Hello,

I am trying to get a basic project working using the CMSIS register definitions. Below is the code I am flashing to my NUCLEO-L432KC. LD3 does not light. My understanding of the demonstration board's pinout is that I should be using pin B3. I am setting the pin high to drive the LED on in push/pull mode per other postings I found (I can't view the board files at the moment).

To build this project, copy the Drivers directory out of the STM32CubeL4 or an empty STM32L432KC project created with STM32CubeMX. Also copy out the files "STM32L432KCUx_FLASH.ld", "startup_stm32l432xx.s", and "Src/system_stm32l4xx.c". You should then be able to compile using Make.

It is my hope that any errors are obvious without testing. Does the default code provided by STM do more initialization than is happening below? What could I possibly be missing? I have tried commenting out my clock setup to see if that was the issue. I was told to step through the code to see if the registers are set and I will try this tonight, but what if everything is set properly (as it is now, I have no reason to believe the register assignments aren't taking)?

Thank you in advance.

```
#include <stm32l432xx.h>

void clock_up(void);
void gpio_up(void);

int
main(void)
{
     clock_up();
     gpio_up();

     GPIOB->BSRR |= GPIO_BSRR_BS3;

     while (1) {
     }
}

void
clock_up(void)
{
     // See reference manual p189 onwards.
     // Enable MSI.
     RCC->CR     |= RCC_CR_MSION;
     // Reset HSEON, CSSON, HSION, and PLLON.
     RCC->CR &= 0xEAF6FFFF;
     // Reset PLLCFGR and set:
     //   * PLLM divisor to 1 (3 bits from 4, 0x0);
     //   * PLLN multiplier to 40 (7 bits from 8, 0x28);
     //   * PLLR divisor to 2 (2 bits from 25, 0x0);
     //   * PLLSRC to MSI;
     // to give an 80MHz clock output (max allowed f_CPU).
     RCC->PLLCFGR = 0x00002801;
     // Reset HYEBYP.
     RCC->CR &= 0xFFFBFFFF;
     // Disable all interrupts.
     RCC->CIER = 0;
     // Reset CFGR - writing to reserved and read-only bits has no effect.
     RCC->CFGR = 0;

     // Turn the PLL on.
     RCC->CR |= RCC_CR_PLLON;
     // Configure clock distribution by setting:
     //   * clock output to disabled;
     //   * APB2 (high speed peripheral bus) prescaler to 1;
     //   * APB1 (low speed peripheral bus) prescaler to 1;
     //   * AHB prescaler to 1;
     //   * SW (system clock) to PLL.
     // SYSCLK is divided by the AHB prescaler before being passed to APB1/2.
     RCC->CFGR = 0x00000003;
}

void
gpio_up(void)
{
     // Reset port B.
     RCC->AHB2RSTR |=   RCC_AHB2RSTR_GPIOBRST;
     RCC->AHB2RSTR &= ~(RCC_AHB2RSTR_GPIOBRST);
     // Distribute clock to port B.
     RCC->AHB2ENR |= RCC_AHB2ENR_GPIOBEN;

     // Set IO mode for port B:
     //   * pin 3 to general IO (2 bits from 6, 0x1).
     // Note this register defaults to 0b11 for most pins.
     GPIOB->MODER &= ~(GPIO_MODER_MODE3_1);
     // Set IO type for port B:
     //   * pin 3 to push/pull.
     GPIOB->OTYPER |= 0;
     // Set IO speed for port B:
     //   * pin 3 to low speed.
     GPIOB->OSPEEDR |= 0;
     // Set IO pull up/pull down configuration for port B:
     //   * pin 3 to no pull up or pull down.
     GPIOB->PUPDR |= 0;
}????????????????????????????????????????????????????????????????????????????????

TARGET      = kb-stm32l432
DEBUG       = 1
BUILD      = ./build

ASSEMBLY = \
     startup_stm32l432xx.s
SOURCE      = \
     $(wildcard *.c)
INCLUDES = \
     -I./Drivers/CMSIS/Include \
     -I./Drivers/CMSIS/Device/ST/STM32L4xx/Include \
```

```
     -I./Drivers/STM32L4xx_HAL_Driver/Inc \
     -I./Drivers/STM32L4xx_HAL_Driver/Inc/Legacy
DEFINES      = -DSTM32L432xx
OPTIONS      = -std=gnu11 -Og
WARNINGS = -Wall -Wextra -pedantic

LDSCRIPT = STM32L432KCUx_FLASH.ld
LIBS     = -lc -lm -lnosys


CROSS_COMPILE = arm-none-eabi-
CC     = $(CROSS_COMPILE)gcc
AS     = $(CROSS_COMPILE)gcc -x assembler-with-cpp
CP     = $(CROSS_COMPILE)objcopy
AR     = $(CROSS_COMPILE)ar
SZ     = $(CROSS_COMPILE)size

CPU     = -mcpu=cortex-m4
FPU = -mfpu=fpv4-sp-d16
FLOAT-ABI = -mfloat-abi=hard
MCU = $(CPU) -mthumb $(FPU) $(FLOAT-ABI)


CFLAGS      = $(MCU) $(DEFINES) $(INCLUDES) $(OPTIONS) -Wall -fdata-sections \
             -ffunction-sections
ASFLAGS      = $(MCU) $(OPTIONS) -Wall -fdata-sections -ffunction-sections
LDFLAGS      = $(MCU) -T$(LDSCRIPT) $(LIBS) -Wl,-Map=$(BUILD)/$(TARGET).map,--cref \
             -Wl,--gc-sections

ifeq ($(DEBUG), 1)
CFLAGS += -g -gdwarf-2
endif

# Generate dependency information.
CFLAGS += -MMD -MP -MF"$(@:%.o=%.d)" -MT"$(@:%.o=%.d)"


# Default target.
all: $(BUILD)/$(TARGET).elf

OBJECT  = $(addprefix $(BUILD)/,$(notdir $(SOURCE:.c=.o)))
vpath %.c $(sort $(dir $(SOURCE)))

OBJECT += $(addprefix $(BUILD)/,$(notdir $(ASSEMBLY:.s=.o)))
vpath %.s $(sort $(dir $(ASSEMBLY)))

$(BUILD)/%.o: %.c Makefile | $(BUILD)
     $(CC) -c $(CFLAGS) -Wa,-a,-ad,-alms=$(BUILD)/$(notdir $(<:.c=.lst)) $< -o $@

$(BUILD)/%.o: %.s Makefile | $(BUILD)
     $(AS) -c $(CFLAGS) $< -o $@

$(BUILD)/$(TARGET).elf: $(OBJECT) Makefile
     $(CC) $(OBJECT) $(LDFLAGS) -o $@
     $(SZ) $@

$(BUILD):
     mkdir $(BUILD)

clean:
     -rm -fR .dep $(BUILD)

-include $(shell mkdir .dep 2>/dev/null) $(wildcard .dep/*)
??????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????????
```