DZ_Generator_NOTES-2

English

I will deepen my knowledge with Arduino for our project, but I see that Arduino offers a function that manages time in microconds, with a precision of up to 3 µs.

This function is: delayMicroseconds ()

I think it's the one to use.

See this page:
http://www.mon-club-elec.fr/pmwiki_reference_arduino/pmwiki.php?n=Main.DelayMicroseconds

Example of a minimalist code that uses the method:

// For our example, we need a 1340 microsecond (µs) relay interrupt after it is powered up

// Obviously, it will be necessary to adjust taking into account its own reaction time at closing and opening

int pinRelay_8 = 8; // use digital pin n ° 8

void setup ()

{

   pinMode (pinRelay_8, OUTPUT); // configure pin 8 output

}

void loop () // loop every 16666 microseconds corresponding to the closing cycle of the 60 Hz relay

                // We must subtract the pause of 1340 microseconds from the actionRelay method, which is blocking. This done 16666 - 1340 = 15326 µs

{

   actionRelay (pinRelay_8); // call the method that closes the relay for 1340 microseconds

   delayMicroseconds (15326); // pause 15326 microseconds before releasing the relay again

}

```
void actionRelay (int relayNumber)

{

    digitalWrite (relayNumber, HIGH); // Closing the relay

    delayMicroseconds (1340); // break 1340 microseconds

    digitalWrite (relayNumber, LOW); // Opening the relay

}
```

To everyone

I have some good news which I've been holding on for some days until we had it tested and confirmed to work.

Forum member Chet made arrangements with slider2732 (member of our group) to write an Arduino switching sequence program for me since I don't know how to do that.

However, Pierre sent me his program he used in his demonstrated device. So I sent the program to slider2732 for testing. He was happy to report it was a much simpler program then he was writing.   He loaded it in his Mega 2560 and the program worked right away.

He even made a demo video to show it working: https://www.youtube.com/watch?v=fPZn3CcdVVc

Pierre requested I share the program with serious builders only. This will be your gift from Pierre once you get to the point of needing it.

So please stop trying to figure out how the Mega 2560 Arduino will work since we have already confirmed it works.

If you want to receive the program, then you have to build it to the point that only the program is needed. Once I see your video demo to confirm you are ready then I will send you the program.

Please don't ask for it otherwise. I am only following Pierre's request.

Regards Luc

Pierre just posted a video of his code: https://www.youtube.com/watch?v=TACAVCU2fIY

Looks like his potentiometer value is mapped to a range from x=1 to 10ms. He sets three outputs high at a time and then issues a delay(x) and then turns the next three coils on, delay(x), turn the first three coils off, etc.

Each relay stays on for 3*x ms, then remains off for 33*x ms. Switching 12 succeeding relays in sequence takes 36*x ms. His program describes one cycle (of the three per full rotation), then repeats.

loop {

1 on 13 on 25 on

delay(x)

(bug, see below: bold lines below should actually go here)

2 on 14 on 26 on

delay(x)

1 off 13 off 25 off

delay(x)

3 on 15 on 27 on

delay(x)

2 off 14 off 26 off

delay(x)

etc..

delay(x)

12 on 24 on 36 on

delay(x)

11 off 23 off 35 off

delay(x)

12 off 24 off 36 off

delay (x)

}

I do note he might have a bug in his code wrapping the loop: this last line should have been positioned after the very first two lines :

1 on 13 on 25 on

delay(x)

12 off 24 off 36 off

delay (x)

2 on 14 on 26 on

delay(x)

But hey, maybe this is what causes overunity   ;)   (just kidding).   Pm

====================
====================

More news

I just checked with Pierre to see if the program can now go public and he replied YES! so here it is.

français

Plus de nouvelles

Je viens de vérifier avec Pierre pour voir si le programme peut maintenant être rendu public et il a répondu OUI! donc c'est ici.


```
/*

  Blink    This example code is in the public domain.


  modified 8 May 2014

  by Scott Fitzgerald

 */


// the setup function runs once when you press reset or power the board

void setup() {
```

```
// initialize digital pin 13 as an output.


  pinMode(1, OUTPUT);

   pinMode(2, OUTPUT);

  pinMode(3, OUTPUT);

   pinMode(4, OUTPUT);

  pinMode(5, OUTPUT);

   pinMode(6, OUTPUT);

    pinMode(7,OUTPUT);

  pinMode(8, OUTPUT);

   pinMode(9, OUTPUT);

  pinMode(10, OUTPUT);

   pinMode(11, OUTPUT);

  pinMode(12, OUTPUT);

   pinMode(13, OUTPUT);

    pinMode(14,OUTPUT);

  pinMode(15, OUTPUT);

   pinMode(16, OUTPUT);

   pinMode(17, OUTPUT);

   pinMode(18, OUTPUT);

  pinMode(19, OUTPUT);

   pinMode(20, OUTPUT);

  pinMode(21, OUTPUT);

   pinMode(22, OUTPUT);

    pinMode(23,OUTPUT);
```

```
    pinMode(24, OUTPUT);

     pinMode(25, OUTPUT);

    pinMode(26, OUTPUT);

     pinMode(27, OUTPUT);

    pinMode(28, OUTPUT);

     pinMode(29, OUTPUT);

      pinMode(30,OUTPUT);

    pinMode(31, OUTPUT);

    pinMode(32, OUTPUT);

       pinMode(33, OUTPUT);

     pinMode(34, OUTPUT);

      pinMode(35,OUTPUT);

    pinMode(36, OUTPUT);

     pinMode(37, OUTPUT);

}


int x = 0;


// the loop function runs over and over again forever

void loop(){

  int y= analogRead(0); // transforme x en une valeur de 0 à 20 (millisecondes)

   x= map(y,0,1000 ,1,100);


   digitalWrite(1,HIGH), digitalWrite(13,HIGH), digitalWrite(25,HIGH);     // turn the LED on (HIGH is the
voltage level)

   delay(x);                    // wait for a second
```

```
    digitalWrite(2,HIGH), digitalWrite(14,HIGH), digitalWrite(26,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                         // wait for a second

    digitalWrite(1,LOW),    digitalWrite(13,LOW),    digitalWrite(25,LOW);         // turn the LED off by
making the voltage LOW

    delay(x);                         // wait for a second

    digitalWrite(3,HIGH), digitalWrite(15,HIGH), digitalWrite(27,HIGH);      // turn the LED on (HIGH is the
voltage level)

    delay(x);                         // wait for a second

    digitalWrite(2,LOW),    digitalWrite(14,LOW),    digitalWrite(26,LOW);         // turn the LED off by
making the voltage LOW

    delay(x);                         // wait for a second

    digitalWrite(4,HIGH), digitalWrite(16,HIGH), digitalWrite(28,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                         // wait for a second

    digitalWrite(3,LOW),    digitalWrite(15,LOW),    digitalWrite(27,LOW);         // turn the LED off by
making the voltage LOW

    delay(x);                         // wait for a second

    digitalWrite(5,HIGH), digitalWrite(17,HIGH), digitalWrite(29,HIGH);      // turn the LED on (HIGH is the
voltage level)

    delay(x);                         // wait for a second

    digitalWrite(4,LOW),    digitalWrite(16,LOW),    digitalWrite(28,LOW);         // turn the LED off by
making the voltage LOW

    delay(x);                         // wait for a second

    digitalWrite(6,HIGH), digitalWrite(18,HIGH), digitalWrite(30,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                         // wait for a second

    digitalWrite(5,LOW),    digitalWrite(17,LOW),    digitalWrite(29,LOW);         // turn the LED off by
making the voltage LOW
```

```
    delay(x);                       // wait for a second

   digitalWrite(7,HIGH), digitalWrite(19,HIGH), digitalWrite(31,HIGH);      // turn the LED on (HIGH is the
voltage level)

    delay(x);                       // wait for a second

   digitalWrite(6,LOW),    digitalWrite(18,LOW),    digitalWrite(30,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                       // wait for a second

   digitalWrite(8,HIGH), digitalWrite(20,HIGH), digitalWrite(32,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                       // wait for a second

   digitalWrite(7,LOW),    digitalWrite(19,LOW),    digitalWrite(31,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                       // wait for a second

   digitalWrite(9,HIGH), digitalWrite(21,HIGH), digitalWrite(33,HIGH);      // turn the LED on (HIGH is the
voltage level)

    delay(x);                       // wait for a second

   digitalWrite(8,LOW),    digitalWrite(20,LOW),    digitalWrite(32,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                       // wait for a second

   digitalWrite(10,HIGH), digitalWrite(22,HIGH), digitalWrite(34,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                       // wait for a second

   digitalWrite(9,LOW),    digitalWrite(21,LOW),    digitalWrite(33,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                       // wait for a second

   digitalWrite(11,HIGH), digitalWrite(23,HIGH), digitalWrite(35,HIGH);      // turn the LED on (HIGH is
the voltage level)

    delay(x);                       // wait for a second

   digitalWrite(10,LOW),    digitalWrite(22,LOW),    digitalWrite(34,LOW);        // turn the LED off by
```

making the voltage LOW

```
    delay(x);                    // wait for a second

    digitalWrite(12,HIGH), digitalWrite(24,HIGH), digitalWrite(36,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                    // wait for a second

    digitalWrite(11,LOW),    digitalWrite(23,LOW),    digitalWrite(35,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                    // wait for a second

    digitalWrite(12,LOW),    digitalWrite(24,LOW),    digitalWrite(36,LOW);        // turn the LED off by
making the voltage LOW

    delay(x);                    // wait for a second


    }
```

==================
==================

Here's Pierre's sketch altered for Luc's 30-slot stator. I followed the original pattern but with 10 instead of 12 pins between each group member.

Please check my work!

Now we have 10 groups of 3 output pins, in overlapping sequence.

I've preserved the odd timing of the original (groups 1 and 10 only on for 2 timing intervals instead of 3.)

I also compressed the pin declarations into a for loop and changed some variable names and comments to make it clearer.

français

Voici le croquis de Pierre modifiée pour le stator à 30 emplacements de Luc. J'ai suivi le modèle original mais avec 10 broches au lieu de 12 entre chaque membre du groupe.

S'il vous plaît vérifier mon travail!

Maintenant, nous avons 10 groupes de 3 broches de sortie, en séquence de chevauchement.

J'ai conservé le timing étrange de l'original (groupes 1 et 10 seulement sur 2 intervalles de temps au lieu de 3.)

J'ai également compressé les déclarations de broche dans une boucle for et changé quelques noms de variables et commentaires pour le rendre plus clair.

```
/*

   Pierre's Original Sketch

   Modded by TK 24 Mar 2018

   altered for 30-slot stator (gotoluc)


 Speed control pot: connect wiper to Analog 0, legs of pot to +5v and GND


 */

 int stepspeed = 0, rawspeed = 0;
void setup() {

    // initialize all needed digital pins as OUTPUT

   int pin = 1;

   for (pin = 1; pin <= 31; pin ++)

      pinMode(pin, OUTPUT);

   pinMode(A0, INPUT);

 }
void loop() {

    rawspeed = analogRead(0); // read speedpot on Analog Pin 0

   stepspeed = map(rawspeed,0,1023,1,100);    //transforme rawspeed en une valeur de 0 à 100
(millisecondes)

   digitalWrite(1,HIGH), digitalWrite(11,HIGH), digitalWrite(21,HIGH);      // turn the LED on (HIGH is the
voltage level)

    delay(stepspeed);                        // wait for stepspeed value in milliseconds

   digitalWrite(2,HIGH), digitalWrite(12,HIGH), digitalWrite(22,HIGH); // turn the LED on (HIGH is the
voltage level)
```

```
    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(1,LOW),   digitalWrite(11,LOW),   digitalWrite(21,LOW);      // turn the LED off by
making the voltage LOW

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(3,HIGH), digitalWrite(13,HIGH), digitalWrite(23,HIGH);     // turn the LED on (HIGH is the
voltage level)

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(2,LOW),   digitalWrite(12,LOW),   digitalWrite(22,LOW);      // turn the LED off by
making the voltage LOW

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(4,HIGH), digitalWrite(14,HIGH), digitalWrite(24,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(3,LOW),   digitalWrite(13,LOW),   digitalWrite(23,LOW);      // turn the LED off by
making the voltage LOW

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(5,HIGH), digitalWrite(15,HIGH), digitalWrite(25,HIGH);     // turn the LED on (HIGH is the
voltage level)

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(4,LOW),   digitalWrite(14,LOW),   digitalWrite(24,LOW);      // turn the LED off by
making the voltage LOW

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(6,HIGH), digitalWrite(16,HIGH), digitalWrite(26,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(5,LOW),   digitalWrite(15,LOW),   digitalWrite(25,LOW);      // turn the LED off by
making the voltage LOW

    delay(stepspeed);                    // wait for stepspeed value in milliseconds

    digitalWrite(7,HIGH), digitalWrite(17,HIGH), digitalWrite(27,HIGH);     // turn the LED on (HIGH is the
```

voltage level)

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(6,LOW),   digitalWrite(16,LOW),   digitalWrite(26,LOW);   // turn the LED off by making the voltage LOW

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(8,HIGH), digitalWrite(18,HIGH), digitalWrite(28,HIGH); // turn the LED on (HIGH is the voltage level)

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(7,LOW),   digitalWrite(17,LOW),   digitalWrite(27,LOW);   // turn the LED off by making the voltage LOW

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(9,HIGH), digitalWrite(19,HIGH), digitalWrite(29,HIGH);   // turn the LED on (HIGH is the voltage level)

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(8,LOW),   digitalWrite(18,LOW),   digitalWrite(28,LOW);   // turn the LED off by making the voltage LOW

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(10,HIGH), digitalWrite(20,HIGH), digitalWrite(30,HIGH); // turn the LED on (HIGH is the voltage level)

   delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(9,LOW),   digitalWrite(19,LOW),   digitalWrite(29,LOW);   // turn the LED off by making the voltage LOW

   delay(stepspeed);   // wait for stepspeed value in milliseconds

 // digitalWrite(11,HIGH), digitalWrite(21,HIGH), digitalWrite(31,HIGH);   // turn the LED on (HIGH is the voltage level)

 // delay(stepspeed);   // wait for stepspeed value in milliseconds

   digitalWrite(10,LOW),   digitalWrite(20,LOW),   digitalWrite(30,LOW);   // turn the LED off by making the voltage LOW

   delay(stepspeed);   // wait for stepspeed value in milliseconds

}

Just a quick note about your changes TK.

The curled brackets are missing in the pins loop.

So it should read:

for (pin = 1; pin <= 31; pin ++)

    { pinMode(pin, OUTPUT); }

  fr. Juste une petite note sur vos changements TK.

Les accolades manques dans la boucle des broches.

Donc, il devrait lire:

for (pin = 1; pin <= 31; pin ++)

    { pinMode(pin, OUTPUT); }

==================

Ya, thanks, you are right the curly brackets should be used even if there is only a single statement in the for loop.

At any rate... here is the sketch modified for Luc's 30 slot stator, with the timing "bug-error-OUSecret" corrected so

that there is a smooth flow around the outputs. I think.

fr. Oui merci, vous avez raison, les accolades doivent être utilisées même s'il n'y a qu'une seule instruction dans la boucle for.

En tout cas ... voici l'esquisse modifiée pour le stator à 30 slots de Luc, avec le timing "bug-error-OUSecret" corrigé donc

qu'il y a un flux fluide autour des sorties. Je pense.


```
/*********************************************
```

  Pierre's Original Sketch

Modded by TK 24 Mar 2018

altered for 30-slot stator (gotoluc)

TKTest Version using pins 2-11,12-21,22-31


```
**********************************************/
  // int stepspeed = 0, rawspeed = 0;    // not used
void setup() {
    // initialize all needed digital pins as OUTPUT, plus a couple extras
   int pin = 1;
   for (pin = 1; pin <= 33; pin ++) {
     pinMode(pin, OUTPUT);
    }
      pinMode(A0, INPUT);    // speed potentiometer wiper pin
 }
void loop() {
    // rawspeed = analogRead(0); // read speedpot on Analog Pin 0
   // stepspeed = map(rawspeed,0,1023,1,200);    //transforme rawspeed en une valeur de 0 à 200
(millisecondes)
   //
   // (better to read pot after each switch command for real-time speed control, as below)
   digitalWrite(11,LOW), digitalWrite(21,LOW), digitalWrite(31,LOW);          // turn the LED off
   delay(map(analogRead(0),0,1023,1,200));                             // wait for stepspeed value in milliseconds
   digitalWrite(3,HIGH),    digitalWrite(13,HIGH), digitalWrite(23,HIGH);      // turn the LED on
   delay(map(analogRead(0),0,1023,1,200));                      // wait
   digitalWrite(2,LOW),    digitalWrite(12,LOW),    digitalWrite(22,LOW);          // turn the LED off
```

```
delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(4,HIGH),   digitalWrite(14,HIGH), digitalWrite(24,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(3,LOW),    digitalWrite(13,LOW),   digitalWrite(23,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(5,HIGH),   digitalWrite(15,HIGH), digitalWrite(25,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(4,LOW),    digitalWrite(14,LOW),   digitalWrite(24,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(6,HIGH),   digitalWrite(16,HIGH), digitalWrite(26,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(5,LOW),    digitalWrite(15,LOW),   digitalWrite(25,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(7,HIGH),   digitalWrite(17,HIGH), digitalWrite(27,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(6,LOW),    digitalWrite(16,LOW),   digitalWrite(26,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(8,HIGH),   digitalWrite(18,HIGH), digitalWrite(28,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(7,LOW),    digitalWrite(17,LOW),   digitalWrite(27,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(9,HIGH),   digitalWrite(19,HIGH), digitalWrite(29,HIGH);    // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(8,LOW),    digitalWrite(18,LOW),   digitalWrite(28,LOW);     // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait
```

```
digitalWrite(10,HIGH), digitalWrite(20,HIGH), digitalWrite(30,HIGH);      // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(9,LOW),    digitalWrite(19,LOW),    digitalWrite(29,LOW);        // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(11,HIGH), digitalWrite(21,HIGH), digitalWrite(31,HIGH);      // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(10,LOW), digitalWrite(20,LOW),    digitalWrite(30,LOW);        // turn the LED off

delay(map(analogRead(0),0,1023,1,200));                    // wait

digitalWrite(2,HIGH), digitalWrite(12,HIGH), digitalWrite(22,HIGH);      // turn the LED on

delay(map(analogRead(0),0,1023,1,200));                    // wait
 }
```

==================

I received a very bad message from Pierre:

it only rotates a magnetic field, you will not get much voltage with it as there's something else I kept secret. It's like an engine with no pistons, it won't start. It's the same thing for the dz generator, the important thing is not what we see but rather what you don't see.

I have seen too many people who want to take credit in my place. So my contribution stops here. You'll have to discover the rest on your own. I can tell you it's not easy to achieve, it took me nearly 2 years to find the solution as rotation is only one aspect of the dz, it's really more complicated than that.

At least people will be able to learn how to rotate a magnetic field but it will stop there, they won't get the result I have.

I even still have issues to fix but I think I can solve everything with my second prototype that's presently being built and will take several months. It all need to be started from the beginning, new circuit, new more powerful coils and a new sequence program that looks to be complicated.

So I wish you the best of luck with the continuation and beware of those who try to keep the information for themselves and don't reveal anything else. I wonder why, humm

When I've applied for a patent I can show you the rest of the version 2 dz generator but not before. Thank you.

================= on or about 25 March 2018 =========

Well, that's unfortunate to loose Pierre's trust. I knew something was up but I didn't want to alarm the group. I thought once we had this new moderated topic and Pierre could see we are serious researchers and truly want to work together with him to understand the effect and then further improve it, in time would come to understand that Erfinder's information was all a purposeful sabotage to stop the research.

I'm quite sure Pierre has lost trust in me (mentioning Patent) with all the misinformation Erfinder posted about me in the first topic which none of it was translated and if Pierre used google translate that would causes even more confusion as it does a really bad job from English to French.

What Erfinder did It's an old tactic call divide and conquer.   You must all remember how Arron did the same thing to get Eric Dollard to himself.

Anyways, what can we do other then move forward with what we have. Maybe if Pierre sees we don't give up and work together he will understand that all that was misinformation purposely done to divide us so we give up and go our separate ways.

So, the question now is will you guys let that happen?   or will we stay and work together?

Please post your truthful replies as I'm sure Pierre is reading what's going on.

Kind regards Luc

The below is my translation of Pierre's message cheors post #65

it only rotates a magnetic field, you will not get much voltage with it as there's something else I kept secret. It's like an engine with no pistons, it won't start. It's the same thing for the dz generator, the important thing is not what we see but rather what you don't see. I have seen too many people who want to take credit in my place. So my contribution stops here. You'll have to discover the rest on your own. I can tell you it's not easy to achieve, it took me nearly 2 years to find the solution as rotation is only one aspect of the dz, it's really more complicated than that. At least people will be able to learn how to rotate a magnetic field but it will stop there, they won't get the result I have. I even still have issues to fix but I think I can solve everything with my second prototype that's presently being built and will take several months. It all need to be started from the beginning, new circuit, new more powerful coils and a new sequence program that looks to be complicated. So I wish you the best of luck with the

continuation and beware of those who try to keep the information for themselves and don't reveal anything else. I wonder why, humm

When I've applied for a patent I can show you the rest of the version 2 dz generator but not before. Thank you.

----
https://overunity.com/17653/pierres-170w-in-1600w-out-looped-very-impressive-build-continued-moderated/msg518915/#msg518915 --------

I never told the world to replicate the generator DZ.

I just wanted to show on YouTube what I and nobody else could do.

These are the people who wanted to know how the DZ generator worked and all these look excited when I talked about the basics and started to watch the video frame by frame to know my secrets and you have a big chunk of these secrets.

Later, I mentioned to Luc that the first prototype was not yet developed and he still wanted to build it. What do you want me to do.

Even though I told him I was starting a second prototype he wanted to make a replica of my first one that did not work 100%. Free to him and the others.

All I said and mentioned for the DZ is true, even the program I put on YouTube.

You will be able to rotate a magnetic field. You just have to find the next one if you like it. I never forced anyone to copy the generator DZ.

Anyway all I read on this forum this week said a lot about someone who wanted to get ahead of others to patent my invention. From that, do you think seriously that I am interested in revealing everything to the world to make me a big finger of honor by another person who would have left to apply for a patent in my place. If it starts to be too heavy to manage I will close my account on YouTube and I will focus on my device and no one will hear from me.

------------ ....

Last night after locking the topic I received the below message from Pierre.

Fr. La nuit dernière, après avoir verrouillé le sujet, j'ai reçu le message ci-dessous de Pierre.

hello Luc, I want to let you know that what I wrote was not aim at you. I was addressing the whole forum.

Yes, I slightly lost confidence of the discussion group. I know there are some honest people in the group

but there are also some rotten apples and they are the ones I'm wary the most of. I'm sure there is one or more people ill-intentioned on the forum.   Yes i wanted to give my invention to the world but the the only thing I do not want is for someone to takeover my idea to make a personal profit.

  How would you feel if you invent something revolutionary and another person takes credit for it in you place. I would not be very happy.

I don't give a dam about money this can bring, I'm not rich but I'm very happy, so money doesn't necessarily give happiness and I really don't do this for money.   All I wanted to start with was to demonstrate my invention that took me months to build and I managed to get it to run.

The test that you saw in my first video that some thinks it's a fake was actually my first real test of the dz generator. I was very happy with the test results and decided to share it online.

Then everything got crazy. I didn't want everyone to try to replicate what I achieved, even you wanted to replicate despite my mentioning that my prototype was only a draft and did not work at its best. I had a lot of improvement to bring to the project even with my second prototype. I'm not even sure of all the needed changes yet, so why all the eagerness to make a replication which in my opinion is not yet operating to full capacity?

When I first saw it working I was wondering what to do with it, I just want to give my invention to the whole world but how do I prevent a person from taking over of my idea to profit from it?   at this moment a patent sounds like the best option, even if I don't have the means to spend so much money for a piece of paper. Until I have the money for a patent the rest of my secrets will stay with me, unless you prove otherwise that if it's made public nobody can apply for a patent in my place?

In the time of Tesla everything was so different, today it's so complicated. So for now I've removed all my youtube videos. You've mention that you're fingers are tired of translating, for me it never stopped ever since I put my video on youtube. So I'll take a break of a few months to free my mind and carefully think this all over.

I want to focus on my second prototype instead of spending hours reading and responding to the good and not so good comments. Looking at what you've built so far as replication it's really a beautiful job and nothing is preventing you to continue, you're on a good path.

The only concern I have are the mosfet's you use, I'm not sure if they will withstand the the input amperage? just a single coil takes a lot of amperage, check before how much one coil draws compared to the 2 amps the mosfet's can handle. That's it for tonight. My week starts very early in the morning and I'm going to rest a little bit.   Pierre Cotnoir

--------------------------------=====================================--------------------

With regards to other things shifting/rotating, that we don't see...

I you look in the back stator region, (see pmgs simulation result without rotor in place), apart from the

pole flux there is also the counter flux due to the rotor load.

The counter flux will be opposed for each pair of adjacent poles, shown in red, assuming both adjacent poles are present during the same period.

Depending on timing, the counter flux could actually aid the adjacent poles flux, if this presented a lower reluctance path.
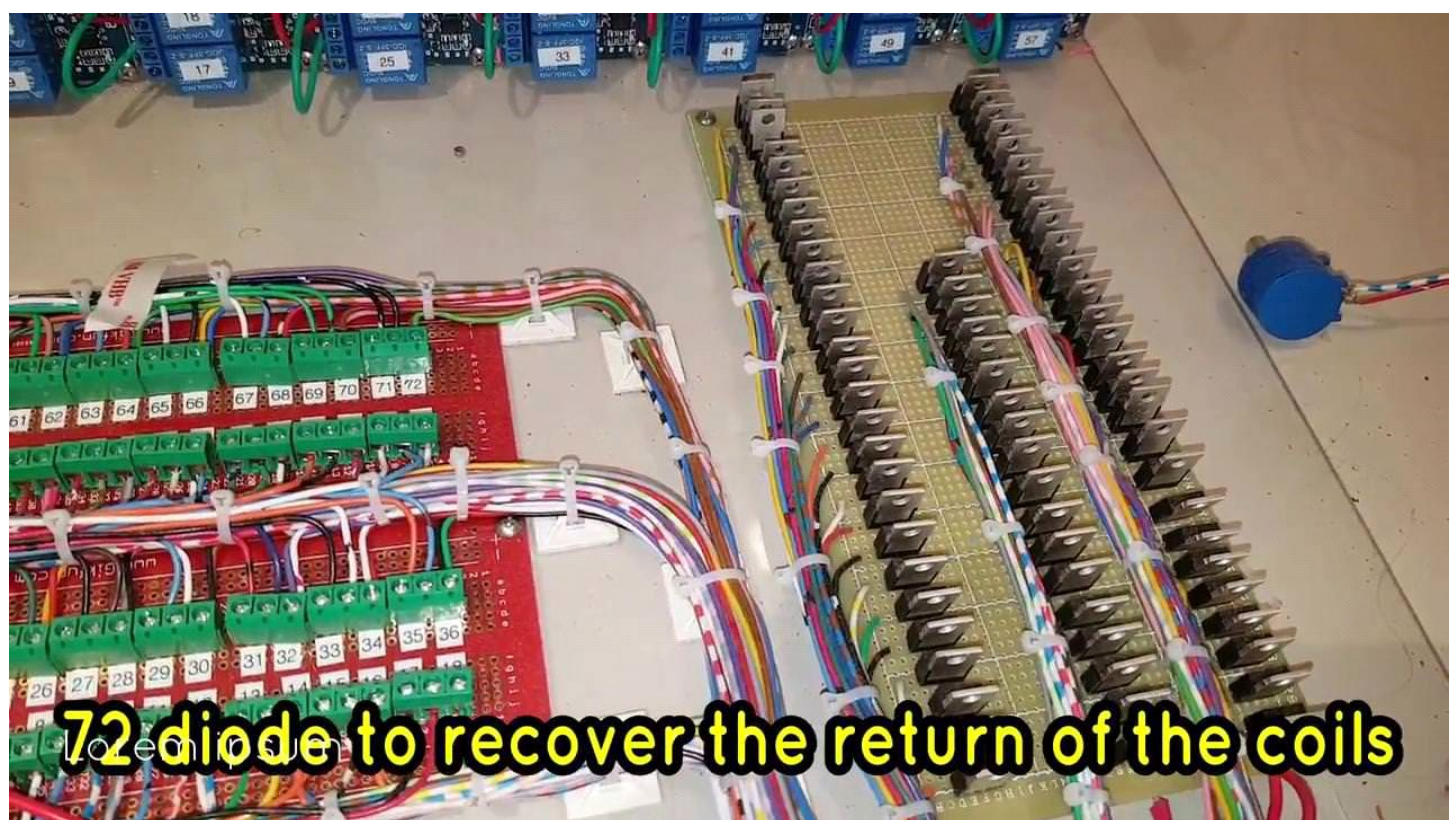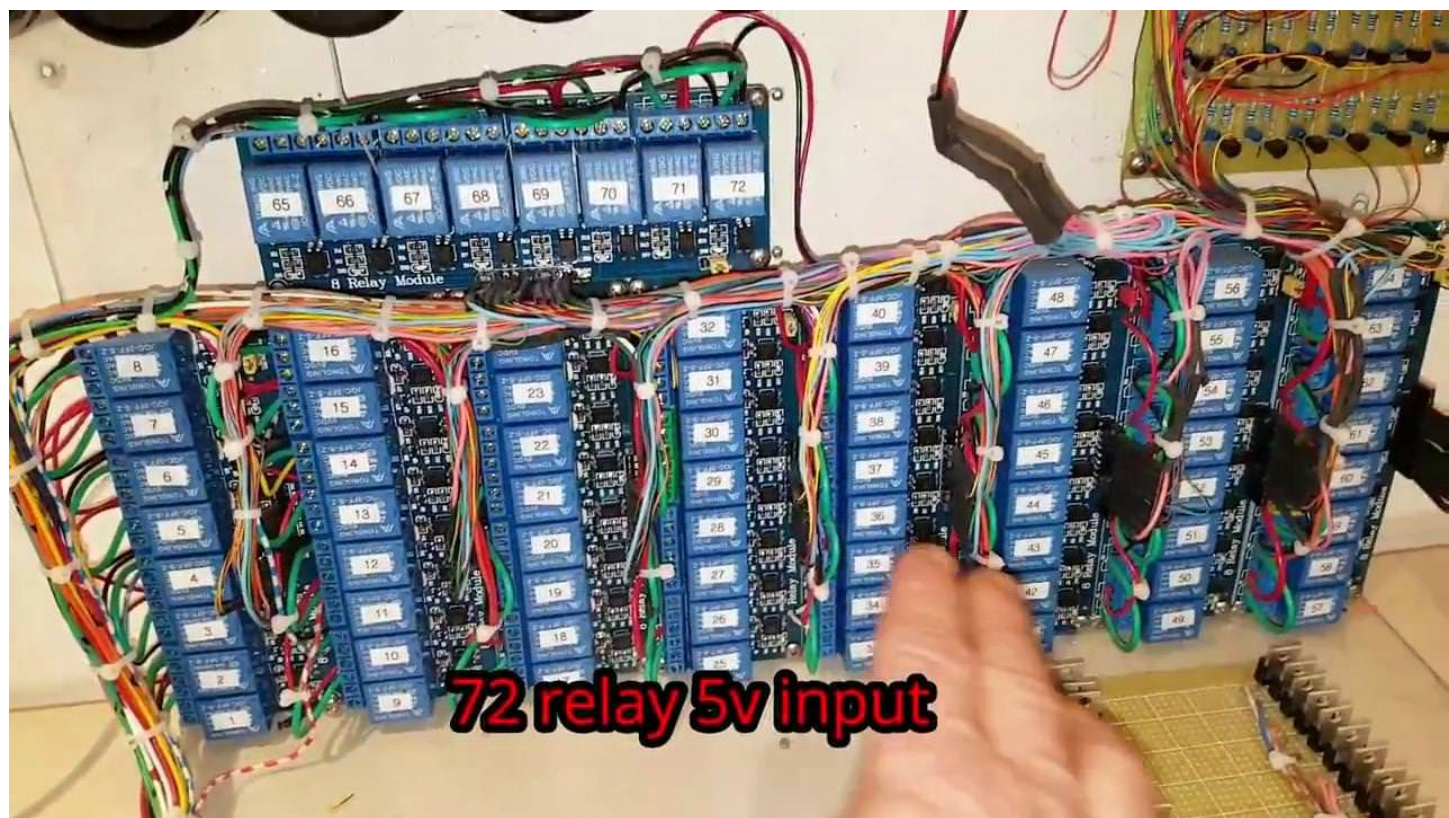
Maybe pmgr can mo

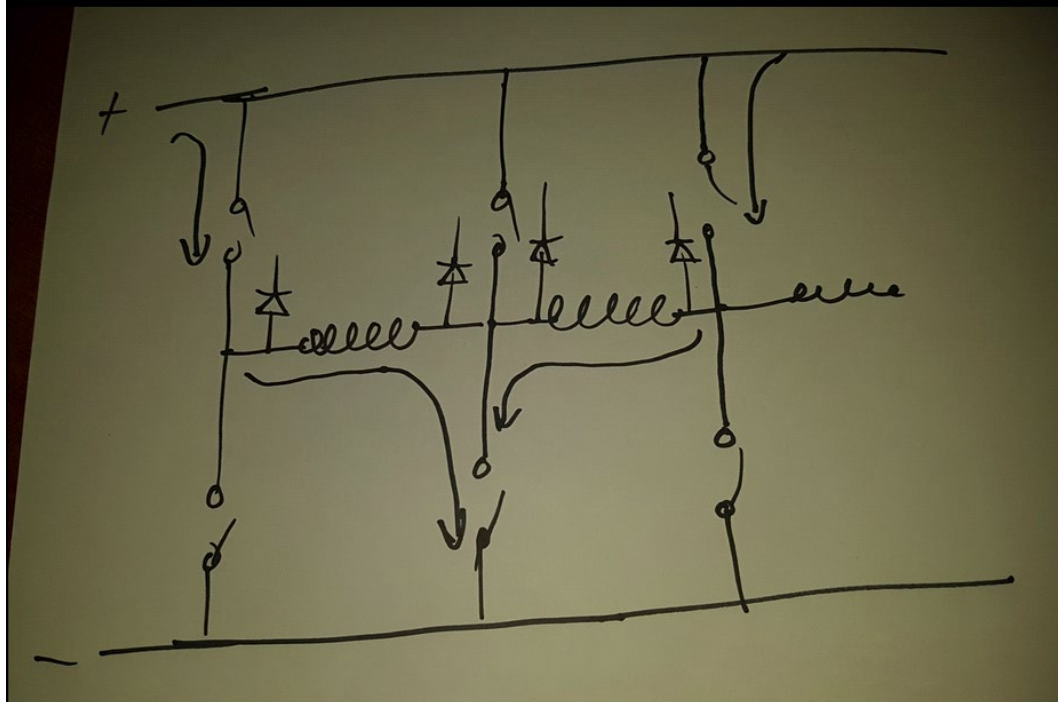del the counter flux, so we can see exactly   what happens here?

Regards L192

================

72 relay 5v input

72 diode to recover the return of the coils
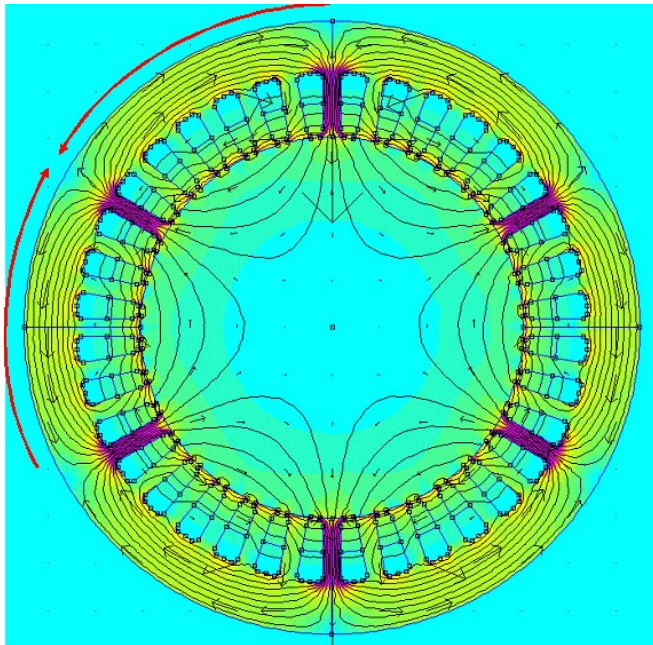
the return wire of the coil that will recharge the supercapacitor

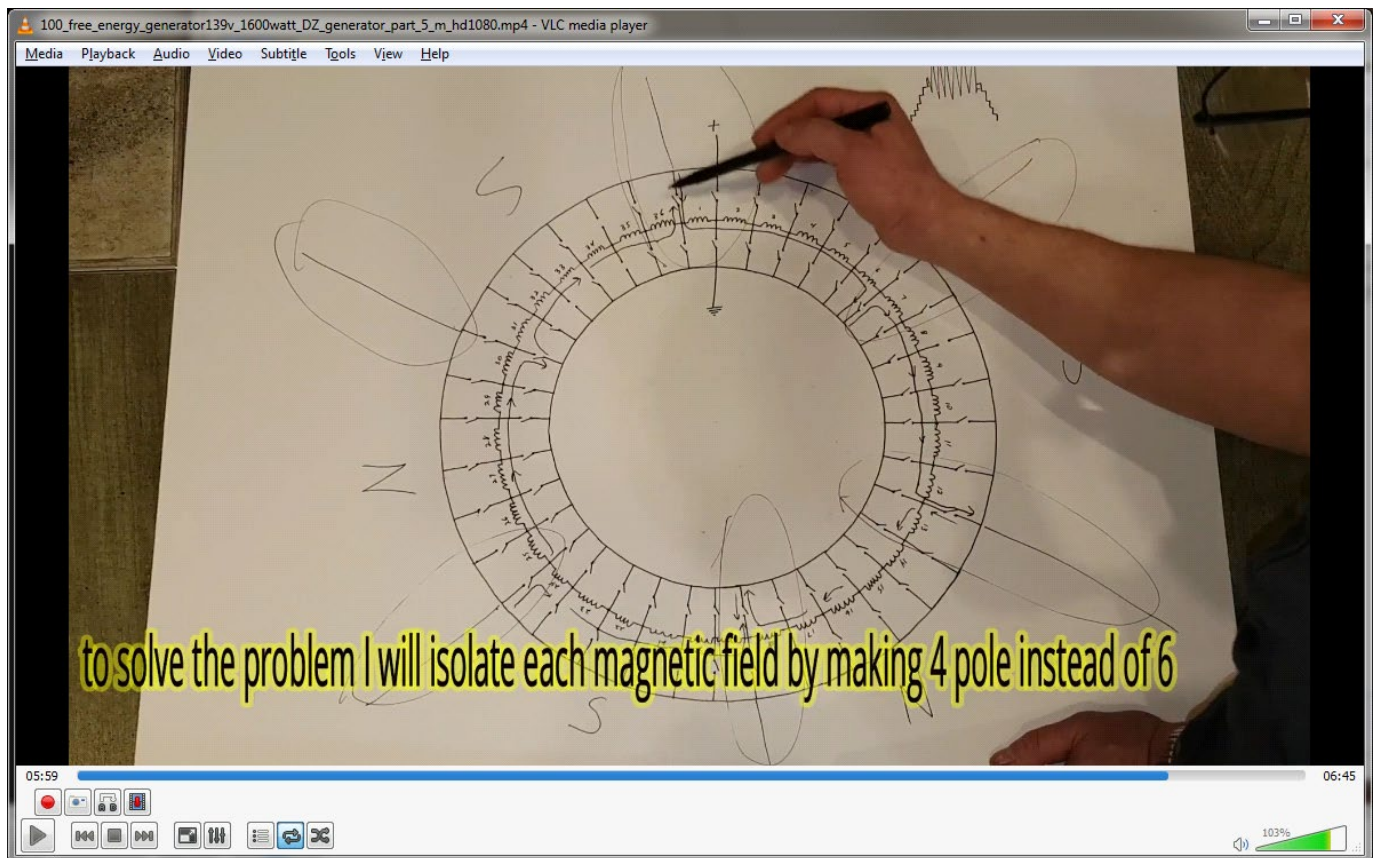700 foot coil 2 x 20 awg wind on iron core laminate whith voltmeter

===============

Eng. About the sequence of the arduino, you will have to turn on one coil and then the other and then turn off the first coil otherwise you will break the magnetic field. The goal is not to turn off the magnetic flux but only to keep it moving ex: 1 on, 2 on, 1 off, 3 on, 2 off etc.

=======================

to solve the problem I will isolate each magnetic field by making 4 pole instead of 6

==================

3 days ago I sent a message to Pierre asking him to confirm if his coils are powered in series. I've just received a reply from him which I translated and below is his original French message.

in my assembly there is 6 coils in series except that there is one that I feed for a fraction of a second so not to lose the magnetic field, so it leaves me with 5 coils operating in reality, example: 1 and 2 are powered by the positive and then 1 goes off to allow the magnetic flux to move etc. So in your case (Pierre is referring to my 30 slot) there should be 4 coils operating using the program I gave you which is good. The only thing I may of modified is the value x and y and played a bit with the speed. Everybody is making too much of a deal with the frequency. I already mentioned that the frequency meter may not of been correct. We can see all the uneven switching noise on the oscilloscope which affected the frequency meter. I adjust the potentiometer but most likely the frequency meter did not read the right frequency. Instead of having a discussion on frequency, people should be working on rotating the magnetic field. Then they can start playing with frequency and increase it to the desired speed.

==================

These lines in Pierre's original code, as an example for the first and second coils support an overlap.


```
  digitalWrite(1,HIGH), digitalWrite(13,HIGH), digitalWrite(25,HIGH);      // turn the LED on (HIGH is the
voltage level)
```

```
    delay(x);                        // wait for a second

    digitalWrite(2,HIGH), digitalWrite(14,HIGH), digitalWrite(26,HIGH); // turn the LED on (HIGH is the
voltage level)

    delay(x);                        // wait for a second

    digitalWrite(1,LOW),   digitalWrite(13,LOW),   digitalWrite(25,LOW);       // turn the LED off by
making the voltage LOW
```

============

Mile High

I tried posting some technical comments on Luc's clip but he deleted them right away and accused me of trolling.   It must be miserable to live in an Orwellian nightmare sometimes, sort of like the movie Fahrenheit 451.   So let's break out of the straitjacket and actually use our critical thinking skills and rebel against the thought-control oppressors that want to stomp out free thought and burn books, much to the chagrin of Luc.

  Here you go, think free and live free or get zapped by a discharging inductor: So you have done the build and you have the working code to energize the coils to make a rotating magnetic field. So now it all comes down to basics.

The energizing coils will put flux through the stationary rotor.

Naturally when the coils that line up with the stationary rotor are energized, then more flux passes through the stationary rotor. By the same token, the coils that do not line up with the stationary rotor put zero or very little flux through the stationary rotor. Makes sense, correct?

So when the Arduino energizes these coils that don't line up with the stationary rotor, that represents the burning of input power with no real output power. It represents power being poured down the drain and all that it really does is heat the coils.

Let's say for the sake of argument you are losing 1/4 of your input power to heat because of the energizing of the coils that don't line up with the stationary rotor. You have enough experience under your belt such that I don't think that you are going to disagree with me on this critical point.

Pierre has already said the classic line, collect the back-EMF pulses from all of the coils. You have been though that exercise 100 times by now, and you know there is no magic there and you are not even close to considering changing your H-bridge setup to collect back-EMF pulse energy into a capacitor. That's a dead horse that you have seen too many times by now such that you can almost do it in your sleep.

So let's talk about the output now. You have a coil wrapped around the stationary rotor and that is where you get the output from the device. So we are talking about the basic nuts and bolts of a generator here. The more turns you use for the output coil, the higher the voltage output. And of course the higher the total resistance of the wire. The more turns you have, the higher the voltage, and therefore the higher value of load resistor the output coil can drive.

In general terms, let's suppose that you wrap enough turns where the wire resistance is 10 ohms. That

implies that you will get the most output power when you have an impedance-matching load resistor of 10 ohms. However, a note of caution here in the sense that the 10-ohm coil will also dissipate the same amount of power as the 10-ohm load resistor, which means right away that you are only 50% efficient in getting output power from the coil.

If you increase the load resistor to say 30 ohms, you will get less total output power in the load resistor, and less dissipated power in the 10-ohm resistance of the coil, giving you a more efficient coil-to-load resistor output power efficiency, but less total output power. So what's the take-away? The take-away is to experiment with different load resistors where the load resistor is less than the coil resistance, equal to the coil resistance, and greater than the coil resistance.

So this glorified Arduino setup with the fancy rotating magnetic field is doing on primary thing - putting changing flux through the stationary rotor.   When you wrap a coil around the stationary rotor and have no load resistor you are seeing the pure EMF being generated by the output coil.   That EMF waveform represents the rate of change of magnetic flux through the stationary rotor with respect to time.   Also known as the first derivative of the flux with respect to time.

I don't know if Luc's DSO has an integration function.   Why do you want the integration function? The answer is that if you integrate on the EMF waveform then you will get a waveform for the actual amount of flux that is passing through the stationary rotor from the Arduino setup.   Even if you don't have an integration function in the DSO, you can do it by hand on paper.

Why do you do it?   Because it is interesting to know how much flux is passing through the stationary rotor.Let's discuss the output when there is an appropriate load resistor.   When there is a lot of current flowing through the load resistor that means that the stator coils that are lined up with the stationary rotor are doing real work, and there should be a corresponding increased current draw when those coils are doing work.

That implies that when you scope the current draw for the whole device you should see lower current draw for the stator coils that don't line up with the stationary rotor (the "dead" current draw that does nothing but heat the stator coils) and increased current draw for the stator coils that do line up with the stationary rotor.

Wrapping more turns around the stationary rotor will NOT give you more output power.   That is just an old wives' tale.   The amount of power you can extract from the stationary rotor is dependent on the amount of changing flux that passes though the rotor, not the amount of turns.

The amount of turns determines the best value for the load resistor, that's all.

So, let's ballpark the maximum output power to input power efficiency for this device.   I have already stated it above.   It's roughly 75% efficient for the stator coils because half of them are "dead" and do nothing.   I am saying this because the unloaded stator coils that do nothing will actually draw very little current, and I am bundling in all of the H-bridge overhead.

Then we know on the output coil side when you use an impedance matching load for maximum power output the efficiency is maximum 50%.   So the ballpark efficiency will be 0.75 x 0.5 = less than 37% efficient.That's a lot of spicy meatballs worth of work with no secret spicy meatball sauce in sight to get less than 37% efficiency.

====================

Hi Luc

I saw and translated Pierre's reply on your newest video he was asking what are you doing with your coils "return" and said this is the secret if you want it to work.

So will guess he means the backemf/recoil/flyback diodes...

====================

The below are two new messages I received from Pierre after he viewed my last French video.

Yes, I know that 6 pole without current variation does not give much but my device has a variation of current.

Congratulations you just discovered one of my secret. There still remains more to discover, then you will see the performance I achieved. And yes, there is really 6 field that rotate, except, there really is a variation which you need to know how to do so you can vary the coils. It may be possible that your configuration can achieve it but I'm not sure of your configuration, you will need to experiment. One thing is sure, if you had 5 pole it will not work. You see, you start to find solutions to the problems.

Sorry, I had left you a quick message since I was not home.

Yes you can make a fluctuation between the field while having 6 magnetic field and can be all done in series. I was able to do it it. There are several ways to do it and you could have done it with only 2 pole in series, 1 north and 1 south with all coils on.

I would of needed to considerably increased the speed but with relays it is not possible. You can do it as you want but it is one of the conditions to get overunity.

You still have one solution to find but the basis is to have a good rotation of the field, a north at one end of the rotor and a south to the other side and a variation of current if you want a maximum of amperage and it's better kept all the coils in series in this configuration but you really are on the right track. Now I understand why you did not have much output with the program that I saw. Everything is explainable. Have a good day

========================

Hi Everyone

Why no backemf/recoil recovery steering diodes on each relay or H bridge switch???????

This energy is being totally disregarded and shoved into garbage can, like most electrical engineers with the proper schooling will do.

Again I mention Pierre said plain as day in Luc's video that the "coil return" (he calls it) is the secret!!!

Why not take what he points out as truth??

That backemf/recoil energy is very much stronger with mechanical switching since such low resistance to the switching....also note that you place steering diodes on the SWITCH, not really the inductor - the energy "comes" from the switch you could say.

So there is that point I want to make the other point maybe even more important (maybe not)

Lets say you shove that magnet wire up through slot 1 and down through slot 6....now you have created an electromagnets of that distance through the stator poles, N one end, S other for example....

So very simple.

Now what about all the winds in between 1 and 6?

They all become INDDUCED   PICKUP WINDS (sharing common core) and they will for sure make lots and lots of power especially all combined together too....

BUT will only "make" this power if you put DIODES on those winds and guide this energy into a cap bank for example where to put it (what Pierre does!)

No diodes, you will see nothing!

My personal perhaps warped opinion is that THIS is where the huge amount of energy produced in Pierres DZ generator comes from (my theory sorry)

Pulse one coil - induce 4 or 5 in

concsequence but it is not a consequence it is a reinforcement of energy created   and this energy fills up the cap   bank, keeps it looping with no more input, once you also pre0charge the cap bank.

ALSO

There is a "combination" (coherence as Ole will call it) of the induced pickup winds, and the backemf/recoil/flyback energy - this makes the caps fill up even faster....

Anyways in conclusion it looks like to me the diodes are being completely ignored in Pierre's build, and I will guess everyone is thinking the internal diodes in the H-bridges are doing this work, but actually they will not do shit (sorry) they are only there to be a protection mechanism, not a steering mechanism of this energy....

Try to pull out the backemf/recoil with any transistor or mosfet without some fast steering diodes guiding into caps....

Try tp put any emergy into caps with induced pickup winds that share common core with primary pulse??

All will be fruitless without the diodes into caps.

=============== 23 May 2018 ===============

Pierre told me that some time ago most of the relays on his device caught on fire. The stator is fine but all the rest is fried.

So even if he would of been opened to having someone go for a live demo he had nothing to show. Lucky for us he made the videos before the fire.

So you are right, those who chose relays will only have a certain amount of time before your relays contacts fail and weld shut and possibly catch fire.

Pierre decided to rebuild with improvements but that will take take him months. Meanwhile we are free to use what he shared and from our tests we can try to find the missing ingredient.

Jerdee lives very close to me and we've been working together as a team. He does the programming and research (think tank) and I build and test.

Working together we think we may of found something but I have to rebuild using the 40 Amp Dual Half Bridge to test if it works as Jerdee thinks it will.

So nothing is certain but for sure there are clues to be found which Jerdee has spent weeks researching.

We will give an update when we can but we also have to respect Pierre's wishes which is to protect him and his discovery from theft. So we should not publicly share until we know Pierre has protection in place which I will know when it is and notify the group. After that, Pierre or any of us (who have figured it out) can publicly share.

I hope you and everyone else building Pierre's device understand and respect his wishes.

En.   Why repair and put effort on something that worked but was not able to last plus had other problems. I have no patience to rebuild the same prototype with the same problems. Since I designed and built the first functional prototype I know very well how to replicate it and will improve all the weak points that failed. That's why I decided to go ahead with an improved version that's better than the first dz.

I must say there wasn't much good parts left of my first prototype and since I'm not wealthy I tried to reuse parts tying my best to manage with what I have available. It will take some time to complete as I explain to Luc. Plus the fact that I work between 50 and 60hrs a week on top of having children to take care of plus the normal maintenance work of my house. Unfortunately I'm not retired yet and I havn't invented a device to lengthen days. I work on the dz when ever I have some free time, so it's going to take some time to complete. It wouldn't be possible for me to work on the dz full time. My first dz took almost two years to complete but I was starting from scratch that time. Now I have the foundation for the second device.

I know it's a great discovery but I won't let it affect my lifestyle. I've had a heart attack in the past   and if I don't want that to happen again, I need to maintain myself. I already gave a lot.   I only sleep 5 to 6hrs a night. So if you don't want me to leave this world and bring my secrets with me, I have to work at my own pace.   I hope to finish the new dz as soon as possible but I need time and I don't believe that some months will change things in the end. Thank you for your understanding. I would like to give a big thank you to Luc because no doubt he works very hard to take care of the overunity site and translates my comments in addition to the projects that he works on and I know that it is not always easy for him.

Thank you Luc.   See you soon   Pierre C.

====================

En.   Good evening jerdee!   Yes. it would be better to have a tri-phase, except the link you posted is about a motor with a rotating magnet. Of course you can make the current like that, except my principle is the reverse. It's the field of the stator that rotate. The 6 rotating fields was just to have a higher frequency because the relays would not give me a high enough frequency. By having 6 rotating fields it

multiplies the relay frequency by 3.   If I would of had some mosfet's like Luc I could of used only 2 poles and could of easily had the desired frequency and probably better results than I obtained because the field would have been 3 times stronger then 6 fields.

--- 19 June 2018 ---

Hi all

From my experience rotating permanent magnets sweeping past   coils show very strong magnetic force at the leading edge (front) and at the magnets trailing edge (back)

This is where the power is ....at the edges of the magnet.

Also there is no power produced at all if the coil does not have resistive load....and now Lenz law appears too, and it now might take 8 or 10 times more power to be rotating the magnet past than what the coil produces in power...depends on the resistive load a lot. And Airgap and magnet strength size and speed of rotation decide how bad Lenz law braking effect will be.

One way to prevent Lenz law is don't apply direct resistive load to coils instead fill caps-only then have 2nd stage where only caps hit the load for power. Cap size and resistance of load important plus frequency and speed of filling and discharging caps is important

Another way to prevent Lenz law is to not rotate the permanent magnets but rotate a field of electromagnets which is the dz generator more or less.

Hope.some.of this helps a little bit in thinking DZ generator how it works how to simulate sweeping permanent magnets electronically.

One other thing it is possible to chop narrow pulses of the generator coils and fill caps at sinewave peaks

....why fill caps at sinewave decension period... caps will not fill up anymore than what is at peak...

Also the coil can be briefly shorted at peak with fwbr ac side across coil and   the dc side of fwbr filling caps....now caps fill much faster and will go x20 or x50 more voltage

...release caps to load by themselves in two stage type output

----

En. Congratulations konehead, you've just visualized everything!   only need to replicate what you mention and it's not so easy to do but it's doable.   The proof is my first prototype that ran momentarily before burning several relays but it to worked.

Just replicate what you mention in a more efficient way than my first dz. Especially so that it can withstand long run times and Bingo you'll have it.

After that, the hardest thing that will happen is to fight the multinationals and governments who really won't   like the prospective of losing the energy monopoly.   You probably know that the global economy is based on energy.   So I fear a little in seeing the consequences this can create but I know that such an invention has a good side, like eliminating pollution, electric nuclear reactors, petrol cars, the electrical grid, wind generators etc.   In fact, only the future will tell what awaits the DZ generator. Pierre c.

=================Here is the answer to the rotor flux problem.

My M16 steel stator was saturating at 1.5T(purple color on previous posted plots).A linear B/H plot from FEMM demonstrates this below.

If I apply a stator model equivalent to Pierres stator i.e. 33% pole length 66% solid steel, then 1.4T is achieved with only 5A through the coils, without any other input on the rotor.

The attached flux plot shows 1.4T or 2.8T swing with 1A of counter flux representing a 236W load.There is also more flux capacity on the stator and rotor.

Pierre's   stator is unusually thick and I think we will not be able to replicate a working device without the thicker stator.

No matter how the flux is generated, if the stator steel saturates, the flux will not be fully coupled to the rotor.
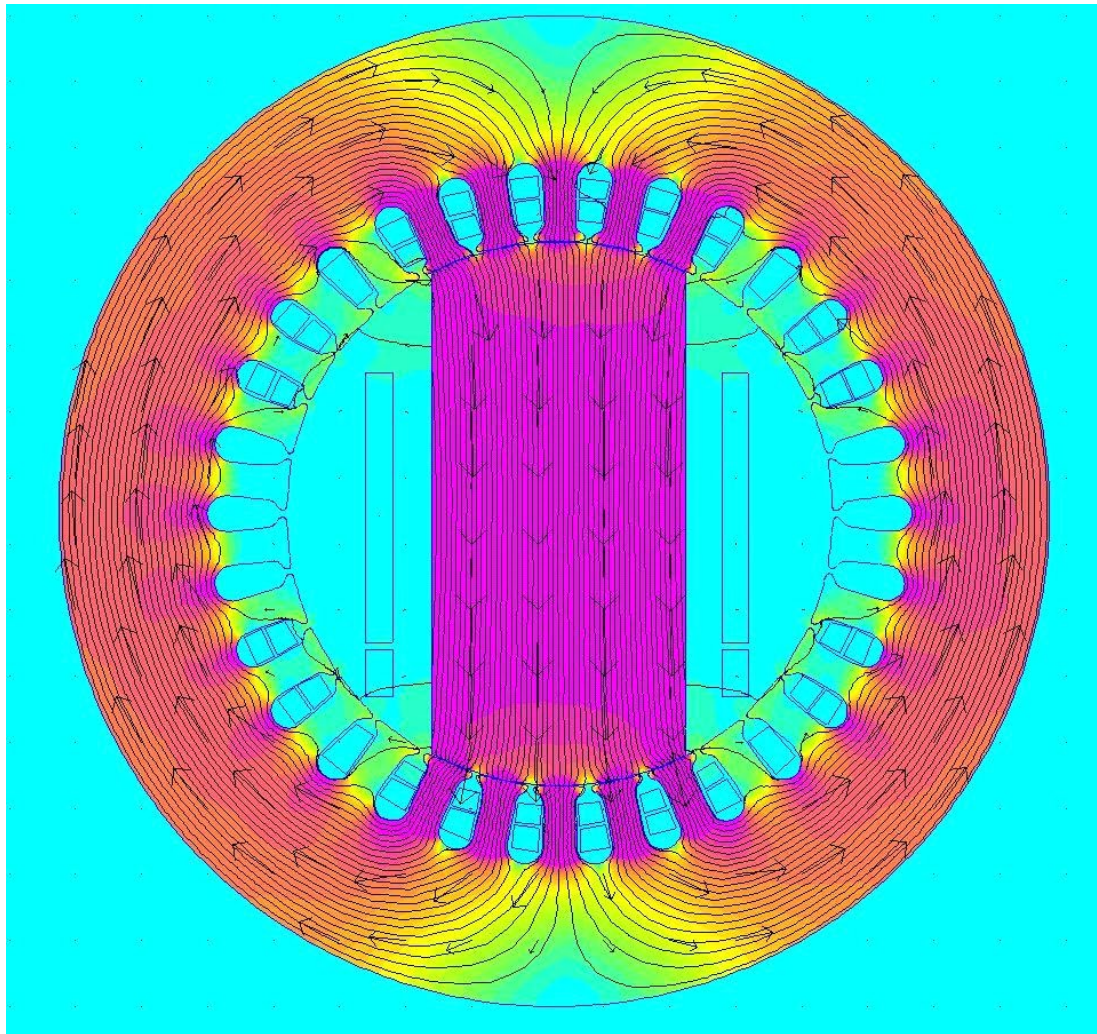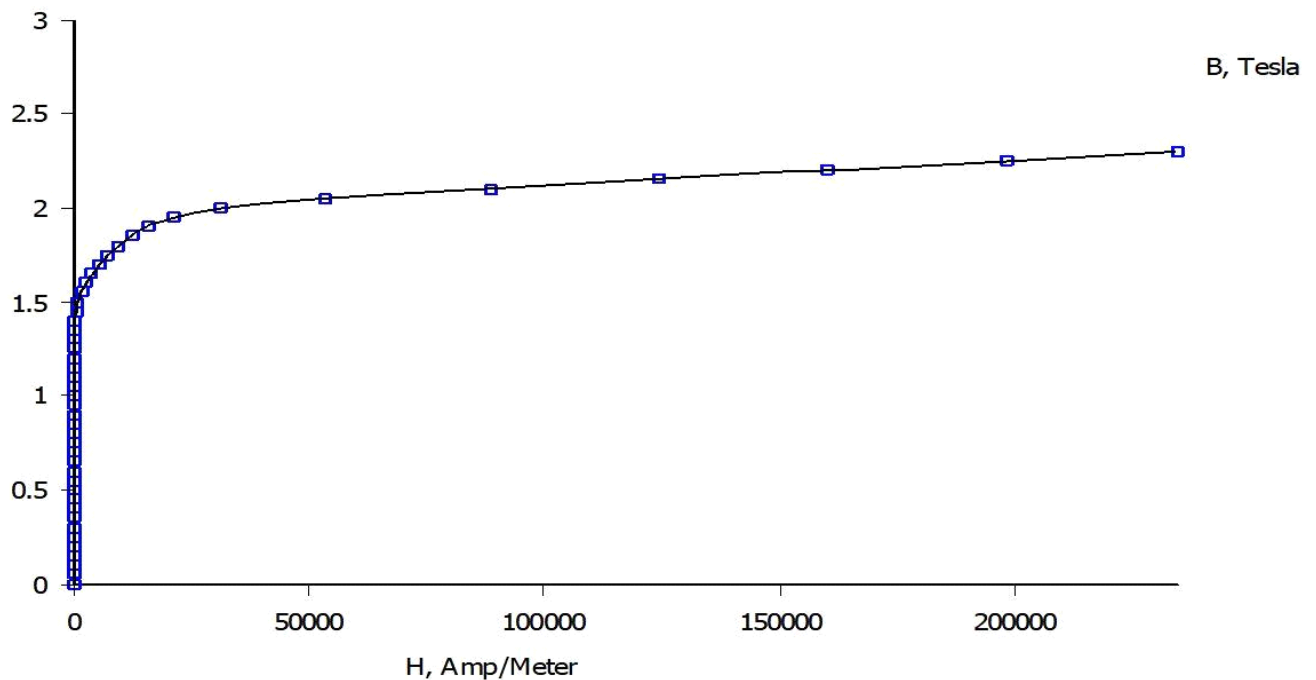
I adjusted the proportions of my FEMM model to more accurately represent my stator.

Driving the 2 x 7 coils at 1.5A average (this would be a much higher pulse current) each, brings the stator up to 1.5T and the start of saturation for M15 steel.

I modeled a 0 deg and a 90 deg case (note: coils relative to rotor). Easier to fix the coils and move the rotor. In the 90 deg case you can see that there is no flux through the rotor, only a very low air coupled flux (0.045T) through the side of the rotor.

Also attached, shot of coil composite coil current (dark Blue) & output EMF (loaded). Used a large current sensor around the coil cables.

Further measurement shows that there is a primary power supply current draw that is coincident with this period but does not show up coincident with the composite coil current.

I will start working through the individual bridges to determine which ones are responsible for this.
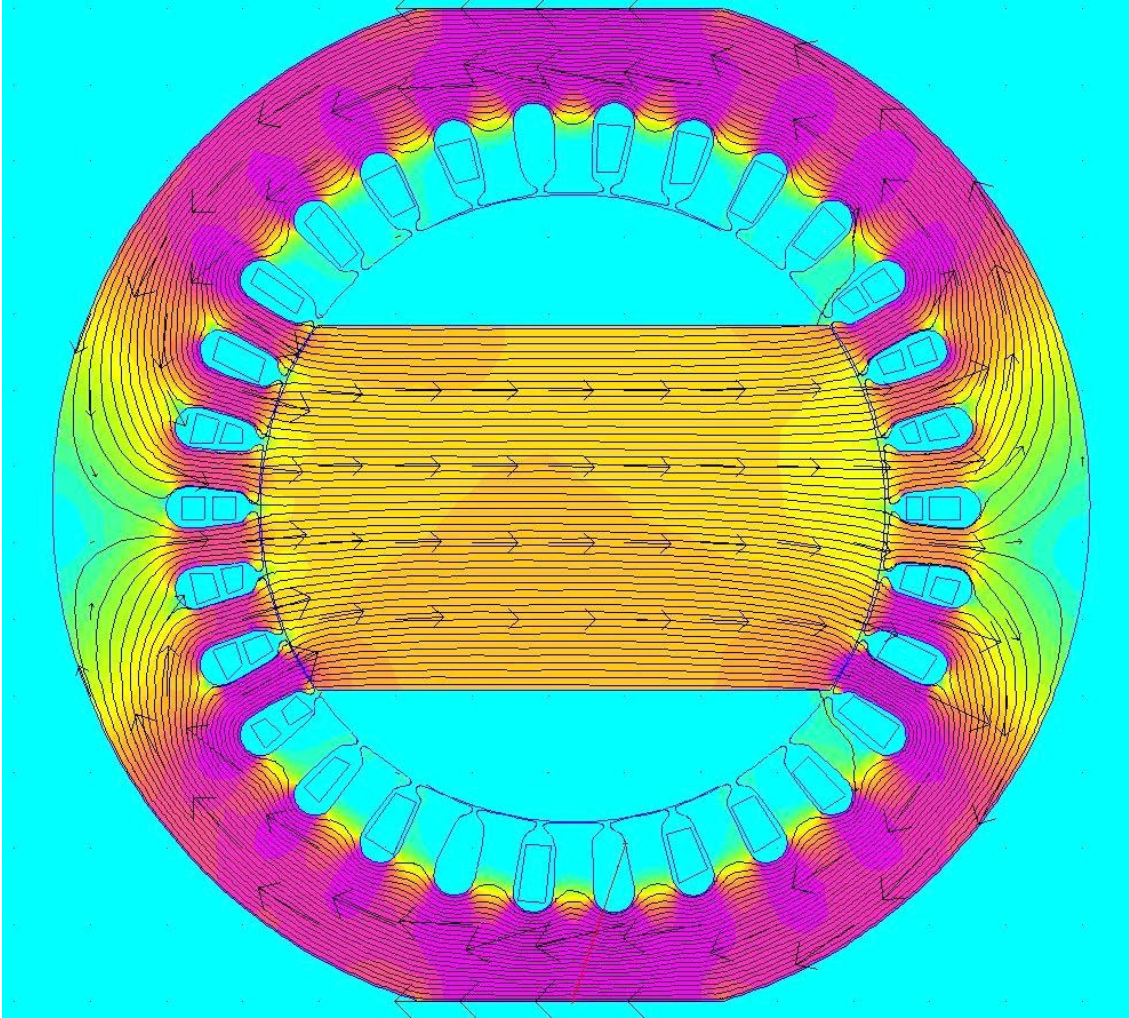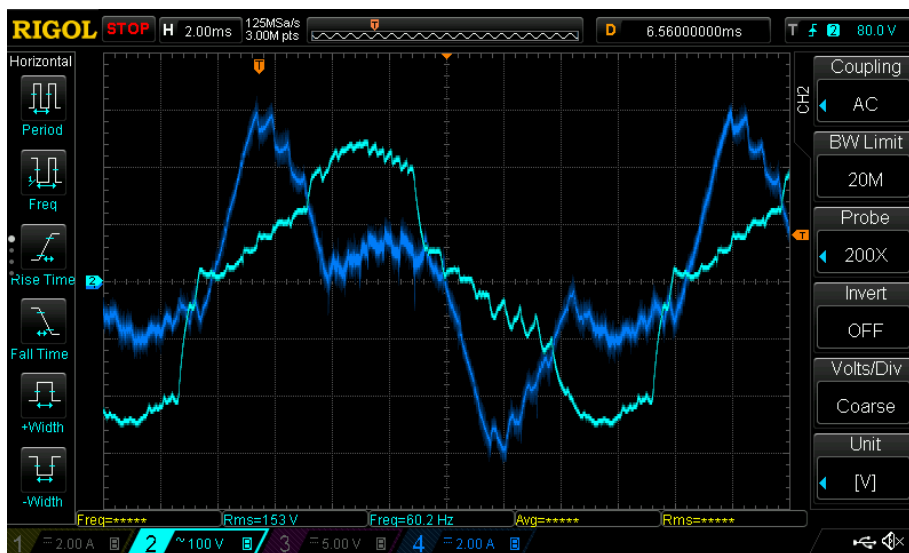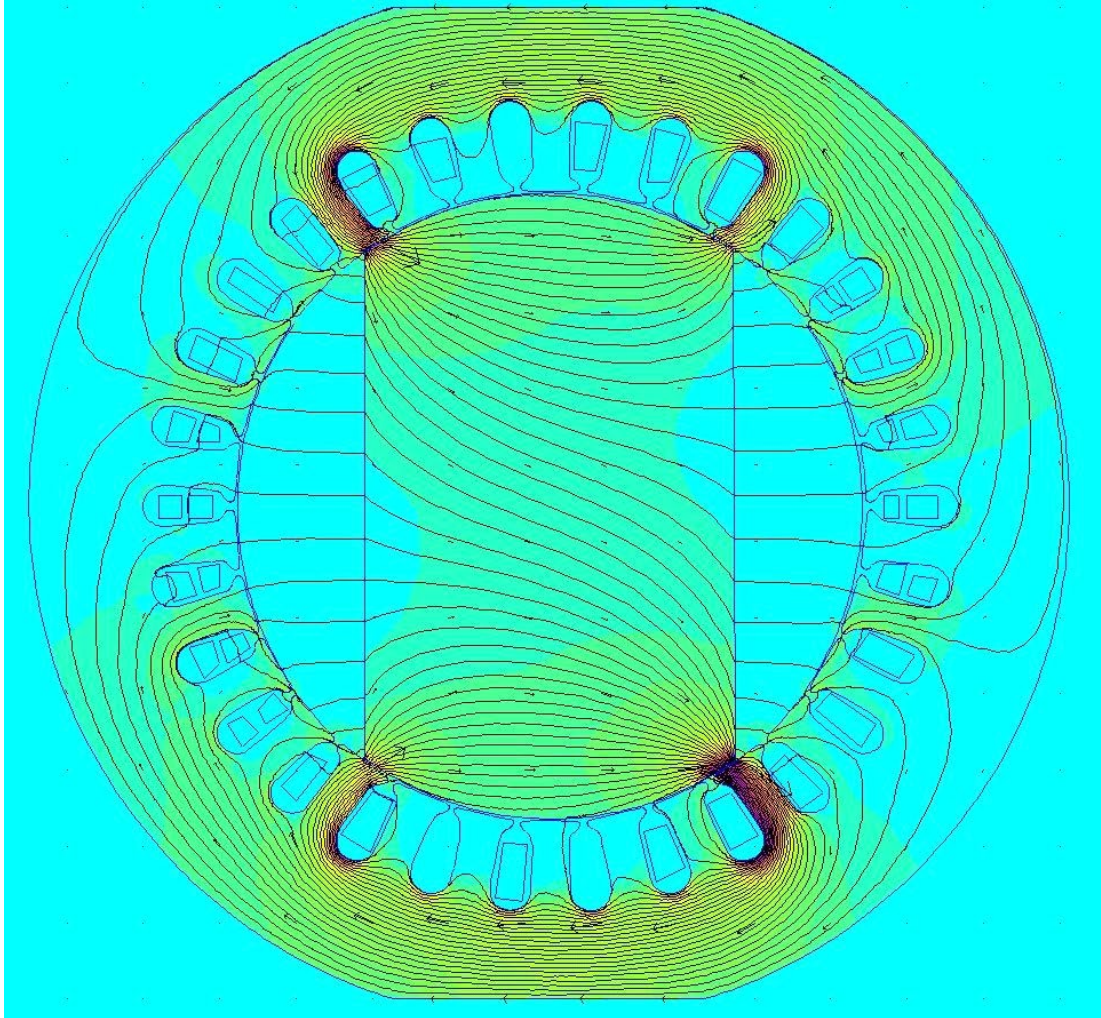
ttached is a slightly improved waveform, more symmetrical which produces a fairly good sine when filtered.

This was achieved by fine   adjustments in code delays. I achieve about 150W maximum output with this waveform but for about 450W DC input, so pretty inefficient.

A vast amount of power is being dissipated as coil heat.   You can see from the voltage and current waveforms that the 60Hz wave front is reasonable well formed.

I checked the coil currents for those in registration with the rotor and those at 90 degrees to the rotor. Interestingly, I found both to be almost identical in amplitude, not what I was expecting, as the coils not in registration should be of lower inductance due to lack of the rotor flux coupling.

This implying that a varying flux is established in the stator that is fairly uniform and that the rotor does not offer an appreciably lower reluctance path, as far as this flux is concerned.In an AC   generator with a DC energized rotor, there is a point coupling of flux onto the stator teeth in registration , it is a single flux that does not compete with any other flux (ignoring lenz for the moment).

The   flux spreads out around the stator in both directions and is at right angles to all of the stator coils, only where the rotor flux couples with the stator teeth, are those coils subjected to a changing flux that produces induction.

Is this flux totally uniform or is there something happening in the direction of rotation along the stator?

Well yes there is and this is normally referred to as armature reaction, see attached diagram.

Note how the flux crossing point from rotor to stator is skewed in the air gap, the lines of flux trail behind the turning rotor.

In an AC generator, the effect of armature reaction depends on the power factor i.e the phase relationship between the terminal voltage and armature current. Reactive power (lagging) is the magnetic field energy, so if the generator supplies a lagging load, this implies that it is supplying magnetic energy to the load. Since this power comes from excitation of the rotor, the net reactive power gets reduced in the generator. Hence, the armature reaction is demagnetizing.

Similarly, the armature reaction has magnetizing effect when the generator supplies a leading load (as leading load takes the leading VAR) and in return gives lagging VAR (magnetic energy) to the generator.

As in this case both armature flux and rotor flux lead, induced emf E by 90o, it can be said, rotor flux and armature flux are in the same direction. Hence, the resultant flux is simply arithmetic sum of rotor flux and armature flux.

In case of a purely resistive load, the armature reaction is cross magnetizing only.

Now, how much of the above is reproduced   in the DZ?
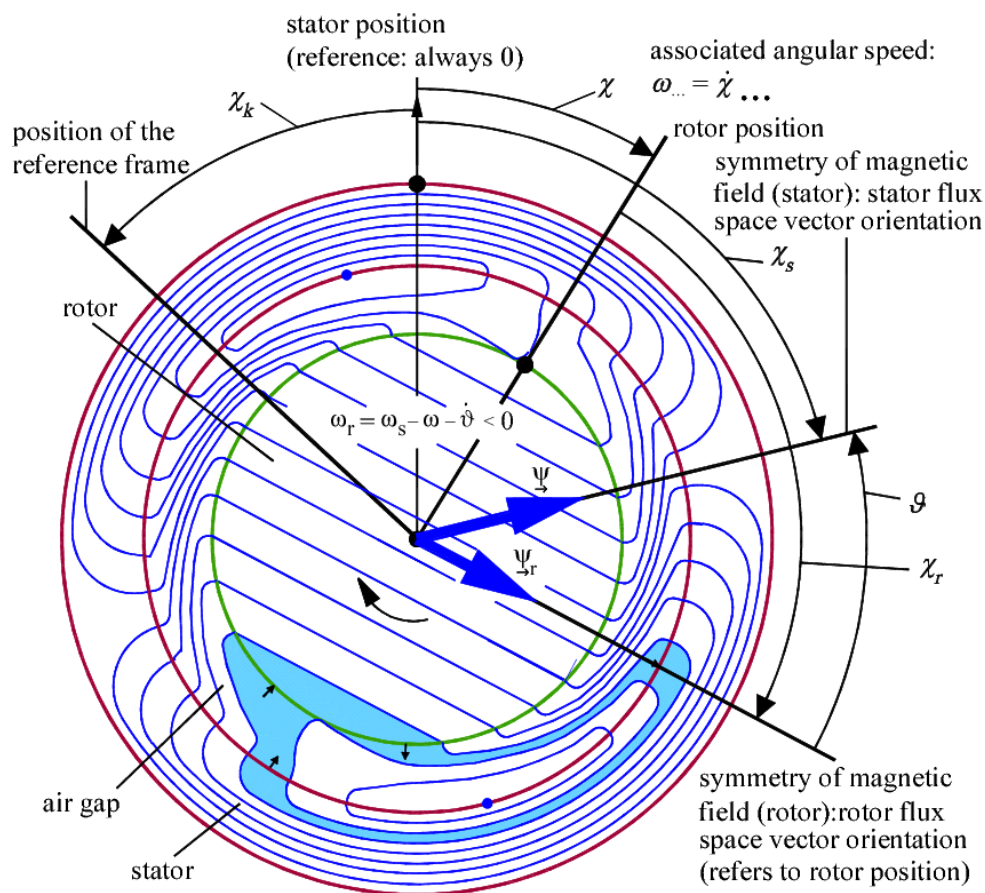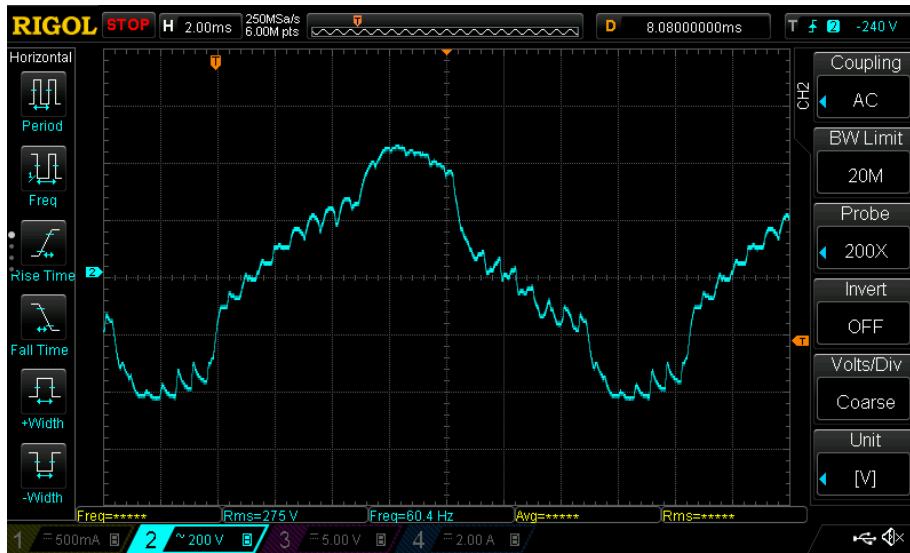
Simple answer to the coil current issue.

As I am running the coils in series albeit each coil is also in parallel with an opposite coil, of course the current is shared, so the coils not in registration do not contribute to output but have limited inductance and their resistance will consume power.

Compare this to a generator with rotating flux source and you find that it is similar, with several stator

coils in series forming a distributed winding (to produce a better sine). So some of these coils will be out of registration and will only have limited inductance and losses due to resistance.

13 Nov 18

Here is a quick check on my current build performance.

INPUT	OUTPUT	RECOVERY
117.5W Av	29W Av	32W Av@25V
224W Av	59W Av	67W Av@35V
373.5W Av	91.7W Av	110W Av@45V

Note: @50V, the max output with the same load used for these tests, is about 138W

The boost recovery circuit is working reasonably.

The ratio of input to output is constant.

A check on flux linkage using 7 + 7 coils in series/parallel energized with 50Hz sine indicates about 0.89%, (close to reasonable transformer operation), which ties up with the recovery to output ratio.

Obviously the input current is way too high, likely due to holding coils on for 7 steps.

It all comes down to generating the advancing MMF wave front in the stator with the overlapping coils.

The low turns value for each coil (35) in combination with parallel coil operation is required to get current to rise in the coil within the step period. This is at odds with what is required once a flux level has been established in the stator. The flux could be maintained by coils with a large number of turns and a lower current however, these coils would be problematic as they would present a high inductance to overcome to rise the current during the step period.

If a sine current were applied to these 35T coils to establish flux linkage   i.e 100W input (88W output), then a pulsed composite signal is established on top of the sine current, synchronized in phase.

We know that at least 80% of this pulse portion of the output could be recovered.

We also know that recovery energy is proportional to load, so the real question with this scenario, is what happens to the overall input current when it is always increasing or decreasing due to the slope of the sine i.e there is no time that current is static?

Note: this would not be using half sines as the input voltage for the switches.

I have not figured out how to apply the current sine to the coils yet.

----

lux cutting occurs in a generator with a physically rotating rotor, irrespective of armature reaction.

The rotor flux flows into the stator and distributes left and right around the stator. As the rotor   moves across a given coil the flux level increase and cuts both edges of the coil however, it is the increase in

flux density across the coil that causes induction, as this is the effective movement into the coil (right hand rule applies). If the rotor was static, flux cutting could still occur if the flux level in the rotor was changed by a rise and fall in rotor coil current.

In the DZ, the flux from a stator coil enters the rotor but there is no path at 90 degrees to the rotor coil, so the flux passes through the center of the coil, changing the flux density hence producing induction by flux linkage.

Flux cutting offers no advantage over flux linkage.

Basically the device is a rotary transformer but not an efficient one in my current configuration.

If you take the same coils, in this case 11 connected in series, with the inverse parallel windings (a further 11 coils 180 degrees apart) in place and apply say 30V 50 or 60Hz AC, you get over 85% of input coupled to the rotor coil. This demonstrates that the magnetic circuit is capable of achieving normal transformer efficiencies.

This indicates that I am still way off base in a stepped synthesis of a sine.

This is as much about coupling energy, as it is about not wasting energy. There is way too much power being dissipated as heat in the coils.

L192

================
https://overunity.com/17653/pierres-170w-in-1600w-out-looped-very-impressive-build-continued-moderated/msg558703/#msg558703 ======

19 June 2021 - Listener192

It's a hoax. Logically it cannot work for all the reasons I provided previously.

Pierres new build tries to keep the flux in the stator except when in registration with the rotor. This was the fundamental flaw of the original design, a lot of energised coils with low inductance and no flux coupling. The only result I achieved was with a static set of coils and a switching pattern that varied the current through those coils.. about 40% coupling. In the end I used bidirectional MOSFET switches to simulate the relays. A transformer supplied half wave supply was used to vary coil current as another method.

In a real generator the DC energised rotor is producing induction in the stator coils through 360 degrees. How can simulating the moving field in the stator hope to have any efficiency when only a small number of degrees of rotation will couple the stator to   the rotor. I use the word simulating rather than emulating as you cannot emulate a field moving through space with stepping discrete coils.

I don't believe that lenz aw can be circumvented, so on that premise .. where is the additional energy source with this device?

L192

===========================