

Applicazione per pure.box:

## Eseguire il proprio programma C sulla pure.box

Panoramica del prodotto

Panoramica dell'applicazione

Il seguente esempio fornisce una panoramica delle operazioni necessarie, per compilare un testo sorgente C per il sistema finale della pure.box, caricare il programma compilato sulla box ed eseguirlo periodicamente. La funzione del presente programma d'esempio consiste nel commutare ogni minuto lo stato di attivazione di un output di un Web-IO digitale.

La pure.box consente di eseguire programmi compilati per il suo sistema all'avvio del sistema, in modo temporizzato o alla ricezione di dati TCP o UDP. Vengono supportate le librerie:

- GNU C-Lib v2.1
- SQLite v3.6
- OpenSSL v0.09
- zlib v1.2

### Download e installazione del cross-compiler

Nel processore Coldfire impiegato nella pure.box gira un sistema operativo GNU/Linux. Il cross-compiler necessario per questa combinazione processore/sistema operativo (incl. documentazione) può essere scaricato [qui](#) gratuitamente nella versione Lite per Windows e Linux. Potete ottenere informazioni sul processo d'installazione nel sistema host dalla documentazione.

### Compilare testo sorgente in Windows

Il testo sorgente per il presente programma d'esempio può essere scaricato [qui](#). Oltre al file C l'archivio contiene anche il programma compilato pure.box già pronto (**output\_toggle**).

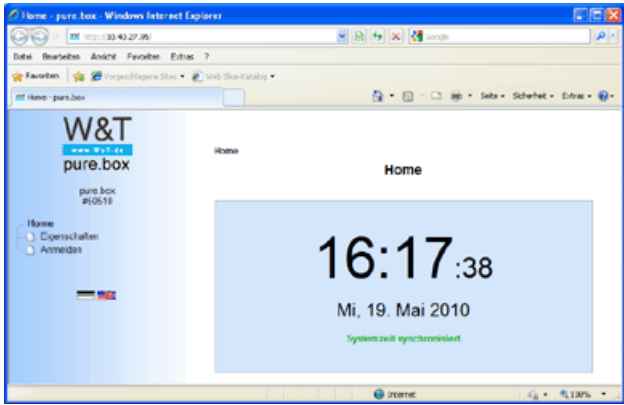
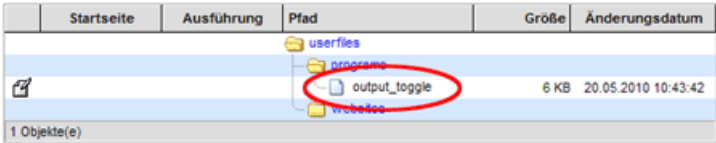

Salvare il file **output\_toggle.c** sul disco rigido e aprire la richiesta di immissione Window. Navigare nella directory in cui avete precedentemente archiviato il file del testo sorgente. Il programma compilato genera il seguente comando:

```
m68k-linux-gnu-gcc -mcpu=54450 -o output_toggle output_toggle.c
```

Il compiler **m68k-linux-gnu-gcc** viene richiamato indicando anche il tipo di processore della piattaforma target (**mcpu=54450**). Il file oggetto da generare viene fissato con **-o output\_toggle**. Infine viene indicato il file C (**output\_toggle.c**). Se il comando è stato eseguito correttamente, troverete il file del programma generato **output\_toggle** nella stessa directory del testo sorgente originale.

### Configurazione della pure.box

Salvare il file **output\_toggle** per FTP o via condivisione SMB nella cartella **programs** sulla pure.box. Infine sono necessarie le seguenti operazioni di configurazione:

<p>Richiamate la <b>homepage</b> del pure.box inserendo l'<b>indirizzo IP</b> nella riga dell'indirizzo del vostro browser.</p>	
<p>Scegliere <b>Login</b> nel menu ad albero ed effettuare il login attraverso la finestra di dialogo richiamata con la vostra password.</p>	<p style="text-align: center;"><b>Anmelden</b></p> <div style="text-align: center;"> <input type="password" value="Passwort"/>   <input type="button" value="Senden"/> </div>
<p>Navigare sulla pagina <b>Home &gt;&gt; Configurazione &gt;&gt; Propri file</b>. Cliccare nella panoramica file sulla cartella <b>programs</b>. Il file <b>save_data.php</b> viene visualizzato.</p>	
<p>Un click sull'icona <b>Modificare le proprietà del file</b> (a sinistra vicino <b>output_toggle</b>) apre la finestra di dialogo Modifica.</p>	
<p>Attivare l'opzione <b>Programma</b>. Inserire in <b>parametri per richiamare</b></p>	

#### dati:

indirizzo IP del Web-IO digitale  
Numero della porta HTTP  
canale da azionare periodicamente  
ev. password assegnata  
Se non avete assegnato nessuna  
password, il richiamo potrebbe essere  
simile al seguente:  
**10.40.27.10 80 5 ""**

L'allineamento dei **parametri per richiamare dati** è identica a quella di un richiamo diretto del programma dalla console.

Ausführen als	<input type="radio"/> PHP-Script <input checked="" type="radio"/> Programm (kommandozeilenausführbar)	Ausführung über den Webserver oder die Kommandozeile
Aufrufparameter	10.40.27.10 80 5 "" z.B.: x=1&y=hello%20world	

Attivare **in modo temporizzato** e accertatevi che in ognuno dei seguenti campi di testo sia segnato un \*. Secondo questa definizione CRON il programma **output\_toggle** ogni minuto con i valori fissati nei **parametri per richiamare dati**.

Accettate tutte le nuove impostazioni con **Salva... Fatto!**

Zeitgesteuert	<input checked="" type="checkbox"/>	Komplette CRON-Syntax erlaubt
Minuten	<input type="text"/>	0-59
Stunden	<input type="text"/>	0-23
Monatstage	<input type="text"/>	1-31
Monate	<input type="text"/>	1-12
Wochentage	<input type="text"/>	0-6 (0 = Sonntag)

Da adesso il Web-IO specificato nei parametri per il richiamo di dati viene interessato dalla pure.box via rete Lo stato di attivazione dell'output indicato viene cambiato ogni volta.

## Il testo sorgente

Nelle prime righe vengono indicati i file header necessari.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <netinet/in.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <arpa/inet.h>
```

Il programma è costituito esclusivamente da una funzione principale. All'inizio vengono stabilite le variabili locali. Poi vengono visualizzati i valori trasmessi tramite trasmissione dei parametri. Questa visualizzazione viene soppressa all'avvio periodico attraverso la pure.box, tuttavia è molto utile nella fase di sviluppo, in cui il programma viene eseguito anche localmente al PC.

```
int main(int argc, char **argv) {
    int handle;
    char command[128];
    struct sockaddr_in remote;
    printf("address: %s\r\n", argv[1]);
    printf("port: %i\r\n", atoi(argv[2]));
    printf("channel: %i\r\n", atoi(argv[3]));
    printf("password: %s\r\n", argv[4]);
    ...
}
```

Infine viene creato il comando per il blocco dell'output e salvato nelle variabili **command**.

```
sprintf(command, "GET /outputaccess%i?PW=%s&State=TOGGLE&\r\n", atoi(argv[3]), argv[4]);
```

**handle** tiene il Socket, con cui avviene la comunicazione con il Web-IO. Se non può essere creato, il programma viene abbandonato.

```
handle = socket(PF_INET, SOCK_STREAM, 0);
if(socket < 0) {
    printf("Socket cannot be created\r\n");
    exit(-1);
}
```

Creazione del punto di collegamento finale e realizzazione della connessione. Se si presenta un errore, il programma viene abbandonato.

```
inet_aton(argv[1], &remote.sin_addr);
remote.sin_family = AF_INET;
remote.sin_port = htons(atoi(argv[2]));
if(connect(handle, (struct sockaddr *) &remote, sizeof(remote)) != 0) {
    printf("Connection cannot be established\r\n");
    exit(-1);
}
```

La stringa di comando precedentemente già creata viene inviata al Web-IO tramite la connessione aperta.

```
if(send(handle, (const void *)command, sizeof(command), 0) == 0) {
    printf("Error at sending data\r\n");
}
```

Chiusura della connessione e abbandono regolamentare del programma.

```
close(handle);
exit(0);
```

In questo modo il programma è completamente esaurito e viene lasciato. Alla prossima esecuzione viene nuovamente avviato e funziona nuovamente come descritto sopra.

## In conclusione

Con semplici mezzi la pure.box può essere utilizzata come supporto di propri programmi C e C++. Mediante l'accesso alla rete e al

sistema di file interno si possono gestire compiti molto complessi.



Saremo lieti di fornirvi una consulenza personalizzata!

Wiesemann & Theis  
GmbH  
Porschestr. 12  
42279 Wuppertal  
Tel.: +49 202/2680-110 (Lun-Ven. 8-17)  
Fax: +49 202/2680-265  
info@wut.de

© Wiesemann & Theis GmbH, con riserva di errori e modifiche: poiché possono verificarsi errori, nessuna nostra informazione deve essere utilizzata senza essere stata verificata. Vi preghiamo di comunicarci tutti gli errori o gli equivoci che avete rilevato in modo tale che possiamo riconoscerli ed eliminarli quanto prima.

[Protezione dei dati](#)