



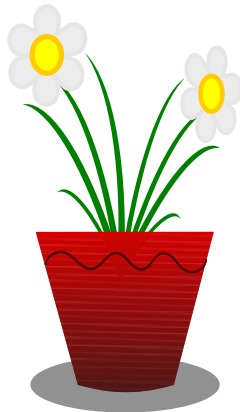
BALTHASAR NEUMANN TECHNIKUM TRIER

Modul 10

Herr Fink

Projektarbeit

Vollautomatisierte Pflanzenbewässerungsanlage



vorgelegt von:

| | |
|---------------|-----------------------------|
| Name: | Michael Büsch |
| Geburtsdatum: | 07.12.1984 |
| Adresse: | Güterstraße 49, 54295 Trier |
| E-Mail: | m@bues.ch |

Trier, den 3. Januar 2014

Inhalt

| | |
|---|-----------|
| 1 PROJEKT BESCHREIBUNG..... | 1 |
| 1.1 AUFGABENSTELLUNG..... | 1 |
| 1.2 BESCHREIBUNG DER FUNKTION..... | 1 |
| 2 AUFBAU..... | 2 |
| 2.1 MECHANISCHER AUFBAU..... | 2 |
| 2.2 AUFBAU DER STEUERUNG..... | 4 |
| 2.2.1 Blockschaltbild der Steuerung..... | 4 |
| 2.2.2 Der Regelkreis..... | 5 |
| 2.2.3 Datenblätter der Hardwarekomponenten..... | 6 |
| 2.3 SOFTWARE..... | 7 |
| 2.3.1 Quelltextverzeichnisse..... | 7 |
| 2.3.2 Softwareabhängigkeiten des PC-Programmes..... | 7 |
| 2.3.3 Parametrierung der Anlage über das PC-Programm..... | 8 |
| 2.4 AVR ATMEGA FIRMWARE..... | 10 |
| 2.4.1 Telegrammformat der seriellen Datenübertragung..... | 10 |
| 2.4.1.1 Logbucheintrag..... | 11 |
| 2.4.1.2 Real-Time-Clock-Daten..... | 12 |
| 2.4.1.3 Globale Konfiguration..... | 13 |
| 2.4.1.4 Pflanzentopfkonfiguration..... | 14 |
| 2.4.1.5 Pflanzentopfstatus..... | 15 |
| 2.4.1.6 Handbetrieb..... | 16 |
| 2.4.2 Die zentralen Datenstrukturen und Datentypen..... | 17 |
| 2.4.2.1 Konfiguration..... | 17 |
| 2.4.2.2 Zustand der Sensormessungen..... | 20 |
| 2.4.2.3 Zustand des Reglers..... | 21 |
| 2.4.3 Zustandsdiagramm des Reglers..... | 22 |
| 3 PERSÖNLICHE REFLEXION..... | 23 |
| 4 ANHANG..... | I |
| 5 QUELLEN..... | II |

Verzeichnis der Abbildungen

| | |
|---|----|
| Abb. 1: Mechanischer Aufbau..... | 2 |
| Abb. 2: Ein Pflanzentopf..... | 3 |
| Abb. 3: Sensorplatten..... | 3 |
| Abb. 4: Blockschaltbild der Steuerung..... | 4 |
| Abb. 5: Blockschaltbild des Regelkreises..... | 5 |
| Abb. 6: Hauptbild..... | 8 |
| Abb. 7: Hauptbild (Erweiterte Einstellungen)..... | 8 |
| Abb. 8: Zustandsdiagramm (Regler)..... | 22 |

Verzeichnis der Tabellen

| | |
|---|----|
| Tabelle 1: Datenblätter der Hardwarekomponenten..... | 6 |
| Tabelle 2: Quelltextverzeichnisse..... | 7 |
| Tabelle 3: Softwareabhängigkeiten..... | 7 |
| Tabelle 4: Beschreibung der Bedienelemente..... | 9 |
| Tabelle 5: Telegrammformat: Logbucheintrag (Sensordaten)..... | 11 |
| Tabelle 6: Telegrammformat: RTC-Daten..... | 12 |
| Tabelle 7: Telegrammformat: Globale Konfiguration..... | 13 |
| Tabelle 8: Telegrammformat: Pflanzentopfkonfiguration..... | 14 |
| Tabelle 9: Telegrammformat: Pflanzentopfstatus..... | 15 |
| Tabelle 10: Telegrammformat: Handbetrieb..... | 16 |
| Tabelle 11: Datenstruktur: Konfiguration (Topf)..... | 18 |
| Tabelle 12: Datenstruktur: Konfiguration (Global)..... | 19 |
| Tabelle 13: Datenstruktur: Sensor..... | 20 |
| Tabelle 14: Datenstruktur: Reglerzustand..... | 21 |

Verzeichnis des Anhangs

| | |
|-------------------------|---|
| Schaltpläne..... | I |
| Programmablaufplan..... | I |
| Lizenzvereinbarung..... | I |

1 Projektbeschreibung

1.1 Aufgabenstellung

Es soll eine vollautomatische Bewässerungseinrichtung für Zimmerpflanzen entwickelt werden, die eine Versorgung der Pflanzen mit Wasser über mehrere Wochen autark sicherstellen kann.

Auch unter sich ändernden Umgebungsbedingungen, wie z. B. Sonneneinstrahlung oder Luftfeuchtigkeitsänderungen, soll die Erde mit der passenden Menge an Wasser versorgt werden, um die parametrisierte Bodenfeuchtigkeit zu erhalten.

Die Anlage soll einen geschlossenen Regelkreis darstellen, damit Störgrößen erkannt und automatische Gegenmaßnahmen getroffen werden können.

Die Parametrierung und Diagnose soll von einem PC aus per serieller Schnittstelle erfolgen.

1.2 Beschreibung der Funktion

Zur Ermittlung der Rückführgröße des Regelkreises kommt eine Widerstandsmessung der Pflanzenerde zum Einsatz. Sinkt der Leitwert der Erde unter einen individuell festgelegten Wert, öffnet ein elektronischer Zweipunktregler das entsprechende Sitzventil und lässt eine vorgegebene Menge Wasser in der Erde versickern. Nach einer kurzen Verweilzeit mit geschlossenem Ventil, wird erneut eine Widerstandsmessung durchgeführt.

Beeinflusst wird die Durchführung einer Messung durch die aktuelle Tageszeit und den Wochentag. So kann z. B. erzwungen werden, dass nachts oder an bestimmten Wochentagen keine Bewässerungsvorgänge durchgeführt werden.

Die Anlage bietet die Möglichkeit, durchgeführte Bewässerungen detailreich in einem Logbuch zu erfassen und später auszuwerten.

Außerdem ist eine Handbetriebsart vorgesehen, die die manuelle Steuerung und Beeinflussung der Ventilfunktion und des Regelvorganges ermöglicht.

2 Aufbau

2.1 Mechanischer Aufbau

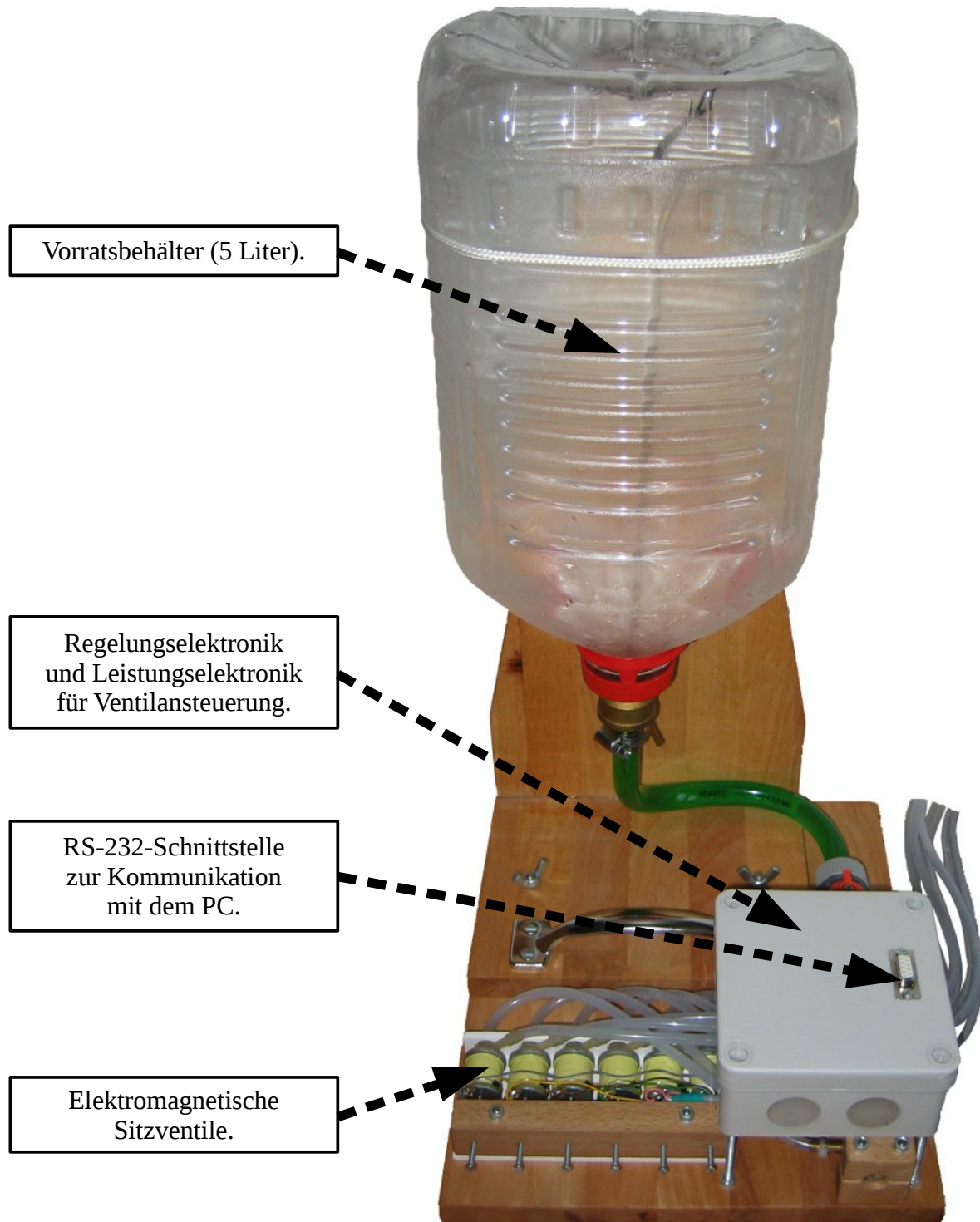


Abb. 1: Mechanischer Aufbau

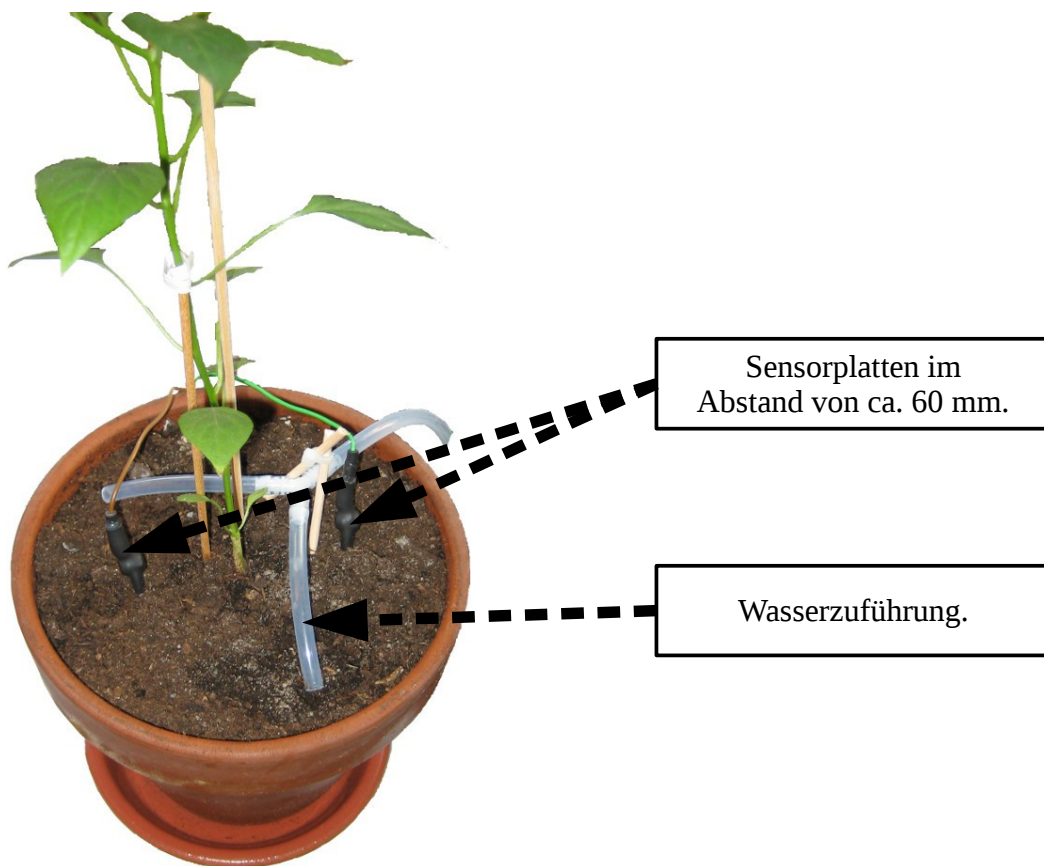


Abb. 2: Ein Pflanzentopf

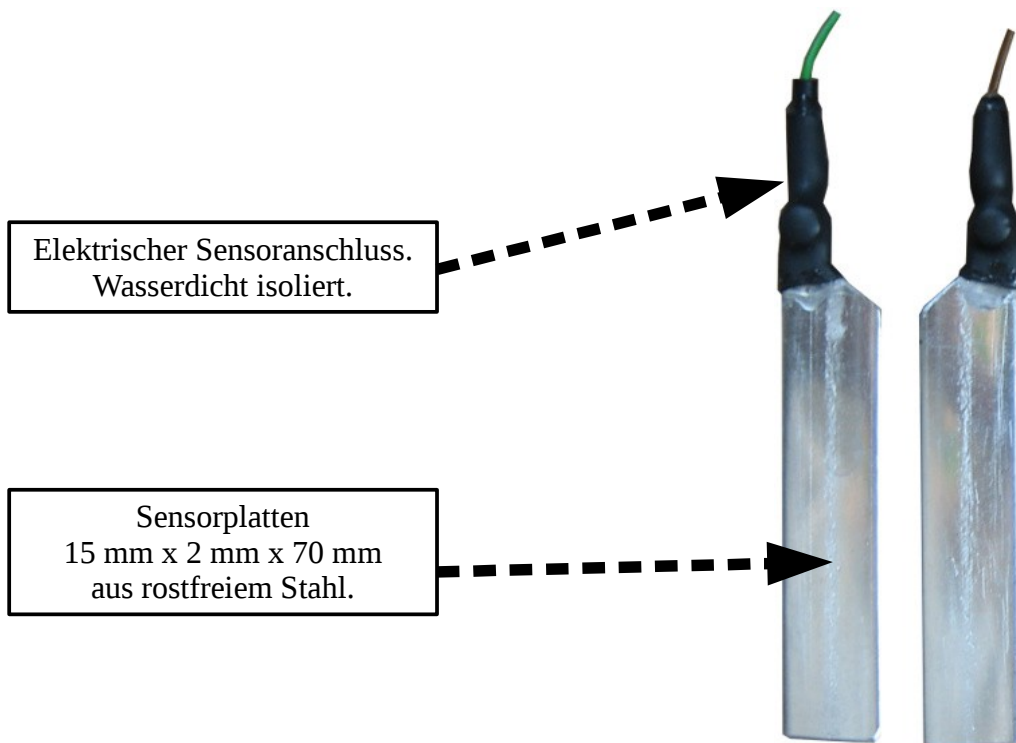


Abb. 3: Sensorplatten

2.2 Aufbau der Steuerung

2.2.1 Blockschaltbild der Steuerung

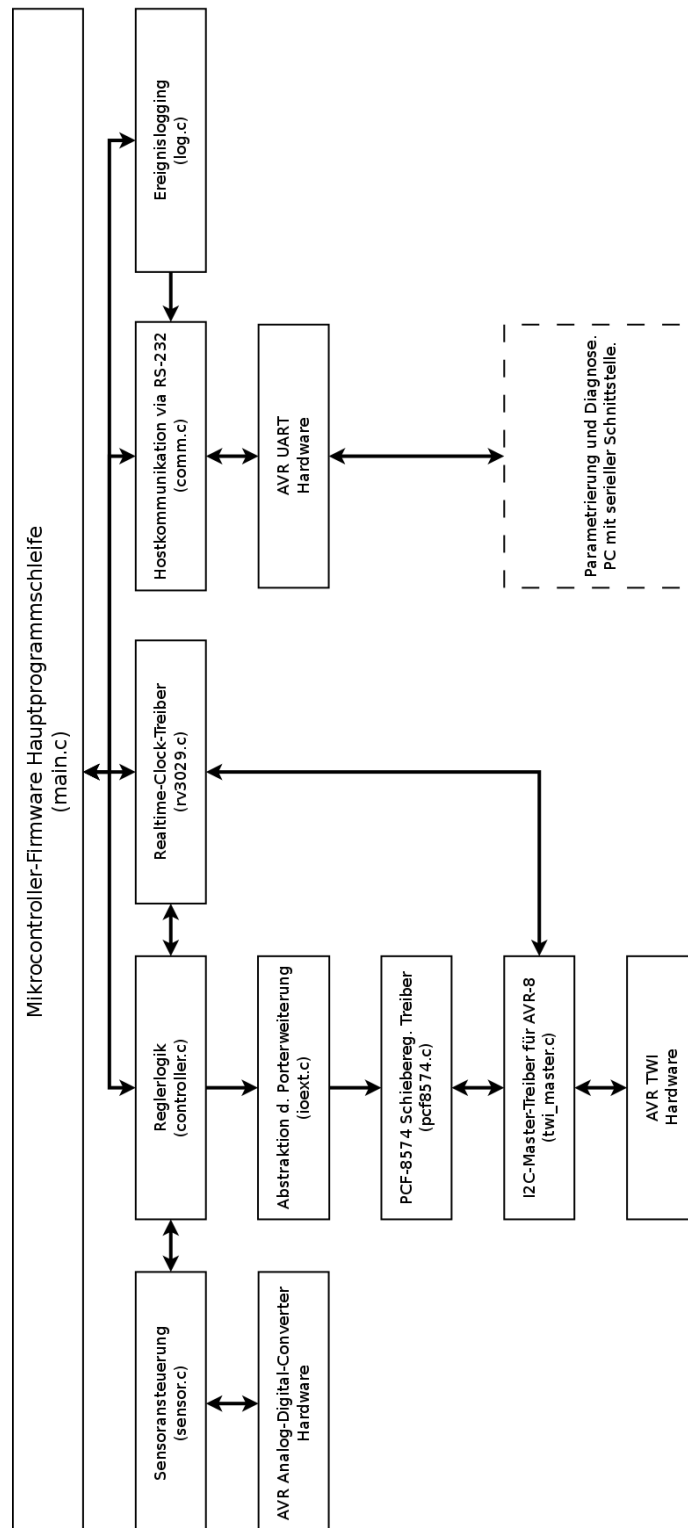


Abb. 4: Blockschaltbild der Steuerung

2.2.2 Der Regelkreis

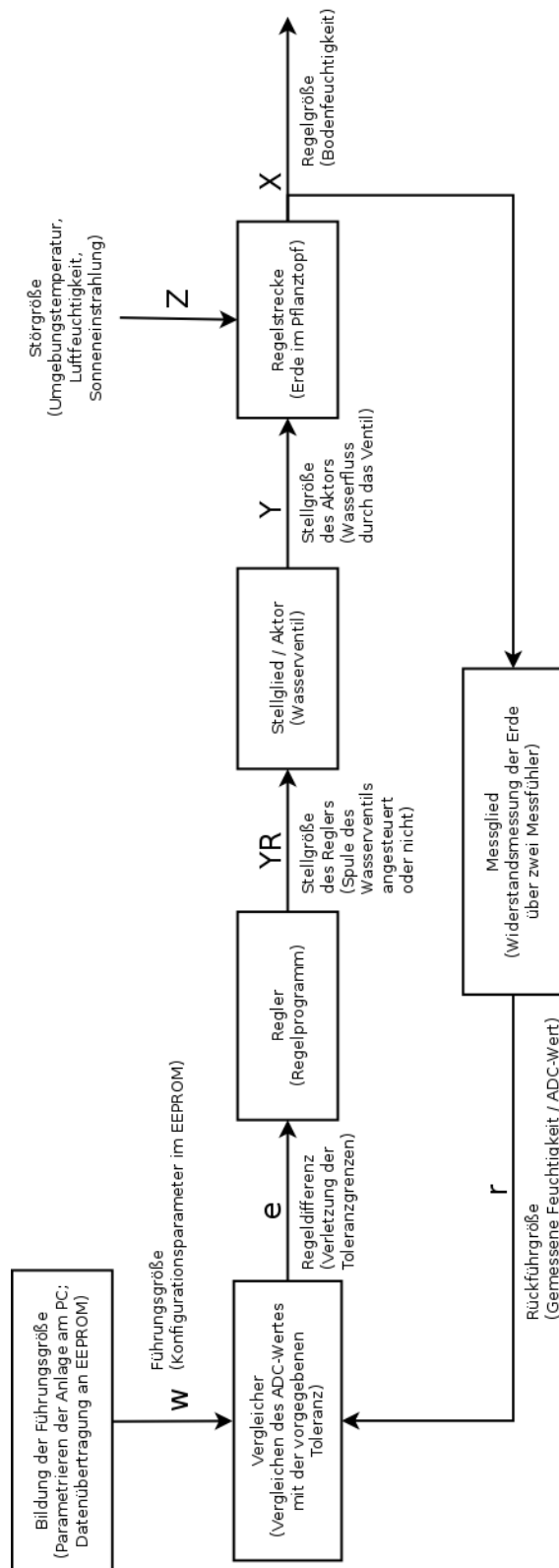


Abb. 5: Blockschaltbild des Regelkreises

2.2.3 Datenblätter der Hardwarekomponenten

Aus urheberrechtlichen Gründen sind die Datenblätter zu den verwendeten Hardwarekomponenten nicht im Archiv dieses Projektes enthalten. Die originalen Datenblätter im PDF-Format können jedoch über folgende Internetlinks heruntergeladen werden:

| Hardwarekomponente | Link |
|---------------------------|---|
| 74HC04 | http://www.datasheetcatalog.com/datasheets_pdf/7/4/H/C/74HC04.shtml |
| Atmel AtMega8A | http://www.atmel.com/Images/Atmel-8159-8-bit-AVR-microcontroller-ATmega8A_datasheet.pdf |
| IRF530 | http://www.datasheetcatalog.com/datasheets_pdf/I/R/F/5/IRF530.shtml |
| Maxim MAX3232 | http://datasheetcatalog.com/datasheets_pdf/M/A/X/3/MAX3232.shtml |
| MICRO CRYSTAL RV-3029-C2 | https://secure.reichelt.de/Real-Time-Clock-Module/RV-3029-C2/3//index.html?ACTION=3&GROUPID=4623&ARTICLE=101070&SEARCH=rv3029&SHOW=1&OFFSET=16& |
| PCF 8574A | http://www.datasheetcatalog.com/datasheets_pdf/P/C/F/8/PCF8574A.shtml |

Tabelle 1: Datenblätter der Hardwarekomponenten

2.3 Software

2.3.1 Quelltextverzeichnisse

Das Projektarchiv ist in folgende Unterverzeichnisse gegliedert:

| <i>Verzeichnisname</i> | <i>Inhalt</i> |
|------------------------|--|
| doc | Quelldateien der Dokumentation |
| firmware | Programm des AVR AtMega-8 Mikrocontrollers |
| host | PC-Programm (Python-basiert) zur Parametrierung der Anlage |
| schematics | Quelldateien der Schaltpläne (KiCAD-basiert) |

Tabelle 2: Quelltextverzeichnisse

2.3.2 Softwareabhängigkeiten des PC-Programmes

Zum Ausführen des PC-Programmes müssen einige andere Softwarepakete auf dem PC vorinstalliert sein:

| <i>Programmname</i> | <i>URL</i> |
|------------------------------|--|
| Python v3.3.x | http://python.org/download/ Für MS-Windows (32 Bit): http://python.org/ftp/python/3.3.2/python-3.3.2.msi |
| PySide v1.2.x/py3.3 | http://qt-project.org/wiki/Category:LanguageBindings::PySide::Downloads Für MS-Windows (32 Bit): http://download.qt-project.org/official_releases/pyside/PySide-1.2.1.win32-py3.3.exe |
| PySerial v2.7 (für Python 3) | https://pypi.python.org/pypi/pyserial Für MS-Windows (32 Bit): https://pypi.python.org/packages/any/p/pyserial/pyserial-2.7.win32_py3k.exe#md5=c6fb580ae7763671297794b8a1d91c9e |

Tabelle 3: Softwareabhängigkeiten

Nachdem diese Vorbedingungen erfüllt wurden, kann das PC-Programm ausgeführt werden. Unter MS-Windows muss dazu die Datei **moistcontrol-gui-win.bat** ausgeführt werden (Doppelklick). Sollte eine Vorbedingung nicht erfüllt sein, wird eine entsprechende Meldung im erscheinenden Konsolenfenster ausgegeben.

Unter anderen Betriebssystemen, muss die Datei **moistcontrol-gui** ausgeführt werden.

2.3.3 Parametrierung der Anlage über das PC-Programm

Nach einem erfolgreichen Start des PC-Programmes, erscheint das folgende Fenster.

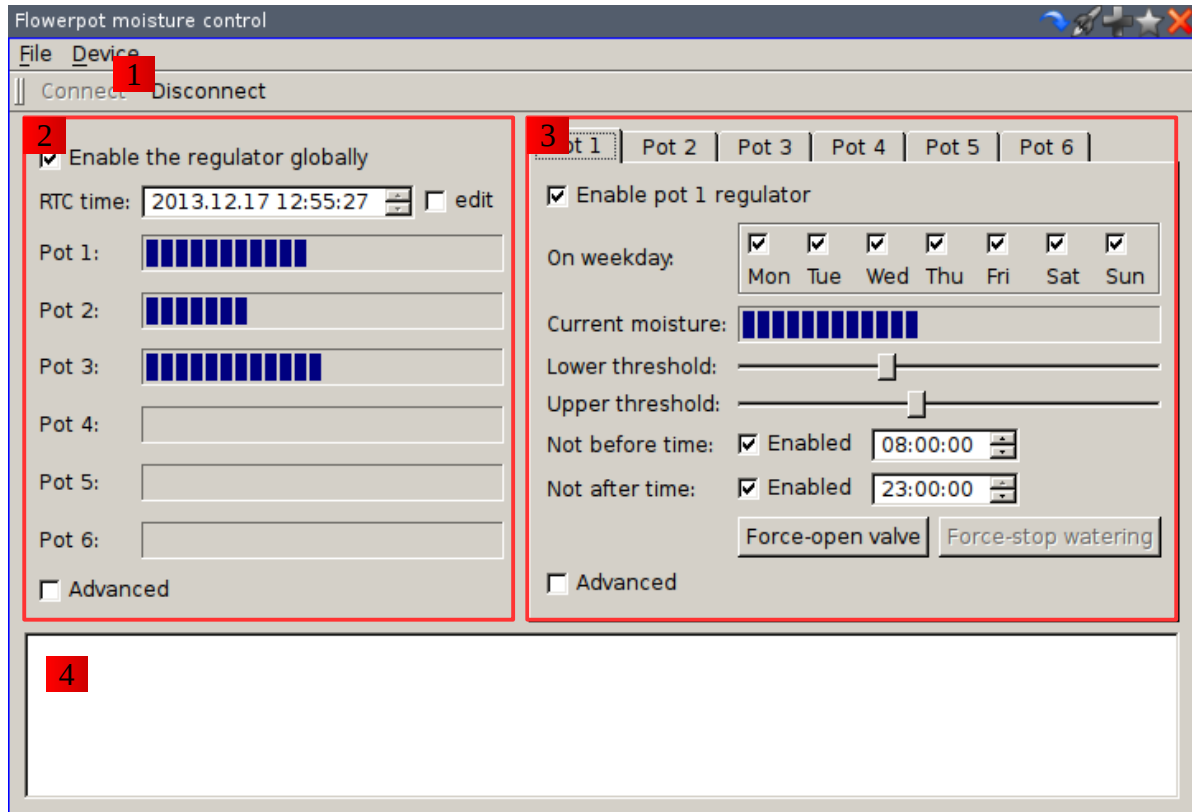


Abb. 6: Hauptbild

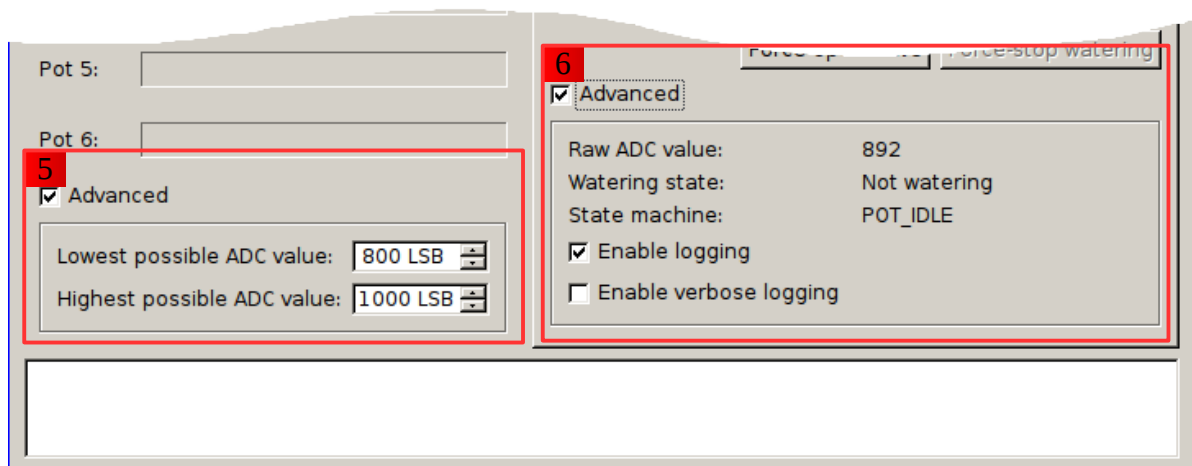


Abb. 7: Hauptbild (Erweiterte Einstellungen)

| | |
|---|---|
| 1 | <p>Schaltflächen zum Auf- und Abbau der seriellen Kommunikationsverbindung.</p> <p>Nach dem Anklicken der "Connect"-Schaltfläche, erscheint ein Dialogfenster zur Auswahl der seriellen Schnittstelle. Hier ist die Schnittstelle zu wählen, mit der die Hardware verbunden ist.</p> |
| 2 | <p>Globaler Konfigurations- und Statusbereich.</p> <p>In diesem Bereich werden globale Statusinformationen wie z. B. eine Übersicht über alle Feuchtigkeiten dargestellt. Desweiteren können globale Konfigurationen wie das Stellen der Echtzeituhr vorgenommen werden.</p> <p>Wenn das Häkchen zur globalen Aktivierung des Reglers nicht gesetzt ist, ist die Messung und Bewässerung aller Pflanzentöpfe stillgelegt.</p> |
| 3 | <p>Konfigurations- und Statusbereich für alle Pflanzentöpfe.</p> <p>Dieser Bereich bietet einen Reiter pro Topf. In diesen Reitern können alle topfspezifischen Statusinformationen abgerufen und Konfigurationen parametrierbar werden.</p> <p>So kann die Regelung für diesen Topf generell, oder nur an bestimmten Wochentagen ab- und eingeschaltet werden. Des Weiteren ist das Einstellen des unteren und oberen Grenzwertes für den Zweipunktregler möglich. Schließlich kann noch ein Tageszeitbereich festgelegt werden, in dem der Regler aktiv sein darf. Außerhalb dieses Zeitbereiches findet keine Bewässerung statt.</p> <p>Über die "Force"-Schaltflächen können Aktionen wie das Öffnen des Ventils oder das Abbrechen des aktuell laufenden Bewässerungsvorganges erzwungen werden.</p> |
| 4 | <p>Statusbereich.</p> <p>In diesem Bereich werden Status- und Fehlermeldungen angezeigt.</p> |
| 5 | <p>Erweiterter globaler Konfigurationsbereich.</p> <p>Hier kann der ADC-Messbereich global begrenzt werden. Messwerte, die außerhalb dieses Bereiches liegen, werden auf den angegebenen Grenzwert begrenzt. Dieser ADC-Messbereich wird intern im Regler auf den festen Feuchtigkeitsmesswertbereich zwischen 0 und 255 skaliert. Das heißt der angegebene untere ADC-Grenzwert entspricht einem Feuchtigkeitswert von 0 und der angegebene obere ADC-Grenzwert entspricht einem Feuchtigkeitswert von 255.</p> |
| 6 | <p>Erweiterter Topfstatus- und Topfkonfigurationsbereich.</p> <p>Hier sind erweiterte Statusinformationen, wie der zuletzt gemessene ADC-Wert oder der aktuelle Zustand des Reglers hinterlegt. Außerdem kann ausgewählt werden, wieviele Statusinformationen über diesen Topf im Statusbereich erscheinen werden.</p> |

Tabelle 4: Beschreibung der Bedienelemente

2.4 AVR AtMega Firmware

2.4.1 Telegrammformat der seriellen Datenübertragung

Die Telegramme haben eine feste Länge von 18 Bytes und sind gegliedert in die OSI-Schicht 2 (Buszugriffskontrolle) und Schicht 7 (Anwendungsschicht).

Grau markierte Felder sind Teil der Buszugriffskontrolle / Data Link Layer (2).

Blau markierte Felder sind Teil der Anwendungsebene / Application Layer (7).

Der Anwendung stehen konstant 12 Bytes Nutzdaten zur Verfügung.

Ungenutzte Bytes werden immer mit Null beschrieben.

Alle Datenfelder werden im "Little-Endian"-Format übertragen und gespeichert. Das heißt, das niederwertigste Byte eines Feldes, bei Feldern mit einer Länge von mehr als einem Byte, wird zuerst übertragen.

2.4.1.1 Logbucheintrag

Das Abrufen eines Logbucheintrages erfolgt mit folgendem Telegrammformat.

Jeder Logbucheintrag enthält eine Zeitkennung, die eine eindeutige chronologische Zuordnung zulässt.

| | Feldname | Datentyp | Beschreibung |
|--|-----------------|-----------------|--|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x00: ID-Kennung des Telegramms. |
| | type_flags | uint8_t | Typkennung des Logbucheintrages |
| | timestamp | uint32_t | Zeitmarkierung des Logbucheintrages. Enthält das Datum und die Zeit beim Auftreten des Logereignisses. |
| | sensor_data | uint16_t | ADC-Wert des Sensors (Bits 0 bis 9). Sensornummer (Bits 10 bis 15). |
| | (reserviert) | 4 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Table 5: Telegrammformat: Logbucheintrag (Sensordaten)

2.4.1.2 Real-Time-Clock-Daten

Die Echtzeituhr kann sowohl ausgelesen, als auch beschrieben werden. Dazu kommt das folgende Telegrammformat zum Einsatz. Alle Datenfelder sind im Binärformat ausgelegt. Die Konvertierung in das BCD-Format der RTC erfolgt in der Firmware.

| | Feldname | Datentyp | Beschreibung |
|--|-----------------|-----------------|--|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x02: ID-Kennung des Telegramms. |
| | second | uint8_t | Sekundenwert. Bereich 0 bis 59. |
| | minute | uint8_t | Minute. Bereich 0-59. |
| | hour | uint8_t | Stunde. Bereich 0-59. |
| | day | uint8_t | Tag. Bereich 0-30. |
| | month | uint8_t | Monat. Bereich 0-11. |
| | year | uint8_t | Jahr. Bereich 0-99. |
| | day_of_week | uint8_t | Wochentag. Bereich 0-6. 0=Montag, 1=Dienstag, u. s. w. |
| | (reserviert) | 4 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Tabelle 6: Telegrammformat: RTC-Daten

2.4.1.3 Globale Konfiguration

Zum Übertragen der globalen Konfiguration an die Firmware, wird das folgende Datenformat festgelegt. Die globale Konfiguration legt topübergreifende Einstellungen fest.

| | Feldname | Datentyp | Beschreibung |
|--|-----------------|------------------------------------|---|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x04: ID-Kennung des Telegramms. |
| | conf | struct controller_global_config | Die globalen Konfigurationsdaten. Für eine Beschreibung der Konfigurationsdaten, siehe Seite 19 zu struct controller_global_config. |
| | (reserviert) | 6 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Tabelle 7: Telegrammformat: Globale Konfiguration

2.4.1.4 Pflanzentopfkonfiguration

Zum Übertragen einer neuen Topfkonfiguration wird ein "struct flowerpot_config" in das Kommunikationstelegramm eingebettet. Zur genauen Definition der "struct flowerpot_config" Datenstruktur, siehe Seite 17.

| | Feldname | Datentyp | Beschreibung |
|--|-----------------|-------------------------|--|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x06: ID-Kennung des Telegramms. |
| | pot_number | uint8_t | Indexnummer des Pflanzentopfes, der konfiguriert werden soll. |
| | conf | struct flowerpot_config | Die Pflanzentopfkonfigurationsdaten. Für eine Beschreibung der Konfigurationsdaten, siehe Seite 17 zu struct flowerpot_config. |
| | (reserviert) | 3 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Tabelle 8: Telegrammformat: Pflanzentopfkonfiguration

2.4.1.5 Pflanzentopfstatus

Zur Abfrage des aktuellen Topfstatus wird ein "struct flowerpot_state" in das Kommunikationstelegramm eingebettet. Zur genauen Definition der "struct flowerpot_state" Datenstruktur, siehe Seite 21.

| | Feldname | Datentyp | Beschreibung |
|--|-----------------|------------------------|---|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x06: ID-Kennung des Telegramms. |
| | pot_number | uint8_t | Indexnummer des Pflanzentopfes, dem dieser Datensatz zugeordnet wird. |
| | state | struct flowerpot_state | Die Pflanzentopfstatusdaten. Für eine Beschreibung der Konfigurationsdaten, siehe Seite 21 zu struct flowerpot_state. |
| | (reserviert) | 5 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Tabelle 9: Telegrammformat: Pflanzentopfstatus

2.4.1.6 Handbetrieb

Die Firmware bietet die Möglichkeit von Hand in die Regelung oder die Ventilsteuerung einzugreifen. Dazu ist das folgende Telegrammformat definiert.

| | Feldname | Datentyp | Beschreibung |
|--|--------------------------|-----------------|---|
| | fc | uint8_t | MAC Frame Control Bits. |
| | seq | uint8_t | Fortlaufende Sequenznummer. |
| | addr | uint8_t | Sender- und Empfängeradresse. |
| | reserved | uint8_t | Reservierter Bereich. |
| | id | uint8_t | Wert 0x06: ID-Kennung des Telegramms. |
| | force_stop_watering_mask | uint8_t | Bitmaske der Pflanzentöpfe, auf denen die Bewässerung abgebrochen werden soll. (Bit 0 → Topf 1, Bit 1 → Topf 2, u. s. w.) |
| | valve_manual_mask | uint8_t | Bitmaske der Pflanzentöpfe, auf denen die manuelle Ventilsteuerung aktiviert werden soll. (Bit 0 → Topf 1, Bit 1 → Topf 2, u. s. w.). Bit gesetzt → aktiv Bit nicht gesetzt → inaktiv |
| | valve_manual_state | uint8_t | Bitmaske der Pflanzentöpfe, auf denen das Ventil manuell geöffnet oder geschlossen werden soll. Das entsprechende Bit in valve_manual_mask muss gesetzt sein, damit dieses Bit effektiv wird. (Bit 0 → Topf 1, Bit 1 → Topf 2, u. s. w.). Bit gesetzt → Ventil auf Bit nicht gesetzt → Ventil zu |
| | (reserviert) | 8 Bytes | Reservierter Bereich, mit Nullen gefüllt. |
| | CRC | uint16_t | CRC-16 Prüfsumme des gesamten Datenpaketes. |

Tabelle 10: Telegrammformat: Handbetrieb

2.4.2 Die zentralen Datenstrukturen und Datentypen

2.4.2.1 Konfiguration

Die Konfigurationsdatenstrukturen werden über das PC-Programm parametrisiert und sowohl im RAM als auch im EEPROM des AtMega abgelegt.

```
enum flowerpot_config_flag {
    POT_FLG_ENABLED      = 0x01,
    POT_FLG_LOG          = 0x02,
    POT_FLG_LOGVERBOSE   = 0x04,
};

typedef uint16_t time_of_day_t;

struct flowerpot_config {
    uint8_t flags;
    uint8_t min_threshold;
    uint8_t max_threshold;
    time_of_day_t start_time;
    time_of_day_t end_time;
    uint8_t dow_on_mask;
};
```

| Feldname / Typname | Beschreibung |
|---------------------------|---|
| POT_FLG_ENABLED | Konstante Bitmaske des "enabled"-Bits im "flags"-Feld. Wenn dieses Bit gesetzt ist, ist der Regler für diesen Topf aktiv. |
| POT_FLG_LOG | Konstante Bitmaske des "log"-Bits im "flags"-Feld. Wenn dieses Bit gesetzt ist, werden Statusinformationen über diesen Topf in das Logbuch geschrieben. |
| POT_FLG_LOGVERBOSE | Konstante Bitmaske des "verbose-log"-Bits im "flags"-Feld. Wenn dieses Bit gesetzt ist, werden erweiterte Statusinformationen über diesen Topf in das Logbuch geschrieben. |
| time_of_day_t | Ein vorzeichenloser 16 Bit breiter Typ, der die Anzahl der doppelten Sekunden seit Mitternacht darstellt. Daraus resultiert eine Auflösung von zwei Sekunden. 0:00 Uhr wird dargestellt als binäre Null. 23:59:58 Uhr (der letzte darstellbare Wert) wird kodiert als binäre 43199. |
| flags | Bitmaske aus POT_FLG_...-Bits |
| min_threshold | Der untere Grenzwert, ab dem der Zweipunktregler mit dem Bewässerungsvorgang beginnt. Wertebereich von 0 bis 255. |
| max_threshold | Der obere Grenzwert, bis zu dem der Zweipunktregler den Bewässerungsvorgang durchführt. Wertebereich von 0 bis 255. |
| start_time | Die Tageszeit, zu der der Regler seinen Betrieb aufnimmt. Vor dieser Zeit ist der Regler inaktiv. |
| end_time | Die Tageszeit, zu der der Regler seinen Betrieb einstellt. Nach dieser Zeit ist der Regler inaktiv. |
| dow_on_mask | Eine Bitmaske der Wochentage, an denen der Regler aktiv ist. Bit 0 steht für Montag, Bit 1 für Dienstag, u. s. w. |

Tabelle 11: Datenstruktur: Konfiguration (Topf)

```

enum controller_global_flags {
    CONTR_FLG_ENABLE = 0x01,
};

struct controller_global_config {
    uint8_t flags;
    uint16_t sensor_lowest_value;
    uint16_t sensor_highest_value;
};

```

| Feldname | Beschreibung |
|----------------------|--|
| CONTR_FLG_ENABLE | Konstante Bitmaske des "enabled"-Bits im "flags"-Feld. Wenn dieses Bit nicht gesetzt ist, sind alle Regler deaktiviert. |
| flags | Bitmaske aus CONTR_FLG_... Bits. |
| sensor_lowest_value | Der niedrigste ADC-Wert, der bei einer Messung auftreten kann. Dieser Wert wird für die Skalierung des Messwertes in den "0-255"-Bereich benötigt. |
| sensor_highest_value | Der höchste ADC-Wert, der bei einer Messung auftreten kann. Dieser Wert wird für die Skalierung des Messwertes in den "0-255"-Bereich benötigt. |

Tabelle 12: Datenstruktur: Konfiguration (Global)

2.4.2.2 Zustand der Sensormessungen

```

enum sensor_status {
    STAT_IDLE,
    STAT_WAIT,
    STAT_WARMUP_P0,
    STAT_WARMUP_P1,
    STAT_ADC_CONV,
};

struct sensor_context {
    enum sensor_status stat;
    jiffies_t timer;
    uint8_t nr;
    uint16_t values[3];
    uint8_t value_count;
};

```

| Feldname | Beschreibung |
|-----------------|--|
| STAT_IDLE | Sensorzustand: Der Ruhezustand. Es wird keine Messung durchgeführt. |
| STAT_WAIT | Sensorzustand: Der Wartezustand zwischen den einzelnen Messungen. Es wird gewartet, bis "timer" abgelaufen ist. |
| STAT_WARMUP_P0 | Sensorzustand: Aufwärmphase mit negativer Polarität. Die Aufwärmphase wird beendet, wenn der "timer" abgelaufen ist. |
| STAT_WARMUP_P1 | Sensorzustand: Aufwärmphase mit positiver Polarität. Die Aufwärmphase wird beendet, wenn der "timer" abgelaufen ist. |
| STAT_ADC_CONV | Sensorzustand: Warten auf ADC-Konvertierung. Der Zustand wird verlassen, wenn die Konvertierung beendet wird. |
| stat | Die Variable für den Sensorzustand STAT_... |
| timer | Zeitählvariable für die Wartezeiten. |
| nr | Die Sensornummer der aktiven Messung. |
| values[3] | Temporäres Feld für die einzelnen Messwerte. Als Endmesswert wird der Median aller Einzelmesswerte genommen. |
| value_count | Anzahl der gültigen Messwerte im "values"-Feld. |

Tabelle 13: Datenstruktur: Sensor

2.4.2.3 Zustand des Reglers

```

enum flowerpot_state_id {
    POT_IDLE = 0,
    POT_START_MEASUREMENT,
    POT_MEASURING,
    POT_WAITING_FOR_VALVE,
};

struct flowerpot_state {
    enum flowerpot_state_id state_id;
    bool is_watering;
    uint16_t last_measured_raw_value;
    uint8_t last_measured_value;
};

```

| Feldname | Beschreibung |
|-------------------------|--|
| POT_IDLE | Reglerzustand: Der Ruhezustand. |
| POT_START_MEASUREMENT | Reglerzustand: Eine Messung wurde angefordert. Es wird auf den Zugriff zum ADC-Wandler gewartet. |
| POT_MEASURING | Reglerzustand: Der ADC-Wandler ist aktiv. Es wird auf eine Rückmeldung vom ADC-Wandler gewartet. |
| POT_WAITING_FOR_VALVE | Reglerzustand: Der Ventilzyklus ist aktiv. Es wird auf das Beenden des Ventilzykluses gewartet. |
| state_id | Die Variable für den Reglerzustand: POT_... |
| is_watering | Diese Variable ist "1", wenn der Bewässerungsvorgang aktiv ist. Das bedeutet der untere Grenzwert wurde unterschritten und der obere Grenzwert wurde jedoch noch nicht durch Anreicherung des Bodens mit Wasser erreicht. |
| last_measured_raw_value | Speicherort für den letzten gemessenen ADC-Wert. Bereich: 0 – 1023. Zu Diagnosezwecken. |
| last_measured_value | Speicherort für den letzten gemessenen und skalierten Sensorwert. Bereich: 0 – 255. Zu Diagnosezwecken. |

Tabelle 14: Datenstruktur: Reglerzustand

2.4.3 Zustandsdiagramm des Reglers

In der Firmware befindet sich pro Pflanzentopf eine Instanz des Feuchtereglers. Dieser arbeitet nach dem Grundprinzip eines Zweipunktreglers. Aus dem folgenden Diagramm können die möglichen Zustandsänderungen des Reglers entnommen werden.

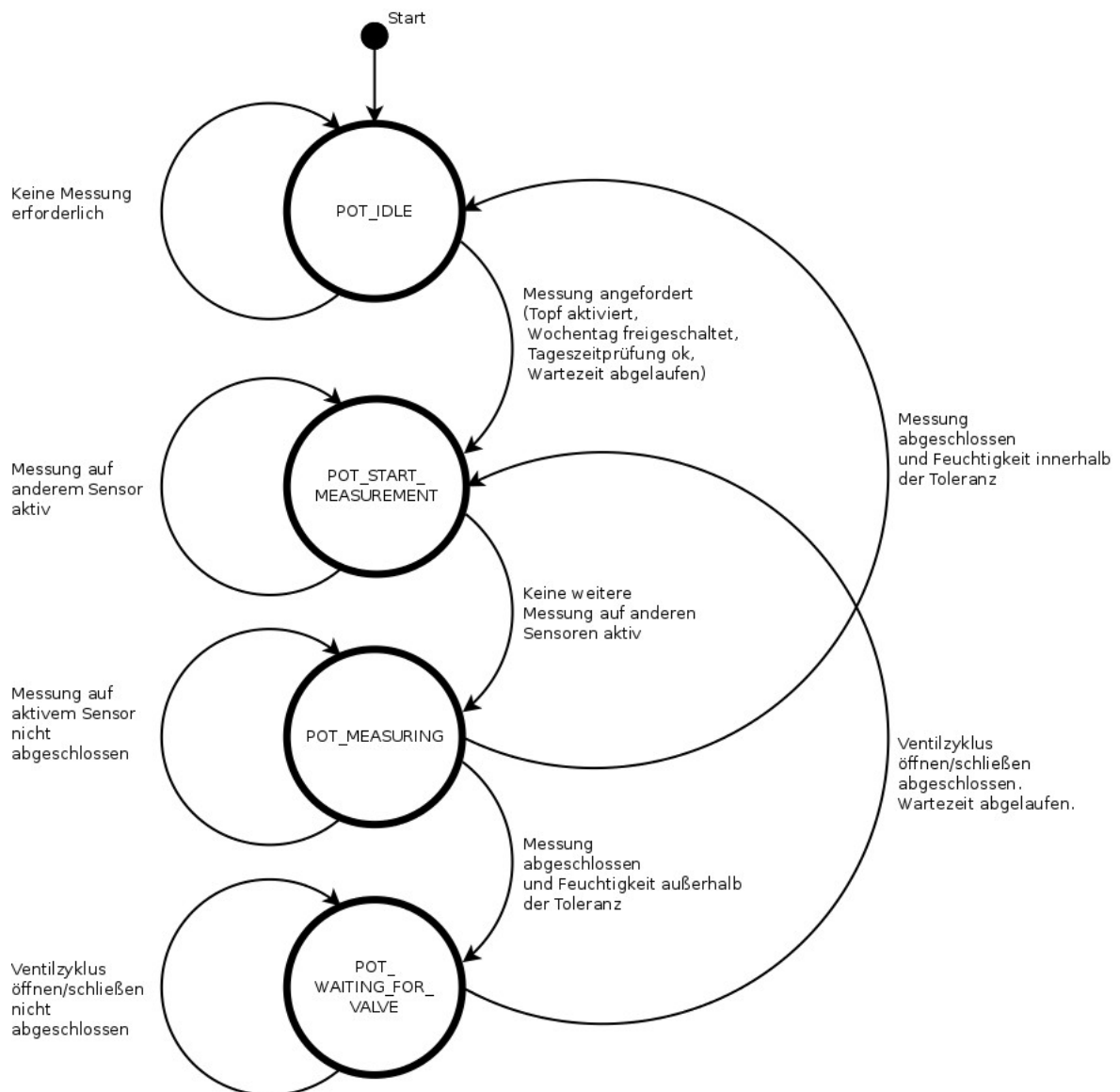


Abb. 8: Zustandsdiagramm (Regler)

3 Persönliche Reflexion

Die Motivation dieses Projektes bestand darin, eine optimale Versorgung von Zimmerpflanzen sicherzustellen und nebenbei auch den Arbeitsaufwand zum Halten dieser Pflanzen zu minimieren. Dieses Ziel wurde erreicht. Bereits eine grobe Einstellung der Regelparameter reicht zur Versorgung relativ anspruchsloser Pflanzen aus. Aber auch anspruchsvollere Pflanzen sind mit dieser Anlage nach sorgfältiger Konfiguration gut zu handhaben.

Trotz der relativ einfachen Zweipunktregelung funktioniert die Regelung der Bodenfeuchte wesentlich genauer als eine vergleichbare manuelle Befeuchtung. Dies kommt empfindlichen Pflanzen in erheblichem Maße zugute.

Der Umfang des Projektes stellte sich im Nachhinein größer als geplant dar. Da die Projektzeit aber sehr großzügig bemessen war, stellte dies kein Problem dar.

4 Anhang

Schaltpläne

Die Schaltpläne der elektrischen Schaltungen können der Datei **schaltplan.pdf** entnommen werden.

Programmablaufplan

Ein Ablaufplan der Firmwarelogik ist in der Datei **programmablaufplan.pap** hinterlegt. Zum Anzeigen des Plans wird die Software **PapDesigner Version 2.2.0.8¹** benötigt.

Lizenzvereinbarung

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.

Alle Dateien dieses Projektes unterliegen der *GNU General Public License Version 2*.

Eine Kopie der *GNU General Public License Version 2* liegt dem Projekt in der Datei **LICENSE.txt** bei.

¹ PapDesigner Download: <http://friedrich-folkmann.de/papdesigner/Download.html>

5 Quellen

Dokumentvorlage, <http://templates.libreoffice.org/template-center/vorlage-fuer-eine-wissenschaftliche-hausarbeit-diplomarbeit-seminararbeit-bachelorarbeit-masterarbeit-examensarbeit> [Stand 20.10.2013].

BNT-Logo, <http://www.bnt-trier.de/fileadmin/templates/images/bnt.gif> [Stand 20.10.2013].

Zeichnung einer Topfpflanze, http://openclipart.org/people/gsagri04/Flower_Pot.svg [Stand 20.10.2013]

Datenblätter der Hardwarekomponenten, siehe Seite 6 dieses Dokumentes. [Stand 16.12.2013].

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten und nicht veröffentlichten Schriften entnommen wurden, sind als solche kenntlich gemacht. Die Arbeit ist in gleicher oder ähnlicher Form oder auszugsweise im Rahmen einer anderen Prüfung noch nicht vorgelegt worden.

Trier, den 03.01.2014

Michael Büsch