

MYC Logicdevice

Preliminary

Author: DK1RI

Version V01.03 20250309

This paper is published in <https://github.com/dk1ri> as well

Introduction

This paper describes the logic-device

For more details of the MYC system please check the reference.

The program [17] will be integrated into the commandrouter. Corrections and enhancements will be found there [7]

The program is in development. It is for test purposes only! If you want to test the program [17], you should be familiar with the MYC system. Wrong data input will crash the program. The CR, which will deliver the data, deliver correct data only. There may be errors in the program as well.

Definitions and formats

see <https://dk1ri.de/myc/Definitions.txt> or <https://dk1ri.de/myc/Definitions.pdf>

The Idea

The idea is similar that, what IFTTT (If This Than That) implemented same years after the first edition of this document was published: create and execute rules, that exist within a device and between different devices.

Maybe, that IFTTT can be adapted to work with a MYC system.

Rules for the devices are given with the device but the program to create the rules between the devices (RU), is not ready. They must be created manually now.

The RU must generate rules and send them to the LD, which can be done by AI (in future).

Description

The LD can be described as a state-machine. It gets inputs (command with parameters) from the SK by the CR, handles them using a defined algorithm and the internal state, and send the result back to the CR, which forward them to the devices.

The algorithm uses the rules from the RU and the devices.

It also gets all answers from any attached device to update its state. Data from and to SK have a different prefix than data from devices because one data channel is used only.

The LD has the status of switch, range and array memory commands but do not store other memory content.

The internal state reflect the state of the MYC system. If a state of a FU change without a command, the LD get an info, which will modify the internal state of the LD.

The LD get rules from the RU and the rules of the device by the CR; or uses the data of the CR.

The LD test programm will have the basic &H00 &H01 &H02 and &HFD own announcement only.

These command &HFD without prefix are used by the CR to check that the LD is alive: The CR answers with \$HFD04. The &H01 &H02 commands are used to transfer rules data.

This is not necessary anymore, when the LD is integrated to the CR.

The LD do not check input data. The SK and devices should send data with correct number of parameters with correct parameters. The CR will ignore not valid commands.

The LD may provide some special functions as providing time data. This is not yet defined.

I did not found a state machine program, that can be used as LD, and most of them are not open source. The effort to write a program for the LD is less that expected.

If the program is finished, it will be part of the CR. This simplifies data transfer and will be faster. The detailed description of the program can be found with the program.

Handling (test program [17])

The CR analyze the data from SK or FU, split them to commamndtoken and parameters and send the data as string to the LD. So the LD do not need to analyze the data again and send the data back to the CR if not blocked. But the LD concatenate the splitted data to one string. This avoid to analyze the direct commands; these are given in the same format.

A sample announcement file is given for test, The rules for test must be uncommented.

A command input start with "s": from / to skin or "u": from / to device.

The data of the commands must be in hex. Parameters are separated by the space character and should have correct length. The command is ended with a "t"

Example:

```
d01 0202 t
```

When the LD is integrated to the CR, data exchange is done by an internal variable.

Copyright

Dieses Dokument darf unverändert kopiert werden.

Die Ideen in diesem Dokument unterliegen der GPL (Gnu Public Licence,V2) soweit keine früheren, anderen Rechte betroffen sind.

Die Verwendung der Unterlagen erfolgt auf eigene Gefahr; es wird keinerlei Garantie übernommen.

This document can be copied without changes.

The ideas of this document can be used under GPL (Gnu Public License, V2) as long as no earlier other rights are affected.

The usage of this document is on own risk, there is no warranty.

Reference

- [1] <https://dk1ri.de/myc/MYC.pdf> (german)
- [2] <https://dk1ri.de/myc/MYC.en.pdf>
- [3] <https://dk1ri.de/myc/Description.txt> or <https://dk1ri.de/myc/Description.pdf>
- [4] <https://dk1ri.de/myc/commands.txt> or <https://dk1ri.de/myc/commands.pdf>
- [5] https://dk1ri.de/myc/Reserved_tokens.txt or https://dk1ri.de/myc/Reserved_tokens.pdf
- [6] <https://dk1ri.de/myc/Rules.txt> or <https://dk1ri.de/myc/Rules.pdf>
- [7] <https://dk1ri.de/myc/commandrouter.txt> or <https://dk1ri.de/myc/commandrouter.pdf>
- [8] https://dk1ri.de/myc/Rules_device.txt or https://dk1ri.de/myc/Rules_device.pdf
- [9] <https://dk1ri.de/myc/skin.txt> or <https://dk1ri.de/myc/skin.pdf>
- [10] <https://dk1ri.de/myc/logicdevice.txt> or <https://dk1ri.de/myc/logicdevice.pdf>
- [11] <https://dk1ri.de/myc/Definitions.txt> or <https://dk1ri.de/myc/Definitions.pdf>
- [12] https://dk1ri.de/myc/spec_version.txt or https://dk1ri.de/myc/spec_version.pdf
- [13] <https://dk1ri.de/myc/webserver.txt> or <https://dk1ri.de/myc/webserver.pdf>

- [14] <https://dk1ri.de/myc/ki.txt> or <https://dk1ri.de/myc/ki.pdf>
- [15] <https://dk1ri.de/myc/communication.txt> or <https://dk1ri.de/myc/communication.pdf>
- [16] <https://dk1ri.de/myc/Security.txt> or <https://dk1ri.de/myc/Security.pdf>
- [17] <https://dk1ri.de/myc/logicdevice.zip> or <https://dk1ri.de/myc/Security.pdf>