**Маслов С.П., Рамиль Альварес Х., Сидоров С.А.**

###### Реализация МСО “Наставник” на микрокалькуляторе МК-85

Обосновывается перспективность развития системы обучения “Наставник” [1] с ориентацией на использование автономных, т.е. не нуждающихся в постоянной связи с центральным компьютером, терминалов, реализованных путем программной модификации существующих тиражируемых устройств. Обсуждается специфика и способы реализации терминала на таких устройствах. Приводятся краткие сведения об отечественном калькуляторе "Электроника МК-85" и его штатном ПО. Описываются процесс формирования модифицированного ПО и используемые для этого средства. Сообщаются данные о программе.

**Введение**

Версия “Наставника”, описание которой является предметом этой статьи, юбилейная - в 2002 году системе исполнилось 30 лет. Столь долгая жизнь и востребованность в условиях быстрой смены поколений аппаратуры и программного оснащения без всякой натяжки могут считаться феноменальными. В основе феномена лежит заложенная в "Наставник" и полностью оправдавшая себя на деле реалистическая концепция использования компьютеров для обучения, наличие учебных и методических материалов, оперативный перенос системы на разные компьютерные платформы и своевременное совершенствование ее аппаратуры. Вместе с тем, в эволюции "Наставника" имели место и негативные моменты - несмотря на самое благоприятное к ней отношение со стороны потенциальных потребителей, система не получила широкого распространения. Продолжительный опыт разработки и внедрения многочисленных версий "Наставника" дает основание назвать две главные причины этого: уникальность аппаратуры системы и использование проводов для связи управляющего компьютера с терминалами.

В большинстве версий системы [2] применялись специально разработанные (уникальные) узлы. Это было вызвано тем, что в ассортименте готовых изделий не удавалось найти ничего подходящего. Уникальные устройства (в первую очередь терминалы) проектировались предельно простыми и должны были стать недорогими в производстве. Однако найти желающих делать дешевые устройства удавалось с трудом. Даже в тех случаях, когда выпуск уникальных узлов был организован, существовала причина, затрудняющая установку системы для тех, кто смог их приобрести. Она заключалась в наличии проводов между компьютером и терминалами - разводки. Практика показала, что самостоятельно сделать разводку оказывалось не по силам большинству желающих иметь “Наставник”. Разводка - "ручная работа", а она простой и дешевой не бывает.

Выходом из положения был бы отказ от проводов вообще и организация дела таким образом, что деятельность обучаемого (хотя бы на протяжении занятия) осуществляется вообще без вмешательства центрального компьютера. Терминал при этом самостоятельно управляет обучаемым, а связь с компьютером (при необходимости) происходит вне занятия: до него (загрузка управляющей информации) и после (выгрузка протокола проведенного занятия) через стандартный связной порт.

Такой автономный терминал на нынешнем уровне цифровой техники вполне реализуем на микропроцессоре. Может показаться, что при этом будет устранена лишь одна из причин малой распространенности системы - отпадет необходимость в разводке, а сам терминал по-прежнему останется уникальным. Это не так. В последнее десятилетие появились и получили широкое распространение недорогие устройства типа электронных органайзеров, записных книжек, переводчиков, калькуляторов, телефонных аппаратов – "ширпотребе". Эти устройства выполнены на микропроцессорах и, как правило, могут быть модифицированы таким образом, что имеющийся ассортимент функций дополнится возможностью работы в качестве автономного терминала "Наставника". Модификация заключается в пополнении штатного ПО. Само устройство не подвергается переделкам и не лишается имеющихся возможностей. Такой терминал не будет уникальным и дорогим, поскольку речь не идет о создании и организации производства нового устройства, а лишь о модификации существующего. Успешный опыт реализации автономного (“карманного”) терминала на базе органайзера PSION XP-II [3] убедительно подтверждает это.

Отметим, что "Наставник" на автономных терминалах приобретает дополнительные полезные качества: возможность работы вне класса, удаленная загрузка курсов через Интернет и др. Кроме того, использование модифицированных устройств "автоматически" способствует популяризации системы. До сих пор желание иметь свой "Наставник" возникало, как правило, лишь после работы в нем. Тех, кому удалось это сделать, было немного. Теперь же, купивший модифицированный калькулятор (записную книжку, органайзер, телефон...), может заинтересовать его новыми качествами и пожелать их использовать. Для укрепления этого желания целесообразно иметь "зашитым" в устройство хорошо составленный курс, обучающий работе с ним самим.

Описываемый далее автономный терминал "Наставника", реализованный на калькуляторе "Электроника МК-85", представляет собой вторую успешную попытку развития системы в новом, перспективном направлении. Помимо того, что в основе терминала лежит отечественное изделие, реализация интересна тем, что выполнена существенно отличным от своего предшественника способом.

Специфика реализации терминала на "ширпотребе"

Устройства, пригодные для модификации под "Наставник" живут недолго, обычно не более 2-3 лет, а их следующие модели обычно настолько несовместимы с предшественниками, чтобы в них без изменений можно применять программы для текущей модели. Их разработчики не ставят перед собой такой цели, в отличие от разработчиков универсальных компьютеров.

Серийный производитель, как правило, работает независимо от разработчика и не имеет с ним прямого контакта. Документация, поставляемая с устройством, и та, что имеется у производителя, не содержит и не должна содержать сведений, необходимых для внесение изменений в устройство. Даже если удается наладить контакт с разработчиками устройства, это бывает мало продуктивным – они уже заняты другими делами и не хранят документацию по прошлым разработкам, т.к. сопровождать ширпотреб не имеет смысла – проще и дешевле выпустить на рынок новую модель.

Еще одна проблема – ограниченность ресурсов устройства, прежде всего памяти для программ и данных. Небольшая резерв памяти обычно имеется и нужно в него уместиться.

Возможны два подхода к реализации «Наставника» на серийно выпускаемом устройстве. Первый был опробован на органайзере PSION XP-II [3], программирование на котором было возможно на встроенном процедурном языке OPL. Выбор таких "открытых" устройств весьма ограничен. Другой подход – расширение функциональных возможностей устройства путем его доделки, вмешательства в «штатную» работу. Устройство при этом должно сохранить свою функциональность. Именно этот подход оказался приемлем для МК-85.

Суть его состоит в следующем. За основу был взят однотерминальный вариант «Наставника», написанный на языке диалоговой системы структурированного программирования ДССП [4]. ДССП представляет собой интерпретирующую систему со своим машинно-независимым языком программирования. Система построена методом «раскрутки»: небольшое машинно-зависимое ядро, содержащее набор базовых команд-примитивов и интерпретатор внутреннего представления программ – сшитого кода, реализует язык ДССП нижнего уровня. Этот язык - набор слов-операций, можно назвать языком ассемблера ДССП-машины. На этой основе определены слова более высокого уровня, и т.д. Вся остальная часть системы (средства разработки программ и средства поддержки диалога – интерпретатор входного языка ДССП, компилятор программ во внутреннее представление, отладчик, редактор текстов) написана на своем собственном языке.

В качестве первого шага из всей совокупности подсистем МСО «Наставник» была взята наиболее важная часть – подсистема обучения и использовано имеющееся ядро ДССП для микропроцессора с архитектурой PDP-11. Метод построения готовой системы ранее был неоднократно опробован при переносе самой ДССП на различные компьютеры [?].

На полномасштабном компьютере, где достаточно памяти, ДССП-программа обычно исполняется в среде системы разработки. В целевых системах обычно ограничиваются удалением крупных компонент – редактора текстов и отладчика. В нашем случае этого было явно недостаточно, поэтому было решено воспользоваться «целевой компиляцией», т.е. оставить в готовой системе только целевую программу и ядро ДССП, исключив оттуда интерпретатор входного потока, словарь, компилятор и прочие компоненты.

**Особенности реализации терминала на калькуляторе МК-85**

МК-85 представляет собой типичную микрокомпьтерную структуру, содержащую микропроцессор с системой команд PDP-11, ПЗУ 32KB, статическое ОЗУ емкостью до 6KB и контроллер матричного ЖК-дисплея. В микропроцессор встроен контроллер 54-клавишной клавиатуры и 15-разрядный параллельный I/O порт, подключиться к которому рядовому пользователю нельзя. В младшей половине ПЗУ (16KB) размещена программа, реализующая БЭЙСИК, со встроенными в нее драйверами клавиатуры и дисплея. Старшая половина ПЗУ свободна.

Наличие в МК-85 свободной программной памяти дало возможность разместить модифицирующую компоненту ПО в ней, а наличие БЕЙСИК-а - использовать имеющиеся драйверы клавиатуры и дисплея. Основное препятствие на этом пути состояло в том, что отсутствовала достоверная документация на ПО МК-85. Другое препятствие - отсутствие у МК-85 "связи с внешним миром". Для автономного терминала такая связь (в виде, например, COM-порта) желательна. В описываемой реализации достоверные коды получены чтением через программатор содержимого штатного ПЗУ, а COM-порт предполагается программно эмулировать на имеющемся параллельном порте МК-85 путем его доработки.

На первом этапе разработки ПО было необходимо разобраться в общей структуре считанного из ПЗУ МК-85 кода. После дизассемблирования всего кода программы и его изучения удалось найти процедуры чтения кода нажатой клавиши и вывода литеры в определенное знакоместо на индикаторе МК-85. Эти две процедуры обеспечивают всю потребность «Наставника».

Основные этапы сборки и отладки системы таковы: написанное на языке ассемблера машинно-зависимое ядро ДССП компилируется, получается код ядра. Для этого используется кросс-ассемблер для PDP-11, созданный специально для этих целей в ДССП [?].

Программа-компоновщик собирает подсистему обучения МСО «Наставник» из кода ядра ДССП и текста программы «Наставника», переводя его во внутреннее представление. Компоновщик также был создан специально для этих целей на основе имевшихся разработок штатного компоновщика ДССП и компоновщика, применявшегося для сборки ДССП, написанной на языке Си [6].

Простая программа-построитель образа ПЗУ формирует код прошивки для записи в ПЗУ калькулятора:

1. укладывает в этот образ исходный код МК-85, код «Наставника», управляющую информацию по обучающим курсам;
2. в свободную позицию таблицы переключения режимов на МК-85 прописывает ссылку на начало «Наставника»;
3. в свободную ячейку образа ПЗУ записывает корректирующую константу, чтобы сходилась контрольная сумма;
4. сохраняет полученный образ.

Для целей отладки из калькулятора была выведена колодка, в которую вставляется сменное ПЗУ. Таким образом можно былолегко опробовать различные варианты. Способ далеко не самый оперативный, но необходимый по крайней мере на этапе отладки ввода-вывода.

Окончательная отладка алгоритма проводилась как на самом МК-85, так и на программном эмуляторе PDP-11 [5]. Для предварительной отладки «Наставника» в ДССП применялся простой эмулятор «окружения», предоставляемого калькулятором.

**Оснащение МК-85 асинхронным COM-портом**

Наличие связи с внешним миром для автономного терминала "Наставника" обязательно. Конечно, можно изначально “зашить” в него некоторый набор курсов и этим ограничиться. Однако такая система лишится одного из основных достоинств - наличия обратной связи с автором курса. Многие готовые устройства-прототипы (такие, например, как телефонные аппараты) имеют встроенные связные возможности. Их и следует выбирать в качестве основы автономного терминала. В нашем случае (ориентация на отечественное изделие) выбирать было не из чего - МК-85 не имел ни штатного COM-порта, ни даже доступа к выводам имеющегося параллельного порта. Поэтому была проведена доработка параллельного порта путем преобразования TTL сигналов последнего в сигналы стандарта RS-232C, используемых в COM-портах PC. Программная эмуляция UART-а (асинхронного приемника/передатчика) обычно основывается на том, что стандартная длительность старт-стопных посылок формируются путем задания числа циклов в программе. В данном случае этот способ оказался непригодным, поскольку тактовая частота процессора в МК-85 не стабилизирована кварцевым резонатором. Поэтому длительность старт-стопных посылок имеет значительный разброс у разных экземпляров и нестабильна во времени даже у одного и того же. Для преодоления этого недостатка программа, эмулирующая UART, должна быть адаптивной - уметь автоматически подстраиваться под конкретное значения скорости процессора. Для этого в начале сеанса связи МК тестирует входную линию и ждет изменения ее состояния. РС посылает байт, состоящий из одних нулей. Настраивающая программа ждет возврата линии в 1 и считает циклы. Получившееся число делиться на 9. Т.о. определяется число циклов, соответствующее текущей длительность бита. Затем РС посылает байт, состоящий из чередующихся 1 и 0, а программа приема тестирует линию с интервалом в один бит. Приняв байт, его посылают назад в РС, формируя старт-стопную посылку программно, т.е. засылая 0 и 1 с требуемым интервалом в выделенный разряд параллельного порта. Если обнаруживается несовпадение, процесс повторяется с меньшей скоростью. Установление связи и настройка скорости обмена должны происходить в каждом сеансе связи, т.к. временные параметры могут меняться, например, с нагревом аппаратуры.

Объем программы, обеспечивающей такой обмен, менее 1 КБ.

**Выводы**

Получившийся вариант подсистемы обучения МСО «Наставник» оказывается вполне жизнеспособным, несмотря на весьма ограниченные ресурсы микрокалькулятора. После разработки программы обмена по последовательному порту подсистема обучения займет около 5 КБ, для управляющей информации по обучающим курсам останется 11 КБ, что достаточно для 3-5 курсов. Из имеющихся 6 КБ ОЗУ занято менее 0.5 КБ, остальную память можно использовать для загрузки управляющей информации из компьютера.

В то же время дальнейшее продвижение этой системы связано с проблемами, лежащими вне научных исследований. Поскольку необходима доработка серийного устройства, требуется участие разработчика и производителя.

Сегодня появился "ширпотреб" особого рода: мобильные телефоны, имеющие встроенный интерпретатор Java, сменные карты flash-памяти на десятки мегабайт, штатные средства связи с компьютером, в том числе беспроводные – инфракрасный порт. Стоимость их превышает стоимость калькуляторов, но через год-полтора они окажутся доступными массовому потребителю. Новая реализация МСО «Наставник» должна ориентироваться на них..

**Литература**

1. Брусенцов Н.П., Маслов С.П., Рамиль Альварес Х. Микрокомпьютерная система обучения "Наставник" - М.: Наука, 1990. - 223с.

2. Брусенцов Н.П., Маслов С.П., Рамиль Альварес Х. Методическое пособие по разработке учебных материалов в микрокомпьютерной системе обучения "Наставник" - М.: Изд-во Моск.ун-та, 1992. - 95с.

3. Маслов С.П. Карманная АСО "Наставник" (стартовая версия). //Программные системы и инструменты. Тематический сборник №2, М.: Издательский отдел факультета ВМиК МГУ, 2001.

4. Брусенцов Н.П., Захаров В.Б., Руднев И.А., Сидоров С.А., Чанышев Н.А. Развиваемый адаптивный язык РАЯ диалоговой системы программирования ДССП. М, Изд-во МГУ, 1987.

5. ERSATZ-11 PDP-11 Emulator, 2000. www.dbit.com

6. Бурцев А.А., Франтов Д.В., Шумаков М.Н. Разработка интерпретатора сшитого кода на языке Си. //Вопросы кибернетики. Средства разработки внутреннего программного обеспечения. Микроэлектроника. Перепрограммируемые системы. М., 1999. с.64-76.