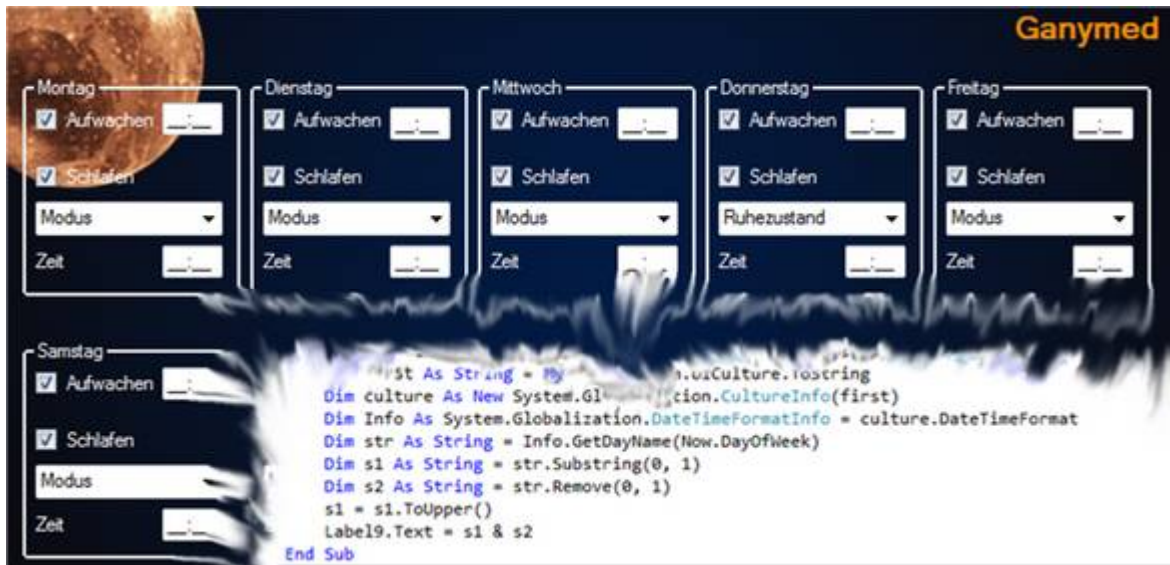




# Stromsparen durch Ruhezustand

---



Projektteam: Züger Raphael und Hofmann Tobias

Beruf: Geomatiker in Ausbildung

Schule: Berufsmaturitätsschule Zürich

Lehrperson: Dr. Remo Häuselmann

Kategorie: Energieprojekt

Thema: **Stromsparen durch Ruhezustand des Computers**

Beschreibung: Entwicklung eines Programmes, welches den Computer aufgrund eines Wochenzeitplanes in den Ruhezustand versetzt und wieder startet. Messungen und Berechnung, wie viel Strom dadurch gespart werden kann.



# Inhalt

<b>1. Einleitung .....</b>	<b>5</b>
1.1. Ausgangslage .....	5
1.2. Motivation.....	6
<b>2. Ideensuche / Projektdefinition.....</b>	<b>6</b>
2.1. Projektdefinition und -zielsetzung .....	6
2.2. Umsetzbarkeit .....	7
<b>3. Projektplanung .....</b>	<b>8</b>
3.1. Zeitplan .....	8
3.2. Die wichtigsten Meilensteine.....	9
3.3. Detaillierter Aufgabenplan.....	9
<b>4. Konkrete Umsetzung.....</b>	<b>10</b>
4.1. Die Programmteile .....	10
4.1.1. Ganymed.exe.....	11
4.1.1a Logviewer.....	12
4.1.1b Blacklist.....	13
4.1.1c Timer 3.....	13
4.1.2. Jupiter.exe .....	14
4.1.3. Wakelib.dll.....	14
4.2. Installer .....	14
4.3. Programmtest / Versionen .....	15
4.4. Produktdokumentationen .....	16
4.5. DVD-Hülle .....	16
4.5.1. Installationsanleitung.....	17
4.6. Protokolle .....	17
4.6.1. Allgemein .....	17
4.6.2. Protokollarten.....	18
4.6.2a Programmtestprotokoll [PTP] .....	18
4.6.2b Messprotokoll [MP].....	18
4.6.2c Arbeitsprotokoll [AP].....	18

<b>5. Berechnung</b> .....	<b>19</b>
5.1. Hinweise zur Berechnung.....	19
5.2. Messergebnisse .....	20
5.3. Auswertung .....	21
<b>6. Auswertung der Projektarbeit</b> .....	<b>23</b>
6.1. Rückblick.....	23
6.2. Erkenntnisse.....	24
6.3. Perspektiven.....	24
<b>7. Quellen und Abbildungen</b> .....	<b>25</b>
7.1. Quellen .....	25
7.2. Abbildungsverzeichnis.....	25
<b>Anhang</b> .....	<b>27</b>

# 1. Einleitung

## 1.1. Ausgangslage

Eines der allgegenwärtigen Themen, die uns seit Jahren immer wieder beschäftigen, ist der Klimawandel. Obwohl es immer noch Leute gibt, die den Klimawandel leugnen oder herabspielen, ist die Veränderung an unserer Welt doch offensichtlich. Die Gletscher gehen zurück, die Winter werden kürzer, wenn auch nicht weniger heftig. Ganze Arten suchen sich neue Lebensräume, immer mehr sterben aus.

Es steht fest, dass man etwas unternehmen muss, um die Klimaerwärmung, wenn auch nicht rückgängig zu machen, doch einzuschränken. Viele Massnahmen wurden bereits getroffen, wie zum Beispiel das Verbot zum Verkauf von normalen Glühbirnen. Stattdessen verwenden wir nun Sparlampen oder LEDs.

Doch damit wir wirklich etwas ändern können, müssen alle mit anpacken. Wir müssen uns klar machen, dass jeder mithelfen muss, die Ziele zu erfüllen, wenn wir unseren Nachfahren eine Welt überlassen wollen, in der es sich zu leben lohnt!

Bei der Reduktion des Stromverbrauchs kann jeder mithelfen. Zusätzlich zu den normalen Schwierigkeiten, die es zu bewältigen gibt, ist letztes Jahr noch eine weitere dazugekommen. Denn nach den Ereignissen in Fukushima, Japan, beschlossen sowohl der Bundesrat wie auch der National- und Ständerat im Jahr 2011, dass wir in der Schweiz unsere vier Atomkraftwerke bis ins Jahr 2034 vom Netz nehmen.

In der Schweiz wird aktuell etwa 40% unseres Stromes von Atomkraftwerken produziert. Wenn wir diese nun in 22 Jahren stilllegen, müssen wir dafür sorgen, dass unser Strombedarf immer noch gedeckt werden kann.

All dies kann man auf verschiedenen Varianten erreichen:

1. Durch den Bau von neuen Wind-, Wasser- und Solarkraftwerken
2. Durch das Einkaufen von Strom aus dem Ausland oder
3. durch **Energiesparmassnahmen.**

## **1.2. Motivation**

Und hier, bei den Energiesparmassnahmen, setzen wir an. Anstatt neue Kraftwerke zu bauen, was ziemlich teuer werden kann, konzentrieren wir uns darauf, stattdessen mit relativ einfachen Mitteln Strom einzusparen. Diese Energiesparmassnahmen laufen weitgehend im Hintergrund.

Wir, Raphael Züger und Tobias Hofmann, wollen mit diesem Projekt unseren Beitrag zum Thema „Stromsparen“ und damit zum Klimaschutz leisten. Mit diesem Projekt hoffen wir, dass wir dem grossen Ziel einen weiteren kleinen Schritt näherkommen.

## **2. Ideensuche / Projektdefinition**

### **2.1. Projektdefinition und -zielsetzung**

Uns war von Anfang an klar, dass unser Projekt wahrscheinlich etwas mit Computern zu tun haben wird. Es gibt Betriebe, bei denen es normal ist, dass einige Computer 24h am Tag eingeschaltet sind. Dies, damit zum Beispiel die Mitarbeiter auch von Zuhause auf ihre PCs zugreifen und weiterarbeiten können. Was für eine Arbeitsmoral! Allerdings werden auch diese arbeitsfreudigen Kollegen den PC nicht zwischen 22 Uhr und 6 Uhr früh verwenden.

Daraus entwickelte sich dann ziemlich schnell unsere Projektidee. Da Raphael ein ziemlich erfahrener und routinierter Programmierer ist, wollen wir ein Programm entwickeln, welches zu festgelegten Zeiten den jeweiligen Computer in den Ruhezustand versetzt und später wieder einschalten wird.

Nun werden sie vielleicht denken, dass es solche Programme schon zu genüge gibt. Zum Teil müssen wir Ihnen Recht geben. Doch diese Programme sind zumeist kostenpflichtig und unübersichtlich. Um sie korrekt zu bedienen, muss man sich durch viele Fenster kämpfen. Bei unserem Programm wird man die Start- und Ausschaltzeiten für die ganze Woche über eine einzige Benutzeroberfläche definieren können. Man kann somit die ganze Woche planen und sich dann der Arbeit widmen.

Bei unserem Projekt handelt es sich eindeutig um ein Energieprojekt. Wir wollen verhindern, dass Computer die ganze Nacht hindurch unnütz laufen und Strom verbrauchen. Dadurch kann übrigens auch mit einer längeren PC-Lebenszeit gerechnet werden, was wiederum sowohl dem eigenen Geldbeutel wie schlussendlich auch der Umwelt zugutekommt.

## 2.2. Umsetzbarkeit

Bei unserem Projekt sollten eigentlich keine grossen Hindernisse auftauchen. Wir müssen aber gestehen, dass wir kaum den gesamten Software-Code selbst entwickeln können, sondern zum Teil auf bekannten Code zurückgreifen und ihn für unser Projekt anpassen werden.

Das Problem hierbei ist, den Computer wieder aufwecken zu können. Nachdem unser Programm den PC in den Ruhezustand heruntergefahren hat, befindet sich ja auch unser Programm im Ruhezustand. Um den Computer wieder aufwecken zu können, muss dieser „Weck-Befehl“ auf das BIOS geladen werden. Wir sind aber zuversichtlich, dass wir dieses Problem über alternative Quellen lösen können.

Ist diese Hürde schliesslich umgangen, kann unserer nach Meinung praktisch nichts mehr schiefgehen.

### 3. Projektplanung

#### 3.1. Zeitplan

	KW 46	KW 47	KW 48	KW 49	KW 50	KW 51	KW 52	KW 01	KW 02	KW 03 bis KW 07	KW 08
<b>1. Planen</b>											
Projektwahl											
Themavertiefung											
Ideen sammeln & Entwürfe											
<b>2. Realisieren</b>											
Erstellung Software											
Software-Test, Anpassungen											
Messungen											
Auswertung											
<b>3. Auswertung</b>											
Erstellung Dokumentationen											
Abgabe Dokumentation											

Tabelle 1: Zeitplanung

KW 46: 12.-18. November 2012

KW 08: 18.-24. Februar 2013



### 3.2. Die wichtigsten Meilensteine

Was	Termin
Fertigstellung Software	23. Dezember 2012
Auswertung der Vergleichsmessungen	13. Januar 2013
Fertigstellung der Dokumentation	22. Februar 2013

Tabelle 2: Meilensteine

### 3.3. Detaillierter Aufgabenplan

Was	Wer	Bis wann
Programmierung Software	Raphael	16.12.12
Testen und Anpassen Software	Raphael & Tobias	23.12.12
Durchführung der Vergleichsmessungen	Raphael & Tobias	13.01.13
Erstellung der Berechnungen	Tobias & Raphael	13.01.13
Fertigstellung der Dokumentation	Tobias	22.02.13

Tabelle 3: Aufgabenplan

## 4. Konkrete Umsetzung

### 4.1. Die Programmteile

Hauptsächlich müssen wir für unser Projekt eine Anwendung entwickeln. Diese wird in VisualBasic geschrieben sein. Der Code von Ganymed ist im Anhang beigelegt, allerdings gibt es keine kommentierte Version.

Die Anwendung lässt sich in verschiedene Programmteile unterteilen. Unten sehen sie ein Schema, welches ihnen einen Überblick über die wichtigsten Programmteile verschafft.

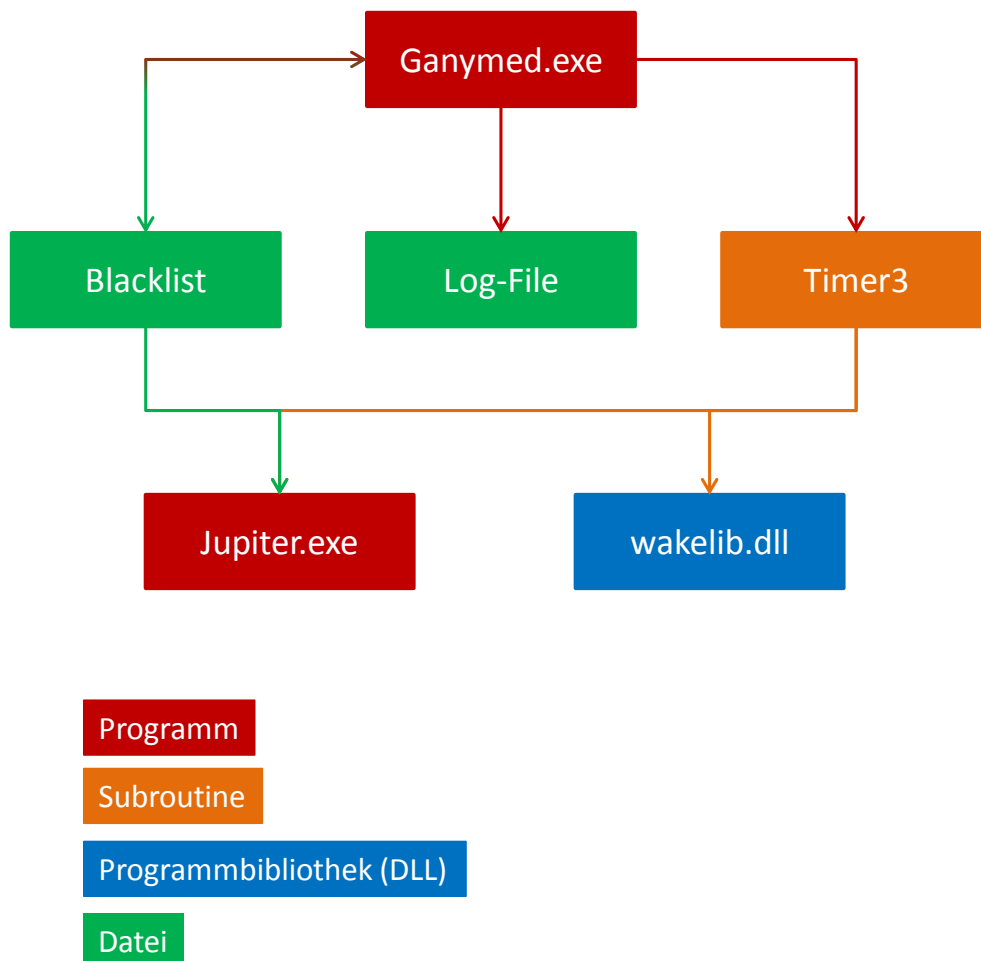


Abbildung 1: Schema Anwendung

Die im Schema dargestellten Programmteile werden im Folgenden noch näher erläutert.

### 4.1.1. Ganymed.exe

Ganymed ist der Name des Hauptprogrammes. Es enthält die meisten Subroutinen und stellt die grafische Oberfläche bereit. Mit Ganymed erhält man Zugriff auf die Blacklist und das Logfile, stellt die Uhrzeiten für die „Aufweck“- und „Schlaf“-Zeiten sowie die Modi Ruhezustand/Herunterfahren. Ganymed übergibt die Zeitangaben an seine wichtigste Subroutine, Timer 3.

Ganymed setzt sich aus insgesamt 80 Unterprozeduren (Subs) und 4 Funktionen (Functions) zusammen. Diese enthalten total etwa 1700 Zeilen Code. Weiter besteht Ganymed aus 4 Fenster (Forms): Die Hauptbenutzeroberfläche, das Info-Fenster und die Oberflächen für die Blacklist und den Logviewer. Das gesamte Projekt, also inklusive Jupiter und die DLL-Dateien, ist etwa 10 Mbyte gross. Getestet wurde es vor allem auf Windows 7-Systemen. Es sollte aber auch auf Windows 8-Systemen laufen.

Die Bezeichnung Ruhezustand meint genauer die Variante S4 des Schlafzustandes G1 nach ACPI-Standard<sup>1</sup>. Dabei wird die meiste Hardware der Hauptplatine abgeschaltet und der Betriebszustand auf der Festplatte gespeichert. S4 wird auch als *suspend to disk* oder *hibernation* bezeichnet. Der grosse Vorteil dieses Zustandes ist es, dass der Computer darin praktisch keinen Strom verbraucht, der Computer schneller wieder betriebsbereit ist, als wenn man ihn neu startet und die aktuelle Sitzung nicht unterbrochen wird.



Abbildung 2: Screenshot Ganymed

<sup>1</sup> Das Advanced Configuration and Power Interface (ACPI) ist ein offener Industriestandard für die Energieverwaltung. Weiter Informationen sind unter der Website [http://de.wikipedia.org/wiki/Advanced\\_Configuration\\_and\\_Power\\_Interface](http://de.wikipedia.org/wiki/Advanced_Configuration_and_Power_Interface) zu finden

### 4.1.1a Logviewer

Der Logviewer ist die Anzeigefläche für das Logfile und ist über die Hauptbenutzeroberfläche von Ganymed zu öffnen. Das Logfile zeichnet mit Datums- und Zeitangabe die Programmstart- bzw. Programmende-Events auf. Weiter werden die Computerabschaltungen und –aktivierungen verzeichnet. Das Log-File ist eine ANSI<sup>2</sup>-formatierte Textdatei und ist im AppData<sup>3</sup>-Ordner des Benutzers zu finden. Wird Ganymed deinstalliert, wird das Logfile im AppData-Ordner nicht gelöscht. Pro Version Ganymeds wird ein neues Logfile erstellt.

Im Verlauf des Projektes hat sich der Logviewer stark verändert.

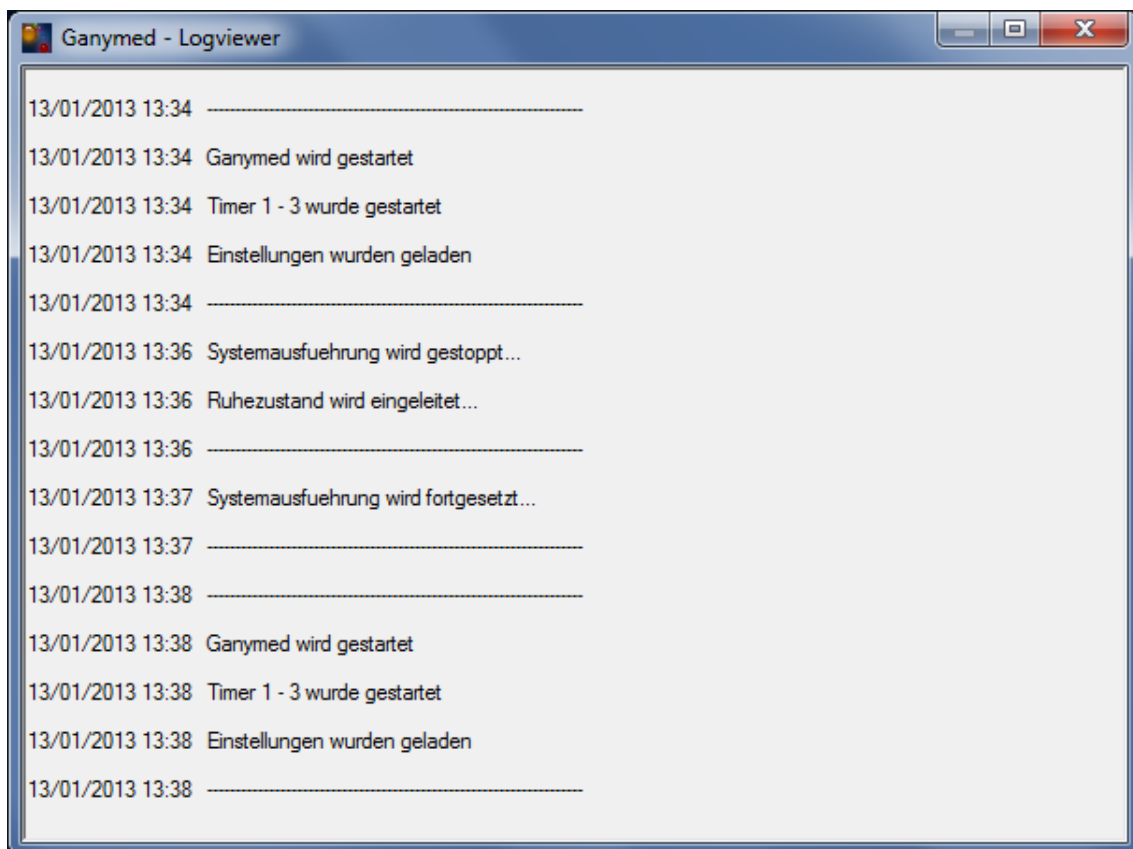


Abbildung 3: Screenshot Logviewer

<sup>2</sup> Das American National Standards Institute (ANSI) ist die US-amerikanische Stelle zur Normung industrieller Verfahrensweisen.

<sup>3</sup> Im AppData-Ordner werden zum Beispiel auch temporäre Dateien und Benutzereinstellungen gespeichert. Im normalen Windows-Betrieb ist er ausgeblendet.

### 4.1.1b Blacklist

Die Blacklist ist eine Funktion, mit der man das Herunterfahren des Computers verhindern kann, wenn noch wichtige Prozesse ausgeführt werden. Sie enthält eine Liste von Prozessnamen und kann im laufenden Betrieb angepasst werden. Weiter kann sie exportiert und in einen anderen Computer importiert werden. Dies ermöglicht die Übertragung auf verschiedene Computer, die alle die gleichen Ausnahmeregeln haben sollen.

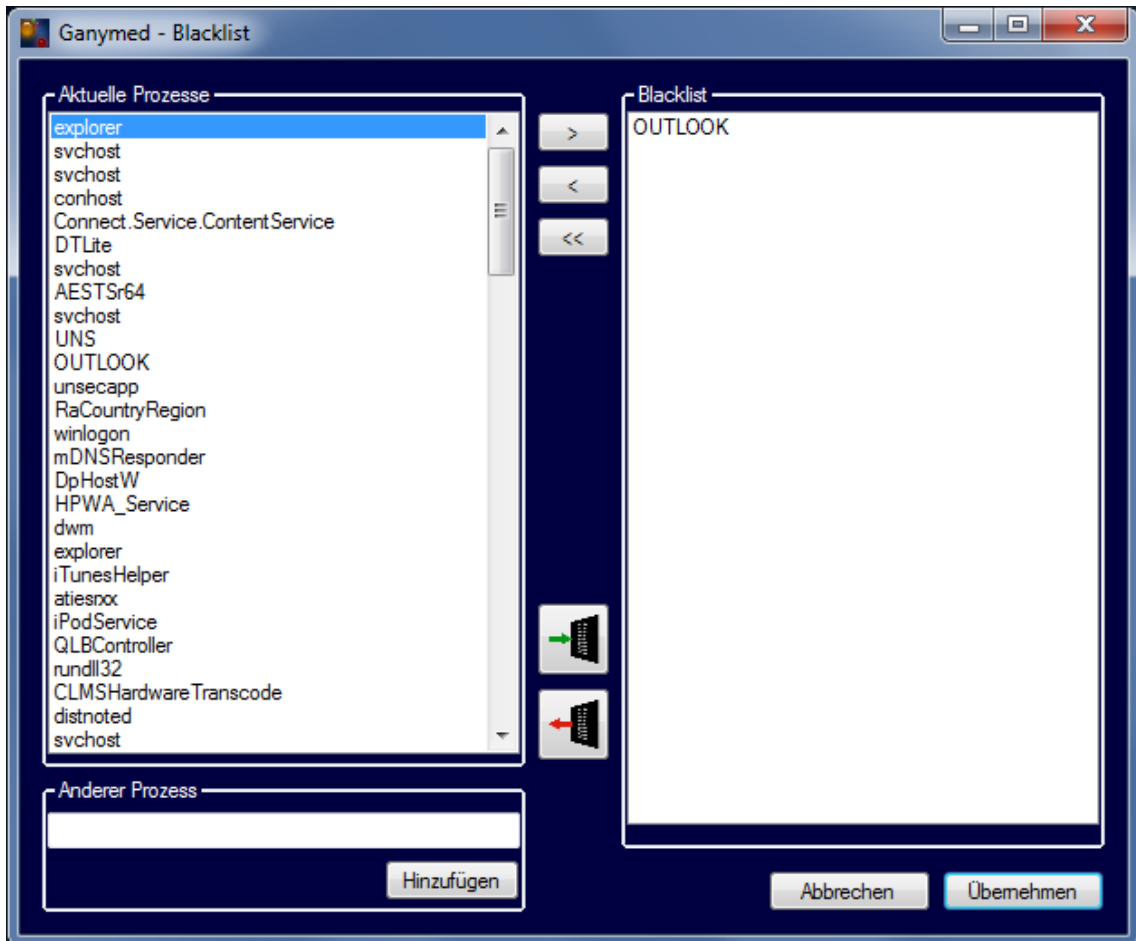


Abbildung 4: Screenshot Blacklist

### 4.1.1c Timer 3

Der Timer 3 ist ein elementarer Bestandteil von Ganymed. Seine Aufgabe ist es, die Zeiten, wann der PC in den Ruhezustand gehen und wann er wieder gestartet werden soll, zusammen mit dem Datum des Tages an das Hilfsprogramm Jupiter bzw. an die wakelib.dll zu schicken.

Um die Arbeitsweise des Timers zu verdeutlichen, befindet sich im Anhang der Programmablaufplan (PAP) nach DIN66001.

### 4.1.2. Jupiter.exe

Das Hilfsprogramm Jupiter ist relativ simpel. Es überprüft den in Ganymed gewählten Modus für das Ausschalten (Herunterfahren/Ruhezustand) und führt die betreffende Aktion aus. Ausserdem verhindert es das Ausschalten, solange ein in der Blacklist enthaltener Prozess noch ausgeführt wird. Jupiter hat im Grunde auch ein Fenster, welches aber nur im Hintergrund läuft und nicht sichtbar ist.

Das Hilfsprogramm besteht aus 3 Unterprozeduren, die total 92 Zeilen Code beinhalten, sowie 3 Funktionen.

### 4.1.3. Wakelib.dll

Die wakelib.dll wurde nicht von uns entwickelt. Der Code wurde uns vom hal2000<sup>4</sup> (aus dem Visual Basic-Forum [www.vb-paradise.de](http://www.vb-paradise.de)) zur Verfügung gestellt. Die wakelib.dll ruft weitere systemeigene .dll's auf und übergibt so die „Aufwach“-Zeit und das Datum an die Systemuhr des BIOS<sup>5</sup>.

Die Wakelib.dll besteht aus einer Unterprozedur und zwei Funktionen. Sie hat eine Grösse von etwa 440 kByte.

## 4.2. Installer

Für den Installer<sup>6</sup> wurde der Inno Setup Compiler Version 5.5.2<sup>7</sup> verwendet. Der Installer für die letzte Version Ganymeds hat eine Grösse von etwa 1.4 Mbyte und trägt den Namen *Ganymed-Setup.exe*. Eine \*.exe-Datei ist ein Selbstextrahierendes Archiv<sup>8</sup>.

Der Installer ist auf der Installations-CD zu finden. Wird er ausgeführt, muss man als erstes der Lizenzvereinbarung zustimmen. Als nächstes kann man den Installationsort ändern und weiter einen Startmenüeintrag erstellen lassen. Danach kopiert er alle Dateien an den Installationsort (Programmordner) und erstellt die erforderlichen Verknüpfungen in der Registry.

---

<sup>4</sup> Name bekannt, wird aus Personenschutzgründen hier nicht bekanntgegeben.

<sup>5</sup> Ein BIOS (Basic Input Output System) dient dazu, Systemkomponenten zu prüfen und das Betriebssystem zu laden.

<sup>6</sup> <http://de.wikipedia.org/wiki/Installer>

<sup>7</sup> <http://www.jrsoftware.org/>

<sup>8</sup> [http://de.wikipedia.org/wiki/Selbstextrahierendes\\_Archiv](http://de.wikipedia.org/wiki/Selbstextrahierendes_Archiv)

### 4.3. Programmtest / Versionen

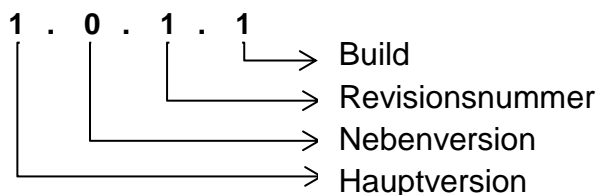
Um sicherzustellen, dass Ganymed korrekt ausgeführt wird, haben wir nach und vor allem während der Entwicklung wiederholt Programmtests durchgeführt. Für grössere Programmtests wurde jeweils ein Programmtestprotokoll erstellt. Diese sind im Anhang zu finden.

Auf diese Weise sind auch verschiedene Programmversionen zustande gekommen. Das folgende Verzeichnis soll einen Überblick über die verschiedenen Versionen Ganymeds mit den jeweiligen Neuerungen und den zugehörigen Programmtestprotokollen bieten.

Version	Neuerungen	Bemerkungen
1.0.0.0	-	Urversion
1.0.0.1	- Hauptteil noch provisorisch - Implementierung einer Blacklist	Blacklist nicht lauffähig
1.0.0.2	-Hauptteil fertiggestellt	Keine Blacklist
1.0.0.3	- Änderung der Benutzeroberfläche	-
1.0.0.4	- Funktionierende Blacklist - Oberfläche fertiggestellt	Blacklist lauffähig
1.0.0.5	- Implementierung eines Logfiles	Bereit für Alphatest
	PTP-12-001-T, PTP-12-002-R	
1.0.0.6	- Verbesserung der Oberfläche - Implementierung eines Autostarts	Anpassungen nach Alphatest
	PTP-12-003-T, PTP-12-004-T	
1.0.1.1	- Verbesserung der Bedienbarkeit - Funktionstüchtiger Autostart - Starten im Minimierten Modus	Anpassungen nach Alphatest
	PTP-12-005-R, PTP-12-006-T	

Tabelle 4: Versionsverzeichnis

Hinweise zur Versionsnummer:



#### 4.4. Produktdokumentationen

Die Produktdokumentation von Ganymed umfasst neun Seiten und wird als PDF auf der Installations-CD beigelegt. In ihr enthalten sind eine kleine Einführung, die Bedienungsanleitung und unseren Dank an hal2000 (www.vb-paradise.de)



Abbildung 5: Produktdokumentation Ganymed

#### 4.5. DVD-Hülle

Auf der Rückseite der DVD-Hülle findet man einen kleinen Informationstext über Ganymed, ein paar Screenshots und die Systemvoraussetzungen. Weiter enthält sie die Installationsanleitung.

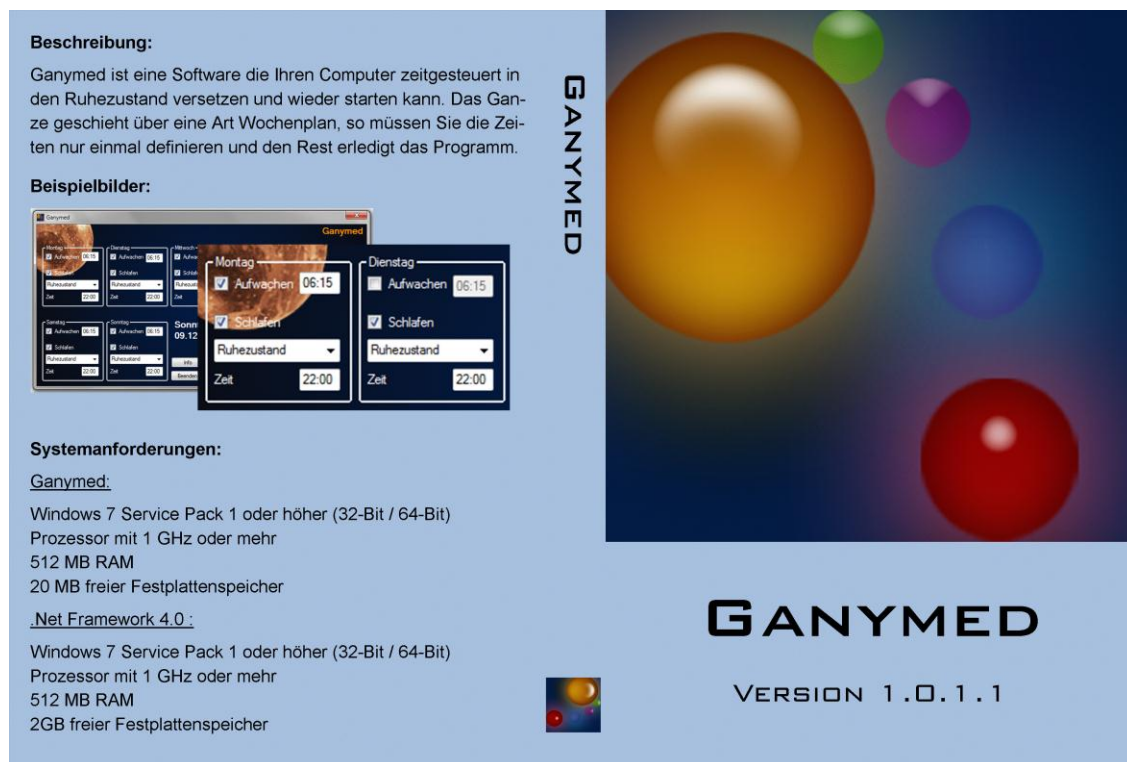


Abbildung 6: DVD-Hülle



## 4.5.1. Installationsanleitung

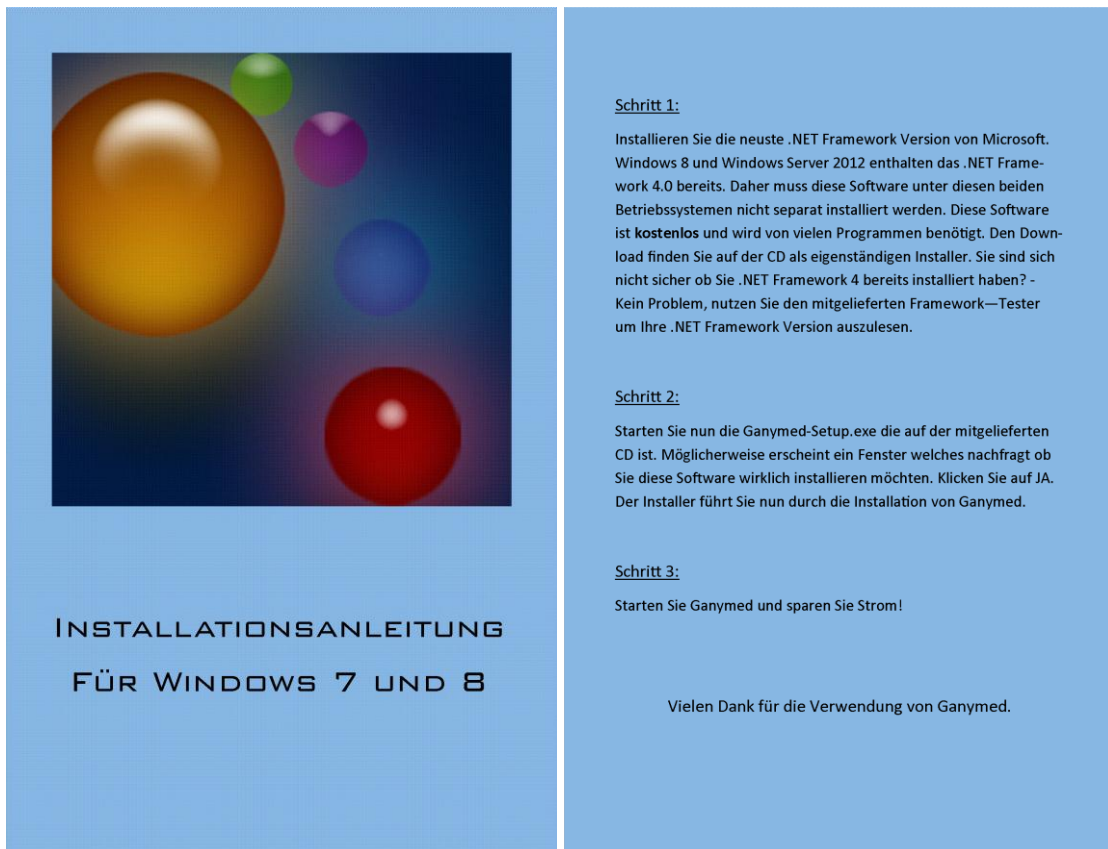


Abbildung 7: Installationsanleitung

## 4.6. Protokolle

### 4.6.1. Allgemein

Es gibt drei verschiedene Protokollarten: Programmtestprotokoll, Messprotokoll und Arbeitsprotokoll. Auf die Protokollarten wird später noch eingegangen.

Jedes Protokoll wird unterschrieben und hat eine eindeutige Protokollnummer, z.B. AP-12-001-T. Eine Protokollnummer setzt sich aus den folgenden Bestandteilen zusammen:

AP Protokollart

12 Jahr (letzte zwei Ziffern der Jahreszahl z.B. **2012**)

001 Protokollnummer (Eindeutig)

T Protokollersteller (T für Tobias Hofmann, R für Raphael Züger)

## **4.6.2. Protokollarten**

### **4.6.2a Programmtestprotokoll [PTP]**

Mithilfe des Programmtestprotokolls werden die Art und die Ergebnisse eines Programmtests, dazugehörige Bemerkungen sowie Verbesserungsvorschläge für das getestete Programm festgehalten. Die Programmtestprotokolle sind im Anhang zu finden.

Die Programmtestprotokolle wurden während und nach der Entwicklungsphase von Ganymed erstellt und sind nach dem jeweiligen Erstellungsdatum nummeriert. Insgesamt wurden sieben Protokolle erstellt. Das Letzte mit der Bezeichnung PTP-13-007-T ist erst ganz an Ende der Entwicklung von Ganymed erstellt worden. Es wurde dabei die finale Version nochmals überprüft.

### **4.6.2b Messprotokoll [MP]**

Mithilfe des Messprotokolls werden die Energiesparmessungen festgehalten. Dazu gehören unter anderem die Angabe über die Messphase, der Messzeitraum und die gemessenen Energiewerte sowie dazugehörige Bemerkungen. Auch die Messprotokolle sind im Anhang zu finden.

Insgesamt gibt es neun Messprotokolle. Die Messwerte sind am Ende der Protokolle mit einem Diagramm visualisiert. Ausserdem ist das Sparpotenzial aufgeführt.

### **4.6.2c Arbeitsprotokoll [AP]**

Im Arbeitsprotokoll werden die erledigten Arbeiten bzw. Arbeitsvorgänge festgehalten. Das Arbeitsprotokoll wird Wochenweise erstellt und an Hr. Remo Häuselmann geschickt. Es wird von allen Projektteilnehmern kontrolliert und unterschrieben. Total wurden 6 Arbeitsprotokolle erstellt.

## 5. Berechnung

### 5.1. Hinweise zur Berechnung

Um das Energiesparpotenzial unseres Programmes zu bestimmen, haben wir auf insgesamt neun Computern den Energieverbrauch gemessen. Die Testcomputer waren während der Messphase mehrmals wechselnd im Leerlauf und unter Volllast.

Für die Messungen haben wir handelsübliche Energiemessgeräte verwendet. Die gemessenen Werte wurden in total neun Messprotokollen festgehalten. Diese sind im Anhang zu finden. Bei Notebooks haben wir vor Beginn der Messung den Akku herausgenommen. Es handelt sich also auch dort um die reinen Verbrauchswerte, ohne den Ladevorgang.

Da der Stromverbrauch im Ruhezustand sehr klein ist, haben wir entschieden, ihn bei unseren Berechnungen zu vernachlässigen.



**Abbildung 8: Energiemessgerät**

Quelle: [www.ideal.de](http://www.ideal.de)

[Zugriff: 09.02.2013]

## 5.2. Messergebnisse

Computer	Betrieb 24h	Betrieb 16h *
Testcomputer 1	2.26 kWh	1.50 kWh
Testcomputer 2	0.74 kWh	0.56 kWh
Testcomputer 3	1.17 kWh	0.86 kWh
Testcomputer 4	0.62 kWh	0.42 kWh
Testcomputer 5	0.66 kWh	0.48 kWh
Testcomputer 6	0.94 kWh	0.66 kWh
Testcomputer 7	2.16 kWh	1.44 kWh
Testcomputer 8	0.88 kWh	0.65 kWh
Testcomputer 9	2.19 kWh	1.51 kWh

Tabelle 5: Messergebnisse

\* Laufzeit 16h, Ruhezustand/Heruntergefahren 8h

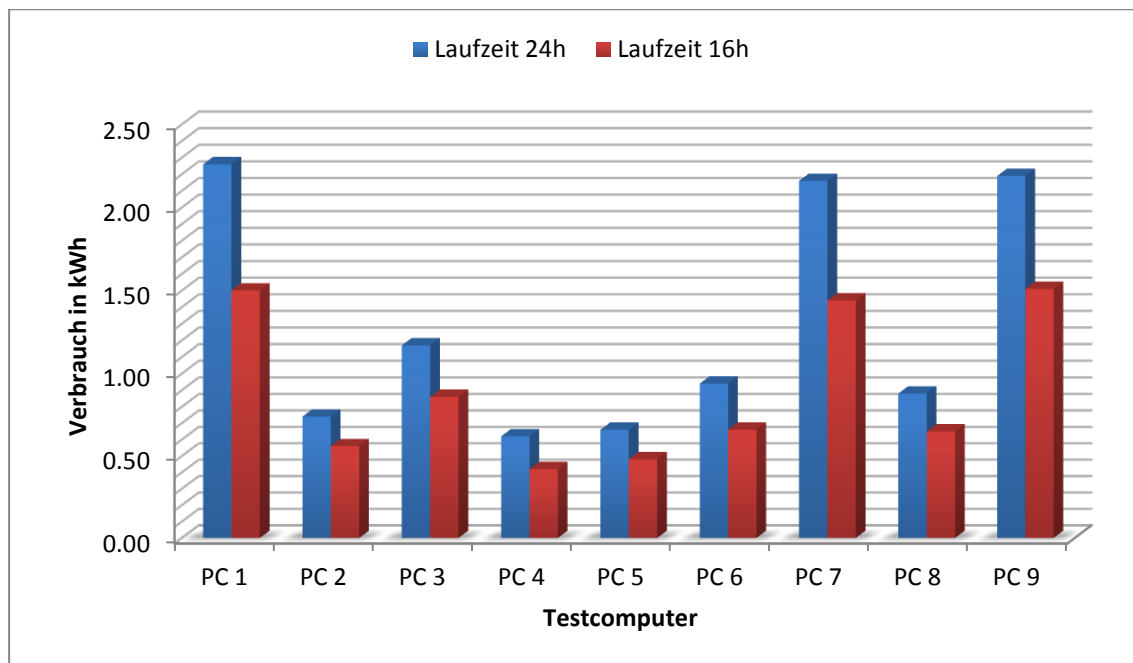


Diagramm 1: Energieverbrauch der Testcomputer pro Tag

### 5.3. Auswertung

Jetzt wollen wir den Energieverbrauch pro Jahr und pro Testcomputer bei einer Betriebszeit von 24 bzw. 16 Stunden und die gesparte Energie berechnen. Dazu multiplizieren wir den Energieverbrauch pro Tag mit der Anzahl Betriebstage.

Die Anzahl der Betriebstage setzen wir gleich mit der Anzahl Arbeitstage im Jahr 2013. Unter der Woche, also von Montag bis Freitag, haben wir dieses Jahr total 261 Tage und 8 gesetzlich anerkannte Feiertage (Kanton Zürich). Ausserdem nehmen wir 5 Wochen Ferien an. Total ergeben sich also ungefähr 228 Arbeits- bzw. Betriebstage.

#### Spaltendefinition und Formeln zur Berechnung:

Verbrauch b. 24h: Verbrauch während eines Jahres bei einer Laufzeit von 24h  
[Verbrauch 24h pro Tag] \* [228 Betriebstage]

Verbrauch b. 16h: Verbrauch während eines Jahres bei einer Laufzeit von 16h  
[Verbrauch 16h pro Tag] \* [228 Betriebstage]

Ersparnis: [Verbrauch b. 24h] - [Verbrauch b. 16h]

Computer	Verbrauch b. 24h	Verbrauch b. 16h	Ersparnis
Testcomputer 1	515.28 kWh	342.00 kWh	173.28 kWh
Testcomputer 2	168.72 kWh	127.68 kWh	41.04 kWh
Testcomputer 3	266.76 kWh	196.08 kWh	70.68 kWh
Testcomputer 4	141.36 kWh	95.76 kWh	45.60 kWh
Testcomputer 5	150.48 kWh	109.44 kWh	41.04 kWh
Testcomputer 6	214.32 kWh	150.48 kWh	63.84 kWh
Testcomputer 7	492.48 kWh	328.32 kWh	164.16 kWh
Testcomputer 8	200.64 kWh	148.20 kWh	52.44 kWh
Testcomputer 9	499.32 kWh	344.28 kWh	155.04 kWh
<b>Total</b>	<b>2649.36 kWh</b>	<b>1842.24 kWh</b>	<b>807.12 kWh</b>

Tabelle 6: Energieersparnis pro Jahr

## Stromsparen durch Ruhezustand

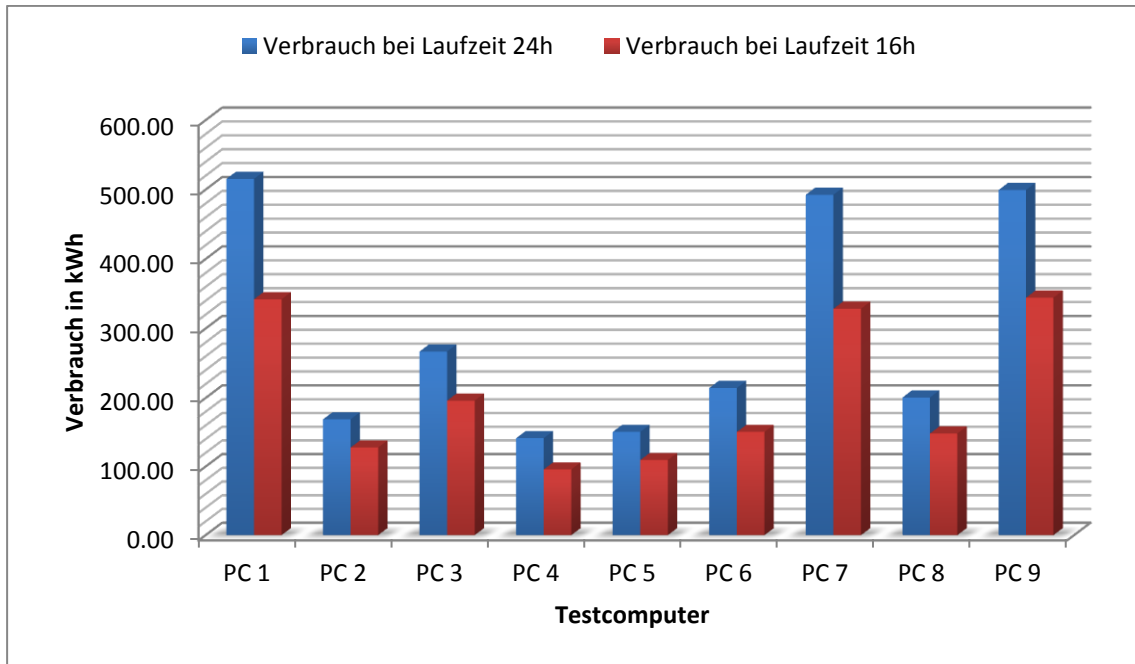


Diagramm 2: Energieverbrauch der Testcomputer pro Jahr

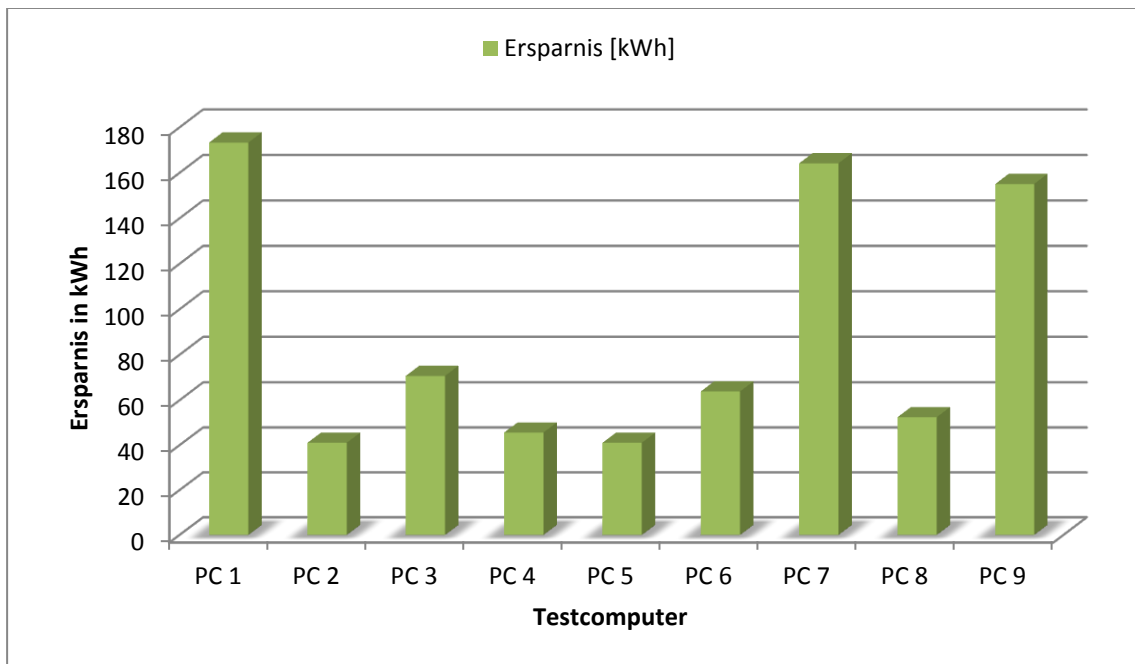


Diagramm 3: Ersparnis pro Jahr in kWh

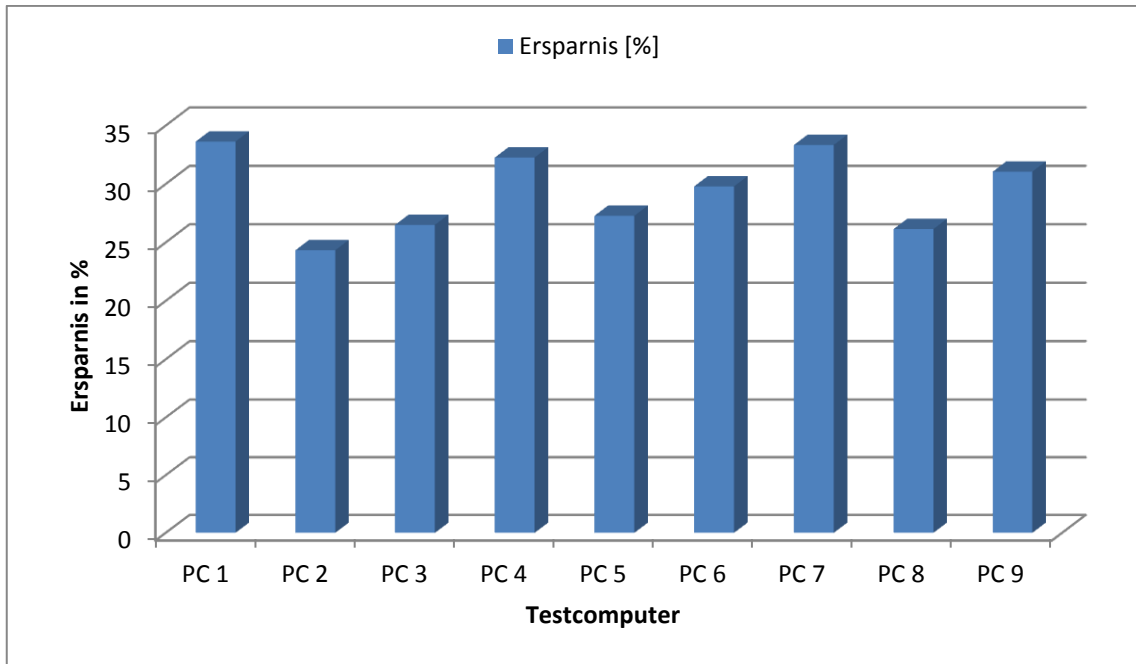


Diagramm 4: Ersparnis in Prozent

## 6. Auswertung der Projektarbeit

### 6.1. Rückblick

Uns hat das Projekt viel Spass gemacht. Es ist eine Herausforderung, ein Projekt auf diese Weise durchzuführen, und wir haben sie wohl gemeistert. Ganymed wird unsere anfänglichen Vorstellungen nicht nur gerecht, sondern hat sie auch übertroffen. Während der Entwicklung ist uns immer wieder eine neue Funktion, ein neues Tool, eine neue Optimierung eingefallen, die wir schliesslich auch umsetzen konnten. Und das Endprodukt lässt sich durchaus präsentieren!

Klar, gab es auch bei unserem Projekt Hoch- und Tiefpunkte. Die Motivation erreichte den tiefsten Punkt, als wir zu unserem Entsetzen feststellen mussten, dass das Aufstarten durch Ganymed nicht funktioniert. Zuerst suchten wir den Fehler am Programm. Es wurde uns schnell klar, dass es nur zwei Möglichkeiten gibt: Entweder hat Ganymed einen sehr groben Fehler, den wir nicht entdecken können, oder das Problem ist gar nicht Ganymed. Beflügelt durch die zweite Möglichkeit fanden wir eine Einstellung in den erweiterten Energieoptionen, laut der das Starten mittels eines Zeitplanes deaktiviert war. Nach der Aktivierung der Einstellung funktionierte Ganymed richtig, und wir konnten aufatmen.

Wie bereits im Kapitel 4 geschrieben ist, haben wir die Wakelib.dll nicht selbst geschrieben, sondern fanden sie in einem Internetforum. Eine Anfrage an den Verfasser (hal2000) stellte sicher, dass wir sie in unserem Programm verwenden dürfen. Im Allgemeinen mussten wir für Ganymed wenig auf Code-Quellen zurückgreifen.

## **6.2. Erkenntnisse**

Zuallererst haben wir mit dem Projekt bewiesen, dass jeder, der willens ist, etwas zum Thema Umweltschutz beitragen kann. Wir sind keine Profis auf dem Gebiet des Umweltschutzes, und dennoch konnten wir unseren Teil dazu leisten. Dies soll als Weckruf an alle gelten, die denken, nichts für die Umwelt tun zu können.

Weiter soll das Projekt ja auch als eine Art „Aufwärmübung“ für die Berufsmaturitätsarbeit dienen. Besonders in den Bereichen Zeitplanung und Dokumentationsverfassung war es toll, schon einmal etwas getan zu haben. Einige hatten in der Sekundarschule keine Abschlussarbeit, was man mit diesem Projekt etwas kompensieren kann.

## **6.3. Perspektiven**

Die Zukunft des Projekts und natürlich besonders Ganymeds ist ziemlich offen. Ich möchte an dieser Stelle ein kleines Beispiel machen.

Nehmen wir an, im Betrieb A laufen 20 Computer mit ähnlichen Energiewerten wie unser Testcomputer 9 den ganzen Tag. Werden diese in der Nacht neu für 8 Stunden in den Ruhezustand geschaltet, kann man pro Jahr 3100 kWh an Strom sparen. Dies entspricht nach der Internetseite <http://www.ista-swiss.ch> etwa 1829 Kg CO<sub>2</sub> pro Jahr. Um diesen Ausstoss anderweitig zu kompensieren, müsste man 166 Fichten pflanzen.



## 7. Quellen und Abbildungen

### 7.1. Quellen

hal2000. (6. Mai 2009). *PC aus Ruhezustand wecken* . Abgerufen am 24. November 2012 von <http://www.vb-paradise.de>

Redaktion Planeten.ch. (kein Datum). *Monde*. Abgerufen am 8. Dezember 2012 von <http://www.planeten.ch>

Wikimedia Foundation Inc. (6. September 2012). *Advanced Configuration and Power Interface*. Abgerufen am 9. Februar 2013 von <http://wikipedia.org>

Wikimedia Foundation Inc. (30. Dezember 2012). *ANSI*. Abgerufen am 11. Februar 2013 von <http://wikipedia.org>

Wikimedia Foundation Inc. (21. August 2012). *Atomausstieg*. Abgerufen am 5. November 2012 von <http://www.wikipedia.org>

Wikimedia Foundation Inc. (21. Januar 2013). *Wattstunde*. Abgerufen am 9. Februar 2013 von <http://www.wikipedia.org>

### 7.2. Abbildungsverzeichnis

Abbildung 1: Schema Anwendung .....	10
Abbildung 2: Screenshot Ganymed.....	11
Abbildung 3: Screenshot Logviewer .....	12
Abbildung 4: Screenshot Blacklist .....	13
Abbildung 5: Produktdokumentation Ganymed .....	16
Abbildung 6: DVD-Hülle .....	16
Abbildung 7: Installationsanleitung.....	17
Abbildung 8: Energiemessgerät .....	19
Tabelle 1: Zeitplanung.....	8
Tabelle 2: Meilensteine .....	9
Tabelle 3: Aufgabenplan .....	9
Tabelle 4: Versionsverzeichnis.....	15
Tabelle 5: Messergebnisse .....	20
Tabelle 6: Energieersparnis pro Jahr .....	21
Diagramm 1: Energieverbrauch der Testcomputer pro Tag .....	20
Diagramm 2: Energieverbrauch der Testcomputer pro Jahr .....	22
Diagramm 3: Ersparnis pro Jahr in kWh .....	22
Diagramm 4: Ersparnis in Prozent .....	23



## **Anhang**

### **1 Code Ganymed**

### **2 Programmablaufplan**

Programmablaufplan Timer3 – Teil 1

Programmablaufplan Timer3 – Teil 2

### **3 Produktdokumentation**

### **4 Programmtestprotokolle**

PTP-12-001-T

PTP-12-002-R

PTP-12-003-T

PTP-12-004-T

PTP-12-005-R

PTP-12-006-T

PTP-13-007-T

### **5 Messprotokolle**

MP-12-001-R

MP-12-002-R

MP-12-003-R

MP-12-004-R

MP-12-005-R

MP-13-006-R

MP-13-007-R

MP-13-008-T

MP-13-009-T



```
Imports System.IO
Imports wakelib
Imports Microsoft.Win32
```

```
Public Class Form1
```

```
#Region "Functions"
```

```
Public Function CreateShortcut(ByVal sLinkFile As String, _
ByVal sTargetFile As String, _
Optional ByVal sArguments As String = "", _
Optional ByVal sDescription As String = "", _
Optional ByVal sWorkingDir As String = "") As Boolean
Try
Dim oShell As New Shell32.Shell
Dim oFolder As Shell32.Folder
Dim oLink As Shell32.ShellLinkObject
Dim sPath As String = sLinkFile.Substring(0, sLinkFile.LastIndexOf("\"))
Dim sFile As String = sLinkFile.Substring(sLinkFile.LastIndexOf("\") + 1)
Dim F As Short = FreeFile()
FileOpen(F, sLinkFile, OpenMode.Output)
FileClose(F)
oFolder = oShell.Namespace(sPath)
oLink = oFolder.Items.Item(sFile).GetLink
With oLink
If sArguments.Length > 0 Then .Arguments = sArguments
If sDescription.Length > 0 Then .Description = sDescription
If sWorkingDir.Length > 0 Then .WorkingDirectory = sWorkingDir
.Path = sTargetFile
.Save()
End With
oLink = Nothing
oFolder = Nothing
oShell = Nothing
Return True
Catch ex As Exception
If System.IO.File.Exists(sLinkFile) Then Kill(sLinkFile)
logger.logfiler("-----")
logger.logfiler(ex.Message)
logger.logfiler("-----")
Return False
End Try
End Function
```

```
Dim wk As New WeekCounter.TAGWAEHLEN
```

```
#End Region
```

```
#Region "Allgemeine Subs"
```

```
Private Sub OnPowerModeChanged(ByVal sender As Object, ByVal e As
Microsoft.Win32.PowerModeChangedEventArgs)
If e.Mode = Microsoft.Win32.PowerModes.Resume Then
logger.logfiler("Systemausführung wird fortgesetzt...")
logger.logfiler("-----")

ElseIf e.Mode = Microsoft.Win32.PowerModes.Suspend Then
Select Case My.Computer.Clock.LocalTime.DayOfWeek
Case DayOfWeek.Monday
If ComboBox1.SelectedIndex = 0 Then
logger.logfiler("Systemausführung wird gestoppt...")
logger.logfiler("Ruhezustand wird eingeleitet...")
logger.logfiler("-----")
ElseIf ComboBox1.SelectedIndex = 1 Then
logger.logfiler("Systemausführung wird gestoppt...")
logger.logfiler("Herunterfahren wird eingeleitet...")
logger.logfiler("-----")
Else
End If
Case DayOfWeek.Tuesday
If ComboBox2.SelectedIndex = 0 Then
logger.logfiler("Systemausführung wird gestoppt...")
logger.logfiler("Ruhezustand wird eingeleitet...")
logger.logfiler("-----")
ElseIf ComboBox2.SelectedIndex = 1 Then
logger.logfiler("Systemausführung wird gestoppt...")
logger.logfiler("Herunterfahren wird eingeleitet...")
logger.logfiler("-----")
Else
End If
End If
```

```

Case DayOfWeek.Wednesday
    If ComboBox3.SelectedIndex = 0 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Ruhezustand wird eingeleitet...")
        logger.logfiler("-----")
    ElseIf ComboBox3.SelectedIndex = 1 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Herunterfahren wird eingeleitet...")
        logger.logfiler("-----")
    Else
    End If
Case DayOfWeek.Thursday
    If ComboBox4.SelectedIndex = 0 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Ruhezustand wird eingeleitet...")
        logger.logfiler("-----")
    ElseIf ComboBox4.SelectedIndex = 1 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Herunterfahren wird eingeleitet...")
        logger.logfiler("-----")
    Else
    End If
Case DayOfWeek.Friday
    If ComboBox5.SelectedIndex = 0 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Ruhezustand wird eingeleitet...")
        logger.logfiler("-----")
    ElseIf ComboBox5.SelectedIndex = 1 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Herunterfahren wird eingeleitet...")
        logger.logfiler("-----")
    Else
    End If
Case DayOfWeek.Saturday
    If ComboBox6.SelectedIndex = 0 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Ruhezustand wird eingeleitet...")
        logger.logfiler("-----")
    ElseIf ComboBox6.SelectedIndex = 1 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Herunterfahren wird eingeleitet...")
        logger.logfiler("-----")
    Else
    End If
Case DayOfWeek.Sunday
    If ComboBox7.SelectedIndex = 0 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Ruhezustand wird eingeleitet...")
        logger.logfiler("-----")
    ElseIf ComboBox7.SelectedIndex = 1 Then
        logger.logfiler("Systemausführung wird gestoppt...")
        logger.logfiler("Herunterfahren wird eingeleitet...")
        logger.logfiler("-----")
    Else
    End If
End Select

End If

End Sub

Public Sub New()
    My.Application.ChangeUICulture("de")
    AddHandler Microsoft.Win32.SystemEvents.PowerModeChanged, AddressOf OnPowerModeChanged
    InitializeComponent()
End Sub

Private Sub OpenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles
OpenToolStripMenuItem.Click
    Me.WindowState = FormWindowState.Normal
    Me.BringToFront()
    Me.Visible = True
End Sub

Private Sub Timer1_Tick(sender As System.Object, e As System.EventArgs) Handles Timer1.Tick
    Select Case My.Computer.Clock.LocalTime.Hour
        Case 0 To 9
            Select Case My.Computer.Clock.LocalTime.Minute
                Case 0 To 9
                    Label18.Text = "0" & My.Computer.Clock.LocalTime.Hour & ":0" &
My.Computer.Clock.LocalTime.Minute

```

```

                Case Else
                    Label18.Text = "0" & My.Computer.Clock.LocalTime.Hour & ":" &
My.Computer.Clock.LocalTime.Minute
                End Select
            Case Else
                Select Case My.Computer.Clock.LocalTime.Minute
                    Case 0 To 9
                        Label18.Text = My.Computer.Clock.LocalTime.Hour & ":0" &
My.Computer.Clock.LocalTime.Minute
                    Case Else
                        Label18.Text = My.Computer.Clock.LocalTime.Hour & ":" &
My.Computer.Clock.LocalTime.Minute
                End Select
            End Select
        End Sub

Private Sub Timer2_Tick(sender As System.Object, e As System.EventArgs) Handles Timer2.Tick
    Dim first As String = My.Application.UICulture.ToString
    Dim culture As New System.Globalization.CultureInfo(first)
    Dim Info As System.Globalization.DateTimeFormatInfo = culture.DateTimeFormat
    Dim str As String = Info.GetDayName(Now.DayOfWeek)
    Dim s1 As String = str.Substring(0, 1)
    Dim s2 As String = str.Remove(0, 1)
    s1 = s1.ToUpper()
    Label19.Text = s1 & s2
    Select Case My.Computer.Clock.LocalTime.Day
        Case 0 To 9
            Select Case My.Computer.Clock.LocalTime.Month
                Case 0 To 9
                    Label111.Text = "0" & My.Computer.Clock.LocalTime.Day.ToString & "." & "0" &
My.Computer.Clock.LocalTime.Month.ToString & "." & My.Computer.Clock.LocalTime.Year.ToString
                Case Else
                    Label111.Text = "0" & My.Computer.Clock.LocalTime.Day.ToString & "." &
My.Computer.Clock.LocalTime.Month.ToString & "." & My.Computer.Clock.LocalTime.Year.ToString
            End Select
        Case Else
            Select Case My.Computer.Clock.LocalTime.Month
                Case 0 To 9
                    Label111.Text = My.Computer.Clock.LocalTime.Day.ToString & "." & "0" &
My.Computer.Clock.LocalTime.Month.ToString & "." & My.Computer.Clock.LocalTime.Year.ToString
                Case Else
                    Label111.Text = My.Computer.Clock.LocalTime.Day.ToString & "." &
My.Computer.Clock.LocalTime.Month.ToString & "." & My.Computer.Clock.LocalTime.Year.ToString
            End Select
        End Select
    End Sub

Private Sub Form1_Closing(ByVal sender As Object, _
    ByVal e As System.ComponentModel.CancelEventArgs) Handles MyBase.Closing
    Call Button1_Click(Nothing, Nothing)
    e.Cancel = True
End Sub

Private Sub Form1_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
    Timer2_Tick(Nothing, Nothing)
    Me.Text = "Ganymed"
    logger.logfiler("-----")
    logger.logfiler("Ganymed wird gestartet")
    logger.logfiler("Timer 1 - 3 wurde gestartet")
    Call loadsettings()
    Dim KeyPath As String = "HKEY_CURRENT_USER\Software\PROGRAMM\"
    If My.Computer.Registry.GetValue(KeyPath, "FirstRun", 0) = 0 Then
        My.Computer.Registry.SetValue(KeyPath, "FirstRun", 1)
        logger.logfiler("Ganymed Version " & My.Application.Info.Version.ToString & " wird zum ersten mal
ausgefuehrt.")
        CreateShortcut(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk",
My.Application.Info.DirectoryPath & "\Ganymed.exe")
    End If
    NotifyIcon1.Icon = System.Drawing.Icon.FromHandle(My.Resources.Ganymed_ICON.GetHicon)
    My.Settings.Reload()
    If My.Settings.Autostart = True Then
        KeinAutostartToolStripMenuItem.Checked = False
        My.Settings.Autostart = True
        File.Delete(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk")
        CreateShortcut(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk",
My.Application.Info.DirectoryPath & "\Ganymed.exe")
        My.Settings.Save()
    Else
        KeinAutostartToolStripMenuItem.Checked = True
        My.Settings.Autostart = False
        File.Delete(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk")
    End If
End Sub

```

```

        My.Settings.Save()
    End If
    Call setsettings()
    Call starter()
End Sub

Sub starter()
    Me.WindowState = FormWindowState.Minimized
    NotifyIcon1.ShowBalloonTip(100)
End Sub

Sub setsettings()
    Try
        If CheckBox1.Checked = True Then
            My.Settings.c1 = 1
        Else
            My.Settings.c1 = 0
        End If

        If CheckBox2.Checked = True Then
            My.Settings.c2 = 1
        Else
            My.Settings.c2 = 0
        End If

        If CheckBox3.Checked = True Then
            My.Settings.c3 = 1
        Else
            My.Settings.c3 = 0
        End If

        If CheckBox4.Checked = True Then
            My.Settings.c4 = 1
        Else
            My.Settings.c4 = 0
        End If

        If CheckBox5.Checked = True Then
            My.Settings.c5 = 1
        Else
            My.Settings.c5 = 0
        End If

        If CheckBox6.Checked = True Then
            My.Settings.c6 = 1
        Else
            My.Settings.c6 = 0
        End If

        If CheckBox7.Checked = True Then
            My.Settings.c7 = 1
        Else
            My.Settings.c7 = 0
        End If

        If CheckBox8.Checked = True Then
            My.Settings.c8 = 1
        Else
            My.Settings.c8 = 0
        End If

        If CheckBox9.Checked = True Then
            My.Settings.c9 = 1
        Else
            My.Settings.c9 = 0
        End If

        If CheckBox10.Checked = True Then
            My.Settings.c10 = 1
        Else
            My.Settings.c10 = 0
        End If

        If CheckBox11.Checked = True Then
            My.Settings.c11 = 1
        Else
            My.Settings.c11 = 0
        End If

        If CheckBox12.Checked = True Then
            My.Settings.c12 = 1
        End If
    End Try
End Sub

```



```

Else
    My.Settings.c12 = 0
End If

If CheckBox13.Checked = True Then
    My.Settings.c13 = 1
Else
    My.Settings.c13 = 0
End If

If CheckBox14.Checked = True Then
    My.Settings.c14 = 1
Else
    My.Settings.c14 = 0
End If

My.Settings.m1 = MaskedTextBox1.Text
My.Settings.m2 = MaskedTextBox2.Text
My.Settings.m3 = MaskedTextBox3.Text
My.Settings.m4 = MaskedTextBox4.Text
My.Settings.m5 = MaskedTextBox5.Text
My.Settings.m6 = MaskedTextBox6.Text
My.Settings.m7 = MaskedTextBox7.Text
My.Settings.m8 = MaskedTextBox8.Text
My.Settings.m9 = MaskedTextBox9.Text
My.Settings.m10 = MaskedTextBox10.Text
My.Settings.m11 = MaskedTextBox11.Text
My.Settings.m12 = MaskedTextBox12.Text
My.Settings.m13 = MaskedTextBox13.Text
My.Settings.m14 = MaskedTextBox14.Text
My.Settings.cb1 = ComboBox1.SelectedIndex
My.Settings.cb2 = ComboBox2.SelectedIndex
My.Settings.cb3 = ComboBox3.SelectedIndex
My.Settings.cb4 = ComboBox4.SelectedIndex
My.Settings.cb5 = ComboBox5.SelectedIndex
My.Settings.cb6 = ComboBox6.SelectedIndex
My.Settings.cb7 = ComboBox7.SelectedIndex
Catch ex As Exception
    logger.logfiler("-----")
    logger.logfiler(ex.Message)
    logger.logfiler("-----")
End Try
My.Settings.Save()
End Sub

Sub loadsettings()
    My.Settings.Reload()
    Try
        If My.Settings.c1 = 1 Then
            CheckBox1.Checked = True
        Else
            CheckBox1.Checked = False
        End If

        If My.Settings.c2 = 1 Then
            CheckBox2.Checked = True
        Else
            CheckBox2.Checked = False
        End If

        If My.Settings.c3 = 1 Then

```

```
        CheckBox3.Checked = True
Else
    CheckBox3.Checked = False
End If

If My.Settings.c4 = 1 Then
    CheckBox4.Checked = True
Else
    CheckBox4.Checked = False
End If

If My.Settings.c5 = 1 Then
    CheckBox5.Checked = True
Else
    CheckBox5.Checked = False
End If

If My.Settings.c6 = 1 Then
    CheckBox6.Checked = True
Else
    CheckBox6.Checked = False
End If

If My.Settings.c7 = 1 Then
    CheckBox7.Checked = True
Else
    CheckBox7.Checked = False
End If

If My.Settings.c8 = 1 Then
    CheckBox8.Checked = True
Else
    CheckBox8.Checked = False
End If

If My.Settings.c9 = 1 Then
    CheckBox9.Checked = True
Else
    CheckBox9.Checked = False
End If

If My.Settings.c10 = 1 Then
    CheckBox10.Checked = True
Else
    CheckBox10.Checked = False
End If

If My.Settings.c11 = 1 Then
    CheckBox11.Checked = True
Else
    CheckBox11.Checked = False
End If

If My.Settings.c12 = 1 Then
    CheckBox12.Checked = True
Else
    CheckBox12.Checked = False
End If

If My.Settings.c13 = 1 Then
    CheckBox13.Checked = True
Else
    CheckBox13.Checked = False
End If

If My.Settings.c14 = 1 Then
    CheckBox14.Checked = True
Else
    CheckBox14.Checked = False
End If

MaskedTextBox1.Text = My.Settings.m1.ToString
MaskedTextBox2.Text = My.Settings.m2.ToString
MaskedTextBox3.Text = My.Settings.m3.ToString
MaskedTextBox4.Text = My.Settings.m4.ToString
MaskedTextBox5.Text = My.Settings.m5.ToString
```

```

MaskedTextBox6.Text = My.Settings.m6.ToString
MaskedTextBox7.Text = My.Settings.m7.ToString
MaskedTextBox8.Text = My.Settings.m8.ToString
MaskedTextBox9.Text = My.Settings.m9.ToString
MaskedTextBox10.Text = My.Settings.m10.ToString
MaskedTextBox11.Text = My.Settings.m11.ToString
MaskedTextBox12.Text = My.Settings.m12.ToString
MaskedTextBox13.Text = My.Settings.m13.ToString
MaskedTextBox14.Text = My.Settings.m14.ToString

ComboBox1.SelectedIndex = My.Settings.cb1
ComboBox2.SelectedIndex = My.Settings.cb2
ComboBox3.SelectedIndex = My.Settings.cb3
ComboBox4.SelectedIndex = My.Settings.cb4
ComboBox5.SelectedIndex = My.Settings.cb5
ComboBox6.SelectedIndex = My.Settings.cb6

ComboBox7.SelectedIndex = My.Settings.cb7
logger.logfiler("Einstellungen wurden geladen")
logger.logfiler("-----")
Catch ex As Exception
logger.logfiler("-----")
logger.logfiler(ex.Message)
logger.logfiler("-----")
End Try
End Sub

Private Sub NotifyIcon1_MouseDoubleClick(sender As System.Object, e As
System.Windows.Forms.MouseEventArgs) Handles NotifyIcon1.MouseDoubleClick
Me.BringToFront()
Me.WindowState = FormWindowState.Normal
Me.Visible = True
End Sub

#End Region

#Region "Hauptprozedur"

Private Sub Timer3_Tick(sender As System.Object, e As System.EventArgs) Handles Timer3.Tick
Dim dtime As String
If My.Computer.Clock.LocalTime.Hour.ToString.Length > 1 Then
dtime = My.Computer.Clock.LocalTime.Hour & ":" & My.Computer.Clock.LocalTime.Minute & ":" &
My.Computer.Clock.LocalTime.TimeOfDay.Seconds
Else
dtime = "0" & My.Computer.Clock.LocalTime.Hour & ":" & My.Computer.Clock.LocalTime.Minute & ":" &
My.Computer.Clock.LocalTime.TimeOfDay.Seconds
End If
If My.Computer.Clock.LocalTime.Minute.ToString.Length > 1 Then
dtime = My.Computer.Clock.LocalTime.Hour & ":" & My.Computer.Clock.LocalTime.Minute & ":" &
My.Computer.Clock.LocalTime.TimeOfDay.Seconds
Else
dtime = My.Computer.Clock.LocalTime.Hour & ":0" & My.Computer.Clock.LocalTime.Minute & ":" &
My.Computer.Clock.LocalTime.TimeOfDay.Seconds
End If
Select Case My.Computer.Clock.LocalTime.DayOfWeek
Case DayOfWeek.Monday
If CheckBox2.Checked = True Then
Select Case ComboBox1.SelectedIndex
Case 0
cmb1:
If MaskedTextBox2.Text & ":0" = dtime Then
If File.Exists(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp") = True Then
Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
Dim SW As New StreamWriter(FS)
SW.WriteLine(ComboBox1.SelectedIndex.ToString)
SW.Close()

```

```

Else
    Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
    Dim SW As New StreamWriter(FS)
    SW.WriteLine(ComboBox1.SelectedIndex.ToString)
    SW.Close()
End If

If CheckBox4.Checked = True Then
    If MaskedTextBox3.Text = ":" Then
    ElseIf MaskedTextBox3.Text = "" Then
    Else
        Dim tmond As Array = MaskedTextBox3.Text.Split(":")
        Dim mondhr As String = tmond(0)
        Dim mondmin As String = tmond(1)
        Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)
        Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide,
False, -1)

        Wake.SetWaitForWakeUpTime(tomorrow)

    End If
Else
    Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide, False, -
1)

End If

End If
Case 1
    GoTo cmb1
Case Else
    ComboBox1.SelectedIndex = 0
    GoTo cmb1
End Select
Else
End If
Case DayOfWeek.Tuesday
    If CheckBox3.Checked = True Then
        Select Case ComboBox2.SelectedIndex
            Case 0
                cmb2:
                    If MaskedTextBox4.Text & ":0" = dtime Then
                        If File.Exists(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp") = True Then
                            Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
                            Dim SW As New StreamWriter(FS)
                            SW.WriteLine(ComboBox2.SelectedIndex.ToString)
                            SW.Close()
                        Else
                            Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
                            Dim SW As New StreamWriter(FS)
                            SW.WriteLine(ComboBox2.SelectedIndex.ToString)
                            SW.Close()
                        End If
                    If CheckBox6.Checked = True Then
                        If MaskedTextBox5.Text = ":" Then
                        ElseIf MaskedTextBox5.Text = "" Then
                        Else
                            Dim tmond As Array = MaskedTextBox5.Text.Split(":")
                            Dim mondhr As String = tmond(0)
                            Dim mondmin As String = tmond(1)
                            Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)
                            Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide,
False, -1)

                            Wake.SetWaitForWakeUpTime(tomorrow)
                        End If
                    Else
                        Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide, False, -
1)

                    End If
                End If
            Case 1
                GoTo cmb2
            Case Else
                ComboBox2.SelectedIndex = 0
                GoTo cmb2
        End Select
    End If

```

```

Else
End If
Case DayOfWeek.Wednesday
If CheckBox5.Checked = True Then
Select Case ComboBox3.SelectedIndex
Case 0
cmb3:
If MaskedTextBox6.Text & ":0" = dtime Then
If File.Exists(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp") = True Then
Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
Dim SW As New StreamWriter(FS)
SW.WriteLine(ComboBox3.SelectedIndex.ToString)
SW.Close()
Else
Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
Dim SW As New StreamWriter(FS)
SW.WriteLine(ComboBox3.SelectedIndex.ToString)
SW.Close()
End If

If CheckBox8.Checked = True Then
If MaskedTextBox7.Text = ":" Then
ElseIf MaskedTextBox7.Text = "" Then
Else
Dim tmond As Array = MaskedTextBox7.Text.Split(":")
Dim mondhr As String = tmond(0)
Dim mondmin As String = tmond(1)
Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)
Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide,
False, -1)

Wake.SetWaitForWakeuptime(tomorrow)
End If
Else
Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide, False, -
1)

End If
End If
Case 1
GoTo cmb3
Case Else
ComboBox3.SelectedIndex = 0
GoTo cmb3
End Select
Else
End If
Case DayOfWeek.Thursday
If CheckBox7.Checked = True Then
Select Case ComboBox4.SelectedIndex
Case 0
cmb4:
If MaskedTextBox8.Text & ":0" = dtime Then
If File.Exists(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp") = True Then
Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
Dim SW As New StreamWriter(FS)
SW.WriteLine(ComboBox4.SelectedIndex.ToString)
SW.Close()
Else
Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
Dim SW As New StreamWriter(FS)
SW.WriteLine(ComboBox4.SelectedIndex.ToString)
SW.Close()
End If

If CheckBox10.Checked = True Then
If MaskedTextBox9.Text = ":" Then
ElseIf MaskedTextBox9.Text = "" Then
Else
Dim tmond As Array = MaskedTextBox9.Text.Split(":")
Dim mondhr As String = tmond(0)
Dim mondmin As String = tmond(1)
Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)

```



```

        SW.WriteLine(ComboBox6.SelectedIndex.ToString)
        SW.Close()
    End If

    If CheckBox14.Checked = True Then
        If MaskedTextBox13.Text = ":" Then
            ElseIf MaskedTextBox13.Text = "" Then
            Else
                Dim tmond As Array = MaskedTextBox13.Text.Split(":")
                Dim mondhr As String = tmond(0)
                Dim mondmin As String = tmond(1)
                Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)
                Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide,
False, -1)

                Wake.SetWaitForWakeUpTime(tomorrow)
            End If
        Else
            Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide, False, -
1)

        End If
    End If
    Case 1
        GoTo cmb6
    Case Else
        ComboBox6.SelectedIndex = 0
        GoTo cmb6
    End Select
Else
End If
Case DayOfWeek.Sunday
    If CheckBox13.Checked = True Then
        Select Case ComboBox7.SelectedIndex
            Case 0
cmb7:
                If MaskedTextBox14.Text & ":0" = dtime Then
                    If File.Exists(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp") = True Then
                        Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
                        Dim SW As New StreamWriter(FS)
                        SW.WriteLine(ComboBox7.SelectedIndex.ToString)
                        SW.Close()
                    Else
                        Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.Temp &
"\Settime.slp", FileMode.OpenOrCreate, FileAccess.ReadWrite, FileShare.Read)
                        Dim SW As New StreamWriter(FS)
                        SW.WriteLine(ComboBox7.SelectedIndex.ToString)
                        SW.Close()
                    End If
                End If
            If CheckBox1.Checked = True Then
                If MaskedTextBox1.Text = ":" Then
                    ElseIf MaskedTextBox1.Text = "" Then
                    Else
                        Dim tmond As Array = MaskedTextBox1.Text.Split(":")
                        Dim mondhr As String = tmond(0)
                        Dim mondmin As String = tmond(1)
                        Dim tomorrow As New DateTime(My.Computer.Clock.LocalTime.Year,
My.Computer.Clock.LocalTime.Month, My.Computer.Clock.LocalTime.Day + 1, Val(mondhr) - 1, mondmin, 0)
                        Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide,
False, -1)

                        Wake.SetWaitForWakeUpTime(tomorrow)
                    End If
                Else
                    Shell(Application.StartupPath & "\Jupiter.exe", AppWinStyle.Hide, False, -
1)

                End If
            End If
        Case 1
            GoTo cmb7
        Case Else
            ComboBox7.SelectedIndex = 0
            GoTo cmb7
        End Select
    Else
    End If
End Select
End Sub
#End Region

```

```
#Region "Checkboxes"
```

```
Private Sub CheckBox1_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox1.CheckedChanged  
If CheckBox13.Checked = True Then  
If CheckBox1.Checked = True Then  
MaskedTextBox1.Enabled = True  
Else  
MaskedTextBox1.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox4_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox4.CheckedChanged  
If CheckBox2.Checked = True Then  
If CheckBox4.Checked = True Then  
MaskedTextBox3.Enabled = True  
Else  
MaskedTextBox3.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox6_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox6.CheckedChanged  
If CheckBox3.Checked = True Then  
If CheckBox6.Checked = True Then  
MaskedTextBox5.Enabled = True  
Else  
MaskedTextBox5.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox8_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox8.CheckedChanged  
If CheckBox5.Checked = True Then  
If CheckBox8.Checked = True Then  
MaskedTextBox7.Enabled = True  
Else  
MaskedTextBox7.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox10_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox10.CheckedChanged  
If CheckBox7.Checked = True Then  
If CheckBox10.Checked = True Then  
MaskedTextBox9.Enabled = True  
Else  
MaskedTextBox9.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox12_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox12.CheckedChanged  
If CheckBox9.Checked = True Then  
If CheckBox12.Checked = True Then  
MaskedTextBox11.Enabled = True  
Else  
MaskedTextBox11.Enabled = False  
End If  
End If  
End Sub
```

```
Private Sub CheckBox14_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles  
CheckBox14.CheckedChanged  
If CheckBox11.Checked = True Then  
If CheckBox14.Checked = True Then  
MaskedTextBox13.Enabled = True  
Else  
MaskedTextBox13.Enabled = False  
End If  
End If  
End Sub
```



```

Private Sub CheckBox2_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox2.CheckedChanged
    If CheckBox2.Checked = True Then
        CheckBox4.Enabled = True
        ComboBox1.Enabled = True
        MaskedTextBox2.Enabled = True
    Else
        CheckBox4.Checked = False
        CheckBox4.Enabled = False
        MaskedTextBox3.Enabled = False
        ComboBox1.Enabled = False
        MaskedTextBox2.Enabled = False
    End If
End Sub

Private Sub CheckBox3_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox3.CheckedChanged
    If CheckBox3.Checked = True Then
        CheckBox6.Enabled = True
        ComboBox2.Enabled = True
        MaskedTextBox4.Enabled = True
    Else
        CheckBox6.Checked = False
        CheckBox6.Enabled = False
        MaskedTextBox5.Enabled = False
        ComboBox2.Enabled = False
        MaskedTextBox4.Enabled = False
    End If
End Sub

Private Sub CheckBox5_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox5.CheckedChanged
    If CheckBox5.Checked = True Then
        CheckBox8.Enabled = True
        ComboBox3.Enabled = True
        MaskedTextBox6.Enabled = True
    Else
        CheckBox8.Checked = False
        CheckBox8.Enabled = False
        MaskedTextBox7.Enabled = False
        ComboBox3.Enabled = False
        MaskedTextBox6.Enabled = False
    End If
End Sub

Private Sub CheckBox7_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox7.CheckedChanged
    If CheckBox7.Checked = True Then
        CheckBox10.Enabled = True
        ComboBox4.Enabled = True
        MaskedTextBox8.Enabled = True
    Else
        CheckBox10.Checked = False
        CheckBox10.Enabled = False
        MaskedTextBox9.Enabled = False
        ComboBox4.Enabled = False
        MaskedTextBox8.Enabled = False
    End If
End Sub

Private Sub CheckBox9_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox9.CheckedChanged
    If CheckBox9.Checked = True Then
        CheckBox12.Enabled = True
        ComboBox5.Enabled = True
        MaskedTextBox10.Enabled = True
    Else
        CheckBox12.Checked = False
        CheckBox12.Enabled = False
        MaskedTextBox11.Enabled = False
        ComboBox5.Enabled = False
        MaskedTextBox10.Enabled = False
    End If
End Sub

Private Sub CheckBox11_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox11.CheckedChanged
    If CheckBox11.Checked = True Then
        CheckBox14.Enabled = True
        ComboBox6.Enabled = True
        MaskedTextBox12.Enabled = True
    End If
End Sub

```

```

Else
    CheckBox14.Checked = False
    CheckBox14.Enabled = False
    MaskedTextBox13.Enabled = False
    ComboBox6.Enabled = False
    MaskedTextBox12.Enabled = False
End If
End Sub

Private Sub CheckBox13_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles
CheckBox13.CheckedChanged
    If CheckBox13.Checked = True Then
        CheckBox1.Enabled = True
        ComboBox7.Enabled = True
        MaskedTextBox14.Enabled = True
    Else
        CheckBox1.Checked = False
        CheckBox1.Enabled = False
        MaskedTextBox1.Enabled = False
        ComboBox7.Enabled = False
        MaskedTextBox14.Enabled = False
    End If
End Sub

#End Region

#Region "Buttons"

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
    Try
        Call setsettings()
        Me.Visible = False
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
    Dim result = MsgBox("Möchten Sie Ganymed wirklich beenden?", MsgBoxStyle.YesNo, "Ganymed")
    If result = MsgBoxResult.Yes Then
        logger.logfiler("-----")
        logger.logfiler("Ganymed wird für das Beenden vorbereitet")
        logger.logfiler("Einstellungen werden gespeichert")
        Call setsettings()
        logger.logfiler("Prozess wird gestoppt")
        logger.logfiler("Ganymed wurde beendet...")
        logger.logfiler("-----")
    End If
    Exit Sub
End Sub

Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
    Try
        Blacklist.Show()
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click
    If File.Exists(My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData &
"\Ganymedlog.log") = False Then
        Dim FS As New FileStream(My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData &
"\Ganymedlog.log", FileMode.Append, FileAccess.Write, FileShare.Read)
        Dim SW As New StreamWriter(FS)
        SW.WriteLine("Ganymed: " & Application.ProductVersion.ToString)
        SW.Close()
    End If
    Logviewer.Show()
End Sub

Private Sub FürAlleÜbernehmenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs)
Handles FürAlleÜbernehmenToolStripMenuItem.Click
    Dim stext As String = ""
    If MaskedTextBox1.Focused = True Then

```

```

    stext = MaskedTextBox1.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox3.Focused = True Then
    stext = MaskedTextBox3.Text
    MaskedTextBox1.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox5.Focused = True Then
    stext = MaskedTextBox5.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox1.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox7.Focused = True Then
    stext = MaskedTextBox7.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox1.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox9.Focused = True Then
    stext = MaskedTextBox9.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox1.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox11.Focused = True Then
    stext = MaskedTextBox11.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox1.Text = stext
    MaskedTextBox13.Text = stext
End If
If MaskedTextBox13.Focused = True Then
    stext = MaskedTextBox13.Text
    MaskedTextBox3.Text = stext
    MaskedTextBox5.Text = stext
    MaskedTextBox7.Text = stext
    MaskedTextBox9.Text = stext
    MaskedTextBox11.Text = stext
    MaskedTextBox1.Text = stext
End If

If MaskedTextBox2.Focused = True Then
    stext = MaskedTextBox2.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox14.Text = stext
End If
If MaskedTextBox4.Focused = True Then
    stext = MaskedTextBox4.Text
    MaskedTextBox2.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox14.Text = stext
End If

```

```

If MaskedTextBox6.Focused = True Then
    stext = MaskedTextBox6.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox2.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox14.Text = stext
End If
If MaskedTextBox8.Focused = True Then
    stext = MaskedTextBox8.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox2.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox14.Text = stext
End If
If MaskedTextBox10.Focused = True Then
    stext = MaskedTextBox10.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox2.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox14.Text = stext
End If
If MaskedTextBox12.Focused = True Then
    stext = MaskedTextBox12.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox2.Text = stext
    MaskedTextBox14.Text = stext
End If
If MaskedTextBox14.Focused = True Then
    stext = MaskedTextBox14.Text
    MaskedTextBox4.Text = stext
    MaskedTextBox6.Text = stext
    MaskedTextBox8.Text = stext
    MaskedTextBox10.Text = stext
    MaskedTextBox12.Text = stext
    MaskedTextBox2.Text = stext
End If
End Sub

```

```

Private Sub AlleAuswählenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles
AlleAuswählenToolStripMenuItem.Click

```

```

    CheckBox1.Checked = True
    CheckBox2.Checked = True
    CheckBox3.Checked = True
    CheckBox4.Checked = True
    CheckBox5.Checked = True
    CheckBox6.Checked = True
    CheckBox7.Checked = True
    CheckBox8.Checked = True
    CheckBox9.Checked = True
    CheckBox10.Checked = True
    CheckBox11.Checked = True
    CheckBox12.Checked = True
    CheckBox13.Checked = True
    CheckBox14.Checked = True
    MaskedTextBox1.Enabled = True
    MaskedTextBox2.Enabled = True
    MaskedTextBox3.Enabled = True
    MaskedTextBox4.Enabled = True
    MaskedTextBox5.Enabled = True
    MaskedTextBox6.Enabled = True
    MaskedTextBox7.Enabled = True
    MaskedTextBox8.Enabled = True
    MaskedTextBox9.Enabled = True
    MaskedTextBox10.Enabled = True
    MaskedTextBox11.Enabled = True
    MaskedTextBox12.Enabled = True
    MaskedTextBox13.Enabled = True
    MaskedTextBox14.Enabled = True
End Sub

```

```

Private Sub KeineAuswählenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles
KeineAuswählenToolStripMenuItem.Click

```

```

CheckBox1.Checked = True
CheckBox2.Checked = True
CheckBox3.Checked = True
CheckBox4.Checked = True
CheckBox5.Checked = True
CheckBox6.Checked = True
CheckBox7.Checked = True
CheckBox8.Checked = True
CheckBox9.Checked = True
CheckBox10.Checked = True
CheckBox11.Checked = True
CheckBox12.Checked = True
CheckBox13.Checked = True
CheckBox14.Checked = True
MaskedTextBox1.Enabled = True
MaskedTextBox2.Enabled = True
MaskedTextBox3.Enabled = True
MaskedTextBox4.Enabled = True
MaskedTextBox5.Enabled = True
MaskedTextBox6.Enabled = True
MaskedTextBox7.Enabled = True
MaskedTextBox8.Enabled = True
MaskedTextBox9.Enabled = True
MaskedTextBox10.Enabled = True
MaskedTextBox11.Enabled = True
MaskedTextBox12.Enabled = True
MaskedTextBox13.Enabled = True
MaskedTextBox14.Enabled = True
CheckBox1.Checked = False
CheckBox2.Checked = False
CheckBox3.Checked = False
CheckBox4.Checked = False
CheckBox5.Checked = False
CheckBox6.Checked = False
CheckBox7.Checked = False
CheckBox8.Checked = False
CheckBox9.Checked = False
CheckBox10.Checked = False
CheckBox11.Checked = False
CheckBox12.Checked = False
CheckBox13.Checked = False
CheckBox14.Checked = False
End Sub

Private Sub KeinAutostartToolStripMenuItem_Click_1(sender As System.Object, e As System.EventArgs) Handles
KeinAutostartToolStripMenuItem.Click
    If KeinAutostartToolStripMenuItem.Checked = True Then
        KeinAutostartToolStripMenuItem.Checked = False
        My.Settings.Autostart = True
        File.Delete(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk")
        CreateShortcut(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk",
My.Application.Info.DirectoryPath & "\Ganymed.exe")
        My.Settings.Save()
        logger.logfiler("Autostarteintrag wurde gesetzt")
        logger.logfiler("-----")
    Else
        KeinAutostartToolStripMenuItem.Checked = True
        My.Settings.Autostart = False
        File.Delete(Environment.GetFolderPath(Environment.SpecialFolder.Startup) & "\Ganymed.lnk")
        My.Settings.Save()
        logger.logfiler("Autostarteintrag wurde entfernt")
        logger.logfiler("-----")
    End If
End Sub

Private Sub BeendenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs) Handles
BeendenToolStripMenuItem.Click
    Call Button2_Click(Nothing, Nothing)
End Sub

Private Sub Button5_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click
    INF.Show()
End Sub

#End Region

#Region "Comboboxen"

Function switch(ByVal cmb As ComboBox, che As CheckBox)
    Dim ans As Boolean
    Try

```

```

        If cmb.SelectedIndex = 1 Then
            che.Checked = False
        Else
            che.Checked = True
        End If
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
        Return ans = False
    End Try
    Return ans = True
End Function

Private Sub ComboBox1_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    switch(ComboBox1, CheckBox4)
End Sub

Private Sub ComboBox2_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
    switch(ComboBox2, CheckBox6)
End Sub

Private Sub ComboBox3_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox3.SelectedIndexChanged
    switch(ComboBox3, CheckBox8)
End Sub

Private Sub ComboBox4_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox4.SelectedIndexChanged
    switch(ComboBox4, CheckBox10)
End Sub

Private Sub ComboBox5_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox5.SelectedIndexChanged
    switch(ComboBox5, CheckBox12)
End Sub

Private Sub ComboBox6_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox6.SelectedIndexChanged
    switch(ComboBox6, CheckBox14)
End Sub

Private Sub ComboBox7_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles
ComboBox7.SelectedIndexChanged
    switch(ComboBox7, CheckBox1)
End Sub

#End Region

#Region "MaskedTextBoxen Prüfen"

Private Sub MaskedTextBox1_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox1.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox2_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox2.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox3_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox3.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

```

```

Private Sub MaskedTextBox4_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox4.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox5_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox5.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox6_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox6.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox7_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox7.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox8_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox8.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox9_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox9.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox10_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox10.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox11_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox11.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

Private Sub MaskedTextBox12_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox12.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
        End If
    End Sub

```

```

Private Sub MaskedTextBox13_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox13.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
    End If
End Sub

Private Sub MaskedTextBox14_TypeValidationCompleted(sender As Object, e As
System.Windows.Forms.TypeValidationEventArgs) Handles MaskedTextBox14.TypeValidationCompleted
    If (Not e.IsValidInput) Then
        MsgBox("Geben Sie eine gültige Zeit zwischen 00:00 und 23:59 ein", MsgBoxStyle.Exclamation,
"Falsche Eingabe")
    Else
    End If
End Sub

#End Region

#Region "Ganzes Datum Anzeigen in Gruppen"

Dim arr As Array = wk.tagwaehlen
Private Sub Label12_MouseHover(sender As Object, e As System.EventArgs) Handles Label12.MouseHover
    Tooltip1.SetToolTip(Label12, GroupBox1.Text & " " & arr(0).ToString)
End Sub

Private Sub Label13_MouseHover(sender As Object, e As System.EventArgs) Handles Label13.MouseHover
    Tooltip1.SetToolTip(Label13, GroupBox2.Text & " " & arr(1).ToString)
End Sub

Private Sub Label19_MouseHover(sender As Object, e As System.EventArgs) Handles Label19.MouseHover
    Tooltip1.SetToolTip(Label19, GroupBox3.Text & " " & arr(2).ToString)
End Sub

Private Sub Label15_MouseHover(sender As Object, e As System.EventArgs) Handles Label15.MouseHover
    Tooltip1.SetToolTip(Label15, GroupBox4.Text & " " & arr(3).ToString)
End Sub

Private Sub Label16_MouseHover(sender As Object, e As System.EventArgs) Handles Label16.MouseHover
    Tooltip1.SetToolTip(Label16, GroupBox5.Text & " " & arr(4).ToString)
End Sub

Private Sub Label17_MouseHover(sender As Object, e As System.EventArgs) Handles Label17.MouseHover
    Tooltip1.SetToolTip(Label17, GroupBox6.Text & " " & arr(5).ToString)
End Sub

Private Sub Label18_MouseHover(sender As Object, e As System.EventArgs) Handles Label18.MouseHover
    Tooltip1.SetToolTip(Label18, GroupBox7.Text & " " & arr(6).ToString)
End Sub

#End Region

End Class

```

```

Imports System.IO
Imports System.Security.Cryptography

```

```

Public Class Blacklist
    Dim path As String = My.Computer.FileSystem.SpecialDirectories.Temp & "\Prozesse.blp"

    Public Function Ired(ByVal ListBox As ListBox, ByVal datn As String) As Boolean
        Dim res As Boolean = False
        If IO.File.Exists(datn) Then
            Try
                With ListBox.Items
                    .Clear()
                    .AddRange(My.Computer.FileSystem.ReadAllText( _
                        datn, System.Text.Encoding.Default).Split(vbCrLf))
                End With
                res = True
            Catch ex As Exception
                logger.logfiler("-----")
                logger.logfiler(ex.Message)
                logger.logfiler("-----")
            End Try
        End If
        Return (res)
    End Function
End Class

```



```

End Function

Public Function lsave(ByVal ListBox As ListBox, ByVal Filename As String) As Boolean
    Dim str As IO.StreamWriter = Nothing
    Try
        str = New IO.StreamWriter(Filename, False, System.Text.Encoding.Default)
        With ListBox.Items
            For i As Integer = 0 To .Count - 1
                str.Write(.Item(i) & IIf(i < .Count - 1, vbCrLf, ""))
            Next
        End With
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
    Return Nothing
End Function

Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
    Me.Close()
End Sub

Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
    Dim selected As String = ListBox1.SelectedItem.ToString
    For i As Integer = 0 To ListBox2.Items.Count - 1
        ListBox2.SelectedIndex = i
        If ListBox2.SelectedItem.ToString = selected Then
            MsgBox("Sie können einen Prozess nur einmal hinzufügen!" & _
                vbCrLf & "You can add an Process only once!" & _
                vbCrLf & "Vous pouvez ajouter un processus seule une fois!" & _
                vbCrLf & "È possibile aggiungere un processo solo una volta!", MsgBoxStyle.Information)
            Exit Sub
        End If
    Next
    Try
        ListBox2.Items.Add(selected)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Blacklist_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
    ListBox1.Items.Clear()
    For Each Prozess As Process In Process.GetProcesses()
        ListBox1.Items.Add(Prozess.ProcessName.ToString)
    Next Prozess
    Me.Text = "Ganymed - Blacklist"
    If File.Exists(path) = True Then
        ListBox2.Items.Clear()
        lred(ListBox2, path)
    End If
    ListBox1.SelectedIndex = 0
    ToolTip1.SetToolTip(Button7, "Liste importieren")
    ToolTip1.SetToolTip(Button8, "Liste exportieren")
    ToolTip1.SetToolTip(Button3, "Selektion in Blacklist übernehmen")
    ToolTip1.SetToolTip(Button4, "Selektion aus Blacklist entfernen")
    ToolTip1.SetToolTip(Button6, "Alle aus Blacklist entfernen")
End Sub

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click
    Dim selentf As Integer = ListBox2.SelectedIndex
    Try
        ListBox2.Items.RemoveAt(selentf)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
    If File.Exists(path) = True Then
        File.Delete(path)
    Else
    End If
    Dim i As Integer = 0
    For Each item In ListBox2.Items
        If i = ListBox2.Items.Count - 1 Then

```

```

        TextBox2.Text = TextBox2.Text + ListBox2.Items.Item(i)
    Else
        TextBox2.Text = TextBox2.Text + ListBox2.Items.Item(i) & vbCrLf
    End If
    i = i + 1
Next
My.Computer.FileSystem.WriteAllText(path, TextBox2.Text, False)
Me.Close()
End Sub

Private Sub Button5_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click
    Dim selected As String = TextBox1.Text.ToString
    For i As Integer = 0 To ListBox2.Items.Count - 1
        ListBox2.SelectedIndex = i
        If ListBox2.SelectedItem.ToString = selected Then
            MsgBox("Sie können einen Prozess nur einmal hinzufügen!" & _
                vbCrLf & "You can add an Process only once!" & _
                vbCrLf & "Vous pouvez ajouter un processus seule une fois!" & _
                vbCrLf & "È possibile aggiungere un processo solo una volta!", MsgBoxStyle.Information)
            Exit Sub
        End If
    Next
    Try
        ListBox2.Items.Add(selected)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button6_Click(sender As System.Object, e As System.EventArgs) Handles Button6.Click
    ListBox2.Items.Clear()
End Sub

Private Sub Button7_Click(sender As System.Object, e As System.EventArgs) Handles Button7.Click
    Dim ret As Byte
    Dim open As New OpenFileDialog
    open.Filter = "Blacklist|*.blp"
    open.Title = "Blacklist-Import"
    ret = open.ShowDialog()
    Try
        If ret = vbOK Then
            logger.logfiler("Importiere Blacklist")
            logger.logfiler("Pfad: " & open.FileName)
            logger.logfiler("Import abgeschlossen")
            logger.logfiler("-----")
        Else
            logger.logfiler("Beim Import einer Blacklist ist ein Fehler aufgetreten")
            logger.logfiler("Fehler bei der Pfadangabe, es wurde keine Datei ausgewählt")
            logger.logfiler("-----")
        End If
        Ired(ListBox2, open.FileName)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button8_Click(sender As System.Object, e As System.EventArgs) Handles Button8.Click
    Dim ret As Byte
    Dim save As New SaveFileDialog
    save.Filter = "Blacklist|*.blp"
    save.Title = "Blacklist-Export"
    ret = save.ShowDialog()
    Try
        If ret = vbOK Then
            logger.logfiler("Exportiere Blacklist")
            logger.logfiler("Pfad: " & save.FileName)
            logger.logfiler("Export abgeschlossen")
            logger.logfiler("-----")
        Else
            logger.logfiler("Beim Export einer Blacklist ist ein Fehler aufgetreten")
            logger.logfiler("Fehler bei der Pfadangabe, es wurde kein Zielverzeichnis angegeben")
            logger.logfiler("-----")
        End If
        Isave(ListBox2, save.FileName)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
    End Try
End Sub

```

```

        logger.logfiler("-----")
    End Try
End Sub

End Class

Imports System.IO
Imports System.Security.Cryptography

Public Class Blacklist

    Dim path As String = My.Computer.FileSystem.SpecialDirectories.Temp & "\Prozesse.blp"

    Public Function lred(ByVal ListBox As ListBox, ByVal datn As String) As Boolean
        Dim res As Boolean = False
        If IO.File.Exists(datn) Then
            Try
                With ListBox.Items
                    .Clear()
                    .AddRange(My.Computer.FileSystem.ReadAllText( _
                        datn, System.Text.Encoding.Default).Split(vbCrLf))
                End With
                res = True
            Catch ex As Exception
                logger.logfiler("-----")
                logger.logfiler(ex.Message)
                logger.logfiler("-----")
            End Try
        End If
        Return (res)
    End Function

    Public Function lsave(ByVal ListBox As ListBox, ByVal Filename As String) As Boolean
        Dim str As IO.StreamWriter = Nothing
        Try
            str = New IO.StreamWriter(Filename, False, System.Text.Encoding.Default)
            With ListBox.Items
                For i As Integer = 0 To .Count - 1
                    str.Write(.Item(i) & IIf(i < .Count - 1, vbCrLf, ""))
                Next
            End With
        Catch ex As Exception
            logger.logfiler("-----")
            logger.logfiler(ex.Message)
            logger.logfiler("-----")
        End Try
        Return Nothing
    End Function

    Private Sub Button2_Click(sender As System.Object, e As System.EventArgs) Handles Button2.Click
        Me.Close()
    End Sub

    Private Sub Button3_Click(sender As System.Object, e As System.EventArgs) Handles Button3.Click
        Dim selected As String = ListBox1.SelectedItem.ToString
        For i As Integer = 0 To ListBox2.Items.Count - 1
            ListBox2.SelectedIndex = i
            If ListBox2.SelectedItem.ToString = selected Then
                MsgBox("Sie können einen Prozess nur einmal hinzufügen!" & _
                    vbCrLf & "You can add an Process only once!" & _
                    vbCrLf & "Vous pouvez ajouter un processus seule une fois!" & _
                    vbCrLf & "È possibile aggiungere un processo solo una volta!", MsgBoxStyle.Information)
                Exit Sub
            End If
        Next
        Try
            ListBox2.Items.Add(selected)
        Catch ex As Exception
            logger.logfiler("-----")
            logger.logfiler(ex.Message)
            logger.logfiler("-----")
        End Try
    End Sub

    Private Sub Blacklist_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        ListBox1.Items.Clear()
        For Each Prozess As Process In Process.GetProcesses()

```

```

        ListBox1.Items.Add(Prozess.ProcessName.ToString)
    Next Prozess
    Me.Text = "Ganymed - Blacklist"
    If File.Exists(path) = True Then
        ListBox2.Items.Clear()
        lred(ListBox2, path)
    End If
    ListBox1.SelectedIndex = 0
    ToolTip1.SetToolTip(Button7, "Liste importieren")
    ToolTip1.SetToolTip(Button8, "Liste exportieren")
    ToolTip1.SetToolTip(Button3, "Selektion in Blacklist übernehmen")
    ToolTip1.SetToolTip(Button4, "Selektion aus Blacklist entfernen")
    ToolTip1.SetToolTip(Button6, "Alle aus Blacklist entfernen")
End Sub

Private Sub Button4_Click(sender As System.Object, e As System.EventArgs) Handles Button4.Click
    Dim selentf As Integer = ListBox2.SelectedIndex
    Try
        ListBox2.Items.RemoveAt(selentf)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button1_Click(sender As System.Object, e As System.EventArgs) Handles Button1.Click
    If File.Exists(path) = True Then
        File.Delete(path)
    Else
    End If
    Dim i As Integer = 0
    For Each item In ListBox2.Items
        If i = ListBox2.Items.Count - 1 Then
            TextBox2.Text = TextBox2.Text + ListBox2.Items.Item(i)
        Else
            TextBox2.Text = TextBox2.Text + ListBox2.Items.Item(i) & vbCrLf
        End If
        i = i + 1
    Next
    My.Computer.FileSystem.WriteAllText(path, TextBox2.Text, False)
    Me.Close()
End Sub

Private Sub Button5_Click(sender As System.Object, e As System.EventArgs) Handles Button5.Click
    Dim selected As String = TextBox1.Text.ToString
    For i As Integer = 0 To ListBox2.Items.Count - 1
        ListBox2.SelectedIndex = i
        If ListBox2.SelectedItem.ToString = selected Then
            MsgBox("Sie können einen Prozess nur einmal hinzufügen!" & _
                vbCrLf & "You can add an Process only once!" & _
                vbCrLf & "Vous pouvez ajouter un processus seule une fois!" & _
                vbCrLf & "È possibile aggiungere un processo solo una volta!", MsgBoxStyle.Information)
            Exit Sub
        End If
    Next
    Try
        ListBox2.Items.Add(selected)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

Private Sub Button6_Click(sender As System.Object, e As System.EventArgs) Handles Button6.Click
    ListBox2.Items.Clear()
End Sub

Private Sub Button7_Click(sender As System.Object, e As System.EventArgs) Handles Button7.Click
    Dim ret As Byte
    Dim open As New OpenFileDialog
    open.Filter = "Blacklist|*.blp"
    open.Title = "Blacklist-Import"
    ret = open.ShowDialog()
    Try
        If ret = vbOK Then
            logger.logfiler("Importiere Blacklist")
            logger.logfiler("Pfad: " & open.FileName)
            logger.logfiler("Import abgeschlossen")
            logger.logfiler("-----")
        End If
    End Try

```

```

Else
    logger.logfiler("Beim Import einer Blacklist ist ein Fehler aufgetreten")
    logger.logfiler("Fehler bei der Pfadangabe, es wurde keine Datei ausgewählt")
    logger.logfiler("-----")
End If
lred(ListBox2, open.FileName)
Catch ex As Exception
    logger.logfiler("-----")
    logger.logfiler(ex.Message)
    logger.logfiler("-----")
End Try
End Sub

Private Sub Button8_Click(sender As System.Object, e As System.EventArgs) Handles Button8.Click
    Dim ret As Byte
    Dim save As New SaveFileDialog
    save.Filter = "Blacklist|*.blp"
    save.Title = "Blacklist-Export"
    ret = save.ShowDialog()
    Try
        If ret = vbOK Then
            logger.logfiler("Exportiere Blacklist")
            logger.logfiler("Pfad: " & save.FileName)
            logger.logfiler("Export abgeschlossen")
            logger.logfiler("-----")
        Else
            logger.logfiler("Beim Export einer Blacklist ist ein Fehler aufgetreten")
            logger.logfiler("Fehler bei der Pfadangabe, es wurde kein Zielverzeichnis angegeben")
            logger.logfiler("-----")
        End If
        lsave(ListBox2, save.FileName)
    Catch ex As Exception
        logger.logfiler("-----")
        logger.logfiler(ex.Message)
        logger.logfiler("-----")
    End Try
End Sub

End Class

Public Class Logviewer

    Private Sub Logviewer_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load
        Try
            RichTextBox1.Text = Nothing
            RichTextBox1.LoadFile(My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData &
"\Ganymedlog.log", RichTextBoxStreamType.PlainText)
        Catch ex As Exception
            MsgBox(ex.Message)
            logger.logfiler("-----")
            logger.logfiler(ex.Message)
            logger.logfiler("-----")
        End Try
        RichTextBox1.Focus()
    End Sub

    Private Sub BeinhaltenderOrdnerAnzeigenToolStripMenuItem_Click(sender As System.Object, e As
System.EventArgs) Handles BeinhaltenderOrdnerAnzeigenToolStripMenuItem.Click
        Try
            Microsoft.VisualBasic.Shell("explorer /e,/select," &
My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData & "\Ganymedlog.log",
AppWinStyle.NormalFocus)
        Catch ex As Exception
            logger.logfiler("-----")
            logger.logfiler(ex.Message)
            logger.logfiler("-----")
        End Try
    End Sub

    Private Sub RichTextBox1_GotFocus(sender As Object, e As System.EventArgs) Handles RichTextBox1.GotFocus
        With Me.RichTextBox1
            .SelectionLength = 0
            .SelectionStart = Len(.Text)
        End With
    End Sub
End Class

```

```

    Private Sub AlleEinträgeLöschenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs)
Handles AlleEinträgeLöschenToolStripMenuItem.Click
    Dim sw As New IO.StreamWriter(My.Computer.FileSystem.SpecialDirectories.CurrentUserApplicationData &
"\Ganymedlog.log")
    sw.WriteLine("Ganymed Version: " & My.Application.Info.Version.ToString)
    sw.Close()
    Call Logviewer_Load(Nothing, Nothing)
End Sub

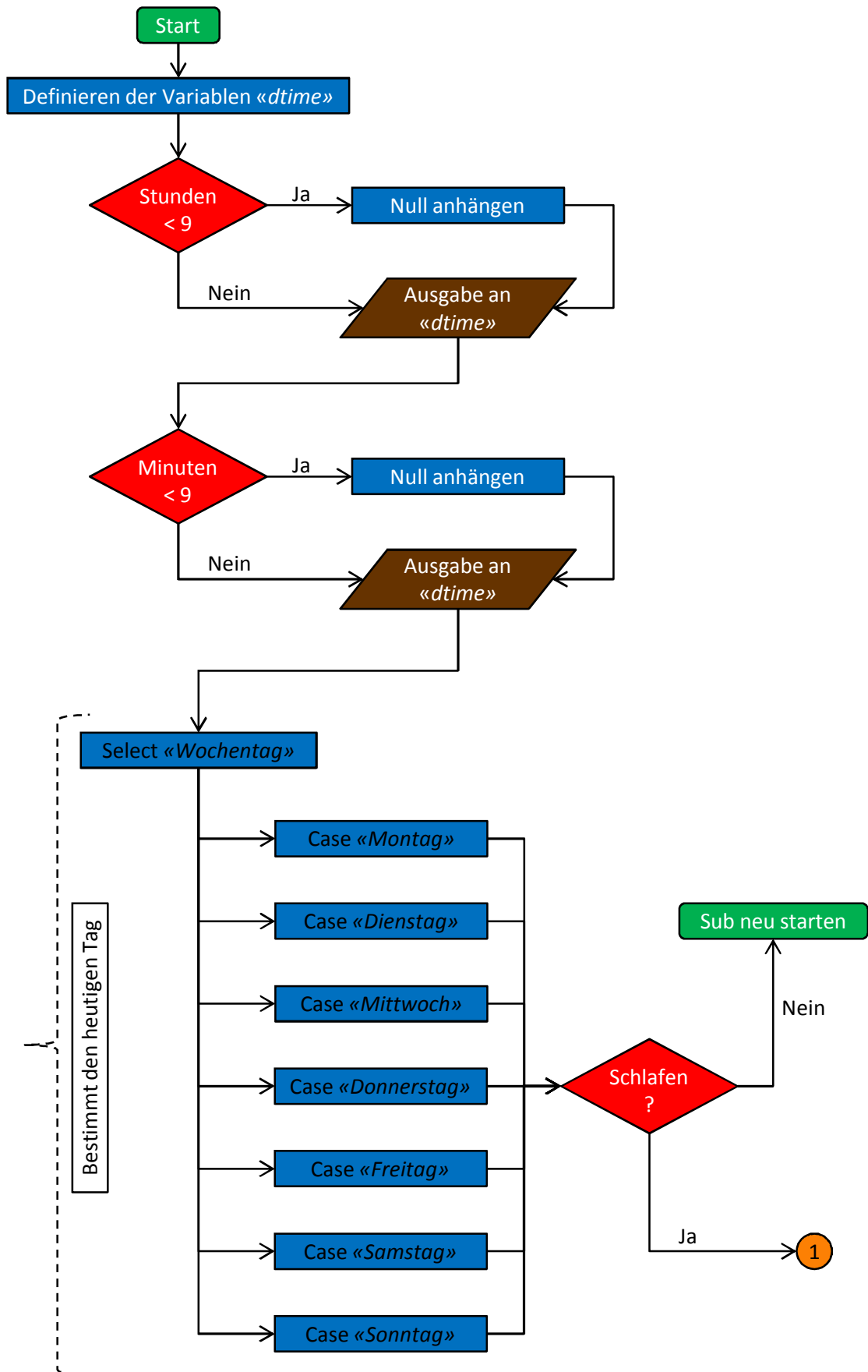
    Private Sub InTextdateiExportierenToolStripMenuItem_Click(sender As System.Object, e As System.EventArgs)
Handles InTextdateiExportierenToolStripMenuItem.Click
    Dim save As New SaveFileDialog
    save.Title = "Ganymed LOG exportieren"
    save.Filter = "Rich Text Format|*.rtf"
    save.ShowDialog()
    Try
        Dim sw As New IO.StreamWriter(save.FileName)
        sw.Write(RichTextBox1.Text)
        sw.Close()
    Catch ex As Exception

    End Try
End Sub

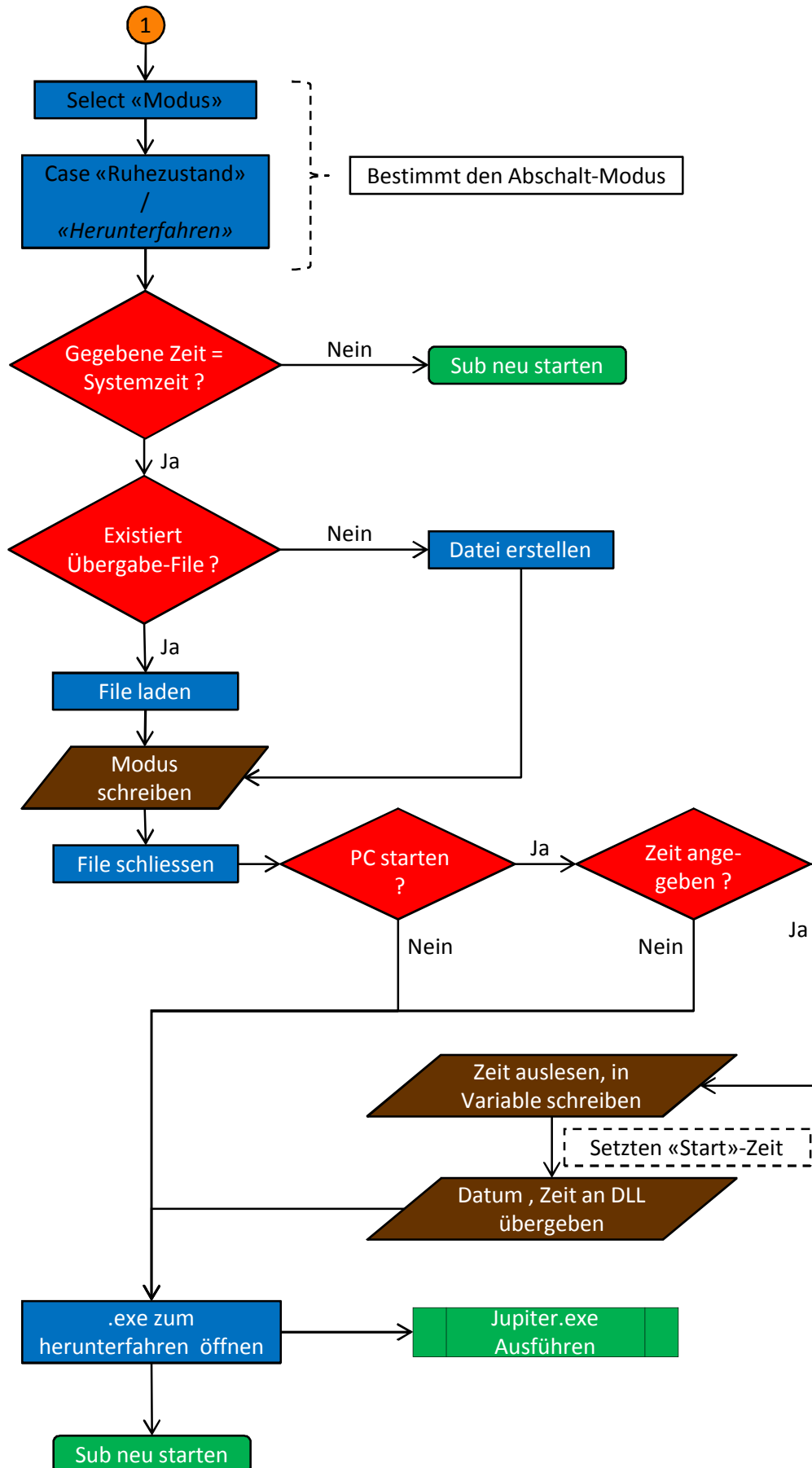
End Class

```

# PAP Timer 3 – Teil 1



## PAP Timer 3 – Teil 2





# Produktdokumentation

# Ganymed

2013

## *Inhaltsverzeichnis*

Inhaltsverzeichnis.....	2
Einführung.....	3
Was ist Ganymed.....	3
Wie funktioniert der Ruhezustand.....	3
Wieso Ruhezustand und nicht Herunterfahren.....	3
Benutzeroberfläche und Hauptfunktionen.....	4
Hauptelemente.....	4
Modus auswählen.....	4
Eingeben einer Zeit für den Ruhezustand.....	5
Eingeben einer Zeit für das Herunterfahren des Systems.....	5
Setzen einer Zeit für das Aufwecken.....	5
Versteckte Funktionen.....	5
Autostart.....	6
Systemeinstellungen.....	6
Blacklist.....	6
Einführung.....	6
Export / Import.....	7
Prozessnamen.....	7
Logfile.....	7
Einführung.....	7
Funktionen.....	8
Dank.....	9
Abbildungsverzeichnis.....	9

## *Einführung*

### Was ist Ganymed

Ganymed ist eine Software die Ihnen dabei hilft Energie zu sparen. Ganymed versetzt den Computer zeitgesteuert in den Ruhezustand und weckt ihn nach einem Zeitplan am nächsten Morgen wieder auf. Dadurch wird während der Nacht keine unnötige Energie verbraucht. Das klingt ein wenig kompliziert, ist es aber nicht. Die einfach gestaltete Benutzeroberfläche von Ganymed kann von fast allen bedient werden. Für fortgeschrittene Benutzer bietet Ganymed auch noch weitere Funktionen, wie zum Beispiel die Blacklist.

### Wie funktioniert der Ruhezustand

Der Ruhezustand funktioniert folgendermassen: Der Computer schreibt alle Daten aus dem Arbeitsspeicher auf die Festplatten. Somit können Sie alle Fenster geöffnet lassen und wenn Sie den Computer wieder starten werden die Fenster wieder angezeigt. In diesem Zustand benötigt der Computer auch keine Energie, das heisst Sie können den Computer vom Netz trennen und wieder verbinden, wenn Sie nun den Computer wieder starten, sollten alle Fenster geöffnet wieder erscheinen. Ist dies nicht der Fall, kann das daran liegen, dass die Stützbatterie des Computers leer ist. Diese Batterie befindet sich auf der Hauptplatine Ihres Computers.

### Wieso Ruhezustand und nicht Herunterfahren

Wenn Sie den Computer herunterfahren, werden alle Prozesse beendet und eben nicht auf die Festplatte geschrieben. So mit startet der Computer schneller und Sie können genau am gleichen Ort weiterarbeiten, wo Sie aufgehört haben. Ausserdem funktioniert das Aufwecken nur aus dem Ruhezustand.

## Benutzeroberfläche und Hauptfunktionen

### Hauptelemente

Die Benutzeroberfläche ist sehr einfach gestaltet und gliedert sich in zwei Teile, den Eingabebereich mit den sieben Wochentagen und den Ereignisbereich mit der aktuellen Uhrzeit und den verschiedenen Buttons. Untenstehend sehen Sie die Benutzeroberfläche:



Abbildung 1 Benutzeroberfläche von Ganymed

Wie Sie sehen gibt es für jeden Tag eine Gruppe. Als Titel steht der Wochentag, auf die in der Gruppe enthaltenen Elemente möchten wir im Folgenden eingehen.

### Modus auswählen

Um Ganymed zu nutzen müssen Sie für jeden Tag einen Modus angeben. Markieren Sie erst das Kästchen mit der Beschriftung „Schlafen“ Unter Modus versteht man hier, welche Aktion beim Erreichen der eingestellten Zeit passieren soll. Möchten Sie, dass der Computer heruntergefahren wird, wählen Sie als Modus Herunterfahren. Soll der Computer aber vom Programm wieder geweckt werden, müssen Sie zwingend Ruhezustand auswählen. Den gewünschten Modus können Sie in der Auswahlliste auswählen, siehe Abbildung 2.

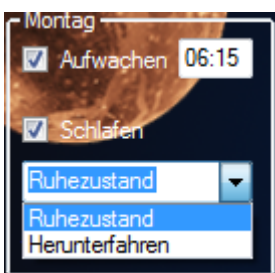


Abbildung 2 Modus

Falls Sie nichts auswählen, wird beim Erreichen des aktuellen Tages automatisch der Modus Ruhezustand ausgewählt. Der Modus wird automatisch beim beenden von Ganymed gespeichert, das heisst, dass Sie nicht bei jedem Start von Ganymed alles neu eingeben müssen.

## Eingeben einer Zeit für den Ruhezustand

Wie bei Modus schon beschrieben, müssen Sie das Kästchen „Schlafen“ auswählen. Erst jetzt können Sie darunter eine Zeit eingeben. Das Format ist durch das Programm gegeben. Jetzt ist in der Gruppe des darauffolgenden Tages auch das Kästchen „Aufwachen“ verfügbar. Wenn Sie eine Zeit eingegeben haben, wird diese immer mit der Systemzeit abgeglichen. Das heisst, sobald die Systemzeit die eingegebene Zeit erreicht, wird der Computer in den Ruhezustand versetzt. Sie können aber bis eine Minute vor Erreichen der eingegebenen Zeit, diese noch ändern. Falls Sie eine Zeit in der Vergangenheit eingeben wird der Computer nicht in den Ruhezustand versetzt.



Abbildung 3 Verschiedene gesetzte Zeiten

## Eingeben einer Zeit für das Herunterfahren des Systems

Dieser Vorgang funktioniert gleich wie das eingeben einer Zeit für den Ruhezustand. Wählen Sie als Modus einfach „Herunterfahren“. Der Unterschied ist, dass das Kästchen „Aufwachen“ in der darauffolgenden Gruppe nicht verfügbar ist.

## Setzen einer Zeit für das Aufwecken

Damit Sie eine Zeit eingeben können, müssen Sie erst das Kästchen „Aufwachen“ aktivieren. Nun geben Sie einfach eine Zeit in das Feld hinter diesem Kästchen ein. Wenn Sie die Abbildung 3 betrachten, sehen Sie, dass der Computer um 22:00 Uhr in den Ruhezustand versetzt wurde und um 06:15 Uhr wieder geweckt wird.

## Versteckte Funktionen

Ganymed bietet verschiedene Funktionen damit Sie noch schneller und einfacher Zeiten setzen oder deaktivieren können. Sie können per Rechtsklick auf das Eingabefeld der Aufweckzeit, diese für alle Aufweckzeiten übernehmen, siehe Abbildung 4. Das gleiche gilt für die Zeiten für den Ruhezustand oder für die Zeiten für das Herunterfahren. Mittels

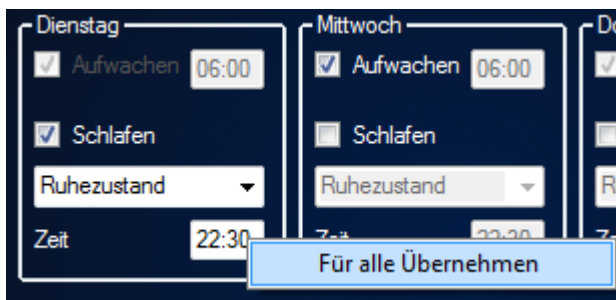


Abbildung 4 Kontextmenü für die Übernahme von Zeiten

Rechtsklick auf ein verfügbares Kästchen können Sie alle Kästchen auswählen lassen. Ausserdem gibt es in diesem Kontextmenü noch die Funktion „Keine Auswählen“, die die Häkchen bei allen entfernt.

## Autostart

Ganymed legt Standardmässig einen Autostarteintrag fest. Dadurch wird gewährleistet, dass Ganymed auch nach einem Neustart des Computers noch ausgeführt wird. Diese Funktion kann natürlich auch deaktiviert werden. Klicken Sie dafür mit der rechten Maustaste auf eine freie Fläche von Ganymed. Dort setzen Sie das Häkchen bei „Kein Autostart“, siehe Abbildung 5.

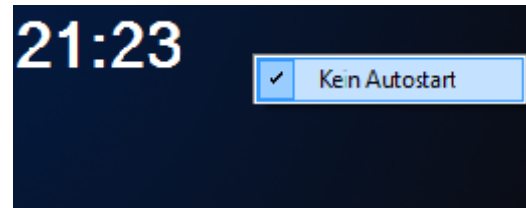


Abbildung 5 Autostart entfernen

## Systemeinstellungen

Trotz richtig eingestellten Zeiten kann es sein, dass Ganymed den Computer nicht mehr starten lässt. Falls das eintritt, starten Sie den Computer manuell und ändern Sie die Energieoptionen wie folgt. Bei Windows 7 finden Sie die Energieoptionen unter *Systemsteuerung\Hardware und Sound\Energieoptionen\Energiesparplaneinstellungen bearbeiten* dort wählen Sie „Erweiterte Energieeinstellungen ändern“ Es erscheint ein Fenster, wo Sie die Einstellungen unter Energie sparen, Zeitgeber zur Aktivierung zulassen, auf Aktivieren setzen müssen.

## Blacklist

### Einführung

Die Blacklist verwaltet die Prozesse, die den Ruhezustand oder das Herunterfahren verzögern sollen. Sie wählen einfach einen Prozess in der linken Liste aus und übernehmen ihn in die Blacklist, oder Sie geben einen anderen Prozess unter „Anderer Prozess“ an.

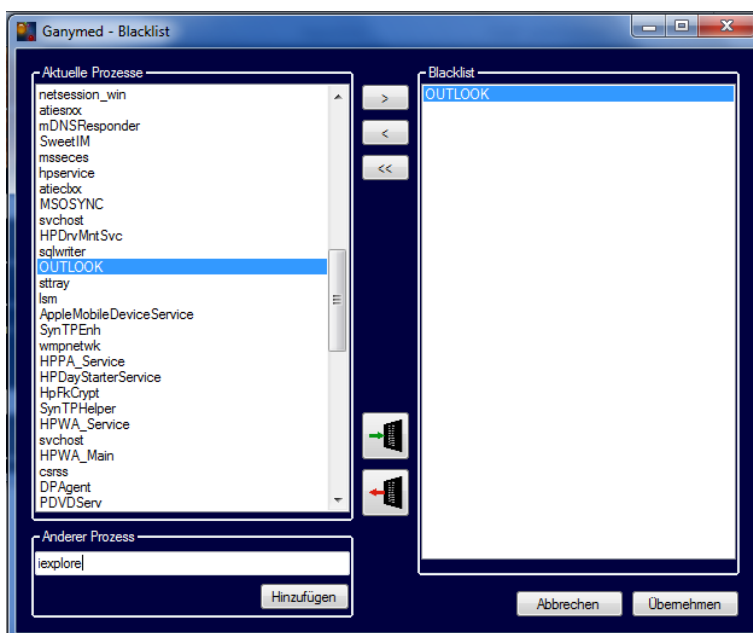


Abbildung 6 Blacklist mit einem gewählten Prozess

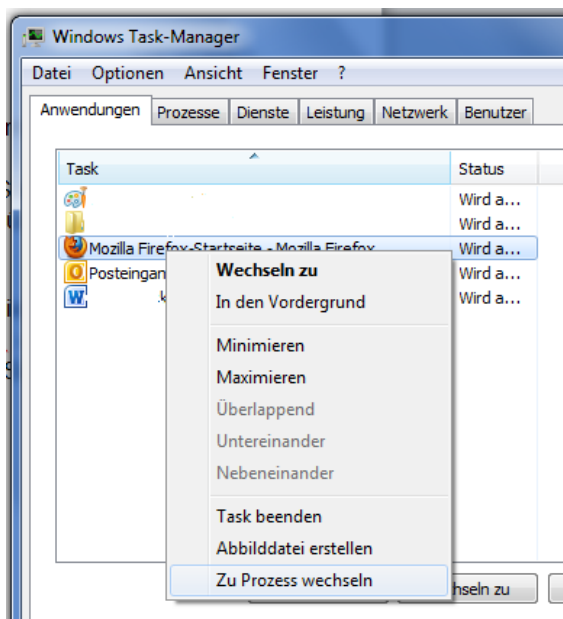
Die Hauptfunktionen zur Übernahme von Prozessen befinden sich im oberen Teil zwischen den beiden Listen. Mittels der Taste „>“ Können Sie einen links ausgewählten Prozess in die Blacklist übernehmen und mittels „<“ wieder entfernen. Mit der Funktion „<<“ entfernen Sie alle Prozesse aus der Blacklist.

## Export / Import

Im unteren Bereich befinden sich die Funktionen für den Export oder für den Import. Nach dem Klick auf die Taste Export erscheint ein Fenster wo Sie ein Zielverzeichnis und einen Namen für die neue Liste angeben können. Diese Funktion dient nur dazu eine Liste von einem Computer auf einen anderen übernehmen zu können, oder um die Liste nicht bei einem Update zu verlieren.

Um eine Liste zu importieren, klicken Sie auf die Taste Import. Wählen Sie im Dialog die zu importierende BLP-Datei aus, beachten Sie, dass die bestehende Liste überschrieben wird.

## Prozessnamen



Falls Sie den Prozessnamen eines Programmes nicht wissen, starten Sie das betreffende Programm und öffnen Sie anschliessend den Task-Manager mittels Ctrl – Alt – Delete. Unter Windows 7 wählen Sie die Registerkarte „Anwendungen“. Dort suchen Sie Ihr Programm und klicken mit der rechten Maustaste darauf, siehe Abbildung 7. Wenn Sie auf „Zu Prozess wechseln“ klicken gelangen Sie direkt zum entsprechenden Prozess. Da Sie nun den Namen des Prozesses wissen, können Sie ihn nach oben genanntem Schema hinzufügen. Geben Sie den Prozess immer ohne die Endung an (exe).

Abbildung 7 Prozess einer Anwendung herausfinden

## Logfile

### Einführung

Das Logfile zeichnet alle möglichen Eingaben und Aktionen des Programmes auf. Im Falle Ganymed wird jeder Systemstart aufgezeichnet. So können Sie einfach und schnell nachvollziehen, wann Ihr Computer gestartet wurde, oder wann der Ruhezustand eingetreten ist. Ebenfalls gespeichert werden Fehler, die auftreten können wenn Sie zum Beispiel eine falsche Eingabe machen.

## Funktionen

Der sogenannte Logviewer zeigt Ihnen das Logfile von Ganymed an. Mittels Rechtsklick können Sie den Ordner anzeigen lassen, der das Logfile enthält. Oder Sie können alle Einträge exportieren und anschliessend löschen.

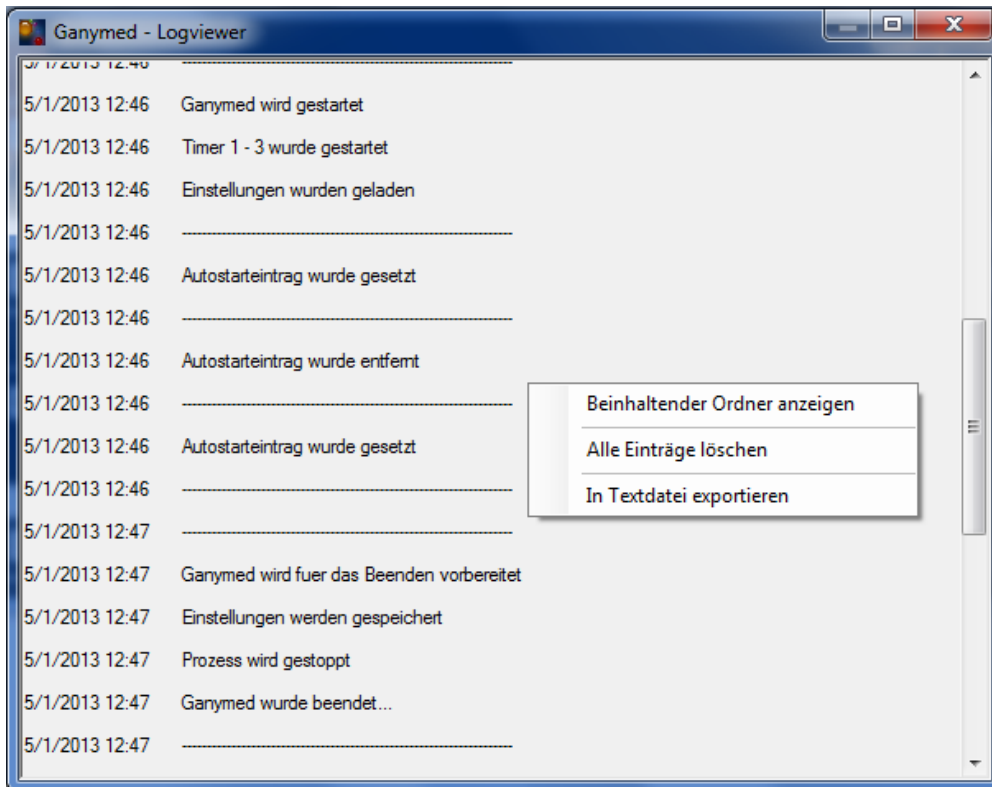


Abbildung 8 Logviewer mit verschiedenen Einträgen



## *Dank*

Ich möchte hal2000 vom Visual Basic Forum VB-Paradise danken. Er hat uns seine wakelib.dll zur Verfügung gestellt. Ohne diese Programmbibliothek wäre Ganymed nicht in der Lage den Computer wieder zu Wecken.

## *Abbildungsverzeichnis*

Abbildung 1	Benutzeroberfläche von Ganymed.....	4
Abbildung 2	Modus.....	4
Abbildung 3	Verschiedene gesetzte Zeiten .....	5
Abbildung 4	Kontextmenü für die Übernahme von Zeiten.....	5
Abbildung 5	Autostart entfernen .....	6
Abbildung 6	Blacklist mit einem gewählten Prozess .....	6
Abbildung 7	Prozess einer Anwendung herausfinden.....	7
Abbildung 8	Logviewer mit verschiedenen Einträgen .....	8



## Programmtestprotokoll

Datum: 7. Dezember 2012

Protokoll-Nr. PTP-12-001-T

Ort: BMS Zürich

Programm: Ganymed V. 1.0.0.5, Jupiter V. 1.0.0.0

Testsystem: Windows 7 Professional 64-bit SP1, HP Probook 4720s

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Installation mit Installer verlief ohne Probleme. Programm konnte anschliessend gestartet werden.

Programmoberfläche: Die Programmoberfläche wird korrekt angezeigt. Tag, Datum und Uhrzeit werden richtig angezeigt. Änderungen an den Aufwachen/Schlafen-Einstellungen werden für den nächsten Programmstart gespeichert.

Blacklist: Die Auswahl eines Prozesses in die Blacklist verhindert programmgemäss den Ruhezustand. Achtung: Es steht „Aktuelle Prozesse“

Ruhezustand: Nach Eingabe der Schlafen-Zeit wird der Computer programmgemäss in den Ruhezustand versetzt. Es wird nur der Freitag getestet.

Logfile: Logfile wird nachgeführt. Die geplanten „Schlafen“-Aktionen werden aufgeführt. Auch die „Schlafen“- und „Aufwachen“-Aktionen werden aufgeführt. Texte können noch optimiert werden. „Aufwach“-Aktion wird doppelt aufgeführt.

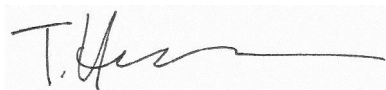
### Bemerkungen:

Die Funktionen des Programms laufen wunschgemäss.

### Verbesserungsvorschläge:

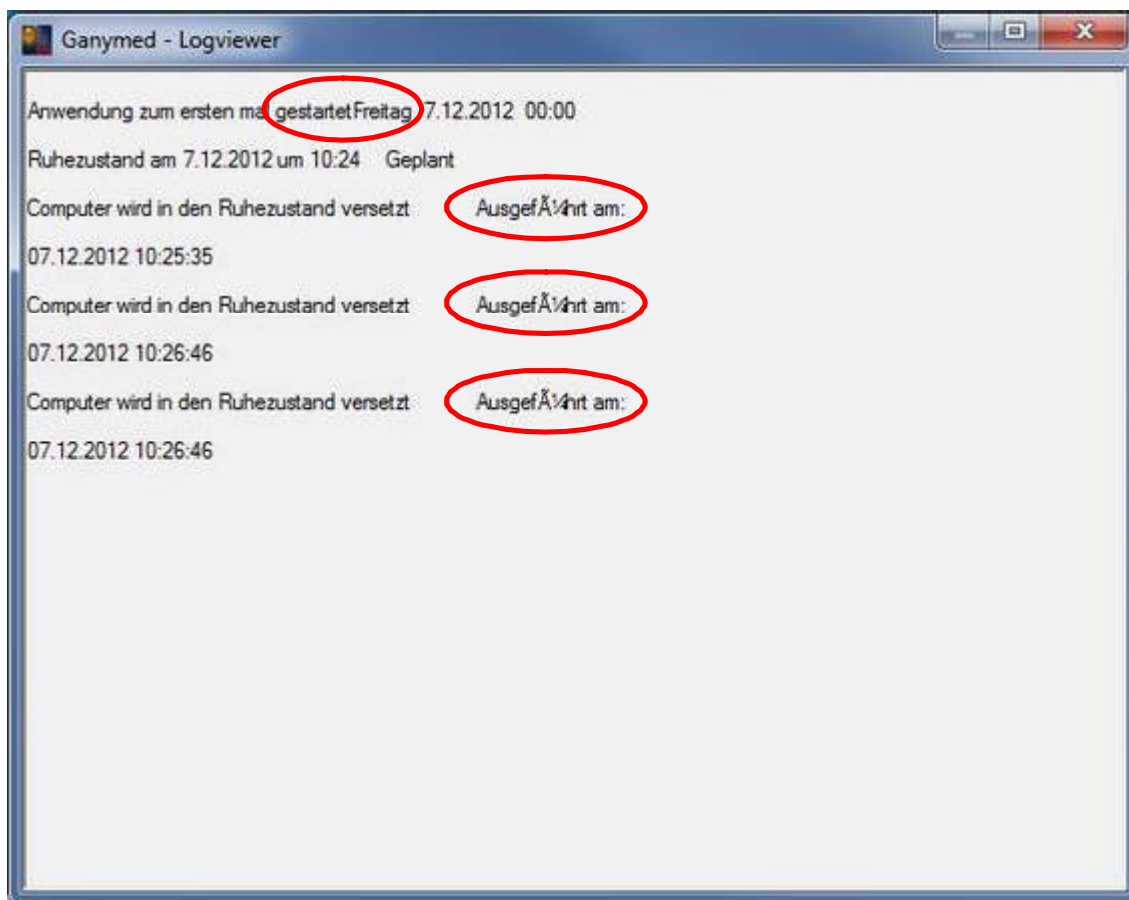
Blacklist: Schreibfehler, Log: s. Bild im Anhang

Unterschrift: \_\_\_\_\_



Stromsparen durch Ruhezustand

## Anhang PTP-12-002-T



## Programmtestprotokoll

Datum: 11. Dezember 2012

Protokoll-Nr. PTP-12-002-R

Ort: Galgenen SZ

Programm: Ganymed V. 1.0.0.5, Jupiter V. 1.0.0.0

Testsystem: Windows 7 Professional 64-bit SP1, Eigenbau

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Das Programm startet ohne Probleme und recht zügig.  
Die Blacklist lässt sich einfach bedienen und der Computer wird erst in den Ruhezustand versetzt wenn das letzte Programm auf der Liste geschlossen wurde.

Das Programm läuft stabil und ist während normaler arbeiten nicht bemerkbar. Somit stört es nicht und das System wird auch nicht verlangsamt.

Nicht getestet:

- Aufweckfunktionen
- Logfile
- Systemstart
- Herunterfahren

### Bemerkungen:

Die Funktionen des Programmes laufen hervorragend.

### Verbesserungsvorschläge:

Schreibfehler in der Blacklist korrigieren.

Unterschrift:     R Züger



## Programmtestprotokoll

Datum: 14. Dezember 2012

Protokoll-Nr. PTP-12-003-T

Ort: BMS Zürich

Programm: Ganymed V. 1.0.0.6

Testsystem: Windows 7 Professional 64-bit SP1, HP Probook 4720s

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Programm schaltet den Computer programmgemäss in den Ruhezustand.

Der Start des Computers am Morgen konnte an mehreren Tagen hintereinander nicht ausgeführt werden. Der Fehler konnte nicht sofort gefunden werden.

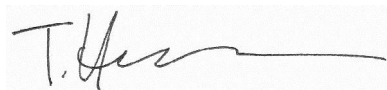
### Bemerkungen:

Die Funktionen des Programms bis auf den Start funktionieren wunschgemäss. Der Systemstart konnte nicht ausgeführt werden.

### Verbesserungsvorschläge:

Es muss festgestellt werden, warum das Programm nicht korrekt funktioniert.

Unterschrift: \_\_\_\_\_







## Programmtestprotokoll

Datum: 15. Dezember 2012

Protokoll-Nr. PTP-12-004-T

Ort: BMS Zürich

Programm: Ganymed V. 1.0.0.6

Testsystem: Windows 7 Professional 64-bit SP1, HP Probook 4720s

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Nach Identifikation der Fehlerquelle, die das Starten des Computers verhindert (s. Bemerkungen) funktionierte Ganymed wunschgemäss. Der Computer konnte gestartet werden.

### Bemerkungen:

Der Fehler, der verhindert, dass das System startet, konnte gefunden werden. Es handelt sich um eine Einstellung in den Erweiterten Energieoptionen des aktuellen Energiesparplans. Diese Einstellung ist teilweise deaktiviert, und verhindert so den Start des Computers mittels eines Zeitgebers.

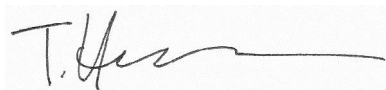
### Verbesserungsvorschläge:

Die Kernfunktionen von Ganymed sind voll Funktionstüchtig.

An der Oberfläche können noch die folgenden Anpassungen durchgeführt werden:

- Funktionen „Alle auswählen“ bzw. „Alle abwählen“
- Export/Import Blacklist
- Autostart funktioniert nicht.

Unterschrift: \_\_\_\_\_





## Programmtestprotokoll

Datum: 15.12.2012 – 20.12.2012 Protokoll-Nr. PTP-12-005-R

Ort: Galgenen

Programm: Ganymed Version 1.0.1.1\_

Testsystem: Windows 7 Professional 64-bit SP1, Eigenbau

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Das Programm wurde über mehrere Tage getestet. Alle gewünschten Aktionen, ob Ruhezustand oder Start, wurden ohne Probleme ausgeführt.

Die Benutzeroberfläche ist einfach zu bedienen.

Bei laufenden Programmen wird der Ruhezustand erfolgreich verzögert.

Seit dieser Version kann man die Blacklist auch exportieren, um sie auf verschiedenen Computern zu nutzen.

### Bemerkungen:

Programm läuft wunschgemäss.

### Verbesserungsvorschläge:

-

Unterschrift: R. Züger



## Programmtestprotokoll

Datum: 21. Dezember 2012

Protokoll-Nr. PTP-12-006-T

Ort: BMS Zürich

Programm: Ganymed V. 1.0.1.1

Testsystem: Windows 7 Professional 64-bit SP1, HP Probook 4720s

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Bei der neusten Version 1.0.1.1 von Ganymed funktionieren alle Funktionen wunschgemäss. Alle offenen Pendenzen wurden in dieser Version abgearbeitet.

Der Ruhezustand wie auch der Systemstart werden korrekt durchgeführt. Die Einträge ins Logfile sind in Ordnung. Die Blacklist ist ebenfalls lauffähig.

### Bemerkungen:

Keine weiteren Bemerkungen

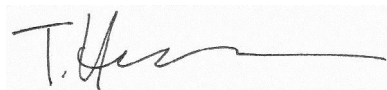
### Verbesserungsvorschläge:

Die Kernfunktionen von Ganymed sind voll Funktionstüchtig.

An der Oberfläche können noch die folgenden Anpassungen durchgeführt werden:

- Funktionen „Alle auswählen“ bzw. „Alle abwählen“
- Export/Import Blacklist
- Autostart funktioniert nicht.

Unterschrift: \_\_\_\_\_





## Programmtestprotokoll

Datum: 02. Februar 2013

Protokoll-Nr. PTP-13-007-T

Ort: Samstagen

Programm: Ganymed V. 1.0.1.1, finale Ausgabe

Testsystem: Windows 7 Professional 64-bit SP1, HP Probook 4720s

Ausgeführte Tests:  Programmstart  Stabilität  Funktionalität  
 Systemstart  Ruhezustand  Herunterfahren

### Testergebnisse:

Ganymed startet zügig und im minimierten Modus auf. Ein Infofeld wird aufgezeigt. Die Darstellung der Benutzeroberfläche entspricht den Vorstellungen. Das Entfernen bzw. Hinzufügen der Autostart-Verknüpfung über das Kontextmenu funktioniert. *Für alle übernehmen* beiden Zeitfenstern bzw. *Alle/Keine Auswählen* bei den Häkchen funktioniert. Blacklist kann angepasst bzw. ex- und importiert werden und verzögert wunschgemäss das Ausschalten. Ruhezustand/Herunterfahren und das Starten funktioniert tadellos. Das angepasste Logfile zeigt alle Aktionen sauber auf.

### Bemerkungen:

Hierbei handelt es sich um einen Abschlusstest. Ganymed liegt bereits auf der CD vor. Bezüglich der Funktionen von Ganymed sind keine Verbesserungsvorschläge da, alle Pendenzen wurden abgearbeitet.

### Verbesserungsvorschläge:

Keine Verbesserungsvorschläge

Unterschrift:







## Messprotokoll

Datum: 17.12.2012 – 19.12.2012 Protokoll-Nr. MP-12- 001-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 1, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

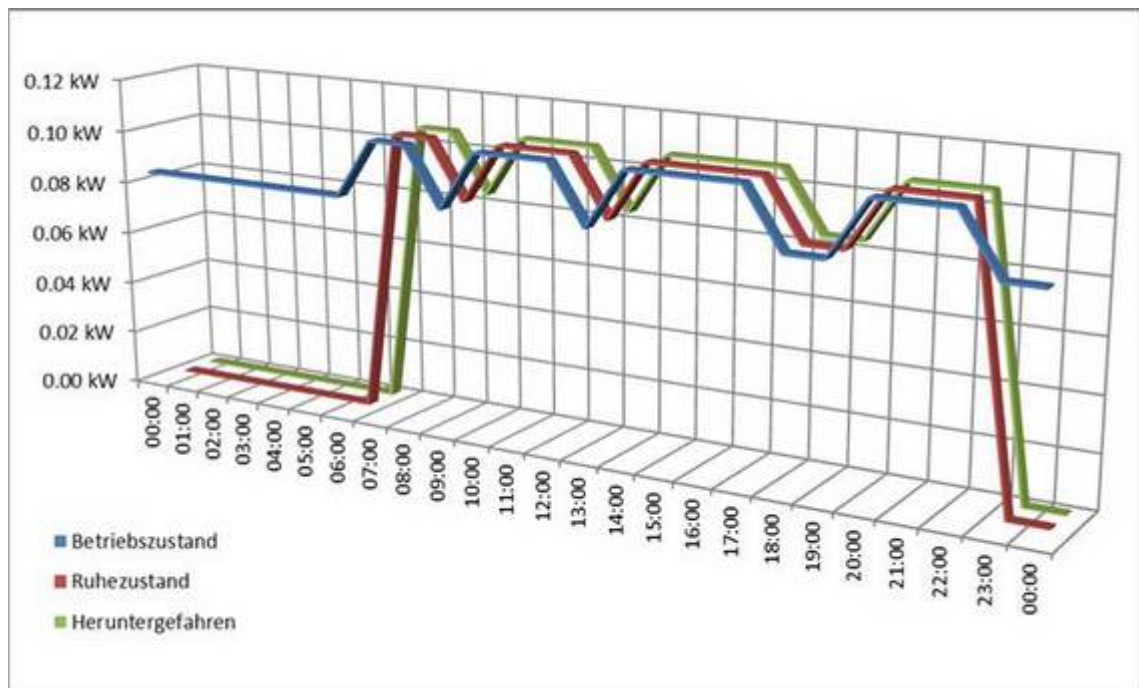
Verbrauch Testcomputer (Laufzeit 24h): 2.26 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 1.50 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 1.50 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.76 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 20.12.2012 – 22.12.2012

Protokoll-Nr. MP-12- 002-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 2, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

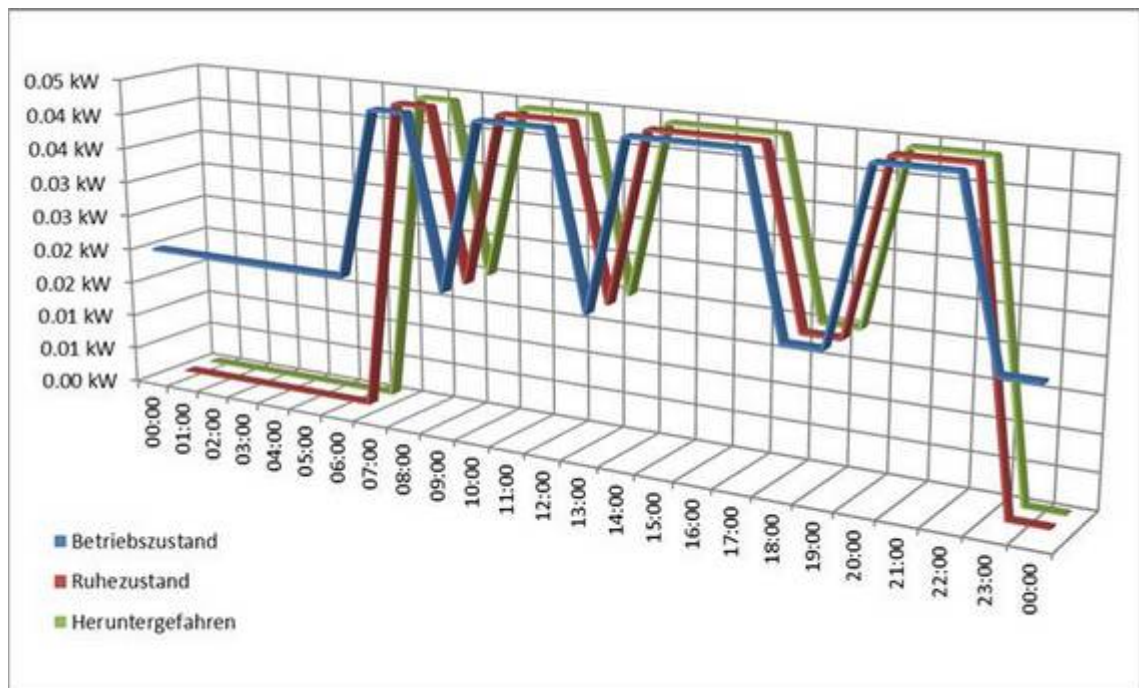
Verbrauch Testcomputer (Laufzeit 24h): 0.74 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.56 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.56 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.18 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 22.12.2012 – 24.12.2012

Protokoll-Nr. MP-12- 003-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 3, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

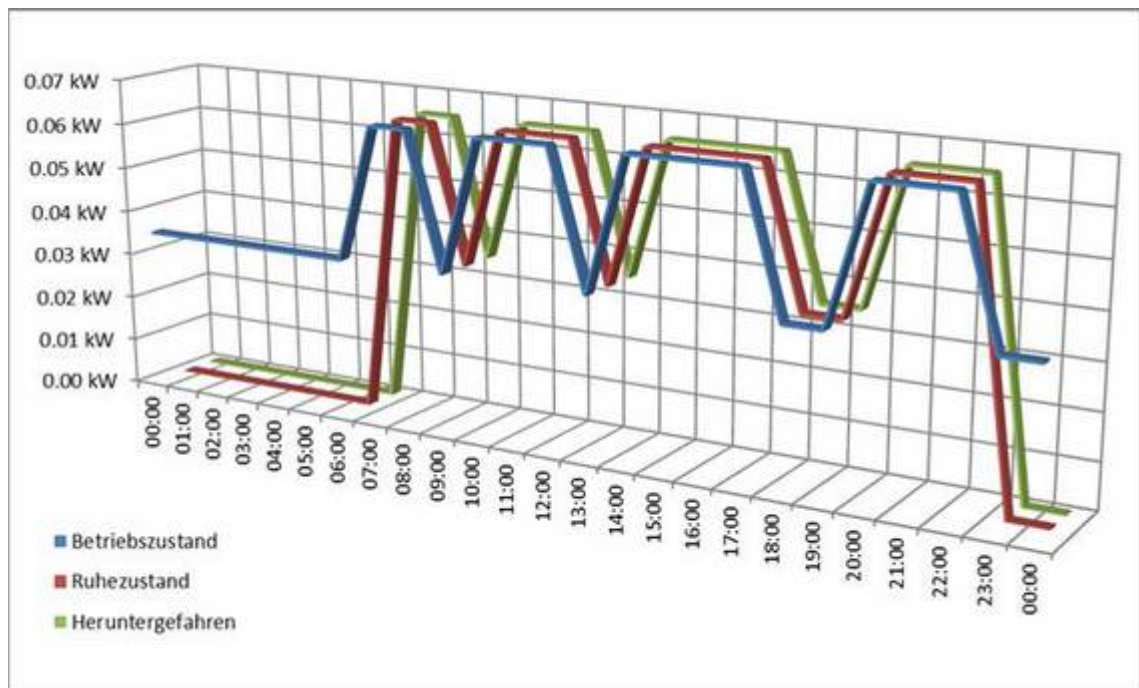
Verbrauch Testcomputer (Laufzeit 24h): 1.17 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.86 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.86 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.31 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 25.12.2012 – 27.12.2012

Protokoll-Nr. MP-12- 004-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 4, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

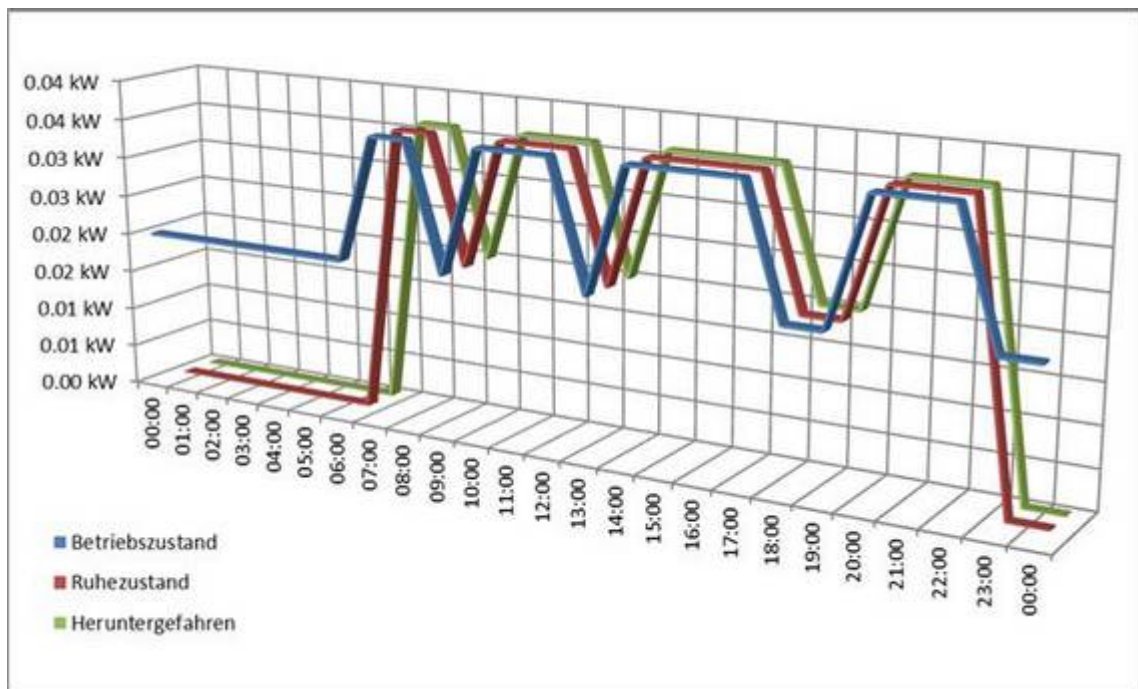
Verbrauch Testcomputer (Laufzeit 24h): 0.62 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.42 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.42 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.20 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger





## Messprotokoll

Datum: 28.12.2012 – 30.12.2012

Protokoll-Nr. MP-12- 005-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 5, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

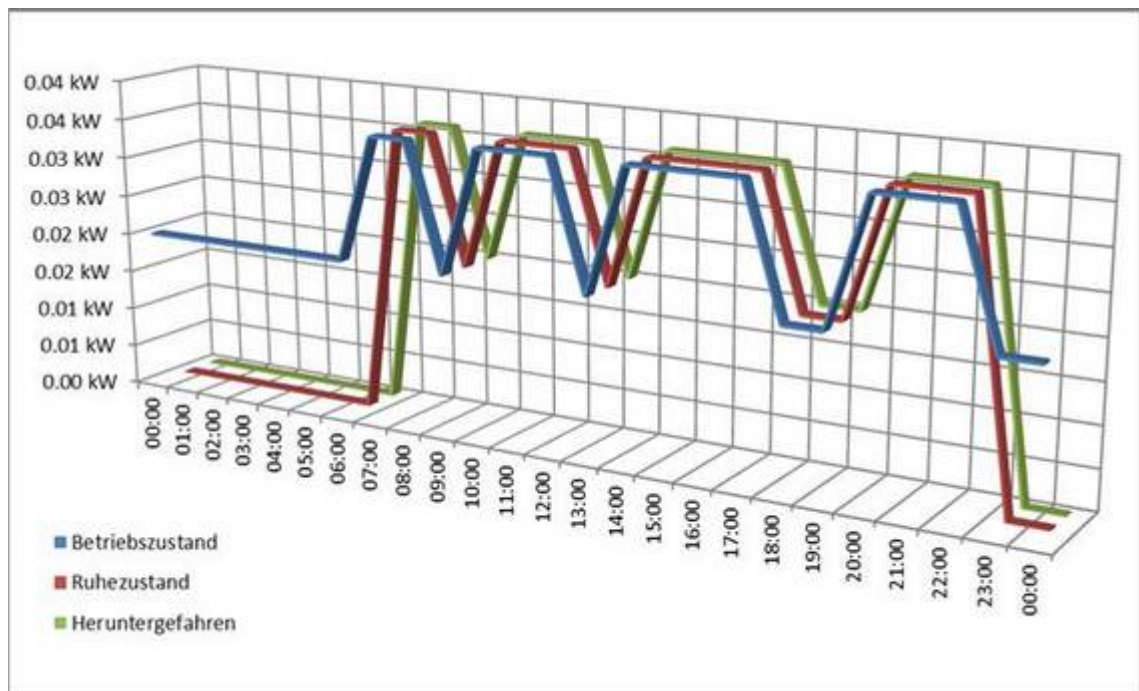
Verbrauch Testcomputer (Laufzeit 24h): 0.66 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.48 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.48 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.18 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 01.01.2013 – 03.01.2013

Protokoll-Nr. MP-13- 006-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 6, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

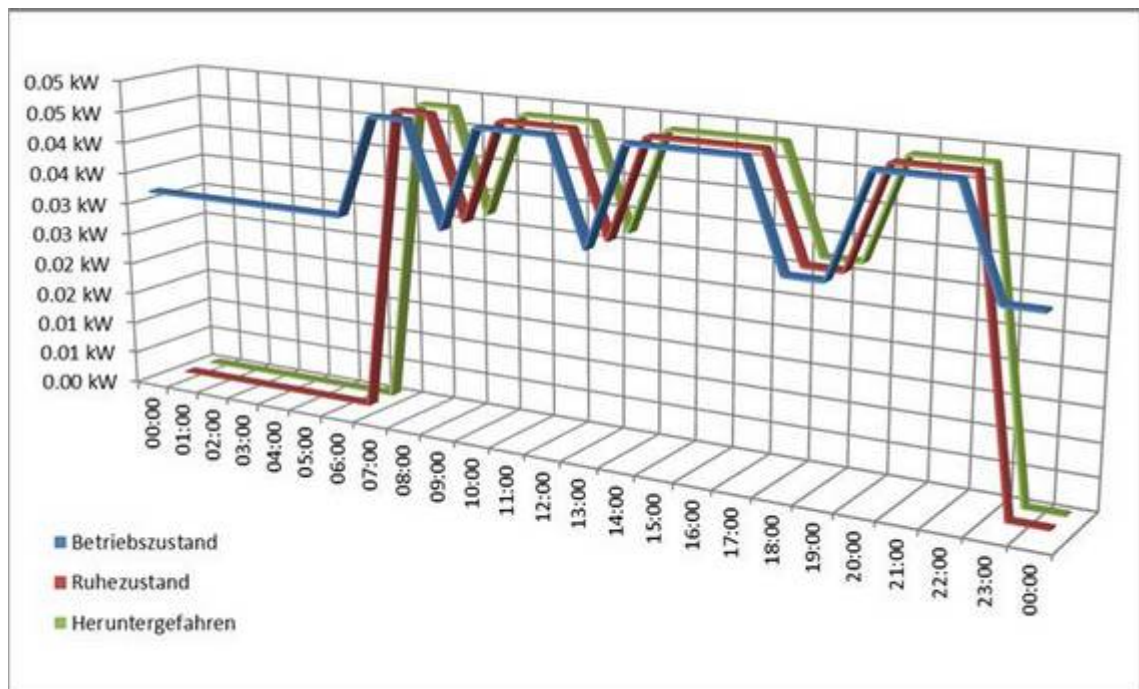
Verbrauch Testcomputer (Laufzeit 24h): 0.94 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.66 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.66 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.28 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 04.01.2013 – 06.01.2013

Protokoll-Nr. MP-13- 007-R

Ort: Galgenen

Messgerät: Power Calculator 1808944

Computer: Testcomputer 7, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

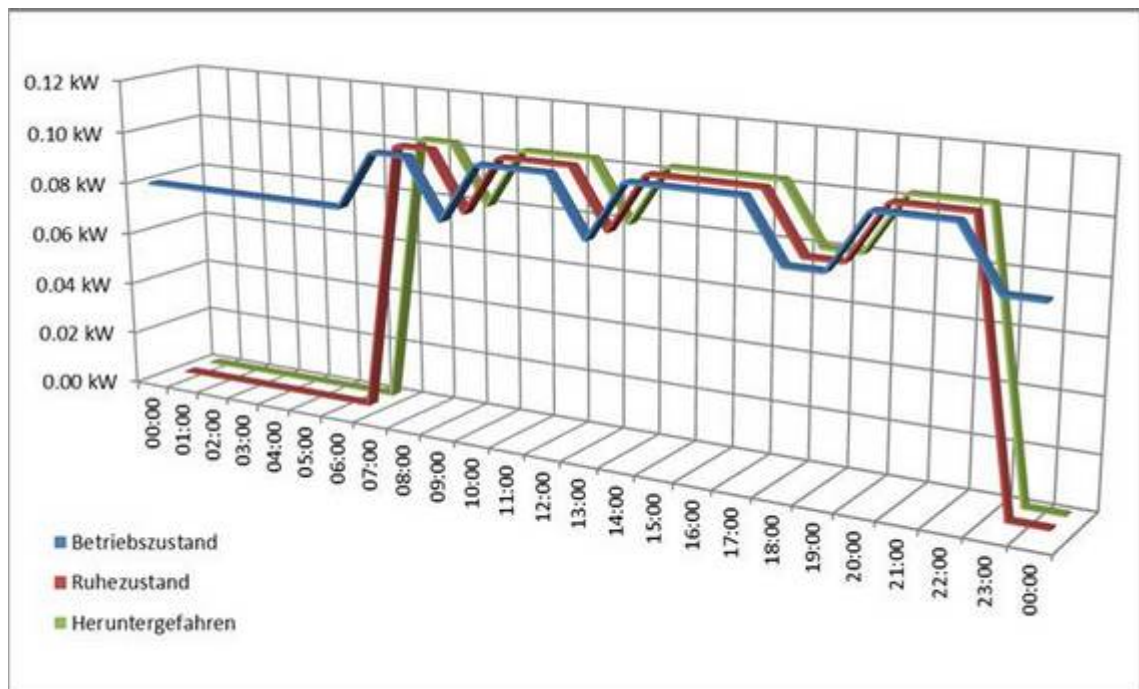
Verbrauch Testcomputer (Laufzeit 24h): 2.16 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 1.44 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 1.44 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.72 kWh pro Tag**

### Diagramm



Unterschrift: R. Züger



## Messprotokoll

Datum: 27.12.2012 – 29.12.2012 Protokoll-Nr. MP-13-008-T

Ort: Samstagen

Messgerät: Tchibo Messgerät (*Modellbezeichnung unbekannt*)

Computer: Testcomputer 8, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: MP-12-001-R

### Messergebnisse:

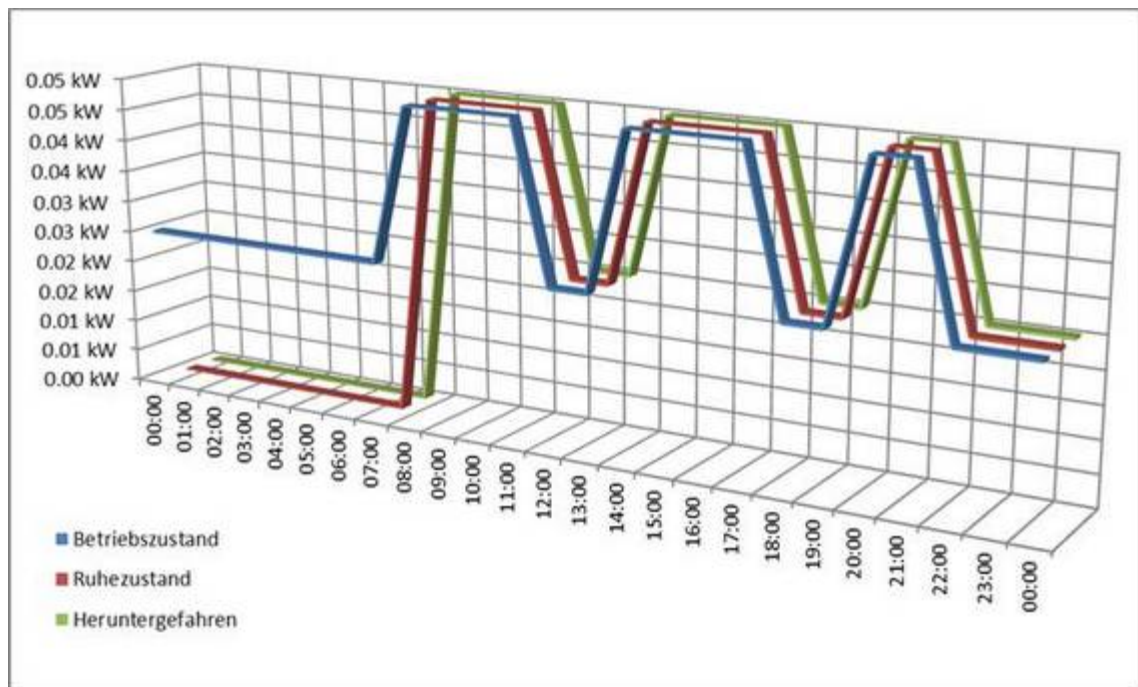
Verbrauch Testcomputer (Laufzeit 24h): 0.88 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 0.65 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 0.65 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.23 kWh pro Tag**

### Diagramm



Unterschrift: T. Hofmann





## Messprotokoll

Datum: 15.01.2013 – 17.01.2013 Protokoll-Nr. MP-13-009-T

Ort: Wädenswil

Messgerät: Tchibo Messgerät (*Modellbezeichnung unbekannt*)

Computer: Testcomputer 9, Details siehe Anhang

Messphase:  Betriebszustand  Ruhezustand  Herunterfahren

Messzeitraum: 24 Stunden

### Messergebnisse:

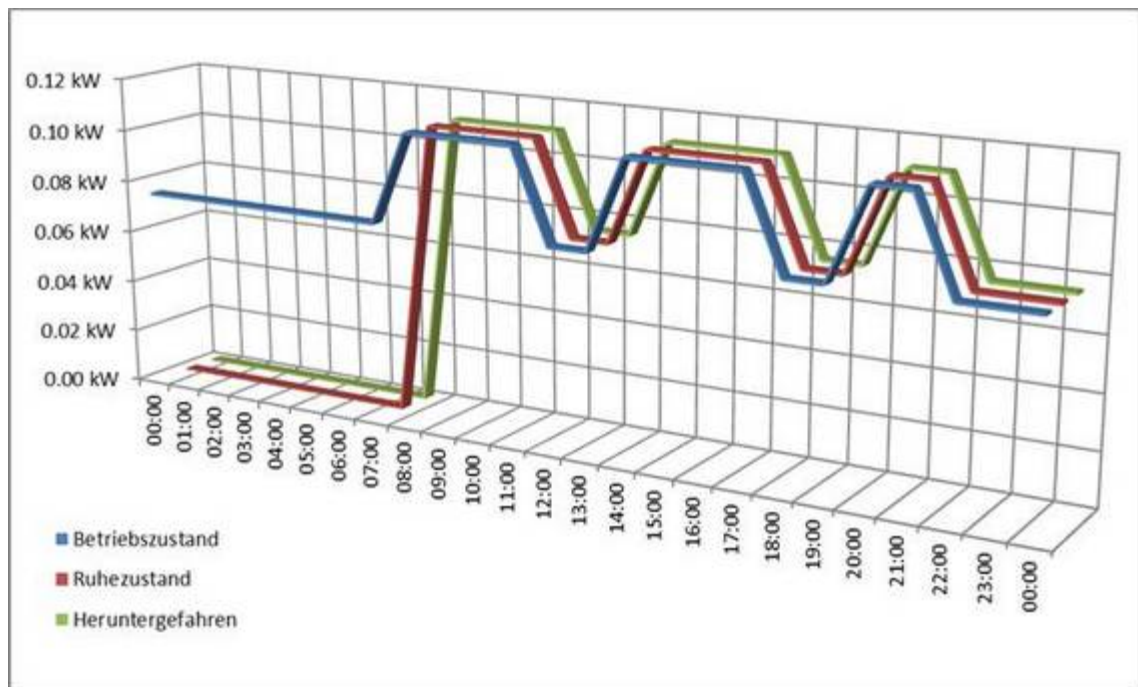
Verbrauch Testcomputer (Laufzeit 24h): 2.19 kWh

Verbrauch Testcomputer (Laufzeit 16h, Ruhezustand 8h): 1.51 kWh

Verbrauch Testcomputer (Laufzeit 16h, Heruntergefahren 8h): 1.51 kWh

**Sparpotential durch Ruhezustand/Herunterfahren: 0.68 kWh pro Tag**

### Diagramm



Unterschrift: T. Hofmann



## Anhang Messprotokolle

<b>Testcomputer 1</b>	
Betriebssystem	Microsoft Windows 7 Home Premium 64-bit
Modell	Eigenbau (Desktop)
Prozessor	Intel Pentium Dual Core E2220 @ 2400 MHz (2 Cores)
RAM	Apacer Technology: 1024 Mbytes @ 333.3 MHz 1024 Mbytes @ 333.3 MHz OCZ: 1024 Mbytes @ 333.3 MHz 1024 Mbytes @ 333.3 MHz
BIOS	American Megatrends Inc. Version 1306 (20.08.2008), 1024 KBytes
<b>Testcomputer 2</b>	
Betriebssystem	Microsoft Windows 7 Home Premium 32-bit
Modell	HP ProBook 4520s (Notebook)
Prozessor	Intel Core i3-350M @ 2266.7 MHz (2 Cores)
RAM	Ramaxel Technology: 2048 Mbytes @ 666.7 MHz Samsung: 4096 Mbytes @ 666.7 MHz
BIOS	Hewlett-Packard Version 68AZZ Vers F.20 (08.09.2011), 1560 KBytes
<b>Testcomputer 3</b>	
Betriebssystem	Microsoft Windows 7 Professional 64-bit
Modell	Eigenbau (Desktop)
Prozessor	Intel Core i7-2600 @ 3400.0 MHz (4 Cores)
RAM	Samsung: 2048 MBytes @ 666.7 MHz 4096 Mbytes @ 666.7 MHz 2048 MBytes @ 666.7 MHz
BIOS	FUJITSU // American Megatrends Inc. V4.6.4.0 R1.24.0 for D3062-A1x(25.05.2012), 8192 KBytes

<b>Testcomputer 4</b>	
Betriebssystem	Microsoft Windows 8 Professional 64-bit
Modell	HP Pavilion dv7 (Notebook)
Prozessor	Intel Core i5-2520M @ 2500.0 MHz (2 Cores)
RAM	Samsung: 4096 Mbytes @ 666.7 MHz
BIOS	Hewlett-Packard 68SCE Ver. F.27 (14.06.2012), 2560 KBytes
<b>Testcomputer 5</b>	
Betriebssystem	Microsoft Windows 8 Professional 64-bit
Modell	HP ProBook 6360b (Notebook)
Prozessor	AMD A4-3300M @ 4000.0 MHz (2 Cores)
RAM	Samsung: 4096 Mbytes @ 666.7 MHz
BIOS	Hewlett-Packard F.35 (17.10.2011), 2048 KBytes
<b>Testcomputer 6</b>	
Betriebssystem	Microsoft Windows 7 Home Premium 32-bit
Modell	HP Pavilion g6 (Notebook)
Prozessor	Intel Core 2 Duo P8700 @ 2533.3 MHz (2 Cores)
RAM	Samsung: 2048 Mbytes @ 400.0 MHz 2048 Mbytes @ 400.0 MHz
BIOS	Hewlett-Packard
<b>Testcomputer 7</b>	
Betriebssystem	Microsoft Windows 7 Enterprise 32-bit
Modell	Eigenbau (Desktop)
Prozessor	Intel Core 2 Duo E6750 @ 2660.0 MHz (2 Cores)
RAM	Samsung: 4096 Mbytes @ 666.7 MHz
BIOS	Hewlett-Packard 786F1 v01.0 (18.07.2007)

## Stromsparen durch Ruhezustand

<b>Testcomputer 8</b>	
Betriebssystem	Microsoft Windows 7 Professional 64-bit
Modell	HP ProBook 4720s (Notebook)
Prozessor	Intel Core i5 M460 @ 2534.0 MHz (2 Cores)
RAM	Kingston:                   4096 Mbytes @ 1333.0 MHz 4096 Mbytes @ 1333.0 MHz
BIOS	Hewlett-Packard 68AZZ Ver. F.20 (08.09.2011)
<b>Testcomputer 9</b>	
Betriebssystem	Microsoft Windows 7 Enterprise 32-bit
Modell	Gigabyte EP45-UD3LR (Desktop)
Prozessor	Intel Core 2 Duo E8400 @ 3000.0 MHz (2 Cores)
RAM	Samsung:                   4096 Mbytes @ 666.7 MHz