

Разработка игрового приложения для нестандартного восьмипроцессорного устройства с осязательным интерфейсом

С.О. Орлов, И.С. Егорушкин, Е.В. Никульчев

Аннотация – Рассматриваются используемые в современных гаджетах функции, такие как разделенный экран, и возможности применения гироскопа и акселерометра на примере нестандартного игрового устройства Wowcube. Данное устройство является умной игрушкой с осязательным интерфейсом и представляется собой аналог портативной консоли, с возможностью запуска не только приложений. Описывается конструкция программно-аппаратного комплекса нестандартного восьмипроцессорного устройства с осязательным интерфейсом. Приводится структура пакета, которым модули аппаратного комплекса обмениваются друг с другом для обеспечения синхронизации. Описывается способ обмена сообщениями и время жизни сообщений. Приводятся команды, применяемые при обмене сообщениями между платформой и запущенным приложением. Определяется список команд, используемых платформой и приложениями. С целью определения топологии сети модулей куба рассматривается пример работы механизма обмена сообщениями не только между платформой и приложением, но и между модулями. На примере разработанного приложения для нестандартного вычислительного комплекса, учитывающего его особенности и ограничения, описан процесс взаимодействия между его частями. Так же приведен пример работы с главным меню устройства. Описаны способы взаимодействия пользователя с осязательным интерфейсом.

Ключевые слова – программно-аппаратный комплекс, осязательный интерфейс, микроконтроллер, UART, геймификация, умная игрушка.

I. ВВЕДЕНИЕ

В некоторые современные мобильные гаджеты встроены гироскопы, использующиеся для измерения тангажа, крена и рыскания [1]. В сочетании с

акселерометром, который измеряет ускорение (и вибрацию), разработчики могут создавать приложения, которые способны определять движение по шести осям — вверх/вниз, влево/вправо, вперед/назад в сочетании с вращением вокруг трех перпендикулярных осей — тангажа, рыскания и крена [2]. С помощью гироскопа можно имитировать в гаджете: поворот объекта на экране, перемещение предметов, круговой обзор предметов и пространства и др. «Шутеры от первого» лица были первыми приложениями, использующими гироскоп. Часто планшет или смартфон используется как виртуальное окно.

Игры с использованием акселерометра применяются для отслеживания состояния здоровья и реабилитации [3, 4], контроля тренировочного процесса у спортсменов [5]. Акселерометр предшествовал гироскопу на мобильных устройствах, поэтому существует множество приложений, которые используют акселерометр. Например, функция «встряхнуть, чтобы перемешать» и гоночные игры используют акселерометр.

Приложение «Spacedesk», «Multiplicity» и им подобные позволяют использовать рабочее пространство гаджетов, таких как ПК, смартфоны, планшеты как единое пространство. Данные приложения обеспечивают соединение устройств используя следующие технологии: Wi-Fi, USB, LAN, Kernel-based Virtual Machine.

Разрабатываются и специальные устройства под определенные способы взаимодействия, к примеру, SandScape [6, 7].

В статье описывается разработанное приложение для уникального программно-аппаратного комплекса Wowcube — восьмипроцессорного устройства с осязательным интерфейсом, обладающего общим рабочим пространством, гироскопом, акселерометром, модулями комплекса общающимся между собой с помощью универсального асинхронного передатчика приемника (UART).

II. АППАРАТНЫЙ КОМПЛЕКС WOWCUBE

Wowcube - это электронное устройство (умная игрушка) [8], состоящее из 8 полностью идентичных модулей.

Статья получена 21.01.2020

Орлов Семен Олегович, Cubios Inc., Novato, CA, USA. (email: shtorkin@wowcube.com)

Егорушкин Илья Сергеевич, аспирант кафедры управления и моделирования систем, МИРЭА – Российский технологический университет (email: egoryshkin-ilya@rambler.ru)

Никульчев Евгений Витальевич, доктор технических наук, профессор кафедры управления и моделирования систем, МИРЭА – Российский технологический университет (email: nikulchev@mail.ru)

В центре куба механический элемент, к которому крепятся модули, вокруг этого элемента и происходит вращение. Каждый модуль имеет по экрану на трех «внешних» сторонах и по группе из 4 неодимовых магнитов на трех «внутренних». Внутри модуля расположен недорогой микроконтроллер, батарейка, шлейфы для экранов и контактов для магнитов [9].

Для взаимодействия модулей используется универсальный асинхронный передатчик приемник. Порты для обмена работают в дуплексном режиме и позволяют обмениваться данными не только с соседями, но и с любым другим модулем. В последнем случае модуль выступает в роли ретранслятора.

Каждая группа четырех магнитов состоит из следующих контактов: земля, зарядка, прием и передача (рисунок 1).

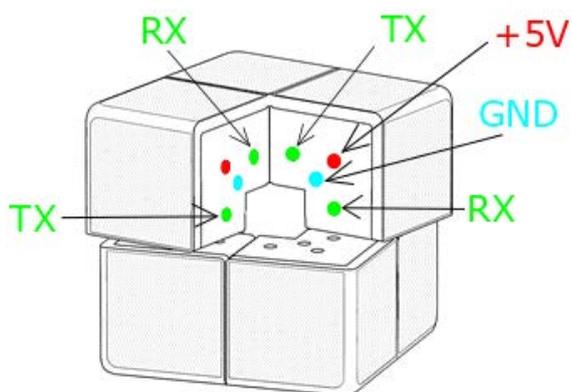


Рисунок 1 – Назначение контактов: TX – линия передачи UART, RX – линия приема UART, Зарядка с вольтажом 5V служащая обычной линией зарядки для всех модулей и Земля.

Линия зарядки и земли находятся напротив таких же линий соседнего модуля. В свою очередь линии RX и TX расположены в соответствии с протоколом передачи данных – линия передачи напротив линии приема.

Каждая линия представляет собой неодимовые магниты круглой формы, для переориентации и сдерживаемые пружиной. Внешний вид устройства представлен на рисунке 2.

Каждый микроконтроллер ограничен производственной мощностью и памятью, в которую должны вмещаться код платформы, скриптов и ресурсов приложений. Код приложений на кубе пишется на скриптовом языке PAWN. Код платформы, своего рода операционной системы для микроконтроллера, написан на языке C, на нем же написан PAWN. Язык PAWN выбран по ряду причин: простой и понятный синтаксис, т.к. он написан на C - это обеспечивает портируемость, и PAWN позволяет

запускать большие скрипты при ограниченных ресурсах памяти.



Рисунок 2 – Внешний вид wowcube

III. ОБМЕН ПАКЕТАМИ МЕЖДУ МОДУЛЯМИ

Модули постоянно обмениваются данными для обеспечения синхронизации (состояния, запущенного скрипта, приложения) и определения топологии, т.е. взаимного расположения модулей комплекса относительно друг друга. Структура сообщения:

- Уникальный идентификатор каждого модуля;
- Версия прошивки;
- Запущенный скрипт;
- Уровень подсветки экрана;
- В каком состоянии находится модуль сон/перезагрузка/нормальное;
- Заряд батареи;
- Резерв для дальнейшего расширения платформы;
- Вкл/выкл Bluetooth;
- Вкл/выкл режима отладки.

Вышеуказанным сообщением модули обмениваются все время, в том числе и в спящем режиме.

Для определения как далеко уйдет сообщение по сети, в сообщениях ставится флаг TTL (Time to live) или время жизни пакета. Если флаг равен 0, то пакет передается только соседнему модулю. Если равен 1, пакет получит соседний модуль и его сосед. Когда флаг равен 2, в платформе это максимальное значение, то пакет передается самому дальнему модулю от текущего - т.е. диагональному.

Как упоминалось ранее, каждый модуль имеет 3 линии UART, что дает возможность передавать сообщение как массовой рассылкой (как при broadcast) или по заданной линии.

При массовой рассылке с максимальным временем жизни сообщения, каждый модуль может получать одни и те же сообщения с разных линий несколько раз. Подобные случаи обрабатываются программно либо самой платформой, либо внутри приложения.

IV. ОБМЕН СООБЩЕНИЯМИ МЕЖДУ ПРИЛОЖЕНИЯМИ И ПЛАТФОРМОЙ

Максимальный размер сообщения при обмене между платформой и запущенным приложением равен 160 битам.

Тип сообщения определяется первым байтом сообщения. Платформа отправляет приложению следующие команды:

CMD_TICK – инициализирует один цикл работы приложения, посылается каждые 50 миллисекунд;

CMD_GEO – определяет порядковый номер модуля и его соседей;

CMD_NET_RX – получение сообщений от других модулей;

CMD_MTD – значения акселерометра и гироскопа на данном модуле.

Рассмотрим пример отправки сообщения для определения топологии. В приложение отправляется массив `uint8_t amx_pkt_geo[20]`, где каждый элемент имеет размер 8 бит, $8 * 20 = 160$ бит. В элементы массива данные записываются следующим образом:

0 - команда **CMD_GEO**

1 – максимальное количество экранов модуля

Далее последовательность повторяется для каждого экрана модуля:

2 - номер текущего (ID) модуля (номер 0-7)

3 - экран

4 - номер модуля сверху

5 - номер экрана модуля сверху

6 - номер модуля слева

7 - номер экрана модуля слева

В приложении данный пакет десериализуется Массив, хранящий топологию сети аппаратного комплекса, `abi_TRBL[CUBES_MAX][FACES_MAX]` записывается порядковый номер текущего модуля. Остальные элементы массива заполняются путем обмена командами (**CMD_NET_RX**) в методе `trbl_on_tick()`.

Выше упомянутый метод каждый цикл приложения, т.е. когда приходит команда **CMD_TICK**, отправляет свои данные по сети.

Каждый элемент массива заполняется при помощи побитовых операций сдвига и ИЛИ. Т.к. мы отправляем массив в приложение, где каждый элемент 8 бит, а в языке PAWN каждый элемент (ячейка) имеет размерность 32 бита, то чтобы узнать

верхнего и левого соседа каждого экрана мы используем следующие методы:

```
abi_topCubeN
abi_topFaceN
abi_leftCubeN
abi_leftFaceN
```

Данные методы имеют одинаковую логику работы, поэтому приведем пример кода только одного метода

```
abi_topCubeN(const _cubeN, const _faceN)
{
    return ((abi_TRBL[_cubeN][_faceN] >> 24) & 0xFF);
}
```

Отличие состоит только в количестве бит для сдвига. Для определения номера верхнего модуля сдвигается на 24 бита, для экрана верхнего модуля 16 бит, для левого модуля 8 бит, для экрана левого модуля сдвиг не применяется, а используется только побитовая маска.

Приложения так же отправляют команды платформе:

CMD_REDRAW – выводит содержимое фрейм буфера на заданный экран;

CMD_DRAW_OVERLAY – выводит общую отладочную информацию;

CMD_FILL – заполняет экран одним цветом;

CMD_LINE – помещает в фрейм буфер линию с заданными параметрами по заданным координатам;

CMD_RECT – помещает в фрейм буфер прямоугольник с заданными параметрами по заданным координатам;

CMD_NET_TX – с данной командой идет пакет, который требуется передать на другие модули;

CMD_SLEEP – переводит куб в спящий режим;

CMD_BITMAP – помещает в фрейм буфер спрайт с заданными координатами.

Как видно из вышеперечисленных команд, большинство команд связаны с рендером. Исключением являются команды вывода отладочной информации, которой пользуется приложение настроек и команда ухода в спящий режим из приложения выключения.

V. 4 ИГРОВОЕ ПРИЛОЖЕНИЕ

С целью демонстрации и использования основных возможностей устройства [10] было написано игровое приложение Pong (рисунок 3). Задача игрока - собрать все бонусы на всех экранах каждого модуля, активно

используя осязательный интерфейс и объединенное рабочее пространство.



Рисунок 3 – Внешний вид приложения Pong

Как упоминалось выше, обмен между модулями осуществляется через UART. Скрипт приложения и платформа обмениваются сообщениями посредством двоичного (бинарного) интерфейса приложений. Процесс перемещения объекта (в данном игровом приложении мяча) с модуля на модуль выглядит следующим образом:

1 модуль - Скрипт приложения отправляет сообщение на платформу;

```
abi_CMD_NET_TX(const line_tx, const TTL, const data[],
const with_pool = NET_SEND_WITH_TX_POOL)
{
    new pkt[5] = 0;
    pkt[0] = (((with_pool & 0xFF) << 24) | ((TTL & 0xFF) <<
16) | ((line_tx & 0xFF) << 8) | (CMD_NET_TX & 0xFF));
    for(new i=1; i<5; i++) pkt[i] = data[i-1];
}
```

1 модуль - Платформа ставит пакет в очередь отправки сообщений;

```
static void sendAMXPacket(uint8_t line_tx, uint8_t ttl,
uint32_t payload[NET_PKT_AMX_PAYLOAD_SIZE / 4]) {
    net_pkt_AMX_t net_pkt;
    PKT_createAMX(&net_pkt, line_tx, ttl, payload);
```

```
NVIC_DisableIRQ(TIM6_DAC_IRQn);
NETQ_TX_push(line_tx, (net_pkt_raw_t*)&net_pkt);
```

```
NVIC_EnableIRQ(TIM6_DAC_IRQn);
}
```

1 модуль - Сообщение отправляется через UART на соседний модуль;

```
if (((isrflags & USART_SR_TXE) != RESET) && ((cr1its &
USART_CR1_TXEIE) != RESET)) {
    CLEAR_BIT(huart->Instance->CR1, USART_CR1_TXEIE);
    huart->gState = HAL_UART_STATE_READY;
    NET_startTransmission(huart);
}
```

2 модуль - Полученное сообщение ставится в очередь полученных сообщений;

```
static void
restartDMAReceiveCallback(DMA_HandleTypeDef
*hdma) {
    UART_HandleTypeDef* huart = (
UART_HandleTypeDef* )(DMA_HandleTypeDef*
)hdma->Parent;
    huart->RxXferCount = 0U;
    NET_startReceive(huart);
}
```

2 модуль - Платформа отправляет в скрипт приложения сообщение о местоположении объекта;

```
for (int line_id = 0; line_id < NET_LINES_MAX; line_id++) {
    net_pkt_AMX_t pkt;
    while (AMXQ_pull(line_id, (net_pkt_raw_t*)&pkt)) {
        uint32_t amx_pkt_rx[5];
        amx_pkt_rx[0] = (((CMD_NET_RX & 0xFF) << 0)
((line_id & 0xFF) << 8) |
((pkt.header.neighbor_line_tx & 0xFF) << 16) |
((pkt.header.ttl & 0xFF) << 24));
        for (int i = 0; i < 4; i++) {
            amx_pkt_rx[i + 1] = pkt.data[i];
        }
        int amx_err = aux_Run((cell*)amx_pkt_rx,
sizeof(amx_pkt_rx) / sizeof(cell));
        if (amx_err != AMX_ERR_NONE) {
            SWITCH_ERROR_MODE("AMX message fail",
formatAMXError(amx_err));
            return;
        }
    }
}
```

2 модуль - Объект отрисовывается на заданном экране по заданным координатам.

```

abi_CMD_BITMAP(const resID, const x, const y, const
angle, const mirror)
{
  new pkt[3] = 0;
  pkt[0] = ((x & 0xFF) << 24) | ((resID & 0xFFFF) << 8) |
(CMD_BITMAP & 0xFF);
  pkt[1] = ((angle & 0xFF) << 24) | ((y & 0xFFFF) << 8) | ((x
& 0xFF00) >> 8);
  pkt[2] = (((mirror & 0xFF) << 8) | (angle & 0xFF00) >> 8);
}

```

Таким образом при помощи программно-аппаратного комплекса происходит обмен объектами между модулями, для конечного пользователя подобный переход выглядит бесшовным, а пространство единым целым (рисунок 4).



Рисунок 4 – Вид общего пространства для конечного пользователя

Подобный принцип перемещения объекта с одного модуля на другой используется и в главном меню устройства, для перемещения курсора. Внешний вид курсора и главного меню представлен на рисунке 5.



Рисунок 5 – Фото внешнего вида главного меню с курсором

Курсор перемещается в пределах плоскости стороны куба (как мы знаем из геометрии у куба 6 сторон). Для перемещения курсора на другую плоскость, необходимо поворачивать половину куба с курсором пока курсор не окажется на нужной плоскости. Для подтверждения выбора приложения используется гироскоп. Чтобы зайти в приложение, например в ранее описанное приложение Pong, необходимо потрясти куб три раза (аналог двойного щелчка мыши на персональных компьютерах).

VI. ЗАКЛЮЧЕНИЕ

В статье описана работа и взаимодействие программно-аппаратного комплекса нестандартного восьмипроцессорного устройства с осязательным интерфейсом — Wowcube. Описан процесс синхронизации работы модулей программно-аппаратного комплекса, а так же его особенности включающие время жизни сообщений. Представлен набор команд, применяемых для взаимодействия между платформой и запущенным приложением. Приведен пример отправки команды платформой и приема ее приложением. Описана работа игрового приложения учитывающего и использующего особенности данного устройства.

БИБЛИОГРАФИЯ

- [1]Славин О. А., Гринь Е. С. Обзор технологий виртуальной и дополненной реальности // Труды ИСА РАН. 2019. Т. 69. № 3. С. 42-54.
- [2]Pinho A. S., Salazar A. P., Hennig E. M. et al. Can We Rely on Mobile Devices and Other Gadgets to Assess the Postural Balance of Healthy Individuals? A Systematic Review // Sensors. 2019. Vol. 19. No. 13. P. 2972.
- [3]Höchsmann C., Walz S.P., Schäfer J. et al. Mobile Exergaming for Health—Effects of a serious game application for smartphones on physical activity and exercise adherence in type 2 diabetes mellitus—study protocol for a randomized controlled trial // *Trials*. 2017. Vol. 18. P. 103.

- [4] Kan V., Rajanen D., Asare K. O., Ferreira D. STOP: A Smartphone-based Game for Parkinson's Disease Medication Adherence // Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers. — ACM, 2018. P. 373-376
- [5] Подгурский, С. Е. Контроль скоростно-силовых возможностей квалифицированных спортсменов в тайландском боксе: современное состояние, проблемы и пути их решения // Научный часопис НПУ імені М.П. Драгоманова. 2019. № 3(111). С. 136-141.
- [6] Ratti C., Wang Y., Ishii H., Piper B., Frenchman D. Tangible User Interfaces (TUIs): a novel paradigm for GIS // Transactions in GIS. 2004. Vol. 8. No. 4. P. 407–421.
- [7] Piper B., Ratti C., Ishii H. Illuminating clay: a 3-D tangible interface for landscape analysis // Proceedings of the SIGCHI conference on Human factors in computing systems. — ACM, 2002. P. 355–362.
- [8] Osipov I. V., Nikulchev E. Review puzzles and construction sets falling under the category of augmented reality games // ITM Web of Conferences. 2008. Vol. 18. P. 02003.
- [9] Osipov, I. V. Cubios Transreality Puzzle as a Mixed Reality Object // International Journal of Virtual and Augmented Reality Vol. 1. No. 2. P. 1-17.
- [10] Osipov I. V., Nikulchev E. WOWcube Puzzle: A Transreality Object of Mixed Reality // Advances in Intelligent Systems and Computing. 2019. Vol. 881. P. 22–33.

Орлов Семен Олегович, инженер Cubios Inc., P.O. Box 22, Novato, CA, 94948, USA.

Егорушкин Илья Сергеевич, аспирант кафедры управления и моделирования систем, МИРЭА – Российский технологический университет

Никольчев Евгений Витальевич, доктор технических наук, профессор, профессор РАО, профессор кафедры управления и моделирования систем, МИРЭА – Российский технологический университет

Development of a game application for a custom eight-processor device with a tangible interface

S.O. Orlov, I.S. Egoryshkin, E.V. Nikulchev

Abstract - the article briefly discusses functions that are often used in modern gadgets, such as a shared working space in different devices, and features, such as a gyroscope and accelerometer, as well as an example of a non-standard device that uses this functionality. This device is a smart toy with a tangible interface and is an analogue of a portable console, with the ability to launch not only gaming applications. The second section describes the design of the hardware-software complex of a non-standard eight-processor device with a tangible interface. The third section describes the structure of the package, which the hardware modules exchange with each other to ensure synchronization. The method of messaging and the lifetime of messages are described. The fourth section describes the commands used when exchanging messages between the platform and the running application. The list of commands used by the platform and applications is determined. In order to determine the topology of the network of cube modules, an example of the operation of the messaging mechanism not only between the platform and the application, but also between the modules is considered. The fifth section, using the developed application for this complex as an example, taking into account its features and limitations, describes the process of interaction between its parts. An example of working with the main menu of the device is also given. The methods of user interaction with a tangible interface are described.

Keywords - hardware-software complex, tangible interface, microcontroller, UART, gamification, smart toy.

REFERENCES

- [1] Slavin O. A., Grin E. S. (2019) Overview of virtual and augmented reality technology, *Transactions of ISA RAS*, **69**(3), 42-54
- [2] Pinho A. S., Salazar A. P., Hennig E. M., Spessato B. C., Domingo A., Pagnussat A. S. (2019) Can We Rely on Mobile Devices and Other Gadgets to Assess the Postural Balance of Healthy Individuals? A Systematic Review, *Sensors*, **19**(13), 2972.
- [3] Höchsmann, C., Walz, S.P., Schäfer, J. *et al.* (2017) Mobile Exergaming for Health—Effects of a serious game application for smartphones on physical activity and exercise adherence in type 2 diabetes mellitus—study protocol for a randomized controlled trial. *Trials* **18**, 103
- [4] Kan V., Rajanen D., Asare K. O., Ferreira D. (2018) STOP: A Smartphone-based Game for Parkinson's Disease Medication Adherence. In Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, pp. 373-376.
- [5] Podhurskyi S. Control of the speed-strength capabilities of qualified athletes in Muay Thai: current status, problems and solutions. *Scientific journal National Pedagogical Dragomanov University*, **3**(111), 136-141
- [6] Ratti C., Wang Y., Ishii H., Piper B., Frenchman D. (2004) Tangible User Interfaces (TUIs): a novel paradigm for GIS. *Transactions in GIS*, **8**(4), 407-421.
- [7] Piper B., Ratti C., Ishii H. (2002) Illuminating clay: a 3-D tangible interface for landscape analysis. In Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 355-362.
- [8] Osipov I. V., Nikulchev E. (2018). Review puzzles and construction sets falling under the category of augmented reality games. *ITM Web of Conferences*, **18**, 02003.
- [9] Osipov, I. V. (2017). Cubios Transreality Puzzle as a Mixed Reality Object. *International Journal of Virtual and Augmented Reality*, **1**(2), 1-17.
- [10] Osipov I. V., Nikulchev E. (2019) WOWcube Puzzle: A Transreality Object of Mixed Reality. *Advances in Intelligent Systems and Computing*, **881**, 22-33.