

C_n^k – количество сочетаний;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Способ третий. Размещения по сути, являются сочетаниями, в которых важен порядок элементов. Другими словами, это перестановки сочетаний. Количество размещений находится по формуле:

$$A_n^k = \frac{n!}{(n-k)!} \quad (3)$$

A_n^k – количество размещений;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Способ четвертый. Размещения с повторениями – это все варианты массивов длиной k , на каждой позиции которых может быть любой элемент из множества одномерного массива n . Количество размещений с повторениями находится по формуле:

$$\overline{A_n^k} = n^k \quad (4)$$

$\overline{A_n^k}$ – количество размещений с повторениями;
 n – количество элементов одномерного массива;
 k – количество элементов необходимой комбинации.

Перебор различных комбинаторных конфигураций объектов и выбор среди них наилучшего – довольно часто встречаемая задача в работе программиста. Потому знание алгоритмов генерации наиболее распространенных комбинаторных конфигураций является необходимым условием успешной работы IT-специалиста. Важно также знать количество различных вариантов для каждого типа комбинаторных конфигураций, так как это позволяет реально оценить вычислительную трудоемкость выбранного алгоритма решения той или иной задачи на перебор вариантов и приемлемость этого алгоритма для решения рассматриваемой задачи, с учетом ее размерности.

Также при решении задач необходимо умение выполнять следующие операции для каждой из комбинаторных конфигураций: по имеющейся конфигурации получать следующую за ней в лексикографическом порядке; определять номер данной конфигурации в лексикографической нумерации всех конфигураций; и, наоборот, по порядковому номеру выписывать соответствующую ему конфигурацию.

Литература:

1. Бродский Я. С. Статистика. Вероятность. Комбинаторика / Я. С. Бродский. – М.: ООО «Издательство Оникс»: ООО «Издательство «Мир и Образование», 2008. – 544 с.: ил. – (Школьный курс математики).
2. Семакин И. Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И. Г. Семакин, А. П. Шестаков. – М.: Издательский центр «Академия», 2013. – 304 с.

РАЗРАБОТКА СЕРВОТЕСТРА

Мацкевич А.С.,

*обучающийся 3 курса Оршанского колледжа ВГУ имени П.М. Машерова,
г. Орша, Республика Беларусь*

Научный руководитель – Романцов Д.Ю., магистр техн. наук, преподаватель

Введение. Сервотестер – это устройство для тестирования и настройки сервоприводов, которые являются ключевыми компонентами в системах управления механическими устройствами. Сервопривод или сервомашинка – это электродвигатель с блоком управления, который за счёт обратной связи может точно поддерживать заданное положение вала и необходимый угол поворота [1].

Основная задача сервоприводов заключается в точном управлении положением и движением в механизмах, что делает их очень важными для робототехники, авиации, автомобильной промышленности и других сфер, где требуется высокая степень контроля. Сервотестеры дают возможность проверять работоспособность и просматривать параметры сервоприводов без необходимости подключения их к сложным системам управления. Они обеспечивают простой и эффективный способ обслуживания и настройки сервомашинки, что существенно сокращает время и ресурсы, затрачиваемые на тестирование и ремонт.

Цель исследования: разработка собственного сервотестера с возможностью подключения нескольких сервомашинки. Также должны быть: настройка угла поворота вала сервомашинки, настройка тримера, проведение теста на усилие, проведение теста времени поворота вала на 60 градусов, изменение выходного напряжения на сервомашинки, замер силы тока и справочные материалы.

Материалы и методы. Разработка проводилась в несколько этапов.

1) Подбор компонентов и проектирование системы. На данном этапе был выбран микропроцессор и архитектура, отвечающие требованиям. В начале исследования за основу был взят микропроцессор ATmega168PA, в последующем ATmega328P, которые различаются лишь объёмом памяти. А также датчик веса XFW-HX711, дисплей ST756S7-128X64 с разрешением 168 на 64 пикселя, шунт с сопротивлением 0.0075 Ом, Bluetooth модуль UART и дигипод 41010 SN1047. Затем была спроектирована плата, на которой в были размещены компоненты.

2) Написание кода. В качестве языка программирования был выбран язык C, который подходит под поставленную задачу и не имеет ООП что даёт уменьшение веса прошивки [2]. Написание кода происходило в IDE Microchip Studio. Для отладки сигналов использовался осциллограф и анализатор сигналов.

3) Корпус был смоделирован через 3D редактор Blender и затем распечатан на 3D-принтере.

Результаты и их обсуждение. Во время разработки были следующие сложности и недочёты:

1) Объёма ATmega168PA (16Kб) оказалось недостаточно, поэтому было принято решение о смене микропроцессора на ATmega328P (32Kб).

2) Отсутствовала готовая библиотека под AVR для контроллера ST7567S дисплея по протоколу I²C. Была разработана собственная низкоуровневая библиотека [3][4]. При этом использовались осциллограф и анализатор сигналов.

Так же при анализе сигнала были замечены завалы на фронтах, что могло сказаться на качестве передачи сигнала. Было принято решение разогнать микропроцессор при помощи внешнего кристалла до 16МГц, что так же дало увеличение скорости отрисовки информации на дисплее.

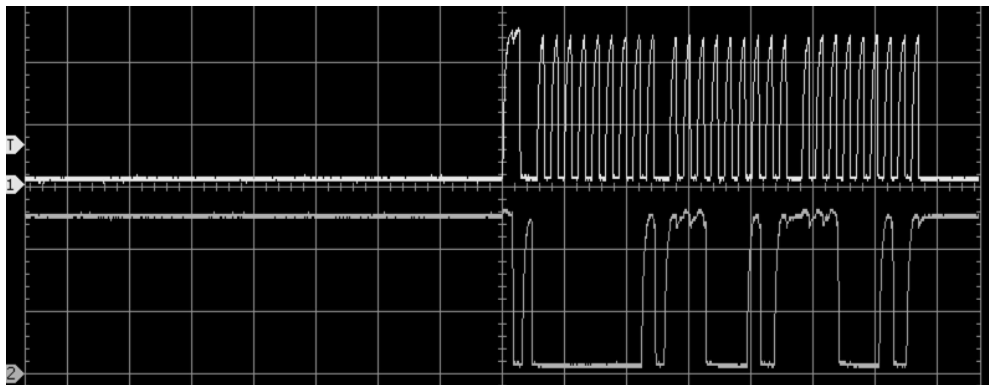


Рисунок 1 – Передача 3 байт по I²C с осциллографа



Рисунок 2 – Передача команды 0xE3 на дисплей по I²C

3) Зависание шины контроллера из-за некорректной отправки сообщения и бесконечного цикла ожидания ответа дисплея I²C. Проблема была решена использованием 8-битного таймера/счётчика 0, который запускается во время отправки данных и следит за временем. В случае ошибки – перезапуск I²C.

4) Проблема с дребезгом контактов у энкодера, а также зависание контроллера. Первая проблема была решена добавлением в схему конденсаторов, вторая – переходом на обработку сигналов через прерывания [3].

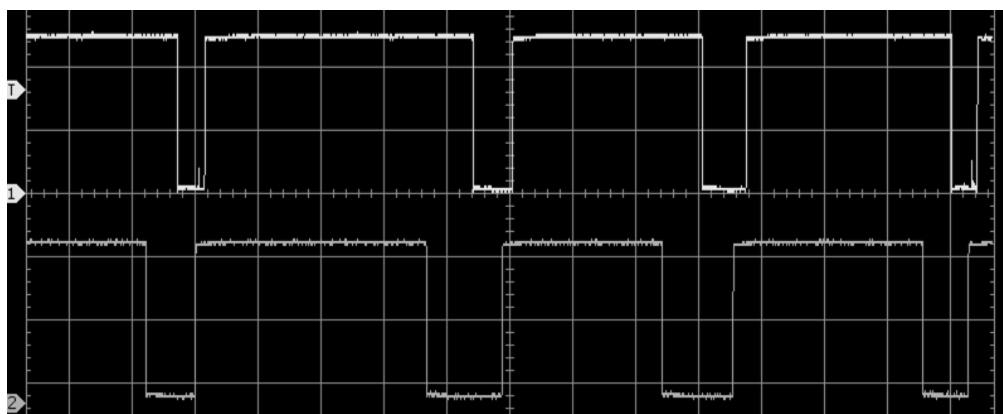


Рисунок 3 – Поворот ручки энкодера против часовой стрелки

5) Датчик усилия не имел готовой библиотеки и необходимо было учесть, как именно предаются данные [2] – есть положительные и отрицательные значения в унциях. Подобрали для значения тип данных long int, придумали систему обнуления веса и перевода из унций в килограммы.

6) Так как микропроцессор не может измерять силу тока, а только напряжение, то была реализована схема с применением закона Ома.

$$I_{\text{нагрузки}} = R_{\text{шунта}} * U_{\text{падения}}$$

где $U_{\text{падения}}$ определяется как разность напряжений на концах шунта.

$$U_{\text{падения}} = U_{\text{до_шунта}} - U_{\text{после_шунта}}$$

Однако купленный шунт на 0,0075 Ом пришлось заменить в связи с слишком маленьким сопротивлением на резистор 0,1 Ом.

7) Корпус с электроникой пришлось разбить на две части: сервотестер и модуль для тестирования усилия. Это связано с невозможностью разместить две сервомашинки и рельсу одновременно для замера усилия прижимной силы двигателя на основном корпусе.

Закключение. По итогу исследования был разработан сервотестер и низкоуровневые библиотеки для взаимодействия с компонентами полученной системы.

Литература:

1. Романцов, Д.Ю. Микроконтроллеры AVR. / Д. Ю. Романцов // Управление сервоприводом. [Электронный ресурс], 2015 – 45 с.
2. Шпак, Ю. А. Программирование на языке C для AVR и PIC микроконтроллеров. Изд. 2-е, переработанное и дополненное / Ю.А. Шпак – К.: «МК-Пресс», СПб.: «КОРОНА-ВЕК», 2011. – 544 с.

3. Atmel Corporation. Atmel AVR 8-bit with 8KBytes In-System Programmable Flash [Электронный ресурс] / www.promelec.ru ATmega8 Datasheet //Режим доступа: <https://www.promelec.ru/fs/sources/e4/09/f3/32/e79e1c9c5417d1d11d1a7cc7.pdf> – Дата доступа: 14.02.2024.

4. Sitronix Technology Corporation. ST7567S 132X65 Dot Matrix LCD Controller Datasheet [Электронный ресурс] / buydisplay.com Sitronix ST7567S Controller Datasheet //Режим доступа: <https://www.buydisplay.com/download/ic/ST7567S.pdf> – Дата доступа: 22.02.2024.

ПРИВЕДИТЕ В ПОРЯДОК ЛИЧНЫЕ ФИНАНСЫ И СЕМЕЙНЫЙ БЮДЖЕТ С МОБИЛЬНЫМ ПРИЛОЖЕНИЕМ «HOME ACCOUNTING»: РАЗРАБОТКА ДОМАШНЕЙ БУХГАЛТЕРИИ

Михеева Е.А.,

обучающаяся ГАПОУ ТО «Колледж цифровых и педагогических технологий»,

г. Тюмень, Российская Федерация

Научный руководитель – Кучина И.А., преподаватель

Практически каждый человек сталкивается с нехваткой денег на повседневные нужды или же с тем, что доходов хватает ровно на жизнь, и откладывать на что-то важное и нужное средств не хватает. Именно поэтому важно контролировать свои финансы и семейный бюджет.

В процессе моей работы над проектом я выяснила, что людям тяжело приучить себя контролировать свой или семейный бюджет. Это связано преимущественно с нехваткой времени и не удобством.

Проект по разработке домашней бухгалтерии для ОС Android может стать отличным решением для людей, находящихся в постоянном движении и желающих вести учёт личных финансов. Основным достоинством итогового приложения будет возможность контроля финансов на ходу. Учёт доходов и расходов позволит вам: узнать свои реальные доходы и найти способы их приумножения, начать планировать бюджет, понять, куда утекают деньги, начать делать накопления, избавиться от кредитов.

Исследование связано с решением следующих проблем: нехватка финансов на полноценную счастливую жизнь, трудность ведения личного бюджета и невозможность делать накопления, нехватка времени на учёт доходов и расходов, недостаток знаний и мотивации у людей для контроля финансов. *Цель исследования* заключается в решении вышеперечисленных проблем за счёт разработки мобильного приложения «Home Accounting» для ведения домашней бухгалтерии.

Задачи:

- провести анализ предметной области;
- изучить современный стек технологий;
- провести анализ существующих аналогов приложений;
- разработка требований к приложению;
- разработать интерфейс;
- разработка приложения домашней бухгалтерии.

Функционал итогового приложения: возможность хранения данных о доходах и расходах, расчёт остатков на счетах, анализ финансов за определённый период, просмотр истории транзакций с возможностью фильтрации по месяцу и виду транзакции.