



Foto 1: MFH gegenüber der Schule

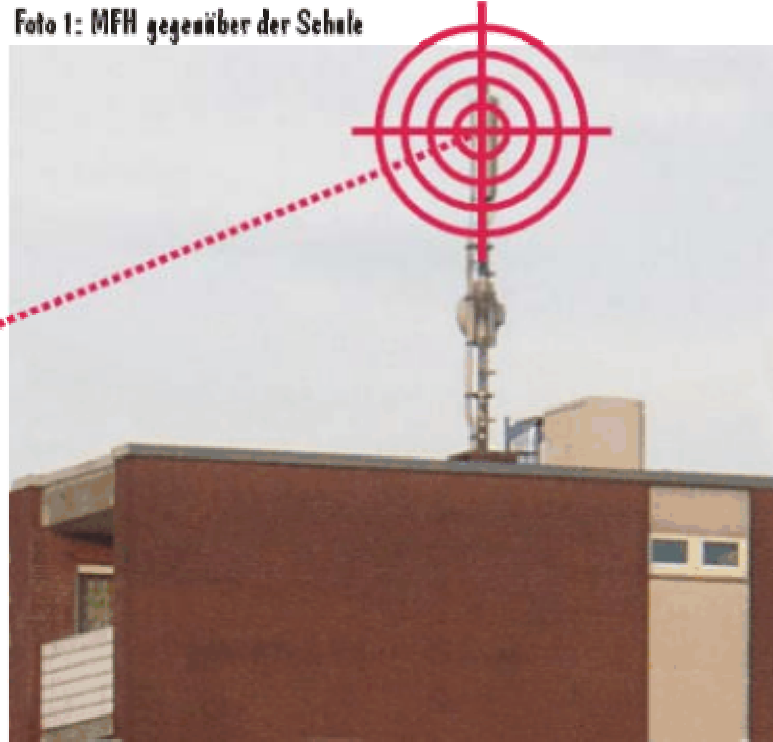


Foto 2: Mein Handy



Foto 3: Mein Pocket-PC

Standort:
Gebäude
gegenüber
"Gymnasium
in den
Filder Benden"

jugend  forscht 2004
schüler experimentieren

Lokale Ortung durch komplementäre Handy-Nutzung

Eingereicht von:

Florian Wetzel

Inhaltsverzeichnis

1. EINLEITUNG	3
2. DER BESONDERE KANAL	4
2.1 KANAL-CODIERUNG.....	4
3. SOFTWARE ZUR UMSETZUNG DES ORTUNGSKONZEPTES	6
3.1 VORÜBERLEGUNGEN	6
3.1.1 Auswahl der hardwarebestimmten Darstellungsplattform	6
3.1.2 Festlegung auf Programmiersprache	6
3.1.3 Anforderungen an das Weiterverarbeitungsprogramm	6
3.2 PROGRAMMERSTELLUNG	7
3.2.1 Struktur der erforderlichen Programmierschritte	7
3.2.2 Abruf der Ortungsdaten.....	7
3.2.2.1 Übermittlung von AT-Befehlen an das Handy	7
3.2.2.1.1 Schwierigkeiten bei der Kommunikation von PC und Handy	8
3.2.2.2 PDU-Format der weitergereichten CB-Nachrichten.....	8
3.2.2.2.1 Decodieren des PDU-Codes in der Theorie.....	9
3.2.3 Weiterverarbeitung der GK-Ortungsdaten.....	10
3.2.3.1 Umwandlung der Gauß-Krüger-Daten in Längen- und Breitengrade	10
3.2.3.2 Kartografische Darstellung der Längen- und Breitengrad-Daten.....	11
3.2.3.2.1 Anforderungen an das Kartenmaterial.....	11
3.2.3.2.2 Umrechnung der Positionsangaben in konkrete Pixelwerte.....	11
3.2.3.2.2.1 Ohne Zoom- und Verschiebefunktion:	11
3.2.3.2.2.2 Mit Zoom- und Verschiebefunktion:	12
3.2.3.2.2 Spezielle Probleme bei der grafischen Darstellung	12
4. ERGEBNISSE UND OPTIONEN	13
4.1 ERGEBNISDARSTELLUNG.....	13
4.2 ANWENDUNGSBEREICH DER ERGEBNISSE	14
4.3 FORTSCHREIBUNG DES PROGRAMMS.....	14
5. QUELLENVERZEICHNIS	15
5.1 INTERNETQUELLEN	15
5.2 BILDQUELLEN	15

1. Einleitung

Seit etwa drei Jahren besitze ich ein Handy mit einem Nutzungsvertrag bei O₂ („Otvo“ - früher Viag Interkom). Der Vertragstyp „Genion“ bietet mir einen besonders günstigen Tarif für einen Nahzonenbereich („Homezone“) von im Vertrag festgelegter räumlicher Größe.

Ich interessierte mich gleich für die komplexe Funktionsweise meines Handys und wollte verstehen, wie die Netzbetreiber feststellen, in welchem örtlichen Bereich ich mich gerade befinde, und woher mein Handy überhaupt weiß, dass ich mich in meinem Homezone-Bereich aufhalte.

(Mein Handy zeigt ein besonderes (Haus-)Symbol¹ an, dass mir signalisiert, dass ich mich im Homezone-Bereich mit dem vergünstigten Tarif befinde.)

Durch Internet-Recherche über das Zustandekommen dieser örtlichen Hausanzeige bin ich darauf gestoßen, dass der momentan in Kommunikation mit meinem Handy stehende Sendemast meines Netzbetreibers O₂ über seinen Standort Informationen übermittelt, welche vom Handy ausgewertet und durch das Haussymbol dargestellt werden.

Diese vom Handysendemast übermittelten Standortinformationen machten mich neugierig, ob man diese nicht nur netzbetreiberseitig sondern auch (handy-) **nutzerseitig** auswerten und anderweitig (kostenlos!) verwenden könne.

Dazu bedurfte es der Nachrecherche (u.a. im Internet), wie die Netzbetreiber diese Information an das Handy übermitteln. Dies geschieht ja ohne Wissen des Nutzers.

Wir Jugendlichen nutzen ja Handys u.a. auch dazu, SMS (Textmitteilungen mit bis zu 160 Zeichen) zu versenden. Ähnlich könnte dies ja auch mit den Standortinformationen geschehen, die evtl. als Textinformationen übertragen, vom Handy ausgewertet und dann grafisch sichtbar gemacht werden.

¹ siehe Photo 2 auf der Titelblattseite

2. Der besondere Kanal

Die Recherche ergab², dass die Standortinformationen durch den Netzbetreiber nicht via SMS, sondern über einen bestimmten Kanal des sogenannten CB-Dienstes (Cell Broadcast) erfolgt. Diese Dienste werden kontinuierlich und einseitig ausgehend von allen Sendemasten des Netzbetreibers in bestimmten kurzzeitigen Intervallen ausgestrahlt. Sie sind vergleichbar mit dem Videotext beim Fernseher.

O₂ bietet als einziger Netzbetreiber im Rahmen dieser Dienste u.a. einen Kanal (221) an, der die Standortinformationen ausstrahlt, die sich ja **nutzerseitig** kostenlos verwenden lassen müssten. Durch wenige Einstellungen an meinem Handy ließ sich der Inhalt der CB-Nachrichten dieses Kanals textlich auf dem Display darstellen. (siehe Foto 2 auf der Titelseite)

2.1 Kanal-Codierung

Der CB-Kanal 221 liefert die Standortinformationen in Form eines zwölfstelligen numerischen Codes, der einem um zwei Stellen gekürzten Gauß-Krüger-Code entspricht.

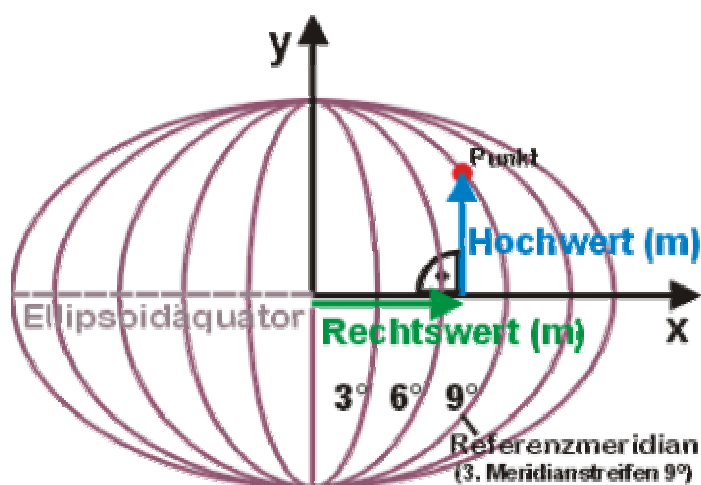


Bild 1: Ortsfestlegung bei O₂ nach Gauß-Krüger-Projektion

Der Gauß-Krüger-Code ist die Transformation einer geographischen Ortsangabe im Rahmen eines rechtwinklig metrischen Koordinatensystems³. Er besteht aus der Angabe des Wertes einer x-Achse, was der Kartenebene nach Osten entspricht (Rechtswert), und einer y-Achse, was der Kartenebene nach Norden ent-

spricht (Hochwert), sowie der Angabe eines bestimmten festgelegten Meridianstreifens der ellipsoidischen Erdkugel (siehe Bild 1). O₂ verwendet als Referenzmeridian für ganz Deutschland den Längengrad bei 9° östlicher Länge.⁴

Diese über den CB-Kanal 221 ausgestrahlten Nachrichten in Form 12stelliger Zahlen, sind in zwei Sechszifferngruppen teilbar. Die ersten 6 Zeichen gehören zum „Rechts-

² www.nobbi.com/glossar.htm; a. a. O.

³ siehe dazu Begriffserklärungen bei: www.geoinformatik.uni-rostock.de/einzel.asp?ID=743; a. a. O.

⁴ www.rschlichte.de/gsm/general.html; a. a. O.

wert“, die zweiten sechs zum „Hochwert“. Um jedoch einen echten Gauß-Krüger-Wert mit 14 Stellen zu erhalten, wird beiden Zahlensektoren eine 0 angehängt, wodurch eine Ortung auf 10 Meter Genauigkeit erreicht wird⁵.

Im Rahmen meiner Internet-Recherche stieß ich auf Internetseiten, die eine Umrechnung der O₂-Koordinaten (12stelliger Gauß-Krüger-Code) in die geografische Grad-Notation ermöglichen.⁶ Dies habe ich für mehrere mir bekannte Standorte (in Moers) stichprobenweise ausprobiert. Nach Einsetzen der zwölf Ziffern des O₂-Zahlencodes erhielt ich die korrekten Standortangaben als Dezimalzahl angezeigt.

Mir drängte sich dann die Frage auf, ob man diese Zahlenangabe nicht direkt durch ein Softwareprogramm weiterverarbeiten und anschließend sogar direkt kartografisch darstellen könne, also ohne Zwischenschritt über einen wie auch immer gearteten (kostenpflichtigen) Stadtplandienst.

⁵ Es werden allerdings nicht die wirklichen Koordinaten des Antennenmastes ausgestrahlt, sondern eine um 50 m in Abstrahlrichtung versetzte Position. Dies, um die drei 120°-Antennensegmente einer Basisstation abzugrenzen; siehe: www.heise.de/ct/99/18/174; a. a. O.

⁶ u. a.: www.gsmsite.de/viagkoordinaten.htm; a. a. O.

3. Software zur Umsetzung des Ortungskonzeptes

3.1 Vorüberlegungen

3.1.1 Auswahl der hardwarebestimmten Darstellungsplattform

Die Frage, ob das zur Darstellung erforderliche Programm auf dem Handy selbst implementiert werden kann oder dazu ein externer Computer erforderlich ist, wurde durch meinen persönlichen Ausstattungsfundus beantwortet: Mein Handy (Siemens S 45i) hat ein sehr kleines Display, das zudem nur in schwarz-weiß darstellt. Ihm fehlen auch die Speichermöglichkeiten für das umfangreiche Kartenmaterial, und seine Rechenleistung fällt gegenüber einem Computer geringer aus.

Ich besitze einen Kleinstcomputer („Pocket-PC“)⁷, der sich aufgrund seiner geringen Größe in Kombination mit einem Handy, geradezu als Plattform des zu erarbeitenden Programms anbietet, da das System unterwegs flexibel einsetzbar sein sollte.

3.1.2 Festlegung auf Programmiersprache

Die Weiterverarbeitung der vom Handy empfangenen Standortdaten auf dem Pocket-PC ließe sich mit C++ oder Visual Basic programmieren. Mit Visual Basic habe ich schon einen großen Erfahrungshorizont, während in den anderen Sprachen zeitaufwändiges Einarbeiten erforderlich würde.

Weil der Zeitrahmen für dieses Jugend forscht – Projekt zudem recht beengt ist, lag es nahe, die Weiterverarbeitung der Ortungsdaten mit Visual Basic zu programmieren.

3.1.3 Anforderungen an das Weiterverarbeitungsprogramm

Das u.a. mit Visual Basic umgesetzte Programm müsste folgende Anforderungen erfüllen:

- Erstens den von unterschiedlichen Handymodellen empfangenen um zwei Stellen reduzierten Gauß-Krüger-(GK-)Code abfragen,
- zweitens, mit Hilfe dieses Codes den aktuellen Standort auf einer auf dem (Pocket-) PC hinterlegten Karte beliebigen Abbildungsmaßstabs konkret sichtbar machen.

⁷ Siehe Foto 3 auf der Titelseite

3.2 Programmierstellung

3.2.1 Struktur der erforderlichen Programmierschritte

Anfänglich habe ich die Funktionsweise bezüglich der Koppelung meines Handys mit dem Pocket-PC analysiert. Erforderliche Arbeitsvorgänge wurden beschrieben und strukturiert. Das nachstehende Bild stellt diese Überlegungen schematisch dar.

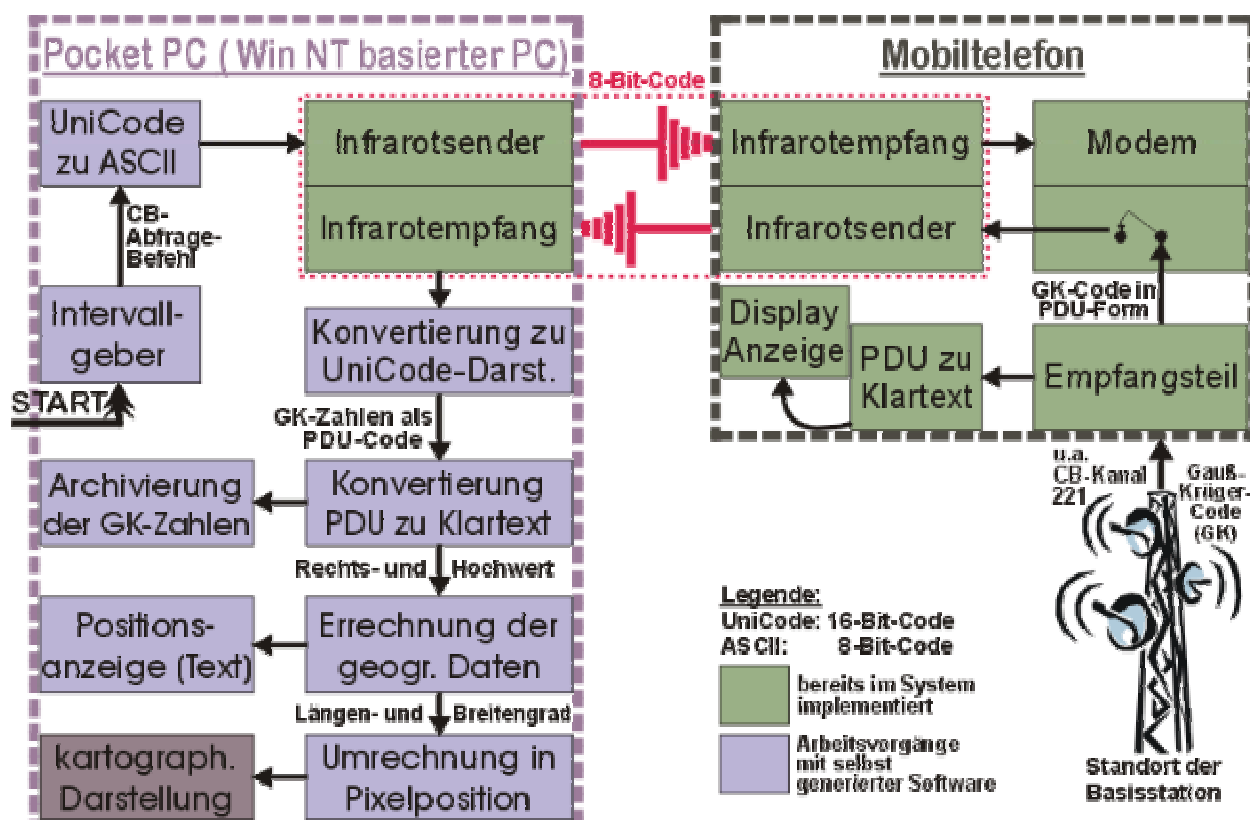


Bild 2: Schematische Darstellung der Arbeits- bzw. Programmierschritte der Handy-PC-Kommunikation
Im Folgenden werden die (z.T. neu aufgetretenen) Probleme beim Programmieren der einzelnen Arbeitsschritte und deren Lösung bzw. informatische Umsetzung dargelegt.

3.2.2 Abruf der Ortungsdaten

3.2.2.1 Übermittlung von AT-Befehlen an das Handy

Nahezu alle Modems, auch die in Handys integrierten, reagieren auf so genannte „AT-Befehle“ (AT = „attention“). Mit Hilfe spezieller nur für Mobiltelefone geltender AT-Befehle, lässt sich der Versand und Empfang von Nachrichten (z.B. SMS- oder CB-Nachrichten) steuern⁸. Somit ist es möglich, die vom Handy empfangenen Daten des CB-Kanals 221 auszulesen und extern auszuwerten.

⁸ zu den AT-Kommandos für GSM-Geräte: www.nobbi.com/atgsm.htm; a. a. O.

Die zwei Hauptbefehle für diesen Vorgang heißen: „at+cnmi=1,0,2,0,1“ und „at+cscb=0,“221“,“0““. Mit dem ersten Befehl weise ich das Handy an, CB-Nachrichten über das Modem an einen Computer weiterzugeben. Der zweite sorgt dafür, dass vom Handy der CB-Kanal 221 empfangen wird.

In meinem Programm werden noch viele weitere Befehle an das Handy übertragen, um so u.a. den Handyhersteller und das Handymodell abzufragen. Dies ist notwendig, weil sich die AT-Befehle gerätespezifisch leicht unterscheiden können, und das Programm dann eine Gerätemodellunterscheidung vornehmen muss.

3.2.2.1.1 Schwierigkeiten bei der Kommunikation von PC und Handy

Um Daten in meinem Pocket-PC über dessen serielle oder infrarote Schnittstelle empfangen und senden zu können, war eine Konvertierung erforderlich. Der Pocket-PC arbeitet betriebssystemspezifisch, wie alle WinNT[®]-basierten PC's, mit dem UniCode (16 Bit pro Zeichen)⁹. Das Handy jedoch kommuniziert stets über 8 Bits pro Zeichen.¹⁰

Bei der Programmierung dieser Konvertierung, konnte ich auf Visual Basic-Befehle zurückgreifen, die zumindest die Konvertierung einzelner Zeichen ermöglichen.

Die Rückantwort des Handys in 8-Bit-Form wird gleichermaßen durch mein Programm¹¹ wieder in den 16-Bit-UniCode transferiert.

Um einen möglichen Programmabsturz zu verhindern, der ausgelöst würde, wenn das Handy einen weiteren zusätzlichen Befehl erhielte, ohne, dass die vorgehende Kommandoausführung abgeschlossen wäre, war es notwendig, eine Art Warteschleife zu programmieren, in der die Befehle zwischengespeichert und dann nacheinander abgearbeitet werden. So kann auch bei gestörter Infrarot-Kommunikation der vom Handy unbeantwortete Befehl wiederholt gesendet werden.

Ist eine Maximalzahl von Sendeversuchen überschritten worden, können nicht zwingend erforderliche AT-Befehle (z.B. die, die einen akustischen Hinweis bewirken sollen) auch übersprungen werden. Dies erhöht die Kompatibilität mit anderen Handymodellen, deren Befehlssatz noch nicht vollständig in mein Programm aufgenommen wurde.

3.2.2.2 PDU-Format der weitergereichten CB-Nachrichten

Mobiltelefone empfangen Textnachrichten vom Netzanbieter generell im so genannten „PDU-Format“. Die im PDU-Code („Packet Data Units“) übermittelten Nachrichten las-

⁹ Moderne PC's nutzen diese Zuordnungstabelle um bis zu 32768 Zeichen darstellen zu können. (z.B. für Japanisch)

¹⁰ Diese acht Bits entsprechen dem erweiterten ASCII-Code, der 256 verschiedene Zeichen zuordnen kann.

¹¹ Wegen der Seitenzahlbegrenzung kann der Quellcode nicht abgedruckt werden. (Vorlage am Präsentationstag!)

sen sich im Gegensatz zum direkt lesbaren Klartext erst nach Anwendung eines Decodier-Algorithmus verständlich lesen. Da mein Handtyp, wie viele andere auch, Textnachrichten über das Modem nicht im decodierten Klartext übermitteln können, war ich gezwungen, mich mit diesem Protokoll näher vertraut zu machen.

Das PDU-Format dieser eintreffenden Nachrichten bzw. Daten beinhaltet einen angepassten (7-Bit-)ASCII-Code, der sich zu seiner Original-Definition insofern unterscheidet, dass selten benutzte Zeichen darin ersetzt werden durch Zeichen, die in ihm nicht implementiert sind, aber vom Benutzer benötigt werden (u.a. „ä“, „ö“, „ü“, „ß“).

In diesem angepassten ASCII-Code des Mobiltelefons besteht ein Zeichen aus 7 Bits. Der Netzbetreiber überträgt jedoch im Prinzip mit 8 Bit pro Zeichen. Überführt man nun jeweils acht 7-Bit-Zeichen mit einem festgelegten Algorithmus in sieben 8-Bit-Zeichen, so erspart sich der Netzanbieter ein zu übertragenes Byte. Die hexadezimale Darstellung dieser 8-Bit-Zeichen entspricht dem sogenannten PDU-Code.

3.2.2.2.1 Decodieren des PDU-Codes in der Theorie

Wie kann man nun den über den CB-Kanal 221 vom Handy empfangenen hexadezimalen PDU-Code für die Weiterverarbeitung auf meinem Pocket-PC dekodieren ?

Auf dem Handy-Display wird dieser ja schon zum Klartext entschlüsselt als ein

Transformation hexadezimal → dual	
Nibble	dual
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Tabelle 1

12stelliger Gauß-Krüger-Code dargestellt (siehe Photo 2 auf der Titelseite). Diese Zahl „333535570341“ lautet im vom Modem ausgegebenen PDU-Code: (...)“B3D9AC36ABD56EB0192D“(…) ¹²

Die Überführung dieser Hexadezimalzahl in den 8-Bit-Code möchte ich exemplarisch an den ersten 7 Zeichen aufzeigen:

Zunächst ist es notwendig, für jede Hexadezimalzahl (bestehend aus zwei „Nibbles“) unter Heranziehung der Tabelle 1 ¹³ ein Byte aus acht Bits zusammen zu setzen. So ergibt sich für das Hexadezimalzahl-Byte „B3“ mit den Nibbles „B“, „3“ zusammengesetzt der Bit-Code „10110011“.

Tabelle 2 zeigt die weiteren Ergebnisse.

Hexa-dezimal	Dual (8-Bit-Code)
B3	10110011
D9	11011001
AC	10101100
36	00110110
AB	10101011
D5	11010101
6E	01101110

Tabelle 2

¹² CB-Nachrichten werden stets mit Sonderzeichen auf 93 (7-Bit)Zeichen Länge aufgefüllt. Der hier abgedruckte, gekürzte PDU-Code zeigt jedoch nur relevante Nutzdaten. (siehe <http://www.nobbi.com/glossar.htm>)

¹³ Hier wird aus Erklärungsgesichtspunkten vereinfacht dargestellt. Mein Programm errechnet jedoch bei der Konvertierung von Zeichen- in Bit-Folgen (und umgekehrt) stets erst die Dezimalwerte.

In einem weiteren Schritt transformiert man die sieben 8-Bit-Bytes in acht 7-Bit-Bytes (Bild 3): Die Bits werden vom 8-Bit-Code zum 7-Bit-Code stets von links nach rechts übernommen. Nun braucht man nur noch die acht (7-Bit-)Dualzahlen entsprechend der ASCII-Tabelle als Zeichen (hier Ziffern) darstellen,



Bild 3: 8- auf 7-Bit- auf ASCII-Konvertierung

und schon haben wir die ersten acht Ziffern des Gauß-Krüger-Codes (s. rechts außen).

Um die nachfolgenden Zeichen zu erhalten, muss man erneut ein Paket aus sieben achtbittigen Bytes (auch „Octets“ genannt) nach dem gleichen Verfahren umwandeln. Die informatische Umsetzung der dargelegten Transformationen habe ich in Visual Basic vorgenommen.

Wegen des vorgegebenen Seitenumfangs kann ich den Quell-Code aber hier nicht abdrucken. Am Präsentationstag liegt dieser Quellcode natürlich zur Einsicht vor.

3.2.3 Weiterverarbeitung der GK-Ortungsdaten

3.2.3.1 Umwandlung der Gauß-Krüger-Daten in Längen- und Breitengrade

Da der Anwendungsbereich der Gauß-Krüger-Notation durch die Festlegung des Bezugssystems eingeschränkt ist, empfahl sich eine Konvertierung in geografische Gradangaben. Diese können dann allgemein verständlich weitergegeben werden und eignen sich besser für die kartografische Darstellung, da der größere Teil der Kartenmaterialien mit geografischen Gradangaben, und nicht mit GK-Codes versehen ist. Die dazu benötigte Transformationsformel des Gauß-Krüger-Codes in geografische Längen- und Breitengrad-Notation beinhaltet viele Differentialgleichungen.¹⁴ Hier galt es, eine möglichst einfache und doch noch ausreichend genaue Transformationsformel zu finden, die sich in mein Programm einbinden ließe.

Bei meiner Internetrecherche stieß ich auf einen in Delphi programmierten Programmcode, der diese Umwandlung übernehmen konnte.¹⁵ Da der Delphi-Code hauptsächlich mathematische Rechnungen beinhaltete war es leicht, ihn für Visual Basic anzupassen.

¹⁴ <http://www.gso-koeln.de/aktuell/archiv/forum/messages/630.htm>; a. a. O.

¹⁵ <http://www.tipps.delphi-source.de/mathematik/tut20020202-1.shtml>; a. a. O.

3.2.3.2 Kartografische Darstellung der Längen- und Breitengrad-Daten

3.2.3.2.1 Anforderungen an das Kartenmaterial

Es sind nur plan abbildende Karten verwendbar. Bei Karten, die z.B. in winkelscher Projektion abbilden, kann die in Kapitel 3.2.3.1.2 beschriebene Rechnung nicht angewendet werden. Des weiteren müssen die geografischen Gradangaben zweier festgelegter Eckpunkte bekannt und in einer Datei hinterlegt sein.

Um die Dateigrößen zu minimieren empfiehlt sich der Einsatz von Bitmap-Karten mit maximal 8-Bit Farbtiefe (256 verschiedene Farben). Da Bitmap-Karten jedoch im Gegensatz zu Vektorgrafiken nicht ohne Qualitätsverlust vergrößerbar („zoombar“) sind, sollte das Kartenmaterial von Anfang an der gewünschten Detailgetreue entsprechen. Bei der Erstellung solcher Bitmap-Karten soll der Benutzer auf sein eigenes, bereits vorhandenes Kartenmaterial zurückgreifen. Es können von Atlanten eingescannte oder von Routenplanern exportierte Karten, sogar Satellitenaufnahmen eingesetzt werden. Das einzig einschränkende Kriterium ist die Kenntnis der Gradmaße.

3.2.3.2.2 Umrechnung der Positionsangaben in konkrete Pixelwerte

3.2.3.2.2.1 Ohne Zoom- und Verschiebefunktion:



Bild 4: Bezugssysteme bei der Umrechnung in Pixelwerte

Will man eine Längen- und Breitengradposition auf einer plan abbildenden Karte darstellen, so benötigt man eine Art „Grad-Maßstab“ dieser Karte. Man muss also wissen, welchem Winkelunterschied des Längen- und Breitengrads es entspräche, eine bestimmte Strecke auf der Karte nach rechts oder oben zu gehen. Diesen Maßstab bestimmt mein Programm aus den Gradangaben zweier Eckpunkte (oben linke, unten rechte Ecke) sowie den Abmaßen der Karte. Der Pixelkoordinatenursprung und die Zählrichtung werden von Visual Basic leider ungünstig vorgegeben. (z.B. läuft die y-Koordinaten-Zählrichtung entgegen der der Breitengrade) (s. Bild 4) Um die x- Koordinate (horizontale Achse) eines Punktes zu erhalten, muss man den Grad-Maßstab mit dem entsprechenden Abstand des aktuellen

Längengrads eines virtuellen Gradmaßursprungs (unten linke Ecke der Karte) multi-

plizieren. Durch den Ansatz des virtuellen Ursprungs ist der Dreisatz anwendbar. Es ergibt sich daraus folgende Formel: („OL“ steht für „oben links - „UR“ für „unten rechts“)

$$xKoordinate = \frac{Kartenbreite}{LängengradUR - LängengradOL} \cdot (AktuellerLängengrad - LängengradOL)$$

Hier gilt: Je größer der aktuelle Längengrad, desto größer wird auch die x-Koordinate.

Für die y-Koordinate ergibt sich aufgrund der Richtung der Achse nach unten:

$$yKoordinate = \frac{Kartenhöhe}{BreitengradOL - BreitengradUR} \cdot (BreitengradOL - AktuellerBreitengrad)$$

Hier gilt je größer der Breitengrad, desto kleiner wird die y-Koordinate.

3.2.3.2.2 Mit Zoom- und Verschiebefunktion:



Bild 5: Bezugsgrößen bei Verschiebung

Funktionen wie die Verschieb- und Zoombarkeit der Karten werden heute vom Benutzer erwartet. Die Darstellung verschobener oder gezoomter Karten war kein großer Umstand, weil die Bildlade-Funktionen von Visual Basic sich dafür recht gut anwenden ließen. Doch dazu waren meine in Kapitel 3.2.1.2.1 hergeleiteten Formeln abzuändern:

$$xKoordinate = \left(\frac{Kartenbreite}{LängengradUR - LängengradOL} \cdot (AktuellerLängengrad - LängengradOL) + KartenStartX \right) \cdot ZoomFaktor$$
$$yKoordinate = \left(\frac{Kartenhöhe}{BreitengradOL - BreitengradUR} \cdot (BreitengradOL - AktuellerBreitengrad) + KartenStartY \right) \cdot ZoomFaktor$$

3.2.3.2.2 Spezielle Probleme bei der grafischen Darstellung

Gerade bei Verwendung großer Karten, aber auch beim Hervorheben der Schaltflächen (siehe Foto 3, Titelseite) wurde mir die Programmausführung zu langsam. Deswegen suchte ich im Internet nach einer Möglichkeit, direkter auf die Grafikkomponenten zugreifen zu können. Ich stieß auf einen Befehl namens „BitBlt“, der in Visual Basic selbst nicht implementiert ist, aber dies zu beschleunigen vermochte.¹⁶ Die grafische Anzeige des Herunterdrückes einer Schaltfläche dauerte nun nicht mehr Sekunden. Hier wurden mir die Leistungsgrenzen der Programmiersprache Visual Basic hinsichtlich der Ausführungsgeschwindigkeit bewusst.

¹⁶ http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_0fzo.asp; a. a. O

4. Ergebnisse und Optionen

4.1 Ergebnisdarstellung

Die von mir programmierte Software wurde auf verschiedenen Pocket-PCs (herauf bis Version 2003) erfolgreich getestet. Mit einigen Modifikationen ist das Programm auch auf herkömmlichen Windows NT®-basierten (inklusive 2000/XP) PC's lauffähig.

Das Programm forderte vom empfangsbereiten Handy permanent die Ortungsdaten der nächstgelegenen Basisstation ab und setzte sie in Form eines gelbfarbenen Punktes in einer auf dem Display sichtbaren Karte ein, die in unterschiedlichen Abbildungsmaßstäben zoombar ist (siehe Foto 3 auf der Titelseite).

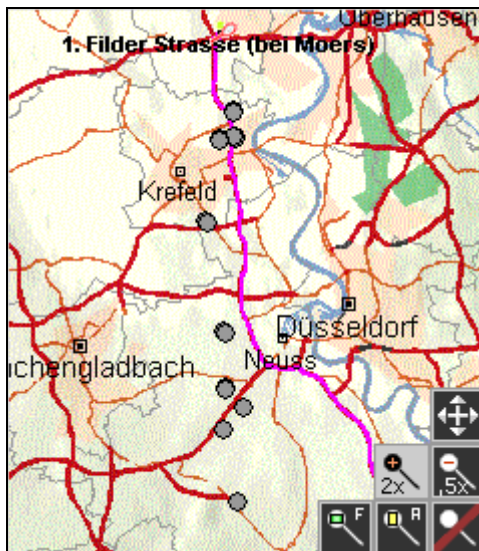


Bild 6: Beispiel für ein Bewegungsprofil durch Aneinanderreihung mehrerer Ortungspunkte

Durch die im Programm eingearbeitete Speichermöglichkeit mehrerer nach Ortswechsel abgerufener Basissenderdaten und anschließender gemeinsamer kartografischer Darstellung in Form graufarbener Punkte, lassen sich Bewegungsprofile sichtbar machen (siehe Bild 6 links).

Die Software erfüllt ihre Aufgabe auch bei anderen Handymodellen. Zwei meiner Freunde nutzen auch den Netzanbieter O₂ (vertragsgebunden bzw. durch eine Prepaid-Karte), besitzen aber Handys von Nokia. Die programmgesteuerte Ortungsdatenabfrage von deren Handys verlief reibungslos.

Bei weiter entfernten Fahrten mit meinem Vater testete ich auf dem Beifahrersitz das gebietsmäßig erweiterte Kartenmaterial erfolgreich. Auch hier wurden die abgefahrenen Basissenderstandorte in die Karte eingezeichnet. (sinng. wie Bild 6)

Da das Programm das Speichern bzw. Archivieren der abgerufenen Ortungsdaten vornimmt, um diese kartografisch darzustellen, war es zudem möglich, die interne Senderstandort-Datenbank in eine von Microsoft Excel® auswertbare Datei zu exportieren, die sowohl tabellarisch als auch grafisch z.B. als Fahrtennachweis nutzbar wäre. Das Programm kann auch als Navigationshilfe dienen: Man kann einen Zielpunkt festlegen (z.B. durch Tippen auf eine Stelle der Karte), wozu stets die direkte Luftlinienentfernung angezeigt wird (siehe Bild 7). Zusätzlich kann diese Strecke vom Programm errechnet und ausgewiesen werden.

4.2 Anwendungsbereich der Ergebnisse

Da nur O₂ den nutzerseitigen Empfang des Gauß-Krüger-Codes (Kanal CB 221) ermöglicht, reduziert sich folglich vorerst der Anwenderkreis meines Programms.

Des weiteren wird zukünftig durch den engmaschigeren Ausbau des Netzes der Basisstationen die Genauigkeit der Ortsangabe erhöht. Beim Trend nach Basisstationen mit geringeren Leistungen - das Thema „Elektrosmog“ sei hier angemerkt - wird die feinere Strukturierung des Netzes von O₂ auf längere Sicht ohnehin unumgänglich sein.

Wie in Bild 7 zu erkennen ist, testete ich das Programm auch erfolgreich auf sogenannte „Smartphones“, (Pocket-PCs mit Mobiltelefonfunktionalität). Bei solchen Geräten bietet sich die Anwendung meines Programms besonders an. *Es braucht einfach nur gestartet werden, ohne vorher die Hardware aus Hand und (Pocket-)PC verbinden zu müssen.*¹⁷



Bild 7: Aktuell auf den Markt gekommenes Smartphone von O₂

4.3 Fortschreibung des Programms

Auf solchen Smartphones könnten durch die ständig aufbaubare Internetverbindung, auch Karten von Web-Diensten (z.B. „MapQuest“, „Route24“) geladen werden.

Des weiteren ließe sich das Programm dahingehend fortentwickeln, dass externe dritte Personen das Bewegungsprofil, also die archivierten Routendaten, per SMS abfragen könnten. Den für den SMS-Versand notwendigen PDU-Encoder habe ich bereits programmiert.

Ich sehe hier die datenschutzrechtliche Brisanz dieses

Vorhabens und die immer wieder diskutierte Problematik „Technischer Fortschritt – Fluch oder Segen“. Die Standortbestimmung des Einzelnen kann unter Freunden von großem Nutzen sein, birgt aber auch stets die Gefahr des Missbrauchs in Form der absoluten Überwachung von Individuen durch u.a. Arbeitgeber, Staat, oder: Eltern ...

Mich motiviert jedoch alles technisch Machbare, was den Menschen hilft. Ich darf aber den Blick für die sozialen Folgen (auch meines Tuns) nie aus den Augen verlieren.

¹⁷ Leider hat O₂ den internen Zugriff auf das interne Modem über AT-Befehle noch nicht implementieren können. Es soll bald ein Firmware-Update geben, dass auch das ermöglichen soll.

5. Quellenverzeichnis

5.1 Internetquellen

1. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/gdi/bitmaps_0fzo.asp; Windows GDI – BitBlt; Ausdruck vom 05.01.2004
2. <http://rschlichte.de/gsm/general.html>; Reiner Schlichte: O2-Senderliste Baden-Württemberg, Senderkoordinaten; Ausdruck vom 20.12.2003
3. http://www.gate5.de/english/news/coverages/view_focus152003.html; Ulf Hannemann: Freunden auf der Spur, Die Mobilfunkbranche entwickelt neue Handy-Dienste; Ausdruck vom 22.12.2003
4. <http://www.geoinformatik.uni-rostock.de/einzel.asp?ID=743>; Geoinformatik-Service; Begriff: Gauß-Krüger-Projektion; Ausdruck vom 21.12.2003
5. <http://www.gsmsite.de/viagkoordinaten.htm>; Umrechnung der O2-Koordinaten; Ausdruck vom 21.12.2002
6. <http://www.gso-koeln.de/aktuell/archiv/forum/messages/630.htm>; Dirk Koopmann: Gauß-Krüger Koordinatensystem; Ausdruck vom 21.12.2003
7. <http://www.heise.de/ct/99/18/174>; Bögeholz, Harald/Zivadinovic, Dusan: Die Technik des Mobilfunknetzes von Viag Interkom; Ausdruck vom 20.12.2003
8. <http://www.nobbi.com/glossar.htm>; Das GSM – Glossar: Merkwürdige Begriffe verwirrend erklärt ..., Begriff: Cell Broadcast; Ausdruck vom 20.12.2003
9. <http://www.nobbi.com/atgsm.htm>; AT-Kommandos für GSM-Geräte; Ausdruck vom 22.12.2003
10. <http://www.tipps.delphi-source.de/mathematik/tut20020202-1.shtml>; Wie kann ich geographische in Gauß-Krüger-Koordinaten umrechnen?; Ausdruck vom 05.01.2004

5.2 Bildquellen

Photos und Abbildungen sind eigenerstellt