



## ИСПОЛЬЗОВАНИЕ ПЛАТФОРМЫ ARDUINO В ИЗМЕРЕНИЯХ И ФИЗИЧЕСКОМ ЭКСПЕРИМЕНТЕ

*В.И. Пономаренко<sup>1,2</sup>, А.С. Караваев<sup>1,2</sup>*

<sup>1</sup>Саратовский государственный университет им. Н.Г. Чернышевского

<sup>2</sup>Саратовский филиал Института радиотехники и электроники им. В.А. Котельникова РАН

В работе обсуждаются возможности аппаратно-программной платформы Arduino, как достаточно универсального и простого инструмента, способного занять определенную нишу в исследовательском инструментарии. Представлен пример создания на базе данной платформы радиофизической установки – гибридного хаотического генератора с запаздывающей обратной связью.

*Ключевые слова:* Нелинейная динамика, хаотический генератор, система с запаздыванием, программируемый микроконтроллер, система сбора данных, радиофизическая установка.

### Введение

Проведение экспериментов в радиофизике и электронике требует построения экспериментальных макетов различных устройств, измерения различных физических величин и зачастую автоматизированного управления ходом эксперимента. Современная база электронных компонент и измерительная техника предоставляет весьма широкие возможности для экспериментатора и позволяет создавать измерительные комплексы, обеспечивающие не только сбор данных, но и организацию обратной связи для управления экспериментальным макетом. В то же время для создания специализированного измерительного комплекса необходимы большие временные вложения, а также высокая инженерно-техническая квалификация разработчиков. С другой стороны, универсальные платформы, предлагающие готовые мощные аппаратные модули сбора данных и управления, работающие в широком диапазоне частот и амплитуд сигналов и обеспечивающие их анализ и обработку, а также управление оборудованием с помощью легких в освоении языков графического программирования, обладают при решении ряда задач недостаточной гибкостью. Они отличаются высокой ценой и часто требуют выделения значительных рабочих площадей.

В такой ситуации определенную тактическую нишу занимает программно-аппаратная платформа Arduino. С ее помощью может быть эффективно и быстро

решен целый ряд задач, связанных с постановкой радиофизического эксперимента. Платформу отличает низкая цена, наличие бесплатного программного обеспечения, возможность быстрого освоения за счет использования упрощенных языков программирования, минимальные требования к наличию дополнительных элементов и монтажного оборудования. При этом платформа предоставляет широкие возможности по генерации сигналов, сбору данных и управлению ходом эксперимента.

В данной работе обсуждаются возможности платформы Arduino и приводятся примеры ее использования в качестве системы сбора экспериментальных данных, а также для создания лабораторной установки – гибридного хаотического генератора с запаздывающей обратной связью.

В настоящее время существует множество вариантов построения лабораторных систем, позволяющих осуществлять сбор данных, обработку и анализ сигналов, а также управление внешним оборудованием. Среди таких решений можно выделить платформы, объединяющие достаточно мощные аппаратные модули сбора данных и управления с универсальными драйверами и прикладным программным обеспечением, реализующим графические языки программирования. Такие системы позволяют эффективно решать многие технические и исследовательские задачи, минимизируя затраты времени на разработку, так как предлагаемые аппаратные модули достаточно универсальны, а системы графического программирования требуют минимального времени освоения. Разработчику не требуется тратить значительные ресурсы на изучение языков программирования и совершенствование своей квалификации в разработке электронных систем с программируемыми компонентами. Пожалуй, наиболее известным примером таких систем является продукция National Instruments [1], объединяющая десятки различных модулей сбора данных, универсальные драйвера под все распространенные операционные системы и систему графического программирования LabVIEW. Другим примером таких платформ является профессиональная отечественная система ZETLAB [2], которая хорошо зарекомендовала себя, в том числе, при решении исследовательских задач [3].

Однако платой за универсальность и минимальное время освоения таких платформ является их недостаточная гибкость – при решении ряда задач не хватает возможностей систем графического программирования и приходится программировать аппаратные модули, используя обычные языки программирования. Это требует достаточно детального изучения технических особенностей устройства используемых аппаратных модулей сбора данных и управления. Кроме того, распространение таких платформ ограничивает их высокая цена.

Наиболее гибкие возможности дает создание лабораторных установок «с нуля» с использованием дискретных электронных компонент и относительно мелких микропроцессорных модулей. Например, широкое распространение для таких систем получили Parallax Basic Stamp [4], Handy Board [5], Raspberry PI [6] и др. В этом случае подразумевается самостоятельная разработка программного обеспечения управляющих микропроцессоров на языках программирования высокого уровня и языке ассемблера. Однако такая гибкость и минимальные затраты на материалы достигаются значительными временными затратами и требуют очень высокой инженерной квалификации разработчиков: электронщиков и программистов.

Между двумя подходами, перечисленными выше, существуют промежуточные решения, объединяющие небольшие, недорогие аппаратные модули и программное обеспечение в виде «упрощенных» языков программирования, обладающих существенно большей гибкостью по сравнению с графическими языками программиро-

вания, но не требующих детального изучения особенностей архитектуры конкретных семейств микропроцессоров. Одним из наиболее удачных представителей такого класса аппаратно-программных платформ, по мнению авторов, является в настоящее время платформа Arduino.

Значительный опыт авторов по созданию экспериментальных радиофизических установок с использованием всех перечисленных выше подходов позволяет утверждать, что платформа Arduino, занимает определенную тактическую нишу, существенно снижая время и стоимость разработки при решении целого ряда научно-технических задач.

Arduino – это простая в использовании открытая электронная платформа, включающая так называемые стартовые наборы разработчика (starter kit) и открытое программное обеспечение и предназначенная для быстрого создания интерактивных электронных устройств [7]. Она была создана группой энтузиастов, которые позиционировали свою разработку как платформу для быстрой реализации небольших проектов. Arduino строится на базе микроконтроллеров Atmel [8, 9] и используется для получения сигналов от аналоговых и цифровых датчиков, управления различными исполнительными устройствами и обмена информацией с компьютером при помощи различных интерфейсов.

Все эти устройства представляют собой простые в использовании наборы, предлагающие схожую функциональность. Arduino, в свою очередь, тоже упрощает процесс работы с микроконтроллерами и позволяет при создании простых проектов обойтись без пайки элементов, используя сборку электромеханическими соединителями на макетных платах. Arduino имеет ряд преимуществ перед другими устройствами. Во-первых, это низкая стоимость (розничная цена некоторых модулей Arduino ниже 10 долларов). Во-вторых, программное обеспечение Arduino работает под управлением всех наиболее распространенных операционных систем: Windows, Macintosh OS X и Linux, в то время как большинство других устройств ограничивается одной системой (либо Windows, либо Linux). В-третьих, простая и понятная среда программирования – среда Arduino – подходит как для начинающих пользователей, так и для опытных разработчиков, экономящих свое время. Само программное обеспечение Arduino может быть модифицировано опытными пользователями, поскольку весь проект изначально предполагает открытую архитектуру с возможностью расширения. Пользователи, желающие максимально полно и эффективно использовать возможности микроконтроллера, могут без ограничений использовать любые сторонние компиляторы и внутрисхемные программаторы.

Эта реализация оказалась настолько удачной, что получила широкое распространение для применения в быту, при создании игрушек, а также среди любителей робототехники. В последнее время профессионалы также все чаще используют Arduino для быстрого решения текущих задач. Существует множество интернет-ресурсов, ориентированных на поддержку разработок на базе Arduino, в том числе на русском языке [10].

## **1. Компоненты платформы Arduino**

**1.1. Аппаратная часть платформы Arduino.** Существует более 20 вариантов базовых модулей Arduino [11] и не менее десятка плат расширения для них. Существует также огромное количество разработок как азиатских, так и отечественных компаний, обеспечивающих совместимость с Arduino. Базовые модули отлича-

ются габаритами, моделями и количеством предустановленных микроконтроллеров, а также набором предустановленных дополнительных элементов. К последним относятся: стабилизаторы на разные напряжения питания, светодиоды, тактовые кнопки, разъемы цифровых портов и коммуникационные разъемы (USB, COM-порты и другие), компоненты, обеспечивающие зарядку Li-Pol аккумуляторов и т.п.

Самый маленький базовый модуль – Arduino Mini (рис. 1, а) имеет стабилизатор питания на 5 В, микроконтроллер Atmel ATmega168, 16 МГц кварцевый резонатор и 24-контактный разъем, контакты которого могут быть программно сконфигурированы для использования в качестве цифровых линий ввода-вывода, выходов ШИМ, последовательных интерфейсов UART, SPI, I2C. Несколько линий могут использоваться в качестве входов 10-битного АЦП.

Базовый модуль Arduino Uno (рис. 1, б) имеет 2 предустановленных микроконтроллера: прогрессивный ATmega328, поддерживающий частоты тактирования до 20 МГц и имеющий 32 Кбайт флеш-памяти команд и 2 Кбайт SRAM-памяти данных, а также ATmega8U2, аппаратно поддерживающий обмен данными через интерфейс USB.

Для разработчиков, которым важно наличие большого объема памяти, нескольких цифровых интерфейсов и большого количества цифровых линий ввода-вывода, удачным решением может оказаться Arduino Mega ADK с установленным микроконтроллером ATmega2560, поддерживающая режим USB-host. Arduino Mega ADK поддерживает до 54 цифровых линий ввода-вывода, до 14 каналов ШИМ, до 16 входов 10-битного АЦП. Имеется 4 последовательных порта UART, 5 интерфейсов SPI, 6 независимых таймеров. Микросхема имеет 256 Кбайт встроенной флеш-памяти, 8 Кбайт SRAM-памяти данных, 4 Кбайт EEPROM и работает на скорости до 16 MIPS.

Самым мощным решением в настоящее время является плата Arduino Due (рис. 2), имеющая предустановленный 32-битный микроконтроллер с ARM-ядром (Atmel SAM3X8E ARM Cortex-M3 в 144-выводном корпусе). Процессор работает на тактовой частоте 84 МГц (до 84 MIPS) и имеет 96 Кбайт ОЗУ, 512 Кбайт флеш-памяти, контроллер прямого доступа к памяти. Плата поддерживает до 54 цифровых линий ввода-вывода, до 12 каналов ШИМ, 16 каналов 12-битного АЦП, 2 канала 12-битного ЦАП, 4 UART, 3 интерфейса SPI, 2 интерфейса I2C, поддерживается USB

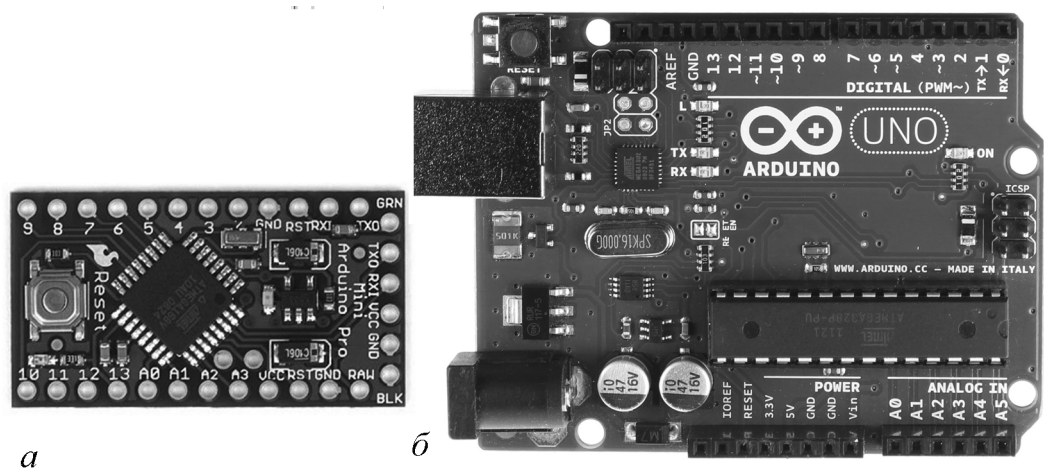


Рис. 1. Внешний вид Arduino Mini (а) и Arduino UNO (б)

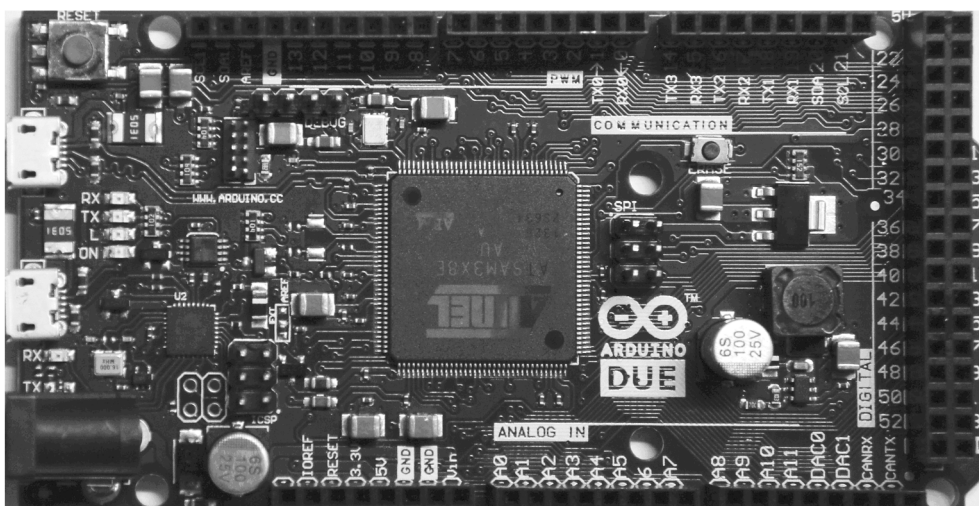


Рис. 2. Внешний вид Arduino DUE

OTG.

Разработчики Arduino используют в своих платах микропроцессоры фирмы Atmel, являющейся признанным мировым лидером в сегменте 8-битных микроконтроллеров.

Платы расширения подключаются через электромеханические разъемы к базовым платам, обеспечивая возможности подключения к компьютерной сети по проводным каналам (плата расширения Ethernet Shield), беспроводным каналам (плата расширения WiFi), организацию передачи данных между устройствами по цифровому радиоканалу Zigbee (Maxstream Xbee Zigbee). Существуют платы расширения со специализированными токовыми драйверами, обеспечивающие подключение электромоторов постоянного тока, шаговых двигателей и сервомоторов (Motor Shield) и другие платы [4].

**1.2. Программное обеспечение.** Программный комплекс состоит из среды разработки Arduino и программного модуля, расположенного в разделе загрузчика (bootloader) флеш-памяти микроконтроллера. Наличие загрузчика позволяет не использовать внешний ISP-программатор, а загружать микрокод непосредственно через USB, используя специализированную утилиту. Среда разработки Arduino написана на языке JAVA и является кроссплатформенной. Она включает редактор программного кода, отладчик, утилиту-программатор, систему помощи, библиотеки подпрограмм и др. элементы. Среда имеет дружелюбный интуитивно понятный интерфейс в стиле Windows (рис. 3).

Доступны версии для Linux, Mac OS X и Windows. Для загрузки программ и связи с микроконтроллером среда разработки подключается к аппаратной части Arduino по USB-интерфейсу в режиме виртуального COM-порта. Среда разработки основана на языке программирования Processing [12], специально спроектированном для быстрого освоения людьми, не являющимися профессиональными программистами и не имеющими специального опыта работы с микроконтроллерами. Именно поэтому программирование на Arduino очень просто осваивается даже школьниками. Язык программирования аналогичен используемому в проекте Wiring [13]. Это

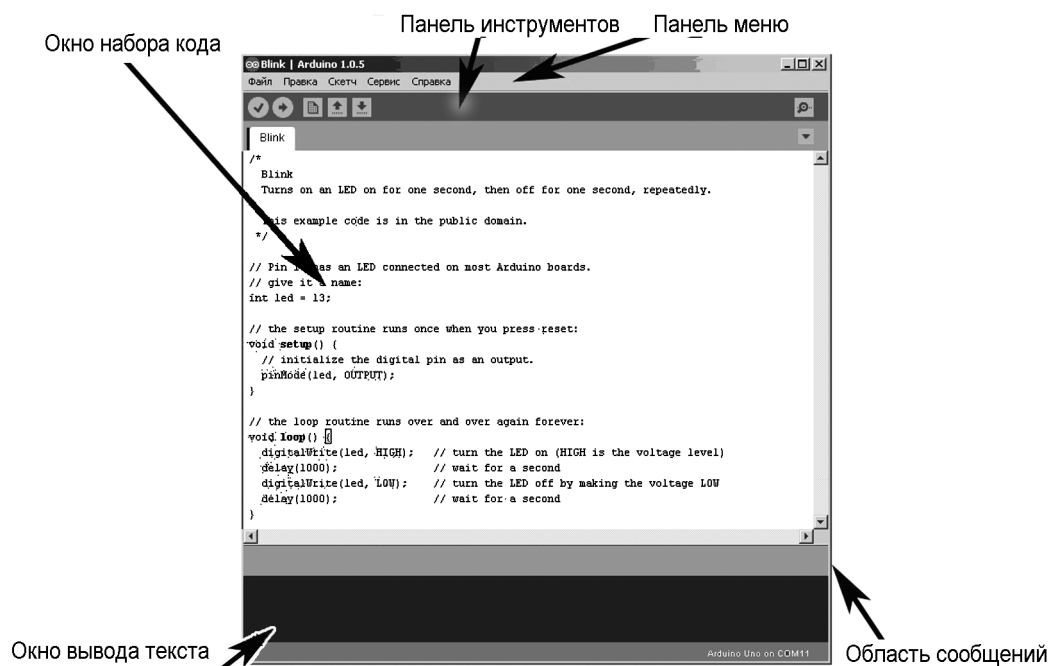


Рис. 3. Окно программной среды Arduino с программой, обеспечивающей мигание светодиодам

язык стилистически очень близкий к C++ и дополненный большим набором библиотек, позволяющих легко настраивать и использовать аппаратные модули микроконтроллеров, не тратя время на детальное изучение особенностей внутренней программно-аппаратной реализации этих модулей. Все программное обеспечение является свободно распространяемым, бесплатным и имеет открытый исходный код. Проект является плодом коллективных усилий большого числа людей во всем мире. Все исходные коды, библиотеки и примеры программ доступны для скачивания, использования и модификации (см., например, [7,8]).

Программа, написанная в среде Arduino, называется «скетч». После написания скетча его нужно загрузить в контроллер (кнопка «загрузить» в среде), после чего программа немедленно начинает выполняться.

**1.3. Модульная структура платформы Arduino.** Благодаря тому, что весь проект Arduino является открытым, и его программные продукты и принципиальные схемы доступны в сети Интернет, многие сторонние производители выпускают совместимые модули, существенно расширяющие функционал базовой платы. Открытое программное обеспечение, в свою очередь, позволяет сторонним разработчикам создавать дополнительные программные модули, поддерживающие новые или существующие аппаратные элементы.

Для управления двигателями постоянного тока, шаговыми двигателями и сервомоторами разработаны силовые модули управления двигателями, для обеспечения беспроводной связи выпускают радиомодули, поддерживающие ZigBee, Bluetooth, WiFi и другие стандарты радиопередачи. Есть специализированные модули, обеспечивающие связь по Ethernet. Существуют также модули, предназначенные для организации работы с ЖК и LCD индикаторами, подключаемые к Arduino видеоредакторы

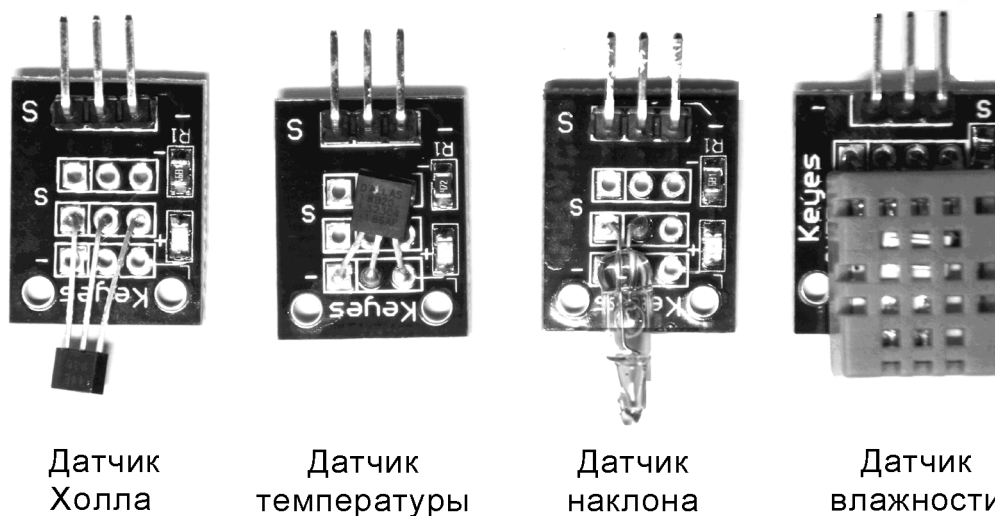


Рис. 4. Некоторые датчики платформы Arduino

и многое другое.

Платформа Arduino активно развивается, и в настоящее время доступно большое число аналоговых и цифровых датчиков, предназначенных для оценки самых разнообразных аналоговых величин [14]. К ним относятся датчики магнитного поля, температуры, влажности, освещенности, ультразвуковые датчики для измерения расстояния и многие другие. На рис. 4 приведены некоторые из датчиков, подключаемых к Arduino.

Существуют как аналоговые, так и цифровые датчики для платформы. Аналоговые датчики подключаются к входам АЦП, цифровые могут быть подключены к цифровым линиям ввода-вывода. Разрешение АЦП для модулей с 8-битными контроллерами Atmel составляет 10 бит, для ARM-микроконтроллера на Arduino Due – 12 бит.

## 2. Пример программирования: получение данных натурного эксперимента при помощи Arduino

Важным этапом практически любого современного натурного эксперимента является сбор данных с установки и их сохранение в цифровом виде на компьютере для последующей обработки и анализа.

Даже такая простая плата, как Arduino Nano, позволяет оцифровывать и передавать в компьютер экспериментальные сигналы с 10-битным разрешением, которого зачастую достаточно для проведения измерений. В качестве примера приведем текст скетча (Листинг 1), позволяющего оцифровывать аналоговый сигнал с датчика, подключенного к входу A0 платы, с частотой 500 Гц и отправлять результат в персональный компьютер (ПК) по интерфейсу UART (COM-порт).

```
int sensorValue=0; // Величина измеренного напряжения
void setup() { // Раздел однократной инициализации
  Serial.begin(9600); // Инициализация UART, скорость 9600 бод
```

```

}
void loop() { // Раздел циклических действий
  sensorValue = analogRead(A0); // Читаем вход A0
  Serial.println(sensorValue); // результат отправляем в UART
  delay(2); // Задержка 2 мс перед следующим чтением
}

```

Листинг 1. Текст скетча для оцифровки аналогового сигнала с датчика

Скетч может работать на всех платах Arduino. Платы, поддерживающие интерфейс USB (например, Arduino Nano), позволяют передавать данные в ПК по этому физическому интерфейсу в режиме виртуального СОМ-порта (VCP). Таким образом, с точки зрения программиста, на ПК данные приходят с СОМ-порта и могут быть сохранены стандартным программным обеспечением, например, программой Nupur Terminal.

Аналогичная программа для платы Arduino, написанная на языке Си, получается значительно более громоздкой и менее наглядной. Приведенный ниже пример написан для популярного в среде программистов микроконтроллеров Atmel компилятора WinAVR. Предполагается, что используется внешний кварцевый резонатор с частотой 16 МГц как, например, в Arduino Nano (Листинг 2).

```

#include <avr\io.h> // Необходимые заголовочные
#include <avr\interrupt.h> // файлы подключаются
#include <avr\signal.h> // здесь
int sensorValue=0; // Величина измеренного напряжения
unsigned int analogRead(unsigned char Channel){
  //Функция опрашивает канал Channel
  ADMUX=(1<<REFS0)|(1<<ADLAR)|Channel; //Источник опоры AVCC,
  //Single Ended канал Channel
  ADCSRA|=(1<<ADSC); // Запуск преобразования
  while (ADCSRA&(1<<ADSC)); // Ожидаем окончания преобразования
  return ADC; //Возвращаем 10 битный результат
}

void adcInit(void){ //Однократная инициализация АЦП
  ADMUX=(1<<REFS0); //Опора AVCC
  //Включить АЦП, делитель такта АЦП /128,
  ADCSRA=(1<<ADEN)|(1<<ADPS2)|(1<<ADPS1)|(1<<ADPS0);
  adcGet8(0); //Инициализирующий первичный опрос АЦП "вхолостую"
}

void uartSend(unsigned char X){ //Асинхронная передача байта
  while( !( UCSRA & (1<<UDRE)) ); //Ждем окончания предыдущей передачи
  UDR=X; //Послать байт в UART
}

void uartInit(void){ //Однократная инициализация UART
  //Инициализация UART: 9600 бод при 16 МГц кварцевом резонаторе
  //Формат: 8 бит данных, 1 стоп-бит, контроля четности нет
  UBRRL=103; //Настройка счетчика UART 9600 бод
  UCSRB=(1<<TXEN)|(1<<RXEN); // Включить приемник и передатчик
}

```



```

}
unsigned char timerDelay2ms(void){ //Задержка 2 мс
if(TCNT1>=125) {
TCNT1=0;
return 1;
}
return 0;
}
void timerInit(void){ //Однократная инициализация таймера 1
TCCR1B=(1<<CS12); //Старт таймера с делителем частоты 256
TCNT1=0; //Сброс счетчика таймера
}
void main(void) {
adcInit();
uartInit();
timerInit();
while(1) { // "Вечный"цикл
if (timerDelay2ms()){
sensorValue=analogRead(0); // Читаем вход 0 АЦП
//Результат отправляем в UART двумя 8-битными посылками
uartSend(sensorValue>>8); // Сначала старшие 2 бита
uartSend(sensorValue); // затем младшие 8 бит
}
}
}

```

Листинг 2. Пример оцифровки аналогового сигнала на языке Си

Программа на языке Си имеет заметно больший объем [15]; кроме того, для программирования микроконтроллера требуется наличие специального программатора и специализированных знаний об аппаратных особенностях реализации конкретного микроконтроллера, а также навыков и опыта его программирования. Программа для Arduino на языке Processing является более компактной и универсальной, оставаясь работоспособной не только для всех плат с 8-битными микроконтроллерами, но даже для Arduino Due с микроконтроллером ARM. Кроме того, при создании программы на языке Processing не требуется тратить значительное время на изучение особенностей программирования конкретных аппаратных модулей конкретных микроконтроллеров, что неизбежно при использовании компиляторов Си. Таким образом, использование платформы Arduino позволяет быстро и с минимальными затратами проводить разнообразные измерения в физическом эксперименте.

### 3. Генератор с запаздыванием на Arduino DUE

Arduino DUE – самая мощная на сегодняшний день плата платформы, со встроенным 12-битными АЦП и ЦАП. Наличие богатой периферии позволяет создать множество полезных приложений для использования в эксперименте.

В качестве примера использования Arduino DUE ниже описана созданная нами с минимальным количеством дополнительных электронных компонент радиофизическая экспериментальная установка – хаотический генератор с запаздывающей обратной связью, описываемый уравнением

$$\varepsilon \dot{x}(t) = -x(t) + \lambda - x^2(t - \tau), \quad (1)$$

где  $x(t)$  – состояние системы в момент времени  $t$ ;  $\lambda$  – управляющий параметр;  $\tau$  – время запаздывания;  $\varepsilon$  – параметр, характеризующий инерционность системы.

Генератор (1) является популярным объектом исследования в нелинейной динамике и представляет собой кольцо, состоящее из линии задержки, квадратичного нелинейного элемента и фильтра низких частот первого порядка. Блок-схема такого генератора, созданного на базе Arduino DUE, приведена на рис. 5.

Линия задержки создана в оперативной памяти микроконтроллера в виде кольцевого буфера. Нелинейная функция реализована при помощи цифровых вычислений внутри микроконтроллера, а интегрирование сигнала осуществляется аналоговым RC-фильтром нижних частот первого порядка. Величина резистора используется в качестве управляющего параметра ( $R$  – переменный резистор 200 кОм,  $C=2200$  пФ). На входе и выходе фильтра включены повторители DA1.1, DA1.2, собранные на операционном усилителе AD822.

Благодаря наличию аналогового интегратора, данная лабораторная модель представляет собой полноценный радиофизический хаотический генератор с запаздыванием. Вместе с тем цифровая реализация линии задержки и нелинейного преобразования позволяет в ходе исследований в широких пределах изменять время запаздывания  $\tau$  и управляющий параметр  $\lambda$ , задавая эти параметры с высокой точностью. Устройство с аналогичной компоновкой использовалось, в частности, в экспериментальных исследованиях в работе [16].

Особенности реализации генератора с запаздыванием на базе микроконтроллеров Atmel с использованием целочисленной арифметики подробно обсуждаются в [17,18]. Программа на языке Processing, реализующая описанный в указанных работах алгоритм, компактна, и ее листинг приводим полностью (Листинг 3).

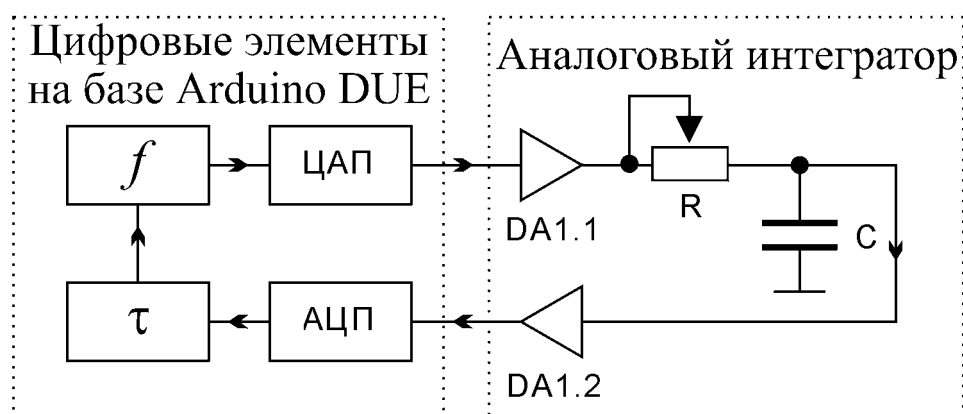


Рис. 5. Блок-схема генератора с запаздыванием на Arduino DUE

```

#define TAU 100 // Временная задержка
#define IC 5 // Начальные условия
int DelayLine[TAU]; // Линия задержки
int RP, WP; // Указатели, хранящие индексы элементов массива,
// соотв. откуда/куда обратиться следующая команда
// чтения/записи DelayLine
int i;
int Lambda; // Управляющий параметр
long XSQR; // Квадрат динамической переменной
int X; // динамическая переменная X(t)
int XDL; // X(t-TAU)
void setup() {
  RP=0;
  WP= TAU -1;
  //Инициализация линии запаздывания и динамической переменной
  X=IC;
  for(i=0;i<=TAU-1;i++) DelayLine[i]=IC;
  Lambda=2000; // Значение управляющего параметра 1.95,
  // соответствующее хаотическим автоколебаниям
  analogReadResolution(12); // Задать разрешение АЦП, бит
  analogWriteResolution(12); // Задать разрешение ЦАП, бит
}
void loop() {
  X=analogRead(A0); // считываем динамическую переменную с
  // выхода фильтра через АЦП
  X=X-2048; // Учет сдвига динамического диапазона АЦП
  DelayLine[WP]=X; // Помещаем значение с выхода фильтра в
  // линию задержки,
  if (WP>=TAU-1) // реализованную в виде кольцевого буфера
  WP=0;
  else
  WP++;
  XDL=DelayLine[RP]; // извлекаем из линии задержки X[t-TAU]
  if (RP>=DELAY-1)
  RP=0;
  else
  RP++;
  XSQR=XDL*XDL/1024; // Возводим X(t-TAU) в квадрат и масштабируем
  X=Lambda-XSQR; // Вычисляем нелинейное преобразование
  X=X+2048; // Учет сдвига динамического диапазона ЦАП
  analogWrite(DAC0,X); // Отправляем результат вычислений на вход
  // фильтра через ЦАП
}

```

Листинг 3. Реализация хаотического генератора с запаздыванием на языке Processing

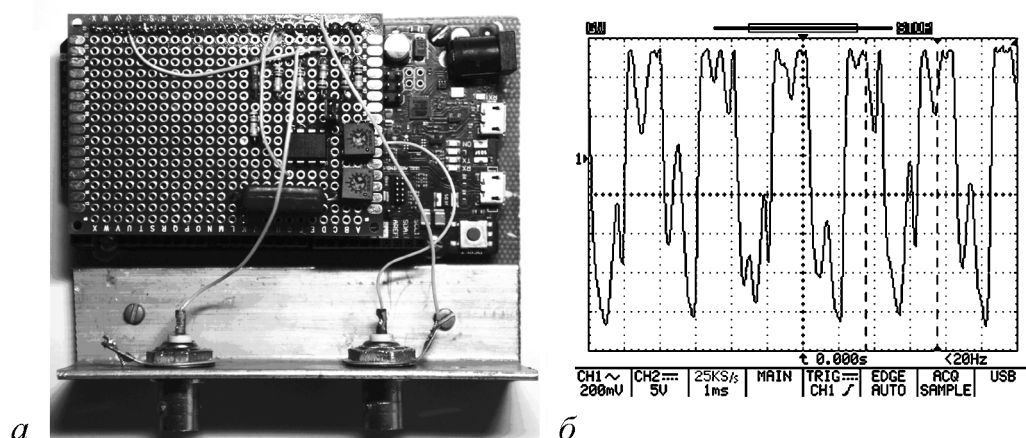


Рис. 6. *а* – фотография лабораторной установки – генератора с запаздывающей обратной связью, описываемого уравнением (1). Аналоговый интегратор (см. рис. 5) собран в виде мезонинной платы над платой Arduino DUE, реализующей цифровые преобразования. *б* – временная реализация генератора в хаотическом режиме

На рис. 6 приведена фотография лабораторной установки – генератора с запаздывающей обратной связью (*а*) и осциллограмма напряжения на выходе ЦАП Arduino DUE (*б*), демонстрирующая хаотические колебания генератора с запаздывающей обратной связью с квадратичной нелинейностью.

### Заключение

Использование платформы Arduino в физическом эксперименте дает возможность просто и быстро решить множество технических задач, связанных с измерениями, передачей данных в компьютер и управлением исполнительными устройствами при весьма умеренной стоимости.

Продemonстрировано использование платформы Arduino для простейших измерений аналогового напряжения, а также для создания экспериментальной радиофизической установки – гибридного хаотического генератора с запаздывающей обратной связью с квадратичной нелинейностью, инерционный элемент которого представляет собой аналоговый RC-фильтр.

Немаловажно, что при построении макетов от монтажа пайкой удастся полностью отказаться или свести его к минимуму. Как показал опыт практического использования, возможности платформы Arduino при создании экспериментальных макетов позволяют существенно сэкономить временные и материальные затраты.

*Работа выполнена при государственной поддержке ведущих научных школ, грант НШ-1726.2014.2 и при поддержке РФФИ, грант № 13-02-00227.*

### Библиографический список

1. [Электронный ресурс]. Режим доступа: <http://www.ni.com>, свободный.
2. [Электронный ресурс]. Режим доступа: <http://www.zetms.ru>, свободный.
3. Корчагин С.А. Модернизация физических лабораторий посредством внедрения

интегрированных информационно-измерительных систем // Информационные технологии в образовании. Саратов: Изд-во ООО «Наука», 2013. С. 139.

4. Parallax Inc [Электронный ресурс]. Режим доступа: <http://www.parallax.com/catalog/microcontrollers/basic-stamp>, свободный.
5. [Электронный ресурс]. Режим доступа: <http://www.handyboard.com/>, свободный.
6. [Электронный ресурс]. Режим доступа: <http://www.raspberrypi.org/>, свободный.
7. Arduino [Электронный ресурс]. Режим доступа: <http://arduino.cc/>, свободный.
8. Евстифеев А.В. Микроконтроллеры AVR семейства Classic фирмы Atmel. 3-е изд., стер. М.: Издательский дом «Додэка-XXI», 2006. 288 с., ил.
9. Atmel corporation [Электронный ресурс]. Режим доступа: <http://www.atmel.com/products/microcontrollers/avr/default.aspx>, свободный.
10. Arduino [Электронный ресурс]. Режим доступа: <http://arduino.ru/>, свободный.
11. Википедия [Электронный ресурс]. Режим доступа: <http://ru.wikipedia.org/wiki/Arduino>, свободный.
12. Википедия [Электронный ресурс]. Режим доступа: <http://ru.wikipedia.org/wiki/Processing>, свободный.
13. Википедия [Электронный ресурс]. Режим доступа: <http://wiring.org.co>, свободный.
14. Fisher D.K., Gould P.J. Open-source hardware is a low-cost alternative for scientific instrumentation and research // Modern Instrumentation. 2012. Vol. 1. P. 8.
15. Белов А.В. Микроконтроллеры AVR в радиолюбительской практике. СПб.: Наука и техника, 2007. 352 с., ил.
16. Bezruchko B.P., Karavaev A.S., Ponomarenko V.I., Prokhorov M.D. Reconstruction of time-delay systems from chaotic time series // Physical Review E. 2001. Vol. 64. P. 056216.
17. Ponomarenko V.I., Prokhorov M.D., Karavaev A.S., Kulminskiy D.D. An experimental digital communication scheme based on chaotic time-delay system // Nonlinear Dynamics. 2013. Vol. 74. P. 1013.
18. Караваев А.С., Кульминский Д.Д., Пономаренко В.И., Прохоров М.Д. Система цифровой передачи информации, маскируемой хаотическим сигналом системы с запаздыванием // Информационно-управляющие системы. 2013. № 4. С. 30.

*Поступила в редакцию*      5.05.2014  
*После доработки*            20.06.2014

## **USING ARDUINO PLATFORM IN THE MEASUREMENTS AND THE PHYSICAL EXPERIMENT**

*V. I. Ponomarenko<sup>1,2</sup>, A. S. Karavaev<sup>1,2</sup>*

<sup>1</sup> Saratov State University

<sup>2</sup> Kotelnikov Institute of Radio-engineering and Electronics of RAS, Saratov Branch

This paper discusses the possibility of a hardware-software platform Arduino, as

a relatively simple and flexible tool that could occupy a niche in the research tools. Radiophysical chaotic oscillator with delayed feedback was created on the base of Arduino.

*Keywords:* Nonlinear dynamics, chaotic oscillator, delayed feedback system, programmable microcontroller, data acquisition system, radiophysical installation.



*Пономаренко Владимир Иванович* – родился в Саратове (1960), окончил Саратовский госуниверситет (1982). В настоящее время ведущий научный сотрудник Саратовского филиала Института радиотехники и электроники им. В.А. Котельникова РАН, профессор кафедры динамического моделирования и биомедицинской инженерии Саратовского госуниверситета. Доктор физико-математических наук (2009, СГУ) по специальности «Радиофизика». Автор более 100 статей. По данным РИНЦ число цитирований более 700, h-индекс – 12.

410012 Саратов, ул. Астраханская, 83  
Саратовский государственный университет им. Н.Г. Чернышевского  
410019 Саратов, ул. Зеленая, 38  
Саратовский филиал Института радиотехники и электроники РАН  
E-mail: ponomarenkovi@gmail.com



*Караваев Анатолий Сергеевич* – родился в Саратове (1981), окончил Саратовский госуниверситет (2004). В настоящее время доцент СГУ, старший научный сотрудник СФ ИРЭ им. В.А. Котельникова РАН. Защитил диссертацию на соискание ученой степени кандидата физико-математических наук (2007, СГУ) в области радиофизики и нелинейной динамики. Опубликовал 36 научных статей. По данным РИНЦ число цитирований около 200, h-индекс – 5.

410012 Саратов, ул. Астраханская, 83  
Саратовский государственный университет им. Н.Г. Чернышевского  
410019 Саратов, ул. Зеленая, 38  
Саратовский филиал Института радиотехники и электроники РАН  
E-mail: karavaevas@gmail.com