

$$x_{\text{hor}} = x \cdot \cos(90 - \text{lat}) - z \cdot \sin(90 - \text{lat}); \quad (22)$$

$$y_{\text{hor}} = y; \quad (23)$$

$$z_{\text{hor}} = x \cdot \sin(90 - \text{lat}) + z \cdot \cos(90 - \text{lat}), \quad (24)$$

где lat – широта.

Для получения азимута и высоты солнца, преобразуем его в сферические координаты ($r = 1$):

$$\text{azimuth} = \arctg \frac{y_{\text{hor}}}{x_{\text{hor}}}; \quad (25)$$

$$\text{altitude} = \arcsin(z_{\text{hor}}) = \arctg \frac{z_{\text{hor}}}{\sqrt{x_{\text{hor}}^2 + y_{\text{hor}}^2}}. \quad (26)$$

На основе вычислений высоты и азимута солнца не составляет проблем определить оптимальное положение солнечных панелей. Оптимальным положением является то, при котором солнечные лучи перпендикулярны фотоэлектрическому модулю.

Для управления работой ориентации фотоэлектрической станции была написана программа для Arduino Uno, которая регулирует положение фотоэлектрических модулей в зависимости от положения Солнца. Для работы программы необходимо указать долготу и широту, а также подключить и настроить часы реального времени. Программный модуль разрабатывался в среде программирования Microsoft Visual Studio на языке C++.

Литература

1. Meeus, J. *Astronomical Algorithms* / Jean Meeus. – Richmond: Willmann-Bell, 1998. – 477 p. – ISBN 978-0943396613.

П. К. Банза

(ГГТУ имени П. О. Сухого)

Науч. рук. **В. В. Комраков**, канд. техн. наук, доцент

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ДЛЯ УПРАВЛЕНИЯ РАБОТОЙ МОДУЛЯ ВРАЩАТЕЛЬНОГО ДВИЖЕНИЯ МАШИНЫ ТРЕНИЯ

Известно, что для повышения эффективности какого-либо вида деятельности человека, необходима его автоматизация. При этом

очень важно проводить автоматизацию не только опасной для человека деятельности, но также рутинных операций, например проведения трибологических исследований различных материалов.

В работе [1] проведен анализ существующих схем трибологических испытаний [2]. Были выявлены элементарные вращательное и поступательное движения.

На основании этого в работе [1] был предложен модульный принцип построения машины трения. С этой целью было предложено создать два типа модулей:

1. модуль вращательного движения;
2. модуль поступательного движения.

В данной работе рассматривается программное обеспечение для управления работой модуля вращательного движения машины трения, который состоит из шагового двигателя (NEMA17), на вал которого установлен образец цилиндрической формы. Один или два модуля вращательного движения могут реализовать различные схемы трибологических испытаний.

В качестве аппаратной платформы для управления модулем применялась плата Arduino 2560 с платой расширения Ramps 1.4 с установкой на последней драйверов шаговых двигателей DRV8825 и подключением дисплея LCD 12864. Схема подключения приведена на рисунке 1.

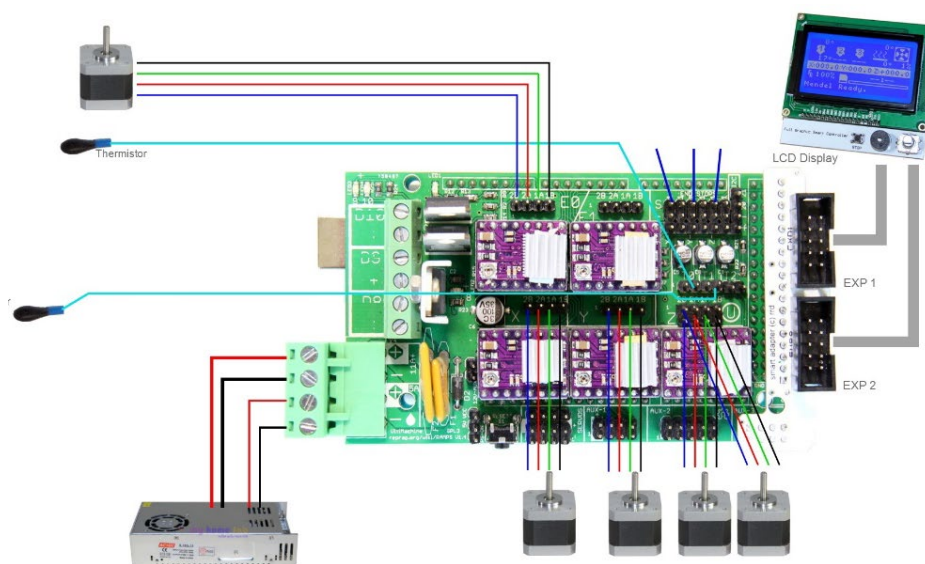


Рисунок 1 – Схема подключения

Для управления шаговым двигателем в Arduino применялась стандартная библиотека Stepper. С ее помощью устанавливается

скорость вращения шагового двигателя, а также устанавливается число шагов. Это позволяет вращаться двигателю определенное число оборотов на заранее установленной скорости.

Для работы с дисплеем LCD 12864 применяли библиотеку U8glib. При этом на дисплей выводится текстовая информация об установленных параметрах вращения шаговых двигателей.

Плата Arduino 2560 подключается к порту USB компьютера.

Для управления работой двух модулей вращательного движения было создано программное обеспечение на языке высокого уровня Python. Средой разработки являлась PyCharm Community IDE.

Программное обеспечение имеет графический интерфейс, который реализован с помощью библиотеки PyQt5. Для разработки шаблона интерфейса применялась среда Qt Designer. На данный момент программное обеспечение обеспечивает ввод частоты вращения двух двигателей и время их работы, а также передает эти данные в Arduino 2560 через Serial-порт. Для работы с этим портом необходима установка в PyCharm библиотеки PySerial. Программное обеспечение дополнительно обеспечивает сканирование имеющихся в системе Serial-портов с возможностью выбора одного из них для подключения и установкой скорости передачи данных.

Для получения данных платой Arduino 2560 применяется набор функций Serial. Для передачи данных используются цифровые порты ввода/вывода 0 и 1, а также USB-порт.

В качестве примера на рисунке 2 приведена реализации схемы «два вращающихся вала». При этом два модуля вращательного движения установлены на кронштейнах и неподвижно закреплены при помощи винтов.

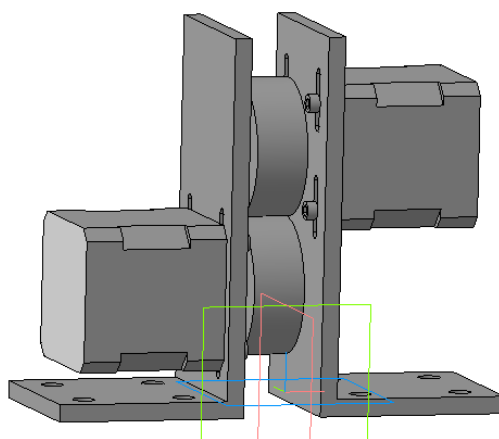


Рисунок 2 – Схема установки для реализации схемы испытаний «два вращающихся вала»

Модульный принцип построения машины трения позволит провести реализацию трибологических испытаний не только по рассмотренным выше схемам трения, но и добавить новые элементы, реализующие другие схемы трения, а также проводить дальнейшие работы по автоматизации машины трения.

Литература

1. Банза П. К. Роботехнический комплекс для проведения трибологических испытаний // Современные проблемы математики и вычислительной техники: сборник материалов XI Республиканской научной конференции молодых учёных и студентов. – 2019. – С. 7–9.
2. Технические характеристики. Испытательная машина 2070 СМТ-1. 2012 г. – Электрон. данные. URL: <https://zapadpribor.com/2070-smt-1/>. – Дата обращения 04.08.2020.

М. С. Березовский

(ГГУ имени Ф. Скорины, Гомель)

Науч. рук. **М. И. Жадан**, канд. физ.-мат. наук, доцент

О ТЕСТИРОВАНИИ РАЗРАБОТАННОГО ИГРОВОГО ПРИЛОЖЕНИЯ «SPACESHIP»

В современном программировании на равном уровне с разработкой программного продукта стоит его качество. Хорошей практикой для обеспечения качества разработанного продукта является создание сценариев различного рода использования приложения, а также покрытие исходного кода программы тестами.

Для покрытия исходного кода разработанного игрового приложения «Spaceship» был создан отдельный проект, предназначенный для хранения и запуска тестов. В данном проекте были созданы блочные тесты. Основное назначение блочных тестов – тестирование каждого блока (модуля) исходного кода программы вне зависимости от других модулей. Для корректного функционирования данных тестов был использован фреймворк, позволяющий предоставлять «поддельные» данные модулю, тестирование которого производится вместо данных, которые должен был предоставлять другой модуль.