

## РАЗРАБОТКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ УПРАВЛЕНИЯ РОТАЦИОННЫМ КОПРОМ

Романов Д.Б., студент 5 курса инженерного факультета  
Приказчиков В.С., студент 4 курса инженерного факультета  
Научный руководитель – Аюгин Н.П., кандидат технических наук,  
доцент  
ФГБОУ ВО Ульяновский ГАУ

***Ключевые слова:** программное обеспечение, ротационный копер, Arduino, реле, дисплей.*

*Работа посвящена разработке программного обеспечения, которое позволит управлять ротационным копром и считывать количество оборотов маховика после осуществления среза. Исследование выполнено за счет гранта Российского научного фонда № 24-26-00057, <https://rscf.ru/project/24-26-00057/>).*

**Введение.** Определение ударной прочности различных образцов, в том числе растительных материалов, осуществляют за счет использования маятниковых и ротационных копров. Наибольшее распространение получили маятниковые копры, поскольку им присуще простота конструкции, однако, изучение свойств материалов на скоростях выше 7...10 м/с проблематично из-за необходимости увеличения длины маятника.

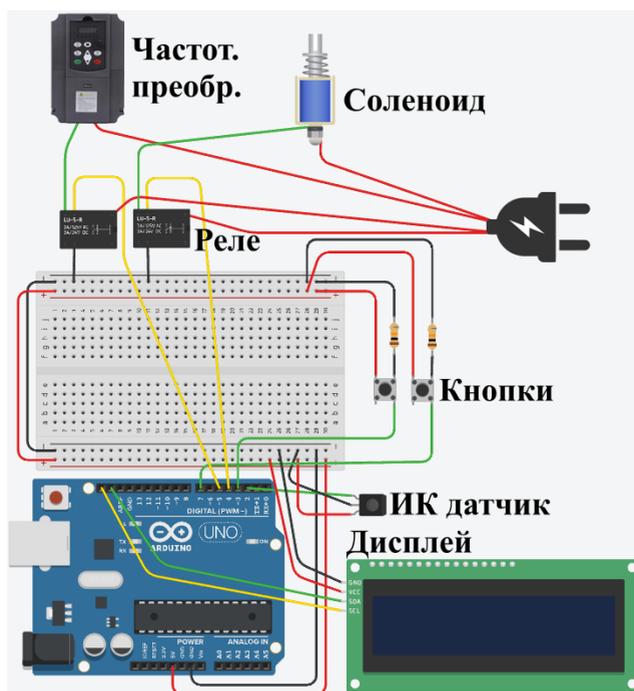
Методика исследований. Для изучения процесса резания растительных кормов на кафедре «Технология производства и ремонт машин» Ульяновского ГАУ был разработан ротационный копер [1]. Это устройство представляет собой установку, в которой нож закреплен на маховике, установленном на валу, крутящий момент на который передается от электродвигателя через обгонную муфту.

**Результаты исследований.** Для работы с ротационным копром необходимо специализированное программное обеспечение, управляющее подачей электропитания на соленоид и электродвигатель, а также позволяющее синхронизировать работу соленоида в

соответствии с положением ножа копра. Для этих целей на платформе Arduino был разработан блок управления ротационным копром, позволяющий включать электродвигатель и отключать его при достижении требуемой линейной скорости ножа, синхронизировать работу соленоида относительно положения маховика, на котором установлен нож, а также подсчитывать число оборотов маховика после отключения питания электродвигателя, в том числе после осуществления реза. Управление ротационным копром осуществляется по нажатию кнопок на пульте управления.

Блок управления реализован на платформе Arduino и включает в себя инфракрасный (ИК) датчик, LCD дисплей, две кнопки, блок реле и микроконтроллер Arduino UNO R3 [2-6].

Схема соединения представлена на рисунке 1.



**Рис. 1 – Схема соединения элементов установки управления ротационным копром**

Программирование контроллера Arduino UNO R3 было осуществлено в среде программирования Arduino IDE. Далее приведен программный код [2] работы управления ротационным копром с комментариями назначения основных команд кода (стейтментов).

```
#include <LiquidCrystal_I2C.h> // Импортирование библиотеки для LCD дисплея.
```

```
#define IR_SENSOR_PIN 2 // Пин инфракрасного датчика.
```

```
#define FIRST_BUTTON_PIN 3 // Пин 1-й кнопки.
```

```
#define SOLENOID_PIN 4 // Пин реле соленоида.
```

```
#define FREQ_CHANGER_PIN 5 // Пин реле частотного преобразователя.
```

```
#define SECOND_BUTTON_PIN 7 // Пин 2-й кнопки.
```

```
LiquidCrystal_I2C lcd(0x3F, 16, 2); // Создание объекта класса LCD дисплея.
```

```
int count; // Объявление переменной количества оборотов.
```

```
bool is_once; // Объявление переменной для проверки.
```

```
bool buttonWasUp1 = true; // Объявление переменной нажатия 1-й кнопки.
```

```
bool buttonWasUp2 = true; // Объявление переменной нажатия 2-й кнопки.
```

```
bool buttonIsUp1; // Объявление переменной нажатия 1-й кнопки.
```

```
bool buttonIsUp2; // Объявление переменной нажатия 2-й кнопки.
```

```
void rotations() { // Функция добавления оборотов в переменную.
```

```
if (digitalRead(IR_SENSOR_PIN)) { // Условие выполняется, когда ИК датчик не регистрирует метку на маховике.
```

```
is_once = true; // Обновление переменной для проверки.
```

```
} else if (is_once) { // Условие выполняется, когда переменная для проверки истинна.
```

```
count++; // Добавление 1 оборота в переменную.
```

```
is_once = false; // Обновление переменной для проверки.
```

```
}
```

```
}
```

```
void setup() {
```

```
pinMode(SOLENOID_PIN, OUTPUT); // Настройка пина реле соленоида на вывод.
```

```
pinMode(FREQ_CHANGER_PIN, OUTPUT); // Настройка пина
реле частотного преобразователя на вывод.
pinMode(FIRST_BUTTON_PIN, INPUT_PULLUP); // Настройка
пина 1-й кнопки на ввод и подтягивающий резистор.
pinMode(IR_SENSOR_PIN, INPUT); // Настройка пина ИК
датчика на ввод.
pinMode(SECOND_BUTTON_PIN, INPUT_PULLUP); //
Настройка пина 2-й кнопки на ввод и подтягивающий резистор.
is_once = true; // Обновление переменной для проверки.
lcd.init(); // Инициализация класса LCD дисплея.
lcd.setCursor(0, 0); // Установка курсора на 1-ю строку, 1-й
символ.
lcd.print("Rotations: "); // Вывод строки на LCD дисплей.
}
void loop(){
rotations(); // Запуск функции добавления оборотов.
lcd.setCursor(12, 0); // Установка курсора на 1-ю строку, 12-й
символ.
lcd.print(count); // Вывод количества оборотов на LCD дисплей.
buttonIsUp1 = digitalRead(SECOND_BUTTON_PIN); // Получение
состояния 2-й кнопки.
buttonIsUp2 = digitalRead(FIRST_BUTTON_PIN); // Получение
состояния 1-й кнопки.
if (!buttonIsUp1 == HIGH && !digitalRead(IR_SENSOR_PIN)) { //
Условие выполняется, когда нажата 2-я кнопка и ИК датчик
регистрирует метку.
digitalWrite(FREQ_CHANGER_PIN, LOW); // Отключение реле
частотного преобразователя.
digitalWrite(SOLENOID_PIN, HIGH); // Включение реле
соленоида.
count = count * 0; // Обнуление переменной количества оборотов.
lcd.clear(); // Очистка дисплея.
lcd.setCursor(0, 0); // Установка курсора на 1-ю строку, 1-й
символ.
lcd.print("Rotations: "); // Вывод строки на LCD дисплей.
}
```

```
if (buttonWasUp2 && !buttonIsUp2) { // Условие выполняется,  
когда нажата 1-я кнопка.  
    digitalWrite(FREQ_CHANGER_PIN, HIGH); // Включение реле  
частотного преобразователя.  
    digitalWrite(SOLENOID_PIN, LOW); // Отключение реле  
соленоида.  
}  
buttonWasUp1 = buttonIsUp1; // Обновление переменной,  
указывающей была ли нажата 2-я кнопка.  
buttonWasUp2 = buttonIsUp2; // Обновление переменной,  
указывающей была ли нажата 1-я кнопка.  
}
```

**Заключение.** Разработанное программное обеспечение, основанное на платформе Arduino, представляет собой оптимальное решение для управления ротационным копром. Это подчеркивает доступность и гибкость системы, обеспечивая простоту внедрения и настройки.

#### **Библиографический список:**

1. Аюгин, Н.П. Разработка установки для изучения процесса резания растительных кормов/ Н.П. Аюгин, Д.Б. Романов // XXVII Международная научно-производственная конференция "Вызовы и инновационные решения в аграрной науке". 12 апреля 2023 года. – Белгород: Белгородский ГАУ, 2023 – С.104-105.

2. Романов, Д.Б. Программное обеспечение удаленного управления гидропонной установкой / Д.Б. Романов // В мире научных открытий: материалы VII Международной студенческой научной конференции. 14-15 марта 2023 г. - Ульяновск: УлГАУ, 2023 - С. 2403-2406.

3. Романов, Д.Б. Алгоритм и программный код работы гидропонной установки / Д.Б. Романов // Наука в современных условиях: от идеи до внедрения: материалы Национальной научно-практической конференции с международным участием, посвященной 80-летию Ульяновского государственного аграрного университета имени П.А. Столыпина. 15 декабря 2022 г. - Ульяновск: УлГАУ, 2022 - С. 2678-2685.

---

4. Аюгин, Н.П. Разработка гидропонной установки / Н.П. Аюгин, Д.Б. Романов // XXVII Международная научно-производственная конференция "Вызовы и инновационные решения в аграрной науке". 12 апреля 2023 года. – Белгород: Белгородский ГАУ, 2023 – С.220-221.

5. Аюгин, Н.П. Программирование гидропонной установки / Н.П. Аюгин, Д.Б. Романов // XXVII Международная научно-производственная конференция "Вызовы и инновационные решения в аграрной науке". 12 апреля 2023 года. – Белгород: Белгородский ГАУ, 2023 – С.279-280.

6. Аюгин, Н.П. Удаленное управление гидропонной установкой/ Н.П. Аюгин, Д.Б. Романов // XXVII Международная научно-производственная конференция "Вызовы и инновационные решения в аграрной науке". 12 апреля 2023 года. – Белгород: Белгородский ГАУ, 2023 – С.277-278.

## **DEVELOPMENT OF SOFTWARE FOR THE MANAGEMENT OF A ROTARY IMPACT TESTER**

**Romanov D.B., Prikazchikov V.S.**  
**Scientific supervisor – Ayugin N.P.**  
**Ulyanovsk State Agrarian University**

**Keywords:** *software, rotary impact tester, Arduino, relay, display.*

*The work is devoted to the development of software that will allow you to control the rotary impact tester and read the number of revolutions of the flywheel after the cut. The research was carried out at the expense of a grant from the Russian Science Foundation № 24-26-00057, <https://rscf.ru/en/project/24-26-00057/>.*