

## Машина Морзе на платформе Arduino

*Кизянов Антон Олегович*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В данной статье описан процесс создания системы распознавания кодов Морзе. Для создания потребуется кнопка, зуммер, светодиод и плата Arduino. Созданный проект позволяет набирать коды Морзе через кнопку, а Arduino преобразовывает их в буквенные значения на экран компьютера.

**Ключевые слова:** Arduino, зуммер

## Morse machine on the Arduino platform

*Kizyanov Anton Olegovich*

*Sholom-Aleichem Priamursky State University*

*student*

### Abstract

This article describes the process of creating a Morse code recognition system. To create you need a button, buzzer, LED and Arduino board. The created project allows you to type Morse codes through a button, and the Arduino converts them into alphabetic values on the computer screen.

**Keywords:** Arduino, buzzer

Азбука Морзе — код Морзе, «Морзянка» — способ кодирования букв алфавита, цифр, знаков препинания и других символов при помощи длинных и коротких сигналов, так называемых «тире» и «точек» (а также пауз, разделяющих буквы).

За единицу времени принимается длительность одной точки. Длительность тире равна трём точкам. Пауза между знаками в букве — одна точка, между буквами в слове — 3 точки, между словами — 7 точек.

Цель исследования – создать преобразователь кодов азбуки Морзе в буквенные значения на базе платы Arduino.

Ранее этим вопросом интересовались Л.М. Шаги, Б.К. Абдураимова развивали тему «Разработка алгоритма для аппаратной реализации криптографической защиты» [1] в которой описывается проблемы малой и нестандартной разрядности АЦП микроконтроллеров, для использования алгоритмов шифрования. Проанализированы характерные особенности построения алгоритма шифрования для нестандартной разрядности. Проведена работа над гипотезами по отношению к алгоритмам шифрования. На основе проведенного исследования автором предлагается

схема минимизированного аналога стандарта AES. С.С. Белый, А.П. Борисов с темой «Программа для передачи данных радиомодемами с криптографической защитой» [2], а подробнее про моделирование передачи данных радиомодемами на платформе Arduino с шифрованием данных при использовании протоколов RSA, AES, A5.1. Программа обеспечивает: прием данных и выдачу пользователю сообщения о приеме с возможностью ввода ключа для расшифровки, в случае, если сообщение было зашифровано отправителем; хранение исходящих и входящих сообщений с возможностью очистки хранилищ. Н.А. Буракова, В.А. Кашицын опубликовали статью «Устройство генерации секретного ключа для обмена сообщениями на основе древовидных машин четности» [3] описали подход к проектированию устройств генерации и обмена секретными ключами, поскольку используется динамическая стохастическая синхронизация двух древовидных машин четности, запрограммированных на микроконтроллерах. Такой метод не предполагает сложных вычислений из области теории чисел, а вместо этого работает с базовыми арифметическими операциями. В результате разработан безопасный протокол взаимодействия в сети.

Для этого потребуется:

- Плата Arduino;
- Соединительные провода;
- Зуммер;
- Кнопка;
- Светодиод;
- 1 резистора на 220 Ом.

Схема подключения представлена на рисунке 1.

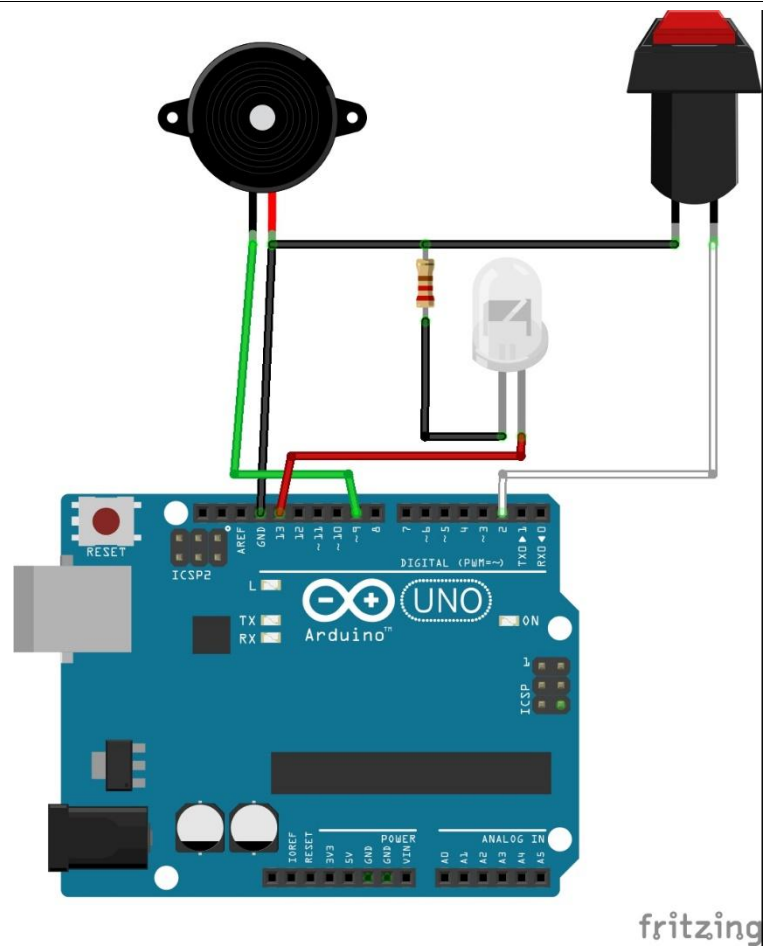


Рис. 1 Схема подключения к плате Arduino

```

const int buttonPin = 2;
const int ledPin = 13;
const int buzzer = 9;

int ledState = HIGH;
int buttonState = LOW;
int lastButtonState = LOW;
int doesitwork = LOW;

int pausetime_value = 250;
long signal_length = 0;
long pausetime = 0;

String morse = "";
String dash = "-";
String dot = "*";

boolean cheker = false;
boolean linecheker = false;

long lastDebounceTime = 0;
long debounceDelay = 50;
void setup()
{
  Bridge.begin(9600);
  pinMode(buttonPin, INPUT);

```

```
pMode(ledPin, OUTPUT);
pMode(buzzer, OUTPUT);
while (!digitalRead(buttonPin))
}
void loop() {
  buttonState = digitalRead(buttonPin);
  if (buttonState && lastButtonState)
  {
    ++signal_length;
    if (signal_length < 2 * pausetime_value)
    {
      tone(buzzer, 1500) ;
    }
    else
    {
      tone(buzzer, 1000) ;
    }
  }
  else if (!buttonState && lastButtonState)
  {
    if (signal_length > 50 && signal_length < 2 * pausetime_value )
    {
      morse = morse + dot;
    }
    else if (signal_length > 2 * pausetime_value)
    {
      morse = morse + dash;
    }
    signal_length = 0;
    digitalWrite(13, LOW);
    noTone(buzzer);
  }
  else if (buttonState && !lastButtonState)
  {
    pausetime = 0;
    digitalWrite(13, HIGH);
    cheker = true;
    linecheker = true;
  }
  else if (!buttonState && !lastButtonState)
  {
    ++pausetime;
    if (( pausetime > 3 * pausetime_value ) && (cheker))
    {
      showaj(morse);
      cheker = false;
      morse = "";
    }
    if ((pausetime > 15 * pausetime_value) && (linecheker))
    {
      Bridge.showln();
      linecheker = false;
    }
  }
  lastButtonState = buttonState;
  delay(1);
}
```

```
void showaj(String translate)
{ if (translate == "*-")
    Bridge.show("A");
  else if (translate == "-***")
    Bridge.show("B");
  else if (translate == "-*-")
    Bridge.show("C");
  else if (translate == "-**")
    Bridge.show("D");
  else if (translate == "**")
    Bridge.show("E");
  else if (translate == "***-")
    Bridge.show("F");
  else if (translate == "--*")
    Bridge.show("G");
  else if (translate == "****")
    Bridge.show("H");
  else if (translate == "**")
    Bridge.show("I");
  else if (translate == "*---")
    Bridge.show("J");
  else if (translate == "-*-")
    Bridge.show("K");
  else if (translate == "*-**")
    Bridge.show("L");
  else if (translate == "--")
    Bridge.show("M");
  else if (translate == "-*")
    Bridge.show("N");
  else if (translate == "---")
    Bridge.show("O");
  else if (translate == "*--*")
    Bridge.show("P");
  else if (translate == "--*-")
    Bridge.show("Q");
  else if (translate == "*-")
    Bridge.show("R");
  else if (translate == "****")
    Bridge.show("S");
  else if (translate == "-")
    Bridge.show("T");
  else if (translate == "***-")
    Bridge.show("U");
  else if (translate == "****-")
    Bridge.show("V");
  else if (translate == "*--")
    Bridge.show("W");
  else if (translate == "-**")
    Bridge.show("X");
  else if (translate == "-*---")
    Bridge.show("Y");
  else if (translate == "--**")
    Bridge.show("Z");
  else if (translate == "*----")
    Bridge.show("1");
  else if (translate == "***---")
    Bridge.show("2");
  else if (translate == "****--")
```

```
    Bridge.show("3");  
else if (translate == "****-")  
    Bridge.show("4");  
else if (translate == "*****")  
    Bridge.show("5");  
else if (translate == "-****")  
    Bridge.show("6");  
else if (translate == "--***")  
    Bridge.show("7");  
else if (translate == "---**")  
    Bridge.show("8");  
else if (translate == "----*")  
    Bridge.show("9");  
else if (translate == "-----")  
    Bridge.show("0");  
Bridge.show(" ");  
translate = "";  
}
```

Результат работы можно увидеть на рисунке 2 и 3.

На рисунке 3 отображается декодированное сообщение переданное через Arduino.

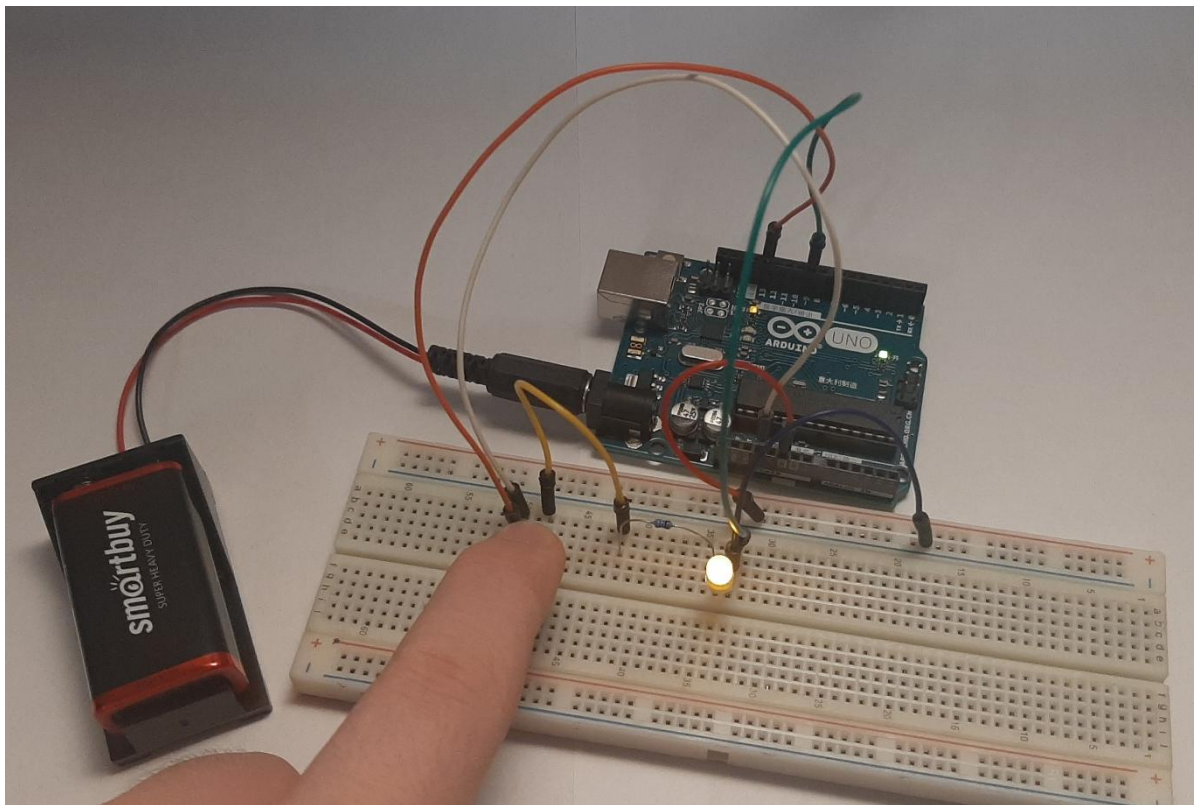


Рис. 2 Схема в собранном состоянии

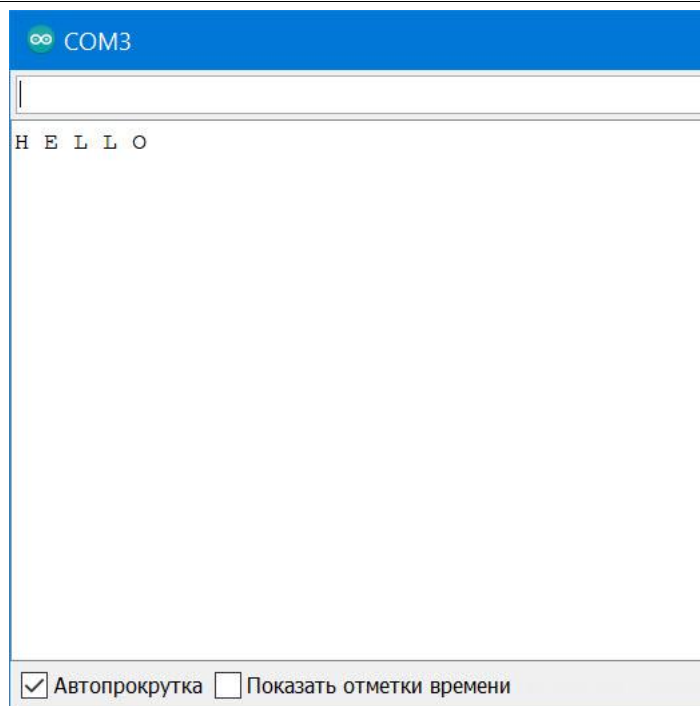


Рис. 3 Вывод сообщения от Arduino

### Вывод

Результатом статьи была создана работающая схема преобразования кодов азбуки Морзе в буквенные значения через плату Arduino. Схема и реализация получились довольно простыми, что ещё раз доказывает, что Arduino не является чем-то сложным и заумным, для этого минимальных навыков писать код и читать документацию.

### Библиографический список

1. Шаги Л.М., Абдураимова Б.К. Разработка алгоритма для аппаратной реализации криптографической защиты // Вопросы науки и образования. 2018. № 23 (35). С. 13-17. URL: <https://www.elibrary.ru/item.asp?id=36292932> (Дата обращения: 30.08.2020)
2. Белый С.С., Борисов А.П. Программа для передачи данных радиомодемами с криптографической защитой // Свидетельство о регистрации программы для ЭВМ RU 2017615253, 10.05.2017. Заявка № 2017612081 от 15.03.2017.
3. Буракова Н.А., Кашицын В.А. Устройство генерации секретного ключа для обмена сообщениями на основе древовидных машин четности // Colloquium-journal. 2019. № 13-2 (37). С. 302-308. URL: <https://www.elibrary.ru/item.asp?id=38533106> (Дата обращения: 30.08.2020)