

Содержание

Генератор трафика CAN на основе STM32	3
Базовый минимум	3
Подготовка	3
Подключение программатора	3
Сборка и прошивка	3
Описание работы программы	4
Отправка	4
Приём	4
Вывод ошибок	5

Генератор трафика CAN на основе STM32

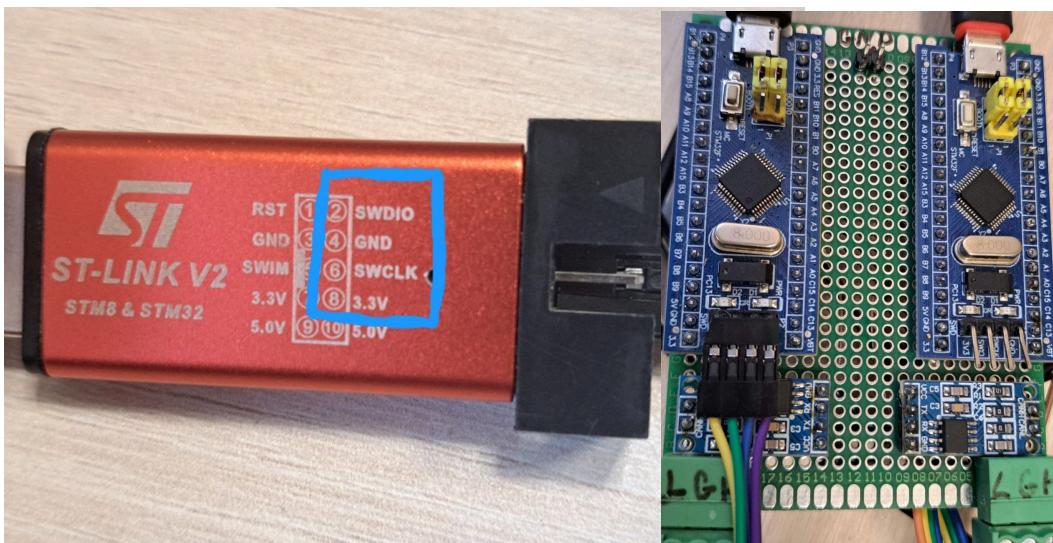
Базовый минимум

- STM32CubeIDE
- Программатор для STM32 st-link v2
- Проект ([ссылка на репозиторий](#))

Подготовка

Подключение программатора

Необходимо соединить пины SWDIO, GND, SWCLK и 3.3V с соответствующими пинами на микроконтроллере.



Сборка и прошивка

В STM32CubeIDE открыть проект.

Для сборки и прошивки используется конфигурация CAN_STm32F103C6T6_Debug

workspace_2.0.0 - CAN_STm32F103C6T6

File Edit Source Refactor Navigate Search Project Run Window Help

Project Explorer CAN_STm32F103C6T6

- Binaries
- Includes
- Core
 - Inc
 - Src
 - main.c
 - stm32f1xx_hal_msp.c
 - stm32f1xx_it.c
 - syscalls.c
 - sysmem.c
 - system_stm32f1xx.c
 - Startup
- Drivers
- Debug
 - Core
 - Drivers
- CAN_STm32F103C6T6.elf - [arm/le]

Run CAN_STm32F103C6T6.elf

```

1 /* USER CODE BEGIN Header */
2 /**
3  * @file           : main.c
4  * @brief          : Main program body
5  * @attention
6  * 
7  * Copyright (c) 2025 STMicroelectronics.
8  * All rights reserved.
9  * 
10 * This software is licensed under terms that can
11 * be found in the root directory of this software component.
12 * If no LICENSE file comes with this software, it
13 * is provided AS-IS.
14 * 
15 */
16 /**
17 */
18 /* USER CODE END Header */
19 /* Includes -----
20 #include "main.h"
21 
22 /* Private includes -----
23 /* USER CODE BEGIN Includes */
24 
25 /* USER CODE END Includes */
26

```

Нажать, подождать, готово!

Описание работы программы

Отправка

Отправка сообщений происходит в бесконечном цикле, находящемся в функции `main`.

Генерация сообщений происходит следующим образом:

- Очищается буфер с данными.
- При помощи стандартной функции `rand` генерируется число `n` - количество шестнадцатеричных (4-битных) чисел, после чего вычисляется `b` - количество байт данных, которое будет отправлено (`TxHeader.DLC`).
- Далее генерируется `n` 4-битных чисел и заполняется буфер данных.
- По алгоритму CRC-11 вычисляется заголовок (`TxHeader.StdId`), по которому можно проверить целостность данных при приёме.
- Происходит отправка сообщения.

Приём

Приём происходит в функции `HAL_CAN_RxFifo0MsgPendingCallback`.

По полученным данным вычисляется идентификатор (по тому же алгоритму CRC-11), который сравнивается с заголовком.

Вывод ошибок

Во время работы программы считается общее количество отправленных/принятых сообщений, а также считаются ошибки.

Каждый раз при возникновении ошибки (будь то несоответствие чексуммы данных, или при аппаратных ошибках) все эти данные отправляются по UART.

Для данного проекта используются следующие пины на микроконтроллере:

- A9 → TX
- A10 → RX

Для подключения из консоли можно использовать следующую команду:

```
picocom -b 115200 <port>
```

Так выглядит вывод ошибок:

```
RX_CNT: 9199    TX_CNT: 9200
RX_ERRORS: 10    TX_ERRORS: 0      HARD_ERRORS:0

RX_CNT: 583258   TX_CNT: 586978
RX_ERRORS: 10    TX_ERRORS: 0      HARD_ERRORS:1

RX_CNT: 787558   TX_CNT: 791280
RX_ERRORS: 10    TX_ERRORS: 0      HARD_ERRORS:2

RX_CNT: 2006815  TX_CNT: 2010547
RX_ERRORS: 10    TX_ERRORS: 0      HARD_ERRORS:3

RX_CNT: 2513522  TX_CNT: 2517258
RX_ERRORS: 11    TX_ERRORS: 0      HARD_ERRORS:3

RX_CNT: 2514501  TX_CNT: 2518236
RX_ERRORS: 12    TX_ERRORS: 0      HARD_ERRORS:3

RX_CNT: 2515047  TX_CNT: 2518782
RX_ERRORS: 13    TX_ERRORS: 0      HARD_ERRORS:3
```

- RX_CNT и TX_CNT - счётчики принятых и отправленных сообщений.
- RX_ERRORS и TX_ERRORS - счётчики ошибок при принятии и отправке сообщений.
- HARD_ERRORS - аппаратные ошибки.