

момент именно геймдев сильнее всего стимулирует развитие технологий, связанных с компьютерной графикой и рендером. Именно в разрезе игр впервые обрело законченный вид такое направление, как «виртуальная реальность». Так же, растущие объёмы обработки подталкивают к созданию всё более быстродейственных алгоритмов.

Как уже отмечалось ранее, любой контент, который создаётся для игрового проекта, должен обладать максимальной гибкостью, иметь возможность быстро изменять параметры и расширяться. Как пример можно привести создание искусственного интеллекта. Ограничения по производительности сразу ставят жесткие рамки для написания логики. Так же, крайне важно, чтобы, написав один модуль, можно было получить множество совершенно различных ИИ, регулируя лишь несколько параметров. Благодаря этому, экономится не только время на разработку, но и уменьшается вероятность возникновения ошибок.

Так же, при разработке игрового проекта, большое значение имеет раннее прототипирование. Крайне важно иметь возможность быстро проверить идею на жизнеспособность, затратив на это минимум сил и времени. Данный этап во многом зависит от технологий, которые используются при разработке, и от навыков команды. На данном этапе крайне важно понимать, что качество не имеет никакого значения. Основная идея – посмотреть на то, как будет воспринят итоговый продукт потенциальной аудиторией.

Подводя итог: разработка игр является комплексной задачей, которая за своей кажущейся несерьёзностью скрывает внушительный объем решаемых технических задач и, в общем, вовсе не является тривиальной. Популярность данного направления в данный момент привлекает всё более серьёзные вложения и, как следствие, технологии в геймдев и превращает его в серьёзного игрока на рынке IT.

Список использованных источников:

1. Документация к UnrealEngine 4, URL: <https://docs.unrealengine.com/latest/INT/>
2. Scott Rogers "Levelup! TheGuidetoGreatVideoGameDesign". – Wiley, 2014. – 492 с.

СИСТЕМА УПРАВЛЕНИЯ КОДОВЫМ ЗАМКОМ НА БАЗЕ ПЛАТФОРМЫ ARDUINOMEGA 2560

*Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь*

Порхун М. И.

Качинский М. В. – кандидат технических наук, доцент

В современном мире одной из главных задач является обеспечение безопасности. Это обязывает использовать системы контроля доступа к информации. Одним из представителей такой системы является кодовый замок.

Применение подобного способа контроля доступа в помещение подразумевает использование секретной комбинации десятичных цифр (в нашем случае четырёх), позволяющей осуществлять управление кодовым замком. Для реализации системы был выбран микроконтроллер ATmega 2560, как одна из самых популярных платформ для реализации различных устройств. Платформа запрограммирована в соответствии с алгоритмом работы устройства на языке, схожим с C/C++. Для проверки работы устройства выбрана система автоматизированного проектирования Proteus.

Данный кодовый замок обеспечивает:

- 1) Доступ в помещение только после набора секретной кодовой комбинации, представляющей собой четырёхзначное десятичное число;
- 2) Предоставление ограниченного количества попыток (в нашем случае трёх) набора секретной кодовой комбинации;
- 3) Звуковую и визуальную сигнализацию, а также блокировку двери на пять минут при попытке несанкционированного доступа;
- 4) Оперативную смену секретной кодовой комбинации;
- 5) Подсветку клавиатуры при приближении человека к двери;
- 6) Оригинальное звуковое сопровождение при открытии двери;

Структурная схема устройства приведена на рисунке 1.

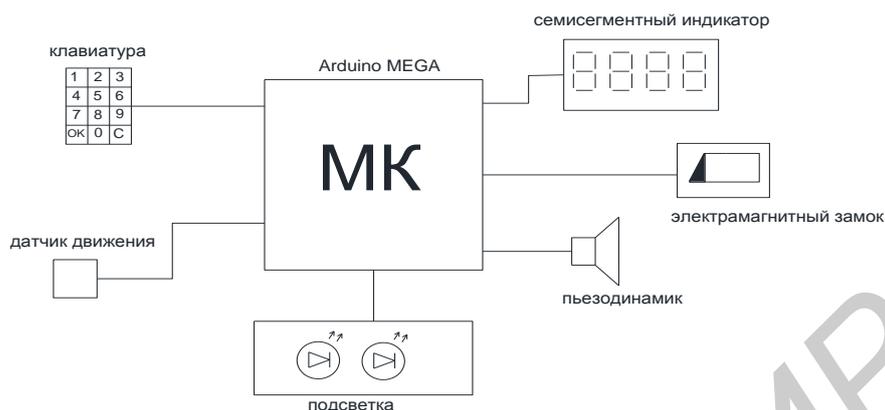


Рис. 1. Структурная схема системы управления кодовым замком

Упрощённый алгоритм работы устройства приведён ниже:

Шаг 1. Начальная инициализация состояния микроконтроллера;

Шаг 2. Анализа датчика движения:

- Если он активен в течение пяти секунд, то перейти на шаг 3;
- Если нет, то вернуться к первому шагу.

Шаг 3. Зажечь подсветку клавиатуры;

Шаг 4. Анализ клавиатуры (ожидание нажатия клавиши (а точнее четырёх клавиш)):

- 1) Для цифровых клавиш:
 - Вывод десятичного эквивалента нажатой цифры на семисегментный индикатор;
 - Запись кода нажатой клавиши в буфер;
- 2) Для функциональных клавиш:

Если нажат сброс:

- Очистить дисплей и буфер кодов введённых клавиш. Перейти к шагу 2.

Если нажата клавиша ОК:

- Проверить соответствие секретной кодовой комбинации введённой комбинации;
- Если коды совпадают, то перейти к шагу 5, если нет, то к шагу 7.

Если введена последовательность четырёх нулей (режим программирования):

- Просьба ввести секретную комбинацию;
- Если нажата клавиша сброса, то очистить семисегментные индикаторы и буфер введённых клавиш, перейти к шагу 2;
- Если она верна, то просьба ввести новый код, если нет, то перейти к шагу 2;
- Сохранение кода, переход к шагу 2.

Шаг 5. Процедура открытия замка:

- Открыть замок;
- Очистить семисегментный индикатор;
- Проиграть звуковой сигнал;
- Подождать десять секунд. Перейти к шагу 6.

Шаг 6. Закрыть замок, выключить подсветку клавиатуры. Перейти к шагу 2.

Шаг 7. Процедура обработки ошибочно введённой секретной кодовой комбинации:

- Инкрементировать «счётчик ошибок»;
- Проверить, третий ли раз был неверно введён код? Если да, то:
 - Перевести клавиатуру в «режим блокировки» (на пять минут);
 - «Поморгать» подсветкой клавиатуры (в течение десяти секунд);
 - Проиграть звуковое предупреждение;
 - Ждать пять минут;
 - После истечения этого времени перейти к шагу 2.
- Если нет, то перейти к шагу 2.

Замечания по данному алгоритму:

- 1) Клавиша ОК сбрасывает только после ввода четырёх цифровых значений;
- 2) Сброс семисегментного индикатора и буфера нажатых клавиш производится нажатием на клавишу С;
- 3) Вход в режим программирования осуществляется путём ввода заранее определённой кодовой комбинации (четыре нуля) и только при сброшенном буфере нажатых клавиш;

Для определения надёжности системы была построена её модель в САПР Proteus и протестирована в различных ситуациях. Симуляция показала, что устройство работает корректно.

Таким образом, была разработана система управления кодовым замком на базе платформы

Arduino MEGA 2560. Рассматриваемая система за счет представленного вышеалгоритма обеспечивает отличную защиту от несанкционированного доступа в помещение, звуковую и визуальную сигнализацию, при попытке несанкционированного доступа, оперативную возможность смены секретной кодовой комбинации и простой, интуитивно понятный для пользователя интерфейс – всё это выделяет ее на фоне других систем.

Список использованных источников:

1. Соммер У. Программирование микроконтроллерных плат Arduino/Freduino / СоммерУ // Для радиолюбителей. – СПб.: БХВ – Петербург, 2012 г. – 256 с.
2. Сайт, посвященный программированию платформы Arduino [Электронный ресурс]: 2016 г. URL: <http://arduino.ru/>
3. Сайт, посвященный программированию платформы Arduino [Электронный ресурс]: 2016 г. URL: <http://arduino.cc/>

СИСТЕМА ДЕТЕКТИРОВАНИЯ КЛЮЧЕВЫХ ФРАЗ В РЕЧЕВОМ СИГНАЛЕ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Суша А. В.

Вашкевич М. И. – к-т. техн. наук, доцент

В работе приводится краткая характеристика основных методов детектирования (поиска) ключевых фраз в речевом сигнале. Так же описывается алгоритм работы системы, построенной с применением метода поиска по шаблону. Сравнение шаблона ключевой фразы с фрагментом сигнала осуществляется с помощью DTW-алгоритма.

Поиск ключевых фраз (англ. – keyword spotting) – речевая технология, связанная с решением задачи поиска и локализации заданного слова или целой фразы в речевом потоке. Задача поиска ключевых слов в речевом потоке может быть разделена на несколько направлений:

- непосредственно выявление и определение места ключевого слова или словосочетания (поиск в фонограмме речи, речевых базах, поиск в реальном времени);
- распознавание команд в слитном потоке речи;
- понимание смысла речи посредством поиска ключевых слов или фраз, для диалоговых систем.

Для решения задачи поиска ключевых фраз было разработано множество различных способов, из которых можно выделить основные методы. Ниже приведены широко используемые в настоящее время методы поиска ключевых фраз [1].

Метод *поиска по шаблону* является наиболее распространённым. Суть его заключается в том, что для каждого ключевого слова записывается несколько вариантов его произнесения различными дикторами, на основе чего создаётся шаблон, который используется для организации поиска данного слова в речевом потоке. Главным недостатком такого подхода является то, что создание шаблона требует запись каждого слова микрофоном (слово нельзя просто ввести с клавиатуры).

Скрытые марковские модели (сокращённо – СММ) широко используются для решения задачи поиска ключевых фраз. Для каждого ключевого слова создаётся соответствующая СММ, которая используется при поиске в речевом потоке, и одну «модель мусора» для всех остальных слов. Временная последовательность символов ключевых слов и мусор символов формируется в результате распознавания речевой последовательности. Недостатком метода является то, что для каждого нового ключевого слова нужно не только обучать новую СММ модель, но также нужно заново обучать модель мусора.

Стоит отметить метод *поиска ключевых фраз по фонемной (слоговой) решётке*. Метод основывается на конструировании большой сети возможных звуков (слов) в различные моменты времени, по которой осуществляется поиск возможных произношений слова или фразы. Основное преимущество этого метода в том, что он обладает большой гибкостью: даже если фонема ключевого слова не является лучшей гипотезой между узлами решётки, она все равно сохраняется в результате распознавания. Но в то же время, построение такой решётки довольно трудоёмкий процесс.

В качестве основы был выбран метод поиска ключевых фраз по шаблону. Данный метод был выбран в связи с его относительной простотой и отсутствием необходимости предварительного обучения и переобучения. Описанный далее алгоритм работы системы детектирования ключевых фраз в речевом сигнале основывается на алгоритмах, описанных в публикациях [2, 3]. Структурная схема алгоритма приведена на рисунке 1. Рассмотрим алгоритм работы системы по шагам.

Подготовительный шаг. Осуществляется фрагментация шаблона ключевого слова на M фреймов, длиной 30 мс. На основании из каждого i -го фрейма формируется вектор характеристических признаков $\bar{a}_i, i = 0, M - 1$. В качестве этих признаков были выбраны *мел-частотные кепстральные коэффициенты* (сокращённо – MFCC) [4]. Длина вектора характеристических признаков выбирается произвольно, в работе использовались 13 коэффициентов. В результате этих действий, формируется последовательность векторов характеристических признаков $A_M = \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_{M-1}\}$.