

MYC, an additional (not only) shack / amateur-radio control

There are a lot of controls and programs for control of radios and radio equipment
But unfortunately most of them do not work together. This is a proposal how this can be changed.

- MYC is a semantic protocol for the transmission of control signals, mainly not for transmission of streams
- In a MYC system devices, which understand the protocol can be added and removed when the system is running.
- All devices can announce their commands. Additionally rules for interaction of the devices are announced.
- Different methods for the user-interface can be used.
- The MYC protocol is designed to work between devices and programs. Data transmission is as short as possible. The commands are not simply readable.

Preface

I have a program for digital modes with an integrated logbook. I do not like this logbook; usually I use a different one. This program can control transceivers, but I use different transceivers for HF and VHF and sometimes even a second one for HF. So I have to reconfigure the program every time. The rotator for HF has no computer control at all and the rotator for VHF has remote control using infrared. In fact, nothing works together and I need a more universal solution.

This problem really is not a new one and there is a lot of equipment for control of the shack equipment.

Most solutions are quite special: not supported and new equipment cannot be used.

With a MYC System you can control almost everything that can be controlled and has a documented protocol. Not radio equipment only. You need to build the interface, for which some prototypes are existing and attach it to the system. Maybe, that there is some more work... but more on this later on.

The system described here is not a device or program ready for use but a system, that must be built individually. Therefore the name MYC for MY Control. The system can be adapted individually, some parts, even essential parts, are still missing. Some parts can be built quite easily, some will be very difficult.

The concept and an additional protocol

A MYC system contains a command-router and some devices, which exchange data with the command-router via the MYC-protocol. Each device can send commands and infos to other devices or receive them. The command-router routes the commands to the correct devices.

Each device has a “fixed” connection to the command-router. The MYC protocol is a semantic protocol . Like CAT protocol it describes functions only and there are no commands for connect, route etc. In most cases this is not a problem: a MYC usually is a collection of equipment at one place, which are connected via fixed cables or paths, and where the configuration do not change very often. Nevertheless there can be a device, which can establish a connection via internet or all devices communicate via internet.

Different MYC systems can work together in multi level systems: a command-router would work as a device as well and communicate with a higher level command-router.

Unfortunately there is a additional new protocol for data exchange between the devices necessary. I do not know a suitable free one which can be used that way.

Devices

A device is a radio for example which accept the MYC protocol. Now it is a radio with a small interface, which translates the MYC protocol to the CAT protocol of the radio.

The device send an “announcement” with a list of all commands it will understand to the command-router on request.

A universal usable MYC interface should translate all functions of the attached equipment.

There can be two ways to achieve this. The more simple way is to make a simple one to one translation. This works quit well in most cases. But there is a problem when there are no CAT commands for receiving the actual status of the attached equipment. You can tell the MYC interface of a transceiver which are controlling the memory canals with + and -command only, to select canal 25 and the interface would send a series of + or – commands. This implies, that the interface knows the correct status all the time. The interface would know a command, which is not translated directly to a CAT command..

Otherwise a MYC system can have a frequency management as well and the frequency memory function of the radio will never be used. The Interface would need less commands than the CAT protocol. But you never know, how the final system would look like. So all CAT command should be translated.

A program with a MYC interface can be a device as well. A logbook program may get the frequency from the transceiver and call and mode from the decoding RTTY program.

Some devices will not only react on commands but also send infos without command. Frequency values for example, if the transceiver is operated manually.

There are also devices, which can send commands, a user interface program for example. Those programs must have the behavior of a device and send its commands via the command router. They are called skins.

If you want to use a existing user program in a MYC system, you need a interface (hardware or program), which behave as the controlled original device on one side. On the MYC it will send so called “required” commands, which are somewhere else translated to commands, a device will

understand.

Command-router

Every device will get the commands by the command-router. The command-router knows, which commands a device can handle. The commands are identified by a command-token (simply a number) and the command router produces a new list of command token of all command of all connected devices. This list it sends to the skins, which use these commands.

If the command-router gets commands, it will forward them to the appropriate devices, except all announce commands and some other reserved commands, which it will respond by itself.

Logic device

Sometimes things are a little bit more complicate, because there is another task within the complete system.

If for example the user interface program only has one frequency control for HF and VHF but the hardware has two different radios, the command-router must send the PTT command to the correct radio. May be that antennas and audio must be switched as well and the command-router must tell the user interface, that for VHF other controls are necessary than for HF.

This “logic” is handled by the “logic-device”. The logic-device and the command-router will exist in a system only once.

The logic device gets all the commands and infos, which are send to the command-router, works on them and send them back to the device via the command-router. So all command and infos have a deviation via the logic-device.

There is another kind of logic, which must be handled.

The most simple case is, that any command will work at any time. But can you really change the band or mode during transmitting? Or do you really want this? Or can you transmit outside the amateur radio bands? Without logic the user interface would show this, but the transmitter would simply not follow.

The device can tell the logic-device these device specific conditions by rules.

Rules, which describe the interconnection of the devices can be given by an additional system-device. It knows, how antennas, audio and transceivers are connected and form a set of rules out of this

Some transceivers send a “not ok” messages when receiving wrong or not valid commands, but not all do this on all error conditions. Therefore the MYC Interface must take care, that by proper announcements and rules only valid commands are have another error handling.

It may be quite tedious to find all error-conditions of a system. If you have 5 devices with 10 commands each you have to check whether each of the 50 commands influences each of the other 49 commands. But usually the situation is much more simple. But if you have a mixture of radios other house equipment, there are combinations like “lower antenna depending on wind speed” or “switch off equipment, when none is at home” and a lot of other conditions.

It is necessary, that the logic-device always knows the actual status of the whole system, and remember the state. The output commands depend on the incoming commands, the actual state and the rules. The logic-device can be seen as a so called statemachine. There are programs to build up a statemachine on a PC; but it has to be clarified, whether they are available for free.

The rules are provided by the rules-device. They can be programmed directly to make it simple, but a configuration file would be more smart. Even better would be a graphical editor, which would create such a configuration file.

How the logic device can be realized is not decided yet.

As a summary the logic-device will get all all commands, modifies them if necessary and send the back to the command-router for further routing to the devices.

The Skin

Today a user program (skin) works as follows: I load the program, but I do not like the GUI too much (that happens quite often). I do not like the arrangement of the knobs, I like sliders more than knobs; instead of a button panel I would prefer a knob, if this is seldom used. Some functions of the menu want to see on the main screen and colors are also not nice.

Using a MYC System the user interface program (a browser with a web server or another program) could get all possible functions from the command-router. I would associate knobs, sliders etc to the function arrange everything as I like it, store my configuration and would have my best user interface.

Using a smart phone, the GUI would be different from the GUI on a tablet or notebook. Perhaps I would concentrate on the most important functions.

It is not clarified now, whether such a program exist now.

Now the most simple method you can input the command using a terminal-program. A self programmed website is another solution.

Multiple User interfaces

It is allowed, that more than one user-interface is active and send commands. This is no problem, because the display of the user-interface get its data from the logic-device and the logic-device has knowledge of status of the system at any time. If you want to use a existing user interface or radio control program, which do not support the MYC protocol, you will need a MYC-device as interface, which can be hardware or software. The MYC device must emulate the radio, the program want to control. Usually the interface would be a virtual COM Port. May be, that not all control programs poll the hardware for correct data, but in this there is no difference whether they are connected to the real radio or the emulation.

This is a way to connect any program to a MYC system, even if they do not have a MYC interface.

Data transmission

There are no rules for data transmission between the devices and the command-router. RS232, I2C, SPI are planned. Others are possible, especially Ethernet.

The different protocols have advantages and disadvantages:

RS232 is a 1:1 connection. You need an interface for each device. This should be used only, if you have few devices.

For using I2C it is necessary, that all devices on the bus are connected to power. The connection length is limited and noise immunity not very high. If you have devices as slaves only, the command-router must poll the devices for info. A multi-master system is more difficult to handle. I2C can be used for a local system, where the device are not too far from each other.

Similar limitations are for SPI with higher data rates.

Ethernet is bidirectional, galvanic decoupled and fast. If you use modules ready to use, the programming may be simple. Otherwise this solution is not cheap.

The MYC protocol cannot initiate a connection. This must be handled separately.

If the connection between device and command-router is via I2C you need addresses for them. The addresses are absolutely different addresses than the command-tokens of the MYC protocol.

The MYC protocol is a protocol between computers, therefore commands are not readable and as short as possible to allow slow data-transmission. The MYC protocol has no possibility for data correction, encryption and so on. This must be done in a lower communication layer.

Commands

Each device will understand and handle a set of commands. Command are transferred as 1, 2 or 4 byte number depending on the number of commands. Each command is member of a command-type as “switch”. The command-type defines the syntax of the command and how the command is displayed on the skin. I was really surprised, that there are not many command-type at all, even if a MYC system is not only working with radios, but with almost any other application. You will find the list of command-types and the description in [4]

Individualization

The announcements describe a device type. The command-router can associate the command to devices. If there are more than one devices of the same

type, the devices need a individual identification so that the user interface can distinguish them. So the devices may get a individual number and name before regular use, and, if necessary , some other information as connection information. This individualization can be done in a MYC system. with a command and must be done before regular use. The description of this and other reserved command token are in [5]..

Possible applications

My first main application is the radio shack. But control of house functions is possible as well . For tv/radio there is a framework for a infrared remote control interface. The configuration must be done individually depending on the equipment.

Hardware and usage

The system described here is a modular system with different parts, which can be constructed more or less independent. It is not necessary, to realize all parts to build up a system. In the most simple case you only have a system with MYC commands. For this you only need the interfaces for the equipment and a interface for input with a terminal program. Other devices as command-router logic-device and other special devices you can add later on.

Next task would be a command-router. A raspberry pi most probably would work.

More difficult is a web site adapted to your need.

At the end you can realize the logic interface and the a universal user interface.

What is ready?

I have realized about 20 different devices, ready to use (end 2016). They showed, that a MYC system would work.

I work on the command-router now.

Conclusion and what is going on

I described here not a equipment ready to use, but a concept for a system for further development.

I have not too much ready now and it is not easy to realize a ready working system. If other people also support the system, system realization will be more easy in future.

The project is distributed in <https://github.com/dk1ri> as well

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] <http://dk1ri.de/myc/MYC.pdf> (deutsch)
- [2] http://dk1ri.de/myc/MYC_en.pdf (not up to date)
- [3] <http://dk1ri.de/myc/Description.pdf>
- [4] <http://dk1ri.de/myc/commands.pdf>
- [5] http://dk1ri.de/myc/Reserved_tokens.pdf
- [6] <http://dk1ri.de/myc/Rules.pdf>
- [7] <http://dk1ri.de/myc/commandrouter.pdf>
- [8] http://dk1ri.de/myc/Rules_device.pdf
- [9] <http://dk1ri.de/myc/skin.pdf>
- [10] <https://github.com/dk1ri>