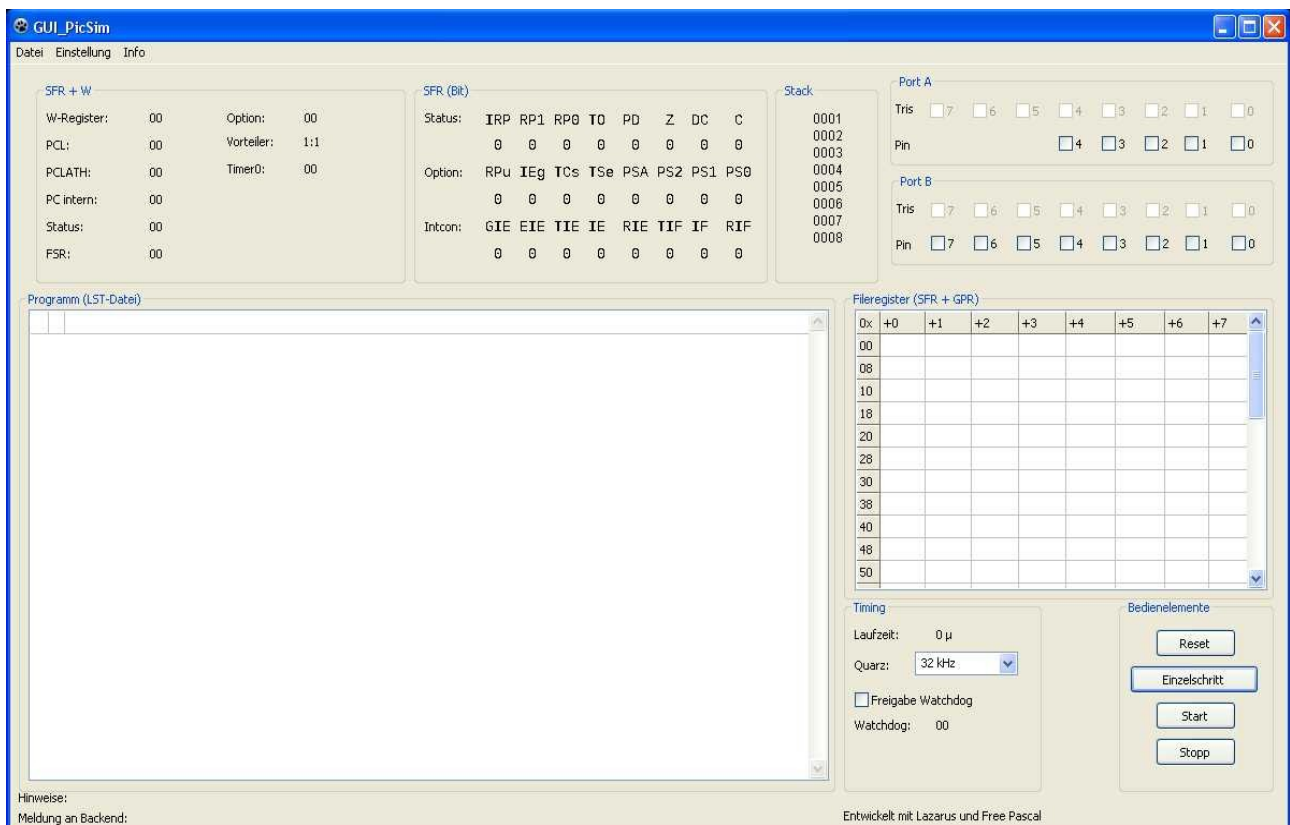


# GUI\_PicSim

Stand: 24.04.2020

Copyright:  
Stefan Lehmann



Diese Software darf nicht gewerblich genutzt werden.

# Inhaltsverzeichnis

0 Aktuelle Änderungen.....	3
1 Einleitung.....	4
2 Installation von GUI_PicSim.....	5
2.1 Unter Windows.....	5
2.2 Unter Linux.....	5
3 Die Oberfläche.....	6
3.1 Fenster für LST-Datei.....	6
3.2 Fenster für File Register.....	6
3.3 Fenster für Timing.....	7
3.4 Fenster SFR + W.....	7
3.5 Fenster SFR (Bit).....	7
3.6 Fenster Stack.....	7
3.7 Fenster Port A.....	7
3.8 Fenster Port B.....	8
3.9 Fenster Bedienelemente.....	8
3.10 Menüpunkt Datei.....	8
Datei laden.....	8
Einstellungen speichern.....	8
Programm beenden.....	8
3.11 Menüpunkt Einstellungen.....	8
3.12 Menüpunkt Hilfe.....	8
4 Datenaustausch.....	9
4.1 Backend an Frontend.....	9
4.2 Frontend an Backend.....	10
5 Anhang I.....	12
5.1 Installation einer RAM-Disk unter Windows.....	12
5.2 Installation einer RAM-Disk unter Linux.....	12

# 0 Aktuelle Änderungen

Version 0.8.1 24.04.2020

Menüpunkt zum Löschen der GUI\_CHANGE Datei.

Anzeige des zuletzt an das Backend gesendeten Befehls.

Zeichensatz im LST-Fenster auf Courier New gesetzt.

Dokumentation überarbeitet

# 1 Einleitung

GUI\_PicSim ist eine vorgefertigte grafische Oberfläche um die Erstellung des PIC-Simulators im Rahmen des Rechnerarchitektur-Praktikums zu vereinfachen. Dabei stellt GUI\_PicSim das Frontend und das eigentliche Simulationsprogramm das Backend dar. Dieses Backend muss in jedem Fall von den Studenten erstellt werden.

Um eine plattformunabhängige Applikation zu erstellen, wurden einige Vereinfachungen und Vorgaben festgelegt. Aktuell lässt sich dieses Programm auf Windows-Rechner starten. Linux wird auch sehr schnell zur Verfügung stehen. Bei MAC-Rechnern wird es noch etwas dauern.

Die Kommunikation zwischen dem Front- und dem Backend erfolgt über einfache Textdateien. Das Frontend erzeugt dabei die Datei GUI\_CHANGE.DAT, das Backend die Datei GUI\_SET.DAT. Beide Dateien können nur geschrieben bzw. erzeugt werden, wenn sie noch nicht existieren. Erst durch das Löschen der Datei durch den Empfänger ist gewährleistet, dass keine Änderungen verloren gehen.

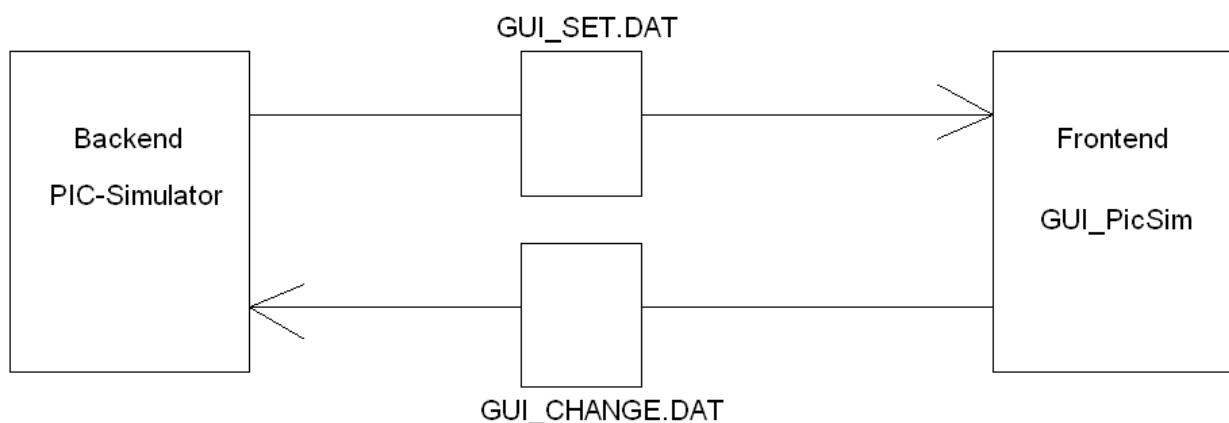


Abbildung 1: So erfolgt der Datenaustausch

Damit dieser Vorgang schnell und unter Umgehung der Festplatte erfolgen sollte, macht der Einsatz einer RAM-Disk Sinn. Wie eine solche RAM-Disk unter Windows bzw. Linux installiert wird, wird im Abschnitt 5.1 und 5.2 erläutert.

Eine Kommunikation über STDIN und STDOUT ist in Planung.  
~~Eine Kommunikation über das CORBA-Interface ist in Planung.~~

## Hinweis:

Der Einsatz von GUI\_PicSim ist nicht verpflichtet. Es ist jedem freigestellt seine eigene GUI zu programmieren, denn nur dann lässt sich die Oberfläche frei nach den eigenen Vorstellungen gestalten. Denkbar ist auch, diese vorgefertigte GUI für die ersten Schritte zu verwenden um dann später die eigene GUI zu realisieren.

**Bei der Projektabgabe werden zwei unterschiedliche Punktetabellen verwendet. Die eine für Projekte mit eigener GUI und eine zweite für die, die mit dieser GUI arbeiten.**

## 2 Installation von GUI\_PicSim

### 2.1 Unter Windows

Laden Sie zuerst von Moodle die Windows-Version herunter. Es ist keine besondere Installation der Software notwendig. Einfach die Dateien in ein mit Schreib- Lese- und Execute-Rechten versehenes Verzeichnis kopieren und die EXE-Datei starten.

### 2.2 Unter Linux

Laden Sie zuerst von Moodle die Linux-Version herunter. Für dieses Programm ist keine besondere Installation der Software notwendig. Einfach die Dateien in ein mit Schreib- Lese- und Execute-Rechten versehenes Verzeichnis kopieren. Danach die Datei ausführbar machen (Eigenschaften → Zugriffsrechte). Anschließend kann man das Programm starten. Gegebenenfalls eine Verknüpfung auf dem Desktop erstellen.

### 3 Die Oberfläche

Abbildung zeigt das Erscheinungsbild der GUI\_PicSim-Software.

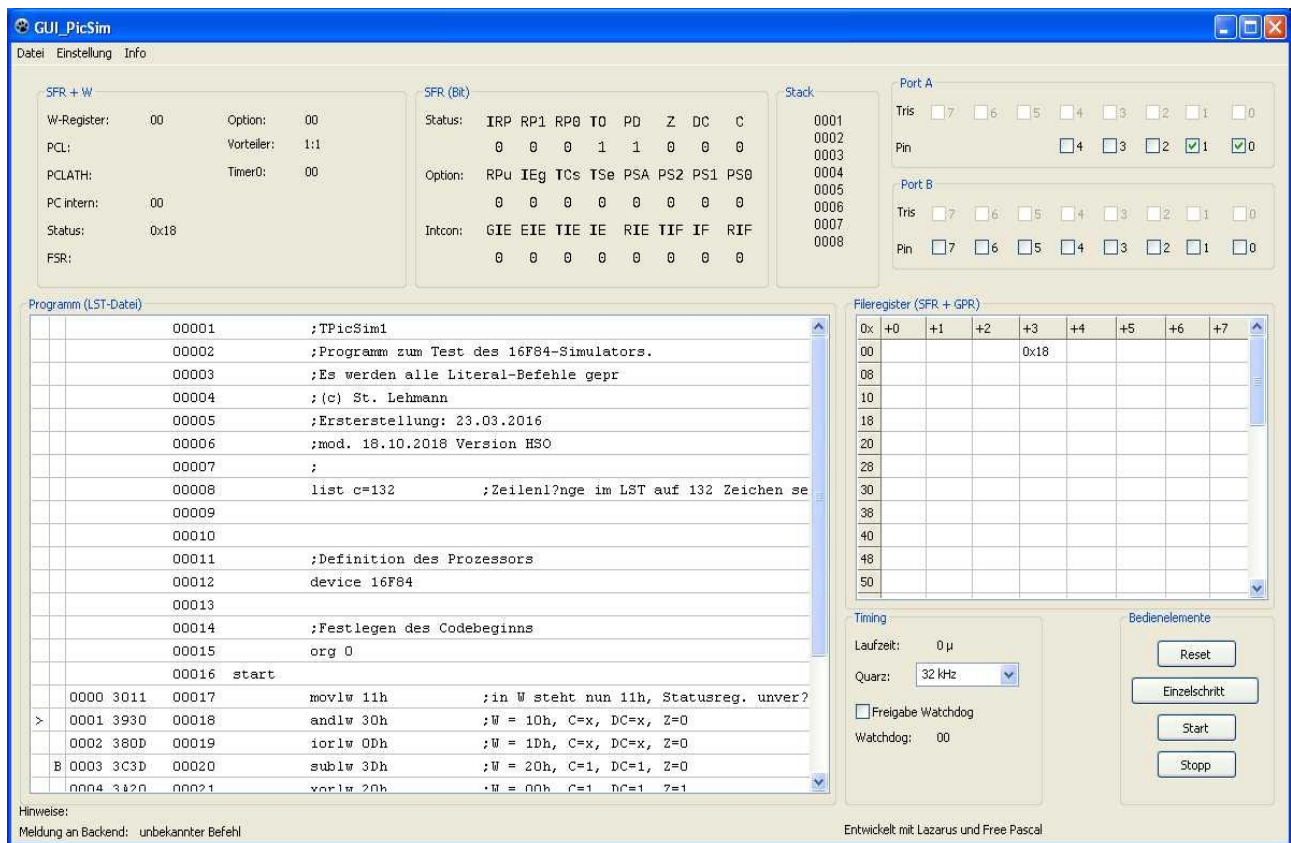


Abbildung 2: Oberfläche von GUI\_PicSim

#### 3.1 Fenster für LST-Datei

In diesem Fenster ist die LST-Datei des geladenen Testprogramms zu sehen. Diese Datei wird ohne Änderung in dieses Fenster geladen. Das Fenster besteht aus einem 3-spaltigen Grid. In der ersten Spalte ist die Markierung für die als nächste auszuführende Zeile. Diese Markierung muss über `SETLINE zlnr` und `RESLINE zlnr` vom Backend verwaltet werden. In der zweiten Spalte sind die Breakpoint-Zeilen markiert. Auch hier muss das Backend mit `SETBREAK zlnr` und `RESBREAK zlnr` die Verwaltung übernehmen. Die LST-Datei landet in der dritten (größten) Spalte.

Alle Zeilenangaben für die aktuelle Zeile bzw. die Breakpoint-Zeile beziehen sich auf die fortlaufende Nummerierung, die in Spalte 3 angezeigt wird.

#### 3.2 Fenster für File Register

In den File Register liegen neben den SFR (Special Function Registers) auch die GPR (General Purpose Registers). Sie haben aufgrund der Harvard-Architektur des PIC-Mikrocontrollers einen eigenen Adressraum, der in diesem Fall die Adressen 0x00 bis 0xFF umfasst. Allerdings sind nicht alle Adressen mit einem Speicherelement besetzt (siehe Datenblatt). Die Inhalte dieser Register können vom Backend mit dem Befehl `FREG adr, wert` an GUI\_PicSim gesendet werden. Die

Adresse darf dabei sowohl als Dezimalzahl als auch als Hexzahl (0x10) übergeben werden. Der Inhalt von Wert wird als String direkt in die entsprechend ausgewählte Adresse geschrieben. Somit bestimmt das Backend, ob hier ein Wert in dezimal oder hex erscheint.

### 3.3 Fenster für Timing

Das Feld Laufzeit lässt sich durch `TIME wert` beschreiben. Auch hier wird `wert` als Zeichenkette verarbeitet und ohne Änderung übernommen. Damit kann das Backend auch die Einheit (z.B. `µs` oder `ms`) mit angeben.

Weitere Felder die durch das Backend beschrieben werden können sind Watchdog (`WATCHDOG wert`), Vorteiler (`PRESCALER wert`) und Timer0 (`TIMER0 wert`).

Bei einer Änderung der Quarzfrequenz (`XTAL wert`), wird dies dem Backend mitgeteilt. Ebenso, ob der Watchdog (`WATCHDOG ON / WATCHDOG OFF`) aktiv sein soll.

### 3.4 Fenster SFR + W

Mit `WREG wert` wird das W-Register angesprochen. Das Gleich gilt für `PCL wert`, `PCLATH wert`, `PCINTERN wert`, `STATUS wert` und `FSR wert`.

### 3.5 Fenster SFR (Bit)

Hier werden die wichtigsten SF-Register zusätzlich als Bitmuster dargestellt. Angesprochen werden diese über `STATUSBIT x, y`. Mit `x` = Bitnummer von 0 bis 7 und `y` mit 0 oder 1. Dies gilt ebenso bei `OPTIONBIT x, y` und `INTCONBIT x, y`.

### 3.6 Fenster Stack

Hier kann der Stackinhalt, der von `CALL` und `RETURN` beeinflusst wird, angezeigt werden. Der Befehl dazu lautet: `STACK wert1, wert2, ... , wert8`. Es müssen nicht unbedingt alle 8 Parameter angegeben werden, die fehlenden Werte ergeben Leerzeilen.

### 3.7 Fenster Port A

Hier werden neben der TRIS-Information, die nicht durch Klicks veränderbar ist auch die Zustände der IO-Leitungen dargestellt. Eine markierte Checkbox bedeutet High-Pegel (= 1), eine nicht markierte Box bedeutet Low-Pegel (= 0). Die Checkboxen für die IO-Leitungen können angeklickt werden. Jeder Klick wird an das Backend mit den Befehlen `PORTA 0, 0` bis `PORTA 7, 1` gemeldet. Dabei bedeutet `PORTA 0, 0`, dass die Checkbox für Port A0 zurückgesetzt ist. Eine gesetzte Checkbox liefert das Ergebnis in Form von `PORTA 0, 1`.

Die Checkboxes des Port A können per Mausklick umgeschaltet werden. Bei jedem Klick wird dem Backend dieses Ereignis in Form von `PORTA x, y` gesendet.

## 3.8 Fenster Port B

Hier gilt das oben Beschriebene analog für Port B.

## 3.9 Fenster Bedienelemente

Über diese Buttons soll das Backend steuerbar sein. GUI\_PicSim liefert folgende Befehle an das Backend:

RESET

STEP (Einzelschritt)

RUN

STOPP

## 3.10 Menüpunkt Datei

### Datei laden

Ein Dateiauswahlmenü wird geöffnet. Hier mittels Maus das gewünschte Verzeichnis und die gewünschte Datei auswählen. Wird die Eingabe bestätigt, lädt die Software das Testprogramm in das LST-Fenster. Außerdem wird dem Backend der Dateiname incl. Verzeichnisangaben mit dem Befehl LOAD übergeben.

Das Backend muss nun ebenfalls diese Datei laden und entsprechend den Vorgaben verarbeiten.

### Einstellungen speichern

Hier werden die aktuellen Einstellung in einer INI-Datei (GUI\_PicSim.INI) gespeichert. Fehlt diese Datei wird das Programm mit Defaultwerten gestartet.

### Programm beenden

GUI\_PicSim wird beendet. Eine Speicherung der aktuellen Einstellungen erfolgt nicht!

## 3.11 Menüpunkt Einstellungen

Hier kann das Verzeichnis angegeben werden, das von den Kommunikationsdateien verwendet werden soll. Arbeiten Sie an dieser Stelle möglichst mit einer RAM-Disk.

## 3.12 Menüpunkt Hilfe

Öffnet eine Kurzanleitung.



## 4 Datenaustausch

Der Datenaustausch erfolgt über einfache Textdateien. Sie sind wie folgt aufgebaut:

BEFEHL *leerzeichen* PARAMETER (, PARAMETER)

Benötigt der Befehl zusätzliche Parameter, sind diese durch ein (einziges) Leerzeichen voneinander zu trennen. Wird eine Parameterliste benötigt, sind die einzelnen Parameter durch Kommata zu trennen.

Ist eine Änderung der Daten erfolgt, muss entschieden werden, ob diese Änderung dem anderen Programm mitgeteilt werden muss. Hat beispielsweise das Backend einen neuen Wert für ein File-Register errechnet, muss dies der GUI mitgeteilt werden. In diesem Fall prüft das Backend, ob die Datei GUI\_SET.DAT existiert. Falls ja, muss das Programm warten, bis die GUI (= Empfänger) diese Datei mit den alten Daten gelöscht hat. Nun erzeugt das Backend die Datei GUI\_SET.DAT neu und schreibt folgende Anweisung hinein:

FREG adr,wert

Das Backend darf mehrere Anweisungen gleichzeitig in diese Datei schreiben. Dabei wird jede dieser Anweisungen in eine eigene Zeile geschrieben. Damit kann z.B. auch die aktuelle Zeilenmarkierung entfernt und eine neue gesetzt werden.

Beispiel:

FREG adr,wert  
RESLINE alte\_zlnr  
SETLINE neue\_zlnr

Ist die Datei vom Empfänger gelesen und die Nachricht verarbeitet worden, löscht er Empfänger diese. Erst jetzt darf der Sender eine neue Datei erstellen.

### 4.1 Backend an Frontend

Das Backend erzeugt die Datei GUI\_SET.DAT. Es stehen folgende Befehle zur Verfügung:

FREG adr,wert	schreibt <i>wert</i> als String an die Adresse <i>adr</i> des RAM-Speichers. Die Adresse darf sowohl als Dezimalwert als auch Hexwert (z.B. 0x12) übergeben werden. <i>Wert</i> wird direkt übernommen. Auf diese Weise kann die Zahlendarstellung durch das Backend erfolgen.
SETLINE zlnr	Markiert die durch <i>zlnr</i> bestimmte Zeile im LST-File.
RESLINE zlnr	Hebt die Markierung an der durch <i>zlnr</i> bestimmten Zeile wieder auf.
SETBREAK zlnr	Setzt den Breakpoint an der durch <i>zlnr</i> bestimmten Zeile im LST-File.

RESBREAK zlnr	Hebt die Breakpoint-Markierung wieder auf.
TIME wert	Schreibt wert in das Feld des Laufzeitzählers.
RAMDIR wert	In wert steht das Verzeichnis in dem sich die RAM-Disk befindet. Falls dieser Wert nicht definiert wird, wird das Eigenverzeichnis des GUI_PicSim-Programm verwendet.
WReg wert	Schreibt wert im Fenster <b>SFR + W</b> in das W-Feld.
PCL wert	Schreibt wert im Fenster <b>SFR + W</b> in das PCL-Feld.
PCLATH wert	Schreibt wert im Fenster <b>SFR + W</b> in das PCLATH-Feld.
PCINTERN wert	Schreibt wert im Fenster <b>SFR + W</b> in das PCINTERN-Feld.
STATUS wert	Schreibt wert im Fenster <b>SFR + W</b> in das STATUS-Feld.
FSR wert	Schreibt wert im Fenster <b>SFR + W</b> in das FSR-Feld.
OPTION wert	Schreibt wert im Fenster <b>SFR + W</b> in das OPTION-Feld.
STATUSBIT x,y	Setzt das entsprechende Bit des Statusregisters im Fenster <b>SFR (Bit)</b> . X = Bitnummer, y = 0 oder 1
OPTIONBIT x,y	Setzt das entsprechende Bit des Optionregisters im Fenster <b>SFR (Bit)</b> . X = Bitnummer, y = 0 oder 1
INTCONBIT x,y	Setzt das entsprechende Bit des Intconregisters im Fenster <b>SFR (Bit)</b> . X = Bitnummer, y = 0 oder 1
TRISA x,y	Setzt Bit x im TRIS-Register A auf y (0 oder 1)
PORTA x,y	Setzt Bit x im PORT-Register A auf y (0 oder 1)
	Hinweis: Bit 5 bis 7 sind nicht nutzbar
TRISB x,y	Setzt Bit x im TRIS-Register B auf y (0 oder 1)
PORTB x,y	Setzt Bit x im PORT-Register B auf y (0 oder 1)
STACK wert1,...,wert8	Schreibt die Werte 1 bis 8 in das Stackfenster
TIMER0 wert	Zeigt Wert bei Timer0 im Feld TIMING an.
WATCHDOG wert	Zeigt die bereits abgelaufene Zeit des Watchdogs an
PRESCALER wert	Ermöglicht die Anzeige des Teilerfaktors

## 4.2 Frontend an Backend

Das Frontend erzeugt die Datei GUI\_CHANGE.DAT. Hier werden alle Klicks und Eingaben auf der GUI zum Backend weitergeleitet.

LOAD dateiname	Wird im Dateimenü eine LST-Datei für das Laden ausgewählt, wird diese sofort in das entsprechende Fenster geladen und mit diesem Befehl der Vorgang an das Backend gemeldet. Dort muss diese Datei erneut geladen und geparkt werden.
XTAL wert	Die Quarzfrequenz wurde verändert
RESET	Der Reset-Button wurde angeklickt

STEP	Der Button für den Einzelschritt wurde angeklickt
RUN	Der RUN-Button wurde angeklickt
STOP	Es gab einen Klick auf den Stop-Button
PORTA x, y	Der Haken in der Checkbox x von Port A wird auf y gsetzt
PORTB x, y	Der Haken in der Checkbox x von Port B wird auf y gsetzt
WATCHDOG ON	Watchdog wurde eingeschaltet
WATCHDOG OFF	Watchdog wurde ausgeschaltet
PRESCALER wert	Der aktuelle Stand des Vorteilers wird ins entsprechende Feld geschrieben.

Wichtig!

Achten Sie darauf, dass beim Programmstart ihres Backends die beiden Dateien GUI\_SET.DAT und GUI\_CHANGE.DAT gelöscht sind.

## 5 Anhang I

Hier folgt eine Beschreibung wie eine RAM-Disk installiert werden kann. Es ist nur ein Beispiel. Im Internet sind weitere Möglichkeiten für die Installation einer solchen RAM-Disk zu finden.

### 5.1 Installation einer RAM-Disk unter Windows

### 5.2 Installation einer RAM-Disk unter Linux

Das ramfs-Dateisystem unterscheidet sich vom tmpfs-Dateisystem vor allem dadurch, dass es keine Daten in den Swap auslagert, also ein reines RAM-Dateisystem ist. Die Vorgehensweise ist hier noch simpler:

```
sudo mkdir /media/ramdisk  
sudo mount -t ramfs ramfs /media/ramdisk
```

Damit erhält man eine RAM-Disk, die sich (wie tmpfs) dynamisch der benötigten Größe anpasst. Um die Partition beim Systemstart automatisch einzuhängen, öffnet man einen Editor mit Root-Rechten und fügt folgende Zeile in die [/etc/fstab](#) hinzu:

```
ramfs    /media/ramdisk    ramfs    defaults          0        0
```

#### **Achtung!**

Das ramfs-Dateisystem hat im Gegensatz zu tmpfs leider keine Mountoptionen und bietet somit auch keine Möglichkeit, die Größe zu limitieren. Das bedeutet, dass im Extremfall das System keinen freien Hauptspeicher mehr zur Verfügung hat und nur noch auf die Festplatte auslagern kann.

(Quelle: [https://wiki.ubuntuusers.de/RAM-Disk\\_erstellen/](https://wiki.ubuntuusers.de/RAM-Disk_erstellen/))