Zum Mitmachen

Universelle Steuerung

Günter Klomann, DK1RI

Es handelt sich um ein Projekt aus dem Bereich „Internet der Dinge“, also etwas sehr Aktuelles. Ich habe es MYC genannt, das steht für My Control.

Einige Teile sind realisiert, aber das Projekt ist nahezu beliebig erweiterbar. Man kann noch Hard- und Software entwickeln, es gibt noch Möglichkeiten bei Projektierung und Service. Dies ist also ein Versuch, weitere Funkamateure zum Mitmachen beim MYC-Projekt zu bewegen.

Hintergrund und Motivation

Es gibt schon viele Schaltungen und Programme zur Steuerung von Funktechnik, aber leider passen oft nicht alle zueinander. Dies ist ein Vorschlag, wie man das ändern könnte.

Ich habe ein Programm für digitale Betriebsarten mit Logbuchzusatz. Der gefällt mir aber nicht, denn er kann zwar die verschiedensten Transceiver steuern, aber ich verwende für KW und UKW verschiedene Transceiver, manchmal auch einen zweiten für KW. Also muss ich jedes Mal umkonfigurieren. Der Rotor für KW hat gar keine Rechnersteuerung, der zweite für UKW nur eine IR-Fernbedienung. Eigentlich passt nur wenig zusammen, und irgendwie sollte eine universelle Lösung gefunden werden.

Das Problem ist sicher nicht neu, und es gibt ja diverse Geräte zur Stationssteuerung. Meist sind das proprietäre Lösungen, d.h., neue Geräte lassen sich nicht einbinden.

Ich beschreibe hier kein fertiges Gerät oder Programm, sondern ein System, das individuell konfiguriert werden muss, dies aber (im endgültigen Zustand) auch von Computerlaien. Daher auch der Name. Ein MYC-System ist individuell erweiterbar, nicht nur mit Funkzubehör, sondern eigentlich mit allem, was sich steuern lässt.

Das Konzept und ein Protokoll

Ein MYC-System besteht aus einem Command Router (Vermittler für Befehle) und mehreren Devices (Geräten), die über den Router mithilfe des MYC-Protokolls

Ein Blick ins Shack beim Autor zeigt bereits: Viele Geräte zieren die Amateurfunkstation. Wie lässt man diese am besten miteinander interagieren? Dazu gibt es die Idee eines universellen verwendbaren Protokolls

Daten austauschen. Jedes Device kann an andere Devices Befehle und Infos schicken; der Router vermittelt diese.

Jedes Device hat genau eine „feste“ Verbindung zum Router. Das MYC-Protokoll ist semantisch; es beschreibt nur Funktionen, ähnlich wie ein CAT-Protokoll und es gibt also keine Befehle wie verbinden, routen usw. Dies stört nicht, denn ein MYC-System ist üblicherweise eine örtliche Ansammlung von Geräten, die über feste Kabel verbunden sind und deren Anordnung sich kaum ändert.

Grundsätzlich können auch mehrere MYC-Systeme zusammenarbeiten.

Die Welt der Devices

Ein Device ist zum Beispiel ein Funkgerät, das das MYC-Protokoll versteht, also ein kleines Interface besitzt, welches das MYC-Protokoll in das gerätespezifische CAT-Protokoll übersetzt.

Ein Device kann bei Aufforderung eine Liste aller Befehle und Infos, die es versteht – eine so genannte Announce (Ankündigung) – an den Command Router geben.

Ein MYC-Interface sollte möglichst alle Funktionen des Geräts mit dem MYC-Protokoll abbilden. Dazu gibt es unterschiedliche Vorgehensweisen. Die einfachste: Alle CAT-Befehle werden in MYC-Befehle übersetzt. Das funktioniert häufig ganz gut, verursacht aber ein Problem, wenn CAT-Befehle zum Abfragen des gerade gültigen Status‘ fehlen.

Einem Transceiver, der zum Einstellen der Memory-Kanäle nur + und – als CAT-Befehle kennt, könnte man mit dem MYC-Protokoll sagen, er soll z.B. Kanal 25 einstellen. Das Interface muss dazu immer den aktuellen Memory-Kanal wissen und kann dann entsprechend viele Kommandos geben. Es würde so einen Befehl kennen, der nicht direkt in das CAT-Protokoll übersetzt wird.

Andererseits kann das MYC-System auch eine zentrale Frequenzverwaltung haben, und im Funkgerät wird der Memory-Befehl überhaupt nicht verwendet. Das Interface hätte dann weniger Befehle als das CAT-Protokoll.

Manchmal gibt es auch Befehle, die sich nur auf die Anzeige am Gerät – wie etwa die Helligkeit – beziehen und für eine reine Fernsteuerung keinen Sinn machen.

Aber da man nie weiß, wie das endgültige System wirklich aussieht, wird man im MYC-Interface alle CAT-Befehle realisieren.

Auch ein Programm mit einer MYC-Protokollschnittstelle kann ein MYC-Device sein, das allerdings auch Befehle senden kann. Ein Logbuchprogramm z.B. holt

sich die Frequenz vom Transceiver und Rufzeichen und Betriebsart vom decodierenden RTTY-Programm.
Normale Devices können nicht nur auf Befehle reagieren, sondern auch Infos ohne Befehl senden, etwa Frequenzwerte, wenn der Transceiver manuell bedient wird.
Es gibt noch weitere Devices, die Befehle senden können, z.B. ein Bedienprogramm zur Steuerung des Systems. Auch dieses muss sich wie ein Device verhalten und schickt die Befehle über den Command Router.

Der Command Router

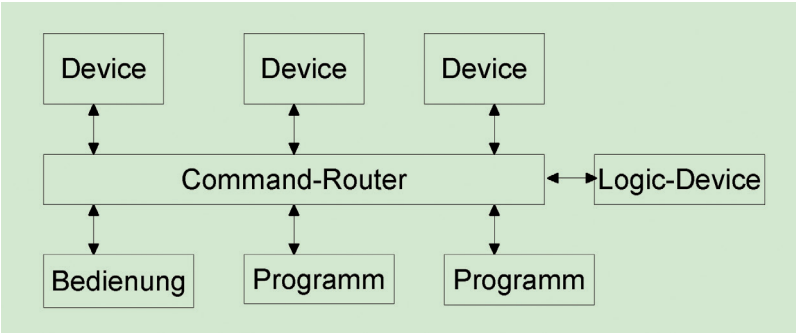
Jedes Device erhält die Befehle von diesem Router. Der muss also wissen, welche Befehle (Commandtokens, das sind einfach Nummern) die einzelnen Devices verstehen. Diese Information fragt der Router (im einfachsten Fall) bei den Devices mit dem standardisierten Announce-Befehl ab. Der Router weiß dann, welches Gerät welche Befehle empfangen und bearbeiten kann und macht daraus eine Gesamtliste der Befehle aller Devices mit übersetzten Commandtokens. Diese kann er an Devices senden, die Befehle senden wollen, wie etwa das Bedienprogramm.
Wenn der Router dann Befehle (Commandtokens) erhält, übersetzt er sie zurück und gibt sie an die richtigen Devices weiter. Alle Announce-Befehle beantwortet er allerdings selbst.
Es gibt noch einige weitere reservierte Befehle, die der Router auch als eigene Befehle versteht [1].

Das Logic Device

Manchmal ist aber alles nicht so einfach. Es gibt daher noch eine weitere Aufgabe in einem etwas komplizierteren Gesamtsystem.
Wenn z.B. das Bedienprogramm nur eine Frequenzeinstellung für KW und UKW kennt, aber zwei Geräte vorhanden sind, muss der Router den PTT-Befehl an das richtige Gerät schicken, möglicherweise auch Antennen und NF umschalten und dem Bediengerät sagen, dass bei UKW andere Bedienelemente vorhanden sind als bei KW.
Wenn man in einem MYC-System neben Funkgeräten auch die Haussteuerung einbindet, gibt es auch logische Verbbindungen, wie „Antenne herunterfahren und ausrichten abhängig vom Windmesser, Sendung einstellen“ oder „Geräte ausschalten, wenn niemand zu Hause ist“ usw. Die Regeln für das Gesamtsystem können vielfältig sein. Diese „Logik“ übernimmt das Logic Device, das ebenso wie der Command Router nur einmal im System vorkommt. Es erhält daher alle Befehle und Infos, die an den Router geschickt werden, verarbeitet sie, und erst diese Befehle werden über den Router an die Devices zur Bearbeitung geschickt. Alle Befehle und Infos machen also einen Umweg über das Logic Device.
Diese Zusammenhänge werden mit Regeln beschrieben, die entweder in einer Datei stehen oder von einem System Device abgerufen werden können.
Es gibt noch eine weitere Form von Logik, die das Logic Device übernehmen muss: Im einfachsten Fall sind alle Befehle eines Devices zu jeder Zeit erlaubt und funktionieren.
Aber kann der Transceiver beim Senden wirklich das Band oder die Betriebsart wechseln? Ist das eigentlich erwünscht? Oder kann er außerhalb der Amateurfunkbänder senden? Ohne Logik zeigt das Bediengerät den Sendebetrieb außerhalb des Bandes an, der Transceiver weigert sich aber einfach, zu senden. Solche gerätespezifischen Bedingungen kann das Device mit Rules (Regeln) dem Logic Device mitteilen.
Nicht alle Transceiver geben bei falschen oder ungültigen Befehlen „Nicht ok“ zurück. Daher muss das MYC-Interface dafür sorgen, dass das Logic Device mithilfe der Rules und Announcements nur gültige Befehle sendet oder, falls doch etwas schief geht, eine Fehlermeldung schicken.
Es kann ziemlich aufwändig sein, wirklich alle Fehlerbedingungen eines Gerätes herauszufinden. Ein nicht mehr ganz aktuelles Gerät wie der IC-7000 hat schon ca. 300 Befehle. Aus den Beschreibungen der CAT-Befehle gehen nicht immer alle Bedingungen eindeutig hervor; man muss im Zweifelsfall Vieles ausprobieren. Auch hat die Beschreibung möglicherweise Fehler oder ist nicht vollständig.
Das Logic Device muss dauernd den aktuellen Status des Gesamtsystems kennen. Aus den Befehlen und Infos, die es erhält, und aus dem aktuellen Status ergeben sich dann Ausgabebefehle. Das Logic Device ist eine so genannte State Machine.
Die Funktion der Regeln kann man zwar im einfachsten Fall direkt programmieren, also alle Rules in den Programmcode umsetzen, aber eleganter wäre es, die Struktur in einem Konfigurationsfile abzulegen und noch besser wäre ein Grafikeditor, der so eine Datei anlegen kann. Es gibt solche Programme zum Aufbau einer State Machine für den PC.
Zusammenfassend ist zu sagen, dass das Logic Device vom Command Router alle Befehle erhält, diese gegebenenfalls umarbeitet und an den Router zur Weitergabe zurückschickt. Einfache Systeme benötigen kein Logic Device.

Das Bediengerät

Bisher funktioniert ein Bedienprogramm so: Ich lade es, aber mir gefällt häufig einiges nicht, etwa die Art und Anordnung der Knöpfe. Oder einige Funktionen der Menüs würde ich lieber dauernd sehen, und die Farben sind unpassend.



Das Grundkonzept des MYC-Systems

In einem MYC-System könnte sich das Bedienprogramm (der Webserver für den Browser oder ein anderes Programm) alle verfügbaren Funktionen vom Commandrouter holen, und ich könnte sie in einem Edit-Modus nach meinen Vorstellungen anordnen und anpassen. Auf einem Smartphone würde das natürlich anders aussehen als auf einem Tablet oder Notebook.
Wenn man die Realisierung als Webserver/Browser betrachtet, hat man im Editier-Mode einen (stark einschränkten) Webeditor, im Bedienmode einen normalen Browser.
Das Bediengerät kann Befehle zur Anzeige entgegennehmen und Befehle abschicken. Es wird diese nicht selbst darstellen, sondern nur das, was das Logic Device vorgibt.
Die Realisierung dieser Art von Bedienprogramm ist noch zu klären. Solange

Zur Person

Günter Klomann, DK1RI
Jahrgang 1948, Amateurfunkgenehmigung seit 1966
Studium der Elektrotechnik, inzwischen Rentner
Besondere Interessen: digitale Betriebsarten

Anschrift:
Gartenstraße 23
68535 Edingen
dk1ri@darc.de
www.dk1ri.de

kann man die Befehle im einfachsten Fall auch mit einem Terminalprogramm eingeben oder (etwas komfortabler, aber mühsam und unflexibel) eine spezielle Webseite aufbauen.

Mehrfache Bediengeräte

Es ist möglich, dass mehrere Bediengeräte bzw. Programme gleichzeitig aktiv sind, weil das Logic Device immer den aktuellen Stand des Systems weiß und die Daten allen Bediengeräten mitteilt.
Wenn man ein Programm benutzen will, das das MYC-Protokoll nicht unterstützt, benötigt man ein MYC-Device als Interface. Das kann, wie alle Devices, Hardware oder Software sein. Das Interface emuliert dann das Gerät, welches das Programm bedienen will.

Datenübertragung

Für die Datenübertragung zwischen Devices und Command Router gibt es keine Vorgaben. Es sind Schnittstellen für RS232, I²C und SPI auf Mikrocontrollern vorgesehen. Andere, insbesondere Ethernet, sind möglich.
Alle Verfahren haben Vor- und Nachteile:

- RS232 ist eine 1:1-Verbindung. Man benötigt für jedes Device eine Schnittstelle. Diese Lösung eignet sich nur bei wenigen Devices.
- I²C setzt voraus, dass alle Devices an Spannung liegen. Die Leitungslänge ist begrenzt und die Störsicherheit nicht hoch. Da ein Device als Slave von sich aus keine Daten senden kann, muss der Command Router dauernd die Devices abfragen. Man kann auch ein Multimaster-System verwenden, das aber etwas komplizierter zu handhaben ist. I²C eignet sich für dicht zusammenliegende Devices.
- Für SPI gilt Ähnliches, wenn auch die Datenrate höher ist.
- Ethernet ist bidirektional, die Verbindung ist galvanisch getrennt und sehr schnell. Wenn man fertige Bausteine verwendet, ist die Programmierung vielleicht einfach; allerdings ist diese Lösung etwas teurer. Mit Ethernet an der Schnittstelle zum Bediengerät ist auch eine Steuerung über Internet und vom Smartphone aus möglich.

Der Command Router kann verschiedene Übertragungsprotokolle gleichzeitig bedienen, abhängig von der verwendeten Hardware. Das MYC-Protokoll macht da keine Vorgaben. Es ist ein Protokoll zwischen Rechnern, hat also möglichst kurze, aber auch nicht einfach lesbare Befehle, um auch langsame Datenübertragung zu den Devices zu ermöglichen.
Das semantische MYC-Protokoll hat keine Möglichkeit für Verbindungsaufbau, Fehlerkorrektur, Verschlüsselung usw. Das muss bei Bedarf getrennt erfolgen.

Die Befehle

Jedes Device versteht und bearbeitet einen Satz von Befehlen, die 1, 2 oder 4 Byte lang sein können, je nachdem, wie viele davon ein Device versteht oder sendet.
Jeder Befehl gehört zu einem Befehlstyp, z.B. „Schalter“. Der Befehlstyp bestimmt die Syntax des Befehls und „prinzipiell“ die Art der Anzeige am Bediengerät. Ein mehrstufiger Schalter kann dabei als Tastenfeld, Drehschalter, Schiebeschalter oder irgendwie anders dargestellt werden; der MYC-Befehl ist identisch.
Es war für mich schon überraschend, mit wie wenigen Befehlstypen man eigentlich auskommt. Dabei ermöglicht das MYC-System nicht nur die Steuerung von Funkgeräten, sondern von fast beliebigen anderen Geräten.
Eine Liste der Befehlstypen und die Beschreibung dazu steht in [2].

Personalisierung

Die Announcements eines Devices beschreiben einen Device-Typ eindeutig. Der Command Router kann die Befehle den Devices zuordnen. Wenn es mehrere gleiche Devices gibt, benötigen diese eine Identifikation, damit ein Bediengerät sie unterscheiden kann. Dazu erhalten sie bei Bedarf vor der Einbindung in das eigentliche Ziel-MYC-System eine individuelle Nummer, einen Namen und je nach Bedarf weitere Informationen, wie etwa Verbindungsinformationen. Die Personalisierung kann auch mit einem MYC-Befehl erfolgen [1].

Hardwareauswahl und Praxis

Das ganze System, so wie bisher beschrieben, ist modular aufgebaut mit verschiedenen Teilen, die weitestgehend unabhängig voneinander realisiert werden können. Es sind auch nicht unbedingt alle Teile notwendig, um ein System zur Steuerung aufzubauen. Im einfachsten Fall hat man nur Geräte mit MYC-Interface und eine Schnittstelle für die Eingabe mit einem Terminalprogramm. Das ist zwar ziemlich unbequem, aber es zeigt, dass das MYC-System funktioniert und ist auch mein derzeitiger Entwicklungsstand.
Für die MYC-Interfaces zu den Geräten reicht meist ein kleiner Mikrocontroller oder ein Arduino.
Meine nächste Aufgabe ist ein Command Router, für den ein Raspberry Pi ausreichen dürfte. Ob das auch für den Webserver für die Bedienung gilt, ist noch zu klären.

Anwendungen

Meine erste Hauptanwendung liegt natürlich im Shack. Aber auch eine Haussteuerung ist möglich. Ein Interface für das FS20-System ist in Planung. Für Unterhaltungselektronik gibt es ein MYC-Interface mit Infrarotsender als Vorlage. Das Steuerprogramm muss man allerdings, wie bei den meisten Interfaces, an die realen Geräte anpassen.

Zusammenfassung

- Ein MYC-System kann über relativ einfache Interfaces beliebige Geräte steuern.
- Mit dem noch zu erstellenden Bedienprogramm entfällt die Programmierung von Benutzeroberflächen; es genügt eine einfache Konfiguration.
- MYC ist ein modulares System, die Devices können unabhängig voneinander erstellt werden.

Dieser Artikel ist eine Einführung, Details stehen auf meiner Webseite. Für weitere Ergänzungen und Anregungen wäre ich dankbar.
Was gibt es zu tun?:

- Das Wichtigste: Für ein komplettes Grundsystem fehlen Benutzerprogramm und Logic Device. Der Command Router ist auch noch nicht fertig.
- Es gibt viele Geräte ohne MYC-Interfaces.
- Auch bei einem fertigen System wird es Funkamateure geben, die bei der Konfiguration der Bedienung und der Definition der Systemregeln Hilfe benötigen. Hierfür wäre ein Service hilfreich.

CQDL

Literatur und Bezugsquellen

[1] www.dk1ri.de/common/myc/MYC_basic/Reserved_tokens.pdf
[2] www.dk1ri.de/common/myc/MYC_basic/commands.pdf
[3] www.dk1ri.de/common/myc/MYC_basic/MYC.pdf (deutsch)