

Kleine Tablets umbauen

Kurzbeschreibung

Stand: 1.1 vom 22. 2. 2016



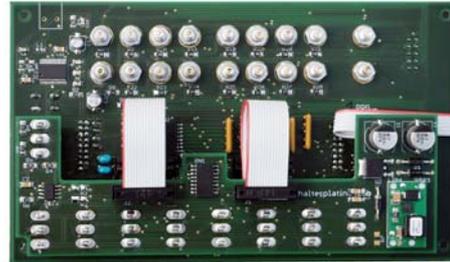
Sinn und Zweck

Gelegentlich braucht man anwendungsspezifische Bedien-, -Anzeige- und Steuereinrichtungen. Dem Stand der Technik entsprechend löst man solche Entwicklungsaufgaben typischerweise so, daß ein Mikrocontroller mit Bedien- und Anzeigeeinheiten zusammenschaltet wird. Heutzutage kosten aber kleine Tabletcomputer oftmals viel weniger als die Bauelemente und Funktionseinheiten, die man benötigt, um eine anwendungsspezifische Lösung aufzubauen. Deshalb liegt es nahe, solche Computer sinngemäß auszunutzen.

Neumodische Kippschaltergeräte. Allein das Material kostet weit mehr als ein kleiner Tabletcomputer.



Sio sieht's innen aus. Der Aufwand ist offensichtlich.



Grundsätzliche Beschränkungen

Es geht um kleine, kostengünstige Tablet-Computer. Formfaktor beispielsweise 7" bis 10". Sie sollen als Bedien-, Anzeige- und Steuereinrichtungen ausgenutzt werden. Hierzu sollen sie bedarfsweise in anwendungsspezifische Gehäuse eingebaut werden. Der Einsatz ist auf typische Büro- oder Laborumgebungen beschränkt. Der handelsübliche Tablet-Computer ist kein Industriecomputer. Der Computer selbst soll bleiben, wie er ist; mechanische Eingriffe und Umbauten sind nicht vorgesehen.

Betriebssystem und Programmierung

Die Anwendungsprogramme müssen anwendungsspezifisch geschrieben werden. Das sollte freizügig möglich sein und nur wenig kosten. Die Programme sollen direkt auf dem Prozessor laufen und nicht über einen Interpreter ausgeführt werden. Es soll möglich sein, die Programme auf üblichen Personalcomputern zu entwickeln und zu erproben. Beim Stand der Technik werden diese Anforderungen auf einfachste Weise erfüllt, wenn man eine Windows-Systemumgebung verwendet.

Anwendungsschnittstellen

Die kleinen Tablets haben nur USB, Bluetooth und WLAN. Die einfachste Anwendungsschnittstelle (im Hinblick auf Kosten und einfaches Programmieren) ist USB mit Wandlung auf seriell.

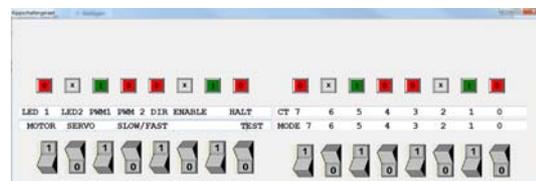
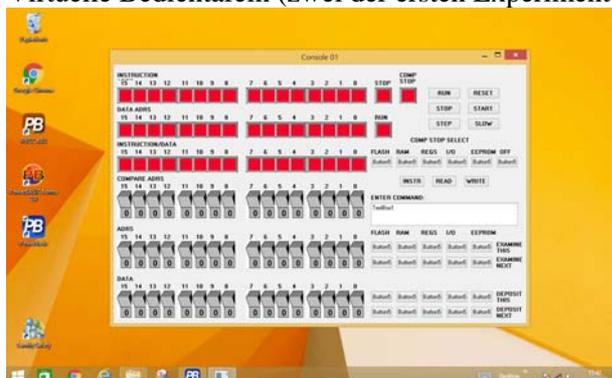
Grundkonfiguration

Tablet – USB-Verteiler (Hub) – Wandler USB-seriell – Mikrocontroller – Bedienmittel und Anwendungsperipherie.

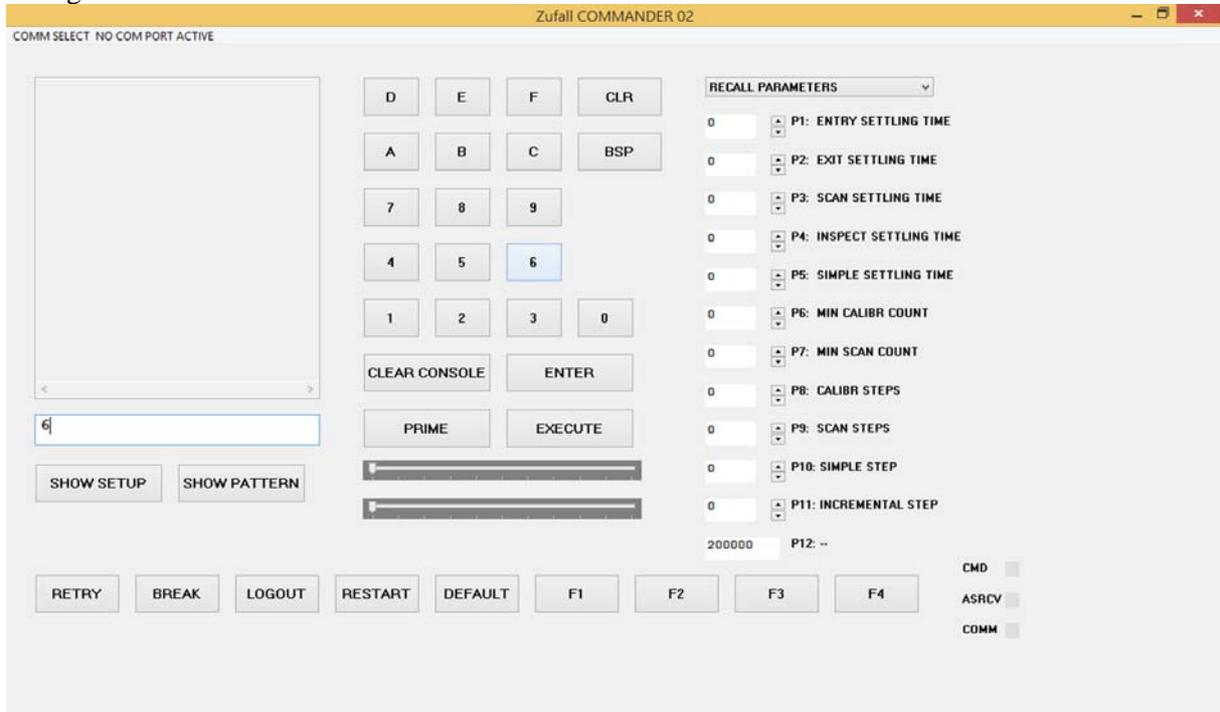
Der Touchscreen

Die Touchbedienung sollte eigentlich zusätzliche Bedienmittel überflüssig machen. Das ist aber nicht immer praktikabel.

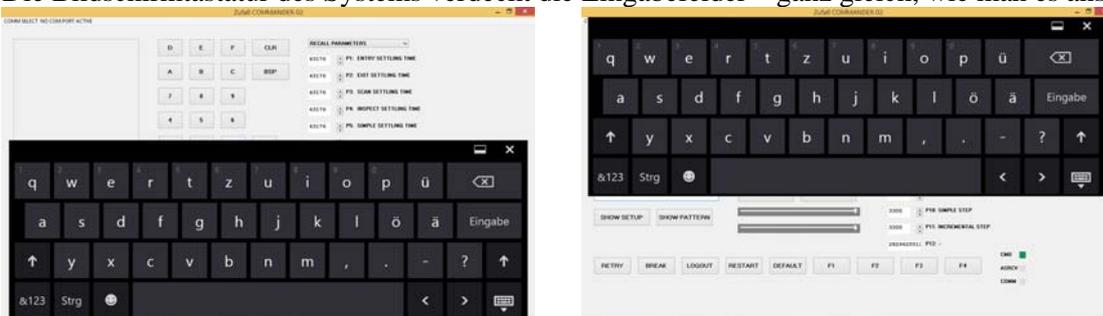
Virtuelle Bedientafeln (zwei der ersten Experimente).



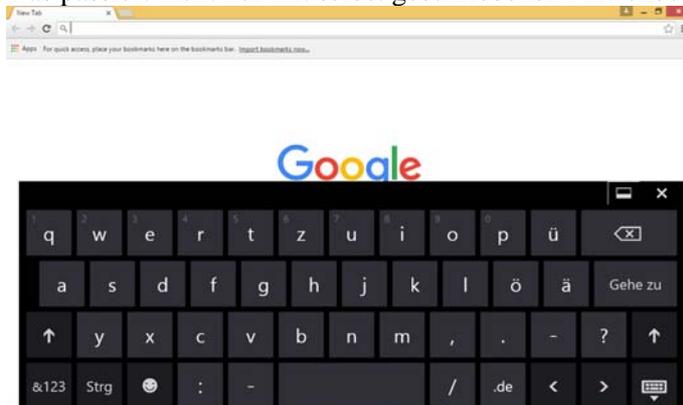
Ein Praxisbeispiel. Auf dem kleinen Bildschirm wird alles sehr klein... Die Schaltflächen (Buttons) lassen sich noch mit dem Finger bedienen, die Schieber (Sliders) und Drehfelder (Spin Boxes) hingegen kaum oder gar nicht.



Die Bildschirmtastatur des Systems verdeckt die Eingabefelder – ganz gleich, wie man es anstellt...



Das passiert nicht nur mit selbst geschriebenen Anwendungen:



Für kleine Tablets programmieren

Der zwar hochauflösende, aber kleine Bildschirm bedingt eine besondere Programmierweise. Berührbare Flächen (zur Touchbedienung) müssen groß genug sein. Die gesamte Bildschirmfläche muß ausgenutzt werden. Entsprechend dimensionierte Steuerelemente und Fenster werden aber auf üblichen PC-Bildschirmen meist viel zu groß dargestellt. Manche Steuerelemente lassen sich auf dem Touchscreen nur schwer bedienen. Das betrifft u. a. Schieber (Sliders), Bildlaufleisten (Scroll Bars) und Drehfelder (Spin Boxes). Die Bildschirmtastaturen des Systems nehmen sehr viel Fläche weg. Eingabefelder müßten so gelegt werden, daß sie von der Bildschirmtastatur nicht verdeckt werden. Das sieht aber nicht immer schön aus.

Entwurfsentscheidungen

Nur Touchbedienung oder zusätzliche Bedienelemente?

Das hängt von der Anwendung ab. Wenn das Tablet nur als Kippschalter- und LED-Ersatz dienen soll, genügt die Touchbedienung. Es ist ja genau der gewünschte Effekt, nämlich auf kostspielige und der mechanischen Gerätekonstruktion aufwendige Bedienelemente zu verzichten. Womöglich kostet es mehrere Anläufe, bis die Bedienoberfläche als gelungen angesprochen werden kann. Im Gegensatz zur Mechanik kann aber dabei nichts wirklich schiefgehen (falsch bemaßte oder zu Ausschuß gebohrte Frontplatten, übersehene Konfliktstellen beim Zusammenbau der Leiterplatten usw.).

Mausbedienung statt Touchbedienung?

Die Größe der anzuklickenden Steuerelemente wird im wesentlichen durch die Erkennbarkeit und Lesbarkeit (Beschriftung, Texte) bestimmt. Über die Maus läßt sich alles bedienen, was mit Klicken und Ziehen zu erledigen ist. Die Maus ist kostengünstig, handlich und einfach an den USB anzuschließen. Es ist aber ein zusätzliches Gerät, das zudem eine eigene Bedienfläche braucht.

Tastatureingabe?

Die echte Tastatur erspart die Bildschirmtastatur, so daß keine Bildschirmfläche für Eingabezwecke verloren geht. Anschluß über USB (mit oder ohne Draht) oder über Bluetooth. Die Texteingabe wird viel einfacher. Es ist aber auch ein zusätzliches Gerät.

Das Bildschirmtastaturproblem in der Anwendung lösen

Soll heißen: die Tasten, die man zum Eingeben in die Anwendung braucht, werden in der Anwendung selbst als Bildschirmtasten vorgesehen. Dann kann man sie dorthin setzen, wo es nicht stört, ja womöglich sogar gefällig aussieht. Dezimale und hexadezimale Tastenfelder sind typischerweise unproblematisch. Das Tablet ist nicht zum Surfen im Internet vorgesehen. Es soll vielmehr als eine Art Einzweckmaschine (elektronische Bedientafel usw.) für spezifische Anwendungen programmiert werden. Dann kann und sollte) man auch den gesamten Bildschirm ausnutzen. Anregungen zur Texteingabe:

- Eine eigene Bildschirmtastatur als graphisches Fenster, das frei über das Anwendungsbild geschoben werden kann. Abmessungen so, daß alles zusammenpaßt.
- Die Bildschirmtastatur mit anderem Formfaktor und Layout (kein QWERTZ). Anregungen geben u. a. manche der kleinen Beschriftungsgeräte (für Etiketten usw.). Nutzung ist aber gewöhnungsbedürftig...
- Kleine Bildschirmtastatur und Mausbedienung.
- Texteingabe über Dezimalastatur (wie SMS (Mobiltelefon)).

Zusätzliche eigene Bedienmittel?

Manchmal ist es eine Überlegung wert. Entscheidend ist, daß man Bedien- und Anzeigeeinrichtungen zum Einsatz bringen kann, die im Bereich der Personalcomputer nicht handelsüblich sind, wie Inkrementalgeber, Wahlschalter, Leuchtanzeigen, kleine Displays usw.

Ein erster Versuch.



Die Frage ist, wie man solche Armaturen ans System anschließt. Natürlich soll es einfach sein und nicht viel kosten.

1. Die Bedieneinrichtung erscheint wie eine Windows-Tastatur.

Das gehört seit vielen Jahren zum Stand der Technik. Es gibt eigens Firmen, die solche Dienstleistungen anbieten (also x-beliebige, manchmal sehr spezialisierte, Bedienpaneele mit einer standardkonformen Tastaturschnittstelle (USB oder PS/2) ausrüsten). Man braucht keine Gerätetreiber und kann die Ereignissteuerung des Systems direkt ausnutzen, um die Bedienfunktionen auszuprogrammieren. Bleibt die Frage nach einfacheren (bezahlbaren) Lösungen:

- Einen passenden USB-Controller einsetzen, entweder einen Tastaturcontrollerschaltkreis oder eine entsprechend programmierten Mikrocontroller mit USB-Schnittstelle und entsprechender Protokollunterstützung.
- Den USB-Controller einer Tastatur verwenden und die eigenen Bedienelemente passend an die Abfragematrix anschließen. Eine typische Bastellösung (zu der es etliche Anregungen im Internet gibt).
- Mit einem Mikrocontroller die PS/2-Schnittstelle nachbilden (viel einfacher als USB) und einen handelsüblichen Wandler PS/2 auf USB einsetzen.
- Die Windows-Eingabefunktion SERIALKEYS verwenden (Übertragung von Tastatur- und Maussignalen über die serielle Schnittstelle).

Ein typischer fertiger Tastaturcontroller hat auf der Geräteseite Abfragesignale für eine Kontaktmatrix. Sind andere Bedienelemente zu unterstützen (wie beispielsweise Inkrementalgeber, Wahlschalter und Potentiometer) verwendet, so muß der Tastaturcontroller so angesteuert werden, als ob eine Matrix aus abfragbaren Kontakten angeschlossen wäre. Die Bedienfunktionen müßten ggf. über einen Mikrocontroller umgesetzt werden (anstelle der Kontakte werden Schalttransistoren in die Abfragematrix eingebaut und passend angesteuert, bis hin zur Simulation von Zahleneingaben usw.).

2. Die Nutzung der externen Bedienelemente wird auf die eigene Anwendungsprogrammierung beschränkt.

Schema: eigene Bedienelemente – Mikrocontroller – serielle Schnittstelle – ggf. USB-Wandler. Eigene, anwendungsspezifische Signalprotokolle. Diese Auslegung läßt völlige Freiheit. Man kann darüber nicht nur Bedienelemente, sondern auch beliebige Anzeigeeinrichtungen unterstützen. Es gibt keinen Ärger mit der Windows-API oder mit den USB-Protokollen. Daß man das System selbst damit nicht bedienen kann, ist angesichts des besonderen Einsatzfalls nicht von Bedeutung.