TA4 ICTA Controller

Der ICTA Controller (ICTA steht für "If Condition Then Action") dient zur Überwachung und Steuerung von Maschinen. Mit dem Controller kann man Daten von Maschinen und anderen Blöcken einlesen und abhängig davon andere Maschinen und Blöcke ein-/ausschalten.

8 Steuerregeln

Der Controller arbeitet auf das Basis von Regeln, wobei bis zu 8 Regeln pro Controller angelegt werden können.

Beispiele für Regeln sind:

- Wenn ein Verteiler verstopft ist (`blocked`), soll der Schieber davor ausgeschaltet werden
- Wenn eine Maschine den Fehlerzustand (`fault`) anzeigt, soll eine Lampe zur Fehleranzeige eingeschaltet werden
- Wenn ein Spieler in der Nähe eines Spieler Detektors ist, soll sein Name auf einem Display ausgegeben werden
- Wenn ein Minecart am Cart-Sensor erkannt wird, soll das Cart beladen werden (Schieber eingeschaltet)

Alle Regeln sollten nur so oft wie notwendig ausgeführt werden. Dies hat zwei Vorteile:

- die Batterie des Controllers hält länger (jeder Controller benötigt eine Batterie)
- die Last für den Server ist geringer (damit weniger Lags)

Zyklische Ausführung von Regeln

Diese Regeln werden vom Controller zyklisch geprüft. Ist eine Bedingung (condition) erfüllt, wird die Aktion (action) ausgeführt. Solange die Bedingung nicht erfüllt ist, passiert nichts. Auch wenn die Bedingung bei der letzten Bearbeitung der Regel schon erfüllt war und damit die Aktion ausgeführt wurde, passiert nichts mehr. Die Bedingung muss zuerst ungültig und dann wieder gültig werden, so dass die Aktion erneut ausgeführt wird.

Wie oft eine Regel vom Controller geprüft wird, kann für jede Regel einzeln konfiguriert werden. Pro Regel muss dazu eine Zykluszeit in Sekunden (`Cycle/s`) angegeben werden (1..1000).

Ereignis gesteuerte Ausführung von Regeln

Alternativ zu den zyklisch geprüften Regeln gibt es auch die Ereignis gesteuerte Ausführung von Regeln.

Ereignisse sind Kommandos, die von anderen Blöcken an den Controller gesendet werden. Beispiele sind Sensoren und Schalter. Diese senden `on`/`off` Kommandos. Wird bspw. der Schalter eingeschaltet, sendet dieser ein `on` Kommando, wird er ausgeschaltet, sendet dieser ein `off` Kommando an den Block mit der Nummer, die beim Schalter konfiguriert wurde.

Bei Regeln, die Ereignis-gesteuert ausgeführt werden sollen, muss als Zykluszeit 0 angegeben werden.

Verzögerungszeit

Man muss für jede Aktion eine Verzögerungszeit (`after/s`) einstellen. Soll die Aktion sofort ausgeführt werden, ist 0 einzugeben.

Bedingungen / Conditions

Für jede Regel kann eine der folgenden Bedingungen konfiguriert werden. Pro Regel kann aber immer nur eine Bedingung konfiguriert werden.

- `initial` - Diese Bedingung ist immer nach dem Einschalten des Controllers erfüllt und dient bspw. dazu, eine Lampe auszuschalten, um sie dann beim Auftreten eines Fehlers wieder einschalten zu können.
- `true` - Diese Bedingung ist immer erfüllt und dient bspw. dazu, eine Lampe blinken zu lassen . Dazu werden zwei Regeln benötigt. Haben bspw. beide Regeln eine Zykluszeit von 2 s, aber die erste Regel eine Verzögerungszeit von 0 s und die zweite Regel eine Verzögerungszeit von 1 s, so kann damit eine Lampe zyklisch ein- und wieder ausgeschaltet werden.
- `condition` - Hier kann abhängig von einer anderen Regel eine Aktion gestartet werden. Dazu muss die Nummer der anderen Regel (1..8) angegeben werden. Damit können 2 Aktionen mit einer `condition` ausgeführt werden. Über die zusätzlich konfigurierbare Bedingung kam mit `was not true` erreicht werden, dass bspw. eine Lampe wieder ausgeschaltet wird, wenn die Bedingung nicht mehr erfüllt ist.
- `inputs` - Damit kann der empfangene Wert `on` / `off` eines Kommandos (Ereignis) ausgewertet werden. Hier bitte beachten: Bei Regeln, die Ereignis-gesteuert ausgeführt werden sollen, muss als Zykluszeit 0 angegeben werden.
- `read block state` - Damit kann der Status einer Maschine abgefragt werden. Die Nummer der Maschine muss eingegeben werden. Mögliche Maschinenzustände sind:
  - `running` --> Maschine ist am arbeiten
  - `stopped` --> Maschine ist ausgeschaltet
  - `standby` --> Maschine hat nichts zu tun, da bspw. das Inventar leer ist
  - `blocked` -->Maschine kann nichts tun, da bspw. das Ausgangs-Inventar voll ist
  - `fault` --> Maschine hat einen Fehler. Weitere Informationen liefert ggf. das Maschinen-Menü
  - `unloaded` --> Maschinen in größerer Entfernung können ohne Forceload Block vom Server entladen worden sein. Diese sind dann nicht aktiv.

  Ist eine konfigurierte Bedingung erfüllt, also bspw. `block nummer 456 is stopped`, so wird die Aktion ausgeführt.

  Welche Maschinen welche Statusinformationen liefern, kann am einfachsten mit dem Schraubenschlüssel /Techage Info Werkzeug direkt an der Maschine festgestellt werden.
- `read amount of fuel` - Damit kann ausgelesen werden, wie viel Sprit eine Maschine noch hat (typisch 0-99 Einheiten) und mit einem Wert auf 'größer' oder 'kleiner' verglichen werden. Ist die konfigurierte Bedingung erfüllt, wird die Aktion ausgeführt. `read power/liquid load` - Damit kann die Ladung eines Akkus oder des Wärmespeichers in Prozent (Werte von 0..100) abgefragt und mit der konfigurierten Bedingung auf 'größer'/'kleiner' geprüft werden. Ist die Bedingung erfüllt, wird die Aktion ausgeführt.
- `read delivered power` - Damit kann die Strommenge abgefragt werden, die ein Generator (in ku) abgibt. Der Wert kann mit der konfigurierten Bedingung auf 'größer'/'kleiner' geprüft werden. Ist die Bedingung erfüllt, wird die Aktion ausgeführt. Da Akkus nicht nur Strom abgeben sondern auch aufnehmen, ist dieser Wert, wenn der Akku geladen wird, negativ.

- `read chest state` - Damit kann der Zustand eines TA3/TA4 Chests/Kiste abgefragt und ausgewertet werden. Kisten liefern die Zustände:
  - `empty` - Die Kiste ist leer
  - `loaded` - Die Kiste teilweise gefüllt
  - `full` - Alle Stacks der Kiste sind zumindest teilweise belegt

  Ist die Bedingung erfüllt, wird die Aktion ausgeführt.
- `read Signal Tower state` - Damit kann die Farbe eines Signal Towers abgefragt und geprüft werden. Signal Tower liefern die Werte `off`, `green`, `amber`, `red`. Ist die Bedingung erfüllt, wird die Aktion ausgeführt.
- `read Player Detector` - Damit kann ein Spieler Detektor abgefragt werden. Der Detektor liefert den Namen des Spielers in der Nähe des Detektor. Soll nicht nur ein bestimmter, sondern jeder Spielername an einem Display ausgegeben werden, so ist bei 'player name(s)' `*` einzugeben. Es können auch mehrere Namen durch Leerzeichen getrennt eingegeben werden. Soll die Aktion ausgeführt werden, wenn kein Spieler in der Nähe ist, ist `-` einzugeben.

Aktionen /Actions

Für alle Aktionen, die einen Block (wie bspw. eine Lampe) steuern, muss die Nummer des Blocks bei der Aktion angegeben werden. Pro Regel kann nur eine Aktion konfiguriert werden.

- `print to outpit window` - Bspw. für Testzwecke kann ein Text im Controller-Menü (unter 'outp') ausgegeben werden. Dies ist vor allem bei der Fehlersuche hilfreich.
- `send Signal Tower command` - Damit kann die Farbe des Signal Towers gesetzt werden. Mögliche Werte sind: `off`, `green`, `amber`, `red`.
- `turn block off/on` - Damit kann ein Block oder Maschine aus- bzw. wieder eingeschaltet werden.
- `Display: overwrite one line` - Damit kann ein Text auf dem Display ausgegeben werden. Die Zeilennummer auf dem Display (1..5) muss dabei angegeben werden. Soll der Spielername des Spieler Detektors aus der Bedingung ausgegeben werden, ist bei 'text' ein `*`-Zeichen einzugeben.
- `Display: Clear screen` - Löschen des Bildschirms
- `send chat message` - Damit kann man sich selber eine Chat Nachricht senden.
- `open/close door` - Damit können die Standard-Türen geöffnet und geschlossen werden. Da die Türen keine Nummern haben, müssen die Koordinaten der Türe eingegeben werden. Die Koordination einer Türe können sehr einfach mit dem Schraubenschlüssel /Techage Info Werkzeug bestimmt werden.
- `turn Distributor filter on/off` - Damit können die Filter/Ausgänge eines Verteilers ein- und ausgeschaltet werden. Der entsprechende Ausgang muss über die Farbe angegeben werden.

Sonstiges

Der Controller hat eine eigene Hilfe und Hinweise zu allen Kommandos über das Controller-Menü.

Einlesen von Maschinendaten sowie das Steuern von Blöcken und Maschinen erfolgt über sogenannte Kommandos. Für das Verständnis, wie Kommandos funktionieren, ist das Kapitel TA3 -> Logik-/Schalt-Blöcke in der In-Game Hilfe (Konstruktionsplan) hilfreich.

Die Hilfe existiert auch als PDF zum Drucken oder offline Lesen.