

HMI UART NDJSON – Protokoll + Integrationsanleitung

Getestet mit:

- WZ8048C070 – 7" Display
CrowPanel 7.0"-HMI ESP32 Display 800x480 RGB TFT LCD Touch Screen
- WZ8048C050 – 5" Display
CrowPanel 5.0"-HMI ESP32 Display 800x480 RGB TFT LCD Touch Screen

Schnittstelle: UART (8N1) + NDJSON (1 JSON-Objekt pro Zeile, endet mit `\n`)

Hinweis: Es wird **nur die Firmware** bereitgestellt, kein Quellcode.

1) Transport / Rahmen

- **UART:** 8N1 115200 Baud (Flow Control aus)
- **NDJSON:** jede Nachricht ist **ein JSON-Objekt pro Zeile** (endet mit `\n`).
- **UTF-8** für Text.

Pflichtfelder (Host → HMI)

- `seq` (int): fortlaufende Sequenznummer vom Host
- `cmd` (string): Befehl

Pflichtfelder (HMI → Host)

- `ack` (int) bei ACK (referenziert Host-`seq`)
 - `ok` (bool) bei ACK
 - oder `evt` (string) bei Events
-

2) Boot & Handshake

2.1 READY (HMI → Host)

HMI meldet sich nach Boot/Init:

```
{"evt": "sys", "state": "ready"}
```

Regel:

- **Vor** ```**` keine Commands senden******.

2.2 ACK (HMI → Host)

ACK OK:

```
{"ack":60,"ok":true}
```

ACK Error:

```
{"ack":60,"ok":false,"err":{"code":"bad_args","msg":"missing id"}}
```

Standard-Error-Codes:

- `bad_args`
- `unknown_cmd`
- `unknown_id`
- `oom`
- `internal`

3) Grundbefehle

3.1 Screen leeren (sichtbare Seite)

```
{"seq":10,"cmd":"cls"}
```

3.2 Objekt löschen

```
{"seq":11,"cmd":"del","id":123}
```

4) Pages (optional, für Fortgeschrittene)

4.1 Ohne Pages arbeiten (Standard / Anfänger)

Default-Verhalten:

- Alle Create/prop/del wirken **sofort auf die sichtbare UI**.
- Du bekommst direkt Feedback (sehr gut zum Debuggen).

➡ Das ist der empfohlene Start.

4.2 View + Edit parallel (Page-Buffering)

Ziel: Eine Seite bleibt sichtbar (**View**), während du eine andere im Hintergrund neu aufbaust (**Edit**). Danach schaltest du flackerfrei um.

page_begin (Edit-Page starten / leeren)

```
{"seq":30,"cmd":"page_begin","id":2}
```

- Setzt die **Edit-Page** auf `id`
- Löscht Inhalte dieser Edit-Page
- Ab jetzt beziehen sich Create/prop/del **standardmäßig auf Edit**

page_commit (Edit → View umschalten)

```
{"seq":31,"cmd":"page_commit"}
```

- Macht die aktuelle Edit-Page sichtbar (flackerfrei)

page_show (bereits aufgebaute Page anzeigen)

```
{"seq":32,"cmd":"page_show","id":1}
```

- Schaltet direkt auf eine vorhandene Page (wenn du Caching nutzt)

Optional: page (Kurzform)

Wenn du eine einfache „Seite neu“-Semantik willst (wie `cls` + Page-ID setzen):

```
{"seq":9,"cmd":"page","id":1}
```

Verhalten:

- Entspricht funktional `cls`
- Page-ID wird gesetzt (informativ)

Target-Override (auf View etwas hinzufügen, obwohl Edit aktiv ist)

Wenn du im Edit-Modus bist, aber **trotzdem auf der sichtbaren View** z. B. einen Button hinzufügen willst:

```
{"seq":50,"cmd":"btn","target":"view","id":99,"x":10,"y":300,"w":200,"h":50,"text":"Neu"}
```

- `target` ist optional.
 - Werte: `"view"` oder `"edit"`
 - Ohne `target`: Default hängt davon ab, ob gerade `page_begin` aktiv ist.
-

5) Commands (alphabetisch)

arc

Kurz: Interaktiver Bogen-/Drehregler.

Pflichtfelder:

- `id` (int)
- `x,y,w,h` (int)
- `min` (int)
- `max` (int)
- `val` (int)

Optionale Felder:

- `start` (int, Default: 0)
- `end` (int, Default: 360)
- `rot` (int)
- `mode` ("normal" | "reverse" | "sym")
- `width` (int 1..64)
- `color` ("#RRGGBB" oder int)
- `opa` (int 0..255)

Beispiel:

```
{"seq":140,"cmd":"arc","id":95,"x":20,"y":260,"w":160,"h":160,"min":0,"max":100,"val":50,"start":135,"end":45,"width":6,"color":"#00FF00"}
```

Event:

```
{"evt":"arc","id":95,"action":"changed","val":73}
```

btn

Kurz: Klickbarer Button.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `text`

Optionale Felder:

- Styles über `prop` (bg, pressed, radius, font)

Beispiel:

```
{"seq":21,"cmd":"btn","id":2,"x":10,"y":60,"w":200,"h":50,"text":"OK"}
```

Event:

```
{"evt":"btn","id":2,"action":"clicked"}
```

chk

Kurz: Checkbox.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `text`
- `checked` (bool)

Beispiel:

```
{"seq":24,"cmd":"chk","id":5,"x":10,"y":225,"w":300,"h":40,"text":"DHCP  
aktiv","checked":true}
```

Event:

```
{"evt":"chk","id":5,"action":"toggled","checked":false}
```

cls

Kurz: Löscht alle Objekte der aktuellen Zielseite (View oder Edit).

```
{"seq":10,"cmd":"cls"}
```

dd

Kurz: Dropdown-Auswahl.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `options` (String, getrennt mit `|`)
- `idx` (int)

Beispiel:

```
{"seq":23,"cmd":"dd","id":4,"x":10,"y":170,"w":300,"h":45,"options":"Auto  
Manuell  
Aus","idx":0}
```

Event:

```
{"evt":"dd","id":4,"action":"changed","idx":1,"text":"Manuell"}
```

del

Kurz: Löscht ein Objekt per ID.

Pflichtfelder:

- `id`

```
{"seq":11,"cmd":"del","id":123}
```

input

Kurz: Texteingabefeld.

Pflichtfelder:

- `id`
- `x,y,w,h`

Optionale Felder:

- `text`
- `placeholder`
- `pw` (bool)
- `one_line` (bool)
- `maxlen` (int)
- `kb` ("text" | "hex" | "ip")
- `accepted` (string)
- `clear_on_focus` (bool)
- `cursor_on_focus` (bool)

Beispiel:

```
{"seq":60,"cmd":"input","id":20,"x":20,"y":60,"w":360,"h":50,"placeholder":"IP","one_line":true,"maxlen":15,"kb":"ip"}
```

Event:

```
{"evt": "input", "id": 20, "action": "submit", "len": 12, "text": "192.168.0.10"}
```

label

Kurz: Textanzeige.

Pflichtfelder:

- `id`
- `x, y, w, h`
- `text`

Optionale Eigenschaften (über `prop`):

- `font` (int): Schriftgröße des Labels
Unterstützte Größen: `14, 16, 20, 24, 28, 32`

Beispiel (Label erstellen):

```
{"seq": 20, "cmd": "label", "id": 1, "x": 10, "y": 10, "w": 300, "h": 40, "text": "Hello"}
```

Beispiel (Fontgröße setzen):

```
{"seq": 21, "cmd": "prop", "id": 1, "font": 24}
```

line

Kurz: Zeichnet eine Linie.

Pflichtfelder:

- `id`
- `x1, y1, x2, y2`

Optionale Felder:

- `width`
- `color`
- `opa`

Beispiel:

```
{"seq": 200, "cmd": "line", "id": 50, "x1": 10, "y1": 20, "x2": 300, "y2": 200, "width": 3, "color": "#FF0000"}
```

lchart

Kurz: Einfaches Liniendiagramm.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `min`
- `max`

Optionale Felder:

- `points`
- `data` (CSV)

Beispiel:

```
{"seq":110,"cmd":"lchart","id":92,"x":20,"y":110,"w":420,"h":180,"min":0,"max":100,"points":5,"data":"10,20,30,25,40"}
```

mbox

Kurz: Meldungsbox.

Pflichtfelder:

- `id`
- `text`

Optionale Felder:

- `title`
- `buttons`
- `w,h`

Beispiel:

```
{"seq":33,"cmd":"mbox","id":15,"text":"Speichern?","title":"Wissen","buttons":"Abbruch\nOK","w":400,"h":200}
```

Event:

```
{"evt":"msgbox","id":90,"action":"button","idx":0,"text":"OK"}
```

page / page_begin / page_commit / page_show

Siehe Abschnitt **Pages (optional)**.

prop

Kurz: Ändert Eigenschaften eines bestehenden Objekts.

Pflichtfelder:

- `id`

Unterstützt:

- Werte: `text`, `font`, `val`, `checked`, `idx`, `options`
- Style: `color`, `bg`, `bg_opa`, `border`, `border_color`, `radius`
- Position: `x_ofs`, `y_ofs`, `align`
- Button-States: `btn_bg`, `btn_bg_opa`, `btn_bg_pressed`, `btn_bg_pressed_opa`
- Line: `width`, `color`, `opa`, optional `x1,y1,x2,y2`
- Arc: `val`, `min`, `max`, `rot`, `start`, `end`, `mode`, `width`, `color`, `opa`

Fontgrößen (`font`): 14, 16, 20, 24, 28, 32 (z. B. für Label und Button-Text)

Beispiel:

```
{"seq":4,"cmd":"prop","id":1,"text":"Update OK"}
```

roller

Kurz: Scroll-Auswahl.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `options`

Optionale Felder:

- `idx`
- `rows`

Beispiel:

```
{"seq":120,"cmd":"roller","id":93,"x":20,"y":310,"w":200,"h":120,"options":"Low\nMid\nHigh",  
"idx":1,"rows":3}
```

Event:

```
{"evt":"roller","id":93,"action":"changed","idx":2,"text":"High"}
```

bar

Kurz: Kompakte Anzeige-Bar.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `val`

Optionale Felder:

- `min`
- `max`

Beispiel:

```
{"seq":22,"cmd":"bar","id":21,"x":50,"y":220,"w":700,"h":40,"min":0,"max":100,"val":50}
```

slider

Kurz: Interaktiver Slider.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `min`
- `max`
- `val`

Beispiel:

```
{"seq":3,"cmd":"slider","id":3,"x":50,"y":220,"w":700,"h":40,"min":0,"max":100,"val":50,"live":true}
```

Event:

```
{"evt":"slider","id":3,"action":"changed","val":73}
```

spin

Kurz: Spinbox.

Pflichtfelder:

- `id`
- `x,y,w,h`
- `min`
- `max`
- `val`

Optionale Felder:

- `step`
- `digits`
- `dec`

Beispiel:

```
{"seq":130,"cmd":"spin","id":94,"x":240,"y":310,"w":220,"h":60,"min":0,"max":9999,"val":123,"step":1,"digits":4,"dec":0}
```

Event:

```
{"evt":"spin","id":94,"action":"changed","val":124}
```

SW

Kurz: Switch Ein/Aus.

Pflichtfelder:

- `id`
- `x,y`
- `checked`

Beispiel:

```
{"seq":13,"cmd":"sw","id":6,"x":110,"y":310,"checked":false}
```

Event:

```
{"evt":"sw","id":6,"action":"toggled","checked":false}
```

theme

Kurz: Setzt globale Farben.

Pflichtfelder:

- `bg`
- `text`

```
{"seq":40,"cmd":"theme","bg":"#101820","text":"#E0E0E0"}
```

6) Typischer Ablauf (Beispiele)

6.1 Minimal (ohne Pages)

1. Auf `ready` warten
2. Screen löschen
3. UI erstellen

```
{"seq":10,"cmd":"cls"}  
{"seq":11,"cmd":"label","id":1,"x":10,"y":10,"w":300,"h":40,"text":"Hallo"}  
{"seq":12,"cmd":"btn","id":2,"x":10,"y":60,"w":200,"h":50,"text":"Start"}
```

6.2 Fortgeschritten (Edit im Hintergrund, dann commit)

```
{"seq":30,"cmd":"page_begin","id":2}  
{"seq":31,"cmd":"cls"}  
{"seq":32,"cmd":"label","id":1,"x":10,"y":10,"w":300,"h":40,"text":"Neue Seite"}  
{"seq":33,"cmd":"btn","id":2,"x":10,"y":60,"w":200,"h":50,"text":"OK"}  
{"seq":34,"cmd":"page_commit"}
```

In diesem Beispiel wirkt `cls` auf die **Edit-Page**, weil `page_begin` aktiv ist.

7) Best Practices

- Nach `ready` einmal sauber initialisieren (Theme, cls, UI)
- IDs sauber verwalten (keine wilden Doppel-IDs)
- ACKs auswerten und Fehler loggen
- Events sind asynchron → nie mit ACK beantworten
- Pages nur nutzen, wenn du wirklich flackerfrei umschalten willst