WEB ENABLED MONITORING AND CONTROL DEVICE

Software System Overview
Kernel Selection

- uC/OS-II Real Time Embedded Kernel
  - 1ms Tick Period ~130KB rom ~18KB ram

The uC/OS-II RTK was chosen for its flexibility and ease of configuration. A primary concern for this project is use by future ETEC students, whom develop a working knowledge of uC/OS-II.
Additional Software Packages

- Micrium Coldfire Networking Driver
- Micrium Networking Stack
- Micrium DHCP Client
- Micrium HTTP Server
Task Designation

- (4) Startup Task
- (5) DHCPC Task
- (6) HTTPS Task
- (7) User IO Task
Startup Task

- Initializes on chip resources
- Initializes networking driver
- Initializes global data
- Creates other tasks
- Task delete

The startup task operates similar to the startup tasks created in lab.
HTTP Server Task

- Communicates with Micrium HTTP server
- Process HTTP connections/sessions
- Uses HTTP hooks to gather sensor data
## Module Overview

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Functionality</th>
</tr>
</thead>
<tbody>
<tr>
<td>main.c</td>
<td>Start Task</td>
</tr>
<tr>
<td></td>
<td>Sets up resources and provides communication with the micrium suite of software.</td>
</tr>
<tr>
<td>mic_net_driver.c</td>
<td>Micrium networking driver.</td>
</tr>
<tr>
<td>gpio.c</td>
<td>Provides gpio input/output routines from reading/writing to PTA/PTC.</td>
</tr>
<tr>
<td>a2d.c</td>
<td>Provides the functionality to read from the a2d converter.</td>
</tr>
<tr>
<td>mic_httpd.c</td>
<td>HTTP Server Task</td>
</tr>
<tr>
<td></td>
<td>Micrium http server implementation</td>
</tr>
<tr>
<td>mic_dhcp.c</td>
<td>DHCPC Task</td>
</tr>
<tr>
<td></td>
<td>Micrium dhcp client implementation</td>
</tr>
<tr>
<td>mic_tcp_driver.c</td>
<td>TCP/IP driver and networking stack</td>
</tr>
</tbody>
</table>
Server / Networking Flowchart
GPIO Module

- `GPIO_read()` – read from a gpio port
- `GPIO_write()` – write to a gpio port
- `GPIO_init()` – initialize the gpio ports
- `GPIO_cddr()` – change the DDR for a gpio port

The gpio module simply provides an interface to the gpio hardware and allows for reading/writing of data to/from the device.
A2D Module

- A2D_read() – read from the A2D device
- A2D_init() – initialize the A2D device

The A2D module allows the A2D converter to be initialized, and subsequently read from. The A2D converter is always used as an input device.
The DHCPC module contains the implementation of micriums DHCP client as well as the implementation of the DHCPC task.

Communicates with the TCP/IP stack (layers 2 and 3) via the DHCP protocol.

DHCP is a protocol designed to support automatic setup of a systems networking information as well as dynamic lease management.
HTTPS Module

- Contains micriums implementation of an embedded web (HTTP) server.
- Supports GET/POST requests.
- Support for the presence/absence of a file system
- HIGHLY configurable
- Designed to communicate with micriums TCP/IP stack.
- Hook based server side processing
Hook Based Processing

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml1-strict.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="en" xml:lang="en">

<body onload="nav_handler('about.html','mainp',null)"

...${PROCESS_HANDLER}...

</body>

</html>
${\text{PROCESS\_HANDLER}}$

- Refers to a hook in the server
- Tells the server to execute a function and substitute ${\ldots}$ with the contents returned by the function.
- Substituted data is then sent to the client (server side processing).
- i.e. client never actually sees the ${\text{PROCESS\_HANDLER}}$ directive
/* index.html file */

#define NOFS_INDEX_HTML_CONTENT {
  0x3c,0x68,0x74,0x6d,0x6c,0x3e,0x0d,0x0a,0x0d,0x0a,0x3c,0x68,0x65, \
  0x61,0x64,0x3e,0x0d,0x0a,0x3c,0x6d,0x65,0x74,0x61,0x20,0x68,0x74, \
  0x74,0x70,0x2d,0x65,0x71,0x75,0x69,0x76,0x22,0x20,0x63,0x6f,0x6e, \
  0x74,0x65,0x6e,0x74,0x22,0x66,0x72,