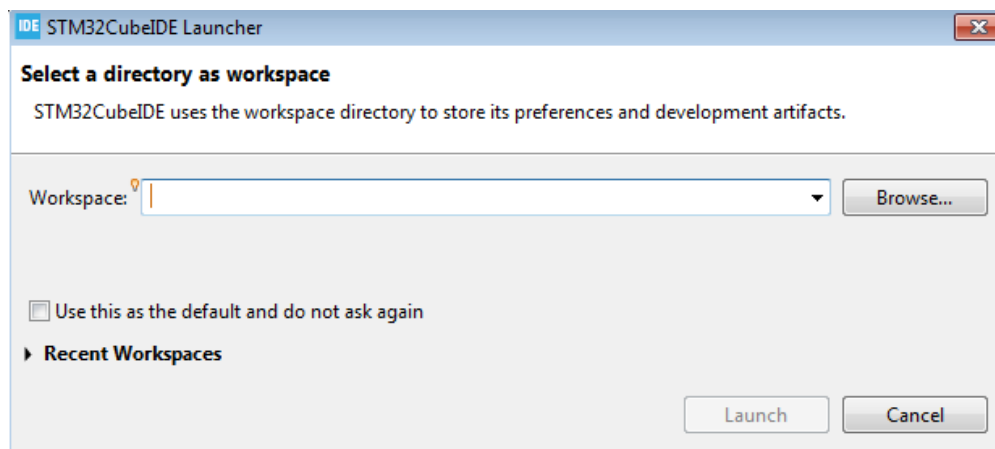
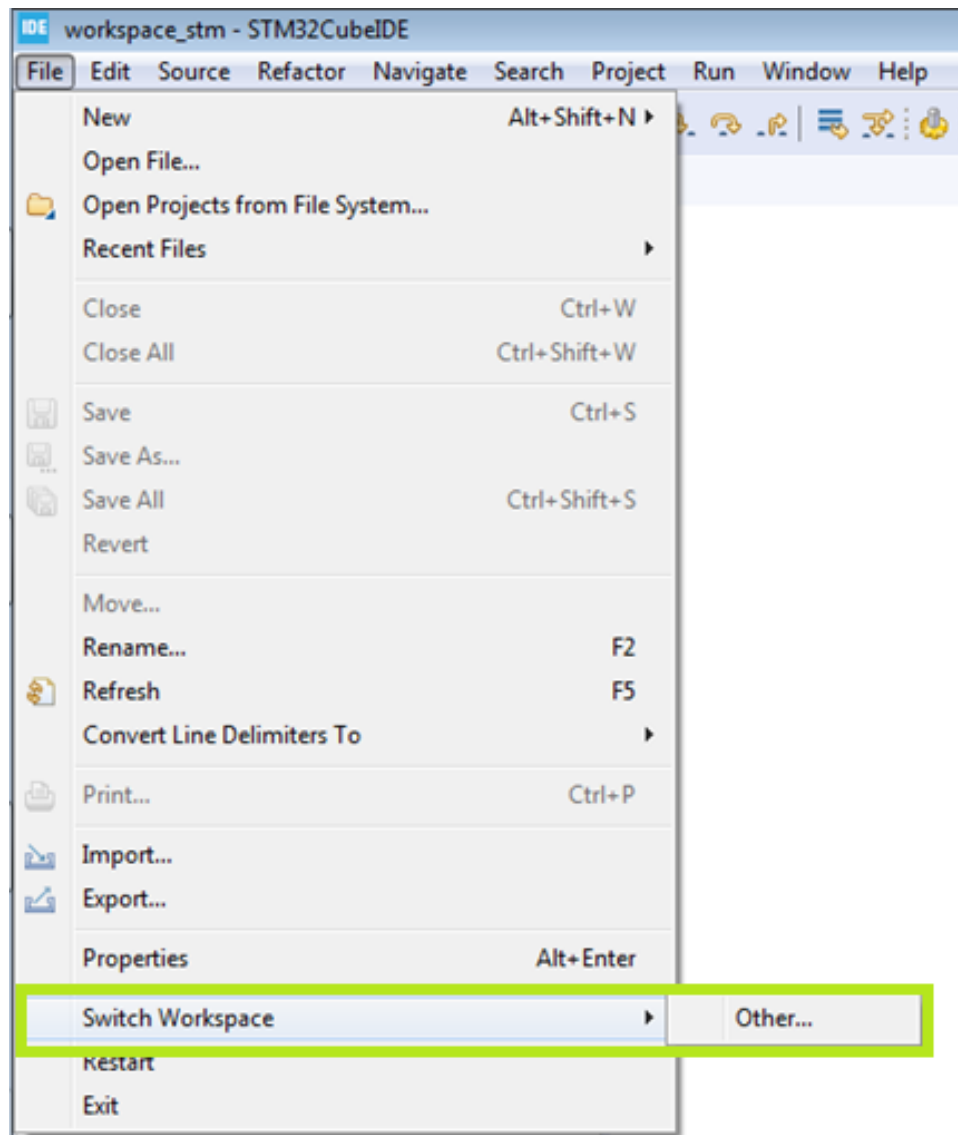


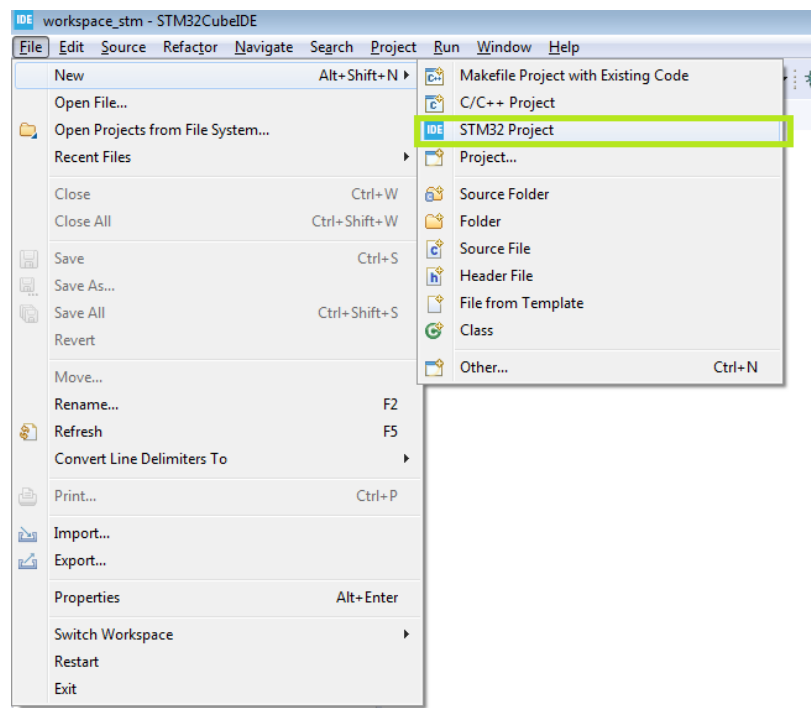
Первоначально необходимо для среды разработки создать папку-рабочую область, в которой будут сохраняться все проекты. При каждом запуске будет появляться следующее окно:



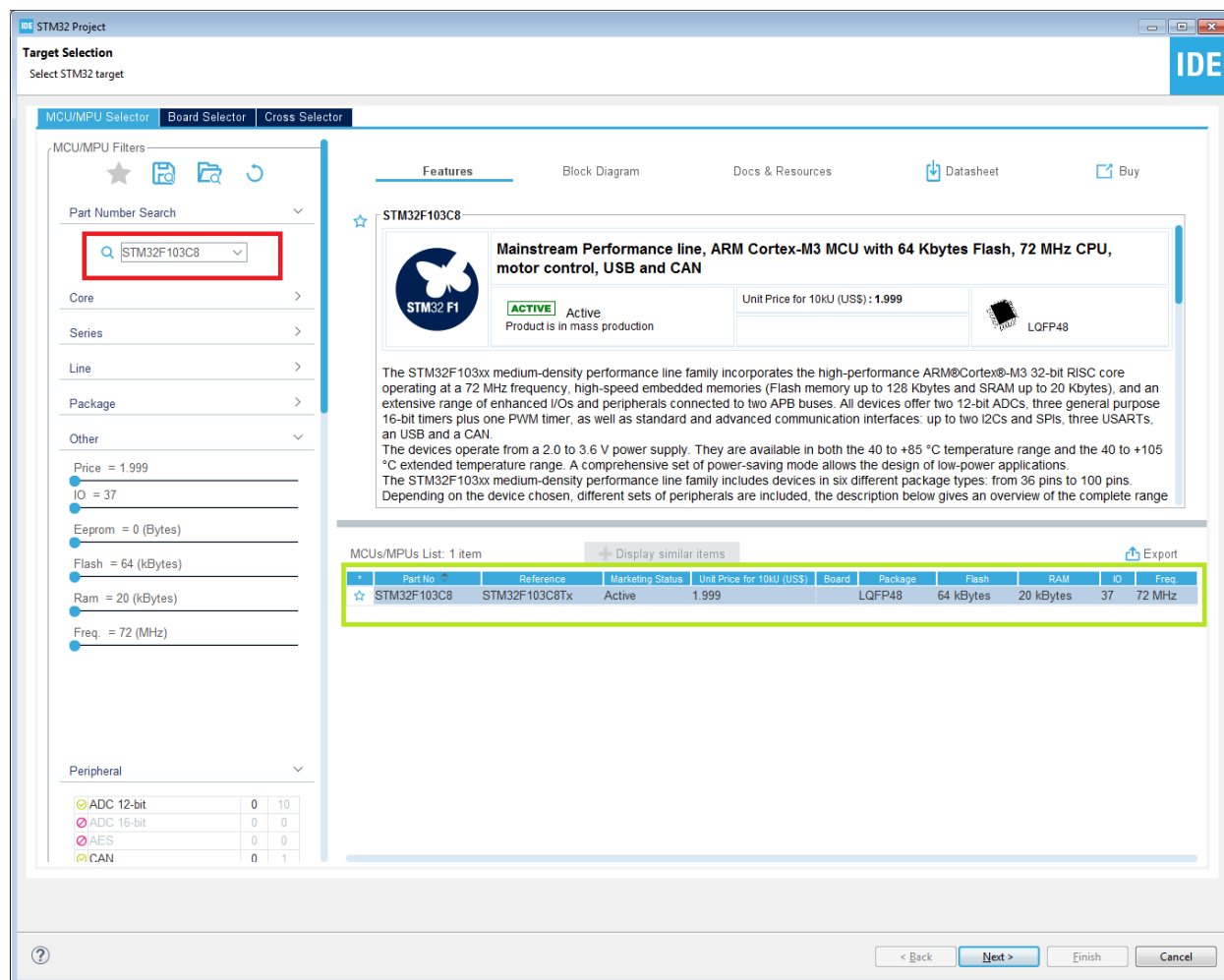
Можно нажать на «**Use this as the default and do not ask again**», если уверены, что по умолчанию будете работать в созданной папке. При необходимости рабочее пространство можно переключать:



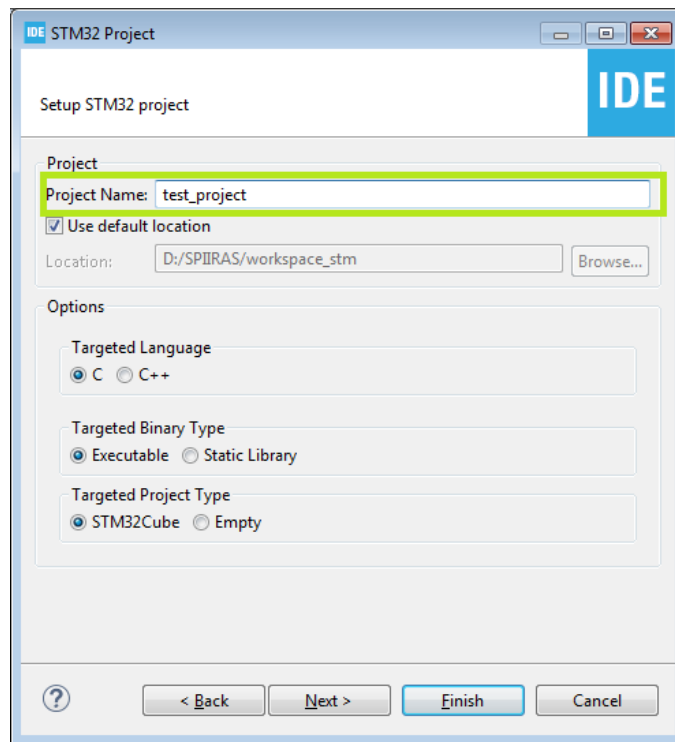
Подготовительный этап. Сначала следует создать проект.



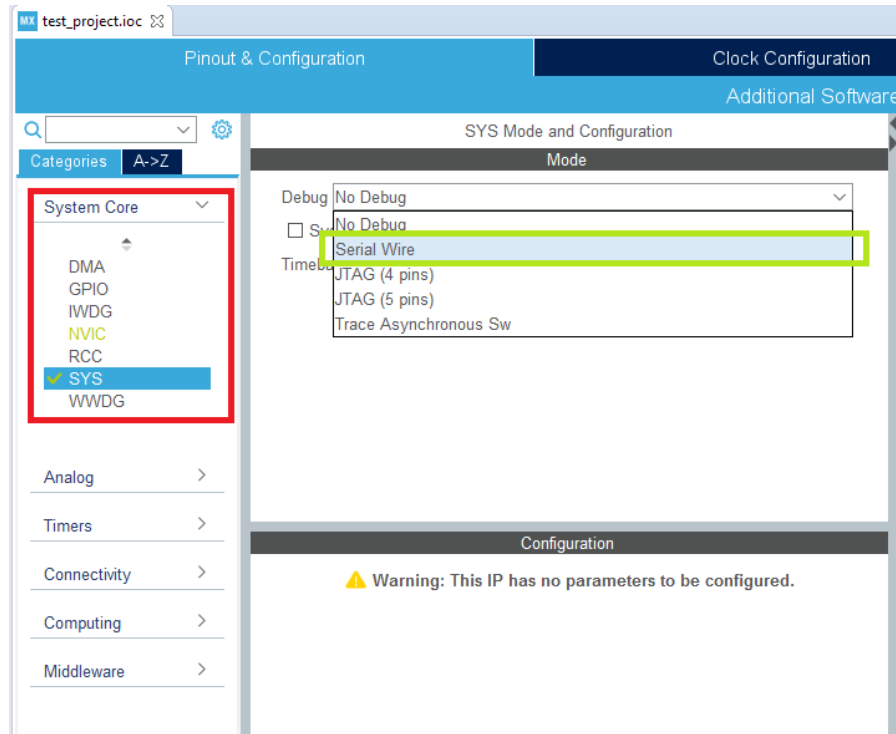
Затем в поле поиска вводите наименование микроконтроллера (выделено **красным**), выбираете из списка предлагаемых изделий в правом окне проверяемый микроконтроллер (выделено **зеленым**).



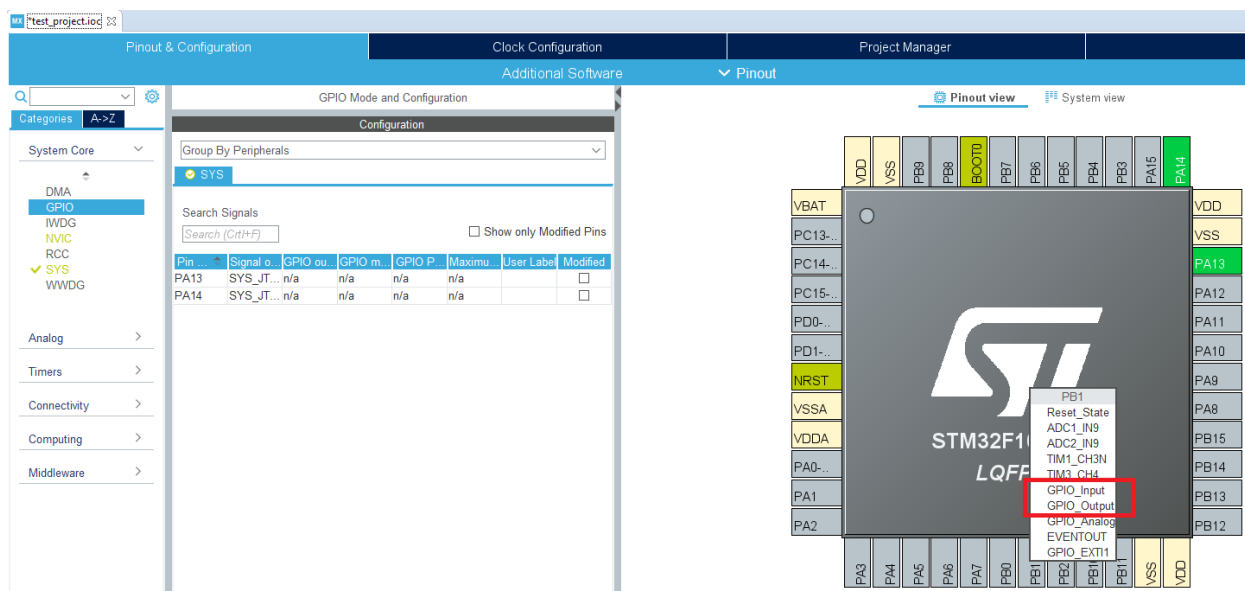
Далее задайте имя проекту (папка проекта будет носить это название, в ней будет весь код и конфигурации). Остальные параметры можно задать по умолчанию и перейти по кнопке **Finish** к окончательному созданию проекта. В процессе создания проекта могут загружаться некоторые библиотеки и файлы, следует немного подождать.



Приступайте к следующему этапу – настройка проекта. **В первую очередь**, если вы используете программатор ST-LINK V2/2.1, включите параметр **Serial Wire** режима **Debug** во вкладке **System Core**/SYS. Без этого МК прошиваться не будет, также не будет работать отладка.

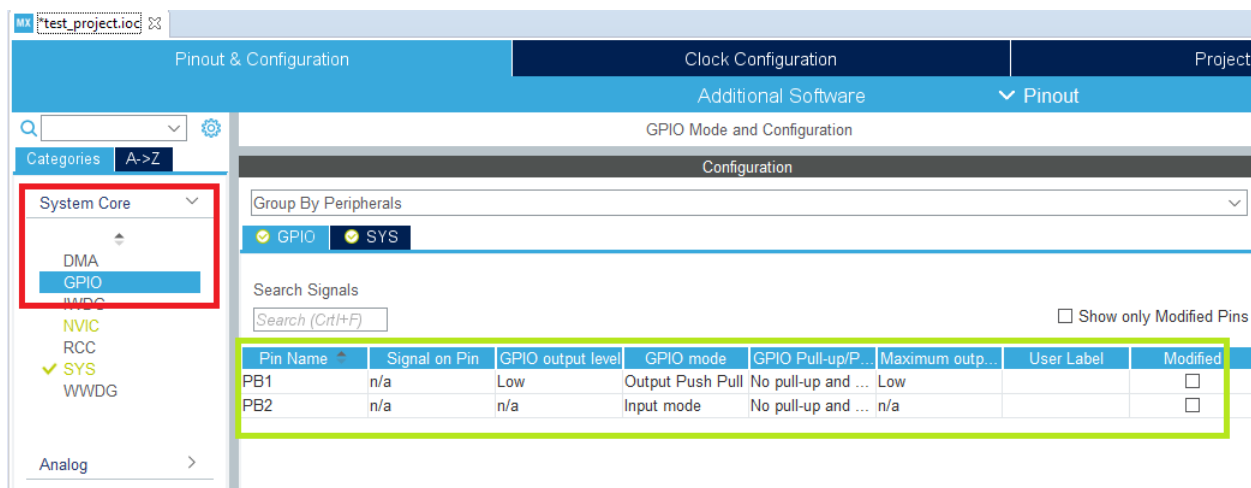


Теперь можно приступить к настройке выводов. Чтобы настроить цифровой **ввод/вывод**, достаточно кликнуть ЛКМ в правом окне с изображением МК на его вывод и выбрать **GPIO\_Input/GPIO\_OUTPUT**.



! Важно знать, что вносимые изменения следует в данном окне (файл **test\_project.ioc**) сохранять в первую очередь, поскольку это основа – программа состоит из нескольких частей, в которой настройка происходит в ПО для генерации кода. В итоге будут создаваться конфигурационные строки кода в **main.c**.

В данном окне достаточно инструментов для тестов. Настроим выбранный вывод **PB1** на цифровой **выход**, а соседний – **PB2**, на цифровой **вход**. Данные настройки отображаются в разделе **System Core/GPIO**.



## Окно настройки выхода PB1:

Configuration

Group By Peripherals

GPIO

SYS

Search Signals

Search (Ctrl+F)

Show only Modified Pins

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/P...	Maximum outp...	User Label	Modified
PB1	n/a	Low	Output Push Pull	No pull-up and ...	Low		<input type="checkbox"/>
PB2	n/a	n/a	Input mode	No pull-up and ...	n/a		<input type="checkbox"/>

PB1 Configuration :

GPIO output level

Low

GPIO mode

Output Push Pull

GPIO Pull-up/Pull-down

No pull-up and no pull-down

Maximum output speed

Low

User Label

где

**GPIO output level:** Low/High. Можно назначить, чтобы изначально выходное напряжение было либо низкого, либо высокого уровня.

**GPIO mode:** Output Push Pull/Output Open Drain. Первый работает как обычно (0 или 1 на выход), второй подходит при больших нагрузках и напряжении до 5 В.

**GPIO Pull-up/Pull-down:** No pull-up and no pull-down/Pull up/Pull down. Режим подтяжки выхода — без подтяжки, к напряжению питания 3.3 В, либо к общему проводу (минус/земля), соответственно.

**Maximum output speed:** Low/Medium/High. Скорость работы вывода.

**User Label:** метка для данного вывода, чтобы обращаться к нему в программе.

## Окно настройки входа PB2:

The screenshot shows the 'Configuration' window with 'Group By Peripherals' selected. The 'GPIO' tab is active. A search bar is present with the text 'Search (Ctrl+F)'. A table lists pins PB1 and PB2. Below the table, the 'PB2 Configuration' section is highlighted with a green border, showing 'GPIO mode' set to 'Input mode', 'GPIO Pull-up/Pull-down' set to 'No pull-up and no pull-down', and an empty 'User Label' field.

Pin Name	Signal on Pin	GPIO output level	GPIO mode	GPIO Pull-up/P...	Maximum outp...	User Label	Modified
PB1	n/a	Low	Output Push Pull	No pull-up and ...	Low		<input type="checkbox"/>
PB2	n/a	n/a	Input mode	No pull-up and ...	n/a		<input type="checkbox"/>

Search Signals  
Search (Ctrl+F)

☐ Show only Modified Pins

Pin Name Signal on Pin GPIO output level GPIO mode GPIO Pull-up/P... Maximum outp... User Label Modified

PB1 n/a Low Output Push Pull No pull-up and ... Low ☐

PB2 n/a n/a Input mode No pull-up and ... n/a ☐

PB2 Configuration :

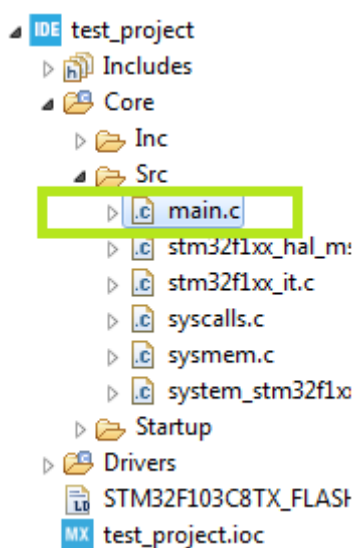
GPIO mode Input mode

GPIO Pull-up/Pull-down No pull-up and no pull-down

User Label

Здесь настроек меньше, они схожи с настройками для цифрового выхода, однако отличается лишь **GPIO mode** – там установлено Input mode.

Если возникнет необходимость писать программу с изменением состояния выхода и чтения входа, то в дереве проекта левого окна открывайте файл **test\_project/Core/Src/main.c**.



В этом файле ищите основной цикл **while(1)**. Пишите **только между** обозначенными комментариями **/\*USER CODE BEGIN\*/** и **/\*USER CODE END\*/**, иначе при следующих изменениях в проекте и кодогенерации то, что будет написано вне этих блоков, **сотрется**.

```

79
80  /* Configure the system clock */
81  SystemClock_Config();
82
83  /* USER CODE BEGIN SysInit */
84
85  /* USER CODE END SysInit */
86
87  /* Initialize all configured peripherals */
88  MX_GPIO_Init();
89  /* USER CODE BEGIN 2 */
90
91  /* USER CODE END 2 */
92
93  /* Infinite loop */
94  /* USER CODE BEGIN WHILE */
95  //ТУТ ПИСАТЬ МОЖНО
96  while (1)
97  {
98      //ТУТ ПИСАТЬ МОЖНО
99      /* USER CODE END WHILE */
100
101      /* USER CODE BEGIN 3 */
102  }
103  /* USER CODE END 3 */
104 }

```

Используемые для обозначенных задач функции:

1. Задержка:

`HAL_Delay(t);` // t – аргумент, пишется значение в миллисекундах

2. Задание низкого (**GPIO\_PIN\_RESET**) или высокого (**GPIO\_PIN\_SET**) уровня напряжения:

`HAL_GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin, GPIO_PinState PinState);` // первый аргумент – порт A, B и т.д.; второй – номер пина; третий – уровень напряжения

3. Переключение состояния вывода.

`HAL_GPIO_TogglePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);` // схожие аргументы

4. Чтение состояния вывода, настроенного на вход:

`HAL_GPIO_ReadPin(GPIO_TypeDef *GPIOx, uint16_t GPIO_Pin);` // аргументы схожи с предыдущими функциями

Пример (если на входе PB2 высокий уровень, то высокий уровень будет на выходе PB1, в противном случае – низкий уровень):



```

/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1) {
    if (HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_2)) { // если 1, то
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_SET);
    } else { // иначе
        HAL_GPIO_WritePin(GPIOB, GPIO_PIN_1, GPIO_PIN_RESET);
    }
/* USER CODE END WHILE */

/* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */

```