

Современные 32-разрядные ARM-микроконтроллеры серии STM32.

Преобразователь аналоговых сигналов управления в цифровые

Олег Вальпа (sandh@narod.ru)

В статье описан преобразователь аналоговых сигналов управления от 0 до 10 В в цифровые с широтно-импульсной модуляцией на основе 32-разрядного ARM-микроконтроллера серии STM32 от компании STMicroelectronics и приведены примеры программ, обеспечивающих работу устройства.

ВВЕДЕНИЕ

Довольно часто в современных системах автоматического управления возникает необходимость использования различных типов интерфейсов. Для одних исполнительных устройств, например, необходим управляющий аналоговый сигнал от 0 до 10 В или токовый выход от 0 до 20 мА, для других – сигнал с широтно-импульсной модуляцией (ШИМ). В управляющем устройстве количество интерфейсов ограничено, что

может приводить к нехватке того или иного их типа. Одним из решений данной проблемы является преобразователь интерфейсов управления. Назначение описанного в статье преобразователя заключается в получении управляющего интерфейса ШИМ из аналогового интерфейса с сигналами от 0 до 10 В.

АППАРАТНАЯ ЧАСТЬ

Преобразователь выполнен на недорогом микроконтроллере STM32 [1]. Для повышения функциональности устройство сделано 4-канальным, т.е. позволяет одновременно преобразовывать 4 входных аналоговых сигнала от 0 до 10 В в выходные сигналы ШИМ. Устройство построено на отладочной плате STM32VLDISCOVERY [2], имеющей в своем составе микроконтроллер STM32F100RBT6В в 64-выводном корпусе LQFP с 128 Кбайт флэш-

памяти и 8 Кбайт оперативной памяти. Кроме того, на плате имеются встроенный программатор с отладчиком по интерфейсу USB, несколько светодиодов и пользовательская кнопка. Большинство выводов микроконтроллера подключены к штырьковым соединителям по периметру платы и имеют маркировку. Внешний вид отладочной платы приведен на рисунке 1.

Для построения преобразователя интерфейсов воспользуемся встроенным в микроконтроллер аналого-цифровым преобразователем и широтно-импульсным модулятором. Благодаря многоканальности этих блоков можно реализовать несколько преобразователей интерфейсов с помощью одного микроконтроллера.

Поскольку микроконтроллер питается от 3,3 В, его 12-разрядный аналого-цифровой преобразователь имеет ограничение входного сигнала по амплитуде 3,3 В. В связи с этим входные сигналы с амплитудой от 0 до 10 В необходимо подключать к входам АЦП микроконтроллера через резистивные делители или переменные резисторы. Можно дополнительно подключить к входам АЦП кера-

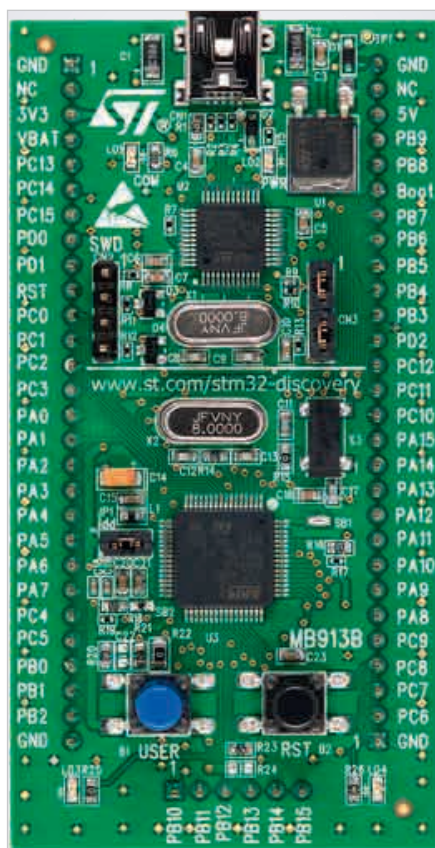
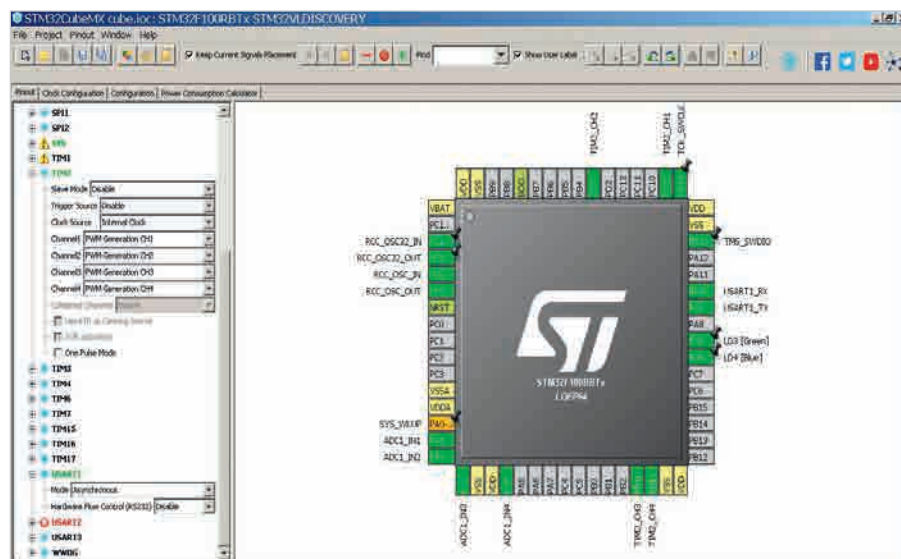


Рис. 1. Внешний вид отладочной платы STM32VLDISCOVERY



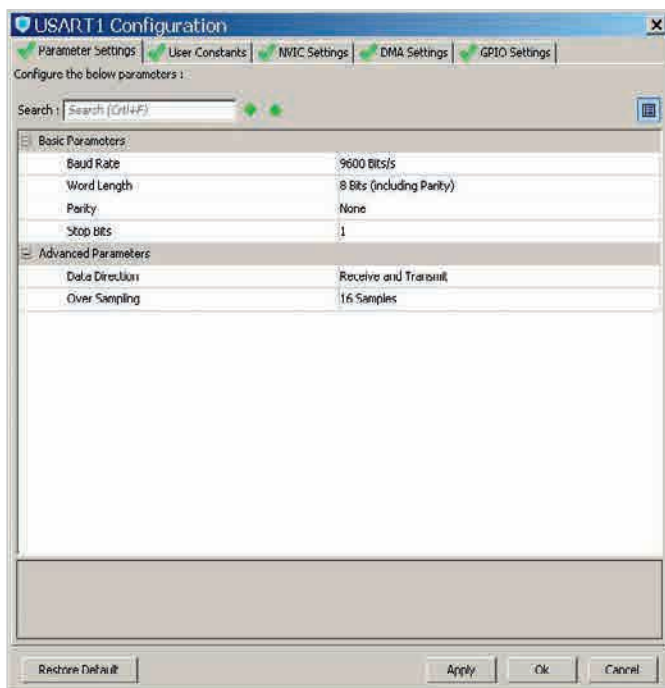


Рис. 3. Окно USART1 генератора кода STM32CubeMX

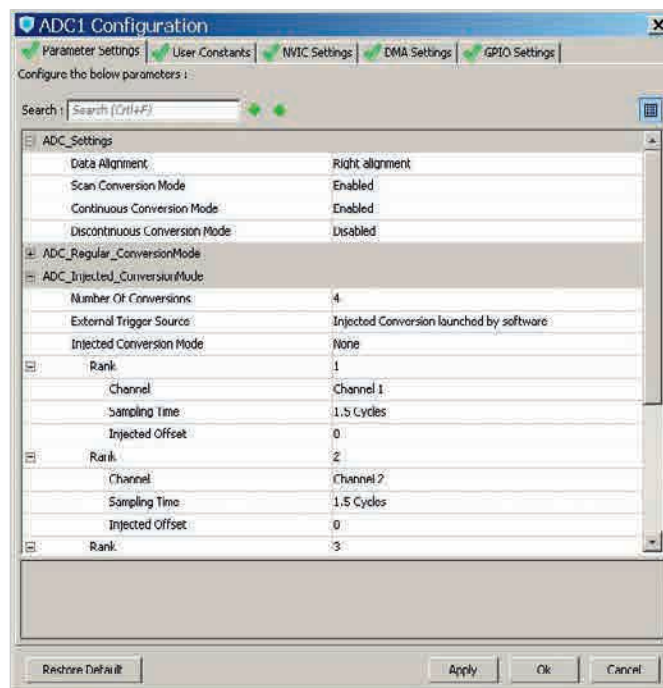


Рис. 4. Окно ADC1 генератора кода STM32CubeMX

мические конденсаторы ёмкостью от 0,1 до 1 мкФ для фильтрации помех. Делители осуществляют преобразование напряжения входных сигналов в напряжение рабочего диапазона АЦП микроконтроллера от 0 до 3,3 В. Нормированные и отфильтрованные сигналы поступают на входы АЦП микроконтроллера. С помощью внутренней программы микроконтроллер будет последовательно опрашивать входы АЦП и преобразовывать их цифровые значения в параметры скважности сигналов ШИМ, формируемых внутренними таймерами микроконтроллера. Выходные сигналы ШИМ необходимо подключить через ограничительные резисторы номиналом 1 кОм к базе выходных транзисторов, работающих в режиме ключей. Таким образом, при изменении аналоговых сигналов на входах преобразователя на его выходах будут формироваться сигналы ШИМ в виде открытого или закрытого состояния транзисторов.

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

Для назначения входных и выходных сигналов отладочной платы STM32VLDISCOVERY использовался свободно распространяемый графический генератор кода STM32CubeMX от компании STMicroelectronics [2]. После установки и запуска данного программного обеспечения необходимо создать новый проект и выбрать во вкладке *Board Selector* отладочную

плату STM32VLDISCOVERY. При этом генератор кода отобразит на экране монитора внешний вид микроконтроллера отладочной платы с подключёнными к нему интерфейсами отладчика, светодиодами и другими цепями.

Подключение необходимых блоков для преобразователя интерфейсов осуществляется во вкладке *Pinout*. Входы АЦП подключаются с помощью установки галочек в полях *IN1...IN4* раздела *ADC Peripherals*. Выходы ШИМ подключаются путём выбора режимов *PWM Generator CH1...CH4* в полях *Channel1...Channel4* раздела *TIM2 Peripherals*. Кроме того, можно подключить в асинхронном режиме последовательный интерфейс в поле *USART1 Peripherals*. Это позволит использовать его в дальнейшем для отладки программы и настройки каналов преобразования интерфейсов. В результате вышеуказанных действий генератор кода STM32CubeMX автоматически распределит подключённые интерфейсы по портам микроконтроллера, как показано на рисунке 2.

После этого необходимо задать режимы работы подключённых интерфейсов и некоторые их параметры. Для этого следует во вкладке *Configuration* генератора кода STM32CubeMX раскрыть нужный блок. Для блока интерфейса USART1 необходимо установить скорость обмена 9600 бод, длину слова 8 бит, отсутствие паритета и 1 стоп-бит (см. рис. 3).

Для блока интерфейса ADC1 необходимо установить 4 инжекторных преобразователя Rank1...Rank4 и назначить для них каналы Cannel1... Cannel4 (см. рис. 4).

Для блока TIM2 требуется задать период 4096, соответствующий 12-разрядному АЦП, и разрешить автоматическую перезагрузку таймера (см. рис. 5).

Теперь необходимо сгенерировать код программы. Для этого следует выбрать в меню *Project* пункт *Generate Code*, затем в открывшемся окне указать среду разработки Keil MDK-ARM V4 [3] и ввести имя проекта. После генерации кода в каталоге проекта будет автоматически создан раздел *MDK-ARM* и *Src* с файлами программы и библиотек, после чего генератор предложит открыть полученный проект в среде разработки Keil MDK-ARM V4.

В данном случае все дальнейшие операции будут связаны только с одним сформированным файлом проекта *main.c*, который является главным модулем и уже содержит все необходимые функции и настройки программы. Этот файл необходимо дополнить строками программы, описывающими алгоритм функционирования преобразователя. Места для строк программы пользователя выделены в файле *main.c* специальным образом с помощью строк `/* USER CODE BEGIN ...*/` и `/* USER CODE END ...*/`.

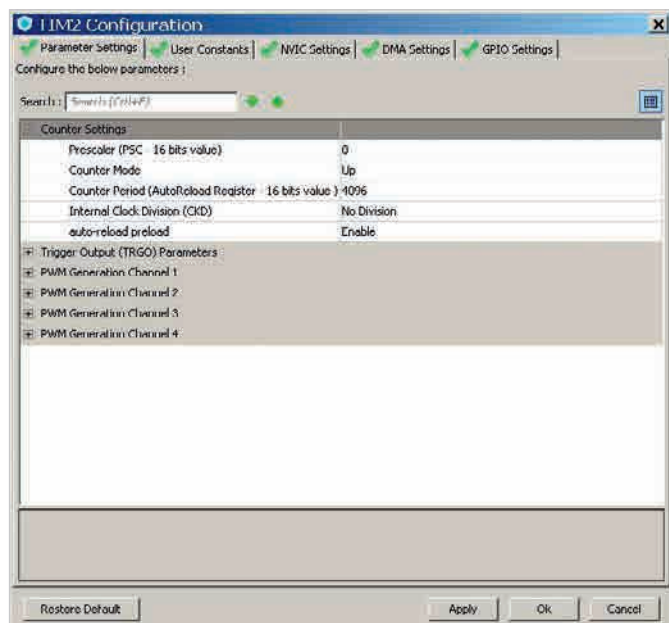


Рис. 5. Окно TIM2 генератора кода STM32CubeMX

Листинг 3

```

/* USER CODE BEGIN 3 */
HAL_ADCEx_InjectedStart(&hadc1); // Пуск инжекторных каналов АЦП
HAL_ADCEx_InjectedPollForConversion(&hadc1, 100); // Ожидание преобразования АЦП
HAL_ADCEx_InjectedStop(&hadc1); // Останов инжекторных каналов АЦП
// Чтение данных АЦП в буфер
ai[0]=ADC1->JDR1; // Канал 1
ai[1]=ADC1->JDR2; // Канал 2
ai[2]=ADC1->JDR3; // Канал 3
ai[3]=ADC1->JDR4; // Канал 4
// Управление скважностью каналов ШИМ
TIM2->CCR1= ai[0]; // ШИМ1
TIM2->CCR2=ai[1]; // ШИМ2
TIM2->CCR3=ai[2]; // ШИМ3
TIM2->CCR4=ai[3]; // ШИМ4
HAL_Delay(100); // Задержка в мс
HAL_GPIO_TogglePin(GPIOC, GPIO_PIN_9); // Переключение зелёного светодиода
if(HAL_GPIO_ReadPin(GPIOA, GPIO_PIN_0)) // Если нажата кнопка пользователя
{ // Передать данные АЦП через порт USART1
printf(str, "AI1=%d AI2=%d AI3=%d AI4=%d%c%c", ai[0], ai[1], ai[2], ai[3], '\n', '\r');
i=0;
while(str[i]!=0) {HAL_UART_Transmit(&huart1, &str[i], 1, 1); i++;}
}
}
/* USER CODE END 3 */

```

Сначала необходимо объявить новые переменные, которые будут использоваться в программе, как показано в листинге 1. Затем вводятся строки программы, выполняемые один раз при включении устройства (см. листинг 2). Они позволяют вывести строку названия проекта через порт UART1, включить два светодиода и запустить каналы ШИМ.

В листинге 3 приведены строки, которые выполняются в бесконечном цикле. Эта часть программы позволяет осуществить преобразование входных сигналов с помощью АЦП, сохранить полученные данные в буфере, загрузить их в каналы ШИМ и передать через последовательный порт USART1 при нажатии пользова-

тельской кнопки. Помимо этого, производится регулярное переключение зелёного светодиода, позволяющее визуально контролировать работу преобразователя. В бесконечном цикле также формируется задержка в 100 мс для удобства наблюдения за процессом преобразования. Эту задержку можно изменять в широком диапазоне в зависимости от условий задачи или вовсе исключить.

Написанную программу необходимо оттранслировать с помощью среды разработки Keil и загрузить в отладочную плату. Сразу после загрузки программа начнёт работать и преобразовывать аналоговые сигналы, поступающие по четырём входным каналам АЦП, в четыре выходных

Листинг 1

```

/* USER CODE BEGIN 1 */
uint32_t i; // Переменная циклов
uint32_t ai[4]; // Буфер данных АЦП
uint8_t str[80]={'4','x','A','I','-','>','4','x','P','W','M',' ','v','1','.', '0','\n','\r','0'}; // Буфер строки вывода
/* USER CODE END 1 */

```

Листинг 2

```

/* USER CODE BEGIN 2 */
// Вывести строку названия проекта через порт UART1
i=0;
while(str[i]!=0) {HAL_UART_Transmit(&huart1, &str[i], 1, 1); i++;}
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_8, GPIO_PIN_SET);
// Включить синий светодиод
HAL_GPIO_WritePin(GPIOC, GPIO_PIN_9, GPIO_PIN_SET);
// Включить зелёный светодиод
// Пуск ШИМ каналов
HAL_TIM_PWM_Start(&tim2, TIM_CHANNEL_1); // Пуск ШИМ1 таймера 2
HAL_TIM_PWM_Start(&tim2, TIM_CHANNEL_2); // Пуск ШИМ2 таймера 2
HAL_TIM_PWM_Start(&tim2, TIM_CHANNEL_3); // Пуск ШИМ3 таймера 2
HAL_TIM_PWM_Start(&tim2, TIM_CHANNEL_4); // Пуск ШИМ4 таймера 2
/* USER CODE END 2 */

```

канала ШИМ. Индикатором нормальной работы будет служить мигающий зелёный светодиод. Кроме того, можно визуально контролировать результат преобразования по всем каналам с помощью терминальной программы при подключении персонального компьютера к отладочной плате через порт USART1 и нажатии пользовательской кнопки.

При необходимости можно изменить временные параметры цифровых сигналов ШИМ с помощью перестройки блока синхронизации в генераторе кода STM32CubeMX с последующим формированием нового кода программы. При повторном формировании файлов проекта введённые в программу строки кода сохранятся, поскольку они расположены в защищённых участках между строками `/* USER CODE BEGIN ...*/` и `/* USER CODE END ...*/`.

Если потребуется преобразовывать аналоговые сигналы с другим диапазоном напряжений, это легко сделать, изменив значения входных резистивных делителей сигналов.

Таким образом, широко распространённую отладочную плату STM32VLDISCOVERY можно превратить в 4-канальный преобразователь сигналов управления для систем автоматики.

ЛИТЕРАТУРА

1. www.st.com
2. <http://www.st.com/en/evaluation-tools/stm32vldiscovery.html>
3. www.keil.com





**TESTING
DAYS
MOSCOW**

Тематическая выставка –
форум систем и технологий
для автомобильных и авиационных
испытаний и тестирования

Одновременно с
Control Days.Moscow



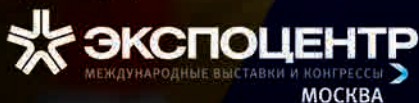
МОСКВА
Экспоцентр

**2-4
апреля
2019**

Акустика Ударные стенды
Пробоподготовка Аэродинамика
Мультиметры Телеметрия
Многоканальные измерительные системы
Анализаторы сигналов ЭМС
Испытания космических средств выведения
Испытательное моделирование
Испытания авиационных систем
Климатические испытания
Сенсорная измерительная аппаратура
Испытания автомобилей Виброиспытания
Моделирование ЛА Летные испытания
Механические испытания

При поддержке:

Реклама



#testingdays_moscow

+7 (495) 78-601-78

www.testingdays.moscow