

Муниципальное бюджетное образовательное учреждение
Центр Дополнительного Образования «ИСТОКИ», Клуб «Омега»
Московская область, г/о Балашиха

**Всероссийская конференция «Юные техники и изобретатели» в
Государственной Думе Федерального Собрания Российской Федерации**

Конкурс: Освоение Арктики.

Тема: Поисково-исследовательский подводный робот «Микро МИР»

Автор работы: Севастьянов Иван Витальевич, 14 лет

Научный руководитель: Катунин Андрей Владимирович,
педагог дополнительного образования

Москва

2015

Содержание работы.

- I.** Введение.
- II.** Актуальность работы.
- III.** Цель работы.
- IV.** Состав работы.
- V.** Описание хода работы.
 - 1. Кинематическая схема робота.
 - 2. Компоновка и водоизмещение.
 - 3. Первая проблема. Течь места входа проводов в корпус.
 - 4. Вторая проблема. Разработка системы управления.
 - 5. Третья проблема. Уменьшение количества жил в проводе.
- VI.** Принцип работы устройства и его дополнительное оснащение.
- VII.** Анализ результатов.
- VIII.** Практические рекомендации и перспективы.
- IX.** Список литературы и программное обеспечение.
- X.** Приложение 1. Схема соединений робота.
- XI.** Приложение 2. Программа для микроконтроллера в работе.
- XII.** Приложение 3. Программа для микроконтроллера в пульте управления.

I. Введение.

В российской части арктического шельфа были обнаружены большие залежи углеводородов. В ближайшем будущем Россия начнёт полномасштабную добычу нефти и газа в Арктике. Разработка будет вестись с помощью морских буровых платформ. Для судоходства в Северном Ледовитом океане применяются специальные ледоколы, эксплуатация которых дороже обычных судов. При большой толщине льда, затруднительном судоходстве и относительно малом расстоянии до берега выгодно применять трубопроводы по дну моря от буровой платформы до материка.

В скором времени для транспортировки грузов будет окончательно освоен Северный Морской Путь, что существенно сократит транспортные расходы на пути от Дальнего Востока до Европы. Сейчас строятся северные порты, количество судов ледокольного класса увеличивается.

II. Актуальность работы.

Климат в арктических широтах очень суровый. Обычные задачи становятся гораздо сложнее при низких температурах. Например, как проверить подводную часть корпуса судна после столкновения с льдиной, если температура воды чуть выше нуля градусов? Как обследовать и устранить течи в трубопроводе на дне холодного моря? Как провести осмотр терпящей бедствие подводной лодки в северных широтах? В ледяной воде человек погибает за несколько десятков минут. Водолазы могут работать лишь небольшое количество времени. Суровый климат Арктики требует применения новых технических решений, например, подводных роботов. Я решил сделать такого робота на основе микроконтроллера Arduino Nano [2][4].

III. Цель работы.

Целью работы является создание небольшого робота с видеокамерой для подводных исследований в сложных условиях. Устройство должно быть

простым, недорогим и надёжным. Для передачи видео необходимо использовать обычную web-камеру, чтобы изображение было не только у оператора, но и передавалось через интернет в отдалённый координирующий центр.

IV. Состав работы.

Предлагаемая работа включает в себя описание этапов создания подводного робота, возникшие при работе три основных проблемы и их решение, анализ результатов, практические рекомендации и перспективы на будущее.

V. Описание хода работы.

1. Кинематическая схема робота.

Работа над проектом началась с определения способа погружения устройства. Изучив существующие конструкции лёгких подводных роботов, я остановился на четырёхмоторной схеме. Корпус робота решено было сделать из сантехнических пластиковых труб, закреплённых на каркасе из алюминиевых профилей. Герметичное соединение труб и полиуретановая платформа обеспечат положительную плавучесть робота в выключенном состоянии. Робот будет оснащён четырьмя моторами, каждый из которых через редуктор будет вращать свой ходовой винт. Вертикальные и горизонтальные моторы разместятся в трубах рядом друг с другом. При вращении горизонтальных моторов робот будет двигаться вперёд-назад, а при вращении вертикальных моторов - погружаться и всплывать (рис. 1).

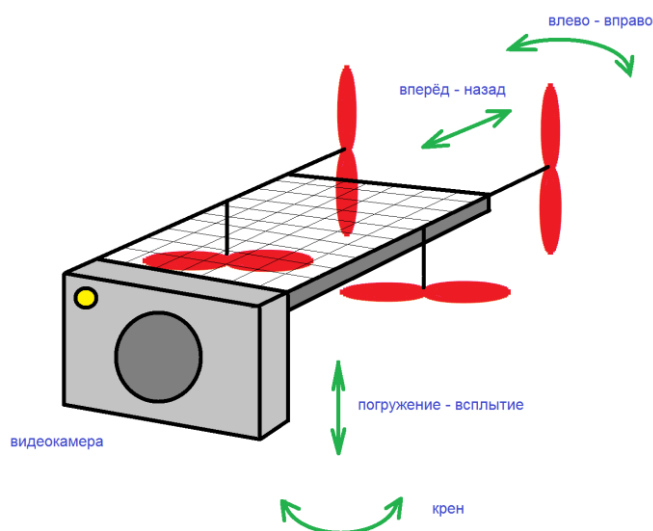


Рис.1. Кинематическая схема робота.

2. Компоновка и водоизмещение.

Если с принципом движения всё было понятно, то с компоновкой и водоизмещением робота пришлось работать долго.



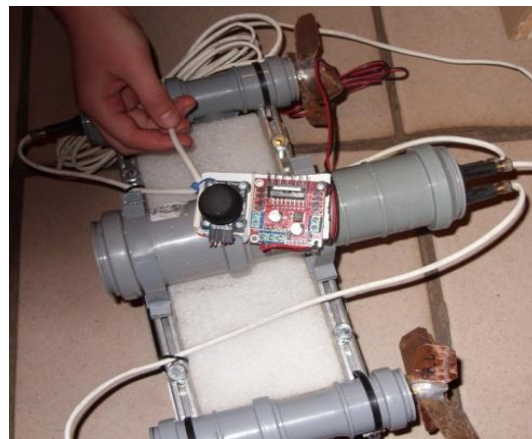
Рис. 2. Первый вариант робота.



Рис. 3. Второй вариант робота.

Первый вариант (рис.2) имел задние винты и очень большое водоизмещение. Мощности моторов с двухлопастными винтами не хватало, и скорость робота была очень медленной. Второй вариант (рис.3) был оснащён передними четырёхлопастными винтами. Этот вариант был легче, но его скорость всё равно была недостаточной.

Рис. 4. Третий вариант робота с простым джойстиком для испытаний.



В третьем варианте (рис.4) была полностью изменена конструкция робота. Единый корпус был разбит на 3 части: левый моторный отсек, правый моторный отсек и центральную часть для электроники. Масса робота значительно уменьшилась, скорость увеличилась. Появилась возможность добавить в конструкцию платформу для подвеса дополнительного оборудования весом до 100 граммов.

3. Первая проблема. Течь места входа проводов в корпус.

Проблема течи места входа проводов в корпус возникла еще в самом начале работы и сейчас возросла, т.к. между отсеками появились дополнительные провода. Решить эту задачу получилось таким образом. Я просверлил насквозь вместе с колпачком ниппель для автомобильного бескамерного колеса и вставил туда провод. Между ниппелем и колпачком надел на провод резиновое кольцо. В раскрученном виде провод свободно перемещается в ниппеле. При завинчивании колпачка кольцо деформируется и плотно обжимает провод (рис.5).



Рис. 5. Автомобильные ниппели. Уплотняющее кольцо на раскрученном ниппеле. Верхний ниппель запасной (для провода от будущих внешних датчиков).

4. Вторая проблема. Разработка системы управления.

Во время работы над конструкцией и определения ходовых качеств робота я применял простой способ управления (рис. 4). Одиночный джойстик был подключен так, что при его смещении вперед или назад вращались сразу два горизонтальных мотора, и робот двигался вперед или назад соответственно. Аналогично, при движении джойстика влево или вправо вращались сразу два вертикальных мотора, и робот просто всплывал или погружался. Робот не мог ни поворачивать, ни одновременно двигаться вперед с погружением или всплытием. Для полноценной работы устройства нужна система управления с двумя джойстиками, которая реализует все способы движения робота. Один джойстик должен отвечать за перемещения робота в горизонтальной плоскости, а другой – в вертикальной (рис.6).

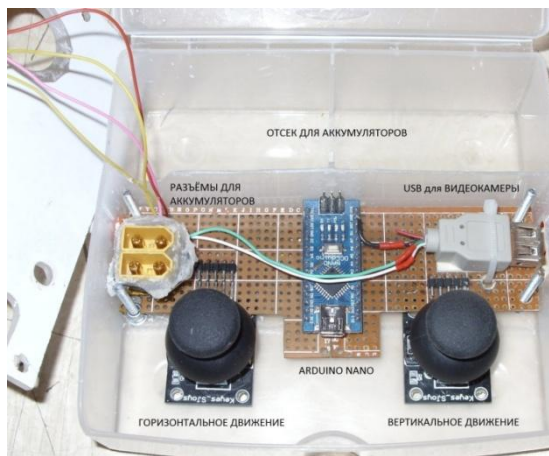
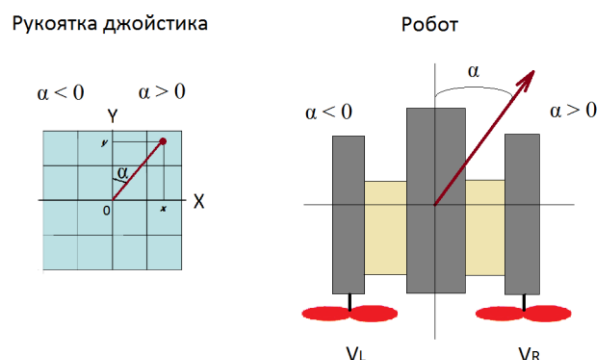


Рис.6. Пульт управления с 2 джойстиком.

Пришла очередь делать полноценную систему управления[1]. Здесь возникла вторая проблема. Нужна закономерность, по которой можно определить скорость моторов в зависимости от углов положения двух джойстиков. Я испытал много вариантов программы управления моторами, пока не вывел нужную формулу (рис.7 и рис 8.) и добился желаемого результата. После долгой настройки системы управления удалось обеспечить чёткие горизонтальные и вертикальные перемещения робота.

Рис. 7. Необходимое направление движения робота при отклонении джойстика на угол α . V_R и V_L - скорости левого и правого горизонтальных моторов соответственно.



$$\left\{ \begin{array}{l} V_R = \frac{\left(\frac{\pi}{2} - \alpha\right) V}{\frac{\pi}{2}} ; \text{ при } 0 \leq \alpha \leq \frac{\pi}{2} \\ V_L = V \end{array} \right.$$

$$\left\{ \begin{array}{l} V_L = \frac{\left(\frac{\pi}{2} - \alpha\right) V}{\frac{\pi}{2}} ; \text{ при } -\frac{\pi}{2} \leq \alpha < 0 \\ V_R = V \end{array} \right.$$

Рис. 8. Формулы зависимости скорости правого (V_R) и левого (V_L) горизонтальных моторов от угла α поворота джойстика и скорости V движения робота вперёд. Такая же зависимость и для вертикальных моторов.

5. Третья проблема. Уменьшение количества жил в проводе.

Установка web-камеры и улучшение пульта управления выявило третью проблему. Выяснилось, что провод от пульта к роботу должен иметь 2 линии на питание, 6 каналов от джойстиков и 4 линии для USB разъёма web-камеры. Такой двенадцатизильный провод слишком толстый, будет тянуть робота на дно и не сможет пройти сквозь ниппель (рис.9).

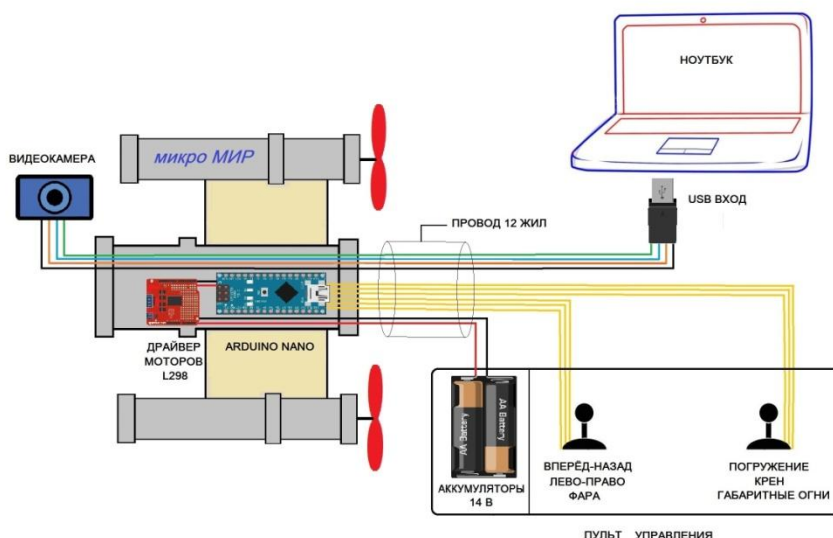


Рис.9.
Первоначальный провод с 12 жилами.

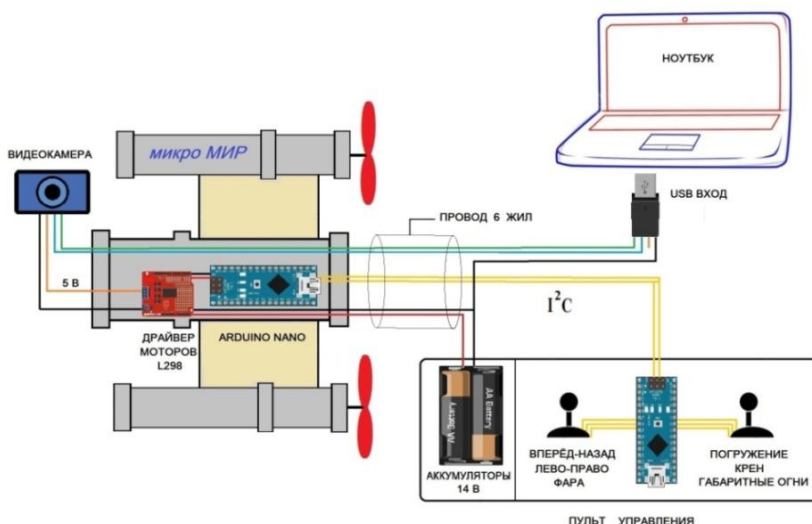


Рис.10.
Применение протокола I2C и оптимизация питания камеры сократили количество жил вдвое.

Для решения этой задачи я применил 2 микроконтроллера Arduino Nano: один установил в роботе, другой – в пульте управления. Среда разработки Arduino[2][4] имеет готовую библиотеку, позволяющую применять двухпроводной протокол передачи данных I2C[6]. Я организовал обмен

данными между двумя микроконтроллерами по этому протоколу. Это позволило передавать сигналы шести каналов джойстиков всего по двум проводам. Питание 5 вольт для USB линии web-камеры было взято от регулятора напряжения внутри робота. В итоге количество жил в проводе сократилось до 6, и нашёлся подходящий тонкий провод.

VI. Принцип работы устройства и его дополнительное оснащение.

Микроконтроллер Arduino в пульте считывает сигналы с каналов джойстика и посылает их по протоколу I2C. Arduino внутри робота принимает показания джойстиков и вычисляет необходимые для манёвра скорости вращения моторов. Сигналы на моторы усиливаются парой драйверов L298[5]. Робот дополнительно оборудован яркой светодиодной матрицей-фарой для обнаружения объектов и цветными габаритными огнями, чтобы его хорошо было видно под водой. Включение фары и габаритных огней происходит с помощью нажатия на левый и правый джойстик соответственно. Робот также оснащён платформой для подвеса дополнительного оборудования.

VII. Анализ результатов.

Разработанное устройство получило название «Микро МИР». «Микро МИР» - это четырёхмоторный подводный робот с дистанционным управлением, видеокамерой и фарой (рис.11). Может использоваться для подводных исследований в сложных условиях с трансляцией видеосигнала не только для оператора, но и через интернет для отдалённого координирующего центра.

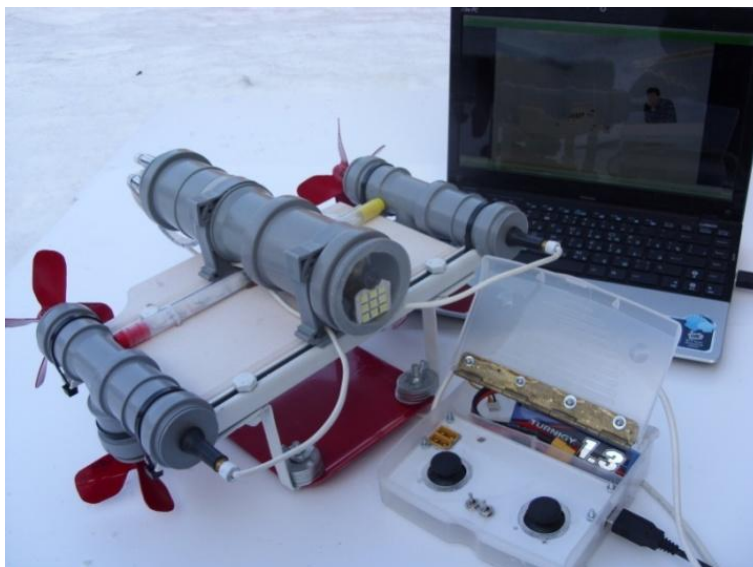


Рис. 11. Робот «Микро МИР» с пультом управления и ноутбуком.

Предварительные испытания робота были проведены в бассейне (рис. 12). Имитировалась работа по обнаружению подводной лодки, а также отработывались приёмы эксплуатации робота в темноте. Робот чётко выполнял все команды оператора. Производилась запись видео с камеры, а также передача изображения в интернет через Skype. Поставленные задачи были выполнены.

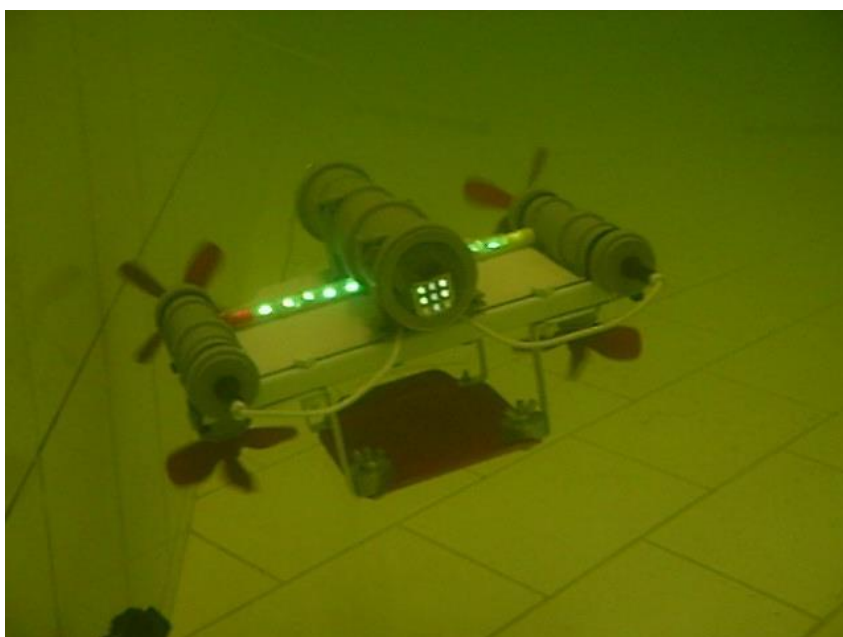


Рис. 12. Испытания в бассейне.

Вторые испытания проводились на открытой воде в начале апреля этого года. Низкая температура воды, лёд и холодная погода не повлияли на работу устройства. «Микро МИР» стабильно работал подо льдом на максимальном удалении от оператора.

Как любому новому устройству, роботу при дальнейшей эксплуатации предстоят доработки и усовершенствования. На данный момент была достигнута основная цель работы - создание простого подводного робота, который подтвердил свою работоспособность в холодных условиях.

VIII. Практические рекомендации и перспективы.

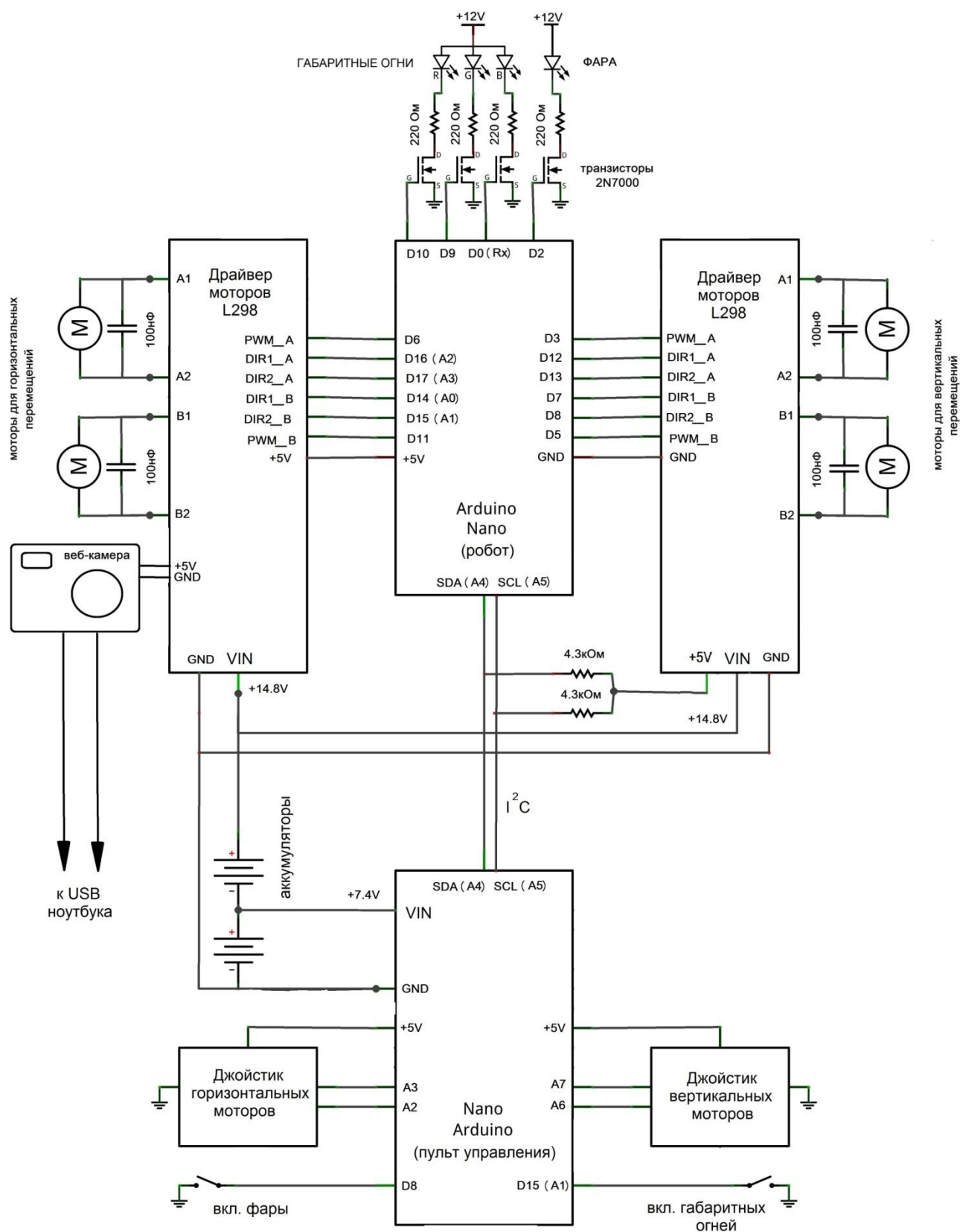
Применение фары в мутной воде приводило к ухудшению изображения на мониторе. Яркий свет отражался от мелких частичек, и молочная пелена ухудшала обзор. Я рекомендую не всегда включать фару. Есть задумка оснастить робота инфракрасной фарой. В этом случае видеокамера хорошо примет отраженное ИК излучение, и видимость улучшится.

Я планирую оснащение моего устройства датчиком-гироскопом. Это позволит добавить в систему управления стабилизацию, позволяющую роботу «зависать» в одной точке под водой. Такое свойство очень удобно для контроля мест утечек морских трубопроводов и упрощает работу оператора при подводных течениях. Добавление датчика-комаса, акселерометра, ультразвукового сонара позволит в дальнейшем применить режим «автопилот» для работы устройства.

IX. Литература.

1. В. Карпов, Т. Волкова. Управление роботом при движении по вектору / Лаборатория робототехники и искусственного интеллекта, Политехнический музей, Москва, 2010
2. У. Соммер. Программирование микроконтроллерных плат Arduino. / С.-Петербург, БХВ-Петербург, 2012
3. С. А. Филиппов. Робототехника для детей и родителей (стр. 178-192) / С.-Петербург, Наука, 2011
4. Б. Эванс. Arduino блокнот программиста./ 2007.
5. J.-D. Warren, J. Adams, H. Molle. Arduino robotics (стр. 1-51, 57-62)/ Apress, 2011
6. Phillips Semiconductors Co. The I2C-bus specification, version 2.1 / The Netherlands, 2000
7. Среда разработки программ Arduino IDE v.1.5.4.
8. Среда разработки технической документации для электроники Fritzing.2013.07.27.

Схема соединений робота Микро МИР (Fritzing.2013.07.27)[8].



Программа для микроконтроллера в работе (Arduino IDE v1.5.4)[7].

```
#include <Wire.h>                                // подключение библиотеки протокола I2C
#define R_MOTOR_PWM 6                            // pin D6 скорость правого горизонтального мотора
#define R_MOTOR_DIR1 16                          // pin D16(A2) направление 1 правого горизонтального мотора
#define R_MOTOR_DIR2 17                          // pin D17(A3) направление 2 правого горизонтального мотора
#define L_MOTOR_DIR1 14                          // pin D14(A0) направление 1 левого горизонтального мотора
#define L_MOTOR_DIR2 15                          // pin D15(A1) направление 2 левого горизонтального мотора
#define L_MOTOR_PWM 11                           // pin D11 скорость левого горизонтального мотора
#define R_MOTORUPDOWN_PWM 3                      // pin D3 скорость правого вертикального мотора
#define R_MOTORUPDOWN_DIR1 12                   // pin D12 направление 1 правого вертикального мотора
#define R_MOTORUPDOWN_DIR2 13                   // pin D13 направление 2 правого вертикального мотора
#define L_MOTORUPDOWN_DIR1 7                     // pin D7 направление 1 левого вертикального мотора
#define L_MOTORUPDOWN_DIR2 8                     // pin D8 направление 2 левого вертикального мотора
#define L_MOTORUPDOWN_PWM 5                     // pin D5 скорость левого вертикального мотора
#define FARA 2                                    // pin D2 включение фары
#define RED 10                                   // pin D10 включение красного цвета габаритных огней
#define GREEN 9                                  // pin D9 включение зелёного цвета габаритных огней
#define BLUE 0                                   // pin D0(Rx) включение синего цвета габаритных огней

int Joystick_Address = 3;                       // I2C адрес Arduino в пульте управления
int fb, lr, ud, rot, swr, swl;                  // переменные вперёд-назад, лево-право, вверх-вниз, крен, прав. кнопка дж., лев. кнопка дж.
boolean flagFara, flagGab;                     // флаг включения фары, флаг включения габаритных огней
byte flagColor, cntFara, cntGab;               // флаг переключения цвета габаритных огней
int cnt;                                       // счётчик переключения цвета габаритных огней
byte Delt=2;                                  // зона нечувствительности джойстиков

void setup(){                                  // ----- блок инициализации -----
    Wire.begin();                             // включение библиотеки I2C
    pinMode(R_MOTORUPDOWN_PWM,OUTPUT);
    pinMode(L_MOTORUPDOWN_PWM,OUTPUT);
    pinMode(R_MOTOR_PWM,OUTPUT);
    pinMode(L_MOTORUPDOWN_DIR1,OUTPUT);
    pinMode(L_MOTORUPDOWN_DIR2,OUTPUT);
    digitalWrite(R_MOTORUPDOWN_PWM,LOW);
    digitalWrite(L_MOTORUPDOWN_PWM,LOW);        // инициализация портов микроконтроллера
    digitalWrite(R_MOTOR_PWM,LOW);
    digitalWrite(L_MOTORUPDOWN_DIR1,LOW);
    digitalWrite(L_MOTORUPDOWN_DIR2,LOW);
    for (byte i=11; i<=17; i++){
        pinMode(i,OUTPUT);
        digitalWrite(i,LOW);
    }
    pinMode(FARA,OUTPUT);
    pinMode(RED,OUTPUT);
    pinMode(GREEN,OUTPUT);
    pinMode(BLUE,OUTPUT);
    digitalWrite(FARA,LOW);
    digitalWrite(RED,LOW);
    digitalWrite(GREEN,LOW);
    digitalWrite(BLUE,LOW);
}                                               // ----- конец блока инициализации -----

void loop(){                                  // ----- исполняемый блок -----
    getJoystickValues();                      // запрос о положении джойстиков по протоколу I2C
    delay(50);                               // задержка 50 миллисекунд
    fb = map( fb, 0, 1023, 15, -15);          // аналоговое показание левого джойстика вперёд-назад(fb) масштабируется в пределах от -15 до 15
    lr = map( lr, 0, 1023, 15, -15);          // аналоговое показание левого джойстика лево-право(lr) масштабируется в пределах от -15 до 15
    ud = map( ud, 0, 1023, 15, -15);          // аналоговое показание прав. дж. погружение-всплытие(ud) масштабируется в пределах от -15 до 15
    rot = map(rot, 0, 1023, 15, -15);          // аналоговое показание правого джойстика крен(rot) масштабируется в пределах от -15 до 15
    byte modul_fb = abs(fb);
    byte modul_lr = abs(lr);                  // значения модулей скоростей
    byte modul_ud = abs(ud);
    byte modul_rot= abs(rot);

    if(swl == 0) {
        if(cntFara >= 2) { flagFara=!flagFara; digitalWrite(FARA,flagFara); cntFara=0;}
        cntFara++; }                        // вкл.-выкл. фары при нажатии на левую кнопку джойстика

    if(swr == 0) {
```

```

        if(cntGab >= 2){flagGab=!flagGab; cntGab=0;}
        cntGab++; } // вкл.-выкл. габаритов при нажатии на правую кнопку джойстика
if (flagGab){
    digitalWrite (BLUE,flagColor==0);
    analogWrite(GREEN+(flagColor>1),(255 - cnt)*(flagColor!=0)); //переключение цвета габаритных огней
    cnt=cnt+25;
    if (cnt==275){
        flagColor++;
        flagColor=flagColor%3;
        cnt=0;
    }
}
else{
    digitalWrite(RED,LOW);
    digitalWrite(GREEN,LOW);
    digitalWrite(BLUE,LOW);
}
if ( modul_fb < Delt ){
    if( modul_lr > Delt ){ // если скорость fb (вперёд-назад) внутри зоны нечувствит. джойст. +-Delta,
        digitalWrite(R_MOTOR_DIR1, lr < 0); // а скорость lr (лево-право) не внутри +-Delta, то
        digitalWrite(R_MOTOR_DIR2, lr > 0); // поворачиваем на месте со скоростью lr
        digitalWrite(L_MOTOR_DIR1, lr > 0);
        digitalWrite(L_MOTOR_DIR2, lr < 0);
        analogWrite(R_MOTOR_PWM, modul_lr*17);
        analogWrite(L_MOTOR_PWM, modul_lr*17);
    }
    else{ // если -Delt< скорость lr (лево-право) <Delt,
        digitalWrite(R_MOTOR_PWM,LOW); // и -Delt< скорость fb (вперёд-назад) <Delt,
        digitalWrite(L_MOTOR_PWM,LOW); // то стоим
    }
}
else if( modul_lr < Delt ){ // если скорость fb (вперёд-назад) не внутри +-Delta,
    digitalWrite(R_MOTOR_DIR1, fb < 0); // а скорость lr (лево-право) внутри +-Delta, то
    digitalWrite(R_MOTOR_DIR2, fb > 0); // винты вращаем вперёд или назад со скоростью fb
    digitalWrite(L_MOTOR_DIR1, fb < 0);
    digitalWrite(L_MOTOR_DIR2, fb > 0);
    analogWrite(R_MOTOR_PWM, modul_fb*17);
    analogWrite(L_MOTOR_PWM, modul_fb*17);
}
else{
    float alfa=abs(atan(lr/fb)); // общий случай, скорости fb и lr не внутри +-Delt
    float medleno =(HALF_PI-alfa)*modul_fb/HALF_PI; // вычисление скорости притормаживающего винта
    digitalWrite(16-2*(lr > 0),(fb * medleno) < 0); // по углу альфа отклонения джойстика
    digitalWrite(17-2*(lr > 0),(fb * medleno) > 0);
    digitalWrite(16-2*(lr < 0),fb < 0);
    digitalWrite(17-2*(lr < 0),fb > 0);
    analogWrite(6+5*( lr > 0),int(abs(medleno))*17); // один винт притормаживает со скоростью medleno
    analogWrite(6+5*( lr < 0),modul_fb*17); // другой винт вращается со скоростью fb
}
if ( modul_ud < Delt ){
    if( modul_rot > Delt ){ // если скор. ud (погруж.-всплытие) внутри зоны нечувствит. джойст. +-Delta,
        digitalWrite(R_MOTORUPDOWN_DIR1, rot < 0); // а скорость rot (крен) не внутри +-Delta, то
        digitalWrite(R_MOTORUPDOWN_DIR2, rot > 0); // поворачиваем на месте со скоростью rot
        digitalWrite(L_MOTORUPDOWN_DIR1, rot > 0);
        digitalWrite(L_MOTORUPDOWN_DIR2, rot < 0);
        analogWrite(R_MOTORUPDOWN_PWM, modul_rot*17);
        analogWrite(L_MOTORUPDOWN_PWM, modul_rot*17);
    }
    else{ // если -Delt< скорость rot (крен) <Delt,
        digitalWrite(R_MOTORUPDOWN_PWM,LOW); // и -Delt< скорость ud (погружение-всплытие) <Delt,
        digitalWrite(L_MOTORUPDOWN_PWM,LOW); // то стоим
    }
}
else if( modul_rot < Delt ){ // если скорость ud (погружение-всплытие) не внутри +-Delta,
    digitalWrite(R_MOTORUPDOWN_DIR1, ud < 0); // а скорость rot (крен) внутри +-Delta, то
    digitalWrite(R_MOTORUPDOWN_DIR2, ud > 0); // винты вращаем вперёд или назад со скоростью ud
    digitalWrite(L_MOTORUPDOWN_DIR1, ud < 0);
    digitalWrite(L_MOTORUPDOWN_DIR2, ud > 0);
    analogWrite(R_MOTORUPDOWN_PWM, modul_ud * 17);
    analogWrite(L_MOTORUPDOWN_PWM, modul_ud * 17);
}
else {
    float alfa=abs(atan(rot/ud)); // общий случай, скорости ud и rot не внутри +-Delt
    float medleno =(HALF_PI-alfa)*modul_ud/HALF_PI; // вычисление скорости притормаживающего винта

```

```

digitalWrite(12-5*(rot > 0),(ud * medleno) < 0);          // по углу альфа отклонения джойстика
digitalWrite(13-5*(rot > 0),(ud * medleno) > 0);
digitalWrite(12-5*(rot < 0),ud < 0);
digitalWrite(13-5*(rot < 0),ud > 0);
    analogWrite(3+2*( rot > 0),int(abs(medleno)) * 17);    // один винт притормаживает со скоростью medleno
    analogWrite(3+2*( rot < 0),modul_ud * 17);            // другой винт вращается со скоростью ud
}
// ----- конец исполняемого блока -----

void getJoystickValues(){
    Wire.requestFrom(Joystick_Address,10);                // функция запроса на получение 10 байтов от Arduino в пульте управления
    while(!Wire.available()) {}
    fb = Wire.read();
    fb = fb << 8;
    fb |= Wire.read();
    lr = Wire.read();
    lr = lr << 8;
    lr |= Wire.read();
    ud = Wire.read();
    ud = ud << 8;
    ud |= Wire.read();
    rot = Wire.read();
    rot = rot << 8;
    rot |= Wire.read();
    swr = Wire.read();
    swl = Wire.read();
}

```


Программа для микроконтроллера в пульте управления (Arduino IDE v1.5.4)[7].

```

#include <Wire.h>                                // подключение библиотеки протокола I2C
#define FRONT_BACK A3                          // аналоговый вход A3 с Y-потенциометра  левого джойстика
#define LEFT_RIGHT  A2                          // аналоговый вход A2 с X-потенциометра  левого джойстика
#define UP_DOWN     A7                          // аналоговый вход A7 с Y-потенциометра  правого джойстика
#define ROTATION    A6                          // аналоговый вход A6 с X-потенциометра  правого джойстика
#define SW_R        A1                          // кнопка правого джойстика
#define SW_L        8                          // кнопка левого джойстика

int Joystick_Address = 3;                       // I2C адрес Arduino пульта управления
int fb, lr, ud, rot;
byte val[10];

void setup(){
    Wire.begin(Joystick_Address);               //-----  блок инициализации  -----
    Wire.onRequest(requestEvent);               // соединение по протоколу I2C в форме ожидания запроса
    pinMode(SW_R, INPUT_PULLUP);               // инициализация цифровых портов кнопок
    pinMode(SW_L, INPUT_PULLUP);               //-----  конец блока инициализации  -----
}

void loop(){
    fb = analogRead(FRONT_BACK);                //-----  исполняемый блок  -----
    val[0] = (fb & 0xff00) >> 8;
    val[1] = fb & 0xff;
    lr = analogRead(LEFT_RIGHT);                // разбиение 2-х байтовых переменных (int) fb, lr, ud и rot
    val[2] = (lr & 0xff00) >> 8;                // на одиночные байты
    val[3] = lr & 0xff;
    ud = analogRead(UP_DOWN);
    val[4] = (ud & 0xff00) >> 8;
    val[5] = ud & 0xff;
    rot = analogRead(ROTATION);
    val[6] = (rot & 0xff00) >> 8;
    val[7] = rot & 0xff;
    val[8] = digitalRead(SW_R);                 // добавление байтов со значениями портов кнопок
    val[9] = digitalRead(SW_L);                 //-----  конец исполняемого блока  -----
}

void requestEvent(){
    Wire.write(val,10);                         // функция, выполняемая при запросе от Arduino в работе
}

```