ROAD 1.0 RC2 – mehr Tempo, und warum die Straße bei BOX bleibt

Kurzbericht zu den Messungen auf dem Pico Computer 3 · 378 MHz · MODE 2 · Firmware 7 und 8.00.00b2 · Matthias (homa), 2026

Kurz gesagt: RC2 ändert nur das Tempo, sonst nichts. In der dichtesten Szene (Start von Green Valley, rund 40 Palmen) steigt die Bildrate von **14 auf 17–18 fps**, auf freier Strecke laufen 20–23 fps. RC2 läuft weiter auf Firmware 7 und auf Firmware 8. Neu ist nur ROAD.BAS; die Streckendateien bleiben, wie sie sind.

1. Wohin die Zeit geht

Ein Bild am Start von Green Valley dauerte 59 ms (17 fps). So verteilt sich das:

Teil	ms	Was passiert
BG	7	Himmel und Hintergrundbilder (BLIT FLASH)
RD + PO	19 + 4	Straße: 120 Segmente projizieren, Zeilen als BOX sammeln und zeichnen
SP	20	Sprites: Palmen, Schilder, Gegner, eigenes Auto
Rest	9	Spiellogik, Gegner, Ton, HUD, Bild auf den Schirm

Die wichtigste Regel: Auf dem PicoMite kostet jeder *BASIC-Befehl* Zeit, kaum jedes *Pixel*. Auch mit OPTION COMPILE ON, bei dem der Compiler das Programm vorab in fertige Arbeitsschritte übersetzt, bleibt jeder Befehl ein eigener Schritt mit eigenem Verwaltungsaufwand. Ein Firmware-Befehl wie BOX erledigt dagegen den ganzen Zeichenauftrag in einem Zug. Gemessen: 480 BOXen zu zeichnen kostet 0,55 ms – die BASIC-Rechnung, die diese 480 BOXen vorbereitet, rund 10 ms. Schneller wird ROAD also nur mit *weniger Befehlen*, nicht mit schnelleren Zeichenbefehlen.

2. Was ausprobiert wurde

Idee	Ergebnis	
BLIT RESIZE pro Straßenzeile (Idee aus dem Forum)	Straße 30 statt 26 ms, ferne Mittelstreifen fehlen teils	nein
Abkürzungen im BOX-Weg (Randstreifen, Linien, Gras auslassen)	Straße 2–3 ms langsamer: die Prüfungen kosten mehr, als sie sparen	nein
FRAMEBUFFER WARPH (neu in FW8)	selbst sehr schnell (0,4 ms), passt aber nicht zu ROAD – siehe 3.	nein
Sprites außerhalb des Schirms vorher aussortieren	nur 3 von 42 lagen außerhalb – kein Gewinn	nein
Deko-Sprites ohne SUB-Aufruf zeichnen	5–7 ms weniger pro Bild, gleiches Bild	RC2
Deko-Sichtweite 60 statt 80 Segmente, nur auf dichten Abschnitten	+1–2 fps; 40 Segmente wären sichtbar aufgepoppt	RC2

3. Warum FRAMEBUFFER WARPH hier nicht greift – einfach erklärt

WARPH kopiert ein Bild Zeile für Zeile und darf dabei jede Zeile seitlich verschieben und aus einer beliebigen Quellzeile nehmen. Für die klassische Arcade-Straße ist das ideal: Jede Bildschirmzeile hat dort eine feste Straßenbreite, Kurven sind nur Verschiebungen, Hügel ein Stauchen der Zeilen. Die ganze Straße kostet so unter 1 ms. ROAD ist anders gebaut:

- **Die Straße ist breiter als der Bildschirm.** Unten misst sie rund 630 Pixel, der Schirm zeigt 320. WARPH kann eine Zeile nur *verschieben*, nicht strecken (der Befehl hat keinen Parameter zum Skalieren) – und eine Quellzeile ist höchstens 320 Pixel breit, also nur die halbe Straße. Wechselt das Auto die Spur, fehlt genau der Teil, der jetzt ins Bild rückt (Bild unten).
- **Die Breite ist nicht fest.** Echte Hügel, schmalere Zonen und wechselnde Spurzahl brauchen für jede Breite, Farbe und Spurzahl eine eigene Quellzeile: über 270 Zeilen. Frei sind im Puffer nur 22 – der Rest enthält die Sprites. Die zusätzlichen Puffer L und T werden in MODE 2 immer angezeigt und taugen deshalb nicht als verstecktes Lager.
- **Die Zeit steckt woanders.** Selbst wenn die Straße gar nichts kostete (gemessen, indem sie testweise einfach weggelassen wurde), stiege die Bildrate in der Palmenallee nur von 14 auf 18 fps.

Die Palmen kosten mehr als die ganze Straße.



Unterste Bildzeile: Nur die Mitte der Straße passt in eine Quellzeile. Fährt das Auto nach rechts, rückt Straße ins Bild, die WARPH nicht hat.

WARPH ist damit kein schlechter Befehl – er ist für eine Straße gemacht, die um ihn herum entworfen wird. ROAD folgt dem javascript-racer mit echten Segmenten, und die braucht es ohnehin für die Positionen der Palmen und Autos.

4. Was RC2 genau macht

- **Deko ohne SUB-Aufruf:** Breite und Ausrichtung jeder Palme werden einmal beim Laden ausgerechnet; beim Zeichnen bleiben nur wenige Befehle pro Sprite. Das Bild ist dasselbe.
- **Deko-Sichtweite automatisch:** Beim Laden zählt ROAD, wo mehr als 30 Deko-Sprites voraus liegen. Dort sinkt die Sichtweite von 80 auf 60 Segmente – 100 Segmente vorher beginnend und um ein Segment pro Bild, also ohne Sprung. Wer will, stellt das pro Strecke ein: `SCENERY, 80, 60, 30` in der .trk-Datei.

Messung (378 MHz)	RC1	RC2
Start Green Valley, Auto steht	14 fps	17–18 fps
Freie Strecke, Vollgas	18–20 fps	20–23 fps
Überfahrt der Ziellinie	13–14 fps	15–18 fps

5. Ideen für später

- **ROAD 1.1 – Ton mit Firmware 8:** Die neuen MOD-Funktionen von FW 8 (z. B. die Doppler-Demo) kommen in ROAD 1.1. Ab dann braucht ROAD Firmware 8; RC2 bleibt die Version für Firmware 7.
- **Ölflecken und Ähnliches:** geht schon heute – als flaches Sprite auf der Fahrbahn, ausgelöst wie die Schädel in Texas, nur mit Rutschen statt Crash.
- **Fahrbahn-Bilder** wie Zebrastreifen, Pfeile oder Risse mit richtiger Perspektive bräuchten Zeile für Zeile eigene Bilder – in BASIC zu langsam, mit WARPH-Straße oder C möglich.
- **„ROAD Turbo“ mit CSUB:** Straße und Sprite-Liste in C gerechnet – das sollte deutlich mehr fps bringen. Dann aber kein reines MMBasic mehr – deshalb, wenn überhaupt, als eigenes Experiment.
- **Eine klassische WARPH-Straße** als eigene Spielvariante: sehr schnell, aber mit anderem Blick (höhere Kamera, schmalere Straße).

Alle Werte auf echter Hardware gemessen (Pico Computer 3, 378 MHz, MODE 2), als Mittel über je eine Sekunde aus der Infozeile (Taste I). Have a good drive!