

Мотивация

Современная цивилизация зиждется на информационных технологиях. Они прочно вошли в нашу повседневность, и уже сложно представить мир без них. Тем временем эти самые информационные технологии не являются теми четырьмя слонами, на которых, как нам известно, стоит мир: как минимум есть еще исполинская черепаха по имени электроника.

Без последней вы не смогли бы проверить электронную почту, используя свой телефон, потому что не было бы никакого телефона, а также Wi-Fi (тем более бесплатного), сотовой связи, и вообще — не было бы никакой электронной почты, потому что если бы не электроника, то не существовало бы серверов, на которых работают подобные сервисы.

Всё больше молодых людей устремляют свой взгляд в сторону создания приложений для мобильных платформ или создают собственные, безусловно, полезные сервисы, но вместе с тем совершенно забывают о такой замечательной и не менее интересной области, как электроника.

Вы всё еще спрашиваете, зачем заниматься электроникой? Ну может, хотя бы потому, что без неё была бы невозможна современная цивилизация!

В Интернете уже существует множество разнообразных курсов по программированию в разных форматах, от простого набора статей до видеолекций и вебинаров. Ввиду того, что для программирования требуется только компьютер, никаких сложностей с обучением и самообучением у людей не возникает. С электроникой и микроконтроллерами всё иначе — это занятие не только затратное по сравнению с информационными технологиями, но и к тому же имеет более высокий порог вхождения. Быть может, по этой причине курсов по микроконтроллерам и электронике (в целом) в сети меньше. Однако стоит заметить, что в Интернете уже имеется некоторый спектр ознакомительных статей, посвященных разработке электроники, и целые курсы на тему микроконтроллерной техники. Но курса, который бы охватывал все стороны создания цифрового устройства, пока нет.

Следующее, что необходимо обосновать, — это использование STM32 вместо Arduino. Платформа Arduino семимильными шагами вошла в среду электронщиков и значительно пополнила их ряды. Теперь люди, плохо знакомые с электроникой и программированием, в состоянии решать насущные проблемы быстро и легко. Сама платформа Arduino — это замечательный инструмент для прототипирования, проверки идей и обучения неофитов, однако «за всё нужно платить» ¹.

Ценой упрощения стало качество понимания самой сути. Готовые платы расширения (в транслитерации «шилды», от англ. shields) не дают понимания схемотехники и превращают создание прототипа в комбинаторику. В случае если будет необходимо использовать что-то более специфическое, скажем, датчик, под который не написана библиотека, неопытный пользователь столкнется, возможно, с непреодолимыми трудностями. Как же он поступит? Согласно инстинкту самосохранения, он полезет искать другой датчик, библиотека под который уже кем-то написана. В противном случае ему придется перейти на следующий уровень и стать настоящим разработчиком. Этим хочется лишь подчеркнуть, что нельзя всё свести к перекидыванию спичек из одного коробка в другой. Спички тоже кто-то должен уметь делать.

Конечно, большим вопросом является то, как именно нужно объяснять работу устройства под названием микроконтроллер. Можно не уходить вглубь и сказать, что вызов `digitalRead(pin)` вернет состояние на ножке, а можно рассказать о триггере внутри, регистре, схеме защиты, да и заодно показать в целом схему типичного порта ввода-вывода и обосновать, что режимов работы такого порта может быть больше, чем два (условно «на вход» и «на выход»).

Итак, почему этот курс не про Arduino? Во-первых, их и так много в сети. Во-вторых, копать надо глубже. Иногда это больно, иногда неприятно, но, как известно, «по pain, по gain». Вряд ли современный (на границе технологий) продукт можно выпустить на Arduino. Во всяком случае, пока iPhone делают не на данной платформе.

В курсе совершается уход от типичной отладочной платы, так как основной задачей является не что иное, как попытка продемонстрировать практически весь жизненный цикл цифрового устройства, в частности – часов. Почему часы? Это могло быть и другое устройство... Подробнее этот вопрос будет раскрыт в последней главе, а сейчас лишь обозначим следующее: на примере часов можно научиться программировать МК, заглянуть внутрь него и в итоге получить не просто плату с напаянными элементами, отслужившую свое и оставленную где-то на полке под слоем пыли (её целью было научить), а действительно законченное и полезное устройство, которое вы можете поставить к себе на тумбочку. Распечатайте на 3D-принтере корпус, и никто даже не поймет, что эти часы сделали вы, а не кто-то где-то на заводе в Поднебесной.

Конечно, в курсе процесс разработки показан не в том порядке, в каком он происходит в реальности, но это плата за то, чтобы сохранить у вас интерес к электронике. Путь тернист, и такой подход, как представляется, позволит провести вас мимо потенциальных «граблей» и первых неудач. Допустим, вы потратили не один день, чтобы составить схему устройства, затем еще пару дней, чтобы его изготовить. А потом оно не заработало. Наверняка ошибка была допущена не фатальная и её даже можно быстро исправить, но уже прошло много времени, а результат нулевой, и запал уже не тот. Чтобы избежать этого, в курсе предлагается начать с готового устройства. Это позволит вам четко осознать все особенности устройства и принципы его работы по ходу его создания. В самом начале вам предстоит спаять устройство, а затем, постепенно изучая работу с МК, оживлять его. Большинство из изученного вами так или иначе пригодится в следующих, собственных, проектах. В последней главе мы детально рассмотрим, как проектировалось данное устройство, и эти знания помогут вам разработать что-то свое!

В заключение хотелось бы отметить, что современный мир разговаривает на «латыни» под названием английский язык (увы, это не русский) — знание которого необходимо любому более или менее профессиональному электронщику. Разумеется, данный курс не замахивается на обучение вас английскому, однако в рамках него вам придется работать с англоязычной документацией. В этой связи все термины дублируются на английском, а в конце курса вы найдете словарь.

[Назад](#) | [Оглавление](#) | [Дальше](#)

1. Советую вам к прочтению заметку моего коллеги по цеху: «[Ардуино vs STM32](#)». Она написана в пассивно-агрессивном стиле, но вполне обоснована. 