Привет ребята, это первая статься из цикла «**Zephyr ELI5: From a Newbie to Newbies**».   
Тут мы разберем процесс описания новой, кастомной платы, создание .dts Kconfig, board.yaml файлов. Я такоже изучающий как и вы, я хоть и буду стараться следовать лучшим практикам, но не обещаю что на все 100 процентов, потому что я знаю не все. Поэтому профессионалы в zephyr   
 Я дал это статье номер 1, если понадобится написать статью как заставить работать Zephyr SDK и Clion IDE и в принципе установить Zephyr SDK - будет написана статья с номером 0.  
Так же я рекомендую иметь stm32Cube IDE или stm32Cube MX, эти программы имеют очень классный интерфейс для настройки тактирования. Он нам еще не раз пригодится, очень пригодится.   
  
Не которое пояснение перед началом. West – программа сборки zephyr. Нам не понадобится больше знания этого ПО сейчас, но имейте ввиду что нужно потратить время и ознакомиться с ним. Все файлы которые мы должны создать в статье, так же можно скачать с github. В конце статьи будет представлена ссылка на коммит где лежат файлы в том состоянии как будто мы их создали в процессе прочтения статьи.  
  
Итак, вы решили создать свою плату, первое что нужно сделать это выбрать микроконтроллер на основе которого мы создаем плату. Как вы догадались, тут мы будем использовать STM32 серию а именно STM32f411CE, 128 Kb RAM , 512 Kb FLASH, саму схему платы мы можем найти тут (ссылка).  
И первое что мы сделаем – обратимся к официальному Гайду по портированию печатных плат – (https://docs.zephyrproject.org/latest/hardware/porting/board\_porting.html) все расходимся…..  
Но если решили продолжить, в официальном гайде говориться что для начала нам достаточно всего 3-х файлов.   
board.yml  
.dts   
Kconfig   
что-ж у нас нет оснований не доверять. Обратите внимание, что с таким набором файлов мы сможем скомпилировать прошивку не не сможем запусить отладку или прошивку средствами west, но это сейчас и не нужно, мы будем заниматься этим позже.  
Я работаю на Mac, файл zephyr установлены в домашнюю директорию моего пользователя  
/Users/kiro/  
Вот вывод  
  
kiro@Kirills-MacBook-Pro / % ls /Users/kiro/ | grep zeph

zephyr-sdk-0.17.4

zephyrproject  
  
тут мы видим 2 папки zephyr-sdk-0.17.4 и zephyrproject, у вас должно быть приблизительно так же.  
  
Переходим в zephyrproject/zephyr  
  
kiro@Kirills-MacBook-Pro zephyr % cd /Users/kiro/zephyrproject/zephyr/

Внутри мы увидим множество файлов и папок, но нас интересует только папка boards, тут хранятся описания плат которые поставляются вместе со zephyr, описание своей мы создадим тут же просто чтобы дополнительно не говорить west где искать нашу плату.   
  
Переходим внутрь boards.   
cd /Users/kiro/zephyrproject/zephyr/boards/  
  
Тут находятся папки произодителей плат, так как мы создаем свою кастомную плату то выберем для примера папку arm, в этот папке мы будем создавать свою плату. Это не очень хороший путь чтобы вести взрослую разработку, потому что вы смешиваете файлы из репозитория Zephyr с теми что вы создали сами. Но это неплохой путь чтобы просто начать.  
Итак   
перейдем в arm   
cd /Users/kiro/zephyrproject/zephyr/boards/arm   
  
И создадим папку reddit\_board   
mkdir /Users/kiro/zephyrproject/zephyr/boards/arm/reddit\_board  
Наша плата будет называться reddit\_board как и папка для нее. Теперь наполним папку файлами.  
  
Раздел @создание файла Kconfig. @  
Первое что нужно – создать файл Kconfig. С правильным именем. После точки должно следовать имя вашей платы. В данном случае у нас название платы - reddit\_board поэтому создаем файл Kconfig.reddit\_board  
  
команда для этого   
touch /Users/kiro/zephyrproject/zephyr/boards/arm/reddit\_board/Kconfig.reddit\_board  
  
внутри этого файла мы должны указать название микроконтроллера который установлен на нашей плате. Впишите такой текст в файл   
  
config REDDIT\_BOARD

select SOC\_STM32F411XE  
  
Да, название платы капслоком.   
Как узнать, откуда вообще брать это название микроконтроллера SOC\_STM32F411XE.   
Все эти названия берутся из базы поддерживаемых, это просто файлы.   
Откроем файл   
cat /Users/kiro/zephyrproject/zephyr/soc/st/stm32/stm32f4x/Kconfig.soc  
Как мы сидим из пути, все описания микроконтроллеров находятся в папке soc, а далее сгруппированы по производителю (st), семейству (stm32), и линейке (stm32f4x), таким образом вы сможете найти любой интересующий вас микроконтроллер.   
  
Вернемся к файлу   
/Users/kiro/zephyrproject/zephyr/soc/st/stm32/stm32f4x/Kconfig.soc  
Отсюда SOC\_STM32F411XE мы и берем. Все варианты перечислены в самом конце в строка в виде   
default "stm32f411xe" if SOC\_STM32F411XE  
Тоесть тут мы видим все варианты названий SOC для серии STM32F4.  
На этом редактирование Kconfig.reddit\_board закончено.  
  
Раздел создание @ board.yml@ файла  
  
В этом файле мы описываем метаданные высокого уровня, название микроконтроллера, название платы.   
Впишем туда такую такой текст   
  
  
board:

name: reddit\_board

full\_name: STM32F411ce reddit\_board

vendor: st

socs:

- name: stm32f411xe  
  
Тут мы вновь вписали название процессора, имя доски (name: reddit\_board) должно совпадать с именем файла Kconfig.reddit\_board , именем .dts файла. Совпадение с именем директории в которой находятся файлы - не обязательно, в одной директории могут быть описаны даже 2 и более плат, но в данном примере у нас одна плата в директории, и название директории и платы совпадает, чтобы упростить задачу.  
В поле full\_name: STM32F411ce reddit\_board просто впишем какое то более человекопонятное название. Завершаем редактирование файла board.yml.   
  
  
Секция @редактирование reddit\_board.dts @  
dts файлы построены таким образом чтобы идти от общего к частному, тоесть файлы образуют цепочку наследования на каждом слое добавляя какие то спецефичные для этой платформы блоки.  
Если мы откроем файл   
/Users/kiro/zephyrproject/zephyr/dts/arm/st/f4/stm32f411Xe.dtsi и пройдем по кажому #include то увидим такую цепочку наследования   
stm32f411Xe.dtsi -> stm32f411.dtsi -> stm32f401.dtsi -> stm32f4.dtsi -> armv7-m.dtsi -> skeleton.dtsi  
  
Тоесть на каждом уровне добавляется описаение переферии которое специфично только этому конкретному семейству микроконтроллеров, и с каждым уровнем добавляются все более и более специфичные блоки, то-есть мы идем от общего к частному. Если мы импортируем файл stm32f411Xe.dtsi то мы получим доступ ко всей переферии которая была описана в наследуемых файлах, ее нужно просто включить и в некоторых случаях настроить. Так же мы добавим свою переферии которая уже специфична для нашей конкретной доски, например микросхему флеш памяти w25q128 которая в свою очередь подключена к интерфейсу SPI микроконтроллера.   
Подробнее о том как создать свой DTS файл , описано тут <https://docs.zephyrproject.org/latest/build/dts/intro.html#devicetree-intro> .  
Сейчас же мы создадим простой файл который позволит нам скомпилировать пример для нашего микроконтроллера.  
touch /Users/kiro/zephyrproject/zephyr/boards/arm/reddit\_board/reddit\_board.dts  
Теперь откроем файл текстовым редактором и добавим такое содержимое.

/dts-v1/; // Device Tree Source version 1 — required header for DTS files

#include <st/f4/stm32f401Xe.dtsi> // Include the SoC-specific base definitions for STM32F401xE

#include <st/f4/stm32f401c(d-e)ux-pinctrl.dtsi> // Include pin control definitions for STM32F401C(D/E)Ux series

#include <zephyr/dt-bindings/input/input-event-codes.h> // Include input event key codes (e.g., KEY\_0)

/ {

model = "Reddir v1 Board"; // Human-readable model name of the board

compatible = "st,reddit\_board"; // Compatible string used for matching in drivers or overlays

chosen {

zephyr,console = &usart1; // Set USART1 as the system console (e.g., printk/log output)

zephyr,shell-uart = &usart1; // Use USART1 for shell interface (if enabled)

zephyr,sram = &sram0; // Define main SRAM region

zephyr,flash = &flash0; // Define main flash region

};

leds {

compatible = "gpio-leds"; // Node for GPIO-controlled LEDs

user\_led: led { // Define a label for the user LED

gpios = <&gpioc 13 GPIO\_ACTIVE\_LOW>; // LED is on GPIOC pin 13, active low

label = "User LED"; // Human-readable label for the LED

};

};

gpio\_keys {

compatible = "gpio-keys"; // Node for GPIO button inputs (key events)

user\_button: button { // Define a label for the user button

label = "KEY"; // Human-readable label

gpios = <&gpioa 0 (GPIO\_ACTIVE\_LOW | GPIO\_PULL\_UP)>; // Button on GPIOA pin 0, active low with pull-up

zephyr,code = <INPUT\_KEY\_0>; // Logical input code, like KEY\_0

};

};

aliases {

led0 = &user\_led; // Alias 'led0' used by Zephyr subsystems (e.g., samples)

sw0 = &user\_button; // Alias 'sw0' used for button handling (e.g., in samples or input drivers)

};

};

&usart1 {

pinctrl-0 = <&usart1\_tx\_pa9 &usart1\_rx\_pa10>; // Define the TX/RX pins for USART1: TX = PA9, RX = PA10

pinctrl-names = "default"; // Define pinctrl configuration name (required)

status = "okay"; // Enable this peripheral in the build

current-speed = <115200>; // Set UART baudrate to 115200

};

&clk\_hse {

clock-frequency = <DT\_FREQ\_M(25)>; // Use external crystal with 25 MHz frequency

status = "okay"; // Enable HSE (High-Speed External) oscillator

};

&pll {

div-m = <25>; // PLL input divider

mul-n = <200>; // PLL multiplier

div-p = <2>; // PLL output divider for main system clock

div-q = <7>; // PLL output divider for main system clock

clocks = <&clk\_hse>; // PLL source ( in this exaple - use external oscillator (HSE))

status = "okay"; // Enable PLL

};

&rcc {

clocks = <&pll>; // Use PLL as system clock source

clock-frequency = <DT\_FREQ\_M(100)>; // Final core system clock frequency after applying PLL: 100 MHz

ahb-prescaler = <1>; // Division on AHB bus

apb1-prescaler = <2>; // Divide APB1 clock by 2 (max 50 MHz for STM32F4)

apb2-prescaler = <1>; // Division on APB2

};

Мы создали все необходимые файлы, теперь если ввести команду   
west boards мы должны увидеть нашу плату reddit\_board  
  
Убедитесть что вы вошли в виртуальное окружение, это делается такой командой (для моего расположения файлов), вам необходимо ввести правильное расположение до файла activate.   
source /Users/kiro/zephyrproject/.venv/bin/activate   
  
Дале введите команду   
  
west boards | grep reddit  
  
вы должны получит такой вывод   
reddit\_board  
  
Это значит что west нашел конфигурацию для нашей платы.   
Мы подошли к моменту создания последнего файла. Он не является необходимым но будет удобен для того чтобы проверить работоспособность платы.   
Создайте файл reddit\_board\_defconfig  
Я делаю это такой командой  
touch /Users/kiro/zephyrproject/zephyr/boards/arm/reddit\_board/reddit\_board\_defconfig  
  
Откройте этот файл и впишите туда такое содержимое   
  
# SPDX-License-Identifier: Apache-2.0

# Enable MPU

CONFIG\_ARM\_MPU=y

# Enable HW stack protection

CONFIG\_HW\_STACK\_PROTECTION=y

# Serial Drivers

CONFIG\_SERIAL=y

CONFIG\_UART\_INTERRUPT\_DRIVEN=y

CONFIG\_CONSOLE=y

CONFIG\_UART\_CONSOLE=y

# Enable GPIO

CONFIG\_GPIO=y

CONFIG\_SHELL=y

CONFIG\_KERNEL\_SHELL=y

CONFIG\_SHELL\_BACKEND\_SERIAL=y  
  
  
Это файл c опциями по умолчанию которые применяются к проекту когда вы выбираете эту плату для компиляции с проектом, это те же опции которые вписываются в файл proj.conf внутри проекта. Файл reddit\_board\_defconfig удобно использовать для опций которые специфичны для вашей конткретной платы.   
  
Теперь можно запустить компиляцию проекта из примеров.   
Перейдите в папку с тестовым проектом  
cd /Users/kiro/zephyrproject/zephyr/samples/basic/blinky  
  
Запустите компиляцию   
  
west build -p always -b reddit\_board  
  
После того как компиляция завершится, вы сможете найти .hex файл с прошивкой тут.  
  
/Users/kiro/zephyrproject/zephyr/samples/basic/blinky/build/zephyr/zephyr.hex  
  
Прошейте этот файл в stm32f401ceu6 с помощью отладчика или с помощью DFU режима через USB порт. Светодиод должен мограть раз в секунду, на UART1 (tx\_pa9, rx\_pa10) будет доступна консоль которая работает на скорости 115200.   
  
Типичные ошибки.   
  
(defined at boards/arm/reddit\_board/Kconfig.reddit\_board:4) defined without a type  
  
Проверьте что конструкция select SOC\_....... соответствует существующему SOC, в нашем случае SOC\_STM32F401XE  
  
Или проверте что config BOARD\_REDDIT\_BOARD начинается со слова BOARD\_ и корректность названия платы.