

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доц., канд. техн. наук		О.А. Кононов
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №8

по курсу: ОСНОВЫ МИКРОПРОЦЕССОРНОЙ ТЕХНИКИ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №	4711		Хасавнов Б.Р.
		подпись, дата	инициалы, фамилия

Санкт-Петербург 2020

Цель работы

Изучить работу по определению температуры микропроцессора при помощи встроенного датчика температуры, используя АЦП, на примере процессора STM32F407VG.

1 Теоретические сведения

По таблице 1 определяются коэффициенты для датчика температуры. Стоит заметить, что при настройке датчика их можно менять с целью подгонки температуры к истинной. Это делается для настройки датчика температуры [1-2].

Таблица 1 – Характеристики температурного датчика

Symbol	Parameter	Min	Typ	Max	Unit
$T_L^{(1)}$	V_{SENSE} linearity with temperature	-	± 1	± 2	$^{\circ}\text{C}$
Avg_Slope ⁽¹⁾	Average slope	-	2.5		mV/ $^{\circ}\text{C}$
$V_{25}^{(1)}$	Voltage at 25 $^{\circ}\text{C}$	-	0.76		V
$t_{START}^{(2)}$	Startup time	-	6	10	μs
$T_{S_temp}^{(2)}$	ADC sampling time when reading the temperature (1 $^{\circ}\text{C}$ accuracy)	10	-	-	μs

1. Guaranteed by characterization.

2. Guaranteed by design.

Для считывания температуры необходимо сделать следующее (что советует производитель)

Использовать датчик:

1. Выберите входной канал ADC1_IN16 или ADC1_IN18.
2. Выберите время выборки больше минимального времени выборки, указанного в спецификации.
3. Установите бит TSVREFE в регистре ADC_CCR, чтобы датчик температуры вышел из режима отключения питания.
4. Запустить преобразование АЦП путем установки бита SWSTART (или внешнего триггера).
5. Считайте результирующие данные VSENSE в регистре данных АЦП.

6. Рассчитайте температуру по следующей формуле:

$$\text{Температура (в } ^\circ\text{C)} = \{(V_{\text{SENSE}} - V_{25}) / \text{Avg_Slope}\} + 25$$

Где:

- V_{25} = значение V_{SENSE} для 25°C

- Avg_Slope = средний наклон температуры по отношению к кривой V_{SENSE} (задается в мВ/С или мкВ/С)

Действительные значения V_{25} и Avg_Slope см. в разделе "Электрические характеристики".

После выхода из режима энергосбережения датчик имеет время ввода в эксплуатацию, прежде чем он сможет выдать V_{SENSE} на правильном уровне. АЦП также имеет время запуска после включения питания, поэтому для минимизации задержки необходимо одновременно устанавливать биты ADON и TSVREFE.

Выходное напряжение термодатчика изменяется линейно с температурой. Смещение этой линейной функции зависит от каждой микросхемы из-за изменения процесса (до 45°C от одной микросхемы к другой) [2].

2 Листинги программы, написанной на языке программирования С

Во время выполнения работы использовалась программа STM32CubeMX, позволяющая сгенерировать проект, а затем экспортировать его в Keil для дальнейшего написания программы.

Процесс написания программы можно условно разбить на два этапа: настройка проекта в среде «CubeMX» и написание программы в среде разработки «Keil». На рисунках 1 – 3 представлены этапы настройки проекта в CubeMX.

В CubeMX производится настройка портов: PD15-PD12 – светодиоды, включение АЦП в качестве измерителя температуры (подключаем его к встроенному датчику температуры), RCC – подключение тактирования.

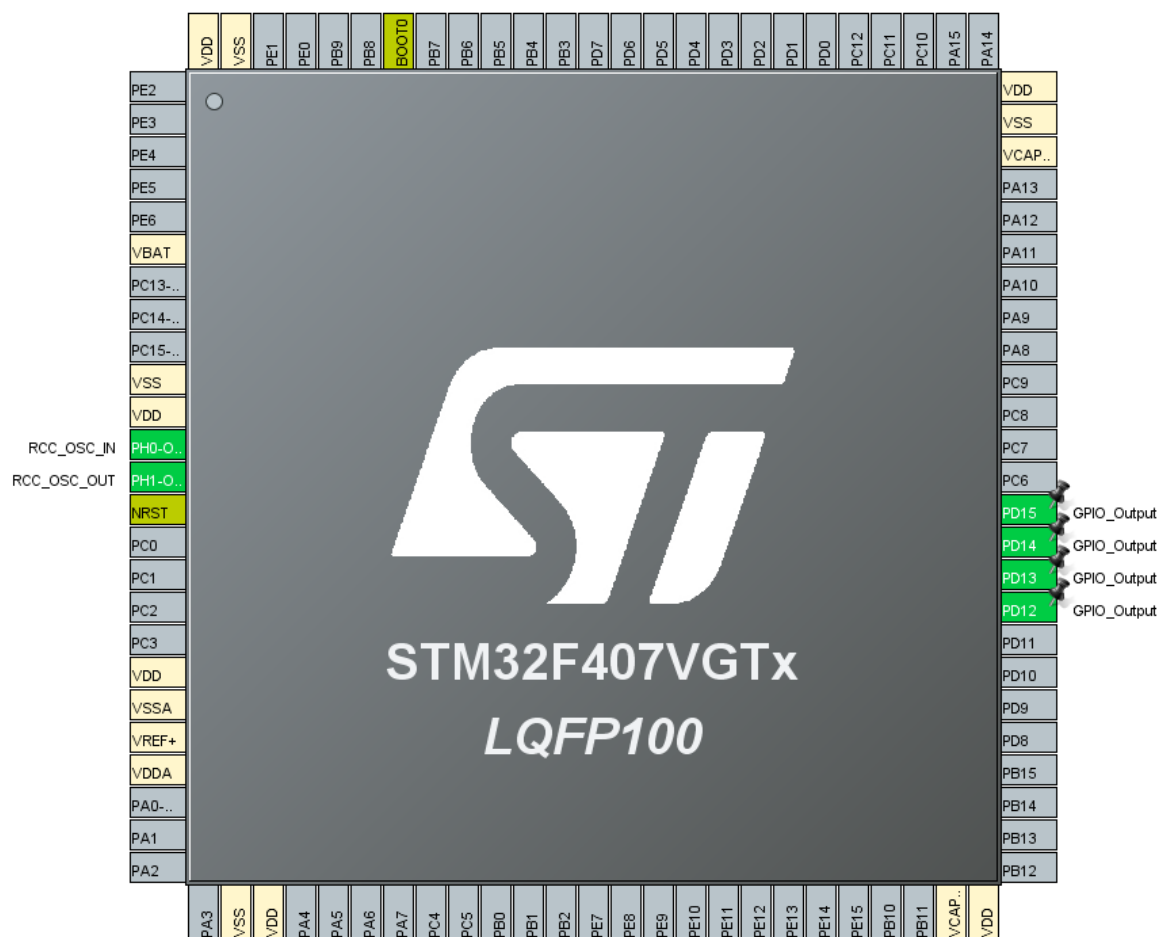


Рисунок 1 - Настройка STM32F407VG, Pinout View

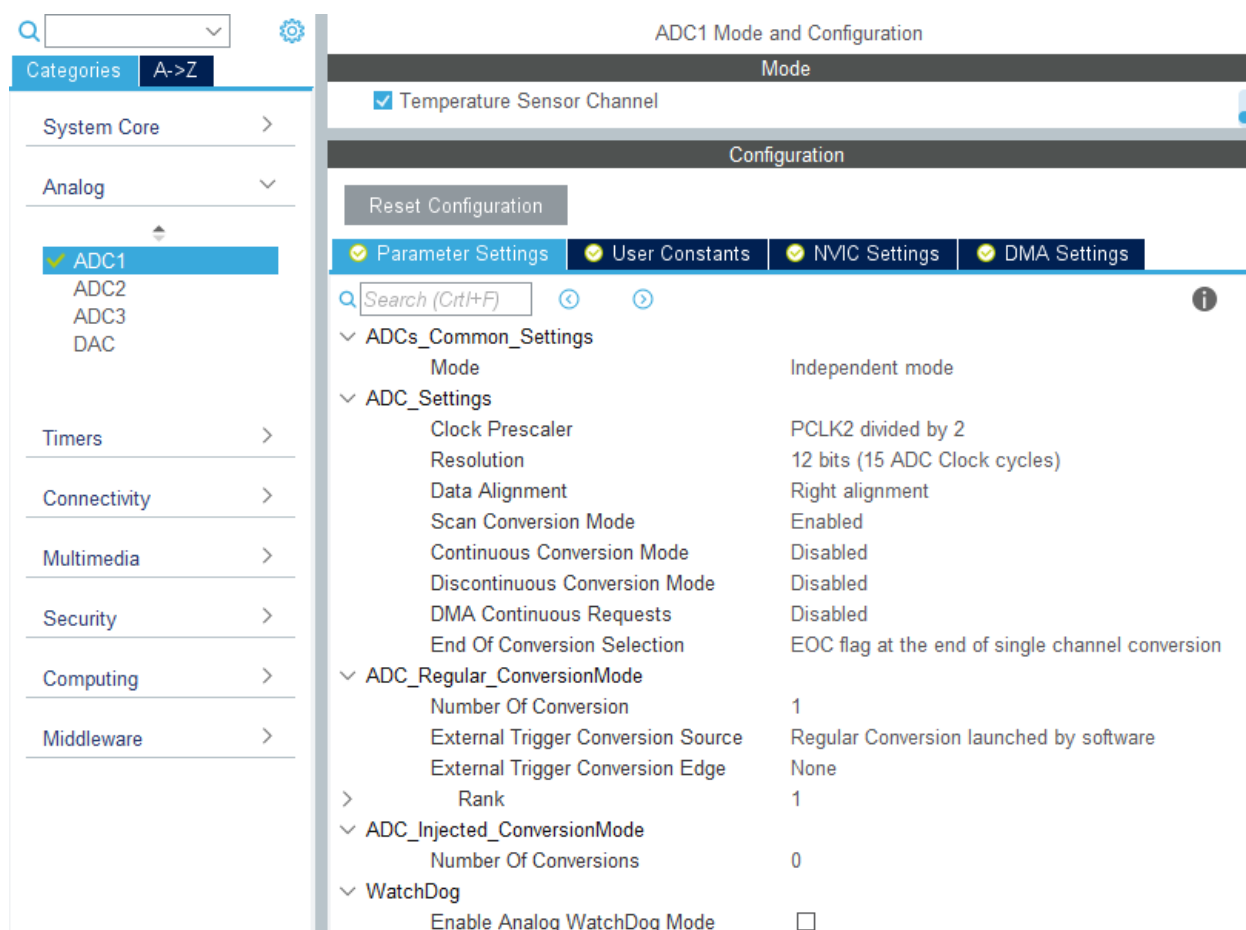


Рисунок 2 – Включение и настройка АЦП на работу с встроенным датчиком температуры

Далее мы должны настроить тактирование. Задаем входную частоту 8MHz. Так же при помощи делителей уменьшаем частоту на выводах – делается все это для уменьшения энергозатрат.

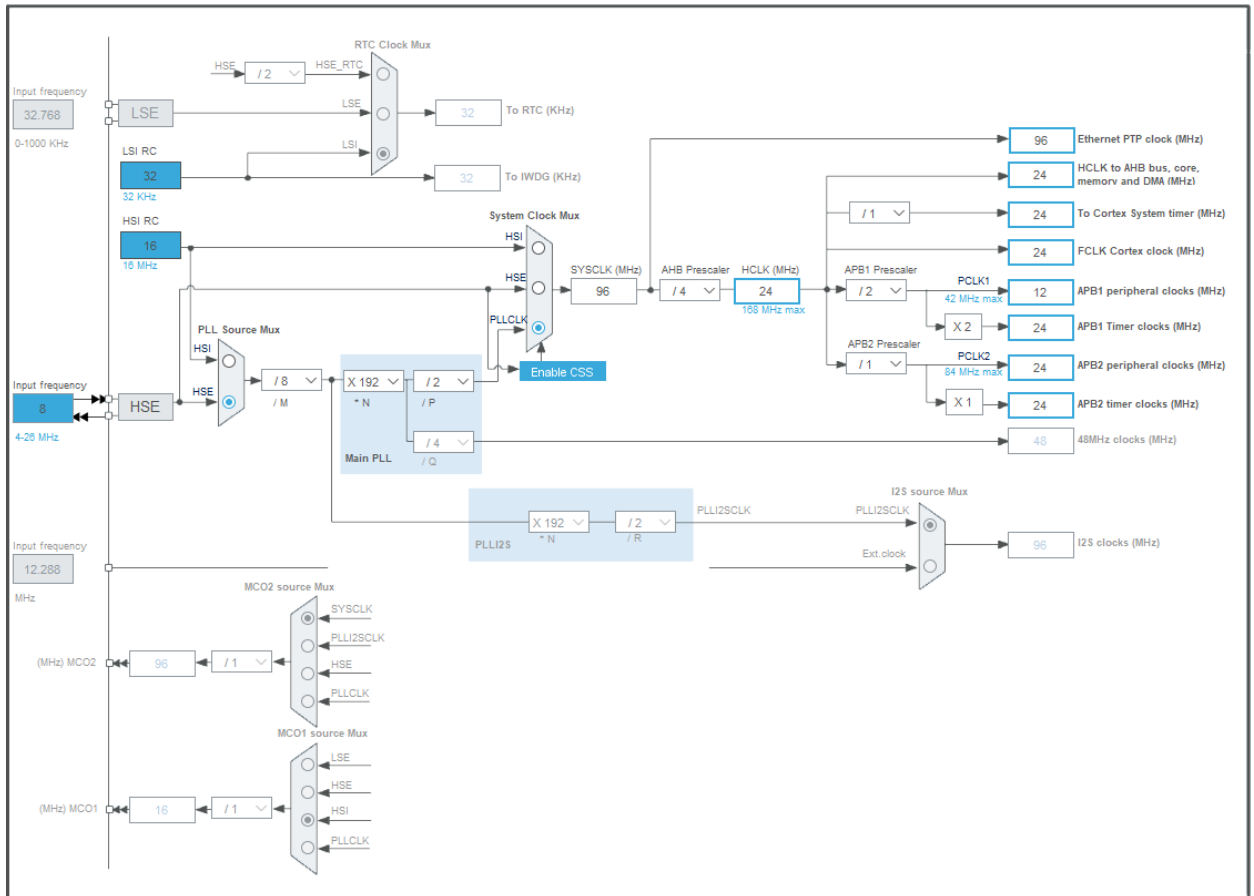


Рисунок 3 - Clock Configuration

Напишем же теперь программу:

```
#include "main.h"
```

```
#define tV_25 0.76f // Напряжение (в вольтах) на датчике при температуре 25 °C.
```

```
#define tSlope 0.0025f // Изменение напряжения (в вольтах) при изменении температуры на градус.
```

```
#define Vref 3.3f // Образцовое напряжение АЦП (в вольтах).
```

```
ADC_HandleTypeDef hadc1;
```

```
int16_t Result=0;
```

```
float temp;
```

```
void SystemClock_Config(void);
```

```

static void MX_GPIO_Init(void);
static void MX_ADC1_Init(void);

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_ADC1_Init();
    int y=0;
    while (1)
    {
        HAL_Delay(1000); // Задержка 1000 мс.
        HAL_ADC_Start(&hadc1); // Запуск АЦП
        if (HAL_ADC_PollForConversion(&hadc1, 100) == HAL_OK) // Ожидание
            завершения преобразования.
        {
            Result = HAL_ADC_GetValue(&hadc1); // Считывание с АЦП.
            temp = (float) Result/4096*Vref; // Напряжение в вольтах на датчике
            temp = ((temp - tV_25)/tSlope) + 20; // Температура в градусах
            if ( 20 > temp && temp > 10) { // условие для светодиода
                HAL_GPIO_WritePin( GPIOD, GPIO_PIN_15, GPIO_PIN_SET); // Включение
                светодиода
            }
            if ( 40 > temp && temp > 30 ){
                HAL_GPIO_WritePin( GPIOD, GPIO_PIN_12, GPIO_PIN_SET);
            }
        }
    }
}

```

```

}
if ( 60 > temp && temp > 50 ){
HAL_GPIO_WritePin( GPIOD, GPIO_PIN_13, GPIO_PIN_SET);
}
if ( temp > 60 ){
HAL_GPIO_WritePin( GPIOD, GPIO_PIN_14, GPIO_PIN_SET);
}
}
HAL_ADC_Stop(&hadc1); //Выключение АЦП
}
}

```

3 Результаты работы программы

Программа работает следующим образом: АЦП получает данные с датчика температуры и преобразует их в цифровой вид. Далее микроконтроллер включает светодиод в соответствии с условием определенным. При температуре равной от 10 до 20 градусов Цельсия должен загореться синий светодиод. При температуре равной от 30 до 40 градусов Цельсия должен загореться зеленый светодиод. При температуре равной от 50 до 60 градусов Цельсия должен загореться оранжевый светодиод. При температуре равной более 60 градусов Цельсия должен загореться красный светодиод. Данные значения были подобраны опытным путем, так как контроллер показывал температуру примерно равной 55 градусов Цельсия. Такая температура не является действительной, так как микроконтроллер при работе не был нагрет, в комнате было прохладно. Из-за отсутствия термометра узнать точную температуру в комнате не представляется возможным, из-за чего невозможно изменить коэффициенты «tV_25» и «tSlope». Данные коэффициенты настраиваются один раз для каждого микроконтроллера, после чего устройство будет показывать более точную температуру.

На рисунке 4 видно, что компиляция проекта была произведена корректно.

```
../Core/Src/main.c: 1 warning, 0 errors  
linking...  
Program Size: Code=4596 RO-data=440 RW-data=24 ZI-data=1096  
FromELF: creating hex file...  
"LR8delaism\LR8delaism.axf" - 0 Error(s), 1 Warning(s).  
Build Time Elapsed: 00:00:08
```

Рисунок 4 – Компиляция программы (build output)

Вывод

В результате выполнения лабораторной работы была изучена работа таймера по определению температуры микропроцессора при помощи встроенного датчика температуры, используя АЦП, на примере процессора STM32F407VG. Настройка проводилась при помощи среды разработки STM32 CubeMX.

Работа с платой STM32 показала, что программа написана верно – таймеры меняют яркость светодиодов. Определение температуры производится некорректно в виду отсутствия возможности настройки датчика температуры, из-за отсутствия возможности сравнить результат с эталонным термометром.

Стоит отметить, что полученный результат, примерно 55-56 градусов по Цельсию, нельзя принимать за Фаренгейты, переводя их в Цельсия и получая примерно 12-15 градусов. Такое рассуждение неверно. В технической литературе указано, что в результате измерений получается температура именно в Цельсиях.

Список источников

1 Техническая документация по STM32F405xx STM32F407xx/
STMicroelectronics STM32: 2020 – 203 с.

2 Reference manual/ STMicroelectronics STM32: 2019 – 1749 с.