

Примеры тезисов научных работ победителей Всероссийского конкурса научных работ школьников Юниор по инженерным наукам 2016-2017 учебный год

Секция «Информатика»

Областное государственное казенное учреждение дополнительного образования «Костромской областной центр научно-технического творчества «Истоки»

156000, Костромская область, г. Кострома, ул. 1 мая, д. 4/9;

тел.: (4942)31-10-94; e-mail: techno302@mail.ru

Группа электронных ударных инструментов на базе AVR микроконтроллеров

Евсеев Илья, Челноков Данил

Класс: 10, 9

156016, г. Кострома, ул. Профсоюзная, д. 30, кв. 169;

тел.: +7(953)646-56-91; e-mail: Evseev.ilia-44@yandex.ru

156002, г. Кострома, ул. Депутатская, д. 62, кв. 4, ком. 1;

тел.: +7(953)656-87-05; e-mail: Chelnokov.dania@yandex.ru

Научный руководитель: Шестаков Александр Александрович, Заслуженный рационализатор Костромской области, педагог дополнительного образования

1. Цель работы: решение проблемы отсутствия своей ударной установки в вокально-инструментальном ансамбле КОЦНТТ «Истоки» для репетиционных занятий и выступлений.

2. Методы исследования:

1. Теоретический метод – по специальной литературе, а также материалам сети Internet проведен анализ имеющегося в продаже оборудования для решения этой проблемы, а также рассмотрена возможность создания своего оборудования, которое можно было бы собрать в условиях объединения радиотехнического конструирования «Электроник» и которое бы отвечало требованиям, запрашиваемым ВИА.

2. Экспериментальный метод - были проведены эксперименты по использованию в барабанной установке различных датчиков вибрации и схем для снятия с них данных.

3. Практический метод исследования:

- была спроектирована и собрана электронная барабанная установка с подключением к персональному компьютеру;

- было написано специализированное программное обеспечение для барабанной установки.

4. Проектный метод – спроектировано несколько вариантов несущих конструкций и стоек для электронных барабанов с возможностью смены мест их расположения.

3. Ход работы над проектом.

Мы приняли решение собирать свою электронную барабанную установку с использованием AVR микроконтроллеров, а именно их аналого-цифровых преобразователей с подключением к персональному компьютеру, т.к. при таком варианте построения барабанной установки используется минимальное количество деталей, а соответственно на ее сборку нужно минимум средств. В качестве датчиков удара в установке мы решили применить имеющиеся у нас в большом количестве советские мембранные наушники. При ударах по мембране на выводах катушки наводится мизерное напряжение. Уровень этого напряжения зависит от силы удара, что и требуется для электронной барабанной установки, чтобы реализовать зависимость громкости и продолжительности воспроизводимого звукового сигнала от силы удара по «барабану». Так как напряжение, генерируемое катушкой, слишком мало, было решено применить стандартную усилительную схему. На выходе усилителя мы имеем аналоговый сигнал, изменяющийся от 0 до 3 вольт в зависимости от силы удара по мембране. Аналоговый сигнал сложно обрабатывать с помощью компьютера, поэтому мы приняли решение преобразовать его в цифровой. Для решения этой задачи идеально подходят AVR микроконтроллеры, имеющие в своем составе АЦП. Мы выбрали микроконтроллер ATtiny45. Один контроллер обрабатывает сигналы с двух датчиков (барабанов). Для обработки аналогового сигнала с помощью АЦП микроконтроллера ATtiny45 потребовалось написание прошивки, она написана на языке «C» и работает следующим образом: если на входах двух аналого-цифровых преобразователей напряжение отсутствует, то с выхода микроконтроллера в USB порт компьютера в виде битов посылаются два ноля, каждый из которых соответствует одному из АЦП микроконтроллера. Если напряжение на входе АЦП изменяется, то изменяется и число, отправляемое микроконтроллером на USB порт компьютера. В дальнейшем эти числа обрабатываются программой на компьютере, которая также была написана в процессе работы над проектом на языке «C#».

Эта программа работает следующим образом: из программы библиотеке USB Score передаются идентификаторы устройства. Она считывает данные, приходящие с микроконтроллера, соответствующего переданным идентификаторам и пересылает эти данные в программу. Когда в программу приходит число, отличное от нуля, она запоминает это число. Если дальше приходит большее число, она снова его запоминает, и так до тех пор, пока пришедшее число не будет меньше запомненного. Тогда программа при помощи модуля DirectSound, входящего в состав DirectX, воспроизводит звук (файл расширения .wav), соответствующий нужному барабану, с громкостью, соответствующей запомненному числу. Этот механизм нужен для того, чтобы выбрать максимальное значение, приходящее на АЦП вследствие удара, т.к. по закону самоиндукции напряжение, соответствующее силе удара, выдается датчиком не мгновенно. Нужный барабан выбирается в соответствии с номером микроконтроллера и номером АЦП, напряжение с которого учитывается при выборе громкости воспроизведения. В окне программы имеются: 2 числа, приходящие с микроконтроллера и показывающие напряжения на двух АЦП микроконтроллера (в милливольтках); графики напряжений на двух АЦП микроконтроллера.

Посоветовавшись с творческим коллективом ВИА, мы решили, что наша электронная барабанная установка будет укомплектована семью барабанами: 1.Малый («рабочий») барабан; 2.Бас-барабан («бочка»); 3.Бас-том; 4.Том №1; 5.Том №2; 6.Тарелка Hi-hat; 7.Тарелка Crash. При этом для изменения типа барабана требуется всего лишь замена звукового файла (.wav) в программе. Для работы семи барабанов потребовалось

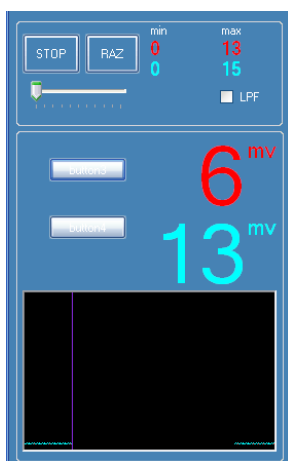
собрать семь усилительных схем и четыре схемы с микроконтроллерами. Кроме этого для работы всей электронной барабанной установки, состоящей из семи барабанов, требуется одновременный запуск четырех программ, причем, по крайней мере три окна должны воспроизводить звук даже тогда, когда они неактивны; это реализовано при помощи установки флага GlobalFocus в дескрипторе звукового буфера.

4. Результаты работы.

Барабаны на стойках имеют такое же расположение, как и на акустических ударных установках, а громкость воспроизведения регулируется с компьютера, что позволяет репетировать в наушниках, не мешая окружающим, а на выступлениях, при подключении активных звуковых колонок, использовать установку на полную мощность, в точности повторяя звучание реальных барабанов.

5. Используемые источники информации:

1. Белов А. В. – «Самоучитель разработчика устройств на микроконтроллерах AVR»
2. Том Миллер – «Managed DirectX – Программирование графики и игр»
3. <http://msdn.microsoft.com/ru-ru/library/kx37x362.aspx>
4. <http://www.mirmk.net/content/view/88/29/>



Примеры тезисов научных работ победителей Всероссийского конкурса научных работ школьников Юниор по инженерным наукам 2016-2017 учебный год

Секция «Математика»

Секция: математика

ГБОУ Лицей «Вторая школа», г. Москва

119333, г. Москва, ул. Фотиевой, д.18

Тел.: (499)-137-17-69; E-mail: lycuz2@edu.mos.ru

Обобщенная задача Аполлония

Морозов Егор

Класс: 11

117342, г. Москва, ул. Бутлерова, д. 22, кв. 65

Тел.: (985)-181-28-82; E-mail: gorg.morozov@gmail.com

Научный руководитель: Нилов Фёдор Константинович, преподаватель-методист, центр онлайн-обучения «Фоксфорд»

Текст тезисов

Обобщенная окружность – окружность, прямая или точка на плоскости.

Наша цель – обобщить исследование следующей геометрической задачи на построение **(Классическая) задача Аполлония (III в. до н. э.)**. *На плоскости даны 3 различные обобщенные окружности, не все касающиеся в одной точке. Построить циркулем и линейкой обобщенную окружность, касающуюся каждой из данных.*

на случай большего числа исходных окружностей. Известно, что задача Аполлония имеет не более 8-ми решений (см. [2, 4]). В связи с этим, заслуживает внимания следующая задача.

Обобщенная задача Аполлония. *На плоскости даны несколько (>3) различных обобщенных окружностей, не все они касаются в одной точке. Найти максимальное возможное число обобщенных окружностей, касающихся каждой из данных.*

Здесь и далее данные в (обобщенной) задаче Аполлония обобщенные окружности называются *исходными*, а обобщенная окружность, касающаяся каждой из исходных – *решением*.

Поставленная задача интересна прежде всего тем, что её формулировка представляется одним из наиболее естественных обобщений задачи Аполлония. Тем не менее основные результаты работы, по-видимому, являются новыми (см. [1, стр. 3]).

Все методы решения задачи не выходят за рамки углубленной школьной программы. Основную роль играет инверсия, которая позволяет уменьшить изначально весьма громоздкий перебор. Из преобразований плоскости также иногда применяются некоторые движения, например, симметрия и поворот.

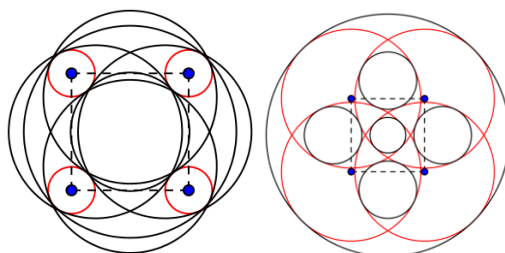
Используются и соображения о самих окружностях различного характера: как совсем простые, например, что две различные окружности имеют не более двух общих точек, так и, скажем, соображение о том, что решения классической задачи Аполлония разбиваются на пары, такие, что окружности в каждой паре инверсны относительно радикального центра исходных окружностей (см. [4, разд. «Pairs of solutions by inversion»]).

Иногда приходится использовать некоторые результаты полного перебора всех случаев классической задачи Аполлония (см. [2]). Впрочем, используемые результаты тоже достаточно известны. Полученные в процессе исследования результаты следующие.

Теорема 1. *На плоскости даны 4 различные обобщенные окружности, не все касающиеся в одной точке. Тогда существует не более 6-и обобщенных окружностей, касающихся каждой из данных.*

Теорема 2. *На плоскости дано 5 различных обобщенных окружностей, которые не все касаются в одной точке. Тогда существует не более 4-х обобщенных окружностей, касающихся каждой из данных.*

Доказательство теоремы 1 можно найти в [1]. Ведется аналогичная работа по изложению доказательства теоремы 2.



Примеры четверок окружностей с 6-ю решениями

Заметим, что существуют четверки окружностей с 6-ю решениями (см. рис.), из которых, очевидно, можно получить примеры пятерок окружностей с 4-я решениями. Поэтому оценки теорем 1 и 2 точные.

Заметим также, что данные результаты дают полное решение поставленной задачи, т. е. дальнейшее увеличение числа исходных окружностей не приведет к новым результатам. Пусть, например, дано 6 обобщенных окружностей. Если бы для них существовало более 4-х решений, то, рассмотрев решения в качестве исходных окружностей, мы получили бы 5 обобщенных окружностей, для которых существует 6 решений – противоречие с теоремой 2.

Литература

- [1]: E. Morozov, «Generalized problem of Apollonius», обновляемая версия: arxiv.org/abs/1611.03090
- [2]: Bruen A, Fisher JC, Wilker JB (1983). "Apollonius by Inversion". *Mathematics Magazine* **56** (2): 97–103
- [3]: Р. Курант, Г. Роббинс «Что такое математика?», М.: Издательство Московского центра непрерывного математического образования, 2010.
- [4]: https://en.wikipedia.org/wiki/Problem_of_Apollonius

Примеры тезисов научных работ победителей Всероссийского конкурса научных работ школьников Юниор по инженерным наукам 2016-2017 учебный год

Секция «Робототехника»

Целесообразность использования поворотных систем слежения за солнцем для сбора солнечной энергии.

Цель работы

Определить эффективность использования поворотной системы, ориентирующей солнечные панели относительно положения солнца (солнечного трекера).

Описание работы

По данным Росбалт уже в 30 странах возобновляемая энергия стала дешевле нефти и газа (<http://www.rosbalt.ru/business/2016/12/27/1579260.html>). В них вошли Австралия, Мексика, Бразилия и Чили. Другие страны в отчете ВЭФ не указаны. Однако большинство этих стран находятся ближе к экватору, где солнце светит 12 часов в сутки, и собранного количества энергии хватает для жизнедеятельности потребителей. Но как получить такое же количество энергии в странах, находящихся дальше от экватора, например, у нас в России? Для ответа на вопрос было просмотрено много информации. Существует большое количество систем слежения за солнцем. Одним из вариантов является использование солнечных трекеров на микроконтроллерах. На сколько энергии получится больше, используя солнечный трекер? Проведем эксперимент.

Для эксперимента был собран на основе Arduino nano солнечный трекер (<https://www.arduino.cc/en/Main/ArduinoBoardNano>) (рисунки 1 и 2)

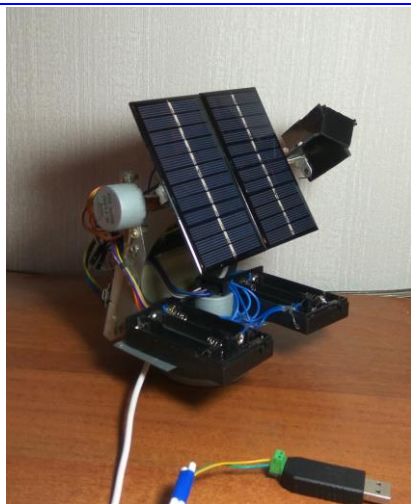


Рисунок 1. Вид спереди

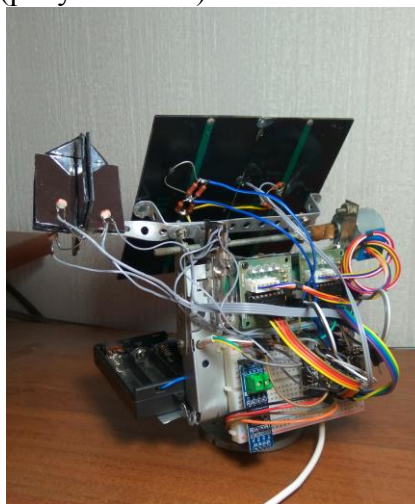


Рисунок 2. Вид сзади

В тех вариантах солнечных трекеров, которые были найдены в интернете, используются сервоприводы. Было решено использовать дешевые шаговые двигатели (рисунки 3 и 4), но в режиме покоя для уменьшения потребления электроэнергии удерживающий ток подаваться на их обмотки не будет, а удержание будет происходить за счет редукторов самих двигателей.



Рисунок 3. Используемый двигатель

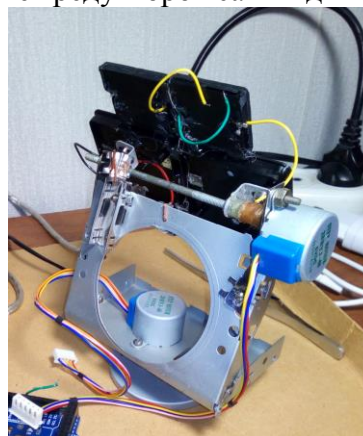


Рисунок 4. Расположение двигателей

Для этого в библиотеку stepper.h мы внедрим новый метод, который позволит удерживать двигатели не за счет тока, а только за счет редукторов. (рисунок 5)


```

// pin_count is used by the stepMotor() method:
this->pin_count = 5;
}

/*
 * Sets the speed in revs per minute
 */
void Stepper::setSpeed(long whatSpeed)
{
  this->step_delay = 60L * 1000L * 1000L / this->
number_of_steps / whatSpeed;
}

/*
 * no power/hold in standBy mode
 */
void Stepper::noHold(void)
{
  digitalWrite(motor_pin_1, LOW);
  digitalWrite(motor_pin_2, LOW);
  digitalWrite(motor_pin_3, LOW);
  digitalWrite(motor_pin_4, LOW);
  digitalWrite(motor_pin_5, LOW);
}

/*
 * Moves the motor steps_to_move steps. If the number is
negative,
 * the motor moves in the reverse direction.
 */
void Stepper::step(int steps_to_move)
{
  int steps_left = abs(steps_to_move); // how many steps to
take

  // determine direction based on whether steps_to_mode is +
or -:
  if (steps_to_move > 0) { this->direction = 1; }
  if (steps_to_move < 0) { this->direction = 0; }
}

```

Рисунок 5. Модификация библиотеки stepper.h

Движение двигателей по горизонтали не имеет ограничений (рисунок 6), а по вертикали ограничено двумя концевыми выключателями (рисунок 7)

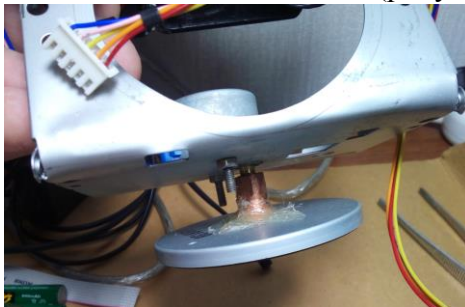


Рисунок 6. Горизонтальное движение

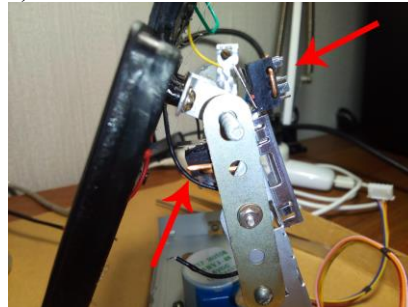


Рисунок 7. Вертикальное движение и его ограничение

Для Arduino была написана программа в Arduino IDE 1.6.13 (рисунок 8)

```

//подключаем библиотеки
#include <SimpleModbusSlave.h>
#include <Stepper.h>
Stepper _StM1(32, 6, 4, 5, 3); //горизонтальное движение
Stepper _StM2(32, 10, 8, 9, 7); //вертикальное движение
enum
{
    HOLDING_REGS_SIZE = 20 //количество каналов
};
unsigned int holdingRegs[HOLDING_REGS_SIZE];
int a0, a1, a2, a3, a4, a5, a6, a7, delta = 20;
bool topSwitch, bottomSwitch, man, left, right, up, down;

void setup()
{
    modbus_configure(&Serial, 9600, SERIAL_8N2, 1, 2, HOLDING_REGS_SIZE, holdingRegs); //вводим скорость опроса
    modbus_update_comms(9600, SERIAL_8N2, 1);
    _StM1.setSpeed(200);
    _StM2.setSpeed(200);
    //pinMode(13, OUTPUT);
}

void loop()
{
    a0 = analogRead(A0); //A1-bottom
    a1 = analogRead(A1); //A2-left_top
    a2 = analogRead(A2); //A3-right
    a3 = analogRead(A3); //A4-U_1_battery
    a4 = analogRead(A4); //A4-U_2_battery
    a5 = analogRead(A5); //A4-U_3_battery
    a6 = analogRead(A6); //A4-U_4_battery
    a7 = analogRead(A7); //A5-U_R10hm
    topSwitch = digitalRead(11);
    bottomSwitch = digitalRead(12);

    //горизонтальное движение
    if (man) { //auto
        if ((a1 - a2) > (delta)) {
            _StM1.step(-1);
        }
        else {
            if ((a1 - a2) < (-delta)) {
                _StM1.step(1);
            }
            else {
                _StM1.noHold();
            }
        }
    }
    else { //manual
        if (left) {
            _StM1.step(-1);
        }
        else {
            if (right) {
                _StM1.step(1);
            }
            else {
                _StM1.noHold();
            }
        }
    }
}
}

```

```

//вертикальное движение
if (man) { //auto
  if (topSwitch) {
    if ((a1 - a0) >= (-delta)) {
      _StM2.noHold();
    }
  }
  else {
    if ((a1 - a0) > (delta)) {
      _StM2.step(1); //вниз
    }
  }
  if (bottomSwitch) {
    if ((a1 - a0) <= (delta)) {
      _StM2.noHold();
    }
  }
  else {
    if ((a1 - a0) < (-delta)) {
      _StM2.step(-1); //вверх
    }
  }
  if ((a1 - a0) >= (-delta) && (a1 - a0) <= (delta)) {
    _StM2.noHold();
  }
} else { //manual
  if (!bottomSwitch && down) {
    _StM2.step(-1); //вниз
  }
  else {
    if (!topSwitch && up) {
      _StM2.step(1); //вверх
    }
    else {
      _StM2.noHold();
    }
  }
}
}

// Полученные значения передаем в OPC
holdingRegs[0] = a0; //int
holdingRegs[1] = a1; //int
holdingRegs[2] = a2; //int
holdingRegs[3] = a3; //int
holdingRegs[4] = a4; //int
holdingRegs[5] = a5; //int
holdingRegs[6] = a6; //int
holdingRegs[7] = a7; //int
holdingRegs[8] = delta; //int
holdingRegs[9] = topSwitch; //bool
holdingRegs[10] = bottomSwitch; //bool
holdingRegs[11] = man; //bool

modbus_update();
delta = holdingRegs[8];
man = holdingRegs[11];
digitalWrite(13, man);
left = holdingRegs[12];
right = holdingRegs[13];
up = holdingRegs[14];
down = holdingRegs[15];
}

```

Рисунок 8. Программа

Справа от солнечных батарей стоят 3 фоторезистора, которые меняют сопротивление в зависимости от падающего на них света. Напряжение питания Arduino +5 В через них подается на соответствующие аналоговые входы. Пара 1-2 отвечает за горизонтальное позиционирование, а пара 2-3 за вертикальное. Между ними зеркальные перегородки для увеличения углов реагирования на источник света (рисунки 9 и 10)

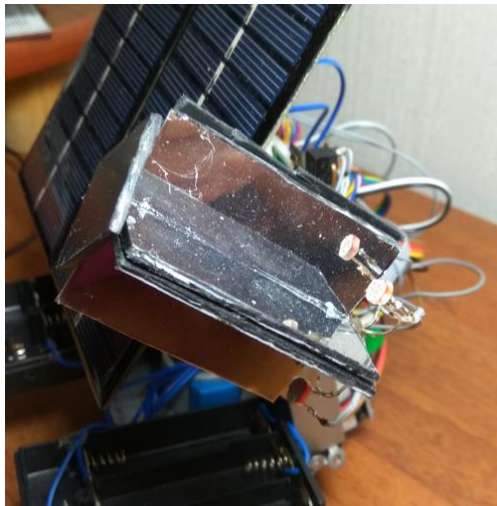


Рисунок 9. Вид сбоку

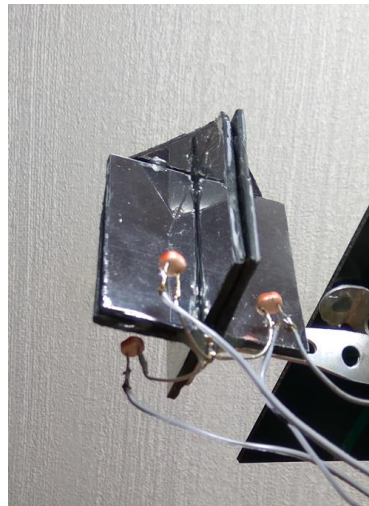


Рисунок 10. Вид сзади

Также на другие аналоговые входы подается напряжение с делителей, подключенных к солнечным батареям. Делители разные по сопротивлению для имитации разных режимов работы. Считанные со входов данные помещаются в HOLDING REGISTERS платы Arduino.

Для отправки данных на компьютер организована линия передачи TTL - RS485 - USB порт компьютера. OPS сервер опрашивает HOLDING REGISTERS солнечного трекера по протоколу ModBUS (рисунок 12), а для архивирования, отображения мнемосхемы и управления используется MasterSCADA (рисунок 11) компании Insat (<http://www.insat.ru/>).

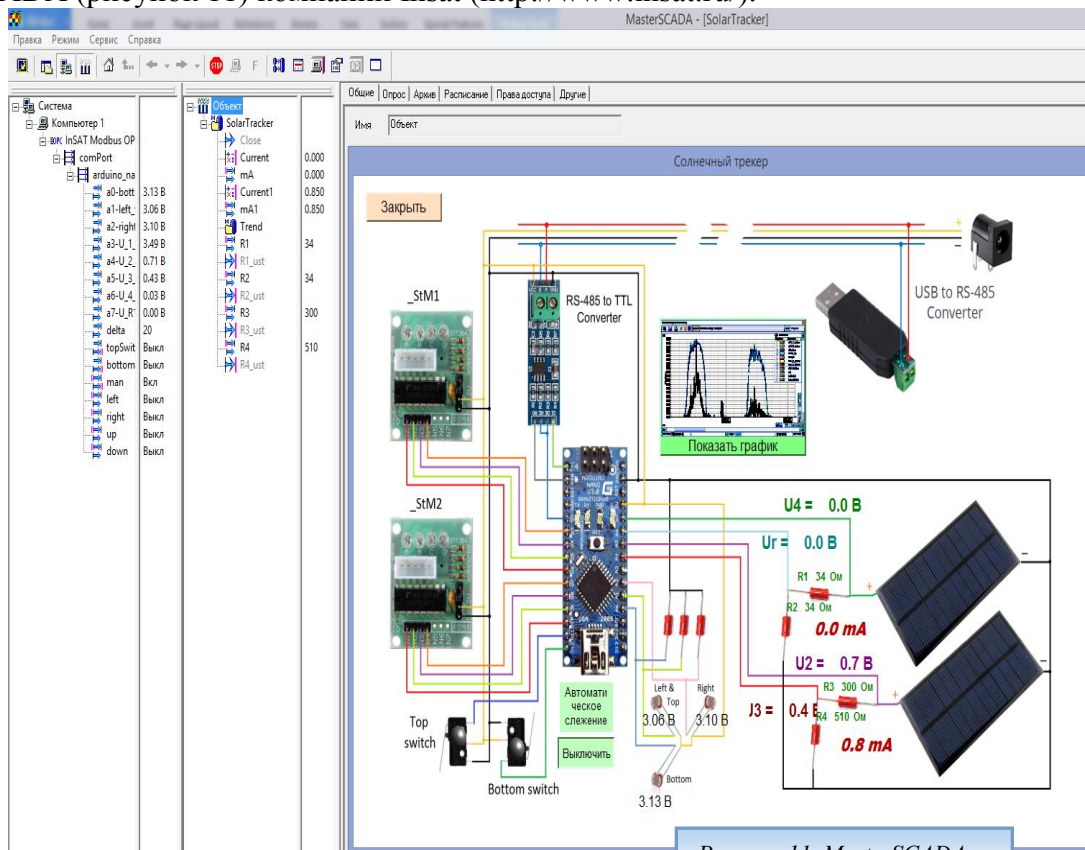


Рисунок 11. MasterSCADA

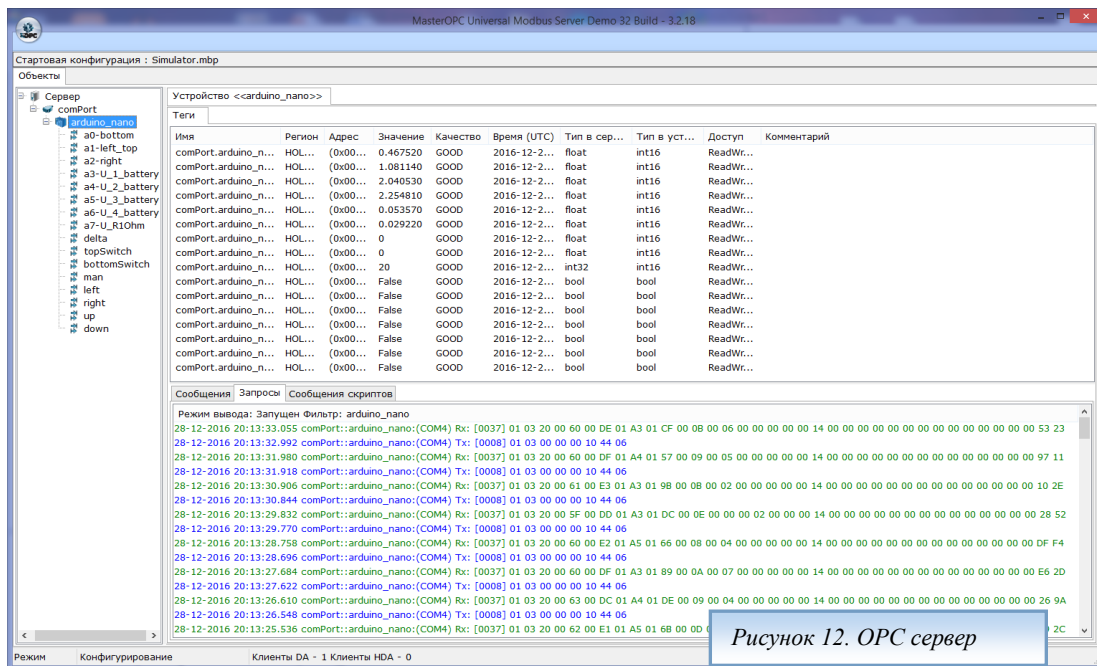


Рисунок 12. OPC сервер

Для имитации Солнца в эксперименте использовалась лампа накаливания 100 Вт, а для позиционирования на одинаковом расстоянии “Солнца от Земли” - половина обруча. Обруч разделен на 14 равных промежутков, которые использованы для дискретизации траектории перемещения “Солнца”. Изменение положения лампы происходит по сигналу метронома через каждые 5 секунд. В реальном времени идет запись значений и построение графика в Master Scada (рисунок 13)

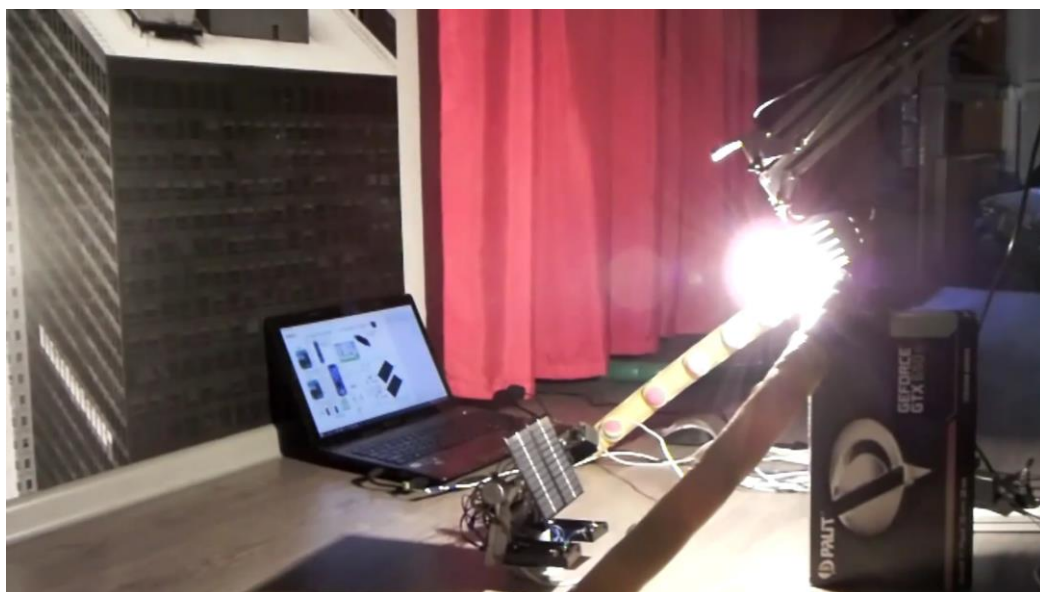


Рисунок 13. Эксперимент

В первой части эксперимента трекер “следит за Солнцем” (рисунок 14)

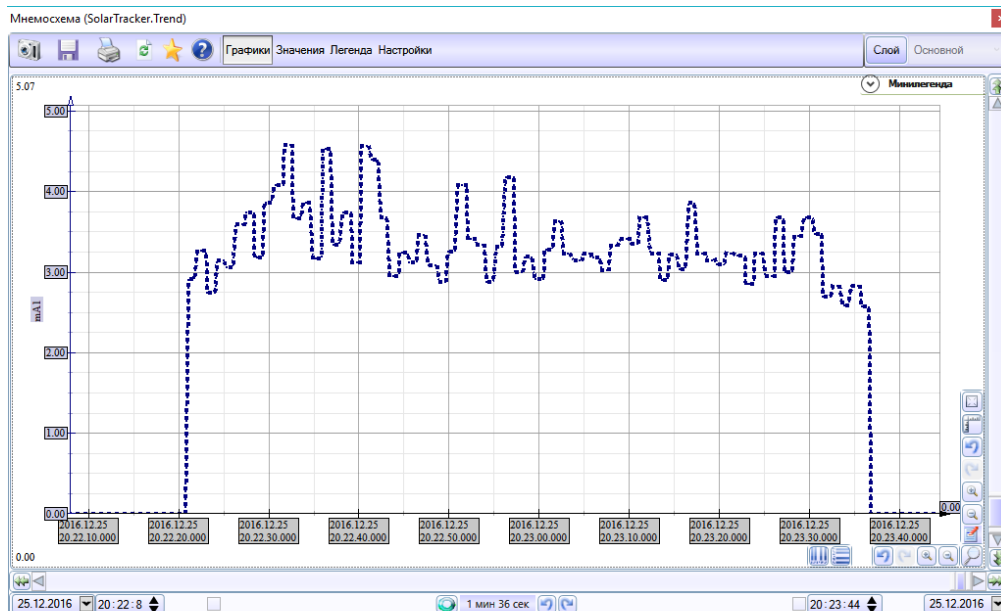


Рисунок 14. График со включенным автоматическим слежением

Во второй части трекер направлен на центральную точку обруча и обездвижен (рисунок 15)

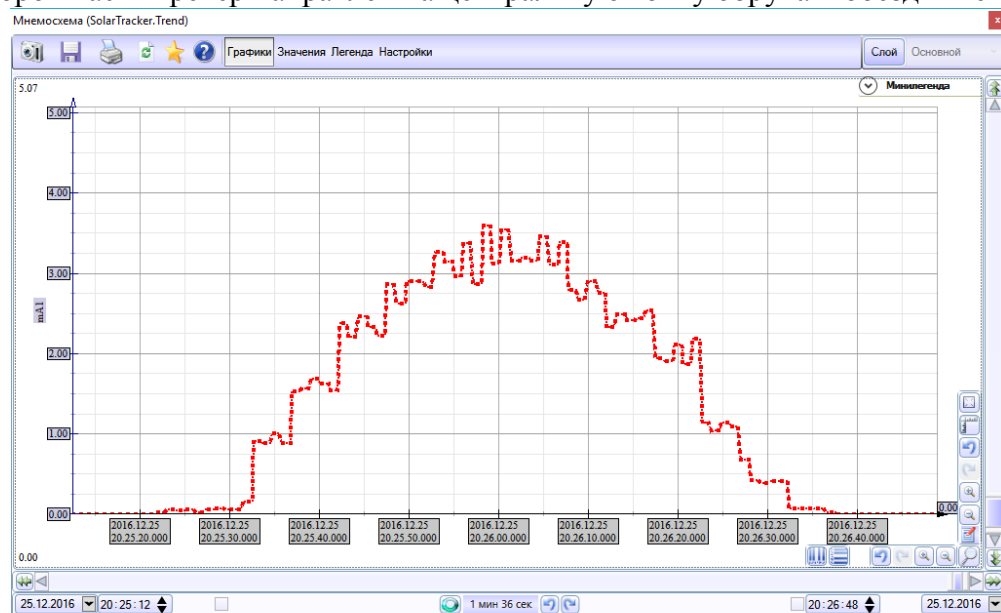


Рисунок 15. График с выключенным автоматическим слежением

Проанализировали полученные графики (рисунок 16)



Рисунок 16. Анализ полученных графиков

Отношение площадей получившихся фигур примерно равно отношению собранных энергий, получаемых с использованием и без использования поворотного устройства.

В проведенном эксперименте солнечный трекер позволил собрать энергии на 86% больше. Рассчитаем затраты энергии на поворотную систему.

Найдем мощность, которую потребляет один шаговый двигатель по формуле:

$$P = U \cdot I$$

Потребляемая мощность одного шагового двигателя равна (процесс измерения представлен на нашем видео <https://www.youtube.com/watch?v=biYNSnW62qI>)

$$160\text{mA} \cdot 5\text{В} = 800\text{mВт}$$

Соответственно 2 шаговых двигателя будут потреблять 1600мВт. Будем исходить из того, что двигатели работают 0,5 часа в сутки, что более чем достаточно, т.к. полный оборот трекер делает меньше, чем за минуту. Соответственно энергия на шаговых двигателях за сутки будет равна

$$1600\text{mВт} \cdot 0,5\text{ч} = 800\text{mВт} \cdot \text{ч}$$

Плата Arduino тоже потребляет энергию. В других просмотренных мной подобных проектах используется Arduino Mega, которая потребляет 40мА тока. В этом проекте стоит Arduino Nano 3.0 (340), которая потребляет 20мА тока (источник https://www.arduino.md/arduino_power_consumption/)

Соответственно энергия, потребляемая платой за сутки равна

$$5\text{В} \cdot 20\text{mA} \cdot 24\text{ч} = 2400\text{mВт} \cdot \text{ч}$$

Итого потребление трекера равно

$$2400 + 800 = 3200\text{mВт} \cdot \text{ч} = 3,2\text{Вт} \cdot \text{ч}$$

Одна солнечная панель имеет мощность около 0,6Вт (источник

<http://mysku.ru/blog/aliexpress/39516.html>). У нас их 2, значит их мощность будет равна

$$0,6\text{Вт} \cdot 2 = 1,2\text{Вт}$$

Найдем время в течении которого трекер будет автономным и не будет тратить энергии больше, чем получает.

$$\frac{3,2\text{Вт} \cdot \text{ч}}{1,2\text{Вт}} = 2,66\text{ч}$$

Расчеты показывают, что трекер сможет работать в автономном режиме в течении суток, если за это время минимум 2ч 40мин солнце будет активным. Если солнце будет более активным, то оставшееся время позволит собирать ему энергию для других нужд. Мощность шаговых двигателей позволяет, не меняя затрат, увеличить количество солнечных панелей в несколько раз.

Таким образом, использование подобных поворотных систем, что следует из проведенного эксперимента, является целесообразным для сбора электрической энергии. Затрачивая минимум энергии солнечный трекер позволяет собрать на 86% энергии больше. Это позволит странам нахо-

дящимся дальше от экватора собирать энергии столько же, сколько в данный момент собирают на экваторе солнечные панели, например, расположенные на крышах домов.

В данное время трекер получает энергию от блока питания. В последствии планируется добиться его автономной работы.

А также планируется подключить солнечные панели к DC-DC преобразователю, на выходе которого выставить зарядный ток автономного питания, что позволит собирать электроэнергию, начиная с 2-3 В напряжения на солнечных панелях, т.е. в пасмурную и другую неблагоприятную погоду.

За ходом предстоящих экспериментов можно наблюдать на <http://skif.tk/>

Примеры тезисов научных работ победителей Всероссийского конкурса научных работ школьников Юниор по инженерным наукам 2016-2017 учебный год

Секция «Физика»

Секция: физика и астрономия
«Национальный исследовательский ядерный университет «МИФИ»
Саровский физико-технический институт —
филиал федерального государственного автономного образовательного учреждения высшего образования «Национальный исследовательский ядерный университет «МИФИ»
(СарФТИ НИЯУ МИФИ)
607186, Нижегородская область, г. Саров, ул. Духова, д. 6
Тел. (83130) 7-02-22, тел./факс (83130) 3-48-09 E-mail: sarfti@mephi.ru

ОПРЕДЕЛЕНИЕ РАЗМЕРОВ И КОНЦЕНТРАЦИИ КАПЕЛЬ ВОДЫ В ЭКСПЕРИМЕНТАХ С КАМЕРОЙ ВИЛЬСОНА
DEFINITION OF SIZE AND CONCENTRATION OF WATER DROPLETS IN A CLOUD CHAMBER EXPERIMENTS

Гончаров Евгений

Класс: 11

607181 г. Саров ул. Березовая д.4 кв.15

Тел: 8(83130)90497 сот. 89100069332

E-mail: e-point-black@yandex.ru

Научные руководители: Базаров Ю.Б., начальник лаборатории, ФГУП РФЯЦ-ВНИИЭФ

Голубев М.Б., старший научный сотрудник, ФГУП РФЯЦ-ВНИИЭФ;

Процессы испарения и конденсации воды играют определяющую роль во многих атмосферных явлениях. Кинетикой конденсации определяется их характер развития, например, формирование грозы или торнадо. В отличие от испарения конденсация может происходить как за доли секунд, так и за часы, и выделение энергии при этом можно сравнивать либо с взрывом, либо с медленным горением. Для расчетно-теоретического моделирования этих процессов необходима надежная и достоверная модель кинетики испарения и конденсации.

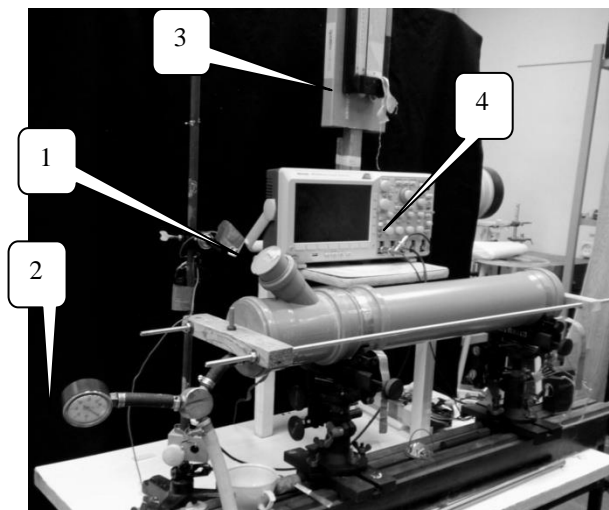
Существующие в настоящее время теоретические модели сталкиваются с принципиальными трудностями, например, такими как необходимость наличия центров конденсации в виде ионов, аэрозолей и т.д. В представленной ситуации на первый план выступают экспериментальные исследования.

На актуальность проблемы указывает тот факт, что с 2009 года в CERNе реализуется проект CLOUD [1], цель которого – исследование образования и эволюции облаков. В проекте принимают участие специалисты 17-ти институтов. Эксперименты проводятся с использованием камеры Вильсона объемом 26 м³ (рисунок 1).



Рисунок 1. Установка проекта CERN CLOUD.

На базе гидродинамической лаборатории СарФТИ НИ ЯУ МИФИ экспериментальные исследования процесса конденсации водяного пара проводятся также с 2009 года. Для экспериментальной верификации теоретических моделей старшеклассниками школ г. Сарова была построена установка, основанная на принципе камеры Вильсона. Визуальные наблюдения процесса конденсации дали основания предполагать, что размер и концентрация капель воды в камере зависят от начального давления: чем выше начальное давление в камере, тем больше капель в единице объема, и тем меньше их размер. На рисунке 2 представлена одна из модификаций установки, целью которой является оценка размеров сконденсировавшихся капель и их концентрации внутри камеры.



1 - камера; 2 - манометр; 3 - гигрометр психрометрический; 4 - осциллограф
Рисунок 2 - экспериментальная установка.

Для измерений использовался метод, измерения ослабления сигнала фотоприёмника (фотодиод ФД24), регистрировавшего рассеяние луча лазера на каплях, с одновременным измерением температуры внутри камеры.

Типичные осциллограммы, зарегистрированные в экспериментах, приведены на рисунке 3.

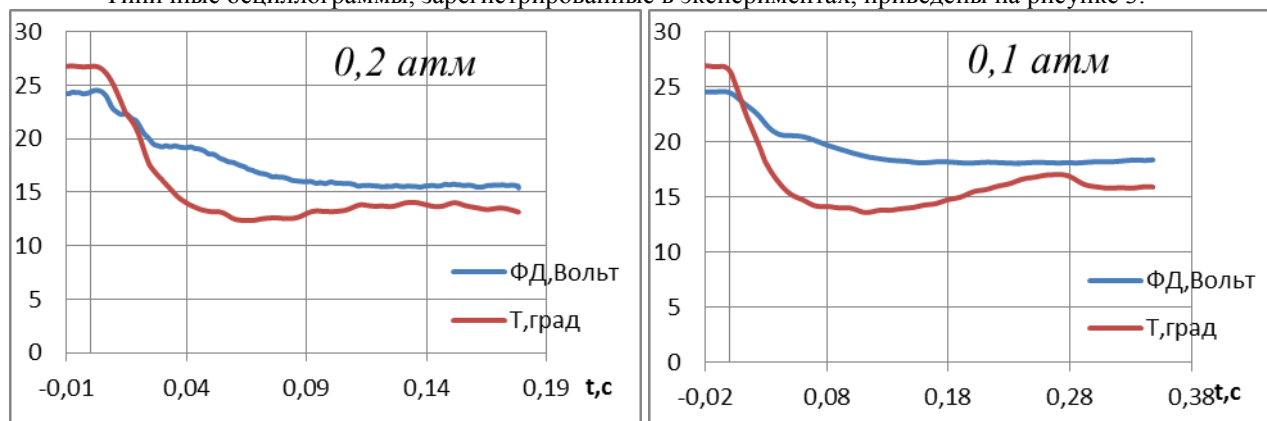


Рисунок 3 – зависимость температуры и сигнала с фотодиода от времени для разных начальных давлений.

В приведённых экспериментах рассчитанный радиус каплей составил $5,72 \cdot 10^{-5}$ см и $1,89 \cdot 10^{-4}$ см соответственно, а их концентрация (количество) $1,13 \cdot 10^6$ и $7,31 \cdot 10^4$ на см^3 .

В докладе приводятся результаты экспериментов по измерению размеров и концентрации каплей воды, сконденсировавшихся в камере после сброса давления.

Литература

1. <http://cloud.web.cern.ch/cloud>

