Machine state format

Definition

As of version 2.5.2

```
package main

type MachineState struct {
    PartCount        uint           `json:"part_count"`
    CycleState       string         `json:"cycle_state"`
    EStopped         bool           `json:"estopped"`
    ProgramName      string         `json:"program_name"`
    Alarms           MachineAlarms  `json:"alarms"`
    CommonVars       CVMap          `json:"common_vars"`
    FeedRateOverride *int           `json:"feed_rate_override"`
    Alerts           []string       `json:"alerts"`
}

type MachineAlarms map[string]float32

type CVMap map[uint16]float64
```

Fields

PartCount

Monotonically increasing, but can be reset to zero. If this decreases between two reports, the server will assume part counting has started over and parts will not be counted until it increases again.

CycleState

This field can have any value, but the following values have special meanings to the server:

Running

- ACTIVE
- STARTED
- Printing

Feed hold

May send a notification if the user has configured it.

- FEED_HOLD
- HOLD
- Paused

Optional stop

May send a notification if the user has configured it.

- PROGRAM_OPTIONAL_STOP
- TOOL/MDI
- Paused

EStopped

True if emergency stopped, false otherwise.

ProgramName

Ideally matches the control's program name display, but can fall back to O# if reading program names/comments isn't supported.

A program will be created on the server each time this is set to a value that a particular machine hasn't reported before.

An empty string indicates we do not know the program name. This will show up in OnTakt as "(No Program)".

Alarms

Maps alarm text to the UNIX timestamp representing the start of the alarm.

Floating-point values may be used for sub-second precision, but integral values are sufficient.

CommonVars

Maps variable number to the value of each variable.

Keys must be integers, but values can each be an integral or floating-point number, and consistency between the two is not required.

FeedRateOverride

The current feed rate override as an integral percentage (for example, 100% would be stored as `100`).

Unlike other fields, this can be set to `null` to indicate the value is unknown.

Alerts

A list of OnTakt system alerts that we should communicate to the user.

For example, "Not reporting common variables".

If a machine update provides an array of alert messages, they will be shown in a new alert in the Alerts panel.

These are different from alarms in that alarms are generated by the control, while alerts are generated by the application interfacing with the control.

Examples

Minimal

This will update the "last reported" time in OnTakt without changing any other state.

```
{
  "machine_id": 1,
  "src": "nc"
}
```

Emergency stopped, alarm present, common variables reported

```
{
  "part_count": 42,
  "cycle_state": "STOPPED",
  "estopped": true,
  "program_name": "TEST.MIN",
  "alarms": {
    "VOLTAGE FLUTTER": 1663713768
  },
  "common_vars": {
    "1": 0,
    "2": 0.123,
    "3": 255,
    "4": 1.125
  },
  "feed_rate_override": 90,
  "alerts": []
}
```

Running, no alarms, no variables reported

```
{
  "part_count": 45,
  "cycle_state": "ACTIVE",
  "estopped": false,
  "program_name": "O456",
  "alarms": {},
  "common_vars": {},
  "feed_rate_override": null,
  "alerts": [
    "Machine is not reporting common variables"
```

```
  ]
}
```