

Neueste Entwicklung an Netzer 2:0

Features

- Steuercontroller (PIC18, PIC24, dsPIC oder PIC32)
- Ethernetcontroller ENC424J600
- Pin- und formfaktorcompatibles Interface zu Netzer 1
- MicroSD-Kartenslot (per SPI angebunden)
- Externer RAM-Speicher (Parallel oder per SPI angebunden)
- Optional: Externes EEPROM (Per SPI angebunden)
- Versorgung mit 3.3V oder 5V (?)

Überlegungen zu den Controllertypen

Aus Platz- und Preisgründen wird ein 64-Pin-Gehäuse in der Bauform QFN bevorzugt. Soweit bekannt, gibt es von jeder der angestrebten PIC-Familien ein passendes Derivat.

PIC24



Hier ist der Typ **PIC24FJ256GA106** der Passende.

Für das 64-Pin Gehäuse hier noch die passende Pinmap:



Stromversorgung



 Der Controller verfügt über einen internen Spannungswandler, der durch externe Beschaltung konfiguriert wird. Für Netzer **muss** der Spannungswandler verwendet werden, um eine Stromversorgung mit 3,3V zu machen.

Die Operation bei maximalen Takt von 32 MHz ist gewährleistet, da der Spannungswandler ausreichend Spannung für VDDcore erzeugt. **Brown-Out-Detektor** ist vorhanden (Spezifikation siehe Tabelle).

Reset (MCLR)

Empfohlener Schaltkreis laut Datenblatt (entspricht der aktuellen Netzer-Schaltung).

OSC Beschaltung



Es gibt folgende Möglichkeiten:

1. EC (Externer Takt mit 0-32 MHz)
2. ECPLL (Externer Takt mit 4-8MHz, mit PLL)
3. HS (10MHz-32MHz Quarz)
4. HSPLL (10MHz-32MHz Quarz, mit PLL)
5. Intern (mit und ohne PLL)

Die PLL arbeitet mit einem fixen Faktor von 4. 32MHz ist der höchste interne Takt.

Programmierung

Es gibt drei ICD-Pinpaare, die per Fuse konfiguriert werden können. Default sind PGEC1 (Pin 15) und PGED1 (Pin 16) ausgewählt, ggf. kann das aber durch einen vorinstallierten Bootloader geändert werden. Die beiden Pins können auch als analoge Eingänge verwendet werden.

PIC18



Hier ist der Typ **PIC18F67K22** der Passende.

Für das 64-Pin Gehäuse hier noch die passende Pinmap:



Stromversorgung



Der Controller verfügt über einen internen Spannungswandler, der durch externe Beschaltung konfiguriert wird.

Für maximalen Takt muss **mindestens mit 3V** versorgt werden. **Brown-Out-Detektor** ist vorhanden und konfigurierbar (1,8V, 2,0V, 2,7V und 3,0V).

Übersicht

| Signal | Pinliste |
|---------|--|
| GND | 9, 18 (ENVREG disabled), 20 (AD), 25, 41 und 56 angeschlossen werden. |
| VDD | Pins 19 (AD), 26, 38 und 57 angeschlossen werden. |
| VDDCore | Pin 10 (wie links in der Grafik zu sehen.) Cf ist ein low ESR Typ (< 5 Ohm) mit Wert 10uF. |

Reset (MCLR)

Empfohlener Schaltkreis laut Datenblatt (entspricht der aktuellen Netzer-Schaltung).

OSC Beschaltung



Der maximale durch PLL generierte Takt ist 64 MHz. Die PLL läuft mit einem fixen Faktor von 4, d.h. ein Quarz mit **16 MHz** ist die ideale Taktquelle. Für serielle Anwendungen ist ein Quarz mit 14,7456 MHz ideal. Möglicherweise kann auch alles über den internen Takt gemacht werden, sollte das RTCC-Modul genaue Zeitstempel für das Task-Switching generieren können.

Programmierinterface

Wie gewohnt werden PGD (RB7, Pin 37) und PGC (RB6, Pin 42) rausgeführt, ggf. auf Pinleisten. Falls sie auf Pinleisten geführt werden, sollten Serienwiderstände bis 100 Ohm eingefügt werden.

Parallelport

Der Ethernetcontroller und ggf. ein SRAM-Baustein werden über ein paralleles Interface angeschlossen. Dieser Controller weist keine dedizierte parallele Schnittstelle auf, so dass beliebige Portpins verwendet werden können.

Pinconstraints

Reset

Der RESET-Pin ist für alle hier betrachteten Controller **Pin 7**. Die externe Beschaltung ist ähnlich. Der Serienwiderstand variiert im Wert (Bestückungsoption?!). Ggf wird der Stützkondensator für Debugging und Programmierung entfernt. In der Serie wird ein Bootloader verwendet, so dass der Kondensator bestückt wird. Der Pin wird wie bei Netzer 1.X auf den Netzer-Pinheader geführt.



Stromversorgung

Pin 18

ENVREG des PIC18 kollidiert mit RB7 der anderen Controller. Für die weitere Diskussion wird angenommen, dass RB7 immer als Eingang konfiguriert ist.

1. ENVREG wird hart auf Masse gelegt, dann operiert der PIC18 immer mit 3,3V.
2. ENVREG wird per Widerstand auf Masse oder VCC gelegt. Dann wäre theoretisch auch VCC=5V möglich, jedoch müssen dann alle ICs auf dem Netzer das können, was mit dem EthernetController schonmal nicht geht.

Entscheidung: **Möglichkeit 1)**

Pin 10

VDDcore des PIC18 kollidiert mit VDD der anderen Controller. In jeden Fall wird ein 100nF nach Masse gesetzt. Ein optionaler 0R-Widerstand für PIC24 und PIC32 nach VDD stellt die Verbindung her. Im Falle des PIC18 bleibt dieser Widerstand unbestückt.

Pin 56

Hier kollidiert VSS des PIC18 mit VDDCORE der anderen Controller. Im Falle der anderen Controller wird ein 10uF an der Stelle benötigt, der im PIC18-Fall einfach durch einen 0R-Widerstand ersetzt wird.

Keine weiteren Konflikte.

Quarz

Der PIC18 kann mit maximal 16 MHz versorgt werden, die er intern auf 64 MHz hochtaktet. Der PIC24 kann von 8 MHz bis 32 MHz versorgt werden (ideal sind **8 MHz**). PIC32 muss noch näher evaluiert werden. Es sollte außerdem die Option vorgesehen werden, den Ausgangstakt des Ethernet-Controllers als Taktquelle zu nutzen, dieser wird dann an OSCIn angeschlossen.

Taktoptionen des ENC424J600:



Parallel angebundener Ethernet-Controller

Der Ethernet-Controller ENC424J600 wird parallel angebunden (Mode 5 oder 6), um einen höheren Durchsatz als bei SPI-Anbindung zu erreichen. Der erweiterte Adressbus wird ebenfalls angebunden, um vollen Zugriff auf den Speicher des Controllers zu erlangen. Einige Controller verfügen über ein Parallel Master Port (PMP), damit geht der Zugriff dann noch schneller. Für das PMP wurde das Pinning optimiert.

| Pinname | ENC424 | Netzer-uC | PIC18F6XK22 | PIC24FJ256GA106 | dsPIC PIC32 |
|---------|----------|------------------------------|--------------------------|--------------------------------|-------------|
| AD0 | 38 | 60 | RE6/P1B/CCP6 | CN58/PMD0/RE0 | |
| AD1 | 39 | 61 | RE5/P1C/CCP7 | CN59/PMD1/RE1 | |
| AD2 | 40 | 62 | RE4/P3B/CCP8 | CN60/PMD2/RE2 | |
| AD3 | 41 | 63 | RE3/P3C/CCP9/REF0 | CN61/PMD3/RE3 | |
| AD4 | 5 | 64 | RE2/CS/P2B/CCP10 | CN62/PMD4/RE4 | |
| AD5 | 6 | 1 | RE1/WR/P2C | CN63/PMD5/RE5 | |
| AD6 | 7 | 2 | RE0/RD/P2D | SCL3/CN64/PMD6/RE6 | |
| AD7 | 8 | 3 | RG0/ECCP3/P3A | SDA3/CN65/PMD7/RE7 | |
| AD8 | 25 | 32 | RC7/RX1/DT1 | SCL2/RP17/CN18/PMA8/RF5 | |
| AD9 | 26 | 31 | RC6/TX1/CK1 | SDA2/RP10/CN17/PMA9/RF4 | |
| AD10 | 27 | 28 | RA4/T0CKI | TDI/PMA10/AN13/CTED1/CN31/RB13 | |
| AD11 | 28 | 27 | RA5/AN4/T1CKI/T3G/HLVDIN | TCK/AN12/CTED2/CN30/PMA11/RB12 | |
| AD12 | 29 | 24 | RA0/AN0/UPLWU | TDO/AN11/CN29/PMA12/RB11 | |
| AD13 | 30 | 23 | RA1/AN1 | TMS/AN10/CVREF/CN28/PMA13/RB10 | |
| AD14 | 31 | 44 | RB4/KBI0 | RP3/CN55/PMCS2/RD10 | |
| CS | 34 (VDD) | NC | | | |
| WR/EN | 35 | 49 auf PIC18, sonst 52 | RD7/PSP7/SS2 | RP25/CN13/PMWR/RD4 | |
| RD/RW | 36 | 53 | RD3/PSP3 | RP20/CN14/PMRD/RD5 | |
| AL | 37 | 30 | RC0/SOSCO/SCLKI | AN15/REFO/RP29/CN12/PMA0/RB15 | |
| INT | 24 | 43 | RB5/KBI1/T3CKI/T1G | RP4/CN54/RD9 | |

From:

<http://www.mobacon.de/wiki/> - MoBaCon Wiki

Permanent link:

<http://www.mobacon.de/wiki/doku.php/intern/dev>

Last update: 2014/02/09 13:32

