

ФУРТАГ – НОВЫЙ СТАНДАРТ ПРОТОКОЛА ДЛЯ ЛАЗЕРТАГ

Бескин А.Л., Власов А.А., Одинокоев Г.А.

Московский технологический университет (МИРЭА), 119454, Россия, г. Москва, проспект Вернадского, 78, hi-jera@yandex.ru

Представлены результаты разработки программно-аппаратного комплекса Фуртаг для военно-полевой игры Лазертаг. Комплекс отличается от аналогов модульной архитектурой, гибкостью программирования параметров, лучшим соотношением цена-качество. Низкое энергопотребление обеспечено применением одного кристалла АТМеха8. Встроенное программное обеспечение создано в среде АWR Studio на ассемблере. Проект выполнялся в рамках конкурса УМНИК.

Ключевые слова: Лазертаг, Фуртаг, АТМеха8, модульная схема, CRC, Miles Tag, ассемблер

NEW STANDARD «FURTAG» OF CONNECTION PROTOCOL FOR CYBERSPORT

Beskin A.L., Vlasov A.A., Odinokov G.A.

Moscow Technological University (MIREA), 119454, Russia, Moscow, Vernadskogo avenue, 78, hi-jera@yandex.ru

Results of development of a hardware and software system of Furtag for the field game Laser tag are provided. The complex is distinguished from analogs by modular architecture, flexibility of programming of parameters, the best ratio the price quality. Low energy consumption is provided with application of one crystal of АТМеха8. The firmware is created in the environment of АWR Studio on the assembler. The project was executed within the competition УМНИК

Keywords: Lasertag, Furtag, АТМеха8, modular circuit, CRC, Miles Tag, Assembler

Введение

Военно-полевая игра «Лазертаг» приобрела всемирную популярность благодаря максимальному приближению к реальной боевой обстановке и полной безопасности. В России существует множество клубов поклонников игры и сложился рынок оборудования Лазертаг. На этом рынке доминирует компания LASERWAR [2], предлагающая сотни вариантов комплектов оборудования. Самое популярное оружие этой компании «Хищник» стоит 14.600 руб., не считая головной повязки с фотоприемниками. Для большей части целевой группы населения России эти цены неприемлемы (старшеклассники, студенты, молодые госслужащие в регионах).

Проект, описанный в данной статье, ставил цель создать программно-аппаратный комплекс, который мог конкурировать по своему функционалу на рынке оборудования Лазертаг, но имел бы следующие преимущества:

- модульная компоновка;
- использование дешевой и энергоэффективной элементной базы;
- гибкость настроек оружия и стратегии командной игры;
- совмещения двух видов оружия в стандартном решении;
- существенно более низкие цены на оборудование.

Разработка велась в рамках конкурса УМНИК (договор 8149ГУ2/2015 от 24.11.2015. Проект финансировал Фонд содействия инновациям [1]).

Основы игры Лазертаг и оборудования Фуртаг

Лазертаг или лазерный бой – это высокотехнологичная игра, происходящая в реальном времени и пространстве. Суть игры состоит в поражении игроков-противников безопасными инфракрасными выстрелами из бластера-автомата. «Поражение» игрока происходит путем регистрации луча бластера-автомата специальными датчиками оппонента (сенсорами), закрепленными на одежде игрока или на специальном жилете или головной повязке.

ATmega8 – восьмибитный микроконтроллер фирмы Atmel, используемый при создании комплекса Фуртаг.

Miles Tag II – доработанная и улучшенная версия протокола Лазертага, взятая за основу в Фуртаг[3].

«ИК-нож» – второй излучатель в конструкции одного Фуртаг-оружия.

Комплекс Фуртаг представляет собой приемо-передающие устройства в ИК-диапазоне, сконфигурированные особым образом, имитирующее какое-либо реально существующее или не существующее вооружение. Внешний вид вооружения может подбираться произвольным образом, обязательным элементом остается лишь обеспечивающая работу плата на основе ATmega8, а также ИК-передатчик и приемник. Кроме того, благодаря введению поддержки дополнительного ИК-излучателя («ИК-ножа»), есть возможность подключить два передатчика к одной плате, один из которых будет использоваться как оружие с бесконечными патронами.

В ходе работы были успешно достигнуты не только поставленные условия, но и добавлены несколько конструктивных особенностей:

- «ИК-нож»;
- чтение конфигурации оружия с microSD;
- улучшенное качество звука;
- работа от трех элементов питания формата AAA.

Протокол ИК-обмена оборудования основан на открытом протоколе Miles Tag II Data Protocol [3]. При разработке Фуртаг ставилась задача максимально обеспечить совместимость с оборудованием, использующим данный протокол. Тем не менее есть некоторые различия. При выстреле в протоколе Miles Tag II используется передача информационной посылки из 14 бит. В оборудовании используется дополненный формат (протокол Fur Tag), в котором задействовано 24 бита:

[Заголовок]-[0ppppppp]-[ttddd00]-[ccccccc], где

- p* - биты идентификатора игрока,
- t* - биты идентификатора команды,
- d* - биты отвечающие за силу выстрела,
- c* - контрольная сумма CRC.

Таким образом, можно сравнить информационные посылки, передаваемые в Miles Tag II и Fur Tag:

Miles Tag II:

[Заголовок]-[0ppppppp]-[ttddd]

Fur Tag:

[Заголовок]-[0ppppppp]-[ttddd00]-[ccccccc].

Благодаря идентичности начала посылок, оборудование на основе протокола Miles Tag II без проблем принимает пакеты Fur Tag. Оборудование Фуртаг проверяет число принятых байт в посылке - если их 14, то принятый сигнал обрабатывается как сигнал Miles Tag II. Если же принято 24 бита, то сначала проверяется правильность принятого информационного пакета по контрольной сумме, переданной вместе с ним, и в случае совпадения контрольных сумм выполняется часть кода, отвечающая за обработку попадания. Введение контрольной суммы позволяет значительно понизить количество ложных попаданий.

Структура команды/сообщения в Miles Tag II:

[Заголовок]-[Байт1]-[Байт2]-[0xE8]

Структура команды в Fur Tag:

[Заголовок]-[Байт1]-[Байт2]-[Контрольная сумма]

В Fur Tag были добавлены новые по сравнению с Miles Tag II команды. Ниже приведена таблица добавленных команд:

Таблица 1. Пакет протокола Fur Tag

Байт 1	Значение байта 1	Байт 2	Значение байта 2
0x83	Команда	0x11	Точка захвачена
		0x12	Бомба разминирована
0xA0	Радиация	1 – 100	Отнимает (байт 2) здоровья
0xA1	Аномалия	1 - 100	Прибавляет (байт 2) здоровья
0xA9	Сменить цвет	0 - 3	цвет команды
0xA2	Установить значение здоровья	1 – 200	Число единиц здоровья
0xA3	Дружественный огонь	0/1	Выкл./вкл.
0xA4	Дружественный ИК-нож	0/1	Выкл./вкл.
0xA5	Суицид	0/1	Выкл./вкл.
0xA6	Яркость основного ИК-излучателя	0 – 10	Процент яркости
0xA7	Яркость ИК-ножа	0 – 10	Процент яркости

Аппаратная платформа

Реализована аппаратная часть на базе микросхемы AtMEGA8L, спецификация L выбрана для снижения потребляемой мощности. Оборудование состоит из трех модулей: управляющий модуль (рис. 1,4) в котором находится микросхема и периферийные элементы, повязка с приемником сигнала (рис. 3) и само оружие содержащее передатчик (рис. 2).

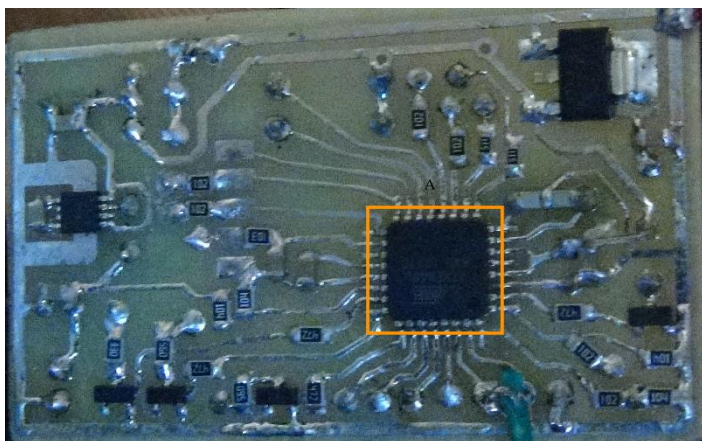


Рис. 1. Фото платы модуля обработки событий (А - микросхема ATmega8L)



Рис. 2. Фото оружия (А -передатчик инфракрасного сигнала, Б - коннектор, подключаемый к главному модулю)



Рис. 3. Фото повязки (А - приемник инфракрасного сигнала, Б - коннектор, подключаемый к главному модулю)



Рис. 4. Фото модуля обработки событий (А - вход для коннектора повязки, Б - вход для коннектора повязки, В - вход для коннектора зарядки)

Протокол передачи данных на внешний индикатор

Пакеты передаются друг за другом непрерывно, время между пакетами, как и между байтами данных не нормировано и не ограничено таймаутами. Для защиты от ошибок кадрирования применяются спецсимволы, кодирующие «Старт» и «Стоп», то есть разделение между пакетами. Для контроля целостности используется CRC-16.

При передаче каждая область данных передается в отдельном пакете, в начале пакета указывается код области, что бы было понятно какие данные в передаваемом пакете. Длина данных может варьироваться, поэтому расположение контрольной суммы (CRC-16) необходимо определять по двум последним байтам перед кодом «Стоп». Пакеты имеет формат, представленный на рисунке 5.

Старт	Код области	Данные	CRC-16	Стоп
-------	-------------	--------	--------	------

Рис. 5. Формат пакета передачи данных на внешний индикатор

Старт и Стоп – 1 байт начала и конца пакета, всегда равен 0x7E (126).

Код области – 1 байт, указывающий на тип данных, которые содержатся в данном пакете. Различаются следующие области:

0x00 – системная информация (номер версии железа, прошивки и т.д.);

0x01 – текущая конфигурация комплекта, транслируется конфигурация в том же виде, в какой она записана на флэш-карте начиная с идентификатора игрока;

0x02 – текущее состояние комплекта (игра, игрок убит, вне игры и т.д.);

0x03 – текущее состояние показателей игрока (патроны, обоймы, здоровье, время и т.д.).

Данные – собственно данные области, длина произвольная, минимум 1 байт.

CRC-16 – 2 байта (сначала младший, потом старший), контрольная сумма всего пакета без учета байта старта, стопа и самой контрольной суммы.

Инкапсуляции подвергаются все байты кроме старта и стопа, и значение которых равно 0x7E и 0x7D. Для передачи числа 0x7E используется последовательность из двух байт 0x7D, 0x5E. А для передачи числа 0x7D используется последовательность 0x7D, 0x5D.

$$[0x7E] = [0x7D, 0x5E];$$

$$[0x7D] = [0x7D, 0x5D].$$

Остальные байты передаются без изменений. При приеме пакета необходимо производить декапсуляцию, а уже потом обрабатывать принятый пакет. Декапсуляция сводится к поиску в принятых данных числа 0x7D и логической операцией XOR (исключающее ИЛИ) над следующим байтом числом 0x20, полученный результат и будет истинное значение инкапсулированного числа.

$$0x5E \text{ xor } 0x20 = 0x7E;$$

$$0x5D \text{ xor } 0x20 = 0x7D.$$

В алгоритм передачи команд также была добавлена контрольная сумма, алгоритм расчета которой отличается от алгоритма расчета контрольной суммы для выстрела. Контрольная сумма рассчитывается по принципу циклической контрольной суммы по полиному 0xA001. При инициализации в регистр суммы записывается число 0xFFFF.

Алгоритм расчета приведен на рисунке 6 (повторяется для каждого байта 8 раз):

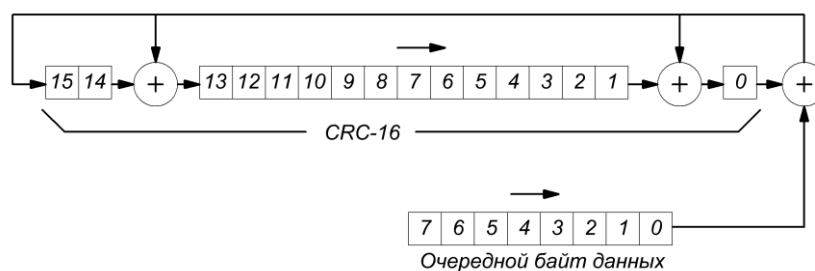


Рис. 6. Схема расчета контрольных сумм

Каждый байт пропускается поочередно. Итоговый результат, получившийся в 16-битном регистре контрольной суммы, является контрольной суммой пакета и передается вслед за данными.

Программное обеспечение

Программное обеспечение оборудования было разработано с использованием AVRStudio на ассемблере. Ассемблер был выбран вследствие ограниченной производительности и возможностей микроконтроллера ATmega8. Использование ассемблера позволило нам одновременно обрабатывать входные и выходные данные, а также воспроизводить аудиофайлы, избегая замираний на отдельных алгоритмах в классической последовательной схеме. Общий объем ПО на языке ассемблера составил около 2800 строк.

Для подготовки карт microSD было написано приложение Windows в открытой среде программирования Lazarus. Приложение размечает карту памяти по специальному шаблону и записывает данные, определяющие конфигурацию оружия и другие параметры игры (см. рисунок 7).

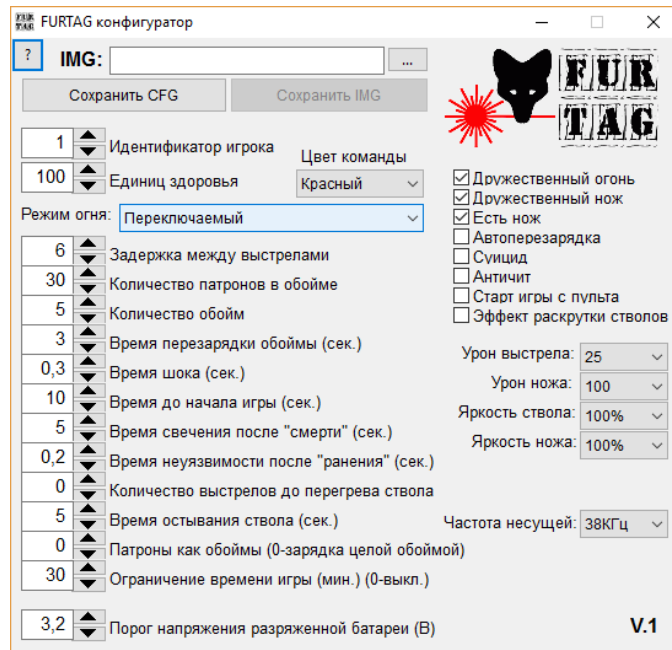


Рис. 7. Форма приложения-конфигуратора

Отдельная утилита (см. рис.8), также сделанная в среде Lazarus, позволяет создавать множество образов карт памяти с общей конфигурацией, но отличающиеся набором звуков (до 128 файлов типа snd***.wav). На входе – файл конфигурации *.cfg и звуковые файлы. На выходе – готовая прошивка (файл *.img).

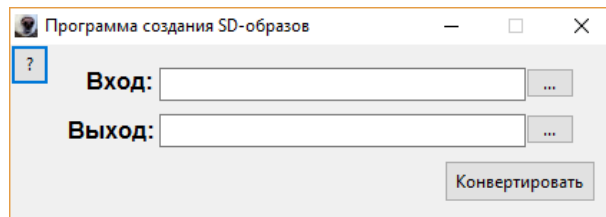


Рис. 8. Форма создания образов карт со звуками

Схема работы программного обеспечения, встроенного в оружие, представлена на рисунке 9.



Рис. 9. Слева – классическая схема обработки данных, справа – с использованием прерываний

С точки зрения ПО, аудиоданные являются потоком байтов, которые считываются из SD-карты и передаются на один из выходов микроконтроллера в строго ограниченном объёме, чтобы уложиться в определённое число тактов. Это позволяет избежать ситуации, когда звук воспроизводится, а входные и выходные данные никак не обрабатываются (выстрелы не производятся, попадания не принимаются).

Реализация ПО на языке высокого уровня потребовала бы больших усилий и продумывания архитектуры, поэтому при решении аналогичных задач с использованием языка C обычно ставят дополнительный чип для обработки звука. Таким образом распараллеливается работа устройства и обеспечивается корректная работа оружия и своевременное воспроизведение звука. Мы же смогли обойтись всего одним микроконтроллером для всех комплектов оружия и вариантов его прошивки.

Для перехвата управления и перехода на нужный блок мы использовали одну из особенностей микроконтроллера – обработку прерываний. Прерывания – это событие (прием сигнала, по которому прерывается исполнение основного кода программы и управление передаётся функции обработчику прерывания).

Ниже представлен небольшой фрагмент кода на языке ассемблера для процессора ATmega8.

```

sei
rcall    led1_on           ;стартовое мигание LED1
ldi     r18,50
rcall    pause
rcall    led1_off
lds     r16,k_now         ;проверка удержания "НОЖ"
sbr     r16,4
cpi     r16,0b1111101
brne    ok1
ser     r16               ;поднимаем флаг, что флешки нет
sts     no_sd,r16
rjmp    skip_ini         ;пропускаем инициализацию и тест железа
ok1:    rcall    ini_sd    ;инициализация SD
lds     r16,no_sd        ;загрузка таблицы расположения данных на SD
tst     r16
brne    no_sd1
ldi     r16,$51          ;отправка команды ЧТЕНИЕ (CMD17)
sts     pack,r16

```

Основной алгоритм встроенного в оружие Фуртаг программного обеспечения представлен на рисунке 10. Программное обеспечение (рис.7 и 8) для настройки оборудования было обновлено для обеспечения совместимости с большим количеством операционных систем.

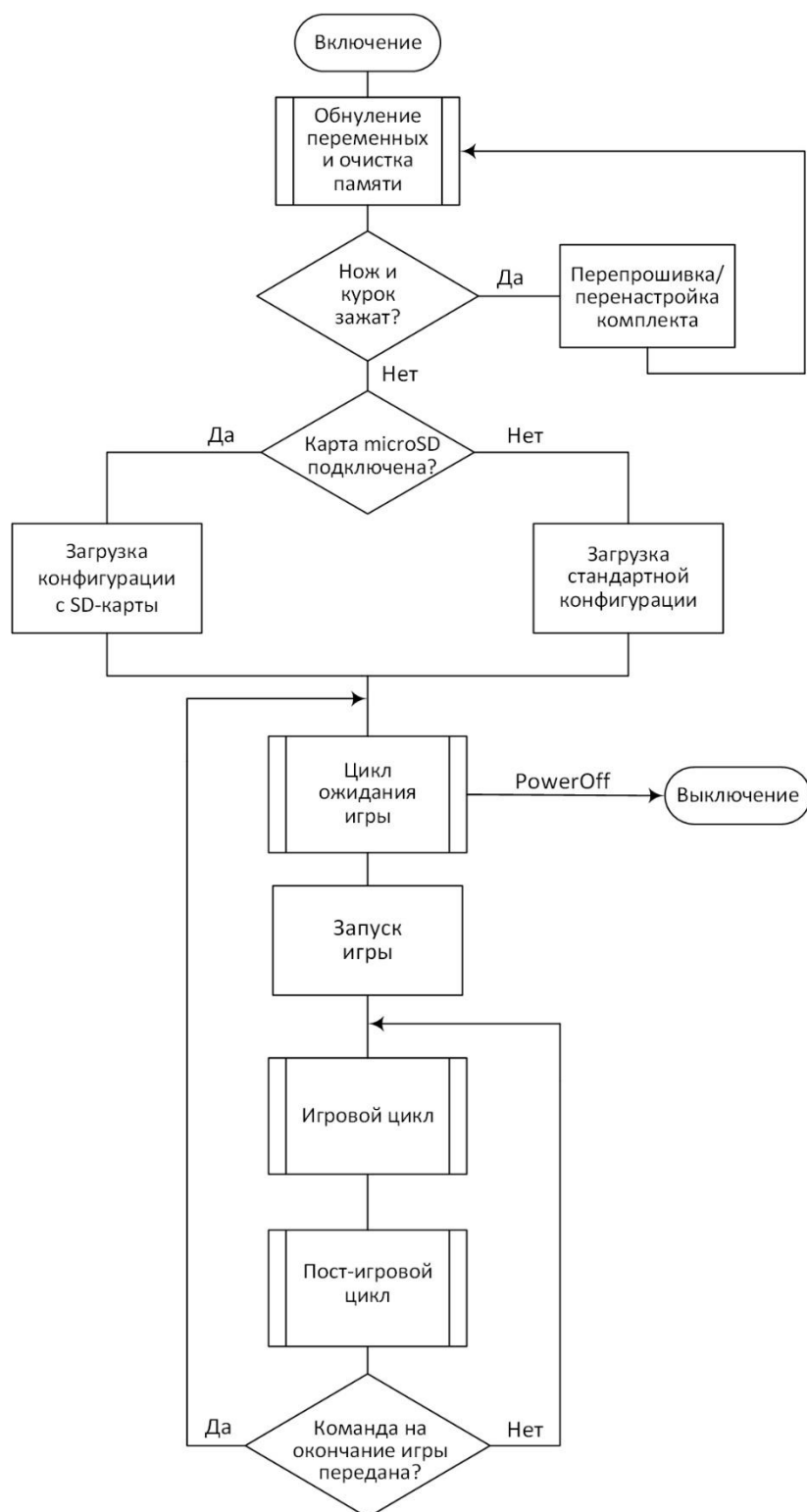


Рис. 10. Алгоритм встроенного ПО Фуртаг

Для отладки корректности передаваемых пакетов ИК-сигналы наводились на инфракрасный порт, встроенный в стандартный компьютер. Также проводились испытания на основе двух готовых комплектов Фуртаг.

Калибровка точности попадания ИК сигнала производилось при помощи анализа излучения на предмет

попадания в нарисованную мишень, расположенную на расстоянии метра. Так как невооруженным глазом увидеть инфракрасное излучения невозможно, применялась бытовая веб-камера со снятым инфракрасным фильтром, установленная на штатив. Камера отслеживала сигналы из ИК-излучателя, которые видны как яркие белые вспышки, и точка попадания сигнала сопоставлялась с точкой прицеливания.

Для увеличения возможностей комплекса и его гибкости была разработана модульная концепция оборудования (рисунок 11).

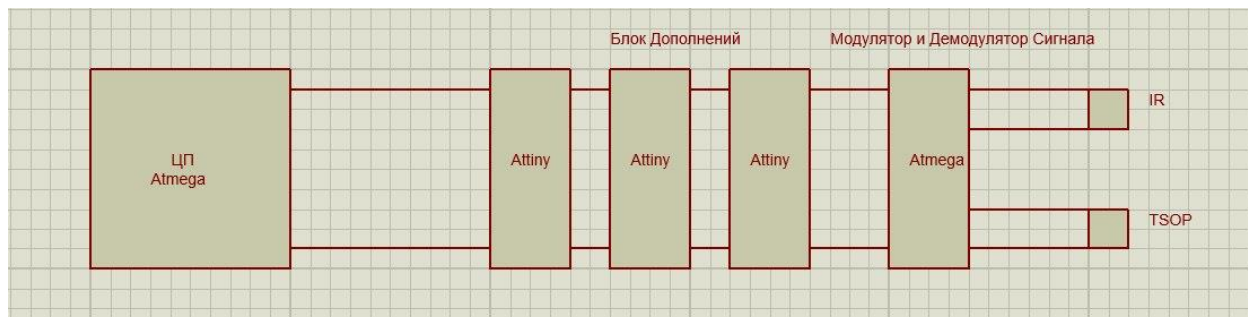


Рис. 11. Модульная схема с пред- и пост-обработкой сигналов

Альтернативная схема, изображенная на рис.12, полностью обрабатывающая сигнал. Такой подход позволит сохранить совместимость модулей со старым оборудованием.

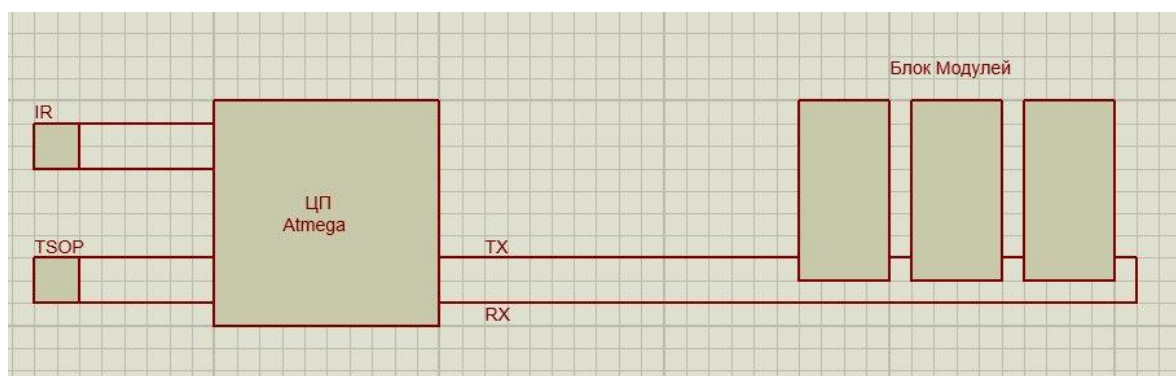


Рис. 12. Модульная схема полной обработки сигналов

Были исследованы предпочтительные материалы корпуса, переход на которые позволит увеличить прочность оборудования. Сравнение наиболее подходящих материалов приведено в таблице 2.

Таблица 2. Параметры материалов для корпуса

Материал	Полипропилен	АБС-пластик	Капролон
Плотность материала, кг/м. куб.	900-920	1020-1080	1150-1160
Температура плавления, С	160 – 170	110 -130	220-225
Рабочий диапазон температур, С	-40 – 70	-40 – 70	-40 - 70
Твердость по Бринеллю, МПа	60-65	90-150	Не менее 130
Предел прочности при сжатии, МПа	80-110	46-80	Не менее 90
Предел прочности при изгибе, МПа	55-130	45-87	Не менее 80
Модуль упругости, МПа	6.7-11.9	1.7-2.9	2.0-4.0

Предпочтительным материалом для корпусов был выбран капролон (полиамид-6, блочный), как наиболее прочный. На основе капролона был сделан первый прототип корпуса. Каркасный вариант (рисунок 13) позволяет добиться большей прочности конструкции при меньших затратах на производство и гораздо меньшей сложности процесса производства.

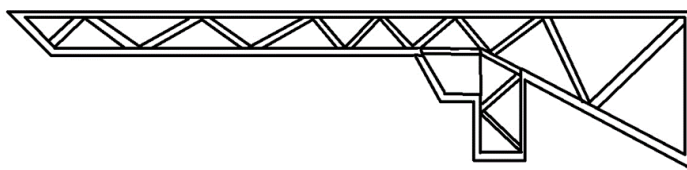


Рис. 13. Проект каркаса имитатора стрельбы

Все аналоги используют технологию литья корпуса (рис.14), которая требует специализированного оборудования. Каркасный вариант позволяет создать образец на основе заменяемых модулей, что дает возможность сделать устройство универсальным. Имея одну основу и много различных заменяемых модулей, мы получаем возможность их комбинирования, что при использовании технологии литья обычно приводит к потере прочности. Модульный вариант также упрощает процесс производства и замены комплектующих и ремонт в случае поломки.



Рис. 14. Образцы корпусов, выполненных методом литья



Рис. 15. Прототип каркаса для навески модулей оборудования

На рисунке 15 изображен первый прототип каркаса с предварительно размещенной электроникой. Каркасный вариант сильно упрощает процесс размещения электроники, позволяет сделать большое количество вариантов внешнего дизайна, не меняя при этом основу конструкции и не требует для этого создания целиком новой формы для полного изменения внешнего вида. Это дает возможность потребителю

выбирать понравившийся ему дизайн из линейки предложенных, а также изготавливать дизайн на заказ с минимальными затратами.

Заключение

В результате был разработан достаточно бюджетный вариант Лазертаг-оружия, обладающего унифицированными основными элементами, что позволяет удешевить производство. Комплекс обладает характеристиками на уровне современного оборудования российских фирм, превосходит его по некоторым параметрам. Себестоимость комплекта оборудования Фуртаг составила около 5 тысяч рублей.

Модульная конструкция позволяет переносить блоки оборудования на другие платформы, в том числе на управляемые модели автомобилей и летательные аппараты. Проведены предварительные расчеты для коммерциализации командной игры на базе оборудования Фуртаг, размещенного на радиоуправляемых моделях автомобилей.

Список литературы

1. Фонд содействия инновациям. Конкурс УМНИК.
http://fasie.ru/competitions/?PROGRAM_ID=130
2. Компания LASERWAR. <https://laserwar.ru>
3. Протокол Miles Tag II. <http://hosting.cmалton.me.uk/chrism/lasertag/MT2Proto.pdf>
4. Build Your Own Laser Tag System. <http://www.lasertagparts.com>
5. Применение MMC-карт в микроконтроллерных системах
<http://www.soel.ru/cms/f/?/311635.pdf>

References

1. Fund of assistance to innovations. Competition UMNİK
http://fasie.ru/competitions/?PROGRAM_ID=130
2. Company LASERWAR. <https://laserwar.ru>
3. Miles Tag II protocol. <http://hosting.cmалton.me.uk/chrism/lasertag/MT2Proto.pdf>
4. Build Your Own Laser Tag System. <http://www.lasertagparts.com>
5. Use of MMC cards in microcontroller systems
<http://www.soel.ru/cms/f/?/311635.pdf>