

Состав скетча проекта „Skripysh”

/* Двухзвенное шасси „Skripysh” с колесной формулой 8x8, роликовым межколёсным приводом, балансирной подвеской и ломающейся рамой, на дистанционном управлении со смартфона.

```

Автор: Захаров Михаил
2022 год
*/

#define REMOTEXY_MODE__HARDSERIAL // Определение режима соединения и
подключение библиотеки RemoteXY.
#include <RemoteXY.h>

#define REMOTEXY_SERIAL Serial // Настройки соединения.
#define REMOTEXY_SERIAL_SPEED 9600

#pragma pack(push, 1) // Конфигурация интерфейса.
uint8_t RemoteXY_CONF[] =
{ 255,4,0,0,0,43,0,10,13,0,
  4,176,-6,46,71,14,2,26,1,1,
  82,2,16,20,2,31,226,134,145,0,
  1,1,82,41,16,20,2,31,226,134,
  147,0,3,131,11,9,32,12,2,26 };

struct { // Структура определяет все переменные и события
интерфейса управления.
  int8_t rudder; // Руль: 100 - -100 положение слайдера.
  uint8_t forward; // Вперёд: 1 если кнопка нажата, иначе 0.
  uint8_t back; // Назад: 1 если кнопка нажата, иначе 0.
  uint8_t speed; // Скорость: если переключатель в положении А равно 0, если в
положении В равно 1,если в положении С равно 2.

  uint8_t connect_flag; // 1 если есть подключение, иначе 0.
}

RemoteXY;
#pragma pack(pop)

#define PIN_FORWARD 2 // Пины подключения L298N к Arduino.
#define PIN_BACK 4
#define PIN_SPEED 3

#include <Servo.h>
Servo servoRudder;
int deg; // Переменная угла поворота руля.

void setup() {

```

```
RemoteXY_Init ();

pinMode (PIN_FORWARD, OUTPUT);
pinMode (PIN_BACK, OUTPUT);
pinMode (PIN_SPEED, OUTPUT);

servoRudder.attach(5);           // Пин подключения серво-мотора руля.
RemoteXY.rudder = 0;
}

void loop() {
  RemoteXY_Handler ();

  if (RemoteXY.speed == 0) {           // Управление скоростью.
    analogWrite (PIN_SPEED, 150);
  }
  if (RemoteXY.speed == 1) {
    analogWrite (PIN_SPEED, 200);
  }
  if (RemoteXY.speed == 2) {
    analogWrite (PIN_SPEED, 250);
  }

  digitalWrite(PIN_FORWARD, (RemoteXY.forward == 0) ? LOW : HIGH); // Назад и вперёд.
  digitalWrite(PIN_BACK, (RemoteXY.back == 0) ? LOW : HIGH);

  deg = RemoteXY.rudder;           // Управление рулём: 135 и 35 - это
  крайние правое и левое положения.
  deg = map (deg, 100, -100, 135, 35);
  servoRudder.write (deg);
}
```

Состав скетча проекта „RoboLift”

/* Роботизированный погрузо-разгрузочный подъемный комплекс "RoboLift"

Автор: Захаров Михаил

2022 год

*/

```
#include <Servo.h> // подключаем библиотеку для работы с сервоприводом
Servo servo1; // объявляем переменную Серво1 челюстного захвата
Servo servo2; // объявляем переменную Серво2 выдвигной платформы
Servo servo3; // объявляем переменную Серво3 наклона захвата
int angle1 = 93; // переменная для угла поворота Серво1 и её начальное значение
int angle2 = 12; // переменная для угла поворота Серво2 и её начальное значение
int angle3 = 75; // переменная для угла поворота Серво3 и её начальное значение

#include <Stepper.h> // подключаем библиотеку шагового мотора
Stepper LiftStepper(2048, 7, 5, 6, 4); // назначаем имя ШМ, кол-во шагов на 1об и пины
```

```
volatile boolean flag = false; // объявляем флаг
```

```
int led = 13 ; // назначаем пин светодиода
```

```
void setup () // объявляем начальные установки
```

```
{
  pinMode (led, OUTPUT); // переводим пин светодиода в режим вывода
  servo1.attach(8); // привязываем Серво1 к пину 8
  servo1.write(angle1); // задаем начальное положение угла поворота Серво1
  servo2.attach(9); // привязываем Серво2 к пину 9
  servo2.write(angle2); // задаем начальное положение угла поворота Серво2
  servo3.attach(10); // привязываем Серво3 к пину 10
  servo3.write(angle3); // задаем начальное положение угла поворота Серво3
  LiftStepper.setSpeed(15); // устанавливаем номинальную скорость ШМ
  attachInterrupt(0, detector, RISING); // задаем прерывание датчика на приближение магнита
}
```

```
void detector () // объявляем выполнение функции прерывания
```

```
{
  flag = true; // поднимаем флаг
}
```

```
void loop () // объявляем начало основного цикла работы
```

```
{
  if (flag) // условие, когда датчик сработал на появление магнитного поля
  {
    digitalWrite (led, HIGH); // включается (горит) светодиод
    delay(2000); // пауза 2 секунды
    zagruzkaFunction(); // вызываем функцию загрузки подъемника
    delay(2000); // пауза 2 секунды
    liftupFuction(); // вызываем функцию подъема кабины
    delay(2000); // пауза 2 секунды
    razgruzkaFunction(); // вызываем функцию разгрузки подъемника
  }
}
```

```

delay(2000); // пауза 2 секунды
liftdownFunction(); // вызываем функцию спуска кабины
flag = false; // цикл выполнится один раз после появления магнитного поля
}
{ // иначе, когда магнитного поля нет
digitalWrite (led, LOW); // светодиод выключается (не горит)
}
}

```

```

void zagruzkaFunction() //объявляем начало цикла загрузки подъемника

```

```

{
while (angle1 > 60) // челюстной захват плавно раскрывается
{
angle1--; // уменьшаем угол поворота Серво1 на 1 град.
servo1.write(angle1); // сообщаем Серво1 угол поворота
delay(20); // ждем 20 мс, пока Серво1 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle2 < 128) // платформа плавно выдвигается
{
angle2++; // увеличиваем угол поворота Серво2 на 1 град.
servo2.write(angle2); // сообщаем Серво2 угол поворота
delay(15); // ждем 15 мс, пока Серво2 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle1 < 93) // челюстной захват плавно закрывается
{
angle1++; // увеличиваем угол поворота серво1 на 1 град.
servo1.write(angle1); // сообщаем серво1 угол поворота
delay(20); // ждем 20 мс, пока Серво1 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle3 < 95) // захват плавно приподнимается
{
angle3++; // увеличиваем угол поворота Серво3 на 1 град.
servo3.write(angle3); // сообщаем Серво3 угол поворота
delay(20); // ждем 20 мс, пока Серво3 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle2 > 12) // платформа плавно задвигается
{
angle2--; // уменьшаем угол поворота Серво2 на 1 град.
servo2.write(angle2); // сообщаем Серво2 угол поворота
delay(15); // ждем 15 мс, пока Серво2 не сделает поворот
}
return; // возврат в основной цикл
}

```

```

void razgruzkaFunction() //объявляем начало цикла разгрузки подъемника

```

```

{
while (angle2 < 128) // платформа плавно выдвигается
{

```

```

    angle2++; // увеличиваем угол поворота Серво2 на 1 град.
    servo2.write(angle2); // сообщаем Серво2 угол поворота
    delay(15); // ждем 15 мс, пока Серво2 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle3 > 75) // захват плавно опускается
{
    angle3--; // уменьшаем угол поворота Серво3 на 1 град.
    servo3.write(angle3); // сообщаем Серво3 угол поворота
    delay(20); // ждем 20 мс, пока Серво3 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle1 > 60) // челюстной захват плавно раскрывается
{
    angle1--; // уменьшаем угол поворота Серво1 на 1 град.
    servo1.write(angle1); // сообщаем Серво1 угол поворота
    delay(20); // ждем 20 мс, пока Серво1 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle2 > 12) // платформа плавно задвигается
{
    angle2--; // уменьшаем угол поворота Серво2 на 1 град.
    servo2.write(angle2); // сообщаем Серво2 угол поворота
    delay(15); // ждем 15 мс, пока Серво2 не сделает поворот
}
delay(2000); // пауза 2 секунды
while (angle1 < 93) // челюстной захват плавно закрывается
{
    angle1++; // увеличиваем угол поворота серво1 на 1 град.
    servo1.write(angle1); // сообщаем серво1 угол поворота
    delay(20); // ждем 20 мс, пока Серво1 не сделает поворот
}
return; // возврат в основной цикл
}

```

```

void liftupFuction() // объявляем начало цикла подъема кабины
{
    LiftStepper.step(5504); // выполняем подъем кабины
    return; // возврат в основной цикл
}

```

```

void liftdownFuction() // объявляем начало цикла спуска кабины
{
    LiftStepper.step(-5504); // выполняем спуск кабины
    return; // возврат в основной цикл
}

```
