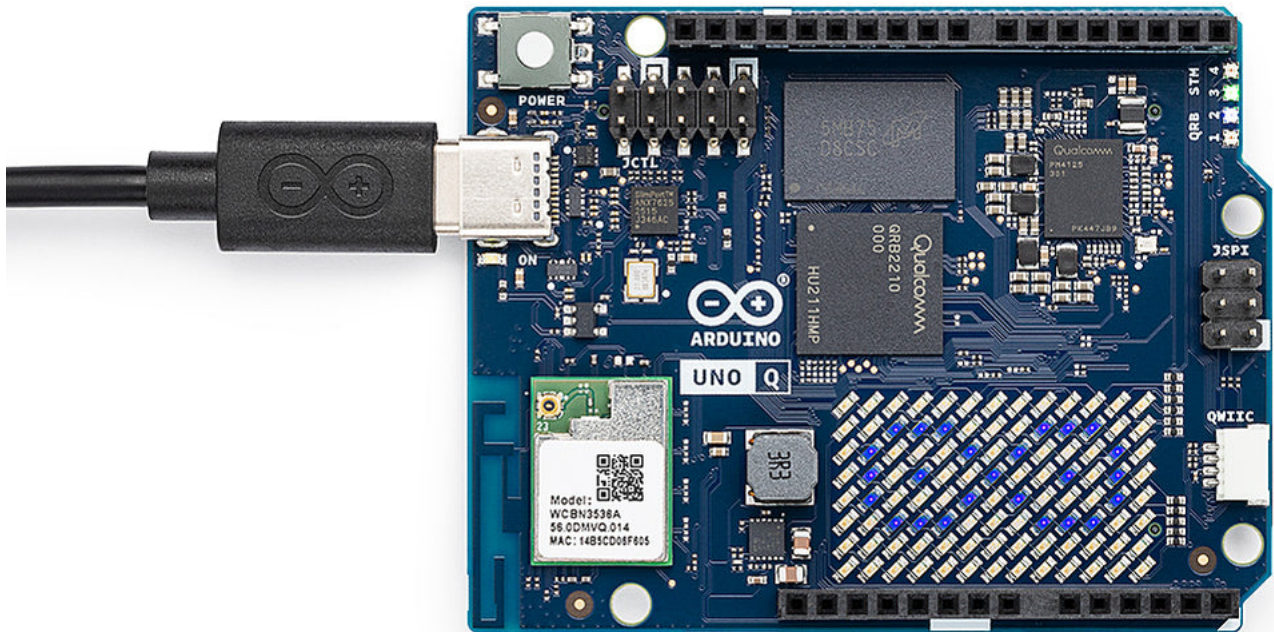


Update : Acia6850 01.08.2026

Arduion Q Bridge Lazarus Demos

Einführung



Der Arduino Q hat zwei Prozessoren :

MPU für Linux

Implementiert ist ein Linux Debian Trixie Aarch64

Es laufen alle Programme und Tools welches Debian Trixie zu Verfügung stellt.

Man kann Lazarus für Aarch64 Installieren und somit Lazarus Konsolen und Gui Programme erzeugen.

MCU für Echtzeit und IO

Hier wird mit der Arduino IDE 2.xx programmiert in C oder C++.

Es sind für die Arduino Welt die Libs und Demos zur Verfügung.

Die Beispiele für den Arduino Q sind größtenteils in Python und C oder C++.

Es gibt eine Entwicklungsumgebung **Arduino App Lab**

Die ist Teil der Arduino Q Installation und immer zur Verfügung.

Die beiden Prozessoren (MPU und MCU) sind über eine Software Bridge Seriell verbunden.

Siehe Doku.

Lazarus Demos :

Für die Steuerung der MCU von der MPU muss man die Software Bridge verwenden.
Das geht über UNIX SOCKETS und Msgpack.

In den Demos habe ich mit Lazarus die Unix-Sockets und eine für die Demos ausreichender Nachbildung der Msgpack Lib programmiert.

Wer mehr Funktionen von der Msgpack benötigt kann die C++ Msgpack lib mit einem C++ Wrapper in Lazarus einbinden.

Es gibt auch für Delphi eine Msgpack Lib.

In der Arduino Q Doku „routerbridge-multilanguage“ wird die Einbindung der Bridge in C++ und Python beschrieben mit Demos.

Ich habe die C++ Demos in Lazarus nachgebildet.

Demos :

cLazQLedBlink

cLazQSensor

CLazQStatusLed (steuert direkt über File Zugriff die MPU Status Leds)

CLazQBridgeDemoGui

Die Demos sind im Source erhalten und sind frei zur Verfügung und Änderung.

Viel Spaß mit dem Arduino Q und Lazarus.

Arduino Q Doku links :

<https://www.arduino.cc/product-uno-q>

<https://docs.arduino.cc/tutorials/uno-q/routerbridge-multilanguage/>

<https://forum.arduino.cc/t/let-us-learn-arduino-uno-q-step-by-step/1426536>

<https://github.com/msgpack/msgpack/blob/master/spec.md#overview>

