lome6

(Lights-Out-Management-Ether6)

by warhog <warhog@gmx.de> 2011

Inhaltsverzeichnis

1 GPL

2 Einleitung

Da meine Server (Backup und Web+Fileserver, beide Linux) im Keller stehen und ab und an (sei es durch Spieltrieb oder Stromausfall etc...) diverse Probleme haben, habe ich beschlossen ein einfaches Lights Out Management (LOM) System auf Basis des Ethersex Projektes zu realisieren.

3 Funktionen

Anforderungen an den ersten Prototyp:

- Reset und Power Taster vom Gehäuse über das LOM durchschleifen (Reset, Power kurz und Power lang drücken)
- Serielle Schnittstelle per yPort weiterleiten
- Temperatursensoren für Lufteinlass, RAM und Netzteilauslass per OneWire (DS18x20)
- LCD Display vorne am Server
- Status LEDs vorne am Server
- Externe Temperatur für CPU und SB (Southbridge) vom Mainboard
- Daemon der CPU- und Southbridgetemperatur und die Serveruptime ans LOM schickt
- WHM/Uptime
- Anzeige Serverstatus (An, Aus) per Optokoppler von der PowerLED abgegriffen
- LOM disable Schalter um das LOM temporär zu deaktivieren

Zukünftig angedacht:

- Serverdaemon der ein Heartbeat Signal ans LOM schickt, fehlt dieses für ~10sek -> Server hart resetten
- SNMP Support
- mehr LCD Typen
- IPv6 Support (mangels Testnetz und eigenem Wissen aktuell nicht möglich)

4 Erster Prototyp

Mittlerweile läuft der erste Prototyp stabil, der Schaltplan findet sich unter Downloads.
Eine erste Prototypenplatine findet sich auch, allerdings hat diese noch ein paar Fehler, sodass eine neue geroutet werden sollte.
Der Prototyp benutzt statt Optokopplern noch Reedrelais und holt den Power Status noch direkt vom +5V Pin des ATX Stecker. Ebenso kommt die Stromversorgung noch direkt vom ATX Stecker Standby Pin.

Ein Problem im Betrieb trat mit dem yPort Protokoll auf. Regelmäßig verschluckte dieses Zeichen oder hängte sogar das Ethersex auf.
Besonders ärgerlich war das wenn man dem Server beim booten zugeschaut hat. Das Problem lässt sich beheben wenn man die Größe des Puffers von yPort auf 512byte oder mehr erhöht.

5 Hardware

Die Hardware ist an das Netio angelehnt.
Als CPU habe ich mich für einen atmega1284 entschieden, dieser hat genug RAM und Flashspeicher für zukünftige Entwicklungen. Ein atmega644 reicht aber auch, wenn man HTTP und LCD deaktiviert reicht theoretisch sogar ein atmega32 (nicht getestet).
Als LCD wird aktuell nur ein HD44780 16x2 Zeichen unterstützt.
Unter /pinning/hardware/ findet sich die lome6.m4. Dort sind alle Portdefinitionen eingetragen.
Die Stromversorgung ist über die Standby Versorgung der internen USB Header realisiert.
Den aktuellen Power Status (Server ein/aus) findet das lome6 über einen weiteren Optokoppler von der PowerLED.
Die Taster (Reset + Power) vom Gehäuse zum Mainboard werden über das LOM geschleift und dort per Optokoppler vom LOM getriggert.
Die serielle Schnittstelle wird über einen MAX232 realisiert.

6 Software

6.1 Ethersex

Eingestellt werden die Daten des lome6 Moduls im Makeconfig Menü unter "Applications/lome6 Service".
Dort kann der LCD und OneWire Support und die Zeiten für das Triggern der Taster eingestellt werden. Zur Sicherheit sollte noch eins der PAM Module aktiviert werden!

6.2 Externer Client

Das lome6 kann über den externen Client gesteuert werden (lome6-gui).
Dieser ist in services/lome6/lome6-gui zu finden. In bin/ liegt die x64 Binary. Zum kompilieren benötigt man Qt.
Es sind noch nicht alle Features fertig, z.B. kann man die Konfiguration bisher nur auslesen und es ist bisher nur IPv4 getestet.

6.3 Webseite

Die andere Möglichkeit zum Steuern des lome6 ist der HTTP Server. Dazu den Inhalt des Verzeichnisses services/lome6/embed ins embed/ kopieren und den HTTP Server anschalten.

6.4 Serverdaemon

Ich habe einen kleinen Daemon für den Server geschrieben der die aktuelle CPU und Southbridge Temperature per lm-sensors und die Uptime ausliest und per ecmd ans lome6 sendet. Dieser liegt im services/lome6/lome6d Verzeichnis und ist in C++/Qt geschrieben. Binaries für x64 liegen im bin/ Verzeichnis.
Ist der Server aus, werden die Werte automatisch resettet bzw. als 0°C oder 0h angezeigt.

7 ECMD

lome6 erweitert die ECMD um folgende Befehle:

lome6 state

Power Status (on, off)

lome6 power

Trigger Power Taster kurz

lome6 power long

Trigger Power Taster lang

lome6 reset

Trigger Reset Taster

lome6 set_t <*type*> <*temperatur*>

Setze Temperatur, *type* kann "cpu" oder "sb" sein, *temperatur* in Decigrad!

lome6 get_t <*type*>

Hole Temperatur, *type* kann "cpu", "sb", "ram", "psu" oder "air" sein

lome6 uptime <*uptime*>

Wenn *uptime* angegeben, wird diese gesetzt (in Sekunden).

Ansonsten wird die Uptime (in Sekunden) ausgegeben.