

Такие приборные панели встречаются в автомобилях класса люкс. Цифровая приборная панель представляется в виде кластера цифровых инструментов, которая заменит или дополнит механические приборы виртуальными, отображаемыми на жидкокристаллических дисплеях, и управляется микропроцессором. Вследствие этого появляется возможность полной настройки интерфейса, расширения штатных возможностей.

Разработка цифровой приборной панели Witronic обусловлена ее актуальностью и практической значимостью. Автовладельцы получают возможность иметь приборную панель с полностью настраиваемым интерфейсом под свои требования, дополнять ее различными системами начиная от системы контроля давления в шинах и навигации и заканчивая системами помощи водителю на дороге.

Структура аппаратной части приборной панели выглядит следующим образом: сигналы от датчиков автомобиля поступают в блок Witronic, который обработав их упаковывает в пакет данных и передает его в бортовой компьютер, в качестве которого используется Raspberry PI. На стороне бортового компьютера формируется графическая часть приборной панели, далее изображение передается через HDMI-кабель на драйвер дисплея, который через интерфейс LVDS передает данное изображение на сам дисплей.

Блок Witronic построен на базе микроконтроллера фирмы ATMEL. Он имеет 2 основных разъема с 48 выводами для подключения к автомобилю, а также 2 разъема для подключения датчиков наружной и внутренней температуры автомобиля, 1 USB разъем с управляемым питанием +5 В 3 А. Основные разъемы позволяют подключить 7 аналоговых датчиков автомобиля (+ 1 канал подключен в схеме платы на вход питания, для замера напряжения бортовой сети), 34 цифровых канала для подключения к лампам, тахометру, спидометру и т.п. автомобиля, из которых 1 канал управляет яркостью подсветки экрана ШИМ-сигналом, 1 канал служит для подключения спидометра и 1 канал для подключения тахометра. Блок представлен на рисунке 1.

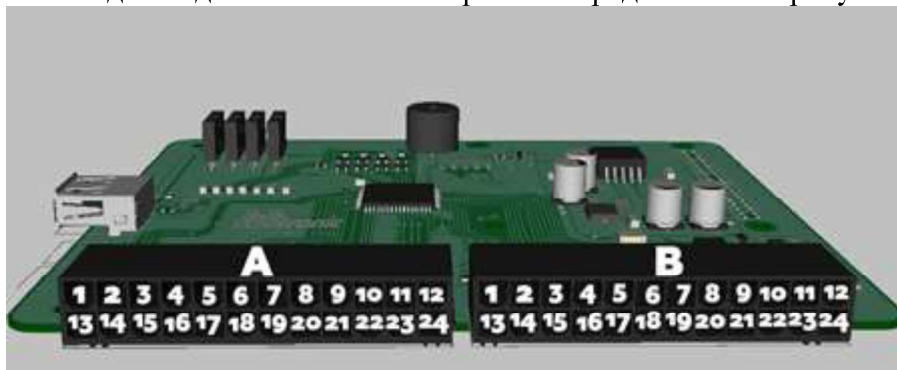


Рис.1 – Блок Witronic

Также на одном из разъемов имеются 2 управляемых силовых выхода с замыканием цепи на землю, 1 выход 12 В от подачи зажигания, 1 вход зажигания, 1 вход общего питания и общий провод GND.

В блоке Witronic предусмотрен управляемый вывод +12 В, который в приборной панели управляет питанием экрана (драйвер + подсветка).

Для чёткого восприятия аварийных режимов (давление масла, напряжение бортовой сети) водителем, помимо графического отображения этих режимов, в блоке установлен бuzzer.

Разработка программного обеспечения приборной панели осуществлена с использованием языков программирования QML[2] и C++[3, 4].

QML является декларативным языком программирования, основанным на JavaScript и предназначен для дизайна приложений делающих основной упор на пользовательский интерфейс, в связи с этим он был выбран для разработки главного окна приложения и окон настроек. QML-документ представляет собой дерево элементов, каждый элемент представляет собой совокупность блоков: графических (таких как прямоугольник, изображение) и поведенческих (таких как состояние, переходы, анимация). По этому принципу и строиться внешний вид приборной панели.

Интерфейс панели приведен на рисунке 2.



Рис.2-Интерфейс панели

Обработка поступающих сигналов с датчиков автомобиля осуществляется кодом реализованном на C++. Язык программирования C++ был использован для написания прошивки блока Witronic и программного кода, выполняющегося на стороне бортового компьютера и отвечающего за прием данных по сетям коммуникаций и их дальнейшую обработку.

На данный момент реализованы все необходимые графические компоненты для отображения показаний датчиков:

- тахометр;
- указатель температуры охлаждающей жидкости;
- уровень топлива в баке автомобиля;
- контрольные лампы (STOP, ремни безопасности, CHECK и пр.);
- спидометр;
- указатели поворота;
- показатели температур снаружи и внутри автомобиля.

Разработан прототип цифровой приборной панели Witronic. Приборная панель построена на базе блока сбора данных Witronic и бортового компьютера Raspberry PI. Связь между, которыми осуществляется посредством UART или Wi-Fi сети.

На данный момент приборной панели Witronic установлена на автомобиле VW Golf, где проходят тесты и отладка программно-аппаратного комплекса панели.

#### Список использованных источников

1. Секунов, Н. Программирование на C++ в Linux / Н. Секунов. - М.: БХВ-Петербург, 2016. - 425 с.
2. Полубенцева, М. C/C++. Процедурное программирование / М. Полубенцева. - М.: БХВ-Петербург, 2014. - 448 с.
3. Э. Грик. Цифровые приборные панели для автомобилей: техническое решение и рыночные преимущества. [Электронный ресурс] -URL: <http://www.swd.ru/index.php3?pid=899> дата обращения (18.10.18)

4. A Book about Qt 5.[Электронный ресурс] – URL : <https://qmlbook.github.io/> дата обращения (18.10.18)

27.37.17

## **ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ ДЛЯ ЗАДАЧ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ**

**Автор:** Карпов Максим Александрович, студент 3 курса направления «Информатика и вычислительная техника» филиала «Протвино» государственного университета «Дубна»

**Научный руководитель:** Нурматова Елена Вячеславовна, к.т.н., доцент кафедры информационных технологий филиала «Протвино» государственного университета «Дубна»

### **Аннотация**

В данной работе рассматривается решение параллельных задач не линейного программирования при линейных ограничениях, как с помощью параллельных вычислений можно ускорить решение поставленной задачи.

In this paper, we consider the solution of parallel problems of non-linear programming with linear constraints, as with the help of parallel calculations, the solution of the problem can be accelerated.

В наше время практически нет такой сферы деятельности, где бы в той или иной форме не использовалось численное моделирование. Современные технологии, относящиеся к областям науки, очень часто обращаются к численным методам. Усложняющиеся технологические задачи, в поисках новых знаний, привели к пониманию, что эмпирический путь решения сложных проблем давно себя исчерпал. Использование накопленного многими поколениями запаса знаний должно быть направлено на построение математических моделей, достаточно адекватно отражающих суть изучаемых явлений или объектов. Как правило, математическая модель представляет собой некоторый комплекс дифференциальных, интегральных, алгебраических или других соотношений, связанных с предметом исследований.

Параллельные вычисления – вычисления, которые можно реализовать на многопроцессорных структурах с выполнением многих действий одновременно, которые, в свою очередь, могут породить процессорные решения одной или многих задач одного проекта.

Основная цель параллельных вычислений – сокращение времени решения задачи. Для решения многих задач в реальном времени требуется большой объём вычислений. Большое количество числа процессоров не всегда приводит к снижению затрат времени для решения задачи.

Задача параллельных вычислений – нахождения параллельного алгоритма в процессах решения задач и управление работой этого параллелизма с целью получения наибольшей эффективности применяя многопроцессорную вычислительную технику. Параллельный алгоритм решения задачи можно найти путём разработки нового параллельного алгоритма или распределением имеющегося последовательного алгоритма. Алгоритм решения задачи необходимо модифицировать для осуществления распараллеливания (например, устранить некоторые зависимости между операциями).

Параллельная обработка данных – это не единственный путь прироста скорости вычислений. Другой подход – повышать мощность процессорных устройств. Ограничениями такого подхода являются:

1. *Лимитированность скорости переключения.* Даже при самых быстрых коммуникациях скорость переключения не может превышать скорость света.

2. *Ограниченность размеров переключателей.* Устройство может работать быстрее с уменьшением его в размерах. Существует физический предел на размер компонентов, что связано с их молекулярным и атомным строением.

3. *Экономические ограничения.* Для увеличения скорости процессора, числа слоев в кристалле, плотности упаковки приходится решать все более сложные научные, производственные проблемы. Вот почему каждое новое поколение процессоров дорого стоит.

При создании параллельного алгоритма очень важно знать тип оперативной памяти, она определяет способ взаимосвязи между частями параллельной программы. В зависимости от организации подсистем оперативной памяти параллельные компьютеры можно разделить на следующие два класса.

*Системы с распределенной памятью* (мультикомпьютеры). Каждый процессор резервирует свою локальную оперативную память, а другие процессоры в это время не могут подключиться к этой памяти.

*Системы с разделяемой памятью* (мультипроцессоры). Они имеют одну виртуальную память, а все процессоры имеют равный доступ к данным и командам, держащийся в этой памяти (uniform memory access или UMA). По этому принципу разрабатываются симметричные мультипроцессоры (symmetric multiprocessor или SMP) и векторные параллельные процессоры (parallel vector processor или PVP).

Рассмотрим пример решения параллельных задач не линейного программирования при линейных ограничениях.

Пусть решается задача

$$f(x_1, \dots, x_n) \rightarrow \max$$

При ограничениях

$$g_1(x_1, \dots, x_n) \geq 0$$

$$g_2(x_1, \dots, x_n) \geq 0$$

...

$$g_m(x_1, \dots, x_n) \geq 0$$

и при условиях:  $g_i$  – линейны,  $x_i \geq 0, i = 1, \dots, m$ .

Представим, что гиперплоскости  $g_i, i = 1, \dots, m$ , возможно, совместно с координатными плоскостями, образуют в  $n$ -мерном пространстве выпуклый многогранник  $R$  допустимых решений, т.е. область задания функции  $f$ .

Обозначим все вершины  $\{X_1, \dots, X_N\}, X_l = (x_1^l, \dots, x_n^l), l = 1, \dots, N$ , этого многогранника в результате решения  $C_{m+n}^n$  систем  $n$  линейных уравнений на основе всех заданных и возможных его граней

$$g_1 = 0$$

...

$$g_m = 0$$

$$x_1 = 0$$

...

$$x_m = 0.$$

Тогда множество точек  $X = (x_1, x_2, \dots, x_n)$  этого многогранника описывается как  $X = \sum_{i=1}^N k_i X_i$ , где  $0 \leq k_i \leq 1, \sum_{i=1}^N k_i = 1$ . Или: