

Эмулирование HID устройств с помощью микроконтроллера Arduino Pro Micro

Болтовский Гавриил Александрович

Приамурский государственный университет им. Шолом-Алейхема

Студент

Аннотация

Целью исследования является разработка устройства, эмулирующего поведение USB мыши компьютера с использованием Arduino Leonardo. Было разработано устройство, эмулирующее USB мышью с использованием Arduino.

Ключевые слова: Arduino Leonardo, программа, устройство, HID, USB мышью, разработка.

Emulating HID devices using the Arduino Pro Micro microcontroller

Boltovsky Gavriil Alexandrovich

Sholom-Aleichem Priamursky State University

Student

Abstract

The purpose of this article is to develop a device that emulates the behavior of the USB mouse on a computer using the Arduino Leonardo. This article was developed device emulating a USB mouse using the Arduino.

Keywords: Arduino Leonardo, program, device, HID, USB mouse, development.

В сообществе Arduino разработчиков особое место занимают платы на базе процессора ATmega32U4. Его особенность заключается в том, что он умеет эмулировать USB и выполнять роль HID (human interface device) устройства (клавиатура, мышь, геймпад). Благодаря чему энтузиасты превратили платы семейства Leonardo в настоящие полигоны для испытаний. Пример использования будет описан в статье.

Семейство плат Arduino Leonardo обширно, и представлено платами arduino leonardo (atmega32u4), arduino micro (atmega32u4), arduino pro micro (atmega32u4). Эти платы отличаются размерами, количеством контактов, разводкой питания; существуют многочисленные китайские клоны. В этой статье использована плата Arduino pro micro (рис. 1).

Целью исследования является разработка HID устройства с использованием Arduino Leonardo.

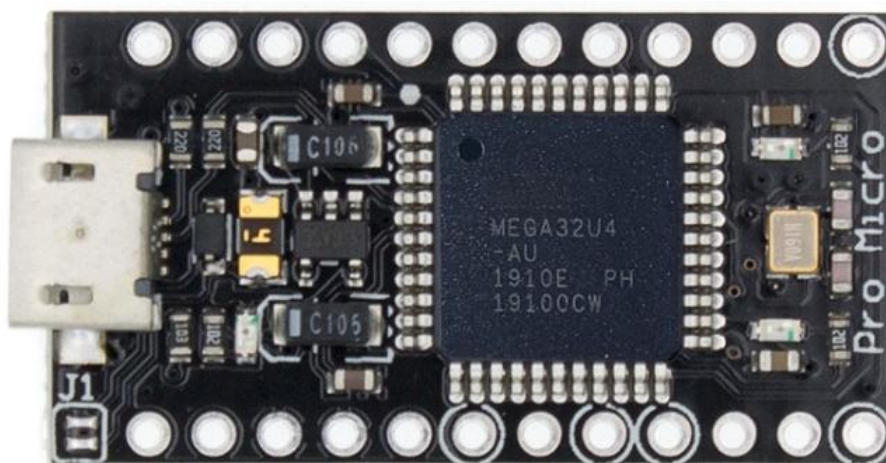


Рисунок 1 – Arduino pro micro

Основные принципы разработки на Arduino были рассмотрены Ю.Абдулахи [1]. Процесс установки IDE, прошивки простейших программ, работа с монитором порта, построение графиков рассмотрено в статье М.Fezari и А. Dahoud [2]. Исторический экскурс провёл В. Л. Шевченко, описав историю популяризации брэнда Arduino в России [3].

Вход в сообщество Arduino не сложен, начать разработку могут даже совсем новички. Существуют решения, позволяющие обходиться без кода в своих проектах, такие как Ardublock и miniBloq. Они позволяют строить код в виде блоков, что даёт доступ к программированию даже детям [4]. Среди всех плат, отдельное место занимает Arduino LilyPad, которая специально разработана специально для использования в умной одежде, история её возникновения описывает Ф.В. Патюченко в своей статье [5]. Одно из применений – медицинское. В статье Н.С. Бочкарёва предлагается использовать плату для удалённого наблюдения за состоянием пациентов больницы [6].

В этом проекте будет рассмотрено управление курсором на экране ПК. Манипулятором будет выступать джойстик-модуль. В нем есть также кнопка, то есть нажатием на джойстик можно эмулировать левую кнопку мыши. Сигнальные контакты подключаются к аналоговым выходам платы, кнопка – к цифровому, а для её подтяжки используется резистор на 10 кОм. При этом дополнительное питание к плате не требуется, питание из USB порта компьютера будет достаточно для конечного устройства.

Необходимые компоненты (Рис. 2):

1. Arduino Pro Micro;
2. 10 кОм, 5 Вт резистор;
3. Джойстик модуль;
4. Макетная плата;
5. Набор для пайки;
6. USB кабель.



Рисунок 2 – Необходимые компоненты

Полная принципиальная схема приведена на рис. 3.

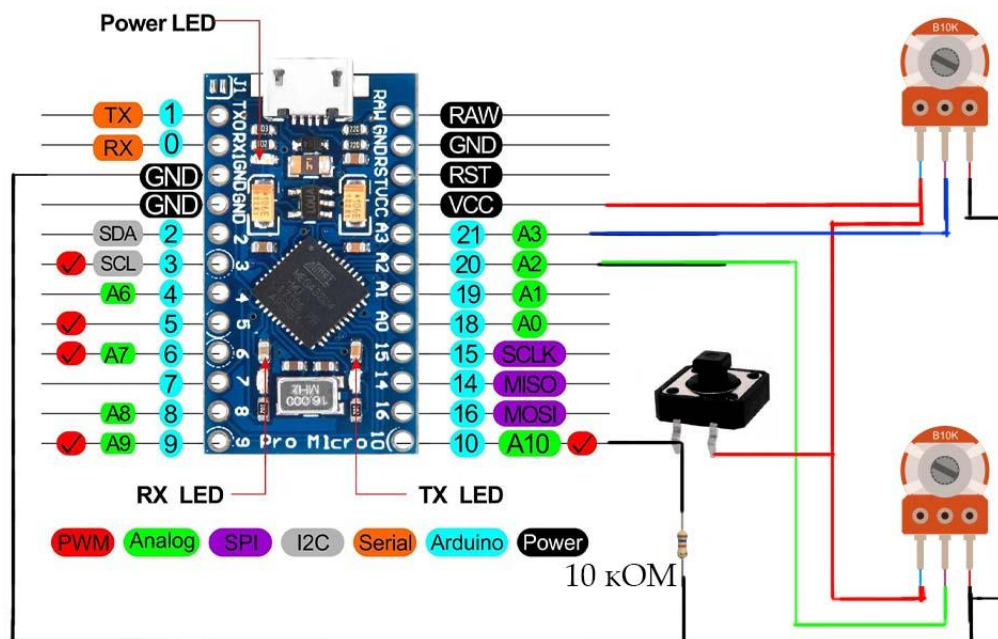


Рисунок 3 – Принципиальная схема

В принципиальной схеме нет изображения джойстика, как отдельного модуля, а показана его внутренние элементы – кнопка и два переменных резистора. В джойстике движение по оси изменяет сопротивление на одном из переменных резисторов (центральный контакт, остальные два на +5В и на массу). Именно это сопротивление и считывает микроконтроллер. В джойстике два таких резистора на каждую ось, зная сопротивления на каждом, можно узнать положение джойстика.

Кнопка подключена к 10 цифровому контакту. Её замыкание приводит к падению сопротивления, для подтяжки использован резистор на 10 кОм. Он нужен для того, чтобы во время размыкания кнопки на контакт не поступали случайные значения.

Собранное устройство изображено на рис. 4.

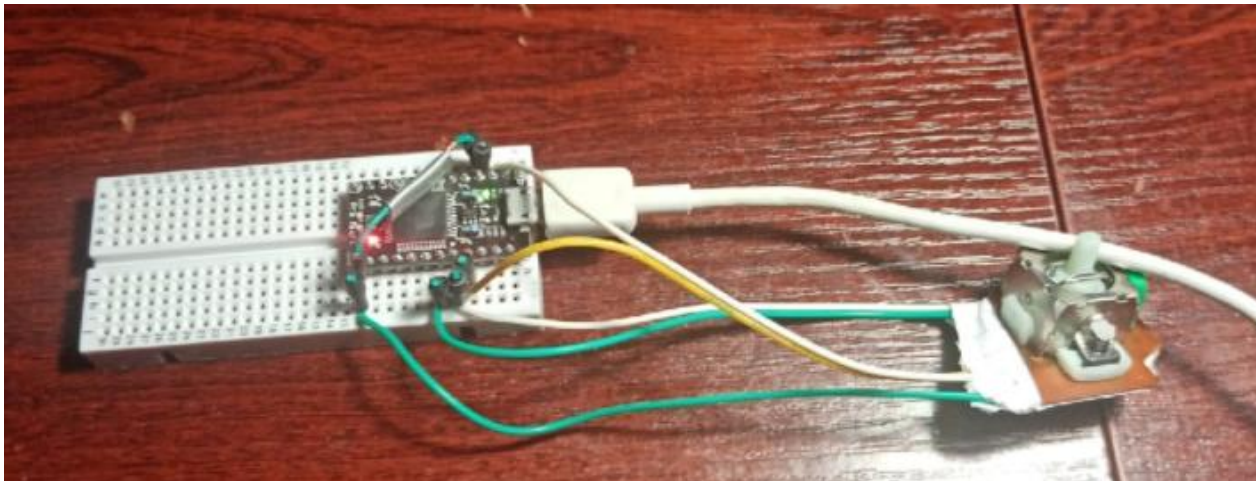


Рисунок 4 – Собранное устройство

Для проверки правильности сборки устройства можно воспользоваться встроенным в Arduino IDE [7] монитором порта. Проверка схемы через монитор порта позволяет выяснить корректно ли отображаются значения, получаемые с выводов платы. Для этого будет использован следующий код: (рис. 5)

```
const int buttonPin = 10; // здесь указываем пин, к которому подключена кнопка
int buttonState = 0; // указываем первоначальное состояние кнопки
void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(buttonPin, INPUT); // указываем что на 10 контакте находится кнопка ввода
}

void loop() {
  // put your main code here, to run repeatedly:
  int stickOneX = analogRead(A3); // считываем показания со стика
  int stickOneY = analogRead(A2);
  buttonState = digitalRead(buttonPin); // считывание состояние кнопки; возвращается либо low либо high

  if (buttonState == HIGH) {
    // выводим в монитор состояние кнопки
    Serial.print("кнопка зажата ");
  }
  else {
    // выводим в монитор состояние кнопки
    Serial.print("кнопка не зажата ");
  }

  //выводим состояние стика
  Serial.print(stickOneX);
  Serial.print(" ");
  Serial.println(stickOneY);
  delay(1);
}
```

Рисунок 5 – Код программы-отладчика

Вывод из монитора порта можно видеть на рис. 6.

кнопка зажата	0	264	кнопка не зажата	0	650
кнопка зажата	0	273	кнопка не зажата	0	648
кнопка зажата	0	276	кнопка не зажата	0	648
кнопка зажата	0	279	кнопка не зажата	0	646
кнопка зажата	0	288	кнопка не зажата	0	647
кнопка зажата	0	305	кнопка не зажата	0	645
кнопка зажата	0	322	кнопка не зажата	0	645
кнопка зажата	0	336	кнопка не зажата	0	646
кнопка зажата	0	341	кнопка не зажата	0	645
кнопка зажата	0	363	кнопка не зажата	0	644
кнопка зажата	0	366	кнопка не зажата	0	646
кнопка зажата	0	381	кнопка не зажата	0	649
кнопка зажата	0	392	кнопка не зажата	0	641
кнопка зажата	0	400	кнопка не зажата	0	642
кнопка зажата	0	403	кнопка не зажата	0	641
кнопка зажата	0	406	кнопка не зажата	0	638
кнопка не зажата	0	413	кнопка не зажата	75	639
кнопка не зажата	0	419	кнопка не зажата	136	635
кнопка не зажата	0	422	кнопка не зажата	141	629
кнопка не зажата	0	431	кнопка не зажата	153	630
кнопка не зажата	0	441	кнопка не зажата	159	635
кнопка не зажата	0	447	кнопка не зажата	164	636
кнопка не зажата	0	492	кнопка не зажата	171	633
кнопка не зажата	0	497	кнопка не зажата	186	632
кнопка не зажата	0	495	кнопка не зажата	197	619
кнопка не зажата	0	498	кнопка не зажата	204	624
кнопка не зажата	0	497	кнопка не зажата	213	620
кнопка не зажата	0	501	кнопка не зажата	222	606
кнопка не зажата	0	500	кнопка не зажата	242	605
кнопка не зажата	0	500	кнопка не зажата	266	602
кнопка не зажата	0	504	кнопка не зажата	278	605
кнопка не зажата	0	502	кнопка не зажата	284	602
кнопка не зажата	0	503	кнопка не зажата	297	593
кнопка не зажата	0	503	кнопка не зажата	304	592
кнопка не зажата	0	501	кнопка не зажата	313	591
кнопка не зажата	0	502	кнопка не зажата	317	593
кнопка не зажата	0	503	кнопка не зажата	325	585
кнопка не зажата	0	503	кнопка не зажата	336	562
кнопка не зажата	0	503	кнопка не зажата	349	561
кнопка не зажата	0	504	кнопка не зажата	350	572
кнопка не зажата	0	505	кнопка не зажата	357	521
кнопка не зажата	0	503	кнопка не зажата	363	515
кнопка не зажата	0	504	кнопка не зажата	367	514
кнопка не зажата	0	502	кнопка не зажата	374	508
кнопка не зажата	0	504	кнопка не зажата	380	507

Рисунок 6 – Вывод из монитора порта

Конечная версия кода на рис. 7.

```

1  #include "Mouse.h"
2  const int mouseButton = 10;    // указываем пин, на котором находится кнопка
3  const int xAxis = A2;         // пин, откуда считывать X координаты
4  const int yAxis = A3;         // пин, откуда считывать X координаты
5  // настройки для стика
6  int range = 12;               // выходной диапазон движения X или Y
7  int responseDelay = 5;        // задержка срабатывания мыши, в мс
8  int threshold = range / 4;    // порог покоя
9  int center = range / 2;       // значения положения покоя
10
11 void setup() {
12     Mouse.begin(); //запускаем эмуляцию
13 }
14
15 void loop() {
16
17     int xReading = readAxis(A2); //считываем X Y
18     int yReading = readAxis(A3);
19
20     Mouse.move(xReading, yReading, 0); // двигаем курсор в соответствии с тем, что считали
21     // реализация работы левой кнопки мыши
22     if (digitalRead(mouseButton) == HIGH) {
23         if (!Mouse.isPressed(MOUSE_LEFT)) {
24             Mouse.press(MOUSE_LEFT);
25         }
26     }
27     else {
28         if (Mouse.isPressed(MOUSE_LEFT)) {
29             Mouse.release(MOUSE_LEFT);
30         }
31     }
32
33     delay(responseDelay);
34 }
35
36
37
38 int readAxis(int thisAxis) {
39     int reading = analogRead(thisAxis);
40
41     reading = map(reading, 0, 1023, 0, range);
42
43     int distance = reading - center;
44
45     if (abs(distance) < threshold) {
46         distance = 0;
47     }
48
49     return distance;
50 }

```

Рисунок 7 – Конечная версия кода

В коде наличествуют пояснения, можно лишь уточнить, что с 38 строки прописана ситуация, когда значения, приходящие с джойстика, не меняются и ситуация, когда колебания сопротивления на переменных резисторах почти не меняется. Это позволяет избавиться от ложных срабатываний, и обеспечивает неподвижное положение указателя мыши.

Таким образом, было разработано устройство, частично эмулирующее мышшь компьютера на базе Arduino Leonardo. Возможна полная эмуляция, – плата позволяет подключить до 5 дополнительных кнопок.

Библиографический список

1. Yusuf A. The Working Principle Of An Arduino // ACADEMIA. 2014. С. 1-3. URL: <https://www.academia.edu/40296673> (дата обращения: 31.05.2021).
2. Mohamed F., Ali D. Integrated Development Environment “IDE” For Arduino // YA Badamasi 11th international conference. 2014. С. 1-2. URL: <https://www.researchgate.net/publication/328615543> (дата обращения: 31.05.2021).
3. Шевченко В.Л. Платформа Arduino как развитие познания людей в области электроники и программирования. // Человек и Вселенная, 2020. – № 1 (99). С. 41-44. URL: <https://elibrary.ru/item.asp?id=42956009> (дата обращения: 31.05.2021).
4. Pratomo A. B., Perdana, R. S. Arduviz, a visual programming IDE for arduino. // International Conference on Data and Software Engineering (ICoDSE). 2017. С. 4-6.
5. Патюченко Ф.В. ARDUINO LILYPAD // Modern Science, 2020. С. 329-331. URL: <https://elibrary.ru/item.asp?id=42316004> (дата обращения: 31.05.2021).
6. Бочкарёв Н.С. Разработка системы удаленного мониторинга за состоянием пациента // Информация и образование: границы коммуникаций. 2017. С. 95-96.
7. Arduino IDE. URL: <https://www.arduino.cc/en/software> (дата обращения: 31.05.2021).