Irrigation system for private gardens

Inhaltsverzeichnis

1Introduction....................2
2Scope....................2
3Hazard and Risk analyis....................2
3.1Hazards 2
3.2Risk assessment....................3
4Requirements....................3
4.1Functions....................3
4.1.1Irrigation....................3
4.1.2User Interface....................4
4.1.3Power up/down and Parameters....................5
4.1.4Robustness and Diagnosis....................5
4.2Power supply and valve control....................5
5Architecture....................7
6Hardware Design....................8
6.1Part List (big things)....................8
6.2Terminals 9
6.3Power supply and overcurrent protection....................9
6.4ARDUINO pin assignment....................10
6.5Electromagnetic compatibility (EMC)....................10
6.6Soil Moisture Sensor....................10
7Software Design....................11
7.1Software architecture....................11
7.2Data structure....................12
7.2.1EEPROM....................12
7.2.2Diagnostic Fault Codes (DFC)....................12
7.2.3Global variables (selection)....................12
Diag 12
7.3 Essential structures....................13
7.4Network security....................14
7.5On Board Diagnosis (OBD) and robustness....................14
7.5.1OBD....................14
7.5.2Watchdog....................15
7.5.3Irrigation time monitoring....................15
7.6User Interface Module [ComTelnet]....................15
7.6.1Concept alternatives....................15
7.6.2Telnet State Machine....................16
7.7Internet Time Module [GetTime]....................16
7.8Soil moisture [Humid]....................16
7.9Irrigation Module [Irrig]....................17
7.9.1State Machine....................17
7.10Control the relays [RelayCtrl]....................17
8Areas of improvement, known issues & lessons learned....................18
8.1Software (based on V5.4)....................18
9Initial Setup....................18

9.1SEEED Grove Relais-Modul 5V I2C....18
9.2Setup Network communication....18
10User Manual 20
10.1[s]etup....21
10.2[m]ode....22
10.3[diag]....22

1 Introduction

With longer dry periods in summer the daily irrigation of private gardens becomes more and more time consuming. So some automatic system which irrigates on actual need basis and at the right time (in the morning from 2 to 4) and which can be controlled by your devices in the private IT network is what you need. One solution is described here.

2 Scope

Electric system to control and drive standard irrigation components (el. valves, tubes, nozzles) consisting of:

- Housing with main switch, cable in- and outlets
- internal power supply
- µC – and Ethernet board
- power switching relays
- soil moisture sensor
- software

Not in scope:

- irrigation valves and hydraulic installation
- LAN
- Communication front end app @ user interface (available as free standard apps)

3 Hazard and Risk analyis

3.1 Hazards

(a) to (h) according EU Low Voltage Directive

(a) unsufficient irrigation

(b) excessive irrigation

(c) risks from temperatures, arcs or radiation

(d) electric shock by direct or indirect contact

(e) appropriate isolation

(f) appropriate mechanic properties

(g) robustness against forseeable environment conditions

(h) robustness against forseeable overload conditions

(i) LAN security leak

3.2 Risk assessment

(a) likely to be recognized in time; limited damage → no unreasonable risk

(b)

1. flooding; potential safety issue: drowning of people;
   - concave surface with brim that soaks with water and break suddenly (< 0,01)
   - defect of the system under scope that leads to excessive irrigation (<0,1)
   → no SIL level; measures see 4.1.4 → no unreasonable risk

2. high water cost; many defects can cause excessive irrigation → likely; measures see 4.1.4 → no unreasonable risk

(c) standard measures, see 4.2 and 6.3 → no unreasonable risk

(d) standard measures, see 4.2 and 6.3 → no unreasonable risk

(e) see 6; tool required for opening the housing; protected against direct contact by finger tip when housing is open; mounting plate PE protected → appropriate isolation

(f) use of standard housing → no unreasonable risk

(g) use of standard housing IP65 → no unreasonable risk

(h) standard measures, see 4.2 and 6.3 → no unreasonable risk

(i) see 7.4 → no unreasonable risk

4 Requirements

4.1 Functions

4.1.1 Irrigation

- R1: The start of a irrigation circle shall depend on day time
- R2: If the planned day time is reached and the soil moisture is low an irrigation cycle shall be started;
- R3: The irrigation cycle may include a number of sub cylces for actuating valves in a sequence; background: if too many valves are open at the same time, pressure is likely to drop below the minimum value
- Rxy: The irrigation cycle shall be repeatable with a fixed duty cycle

- Rxy: the software shall provide a parameter to control the number of cycle repetitions; the parameter shall be editable by the user interface
- R4: the SW shall provide a valve test cycle where all valves are actuated in a sequence
- R5: Whether the soil moisture is taken into accout shall be configurable
- R6: The SW shall provide a ad hoc irrigation cycle
- R7: The SW shall provide a manual mode to control the valves for service purposes

4.1.2 User Interface

- R8: The SW shall be controlled by a user interface
- R9: the user interface shall be provided to devices in the local network (not the internet)
- R10: the user interface shall provide these controls
    - activate / deactivate the auto mode
    - start ad hoc irrigation cycle
    - pause / resume the irrigation cycle
    - suspend f. x days (if rain is announced)
    - period for irrigation cycles (every day, every second day, ...)
    - start the test cycle
    - stop all operation
    - hand mode to control each valve separately
    - setup mode for parameters
    - clear on board diagnosis (OBD) fault memory
- R11: the user interface shall indicate the status
    - time
    - mode & state
    - suspend counter
    - time to go valve x
    - OBD (on board diagnosis) fault status
- R12: The SW shall allow the user to use the user interface while the irrigation functions are being executed
- R13: The user interface shall either allow to access from multiple devices at one time or it shall timeout the connection after reasonable time in order to allow access from another device

4.1.3 Power up/down and Parameters

- R10: In case of power loss, the SW shall not lose parameters
- R11: in case of power up, the SW shall return to the mode which was active at power down
- R11: if an irrigation cycle was in progress @ power down, the SW shall not shall not resume the irrigation cycle
- R12: The SW shall have these parameters accessable through the user interface:
  - irrigation duration for each valve; 0 = valve deactivated
  - Auto time mode; auto time & moisture mode
  - irrigation cycle repetition number (for new lawn)
  - Duration for valve actuation during test mode

4.1.4 Robustness and Diagnosis

- Rxy: In case of unplausible program flow the software shall use the µC built in hardware watch dog to trigger a reset
- Rxy: The software shall detect incorrect total irrigation time;
- Rxy: The calculation of the total irrigation time shall use a different approach than for the irrigation control
- Rxy: if the software has detected too high irrigation time the software shall irreversibly inhibit further irrigation
- Rxy the inhibition of further irrigation shall be controlled through a different hardware way than the irrigation control
- R13: The software shall display a fault if the signal from soil moisture sensor is out of range
- R14: The software shall display a fault if the day time information is not received from the internet or if it is unplausible
- ~~R15: The software shall display a fault if the pressure drop at the activation of the valve is not plausible (cancelled due to tight controller resource)~~
- R16: The software shall display a fault if an unreasonable frequency of power-up cycles (resets) is detected
- Rxy: The software shall not lose the fault information in case of reset or power down

4.2 Power supply and valve control

- Rxy: the system shall support to control up to 8 24V AC solenoid valves
- Rxy: In order to avoid dendrides the valves shall be supplied with 24VAC

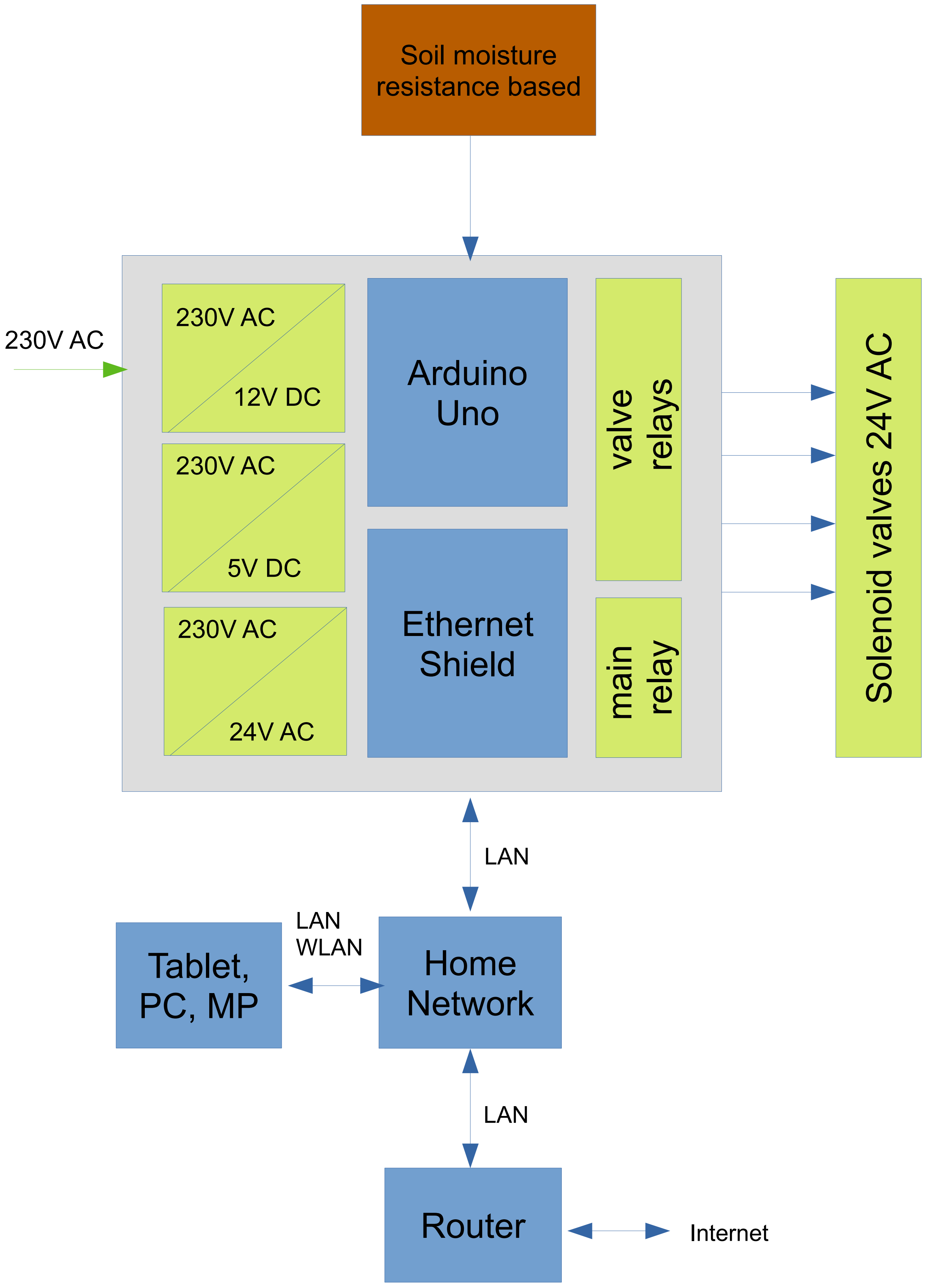
  Safety against electric shock

- Rxy: The 24V AC shall have galvanic separation to the 230V AC public power net
- Rxy: The 5VDC and the 12VDC supply shall have galvanic separation to the 230VAC public power net
- The housing of the irrigation control shall have IP xy
- A special tool shall be required to open the housing of the irrigation control

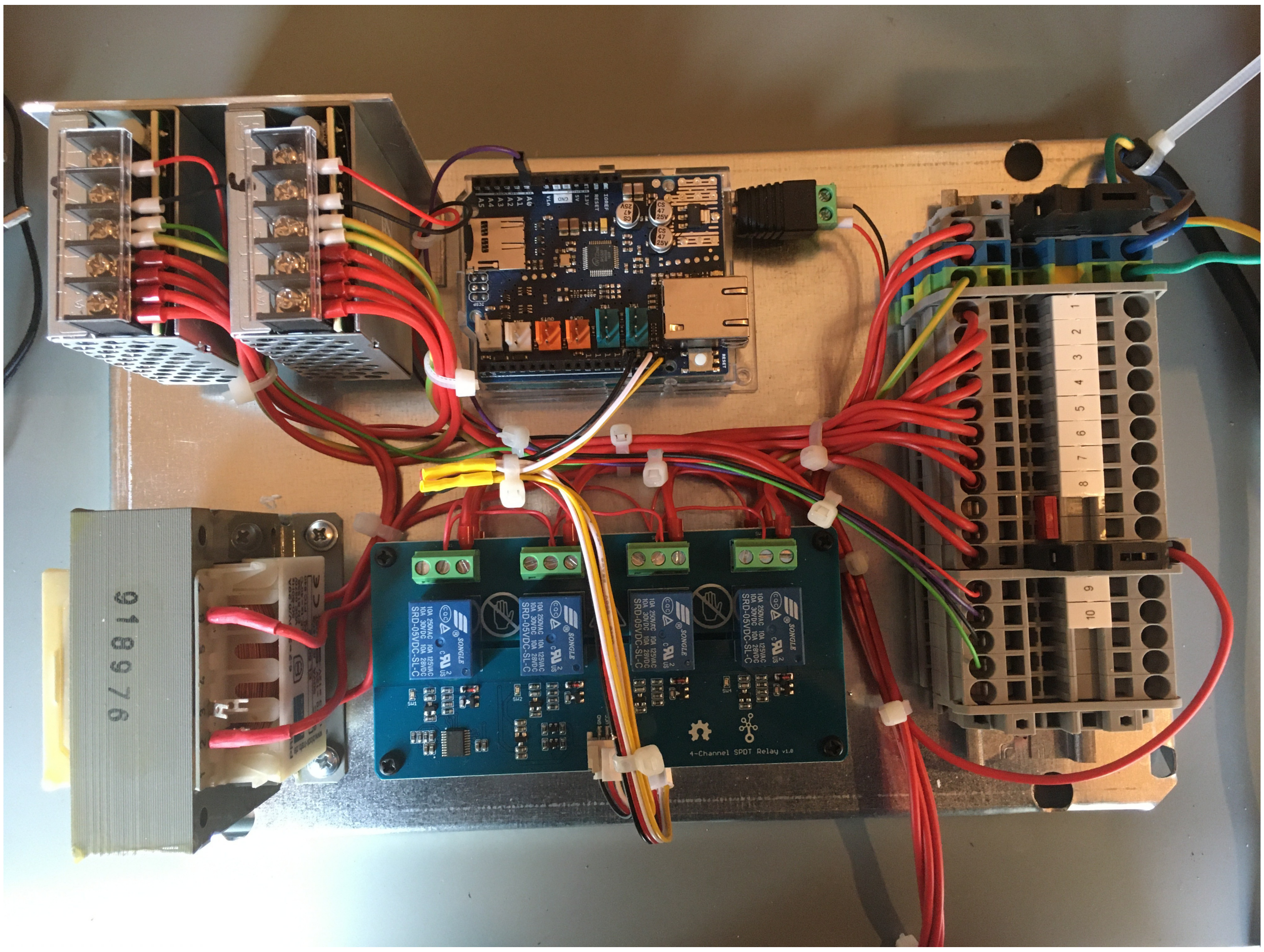
Safety against fire

- a fuse on 230V AC level shall protect against fire
- a fuse on 24V AC level shall protect against overheating of the transformator and the solenoid valves

5 Architecture



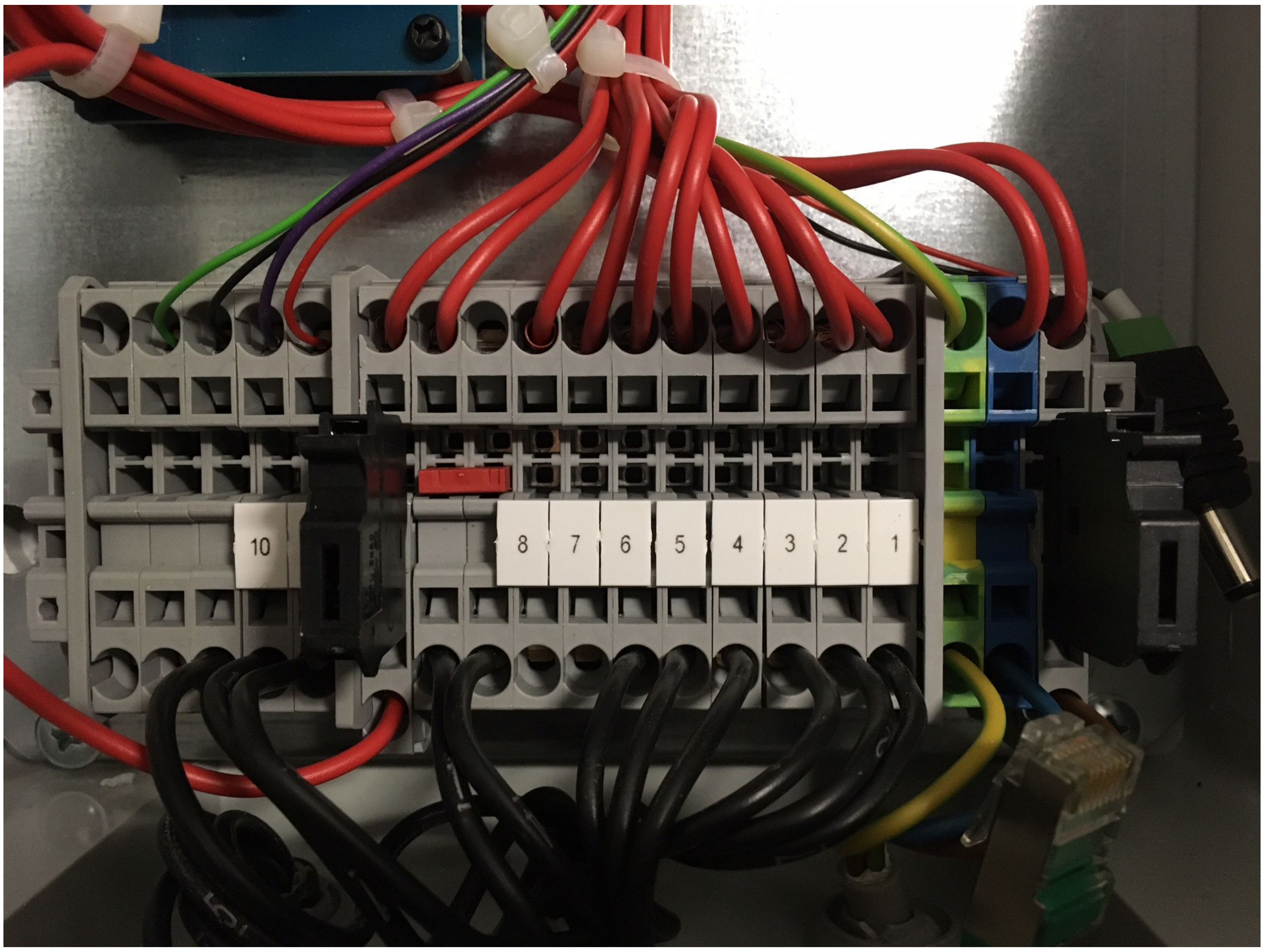
6 Hardware Design



6.1 Part List (big things)

no	part	comment
1	Arduino(TM) UNO Rev3	
1	Arduino Ethernet Shield 2 ohne PoE	
2	SEEED Grove Relais-Modul 5V I2C	actuated by I2C bus
1	SEED Grove Relais, HLS8L with own 5V supply	low actuation current for direct connection with ARDUINO digital out
1	Moisture sensor module	
1	Switching power supply 12 V DC, 15 W	
1	Switching power Supply 5V DC, 15 W	
1	Gehäuse (nur mit Spezialschlüssel zu öffnen)	
1	Nockenschalter 2-polig aus für Fronteinbau	
1	Trafo 48VA 230V / 24V	
	series terminal with 2 fuses	
1		

6.2 Terminals



- 230v Net Power with fuse 1bn, 1bl, 1 gn/ye
- 1 .. 8: valve live lines 24V AC
- 8a, 8b: valve 24V AC com
- 8c: Fuse 24V AC com
- 9: +5V DC soil moisture sensor supply
- 10: soil moisture sensor signal 0 .. 5V DC
- 11: gnd DC soil moisture sensor supply
- 12: n.c.

6.3 Power supply and overcurrent protection

Protection against electric shock and fire

- Fuse on net power live line input is connected to:

    - Main switch on 230V disconnects complete circuits
    - 230V AC / 12V DC Arduino power supply
    - 230V AC / 5V DC supply for relay board, single relay, soil moisture sensor circuit; only Gnd connected with ARDUINO-Gnd
    - 230V AC / 24V AC transformator connected via 230V AC / 5V DC relay
- Fuse on 24V AC (secondary side of the transformator) integrated into terminals
- 5V DC – and 12V DC power supplies (EN62386-1) as well as the transformator are designed with galvanic separation

6.4 ARDUINO pin assignment

A0: Analog input soil moisture sensor

SCL: I2C Bus to 2 x 4 relay boards

SDA: I2C Bus to 2 x 4 relay boards

3: Digital in; default IP address button; low: read user defined (setupIP) IP config from EEPROM, high: read default IP config

4: used by the ethernet shield for SD card control

5: Digital out; main relay for 230V transformator supply; high: Relay closed

10: SS SPI ethernet shield

11: MOSI SPI ethernet shield

12: MISO SPI ethernet shield

13: SCK SPI ethernet shield

6.5 Electromagnetic compatibility (EMC)

Switching off inductive loads may make arcs igniting in the relay causing high dB/dt. This may lead to SW crashes on the arduino.

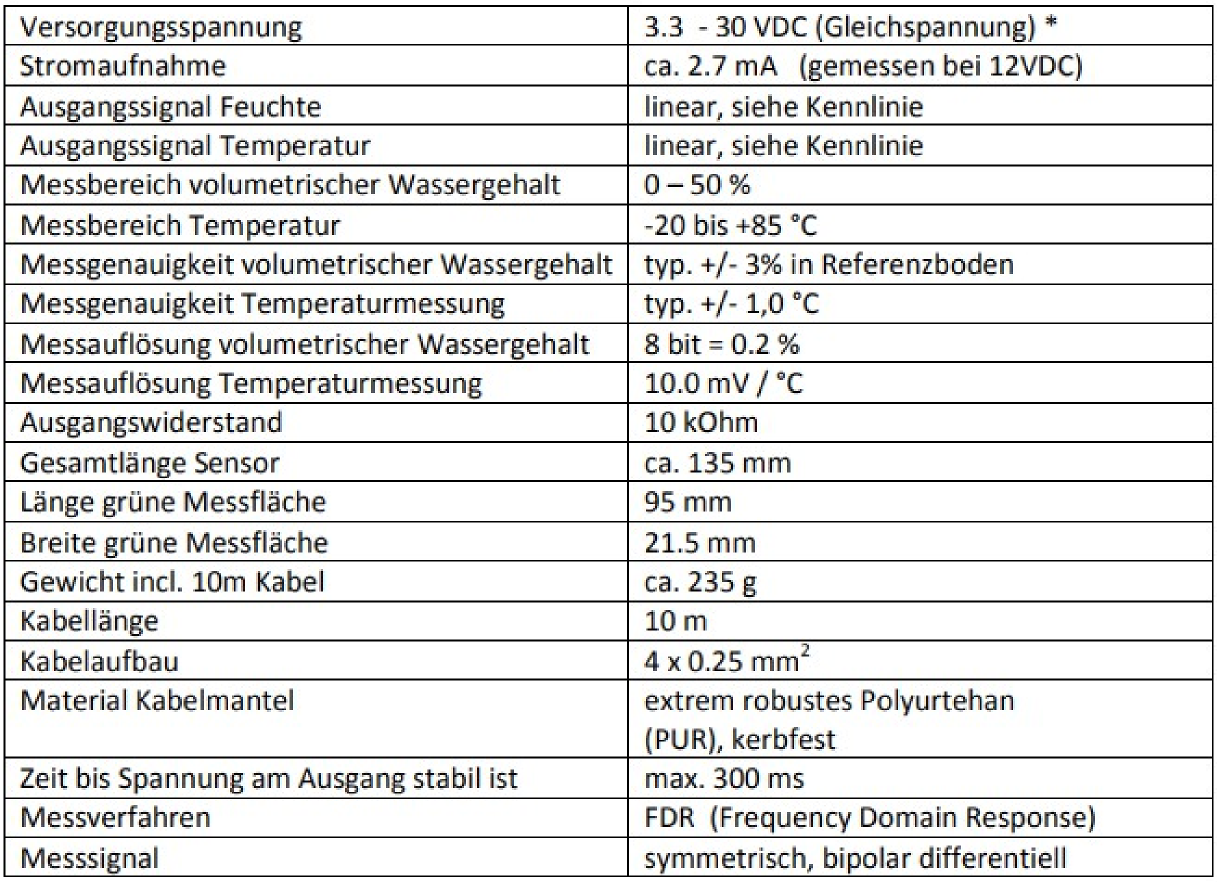
Design:

- Varistor parallel to the 230V transformator primary side
- SW switches the 24VAC relays only when the 24V power supply is off (see 7.10)

6.6 Soil Moisture Sensor

Manufacturer: Truebner

Type: SMT50



Versorgungsspannung	3.3 - 30 VDC (Gleichspannung) *
Stromaufnahme	ca. 2.7 mA (gemessen bei 12VDC)
Ausgangssignal Feuchte	linear, siehe Kennlinie
Ausgangssignal Temperatur	linear, siehe Kennlinie
Messbereich volumetrischer Wassergehalt	0 – 50 %
Messbereich Temperatur	-20 bis +85 °C
Messgenauigkeit volumetrischer Wassergehalt	typ. +/- 3% in Referenzboden
Messgenauigkeit Temperaturmessung	typ. +/- 1,0 °C
Messauflösung volumetrischer Wassergehalt	8 bit = 0.2 %
Messauflösung Temperaturmessung	10.0 mV / °C
Ausgangswiderstand	10 kOhm
Gesamtlänge Sensor	ca. 135 mm
Länge grüne Messfläche	95 mm
Breite grüne Messfläche	21.5 mm
Gewicht incl. 10m Kabel	ca. 235 g
Kabellänge	10 m
Kabelaufbau	4 x 0.25 mm^2
Material Kabelmantel	extrem robustes Polyurtehan (PUR), kerbfest
Zeit bis Spannung am Ausgang stabil ist	max. 300 ms
Messverfahren	FDR (Frequency Domain Response)
Messsignal	symmetrisch, bipolar differentiell

7 Software Design

7.1 Software architecture

- The software is partitioned into cohesive modules so that a low number of global variables is needed for interfaces
- Modules*
    - [IrrigationSys_Vx.y]: First Module in the order;
        - holds #includes and #defines for other modules;
        - implements the scheduler in which processes (as functions) from the other modules are called; Tasks: 1s, 100ms;
        - EEPROM Layout
    - [Diag]: Encapsulates the on board diagnosis (OBD), the monitoring and the main relay caontrol
    - [ComCom]: Common definitions and initializations for ethernet communication
    - [ComTelnet]: Telnet server with user interface

  - [GetTime]: Communication with LAN time server; provision of updated day time as global variable
  - [Humid]: Convert the raw signal DC 0 – 5V from the resistance based soil moisture sensor into a 0 – 100% moisture value
  - [Irrig]: State machine to control the irrigation
  - [RelayCtrl]: Control the two arrays with for the 24V valve relays each by means of the I2C bus; control the 230V main relay by digital out

*) in ARDUINO IDE these are tabs rather than modules in a C language sense. ARDUINO build will put the tabs from left to right into <u>one</u> C-module.

7.2 Data structure

7.2.1 EEPROM

- `EEPROM_START_ADR 0`
- `EEPROM_ComTel_SetupData EEPROM_START_ADR`
  - `user defined parameters`
- `EEPROM_AutoBeforePowerOff (EEPROM_ComTel_SetupData + COMTEL_SETDATAARRSIZ)`
  - `target mode @ startup`
- `EEPROM_IpAdr (EEPROM_AutoBeforePowerOff + 1)`
  - `user defined ip parameters`
- `EEPROM_DiagDfcs (EEPROM_IpAdr + COMCOM_IPADRNO * 4)`
  - `fault memory (flags and fault counters)`
- `EEPROM_Diag_resetCurrCyc (EEPROM_DiagDfcs + DIAG_NODFCS * 2)`
  - `to remember a reset in a diag. cycle`
- `//#define EEPROM_next (EEPROM_Diag_resetCurrCyc + 1)`

7.2.2 Diagnostic Fault Codes (DFC)

- Each DFC is implemented as struct {byte flags, byte debouceThres, byte debouceCtr, byte faultCtr]
- Diag_dfcs[DIAG_NODFCS] holds the pointers to the structs of the DFCs
- the central fault memory functions (trigger diag cycle, clear fault memory, print fault memory, set/reset fault flags) uses Diag_dfcs[] for processing; code maintainance could be improved by code generation to generate the definition and init of Diag_dfcs[]
- the fault counter cannot be incremented by more than one in one diagnosis cycle

7.2.3 Global variables (selection)

Diag

- Diag_dfcs[], see 7.2.2
- Diag_MainPowerRelay: boolean; to control the digital out for the main power relay (primary side of the 230V / 24V AC transformator)

ComCom

- ComCom_Ip[]: holds the IP addresses, configured by the user; stored in EEPROM

ComTelnet

- ComTel_StmP2Fcns: central pointer of the statemachine, see 7.3
- ComTel_nextCharFromClient: updated in state setup only; used to control the update procedure as well as to transport the data to the internal data structure
- ComTel_SetupData[]: user defined parameters

GetTime

- GetTime_Mez[3]: hh, mm, ss calculated from network time protocol (NTP) message

Humid

- Humid_SoilHumidPercent: soil moisture value; linear mapping of raw value 0 .. 5V to 0..100%

Irrig

- Irrig_ModeDisplayStringHeadl[]: String (char array) for displaying the current mode
- Irrig_ModeDisplayStringDetail[]: String (char array) for displaying detailed information e.g. time elapsed etc.
- Irrig_ActiveValvePattern: byte; each bit corresponds to an active valve
- Irrig_HumidHistory[8]: soil moisture values for last 8 days
- Irrig_IrrigHistory: byte; each bit indicates an active irrigation during the last 8 days

7.3 Essential structures

Scheduler

- Time triggered Tasks1s, 100ms; no interrupts, no measures for excessive runtime; module processes are called cyclically without branching
- Init task contains init processes from modules
- Time task init: at the end of the init task the time tasks are called each one time before entering into the regular time triggering

State machines concept

- one struct per state holds the addresses of the state's entry-, cond&exit- and action code
- one stm pointer variable holds the address to the current state's struct
- the stms core is implemented as 4 lines of code, tey are executed in cycl. tasks
  (1) execute* cond&exit-code of the state
  (2) if (condition for state change)

(3) execute* entry code of new state
(4)execute* action code of the (new) state

- *) dereference the element of the struct (pointer to function) where the stm pointer variable is pointing to: stmPVar → struct_state_x{ → entry code, → cond/exit code, → act. code}
- function pointers are changed in cond&exit-codes:
  - if (condition for state change)
  - stmPVar = address of the struct of the new state

Debugging

tbd

7.4 Network security

As the ressources of the Arduino Uno with ethernet shield are too limited for application of secure networking protocols as SSL or https it shall not have access to/from the internet.

Example: Fritz Box as Router: in section Filter: „Irrigate-Wz gesperrt".

Telnet access from the internet can be realized through VPN.

7.5 On Board Diagnosis (OBD) and robustness

7.5.1 OBD

- Diagnostic fault code (DFC) manager
  - triggers diag cycle =1d at midnignt
  - sets / clears fault flag, cycle flag and healed flag
  - increments fault counter (max. 1 increment per diag cycle)
  - clears fault memory on user demand
  - stored in EEPROM
- Diag_DebounceUpDown(..) is used as an interface to the diagnostic fault code manager; parameters: pointer to the diagnostic fault code; instantaneous fault condition
- DFCs (faults)
  - soil moisture sensor short to batt
  - soil moisture sensor short to ground
  - time from time server not received or not plausible
  - unreasonable number of resets or power up cycles
  - max. irrigation time exceeded

7.5.2 Watchdog

Robustness measure against „SW hang up“ as preventive measure for excessive irrigation

- the µC's periferal watchdog which is driven by a independent but still internal µC clock is activated
- watchdog time: 4s
- periodic reset is done as last instruction of the regular 1s task

7.5.3 Irrigation time monitoring

- Calculation of actual irrigation time: increment time as long as valve actuation pattern >0
- Compare actual inrrigation time with max. time = 5h
- If acutal time > 5h set DFC and deactivate main relay

7.6 User Interface Module [ComTelnet]

7.6.1 Concept alternatives

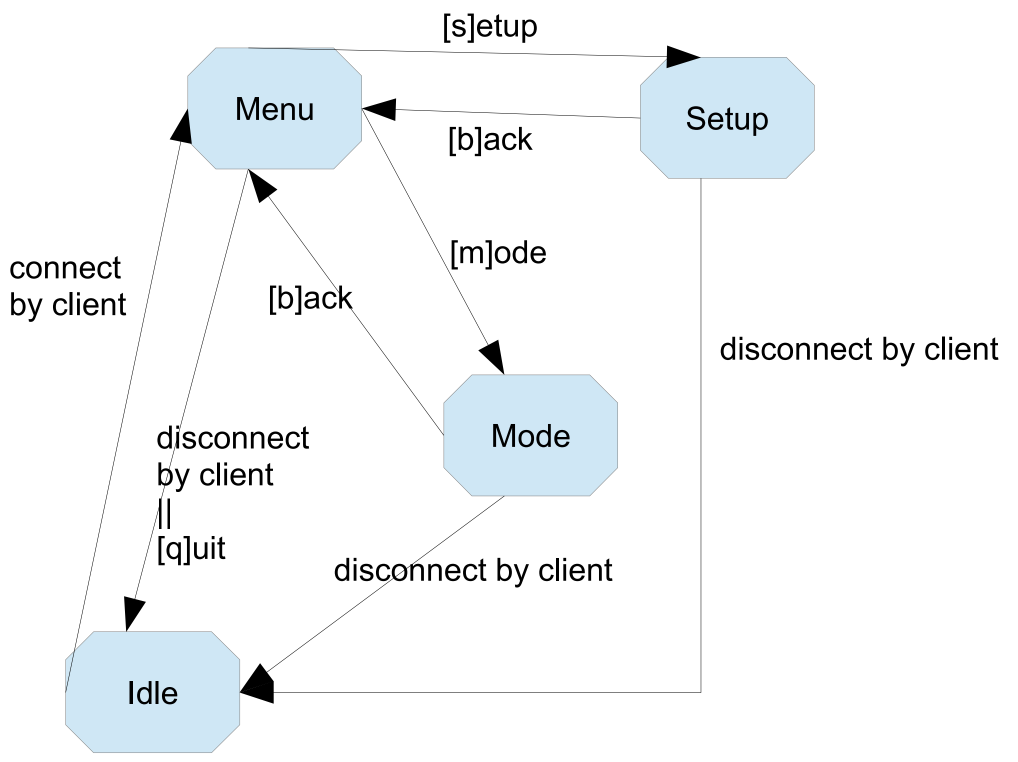
1. [decided] Arduino as telnet server; use a telnet client app on the LAN / WLAN devices
    - telnet is text based and belongs to the app. Layer of the TCP/IP protocol stack
    - not feasible for using from the internet due to security (encryption is not possible) but ok behind the router fire wall
    - user installs any free telnet app on handheld or PC within the LAN
2. [discarded] Use udp
   as Udp is on transport layer level it is too much to program an application on it

3. [discarded]Web server with html page
    - likely usage (at the same time) of SD card and ethernet required
    - html seems quite complex for programming a interactive interface (at least I have no idea...)
4. [discarded] App f. Android und ios
    - never done before, not so much time …

- [discarded] Home automation
    - Irrigation system would be the only application → effort overhead

○

7.6.2 Telnet State Machine



In state „Mode“ incoming characters typed in by the user into the telnet frontend app. are forwarded by global variable to the irrigation state machine.

State SetupIP is used for setup the network data like subnet mask, Gateway, IP address, … . The state is similar to state setup but not depicted.

7.7 Internet Time Module [GetTime]

- Use the LAN/WLAN DSL routers as time server in the LAN (rather than a internet time server)
- access the time server by a UDP NTP (network time protocol) time request message
- calculate day time from time servers UDP reply message (time in seconds since 1970)

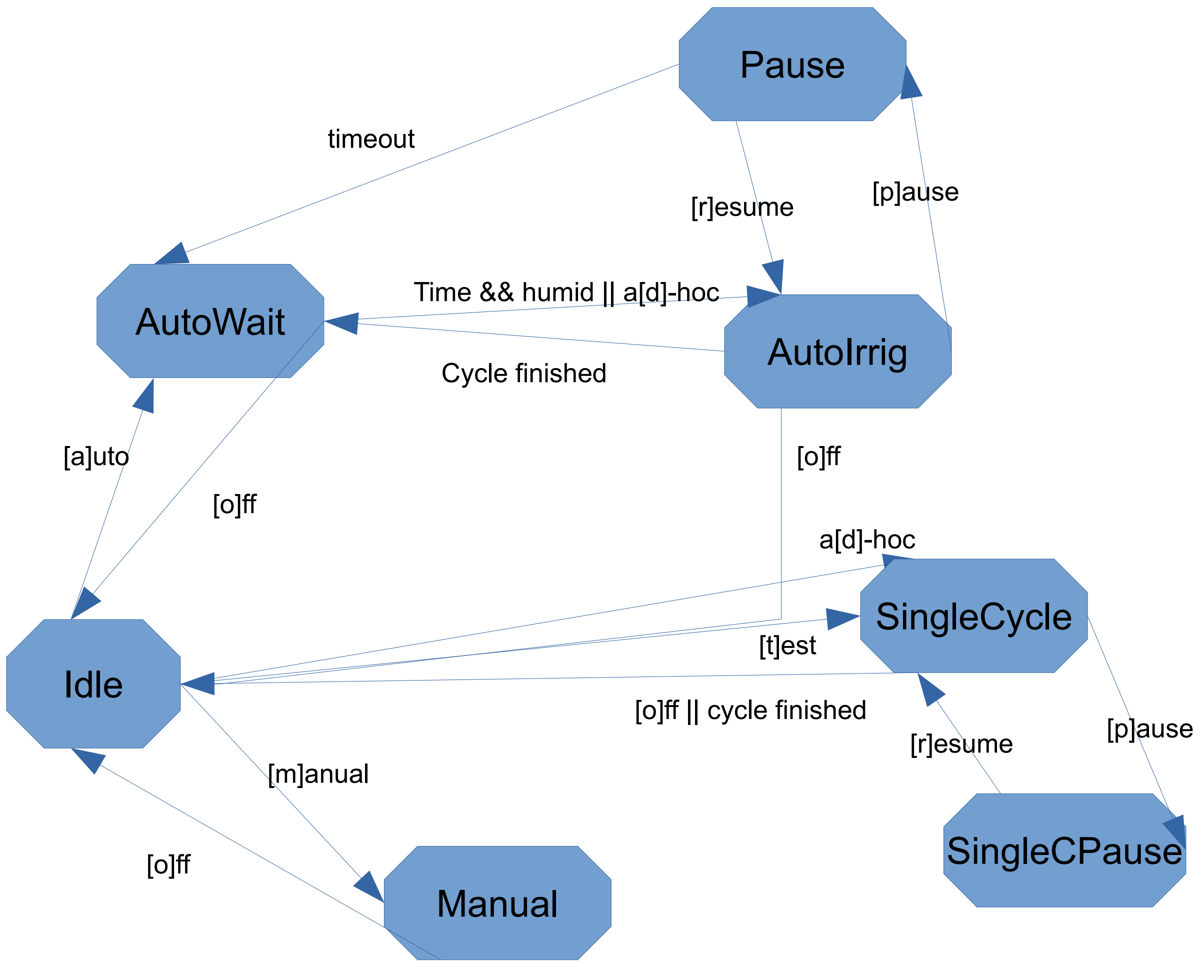
7.8 Soil moisture [Humid]

- linear mapping of 0..5V sensor signal to software variable
- Diagnosis of the sensor signal: SCB (short to batt), (SCG) short to ground
- mapping of raw value to phys. based moisture definition is not done; rationale: sensor remains at fixed place so stable parameters (soil density, chemical structure) ensure sufficient repeat accuracy; threshold is set by the user based on optical quality criterion

7.9 Irrigation Module [Irrig]

7.9.1 State Machine

- All functionality is covered within the state machine.
- Commands for changing mode, setting pause state or suspending irrigation cycles are received by global variable from ComTelnet-module.
- „automaitc“ state is saved in EEPROM. In case of power loss or reset the automatic state is set depending on EEPROM's value.
- The irrigation durations for valve 1 ..8 are executed in sequence starting from valve 1
- the irrigation history (8 days) is recorded and displayed (not stored in EEPROM)
- the soil moisture is recorded (8 days) at midnigt and, in case of starting a irrigation cycle, updated right at the start of the cycle



7.10 Control the relays [RelayCtrl]

- control 2 boards of 4 relays for the 8 valves by I2C bus; change the status of the relays (on --> off; off--> on) only if the 24V power supply is off
- control the main relay (230V AC primary side of the transformator) by digital out

8 Areas of improvement, known issues & lessons learned

8.1 Software (based on V5.4)

- F( )-Macro was used in order to store constant character arrays for print outputs in the program memory instead of the RAM. .Used in .. . print[ln](F(...)) it will send each single character as one package on ethernet instead of making bigger packages. For alternative see link in the source code comments.
- Irrigation schedule depends on internet time using the NTP (network time protocol). Sometimes the protocol stops working and the time is no longer updated and it would only heal throgh a device reset. As the root cause could not be determined this workaround has been implemented: if the time doesn't update it is incremented locally; if there is no update through NTP for ~ 3 weeks, a device reset will be triggered from SW.
- Bug: sometimes the terminal responses „host could not be contacted; connection refused"; occurrence: ~ once per 100 contacts
  healing: power on reset; sometimes it heals overnight, sometimes not

9 Initial Setup

9.1 SEEED Grove Relais-Modul 5V I2C

For the 8 channels two relay boards controlled by i2c bus are used. As they are delivered with the same i2c-address, usually 0x11, the address on one of the boards needs to be changed by a flashing procedure which is described on the web page of the supplier.

In the code IrrigationSys_V5.4.ino the adresses are defined:

```
#define RELAYCTRL_8CHANNEL //change this into a comment if you use
only 4 relays

#define RELAYCTRL_ADRLS4BITS  0x11  // lower 4 channels

#define RELAYCTRL_ADRMS4BITS  0x12  // higher 4 channels
```

9.2 Setup Network communication

Due to resource limitation no DHCP is realized, thus fixed network adress is applied.

- Get the MAC address of your arduino, go to the module ComCom and update `#define MAC_ADR 0x.., 0x.., 0x.., 0x.., 0x.., 0x..`

- Check whether the IP address 192, 168, 178, 47 is free in your LAN. If not, enter some free IP address in ComCom module

  ```
  byte ComCom_Ip[COMCOM_IPADRNO][4] = {
                      {192, 168, 178, 47},  // ip
                      {192, 168, 178, 1},   // dns
                      {192, 168, 178, 1},   // gateway
                      {255, 255, 255, 0},   // subnet mask
                      {192, 168, 178, 1}    // time server
                    };
  ```

  You may also edit the addresses for dns, gateway and time server directly here. If you then keep pin 3 on high level voltage (="use default IP config") you can skip the following steps
- apply high voltage level to pin 3 and do a power on reset to the arduino; it will apply the IP adresses as given (or modified) above
- Install a telnet front end (telnet client is sufficient) on one of your network devices e.g. „Termius" available for Android or iOS;
- Configure a new host by Alias = sth. like „Wenzler Garten", Hostname = [ip address], use SSH = No, Port = 23 (standard for telnet)
- Call host „garden irrigation"; you should see

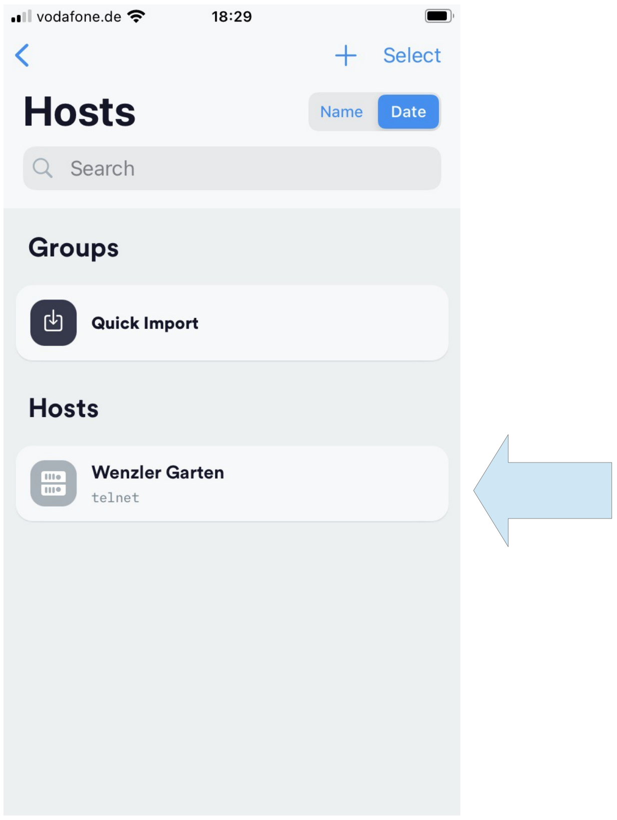
  ```
  Wenzler's Gartengießer V5.4
  ==========================
  Gießtage 0  0  0  0  0  0  0  0
  Feuchte  0 0 59 60 58 60 59 61


  Menu: [m]ode, [s]etup, setup[i]p, [d]iag, [q]uit


  18:41:00
  ```

- type „i" for entering the 5 IP address information according to the structure given above. In a normal home setup the dns, gateway and time server is the address of your internet router
- In your internet router:
  - configure the xy //ip as fixed address
  - make sure „time server" is activated in your router (your router shall provide a time server to your LAN)

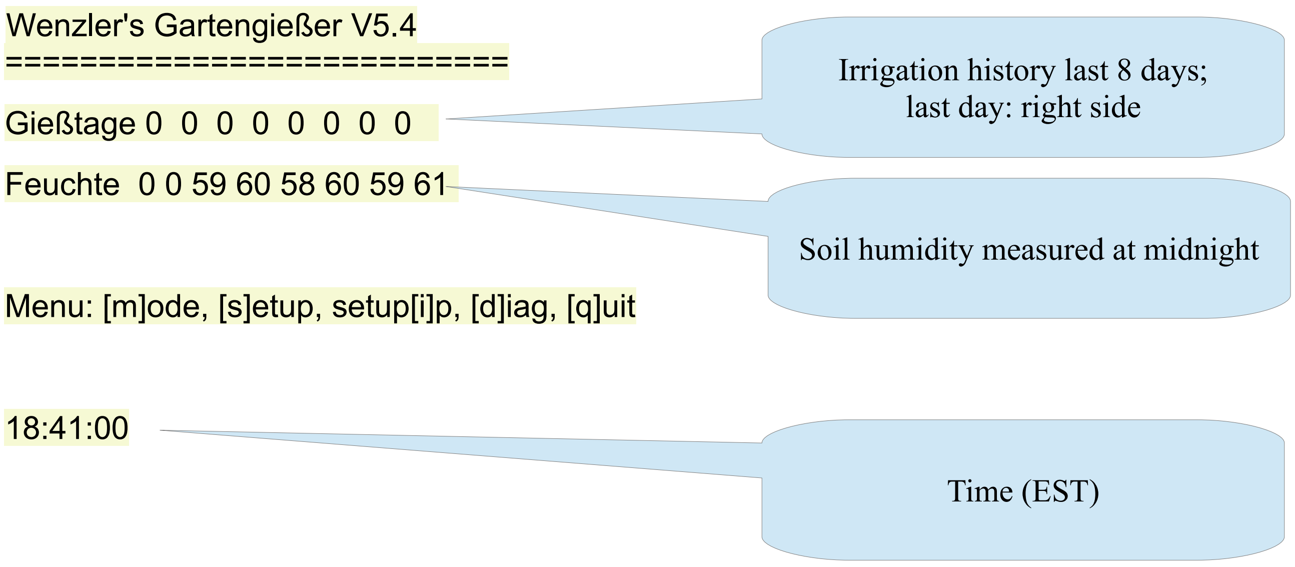
    - deactivate access to and from internet to the xy //ip address of the arduino; this is strongly recommended for IT security
- apply low voltage to pin 3 (= “use user’s IP setup“) and and do a power on reset to the arduino (from now the arduino will take the IP addresses which you configured under setup[i]p)

10 User Manual

Connect to the irrigation system by using the telnet front end on your mobile phone or other device in your LAN/WLAN as described under Setup Network communication.



You should see the main menu:

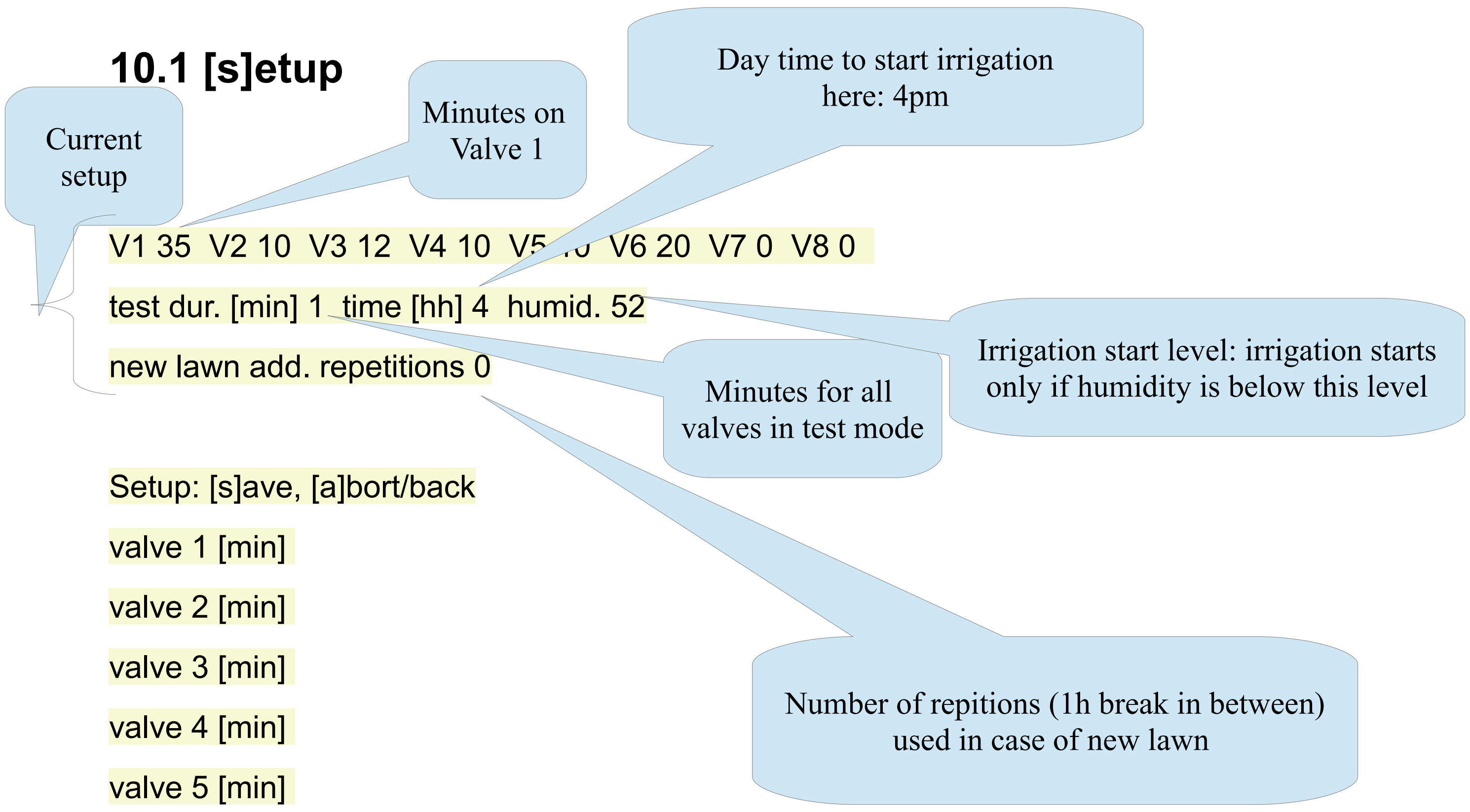


```
Wenzler's Gartengießer V5.4
============================

Gießtage 0  0  0  0  0  0  0  0

Feuchte  0 0 59 60 58 60 59 61


Menu: [m]ode, [s]etup, setup[i]p, [d]iag, [q]uit


18:41:00
```

Irrigation history last 8 days; last day: right side
Soil humidity measured at midnight
Time (EST)

You can connect to the irrigation system only from one device at a time. After 5mins without activity the irrigation system will terminate the session. [q]uit will terminate the session immediately.

10.1 [s]etup


Current setup
Minutes on Valve 1
Day time to start irrigation here: 4pm
Irrigation start level: irrigation starts only if humidity is below this level
Minutes for all valves in test mode
Number of repitions (1h break in between) used in case of new lawn

```
V1 35  V2 10  V3 12  V4 10  V5 10  V6 20  V7 0  V8 0
test dur. [min] 1  time [hh] 4  humid. 52
new lawn add. repetitions 0

Setup: [s]ave, [a]bort/back
valve 1 [min]
valve 2 [min]
valve 3 [min]
valve 4 [min]
valve 5 [min]
valve 6 [min]
valve 7 [min]
valve 8 [min]
test dur. [min]
time [hh]
humid. 0-100
new lawn add. repetitions, max.4
```

Type in numbers correctly as there is no plausibility check in the software. In case of a typo press [a]bort at the end and repeat.

10.2 [m]ode

Mode: [a]uto [o]ff [p]ause [r]esume a[d]-hoc [t]est [m]anual [b]ack2menu [s]uspend+1d s[u]spend-1d

status: auto

susp h 0 hum 60

- [a]uto: it will execute an irrigation cycle operating V1 to V8 in a sequence if at start time the humidity is below the setup value;
  if „new lawn add. repetitions“ is setup x > 0 the cycle is repeated x times every 1 hour; its purpose is that the seed of a new lawn is not drying out too much on sunny days
- [s]uspend+/-1d: irrigation can be suspended in 24h-steps by the user; purpose: in case of rain forecasted for the next day
- a[d]-hoc: an irrigation cycle can be started by the user immediately during [a]uto and during [o]ff.
- [p]ause / [resume]: A cycle in progress can be paused / resumed by the user. Note that after 30mins of pause the cycle gets aborted. Thus in [a]uto the subsequente cycles are not influenced by [p]ause.
- [t]est: one test cycle will be triggered; it will execute an irrigation cycle operating V1 to V8 in a sequence, all valves with the same duration „test dur.“

10.3 [diag]

The purpose of the diagnosis menu is to support trouble shooting by providing information about fault status from within the irrigation system. Fault information will not be lost at power loss.

The software contains functions to identify 5 faults including their history.

Flag positions: [healed] [cycle] [fault cond] [fault]

Flag definitions:

healed: fault is not present (confirmed)

cycle: don’t care about this information, users

fault cond: the instantanious condition for the fault is true (fault not confirmed)

fault: fault present (confirmed)

fault count: number of days on which the fault was present (in confirmed status)

time plaus // persistently invalid or no time information from time server

flags 1100

fault count 1

e.g.:
healed (confirmed), fault cond not present, fault not confirmed

humid scg // short circuit to ground humidity sensor

flags 0

fault count 0

humid scb // short circuit to battery (power supply +) humidity sensor

flags 0

fault count 0

PowUpPlaus //number of unplausible power up cycles; note that the main switch will cause this too

flags 0

fault count 0

MonIrrTime //unplausible total irrigation time detected; (protects against excessive irrigation); defined reaction: deactivate main relay

flags 0

fault count 0

[b]ack to menu [c]lear fault memory [r]eset device

[c]lear fault memory: Delete all stored information about faults

[r]eset device: restart the system (same as switch off → switch on)