

HÖHERE TECHNISCHE BUNDESLEHRANSTALT HOLLABRUNN

Höhere Abteilung für Elektronik – Technische Informatik

Klasse/ Jahrgang: 5BHEL	Gruppe: 3 / b	Übungsleiter: Prof. Wihsböck
Übungsnummer: III / 6	Übungstitel: Medizintechnik	
Datum der Übung: 15.03.16	Teilnehmer: Fleischmann, Mötz, Nemecek, Oberhamberger	
Datum der Abgabe: 05.04.2016	Schritfführer: Dorner Fabian	Unterschrift:

	Beurteilung
Aufgabenstellung	
Dokumentation	
Messschaltungen	
Messtabellen	
Berechnungen	
Programmlistings	
Auswertung	
Diagramme	
Berechnungen	
Simulationen	
Schlussfolgerungen	
Kommentare	
Inventarliste	
Messprotokoll	
Form	
Summe	

Allgemeiner Teil

Titel der Übung: Medizintechnik

Übungsnummer: III / 6

Übungsplatz: T1

Datum der Übung: 05.04.2016

Klasse: 5BHEL

Schritfführer: Dorner Fabian

Übungsteilnehmer: Dorner Fabian

Fleichmann Raphael

Mötz Florian

Nemecek Thomas

Oberhamberger Christoph

Inhaltsverzeichnis

1. Übungsziel	3
2. Übungsvorbereitung	3
3. Übungsaufbau	3
4. Inbetriebnahme	3
5. g.Record	5
6. Messwerte aufnehmen	6
6.1. EKG-Signal	6
6.2. Pleth-Signal	7
7. Anzeige des Messsignals	7
8. Auswertung	8
8.1. EKG-Signal	8
8.2. Pleth-Signal	10
9. MATLAB	12
10. Erkenntnisse	14
Messprotokoll	15
Inventarliste	18

1. Übungsziel

Diese Übung soll einen Einblick in die Medizintechnik aus Sicht der Elektronik gewähren. Ziel der Übung war es diverse Biosignale des Körpers der Übungsteilnehmer zu messen und mit eigenen Programmen auszuwerten.

Für die Messung wurde g.MOBllab+ von Guger Technologies verwendet. Damit können verschiedene Körpersignale gemessen und kabellos per Bluetooth an einen PC übermittelt werden.

2. Übungsvorbereitung

Neben dem g.MOBllab+ System wird ein PC oder Laptop für die Übungsdurchführung benötigt. Dieses Computer-System benötigt eine VM-Ware Virtualisierungs-Software (Player oder Workstation) der Version 11 oder höher installiert.

Vom Übungsleiter wurde eine Virtuelle-Maschine mit Windows7 (64bit) bereitgestellt. Darauf waren all die für die Übung benötigten Programme (sämtliche g.MOBllab+ Anwendungen, HDFView, Code::Blocks, Matlab R2014a 64bit) installiert.

3. Übungsaufbau



Der Messaufbau um die diversen Signale des Körpers zu messen sieht etwa wie oben aus. Die Sensoren werden an die vorgesehenen Stellen am Körper angebracht und mit der zentralen Einheit des g.MOBllab+ Kits verbunden. Danach können die Messdaten via Bluetooth an einen Laptop der Übungsteilnehmer übertragen werden.

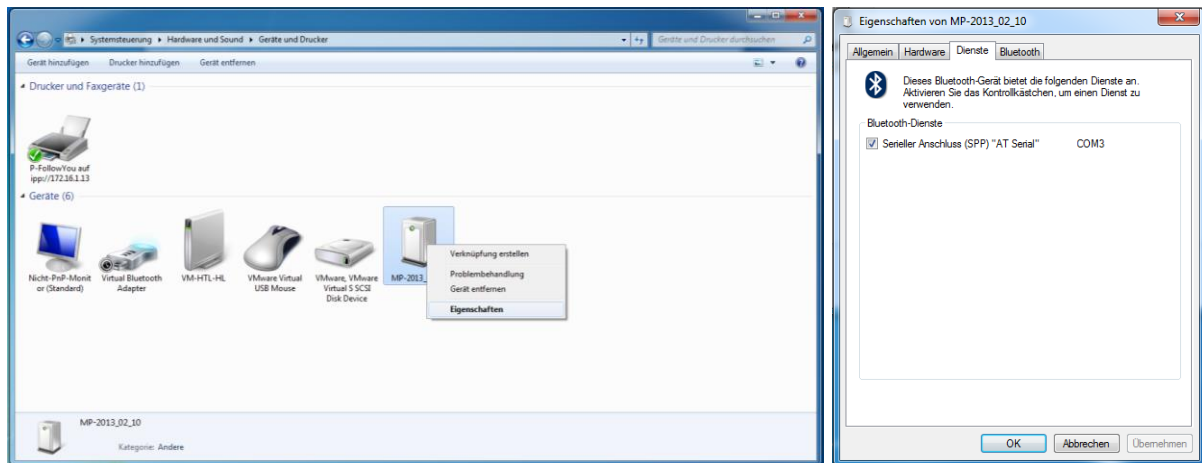
Um verschiedene Biosignale (z.B. EKG, EMG, ...) zu messen müssen in diesem Aufbau nur die Sensoren ausgetauscht werden.

4. Inbetriebnahme

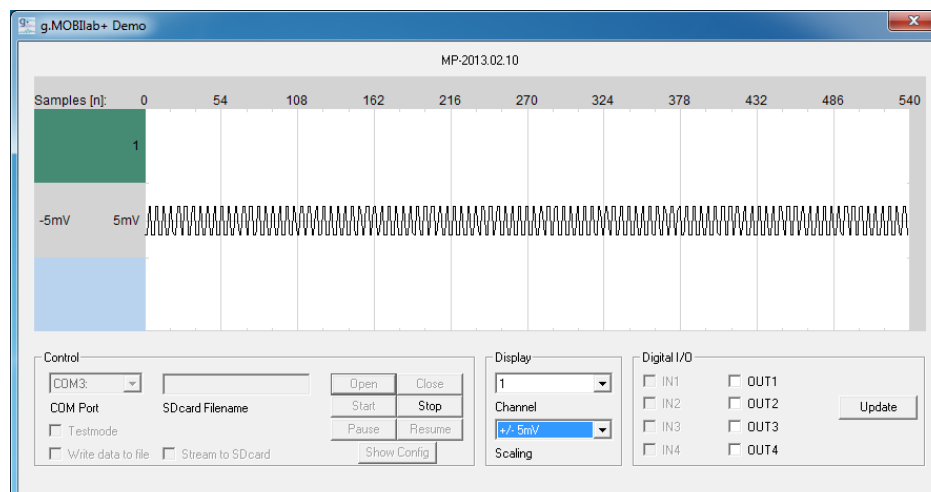
Um eine Messung vornehmen zu können müssen zuerst vier Batterien des Typ AA in die Zentraleinheit eingelegt werden und mit dem Kippschalter eingeschaltet werden.

Danach kann mit dem Bluetooth-Gerät welches den Namen der Seriennummer der Zentraleinheit hat gekoppelt werden, das Passwort dafür ist 1234. Dieser Schritt wird noch im nativen Betriebssystem durchgeführt, danach kann in die VM gewechselt werden.

Als nächstes muss die COM-Schnittstelle des verbundenen Geräts herausgefunden werden. Dazu wird in der VM die Systemsteuerung geöffnet und zu „Geräte und Drucker“ gewechselt. Mit Rechtemaustaste → Eigenschaften → Reiter „Dienste“ wird die COM-Schnittstelle sichtbar, in diesem Fall war dies COM3.



Anschließend wurde die Verbindung mit Hilfe des Programms g.MOBILab+ Demo getestet. Dazu muss auf „Open“ geklickt, der COM Port, in unserem Fall COM3 ausgewählt und zu Letzt noch die Übertragung mit „Start“ gestartet werden. In der Box „Display“ kann der entsprechende Kanal und die Skalierung für den angeschlossenen Sensor eingestellt werden.

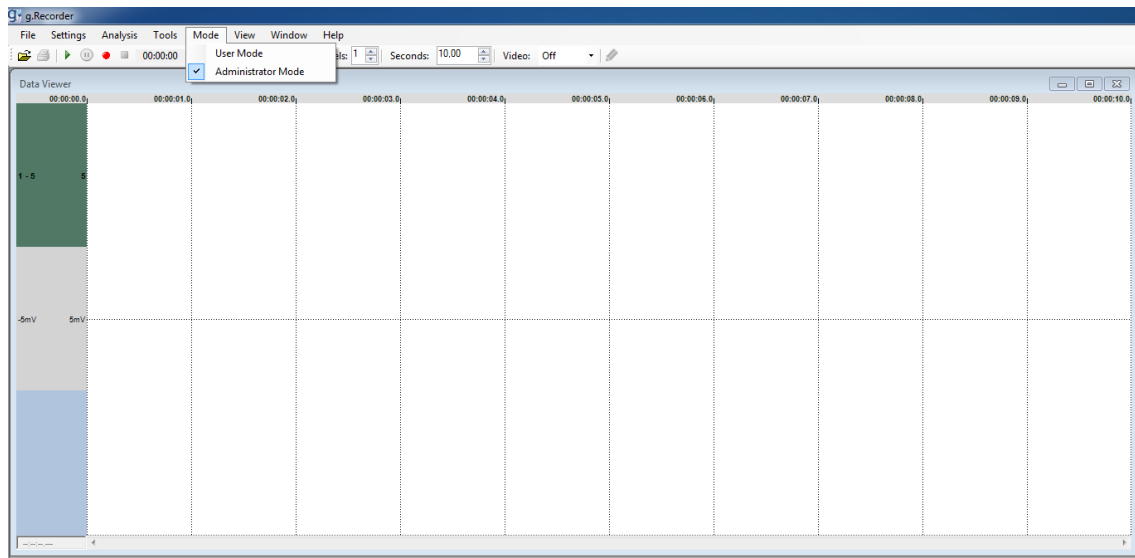


Beendet wird die Verbindung mit klicken auf „Stop“ und „Close“, danach kann das Programm beendet werden.

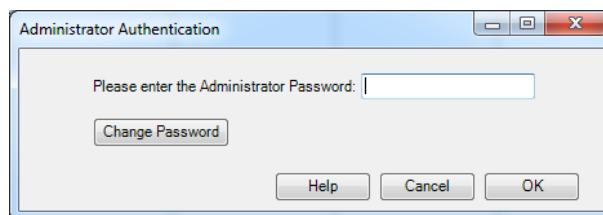
5. g.Recorder

Um den g.Recorder nutzen zu können ist ein Hardware-Dongle erforderlich, dieser wird an einen der USB-Ports des Rechners angeschlossen, danach kann das Programm ausgeführt werden.

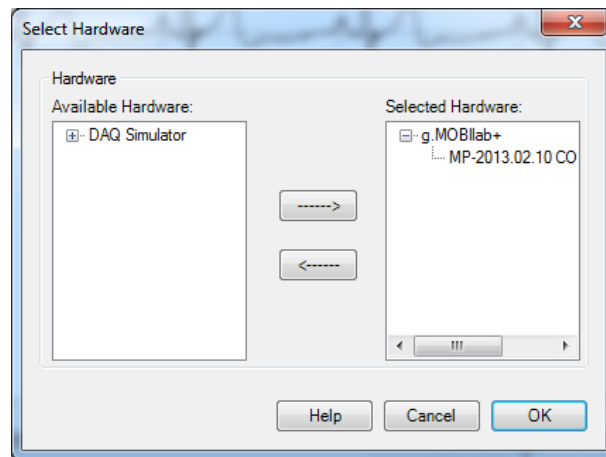
Weiters muss die Hardware in den Administrationsmodus gebracht werden. Dazu wird wie im nächsten Bild sichtbar auf „Mode“ geklickt und „Administrator Mode“ ausgewählt.



Es wird nach einem Passwort gefragt hier wird einfach mit „OK“ bestätigt, da kein Passwort vergeben wurde.



Als nächstes wird unter dem Punkt „Settings“ → „Select Hardware ...“ das Gerät ausgewählt.

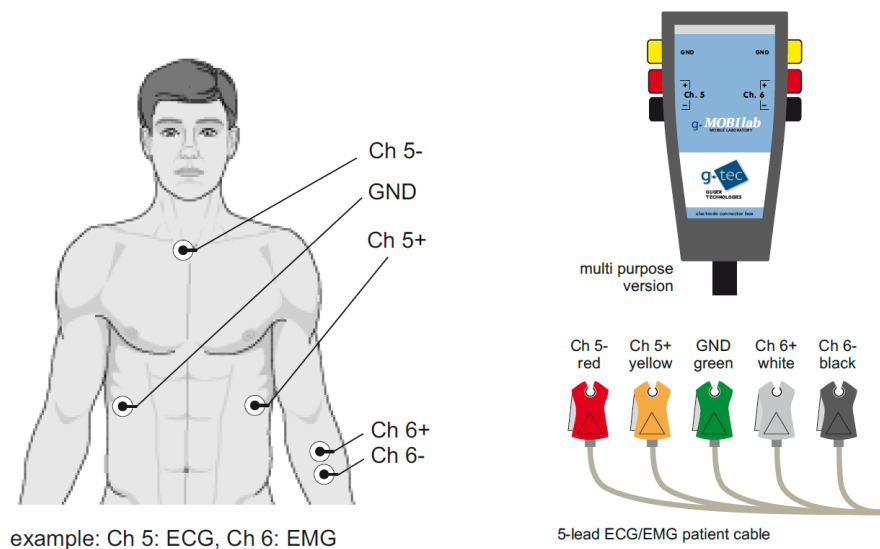


Unter dem Punkt „Settings“ → „g.MOBIlab+“ werden abschließend noch die Kanaleinstellungen vorgenommen, wie die Auswahl des richtigen Kanals oder die Verwendung eines Notch-Filters, um das Netzbrummen zu unterdrücken.

6. Messwerte aufnehmen

6.1. EKG-Signal

Um das EKG Signal zu messen wurden die Elektroden an einen der Übungsteilnehmer angebracht und mit der Zentraleinheit verbunden. Die Elektroden sind wie auf dem folgenden Bild zu sehen ist anzubringen.



Um ein EKG zu messen sind nur die Kanäle CH 5+, CH5- und GND notwendig.

Nachdem die Sensoren mit der Zentraleinheit verbunden wurden, kann durch Klicken auf den grünen „Play“-Button  das Messsignal angezeigt werden. Mit dem roten „Record“-Button  wird die Aufzeichnung des Messsignals gestartet.

Für einige der Übungsteilnehmer wurde ein EKG über mehrere Minuten aufgenommen.

6.2. Pleth-Signal

Das Pleth Signal gibt Auskunft über das Pumpverhalten des Herzens. Es wurde mit dem Fingerklipp-Sensor und der SpO2-Sensor-Box gemessen.



Der Fingerklipp wird dazu am Zeigefinger eines der Probanden angemacht.

Die SpO2-Box hat zwei Status-LEDs. Eine zeigt ob die Batterie noch genügend Spannung liefert, die andere gibt Auskunft über die Qualität des Signals. Sie blinkt sobald der Sensor ein Signal liefert und leuchtet durchgehend sobald das Signal gut ist.

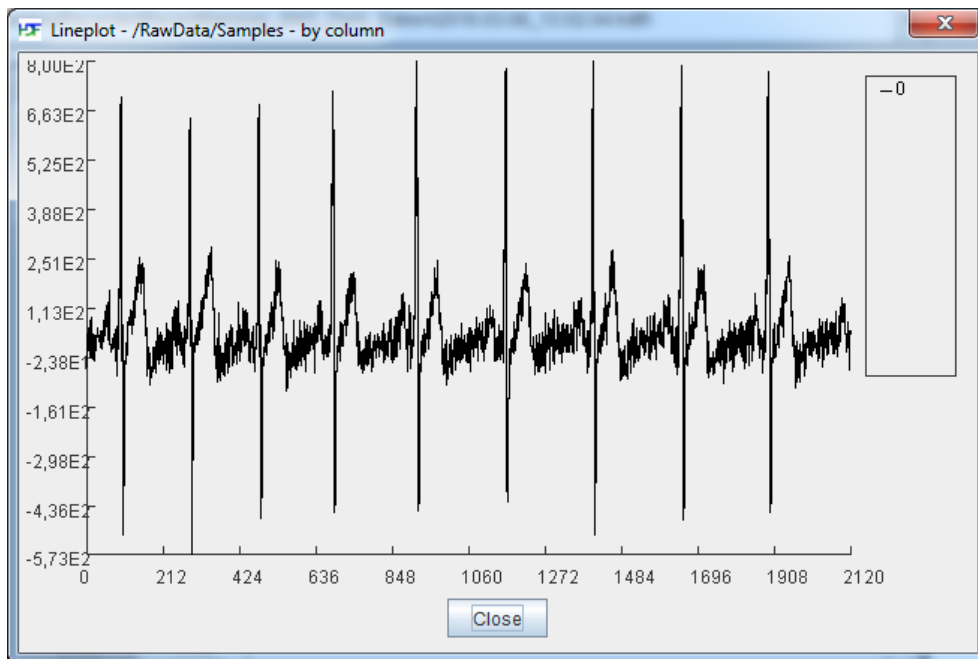
Sobald die Verbindung stimmt und das Signal gut ist kann das gemessene Biosignal wieder angezeigt oder aufgenommen werden, wiederum durch Klicken auf den grünen „Play“-Button  bzw. den roten „Record“-Button .

7. Anzeige des Messsignals

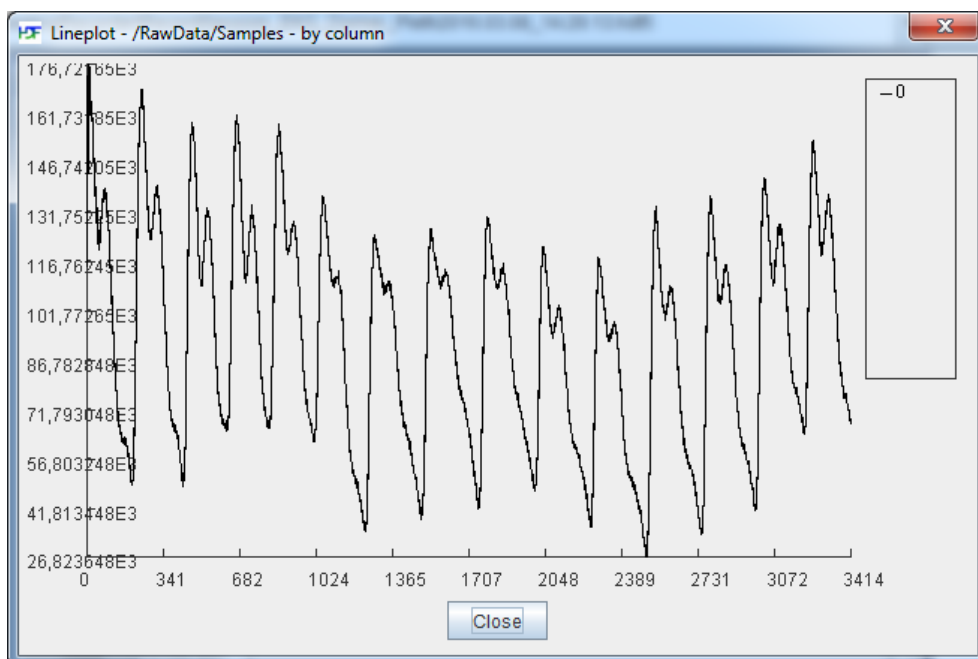
Die aufgenommenen Messsignale wurden im HDF-Dateiformat aufgenommen. Mit Hilfe des Programms HDFView können die Aufgenommenen Dateien geöffnet werden.

Wurde das gewünschte File mit dem Programm geöffnet können alle Messpunkte (Samples) angezeigt werden. Dazu wird unter dem Punkt „RawData“ doppelgeklickt auf den Punkt „Samples“. Nun können gewünschte Punkte ausgewählt und „Line Plot“ ausgewählt werden um die ausgewählten Samples in einem Graphen anzuzeigen.

Im Folgenden sind die Verläufe der EKG und Pleth Messungen von einem der Übungsteilnehmer zu sehen.



EKG-Signal



Pleth-Signal

8. Auswertung

Während der Übung wurden zwei C-Programme geschrieben welches die Anzahl der QRS-Komplexe des EKG-Signals bzw. die Ausschläge des Pleth-Signals zählt und somit die Herzfrequenz bestimmt.

8.1. EKG-Signal

Um aus dem aufgenommenen EKG-Signal die Herzfrequenz zu bestimmen wurden immer 100 Messwerte aus dem File ausgelesen und der Mittelwert davon

gebildet. Dieser Mittelwert ist notwendig um einen Schwellwert zu berechnen. Dieser darf nicht zu niedrig, um auch die P- und T-Wellen zu detektieren, aber auch nicht zu hoch gewählt werden, um keine Spitzen zu erkennen. Daher kann mit Hilfe des Mittelwerts ein dynamischer Offset generiert werden zu welchem ein konstanter Schwellwert dazu addiert wird.

Programmlisting:

```
#include <iostream>
#include <stdio.h>

#define SAMPLING_FREQUENCY 256
#define THRESHOLD_VALUE 1000
#define BUFFER_LENGTH 64

using namespace std;

float calculatePulse(char* fileName, int* count) {
    FILE* input;
    input = fopen(fileName, "r");

    float lastValues[100];
    float lastValue = 0, value = 0;
    int impulse = 0;
    int impulsePerMin = 0;
    char line[BUFFER_LENGTH];
    float pulse = 0;
    int lastValuesIndex = 0;
    float threshold = 0;
    int i = 0;
    *count = 0;

    for (i=0; i<100; i++) {
        lastValues[i] = 0;
    }

    while(fgets(line, BUFFER_LENGTH, input)) {
        sscanf(line, "%f", &value);
        (*count)++;

        if ((*count)%10 == 0) {
            threshold = 0;
            for (i=0; i < 100; i++) {
                threshold += lastValues[i];
            }
            threshold /= 100;
        }

        if(lastValue < threshold+500 && value > threshold+500) {
            impulse++;
            impulsePerMin++;
        }
        lastValue = value;
        lastValues[lastValuesIndex++ % 100] = value;
    }
}
```

```

        if((*count) % (60*SAMPLING_FREQUENCY) == 0 && *count > 0) {
            cout << "Puls in Minute " <<
            (*count)/(60*SAMPLING_FREQUENCY) << ": " << impulsePerMin << endl;
            impulsePerMin = 0;
        }
    }

    fclose(input);

    return (impulse / ( ((*count)/(float) SAMPLING_FREQUENCY) / 60));
}

void displayValues(char* fileName) {
    int count;
    float pulse;

    cout << fileName << endl;
    pulse = calculatePulse(fileName, &count);
    cout << "Durchschnittspuls: " << pulse << endl;
    cout << "Anzahl der Werte: " << count << endl << endl;
}

int main()
{
    displayValues("../Oberhammer.ekg");

    return 0;
}

```

8.2. Pleth-Signal

Um die Herzfrequenz aus dem Pleth-Signal zu ermitteln wurde die Anzahl der Spitzen gezählt und durch zwei dividiert, da es zwei Ausschläge pro Puls gibt. Um die Ausschläge zu detektieren werden immer 10 Messwerte gemittelt und mit den nächsten und übernächsten 10 verglichen. Befindet sich ein Ausschlag in der Mitte dieser 30 Werte kann dieser erkannt werden.

Programmlisting:

```

float calculatePleth(char* fileName, int* count) {
    FILE* input;
    input = fopen(fileName, "r");

    float lastValues[30];
    float avgValues[3];
    float value = 0;
    int impulse = 0;
    int impulsePerMin = 0;
    char line[BUFFER_LENGTH];
    float pulse = 0;
    int i = 0, j = 0;
    int cont = 0;
    *count = 0;

    for (i=0; i<30; i++) {
        lastValues[i] = 0;
    }
}

```

```

while(fgets(line, BUFFER_LENGTH, input)) {
    sscanf(line, "%f", &value);
    (*count)++;
    cont++;
    for (i=0; i<3; i++) {
        avgValues[i] = 0;
        for (j=0; j<10; j++) {
            avgValues[i] += lastValues[i*10 + j];
        }
        avgValues[i] /= 10;
    }

    if(avgValues[2] <= avgValues[1] && avgValues[0] <= avgValues[1]
    && cont >= 50) {
        impulse++;
        impulsePerMin++;
        cont = 0;
    }

    for (i=0; i<29; i++) {
        lastValues[i+1] = lastValues[i];
    }
    lastValues[0] = value;

    if((*count) % (60*SAMPLING_FREQUENCY) == 0 && *count > 0) {
        cout << "Puls in Minute " <<
        (*count)/(60*SAMPLING_FREQUENCY) << ": " << impulsePerMin/2 << endl;
        impulsePerMin = 0;
    }
}

fclose(input);

return (impulse / ( ((*count)/(float) SAMPLING_FREQUENCY) / 60));
}

void displayPlethValues(char* fileName) {
    int count;
    float pulse;

    cout << fileName << endl;
    pulse = calculatePleth(fileName, &count)/2;
    cout << "Durchschnittspuls: " << pulse << endl;
    cout << "Anzahl der Werte: " << count << endl << endl;
}

int main()
{
    displayPlethValues("../OberhammerPleth.ple");

    return 0;
}

```

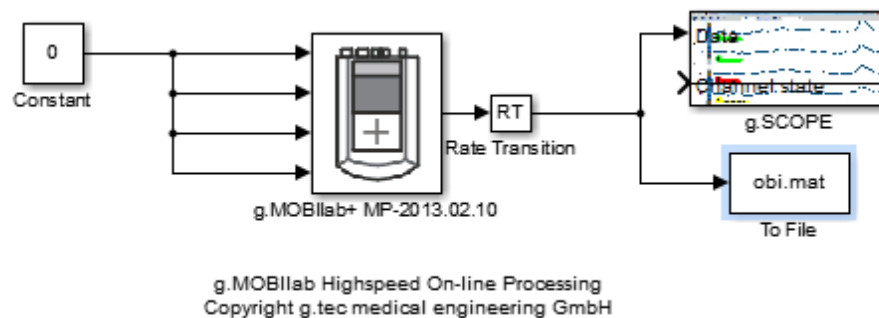
9. MATLAB

Zuletzt wurden noch Messungen mit dem g.MOBilab+ System und MATLAB durchgeführt.

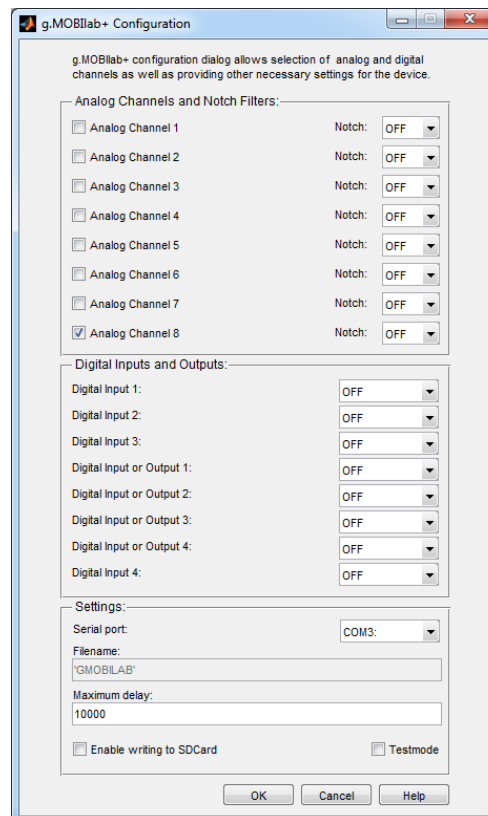
Dazu wurde MATLAB in der VM gestartet, da dort die benötigte Erweiterung bereits installiert ist. Weiters ist auch hier der Hardware-Dongle notwendig um die Erweiterung verwenden zu können.

In der VM sind bereits mehrere MATLAB-Beispielprogramme für das g.MOBilab+ vorhanden. In der Laborübung wurden die Programme gMOBilabplusedemo1 und 3 verwendet.

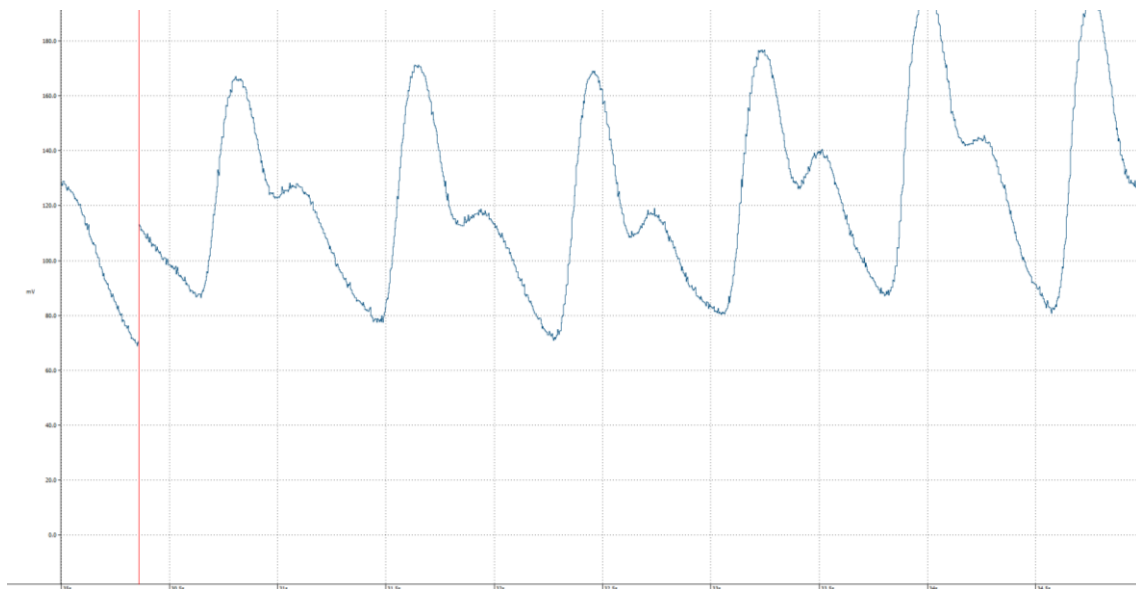
In dem nachfolgenden Aufbau wird die Messung in einem Scope ausgegeben und in dem File „obi.mat“ gespeichert.



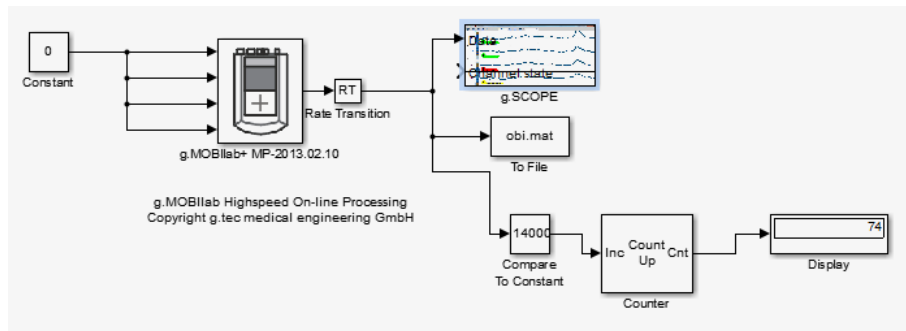
Nachdem eines dieser Programme geöffnet wurde, wurde mit einem Doppelklick auf den g.MOBilab+ Baustein um die Einstellungen zu öffnen. Hier werden wiederum die Schnittstelle und der Kanal festgelegt.



Danach kann die Simulation gestartet werden. Die Ausgabe des Graphen sieht wie folgt aus.



Anschließend wurde der Aufbau soweit verändert, dass ein Zähler um 1 erhöht wird wenn ein gewisser Schwellwert überschritten wird, in diesem Fall 14000.



10. Erkenntnisse

Die Übung gab einen Einblick in die Anwendung von Elektronik und Programmierung in der Medizintechnik, sowie die Analyse verschiedener Körpersignale.