

---

---

# **IT-Projekt**

## **Automatisierung einer Waschmaschine**

Mai 2020

## Inhaltsverzeichnis

IT-Projekt.....	1
Automatisierung einer Waschmaschine.....	1
Inhaltsverzeichnis.....	2
Ziele.....	3
Simulation der Waschmaschine.....	4
Programm WashingMachine.....	5
Technologischen Größen.....	6
Betriebsmodi.....	7
Bedienung.....	8
Steuerung und Regelung der Waschmaschine.....	11
Verbindung von Steuerung und Anlage.....	12
Anschlussbelegung.....	14
Arbeitsweise der Steuerung und Regelung.....	15
Verhalten bei Fehlern.....	16
Datenübertragung.....	17
Tools.....	18
Windows Bedienprogramm.....	19
Dokumentation.....	20
Doxygen.....	20
SVN.....	20
Fritzing.....	20
Gnuplot.....	21

## **Ziele**

Eine haushaltsübliche Waschmaschine soll automatisiert werden. Hierzu gehören die erforderlichen Steuerungen und Regelungen zur Realisierung mehrerer Waschprogramme.

Da bei einer realen Waschmaschine die Waschvorgänge für ein IT-Projekt unbequem lange dauern und da man nasse Hände bekommen könnte, wird die Waschmaschine durch eine vorhandene Software simuliert. Diese Simulation kann zwischen Zeitraffer und normaler Geschwindigkeit umgeschaltet werden.

Zunächst sind die übliche Arbeitsweise von Waschmaschinen zu ermitteln. Daraus folgt eine Definition der Arbeitsschritte ein oder mehrerer Waschprogramme und es kann ein Pflichtenheft über die Funktionen der Steuerung und Regelung erstellt werden.

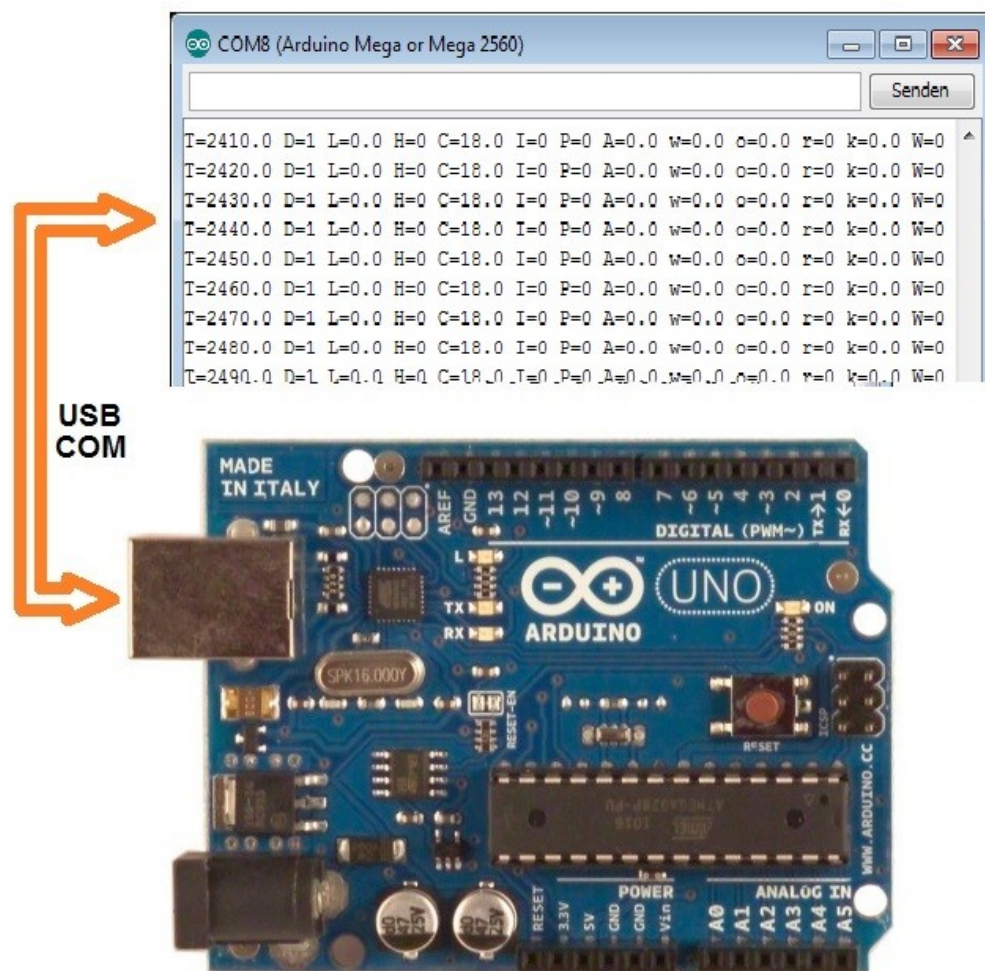
Soweit durch die bestehende Simulation abgedeckt, ist dann ein geeignetes Programm zur Steuerung der Waschmaschine in C/C++ zu erstellen.

Die erreichte Funktionalität ist zu dokumentieren. Ziel ist eine Beschreibung des Programms und seiner Funktionen. Hierzu sind auch Zeitverläufe von ausgewählten Waschvorgängen, Temperaturverläufen etc. aufzuzeichnen.

Optional kann eine Bedienung von einem graphischen PC-Programm erstellt werden.

## Simulation der Waschmaschine

Die vorbereitete Simulation der Waschmaschine ohne eigenständige Steuerung und Regelung befindet sich in dem Programm genannt WashingMachine auf einem Arduino Board. Über die USB-Kommunikationsschnittstelle des Boards wird eine serielle Kommunikation (COM-Schnittstelle) realisiert. Mit dieser Datenverbindung kann der aktuelle Zustand der Waschmaschine in einfacher Form angezeigt werden und es können über manuelle Steuerbefehle Eingriffe in die Waschmaschine erfolgen.



In der Grundeinstellung des Monitors sollte automatisches Scrolling eingeschaltet sein, die Baudrate auf 9600 stehen und kein Zeilenende hinzugefügt werden.

Statt des integrierten seriellen Monitors der Arduino IDE kann auch jedes andere COM-Port Terminal verwendet werden. Einige derartige Terminals erlauben die empfangenen Daten in eine Log-Datei zu kopieren (z.B. <https://sourceforge.net/projects/realterm/>). Diese Log-Dateien können später für weitere Aufgaben ausgewertet oder dokumentiert werden.

**Programm WashingMachine**

Im Programm WashingMachine wird eine haushaltsübliche Waschmaschine bis zu einer handhabbaren Detailtiefe simuliert. Dies ist nötig um einerseits die Aufgabenstellung nicht ausufern zu lassen und andererseits um Versuche in einem angemessenen Zeitrahmen zu ermöglichen.

Die Simulation arbeitet zudem auch in einem Zeitraffer-Modus. Wahlweise entspricht 1 Sekunde der realen Zeit entweder 1 Sekunde oder 20 Sekunden der simulierten Zeit, d.h. 1-faches bzw. 20-faches Zeitraffer. Mit der langsameren Varianten können Übergangsvorgänge mit guter Genauigkeit beobachtet werden und mit der schnelleren Variante ist eine Simulation eines kompletten Waschprogramms in noch akzeptabler Zeit möglich.

### Technologischen Größen

Folgende technologischen Größen werden von der Simulation bereitgestellt.

Größe	Sensor oder Aktuator	Bedeutung
Zeit	(S)	Simulationszeit in Sekunden
Zeitrafferfaktor	A	Verhältnis von Echtzeit zu Simulationszeit
Türsensor	S	Zustand der Tür, offen (0) oder geschlossen (1)
Wäschemenge	A	in kg, maximal 9 kg
Waschmittelzugabe	A	in g, max 50 g
Detergenzienmenge	(S)	(noch) vorhandene Waschmittel- oder Weichspülermenge
Wasserzulauf	A	Einlaufventil offen oder geschlossen möglichst max 20 kg Überlauf bei 80 kg
Wasserpumpe	A	Ein- oder Ausgeschaltet
Wasser-Maximum-Sensor	S	Sensor für erlaubten Höchststand
Heizung	A	Ein- oder Ausgeschaltet
Energieverbrauch	(S)	in kWh
Temperatur	S	in °C
Drehzahl	S	>0 rechts, <0 links

Diese Größen sind als Maschinendaten fest vorgegeben. In dieser Tabelle gelten alle Daten als Sensor, die ausgelesen werden können. Hierzu zählen auch einige reine Rechenwerte wie z.B. die Zeit. Entsprechend sind alle einstellbaren Daten als Aktuatoren aufgeführt.

Digitale Maschinendaten werden über Ein- und Ausgänge der Arduino Boards ausgetauscht. Für die anderen Daten wird eine I<sup>2</sup>C-Verbindung zwischen den Boards eingesetzt. Über diese Verbindung können auch analoge Daten übertragen werden.

## **Betriebsmodi**

Die Anlage kann manuell durch eine Kommunikation über die USB-Schnittstelle durch einfache Kommandos am PC eingestellt. Hiermit ist ein experimenteller Betrieb der Maschine möglich. Insbesondere bei ersten Versuchen zum Kennenlernen der Waschmaschine ist diese Einstellung sehr geeignet.

In der eigentlichen Betriebseinstellung folgen die Aktuatoren den Eingängen des Arduino-Boards bzw. den übertragenen Werten über den I<sup>2</sup>C-Bus. Diese sind mit den entsprechenden Ausgängen des Controllers als PLC verbunden.

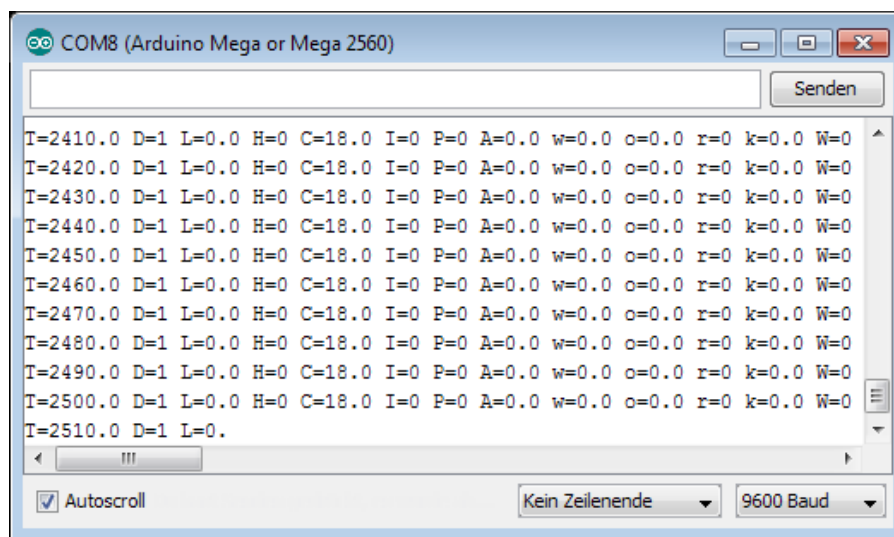
Die Sensoren werden in jedem Fall an die entsprechenden Ausgänge des Arduino-Boards geleitet und werden gemeinsam mit den Werten der Aktuatoren zusätzlich immer an der USB-Schnittstelle der Anlagen-Simulation ausgegeben.

Zu beachten ist, dass die jeweilig zusammengehörenden Ein- und Ausgänge immer miteinander verbunden sein sollten. Offene Eingänge von Arduino-Boards können zufällige oder falsche Werte liefern.

### Bedienung

Eine manuelle Bedienung der Waschmaschine kann mittels der serieller Kommunikation über die USB/COM-Schnittstelle erfolgen. Hiermit können Abläufe vorab getestet werden oder während eines automatisierten Ablaufs korrigierend verändert werden sofern die Automatisierung die manuellen Werte nicht wieder übersteuert.

Von der seriellen Kommunikation wird stetig oder als Antwort zu dem ?-Kommando eine durch Leerzeichen getrennte Folge von aktuellen Werten oder Zuständen gesendet. Die Kurzbezeichnungen bestehend aus einem Buchstaben und einem Gleichheitszeichen können mittels des Verbose Flags ein- oder ausgeschaltet werden.



Angezeigt werden in dieser Reihenfolge

Name	Bedeutung
<b>T</b>	Zeit in Minuten
<b>D</b>	Türzustand auf (0) oder zu (1)
<b>L</b>	Wäschemenge in kg
<b>H</b>	Heizung Ein/Aus
<b>C</b>	Temperatur in °C
<b>I</b>	Wassereinlaufventil Ein/Aus
<b>P</b>	Abwasserpumpe Ein/Aus
<b>A</b>	Wassermenge in kg
<b>w</b>	Wassermenge in der Wäsche in kg
<b>o</b>	enthaltene Waschmittelmenge in g
<b>r</b>	Drehzahl in rpm
<b>k</b>	Heizenergieverbrauch in kWh
<b>W</b>	Warnung oder Fehler



Die Bedienung, von Hand oder programmgesteuert, erfolgt durch parametrisierte, einbuchstabige Kommandos, die an die Kommunikation über COM/USB oder I<sup>2</sup>C übergeben werden und die zugehörigen Werte als Antwort zurückliefern.

<b>Kommando</b>	<b>Bedeutung</b>
<b>t=x</b>	Zeitraffer-Modus, schnell/langsam = 1/0
<b>D=x</b>	Tür öffnen/schließen = 0/1
<b>D?</b>	Abfrage der Tür, geschlossen = 1
<b>L=n</b>	Wäschemenge 0..9 kg (ganzzahlig)
<b>L?</b>	Abfrage Wäschemenge
<b>I=x</b>	Wassereinlaufventil Auf/Zu = 1/0
<b>I?</b>	Abfrage Wassereinlaufventil
<b>H=x</b>	Heizung Ein/Aus = 1/0
<b>H?</b>	Heizung abfragen
<b>P=x</b>	Abwasserpumpe Ein/Aus = 1/0
<b>P?</b>	Abfrage Abwasserpumpe
<b>O=n</b>	Waschmittelzugabe in g (ganzzahlig), additiv
<b>o=n</b>	Weichspülerzugabe in g (ganzzahlig), additiv
<b>o?</b>	Abfrage der Waschmittel- oder Weichspülermenge
<b>r=n</b>	Trommeldrehzahl 0 bis +/- 1500 rpm (ganzzahlig) <110 solange Wasser (>=1kg) in der Maschine
<b>r?</b>	Trommeldrehzahl ermitteln
<b>T?</b>	Simulationszeit erfragen
<b>t?</b>	Zeitraffer-Modus ermitteln
<b>C?</b>	Abfrage der Temperatur
<b>A?</b>	Abfrage Wassermenge
<b>w?</b>	Wassermenge in der Wäsche
<b>k?</b>	verbrauchte Heizenergie
<b>W=0</b>	Fehler oder Warnung zurücksetzen
<b>W?</b>	Fehler oder Warnung auslesen
<b>V=x</b>	Verbose Flag ein/aus
<b>R</b>	(Neu-)Initialisierung (Reset)

Bei den Einstell-Kommandos ist das zweite Zeichen ein Gleichheitszeichen ('='). Als Antwort erhält man den jeweiligen neuen Zustand. In der Regel wird also das Einstell-Kommando wieder zurückgegeben.

Beispiel	Kommando	Antwort	
	H=1	H=1	Heizung wird eingeschaltet

---

## IT-Projekt Waschmaschine

---

Bei den Abfrage-Kommandos ist das zweite Zeichen ein Fragezeichen ('?'). Als Antwort erhält man den jeweiligen aktuellen Zustand.

Beispiel	Kommando	Antwort
	I?	I=1      Wassereinlaufventil ist geöffnet

Für die Steuerung und Regelung sollten einige der Messwerte tunlichst nicht verwendet werden, weil diese für die Entwicklung zwar praktisch aber in einer realen Waschmaschine nicht vorhanden sind. Beispielsweise ist eine Messung der Restmenge des Waschmittels in der Wäsche nicht möglich und eine Messung der Restfeuchte (Wassermenge in der Wäsche) allenfalls bei Trocknern vorhanden (indirekt über Abluftfeuchtigkeit).

Mit dem Kommando R wird die Anlage in einen definierten Anfangszustand versetzt. Dies ist beispielsweise nach Störungen nötig. Allerdings werden die Einstellungen für Zeitraffer und Verbose unverändert beibehalten.

Bei fehlerhafter Steuerung werden Warnungen in Form von Bits innerhalb einer Zahl erzeugt. Die einzelnen Bits haben folgende Bedeutungen.

Bit (hexadezimal)	Bedeutung
0	keine Fehler
0x0001	door open
0x0002	door closed
0x0004	still heating
0x0008	too hot
0x0010	drum filled with water
0x0020	rotation not zero
0x0040	rotation too fast
0x0080	water overflows
0x0100	water is boiling
0x0200	too much water
0x0400	too much laundry
0x0800	too much detergent

Siehe auch Helpfile WashingMachine.CHM.

## **Steuerung und Regelung der Waschmaschine**

Für Steuerungs- und Regelungsaufgaben verwendet man bei einfachen Waschmaschinen typischerweise Schrittschaltwerke. Neuere und aufwendiger gebaute Geräte arbeiten mit Embedded Controllern.

Hier ist für Steuerung und Regelung der Waschmaschine als Starthilfe für den Entwurf ein Programm Controller für ein zweites Arduino UNO Board vorbereitet.

Das Programm Controller besitzt bereits eine rudimentäre Bedienung über die USB/COM Schnittstelle mit den gleichen Kommandos wie die Simulation der Anlage. Diese kann als Basis für den Datenaustausch von Steuerung und Waschmaschine dienen.

Als eine kleine Hilfe sind bereits Abfragen der Zeit (als Sensor), der Temperatur des Wassers und des Füllstandes in der Trommel programmiert.

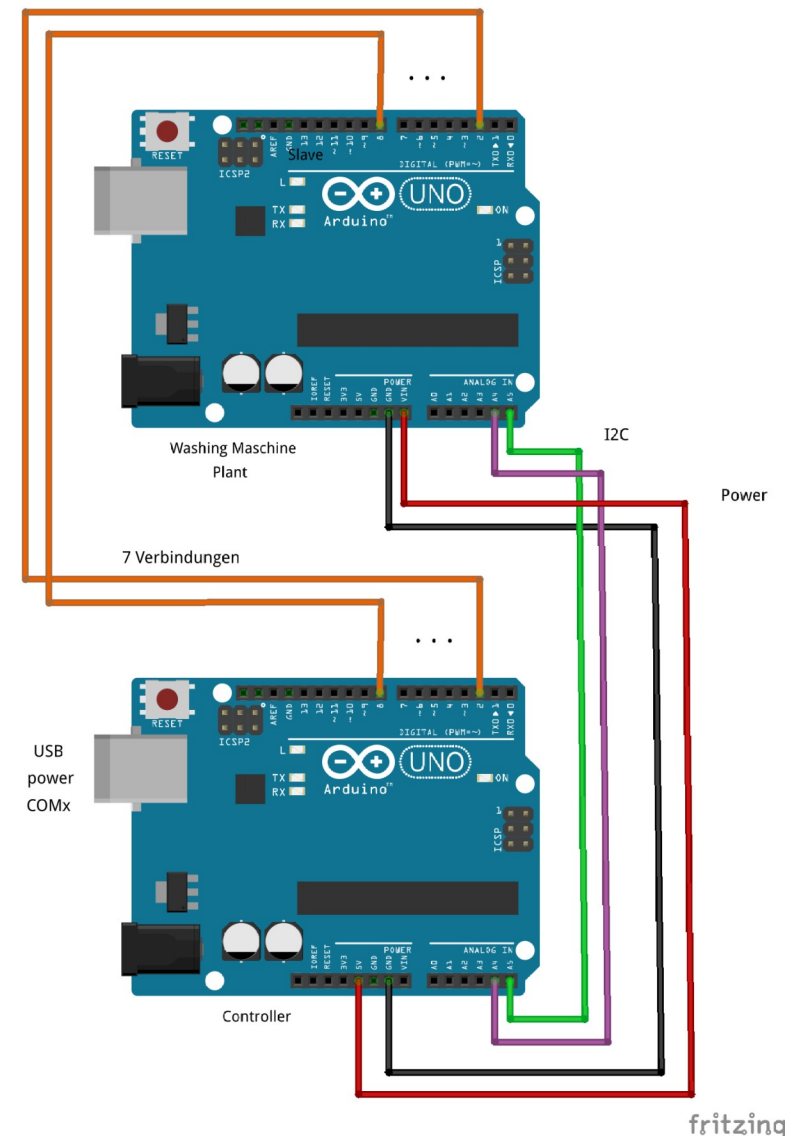
In einem ersten Schritt sollte die Funktion der Waschmaschine bzw. deren Simulation von Hand ausprobiert werden. Also Wäsche zuführen, Türe schließen, Wasser einlaufen lassen, evtl. Waschmittelzugabe, Aufheizen, Trommelbewegung, Wasser abpumpen, Türe öffnen. Man erhält hierbei etwas mehr "Gefühl" für die Funktionsweise und Fehlerquellen.

Wichtig ist auch eine Recherche, welche Funktionen oder Programme eine moderne Waschmaschine besitzt und welche davon hier realisiert werden können bzw. sollen.

Als ersten Versuch eines Steuerprogramms kann ein abwechselndes Füllen und Leeren der Waschmaschine dienen. Hierauf kann danach eine schrittweise Ergänzung des Steuerprogramms folgen. Entsprechende Vorgänge sind dazu durch das Programm zu automatisieren. Als "Bedienung" kämen dann nur noch die Türe, die Wäschemenge, evtl. das Waschprogramm und ein Einschaltknopf in Betracht. Diese Bedienung kann beispielsweise direkt durch manuelle Eingaben über die USB/COM Schnittstelle realisiert werden. Der übrige Vorgang sollte dann automatisch ablaufen.

### Verbindung von Steuerung und Anlage

Als Verbindung zwischen der Steuerung und den Bauteilen der Waschmaschine ist eine Kommunikation über I<sup>2</sup>C vorbereitet. Dies übernimmt die Rolle eines sonst üblichen Feldbusses oder einer direkten Verdrahtung.



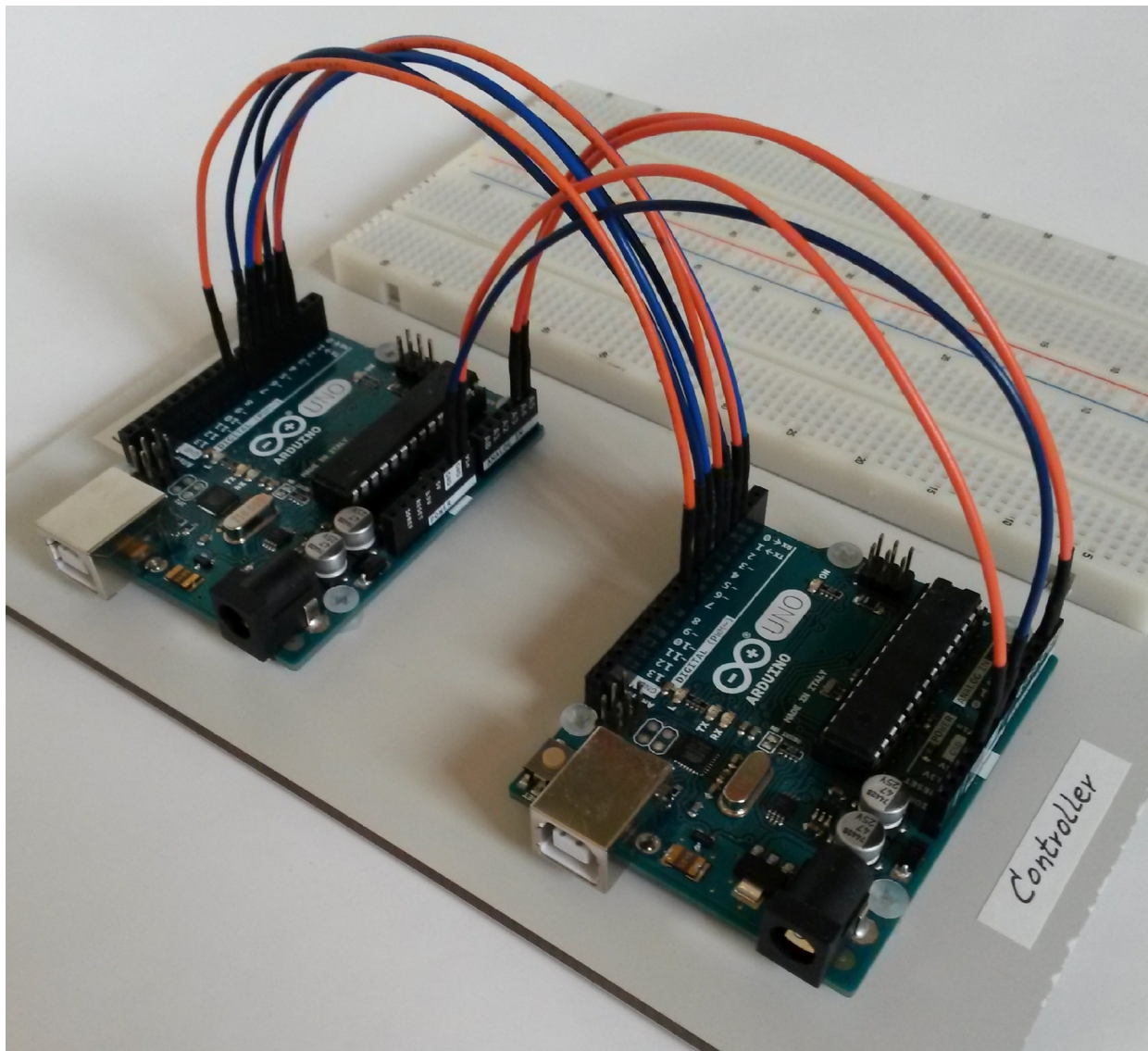
Zwei Verbindungen führen Masse und +5 Volt als Spannungsversorgung. So geschaltet, ist bereits eine Stromversorgung über ein USB-Kabel ausreichend. Es dürfen aber auch beide USB-Anschlüsse von zwei PCs aus genutzt werden. Dies ist hilfreich, wenn die Signale und Daten auf der Seite der Waschmaschine aufgezeichnet oder visualisiert werden sollen.

Die Kommunikation benutzt die Master-Slave- bzw. Client-Server-Architektur des I<sup>2</sup>C-Bus. Die Anlage bzw. deren Simulation übernimmt dabei die Rolle eines Slave bzw. Servers. Ein Slave wartet auf Anfragen und beantwortet diese. Er wird niemals selbst aktiv. Der Master bzw. Client ist

der aktive Teil, also hier der Controller. Er entscheidet wann, wen und was angefragt wird. Ausgetauscht werden hier die gleichen kurzen Kommandos in Form von (lesbarem) Text wie an der USB/COM-Schnittstelle.

Der I<sup>2</sup>C-Bus zwischen den beiden Boards wird hier ohne die eigentlich erforderlichen Abschlußwiderstände und ohne eine aufwendige Störfehlerbehandlung betrieben. Es kann daher gelegentlich zu Abbrüchen der I<sup>2</sup>C-Verbindung kommen. Diese sind oft nur durch einen Reset der Arduinos zu beseitigen.

Die übrigen 7 Leitungen (und die Masse als Rückleitung) verbinden die Steuerung auf einem Board mit der Simulation auf dem anderen Board.



**Anschlussbelegung**

<b>Anschluß</b>	<b>Bedeutung</b>
<b>I<sup>2</sup>C</b>	Zeit in Sekunden, Warnung oder Fehler etc.
<b>D2</b>	Wasserzufuhrventil
<b>D3</b>	Wasserpumpe
<b>D4</b>	Heizung
<b>D8</b>	Türsensor
<b>I<sup>2</sup>C</b>	Temperatur
<b>I<sup>2</sup>C</b>	Trommeldrehzahl
<b>I<sup>2</sup>C</b>	Waschmittelzugabe
<b>I<sup>2</sup>C</b>	Wasserstand

Nicht alle vorhandenen Daten sind für die Steuerung zugänglich. Einige Daten liefert die Simulation lediglich über die USB/COM-Schnittstelle zum Zwecke der Dokumentation, z.B. die verbrauchte Heizenergie.

Die standardmäßig definierte LED auf dem Board am Anschluss D13 wird als blinkendes Arbeitskennzeichen verwendet.

## **Arbeitsweise der Steuerung und Regelung**

Für Steuerungs- und Regelungsaufgaben verwendet man typischerweise SPSen oder Embedded Controller. Da der Prozess hier langsam und gutmütig ist, kann er problemlos von einem Arduino Board gesteuert und geregelt werden.

Die reale Abtastzeit im Programm Controller wird durch einen stetigen Prozess und abhängig von der aktuellen Zeit (im systemweiten Zeittaktzähler) realisiert. Dies entspricht in etwa der Arbeitsweise von SPSen.

Der größere Teil der Steuerung kann mit einer Abtastzeit von 100 msec (Realzeit) erfolgen. Soweit möglich dürfen weniger zeitkritische Steuerungsvorgänge nur jede Sekunde bearbeitet werden. Nur sehr wenige Teile der Automatisierung benötigen eine Abtastzeit von 10 msec (Realzeit).

Es ist besonders wichtig, dass keine Teilaufgabe eine länger Rechenzeit als ihre Abtastzeit benötigt. Insbesondere darf an keiner Stelle der Steuerung auf eine Eingabe des Benutzers oder eine Antwort der Kommunikation gewartet werden.

Die Verbindung von Steuerung und Prozess erfolgt über die Ein- und Ausgänge der Arduino-Boards und durch eine Kommunikation (I<sup>2</sup>C) ähnlich einer Verbindung über einen Feldbus oder ein Netzwerk.

Eine kontinuierliche Einstellung von Drehzahl und Drehrichtung der Trommel erfolgt über I<sup>2</sup>C. Die Simulation ändert die Drehzahl jedoch nicht sofort, sondern beachtet ein langsames Anlaufen und Auslaufen. Die Umkehr der Drehrichtung darf nur bei Stillstand der Trommel erfolgen.

Drei Leitungen führen die Signale zur Steuerung des Ventils für den Wasser-Zulauf, für den Betrieb der Pumpe und für die Heizung.

Eine letzte Leitung liefert das Sensorsignal der Tür.

Einlegen der Wäsche und Schließen der Tür erfolgt nicht über die Steuerung sondern ausschließlich von außen. Hier also von der Simulation der Waschmaschinen aus oder wahlweise auch durch Eingabe am Controller oder von einem Bedienprogramm aus.

Es ist sinnvoll Daten aus der Steuerung in einer Datei abzulegen. Hier können solche Dateien zur Darstellung von Zeitverläufen mit Gnuplot verwendet werden.

Siehe auch Helpfile Controller.CHM.

### **Verhalten bei Fehlern**

Die Simulation der Waschmaschine berücksichtigt eine Reihe von Situationen als Fehler und erzeugt dafür die oben genannten Warnungen. So kann bei geschlossener Tür keine Wäsche eingelegt werden, bei drehender Trommel, zu hoher Temperatur (45°C) oder zu hohem Wasserstand (3 kg) die Tür nicht geöffnet werden.

In dem zu erstellenden Programm für die Steuerung und Regelung müssen diese Situationen durch das Programm behandelt oder gegebenenfalls verhindert werden.



### Datenübertragung

Alle binären Daten von der Steuerung im Controller zu Heizung in der Simulation des Hauses und umgekehrt werden über die Ein- und Ausgänge der Arduino-Boards abgewickelt.

Beispielsweise wird mittels

```
digitalWrite(nWaterIntake, 1); // open water valve
```

das Zulaufventil für Wasser geöffnet. Und mit

```
bDoorOpen = digitalRead(nDoorOpen); // get door open state
```

kann die Stellung der Tür abgefragt werden.

Für alle nicht binären Daten müssen Kommandos über den I<sup>2</sup>C-Bus gesendet und die Antworten ausgewertet werden. Hier sind in Controller.ino bereits Beispiele vorhanden, z.B.

```
case 0: // request time
  strcpy(szCommand, "T?"); // build command
  ++nIndex; // goto next nIndex
  break;
```

Hier sind für alle erforderlichen Daten ergänzende Kommandos nötig. Die Antworten werden etwas weiter im Programm angenommen, z.B.

```
case 'T': // got a fresh dTime value
  dTime = atof(szResponse+2); // convert response to double
  break;
```

Ähnlich können andere Werte in passenden Variablen gespeichert werden und stehen dann für die eigentliche Steuerung und Regelung zur Verfügung.

Für die Übertragung von Stellwerten genügt es Kommandos mit den Werten aus passenden Variablen regelmäßig zu senden. Die Antworten können dann ignoriert werden oder die Ausgaben mit Serial.print im default-Zweig werden auskommentiert.

## **Tools**

Für die Bearbeitung der Aufgabenstellung sind schon ein paar Hilfen außer dem Steuerprogramm vorbereitet.

Vgl. Beispiele in Verzeichnissen auf dem Server.

## Windows Bedienprogramm

Ein Programm zur Bedienung wäre ein nice to have. Eine denkbare Form, etwa mit wxWidgets programmiert, wäre folgende



Zusätzlich könnte auch eine Anzeige des aktuellen Zustands der Waschmaschine erfolgen.

Prinzipiell entspräche ein solches Programm einem HMI-Interface, auch SCADA genannt. Es könnte über eine Vielzahl weiterer Funktionen verfügen – evtl. graphische Anzeige als y-t-Diagramm und mehr. Der Aufwand ist aber erheblich.

Alternativ wäre auch eine Smartphone App (z.B. mit MIT App Inventor) oder eine Anzeige auf einem kleinen Display denkbar.

## **Dokumentation**

Ein wesentlicher Teil des IT-Projektes ist eine Dokumentation. Diese sollte aus einer funktionalen Beschreibung der Fertigung und der erstellten Software für die Steuerung bestehen.

Die Form der Dokumentation ist freigestellt. Typisch und häufig verwendet wird hierzu Word. Es ist aber auch möglich (und vielleicht sogar handlicher und leistungsfähiger) die Dokumentation direkt im Quellcode einzupflegen und mit Doxygen zu erstellen. Die Dokumentation der GUI-Bibliothek wxWidgets (einige tausend Seiten) ist beispielsweise hiermit erstellt.

### **Doxygen**

Doxygen ist eine weitverbreitete Software zur Dokumentation von Software innerhalb ihres Quellcodes. Gesteuert von speziellen Markierungen in den Quelldateien kann sie Dokumentationen in verschiedenen Formaten erstellen. Hier im IT-Projekt ist eine Dokumentation in Form von HTML-Seiten und kompiliertem HTML etwas vorbereitet. Die HTML-Seiten werden mit einem HTML-Compiler in CHM-Dateien umgestzt und komprimiert.

Siehe auch Controller.doxy

### **SVN**

Software soll meist über den Tag hinaus bestehen bleiben und muss reproduzierbar sein. Da Software meist weiterentwickelt wird, sind zur Fehlersuche häufig auch die vorangegangenen Versionen wichtig. Als erstes Backup und zur Versionsverwaltung werden verschiedene Software eingesetzt.

Eine solche, weit verbreitete und moderne Versionsverwaltung ist SVN (Apache Subversion). Die Software des IT-Projektes soll auf einem SVN-Server hinterlegt werden.

### **Fritzing**

Fritzing ist eine Entwurfs-Software für Arduino und andere derartige Boards. Sie kann beispielsweise zur Darstellung von Steckplatine und Schaltplan verwendet werden.

Eine Alternative aber deutlich komplexer ist KiCAD.

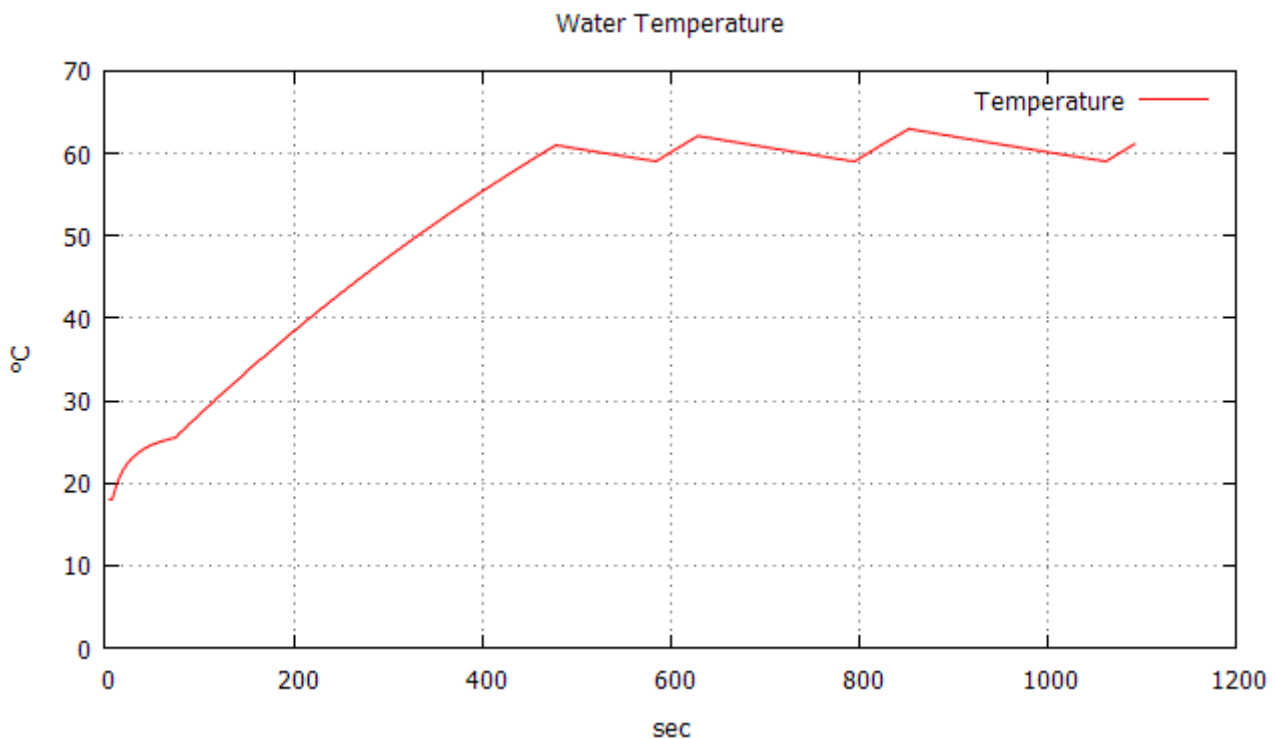
## Gnuplot

Gnuplot ist eine viel verwendete, freie Software zur Anzeige von datenbasierten Graphiken. Mit ihr kann u.a. das zeitliche Verhalten der Anlage in Form von y-t-Diagrammen dargestellt werden.

Im einfachsten Fall wird am USB/COM-Anschluß des Arduinos mit der Simulation der Anlage ein Speicherndes Terminal angeschlossen und die Ausgaben bzw. Messdaten in eine Textdatei protokolliert.

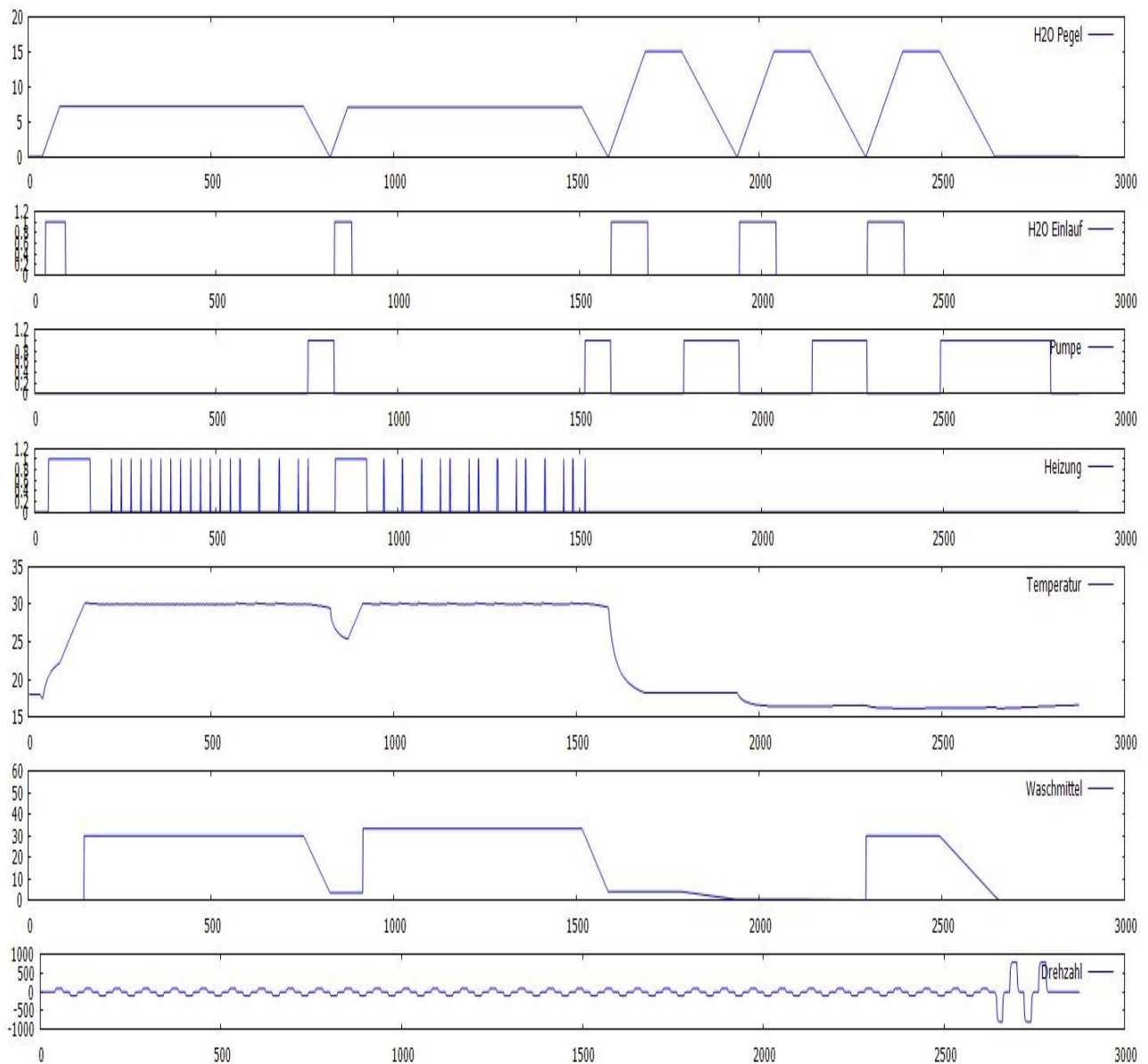
Erforderliche Eingaben im Gnuplot Kommando-Fenster wenn die Messdaten als Text in der Datei C:\it-project\WM\w.data vorliegen.

```
cd 'C:\it-project\WM'  
set yrange [0:30]  
set title("Water Temperature");  
set xlabel("sec");  
set ylabel("'°C");  
set grid;  
plot "w.data" using 1:5 with lines t "Temperature"
```



Diese Darstellung zeigt den Verlauf des ersten Aufheizvorgangs bis zum Beginn der Temperaturregelung.

Gnuplot erlaubt auch die Verwendung mehrerer Ausgabefenster, mehrerer Graphen in einem Fenster oder auch die Zusammenfassung mehrerer Graphen in ein Plot. Es können auch Skripte (vgl. w.plt) mit einer Liste von Gnuplot-Anweisungen verwendet werden.



Auch eine kontinuierliche Darstellung während des Betriebes ist möglich (vgl. wc.plt).