Hinweis für die Textformat-Version; um diese Version im PDF Format zu lesen, laden Sie bitte https://www.dk1ri.de/myc/Description.pdf

MYC System Description, Handling and Environment

Author: DK1RI
Version V01.08.04 20251205
This paper is published in https://github.com/dk1ri as well

Introduction

This paper describes the basics of a MYC system
For more details of the MYC system please check the reference.

Definitions and formats

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

Some explanation

The MYC protocol is used to control electronic devices as radio, TV, home automation and also control robotics and other mechanical devices.
It also can control any program, which has a MYC protocol interface.
It can be used as a ”glue” to combine so called microservices.
It can control data stream devices and also handle streams in a very simple manner, but streams are not the major focus of the MYC protocol.
The MYC protocol is a pure semantic protocol.
It has no acknowledge, error correction, encryption, routing, connecting and sequencing and simple user management only, This must be done in the other communication layers. But the MYC protocol can control devices with this functions.
MYC data packets have no header, no end marker. All data packets have a well defined length; so begin and end can be found.
There are commands with or without answer and infos, which are sent without command.
There are some meta commands to control the system and for simplification.
The basic command and the individualization commands are mandatory command for any MYC device.
Any device will ignore unknown or non-valid data immediately after detection. The next byte is handled as a start of a new command / answer / info.
All commands with the parameters a device can handle are described by command-announcements using an announcement list.
Announcement list of all devices are connected together via the CR, which routes the data to other devices or handles them by itself.
A MYC system can be part of another MYC system.

A MYC system contains a set of devices with a MYC protocol interface. Controlling and controlled device can be a hardware or programs. Controlling devices are called “skins”. Controlled devices are called a “func”.
There are three more devices in a system, which exist only once in a MYC system.

Skin devices, SK, can be multiple in a MYC system:
The skin device initiate commands it is a controlling device.
Some skin devices can handle one user, it may be a human interface. Other SK - like a web server –

may handle more than one human interface.
A simple skin (SS) is handled in a different way than normal skins. Those devices are simple switches or actors.
For more details see [9]

Functions, Func, FU, can be multiple in a MYC system:
A FU is a controlled devic and will respond to commands with an action or with sending an answer back.
Info can be sent by a FU without a command and is routed to the LD and SK. This may be useful for metering info eg, or if a function device is operated manually. An info informs the LD about changed values or errors.
Because some transmission protocols (like I2C in normal mode) do not allow slave devices to send by themselves, the CR must poll the devices in this case with special command and forward info to the LD / SK.

Command-router CR, one in a system:
This device connects each device and collect their individual announcements, generate a full announcement-list with all (unique) command-tokens and send them on request to other devices.
If it get a command out of the full list, it will translate it back and route to the appropriate device.
The command-router do not accept any command except own reserved commands. All other valid commands are routed (or ignored otherwise).
The command-router must know all devices belonging to its system and have the connect information.
Depending on the implementation of the CR, it get the connect information from a device (ie some database), or it knows the connect info by itself.
As an option the CR do not send the list of all commands of all devices but will only send the basic announcements. A requesting SK calculate the tokens of the missing commands, if it knows the commands already. If these devices do not know the details of the individual devices, it will send a command for details or ask for the complete list.
Simple SS with a limited predefined command set (as switches) will not ask for a full list.
The CR also checks for new devices and check for devices not active anymore from time to time.

Rules-device, RU, one a MYC system
This device defines the rules between the different FUs and other system-wide rules and send them to the LD.
It also handles the user login.

Logic device LD, one a MYC system
Most commands of SK will be received via the CR by the LD, which modifies them according to the rules if necessary and send them via the CR to the destination devices to do the action.
The logic device collects the rules of the individual devices and the rules of the system from the RU.
The LD knows the state of the system at any time, as long as it is affected by the rules.
The LD should know, what to do, if a device fails.
In the existing version of the system the LD send all commands / infos via the CR. Later versions may send them directly to the FU and SK.

The implementation of the logic device may be differnt. In principle it is a configurable state machine. If available a “standard” state machine van be used. Actual implemtation is a python program.

Start up behavior

Any device may start up and shut down at any time.
As long as the CR is down, all data are lost. Devices except FU will send busy commands to the CR and will detect, that there is no answer, go to a startup mode and stop normal sending. The system wide timeout is 2 seconds.
For 2 seconds after starting, the CR will have no other communication. This ensures, that SK, LD and RU fall to startup mode.
At start up, the CR will find a list of all known devices. It will communicate with those only and check this list later from time to time.
The CR may have the announcements of all devices from previous connections.
If an announcement for a FU is missing, the CR will ask for this by sending a 0xxxf0 command.
It will ask the actual announce-list of lower level CR as well by sending a 0x00, will get the 0x00<announcement> answer with correct commandtokenlength and then send a 0xxxf0 command.
Then it will build the full announcement-list. Announcements of SK and higher level CR are not included and so no device can send to them except the CR. Announcements of SK are included but not sent to SK. There are gaps in the numbering instead.
Rules from the rules device may have influence on the full list as well and will be sent to the LD.
To inform the device, that the CR is ready, it send a 3 byte 0xfd0x010x04 ready info to SK, RU, LD and higher level CR.
All devices, which require the complete announcement-list, will send a 0x00 command and after getting the answer with correct commandtokenlength send the 0xxxf0 or 0xxxf1 command.
The CR will answer with the actual list.
Each user has an individual SK. But if the SK is implemented by a web server eg, this web server may deliver more than one SK. The web server therefore may need a multiple user access to the CR via one interface. The web server must request a channel number for each SK by sending a 0xxxfd<n> via channel 0. n is the position of <USERNUMBER> in the announcement. The answer is a channel number > 0, which will be open until USERTIMEOUT. All communication with interfaces supporting multi-channel is done by preceding the communication by the (one byte) channel-number. Interfaces supporting multi-channel have a “M” at the end: as USBM in the INDIVIDUALIZATION announcement of the CR.
Other interfaces have no channel-number.
The CR do not keep track on commands of the different interfaces / channels. So by default all interfaces will get all answers and infos. Multi-channel interfaces will get it via channel 255 only and the interface must handle the distribution to the SK.
If this is restricted by 0xxxf9 ANSWERS, the CR will send individual infos / answers.

The rules device will send all actual rules and additional commands to the CR and LD, if necessary.
The LD must get the actual state of the complete system and will ask the devices, or set a default as defined by the rules. The LD will not send own commands, It may respond to a busy request 0hfd with 0h04.
Because all answers are sent to all SK as well as default, SK has the actual state as well.
This ends the system startup procedure.

For FU there is nothing special at startup, it will start working immediately
SK and higher level CR can check, whether CR is ready by sending a 0xxxfd01 command. This also avoid their timeout. If the CR do not answer, or after a SK startup, the SK will send a 0x00 (CR basic command) and then the 0xxxf0 command as described above.
The RU at start up will check, weather the CR is active and send a set of rules to LD on request.

The LD at start up will check, weather the CR is active and ask for the rules, check the system status for all devices and send the status to SK and higher level CR by info.

If it is necessary for the CR to create a renumbered list, it will shut down and force a new startup procedure.
When the CR create a new command-list, with added or removed devices without renumbering it sends a 0xxxfd info to RU and LD. If RU and LD got their new list, CR send the 0xxxfd info to SK and higher level CR. So during the meantime dropped commands are ignored by the LD, added are not known yet.

Data transfer and data flow

The MYC protocol send commands / properties with a defined length. The length is defined by the used command-token and the corresponding announcement.
The protocol has no limit of the size of the transferred data. There is a limitation due to the used data pointers: most FU: 256 byte or less (1 byte pointer), some FU: 65545 byte (2 byte pointer). By designing a FU, ist must be taken into account, that some radio interfaces do not allow high data volumes!
If an answer is sent by a FU, the length is defined as well, because the answer is preceded by the command-token.
It is not necessary, that the CR wait for answers, because any answers and info can be associated to a command. But for polled devices the CR do not know anything about the FUs buffer. So in this case the CR will request the answer before sending the next answer command. The time for requesting the complete answer is limited to 1 second (default).
Multiuser mode for the interface is not part of the MYC protocol but implemented in this way by the CR.
If the interface is in multi user mode, command, answers and info are preceded by the one byte channel number. 0 is the administrative channel for the interface and 0xff the answer / info channel to all. The SK must distribute the answers and info by itself. By default all answers and infos are sent via channel 0xff.
Higher level CR will send commands and receive info and answer as info. They can request multi user as well.
SK and higher level CR will not wait for answers.
Data format for answers and info is identical.

The CR will communicate with FU using the original command-token of the FU. Communication with other devices use the translated token of the full announcelist.

The LD get all accepted commands from SK and higher level CR, also all infos and answers to SK. It will send commands to the devices and answer / infos to SK via CR. Because the CR cannot distinguish between commands and answers / info, these are preceded by a one byte direction identifier: `fe: LD -> CR → device, ff: LD → CR -> SK`

There is no limitation for the number of devices and SK by the MYC protocol. But a real CR may get more data from the SK than the CR can handover to the devices. So the CR may stop data data input by sending a STOP command (0xxxf9) to the SK, if a limit is reached. The actual limit is 10kByte.

Limitation due to hardware

The realization of the MYC protocol on real hardware / software will have some limitations mainly for the FU. CR and LD have some limititions as well – mainly due to limited memory – but they should work with all FU.
- Most FU use 1 byte commandzoken, some 2 byte. CR / LD can handle 4 byte.
- Command length is limited to 256 byte for most FU. CR /LD can handle more.
- Some FU can handle one command in a data packet only. After finishing the command, the FU delete all other input. The CR has different data streams for different SK and will send data to FU, if a command is complete. If different commands should be sent to the same FU, it will wait a while. In cases, when the FU has a timeout, commands may be lost.

Security

Security is one of the most important aspects of a control system.
MYC is a OSI layer 7 protocol and defines the commands and their methods only. Nothing about security is defined. Nevertheless there are some aspects:
There is no readable text (except when strings are transmitted), all data are binary and someone reading the transmitted data, must know the connected device and their announcements as well.
The system is hierarchical: Because a FU checks any byte and discard any unknown data, it is not easy to insert meaningful bad code. The FU should be designed in a way, that the configured interface is active only. So the CR is the only interface and the CR will discard any unknown answers.
The other OSI layers can be chosen by demand and they will define the system security.
The following should give some rules how to increase system security.
Funcs, RU, LD
Funcs are designed to accept their valid command with their valid parameters only. Any other data are ignored immediately and may produce an error message.
This is also valid for subsystems as FS20, ZWAVE and others connected via a MYC interface.
The devices are connected to the CR (up to now) by cables. It it assumed, that this system works in a save environment.
If this cannot be guaranteed modified devices with built in encryption must be used (not available yet). The command-router also must communicate with this format. The same is valid for wireless devices.
For RU and LD the same apply.
If these devices run as programs on the same computer as the secure message transfer usually is no problem. Otherwise encryption is recommended.

Command-router CR and skin SK
The CR communicates with the SK usually via internet. So at this point ssh and login processes must be provided. The MYC protocol define a command for login but says nothing about the implementation. The CR also communicate by MYC commands only and ignore all non valid commands and parameters.
The login to lower level systems can also be used.

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