Energis PDU program flow

[DEBUG] = only prints if DEBUG=1

SYS START

Init process: startup.c

* startup.c/ bool startup\_init(void) //init all GPIO
  + init UART
  + init I2C0 @400kHz
  + init Relay-Board RST pins
  + init I2C1 @400kHz
  + init Display-Board RST pins
  + init LCD @spi1, 62.5MHz
  + init LCD control pins
  + init W5500 @spi0, 40MHz
  + init W5500 control pins
  + init Button pins
  + init ADCs
* startup/ bool core0\_init(void)
  + [DEBUG] i2c\_scan\_bus(i2c0, "I2C0");
  + [DEBUG] i2c\_scan\_bus(i2c1, "I2C1");
  + CAT24C512\_Init(); // just init the I2C bus
  + write\_dummy\_eeprom\_content(); //currently dummy
  + mcp\_display\_init(); //init display-board IO expander, all output low
  + mcp\_relay\_init(); //init relay-board IO expander, all output low
  + mcp\_display\_write\_pin(PWR\_LED, 1); // turn on PWR LED
  + mcp\_display\_write\_pin(FAULT\_LED, 1); //turn on FAULT LED until init finished succesfully
  + ILI9488\_Init(); // Reset the display and set up the screen
  + PDU\_Display\_Init(); //Draw the startup UI
  + If bool core0\_init true:   
    mcp\_display\_write\_pin(FAULT\_LED, 0); //Turn off fault led
* startup/ core1\_init
  + wizchip\_spi\_initialize();
  + wizchip\_cris\_initialize();
  + wizchip\_reset();
  + wizchip\_initialize();
  + wizchip\_check();
  + network\_initialize(g\_net\_info);
  + print\_network\_information(g\_net\_info);
  + PDU\_Display\_UpdateIP(g\_net\_info.ip); //Display IP
  + PDU\_Display\_UpdateConnectionStatus(1); // Connected status on LCD
  + mcp\_display\_write\_pin(ETH\_LED, 1); //Turn on ETH LED if connected
* core1\_task
  + Initialize socket and enter main loop: listen to connection and handle html

Relay switching:

* void mcp\_relay\_write\_pin(uint8\_t pin, uint8\_t value)
* After this the indicators shall be updated:
  + To turn on the indicator LED:   
    void mcp\_display\_write\_pin(uint8\_t pin, uint8\_t value)
  + To set the ON/OFF state on the display:   
    void ILI9488\_DrawText(uint16\_t x, uint16\_t y, const char \*text, uint32\_t color)
  + Function to include all these:   
    void PDU\_Display\_ToggleRelay(uint8\_t channel)

Inside of core1\_task.c

* bool core1\_init(void)
  + memcpy(&g\_net\_info, &netConfig, sizeof(wiz\_NetInfo)); //Load network config from EEPROM
  + wizchip\_spi\_initialize();
  + wizchip\_cris\_initialize();
  + wizchip\_reset();
  + wizchip\_initialize();
  + wizchip\_check();
  + network\_initialize(g\_net\_info);
  + print\_network\_information(g\_net\_info);
  + PDU\_Display\_UpdateIP(g\_net\_info.ip); // Updates the IP ont he LCD
  + PDU\_Display\_UpdateConnectionStatus(1); //Puts „Connected” status on LCD
  + mcp\_display\_write\_pin(ETH\_LED, 1); //Turns on ETH LED on displayboard
  + start a socket

main loop in ENERGIS.c:

* stdio\_init\_all();
* startup\_init
* core0\_init
* core1\_init
* multicore\_launch\_core1(core1\_task);
* while(1) // Currently empy