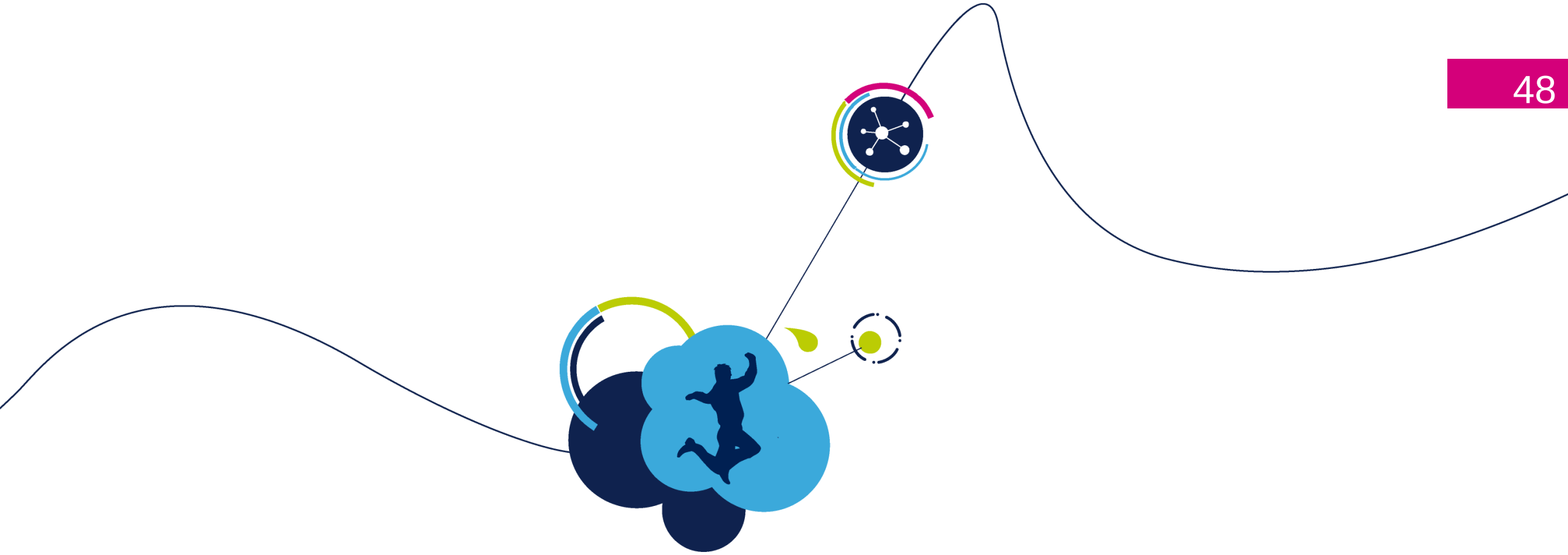


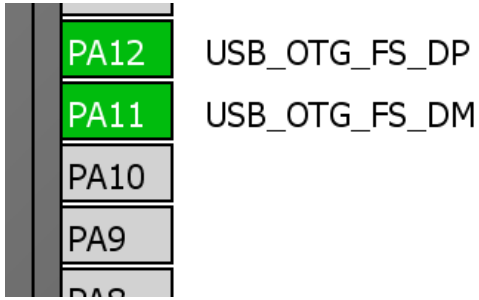
USB HID Device lab

USB HID Device lab

- HID device communicate over interrupt endpoint which guarantee the delivery in finite time
- In CubeMX library is implemented the mouse functionality as default example
- For change of the functionality for example to keyboard you need to modify report descriptor
- No need for additional driver for common HID devices on windows PCs

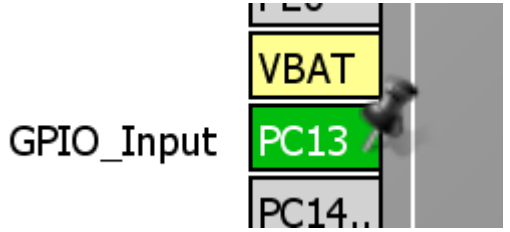
USB HID Device lab

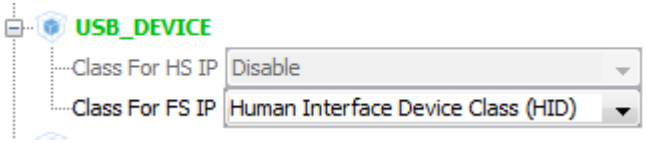
- Create project in CubeMX, configuration is the same like for VCP device
  - Menu > File > New Project
  - Select STM32F4 > STM32F446 > LQFP144 > STM32F446ZETx
- Select USB FS OTG in device mode
- Select HSE clock
  - (Bypass HSE from STlink)



USB HID Device lab

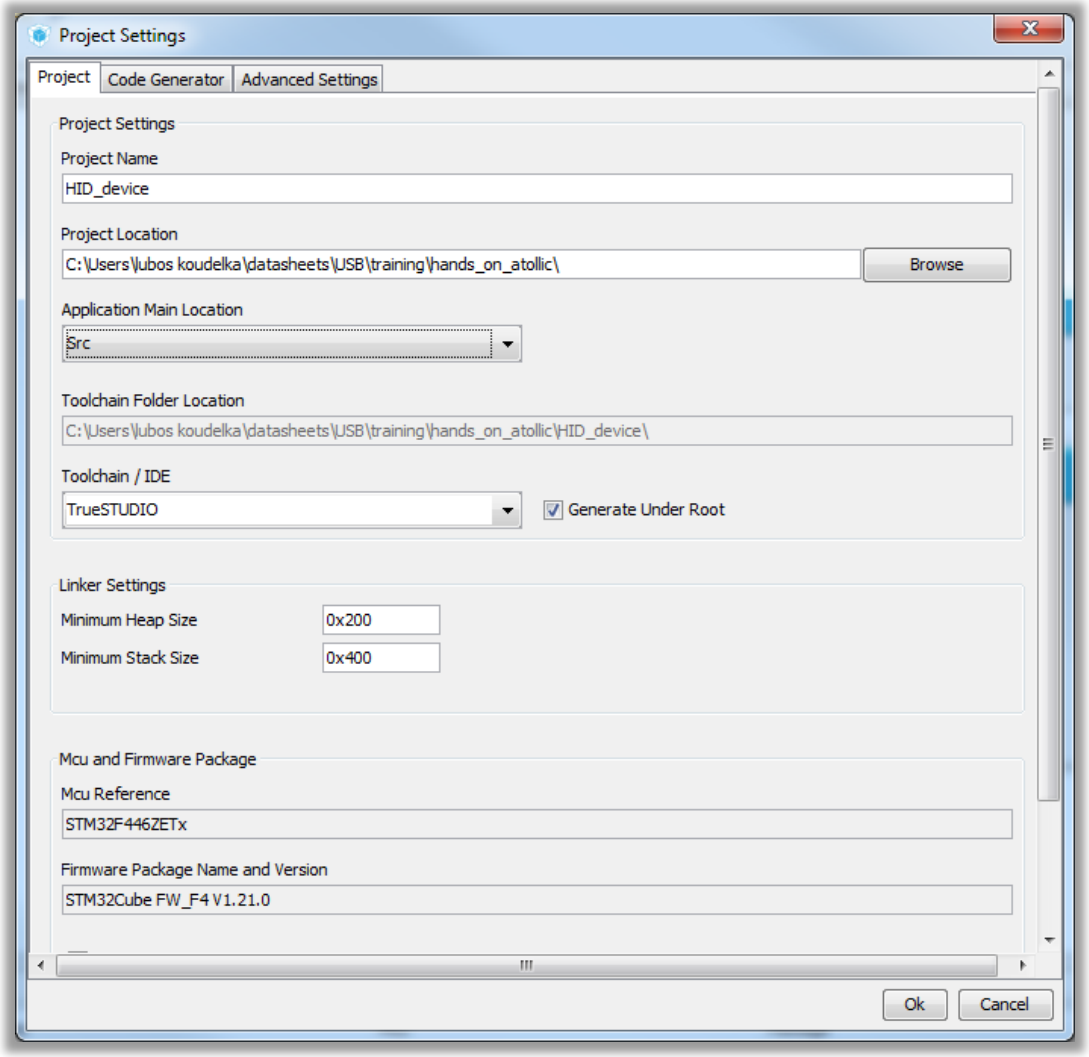
- Configure PC13 as input – key button
- Select HID class in MiddleWares
- Configure RCC clocks
  - Set 8 MHz HSE as PLL input
  - Set HCLK frequency 168 MHz
  - PLL parameters will be computed automatically





USB HID Device lab

- Now we set the project details for generation
  - Menu > Project > Project Settings
  - Set the project name
  - Project location
  - Type of toolchain
- Linker Settings
  - For HID class default Heap size is sufficient
- Now we can Generate Code
  - Menu > Project > Generate Code



life.augmented

USB HID Device lab

- Format of messages sent by HID device is defined in REPORT descriptor
- This format have only basic rules but descriptor for one device can look very different even when functionality will be same
- Handling and parsing descriptors is on host side
- Descriptor generated by CubeMX PC expects in this format:

USB HID Device lab

- If you want to change format of this message you need to change the REPORT DESCRIPTOR in file usbd_hid.c the report descriptor array is called HID_MOUSE_ReportDesc
  - There is a HID parser available on usb.org
  - Example settings for other configurations in HID specification or other online sources can be used

USB HID Device lab

- We will work only in main.c
- First include hid header file

```
/* USER CODE BEGIN Includes */
#include "usbd_hid.h"
/* USER CODE END Includes */
```

- Define buffer which will be send to the host

```
/* USER CODE BEGIN PFP */
uint8_t buffer[4];
/* USER CODE END PFP */
```

USB HID Device lab

- USBD_HID_SendReport function will send the buffer on button press
- buffer contains data about the mouse move and state of buttons
- With this settings every, button press move with cursor

```
/* USER CODE BEGIN 2 */
 buffer[0]=0;//buttons - first 3 bits(LSB)
 buffer[1]=100;//X axis 8bit value signed
 buffer[2]=0;//Y axis 8bit value signed
 buffer[3]=0;//Wheel 8bit value signed
 /* USER CODE END 2 */

 /* USER CODE BEGIN WHILE */
 while (1)
 {
 /* USER CODE END WHILE */
 /* USER CODE BEGIN 3 */
   if(HAL_GPIO_ReadPin(GPIOC,GPIO_PIN_13)==GPIO_PIN_SET){
     USBD_HID_SendReport(&hUsbDeviceFS,buffer,4);
     HAL_Delay(100);
   }

 }
```