

Министерство образования и науки
Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра конструирования узлов и деталей радиоэлектронной аппаратуры
(КУДР)

А.К. Пащенко, А.А. Бомбизов

АНАЛОГО-ЦИФРОВОЙ ПРЕОБРАЗОВАТЕЛЬ

Методические указания к выполнению
лабораторной и самостоятельной работы
по дисциплине «Микропроцессорные устройства»

УДК 004.42

ББК 32.973

П226

Рецензент:

Тренкаль Е.И., доцент кафедры конструирования узлов и деталей радиоэлектронной аппаратуры ТУСУР, канд. техн. наук

Пащенко Антон Константинович, Бомбизов Александр Александрович

П226 Аналого-цифровой преобразователь. Методические указания к выполнению лабораторной и самостоятельной работы по дисциплине «Микропроцессорные устройства» / А.К. Пащенко, А.А. Бомбизов. – Томск. гос. ун-т систем упр. и радиоэлектроники, 2026. – 18 с.

Методическое пособие содержит краткую теорию по работе с аналогово-цифровым преобразователем и порядок выполнения лабораторной работы.

Одобрено на заседании каф. КУДР, протокол № 261 от 6 февраля 2026 г.

УДК 004.42

ББК 32.973

© Пащенко А.К., Бомбизов А.А., 2026

© Томск. гос. ун-т систем упр. и радиоэлектроники, 2026

Содержание

1 Введение.....	4
2 Аналого-цифровой преобразователь.....	4
2.1 Основные положения.....	4
2.2 Описание АЦП в МК STM32F429	8
2.3 Каналы АЦП	10
2.4 Регистры АЦП	11
2.5 Одиночное преобразование	13
3 Вывод текстовой информации на дисплей.....	13
4 Порядок выполнения работы	15
5 Контрольные вопросы	17
Список использованных источников	18

1 Введение

Аналого-цифровой преобразователь (АЦП) является одним из важных элементов современных измерительных и управляющих систем. Его основная функция – преобразование непрерывного аналогового сигнала (например, напряжения или тока) в дискретный цифровой код, который может быть обработан, проанализирован, сохранен или передан с помощью цифровых устройств, таких как микроконтроллеры, процессоры или компьютеры.

Таким образом, АЦП служит мостом между миром аналоговых величин и дискретных цифровых сигналов, позволяя вычислительным устройствам обрабатывать различные физических величины, такие как температура, давление, звук, освещенность, скорость и другие.

Целью настоящей работы является освоение принципов работы аналого-цифрового преобразователя, при помощи которого микроконтроллер может считывать внешние аналоговые сигналы представляя их в цифровом эквиваленте дискретных отсчетов для последующей обработки.

2 Аналого-цифровой преобразователь

2.1 Основные положения

Аналого-цифровой преобразователь (АЦП, англ. Analog-to-digital converter, ADC) – устройство, преобразующее входной аналоговый сигнал в дискретный код (цифровой сигнал).

Основными параметрами, характеризующими АЦП являются:

- разрешение – минимальное изменение величины аналогового сигнала, которое может быть преобразовано данным АЦП;
- разрядность – характеризует количество дискретных значений, которые преобразователь может выдать на выходе. Так, 8-битный АЦП разбивает диапазон измерений на 256 дискретных значений (от 0 до 255);

– частота дискретизации (sample rate) – частота отбора отсчетов непрерывного во времени сигнала при его дискретизации. Измеряется в Герцах или MSPS.

Существуют несколько видов АЦП, отличающихся своим принципом действия. Наиболее распространены следующие типы:

- АЦП прямого преобразования;
- АЦП последовательного преобразования;
- дельта-сигма АЦП.

АЦП в используемом микроконтроллере представляет собой преобразователь последовательного приближения (SAR, Successive Approximation Register) (рисунок 2.1).

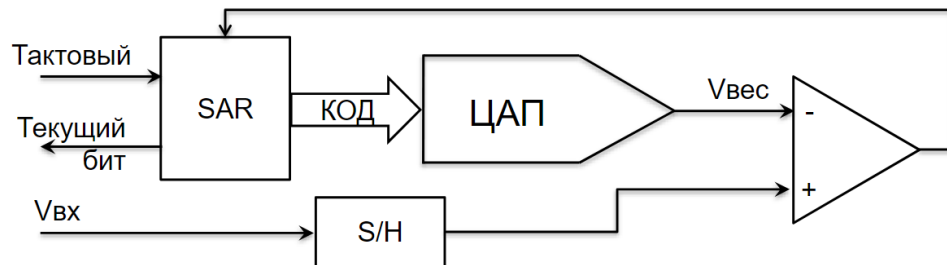


Рисунок 2.1 – Структура АЦП последовательного приближения

Для получения корректных значений на выходе АЦП последовательного приближения важно, чтобы входное напряжение сохраняло неизменную величину в течение всего цикла преобразования. Реальные сигналы имеют свойство меняться во времени. Чтобы решить данную проблему в схему АЦП включен блок выборки-хранения (Sample/Hold, *S/H*). Из рисунка 2.1 видно, что входной сигнал $V_{ВХ}$ поступает на блок *S/H*, где сохраняется и остаётся неизменным на протяжении всего времени преобразования. Зафиксированный сигнал подаётся на компаратор, где сравнивается со значением, генерируемом на цифро-аналоговом преобразователе (ЦАП).

Генерация значения на ЦАП, а вместе с тем и измерение входного аналогового сигнала осуществляется путем нескольких последовательных «взвешиваний», то есть сравнений величины входного напряжения с рядом величин, генерируемых следующим образом:

Шаг 1. На выходе встроенного ЦАП устанавливается напряжение $V_{\text{ВЕС}}$, соответствующее $1/2V_{\text{REF}}$ (здесь и далее предполагается, что сигнал находится в интервале $(0 - V_{\text{REF}})$).

Шаг 2. Затем компаратор осуществляет сравнение. Если входной сигнал больше установленной на ЦАП величины, то в блок SAR будет передано 1, иначе 0.

Шаг 3. С учетом полученного значения SAR генерирует новую последовательность для ЦАП, добавляя или убирая величину прошлого уровня сигнала при формировании нового. К итоговому уровню добавляется величина, посередине оставшегося интервала, т.е. $1/4V_{\text{REF}}$. Таким образом, если сигнал был выше установленного уровня, тогда следующее сравнение будет производиться с уровнем $3/4V_{\text{REF}}$, иначе – с уровнем $1/4V_{\text{REF}}$.

Шаг 4. Шаги 2 и 3 повторяется N раз. Таким образом, N сравнений («взвешиваний») порождает N бит результата. N характеризуется разрядностью АЦП.

Рассмотрим пример работы АЦП при следующих параметрах:

- опорное напряжение $V_{\text{REF}} = 3 \text{ В}$;
- входное напряжение $1,65 \text{ В}$;
- разрядность АЦП 4 бит (диапазон дискретных значений от 0 до 15, разрешение составит $0,2 \text{ В}$).

На первом шаге будет произведено сравнение входного сигнала с напряжением ЦАП примерно равным половине опорного напряжения: $8/15V_{\text{REF}}$. По результату сравнения будет установлено значение старшего бита в регистре данных АЦП.

Доля от V_{REF}	Добавочное значение, В	Сравнение уровней	Результат сравнения	Бит регистра данных АЦП	
				Номер	Значение
8/15	1,6	$1,65 > 1,6$	истина	3	1

В результате сравнения уровень напряжения входного сигнала оказался выше напряжения на ЦАП. Таким образом, старший бит в регистре выходных данных АЦП будет установлена в 1.

На следующем шаге сравнения, к текущему уровню на ЦАП будет добавлена $4/15V_{REF}$ для дальнейшего сравнения.

Доля от V_{REF}	Добавочное значение, В	Сравнение уровней	Результат сравнения	Бит регистра данных АЦП	
				Номер	Значение
4/15	0,8	$1,65 > 2,4$	ложь	2	0

В результате сравнения уровень напряжения входного сигнала оказался ниже напряжения на ЦАП. Таким образом, следующий бит в регистре выходных данных АЦП будет сброшен в 0.

Далее, к предыдущему значению уровня напряжения ЦАП 1,6 В, будет добавлено $2/15V_{REF}$.

Доля от V_{REF}	Добавочное значение, В	Сравнение уровней	Результат сравнения	Бит регистра данных АЦП	
				Номер	Значение
2/15	0,4	$1,65 > 2,0$	ложь	1	0

В результате уровень напряжения входного сигнала ниже напряжения на ЦАП, следующий бит регистра равен 0.

Аналогичная операция повторяется для значения $1/15V_{REF}$.

Доля от V_{REF}	Добавочное значение, В	Сравнение уровней	Результат сравнения	Бит регистра данных АЦП	
				Номер	Значение
1/15	0,2	$1,65 > 1,8$	ложь	0	0

По результату сравнения последний бит регистра равен 0.

Итоговое значение в регистре данных АЦП составит $1000_2 = 8_{10}$.

Ниже приведена формула перевода результата преобразования АЦП в напряжение (2.1):

$$V_{изм} = \frac{ADC_DR \times V_{REF}}{2^n - 1} \quad (2.1)$$

где ADC_DR – содержимое регистра данных АЦП;

V_{REF} – опорное напряжение АЦП;

n – разрядность АЦП.

Используя формулу (2.1) можно преобразовать значение из регистра данных АЦП в измеренный уровень напряжения входного сигнала:

$$V_{\text{изм}} = \frac{8 \times 3}{2^4 - 1} = \frac{24}{15} = 1,6 \text{ (В)}$$

2.2 Описание АЦП в МК STM32F429

Используемый в лабораторном курсе микроконтроллер включает в себя 3 блока АЦП, которые могут функционировать независимо или совместно [1, раздел 13].

Каждый из АЦП характеризуется следующим набором параметров:

- настраиваемая разрядность 6, 8, 10 или 12 бит;
- одиночное и непрерывное преобразование;
- режим сканирования каналов по заданному списку;
- вариативность запуска преобразования: программное, по прерыванию, от внешнего источника;
- 16 аналоговых каналов;
- диапазон значений входного сигнала $V_{REF-} \leq V_{IN} \leq V_{REF+}$;
- скорость оцифровки до 2,4 MSPS. В совместном (утроенном) режиме до 7,2 MSPS.

Аналоговые каналы могут быть обработаны в составе регулярной или инжектированной группы.

В регулярной группе каналы обрабатываются последовательно друг за другом, а результат сохраняется в единственный регистр.

Каналы в инжектированной группе обрабатываются независим друг от друга, а результат преобразования сохраняется в индивидуальный для каждого канала регистр.

Каналы в инжектированной группе имеют более высокий приоритет перед каналами в регулярной группе. Таким образом, при одновременном возникновении сигнала начала преобразования для каналов в обеих группах,

сначала будет выполнено преобразование канала из инжектированной группы, а потом из регулярной.

Так как инжектированные каналы представляют собой, по сути, отдельные каналы, на их обслуживание требуется больше ресурсов. В связи с этим, в МК можно настроить только 4 инжектированных канала, в то время как в регулярную группу могут быть добавлены все 16 каналов.

На рисунке 2.2 приведена структурная схема АЦП [1, рисунок 44].

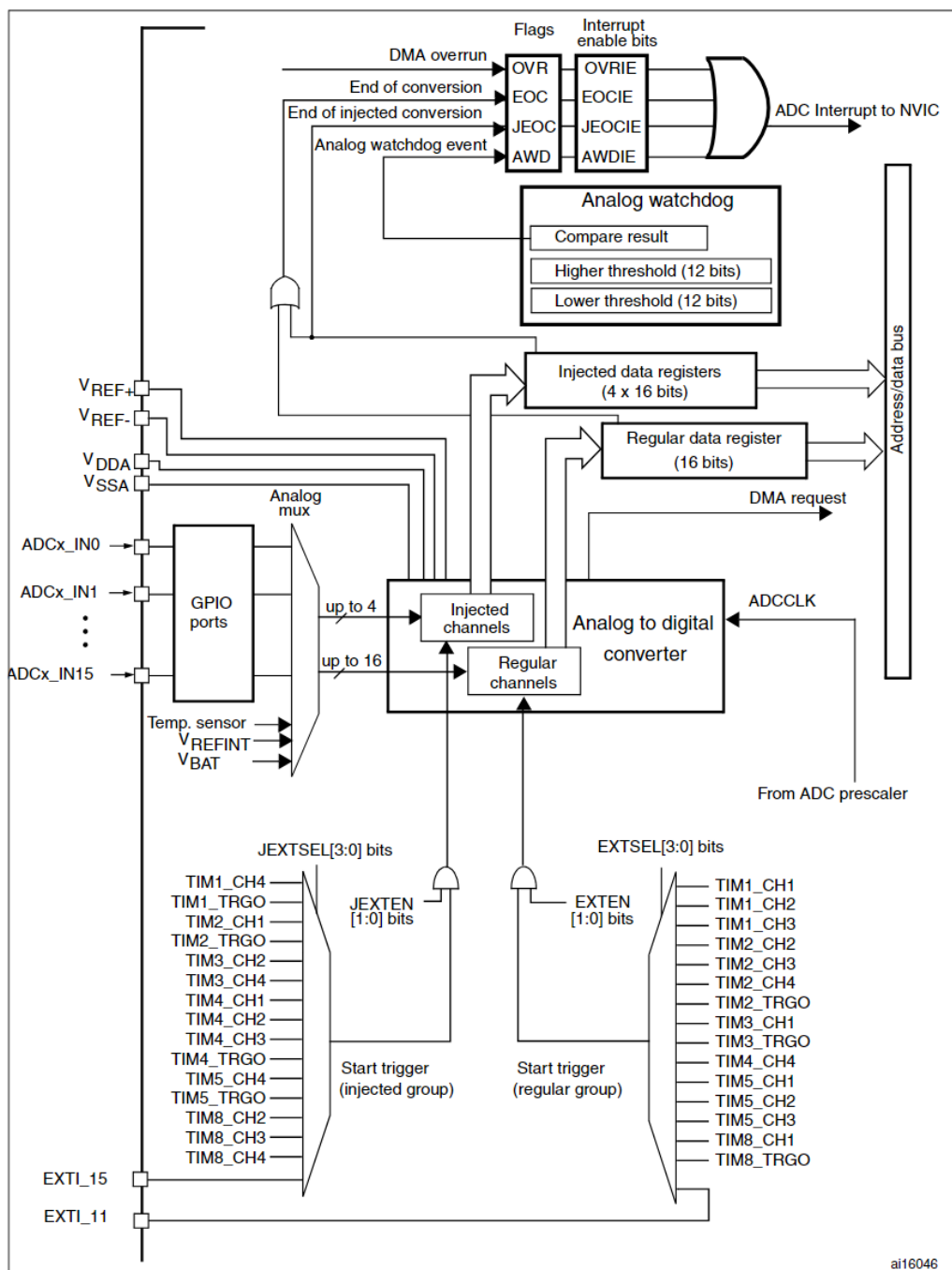


Рисунок 2.2 – Структурная схема АЦП

Из структурной схемы (рисунок 2.2) видно, что все аналоговые каналы ADCx_IN0...ADCx_IN15 представляют собой порты ввода-вывода (GPIO), настроенные в аналоговом режиме. Помимо внешних каналов, существуют и внутренние: сенсор температуры (ADC1_IN18), опорное напряжение (ADC1_IN17), напряжение аккумулятора (ADC1_IN18) [1, 13.3.3]. При этом, в МК STM32F429 каналы сенсора температуры и напряжения аккумулятора объединены в один канал АЦП. Таким образом, в один момент времени может быть произведено преобразования только одного из выбранных параметров.

При помощи аналогового мультиплексора задаются каналы, которые будут обработаны в регулярной или инжектированной группах.

В нижней части схемы изображены источники запуска преобразований АЦП.

В верхней части схемы изображены генерируемые АЦП прерывания. Так после окончания преобразования значения регулярной группы будет установлен флаг **ЕОС** и может быть сгенерирован сигнал **ЕОСИЕ**, который будет обработан контролером прерываний NVIC.

2.3 Каналы АЦП

Как было ранее отмечено, каналы АЦП представляют собой выводы GPIO, настроенные в аналоговом режиме.

Распределение аналоговых входов по выводам микроконтроллера приведено в спецификации [2, раздел 4, таблица 10].

Для примера, ниже приведена часть таблицы, описывающей назначение выводов микроконтроллера (рисунок 2.3).

Pin number								Pin name (function after reset) ⁽¹⁾	Pin type	I / O structure	Notes	Alternate functions	Additional functions
LQFP100	LQFP144	UFBGA169	UFBGA176	LQFP176	WLCSP143	LQFP208	TFBGA216						
-	21	NC (2)	L2	27	G8	30	L2	<u>PF9</u>	I/O	FT	(5)	SPI5_MOSI, SAI1_FS_B, TIM14_CH1, FMC_CD, EVENTOUT	<u>ADC3_IN7</u>
-	22	H1	L1	28	G9	31	L1	<u>PF10</u>	I/O	FT	(5)	FMC_INTR, DCMI_D11, LCD_DE, EVENTOUT	<u>ADC3_IN8</u>
12	23	G2	G1	29	J11	32	G1	PH0-OSC_IN (PH0)	I/O	FT		EVENTOUT	OSC_IN ⁽⁵⁾
13	24	G1	H1	30	H10	33	H1	PH1- OSC_OUT (PH1)	I/O	FT		EVENTOUT	OSC_OUT ⁽⁵⁾
14	25	H2	J1	31	H9	34	J1	NRST	I/O	RS T			
15	26	G6	M2	32	H8	35	M2	<u>PC0</u>	I/O	FT	(5)	OTG_HS_ULPI_STP, FMC_SDNWE, EVENTOUT	<u>ADC123_</u> <u>IN10</u>

Рисунок 2.3 – Часть таблицы описания выводов МК

Из рисунка 2.3 видно, что вывод PF9 является каналом 7 для АЦП3. Некоторые выводы микроконтроллера могут представлять собой один и тот же аналоговый канал для различных АЦП. Например, вывод PC0 является каналом 10 одновременно для всех АЦП.

2.4 Регистры АЦП

ADC_SR – регистр статуса, позволяющий отслеживать состояния АЦП:

- окончание преобразования канала в регулярной группе (EOC);
- окончание преобразования канала в инжектированной группе (JEOC);
- прочее.

ADC_CR1 – первый регистр настроек, позволяющий:

- настроить разрешение АЦП (RES);
- настроить количество каналов в регулярной группе (DISCNUM);
- активировать режим последовательного опроса каналов (SCAN);

- включить прерывание по окончании преобразования (EOCIE);
- и прочее.

ADC_CR2 – второй регистр настроек, позволяющий:

- включить или выключить АЦП (ADON);
- включить режим непрерывного преобразования (CONT);
- программно запустить процесс преобразования (SWSTART);
- прочее.

ADC_SMPRx (x=1..2) – регистры настройки времени семплирования.

Чем больше времени будет затрачено на преобразование, тем более точное значение может быть получено.

ADC_SQRx (x=1..3) – регистры настройки мультиплексора регулярных каналов, позволяющий:

- задать общее количество каналов для оцифровки (L);
- номера аналоговых каналов для оцифровки (SQ1–SQ16).

ADC_JSQR – регистр настройки мультиплексора инжектированных каналов. Функции аналогичны регистрам ADC_SQRx.

ADC_DR – регистр данных, хранит результат преобразования канала из регулярной группы.

ADC_JDRx (x= 1..4) – регистры данных. Каждый из регистров хранит результат преобразования отдельного канала из инжектированной группы.

ADC_CCR – регистр общих настроек, позволяющий:

- активировать каналы оцифровки значений сенсора температуры и опорного напряжения (TSVREFE);
- переключить канал оцифровки значения сенсора температуры на отслеживание напряжения аккумулятора (VBATE);
- настроить предделитель тактового сигнала для АЦП (ADCPRE);
- прочее.

2.5 Одиночное преобразование

Для запуска одиночного преобразования пользователю необходимо произвести следующие действия:

- разрешить тактирование необходимых портов ввода-вывода;
- настроить необходимые выводы в аналоговый режим;
- разрешить тактирование АЦП;
- задать количество каналов, которое будет оцифровано (регулярной или инжектированной группы);
- задать номера каналов, которые необходимо оцифровать (регулярной или инжектированной группы);
- установить режим одиночного преобразования;
- включить АЦП;
- запустить преобразование (например, программно).

Для того, чтобы прочитать оцифрованное значение, необходимо:

- ожидать установки флага по окончании преобразования (регулярной или инжектированной группы);
- если флаг установлен, прочитать значение из регистра данных (регулярной или инжектированной группы).

3 Вывод текстовой информации на дисплей

Для корректного вывода оцифрованных значений в текстовом формате на дисплей, потребуется рассмотреть несколько функций из графических библиотек.

Вывод текстовой информации на дисплей осуществляется при помощи соответствующей библиотечной функции:

```
void UB_Font_DrawString(uint16_t x, uint16_t y,  
                        char *ptr, UB_Font *font,  
                        uint16_t vg, uint16_t bg);
```

где x и y – координаты начала строки;

ptr – указатель на начало массива символов;

font – указатель на шрифт;
vg – цвет текста;
bg – цвет фона строки.

В качестве шрифта функция принимает параметр типа **UB_Font**, представляющего собой следующую структуру:

```
typedef struct UB_Font_t {  
    const uint16_t *table; // Таблица символов  
    uint16_t width;        // Ширина символа (в пикселях)  
    uint16_t height;       // Высота символа (в пикселях)  
} UB_Font;
```

В библиотеке “stm32_ub_font.h” пользователю доступны следующие шрифты:

- Arial_7x10; //высота 7, ширина 10
- Arial_8x13; //высота 8, ширина 13
- Arial_10x15; //высота 10, ширина 15
- Arial_11x18; //высота 11, ширина 18
- Arial_13x19; //высота 13, ширина 19
- Arial_14x22; //высота 14, ширина 22
- Arial_16x25. //высота 15, ширина 25

Пример использования функции отображения текста:

```
UB_Font_DrawString(10, 100, "Hello", &Arial_14x22,  
                   RGB565(0,0,0), RGB565(31,63,31));
```

Однако, если сразу после инициализации дисплея вывести текст на экран, он будет перевернутым. Чтобы это исправить, необходимо воспользоваться функцией

```
void UB_LCD_Rotate_180(void),
```

которая осуществляет переворот дисплея на 180°. Пример результата от использования данной функции приведен на рисунке 3.1.

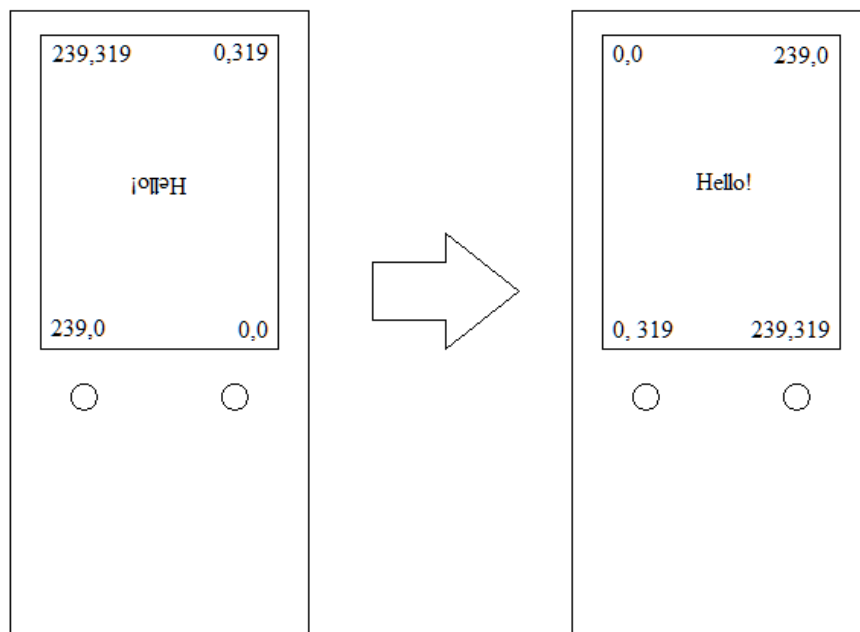


Рисунок 3.1 – Результат переворота дисплея на 180°

В ходе работы потребуется функция для осуществления задержки. В графической библиотеке эту роль выполняет функция

```
void UB_Systick_Pause_ms(uint32_t pause),
```

которая принимает в качестве параметра значение задержки в миллисекундах.

Для преобразования значения напряжения из числового типа в строковый была добавлена функция

```
void Convert_volt_to_str(char *str_volt, int volt),
```

которая принимает два параметра:

- str_volt – указатель на символьный массив, куда сохранить строку;
- volt – целочисленное значение.

4 Порядок выполнения работы

В ходе данной работы будут освоены следующие разделы: 1) освоение работы с аналого-цифровым преобразователем; 2) вывод текстовой информации на дисплей.

1. Изучите предложенный в п. 2 и п. 3 теоретический материал.
2. Откройте проект-заготовку для лабораторной работы.
3. По аналогии с предыдущей работой сконфигурируйте дисплей.

4. Добавьте вызов функции инициализации системного счетчика `UB_Systick_Init()`;
5. Создайте целочисленную беззнаковую двухбайтовую переменную `lcd_bg` и присвойте ей значение цвета фона нижнего слоя дисплея.
6. Измените ориентацию экрана выполнив функцию `UB_LCD_Rotate_180()`.
7. Определите букву порта и номер вывода для 7 канала ADC1.
8. Разрешите тактирование и настройте этот вывод в аналоговый режим.
9. Определите шину тактирования ADC1 и разрешите тактирование АЦП.
10. Настройте седьмой канал в мультиплексоре регулярных каналов АЦП.
11. Разрешите работу АЦП в регистре CR2;
12. Установите бит программного запуска преобразования АЦП.
13. Выполните переключение дисплея на нижний слой.
14. Внутри бесконечного цикла (`while(1)`) добавьте проверку готовности оцифрованного значения (окончания преобразования).
15. Если значение было оцифровано:
 - а. Создайте целочисленную переменную `val_of_ADC` и запишите в неё содержимое регистра данных АЦП.
 - б. Создайте целочисленную переменную `volt`.
 - в. Преобразуйте содержимое переменной `val_of_ADC` в милливольты (опорное напряжение 3 В, разрядность АЦП 12 бит) и присвойте результат в переменную `volt`.
 - г. Создайте символьный массив `str_volt` длиной 17 байт;
 - д. Преобразуйте полученное значение напряжения в строку: `Convert_volt_to_str (str_volt, volt)`;
 - е. Выведите строку на дисплей используя функцию `UB_Font_DrawString`. В качестве цвета фона используйте ранее определенную переменную `lcd_bg`.
 - ж. Используя функцию `UB_Systick_Pause_ms`, добавьте задержку длительностью 1 секунда;

- з. Установите бит программного запуска преобразования АЦП.
16. Соедините проводом выводы PA7 и 3V на плате.
17. Выполните сборку проекта, после чего загрузите программу на плату.
На дисплее должно отобразиться значение измеренного напряжения, например “3000 mV”.
18. Соедините проводом выводы PA7 и GND на плате. Убедитесь, что значение на дисплее изменилось.
19. Оформите отчет, содержащий титульный лист, введение, ход выполнения работы, ответы на контрольные вопросы и выводы.
20. Защитите отчет у преподавателя.

5 Контрольные вопросы

- 5.1 Для чего нужен АЦП?
- 5.2 Что такое разрешение, разрядность и дискретность АЦП?
- 5.3 Чем отличаются регулярные каналы от инжектированных?
- 5.4 Какое максимальное количество регулярных каналов?
- 5.5 Какое максимальное количество инжектированных каналов?
- 5.6 Какой флаг в регистре статуса устанавливает АЦП после окончания преобразования в регулярной группе?

Список использованных источников

- 1 RM0090. Reference manual. STM32F405/415, STM32F407/417, STM32F427/437 and STM32F429/439 advanced ARM®-based 32-bit MCUs.– URL: www.st.com/resource/en/reference_manual/DM00031020.pdf (дата обращения: 17.11.2025);
- 2 Datasheet. STM32F427xx/ STM32F429xx ARM Cortex-M4 32b MCU+FPU, 225DMIPS, up to 2MB Flash/256+4KB RAM, USB OTG HS/FS, Ethernet, 17 TIMs, 3 ADCs, 20 comm. interfaces, camera & LCD-TFT. – URL: <https://static.chipdip.ru/lib/691/DOC034691022.pdf> (дата обращения: 18.11.2025)