The DIP204 display

Create new project, from template

Include drivers for CPU – cycle counter, FLASHC, GPIO, INTC, PM, PWM, SPI

```
#include <avr32/io.h>
#include "board.h"
#include "dip204.h"
#include "pm.h"
#include "intc.h"
#include "gpio.h"
#include "spi.h"
#include "delay.h"


int main(void) {

           static const gpio_map_t DIP204_SPI_GPIO_MAP =
{
{DIP204_SPI_SCK_PIN,  DIP204_SPI_SCK_FUNCTION },  // SPI Clock.
{DIP204_SPI_MISO_PIN, DIP204_SPI_MISO_FUNCTION},  // MISO.
{DIP204_SPI_MOSI_PIN, DIP204_SPI_MOSI_FUNCTION},  // MOSI.
{DIP204_SPI_NPCS_PIN, DIP204_SPI_NPCS_FUNCTION}   // Chip Select NPCS.
};

            pm_switch_to_osc0(&AVR32_PM, FOSC0, OSC0_STARTUP);

            Disable_global_interrupt();
            INTC_init_interrupts();
            Enable_global_interrupt();

spi_options_t spiOptions =
{
.reg          = DIP204_SPI_NPCS,
.baudrate     = 1000000,
.bits         = 8,
.spck_delay   = 0,
.trans_delay  = 0,
.stay_act     = 1,
.spi_mode     = 0,
.modfdis      = 1
};


gpio_enable_module(DIP204_SPI_GPIO_MAP,
sizeof(DIP204_SPI_GPIO_MAP) / sizeof(DIP204_SPI_GPIO_MAP[0]));
spi_initMaster(DIP204_SPI, &spiOptions);
spi_selectionMode(DIP204_SPI, 0, 0, 0);
spi_enable(DIP204_SPI);
spi_setupChipReg(DIP204_SPI, &spiOptions, FOSC0);

delay_init( FOSC0 );

           dip204_init(100,1);
           dip204_clear_display();
           dip204_set_cursor_position(1,1);
           dip204_show_cursor();
           dip204_write_string("Text to write");
```

```
          return 0;
}
```

Other nice functions:

```
dip204_write_data(variable);
dip204_printf_string(“%u”, variable);
```

Formatters

formatters can be used when printing strings, try for example:

%d %i	Decimal signed integer.
%o	Octal integer.
%x %X	Hex integer.
%u	Unsigned integer.
%c	Character.
%s	String. See below.
%f	double
%e %E	double.
%g %G	double.
%p	pointer.
%n	Number of characters written by this printf.