

31.1 Zusammenfassung

Da es das ideale, in einer Hand zu haltende, frei programmierbare und mit griffiger Tastatur ausgestattete Hand held Terminal offenbar noch nicht gibt, bietet der EPSON-HX 20 Rechner vorerst einen ernst zu nehmenden Kompromiß. Das integrierte relativ komfortable MBASIC läßt kaum Wünsche offen. Besonders nützlich für die Datenerfassung ist die Möglichkeit, Datensätze definierbarer fester Länge auf RAM-Files abzulegen und darauf im Direct Access zugreifen zu können. Auf der optional erhältlichen integrierten Mikroassette können die Dateien sequentiell abgelegt und damit gesichert werden. Ebenso ist die Ausgabe am integrierten Minidruker möglich. Über die integrierte RS 232 Schnittstelle können die auf der Kassette abgelegten Dateien auf einen PC (ITT-3030) übertragen werden, sofern ein Sendeprogramm am HX 20 und ein darauf abgestimmtes Empfangsprogramm am PC zur Verfügung stehen.

31.2 Zur Problematik der mobilen Datenerfassung

Obwohl es einen weiten Anwendungsbereich für die mobile Datenerfassung gibt - und dies nicht nur in Landwirtschaft und Gartenbau - fehlt bislang das universelle Hand Held Terminal, das unseren Erachtens folgende Bedingungen zu erfüllen hätte:

- o Handlichkeit, d.h. in einer Hand zu halten
- o große, übersichtliche Tasten
- o LCD-Display mindestens 2x16 (besser 4x20) Zeilen
- o freie Programmierbarkeit in einer problemorientierten Sprache (z.B.BASIC)
- o Speicherkapazität (einschl.Programme) ca. 16k
- o RS 232-Schnittstelle
- o Übertragungsverfahren zu PC's und Telefonadaptern (in beiden Richtungen) und zu Druckern
- o wasserdichtes Gehäuse
- o Unabhängigkeit vom Stromnetz für mindestens 12 Stunden
- o akzeptabler Preis

Auf der einen Seite gibt es heute preiswerte, kompakte und leistungsfähige programmierbare Taschenrechner (z.B. Sharp, CASIO, HP), die eigentlich nur noch in der Größe des Display, entscheidend für die Masseneingabe aber leider in der Griffbarkeit der Tastatur Wünsche offen lassen. Andererseits gibt es Hand Held Terminals, welche die den Taschenrechnern fehlenden Attribute erfüllen, jedoch nur über EPROM's in Maschinensprache programmierbar und damit im täglichen Einsatz wenig flexibel sind. Das Erlernen des Programmierens und Erstellens von

EPROM's ist für die meisten Anwender umständlich. Die Hersteller der Terminals jedoch verlangen für die EPROM-Herstellung nach Anwenderwünschen hohe Preise. So wird der Interessentenkreis 'Hand-Held-Terminals' weiter nach dem idealen Gerät Ausschau halten müssen.

Ein im Preis/Leistungsverhältnis und in der Vielfalt der Anwendungsmöglichkeiten interessantes Gerät ist angesichts des Fehlens eines 'Ideal-Terminals' nach wie vor der EPSON-HX 20. Einziger Nachteil bei der mobilen Datenerfassung ist die Größe (Grundriß DIN A4, Höhe 4cm), die das Halten in einer Hand unmöglich macht. Bei der Datenerfassung im Labor ist das kein Problem. Im Freien jedoch benötigt der Benutzer ein Tablettgestell, das die Hände frei hält. Man kann es sich leicht selbst anfertigen. Dieser Kompromiß ist nach unserer Erfahrung vertretbar, wenn die Datenermittlung erfolgen kann, ohne daß die eingebende Person Rechner und Tablet immer wieder ablegen muß. Nimmt man den genannten Kompromiß in Kauf, so hat man mit diesem Rechner ein recht vielseitig einsetzbares Gerät, dessen Anwendungsmöglichkeiten weit über die mobile Datenerfassung hinausgehen. Die Kapazität der Akkus erlaubt einen etwa 2-tägigen netzunabhängigen Betrieb.

31.3 Die Konfiguration

Als Konfiguration hat sich bei uns die Standardausführung mit als Zubehör erhältlichem Kassettenuaufwerk bewährt. Serienmäßig verfügt der Rechner über fest integriertes MBASIC (von Microsoft), frei für Programme und Daten verfügbare 16k, 4x20-Zeichen LCD-Display (virtuell 8x40), integrierten Nadeldrucker (20Zeichen/Zeile) und deutscher Tastatur mit vollständigem ASCII Zeichensatz. RS 232-Schnittstelle, Anschlüsse für Telefonadapter, Fernsehschirm-Interface und Barcode-Lesestift sind serienmäßig vorhanden, ebenso eine Echtzeituhr. Eine als Zubehör erhältliche Expansion Unit zur Verdopplung des Speichers auf 32k erwies sich bei uns als nicht erforderlich, zumal dadurch der Rechner um ca. 10cm verbreitert und damit unhandlicher wird.

31.4 Das Konzept der mobilen Datenerfassung

Abgestimmt auf das jeweilige Erfassungsproblem ist zunächst ein BASIC-Programm zu erstellen, bzw. ein vorhandenes Programm zu modifizieren. Unsere Erfassungsprogramme enthalten das folgende Grundkonzept:

- o Start durch Programmaufruf
- o Anzeige programminternes Menü:
 1. Datenerfassung
 2. Korrektur von Datensätzen
 3. Datensicherung auf Kassette
 4. Ausgabe der Datensätze am integrierten Drucker
 5. STOP
- o Auswahl und Ausführung des gewünschten Programmteils
- o Rückkehr in das Menü

31.4.1 Datenerfassung

Auf dem LCD-Display werden die angeforderten Eingaben und dahinter die eingegebenen Werte im Klartext angezeigt, wobei bis zu 4 Zeilen gleichzeitig im Display erscheinen. Sofortige Korrektur vor dem 'Abschicken' des Datensatzes ist möglich. Noch vor der Abspeicherung erfolgt eine Plausibilitätskontrolle mit gegebenenfalls akustischem Signal und Wiederholung der Abfrage.

Eine nützliche Eigenschaft des EPSON-Konzeptes ist die Möglichkeit, Dateien von bis zu 256 Zeichen je Satz in RAM-Files abzulegen und darauf im Direct Access zugreifen zu können. Die Definition des zu reservierenden RAM-Bereiches erfolgt mit dem Befehl CLEAR <n,m>, wobei n den Bereich im Übertragungspuffer (z.B. 300 Bytes), m den zu reservierenden RAM-Speicherbereich (z.B. 10000 Bytes) definiert. Dieser Bereich und sein Inhalt bleibt solange erhalten, bis er mit einem CLEAR Befehl anders definiert oder mit anderen Daten überschrieben wird. Durch Abschalten oder andere Aktivitäten mit dem Rechner wird er jedoch nicht gelöscht, auch nicht bei Erweiterung von Programmen, für die dann der jeweilige Rest des Gesamtspeichers verfügbar ist. Bei Überlauf erfolgt Fehleranzeige. Der RAM-Filebereich bleibt jedoch geschützt. Nach unserer Erfahrung reichen 4k auch für komplexere Datenerfassungsprogramme, sodaß man im allgemeinen über 10k RAM-File verfügen kann. Außerdem hat man ja die Möglichkeit, außer Datenfiles auch Programme auf der Kassette abzuspeichern und bei Bedarf wieder zu laden.

Während der CLEAR-Befehl unabhängig von anderen Programmabläufen bis auf Widerruf gilt, muß in einem Direct Access Programm der Zugriff auf den RAM-File jeweils am Programmbeginn definiert sein, und zwar durch das Statement DEFFIL <n,m>, wobei n die Direct Accessnummer des ersten Datensatzes, m die Anzahl Bytes je Satz angibt. Da eine Integer-Zahl bis +/-32767 zwei Bytes und eine Real-Variable mit 8 Stellen Genauigkeit vier Bytes benötigt, können z.B. mit DEFFIL 0,20 beginnend mit der Satznummer 0 je Satz 10 Integer oder 5 Real-Variablen oder entsprechende Mischungen von beiden beziehungsweise ein String von 20 ASCII-Zeichen gespeichert werden. Wichtig ist dabei, daß der Speicher entsprechend der Definition mit DEFFIL voll ausgenutzt wird, d.h., daß die Sätze unmittelbar hintereinander gelegt werden. Bei einem mit CLEAR reserviertem RAM-Bereich von 10k sind das immerhin 10000/20, also 500 Sätze. Die Satznummer wird vom System verwaltet und braucht daher nicht mitgespeichert zu werden. Für Korrekturzwecke, Auflistungen usw. hat man einen Direct Access, d.h. unmittelbaren Zugriff zu jeder Satznummer.

Die Verwaltung der RAM-Files kann man dadurch zusätzlich automatisieren und damit menschliche Irrtümer vermeiden, daß man den ersten Satz (der die Nummer 0 hat) für Steuerparameter reserviert. Auf diesem Satz legt man dann einige für den Programmablauf wichtige Parameter ab, die vor dem Abschalten automatisch aktualisiert werden, so daß z.B. die letzte belegte Satznummer, das Ende der zuletzt auf der Kassette abgespeicherte Datei (als Zähler) u.a..

31.4.2 Korrektur der Datensätze

Es ist eine einfache Programmierungsfrage, für Korrekturzwecke den letzten bereits abgespeicherten Satz oder auch rückwärts Satz für Satz für Korrekturzwecke in das Display zu holen und bei Bedarf zu überschreiben. Auch selektives Suchen nach bestimmten Parametern ist möglich.

31.4.3 Datensicherung auf Kassette

Durch die Prozedur des Ladens vom RAM-File in den Programmspeicher und der Ausgabe von dort auf die Kassette ist auf einfache Weise eine zuverlässige Datensicherung möglich. Dabei werden die Daten sequentiell unter einem Dateinamen (bis zu 8 Zeichen, Punkt und 3 Zeichen Extension) abgespeichert. So kann bei-

spielsweise die Arbeit eines halben Tages unter einem Namen abgespeichert werden, der im Klartext eine Versuchsnummer oder z.B. das Datum enthält. Der Dateiname FRI50884.PM könnte z.B. bedeuten, daß Fruchtmessungen (FR) vom 15.08.84 nachmittags (PM) abgespeichert wurden. Danach könnte die Satznummer im RAM-File wieder automatisch zurückgesetzt und damit der File überlesen werden.

31.4.4 Ausgabe der Datensätze am integrierten Drucker

Entsprechend der Übergabe auf die Kassette kann der Inhalt des RAM-Files auch auf dem integrierten Drucker ausgegeben werden. Damit ist eine Datensichtkontrolle möglich und eine zusätzliche Datendokumentation sichergestellt. Darüberhinaus können auch bereits auf der RAM-File-Ebene statistische Auswerteprogramme mit Ergebnisausgabe am Drucker oder am Display eingesetzt werden. In unserer Konzeption ziehen wir es allerdings zumeist vor, die auf der Kassette gespeicherten Daten zuvor auf einen PC zu übertragen.

31.4.5 Die Übergabe von Dateien vom EPSON HX-20 auf einen PC (Itt-3030)

Die integrierte RS 232-Schnittstelle des HX-20 kann in BASIC über das Statement COMO direkt angesprochen werden. Ein Übergabeprogramm im HX-20 fragt nach dem gewünschten Dateinamen, identifiziert die gesuchte Datei auf der Kassette und sendet die unter dem Dateinamen gespeicherten Daten zeichenweise online über ein Verbindungskabel an den PC.

Zuvor muß aber auf dem PC ein Empfängerprogramm gestartet worden sein, das einen sequentiellen File, gegebenenfalls des gleichen Namens wie auf dem HX-20, eröffnet und die übertragenen Daten zeichenweise übernimmt und abspeichert. Damit steht auf dem Datenträger des PC eine CP/M oder MS-DOS kompatible Datei zur weiteren Verarbeitung in beliebiger Programmiersprache zur Verfügung.