particularly difficult for me to design these assemblies. It was motivating to follow a project from start to finish, to develop solutions, to plan improvements, and to view and perfect the result.

By far the largest part of the work consisted of learning and using the CAD software, since I had only used it superficially before.

Since all parts of the project worked without any problems, the project could now be completed to my complete satisfaction.

Speed:

The transmission speed of the modules via the serial interface is 115200 baud. Considering the transmission with the set values (1 start bit, 8 data bits, no parity, 1 stop bit), this results in 10 bits per character, and for a transmission with 7 characters per millisecond this is 70000 bits per second. Since with a purely binary, serial transmission the baud rate can be regarded as bit/s specification, one sees clearly that this transmission rate is sufficient and even has reserves upward.

Energy efficiency:

Considering the battery capacity (with the specified 800mAh), the following maximum runtimes of the devices result with the used power consumption:

$$Max\ sensor\ lifetime = \frac{800\,mAh}{20\,mA} = 40\ Hours$$

$$Max\ interface\ lifetime = \frac{800\,mAh}{60\,mA} = 13,33\ Hours$$

This is within reasonable and acceptable limits for this project.

All used pictures were created by myself.