

Измерение переменного напряжения и тока с помощью ADE7758 и STM32F103

В.А. Жмудь, А.И. Незванов, В.Г. Трубин

Новосибирский государственный технический университет, Новосибирск, Россия

Аннотация: В данном материале рассматриваются задачи, довольно часто возникающие при разработке электронных устройств, взаимодействующих с бытовой электрической сетью - измерение напряжения и потребляемого тока. Эти данные могут понадобиться для широкого круга задач. Например, становится возможным определение отсутствия одной или нескольких фаз, наличие перегрузок питающей сети и неравномерное распределение нагрузки по фазам. Также становится доступным наблюдение за потреблением энергии в течение определённого промежутка времени. В данном материале рассмотрены основные характеристики переменного напряжения, информация о подключении схемы-измерителя к микроконтроллеру, подключение трансформатора тока и реализация гальванической развязки микроконтроллера от сети. Также приведён исходный код для микроконтроллера STM32F103, алгоритм перевода значений регистров ADE7758 к пригодным к интерпретации величинам.

Ключевые слова: переменное напряжение и ток, действующее и амплитудное значение напряжения, синусоида, трансформатор тока, гальваническая развязка, источник питания постоянного тока, SCT-013, ADE7758, ADuM1201, ADuM1200, STM32F103.

ВВЕДЕНИЕ

В процессе разработки электронных устройств часто требуется добавить возможность анализа параметров сети переменного тока. Но это довольно непростая задача ввиду множества нюансов: специфика переменного тока, точность измерений и, главное, электробезопасность. Исходя из этого использование готовой микросхемы предпочтительно, так как производитель продумал все нюансы и требуется лишь правильно подключить её к микроконтроллеру. В статье рассмотрены вопросы, возникающие при создании таких устройств: основная информация о переменном токе, выбор и подключение микросхемы к сети, микроконтроллеру (МК), использование трансформатора тока и измерение значений тока и напряжения в бытовой сети.

1. Характеристики переменного тока

Переменный ток (*Alternating Current - AC*) [1] — электрический ток, который с течением времени изменяется по величине и направлению. В отличие от постоянного тока, график которого в зависимости от времени представляет собой прямую, переменный ток в бытовой сети имеет синусоидальную форму (Рис. 1). Линия из точек — действующее напряжение, прерывистая — амплитудное напряжение. Разницу между верхним и нижним амплитудным значением напряжения называют двойным амплитудным значением.

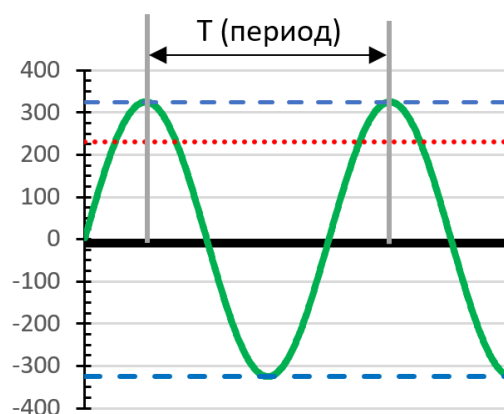


Рис. 1. График переменного тока

Частота переменного тока для бытового и промышленного использования в Российской Федерации — 50 Гц , то есть за 1 секунду ток меняет свое направление 50 раз (длительность периода — 20 мс). Частота в сети поддерживается с достаточно большой точностью — не хуже $\pm 0,2 \text{ Гц}$. Текущее значение частоты в энергосистеме России можно посмотреть на сайте системного оператора ЕЭС России [2].

Стандартное напряжение в однофазной электрической сети 230 В . Данное напряжение принято называть действующим (в иностранной документации — *RMS*). Проще говоря, нагревательный элемент будет выделять одинаковое количество тепла, независимо от того, включен ли он в сеть постоянного тока 220 В или в сеть переменного тока с действующим значением напряжения 220 В . При этом амплитудное напряжение в бытовой сети больше действующего и равно $230 \times \sqrt{2} \approx \pm 325 \text{ В}$.

Возможно, внимательный читатель заметил,

что местами мы указываем 220 В, а местами – 230 В. Это связано с тем, что в СССР, а затем РФ, было принято напряжение 220 В в бытовой сети (380 В между фазами в трёхфазной сети), а начиная с 2003 года ГОСТ 29322-92 [3] регламентировал переход РФ на 230/400 В.

2. Микросхема ADE7758

Для реализации в электронных устройствах возможности измерения переменного напряжения, тока и других параметров, возможно использование готовых микросхем, имеющих в своём составе все необходимые вычислительные и измерительные модули, освобождающие разработчика от необходимости в довольно трудоёмких расчётах на основном микроконтроллере (МК). Это несёт в себе несколько преимуществ:

- Высвобождение вычислительных ресурсов микроконтроллера;
- Повышение точности измерений, т.к. разработчик микросхем с большей вероятностью учтёт все аспекты и нюансы в работе с переменным током;
- Компактность разрабатываемого устройства – в готовой микросхеме все измерительные и вычислительные узлы собраны в одном компактном корпусе;
- Большой функционал, труднодостижимый для самостоятельно изготовленного измерительного устройства.

Исходя из вышеперечисленных преимуществ было принято решение об использовании готовой микросхемы ADE7758 [4]. Эта микросхема многофазного измерителя энергии с возможностью получения информации отдельно по каждой фазе. Данная микросхема выпускается достаточно давно и неплохо себя зарекомендовала. ADE7758 имеет следующие характеристики:

- Измерение (вычисление) действующего напряжения, тока, активной, реактивной мощности по трём фазам;
- Калибровка измерений с помощью программирования регистров;
- Интерфейс SPI для связи с ведущим МК;
- Питание – 5 Вольт постоянного тока, потребляемая мощность около 70 мВт.

Из вышеперечисленных характеристик

видно, что микросхема имеет широкие возможности. Для её подключения требуется небольшое количество внешних элементов – резисторы, конденсаторы и кварцевый резонатор. Схема подключения представлена на Рис. 2.

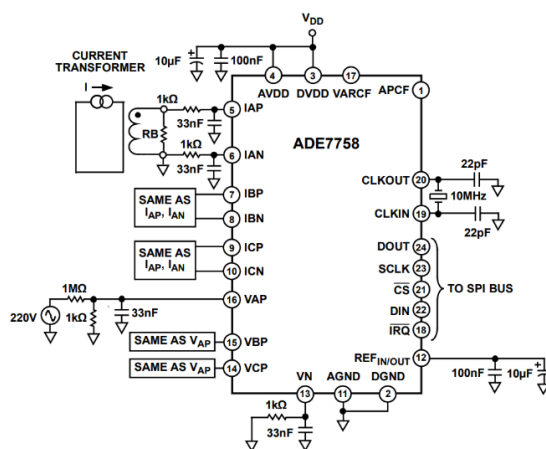


Рис. 2. Схема подключения ADE7758

Из представленной схемы видно, что измерение напряжения происходит с помощью делителя напряжения с коэффициентом 1000:1. То есть при напряжении в 230 В, на микросхему поступает всего 0,23 В, это действующее напряжение. При этом амплитудное напряжение, поступающее на вход микросхемы будет $0,23 \times \sqrt{2} = \pm 0,325$ В. Максимально допустимое напряжение на входах ADE7758 составляет $\pm 0,5$ В.

Питание микросхемы желательно обеспечить за счёт трансформаторного линейного источника питания, так как импульсные блоки питания (ИБП) при рассмотрении формы их выходного сигнала имеют некоторые пульсации напряжения (в качественных ИБП имеется хороший сглаживающий фильтр, но он не гарантирует отсутствие пульсаций ввиду особенностей работы схемы ИБП). ADE7758 анализирует параметры напряжения и тока с помощью АЦП, на работу которого пульсации напряжения питания могут повлиять, внося погрешность в измерения. Рекомендуемая схема источника питания представлена на Рис. 3.



Рис. 3. Блок-схема источника питания

Чтобы исключить сбой и возможность выхода МК и низковольтной части разрабатываемой схемы из строя, необходимо обеспечить гальваническую развязку. Хорошим

вариантом является ADuMI200/1201 [5]. Это восьмывыводная микросхема, которая имеет два однонаправленных канала либо два разнонаправленных. Допускается питание от 3 до 5 Вольт. Скорость передачи данных до

25 Мбит/с, что более чем достаточно для решаемой задачи.

3. Трансформатор тока SCT-013

Существует множество способов измерения тока в цепи, самые распространённые из которых:

- Токоизмерительный прецизионный шунт;
- Трансформатор тока;
- Датчики на основе эффекта Холла [6].

Измерение с помощью шунта является более точным, но имеет серьёзный недостаток – прямое соединение с измеряемой средой, что значительно повышает опасность поражения человека сетевым напряжением. Далее для измерения тока рассмотрим использование трансформатора тока. Это трансформатор, сердечник которого надевается на проводник, по которому протекает ток. Таким образом, измерительное устройство не имеет соединения с измеряемой средой, что обеспечивает дополнительную защиту.

Важно: необходимо отметить, что трансформатор тока ни в коем случае нельзя крепить к проводнику без подключения нагрузочного резистора («Burden» в иностранной литературе), иначе на выводах трансформатора возникает напряжение величиной до нескольких тысяч вольт, что может привести к выходу из строя схем и травмированию человека; необходимо припаять этот резистор непосредственно к выводам трансформатора для избежания опасных ситуаций.

Для измерения тока используем трансформатор тока YHDC SCT-013, который представлен на Рис. 4.



Рис. 4. Трансформатор тока YHDC SCT-013

Используемая модификация обеспечивает на выходе ток 50 мА при 100 А на измеряемом проводнике, то есть коэффициент трансформации 2000:1. Согласно найденной документации, точность данного трансформатора около 3%, что достаточно в бытовом использовании.

Нагрузочный резистор необходимо припаять

к выводам, как показано на Рис. 5. Номинал резистора определяется согласно требуемому напряжению на выходе. Микросхема ADE7758 допускает на входах измерения тока напряжение в диапазоне ±0,5 В, трансформатор выдаёт ток номиналом 50 мА. Таким образом, согласно закону Ома $R = U/I = 0.5/0.05 = 10 \text{ Ом}$. Желательно использовать проволочный резистор ввиду его большей механической надёжности и большей перегрузочной способности.

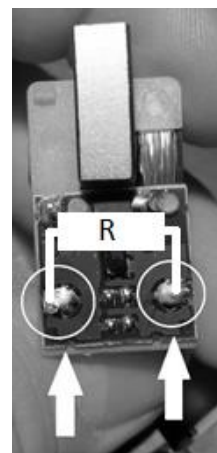


Рис. 5. Трансформатор тока YHDC SCT-013

4. Подключение к микроконтроллеру

Подключение к микроконтроллеру осуществляется с помощью микросхем ADuM1200/1201, которые обеспечивают гальваническую развязку микроконтроллера от бытовой электрической сети. В качестве трансформатора, питающего микросхему ADE7758, использован герметичный трансформатор HAHN BV EI 302 3021 (230/7.5 Вольт, 1.8 ВА), диодный мост рассчитан на 2 Ампера (2W10M), линейный стабилизатор – L7805. Для защиты схемы от короткого замыкания трансформатора установлен плавкий предохранитель на 1 Ампер.

Схема подключения и номиналы элементов указаны на Рис. 6.

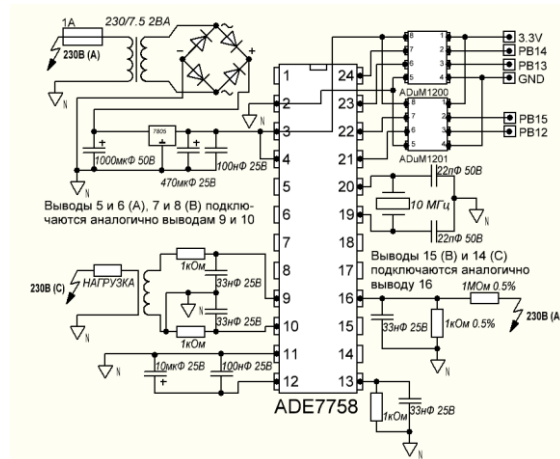


Рис. 6. Принципиальная схема подключения

ADE7758 к STM32F103.

Обмен данными между микроконтроллером и ADE7758 происходит по интерфейсу SPI. Для получения данных, например, о напряжении фазы A, необходимо передать ADE7758 восемь бит адреса регистра (0x0D). После передачи адреса необходимо передать три нулевых байта, при этом в ответ ADE7758 передаст данные о напряжении фазы A (три байта, начиная со старшего). Подобным образом производится чтение любых 24-х битных регистров. Более подробно об этом можно прочитать в документации на ADE7758 [4, раздел «Timing diagrams»].

Отдельно необходимо определиться с «расшифровкой» значений – при чтении там не будет ожидаемых вольт, ампер и т.д. Каждый регистр содержит число, которое необходимо перевести к понятному виду согласно нескольких условий. Разберём эти условия для значений регистров тока и напряжения.

Регистр AIRMS (0x0A) возвращает значение тока. Схема расчёта представлена на Рис. 7.

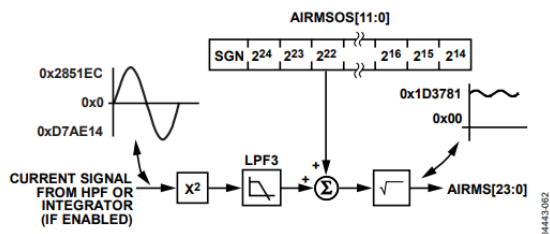


Рис. 7. Схема расчёта тока в ADE7758

Из схемы расчёта видно, что сигнал с трансформатора тока после обработки записывается в регистр AIRMS, диапазон значений которого лежит от 0 до 0x1D3781 (1 914 753) при максимальном токе (на входе с трансформатора тока ±0,5 В). То есть, при токе в 100 Ампер в регистр будет записано значение 1 914 753. Таким образом,

$$I = AIRMS / 1914753 \times 100 \text{ A.}$$

Регистр AVRMS (0x0D) возвращает значение напряжения. Схема расчёта представлена на

Рис. 8.

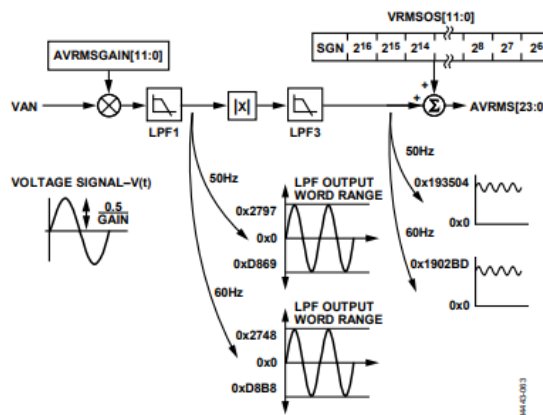


Рис. 8. Схема расчёта напряжения в ADE7758

Из схемы расчёта видно, что сигнал с делителя напряжения (1 МОм: 1 КОм = 1000:1) обрабатывается ADE7758 и записывается в регистр AVRMS, диапазон значений которого для сети с частотой 50 Гц лежит от 0 до 0x193504 (1 651 972) при напряжении на входе VAP ±0,5 В. То есть, при амплитудном напряжении 500 В (500 В/1000 = 0,5 В) в регистр будет записано значение 1 651 972. Но отдельно стоит отметить, что в регистре AVRMS содержится амплитудное значение напряжения и его необходимо привести к действующему. Таким образом,

$$U = (AVRMS \times 500) / (1651972 \times \sqrt{2}) \approx \approx AVRMS / 4672 \text{ В.}$$

Отдельно необходимо отметить, что в программном коде используется библиотека *xsprintf* [7], которая является компактным инструментом для микроконтроллеров, имеющих небольшой объём памяти. Данная библиотека предоставляет аналог функции *sprintf*, с помощью которой удобно приводить числовые значения к строковому виду и форматировать строки.

5. ПРОГРАММНЫЙ КОД

Программный код приведен ниже.


```

#include <stm32f10x.h>
#include "xprintf/xprintf.c"
#define Blue GPIO_Pin_13
#define LED_OFF() GPIO_SetBits(GPIOC, GPIO_Pin_13)
#define LED_ON() GPIO_ResetBits(GPIOC, GPIO_Pin_13)
GPIO_InitTypeDef Init_PORTB, Init_PORTA, Init_PORTC;
USART_InitTypeDef USART_InitStructure;
SPI_InitTypeDef SPI_InitStructure;
NVIC_InitTypeDef NVIC_InitStructure;
TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
ErrorStatus HSEStartUpStatus;
volatile uint32_t timer = 0;

void Tx(uint8_t Symbol) {
while (USART_GetFlagStatus(USART1, USART_FLAG_TC) == RESET) {};
USART_SendData(USART1, Symbol);
}
void TxStr(uint8_t *pStr) {
static uint8_t i; i = 0; do Tx( pStr[i++] ); while (pStr[i] != 0);
}
void SetSysClockToHSE(void){
RCC_DeInit(); RCC_HSEConfig(RCC_HSE_ON); HSEStartUpStatus = RCC_WaitForHSEStartUp();
if (HSEStartUpStatus == SUCCESS){
FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable); FLASH_SetLatency(FLASH_Latency_2);
RCC_HCLKConfig(RCC_SYSCLK_Div1); RCC_PCLK2Config(RCC_HCLK_Div1);
RCC_PCLK1Config(RCC_HCLK_Div2); RCC_PLLConfig(RCC_PLLSource_HSE_Div1,RCC_PLLMul_9);
RCC_SYSClkConfig(RCC_SYSCLKSource_HSE); RCC_PLLCmd(ENABLE);
while (RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET) {};
RCC_SYSClkConfig(RCC_SYSCLKSource_PLLCLK); while(RCC_GetSYSCLKSource() != 0x08) {};
} else while (1) {};
}
void TIM2_IRQHandler(void){
if (TIM_GetITStatus(TIM2, TIM_IT_Update) == SET) {
TIM_ClearITPendingBit(TIM2, TIM_IT_Update); timer++;
}
}
void delay_ms(uint32_t time) { timer = 0; while(time > timer); }
void chip_on(void) { GPIO_ResetBits(GPIOA, GPIO_Pin_4); }
void chip_off(void) { GPIO_SetBits(GPIOA, GPIO_Pin_4); }

void ADE7758WriteReg8(uint8_t Addr, uint8_t Data) {
static uint8_t Xtmp;
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, Addr);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
Xtmp = SPI_I2S_ReceiveData(SPI1);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, Data);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
Xtmp = SPI_I2S_ReceiveData(SPI1);
}
void ADE7758ReadReg24(uint8_t Addr, uint8_t type) {
static uint8_t Xtmp, X0, X1, X2; static int32_t Rez = 0; char * mass[20];
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, Addr);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
Xtmp = SPI_I2S_ReceiveData(SPI1);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, 0);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
X2 = SPI_I2S_ReceiveData(SPI1);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, 0);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
X1 = SPI_I2S_ReceiveData(SPI1);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_TXE) == RESET); SPI_I2S_SendData(SPI1, 0);
while (SPI_I2S_GetFlagStatus(SPI1, SPI_I2S_FLAG_RXNE) == RESET);
X0 = SPI_I2S_ReceiveData(SPI1);
Rez = X2; Rez <<=8; Rez |= X1; Rez <<=8; Rez |= X0;
if(type == 2) { // current
uint16_t res = Rez/(19148); xsprintf(mass,"%d Amperes. ",res); TxStr(mass);
} else if(type == 1) { //voltage
uint16_t res = (Rez/4672); xsprintf(mass,"%d Volts. ",res); TxStr(mass);
}
}

int main(void) {
SetSysClockToHSE();
RCC_ADCLKConfig(RCC_PCLK2_Div2);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA | RCC_APB2Periph_GPIOB, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOC, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1, ENABLE);
RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2, ENABLE);
RCC_APB2PeriphClockCmd(RCC_APB2Periph_SPI1, ENABLE);
}

```

```

Init_PORTA.GPIO_Pin = (GPIO_Pin_4);//cs
Init_PORTA.GPIO_Mode = GPIO_Mode_Out_PP;
Init_PORTA.GPIO_Speed = GPIO_Speed_10MHz;
GPIO_Init(GPIOA, &Init_PORTA);
Init_PORTA.GPIO_Pin = GPIO_Pin_6 ; //miso
Init_PORTA.GPIO_Speed = GPIO_Speed_10MHz;
Init_PORTA.GPIO_Mode = GPIO_Mode_IPD;
GPIO_Init(GPIOA, &Init_PORTA);
Init_PORTA.GPIO_Pin = GPIO_Pin_5 | GPIO_Pin_7; //SPI sck/mosi/miso
Init_PORTA.GPIO_Speed = GPIO_Speed_10MHz;
Init_PORTA.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_Init(GPIOA, &Init_PORTA);
SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
SPI_InitStructure.SPI_CPHA = SPI_CPHA_2Edge;
SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_64;
SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB; SPI_Init(SPI1, &SPI_InitStructure);
SPI_Cmd(SPI1, ENABLE);
Init_PORTC.GPIO_Pin = GPIO_Pin_13;
Init_PORTC.GPIO_Mode = GPIO_Mode_Out_PP;
Init_PORTC.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOC, &Init_PORTC);
GPIO_SetBits(GPIOC, GPIO_Pin_13);
Init_PORTA.GPIO_Pin = GPIO_Pin_10; // USART1 Rx PA10
Init_PORTA.GPIO_Speed = GPIO_Speed_10MHz;
Init_PORTA.GPIO_Mode = GPIO_Mode_IPU;
GPIO_Init(GPIOA, &Init_PORTA);
Init_PORTA.GPIO_Pin = GPIO_Pin_9; // USART1 Tx PA9
Init_PORTA.GPIO_Speed = GPIO_Speed_10MHz;
Init_PORTA.GPIO_Mode = GPIO_Mode_AF_PP; // Alternate function push-pull
GPIO_Init(GPIOA, &Init_PORTA);
USART_InitStructure.USART_BaudRate = 9600;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = USART_StopBits_1;
USART_InitStructure.USART_Parity = USART_Parity_No;
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
USART_Init(USART1, &USART_InitStructure);
TIM_TimeBaseStructure.TIM_Period = 50; // 1 мсек.
TIM_TimeBaseStructure.TIM_Prescaler = 1440 - 1;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);
TIM_ARRPreloadConfig(TIM2, ENABLE);
NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
TIM_ITConfig(TIM2, TIM_IT_Update, ENABLE);
TIM_Cmd(TIM2, ENABLE); /* TIM enable counter */
USART_Cmd(USART1, ENABLE);chip_on();

TxStr("Start UART...\n\n\n");
// Производим настройку ADE7758
ADE7758WriteReg8(0x13, 4); // OPMODE
ADE7758WriteReg8(0x14, 0xFC); // MMODE
while (1) {
    LED_ON();
    ADE7758ReadReg24(0xD,1); ADE7758ReadReg24(0xE,1); ADE7758ReadReg24(0xF,1);
    TxStr("\r\n");
    ADE7758ReadReg24(0xA,2); ADE7758ReadReg24(0xB,2); ADE7758ReadReg24(0xC,2);
    TxStr("\r\n\r\n"); delay_ms(1000); LED_OFF(); delay_ms(1000);
}
}

```

6. Результаты испытаний

Для проверки работоспособности схемы проведём замеры по следующей методике:

- Подключим в разрыв питающего провода токовый шунт 75 ШИП1-10 (точность 0.5%, выдаёт напряжение 75 мВ при токе 10 А);
- Закрепим трансформатор тока на этот же провод и присоединим его к схеме с ADE7758;
- Последовательно снимем полученные данные для каждой из тестируемых

нагрузок.

Замеры представлены в таблице на *Рис 9*.

	Лампа 95 Вт	Бытовой фен 600 Вт	Бытовой фен 1200 Вт	Бытовой чайник 2300 Вт
Шунт 75 ШИП1-10	0.4	2.25	4.8	9.1
Трансформатор тока + ADE7758	0.38	2.28	4.85	9.05

Рис. 9. Результаты замеров, ток (А)

По результатам замеров можно отметить, что измерение тока с помощью трансформатора тока и ADE7758 имеет небольшое отклонение от замеров с помощью шунта. Таким образом, измерение тока и напряжения с помощью микросхемы способно удовлетворить большинство потребностей.

ЗАКЛЮЧЕНИЕ

1. Микросхема ADE7758 является недорогим и простым в применении средством для измерения параметров электрической сети.

2. С помощью ADE7758 можно достичь хорошей точности измерений.

3. Несмотря на множество возникающих вопросов при первом применении ADE7758 и достаточно сложной документации, использование базовых функций представляется достаточно простым.

4. При работе с сетевым напряжением необходимо соблюдать меры предосторожности, особенно при работе с трансформатором тока.

ЛИТЕРАТУРА

- [1] Wikipedia – Переменный ток. URL: https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%82%D0%BE%D0%BA

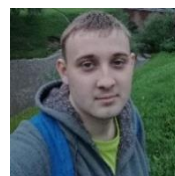
- [2] Системный оператор ЕЭС – Частота в ЕЭС России. URL: https://www.sops.ru/index.php?id=ees_freq&no_cache=1
- [3] ГОСТ 29322-92. Стандартные напряжения. URL: <http://docs.cntd.ru/document/gost-29322-92>
- [4] ADE7758 - Datasheet. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7758.pdf>
- [5] ADuM 1200/1201. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ADuM1200_1201.pdf
- [6] Использование датчика тока ACS712. URL: <https://www.rlocman.ru/shem/schematics.html?di=113339>
- [7] Библиотека XSPRINTF. URL: <http://elm-chan.org/fsw/strf/xprintf.html>



Вадим Аркадьевич Жмуд - заведующий кафедрой Автоматики НГТУ, профессор, доктор технических наук, заместитель генерального директора Новосибирского института программных систем по науке (ПАО «НИПС») E-mail: oao_nips@bk.ru 630073, Новосибирск, Просп. К. Маркса, д. 20 630090, Новосибирску, просп. Академика Лаврентьева, д 6/3



Виталий Геннадьевич Трубин – старший преподаватель кафедры Автоматики НГТУ, директор ООО «КБ Автоматика». E-mail: cpm@ait.cs.nstu.ru



Александр Игоревич Незванов – студент 1 курса магистратуры кафедры Автоматики НГТУ. E-mail: nezvanovml@mail.ru

Статья поступила 22.02.2019

Measurement of AC Voltage and Current with ADE7758 and STM32F103

V.A. Zhmud, A.I. Nezvanov, V.G. Trubin
Novosibirsk State Technical University, Novosibirsk, Russia

Abstract: This material deals with a problem that often occurs in the development of electronic devices interacting with the household electric network. For example, measurement of voltage and current consumption. This data may be needed for a wide range of purposes. For example, it becomes possible to determine the absence of one or more phases, the presence of power overloads, and the uneven distribution of load across phases. Observation of energy consumption for a certain period of time also becomes available. This material will discuss the main characteristics of AC voltage, information on connection of the measuring circuit to the microcontroller, connection of the current transformer and implementation of galvanic isolation of the microcontroller from the network. The full source code for the microcontroller STM32F103 will also be given, the scheme of translation of register values ADE7758 to the values to be interpreted.

Keywords: AC voltage and current, active and amplitude value, sinusoid, current transformer, galvanic isolation, linear type DC power supply, SCT-013, ADE7758, ADuM1201, ADuM1200, and STM32F103.

REFERENCES

- [1] Wikipedia – Alternating current. URL: https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D0%BC%D0%B5%D0%BD%D0%BD%D1%8B%D0%B9_%D1%82%D0%BE%D0%BA
- [2] System operator of energetic system – Frequency in Russia's electric system. URL: https://www.soups.ru/index.php?id=ees_freq&no_cache=1
- [3] GOST 29322-92. Standart voltages. URL: <http://docs.cntd.ru/document/gost-29322-92>
- [4] ADE7758 – Datasheet. URL: <https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7758.pdf>
- [5] ADuM 1200/1201. URL: https://www.analog.com/media/en/technical-documentation/data-sheets/ADuM1200_1201.pdf
- [6] Using of current sensor ACS712. URL: <https://www.rlocman.ru/shem/schematics.html?di=113339>
- [7] Library XSPRINTF. URL: <http://elm-chan.org/fsw/stf/xprintf.html>



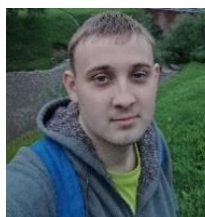
Vadim Zhmud – Head of the Department of Automation in NSTU, Professor, Doctor of Technical Sciences. Vice-director of Novosibirsk Institute of Program Systems

E-mail: oao_nips@bk.ru
630073, Novosibirsk,
str. Prosp. K. Marksa, h. 20
630090, Novosibirsk,
Pr. Ak. Lavrientievs, 6/1



Vitaliy G. Trubin – lecturer of Department of Automation, NSTU.
E-mail: cpm@ait.cs.nstu.ru

20 Prospekt K. Marksa, Novosibirsk,
630073



Alexander I. Nezvanov – student of Department of Automation, NSTU.
E-mail: nezvanovml@mail.ru

20 Prospekt K. Marksa, Novosibirsk,
630073

The paper has been received on 22/02/2019.