ORS Vaje 4

Delo z GPIO – knjižnica STM

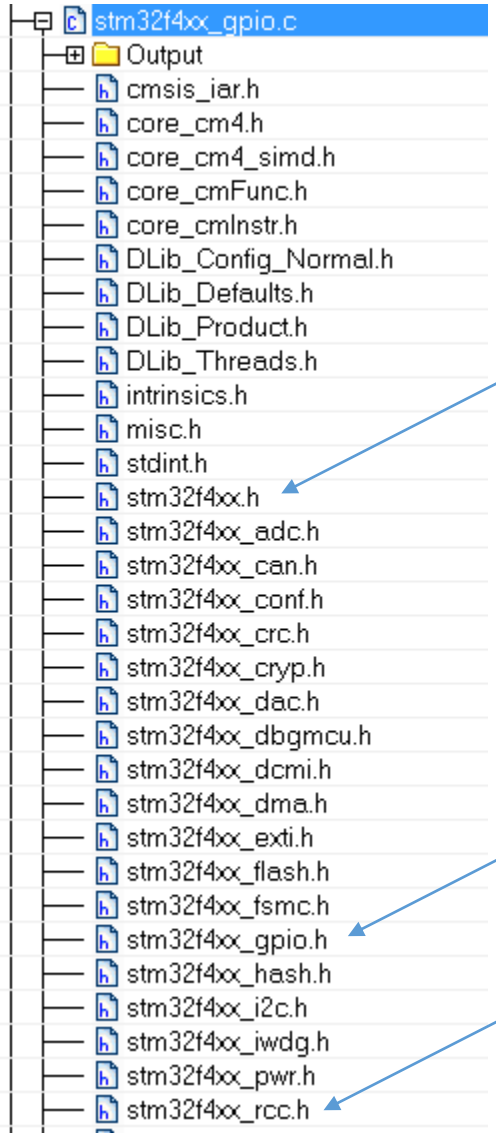
Delo z napravami

- Če želimo inicializirati ali delati s poljubno napravo v sistemu imamo več možnosti
  - Vedno delamo neposredno z biti na podlagi dokumentacije
  - Naredimo si knjižnico (naloga iz prejšnje vaje), ki jo uporabljamo
  - Uporabimo pripravljeno knjižnico od proizvajalca

Delo z napravami

- Če želimo inicializirati ali delati s poljubno napravo v sistemu imamo več možnosti
  - Vedno delamo neposredno z biti na podlagi dokumentacije
  - Naredimo si knjižnico (naloga iz prejšnje vaje), ki jo uporabljamo
  - **Uporabimo pripravljeno knjižnico od proizvajalca**

Lokacija knjižnice – strukture &konstante



Strukture in naslovi vseh naprav v sistemu

Konstante za GPIO

Konstante za RCC

stm32f4xx.h

```
typedef struct
{
  __IO uint32_t MODER;    /*!< GPIO port mode register,               Address offset: 0x00      */
  __IO uint32_t OTYPER;   /*!< GPIO port output type register,        Address offset: 0x04      */
  __IO uint32_t OSPEEDR;  /*!< GPIO port output speed register,       Address offset: 0x08      */
  __IO uint32_t PUPDR;    /*!< GPIO port pull-up/pull-down register,  Address offset: 0x0C      */
  __IO uint32_t IDR;      /*!< GPIO port input data register,         Address offset: 0x10      */
  __IO uint32_t ODR;      /*!< GPIO port output data register,        Address offset: 0x14      */
  __IO uint16_t BSRRL;    /*!< GPIO port bit set/reset low register,  Address offset: 0x18      */
  __IO uint16_t BSRRH;    /*!< GPIO port bit set/reset high register, Address offset: 0x1A      */
  __IO uint32_t LCKR;     /*!< GPIO port configuration lock register, Address offset: 0x1C      */
  __IO uint32_t AFR[2];   /*!< GPIO alternate function registers,     Address offset: 0x20-0x24 */
} GPIO_TypeDef;
```

stm32f4xx.h

```
#define GPIOA                ((GPIO_TypeDef *) GPIOA_BASE)
#define GPIOB                ((GPIO_TypeDef *) GPIOB_BASE)
#define GPIOC                ((GPIO_TypeDef *) GPIOC_BASE)
#define GPIOD                ((GPIO_TypeDef *) GPIOD_BASE)
#define GPIOE                ((GPIO_TypeDef *) GPIOE_BASE)
#define GPIOF                ((GPIO_TypeDef *) GPIOF_BASE)
#define GPIOG                ((GPIO_TypeDef *) GPIOG_BASE)
#define GPIOH                ((GPIO_TypeDef *) GPIOH_BASE)
#define GPIOI                ((GPIO_TypeDef *) GPIOI_BASE)

#define GPIOA_BASE             (AHB1PERIPH_BASE + 0x0000)
#define GPIOB_BASE             (AHB1PERIPH_BASE + 0x0400)
#define GPIOC_BASE             (AHB1PERIPH_BASE + 0x0800)
#define GPIOD_BASE             (AHB1PERIPH_BASE + 0x0C00)
#define GPIOE_BASE             (AHB1PERIPH_BASE + 0x1000)
#define GPIOF_BASE             (AHB1PERIPH_BASE + 0x1400)
#define GPIOG_BASE             (AHB1PERIPH_BASE + 0x1800)
#define GPIOH_BASE             (AHB1PERIPH_BASE + 0x1C00)
#define GPIOI_BASE             (AHB1PERIPH_BASE + 0x2000)

#define AHB1PERIPH_BASE        (PERIPH_BASE + 0x00020000)

#define PERIPH_BASE            ((uint32_t)0x40000000)
```

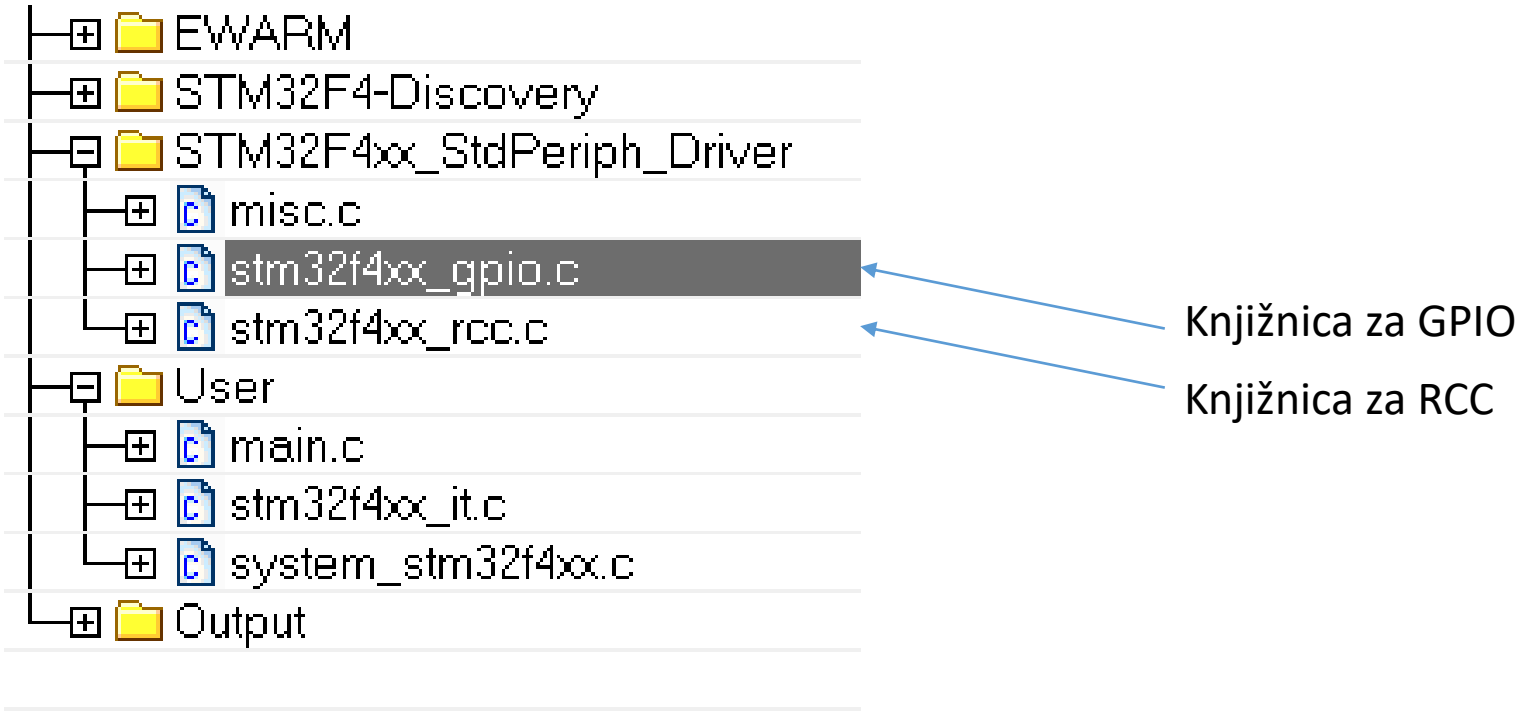
stm32f4xx_gpio.h

```
typedef enum
{
  GPIO_Mode_IN   = 0x00, /*!< GPIO Input Mode */
  GPIO_Mode_OUT  = 0x01, /*!< GPIO Output Mode */
  GPIO_Mode_AF   = 0x02, /*!< GPIO Alternate function Mode */
  GPIO_Mode_AN   = 0x03  /*!< GPIO Analog Mode */
}GPIOMode_TypeDef;
```

```
typedef enum
{
  GPIO_PuPd_NOPULL = 0x00,
  GPIO_PuPd_UP     = 0x01,
  GPIO_PuPd_DOWN   = 0x02
}GPIOPuPd_TypeDef;
```

...

Lokacija knjižnice - funkcije



stm32f4xx_gpio.c

- inicializacijska funkcija

```
void GPIO_Init(GPIO_TypeDef* GPIOx, GPIO_InitTypeDef*
GPIO_InitStruct);
```

- nastavljanje izhoda

```
GPIO_SetBits(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);

GPIO_ResetBits(GPIO_TypeDef* GPIOx,uint16_t GPIO_Pin);

GPIO_ToggleBits(GPIO_TypeDef* GPIOx,uint16_t GPIO_Pin);
```

- branje vhoda

```
uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx,
uint16_t GPIO_Pin);
```

Inicializacija

```
void GPIO_Init(GPIO_TypeDef* GPIOx,
GPIO_InitTypeDef* GPIO_InitStruct);
```

- Inicializacijska struktura
  - V funkciji se pretvori/prepiše v strukturo naprave

```
typedef struct
{
  uint32_t GPIO_Pin;              /*!< Specifies the GPIO pins to be configured.
                                       This parameter can be any value of @ref GPIO_pins_define */

  GPIOMode_TypeDef GPIO_Mode;     /*!< Specifies the operating mode for the selected pins.
                                       This parameter can be a value of @ref GPIOMode_TypeDef */

  GPIOSpeed_TypeDef GPIO_Speed;   /*!< Specifies the speed for the selected pins.
                                       This parameter can be a value of @ref GPIOSpeed_TypeDef */

  GPIOOType_TypeDef GPIO_OType;   /*!< Specifies the operating output type for the selected pins.
                                       This parameter can be a value of @ref GPIOOType_TypeDef */

  GPIOPuPd_TypeDef GPIO_PuPd;     /*!< Specifies the operating Pull-up/Pull down for the selected pins.
                                       This parameter can be a value of @ref GPIOPuPd_TypeDef */
}GPIO_InitTypeDef;
```

GPIO_Pin

- Možne vrednosti

```
/** @defgroup GPIO_pins_define
  * @{
  */
#define GPIO_Pin_0                 ((uint16_t)0x0001)  /* Pin 0 selected */
#define GPIO_Pin_1                 ((uint16_t)0x0002)  /* Pin 1 selected */
#define GPIO_Pin_2                 ((uint16_t)0x0004)  /* Pin 2 selected */
#define GPIO_Pin_3                 ((uint16_t)0x0008)  /* Pin 3 selected */
#define GPIO_Pin_4                 ((uint16_t)0x0010)  /* Pin 4 selected */
#define GPIO_Pin_5                 ((uint16_t)0x0020)  /* Pin 5 selected */
#define GPIO_Pin_6                 ((uint16_t)0x0040)  /* Pin 6 selected */
#define GPIO_Pin_7                 ((uint16_t)0x0080)  /* Pin 7 selected */
#define GPIO_Pin_8                 ((uint16_t)0x0100)  /* Pin 8 selected */
#define GPIO_Pin_9                 ((uint16_t)0x0200)  /* Pin 9 selected */
#define GPIO_Pin_10                ((uint16_t)0x0400)  /* Pin 10 selected */
#define GPIO_Pin_11                ((uint16_t)0x0800)  /* Pin 11 selected */
#define GPIO_Pin_12                ((uint16_t)0x1000)  /* Pin 12 selected */
#define GPIO_Pin_13                ((uint16_t)0x2000)  /* Pin 13 selected */
#define GPIO_Pin_14                ((uint16_t)0x4000)  /* Pin 14 selected */
#define GPIO_Pin_15                ((uint16_t)0x8000)  /* Pin 15 selected */
#define GPIO_Pin_All               ((uint16_t)0xFFFF)  /* All pins selected */
```

GPIO_Mode

```
/**
  * @brief  GPIO Configuration Mode enumeration
  */
typedef enum
{
  GPIO_Mode_IN   = 0x00, /*!< GPIO Input Mode */
  GPIO_Mode_OUT  = 0x01, /*!< GPIO Output Mode */
  GPIO_Mode_AF   = 0x02, /*!< GPIO Alternate function Mode */
  GPIO_Mode_AN   = 0x03  /*!< GPIO Analog Mode */
}GPIOMode_TypeDef;
```

GPIO_Speed

```
/**
  * @brief  GPIO Output Maximum frequency enumeration
  */
typedef enum
{
  GPIO_Speed_2MHz   = 0x00, /*!< Low speed */
  GPIO_Speed_25MHz  = 0x01, /*!< Medium speed */
  GPIO_Speed_50MHz  = 0x02, /*!< Fast speed */
  GPIO_Speed_100MHz = 0x03  /*!< High speed on 30 pF (80 MHz Output max speed on 15 pF) */
}GPIOSpeed TypeDef;
```

GPIO_OType

```
/**
  * @brief  GPIO Output type enumeration
  */
typedef enum
{
  GPIO_OType_PP = 0x00,
  GPIO_OType_OD = 0x01
}GPIOOType_TypeDef;
```

GPIO_PuPd

```
/**
  * @brief  GPIO Configuration PullUp PullDown enumeration
  */
typedef enum
{
  GPIO_PuPd_NOPULL = 0x00,
  GPIO_PuPd_UP     = 0x01,
  GPIO_PuPd_DOWN   = 0x02
}GPIOPuPd_TypeDef;
```

stm32f4xx_rcc.c & stm32f4xx_rcc.h

```
void RCC_AHB1PeriphClockCmd(uint32_t RCC_AHB1Periph, FunctionalState NewState);
```

```
typedef enum {DISABLE = 0, ENABLE = !DISABLE} FunctionalState;
```

```
#define RCC_AHB1Periph_GPIOA             ((uint32_t)0x00000001)
#define RCC_AHB1Periph_GPIOB             ((uint32_t)0x00000002)
#define RCC_AHB1Periph_GPIOC             ((uint32_t)0x00000004)
#define RCC_AHB1Periph_GPIOD             ((uint32_t)0x00000008)
#define RCC_AHB1Periph_GPIOE             ((uint32_t)0x00000010)
#define RCC_AHB1Periph_GPIOF             ((uint32_t)0x00000020)
#define RCC_AHB1Periph_GPIOG             ((uint32_t)0x00000040)
#define RCC_AHB1Periph_GPIOH             ((uint32_t)0x00000080)
#define RCC_AHB1Periph_GPIOI             ((uint32_t)0x00000100)
```

Naloga

- Ponovite nalogo iz prejšnjih tednov z uporabo STMove knjižnice