

Verwendung

Zweck des Dokuments

Dies ist eine Anleitung wie das Kommandozeilentool QtRoboParser zu verwenden ist. Außerdem wird kurz die interne Funktionsweise dargestellt.

Voraussetzungen

Die Applikation nutzt Unix Domain Sockets, weshalb sie nur unter Linux Distributionen ausgeführt werden sollte. Unix Domain Sockets sind auch unter Windows verfügbar, allerdings ist die Anwendung nur unter Arch Linux getestet und keine Portierung vorhanden.

QtRobo muss zum Senden von Events verwendet werden.

Bauen der Applikation

Voraussetzung zum Bauen der Applikation ist die Installation von make zusammen mit einem C++ Compiler. Das Projekt enthält ein Makefile. Dieses beschreibt die zum Bauen der Applikation benötigten Schritte für den verwendeten C++ Compiler. Solange make installiert ist, ermöglicht das Ausführen des Kommandos make im ./src Verzeichnis des Projekts auf dem Terminal die Erstellung der verschiedenen Object Dateien (.o) und das anschließende Linken zur Main executable.

Funktionsweise

QtRoboParser erstellt einen Unix Domain Socket und wartet auf eine Verbindung von QtRobo. Sobald eine Zeile auf dem Socket übertragen wird, parst QtRoboParser diese in ein Event, welches den internen mit 0 für proportionale und false für binäre Kanäle initialisierten Buffer verändert. Gleichzeitig wird bei Verbindung eines Clients das Senden der SumD Event-Frames auf den Standard Output gestartet. Jeder Name eines Controls in QtRobo muss deshalb dem folgenden Schema entsprechen:

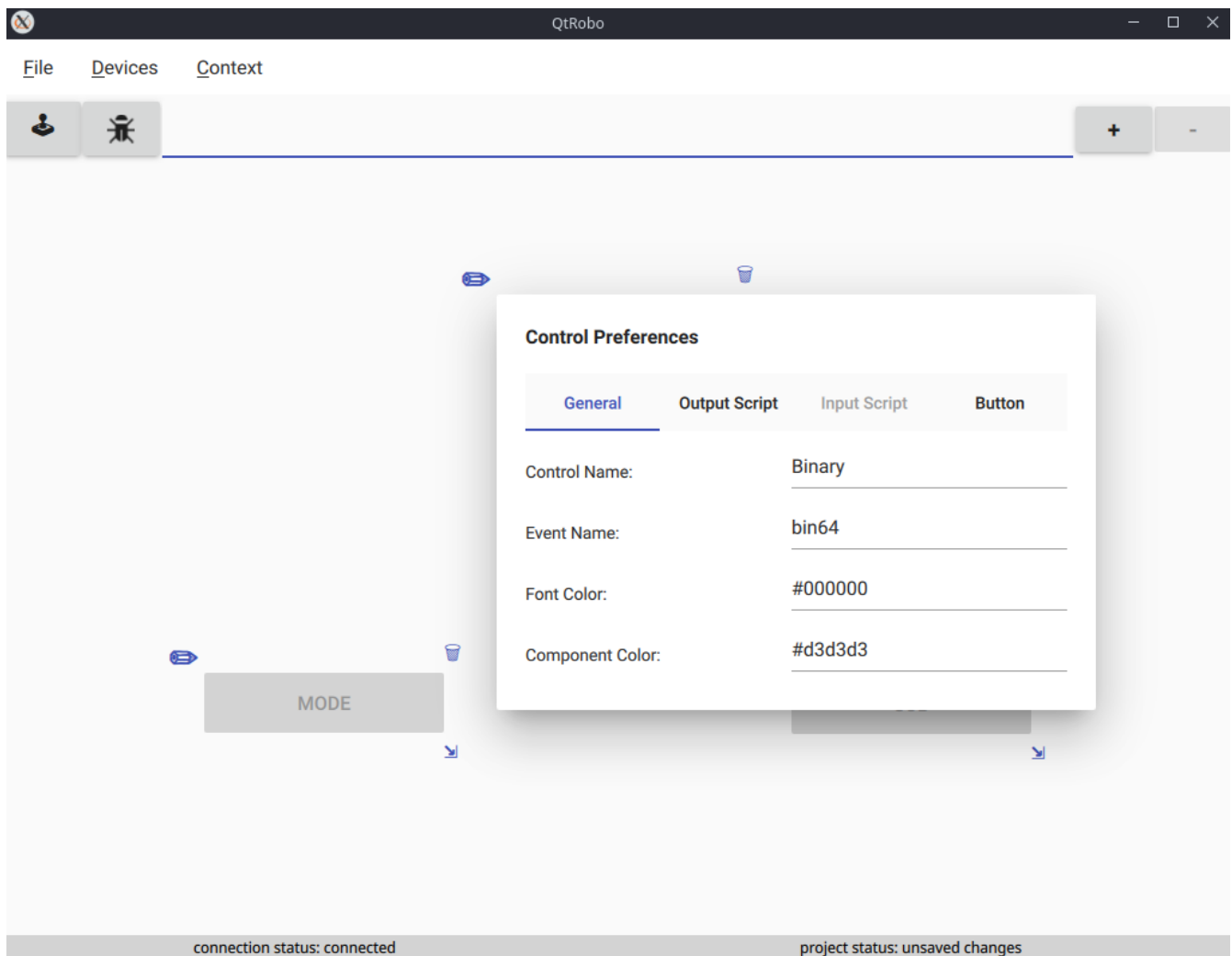


Figure 1. Beispiel für ein binäres QtRoboEvent mit Kanal 64

Starten der Anwendung

QtRoboParser ist eine Kommandozeilenanwendung. Sie wird deshalb auf dem Terminal über den Befehl `./Main` gestartet. Die Anwendung muss zum Starten die in QtRobo für die einzelnen Controls verwendeten Präfixe übergeben bekommen. Außerdem ist der Pfad, unter dem der Unix Domain Socket angelegt werden soll, konfigurierbar.

Die genaue Syntax des Aufrufs kann mit `./Main -h | --help` angezeigt werden.

Danach wartet die Anwendung auf die Verbindung eines Clients. In QtRobo muss deshalb als nächstes die Connection mit dem übergebenen Socketnamen eingerichtet werden. Dies ist unter `Devices → Connection → Local Socket` möglich. Der Name muss dem an QtRoboParser übergebenen Namen entsprechen und ein valider Unix Domain Socket sein - beispielsweise `/tmp/abc`. Angelegte Controls, die dem o.g. Namensschema entsprechen und das übergebene Präfix verwenden, führen bei Veränderung ihres Wertes im Spielmodus von QtRobo zur Übertragung an den in QtRoboParser erstellten Socket.

Achtung: QtRobo sendet bei zu langen Eventnamen unlesbare Informationen an den Socket. Präfixe und Namen sollten entsprechend kurz gewählt werden.

Auf dem Terminal in dem QtRoboParser gestartet wurde, sollten anschließend SumD Event Frames

geschrieben werden. Sollte etwas nicht funktioniert haben, gibt die Anwendung eine Fehlerbeschreibung auf dem Standard Error Output aus. Ist die Option -d bzw. --debug true gesetzt, ist die Ausgabe lesbar. Im normalem Betrieb werden zur Verwendung von Pipes zur Kommunikation mit anderen Unix Prozessen nicht lesbare Binärdaten ausgegeben.

Beispiel

Ein zur Demonstration aller Eventtypen erstelltes QtRobo Layout ist in diesem Projekt als TestLayout zu finden. Diese kann in QtRobo geöffnet und nach der Verbindung mit dem vorkonfigurierten Socket /tmp/abc verwendet werden.

QtRoboParser muss dazu vorher mit dem folgenden Befehl gestartet werden: `./Main -b bin -p prop -m mode -s sub -u /tmp/abc` Soll die Ausgabe anstatt Binärdaten sichtbar sein, muss folgender Befehl verwendet werden: `./Main -b bin -p prop -m mode -s sub -u /tmp/abc -d true`

Ende der Verbindung durch den Client

Sobald der verbundene Client die Verbindung trennt, wartet QtRoboParser auf eine erneute Verbindung eines Clients.

Beenden der Anwendung

Die Anwendung kann jederzeit mit Ctrl+C beendet werden. Alle bestehenden Sockets werden dabei gelöscht.