Ctek
Industrial Control

Scheduling Devices with Automation Control

Tech Note – TN045
June 5, 2020
Version 5.5

1891 N. Gaffey St. Ste. E
San Pedro, CA 90731
p. 310.241.2973
support@ctekproducts.com
www.ctekproducts.com

Table of Contents

Introduction

Ctek Automation Control supports a range of applications where it is necessary to automatically turn devices on and off on a scheduled basis. Ctek Automation Control provides a scheduling module to address these applications and supports various methods for loading complete schedules for up to 256 devices either locally or over the air.

Currently the scheduling module supports 3 unique types of schedules and each type is defined within this document. **It is important to note that the scheduler module supports all 3 schedule types concurrently which means that there can be one schedule of each type active. The scheduler provides storage for a maximum of 4000 schedule elements allowing for a single schedule type maximum of 4000 elements or multiple schedule types with elements totaling to a maximum of 4000 elements.**

Applicability

This document is applicable to the following hardware models:

- Z4500XXX
- Z4550XXX

This feature is enabled from firmware release 7.01.03

Methodology

1. The schedule file to be loaded must be created in the JSON format defined in this document.

2. The schedule file must be loaded onto the target device using:
    a. FTP
    b. HTTP post

Type 1 Mode of Operation

A Type 1 schedule is made of up of a number of totally independent elements that will be executed based on their date dependent **Start time** and discarded after their date dependent **Stop time** as seen in the sample schedule below.

A new schedule must be received in its entirety before it will be put into use. When a new schedule has been completely received, any **Outputs** in scheduled operation (on) under the existing schedule, will be terminated (turned off). The current schedule will then be deleted and the new schedule will be installed for immediate use.

Type 1 Schedule Format

The schedule is taken as a JSON formatted file that describes a schedule type and the outputs to be scheduled. Each output item describes the linear pin number of the output to be scheduled along with a start and stop date stamp. The schedule JSON object is defined below.

Schedule Type: "Type":<String>

An ID that identifies the type of schedule that is contained within the Elements array. Type must be set to: "001".

Schedule Active: "Active":<String>

Defines state of the schedule with 0 = Pause, 1 = Resume.

Pause/Resume Dashboard Button: "AddButton":<String>

Defines Digital Output pin on dashboard for pause/resume = A configured Digital Output pin number.

Schedule Status Dashboard Variable: "AddStatus":<String>

Defines Hex Input pin on dashboard for schedule status = A configured Hex Input pin number.

Schedule Size Dashboard Variable: "AddSize":<String>

Defines Numeric Input on dashboard for schedule element size = A configured Numeric Input pin number.

Schedule Items: "Elements":<Array>

Elements is an array of objects. Each object is a schedule item which describes the Output to be controlled. The 3 elements required are Output, Start_time and Stop_time.

Scheduled Output: "Output":<String>

The Output identifies the linear output number within Automation Control that this schedule item with start and stop. Valid values are 1 through 128

Scheduled Start Time: "Start_time":<String>

Start_time is string date/time stamp following ISO-8601 format: yyyy-mm-ddThh:nn

- yyyy = year (ex. 2018)
- mm = month (01 – 12)
- dd = day (01 – 31)
- hh = hours (0 – 23)
- nn = minutes (0 – 59)

Scheduled Stop Time: "Stop_time":<String>

Stop_time is also a string following the same format as Start_time

Sample Type 1 Schedule


```
{
        "Type": "001",
        "AddButton": "1",
        "AddStatus": "1",
        "AddSize": "2",
        "Elements": [
         {
            "Output":       "1",
            "Start_time":   "2018-06-15T11:09",
            "Stop_time":    "2018-06-15T12:20"
         },
         {
            "Output":       "5",
            "Start_time":   "2018-06-15T12:00",
            "Stop_time":    "2018-06-15T13:22"
         },
          {
            "Output":       "1",
            "Start_time":   "2018-07-13T15:05",
            "Stop_time":    "2018-07-13T16:10"
         }
   ]
}
```

Type 2 Mode of Operation

A Type 2 schedule is a weekly schedule of elements arranged by day of the week and further by time of day. The schedule will repeat on a weekly basis until it is disabled, removed or replaced.

A new schedule must be received in its entirety before it will be put into use. When a new schedule has been completely received, any **Outputs** in scheduled operation (on) under the existing schedule, will be terminated (turned off). The current schedule will then be deleted and the new schedule will be installed for immediate use.

Type 2 Schedule Format

The schedule is taken as a JSON formatted file that describes a schedule type and the outputs to be scheduled. Each output item describes the linear pin number of the output to be scheduled along with a start and stop date stamp. The schedule JSON object is defined below.

Schedule Type: "Type":<String>

An ID that identifies the type of schedule. Type must be set to "002".

Schedule Active: "Active":<String>

Defines state of the schedule with 0 = Pause, 1 = Resume.

Pause/Resume Dashboard Button: "AddButton":<String>

Defines Digital Output pin on dashboard for pause/resume = A configured Digital Output pin number.

Schedule Status Dashboard Variable: "AddStatus":<String>

Defines Hex Input pin on dashboard for schedule status = A configured Hex Input pin number.

Schedule Size Dashboard Variable: "AddSize":<String>

Defines Numeric Input on dashboard for schedule element size = A configured Numeric Input pin number.

Week Elements: "Elements":<Array>

Week Elements is an array of day elements. Each day element is an array representing a day of the week with "0" = Sunday, "1" = Monday, "2" = Tuesday, "3" = Wednesday, "4" = Thursday, "5" = Friday, and "6" = Saturday.

Day Elements: “0”, “1”,”2”,”3”,”4”,”5”,”6”:<Array>

Day Elements are arrays of schedule objects. Each schedule object must contain 3 elements: Output, Start_time, and Duration.

Scheduled Output: “Output”:<String>

The Output identifies the linear output number within Automation Control that this schedule item with start and stop. Valid values are 1 through 128

Scheduled Start Time: “Start_time”:<String>

Start_time is string time stamp: hh:mm:ss. Start_time defines the time of the day that the output will turn on.

- hh = hours (0 – 23)
- mm = minutes (0 – 59)
- ss = seconds (0 – 59)

Duration: “Duration”:<String>

Duration defines the number of minutes the scheduled output will remain on. Valid range in minutes (1 – 65535)

Sample Type 2 Schedule


```
{
        "Type": "002",
        "Active": "0",
        "AddButton": "1",
        "AddStatus": "1",
        "AddSize": "2",
        "Elements": [
            {

    "0": [
            {
                "Output":       "1",
                "Start_time":   "11:09:00",
                "Duration":     "30"
            },
            {
                "Output":       "5",
                "Start_time":   "12:00:00",
                "Duration":     "123"
            },
             {
                "Output":       "1",
                "Start_time":   "15:05:00",
                "Duration":     "55"
            }
        ],
    "1": [],
    "2": [
            {
                "Output":       "1",
                "Start_time":   "10:00:00",
                "Duration":     "20"
            }
        ],
    "3": [],
    "4": [],
    "5": [],
    "6": [
            {
                "Output":       "7",
                "Start_time":   "16:59:00",
                "Duration":     "25"
            },
             {
                "Output":       "8",
                "Start_time":   "00:00:00",
                "Duration":     "77"
            }
        ]
    }
    ]
}
```

Type 3 Mode of Operation

A Type 3 schedule is made of up of a number to totally independent elements that will be executed based on their **Start time** and **Duration** and all timeframes are interpreted in minutes**.** Start time is relative to time when the schedule starts execution. Schedule elements will repeat at the rate defined by **Repeat Time** until the schedule is disabled, removed or replaced. Repeat time is calculated from completion of Duration.

A new schedule must be received in its entirety before it will be put into use. When a new schedule has been completely received, any **Outputs** in scheduled operation (on) under the existing schedule, will be terminated (turned off). The current schedule will then be deleted and the new schedule will be installed for immediate use.

Type 3 Schedule Format

The schedule is taken as a JSON formatted file that describes a schedule type and the outputs to be scheduled. Each output item describes the linear pin number of the output to be scheduled along with a start and stop date stamp. The schedule JSON object is defined below.

Schedule Type: "Type":<String>

An ID that identifies the type of schedule. Type must be set to "003".

Schedule Active: "Active":<String>

Defines state of the schedule with 0 = Pause, 1 = Resume.

Pause/Resume Dashboard Button: "AddButton":<String>

Defines Digital Output pin on dashboard for pause/resume = A configured Digital Output pin number.

Schedule Status Dashboard Variable: "AddStatus":<String>

Defines Hex Input pin on dashboard for schedule status = A configured Hex Input pin number.

Schedule Size Dashboard Variable: "AddSize":<String>

Defines Numeric Input on dashboard for schedule element size = A configured Numeric Input pin number.

Schedule Start Time: "AddStart":<String>

Since Schedule Type 3 is a repetitive schedule, you can define a daily start time for initiation of the schedule sequence. This parameter defines the daily schedule start time in hours and minutes HH:MM (24 hour format).

Schedule Stop Time: “AddStop”:<String>

Since Schedule Type 3 is a repetitive schedule, you can define a daily stop time for termination of the schedule sequence. This parameter defines the daily schedule stop time in hours and minutes HH:MM (24 hour format).

Elements: “Elements”:<Array>

Elements is an array of schedule objects. Each schedule object must contain 4 elements: Output, Start_time, Duration, and Repeat_time.

Scheduled Output: “Output”:<String>

The Output identifies the linear output number within Automation Control that this schedule item with start and stop. Valid values are 1 through 128

Scheduled Start Time: “Start_time”:< String >

Start_time is string defining the number of minutes after schedule initiation when the output will turn on. Valid range in minutes (0 – 65535)

Duration: “Duration”:< String >

Duration defines the number of minutes the scheduled output will remain on. Valid range in minutes (0 – 65535).

Repeat Time: “Repeat_time”:< String >

Duration defines the number of minutes until execution repeats. Valid range in minutes (0 – 65535).

Sample Type 3 Schedule

```
{
        “Type”: “003”,
        “Active”: ”0”,
        “AddButton”: “1”,
        “AddStatus”: “1”,
        “AddSize”: “2”,
        “AddStart”: “13:10”,
        “AddStop”: “19:00”

        “Elements”: [
        {
                “Output”:          “1”,
                “Start_time”:      “15”,
                “Duration”:        “15”,
                “Repeat_time”: “30”
        },
        {
                “Output”:          “5”,
                “Start_time”:      “0”,
                “Duration”:        “5”,
                “Repeat_time”:  “45”
        },
        {
                “Output”:          “6”,
                “Start_time”:      “30”,
                “Duration”:        “10”,
                “Repeat_time”:  “60”
        }
        ]
}
```

Type 3 Schedule Deletion

To delete an executing Type 3 schedule, post a new Type 3 schedule containing 1 Element. The Element must be configured with Duration set to a value of zero.

Posting a Schedule Remotely

New schedules can be manually loaded in the specified area or they can be pushed remotely. To load a new schedule remotely simply post a JSON schedule to the device. Schedules can be sent using an HTTP Post to the controller’s web server at http://mydevice.com/cgi-bin/rtu_configure.cgi/load_schedule

This URL on the controller will only accept HTTP POST request with JSON in the body. Content type should be set to application/json. To post the schedule the controller must have a user

account with those privileges. The user can be created through the web administration interface on the controller.

The JSON in the body of the HTTP POST request must be an object that includes the properties username, password and schedule as seen below. The schedule element is just the JSON schedule object described in the previous section.

Sample JSON for HTTP POST

```
{
  "username":   "name",
  "password":   "pass",
  "schedule":
  {
          "Type": "001",
          "Elements": [
           {
              "Output":    "1",
              "Start_time":          "2018-06-15T11:09",
              "Stop_time":           "2018-06-15T12:20"
           },

           {
              "Output":    "5",
              "Start_time":          "2018-06-15T12:00",
              "Stop_time":           "2018-06-15T13:22"
           },
            {
              "Output":    "1",
              "Start_time":          "2018-07-13T15:05",
              "Stop_time":           "2018-07-13T16:10"
           }
     ]
  }
}
```

IMPORTANT Notes Regarding Remote Posting of Schedules

As stated previously schedules are limited to 4000 elements. When posting schedules, overall file size is limited to 100K bytes. This size is sufficient for 4000 elements as long as unnecessary formatting spaces are removed. Programs generating schedules for remote posting should flatten JSON output to stay within this size restriction.

Retrieving a Schedule Remotely

The current schedule of a remote device may be retrieved by posting an empty JSON schedule structure to the device. This will be interpreted as a request for the current schedule. Schedules can be retrieved using an HTTP Post to the controller's web server at http://mydevice.com/cgi-bin/rtu_configure.cgi/get_schedule.

This URL on the controller will only accept HTTP POST request with JSON in the body. Content type should be set to application/json. To post the schedule the controller must have a user account with those privileges. The user can be created through the web administration interface on the controller.

The JSON in the body of the retrieval HTTP POST request must be an object that includes the properties username, password and schedule as seen below.

Sample JSON for HTTP POST

```
{
  “username”:  ”name”,
  “password”:  ”pass”,
  “schedule”:
  {
    “Type”: “001”,
    “Elements”: [ ]
 }
}
```