

Встроенные системы

Содержание

1	Компьютеры везде	2
2	Киберфизические системы	4
3	Встроенные системы	7
4	Жизненный цикл встроенной системы	10
5	Распределенные встроенные и мобильные системы	12
6	Платформы встроенных систем	16
7	Встроенные системы в образовании и науке	18

1 Компьютеры везде

В первой серии лекции «Встроенные и мобильные компьютерные системы» мы расширим ваше представление о том, что такое компьютеры сегодня. Почему в современном городе, образно говоря, нельзя даже чихнуть без участия компьютера, а если и можно – то это определённо чья-то недоработка. Мы покажем истинное разнообразие компьютеров и компьютерных систем, от самых маленьких до самых крупным, от самых распространённых до действительно необходимых нам с вами.

Начнём с простого вопроса: *что такое компьютер?* Представляется ноутбук или монитор с мышкой и клавиатурой. У тех, кто обладает минимальной компьютерной грамотностью, также представится большой и пыльный ящик где-то поблизости. Начитанные вспомнят такие слова как сервера, что сегодня живут в облаках, мейнфреймы или даже ЭВМ. Наиболее сообразительные из вас просто скосят взгляд на лежащей на столе мобильный телефон.

При этом если немного присмотреться к содержимому карманов вы можете найти фитнес браслет (технические характеристики которого всего лет 10 назад казались фантастикой), связку ключей с «кнопкой» от домофона, банковские карты или проездной билет, в каждом из которых есть свой микропроцессор, который попадая в магнитное поле терминала получает достаточный объём энергии для работы что бы включиться, «поговорить» с терминалом и тем самым выполнить свою работу.

Да даже бумажные книги в своём большинстве, сегодня, оснащаются компьютерами, не говоря уже о музыкальных открытках. Или каким образом рамка на выходе из магазина узнаёт о попытке незаконного выноса товара?

Но всё это не мелочи, отсутствие которых конечно не радует, но и не является критическим. При этом, чтобы увидеть действительно важное не требуется серьёзных размышлений.

Почему вы можете позвонить по телефону и посмотреть видео с котиками? Потому что город, а в принципе и почти вся территория, где живут люди оборудована вышками сотовой связи со сложнейшим оборудованием, работающим 24 на 7 холодными зимними ночами под сугробом в минут 30 и жаркими летними днями под палящим солнцем в железных ящиках с температурой более 60 градусов с минимальным обслуживанием долгие годы. А сколько сегодня ведётся работ над тем, чтобы покрыть связью весь земной шар? Дать доступ online каждому уголку планеты? Воздушные шары, спутники, дрейфующие в небе беспилотники? А знаете, что провода с интернетом уже долгие годы протянуты дну мировых океанов?

А за счёт чего всё это работает, все эти вышки и телефоны? За счёт крупнейшей в мире единой энергетической системы, опутавшей всю Россию. Что

это значит? Что есть единый пульт управления энергетической системой и генерирующими мощностями, такими как атомные электростанции, станции, дамбы ГЭС, тепловые станции.

Осуществляется управление распределением нагрузки, резервирование источников энергии. Не стоит упоминать в этом контексте такие мелочи как учёт электричества в жилом секторе, на производствах или управление городским освещением.

Само собой, количество компьютерной техники и алгоритмов, заложенных в эту систему огромно, оно включает результаты работы тысяч коллективов в течение десятилетий, в том числе и полученные в прошлом веке и на ныне забытых технологиях. А представляете какие изменения в ней должны произойти сегодня, когда генерирующие мощности может установить каждый, поставив солнечную панель или ветреную электростанцию?

И снова можно задать аналогичный вопрос, а откуда это всё взялось? Тяжелая и лёгкая промышленность не мыслимы сегодня без автоматизации, роботов, станков с ЧПУ и, как следствие, компьютеров. Представьте себе цех площадью в пару футбольных полей забитый под завязку оборудованием, деталями, продукцией, людьми, а под 20 метровым потолком ездят мостовые краны с грузоподъемностью в 100 тон. Этот цех расположен в корпусе совместно с 2 примерно такими же цехами. А на заводе таких корпусов десятки. И весь это завод работает как единый механизм, пронизанный нервными окончаниями информационной системы, доходящей до каждого станка, каждой детали и каждого сотрудника.

Сейчас мы нанесли крупными небрежными мазками несколько инфраструктурных областей, а именно коммуникации, энергетику и промышленность, существование которых невозможно без компьютеров и вычислительной техники. В тоже время, невозможно и существование привычного для нас мира.

Другими примерами являются:

- медицина, немыслимая без рентгенографии, магнитно-резонансной томографии, анализаторов крови и много другого, активно развивающаяся в направлении автоматизированной диагностики и телемедицины;
- агропромышленность, с системами посадки, сбора, сортировки и хранения урожая;
- транспорт, от автомобилей до авиации и морского транспорта;
- робототехника, всевозможные дроны, самодвижущиеся повозки и разные игрушки.

Этот список можно продолжать практически бесконечно. Конечно, во всех упомянутых областях используются персональные компьютеры, сервера, мобильные телефоны. Куда без них? Но это, как правило, либо «пользовательские интерфейсы» (так называемые «автоматизированные рабочие места»), либо хранилища и системы обработки данных. Другими словами, как в том анекдоте, системы, которые просто подают ключи.

Большая же часть реальной работы производится в поле, на компьютерах, установленных на непосредственно упомянутых выше станках, генераторах, кранах, вышках посреди Сибири, фонарных столбах или в холодной темноте за орбитой Плутона под ионизирующим излучением Солнца. Ее выполняют тысячи и миллионы специализированных компьютеров и вычислительных систем, обладающих вычислительной мощностью от мейнфремов 60 годов (то есть очень маленькой) до сумасшедших гигафлопсов, причем с гарантированным временем реакции.

Эти специализированные компьютеры, называемые встроенными системами, объединены сетями, которые позволяют им работать сообща для достижения единой цели, решая задачи гигантской вычислительной сложности и распределенные по огромной территории.

Далее в этой лекции мы расскажем о том, что такое встроенные системы, почему их разработка это не программирование, как они помогают решать задачи в других областях науки и повседневной жизни, куда все это развивается и чего нам не хватает для того, что бы оказаться в мире Гибсона.

2 Киберфизические системы

Подавляющее количество упомянутых в предыдущей серии систем относятся к классу кибер-физических. Может показаться, что первые кибер-физические системы появились вместе с первыми компьютерами, то есть в прошлом веке (если мы говорим о ЭВМ) или даже в начале позапрошлого (если мы говорим о Жакардовом ткацком станке), так как именно тогда появились первые системы сочетающие в себе кибернетическую (то есть компьютерную) составляющую и физические компоненты в виде каких-то манипуляторов, двигателей или датчиков.

Но это понятие появилось относительно недавно, в 2006 году, и ознаменовало выделение примеров лучших информационно-управляющих систем в отдельную категорию с формированием нового киберфизического подхода в разработке.

Давайте попытаемся разобраться в чем заключается качественный отличия нового подхода и нового класса систем.

Грубо, информационно-управляющую систему можно представить как:

- некоторую установку (допустим – конвейер производственной линии);
- стойку с промышленными контроллерами и другим компьютерным оборудованием;
- а также толстый жгут проводов между ними.

Причём зачастую, установка разрабатывается или разрабатывалась независимо по отношению к системе управления, которая была добавлена уже постфактум, для автоматизации работы уже существующего оборудования. Зачастую, управлением этой установкой ранее занимались специально обученные люди, замена которых на машины лишь ускорила работу и снизила затраты, но никак принципиально не изменила принципы её функционирования. Мы всегда можем понять, что является в системе физическим поведением, что кибернетическим. Это крайне удобно с точки зрения разработки.

Но переход к киберфизическим системам обусловлен постоянным ростом требований к функциональности, удобству и энергоэффективности существующих систем, в результате которого пришлось принципиально пересмотреть весь подход к их проектированию. Если раньше можно было разрабатывать системы трём относительно независимым специалистам: специалисту по управлению, компьютерам и информационным коммуникациям, то сегодня такой фокус уже не проходит.

Мы подошли к границам применимости «типовых» решений.

1. Разрабатываемые алгоритмы управления не могут быть рассчитаны с необходимой скоростью и точностью по времени исполнения расчёта на современных микроконтроллерах, требуется создавать специализированные процессора и цифровые схемы.
2. Требуемая точность работы не может быть реализована в рамках идеализированных моделей и алгоритмов. Требуется непосредственно учитывать особенности реализации, задержки, помехи, износ и постоянное изменение параметров в физических установках.
3. Коммуникационные каналы не позволяют пропускать через себя весь необходимый поток данных без искажений и потерь.
4. Современные процессоры и вычислительные системы общего назначения потребляют слишком много энергии, сильно греются и требуют «тепличных» условий эксплуатации.
5. Современные технологии разработки программного обеспечения плохо приспособлены к созданию систем, работающих в реальном времени и крайне сложны в использовании для не тривиальных применений.

В совокупности это вынуждает строить тесно связанные между собой междисциплинарные коллективы, в рамках которых проектные решения принимаются согласовано между специалистами из разных областей. Ищутся компромиссные варианты.

Попробуем привести один очень простой, скорее даже игрушечный пример киберфизической системы. Все мы много раз видели в торговых центрах вращающиеся двери, обычно с тремя, а иногда и с четырьмя створками. Первые такие двери не имели никакой автоматизации и толкались человеком.

Потом началась автоматизация, в рамках которой дверцу оборудовали двигателем и датчиком приближения человека (оптическим или «кнопкой в полу»), а также системой безопасности, которая если что останавливает двери что бы ничего плохого не случилось (а ведь прецеденты были). Каждый из нас сталкивался с такими дверьми. Которые игнорируют тебя и вращаются, когда никого уже нет рядом. Которые останавливаются если их чуть-чуть задеть, а попытку её подтолкнуть вообще воспринимают как нападение. В принципе, такое поведение полностью соответствует классическому пониманию неудачной информационно-управляющей системы.

Современные автоматические двери работают иначе. Они не нуждаются ни в каких внешних датчиках и вероятность ложного срабатывания у них минимальна. Такие двери, как и в старь необходимо толкнуть рукой, после чего она начнёт медленно вращаться до того положения, когда вы сможете выйти с другой стороны. При этом, если вы попытаетесь разогнать дверь быстрее положенного – она не даст вам этого сделать, что бы вы не смогли случайно ударить другого человека. В основе такой киберфизической системы лежит математическая модель двери и её двигателя, параметры которой позволяют точно прогнозировать движение и управлять ею, а вычислительная система гарантирует точность работы управляющего алгоритма.

У такой двери будет намного меньше сбоев, ниже стоимость (так как не нужны лишние датчики) и гораздо более интуитивное поведение.

Другими классическими примерами кибер-физических систем являются самые разные роботы на равне с так называемыми smart grid-ами – гибкими и адаптивными системами управления электроэнергией в масштабах городов.

Заканчивая серию, хочется отметить, что раздел киберфизических систем является одним из ключевых в области технических наук. Связанные с ним задачи лежат в областях:

- алгоритмов робастного и адаптивного управления;
- машинного обучения, компьютерного зрения и нейронных сетей;
- информационной безопасности;

- проектирования вычислительных и программных систем, разработки информационных сетей;
- создания новых процессоров и оригинальных вычислительных архитектур;
- системы автоматизированного проектирования и методики разработки сложных систем, компьютерных систем;
- верификации и валидации существующих систем, направленных на доказательство правильности работы и вносимых в них изменений.

3 Встроенные системы

В этой серии мы расскажем о том классе «непривычных» компьютеров, которые используются для рассмотренных задач, и являются базисом киберфизических систем, а именно о встроенных компьютерных системах.

Покажем их отличия и специфику работы с реальным, а не цифровым миром, с установками и физическими объектами, а не окошками, кнопками и пользователями.

Для лучшего понимания масштаба рассматриваемого явления начнём с нескольких цифр. На приведенном графике можно видеть данные аналитиков о выручке на глобальном рынке встроенных систем. По данным на 2015 год, это не много не мало порядка 160 миллиардов долларов. К 2021 году ожидается рост почти на 40%, до 225 миллиардов долларов.

Более того, никаких видимых причин почему этот рост должен остановиться или замедлиться не наблюдается, а скорее наоборот.

Есть и более детальные данные, из которых видно в каких отраслях прирост наиболее велик, правда, только для европейского рынка. Видно, что Европа составляет примерно четверть от всего рынка, а быстрее всего растут области:

- автоматизации (в значительной степени это промышленность, индустрия 4.0, робототехника);
- медицина (в виде всевозможной медтехники, носимой электроники, оборудование для диагностики);
- потребительская электроника (игрушки, холодильники, стиральные машины, умные дома и прочее);
- военная и авиакосмическая техника.

Теперь, понимая области и масштаб использования встроенных компьютерных систем и рассмотрев множество примеров, определим, что это такое.

Как всегда, для любого сложного вопроса, существует множество трактовок и пониманий, но мы остановимся на следующем:

Встроенная система, это специализированная (заказная) вычислительная система, непосредственно взаимодействующие с объектом контроля или управления, и зачастую, объединенная с ним единой конструкцией.

О каких важных свойства оно говорит?

Во-первых, встроенная система является заказной, а не универсальной системой, а как известно, универсальные системы — это те системы, которые плохо подходят ко всем задачам. Самым ярким примером будет швейцарский нож на 43 инструмента, которым в принципе можно сделать все что угодно, но на практике это лучше не проверять.

В качестве более точного примера можно привести сравнение человека и насекомого, а именно уровня их адаптивности к окружающей среде и эффективности решения повседневных задач.

Человек рождается совершенно беспомощным существом, достигает своей зрелость в течение десятков лет после чего начинает медленно и неповоротливо решать свои, пусть и сложнейшие, задачи, адаптируясь к окружающему миру и изменяя окружающий мир под себя в течение всей жизни. Именно такая гибкость и адаптивность позволяет людям жить практически в любых условиях. Но у этого и высокая цена: необходимость длительного «выращивания потомства», высокая стоимость обучения индивида, огромное энергопотребление (львиная часть которого приходится на мозг и тёплую кровь), регулярные ошибки и отвратительная реакция.

И практически противоположная ситуация у насекомых, с очень быстрым взрослением, невероятно эффективным и быстрым решением своих задач (например, во время охоты такая скорость реакции в принципе не достижима для как для людей, так и для млекопитающих в целом). Это позволяет им выживать в очень жёстких и ограниченных условиях, в рамках которых любое отклонение, любая ошибка будет означать смерть. Но у этого есть обратная сторона медали — многие насекомые в принципе не смогут перестроиться вслед за окружающей средой.

Точно также обстоят дела и в области компьютерной техники: если мы хотим решать какие-то задачи с высокой эффективностью, будь-то долгая работа от батарейки, малый размер и вес, высокая надёжность и защищённостью или низкая себестоимость (как например в случае rfid меток используемых вместо штрихкодов), то мы вынуждены подчинить все аспекты системы той цели, ради которой она создаётся.

Мы вынуждены убирать всё лишнее, искать нестандартные решения, оригинальные подходы и подходить к решению задачи комплексно. Это отра-

жается на выбранных технологиях, программе и аппаратной составляющей.

Вторым важным свойством встроенной системы является то, что она непосредственно взаимодействует с объектом управления. У нас есть компьютерный процесс преобразования информации, определенный разработчиком, и физический процесс, связь между которыми обеспечивается исключительно датчиками и исполнительными устройствами.

При этом, важна не только получаемая с датчиков информация и посылаемые в ответ на нее команды, но и работа вычислительной системы в реальном, физическом, времени, с точным соблюдением временных задержек, когда работа должна быть выполнена точно в срок, ни быстрее, ни медленнее. Нарушен срок – значит работа провалена.

Причем важно понимать, что работа в реальном времени никак не говорит о скорости вычислительного процесса или встроенной системы, что отлично демонстрируют два примера:

- системы противовоздушной обороны, которые должны очень быстро обнаружить угрозу, спрогнозировать ее траекторию и перехватить ее;
- системы автоматического управления ГЭС, которые в случае превышения допустимого уровня воды должны через 5 минут начать открывать створки водосброса. Открытие длится порядка 7 минут, иначе может произойти гидроудар и разрушение плотины.

Третьим важным свойством встроенных систем является то, что они, как правило, объединены с объектом управления в единую конструкцию. В простом варианте это означает, что встроенный компьютер будет установлен непосредственно на объект управления со всеми вытекающими в виде:

- ограничений по размеру, выделению тепла и энергопотреблению;
- устойчивости к вибрациям, ударам, пыли, воде, теплу, соли;
- защищенности от электромагнитных помех, вызванных непосредственной близостью силовых линий, мощных электродвигателей
- и других схожих обстоятельств.

А в сложном случае, как например в любом современном автомобиле, бортовая встроенная компьютерная система развалится на сотни микроконтроллеров, объединенных между собой бортовой компьютерной сетью. Каждый из них решает свою задачу, обмениваясь информацией с «коллегами», в условиях регулярных сбоев, зависаний и искажений данных при передаче.

Некоторые специалисты из этой области в принципе не накладывают никаких ограничений на масштаб встроенных систем, выделяя их по принципам

проектирования. В таком случае сюда также включаются системы городской автоматики или системы геопозиционирования, такие как ГЛОНАС или GPS.

4 Жизненный цикл встроенной системы

Ранее, мы рассмотрели, что такое встроенные системы, а теперь давайте рассмотрим их жизненный цикл с начала и до самого конца. Как и для прочих систем он начинается с замысла, в рамках которого определяется решаемая проблема, находятся все значимые заинтересованные стороны и фиксируются предъявляемые ими требования к системе.

При этом спектр рассматриваемых требований выходит за пределы привычных для программных систем списков функций, требований к входным/выходным данным и пространных тезисов касательно usability и user experience. В него также включаются:

- требования по временным задержкам и приоритетам между задачами (на случай, если всю работу невозможно выполнить в срок);
- требования к условиям эксплуатации (температура, вибрации, туман, вода и многие другие);
- аппаратные интерфейсы ввода / вывода, источник питания, электрические помехи;
- а также самые разные требования по сертификации, начиная с электробезопасности, без которых система в принципе не сможет быть принята в эксплуатацию.

Затем начинается наиболее интересная стадия жизненного цикла системы: ее проектирование и разработка.

Она начинается с архитектурного проектирования, в рамках которого принимаются все наиболее важные проектные решения, будь-то выбор вычислительной платформы, централизованного или распределённого исполнения, операционной системы, языков и инструментов проектирования, декомпозиция системы на части, распределение целевой функциональности между разными уровнями организации вычислительной системы и узлами, технологии тестирования и многое другое. Тут же решается вопрос того, как организовать процесс разработки таким образом, чтобы получить требуемый результат в срок и с необходимым качеством, какую методологию выбрать: гибкую или не очень.

Затем выполняется непосредственно разработка, результатом которой являются:

- электрические схемы, 3д модели и документация для производства печатных плат и корпусов, сборки электроники;
- исходный код программного обеспечения, который будучи запущенным на аппаратуре позволит решить поставленные задачи;
- документация на продукт, повествующая о том, как она должна эксплуатироваться, обслуживаться и развиваться.

Вероятно, вы могли предположить, что разработка встроенной системы имеет строгую последовательность: сперва аппаратура, затем реализация ПО на ней. Ведь нельзя же строить дом одновременно с закладкой его фундамента? Но нет, технологии это уже давно-то позволяют. Называется это модульным строительством, когда здание собирается на месте из заранее изготовленных на заводе модулей и представляющих из себя готовые гостиные, кухни, санузлы и коридоры.

В вычислительной технике дела обстоят аналогично, параллельная разработка столь тесно связанных компонент получила название совместного проектирования или CoDesign, когда программы пишутся одновременно с разработкой аппаратуры, что достигается за счёт инструментов моделирования, унификации и стандартизации вычислительных платформ, автоматизации самого процесса проектирования и модель ориентированной инженерии, когда части системы динамически адаптируются под изменения друг друга.

Параллельная разработка позволяет не только значительно сократить время на создание системы, но и снижает риски, так как позволяет раньше проверить их совместимость и не препятствует «перекосам проекта» в одну из сторон. Когда основные решения идут от аппаратной и программной составляющей.

Другой важной составляющей разработки именно встроенных и мобильных систем является необходимость создания специальных инструментов под конкретный проект. Например, для быстрого тестирования аппаратуры после её производства, для отладки алгоритмов, так как зачастую лучшим средством отладки является светодиод, осциллограф и логический анализатор, а для мобильных систем в полный рост стоит проблема журналирования и мониторинга, так как зачастую нет возможности просто сохранить весь журнал событий.

После того как разработка закончена (и в большинстве случаев это не стоит воспринимать слишком буквально), начинается производство и испытание оборудования. Для этого закупаются комплектующие, запускается производственная линия, настраиваются станки, которые расставляют и паяют компоненты на платы. Осуществляют сборку конечной продукции.

Ключевое значение здесь играет тираж, который может варьироваться от единичного (автоматизация конкретной производственной линии), до

многомиллионного – на пример умные светодиодные лампы. Именно он определяет приемлемое соотношение затрат на разработку и производство.

Очень простой пример: вы можете поставить процессор за 1 доллар или за 2 доллара. При этом использование дешёвого процессора увеличит работу программиста в два раза. На фоне зарплаты программиста 10 долларов за мощные процессора будут не заметны, но если их нужно 100 тысяч?

Проведение испытаний готовой продукции тоже весьма длительный процесс, включающий в себя не только функциональное тестирование, но и длительные испытания на соответствие условиям окружающей среды. Плавные нагревания до 80 градусов (имитирующих железный ящик на летнем солнце) и охлаждения с выпадением росы до минус 40 (имитирующие условия крайнего севера). Есть, например, весьма забавное испытание на электробезопасность, когда проверяется невозможность касания открытых контактов пальцем.

Затем, уже испытанная продукция поступает к пользователю, где эксплуатируется, а для разработчика наступает длительная стадия поддержки, в рамках которой отлавливают оставшиеся ошибки. Хорошо если они программные и они могут быть исправлены при обновлении, но иногда они связаны с аппаратным устройством, что может привести к отзыву партии, как это было недавно с одной зажигательной моделью телефон.

Иногда, продукция возвращается в ремонт и встаёт проблема поиска причины, почему она вышла из строя. А если система существует и эксплуатируется уже не первый десяток, то встаёт проблема поиска комплектующих для её ремонта, так как часть элементной базы может быть снята с производства. Периодически можно видеть, как большие и уважаемые компании ищут по барахолкам типа ebay или avito старые процессора.

Окончание жизненного цикла встроенной системы может наступить спустя десятки лет после её разработки и ввода в эксплуатацию, ведь в реальных системах все стараются следовать принципу работает – не трогай. Оно заключается в выводе системы из эксплуатации или её модернизации, путём замены старого компьютерного мозга на новый. При этом крайне желательно не потерять весь накопленный опыт, чтобы при разработке на наступать повторно на те же грабли.

5 Распределенные встроенные и мобильные системы

Ранее, говоря о встроенных и киберфизических системах, мы говорили о них как о целостных объектах. Целостных с точки зрения их назначения, где каждая часть служит той цели, для которой система создавалась. При этом, саму встроенную систему довольно сложно назвать целостной, так как

она может состоять их множества маленьких компьютеров и иметь распределенную организацию.

В этой серии мы сосредоточим свое внимание на распределенных встроенных системах. Рассмотрим почему современные встроенные системы, как правило, имеют распределенную организацию, какие особенности возникают при разработке распределённых систем

Для начала, давайте представим относительно простую централизованную систему: а именно умную светодиодную лампочку, внутри которой помимо светодиодов и цоколя также располагается:

- контроллер управления, осуществляющий зажигание светодиодов с нужной яркостью и теплотой света в зависимости от времени суток;
- приемопередатчик беспроводной сети, например, wifi;
- микрофон для включения по хлопку.

Все перечисленные элементы располагаются непосредственно в корпусе лампы и могут успешно решать поставленный задачи.

Но стоит нам захотеть сделать ее немного умнее. Например, чтобы она загоралась автоматически, когда кто-нибудь входит в помещение или адаптировалась под текущее освещение.

Мы будем вынуждены сделать нашу систему распределенной, вынося датчики движения и освещенности за пределы корпуса лампы, а значит и строить компьютерную сеть, позволяющую им обмениваться данными.

При этом у нас есть множество вариантов того, как именно построить эту сеть:

- Мы можем воспользоваться wifi сетью, развернутой либо самой лампой, либо домашним роутером, и через эту сеть передавать данные.
- Казалось бы – идеальный вариант, так как wifi есть везде, но это автоматически означает высокое энергопотребление для датчиков, а значит либо пользователю придется регулярно менять батарейки, либо подключить их к сети 220 вольт, что в свою очередь требует размещения блоков питания и ведет росту размера и цены устройств.
- Мы можем использовать специализированные локальные беспроводные сети такие как zigbee или Z-wave, которые позволят установить датчики на батарейках с крайне низким энергопотреблением, но это привяжет нас к определенному типу беспроводной сети, а значит и к конкретным производителям оборудования, поддерживающим его.

- В случае если по тем или иным причинам мы не хотим использовать беспроводные каналы, мы всегда можем воспользоваться каким-нибудь проводным интерфейсом, для чего добавим в лампу специальные разъемы для подключения внешних датчиков. Питание можно будет передать по тому же интерфейсу, как это сегодня делается в случае с USB.
- Можно воспользоваться и слегка экзотическим вариантом, подключив все устройства к сети 220 В и передавать данные по силовой линии (так называемый power line communication).
- А можно и любым другим из доступных нам способов, например, беспроводными сетями для интернета вещей, такими как LoRaWan, или мобильными сетями.

Наверное, у вас возник вопрос зачем в 21 веке кому-то могут понадобиться проводные сети? И в случае домашнего использования вы скорее всего будете правы, так как количество датчиков и ламп будет весьма ограничено, эксплуатироваться они будут не долго, а если в каком-нибудь датчике сядет батарейка, то с высокой вероятностью его просто выкинут и поставят новый. Но в случае промышленных и индустриальных применений, когда системе для работы требуется большое количество энергии, что в любом случае требует ее подключения к проводным сетям; или в условиях сильных электро-магнитных помех, когда беспроводной интерфейс просто не сможет обеспечить необходимую скорость и стабильность передачи данных; или в случае если необходимо обеспечить высокую пропускную способность; или если система должна эксплуатироваться годами без обслуживания и в изменяющейся среде.

Зачастую у проводных интерфейсов на сегодня просто нет альтернатив. А теперь, давайте рассмотрим другой вариант развития исходной умной лампы. А именно – давайте представим, что нам нужно осветить не комнату, а, скажем, дворцовую площадь, зимний дворец и здание главного штаба. И не просто осветить, а так что бы мы могли установить в центре площади сцену и устроить там живой музыкальный концерт под открытым небом со светом-музыкой.

Очевидно, что такая система будет действительно распределенной и при ее построении мы столкнемся с принципиально новыми проблемами, как на пример:

- Управление нагрузкой, когда все используемые лампы должны будут одновременно загореться электрическая сеть может неправильно на это отреагировать.

- Передача большого объема данных на значительное расстояние, так как для формирования сложного и динамического узора необходимо управлять каждой лампой в отдельности или небольшими группами.
- Синхронизация работы ламп между собой, не смотря на то что они находятся на разном расстоянии друг от друга, а передача данных без искажений не гарантируется. Нам ведь необходимо, что бы все лампы на дворцовой площади могли вспыхнуть одновременно. Причем, в виду специфики живого концерта, мы не можем заранее предугадать, когда именно музыкант начнет исполнять свое соло.
- Более того, светомузыка не должна запаздывать относительно музыки обычной. Многие из нас слышали звук в наушниках от микрофона подключенного к компьютеру и знают, на сколько сильно такая задержка раздражает. Должно создаваться впечатление одновременности музыки и света.

Ну и конечно совершенно утилитарные вопросы:

- Сложности установки и настройки оборудования на месте, ведь мы не имеем права перепутать несколько светодиодных лент местами.
- Сложности поддержки системы в работоспособном состоянии, ведь чем больше ламп установлено под дождем и открытым небом, тем большее вероятность выхода чего-либо из строя. И хорошо если она просто погаснет, а если будет постоянно гореть красным?

Само собой, особенности проектирования распределенных встроенных систем не ограничиваются этим списком. Переходя в любую конкретную область, мы будем сталкиваться с новыми вопросами и проблемами, для части из которых сегодня уже есть готовые подходы и решения, а часть являются открытыми научными вопросами.

Сегодня, передовым краем развития распределенных встроенных систем являются такие направления как:

- Интернет вещей, в рамках которого ставится задача связать все бытовые устройства в единую информационную сеть, что позволит сделать окружающий нас мир по настоящему умным и удобным.
- Индустрия 4.0, в рамках которой ставится задача объединить в единое целое огромные производства таким образом, что все его процессы стали прозрачны и прогнозируемы, а управление им было сведено к простой панели, на которой пользователь может описать желаемый продукт и получить ее на выходе в кратчайший срок.

- Умный город, где вся городская инфраструктура сможет работать как единый слаженный механизм.

6 Платформы встроенных систем

Рассказывая о жизненном цикле ВcC и мобильных систем, мы упомянули о том, что в большинстве случаев в процессе разработки приходится разделить систему на программную и аппаратную составляющую.

При этом аппаратная составляющая выступает в качестве платформы или базиса, поверх которого программист может реализовать свою прикладную задачу.

Чем обусловлено разделение на программную и аппаратную составляющую? Перед тем как ответить на этот вопрос, давайте вспомним иностранную терминологию, так как в данном случае английский язык лучше передаёт качественную характеристику рассматриваемых понятий, в то время как русский вариант скорее акцентирует внимание на материальном носителе.

В английском для обозначения аппаратной составляющей используется термин Hardware, что можно дословно перевести как «трудное изделие». А программное обеспечение как Software – «мягкое изделие».

Как отмечают люди, стоявшие у истоков информатики и вычислительной техники, «трудность» hardware связана во многом с производственными особенностями, из-за которых изготовление требует много времени, а в его результат либо нельзя вносить изменения, либо их внесение крайне ограничено и затратно. Например – в печатной плате можно изменить несколько «дорожек», а в случае кремниевого чипа – скорее всего и это не будет возможным. Во многом поэтому аппаратная составляющая обычно значительно проще и регулярнее, по сравнению с программной.

А «мягкость» software обусловлена возможностью легко, быстро и в широких пределах его изменить, запустить, проверить, да еще и влезть в любую его часть в процессе работы для отладки. Именно по этому, «прикладная функциональность» реализуется на уровне ПО, включая все «особые случаи» и исключительные ситуации. Именно при помощи ПО »отрабатываются» новые пожелания пользователя и найденные ошибки в аппаратуре.

Сегодня, ничего принципиально не изменилось, за тем исключением, что сложность возросла многократно, а разработчики вооружились огромным количеством прекрасных инструментов. Можно утверждать, что для многих современных проектов hardware включает в себя помимо непосредственно аппаратуры также и операционную систему, и виртуальную машину, и какой-нибудь framework-и.

А software подразумевает наличие инструментария в виде всевозможных

компиляторов, трансляторов и сред разработки.

Если же мы уберем из получившейся картины конкретное прикладное ПО, оставив hardware и все инструментальные средства то мы получим то, что принято называть в вычислительной технике «платформой», а именно совокупность технических и инструментальных средств, на базе которых можно решать прикладные задачи.

В простейшем случае платформой является ПК (включая процессор и операционную систему), но, как мы говорили выше, обычно они плохо применимы в области встроенных и киберфизических систем.

Давайте рассмотрим основные варианты вычислительных платформ, для этой области. Первым вариантом является промышленный ПК, который по сути является тем же ПК, за исключением специализированного ПО, большего количества портов ввода/вывода и кондового исполнения позволяющего его поставить в пыльный не отапливаемый цех и выдерживать периодические удары киянкой.

Далее, мы будем спускаться в направлении снижения избыточности аппаратной составляющей и роста «специальной» производительности - производительности относительно конкретной задачи на ватт потребленной энергии.

В след за пром. ПК у нас будут идти промышленные логические контроллеры или ПЛК - специальные устройства предназначенные для автоматизации управления технологическими процессами. Они отличаются наличием специальных средств программирования для специалистов по автоматизации производств и, как правило, просто огромным количеством интерфейсов и портов ввода/вывода.

Далее идут микроконтроллеры – программируемые компьютеры для широкого класса алгоритмов со встроенной периферией, широко применяемые в области встроенных систем. Как правило отличаются низким энергопотреблением, компактностью и программируют с учетом непосредственного взаимодействия с «железом».

Потом идут сигнальные процессоры – уже специализированные программируемые процессоры ориентированные на конкретные типы алгоритмов, например – фильтрация цифровых сигналов с радио-приемника.

А лидируют в нашем списке полностью специализированные вычислители или СБИС, разработанные под конкретную задачу и практически полностью избавленные от какой бы то не было избыточности, которые могут быть быстро и относительно дешево реализованы при помощи программируемых логических интегральных схем (сокращенно ПЛИС), либо в случае огромной серии «выпечены» на кремниевом производстве.

В качестве точек над е отметим еще мобильные устройства (обычно это некоторый гибрид между ПК с точки зрения технологий программирования и

микроконтроллера с точки зрения аппаратной составляющей) и контрольные сети, на базе которых строятся распределенные встроенные системы.

Финальным штрихом данной серии станет вопрос о том, почему все не делают заказные СБИС, если они так хороши, как это показано на схеме? Дело в том, что с ростом специализированности вычислительной платформы многократно растет сложность и стоимость ее разработки, и одновременно с сокращением число использования.

7 Встроенные системы в образовании и науке

Ранее в данной лекции основной акцент делался на вопросах практического использования встроенных и киберфизических системы, их внутреннем устройстве и жизненном цикле. В этой серии давайте рассмотрим их с точки зрения использования для решения образовательных и научных задач.

Разработка сложной встроенной системы может включать в себя:

1. Проектирование, разводку и производство (в том числе и на кухне) печатных плат с цифровой, аналоговой и силовой частью.
2. Разработку цифровых схем и процессоров для реализации ускорителей, периферийных интерфейсов, обработки данных в реальном времени (например, для детектирования радиосигналов).
3. Разработку системного -программного обеспечения, непосредственно взаимодействующего с аппаратной составляющей, в том числе операционных систем и компиляторов.
4. Реализация сетевых протоколов для обеспечения связи ВСС с окружающим миром.
5. Разработка компьютерных сетей.
6. Разработка драйверов операционных систем (кто не хочет закомитить свой код в Linux?).
7. Серверное или прикладное программное обеспечения для сбора и обработки данных или пользовательского интерфейса.
8. Нейронные сети и машинное обучение для реализация автономных интеллектуальных систем, например, систем виртуальной или дополненной реальности, систем с речевым интерфейсом или распознаванием объектов

Вот что такое full stack developer. Это реальное понимание всех уровней функционирования вычислительной системы, от физики до «архитектурных абстракций и use case диаграмм». Но не стоит опасаться объема, обучение всему этому осуществляется постепенно и аккуратно и, само собой, не исключает специализации на каких-то вопросах, но понимание картины в целом даст вам значительную фору по сравнению с коллегами.

Еще интереснее дела обстоят с разработкой КФС, так как для них в проект придется подключать людей совершенно разных специальностей наравне с IT специалистами, а именно технологов, специалистов по алгоритмам управления, математиков, безопасников, конструкторов и многих других. В результате, опять же, участники таких проектов будут вынуждены в процессе обучения приобрести необходимые soft skills, которые научат вас общаться с людьми других специальностей, в речи которых с высокой вероятностью вы просто не поймете значительную часть слов.

А если у вас хватит таланта и креативности на организацию собственного проекта в этой области (что само собой будет поддержано со стороны Университета), вы приобретете бесценный опыт управления разношерстным коллективом и поиска компромиссов между абсолютно противоречивыми требованиями и столько «очевидными» требованиями, что профильным специалистам даже не придет в голову вам о них сообщить.

Использование встроенных и киберфизических систем для обучения не IT специальностям также широко применяется. На таких очевидных примерах как робототехника и промышленная автоматика даже не хочется отдельно останавливаться, так как ВcC являются фактически базисом для любых работ в этих областях, будь-то разработка учебной производственной линии или макет квадрокоптера, с использованием которого можно на практике проверят работоспособность разработанных математических моделей управления.

Огромное количество лабораторных установок для естественно-научных дисциплин вы будете использовать при выполнении лабораторных работ, будь-то работа с лазерами на физике, или исследование материалов на внешнее воздействие в рамках дисциплины сопромата. Если хотите посмотреть пример одной из самых дорогих и масштабных лабораторных установок, посмотрите информацию по проекту ИТЭР.

Если же вам понадобится поставить свой оригинальный эксперимент – вы всегда сможете это сделать при помощи как простых решений на базе ардуино или расбери пи, так и воспользовавшись профессиональным оборудованием и инструментарием, как например LabView или машинами реального времени от SpeedGoat.

Нельзя также не упомянуть огромную важность ВcC и КФС для не технических областей знаний, например, для социологии, экологии, биологии,

зоологии и так далее. ВсС системы позволяют собрать огромные объемы реальных данных о состоянии исследуемых объектов и протекающих в них процессах и окружающем нас мире, например:

- о миграции животных;
- об изменении климата;
- о поведении людей и их образе жизни.

Вся эта информация позволяет решать множество актуальных проблем и как следствие, каждый ученый и исследователь должен знать о таком инструменте как встроенные, мобильные и КФС, понимать границы применимости и уметь ставить соответствующие задачи.