How to use POSIX Threads with Embedded Linux

Some C programs needs additional libraries. A typical sample is the usage of POSIX Threads. The document describes how to find out what libraries are necessary and how to install these libraries within the (A)DNP/1486 file space.

- **1. Step**: Generate a C source code. Make sure to include **pthread.h**. Use the Thread functions. The following code is a sample for a simple program with POSIX Threads.

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>
#include <pthread.h>

void *thread_function (void);

int counter= 0;

int main (void)
{
   int iThread, mirror;
   pthread_t mythread;

   iThread= pthread_create (&mythread, NULL, thread_function, NULL);
   if (iThread != 0) {
      printf ("Can't create thread...\n");
      exit (EXIT_FAILURE);
   }

   printf ("Counter (incremented by Thread)= ");
   mirror= counter;
   while (1) {
      if (mirror != counter) {
         printf ("%04d\b\b\b\b", counter);
         fflush (stdout);
         mirror= counter;
      }
   }
}

void *thread_function (void)
{
   while (1) {
      counter++;
      sleep (1);
   }
}
```

- **2. Step:** Run the Linux/GNU C compiler and build a executable from your C source code file. The following command lines assumes that **pthread1.c** is your C source code file and **pthread1** the name of the executable.

  ```
  gcc -o pthread1 pthread1.c
  ```

- **3. Step:** Check with the **ldd** utility program the names of the dynamic link libraries, which are necessary to run your executable on the (A)DNP/1486.

  ```
  ldd pthread1
  ```

  We assume that your executable needs **libc.so.6**, **libm.so.6** and **libstdc++-libc6.2-2.so.3**. The library **libc.so.6** is already present within the ADNP/1486 root file system (see directory **/lib**).