

Erläuterungen zur UTM-Anzeige von StefanDausR zusammengestellt von George 2007

Voraussetzung

Es wird ein virtueller COM-Port-Treiber benötigt, wie z.B. VirtCom oder Serilot.

Zur Installation:

- 1) Falls noch nicht geschehen, einen virtuellen COM-Port-Treiber installieren, so wie er z.B. beim POIObserver dabei ist. Es wird nur der Treiber benötigt, nicht das Programm!
- 2) Den GPS-Port von COM2: auf z.B. COM5: umleiten und GoPal dann auch auf den virtuellen COM5: einstellen.
- 3) Die neue UTM-Anzeige liest nur den NMEA-Datenstrom mit und schreibt **keine** Daten auf den COM-Port. Es wird also der GPS-Empfänger **nicht** initialisiert! Das bedeutet, dass GoPal laufen muss, damit die Anzeige funktioniert!

Programmaufruf

Über das Koordinatenprogramm (ab Version 1.093) können eine Vielzahl von GPS-Informationen in Menü- und Ansichtsfenstern eingeblendet werden. Um dies zu erreichen, muss man das Programm Koord465 zusammen mit den unten erläuterten Parametern aufrufen. Dies kann z.B. über einen neuen Button in dem Original-Skin erfolgen und so aussehen:

```
launch: "\My Flash Disk\Programme\koord465.exe";  
params: "hier folgen dann die Parameterzeilen, die unten erläutert sind";
```

Da recht lange Parameterzeilen notwendig sind, wenn man verschiedene Informationen in verschiedenen Fenstern einblenden möchte, wurde der Befehl Include= eingeführt, der dann auf eine Text-Datei verweisen kann, in der die Parameter enthalten sind:

```
launch: "\My Flash Disk\Programme\koord465.exe";  
params: "Include=\hier folgt dann der Pfad zu der Text-Datei";
```

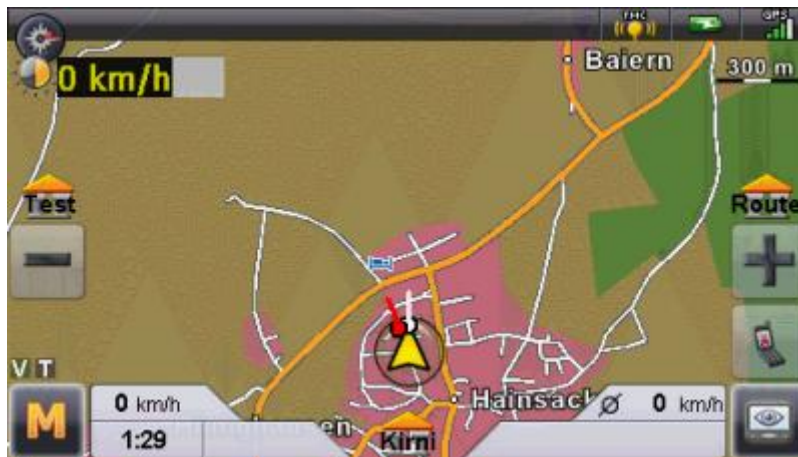
Eine ausführliches Beispiel für die Verwendung des Include= Befehls ist weiter unten zu finden.

Anzeigeparameter



Zum Bild in der Kompass-Ansicht gehören folgende Parameter: (alles in einer Zeile!)

```
Window=0 KoordServ=0 ServWnd=utm,80,5,180,20,-20,255,50,100\  
orthodrome,5,35,150,30,-30,0,0,128{%.3fkm}\  
direction,5,60,150,16,16,0,0,128\wgs84_s,200,35,200,45,20,0,0,100\  
target_d,50,215,200,16,-16,0,100,0{Ziel:%f° %f°}\ | t=1 GPSPort=com5:
```



Die Anzeige der Geschwindigkeit in der MapView erfolgt so:

```
Window=0 KoordServ=1 ServWnd=speed,30,30,100,24,-24,200,200,0{%.0f km/h}\ |  
t=0 rb=20 gb=20 bb=20 GPSPort=com5:
```

Erläuterungen

Man muss `Window=0` angeben, damit das Fenster des Koordinatenprogramms bei dem Aufruf **nicht** geöffnet wird.

`KoordServ=1` bewirkt, dass ein Fenster (bzw. Fensterausschnitte) aufgemacht wird, das immer im Vordergrund liegt. Auf dieses Fenster wird gezeichnet. Das hat den Vorteil, dass die Ausgaben nicht durch GoPal gelöscht werden können.

`KoordServ=0` bewirkt, dass direkt auf den Bildschirm gezeichnet wird. Somit kann der Hintergrund transparent (`t=1`) gemacht werden. Der Nachteil ist ein mehr oder weniger starkes Flackern der Texte. Per Default werden die Texte alle 500ms neu gezeichnet, ich weiß aber nicht, wann GoPal den Bildschirm updatet.

Mit `servwnd=` kann man nun die einzelnen Ausgabe(-teil)-fenster definieren:
Die Teil-Fenster werden durch einen Backslash "\" voneinander getrennt.
Der Befehl selbst muss mit dem "Pipe"-Zeichen (senkrechter Strich) "|" abgeschlossen werden.

Ein Teilfenster besteht aus diesen, durch Komma "," getrennten Pflichtangaben

```
speed,30,30,100,24,-24,200,200,0{% .0f km/h}
```

- Name
- x-Koordinate
- y-Koordinate
- Fensterbreite
- Fensterhöhe
- Schrifthöhe (negativ=Fett)
- Rotanteil der Schriftfarbe
- Grünanteil der Schriftfarbe
- Blauanteil der Schriftfarbe

Danach kann noch ein selbstdefinierter Formatstring (printf-Format in "C") folgen, der in geschweiften Klammern stehen muss. Dabei muss aber unbedingt die vorgegebene Reihenfolge und der Typ der Parameter eingehalten werden! Man kann aber z.B. die Anzahl der Nachkommastellen variieren oder Texte einfügen.

Diese Ausgabemöglichkeiten (=Name) gibt es momentan. Der hier angegebene Formatstring ist der Defaultwert:

utm -> UTM Koordinaten (4 Felder): %i %c %07.0f %07.0f

speed -> GPS-Geschwindigkeit: %.1f km/h

course -> GPS-Kurs (aktuelle Abweichung von Nord): Kurs:%.1f°

wgs84_d -> aktuelle Koordinate in Dezimalschreibweise (2 Felder): Lat=%.6f Lon=%.6f°

target_d -> Ziel-Koordinate in Dezimalschreibweise (2 Felder) (aus der Registry "DestLatitude" und "DestLongitude"): tLa=%.6f tLo=%.6f°

orthodrome -> Luftlinie (genau gerechnet!) zwischen Ziel und akt. Position: LL=%.3fkm

bearing -> Peilung zum Ziel (aktuelle Abweichung von Nord): Peilung:%.1f°

direction -> aktuelle Richtung zum Ziel (bearing - course) (als 2. Feld wird noch die "Marschzahl" angegeben): Richtung: %.1f° MZ %.0f

wgs84_s -> aktuelle Koordinaten in Himmelsrichtung, Grad, Minute, Sekunde (dez.) (8 Felder): %c %02i° %02i' %04.1f''\n%c %03i° %02i' %04.1f''

wgs84_m -> aktuelle Koordinaten in Grad, Minute (dez.), Himmelsrichtung (6 Felder): %02i° %07.4f' %c\n%03i° %07.4f' %c

mgrs -> MGRS-Koordinaten: %c

Weiteres Beispiel



Beispiel (s. Bilder):

- In der **MapView** sollen Entfernung zum Ziel (Luftlinie) und geographische Koordinaten angezeigt werden



- In der Kompass-Ansicht sollen Entfernung zum Ziel (Luftlinie) und UTM-Koordinaten angezeigt werden
- In allen anderen Ansichten soll nichts erscheinen

Mit diesem einen Parameter ist das (im orig. Skin!) möglich: (alles in einer Zeile!)

```
Window=0 KoordServ=1  
PixWnd=0,78,257,0,0,0\0,82,257,0,0,0\1,12,12,0,0,0,12,21,0,0,0\  
ServWnd=utm,80,5,180,20,20,255,50,100,0x4\  
orthodrome,5,35,150,30,-30,0,0,128,0xF{% .3fkm}\  
wgs84_s,200,35,130,45,20,0,0,100,0x3\ t=1 GPSPort=com5:
```

Mit **PixWnd=** kann man die (GoPal-)Fenster per GetPixel abfragen.

Der erste Wert legt eine (selbst definierte) Fensternummer fest. Ich habe **0** für die **MapView** und **1** für die **Kompassansicht** genommen.

Der 2. und 3. Wert sind die x- und y-Koordinaten des abzufragenden Pixels.

Der 4. bis 6. Wert stellt den Soll-RGB-Wert des Pixels dar.

Optional kann noch ein zweiter Pixel definiert werden (dann müssen beide Pixel die angegebenen Farben haben):

Der 7. und 8. Wert sind die x- und y-Koordinaten des zweiten abzufragenden Pixels

Der 9. bis 11. Wert stellt den Soll-RGB-Wert des zweiten Pixels dar.

Die einzelnen Fenster werden mit "\ " beendet. Der Parameter selbst mit "\|"

Hinweis: Die Fenster-Nummer (1. Wert) kann auch mehrmals vorkommen, dann entspricht das einer "ODER-Bedingung" (z.B. für Sonnenbrille)

3) Um den Ausgabe-Werten, die ja mit **servWnd=** definiert werden, das oder die Fenster (PixWnd=) zuzuordnen, kann noch ein weiterer Wert angegeben werden. Dieser ist folgendermaßen Bit-codiert:

Ein Fenster besteht aus **2 bit**.

Ist das niederwertige bit gesetzt, so muss GPS-Empfang vorliegen.

Ist das höherwertige bit gesetzt, so darf kein GPS-Empfang vorliegen.

Sind beide bits gesetzt, so ist es egal, ob ein GPS-Signal anliegt oder nicht!

Ist kein bit gesetzt, so erfolgt auch keine Anzeige.

Das Fenster mit der (mit PixWnd= definierten) Nummer 0 wird durch die ersten beiden bits (0 und 1) repräsentiert. Das Fenster 1 gehört zu bit 2 und 3 usw.

Erläuterung am obigen Beispiel:

Die MapView erkenne ich an dem schwarzen Doppelpunkt in der Uhrzeit. Ich nehme hier nur den oberen Punkt.

Die Position des Punktes ist aber abhängig von der Stellenanzahl der Stunden, daher habe ich 2 verschiedenen Koordinaten definiert:

```
PixWnd=0,78,257,0,0,0\0,82,257,0,0,0\
```

MapView sei Fenster 0

Das schwarze (0,0,0) Pixel des Doppelpunktes sei entweder bei 78,257 oder 82,257

Das Kompass-Fenster erkenne ich am schwarzen K in der Titelzeile:

```
PixWnd=1,12,12,0,0,0\12,21,0,0,0\
```

Kompass-Fenster sei Fenster 1

Es muss ein schwarzes (0,0,0) Pixel sowohl auf 12,12 als auch auf 12,21 sein.

Diese Ausgaben sollen erfolgen (ServWnd=):

```
wgs84_s,200,35,130,45,20,0,0,100,0x3
```

0x3 ist hexadezimal und stellt die bit-Folge 00000011 dar. Das bedeutet, dass die Anzeige nur im Fenster 0 (MapView) erfolgen soll und es egal ist, ob ein GPS-Signal anliegt.

```
utm,80,5,180,20,20,255,50,100,0x4
```

0x4 stellt die bit-Folge 00000100 dar. Das bedeutet, dass die Anzeige nur im Fenster 1 (Kompass) erfolgen soll und dass ein GPS-Signal anliegen muss.

```
orthodrome,5,35,150,30,-30,0,0,128,0xF
```

0xF (=15) stellt die bit-Folge 00001111 dar. Das bedeutet, dass die Anzeige im Fenster 0 und 1 erfolgen soll und es egal ist, ob ein GPS-Signal anliegt.

In allen anderen Fenstern erfolgt keine Ausgabe!

Frage: *Was müsste man denn für Fensteransicht nur bei GPS-Empfang anfügen?*

Antwort:

Also wenn Du nur ein Fenster hast (z.B. die MapView) und diesem die Nummer 0 gegeben hast, dann musst Du **0x1** oder auch einfach 1 angeben.

Hast Du noch ein zweites Fenster definiert (z.B. Kompass mit Nummer 1) und du willst Die Anzeige nur auf diesem Fenster haben, dann wäre die **0x4** oder auch einfach 4 anzugeben.

Willst Du nun dieselbe Anzeige auf beiden Fenstern haben (und nur dort), dann musst Du die beide Werte mit logisch "OR" verknüpfen, also 0x1 OR 0x4 ergibt 0x5

Ich versuche es hier nochmal in Binärschreibweise zu verdeutlichen:

PixWnd=0, ...

PixWnd=1, ...

fest programmierte Bitmasken:

00 -> keine Anzeige

01 -> Anzeige bei GPS

10 -> Anzeige bei keinem GPS

11 -> immer Anzeige

Zuweisung der Fenster:

_7 _6_ 5 _4 _3 _2 _1 _0 (Fensternummer)

00 00 00 00 00 00 00 00 (Bitdarstellung in ServWnd=...,0x...)

Im Beispiel:

nur Fenster 0 und nur bei GPS:

_7 _6_ 5 _4 _3 _2 _1 _0

00 00 00 00 00 00 00 01 = 0x1 = 1

nur Fenster 0 und 1 und nur bei GPS:

_7 _6_ 5 _4 _3 _2 _1 _0

00 00 00 00 00 00 01 01 = 0x5 = 5

Fenster 0 nur bei GPS und Fenster 1 immer:

_7 _6_ 5 _4 _3 _2 _1 _0

00 00 00 00 00 00 11 01 = 0xD = 13

Der Parameter **Include=**:

param= Include=\Storage Card\Programme\KOORD\FensterTest.cfg

bzw.

params:"Include=\Storage Card\Programme\KOORD\FensterTest.cfg";

lädt die angegebene Datei FensterTest.cfg (kann natürlich auch xyz.txt oder sonst wie heißen) zusätzlich in den Parameterstring.

Diese Datei kann zeilenweise aufgebaut sein, allerdings darf ein Befehl (noch) nicht über mehrere Zeilen gehen. Es dürfen aber natürlich mehrere Befehle in einer Zeile stehen!

Beispiel:

```
// KoordServer Include Testdatei
```

```
//
```

```
window=0
```

```
LKlick1=100 x1=300 y1=100
```

```
KoordServ=1 t=1
```

```
PixWnd=0,80,228,0,0,0\0,76,228,0,0,0\|
```

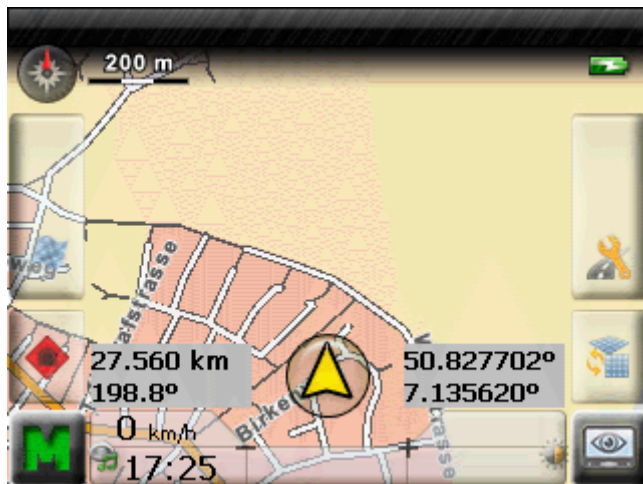
```
ServWnd=orthodrome,42,168,80,16,-16,0,0,0,0x1{%.3f
```

```
km}\direction,42,184,80,16,-16,0,0,0,0x1{%.1f°}\wgs84_d,198,168,80,32,-16,0
```

```
,0,0,0x1{%.6f°\n%.6f°}\|
```

```
GPSPort=com3:
```

Weitere Beispiele für 3 Zoll Geräte und Umsetzung in einem Skin



// dezimal Anzeige

//

window=0

LKlick1=100 x1=300 y1=100

KoordServ=1 t=1

PixWnd=0,80,228,255,255,255\0,76,228,255,255,255\|

ServWnd=orthodrome,42,168,80,16,-16,0,0,0,0x1{%.3f km}\direction,42,184,80,16,-

16,0,0,0,0x1{%.1f°}\wgs84_d,198,168,80,32,-16,0,0,0,0x1{%.6f°\n%.6f°}\|

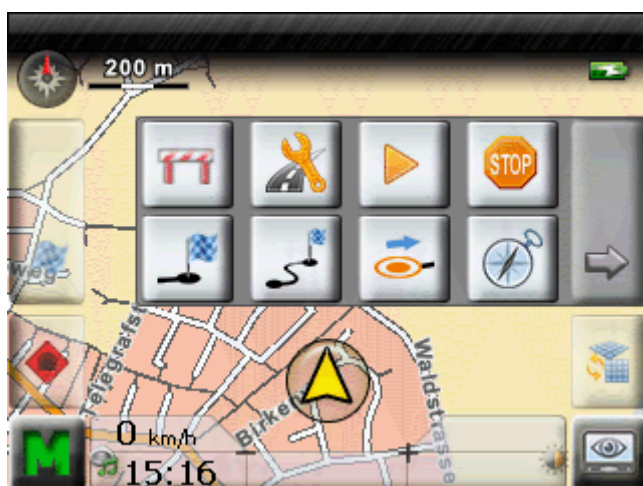
GPSPort=com3:

Hier wird auf einen virtuellen Port - durch einen Serilot ComPort-Splitter erzeugt - auf com3: zugegriffen.

Der Klickbefehl dient dazu das Fenster zu schließen, aus dem heraus die Anzeige aktiviert wird. Dies erfolgt über den Kompass-Button in der Abbildung und der Klick auf die Zurücktaste am rechten Ansichtsrand schließt das Fenster. Der Button löst folgenden launch Befehl aus:

launch: "\My Flash Disk\programme\koord465.exe";

params: "Include=\My Flash Disk\programme\koordanzeige.txt";



Um anstelle der Dezimal-Anzeige eine UTM-Koordinatenanzeige zu erreichen, muss der folgende Parameter Text aufgerufen werden:

```
// UTM Anzeige
//
window=0
LKlick1=100 x1=300 y1=100
KoordServ=1 t=1
WndRefresh=200
PixWnd=0,80,228,255,255,255\0,76,228,255,255,255\|
ServWnd=orthodrome,42,168,80,16,-16,0,0,0,0x1{%.3f km}\direction,42,184,80,16,-16,0,0,0,0x1{%.1f°}\utm,198,168,80,32,-16,0,0,0,0x1{%i%c%07.0f\n %07.0f}\|
GPSPort=com3:
```

Die Anzeigenart lässt sich über einen Button im Einstellungs Menü ändern:

```
launch: "\My Flash Disk\Programme\koord465.exe";
params: "window=0 copy=\My Flash Disk\programme\koordutm.txt|\My Flash Disk\programme\koordanzeige.txt|";
```

Das heißt hiermit wird nun der Inhalt der Parameter-Textdatei für die UTM-Anzeige 'koordutm.txt' in die Datei 'koordanzeige.txt' kopiert, welche durch den Kompassbutton in der Kartenansicht aufgerufen wird.



Bei der Umstellung zurück auf dezimal werden die zugehörigen Parameter, die in der 'koorddezimal.txt' abgelegt sind, in die koordanzeige.txt kopiert. Dies wird über einen entsprechenden Button ausgelöst (in der Abbildung liegt der Button unter dem Kompass).

```
launch: "\My Flash Disk\Programme\koord465.exe";
params: "window=0 copy=\My Flash Disk\programme\koorddezimal.txt|\My Flash Disk\programme\koordanzeige.txt|";
```