ВЫСОКОЧАСТОТНЫЙ ГЕНЕРАТОР НА ОСНОВЕ СИНТЕЗАТОРА AD9851 С ИНДИКАТОРОМ ЧАСТОТЫ СИГНАЛА

Кулаков Владимир Геннадьевич
SPIN РИНЦ: 2111-7702
Контакт с автором: kulakovvlge@gmail.com

Цифровой синтезатор AD9851, выпускаемый корпорацией Analog Devices, в настоящее время является, вероятно, самым высокопроизводительным синтезатором из тех, что используются совместно с контроллером Arduino. Он способен формировать сигналы синусоидальной и прямоугольной формы с частотой от 0 до 72 МГц, а частота его тактового сигнала может достигать 180 МГц.

Частота и фаза выходного сигнала AD9851 являются программируемыми: программирование осуществляется с внешнего микроконтроллера либо по параллельному, либо по последовательному интерфейсу. Так как количество выходных линий у младших моделей Arduino, например Arduino Nano, невелико, при работе с контроллером Arduino удобнее использовать последовательный интерфейс.

На рисунке 1 в качестве примера показана схема генератора синусоидального сигнала, построенного на основе микросхемы AD9851BRSZ и контроллера Arduino. Частота сигнала на выходе генератора определяется информацией, поступающей с инкрементального энкодера на контроллер Arduino, программой, загруженной в этот контроллер, а также частотой тактового сигнала, подаваемого на микросхему DA2 синтезатора сигнала AD9851 с кварцевого генератора DD1.

В данном примере используется тактовый сигнал с частотой 160 МГц. Код K выходной частоты F для синтезатора AD9851 в таком случае вычисляется по следующей формуле:

$$K = F \times 2^{32} / 160000000.$$

Результат вычислений должен быть округлен до 32-разрядного целого числа без знака, а затем загружен в регистр синтезатора, хранящий код частоты.

Современные цифровые осциллографы обычно оснащены встроенным частотомером, но для повышения удобства работы с генератором его желательно оснастить собственным встроенным индикатором частоты выходного сигнала. В схеме, показанной на рисунке 1, в качестве индикатора используется самый простой из возможных вариантов – четырехзначный светодиодный дисплей на основе микросхемы TM1637 (Grove – 4-Digit Display).

Напряжение питания +5В, необходимое для работы кварцевого генератора и синтезатора, формирует стабилизатор DA1 (отдельный стабилизатор в данном случае необходим, так как при высокой частоте тактового сигнала синтезатор AD9851 потребляет ток силой около 100 мА), а питание индикатора обеспечивает стабилизатор, встроенный в контроллер Arduino.

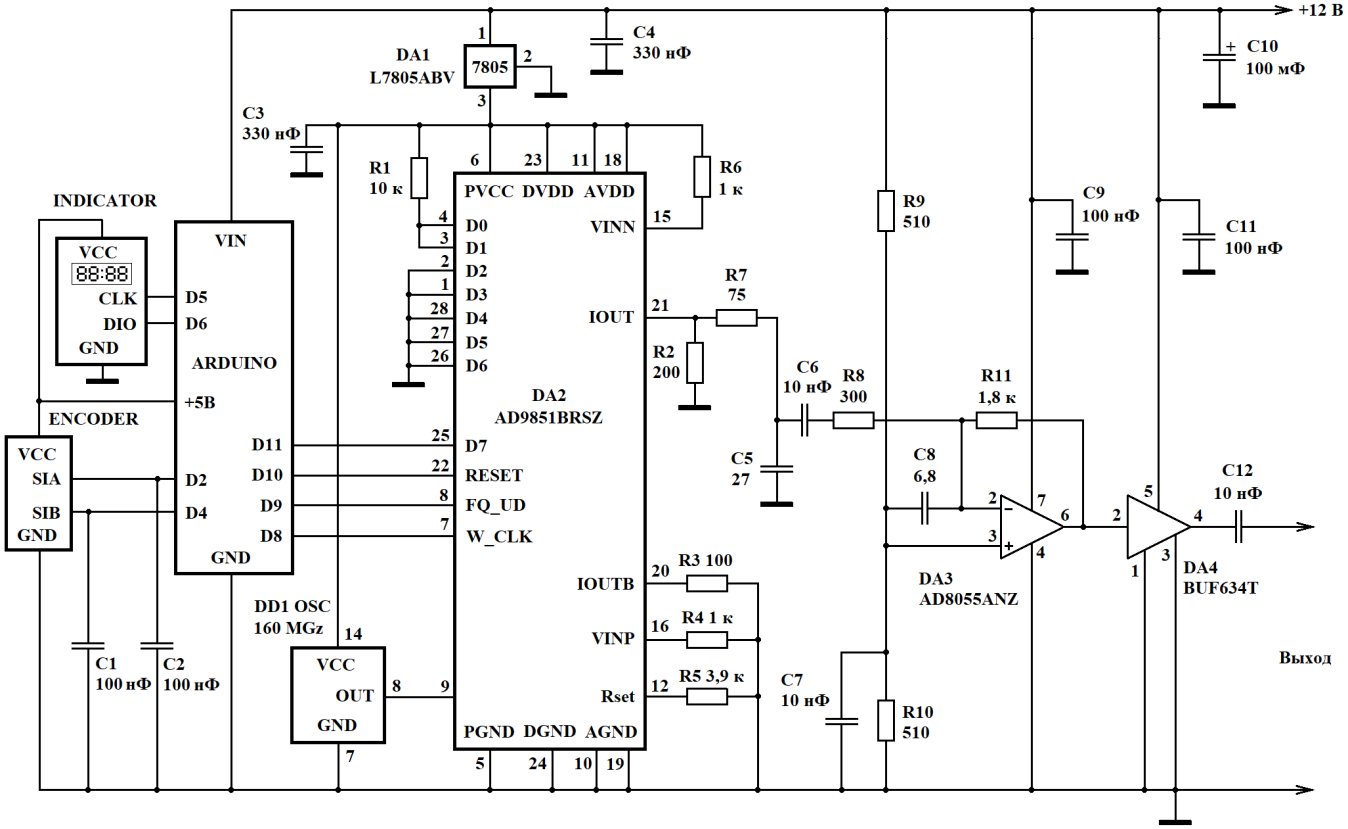

Рисунок 1. Схема высокочастотного генератора на микросхеме AD9851 с контроллером Arduino, энкодером и четырехзначным светодиодным индикатором частоты выходного сигнала

Здесь следует также отметить, что синтезатор AD9851 имеет встроенный умножитель частоты тактового генератора с фиксированным коэффициентом умножения 6, что позволяет использовать совместно с данным синтезатором не только высокочастотные, но и низкочастотные тактовые генераторы (с выходной частотой до 30 МГц). Однако использование данной возможности приводит к появлению дополнительных паразитных высокочастотных составляющих в выходном сигнале, что порождает потребность в применении более сложного фильтра на выходе синтезатора, поэтому в приведенном примере умножитель частоты не используется.

Пример программы, позволяющей перестраивать частоту генератора от 1 до 40 МГц с шагом 50 кГц, приведен в листинге 1. При повороте ручки энкодера в ту или иную сторону генерируется прерывание и контроллер либо увеличивает, либо уменьшает значение переменной-счетчика положения энкодера Position. После любого изменения значения данной переменной контроллер заново вычисляет код частоты и загружает его в микросхему AD9851, а также передает на цифровой индикатор код нового значения выходной частоты: две старшие цифры соответствуют частоте в мегагерцах, а две младшие – сотням и десяткам килогерц.

Листинг 1. Программа, предназначенная для управления работой цифрового синтезатора сигналов AD9851 и светодиодного индикатора.

```
#include <Arduino.h>
#include "TM1637.h"

#define pin_CLK  2 // Энкодер: пин A
#define pin_DT   4 // Энкодер: пин B
#define CLK      5 // Индикатор: CLK
#define DIO      6 // Индикатор: DIO
#define W_CLK    8 // AD9851: W_CLK
#define FQ_UD    9 // AD9851: FQ_UD
#define RESET   10 // AD9851: RESET
#define DATA    11 // AD9851: Serial data
#define PosLim 780 // Ограничитель изменения положения энкодера

const float fgen = 160.0E6; // Частота сигнала ГТИ, Гц
const float sf = 1.0E6;     // Начальное значение выходной частоты, Гц
const float df = 50000.0;   // Шаг приращения частоты, Гц
volatile long Position = 0; // Текущее значение положения энкодера
long oldPosition = 0;       // Предыдущее положение энкодера
float ff;        // Код частоты в формате с плавающей точкой
unsigned long f; // Двоичный код выходной частоты

// Процедура для обработки прерывания от энкодера
void EncoderRotate() {
  if (digitalRead(pin_CLK) == digitalRead(pin_DT)) {
    if(Position < PosLim) Position++;
  } else {if(Position > 0) Position--;}
}

// Процедура инициализации AD9851
void AD9851init(){
  digitalWrite(RESET, HIGH);
  digitalWrite(RESET, LOW);
  digitalWrite(W_CLK, HIGH);
  digitalWrite(W_CLK, LOW);
  digitalWrite(FQ_UD, HIGH);
  digitalWrite(FQ_UD, LOW);
}

// Процедура для передачи кода частоты синтезатору AD9851
// по последовательному интерфейсу
// Параметры:
// Freq - двоичный код частоты выходного сигнала,
// Cont - код управления работой синтезатора.
void AD9851setfreq(unsigned long Freq, byte Cont){
  int i;
  unsigned long d;
  byte u;
  d = Freq;
  u = Cont;
```

```
  // Загрузка кода частоты
  for(i=0; i<32; i++)
  {
    if(d & 1 != 0) digitalWrite(DATA, HIGH);
    else digitalWrite(DATA, LOW);
    digitalWrite(W_CLK, HIGH);
    digitalWrite(W_CLK, LOW);
    d = d >> 1;
  }
  // Загрузка управляющего байта
  for(i=0; i<8; i++)
  {
    if(u & 1 != 0) digitalWrite(DATA, HIGH);
    else digitalWrite(DATA, LOW);
    digitalWrite(W_CLK, HIGH);
    digitalWrite(W_CLK, LOW);
    u = u >> 1;
  }
  digitalWrite(DATA, LOW);
  // Завершение загрузки данных
  digitalWrite(FQ_UD, HIGH);
  digitalWrite(FQ_UD, LOW);
}

// Задаем номера линий, к которым подключен индикатор
TM1637 tm1637(CLK,DIO);

// Процедура для отображения частоты на индикаторе
void ShFreq() {
  long p;
  int8_t dig;
  // Умножаем позицию на шаг и прибавляем нач. значение частоты
  p = Position*((long)df/10000l) + (long)sf/10000l;
  // Выводим на индикатор старшую цифру (десятки МГц)
  dig = (int8_t)(p%10);
  tm1637.display(3,dig);
  p = p/10;
  // Выводим на индикатор вторую цифру (единицы МГц)
  dig = (int8_t)(p%10);
  tm1637.display(2,dig);
  p = p/10;
  // Выводим на индикатор третью цифру (сотни кГц)
  dig = (int8_t)(p%10);
  tm1637.display(1,dig);
  p = p/10;
  // Выводим на индикатор младшую цифру (десятки кГц)
  dig = (int8_t)p;
  tm1637.display(0,dig);
  // Включить на индикаторе двоеточие - разделитель
  tm1637.point(POINT_ON);
}
```

```
// Инициализируем контроллер Arduino и индикатор
void setup() {
  int i;
  // Определяем, какие линии станут выходными, а какие - входными
  pinMode(W_CLK, OUTPUT);
  pinMode(FQ_UD, OUTPUT);
  pinMode(DATA, OUTPUT);
  pinMode(RESET, OUTPUT);
  pinMode(pin_CLK, INPUT);
  pinMode(pin_DT, INPUT);
  // Инициализируем индикатор
  tm1637.init();
  tm1637.set(BRIGHT_TYPICAL);
  ShFreq();

  // Вычисляем начальный код частоты
  // в формате с плавающей точкой
  ff = sf*(float)0x10000l*(float)0x10000l/fgen;
  // Получаем двоичный код выходной частоты
  f = (unsigned long)ff;
  // Инициализируем микросхему AD9851
  AD9851init();
  // Загружаем код начального значения частоты, а встроенный
  // умножитель частоты ГТИ отключаем
  AD9851setfreq(f, 0);

  // Подключаем обработчик прерывания энкодера
  attachInterrupt(digitalPinToInterrupt(pin_CLK),
    EncoderRotate, RISING);
}

// Рабочий цикл контроллера Arduino
void loop() {
  int8_t dig;
  long p;
  if (oldPosition != Position)
  {
    // Запомнаем текущую позицию
    oldPosition = Position;
    // Отображаем новое значение частоты на индикаторе
    ShFreq();
    // Вычисляем код частоты в формате с плавающей точкой
    ff = (sf + df*(float)Position)*
      (float)0x10000l*(float)0x10000l/fgen;
    // Получаем двоичный код выходной частоты
    f = (unsigned long)ff;
    // Загружаем код частоты в синтезатор
    AD9851setfreq(f, 0);
  }
}
```

В диапазоне от 1 до 30 МГц амплитуда выходного сигнала на активной нагрузке с сопротивлением 50 Ом составляет около 3 В. Форма сигнала на выходе генератора при частоте колебаний 10 МГц показана на рисунке 2.

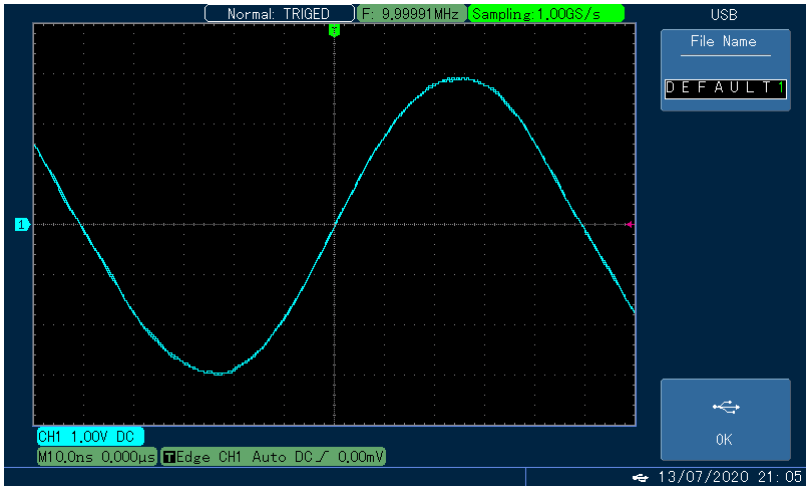

Рисунок 2. Осциллограмма сигнала на выходе генератора при частоте 10 МГц

Теоретически, синтезатор AD9851 способен вырабатывать сигнал с частотой, составляющей до половины от частоты тактового сигнала, поступающего от кварцевого генератора. Однако в том случае, когда частота выходного сигнала синтезатора превышает четверть тактовой частоты, становится заметной паразитная модуляция выходного сигнала, поэтому в данном примере выходная частота генератора ограничена значением 40 МГц.

При увеличении частоты до 40 МГц амплитуда сигнала уменьшается до 1,6 В (рисунок 3).

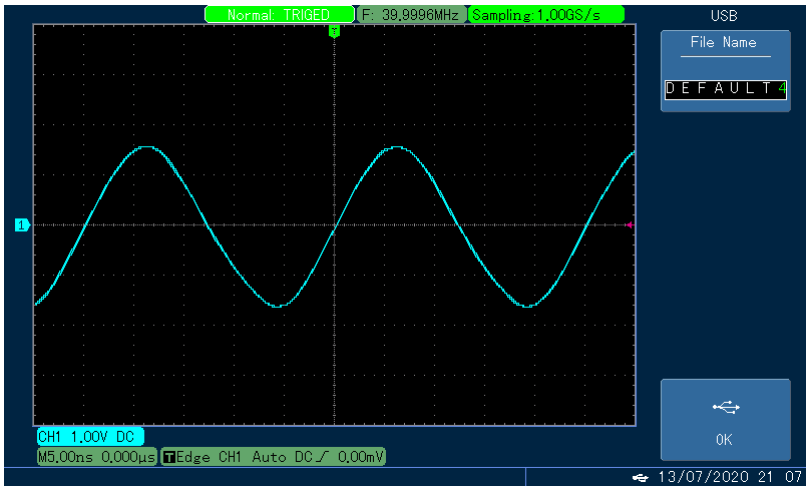

Рисунок 3. Осциллограмма сигнала на выходе генератора при частоте 40 МГц

Микросхема стабилизатора питания DA1 и микросхема буферного усилителя DA4 должны быть оснащены радиаторами воздушного охлаждения площадью не менее 4 см^2.

Список использованной литературы

1. CMOS 180 MHz DDS/DAC Synthesizer AD9851, Rev. D – Analog Devices, Inc., 2004.
2. Low Cost, 300 MHz Voltage Feedback Amplifiers AD8055/AD8056, Rev. E – Analog Devices, Inc., 2001.
3. BUF634 250-mA High-Speed Buffer – Texas Instruments Incorporated, 2019.
4. Кулаков В.Г. Высокочастотный генератор на микросхеме AD9833 с многозвенным RC-фильтром. [Электронный ресурс]. URL: http://new-idea.kulichki.net/pubfiles/200820074529.pdf (дата обращения: 20.08.2020).
5. Кулаков В.Г. Перестраиваемый высокочастотный генератор на основе синтезатора AD9850 и контроллера Arduino. [Электронный ресурс]. URL: http://new-idea.kulichki.net/pubfiles/210720133714.pdf (дата обращения: 20.07.2021).