Отладочная плата STM32F429I в качестве осциллографа

Валентин Юрзин, Сергей Недорезов (г. Апатиты, Мурманская обл.)

Статья содержит материалы, которые могут быть полезными при изучении микроконтроллеров с архитектурой ARM Cortex-M4. Приведена информация, необходимая для начала работы с отладочной платой STM32F429I-Discovery, и краткое описание работы встроенных в микроконтроллер АЦП, графического ускорителя и флэш-памяти.

Компания STMicroelectronics представила в конце 2013 года отладочную плату STM32F429 в серии Discovery [1]. Новая версия платы предоставляет больше возможностей, так как содержит цветной графический ЖК-дисплей. Для оценки производительности системы отображения и АЦП (ADC – Analog-to-Digital Converter) микроконтроллера авторами статьи было написано приложение, которое превращает отладочную плату в цифровой осциллограф.

Постановка задачи

Задача осциллографирования заключается в том, чтобы преобразовывать аналоговый сигнал на входе АЦП в цифровой код и выводить его на дисплей в реальном масштабе времени. Отображение данных на ЖК-дисплее будет формироваться условными «кадрами», примерно такими же, как в обычных мониторах с частотой развёртки 50–100 Гц. Для этого создан массив, который заполняется данными из АЦП и выводится на дисплей с интервалом 10–20 мс. Рисование кадра и его стирание занимает определённое время, поэтому требуется найти оптимальное решение для прорисовки дисплея.

Аналого-цифровой преобразователь имеет несколько режимов работы с использованием прямого доступа к памяти ПДП (DMA – Direct Memory Access), которые будут проверены на производительность. Чтобы оперативно изменять режимы работы осциллографа, используется электромеханическое устройство ввода (энкодер). Для сохранения некоторых текущих параметров управления понадобится флэш-память контроллера. Полный текст программы осциллографа приведён в дополнительных материалах к статье, доступных для скачивания на сайте журнала.

Краткое описание платы STM32F429I-Discovery

Отладочная плата STM32F429I-DISCOVERY от компании STMicroelectronics предназначена для оценки высокопроизводительного микроконтроллера STM32F429ZIT6. Плата оснащена всем необходимым для начала разработки собственных приложений и содержит микроконтроллер STM32F429ZIT6 в корпусе LQFP-144, 2,4-дюймовый цветной графический ЖК-дисплей (240 × 320 пикселей) с сенсорной панелью, 64 Мбит внешней памяти типа SDRAM, 3-осевой гироскоп МЭМС, интерфейс USB OTG, индикаторные светодиоды, управляющие кнопки и встроенный программатор – отладчик ST-LINK/V2 с интерфейсом SWD (см. рис. 1). Большое количество демонстрационных приложений с исходными кодами доступно на интернет-странице компании-производителя [2].

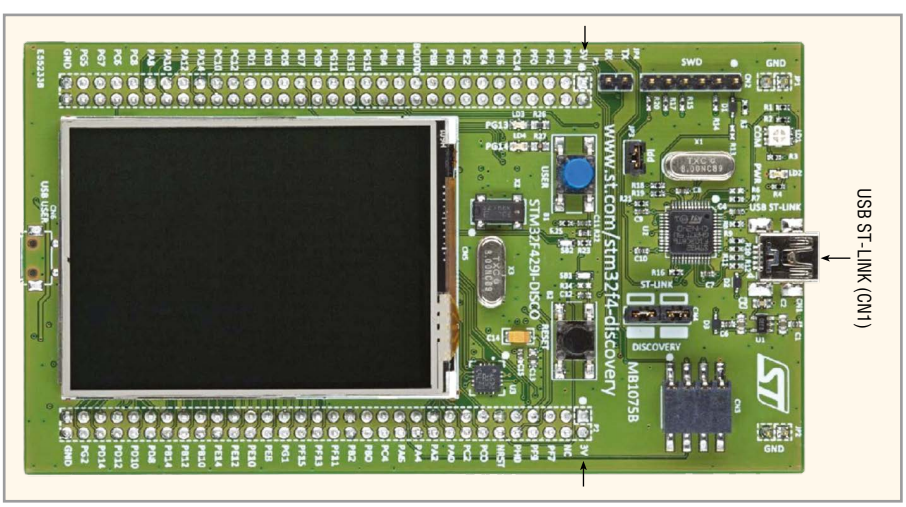

Рис. 1. Отладочная плата STM32F429I-DISCOVERY

Особенности микроконтроллеров STM32F429/439

Микроконтроллеры семейства STM32F4 построены на базе процессора ARM Cortex-M4, который представляет собой высокопроизводительный 32-битный RISC-процессор с поддержкой инструкций цифровой обработки сигналов (DSP) и вычислений с плавающей точкой. Для оптимизации работы с памятью микроконтроллеры семейства снабжены ускорителем (ART Accelerator). В своём классе приборы STM32F429/STM32F439 в настоящий момент обладают максимальной производительностью и широким набором периферии (см. рис. 2).

Процессор ARM Cortex-M4 работает на частоте 180 МГц, достигая производительности 210 DMIPS. Объём флэш-памяти расширен до 2 Мбайт, а ОЗУ – до 256 кбайт. Появилась новая периферия: последовательный звуковой интерфейс SAI и контроллер внешней памяти Flexible Memory Controller (FMC), поддерживающий ИС памяти типа SDRAM. Самыми важными усовершенствованиями линейки МК являются интегрированный контроллер ЖК-дисплея и графический ускоритель Chrom-ART (DMA2D) (см. рис. 3).

Контроллер дисплея LTDC обеспечивает 24-битный RGB-интерфейс и все необходимые сигналы для взаимодействия с ЖК-панелями с разрешением до 640 × 480 (480 строк по 640 пикселей). Он имеет два экранных слоя с буферами FIFO (64 × 32 бит), до восьми форматов цвета на каждый слой, включая ARGB8888, RGB888, RGB565, ARGB1555, ARGB4444, L8, AL44, AL88 и таблицу цве-

тов (Color Look-Up Table) с поддержкой до 256 24-битных цветов на каждый слой [4].

Подключение платы и создание проекта в программной среде IAR Embedded Workbench

Одной из причин популярности микроконтроллеров STMicroelectronics является доступность готовых программных решений и библиотек для различных сред разработки и отладки программного обеспечения (Keil uVision, IAR и др.) [2]. Перед началом работы следует установить 30-дневную версию среды разработки IAR Embedded Workbench для ARM-процессоров, которую компания IAR Systems предоставляет для свободного скачивания. Для подключения платы к ПК через отладчик ST-LINK потребуется кабель USB с разъёмами USB типа A и mini-USB типа B; второй порт USB предназначен для непосредственного подключения к микроконтроллеру STM32F429ZI.

Для работы с платой необходимо установить драйверы операционной системы, а также загрузить и установить программу STM32 ST-Link Utility. Среда программирования IAR Embedded Workbench содержит функцию прошивки микроконтроллеров. Существуют и отдельные программы, которые предназначены для работы с памятью микроконтроллеров, например, ST Visual Programmer [3].

Создание нового проекта проще всего начать с готового примера [2]. Скопируем пример Touch_Panel в новый каталог и откроем его для работы. Сенсорную панель использовать не будем, поэтому удалим код инициализации и функцию вызова TP_Config(void). В данном примере уже подключены все необходимые библиотеки для работы с ЖК-дисплеем:

```
#include "stm32f4xx.h" #include
<stdio.h> #include "stm32f429i_
discovery.h";
#include "stm32f429i_discovery_
lcd.h" #include "stm32f429i_
discovery_ioe.h".
```

Дополним код счётчиком миллисекунд. Для этого в файле main.h объявим экспорт функции void TimingDelay_Decrement(void); в файле main.c создадим функцию void TimingDelay_Decrement(void) и объявим переменную uint16_t TimingDelay = 2000. В файле stm32f4xx_it.c заполним пустую функцию void SysTick_Handler(void) {TimingDelay_Decrement();}.

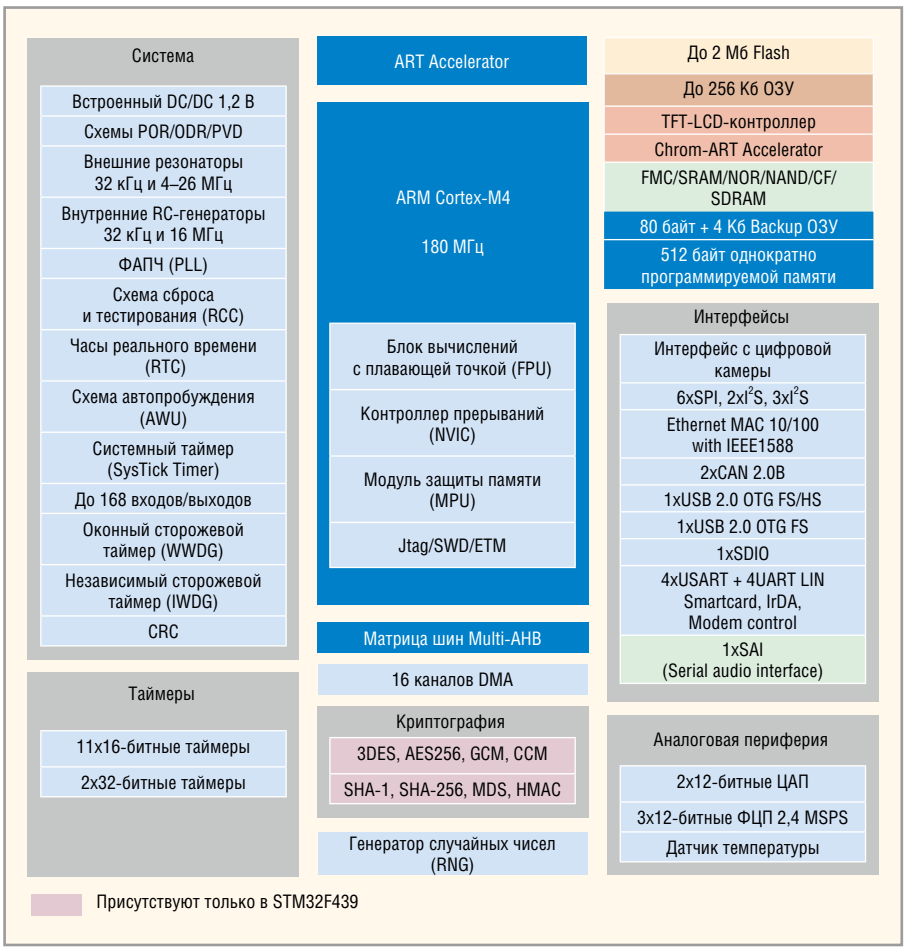

Рис. 2. Структурная схема микроконтроллера STM32F429

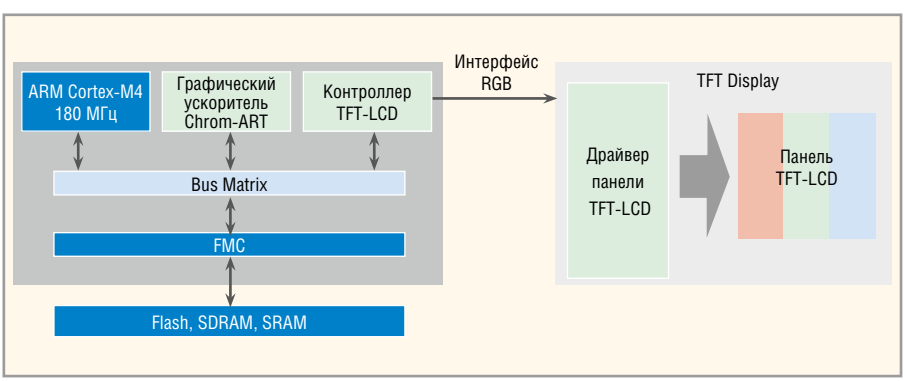

Рис. 3. Функциональная схема контроллера графического ЖК-дисплея

Работа с ЖК-дисплеем

На плате установлен графический ЖК-дисплей (типа TFT) с разрешением 240 × 320 пикселей. При выполнении какой-либо графической функции активизируется модуль прямого доступа к памяти DMA2D, который и является графическим ускорителем Chrom-ART. Он представляет собой специализированный контроллер ПДП, созданный для работы с графикой дисплея, и выполняет операции заливки изображения заданным цветом, копирование исходного изображения в заданную область памяти с дополнительным преобразованием формата цвета пикселей, смешивание изображений с дополнительным преобразованием формата цвета пикселей. Контроллер внешней памяти SDRAM повышает производительность при построении графических изображений.

Нарисуем осциллограмму из 285 пикселей по оси X вдоль длинной части

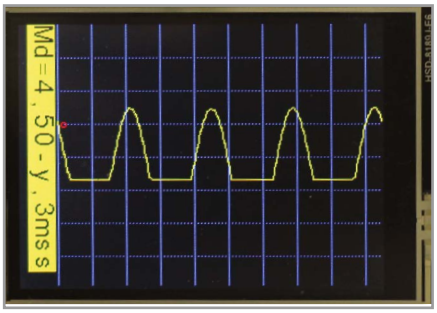

Рис. 4. Время прорисовки экрана дисплея составляет 3 мс

дисплея. В оставшуюся часть дисплея выведем текстовую строку по оси Y, где будут отображаться параметры развёртки, значение уровня захвата сигнала (красный кружок в левой части экрана) и режим отображения (Md = 1, 2, 3, 4). Установку цвета фона и текста производят функции:

```
LCD_SetBackColor(LCD_COLOR_
YELLOW);
LCD_SetTextColor(LCD_COLOR_
BLACK).
```

За вывод служебной строки с данными в нужную позицию отвечает функция:

```
LCD_DisplayStringLine(LCD_
LINE_12, (uint8_t*)aTextBuffer).
```

В графической библиотеке приведено несколько способов вывода точки на дисплей: в виде пикселя со значением закрашенного единичного радиуса, в виде одной линии и в виде линии по указанным точкам. Самым быстрым методом рисования линий оказалась функция отображения точками. Но она не подходит в данном примере, так как при больших изменениях значения по оси Y не прорисовывается линия связи. Поэтому используем функцию в виде линии по указанным точкам:

```
LCD_DrawUniLine(uint16_t x1,
uint16_t y1, uint16_t x2,
uint16_t y2).
```

Для отображения новой линии на дисплее предыдущая информация должна стираться, то есть требуется найти самый быстрый способ обновления дисплея. Для исключения мерцания при обновлении дисплея на обычном ПК изображение сначала формируется вне экранной области памяти, а затем готовый кадр копируется в область памяти дисплея. Такой способ обновления изображений в данной плате не тестировался из-за большого объёма передаваемых данных (копируются пиксели всего дисплея), что приводит к значительной задержке. Прорисовка одного кадра на дисплее с последующим стиранием также не дали положительный результат, так как функция стирания LCD_Clear(uint16_t Color) оказалась затратной по времени.

Оптимальным способом рисования линий с обновлением дисплея без мерцаний оказался метод, при котором перед прорисовкой новой линии с минимальным приращением в два пикселя по оси X необходимо стирать предыдущую информацию на дисплее рисованием цветом фона трёх линий толщиной в один пиксель по оси Y:

```
LCD_SetTextColor(LCD_COLOR_
BLACK);
LCD_DrawLine(0, X,237,LCD_DIR_
HORIZONTAL);
LCD_DrawLine(0, X-1,237,LCD_DIR_
HORIZONTAL);
LCD_DrawLine(0, X-2,237,LCD_DIR_
HORIZONTAL).
```

Вслед за формированием изображения сигнала, в этой же области дисплея рисуется масштабная сетка с клетками 30 × 30 пикселей с помощью выбранных методов:

```
// Сетка линии по горизонтали
LCD_SetTextColor(LCD_COLOR_
BLUE);
LCD_DrawCircle(30, X, 0);
// Сетка линии по вертикали
LCD_SetTextColor(LCD_COLOR_
BLUE);
LCD_DrawLine(0, X,237,LCD_DIR_
HORIZONTAL);
```

Стирание и прорисовка изображения на дисплее данными из АЦП происходит в цикле. Затем выводятся значения некоторых параметров для четырёх режимов работы:

```
/* Установки дисплея, цвет фона
и цвет текста */
 LCD_SetBackColor(LCD_COLOR_
YELLOW);
 LCD_SetTextColor(LCD_COLOR_BLACK);
 sprintf((char*)aTextBuffer,
MESSAGE6, Mode ,delay_zad/16 ,
Vmax/50,(Vmax%50)/5);
 sprintf((char*)aTextBuffer,
MESSAGE5, Mode ,delay_zad/3 ,
Vmax/50,(Vmax%50)/5);
 sprintf((char*)aTextBuffer,
MESSAGE3, Mode ,DelU , TimTemp );
 sprintf((char*)aTextBuffer,
MESSAGE4, Mode ,SensU , TimTemp );
```

Время обновления изображения на дисплее (вместе с вычислениями) составило от 12 до 15 мс в зависимости от плотности прорисовки. Время прорисовки экрана дисплея составило 2–3 мс в режиме отображения Md = 4 (см. дополнительные материалы к статье на сайте и рис. 4).

Настройка энкодера

Для переключения значений задержки, установки времени развёртки и изменения коэффициента деления используется энкодер PEC12–4. В основном цикле программы опрашивается кнопка, а в прерывании по таймеру (примерно 1 мс) определяется направление вращения энкодера. Далее в основном цикле программы считывается это направление, увеличиваются или уменьшаются значения и снова разрешается определение направления вращения. Для инициализации энкодера потребуется 5 свободных выводов контроллера (см. дополнительные материалы к статье на сайте).

Модуль флэш-памяти

Чтобы при отключении питания некоторые последние значения переменных сохранялись, записываем их во флэш-память микроконтроллера STM32F4xx. При этом необходимо помнить, что в ней хранится программа микроконтроллера. Поэтому область для записи данных необходимо выбрать в свободном от кода программы пространстве памяти, например, на последней странице:

```
/* Базовый адрес флэш-памяти
и его сектор */
#define DEVICE_ADDRESS
0x080E0000 // нач. адрес
#define DEVICE_SECTOR FLASH_
Sector_11 // сектор
```

Флэш-память МК делится на сектора. В контроллере находится 12 секторов, и каждый сектор отвечает за заданную область памяти. Стирать память можно только по секторам. Если необходимо сохранить какие-то данные в сектор памяти, но в нём уже хранится нужная информация, необходимо скопировать

её во временные переменные, а затем сохранить новые и старые данные. Для чтения из флэш-памяти достаточно указать необходимый адрес. Полный код работы с флэш-памятью приведён в дополнительных материалах к статье на сайте.

Использование АЦП и ПДП

Микроконтроллер STM32F429ZIT6 содержит три 12-разрядных АЦП поразрядного уравновешивания. Их мультиплексированные входные каналы позволяют измерять сигналы от 16 внешних источников, двух внутренних источников и канала Vbat. Преобразование может осуществляться в режиме непрерывного сканирования или в прерывистом режиме. Результат преобразования сохраняется и выравнивается по левому или правому краю в 16-разрядном регистре данных.

Основными функциями АЦП являются 12-, 10-, 8- или 6-разрядное преобразование, генерация прерывания в конце преобразования, выравнивание данных со встроенной последовательностью данных, внешний запуск, режимы с двойным и тройным чередованием (на устройствах с двумя или тремя АЦП), конфигурируемое хранение данных ПДП в указанных режимах.

Рассмотрим режимы преобразования АЦП. Режим одиночного преобразования использует ПДП для непрерывной передачи данных в память в циклическом режиме. Если при конфигурации АЦП установить время выборки 3 цикла и разрядность преобразования 12 бит, полное время преобразования составит 0,41 мкс (скорость 2,4 Мвыб/с). В двойном режиме происходит чередование записи данных двух АЦП в память с помощью контроллера ПДП. При конфигурировании задержки АЦП в 6 циклов и тактовой частоте АЦП 36 МГц, скорость преобразования в этом режиме составит 6 Мвыб/с. В режиме тройного чередования каналов АЦП последовательно генерируются три запроса ПДП, а скорость преобразования достигает 7,2 Мвыб/с.

Для осциллографирования мы будем использовать АЦП в режиме двойного чередования с задержкой в 5 циклов, тогда скорость преобразования при тактовой частоте 36 МГц должна составить 36/5 = 7,2 Мвыб/с. Если точности АЦП не хватает, задержку можно увеличить до 6 циклов.

Настройка АЦП производится следующим образом: включение тактирования порта, настройка выводов, включение тактирования АЦП, настройка АЦП, включение прерывания, включение АЦП. В качестве входа преобразователя ADC1 используется вывод PC.03 на внешнем разъёме. Входной сигнал может находиться в диапазоне 0...3 В.

```
/* Конфигурация АЦП: двойное чередование с задержкой 5 циклов */
 ADC_CommonInitStructure.ADC_Mode = ADC_DualMode_Interl;
 ADC_CommonInitStructure.ADC_TwoSamplingDelay = ADC_TwoSamplingDelay_5Cycles;
 ADC_CommonInitStructure.ADC_ПДП AccessMode = ADC_DMAAccessMode_2;
 ADC_CommonInitStructure.ADC_Prescaler = ADC_Prescaler_Div2;
 ADC_CommonInit(&ADC_CommonInitStructure);
 /* 12-разрядное преобразование*/
 ADC_InitStructure.ADC_Resolution = ADC_Resolution_12b;
```

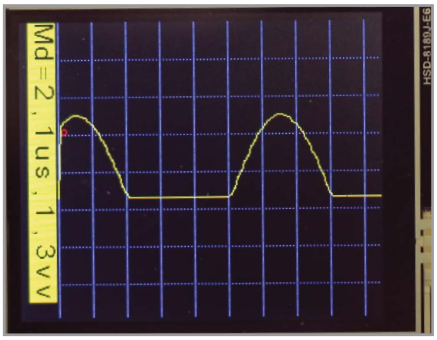

Рис. 5. Осциллограмма сигнала частотой 100 кГц

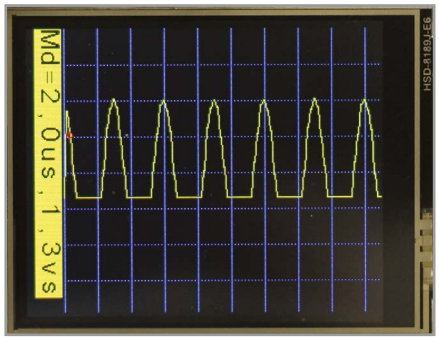

Рис. 6. Осциллограмма сигнала частотой 400 кГц

Пояснения:

ContinuousConvMode – этот режим запускает следующее преобразование сразу по окончании предыдущего, чем достигается максимальная скорость АЦП;

DataAlign – выравнивание данных в 2-байтном слове;

ADC_DataAlign_Right – данные выравниваются по правому краю;

ADC_DataAlign_Left – данные выравниваются по левому краю;

ExternalTrigConv – событие, которое запускает АЦП (можно сконфигурировать запуск по таймеру);

NbrOfConversion – число каналов, которые будет сканировать МК;

ScanConvMode – определяет, будет ли АЦП сканировать несколько каналов.

```
 /* ADC1 regular channel 13
configuration */
 ADC_InitStructure.ADC_
ScanConvMode = DISABLE;
 ADC_InitStructure.ADC_
ContinuousConvMode = ENABLE;
 ADC_InitStructure.ADC_
ExternalTrigConvEdge = ADC_
ExternalTrigConvEdge_None;
 ADC_InitStructure.ADC_
ExternalTrigConv = ADC_
ExternalTrigConv_T1_CC1;
 ADC_InitStructure.ADC_DataAlign
= ADC_DataAlign_Right;
 ADC_InitStructure.ADC_
NbrOfConversion = 1;
 /*Конфигурация преобразователя
АЦП1 и канала 13 */
 ADC_Init(ADC1, &ADC_
InitStructure);
 /* Конфигурация преобразователя
АЦП1 и канала 13 */
 ADC_RegularChannelConfig(ADC, ADC_
CHANNEL, 1, ADC_SampleTime_3Cycles);
 /* Включить ПДП после последней
передачи данных из АЦП (режим
мульти-АЦП) */
 ADC_MultiModeDMARequestAfterLas
tTransferCmd(ENABLE);
 /* Включить АЦП1 */
 ADC_Cmd(ADC1, ENABLE);
```

Для передачи результатов преобразования АЦП в память МК используется прямой доступ к памяти. Контроллер ПДП обеспечивает высокоскоростную передачу данных между периферийными устройствами и памятью, между памятью и памятью. Данные ПДП могут быстро перемещаться без использования ЦП, что сохраняет ресурсы ЦП свободными для других операций. Контроллер ПДП комбинирует архитектуру передачи данных, чтобы оптимизировать пропускную способность системы. В контроллерах DMA1 и DMA2 реализовано 16 потоков (8 для каждого контроллера). Каждый выделенный канал управления доступом к памяти запрашивает периферийное устройство. Каждый поток может иметь до 8 каналов (запросов), и у каждого есть арбитр для того, чтобы обработать приоритет запросов ПДП.

Плата тестировалась с различными настройками аналого-цифрового преобразования. АЦП контроллера был настроен на двойное чередование каналов с задержкой на преобразование, равной 5 циклам. Данные АЦП поочередно сохранялись в одной переменной, чем достигалась максимальная частота преобразования. Однако для использования осциллографа необходимо ввести регулируемые задержки при формировании буфера отображения, чтобы показать сигнал во всём диапазоне частот. Поэтому было введено управление АЦП по таймеру, что привело к небольшим задержкам по времени дискретизации. Процесс пересылки данных из АЦП через ПДП ожидает разрешения от таймера, задержка таймера при этом нулевая (несколько машинных тактов, которые проверяют время, равное нулю, уже вызывают задержку).

На рисунке 5 показана осциллограмма сигнала частотой 100 кГц и амплитудой 1,3 В. Данные АЦП сохранялись в одной переменной поочередно, с нулевой программной задержкой. Этот вариант настройки АЦП с регулируемой задержкой формирования буфера отображения представляется оптимальным для просмотра сигналов в диапазоне частот до 200 кГц. Максимальная частота сигнала, доступная для наблюдения на экране, составила 400 кГц (см. рис. 6).

Для отображения сигналов низкой частоты (100 Гц и ниже) необходимо вводить большие задержки при формировании буфера отображения. Поэтому формирование буфера перенесено в функцию прерывания по таймеру и введена небольшая регулируемая задержка. На дисплее это отображается в виде режима Md = 1 (см. дополнительные материалы к статье на сайте).

Заключение

Плата STM32F429IDiscovery удобна для программирования и является одной из самых производительных в своём классе. Тестовая задача осциллографирования показала хорошую работу контроллера дисплея и графического ускорителя DMA2D. В микроконтроллер интегрирован не самый быстрый АЦП – максимальная частота, которую удалось наблюдать на экране дисплея, составила 400 кГц. Но различные режимы конфигурации позволяют настроить АЦП для разнообразных задач наблюдения сигналов.

Литература

1. Описание платы STM32F429IDiscovery. http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090.
2. Примеры с исходными кодами для платы STM32F429IDiscovery. http://www.st.com/web/catalog/tools/FM116/SC959/SS1532/PF259090.
3. Утилита программирования STM32 ST-LINK Utility. http://www.st.com/web/catalog/tools/FM146/CL1984/SC720/SS1454/PF219866?sc=internet/evalboard/product/219866.jsp.
4. *Вячеслав Гавриков.* Мастера графики: новое поколение STM32F4 с поддержкой контроллера TFT. http://www.compel.ru/lib/ne/2013/8/6-mastera-grafiki-novoe-pokolenie-stm32f4-s-podderzhkoy-kontrollera-tft/.