УДК 004

Многозадачность в условиях однопоточности при разработке под AVR

Болтовский Гавриил Александрович
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Целью данной статьи является написание алгоритма, который выполняет несколько действий одновременно. Для этого будут рассмотрены существующие решения для задач такого типа. Результатом исследования станет ряд программ, которые реализуют мультизадачность и которые могут прошиваться на одноядерные микроконтроллеры.
Ключевые слова: встраиваемые системы, Arduino, многозадачность

Multitasking in single-threaded conditions when developing for AVR

Boltovskiy Gavriil Aleksandrovich
Sholom-Aleichem Priamursky State University
Student

Abstract
The purpose of this article is to write an algorithm that performs several actions at the same time. To do this, existing solutions for problems of this type will be considered. The result of the study will be a number of programs that implement multitasking and that can be flashed onto single-core microcontrollers.
Keywords: embedded systems, Arduino, multitasking

1. Введение

1.1 Актуальность исследования

Микросхемы, используемые на платах Arduino, относятся к семейству восьмибитных микроконтроллеров AVR. Все они имеют одно вычислительное ядро и могут выполнять код только последовательно. Это существенное ограничение, которое мешает опрашивать одновременно несколько модулей, кнопок и выводить изображение на дисплей. Для решения этой проблемы существует несколько подходов.

1.2 Обзор исследований

В общем случае для разработки под микроконтроллеры AVR используются специализированный программный пакет AVR Studio. Его использование рассмотрено в статье [1] Н. Королева и Д. Королева. Оценка существующих микроконтроллеров AVR компании Atmel проведена в исследовании [2]. При реализации мультизадачности на микроконтроллерах, имеющих одно вычислительное ядро, могут использоваться специальные

операционные системы. А. Курниц рассматривает такую операционную систему в своём исследовании [3]. Реализация мультизадачаности является алгоритмической проблемой. Оценку сложности алгоритмов для микроконтроллеров произвели Ю. А. Башвеев, О. С. Литвинская в своём исследовании [4].

1.3 Цель исследования

Создать алгоритм, в котором будет реализована мультизадачность.

2. Методы исследования

В качестве микроконтроллера AVR будет использоваться AtmelMega328P, устанавливаемый на Arduino Nano. Разработка будет вестись на Arduino Wiring из-под официальной IDE [5]. Будет рассмотрено несколько вариантов в реализации мультизадачности.

3. Результаты и дискуссия

Скетч Blink из библиотеки позволяет мигать светодиодом раз в секунду. Если усложнить задачу и заставить несколько светодиодов мигать с разным периодом, то алгоритм, предлагаемый в Blink, не подходит.

Вычислительного ядра Arduino Nano достаточно для выполнения 16 млн. команд в секунду. При такой скорости последовательное выполнение нескольких задач будет казаться одновременным, такой подход называется псевдомультизадачностью.

Для AVR существует два варианта реализации мультизадачности. Создание суперцикла с прерываниями и использование операционной системы или диспетчера задач.

Второй вариант используется для крупных проектов на более мощных микроконтроллерах с большим количеством памяти. Одной из самых популярных операционных систем является freeRTOS. Для плат семейства Arduino такой подход не популярен.

Суперцикл строится по следующему принципу:

1. Внутри программы есть главный цикл, внутри которого обновляются и опрашиваются различные модули (датчики, кнопки, дисплеи и так далее);

2. Фоном ведётся счёт времени и работают прерывания (их могут вызывать тахометры, энкодеры и так далее).

По умолчанию на плате уже работают некоторые прерывания, например, системный счётчик времени для millis(). Свои прерывания есть и для UART. Именно они позволяют опрашивать порт в фоне.

Каждая задача имеет своё время выполнения, поэтому весь цикл будет выполняться с периодом, равным сумме времён выполнения всех задач в цикле.

Существует конструкция (рис. 1), которая позволяет, используя один аппаратный таймер, создать условно неограниченное количество программных таймеров.

```
uint32_t tmp; // unsigned long, 4 bite
/*
количество программных таймеров может
быть условно неограниченное количество
*/

if (millis() - tmp >= period) {
    tmp = millis();

}
```

Рисунок 1 – Программный таймер

Конструкция требует одной переменной, которая хранит в себе время последнего срабатывания таймера, в случае с Arduino и millis() это 4 байта целое беззнаковое.

Рассмотрим пример с двумя программными таймерами (рис. 2)

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:
  static uint32_t tmr1;
  if (millis() - tmr1 >= 1000){
    tmr1 = millis();
    Serial.println("tmr1")
  }

  static uint32_t tmr2;
  if (millis() - tmr2 >= 200){
    tmr2 = millis();
    Serial.println("tmr2")
  }
}
```

Рисунок 2 – Печать в Serial

Слово «tmr1» будет печататься в мониторе порта 1 раз в секунду, а слово «tmr2» 5 раз в секунду. Код, подобный данному будет корректно работать только в том случае, если нигде не будет задержек.

Следующий код позволяет мигать каждым светодиодом раз в полсекунды (рис. 3).

```
void setup() {
  // put your setup code here, to run once:
  pinMode(10, OUTPUT);
  pinMode(9, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  led1blink();
  led2blink();
}

void led1blink(){
    static uint32_t tmr;
    if (millis() - tmr >= 500){
        tmr = millis();
        digitalWrite(10, !digitalRead(10));
    }

}

void led2blink(){
    static uint32_t tmr;
    if (millis() - tmr >= 500){
        tmr = millis();
        digitalWrite(9, !digitalRead(9));
    }

}
```

Рисунок 3 – Управление светодиодами по таймеру

3. Выводы

Таким образом были рассмотрены принципы построения суперциклов, позволяющих реализовать псевдомультизадачность. Изложенные принципы актуальны для любых микроконтроллеров AVR.

Библиографический список

1. Королев Н. AVR: программирование в среде AVR Studio // Компоненты и технологии. 2004. № 3(38). С. 146-149.
2. Микроконтроллеры семейства AVR // Актуальные исследования. 2022. № 50-1(129). С. 37-39.
3. Курниц А. FreeRTOS - операционная система для микроконтроллеров Компоненты и технологии. 2011. № 2(115). С. 96-100.
4. Башвеев Ю. А. Сложность алгоритма при выборе микроконтроллеров // Современные наукоемкие технологии. 2014. № 5-2. С. 86-87.
5. Arduino IDE. URL: https://www.arduino.cc/en/software (дата обращения: 02.02.2023).