

Модуль EHS6 и макетная плата Cinterion Concept Board.

Пример применения Java в модулях Cinterion

В статье представлен опыт начала работы с макетной платой Cinterion Concept Board Gemalto, на которой установлен однокорпусный модуль (System In a Package, SIP) Cinterion EHS6.

Сергей Дронский

Общее описание однокорпусной системы EHS6

EHS6 — модуль пятого поколения, предназначенный для построения интерфейсов типа M2M (Machine to Machine). В его основе лежит архитектура ARM11, на базе которой работает исполнительный модуль Java ME 3.2, оптимизированный для M2M-приложений, ограниченных в ресурсе. Наличие большого количества готовых программных модулей позволяет сократить время разработки и уменьшить стоимость готовых изделий. В компьютере Raspberry Pi также используется процессор на базе архитектуры ARM11.

По сути, EHS6 представляет собой сотовый модем с Java-машиной. Управлять системой можно как снаружи, AT-командами (например, от внешнего процессора), так и Java-программой (изнутри). Модуль содержит сотовый интерфейс (пятидиапазонный 3G HSPA, четырехдиапазонный GPRS/EDGE Class12). На макетной плате размещены встроенная антенна сотовой связи и держатель SIM-карты стандартного размера с функцией определения ее наличия. Скорость передачи данных «вниз» достигает 7,2 Мбит/с, а «вверх» — 5,7 Мбит/с. Модуль поддерживает многопоточное и многопротокольное выполнение Java midlet (Multi MIDlet Java execution).

Программе доступны 6 Мбайт RAM и 10 Мбайт flash-памяти.

Технические характеристики EHS6 приведены в табл. 1. Модуль заключен в корпус LGA, его мощность потребления и тепловыделение оптимизированы для промышленного применения в тяжелых условиях.

Интерфейсы и входы/выводы

Шестнадцать выводов общего назначения (GPIO), доступных из Java-мидлета, могут быть использованы для нескольких интерфейсов (сигналы DSR, DTR, DCD интерфейса ASC0; RXD, TXD, RTS, CTS интерфейса ASC1; SPI; Fast-Shutdown; Network-Status-Indication; PWM; Pulse-Counter lines).

В табл. 2 приведены варианты использования GPIO для формирования разных интерфейсов. Если вывод применяется для работы в составе какого-либо интерфейса, то он уже не может служить как вывод общего назначения. Например, если в приложении используется SPI, то выводы 3, 16, 17, 19 станут недоступными в качестве портов ввода/вывода общего назначения.

Цифровой аудиоинтерфейс, который можно использовать для передачи аудиоданных по каналу связи, требует применения внешнего соответствующего кодера/декодера.

Таблица 1. Технические характеристики модуля EHS6

Частотный диапазон, МГц	800/850/900/1900/2100
Интерфейсы	2 UART, USB 2.0, SPI, I ² C, 16xGPIO, SIM-интерфейс 1.8/3.3 В, цифровой звук
Специальные функции	Встроенный TCP/IP-стек Java ME Embedded 3.2 Мониторинг сетей без SIM-карты Jamming Detection (обнаружение помех) FOTA (обновление прошивки модема и ПО пользователя «по воздуху»)
Габариты, мм	27,6×25,4×2,2
Напряжение питания, В	3,3–4,5
Диапазон рабочих температур, °С	–30...+85

Таблица 2. Варианты использования выводов GPIO для организации различных интерфейсов

Линии ввода/вывода	Сигнал выключения	Светодиод режима	Выход ШИМ	Счетчик импульсов	Порт сериал0	Порт сериал1	Порт SPI	Межмодульный высокоскоростной интерфейс	Цифровой аудиоинтерфейс
GPIO1					DTR0				
GPIO2					DCD0				
GPIO3					DSR0		SPI_CLK		
GPIO4	FST_SHDN								
GPIO5		Status_LED	PWM2						
GPIO6			PWM1	Counter					
GPIO7									
GPIO8									
GPIO9									
GPIO10									
GPIO11									
GPIO12									
GPIO13									
GPIO14									
GPIO15									
GPIO16						RXD1	MOSI	AP_WAKEUP	
GPIO17						TXD1	MISO	HOST_ACTIVE	
GPIO18						RTS1		CP_WAKEUP	
GPIO19						CTS1	SPI_CS	SUSPEND	
GPIO20									TXDDAI
GPIO21									RXDDAI
GPIO22									TFSDAI
GPIO23									SCLK
GPIO24					RING0				

Температурный контроль

В модуле имеются встроенные механизмы управления температурой (Advanced temperature management). При выходе температуры за допустимые пределы ($-40...+90$ °C) он выключится с выдачей соответствующего сообщения. Температура модуля, которая контролируется NTC-резистором, установленным на плате, может не быть равна внешней температуре. Если выдача предупреждающих сообщений разрешена, то при возникновении угрожающего состояния такое сообщение будет выдано. Угрожающее состояние — подход близко к перегреву или переохлаждению. В этом состоянии модуль продолжит работать. Проектировщик системы может предусмотреть обработку таких сообщений и реакцию на них, например включить подогрев или охлаждающий вентилятор (либо холодильник на элементах Пельтье), для того чтобы температура системы вернулась в норму. Если же температура модуля выйдет за пределы безопасной эксплуатации, то модуль выдаст тревожное сообщение и выключится. Причем нельзя запретить выдачу тревожного сообщения о перегреве или переохлаждении.

Определение подавления радиосигнала

В модуле предусмотрен механизм определения подавления радиосигнала (RLS monitoring/jamming detection). Если модуль используется в составе охранной системы и сигнал сотовой связи подавляется внешними устройствами, то модуль определит факт такого подавления. Работающая программа отреагирует на недоступность канала связи и изменит ход выполнения в соответствии с логикой, заложенной программистом.

Обновление программного обеспечения

Механизм загрузки обновленных программ через сотовый интерфейс (FOTA) можно

настраивать под собственные потребности. Он бесплатен, так как представляет собой Java-мидлет с открытым кодом. Возможности FOTA в EHS6 очень широки: допустимо обновлять не только ПО модуля, но и Java-мидлеты пользователя; ПО разрешается загружать с обычного HTTP/FTP-сервера; команда на обновление может быть как локальной, так и удаленной (через IP-соединение или SMS). Наличие такого механизма предоставляет широчайшие возможности для разработчика по удаленному управлению модулем и системой на базе такого модуля. Можно корректировать ошибки и загружать исправленные версии, добавлять новые

возможности в программу и т. д., а также, что особенно приятно, автоматизированно загружать исправленное ПО без вмешательства конечного пользователя.

Макетная плата Cinterion Concept board

На макетной плате Cinterion Concept Board (рис. 1) смонтированы модуль EHS6 и вспомогательная электроника. Ее структурная схема показана на рис. 2.

Разработчики макетной платы предусмотрели ее физическую совместимость с интерфейсом популярного семейства «Ардуино»: физические размеры разъема, электрические параметры

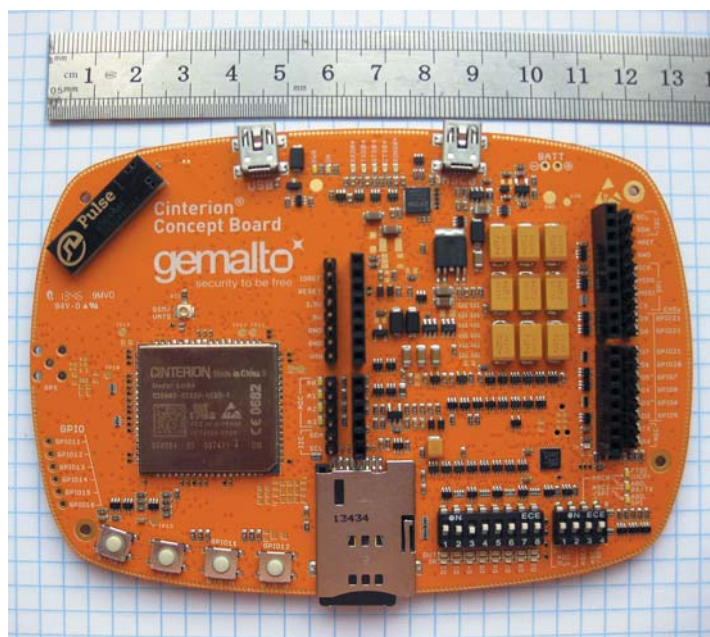


Рис. 1. Макетная плата Cinterion Concept Board Gemalto

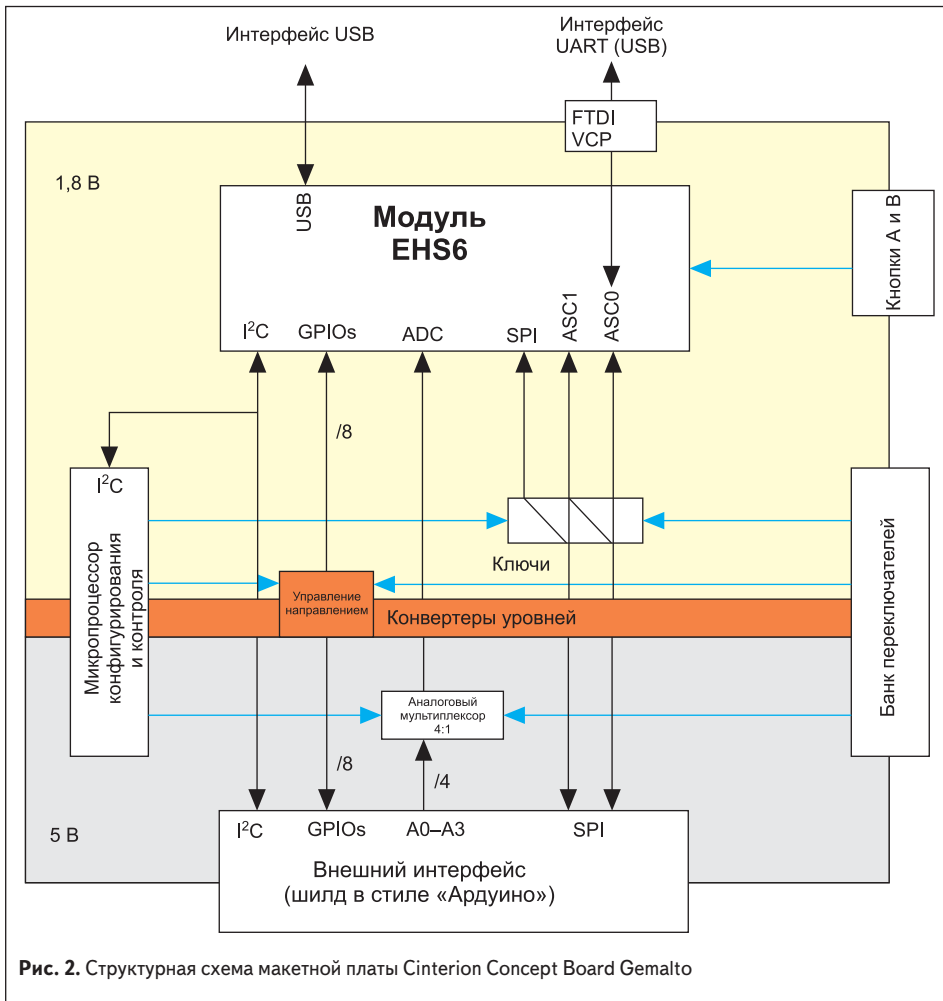


Рис. 2. Структурная схема макетной платы Cinterion Concept Board Gemalto

и набор сигналов обеспечивают прямое подключение платы «Ардуино» и/или разнообразных датчиков из бесчисленного множества предназначенных для нее устройств.

На плате установлен вспомогательный микропроцессор типа STM8L151GX (рис. 3). Разработчики назвали его CCU — configuration and control unit, что можно перевести как «устройство конфигурирования и контроля». Микропроцессор CCU позволяет программно управлять направлением каналов ввода/вывода и выбрать один из четырех каналов

аналогово-цифрового преобразователя, а также asc1 или канал SPI.

Выбор делается либо переключателем на плате, либо программно. Может возникнуть ситуация, когда переключатель и CCU дают разные команды. Управление от CCU всегда имеет более высокий приоритет над микропереключателями. Состояние вывода (вход или выход) всегда правильно отображается светодиодом, находящимся на плате, независимо от положения соответствующего переключателя.

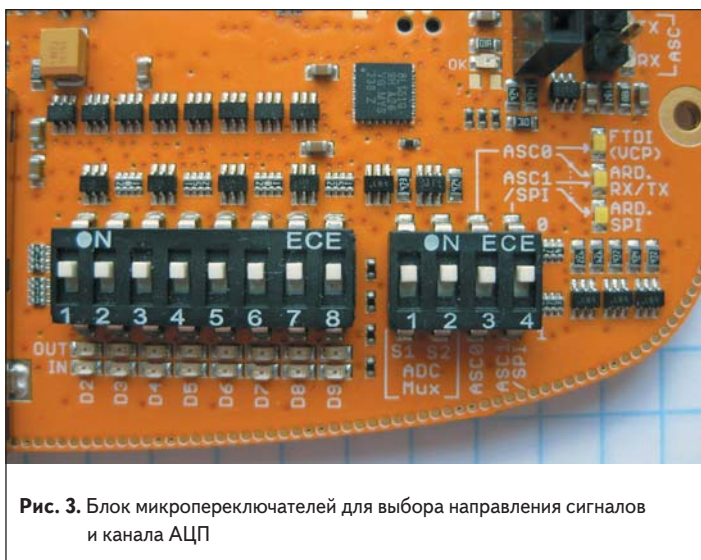


Рис. 3. Банк микропереключателей для выбора направления сигналов и канала АЦП

Упрощенная схема коммутации представлена на рис. 4. Рассмотрим, как она работает.

Режим ручного управления

Выход микросхемы IC1 находится в третьем состоянии и не влияет на потенциал в точке соединения резисторов R1 и R2. Потенциал этой точки определяется состоянием контактов микропереключателя S1. По умолчанию (в режиме заводской поставки) контакты микропереключателя S1 замкнуты, что дает высокий потенциал на входе микросхемы IC2, и схема преобразования направления и уровней находится в режиме «выход», обеспечивая на GPIO сигналы в диапазоне 0...+5 В.

При разомкнутом переключателе схема согласования уровней переходит в режим входа, обеспечивая трансляцию уровней 0...+5 В в уровни 0–1,8 В.

Режим программного управления

Для перехода в режим программного управления выход микросхемы IC1 надо переключить из режима третьего состояния в режим активного выхода. Поскольку выходной импеданс микросхемы пренебрежимо мал по сравнению с импедансом резисторного делителя (разница примерно на три порядка), то уровень сигнала на выходе системы управления будет целиком определяться состоянием выхода микросхемы IC1. А состоянии контактов ключа S1 не будет влиять на выходной сигнал.

Транзистор T4 показан, чтобы напомнить о возможности перевести вывод микросхемы IC1 в режим ввода и программно опросить состояние микропереключателя S1. Модуль CCU подключен к модулю EHS6 по каналу I2C. Адрес CCU — 0x69. Линии интерфейса I2C (кроме CCU), также выведенные на разъем, позволяют подключаться внешним I2C-контроллерам. Линии I2C также выведены на внешний разъем через двунаправленные преобразователи уровней 1,8–5,0 В на основе полевых транзисторов.

Модуль CCU может управлять направлением передачи информации в портах GPIO0GPIO7 выбором одного из четырех

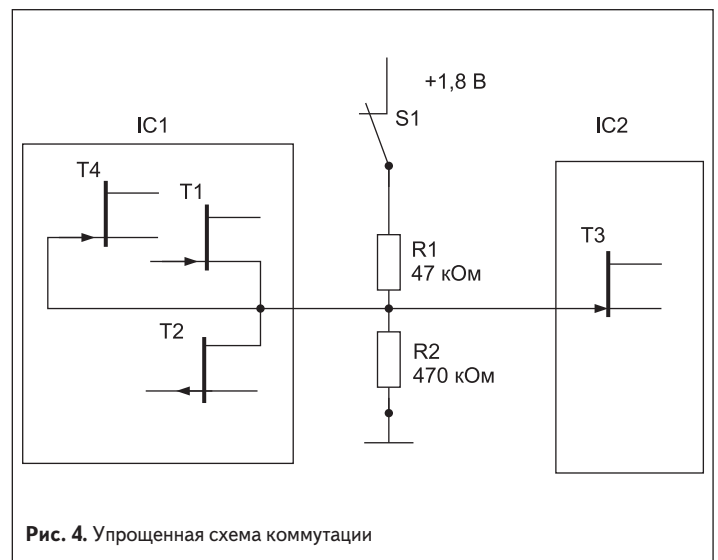


Рис. 4. Упрощенная схема коммутации

каналов АЦП, подключением порта ASC0 к разъему «Ардуино» или к преобразователю FT232 и подключением порта ASC1 либо к выводам сериал-порта на разъеме «Ардуино», либо к выводам SPI.

На макетной плате размещен коммутатор аналогового сигнала 0–5 В в один канал 0–1,2 В. Выбор канала осуществляется либо блоком микропереключателей, либо программно. ADC в EHS6 может преобразовывать напряжение максимум в 1,2 В. Для расширения входного диапазона до 5 В применен преобразователь на операционном усилителе. Для корректной обработки результатов следует использовать коэффициент 4,17. Например, если в Java-программе считан результат 1 В, то его следует умножить на 4,17, и уже этот полученный результат как раз и будет истинным входным напряжением.

Сериал-порт ASC0 модуля EHS6 может быть подключен системой управления (CCU + микропереключатели) либо напрямую к выводам разъема «Ардуино», либо к микросхеме трансляции сериал-порта в USB FTDI FT232RQ. Интерфейс ASC0 содержит девять сигналов для управления потоком данных. Можно использовать подключение через FT232 по USB с драйвером виртуального сериал-порта. Питание модуля также возможно через второй мини-USB разъем.

Выводы GPIO3 и GPIO16–19 модуля EHS6 могут быть использованы как сериал-порт ASC1 или как SPI. В обоих случаях автоматически обеспечивается нужное направление преобразования сигналов 1,8–5,0 В для согласования электрических параметров интерфейса. Чтобы включить интерфейс ASC1 или SPI, нужно в программе выполнить две операции: дать соответствующую команду CCU и создать SPI-соединение и потоки данных. Интерфейс ASC1 содержит четыре сигнала: RX, TX, RTS, CTS.

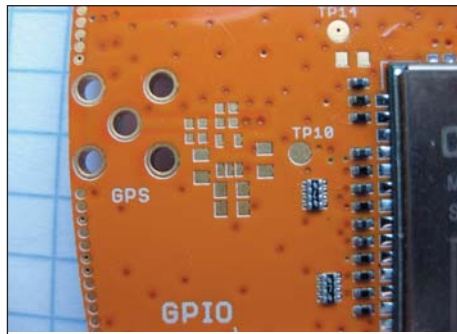


Рис. 5. Места под установку вспомогательных элементов GNSS/GPS

На макетной плате предусмотрены выводы для подключения внешнего литиевого аккумулятора. Зарядка батареи происходит током около 250 мА. При выборе аккумулятора следует иметь в виду, что когда модуль работает на передачу, он потребляет значительный ток (примерно 2 А), и нельзя допустить просадку напряжения ниже 3,3 В, так как при этом модуль отключится. Изготовитель рекомендует аккумулятор емкостью не ниже 800 мА·ч.

В модуле EHS6 имеются часы реального времени, но для их работы нужно, чтобы к макетной плате был постоянно подключен аккумулятор.

Ни EHS6, ни макетная плата не поддерживают прерываний. Единственный способ реакции на изменение входных сигналов — опрос.

На макетной плате также предусмотрены места под установку разъема и вспомогательных радиоэлементов, помеченных GPS (рис. 5). В производственной гамме Gemalto имеются модули с поддержкой ГЛОНАСС и даже ЭРА-ГЛОНАСС, например модуль



Рис. 6. U-FI разъем для внешней GSM-антенны

EHS5, полностью совместимый по выводам с описываемым EHS6.

Также на плате находится U-FI разъем для подключения внешней сотовой антенны (рис. 6). По спецификации модуль EHS6 полностью защищен от несогласованной антенны и способен выдержать ее без повреждений. По спецификации модуль может отдавать в нагрузку до 2 Вт, и если она предельно не согласована, то вся эта мощность будет рассеяна внутри него. Модуль оснащен встроенными средствами контроля температуры и при перегреве должен отключиться. Однако всегда следует помнить один из законов Мерфи: «Транзистор, защищенный предохранителем, вылетит первым, защитив предохранитель!» — и быть аккуратным с внешней антенной. Да и лишнее СВЧ-излучение организму тоже бесполезно.

На макетной плате отсутствуют источники питания для поддержки часов реального времени. Чтобы получить такую поддержку, надо припаять литиевую батарею к выводам BATT.

На макетной плате установлены четыре кнопки управления (рис. 8).

Кнопка Start служит для подачи сигнала Ignition для EHS6, кнопка OFF выключает модуль. Еще две кнопки BTN-A и BTN-B, подключенные к выводам GPIO11 и GPIO12 соответственно, могут произвольным образом использоваться в программе пользователя.

Установка программного обеспечения

Для начала следует зарегистрироваться на <http://m2m.gemalto.com/> и указать IMEI-номер системы, который находится на модуле EHS6

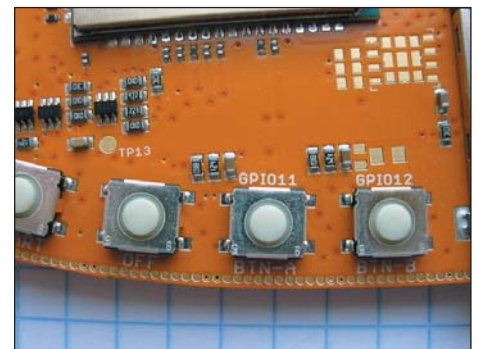


Рис. 8. Кнопки управления платой

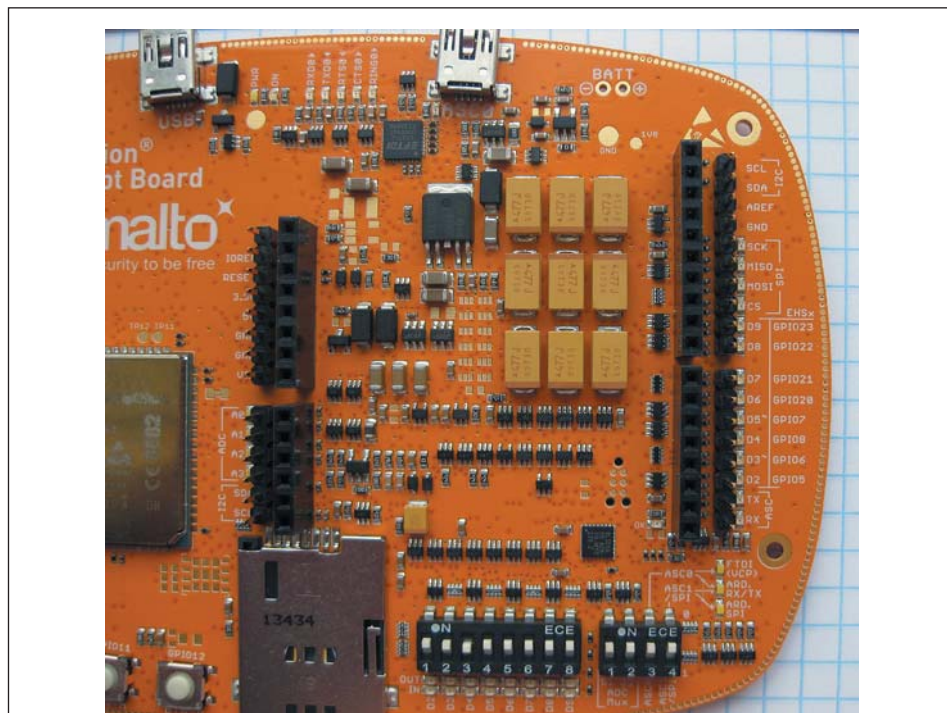


Рис. 7. Гребенка контактов в стиле «Ардуино» (для прямого подключения шилдов)



Рис. 9. IMEI-номер системы

в нижней строке (рис. 9). В данном случае это будет 358884 05 007431 4.

Через некоторое время (в моем случае ожидание заняло около 2 ч) приходит письмо с логином и паролем. С их помощью следует осуществить вход в систему и зайти в Cinterion Concept Board Support. Для установки ПО требуются файлы, названия которых показаны на рис. 10.

Теперь надо распаковать Installation Package и произвести установку путем запуска Setup. Я поставил систему на виртуальную машину под управлением 32-бит Windows 7, но, по информации специалистов Gemalto, можно использовать и 64-бит систему.

В процессе установки надо будет подключить плату и нажать кнопку «Старт». Возможно, драйверы платы устанавливаются автоматически, но если потребуются их указать, то они находятся в файле eh6_drivers_1112.

Установщик нигде не размещает ссылку на eclipse.exe, так что это нужно сделать вручную. Выйти к программе можно так: C:\Users\Public\Eclipse\eclipse.exe

В результате в системе появятся новые устройства (рис. 11).

Для общения с устройством по сериал-интерфейсу можно воспользоваться какой-либо доступной программой, например Putty (рис. 12).

Первая команда — запрос идентификации. Вторая — запрос напряжения питания, в дан-

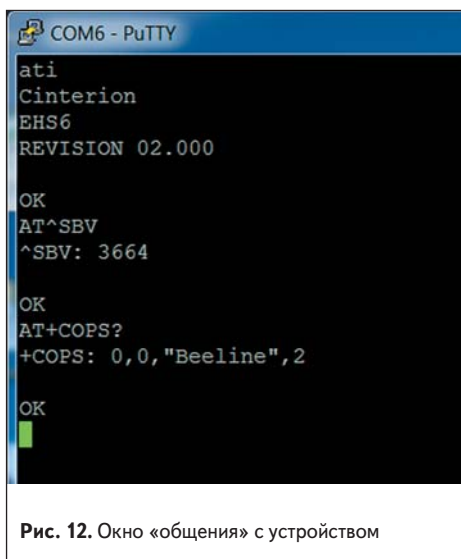


Рис. 12. Окно «общения» с устройством

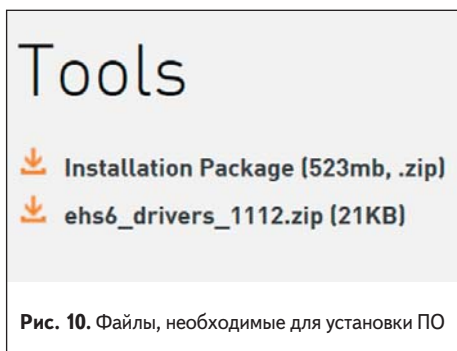


Рис. 10. Файлы, необходимые для установки ПО

ном случае это будет 3,664 В. Третья — запрос регистрации в сотовой сети. В данном случае модуль зарегистрирован в «Билайне».

После успешной установки Eclipse можно импортировать в Workspace демонстрационные программы и исследовать возможности Java-мидлетов. Демонстрационных программ вполне достаточно, для того чтобы показать работу системы и взаимодействие приложения с аппаратурой на плате. Есть традиционный Hello World и не менее традиционное моргание светодиода на плате. Импорт готовых проектов осуществляется так, как показано на рис. 13.

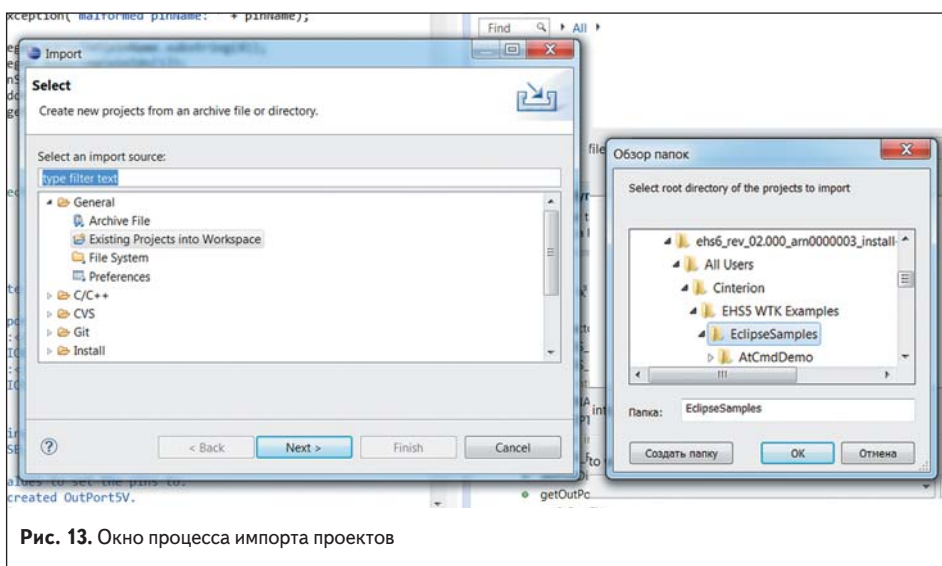


Рис. 13. Окно процесса импорта проектов

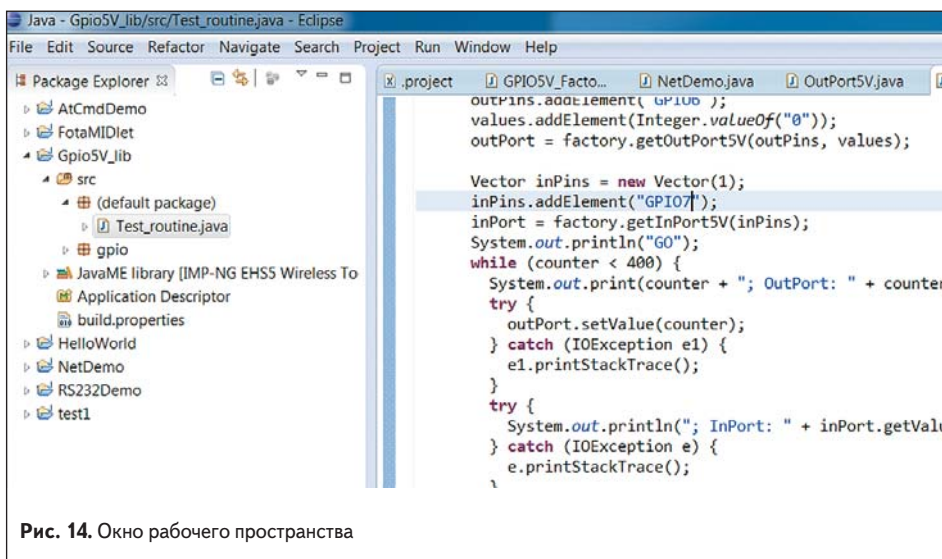


Рис. 14. Окно рабочего пространства

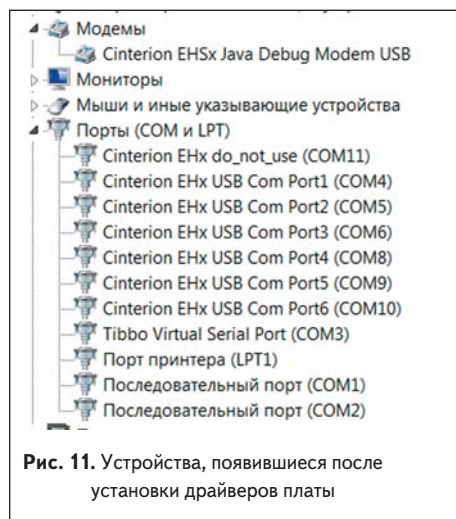


Рис. 11. Устройства, появившиеся после установки драйверов платы

В результате будет получено рабочее пространство, показанное на рис. 14.

Можно запустить традиционный Hello World и посмотреть на вывод в консоль (рис. 15).

Можно запустить программу мигания светодиодом Test_routine из проекта GPIO5v_lib. Но с ней будет не все так гладко. В этой

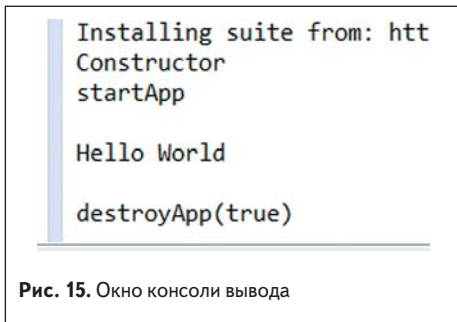


Рис. 15. Окно консоли вывода

программе не сделан перевод вывода GPIO7 в режим ввода. Поэтому вывод будет выглядеть так, как показано на рис. 16. А на самой плате Cinterion будет мигать только один светодиод.

Как видим, сигнал InPort всегда будет «0», а должен быть равен сигналу OutPort. Исправить ситуацию можно двумя путями: либо использовать микропереключатель, либо сделать это программным путем. Для ручного переключения GPIO7 в режим ввода надо перевести микропереключатель D5 в нижнее положение. Вместо желтого светодиода включится рыжий.

Практический пример

Подробное описание протокола обмена с CCU содержится в документе concept_board_hd_v01a.pdf. Его можно загрузить с сайта <http://m2m.gemalto.com/>, который становится доступным после регистрации. Приведу лишь практическую «выжимку» для конкретного случая.

Адрес CCU на шине I²C 0x69 или 1101001 в двоичном виде. Команда запись («1») или чтение («0») добавляется в конец послышки и становится восьмым битом. Таким образом, получается 0xD2 для записи или 0xD3 для чтения.

Следующий байт — адрес внутреннего регистра CCU. Нас интересует GPIO7, его адрес 0x13.

А последний байт послышки — команда включить канал GPIO на ввод 0x00 (на вывод — 0x01).

Итак, полная команда для перевода GPIO7 в режим чтения будет выглядеть так:

```
aD21300
```

Префикс «a» в начале строки — Message ID, он может быть только одной буквой.

Допишем в текст программы следующие строки:

```
I2cBusConnection cc = (I2cBusConnection) Connector.  
open("i2c:0;baudrate=400");
```

```
//int baudrate = cc.getBaudRate();  
InputStream inStream = cc.openInputStream();  
OutputStream outStream = cc.openOutputStream();  
String data = "aD21300";  
outStream.write(data.getBytes(), 0, data.length());  
outStream.flush();  
byte[] inBuf = new byte[12];  
if (inStream.available() > 0)  
inStream.read(inBuf);  
inStream.close();  
outStream.close();  
cc.close();
```

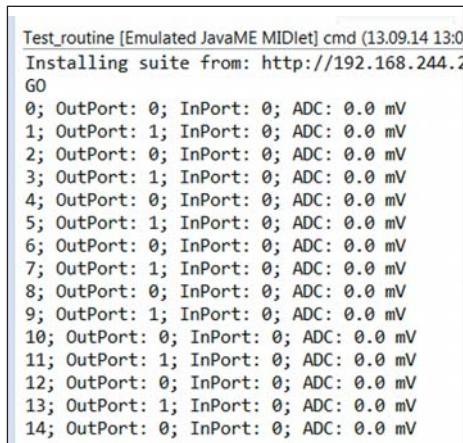


Рис. 16. Окно консоли вывода тестовой программы

Теперь в процессе выполнения программы канал GPIO7 будет автоматически переключен на ввод и программа целиком заработает так, как было задумано, без необходимости манипулировать вручную микропереключателями (рис. 17).

Соответственно получим синхронное мигание светодиодов (рис. 18).

Обратите внимание на положение микропереключателя D5, управляющего направлением порта GPIO7. Он стоит в положении «вывод», но включен рыжий нижний светодиод. Программно от CCU включен режим «ввод», и команда от CCU имеет более высокий приоритет.

Также я добавил в программу чтение показаний ADC:

```
ADC adc = new ADC(0,0);
```

И в основной цикл:

```
System.out.println("ADC: " + adc.getValue()  
*4,17 + " mV");
```

Не забываем умножить на 4,17 — учитывая поправочный коэффициент преобразователя напряжения на входе. Полученные 4,75 В — это

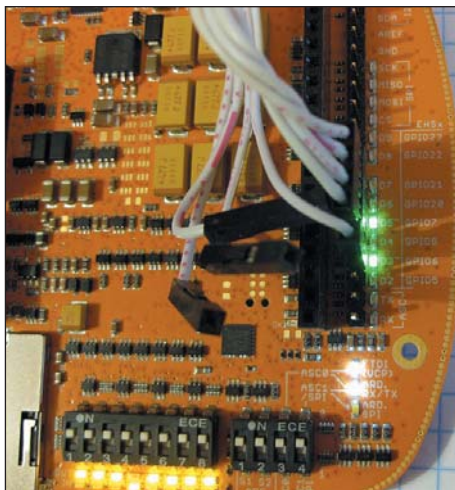


Рис. 18. Синхронное мигание светодиодов

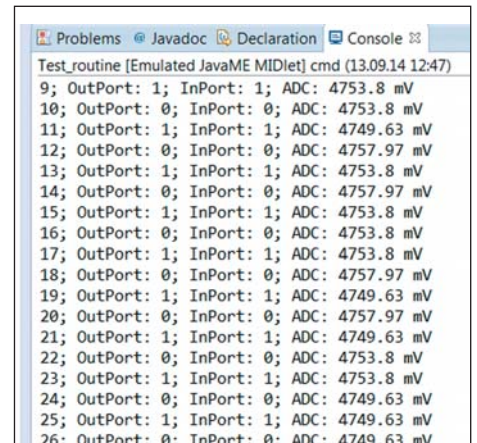


Рис. 17. Окно исправленной программы

напряжение питания минус падение напряжения на развязывающих элементах.

Добавим считывание нажатия кнопок на макетной плате и изменим программу так, чтобы она останавливалась нажатием любой из кнопок.

Добавим в текст программы строки:

```
Vector pins1 = new Vector(2);  
pins1.addElement("GPIO11");  
pins1.addElement("GPIO12");  
InPort inport1 = new InPort(pins1);
```

Изменим условие продолжения цикла:

```
while (inport1.get_Value() == 0).
```

Теперь программа крутится бесконечно, пока не будет нажата любая из кнопок на макетной плате.

И в качестве «вишенки на торте» добавим послышку SMS после нажатия любой из кнопок. Такая программа может рассматриваться как proof of concept работы системы, реагирующей на срабатывание внешнего датчика и посылающего SMS, информирующие пользователя о возникновении некоего условия. Модуль EHS6 правильно работает с AT-командами послышки текстовых SMS, что позволило не заниматься генерацией PDU-сообщения и решить задачу простым путем. В функции послышки SMS предварительно проверяется статус модуля — зарегистрирован ли он в сотовой сети.

Здесь будут приведены только дополнения, полный текст программы можно скачать по ссылке <http://ptelectronics.ru/stati/modul-ehs6-i-maketnaya-plata-cinterion-concept-board-v-rukah-spetsialista>.

В начале программы надо указать:

```
import com.cinterion.io.ATCommand;  
import com.cinterion.io.ATCommandFailedException;
```

Добавить модуль:

```
protected void sendSMS()  
{  
try {
```

```

System.out.println("Sending SMS in progress...");
ATCommand m_Cmd = new ATCommand(false);
String registration_response = m_Cmd.
send("AT+CREG?\r"); //Checking if module is registered
to the network
System.out.println(registration_response);
int locally_registered = registration_response.indexOf(«,1»);
System.out.println(locally_registered);
int roaming_registered = registration_response.
indexOf(«,5»);
if((locally_registered >-1) || (roaming_registered >-1)){
System.out.println("Module registered to the network");
String Response = m_Cmd.send("AT+cmgf = 1\r");
System.out.println(Response);
Response = m_Cmd.send("AT+CMGS=\"+79030189102\"
\r"); //here you put your number
System.out.println(Response);
Response = m_Cmd.send("(c) Sergey Dronsky" + (char)26);
//here you put text that is inside the SMS

```

```

System.out.println(Response);
}else{
System.out.println("Module not registered to the network");
}
} catch (ATCommandFailedException ex) {
ex.printStackTrace();
} catch (IllegalStateException ex) {
ex.printStackTrace();
} catch (IllegalArgumentException ex) {
ex.printStackTrace();
}
}
}

```

А также добавить вызов подпрограммы отправки SMS в конце программы:

```

outPort.release();
inPort.release();
sendSMS();

```

Заключение

В статье рассмотрены модуль EHS6 и макетная плата Cinterion Concept Board. Приведены простейшие способы общения с аппаратурой, вывод информации на разъем, считывание данных с разъема и аналого-цифрового преобразователя, отправление SMS из мидлета, установка и использование системы Eclipse для создания и отладки программ на Java ME 3.2. Все это можно реализовать на базе модуля EHS5, который является продолжением линейки Industrial и полностью совместим с модемами 2G BGS2 и BGS5. Следует отметить, что для полноценного использования модуля необходимо детально изучить всю сопутствующую документацию — и размещенную на сайте, и устанавливаемую на компьютер. ■