

Pierwszy program na procesor ARM9

Wpisany przez Dominik

niedziela, 30 stycznia 2011 21:39 - Poprawiony niedziela, 30 stycznia 2011 21:55



Niniejszy artykuł przedstawia sposób na uruchomienie pierwszego programu na procesorze ARM9. Procesor użyty w przykładzie to EP9302 produkowany przez firmę Cirrus Logic. Programy są uruchamiane na zestawie uruchomieniowym z zainstalowanym systemem linux:

<http://www.olimex.com/dev/cs-e930x.html>

Skompilowane programy najwygodniej zgrywać na procesor przy użyciu połączenia sieciowego. W tym celu najlepiej połączyć komputer, który posłuży nam do kompilacji programów i zestaw uruchomieniowy z procesorem ARM9 do tej samej sieci lokalnej.

Na początku na komputerze należy zainstalować serwer tftp (ang. Trivial File Transfer Protocol server). Jeżeli pracujemy pod linuxem najwygodniej jest zainstalować dowolny wybrany serwer tftp dostępny w repozytoriach. Dokładny opis można znaleźć tutaj:

<http://malsecure.blogspot.com/2007/10/setup-tftp-server-on-ubuntukubuntu.html>

Dla systemu Windows może to być np.:

http://www.solarwinds.com/products/freetools/free_tftp_server.aspx

Do kompilowania programów pod linuxem najlepiej użyć gcc:

Pierwszy program na procesor ARM9

Wpisany przez Dominik

niedziela, 30 stycznia 2011 21:39 - Poprawiony niedziela, 30 stycznia 2011 21:55

<http://arm.cirrus.com/files/tools/arm-linux-gcc-3.4.3-1.0.1.tar.bz2>

Rozpakowujemy archiwum do naszego systemu plików. Teraz możemy napisać nasz pierwszy program. Dostęp do peryferiów uzyskamy dzięki poleceniu `mmap()`. Tworzymy zatem plik `test.c`:

```
#include<unistd.h>
#include<sys/types.h>
#include<sys/mman.h>
#include<fcntl.h>
#include<stdio.h>

volatile unsigned int *PADR, *PADDR, *PEDR, *PEDDR, *PBDR, *PBDDR, *GPIOBDB,
*GPIOADB; unsigned char *start;
int main(void){

    int fd = open("/dev/mem", O_RDWR);
    start = (unsigned char *)mmap(0, getpagesize(), PROT_READ|PROT_WRITE, MAP_SHARED,
    fd, 0x80840000);
    PADR = (unsigned int *)(start + 0x00); //port A
    PADDR = (unsigned int *)(start + 0x10); //port B direction
    PBDR = (unsigned int *)(start + 0x04); //port B
    PBDDR = (unsigned int *)(start + 0x14); //port B direction
    PEDR = (unsigned int *)(start + 0x20); //port E data
    PEDDR = (unsigned int *)(start + 0x24); //port E direction
    GPIOBDB = (unsigned int *)(start + 0xC4); // debounce on port B
    GPIOADB = (unsigned int *)(start + 0xA8); // debounce on port b
    *PADDR = 0x01; // ustawiamy pin PA0 jako wyjście (może posłużyć do migania dioda)

    *PBDDR = 0x00; //port B jako wejście
    *PEDDR = 0x00; // port E jako wejście

    *PADR = ~0x01; //zapalamy diode
```

Pierwszy program na procesor ARM9

Wpisany przez Dominik

niedziela, 30 stycznia 2011 21:39 - Poprawiony niedziela, 30 stycznia 2011 21:55

```
volatile float a=1;
```

```
for(int i=0;i<50000;i++) a+=0.01;// czekamy chwilkę
```

```
*PADR = 0x01;// gasimy diode
```

```
printf("program finished!") ;//wyswietlamy komunikat w konsoli
```

```
return 1;
```

```
}
```

Teraz musimy skompilować program poleceniem:

```
usr/local/arm/3.4/bin/arm-linux-g++ test.c -o test
```

W wyniku powstanie plik test. Należy go udostępnić przez serwer tftp. Pod linuxem wystarczy skopiować plik *test* do katalogu /tftpboot. Teraz możemy zalogować się na naszą płytę uruchomieniową. Użyjemy protokołu telnet:

```
telnet xx.xx.xx.xx
```

gdzie `xx.xx.xx.xx` to adres IP naszej płyty uruchomieniowej.

Pierwszy program na procesor ARM9

Wpisany przez Dominik

niedziela, 30 stycznia 2011 21:39 - Poprawiony niedziela, 30 stycznia 2011 21:55

Teraz musimy skopiować nasz program na płytę z procesorem ARM:

```
tftp -g -r test yyy.yyy.yyy.yyy
```

gdzie yyy.yyy.yyy.yyy to adres IP naszego komputera, na którym kompilowaliśmy program.

Pozostaje tylko uruchomienie programu. Nadajmy uprawnienia do uruchamiania dla przegranego pliku, najlepiej:

```
chmod 777 test
```

uruchamiamy program:

```
./test
```

i obserwujemy wynik.

Aktualnie system operacyjny i programy są umieszczone w pamięci RAM. Wyłączenie zasilania spowoduje, że program musi być wgrany ponownie. W następnej części przedstawiony zostanie sposób na nagranie systemu operacyjnego i programu do pamięci Flash.