# DE-4000 SCRIPTING REFERENCE MANUAL

DE-4000 Series Configurable Safety Shutdown and Control System

Form DE-4000 OCM 02-22

altronic

All rights reserved © ALTRONIC, LLC 2022

This manual contains information on the operation and configuration of a DE-4000 Safety Shutdown and Control System. This manual supplements the DE-4000 Safety Shutdown and Control System Installation Instructions, Form DE-4000 II.


All rights reserved © ALTRONIC, LLC 2022

# Table of Contents

DE4000 Lua Script API ..... 1


All rights reserved © ALTRONIC, LLC 2022

All rights reserved © ALTRONIC, LLC 2022

# DE4000 Lua Script API

---

#### create_param("index",default,"catergory","description")

- creates a user configurable parameter
- parameter is stored as `index`,
- default value(If not changed by user) is `default`
- parameters will be grouped on the Global/Params page by `category`
- `description` is text to describe the parameter to the user

**Example:**

```
create_param("NumEngCyl",8,"Engine Params","Num. of Engine Cylinders")
```

---

#### get_channel_val(terminal,channel)

- returns current value of analog input `channel` on terminal module `terminal`
- return value type is numeric

**Example:**

```
local sp = get_channel_val(1,5)
```

> reads value of Suction Pressure from Terminal Module #1 , Input #5

---

#### get_gbl("index",default)

- returns global config setting stored under `index` or returns `default` if not defined

> **note:** get_gbl is used to retrieve global CONFIGURATION settings that are typically set when the system is configured and do not change as the system is running. If you want to set and retrieve global STATUS variables use the get_sGbl() and set_sGbl() functions >If you want to create and read virtual channels use the set_sVirt() and get_sVirt() functions.

**Example:**

```
local nt = get_gbl("NumTerm",1)
```


All rights reserved © ALTRONIC, LLC 2022

> gets the number of terminal boards installed in the system

---

#### get_param("index")

- return either the default value or the user configured value of the parameter `index`

**Example:**

```
get_param("NumEngCyl")
```

>gets the configured parameter for number of engine cylinders

---

#### get_rpm(channel)

- reads the RPM input `channel` in units of revolutions per minute

> **note:** valid channel numbers are 1 - 10(2 channels per board, up to 5 terminal boards)

Each Terminal Module has 2 RPM inputs (RPM1 and RPM2)

- Terminal Module **#1** RPM channels are **1,2**
- Terminal Module **#2** RPM channels are **3,4**
- Terminal Module **#3** RPM channels are **5,6**
- Terminal Module **#4** RPM channels are **7,8**
- Terminal Module **#5** RPM channels are **9,10**

**Example:**

```
local engineRPM = get_rpm(1)
local turboRPM = get_rpm(6)
```

> Read RPM1 channel from terminal module #1 and read RPM2 channel from Terminal module #3

---

#### get_sGbl("index", default)

- If `index` is defined in the global status table then it returns the value associated with `index`
- If `index` is not defined and optional `default` is provided then returns `default`


All rights reserved © ALTRONIC, LLC 2022

>**note:** It is recommended to always provide a default value when using this function

**Example:**

```
local cp = get_sGbl("calculatedPressure",0)
```

> get the previously stored value "calculatedPressure", Returns `0` if not found.

---

#### get_state()

- returns the current engine state(possible values currently 0 - 10)

**Example:**

```
local engineState = get_state()
if engineState > 7 then
    set_timer("WarmupTimer",1000)
end
```

---

#### get_sVirt("index")

- returns the value of virtual channel `index` or returns `default` if the virtual channel does not exist.

**Example:**

```
local tl = get_sGbl("timeLimit")
local et = get_sVirt("ElapsedTime",0)
if et > tl then
    set_sGbl("timeExceeded",true)
else
    set_sGbl("timeExceeded",false)
end
```

>Gets the value of virtual channel ElapsedTime and set value of status global "timeExceeded" if ElapsedTime is greater than status global "timeLimit"

---

#### get_time()

- returns the UNIX "epoch" time (Defined as the number of seconds elapsed since Jan 1, 1970)


All rights reserved © ALTRONIC, LLC 2022

**Example:**

```
local startTime = get_sGbl("startTime",0)
if startTime == 0 then
    local currentTime = get_time()
    startTime = currentTime
    set_sGbl("startTime",currentTime)
end
local et = get_time() - startTime
set_sVirt("ElapsedTime",et)
```

>Stores current time if first time through, otherwise calculate elapsed time

---

#### get_timer("index")

- returns 1 or 2 values
- First return value(Boolean) is **true** if timer is active(counting down) or **false** if timer is expired or has not been set yet
- Second return value is the number of seconds remaining or **-1** if timer is not active or has not been set yet

**Example:**

```
if not get_timer("myTimer") then
    set_sGbl("timedOut",true)
else
    set_sGbl("timedOut",false)
end
```

> if timer is expired, then set global status "timedOut" to **true**

```
local active,remaining = get_timer("myTimer")
if not active then
    set_sVirt("timeRemaining","Expired")
else
    set_sVirt("timeRemaining",remaining)
end
```

---

#### getStateLabel(state)

- return the label for the engine state corresponding to the parameter `state`


All rights reserved © ALTRONIC, LLC 2022

**Example:**

```
local stateLabel = getStateLabel(get_state())
local active, remaining = get_timer("myTimer")
if remaining > 0 then
    stateLabel == StateLabel.." "..remaining
end
set_sVirt("Countdown",stateLabel)
```

---

#### set_sGbl("index",value)

- store `value` in the global status table under `index`
- value can be a number or string but if storing a boolean use the **tostring()** function

**Example:**

```
local mpe = false
local sp = get_channel_val(1,5)
if sp > 15 then
    mpe = true
end
set_sGbl("minPressureExceeded",tostring(mpe))
```

> store boolean value **minPressureExceeded**

---

#### set_sVirt("index",value)

- sets a virtual status channel with channel name `index`

> **Note:** Once you create a virtual channel, you can add that channel to the dashboard using the channel name `index`

**Example:**

```
local sp = get_channel_val(1,5) --suction pressure
local dp = get_channel_val(1,6) --discharge pressure
local diffPress = dp - sp
set_sVirt("SuctDischDiff",diffPress)
```


All rights reserved © ALTRONIC, LLC 2022

> calculate the differential between suction and discharge pressure and assign to virtual channel

---

**set_timer("index",secs)**

- activate timer `index` and set countdown time to `secs`

**Example:**

```
set_timer("myTimer",300)
```

> create timer `myTimer` and start countdown time to 300 seconds

---

From:
https://altronic.a2hosted.com/ - **Documentation Wiki**

Permanent link:
**https://altronic.a2hosted.com/doku.php?id=documents:de4000:de4000script&rev=1642108314**

Last update: **2022/01/13 16:11**


All rights reserved © ALTRONIC, LLC 2022