# How to Access purplebox Device Registers

This note provides information on how to access I/O device registers on the purpleboxes from a QNX program.

To access a device register, there are a couple of initialization steps that you have to perform.

1. You first need to request access rights to the hardware I/O for the thread. This is done through the `ThreadCtl()` function. This will allow you to execute the QNX I/O functions, and attach interrupt handlers.

Note that this is a thread-level permission so you have to do it for every thread that needs access to the device registers. Threads created by a thread which already has I/O access permission will inherit that permission.

Here is an example request for I/O access permission. Note that that function returns -1 on error and sets the `errno` global variable with an error code. The `perror()` function will print out on `STDERR` a message associated with the error code.

```
#include <sys/neutrino.h>

if ( ThreadCtl(_NTO_TCTL_IO, NULL) == -1)
{

    perror("Failed to get I/O access permission");
    return 1;

}
```

2. The second initialization step is to map the I/O ports for the device registers which you need to access into the address space for your program. The Pentium architecture for the processors in the purpleboxes have I/O ports which are in an I/O space separate from memory addresses. You will need to map the I/O port for each device register that you need to access. If you have a number of registers you will work with, consider using a table to hold all the values needed for initialization. If you are working in C++, you could also consider creating an `IOPort` class.

Here is an example of a port mapping.

```
#include <stdint.h>
#include <sys/mman.h>

#define IO_PORT_SIZE 1
#define CTRL_ADDRESS 0x37A

uintptr_t ctrlHandle;

ctrlHandle = mmap_device_io(IO_PORT_SIZE, CTRL_ADDRESS);
if(ctrlHandle == MAP_DEVICE_FAILED)
{

    perror("Failed to map control register");
    return 2;
```

```
        }
```

You now have a handle to the device register which you can use in a call to any of the `in*()` or `out*()` functions that QNX provides.

---

*$Id: AccessingIOPorts.html 32 2009-03-30 14:47:58Z rtembed $*