Raspberry Pi (RPI - in House)

Setup process

The RPi operates from an SD Card on which has been installed 64 bit Debian OS. From https://www.raspberrypi.com/software/ Raspberry Pi Imager) is downloaded to create boot data on SD card. From there the latest Debian 64 Linux OS is installed on the SD Card, with user credentials being set and SSH enabled. This the enables login via ssh. Docker docker.sh is installed and thereafter a a docker container is installed for HomeAssistant with Home Assistant then installed with user credentials.

Mosquitto is similarly created in its own container creating an MQTT service account. Running these is the essential task of this computer.

Data Sent and Received

RPI receives and transmits messages from and to the remote Home Assistant Dashboard via the internet. The local version of Home Assistant receives and transmits messages from and to AR1 via the MQTT broker.

Aduino 1 (AR1 - in House)

Data Sent and Received

Directly receives sensor data via core wires of the multicore cable.

- Flame sensor
- Temperature sensors
- Heartbeat

Receives sensor outputs via the RS485 communication module A & B wires of the multicore cable currently:

- Battery Voltage (locally measured)
- Water Level in Tank
- Flow Rate

Transmits relay action instructions via the wires of the multicore cable for:

- Pump (ON/OFF)
- RV1 (Relay Valve 1 – High / Reduced flow)
- RV2 (Relay Valve 2 – Intermittent Flow ON/OFF)
- RV3 (Relay Valve 3 – Shed cooling ON/OFF)
- RV4 (Relay Valve 4 – Gel injection ON/OFF)

Transmits all sensor and relay action data to the MQTT Broker running in RPI. RPI in turn retransmits this via internet to Home Assistant Cloud.

Arduino 2 (AR2 - Autonomous Support Module – in Shed)

Data Sent and Received

Re-transmits selected sensor outputs to AR1 via the the RS485 communication module A & B Wires. Sensors currently involved are:

- Battery Voltage (locally measured)
- Water Level in Tank
- Flow Rate
- Pump oil and water pressure
- Ember sensor
- AR1 heartbeat

Transmits this data to AR3 via local wifi peer to peer connection.

Arduino 3 (AR3 - Autonomous Module - in Shed)

Data Sent and Received

Receives sensor outputs from AR2 via local wifi peer to peer connection.

- Battery Voltage (locally measured)
- Water Level in Tank
- Flow Rate
- Pump oil and water pressure (High/Low)
- Ember sensor

Receives in parallel the data received by AR1 from in-shed sensors and action commands from AR1:
Sensor Data

- Temperature sensors

When contact is lost with AR1, takes control relaying action instructions via the wires of the multicore cable for:

- Pump start/stop
- RV1 (Relay Valve 1 – High / Reduced flow
- RV2 (Relay Valve 2 – Intermittent Flow ON/OFF)
- RV3 (Relay Valve 3 – Shed cooling ON/OFF)
- RV4 (Relay Valve 4 – Potentially used for Gel injection ON/OFF)

Simulation Modes

For purposes of testing and development sensor inputs can, in most cases, be simulated when the sensor is not attached by setting SIMULATION_MODE = true in *Simulation Flags and Settings*. With this the simulation of indvidual sensors can be set with corresponding settings within that program block.

In addition, the capability is provided to command changes to key parameters from the Arduino IDE serial monitor window while the program is running. This enables testing scenarios to be set up. For details of the simulation serial monitor commands see these Technical Notes section: **Error! Reference source not found.**.

AUTOTEST CAPABILITY

The shed Autonomous Module is equipped with code to run one or more Automatic test sequences. The first of these sequences can be automatically initiated by pressing the AUTOTEST Button briefly which is attached to the Autonomous module housing.

The serial monitor can also be used to issue the commands used to set up test sequences which may be used prior to the Arduino being connected to actual sensors and relays. These sequences can be pre-programmed in the *Autotest Execution SETTINGS* block of the program. After each step in the sequence the command "PRINT_STATE" shows the resulting settings.

The test sequences are referenced by number so that for, say, sequence 2, they may be set in motion while the program is running by typing AUTOTEST_ON 2 into the serial monitor window. Testing can be assisted by utilisng a strip of LED lights, each linked to a single relay controlling pin on Arduino 3 (as in the picture below).

In order to speed up simulation, the program allows the pre-programming in the *AUTOTEST SETTINGS* block, of scaling constants. Here we can set AUTOTEST_INTERVAL_MS - the time between successive steps in the sequence (in milliseconds), a SHORT_AUTOTEST_INTERVAL_MS which allows a test to quickly skip through the first SHORT_AUTOTEST_STEPS, and an INTERMITTENT_SCALE_FACTOR which can be used to speed up or slow down the intermittent cycle timings (so that with a scale factor of 0.05 1 min is performed with a scaled time of 3 sec.

Appendix 18 The Sprinkler Cycle Logic.

- `SprCycle` = Sprinkler Cycle Switch (ON/OFF)
- `Mode` = Cycling Mode selection (`VALVE_CYCLING` or `PUMP_CYCLING`)
- `Tmin` = minimum minutes threshold for allowing Pump-Cycling (e.g., 10)
- `TimeON`, `TimeOFF` in minutes (or use ms in code and compare against `Tmin*60000`)
- Valve states: `PASS` (pass-through), `RECIRC` (recirculation)
- Pump states: `ON`, `OFF`, `NC` (not controlled by sprinkler cycle)

SPRINKLER CYCLING POLICY TABLE (A4)

ID	SprCycle	TimeON condition	TimeOFF condition	Mode selected	Mode effective	Valve during TimeON	Valve during TimeOFF	Pump during TimeON	Pump during TimeOFF	Notes / intent
1	OFF	any	any	any	n/a	PASS	PASS	NC	NC	Base safe state. No inadvertent recirc. Pump may run via other controls.
2	ON	=0	=0	any	n/a	PASS	PASS	OFF	OFF	Safety idle. "Valve pass-through, pump off."
3	ON	>0	=0	VALVE_CYCLING	VALVE_CYCLING	PASS (continuous)	n/a	ON (continuous)	n/a	Continuous flow, no OFF phase.
4	ON	=0	>0	VALVE_CYCLING	VALVE_CYCLING	n/a	RECIRC (continuous)	ON (continuous)	n/a	Continuous recirc while SprCycle ON.
5	ON	>0	>0	VALVE_CYCLING	VALVE_CYCLING	PASS	RECIRC	ON	ON	Default "recirculation cycling": valve toggles; pump stays on.
6	ON	>0	>0	PUMP_CYCLING	PUMP_CYCLING if (TimeON > Tmin AND TimeOFF > Tmin) else VALVE_CYCLING	PASS	PASS	ON	OFF	Pump cycles ON/OFF; valve stays pass-through. Allowed only if both durations exceed Tmin.
7	ON	0<TimeON≤Tmin OR 0<TimeOFF≤Tmin	(any)	PUMP_CYCLING	VALVE_CYCLING	PASS	RECIRC (if TimeOFF>0)	ON	ON	Protection rule: refuse pump-cycling when either duration ≤ Tmin; downgrade to valve-cycling.