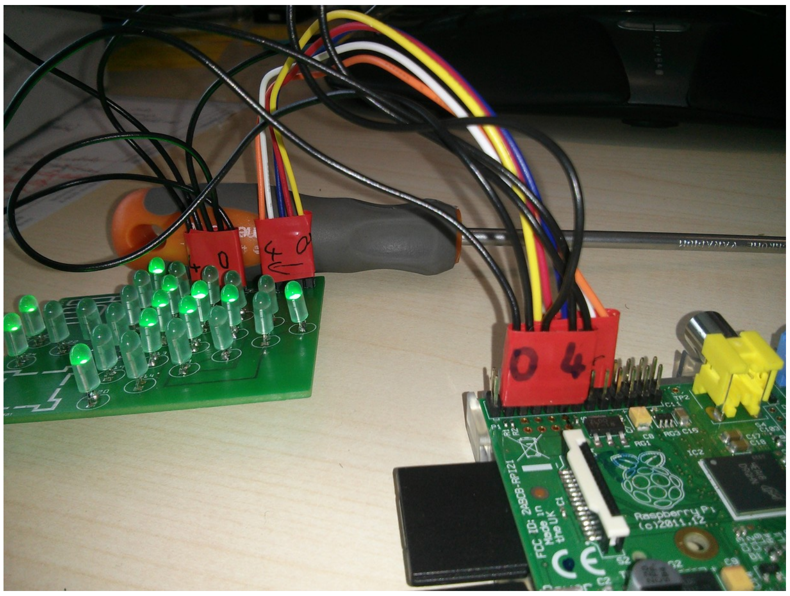
This box should contain:
-raspberry 2 with SD card containing the assembler code to run the LED tree
-4 LED tree pcb boards. 2 with soldered leds
-2 cables (1 black, one colored) to attach LED tree pcb board to Raspberry
-Power adapter for Raspberry
-This instruction sheed

Example of ARMv6 assembler. Shows given bitmask on 5x5 LED matrix attached to raspberry pi.

This project uses a raspberry pi model B to control a 5x5 LED matrix. Function bitMask2LED takes 32 bit of register r0, and lights up every 1bit on the LED matrix. Bits 24-31 are ignored.
The code is written in ARMv6 assembler.
The specific adresses used to control the GPIO pins can be found in the "BCM2835 ARM Peripherals" pdf document.
GPIO pins 2,3,4,17,27 are used as GND pins. GPIO pins 14,15,18,23,24 are used as Voltage pins.
For the GPIO connections on the raspberry see
https://en.wikipedia.org/wiki/Raspberry_Pi#General_purpose_input-output_.28GPIO.29_connector.



To run this you need a working arm toolchain, the specific commands used can be seen in the Makefile.
As promised by ZID this should still work in the "Rechnerräume" as it did in the WS15/16.
To run it under debian based OSs the following packages have to be installed: binutils-arm-linux-gnueabi

1)Connect your raspberry to the LED Matrix. Typically a LED Matrix with N LEDs has on one side sqr(N) connections
for anodes (+) and on another side sqr(N) connections for cathodes (-). For our 5x5 Matrix this results in 5 anode
pins and 5 cathode pins. Connect the GPIO ground pins (here 2,3,4,17,27) to the cathode pins and the GPIO voltage
pins (14,15,18,23,24) to the anodes.

2)Change the commands in the Makefile depending on your OS (arm-linux-gnueabi vs arm-none-eabi). Run "make kernel.img"
This creates a new 'build' folder including files needed for compilation and two additional files
'kernel.img' and 'kernel.map'

3)Boot the raspberry from the 'kernel.img' file.
The simplest way to do this is to create a bootable SD card with a raspberry pi OS (e.g. raspbian). Then overwrite the
'kernel.img' file on the SD card with the one we created in step 2). Insert the SD card in your raspberry at boot it up.