Visualisierung einer Wetterstation

mit

dem C++Builder



Technikerarbeit 2002

von Michael Wippl

Inl	Inhaltsverzeichnis 1				
1.	. Vorwort				
2.	Auf	gabenstellung	5		
3.	Def	Definition von Bestandteilen und Funktionsweise			
	eine	er Wetterstation	6		
	3.1	Basis-Station	7		
	3.2	Aufbau der Wetterstation	8		
	3.3	Messwertgewinnung	8		
	3.4	Framerate	9		
	3.5	Temperatur-Modul	9		
	3.6	Luftdruck-Modul	11		
	3.7	Geigerzähler-Modul	12		
	3.8	Windgeschwindigkeits-Modul	14		
	3.9	Windrichtungs-Modul	15		
4.	Dat	enspion	16		
	4.1	Anschluss und Einstellung	17		
	4.2	Schaltplan des Spions	18		
	4.3	Software	19		
	4.4	Anwendung	20		
	4.5	Schnittstelleneinstellung	20		
	4.6	Datenwandlung	22		

5.	Grundlagen der Entwicklungsumgebung			
	5.1	24		
	5.2	Symbolleiste	24	
	5.3	Komponentenleiste	25	
	5.4	Formulardesigner	26	
	5.5	.5 Objektinspektor Fenster		
	5.6	Fenster zur Projektverwaltung	29	
	5.7	Quelltextfenster	30	
	5.8	Festlegen der Programmiereigenschaften	31	
	5.9	Installation des Borland C++Builder	33	
6.	Vora	rbeiten zur Programmierung	34	
	6.1	Allgemein	35	
	6.2	Recherchen und Einarbeitung in die		
		Entwicklungsumgebung	35	
	6.3	Aufbau der Wetterstation	36	
	6.4	Abhören der Schnittstelle	38	
	6.5	Die Spionage beginnt	39	
	6.6	Bytezuweisung	40	
	6.7	Tabellarische Festlegung der Messwerte	41	

7.	Das Progra	nmm	43
-	7.1 Allgemei	n	44
-	7.2 Messwer	tanzeige Beschreibung	45
	7.2.1	Luftfeuchtigkeit	46
	7.2.2	Temperatur	47
	7.2.3	Windrichtungsanzeige	48
	7.2.4	Windgeschwindigkeit	49
	7.2.5	Lichtstärke	51
	7.2.6	Gammastrahlung	52
	7.2.7	Luftdruck	53
	7.2.8	Kennliniendarstellung	54
	7.2.9	Ein / Ausschalter	56
	7.2.10	ComPort	58
8.	Schlusswo	rt	62
9.	Anhang		63

1. Vorwort

Diese Technikerarbeit bestand im Hauptziel darin, eine graphische Darstellung der Daten, welche an der Seriellen-Schnittstelle anliegen, zu erreichen. Dazu wurde der Borland C++Builder.5 verwendet. Es gab aber noch viele Unterziele bis das Hauptziel erreicht war.

- Zusammenbau der Wetterstation
- Funktionsweise der Wetterstation
- Abhören des Datenverkehrs auf der RS232
- Entwicklung eines Programms mit dem C++Builder

In der folgenden Dokumentation ich werde auf sämtliche Unterziele sowie auf das Hauptziel intensiv eingehen. Das betrieft den Lösungsweg zum Erreichen der einzelnen gesetzten Ziele sowie die Behebung der aufgetretenen Probleme.

An dieser Stelle möchte ich mich bei allen Personen bedanken, welche mich bei der Fertigung dieser Technikerarbeit unterstützt haben.

Marlen den 27.05.2002

Michael Wippl

2. Aufgabenstellung

- Analyse des Datenverkehrs auf der RS232 Dies geschieht mit Hilfe eines Schnittstellenspions,.der durch eine Delphi5 programmierte PC- Software unterstützt wird.
- Netzwerkprogrammierung.
- Programmieren mit C++Builder.
 Wird für die Schnittstelle benutzt.
- ❖ Visuelle Programmierung mit Abakus- Komponenten

Zusätzliche Aufgabenstellung

Datenlogger (Langzeitanzeige)
 Dadurch wird die Wetterabfrage von vergangenen Tagen möglich.

3. Definition von Bestandteilen und Funktionsweise einer Wetterstation

Zur Unterstützung dieser Definition wurden ausschließlich die mitgelieferten Kurzbeschreibungen sowie dass was auf der Internetseite von Conrad- Elektronix zu finden war verwendet. Die Bilder und Tabellen wurden selber erstellt.

3.1 Basis-Station

Die Wetterstation ist als modulares Telemetriesystem aufgebaut. Basierend auf dem Hauptmodul können durch Einstecken unterschiedlichster Sensormodule verschiedene Messgrößen am PC visualisiert werden. Dies kann durch Funk oder eine Leitungsverbindung auf die RS232 geschehen.

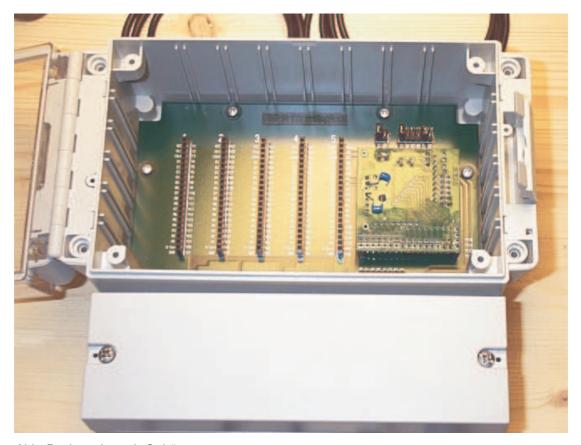


Abb. Basisstation mit Gehäuse.

3.2 Aufbau der Wetterstation

Das Hauptmodul der Wetterstation ist in einem Plastikgehäuse untergebracht, da je nach Anwendung ein sehr hoher Schutz vor Wasser und Staub gegeben sein muss. Im Innern der Wetterstation befindet sich die Busplatine.

Auf ihr ist das Hauptmodul, die Steckleisten für die Sensoren (S1 bis S5), der Schnitt-

stellenstecker (X4) und die Anschlussklemmen für die Versorgungsspannung (X5). Des weiteren sind einige Signale der Sensorsteckleisten herausgeführt, um externe Sensormodule anschließen zu können.

3.3 Messwertgewinnung

Auf dem Hauptmodul befindet sich ein Mikrocontroller mit acht 8-Bit-Analog-Digital Wandlereingänge, acht Digitalports und einem Zähler/Frequenzmesseingang. Die Messeingänge sind mit den Sensorsteckplätzen verbunden. Ein Sensormodul

wandelt physikalische Größen in
eine elektrische Spannung von 0
bis 2.5 Volt um. Diese Spannung wird
vom Analog-Digital-Wandler in einen
Wert von 0 bis 255 umgewandelt.
Alle Messwerte werden vom Hauptmodul zu einem Datenpaket zusammengefasst und an die serielle Schnittstelle ausgegeben. Über den Schnittstellenstecker gelangen die Datenpakete

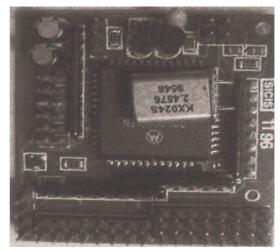


Abb. Mikrocontroller

per Draht- oder Funkverbindung an den PC, wo sie dann aufgezeichnet werden können.

3.4 Framerate

Mit den Jumpern F0 bis F2 an X1 wird die Framerate eingestellt. Sie ergibt sich aus dem Zeitabstand, in dem das Hauptmodul Messungen durchführt und die erfasten Werte zur Datenerfassung sendet. Verschiedene Rahmenabstände von einer halben Sekunde bis zu einer Stunde sind hier einstellbar. Die folgende Tabelle zeigt die unterschiedlichen Einstellungen.

F0	F1	F2	Rahmenabstand
1	1	1	0,5sek
0	1	1	1sek
1	0	1	10sek
0	0	1	30sek
1	1	0	1min
0	1	0	5min
1	0	0	10min
0	0	0	1Sth
Jumper gesetzt =1 // Jumper nicht gesetzt =0			

3.5 Temperatur-Modul

Auf der Temperatur-Modul-Steckkarte ebenfalls der Feuchtigkeitssensor sowie der Lichtstärkensensor in Form eines lichtempfindlichen Photoelement integriet. Das Temperatur-Modul muss bei abgeschalteter Betriebsspannung und trennung der Direktkabelverbindung zum PC, in die Steckleiste(S1 bis S5), welche sich auf der

Busplatine befindet, eingesetzt werden. Nach der Zuschaltung der Spannungsversorgung benötigt das Modul etwa eine Minute zur internen Stabilisation.

Deswegen muss beachtet werden, dass vorher keine gültigen Messwerte gewonnen werden können.

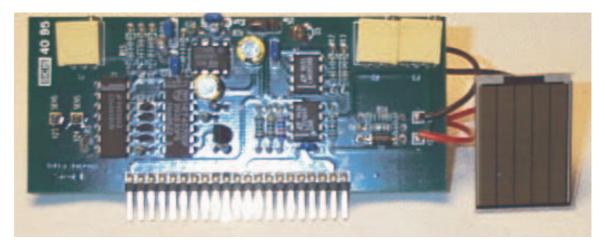


Abb.Temperatur-Modul

Neben der Wetterbeobachtung sind diese erfassbaren Größen auch in solchen Bereichen interessant, wo das Klima geschlossener Räume zu überwachen ist, wie beispielsweise in Museen oder Lagerräumen für Agrarprodukte.

Die physikalischen Größen Temperatur, Luftfeuchtigkeit und Lichtstärke beeinflussen die elektrischen Eigenschaften der entsprechenden Sensorbauelemente.

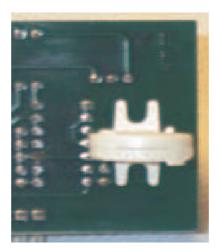


Abb. Feuchtsensor auf der Rückseite des Moduls

3.6 Luftdruck-Modul

Die elektronische Schaltung des Luftdruck-Moduls befindet sich auf der Steckkarte. Darauf befindet sich auch der eigentliche Sensor, welcher durch sein markantes Plastikgehäuse gut zu erkennen ist.

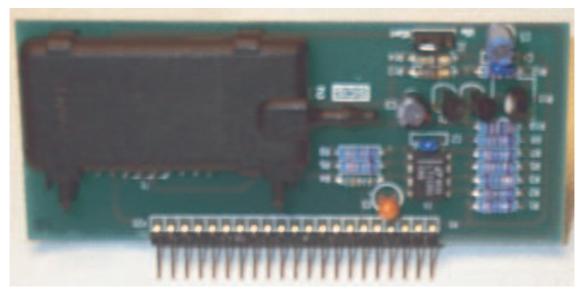


Abb. Luftdruck-Modul

Beim Einsetzen und Inbetriebnahme des Luftdruck-Moduls muss die selbe Vorgehensweise wie beim Temperatur-Modul eingehalten werden. Anders als bei dem Temperatursensor wird die Luftdruckänderung erst auf mechanischem Wege wahrgenommen. Dies geschieht durch eine Dehnungsmembrane Grundlage für dieses Messverfahren ist die Tatsache, dass bei der Dehnung eines Materials eine Querschnittverringerung stattfindet. Dadurch ergibt sich eine Widerstandsänderung.

Die gewohnte Einheit für den Luftdruck ist hPa(Hekto-Pascal).

3.7 Geigerzähler-Modul

Die elektronische Schaltung des Geigerzähler-Moduls ist mit einem Geiger-Müller-Zählrohr aufgebaut. Der Zählimpuls wird durch einfallende radioaktive Teilchen ausgelöst. Das Geigerzähler-Modul muss unter der selben Vorgehensweise wie beim Temperatur-Modul eingesetzt werden.

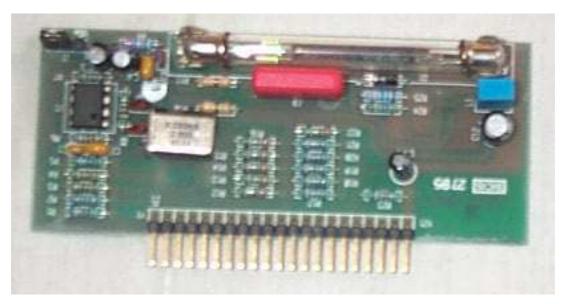


Abb. Geigerzähler-Modul

Maß für die Strahlungsbelastung eines Stoffes und auf die dabei auftretenden Absorption von Energie pro Masse ist die Energiedosis D mit der Maßeinheit Rad (engl. Radiation absorbed dose).

1 Rad = 1 rd 10-2 Joule pro Kilogramm

Die Wirkung radioaktiver Strahlung auf den menschlichen Organismus hängt neben der Energiedosis auch von der Energieverteilung innerhalb der Zellen ab. Als Ausdruck für die Schädlichkeit wird ein Bewertungsfaktor q verwendet. Für die einzelnen Strahlungsarten hat er folgende Werte:

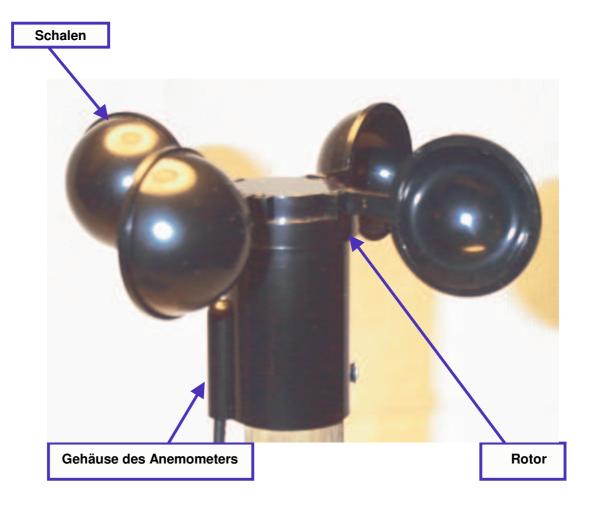
Strahlungsart	Bewertungsfaktor q
Alphastrahlung	10
Betastrahlung	1
Gammastrahlung	1
Röntgenstrahlung	1
Neutronenstrahlung	5-10
Protonenstrahlung	10-2

Abb. Strahlungsarten-Tabelle

3.8 Windgeschwindigkeits-Modul

Der Windgeschwindigkeitsmesser dient der Erfassung der momentanen Windgeschwindigkeit. Dieser Sensor ist zur Außenmontage vorgesehen. Der Windgeschwindigkeitsmesser besteht aus einem Schalenkreuzanemometer, welcher durch seine Schalen vom Wind in eine Drehbewegung versetzt wird.

Die Drehzahl des Rotors ändert sich proportional mit der Windgeschwindigkeit. Ein Magnet auf der Achse des Rotors schließt bei jeder Umdrehung kurzzeitig einen Reed-Kontakt, welcher im feststehenden Gehäuse des Anemometers eingebaut ist. Dieser Messwert wird also in Form von einem Impuls geliefert, dessen Frequenz ein Maß für Windgeschwindigkeit ist. Diese Impulsfrequenz wird ungerechnet und in der gewohnten Einheit km/h oder m/s angezeigt.



3.9 Windrichtungs-Modul

Dieser Sensor ist ebenfalls zur Außenmontage vorgesehen. Der Windrichtungsmesser besteht aus einer drehbar gelagerten Fahne. Durch die spezielle Formgebung richtet sie sich selbsttätig nach der jeweiligen Windrichtung aus. Die Achse der Windfahne ist mit dem Schleifer eines Potentiometers verbunden. Es handelt sich dabei um eine Sonderbauform ohne Anschlag, d.h. die Fahne lässt sich "durchdrehen".

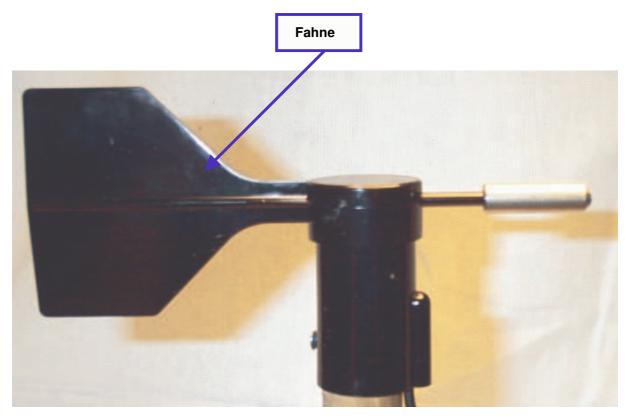


Abb. Windrichtungs-Modul

4. Der Datenspion

Für die Dokumentation des Datenspions wurden zwei DINA3 Blätter, welche ich von "Herr Schnaiter" erhalten habe, sowie die gesammelten Informationen die ich auf der Internetseite von der Firma Elektor gefunden habe, ausgewertet.

4.1 Anschluss und Einstellung

Mit diesem Datenspion kann der Datenaustausch zwischen zwei Hardware-Komponenten (w.z.B. PC und Drucker), aber nur in eine Richtung, ausspioniert werden. Diese Daten können dann wahlweise in ASC2, hexadezimal oder dezimal durch die Software Bin_Term auf dem Bildschirm dargestellt werden.

Für die Datenspionage werden lediglich die Datenleitungen von den zwei Hardware-Komponenten benötigt.

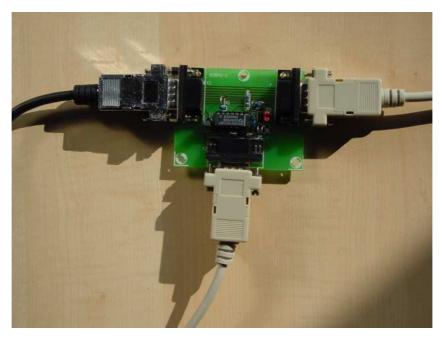


Abb. Datenspion

Die Schaltung ist wie ein Zwischenstecker mit zwei Sub-D-Steckverbindern ausgeführt, der zwei serielle Kabel miteinander verbindet. An einem dritten Sub-D-Steckverbinder wird der zusätzliche PC (Toshiba Satellite Pro 4600) angeschlossen, mit dem der Datenstrom analysiert wird.

Die Schaltung benötigt keine eigene Stromversorgung, sie bezieht ihren Betriebsstrom aus der RS232-Schnittstelle des die Daten abgreifenden PCs.

4.2 Schaltplan des Spions

Die Schaltung besteht aus nicht viel mehr als einem Flip-Flop(IC1) und ein paar passiven Bauteilen. Über K1/K2 läuft die serielle Verbindung durch die Schaltung, Daten und Steuerleitungen sind direkt durchverbunden. Werden von der PC-Seite oder von der Hardware-Komponente Daten empfangen, so wird das Flipflop (IC1) dementsprechend gesetzt oder zurückgesetzt. Über die DSR-Steuerleitung wird dem Datenspion die Datenrichtung angezeigt. Mit den Jumpern JP1 und JP2 kann ausgewählt werden, aus welcher Richtung die Daten abgehört werden. Ist JP1 gesetzt, werden die Daten vom PC zur Hardware-Komponente durchgeschleift. Ist hingegen JP2 gesetzt, sind es die Daten von der Hardware-Komponente , die in der Folge als SPS (serielles Peripherie-System)bezeichnet wird. Über D1/D2 und R4 werden die Signale zusammengeführt und an Pin 2 (RXD) von K2 für das Notebook bereitgestellt.

Die Widerstände R1 und R2 halten Spannungsspitzen und negative Signale von IC1 fern. Die gewählten Widerstandswerte führen in Verbindung mit der Eigenkapazität von IC1 dazu, dass die höchste Baudrate 38400 Baud beträgt.

Die Spannungsversorgung wird von der seriellen Schnittstelle des Notebooks über D3(DTR), D4(RTS) und C1,C2 gewonnen.

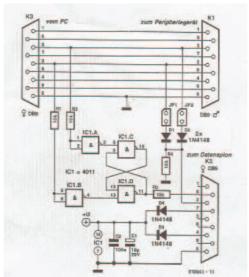


Abb. Schaltplan des Spions

4.3 Software

Das Programm mit der Bezeichnung Bin_Term (Binäres Terminal) wurde unter Borland Delphi 5 erstellt und kompiliert. Das Projekt besteht aus einer Form und zwei Units. Unit Fmain.pas enthält alle sichtbaren Elemente. Unit Seriell.pas enthält die Ansteuerung/Abfrage der seriellen Schnittstelle. Das Programm läuft unter Win9x, WinNT4 und umfasst nur eine Datei(BinTerm.exe). Beim Beenden des Programms werden alle Einstellungen in einer INI-Datei gespeichert.

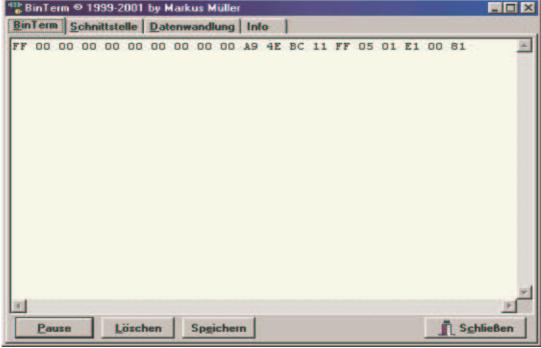


Abb. Einstellungsoberfläche der Bin_Term

Visualisierung einer Wetterstation mit dem C++Builder

4.4 Anwendung

Nach dem Starten von BinTerm.exe erscheint das Bin-Term-Fenster, in dem die

ankommenden Daten angezeigt werden. Die unten im Anzeigefenster zu sehenden

Tasten PAUSE, LÖSCHEN, SPEICHERN und SCHLIESSEN haben folgende

Funktionen:

PAUSE: Anhalten der Datenabfrage

LÖSCHEN: Löscht den Bildschirminhalt(Textboxen)

SPEICHERN; Speichert den angezeigten Inhalt in einer Textdatei

SCHLIESSEN: Beendet das Programm

4.5 Schnittstelleneinstellungen

Nach einem Klick auf "Schnittstelle" erscheint das entsprechende Einstellungs-

fenster. Die Funktion der Knöpfe unten erklärt sich fast von selbst:

Übernehmen: Neu Öffnen der seriellen Schnittstelle mit den gewählten

Einstellungen.

Schließen: Beendet das Programm.

20

Unter Schnittestelle A läst sich die Schnittstelle einstellen, an welche der Schnittstellenspion eingestöpselt ist.

In dem Feld Art läst sich die Richtung der Abhörung festlegen. Wobei mit diesem Datenspion die Abhörung beider Richtungen gleichzeitig nicht möglich ist.

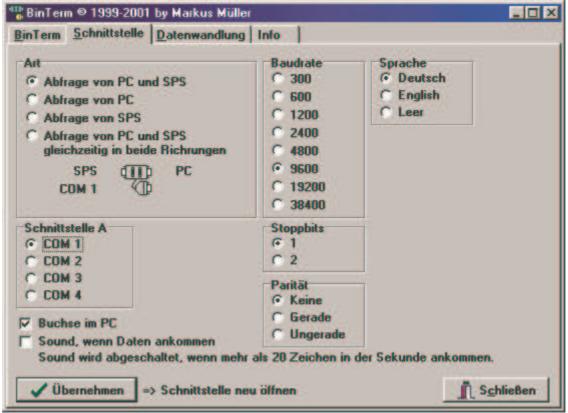


Abb. Einstellungsoberfläche der Schnittstelle

Beide abzufragende Seiten müssen die gleiche Baudrate haben. Die Einstellungen der Stoppbits und der Parität hatten komischerweise keinen Einfluss auf unsere Aufzeichnungen!

Die übrigen Einstellungen müssen exakt stimmen.

4.6 Datenwandlung

Nach einem Klick auf "Datenwandlung" erscheint das entsprechende Einstellungsfenster. Für die Wandlung der Zeichen für die Anzeige kann man aus drei Arten auswählen: ASCII, Dezimal und Hexadezimal. Für jedes Zeichen kann ein Zeilenumbruch eingestellt werden. Die Zeichen, die einen Zeichenumbruch auslösen, werden in einer Liste angezeigt. Doppelklick auf einen Eintrag löscht diesen aus der Liste. Bei einem Zeilenwechselzeichen vom PC wird "Text bei Zeilenende PC" und bei Zeilenwechselzeichen vom SPS der "Text bei Zeilenende SPS" angezeigt.

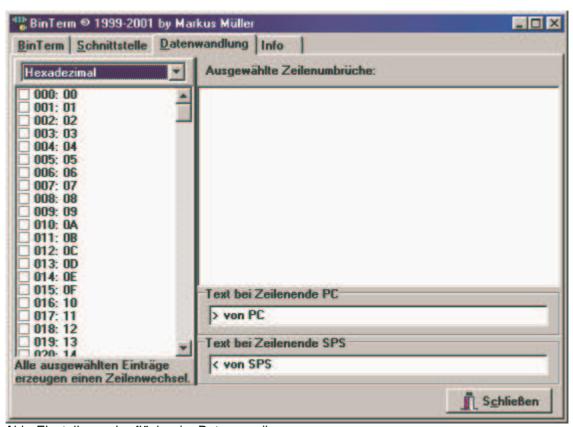


Abb. Einstellungsoberfläche der Datenwandlung

Die letzte Karteikarte des Bin_Term-Fensters erscheint nach dem Klick auf "Info" und verrät den Namen, die E-Mail-Adresse sowie die Homepage des Autors.

5. Grundlagen der Entwicklungsumgebung

Die Grundlagen der Entwicklungsumgebung wurden zum grossteil von einer Internetseite der Uni-Regensburg übernommen.

5.1 Hauptfenster der IDE

Das eigentliche Hauptfenster der IDE besteht aus dem Menü, der Symbolleiste und der Komponentenleiste.



Abb. Das IDE-Fenster des C++- Builder

5.2 Symbolleiste

Die Symbolleiste befindet sich auf der linken Seite unter der Menüleiste. Jeder Symbolleistenschalter ist einem Befehl im Hauptmenü zugeordnet. Mit Hilfe dieser Leiste können häufig verwendete Befehle durch einen einfachen Mausklick gestartet werden.



Abb. Symbolleiste

5.3 Komponentenleiste

Die Komponentenleiste besteht aus mehreren Registerblättern (auch Tabs genannt). Jedes Registerblatt beschreibt eine Gruppe von Steuerelementen.

Die Komponentenleiste hat keine direkte Verbindung zum Hauptmenü. In ihr sind alle Steuerelemente dargestellt, die für das aktuelle Programm zur Verfügung stehen. Steuerelemente sind bspw. Schaltflächen, Eingabefelder usw. Diese Elemente können mit Hilfe von Drag-and-Drop auf ein Formular gelegt werden.

Eine bestimmte Komponente, deren Name bekannt ist, aber nicht das Registerblatt, auf welchem sie sich befindet, kann mit Hilfe des Fensters KOMPONENTEN in der Komponentenleiste gefunden werden. Geöffnet wird das Fenster über das Menü ANSICHT | KOMPONENTENLISTE



Abb. Das Komponentenfenster zum Verwalten der Komponentenleiste

Über "Namen suchen" kann der Name der gesuchten Komponente eingegeben werden. Wurde sie gefunden, dann kann sie über Doppelklick oder mit einem Klick auf die Schaltfläche HINZUFÜGEN zum aktuellen Formular hinzugefügt werden. Eine Komponente ist ein selbstständiger binärer Software-Baustein, der eine vordefinierte Aufgabe erfüllt, z.B. Textfelder (Labels), Editierfelder oder Listenfelder.

5.4 Formulardesigner

Ein Formular ist das Fenster der zu entwickelnden Anwendung. Hier werden die benötigten Steuerelemente abgelegt und verwaltet.

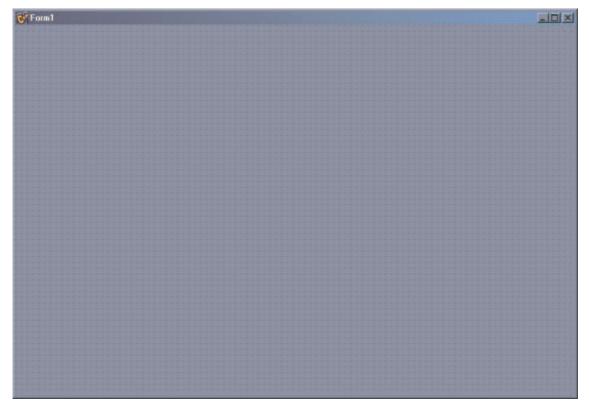


Abb. Der Formulardesigner

Von einem Hauptformular aus, können beliebig viele andere Formulare aufgerufen und verwaltet werden. Der Formular-Designer wird bei der Erstellung von Formularen zum Plazieren, Verschieben oder Verändern von Komponenten eingesetzt. Hinter dem Formular Designer steht der sog. Quelltexteditor zur Eingabe des Quelltexts. Das Formular Gitter: Der Formular-Designer verfügt über ein eingebautes Gitter. Dieses Gitter wird im C++Builder standardmäßig mit anfänglichen Abständen von acht Pixel horizontal und acht Pixel vertikal angezeigt. Ist im Formular-Designer das Formulargitter eingeschaltet, wird an allen Schnittpunkten der Gitterlinien ein Punkt angezeigt. Komponenten richten sich bei ihrer Platzierung im Formular automatisch am nächsten Gitterpunkt aus. Ausrichten bedeutet: Die linke obere Ecke der Komponente springt automatisch zum nächstliegenden Gitterpunkt. Die Gittergröße kann über die Seite Präferenzen im Dialogfeld Umgebungsoptionen eingestellt werden.

5.5 Objektinspektor-Fenster

Das Objektinspektor-Fenster enthält alle Eigenschaften eines Steuerelements und alle Ereignisse, die auf dieses Steuerelement wirken oder von diesem ausgelöst werden können.

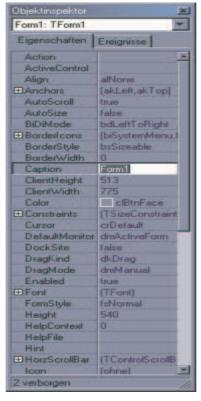


Abb. Der Objektinspektor

Der Objektinspektor besitzt zwei Tabs, mit denen man zwischen Eigenschaften und Ereignissen eines Steuerelements unterscheiden kann.

Eigenschaften: Sie definieren Struktur und Aussehen eines Steuerelements.

Ereignisse: Mit den Ereignissen wird das gesamte zu entwickelnde Programm gesteuert. Ein Ereignis, das auf ein Steuerelement wirkt, ist bspw. das Ereignis des Anklickens mit der Maus: Tritt das Ereignis OnClick ein, so wird der Programmcode, der diesem Ereignis hinterlegt ist, ausgeführt.

5.6 Fenster zur Projektverwaltung

Das Fenster zur Projektverwaltung zeigt sämtliche im aktuellen Projekt benutzte Formulare und Programmdateien. Diese Programmdateien, die keinem Formular zugeordnet sind, heißen Units. Weiterhin dient dieses Fenster für das Hinzufügen, Kopieren, Löschen oder Speichern einer Datei zum aktuellen Projekt. Das Projektverwaltungsfenster lässt sich über das Menü ANSICHT |PROJEKTVERWALTUNG öffnen.

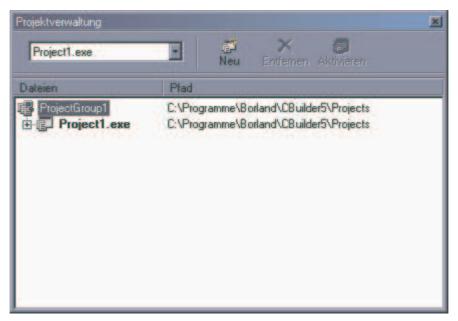


Abb. Das Fenster zur Projektverwaltung

Erzeugen von neuen Projekten: Mit dieser Schaltfläche können neue Projekte erzeugt werden, die dann ebenfalls im Projektverwaltungsfenster angezeigt werden. Nach Betätigen der Schaltfläche erscheint ein Auswahlfenster, in dem die Art der neuen Applikation festgelegt wird.

Entfernen eines Projekts aus der Projektverwaltung: Mit dieser Schaltfläche wird ein Projekt aus dem Projektverwaltungsfenster entfernt.

Aktivieren eines Projekts: In der Entwicklungsumgebung des C++Builders ist immer nur ein Projekt aktiv. Falls in der Projektverwaltung mehrere Projekte aufgeführt sind, kann mit dieser Schaltfläche ein inaktives Projekt aktiviert werden.

5.7 Quelltextfenster

Der Programmcode, der vom C++Builder generiert oder vom Entwickler eingegeben wird, ist im Quelltextfenster sichtbar.

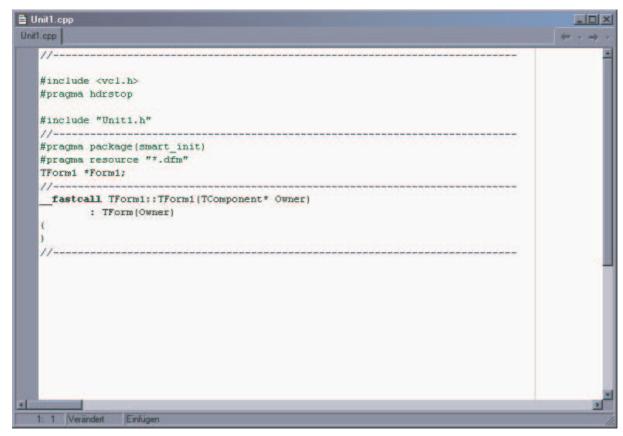


Abb. Das Quelltextfenster

Zum Öffnen des Quelltextfensters gibt es zwei Möglichkeiten:

- Doppelklicken auf eine Datei im Projektbrowser
- Das Menü ANSICHT | PROJEKT-QUELLTEXT aktivieren.

5.8 Festlegen der Programmeigenschaften

Zum Festlegen der Programmeigenschaften gibt es unter dem C++Builder das Dialogfenster PROGRAMMEIGENSCHAFTEN. Der Dialog kann über das Menü PROJEKT | OPTIONEN aktiviert werden.

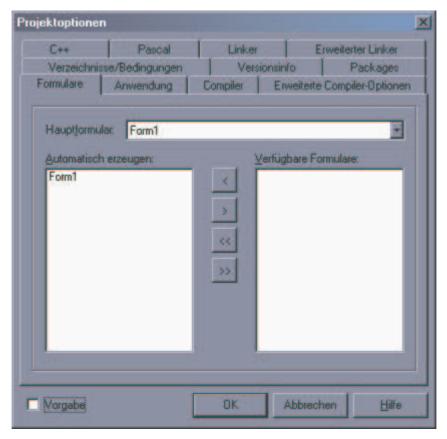


Abb. Registerblatt Formulare der Projektoptionen

Auf diesem Formular befinden sich einzelne Registerblätter, die einzelne Themenbereiche für die Projekteinstellungen zusammenfassen. Das Kontrollfeld "Vorgabe" ist Registerblatt übergreifend und definiert, dass die aktuellen Projekteinstellungen für alle Projekte gültig sind, die neu erzeugt werden müssen.

Formulare: Hier werden alle Formulare des aktuellen Projekts aufgeführt. In dem Feld "Hauptformular" steht der Name des Formulars, welcher der Benutzer beim Start der Anwendung sieht.

Anwendung: Hier werden alle Einstellungen vorgenommen, mit denen sich die Applikation identifiziert. Der Titel der Anwendung ist der Name, der in der Startleiste rechts vom Programmsymbol erscheint. Compiler-Optionen: Hier werden die allgemeinen Einstellungen bzgl. des Übersetzungsvorgangs festgelegt. Erweiterte Compiler-Optionen: Hier werden die speziellen Einstellungen für die Programmübersetzung festgelegt. C++: Diese Registerkarte enthält alle C++-sprachspezifischen Einstellungen des Projekts, die für den Compiler von Bedeutung sind. Pascal: Linker: Über ihn werden die einzelnen kompilierten Dateien zu einer einzigen Datei zusammengeführt. Verzeichnisse/Bedingungen: Versions-Info: Packages: Tasm: Dieses Registerblatt betrifft den (mitgelieferten) Turbo-Assembler von Borland. CodeGuard:

5.9 Installation des Borland C++Builder

Nach der Installation und Einbinden der Abakus- und CPort-Komponenten, müssen einige Einstellungen vorgenommen werden. Bei der Abspeicherung eines neuen Projektes erscheint eine Fehlermeldung, dass das Programm eine Datei mit der Endung (.h) nicht findet. Diese muss dann im Fehleranzeigefeld unter Durchsuchen mit einer ähnlichen Datei mit der Endung (.hpp) ersetzt werden.

Auch bei dem ersten Compilieren tauchen mehrere Fehlermeldungen auf, welche nach einer beigelegten Beschreibung behoben werden müssen. Diese Beschreibung ist in Englisch geschrieben und ist im Anhang beigefügt.

6. Vorarbeiten zur Programmierung

Zum einarbeiten in die Entwicklungsumgebung wurde zum grossteil ein Buch von Richard Kaiser mit der ISBN: 3-540-62994-7 und dem Titel "C++ mit dem Borland C++Builder. Die Unterlagen der Schule, sowie das Internet wurden auch mit einbezogen.

6.1 Allgemein

Als ich die Aufgabenstellung meiner Techniker-Arbeit erhalten hatte, legte ich mir als erstes ein Pflichtenheft an. Darin wurde die Aufgabenstellung dokumentiert, die Unterteilung in einzelne Arbeitsabschnitte vorgenommen und ein Zeitplan in Form eines Balkendiagramms erstellt.(siehe Pflichtenheft)

Dabei kann man sagen, dass die Zeiteinhaltung nicht wie geplant eingehalten wurde, weil immer wieder Probleme auftauchten, welche die Einteilung zunichte machten.

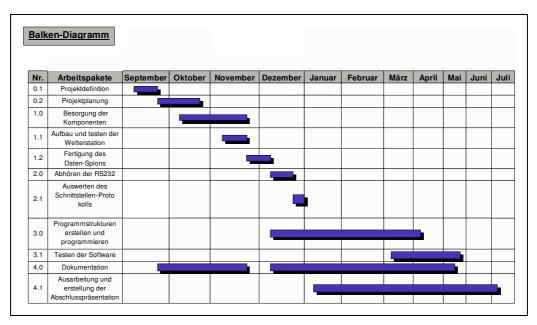


Abb. Zeiteinteilung als Balkendiagram.

6.2 Recherchen und Einarbeitung in die Entwicklungsumgebung.

Hierbei war das Internet und ein Buch "C++mit dem Borland C++Builder" meine größte Hilfe. Es galt nun viel zu lesen.

6.3 Aufbau der Wetterstation

Zunächst mal musste eine Holzplatte mit den Maßen (470x470x35), für den Versuchsaufbau zugeschnitten und behandelt werden. Zusätzlich wurden noch zwei Aluminiumrohre mit einem Durchmesser von 29,5mm und einer Länge von 500mm auf der Platte befestigt, um später die externen Messsensoren für die Windrichtung und Windgeschwindigkeit montieren zu können.

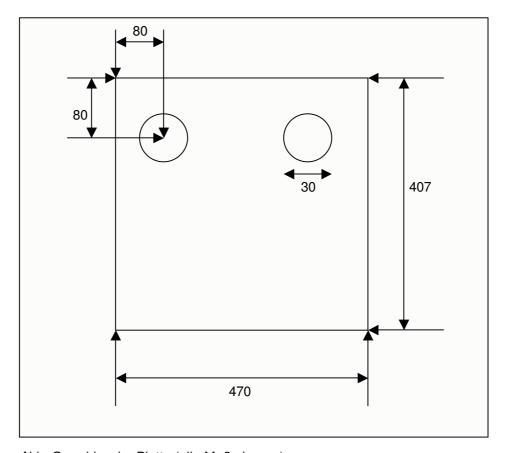


Abb. Grundriss der Platte (alle Maße in mm)

Nun stand die Montage der Wetterstation an. Zunächst wurde die Löcher für die Leitungseinführung, sowie zwei Löcher am Boden des Gehäuses für die optimale Messwertgewinnung von Luftdruck und Luftfeuchtigkeit gebohrt. Nachdem diese Arbeiten erledigt waren, konnte das Gehäuse auf der Montageplattebefestigt werden. Da die mitgelieferten Leitungen für die Betriebsspannung und Datenübertragung zu kurz waren, mussten neue Leitungen besorgt werden und die vorhandenen Stecker angelötet werden. Die Leitungen für Betriebsspannung, Datenübertragung und der externen Sensoren wurden nun in das Gehäuse eingeführt.

Nun folgte das Einsetzen der Sensor-Module sowie das Befestigen des Lichtstärken-Sensor an dem Deckel des Gehäuses.

6.4 Abhören der Schnittstelle

Dies erfolgte mit einem Datenspion, welcher aber zuerst noch gebaut werden musste. Da ich die dazugehörige Platine bei der Firma Elektor käuflich erwerben konnte und der Bestückungsplan schon vorhanden war, wurde dieser Teil derArbeit zügig erledigt. Von der Wetterstation wurde nun eine Datenverbindung zum Datenspion an den Anschluss K1 gelegt. Am Datenspionanschluss K3 wurde der PC Angeschlossen, auf dem die Software Telemetrie der Wetterstation installiert war. Zu guter letzt musste noch eine Verbindung vom Datenspionanschluss K2 an einen zweiten PC oder ein Laptop geschaffen werden, wo die Abhörsoftware Bin_Term für den Datenspion installiert war.

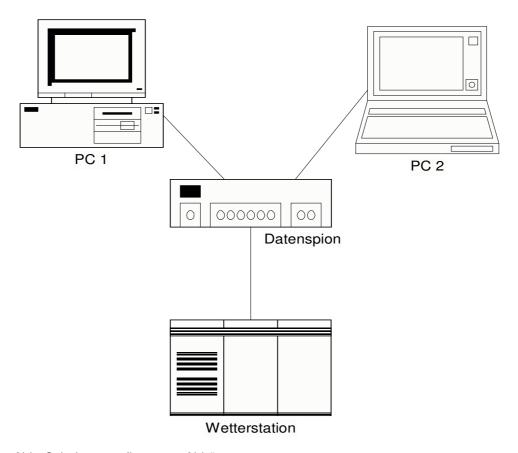


Abb. Schaltungsaufbau zum Abhören.

6.5 Die Spionage beginnt

Dadurch dass man die verwendete Bautrate, die Anzahl der Stopbits und die Parität des Datenaustausches zwischen Wetterstation und PC an der Software Telemetrie ablesen konnte, war das Einstellen der Abhörsoftware kein Problem.

Als erstes galt es herauszufinden, ob ein Datenaustausch vom PC zur Wetterstation stattfindet. Da dies nicht der Fall war wurde klar, dass in dem zu erstellenden Programm kein Software-Handshake vorhanden sein darf.

Durch das Umstecken des Jumpers, welcher sich auf dem Datenspion befindet, wurde die Abhörrichtung von der Wetterstation zum PC umgeschaltet. Bei dieser Datenaustauschrichtung wurde die Erkenntnis gewonnen, dass die Wetterstation kontinuierlich einen Datenrahmen von 14 Bytes sendet.

```
Schritte (3-5) 0D < von SPS
00 < von SPS
3F 98 1C 2D C3 5F FF 00 < von SPS
25 01 < von SPS
8F 00 < von SPS
0D < von SPS
0D < von SPS
3E 98 1C 2E C2 5F FF 00 < von SPS
25 01 < von SPS
90 00 < von SPS
```

Abb. Auszug der Datenaufzeichnung.

6.6 Bytezuweisung

Durch die gewonnene Erkenntnis, galt es jetzt die Zuständigkeit der einzelnen Bytes für die einzelnen Messwerte Offenzulegen. Durch gezieltes Verändern der einzelnen Sensoren an der Wetterstation wurde dies anschaulich. Als Hilfsmittel zur Veränderung der Werte an den Sensoren kamen ein Föhn, eine dimmbare Schreibtischlampe sowie Eisspray zum Einsatz.

Daraus ergab sich folgende Bytezuweisung:

Byte	Messwert	
0	Windrichtung	
1		
2		
3	Gammastrahlung	
4	Lichtstärke	
5	Luftfeuchtigkeit	
6	Temperatur	
7	Luftdruck	
8		
9	Windgeschwindigkeit	
10		
11		
12		
13		

Abb. Tabelle für Bytezuweisung

6.7 Tabellarische Festlegung der Messwerte.

Weil der einzige linear verlaufende Messwert die Windgeschwindigkeit war, musste hier nur der übermittelte Wert mit einem Faktor multipliziert werden. Dieser Faktor war in diesem Fall 0,0179.

Für die Messwerte von Temperatur, Luftdruck, Lichtstärke, Luftfeuchtigkeit, Gammastrahlung sowie Windrichtung musste ein tabellarischer Verlauf angelegt werden.

Dies gelang dadurch, das jedes einzelne Byte von 0 bis 255, immer um eins erhöht wurde. Der angezeigte Wert auf dem Bildschirm und der eingestellte Bytewert wurden schriftlich festgehalten und tabellarisch angelegt.

Diese Arbeit war sehr zeitaufwendig und ich musste sehr viel Geduld beweisen.

Bytewert	Windrichtung	Lichtstärke	Strahlung	Luftdruck	Luftfeuchtigkeit	Temperatur
0	0,0	840	0,00	0	100	-27,5
1	1,4	841	0,01	0	100	-27,0
2	2,8	842	0,02	10	100	-26,0
3	4,2	843	0,03	30	100	-25,5
4	5,6	844	0,04	50	100	-24,5
5	7,1	845	0,05	80	100	-24,0
6	8,5	846	0,06	100	100	-23,5
7	9,9	847	0,07	100	100	-22,5
8	11,3	848	0,08	100	100	-22,0
9	12,7	849	0,09	100	100	-21,5
10	14,1	850	0,10	110	100	-20,5
11	15,5	851	0,11	110	100	-20,0
12	16,9	852	0,12	110	100	-19,5
13	18,4	853	0,13	120	100	-18,5
14	19,8	854	0,14	130	100	-18,0
15	21,2	855	0,15	140	100	-17,5
16	22,6	856	0,16	150	100	-17,0
Al-l- A		Haladia				-

Abb. Auszug von Messwerttabelle.

Nachdem jetzt sämtliche Auswertungen über Byteanzahl, Bytezuweisung und die Messungen zur tabellarischen Darstellung der Messwerte erledigt waren, war die Arbeit mit dem Datenspion beendet und das Programmieren konnte beginnen.

7. Das Programm

7.1 Allgemein

Für das Erstellen der graphischen Darstellung wurde die Entwicklungsumgebung des Borland C++Builder5 verwendet. Als erster Schritt musste die Komponente Abakus in das Programm eingebunden werden, welche die passenden analogen und digitalen Anzeigen beinhaltete. Die Einbindung der ComPort-Komponente, welche notwendig war, um die serielle Schnittstelle ansprechen zu können, war der zweite Schritt. Nun folgte die Auswahl und Anordnung der graphischen Anzeigen auf der Benutzeroberfläche, sowie die Beschriftung der Anzeigen.



Abb. Anordnung und Beschriftung der Anzeigen.

Nach der Anordnung der Anzeigen, mussten für jede Anzeige unter Eigenschaften die richtigen Einstellungen vorgenommen werden, um das Layout gleichmäßig darzustellen. Diese Einstellungen nahmen einige Zeit in Anspruch, da eine riesige Auswahl unter der Anwendung Eigenschaft vorhanden ist.

Visualisierung einer	Wetterstation	mit
dem C++Builder		

7.2 Messwertanzeige-Beschreibung

7.2.1 Luftfeuchtigkeit

Die Luftfeuchtigkeit wird mit einem Zeigerinstrument graphisch dargestellt. Es wird in der Einheit % gemessen und hat eine Skala von 0 bis 100, welche in 5er-Schritte eingeteilt ist.



Abb. Feuchtigkeitsanzeige

Da die Luftfeuchtigkeit nicht linear steigt, musste ein Array mit 256 Stellen angelegt werden. An diesen Stellen wurden die Messwerte hineingeschrieben, welche sich aus der Aufzeichnung für die tabellarische Darstellung ergaben.

Ein Auszug aus dem Array für die Luftfeuchtigkeit:

7.2.2 Temperatur

Bei der Temperaturdarstellung habe ich mich für einen Thermometer entschieden, bei dem die Quecksilberflüssigkeit als eine LED-Anzeige dargestellt wird. Die Temperatur hat die Einheit Celsius. Die Skala geht von –40° bis +60° und ist in 5er-Schritte unterteilt.



Abb. Temperaturanzeige.

Die Temperatur besitzt ebenfalls eine nicht lineare Funktion. Da hier kein Faktor für die Wertgewinnung angewendet werden konnte, musste ein Array mit 256 Zeichen angelegt werden. In dieses Array wurden die ermittelten Wert für die Temperaturanzeige hineingeschrieben.

 $tempwert[] = \{-27.5, -27.0, -26.0, -25.5, -24.5, -24.0, -23.5, -22.5, -22.0, -21.5, -24.0, -23.5, -24.0,$

7.2.3 Windrichtungsanzeige

Die Windrichtungsanzeige ist hier in Form eines Kompasses dargestellt. Der Kompass hat drei Darstellungsarten des momentanen Wertes. Die erste Form ist eine Richtungsanzeige, welche durch einen gelben Zeiger, den Wert in schriftlicher Form angibt (z.B. NW, S, SW).

Der digitalen Darstellung der Windrichtung entsprechen die anderen zwei Formen der graphischen Ansicht. Hier wird einmal die Gradzahl der Richtung und zum anderen die negative Gradzahl angegeben.



Abb. Windrichtungsanzeige.

Die Windrichtung hat die Einheit Grad und geht von 0 bis 360 Grad. Diese Werte sind in einem Array festgelegt.

7.2.4 Windgeschwindigkeit

Ein digitales Anzeigefeld stellt den Windgeschwindigkeitswert dar. Er wird in der Einheit Meter pro Sekunde dargestellt, weil diese Einheit für unsere Region am realistischsten ist. Für diesen Wert konnte man einen Faktor nehmen, weil er eine lineare Funktion nachweist. Der Faktor beträgt übrigens 0,0179.



Abb. Windgeschwindigkeitsanzeige.

Für die Windstärke gibt es noch eine Bezeichnung, die von der Stärke des Windes abhängt.

Windstärke	Bezeichnung	M/sec
0	Windstill	0,1 bis 0,3
1	Leichter Zug	0.4 bis 1,5
2	Sanfte Brise	2,0 bis 3,5
3	Leichte Brise	4,0 bis 5,5
4	Mäßige Brise	6,0 bis 8,0
5	Mäßiger Wind	8,5 bis 11,0
6	Starker Wind	11,5 bis 14,0
7	Sehr starker Wind	14,5 bis 17,0
8	Stürmischer Wind	17,5 bis 20,5
9	Sturm	21,5 bis 24,5
10	Heftiger Sturm	25,5 bis 28,5
11	Orkanartiger Sturm	29,0 bis 32,5
12	Orkan	33,0 bis 34,0

Abb. Tabelle für Windbezeichnung.

7.2.5 Lichtstärke

Sie wird in einer Digitalanzeige visualisiert und mit der Einheit LUX gemessen. Der Wert für die Lichtstärke ist in einem Array festgelegt, beginnt bei 0 LUX an und endet bei 85000 LUX auf.



Abb. Lichtstärkenanzeige.

Hier der Auszug aus den Array für die Lichtstärke:

7.2.6 Gammastrahlung

Die Gammastrahlung wird in der Einheit uS/h gemessen und als ein digitaler Wert dargestellt. Die Werte der Gammastrahlung sind ebenfalls in einem Array festgelegt, welches einen Wert von 0 bis 2.5 hat. Meine Probemessungen haben ergeben, dass eine Strahlung von 0.03 im Alttag normal ist.



Abb. Gammastrahlungsanzeige

Hier der Auszug aus den Array für die Gammastrahlung:

gamawert[]={ 0.00,0.01,0.02,0.03,0.04,0.05,0.06,0.07,0.08,0.09,0.10,0.11, 0.12,0.13,0.14,0.15,0.16,0.17,0.18,0.19,0.20,0.21,0.22,0.23, 0.24,0.25,0.25,0.26,0.27,0.28,0.29,0.30,0.31,0.32,0.33,0.34, 0.35,0.36,0.37,0.38,0.39,0.40,0.41,0.42,0.43,0.44,0.45,0.46, 0.47,0.48,0.49,0.50,0.51,0.52,0.53,0.54,0.55,0.56,0.57,0.58, 0.59,0.60,0.61,0.62,0.63,0.64,0.65,0.66,0.67,0.68,0.69,0.70,

7.2.7 Luftdruck

Bei diesem Messwert habe ich mich ebenfalls für eine digitale Anzeige entschieden, weil es wenig Sinn machen würde, diesen Wert mit einem Zeigerinstrument darzustellen. Die gewohnte Einheit für den Luftdruck ist hPa (Hekto-Pascal). Das dafür angelegte Array hat den Wert von 0 bis 1095 hPa.



Abb. Luftdruckanzeige.

Hier der Auszug aus den Array für den Luftdruck:

```
luftwert[]={840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851,852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 962, 972, 982, 929, 930,
```

7.2.8 Kennliniendarstellung

Dieses Anzeigefeld ist in der Abakus-Komponente enthalten und eignet sich hervorragend für die Kennliniendarstellung von Messwerten. Es ist in zwei Achsen aufgeteilt. Die horizontale Achse gibt die Einheit des Messwertes an und auf der vertikalen Achse befindet sich die Zeit.

In meinem Fall wurde dieses Instrument für die Aufzeichnung von der Temperatur und der Lichtstärke verwendet.



Abb. Anzeige für Kennlinienverlauf.

Den anzuzeigenden Wert erhält dieses Messgerät immer von den zuständigen digitalen Anzeigen. Ch1 steht für Chanel1 und ist dem Messwert von der Temperatur zugewiesen. Es gibt auch einen Chanel2 (Ch2), welcher dem Messwert von der Lichtstärke zugeteilt ist.

Zwischen diesen beiden Chanels läst sich durch einen Schalter beliebig hin und her schalten. Von dieser Schalterstellung ist auch der Label 7 (Bild oben) abhängig. Je nach Schalterstellung steht im Label 7 der Name des derzeit angezeigten Messwertes



Abb. Schalter für die Anzeigeauswahl.

Als nächstes habe ich noch eine Veränderungsmöglichkeit des Interwals eingebunden. d.h man kann die Zeit, die zwischen der ersten Signalaufzeichnung und der darauffolgende Aufzeichnung vergeht, beeinflussen.

Dies geschieht mit vier Buttons, auf denen die Zeiteinstellung lesbar ist.

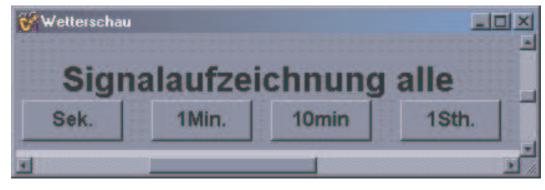


Abb. Zeiteinstellung der Signalaufzeichnung.

7.2.9 Ein- Ausschalter

Dies ist die einzige Komponente, welche auf der compilierten Oberfläche betätigt werden kann.



Abb. Schalter

In dieser Funktion werden einige Eigenschaften der eingebundenen Komponenten beeinflusst. Sobald der Schalter auf true (ein) gesetzt wird, wird die Komponente ComPort auf open gesetzt.

ComPort1->Open();

Die folgenden zwei Zeilen des Programms gehören auch zu der Komponente ComPort. Sie waren der Schlüssel zu meinem größten Problem während der gesamten Techniker-Arbeit. In den Abschnitt (8.10 ComPort) gehe ich auf dieses Problem näher ein.

ComPort 1 -> Flow Control -> Control DTR = dtr Enable;

ComPort1->FlowControl->ControlRTS=rtsEnable;

Die Kennlinienanzeige musste ebenfalls über den Schalter beeinflusst werden, da sonst beim Öffnen der compilierten Anwendung die Zeit, welche sich auf der vertikalen Achse befindet, automatisch gestartet wurde.

AbTrend1->Enabled=true;
AbTrend1->StartRecording = true;

Ebenfalls wird beim Einschalten ein Set-Up Dialog für die serielle Schnittstelle aufgerufen, in dem sich Einstellungen für die Schnittstelle vornehmen lassen.

ComPort1->ShowSetupDialog();

Wenn der Schalter nun wieder auf false (aus) gesetzt wird, hat das zur Folge, dass die Komponente ComPort geschlossen wird und sämtliche Anzeigen auf der Benutzeroberfläche auf den Wert 0 zurückgesetzt werden.

7.2.10 ComPort

Aus der ComPort Komponente wurde die Anwendung TcomPort verwendet, welche mit dem Event "ComPort1RxChar" zum Einsatz kam. In diesem Abschnitt des Programms werden die ganzen Anwendungen der Benutzeroberfläche bearbeitet.



Abb. TcomPort-Symbol

Hier sind die ganzen Arrays mit den tabellarisch ermittelten Werten für die Anzeigefeldern aufgeführt. Des weiteren ist hier ein Array messw. mit 14 Stellen initialisiert, in welches die ankommenden Werte von der Wetterstation hineingeschrieben werden. Ich werde nun anhand des Temperaturanzeigewertes den Ablauf des Programms erklären, weil dieser Ablauf bei den anderen Wertzuweisungen identisch ist.

Als erstes muss der Schalter auf true gesetzt werden um den ComPort zu öffnen. Bevor dies geschieht, erscheint zunächst ein Anzeigefeld, in dem die Eigenschaften für die Schnittstelle eingestellt werden müssen.

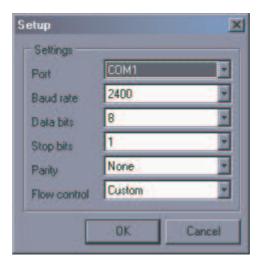


Abb. Setup für die Schnittstelle

Nachdem nun der Port, die Baudrate, Databits, Stopbits, Parity und der Flow control richtig eingestellt wurden, müsste eigentlich der Event "ComPort1RxChar aufgerufen werden.

Da kam dann mein großes Problem zum Vorschein, welches mich einige Zeit beschäftigte. Als erstes probierte ich es mit einem zusätzlichen einfachen Programm, welches beim Aufrufen des Events einen Text in einem Textfeld ausgeben sollte.

Weil das auch nicht funktionierte, stellte ich Recherchen im Internet an.

Dabei fand ich eine Seite http://www2.arnes.si/~sopecrni/; dessen Besitzer Dejan Crnila; Dolenja vas 111; 3312 Prebold; Slovenia ist. Dejan Crnila ist ebenfals der Anbieter der Freeware Komponente ComPort. Als ich mich dann im Forum einlockte, wurde mir schnell klar, dass ich nicht der einzige mit diesem Problem bin. Leider konnte ich auch dort keine Hilfe für meine Problemlösung finden.

Nachdem ich so ziemlich alle Möglichkeiten, welche mir bekannt waren, zur Lösung des Problems ausprobiert hatte, wusste ich mir keinen Rat mehr. Als dann mein betreuender Lehrer, "Herr Schnaiter", mir den Tipp mit dem Datenleitungstester gab, kam ich dem Problem auf die Schliche.

Der Datenleitungstester wurde zwischen Wetterstation und PC eingebaut und auf dem PC wurde die Software Telemetrie gestartet. So wurde gleich sichtbar, dass die Leitungen DTR (DATA-TERMINAL-READY) und RTS (READY TO SEND) gesetzt sein müssen. Nun startete ich mein eigenes Programm und habe festgestellt, dass die DTR-Leitung nicht gesetzt war. Also war klar, dass hier die Lösung des Problems lag.

Der nächste Schritt war die ComPort Hilfe zu erforschen. Dort wurde ich auch fündig mit einem Verweis auf die Eigenschaft Flow control, unter der ich auch die beiden Eigenschaften Control DTR und Control RTS fand. Diese beiden Leitungen mussten nun auf ControlDTR=dtrEnable und ControlRTS=rtsEnable gesetzt werden. (siehe Anhang Quelltext)

Wobei noch zu beachten ist, dass beim Setup der Schnittstelle Flow control immer auf Custom steht. Bei den anderen 3 Einstellungsmöglichkeiten wird immer eine Leitung wieder rückgesetzt.



Abb. Setup

Da das Problem jetzt endlich behoben war, wurde der Event auch aufgerufen und Ich kann mit meiner Erklärung fortfahren.

Das Programm schreibt als erstes die an der Schnittstelle anliegenden Werte in das Array messw, welches am Anfang des Events deklariert ist.

Dies wird ausgelöst durch den Befehl

ComPort1->Read(messw,Count)

Dieses Array wird dann zerteilt und die einzelnen Bytes den extra dafür angelegten Variablen zugewiesen. In diesem Fall ist es die Variable temp, welcher das sechste Byte zugewiesen wird.

Durch eine tabellarische Festlegung der Messwerte, konnte ein Array mit 256 Temperaturwerten angefertigt werden. Der eingelesene Wert in der Variable temp, zeigt nun auf den jeweiligen reellen Temperaturwert. Der Befehl für diesen Ablauf lautet:

tempwert[temp]

Über die Eigenschaft Value der Temperaturanzeige wird der so resultierende Wert direkt zugewiesen. Diese Zuweisung geschieht durch:

Temp -> Value = tempwert[temp]

So ist der Programmverlauf bei jeder Wertzuweisung, außer der Windgeschwindigkeit. Weil die Windgeschwindigkeit eine lineare Funktion besitzt, brauchte ich kein zusätzliches Array anzulegen. Der Wert, welcher an der Schnittstelle anliegt, konnte der Variablen wind zugewiesen werden und mit einem Faktor multipliziert werden. Danach konnte die Variable der Anzeige der Windgeschwindigkeit direkt über die Eigenschaft Value zugewiesen werden.

float wind=messw[9]*0.0179

8.Schlusswort

Die Techniker-Arbeit war für mich eine Herausforderung, da ich mich mit der C-Programmierung sowie mit dem Datenverkehr auf der seriellen Schnittstelle noch nicht so intensiv befasst habe.

Dadurch dass immer wieder neue Probleme auftauchten, wurde ich ständig vor neue Herausforderungen gestellt.

Bei dem Arbeiten mit dem C++Builder von Borland merkte ich erst, wie vielfältig diese Entwicklungsumgebung ist und was man alles erstellen kann. Leider reichte es mir zeitlich nicht mehr zur Netzwerkprogrammierung und der Langzeitabspeicherung. Meine Intentionen gehen dahin, diese beiden Punkte für mich privat noch fertig zu stellen.

Anhang

Tabellarische Darstellung Messwerte

Programm Quelltext

Abakus Einbindungshilfe

ComPort Einbindungshilfe auf Englisch

Byteanordnung

Foto der Wetterstation

Pflichten-Heft

CD-ROM