THE LITTLE CONTEXT BOOK

RUNNING TCX ON AN ARDUINO UNO Q

the gadget

The Arduino Uno Q is tiny board with a Qualcomm® Dragonwing™ QRB2210 microprocessor (MPU) and STM32U585 microcontroller (MCU). You can best take the one with 4GB memory (DDR4) and 32GB eMMC disk. It has WIFI and more on board. It has a full blown USB-C connector which interfaces with an external hub so that you can have HDMI, Ethernet and several USB ports. Power then comes via that connection. It runs Debian Linux.

When booted up you have to answer some questions, like network connection and password. However, when you get this 'flickering' screen that makes it impossible to set it up, you can use 'arduino app lab' on a laptop to set up the initial password and network.

When the desktop pops up you can launch a console and:

```
sudo apt update
sudo apt upgrade
```

and then

```
sudo apt install okular
```

Maybe you want to do:

```
sudo hostnamectl set-hostname cms-xx
sudo hostnamectl
```

the tools

Normally I put all under `/data` but in this case we have to go under the home directory because the root sector is kind of small. Beware, the used name is `arduino` which is in group `arduino`.

```
mkdir ~/data
mkdir ~/data/context
mkdir ~/data/scite
mkdir ~/data/test
```

Next one has to install the `aarch64` version of ConTEXt, here under `~/data/context`. Installing SciTE can be done from the installation arrchive. The `readme.txt` file shows where files fly. Make sure to set the permissions on the binary. You might also decide to change the user, group and permissions on the `share/scite` path and files so that you can update as well as adapt the properties file:

```
/usr/share/scite/context/scite-context.properties
```

The relevant variable is:

```
name.flag.pdfopen=--autopdf=okular
```

You need to add the binaries path (console or in `~/.profile` or `~/.xsessionrc`):

```
export PATH=~/data/context/tex/texmf-linux-aarch64/bin:$PATH
```

Then you can test:

```
cd ~/data/test
SciTE test.tex &
```

Running ConTEXt should happen with `control+F7`, `control+F12` or `control+1`; it depends on the operating system. Of course a first run will cache some fonts.

the clock

If you want the signal gadget (aka ConTEXt Watch) to work you need to do this:

```
mkdir ~/data/context/tex/texmf-local/context
mkdir ~/data/context/tex/texmf-local/context/data
mkdir ~/data/context/tex/texmf-local/context/data/signal

cp  ~/data/context/tex/texmf-context/context/data/signal/ctxsignals-template.lua \
    ~/data/context/tex/texmf-local/context/data/signal/ctxsignals.lua

mtxrun --generate
```

You also need to permit control over the serial bus, something that needs a reboot to be effective.

```
sudo usermod -a -G dialout yourname (and reboot)
sudo reboot
```

the rest

Here are some more useful helpers:

```
sudo apt install mupdf
sudo apt install mupdf-tools
sudo apt install qpdfview
```

If you're deep into tagging, you might want to also install `verapdf` which is of course not the fastest feature then. First you need to install a runtime:

```
sudo apt install default-jre
```

We then download the installer and put it in:

```
~/data/verapdf
```

After unpacking we run

```
./verapdf-install
```

And when asked for a next step make sure we adapt the directory it will end up in: `~/data/verapdf` and adapt `.xsessionrc`:

```
export PATH=/home/arduino/data/context/tex/texmf-linux-aarch64/bin:\
/home/arduino/data/verapdf:\
$PATH
```

After this one-liner has been set you need to login again to make it effective but after that `mtxrun --script pdf --validate ...` should work. Don't expect spectacular performance; we just wanted to demo it but never will use it on such an under-powered device.

the end

A shutdown is a bit special on a device that is supposed to be always on. A more permanent ending is one of:

```
sudo shutdown -h now
sudo halt
```

If you want a text on the display ... there are ways to do this but you need to make an app and get that in the controler. At some point we might put one in the distribution.

the wrapper

This setup was used at BachoTEX 2026 by Hans Hagen, Mikael Sundqvist and Willi Egger for a presentation. There we used this board in small booklets (dwarsliggers) made by Willi for this occasion.

Of course this is way more cool than a plastic enclosure. If I had a 3D printer I'd play with

```
https://makerworld.com/en/models/2218829-arduino-uno-q-case-kit-snap-fit-wall-mount\
#profileId-2412804
```

in order to get a 'server' setup.