

## Appendix 2

Definition of how YAML objects should be formatted. Version 2. Written 1<sup>st</sup> April 2019

References to sub-sequences can be done with their names.

The ID of regimens and sequences, which is defined by the FarmBot when it receives it, will be written to internal storage automatically.

Example of a YAML file for commands:

```
""""
```

```
options:
```

```
  scale : "10 cm"           # option to set the scale of coordinates
  default_speed : 50         # option to define a default speed
  default_z : 0              # option to define a default z-coordinate value
  default_x_offset : 0       # option to define a default x-offset value
  default_y_offset : 0       # option to define a default y-offset value
  default_z_offset : 0       # option to define a default z-offset value
  other_files: ["yaml_file1", "yaml_file2      "] # you can also enter more YAML files in the command line.
  pin_aliases: {"light" : [1840, "p"], "moisture" : [204, "s"]} # optional labels for pins. "s" is for sensor, "p" is for peripheral
```

```
# The program assumes every object with a "start_time" field is a Farm Event
```

```
my_event:
```

```
  start_time: date_time
  repeat_event: {every: <<default 1>>, unit = "minutes/hours/days/weeks/months/years", until: <<date_time>>} #optional
  schedule: [{group: [optional], type: [optional], days: [], times: [], actions: <<list of actions or name of sequence>>}]
```

```
# Each list of actions can be iterated over a group or plant type, as defined in the CSV file
```

```
# The existence of a schedule field means each list of actions is repeated at the set days and times
```

```
# The list can be defined as a range. For example, "1,10,2" means every second number starting from 1 and smaller than 10.
```

```
# and "23:00,0:00,0:10" means every 10 minutes, starting from 23:00 and ending before 0:00
```

```
my_event:
```

```
  start_time: date_time
```

```

group: ["extra_water_group"]
type: ["radish"]
schedule: [{group: [optional], type: [optional], every: 4, unit: "minutes/hours/days/weeks/months/years", max: <<num_of_loops>>, actions: []}]
# If an item in the schedule list does not have group and does not have type, it defaults to the option defined above it
# This Farm Event also showcases an alternative way of defining schedules

my_event:
start_time: date_time
group: ["extra_water_group"]
type: ["radish"]
actions: [<<actions or name of sequence>>]
# If the object also has a "schedule" field, the "action" field will be ignored

# The program assumes every object without a "start_time" field but with a "schedule" field is a regimen
my_regimen:
schedule: [{group: [optional], type: [optional], every: 4, unit: "minutes/hours/days/weeks/months/years", actions: []}]
color: "gray/green/blue/yellow/orange/purple/red" #optional

# The program assumes every object without a "start_time" and without a "schedule" is a sequence
my_sequence:
color: #optional
actions: []
# Individual actions in a list can also be a string that refers to the name of a sequence
"""

```

The following are the keywords and formats we defined for single FarmBot commands such as "Move Absolute" and "Write Pin".

```

move_rel: {x: 0, y: 0, z: <<default 0 or as defined>>, speed: <<default 50 or as defined>>}
move_abs: {x: 0, y: 0, z: <<default 0 or as defined>>, x_off: <<default 0 or as defined>>, y_off: <<default 0 or as defined>>, z_off: <<default 0 or as defined>>, speed: }
if : {cond: <<see below>>, then: , else: }

cond: "x = 0" or "y > 0" or "pin1 < 0" or "sensor_pin = UNKNOWN"

```

```

find_home : [x,y,z]
wait : "time in milliseconds"
read_pin : {pin : "<<PIN_# or a name you chose>>", label: <<optional, default is pin name>>, mode : "D/A"}
write_pin : {pin : "<<PIN_# or a name you chose>>", value: "ON/OFF", mode : "D"}
write_pin : {pin : "<<PIN_# or a name you chose>>", value: "0 to 1023", mode : "A"}

```

```

to_self # Go to the current plant in a loop
to_plant # go to a plant with this (unique) name, this saves typing in coordinates by hand if you have a special plant.
e.g. we can have:
my_sequence:
  group: ["water_group"]
  actions: [{wait:"1000"}, to_self:{x_off:10}, "water_plant", to_plant:"plant_name"]

```

### Appendix 3

Definition of CSV format to be used for Farm Maps (order of the columns does not matter)

name	type	group	x	y	z
optional	optional	optional	0	0	optional