

Änderungen bei PIOS Bookworm

WiringPi muss lauffähig gemacht werden.

1. Wiringpi benötigt zusätzlich zu „RPI_Lazarus-master.zip“ und den Eintrag „Projekt→ Projekteinstellungen→ Compilereinstellungen→ Pfade→ Andere Units→“ (dort hinzufügen „< Phath>/myLib/wiringPi“).
2. Die Installation von „WiringPi-arm64-master.zip“ (von <https://github.com/guotion/WiringPi-arm64> ist leider nicht automatisch enthalten) nach entpacken entsprechend der Datei „INSTALL“, sonst zeigt der linker „Error: /usr/bin/ld.bfd: cannot find -lwiringPi“.
3. Wegen dem Fehler "No "Hardware" line" muss danach noch die Datei „...\\WiringPi-arm64-master/wiringPi/wiringPi.c“ editiert werden. Unter der Zeile 747 mit „int piGpioLayout (void)“ steht die Zeile „#ifdef DONT_CARE_ANYMORE“ . Diese ist nach unten zu verschieben unter die Zeile (ca. Zeile 758) „piGpioLayoutOops ("Unable to open /proc/cpuinfo") ;“
Dann sieht es so aus:

```
... ..
if ((cpuFd = fopen ("/proc/cpuinfo", "r")) == NULL)
    piGpioLayoutOops ("Unable to open /proc/cpuinfo") ;
#ifdef DONT_CARE_ANYMORE
... ..
```

Dadurch wird die Prüfung der (nicht mehr vorhandenen) „Hardware“-Zeile verhindert, aber die Revisionsprüfung bleibt funktionsfähig.

4. Speichern Sie die Datei und führen Sie das WiringPi Build-Skript erneut aus:
cd „...\\WiringPi-arm64-master“
./build

Danach sollte Lazarus wieder normal funktionieren (übernommen von <https://raspberrypi.stackexchange.com/questions/145031/wiringpi-giving-unable-to-determine-board-revision-from-proc-cpuinfo-error-af>)

Kamerastreaming muss ersetzt werden, weil die „picamera“ durch „picamera2“ ersetzt wurde. Das Python Skript muss ersetzt werden. (Z.B. durch „mjpeg_server.py“ von https://github.com/raspberrypi/picamera2/blob/main/examples/mjpeg_server.py)

```
#!/usr/bin/python3
# Mostly copied from https://picamera.readthedocs.io/en/release-1.13/recipes2.html
# Run this script, then point a web browser at http:<this-ip-address>:8000
# Note: needs simplejpeg to be installed (pip3 install simplejpeg).
import io
import logging
import socketserver
from http import server
from threading import Condition
from picamera2 import Picamera2
from picamera2.encoders import JpegEncoder
from picamera2.outputs import FileOutput

PAGE = """\
<html>
<head>
<title>picamera2 MJPEG streaming demo</title>
</head>
<body>
<h1>Picamera2 MJPEG Streaming Demo</h1>

</body>
</html>
"""
```

```

class StreamingOutput(io.BufferedIOBase):
    def __init__(self):
        self.frame = None
        self.condition = Condition()
    def write(self, buf):
        with self.condition:
            self.frame = buf
            self.condition.notify_all()
class StreamingHandler(server.BaseHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            self.send_response(301)
            self.send_header('Location', '/index.html')
            self.end_headers()
        elif self.path == '/index.html':
            content = PAGE.encode('utf-8')
            self.send_response(200)
            self.send_header('Content-Type', 'text/html')
            self.send_header('Content-Length', len(content))
            self.end_headers()
            self.wfile.write(content)
        elif self.path == '/stream.mjpg':
            self.send_response(200)
            self.send_header('Age', 0)
            self.send_header('Cache-Control', 'no-cache, private')
            self.send_header('Pragma', 'no-cache')
            self.send_header('Content-Type', 'multipart/x-mixed-replace; boundary=FRAME')
            self.end_headers()
            try:
                while True:
                    with output.condition:
                        output.condition.wait()
                        frame = output.frame
                        self.wfile.write(b'--FRAME\r\n')
                        self.send_header('Content-Type', 'image/jpeg')
                        self.send_header('Content-Length', len(frame))
                        self.end_headers()
                        self.wfile.write(frame)
                        self.wfile.write(b'\r\n')
            except Exception as e:
                logging.warning(
                    'Removed streaming client %s: %s',
                    self.client_address, str(e))
        else:
            self.send_error(404)
            self.end_headers()
class StreamingServer(socketserver.ThreadingMixIn, server.HTTPServer):
    allow_reuse_address = True
    daemon_threads = True
picam2 = Picamera2()
picam2.configure(picam2.create_video_configuration(main={"size": (640, 480)}))
output = StreamingOutput()
picam2.start_recording(JpegEncoder(), FileOutput(output))
try:
    address = ("", 8000)
    server = StreamingServer(address, StreamingHandler)
    server.serve_forever()
finally:
    picam2.stop_recording()

```

Der **Accesspoint** ist mit dem Netzwerkmanager zu konfigurieren (ohne DHCPD, DNSMasq und HostAPD). Dazu werden zwei Dateien in „/etc/NetworkManager/system-connections/“ gebraucht.

Für das Netzkabel

Wired connection 1.nmconnection

```
[connection]
id=Wired connection 1
uuid=<wird vom Programm erzeugt>
type=ethernet
autoconnect-priority=-999
interface-name=eth0
timestamp=<wird vom Programm erzeugt>
[ethernet]
[ipv4]
address1=192.168.1.x/24,192.168.1.1      muss angepasst werden
dns=192.168.1.1; 1                    muss angepasst werden
method=manual
[ipv6]
addr-gen-mode=default
method=disabled
[proxy]
```

und für den WiFi Accesspoint (mode=ap)

WlanRaspi.nmconnection

```
[connection]
id=WlanRaspi
uuid=<wird vom Programm erzeugt>
type=wifi
interface-name=wlan0
timestamp=<wird vom Programm erzeugt>
[wifi]
band=bg
mac-address=<vom RasPi>
mode=ap
ssid=WlanRaspi
[wifi-security]
key-mgmt=wpa-psk
psk=<Password>
[ipv4]
method=shared
[ipv6]
addr-gen-mode=stable-privacy
method=disabled
[proxy]
```

Das kann auf der Befehlszeile oder halbgrafisch mit „sudo nmtui“ oder grafisch nach Rechtsklick auf das Netzwerksymbol der Taskleiste erfolgen (siehe <https://raspberrypi.com/access-point-setup-raspberry-pi/#setting-up-an-access-point-on-raspberry-pi-os-bookworm>).

Leider muss man den „HotSpot“ (ap) mit den Adresseinstellungen nehmen die automatisch von Bookworm erzeugt werden, da keine Möglichkeit zu finden war, diese zu wählen.