Знаете ли вы, что можно производить электричество c помощью детей? В данной статье описывается научный эксперимент, который наглядно демонстрирует экологичное добывание энергии. Задача состоит в том, чтобы сделать детскую комнату оборудованную полом для выработки энергии.

**Концепция проекта.**

Пьезоэлектричество было открыто в середине **18 века**. Эффект заключается в следующем: электрический **заряд** **накапливается** в некоторых твердых **материалах**, таких как: кристаллы, некоторые виды керамики, в ответ на приложения механической нагрузки. Звучит невероятно! Именно эту технологию и большую активность детей играющих в детских комнатах я решил совместить в своем проекте.

**Почему не использовать динамо-машины.**

Насколько это, возможно, в своих проектах я пытаюсь избегать использованию динамо. Динамо-машины вырабатывают гораздо больше электроэнергии, но от них много шума и устаёт крутить рука или нога. Ну и я бы хотел, чтобы электричество вырабатывалось само, без каких либо усилий либо сложных механизмов.

**Практическое применение.**

Хотя это и звучит смешно, но добыча электроэнергии из детей– далеко не шуточное занятие. К примеру, мою технологию можно использовать для питания электроприборов, работающих на постоянном токе, пример светодиодного освещения или систем типа «умного дома» как в моем проекте.

**Проблемы.**
Угольные электростанция являются наиболее распространёнными источниками электроэнергии во всем мире. Поэтому страны мира ищут альтернативные источники электроэнергии. Атомная энергетика может быть опасной, в случае аварий, как это случилось в Японии и в СССР. Гидроэлектростанции наносят огромный вред сложившимся экосистемам и тоже в случае аварий несут угрозу как людям, так и окружающей среде. Солнечные и ветряные электростанции очень зависимы от погодных условий, которые могут быть непредсказуемыми.

Моей целью было найти новый источник энергии, который не зависел бы от погодных явлений. Пройдя через сотни статей, что связаны с альтернативной энергетикой и выполнив десятки экспериментов, однажды, сказав себе, что просто скопировать или взять за основу чужую идею – это не мой путь. Я как настоящий изобретатель, должен придумать и построить все с самого нуля.

**Будущие применение полученной технологии.**

1. Создание напольного покрытия со встроенными пьезогенераторами.

**Исследования.**

В проекте использовались пьезоэлектрические технологии. Некоторые материалы обладают способностью накапливать электрический заряд под воздействием механических сил. В качестве примера выступают несколько видов керамики, сегнетовой соли, а также другие различные виды твердых частиц. Для примера: Цирконат-титанат-свинца (**PZT**), генерирует измеряемое электричество, когда его структура деформируется примерно на 0,1% от первоначального размера.

В этом проекте величина генерируемой электроэнергия будет определяться и записываться, для того, чтобы в дальнейшем определить, возможно ли заряжать литий-ионный аккумулятор или конденсатор большой емкости.

Пьезоэлектрический эффект, с помощью которого материал генерирует электрический потенциал был изучен Карлом Линнем и Францом Эпинуса в средине 18 века. Опираясь на эти знания Рене Жюст Гаюи и Антуан Сезар Беккерель предложили зависимость между механическими нагрузками и электрическими зарядами. Однако их эксперименты потерпели неудачу.

Первая демонстрация прямого пьезоэлектрического эффекта была представлена в 1880 году братьями Пьером и Жаком Кюри. Они объединили свои знания пьезоэффекта с пониманием основ кристаллических структур. Синтез подобных знаний позволил предсказать поведение кристалла и продемонстрировать эффект возникновения электричества используя кристаллы турмалина, кварца, топаза, тростникового сахара и сегнетовой соли (тартрат тетрагидрата калий натрия). Самый лучшей результат был получен при использовании кварца и сегнетовой соли.

Пьезоэлектрический диск генерирует напряжение при деформации (изменение формы сильно преувеличено). Однако Кюри не стали прогнозировать обратный эффект. Обратный эффект был выведен математически из принципов термодинамики Габриэлем Липпманом в 1881 году. Кюри немедленно подтвердили существование обратного эффекта и отправились на получение количественного доказательства полной обратимости электро-упругой-механической деформации в пьезоэлектрических кристаллах.

В течении следующих нескольких десятилетий, пьезоэлектричество оставалось чем-то вроде лабораторного любопытства. Много роботы было проделано для определения и исследования кристаллических структур, которые накапливают электричество. Это привело к тому, что в 1910 году был опубликован труд, в котором описывались более 20 природных кристаллов, что способны генерировать электроэнергию, строго определены константы и т.д.

**Ход работы**

**Необходимые материалы:**

* батарейный блок с USB;
* Пьезоэлектрические преобразователи– **6 шт**;
* **1N4007 –** выпрямительный диод – **4 шт**;
* Провод – **30 см**;

Спаиваем пьезоэлементы параллельно. Если припаять их последовательно, величина тока вырастит, а напряжения – упадет. Для преобразования переменного тока в постоянный изготавливаем диодный мост из 4х диодов.

Наконец настало время проверить всю теории в практике. Подключим цифровой мультиметр в режиме амперметра, включив 2-значный диапазон измерения постоянного тока. Помните, что ток в элементах образуется в момент нажатия и держится короткий период времени. Поэтому для более читабельных показаний воспользуемся конденсатором на **100 нФ**.

**Показания вольтметра:**
Нажатие рукой = **15.03 В (2 mA);**

Несколько шагов по полу = **18.53 В (5 mA);**

Прыжки = **27.89 В (11 mA);**

Генератор в пиковых значениях выдает до **28 В**. Хотя значение тока и небольшое, но напряжение вполне в состоянии повредить устройство, питающееся от 5В.

**Выводы/отчет.**

Текущие результаты показали, что величина тока генерируемая пьезоэлементом, достаточна для зарядки литий-ионного аккумулятора. Хотя и присутствуют недостатки, текущие положительные результаты дают возможность развивать проект в дальнейшем. Электрический пол вырабатывает достаточно энергии, чтобы запитывать маломощные схемы, такие как микроконтроллеры и передатчики TTL Bluetooth. После всего этого, можно с гордостью сказать, что продукт готов к использованию в детских комнатах. Для примера, мной создана модель «умного дома» (комнаты) с автономным питанием от пьезогенератора, мощности которого хватает для заряда аккумулятора для питания всех автономных систем входящих в систему. Реле - для управления блоками розеток бытовых приборов, например, телевизора, датчик освещенности для контроля работы системы освещения, датчик температуры для управления системой вентиляции, приемника пульта управления системой.

**Дальнейшие планы.**

Использовать мою технологию во всех торговых центров в Санкт – Петербурге

**Использование моего устройства**

Моё устройство служит для подзарядки аккумулятора робота, который с помощью твоей улыбки может пропускать тебя в детскую комнату весёлым способом. Мой робот состоит из аккумулятора, модуля артинтрек, два серводвигателя, камерой, микроконтроллер – трекдуино. Артинтрекэто модуль, осуществляющий обработку видеопотока с помощью нейронных сетей и компьютерного зрения, содержащий в себе. Трекдуино — это Arduino-совместимый многофункциональный **микроконтроллер** полностью российского производства. **Трекдуино** -аппаратно-программный комплекс, основными компонентами которого являются плата ввода и среда разработки.

**Алгоритм**

Алгоритм работы робота состоит в следующем: Робот с помощью камеры фиксирует изображение, затем нейросеть определяет наличие лица в кадре и эмоцию которую оно изображает (грусть, радость, удивление и нейтральное выражение лица) в зависимости от эмоции в контроллер Трекдуино передается переменная присвоенная каждой эмоции. В зависимости от переменной, встроеный светодиод загарается разными цветами, задаются разные углы серводвигателей управляющих руками робота и меняется изображение на экране дисплея, отображающего анимацию эмоции.

**Программа**

Вот программа к моему роботу

#include "Trackcv.h"

#include "String.h"

#include <Display.h>

#include <Servo.h>

bool comm\_send(char data) {return Serial2.write(data) == 1;}

int32\_t comm\_recv() {return (int32\_t)Serial2.read();}

void delay\_ms(uint32\_t value) {delay(value);}

uint32\_t TS\_ms(void) {return (uint32\_t)millis();}

Display display(UART1);

int prev\_emotion = 0;

int color = YELLOW;

int new\_color = YELLOW;

Servo servo\_port\_OUT2;

Servo servo\_port\_OUT3;

int \_ABVAR\_1\_a;

void setup() {

 Serial.begin(115200);

 Serial2.begin(115200);

 display.brightness(500);

 servo\_port\_OUT2.attach(OUT2, SMALL\_SERVO);

 servo\_port\_OUT3.attach(OUT3, SMALL\_SERVO);

 Serial.print("start trackcv...\n");

 Trackcv\_init(comm\_recv, comm\_send, 0);

 servo\_port\_OUT2.write( 0 );

 servo\_port\_OUT3.write( 180 );

}

void clear\_emotion() {

 //clear smile

 display.line(120, 70, 140, 67, BLACK);

 display.line(190, 67, 210, 70, BLACK);

 display.line(160, 160, 120, 140, BLACK);

 display.line(160, 160, 200, 140, BLACK);

 //clear sad

 display.line(130, 70, 150, 70, BLACK);

 display.line(190, 70, 210, 70, BLACK);

 display.line(120, 160, 160, 140, BLACK);

 display.line(160, 140, 200, 160, BLACK);

 //clear no\_emotion

 display.line(120, 160, 200, 160, BLACK);

 //clear anger

 display.line(120, 65, 140, 70, BLACK);

 display.line(190, 70, 210, 65, BLACK);

 display.line(140, 160, 180, 160, BLACK);

 //clear suprise

 display.line(130, 60, 150, 60, BLACK);

 display.line(180, 60, 200, 60, BLACK);

 display.circle(160, 150, 10, BLACK);

}

void smile() {

 delay(250);

// display.clear();

 if (prev\_emotion != 1 || new\_color != color) {

 clear\_emotion();

 color = new\_color;

 display.circle(160, 120, 100, color);

 display.line(120, 70, 140, 67, color);

 display.circleFilled(130, 100, 10, color);

 display.line(190, 67, 210, 70, color);

 display.circleFilled(190, 100, 10, color);

 display.line(160, 160, 120, 140, color);

 display.line(160, 160, 200, 140, color);

 servo\_port\_OUT2.write( 180 );

 servo\_port\_OUT3.write( 0 );

 prev\_emotion = 1;

 } else {

 }

}

void sad() {

 delay(250);

// display.clear();

 if (prev\_emotion != 2 || new\_color != color) {

 clear\_emotion();

 color = new\_color;

 display.circle(160, 120, 100, color);

 display.line(130, 70, 150, 70, color);

 display.circleFilled(130, 100, 10, color);

 display.line(190, 70, 210, 70, color);

 display.circleFilled(190, 100, 10, color);

 display.line(120, 160, 160, 140, color);

 display.line(160, 140, 200, 160, color);

 servo\_port\_OUT2.write( 0 );

 servo\_port\_OUT3.write( 180 );

 prev\_emotion = 2;

 } else {

 }

}

void anger() {

 delay(250);

// display.clear();

 if (prev\_emotion != 3 || new\_color != color) {

 clear\_emotion();

 color = new\_color;

 display.circle(160, 120, 100, color);

 display.line(120, 65, 140, 70, color);

 display.circleFilled(130, 100, 10, color);

 display.line(190, 70, 210, 65, color);

 display.circleFilled(190, 100, 10, color);

 display.line(140, 160, 180, 160, color);

 for (\_ABVAR\_1\_a=1; \_ABVAR\_1\_a<= ( 3 ); ++\_ABVAR\_1\_a )

 {

 servo\_port\_OUT2.write( 0 );

 servo\_port\_OUT3.write( 0 );

 delay( 300 );

 servo\_port\_OUT2.write( 180 );

 servo\_port\_OUT3.write( 180 );

 delay( 300 );

 }

 prev\_emotion = 3;

 } else {

 }

}

void suprise() {

 delay(250);

// display.clear();

 if (prev\_emotion != 4 || new\_color != color) {

 clear\_emotion();

 color = new\_color;

 display.circle(160, 120, 100, color);

 display.line(130, 60, 150, 60, color);

 display.circleFilled(130, 100, 10, color);

 display.line(180, 60, 200, 60, color);

 display.circleFilled(190, 100, 10, color);

 display.circle(160, 150, 10, color);

 for (\_ABVAR\_1\_a=1; \_ABVAR\_1\_a<= ( 5 ); ++\_ABVAR\_1\_a )

 {

 servo\_port\_OUT2.write( 90 );

 servo\_port\_OUT3.write( 90 );

 delay( 200 );

 servo\_port\_OUT2.write( 180 );

 servo\_port\_OUT3.write( 0 );

 delay( 200 );

 }

 prev\_emotion = 4;

 } else {

 }

}

void no\_emotion() {

 delay(250);

// display.clear();

 if (prev\_emotion != 0 || new\_color != color) {

 clear\_emotion();

 color = new\_color;

 display.circle(160, 120, 100, color);

 display.circleFilled(130, 100, 10, color);

 display.circleFilled(190, 100, 10, color);

 // display.line(160, 160, 120, 140, BLUE);

 display.line(120, 160, 200, 160, color);

 servo\_port\_OUT2.write( 90 );

 servo\_port\_OUT3.write( 90 );

 prev\_emotion = 0;

 } else {

 }

}

bool inited = false;

void loop() {

 if(!inited) {

 if (trackcv\_get\_errno() != ERR\_OK) {

 Serial.println("trackcv...fail");

 builtInRGB(RED);

 if(trackcv\_check()) {

 Serial.print("check ok\n");

 } else {

 Serial.print("check fail\n");

 }

 delay(250);

 return;

 } else {

 Serial.print("trackcv...ok\n");

 trackcv\_neural\_start(Neural\_script\_id\_emotion);

 inited = true;

 builtInRGB(OFF);

 }

 }

 if (buttonRead(BTN\_UP))

 {

 builtInRGB(BLUE);

 new\_color = BLUE;

 }

 if (buttonRead(BTN\_DOWN))

 {

 builtInRGB(WHITE);

 new\_color = WHITE;

 }

 if (buttonRead(BTN\_LEFT))

 {

 builtInRGB(YELLOW);

 new\_color = YELLOW;

 }

 if (buttonRead(BTN\_RIGHT))

 {

 builtInRGB(GREEN);

 new\_color = GREEN;

 }

 if(trackcv\_neural\_count() > 0) {

 if(

 trackcv\_neural\_class\_count(0) > 0 &&

 trackcv\_neural\_class\_p(0, 0) > 80

 ) {

// Serial.print("face: ");

// Serial.print(trackcv\_neural\_x(0));

// Serial.print(" ");

// Serial.println(trackcv\_neural\_y(0));

// Serial.print("\n");

 if(trackcv\_neural\_class\_p(0,1) == Neural\_emotion\_HAPPY) { // happy

 builtInRGB(GREEN);

 smile();

// Serial.print("happy \n");

 } else if (trackcv\_neural\_class\_p(0,1) == Neural\_emotion\_SAD) { // sad

 sad();

 builtInRGB(RED);

// Serial.print("sad \n");

 } else if (trackcv\_neural\_class\_p(0,1) == Neural\_emotion\_ANGER) { // anger

 anger();

 builtInRGB(RED);

// Serial.print("happy \n");

 } else if (trackcv\_neural\_class\_p(0,1) == 3) { // suprise

// } else if (trackcv\_neural\_class\_p(0,1) == Neural\_emotion\_SUPRISE) { // anger

 suprise();

 builtInRGB(GREEN);

// Serial.print("anger \n");

 } else {

 builtInRGB(BLUE);

 no\_emotion();

 }

 }

 } else {

 builtInRGB(OFF);

 }

 delay(100);

 Errno errno = ERR\_OK;

 if ((errno = trackcv\_get\_errno()) != ERR\_OK) {

 Serial.print("trackcv err ");

 Serial.println(errno);

 builtInRGB(OFF);

 inited = false;

 return;

 }

}