
HIUC
Retro Style Computer

Der Hive ist ein vollständiger Computer mit selbst entwickelter Hard- und Software. Die Basis bilden drei Mikrocontroller P8X32A “Propeller” der Firma Parallax mit einer minimalistischen externen Beschaltung. Was ist der Hive also? Ein Retrocomputer? Ein Multicorerechner? Eine Bestie? Alle diese Antworten sind nicht wirklich befriedigend.

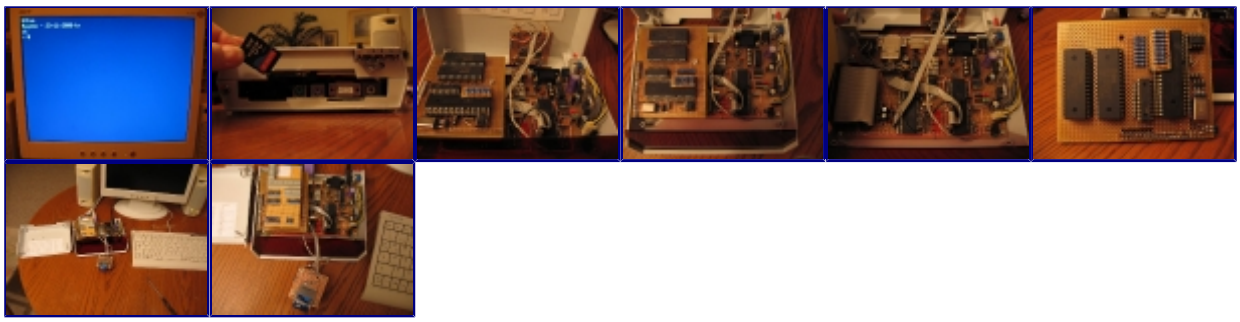
Schauen wir uns einfach mal den Retroaspekt an: Das Gerät orientiert sich im Design nur bei ganz bestimmten Punkten an den alten Geräten die wir unter "Retro" einordnen, verwendet nicht die alten Schaltungen oder bildet alte Hardware nach. Aber dennoch fühlt er sich in seiner Direktheit an wie eines der alten Systeme. Dazu trägt sicher auch sein einfaches und flaches Design bei: Zwischen der Hardware und der realisierbaren Software befinden sich nicht endlos viele abstrakte Ebenen, die es erst zu verstehen gilt. Wenige Zeilen Code bringen erstaunliche Ergebnisse, obgleich man ganz nah an der Hardware hantiert. Das kennen wir aus den Anfängen der Computerkultur. Man muß sich nur bewußt machen wie die Realität heut aussieht: Du hast keine Vorstellung mehr, was in deinem Computer abläuft, in welchem Systemdschungel sich Viren einnisten, oder in welcher der abertausend Registryeinträge eine falsche Einstellung einen Fehler verursacht? Ist diese Komplexität nötig, oder ist es mehr ein Mittel, um dem Benutzer die Kontrolle über das System aus der Hand zu nehmen und ihn zu einem phantasielosen Konsumenten zu machen?

Was also ist der Hive? Ich nenne ihn einen Retro Style Computer, weil es ein Versuch ist, die guten Seiten der alten Technik zu übernehmen und sie in eine neue Hardware, sowie ein einfaches Design zu steckt, und das alles aus einem Grund: um mit den alten Ideen und den neuen Möglichkeiten zu experimentieren. Mit den Retro-Homecomputern konnte man damals experimentell ein Prozessorsystem verstehen lernen, konnte ergründen, wie CPU, RAM, ROM und IO zusammenwirkt. Man konnte einen 8Bitter ohne Probleme selbst zusammenlöten und direkt in der Hardware agieren. Mit dem Hive kann man nun ebenso einfach zusätzlich mit der Parallelverarbeitung experimentieren, kann lernen, wie einfach es ist, Aufgaben auf mehrere Prozessoren zu verteilen und auch welche Probleme es dabei gibt. Mit der Programmiersprache Spin oder in Forth ist das so einfach wie mit dem ROM-Basic der alten Retros. Dabei gibt es keinen Anspruch auf Perfektion und kein Leistungsdenken - der Hive ist ein Outsider und Grenzgänger, und als Besitzer und Schöpfer geniesse ich das Privileg der gestalterischen Freiheit - ganz im Gegensatz zum beherrschenden Mainstream. Alle Informationen über den Hive sind frei zugänglich, die Hardware ist minimalistisch und jeder Interessierte kann sie selbst aufbauen und verstehen.

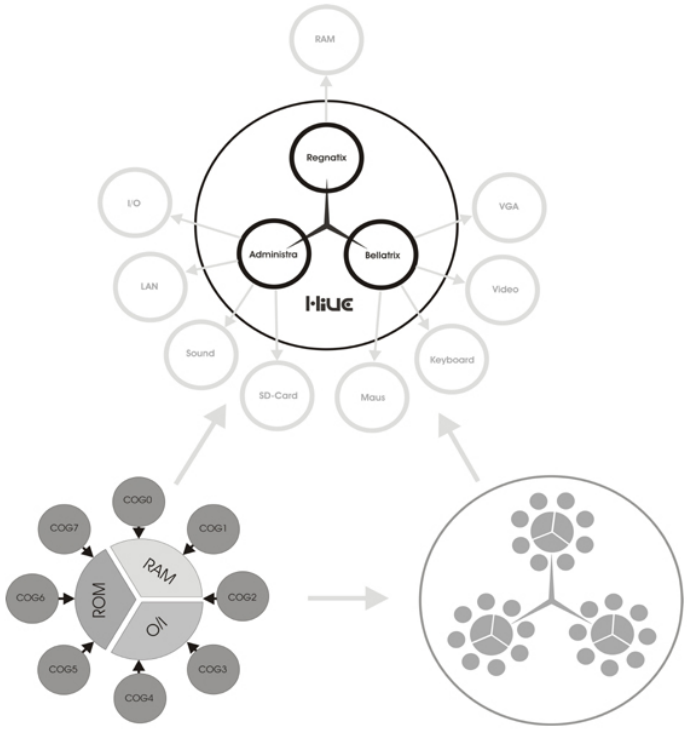
Wie schon geschrieben, ist der Hive kein Clone von einem bestehenden System und prinzipiell sind ihm an einigen Stellen deutliche Grenzen gesetzt. Aber als Kit mit einem Preis deutlich unter 100 €, mit der Möglichkeit wirklich das gesamte Gerät selbst von Hand ganz physisch aufzubauen, zu testen, zu verstehen (Und es ist wirklich einfach!) und in einer einfachen Sprache zu programmieren, ist der Hive sowohl ein guter Einstieg, wie auch durch seine besonderen Möglichkeiten eine Herausforderung. Mit welchem Bausatz kann man denn in so einfacher und vergnüglicher Weise in die Multi-Core-Technik einsteigen?

Auf der einen Seite ist es also ein kleiner Blick in die Zukunft, auf der anderen Seite ist es ein Blick zurück zu den Anfängen der Computertechnik - zum Ursprung. Und was die Grenzen betrifft, so gilt es, diese durch Kreativität zu ersetzen. Oder um es aus einer anderen Perspektive zu sehen: viele wunderbare Gedichte und Geschichten wurden einzig in der minimalistischen Einfachheit der Leere eines unbeschriebenen Blattes

geboren. Alles Fertige und Komplexe stört in diesen Moment - der ewige Ruf nach Leistung und Steigerung tötet das Wesentliche und die Kreativität. Drei gleiche Chips und ein wenig mehr müssen genügen... ;) Und letztlich ist das Gerät genau so in einer vergnüglichen Bastelwoche - just for fun - und ganz ohne große Anstrengungen entstanden: Artikel - Wie alles anfing



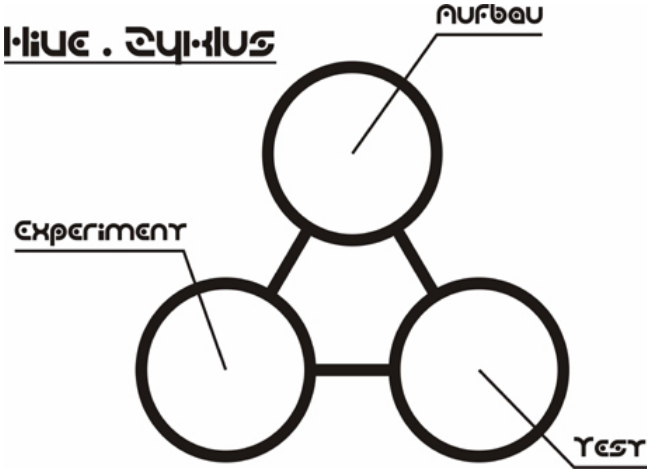
Viele Interessierte hatten Lust auf ein solches Retro-Abenteuer: Eine erste Platinenversion wurde entwickelt und eine Sammelbestellung organisiert. Insgesamt über 200 unbestückte Leiterplatten wurden bei Firmen in Auftrag gegeben und in alle Welt versendet. Mit einem solchen Kit war der Aufbau im Gegensatz zur ersten Version auf Lochrasterplatine zu einem Kinderspiel mutiert - für viele Hive-Erbauer war das Gerät der Erstkontakt mit Lötkolben und Messgerät und der im Selbstversuch erbrachte Beweis für das versprochene "Küchentischdesign". Jeder kann ein solches Gerät bauen, verstehen und auch programmieren!



Aber zurück zum schnöden technischen Design: Was seine Struktur betrifft, erweitert der Hive genau das, was Chip Grace mit dem Propeller-Chip genial vorgedacht hat: Wie acht gleichartige Elemente kombiniert einen universellen Chip bilden, so bildet der Hive eine Ganzheit aus drei dieser gleichartigen Schaltkreisen. Der Vorteil liegt klar auf der Hand: Lerne mit einem Chip umzugehen und du verstehst das ganze Gerät. Einfachheit ist hier das Prinzip. Jedes Teil ist in sich geschlossen, einzeln testbar. Einzig die Funktionskomplexe differenzieren das Ganze.

Die wesentlichen Prämissen beim Entwurf des Gerätes kann man also in folgenden Ideen zusammenfassen:

Einfacher und sicherer Aufbau: Dazu gehört die Verwendung von einfach zu beschaffenden und zu verarbeitenden Bauelementen. Damit verbieten sich alle Komponenten, welche ausschließlich in SMD-Gehäusen geliefert werden. Der Hive besitzt also ein “Küchentisch-Design”: Bei vorhandener Leiterplatte sollte es fast Jedem ohne Probleme möglich sein das Gerät in wenigen Stunden auf dem “Küchentisch” nachzubauen. Alle Bauelemente, bis hin zum Gehäuse und zur Tastatur, sollten aus leicht zugänglichen Quellen beschaffbar sein. Mit der nun verfügbaren Leiterplatte wird der Aufbau zu einem echten Erlebnis, Lötfehler sind selbst bei ungeübten Lötern durch den Stopplack kaum noch möglich.



Ein entscheidender Punkt im Design des Hive ist aber der enorm kurze Zyklus zwischen Aufbau, Test und Experiment. Selbst wenn man erst anfängt den Hive aufzubauen, kann man schon nach kurzer Zeit mit dem System experimentieren. In der Bildergalerie "Prototyp 2 Aufbau" zum Aufbau des zweiten Prototypen habe ich versucht das ein wenig zu dokumentieren: Nach kürzester Zeit konnte ich mit einem kleinen Programmschnipsel für Bellatrix die Heartbeat-LED blinken lassen. Acht weitere Widerstände und eine Buchse (Lötzeit 15 Min) später konnte ich das erste Bild auf einem VGA darstellen. Bei jedem dieser Schritte kann man schon hemmungslos mit dem System experimentieren, hat von Anfang an einen funktionierenden und "lebenden" Experimentierbaukasten, der durch seine Offenheit und Einfachheit zum Spielen einlädt. Und so setzt sich der gesamte Aufbau fort. Wo bei anderen Systemen erst eine relativ große Schwelle bis zum ersten Erfolg zu überwinden ist, wartet beim Hive nach jedem kleinen Aufbauschritt neue Entdeckungen. Und wenn man nach dem Aufbau stolz sein Hive auf den Tisch stellt, hat man quasi nebenbei einen Crashkurs in vielen relevanten technischen Bereichen hinter sich, kennt sich schon bestens mit dem System aus und kann sofort mit den folgenden Abenteuern beginnen.

Einfache Programmierung: Das Betriebssystem TriOS basiert auf der Sprache SPIN, hat eine sehr einfache Struktur und ist gut für Einsteiger geeignet. Mit dem integrierten PropForth ist eine Programmiersprache mit Compiler und Interpreter sofort nach dem Einschalten verfügbar. Und wer mag kann auch ohne Probleme ein völlig eigenes Betriebssystem entwickeln, eigene Vorstellungen verwirklichen oder einfach nur experimentieren. Fühl dich frei: Von der blanken Hardware bis zu einem eigenen OS stehen nur wenige Stunden Arbeit!

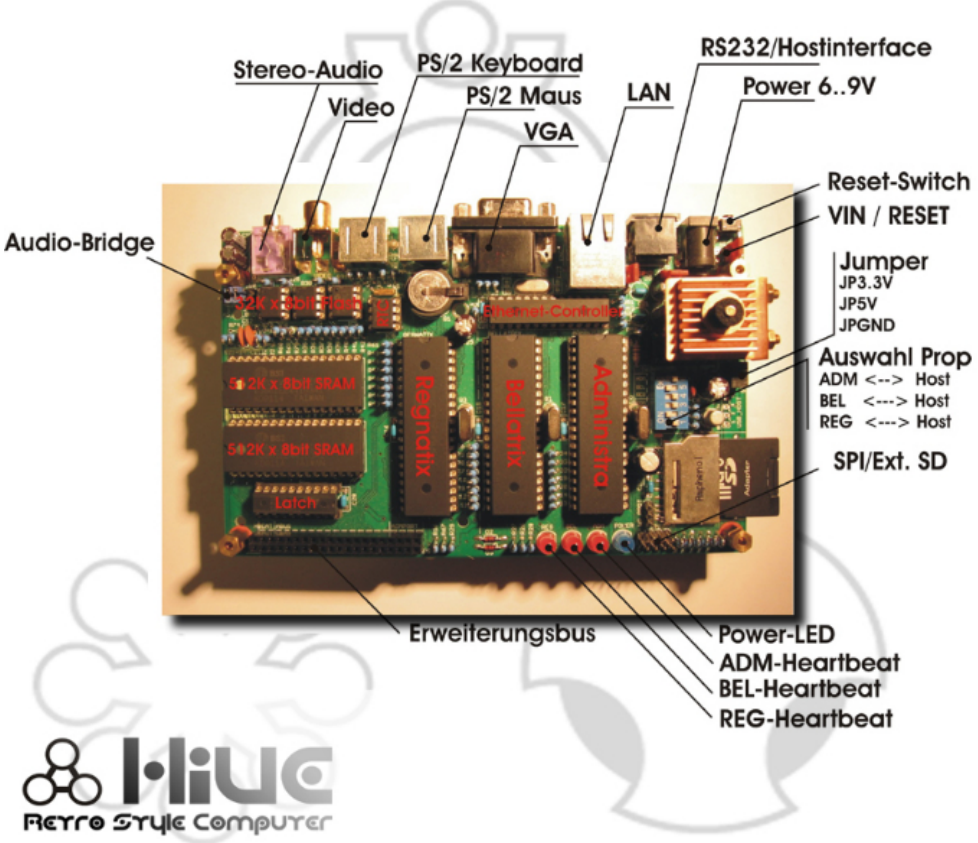
Lernen durch Selbstmachen: Alle Komponenten - sowohl die Hardware als auch die Software - sollten für jeden Interessierten durchschaubar und selbst nachbaubar sein.

Dennoch läßt das System genug Möglichkeiten und Freiheiten für eigene Experimente. Spaß an der Freude ist ein erklärtes Ziel und ist der Grund aus dem der Hive entstanden ist. Durch einen modularen Erweiterungsbus kann jeder das System erweitern, und da sowohl die Hardware wie auch die Software keine Geheimnisse hat, sollte das auch für jeden möglich sein.

Modularität und Erweiterbarkeit: Am aktuellen Prototyp ist es noch nicht sichtbar, aber die fertige Platine wird die Modularität zum Ausdruck bringen, denn sie soll so gestaltet werden, daß man sowohl ein recht kompaktes und kleines, als auch ein modulares Gerät als Steckkartensystem aufbauen kann. So sind Hardwareexperimenten und Erweiterungen keine Grenzen gesetzt.

Technische Daten

Öffnen wir die Motorhaube des Hive und schauen uns sein Herz an - oder besser seine drei Herzen: Der Kern des Gerätes besteht aus drei Mikrocontroller P8×32D40 “Propeller” der Firme Parallax. Jeder dieser Mikrocontroller enthält 8 RISC-Kerne (die sogenannten COG’s) und eine ganze Menge begleitender Peripherielogik. Die einzige nennenswerte externe Zusatzhardware sind ein oder zwei externe Speicherbänke und ein dazugehöriges Adresslatch. Noch einige passive Bauelemente plus ein wenig Verdrahtung und fertig ist der Mini-Supercomputer. Genaue Informationen zum Propeller findest du im Wikipedia und einen Hardwareüberblick hier bei der Herstellerfirma.



Wie schon beschrieben stecken im Hive drei Propellerchips, wobei jeder einen ganz bestimmten Funktionskomplexe bedient:

1. Chip Host “Regnatix” - Hostsystem

- Externer RAM 1 MByte in zwei Bänken 512 kByte statischer RAM
- Bussystem 8 Bit für die Anbindung der zwei anderen Microcontroller und den Erweiterungsbus

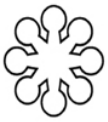
Der erste Chip steht als Host fast vollständig dem Anwender für seine Programme zur Verfügung. Da sehr viele komplexe Systemfunktionen in die beiden Slavecontroller ausgelagert sind, ergeben sich sehr kleine Programmgrößen für die eigentliche Software. Als Beispiel möge der StarTracker dienen: Dieser besitzt eine bescheidene Programmgröße von 4,9 kByte für das Programm selbst im Host. Sieben freie Prozessoren, 32 kByte interner RAM des Chips und 1 MByte externer RAM sollten auch für relativ komplexe Anwendungen ausreichend sein.

2. Chip Slave "Bellatrix" - Intelligentes Grafiksubsystem

- VGA Auflösungen bis 1280 x 1024 Pixel im Textmodus
- Composit-Video
- Anschluß für Tastatur und Maus (PS2)

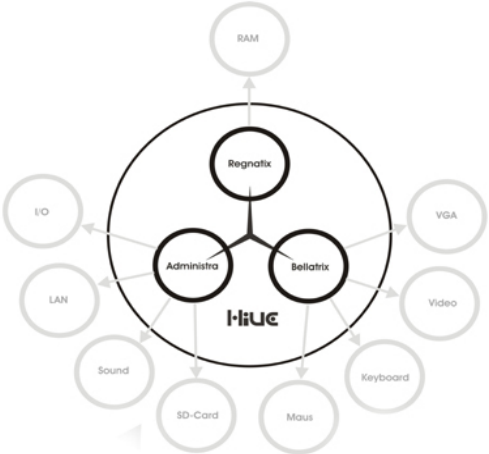
Der zweite Mikrocontroller ist für das Benutzerinterface zuständig. Sowohl Bildausgabe über VGA oder Composit-Video, als auch die Eingabegeräte werden hier angesteuert. Die momentanen Funktionen gestatten es dem Host, einen beliebigen neuen anwendungsspezifischen Programmcode in das Grafiksubsystem zu übertragen, um die dort vorhanden acht Prozessoren in speziellen Funktionen einzubinden. So könnte der Chip durchaus völlig selbstständig Menüs darstellen und verwalten, da er auch die Mausaktionen überwacht. Bei entsprechender Software muß der Hostchip sich also um solche Kleinigkeiten wie Menüs, Fenster und Benutzeraktionen nicht unbedingt kümmern. Als weiteres Beispiel denke man an Computerspiele: In Bellatrix könnte eine komplette Grafikengine laufen, die vielleicht völlig eigenständig 3D-Objekte oder Sprites dreht, transformiert und bewegt, Ebenen scrollt und Effekte erzeugt.

3. Chip Slave "Administra" - Allgemein I/O-Interface



- Ansteuerung und Verwaltung einer SD-Card (bis 2 GByte)
- FAT16/32-Dateisystem mit Verzeichnissen
- Blockinterface mit Screenpuffer für FORTH, schneller wahlfreier Zugriff auf große Datenbestände
- (HSS) Engine um komplexe vierkanalige Soundtrackermodule zu puffern und abzuspielen
- (HSS) 2 Kanal FX-Synthesizer (NFO/LFO/Hüllkurve…)
- (WAV) Stereo Engine zur Wiedergabe von WAV-Dateien direkt von SD-Card
- (SID) Stereo-SID (2 x C64-Soundchip)
- Player SID-DMP-Files direkt von SD-Card
- Ethernet-Interface, TCP/IP-Stack
- IO-Port

Administra bietet dem Host momentan 39 (Stand 03/2009) teilweise recht komplexe Funktionen. So ist es nicht nötig, daß sich der Host um "Kleinigkeiten" wie FAT, Dateinamen oder Sound kümmert. Viele der Funktionen der angeschlossenen Hardware sind in diesen Mikrocontroller ausgelagert.



Wie man sieht hätte ein einzelner Propeller-Chip zu wenig Resourcen um einen kompletten Computer zu realisieren. Aber ein System aus drei dieser Schaltkreise bietet genug Leistung, um ein komfortables und flexibles System zu bilden. Der größte Teil der Funktionen wird durch Software realisiert, was die Schaltung erheblich vereinfacht. So bedarf es als Beispiel für den Anschluß eines VGA-Monitors nur einer VGA-Buchse und acht Widerstände. Den Rest erledigt Software. Genial einfach, einfach genial…

Ein weiterer Nebeneffekt ist bei diesem Design, daß

die Funktionen nicht wirklich scharf voneinander abgegrenzt sind. In der bisherigen Systemsoftware ist Regnatix der Host, aber es ist durchaus auch möglich das Bellatrix die Kontrolle übernimmt, Administra die Zusatzfunktionen realisiert und Regnatix durch die passende Software zu einem schnellen Speichermanager wird. Was du genau aus dem Hive machst, unterliegt einzig der Freiheit deiner Phantasie.