



# BEGINNERS GUIDE



## ПОЗДРАВЛЯЕМ!

**Вы приобрели, получили в подарок, либо собрали своего собственного робота MIRO!**

**Мы, разработчики робота, надеемся, что с этого момента вы окунетесь в огромный мир технологий робототехники и в конце концов присоединитесь к нам!**

Этот документ будет предельно лаконичен. Но достаточен для того, чтобы вы начали свое путешествие.

Основным источником информации про MIRO является официальный сайт: [mirorobot.ru](http://mirorobot.ru). На этом сайте в разделе «Ресурсы» находятся ссылки на все материалы для создания, использования и разработки под MIRO:

- Карта производства робота;
- Инструкция по сборке;
- Методические рекомендации с уроками;
- PDF-версия этого документа (MIRO BEGINNERS GUIDE);
- Исходные коды прошивки для ESP8266;
- Библиотека MIRO для Arduino с API и примерами;
- Образы для Raspberry Pi;
- Образы виртуальных машин VirtualBox;
- Подготовленные файлы и инструкции для модификации Arduino IDE.

Также на сайте имеется ссылка на группу сообщества MIRO в GITTER, где вы можете задать вопросы, найти единомышленников и поделиться своими успехами.

Специальный раздел сайта «Проекты» позволяет загружать и оформлять свои наработки и решения, связанные с MIRO и делиться ими с другими людьми.

Мы не будем дублировать общую информацию с сайта, но дадим некоторые пояснения, которых нет на сайте.

## **И начнем мы с архитектуры аппаратной части робота.**

В основе робота MIRO лежит связка из трех популярных микропроцессорных систем:

ARDUINO (на базе Atmel ATMEGA328, ядро AVR), ESP8266 и Raspberry Pi (какая именно версия сейчас не важно).

Робот MIRO питается от аккумулятора с номинальным напряжением питания 7,4В (Li-Ion 2S). Для получения 5В, требуемых большинству микропроцессорных устройств робота используется преобразователь на микросхеме LM2596S.

В базовой комплектации, всеми датчиками и исполнительными устройствами управляет ATMEGA328.

ESP8266 позволяет осуществлять беспроводное программирование чипа ATMEGA328 из среды ARDUINO IDE, а также служит мостом SERIAL-WIFI, позволяющим обмениваться данными с последовательным портом (UART) ATMEGA328 по беспроводному каналу по протоколу TELNET. Очевидно, что ATMEGA328 и ESP8266 соединены по линиям RX/TX. Кроме того, для осуществления беспроводной прошивки чипа ATMEGA328, линия RESET этого чипа соединена с линией GPIO2 микроконтроллера ESP8266.

Таким образом, даже без Raspberry Pi, MIRO содержит в себе развитую и удобную архитектуру для разработки.

В базовой комплектации робота с модулем SENS1 к микроконтроллеру ATMEGA328 подключаются следующие датчики и устройства:

1. Цифровые датчики одометров (левого и правого колеса)
2. Драйвер левого и правого двигателей на микросхемах L9110S
3. Ультразвуковой датчик линии HC-SR04
4. Фоторезистивные датчики освещенности (левый и правый)
5. Пьезоизлучатель звука
6. Светодиоды передней подсветки (левый и правый)
7. Датчики линии (левый, правый и центральный)
8. Серводвигатель наклона камеры

Устройства 3-7 являются опциональными и могут быть заменены на какие угодно датчики или устройства. Вы можете спроектировать и изготовить свой собственный модуль взамен модуля SENS1 со своими задачами и устройствами.

Устройства 1, 2 и 8 являются обязательными для функционирования робота и не предполагают отключения. Оно и понятно – без драйверов двигателей робот не сможет перемещаться, а без датчиков одометрии вы не сможете ничего сказать о характере перемещения робота и не сможете качественно управлять его перемещением.

Плата Raspberry Pi подключена к ATMEGA328 по интерфейсу I2C (TWI). К плате Raspberry Pi подключена камера RaspiCam.

Полную коммутационную схему робота MIRO вы можете скачать из соответствующего репозитория по ссылке на официальном сайте [mirorobot.ru](http://mirorobot.ru).

## Программная модель и среда разработки.

Важнейшей ценностью, которую вы получаете с MIRO является готовые сценарии разработки.

### Сценарий 1. Разработка под ARDUINO.

При этом сценарии вы можете полностью отключить Raspberry Pi для экономии заряда аккумулятора.

Подготовка программного обеспечения разработчика.

1. Пройдите по ссылке в репозиторий с готовыми дополнениями для среды ARDUINO IDE с предустановленными библиотеками и инструментами беспроводной прошивки для MIRO:  
[https://github.com/fbezruchko/miro\\_arduino\\_ide\\_dfu](https://github.com/fbezruchko/miro_arduino_ide_dfu)  
Следуйте инструкции, приведенным в README файле репозитория.
2. Скачайте и установите MIRO API – библиотеку для ARDUINO:  
[https://github.com/fbezruchko/miro\\_library\\_arduino](https://github.com/fbezruchko/miro_library_arduino).  
Следуйте инструкциям в README файле репозитория.

Почти все готово!

3. Включите питание робота. В пространстве робота должна появиться новая сеть WiFi с названием что-то вроде MIRO\_xxxxxx (где “xxxxxx” – последние цифры MAC-адреса приемопередатчика WiFi микроконтроллера ESP8266 в шестнадцатеричной системе счисления). Подключитесь со своего компьютера к этой сети.
4. В браузере в адресной строке наберите адрес 192.168.240.1 и перейдите на страницу настройки параметров беспроводного подключения.
5. Если у вас есть дома или в офисе своя сеть WiFi выберите сеть, к которой требуется подключиться и задайте пароль этой сети. Подключите робота к вашей сети. Через некоторое время у вас обновиться статус подключения. Запомните или запишите куда-нибудь IP-адрес робота в вашей сети. Если окно браузера не обновилось, ничего страшного – мы скоро узнаем IP-адрес робота другим способом. В дальнейшем, робот будет автоматически подключаться к Вашей сети WiFi с этим IP-адресом. Либо вы можете зафиксировать выделенный роботу адрес, либо любой другой в пространстве вашей сети как статический.
6. Откройте среду разработки ARDUINO IDE, и в ней откройте исходный код примера из библиотеки MIRO – `miro_firmware_arduino`
7. Выберите в списке плат ARDUINO UNO WIFI.
8. Выберите в списке портов строчку с IP-адресом вашего робота. Вот тут вы и сможете узнать этот адрес – если у вас в сети одна плата ARDUINO UNO WIFI (что скорее всего), то соответствующая запись в списке будет единственной.
9. Загрузите прошивку примера. Если все прошло удачно, после прошивки робот должен непременно моргнуть три раза каждым светодиодом. В случае наличия подключенного серводвигателя подвеса камеры – еще и подвигать камерой. После того как оба светодиода потухнут – робот готов принимать команды пользователя.
10. Далее нам понадобится терминальная программа. Если вы пользователь ОС Windows, мы рекомендуем вам воспользоваться популярной бесплатной утилитой PUTTY. Пользователям ОС Linux будет достаточно любой терминальной программы или командной строки.
11. Наберите в строке адреса терминала IP-адрес робота с указанием порта 23. Например, если IP-адрес робота 192.168.1.15, то следует набрать: [192.168.1.15@23](tel:192.168.1.15@23)
12. В случае удачного подключения, нажмите пару раз клавишу ENTER, чтобы отчистить буфер ввода. В результате в терминальной программе будет строка вида:

```
Version = v1.0.0
STATUS: OK
```

показывающая, готовность робота.

Теперь вы сможете подать ему команды. Наберите:

```
miroset -d 0 1 255
```

и робот включит левый светодиод. Чтобы включить правый светодиод наберите:

`miroset -d 1 1 255`

*В загруженном вами примере реализован протокол управления роботом, позволяющий с помощью несложных текстовых команд по TELNET управлять роботом и его устройствами. Полное описание протокола приведено в разделе Wiki репозитория: [https://github.com/fbezruchko/miro\\_library\\_arduino/wiki](https://github.com/fbezruchko/miro_library_arduino/wiki)*

С этого момента вы можете модифицировать любой из имеющихся примеров в библиотеке MIRO, либо написать свой собственный код.

Библиотека MIRO – это API, включающее все необходимые функции и интерфейсы для простой и быстрой разработки. У MIRO пока еще совсем молодое сообщество, но мы надеемся, что со временем, множество людей будут принимать активное участие в разработке этой библиотеки.

Новым разработчикам, мы рекомендуем обратиться к Wiki разделу соответствующего репозитория: [https://github.com/fbezruchko/miro\\_library\\_arduino/wiki](https://github.com/fbezruchko/miro_library_arduino/wiki)

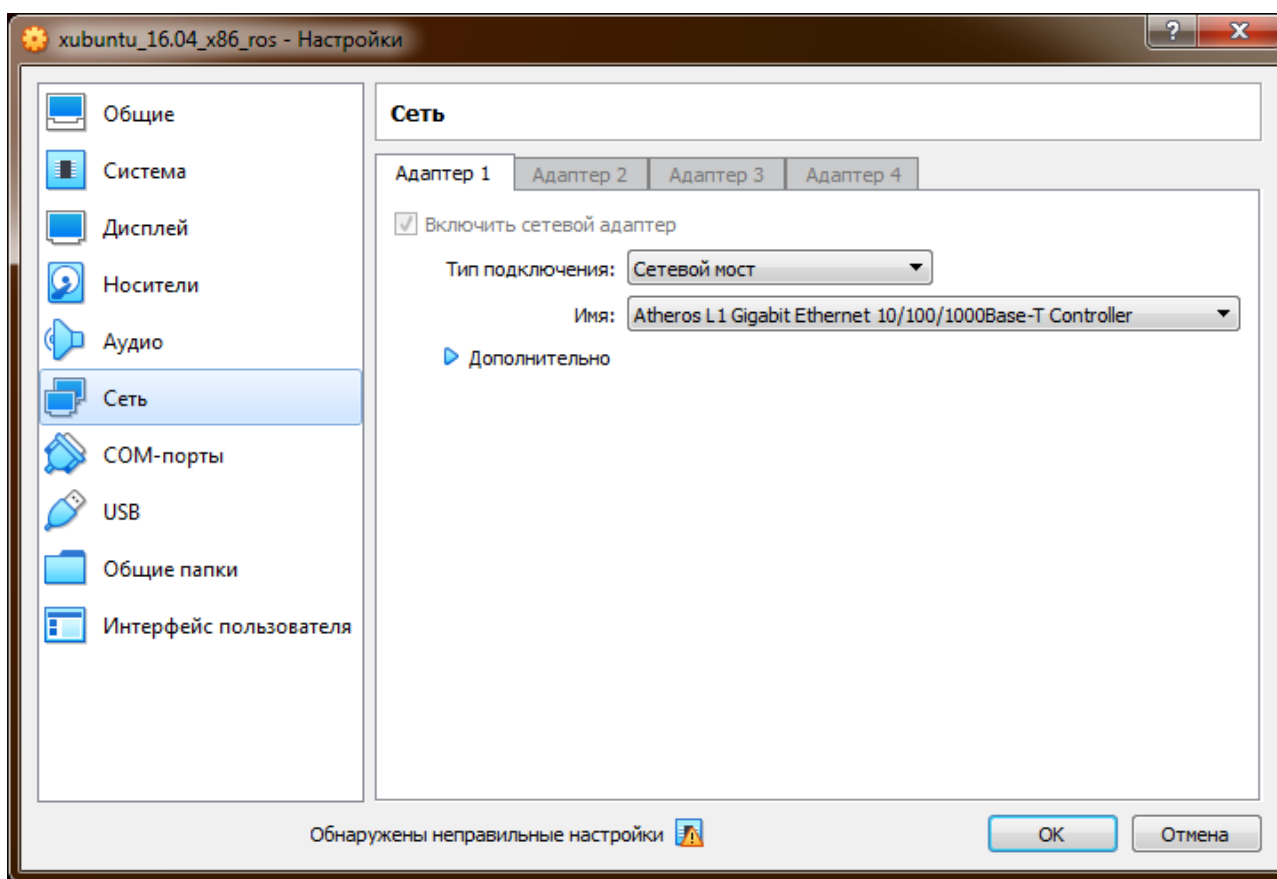
API позволяет сосредоточиться на стратегии поведения робота, не тратя время на реализацию каких-то базовых функций движения, обмена данными с датчиками и пр. По этой причине, разработка с использованием API не очень подходит для образовательных целей, если вы хотите разобраться как работают системы робота и до этого никогда не писали программы под Arduino. Для новичков мы рекомендуем начинать с изучения примеров библиотеки MIRO, не имеющих префикса «API\_» в своем названии.

## Сценарий 2. Разработка под Raspberry Pi.

Разработка решений под Raspberry Pi на базе робота MIRO – это уже несколько иной уровень. Благодаря производительному ARM-процессору Raspberry Pi, разработчик получает в свое распоряжение всю мощь библиотеки OpenCV и фреймворка Robotic Operation System (ROS).

1. В комплекте с роботом поставляется карта памяти формата SD (microSD) на 16 или 32 ГБ. Этого объема достаточно, чтобы уместить множество требуемых нам библиотек. Мы подготовили образы операционной системы Raspbian Lite с скомпилированными и предустановленными OpenCV и ROS. Скачайте образ подходящий для вашей платы, пройдя по ссылке в репозитории: [https://github.com/fbezruchko/miro\\_rpi\\_images](https://github.com/fbezruchko/miro_rpi_images)
2. Установите образ системы на карту памяти, согласно инструкции на официальном сайте: <https://www.raspberrypi.org/documentation/installation/installing-images/> Разница с инструкцией в том, что вам нет необходимости скачивать официальный образ – вместо него вы устанавливаете образ из репозитория MIRO.
3. Мы также подготовили для вас образы виртуальных машин, с предустановленными OS Linux, ROS и OpenCV. Скачайте бесплатную среду виртуализации VirtualBox, пройдя по ссылке: <https://www.virtualbox.org/wiki/Downloads>. Установите VirtualBox согласно инструкции: <https://www.virtualbox.org/manual/UserManual.html#intro-installing>
4. Скачайте образ виртуальной машины, пройдя по ссылке в репозитории: [https://github.com/fbezruchko/miro\\_virtualbox\\_images](https://github.com/fbezruchko/miro_virtualbox_images).
5. Распакуйте и установите образ виртуальной машины согласно инструкции: <https://www.virtualbox.org/manual/UserManual.html#ovf>

6. Настройте беспроводное сетевое подключение вашей Raspberry Pi согласно инструкции: <https://learn.adafruit.com/adafruits-raspberry-pi-lesson-3-network-setup/setting-up-wifi-with-occidentalis> (а именно раздел «Create the file in /boot»).
7. Если прежде вы отключали Raspberry Pi для экономии заряда – самое время подключить плату обратно к цепи питания робота.
8. Включите робота и подождите пару минут, чтобы Raspberry Pi успела загрузиться.
9. Узнайте IP-адрес, который назначил ваш роутер плате Raspberry Pi. В этом вам поможет инструкция: <https://www.raspberrypi.org/documentation/remote-access/ip-address.md>
10. Используя терминальную программу, подключитесь к плате по протоколу SSH. Как это сделать описано здесь: <https://www.raspberrypi.org/documentation/remote-access/ssh/> (начиная с раздела 4 «Set up your client» в зависимости от используемой вами операционной системы на хосте (ПК, с которого осуществляется подключение к Raspberry Pi).
11. В настройках виртуальной машины (в меню самого Virtual Box) в меню «Сеть» -> «Настроить сеть» удостоверьтесь, что в качестве типа подключения вашей виртуальной машины к сети выбран «Сетевой мост». В этом случае, ваша виртуальная машина окажется в той же подсети, что и Raspberry Pi робота Miro. Поменять тип подключения можно даже на запущенной виртуальной машине.



12. Запустите виртуальную машину и дождитесь полной загрузки, пока не увидите рабочий стол:



13. Нажмите комбинацию клавиш CTRL+ALT+T, чтобы запустить терминал.

14. В терминальной программе наберите и выполните команду:

```
$> ifconfig
```

Эта команда выведет параметры доступных сетевых соединений. В этом выводе нам необходимо узнать ip-адрес виртуальной машины. Например, в нашем случае виртуальная машина имеет адрес 192.168.0.15:



```

Terminal - user@ros-pc: ~
File Edit View Terminal Tabs Help
user@ros-pc:~$ ifconfig
enp0s3  Link encap:Ethernet  HWaddr 08:00:27:0b:69:c1
        inet addr:192.168.0.15  Bcast:192.168.0.255  Mask:255.255.255.0
        inet6 addr: fe80::3154:a40e:5ebf:5f66/64 Scope:Link
        inet6 addr: fe80::6972:cbe6:36c:5f49/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
        RX packets:234958 errors:0 dropped:0 overruns:0 frame:0
        TX packets:4253 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:330674423 (330.6 MB)  TX bytes:610568 (610.5 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1  Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING  MTU:65536  Metric:1
        RX packets:959 errors:0 dropped:0 overruns:0 frame:0
        TX packets:959 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:134599 (134.5 KB)  TX bytes:134599 (134.5 KB)

user@ros-pc:~$
    
```

15. Теперь на Raspberry Pi выполните команду:

```
$> raspivid -t 0 -cd MJPEG -w 640 -h 480 -fps 30 -b 16000000 -o - | gst-launch-1.0 fdsrc !
"image/jpeg,framerate=30/1" ! jpegparse ! rtjpegpay ! udpsink host=192.168.0.15 port=4000
```

Где в качестве ip-адреса укажите адрес вашей виртуальной машины.  
А на виртуальной машине выполните команду:

```
$> gst-launch-1.0 udpsrc port=4000 ! "application/x-rtp,media=(string)video,clock-rate=(int)90000,encoding-
name=(string)JPEG,a-framerate=(string)30.000000,a-framesize=(string)640-480,payload=(int)26" !
rtjpegdepay ! decodebin ! autovideosink
```

В результате должно открыться отдельное окно с выводом картинки с камеры робота.  
Картинка может быть перевернутой – это сейчас не важно.

Закрывать выполнение трансляции можно стандартным CTRL+C на каждой из сторон.

Мы только что проверили один из способов трансляции видео с камеры. Длинные команды, которые мы выполняли в пунктах 14 и 15 содержат в себе описание конвейера (pipeline, пайплайна) упаковки и с распаковки видеопотока и трансляции его по сети, выполняемого программой gstreamer. Gstreamer – очень мощный инструмент для работы с аудио-видео информацией, которому стоит обучиться, используя либо официальную документацию, либо множество статей в сети Internet.

В репозитории [https://github.com/fbezruchko/miro\\_rpi\\_images](https://github.com/fbezruchko/miro_rpi_images) есть файл [rpi camera use-cases](#), который содержит еще несколько pipeline, реализующих трансляцию видео с камеры робота. Каждый способ имеет свои плюсы и минусы (скорость, качество, степень и способ сжатия).



TO-DO: Тут про пример: получает видео с камеры робота, обрабатывает первоначально в raspberri, пересылает на хост, обрабатывает на хосте в опенсиви, пересылает обратно команды управления как роботом, так и настройками обработки.