

Описание системы «Автономный гексапод»

Подготовил Yauheni Liauchuk unstutby@gmail.com

24.01.2019

Содержание

Вступление	2
Обзор системы.....	2
Механическая платформа	2
Плата управления сервомашинками.....	2
Компьютер	3
Лидар.....	3
Питание	3
Программное обеспечение	3
Структурная схема	4
Механическая платформа	4
Используемые термины	4
Размеры и углы.....	5
Плата управления сервомашинками.....	6
Сервомашинки.....	6
Калибровка	6
Лидар.....	7
Компьютер	7
Вход по VNC.....	7
Программное обеспечение	7
Координатная система.....	7
Исходный код	7
Краткое описание классов.....	8
Полезная информация.....	8
Принцип работы сервомашинок.....	8
Прошивка платы управления	8
Пример работы над улучшениями	9

Вступление

Система «Автономный гексапод» (далее просто «Паук», несмотря на количество ног) предназначена для обучения робототехнике. На момент написания этой документации система представляет собой работающее сочетание механической, электронной и программной частей, которое позволяет «Пауку» самостоятельно перемещаться по помещению и избегать препятствий. Возможна модификация любой из этих составляющих – добавление новых и замена старых частей системы, изменение алгоритмов. Для успешной работы над дальнейшей модификацией «Паука» заинтересованным студентам желательно представлять, как работает вся система в целом, и как работает часть, которую вы хотели бы модифицировать. Цель этого документа – дать такое представление, чтобы упростить и ускорить начало работы с «Пауком». Описаны составляющие системы, их взаимодействие, принципы работы, а также некоторые наблюдения и советы автора; даны ссылки на инструкции, справочные материалы и полезные статьи.

Обзор системы

Механическая платформа



Сверху закреплён [лидар](#).

Плата управления сервомашинками



механических устройств на основе сервомашинки (например, клешни-захвата).

В системе используется платформа [Lynxmotion CH3-R](#) – гексапод с тремя степенями свободы в каждой конечности (т.е., 6 ног с 3 суставами, всего 18 суставов). Каждый из суставов может быть повернут на произвольный угол, за счёт чего можно реализовать различные алгоритмы походки, поворотов и т.п. Суставы представляет собой сервомашинки модели HS-475/[HS-485](#). Внутри платформы имеется свободное пространство, в котором располагается электроника и аккумуляторы. Для удобного и аккуратного крепления всех внутренних частей изготовлена пластиковая рама.

Управление 18-ю сервомашинками одновременно – достаточно нетривиальная задача. Плата управления сервоприводами [SSC-32](#) берёт на себя эту задачу и существенно упрощает программирование «Паука». Плата управления SSC-32 принимает команды через последовательный интерфейс (UART, 5v). В команде, в частности, можно указать, на какой угол должен быть повернут конкретный сустав и с какой скоростью он должен повернуться в это положение. Плата может управлять 32 сервомашинками, для конечностей «Паука» использовано только 18, так что возможно простое добавление дополнительных механических устройств на основе сервомашинки (например, клешни-захвата).

Компьютер



что для разработки удобнее использовать удалённый доступ через VNC.

Лидар



Основные алгоритмы (обработка данных с лидара, походка) выполняются на одноплатном компьютере [Raspberry Pi 3 B](#). Компьютер работает под управлением операционной системы Raspbian (Linux). Мощность компьютера избыточна для текущих задач и позволяет даже подключить к нему камеру (имеется в лаборатории, использовалась в предыдущей версии системы «Паук») и обрабатывать видео в реальном времени. К компьютеру можно подключить монитор, клавиатуру, мышь, но это может потребовать разборки корпуса «Паука», так

Сверху платформы закреплён лидар [SLAMTEC RPLIDAR A1](#). Лидар имеет вращающуюся часть, в которой расположен инфракрасный лазер и фотодиодная линейка, с помощью которых лидар может измерять расстояние до препятствия в которое попадает луч. Максимальное расстояние, измеряемое данным лидаром – 12 метров. Связь с компьютером осуществляется через UART интерфейс (через переходник UART<->USB).

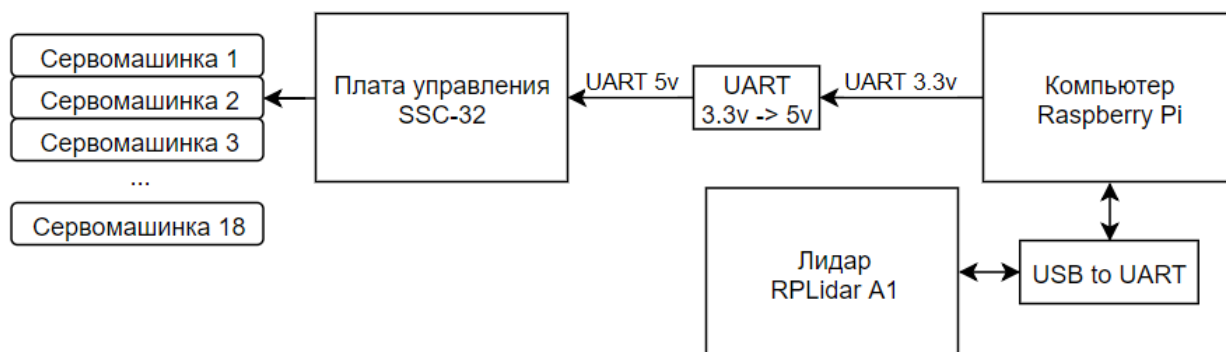
Питание

Питание системы осуществляется от LiPo аккумулятора 11.1 В. Плата управления SSC-32 требует питания напряжением 6 В (определяется сервомашинками, при использовании других сервомашинки напряжение может быть другим). Компьютер Raspberry Pi и лидар питаются напряжением 5 В. Для эффективного понижения напряжения аккумулятора в систему включен DC-DC преобразователь.

Программное обеспечение

Программа, управляющая «Пауком», написана на языке C++. Разработка происходит в IDE Code::Blocks непосредственно в компьютере Raspberry Pi. При работе программа постоянно принимает данные от лидара и вырабатывает команды для движения «Паука».

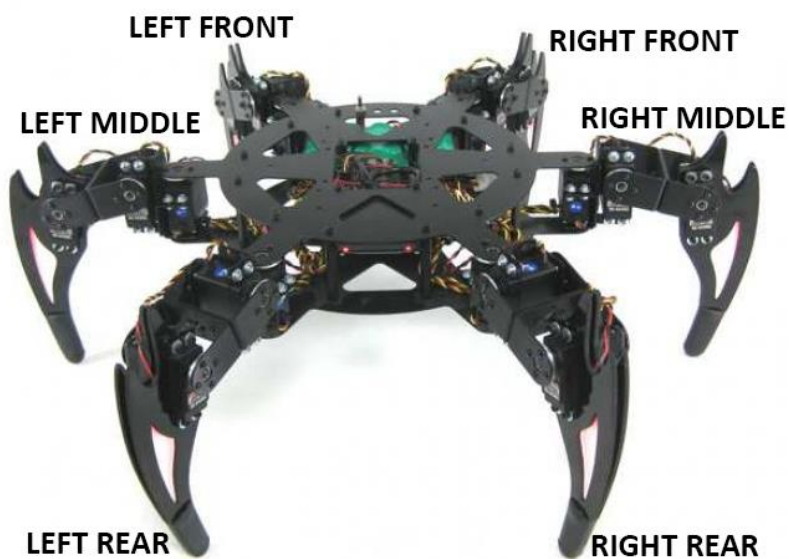
Структурная схема



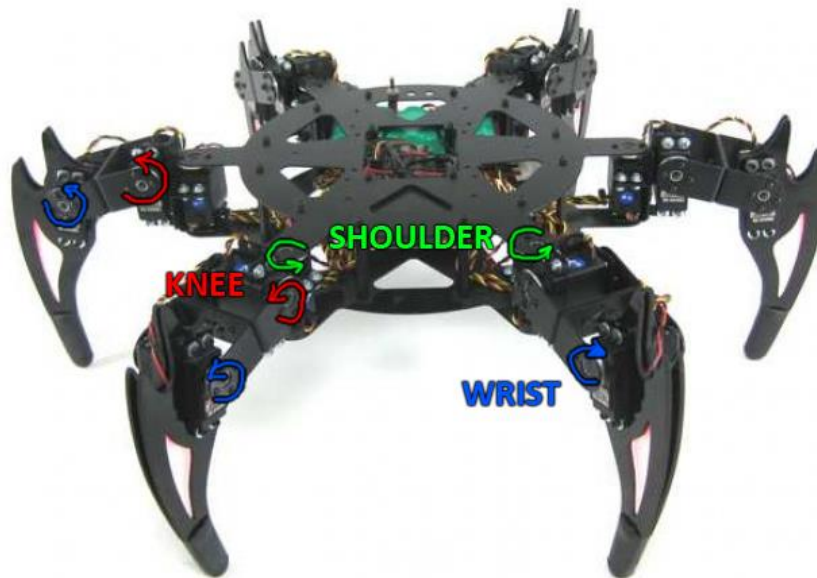
Механическая платформа

Используемые термины

В коде программы и конфигурационном файле используются следующие названия для ног «Паука»:

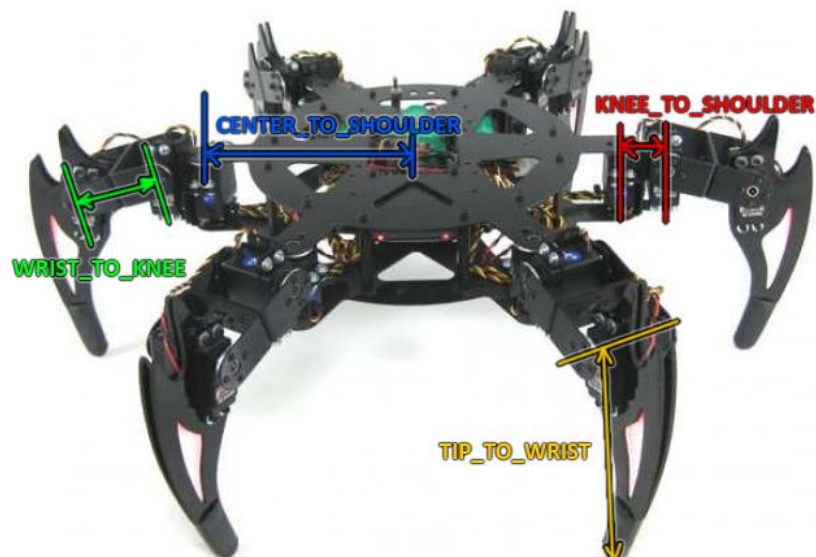


Для обозначения конкретных суставов используются следующие названия:



Размеры и углы

Чтобы переместить конец ноги (tip) в заданную точку пространства, необходимо вычислить углы поворота всех суставов этой ноги. Для этого необходимо знать некоторые размеры. Эти размеры были измерены и сохранены в конфигурационном файле (config.txt в папке программы).



Размеры в миллиметрах:

```
TIP_TO_WRIST=140
WRIST_TO_KNEE=56
KNEE_TO_SHOULDER=28
CENTER_TO_SHOULDER=166
KNEE_TO_BOTTOM=50
```

Плата управления сервомашинками

Исчерпывающую документацию по плате SSC-32 можно найти на сайте производителя:

<http://www.lynxmotion.com/images/data/ssc-32.pdf> . Важно: в документации отсутствует информация о бинарных командах, которые используются в «Пауке» для экономии пропускной способности UART-интерфейса. Информация об этих командах находится тут:

<http://www.lynxmotion.com/images/html/build177.htm>

Каналы управления нумеруются от 0 до 31 (подписаны на плате). Соответствие между номером канала и конкретным суставом задаётся в конфигурационном файле, например:

```
LEFT_FRONT_WRIST_CHANNEL=0
LEFT_FRONT_KNEE_CHANNEL=1
LEFT_FRONT_SHOULDER_CHANNEL=2
```

Сервомашинки

Калибровка

Сервомашинки могут быть закреплены внутри ног «Паука» в произвольном положении, поэтому для корректной работы необходима калибровка, т.е. установление соответствия длительностей импульсов (см. [принципы работы сервомашинки](#)) и углов поворота суставов. Также каждая сервомашинка может быть подключена к любому из 32 каналов платы управления. Соответствие задаётся в конфигурационном файле. Максимальные и минимальные углы поворота суставов задаются для всех конечностей сразу:

```
KNEE_MIN_ANGLE=35
KNEE_MAX_ANGLE=135
WRIST_MIN_ANGLE=20
WRIST_MAX_ANGLE=100
SHOULDER_MIN_ANGLE=-90
SHOULDER_MAX_ANGLE=90
```

Далее для каждого сустава(сервомашинки) указываются соответствующие длительности и номер канала:

```
LEFT_FRONT_WRIST_CHANNEL=0
LEFT_FRONT_WRIST_MIN=900
LEFT_FRONT_WRIST_MAX=1700

LEFT_FRONT_KNEE_CHANNEL=1
LEFT_FRONT_KNEE_MIN=2000
LEFT_FRONT_KNEE_MAX=1000

LEFT_FRONT_SHOULDER_CHANNEL=2
LEFT_FRONT_SHOULDER_MIN=550
LEFT_FRONT_SHOULDER_MAX=2370
```

При замене неисправной сервомашинки на новую необходимо опытным путём установить новые длительности (отправляя команды на плату управления добиться нужного угла) и скорректировать информацию в конфигурационном файле.

Лидар

Общее описание, характеристики:

http://bucket.download.slamtec.com/8e7a1f4490a235717b43fccaf7dcae325dda7dc8/LD108_SLAMTEC_rplidar_datasheet_A1M8_v2.1_en.pdf

Инструкция по использованию:

http://bucket.download.slamtec.com/b42b54878a603e13c76a0a0500b53595846614c6/LR001_SLAMTEC_rplidar_protocol_v1.1_en.pdf

Компьютер

Вход по VNC

Для входа на удалённый рабочий стол через VNC используется логин root и пароль по умолчанию raspberry. В случае каких-либо проблем с подключением через сеть для разрешения ситуации можно подключить к Raspberry Pi клавиатуру и монитор.

Программное обеспечение

Координатная система

Все размеры внутри программы исчисляются в миллиметрах. Углы могут исчисляться в радианах (при любых тригонометрических расчётах), градусах (**только** в конфигурационном файле) и в нормализованном диапазоне от 0 до 1 (минимальный/максимальный возможный угол поворота сустава). Направления отсчёта нормализованных углов поворота суставов соответствует [рисунку с названиями суставов](#). Координатная система, принятая в программе, выглядит следующим образом:



Исходный код

Все исходные коды, актуальные на момент написания документации, доступны тут: <https://github.com/levevg/hexapod>

В паке Firmware находится код программы, выполняемой непосредственно на компьютере «Паука». Для работы с этим проектом используется IDE Code::Blocks. Программа написана на языке C++, поэтому для дальнейшей работы с пауком необходимы хотя бы базовые навыки программирования на C++ и понимание ООП.

В папке Simulation находится код программы, которая была создана для возможности проверки алгоритмов «Паука» на персональном компьютере (без самого «Паука»). Симулятор состоит из двух частей – Hexarod (практически полностью совпадает с проектом Firmware, симулирует ПО «Паука») и Environment (симулирует «железо» «Паука» и помещение). Для работы с этими проектами используется IDE CLion или любая другая, поддерживающая CMake. Для успешной компиляции проекта Environment необходимо скачать и скомпилировать библиотеку freeglut.

Краткое описание классов

Config – класс для загрузки данных из конфигурационного файла

Serial – класс для передачи и приёма данных через последовательный интерфейс

Joint – класс сустава (сервомашинки) с возможностью задавать угол поворота в диапазоне от 0 до 1 (калибровочные данные загружаются в классе Config)

Limb – класс конечности с возможностью перемещать конец ноги в заданные декартовы координаты (расчёт необходимых для этого углов).

Hexarod – класс, собирающий вместе все суставы и конечности, и предоставляющий функции для ходьбы, подъёма и т.п.

Behavior – базовый класс алгоритма поведения «Паука»

SquareTestBehavior – простой пример класса поведения. «Паук» перемещается по траектории в виде квадрата.

LidarBehavior – класс поведения на основе информации от лидара. Пытается идти вперёд избегая препятствий.

LidarAPI – функции для приёма и передачи данных лидара

Полезная информация

Принцип работы сервомашинки

<http://www.lynxmotion.com/images/data/ssc-32.pdf>, страница 4

<http://easyelectronics.ru/servomashinka-hs-311.html>

Прошивка платы управления

Исходные коды прошивки микроконтроллера платы SSC-32 отсутствуют в свободном доступе. В случае выхода микроконтроллера из строя (такое уже однажды случилось) необходимо заменить его и прошить в него bootloader (его можно найти в папке SSC-32 Bootloader репозитория с исходным кодом проекта). После этого перемычки на плате нужно установить для связи с компьютером через COM порт, подключить её к компьютеру и прошить firmware используя программу производителя (её, а так же прошивку можно найти на сайте <http://www.lynxmotion.com/c-15-software.aspx>). Помните, что версия

прошивки должна поддерживать режим двоичных команд (на сайте производителя имеются прошивки не поддерживающие этот режим).

Пример работы над улучшениями

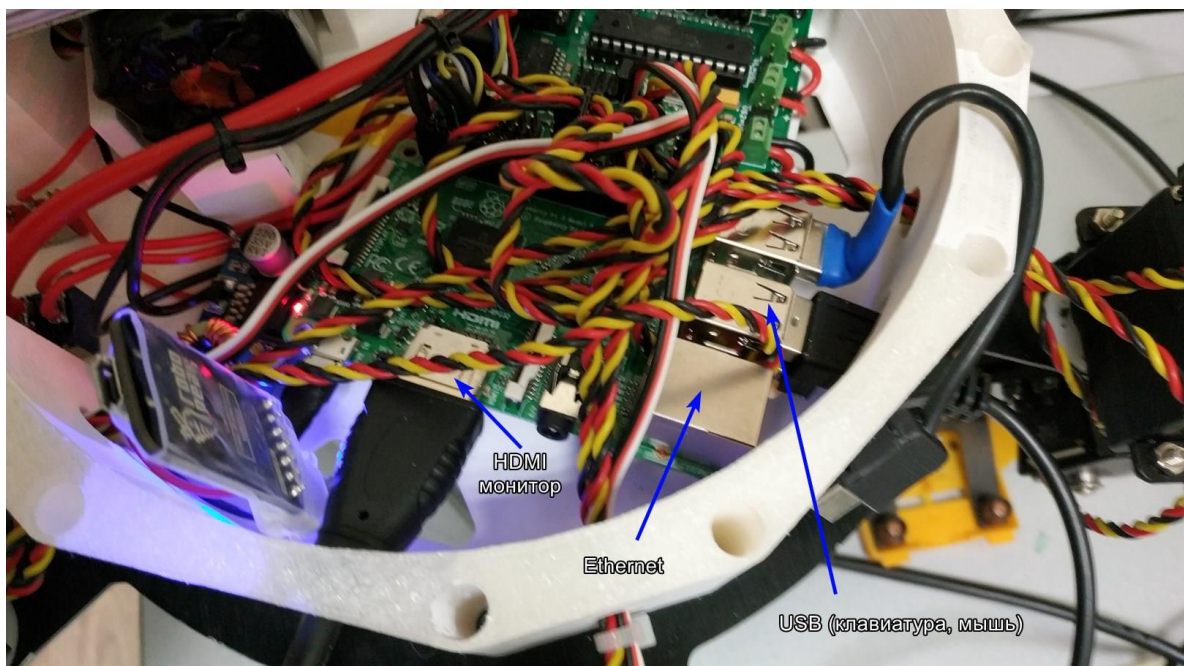
Задача: добавить «Пауку» управляемую клешню на основе дополнительных сервомашинок.

Пошаговое описание порядка работы:

Шаг 1: Необходимо установить связь с компьютером «Паука», на котором происходит программирование.

Прежде всего, на компьютер «Паука» Raspberry Pi необходимо подать питание. Питание компьютера и питание платы управления (и, соответственно, сервомашинок) в «Пауке» раздельное. На данном этапе лучше включить только компьютер.

Способ 1: разобрать корпус «Паука» чтобы получить доступ к разъёмам компьютера Raspberry Pi, подключить к нему монитор (HDMI), клавиатуру, мышь (USB), сеть по необходимости (Ethernet). Способ неудобен из-за необходимости разбирать «Паука» и невозможности полностью проверить его работу не собирая его обратно. К этому способу лучше прибегать только если по каким-то причинам не получается подключиться способом 2.



Способ 2: удалённое подключение VNC через Wi-Fi. При включении питания после загрузки операционной системы компьютер «Паука» автоматически пытается подключиться к Wi-Fi сети с SSID (идентификатор сети) *Hexapod* с паролем *password*. После подключения Raspberry Pi становится доступен в этой сети по адресу *192.168.0.51*. Если для работы с пауком используется ноутбук с Windows 10, такую WiFi сеть можно организовать непосредственно при помощи ноутбука. Для этого в командной строке необходимо выполнить следующие команды (с администраторскими правами):

```
netsh wlan set hostednetwork mode=allow ssid=Hexapod key=password
```

(установка сети, достаточно один раз).

```
netsh wlan start hostednetwork
```

(включение сети, каждый раз при необходимости)

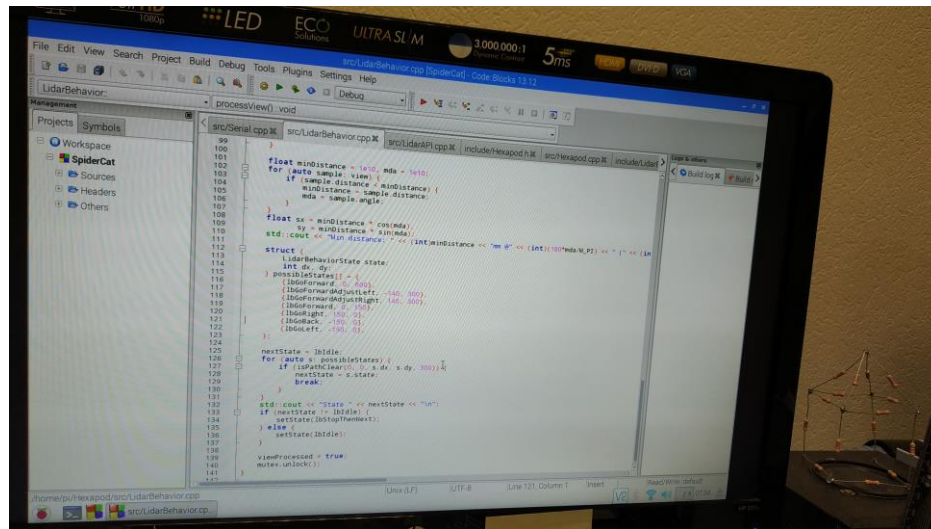
```
netsh wlan stop hostednetwork
```

(выключение сети при необходимости)

После того, как Raspberry Pi подключен к сети, к нему можно подключиться по VNC. Для этого можно использовать бесплатную программу VNC Viewer. Имя пользователя для подключения - *pi*, пароль *raspberry*.

Шаг 2: Доработка программы.

Теперь, используя удалённый рабочий стол VNC, можно редактировать программу паука. Для этого необходимо запустить среду разработки Code::Blocks IDE и открыть проект (путь к проекту */home/pi/Hexapod/SpiderCat.cbproj*).



Чтобы добавить пауку дополнительную сервомашинку, добавим в классе Нехарод новое свойство *Joint *claw*. В конструкторе этого класса также необходимо добавить инициализацию этой переменной (файл Нехарод.h), указав номер канала платы управления, к которому подключена сервомашинка, и минимальную и максимальную длительности импульса для этой сервомашинки (в миллисекундах):

```
hexapod.cpp  src/LidarBehavior.cpp  src/Joint.cpp  *include/Hexapod.h

} Geometry;

struct SequencePoint;
struct SequencePoint {
    float value;
    uint16_t time;
};

class Hexapod
{
public:
    Joint *claw;

    Hexapod() { _behavior = nullptr; claw = new Joint(19, 500, 2500); }

    void setBehavior(AbstractBehavior *behavior);
    void setGeometry(Geometry g);
    const Geometry *geometry() const { return &geometry; }
```

Для примера сделаем чтобы «Паук» открывал и закрывал «клешню» когда стоит на месте. Для этого откроем файл LidarBehavior.cpp, в котором задаётся текущее поведение «Паука» (версия с лидаром).

В функции fsm, реализующей конечный автомат, допишем обработку состояния lbIdle (состояние, в котором «Паук» стоит на месте):



```
19     std::thread readThread(&LidarBehavior::serialReadThreadFunc, this)
20     readThread.detach();
21 }
22
23 void LidarBehavior::die() {
24     _lidar->sendCommand(lcStop);
25     _alive = false;
26 }
27
28 void LidarBehavior::fsm(uint32_t stateTime) {
29     processView();
30     switch (_state) {
31         case lbRaise:
32             _hexapod->raiseSequence();
33             delayMs(500);
34             setState(lbIdle);
35             break;
36         case lbIdle:
37             _hexapod->claw->move(0, 300);
38             delayMs(300);
39             _hexapod->claw->move(1, 300);
40             delayMs(300);
41             break;
42         case lbGoForward:
43             moveY = 30;
44             break;
45         case lbGoForwardAdjustLeft:
46             moveY = 30;
```

Шаг 3: Подключение сервомашинки.

Для подключения новой сервомашинки необходимо разобрать «Паука» чтобы получить доступ к плате управления. Сервомашинка может быть подключена к любому из свободных каналов платы (всего их 32, ноги «Паука» используют только 18). Следите за цветами проводов чтобы не перепутать ориентацию разъёма при подключении.

Шаг 4: Запуск, проверка.

Теперь можно включить питание сервомашинки, скомпилировать и запустить программу. Если всё сделано правильно, «Паук», стоящий на месте (когда препятствия не дают возможности идти ни в одном направлении) будет непрерывно двигать новой сервомашинкой от минимального к максимальному положению.

При перезагрузке компьютера программа запускается автоматически.